Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels

by

Michael J. Neely

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

November 2003

Author . . . .
Department of Electrical Engineering and Computer Science
November, 2003

Certified by . . . .
Eytan Modiano
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels

by

Michael J. Neely

Submitted to the Department of Electrical Engineering and Computer Science
on November, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Satellite and wireless networks operate over time varying channels that depend on attenuation conditions, power allocation decisions, and inter-channel interference. In order to reliably integrate these systems into a high speed data network and meet the increasing demand for high throughput and low delay, it is necessary to develop efficient network layer strategies that fully utilize the physical layer capabilities of each network element. In this thesis, we develop the notion of *network layer capacity* and describe capacity achieving power allocation and routing algorithms for general networks with wireless links and adaptive transmission rates. Fundamental issues of delay, throughput optimality, fairness, implementation complexity, and robustness to time varying channel conditions and changing user demands are discussed. Analysis is performed at the packet level and fully considers the queueing dynamics in systems with arbitrary, potentially bursty, arrival processes.

Applications of this research are examined for the specific cases of satellite networks and ad-hoc wireless networks. Indeed, in Chapter 3 we consider a multi-beam satellite downlink and develop a dynamic power allocation algorithm that allocates power to each link in reaction to queue backlog and current channel conditions. The algorithm operates without knowledge of the arriving traffic or channel statistics, and is shown to achieve maximum throughput while maintaining average delay guarantees. At the end of Chapter 4, a crosslinked collection of such satellites is considered and a *satellite separation principle* is developed, demonstrating that joint optimal control can be implemented with separate algorithms for the downlinks and crosslinks.

Ad-hoc wireless networks are given special attention in Chapter 6. A simple cell-partitioned model for a mobile ad-hoc network with N users is constructed, and exact expressions for capacity and delay are derived. End-to-end delay is shown to be $O(N)$, and hence grows large as the size of the network is increased. To reduce delay, a transmission protocol which sends redundant packet information over multiple paths is developed and shown to provide $O(\sqrt{N})$ delay at the cost of reducing throughput. A fundamental rate-delay tradeoff curve is established, and the given protocols for achieving $O(N)$ and $O(\sqrt{N})$ delay are shown to operate on distinct boundary points of this curve.

In Chapters 4 and 5 we consider optimal control for a general time-varying network. A cross-layer strategy is developed that stabilizes the network whenever possible, and makes fair decisions about which data to serve when inputs exceed capacity. The strategy is decoupled into separate algorithms for dynamic flow control, power allocation, and routing, and allows for each user to make greedy decisions independent of the actions of others. The combined strategy is shown to yield data rates that are arbitrarily close to the optimally fair operating point that is achieved when all network controllers are coordinated and have perfect knowledge of future events. The cost of approaching this fair operating point is an end-to-end delay increase for data that is served by the network.

Thesis Supervisor: Eytan Modiano
Title: Associate Professor

Acknowledgments

I think that every scientist, engineer, and mathematician is, to some degree, a person of faith. It requires faith to believe that there exist simple and elegant solutions to difficult problems, as well as to believe that we are gifted with the capacity to discover them. I take faith a simple step further by believing that both the structured world we live in and the ability for us to understand it are gifts from God. Furthermore, I believe God has given us an even greater gift in Jesus Christ, that we might know him more deeply both now and throughout eternity.

Therefore, I thank God for providing me with all that I need, and for sustaining me throughout my graduate studies. I'm glad to have known his love and to have experienced his faithfulness even in times when my own faith is weak. I'd also like to thank all of my friends, both Christian and non-Christian, who have supported and encouraged me over the years.

I'd like to thank my advisor Eytan Modiano for his research example, guidance, and generous support. According to Eytan, a Ph.D. advisor is "an advisor for life," not just an advisor for the duration of graduate study. I greatly appreciate this openness, and would like to thank him in advance for all of his future advice. Thanks to Charlie Rohrs, who directed my masters thesis and who has continued to meet with me during the course of my Ph.D. The continued encouragements from Charlie have been very helpful. I'd like to thank Bob Gallager for his insightful comments and feedback as a thesis reader and committee member. I also want to thank him for his encouraging remarks about my masters thesis presentation several years ago (which happened to be when I first met Eytan). Thanks also to Vincent Chan, the final reader and committee member on my thesis. I still remember the time a few years ago when he said: "I think your research is pretty cool."

I'd like to thank my undergraduate research advisor at the University of Maryland, Professor Isaak Mayergoyz. He taught me the value in finding simple solutions to difficult problems, and throughout my time here at MIT I've tried to emulate his bold style of research.

Finally, I'd like to thank God for my darling wife Jean, whom I love. Thank you, Jean, for your support during my studies, and for your commitment to live and love along with me.

Contents

Chapter 1

Introduction

Satellite and wireless systems have emerged as a ubiquitous part of modern data communication networks. Demand for these systems continues to grow as applications involving both voice and data expand beyond their traditional wireline service requirements. In order to meet the increasing demand in data rates that are currently being supported by high speed wired networks composed of electrical cables and optical links, it is important to fully utilize the capacity available in satellite and wireless systems, as well as to develop robust strategies for integrating these systems into a large scale, heterogeneous data network. Emerging microprocessor technologies are enabling satellite and wireless units to be equipped with the processing power needed to implement *adaptive transmission techniques* and to make intelligent decisions about packet routing and resource management. It is expedient to take full advantage of these capabilities by designing efficient network control algorithms.

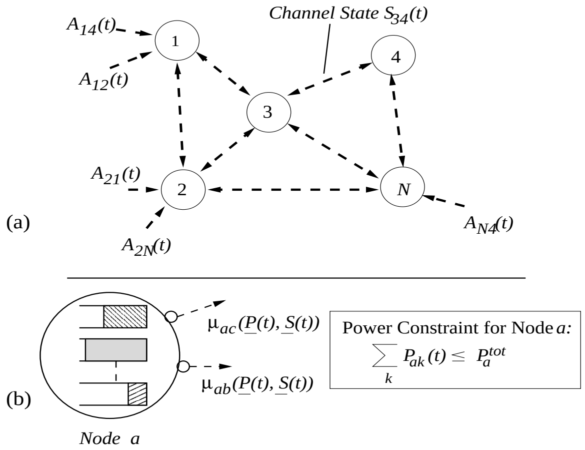
In this thesis, we develop algorithms for dynamic scheduling, routing, and resource allocation for satellite and wireless networks. Analysis is performed at the packet level and considers the complete dynamics of stochastic arrivals and queueing at each node of the network. Indeed, it is shown that queue backlog information is important in the design of robust network controllers which provide high throughput and low delay in the presence of time varying channels and changing user demands.

This research has immediate applications in the design and control of almost all modern data networks, including: multi-beam satellite downlinks, multi-satellite constellations (with RF downlinks and optical crosslinks), fully wireless ground networks, ad-hoc mobile networks, and hybrid networks with both wireless and wireline components in the air and on the ground. This variety of systems is studied through a unified framework, where network analysis and control is performed under the assumption that physical layer communication between network elements is characterized by a set of given (but arbitrary) link budget functions. This abstraction maintains a simple separation between network layer and physical layer concepts, yet is general enough to allow network control algorithms to be suited to the unique capabilities of each data link.

While it is important to understand the properties common to all data networks, it is also important to consider the effects that different physical layer characteristics have upon network design and control. For example, inter-channel interference is negligible or non-existent in traditional wireline networks, but is of primary importance in wireless networks and often necessitates careful coordination and scheduling among different users transmitting within close proximity to each other. Furthermore, note that although the channels for both satellite and wireless systems are time-varying, satellite channels are influenced by atmospheric conditions, scintillation, and the predictable motion of satellite orbits [29] [94] [1] [73] [28], while wireless channels are affected by different types of fading, interference, and user mobility [16] [96] [65] [117]. Such differences lead to dramatically different routing and power allocation algorithms. To address these specifics, special attention is given to multi-beam satellite downlinks in Chapter 3, and to ad-hoc mobile networks at the end of Chapter 4 and all of Chapter 6. Our analysis of satellite systems demonstrates significant performance gains achievable by dynamic power allocation, and our treatment of ad-hoc networks contributes to a growing theory of fundamental capacity and delay limits.

1.1 Problem Description and Contributions

Here we introduce the network model used throughout this thesis. For convenience, we consider any satellite transmitter or wireless user as *wireless node*, and a collection of such nodes communicating with each other forms a *wireless network*. Node-to-node communication depends on the channel conditions and power constraints of each transmitter. Consider the wireless network of Fig. 1-1 consisting of N power constrained nodes. Time is slotted, and every timeslot the channel conditions of each link randomly change (due to external effects such as fading, user mobility, and/or time varying weather conditions). Multiple data streams $A_{ij}(t)$ randomly enter the system, where $A_{ij}(t)$ represents an exogenous process of packets arriving to node i destined for node j. These arrival processes are arbitrary and represent *potentially bursty* data injected into the network by individual users. Packets are dynamically routed from node to node over multi-hop paths using wireless data links.



Channel State $S_{34}(t)$
$A_{14}(t)$
$A_{12}(t)$
1
4
3
$A_{21}(t)$
2
N
(a)
$A_{2N}(t)$
$A_{N4}(t)$
$\mu_{ac}(\underline{P}(t), \underline{S}(t))$
Power Constraint for Node a:
$\sum_k P_{ak}(t) \leq P_a^{tot}$
$\mu_{ab}(\underline{P}(t), \underline{S}(t))$
(b)
Node a

Figure 1-1: (a) A wireless network with multiple input streams, and (b) a close-up of one node, illustrating the internal queues.

Nodes can transmit data over multiple links simultaneously by assigning power to the links for each node pair (a, b) according to a power matrix $\underline{P}(t) = (P_{ab}(t))$, subject to a total power constraint at each node. Transmission rates over all link pairs are determined by the power allocation matrix $\underline{P}(t)$ and the current channel state $\underline{S}(t)$ according to a rate-power curve $\underline{\mu}(\underline{P}, \underline{S})$. Each node contains $N - 1$ internal queues for storing data according to its destination (Fig. 1-1b). A controller allocates power and schedules the data to be routed over the links in reaction to channel state and queue backlog information. The goal of the controller is to stabilize the system and thereby achieve maximum throughput and maintain acceptably low network delay.

We establish the *network capacity region*: The set of all input rate matrices (λ_{ij}) that the system can stably support (where λ_{ij} represents the rate of data entering node i destined for node j). This region is determined by considering all possible routing and power allocation strategies, and can be expressed in terms of the steady state channel probabilities, the node power constraints, and the rate-power function $\underline{\mu}(\underline{P}, \underline{S})$. We emphasize that this is a *network layer* notion of capacity, where $\underline{\mu}(\underline{P}, \underline{S})$ is a general function representing the rate achievable on the wireless links under a given physical layer modulation and coding strategy. This is distinct from the *information theoretic* capacity of the wireless network, which includes optimization over all possible modulation and coding schemes and involves many of the unsolved problems of network information theory [44] [33]. We do not address the information theoretic capacity in this work, and use the term *capacity* to represent *network layer capacity*.

We present a joint routing and power allocation policy that stabilizes the system and provides bounded average delay guarantees whenever the input rates are strictly inside the network capacity region. Such performance holds for general ergodic arrival and channel state processes, even if the specific channel probabilities and packet arrival rates are unknown to the network controller. The strategy involves solving an optimization problem every timeslot. We implement centralized and decentralized approximations of the algorithm for an ad-hoc wireless network, where channel variations are due to user mobility.

1.1.1 Related Work

Previous work on resource allocation for wireless systems is found in [111] [102] [132] [54] [136] [138] [61] [68] [108] [86] [110] [134] [71] [5] [124] [84] [39] [143] [131] [34] [41] [141] [47] [50] [113] [149] [72] [27] [70] [148] [57] [58]. Connectivity and asymptotic capacity analysis for large static networks is presented in [57] [58], and for mobile networks in [54]. The exact capacity of a wireless uplink and downlink with multiple users is developed in [136] [137] [138] [68], where it is assumed that all users have infinite backlog.

Optimization approaches to network resource allocation problems are developed in [148] [70] [34] [86] [92] [40] [76] [93] [12] [78] [69] [121]. In [148], a static routing and power allocation problem is considered for meeting network flow constraints, where link capacities are assumed to be convex functions of an aggregate link resource. In [70], various cost metrics are formulated as geometric programs to address resource allocation and quality of service in networks, again resulting in static resource allocations. Optimal power allocation for minimizing energy expenditure in a network with link-to-link rate requirements is considered in [34] under the assumption that transmission rates are linear functions of the signal to interference ratio on each link. In this case, although the network channels and rate requirements are constant, the optimal solution is not static but requires the computation of a periodic transmission schedule to achieve optimality. Simple approximations to optimal scheduling are developed in [41]. Game theoretic approaches and network pricing issues for wireless downlinks are developed in [86] [93] and for flow networks in [78] [92] [69], where pricing schemes are considered for achieving a static equilibrium with respect to some utility metric. The equilibrium computed in [69] is shown to be within a constant factor of the maximum utility. Similar constant factor bounds are developed in [121] for shortest path routing in static networks, where link costs are are assumed to be convex functions of an aggregate flow parameter.

We note that the optimization and game theoretic approaches of [148] [70] [78] [76] [92] [93] [40] [69] [121] do not consider the real effects of queueing in networks with randomly arriving traffic and potentially time varying data links, and do not provide implementable control algorithms for achieving the desired operating point. For example, queueing delay in [121] is modeled as a pure function of the data rate flowing over each link, using the M/M/1 approximation for steady state delay in queues with Poisson inputs. However, even if inputs to the network are Poisson, the internal queues under a dynamic network control policy will not be M/M/1, unless there is a long period of time between each control decision so that steady state averages can be achieved. The timescales over which the network is measured and the control decisions to take based on these measurements are important questions that need to be addressed.

Such questions fall into the regime of *network control*, where queue management, scheduling, and resource allocation decisions must be made in the presence of stochastic packet arrivals and time varying channel conditions. Control problems are addressed in [132] [133] [23] [5] [110] [141] [50] [47] [149] [124] [143] [63] [39] [71] [131] [108] [113] [111]. In [133], a stabilizing server allocation strategy is developed for a multi-user downlink with random inputs and ON/OFF channel states. Related problems of downlink scheduling are considered in [124] [5] [143], load balancing in cellular networks is treated in [39], and routing over finite buffer downlinks is considered in [110]. In [141], [47], optimal power allocation policies are developed for minimizing the energy expended to transmit data arriving to a downlink node with a single transmitter. In [149], a delay optimal strategy is developed for a multi-access uplink in systems with symmetric user parameters. Asymptotically optimal strategies using heavy traffic limits are developed in [99] [62] [124] for scheduling multiple users over a shared resource. Transmitter scheduling and power control for one-hop static networks are considered in [71], and one-hop networks with time varying topology are considered in [23] [131]. Much of our work is inspired by the approach of Tassiulas in [132], where a Lyapunov drift technique is used to develop a throughput optimal link scheduling policy for a multi-hop packet radio network. Further work on Lyapunov analysis is found in the switching and scheduling literature [95] [88] [109] [81], and a thorough exposition of stochastic systems and drift analysis is found in [98].

1.1.2 Contributions

The main contributions in Chapters 3 and 4 of this thesis are the formulation of a general power control problem for time varying wireless networks, the characterization of the network layer capacity region, and the development of capacity achieving routing and power allocation algorithms that offer delay guarantees and consider the full effects of queueing. These algorithms hold for systems with general arrival and channel processes, including ad-hoc networks with mobility. End-to-end delay is described in terms of a simple set of network parameters, and our analysis can be viewed as a *stochastic network calculus.*[1]

The contribution in Chapter 5 is the development of cross-layer techniques for controlling the network when input rates are outside of the capacity region. Decoupled algorithms for flow control, routing, and power allocation are constructed and the combined policy is shown to drive the network to within a specified distance of an optimally fair operating point. Such convergence is achieved without requiring users to coordinate with each other or to have any knowledge of the capacity region or network topology. The cost of closely approaching this fair operating point is an end-to-end delay increase for data that is served by the network. This work unifies notions of *network capacity*, *network optimization*, and *network control*.

The contribution in Chapter 6 is the development of a simple cell-partitioned mobile network for which *explicit capacity regions and end-to-end delay expressions* can be computed. This work for the first time presents a non-trivial model for which a (relatively) complete network theory of throughput and delay tradeoffs can be established.

Another contribution of this thesis is our treatment of time varying queueing analysis, where we extend and simplify known Lyapunov techniques to treat stability and delay in stochastic queueing networks with general *time varying server processes* and general, *potentially bursty arrival processes.*

[1] A non-stochastic network calculus was invented in [35], [36] for static networks with leaky bucket inputs and fixed routing.

1.2 Network Model

Consider the N node system of Fig. 1-1. We represent the channel process by the channel state matrix $\underline{S}(t) = (S_{ab}(t))$, where $S_{ab}(t)$ represents the current state of channel (a, b) (representing, for example, attenuation values and/or noise levels). Channels are assumed to hold their state for timeslots of length T (representing the coherence time of the channel), with transitions occurring on slot boundaries $t = kT$. We normalize the slot length so that $T = 1$ and all transition times t take integer values. It is assumed that channel states are known at the beginning of each timeslot. Such information can be obtained either through direct measurement (where timeslots are assumed to be long in comparison to the required measurement time) or through a combination of measurement and channel prediction.[2] The channel process $\underline{S}(t)$ takes values on a finite state space, and is ergodic with time average probabilities $\pi_{\underline{S}}$ for each state $\underline{S}$.

Every timeslot, a controller determines transmission rates by allocating a power matrix $\underline{P}(t) = (P_{ab}(t))$ subject to a total power constraint $\sum_{b \neq i} P_{ib}(t) \leq P_i^{tot}$ for all nodes i. Additional power constraints can be introduced, such as constraints on the number of outgoing links that can be activated simultaneously when a node is transmitting or receiving. It is therefore useful to represent the power constraint in the form $\underline{P}(t) \in \Pi$, where Π is a compact set of acceptable power allocations which include the power limits for each node.

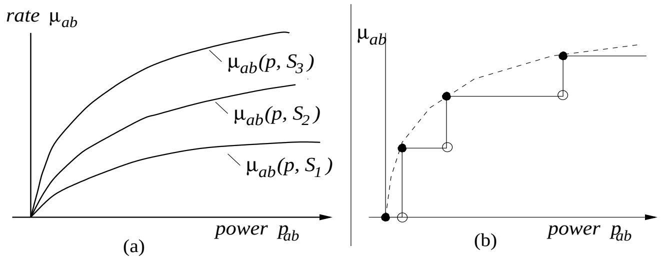


Figure 1-2: (a) A set of rate-power curves for improving channel conditions S_1, S_2, S_3, and (b) a curve restricted to a finite set of operating points corresponding to full packet transmissions. Curves illustrate behavior on link (a, b) when the single power parameter P_{ab} is increased, in which case the concave increasing profiles are typical.

Link rates are determined by a corresponding rate-power curve $\underline{\mu}(\underline{P}, \underline{S}) = (\mu_{ab}(\underline{P}, \underline{S}))$ (see Fig. 1-2). It is assumed that data can be split continuously, so that each timeslot the transmission rate μ_{ab} determines the number of bits that can be transferred over the wireless link (a, b). Such an assumption is valid if variable length packets can be split and

[2] Accurate prediction schemes are developed in [29] [28] [73].

re-packaged with new headers for re-sequencing at the destination (we neglect the extra bits due to such headers in this analysis). Alternately, splitting and relabeling can be avoided altogether if all packets have fixed lengths and the transmission rates μ are restricted to integral multiples of the packet-length/timeslot quotient L/T.

Note that, in general, the transmission rate over a link (a, b) of the network depends on the full matrix of power allocation decisions. This is because communication rates over the link may be influenced by interference from other channels. For example, achievable data rates could be approximated by using the standard CDMA signal-to-interference ratio in the log() formula for the capacity of a white Gaussian noise channel:

Example Rate-Power Curve: $\mu_{ab}\left(\underline{P}, \underline{S}\right) =$

$$\min\left\{\log\left(1 + \frac{\alpha_{ab}P_{ab}}{N_b + \frac{\alpha_{ab}}{G_1}\sum_{j\neq b}P_{aj} + \frac{1}{G_2}\sum_{i\neq a}\alpha_{ib}\sum_j P_{ij}}\right), \mu_{max}\right\} \tag{1.1}$$

where $G_1, G_2 \geq 1$ represent the CDMA gain parameters for signals from the same transmitter and different transmitters, respectively, and N_b and α_{ij} represent noise and fading coefficients associated with the particular channel state $\underline{S}$.

Alternatively, the $\mu_{ab}()$ curves could represent rate curves for a specific set of coding schemes designed to achieve a sufficiently low probability of error. Note that practical systems rely on a finite databank of codes, and hence may be restricted to a finite set of feasible operating points. In this case, rate-power curves are piecewise constant (see Fig. 1-2b). In general, we assume only that $\underline{\mu}\left(\underline{P}, \underline{S}\right)$ is a piecewise continuous function of power for each channel state $\underline{S}$.

More precisely, we assume the function is *upper semi-continuous*[3] in the power matrix, so that at points of discontinuity the limiting function value is less than or equal to the value of the function evaluated at the limit point. (see Fig. 1-2b).

The general rate-power curve description of a wireless link contains as a special case a wired link with fixed data rate, as the $\mu_{ab}(\underline{P}, \underline{S})$ function can take a constant value for all power levels. Note also that a broken or non-existent link can be modeled by a rate-power curve that is zero for all power levels at one or more channel states. Thus, the general power curve formulation provides the ability to address hybrid networks containing both wireline and wireless components.

[3] I.e., that $\lim_{\underline{P}\to\underline{P}^*}\mu_{ab}(\underline{P}, \underline{S}) \leq \mu_{ab}(\underline{P}^*, \underline{S})$ for all (a, b) and all $\underline{P}^*$ and $\underline{S}$ [15].

Let the backlog matrix $\underline{U}(t) = (U_{ij}(t))$ represent the unfinished work in node i destined for node j. The goal of the controller is to maintain low backlog and thereby stabilize the system. Throughout this thesis, we assume that centralized control is possible, so that the network controller has access to the full backlog and channel state matrices $\underline{U}(t)$ and $\underline{S}(t)$ every timeslot. Decentralized control where each node has limited information is considered in the final sections of Chapter 4 and in Chapters 5 and 6.

1.3 Comments on the Physical Layer Assumptions

The network model described above is quite general, although it contains several implicit assumptions. Here we describe the import of each of these assumptions.

1.3.1 The Time Slot Assumption

Timeslots are used to facilitate analysis and cleanly represent periods corresponding to new channel conditions and control actions. However, this assumption presumes synchronous operation, where control actions throughout the network take place according to a common timeclock. Asynchronous networking is not formally considered in this thesis, with the exception of the *Join-the-Shortest-Queue* Policy presented for finite buffer systems in Appendix B.

The assumption that channels hold their states for the duration of a timeslot is clearly an approximation, as real physical systems do not conform to fixed slot boundaries and may change continuously. This approximation is valid in cases where slots are short in comparison to the speed of channel variation. In a wireless system with predictable slow fading and non-predictable fast fading, the timeslot is assumed short in comparison to the slow fading (so that a given measurement or prediction of the fade state lasts throughout the timeslot) and long in comparison to the fast fading (so that a transmission of many symbols encoded with knowledge of the slow fade state and the fast-fade statistics will reach its destination and be successfully decoded with sufficiently low error probability).

1.3.2 The Error-Free Transmission Assumption

All data transmissions from one node to the next are considered to be successful with sufficiently high probability. For example, the link budget curves for wireless transmissions could be designed so that decoding errors occur with probability less than 10^{-6}. In such a system, there must be some form of error recovery protocol which allows a source to re-inject lost data back into the network [14]. If transmission errors are rare, the extra arrival rate due to such errors is small and does not appreciably change network performance. Throughout this thesis, we neglect such errors and treat all transmissions as if they are error-free. An alternate model in which transmissions are successful with a given probability can likely be treated using similar analysis.

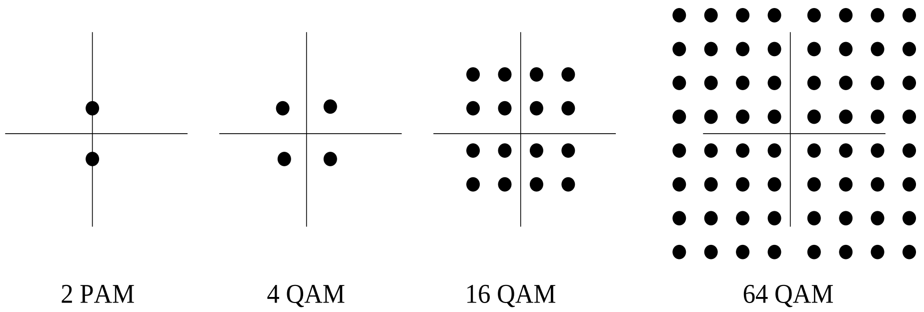
1.3.3 The Peak Power Constraint

The restriction of the power allocation matrix $\underline{P}(t)$ to the compact set Π is similar to a *peak power constraint*, where power must be contained within fixed bounds every timeslot regardless of previous transmissions. The set Π can be generalized to a time varying set $\Pi(t)$ representing the acceptable power levels at each timeslot t, although this does not change the fact that power must be held within pre-established limits on each and every timeslot. Such a constraint is realistic in cases when the electronics driving wireless transmitters must be operated within a certain power range. Furthermore, peak power constraints allow for deterministic guarantees on network lifetime. For example, in a wireless sensor network where nodes are deployed with a fixed amount of energy E_0, the network is guaranteed to last for at least E_0/P_{max} units of time, where P_{max} is the maximum transmission power of any node. Similarly, in a satellite system where energy for downlink transmission is renewed by solar radiation according to a periodic satellite orbit, the maximum power constraint can be selected as the power required to keep the satellite transmitting until the next update.

Much work in the area of wireless communication and wireless networking considers the alternate formulation of *average power constraints*, where energy can be stored and re-used later to either extend network lifetime or enable more powerful future transmissions. This is the approach in [47], where an optimal energy consumption strategy is developed for a satellite transmitter with both peak and average power constraints. A related problem of energy minimization is treated for static wireless channels in [141], where closed form expressions for the optimal power allocation rule are obtained and shown to be channel-independent if packet arrivals are fully known. The objective of minimizing energy in a wireless network in order to maximize network lifetime is treated in [41] [34] [72] [27]. In a stochastic setting, scheduling with such an objective turns network lifetime into a random variable. To ensure more predictable network lifetime guarantees while still optimizing over unused resources, it is perhaps desirable to take the combined approach of transmitting according to a fixed power budget, but updating this budget every few minutes or hours based on energy expenditure since the previous update. We note that if such an approach is used, the network can be viewed as operating according to a peak power constraint for the duration between each budget update.

1.3.4 Modulation Strategies for Transmitting with Power P

Here we provide a simple example to illustrate how a single wireless node can transmit data to another node at a rate $\mu(P, S)$ using power P while in channel state S for the duration of one timeslot. Suppose there is no external interference other than background white Gaussian noise at the receiver. Let N_S represent the noise power associated with channel state S. Furthermore, let α_S represent the attenuation between the transmitting node and its receiver under channel state S. The overall signal-to-noise ratio at the receiver is $\frac{\alpha_S P}{N_S}$.



modulation	bits/symbol	SNR/symbol
2 PAM	1	$0.25\ \Delta^2$
4 QAM	2	$0.50\ \Delta^2$
16 QAM	4	$1.25\ \Delta^2$
64 QAM	6	$5.25\ \Delta^2$

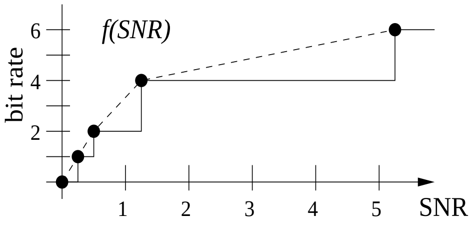
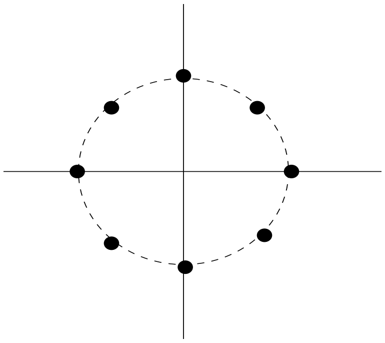


Figure 1-3: A piecewise constant rate curve for the 4 modulation schemes described above. Scaled power requirements are shown in the table, where Δ represents the minimum distance between signal points.

We assume in this example that the rate-power curve is a pure function of the signal-to-noise ratio and is designed so that transmission errors occur with probability less than or equal to 10^{-6}, so that we have:

$$\mu(P, S) = f\left(\frac{\alpha_S P}{N_S}\right)$$

where $f()$ is a piecewise constant function representing the rate achieved by each code within the databank of coding schemes known to the transmitter. Here we assume that each "code" corresponds to a simple QAM modulation strategy as shown in Fig. 1-3, and a given transmission during a timeslot consists of a long chain of symbols that conform to this modulation [85].



8 PSK

Figure 1-4: An illustration of the signal constellation for 8-PSK modulation.

Clearly, the power expended at the transmitter will fluctuate on a symbol-to-symbol basis. Such fluctuations are inherently part of any modulation strategy, with the exception of constant envelope strategies such as phase shift keying with k modulation points arranged on a circle of radius r (see Fig. 1-4). We do not concern ourselves with such symbol-to-symbol fluctuations, and define the transmission power P as *the average symbol power over a timeslot.* It is assumed that all symbol points are equally likely, so that P represents an average of the squared radius of each symbol of a given modulation scheme. For example, under the 2-PAM scheme with a symbol distance of Δ, we have $P = 0.25\Delta^2$. The corresponding power for 4, 16, and 64 QAM is given in the table of Fig. 1-3. The value of Δ depends on the signal attenuation α_S and the background noise N_S, and is chosen so as to maintain a sufficiently low probability of decoding error.

Rate-power functions that correspond to more complex physical layer coding schemes can also be considered. For example, in a system with one transmitter communicating over a single link, the system state S may describe both the background noise level at the receiver and the frequency response of the channel, where the state S includes a collection of attenuation coefficients for each bandwidth interval within a set of intervals comprising the frequency range of interest. In such a case, the transmission rate $\mu(P, S)$ may be determined by an optimal water pouring of the power P over each frequency interval [125] [85]. However, rather than delve into the details of physical layer modulation and coding, throughout this thesis we treat the physical layer as an abstraction that is represented solely in terms of a rate-power function $\underline{\mu}(\underline{P}, \underline{S})$.

1.4 Thesis Outline

In the next chapter we introduce the notion of queue stability and develop the queueing theoretic tools necessary to analyze wireless networks with bursty data and time varying server rates. In Chapter 3 we begin our analysis of optimal network control by investigating the problem of dynamic power allocation in a multi-beam satellite downlink. A joint problem of routing and power allocation is also treated in the chapter. We have published the results of this chapter in [108]. Readers interested only in the general network problem may skip this chapter and proceed directly to Chapter 4 without loss of continuity.

In Chapter 4 we consider the general network problem and establish the wireless network capacity region. Capacity achieving power allocation and routing strategies are developed for systems with both known and unknown arrival and channel statistics. Distributed implementations are considered in Section 4.4, where optimal distributed control is established for networks with independent channels, and a distributed approximation algorithm is developed for networks with interference. This distributed approximation is implemented for ad-hoc mobile networks in Section 4.5, where it is shown through analysis and simulation to offer higher data rates and lower delay than the Grossglauser-Tse relay algorithm of [54]. Satellite constellation networks with optical crosslinks and RF downlinks are considered in Section 4.6, where a *satellite separation principle* is developed that demonstrates joint-optimal control can be decoupled into separate algorithms acting on the constellation and on the individual downlinks. Finally, in Section 4.7 a perspective on *dynamic optimization* is provided by relating these optimal network control algorithms to an iterative solution of a static convex program. We have presented preliminary versions of this work in [111].

In Chapter 5, we consider the problem of optimal networking when data rates exceed the capacity of the network. In this case, it is not possible to serve all of the data, and optimally fair decisions about which data to serve must be made. We develop a simple flow control mechanism that operates together with the dynamic routing and power allocation strategies of Chapter 4. The combined cross-layer algorithm is proven to yield data rates that are arbitrarily close to an optimal operating point that lies on the boundary of the capacity region. Closeness to optimality is determined by a parameter affecting a tradeoff in average delay for data that is served by the network. The flow control and routing algorithms are decoupled from the power allocation decisions and can be implemented in a fully distributed manner. A natural pricing mechanism is constructed and shown to yield similar network performance in a scenario where individual users make greedy decisions independent of the actions of others.

In Chapter 6 we further explore ad-hoc networks with mobility. We impose a cell-partitioned structure on the network and compute exact expressions for network capacity. A capacity achieving control strategy is provided, and exact end-to-end delay under the given strategy is computed for a simplified i.i.d. user mobility model. It is shown that delay can be improved by orders of magnitude by considering schemes that transmit redundant versions of each packet, and a fundamental capacity and delay tradeoff curve is established. This chapter can be read independently of Chapters 2, 3 and 4. We have presented preliminary versions of this work in [106] and [104].

Miscellaneous queueing results on convexity and routing over finite buffer queues are presented in Appendices A and B, which can be read independently. We have presented the results within these appendices in [103] and [110].

Chapter 2

Analytical Tools for Time Varying Queueing Analysis

Here we develop the queueing theoretic tools necessary to analyze wireless networks. We begin with a precise definition of stability. Our definition extends previous definitions in [6] [98] [88] [81] [95] [132] and yields a simplified set of necessary and sufficient conditions that are useful tools for analyzing capacity and delay in time varying wireless networks.

2.1 Stability of Queueing Systems

Consider a single queue in isolation, with an input process $A(t)$ and a time varying server process $\mu(t)$. Because the input stream and server process could arise from an arbitrary, potentially non-ergodic routing and power allocation policy, our definition of queue stability must be robust to handle all possible arrival and server processes. Let the unfinished work function $U(t)$ represent the amount of unprocessed bits remaining in the queue at time t. This unfinished work function evolves according to a probabilistic model determined by the stochastics of the $A(t)$ and $\mu(t)$ processes. As a measure of the fraction of time the unfinished work in the queue is above a certain value V, we define the following "overflow" function $g(V)$:

$$g(V) = \limsup_{t\to\infty} \mathbb{E}\left\{\frac{1}{t}\int_0^t 1_{[U(\tau)>V]}d\tau\right\}$$

where the indicator function 1_X used above takes the value 1 whenever event X is satisfied, and 0 otherwise. The above limit[1] always exists, so that $0 \leq g(V) \leq 1$. In cases where the unfinished work only changes on slot boundaries, the integral above can be replaced by a discrete sum over timeslots.

Definition 1. *A single queue is* stable *if* $g(V) \to 0$ *as* $V \to \infty$.

Note that the expectation $\mathbb{E}\left\{\frac{1}{t}\int_0^t 1_{[U(\tau)>V]}d\tau\right\}$ in the $g(V)$ definition above is equal to $\frac{1}{t}\int_0^t Pr\left[U(\tau) > V\right] d\tau$. Hence, if sample paths of unfinished work in the queue are ergodic and a steady state exists, the overflow function $g(V)$ is simply the steady state probability that the unfinished work in the queue exceeds the value V. Stability in this case is identical to the usual notion of stability defined in terms of a vanishing complementary occupancy distribution (see [98] [6] [81] [88] [132] [95]).

Definition 2. *A* network *of queues is said to be stable if all individual queues are stable.*

2.1.1 On the $\limsup$ definition

The $\limsup$ definition of stability was carefully chosen because of its applicability to networks with arbitrary inputs and control laws. Indeed, the $\limsup$ of any bounded function always exists, whereas the regular limit does not. For the reader unfamiliar with such limits, we note that the $\limsup$ of a function is simply the limiting value of the maximum of the function. For example, the $\limsup$ of the cosine function is equal to 1 (likewise, the $\liminf$ of the cosine function is equal to -1). The $\limsup$ has the following important properties, which follow immediately from the definition:

- For any functions $f(t)$ and $g(t)$ satisfying $f(t) \leq g(t)$, the lim sup of $f(t)$ is less than or equal to the lim sup of $g(t)$.

- The lim sup of a sum of functions is less than or equal to the sum of the lim sups of the functions: $\limsup_{t\to\infty}\sum_k f_k(t) \leq \sum_k \limsup_{t\to\infty} f_k(t)$

Furthermore, the $\limsup$ is equivalent to a regular limit, and hence has all of the same properties, whenever the regular limit converges.

[1] Where the *lim sup* of a bounded function $f(t)$ always exists, and is defined: $\limsup_{t\to\infty} f(t) = \lim_{t\to\infty}\left[\sup_{\tau\geq t} f(\tau)\right]$. The *lim inf* is defined similarly.

While the intuitive notion of stability is clear, its definition was surprisingly difficult to capture mathematically. For example, a queue should certainly be considered stable if its backlog remains bounded for all time. However, it is possible for bounded queues to have non-ergodic variations in backlog, so that regular limits may not exist. Furthermore, stability should not be confined to systems with deterministically bounded queue sizes. Otherwise, no system with Poisson inputs would be stable.

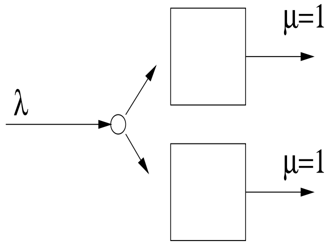


Figure 2-1: A 2-queue system with input stream of rate λ.

One might therefore consider defining stability in terms of a $\liminf$, or in terms of the backlog falling below a given threshold infinitely often. However, these definitions are also insufficient. Indeed, suppose stability were defined in terms of a queue emptying infinitely often. Consider now the simple 2-queue network of Fig. 2-1, where each queue has a constant service rate of $\mu = 1$. A single input stream of rate λ enters the system, and packets must be routed to either the top queue or the bottom queue upon arrival. Clearly, the stability region of such a network should be no larger than the set of all data rates λ such that $\lambda \leq 2$, as this is the maximum output rate of the system. However, under the stability definition of the queue emptying infinitely often, the stability region of the network in Fig. 2-1 becomes the set of all $\lambda < \infty$. This is achievable by the simple (and non-ergodic) policy of routing all data to one queue while the other empties, upon which time the routing is reversed. Using a $\liminf$ leads to a similar counter-example. However, the given definition of stability in terms of a $\limsup$ yields the correct stability region of $\lambda \leq 2$. That is, $\lambda \leq 2$ is a necessary condition for stability, and $\lambda < 2$ is a sufficient condition—provided that a mild additional assumption concerning boundedness of the second moment of arrivals also holds, as described in Section 2.2.[2]

[2] Stability at the point $\lambda = 2$ may or may not be achievable, depending on further details of the arrival process. Boundedness of the second moment is not imperative but facilitates analysis.

2.1.2 A Necessary Condition for Network Stability

Consider a network of N queues with unfinished work levels $U_k(t)$ for $k \in \{1, ..., N\}$, and define:

$$g_k(V) = \limsup_{t\to\infty} \mathbb{E}\left\{\frac{1}{t}\int_0^t 1_{[U_k(\tau)>V]}d\tau\right\}$$
$$g_{sum}(V) = \limsup_{t\to\infty} \mathbb{E}\left\{\frac{1}{t}\int_0^t 1_{[U_1(\tau)+...+U_N(\tau)>V]}d\tau\right\}$$

Lemma 1. *(Network Stability — Necessary Condition) For a network of N queues, we have:*

(a) $g_{sum}(V) \to 0$ if and only if $g_k(V) \to 0$ for all queues $k \in \{1, \ldots, N\}$.

(b) If the network is stable, then for any $\delta > 0$ there exists a finite value V for which arbitrarily large times $\tilde{t}$ can be found so that $Pr[\sum_i U_i(\tilde{t}) \leq V] > 1 - \delta$.

In particular, the probability that work in all queues simultaneously drops below the value V is greater than $1/2$ infinitely often (i.e., for the special case where $\delta = 1/2$).

Proof. (a) Note that for any queue $k \in \{1, \ldots, N\}$, we have:

$$1_{[U_k(t)>V]} \leq 1_{[\sum_i U_i(t)>V]} \leq \sum_i 1_{[U_i(t)>V/N]}$$

where the last inequality follows because the event $\sum_i U_i(t) > V$ implies that $U_i(t) > V/N$ for some $i \in \{1, \ldots, N\}$. Using this together with the fact that the lim sup of a sum is less than or equal to the sum of the lim sups, it follows that for all queues $k \in \{1, \ldots, N\}$, we have:

$$g_k(V) \leq g_{sum}(V) \leq \sum_{i=1}^N g_i(V/N)$$

Taking limits as $V \to \infty$ proves part (a).

Part (b) follows from (a) by noting that stability implies there exists a value V such that $g_{sum}(V) < \delta$ for arbitrarily small δ. By definition, $g_{sum}(V) = \limsup_{t\to\infty} \frac{1}{t}\int_0^t Pr[\sum_i U_i(\tau) > V]d\tau$. Hence, for any arbitrarily large time t_1, there must be a value $\tilde{t} \geq t_1$ for which the integrand is less than δ. □

A more stringent definition of stability which defines the overflow function $g(V)$ without the expectation could also be used, as in [111]. An analogue of Lemma 1 in this case shows that if the network is stable then the unfinished work in all queues simultaneously drops below some threshold value V infinitely often, with probability 1. It turns out that both definitions of stability lead to the same network capacity region (defined in Chapter 4), although the definition provided here which incorporates an expectation is slightly easier to work with and is used throughout this thesis.

2.1.3 A Sufficient Condition for Network Stability

The necessary condition given in the section above is a used to establish the network layer capacity region described in Chapters 3 and 4. A sufficient condition is also required, and for this we extend a well developed theory of Lyapunov drift (see [98] [81] [6] [88] [132] [95]). Consider a network of N queues operating in slotted time, and let $\underline{U}(t) = (U_1(t), \ldots, U_N(t))$ represent a row vector of unfinished work in each of the queues for timeslots $t \in \{0, 1, 2, \ldots\}$. Define a non-negative function $L(\underline{U})$ of the unfinished work vector $\underline{U}$. We call $L(\underline{U})$ a *Lyapunov function*. The lemma below combines the Lyapunov stability analysis presented in [6] [98] and the delay analysis in [88] into a simple and new statement useful for stability and performance analysis in a wireless network.

Lemma 2. *(Network Stability — Sufficient Condition using Lyapunov Drift) If there exists a positive integer K such that for all timeslots t, the Lyapunov function evaluated K steps into the future satisfies:*

$$\mathbb{E}\left\{L(\underline{U}(t+K)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \leq B - \sum_i \theta_i U_i(t) \tag{2.1}$$

for some positive constants B, $\{\theta_i\}$, and if $\mathbb{E}\left\{L(\underline{U}(t_0))\right\} < \infty$ for $t_0 \in \{0, 1, \ldots, K-1\}$, then the network is stable, and:

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \left[\sum_i \theta_i \mathbb{E}\left\{U_i(\tau)\right\}\right] \leq B \tag{2.2}$$

The fact that Lyapunov drift is compared after K slots (rather than after a single slot) is required for systems that approach steady state only over a long period of time. Similar K-slot analysis of Lyapunov drift has been used in [131], and similar drift statements for i.i.d. systems where $K = 1$ are found in [81] [88] [95] [6] [98]. To our knowledge, the statement above is the strongest known sufficient condition applicable to time varying wireless networks, and yields the following simple and self-contained stability proof using the machinery of the $g(V)$ function together with the telescoping series approach taken in [88].

Proof. Consider (2.1) at times $t = mK + t_0$, where $t_0 \in \{0, \ldots, K-1\}$. Taking expectations of this inequality over the distribution of $\underline{U}(mK + t_0)$ and summing over m from $m = 0$ to $m = M - 1$ creates a telescoping series, yielding:

$$\mathbb{E}\left\{L(\underline{U}(MK + t_0))\right\} - \mathbb{E}\left\{L(\underline{U}(t_0))\right\} \leq BM - \sum_{m=0}^{M-1} \sum_i \theta_i \mathbb{E}\left\{U_i(mK + t_0)\right\}$$

Dividing by M and using non-negativity of the Lyapunov function, we have:

$$\frac{1}{M} \sum_{m=0}^{M-1} \sum_i \theta_i \mathbb{E}\left\{U_i(mK + t_0)\right\} \leq B + \mathbb{E}\left\{L(\underline{U}(t_0))\right\}/M$$

The above inequality holds for all t_0. Summing over $t_0 \in \{0, \ldots, K-1\}$ yields:

$$\frac{1}{M} \sum_{\tau=0}^{MK-1} \sum_i \theta_i \mathbb{E}\left\{U_i(\tau)\right\} \leq KB + \sum_{t_0=0}^{K-1} \mathbb{E}\left\{L(\underline{U}(t_0))\right\}/M$$

Dividing by K and taking the lim sup of the above inequality as $M \to \infty$ yields the performance bound (2.2).

To prove stability, note the performance bound implies that for any queue i:

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{U_i(\tau)\right\} \leq B/\theta_i$$

Now considering the overflow function $g_i(V)$, we have:

$$\begin{aligned} g_i(V) &\triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{E}\left\{1_{[U_i(\tau)>V]}\right\} \\ &\leq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{E}\left\{U_i(\tau)/V\right\} \qquad (2.3) \\ &\leq \frac{B}{\theta_i V} \end{aligned}$$

where inequality (2.3) follows because $1_{[U_i>V]} \leq U_i/V$ for any non-negative random variable U_i. Taking limits as $V \to \infty$ shows that $g_i(V) \to 0$ and proves stability. □

For notational convenience, we define:

$$\overline{\sum_i \theta_i U_i} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_i \theta_i \mathbb{E}\left\{U_i(\tau)\right\}\right]$$

so that the Lyapunov drift condition of Lemma 2 implies $\overline{\sum_i \theta_i U_i} \leq B$. In Chapter Appendix 2.A we state conditions under which the limit of time average backlog in each node converges, so that the lim sup is equal to the regular limit and can be pushed through the sum, implying that: $\overline{\sum_i \theta_i U_i} = \sum_i \theta_i \overline{U}_i \leq B$. However, we do not require these convergence conditions, and throughout this thesis we use the more general expression $\overline{\sum_i \theta_i U_i}$. The reader can freely interpret this expression as a weighted sum of individual time averages whenever such time averages converge.

2.2 Delay Analysis Via Lyapunov Drift

It is illuminating to consider the impact of the above result on the study of a single queue with a general arrival process $A(t)$ (representing the number of bits that arrive during slot t) and a general server process $\mu(t)$ (representing the server rate at slot t). Both the arrival and server processes are assumed to be *rate convergent* with average rates λ and μ_{av}, respectively, as defined below.

2.2.1 Rate Convergence

Definition 3. *A process $A(t)$ is* rate convergent *with rate λ if:*

(i) $\frac{1}{t}\sum_{\tau=0}^{t-1} A(\tau) \to \lambda$ with probability 1 as $t \to \infty$

(ii) For any $\delta > 0$, there exists an interval size K such that for any initial time t_0 and regardless of past history, the following condition holds: $\left|\mathbb{E}\left\{\frac{1}{K}\sum_{k=0}^{K-1} A(t_0+k)\right\} - \lambda\right| \leq \delta$

The notion of rate convergence is similar to that of ergodicity, although it is a bit more general as ergodicity is often associated with stationary and non-periodic processes, while rate-convergent processes include any that are periodic and many that are non-stationary.

2.2.2 Generalized Bound for Average Delay

Consider again the rate convergent arrival and server processes $A(t)$ and $\mu(t)$ for a single queue. Suppose the new arrivals $A(t)$ are bounded in their second moments every timeslot, so that $\mathbb{E}\left\{A(t)^2\right\} \leq A_{max}^2$ for all t, regardless of past history. Likewise, suppose the server process is bounded so that $\mu(t) \leq \mu_{max}$ for all t. The following result can be viewed as a generalization of the well known P-K formula for average delay in an $M/GI/1$ queue (see, for example, [49], [14]).

Lemma 3. *(Backlog Bound for */*/1 Queues) For a single queue with rate convergent arrival and server processes $A(t)$ and $\mu(t)$ described above, if $\lambda < \mu_{av}$ then the time average unfinished work in the queue satisfies:*

$$\overline{U} \leq \frac{K(\mu_{max}^2 + A_{max}^2)}{\mu_{av} - \lambda} \tag{2.4}$$

where K is the smallest integer such that at every timeslot t and regardless of past history of the system, the following condition holds:

$$\mathbb{E}\left\{\frac{1}{K}\sum_{\tau=0}^{t+K-1}\mu(\tau) - \frac{1}{K}\sum_{\tau=t}^{t+K-1}A(t+k)\right\} \geq (\mu_{av} - \lambda)/2 \tag{2.5}$$

Proof. The unfinished work in the queue K slots into the future can be bounded in terms of the current unfinished work as follows:

$$U(t+K) \leq \max\left[U(t) - \sum_{\tau=t}^{t+K-1}\mu(\tau), 0\right] + \sum_{\tau=t}^{t+K-1}A(\tau)$$

The above expression is an inequality instead of an equality because new arrivals may depart before the K slot interval is finished. Squaring both sides of the inequality above, we have:

$$U^2(t+K) \;\;\leq\;\; U^2(t) + K^2\mu_{max}^2 + \left(\sum_{\tau=t}^{t+K-1}A(\tau)\right)^2 - 2KU(t)\left[\frac{1}{K}\sum_{\tau=t}^{t+K-1}\mu(\tau) - \frac{1}{K}\sum_{\tau=t}^{t+K-1}A(\tau)\right]$$

Taking expectations, noting that $\mathbb{E}\left\{A(\tau_1)A(\tau_2)\right\} \leq \sqrt{\mathbb{E}\left\{A(\tau_1)^2\right\}\mathbb{E}\left\{A(\tau_1)^2\right\}} \leq A_{max}^2$, and using the definition of K yields:

$$\mathbb{E}\left\{U^2(t+K) - U^2(t)\left|U(t)\right.\right\} \leq K^2\mu_{max}^2 + K^2A_{max}^2 - 2KU(t)(\mu_{av} - \lambda)/2$$

Applying Lemma 2 to the above inequality (using $L(U) = U^2$) proves the result. □

Thus, the unfinished work bound grows linearly with the parameter K, representing the number of timeslots required for the system to reach near steady-state starting from any initial condition. By Little's Theorem, it follows that the average bit delay $\overline{D}_{bit}$ satisfies $\overline{D}_{bit} \leq K(\mu_{max}^2 + A_{max}^2)/[\lambda(\mu_{av} - \lambda)]$. For an intuitive understanding of the tightness of the bound, consider a queue with Poisson inputs but with an ON/OFF server process, where the server rate is 1 in the ON state and 0 in the off state, and state transitions occur with equal probability δ every timeslot. The average server rate for this system is thus $\mu_{av} = 1/2$. Starting in the OFF state, the number of timeslots K required to achieve an expected time average service rate of at least $\mu_{av}/2$ is proportional to $1/\delta$, as is the expected waiting time to reach an ON state for a packet that arrives while the server is OFF. Because half of all packets arrive while the server is OFF, expected delay must also be proportional to $1/\delta$. Hence, expected delay must grow linearly in the K parameter, which is a property that is captured in the upper bound. We note that if arrivals and channel states are i.i.d. every slot, then $K = 1$, and the term $(\mu_{av} - \lambda)/2$ on the right hand side of (2.5) can be replaced by $(\mu_{av} - \lambda)$.

It is easy to see that $\lambda \leq \mu_{av}$ is necessary for queue stability, as otherwise the unfinished work would increase to infinity with probability 1.[3] Defining the *stability region* of the queue as the closure of all stabilizable data rates, it follows that the stability region is the set $\{\lambda \mid \lambda \leq \mu_{av}\}$. That is, $\lambda < \mu_{av}$ is a sufficient condition for stability, and $\lambda \leq \mu_{av}$ is necessary.

2.2.3 Channel Convergence

A wireless data network may have many data links, and the channel conditions of each link could vary from slot to slot. It is useful to develop a notion of *channel convergence*, which is similar to the notion of rate convergence for arrival and server processes. Specifically, suppose there are N links, and let $\vec{S}(t) = (S_1(t), \ldots, S_N(t))$ represent the vector process of link conditions as a function of time. Each component $S_i(t)$ takes values on some finite state space C_i (representing a set of link conditions for channel i), so that the channel process $\vec{S}(t)$ takes values in the finite state space $C_1 \times C_2 \times \cdots \times C_N$. For all states $\vec{S}$ and for any initial time t_0, let $T_{\vec{S}}(t_0, K)$ represent the set of timeslots during the interval $[t_0, t_0 + K - 1]$ at which the system is in state $\vec{S}$. Let $||T_{\vec{S}}(t_0, K)||$ represent the total number of these

[3] Hence, if $\lambda > \mu_{av}$, for any given value V we cannot find arbitrarily large times $\tilde{t}$ such that $Pr[U(\tilde{t}) \leq V] \geq 1/2$.

timeslots.

Definition 4. *A channel process $\vec{S}(t)$ is* channel convergent *with steady state probabilities $\pi_{\vec{S}}$ if:*

(i) For all states $\vec{S}$, $\frac{||T_{\vec{S}}(0,K)||}{K} \rightarrow \pi_{\vec{S}}$ with probability 1 as $K \rightarrow \infty$

(ii) For any $\delta > 0$, there exists an interval size K such that for all channel states $\vec{S}$, all initial times t_0, and regardless of past history, the following condition holds:

$$\sum_{\underline{S}} \left| \frac{\mathbb{E}\left\{||T_{\underline{S}}(t_0, K)||\right\}}{K} - \pi_{\underline{S}} \right| \leq \delta \tag{2.6}$$

Thus, for channel convergent processes, the time average fraction of time in each channel state converges to the steady state distribution $\pi_{\vec{S}}$, and the expected time average is arbitrarily close to this distribution if sampled over a suitably large interval. This notion of channel convergence is important for systems with server rates that depend on channel conditions. Indeed, consider a channel convergent process $\vec{S}(t)$ that determines the expected service rate $\mathbb{E}\left\{\mu(t)\right\}$ for a particular link, so that independently of past history, we have $\mathbb{E}\left\{\mu(t) \mid \vec{S}(t) = \vec{S}\right\} = R_{\vec{S}}$ (for some given set of rates $\{R_{\vec{S}}\}$). We have the following simple lemma.

Lemma 4. *The process $\mu(t)$ defined over the channel convergent process $\vec{S}(t)$ as described above is rate convergent with average rate $\mu_{av} \triangleq \sum_{\vec{S}} \pi_{\vec{S}} R_{\vec{S}}$. Furthermore*

$$\left| \mu_{av} - \mathbb{E}\left\{ \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \mu(\tau) \right\} \right| \leq R_{max}\delta \tag{2.7}$$

where R_{max} is the maximum value of $R_{\vec{S}}$ over all channel states $\vec{S}$, and K and δ are the parameters of the $\vec{S}(t)$ process described in (2.6).

Proof. The difference between μ_{av} and the empirical rate over K slots is given by:

$$\begin{aligned} \mu_{av} - \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mu(\tau) &= \sum_{\vec{S}} \pi_{\underline{S}} R_{\vec{S}} - \frac{1}{K} \sum_{\vec{S}} \sum_{\tau \in T_{\vec{S}}(t_0,K)} \mu(\tau) \qquad (2.8) \\ &= \sum_{\vec{S}} \pi_{\vec{S}} R_{\vec{S}} - \sum_{\vec{S}} \frac{||T_{\vec{S}}(t_0,K)||}{K} \frac{1}{||T_{\vec{S}}(t_0,K)||} \sum_{\tau \in T_{\vec{S}}(t_0,K)} \mu(\tau) \end{aligned}$$

Note that defining $t_0 = 0$ and using the fact that $||T_{\vec{S}}(0,K)||/K \rightarrow \pi_{\vec{S}}$ and the fact that (by the law of large numbers), $\frac{1}{||T_{\vec{S}}(0,K)||}\sum_{\tau\in T_{\vec{S}}(0,K)}\mu(\tau) \rightarrow R_{\vec{S}}$ as $K \rightarrow \infty$ reveals that $\frac{1}{K}\sum_{\tau=0}^{K-1}\mu(\tau) \rightarrow \mu_{av}$, proving the first condition of rate convergence.

Now fix $\delta > 0$ and let K be large enough so that for any initial timeslot t_0, we have $\sum_{\vec{S}}\left|\frac{\mathbb{E}\{||T_{\vec{S}}(t_0,K)||\}}{K} - \pi_{\vec{S}}\right| \leq \delta$. Taking expectations of inequality (2.8) yields

$$\begin{aligned}
\mu_{av} - \mathbb{E}\left\{\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mu(\tau)\right\} &= \sum_{\vec{S}}\pi_{\vec{S}}R_{\vec{S}} - \mathbb{E}\left\{\frac{1}{K}\sum_{\vec{S}}\sum_{\tau\in T_{\vec{S}}(t_0,K)}\mathbb{E}\left\{\mu(\tau)\mid\vec{S}\right\}\right\} \\
&= \sum_{\vec{S}}\pi_{\underline{S}}R_{\vec{S}} - \sum_{\vec{S}}\frac{\mathbb{E}\left\{||T_{\vec{S}}(t_0,K)||\right\}}{K}R_{\vec{S}} \\
&\leq \sum_{\vec{S}}R_{\vec{S}}\left|\pi_{\vec{S}} - \frac{\mathbb{E}\left\{||T_{\vec{S}}(t_0,K)||\right\}}{K}\right|
\end{aligned}$$

and hence $\left|\mu_{av} - \mathbb{E}\left\{\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mu(\tau)\right\}\right| \leq R_{max}\delta$. The expectation of the time average rate can thus be made arbitrarily close to μ_{av}, proving the result. □

Note that in the special case of i.i.d. channel states, the parameters in (2.6) can be set to $K = 1, \delta = 0$, as steady state averages are achieved every timeslot.

2.3 Miscellaneous Queueing Theoretic Results

During the course of this thesis we have developed a number of interesting queueing theoretic results applicable to general queues with time varying server rates. The following results are not used in this thesis but are presented in Appendices A, B, and C.

2.3.1 Convexity in Queues

In Appendix A we prove that any moment of unfinished work in a queue is monotonically increasing and convex in the input rate λ. This holds for general arrival streams, where the data rate is described either in discrete steps corresponding to a finite set of indistinguishable substreams being added or removed from the arrival process to the queue, or as a continuous variable obtained by probabilistically splitting the traffic from an arbitrary stream. The result is intuitive and the analysis is simple and elegant, using a novel form of stochastic coupling. This result establishes an important foundation, as convexity is often assumed when applying optimization techniques to finding optimal flow distributions in queues. Such convexity results can be extended to any system satisfying the *non-negativity*, *symmetry*, and *monotonicity* conditions. This work was presented in [103].

2.3.2 Join-the-Shortest-Queue Routing in Finite Buffer Systems

In Appendix B we consider a problem of routing packets from an arbitrary input stream over a set of N parallel queues with heterogeneous and arbitrarily varying server rates $\mu_1(t), \ldots, \mu_N(t)$. We define $d_\pi(B)$ as the drop rate when each queue has a finite buffer size B when some routing algorithm π is used, and say that the system is stable if $d_\pi(B) \rightarrow 0$ as $B \rightarrow \infty$. Considering the *Join-the-Shortest-Queue (JSQ)* policy in comparison to any other policy (perhaps one with full knowledge of future events), it is shown that:

$$d_{JSQ}(B) \leq d_\pi \left(\frac{B}{N} - L_{max} - \frac{L_{max}}{N} \right)$$

where L_{max} is the maximum length of a packet.

Hence, the *JSQ* strategy yields stability whenever possible, and has a loss rate which is lower than the loss rate of any other policy implemented on queues with a suitably smaller buffer size. Upper and lower bounds on the loss rate can be computed in terms of a single queue with an aggregate server rate $\mu_{sum}(t)$ equal to the sum of the individual rates $\mu_1(t) + \ldots + \mu_N(t)$. This work was presented in [110].

2.3.3 The Jitter Theorem

In Appendix C we prove that any moment of unfinished work in a queue with an arbitrary arrival process and an independent and stationary time varying server process $\mu(t)$ is greater than or equal to the corresponding moment in a system with a constant server rate $\overline{\mu}$ equal to the time average of $\mu(t)$. Two different and simple proofs are given, and a simple upper bound is conjectured.

Chapter Appendix 2.A — Extension of Foster's Criterion

Here we present additional conditions which, together with the K-slot Lyapunov drift condition of Lemma 2, imply that time averages of unfinished work $U_i(t)$ in each node i converge to some finite value $\overline{U}_i$. This involves extending a well known Lyapunov Drift result called *Foster's Criterion* [6] (similar to the Pakes drift lemma given in [14]) to address first moments of backlog in general stochastic systems with uncountably infinite state spaces, as the unfinished work vector $\vec{U}(t)$ has uncountably infinite cardinality in each of the N dimensions.

Lemma 5. *(Extension of Foster's Criterion) Consider an unfinished work process $\vec{U}(t) \in \mathbb{R}^N$ which satisfies the following four conditions:*

(i) The $\vec{U}(t)$ stochastics evolve according to a finite state Markov Chain $M(t)$, so that channel and arrival distributions at slot t are determined by the state of $M(t)$.

We define $F(t) = [\vec{U}(t), M(t)]$ as the combined system state consisting of the current backlog vector and the current state of the Markov chain.

(ii) There exists a positive integer K and values $B < \infty$ and $\epsilon > 0$ such that for all timeslots t_0 and regardless of the initial state $F(t_0)$, a Lyapunov function evaluated K steps into the future satisfies:

$$\mathbb{E}\left\{L(\vec{U}(t_0+K)) - L(\vec{U}(t_0)) \mid F(t_0)\right\} \leq B - \epsilon \sum_i U_i(t_0)$$

(iii) $\mathbb{E}\left\{L(\vec{U}(\tau))\right\} < \infty$ for all $\tau \in \{0, 1, \ldots, K-1\}$.

We define the following compact set Ω_δ parameterized by a value $\delta > 0$:

$$\Omega_\delta \triangleq \left\{\vec{x} \in \mathbb{R}^N \mid \vec{x} \geq 0, \sum_i x_i \leq \frac{B+\delta}{\epsilon}\right\}$$

(iv) There exists a $\delta > 0$ such that for any time t_0 and for any initial backlog $\vec{U}(t_0) \in \Omega_\delta$, there is some positive probability p that the backlog in all queues will simultaneously empty within a finite number of timesteps.

If conditions (i)-(iv) are satisfied, then there are finite values $\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_N$ such that for all i, $\frac{1}{t}\sum_{\tau=0}^{t-1} U_i(\tau) \to \overline{U}_i$ as $t \to \infty$, and these steady state values satisfy $\sum_i \overline{U}_i \leq B/\epsilon$.

Before proving the lemma, we note its applicability to time varying wireless networks and to Lemma 2. Note that conditions (ii) and (iii) are similar to the Lyapunov drift conditions of Lemma 2, and follow from these conditions by defining $\epsilon \triangleq \theta_{min}$. Condition (i) indicates that we are considering only systems with arrival and channel processes modulated by a finite state Markov chain. Finally, condition (iv) holds in cases where every timeslot there is some nonzero probability that no new arrivals enter the network, so that the system will empty after a suitably large number of such successive "no arrival" slots. Lemma 2 together with the lemma above thus imply time averages $\overline{U}_i$ exist and satisfy $\sum_i \theta_i \overline{U}_i \leq B$.

Proof. Suppose conditions $(i)-(iv)$ hold. Condition (ii) combined with the definition of the compact set Ω_δ implies that the K-step Lyapunov drift is less than or equal to $-\delta$ whenever the initial backlog $\vec{U}(t_0)$ is outside of the Ω_δ region. Define $T(\vec{U}, M)$ as the random number of timeslots required to return to the Ω_δ region given that the unfinished work vector leaves Ω_δ starting at a point $\vec{U} \in \Omega_\delta$ when the Markov chain is in state M. By the standard drift theory for Foster's Criterion [6] [14], this time has a finite mean $\overline{T}(\vec{U}, M)$.[4] From this fact together with the fact that any finite function over a compact set has a maximum value [100], it follows that the maximum of $\overline{T}(\vec{U}, M)$ over all M and all $\vec{U} \in \Omega_\delta$ exists and is also finite (as the number of Markov states $||M||$ is finite and the set Ω_δ is compact). Define $\overline{T}$ as this maximum value, representing the maximum mean recurrence time to the Ω_δ region starting at any point within the region.

Let Z represent the number of timeslots required for the system to have a positive probability p of emptying within Z slots, starting from any $\vec{U} \in \Omega_\delta$ and starting in any Markov state. Such a value exists by condition (iv). Consider starting at any $\vec{U} \in \Omega_\delta$ and waiting until either Z slots expire and the unfinished work has not left Ω_δ, or until the unfinished work exits the Ω_δ region and returns again. The average time to wait for such a duration is no more than $Z + \overline{T}$, and the system empties with probability p during each such interval. It follows that the average time to empty starting at any point in the Ω_δ region is no more than $(Z + \overline{T})/p$, independent of past history.

Define $T_0(M \to M)$ as the time required to return to an empty network when the Markov chain is in state M, starting from an empty network in the same Markov state. (That is, $T_0(M \to M)$ is the mean recurrence time to state $[\vec{0}, M]$.) Further define $N_M(t)$ to be the number of times the system is empty while the Markov chain is in state M during

[4] The proof of this fact is similar to (2.12)-(2.14) given in the proof of the claim below.

the interval $0 \leq \tau \leq t$. By renewal theory [49], we have:

$$\lim_{t\to\infty} \frac{N_M(t)}{t} = \frac{1}{\mathbb{E}\left\{T_0(M \rightarrow M)\right\}}$$

so that the limit above always converges, and $\mathbb{E}\left\{T_0(M \rightarrow M)\right\}$ is finite whenever the limit is strictly positive. Now define $N(t) = \sum_M N_M(t)$ as the total number of times the system empties during $0 \leq \tau \leq t$. Because there are a finite number of Markov states M, we can pass the limit through the summation to find:

$$\lim_{t\to\infty} \frac{N(t)}{t} = \sum_M \lim_{t\to\infty} \frac{N_M(t)}{t} = \sum_M \frac{1}{\mathbb{E}\left\{T_0(M \rightarrow M)\right\}} \tag{2.9}$$

However, because the expected duration between emptying times is independently bounded by $(Z + \overline{T})/p$, it follows that:

$$\lim_{t\to\infty} \frac{N(t)}{t} \geq \frac{p}{Z + \overline{T}} \tag{2.10}$$

Combining (2.9) and (2.10), we find that:

$$\sum_M \frac{1}{\mathbb{E}\left\{T_0(M \rightarrow M)\right\}} \geq \frac{p}{Z + \overline{T}} > 0$$

Thus, there is some state M_1 such that $\mathbb{E}\left\{T_0(M_1 \rightarrow M_1)\right\} < \infty$.

We mark the times when the network empties and the Markov chain is in state M_1 as *renewal times*, and note that the system has independent and identical stochastics after renewals. By renewal theory [49], it follows that for each i, we have:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} U_i(\tau) = \frac{\mathbb{E}\left\{A^{(i)}_{M_1 \rightarrow M_1}\right\}}{\mathbb{E}\left\{T_0(M_1 \rightarrow M_1)\right\}}$$

where $\mathbb{E}\left\{A^{(i)}_{M_1 \rightarrow M_1}\right\}$ is defined as the expected sum of unfinished work in queue i between renewal times. If this value is finite for queue i, then the time average unfinished work in this queue is finite and converges to the value $\overline{U}_i \triangleq \frac{\mathbb{E}\left\{A^{(i)}_{M_1 \rightarrow M_1}\right\}}{\mathbb{E}\{T_0(M_1 \rightarrow M_1)\}}$.

Claim: $\sum_i \mathbb{E}\left\{A^{(i)}_{M_1 \rightarrow M_1}\right\} \leq \frac{B\mathbb{E}\{T_0(M_1 \rightarrow M_1)\}}{\epsilon} + \frac{\sum_{k=0}^{K-1} \mathbb{E}\left\{L(\vec{U}(r_0+k))\right\}}{\epsilon}$, where r_0 is a renewal time. In particular, from condition (*iii*) we have that the value $\mathbb{E}\left\{A^{(i)}_{M_1 \rightarrow M_1}\right\}$ is finite for each i.

The claim is proved separately below. From the claim, it follows that time averages of unfinished work converge, so that $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} U_i(\tau) = \overline{U}_i$, where the $\overline{U}_i$ values satisfy:

$$\sum_i \overline{U}_i \leq \frac{B}{\epsilon} + \frac{\sum_{k=0}^{K-1} \mathbb{E}\left\{L(\vec{U}(r_0+k))\right\}}{\epsilon \mathbb{E}\left\{T_0(M_1 \to M_1)\right\}}$$

Note that the above inequality holds for any definition of a renewal interval, provided that renewals begin when the network is empty and the Markov chain is in state M_1 (so that $F(t) = [\vec{0}, M_1]$). Hence, the above analysis can be repeated in the case when renewals are defined on every R^{th} visitation to the state $F(t) = [\vec{0}, M_1]$. Nothing changes except for the denominator of the error term, and we have:

$$\sum_i \overline{U}_i \leq \frac{B}{\epsilon} + \frac{\sum_{k=0}^{K-1} \mathbb{E}\left\{L(\vec{U}(r_0+k))\right\}}{\epsilon R \mathbb{E}\left\{T_0(M_1 \to M_1)\right\}}$$

The above inequality holds for any positive integer R. Taking limits as $R \to \infty$ shows that $\sum_i \overline{U}_i \leq \frac{B}{\epsilon}$, proving the result. □

Proof of Claim: $\sum_i \mathbb{E}\left\{A^{(i)}_{M_1 \to M_1}\right\} \leq \frac{B\mathbb{E}\{T_0(M_1\to M_1)\}}{\epsilon} + \frac{\sum_{k=0}^{K-1}\mathbb{E}\left\{L(\vec{U}(r_0+k))\right\}}{\epsilon}$, where r_0 is a renewal time.

Proof. Consider the system starting out with initial state $F(0) = [\vec{0}, M_1]$, so that time 0 is a renewal time. Recall that $T_0(M_1 \to M_1)$ represents the number of timesteps required to return to this renewal state. For simplicity of notation, we represent this time as T, and note from the above result that $\mathbb{E}\{T\} < \infty$. Recall that $A^{(i)}_{M_1 \to M_1}$ represents the sum of unfinished work for the duration $\{0, 1, \ldots, T-1\}$ between renewal events. We have:

$$\begin{aligned} \mathbb{E}\left\{A^{(i)}_{M_1 \to M_1}\right\} &= \mathbb{E}\left\{\sum_{\tau=0}^{T-1} U_i(\tau)\right\} \\ &= \mathbb{E}\left\{\sum_{\tau=0}^{\infty} U_i(\tau) 1_{[T>\tau]}\right\} \end{aligned} \tag{2.11}$$

Define $H(t)$ as the complete history of the $F(\tau)$ system state for $\tau \in \{0, 1, \ldots, t\}$. Define $Y(t) \triangleq L(\vec{U}(t))1_{[T>t]}$. Imitating the proof for the fact that $\mathbb{E}\{T\} < \infty$ given in [6] [14], we have for any time t:

$$\begin{aligned}
\mathbb{E}\left\{Y(t+K) \mid H(t)\right\} &= \mathbb{E}\left\{L(\vec{U}(t+K))1_{[T>t+K]} \mid H(t)\right\} \\
&\leq \mathbb{E}\left\{L(\vec{U}(t+K))1_{[T>t]} \mid H(t)\right\} &(2.12)\\
&\leq \mathbb{E}\left\{\left[L(\vec{U}(t)) + B - \epsilon\sum_i U_i(t)\right]1_{[T>t]} \mid H(t)\right\} &(2.13)\\
&= Y(t) + B1_{[T>t]} - \epsilon\sum_i U_i(t)1_{[T>t]} &(2.14)
\end{aligned}$$

where (2.12) follows because $1_{[T>t+K]} \leq 1_{[T>t]}$ for all t, and (2.13) follows by condition (ii) of the above lemma together with the fact that $\vec{U}(t)$ and $1_{[T>t]}$ are fixed quantities given $H(t)$. Taking expectations of (2.14) over the distribution of $H(t)$, we thus have:

$$\mathbb{E}\left\{Y(t+K)\right\} - \mathbb{E}\left\{Y(t)\right\} \leq BPr[T>t] - \epsilon\sum_i \mathbb{E}\left\{U_i(t)1_{[T>t]}\right\}$$

Fix $t_0 \in \{0, \ldots, K-1\}$ and apply the above equations at times $t = t_0 + mK$. Summing over m from 0 to $M-1$ yields:

$$\mathbb{E}\left\{Y(t_0+MK)\right\} - \mathbb{E}\left\{Y(t_0)\right\} \leq B\sum_{m=0}^{M-1} Pr[T > t_0+mK] - \epsilon\sum_i\sum_{m=0}^{M-1}\mathbb{E}\left\{U_i(t_0+mK)1_{[T>t_0+mK]}\right\}$$

Summing over t_0 from $t_0 = 0$ to $t_0 = K-1$ yields:

$$\sum_{t_0=0}^{K-1}\mathbb{E}\left\{Y(MK+t_0)\right\} - \sum_{t_0=0}^{K-1}\mathbb{E}\left\{Y(t_0)\right\} \leq B\sum_{\tau=0}^{MK-1} Pr[T>\tau] - \epsilon\sum_i\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{U_i(\tau)1_{[T>\tau]}\right\}$$

Taking limits as $M \to \infty$ and using the fact that $0 \leq \mathbb{E}\left\{Y(t)\right\} \leq \mathbb{E}\left\{L(\vec{U}(t))\right\}$, we have:

$$\sum_i\sum_{\tau=0}^{\infty}\mathbb{E}\left\{U_i(\tau)1_{[T>\tau]}\right\} \leq \frac{B\sum_{\tau=0}^{\infty}Pr[T>\tau]}{\epsilon} + \frac{1}{\epsilon}\sum_{t_0=0}^{K-1}\mathbb{E}\left\{L(\vec{U}(t_0))\right\}$$

Using (2.11) and noting that $\mathbb{E}\left\{T\right\} = \sum_{\tau=0}^{\infty} Pr[T>\tau]$, we have:

$$\sum_i \mathbb{E}\left\{A_{M_1\to M_1}^{(i)}\right\} \leq \frac{B\mathbb{E}\left\{T\right\}}{\epsilon} + \frac{1}{\epsilon}\sum_{t_0=0}^{K-1}\mathbb{E}\left\{L(\vec{U}(t_0))\right\}$$

proving the claim. □

Hence, the conditions of Lemma 5 imply that:

$$\frac{1}{t}\sum_{\tau=0}^{t-1} U_i(\tau) \rightarrow \overline{U}_i \triangleq \frac{\mathbb{E}\left\{A_{M_1\rightarrow M_1}^{(i)}\right\}}{\mathbb{E}\left\{T_0(M_1\rightarrow M_1)\right\}}$$

For completeness, we show that $\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U_i(\tau)\right\}$ converges to the same limit.

Proof. Let $R(t)$ represent the number of complete renewal intervals that occur by time t. We have:

$$\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U_i(\tau)\right\} = \mathbb{E}\left\{A_0\right\} + \mathbb{E}\left\{\sum_{r=1}^{R(t)} A_r\right\} + \mathbb{E}\left\{\tilde{A}_{R(t)+1}\right\} \tag{2.15}$$

where A_0 represents the sum of unfinished work in queue i from time 0 up to the time just before the first renewal event, A_r represents the sum of unfinished work in queue i over renewal interval r, and $\tilde{A}_{R(t)+1}$ represents the portion of $A_{R(t)+1}$ summed up to the cutoff time t. Note that for all integers $r \geq 1$, we have:

$$\mathbb{E}\left\{A_r\right\} = \mathbb{E}\left\{A_{M_1\rightarrow M_1}^{(i)}\right\}$$

The process $R(t)$ is a *stopping time process* [49], and hence by Wald's equality, we have:

$$\mathbb{E}\left\{\sum_{\tau=0}^{R(t)} A_r\right\} = \mathbb{E}\left\{A_1\right\}\mathbb{E}\left\{R(t)\right\}$$

Using this identity in (2.15) yields:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U_i(\tau)\right\} = \frac{\mathbb{E}\left\{A_0\right\}}{t} + \mathbb{E}\left\{A_1\right\}\frac{\mathbb{E}\left\{R(t)\right\}}{t} + \frac{\mathbb{E}\left\{\tilde{A}_{R(t)+1}\right\}}{t}$$

Note that $\mathbb{E}\left\{A_0\right\}$ is finite, and that $\mathbb{E}\left\{\tilde{A}_{R(t)+1}\right\} \leq \mathbb{E}\left\{A_{R(t)+1}\right\} = \mathbb{E}\left\{A_1\right\}$. Hence, the first and third terms on the right hand side of the above equation vanish as $t \rightarrow \infty$. It follows that

$$\lim_{t\rightarrow\infty}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U_i(\tau)\right\} = \mathbb{E}\left\{A_1\right\}\lim_{t\rightarrow\infty}\frac{\mathbb{E}\left\{R(t)\right\}}{t}$$

where the limit of $\frac{\mathbb{E}\{R(t)\}}{t}$ converges to $1/\mathbb{E}\left\{T_0(M_1\rightarrow M_1)\right\}$ by renewal theory [49]. □

Chapter 3

Satellite and Wireless Downlinks

Consider the example network of two downlink nodes shown in Fig. 3-1. Data streams X_1, X_2, and X_3 arrive to the system and are intended for destinations $1, 2$, and 3, respectively. Packets from the first input stream arrive to Node A, while packets from the second input stream can be routed to either Node A or Node B. Packets from the third input stream arrive to Node B but can be delivered either directly to destination 3 or indirectly through destination 4. Furthermore, the downlink data rates depend on time varying channel conditions as well as power allocation decisions. We thus have the following joint problem of routing and power allocation:

Routing: In which station do we put packets from Source 2? From the figure, it seems that the most direct path for these packets is through Node A, as Node A is closer to destination 2 than Node B. However, if traffic from stream X_1 is heavy, it may be better to route most packets from Source 2 through Node B, which allows Node A to devote most of its power resources to the Source 1 traffic. How can these decisions be made dynamically without prior knowledge of the arrival patterns of other users?

Another routing issue involves selecting a downlink path for the X_3 data in node B. If destinations 3 and 4 are connected and can forward data to each other, it may be useful to send some of the data destined for node 3 over the $(B, 4)$ downlink, rather than the $(B, 3)$ downlink.

Power Allocation: Each satellite is power constrained, so that downlink allocations must

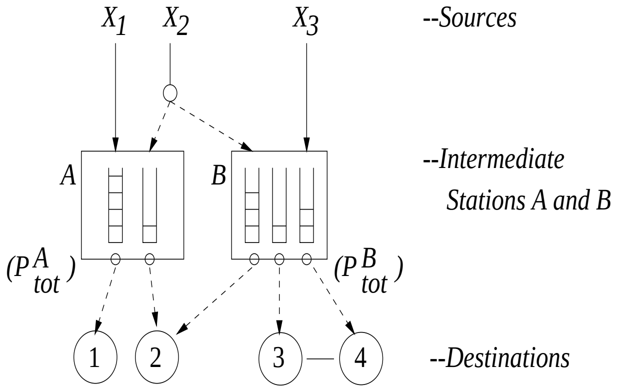


Figure 3-1: An example problem of routing and power allocation for satellite or wireless downlinks.

satisfy the following constraints for all time:

$$\begin{aligned} P_{A1}(t) + P_{A2}(t) &\leq P_{tot}^{A} \\ P_{B2}(t) + P_{B3}(t) + P_{B4}(t) &\leq P_{tot}^{B} \end{aligned}$$

How must power be allocated, and how must this be done dynamically as a function of the queue backlog and the current channel conditions?

A further problem of *packet scheduling* does not arise in this example, but would arise if destination 3 where able to forward packets to destination 2. In this case, data from either source X_2 or source X_3 might be delivered over the downlink $(B, 3)$. Thus, given a power allocation for this link, one must still decide which data to transmit.

In this chapter, we introduce the downlink problem and develop dynamic network control algorithms for the specific application of a multi-beam satellite. The analysis provided here is also applicable to wireless systems. We begin by examining the power allocation problem. The joint problem of routing and power allocation is treated in Section 3.5, and the full problem of routing, power allocation, and scheduling is treated in Chapter 4 where the general network problem is considered. We have published much of the work contained in this chapter in [108] [107].

3.1 The Multi-Beam Satellite Downlink

Consider the multi-beam satellite downlink system of Fig. 3-2, where a single satellite transmits data to N ground locations over N different downlink channels. Each channel is assumed to be time varying (e.g., due to changing weather conditions or satellite motion), and the overall channel state is described by the channel convergent vector process $\vec{S}(t) = (S_1(t), \ldots, S_N(t))$. Packets destined for ground location i arrive from an input stream X_i and are placed in an output queue to await processing. The servers of each of the N output queues may be activated simultaneously at any time t by assigning to each a power level $P_i(t)$, subject to the total power constraint $\sum_i P_i(t) \leq P_{tot}$. The transmission rate of each server i depends on the allocated power $p_i(t)$ and on the current channel state $S_i(t)$ according to a concave rate-power curve $\mu_i(P_i, S_i)$. A controller allocates power to each of the N queues at every instant of time in reaction to channel state and queue backlog information. The goal of the controller is to stabilize the system and thereby achieve maximum throughput and maintain acceptably low levels of unfinished work in all of the queues.

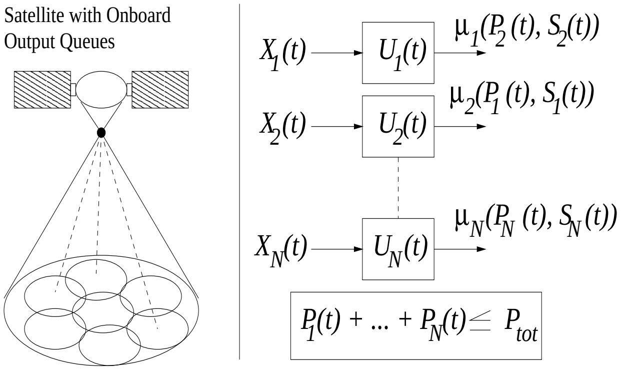


Figure 3-2: A multibeam satellite with N time-varying downlink channels and N onboard output queues.

Related work on queue control strategies and resource allocation for satellite and wireless systems is found in [108] [136] [138] [68] [86] [23] [18] [133] [131] [132] [60] [142] [5] [99] [143] [124] [141] [47] [111] [110] [134]. In [136], the capacity of a wireless uplink is established using optimal information theoretic methods, and a related downlink problem is treated in [138], leading to the *Serve-the-Best-Channel* policy for maximizing the sum output rate. This work assumes that users have an infinite backlog of data which can be delivered upon request. A similar problem is treated in [86] for finding a static power allocation in a wireless downlink to maximize a sum of user utility functions. In [133], dynamic scheduling in a parallel queue system with a single server is examined, where every timeslot the transmit channels of the queues vary between ON and OFF states and the server selects a queue to service from those that are ON. The capacity region of the system is developed when packet arrivals and channel states are i.i.d. Bernoulli processes, and stochastic coupling is used to show optimality of the $Serve-the-Longest-Connected-Queue$ policy in the symmetric situation that arrival and channel processes are identical for all queues (i.e., $\lambda_1 = \ldots = \lambda_N, p_1^{on} = \ldots = p_N^{on}$). Similar server scheduling problems for wireless downlinks are treated in [5] [124]. Such server allocation problems can be viewed as special cases of our power allocation formulation, and in Section 3.4 we verify stability of the $Serve-the-Longest-Connected-Queue$ policy for symmetric and asymmetric systems with multiple servers, as well as provide a delay bound. We note that a rate allocation scheme similar to our power allocation algorithm was independently proposed in [143] for stabilizing a wireless downlink in a cellular context.

In [132], a wireless network of queues is analyzed when input packets arrive according to Poisson processes and have exponentially distributed length. A Lyapunov function is used to establish a stabilizing routing and scheduling policy under network connectivity constraints. In [131], Lyapunov analysis is used to develop a server allocation algorithm in a network with time-varying connectivity. Such a technique has been recently used for establishing stability in an uplink with static channels in [149], [84], in a one-hop static network in [71], and in the switching literature [97] [95] [75] [88] [109]. In [95], an $N \times N$ packet switch is treated and input/output matching strategies are developed to ensure 100% throughput whenever the arrival rates are within the feasible region. In [88], [81] [98], the method of Lyapunov stability analysis is used to prove that queues are not only stable but have finite backlog moments. In particular, a switch with Poisson inputs and operating under the Maximum Weight Matching (MWM) scheduling algorithm is shown in [88] to have delay that is upper bounded by a function which grows linearly in the number of input ports to the switch.[1] A similar delay analysis was independently developed in [71] for scheduling in a one-hop network with memoryless inputs.

The main contribution in this chapter is the treatment of a general power allocation problem for multi-beam satellites with adaptive transmission rates, and the development of throughput maximizing power and server allocation algorithms for systems with general inputs and time varying channels. This is accomplished through the Lyapunov theory established in Chapter 2. This analysis extends to other wireless networking problems

[1] A logarithmic delay algorithm for switch scheduling is developed in [105].

where power allocation and energy efficiency is a major issue. Recent work in [141] treats a problem of minimizing the total energy expended to transmit blocks of data arriving to a single queue, and it is shown that power control can be effectively used to extend longevity of network elements. In [134] power allocation for wireless networks is addressed. The authors consider ON/OFF type power allocation policies and observe that for random networks, capacity regions are not extended much by including more power quantization levels. Our capacity results in this chapter illustrate that the capacity region is often considerably extended if multiple power levels are utilized for the satellite downlink problem.

3.2 Power and Server Allocation

Consider the N queue system of Fig. 3-2. Each time varying channel i can be in one of a finite set of states C_i. We represent the channel process by the channel state vector $\vec{S}(t) = (S_1(t), \cdots S_N(t))$, where $\vec{S}(t) \in C_1 \times \cdots \times C_N$. Time is slotted, and channels hold their states for the duration of a timeslot (where the slot size is normalized to 1 unit of time). It is assumed that the channel states are known at the beginning of each timeslot. The channel process is assumed to be channel convergent with channel probabilities $\pi_{\vec{S}}$ for each state $\vec{S}$. At every timeslot, the server transmission rates can be controlled by adjusting the power allocation vector $\vec{P}(t) = (P_1(t), \ldots, P_N(t))$ subject to the total power constraint $\sum_i P_i(t) \leq P_{tot}$. For any given state S_i of downlink channel i, there is a corresponding rate-power curve $\mu_i(P_i, S_i)$ which is increasing, concave, and continuous in the power parameter (Fig. 3-3). Note that these curves implicitly assume that channels are independent, as the rate over channel i depends only on the power allocated to that channel and not on the power allocated elsewhere. This model is realistic for satellite downlinks in which individual beams do not interfere with each other, and leads to simple real-time control algorithms. We note that general rate-power functions $\underline{\mu}(\underline{P}, \underline{S})$ with arbitrary curvature, interchannel interference, and potential discontinuities are treated in Chapter 4.

The power curve $\mu_i(P_i, S_i)$ could represent the logarithmic Shannon capacity curve of a Gaussian channel, or could represent a rate curve for a specific set of coding schemes designed to achieve a sufficiently low probability of error in the given channel state. In general, any practical set of power curves will have the concavity property, reflecting diminishing returns in transmission rate with each incremental increase in signal power. The

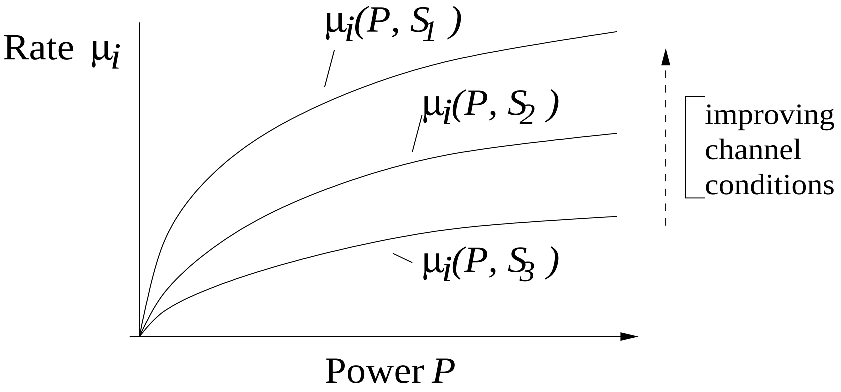


Figure 3-3: A set of concave power curves $\mu_i(P_i, S_i)$ for channel states S_1, S_2, S_3.

continuity property is less practical. A real system will rely on a finite databank of coding schemes, and hence actual rate/power curves restrict operation to a finite set of points. For such a system, we can create a new, virtual power curve by a piecewise linear interpolation of the operating points (see Fig. 3-4a). Such virtual curves have the desired continuity and concavity properties, and are used as the true curves in our power allocation algorithms. Clearly a virtual system which allocates power according to the virtual curves has a capacity region which contains that of a system restricted to allocate power on the vertex points. However, when vertex points are equally spaced along the power axis and integrally divide the total power P_{tot}, the capacity regions are in fact the same, as any point on a virtual curve can effectively be achieved by time-averaging two or more feasible rate-power points over many timeslots. Indeed, in Section 3.4 we design a stabilizing policy for any set of concave power curves which naturally selects vertex points at every timeslot if power curves are piecewise linear.

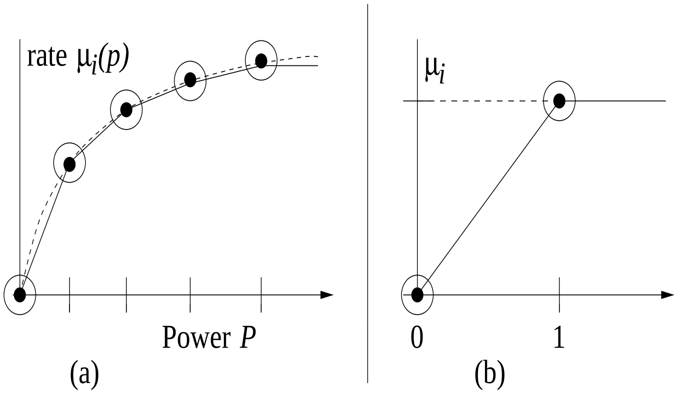


Figure 3-4: Virtual power curves for systems with a finite set of operating points.

This power allocation formulation generalizes a simpler problem of server allocation. Assume that there are K servers, and every timeslot the servers are scheduled to serve K of the N queues ($K < N$). A given queue i transmits data at a fixed rate μ_i whenever a server is allocated to it, and transmits nothing when no server is allocated. This problem can be transformed into a power allocation problem by defining the *virtual power constraint* $\sum_i P_i(t) \leq K$ and the virtual power curves:

$$\tilde{\mu}_i(P) = \begin{cases} \mu_i P & \text{if } 0 \leq P \leq 1 \\ \mu_i & \text{if } P > 1 \end{cases} \tag{3.1}$$

Such a virtual curve contains the feasible points ($P = 0, \tilde{\mu}_i = 0$) and ($P = 1, \tilde{\mu}_i = \mu_i$), corresponding to a server being either allocated or not allocated to queue i (see Fig. 3-4b). However, it suffices to remove this feasible point restriction and treat the system as if it operates according to the continuous virtual power curve (3.1). This preserves the same capacity region, and later it is shown that any stabilizing algorithm which uses the virtual curves can be transformed into a stabilizing algorithm which conforms to the feasible point restriction.

3.2.1 Example Server Allocation Algorithm

One might suspect the policy of serving the K fastest, non-empty queues would maximize data output and achieve stability. However, we provide the following counterexample which illustrates this is not the case. Consider a 3-queue, 2-server system with constant processing rates $(\mu_1, \mu_2, \mu_3) = (1, 1, 1/2)$. All arriving packets have length $L = 1$ and arrive according i.i.d. Bernoulli processes with packet arrival probabilities $(p_1, p_2, p_3) = (p, p, (1-p^2)/2 + \epsilon)$, where $p < 1/2$ and $0 < \epsilon < p^2/2$. Note that the policy of serving the two fastest non-empty queues removes a server from queue 3 whenever there are simultaneous arrivals at queues 1 and 2. This happens with probability p^2, and hence the time average processing rate at queue 3 is no more than $(1 - p^2)/2$ (where the factor 1/2 is due to the rate of server 3). This effective service rate cannot support the input rate, and hence queue 3 is unstable under this server allocation policy. However, the system is clearly stabilizable: The policy of always allocating a server to queue 3 and using the remaining server to process packets in queues 1 and 2 stabilizes all queues.

3.3 The Downlink Capacity Region

Let the arrival process $X_i(t)$ represent the total bits that arrive to queue i during the first t slots, and assume this process is rate convergent with rate λ. It is assumed that all processes have bounded second moments, so that the expected number of total arrivals $\sum_i A_i(t)$ during any slot t satisfies $\mathbb{E}\left\{(\sum_i A_i(t))^2\right\} \leq A_{max}^2$ for all i and for all timeslots t, regardless of past history. Recall that $\vec{S}(t)$ represents the time varying server process, and is channel convergent with state probabilities $\pi_{\vec{S}}$. Let the unfinished work vector $\vec{U}(t)$ represent the state of queue backlog at slot t.

Definition 5. *The* Downlink Capacity Region Λ *is the closure of the set of all rate vectors* $\vec{\lambda}$ *that can be stabilized by some power allocation strategy.*

Establishing that a set Λ represents the capacity region of the downlink requires proving that any inputs with rate vector λ strictly interior to Λ can be stabilized, as well as proving that stability is impossible for any rates outside of Λ. This is accomplished in the following theorem. The theorem further shows that if the channel model and arrival rates are known in advance, any power allocation policy which stabilizes the system—possibly by making use of special knowledge of future events—can be transformed into a stabilizing policy which considers only the current channel state.

Theorem 1. *(Downlink Capacity) The capacity region of the downlink system of Fig. 3-2 with power constraint P_{tot} and rate-power curves $\mu_i(P_i, S_i)$ is the set of all rate vectors $\vec{\lambda}$ such that there exist power levels $P_i^{\vec{S}}$ satisfying $\sum_i P_i^{\vec{S}} \leq P_{tot}$ for all channel states $\vec{S}$ and such that*

$$\lambda_i \leq \sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(P_i^{\vec{S}}, S_i\right) \quad \textit{for all } i \in \{1, \ldots, N\} \tag{3.2}$$

Proof. Using the stationary policy of allocating a power vector $\vec{P}^{\vec{S}} = \left(P_1^{\vec{S}}, \ldots, P_N^{\vec{S}}\right)$ whenever the system is in channel state $\vec{S}$ creates a server rate $\mu_i(t) = \mu_i(P_i(t), S_i(t))$ for all queues i, which is rate convergent with average rates given by the right-hand side of inequality (3.2). Thus, the single-queue delay result of Lemma 3 implies that each queue is stable with bounded delay whenever the rate vector $\vec{\lambda}$ satisfies (3.2) with strict inequality in all entries. We now show that restricting power control to such stationary policies (which use only the current channel state $\vec{S}$ when making power allocation decisions) does not restrict the capacity region and, hence, the region (3.2) captures all input rates which yield stable systems.

Suppose all queues of the downlink can be stabilized with some power control function $\vec{P}(t)$ which meets the power constraints—perhaps a function derived from a policy which knows future events. Under any such scheme, we have the relationship between arrivals, potential service opportunities, and unfinished work:

$$X_i(t) \leq U_i(t) + \int_0^t \mu_i(P_i(\tau), S_i(\tau))d\tau \tag{3.3}$$

This holds because, assuming all queues are initially empty, the total bits that arrive during $[0,t]$ must be less than or equal to the current backlog plus the total bits that could have been served. Let $T_{\vec{S}}(t)$ represent the subintervals of $[0,t]$ during which the channel is in state $\vec{S}$, and define $||T_{\vec{S}}(t)||$ as the total length of these subintervals. Fix $\epsilon > 0$ and let $\left|\vec{S}\right|$ represent the total number of channel states of the system (i.e., it is the product of the cardinalities of the number of states for each channel i). Because the arrival processes are rate convergent, the channel processes are channel convergent, and there are a finite number of queues and channel states, there must exist a time t_1 such that the time-average fraction of time in each channel state and the time-average arrival rates are simultaneously within ϵ of their limiting values for any $t \geq t_1$:

$$\begin{aligned} \frac{X_i(t)}{t} &\geq \lambda_i - \epsilon \qquad \text{for all } i && (3.4) \\ \frac{||T_{\vec{S}}(t)||}{t} &\leq \pi_{\vec{S}} + \epsilon \qquad \text{for all } \vec{S} && (3.5) \end{aligned}$$

However, by the stability necessary condition of Lemma 1, there must exist a threshold value V such that arbitrarily large times $\tilde{t}$ can be found so that $\sum_i U_i(\tilde{t}) \leq V$ with probability at least 1/2. Choose such a time $\tilde{t}$ for which $\tilde{t} \geq t_1$ and $V/\tilde{t} \leq \epsilon$. Considering (3.3) at time $\tilde{t}$ and using (3.4) and (3.5), the following inequality simultaneously holds true for all i with probability at least 1/2:

$$\lambda_i - \epsilon \leq \frac{X_i(\tilde{t})}{\tilde{t}} \leq \epsilon + \frac{1}{\tilde{t}} \int_0^{\tilde{t}} \mu_i(P_i(\tau), S_i(\tau))d\tau \tag{3.6}$$

By breaking the integral into a sum over intervals corresponding to distinct channel states, we have for all i:

$$\begin{aligned}
\lambda_i &\leq 2\epsilon + \sum_{\vec{S}} \frac{||T_{\vec{S}}(\tilde{t})||}{\tilde{t}} \frac{1}{||T_{\vec{S}}(\tilde{t})||} \int_{\tau \in T_{\vec{S}}(\tilde{t})} \mu_i(P_i(\tau), S_i) d\tau \\
&\leq 2\epsilon + \sum_{\vec{S}} \frac{||T_{\vec{S}}(\tilde{t})||}{\tilde{t}} \mu_i \left(\frac{1}{||T_{\vec{S}}(\tilde{t})||} \int_{\tau \in T_{\vec{S}}(\tilde{t})} P_i(\tau) d\tau, S_i \right) \qquad (3.7) \\
&\leq 2\epsilon + \sum_{\vec{S}} (\pi_{\vec{S}} + \epsilon) \mu_i \left(\frac{1}{||T_{\vec{S}}(\tilde{t})||} \int_{\tau \in T_{\vec{S}}(\tilde{t})} P_i(\tau) d\tau, S_i \right) \qquad (3.8)
\end{aligned}$$

where (3.7) follows from concavity of the $\mu_i(P, S_i)$ functions with respect to the power variable P, and (3.8) follows from (3.5). We define for all states $\vec{S}$ and queues i:

$$\tilde{P}_i^{\vec{S}} \triangleq \frac{1}{||T_{\vec{S}}(\tilde{t})||} \int_{\tau \in T_{\vec{S}}(\tilde{t})} P_i(\tau) d\tau \qquad (3.9)$$

Hence, from (3.8) and (3.9)

$$\lambda_i \leq \sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(\tilde{P}_i^{\vec{S}}, S_i \right) + \epsilon (2 + \left| \vec{S} \right| \mu_{max}) \text{ for all } i \qquad (3.10)$$

where μ_{max} is defined as the maximum processing rate of a queue (maximized over all queues and channel states) when it is allocated the full power P_{tot}.

Because the original power function satisfies the power constraint $\sum_i P_i(t) \leq P_{tot}$ for all times t, from (3.9) it is clear that the $\tilde{P}_i^{\vec{S}}$ values satisfy the constraint $\sum_i \tilde{P}_i^{\vec{S}} \leq P_{tot}$ for all channel states $\vec{S}$.

Recall now that the inequality (3.10) is not guaranteed to hold, but holds simultaneously for all i with probability at least 1/2. Thus, *there must exist* a set of power values $\tilde{P}_i^{\vec{S}}$ satisfying the power constraint for all channel states $\vec{S}$ such that (3.10) holds simultaneously for all i (otherwise, the probability of (3.10) holding would be zero). Thus, (3.10) indicates that the arrival vector $\vec{\lambda}$ is arbitrarily close to a point in the region specified by (3.2). Because the region (3.2) is closed, it must contain $\vec{\lambda}$. Thus, the region (3.2) represents the capacity region of the system. □

In the case when the channel does not vary but stays fixed, the rate-power curve for each queue i is given by $\mu_i(P)$, and the expression for the downlink capacity region in Theorem 1 can be greatly simplified:

Corollary 1. *(Static Channel Capacity) The downlink capacity region for static channels is the set of all rate vectors $\vec{\lambda}$ such that:*

$$\sum_{i=1}^{N} \mu_i^{-1}(\lambda_i) \leq P_{tot}$$

where

$$\mu_i^{-1}(\lambda_i) = \begin{cases} \text{The smallest } P \text{ such that } \mu_i(P) = \lambda \\ \infty \ \text{ if no such } P \text{ exists} \end{cases}$$

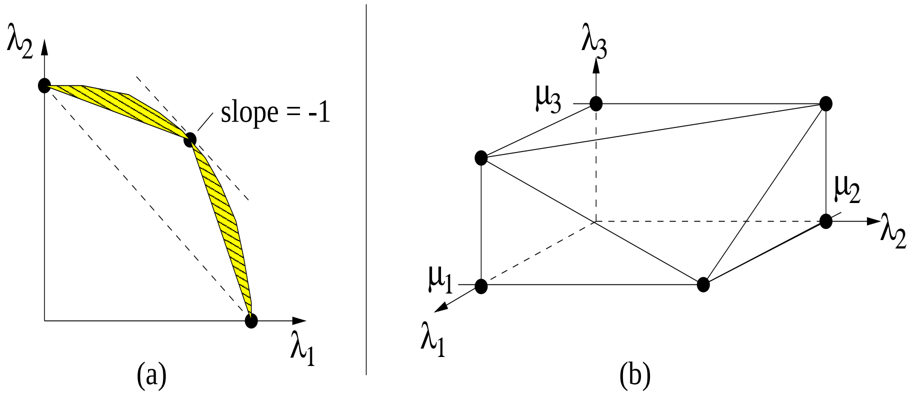


Figure 3-5: Capacity regions for static channels. (a) Two-queue system with power allocation. (b) K-server allocation problem with $K = 2, N = 3$.

In Fig. 3-5a we illustrate a general capacity region for $N = 2$ channels with fixed channel states and concave power curves $\mu_1(P)$ and $\mu_2(P)$. In this case of fixed channel states, one might suspect the optimal solution to be the one which maximizes the instantaneous output rate at every instant of time: allocate full power to one queue whenever the other is empty, and allocate power to maximize the sum output rate $\mu_1(P_1) + \mu_2(P_2)$ subject to $P_1 + P_2 \leq P_{tot}$ whenever both queues are full. Doing this restricts the throughput region to linear combinations of the three operating points, as illustrated in Fig. 3-5a. The shaded regions in the figure represent the capacity gains obtained by power allocation using the full set of power levels. Note that the region is restricted further if only ON/OFF allocations are considered.

Corollary 2. *(Server Allocation Capacity) For the $K-server$ allocation problem where the channel rate of queue i is μ_i when it is allocated a server (and 0 otherwise), the capacity* *region is the polytope set of all $\vec{\lambda}$ vectors such that*

$$\textstyle\sum_i \frac{\lambda_i}{\mu_i} \leq K \tag{3.11}$$

$$0 \leq \lambda_i \leq \mu_i \ , \ \ \textit{for all } i \in \{1, \ldots, N\} \tag{3.12}$$

Proof. Using the virtual power curves and constraints given in Section 3.2, we find by Corollary 1 that the polytope region described by (3.11) and (3.12) contains the true capacity region. However, the K-server problem is constrained to allocate rates only on the vertex points of the polytope (see Fig. 3-5b). Timesharing amongst vertex points, however, achieves any desired point within the polytope. □

3.4 A Dynamic Power Allocation Algorithm

Theorem 1 implies that stability of the downlink channel can be achieved by a stationary power allocation policy which allocates power levels $P_i^{\vec{S}}$ whenever the channel is in state $\vec{S}$. Such power levels can in principle be calculated with full knowledge of the arrival rates λ_i and channel state probabilities $\pi_{\vec{S}}$. However, such computation is impractical if the number of channel states is large, and cannot be done if the arrival and channel state probabilities are unknown. Here we present a power allocation policy which stabilizes the system at every point of the capacity region (3.2) without using the arrival and channel state probabilities. In essence, the policy learns the system parameters indirectly by basing power allocation decisions both on channel state and queue backlog information. Furthermore, because the policy is not bound to a particular set of system parameters, it is shown to be robust to arbitrary changes in the input rates λ_i. The policy operates as follows:

Dynamic Power Allocation Policy: Every timeslot t, observe the unfinished work vector $\vec{U}(t)$ and the channel state vector $\vec{S}(t)$ and allocate a power vector $\vec{P}(t) = (P_1(t), \ldots P_N(t))$ which is a solution to the following maximization problem:

$$\begin{aligned} \text{Maximize:} \quad & \textstyle\sum_{i=1}^{N} \theta_i U_i(t) \mu_i(P_i, S_i(t)) \\ \text{Subject to:} \quad & \textstyle\sum_i P_i \leq P_{tot} \end{aligned} \tag{3.13}$$

where $\{\theta_i\}$ is any arbitrary set of positive weights.

The weights $\{\theta_i\}$ can be chosen to allow the more heavily weighted queues to have better delay guarantees, as described subsequently. Notice that this policy acts only through the current value of $\vec{U}$ and $\vec{S}$ without specific knowledge of the arrival rates or channel probabilities. Intuitively, we desire a policy that gives more power to queues with currently high processing rates (to achieve maximum throughput) as well as gives more power to queues with large backlog (to ensure that these queues are stabilized). The above policy does both by considering as a metric the product of backlog and data rate for each queue.

A similar metric is used in [99] for scheduling packets of different classes over a single server, and in [133] [5] for scheduling a single server over a set of parallel queues. Maximum weight metrics are also considered in the switching and scheduling literature [95] [97] [88] [132] [81] [62], and recently for multi-access uplink communication in [149] [84] and for a single server downlink with heavy traffic in [124].

We analyze the above dynamic strategy by comparing it with the stationary policy. Suppose the channel process $\vec{S}(t)$ is channel convergent with steady state probabilities $\pi_{\vec{S}}$. Likewise, assume the arrival processes $A_i(t)$ are rate convergent with rates λ_i. We assume these arrival rates are strictly interior to the downlink capacity region Λ, so that for some positive $\epsilon > 0$ the vector $(\lambda_1 + \epsilon, \ldots, \lambda_N + \epsilon)$ is also inside Λ. The value of ϵ represents a measure of the distance the arrival rate vector is to the boundary of the capacity region. From the capacity equation (3.2), it follows that there exist power levels $P_i^{\vec{S}}$ satisfying the power constraints and such that for all i:

$$\lambda_i + \epsilon \leq \sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(P_i^{\vec{S}}, S_i \right) \tag{3.14}$$

Consider the stationary policy of allocating these power levels based on channel state. To be explicit, let us denote $\vec{P}^{stationary}(\vec{S}) = (P_1^{stationary}(\vec{S}), \ldots, P_N^{stationary}(\vec{S}))$ as the power vector chosen according to this stationary policy when the system is in channel state $\vec{S}$. The server rate processes under this policy are given by $\mu_i^{stationary}(t) = \mu_i \left(P_i^{stationary}(\vec{S}(t)), \vec{S}(t) \right)$ for all queues i. By Lemma 4 it follows that these server processes are rate convergent with rates $\sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(P_i^{\vec{S}}, S_i \right)$ (because the underlying channel process is channel convergent).

Now define K as the smallest number such that at any time t_0 and regardless of past history, we have for all i:

$$\frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \mathbb{E} \left\{ \mu_i^{stationary}(\tau) - A_i(\tau) \right\} \geq \epsilon/2 \tag{3.15}$$

A finite value of K must exist because for each $i \in \{1, \ldots, N\}$, the server process is rate convergent with average rate given by the right hand side of (3.14) and the arrival process $A_i(t)$ is rate convergent with average rate λ_i.[2]

The value of K represents the timescale over which we can expect the system to yield steady state behavior, and is important in proving stability and establishing a delay bound. We additionally assume that the second moment of the total arrival process is bounded, so that $\mathbb{E}\left\{(\sum_i A_i(t))^2\right\} \leq A_{max}^2$ on every timeslot t, regardless of past history.

Theorem 2. *(Dynamic Power Allocation) The dynamic power allocation policy of choosing a power vector $\vec{P}$ to maximize $\sum_i \theta_i U_i(t)\mu_i(P_i, S_i(t))$ subject to the power constraint stabilizes the downlink system whenever the arrival rate vector λ is strictly interior to the capacity region Λ.*

Furthermore, the average unfinished work in the queues satisfies:

$$\overline{\sum_i \theta_i U_i} \leq \frac{K\theta_{max}B}{\epsilon} + \frac{(K-1)\theta_{max}\tilde{B}}{\epsilon}$$

where $B = (A_{max}^2 + \mu_{max}^2)$, $\tilde{B} = \mu_{max}^{out}(A_{max} + \mu_{max}^{out})/2$, and μ_{max}^{out} represents the maximum sum output rate over all servers (optimized over all channel states and power allocation distributions conforming to the power constraint).

We note that when arrivals and channel states are i.i.d. every timeslot, then the bound reduces to $\overline{\sum_i \theta_i U_i} \leq \frac{B\theta_{max}}{\epsilon}$.

We prove the theorem by first proving stability of a frame-based modification of the dynamic power allocation strategy, which operates on frames of duration K slots and allocates power every slot t to maximize $\sum_i \theta_i U_i(t_0)\mu_i(P_i, S_i(t))$ subject to the power constraint, where $U_i(t_0)$ represents the unfinished work in the system at the beginning of a frame. This frame-based scheme is thus identical to the dynamic power allocation policy with the exception that it uses out-of-date backlog information equal to the backlog present in the system at the start of the frame. Note that, unlike the dynamic power allocation policy (3.13), this frame based scheme cannot be implemented without knowledge of the frame size K. Let $\vec{P}^{frame}(\vec{S}, \vec{U}_0)$ represent the power allocation decisions of this frame based scheme as a function of the channel state and initial backlog.

[2] Note that $\lim_{K\to\infty} \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{\mu_i^{stationary}(\tau) - A_i(\tau)\right\} \geq \epsilon$ by (3.14).

Lemma 6. *(Frame Based Allocation) The frame based power allocation policy stabilizes the system and ensures unfinished work satisfies* $\sum_i \theta_i \overline{U}_i \leq \frac{K\theta_{max}B}{\epsilon}$.

Proof. We make use of the sufficient condition for network stability using Lyapunov drift (Lemma 2 of Chapter 2). Consider the K-step dynamics for unfinished work:

$$U_i(t_0+K) \leq \max\left[U_i(t_0) - \sum_{\tau=t_0}^{t_0+K-1}\mu_i(\tau), 0\right] + \sum_{\tau=t_0}^{t_0+K-1} A_i(\tau)$$

To simplify notation, we define: $U_i \triangleq U_i(t_0)$, $\mu_i \triangleq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mu_i(\tau)$, $A_i \triangleq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} A_i(\tau)$. Squaring both sides of the above inequality yields:

$$\begin{aligned} U_i^2(t_0+K) &\leq (U_i - K\mu_i)^2 + K^2(A_i)^2 + 2KA_i \max[U_i - K\mu_i, 0] \\ &\leq (U_i - K\mu_i)^2 + K^2(A_i)^2 + 2KA_iU_i \\ &\leq U_i^2 + K^2\mu_i^2 + K^2A_i^2 - 2KU_i[\mu_i - A_i] \end{aligned} \tag{3.16}$$

Now define the Lyapunov function $L(\vec{U}) = \sum_i \theta_i U_i^2$. Multiplying (3.16) by the weight θ_i and summing over all i, we find:

$$L(\vec{U}(t_0+K)) - L(\vec{U}(t_0)) \leq \theta_{max}K^2\left(\sum_i A_i^2 + \sum_i \mu_i^2\right) - 2K\sum_i \theta_i U_i(t_0)[\mu_i - A_i] \tag{3.17}$$

Taking conditional expectations given $\vec{U}(t_0)$ and using the inequalities $\mathbb{E}\left\{\sum_i A_i^2 \middle| \vec{U}(t_0)\right\} \leq A_{max}^2$ and $\mathbb{E}\left\{\sum_i \mu_i^2 \middle| \vec{U}(t_0)\right\} \leq (\mu_{max}^{out})^2$ (which are proven in Chapter Appendix 3.A at the end of this chapter), we have:

$$\mathbb{E}\left\{L(\vec{U}(t_0+K)) - L(\vec{U}(t_0)) \middle| \vec{U}(t_0)\right\} \leq K^2\theta_{max}B - 2K\sum_i \theta_i U_i(t_0)\mathbb{E}\left\{[\mu_i - A_i] \middle| \vec{U}(t_0)\right\} \tag{3.18}$$

The above equation represents Lyapunov drift for any power allocation algorithm yielding server rates $\mu_i(t)$. If the stationary power allocation is used so that the server process is $\mu_i^{stationary}(t)$, the expression (3.15) can be inserted directly into the right hand side of the drift equation above, proving that the K-step drift under the stationary policy is less than or equal to $K^2\theta_{max}B - 2K\sum_i \theta_i U_i(t_0)[\epsilon/2]$. By Lemma 2, it follows that unfinished work under the stationary policy satisfies $\sum_i \theta_i \overline{U}_i^{stationary} \leq \frac{K\theta_{max}B}{\epsilon}$.

However, the frame based allocation algorithm is designed to maximize the quantity

$\mathbb{E}\left\{\sum_i \theta_i U_i(t_0)[\mu_i - A_i] \left|\vec{U}(t_0)\right.\right\}$ (which is contained in the drift expression (3.18)) over all other power allocations which conform to the power constraint. In particular, notice that at any time t such that $t_0 \leq t \leq t_0 + K - 1$, we have:

$$\sum_i \theta_i U_i(t_0)\mu_i\left(P_i^{frame}(\vec{S}(t), \vec{U}_0), S_i(t)\right) \geq \sum_i \theta_i U_i(t_0)\mu_i\left(P_i^{stationary}(\vec{S}(t)), S_i(t)\right)$$

This inequality holds deterministically for all initial backlogs $\vec{U}(t_0)$ and all channel states $S_i(t)$ (where $t_0 \leq t \leq t_0 + K - 1$), and is hence preserved under conditional expectations. It follows that the Lyapunov drift of the frame algorithm is less than or equal to the bound $K^2\theta_{max}B - 2K\sum_i \theta_i U_i(t_0)[\epsilon/2]$ computed for the stationary algorithm. Using this bound together with Lemma 2 proves the result. □

The frame based algorithm minimizes the K-step Lyapunov drift, but uses out-of-date backlog information during the last $K - 1$ slots of the frame. Intuitively, the dynamic power allocation algorithm of Theorem 2 should offer better performance, as it uses current values of queue backlog on each slot. However, analytically we can only prove that the performance of this algorithm is no more than a fixed amount worse than the frame based scheme, as described in the next lemma. Let $P_i^{dynamic}(t)$ and $\mu_i^{dynamic}(t)$ represent the power and server rate under the dynamic power allocation policy.

Lemma 7. *Given an initial timeslot t_0, a common backlog vector $\vec{U}(t_0)$, and any time t such that $t_0 \leq t \leq t_0 + K - 1$, we have:*

$$\sum_i \theta_i U_i(t_0)\mathbb{E}\left\{\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mu_i^{dynamic}(\tau)\left|\vec{U}(t_0)\right.\right\} \geq$$
$$\sum_i \theta_i U_i(t_0)\mathbb{E}\left\{\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mu_i^{frame}(\tau)\left|\vec{U}(t_0)\right.\right\} - (K-1)\theta_{max}\tilde{B}$$

Proof. The dynamic power allocation policy optimizes (3.13) on each timeslot. Hence, for all times τ such that $t_0 \leq \tau \leq t_0 + K - 1$, we have:

$$\sum_i \theta_i U_i^{dynamic}(\tau)\mu_i(P_i^{dynamic}(\tau), S_i(\tau)) \geq \sum_i \theta_i U_i^{dynamic}(\tau)\mu_i(P_i^{frame}(\tau), S_i(\tau))$$

Define $\Delta_i(\tau) \triangleq U_i^{dynamic}(\tau) - U_i(t_0)$ as the change in backlog in queue i after running the dynamic power allocation algorithm during slots $\{t_0, \ldots, \tau\}$. Plugging this definition into the above inequality yields:

$$\begin{aligned}\mathbb{E}\left\{\sum_i \theta_i U_i(t_0)\mu_i(P_i^{dyn}(\tau), S_i(\tau))\left|\vec{U}(t_0)\right.\right\} \;\; &\geq \;\; \mathbb{E}\left\{\sum_i \theta_i U_i(t_0)\mu_i(P_i^{frame}(\tau), S_i(\tau))\left|\vec{U}(t_0)\right.\right\} - \\ &\mathbb{E}\left\{\sum_i \theta_i \Delta_i(\tau)(\mu_i^{dyn}(\tau) - \mu_i^{frame}(\tau))\left|\vec{U}(t_0)\right.\right\}\end{aligned}$$

Note that $\left|\mu_i^{dynamic}(\tau) - \mu_i^{frame}(\tau))\right| \leq \mu_{max}^{out}$. Furthermore, $\mathbb{E}\left\{\sum_i |\Delta_i(\tau)|\right\}$ represents the maximum possible change in total backlog in the satellite after $\tau - t_0$ slots, and is upper bounded by $(A_{max} + \mu_{max}^{out})(\tau - t_0)$. (Note that $\mathbb{E}\left\{A_i(\tau)\right\} \leq \sqrt{\mathbb{E}\left\{A_i^2(\tau)\right\}} \leq A_{max}$). Summing from $\tau = t_0$ to $\tau = t_0 + K - 1$, it follows that:

$$\begin{aligned}\sum_i \theta_i U_i(t_0)\mathbb{E}\left\{\sum_{\tau=t_0}^{t_0+K-1} \mu_i^{dynamic}(\tau)\left|\vec{U}(t_0)\right.\right\} \geq \sum_i \theta_i U_i(t_0)\mathbb{E}\left\{\sum_{\tau=t_0}^{t_0+K-1} \mu_i^{frame}(\tau)\left|\vec{U}(t_0)\right.\right\} - \\ \theta_{max}\mu_{max}^{out}(A_{max} + \mu_{max}^{out})\sum_{\tau=t_0}^{t_0+K-1}(\tau - t_0)\end{aligned}$$

The lemma follows by noting that $\mu_{max}^{out}(A_{max} + \mu_{max}^{out})\sum_{\tau=t_0}^{t_0+K-1}(\tau - t_0) = \tilde{B}(K-1)K$. □

Lemma 7 applied to inequality (3.18) in Lemma 6 together prove Theorem 2.

Note that the positive weights $\{\theta_i\}$ in the dynamic power allocation algorithm (3.13) can be chosen arbitrarily. Larger weights can be given to specific queues to improve their relative performance according to the downlink performance bound of Theorem 2. Choosing weights $\theta_i = 1$ for all i yields a policy which chooses a power vector that maximizes $\sum_i U_i\mu_i$ at every timestep. The following corollary makes use of a different set of weights.

3.4.1 Serve the K Longest Queues

Consider again the N-queue, K-server allocation problem where each queue has only 2 channel states, ON or OFF, and these states form the N dimensional vector process $\vec{S}(t)$ which is channel convergent with some steady state distribution. When a server is allocated to queue i while it is in the ON state, the server transmits data from the queue at a rate μ_i (the transmission rate is zero when in the OFF state or when no server is allocated).

Corollary 3. *(Dynamic Server Allocation) For the K-server allocation problem with ON/OFF* *channel states, the policy of allocating the K servers to the K longest ON queues stabilizes the system whenever the system is stabilizable.*

Proof. Assume the system operates according to the *virtual rate-power curves* of (3.1) in Section 3.2. The capacity region under these curves clearly contains the capacity region when power is restricted to the vertex points—corresponding to a feasible allocation of K servers to K queues. Define weights $\theta_i \overset{\triangle}{=} 1/\mu_i$, and implement the stabilizing policy of allocating power to maximize $\sum_i \frac{1}{\mu_i} U_i(t)\tilde{\mu}_i(P_i, S_i)$ (where $S_i \in \{ON, OFF\}$). Clearly this optimization needs not place any power on queues in the OFF state, so the summation can be restricted to queues that are ON:

$$\text{Maximize:} \quad \sum_{\{i|\, S_i = ON\}} U_i(t) \frac{\tilde{\mu}_i(P_i, ON)}{\mu_i} \tag{3.19}$$

$$\text{Subject to:} \quad \sum_i P_i \leq K \tag{3.20}$$

Notice that the above maximization effectively chooses a rate vector $\vec{\mu}$ within the polytope capacity region specified in (3.11) and (3.12). The optimal solution for maximizing a linear function over a polytope will always be a vertex point. Fortunately, such a vertex point corresponds to the feasible allocation of K servers (with full power $P_i = 1$) to K queues. Considering (3.19), the optimal way to do this is to choose the K queues with the largest value of $U_i(t)$. Thus, stability is attained by using pure server allocation, which also proves the capacity region of pure server allocation is the same as the capacity region corresponding to the virtual power curves $\tilde{\mu}()$. □

Using the same reasoning as in the proof above, it follows that the power allocation policy of Theorem 2 naturally chooses a vertex point when power curves are piecewise linear. It follows that optimization can be restricted to searches over the vertex points without loss of optimality.

3.4.2 On the Link Weights θ_i

[3] Although any positive weights θ_i can be used to stabilize the system, dynamically varying these rates cannot guarantee stability for the full capacity region Λ, but only guarantees

[3] This section corrects a claim in paragraph 6, Section IV.D in [111] suggesting that dynamically varying the weights still guarantees stability.

stability for the scaled region $\frac{\theta_{min}}{\theta_{max}}\Lambda$ (where θ_{max} and θ_{min} represent the maximum and minimum weight). This follows because optimizing over any set of weights that are upper and lower bounded by θ_{max} and θ_{min} yields a solution which is within a factor $\theta_{min}/\theta_{max}$ from the corresponding solution for any alternate set of weights conforming to the same bounds.

A counter-example illustrating that dynamic link weights may not achieve stability is a parallel queue system with 2 queues, a single server, and independently time varying output rates. Suppose that every timeslot, the rate of each queue is independently 1 or 2 with equal probability, but weights θ_i vary so that whenever any queue has output rate 2, the weight for that queue is θ, while the weight for any queue with output rate 1 is 2θ. Thus, the multiplication of rate and weight is constant every timeslot, so the system can do nothing but serve the longest queue. Such a policy cannot stabilize the system for all data rates within the capacity region, as it cannot take advantage of good channel states while they last. It does, however, provide stability when the system is half loaded.

3.4.3 Real Time Implementation

The dynamic power allocation policy of the previous section requires solving a nonlinear optimization problem every timeslot (eq.(3.13)). However, because the rate curves $\mu_i()$ are concave in the power parameter for every fixed channel state, the solution can be computed efficiently. Indeed, for positive weights $\{\theta_i\}$ and known unfinished work and channel state vectors $\vec{U}(t)$ and $\vec{S}(t)$, the problem (3.13) becomes a standard concave maximization problem: Maximize $\sum_i \theta_i U_i(t)\mu_i(P_i, S_i(t))$, subject to the simplex constraint $\sum_i P_i \leq P_{tot}$. Using standard Lagrange multiplier techniques [13], it can be shown that a solution is optimal if and only if power is allocated so that the power constraint is met and the scaled derivatives $\theta_i U_i(t)(d/dP_i)\mu_i(P_i, S_i)$ are equalized to some value γ^* for all queues i which receive nonzero power, while all queues which receive zero power have scaled derivatives less than γ^*. A fast bisection-type algorithm can be constructed to find such a solution, where a bracketing interval $[\gamma_1, \gamma_2]$ is found which contains γ^*, and the interval size is decreased iteratively by testing the midpoint γ value to see if the corresponding powers sum to more or less than the power constraint P_{tot}. An illustration of this is given in Fig. 3-6. Such an algorithm yields power allocations whose proximity to the optimal solution converges geometrically with each iteration.

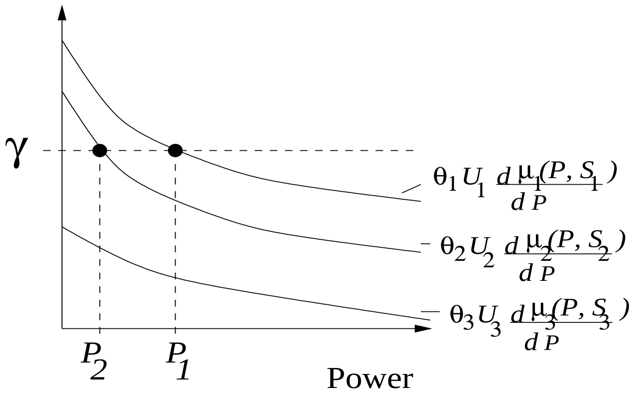


Figure 3-6: An iterative routine for equalizing scaled derivatives of the $\theta_i U_i \mu_i(P, S_i)$ functions.

An important set of rate-power curves to consider are the standard curves for Shannon capacity:

$$\mu_i(P_i, \alpha_i) = \log(1 + \alpha_i P_i)$$

where α_i represents the attenuation-to-noise level for downlink channel i during a particular timeslot. With these curves, the solution to (3.13) is found by the following computation:

$$\begin{aligned}
\Omega &= \text{Set of downlinks } i \in \{1, \ldots, N\} \text{ such that } U_i(t) > 0 \\
P_i &= \frac{\theta_i U_i(t) \left(P_{tot} + \sum_{j\in\Omega} \frac{1}{\alpha_j}\right)}{\sum_{j\in\Omega} \theta_j U_j(t)} - \frac{1}{\alpha_i} \text{ , if } i \in \Omega \\
P_i &= 0 \text{ , if } i \notin \Omega
\end{aligned} \tag{3.21}$$

The above equations produce the optimal power allocations whenever the resulting P_i values are non-negative. If any P_i values are negative, these are set to zero, the corresponding i-indices are removed from the set Ω, and the calculation is repeated—a process ending in at most $N - 1$ iterations.

3.4.4 Robustness to Input Rate Changes

Here we consider the case when arrivals are independent from slot to slot, but the input rates $\vec{\lambda}$ can vary. We demonstrate that the dynamic power allocation policy is robust to arbitrary changes in the input rates as long as the resulting rate vectors $\vec{\lambda}_t$ remain within the capacity region at each timestep. Specifically, suppose that the input rate to the downlink system is $\vec{\lambda}^{(1)}$ for a certain duration of time, then changes to $\vec{\lambda}^{(2)}$—perhaps due to changing user demands. This change will be reflected in the backlog that builds up in the queues of the system. Because the power allocation algorithm bases decisions on the size of the queues, it reacts smoothly to such changes in the input statistics.

Formally, this situation is modeled by defining an input distribution $f_t(\vec{A})$ on the arrival vector $\vec{A}(t)$ every timeslot. The $f_t(\vec{A})$ distributions are arbitrary and unknown to the network controller, although we assume they yield input rates $\vec{\lambda}_t = \mathbb{E}\left\{\vec{A}(t)\right\}$, all of which are within a distance ϵ of the capacity region (so that $(\vec{\lambda}_t + \vec{\epsilon}) \in \Lambda$ for all t). We further assume that second moments are bounded, so that $\mathbb{E}\left\{(\sum_i A_i(t))^2\right\} \leq A_{max}^2$ for all t. We note that because the distributions are varying arbitrarily every timeslot, there is no notion of a steady state arrival rate. However, a meaningful performance bound can be developed using the $\limsup$.

Let K be the smallest time interval over which any target boundary point on the boundary of the capacity region Λ can be achieved to within a distance of $\epsilon/2$ using some stationary power allocation policy $\vec{P}^{stationary}(\vec{S})$ which allocates a fixed power vector whenever the channel is in state $\vec{S}$. Specifically, given any target rate point $\vec{r} \in \Lambda$, assume that K is large enough so that for all i, $\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{\mu_i\left(P_i^{stationary}(\vec{S}(\tau)), S_i(\tau)\right)\right\} \geq r_i - \epsilon/2$. Likewise note that $\mathbb{E}\left\{\frac{1}{K}\left(\vec{A}(t_0) + \vec{A}(t_0+1) + \ldots + \vec{A}(t_0+K-1)\right)\right\}$ yields another rate vector $\vec{r}_{in}$ such that $\vec{r}_{in} + \vec{\epsilon} \in \Lambda$ (and hence we can choose $\vec{r}$ such that $\vec{r} = \vec{r}_{in} + \vec{\epsilon}$). The K-step drift can thus be computed exactly as in the proof of Theorem 2, yielding

$$\mathbb{E}\left\{L(\vec{U}(t_0+K)) - L(\vec{U}(t_0)) \,\middle|\, \vec{U}(t_0)\right\} \leq K^2\theta_{max}B + K(K-1)\theta_{max}\tilde{B} - 2K\sum_i \theta_i U_i(t_0)[\epsilon/2]$$

Thus, using Lemma 2, the following performance bound is guaranteed:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_i \theta_i \mathbb{E}\left\{U_i(\tau)\right\}\right] \leq \frac{K\theta_{max}B}{\epsilon} + \frac{(K-1)\theta_{max}\tilde{B}}{\epsilon}$$

3.5 Joint Routing and Power Allocation

We consider now a collection of M multi-beam satellites and develop a method for jointly routing packets and allocating power over the downlinks. Each satellite has multiple output queues (corresponding to multiple downlink channels) and operates according to individual power constraints (Fig. 3-7). Every timeslot, packets enter the system from N input streams according to input processes $X_1(t), \ldots, X_N(t)$ with arrival rates $(\lambda_1, \ldots, \lambda_N)$. Each input stream i can route incoming packets to a subset of the output queues, where the subsets may overlap with each other and may contain queues from different satellites. The problem is to jointly route packets and allocate power to each of the downlinks in order to stabilize the system and ensure maximum throughput.

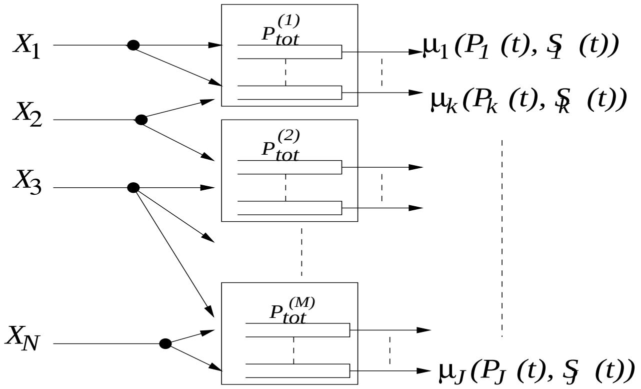


Figure 3-7: A multiuser multisatellite system with joint routing and power control. User X_i can route to queues within set Q_i. Satellite m allocates power subject to $\sum_{j \in Sat(m)} P_j(t) \leq P_{tot}^{(m)}$.

Such a scenario arises, for example, when several satellites have a connection to the same ground unit, and hence packets destined for this unit have several routing options. Alternatively, the routing options may represent a cluster of ground locations connected together by a reliable ground network. In this case, packets arrive to the cluster from the downlinks and are routed to their final destinations using the wire lines on the ground. We note that the formulation of this joint routing and power allocation problem also applies to wireless systems, where base stations communicate with users over a wireless network.

Let J represent the total number of output queues (summed over all satellites), and let each output queue be indexed with a single integer $j \in \{1, \ldots, J\}$. For each satellite m, let $Sat(m)$ represent the set of output queues which it contains (hence, $Sat(m) \subset \{1, \ldots, J\}$ for all m). Likewise, for each input stream i, let Q_i represent the set of all output queues that input i can route packets to (where $Q_i \subset \{1, \ldots, J\}$). Note that the Q_i subsets are arbitrary and need not be disjoint. Channel states vary according to a J-dimensional vector $\vec{S}(t)$ and are channel convergent with steady state probabilities $\pi_{\vec{S}}$. Similarly, the vector of queue backlogs $\vec{U}(t)$ is J-dimensional.

Every timeslot, routing decisions are made and a power vector $\vec{P}(t)$ is allocated according to the individual power constraints of each satellite. In general, the full queue state and channel state vectors $\vec{U}(t)$ and $\vec{S}(t)$ are important in both the routing and power allocation decisions. For example, more power should be allocated to queues which are expected to grow large—which is dependent on the state of unfinished work in other satellites as well as on future routing decisions. Likewise, a router should place packets in faster queues (especially if these rates are likely to be high for one or more timeslots) and should avoid queues likely to be congested because of high contention with other input sessions.

However, here we show that the routing and power allocation decisions can be *decoupled* into two policies: A routing policy which considers only $\vec{U}(t)$, and a power allocation policy which considers both $\vec{U}(t)$ and $\vec{S}(t)$. Furthermore, a router for stream i needs only to consider the entries of the unfinished work vector $\vec{U}(t)$ within the set Q_i of queues to which it can route. Likewise, the power allocation decisions use information local to each satellite: Power is allocated in satellite m based only on the unfinished work and channel state information for queues in $Sat(m)$. The resulting strategy stabilizes the system whenever the system is stabilizable.

Joint Routing and Power Allocation Algorithm:

We assume that enough is known about the channel to identify and remove from the set of routing options any queues which produce zero output rate for all channel states and power allocations.[4] Hence, in the algorithm below, we assume that all queues j have a nonzero probability of being in a functional channel state.

-Power Allocation: At each timestep, each satellite m allocates power as before, using the $\vec{U}(t)$ and $\vec{S}(t)$ vectors to maximize $\sum_{j\in Sat(m)} \theta_j U_j(t)\mu_j(P_j(t), S_j(t))$ subject to $\sum_{j\in Sat(m)} P_j(t) \leq P_{tot}^m$.

-Routing: All packets from stream i are routed to the queue $j \in Q_i$ with the smallest amount of unfinished work.

We note that *Join-the-Shortest-Queue (JSQ)* routing was shown to be delay optimal in [45] for the case of arbitrary arrivals to a set of two homogeneous servers with i.i.d. and exponential service times, and an extension in [142] demonstrates that a threshold policy is optimal when the two exponential servers have different service rates. In [147] an analysis of the waiting time distribution for the *JSQ* strategy is presented for a system with a Poisson

[4] One way to achieve this is to avoid routing to any queue until its associated channel has a demonstrated history of being functional.

input stream, multiple homogeneous servers, and i.i.d. exponential service times. Here, we use the *JSQ* strategy in conjunction with power allocation in a system with multiple input streams with arbitrary arrival processes which are routed over heterogeneous, time varying downlinks. Further results on stability, delay, and near-optimality for *JSQ* routing are provided in Appendix B and in [110].

Theorem 3. *(Joint Routing and Power Allocation) The capacity region Λ for the multi-satellite system with joint routing and power allocation is the set of all arrival vectors $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)$ such that there exist splitting rates (r_{ij}) and power levels $P_j^{\vec{S}}$ such that:*

$$\textstyle\sum_{j \in Q_i} r_{ij} = \lambda_i \qquad \text{for all } i \in \{1, \ldots, N\} \tag{3.22}$$

$$\textstyle\sum_{j \in Sat(m)} P_j^{\vec{S}} \leq P_{tot}^{(m)} \qquad \text{for all } m \text{ and all channel states } \vec{S} \tag{3.23}$$

$$\textstyle\sum_i r_{ij} \leq \sum_{\vec{S}} \pi_{\vec{S}} \mu_j \left(P_j^{\vec{S}}, S_j\right) \tag{3.24}$$

Furthermore, the joint routing and power allocation algorithm described above stabilizes the multi-satellite system whenever the input rates are within this capacity region.

Intuitively, the above theorem says that the system is stabilizable if the input rates can be split amongst the various queues (in accordance with the routing restrictions) so that the aggregate input rates allow each satellite to be stabilized individually.

Proof. *(That $\vec{\lambda} \in \Lambda$ is necessary for stability)* Suppose a stabilizing algorithm exists for some set of routing decisions and power controls $\vec{P}(t)$. Define $X_{ij}(t)$ to be the total amount of data the algorithm routes from input i to queue j during the time interval $[0, t]$. For simplicity, we assume the routing process is ergodic so that $\lim_{t\to\infty} X_{ij}(t)/t$ is well defined for all i and j. (The general non-ergodic case can be handled similarly to the treatment of Theorem 1, and its proof is covered as a special case of the multi-node analysis in Chapter 4). Let $\{r_{ij}\}$ represent these limiting values. The ith input stream $X_i(t)$ can be written $X_i(t) = \sum_{j \in Q_i} X_{ij}(t)$. Dividing both sides by t and taking limits, it follows that $\sum_{j \in Q_i} r_{ij} = \lambda_i$ for all i, and hence condition (3.22) holds. Note that the aggregate data rate entering any queue $j \in \{1, \ldots, J\}$ is $\sum_i r_{ij}$. Because the system is stable, the stability conditions of Theorem 1 must be satisfied for each satellite, and hence the remaining conditions (3.23) and (3.24) must also hold. □

The fact that stability within the set Λ is achievable is demonstrated using the joint routing and power allocation algorithm given above.

Proof. (Stability of the Joint Routing and Power Allocation Algorithm) For simplicity of exposition, we consider only the case when all weights θ_j are identically 1, and prove the result under the assumption that arrivals and channel states are i.i.d. from one slot to the next. Rate convergent arrivals and channel convergent processes can be treated similarly, and such general treatment is provided for the network case in Chapter 4.

Suppose the λ vector is strictly interior to Λ so that conditions (3.22)-(3.24) are satisfied even with an additional input stream of rate ϵ applied to each queue $j \in \{1, \ldots, J\}$. That is, there exist r_{ij} and $P_j^{\vec{S}}$ values such that conditions (3.22) and (3.23) hold, and such that

$$\sum_i r_{ij} + \epsilon \leq \sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(P_j^{\vec{S}}, S_j\right) \text{ for all } j \tag{3.25}$$

Define the Lyapunov function $L(\vec{U}) = \sum_j U_j^2$. Let $A_i(t)$ represent the total bits from packets arriving from stream i during slot t, and let $(a_{i1}(t), \ldots, a_{iJ}(t))$ represent the bit length of packets from stream i routed to queues $j \in \{1, \ldots, J\}$ (where $A_i(t) = \sum_j a_{ij}(t)$, and $\mathbb{E}\{A_i(t)\} = \lambda_i$). Let μ_j represent the transmission rate $\mu_j(P_j(t), S_j(t))$ of queue j during slot t under the specified power allocation policy. As in the stability proof for the dynamic power allocation policy of Theorem 2, we have for all queues j [compare with (3.18)]:

$$\mathbb{E}\left\{L(\vec{U}(t_0+1)) - L(\vec{U}(t_0)) \left| \vec{U}(t_0)\right.\right\} \leq$$
$$A_{max}^2 + M(\mu_{max}^{out})^2 - 2\sum_j U_j(t_0)\left[\mathbb{E}\left\{\mu_j \left|\vec{U}(t_0)\right.\right\} - \sum_i \mathbb{E}\left\{a_{ij}\left|\vec{U}(t_0)\right.\right\}\right] \tag{3.26}$$

where μ_{max}^{out} represents the maximum total output rate of any satellite.

The $\mathbb{E}\left\{\mu_j \left|\vec{U}(t_0)\right.\right\}$ and $\mathbb{E}\left\{a_{ij}\left|\vec{U}(t_0)\right.\right\}$ values in the above inequality are influenced by the power control and routing algorithm, respectively, and will determine the performance of the system. To examine the impact of routing, we switch the sum above to express the routing term as:

$$2\sum_i \sum_{j\in Q_i} U_j \mathbb{E}\left\{a_{ij}\left|\vec{U}(t_0)\right.\right\}$$

Notice that the given routing strategy of placing all bits from stream i in the queue $j \in Q_i$ with the smallest value of unfinished work minimizes the above term over all possible routing strategies, including the strategy of routing according to flow rates r_{ij} of condition (3.22) in Theorem 3, and hence:

$$2\sum_{i}\sum_{j\in Q_i} U_j \mathbb{E}\left\{a_{ij} \left| \vec{U}(t_0)\right.\right\} \leq 2\sum_{i}\sum_{j\in Q_i} U_j r_{ij} \tag{3.27}$$

To examine the power allocation term in (3.26), we rewrite the single summation as a double summation over all satellites $m \in \{1, \ldots, M\}$:

$$2\sum_{j} U_j \mathbb{E}\left\{\mu_j \left| \vec{U}(t_0)\right.\right\} = 2\sum_{m}\sum_{j\in Sat(M)} U_j \mathbb{E}\left\{\mu_j \left| \vec{U}(t_0)\right.\right\} \tag{3.28}$$

Thus, the given power allocation policy maximizes (3.28) over all allocation policies—including the stationary policy of allocating a power vector $\vec{P}^{\vec{S}}(t) = \left(P_1^{\vec{S}}(t), \ldots, P_J^{\vec{S}}(t)\right)$ whenever the channel is in state $\vec{S}$. Hence:

$$2\sum_{j} U_j \mathbb{E}\left\{\mu_j \left| \vec{U}(t_0)\right.\right\} \geq 2\sum_{j} U_j \sum_{\vec{S}} \pi_{\vec{S}} \mu_i \left(P_j^{\vec{S}}, S_j\right) \tag{3.29}$$

Using (3.27), (3.29), and (3.25) in (3.26), we find:

$$\mathbb{E}\left\{L(\vec{U}(t_0+1)) - L(\vec{U}(t_0)) \left| \vec{U}(t_0)\right.\right\} \leq A_{max}^2 + M(\mu_{max}^{out})^2 - 2\sum_{j} U_j(t_0)\epsilon \tag{3.30}$$

The above drift condition together with Lemma 2 proves stability. □

Corollary 4. *The average occupancy under the joint routing and power allocation algorithm satisfies the following performance bound (for i.i.d. channel states and arrivals):*

$$\overline{\sum_{j} \theta_j U_j} \leq \frac{A_{max}^2 + M(\mu_{max}^{out})^2}{2\epsilon}$$

where ϵ is the distance to the boundary of the capacity region (so that $\vec{\lambda} + \epsilon \in \Lambda$).

Proof. This follows immediately from (3.30) and Lemma 2. A similar bound can be derived for general channel and arrival processes using a K-step analysis. □

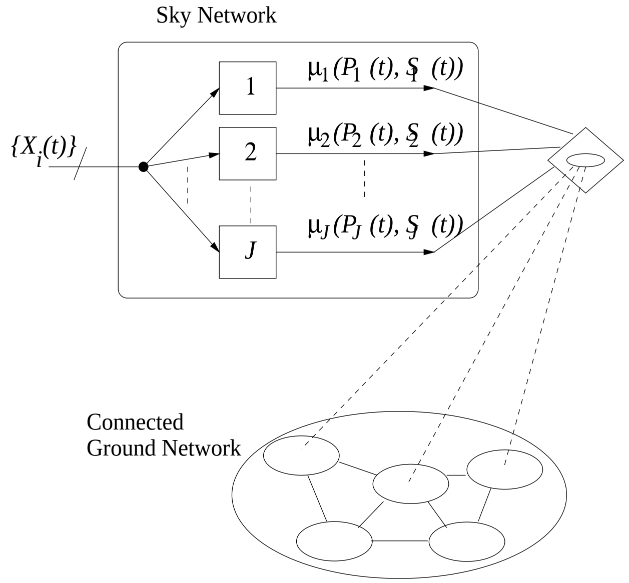


Figure 3-8: A joint routing and power allocation problem where the goal is to transmit the data to any node of the reliable ground network (i.e., $Q_i = \{1, \ldots, J\}$ for all input streams i).

An important special case of the above theorem is when inputs can route to the full set of available queues, i.e., $Q_i = \{1, \ldots, J\}$ for all inputs i. The goal is to simply transmit all the data to the ground as soon as possible. Such a situation arises when the ground units are connected together via a reliable ground network, and the wireless paths from satellite to ground form the rate bottleneck (See Fig. 3-8). In this case, it is shown in [110] (and Appendix B) that the capacity region of Theorem 3 simplifies to the simplex set of all input rates λ such that:

$$\lambda_1 + \cdots + \lambda_N \le \overline{\mu}_{out} \tag{3.31}$$

where

$$\overline{\mu}_{out} \overset{\triangle}{=} \sum_{\vec{S}} \pi_{\vec{S}} \sum_{m=1}^{M} \max_{\sum_{j\in Sat(m)} P_j \le P_{tot}^{(m)}} \left[\sum_{j\in Sat(m)} \mu_j(P_j, S_j) \right]$$

that is, $\overline{\mu}_{out}$ is the average output rate of the system when power is allocated to maximize the instantaneous processing rate at every instant of time.

In Fig. 3-9 we illustrate the capacity region for a 2-queue system with and without routing constraints. As expected, exploiting the full set of routing options considerably expands the capacity region of the system. Indeed, the simplex region (3.31) always contains the capacity region specified in Theorem 3 for joint routing and power allocation. This capacity gain is achieved by utilizing the extra resources offered by the ground network.

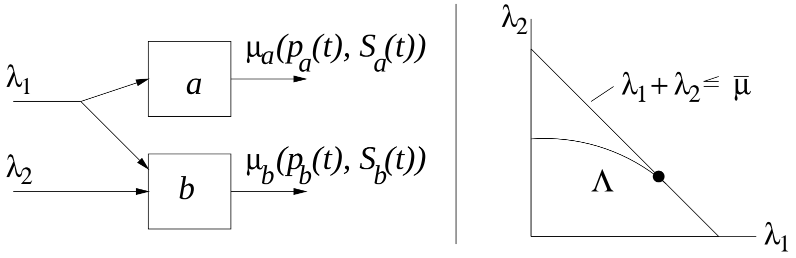


Figure 3-9: Capacity region for a two-queue system with routing and power control. The region Λ corresponds to the routing constraints shown in the figure, and is dominated by the simplex region for unconstrained routing.

We note that this joint routing and power allocation problem has been formulated for the case when data already contained within a single satellite or within a constellation of satellites is to be routed through a choice of downlinks. Hence, it is reasonable to assume the unfinished work values $U_i(t)$ are known to the controllers when making routing decisions. However, it can be shown (using an argument similar to that given in Lemma 7) that one can apply the same strategy when only *estimates* of the true unfinished work values are known. If estimates deviate from actual values by no more than an additive constant, the system will still be stable for all arrival rates within the stability region.

3.6 Connectivity Constraints

It has been assumed throughout that all transmit channels can be activated simultaneously, subject only to the total power constraint $\sum_{j\in Sat(m)} P_j(t) \leq P_{tot}^{(m)}$ for all time t. Hence, it is implicitly assumed that there is no interchannel interference. Such an assumption is valid when there is sufficient bandwidth to ensure potentially interfering channels can transmit using different frequency bands, or when ground users are sufficiently separated so that beamforming techniques can focus each downlink beam on its intended recipient without affecting other users. However, in bandwidth and space limited scenarios, power allocation vectors $\vec{P}(t)$ may be additionally restricted to *channel activation sets*: finite sets $\Pi_1, ..., \Pi_R$, where each set Π_r is a convex set of points $(P_1, ..., P_N)$ representing power vectors which, when allocated, ensure interchannel interference is at an acceptable level. This use of activation sets is similar to the treatment in [132], where activation link sets for scheduling ON/OFF links in a wireless network are considered. Here, the definition is extended from sets of links to sets of power vectors to treat power control.

As an example of an activation set, consider the single satellite system of Fig. 3-2 with N output queues, and suppose that downlink channels 1, 2, and 3 can be activated simultaneously if all other transmitters are silent. Such an activation set can be represented:

$$\Pi_r = \left\{ (P_1, P_2, P_3, 0, \ldots, 0) \in \mathbb{R}^N \;\middle|\; P_j \geq 0, \sum_{j=1}^{3} P_j \leq P_{tot} \right\}$$

Another type of system constraint is when power allocation is further restricted so that no more than K transmitters are active at any given time. Such a constraint corresponds to $\frac{N!}{K!(N-K)!}$ convex activation sets. Multi-satellite systems can also be treated using this activation set model. Indeed, the N output queues of Fig. 3-2 may be physically located in several different satellites. In the following, we assume that each activation set Π_r incorporates the power constraints $\sum_{j \in Sat(m)} P_j \leq P_{tot}^{(m)}$.

Consider the downlink system of Fig. 3-2. Packets arrive according to rate convergent processes with rates $(\lambda_1, \ldots, \lambda_N)$, and channel states $\vec{S}(t)$ vary according to a channel convergent process with steady state probabilities $\pi_{\vec{S}}$. Define the set Π as the union of all power activation sets: $\Pi = \Pi_1 \bigcup \Pi_2 \bigcup \cdots \bigcup \Pi_R$. Each timeslot a power allocation vector $\vec{P}(t)$ is chosen such that it lies within Π, that is, it lies within one of the acceptable activation sets $\{\Pi_1, \ldots, \Pi_R\}$.

Theorem 4. *(Power Allocation with Connectivity Constraints): For the multi-queue system of Fig. 3-2 with power constraints $\vec{P}(t) \in \Pi$:*

(a) The capacity region of the system is the set Λ of all arrival rate vectors λ such that:

$$\vec{\lambda} \in \Lambda \triangleq \sum_{\vec{S}} \pi_{\vec{S}} Convex\ Hull \left\{ \left(\mu_1(\vec{P}, \vec{S}), \ldots, \mu_N(\vec{P}, \vec{S}) \right) \middle| \vec{P} \in \Pi \right\} \tag{3.32}$$

where addition and scalar multiplication of sets has been used above.[5]

(b) The policy of allocating a power vector $\vec{P} = (P_1, ..., P_N)$ at each timestep to maximize the quantity $\sum_j \theta_j U_j(t) \mu_j(P_j, S_j(t))$ (subject to $\vec{P} \in \Pi$) stabilizes the system whenever the $\vec{\lambda}$ vector is in the interior of the capacity region.

We note that the allocation policy specified in part (b) of the theorem involves the non-

[5] For sets A, B and scalars α, β, the set $\alpha A + \beta B$ is defined as $\{\gamma \,|\, \gamma = \alpha a + \beta b$ for some $a \in A$, $b \in B\}$.

convex constraint $\vec{P} \in \Pi = \Pi_1 \bigcup \cdots \bigcup \Pi_R$. Maximizing the given metric over individual activation sets Π_r is a convex optimization problem, although a complete implementation of the given policy is non-trivial if the number of activation sets is large. However, the proof of parts (a) and (b) are simple extensions of the analysis presented in Theorems 1 and 2. For brevity, we omit the proof of (a) (this proof proceeds similarly to the proof of the necessary condition in Theorem 1, and is proven in more generality for the network problem in Chapter 4).

Proof. (Part (b)) For simplicity of exposition, we consider only the case when channel states and arrivals are i.i.d. from slot to slot (the more general case is treated in Chapter 4). Define the Lyapunov function $L(\vec{U}) = \sum_i \theta_i U_i^2$. The proof of Theorem 2 can literally be repeated up to (3.18):

$$\mathbb{E}\left\{L(\vec{U}(t_0+1)) - L(\vec{U}(t_0)) \middle| \vec{U}(t_0)\right\} \leq \theta_{max} B - 2\sum_i \theta_i U_i(t_0)\left[\mathbb{E}\left\{\mu_i \middle| \vec{U}(t_0)\right\} - \lambda_i\right] \quad (3.33)$$

From this point, negative drift of the Lyapunov function can be established by noting that the value of $\mathbb{E}\left\{\mu_i \middle| \vec{U}(t_0)\right\}$ maximizes $\sum_j \theta_j U_j \gamma_j$ over all vectors $\vec{\gamma}$ within the region Λ specified in (3.32). To see this, note that any $\vec{\gamma}$ in Λ can be written $\vec{\gamma} = (\gamma_1, \ldots, \gamma_N)$, where

$$\gamma_i = \sum_{\vec{S}} \pi_{\vec{S}} \sum_{r=1}^{R} \alpha_{\vec{S},\Pi_r} \mu_i\left(\vec{P}^{\vec{S},\Pi_r}, \vec{S}\right)$$

for some vectors $\vec{P}^{\vec{S},\Pi_r} \in \Pi_r$, and some scalar values $\alpha_{\vec{S},\Pi_r} \geq 0$ such that $\sum_{r=1}^{R} \alpha_{\vec{S},\Pi_r} = 1$ for all channel states $\vec{S}$. To see this, define $P_i^{*\vec{S}}$ as the power allocations from the dynamic scheme which optimizes $\sum_i \theta_i U_i(t) \mu_i(P_i, S_i(t))$ every timeslot. We thus have for any $\vec{\gamma}$ vector described above:

$$\begin{aligned}
\sum_i \theta_i U_i \gamma_i &= \sum_i \theta_i U_i \sum_{\vec{S}} \pi_{\vec{S}} \sum_{r=1}^{R} \alpha_{\vec{S},\Pi_r} \mu_i\left(P_i^{\vec{S},\Pi_r}, S_i\right) \\
&= \sum_{\vec{S}} \pi_{\vec{S}} \left[\sum_{r=1}^{R} \alpha_{\vec{S},\Pi_r} \sum_i \theta_i U_i \mu_i\left(P_i^{\vec{S},\Pi_r}, S_i\right)\right]
\end{aligned}$$

By definition of $(P_i^{*\vec{S}})$, we have $\sum_i \theta_i U_i \mu_i(P_i^{*\vec{S}}, S_i) \geq \sum_i \theta_i U_i \mu_i(P_i, S_i)$ for any other power allocations P_i, and hence:

$$\begin{aligned}\sum_i \theta_i U_i \gamma_i &\leq \sum_{\vec{S}} \pi_{\vec{S}} \left[\sum_{r=1}^{R} \alpha_{\vec{S},\Pi_r} \sum_i \theta_i U_i \mu_i (P_i^{*\vec{S}}, S_i) \right] \\ &= \sum_{\vec{S}} \pi_{\vec{S}} \left[\sum_i \theta_i U_i \mu_i (P_i^{*\vec{S}}, S_i) \right] \\ &= \sum_i \theta_i U_i \mathbb{E} \left\{ \mu_i \left| \vec{U} \right. \right\}\end{aligned}$$

Using this fact in (3.33) proves that the Lyapunov drift is less than or equal to $B\theta_{max} - 2\sum_i \theta_i U_i(t_0)\epsilon$, which by Lemma 2 proves the result. □

3.7 Numerical and Simulation Results for Satellite Channels

Here we present numerical and simulation results illustrating the capacity and delay performance provided by the dynamic power allocation policy of Section 3.4 (eq. (3.13)) for a simple satellite downlink consisting of two channels and two queues. We assume the corresponding input streams consist of unit length packets arriving as Poisson processes with rates (λ_1, λ_2). We consider a Markov modulated channel state that is typical of a satellite downlink [29] [28] [42] [51] and demonstrate the ability of the dynamic power allocation policy (3.13) to perform well under general time varying channel conditions.

3.7.1 Downlink Channel Model of the Ka Band

Experimental and modeling work for satellite downlinks is considered in [29] [28] [94] [1] [73] [42] [51]. Channel modeling experiments show that satellite channel states could be modeled as i.i.d. during clear weather conditions (due to the observed rapid fluctuation of signal attenuation from scintillations in the Ka band [29] [94] [28]). However, in rainy weather, future channel states are highly dependent on the current state. In [29] and [28] it is shown that channel state variations can be modeled as a Markov process.

We thus consider the following model. Each downlink channel is modulated by an independent Markov chain with three states corresponding to "Good," "Medium," and "Bad" channel conditions, with transition probabilities shown in Fig. 3-10. Such a three state system has been considered in [28] and extends the well known two-state Gilbert-Elliott model [42] [51] for satellite and wireless channels. In each state, we assume signal attenuation is log-normally distributed with a given mean and variance. Such a distribution is consistent with the Karasawa model [73] based on short term fading measurements in the Ka band.

Log-normal distribution of α_i for each of three channel conditions:

	mean	variance
Good	15 db	.264 db-squared
Medium	10 db	.868 db-squared
Bad	0 db	.145 db-squared

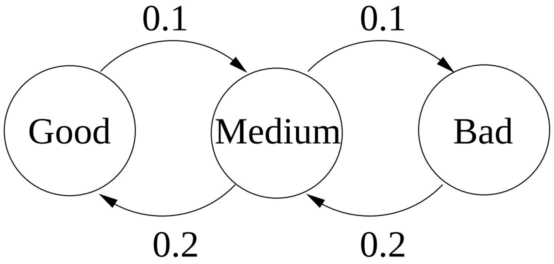


Figure 3-10: A three-state Markov chain representing Good, Medium, and Bad conditions for a single downlink from satellite to ground. In each state, an attenuation level α_i is chosen according to a log-normal distribution with means and variances as shown.

Total transmit power at the satellite is assumed to be 100 Watts. Factoring together the antenna gains, signal attenuation, and receiver noise, the average signal to noise ratio when full power is allocated to a single channel is assumed to be $15db$, $10db$, and $0db$ (for Good, Medium, and Bad conditions). The corresponding variances are $(.264, .868, .145)$db-squared, respectively. These values are based on measurement data in [28] for Ka band satellite channels under different conditions. We consider the Shannon capacity curves for data rate as a function of a normalized signal power:

$$\mu_i(P_i, S_i) = \log(1 + \alpha_i P_i)$$

where $\sum_i P_i \leq 1$ and α_i represents the fading coefficients, chosen according to the specified log-normal distributions with mean and variance determined by the channel state $S_i \in \{Good, Medium, Bad\}$. For the simulation, we discretize the log-normal distribution with 11 quantization levels. The two channels from satellite to ground are assumed to vary independently, each according to the described Markov modulated process.

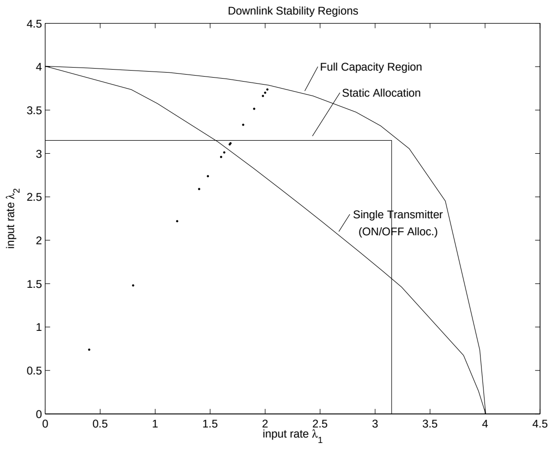


Figure 3-11: Stability regions for the three power allocation algorithms in the 2-queue downlink system. The isolated points represent simulated rate points (λ_1, λ_2) used in Fig. 3-12.

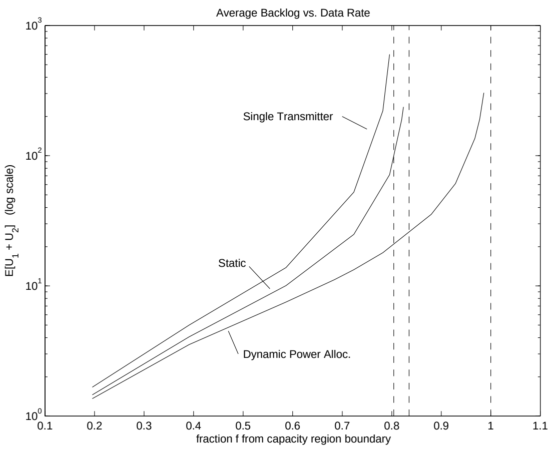


Figure 3-12: Average unfinished work $\mathbb{E}\{U_1 + U_2\}$ obtained from simulations of the three power allocation algorithms using the isolated rate points from Fig. 3-11. Data is plotted on a log scale.

3.7.2 Capacity and Delay Plots

In Fig. 3-11 we plot the downlink capacity region given by Theorem 1 (eq. (3.2)). Notice the non-linear "bulge" curvature, representing capacity gains due to dynamic power allocation. This full region is achievable using the dynamic power allocation algorithm of Theorem 2 (eq. (3.13), (3.21)). We compare the capacity region offered by this algorithm to the corresponding stability regions when power is allocated according to the following alternative strategies:

1) ON/OFF Power Allocation: Only one transmitter can be activated at any time.

2) Static Power Allocation: Constant Power $P_{tot}/2$ is allocated to each channel for all time.

The ON/OFF strategy allocates full power to the channel with the largest rate-backlog index $U_i(t)\mu_i(P_{tot}, S_i(t))$, which, by Theorem 4, achieves full capacity among all policies restricted to using a single transmitter. Notice that the stability region is slightly non-linear, because of the capacity boost due to the diversity offered by the independently time varying channels. The stability region for the static power allocation algorithm has a rectangular shape, as shown in Fig. 3-11. The capacity for this static algorithm is expanded beyond the stability region for the single transmitter algorithm when the input rates λ_1 and λ_2 are roughly within a factor of two of each other, although the single-transmitter algorithm is better for highly asymmetric data rates. Both policies are stable on a significantly reduced subset of the capacity region offered by the dynamic power allocation policy. Note that even in the completely symmetric case $\lambda_1 = \lambda_2$, the stability point of the static power allocation policy is slightly below the stability point of the dynamic power allocation policy, because the static policy cannot take advantage of the time varying channel conditions.

In addition, we simulate system dynamics for 2 million iterations using the three power allocation policies and a variety of data rates which linearly approach a boundary rate point $(\lambda_1, \lambda_2) = (2.05, 3.79)$ of the capacity region. The rates tested are shown in Fig. 3-11. In Fig. 3-12 we plot the empirical average occupancy $\mathbb{E}\{U_1 + U_2\}$ for the two queue system when the multi-beam dynamic power allocation algorithm is used, where power is allocated according to (3.21) (with weights $\theta_i = 1$ for all i). The plot illustrates that the dynamic power allocation policy achieves stability throughout the entire capacity region, with an average delay growing asymptotically as the input data rates approach the boundary point $(2.05, 3.79)$.

We compare the dynamic power allocation algorithm to the two other strategies, whose simulated performance is also shown in Fig. 3-12. From the figure, it is clear that the average occupancy (and hence, average delay) of the multi-beam dynamic power allocation algorithm is significantly lower than the corresponding averages for the other algorithms at all data rates (note that the asymptotes for instability occur earlier for the other two algorithms). For the rate regime tested, the stability region for the constant power allocation algorithm is slightly larger than the single-transmitter dynamic algorithm, and hence the corresponding average occupancies are lower. However, the static policy cannot adjust to asymmetries in data rate, and thus the single transmitter algorithm will perform better in the regime where one input rate is much larger than the other (see capacity plot in Fig. 3-11). The figures illustrate that to enable high data rates and low delay in a satellite downlink, it is essential to dynamically allocate power to the multiple beams.

3.8 Chapter Summary

We have treated data transmission over multiple time-varying channels in a satellite downlink using power control. Processing rates for each channel i were assumed to be determined by concave rate-power curves $\mu_i(P_i, S_i)$, and the capacity region of all stabilizable arrival rate vectors $\vec{\lambda}$ was established. This capacity region is valid for general rate-convergent input streams and channel convergent state processes (including Markovian modulated channel states). Inputs with arrival rates $\vec{\lambda}$ in the interior of the capacity region can be stabilized with a power allocation policy which only considers the current channel state $\vec{S}(t)$. In the case when arrival rates and channel probabilities $\vec{\lambda}$ and $\pi_{\vec{S}}$ are unknown, a stabilizing policy which considers both current channel state and current queue backlog was developed. Intuitively, the policy favors queues with large backlogs and better channels by allocating power to maximize $\sum_i U_i\mu_i$ at every timeslot. The policy reacts smoothly to channel state changes and arbitrary variations in the input rates. A real time implementation of the algorithm was described, and an analytical bound on average bit delay was established. This power control formulation was shown to contain the special case of a server allocation problem, and analysis verified stability and provided a performance bound for the *Serve-the-K-Longest-Connected-Queues* policy.

A joint routing and power allocation scenario was also considered for a system with multiple users and multiple satellites, and a throughput maximizing algorithm and a corresponding performance bound was developed. The structure of this algorithm allows for decoupled routing and power allocation decisions to be made by each user and each satellite based on local channel state and queue backlog information. In the case of interchannel interference, modified power allocation policies were developed when power vectors are constrained to a finite collection of activation sets. The policies offer 100% throughput, although are difficult to implement if the number of activation sets is large.

Stability properties of these algorithms hold for general rate convergent and channel convergent processes, and were established by demonstrating negative drift of a Lyapunov function defined over the current state of unfinished work in the queues. Robustness to arbitrary input rate changes was demonstrated by establishing an upper bound on time average queue occupancy in the case when the arrival rate vector $\vec{\lambda}_t$ is inside the capacity region for all timesteps t. Thus, the algorithms offer desirable performance under a wide variety of input processes and time varying channel conditions.

Our focus was power control for a satellite downlink, although the results extend to other wireless communication scenarios where power allocation and energy efficiency is a major issue. The use of dynamic power allocation can considerably extend the throughput and performance properties of such systems.

Chapter Appendix 3.A

Here we prove the inequalities:

$$\mathbb{E}\left\{\sum_i A_i^2 \left|\vec{U}(t_0)\right.\right\} \leq A_{max}^2 \qquad (\text{where } A_i \triangleq \tfrac{1}{K}\textstyle\sum_{\tau=t_0}^{t_0+K-1} A_i(\tau))$$
$$\mathbb{E}\left\{\sum_i \mu_i^2 \left|\vec{U}(t_0)\right.\right\} \leq (\mu_{max}^{out})^2 \qquad (\text{where } \mu_i \triangleq \tfrac{1}{K}\textstyle\sum_{\tau=t_0}^{t_0+K-1} \mu_i(\tau))$$

which are needed in the proof of Lemma 6.

Proof. We prove only the first inequality (the second is similar). Because $A_i \geq 0$ for all i, we have:

$$\begin{aligned}\sum_i A_i^2 &\leq \left(\sum_i A_i\right)^2 \\ &= \left(\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left(\sum_i A_i(\tau)\right)\right)^2 \\ &\leq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left(\sum_i A_i(\tau)\right)^2 \qquad (3.34)\end{aligned}$$

where (3.34) follows from Jensen's inequality and convexity of the function x^2. Taking conditional expectations yields:

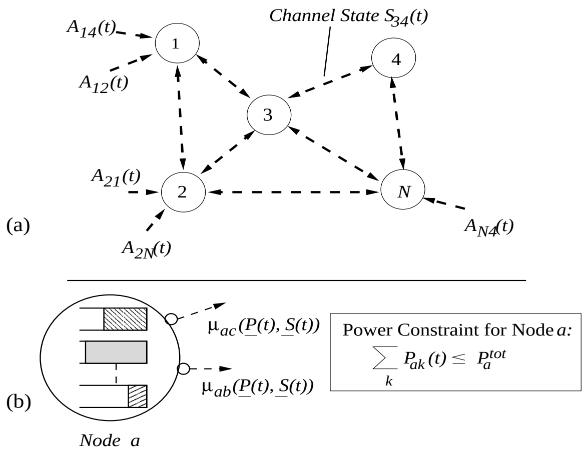
$$\begin{aligned}\mathbb{E}\left\{\sum_i A_i^2 \left|\vec{U}(t_0)\right.\right\} &\leq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mathbb{E}\left\{\left(\sum_i A_i(\tau)\right)^2 \left|\vec{U}(t_0)\right.\right\} \\ &\leq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} A_{max}^2 = A_{max}^2\end{aligned}$$

which proves the result. □

Chapter 4

Network Control

In this chapter we consider joint routing, scheduling, and power allocation in a multi-node, multi-hop network with time varying channels (Fig. 4-1). The network layer capacity region is established, and a dynamic control algorithm for achieving this capacity is constructed. We describe the network in terms of the general rate-power curves $\mu(\underline{P}, \underline{S})$ introduced in Chapter 1, which reflect the physical characteristics of each network element and the interchannel interference properties of each data link. Throughout this chapter we refer to the network as a *wireless network*. However, recall that such curves are general enough to describe satellite networks, switching systems, and hybrid networks with both wireless and wireline components, so that our analytical results can be applied to these systems as well.



$A_{14}(t)$
$A_{12}(t)$
Channel State $S_{34}(t)$
1
4
3
2
N
$A_{21}(t)$
$A_{N4}(t)$
$A_{2N}(t)$
(a)
$\mu_{ac}(\underline{P}(t), \underline{S}(t))$
$\mu_{ab}(\underline{P}(t), \underline{S}(t))$
Power Constraint for Node a:
$\sum_k P_{ak}(t) \leq P_a^{tot}$
(b)
Node a

Figure 4-1: (a) A wireless network with multiple input streams, and (b) a close-up of one node, illustrating the internal queues.

4.1 The General Network Model

Consider a multi-node wireless network characterized by the following properties:

- A channel convergent process $\underline{S}(t)$ with a finite number of channel states and state probabilities $\pi_{\underline{S}}$
- An upper semi-continuous rate-power function $\underline{\mu}\left(\underline{P}, \underline{S}\right)$
- A power constraint $\underline{P} \in \Pi$ for all t (where Π is a compact set of acceptable power allocations)

For convenience, we classify all data flowing through the network as belonging to a particular *commodity* $c \in \{1, \ldots, N\}$, representing the destination node for the data. Let $A_i^{(c)}(t)$ represent the amount of commodity c bits that arrive exogenously to the network at node i during slot t. We assume the $A_i^{(c)}(t)$ process is rate convergent with rates λ_{ij}.

4.1.1 System Parameters

We define the following system parameters which capture all of the features of the network needed to analyze stability and delay.

- **Transmission Rate Bounds** (μ_{max}^{out}) **and** (μ_{max}^{in})**:** Define:

$$\mu_{max}^{out} = \max_{\{i, \underline{S}, \underline{P} \in \Pi\}} \sum_b \mu_{ib}(\underline{P}, \underline{S})$$

$$\mu_{max}^{in} = \max_{\{i, \underline{S}, \underline{P} \in \Pi\}} \sum_a \mu_{ai}(\underline{P}, \underline{S})$$

  The bounds μ_{max}^{out} and μ_{max}^{in} place limits on the maximum transmission rate out of any node and into any node, respectively. Such bounds exist by compactness of the power allocation region Π [100] [15]. In practice, they represent physical limits on the rate at which a node can send or receive under the best channel conditions.

- **Arrival Bound** A_{max}^2**:** We assume that the second moment of exogenous arrivals to any node is bounded every timeslot by some finite maximum value A_{max}^2 regardless of past history, so that

$$\mathbb{E}\left\{\left[\sum_c A_i^{(c)}(t)\right]^2\right\} \leq A_{max}^2$$

for all t, and the same bound holds for conditional expectations given any past event. Note that $\mathbb{E}\left\{A\right\} \leq \sqrt{\mathbb{E}\left\{A^2\right\}}$ for any random variable A, so that the expected first moment of arrivals is bounded by A_{max}:

$$\mathbb{E}\left\{\sum_c A_i^{(c)}(t)\right\} \leq A_{max}$$

We emphasize that this bound A_{max}^2 places a limit on the second moment of arrivals during a timeslot. This allows for arrival processes such as Poisson streams, where the maximum number of arrivals is unbounded but the second moment is finite. In practice, a more stringent peak arrival bound would apply, representing the maximum rate at which the applications at each node can transfer data to the network layer.

- **The Convergence Interval K:** As in Chapter 2, we define $T_{\underline{S}}(t_0, K)$ as the set of timeslots at which $\underline{S}(t) = \underline{S}$ during the interval $t_0 \leq \tau \leq t_0 + K - 1$, and define $||T_{\underline{S}}(t_0, K)||$ as the total number of such slots. For given values $\delta > 0$, $\tilde{\delta} > 0$, we define the *convergence interval* K to be the smallest number of timeslots such that for any t_0, any (i, j), and regardless of past history, we have:

$$\left|\lambda_{ic} - \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{A_{ic}(\tau)\right\}\right| \leq \tilde{\delta} \tag{4.1}$$

$$\sum_{\underline{S}}\left|\frac{\mathbb{E}\left\{||T_{\underline{S}}(t_0, K)||\right\}}{K} - \pi_{\underline{S}}\right| \leq \frac{\delta}{\max\{\mu_{max}^{out}, \mu_{max}^{in}\}} \tag{4.2}$$

Such a value K exists because there are a finite number of arrival processes $A_{ij}(t)$, each of which is rate convergent, and the channel process $\underline{S}(t)$ is channel convergent.[1] This convergence interval represents the time period over which the network is expected to reach steady state, regardless of past history. All of the time varying properties of the network that we use to analyze stability and delay (such as the mobility dynamics, fading distributions, or link outage statistics) are captured in this scalar value K. We note that in systems with i.i.d. arrivals and channel states, steady state is exactly achieved every timeslot, so that $K = 1$ even when both δ and $\tilde{\delta}$ are set to 0 in (4.1)

[1] The right hand side of the bound in (4.2) has the form $\delta/\max\{\mu_{max}^{out}, \mu_{max}^{in}\}$ so that transmission rates of a stationary power allocation policy are rate convergent with K-slot averages that are within δ of the long term input or output rate, as described in Lemma 8.

and (4.2) above.

Below we develop a *stochastic network calculus*[2] for analyzing stability and delay of capacity achieving control schemes in the network layer using these parameters μ_{max}^{out}, μ_{max}^{in}, A_{max}^2, and K.

4.1.2 The Queueing Equation

Each network node i maintains a set of output queues for storing data according to its destination. Let $U_i^{(c)}(t)$ represent the current backlog of bits in node i destined for node c. The $U_i^{(c)}(t)$ processes evolve according to the following queueing dynamics:

$$U_i^{(c)}(t+1) = \max\left[U_i^{(c)}(t) - D_i^{(c)}(t), 0\right] + E_i^{(c)}(t) + A_i^{(c)}(t)$$

where

$$\begin{aligned} D_i^{(c)}(t) &\triangleq \text{ amount of commodity } c \text{ bits transmitted out of node } i \text{ during slot } t \\ E_i^{(c)}(t) &\triangleq \text{ amount of commodity } c \text{ bits that endogenously arrive to node } i \text{ during slot } t \end{aligned}$$

The quantities $D_i^{(c)}(t)$ and $E_i^{(c)}(t)$ are determined by the routing, scheduling, and power allocation decisions made by the network control algorithm.

4.2 The Network Capacity Region

Definition 6. *The* capacity region Λ *is the closed region of* $N \times N$ *rate matrices* (λ_{ic}) *with the following properties:*

- $(\lambda_{ic}) \in \Lambda$ *is a necessary condition for network stability, where all possible ergodic or non-ergodic stabilizing power control and routing algorithms are considered (including algorithms which have full knowledge of future events).*

- (λ_{ic}) strictly interior *to* Λ *is a sufficient condition for the network to be stabilized by some routing and power allocation policy.*

[2] A non-stochastic network calculus was invented in [35], [36] for static networks with leaky bucket inputs and fixed routing.

Remarkably, we show that a stabilizing policy can be developed which does not require knowledge of future events, and hence such knowledge does not expand the region of stabilizable rates. Below we describe the set of rate matrices Λ making up this region, and in Theorem 5 we show this set Λ is the true capacity region by establishing both the necessary and sufficient conditions listed above.

To build intuition, we first consider the capacity region of a traditional wireline network with no time variation, defined on a weighted graph with N nodes, E edges, and node-to-node link capacities given by a link matrix (G_{ab}). The link matrix describes the rate at which node a can deliver data to node b (for all (a, b) node pairs), so that $G_{ab} = 0$ if there is no directed edge from node a to node b, and is equal to the positive transmission rate for that link otherwise. To avoid confusion, we note that there are two sets of rate matrices defined here: the exogenous arrival rate matrix (λ_{ic}) and the node-to-node link transmission rate matrix (or "link capacity" matrix) (G_{ab}). The network capacity region is described implicitly as the set of all arrival rate matrices (λ_{ic}) such that there exist *multi-commodity flow variables* $f_{ab}^{(c)}$ (for $a, b, c \in \{1, \ldots, N\}$) which satisfy the non-negativity, flow efficiency, and flow conservation constraints specified in (4.4)-(4.6), and which additionally satisfy the link constraint $\sum_c f_{ab}^{(c)} \leq G_{ab}$ for all links (a, b). We describe each of these constraints below.

The non-negativity constraint (4.4) ensures that all flow variables $f_{ab}^{(c)}$ are non-negative. The flow efficiency constraints (4.5) imply that data is never transfered from a node to itself, and data is never retransmitted once it has reached its destination. The flow conservation constraint (4.6) is most easily understood when equality holds, which implies that the net influx of commodity c bits is zero at intermediate nodes $i \neq c$. The constraint is relaxed to an inequality because smaller input rates λ_{ic} can also be supported. Finally, the link constraint for a *traditional wireline network* is that $\sum_c f_{ab}^{(c)} \leq G_{ab}$ for all links (a, b), ensuring that the total rate of flow over any link does not exceed the capacity of that link. Together, these constraints indicate that the multi-commodity flow variables $\{f_{ab}^{(c)}\}$ represent a *feasible routing* for all commodities c. Such multi-commodity flows must exist in order for the network to be stable, regardless of whether bits flow as a continuous fluid or bits arrive and are transmitted in packetized form every timeslot.

The major difference between the capacity region of a wireless network and the capacity region of the traditional wireline network comes in the link constraint. First note that, due to the time varying channels, the node-to-node link capacity G_{ab} for any wireless link (a,b) must be defined in a time average sense, where the resulting transmission rate is averaged over all possible channel states. Second, the resulting time average node-to-node rates (G_{ab}) are not fixed, but depend on the power allocation policy. A particular power allocation policy 1 gives rise to a particular time average rate matrix $\underline{G}^{(1)} = (G_{ab}^{(1)})$, while another policy 2 might give rise to another rate matrix $\underline{G}^{(2)} = (G_{ab}^{(2)})$. Thus, instead of describing the network as a single weighted graph (G_{ab}) of link rates, the network is described by a *collection* of graphs, or a *graph family* Γ. We define the graph family Γ as the following set of node-to-node transmission rate matrices:

$$\Gamma = \sum_{\underline{S}} \pi_{\underline{S}} \text{Convex\_Hull} \left\{ \underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi \right\} \tag{4.3}$$

where addition and scalar multiplication of sets is used[3], and the convex hull of a set A is defined as the set of all convex combinations $p_1 a_1 + p_2 a_2 + \ldots + p_k a_k$ of elements $a_i \in A$ (where $\{p_i\}$ are probabilities summing to 1).

Thus, a transmission rate matrix $\underline{G} = (G_{ab})$ is in graph family Γ if and only if $\underline{G}$ can be represented as $\underline{G} = \sum_{\underline{S}} \pi_{\underline{S}} \underline{G}_{\underline{S}}$ for some set of matrices $\underline{G}_{\underline{S}}$, each one being inside the convex hull of the set of node-to-node transmission rates achievable by power allocation under channel state $\underline{S}$ (see Fig. 4-2).

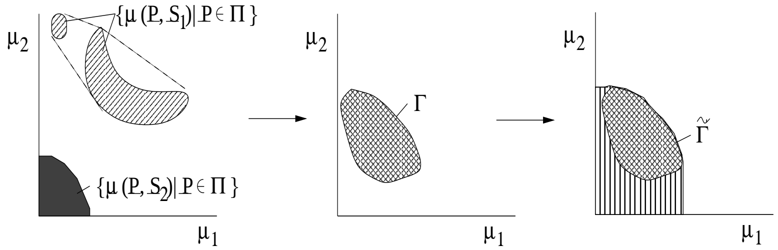

$\{\mu(\underline{P}, \underline{S}_1) | \underline{P} \in \Pi\}$
μ_2
$\{\mu(\underline{P}, \underline{S}_2) | \underline{P} \in \Pi\}$
μ_1
μ_2
Γ
μ_1
μ_2
$\tilde{\Gamma}$
μ_1

Figure 4-2: A construction of the set Γ for the case of 2 dimensions, illustrating the set of all achievable long term link rates (μ_1, μ_2). In this example, we consider only two channel states $\underline{S}_1$ and $\underline{S}_2$, each equally probable. Note that for the first channel state, the set $\left\{\underline{\mu}(\underline{P}, \underline{S}_1) \mid \underline{P} \in \Pi\right\}$ is disconnected and non-convex. Its convex hull is shown in the first plot. The second plot illustrates the weighted sum of the convex hull of the regions associated with each of the two channel states. This is the Γ region, and is necessarily convex. It can be shown that if the $\underline{\mu}(\underline{P}, \underline{S})$ function is upper semi-continuous in the power matrix $\underline{P}$, then the extended set $\tilde{\Gamma}$, formed by considering all rate matrices entrywise less than or equal to some element of Γ, is both convex and closed.

[3] For sets A, B and scalars α, β, the set $\alpha A + \beta B$ is defined as $\{\gamma \mid \gamma = \alpha a + \beta b \text{ for some } a \in A, b \in B\}$.

In the proof of Theorem 5, it is shown that graph family Γ can be viewed as the set of all long-term transmission rates (G_{ab}) that the network can be configured to support on the single-hop wireless links connecting node pairs (a, b). It is useful to define the extended graph family $\tilde{\Gamma}$ as the set of all rate matrices entrywise less than or equal to a matrix in Γ (see Fig. 4-2), as traffic on or below a point in Γ can likewise be supported.

Network Capacity Region: The capacity region Λ is the set of all input rate matrices (λ_{ic}) such that there exist multi-commodity flow variables $\{f_{ab}^{(c)}\}$ satisfying:

$$f_{ab}^{(c)} \geq 0 \qquad \forall a, b, c \qquad \text{(Non-negativity)} \tag{4.4}$$

$$f_{aa}^{(c)} = f_{ab}^{(a)} = 0 \qquad \forall a, b, c \qquad \text{(Flow-efficiency)} \tag{4.5}$$

$$\lambda_{ic} \leq \sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} \quad \forall i, c \text{ such that } i \neq c \quad \text{(Flow conservation)} \tag{4.6}$$

$$\left(\sum_c f_{ab}^{(c)} \right) \leq (G_{ab}) \qquad \text{for some } (G_{ab}) \in \Gamma \quad \text{(Link constraint)} \tag{4.7}$$

where the matrix inequality in (4.7) is considered entrywise. Note that the link constraint (4.7) can equivalently be written $\left(\sum_c f_{ab}^{(c)} \right) \in \tilde{\Gamma}$.

Thus, a rate matrix (λ_{ic}) is in the capacity region Λ if there exists a matrix $(G_{ab}) \in \Gamma$ that defines link capacities in a traditional graph network, such that there exist multi-commodity flow variables $\{f_{ab}^{(c)}\}$ which support the λ_{ic} rates with respect to this graph.

It can be shown using standard convex analysis techniques [15] that the set Γ is convex, and that sets $\tilde{\Gamma}$ and Λ are compact and convex (see Chapter Appendix 4.C). Such structural properties are used in the proof of the following theorem.

Theorem 5. *(Capacity Region for a Wireless Network)*

(a) A necessary condition for stability is $(\lambda_{ic}) \in \Lambda$.

(b) A sufficient condition for stability is that (λ_{ic}) *is* strictly interior *to* Λ.

Proof. The proof of (a) is given in Chapter Appendix 4.A, where it is shown that no control algorithm can achieve stability beyond the set Λ, even if the entire set of future events is known in advance. Part (b) can be shown constructively by routing data according to the flow variables $\{f_{ab}^{(c)}\}$ and allocating power to meet the long-term link capacity requirements (G_{ab}) (where the $f_{ab}^{(c)}$ and G_{ab} values correspond to the input rate matrix (λ_{ic}) via (4.4)-(4.7)). Such a construction together with a bound on average delay is developed in the next section. □

We note that the time varying channel conditions influence the network capacity region only through their steady state probabilities $\pi_{\underline{S}}$. Cross correlations and higher order channel statistics effect network delay (as described in the next section) but do not change network capacity. We thus have the following corollary.

Corollary 5. *(Capacity Region Invariance) A wireless network with a general channel convergent process $\underline{S}(t)$ with time average state probabilities $\pi_{\underline{S}}$ has the same capacity region as an identical network in which channel states are chosen i.i.d. every timeslot according to the $\pi_{\underline{S}}$ distribution.* □

4.3 Stabilizing Control Policies

The capacity region Λ of a wireless network is described in terms of flow variables $f_{ab}^{(c)}$ and a link matrix $(G_{ab}) \in \Gamma$ which satisfy (4.4)-(4.7). In principle, these values can be computed if the arrival rates (λ_{ic}) and channel probabilities $\pi_{\underline{S}}$ are known in advance. This allows us to view power allocation and routing in a decoupled manner, where data is routed according to flow variables $f_{ab}^{(c)}$, and power is allocated to achieve long-term link capacities (G_{ab}). Here we construct such a policy and show it provides a bound on average delay. We then use this analysis to construct a more practical and robust strategy that offers similar performance without requiring knowledge of the input and channel statistics.

4.3.1 Control Decision Variables

A network control algorithm makes decisions about power allocation, routing, and scheduling. As a general algorithm might schedule multiple commodities to flow over the same link on a given timeslot, we define $\mu_{ab}^{(c)}(t)$ as the rate offered to commodity c traffic along link (a,b) during timeslot t. Note that this provides the means of modeling dynamic routing decisions, as commodity c data in a given node can be routed to any of the outgoing links, as determined by the $\mu_{ab}^{(c)}(t)$ rates chosen by the control algorithm. Specifically, a network control algorithm must make the following decisions:

Power Allocation: Choose $\underline{P}(t)$ such that $\underline{P}(t) \in \Pi$.

Routing/Scheduling: Choose $\mu_{ab}^{(c)}(t)$ such that:

$$\sum_c \mu_{ab}^{(c)}(t) \leq \mu_{ab}(t) \triangleq \mu_{ab}(\underline{P}(t), \underline{S}(t))$$

Note that in the special case where there is no power allocation, the $\mu_{ab}(t)$ process is purely determined by the dynamic channel states of the network, and any network control algorithm reduces to pure routing and scheduling.

4.3.2 Stability for Known Arrival and Channel Statistics

To construct a stabilizing policy, we first show that power can be allocated to achieve any long-term link transmission rate matrix (G_{ab}) within the network graph family Γ.

Lemma 8. *(Graph Family Achievability) Let (G_{ab}) be a matrix within the graph family Γ (defined in (4.3)), so that*

$$\sum_{\underline{S}} \pi_{\underline{S}} \underline{G_{\underline{S}}} = (G_{ab}) \tag{4.8}$$

for some matrices $\underline{G_{\underline{S}}}$ within Convex_Hull$\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$. Then:
(a) A stationary randomized *power allocation policy $\underline{P}^{STAT}(\tau)$ can be implemented which yields a transmission rate process $\underline{\mu}^{STAT}(t) \triangleq \underline{\mu}(\underline{P}^{STAT}(t), \underline{S}(t))$ which is entrywise rate convergent with rate matrix (G_{ab}). That is, for all links (a,b), we have $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t} \mu_{ab}^{STAT}(t) = G_{ab}$. Furthermore, for all nodes i and for any time t_0:*

$$\left| \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \left[\sum_b \mathbb{E}\left\{\mu_{ib}(\tau)\right\} \right] - \sum_b G_{ib} \right| \leq \delta \tag{4.9}$$

$$\left| \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \left[\sum_a \mathbb{E}\left\{\mu_{ai}(\tau)\right\} \right] - \sum_a G_{ai} \right| \leq \delta \tag{4.10}$$

where the K and δ parameters are defined in (4.1) and (4.2).

The structure of the policy is as follows: Every timeslot in which the channel state $\underline{S}$ is observed, the power matrix $\underline{P}^{STAT}(\tau)$ is chosen randomly from a finite set of m allocations $\{\underline{P}_{\underline{S}}^1, \ldots, \underline{P}_{\underline{S}}^m\}$ according to a set of probabilities $\{q_{\underline{S}}^1, \ldots, q_{\underline{S}}^m\}$.

(b) If the set $\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ is convex for every channel state $\underline{S}$, then a power control algorithm yielding a rate convergent transmission rate process $\underline{\mu}^{STAT}(t)$ with rates (G_{ab}) and satisfying (4.9) and (4.10) can be implemented by a non-randomized policy, where a fixed power matrix $\underline{P}_{\underline{S}}$ is allocated whenever in channel state $\underline{S}$.

Proof. The proof follows by expressing each $\underline{G_{\underline{S}}}$ matrix as a convex combination of matrices in $\{\underline{\mu}(\underline{P}, \underline{S}) \,|\, \underline{P} \in \Pi\}$ according to Caratheodory's Theorem [15], and defining the probabilities of the stationary randomized scheme according to the weights of the convex combination. A full proof is given in Chapter Appendix 4.D. □

Note that in the policy of Lemma 8, power allocations are only changed on timeslot boundaries and hence it is not necessary to vary power during the course of a single slot to achieve rates in the graph family Γ. The given strategy bases decisions only on the current channel state and does not depend on queue backlogs. However, the policy is rather idealized: The existential nature of Lemma 8 does not provide any practical means of computing the power values and probabilities needed to implement the policy. However, this allocation policy is analyzable using a well developed theory of Lyapunov drift [6] [98] [81] [132] [88] [95].[4] Below we develop a control strategy based on this idealized policy. The strategy is not offered as a practical means of network control, but as a baseline by which our dynamic algorithm of Section 4.3.3 can be compared.

Assume the channel probabilities $\pi_{\underline{S}}$ are known, and that the exogenous arrival rates (λ_{ic}) are known and are *strictly interior* to the capacity region Λ, so that there is a positive value ϵ that can be added to each component of (λ_{ic}) such that $(\lambda_{ic} + \epsilon) \in \Lambda$. Let (G_{ab}) and $\{f_{ab}^{(c)}\}$ represent the network graph and multi-commodity flow variables, respectively, associated with rates $(\lambda_{ic} + \epsilon)$ and satisfying (4.4)-(4.7). In particular:

$$(\lambda_{ic} + \epsilon) \leq \textstyle\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} \quad \text{for } i \neq c \tag{4.11}$$

$$\left(\textstyle\sum_c f_{ab}^{(c)}\right) \leq (G_{ab}) \tag{4.12}$$

The values $f_{ab}^{(c)}$ and G_{ab} could in principle be computed with knowledge of $\pi_{\underline{S}}$ and (λ_{ic}), and we assume in this subsection that they are known to the network controller.

Stationary Randomized Policy (STAT) for Known System Statistics:

Power Allocation: Every timeslot, observe the channel state $\underline{S}$ and allocate power according to the stationary algorithm $\underline{P}^{STAT}(t)$ of Lemma 8, yielding the rate convergent process $\mu_{ab}^{STAT}(t)$ on each link (a, b) with a long-term link capacity matrix (G_{ab}).

Scheduling/Routing: For every link (a, b) such that $\sum_c f_{ab}^{(c)} > 0$, transmit the single

[4] In particular, we use the drift result developed in Chapter 2, which both simplifies and generalizes the known theory while enabling analysis of our general stochastic wireless network.

commodity $\hat{c}_{ab}$, where $\hat{c}_{ab}$ is chosen randomly with probability $f_{ab}^{(c)}/\sum_c f_{ab}^{(c)}$. However, use only a fraction $\frac{\sum_c f_{ab}^{(c)}}{G_{ab}}$ of the instantaneous link rate, so that:

$$\mu_{ab}^{(c)STAT}(t) = \begin{cases} \mu_{ab}^{STAT}(t)\frac{f_{ab}^{(c)}}{G_{ab}} & \text{if } c = \hat{c}_{ab} \\ 0 & \text{otherwise} \end{cases}$$

If a node does not have enough (or any) bits of a certain commodity to send over its output links, *null bits* are delivered, so that links have idle times which are not used by other commodities.

Note that:

$$\mathbb{E}\left\{\mu_{ab}^{(c)STAT}(t) \mid \mu_{ab}^{STAT}(t)\right\} = \mu_{ab}^{STAT}(t)\frac{f_{ab}^{(c)}}{G_{ab}} \tag{4.13}$$

By Lemma 8, we know $\mu_{ab}^{STAT}(t)$ is rate convergent with rate G_{ab}, and hence $\mu_{ab}^{(c)STAT}(t)$ is rate convergent with rate $f_{ab}^{(c)}$.

Fix $\delta \triangleq \tilde{\delta} \triangleq \epsilon/6$, which defines the convergence interval K according to (4.1) and (4.2). The bounds of (4.1), (4.9), and (4.10) thus become:

$$\left|\lambda_{ic} - \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mathbb{E}\left\{A_{ic}(\tau)\right\}\right| \leq \frac{\epsilon}{6} \tag{4.14}$$

$$\left|\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left[\sum_b \mathbb{E}\left\{\mu_{ib}^{(c)STAT}(t)\right\}\right] - \sum_b f_{ib}^{(c)}\right| \leq \frac{\epsilon}{6}\max_{a,b}\left\{\frac{f_{ab}}{G_{ab}}\right\} \leq \frac{\epsilon}{6} \tag{4.15}$$

$$\left|\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left[\sum_a \mathbb{E}\left\{\mu_{ai}^{(c)STAT}(t)\right\}\right] - \sum_a f_{ai}^{(c)}\right| \leq \frac{\epsilon}{6}\max_{a,b}\left\{\frac{f_{ab}}{G_{ab}}\right\} \leq \frac{\epsilon}{6} \tag{4.16}$$

where (4.15) and (4.16) follow by multiplying (4.9), and (4.10) by $f_{ab}^{(c)}/G_{ab}$ and using (4.13).

Inequalities (4.14)-(4.16) state that the K-slot time averages for the exogenous arrival rates, endogenous arrival rates, and transmission rates for each node are within $\epsilon/6$ of their limiting values. Hence, the difference between the transmission rates and the sum of the endogenous and exogenous arrival rates must be within $\epsilon/2$ of the limiting difference. By (4.11), this limiting difference is given by:

$$\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} - \lambda_{ic} \geq \epsilon \quad \text{for } i \neq c \tag{4.17}$$

Hence, we have:[5]

$$\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mathbb{E}\left\{\sum_b \mu_{ib}^{(c)STAT}(\tau) - \sum_a \mu_{ai}^{(c)STAT}(\tau) - A_i^{(c)}(\tau)\right\} \geq \frac{\epsilon}{2} \tag{4.18}$$

The above inequality (4.18) holds at any time t_0, regardless of the condition of the arrival or server process at this time. The convergence interval K was defined for $\delta = \epsilon/6$ according to (4.1) (4.2) in order to establish this inequality. In cases where a smaller value K' can be found for which inequality (4.18) holds, the smaller value can be used in place of K in all of the following analysis. We note that when arrivals and channel states are i.i.d. every timeslot, then $K = 1$, $\delta = \tilde{\delta} = 0$, and the right hand side of the above inequality can be increased from $\epsilon/2$ to ϵ, as steady state averages are achieved exactly on every timeslot (and hence the left hand side of (4.18) reduces to the left hand side of (4.17)).

Theorem 6. *(Stabilizing Policy for Known Statistics) Consider an N node wireless network as described above, with capacity region Λ and input rates (λ_{ic}) such that $(\lambda_{ic}+\epsilon) \in \Lambda$ for some $\epsilon > 0$. Then, jointly routing and allocating power according to the above stationary randomized policy STAT stabilizes the system and guarantees bounded average bit occupancies satisfying:*

$$\overline{\sum_{i,c} U_i^{(c)STAT}} \leq \frac{KBN}{\epsilon} \tag{4.19}$$

where

$$B \triangleq (A_{max} + \mu_{max}^{in})^2 + (\mu_{max}^{out})^2 \tag{4.20}$$

and the overbar notation on the left side of (4.19) is defined

$$\overline{\sum_{i,c} U_i^{(c)}} \triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{i,c}\mathbb{E}\left\{U_i^{(c)}(\tau)\right\}\right]$$

Proof. The K-step dynamics of unfinished work satisfies the following bound for all $i \neq c$:

$$U_i^{(c)}(t_0+K) \leq \max\left(U_i^{(c)}(t_0) - \sum_{\tau=t_0}^{t_0+K-1}\sum_b \mu_{ib}^{(c)}(\tau), 0\right) + \sum_{\tau=t_0}^{t_0+K-1}\sum_a \mu_{ai}^{(c)}(\tau) + \sum_{\tau=t_0}^{t_0+K-1} A_{ic}(\tau) \tag{4.21}$$

[5] Inequality (4.18) follows easily by noting that for any variables $x_1, x_2, x_3, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ satisfying $x_1 - x_2 - x_3 \geq \epsilon$ and $|x_i - \tilde{x}_i| \leq \epsilon/6$ for $i = 1, 2, 3$, then $\tilde{x}_1 - \tilde{x}_2 - \tilde{x}_3 \geq \epsilon/2$.

where (4.21) holds as an inequality instead of an equality because the total bits arriving to node i from other nodes of the network may be less than $\sum_{\tau=t_0}^{t_0+K-1}\sum_a \mu_{ai}^{(c)}(\tau)$ if these other nodes have little or no data to send, and because some arrivals during the K slot interval may also depart in the same interval.

Now define the Lyapunov function $L(\underline{U}) = \sum_{i\neq c}[U_i^{(c)}]^2$, where the notation "$\sum_{i\neq c}$" represents the double summation "$\sum_{i=1}^{N}\sum_{c\in\{1,\ldots,N\}-\{i\}}$". By squaring both sides of (4.21), taking conditional expectations, and performing simple manipulations, we have the following expression for the K-slot Lyapunov drift [see Chapter Appendix 4.E for the detailed manipulations]:

$$\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0))\,|\,\underline{U}(t_0)\right\} \leq K^2BN+ \\ -2K\sum_{i\neq c}U_i^{(c)}(t_0)\tfrac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mathbb{E}\left\{\sum_b \mu_{ib}^{(c)}(\tau) - \sum_a \mu_{ai}^{(c)}(\tau) - A_{ic}(\tau)\,|\,\underline{U}(t_0)\right\} \quad (4.22)$$

where B is defined in (4.20). The above expression for Lyapunov drift holds for any control policy that chooses general $\mu_{ij}^{(c)}(t)$ rates. Implementing the stationary randomized control policy yields rates $\mu_{ij}^{(c)STAT}(t)$. Directly applying the condition (4.18) in the expectation of (4.22) yields:

$$\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0))\,|\,\underline{U}(t_0)\right\} \leq K^2BN - 2K\sum_{i\neq c}U_i^{(c)}(t_0)[\epsilon/2]$$

Applying the Lyapunov Drift Lemma (Lemma 2) to the above inequality proves the result. □

In the case of i.i.d. arrivals and channel states, $K=1$ and the $\epsilon/2$ value in (4.18) can be replaced by ϵ, yielding a performance guarantee of $\overline{\sum_{i,c}U_i^{(c)STAT}} \leq \frac{BN}{2\epsilon}$.

4.3.3 A Dynamic Policy for Unknown System Statistics

The stabilizing policy of the above section requires full knowledge of arrival rates and channel state probabilities, along with the associated multi-commodity flows and the randomized power allocations. Here we present a dynamic power control and routing scheme which requires no knowledge of the arrival rates or channel statistics, yet guarantees performance similar to the previous policy which does use this information. This surprising result arises because the dynamic policy considers both the channel state $\underline{S}(t)$ and the system backlogs

$\underline{U}(t)$ when making control decisions. The policy is inspired by the maximum differential backlog algorithms developed by Tassiulas in [132] for stable server scheduling in a multi-hop radio network and an $N \times N$ packet switch, and generalizes the Tassiulas algorithm by considering the power allocation problem and treating a network with general interference and time varying channel characteristics. Furthermore, we obtain a simple expression for average end-to-end network delay by relating network performance of our dynamic scheme to the performance of the stationary control policy STAT developed in the previous subsection.

Every timeslot the network controller observes the channel state $\underline{S}(t)$ and the matrix of queue backlogs $\underline{U}(t) = (U_i^{(c)}(t))$ and performs routing and power control as follows.

Dynamic Routing and Power Control (DRPC) Policy:

1. For all links (a, b), find commodity $c_{ab}^*(t)$ such that:

$$c_{ab}^*(t) = \arg \max_{c \in \{1,\ldots,N\}} \left\{ U_a^{(c)}(t) - U_b^{(c)}(t) \right\}$$

and define:

$$W_{ab}^*(t) = \max[U_a^{(c_{ab}^*(t))}(t) - U_b^{(c_{ab}^*(t))}(t), 0] \tag{4.23}$$

2. *Power Allocation:* Choose a matrix $\underline{P}(t)$ such that:

$$\underline{P}(t) = \arg \max_{\underline{P} \in \Pi} \sum_{a,b} \mu_{ab}(\underline{P}, \underline{S}(t)) W_{ab}^* \tag{4.24}$$

3. *Routing:* Over link (a, b), send an amount of bits from commodity c_{ab}^* according to the rate offered by the power allocation. If any node does not have enough bits of a particular commodity to send over all its outgoing links requesting that commodity, *null bits* are delivered.

Thus, the corresponding $\mu_{ab}^{(c)DRPC}(t)$ values for this algorithm are given by:

$$\mu_{ab}^{(c)DRPC}(t) = \begin{cases} \mu_{ab}\left(\underline{P}^{DRPC}(t), \underline{S}(t)\right) & \text{if } c = c_{ab}^* \text{ and } W_{ab}^* > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.25}$$

Note that the W_{ab}^* values represent the *maximum differential backlog* of commodity c bits between nodes a and b. The policy thus uses backpressure to find an optimal routing.

We emphasize that this scheme does not use any pre-specified set of routes. The route for each unit of data is found dynamically according to the maximum differential backlog policy.

Theorem 7. *(Stabilizing Policy for Unknown System Statistics) Suppose an N-node wireless network has capacity region Λ and rate matrix (λ_{ic}) such that $(\lambda_{ic} + \epsilon) \in \Lambda$ for some $\epsilon > 0$, although these rates and the channel probabilities $\pi_{\underline{S}}$ are unknown to the network controller. Then, jointly routing and allocating power according to the above DRPC policy stabilizes the system and guarantees bounded average bit occupancies satisfying:*

$$\overline{\sum_{i,c} U_i^{(c)DRPC}} \leq \frac{KBN}{\epsilon} + \frac{(K-1)N\tilde{B}}{\epsilon} \tag{4.26}$$

where B is defined in (4.20), and

$$\tilde{B} \triangleq 2(\mu_{max}^{in} + \mu_{max}^{out})\left(A_{max} + \mu_{max}^{in} + \mu_{max}^{out}\right) \tag{4.27}$$

We prove this theorem through a sequence of two lemmas. The first lemma compares the Lyapunov drift of the STAT algorithm (which is known to be stable by Theorem 6), to the drift of a modified DRPC algorithm we call FRAME. The second lemma compares the drift of FRAME to that of DRPC.

For the Lyapunov function $L(\underline{U}) = \sum_{i \neq c} [U_i^{(c)}]^2$, a general bound on the K-step drift of any control strategy is given in (4.22) [see Theorem 6 and Chapter Appendix 4.E]. Below we rewrite the drift bound (4.22) in terms of a quantity $\Phi\left(\underline{U}(t_0)\right)$, which captures the only component of the bound that depends on the control strategy:

$$\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0)) \mid \underline{U}(t_0)\right\} \leq K^2BN - 2K\left[\Phi\left(\underline{U}(t_0)\right) - \beta\left(\underline{U}(t_0)\right)\right] \tag{4.28}$$

where

$$\Phi\left(\underline{U}(t_0)\right) \triangleq \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{\sum_{i \neq c} U_i^{(c)}(t_0)\left[\sum_b \mu_{ib}^{(c)}(\tau) - \sum_a \mu_{ai}^{(c)}(\tau)\right] \mid \underline{U}(t_0)\right\} \tag{4.29}$$

$$\beta\left(\underline{U}(t_0)\right) \triangleq \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{\sum_{i \neq c} U_i^{(c)}(t_0) A_i^{(c)}(\tau) \mid \underline{U}(t_0)\right\}$$

Note that using the stationary randomized control strategy STAT yields a drift variable $\Phi^{STAT}\left(\underline{U}(t_0)\right)$, and from Theorem 6 we have

$$\Phi^{STAT}\left(\underline{U}(t_0)\right) - \beta\left(\underline{U}(t_0)\right) \geq \frac{\epsilon}{2}\sum_{i\neq c} U_i^{(c)}(t_0) \tag{4.30}$$

We now consider a frame-based modification of the DRPC policy which maximizes the $\Phi\left(\underline{U}(t_0)\right)$ function over all conceivable control policies. The modified algorithm FRAME is defined as follows: Scheduling, power allocation, and routing are done every timeslot exactly as in the DRPC algorithm, with the exception that backlog updates are performed only every K slots. Specifically, for any timeslot τ within a K slot frame $\{t_0, t_0{+}1, \ldots, t_0{+}K{-}1\}$, power is allocated to maximize $\sum_{ab}\mu_{ab}(\underline{P}, \underline{S}(\tau))W_{ab}^*(t_0)$ subject to $\underline{P} \in \Pi$. Thus, current channel state information but out of date backlog information is used every slot (note that $W_{ab}^*(t_0)$ depends only on $\underline{U}(t_0)$ according to (4.23)).

Lemma 9. *The control algorithm FRAME maximizes $\Phi\left(\underline{U}(t_0)\right)$ over all possible power allocation, routing, and scheduling strategies. That is:*

$$\Phi^{FRAME}\left(\underline{U}(t_0)\right) \geq \Phi^{X}\left(\underline{U}(t_0)\right)$$

for any other strategy X, including strategies that have full knowledge of arrival and channel statistics.

Proof. Given in Chapter Appendix 4.F. □

Using the above lemma to compare the FRAME and STAT algorithms, it follows that:

$$\Phi^{STAT}\left(\underline{U}(t_0)\right) \leq \Phi^{FRAME}\left(\underline{U}(t_0)\right) \tag{4.31}$$

It follows that the K-step Lyapunov drift of the FRAME algorithm is less than or equal to the drift bound given for the algorithm STAT, and hence by Lemma 2 we have that:

$$\overline{\sum_{i,c} U_i^{(c)FRAME}} \leq \frac{KBN}{\epsilon}$$

We use the algorithm FRAME as an analytical means to prove stability of the DRPC algorithm, not as a recommended control strategy. Note that FRAME and DRPC are equivalent in the case of i.i.d. arrival and channel statistics where $K = 1$. However, for general arrivals and channels, the FRAME algorithm cannot be implemented without knowledge of the convergence interval K, whereas DRPC does not require this knowledge. Intuitively, the DRPC algorithm should perform better than FRAME, as it uses current backlog information. However, analytically we can only show that DRPC performs no more than a fixed amount worse than FRAME. This worst case bound holds because the unfinished work matrix $\underline{U}(\tau)$ for τ within a given frame $\{t_0, \ldots, t_0 + K - 1\}$ differs from the unfinished work $\underline{U}(t_0)$ at the beginning of the frame by no more than a fixed amount determined by the parameters A_{max}, μ_{max}^{in}, and μ_{max}^{out} governing the maximum number of arrivals and departures, as described in the lemma below.

Lemma 10. *Comparing FRAME and DRPC, we have:*

$$\Phi^{DRPC}\left(\underline{U}(t_0)\right) \geq \Phi^{FRAME}\left(\underline{U}(t_0)\right) - (K-1)N\tilde{B}/2$$

where $\tilde{B}$ is defined in (4.27).

Proof. Given in Chapter Appendix 4.F. □

Combining Lemmas 9 and 10, it follows from (4.28) that the Lyapunov drift of the DRPC algorithm satisfies:

$$\begin{aligned}
\overbrace{\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0)) \,|\, \underline{U}(t_0)\right\}}^{DRPC} &\leq K^2BN - 2K\left[\Phi^{DRPC}(\underline{U}(t_0)) - \beta(\underline{U}(t_0))\right] \\
&\leq K^2BN + K(K-1)N\tilde{B} \\
&\quad -2K\left[\Phi^{FRAME}(\underline{U}(t_0)) - \beta(\underline{U}(t_0))\right] \\
&\leq K^2BN + K(K-1)N\tilde{B} - K\epsilon\sum_{i\neq c} U_i^{(c)}(t_0)
\end{aligned}$$

Where the last inequality follows from the fact that $\Phi^{FRAME}(\underline{U}(t_0)) \geq \Phi^{STAT}(\underline{U}(t_0))$ and from (4.30). This drift bound together with Lemma 2 proves Theorem 7.

4.3.4 Delay Asymptotics

Define the constant

$$B_K \triangleq B + \frac{(K-1)}{2K}\tilde{B} \tag{4.32}$$

so that the performance bound of the DRPC policy can be written: $\overline{\sum_i U_i} \leq KB_K N/\epsilon$ (note that $B_K = B$ for $K = 1$). This bound grows asymptotically like $1/\epsilon$ as the data rates are increased, where ϵ can be viewed as the "distance" measure of the rate matrix to the boundary of the capacity region. Such behavior is characteristic of queueing systems, as exemplified by the standard equation for average delay in an M/G/1 queue [49] [14].

Consider now an input rate matrix (λ_{ij}) where each user sends at the same total rate λ, so that each row i of the matrix has the form $(\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{iN})$, and $\sum_j \lambda_{ij} = \lambda$ for all users i. Suppose this rate matrix is a distance ϵ away from a capacity boundary matrix (r_{ij}), where each row i has the form:

$$(r_{i1}, r_{i2}, \ldots, r_{iN}) = (\lambda_{i1} + \epsilon, \lambda_{i2} + \epsilon, \ldots, \lambda_{iN} + \epsilon)$$

Define $R \overset{\triangle}{=} \lambda + N\epsilon$ as the row sum of these rates, representing the total transmission rate if node i were to send according to the rate vector given above. Let $\rho \overset{\triangle}{=} \lambda/R$ represent the effective loading on each user, assumed constant as the network is scaled. Note that $\epsilon = \frac{R}{N}(1-\rho)$. From Little's Theorem, the average bit delay satisfies:

$$\overline{D}_{bit} = \frac{1}{N\lambda} \sum_{i,c} \overline{U}_i^{(c)} \leq \frac{KB_K N}{\epsilon N \lambda} = \frac{KB_K}{\epsilon \rho R} = \frac{KB_K N}{\rho(1-\rho)R^2} \tag{4.33}$$

In a static network where the average distance between users is $O(\sqrt{N})$, such as that given by the Gupta-Kumar model [58] [57], the maximum data rate R for every user necessarily decreases as $O(1/\sqrt{N})$. Hence, the average bit delay of the DRPC algorithm in this scenario is no more than $O(\frac{KN^2}{1-\rho})$.

In ad-hoc mobile networks with full user mobility, it is shown in [54] that the node to node transmission rate R does not decrease with the number of users, so that R is $O(1)$. In this case, average delay is $O(\frac{KN}{1-\rho})$.

4.3.5 Enhanced DRPC

The DRPC algorithm stabilizes the network by making use of back-pressure, where packets find their way to destinations by moving in directions of decreasing backlog. However, when the network is lightly loaded, packets may take many false turns, which could lead to significant delay for large networks. Performance can often be improved by using the

DRPC algorithm with a restricted set of desirable routes for each commodity. However, restricting the routes in this way may reduce network capacity, and may be harmful in time varying situations where networks change and links fail.

Alternatively, we can keep the full set of routes, but program a *bias* into the DRPC algorithm so that, in low loading situations, nodes are inclined to route packets in the direction of their destinations. We use this idea in the following Enhanced DRPC algorithm, defined in terms of constants $\theta_i^c > 0$ and $V_i^c \geq 0$.

Enhanced DRPC Algorithm: For all links (a, b), find commodity c_{ab}^* such that:

$$c_{ab}^* = \arg \max_{c \in \{1,\dots,N\}} \left\{ \theta_a^c (U_a^{(c)}(t) + V_a^c) - \theta_b^c (U_b^{(c)}(t) + V_b^c) \right\}$$

and define:

$$W_{ab}^* = \theta_a^{c_{ab}^*}(U_a^{c_{ab}^*}(t) + V_a^{c_{ab}^*}) - \theta_b^{c_{ab}^*}(U_b^{c_{ab}^*}(t) + V_b^{c_{ab}^*})$$

Power allocation and routing is then done as before, solving the optimization problem (4.24) with respect to these new W_{ab}^* values.

The Enhanced DRPC algorithm can be shown to be stabilizing and to offer a delay bound for any constants $\theta_i^c > 0$ and $V_i^c \geq 0$, while supporting the following services.

Priority Service: The weights θ_i^c of the DRPC algorithm can be used to offer improved service to priority customers, where a large θ_i^c value gives high priority to commodity c packets in node i. Analysis of such a strategy proceeds in a straightforward way by using the weighted Lyapunov function $\sum_i \theta_i^c [U_i^{(c)}]^2$.

Shortest Path Service: Define biases V_i^c to be the distance (or number of hops) between node i and node c along the shortest path through the network (where $V_{ii} = 0$ for all i). These distances can either be estimated or computed by running a shortest path algorithm. With these bias values, packets are inclined to move in the direction of their shortest paths—providing low delay in lightly loaded conditions while still ensuring stability throughout the entire capacity region.

We note that the combined weight $V_a^c + U_a^c$ can be used in the same manner as a routing table, and the unfinished work quantities can be updated each timeslot by having neighboring nodes transmit their backlog changes over a low bandwidth control channel. As each wireless link transmits only a single commodity every timeslot, the number of such backlog increments required to be transmitted over the control channel by any user is on the order of the number of neighboring nodes. This is $O(N)$ for networks where all nodes can reach all other nodes over one hop, but $O(1)$ in systems where attenuation restricts the range of single hop communication to only a small subset of neighbors.

4.3.6 Approximations for Optimal and Suboptimal Control

Analysis similar to the comparison between the frame based and non-frame based DRPC algorithms also can be used to show that DRPC stabilizes the system in cases of *imperfect backlog information.* Specifically, if unfinished work estimates $\hat{U}_{ij}$ are used instead of the actual unfinished work values, and the difference is bounded so that $\mathbb{E}\left\{\left|\hat{U}_{ij} - U_{ij}\right|\right\} < M$ for some constant M, then the DRPC policy still provides stability whenever possible, and average delay grows proportionally to the estimate distance M. Thus, perfect knowledge of queue backlog is not required for throughput optimality, although such knowledge can improve delay.

However, throughput optimality *cannot* be obtained without perfect knowledge of the channel states. Indeed, simple examples can be constructed where the network controller needs full knowledge of random link states to take advantage of good channel conditions while they last. Thus, there is in general a "gap" between the data rates achievable by a centralized controller with full state information and a distributed controller with partial knowledge. A similar gap phenomenon is discussed in [63] for wireless downlink problems.

Consider now a *sub-optimal* controller which allocates power with efforts to approximately solve the optimization problem (4.24). By scaling the data rates (λ_{ic}) in the Lyapunov argument of Theorem 7, it can be shown that if the sub-optimal controller always comes within a factor γ of the optimal solution to (4.24), then the controller guarantees stability for data rates up to a factor γ of the capacity region. (This easily follows by multiplying the $\Phi(\underline{U}(t_0))$ and $\beta(\underline{U}(t_0))$ values in (4.28) by γ and repeating the proof of Theorem 7.) The interpretation of this result is that any effort to allocate power to increase the value of $\sum_{a,b} \mu_{ab}(\underline{P}, \underline{S}(t))W_{ab}^*$ in (4.24) leads to improved data rates.

4.4 Distributed Implementation

The DRPC algorithm of the previous section involves solving a constrained optimization problem every timeslot, where current channel state and queue backlogs appear as parameters in the optimization. Here we consider decentralized implementations, where users attempt to maximize the weighted sum of data rates in (4.24) by exchanging information with their neighbors. The *current neighbors* of a node i is defined as the set $\Omega_i(t)$, representing the nodes to which node i can currently transmit and receive. Theoretically, all nodes could be neighbors, as the power transmitted from one node may be detected everywhere. However, to limit implementation complexity, it is practical to restrict neighbors to a fixed set of nearby nodes with the best channel conditions. We assume that the neighbor sets $\Omega_i(t)$ are defined according to some such rule, and that nodes have knowledge of the link conditions between themselves and their neighbors and are informed of the queue backlogs of their neighbors via a low bandwidth control channel.

4.4.1 Networks with Independent Channels

Consider a network with independent channels, so that the transmission rate on any given link (a, b) depends only on the local link parameters: $\mu_{ab}(\underline{P}, \underline{S}) = \mu_{ab}(P_{ab}, S_{ab})$. Assume that the rate functions $\mu_{ab}(P_{ab}, S_{ab})$ are concave in the single power variable P_{ab} for every channel state S_{ab} (representing diminishing returns in data rate for each incremental increase in power). These assumptions are valid when all links use orthogonal coding schemes, beamforming, and/or when links are spacially separated such that channel interference is negligible.

In this case, the optimization problem (4.24) has a simple *decoupling property*, where the weighted sum is maximized by separately maximizing each term. This corresponds to nodes making independent power control and routing decisions based only on their local information. Indeed, each node $n \in \{1, \ldots, N\}$ maximizes $\sum_b W_{nb}^*(t)\mu_{nb}(P_{nb}, S_{nb}(t))$ subject to its power constraint $\sum_b P_{nb} \leq P_n^{tot}$, where the summations are taken over all neighboring nodes $b \in \Omega_n(t)$. This optimization is a standard problem of concave maximization subject to a simplex constraint, and can be solved easily in real time with any degree of accuracy. Its solution proceeds according to the standard *water-filling* arguments, where power is allocated to equalize scaled derivatives of the $\mu_{nb}(P_{nb}, S_{nb})$ function for a subset of links with the best channel conditions (see Chapter 3, Section 3.4.3). Thus, independent channels enable optimal control to be implemented in a distributed fashion.

4.4.2 Distributed Approximation for Networks with Interference

Consider a network with rate-power curves described by the $\log(1+SIR)$ function given in (1.1). This network has dependent, interfering channels, and the associated optimization problem (4.24) is nonlinear, non-convex, and difficult to solve even in a centralized manner. Here we provide a simple decentralized approximation, where nodes use a portion of each timeslot to exchange control information with neighbors:

1. At the beginning of a timeslot, each node randomly decides to either transmit at full power P_{tot} or remain idle, with probability q for either decision (where $0 \leq q \leq 1$). A control signal of power γP_{tot} is transmitted, where γ is some globally known scaling factor designed to limit power expended by the control signal.

2. Define Q as the set of all transmitting nodes. Each node b measures its total resulting interference $\gamma \sum_{i\in Q} \alpha_{ib}P_{tot}$, and sends this scalar quantity over a control channel to all neighbors.

3. Using knowledge of the interference, attenuation values, and queue backlogs associated with all neighboring nodes, each transmitting user a decides to transmit using full power to the single neighbor b who maximizes the function:
$$W_{ab}^{*} \log\left(1 + \frac{\alpha_{ab}P_{tot}}{N_b + \frac{1}{G_2}\sum_{i\neq a, i\in Q} \alpha_{ib}P_{tot}}\right)$$

Note that each transmitting user a has knowledge of the denominator term $\sum_{i\neq a, i\in Q} \alpha_{ib}P_{tot}$ because this can be obtained by subtracting its own signal strength $\alpha_{ab}P_{tot}$ from the known interference value $\sum_{i\in Q} \alpha_{ib}P_{tot}$. The above algorithm is not optimal, but is designed to demonstrate a simple distributed implementation. The random transmitter selection in the above algorithm is similar to the technique used in the Grossglauser-Tse relay algorithm of [54]. However, rather than transmitting to the nearest receiver, the algorithm chooses the receiver to improve the backlog-rate metric given in (4.24). In the next section, we show that this algorithm achieves a stability region that contains the stability region of the relay algorithm when transmit probability of the relay algorithm is set to q. In particular, in a fully mobile environment, it achieves a capacity which does not vanish as the number of nodes is increased.

4.5 Capacity and Delay Analysis for Ad-Hoc Mobile Networks

Here we analyze the throughput and delay of the above distributed DRPC policy in the situation of an ad-hoc mobile network with N fully mobile users. The steady state location of each user is uniform over the network area, as in [54], and the transmission model is described by the rate-power curves of the $\log(1 + SIR)$ function given (1.1), for the case $G_1 = G_2 = 1$. Every user i has a specified set of *current neighbors* $\Omega_i(t)$ determined by some arbitrary criterion, although we assume that the non-transmitting user closest to user i is always within the neighbor set $\Omega_i(t)$. The distributed DRPC algorithm is designed as a simple approximation to the centralized DRPC algorithm, and can be viewed as an *exact* implementation of DRPC for the following modified channel model: Each user experiences random channel outages where all outgoing links simultaneously fail with probability $1 - q$ (precisely corresponding to the probability of not being chosen to transmit). Additionally, the rate-power function is identically zero for rates between any two users that are not neighbors, and the power set Π is restricted so that all transmitting users must use full power P_{tot} and can transmit to at most one other user during a timeslot.[6] The algorithm achieves capacity for this modified channel model, and hence no other scheme which conforms to such a model (such as the 2-hop relay algorithm in [54]) offers greater throughput. Furthermore, the algorithm admits analytical guarantees on both throughput and delay. To see this, we first present a result from [54] concerning the throughput of the 2-hop relay algorithm.

Define ϕ_N as the average transmission rate between a transmitting user and the nearest non-transmitting user in a network with unit area, N users with uniform location distributions, and interference properties as given by (1.1) with $G_1 = G_2 = 1$.

Fact1. (From [54]) The average rate ϕ_N for transmission between nearest non-transmitting neighbors converges to a positive scalar ϕ as the network size N increases.[7] □

This fact is used in [54] to show that the 2-hop relay algorithm achieves a throughput of $\phi/2$ for sufficiently large networks, where each user communicates at the same rate to

[6] It can be shown that, in the case $G_1 = 1$, the optimal solution of $\sum_{ab} W_{ab}^* \log(1 + SIR)$ (for SIR defined in (1.1)) has the form where each user transmits to no more than one other user on a given timeslot, following from convexity of the $\log(1 + \frac{p}{n-p})$ function.

[7] Strictly speaking, the analysis in [54] chooses transmit nodes according to a globally known *pseudo-random* schedule, so that $\lfloor qN \rfloor$ users transmit on every slot. It is straightforward to show the result also holds (for a slightly different value of ϕ) if transmitters are chosen independently with probability q.

exactly one unique destination. The same rate provides a lower bound on the *stability region* of the distributed DRPC algorithm, where users can communicate with multiple destinations at different rates. This is made formal in the following theorem.

Theorem 8. *If all users move independently and uniformly over the network area, and every user considers its nearest non-transmitting user as part of its current neighbor set, then*

(a) The stability region of the distributed algorithm given above contains the set of all data rates (λ_{ij}) *satisfying:*

$$\textstyle\sum_j \lambda_{ij} \leq \phi_N/2 \rightarrow \phi/2 \quad \text{for all source nodes } i \in \{1, \ldots, N\} \tag{4.34}$$

$$\textstyle\sum_i \lambda_{ij} \leq \phi_N/2 \rightarrow \phi/2 \quad \text{for all destination nodes } j \in \{1, \ldots, N\} \tag{4.35}$$

(b) The average bit delay $\overline{D}_{bit}$ *satisfies:*

$$\overline{D}_{bit} \leq \frac{4KB_K N}{\rho_N(1-\rho_N)\phi_N^2}$$

where $\rho_N \triangleq \frac{\max_{i,j}\{\sum_b \lambda_{ib}, \sum_a \lambda_{aj}\}}{\phi_N/2}$*, and* K*,* B_K *are the stochastic parameters of Theorem 7 corresponding to the particular arrival and mobility process of the system (*K *is defined in (4.1),(4.2) and* $\tilde{B}$ *is defined in (4.27)).*

Proof. (a) Because the distributed DRPC algorithm achieves capacity over the modified channel model, from Theorem 1 it suffices to find a link matrix (G_{ab}) within the graph family $\tilde{\Gamma}$ for the modified model together with multicommodity flows $f_{ab}^{(c)}$ which support the data rates with respect to this link matrix. We emphasize that the link matrix and flows do not need to be those resulting from the distributed DRPC algorithm, but can be from *any* power allocation and routing strategy that conforms to the channel model.

To find a link matrix (G_{ab}), consider the simple strategy of transmitting with full power to the nearest non-transmitting user. From Fact 1, the average rate of this transmission is ϕ_N. In steady state, the nearest neighbor of a node a is equally likely to be any of the other $N-1$ nodes in the system, so that the long term link rate between any node pair (a, b) is given by $G_{ab} = \phi_N/(N-1)$. Hence, the link matrix $(G_{ab}) = (\phi_N/(N-1))$ is contained in $\tilde{\Gamma}$.

Define the following flow variables $f_{ab}^{(c)}$:

$$
\begin{array}{ll}
f_{ab}^{(a)} = f_{aa}^{(c)} = 0 & \text{for all } a, b, c \in \{1, \ldots, N\} \\
f_{ab}^{(c)} = \frac{\lambda_{ac}}{(N-1)} & \text{for all sources } a\text{, commodities } c\text{, and all nodes } b \notin \{a, c\} \\
f_{ab}^{(b)} = \frac{\sum_k \lambda_{kb}}{(N-1)} & \text{for all nodes } a\text{, } b \text{ such that } a \neq b.
\end{array}
$$

The above flows correspond to routing the λ_{ac} traffic from node a to node c by splitting it equally among $N-1$ parallel paths, consisting of the single hop path $a \rightarrow c$ and the $N-2$ two-hop paths which use an intermediate relay node b to send data along the path $a \rightarrow b \rightarrow c$.

It is easy to verify that these flows satisfy the non-negativity, flow efficiency, and flow conservation constraints of (4.4) - (4.6). It suffices to show that the link constraint (4.7) is satisfied. For all wireless links (a, b), we have:

$$
\begin{aligned}
\sum_c f_{ab}^{(c)} &= f_{ab}^{(a)} + f_{ab}^{(b)} + \sum_{c \notin \{a,b\}} f_{ab}^{(c)} \\
&= 0 + \frac{\sum_k \lambda_{kb}}{(N-1)} + \sum_{c \notin \{a,b\}} \frac{\lambda_{ac}}{N-1} \\
&\leq \frac{\phi_N}{2(N-1)} + \frac{\phi_N}{2(N-1)} = G_{ab} \qquad (4.36)
\end{aligned}
$$

where (4.36) follows from the input rate constraints (4.34), (4.35). Hence, the link constraint is satisfied for all links, and the proof of (a) is complete. Part (b) follows from the delay expression (4.33) established by Theorem 7, noting that $R = \phi_N/2$ in this case. □

4.5.1 Implementation and Simulation of Centralized and Distributed DRPC

Here we apply the Enhanced DRPC policy to an ad-hoc network with mobility and inter-channel interference. Consider a square network with N users, with user locations discretized to an $M \times M$ grid (see Fig. 4-3). The stochastic channel process $\underline{S}(t)$ is characterized by the following stochastic model of user mobility: Every timeslot, users keep their locations with probability 1/2, and with probability 1/2 they move one step in either the North, South, West, or East directions (uniformly distributed over all feasible directions). Each user is power constrained to P_{tot}, is restricted to transmitting to only one other user in a given timeslot, and cannot transmit if it is receiving. Power radiates omnidirectionally, and signal attenuation between two nodes a and b is determined by the 4^{th} power of the distance between them (as in [41]), so that fading coefficients are given by:

$$\alpha_{ab} = \begin{cases} 1/[((x_a - x_b)^2 + (y_a - y_b)^2)^2 + 1] & \text{if } a \neq b \\ \infty & \text{if } a = b \end{cases}$$

where (x_a, y_a), (x_b, y_b) represent user locations within the network. Note that the extra "+1" term in the denominator is inserted to model the reality that attenuation factors α_{ab} are kept below 1 (so that signal power at the receiver is never more than the corresponding power used at the transmitter). The α_{aa} values are set to infinity to enforce the constraint that transmitting nodes cannot receive.

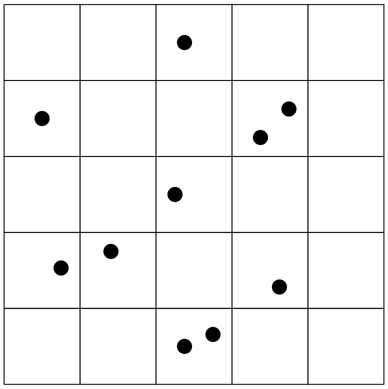


Figure 4-3: An ad-hoc network with 10 mobile users. Locations of each user are discretized to a 5 × 5 grid.

modulation	bits/symbol	SIR/symbol
2 PAM	1	$0.25\,\Delta^2$
4 QAM	2	$0.50\,\Delta^2$
16 QAM	4	$1.25\,\Delta^2$
64 QAM	6	$5.25\,\Delta^2$

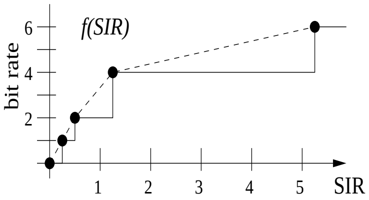


Figure 4-4: A piecewise constant rate curve for the 4 modulation schemes described in the table. Scaled power requirements are shown, where Δ represents the minimum distance between signal points.

Multi-user interference is modeled similarly to the rate-power curve given in (1.1). However, rather than use the $\log(1 + SIR)$ function, we use a rate curve determined by four different QAM modulation schemes designed for error probabilities less than 10^{-6}. The rate function is thus:

$$\mu_{ab}(\underline{P}, \underline{\alpha}) = f(SIR_{ab}(\underline{P}, \underline{\alpha}))$$

where $f()$ is a piecewise constant function of the signal-to-interference ratio defined by the coding schemes given in Fig. 4-4. The $SIR_{ab}()$ function is taken to be the same as that used in eq. (1.1) with the CDMA gain parameters $G_1 = G_2 = 1$.

We consider the Enhanced DRPC algorithm with $\theta_i^c = 1$, $V_i^c = 1$ for all $i \neq c$, and $V_i^i = 0$, and assume the power/noise coefficient is normalized to $P_{tot}/N_b = 20\Delta^2$, where Δ is the minimum distance between signal points in the QAM modulation scheme. The algorithm is approximated using the distributed implementation described in the previous section, where each node transmits using full power with probability $q = 1/2$. As the network is small, we simply define the neighbor set $\Omega_i(t)$ for each user i to be the set of all other nodes in the network (because of attenuation affects, we do not expect performance to be significantly affected if this neighbor set is restricted to the set of users within one or two cells of the transmitter). A centralized implementation is also considered, where the optimization problem (4.24) is implemented using a steepest ascent search on the piecewise linear relaxation of the $f(SIR)$ curve (see Fig. 4-4). The resulting data rates are then "floored" according the threshold levels of the piecewise constant curve $f(SIR)$. Note that the relaxed problem remains non-linear and non-convex (because SIR is non-convex in the power variables, see (1.1)), and hence the result of the steepest ascent search may be sub-optimal.

We simulate the centralized and decentralized implementations of DRPC and compare to the performance offered by the 2-hop relay algorithm presented in [54]. The relay algorithm restricts routes to 2-hop paths, and hence relies on rapid user mobility for delivering data. We set the sender density parameter of the relay algorithm to $q \triangleq 1/2$. To further conform with the Grossglauser-Tse model, we assume both the 2-hop relay algorithm and the distributed DRPC algorithm choose users pseudo-randomly according to this density, so that there are always $N/2$ transmitters. Random and i.i.d. transmitter selection yields similar results. Note that the relay algorithm was developed to demonstrate non-vanishing capacity for large networks, and was not designed to maximize throughput or achieve low delay. Thus, it is not completely fair to compare performance with the DRPC algorithms. However, the comparison illustrates the capacity gains and delay reductions that can be achieved in this mobile ad-hoc network setting.

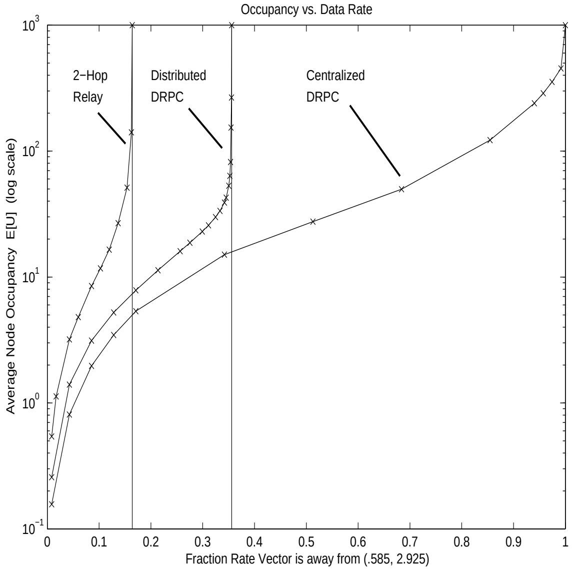


Figure 4-5: Simulation results for the DRPC algorithm and the relay algorithm as rates are increased towards $(\lambda_1, \lambda_2) = (.585, 2.925)$.

The relay algorithm was designed for nodes to transmit data at a fixed rate, attainable whenever the SIR for a given wireless link exceeds a threshold value. However, in order to make a fair comparison, we allow the relay algorithm to transmit at rates given by the full $f(SIR)$ curve.

Here we consider a small network with 10 users communicating on a 5×5 square region (see Fig. 4-3). Following the scenario of [54], we assume user i desires communication with only one other user (namely, user $(i+1) \mod N$). Unit length packets arrive according to Poisson processes, where 9 of the users receive data at rate λ_1, and the remaining user receives data at rate λ_2. In Fig. 4-5 we plot the average network delay from simulation of the three algorithms when the rates (λ_1, λ_2) are linearly scaled upwards to the values $(.585, 2.925)$. From the figure, we see that the centralized DRPC algorithm provides stability and bounded delays at more than four times the data rates of the 2-hop relay algorithm, and more than twice the data rate of the decentralized DRPC algorithm. We further note that the 2-hop relay algorithm relies on full and homogeneous mobility of all users, while the DRPC algorithms have no such requirement and can be used for heterogeneous networks with limited mobility.

4.6 Satellite Constellation Networks

Here we describe an implementation of the DRPC policy for the specific case of control for a satellite network with optical crosslinks between satellites and RF downlinks from satellite to ground (Fig. 4-6). The satellite constellation forms a network with a fixed or periodically varying topology, and the optical crosslinks connecting satellites are reliable and operate at very high capacity. The satellites connect to ground with multi-beam technology, and the downlink channels have lower capacity and are susceptible to random variations due to scintillation and weather conditions.

4.6.1 The Satellite Separation Principle

Consider the system of satellites, crosslinks, and downlinks shown in Fig. 4-6. We again assume a timeslotted structure for the system, so that data arrives and scheduling is initiated on timeslot boundaries. There are M satellites, and J ground users. Let (λ_{ij}) represent the $M \times J$ matrix of arrival rates, where λ_{ij} represents the rate of data arriving to satellite i destined for ground location j. We note that each ground user often has a single designated satellite from which it receives downlink transmissions, as shown in Fig. 4-6. However, we also consider the case in which the multi-beam patterns of different satellites can overlap, so that a single ground user can be reached via two or more satellites.

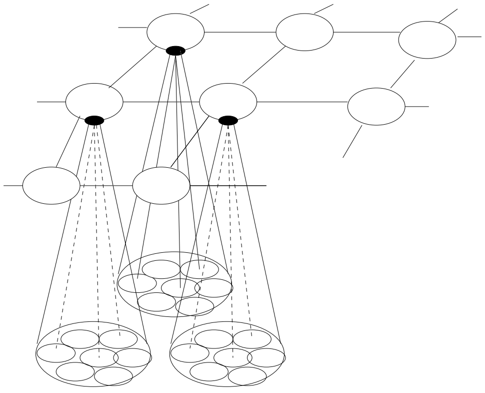


Figure 4-6: A satellite network with optical crosslinks connecting satellites and RF downlinks from satellite to ground.

Each active crosslink can transfer data at a rate of C bits/timeslot. Crosslinks may become inactive due to periodic topology changes or random link failures. For downlink communication, each satellite i has a fixed power P_{tot}^{i} which it allocates to its downlink channels. Data rates on each downlink are determined by the time varying channel conditions and the current power allocation.

Here we address the problem of jointly routing data over the satellite constellation and allocating power at each downlink. The goal is to support the full throughput matrix (λ_{ij}) and thereby stabilize the network. Note that the rate matrix, link failure probability, and time varying channel statistics are unknown to the network controller. As before, we define Λ as the capacity region of the network, that is, Λ is the closure of all stabilizable rate matrices (λ_{ij}). Here we assume that the downlink beams of individual satellites do not interfere with each other, so that each satellite m has an associated rate-power curve $\underline{\mu}_m(\underline{P}^{(m)}, \underline{S}^{(m)})$ which depends only on the channel states and power allocations of its own downlinks (Fig. 4-6). We call this assumption the *satellite independence assumption.* Note that such an assumption is still valid in cases where multiple satellites can transmit to the same ground user, provided that these different transmissions are orthogonal in time or frequency.

Separation Principle: Under the satellite independence assumption, joint optimal control can be *separated* into independent controllers acting on each satellite downlink and a crosslink controller acting on the satellite constellation.

We note that this result is surprising. For example, a particular satellite must make decisions about whether to emphasize the service of $U^{(a)}$ backlog or $U^{(b)}$ backlog for downlink transmissions, and one would expect an optimal decision to require knowledge of how much commodity a traffic and commodity b traffic is entering and congesting other parts of the constellation. Such knowledge is not necessary, and the decoupled downlink and crosslink algorithms that achieve stability are described below.

Downlink Algorithm: The downlink algorithm is similar to that given in Chapter 3, where each satellite stores data in different output queues corresponding to its destination. Let $U_i^{(j)}(t)$ represent the current backlog in satellite i destined for the j^{th} ground user. For each satellite i, define R_i as the set of reachable ground users. Every timeslot t, power is allocated to maximize $\sum_{r\in R_i} \mu_{ir}(\vec{P}, \vec{S}(t))U_i^{(r)}(t)$ subject to the power constraint $\sum_{r\in R_i} P_r \leq P_{tot}^i$, where $\vec{S}(t)$ is the current channel state vector for downlinks from satellite i to users $r \in R_i$, and the function $\mu_{ir}(\vec{P}, \vec{S})$ is the link budget curve for the (i, r) downlink that describes data rate as a function of the allocated power vector and current channel state. Note that this allows for possible interchannel interference for downlink channels of the same satellite.

Crosslink Algorithm: Each satellite keeps internal queues for storing crosslink traffic. Again let $U_i^{(c)}(t)$ represent the current backlog in satellite i destined for the c^{th} ground user, and note that this data is necessarily crosslink traffic if $c \notin R_i$, while the data could either be transferred over the crosslinks to a different satellite or could be directly transmitted to the ground if $c \in R_i$. Throughout the constellation, data destined for a common ground user c is defined as *commodity* c data. Every timeslot and for each active link connecting satellite i to a neighboring satellite j, determine the commodity c which maximizes the *differential backlog* $U_i^{(c)}(t) - U_j^{(c)}(t)$. If the resulting differential backlog is positive, route as much of this commodity over the link as possible, up to the link capacity C bits/slot. If a satellite does not have enough of commodity c data for all of its outgoing crosslink and downlink transmissions, the commodity is split arbitrarily amongst the outgoing links. If there is sufficient capacity available for more transmissions over a particular crosslink, the next commodity which maximizes differential backlog is found and the procedure is repeated.[8]

Because the joint algorithm described above is simply an implementation of the DRPC algorithm applied to this satellite constellation network, it stabilizes the system whenever possible. The downlink algorithm is distributed in that individual downlinks use only local channel information and do not require knowledge of channel conditions or traffic congestion in other parts of the network. The crosslink algorithm is also distributed, although each satellite requires knowledge of queue backlog levels for neighboring satellites.

The differential backlog policy stabilizes the system by making use of *backpressure*, where data finds its way to the destination by moving in the direction of least resistance while being pushed by newly arriving data. As the routing policy does not require knowledge of the constellation topology, it is robust to changing topologies and can quickly adapt to rare events such as link failures. However, the policy suffers from the "infinite random walk" effect that can occur when the system is very lightly loaded, as a single packet can

[8] To prevent excessive crosslink exchange for data that has already reached its primary downlink satellite, it is useful to form unfinshed work values $\tilde{U}_i^{(c)}(t) \triangleq U_i^{(c)}(t) + V_i^{(c)}$, where $V_i^{(c)} \triangleq V$ for all commodities c such that satellite i is not the primary downlink, and $V \geq 0$ is a suitable bias value.

randomly traverse the network and may never find the destination if there are no newly arriving packets to provide backpressure. This problem can be overcome by biasing data so that packets are inclined to move along their shortest paths to the destination but can change course if needed (as described by the Enhanced DRPC policy of Section 4.3.5), or by restricting routing options so that packets only move along paths which bring them closer to their destinations. In the next subsection we provide an example where such restricted routing does not decrease network capacity.

4.6.2 Satellite Constellation Capacity for a Torus Network

Consider a satellite constellation network with a torus topology, as shown in Fig. 4-7.[9] The constellation is composed of $\sqrt{M}$ horizontal rings of satellites and $\sqrt{M}$ vertical rings, so that there are M satellites total. Assume that all crosslinks are bi-directional and can support C bits of data per timeslot in both directions, and that these links never fail. Further assume that each packet is intended for a particular satellite, but can be delivered over any downlink of that satellite, so that the matrix of arrival rates (λ_{ij}) now describes the rate of data entering satellite i and destined for satellite j (so that individual downlinks of each satellite are not indexed). For simplicity, assume that the uplink and downlink channel processes of the network are such that each satellite can exogenously receive a time average input rate of μ_{up} bits per slot (received from the ground), and can deliver a time average output rate of μ_{down} bits per slot to the ground.

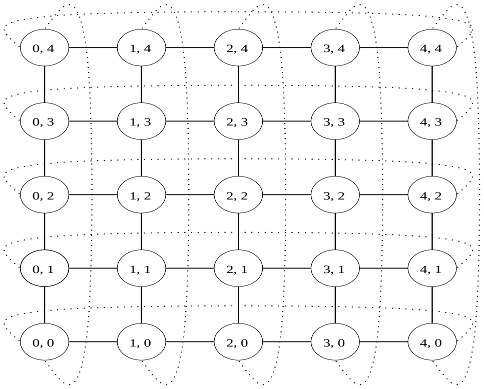


Figure 4-7: A satellite constellation arranged as an $N \times N$ torus network.

It follows that the capacity region Λ for such a network is limited by the following

[9] We would like to thank Jun Sun for donating this figure.

constraints:

$$\sum_i \lambda_{ij} \leq \mu_{down} \text{ for all } j \in \{1, \ldots, M\} \tag{4.37}$$

$$\sum_j \lambda_{ij} \leq \mu_{up} \text{ for all } i \in \{1, \ldots, M\} \tag{4.38}$$

However, if the crosslink capacity satisfies $C \geq \frac{\sqrt{M}}{2}\max[\mu_{up}, \mu_{down}]$, then the above constraints completely describe the capacity region Λ of the satellite constellation. Thus, if the crosslink capacity is suitably larger than the capacity of the uplink or downlink, the constellation is functionally *invisible*, and network capacity is limited only by the uplinks and downlinks.

To see this, note that constraints (4.37) and (4.38) are certainly necessary (otherwise the uplink or downlink constraints would be violated at some satellite). We prove sufficiency by appealing to Theorem 5, which implies that it is sufficient to find a set of multi-commodity flows which successfully route all traffic to its proper destination without violating any link constraints. First note that (4.37) and (4.38) imply that no uplink or downlink is overloaded. Now consider the simple 2-stage, shortest path routing strategy for data originating at satellite i and destined for satellite j: The data is first routed along the vertical ring i in the direction of the horizontal ring on which j lies (requiring at most $\sqrt{M}/2$ hops). Once it reaches this ring, it is routed on the horizontal ring in the direction of the destination.

Thus, vertical rings carry only traffic originating from their respective satellites, and horizontal rings carry only traffic destined for their respective satellites. Each vertical link thus carries flows from at most $\sqrt{M}/2$ source satellites, and hence the maximum sum rate of flow over any such link is $\mu_{up}\sqrt{M}/2$. Likewise, the maximum sum rate over any horizontal ring is $\mu_{down}\sqrt{M}/2$. Therefore, if each crosslink can support at least $C \geq \frac{\sqrt{M}}{2}\max[\mu_{up}, \mu_{down}]$ bits per slot, then no link condition is violated, proving the result. This discussion further proves that under this example of reliable, high capacity internal links, restricting the routing options to shortest hop paths does not restrict the network capacity region.

4.7 Multi-commodity Flows and Convex Duality

The DRPC algorithm stabilizes the network and offers average delay guarantees whenever the input rate matrix is inside the capacity region of the wireless network. Here we consider a related problem of computing an offline multi-commodity flow given a known rate matrix (λ_{ij}). Classical multi-commodity flow problems for wired networks can be reduced to linear programs, and fast approximation algorithms are developed in [87]. A distributed algorithm was first given in [48], and game theory approaches are developed in [78].

Here we consider wireless networks, and note that the problem cannot be distributed unless channels are independent. A convex optimization problem corresponding to a multicommodity flow in the wireless network is formulated, and it is shown that a classical subgradient search method for solving the problem via convex duality theory corresponds exactly to a deterministic network simulation of the DRPC policy. Notions of duality are also used in [148] [92] [78] [93] to consider static network optimization, where dual variables play the role of prices charged by the network to multiple users competing for shared network resources in order to maximize their own utility. In our context, the dual variables correspond to queue backlogs, rather than network prices. This illustrates a relationship between static optimization and the dynamic DRPC policy and contributes to a growing theory of *dynamic optimization*, suggesting that static algorithms can be modified and applied in dynamic settings while preserving analytical optimality.

We restrict attention to time invariant systems, so that the rate-power curve is only a function of power: $\underline{\mu}(\underline{P}, \underline{S}) = \underline{\mu}(\underline{P})$. Given a rate matrix (λ_{ij}), the problem of finding a multi-commodity flow corresponds to the following convex optimization problem.

Maximize: 1

Subject to:

$$\begin{aligned} &\lambda_{ic} + \textstyle\sum_a f_{ai}^{(c)} \leq \sum_b f_{ib}^{(c)} \quad \text{for all nodes } i \neq c \\ &\left(\left\{f_{ab}^{(c)}\right\}, \{\mu_{ab}\}\right) \in \Theta \end{aligned} \tag{4.39}$$

where: $\Theta =$ The set of all variables $\left(\left\{f_{ab}^{(c)}\right\}, \{\mu_{ab}\}\right)$ such that:

$$\begin{aligned} f_{ab}^{(c)} \geq 0 \quad & \text{for all } a, b, c \in \{1, \ldots, N\} \\ f_{aa}^{(c)} = f_{ab}^{(a)} = 0 \quad & \text{for all } a, b, c \in \{1, \ldots, N\} \\ \left(\textstyle\sum_c f_{ab}^{(c)}\right) \leq \mu_{ab} \quad & \text{for some } (\mu_{ab}) \in \Gamma \end{aligned} \tag{4.40}$$

The maximization function "1" is used as an artifice to pose this multi-commodity flow problem in the framework of an optimization problem. Note that the set Θ is convex and compact (it inherits convexity and compactness from the set $\tilde{\Gamma}$ consisting of all link transmission rate matrices (G_{ab}) entrywise less than or equal to some element of Γ). Moreover, the objective function "1" and all inequality constraints are linear. The optimization problem is therefore convex [15], and has a dual formulation, where the optimal solution of the dual problem exactly corresponds to an optimal solution of the original "primal" problem (4.39). To form the dual problem, we introduce non-negative Lagrange multipliers $\{U_i^{(c)}\}$ for each of the inequality constraints in (4.39), and define the dual function:

$$L(\{U_i^{(c)}\}) = \max_{\left(\left\{f_{ab}^{(c)}\right\},\{\mu_{ab}\}\right)\in\Theta} \left[1 + \sum_{i\neq c} U_i^{(c)} \left(\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} - \lambda_{ic}\right)\right] \tag{4.41}$$

The dual problem to (4.39) is:

$$\textit{Minimize: } L\left(\left\{U_i^{(c)}\right\}\right) \qquad \textit{Subject to: } U_i^{(c)} \geq 0 \text{ for all } i, c \in \{1, \ldots, N\} \tag{4.42}$$

The dual problem is always convex, and the minimizing solution can be obtained using classical subgradient search methods (where the function $-L\left(\left\{U_i^{(c)}\right\}\right)$ is maximized). Consider a fixed stepsize method with stepsize $T = 1$. The basic subgradient search routine starts with an initial set of values $U_i^{(c)}(0)$ for the Lagrange multipliers, and upon each iteration $t = \{1, 2, \ldots\}$ these values are updated by computing a subgradient $\underline{\eta}$ for one time unit, and, if necessary, projecting the result back onto the set of non-negative values [15]:

$$U_i^{(c)}(t+1) = \max\left[U_i^{(c)}(t) + \eta_i^{(c)}, 0\right] \tag{4.43}$$

However, it is shown in [15] that a particular subgradient of $-L\left(\left\{U_i^{(c)}\right\}\right)$ is:

$$\underline{\eta} = \left.\left(\sum_a f_{ai}^{*(c)} - \sum_b f_{ib}^{*(c)} + \lambda_{ic}\right)\right|_{(i,c)\in\{1,\ldots,N\}^2} \tag{4.44}$$

where the $\{f_{ab}^{*(c)}\}$ variables are solutions to the maximization in (4.41). Using (4.44) in (4.43) for all $i \neq c$, we find

$$U_i^{(c)}(t+1) = \max\left[U_i^{(c)}(t) + \sum_a f_{ai}^{*(c)} - \sum_b f_{ib}^{*(c)} + \lambda_{ic}, 0\right] \tag{4.45}$$

From the above equation, it is apparent that the Lagrange multipliers $\{U_i^{(c)}(t)\}$ play the role of unfinished work in a multi-node queueing system with input rates λ_{ic}, where $U_i^{(c)}(t)$ represents the amount of commodity c bits in node i. Likewise, the $f_{ab}^{*(c)}$ values can be viewed as the transmission rates allocated to commodity c traffic on the (a, b) link during the first timestep. Equation (4.45) thus states that the unfinished work at time $t+1$ is equal to the unfinished work at time t plus the net influx of bits into node i. Thus, the operation of projecting the Lagrangian variables onto the positive orthant acts exactly as an implementation of the standard queueing equation for backlog at time $t+1$ in terms of backlog at time t.

It is illuminating to calculate the optimal $f_{ab}^{*(c)}$ values by performing the maximization in (4.41). To this end, we need to maximize $\sum_{i\neq c} U_i^{(c)}(t)\left(\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)}\right)$ subject to the constraints of (4.40). However, as in the proof of Theorem 3, we can switch the sum to find:

$$\sum_{i\neq c} U_i^{(c)}(t)\left(\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)}\right) = \sum_{ab}\sum_c f_{ab}^{(c)}\left[U_a^{(c)} - U_b^{(c)}\right]$$

Remarkably, from the right hand side above, it is apparent that the optimal values $f_{ab}^{*(c)}$ are identical to the resulting link rates $\mu_{ab}^{(c)}(\underline{P})$ that would be computed if the DRPC algorithm were used to calculate routing and power allocation decisions in a network problem with unfinished work levels $U_i^{(c)}(t)$. It follows that the DRPC algorithm can be viewed as a dynamic implementation of a subgradient search method for computing the solution to an optimization problem using convex duality. This suggests a deeper relationship between stochastic network control algorithms and subgradient search methods. It would be interesting to explore how the two interact and build upon each other. For example, there are several known improvements to classical subgradient search routines. Perhaps such improvements could reduce the complexity of optimal and sub-optimal dynamic network controllers. Also note that the optimization problem (4.39), which maximizes the function "1," can be adjusted to maximize some other performance criteria, which may offer additional quality of service guarantees in the corresponding dynamic network control problem. This observation inspires our approach to the fairness problem, developed in Chapter 5.

4.8 Chapter Summary

We have formulated a general power allocation problem for a multi-node wireless network with time varying channels and adaptive transmission rates. The problem was formulated at the network layer assuming a given (but arbitrary) set of rate-power functions corresponding to the particular modulation and coding strategy being used at the physical layer. These rate-power curves are general enough to include hybrid networks with both wireless and wireline components. The network capacity region was established, and a Dynamic Routing and Power Control (DRPC) algorithm was developed and shown to stabilize the network whenever the arrival rate matrix is within the capacity region. Such stability holds for arbitrary ergodic arrival and channel processes, even if these processes are unknown to the network controller. Delay bounds were derived and shown to grow asymptotically in N and ϵ, representing the size of the network and the distance the arrival rates are to the capacity region boundary.

The DRPC algorithm was shown to have a decentralized implementation for networks with independent channels. A simple distributed approximation algorithm was developed for networks with inter-channel interference. The algorithm was shown to support a larger set of data rates than the Grossglauser-Tse 2-hop relay algorithm, and explicit throughput regions and delay bounds were computed.

The DRPC algorithm involves solving a constrained optimization problem during each timeslot, where queue backlogs and channel conditions occur as parameters in the optimization. Algorithms which make more effort to maximize the optimization metric by exchanging backlog and channel information were shown to have significant performance advantages, as illustrated by the example simulations. Furthermore, the dynamic control algorithm was shown to be fundamentally related to a classical iterative technique for solving a static convex program, where unification of the two problems is achieved through the theory of convex duality. We believe that such *dynamic optimization* contributes to bridging the gap between theoretical optimization techniques and implementable control algorithms.

Chapter Appendix 4.A — Necessary Condition for Network Stability

Necessary Condition for Network Stability (From Theorem 5): Here we establish that $(\lambda_{ij}) \in \Lambda$ is a necessary condition for stability in a wireless network. The proof uses the following preliminary lemma:

Lemma 11. *(Set Integration) Suppose an instantaneous rate matrix $\underline{\mu}(t)$ is integrable and lies within a set Ω for all time. Then $\frac{1}{||T||}\int_{\tau\in T}\underline{\mu}(\tau)d\tau$ lies within the convex hull of Ω, where T is a set of times with measure $||T||$.*

Proof. The proof relies on the convex set separation theorem [15] and is proven in Chapter Appendix 4.B. □

Theorem 5a. *(Necessary Condition for Stability) The condition $(\lambda_{ic}) \in \Lambda$ is necessary for network stability.*

Proof. Consider a system with ergodic inputs with rates (λ_{ic}), and let process $X_i^{(c)}(t)$ represent the amount of commodity c bits that exogenously enter the network at node i during the interval $[0, t]$. Suppose the system is stabilizable by some routing and power control policy, perhaps one which bases decisions upon complete knowledge of future arrivals and channel states. Note that although the policy stabilizes the system, the power allocations $\underline{P}(t)$ are not necessarily ergodic, nor are the internal bit streams produced by routing decisions. Let $U_i^{(c)}(t)$ represent the resulting unfinished work function for commodity c in node i under this stabilizing policy. Further, let $F_{ab}^{(c)}(t)$ represent the total number of bits from commodity c transmitted over the (a, b) link during the interval $[0, t]$. We have for all time:

$$F_{ab}^{(c)}(t) \geq 0 \quad \forall a, b, c \tag{4.46}$$

$$F_{aa}^{(c)}(t) = F_{ab}^{(a)}(t) = 0 \quad \forall a, b, c \tag{4.47}$$

$$X_i^{(c)}(t) - U_i^{(c)}(t) = \textstyle\sum_b F_{ib}^{(c)}(t) - \sum_a F_{ai}^{(c)}(t) \quad \forall i \neq c \tag{4.48}$$

$$\textstyle\sum_c F_{ab}^{(c)}(t) \leq \int_0^t \mu_{ab}(\underline{P}(\tau), \underline{S}(\tau))d\tau \quad \forall (a, b) \tag{4.49}$$

where (4.48) follows because the unfinished work in any node is equal to the difference between the total bits that have arrived and departed. Inequality (4.49) holds because the total bits transferred over any link (a,b) is less than or equal to the offered transmission rate integrated over the time interval $[0,t]$.

Let $T_{\underline{S}}(t)$ represent the subintervals of $[0,t]$ during which the channel is in state $\underline{S}$, and let $||T_{\underline{S}}(t)||$ denote the total length of these subintervals. Fix an arbitrarily small value $\epsilon > 0$. Because the channel process $\underline{S}(t)$ is channel convergent on a finite state space, and because there are a finite number of rate convergent input streams $X_i^{(c)}(t)$, when measured over any sufficiently large interval $[0,t]$ the time average fraction of time in each channel state and the empirical average data rate of all inputs are simultaneously within ϵ of their limiting values. Furthermore, by the Network Stability Necessary Condition (Lemma 1 of Chapter 2), there must exist some finite value V such that at arbitrarily large times $\tilde{t}$, the unfinished work in all queues is simultaneously less than V with probability at least 1/2. Hence, there exists a time $\tilde{t}$ such that with probability at least 1/2, all of the following inequalities are satisfied:

$$U_i^{(c)}(\tilde{t}) \leq V \text{ for all nodes } i \text{ and commodities } c \tag{4.50}$$

$$\tfrac{V}{\tilde{t}} \leq \epsilon \tag{4.51}$$

$$\tfrac{X_{ic}(\tilde{t})}{\tilde{t}} \geq \lambda_{ic} - \epsilon \;\text{ for all } i \neq c \tag{4.52}$$

$$\tfrac{||T_{\underline{S}}(\tilde{t})||}{\tilde{t}} \leq \pi_{\underline{S}} + \epsilon \;\text{ for all channel states } \underline{S} \tag{4.53}$$

Now define variables $f_{ab}^{(c)} \triangleq F_{ab}^{(c)}(\tilde{t})/\tilde{t}$. It is clear from (4.46) and (4.47) that these flow variables satisfy the non-negativity and flow efficiency constraints (4.4) and (4.5). Using (4.50)-(4.52) in (4.48), it follows that for all $i \neq c$:

$$\lambda_{ic} \leq \sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} + 2\epsilon \tag{4.54}$$

and hence the flow conservation constraint is arbitrarily close to being satisfied. Applying inequality (4.49) at time $\tilde{t}$, dividing by $\tilde{t}$, and considering entrywise matrix inequalities, we have:

$$\begin{aligned}\left(\textstyle\sum_c f_{ab}^{(c)}\right) &\leq \left(\tfrac{1}{\tilde{t}}\textstyle\int_0^{\tilde{t}} \mu_{ab}(\underline{P}(\tau), \underline{S}(\tau))d\tau\right) \\ &= \textstyle\sum_{\underline{S}} \tfrac{||T_{\underline{S}}(\tilde{t})||}{\tilde{t}} \tfrac{1}{||T_{\underline{S}}(\tilde{t})||}\left(\textstyle\int_{\tau \in T_{\underline{S}}(\tilde{t})} \mu_{ab}(\underline{P}(\tau), \underline{S})d\tau\right) \\ &\leq \textstyle\sum_{\underline{S}} \tfrac{||T_{\underline{S}}(\tilde{t})||}{\tilde{t}}(\mu_{ab}^{\underline{S}})\end{aligned} \tag{4.55}$$

where the matrices $(\mu_{ab}^{\underline{S}})$ in (4.55) are elements of Convex_Hull$\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ and exist by the Set Integration Lemma (Lemma 11). Using (4.53) in (4.55), we find:

$$\left(\sum_c f_{ab}^{(c)}\right) \leq \sum_{\underline{S}} \pi_{\underline{S}}(\mu_{ab}^{\underline{S}}) + \epsilon(\mu_{ab}^{max})\text{Card}\{\underline{S}\} \tag{4.56}$$

where Card$\{\underline{S}\}$ represents the number of channel states $\underline{S}$, and μ_{ab}^{max} represents the maximum transmission rate of the (a, b) link over all channel states and power levels $\underline{P} \in \Pi$. Hence, the right hand side of inequality (4.56) is arbitrarily close to a point in Γ (compare with (4.7)).

Hence, with probability greater than 1/2, the multicommodity flows $f_{ab}^{(c)}$ (defined in terms of the $F_{ab}^{(c)}(t)$ processes) come arbitrarily close to satisfying the non-negativity, flow efficiency, flow conservation, and link constraints. It follows that *there must exist* sample paths $F_{ab}^{(c)}(t)$ from which flow variables $f_{ab}^{(c)}$ can be defined that satisfy (4.54) and (4.56) (otherwise, the inequalities would occur with probability 0). As the flow efficiency and link constraints are arbitrarily close to being satisfied, it follows that they *can* be satisfied if each nonzero entry of the (λ_{ic}) rate matrix is reduced by an arbitrarily small amount. This proves that the input rate matrix (λ_{ic}) is a limit point of the capacity region Λ. Because Λ is compact and hence contains its limit points, it follows that $(\lambda_{ic}) \in \Lambda$. □

Chapter Appendix 4.B — Multi-dimensional Integration Theorem

In this section we prove a multi-dimensional integration theorem (from Lemma 11 of Chapter Appendix 4.B, used in the development of the necessary condition for network stability).

Let $\vec{\mu}(t)$ represent a vector function of time taking values in $\mathbb{R}^N$. The sample average of $\vec{\mu}(t)$ taken at times $t_1, t_2, \ldots, t_m$ is written $\frac{1}{m}\sum_{i=1}^{m}\vec{\mu}(t_i)$. If $\vec{\mu}(t)$ takes values in a set A, then this average constitutes a convex combination of points in A, and hence is contained in the convex hull of A. Intuitively, the same result is true for time average integrals of $\vec{\mu}(t)$, because integrals can be represented as limits of finite sums. However, such a limiting argument cannot be used in general, as the set A may not contain all its limit points. The following theorem proves the result by using the *convex set separation theorem* [15], which states that a convex set and a point not in the set can be separated by a hyperplane.

Theorem 9. *(Time Average Integration) If $\vec{\mu}(t)$ is integrable and is contained within a set A for all time, then the time average integral of $\vec{\mu}(t)$ over any set of times T with finite measure $||T||$ is within the convex hull of A, i.e.:*

$$\frac{1}{||T||}\int_{t\in T}\vec{\mu}(t)dt \in Conv(A)$$

Proof. Suppose the result is true when the affine hull[10] of A has dimension less than or equal to $k-1$. The result is trivially true when $k-1=0$, as this implies $\vec{\mu}(t)$ is a single point for all time. We proceed by induction on k.

Assume the affine hull of A has dimension k. By a simple change of coordinates, we can equivalently treat $\vec{\mu}(t)$ as a function taking values in $\mathbb{R}^k$. Let $\vec{p} = \frac{1}{||T||}\int_{t\in T}\vec{\mu}(t)dt$. If the point $\vec{p}$ is within the set $Conv(A)$, we are done. If $\vec{p} \notin Conv(A)$, then by the convex set separation theorem there must exist a hyperplane H which separates $\vec{p}$ from $Conv(A)$, i.e., there exists a vector $\vec{z}$ and a scalar b such that

$$\begin{aligned} \vec{z}'\vec{p} &\leq b \\ \vec{z}'\vec{a} &\geq b \;\text{ for all } \vec{a} \in Conv(A) \end{aligned} \tag{4.57}$$

[10] The *affine hull* of a set A is the set $\vec{a} + X$, where $\vec{a}$ is an arbitrary element of A, and X is the smallest linear space such that $\vec{a} + X$ contains set A [15]. For example, consider a set of points within $\mathbb{R}^N$ which all lie on the same plane, or the same line. Then the affine hull is the 2-dimensional plane, or, respectively, the 1-dimensional line.

where the hyperplane H consists of all points $\vec{x} \in \mathbb{R}^k$ such that $\vec{z}'\vec{x} = b$. Thus, we have:

$$\begin{aligned} b &\geq \vec{z}'\vec{p} \\ &= \frac{1}{||T||}\int_{t\in T} \vec{z}'\vec{\mu}(t)dt \end{aligned} \tag{4.58}$$

However, $\vec{\mu}(t) \in Conv(A)$ for all time, and hence by (4.57) the integrand in (4.58) is greater than or equal to b for all time. This implies that the set of all times $t \in T$ for which $\vec{z}'\vec{\mu}(t) > b$ must have measure zero. Hence:

$$\begin{aligned} \vec{p} &= \frac{1}{||T||}\int_{t\in T} \vec{\mu}(t)dt \\ &= \frac{1}{||T||}\int_{\{t\in T|\, \vec{z}'\vec{\mu}(t)=b\}} \vec{\mu}(t)dt \end{aligned} \tag{4.59}$$

The integral in (4.59) represents the time average of a function contained in the set $A \cap H$, a set of dimension at most $k-1$. It follows by the induction hypothesis that $\vec{p} \in Conv(A \cap H) \subset Conv(A)$, a contradiction. □

Let $T = [0, x]$ for some interval size x.

Corollary 6. *If the set A is closed, then $\lim_{x\to\infty} \frac{1}{x}\int_0^x \vec{\mu}(t)dt \in Conv(A)$, provided that the limit converges.*

Proof. The limit can be approached arbitrarily closely by time average integrals over finite intervals. By Theorem 1, each such time average is contained within $Conv(A)$. The limiting integral is thus a limit point of the closed set $Conv(A)$, and hence is within $Conv(A)$. □

Example: The corollary does not hold if the set A is not closed. Indeed, consider the scalar valued function $\mu(t) = 1 - 1/(t+1)$ contained within the non-closed interval $[0, 1)$ for all $t \geq 0$. Then the time average integral of $\mu(t)$ over any finite interval is within $[0, 1)$, but the limiting average as the interval size $x \to \infty$ is equal to 1, which is not in this interval.

Chapter Appendix 4.C — Structural Properties of Γ and Λ

Here we prove that the graph family Γ is convex, and that the sets $\tilde{\Gamma}$ and Λ are compact and convex. The proofs use elementary facts about convexity and compactness of sets [100] [15].

Recall that:

$$\Gamma = \sum_{\underline{S}} \pi_{\underline{S}} \text{Convex\_Hull} \left\{ \underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi \right\}$$

and that $\tilde{\Gamma}$ is the set of all rate matrices entrywise less than or equal to some element of Γ.

Claim: The sets Γ and $\tilde{\Gamma}$ are convex.

Proof. The set Γ is a weighted sum of convex sets, and is therefore convex. To show that $\tilde{\Gamma}$ is convex, consider two points $\tilde{\gamma}_1, \tilde{\gamma}_2 \in \tilde{\Gamma}$. By definition, there must exist points $\gamma_1, \gamma_2 \in \Gamma$ such that $\gamma_1 \geq \tilde{\gamma}_1$ and $\gamma_2 \geq \tilde{\gamma}_2$. Thus, for any convex combination $q_1\tilde{\gamma}_1 + q_2\tilde{\gamma}_2$ (where q_1 and q_2 are nonnegative values summing to 1), we have:

$$q_1\tilde{\gamma}_1 + q_2\tilde{\gamma}_2 \leq q_1\gamma_1 + q_2\gamma_2 \in \Gamma$$

and hence $q_1\tilde{\gamma}_1 + q_2\tilde{\gamma}_2 \in \tilde{\Gamma}$. □

Claim: $\tilde{\Gamma}$ is compact, i.e., closed and bounded.

Proof. The fact that $\tilde{\Gamma}$ is bounded follows from boundedness of the set $\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ for each channel state $\underline{S}$. To see that the latter is bounded, first suppose that it is not. Then, there must exist a sequence of power matrices $\underline{P}_k \in \Pi$ such that $\underline{\mu}(\underline{P}_k, \underline{S}) \to \infty$ as $k \to \infty$. The infinite sequence of matrices $\underline{P}_k$ are within the compact set Π, and hence there must be a convergent subsequence $s(k)$ such that $\underline{P}_{s(k)} \to \underline{P}^*$ for some $\underline{P}^* \in \Pi$. By upper-semicontinuity, we have: $\infty = \lim_{k\to\infty} \underline{\mu}(\underline{P}_{s(k)}, \underline{S}) \leq \underline{\mu}(\underline{P}^*, \underline{S})$, a contradiction.

To show that $\tilde{\Gamma}$ is closed, we first observe that the "tilde" operator commutes through weighted sums and convex hull operations, that is, the set $\tilde{\Gamma}$ can be written as $\tilde{\Gamma} = \sum_{\underline{S}} \pi_{\underline{S}} \text{Conv}(\tilde{R}_{\underline{S}})$, where $\tilde{R}_{\underline{S}}$ is defined as the set of all rates entrywise less than or equal to some point in the set $\left\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\right\}$ (the proof of this fact is omitted for brevity). Next, we show that each set $\tilde{R}_{\underline{S}}$ is closed: For any limit point $\underline{\tilde{r}}$ of $\tilde{R}_{\underline{S}}$, we have a sequence of points $\underline{\tilde{r}}_k \in \tilde{R}_{\underline{S}}$ such that $\underline{\tilde{r}}_k \to \underline{\tilde{r}}$. For each k, we have $\underline{\tilde{r}}_k \leq \underline{\mu}(\underline{P}_k, \underline{S})$ for some $\underline{P}_k \in \Pi$.

The $\underline{P}_k$ values represent an infinite sequence of matrices within the compact set Π, and hence there must be a convergent subsequence $s(k)$ such that $\lim_{k\to\infty} \underline{P}_{s(k)} = \underline{P}^*$, where $\underline{P}^* \in \Pi$. We thus have:

$$\underline{\tilde{r}} \;=\; \lim_{k\to\infty} \underline{\tilde{r}}_{s(k)} \leq \lim_{k\to\infty} \underline{\mu}(\underline{P}_{s(k)}, \underline{S}) \leq \underline{\mu}(\underline{P}^*, \underline{S})$$

where the last inequality follows by upper-semicontinuity of the function $\underline{\mu}(\underline{P}, \underline{S})$ with respect to the power matrix $\underline{P}$. It follows that $\underline{\tilde{r}} \in \tilde{R}_{\underline{S}}$, so that $\tilde{R}_{\underline{S}}$ is closed. Because the convex hull of a closed set is closed, as is the weighted sum of closed sets, it follows that $\tilde{\Gamma}$ is closed. □

Claim: The capacity region Λ is convex and compact.

Proof. The capacity region Λ is the set of all input rate matrices (λ_{ic}) such that there exist multi-commodity flow variables $\{f_{ab}^{(c)}\}$ satisfying:

$$\begin{aligned}
f_{ab}^{(c)} &\geq 0 && \forall a, b, c \\
f_{aa}^{(c)} = f_{ab}^{(a)} &= 0 && \forall a, b, c \\
\lambda_{ic} \leq \sum_b f_{ib}^{(c)} &- \sum_a f_{ai}^{(c)} && \forall i, c \text{ such that } i \neq c \\
\left(\sum_c f_{ab}^{(c)}\right) &\in \tilde{\Gamma} && \forall a, b
\end{aligned}$$

It is clear that given two such rate matrices (λ_{ic}^1) and (λ_{ic}^2), the convex combination of these matrices satisfies the above constraints for flows which are equal to the convex combination of the flows for each individual matrix (note that $\tilde{\Gamma}$ is convex). Thus, Λ is convex.

Boundedness of the set Λ follows by boundedness of the set Γ, so that clearly Λ is contained in the set of all rate matrices (λ_{ic}) such that $\lambda_{ic} \leq \mu_{max}^{out}$ for all (i, c).

To show closedness, note that the λ_{ic} and $f_{ab}^{(c)}$ variables can be compiled as a stacked vector $\vec{v}$, so that the inequalities specifying the capacity region Λ can be written as $A\vec{v} \in \Omega$ for some matrix A and for a closed set Ω.[11] The set of all vectors $\vec{v}$ satisfying this constraint is the inverse image of a closed set through a linear map, and is hence closed. It follows that Λ is closed. □

[11] The Ω set is closed because of the (non-strict) inequality constraints and because of the closedness of $\tilde{\Gamma}$.

Chapter Appendix 4.D — Graph Family Achievability

He we prove Lemma 8, which shows that power can be allocated according to a stationary randomized rule to achieve any long-term link transmission rate matrix within the network graph family Γ.

(Graph Family Achievability) Let (G_{ab}) *be a matrix within the graph family* Γ *(defined in (4.3)), so that*

$$\sum_{\underline{S}} \pi_{\underline{S}} \underline{G}_{\underline{S}} = (G_{ab})$$

for some matrices $\underline{G}_{\underline{S}}$ *within Convex_Hull*$\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$. *Then:*

(a) A stationary randomized *power allocation policy* $\underline{P}_{STAT}(\tau)$ *can be implemented which yields a transmission rate process* $\underline{\mu}^{STAT}(t) \triangleq \underline{\mu}(\underline{P}_{STAT}(t), \underline{S}(t))$ *which is entrywise rate convergent with rate matrix* (G_{ab}), *and which satisfies the convergence bounds given in (4.9) and (4.10).*

Specifically, every timeslot in which the channel state $\underline{S}$ *is observed, the power matrix* $\underline{P}_{STAT}(\tau)$ *is chosen randomly from a finite set of m allocations* $\{\underline{P}^1_{\underline{S}}, \ldots, \underline{P}^m_{\underline{S}}\}$ *according to probabilities* $\{q^1_{\underline{S}}, \ldots, q^m_{\underline{S}}\}$.

(b) If the set $\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ *is convex for every channel state* $\underline{S}$, *then the power control algorithm can be implemented by a non-randomized policy, where a fixed power matrix* $\underline{P}_{\underline{S}}$ *is allocated whenever in channel state* $\underline{S}$.

Proof. (a) By Caratheodory's Theorem [15], any point $\underline{G}_{\underline{S}}$ in the convex hull of the set $\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ can be expressed as a finite combination of matrices:

$$\underline{G}_{\underline{S}} = q_1^{\underline{S}} \underline{G}_1^{\underline{S}} + \ldots + q_k^{\underline{S}} \underline{G}_k^{\underline{S}}$$

where the $\{q_i^{\underline{S}}\}$ values are nonnegative numbers that sum to 1 and represent probabilities for the randomized algorithm, and $\underline{G}_i^{\underline{S}} \in \{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ for each i. Choosing power allocations $\{\underline{P}_1^{\underline{S}}, \ldots, \underline{P}_k^{\underline{S}}\}$ such that $\underline{\mu}(\underline{P}_i^{\underline{S}}, \underline{S}) = \underline{G}_i^{\underline{S}}$ and allocating power according to the randomized policy ensures $\mathbb{E}\left\{\underline{\mu}(\underline{P}(\tau), \underline{S}(\tau)) \mid \underline{S}(\tau) = \underline{S}\right\} = \underline{G}_{\underline{S}}$ (where the expectation is taken over the $\{q_i^{\underline{S}}\}$ probabilities). By Lemma 4 of Chapter 2, the resulting rates $\mu_{ab}(t)$ are rate convergent with rates G_{ab} for each link (a, b). Hence, the sum processes $\sum_b \mu_{ib}(t)$ and $\sum_a \mu_{ai}(t)$ are rate convergent with rates $\sum_b G_{ib}$ and $\sum_a G_{ai}$, respectively. By Lemma 4 of Chapter 2 together with the channel convergent bounds of (4.1) and (4.2) for K and δ, it follows that

$$\left|\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left[\sum_b \mathbb{E}\left\{\mu_{ib}(\tau)\right\}\right] - \sum_b G_{ib}\right| \leq \mu_{max}^{out}\frac{\delta}{\max\{\mu_{max}^{out}, \mu_{max}^{in}\}} \leq \delta$$

$$\left|\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left[\sum_a \mathbb{E}\left\{\mu_{ai}(\tau)\right\}\right] - \sum_a G_{ai}\right| \leq \mu_{max}^{in}\frac{\delta}{\max\{\mu_{max}^{out}, \mu_{max}^{in}\}} \leq \delta$$

Part (b) follows because each $\underline{G_{\underline{S}}}$ matrix is defined to be within Convex_Hull$\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$ and hence must also be within $\{\underline{\mu}(\underline{P}, \underline{S}) \mid \underline{P} \in \Pi\}$. Thus, for each channel state $\underline{S}$, there is a single power matrix $\underline{P^{\underline{S}}}$ for which $\underline{G_{\underline{S}}} = \underline{\mu}(\underline{P^{\underline{S}}}, \underline{S})$. □

Chapter Appendix 4.E — Bound on Lyapunov Drift

Here we derive the Lyapunov drift bound of (4.22) used in the proof of Theorem 6 in Section 4.3.2.

Consider the Lyapunov function $L(\underline{U}) = \sum_{i\neq j}[U_i^{(c)}]^2$. For any power allocation policy which determines rate processes $\mu_{ij}^{(c)}(t)$, we have [from (4.28)]:

$$\begin{aligned}\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0)) \,|\, \underline{U}(t_0)\right\} \leq K^2NB + \\ -2K\textstyle\sum_{i\neq c}U_i^{(c)}(t_0)\mathbb{E}\left\{\sum_b \tilde{\mu}_{ib}^{(c)} - \sum_a \tilde{\mu}_{ai}^{(c)} - \tilde{A}_{ic} \,|\, \underline{U}(t_0)\right\}\end{aligned}$$

where:

$$\begin{aligned}\tilde{\mu}_{ij}^{(c)} &\triangleq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mu_{ab}^{(c)}(\tau) && (4.60)\\ \tilde{A}_{ic} &\triangleq \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}A_{ic}(\tau) && (4.61)\\ B &\triangleq \left(\mu_{max}^{in} + A_{max}\right)^2 + (\mu_{max}^{out})^2\end{aligned}$$

Proof. From (4.21), we have that the K-step dynamics of unfinished work satisfies the following bound for all $i \neq c$:

$$U_i^{(c)}(t_0+K) \leq \max\left(U_i^{(c)}(t_0) - \sum_b K\tilde{\mu}_{ib}^{(c)}, 0\right) + \sum_a K\tilde{\mu}_{ai}^{(c)} + K\tilde{A}_{ic}$$

Squaring both sides and noting that $\max^2(x,0) \leq x^2$, we have:

$$\begin{aligned}[U_i^{(c)}(t_0+K)]^2 - [U_i^{(c)}(t_0)]^2 \leq \;& K^2\left[\tilde{A}_{ic}^2 + 2\tilde{A}_{ic}\left(\textstyle\sum_a \tilde{\mu}_{ai}^{(c)}\right) + \left(\sum_a \tilde{\mu}_{ai}^{(c)}\right)^2\right] + K^2\left(\textstyle\sum_b \tilde{\mu}_{ib}^{(c)}\right)^2\\ & -2KU_i^{(c)}(t_0)\left[\textstyle\sum_b \tilde{\mu}_{ib}^{(c)} - \sum_a \tilde{\mu}_{ai}^{(c)} - \tilde{A}_{ic}\right] \qquad (4.62)\end{aligned}$$

An expression for Lyapunov drift is obtained from the above inequality by summing over all nodes i and commodities $c \neq i$ and taking conditional expectations. The following inequalities simplify the resulting expression and follow from Jensen's inequality together with the fact that the sum of squares of positive numbers is less than or equal to the square of the sum [see derivation of Claim 1 below]:

$$\mathbb{E}\left\{\sum_{i\neq c}\left(\sum_b \tilde{\mu}_{ib}^{(c)}\right)^2 \,|\, \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{out}\right)^2 \;\;,\;\; \mathbb{E}\left\{\sum_{i\neq c}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right)^2 \,|\, \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{in}\right)^2$$
$$\mathbb{E}\left\{\sum_{i\neq c}\tilde{A}_{ic}^2 \,|\, \underline{U}(t_0)\right\} \leq NA_{max}^2 \;\;,\;\; 2\mathbb{E}\left\{\sum_{i\neq c}\tilde{A}_{ic}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right) \,|\, \underline{U}(t_0)\right\} \leq 2N\left(\mu_{max}^{in}\right)A_{max}$$

Summing (4.62) over $i \neq j$ and using the above inequalities leads to:

$$\begin{aligned}\mathbb{E}\left\{L(\underline{U}(t_0+K)) - L(\underline{U}(t_0)) \,|\, \underline{U}(t_0)\right\} \leq K^2N\left(\left(\mu_{max}^{out}\right)^2 + \left(\mu_{max}^{in}\right)^2 + A_{max}^2 + 2\left(\mu_{max}^{in}\right)A_{max}\right) \\ -2K\sum_{i\neq c}U_i^{(c)}(t_0)\mathbb{E}\left\{\sum_b \tilde{\mu}_{ib}^{(c)} - \sum_a \tilde{\mu}_{ai}^{(c)} - \tilde{A}_{ic} \,|\, \underline{U}(t_0)\right\}\end{aligned}$$

□

Claim 1: (Derivation of bounds used in the proof above)

$$\mathbb{E}\left\{\sum_{i\neq c}\left(\sum_b \tilde{\mu}_{ib}^{(c)}\right)^2 \,|\, \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{out}\right)^2 \;\;,\;\; \mathbb{E}\left\{\sum_{i\neq c}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right)^2 \,|\, \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{in}\right)^2$$
$$\mathbb{E}\left\{\sum_{i\neq c}\tilde{A}_{ic}^2 \,|\, \underline{U}(t_0)\right\} \leq NA_{max}^2 \;\;,\;\; 2\mathbb{E}\left\{\sum_{i\neq c}\tilde{A}_{ic}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right) \,|\, \underline{U}(t_0)\right\} \leq 2N\left(\mu_{max}^{in}\right)A_{max}$$

where $\tilde{\mu}_{ij}^{(c)}$ and $\tilde{A}_{ij}$ are defined in (4.61) and (4.60).

Proof. We prove only the first and last inequalities. (The second and third are similar to the first inequality as well as to the inequalities proved in Chapter Appendix 3.A).

To show that $\mathbb{E}\left\{\sum_{i\neq c}\left(\sum_b \tilde{\mu}_{ib}^{(c)}\right)^2 \,|\, \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{out}\right)^2$, first note that $\tilde{\mu}_{ib}^{(i)} = 0$ always, so that the "$i \neq c$" condition can be neglected. We thus have:

$$\begin{aligned}\sum_i\sum_c\left(\sum_b \tilde{\mu}_{ib}^{(c)}\right)^2 &\leq \sum_i\left(\sum_b\sum_c \tilde{\mu}_{ib}^{(c)}\right)^2 &(4.63)\\ &= \sum_i\left(\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left(\sum_b\sum_c \mu_{ib}^{(c)}(\tau)\right)\right)^2 &(4.64)\\ &\leq \sum_i\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left(\sum_b\sum_c \mu_{ib}^{(c)}(\tau)\right)^2 &(4.65)\\ &\leq \sum_i\frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\left(\mu_{max}^{out}\right)^2 &(4.66)\\ &= N\left(\mu_{max}^{out}\right)^2\end{aligned}$$

where (4.63) follows because the sum of squares of positive numbers is less than or equal to the square of the sum, and (4.65) follows from Jensen's inequality (noting that the function x^2 is convex). $\square$

To show that $\mathbb{E}\left\{\sum_{i\neq c}\tilde{A}_{ic}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right) \mid \underline{U}(t_0)\right\} \leq N\left(\mu_{max}^{in}\right)A_{max}$, we note that $\tilde{A}_{ii} = 0$ for all i, so that the $i \neq c$ constraint can again be neglected. We thus have:

$$\begin{aligned}
\sum_{i\neq c}\tilde{A}_{ic}\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right) &= \sum_i\sum_c\left(\tilde{A}_{ic}\right)\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right) \\
&\leq \sum_i\sqrt{\sum_c(\tilde{A}_{ic}^2)\sum_c\left(\sum_a \tilde{\mu}_{ai}^{(c)}\right)^2} \qquad (4.67) \\
&\leq \sum_i\sqrt{\left(\sum_c \tilde{A}_{ic}\right)^2\left(\sum_a\sum_c \tilde{\mu}_{ai}^{(c)}\right)^2} \\
&\leq \mu_{max}^{in}\sum_i\left(\sum_c \tilde{A}_{ic}\right)
\end{aligned}$$

where (4.67) follows by the Cauchy-Schwartz inequality (or "inner-product" inequality) for sums. Taking conditional expectations and noting that

$$\mathbb{E}\left\{\sum_c \tilde{A}_{ic}(\tau)\,|\underline{U}(t_0)\right\} \leq \sqrt{\mathbb{E}\left\{\left(\sum_c \tilde{A}_{ic}(\tau)\right)^2 |\underline{U}(t_0)\right\}} \leq A_{max}$$

yields the result. $\square$

Chapter Appendix 4.F — Performance of FRAME algorithm

Here we prove Lemmas 9 and 10, which establish that the frame based algorithm maximizes the drift term $\Phi\left(\underline{U}(t_0)\right)$ over all possible network control strategies, and that the DRPC algorithm yields a $\Phi\left(\underline{U}(t_0)\right)$ value which is no more than a fixed distance away from the maximum.

Recall that:

$$\Phi\left(\underline{U}(t_0)\right) \triangleq \frac{1}{K} \sum_{\tau=t_0}^{t_0+K-1} \mathbb{E}\left\{ \sum_{i \neq c} U_i^{(c)}(t_0) \left[\sum_b \mu_{ib}^{(c)}(\tau) - \sum_a \mu_{ai}^{(c)}(\tau) \right] \mid \underline{U}(t_0) \right\} \tag{4.68}$$

Also note that any power control and routing strategy allocates a power matrix $\underline{P}(\tau)$ every timeslot subject to $\underline{P}(\tau) \in \Pi$, and determines transmission rates $\mu_{ab}^{(c)}(\tau)$ for each commodity and each link (a,b) subject to:

$$\sum_c \mu_{ab}^{(c)}(\tau) \leq \mu_{ab}(\underline{P}(\tau), \underline{S}(\tau)) \tag{4.69}$$

The FRAME algorithm acts the same as the DRPC algorithm with the exception that queue backlog is updated only on frame boundaries $t_0 = \{0, K, 2K, \ldots\}$. Thus, for every $\tau \in \{t_0, \ldots, t_0 + K - 1\}$, the algorithm FRAME allocates a power matrix $\underline{P}^{FRAME}(\tau)$ to maximize:

$$\sum_{ab} \mu_{ab}(\underline{P}, \underline{S}(\tau)) W_{ab}^*(t_0)$$

where:

$$W_{ab}^*(t_0) = \max[U_a^{(c_{ab}^*(t_0))}(t_0) - U_b^{(c_{ab}^*(t_0))}(t_0), 0]$$

$$c_{ab}^*(t_0) = \underset{c \in \{1,\ldots,N\}}{\arg\max} \left\{ U_a^{(c)}(t_0) - U_b^{(c)}(t_0) \right\}$$

and hence yields $\mu_{ab}^{(c)}(\tau)$ values satisfying:

$$\mu_{ab}^{(c)FRAME}(\tau) = \begin{cases} \mu_{ab}\left(\underline{P}^{FRAME}(\tau), \underline{S}(\tau)\right) & \text{if } c = c_{ab}^*(t_0) \text{ and } W_{ab}^*(t_0) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.70}$$

Claim (Lemma 9): The control algorithm FRAME maximizes $\Phi\left(\underline{U}(t_0)\right)$ over all possible power allocation, routing, and scheduling strategies. That is:

$$\Phi^{FRAME}\left(\underline{U}(t_0)\right) \geq \Phi^{X}\left(\underline{U}(t_0)\right)$$

for any other strategy X, including strategies that have full knowledge of arrival and channel statistics.

Proof. First note that $U_i^{(i)}(\tau) = 0$ for all time, and hence the $i \neq c$ condition in the sum of (4.68) can be removed. Furthermore, by switching the order of summation, we have the following identity:

$$\sum_{i,c} U_i^{(c)}(t_0)\left[\sum_b \mu_{ib}^{(c)}(\tau) - \sum_a \mu_{ai}^{(c)}(\tau)\right] = \sum_{ab}\sum_c \mu_{ab}^{(c)}(\tau)\left[U_a^{(c)}(t_0) - U_b^{(c)}(t_0)\right] \qquad (4.71)$$

Taking conditional expectations above and summing over τ yields an alternative way to express $\Phi\left(\underline{U}(t_0)\right)$:

$$\Phi\left(\underline{U}(t_0)\right) = \frac{1}{K}\sum_{\tau=t_0}^{t_0+K-1}\mathbb{E}\left\{\sum_{ab}\sum_c \mu_{ab}^{(c)}(\tau)\left[U_a^{(c)}(t_0) - U_b^{(c)}(t_0)\right] \mid \underline{U}(t_0)\right\} \qquad (4.72)$$

The value of $\Phi^X\left(\underline{U}(t_0)\right)$ is obtained from (4.72) by using the $\mu_{ab}^{(c)}(\tau)$ values corresponding to some policy X, and the value of $\Phi^{FRAME}\left(\underline{U}(t_0)\right)$ is obtained by using the $\mu_{ab}^{(c)FRAME}(\tau)$ values associated with the frame based control scheme. However, for any general power allocation and routing scheme using power $\underline{P}(\tau)$ yielding rate values $\mu_{ab}^{(c)}(\tau)$, we have for every slot τ:

$$\begin{aligned}
\sum_{ab}\sum_c \mu_{ab}^{(c)}(\tau)\left[U_a^{(c)}(t_0) - U_b^{(c)}(t_0)\right] &\leq \sum_{ab}\sum_c \mu_{ab}^{(c)}(\tau) W_{ab}^*(t_0) && (4.73)\\
&\leq \sum_{ab} \mu_{ab}(\underline{P}(\tau), \underline{S}(\tau)) W_{ab}^*(t_0) && (4.74)\\
&\leq \sum_{ab} \mu_{ab}(\underline{P}^{FRAME}, \underline{S}(\tau)) W_{ab}^*(t_0) && (4.75)\\
&= \sum_{ab}\sum_c \mu_{ab}^{(c)FRAME}(\tau)\left[U_a^{(c)}(t_0) - U_b^{(c)}(t_0)\right] && (4.76)
\end{aligned}$$

where (4.73) follows by definition of $W_{ab}^*(t_0)$ and non-negativity of the $\mu_{ab}^{(c)}(\tau)$ values, (4.74) follows from (4.69), (4.75) holds because, by definition, the FRAME strategy allocates power to maximize the quantity in (4.74) among all policies conforming to the power constraints, and (4.76) holds by definition of $\mu_{ab}^{(c)FRAME}(\tau)$ in (4.70).

Taking conditional expectations of (4.76) and summing over $\tau \in \{0, 1, \ldots, K-1\}$ demonstrates that the FRAME algorithm maximizes $\Phi\left(\underline{U}(t_0)\right)$ over any other power allocation strategy, proving the claim. □

We now compare the algorithms FRAME and DRPC.

Claim (Lemma 10): $\Phi^{DRPC}\left(\underline{U}(t_0)\right) \geq \Phi^{FRAME}\left(\underline{U}(t_0)\right) - (K-1)N\tilde{B}/2$
where

$$\tilde{B} \triangleq 2(\mu_{max}^{in} + \mu_{max}^{out})\left(A_{max} + \mu_{max}^{in} + \mu_{max}^{out}\right)$$

Proof. Consider an implementation of the DRPC algorithm, and let $\underline{U}(t_0)$ represent the unfinished work matrix at the start of a frame, and let $\underline{U}(\tau)$ represent the unfinished work at some time τ during the frame $\{t_0, \ldots, t_0 + K - 1\}$. At any such time τ, the DRPC algorithm selects transmission rates $\mu_{ab}^{(c)}(\tau)$ that maximize $\sum_{a,b,c} \mu_{ab}^{(c)}(\tau)[U_a^{(c)}(\tau) - U_b^{(c)}(\tau)]$ over all other possible control decisions. Hence:

$$\sum_{a,b,c} \mu_{ab}^{(c)DRPC}(\tau)\left[U_a^{(c)}(\tau) - U_b^{(c)}(\tau)\right] \geq \sum_{a,b,c} \mu_{ab}^{(c)FRAME}(\tau)\left[U_a^{(c)}(\tau) - U_b^{(c)}(\tau)\right]$$

where the values $\mu_{ab}^{(c)FRAME}(\tau)$ represent the control decisions that would be made by the FRAME algorithm at time τ if the backlog matrix at time t_0 were $\underline{U}(t_0)$. Using (4.71) to switch the summation, we have:

$$\sum_{i,c} U_i^{(c)}(\tau)\left[\sum_b \mu_{ib}^{(c)DRPC}(\tau) - \sum_a \mu_{ai}^{(c)DRPC}(\tau)\right] \geq$$
$$\sum_{i,c} U_i^{(c)}(\tau)\left[\sum_b \mu_{ib}^{(c)FRAME}(\tau) - \sum_a \mu_{ai}^{(c)FRAME}(\tau)\right]$$

Defining $\Delta_i^{(c)}(\tau) \triangleq U_i^{(c)}(\tau) - U_i^{(c)}(t_0)$ and noting that $\Delta_i^{(c)}(t_0) = 0$, it follows that:

$$\sum_{i,c} U_i^{(c)}(t_0)\left[\sum_b \mu_{ib}^{(c)DRPC}(\tau) - \sum_a \mu_{ai}^{(c)DRPC}(\tau)\right] + \sum_{i,c}\left|\Delta_i^{(c)}(\tau)\right|(\mu_{max}^{in} + \mu_{max}^{out}) \geq$$
$$\sum_{i,c} U_i^{(c)}(t_0)\left[\sum_b \tilde{\mu}_{ib}^{(c)FRAME}(\tau) - \sum_a \tilde{\mu}_{ai}^{(c)FRAME}(\tau)\right] - \sum_{i,c}\left|\Delta_i^{(c)}(\tau)\right|(\mu_{max}^{in} + \mu_{max}^{out}) \quad (4.77)$$

where we used the fact that:

$$\sum_{i,c} \Delta_i^{(c)}(\tau) \left[\sum_b \mu_{ib}^{(c)}(\tau) - \sum_a \mu_{ai}^{(c)}(\tau) \right] \leq \sum_{i,c} \left| \Delta_i^{(c)} \right| (\mu_{max}^{out} + \mu_{max}^{in})$$

Summing (4.77) over $\tau \in \{t_0, \ldots, t_0 + K - 1\}$ and taking conditional expectations yields:

$$\Phi^{DRPC}\left(\underline{U}(t_0)\right) \quad \geq \quad \Phi^{FRAME}\left(\underline{U}(t_0)\right) - \frac{2}{K} \sum_{\tau=t_0+1}^{t_0+K-1} \mathbb{E}\left\{ \sum_{i,c} \left| \Delta_i^{(c)}(\tau) \right| \right\} \left(\mu_{max}^{in} + \mu_{max}^{out} \right)$$

The expected change in unfinished work from time t_0 to time τ is at most $(\mu_{max}^{out} + \mu_{max}^{in} + A_{max})(\tau - t_0)$ at any node, which leads to

$$\Phi^{DRPC}\left(\underline{U}(t_0)\right) \quad \geq \quad \Phi^{FRAME}\left(\underline{U}(t_0)\right) - \frac{2}{K} \sum_{\tau=t_0+1}^{t_0+K-1} N(\tau - t_0)(\mu_{max}^{out} + \mu_{max}^{in} + A_{max}) \left(\mu_{max}^{in} + \mu_{max}^{out} \right)$$

Changing variables to $v \overset{\triangle}{=} \tau - t_0$ and using the fact that $\sum_{v=1}^{K-1} v = \frac{K(K-1)}{2}$, we have:

$$\Phi^{DRPC}\left(\underline{U}(t_0)\right) \geq \Phi^{FRAME}\left(\underline{U}(t_0)\right) - (K-1)N\tilde{B}/2$$

□

Chapter 5

Network Fairness and Control Beyond Capacity

In the previous chapter we described the network capacity region Λ and constructed a Dynamic Routing and Power Control algorithm (DRPC) for stabilizing the network whenever the input rate matrix is strictly interior to this region. This situation is illustrated in Fig. 5-1, where the rate matrix $\underline{\lambda}^{(1)}$ is shown to be strictly interior to Λ. In this chapter, we address the problem of network control when the rate matrix is *outside* of the capacity region, as illustrated by the $\underline{\lambda}^{(2)}$ matrix in Fig. 5-1. In this case, it is not possible to stabilize the system by serving all of the data, and hence flow control decisions must be made concerning the amount of data to be served from each stream. Given input streams with a rate matrix $\underline{\lambda} = (\lambda_{ic})$ outside of the capacity region, we must find substreams with a rate matrix $\underline{r} = (r_{ic})$ such that:

1. $\underline{r} \in \Lambda$ (because $\underline{r}$ must be supportable by the network)

2. $r_{ic} \leq \lambda_{ic}$ for all (i, c) (because the r_{ic} rate represents a fraction of the total rate λ_{ic} entering the network at node i and destined for node c)

The intersection of the capacity region Λ and the set $\{\underline{r} \mid \underline{r} \leq \underline{\lambda}\}$ is illustrated in Fig. 5-1. We would like to operate the network so that $\underline{r}$ is on or near the boundary of this intersection, with the precise point being chosen according to some notion of fairness. Throughout this chapter, we assume that each user i has a set of utility functions $g_{ic}(r)$ representing a quantitative measure of the "satisfaction" (or "goodness") that user i receives from deliv-

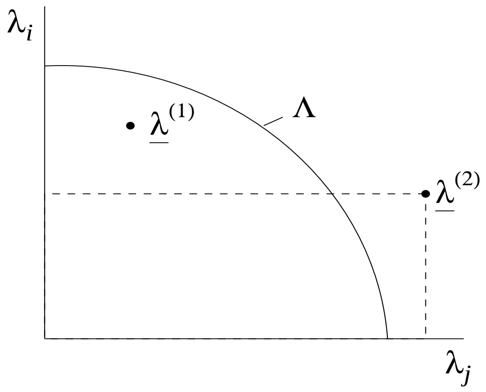


Figure 5-1: A capacity region Λ (illustrated in 2 dimensions) with a rate matrix $\underline{\lambda}^{(1)}$ strictly in the interior. The rate matrix $\underline{\lambda}^{(2)}$ is outside of the capacity region.

ering data from source i to destination c at a rate of r bits per slot. We define an *optimally fair* rate point to be one which maximizes the sum of utilities over all users. Note that if $g_{ic}(r) = \theta_{ic} r$, then the optimally fair point maximizes a weighted sum of throughput, while if $g_{ic}(r) = \log(r)$ then the optimally fair point leads to *proportional fairness* [76].

Thus, given a capacity region Λ and an initial set of exogenous rates $\underline{\lambda}$, we have the following optimization problem:

$$\begin{array}{ll} \text{Maximize :} & \sum_{ic} g_{ic}(r_{ic}) \\ \text{Subject to:} & \underline{r} \in \Lambda \\ & \underline{r} \leq \lambda \end{array} \tag{5.1}$$

Assuming that each utility function is monotonically increasing, the solution of the above optimization is given by $\underline{r} = \underline{\lambda}$ whenever $\underline{\lambda} \in \Lambda$. If $\underline{\lambda} \notin \Lambda$, the optimal $\underline{r}$ vector must have at least one entry r_{ic} strictly less than the corresponding entry λ_{ic}. Such an optimization could in principle be performed if the capacity region Λ and the input rates $\underline{\lambda}$ were known in advance,[1] and all users could coordinate by sending data according to this optimization. However, the capacity region depends on the channel dynamics, which are unknown to the network controllers and to the individual users. Furthermore, the individual users do not know the data rates of other users, and may not even know their own data rates (as these are generated by potentially bursty applications).

[1] Note that a test to determine if $\underline{\lambda} \in \Lambda$ can be performed by carrying out a deterministic version of the DRPC algorithm, as described in Chapter 4.7.

In this chapter, we develop a dynamic control algorithm that uses only knowledge of the current network state to deliver data to the network while jointly making optimal decisions about routing, scheduling, and power allocation. The algorithm operates across two different layers: The *transport layer* (where data is generated based on the specific applications or functions of each node) and the *network layer* (where data is transferred from node to node to its destination). At the network layer, routing and power allocation decisions are made according to the DRPC algorithm of Chapter 4. At the transport layer, data is delivered to the network according to a simple flow control scheme, where the control actions of individual users are decoupled from each other and from the routing and power allocation decisions. The throughput and delay characteristics of the algorithm are similar to those of DRPC when the exogenous data rates are within the capacity region. If the rates are not within the capacity region, the algorithm yields a resulting set of data rates $\underline{r} = (r_{ic})$ which are arbitrarily close to the optimal solution of (5.1). The distance between these solutions is shown to decrease like $1/V$, where V is a control parameter affecting a tradeoff in average delay for data that is served by the network.

5.1 Related Work

Recent work on optimization and fair control for wireless systems is found in [139] [20] [86] [12] [93] [70]. In [86], an optimization problem similar to (5.1) is presented for a static wireless downlink, and pricing schemes are developed to enable power allocations to converge to a fair allocation vector. A similar static optimization problem is constructed in [70] for wireless networks, where geometric programming is used to establish a fair operating point with respect to various utility metrics. Utility based scheduling is further treated in [12] for wireless downlinks.

Dynamic server scheduling in a wireless downlink with input rates that exceed the system capacity is considered in [139]. It is shown that a modified version of the *Serve-the-Longest-Queue* policy maximizes throughput in systems with deterministic channel and traffic characteristics. A fair scheduling algorithm is developed in [20] for a wireless downlink with randomly varying channels, where it is assumed that every packet arriving to the downlink is destined for a distinct user with its own channel. Under symmetric conditions where all channels have identical rate fluctuation statistics relative to their mean, it is shown that serving the user with the largest relative fluctuation above its mean provides each user a fair fraction of the maximum output rate of the downlink. This algorithm requires knowledge of the mean channel rates observed by each user, and is particularly sensitive to the symmetric structure of the problem. Indeed, the algorithm deviates from fair performance in cases of asymmetry, or in cases where a stream of packets are to be transmitted over the same channel.

Optimization approaches to address fairness and utility maximization for static flow networks are considered in [78] [76] [92] [69] [121]. In [78] [76], pricing mechanisms are designed to enable distributed resource allocation in a network. Under this approach, an appropriate utility maximization problem is constructed and solved via the theory of convex duality, where the Lagrange multipliers of the dual problem correspond to "shadow prices" charged by the network. Recent applications of this approach to the area of congestion control and "Fast TCP" for internet traffic are developed in [91] [114] [67]. In [91], congestion control algorithms such as TCP Reno and TCP Vegas are shown to have an interpretation as an approximate primal-dual algorithm in a suitable convex optimization problem, and improved TCP algorithms based on this interpretation are presented in [67]. A similar analysis is used in [114] to construct feedback algorithms for congestion control in a network with fixed routing and linearized dynamics. All of these optimization approaches treat static networks using fluid flow approximations, so that discrete control decisions (such as randomly marking a packet in a TCP Reno implementation) can be viewed in the framework of a deterministic feedback function affecting a fluid rate parameter.

In this chapter, we treat dynamic control for wireless networks with randomly varying channels and randomly arriving packets. We develop a joint optimal strategy for flow control, routing, and power allocation. Our problem formulation and analytical approach is significantly different from all related work in this area, and our results represent a significant contribution to the theory of dynamic network optimization.

5.2 DRPC with Flow Control

We treat the same network as in Chapter 4 (see Fig. 4-1), where channels dynamically vary every slot according to a process $\underline{S}(t)$ with channel probabilities $\pi_{\underline{S}}$. Data streams are generated by applications running at each node according to arrival processes $A_{ic}(t)$ with rates λ_{ic}. For simplicity of exposition, we assume the channel and arrival processes are i.i.d. every timeslot.[2] The network itself is characterized by rate-power functions $\underline{\mu}(\underline{P}, \underline{S})$ that are assumed to be upper semi-continuous, but are otherwise arbitrary. The power matrix $\underline{P}(t)$ is constrained to be within a compact set Π every timeslot.

5.2.1 The flow control valve

Data generated by applications at each node are not delivered immediately to the output queues of that node. Rather, all data from a given node first enters a storage reservoir, as shown in Fig. 5-2. Let $L_{ic}(t)$ represent the current backlog in reservoir (i, c). A valve controls the amount of data from each commodity that drains from the reservoir into the network. Let $R_{ic}(t)$ represent the amount of commodity c bits chosen by the control valve at source node i to be delivered to the network at slot t (where $R_{ic}(t) \leq L_{ic}(t)$). Note that the $R_{ic}(t)$ processes represent the arrival streams received at the network layer.

The backlog $L_{ic}(t)$ in each reservoir evolves according to the following queueing equation:

$$L_{ic}(t) = L_{ic}(t-1) - R_{ic}(t-1) + A_{ic}(t) \tag{5.2}$$

In particular, the reservoir size at any slot t is always at least as large as the amount of bits $A_{ic}(t)$ that arrive during this slot. We emphasize that these storage reservoirs are *outside* of the network, and in particular the reservoir backlog $L_{ic}(t)$ is distinct from the network backlog $U_i^{(c)}(t)$, which (as before) represents the unfinished work of commodity c currently held in network node i.

The repeated use of valve control determines the long run data rates r_{ic} allowed into the network, where $r_{ic} \triangleq \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} R_{ic}(\tau)$. We desire the network itself to be stable, so that all data delivered to the network eventually reaches its destination. Hence, the rates (r_{ic}) must be within the network capacity region Λ. The reservoirs themselves may be "unstable," necessarily growing to infinity in cases where the original input matrix (λ_{ic}) is outside of the capacity region. We do not concern ourselves with such instability, and view the situation of an unstable reservoir as one in which there is always data waiting to be sent.

[2] The situation of i.i.d. arrivals and channels provides all of the intuition needed to treat general rate convergent and channel convergent processes, as made evident by the analysis in Chapter 4. In particular, system dynamics can be analyzed every K slots, where K is an integer chosen so that empirical averages over disjoint intervals of size K are nearly i.i.d., where "nearness" is determined by a parameter ϵ.

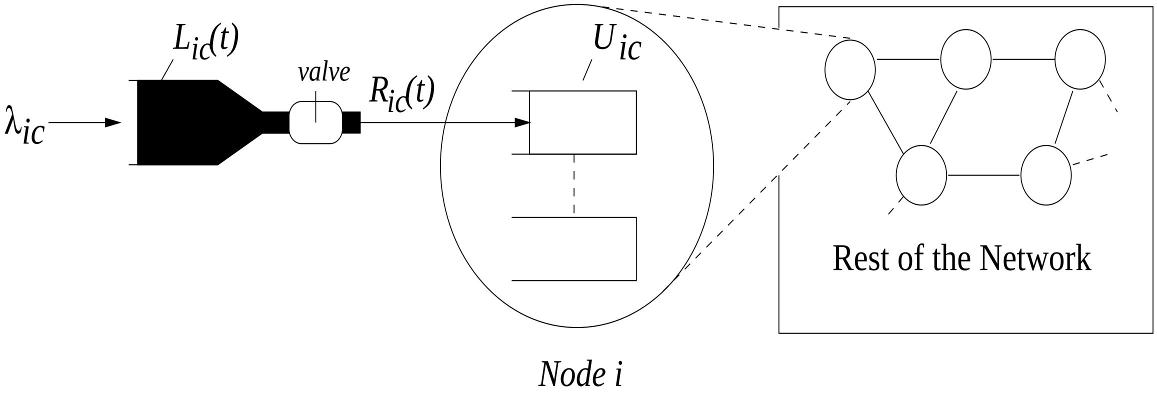


Figure 5-2: A reservoir for storing data from the $A_{ic}(t)$ stream, together with a timeslotted flow control valve that selects an amount of data to deliver to the network on every timeslot.

We consider the class of all possible valve control policies, coupled with all possible routing, scheduling, and power allocation policies in the network. This includes all cross-layer policies which make optimal decisions based on full cooperation and full knowledge of future events (such as link failures, channel variations, or arrival bursts). We design a decoupled algorithm which has no knowledge of the future yet yields an overall utility value which is arbitrarily close to that of any other (possibly anticipatory) policy.

5.2.2 A Cross-Layer Algorithm and Joint Optimality

In order to limit the burstiness of data delivered to the network layer, we define R_{max} as the maximum number of exogenous bits allowed into the network each timeslot at any particular node i, so that $\sum_c R_{ic}(t) \leq R_{max}$ for all t. It is assumed that R_{max} is large enough so that $\sum_c \lambda_{ic} \leq R_{max}$ for all i. In practice, the value of R_{max} is selected by the controllers in advance, and can be set either to a known or estimated peak rate on the inputs, or to a suitable fraction of the reservoir buffer size. In this way, if actual traffic streams send at a sustained rate of more than R_{max} bits per slot, the streams are filtered by the control valve to produce streams which do conform to the R_{max} constraint, and our analytical claims hold only for this filtered traffic. The R_{max} value plays the role of the A_{max} bound on exogenous arrivals from the previous chapter.

The following dynamic network control policy is decoupled into three separate algorithms, consisting of a flow control algorithm FLOW together with the DRPC algorithm for routing and scheduling. Recall that $g_{ic}(r)$ represents the utility function for sending data from source i to destination j at a rate r, and $L_{ic}(t)$ represents the current backlog in reservoir (i, c).

Cross-Layer Network Control Algorithm:

- *Flow Control* — (algorithm FLOW) The flow controller at each node i observes the current level of queue backlogs $U_i^{(c)}(t)$ in this node, and sets $R_{ic} = r_c$ for each commodity c, where the r_c values are chosen as follows:

$$\begin{array}{ll} \text{Maximize :} & \sum_{c=1}^{N} \left[V g_{ic}(r_c) - 2r_c U_i^{(c)}(t)\right] \\ \text{Subject to:} & \sum_{c=1}^{N} r_c \leq R_{max} \;,\; 0 \leq r_c \leq L_{ic}(t) \;\text{ for all } c \end{array} \tag{5.3}$$

  where $V > 0$ is a chosen constant that effects the performance of the algorithm.

- *Routing and Scheduling* — Each node i observes the backlog in all neighboring nodes j to which it is connected by a link (i, j). Let $W_{ij}^{(c)} = U_i^{(c)}(t) - U_j^{(c)}(t)$ represent the differential backlog of commodity c data. Define W_{ij}^* as the maximum differential backlog over all commodities $c \in \{1, \ldots, N\}$, and let c_{ij}^* represent the maximizing commodity. Data of commodity c_{ij}^* is selected to be routed from node i to node j whenever $W_{ij}^* > 0$ (provided that link (i, j) is allocated non-zero rate by the power allocation layer).

- *Power Allocation* — The current channel state $\underline{S}(t)$ is observed, and power is allocated by choosing a power matrix $\underline{P}(t)$ to maximize $\sum_{ij} W_{ij}^* \mu_{ij}(\underline{P}, \underline{S}(t))$ subject to the constraint $\underline{P} \in \Pi$.

The flow control algorithm is decentralized, where the control valves for each node i require knowledge only of the queue backlogs in node i. The routing and scheduling algorithm acts according to the differential backlog strategy, and is decentralized provided that each node i knows the backlog levels of its neighbors. The power allocation strategy of maximizing $\sum_{ij} W_{ij}^* \mu_{ij}(\underline{P}, \underline{S}(t))$ is the most complex part of the algorithm, and cannot be optimally decentralized except when network links are independent, as described in Chapter 4.4. However, in cases where a pre-established (and potentially sub-optimal) power allocation policy is implemented in a distributed fashion, or in the special case when there is no power allocation and link rates randomly vary as pure functions of the channel states, then the combined algorithm is completely decentralized. In such cases, the flow control and routing layers optimally control the network for the channel conditions as given by the power control layer.

We note that rate-power curve formulation contains as special cases the problems of server allocation and hybrid resource allocation in networks with both wireless and wireline components, as described in Chapters 3 and 4. Thus, the fair control algorithm described above also applies to these situations.

5.2.3 Algorithm Performance

Here we describe the performance of the cross-layer algorithm of Section 5.2.2. The algorithm implements DRPC at the network layer, and chooses valve controls $R_{ic}(t)$ at the transport layer according to (5.3). Define:

$$\begin{aligned} \overline{r}_{ic}(t) &\triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\} \\ \overline{r}_{ic} &\triangleq \limsup_{t\rightarrow\infty}\overline{r}_{ic}(t) \end{aligned}$$

The $\overline{r}_{ic}(t)$ value is the resulting traffic rate allowed into the network by valve (i, c) up to time t, and its limiting value $\overline{r}_{ic}$ represents the rate of long term communication between nodes i and c provided that the network is stable.[3] We compare the utility associated with the $(\overline{r}_{ic})$ values with the utility of the operating point (r_{ic}^{opt}), where we define (r_{ic}^{opt}) as the optimal solution of the problem (5.1).

Recall that $A_i^{(c)}(t)$ is the arrival stream entering the (i, c) reservoir, and is assumed to be i.i.d over timeslots with arrival rate λ_{ic}. For the following performance bound, it is useful to define the scalar parameter λ_{sym} as the largest value such that $(\lambda_{sym}) \in \Lambda$. That is, λ_{sym} represents the largest rate that is simultaneously supportable by all user pairs (i, c) under the special case when all users send traffic uniformly to all other users. This turns out to be an important parameter used in the theorem below, which holds for any set of heterogeneous rate matrices (λ_{ic}) inside or outside of the capacity region. We further as-

[3] In Chapter Appendix 5.B it is shown that $\lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\} = \lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)$ whenever the latter limit exists.

sume utility functions are bounded and define $G_{max} \triangleq \max_{i,\sum_c r_c \leq R_{max}} \sum_c g_{ic}(r_c)$, that is, G_{max} represents the maximum utility that any user i can achieve by allocating the flow control variables (r_c) subject only to the constraint $\sum_c r_c \leq R_{max}$.

Theorem 10. *(Performance of Cross-Layer Control Algorithm) Consider a network with i.i.d. arrivals and channel states, operating under the cross-layer control algorithm with flow parameter $V > 0$. Let (r_{ic}^{opt}) represent the optimal solution of the problem (5.1). Suppose the utility functions $g_{ic}(r)$ are continuous, concave, non-negative, and non-decreasing. Then*

(a) The network layer is stable, and time average congestion satisfies

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{N(B + VG_{max})}{2\lambda_{sym}} \tag{5.4}$$

where

$$\begin{aligned} \overline{\sum_{ic} U_i^{(c)}} &\triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \left[\sum_{ic} \mathbb{E}\left\{ U_i^{(c)}(\tau) \right\} \right] \\ B &\triangleq (R_{max} + \mu_{max}^{in})^2 + (\mu_{max}^{out})^2 \end{aligned}$$

(b) If utility functions are linear*, so that $g_{ic}(r) = \theta_{ic} r$ for some non-negative weights $\{\theta_{ic}\}$, and if exogenous arrivals are upper-bounded so that $\sum_c A_{ic}(t) \leq R_{max}$ for all i and all t, then:*

$$\liminf_{t\to\infty} \sum_{ic} \theta_{ic} \overline{r}_{ic}(t) \geq \sum_{ic} \theta_{ic} r_{ic}^{opt} - \frac{BN}{V} \tag{5.5}$$

(c) For general concave utilities $g_{ic}(r)$, for any $x \geq 0$ the time average utility satisfies

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^{opt} - x) \check{q}_{ic}(x) - \frac{BN}{V} \tag{5.6}$$

where

$$\check{q}_{ic}(x) \triangleq \liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} Pr[L_{ic}(\tau) \geq r_{ic}^{opt} - x]$$

A simple proof of this theorem is given in Section 5.5.4. However, before presenting the analysis, in the following sections we consider the implications of this result.

5.2.4 Performing Arbitrarily Close to the Optimal Operating Point

The cross-layer strategy thus stabilizes the network layer and provides performance guarantees on the overall utility that is achieved. For the case of linear utilities, Theorem 10 establishes that:

$$\liminf_{t\to\infty}\sum_{ic}\theta_{ic}\overline{r}_{ic}(t)\geq\sum_{ic}\theta_{ic}r_{ic}^{opt}-\frac{BN}{V}$$

The lim inf is used in the above expression to show that the least possible limiting value of utility differs from the maximum utility by no more than BN/V, which can be made arbitrarily small by increasing the flow control parameter V. This comes at the expense of a potential increase in average network congestion: $\overline{\sum_{ic}U_i^{(c)}}\leq N(B+VG_{max})/(2\lambda_{sym})$. For the case where arrival rates are outside of the capacity region, this tradeoff is intuitive. Indeed, in this case, the optimal operating point lies on the boundary of the capacity region, and the parameter $1/V$ represents a distance measure between this optimal operating point and the operating point achieved by the cross-layer control algorithm. However, for the general case, we note that only the *congestion bound* grows with V, while actual congestion in the network may not grow if the arrival rates (λ_{ic}) are within the capacity region. Indeed, increasing V to infinity allows all arrivals into the network (eliminating valve control altogether) but delay does not grow to infinity if these raw data rates are stabilizable by the network layer.

For the case of general concave utilities, Theorem 10 establishes that for any $x\geq 0$, we have

$$\liminf_{t\to\infty}\sum_{ic}g_{ic}(\overline{r}_{ic}(t))\geq\sum_{ic}g_{ic}(r_{ic}^{opt}-x)\check{q}_{ic}-\frac{BN}{V}$$

where $\check{q}_{ic}\triangleq\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}Pr[L_{ic}(\tau)\geq r_{ic}^{opt}-x]$.

Ideally, we would like to set $x=0$ and have $\check{q}_{ic}=1$. It is clear that in unstable situations where all reservoirs are infinitely backlogged and there is always data waiting to be sent, we can set $x=0$ and have $\check{q}_{ic}=1$ for all (i,c). In this case, we again see that overall utility is arbitrarily close to the target utility, where the proximity to the optimal solution is determined by the parameter V.

Note that $L_{ic}(t)\geq A_{ic}(t)$ for all t, and hence:

$$Pr[L_{ic}(t)\geq r_{ic}^{opt}]\geq Pr[A_{ic}(t)\geq r_{ic}^{opt}]\geq Pr[A_{ic}(t)\geq\lambda_{ic}]$$

where the last inequality follows because $r_{ic}^{opt} \leq \lambda_{ic}$ by definition. Thus, if traffic is *regular* so that on every timeslot we have $A_{ic}(t) = \lambda_{ic}$, then we can again set $x = 0$ and have $Pr[L(t) \geq r_{ic}^{opt}] = 1$ every timeslot t, so that $\check{q}_{ic} = 1$ for all (i, c). Note that any rate-convergent data stream can be *probabilistically regularized* by implementing a preliminary *Z-Slot Smoother*. This device accepts an input stream $A_{ic}(t)$ and returns a *smoothed* stream $\tilde{A}_{ic}(t)$, where the new stream is defined by empirical averages of the input over Z-slot blocks. Specifically, $\tilde{A}_{ic}(t) = 0$ for $t \in \{0, 1, \ldots, Z-1\}$, $\tilde{A}_{ic}(t) = \frac{1}{Z}\sum_{\tau=0}^{Z-1} A_{ic}(\tau)$ for $t \in \{Z, \ldots, 2Z-1\}$, and in general the value of $\tilde{A}_{ic}(t)$ at all times during block $b+1$ is equal to the empirical average over block b (hence, all data exits the smoother within $2Z$ slots after entry).

Suppose such Z-slot smoothers are placed in front of all inputs, so that processes $\tilde{A}_{ic}(t)$ enter the reservoirs. Because $r_{ic}^{opt} \leq \lambda_{ic}$ for all (i, c), and because all inputs are rate convergent, it follows that for any $x > 0$, we have:

$$1 \geq Pr[L_{ic}(t) \geq r_{ic}^{opt} - x] \geq Pr[\tilde{A}_{ic}(t) \geq r_{ic}^{opt} - x] \rightarrow 1$$

where the last term tends to 1 as $Z \rightarrow \infty$, because $\tilde{A}_{ic}(t) \rightarrow \lambda_{ic} \geq r_{ic}^{opt}$. Thus, $\check{q}_{ic} \rightarrow 1$ as $Z \rightarrow \infty$. In particular, for i.i.d. inputs feeding into a Z-slot smoother, for all timeslots $t \geq Z$ we have for each (i, c):

$$\begin{aligned}
\check{q}_{ic} &\geq Pr\left[\tilde{A}_{ic}(t) \geq r_{ic}^{opt} - x\right] \\
&= Pr\left[\frac{1}{Z}\sum_{v=0}^{Z-1}[A_{ic}(v) - \lambda_{ic}] \geq r_{ic}^{opt} - \lambda_{ic} - x\right] \\
&= Pr\left[\frac{1}{\sqrt{Z}}\sum_{v=0}^{Z-1}\frac{A_{ic}(v) - \lambda_{ic}}{\sigma_{ic}} \geq \frac{(r_{ic}^{opt} - \lambda_{ic} - x)\sqrt{Z}}{\sigma_{ic}}\right] \\
&\approx 1 - Q\left(\frac{(\lambda_{ic} + x - r_{ic}^{opt})\sqrt{Z}}{\sigma_{ic}}\right) \geq 1 - Q\left(\frac{x\sqrt{Z}}{\sigma_{ic}}\right)
\end{aligned}$$

where $\sigma_{ic}^2 = \mathbb{E}\left\{A_{ic}(t)^2 - \lambda_{ic}^2\right\}$ is the variance of the random variable $A_{ic}(t)$. The last line follows by the Central Limit Theorem, where $Q()$ represents the tail of a zero mean, unit variance Gaussian random variable. Hence, in the nonlinear case, overall utility can be pushed arbitrarily close to the target utility if the flow control parameter V is sufficiently large and if the smoother block size Z is also sufficiently large.

5.3 Maximum Throughput and the Threshold Rule

Consider the case of linear utilities $g_{ic}(r)$ defined:

$$g_{ic}(r) \triangleq \theta_{ic} r \tag{5.7}$$

for some positive weights θ_{ic}, so that the objective is to maximize the weighted sum of throughput $\sum_{i \neq j} \theta_{ic} \overline{r}_{ic}$. This same objective has been recently considered in [139] for the problem of scheduling a single server over a parallel set of ON/OFF queues.

In this case of linear utility, the flow control optimization (5.3) for each user i is as follows:

$$\begin{array}{ll} \text{Maximize :} & \sum_c \left[V\theta_{ic} r_c - 2U_i^{(c)}(t) r_c \right] \\ \text{Subject to:} & 0 \leq r_c \leq L_{ic}(t), \sum_c r_c \leq R_{max} \end{array}$$

The solution of the above optimization has a simple threshold form, where some commodities receive as much of the R_{max} delivery rate as possible, while others receive none. In the special case where the user at node i desires communication with a single destination node c_i (so that $\lambda_{ic} = 0$ for all $c \neq c_i$), the flow control algorithm reduces to maximizing $V\theta_{ic_i} r - 2U_i^{(c_i)} r$ subject to $0 \leq r \leq \min[R_{max}, L_{ic_i}(t)]$, and the solution is the following threshold rule:

$$R_{ic_i}(t) = \begin{cases} \min[R_{max}, L_{ic_i}(t)] & \text{if } U_i^{(c_i)}(t) \leq \frac{V\theta_{ic_i}}{2} \\ 0 & \text{otherwise} \end{cases}$$

The qualitative structure of this flow control rule is intuitive: When backlog in the source queue is large, we should refrain from sending new data. The simple threshold form is qualitatively similar to the "bang-bang" control policies that often arise in classical control theory problems when the objective function is linear.

More directly, this threshold rule for network flow control is similar to the threshold scheduling rule developed in [139] for server scheduling in a downlink with ON/OFF channels and deterministic constraints on the channel states and packet arrivals. Specifically, the analysis of [139] demonstrates that there exists a threshold T such that serving the longest queue maximizes throughput, where all queues with backlog greater than T are treated as having backlog that is equal to this threshold. Although the structure of the downlink scheduling problem in [139] is different from our problem structure, as are the analytical techniques and resulting scheduling rules, the objective of maximizing a weighted sum of throughput is the same, and hence it is interesting that both sets of results yield threshold-type policies.

5.4 Proportional Fairness and the $1/U$ Rule

Consider now utility functions of the form:

$$g_{ic}(r) = \log(1 + r_{ic}) \tag{5.8}$$

It is shown in [76] that maximizing a sum of such utilities over any convex set Λ leads to *proportional fairness*. Specifically, the resulting (r_{ic}^{opt}) operating point satisfies:

$$\sum_{i \neq c} \frac{r_{ic}^{opt} - r_{ic}}{r_{ic}^{opt} + 1} \geq 0 \tag{5.9}$$

where (r_{ic}) is any other point within the capacity region Λ.[4] In the special case when there is only one destination c_i for each user i, the flow control algorithm reduces to maximizing $V\log(1+r) - 2U_i^{(c_i)}(t)r$ subject to $0 \leq r \leq \min[R_{max}, L_{ic_i}(t)]$, which leads to the following "$1/U$" flow control function:

$$R_{ic_i}(t) = \frac{V}{2U_i^{(c_i)}(t)} - 1 \ , \qquad \text{if } 0 \leq \tfrac{V}{2U_i^{(c_i)}(t)} - 1 \leq \min[R_{max}, L_{ic_i}(t)]$$

If the unfinished work $U_i^{(c_i)}(t)$ is too large for the specified interval, $R_{ic_i}(t)$ is set to 0, and if it is too small then $R_{ic_i}(t)$ is set to the value $\min[R_{max}, L_{ic_i}(t)]$.

Here we see that the flow control valve restricts flow according to a continuous function of the backlog level at the source queue, being less conservative in its admission decisions when backlog is low and more conservative when backlog is high.

[4] Strictly speaking, the proportionally fair allocation seeks to maximize $\sum_{ic}\log(r_{ic})$, leading to $\sum_{i\neq c}\frac{r_{ic}^{opt}-r_{ic}}{r_{ic}^{opt}} \geq 0$. However, utility functions of the form $\log(r)$ can take negative values and are not lower bounded. Thus, we use functions of the form $\log(1+r)$, and thereby obtain a proportionally fair allocation with respect to the quantity $(r_{ic}^{opt}+1)$.

5.5 Performance Analysis

Here we establish the analytical tools used to prove Theorem 10, which describes the performance of the cross-layer control algorithm. We first develop a simple modification of the Lyapunov Drift Lemma of Chapter 2 (Lemma 2), enabling backlog minimization and utility maximization to be performed using a single drift analysis.

5.5.1 Lyapunov Drift with Utility Metric

Let $L(\underline{U})$ represent a Lyapunov function of unfinished work in a timeslotted system with unfinished work process $\underline{U}(t)$. Recall that the only criterion for a Lyapunov function is that it be non-negative.

Lemma 12. *If there is a fixed interval K such that for all timeslots t, the Lyapunov drift satisfies:*

$$\mathbb{E}\left\{L(\underline{U}(t+K)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \leq C(t) - \epsilon \sum_{ic} U_i^{(c)}(t)$$

for some upper bounded process $C(t)$ and some positive constants $\epsilon > 0, V > 0$, and further if $\mathbb{E}\left\{L(\underline{U}(t_0))\right\} < \infty$ for all initial timeslots $t_0 \in \{0, 1, \ldots, K-1\}$, then the system is stable, and

$$\begin{aligned}
\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{ic} \mathbb{E}\left\{U_i^{(c)}(\tau)\right\} &\leq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \frac{\mathbb{E}\left\{C(\tau)\right\}}{\epsilon} \\
\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{ic} \mathbb{E}\left\{U_i^{(c)}(\tau)\right\} &\leq \liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \frac{\mathbb{E}\left\{C(\tau)\right\}}{\epsilon}
\end{aligned}$$

Proof. The proof of the above lemma is similar to the proof of the Lyapunov drift result of Lemma 2 (from Chapter 2) and is given in Appendix 5.A. □

From this simple lemma we develop the following statement concerning Lyapunov drift and utility optimization.

Lemma 13. (Lyapunov Drift with Utility Metric) *Let $L(\underline{U})$ represent a Lyapunov function for a timeslotted system with unfinished work process $\underline{U}(t)$. Let $\underline{R}(t) = (R_{ic}(t))$ represent an input process driving the system, and let $\underline{r}^* = (r_{ic}^*)$ represent any fixed matrix (to be used as a fixed operating point with which to compare network utility). Suppose utility functions $g_{ic}(r)$ are non-negative and bounded so that $0 \leq \sum_c g_{ic}(R_{ic}(t)) \leq G_{max}$ for all i and all t.*

If there is a fixed interval K such that for all timeslots t, the Lyapunov drift satisfies:

$$\mathbb{E}\left\{L(\underline{U}(t+K)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \leq C - V\textstyle\sum_{ic} g_{ic}(r_{ic}^* - x)\mathbb{E}\left\{q_{ic}(t) \left|\underline{U}(t)\right.\right\}$$
$$+V\textstyle\sum_{ic}\mathbb{E}\left\{g_{ic}(R_{ic}(t)) \left|\underline{U}(t)\right.\right\} - \epsilon\textstyle\sum_{ic} U_i^{(c)}(t)$$

for some non-negative real constants $x \geq 0, C \geq 0, V \geq 0$, for a strictly positive value $\epsilon > 0$, and for some bounded process $q_{ic}(t)$ satisfying $0 \leq q_{ic}(t) \leq 1$, and further if $\mathbb{E}\left\{L(\underline{U}(t_0))\right\} < \infty$ for all initial timeslots $t_0 \in \{0, 1, \ldots, K-1\}$, then:

(a) The system is stable, and

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{C}{\epsilon} + \frac{V}{\epsilon}\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{ic}\mathbb{E}\left\{g_{ic}(R_{ic}(\tau)) - g_{ic}(r_{ic}^* - x)q_{ic}(\tau)\right\}$$

In particular, the following congestion bound holds:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{C}{\epsilon} + \frac{VNG_{max}}{\epsilon} \tag{5.10}$$

(b) If utility functions are continuous, concave, and non-decreasing, then

$$\liminf_{t\to\infty}\sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^* - x)\check{q}_{ic} - \frac{C}{V} \tag{5.11}$$

where

$$\begin{aligned}\overline{r}_{ic}(t) &\triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\}\\ \check{q}_{ic} &\triangleq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{q_{ic}(\tau)\right\}\end{aligned}$$

Proof. The proof follows directly from Lemma 12 and is given in Chapter Appendix 5.A. □

Note that the performance guarantee on network congestion in part (a) of the above lemma uses a lim sup, indicating that the greatest possible limiting value of network congestion is upper bounded by the right hand side of (5.10). Likewise, the performance guarantee on utility in part (b) uses a lim inf to indicate that the least possible limiting value of utility is greater than or equal to the right hand side of (5.11).

To help interpret the above lemma, we define $\overline{r}_{ic} \triangleq \limsup_{t\to\infty}\overline{r}_{ic}(t)$, and assume that

$x = 0$ and $q_{ic}(t) = 1$ for all t and all (i, c). Suppose the $g_{ic}(r)$ functions are non-decreasing, continuous, and concave. With these properties, it is not difficult to show the $g_{ic}(r)$ functions satisfy:

$$\sum_{ic} g_{ic}(\overline{r}_{ic}) \geq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{ic} \mathbb{E}\left\{g_{ic}(R_{ic}(\tau))\right\}$$

Using this $\overline{r}_{ic}$ notation, it follows that the Lyapunov drift condition of Lemma 13 implies that:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{C}{\epsilon} + \frac{V}{\epsilon}\sum_{ic} g_{ic}(\overline{r}_{ic}) - \frac{V}{\epsilon}\sum_{ic} g_{ic}(r_{ic}^*)$$

The above inequality expresses a fundamental tradeoff between utility optimization and network congestion for any stochastic network satisfying the drift condition of Lemma 13. Indeed, the equation above implies the following two bounds on congestion and utility:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{C}{\epsilon} + \frac{V}{\epsilon} N G_{max}$$

$$\sum_{ic} g_{ic}(\overline{r}_{ic}) \geq \sum_{ic} g_{ic}(r_{ic}^*) - \frac{C}{V}$$

The use of the valve parameter V is now apparent: The achieved utility differs from the target utility by no more than C/V, which can be made arbitrarily small by choosing suitably large values for V. However, increasing the V parameter increases the bound on average network congestion, which (potentially) causes an increase in network delay.

At first inspection, the drift condition of Lemma 13 may seem esoteric and unlikely to hold for a general stochastic system. Much to the contrary, we find that, given a stochastic system whose stability is achieved by minimizing the drift of a quadratic Lyapunov function, it is often a simple matter to design a control law for the $R_{ic}(t)$ inputs so that the overall system dynamics satisfy the drift condition of Lemma 13. This is made evident in the forthcoming analysis.

5.5.2 A Near-Optimal Operating Point (r_{ic}^*)

We compare the utility achieved by the cross-layer control algorithm presented in Section 5.2.2 to the utility of a near-optimal solution (r_{ic}^*) to the optimization problem (5.1). Specifically, for any $\epsilon > 0$, we define the set Λ_ϵ as follows:

$$\Lambda_\epsilon \triangleq \{\underline{r} = (r_{ic}) \,|\, (r_{ic} + \epsilon) \in \Lambda, r_{ic} \geq 0 \text{ for all } (i,c) \ \} \tag{5.12}$$

Thus, the set Λ_ϵ can be viewed as the resulting set of rate matrices within the network capacity region when an "ϵ-layer" of the boundary is stripped away. Note that $\Lambda_\epsilon \to \Lambda$ as $\epsilon \to 0$. The operating point (r_{ic}^*) is defined as the optimal solution to the following optimization problem:

$$\begin{aligned} \text{Maximize :} \quad & \textstyle\sum_{ic} g_{ic}(r_{ic}) \\ \text{Subject to:} \quad & \underline{r} \in \Lambda_\epsilon \\ & \underline{r} \leq \lambda \end{aligned} \tag{5.13}$$

This optimization differs from the optimization in (5.1) in that the set Λ is replaced by the set Λ_ϵ. The utility at the operating point (r_{ic}^*) increases to the optimal utility given by the solution of (5.1) as $\epsilon \to 0$, as described by the following lemma.

Let λ_{sym} represent the largest value such that $(\lambda_{sym}) \in \Lambda$, that is, all user pairs (i,c) can stably communicate at rate λ_{sym}.

Lemma 14. *(Deviation from Optimality) Let (r_{ic}^{opt}) represent an optimal solution to (5.1). Suppose the utility functions $g_{ic}(r)$ are continuous and differentiable, and let M represent a bound on the derivative, maximized over the interval $0 \leq r \leq R_{max}$ and over all functions $g_{ic}(r)$:*

$$M \triangleq \max_{(i,c)\in\{1,\ldots,N\}^2, 0 \leq r \leq R_{max}} \frac{dg_{ic}(r)}{dr}$$

Consider any ϵ such that $0 < \epsilon \leq \lambda_{sym}$. Then:

$$\sum_{i,c} g_{ic}(r_{ic}^{opt}) \geq \sum_{i,c} g_{ic}(r_{ic}^*) \geq \sum_{i,c} g_{ic}(r_{ic}^{opt}) - \epsilon \frac{NMR_{max}}{\lambda_{sym}}$$

Proof. The inequality $\sum_{i,c} g_{ic}(r_{ic}^{opt}) \geq \sum_{i,c} g_{ic}(r_{ic}^*)$ follows because $\Lambda_\epsilon \subset \Lambda$, and hence the maximum utility over the larger set Λ is greater than or equal to the maximum utility over the smaller set Λ_ϵ.

To prove the second inequality, note that $(r_{ic}^{opt}) \in \Lambda$ and $(\lambda_{sym}) \in \Lambda$, and hence by convexity of the capacity region Λ we have $(1 - \frac{\epsilon}{\lambda_{sym}})(r_{ic}^{opt}) + \frac{\epsilon}{\lambda_{sym}}(\lambda_{sym}) \in \Lambda$. It follows that $(1 - \frac{\epsilon}{\lambda_{sym}})(r_{ic}^{opt}) \in \Lambda_\epsilon$. By definition of (r_{ic}^*) as the utility maximizer of the problem (5.13), we have:

$$\sum_{i,c} g_{ic}(r_{ic}^*) \geq \sum_{i,c} g_{ic}\left((1 - \frac{\epsilon}{\lambda_{sym}})r_{ic}^{opt}\right)$$

However, because derivatives of each utility function are bounded by M, for each (i, c) we have $g_{ic}(r_{ic}^{opt} - \frac{\epsilon}{\lambda_{sym}}r_{ic}^{opt}) \geq g_{ic}(r_{ic}^{opt}) - M\frac{\epsilon}{\lambda_{sym}}r_{ic}^{opt}$. Hence:

$$\begin{aligned}\sum_{i,c} g_{ic}(r_{ic}^*) &\geq \sum_{i,c} g_{ic}(r_{ic}^{opt}) - \sum_{i,c} M\frac{\epsilon r_{ic}^{opt}}{\lambda_{sym}} \\ &\geq \sum_{i,c} g_{ic}(r_{ic}^{opt}) - \frac{M\epsilon}{\lambda_{sym}}\sum_{i=1}^{N} R_{max}\end{aligned}$$

proving the result. □

5.5.3 Achieving (r_{ic}^*)

The operating point (r_{ic}^*) is contained within the set Λ_ϵ, and hence by definition the matrix $(r_{ic}^* + \epsilon)$ is within the capacity region Λ. Thus, by the description of the network capacity region given in Theorem 5, there exists a matrix (G_{ab}) within the network graph family Γ together with multicommodity flows $\{f_{ab}^{(c)}\}$ that satisfy the non-negativity, flow efficiency, flow conservation, and link constraints of (4.4)-(4.7) with respect to the data rates $(r_{ic}^* + \epsilon)$. In particular:

$$\sum_b f_{ib}^{(c)} - \sum_a f_{ai}^{(c)} \geq r_{ic}^* + \epsilon \text{ for } i \neq c \tag{5.14}$$

$$\sum_c f_{ab}^{(c)} \leq (G_{ab})$$

From the analysis in Chapter 4, a stationary randomized control algorithm STAT can be implemented yielding power allocations $\underline{P}^{STAT}(t)$ (chosen by only considering the current channel state). The resulting transmission rates are $\mu_{ab}^{(c)STAT}(t)$, where each $\mu_{ab}^{(c)STAT}(t)$ process is rate convergent with rate $f_{ab}^{(c)}$ [see Lemma 8 together with the description of the policy STAT in Section 4.3.2]. In the special case of i.i.d. channel states, it is clear that the resulting transmission rates $\mu_{ab}^{(c)STAT}(t)$ are i.i.d. and satisfy:

$$\mathbb{E}\left\{\mu_{ab}^{(c)STAT}(t)\right\} = f_{ab}^{(c)} \text{ for all } a, b, c \text{ and all slots } t \tag{5.15}$$

It follows from (5.15) and (5.14) that

$$\mathbb{E}\left\{\sum_b \mu_{ib}^{(c)STAT}(t) - \sum_a \mu_{ai}^{(c)STAT}(t)\right\} \geq r_{ic}^* + \epsilon \text{ for all } i \neq c \tag{5.16}$$

The above inequality is important for evaluating the performance of the stationary allocation scheme with respect to the target rates r_{ic}^*.

5.5.4 Algorithm Analysis

Here we analyze performance of the cross-layer algorithm of Section 5.2.2 and prove Theorem 10. We first present a result showing that performance is arbitrarily close to the near-optimal operating point (r_{ic}^*) of the above section.

We assume throughout that utility functions $g_{ic}(r)$ are continuous, concave, non-decreasing, and bounded so that $0 \leq \sum_c g_{ic}(r_c) \leq G_{max}$ whenever $\sum_c r_c \leq R_{max}$.[5] Fix $\epsilon > 0$, and let (r_{ic}^*) represent the solution of the modified optimization problem (5.13), defined in terms of the set Λ_ϵ. We have:

Lemma 15. *Consider a network operating under the cross-layer control algorithm with flow parameter $V > 0$, and let $(r_{ic}^*) \in \Lambda_\epsilon$ represent an optimal solution of the problem (5.13). If arrivals and channel states are i.i.d. over timeslots, then*

(a) The network layer is stable, and time average congestion satisfies

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{BN + VNG_{max}}{2\epsilon} \tag{5.17}$$

where $B \triangleq (R_{max} + \mu_{max}^{in})^2 + (\mu_{max}^{out})^2$.

(b) If utility functions are linear*, so that $g_{ic}(r) = \theta_{ic} r$ for some non-negative weights $\{\theta_{ic}\}$, and if $\sum_c A_{ic}(t) \leq R_{max}$ for all c and t, then*

$$\liminf_{t\to\infty} \sum_{ic} \theta_{ic}\overline{r}_{ic}(t) \geq \sum_{ic} \theta_{ic} r_{ic}^* - \frac{BN}{V} \tag{5.18}$$

[5] Generalizations for utility functions lacking in one or all of these properties can be developed, but are not required here.

(c) For general concave utilities $g_{ic}(r)$, for any $x \geq 0$ the time average utility satisfies

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^* - x)\check{q}_{ic}(x) - \frac{BN}{V} \tag{5.19}$$

where $\check{q}_{ic}(x) \triangleq \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} Pr[L_{ic}(\tau) \geq r_{ic}^* - x]$.

The above result is similar to the statement of Theorem 10 with the exception that we compare to the operating point (r_{ic}^*) rather than (r_{ic}^{opt}).

Proof. To prove (a), define the Lyapunov function $L(\underline{U}) = \sum_{i,c}[U_i^{(c)}]^2$. Let $\Delta(\underline{U}(t))$ represent the Lyapunov drift between slots t and $t+1$ (it suffices to consider 1-step analysis because arrival and channel states are assumed to be i.i.d. every slot). Note by (4.28) that for any network control algorithm, this drift satisfies:

$$\begin{aligned} \Delta(\underline{U}(t)) &\triangleq \mathbb{E}\left\{L(\underline{U}(t+1)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \\ &\leq BN - 2\Phi\left(\underline{U}(t)\right) + 2\sum_{i,c} U_i^{(c)}(t)\mathbb{E}\left\{R_{ic}(t) \mid \underline{U}(t)\right\} \end{aligned} \tag{5.20}$$

where

$$\Phi\left(\underline{U}(t)\right) \triangleq \sum_{i\neq c} U_i^{(c)}(t)\mathbb{E}\left\{\sum_b \mu_{ib}^{(c)}(t) - \sum_a \mu_{ai}^{(c)}(t) \mid \underline{U}(t)\right\}$$

The value of $\Phi\left(\underline{U}(t)\right)$ represents the component of drift that is affected by the power allocation and routing algorithms. To represent the effects of the flow control algorithm, we define the following quantity:

$$\Psi(\underline{U}(t)) \triangleq \mathbb{E}\left\{\sum_{i\neq c}\left[\frac{V}{2}g_{ic}(R_{ic}(t)) - U_i^{(c)}(t)R_{ic}(t)\right] \mid \underline{U}(t)\right\}$$

Adding and subtracting $2\Psi(\underline{U}(t))$ in the right hand side of (5.20), we have:

$$\Delta(\underline{U}(t)) \leq BN + V\sum_{i\neq c}\mathbb{E}\left\{g_{ic}(R_{ic}(t)) \mid \underline{U}(t)\right\} - 2\Phi\left(\underline{U}(t)\right) - 2\Psi(\underline{U}(t)) \tag{5.21}$$

Let $\Phi^{DRPC}(\underline{U}(t))$ and $\Psi^{FLOW}(\underline{U}(t))$ represent the network and flow control terms when the DRPC algorithm is used at the network layer, and the FLOW algorithm is used at the transport layer. At this point, it should be intuitively clear that the FLOW algorithm was designed to maximize $\Psi^{FLOW}(\underline{U}(t))$ at all timeslots t, while the DRPC algorithm was designed to maximize $\Phi^{DRPC}(\underline{U}(t))$ at all timeslots t. We now show that the above drift condition can be expressed in the same form as the drift condition of Lemma 13. The proof relies on the following simple claims, proven at the end of this subsection.

Claim 5.1: For all t, the DRPC algorithm satisfies:

$$\Phi^{DRPC}(\underline{U}(t)) \geq \sum_{i \neq c} U_i^{(c)}(t)(r_{ic}^* + \epsilon) \tag{5.22}$$

Claim 5.2: If utilities are *linear* so that $g_{ic}(r) = \theta_{ic} r$, and if $\sum_c A_{ic}(t) \leq R_{max}$ for all t and all i, then for all timeslots t the FLOW algorithm satisfies:

$$\Psi^{FLOW}(\underline{U}(t)) \geq \sum_{i \neq c} \frac{V}{2} \theta_{ic} r_{ic}^* - \sum_{i \neq c} U_i^{(c)}(t) r_{ic}^* \tag{5.23}$$

Using the results of the claims by plugging (5.23) and (5.22) into (5.21) yields:

$$\Delta(\underline{U}(t)) \leq BN - V \sum_{i \neq c} \theta_{ic} r_{ic}^* + V \sum_{i \neq c} \mathbb{E}\left\{\theta_{ic} R_{ic}(t) \mid \underline{U}(t)\right\} - 2 \sum_{i \neq c} U_i^{(c)}(t) \epsilon$$

The above inequality is the drift condition we are looking for, and the results of (a) and (b) follow by direct application of Lemma 13 (defining $C \triangleq BN$ and $q_{ic}(t) \triangleq 1$, $x \triangleq 0$).

Claim 5.3: For general concave utilities $g_{ic}(r)$, at every timeslot t and for any value $x \geq 0$, the FLOW algorithm satisfies:

$$\Psi^{FLOW}(\underline{U}(t)) \geq \sum_{i \neq c} \frac{V}{2} g_{ic}(r_{ic}^* - x) Pr[L_{ic}(t) \geq r_{ic}^* - x \mid \underline{U}(t)] - \sum_{i \neq c} U_i^{(c)}(t) r_{ic}^* \tag{5.24}$$

Plugging (5.24) and (5.22) into (5.21) in the same manner as before yields the following expression for Lyapunov drift for this case of general utilities:

$$\begin{aligned} \Delta(\underline{U}(t)) \leq BN + V \textstyle\sum_{i \neq c} \mathbb{E}\left\{g_{ic}(R_{ic}(t)) \mid \underline{U}(t)\right\} \\ -2 \textstyle\sum_{i \neq c} U_i^{(c)}(t) \epsilon - V \textstyle\sum_{i \neq c} g_{ic}(r_{ic}^* - x) Pr[L_{ic}(t) \geq r_{ic}^* - x \mid \underline{U}(t)] \end{aligned}$$

Using this expression in Lemma 13 (defining $C \triangleq BN$, and $q_{ic}(t) \triangleq 1_{[L_{ic}(t) \geq r_{ic}^* - x]}$) proves the result. □

We now prove the claims used in the above lemma.

Claim 5.1: For all t, the DRPC algorithm satisfies:

$$\Phi^{DRPC}(\underline{U}(t)) \geq \sum_{i\neq c} U_i^{(c)}(t)(r_{ic}^* + \epsilon)$$

Proof. From the results of Chapter 4 (Lemma 9) applied to the special case of i.i.d. channels, we know that:[6]

$$\Phi^{DRPC}(\underline{U}(t)) \geq \Phi^{STAT}(\underline{U}(t))$$

where $\Phi^{STAT}(\underline{U}(t)) = \sum_{i\neq c} U_i^{(c)}(t)\mathbb{E}\left\{\sum_b \mu_{ib}^{(c)STAT}(t) - \sum_a \mu_{ai}^{(c)STAT}(t)\,|\underline{U}(t)\right\}$. Using this together with (5.16) yields the result. □

Claim 5.2: If utilities are *linear* so that $g_{ic}(r) = \theta_{ic} r$, and if $\sum_c A_{ic}(t) \leq R_{max}$ for all t and all i, then for all timeslots t the FLOW algorithm satisfies:

$$\Psi^{FLOW}(\underline{U}(t)) \geq \sum_{i\neq c} \frac{V}{2}\theta_{ic} r_{ic}^* - \sum_{i\neq c} U_i^{(c)}(t) r_{ic}^*$$

Proof. Every timeslot t, the FLOW algorithm maximizes $\sum_c \left[\frac{V}{2}\theta_{ic} r_c - U_i^{(c)}(t) r_c\right]$ for each user i over the region $\sum_c r_c \leq R_{max}$, $0 \leq r_c \leq L_{ic}(t)$. Because $\sum_c A_{ic}(t) \leq R_{max}$ and $A_{ic}(t) \leq L_{ic}(t)$ (as the reservoir level is always at least the amount of new arrivals), the point $(r_c) = (\gamma_{ic} A_{ic}(t))$ is within this maximization region for any γ_{ic} values such that $0 \leq \gamma_{ic} \leq 1$. Define $\gamma_{ic} \triangleq r_{ic}^*/\lambda_{ic}$ and note that, by definition, $0 \leq \gamma_{ic} \leq 1$ for all (i, c). Hence:

$$\sum_c \left[\frac{V}{2}\theta_{ic} R_{ic}(t) - U_i^{(c)}(t) R_{ic}(t)\right] \geq \sum_c \left[\frac{V}{2}\theta_{ic}\gamma_{ic} A_{ic}(t) - U_i^{(c)}(t)\gamma_{ic} A_{ic}(t)\right]$$

Taking expectations and using the fact that $\mathbb{E}\{A_{ic}(t)\} = \lambda_{ic}$ (which follows because arrivals are i.i.d. every slot), we have:

$$\mathbb{E}\left\{\sum_c \left[\frac{V}{2}\theta_{ic} R_{ic}(t) - U_i^{(c)}(t) R_{ic}(t)\right] \mid \underline{U}(t)\right\} \geq \sum_c \left[\frac{V}{2}\theta_{ic} r_{ic}^* - r_{ic}^* U_i^{(c)}(t)\right]$$

Summing over $i \neq c$ proves the claim. □

[6] Recall from Chapter 4 that $\Phi^{DRPC}(\underline{U}(t)) = \Phi^{FRAME}(\underline{U}(t))$ for i.i.d. arrivals and channel states.

Claim 5.3: For all slots t and for all values $x \geq 0$, the FLOW algorithm satisfies:

$$\Psi^{FLOW}(\underline{U}(t)) \geq \sum_{i \neq c} \frac{V}{2} g_{ic}(r_{ic}^* - x) Pr[L_{ic}(t) \geq r_{ic}^* - x \mid \underline{U}(t)] - \sum_{i \neq c} U_i^{(c)}(t) r_{ic}^*$$

(where $g_{ic}(r_{ic}^* - x)$ is defined to be zero whenever $r_{ic}^* - x < 0$).

Proof. The FLOW algorithm maximizes $\sum_c \left[\frac{V}{2} g_{ic}(r_c) - U_i^{(c)}(t) r_c\right]$ for each user i over the optimization region $\sum_c r_c \leq R_{max}$, $0 \leq r_c \leq L_{ic}(t)$. Let Ω represent this optimization region, and define the following values (y_{ic}):

$$y_{ic} \triangleq \begin{cases} r_{ic}^* - x & \text{if } 0 \leq r_{ic}^* - x \leq L_{ic}(t) \\ 0 & \text{otherwise} \end{cases}$$

Note that $\sum_c r_{ic}^* \leq R_{max}$ for each i, and hence the same is true for the y_{ic} values. Furthermore, $0 \leq y_{ic} \leq L_{ic}(t)$ for all (i, c), and hence the (y_{ic}) vector is within the optimization region Ω for each i. Comparing with the $(R_{ic}(t))$ vector, it follows that:

$$\sum_c \left[\frac{V}{2} g_{ic}(R_{ic}(t)) - U_i^{(c)}(t) R_{ic}(t)\right] \geq \sum_c \left[\frac{V}{2} g_{ic}(y_{ic}) - U_i^{(c)}(t) y_{ic}\right] \tag{5.25}$$

Define the indicator function $1_{[L_{ic}(t) \geq r_{ic}^* - x]}$ to take the value 1 whenever $L_{ic}(t) \geq (r_{ic}^* - x)$, and 0 otherwise. Without loss of generality, we assume $r_{ic}^* - x \geq 0$ for all (i, c) (otherwise, we simply restrict the summation on the right hand side of the following inequality to (i, c) pairs for which this property holds). From (5.25) and the definition of y_{ic}, we have:

$$\begin{aligned} \sum_c \left[\frac{V}{2} g_{ic}(R_{ic}(t)) - U_i^{(c)}(t) R_{ic}(t)\right] &\geq \sum_c \left[\frac{V}{2} g_{ic}(r_{ic}^* - x) - U_i^{(c)}(t)(r_{ic}^* - x)\right] 1_{[L_{ic}(t) \geq r_{ic}^* - x]} \\ &\geq \sum_c \left[\frac{V}{2} g_{ic}(r_{ic}^* - x) 1_{[L_{ic}(t) \geq r_{ic}^* - x]} - U_i^{(c)}(t) r_{ic}^*\right] \end{aligned}$$

Taking conditional expectations of the above inequality, it follows that:

$$\mathbb{E}\left\{\sum_c \left[\frac{V}{2} g_{ic}(R_{ic}(t)) - U_i^{(c)}(t) R_{ic}(t)\right] \mid \underline{U}(t)\right\} \geq$$
$$\sum_c \left[\frac{V}{2} g_{ic}(r_{ic}^* - x) Pr[L_{ic}(t) \geq r_{ic}^* - x \mid \underline{U}(t)] - U_i^{(c)}(t) r_{ic}^*\right]$$

Summing over all $i \neq c$ proves the claim. □

5.5.5 Optimizing the Bound

Here we complete the proof of Theorem 10 by optimizing the performance bounds given in Lemma 15. Under the conditions of the lemma, we know that the cross-layer control algorithm satisfies the following congestion bound:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{BN + VNG_{max}}{2\epsilon} \tag{5.26}$$

where $B \triangleq (R_{max} + \mu_{max}^{in})^2 + (\mu_{max}^{out})^2$.

Furthermore, for any $x \geq 0$, the resulting utility under the cross-layer algorithm satisfies:

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^* - x)\check{q}_{ic} - \frac{BN}{V} \tag{5.27}$$

where $(r_{ic}^*) \in \Lambda_\epsilon$ is the optimal solution of (5.13). Both of these performance bounds hold for a particular $\epsilon > 0$. However, note that the chosen value of ϵ only influences the performance bounds but does not influence the cross-layer algorithm or change any sample path that the algorithm traverses. Hence, improved bounds can be obtained by optimizing over all valid choices of ϵ, and the optimized choice of this value need not be the same for the utility bound and the congestion bound. Indeed, for the congestion bound (5.26), it is clear that we should make ϵ as large as possible, subject to the constraint that there exists a matrix (r_{ic}^*) such that $(r_{ic}^*) + (\epsilon) \in \Lambda$. We choose $\epsilon = \lambda_{sym}$, where λ_{sym} is defined as the *symmetric capacity* of the network, that is, λ_{sym} is the largest rate that is simultaneously supportable over all input streams (i, c) (so that $(\lambda_{sym}) \in \Lambda$). This choice of ϵ is valid, as it corresponds to $(r_{ic}^*) = (0)$ (and hence $\Lambda_\epsilon = \{0\}$). The new congestion bound is thus:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{BN + VNG_{max}}{2\lambda_{sym}} \tag{5.28}$$

To optimize the utility bound (5.27), note that this bound is written in terms of r_{ic}^* but does not contain the value ϵ. Thus, we can shrink ϵ to zero, allowing r_{ic}^* to tend to the value r_{ic}^{opt} as described in Lemma 14. Because the $g_{ic}(r)$ functions are continuous, we can push the limits through the functions, and we have the following optimized utility bound:

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^{opt} - x)\check{q}_{ic} - BN/V$$

The utility bound for linear utility functions can likewise be optimized, yielding:

$$\liminf_{t\to\infty} \sum_{ic} \theta_{ic}\overline{r}_{ic}(t) \geq \sum_{ic} \theta_{ic} r_{ic}^{opt} - \frac{BN}{V} \tag{5.29}$$

These bounds complete the proof of Theorem 10.

5.6 Mechanism Design, Network Pricing, and Nash Equilibrium

Throughout this chapter we have been considering flow control schemes for maximizing a sum of user utilities, with the implicit assumption that such schemes can be programmed into the communication software of individual users. However, non-compliant "rogue" users can always change their delivery protocol in an effort to improve their own data rates, potentially at the expense of other users. To avoid this behavior, we seek to design a pricing mechanism for charging users according to how much data they send to the network.

Let each user utility function $g_{ic}(r)$ be non-negative, concave, and increasing, taking values in units of dollars, representing the amount user i is willing to pay for service of its commodity c data at rate r. We define the *social optimum operating point* (r_{ic}^{opt}) to be the point that maximizes the sum of utilities $\sum_{i\neq c} g_{ic}(r_{ic})$ subject to $(r_{ic}) \in \Lambda$. We assume that each user is "infinitely backlogged", and hence always has data to send. In this scenario, if each user were to conform to the FLOW strategy, then for each i the $R_{ic}(t)$ values would be determined by $R_{ic}(t) = r_c$, where the r_c values solve:

$$\begin{array}{ll} \text{Maximize :} & \sum_c \left[V g_{ic}(r_c) - 2 r_c U_i^{(c)}(t) \right] \\ \text{Subject to:} & \sum_c r_c \leq R_{max} \end{array} \tag{5.30}$$

and the resulting congestion and utility bounds would be given as follows:

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{N(B + VG_{max})}{2\lambda_{sym}} \tag{5.31}$$

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r_{ic}^{opt}) - \frac{BN}{V} \tag{5.32}$$

and hence choosing a suitably large flow control parameter V maintains network stability while yielding an overall utility which is arbitrarily close to the social optimum.

Now note that the optimization problem (5.30) is equivalent to the following optimization:

$$\begin{array}{ll} \text{Maximize :} & \sum_c \left[g_{ic}(r_c) - \frac{2U_i^{(c)}(t)}{V} r_c \right] \\ \text{Subject to:} & \sum_c r_c \leq R_{max} \end{array} \tag{5.33}$$

The above form of the optimization immediately suggests the following pricing mechanism: Every timeslot, the network charges user i the following per-unit price for sending new commodity c data into the network:

$$PRICE_{ic}(t) = \frac{2U_i^{(c)}(t)}{V} \text{ dollars/bit} \tag{5.34}$$

We note that this pricing strategy is independent of the particular $g_{ic}(r)$ functions, and so the network does not require knowledge of the user utilities. If every timeslot users greedily maximize their own net benefit, equal to utility minus cost, then they naturally send data in conformance with the optimization (5.33). It follows that in a general time varying network with greedy users, backlog-proportional pricing achieves the social optimum, in the sense that the performance bounds (5.31) and (5.32) are satisfied.

It is interesting to consider the impact of a user who decides to delay gratification by sending at a rate different from that of the greedy optimization (5.33), with the hopes of achieving an overall better utility for himself or herself in the long run. We note that as the greedy strategy (5.33) yields a sum utility that is arbitrarily close to the social optimum, any potential improvement for an individual user is necessarily detrimental to others. This gives rise to the question of Nash equilibrium: Does the strategy (5.33), or a suitable variant, impose a Nash equilibrium on the system, making any change in strategy harmful to the individual user? The answer to this question is not clear. However, we note that it would be difficult for users to anticipate the consequences of changing from the greedy strategy to some other strategy, as the network topology and the future time variations in channel states are likely unknown.

5.7 Chapter Summary

In this chapter we have developed a cross-layer control strategy that operates for arbitrary input rates, regardless of whether these rates are inside or outside of the network capacity region. The strategy is decoupled into three separate algorithms, respectively treating flow control, routing, and power allocation. The flow control algorithm is implemented in a distributed fashion, where independent flow controllers act for each input stream and base decisions only on the current backlog at the source queue of that input, without requiring knowledge of the network topology, arrival rates, or channel conditions. The routing algorithm is also distributed, where routing decisions at a particular node are based only on the differential backlog between itself and its neighbors. The power allocation strategy requires full channel state information throughout the network, but can be optimally distributed in cases where channels are independent, and sub-optimally distributed using the simple schemes developed in Chapter 4.

The resulting throughput of the combined algorithm (with optimal power control) is arbitrarily close to the optimally fair operating point that could be achieved with full cooperation among users and with full knowledge of future arrivals and channels of the network. Distance to the optimal operating point decreases like $1/V$, where V is a parameter affecting a tradeoff in average delay experienced by data admitted into the network. Furthermore, the same result holds when power allocation is restricted to a specified scheme, so that the flow control and routing algorithms achieve optimal performance subject to any given power allocation layer running underneath them.

Analysis was performed by developing a Lyapunov drift theorem enabling utility optimization. This builds upon the results of Chapter 4 and contributes to a theory of dynamic network optimization.

Chapter Appendix 5.A — Lyapunov Drift with Utility Metric

Here we prove Lemmas 12 and 13.

Lemma 12: If there is a fixed interval K such that for all timeslots t, the Lyapunov drift satisfies:

$$\mathbb{E}\left\{L(\underline{U}(t+K)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \leq C(t) - \epsilon \sum_{ic} U_i^{(c)}(t) \tag{5.35}$$

for some upper bounded process $C(t)$ and some positive constants $\epsilon > 0, V > 0$, and further if $\mathbb{E}\{L(\underline{U}(t_0))\} < \infty$ for all initial timeslots $t_0 \in \{0, 1, \ldots, K-1\}$, then the system is stable, and

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{ic}\mathbb{E}\left\{U_i^{(c)}(\tau)\right\} \leq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\frac{\mathbb{E}\{C(\tau)\}}{\epsilon} \tag{5.36}$$

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{ic}\mathbb{E}\left\{U_i^{(c)}(\tau)\right\} \leq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\frac{\mathbb{E}\{C(\tau)\}}{\epsilon} \tag{5.37}$$

Proof. Let $t = t_0 + mK$, where $t_0 \in \{0, \ldots, K-1\}$. Taking expectations of (5.35) over the distribution of $\underline{U}(t)$ and summing from $m = 0$ to $m = M-1$ yields:

$$\mathbb{E}\{L(\underline{U}(t_0+MK))\} - \mathbb{E}\{L(\underline{U}(t_0))\} \leq \sum_{m=0}^{M-1}\mathbb{E}\{C(t_0+mK)\} - \epsilon\sum_{m=0}^{M-1}\sum_{ic}\mathbb{E}\left\{U_i^{(c)}(t_0+mK)\right\}$$

Shifting terms and using non-negativity of the Lyapunov function, we have:

$$\epsilon\sum_{m=0}^{M-1}\sum_{ic}\mathbb{E}\left\{U_i^{(c)}(t_0+mK)\right\} - \mathbb{E}\{L(\underline{U}(t_0))\} \leq \sum_{m=0}^{M-1}\mathbb{E}\{C(t_0+mK)\}$$

Summing over $t_0 \in \{0, \ldots, K-1\}$ and dividing by MK yields:

$$\epsilon\frac{1}{MK}\sum_{\tau=0}^{MK-1}\sum_{ic}\mathbb{E}\left\{U_i^{(c)}(\tau)\right\} - \frac{1}{MK}\sum_{t_0=0}^{K-1}\mathbb{E}\{L(\underline{U}(t_0))\} \leq \frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\{C(\tau)\} \tag{5.38}$$

Taking the $\limsup$ of both sides as $M \to \infty$ yields (5.36), while taking a $\liminf$ yields (5.37). Stability follows from (5.36) by the same proof as given in Lemma 2 in Chapter 2. □

<u>Lemma 13</u>: Let $L(\underline{U})$ represent a Lyapunov function for a timeslotted system with unfinished work process $\underline{U}(t)$. Let $\underline{R}(t) = (R_{ic}(t))$ represent an input process driving the system, and let $\underline{r}^* = (r^*_{ic})$ represent any fixed matrix (to be used as a fixed operating point with which to compare network utility). Then for any bounded utility functions $g_{ic}(r)$ (satisfying $0 \leq \sum_c g_{ic}(R_{ic}(t)) \leq G_{max}$), if there is a fixed interval K such that for all timeslots t, the Lyapunov drift satisfies:

$$\mathbb{E}\left\{L(\underline{U}(t+K)) - L(\underline{U}(t)) \mid \underline{U}(t)\right\} \leq C - V\textstyle\sum_{ic} g_{ic}(r^*_{ic} - x)\mathbb{E}\left\{q_{ic}(t) \,|\underline{U}(t)\right\}$$
$$+V\textstyle\sum_{ic} \mathbb{E}\left\{g_{ic}(R_{ic}(t)) \,|\underline{U}(t)\right\} - \epsilon \sum_{ic} U_i^{(c)}(t) \quad (5.39)$$

for some non-negative constants $C \geq 0, V \geq 0, x \geq 0$, some positive value $\epsilon > 0$, and for some bounded process $q_{ic}(t)$ satisfying $0 \leq q_{ic}(t) \leq 1$, and further if $\mathbb{E}\left\{L(\underline{U}(t_0))\right\} < \infty$ for all initial timeslots $t_0 \in \{0, 1, \ldots, K-1\}$, then:

(a) The system is stable, and

$$\overline{\sum_{ic} U_i^{(c)}} \leq \frac{C}{\epsilon} + \frac{V}{\epsilon} \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{ic} \mathbb{E}\left\{g_{ic}(R_{ic}(\tau)) - g_{ic}(r^*_{ic} - x)q_{ic}(\tau)\right\}$$

(b) If utility functions are continuous, concave, and non-decreasing, then

$$\liminf_{t\to\infty} \sum_{ic} g_{ic}(\overline{r}_{ic}(t)) \geq \sum_{ic} g_{ic}(r^*_{ic} - x)\check{q}_{ic} - \frac{C}{V} \quad (5.40)$$

where

$$\overline{r}_{ic}(t) \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{R_{ic}(\tau)\right\}$$
$$\check{q}_{ic} \triangleq \liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{q_{ic}(\tau)\right\}$$

Proof. Defining $C(t) \triangleq C - V\sum_{ic} g_{ic}(r^*_{ic} - x)\mathbb{E}\left\{q_{ic}(t) \,|\underline{U}(t)\right\} + V\sum_{ic} \mathbb{E}\left\{g_{ic}(R_{ic}(t)) \,|\underline{U}(t)\right\}$ and using the lim sup statement of Lemma 12 proves (a).

To prove (b), we keep the same definition of $C(t)$. Using (5.38) from Lemma 12 and noting that unfinished work is non-negative, we have:

$$-\frac{1}{MK}\sum_{t_0=0}^{K-1}\mathbb{E}\left\{L(\underline{U}(t_0))\right\} \leq \frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{C(\tau)\right\}$$

Inserting the definition of $C(t)$ into the above inequality yields:

$$V\textstyle\sum_{ic} g_{ic}(r_{ic}^* - x)\left[\frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{q_{ic}(\tau)\right\}\right] - \frac{1}{MK}\sum_{t_0=0}^{K-1}L(\underline{U}(t_0)) \leq \\ C + V\textstyle\sum_{ic}\left[\frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{g_{ic}(R_{ic}(\tau))\right\}\right] \qquad (5.41)$$

However, because each $g_{ic}(r)$ function is concave, we have by Jensen's inequality:

$$\frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{g_{ic}(R_{ic}(\tau))\right\} \leq g_{ic}(\overline{r}_{ic}(MK))$$

Using this in the right hand side of (5.41) and taking a lim inf of the resulting expression yields:

$$\liminf_{M\to\infty} V\sum_{ic} g_{ic}(r_{ic}^* - x)\left[\frac{1}{MK}\sum_{\tau=0}^{MK-1}\mathbb{E}\left\{q_{ic}(\tau)\right\}\right] \leq C + \liminf_{M\to\infty} V\sum_{ic} g_{ic}(\overline{r}_{ic}(MK)) \quad (5.42)$$

We modify the left hand side of the above inequality by noting that the lim inf of a sum of functions is greater than or equal to the sum of the lim infs. It follows that

$$V\sum_{ic} g_{ic}(r_{ic}^* - x)\check{q}_{ic} \leq C + \liminf_{M\to\infty} V\sum_{ic} g_{ic}(\overline{r}_{ic}(MK))$$

Dividing by V proves the result.[7] □

[7] To address a minor technicality, we note that it is not difficult to prove $\liminf_{M\to\infty}\sum_{ic} g_{ic}(\overline{r}_{ic}(MK)) = \liminf_{t\to\infty}\sum_{ic} g_{ic}(\overline{r}_{ic}(t))$. This follows because the $g_{ic}(r)$ functions are continuous, and because for any fixed integer v, we have $|\overline{r}_{ic}(t) - \overline{r}_{ic}(t-v)| \to 0$ as $t \to \infty$.

Chapter Appendix 5.B — Convergence of the $R_{ic}(t)$ values

Here we describe conditions under which the quantities $\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\}$ and $\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)$ converge as $t \rightarrow \infty$, which may be of mathematical interest to some readers (see also Chapter Appendix 2.A). Suppose the $R_{ic}(t)$ values are determined by the cross-layer control strategy.

Lemma 16. *If for all (i,c), $\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)$ converges to a value $\overline{r}_{ic}$ with probability 1 as $t \rightarrow \infty$, then*

(a) $\lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\} = \overline{r}_{ic}$

(b) $(\overline{r}_{ic}) \in \Lambda$

That is, the expectations converge to the same limiting matrix, and this matrix is inside the network capacity region.

Proof. To prove (a), note that $0 \leq R_{ic}(t) \leq R_{max}$, and hence for any $\epsilon > 0$, we have for all t:

$$\begin{aligned}\mathbb{E}\left\{\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)\right\} \leq \quad & (\overline{r}_{ic}+\epsilon)Pr\left[\tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1}R_{ic}(\tau)\leq \overline{r}_{ic}+\epsilon\right] + \\ & R_{max}\left(1 - Pr\left[\tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1}R_{ic}(\tau)\leq \overline{r}_{ic}+\epsilon\right]\right)\end{aligned}$$

Taking a limit as $t \rightarrow \infty$ and noting that $Pr\left[\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)\leq \overline{r}_{ic}+\epsilon\right] \rightarrow 1$ yields:

$$\lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\} \leq \overline{r}_{ic}+\epsilon$$

This inequality holds for any $\epsilon > 0$. Taking a limit as $\epsilon \rightarrow 0$ yields:

$$\lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_{ic}(\tau)\right\} \leq \overline{r}_{ic}$$

The reverse inequality can be proved similarly, establishing (a).

To prove (b), we note that $(\overline{r}_{ic})$ represents the input rate matrix to the network. As the network is always stable under the cross-layer control algorithm, it follows that $(\overline{r}_{ic}) \in \Lambda$. □

The above lemma shows that we can work with either the actual $R_{ic}(t)$ values, or with their expectations, provided that $\frac{1}{t}\sum_{\tau=0}^{t-1}R_{ic}(\tau)$ converges. We can artificially ensure such convergence by using renewal theory and implementing the following additional action in the cross-layer control algorithm: Every timeslot, the controller independently flips a biased coin which lands on heads with probability p. If heads occurs, all contents of all storage reservoirs and network queues are deleted—marking a renewal time.

It is not difficult to see that for any $p > 0$, the network is stable and the duration between renewal events is geometrically distributed with mean $1/p$. It follows that $\frac{1}{t}\sum_{\tau=0}^{t-1} R_{ic}(\tau)$ converges with probability 1 for any $p > 0$. Further, the rate of throwing away data due to the coin flips can be made arbitrarily small by choosing a suitably small probability p. It is intuitively clear that the network performance for $p > 0$ converges to the performance of the original cross-layer control algorithm (which operates with $p = 0$) as $p \to 0$. Hence, arguing heuristically, we expect the quantity $\frac{1}{t}\sum_{\tau=0}^{t-1} R_{ic}(\tau)$ to also converge with probability 1 when $p = 0$.

Note that a distributed implementation of the coin flip is to have controllers at individual nodes i empty their own reservoirs and queues with probability p. In this case, the controllers at all nodes simultaneously flip heads with probability p^N. This distributed coin-flip policy is also stable for any $p > 0$, and yields convergent data rates. An alternate policy is to only delete data from the storage reservoirs, and we conjecture that this policy has similar stability properties. We note that such policies correspond to the practical control action of throwing away any "ancient" data that has been sitting in a storage reservoir for days or weeks.

Chapter 6

Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks

In Chapters 3 and 4 the notion of a *network layer capacity region* was developed, and power allocation and routing strategies were constructed to achieve this capacity region for satellite downlinks and general multi-node wireless networks. In this chapter, we focus attention on ad-hoc wireless networks with mobility. A simple cell-partitioned model for the network is developed for which simple and exact expressions for network capacity and delay can be derived. We then explore strategies for improving delay by sending redundant copies of each packet, and a fundamental rate-delay tradeoff curve is established. This represents a new dimension in networks research. The material contained in this chapter is significantly different from the material in Chapters 3 and 4, and this chapter can be read independently.

6.1 The Cell Partitioned Network Model

We consider the effects of transmitting redundant packets along independent paths of an ad-hoc wireless network with mobility. Such redundancy improves delay at the cost of increasing overall network congestion. We show that redundancy cannot increase network capacity, but can significantly improve delay performance, yielding delay reductions by several orders of magnitude when data rates are sufficiently less than capacity.

We use the following *cell partitioned* network model: The network is partitioned into C non-overlapping cells of equal size (see Fig. 6-1). There are N mobile users independently roaming from cell to cell over the network, and time is slotted so that users remain in their current cells for a timeslot, and potentially move to a new cell at the end of the slot. If two users are within the same cell during a timeslot, one can transfer a single packet to the other. Each cell can support exactly one packet transfer per timeslot, and users within different cells cannot communicate during the slot. Multi-hop packet transfer proceeds as users change cells and exchange data. The cell partitioning reduces scheduling complexity and facilitates analysis. Similar cell partitioning has recently been considered by Cruz et. al in [34].

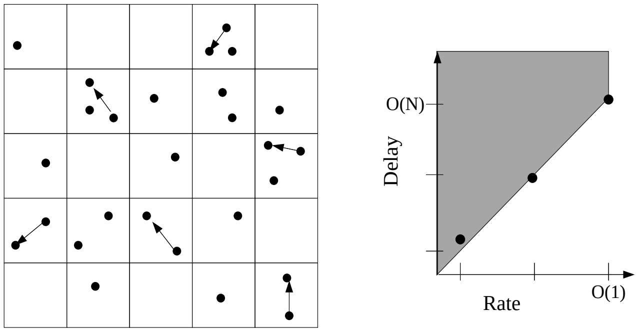


Figure 6-1: A cell-partitioned ad-hoc wireless network with C cells and N mobile users.

We consider the following simplified mobility model: Every timeslot, users choose a new cell location independently and identically distributed over all cells in the network. Such a mobility model is of course an over-simplification. Indeed, actual mobility is better described by Markovian dynamics, where users choose new locations every timeslot from the set of cells adjacent to their current cell. However, analysis under the simplified *i.i.d.* mobility model provides a meaningful bound on performance in the limit of *infinite mobility*. With this assumption, the network topology dramatically changes every timeslot, so that network behavior cannot be predicted and fixed routing algorithms cannot be used. Rather, because information about the current and future locations of users is unknown, one must rely on robust scheduling algorithms. Furthermore, recall from Corollary 5 that the network capacity under an i.i.d. mobility model is identical to the capacity region of a network with non-i.i.d. mobility with the same steady state distribution. Likewise, the delay theory of Chapter 4 shows that delay analysis for non-i.i.d. mobility models can be obtained directly from the i.i.d. analysis. Thus, our capacity results hold also for cases where mobility is described by simple Markovian random walks, considered in Section 6.8. Delay analysis for non-i.i.d. mobility is also presented, and simulation results demonstrate that performance is qualitatively similar to the i.i.d. case.

We compute an exact expression for the per-user transmission capacity of the network (for any number of users $N \geq 3$), and show that this capacity cannot be increased by using redundant packet transfers. When no redundancy is used, a modified version of the Grossglauser-Tse 2-hop relay algorithm in [53] is presented and shown to achieve capacity. The queueing delay in the network is explicitly computed and shown to be $O(N)/(\mu - \lambda_i)$ (where μ is the per-user network capacity, and λ_i is the rate at which user i transfers packets intended for its destination). Furthermore, it is shown that no scheduling algorithm can improve upon $O(N)$ delay performance unless redundancy is used.

We then consider modifying the 2-hop relay algorithm to allow redundant packet transmissions. It is shown that no scheme which restricts packets to two hops can achieve a better delay than $O(\sqrt{N})$. A scheduling protocol that employs redundant packet transmissions is developed and shown to achieve this delay bound when all users communicate at a reduced data rate of $O(1/\sqrt{N})$. A multi-hop protocol is then developed to achieve $O(\log(N))$ delay by further sacrificing throughput. The necessary condition $delay/rate \geq O(N)$ is established for any routing and scheduling algorithm, and the 2-hop relay algorithms are shown to meet this bound with equality while the multi-hop algorithm deviates from optimality by no more than a logarithmic factor.

Previous work on the capacity of ad-hoc wireless networks is found in [34] [111] [54] [57] [58] [8] [55] [116] [140] [106] [150]. Gupta and Kumar present asymptotic results for static networks in [57], [58], where it is shown that per-user network capacity is $O(1/\sqrt{N})$, and hence vanishes as the number of users N increases. The effect of mobility on the capacity of ad-hoc wireless networks was first explicitly developed in [54], where a 2-hop relay algorithm was developed and shown to support constant per-user throughput which does not vanish as the size of the network grows. These works do not consider the associated network *delay*, and analysis of the fundamental queueing delay bounds for general networks remains an important open question.

In [8] it is shown that for a network with a mixture of stationary users and mobile relay nodes, delay can be improved by exploiting velocity information and relaying packets to nodes moving in the direction of their destination. Routing for fully mobile networks using table updates is considered in [55]. Schemes for improving delay via diversity coding and multi-path routing are considered in [116], [140], although this work does not consider delays due to path sharing, queueing, or stochastic arrivals. Delay improvement via redundant packet transfers is considered in [106]. This idea is related to the notion of *content replication* considered for static peer-to-peer systems in [31] and for mobile networks in [150]. Our i.i.d. mobility model is similar to that used in [150], where *mobile infostations* are used to store content for users requesting file access. Mobile infostations are also used for monitoring animal population and roaming patterns in [127].

The contributions of this chapter are threefold: First, we demonstrate network capacity and delay analysis which considers the full effects of queueing, and show that delay grows as $O(N)$ when no redundancy is used. Second, we establish a fundamental delay/rate tradeoff curve that bounds performance of any routing and scheduling algorithm. Third, we develop three different protocols which achieve optimal or near optimal performance in different rate regimes.

In the next section, we establish the capacity of the cell partitioned network and analyze the delay of the capacity achieving relay algorithm. In Section 6.3 we develop delay bounds for transmission schemes with redundancy, and in Sections 6.4 and 6.5 we provide scheduling protocols which achieve these bounds. In Section 6.6 we prove necessity of $delay/rate \geq O(N)$, and show that the given protocols operate on the boundary of this rate-delay tradeoff curve. Simulations and Markovian mobility models are considered in Sections 6.7 and 6.8.

6.2 Capacity, Delay, and the 2-Hop Relay Algorithm

Consider a cell partitioned network such as that of Fig. 6-1. The shape and layout of cell regions is arbitrary, although we assume that cells have identical area, do not overlap, and completely cover the network area. We define:

- N = Number of Mobile Users
- C = Number of Cells
- $d = N/C$ = User/Cell density

Users move independently according to the *full-mobility model*, where the steady state location of each user is uniform over all cells.

Let λ_i represent the exogenous arrival rate of packets to user i (in units of packets/slot). Packets are assumed to arrive as a Bernoulli process, so that with probability λ_i a single packet arrives during the current slot, and otherwise no packet arrives. Other stochastic inputs with the same time average arrival rate can be treated similarly, and the arrival model does not affect the region of rates the network can support (see Chapter 4, Corollary 5).

We assume packets from source i must be delivered to a unique destination j. In particular we assume the number of users N is even and consider the one-to-one pairing: $1 \leftrightarrow 2$, $3 \leftrightarrow 4$, …, $(N-1) \leftrightarrow N$; so that user 1 communicates with user 2 and user 2 communicates with user 1, user 3 communicates with user 4 and user 4 communicates with user 3, and so on. Other source-destination scenarios can be treated similarly (see Section 6.2.2).

Packets are transmitted and routed through the network according to some scheduling algorithm. The algorithm chooses which packets to transmit on each timeslot without violating the physical constraints of the cell partitioned network or the following additional *causality constraint*: A user cannot transmit a packet that it has never received. Note that once a packet has been received by a user, it can be stored in memory and transmitted again and again if so desired. We assume that packets are equipped with header information so that they can be individually distinguished for scheduling purposes.

A scheduling algorithm is *stable* if the λ_i rates are satisfied for all users so that queues do not grow to infinity and average delays are bounded. Assuming that all users receive packets at the same data rate (so that $\lambda_i = \lambda$ for all i), the *capacity* of the network is the maximum rate λ that the network can stably support. Note that this is a purely network layer notion of capacity, where optimization is over all possible routing and scheduling protocols. Below we compute the network capacity, assuming users change cells in an i.i.d. fashion every timeslot. In Chapter 4 it is shown that the capacity region depends only on the steady state user location distribution. Hence, any Markovian model of user mobility which in steady state distributes users independently and uniformly over the network yields the same expression for capacity. A simple example of such a Markovian model is considered in Section 6.8.

Theorem 11. *The capacity of the network is:*

$$\mu = \frac{p+q}{2d} \tag{6.1}$$

where

$$p = 1 - \left(1 - \tfrac{1}{C}\right)^{N} - \tfrac{N}{C}\left(1 - \tfrac{1}{C}\right)^{N-1} \tag{6.2}$$

$$q = 1 - \left(1 - \tfrac{1}{C^2}\right)^{N/2} \tag{6.3}$$

and hence the network can stably support users simultaneously communicating at any rate $\lambda < \mu$.

Note that p represents the probability of finding at least two users in a particular cell, and q represents the probability of finding a source-destination pair within a cell.

Proof. The proof of the above theorem involves proving that $\lambda \leq \mu$ is necessary for network stability, and that $\lambda < \mu$ is sufficient. Sufficiency is established in Subsection 6.2.3, where a stabilizing algorithm is provided and exact expressions for average delay are derived. Here we prove necessity.

Consider any stabilizing scheduling strategy, perhaps one which uses full knowledge of future events. Let $X_h(T)$ represent the total number of packets transferred over the network from sources to destinations in h hops during the interval $[0, T]$. Fix $\epsilon > 0$. For network stability, there must be arbitrarily large values T such that the sum output rate is within ϵ of the total input rate:

$$\frac{\sum_{h=1}^{\infty} X_h(T)}{T} \geq N\lambda - \epsilon \tag{6.4}$$

If this were not the case, the total number of packets in the network would grow to infinity and hence the network would be unstable. The total number of packet transmissions in the network during the first T slots is at least $\sum_{h=1}^{\infty} hX_h(T)$. This value must be less than or equal to the total number of transmission opportunities $Y(T)$, and hence:

$$\sum_{h=1}^{\infty} hX_h(T) \leq Y(T) \tag{6.5}$$

where $Y(T)$ represents the total number of cells containing at least 2 users in a particular timeslot, summed over all timeslots $1, 2, \ldots, T$. By the law of large numbers, it is clear that $\frac{1}{T}Y(T) \to Cp$ as $T \to \infty$, where p is the steady state probability that there are two or more users within a particular cell, and is given by (6.2).

From (6.4) and (6.5), it follows that

$$\frac{1}{T}Y(T) \geq \frac{1}{T}X_1(T) + \frac{2}{T}\sum_{h=2}^{\infty} X_h(T) \geq \frac{1}{T}X_1(T) + 2\left((N\lambda - \epsilon) - \frac{1}{T}X_1(T)\right)$$

and hence

$$\lambda \leq \frac{\frac{1}{T}Y(T) + \frac{1}{T}X_1(T) + 2\epsilon}{2N} \tag{6.6}$$

It follows that maximizing λ subject to (6.6) involves placing as much rate as possible on the single hop paths. However, the time average rate $\frac{1}{T}X_1(T)$ of 1-hop communication between source-destination pairs is bounded. Indeed, the probability q that a particular cell contains a source-destination pair during a timeslot can be written as 1 minus the probability that no such pair is present. For the source-destination matching $1 \leftrightarrow 2, 3 \leftrightarrow 4, \ldots$, this probability is given as the value q specified in (6.3). Let $Z(T)$ represent the total number of cells containing source-destination pairs, summed over all timeslots $1, 2, \ldots, T$. Again by the law of large numbers, it follows that $\frac{1}{T}Z(T) \rightarrow Cq$. Furthermore, it is clear that the number of packets delivered on one hop paths is less than or equal to the number of such opportunities:

$$\frac{1}{T}X_1(T) \leq \frac{1}{T}Z(T) \tag{6.7}$$

Combining constraints (6.6) and (6.7) and taking limits as $T \rightarrow \infty$, we have:

$$\lambda \leq \frac{Cp + Cq + 2\epsilon}{2N} \tag{6.8}$$

The necessary condition follows by using the user/cell density definition $d = N/C$, and noting that ϵ can be chosen to be arbitrarily small. □

Taking limits as $N \rightarrow \infty$, we find the network capacity tends to the fixed value $(1 - e^{-d} - de^{-d})/(2d)$. This value tends to zero as d tends either to zero or infinity. Indeed, if d is too large, there will be many users in each cell, most of which will be idle as a single transmitter and receiver are selected. However, if d is too small, the probability of two users being in a given cell vanishes. Hence, for nonzero capacity, the ratio $d = N/C$ should be fixed as both N and C scale up. The optimal user/cell density d^* and the corresponding capacity μ^* are: $d^* = 1.7933$, $\mu^* = 0.1492$ (see Fig. 6-2). Thus, large cell partitioned networks cannot support more than 0.1492 packets/slot, but can achieve arbitrarily close to this data rate by scaling the number of cells C with N to maintain a constant user/cell density d^*.

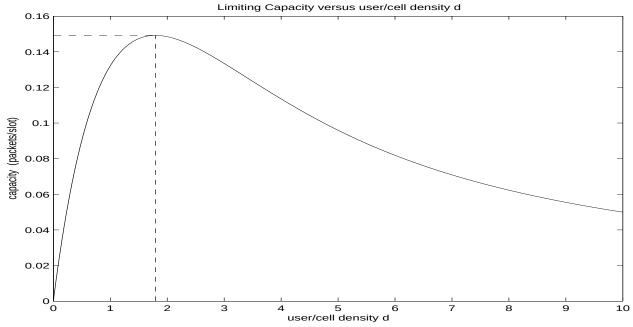


Figure 6-2: A plot of the limiting capacity $(1-e^{-d}-de^{-d})/(2d)$ as a function of the user/cell density d.

This μ^* capacity value is close to the maximum throughput estimate of 0.14 packets/slot for the $O(1)$ throughput strategy given by Grossglauser and Tse in [53], where the 0.14 number is obtained by a numerical optimization over a transmit probability θ. In the Grossglauser-Tse strategy, transmitting users send to their nearest neighbors to obtain a high signal to interference ratio on each transmission. The proximity of their optimal throughput to the value of μ^* suggests that when the transmit probability is optimized, the nearest-neighbor transmission policy behaves similarly to a cell-partitioned network. The same value μ^* arises when users send independent data to a finite collection of other users according to a *rate matrix* (λ_{ij}). In this case, μ^* represents the maximum sum rate into or out of any user provided that no user sends or receives more than any other, as described in Section 6.2.2.

6.2.1 Feedback Does not Increase Capacity

We note that the optimal throughput μ of Theorem 11 cannot be improved even if all users have perfect knowledge of future events (see proof of Theorem 11). Thus, control strategies which utilize redundant packet transfers, enable multiple users to overhear the same transmission, or allow for perfect feedback to all users when a given packet has been successfully received, cannot increase capacity.

Corollary 7. *The use of redundant packet transfers, multi-user reception, or perfect feedback, cannot increase network capacity.*

Proof. The capacity region given in Theorem 11 considers all possible strategies, including those which have perfect knowledge of future events. Hence, with full knowledge of the future, any strategy employing redundant packet transfers, multi-user reception, or perfect feedback can be transformed into a policy which does not use these features simply by removing the feedback mechanism (all feedback information would be a-priori known) and deleting all redundant versions of packets, so that only packets which first reach their destination are transmitted. Thus, such features cannot expand the region of stabilizable rates.

However, the capacity region can be *achieved* without feedback, redundancy, or perfect knowledge of the future (as described in the next section) and hence these features do not impact capacity. □

6.2.2 Heterogeneous Demands

Here we consider communication with heterogeneous rates (λ_{ij}), where λ_{ij} represents the rate user i receives exogenous data intended for user j. Define the *symmetric capacity region* as the region of all stabilizable data rates such that no user is transmitting or receiving at a higher total data rate than any other. Let K represent the maximum number of destination users to which a source transmits (i.e., for each user i, at most K of the λ_{ij} terms are nonzero).

Theorem 12. *The symmetric capacity region of the network has the form:*

$$\sum_j \lambda_{ij} \leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N) \; \forall i \tag{6.9}$$

$$\sum_i \lambda_{ij} \leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N) \; \forall j \tag{6.10}$$

Proof. This proof is similar to the proof of Theorem 11 and is given in Chapter Appendix 6.B. □

We note that the stability proof in Chapter Appendix 6.B involves finding a set of multi-commodity flows which support the data rates, and these flows are similar to the 2-hop routing scheme described in the proof of Theorem 8 in Chapter 4. This scheme can be directly implemented as a stabilizing algorithm by randomly and uniformly routing all data from sources to relay nodes on the first hop, and then routing from relays to destinations on the second hop. Such a *traffic uniformization* scheme is conceptually related to the 2-stage switch scheduling algorithm developed for $N \times N$ packet switches in [79], where packets are randomly assigned to output ports at the first stage so that traffic is uniform at the second stage.

Further note that the DRPC algorithm of Chapter 4 will also stabilize the network whenever the input rates are within the capacity region. In this case, DRPC is implemented by choosing the commodity and the transmitter-receiver pair in each cell which maximizes differential backlog. In the next section we develop an alternative strategy which is simpler to implement and yields an exact delay analysis. To simplify the discussion, throughout the rest of this chapter we assume that each user communicates with rate λ to a unique destination according to the pairing $1 \leftrightarrow 2$, $3 \leftrightarrow 4$, etc., so that $K = 1$ and the exact capacity result $\mu = (p+q)/(2d)$ of Theorem 11 applies for all network sizes N.

6.2.3 Delay Analysis and the 2-Hop Relay Algorithm

In this section, we consider a modified version of the Grossglauser-Tse relay algorithm of [53], and show the algorithm is capacity achieving with a bounded average delay. The algorithm restricts packets to 2-hop paths, where on the first hop a packet is transmitted to any available user. This user will act as a "relay" for the packet. The packet is stored in the buffer of the relay until an opportunity arises for it to be transmitted by the relay to its destination. Note that the notion of relaying is vitally important, as it allows throughput to be limited only by the rate at which a source encounters other users, rather than by the rate at which a source encounters its destination.

Cell Partitioned Relay Algorithm: Every timeslot and for each cell containing at least two users:

1. If there exists a source-destination pair within the cell, randomly choose such a pair (uniformly over all such pairs in the cell). If the source contains a new packet intended for that destination, transmit. Else remain idle.

2. If there is no source-destination pair in the cell, designate a random user within the cell as sender. Independently choose another user as receiver among the remaining users within the cell. With equal probability, randomly choose one of the two options:

- *Send a Relay packet to its Destination:* If the designated transmitter has a packet destined for the designated receiver, send that packet to the receiver. Else remain idle.
- *Send a New Relay Packet:* If the designated transmitter has a new packet (one that has never before been transmitted), relay that packet to the designated receiver. Else remain idle.

Because packets that have already been relayed are restricted from being transmitted to any user other than their destination, the above algorithm restricts all routes to 2-hop paths. The algorithm schedules packet transfer opportunities without considering queue backlog. Performance can be improved by allowing alternative scheduling opportunities in the case when no packet is available for the chosen transmission. However, the randomized nature of the algorithm admits a nice *decoupling* between sessions (see Fig. 6-3), where individual users see the network only as a source, destination, and intermediate relays, and transmissions of packets for other sources are reflected simply as random ON/OFF service opportunities.

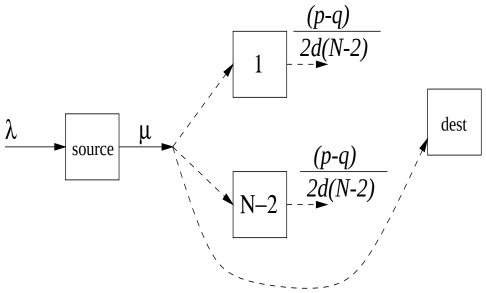


Figure 6-3: A decoupled diagram of the network as seen by the packets transmitted from a single user to the corresponding destination. Service opportunities at the first stage are Bernoulli with rate μ. Service at the second stage (relay) queues is Bernoulli with rate $(p-q)/(2d(N-2))$.

Theorem 13. *Consider a cell partitioned network (with N users and C cells) under the 2-hop relay algorithm, and assume that users change cells i.i.d. and uniformly over each cell every timeslot. If the exogenous input stream to user i is a Bernoulli stream of rate λ_i (where $\lambda_i < \mu$), then the total network delay W_i for user i traffic satisfies:*

$$\mathbb{E}\{W_i\} = \frac{N-1-\lambda_i}{\mu-\lambda_i} \tag{6.11}$$

where the capacity μ is defined in (6.1).

Proof. The proof uses reversibility of the first stage queue, and is provided in Chapter Appendix 6.A. $\square$

Note that the decoupling property of the cell partitioned relay algorithm admits a decoupled delay bound, so that the waiting time for user i packets depends only on the rate of the input stream for user i, and does not depend on the rate of other streams—even if the rate of these streams is greater than capacity. It follows that the network is stable with bounded delays whenever all input streams are less than capacity, i.e., when $\lambda_i < \mu$ for all users i. Thus, the relay algorithm achieves the capacity bound given in (6.1) of Theorem 1. It is perhaps counter-intuitive that the algorithm achieves capacity, as it often forces cells to remain idle even when choosing an alternate sender would allow for a packet to be delivered to its destination. The intuition is that all cases of idleness arise because a queue is empty, an event that becomes increasingly unlikely as load approaches capacity.

The form of the delay expression is worth noting. First note the classic $1/(\mu - \lambda_i)$ behavior, representing the asymptotic growth in delay as data rates are pushed towards the capacity boundary. Second, note that for a fixed loading value $\rho_i = \lambda_i/\mu$, delay is $O(N)$, growing linearly in the size of the network.

The exact delay analysis is enabled by the Bernoulli input assumption. If inputs are assumed to be Poisson, the delay theory of Chapter 2 can be used to develop a delay bound, and the bound for Poisson inputs is not considerably different from the exact expression for Bernoulli inputs given in (6.11). These results can also be extended to the case when the mobility model conforms to a Markovian random walk (see analytical discussion and simulation results in Sections 6.7 and 6.8).

6.3 Sending a Single Packet

In the previous subsection we showed that the cell partitioned relay algorithm yields an average delay of $O(N/(\mu - \lambda_i))$. Inspection of (6.11) shows that this $O(N)$ characteristic cannot be removed by decreasing the data rate λ. The following questions emerge: Can another scheduling algorithm be constructed which improves delay? What is the minimum delay the network can guarantee, and for what data rates is this delay obtainable? More generally, for a given data rate λ (assumed to be less than the system capacity μ), we ask:

What is the optimal delay bound, and what algorithm achieves this? In this section we present several fundamental bounds on delay performance, which establishes initial steps towards addressing these general questions. We assume throughout that the user/cell density d is a fixed value independent of N, and use $d = d^* = 1.7933$ in all numerical examples.

6.3.1 Scheduling Without Redundancy

Suppose that no redundancy is used: that is, packets are not duplicated and are held by at most one user of the network at any given time. Thus, a packet that is transmitted to another user is deleted from the memory storage of the transmitting user. Note that this is the traditional approach to data networking, and that the 2-hop relay algorithm is in this class.

Theorem 14. *Algorithms which do not use redundancy cannot achieve an average delay of less than $O(N)$.*

Proof. The minimum delay of any packet is computed by considering the situation where the network is empty and user 1 sends a single packet to user 2. It is easy to verify that relaying the packet cannot help, and hence the delay distribution is geometric with mean $C = N/d$. □

Hence, the relay algorithm not only achieves capacity, but achieves the optimal $O(N)$ delay performance among all strategies which do not use redundancy. Other policies which do not use redundancy can perhaps improve upon the delay coefficient, but cannot change the $O(N)$ characteristic.

6.3.2 Scheduling With Redundancy

Although redundancy cannot increase capacity, it can considerably improve delay. Clearly, the time required for a packet to reach the destination can be reduced by repeatedly transmitting this packet to many users of the network—improving the chances that some user holding an original or duplicate version of the packet reaches the destination. Consider any network algorithm (which may or may not use redundant packet transfers) that restricts packets to 2-hop paths.

Theorem 15. *No algorithm (with or without redundancy) which restricts packets to 2-hop paths can provide an average delay better than $O(\sqrt{N})$.*

Again consider the sending of a single packet from its source to its destination. Clearly the optimal scheme is to have the source send duplicate versions of the packet to new relays whenever possible, and for the packet to be relayed to the destination as soon as either the source or a duplicate-carrying relay enters the same cell as the destination.

Let T_N represent the time required to reach the destination under this optimal policy for sending a single packet. In the following lemma, we bound the limiting behavior[1] of $\mathbb{E}\{T_N\}$, proving Theorem 15.

Lemma 17. $e^{-d} \leq \lim_{N\to\infty} \frac{\mathbb{E}\{T_N\}}{\sqrt{N}} \leq \frac{2}{1-e^{-d}}$

Proof. Lemma 1 (a) *Lower Bound:* To prove the lower bound, note that during timeslots $\{1, 2, \ldots, \sqrt{N}\}$, there are fewer than $\sqrt{N}$ users holding the packet. Hence, $Pr[T_N > \sqrt{N}] \geq (1-1/C)^{\sqrt{N}\sqrt{N}}$ (where $(1-1/C)^{\sqrt{N}}$ is the probability that nobody within a group of $\sqrt{N}$ particular users enters the cell of the destination during a given timeslot). Recall that the user/cell density d is defined $d \triangleq N/C$. Thus:

$$\begin{aligned} \mathbb{E}\{T_N\} &\geq \mathbb{E}\left\{T_N | T_N > \sqrt{N}\right\} Pr[T_N > \sqrt{N}] \\ &\geq \sqrt{N}\left(1 - \frac{d}{N}\right)^N \to e^{-d}\sqrt{N} \end{aligned}$$

(b) *Upper Bound:* To prove the upper bound, note that $\mathbb{E}\{T_N\} \leq S_1 + S_2$, where S_1 represents the expected number of slots required to send out duplicates of the packet to $\sqrt{N}$ different users, and S_2 represents the expected time until one user within a group of $\sqrt{N}$ users containing the packet reaches the cell of the destination. The probability of the source meeting a new user is at least $1 - (1 - 1/C)^{N-\sqrt{N}}$ for every timeslot where fewer than $\sqrt{N}$ users have packets, and hence the average time to reach a new user is less than or equal to the inverse of this quantity (i.e, the average time of a geometric variable). Hence:

$$S_1 \leq \frac{\sqrt{N}}{1-(1-1/C)^{N-\sqrt{N}}} \to \frac{\sqrt{N}}{1-e^{-d}}$$

To compute S_2, note that $P(success)$, the probability that one of the $\sqrt{N}$ users reaches the destination during a slot, is given by the probability there is at least one other user in the same cell as the destination multiplied by the conditional probability that a packet-carrying

[1] Using the inequality $e^{\frac{-d^2}{N-d}}e^{-d} \leq \left(1 - \frac{d}{N}\right)^N \leq e^{-d}$, explicit bounds of the form $\alpha\sqrt{N} \leq \mathbb{E}\{T_N\} \leq \beta\sqrt{N}$ can also be derived.

user is present given there is at least one other user in the cell. The former probability is $1 - (1 - 1/C)^{N-1}$, and the latter is at least $\sqrt{N}/N$:

$$P(success) \geq \frac{1 - (1 - 1/C)^{N-1}}{\sqrt{N}} \rightarrow \frac{1 - e^{-d}}{\sqrt{N}} \tag{6.12}$$

Hence, $S_2 \leq \frac{\sqrt{N}}{1-e^{-d}}$. Summing S_1 and S_2 proves the result. □

An exact expression for the minimum delay $\mathbb{E}\{T_N\}$ is presented in Chapter Appendix 6.C by using a recursive formula. In Fig. 6-4 we plot the exact expression as a function of N together with the upper and lower bounds of Lemma 17 for the case $d = d^* = 1.7933$.

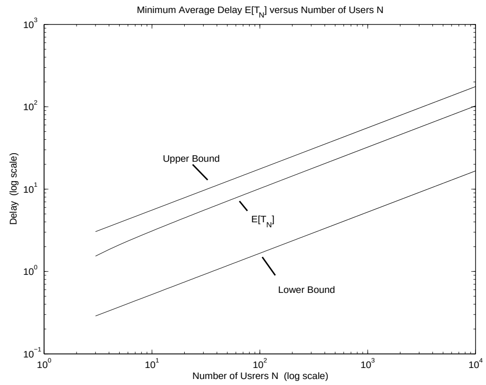


Figure 6-4: The exact minimum delay of a 2-hop scheduling scheme versus the number of users N at the optimal user/cell density d^*, together with the upper and lower bounds of Lemma 17. Curves are plotted on a $\log - \log$ scale and have slope 1/2, illustrating the $O(\sqrt{N})$ behavior.

6.3.3 Multi-User Reception

To increase the packet replication speed throughout the network, it is useful to allow a transmitted packet to be received by *all other users* in the same cell as the transmitter, not just the single intended recipient. This feature cannot increase capacity, but can considerably improve delay by enabling multiple duplicates to be injected into the network with just a single transmission. However, the $O(\sqrt{N})$ result of Theorem 15 cannot be overcome by introducing multi-user reception (see Chapter Appendix 6.D). For the remainder of this paper, we assume multi-user reception is available.

6.4 Scheduling for Delay Improvement

In the previous section an $O(\sqrt{N})$ delay bound was developed for redundant scheduling by considering a single packet for a single destination. Two complications arise when designing a general scheduling protocol using redundancy: (1) All sessions must use the network simultaneously, and (2) Remnant versions of a packet that has already been delivered to its destination create excess congestion and must somehow be removed.

Here we show that the properties of the 2-hop relay algorithm make it naturally suited to treat the multi-user problem. The second complication of excess packets is overcome by the following *in-cell feedback protocol*, in which a receiving node tells its transmitter which packet it is looking for before transmission begins. We assume all packets are labeled with *send numbers* SN, and the in-cell feedback is in the form of a *request number* RN delivered by the destination to the transmitter just before transmission. In the following protocol, each packet is retransmitted $\sqrt{N}$ times to distinct relay users.

In-Cell Feedback Scheme with $\sqrt{N}$ Redundancy: In every cell with at least two users, a random sender and a random receiver are selected, with uniform probability over all users in the cell. With probability 1/2, the sender is scheduled to operate in either 'source-to-relay' mode, or 'relay-to-destination' mode, described as follows:

1. *Source-to-Relay Mode:* The sender transmits packet SN, and does so upon every transmission opportunity until $\sqrt{N}$ replicas have been delivered to distinct users, or until the sender transmits SN directly to the destination. After such a time, the send number is incremented to $SN + 1$. If the sender does not have a new packet to send, remain idle.

2. *Relay-to-Destination Mode:* When a user is scheduled to transmit a relay packet to its destination, the following handshake is performed:

   - The receiver delivers its current RN number for the packet it desires.
   - The transmitter deletes all packets in its buffer destined for this receiver which have SN numbers lower than RN.
   - The transmitter sends packet RN to the receiver. If the transmitter does not have the requested packet RN, it remains idle for that slot.

Notice that the destination receives all packets in order, and that no packet is ever transmitted twice to its destination.

Theorem 16. *The In-Cell Feedback Scheme achieves the $O(\sqrt{N})$ delay bound, with user data rates of $O(1/\sqrt{N})$.*

More precisely, if all users receive exogenous data for their destinations according to a Poisson process of rate λ_i, the network can stably support rates $\lambda_i < \tilde{\mu}$, for the reduced network throughput $\tilde{\mu}$ given by:

$$\tilde{\mu} = \frac{\gamma_N \left(1 - e^{-d}\right)}{4(2+d)\sqrt{N}} \tag{6.13}$$

where γ_N is a sequence that converges to 1 as $N \to \infty$. Furthermore, average end-to-end delay $\mathbb{E}\left\{W_i\right\}$ satisfies:

$$\mathbb{E}\left\{W_i\right\} \leq \frac{1}{2} + \frac{1/\tilde{\mu}}{1-\rho_i}$$

where $\rho_i \triangleq \lambda_i/\tilde{\mu}$.

To prove the result, first note that when a new packet reaches the head of the line at its source queue, the time required for the packet to reach its destination is at most $T_N = S_1 + S_2$, where S_1 represents the time required for the source to send out $\sqrt{N}$ replicas of the packet, and S_2 represents the time required to reach the destination given that $\sqrt{N}$ users have the packet. Bounds on the expectations of S_1 and S_2 which are independent of the initial state of the network can be computed similarly to the proof of Lemma 17. The multi-user environment here simply acts to scale up these expectations by a constant factor due to collisions with other users (compare the upper bound of Lemma 17 with that given in (6.14) below). This factor does not scale with N because the average number of users in any cell is the finite number d. Indeed, in Chapter Appendix 6.E it is shown that:

$$\mathbb{E}\left\{T_N\right\} \leq \frac{4(2+d)\sqrt{N}}{\gamma_N(1-e^{-d})} \tag{6.14}$$

where γ_N is a function that converges to 1 as $N \to \infty$.

Note that the random variable T_N satisfies the *sub-memoryless* property: The residual time of T_N given that a fixed number of slots have already passed (without T_N expiring) is stochastically less than the original time T_N.[2] This is because the topology of the network is independent from slot to slot, and hence starting out with several duplicate packets already

[2] This is often called the “New Better than Used” property, see [119].

in the network yields statistically smaller delay than if no such initial duplicates are present.

The RN/SN handshake ensures that newer packets do not interfere with older packets, but that replication of the next packet waiting at the source queue begins on or before completion of the T_N "service time" for the current packet SN. Packets thus view the network as a single queue to which they arrive and are served sequentially. Although actual service times may not be i.i.d., they are all independently bounded by $\mathbb{E}\left\{T_N\right\}$, as are residual service times seen by a randomly arriving packet. This is sufficient to establish the following lemma, the proof of which is similar to the derivation of the standard P-K formula for average delay in an $M/G/1$ queue.

Lemma 18. *Suppose inputs to a single server queue are Poisson with sub-memoryless service times that are independently bounded by a value $\mathbb{E}\left\{T_N\right\}$. If the arrival rate is λ, where $\lambda < 1/\mathbb{E}\left\{T_N\right\}$, then average delay satisfies:*

$$\mathbb{E}\left\{W\right\} \leq \frac{1}{2} + \frac{\mathbb{E}\left\{T_N\right\}}{1-\rho} \tag{6.15}$$

where $\rho \triangleq \lambda\mathbb{E}\left\{T_N\right\}$. The expression on the right hand side of the above inequality is the standard expression for delay in an M/M/1 queue with i.i.d. service times T_N that are restricted to start on slot boundaries.

Proof. Consider a single packet arriving from a Poisson stream, and let W_q represent the time this packet spends waiting in the queue before reaching the server. We have:

$$W_q = \sum_{i=1}^{N_q} X_i + R \tag{6.16}$$

where N_q is the number of packets already in the queue, $\{X_i\}$ are the service times of these packets, and R represents the residual time until either the packet currently in the server finishes its service, or (if the system is empty) the start of a new timeslot. Note that $\mathbb{E}\left\{R\right\} \leq \rho_{actual}\mathbb{E}\left\{T_N\right\} + (1-\rho_{actual})\frac{1}{2}$, where ρ_{actual} represents the probability that the system is busy with a packet already in service. From Little's Theorem we have that $\rho_{actual} = \lambda\mathbb{E}\left\{X\right\}$, where $\mathbb{E}\left\{X\right\}$ represents the average service time of a generic packet. Because $\mathbb{E}\left\{X\right\} \leq \mathbb{E}\left\{T_N\right\}$, it follows that $\rho_{actual} \leq \rho$. Because $\mathbb{E}\left\{T_N\right\} \geq 1/2$, we can further increase the upper bound on $\mathbb{E}\left\{R\right\}$ by replacing ρ_{actual} with ρ, yielding $\mathbb{E}\left\{R\right\} \leq$

$\rho\mathbb{E}\left\{T_N\right\} + (1-\rho)\frac{1}{2}$. Taking expectations of (6.16) thus yields:

$$\begin{aligned}
\mathbb{E}\left\{W_q\right\} &= \mathbb{E}_{N_q}\left[\sum_{i=1}^{N_q} \mathbb{E}\left\{X_i \left| N_q\right.\right\}\right] + \mathbb{E}\left\{R\right\} \\
&\leq \mathbb{E}_{N_q}\left[\sum_{i=1}^{N_q} \mathbb{E}\left\{T_N\right\}\right] + \rho\mathbb{E}\left\{T_N\right\} + (1-\rho)\frac{1}{2} \\
&= \mathbb{E}\left\{N_q\right\}\mathbb{E}\left\{T_N\right\} + \rho\mathbb{E}\left\{T_N\right\} + (1-\rho)\frac{1}{2} \\
&= \lambda\mathbb{E}\left\{W_q\right\}\mathbb{E}\left\{T_N\right\} + \rho\mathbb{E}\left\{T_N\right\} + (1-\rho)\frac{1}{2} \qquad (6.17)
\end{aligned}$$

where (6.17) follows from Little's Theorem. We thus have:

$$\mathbb{E}\left\{W_q\right\} \leq \frac{\rho\mathbb{E}\left\{T_N\right\}}{1-\rho} + \frac{1}{2}$$

Noting that the total waiting time $\mathbb{E}\left\{W\right\}$ satisfies $\mathbb{E}\left\{W\right\} \leq \mathbb{E}\left\{W_q\right\} + \mathbb{E}\left\{T_N\right\}$ yields the result. □

Defining $\tilde{\mu} \triangleq 1/\mathbb{E}\left\{T_N\right\}$ and using (6.14) proves Theorem 16.

6.5 Multi-Hop Scheduling and Logarithmic Delay

To further improve delay, we can remove the 2-hop restriction and consider schemes which allow for multi-hop paths. Here, a simple flooding protocol is developed and shown to achieve $O(\log(N))$ delay at the expense of further reducing throughput.

To achieve $O(\log(N))$ delay, consider the situation in which a single packet is delivered over an empty network. At first, only the source user contains the packet. The packet is transmitted and received by all other users in the same cell as the source. In the next timeslot, the source as well as all of the new users containing the packet transmit in their respective cells, and so on. If all duplicate-carrying users enter distinct cells every timeslot, and each of these users delivers the packet to exactly one new user, then the number of users containing the packet grows geometrically according to the sequence $\{1, 2, 4, 8, 16, \ldots\}$. The actual growth pattern may deviate from this geometric sequence somewhat, due to multiple users entering the same cell, or to users entering cells that are devoid of other users. However, it can be shown that the *expected* growth is geometric provided that the number of packet-holding users is less than $N/2$.

Define the total time to reach all users as $T_N = S_1 + S_2$, where S_1 and S_2 respectively represent the time required to send the packet to at least $N/2$ users, and the time required to deliver the packet to the remaining users given that at least $N/2$ users initially hold the packet.

Lemma 19. *Under the above algorithm of flooding the network with a single packet, for any network size $N \geq \max\{d, 2\}$, the expected time $\mathbb{E}\{T_N\}$ for the packet to reach every user satisfies $\mathbb{E}\{T_N\} \leq \mathbb{E}\{S_1\} + \mathbb{E}\{S_2\}$, where:*

$$\begin{aligned}\mathbb{E}\{S_1\} &\leq \frac{\log(N)\,(1+d/2)}{\log(2)(1-e^{-d/2})} \\ \mathbb{E}\{S_2\} &\leq 1 + \frac{2}{d}(1+\log(N/2))\end{aligned} \tag{6.18}$$

Proof. The proof is given in Chapter Appendix 6.F. □

6.5.1 Fair Packet Flooding Protocol

Thus, $O(\log(N))$ delay is achievable when sending a single packet over an empty network. To enable $O(\log(N))$ delay in the general case where all sessions are active and share the network resources, we construct a flooding protocol in which the oldest packet that has not been delivered to all users is selected to dominate network resources. We assume that packets are sequenced with SN numbers as before. Additionally, packets are stamped with the timeslot t in which they arrived.

Fair Packet Flooding Protocol: Every timeslot and in each cell, users perform the following: Among all packets contained in at least one user of the cell but which have never been received by some other user in the same cell, choose the packet p which arrived earliest (i.e., it has the smallest timestamp t_p). If there are ties, choose the packet from the session i which maximizes $(t_p + i) \mod N$. Transmit this packet to all other users in the cell. If no such packet exists, remain idle.

The above protocol is "fair" in that in case of ties, session i packets are given top priority every N timeslots. Other schemes for choosing which packet to dominate the network could also be considered. Delay under the above protocol can be understood by comparing the network to a single queue with N input streams of rates $\lambda_1, \lambda_2, \ldots, \lambda_N$ which share a single server with service times T_N. Note that the T_N service time is also sub-memoryless. Thus, from Lemma 18, we have:

Theorem 17. *For Poisson inputs with rates λ_i for each source i, the network under the fair flooding protocol is stable whenever $\sum_i \lambda_i < 1/\mathbb{E}\{T_N\}$, with average end-to-end delay satisfying:*

$$\mathbb{E}\{W\} \leq \frac{1}{2} + \frac{\mathbb{E}\{T_N\}}{1-\rho} \tag{6.19}$$

where $\rho \triangleq \sum_i \lambda_i \mathbb{E}\{T_N\}$, and $\mathbb{E}\{T_N\} = \mathbb{E}\{S_1\} + \mathbb{E}\{S_2\}$. Note that $O(\log(N))$ bounds on $\mathbb{E}\{S_1\}$ and $\mathbb{E}\{S_2\}$ are given in Lemma 19. Thus, when all sources have identical input rates λ, stability and logarithmic delay is achieved when $\lambda = O(\frac{1}{N\log(N)})$. $\Box$

Note that the flooding algorithm easily allows for *multicast sessions*, where data of rate λ is delivered from each source to *all other users.* One might expect that delay can be improved if we only design for unicast. However, it is shown in Chapter Appendix 6.G that logarithmic delay is the best possible for any strategy at any data rate. Hence, communication for unicast or multicast is the same in the logarithmic delay regime. In the next section, we address the following question: Is it possible to increase data rates via some other protocol while maintaining the same average delay guarantees?

6.6 Fundamental Delay/Rate Tradeoffs

Considering the capacity achieving 2-hop relay algorithm, the 2-hop algorithm with $\sqrt{N}$ redundancy, and the packet flooding protocol, we have the following achievable delay/capacity performance tradeoffs.

scheme	capacity	delay
no redundancy	$O(1)$	$O(N)$
redundancy 2-hop	$O(1/\sqrt{N})$	$O(\sqrt{N})$
redundancy multi-hop	$O(\frac{1}{N\log(N)})$	$O(\log(N))$

A simple observation reveals that *delay/rate* $\geq O(N)$ for each of these three protocols. In this section, we establish that this is in fact a necessary condition. Thus, performance of each given protocol falls on or near the boundary of a *fundamental rate-delay curve* (see Fig. 6-1).

Consider a network with N users, and suppose all users receive packets at the same rate λ. A control protocol which makes decisions about scheduling, routing, and packet retransmissions is used to stabilize the network and deliver all packets to their destinations while maintaining an average end-to-end delay less than some threshold $\overline{W}$.

Theorem 18. $\frac{\overline{W}}{\lambda} \geq O(N)$ *is a necessary condition for any conceivable routing and scheduling protocol which stabilizes the network with input rates* λ *while maintaining bounded average end-to-end delay* $\overline{W}$.

In particular, we have:

$$\frac{\overline{W}}{\lambda} \geq \frac{N-d}{4d}(1-\log(2))$$

where log() *denotes the natural logarithm.*

Proof. Suppose the input rate of each of the N sessions is λ, and there exists some stabilizing scheduling strategy which ensures an end-to-end delay of $\overline{W}$. In general, the end-to-end delay of packets from individual sessions could be different, and we define $\overline{W}_i$ as the resulting average delay of packets from session i. We thus have:

$$\overline{W} = \frac{1}{N}\sum_i \overline{W}_i \tag{6.20}$$

Let $\overline{R}_i$ represent the average *redundancy* associated with packets from session i. That is, $\overline{R}_i$ is the number of users who receive a copy of an individual packet during the course of the network control operation, averaged over all packets from session i. Note that all packets are eventually received by the destination, so that $\overline{R}_i \geq 1$. Additional redundancy could be introduced by multi-hop routing, or by any packet replication effort that is used to achieve stability and/or improve delay. The average number of successful packet receptions per timeslot is thus given by the quantity $\lambda \sum_{i=1}^{N} \overline{R}_i$. Because each of the N users can receive at most 1 packet per timeslot, we have:

$$\lambda \sum_{i=1}^{N} \overline{R}_i \leq N \tag{6.21}$$

Now consider a single packet p which enters the network from session i. This packet has an average delay of $\overline{W}_i$ and an average redundancy of $\overline{R}_i$. Let random variables W_i and R_i

represent the actual delay and redundancy for this packet. We have:

$$\begin{aligned}\overline{W}_i &\geq \mathbb{E}\left\{W_i \mid R_i \leq 2\overline{R}_i\right\} Pr[R \leq 2\overline{R}_i] \\ &\geq \mathbb{E}\left\{W_i \mid R_i \leq 2\overline{R}_i\right\} \frac{1}{2} \qquad (6.22)\end{aligned}$$

where (6.22) follows because $Pr[R_i \leq 2\overline{R}_i] \geq \frac{1}{2}$ for any non-negative random variable R_i.

Note that the smallest possible delay for packet p is the time required for one of its carriers to enter the same cell as the destination. Consider now a virtual system in which there are $2\overline{R}_i$ users initially holding packet p, and let Z represent the time required for one of these users to enter the same cell as the destination. Every timeslot the "success probability" for this system is $\phi \triangleq 1 - (1 - \frac{1}{C})^{2\overline{R}_i}$, so that $\mathbb{E}\left\{Z\right\} = 1/\phi$. Although there are more users holding packet p in this system, the expectation of Z does not necessarily bound $\mathbb{E}\left\{W_i \mid R \leq 2\overline{R}_i\right\}$ because conditioning on the event $\{R_i \leq 2\overline{R}_i\}$ might skew the probabilities associated with the user mobility process. However, because the event $\{R_i \leq 2\overline{R}_i\}$ occurs with probability at least 1/2, we obtain the following bound:

$$\mathbb{E}\left\{W_i \mid R_i \leq 2\overline{R}_i\right\} \geq \inf_{\Theta} \mathbb{E}\left\{Z \mid \Theta\right\}$$

where the conditional expectation is minimized over all conceivable events Θ which occur with probability greater than or equal to 1/2.

We now *stochastically couple* the Z variable to an exponential variable $\tilde{Z}$ with rate $\gamma \triangleq \log(1/(1-\phi))$. The variable $\tilde{Z}$ is *stochastically less* than Z because $Pr[\tilde{Z} > \omega] \leq Pr[Z > \omega]$ for all ω (see [119] for a discussion of stochastic coupling and stochastic inequalities). It follows that $\mathbb{E}\left\{W_i \mid R_i \leq 2\overline{R}_i\right\} \geq \inf_{\Theta} \mathbb{E}\left\{\tilde{Z} \mid \Theta\right\}$, and the minimizing event Θ is clearly the event $\{\tilde{Z} \leq \omega\}$, where ω is the smallest value such that $Pr[\tilde{Z} \leq \omega] \geq \frac{1}{2}$. Thus, $Pr[\tilde{Z} > \omega] = e^{-\gamma\omega} = 1/2$, and hence $\omega = \frac{\log(2)}{\gamma}$. Conditioning on this event, we have:

$$\begin{aligned}\mathbb{E}\left\{\tilde{Z} \mid \tilde{Z} \leq \omega\right\} &= \frac{\mathbb{E}\left\{\tilde{Z}\right\} - \mathbb{E}\left\{\tilde{Z} \mid \tilde{Z} > \omega\right\} Pr[\tilde{Z} > \omega]}{Pr[\tilde{Z} \leq 1/2]} \\ &= \frac{\frac{1}{\gamma} - (\omega + \frac{1}{\gamma})\frac{1}{2}}{1/2} = \frac{1 - \log(2)}{\gamma} \qquad (6.23)\end{aligned}$$

From the definitions of γ and ϕ, we have $\gamma = \log\left(1/(1 - \frac{1}{C})^{2\overline{R}_i}\right) = 2\overline{R}_i \log(1 + \frac{1}{C-1})$.

Because $\log(1+x) \leq x$ for any x, we have $\gamma \leq 2\overline{R}_i/(C-1)$. Using this bound together with (6.23) and (6.22), we have:

$$\overline{W}_i \geq \frac{1-\log(2)}{2\gamma} \geq \frac{(C-1)(1-\log(2))}{4\overline{R}_i}$$

Summing this inequality over all i, we have:

$$\begin{aligned}\overline{W} = \frac{1}{N}\sum_{i=1}^{N}\overline{W}_i &\geq \frac{(C-1)\,(1-\log(2))}{4}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\overline{R}_i} \\ &\geq \frac{(C-1)\,(1-\log(2))}{4\frac{1}{N}\sum_{i=1}^{N}\overline{R}_i} \qquad (6.24)\end{aligned}$$

where (6.24) follows from Jensen's inequality, noting that the function $1/R$ is convex. Combining (6.24) and (6.21), we have:

$$\overline{W} \geq \frac{(C-1)\,(1-\log(2))\,\lambda}{4} = \frac{(N-d)\,(1-\log(2))\,\lambda}{4d}$$

Hence, the delay/rate characteristics necessarily satisfy the inequality $\frac{\overline{W}}{\lambda} \geq O(N)$, proving the theorem. □

The fact that $delay/rate \geq O(N)$ establishes a fundamental performance tradeoff, illustrating that no scheduling and routing algorithm can simultaneously yield low delay and high throughput. The $O(N)$ and $O(\sqrt{N})$ scheduling algorithms provided here meet this bound with equality, and the $O(\log(N))$ algorithm lies above the bound by a factor of $O(\log^2(N))$ (see table above). Note that the "redundancy 2-hop" entry in the table demonstrates that a cell partitioned mobile network can emulate the delay/capacity performance of a Gupta-Kumar static network [58], [57]. It is interesting to explore whether this result generalizes to other mobility models.

6.7 Non-i.i.d. Mobility Models

The analysis developed here for the i.i.d. mobility model can be used to bound the performance of a system with a Markovian mobility model. Instead of performing control actions on the network every slot, we decompose the network into a set of K parallel subnetworks. Packets are considered to be of 'type-k' if they arrive during a timeslot t such that $t \mod K = k$. On such timeslots, only control actions on type-k packets take place. The value of K is chosen suitably large to ensure that the user location distribution after K slots is within a constant factor of its steady state value. Specifically, if K is chosen such that, regardless of the initial configuration of users, the probability that two given users are in the same cell after K slots is at least $\frac{1}{2C}$, then delay under the three schemes is bounded by $O(KN), O(K\sqrt{N})$, and $O(K\log(N))$, respectively. The $O(KN)$ result for the 2-hop relay algorithm (with no redundancy) follows by using Lemma 2 in a drift argument similar to that already given in Theorem 8 in Chapter 4. The $O(K\sqrt{N})$, and $O(K\log(N))$ bounds follow by literally repeating the same arguments used for the $\sqrt{N}$ redundancy algorithm and the Fair-Flooding algorithm on a K slot basis.

However, it is possible that alternative scheduling schemes could yield lower delay. Indeed, in the next section it is shown through simulation that applying the 2-hop relay algorithm and the $\sqrt{N}$ redundancy algorithm exactly as before (without the K-subchannel decomposition) yields similar performance for both i.i.d. and non-i.i.d. mobility.

6.8 Simulation Results

Here we compare the average delay obtained through both analysis and simulation as the network is scaled. We consider a network with cells given by an $M \times M$ grid as shown in Fig. 6-1. The number of cells C is equal to M^2 (where M is varied between 3 and 15 for simulations), and the number of users N is chosen as the even integer for which N/C most accurately approximates the optimal user/cell density value $d^* = 1.7933$.

In Fig. 6-5, plots of average end-to-end delay versus the number of users N are provided for the 2-hop relay algorithm and the $O(\sqrt{N})$ redundancy algorithm for both an i.i.d. and a non-i.i.d. mobility model. In the i.i.d. mobility model, users choose new cells uniformly over all cells in the network. In the non-i.i.d. model, each user chooses a new cell every timeslot according to the following Markovian dynamics: With probability α the user stays in the same cell, and else it moves to an adjacent cell to the North, South, East, or West, with each direction equally likely. In the case where a user is on the edge of the network and is selected to move in an infeasible direction, it stays in its place. Using standard random walk theory it is easy to verify that, in steady state, such a Markov model leaves users independently and uniformly distributed over all cells, as the stationary equation for the

Markov chain is satisfied when all cell locations have equal probability [77] [120] [49].[3] In particular, if π_i represents the steady state probability of a particular cell i, we have:

$$\pi_i = \pi_i\alpha + \pi_a\frac{(1-\alpha)}{4} + \pi_b\frac{(1-\alpha)}{4} + \pi_c\frac{(1-\alpha)}{4} + \pi_d\frac{(1-\alpha)}{4}$$

where $\pi_a, \pi_b, \pi_c, \pi_d$ represent steady state probabilities for other cells, possibly including cell i. In the case when cell i is an interior cell, it has four distinct neighbors a, b, c, d. In the case when it is an edge cell with three neighbors a, b, c, we set $d = i$ (so that cell i is its own neighbor). In the case when cell i is a corner cell with 2 neighbors a and b, we set $c = d = i$. Clearly these steady state equations are satisfied when the π_i probabilities are set to $1/C$ for all i.[4] Therefore, the network capacity μ is the same for both the i.i.d. mobility model and the non-i.i.d. mobility model, and is given by $\mu = (p+q)/(2d)$ as described in Theorem 11. In the simulation results we set the α parameter of the non-i.i.d. model to $\alpha = 1/2$.

For the capacity achieving 2-hop relay algorithm, the data rate λ into each user is fixed at 80% of the network capacity μ (given in Theorem 1), so that $\rho = \lambda/\mu = 0.8$. The top three curves for average delay in Fig. 6-5 respectively represent the exact analytical delay for i.i.d. mobility, the simulated performance of the i.i.d. mobility model, and the simulated performance of the Markovian mobility model. Note that the simulation curve for the i.i.d. mobility model is almost indistinguishable from the analytical curve $\mathbb{E}\left\{W\right\} = \frac{N-1-\lambda}{\mu-\lambda}$. The curves are plotted on a $\log\log$ scale and have a slope of 1, indicating $O(N)$ delay. The delay curve for Markovian mobility is situated slightly above the curve for i.i.d. mobility, and also has a slope of 1. This suggests that for Markovian mobility, delay is increased by a constant multiplicative factor but remains $O(N)$.

Results for the $\sqrt{N}$ redundancy protocol are also shown in the figure. Data rates λ are set to the value $\lambda = 0.8\tilde{\mu}$, where $\tilde{\mu}$ is given in (6.13). Note that, unlike the network capacity μ, the throughput $\tilde{\mu}$ decreases as $O(1/\sqrt{N})$. The analytical upper and lower bounds on delay for i.i.d. mobility are shown in the figure, each having a slope of 1/2 indicating $O(\sqrt{N})$ growth (note that the lower bound represents the delay of sending just

[3] Another easy proof of this fact is to note that the steady state Markov equations for this chain are identical to the steady state equations for the Markov chain describing a random walk on an $M \times M$ *torus* (see Fig. 4-7), where transitions beyond the edge of the network are wrapped around to the appropriate cell on the opposite side.

[4] Similar results hold when the random walk has a different behavior at the edges. In particular, if the direction is chosen uniformly over all feasible directions, then the interior cells will have equal probability but the edge cells will have a different probability.

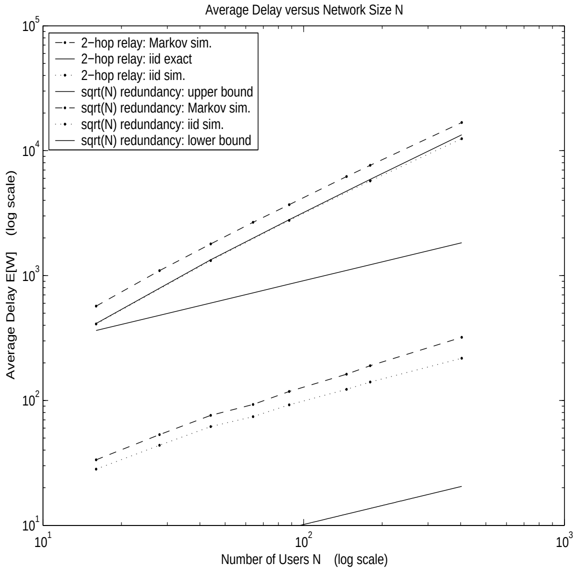


Figure 6-5: Average delay versus the number of users N for the 2-hop relay algorithm and the $\sqrt{N}$ redundancy algorithm.

a single packet). The simulation performance for i.i.d. mobility is shown in the figure and is situated between the upper and lower bounds. The upper bound is larger than the simulated curve by approximately a factor of 10, suggesting that tighter bounds could be obtained through a more detailed analysis. The slope of the simulation curve varies between 5/8 and 1/2. However, because delay is upper and lower bounded by functions of $O(\sqrt{N})$, the average slope would converge to 1/2 if the graph were extended. Simulation of the Markovian mobility model is also provided, and the curve again lies slightly above the i.i.d. mobility curve. This suggests that delay under the Markovian model is close to $O(\sqrt{N})$.

Experiments to simulate the performance of the $O(\log(N))$ scheme were not performed. However, for this case, we would expect a discrepancy between the i.i.d. mobility model and the non-i.i.d. mobility model. Indeed, although the i.i.d. mobility model yields logarithmic delay, the delay under a Markovian mobility model would likely be closer to $O(\sqrt{N})$ due to the time required for a user to travel from one side of the network to the other.

6.9 Chapter Summary

The results of this chapter for the first time present a multi-hop, multi-user system for which a relatively complete network theory can be developed. Exact expressions for network capacity were derived, and a fundamental rate-delay curve was established, representing performance bounds on throughput and end-to-end network delay for any conceivable routing and scheduling policy.

Delay analysis for the network was facilitated using a simple i.i.d. user mobility model. Under this model, an exact expression for end-to-end delay which includes the full effects of queueing was established for the capacity achieving 2-hop relay algorithm. Two other protocols which (necessarily) use redundant packet transfers were provided and shown to improve delay at the expense of reducing throughput. The rate-delay performance of these schemes was shown to lie on the boundary of the fundamental performance curve $delay/rate \geq O(N)$. Analysis of general mobility models can be understood in terms of this i.i.d. analysis, where delay bounds can be scaled by the factor K, representing the number of slots required between sampling points for samples of user locations to look nearly i.i.d.. Furthermore, simulation results suggest that $O(\sqrt{N})$ delay can be achieved for networks with Markovian mobility, as the delay for such systems closely follows the delay curve for a system with i.i.d. mobility.

This inspires a rich set of questions concerning the fundamental limits of data networks. We believe that the condition $delay/rate \geq O(N)$ is necessary for general classes of mobile wireless networks, and that the $(rate, delay) = \Big(O(1/\sqrt{N}), O(\sqrt{N})\Big)$ operating point is always achievable. Such conjectures can perhaps be established using analytical techniques similar to those created here.

Chapter Appendix 6.A — Exact Delay Analysis of the 2-Hop Relay Algorithm

Proof of Delay Bound in Theorem 3: The exact end-to-end network delay under the 2-hop relay algorithm with Bernoulli inputs and i.i.d. mobility is $\mathbb{E}\{W_i\} = \frac{N-1-\lambda_i}{\mu-\lambda_i}$.

Proof. A decoupled view of the network as perceived by a single user i is illustrated in Fig. 2. Because of the i.i.d. mobility, the source user can be represented as a Bernoulli/Bernoulli queue, where every timeslot a new packet arrives with probability λ, and a service opportunity arises with some fixed probability μ. We first show that $\mu = \frac{p+q}{2d}$. The Bernoulli nature of the server process implies that the transmission probability μ is equal to the time average rate of transmission opportunities of source i.[5] Hence, we have $\mu = r_1+r_2$, where r_1 represents the rate at which the source is scheduled to transmit directly to the destination, and r_2 represents the rate at which it is scheduled to transmit to one of its relay users. The cell partitioned relay algorithm schedules transmissions into and out of the relay nodes with equal probability, and hence r_2 is also equal to the rate at which the relay nodes are scheduled to transmit to the destination. The total rate of transmission opportunities over the network is thus $N(r_1 + 2r_2)$. A transmission opportunity occurs in any given cell with probability p, and hence:

$$Cp = N(r_1 + 2r_2) \tag{6.25}$$

Recall that q is the probability that a given cell contains a source-destination pair. Because the cell partitioned relay algorithm schedules the single-hop 'source-to-destination' transmissions whenever possible, the rate r_1 satisfies:

$$Cq = Nr_1 \tag{6.26}$$

It follows from (6.26) that $r_1 = q/d$, and hence by (6.25) we infer that $r_2 = \frac{p-q}{2d}$. The total rate of transmissions out of the source node is thus given by $\mu = r_1 + r_2 = \frac{p+q}{2d}$.

The source is thus a Bernoulli/Bernoulli queue with input rate λ and server probability μ, having an expected number of packets given by $\overline{L}_{source} = \frac{\rho(1-\lambda)}{1-\rho}$, where $\rho \triangleq \lambda/\mu$ [38]. This queue is *reversible* ([49], [38]), and so the output process is also a Bernoulli stream of rate

[5] A *transmission opportunity* arises when a user is selected to transmit to another user, and corresponds to a service opportunity in the Bernoulli/Bernoulli queue. Such opportunities arise with probability μ every timeslot, independent of whether or not there is a packet waiting in the queue.

λ.

A given packet from this output process is transmitted to the *first relay node* with probability $\frac{r_2}{\mu(N-2)}$ (because with probability r_2/μ the packet is intended for a relay node, and each of the $N-2$ relay nodes are equally likely). Hence, every timeslot this relay independently receives a packet with probability $\tilde{\lambda} = \frac{\lambda r_2}{\mu(N-2)}$. The relay node is scheduled for a potential packet transmission to the destination with probability $\tilde{\mu} = \frac{r_2}{(N-2)}$ (because a 'relay-to-destination' opportunity arises with probability r_2, and arises from exactly one of the $N-2$ relay nodes with equal probability). However, packet arrivals and transmission opportunities are mutually exclusive events in the relay node. It follows that the discrete time Markov Chain for queue occupancy in the relay node can be written as a simple birth-death chain which is identical to the chain of a continuous time M/M/1 queue with input rate $\tilde{\lambda}$ and service rate $\tilde{\mu}$ (where $\tilde{\lambda}/\tilde{\mu} = \rho$). This holds for each relay node, and the resulting occupancy at any relay is thus: $\overline{L}_{relay} = \frac{\rho}{1-\rho}$. From Little's Theorem, the total network delay is $\overline{W}_i = \left[\overline{L}_{source} + (N-2)\overline{L}_{relay}\right]/\lambda$, which proves the theorem. □

Chapter Appendix 6.B — Heterogeneous Data Rates

Proof of Theorem 12: Here we prove that for heterogeneous data rates (λ_{ij}) such that there are at most K nonzero λ_{ij} entries in each row i, the symmetric capacity region satisfies:

$$\begin{aligned}\sum_j \lambda_{ij} &\leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N)\ \forall i \\ \sum_i \lambda_{ij} &\leq \frac{(1-e^{-d}-de^{-d})}{2d} + O(K/N)\ \forall j\end{aligned}$$

Before proving the theorem, we first note that whenever $N > d$, we have:

$$e^{\frac{-d^2}{N-d}}e^{-d} \leq \left(1-\frac{d}{N}\right)^N \leq e^{-d}$$

which can be proven by taking the logarithm of the above inequality and using the fact that $\log(1+x) \leq x$ whenever $x > -1$.[6] The difference between the upper and lower bounds is thus $e^{-d}\left(1-e^{\frac{-d^2}{N-d}}\right)$. Using the Taylor expansion $e^{\frac{-d^2}{N-d}} = 1+\frac{-d^2}{N-d}+O(1/N^2)$ reveals that this difference is $O(1/N)$, and hence $(1-\frac{d}{N})^N = e^{-d}+O(1/N)$.

Proof. (Necessity) The proof that the above inequalities are necessary conditions for stability is similar to the proof of Theorem 11, where the equation (6.4) is replaced by:

$$\frac{1}{T}\sum_{h=1}^{\infty} X_h(T) \geq \sum_i \sum_j \lambda_{ij} - \epsilon$$

Repeating the same argument as in Theorem 11, it follows that [compare with (6.8)]:

$$\frac{1}{N}\sum_i \sum_j \lambda_{ij} \leq \frac{Cp+C\tilde{q}+2\epsilon}{2N} = \frac{p}{2d}+\frac{\tilde{q}}{2d}+\frac{\epsilon}{N}$$

where p is the probability that at least two users are within a cell (given in (6.2)), and $\tilde{q}$ is the probability that there exists a source-destination pair within the cell. Note that $\tilde{q}$ may be different from the value of q given in (6.3) because of the different sets of source-destination pairs. However, because each user i has at most K destination nodes to consider, the union bound implies that the probability of any particular user entering a given cell along with at least one of its destinations is less than or equal to $\frac{1}{C}\frac{K}{C}$, so that $\tilde{q} \leq \frac{N}{C}\frac{K}{C} = O(K/N)$.

[6] Note that $\frac{-d}{N-d} \leq -\log\left(1+\frac{d}{N-d}\right) = \log\left(1-\frac{d}{N}\right) \leq \frac{-d}{N}$.

The probability p that at least two users are within a cell satisfies:

$$\begin{aligned} p &= 1-\left(1-\frac{d}{N}\right)^{N}-d\left(1-\frac{d}{N}\right)^{N-1} \\ &= 1-e^{-d}-de^{-d}+O(1/N) \end{aligned}$$

Hence, $\frac{1}{N}\sum_i\sum_j \lambda_{ij} \leq \frac{1-e^{-d}-de^{-d}}{2d} + O(K/N)$. This together with the fact that no user sends or receives more than any other proves the result. □

For sufficiency, we consider a 2-hop routing scheme, where data is routed uniformly over all relay nodes on the first hop regardless of its destination. We note that such a *traffic uniformization* scheme is conceptually similar to the 2-stage switch scheduling algorithm developed for $N \times N$ packet switches in [79], where packets are randomly assigned to output ports at the first stage so that traffic is uniform at the second stage.

Proof. (Sufficiency) From the Network Capacity Theorem developed in [102] [111], we know that it is sufficient to describe a transmission strategy yielding long term node-to-node packet exchange rates μ_{ij} together with a set of multi-commodity flows which route all data to their destinations without exceeding these rates on any link (i, j). Consider the strategy of choosing a transmitter and receiver in each cell completely randomly over all user pairs. As the expected number of packet transfer opportunities over the network is Cp opportunities per slot, the total rate of opportunities between any two links is $\mu_{ij} = \frac{Cp}{N(N-1)}$.

Suppose now the rate of exogenous data arriving to any node i is identically λ (for some data rate λ), as is the sum rate of data entering the network destined for any node j, so that $\sum_i \lambda_{ij} = \sum_j \lambda_{ij} = \lambda$ for all i, j. (Any smaller rate matrix which does not sum to λ in every row and column can be increased to a matrix which does have this property). Consider the 2-hop routing scheme were exogenous packets at a source are routed randomly and uniformly to any available relay node, and these packets are then transferred from relay to destination. Because on the first hop the algorithm routes data independently of its destination, the incoming traffic to the relay nodes is uniformly distributed, so that each relay receives data destined for node j at a rate $\lambda/(N-1)$ for all destinations j.

The total rate of traffic flowing over any link from i to j is thus $2\lambda/(N-1)$ (where a stream of total rate $\lambda/(N-1)$ flows from i to j due to packets from source i being relayed to j, and data of rate $\lambda/(N-1)$ flows from i to j due to traffic being relayed from i to destination

j). This traffic satisfies the link constraint provided that $2\lambda/(N-1) \leq \mu_{ij} = \frac{Cp}{N(N-1)}$, or equivalently that $\lambda \leq \frac{p}{2d}$. Thus, any rate matrices (λ_{ij}) satisfying $\sum_i \lambda_{ij} \leq \frac{p}{2d}$ for all j and $\sum_j \lambda_{ij} \leq \frac{p}{2d}$ for all i are within the capacity region, where $\frac{p}{2d} = \frac{1-e^{-d}-de^{-d}}{2d} + O(1/N)$. □

Chapter Appendix 6.C — Minimum Delay for 2-Hop Routing

Here we derive a recursive formula for the minimum average delay for sending a single packet from source to destination in the case when routing is restricted to 2-hop paths. We assume that multi-user reception is not available, so that at most one user per cell can receive a packet during a single timeslot.

The minimum delay algorithm transfers the packet to its destination whenever the source or a duplicate-carrying relay is in the same cell as the destination, and otherwise schedules the source to deliver a duplicate version of the packet to a new user whenever possible. Let $\mathbb{E}\{T_N\}$ represent the expected time for the packet to reach the destination. The value of $\mathbb{E}\{T_N\}$ can be computed recursively by defining variables $X_1, X_2, \ldots, X_{N-1}$, where X_k represents the expected time for the packet to reach its destination given that k users are carrying duplicates of the packet. The probability that a particular user does not move to the same cell as the destination during a timeslot is $(1 - 1/C)$. Therefore, the probability that at least one user among a group of k users *does* reach the destination is $1-(1-1/C)^k$. Note that because all paths are restricted to 2 hops, the number of users holding a duplicate version of the packet increases by at most one every slot. This number stays the same if the source user does not visit anyone new, and if (independently) all $k-1$ other users holding the packet do not visit the destination. Considering the Markov nature of the problem, we have the following transition probabilities for each state $k \in \{1, \ldots, N-2\}$:

$$\begin{aligned}
Pr[k \rightarrow end] &= 1 - \left(1 - \frac{1}{C}\right)^k \\
Pr[k \rightarrow k] &= \left(1 - \frac{1}{C}\right)^{N-k}\left(1 - \frac{1}{C}\right)^{k-1} = \left(1 - \frac{1}{C}\right)^{N-1} \\
Pr[k \rightarrow k+1] &= 1 - Pr[k \rightarrow end] - Pr[k \rightarrow k]
\end{aligned}$$

In state $k = N-1$, the remaining time to finish is a geometric variable with probability $1 - \left(1 - \frac{1}{C}\right)^{N-1}$. The values of X_i can thus be computed recursively as follows:

$$\begin{aligned}
X_{N-1} &= \frac{1}{1-(1-1/C)^{N-1}} \\
X_k &= 1 + X_k(1-1/C)^{N-1} + X_{k+1}\left[(1-1/C)^k - (1-1/C)^{N-1}\right]
\end{aligned}$$

and $\mathbb{E}\{T_N\} = X_1$.

Chapter Appendix 6.D — Multi-User Reception

Here we show that multi-user reception cannot overcome the $\sqrt{N}$ lower bound on delay for 2-hop routing. Specifically, we show that the delay $\mathbb{E}\left\{T_N\right\}$ for any algorithm which restricts packets to 2-hop paths satisfies:

$$\lim_{N\to\infty}\frac{\mathbb{E}\left\{T_N\right\}}{\sqrt{N}} \geq e^{-d^2}$$

Proof. Consider sending a single packet to its destination over an empty network. Let K_t represent the total number of users who have the packet at the beginning of slot t. Because scheduling restricts transfers to 2-hop paths, the number of users holding the packet increases every timeslot by at most the number of users in the same cell as the source (which is $d - 1/C$ on average). Hence, we have for all $t \geq 1$:

$$\mathbb{E}\left\{K_t\right\} \leq td \tag{6.27}$$

Note that during slots $\{1, 2, \ldots, t\}$ there are at most K_t users holding the packet, and hence during each of these slots the probability that no packet-holding user enters the cell of the destination is at least $(1 - \frac{1}{C})^{K_t}$. Thus:

$$\begin{aligned} Pr[T_N > t \mid K_t] &\geq \left(1 - \frac{1}{C}\right)^{tK_t} \\ &= \left(1 - \frac{d}{N}\right)^{tK_t} \end{aligned} \tag{6.28}$$

We thus have:

$$\begin{aligned} \mathbb{E}\left\{T_N\right\} &\geq tPr[T_N > t] \\ &= t\mathbb{E}_{K_t}\left\{Pr[T_N > t \mid K_t]\right\} \\ &\geq t\mathbb{E}_{K_t}\left\{\left(1 - \frac{d}{N}\right)^{tK_t}\right\} && (6.29) \\ &\geq t\left(1 - \frac{d}{N}\right)^{t\mathbb{E}\{K_t\}} && (6.30) \\ &\geq t\left(1 - \frac{d}{N}\right)^{t^2 d} && (6.31) \end{aligned}$$

where inequality (6.29) follows from (6.28), inequality (6.30) holds by Jensen's inequality (noticing that the function β^x is convex in x for any $\beta > 0$), and (6.31) follows from (6.27). This holds for all integers t. Choosing $t = \sqrt{N}$ yields $\mathbb{E}\{T_N\} \geq \sqrt{N}\left(1 - \frac{d}{N}\right)^{Nd} \rightarrow e^{-d^2}\sqrt{N}$. □

Chapter Appendix 6.E — Delay of $\sqrt{N}$ Redundancy Algorithm

Here we prove eq. (6.14), establishing an $O(\sqrt{N})$ bound on the service time $\mathbb{E}\{T_N\}$ for the partial feedback scheme with $\sqrt{N}$ redundancy. The proof requires the following preliminary lemma.

Lemma 20. *Consider N users which independently choose to enter one of C cells, and recall that $d = N/C$ represents the expected number of users per cell. Let J represent the number of users contained in a given cell. We have:*[7]

$$\mathbb{E}\{J\,|J \geq 1\} \leq 1 + d$$

Proof. Let I_i represent an indicator variable taking the value 1 if the ith user of the subset is in the cell, and 0 otherwise. Define K as the lowest indexed user within the cell, where we let $K = N+1$ if no users are present. Thus, $J = \sum_{i=K}^{N} I_i$. We have:

$$\begin{aligned}
\mathbb{E}\{J \mid J \geq 1\} &= 1 + \mathbb{E}_K\left\{ \sum_{i=K+1}^{N} \mathbb{E}\{I_i| K, J \geq 1\} \right| \left. J \geq 1 \vphantom{\sum_{i=K+1}^{N}}\right\} \\
&= 1 + \mathbb{E}_K\left\{ \sum_{i=K+1}^{N} \mathbb{E}\{I_i \mid K\} \right| \left. J \geq 1 \vphantom{\sum_{i=K+1}^{N}}\right\} \qquad (6.32) \\
&\leq 1 + \mathbb{E}_K\left\{ \sum_{i=1}^{N} \mathbb{E}\{I_i \mid K\} \right| \left. J \geq 1 \vphantom{\sum_{i=1}^{N}}\right\} \\
&= = 1 + \mathbb{E}_K\left\{ \sum_{i=1}^{N} \frac{1}{C} \right| \left. J \geq 1 \vphantom{\sum_{i=1}^{N}}\right\} = 1 + d \qquad (6.33)
\end{aligned}$$

where (6.32) follows because the condition $J \geq 1$ can be inferred by knowledge of K, and (6.33) follows because $\mathbb{E}\{I_i \mid K\} = 1/C$ for all users i. □

To prove the $\sqrt{N}$ bound on $\mathbb{E}\{T_N\}$, recall that $T_N = S_1 + S_2$, where S_1 represents the time required for the source to send out $\sqrt{N}$ replicas of the packet (while competing with other sessions for network resources), and S_2 represents the time required to reach the destination given that $\sqrt{N}$ users have the packet.

[7] An exact value of $\mathbb{E}\{J \mid J \geq 1\} = \mathbb{E}\{J\}/Pr[J \geq 1]$ can easily be computed and leads to tighter but more complicated delay bounds.

Lemma 21.

$$\mathbb{E}\left\{S_1\right\}, \mathbb{E}\left\{S_2\right\} \leq \frac{4+2d}{\gamma_N(1-e^{-d})}\sqrt{N}$$

where γ_N is a sequence that converges to 1 *as $N \rightarrow \infty$.*

Proof. The $\mathbb{E}\left\{S_1\right\}$ bound: Let S_1 represent the time required for the source to deliver a duplicate packet to $\sqrt{N}$ distinct users. For the duration of S_1, there are at least $N - \sqrt{N}$ users who do not have the packet, and hence every timeslot the probability that at least one of these users visits the cell of the source is at least $1-(1-\frac{1}{C})^{N-\sqrt{N}}$. Given this event, the probability that the source is chosen by the partial feedback algorithm to transmit is expressed by the product $\alpha_1\alpha_2$, representing probabilities for the following conditionally independent events: α_1 is the probability that the source is selected from all other users in the cell to be the transmitting user, and α_2 represents the probability that this source is chosen to operate in 'source-to-relay' mode. Let random variable J represent the number of additional users in the cell of the source (excluding the source user itself). The value of α_1 is thus $\alpha_1 = \mathbb{E}\left\{1/(J+1) \mid J \geq 1\right\}$. By Jensen's inequality, we have:

$$\begin{aligned} \alpha_1 &\geq 1/\mathbb{E}\left\{1+J \mid J \geq 1\right\} \\ &\geq 1/(2+d) \end{aligned}$$

where the last inequality follows because $\mathbb{E}\left\{J \mid J \geq 1\right\} \leq 1+d$ (as proven in Lemma 20).

The probability α_2 that the source operates in 'source-to-relay' mode is $1/2$. Thus, every timeslot during the interval S_1, the source delivers a replica packet to a new user with probability of at least ϕ, where

$$\begin{aligned} \phi \geq \left(1-(1-\frac{1}{C})^{N-\sqrt{N}}\right)\frac{1}{2(2+d)} \\ \rightarrow \frac{1-e^{-d}}{4+2d} \end{aligned}$$

The average time until a replica is transmitted to a new user is thus a geometric variable with mean less than or equal to $1/\phi$. It is possible that two or more replicas are delivered in a single timeslot. However, in the worst case, $\sqrt{N}$ of these times are required, so that the average time $\mathbb{E}\left\{S_1\right\}$ is upper bounded by $\sqrt{N}/\phi$. □

Proof. The $\mathbb{E}\{S_2\}$ *bound:* To prove the bound on $\mathbb{E}\{S_2\}$, note that every timeslot in which there are at least $\sqrt{N}$ users with replicas of the packet, the probability that one of these users transmits the packet to the destination is given by the chain of probabilities $\theta_0\theta_1\theta_2\theta_3$. The θ_i values represent probabilities for the following conditionally independent events: θ_0 represents the probability that there is at least one other user in the same cell as the destination (and is given by $\theta_0 = 1-(1-1/C)^{N-1} \to 1-e^{-d}$), θ_1 represents the probability that the destination is selected as the receiver (where, similar to the α_1 computation, we have $\theta_1 \geq 1/(2+d)$), θ_2 represents the probability that the sender operates in 'relay-to-destination' mode (where $\theta_2 = 1/2$), and θ_3 represents the probability that the sender is one of the $\sqrt{N}$ users who have a replica of the packet intended for the destination (where $\theta_3 = \sqrt{N}/(N-1) \geq 1/\sqrt{N}$). Thus, every timeslot, the probability that the S_2 time comes to completion is at least $\frac{(1-e^{-d})}{(4+2d)\sqrt{N}}$. The value of $\mathbb{E}\{S_2\}$ is thus less than or equal to the inverse of this quantity. □

Chapter Appendix 6.F — Logarithmic Delay for Flooding Protocol

Here we prove Lemma 19: Under the algorithm of flooding the network with a single packet, for any network size $N \geq \max\{d, 2\}$, the expected time $\mathbb{E}\left\{T_N\right\}$ for the packet to reach every user satisfies $\mathbb{E}\left\{T_N\right\} \leq \mathbb{E}\left\{S_1\right\} + \mathbb{E}\left\{S_2\right\}$, where:

$$\mathbb{E}\left\{S_1\right\} \leq \frac{\log(N)\,(1+d/2)}{\log(2)(1-e^{-d/2})}$$
$$\mathbb{E}\left\{S_2\right\} \leq 1 + \frac{2}{d}(1+\log(N/2))$$

Proof. (The $\mathbb{E}\left\{S_2\right\}$ *Bound)* Let M represent the number of users who do *not* initially have the packet (so that $M \leq N/2$), and label these M users $\{u_1, u_2, \ldots, u_M\}$. Let X_i represent the number of timeslots it takes for the non-packet holding user u_i to reach a cell containing a user who possesses a packet. Because of the multi-user reception feature, user u_i must receive the packet at this time. The random variable X_i is geometric, in that a 'success' happens on any given timeslot with probability $\psi \geq 1 - (1-\frac{1}{C})^{N/2}$. Thus, we have for all N:

$$\psi \geq 1 - e^{-d/2} \tag{6.34}$$

All times X_i are independent and identically distributed, and hence the random variable S_2 is equal to the maximum value of at most $M = \lfloor N/2 \rfloor$ i.i.d. variables. Hence, $\mathbb{E}\left\{S_2\right\} \leq \mathbb{E}\left\{\max\{X_1, X_2, \ldots, X_M\}\right\}$. To obtain a simple bound on this time, we consider new random variables $\{Y_1, Y_2, \ldots, Y_M\}$ which are i.i.d. and *exponentially distributed* with rate $\lambda = \log(1/(1-\psi))$. Notice that the random variable $1 + Y_i$ is *stochastically greater* than X_i, because the complementary distribution functions satisfy $Pr[1+Y_i > t] \geq Pr[X_i > t]$ for all real numbers t (see [119] for a treatment of stochastic dominance for random variables). It follows that:

$$\begin{aligned}\mathbb{E}\left\{S_2\right\} &\leq \mathbb{E}\left\{\max\{X_1, X_2, \ldots, X_M\}\right\} \\ &\leq 1 + \mathbb{E}\left\{\max\{Y_1, Y_2, \ldots, Y_M\}\right\}\end{aligned}$$

The expected maximum of M i.i.d. exponential variables of rate λ is equal to the expectation of the sum of intervals $I_1 + I_2 + \ldots + I_M$, where I_i represents the duration of time between the $(i-1)^{th}$ and i^{th} completion time. The interval I_1 is the first completion time of M independently racing exponential variables, and hence I_1 is exponentially distributed with rate $M\lambda$. Furthermore, I_2 is the first completion time of $M-1$ racing exponential variables, I_3 is the first completion time of $M-2$ racing exponentials, and so on. It follows that:

$$\mathbb{E}\left\{I_1 + I_2 + \ldots + I_M\right\} = \frac{1}{\lambda}\sum_{m=1}^{M}\frac{1}{m}$$

Hence, $\mathbb{E}\left\{S_2\right\} \leq 1 + \frac{1}{\lambda}\sum_{m=1}^{M}\frac{1}{m}$, which is upper bounded by $1 + \frac{1}{\lambda}(1+\log(M))$. Hence:

$$\mathbb{E}\left\{S_2\right\} \leq 1 + \frac{1+\log(M)}{\log(1/(1-\psi))} \leq 1 + \frac{1+\log(N/2)}{\log(e^{d/2})}$$

□

Proof. *(The* $\mathbb{E}\left\{S_1\right\}$ *bound)* We compute a bound on $\mathbb{E}\left\{S_1\right\}$ by noting that $\mathbb{E}\left\{S_1\right\} \leq \mathbb{E}\left\{\tilde{S}_1\right\}$, where $\tilde{S}_1$ is the time to reach at least $N/2$ users when the multi-user reception feature is turned off, and any transmitted packet is received by at most 1 other user within a cell. It turns out that the variable $\tilde{S}_1$ is easier to work with, as the number of users holding the packet can at most double every timeslot. Let K_t represent the number of users containing a duplicate version of the packet at timeslot $t \in 1, 2, \ldots$ (suppose only the source user has the packet at time 0, so that $K_0 = 1$). Let $u_1, u_2, \ldots, u_{K_t}$ represent the users containing the packet at time t. Each of these users u_i delivers the packet to a_i new users on the next timeslot, where a_i is a binary random variable taking a value of either 0 or 1. Whenever there are at least $N/2$ users which do not currently hold the packet, we have that $\mathbb{E}\left\{a_i\right\} \geq \theta_1\theta_2$, where $\theta_1 = 1 - (1-\frac{1}{C})^{N/2}$ represents a lower bound on the probability that at least one of the new users enters the cell of user u_i, and θ_2 represents a lower bound on the probability that user i is selected to transmit *its* replica among all other packet-holding users within the cell. Define J as the total number of other packet-holding users in the cell (not including user i). It follows that:

$$\begin{aligned} \theta_2 &= \mathbb{E}\left\{\frac{1}{1+J}\right\} & (6.35)\\ &\geq \frac{1}{1+\mathbb{E}\left\{J\right\}} & (6.36)\\ &\geq \frac{1}{1+d/2} & (6.37) \end{aligned}$$

where (6.36) follows by Jensen's inequality and convexity of the function $1/(1+x)$, and (6.37) follows because there are no more than $N/2$ packet holding users, and hence $\mathbb{E}\left\{J\right\} \leq \frac{N}{2C} = d/2$. Thus:

$$\begin{aligned} \mathbb{E}\left\{a_i\right\} &\geq \frac{1-(1-\frac{1}{C})^{N/2}}{1+d/2} \\ &\geq \frac{1-e^{-d/2}}{1+d/2} \qquad (6.38) \end{aligned}$$

where (6.38) follows because $(1-\frac{d}{N})^N \leq e^{-d}$ for all $N \geq d > 0$.

Let $Z_t = K_t/K_{t-1}$ be a random variable representing the multiplicative factor by which the number of packet-holding users grows after one timeslot. (Note that $1 \leq Z_t \leq 2$). It clearly holds that:

$$Z_{t+1} = \frac{K_t + a_1 + a_2 + \ldots + a_{K_t}}{K_t}$$

The a_i random variables are not independent, although they are identical. Thus, for any timeslot t in which fewer than $N/2$ users have packets:

$$\begin{aligned} \mathbb{E}\left\{Z_{t+1} \mid K_t\right\} &= \frac{K_t + K_t\mathbb{E}\left\{a_1\right\}}{K_t} \\ &= 1+\mathbb{E}\left\{a_1\right\} \\ &\geq 1+\frac{1-e^{-d/2}}{1+d/2} \qquad (6.39) \end{aligned}$$

Now consider the stopping time $\tilde{S}_1$ where at $t = \tilde{S}_1 - 1$ there are fewer than $N/2$ users with packets, but at time $t = \tilde{S}_1$ the $N/2$ threshold is either met or crossed. Note that $\tilde{S}_1$ is similar to a *stopping time variable*, treated in [49], [119], although the event $\{\tilde{S}_1 \geq t\}$ is not independent of Z_t. The number of users $K_{\tilde{S}_1}$ containing the packet at time $t = \tilde{S}_1$ satisfies:

$$N \geq K_{\tilde{S}_1} = Z_1 Z_2 \ldots Z_{\tilde{S}_1}$$

and hence

$$\log(N) \geq \log(Z_1) + \log(Z_2) + \ldots + \log(Z_{\tilde{S}_1})$$

Define the indicator random variable I_t to be 1 if $\tilde{S}_1 \geq t$, and 0 otherwise. Taking expectations of the above inequality, we find:

$$\begin{aligned}
\log(N) &\geq \mathbb{E}\left\{\sum_{t=1}^{\tilde{S}_1}\log(Z_t)\right\} \\
&= \mathbb{E}\left\{\sum_{t=1}^{\infty}\log(Z_t)I_t\right\} \\
&= \mathbb{E}\left\{\sum_{t=1}^{\infty}\mathbb{E}\left\{\log(Z_t)I_t \mid K_{t-1}\right\}\right\} \\
&= \mathbb{E}\left\{\sum_{t=1}^{\infty}I_t\mathbb{E}\left\{\log(Z_t) \mid K_{t-1}\right\}\right\}
\end{aligned}$$

where the last inequality follows because the variable K_{t-1} completely determines the binary value of I_t. Recall that $1 \leq Z_t \leq 2$, and hence $\log(Z_t) \geq \log(2)(Z_t - 1)$ (as the lower bound values are points along the chord of the concave function $\log(Z)$ over the interval $1 \leq Z \leq 2$). We thus have:

$$\begin{aligned}
\frac{\log(N)}{\log(2)} &\geq \mathbb{E}\left\{\sum_{t=1}^{\infty}I_t\mathbb{E}\left\{(Z_t - 1)\,|K_{t-1}\right\}\right\} \\
&\geq \left(\frac{1-e^{-d/2}}{1+d/2}\right)\mathbb{E}\left\{\sum_{t=1}^{\infty}I_t\right\} \qquad (6.40) \\
&= \left(\frac{1-e^{-d/2}}{1+d/2}\right)\mathbb{E}\left\{\tilde{S}_1\right\}
\end{aligned}$$

where (6.40) follows from (6.39). Thus, $\mathbb{E}\{S_1\} \leq \mathbb{E}\left\{\tilde{S}_1\right\} \leq \frac{\log(N)(1+d/2)}{\log(2)(1-e^{-d/2})}$. □

Chapter Appendix 6.G — Minimum Delay for Multi-Hop Routing is Logarithmic

Lemma 22. *Starting with a single packet contained in one user in an empty network of size N, the flooding algorithm of delivering the packet to its destination by having every duplicate-carrying user transmit to other users whenever possible has an average delay $\mathbb{E}\{T_N\}$ which is logarithmic. In particular*

$$\lim_{N\to\infty}\frac{\mathbb{E}\{T_N\}}{\log(N)} \geq \frac{1}{\log(1+d)}$$

This bound holds even if multi-user reception is available.

Proof. As in the proof of Lemma 19, define K_t as the number of users holding the packet at time t (where $K_0 = 1$), and let $Z_t = K_t/K_{t-1}$ represent the growth factor after one timeslot. We have:

$$Z_{t+1} = \frac{K_t + a_1 + a_2 + \ldots + a_{K_t}}{K_t}$$

where a_i represents the number of new users to which the ith packet-holding user transmits during a timeslot. We clearly have $\mathbb{E}\{a_i\} \leq d$ during any timeslot, and hence:

$$\mathbb{E}\{Z_{t+1} \mid K_t\} = \frac{K_t + K_t\mathbb{E}\{a_1\}}{K_t} \leq 1+d$$

Because $K_t = Z_1 Z_2 \cdots Z_t$, it follows by recursion that:

$$\mathbb{E}\{K_t\} \leq (1+d)^t \tag{6.41}$$

Note that during slots $\{1, 2, \ldots, t\}$ there are at most K_t users holding the packet, so the probability that none of these users enters the cell of the destination on such a timeslot is greater than or equal to $\left(1-\frac{1}{C}\right)^{K_t}$. Hence, the proof given in Chapter Appendix 6.D for the $\sqrt{N}$ bound for 2-hop routing can be followed exactly up to (6.30). In particular, we have [compare with (6.28)-(6.30)]:

$$\begin{aligned}
\mathbb{E}\{T_N\} &\geq tPr[T_N > t] \\
&= t\mathbb{E}_{K_t}\{Pr[T_N > t \mid K_t]\} \\
&\geq t\mathbb{E}_{K_t}\left\{\left(1-\frac{d}{N}\right)^{tK_t}\right\} \\
&\geq t\left(1-\frac{d}{N}\right)^{t\mathbb{E}\{K_t\}}
\end{aligned}$$

Using (6.41) in the above inequality, we have:

$$\mathbb{E}\{T_N\} \geq t\left(1-\frac{d}{N}\right)^{t(1+d)^t} \tag{6.42}$$

The above inequality holds for all integers $t \geq 0$. For convenience, we choose t to represent a base $(1+d)$ logarithm: $t \triangleq \log_{1+d}(\alpha N^\beta)$, where β is any number less than 1, and α is chosen within the bounds $1 \leq \alpha \leq (d+1)$ so that t is an integer. Using this value of t in (6.42), we have:

$$\mathbb{E}\{T_N\} \geq \frac{(\log(\alpha)+\beta\log(N))}{\log(1+d)}\left[\left(1-\frac{d}{N}\right)^N\right]^{\frac{\alpha N^\beta \log(\alpha N^\beta)}{N\log(1+d)}}$$

Note that $\left(1-\frac{d}{N}\right)^N \rightarrow e^{-d}$ as $N \rightarrow \infty$, and its exponent $\frac{\alpha N^\beta \log(\alpha N^\beta)}{N\log(1+d)}$ converges to 0 whenever $\beta < 1$. It follows that $\left[\left(1-\frac{d}{N}\right)^N\right]^{\frac{\alpha N^\beta \log(\alpha N^\beta)}{N\log(1+d)}} \rightarrow 1$, and hence:

$$\lim_{N\rightarrow\infty} \frac{\mathbb{E}\{T_N\}}{\log(N)} \geq \frac{\beta}{\log(1+d)}$$

for any $\beta < 1$. The bound can be optimized by taking a limit as $\beta \rightarrow 1$, yielding the result. □

Chapter 7

Conclusions

We have developed dynamic control algorithms for networks with time varying channel conditions, random inputs, and adaptive transmission rates. A general network model was constructed in terms of arbitrary rate-power curves. This model allows for a simple separation of network layer and physical layer concepts while enabling the network controller to take full advantage of the physical properties of each data link. A wide variety of networks can be treated according to this framework, including satellite networks with optical crosslinks and RF downlinks, wireless ad-hoc networks, computer networks, and hybrid networks with both wireless and wireline components.

Special attention was given to satellite networks, and it was shown that dynamic power allocation can significantly improve the throughput and delay performance of such systems. A separation principle was developed, demonstrating that the crosslink and downlink layers of satellite networks can in principle be optimized individually while maintaining global optimality.

We have also considered ad-hoc wireless networks with mobility. Distributed control algorithms for these systems were developed to achieve high throughput and low delay, as established both analytically and through simulations. Fundamental rate-delay tradeoffs were explored.

Our approach to data networking considers the full effects of queueing. To meet this purpose, a variety of queueing theoretic tools were constructed, contributing to a theory of queueing analysis for time varying systems. These tools were used both in the analysis of network performance as well as in the design of our network controllers. Indeed, knowledge of the queueing states of the system was used to design robust network controllers which do not require knowledge of channel statistics or traffic rates from other users. These controllers can be implemented in a decentralized fashion whenever channels are independent.

Our discussion of real time implementations in Chapter 3 and the performance gap between centralized and distributed network control in Chapter 4 touches on another dimension of networks research, that of the fundamental tradeoffs between performance and implementation complexity (see [109] [113] for a more direct treatment of the subject). This research builds upon an emerging theory of data networks, with the goal of describing the capacity and delay limits in networks with constrained resources and abilities, as well as specifying the control algorithms which achieve these limits.

Appendix A

Convexity in Queueing Systems

In this chapter we examine a work conserving $*/*/1$ queue and develop fundamental monotonicity and convexity properties of unfinished work and packet waiting time in the queue as a function of the packet arrival rate λ. The "*/*" notation refers to the fact that the input process has arbitrarily distributed and correlated interarrival times and packet lengths. (This differs from the standard GI/GI description, where interarrival times and packet lengths are independent and identically distributed).[1] The arrival process consists of the superposition of two component streams: an arbitrary and uncontrollable background input of the $*/*$ type, and a rate-controllable packet stream input (Fig. A-1). The rate-controllable stream contains a collection of indistinguishable $*/*$ substreams, and its rate is varied in discrete steps by adding or removing these substreams as inputs to the queue. We show that any moment of unfinished work is a convex function of this input rate. Under the special case of FIFO service, we show that waiting time moments are also convex.

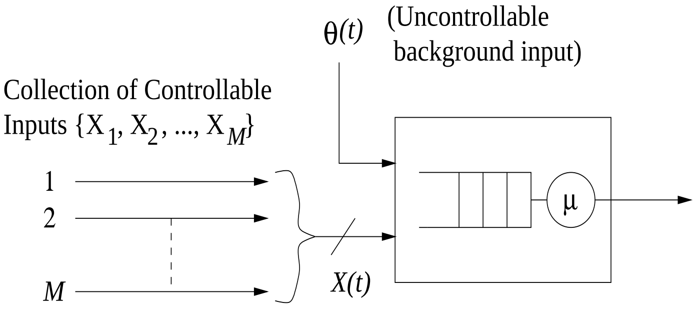

θ(t)
(Uncontrollable background input)
Collection of Controllable Inputs {X1, X2, ..., XM}
1
2
M
X(t)
μ

Figure A-1: A work conserving queue with server linespeed μ, a $*/*$ background input $\theta(t)$, and rate-controllable $*/*$ inputs $X(t) = \{X_1(t), \ldots, X_M(t)\}$.

[1] We avoid the ambiguous notation $G/G/1$, as different texts use this to mean either $*/*/1$ or $GI/GI/1$.

We then extend the convexity result to address the problem of optimally routing input streams over a parallel collection of N queues with different linespeeds $(\mu_1, ..., \mu_N)$. We show that cost functions consisting of convex combinations of unfinished work moments in each of the queues are convex in the N-dimensional rate tuple $(\lambda_1, ..., \lambda_N)$. In the symmetric case where the N queues are weighted equally in the cost function and have identical background processes, this convexity result implies that the uniform rate allocation minimizes cost. In the case of an asymmetric collection of N parallel queues, we present a sequentially greedy routing algorithm that is optimal.

The convexity results and optimization methods are extended to treat queues with time-varying linespeeds $(\mu_1(t), ..., \mu_N(t))$. We show that the amount of unprocessed bits in the multi-queue system remains convex in the input rate vector $(\lambda_1, ..., \lambda_N)$. However, we demonstrate that waiting times are not necessarily convex for general time varying linespeed problems. For simplicity of exposition, we postpone the time-varying analysis until Section A.6.

Convexity properties of single and parallel collections of queues have been developed previously with respect to various parameters and with various assumptions about the nature of the input processes and service time processes. In [82], queues with a large number of i.i.d. input streams are analyzed in the large deviations regime, and the buffer requirement to meet a specified overflow rate is shown to be convex in the server rate μ. The result is extended in [83] to address more general input processes. In [59] a convexity theory of "multi-modular functions" is developed and used to establish an optimal admission control sequence for arrivals to an exponential server, given that a fixed ratio of packets must be accepted. Extensions to queues and tandems with fixed batch arrivals are treated in [4] [2].

In [112] [101], the authors analyze the expected packet occupancy in tree networks of deterministic service time queues. It is shown that expected occupancy of any interior queue of the tree is a concave function of the multiple exogenous input rates, while occupancy in queues on the edge of the network is shown to be convex. Convexity properties of parallel $GI/GI/1$ queues with packet-based probabilistic "Bernoulli" routing are developed in [56] [25] [66], where it is shown under various models and independence assumptions that backlog moments in each queue are convex functions of the Bernoulli splitting probability, and hence uniform Bernoulli splitting minimizes expected backlog in homogeneous systems among the class of all Bernoulli splittings. A related result for homogeneous systems in [80] shows that uniform splitting is optimal for *arbitrary arrivals* in a system of parallel queues with equal exponential servers (i.e., $*/M$ inputs), and stochastically minimizes the total workload at every instant of time.

Our treatment of the convexity problem for streams of $*/*$ inputs is an important feature, since packets from a single source often must be routed together to maintain predictability and to prevent out-of-order delivery. Such situations occur, for example, when we have N streams carrying voice traffic from individual users. In this case, convexity properties are considered with respect to the integer number of streams routed to a queue.[2] However, we also treat the packet-based routing method of [56] [25] [66] [80] in a more general (yet simpler) context. Rather than emphasizing the differences between packet-based and stream-based routing, we discover a fundamental similarity. We consider packet-based routing of a general $*/*$ input stream whose rate can be split according to a continuous rate parameter, using a splitting method such as the probabilistic "Bernoulli" splitting in [56] [25] [66] [80]. We find that convexity for this general packet-based routing problem is a consequence of our stream-based results.

Our analysis is carried out by introducing a new function of the superposition of two input streams that we call the blocking function. Properties of the blocking function are developed by examining sample paths of work conserving queues, and each property corresponds to an intuitive comparison of two different queueing system configurations. These properties are then used to establish the convexity and optimal routing results. As a special case of this analysis applied to the problem of routing over a homogeneous set of queues, we obtain the uniform splitting optimality result of [56] [25] [66] under the more general context of $*/*$ inputs (as well as time varying server rates). The analytical techniques used to prove the results in this chapter are novel and can likely be used in other contexts.

In the next section, we review related work in the area of convexity and monotonicity in queueing systems. In Section A.2 we define the blocking function. In Section A.3 we establish convexity properties of unfinished work and waiting time with respect to a discrete rate parameter corresponding to adding an integer number of indistinguishable $*/*$ streams as inputs to a queue. Convexity properties in terms of a continuous rate splitting parameter

[2] This problem is related to general admission control problems, where convexity properties are essential to establishing the structural properties of an optimal policy, see [128].

are developed in Section A.4, and in Section A.5 we consider example applications to the problem of optimal routing over a parallel set of queues. Time varying server rates are treated in Section A.6.

A.1 Related Work on Monotonicity and Convexity in Queueing Systems

It is often possible to use stochastic inequalities to compare the relative performance of two different queueing systems, even in cases where the exact performance of both systems is unknown or has no closed form analytical expression. For example, Stoyan develops a simple monotonicity result in [129], where it is shown that if a Markov process $X_1(t)$ has a step-by-step transition probability distribution that is *stochastically larger*[3] than the corresponding transition distribution for another process $X_2(t)$, and if the initial value $X_1(0)$ for the first process is greater than or equal to the initial value of the second, then at *any particular instant of time* t^* the random variable $X_1(t^*)$ is stochastically greater than or equal to the random variable $X_2(t^*)$. For queueing systems, such Markov processes correspond to queues with Poisson inputs and independent service times (so called "$M/GI/1$" queues), and the monotonicity theorem establishes the classical result that if two queues with identical exponential servers and service rates λ_1 and λ_2 are compared, then *at every instant of time* the number of packets in the first queue is stochastically less than the number in the second whenever $\lambda_1 \leq \lambda_2$ and the initial backlog of the first queue is less than or equal to that of the second.

Ross develops a similar monotonicity result in [119] by using the $\max(a-b, 0)$ operator to propogate waiting time distributions in a $GI/GI/1$ queue that is initially empty. It is shown that if two queues with the same arrival and service rates are compared, the waiting times in the first queue will be stochastically less than the waiting times in the second whenever the interarrival times and service times of the first queue are stochastically less bursty than the second. Baccelli and Brémaud present similar monotonicity results with respect to the unfinished work in a $GI/GI/1$ queue [see Ex. 4.2.6, pp.283 in [7]], and to the waiting time process in a queue where successive arrival and service time vectors (α_n, σ_n)

[3] A random variable X is said to be *stochastically greater* than a random variable Y if $Pr[X > z] \geq Pr[Y > z]$ for all z. See [119] for an illuminating treatment of the subject of stochastic dominance and convex ordering relations.

are independent but perhaps differently distributed for all n [see Ex. 4.4.7, pp.306 in [7]]. Similar statements for systems driven by doubly-stochastic Poisson processes are given in [26]. These results demonstrate that *regular* (less bursty) sequences are better in terms of reducing queue congestion. Indeed, in [64] it is shown that any moment of waiting time in a general $*/*/1$ queue, that is, a queue *general ergodic arrivals and packet lengths*, is greater than the corresponding moment if the arrival process is replaced by a periodic stream of the same rate, or if the packet lengths are replaced by fixed lengths equal to the sequence average.

A related result by Hajek in [59] establishes the optimal packet acceptance rule for an arrival stream (with independent interarrival times) to a queue with an exponential server, in the case when a fixed ratio of packets must be accepted. The result is generalized to non-exponential servers in [4] [2].

The effect of regularizing traffic is used in [45] to show that round robin routing is the optimal "backlog-blind" strategy for routing an *arbitrary arrival sequence* to two parallel queues with homogeneous exponential servers (the authors also show that *Join the Shortest Queue* routing is optimal for backlog-aware strategies). Similar round robin optimality results are shown in [90] for routing to a parallel set of homogeneous queues with i.i.d. service times that have an "increasing failure rate" property, and in [89] for queues with random and homogeneous server outages. Likewise, round robin is shown to be optimal in [3] for routing to parallel homogeneous tandems under a service time independence assumption—namely, that the service times of a packet routed to tandem 1 are independent of the service times it would experience if routed to tandem 2. We note that round robin is not optimal for general input processes with potentially correlated packet lengths. Indeed, consider a Poisson stream arriving to a system of two queues with identical server rates. If packet lengths alternate between short packets and long packets, then round robin is among the worst of all possible routing strategies. Similar counterexamples can be constructed for stationary service time processes where successive packet lengths are negatively correlated.

Discussions of various routing policies, including backlog-unaware probabilistic routing and backlog-aware Join-the-Shortest-Queue routing are provided in [17]. Shortest queue results similar to [45] are developed in [145] [146] for multiple homogeneous queues, in [135] for homogeneous queues with finite buffers, and in [110] for heterogeneous queues with finite buffers. Load balancing techniques for various symmetric systems with memoryless properties of either service times, interarrival times, or both are considered in [145] [25] [80] [56]. Approximation techniques are developed in [32] for heterogeneous systems, and approaches to optimal routing using convex optimization are described in [14] [21] [130] [19] [22].

The prior works that are most directly related this chapter are the convexity results given in [25] [56] [66] for queues with independence assumptions, and the uniform Bernoulli routing result of [80] developed for general arrival streams to homogeneous exponential servers. These results are based on the theory of majorization and Schur-convex functions, described, for example, in [123] [24]. Our approach is quite different and enables general analysis of both stream based routing as well as a large class of probabilistic splitting methods (including Bernoulli splitting), and does not require independence assumptions on the input traffic. Furthermore, our analysis is self-contained and is based on easily-understood properties of queueing systems.

A.2 The Blocking Function For $*/*/1$ Queues

Consider a work conserving queue with a single server that can process packets at a constant line speed of μ bits/second (Fig. A-2). Variable length packets from input stream X flow into the queue and are processed at the single server according to any work-conserving service discipline (e.g., FIFO, LIFO, Shortest Packet First, GPS, etc.). The input stream is characterized according to two random processes: (i) The sequence $\{a_k\}$ of inter-arrival times, and (ii) The sequence $\{l_k\}$ of packet lengths.

We assume the processes $\{a_k\}$ and $\{l_k\}$ are ergodic with arrival rate λ and average packet length $\mathbb{E}\{L\}$, respectively. In general, inter-arrival times may be correlated with each other as well as jointly correlated with the packet length process. We maintain this generality by describing the input to the queue by the single random process $X(t)$, which represents the amount of bits brought into the queue as a function of time. As shown in Fig. A-2, a particular input $X(t)$ is a non-decreasing staircase function. Jumps in the $X(t)$ function occur at packet arrival epochs, and the amount of increase at these times is equal to the length of the entering packet.

For a given queue with input process $X(t)$, we define the *unfinished work process* $U_X(t)$ as the total amount of unprocessed bits in the queueing system (buffer plus server) as

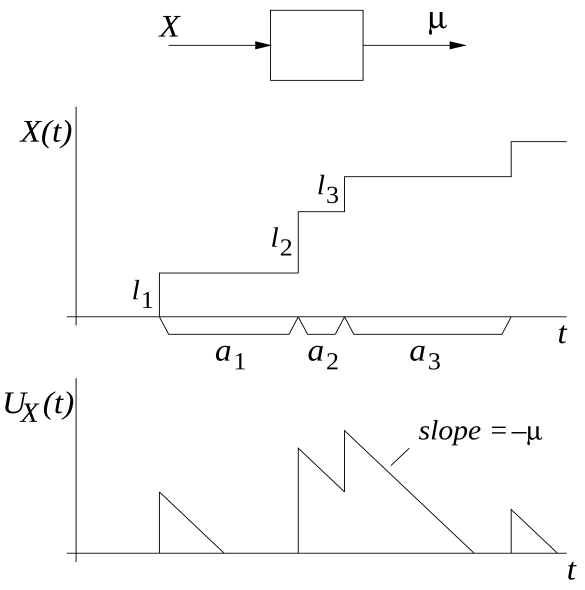


Figure A-2: A work conserving $*/*/1$ queue, and typical sample paths of accumulated and unfinished work.

a function of time. Note that for a system with a processor of rate μ and an amount of unfinished work $U_X(t)$, the quantity $U_X(t)/\mu$ represents the amount of time required for the system to empty if no other packets were to arrive. We assume the queue is initially empty at time $t = 0$. It is clear that $U_X(t)$ is the same for all work conserving service disciplines. It is completely determined by $X(t)$ as well as the server linespeed μ. An example unfinished work function $U_X(t)$ is shown in Fig. A-2. Notice the triangular structure and the fact that each new triangle emerges at packet arrival times and has a downward slope of $-\mu$.

We define the *superposition* of two input streams $X_1(t)$, $X_2(t)$ as the sum process $X_1(t)+X_2(t)$. We make the following sample path observation, which holds for any arbitrary set of sample paths:

Observation 1: For all times t, we have:

$$U_{X_1+X_2}(t) \geq U_{X_1}(t) + U_{X_2}(t) \tag{A.1}$$

Thus, for any two inputs X_1 and X_2, the amount of unfinished work in a work conserving queueing system with the superposition process $X_1 + X_2$ is always greater than or equal to the sum of the work in two identical queues with these same processes X_1 and X_2 entering them individually. This is illustrated in Fig. A-3.

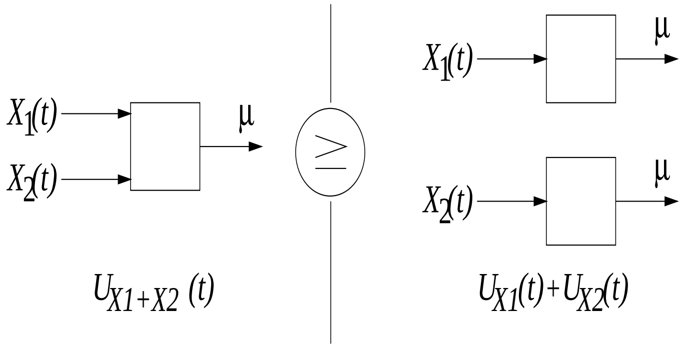


Figure A-3: A queueing illustration of the non-negativity property of the blocking function.

Proof. (Observation 1) We compare the two system configurations of Fig. A-3. Since $U_{X_1+X_2}(t)$ is the same for all work conserving service disciplines, we can imagine that packets from the X_1 stream have preemptive priority over X_2 packets. The queueing dynamics of the X_1 packets are therefore unaffected by any low priority packets from the X_2 stream. Thus, the $U_{X_1+X_2}(t)$ function can be written as $U_{X_1}(t)$ plus an extra amount $extra\_X_2(t)$ due to the X_2 packets, as shown in Fig. A-4. This extra amount (represented as the striped region in Fig. A-4) can be thought of as the amount of unfinished work remaining in a queue with the X_2 input stream alone, where the server goes on idle "vacations" exactly at times when $U_{X_1}(t)$ is nonzero. Clearly, this unfinished work is greater than or equal to the unfinished work there would be if the server did not go on vacations—which is $U_{X_2}(t)$. Thus:

$$U_{X_1+X_2}(t) = U_{X_1}(t) + extra\_X_2(t) \geq U_{X_1}(t) + U_{X_2}(t)$$

□

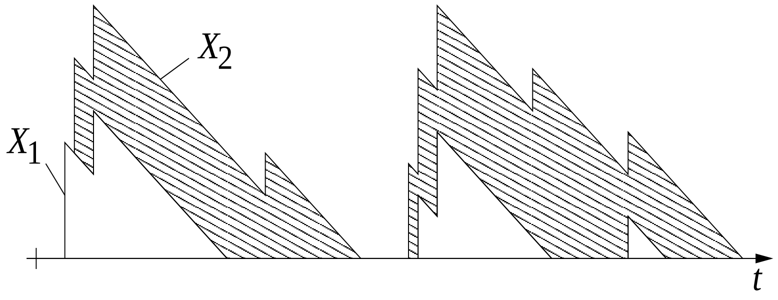


Figure A-4: An example sample path of the unfinished work function $U_{X_1+X_2}(t)$ in a system where X_1 packets have preemptive priority.

This observation places a lower bound on the unfinished work $U_{X_1+X_2}(t)$ in terms of its component streams. We note that Baccelli et. al. give an *upper* bound in terms of different component streams [see prop. 2.11.6, pp. 164 in [7]]. Specifically, in [7] it is shown that the unfinished work at the time of the n^{th} arrival is less than or equal to the unfinished work at the time of the k^{th} arrival plus the unfinished work at the time of the n^{th} arrival in a system where the first $n-k$ original arrivals are removed, for any $k \in \{1, \ldots, n\}$.

The simple observation (A.1) motivates the following definition:

Definition 7. *The* Blocking Function $\beta_{X_1,X_2}(t)$ *between two streams* X_1 *and* X_2 *is the function:*

$$\beta_{X_1,X_2}(t) \triangleq U_{X_1+X_2}(t) - U_{X_1}(t) - U_{X_2}(t) \tag{A.2}$$

Thus, the blocking function is a random process that represents the extra amount of unfinished work in the system due to the blocking incurred by packets from the X_1 stream mixing with the X_2 stream.

Lemma 23. *The blocking function has the following properties for all times* t*:*

$$\begin{array}{ll} \beta_{X_1,X_2}(t) \geq 0 & \textit{(Non-negativity)} \\ \beta_{X_1,X_2}(t) = \beta_{X_2,X_1}(t) & \textit{(Symmetry)} \\ \beta_{X_1+X_2,X_3}(t) \geq \beta_{X_1,X_3}(t) & \textit{(Monotonicity)} \end{array}$$

The non-negativity lemma above is just a re-statement of (A.1), while the symmetry property is obvious from the blocking function definition. Below we prove the monotonicity property.

Proof. (Monotonicity) From the definition of the blocking function in (A.2), we find that the monotonicity statement is equivalent to the following inequality at every time t:

$$U_{X_1+X_2+X_3}(t) - U_{X_1+X_2}(t) - U_{X_3}(t) \geq U_{X_1+X_3}(t) - U_{X_1}(t) - U_{X_3}(t)$$

Cancelling and shifting terms, it follows that we must prove:

$$U_{X_1+X_2+X_3}(t) + U_{X_1}(t) \geq U_{X_1+X_2}(t) + U_{X_1+X_3}(t) \tag{A.3}$$

We have illustrated (A.3) in Fig. A-5. We thus prove that the sum of the unfinished work in Systems A and B of Fig. A-5 is greater than or equal to the sum in A' and B'.

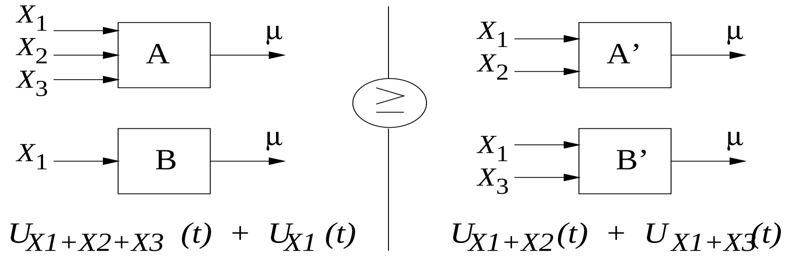


Figure A-5: A queueing illustration of the monotonicity property of the blocking function.

In a manner similar to the proof of Observation 1, we give packets from both the X_1 and X_2 streams preemptive priority over X_3 packets. The queues of Fig. A-5 can thus be treated as having servers that take "vacations" from serving X_3 packets during busy periods caused by the other streams. Comparing the A and A' systems, as well as the B and B' systems, we have:

$$U_{X_1+X_2+X_3}(t) = U_{X_1+X_2}(t) + extra\_in\_System\_A(t) \tag{A.4}$$

$$U_{X_1+X_3}(t) = U_{X_1}(t) + extra\_in\_System\_B'(t) \tag{A.5}$$

where $extra\_in\_System\_A(t)$ represents the amount of unfinished work from X_3 packets in a queue whose server takes vacations during busy periods caused by the X_1 and X_2 streams. Likewise, $extra\_in\_System\_B'(t)$ represents the amount of unfinished work from X_3 packets when vacations are only during X_1 busy periods. Since busy periods caused by the X_1 stream are subintervals of busy periods caused by the combined $X_1 + X_2$ stream, the X_3 packets in System A experience longer server vacations, and we have:

$$extra\_in\_System\_A(t) \geq extra\_in\_System\_B'(t) \tag{A.6}$$

Using (A.4)-(A.6) verifies (A.3) and concludes the proof. □

Intuitively interpreted, the monotonicity property means that the amount of blocking incurred by the (X_1+X_2) process intermixing with the X_3 process is larger than the amount incurred by the X_1 process alone mixing with the X_3 process.

These three lemmas alone are sufficient to develop some very general convexity results for unfinished work in $*/*/1$ queues. It seems reasonable to suspect that the same three lemmas can be re-formulated in terms of packet occupancy (rather than unfinished work) when all packets have FIFO service. More precisely, suppose that $N_X(t)$ represents the number of packets in a FIFO queueing system with input stream $X(t)$. We can define the *Occupancy Blocking Function* $\alpha_{X_1,X_2}(t)$ in a manner similar to (A.2):

Definition 8. *The* occupancy blocking function $\alpha_{X_1,X_2}(t)$ *between two streams* X_1 *and* X_2 *is the function:*

$$\alpha_{X_1,X_2}(t) \triangleq N_{X_1+X_2}(t) - N_{X_1}(t) - N_{X_2}(t) \tag{A.7}$$

With this new definition of blocking in terms of packet occupancy, it can be shown that, for FIFO service, the non-negativity and symmetry properties still hold ($\alpha_{X_1,X_2}(t) \geq 0, \alpha_{X_1,X_2}(t) = \alpha_{X_2,X_1}(t)$). However, in Chapter Appendix A.B we furnish a counterexample demonstrating that, even under FIFO service, the occupancy monotonicity property does not hold for general variable length service time systems.

Such an example relies heavily on the fact that we have variable length packets. Indeed, in Chapter Appendix A.D it is shown that if all packets have fixed lengths L and service is non-preemptive work conserving, then the packet occupancy blocking function $\alpha_{X_1,X_2}(t)$ satisfies the non-negativity, symmetry, and monotonicity properties for all time.

A.3 Exchangeable Inputs and Convexity

In this section and the next, we use the non-negativity, symmetry, and monotonicity properties to show that any moment of unfinished work in a $*/*/1$ queue is a convex function of the input rate λ. To do this, we must first specify how an arbitrary input process can be parameterized by a single rate value. The parameterization should be such that an input stream of rate 2λ can be viewed as being composed of two similar streams of rate λ. Otherwise, it is clear that the convexity result may not hold. Indeed, consider an input stream $X_1(t)$ delivering bursty data at rate λ, and another stream $X_2(t)$ also delivering data at rate λ according to some other, less bursty process. If the $X_1(t)$ and $X_2(t)$ processes are sequentially added as inputs to a queue, the expected increment in unfinished work due to the additional $X_2(t)$ input may not be as large as the initial increment due to the $X_1(t)$ input. This happens if the $X_2(t)$ process is much smoother than $X_1(t)$, or if it is constructed to have packet arrivals precisely at idle periods of the queue with the $X_1(t)$ input alone.

Here, we consider the input rate λ as a discrete quantity which is varied by adding or removing substreams of the same "type" from the overall input process. We begin by developing the notion of exchangeable random variables.

Definition 9. *A collection of* M *random variables are* exchangeable *if:*

$$p_{X_1,X_2,\ldots,X_M}(x_1,\ldots,x_M) = p_{\tilde{X}_1,\tilde{X}_2,\ldots,\tilde{X}_M}(x_1,\ldots,x_M) \tag{A.8}$$

for every $(\tilde{X}_1,\ldots,\tilde{X}_M)$ *permutation of* $(X_1,...,X_M)$*, where* $p_{X_1,X_2,\ldots,X_M}(x_1,\ldots,x_M)$ *is the joint density function.*

Thus, exchangeable random variables exhibit a simple form of symmetry in their joint distribution functions.[4] Definitions for random variables to be *conditionally exchangeable* given some event ω can be similarly defined: The distributions in (A.8) are simply replaced by conditional distributions. It is clear that any set of independent and identically distributed (i.i.d.) random variables are exchangeable. Thus, exchangeable variables form a wider class than i.i.d. variables, and hence statements which apply to exchangeable variables are more general. Unlike i.i.d. variables, however, it can be seen that if random variables $(X_1,...,X_M)$ are conditionally exchangeable given some other random variable θ, then they are exchangeable.

We can extend this notion of exchangeability to include random *processes* that represent packet arrival streams. The following definition captures the idea that for any sample path realization of exchangeable processes $(X_1(t),...,X_M(t))$, the permuted sample path $(\tilde{X}_1(t),\ldots,\tilde{X}_M(t))$ is "equally likely":[5]

Definition 10. *Random processes* $(X_1(t),...,X_M(t))$ *are* exchangeable *if for any permutation* $(\tilde{X}_1(t),...,\tilde{X}_M(t))$*, we have* $\mathbb{E}\left\{\Phi(X_1,\ldots,X_M)\right\} = \mathbb{E}\left\{\Phi(\tilde{X}_1,\ldots,\tilde{X}_M)\right\}$ *for every operator* $\Phi()$ *which maps the processes to a single real number.*

Definition 11. *Random processes* $(X_1(t),...,X_M(t))$ *are* conditionally exchangeable given process $\theta(t)$ *if for every permutation* $(\tilde{X}_1(t),...,\tilde{X}_M(t))$*, we have* $\mathbb{E}\left\{\Phi(X_1,\ldots,X_M,\theta)\right\} = \mathbb{E}\left\{\Phi(\tilde{X}_1,\ldots,\tilde{X}_M,\theta)\right\}$ *for every real valued operator* $\Phi()$ *which acts on the processes.*

Hence, random processes are exchangeable if their joint statistics are invariant under every permutation. Note that the $\Phi()$ operator maps a set of *sample paths* to a real number.

[4] See [119] for an interesting treatment of the properties of exchangeable variables.

[5] The ideas developed here are closely related to the theory of *stochastic coupling*. See [119] for a formal treatment of the theory, and [133] for an application to optimal scheduling in symmetric queueing systems.

For example, it could correspond to the mapping of the input process $X(t)$ to its unfinished work at a particular time t^*. Exchangeable processes have the same properties as their random variable counterparts. In particular, if processes $(X_1, ..., X_M)$ are exchangeable given a process θ, then:

- Processes $(X_1, ..., X_M)$ are exchangeable.
- Processes $(X_k, ..., X_M)$ are exchangeable given processes $X_1, ..., X_{k-1}, \theta$.
- If $\Psi()$ is an operator that maps processes $X_1(t), X_2(t)$ and $\theta(t)$ to another process $Z(t) = \Psi(X_1, X_2, \theta)$, then $\Psi(X_1, X_2, \theta)$ and $\Psi(X_2, X_1, \theta)$ are exchangeable processes given $\theta(t)$.

The above properties are simple consequences of the definitions, where the last property follows by defining the operator $\tilde{\Phi}(X_1, X_2, \theta) \triangleq \Phi(\Psi(X_1, X_2, \theta), \Psi(X_2, X_1, \theta), \theta)$. Below we provide three examples of exchangeable input processes that can act as input streams to a queueing system:

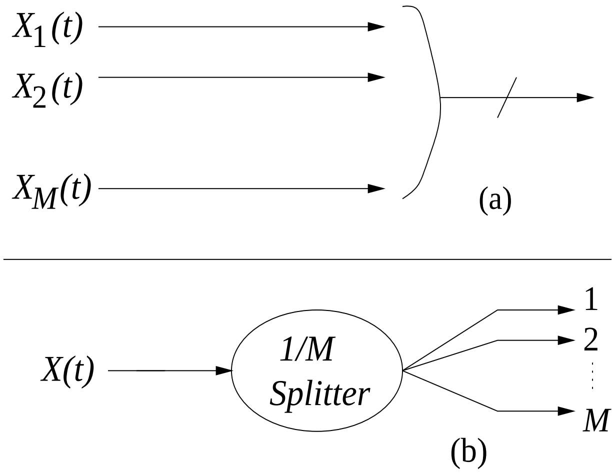


Figure A-6: M exchangeable inputs in the case of (a) the collection of i.i.d. $*/*$ processes $\{X_i\}$ and (b) probabilistic splitting of a $*/*$ process X into M substreams.

Example 1: Any general $*/*$ processes $\{X_i(t)\}$ independent and identically distributed over M input lines (Fig. A-6a).

Example 2: Any general $*/*$ process $X(t)$ which is split into M streams by routing each packet to stream $i \in \{1, \dots, M\}$ with equal probability (Fig. A-6b).

Example 3: Any arbitrary collection of M processes $(X_1(t), ..., X_M(t))$ which are randomly permuted (with each permutation equally likely).

Notice that Example 1 demonstrates the fact that i.i.d. inputs are exchangeable. However, Example 2 illustrates that exchangeable inputs form a more general class of processes by providing an important set of input streams which are not independent yet are still exchangeable. Notice that this probabilistic routing can be extended to include "state-dependent" routing where the probability of routing to queue i depends on where the last packet was placed. The third example shows that an exchangeable input assumption is a good a-priori model to use when an engineer is given simply a "collection of wires" from various sources, and has no a-priori way of distinguishing the process running over "wire 1" from the process running over "wire 2."

We now examine how the unfinished work in a queue changes when a sequence of exchangeable inputs are added. Let $\theta(t)$ be an arbitrary background input process, and let $X_1(t)$ and $X_2(t)$ be two processes which are exchangeable given $\theta(t)$. Let $U_X(t)$ represent the unfinished work process as a function of time in a queue with an input process $X(t)$ running through it. Furthermore, let $f(u)$ represent any convex, non-decreasing function of the real number u for $u \geq 0$. We assume that the expected value of $f(U_X)$ is well defined. (Note that expectations over functions of the form $f(u) = u^k$ represent k^{th} moments of unfinished work). The following theorem shows that incremental values of queue cost are non-decreasing with each additional input.

Theorem 19. *For any particular time t^*, we have:*

$$\mathbb{E}f\left(U_{\theta+X_1+X_2}(t^*)\right) - \mathbb{E}f\left(U_{\theta+X_1}(t^*)\right) \geq \mathbb{E}f\left(U_{\theta+X_1}(t^*)\right) - \mathbb{E}f\left(U_{\theta}(t^*)\right) \tag{A.9}$$

Proof. Define the following processes:

$$\begin{aligned} \Delta_1(t) &\triangleq U_{\theta+X_1}(t) - U_\theta(t) && \text{(A.10)} \\ \Delta_2(t) &\triangleq U_{\theta+X_1+X_2}(t) - U_{\theta+X_1}(t) && \text{(A.11)} \end{aligned}$$

We then find, by using the blocking function properties developed in the previous section:

$$\begin{aligned} \Delta_2(t) = U_{X_2}(t) + \beta_{\theta+X_1,X_2}(t) &\geq U_{X_2}(t) + \beta_{\theta,X_2}(t) \\ &= U_{\theta+X_2}(t) - U_\theta(t) \triangleq \tilde{\Delta}_1(t) && \text{(A.12)} \end{aligned}$$

where we have defined $\tilde{\Delta}_1(t) \triangleq U_{\theta+X_2}(t) - U_\theta(t)$. Because $X_2(t)$ and $X_1(t)$ are exchangeable given $\theta(t)$, and because the $\tilde{\Delta}_1(t)$ and $\Delta_1(t)$ processes are derived from the same mapping of inputs to unfinished work, it follows that $\tilde{\Delta}_1(t)$ and $\Delta_1(t)$ are exchangeable processes given $\theta(t)$. Thus, for any time t^*, inequality (A.12) states that $\Delta_2(t^*)$ is a random variable that is always greater than or equal to another random variable which has the same distribution as $\Delta_1(t^*)$.

We now use an increasing increments property of non-decreasing, convex functions $f(u)$.

Fact: For non-negative real numbers a, b, x, where $a \geq b$, we have:

$$f(a+x) - f(a) \geq f(b+x) - f(b) \tag{A.13}$$

Using this fact and defining $a \triangleq U_{\theta+X_1}(t^*)$, $x \triangleq \Delta_2(t^*)$, and $b \triangleq U_\theta(t^*)$, we have:

$$\begin{aligned} f(U_{\theta+X_1}(t^*) + \Delta_2(t^*)) - f(U_{\theta+X_1}(t^*)) &\geq f(U_\theta(t^*) + \Delta_2(t^*)) - f(U_\theta) && \text{(A.14)} \\ &\geq f(U_\theta(t^*) + \tilde{\Delta}_1(t^*)) - f(U_\theta(t^*)) && \text{(A.15)} \end{aligned}$$

Inequality (A.14) follows from from (A.13) and the fact that $U_{\theta+X_1}(t^*) \geq U_\theta(t^*)$. Inequality (A.15) follows because $\Delta_2(t^*) \geq \tilde{\Delta}_1(t^*)$ (from (A.12)) and from the non-decreasing property of $f()$. Taking expectations of the inequality above, we find:

$$\mathbb{E}f\left(U_{\theta+X_1}(t^*) + \Delta_2(t^*)\right) - \mathbb{E}f\left(U_{\theta+X_1}(t^*)\right) \geq \mathbb{E}f\left(U_\theta(t^*) + \tilde{\Delta}_1(t^*)\right) - \mathbb{E}f\left(U_\theta(t^*)\right) \tag{A.16}$$

Using the fact that $\tilde{\Delta}_1(t)$ and $\Delta_1(t)$ are exchangeable given $\theta(t)$, we can replace the $\mathbb{E}f\left(U_\theta(t^*) + \tilde{\Delta}_1(t^*)\right)$ term on the right hand side of (A.16) with $\mathbb{E}f\left(U_\theta(t^*) + \Delta_1(t^*)\right)$, which yields the desired result. □

The theorem above can be used to immediately establish a convexity property of unfinished work in a work conserving queue with a collection of exchangeable inputs. Assume we have such a collection of M streams $(X_1, ..., X_M)$ which are exchangeable given another background stream $\theta(t)$. Assume that each of the streams X_i has rate λ_δ. The total input process to the queue can then be viewed as a function of a discrete set of rates $\lambda = n\lambda_\delta$ for $n \in \{0, 1, \ldots, M\}$. Let $\mathbb{E}f\left(U[n\lambda_\delta]\right)$ represent the expectation of a function $f()$ of the unfinished work (at some particular time t^*, which is suppressed for notational simplicity) when the input process consists of stream $\theta(t)$ along with a selection of n of the M exchangeable streams. Hence:

$$\mathbb{E}f\left(U[n\lambda_\delta]\right) \stackrel{\triangle}{=} \mathbb{E}f\left(U_{\theta+X_1+\ldots+X_n}(t^*)\right) \;\; (0 \leq n \leq M) \tag{A.17}$$

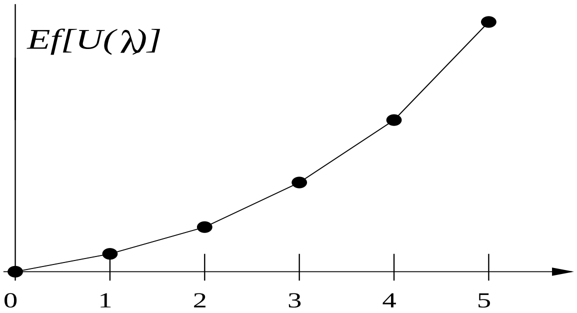


Figure A-7: Convexity of unfinished work as a function of the discrete rate parameter λ.

Theorem 20. *At any specific time t^*, the function $\mathbb{E}f\left(U[\lambda]\right)$ is monotonically increasing and convex in the discrete set of rates λ ($\lambda = n\lambda_\delta$, $n \in \{0, 1, \ldots, M\}$). In particular, any moment of unfinished work is convex (see Fig. A-7).*

Proof. Convexity of a function on a discrete set of equidistant points is equivalent to proving successive increments are monotonically increasing (Fig. A-7). Hence, the statement is equivalent to:

$$\mathbb{E}f\left(U[(n+2)\lambda_\delta]\right) - \mathbb{E}f\left(U[(n+1)\lambda_\delta]\right) \geq \mathbb{E}f\left(U[(n+1)\lambda_\delta]\right) - \mathbb{E}f\left(U[n\lambda_\delta]\right) \tag{A.18}$$

Defining the 'background stream' $\phi(t) = \theta(t) + X_1(t) + \ldots + X_n(t)$, we find that inequality (A.18) follows immediately from Theorem 19. □

A.3.1 Waiting Times

Notice that in Theorems 19 20, expectations were taken at any particular time t^*. It is not difficult to show that this property implies steady state unfinished work is convex, whenever such steady state limits exist. Moreover, we can allow t^* to be a time of special interest, such as the time when a packet from the X_1 stream enters the system. In FIFO queues, the unfinished work in the system at this special time represents the amount of waiting time W that the entering packet spends in the queue before receiving service. In this way, we show that waiting time increments are convex after the first stream is added. Specifically, for a system with a background input $\theta(t)$ and M inputs $\{X_1, ..., X_M\}$ which are exchangeable given $\theta(t)$, we define the following steady state moments (which are functions of the discrete set of input rates $\lambda \in \{0, \lambda_\delta, 2\lambda_\delta, \ldots, M\lambda_\delta\}$):

$\mathbb{E}f\left(W_\theta^{(q)}[\lambda]\right), \mathbb{E}f\left(W_\theta[\lambda]\right) =$ Steady state waiting time moment corresponding to the time a packet from background stream $\theta(t)$ spends in the *queue* and in the *system*, respectively, when the controllable input rate is λ

$\mathbb{E}f\left(W_X^{(q)}[\lambda]\right), \mathbb{E}f\left(W_X[\lambda]\right) =$ Steady state waiting time moment corresponding to the time a packet from a controllable input stream spends in the *queue* and in the *system*, respectively, when the controllable input rate is λ

$\mathbb{E}f\left(N[\lambda]\right) =$ Steady state moment of the number of packets in the system (from both the background and controllable input streams) when the controllable input rate is λ

Formally, the steady state waiting time moments are defined:

$$\mathbb{E}f\left(W\right) \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}f\left(W_k\right)$$

where W_k represents the waiting time of the k^{th} packet of the appropriate input stream. Likewise, the steady state occupancy moment is defined:

$$\mathbb{E}f\left(N\right) \triangleq \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{E}f\left(N(\tau)\right) d\tau$$

Assuming such steady state moments exist for the convex increasing function $f()$ of interest, we have:

Corollary 8. *In FIFO queueing systems:*

(a) $\mathbb{E}f\left(W_\theta^{(q)}[\lambda]\right)$ *and* $\mathbb{E}f\left(W_\theta[\lambda]\right)$ *are non-decreasing and convex in the discrete set of rates* $\lambda \geq 0$ *(i.e.,* $\lambda = n\lambda_\delta, n \in \{0, 1, \ldots, M\}$*).*

(b) $\mathbb{E}f\left(W_X^{(q)}[\lambda]\right)$ *and* $\mathbb{E}f\left(W_X[\lambda]\right)$ *are non-decreasing and convex in the discrete set of rates* $\lambda > 0$*.*

(c) $\mathbb{E}\left\{N[\lambda]\right\}$ *is non-decreasing and convex in the discrete set of rates* $\lambda \geq 0$*.*

Caveat: Note that in (b), waiting times for packets from the controllable input streams are not defined when $\lambda = 0$. Thus, convexity in this case is defined only for $\lambda > 0$. Further note that in (c), the function $f()$ is intentionally absent from the expectation, as we can only establish convexity of the first moment of packet occupancy in this general setting with variable length packets.

Proof. To prove (a), let p be a certain packet from the θ input stream which arrives to the system at time t_p. When n of the controllable inputs are applied to the system, $U_{\theta+X_1+...+X_n}(t_p^-)/\mu$ represents the amount of time packet p is buffered in the queue, and $U_{\theta+X_1+...+X_n}(t_p^+)/\mu$ is the total system time for packet p. From Corollary 20, $\mathbb{E}f\left(U[\lambda]\right)$ is a non-decreasing, convex function of $\lambda \geq 0$ when unfinished work is evaluated at any time t^*, including times $t^* = t_p^-$ and $t^* = t_p^+$. Hence, the expected waiting time of packet p in the queue and in the system is a non-decreasing convex function of the controllable input rate. Because this holds for any packet p from the θ stream, the expected waiting time averaged over all θ packets is also convex, completing the proof.

To prove (b), now let packet p represent a packet from the first controllable input stream X_1. Considering the sum process $\theta(t) + X_1(t)$ as a combined background stream with respect to inputs $\{X_2, ..., X_n\}$ (and noting that $\{X_2, ..., X_n\}$ remain exchangeable given $\theta(t) + X_1(t)$), from (a) we know that the expected queueing time and system time of packet p is a non-decreasing convex function of $\lambda \geq \lambda_\delta$. Because inputs $\{X_1, ..., X_M\}$ are exchangeable, the expected waiting time of a packet from stream X_1 is not different from the expected waiting time of a packet from stream X_k (provided that stream X_k is also applied to the system), and the result follows.

To prove (c), let $f(x) = x$. From (b) we know that $\mathbb{E}\left\{W_X[\lambda]\right\}$ is non-decreasing and convex for $\lambda > 0$. It is straightforward to verify that for any such function, the function $\lambda\mathbb{E}\left\{W_X[\lambda]\right\}$ is non-decreasing and convex for $\lambda \geq 0$, where $\lambda\mathbb{E}\left\{W_X[\lambda]\right\}$ is defined to be 0 at $\lambda = 0$. Let λ_θ represent the rate of the $\theta(t)$ stream. By Little's Theorem, it follows that $\mathbb{E}\left\{N[\lambda]\right\} = \lambda\mathbb{E}\left\{W_X(\lambda)\right\} + \lambda_\theta\mathbb{E}\left\{W_\theta[\lambda]\right\}$ is non-decreasing and convex in λ, as this is the sum of non-decreasing convex functions. □

One might expect the waiting time $\overline{W}_{av}$ averaged over packets from both the controllable and uncontrollable input streams to be convex. However, note that $\overline{W}_{av}[\lambda] = \left(\frac{\lambda_\theta}{\lambda+\lambda_\theta}\right)\mathbb{E}\left\{W_\theta[\lambda]\right\}+\left(\frac{\lambda}{\lambda+\lambda_\theta}\right)\mathbb{E}\left\{W_X[\lambda]\right\}$ is not necessarily convex even though both $\mathbb{E}\left\{W_\theta[\lambda]\right\}$

and $\mathbb{E}\{W_X[\lambda]\}$ are. Indeed, the following simple example shows that $\overline{W}_{av}[\lambda]$ may even *decrease* as λ increases:

Example: Let background input $\theta(t)$ periodically produce a new packet of service time 10 at times $t = \{0, 100, 200, \ldots\}$. Let input $X_1(t)$ consist of packets of service time 2 occurring periodically at times $t = \{50, 150, 250, \ldots\}$. Hence, packets from $\theta(t)$ and $X_1(t)$ never interfere with each other. We thus have $\overline{W}_{av}(0) = 10$ and $\overline{W}_{av}[\lambda_\delta] = (10 + 2)/2 = 6$.

A.3.2 Packet Occupancy $N(t)$

Notice that the non-negativity, symmetry, and monotonicity properties of the blocking function $\beta_{X_1,X_2}(t)$ were the only queueing features needed to establish convexity of unfinished work $U(t)$. Now suppose that all packets have fixed lengths L, and let $N(t)$ represent the number of packets in the queueing system at time t for some arbitrary arrival process. If service in the queue is work conserving and non-preemptive, it can be shown that the occupancy blocking function $\alpha_{X_1,X_2}(t)$ satisfies the non-negativity, symmetry, and monotonicity properties (see Chapter Appendix A.D for a complete proof). We can thus reformulate Theorems 19 and 20 in terms of packet occupancy. Suppose again that input streams $\{X_1, ..., X_M\}$ are exchangeable given background stream θ. We find:

Corollary 9. *If all packets have fixed lengths L and service is non-preemptive work conserving, then at any particular time t^*, the expectation $\mathbb{E}f(N[\lambda])$ is a convex function of the discrete rates $\lambda \in \{0, \lambda_\delta, 2\lambda_\delta, \ldots, M\lambda_\delta\}$.* □

A.4 Convexity over a Continuous Rate Parameter

In the previous section we dealt with streams of inputs and demonstrated convexity of unfinished work and waiting time moments as streams are removed or added. Here, we extend the theory to include input processes that are parameterized by a continuous rate variable λ. The example to keep in mind in this section is packet-by-packet probabilistic splitting, where individual packets from an arbitrary packet stream are sent to the queue with some probability p. However, the results apply to any general "infinitely splittable" input, which are inputs that can be split into *substreams* according to some splitting method, as described below:

Definition 12. *A packet input process $X(t)$ together with a splitting method is said to be* infinitely splittable *if:*

(1) $X(t)$ or any of its substreams *can be split into disjoint substreams of arbitrarily small rate. Any superposition of disjoint substreams of $X(t)$ is considered to be a substream.*

(2) Any two (potentially non-disjoint) substreams that have the same rate are conditionally exchangeable given the rest of the process.

We emphasize that the above definition incorporates both the input process $X(t)$ and the method of splitting. Notice that any $*/*$ process $X(t)$ is infinitely splittable when using the probabilistic splitting method of independently including packets in a new substream i with some probability p_i. Likewise, probabilistic splitting of the lead packet in systems where blocks of K sequential packets must be routed together can be shown to satisfy the conditions of infinite splittability. However, not all splitting methods are valid. Consider for example the "divide by 2" splitting method, where an input stream is split into two by alternately routing packets to the first stream and then the second. Suppose the base input stream $X(t)$ has rate λ and consists of fixed length packets of unit size. Under this splitting method, any substream of rate $\lambda\frac{k}{2^n}$ can be formed by collecting superpositions of disjoint substreams of rate $\lambda/2^n$ (where k and n are any integers such that $k \leq 2^n$).[6] Thus, the first condition of infinite splittability is satisfied. However, it is not clear how a substream $\tilde{X}(t)$ of rate $\lambda/2$ is distributed. For example, the original stream $X(t)$ could be split into two substreams, one of which is $\tilde{X}(t)$ and represents the sequence of every other packet arrival from $X(t)$. (The odd substream may be differently distributed from the even substream, but this can be taken care of by randomly permuting the two so that it is impossible to know if $\tilde{X}(t)$ contains the odd samples or the even samples.) Alternately, the "divide by 2" splitting method might be used to form $\tilde{X}(t)$ by iteratively splitting $X(t)$ into eight substreams of rate $\lambda/8$, a random four of which are grouped together to form the rate $\lambda/2$ substream. Clearly the two approaches to building a rate $\lambda/2$ substream do not generally lead to identically distributed processes, as the first approach leads to a rate $\lambda/2$ substream that never contains two successive packets from the original stream, while the second approach leads to a $\lambda/2$ substream that might contain two successive packets. Thus, the divide-by-2 splitting method satisfies the first condition of the above definition

[6] Formally, streams of arbitrary irrational rates $\tilde{\lambda} < \lambda$ can be formed from the "divide by 2" splitting method by a countably infinite superposition of substreams, where the substreams that are added have progressively smaller rates.

but not the second.

With the above definition, it can be seen that an infinitely splittable input process $X(t)$ can be written as the sum of a large number of exchangeable substreams. Specifically, it has the property that for any $\epsilon > 0$, there exists a large integer M such that:

$$X(t) = \sum_{i=1}^{M} x_i(t) + \tilde{x}(t)$$

where $(x_1(t), ..., x_M(t))$ are exchangeable substreams, each with rate λ_δ, $\tilde{x}(t)$ has rate $\tilde{\lambda}_\delta$, and $\tilde{\lambda}_\delta < \lambda_\delta < \epsilon$.

We now use the blocking function to establish continuity of expected moments of unfinished work as a function of the continuous rate parameter λ. As before, these results also apply to waiting times in FIFO systems.

Again we assume that $f(u)$ is a non-decreasing convex function over $u \geq 0$. Suppose $X(t)$ is an infinitely splittable input process with total rate λ_{tot}. Suppose also that all exchangeable component processes of $X(t)$ are also exchangeable given the background input process $\theta(t)$. Let $\mathbb{E}f\left(U[\lambda_{tot}]\right)$ represent the expectation of a function of unfinished work at a particular time t^* in a queue with this input and background process. We assume here that $\mathbb{E}f\left(U[\lambda_{tot}]\right)$ is finite.

Theorem 21. *$\mathbb{E}f\left(U[\lambda]\right)$ can be written as a pure function of the continuous rate parameter λ, where $\lambda \in [0, \lambda_{tot}]$ is a rate achieved by some substream of the infinitely splittable $X(t)$ input. Furthermore, $\mathbb{E}f\left(U[\lambda]\right)$ is a monotonically increasing and continuous function of λ.*

Proof. The proof is given in Chapter Appendix A.A. We note that the proof of continuity uses a simple ϵ, δ argument (in the manner of the classic proof that the function $f(x) = x^2$ is continuous) together with the machinery of the blocking function. □

The continuity property of Theorem 21 allows us to easily establish the convexity of any moment of unfinished work (and packet waiting time) in a $*/*/1$ queue as a function of the continuous input rate λ. Let $X(t)$ be an infinitely splittable input process, and suppose that every collection of exchangeable components of $X(t)$ are also exchangeable given the background process $\theta(t)$. Then:

Theorem 22. *At any particular time t^*, the function $\mathbb{E}f\left(U[\lambda]\right)$ is convex over the continuous variable $\lambda \in [0, \lambda_{tot}]$. Likewise, if service is FIFO, then $\mathbb{E}f\left(W[\lambda]\right)$ is also convex.*

Proof. We wish to show that the function $\mathbb{E}f\left(U[\lambda]\right)$ always lies below its chords. Thus, for any three rates $\lambda_1 < \lambda_2 < \lambda_3$, we must verify that:

$$\mathbb{E}f\left(U[\lambda_2]\right) \leq \mathbb{E}f\left(U[\lambda_1]\right) + (\lambda_2 - \lambda_1)\frac{\left(\mathbb{E}f\left(U[\lambda_3]\right) - \mathbb{E}f\left(U[\lambda_1]\right)\right)}{(\lambda_3 - \lambda_1)} \tag{A.19}$$

We know from Theorem 20 in Section A.3 that the unfinished work function is convex over a discrete set of rates when the input process is characterized by a finite set of M exchangeable streams $(x_1, ..., x_M)$. We therefore consider a discretization of the rate axis by considering the sub-processes $(x_1, ..., x_M)$ of the infinitely splittable process $X(t)$, where each x_i has a small rate δ. In this discretization, we have rates:

$$\tilde{\lambda}_1 = k_1\delta, \tilde{\lambda}_2 = k_2\delta, \tilde{\lambda}_3 = k_3\delta \tag{A.20}$$

where the rates $(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)$ can be made arbitrarily close to their counterparts $(\lambda_1, \lambda_2, \lambda_3)$ by choosing an appropriately small value of δ. Now, from the discrete convexity result, we know:

$$\mathbb{E}f\left(U[\tilde{\lambda}_2]\right) \leq \mathbb{E}f\left(U[\tilde{\lambda}_1]\right) + (\tilde{\lambda}_2 - \tilde{\lambda}_1)\frac{\mathbb{E}f\left(U[\tilde{\lambda}_1]\right) - \mathbb{E}f\left(U[\tilde{\lambda}_1]\right)}{(\tilde{\lambda}_3 - \tilde{\lambda}_1)} \tag{A.21}$$

By continuity of the $\mathbb{E}f\left(U[\lambda]\right)$ function, we can choose the discretization unit δ to be small enough so that the right hand side of (A.21) is arbitrarily close to the right hand side of the (currently unproven) inequality (A.19). Simultaneously, we can ensure that the left hand sides of the two inequalities are arbitrarily close. Thus, the known inequality (A.21) for the discretized inputs implies inequality (A.19) for the infinitely splittable input. We thus have convexity of unfinished work at any point in time, which also implies convexity of waiting time in FIFO systems. □

A.5 Multiple Queues in Parallel

We now consider the system of N queues in parallel as shown in Fig. A-8. The servers of each queue k have linespeeds μ_k and arbitrary background packet input processes $\theta_k(t)$. An arbitrary input process $X(t)$ also enters the system, and $X(t)$ is rate-controllable in that a router can split $X(t)$ into substreams of smaller rate. These substreams can be distributed according to an N-tuple rate vector $(\lambda_1, ..., \lambda_N)$ over the multiple queues.

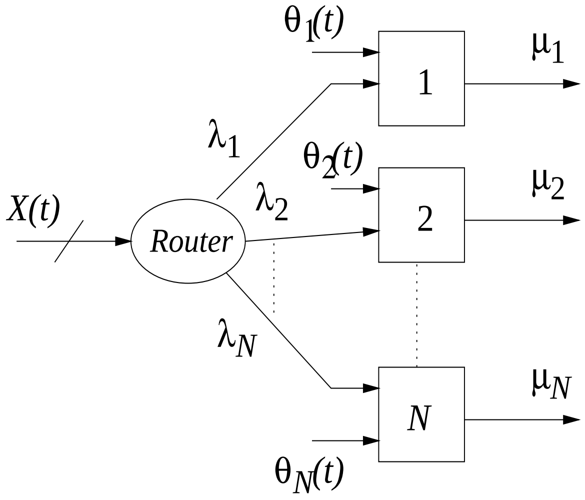


Figure A-8: Multiple queues in parallel with different background processes $\theta_i(t)$ and server rates μ_i.

We consider both the case when $X(t)$ is an infinitely splittable process (as in packet-based probabilistic splitting), and the case when $X(t)$ is composed of a finite collection of M exchangeable streams. The problem in both cases is to route the substreams by forming an optimal rate vector that minimizes some network cost function. We assume the cost function is a weighted summation of unfinished work and/or waiting time moments in the queues. Specifically, we let $\{f_k(u)\}$ be a collection of convex, non-decreasing functions on $u \geq 0$. Suppose that the queues reach some steady state behavior, and let $\mathbb{E}f\left(U_k[\lambda_k]\right)$ represent the steady state moment of unfinished work in queue k when an input stream of rate λ_k is applied. Let $\mathbb{E}f\left(W_k[\lambda_k]\right)$ represent the steady moment of waiting time for queue k.

Theorem 23. *If queues are work conserving and $X(t)$ is either a finitely*[7] *or infinitely rate splittable process given $\{\theta_k(t)\}$, then:*

(a) Cost functions of the form

$$C(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} \mathbb{E}f_k(U_k[\lambda_k]) \tag{A.22}$$

are convex in the multivariable rate vector $(\lambda_1, \ldots, \lambda_N)$.

[7] We note that convexity on a discrete set of points is equivalent to convexity of the piecewise linear interpolation, see Fig. A-7.

(b) If service is FIFO, then cost functions of the form

$$C(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} \lambda_k \mathbb{E} f_k(W_k[\lambda_k]) \tag{A.23}$$

are convex (where the $W_k[\lambda_k]$ values represent waiting times of packets from the controllable inputs).

(c) If service is FIFO and $N_k(\lambda_k)$ represents the number of packets in queue k in steady state, then cost functions of the following form are convex:

$$C(\lambda_1, \ldots, \lambda_N) = \sum_{k=1}^{N} \lambda_k c_k \mathbb{E}\left\{N_k[\lambda_k]\right\} \quad (\{c_k\} \geq 0) \tag{A.24}$$

Proof. Since $\mathbb{E} f_k(W_k[\lambda_k])$ is convex and non-decreasing for $\lambda_k > 0$, the function $\lambda_k \mathbb{E} f_k(W_k[\lambda_k])$ is convex on $\lambda_k \geq 0$. Thus, the cost functions in (a) and (b) are summations of convex functions, so they are convex. Part (c) follows from (b) by noting that, from Little's Theorem, $\mathbb{E}\{N\} = \lambda \mathbb{E}\{W\}$. □

Convexity of the cost function $C(\lambda_1, \ldots, \lambda_N)$ can be used to develop optimal rate distributions $(\lambda_1^o, \ldots, \lambda_N^o)$ over the simplex constraint $\lambda_1 + \ldots + \lambda_N = \lambda_{tot}$. For symmetric cost functions, which arise when the background processes $\{\theta_k(t)\}$ and the linespeeds $\{\mu_k\}$ are the same for all queues $k \in \{1, \ldots, N\}$, the optimal solution is particularly simple. It is clear in this case that the uniform distribution (or as near to this as can be achieved) is optimal and minimizes cost. Thus, in the symmetric case we want to spread the total input stream evenly amongst all of the queues in order to take full advantage of the bandwidth that each queue provides. In the asymmetric case when background streams and linespeed processes are not the same, the optimal rate vector deviates from the uniform allocation to favor queues with faster linespeeds and/or less background traffic.

Let us suppose the cost function $C(\lambda_1, \ldots, \lambda_N)$ is known. Convexity of $C()$ tells us that any local minimum of the cost function we find must also be a global minimum. It also tells us a great deal more [15], as we illustrate for both finitely and infinitely splittable inputs $X(t)$ below.

A.5.1 (Stream-Based) Finitely Splittable $X(t)$

Here we assume that the input $X(t)$ is a finite collection of M streams, which are exchangeable given the background processes $\{\theta_k(t)\}$. We want to distribute the streams over the N queues available. We can thus write the cost function $C(M_1, \ldots, M_N)$ as a function of an integer N-tuple $(M_1, \ldots, M_N)$ (rather than a rate N-tuple) which describes the number of streams we route to each queue.

If the weight functions $f_k()$ are identical for all k, and all queues have identical linespeeds and exchangeable background inputs, then the cost function $C(M_1, \ldots, M_N)$ is convex symmetric and the optimal solution is to allocate according to the load balanced strategy of assigning $\lceil M/n \rceil$ streams to $(M) \mod (N)$ of the queues, and $\lfloor M/N \rfloor$ streams to the remaining queues. In the non-symmetric case, we must consider other allocations and test them by evaluating the cost function.

Lemma 24. *Given a convex cost function $C(M_1, \ldots, M_N)$ of the form specified in Theorem 23, the optimal allocation vector can be obtained by sequentially adding streams, greedily choosing at each iteration the queue which increases the total cost $C()$ the least. This yields a cost-minimizing vector $(M_1^o, \ldots, M_N^o)$ after $M+N-1$ evaluations/estimations of the cost function.*

Proof. This lemma follows as a special case of a theory of integer optimization over separable convex functions (see [46]). We provide a simplified and independent proof in Chapter Appendix A.E for completeness. □

In the special case of routing M exchangeable inputs over N queues with the same server rate μ, it is clear that the load-balanced strategy minimizes all symmetric cost functions, and in particular it minimizes the expected unfinished work in all of the queues at every instant of time, which extends the result of [56] [25] [66] to $*/*$ inputs and stream based routing. It is tempting to conjecture that load balancing also stochastically minimizes the total workload at every instant of time, as is proved in [80] for homogeneous queues, Bernoulli splitting, and $*/M$ inputs. However, this is not the case, as illustrated by a simple counterexample given in Chapter Appendix A.C.

<u>Example:</u> We consider the system of Fig. A-8 when there are $N = 4$ queues and the input $X(t)$ consists of 200 independent streams that produce packets periodically every P seconds. Packets have a fixed length of L bits. The streams are unsynchronized, and hence

200 packets arrive in a uniformly distributed fashion over any time interval of length P. Such input streams are models for continuous bit rate traffic, such as digital voice or video data.

The problem is to distribute the streams over the 4 queues while minimizing the cost function, which we take to be the total expected number of packets in the system (this is the cost function of Theorem 23c). We first assume all server rates μ_i are the same, and all background streams $\theta_i(t)$ are exchangeable. In this case, we immediately know the optimal stream allocation vector is $(50, 50, 50, 50)$. Likewise for this symmetric example, if there are 201 streams, then the optimal vector is $(51, 50, 50, 50)$ (or any of the three other near-symmetric allocations).

Now suppose that we have server rates $(2, 1, 1, 1)$ and that there are 10 background streams of the same type at queue 4. In this asymmetric case, we must use the cost function to determine optimal allocation using the sequentially greedy method. The complementary occupancy distribution in a single queue with M inputs of period P, length L, and server rate μ has been derived explicitly in [118]. The result is:

$$Q_n\left(\frac{L}{\mu P}, M\right) = \sum_{i=1}^{M-n} \left(\begin{array}{c} M \\ i+n \end{array}\right) \left(\frac{iL}{\mu P}\right)^{i+n} \left(1 - \frac{iL}{\mu P}\right) \left(\frac{\mu P/L - M + n}{\mu P/L - i}\right) \quad (0 \leq n \leq M-1)$$

The expected number of packets is hence:

$$\overline{N}\left(\frac{L}{\mu P}, M\right) = \sum_{n=0}^{M-1} Q_n\left(\frac{L}{\mu P}, M\right)$$

We therefore use the greedy algorithm with cost function:

$$C(M_1, M_2, M_3, M_4) = \sum_{i=1}^{4} \overline{N}\left(\frac{L}{\mu_i P}, M_i\right)$$

Using $L = 1, P = 75$, we find that the optimal allocation vector is: $(100, 37, 37, 26)$. From this solution, we see that—as expected because of the 10 background streams in queue 4—there are approximately 10 more streams allocated to queues 2 and 3 than queue 4. Interestingly, the server rate of queue 1 is twice that of the other queues, although, due to statistical multiplexing gains, the number of streams it is allocated is more than twice the number allocated to the others.

A.5.2 (Packet Based) Infinitely Splittable $X(t)$

Here we consider an infinitely splittable process $X(t)$ with total rate λ_{tot} as an input to the system of Fig. A-8. The problem is to optimally distribute the total rate over the N queues so as to minimize a cost function $C(\lambda_1, \ldots, \lambda_N)$. We assume the cost function is of one of the forms specified in Theorem 23. Each of these had the structure:

$$C(\lambda_1, \ldots, \lambda_N) = g_1(\lambda_1) + \ldots + g_N(\lambda_N)$$

for some convex, non-decreasing functions $g_i(\lambda_i)$.

If background inputs are exchangeable, and if all queues are weighted the same in the cost function, then $C()$ is convex symmetric and the optimal rate allocation is $(\lambda_{tot}/N, ..., \lambda_{tot}/N)$. Otherwise, we can take advantage of the convex structure of the $C()$ function to determine the optimal solution [13].

Each of the functions $g_i(\lambda_i)$ is non-decreasing and convex on some open interval $(0, \tilde{\lambda}_i)$. It can be shown that these properties ensure $g_i(\lambda_i)$ is right-differentiable. They are also sufficient to establish the correctness of a Lagrange Multipliers approach to cost minimization. Given the Lagrangian:

$$L(\lambda_1, \ldots, \lambda_N, \gamma) = C(\lambda_1, \ldots, \lambda_N) + \gamma \left(\lambda_{tot} - \sum_{i=1}^{N} \lambda_i \right)$$

where γ is the Lagrange Multiplier, we differentiate (from the right) with respect to λ_i to obtain

$$\frac{d}{d\lambda_i} g_i(\lambda_i) = \gamma \text{ for all } i \in \{1, \ldots, N\}$$

subject to the simplex constraint $\lambda_1 + \ldots + \lambda_N = \lambda_{tot}$. Fig. A-9 illustrates this solution. If we define:

$$\lambda_i(\gamma) = \text{Largest } \lambda \text{ such that } g_i'(\lambda) \leq \gamma$$

then from Fig. A-9 we see that we increase the value of γ until $\lambda_1(\gamma) + \ldots + \lambda_N(\gamma) = \lambda_{tot}$. The resulting rate vector yields the optimal solution. Although the form of the solution appears different from that of the discrete rate scenario, in fact the two can be viewed as exactly the same. Indeed, the continuous rate solution corresponds to sequentially allocating an infinitesimal portion of the total rate in a greedy manner.

Notice that a fast bisection type algorithm can be developed to find this optimal rate vector. First, two bracketing values γ_{Low} and γ_{Hi} are found which yield $\lambda_1(\gamma) + \ldots + \lambda_N(\gamma)$ values above and below λ_{tot}, respectively. The bisection routine proceeds as usual until the rate vector converges to a solution within acceptable error limits.

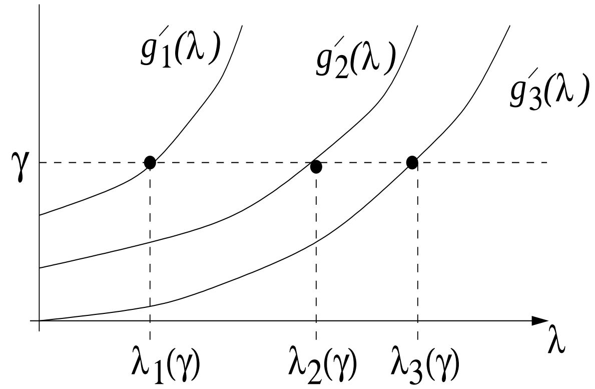


Figure A-9: A simple set of $\frac{d}{d\lambda_i} g_i(\lambda_i)$ curves.

Example: The following simple example for systems with memoryless arrivals and packet lengths demonstrates the method. The example fits into the optimal routing framework of [14] [22] [19], and the solution we obtain can be verified using the techniques developed there. Suppose, for simplicity, that there are no background arrivals $\{\theta_i(t)\}$. Let the arrival process be Poisson with rate λ, and the packet length process be i.i.d. and memoryless with an average packet length of 1 unit. We assume that we are using probabilistic Bernoulli routing where each packet is routed independently of the next. Let $\overline{N}_i$ represent the average number of packets in queue i, and define cost function:

$$C(\lambda_1, \ldots, \lambda_N) = \sum_i c_i \overline{N}_i[\lambda_i]$$

where

$$g_i(\lambda_i) = c_i \overline{N}_i(\lambda_i)$$

The expected number of packets in an M/M/1 queue is:

$$\overline{N}_i = \frac{\lambda_i/\mu_i}{1 - \lambda_i/\mu_i}$$

Multiplying the above equation by c_i, taking derivatives with respect to λ_i, and setting the result equal to γ for all i, as well as considering the constraint $\lambda_1 + \ldots + \lambda_N = \lambda_{tot}$, we find for all $k \in \{1, \ldots, N\}$:

$$\lambda_k = \mu_k + \frac{\sqrt{c_k \mu_k}\,(\lambda_{tot} - \sum_i \mu_i)}{\sum_i \sqrt{c_i \mu_i}}$$

The above equation is a bit deceptive, in that the summations are taken over all i for which λ_i is positive. The positive λ_i's are determined by first assuming that all are positive and applying the above equation. If any of the λ_i's found are zero or negative, these λ_i's are set to zero and the calculation is repeated using the remaining subset of λ_i's.

This approach to optimal rate allocation is similar to the convex optimization routines described in [14]. There, the authors address packet routing in general mesh networks when input streams can be continuously split according to a rate parameter. They presuppose some convex cost function at each node of the network, which (as an idealization) is completely a function of the overall rate routed to that node. Here, we have considered cost functions that reflect the actual queuing congestion at each node when a general $*/*$ input is applied. We have established that, for a simple network consisting of N parallel queues, the cost functions at each queue are continuous and non-decreasing in the rate parameter, and they are indeed convex.

A.6 Time-Varying Server Rates

Here we consider the system of Fig. A-1 when the constant server of rate μ is replaced by a time varying server of rate $\mu(t)$. Characteristics of an unfinished work sample path $U_X(t)$ for time varying servers are similar to those illustrated in Fig. A-2 for constant server systems, with the exception that the $U_X(t)$ function decreases with a time varying slope $-\mu(t)$. This presents a slight problem for waiting time analysis, as waiting times may no longer be convex. Recall that for constant processing rate servers, the unfinished work $U_X(t)$ is proportional to the amount of time it would take the system to empty if no other packets were to arrive (specifically, this time is $U_X(t)/\mu$). However, this emptying time is not causally known from the system state for time varying servers. Hence, the unfinished work in the system at the time of a packet arrival no longer indicates the amount of time this packet will wait.

On the other hand, convexity analysis of the unfinished work $U_X(t)$ and the number of packets $N_X(t)$ can be performed for this time varying context in a manner similar to the treatment for fixed processing rate systems. Indeed, we can define the unfinished work blocking function $\beta_{X_1,X_2}(t)$ and the packet occupancy blocking function $\alpha_{X_1,X_2}(t)$ as before:

$$\begin{aligned}\beta_{X_1,X_2}(t) &\triangleq U_{X_1+X_2}(t) - U_{X_1}(t) - U_{X_2}(t)\\ \alpha_{X_1,X_2}(t) &\triangleq N_{X_1+X_2}(t) - N_{X_1}(t) - N_{X_2}(t)\end{aligned}$$

By literally repeating the arguments of Section A.2, we can establish that the non-negativity, symmetry, and monotonicity properties still hold for $\beta_{X_1,X_2}(t)$ in this time-varying context. Likewise, if all packets have fixed lengths L and service is non-preemptive, it can be shown for time varying servers that the occupancy blocking function $\alpha_{X_1,X_2}(t)$ also satisfies these three properties (see Chapter Appendix A.D).

Consequently, given a collection of N queues with background input processes $\{\theta_i(t)\}$ and server rate processes $\{\mu_i(t)\}$, together with a (finitely or infinitely distributable) input $X(t)$, we can establish:

Theorem 24. *If the exchangeable components of $X(t)$ are exchangeable given $\{\theta_i(t)\}$ and $\{\mu_i(t)\}$, then $\sum \mathbb{E}f_i(U_i[\lambda_i])$ is convex in the rate vector $(\lambda_1, \ldots, \lambda_N)$. If all packets have a fixed length of L bits and service is non-preemptive, then $\sum \mathbb{E}f_i(N_i[\lambda_i])$ is convex in the rate vector.*

Recall from Little's Theorem that if the expected waiting time $\mathbb{E}\{W(\lambda)\}$ is convex in λ, then so is the expected packet occupancy $\mathbb{E}\{N(\lambda)\}$. However, the converse implication does not follow. Indeed, below we provide a (counter) example which illustrates that—even for fixed length packets under FIFO service—waiting times are not necessarily convex for time varying servers.

(Counter) Example: Consider identical input processes X_1, X_2, X_3 which produce a single packet of length $L = 1$ periodically at times $\{0, 3, 6, 9, ...\}$. Let the server rate be periodic of period 3 with $\mu(t) = 1$ for and $\mu(t) = 100$ for $t \in (2, 3)$. Then $\mathbb{E}\{W_{X_1}\} = 1$, $\mathbb{E}\{W_{X_1+X_2}\} = 1.5$, and $\mathbb{E}\{W_{X_1+X_2+X_3}\} = 1.67$. Clearly the increment in average waiting time when stream X_2 is added is larger than the successive increment when stream X_3 is added. Hence, waiting time is not convex in this time-varying server setting. □

Although waiting times are not necessarily convex, notice that minimizing $\overline{W}_{tot}$ in a parallel queue configuration (Fig. A-8) is accomplished by minimizing $\overline{N}_{tot}$ (since $\overline{N}_{tot} = \lambda_{tot}\overline{W}_{tot}$). For fixed length packets, Theorem 24 ensures this is a convex optimization even for time varying servers. Indeed, notice that expected occupancies $\mathbb{E}\{N_{X_1}\}$, $\mathbb{E}\{N_{X_1+X_2}\}$, and $\mathbb{E}\{N_{X_1+X_2+X_3}\}$ for the above example can be obtained by multiplying $\mathbb{E}\{W_{X_1}\}$, $\mathbb{E}\{W_{X_1+X_2}\}$, and $\mathbb{E}\{W_{X_1+X_2+X_3}\}$ by $\lambda = 1/3$, $2/3$, and $3/3$ respectively, and the resulting values are convex. Indeed, the non-convex values $1, 1.5, 1.67$ become $\frac{1}{3}$, 1, and 1.67, which are just on the borderline of convexity.

A.7 Summary

We have developed general convexity results for $*/*/1$ queues using a new function of two packet stream inputs called the blocking function. Non-negativity, Symmetry, and Monotonicity properties of the blocking function were established. These properties proved to be valuable tools for establishing convexity of unfinished work and waiting time moments in terms of both a discrete and a continuous input rate λ.

We then addressed both stream-based and packet-based routing of general inputs over a collection of N parallel queues with arbitrary background inputs and different linespeeds. Optimal routing algorithms were developed utilizing the convexity results.

This convexity theory can be extended to address more complex variants of the parallel queue problem. One might consider a case when we have a collection of K sets of exchangeable inputs, where each set categorizes a different type of input process. For example, set S_1 could contain multiple exchangeable inputs of the “bursty” type, while set S_2 contains exchangeable inputs of the “continuous bit rate” type. This variation of the problem is closely related to the NP-complete bin packing problem. It would also be interesting to explore convexity and optimal routing in more general mesh networks using these techniques. Such an approach could perhaps establish the validity of known convex optimization routines, as well as provide insights into developing new ones.

Appendix A.A — Continuity of Unfinished Work

Here we show that for any particular time t^*, $\mathbb{E}f\left(U[\lambda]\right)$ is a continuous, monotonically increasing function of λ (Theorem 21 of Section A.4). We utilize the following facts about convex, non-decreasing functions:

Fact 1: If $f(u)$ is non-decreasing and convex, then for any fixed $a \geq 0$ there is a function $g(a,x)$ such that $f(a+x) = f(a)+g(a,x)$, where $g(a,x)$ is a convex, non-decreasing function of x for $x \geq 0$.

Fact 2: Any convex, non-decreasing function $g(x)$ with $g(0) = 0$ has the property that $g(x_1 + x_2) \geq g(x_1) + g(x_2)$ for any $x_1, x_2 \geq 0$.

Note that $\mathbb{E}f\left(U[\lambda]\right) \triangleq \mathbb{E}f\left(U_{X_\lambda}(t^*)\right)$, where $X_\lambda(t)$ is any substream of $X(t)$ with rate λ. This is a well defined function of λ because, by the properties of infinitely splittable inputs, all substreams with the same rate are identically distributed. The fact that $\mathbb{E}f\left(U[\lambda]\right)$ is monotonically increasing in λ follows as a simple consequence of the non-negativity property of the blocking function. Indeed, consider a substream $X_\delta(t)$ of rate δ. We have:

$$\mathbb{E}f\left(U[\lambda+\delta]\right) \triangleq \mathbb{E}f\left(U_{X_\lambda + X_\delta}(t^*)\right) \geq \mathbb{E}f\left(U_{X_\lambda}(t^*)\right) \triangleq \mathbb{E}f\left(U[\lambda]\right)$$

proving monotonicity. Below we prove the continuity property.

Proof. (Continuity) Here we prove that the function $\mathbb{E}f\left(U[\lambda]\right)$ is continuous from the right with respect to the λ parameter. Left continuity can be proven in a similar manner.

Take any λ in the set of achievable rates. We show that:

$$\lim_{\delta \to 0} \mathbb{E}f\left(U[\lambda+\delta]\right) = \mathbb{E}f\left(U[\lambda]\right) \tag{A.25}$$

where δ is the rate of a component process of $X(t)$ which make arbitrarily small. By monotonicity, if δ decreases to zero, then $\mathbb{E}f\left(U[\lambda+\delta]\right) - \mathbb{E}f\left(U[\lambda]\right)$ decreases toward some limit ϵ, where $\epsilon \geq 0$. Suppose now that this inequality is strict, so that $\epsilon > 0$. We reach a contradiction.

Consider disjoint component streams $\{x_1, \ldots, x_M\}$, each x_i of rate δ, for some yet-to-be-determined δ and M. We assume that these M sub-streams are disjoint from another sub-stream θ of rate λ, all of which are components of the entire process $X(t)$. Let $U_{\theta+x_1+\ldots+x_M}$ represent the unfinished work in the system at some particular time t^*, with input processes

$\{\theta, x_1, \ldots, x_M\}$. From the definition of the blocking function, we have:

$$U_{\theta+x_1+\ldots+x_M} = U_{\theta+x_1+\ldots,+x_{M-1}} + U_{x_M} + \beta_{\theta+x_1+\ldots+x_{M-1},x_M} \tag{A.26}$$

By recursively iterating (A.26), we find:

$$U_{\theta+x_1+\ldots+x_M} = U_\theta + \sum_{i=1}^{M} U_{x_i} + \sum_{k=0}^{M-1} \beta_{\theta+x_1+\ldots+x_k,x_{k+1}} \tag{A.27}$$

Applying the monotonicity property of the blocking function to (A.27), we obtain:

$$U_{\theta+x_1+\ldots+x_M} \geq U_\theta + \sum_{i=1}^{M} [U_{x_i} + \beta_{\theta,x_i}] \tag{A.28}$$

Now applying the monotonically increasing, convex function $f(u)$ to both sides of (A.28) and writing $f(U_\theta + x) = f(U_\theta) + g(U_\theta, x)$ (from Fact 1), we have:

$$\begin{aligned} f(U_{\theta+x_1+\ldots+x_M}) &\geq f(U_\theta) + g\left(U_\theta, \sum_{i=1}^{m} [U_{x_i} + \beta_{\theta,x_i}]\right) & \text{(A.29)} \\ &\geq f(U_\theta) + \sum_{i=1}^{M} g\left(U_\theta, [U_{x_i} + \beta_{\theta,x_i}]\right) & \text{(A.30)} \end{aligned}$$

Inequality (A.30) holds by application of Fact 2, as $g(U, x)$ is a convex function of x that is zero at $x = 0$. Now notice that $\mathbb{E}f\left(U[\lambda+\delta]\right) - \mathbb{E}f\left(U[\lambda]\right) = \mathbb{E}f\left(U_\theta + x_i\right) - \mathbb{E}f\left(U_\theta\right) = \mathbb{E}\left\{g(U_\theta, U_{x_i} + \beta_{\theta,x_i})\right\}$ for any θ substream of rate λ, and any disjoint x_i substream of rate δ. Hence, by assumption:

$$\mathbb{E}\left\{g(U_\theta, U_{x_i} + \beta_{\theta,x_i})\right\} \geq \epsilon > 0 \tag{A.31}$$

Taking expectations of (A.30) and using (A.31), we find:

$$\mathbb{E}f\left(U_{\theta+x_1+\ldots+x_M}\right) \geq \mathbb{E}f\left(U_\theta\right) + M\epsilon \tag{A.32}$$

Inequality (A.32) above holds whenever $\theta + x_1 + \ldots + x_M$ is a substream of the entire, infinitely splittable process $X(t)$. We now choose M large enough so that $M\epsilon$ is greater than the expectation of $f(U)$ when the entire input $X(t)$ is applied, i.e., $M\epsilon > \mathbb{E}f\left(U_X\right)$. However, we choose a rate δ for each of the x_i substreams that is small enough so that we can still find M such substreams to ensure $\theta + x_1 + \ldots + x_M$ is still a component process of $X(t)$. This implies that, from (A.32), $\mathbb{E}f\left(U_{\theta+x_1+\ldots+x_M}\right) > \mathbb{E}f\left(U_X\right)$, which contradicts monotonicity. Hence, $\epsilon = 0$, (A.25) holds, and the theorem is proven. □

Appendix A.B — A simple counterexample for packet occupancy

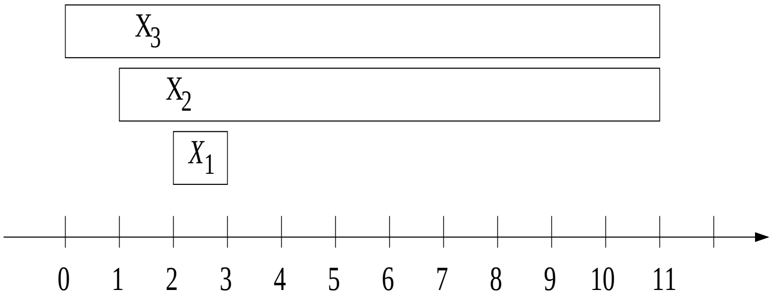


Figure A-10: A timing diagram with packets X_3, X_2, X_1 arriving at times $0, 1, 2$, respectively, illustrating that the occupancy blocking function $\alpha_{X_1,X_2}(t)$ does not have the monotonicity property when packets have variable lengths.

Here we show that under FIFO service with variable length packets, the monotonicity property of the packet occupancy blocking function does not hold, i.e., it is *not* true that $\alpha_{X_1+X_2,X_3}(t) \geq \alpha_{X_1,X_3}(t)$ for all time t.[8]

The counterexample, illustrated in Fig. A-10, is to consider streams X_1, X_2, X_3 consisting only of one packet each, where:

-The X_3 packet enters at time 0 with service time 11.

-The X_2 packet enters at time 1 with service time 10.

-The X_1 packet enters at time 2 with service time 1.

We look at time $t = 4$. At this time, we have: $N_{X_1}(4) = 0$. When X_1 and X_2 are combined, the X_2 packet blocks the X_1 packet from being served, hence $N_{X_1+X_2}(4) = 2$. Likewise, $N_{X_1+X_3}(4) = 2$, since the X_2 and X_3 packets are both long in comparison to the X_1 packet. However, because of this, when the X_3 packet is applied to a queue with the X_1 and X_2 packets, it will -The X_1 packet enters at time 2 with service time 1.

We look at time $t = 4$. At this time, we have: $N_{X_1}(4) = 0$. When X_1 and X_2 are combined, the X_2 packet blocks the X_1 packet from being served, hence $N_{X_1+X_2}(4) = 2$.

[8] Note that in cases of non-FIFO service, simple counterexamples can be constructed to show that the occupancy blocking function satisfies neither the monotonicity property nor the non-negativity property.

Likewise, $N_{X_1+X_3}(4) = 2$, since the X_2 and X_3 packets are both long in comparison to the X_1 packet. However, because of this, when the X_3 packet is applied to a queue with the X_1 and X_2 packets, it will not generate any extra packets due to blocking. Hence, $N_{X_1+X_2+X_3}(4) = 3$, and:

$$N_{X_1+X_2+X_3}(4) + N_{X_1}(4) = 3 < 4 = N_{X_1,X_2}(4) + N_{X_1,X_3}(4) \tag{A.33}$$

Thus, $\alpha_{X_1+X_2,X_3}(4) < \alpha_{X_1,X_3}(4)$, completing the example.

Appendix A.C — A Simple Counterexample for Load Balancing

Here we show that allocating exchangeable inputs to homogeneous queues according to the load balanced strategy does not necessarily stochastically minimize the total workload at every instant of time. That is, while it is true that the expected sum of moments is minimized, it is *not generally true* that for all values z and at every instant of time, the probability the total unfinished work in a load balanced system is greater than z is less than or equal to the corresponding probability in an unbalanced system.

Indeed, consider the following example of routing four streams to two queues with unit server rates. All streams deliver fixed length packets with unit packet lengths. Three of the streams have periodic packet arrivals with rates 0.2, while the fourth stream has periodic arrivals with rate 0.9. The first packet from all four streams arrives at time 0. The streams are randomly permuted so that it is impossible to know which is the high-rate stream, and hence the four streams are exchangeable.

The load balanced strategy is to allocate two streams to each queue, while the unbalanced strategy is to allocate three streams to the first queue and the remaining stream to the second queue. Under the balanced strategy, there must be one queue which is unstable, receiving periodic arrivals of total rate 1.1. It follows that the total unfinished work in this system is at least $0.1t$ at every instant of time. Thus, with probability 1, the unfinished work is greater than 5 whenever $t > 50$. However, the corresponding probability in the unbalanced system is at most 3/4. This is true because, in the case of the probability 1/4 event that the high-rate stream is placed alone in the second queue, the total number of packets in the first queue never exceeds three while the total number of packets in the second queue never exceeds one. However, one can directly verify that the expected workload in the balanced system is less than the expected workload in the unbalanced system at every instant of time, as proven for the general case by Lemma 24.

Appendix A.D — The occupancy blocking function

Here we prove the non-negativity, symmetry, and monotonicity properties of the packet occupancy blocking function $\alpha_{X_1,X_2}(t)$ when packets have fixed lengths L and service is non-preemptive work conserving. The general case of time varying processing speeds $\mu(t)$ is considered. We begin with a few simple lemmas comparing the systems of Fig. A-11.

D1. Consider the systems shown in Fig. A-11. All Systems A, B, and C have identical time-varying processing speeds $\mu(t)$. Two arbitrary input streams $X_1(t)$ and $X_2(t)$ together enter System A, and are applied individually to Systems B and C. We assume throughout that packets from the X_1 input have non-preemptive priority over X_2 packets, and that packets from the same stream are served in FIFO order. Define:

$$\begin{aligned} U_{X_1}^{A}(t) &= \text{Unfinished work in System } A \text{ due to } X_1 \text{ packets} \\ U_{X_2}^{A}(t) &= \text{Unfinished work in System } A \text{ due to } X_2 \text{ packets} \end{aligned}$$

Observation 1*:* $U_{X_1}^{A}(t) \geq U_{X_1}(t)$ for all time t.

Observation 2*:* $U_{X_2}^{A}(t) \geq U_{X_2}(t)$ for all time t.

Proof. (Observations 1 and 2) Note that $U_{X_1}^{A}(t)$ can be viewed as the unfinished work due to packets in a system with time varying processing rate $\mu(t)$ and a single input stream $X_1(t)$ with server vacations taking place at intervals corresponding to service intervals of X_2 packets in System A. This unfinished work is clearly less than the work $U_{X_1}(t)$ in a system with no vacations, and proves Observation 1. Observation 2 is proved similarly. □

We now consider individual packets p_1 and p_2 chosen arbitrarily from the X_1 and X_2 input streams of Fig. A-11, respectively. At any given time, we have:

Lemma D.1: If a packet p_1 from the X_1 stream is in System B, it is also in System A.

Lemma D.2: If a packet p_2 from the X_2 stream is in System C, it is also in System A.

Lemma D.3: There is at most 1 more packet from the X_1 stream in System A than in System B. In the case when there is 1 more, a packet from the X_2 stream must have been completely served during the current busy period of System A.

Proof. (Lemmas D.1 and D.2) To prove Lemma D.1, suppose packet p_1 enters from the X_1

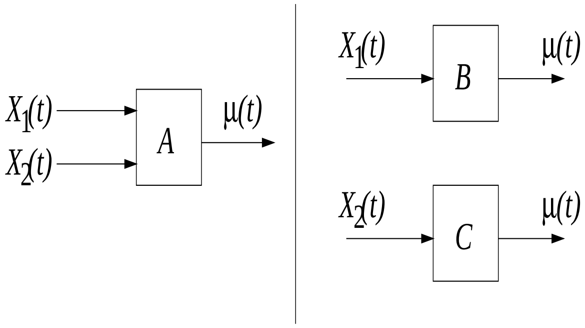


Figure A-11: Queueing systems A, B, and C with inputs $X_1(t)$ and $X_2(t)$ and time-varying processing rate $\mu(t)$.

stream at time t. In System A, this packet sees an amount of unfinished work $U^A_{X_1}(t^-) + R_{X_2}(t^-)$ in front of it (where R_{X_2} represents the residual service time if an X2 packet is in the server at time t). In System B, it sees $U_{X_1}(t^-)$. The packet exits System A at time $t+\tau_A$, and exits System B at time $t+\tau_B$, where τ_A and τ_B satisfy:

$$\begin{aligned}\int_t^{t+\tau_A} \mu(v)dv &= U^A_{X_1}(t^-) + R_{X_2}(t^-) + L \\ \int_t^{t+\tau_B} \mu(v)dv &= U_{X_1}(t^-) + L\end{aligned}$$

From Observation 1, we know $U^A_{X_1}(t^-) + R_{X_2}(t^-) \geq U_{X_1}(t^-)$, and hence $\tau_A \geq \tau_B$, proving Lemma D.1. Lemma D.2 is proved similarly. □

Proof. (Lemma D.3) Suppose there are more X_1 packets in System A than in System B at time t. Consider time t_1, the beginning of the current busy period in System A. If this busy period was completely composed X_1 packets, then System B must have an identical busy period and hence must have the same number of X_1 packets within it—which yields a contradiction.

Thus, the System A busy period must have contained an X_2 packet. Let t_2 be the time the latest X_2 packet started service in System A. Because of the priority scheme, no X_1 packets were in System A at this time (and hence—from Observation 1—System B must be empty at this time). It follows that System B can serve at most 1 more X_1 packet than System A during the interval $[t_1, t]$. Because System A has more X_1 packets at time t, it follows that it has exactly one more, that at least one departure from System B has occurred during $[t_1, t]$, and hence that the X_2 packet that started service at time t_1 in System A has already departed. $\square$

D2. Recall the packet occupancy blocking function is defined: $\alpha_{X_1,X_2}(t) \triangleq N_{X_1+X_2}(t) - N_{X_1}(t) - N_{X_2}(t)$

Theorem 25. *For time varying server speeds $\mu(t)$, if all packets have fixed lengths and service is non-preemptive and work conserving, the occupancy blocking function satisfies the non-negativity, symmetry, and monotonicity conditions, i.e., for all time t:*

(a) $\alpha_{X_1,X_2}(t) \geq 0$ (Non-negativity)

(b) $\alpha_{X_1,X_2}(t) = \alpha_{X_2,X_1}(t)$ (Symmetry)

(c) $\alpha_{X_1+X_2,X_3}(t) \geq \alpha_{X_1,X_3}(t)$ (Monotonicity)

Proof. Note that if packets are fixed in length, the number of packets in a queueing system is independent of the service order as long as service is non-preemptive and work conserving. Hence, without loss of generality, throughout we assume non-preemptive priority service with priority ordering $X_1 \rightarrow X_2 \rightarrow X_3$ (where X_1 packets receive the highest priority, etc.). To prove (a), it suffices to show that $N_{X_1+X_2}(t) \geq N_{X_1}(t) + N_{X_2}(t)$. This follows immediately from Lemmas $D.1$ and $D.2$. The symmetry property (b) is clear from the definition of $\alpha_{X_1,X_2}(t)$. Note that these conditions also hold for variable length packets when service is FIFO.

To prove the monotonicity property (c), it suffices to show that for all time t:

$$N_{X_1+X_2+X_3}(t) + N_{X_1}(t) \geq N_{X_1+X_2}(t) + N_{X_1+X_3}(t) \tag{A.34}$$

and hence that the combined number of packets in Systems A and B of Fig. A-12 below is greater than or equal to the number in Systems A' and B'. To show this, we have from Lemmas $D.1$ and $D.2$:

- If a packet p_1 from X_1 is in System A', it is also in System A
- If a packet p_2 from X_2 is in System A', it is also in System A.
- If a packet p_3 from X_3 is in System B', it is also in System A.

From Lemma $D.3$, there is at most one more packet from X_1 in System B' than in System B. If there are no more in System B' than in System B, we are done. Otherwise, let p_1 represent the single packet from the X_1 stream which is in System B' and is not in System B. By Lemma $D.1$, we know this packet is also in System A. If it is not in System A', we are done. Otherwise, packet p_1 is also in System A', and hence by Lemma $D.3$ we know there was a packet p_2 from X_2 that was served in the current busy period of System A', as well as a packet p_3 from X_3 that was served in the current busy period of System B'. Because both p_2 and p_3 precluded service of p_1 (in Systems A' and B', respectively), if either is served in System A, the server of A must thereafter serve X_1 packets, and hence the other must be currently contained in A. Thus, System A contains either p_2 or p_3—both of which are absent from Systems A' and B'. This extra packet in System A makes up for the 1 packet deficit in System B and hence preserves inequality (A.34). □

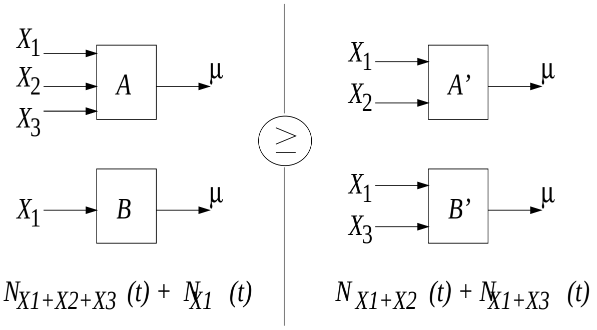


Figure A-12: A queueing illustration of the monotonicity property of the occupancy blocking function $\alpha_{X_1,X_2}(t)$ for fixed length packets.

Appendix A.E — Optimality of the Greedy Policy

Here we prove Lemma 24: Given a convex cost function $C(M_1, \ldots, k_N)$ of the form specified in Theorem 23, the optimal allocation vector can be obtained by sequentially adding streams, greedily choosing at each iteration the queue which increases the total cost $C()$ the least. This yields a cost-minimizing vector $(k_1^o, \ldots, k_N^o)$ after $M + N - 1$ evaluations/estimations of the cost function.

Proof. The lemma is clearly true for $M = 1$ stream. We assume that it is true for $M = k$ streams, and by induction prove it holds for $M = k + 1$.

Let $(k_1^o, \ldots, k_N^o)$ represent the optimal allocation vector for $M = k$ streams, which is obtained by the sequentially greedy algorithm. We thus have $\sum_i k_i^o = k$.

Now we add an additional stream in a greedy manner by placing it in the queue which increases the cost function the least. Without loss of generality, we assume this queue is queue 1, and we have a new allocation vector $(k_1^o + 1, \ldots, k_N^o)$. Suppose there is some other vector $(\tilde{k}_1^o, \ldots, \tilde{k}_N^o)$ whose elements sum to $k + 1$, such that $C(k_1^o + 1, \ldots, k_N^o) > C(\tilde{k}_1^o, \ldots, \tilde{k}_N^o)$.

Case 1 ($\tilde{k}_1 \geq \tilde{k}_1^o + 1$): In this case, we take away an input stream from the first entry of both vectors. This effects only the queue 1 term $\mathbb{E}f_1[k]$ in the cost function. Because this function is convex and non-decreasing, $\mathbb{E}f_1[k_1^o + 1]$ decreases by less than or equal to the amount that $\mathbb{E}f_1[\tilde{k}_1]$ decreases. Hence, it must be that $C(k_1^o, \ldots, k_N^o) > C(\tilde{k}_1^o - 1, \ldots, \tilde{k}_N^o)$, which contradicts the fact that the $(k_1^o, \ldots, k_N^o)$ vector is optimal for $M = k$ streams.

Case 2 ($\tilde{k}_1 < \tilde{k}_1^o + 1$): In this case, there exists some queue j such that $\tilde{k}_j > k_j^o$. We take the additional input we added to queue 1 and move it to queue j, forming a new vector $k_1^o, \ldots, k_j^o + 1, \ldots, k_N^o)$. Notice that this change cannot decrease the cost function, since this input was originally added greedily to queue 1. We now have $\tilde{k}_j \geq k_j^o + 1$, which reduces the problem to Case 1.

Thus, the sequentially greedy algorithm is optimal. It can be implemented with $M + N - 1$ evaluations of the cost function by keeping a record of the $N - 1$ queue increment values for the $N - 1$ queues not chosen at each step. □

Appendix B

Routing in Finite Buffer Queues

In this appendix, we consider the problem of routing packets from an arbitrary input stream $X(t)$ over a collection of heterogeneous queues in parallel. When the processing rates $(\mu_1, \ldots, \mu_N)$ of the queues are constant, a simple work conserving routing strategy π_{WC} is shown to hold total system backlog within a fixed upper bound from the resulting backlog of any other policy. Similar results apply to systems with time varying processing rates $(\mu_1(t), ..., \mu_N(t))$ when routing decisions can be postponed by placing packets in a pre-queue.

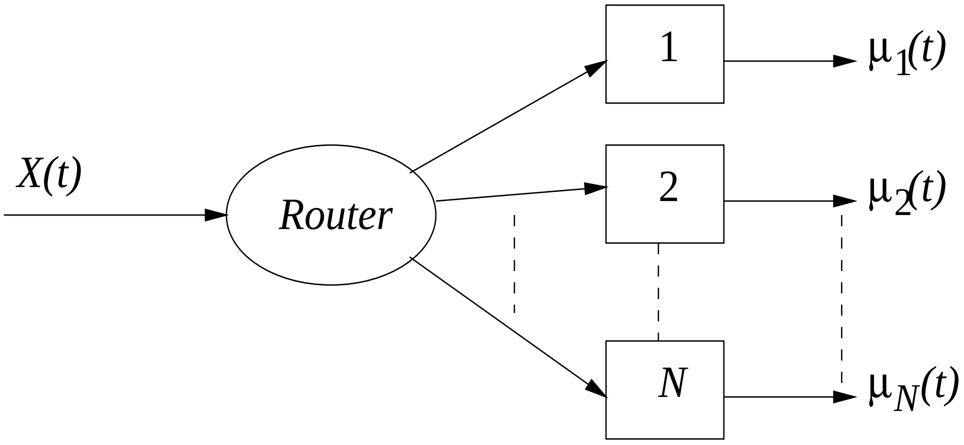


Figure B-1: Routing over a set of parallel queues with time-varying processing speeds $\mu_i(t)$.

For the case when routing decisions must be made immediately upon arrival, it is impossible to bound the number of packets or the amount of unprocessed bits in the queues unless complete knowledge of future events is known. However, we demonstrate that the simple non-predictive policy of routing packets to the shortest queue ensures system stability whenever possible. Of particular interest in this appendix is our treatment of finite buffer queueing analysis. We consider the *Join-the-Shortest-Queue* policy when all queues have finite buffers, and establish upper and lower bounds on packet drop rates. Stability properties are proven under the following new notion of stability: A finite buffer system is *stable* if the packet drop rate can be made arbitrarily small by increasing the buffer size.

We apply these results to a joint problem of routing and power allocation, where for each time varying channel state $S_i(t)$, the rate of each queue i can be varied by adjusting a power parameter P_i (subject to power constraints) according to a rate-power curve $\mu_i(S_i, P_i)$. A throughput maximizing algorithm is developed for this joint problem. This work supplements our analysis presented in Chapter 3. Indeed, in Chapter 3 we developed a power allocation scheme for a satellite downlink and proved stability of the *Join-the-Shortest-Queue* routing strategy using a Lyapunov function for infinite buffer systems. Our approach to the finite buffer problem uses a completely different technique.

Previous work on routing and queue control policies is found in [147] [45] [142] [60] [145] [146] [135] [17] [110] [108] [103]. In [147] an exact analysis of the *Join-the-Shortest-Queue* policy for parallel queues is developed for M/M/1 systems. In [45] the *Join-the-Shortest-Queue* strategy is shown to be optimal for minimizing average backlog in a system with an arbitrary packet arrival process entering a system of two queues with homogeneous exponential servers. An extension to the case when the two servers have different service rates is treated in [142], where an optimal threshold policy is developed using dynamic programming. Shortest queue optimality results similar to [45] are developed in [145] [146] for multiple homogeneous queues, and in [135] for homogeneous queues with finite buffers. Static or 'backlog-blind' routing policies such as round robin or probabilistic splitting are considered with various assumptions on the inputs in [45] [90] [89] [3] [25] [80] [56] [32] [14] [21] [130] [19] [22] [103]. In Appendix A, we develop convexity properties of queueing systems with general $*/*$ inputs and time varying server rates, and use them to establish an optimal static policy for routing multiple data streams over N parallel queues.

A related NP-complete problem of batch packet arrivals is considered in [52] [30] where the goal is to route packets to parallel queues to minimize the total delay for transmitting one batch. In [126] [9] [74] an online job scheduling problem is considered where N identical servers work on K jobs which arrive randomly over an interval of time $[0, T]$. Simple algorithms which finish within twice the minimum possible completion time are developed.

The main contribution in this paper is to treat stochastic queueing systems and to provide tight, worst case bounds on system performance for arbitrary input processes. Using sample path techniques, we develop additive bounds on the amount of unfinished work in the queues for the entire timeline $t \geq 0$. For finite buffer systems, we present additive bounds on packet drops. Such bounds directly translate into statements about queue stability and buffer requirements in the system. This approach differs significantly from the dynamic programming techniques which address systems with particular types of input and server processes.

In the next section we introduce the routing problem by comparing two natural routing strategies: a greedy routing strategy and a work conserving routing strategy. In Section B.2 a simple queueing inequality is developed for time-varying systems and is used as a tool for comparing routing schemes and proving performance bounds. In Sections B.3 and B.4 we treat routing in systems with a pre-queue and without a pre-queue, respectively, and in Section B.5 we apply these results to a power allocation problem. To provide the most general results while facilitating intuition, all buffers are assumed to be infinite until the finite buffer analysis is presented in Section B.4.

B.1 The Greedy and Work Conserving Algorithms

Consider the system of Fig. B-1 with an arbitrary arrival stream $X(t)$ sending packets to be routed over the N queues. Assume all processing rates $(\mu_1, \ldots, \mu_N)$ are constant. The goal is to route packets in a manner that ensures an acceptably low level of unfinished work $U(t)$ and number of packets $N(t)$ in the system for all time. It can be shown that a policy π_{greedy} (described below) is optimal in minimizing $N(t)$ at every instant of time if all packets have fixed lengths and arrive in a single burst at time zero. However, for general streams $X(t)$ with arrivals occurring over the timeline $t \in [0, \infty)$, it is not possible to minimize $N(t)$ or $U(t)$ at every instant of time—even if the entire future is known in advance. Here we seek a robust strategy, one whose performance at every instant of time t is sufficiently close to that of a system optimized to minimize backlog at that particular time instant.

Two natural routing strategies emerge, the greedy strategy π_{greedy} and the work conserving strategy π_{WC}:

1. The greedy strategy routes the current packet i to the queue that allows it to exit

first:

π_{greedy}: Choose queue k such that $k = \arg\min_{j\in\{1,\ldots,N\}} \left\{\frac{L_i+U_j(t)}{\mu_j}\right\}$

(where L_i is the length of the current packet i, and $U_j(t)$ is the unfinished work in queue j at time t).

2. The work conserving strategy routes to the queue which will empty first:

   π_{WC}: Choose queue k such that $k = \arg\min_{j\in\{1,\ldots,N\}} \left\{\frac{U_j(t)}{\mu_j}\right\}$

Notice that policies π_{greedy} and π_{WC} are identical if all server speeds μ_j are the same, although they may differ considerably under heterogeneous server rates. Because the greedy strategy uses the length of the current packet when making a routing decision, one would expect this policy to offer better performance. However, for suitable choices of the linespeeds $(\mu_1, \ldots, \mu_N)$, a system under the greedy strategy can have arbitrarily more unfinished work within it than the same system operating under the work conserving strategy, as the following example illustrates.

Suppose a burst of B packets arrive to the system at time 0. After this initial burst, a single packet arrives periodically at times $\{1, 2, 3, \ldots\}$. Assume all packets have fixed lengths $L = 1$, and that $(\mu_1, \mu_2, \ldots, \mu_N) = (1, \epsilon, ..., \epsilon)$. Suppose that $\epsilon > 0$ is sufficiently small to ensure that all packets from the initial burst as well as all packets thereafter are routed to queue 1 under strategy π_{greedy}. Thus, under the greedy strategy, there are always B packets in the system.

Now consider a different set of routing decisions (which we represent as policy π): route the B packets from the initial burst amongst the $N-1$ queues of rate ϵ, and route all packets from the periodic stream thereafter to queue 1. Under this policy, queue 1 always has exactly one packet within it. However, the B packets from the initial burst eventually depart from queues $\{2, 3, \ldots, N\}$, leaving these queues empty after some finite time T. Hence, after time T the greedy strategy π_{greedy} results in $B-1$ *more packets* in the system than policy π, where $B-1$ can be made arbitrarily large. Alternatively, in Section B.2 it is shown that the work conserving strategy π_{WC} is fundamental in that it never produces more than $N-1$ extra packets in the system compared to any other policy.

B.2 A Multiplexing Inequality

Here we develop a queueing inequality useful for establishing performance bounds on routing policies. Let $X(t)$ represent an arbitrary packet arrival process on the interval $[0, \infty)$. A particular $X(t)$ sample path is a non-decreasing staircase function representing the total number of bits that have arrived to the system during $[0, t]$. Jumps in the $X(t)$ function occur at packet arrival epochs and have magnitudes equal to the length of the arriving packet. We assume there is a maximum packet length L_{max}.

Consider the single server and multi-server queueing systems of Fig. B-2. The linespeed processes $\mu(t)$ and $\{\mu_i(t)\}$ represent instantaneous server processing rates (in units of bits per second). Here we assume that the rate of the single server queue of Fig. B-2a is equal to the sum of the individual server rates in Fig. B-2b, i.e., $\mu(t) = \mu_1(t) + \ldots + \mu_N(t)$. Assume both systems of Fig. B-2 are initially empty and the same input process $X(t)$ is applied to each. Packets are immediately processed in the single server system according to a non-idling service policy. However, in the multi-server system packets are routed to the N queues using any conceivable routing mechanism. All buffers in the queues and in the router device are assumed to be infinite, so that no packets are lost. Let $U_{single-server}(t)$ represent the unfinished work (in bits) in the single server queue, and let $U_{multi-server}(t)$ represent the total amount of unfinished bits in the multi-queue system.

Lemma 25. *(Multiplexing Inequality): For all time $t \geq 0$:*

$$U_{single-server}(t) \leq U_{multi-server}(t) \tag{B.1}$$

Proof. Let $D_{single-server}(t)$ and $D_{multi-server}(t)$ represent the amount of processed bits or "departures" from the single server system and multi-server system, respectively, during $[0, t]$. Clearly we have $D_{single-server}(t) = X(t) - U_{single-server}(t)$ and $D_{multi-server}(t) = X(t) - U_{multi-server}(t)$, and it suffices to show that $D_{single-server}(t) \geq D_{multi-server}(t)$. Notice that the departure functions are both initially 0 at time $t = 0$. Whenever the single-server system is empty, $D_{single-server}(t) = X(t) \geq D_{multi-server}(t)$, and hence the inequality is satisfied at such times. If the single-server queue is not empty, then it is processing packets at the instantaneous rate $\mu(t)$, which is greater than or equal to the instantaneous departure rate of bits from the multi-server system. Hence, the departures from the multi-server

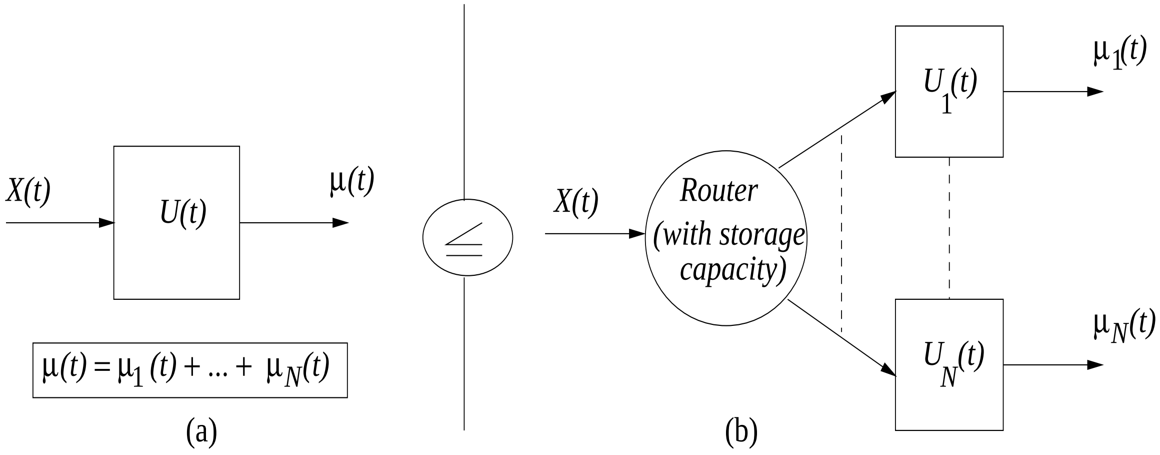


Figure B-2: A single server queue and a set of multi-server queues with an arbitrary router device. The sum of the time varying server rates of the servers in (b) equals the rate $\mu(t)$ in (a).

system can never overtake the departures from the single-server queue. □

A finite buffer version of this statement is given in Section B.4. This multiplexing inequality demonstrates that it is always better to multiplex data streams from individual queues to a single queue whose rate is equal to the sum of the individual processing rates. It is useful to consider such a virtual single-server queue to provide a baseline for measuring the performance of routing policies. Strategies yielding unfinished work functions close to the $U_{single-server}(t)$ lower bound are desirable.

Consider now a particular work conserving routing strategy π_{WC} that operates on the multi-queue system of Fig. B-2b. The policy π_{WC} places all incoming packets in a shared buffer device or "pre-queue." Whenever any server becomes available, the pre-queue instantaneously routes the next packet to that server. If there is more than one server available, the choice is made arbitrarily. Thus, the policy π_{WC} is "work conserving" in that no servers are idle whenever there are buffered packets waiting to be processed. Let $U_{WC}(t)$ represent the total unfinished work at time t in the multi-server system of Fig. B-2b under this policy.

Lemma 26. *(Performance Tracking): At every instant of time t:*

$$U_{single-server}(t) \leq U_{WC}(t) \leq U_{single-server}(t) + (N-1)L_{max} \tag{B.2}$$

Proof. The first inequality is just a particular case of the multiplexing inequality (Lemma 25). To prove the second inequality, we compare the departed bits $D_{WC}(t)$ and $D_{single-server}(t)$.

It suffices to show that $D_{single-server}(t) \leq D_{WC}(t) + (N-1)L_{max}$. For simplicity, assume that $\mu_i(t) < \infty$ for all time t and all i, so that departures drain continuously from the queues. For the above departure inequality to be violated, there must be some crossing time t^* such that $D_{single-server}(t^*) = D_{WC}(t^*) + (N-1)L_{max}$. If the single-server system is empty at this time, the departure function $D_{single-server}(t)$ cannot increase, and hence t^* cannot be a crossing time. Otherwise, the multi-server system holds strictly more than $(N-1)L_{max}$ bits, and hence contains at least N distinct packets. By the nature of the work conserving policy π_{WC}, all servers of the multi-server system must be actively processing these packets. The departure functions $D_{single-server}(t)$ and $D_{WC}(t)$ are thus increasing at the same rate at time t^*, and the departures from the single-server system cannot overtake the bound. □

Notice that the bounds of Lemma 26 imply that the work conserving routing strategy π_{WC} is stable (in the usual sense of infinite buffer systems) if and only if the single queue system with the same inputs is stable. Stability issues for finite buffers systems are addressed further in Section B.4.

B.3 Systems with Pre-Queues

The results of the previous section allow routing policy π_{WC}—which is implemented with a pre-queue—to be compared to any other policy π which operates on the same multi-server system. Here we show π_{WC} is *minimax optimal.* Let $U_{WC}(t)$, $N_{WC}(t)$, $U_{\pi}(t)$, and $N_{\pi}(t)$ represent the unfinished work and number of packets in the multi-queue system of Fig. B-2b under policies π_{WC} and some other (arbitrary) policy π, respectively.

B.3.1 Variable Length Packets

Here we treat systems with variable length packets. All packets are bounded by a maximum packet length L_{max}. We show that the π_{WC} routing strategy provides the best worst-case performance guarantee of all policies for routing packets non-preemptively over multiple time varying servers.

A policy π is *non-preemptive* if it does not interrupt a packet that has initiated processing at a server. If the policy does not require knowledge of the future, we say it is *non-predictive.* If a policy π has full knowledge of future events, it can be designed to minimize unfinished work at some particular time instant τ. Let $\Phi_{X,\vec{\mu}}(\tau)$ represent this minimum value of unfinished work in a system with input stream $X(t)$ and linespeeds $\vec{\mu}(t) = (\mu_1(t), \ldots, \mu_N(t))$.

Consider a game where a scheduler makes routing decisions (according to some non-predictive policy π) while an adversary dynamically creates an input stream $X(t)$ and varies the linespeeds $(\mu_1(t), \ldots, \mu_N(t))$ in order to maximize the difference between $U_\pi(t)$ and $\Phi_{X,\vec{\mu}}(\tau)$. The goal of the scheduler is to minimize the worst case deviation from optimality, i.e., minimize the value of $\max_{t\geq 0}\left\{U_\pi(t) - \Phi_{X,\vec{\mu}}(t)\right\}$.

Theorem 26. *(a) For all policies π (possibly predictive and preemptive), we have:*

$$U_{WC}(t) \leq U_\pi(t) + (N-1)L_{max} \tag{B.3}$$

(b) The work conserving policy π_{WC} is the minimax optimal non-predictive, non-preemptive routing strategy over the set of all possible inputs and linespeed variations.

Proof. Part (a) follows immediately from Lemmas 25 and 26. To establish that the policy π_{WC} is minimax optimal, it suffices to show that any non-predictive, non-preemptive policy π can be forced to meet the $(N-1)L_{max}$ bound. The idea is for an adversary to force policy π to route maximum length packets to distinct servers, and then to trap these packets by setting their server rates to 0. Specifically, the adversary sends $(N-1)$ maximum length packets at time 0. The adversary then maintains a constant output rate of $\mu_i(t) = \mu$ for all servers i that have no packets within them. Whenever a packet is routed to a server under policy π, that server rate is set to 0. After $(N-1)L_{max}/\mu$ seconds have elapsed, no unfinished work has been processed, and there must be some server j^* that has remained empty for the full time interval, with an unused processing rate $\mu_j(t) = \mu$. Hence, given this particular sample path of server rates and packet arrivals, an alternative routing scheme that sends all packets to server j^* would have allowed the system to be empty at this time, and so the $(N-1)L_{max}$ bound is met with equality. □

B.3.2 Fixed Length Packets

In the case when all packets have fixed lengths, a version of inequality (B.3) can be established in terms of the number of packets $N(t)$ in the system.

Theorem 27. *If all packets have fixed lengths L, then:*

$$N_{WC}(t) \leq N_{\pi}(t) + N - 1 \tag{B.4}$$

Proof. Define a *completely busy period* as an interval of time when all servers of the parallel queue system are busy. Because π_{WC} never buffers packets if a server is idle, the inequality holds whenever t is not within a completely busy period. Suppose now that t lies within a completely busy period, and let τ_B be the beginning time of this period. We have:

$$\begin{aligned} N_{WC}(t) &= N_{WC}(\tau_B^-) + A(\tau_B, t) - D_{WC}(\tau_B, t) & \text{(B.5)} \\ N_{\pi}(t) &= N_{\pi}(\tau_B^-) + A(\tau_B, t) - D_{\pi}(\tau_B, t) & \text{(B.6)} \end{aligned}$$

where τ_B^- represents the time just before the arrival initiating the completely busy period, $A(\tau_B, t)$ is the number of arrivals during the interval $[\tau_B, t]$, and $D_{WC}(\tau_B, t)$, $D_{\pi}(\tau_B, t)$ respectively represent the number of packet departures from the π_{WC} system and the π system during this interval. The above departure functions are composed of individual terms D_{WC}^i and D_{π}^i representing departures from queue i:

$$D_{WC}(\tau_B, t) = \sum_{i=1}^{N} D_{WC}^i \ , \ D_{\pi}(\tau_B, t) = \sum_{i=1}^{N} D_{\pi}^i \tag{B.7}$$

During the time interval $[\tau_B, t]$, each queue of the π_{WC} system continuously processes fixed length packets. Thus, $D_{WC}^i \geq D_{\pi}^i - 1$ for all queues i, and equality holds only if there was a packet being processed by queue i under the π routing policy at time τ_B^-. Suppose there are k such cases, so that $N_{\pi}(\tau_B^-) = k$ (where $k \leq N$). Thus, $D_{WC}(\tau_B, t) \geq D_{\pi}(\tau_B, t) - k$, and we have:

$$\begin{aligned} N_{WC}(t) &\leq N_{WC}(\tau_B^-) + A(\tau_B, t) - D_{\pi}(\tau_B, t) + k & \text{(B.8)} \\ &= N_{WC}(\tau_B^-) + N_{\pi}(t) - N_{\pi}(\tau_B^-) + k & \text{(B.9)} \\ &\leq (N-1) + N_{\pi}(t) - k + k & \text{(B.10)} \end{aligned}$$

where (B.9) follows from (B.6), and (B.10) follows because the work conserving strategy contains fewer than N packets just before the completely busy period. □

A technique similar to the one used in the proof of Theorem 26 shows that this $(N-1)$ bound is tight and is minimax optimal over all non-predictive, non-preemptive strategies.

Similar routing and scheduling problems have been treated in [45] [142] [60] [144] for systems with two heterogeneous servers ($N = 2$) with memoryless assumptions on the server or arrival processes. With such a formulation applied to the problem of routing Poisson inputs with fixed-length packets in a two-queue system, it is possible to prove that the optimal routing strategy for minimizing expected occupancy in the system has a threshold structure. A complex dynamic program can be developed to numerically compute the exact threshold function. However, here we find that—with arbitrary input processes $X(t)$—the simple work conserving strategy π_{WC} ensures no more than one extra packet in the system compared to any other strategy at any time.

B.4 Systems Without a Pre-Queue

B.4.1 Constant Rate Servers

The implementation of the work conserving policy π_{WC} for time varying servers uses a pre-queue to store packets until the next server becomes available. In many systems it is undesirable or even impossible to implement a pre-queue. For example, in a satellite network, queues might be aboard different satellites, which may require routing decisions to be made immediately upon packet arrival. Here we show that the same results can be obtained in systems without a pre-queue if the server rates $(\mu_1, \ldots, \mu_N)$ are known constants.

Observation: For constant server rates $(\mu_1, \ldots, \mu_N)$, the strategy of routing an incoming packet to the queue k with the smallest value of $U_k(t)/\mu_k$ (the π_{WC} strategy as described in Section B.1) is the same as the π_{WC} strategy described for time varying servers in Section B.2. Thus, a pre-queue is not needed.

Proof. The strategy always routes a new packet to the queue that will empty first. Thus, if there is ever an empty server i, there can be no more than 1 packet in each of the $(N-1)$ other queues. □

Thus, the bounds in Theorems 26 and 27 apply to heterogeneous, constant rate servers when this routing method is used.

B.4.2 Time Varying Servers

Consider routing over a multi-queue system with time-varying processing rates when no pre-queue is available and all routing decisions are made immediately upon packet arrival. We examine the *Join-the-Shortest-Queue* (*JSQ*) policy: route the current packet to the queue i with the smallest value of $U_i(t)$. Intuitively, this strategy is the closest match to the work conserving strategy given that we cannot predict future values of server rates.

We seek to prove that this strategy stabilizes the multi-queue system whenever it is stabilizable. This is done for general ergodic input sources and linespeed processes by introducing a new notion of stability defined in terms of finite buffer systems. Consider the single queue system of Fig. B-3a with an ergodic input process $X(t)$ of bit rate λ, a linespeed process $\mu(t)$ with ergodic rate μ_{av}, and assume this system has a finite buffer capacity of M bits. We assume that a full packet of size L is dropped if it arrives when $M - U(t) < L$. Let $G^M(t)$ represent the total bits dropped (or placed into the *Garbage*) during $[0, t]$ when the buffer size is M. Further let $DR(M)$ represent the *drop rate* (measured in bits) of the system as a function of buffer space M:

$$DR(M) = \limsup_{t\to\infty} \frac{G^M(t)}{t}$$

Definition 13. *A system is* loss rate stable *if the drop rate can be made arbitrarily small by increasing buffer capacity, i.e., if $DR(M) \to 0$ as $M \to \infty$.*

For the remainder of this appendix, we consider only finite buffer systems, and use the term *stable* to mean *loss-rate-stable.*

It can be shown that a necessary condition for loss rate stability is $\lambda \leq \mu_{av}$. Furthermore, if the input stream and server rate processes evolve according to an underlying finite state ergodic Markov chain, a sufficient condition for loss rate stability is $\lambda < \mu_{av}$. This notion of stability is closely tied to the standard notion defined in terms of a vanishing complementary occupancy distribution for infinite buffer capacity queues [98] [6] [81] [88] [132] [95].

Before analyzing the *JSQ* strategy, we present a finite buffer version of the multiplexing inequality. Consider the two systems of Fig. B-3. The server rates $\mu_1(t), \ldots, \mu_N(t)$ of the multi-queue system sum to the single-server linespeed $\mu(t)$. Suppose the single queue system has buffer size M, while the multi-queue system has buffer sizes $M_1, \ldots, M_N$. Let $G_{single}(t)$ represent the total bits dropped by the single server system during $[0, t]$, and let

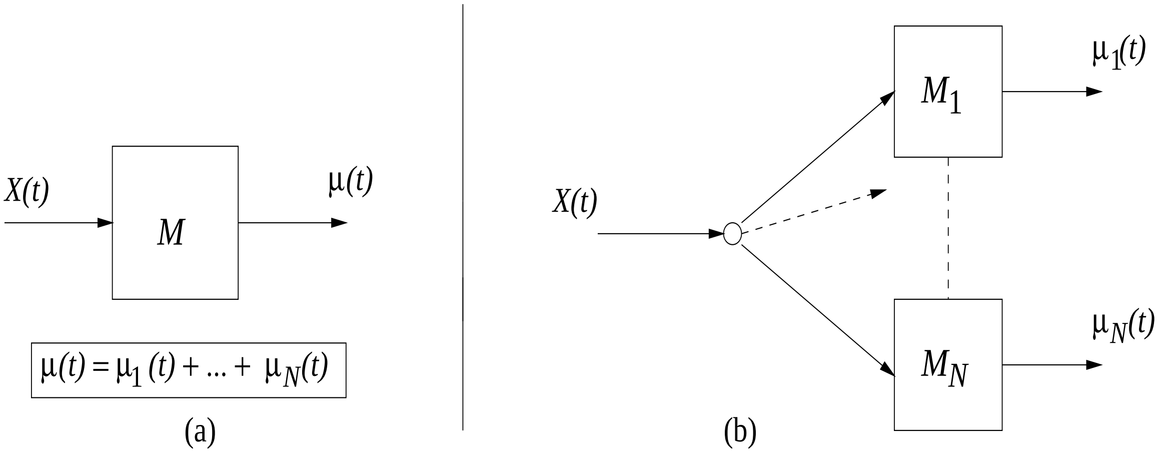


Figure B-3: A single stage system with a finite buffer of size M and an aggregated server processing rate $\mu(t)$ compared to a multi-queue system with finite buffers.

$G_{multi}(t)$ represent the bits dropped by the multi-server system with the same inputs (using any arbitrary set of routing decisions). Both systems are assumed empty at time 0.

Theorem 28. *(Finite Buffer Multiplexing Inequality): For arbitrary inputs $X(t)$, if $M \geq M_1 + \ldots + M_N + L_{max}$, then for all t:*

(a) Departures satisfy: $D_{single}(t) \geq D_{multi}(t)$

(b) Packet drops satisfy: $G_{single}(t) \leq G_{multi}(t)$

Proof. See Appendix B.A. □

Thus, a single-queue system in which all buffer slots and linespeeds are aggregated (with an additional buffer slot of size L_{max}) always outperforms the multi-queue system. We now consider the particular routing strategy *JSQ*. Let $DR_{JSQ}(M)$ represent the drop rate of the multi-queue system (operating under the *JSQ* policy) when all queues have finite buffer storage M.

Theorem 29. *For all buffer sizes M:*

$$DR_{JSQ}(M + NL_{max}) \leq DR_{single-queue}(M) \qquad \text{(B.11)}$$

and hence a multi-queue system under the JSQ *strategy with a buffer size of $M + NL_{max}$ in each queue drops fewer packets than the single queue with buffer size M.*

Proof. : We prove a stronger result: $G_{JSQ}(t) \leq G_{single}(t)$ for all $t \geq 0$, where $G_{JSQ}(t)$ and $G_{single}(t)$ respectively represent the total bits dropped by the multi-queue system (under

JSQ) and the single queue system. Suppose this inequality is first violated at some time τ (we reach a contradiction). It follows that an arriving packet must have been dropped by the *JSQ* system at this time, and thus all servers of the multi-queue system are busy. Let t_B represent the start of this completely busy period, so that bits depart from the *JSQ* system at the full service rate during $[t_B, \tau]$. Let $U_{JSQ}(t)$ represent the unfinished work in the *JSQ* system at time t. Further define:

$$\begin{aligned} a &\triangleq \text{Arrivals during } [t_B, \tau] \\ d_{JSQ} &\triangleq \text{Bits processed by the } JSQ \text{ system during } [t_B, \tau] \\ g_{JSQ} &\triangleq \text{Bits dropped by the } JSQ \text{ system during } [t_B, \tau] \end{aligned}$$

Define d_{single}, g_{single}, and $U_{single}(t)$ similarly for the single queue system. Note that $g_{JSQ} > g_{single}$ because the packet drop inequality $G_{JSQ}(t) \leq G_{single}(t)$ is first violated at time τ. The following bit-conservation equalities hold:

$$U_{JSQ}(\tau) = U_{JSQ}(t_B^-) + a - d_{JSQ} - g_{JSQ} \tag{B.12}$$

$$U_{single}(\tau) = U_{single}(t_B^-) + a - d_{single} - g_{single} \tag{B.13}$$

Just before the completely busy period, at least one queue of the multi-server system is empty, and hence:

$$U_{JSQ}(t_B^-) \leq (N-1)\left[M + NL_{max}\right] \tag{B.14}$$

Because a packet is dropped by the *JSQ* system at time τ, all queues must have more than $[M + (N-1)L_{max}]$ unfinished work within them, and hence:

$$U_{JSQ}(\tau) > N\left[M + (N-1)L_{max}\right] \tag{B.15}$$

Using (B.14) and (B.15) in (B.12), we have:

$$N\left[M + (N-1)L_{max}\right] < (N-1)\left[M + NL_{max}\right] + a - d_{single} - g_{single}$$

and hence

$$a - d_{JSQ} > M + g_{JSQ} \tag{B.16}$$

The unfinished work in the single queue can thus be bounded:

$$\begin{aligned}
U_{single}(\tau) &\geq a - d_{single} - g_{single} && \text{(B.17)}\\
&\geq a - d_{JSQ} - g_{single} && \text{(B.18)}\\
&> M + g_{JSQ} - g_{single} && \text{(B.19)}
\end{aligned}$$

where (B.17) follows from (B.13), (B.18) follows because the *JSQ* system processes packets at the full rate $\mu(t)$ during $[t_B, \tau]$, and (B.19) follows from (B.16). Now, because of the finite buffer constraint, $M \geq U_{single}(\tau)$, and hence (B.19) yields $g_{JSQ} < g_{single}$, a contradiction. □

Theorems 28 and 29 imply that

$$DR_{single}(MN + (N^2+1)L_{max}) \leq DR_{JSQ}(M + NL_{max}) \leq DR_{single}(M)$$

and hence the multi-queue system under the *JSQ* routing strategy is stable if and only if the corresponding single queue system is stable. Furthermore, (B.11) provides a simple and useful bound on the packet drop rate in the multi-queue system in terms of a single queue with a finite buffer. Inequality (B.11) can be used together with the finite buffer multiplexing inequality to bound the performance of *JSQ* in terms of any other routing policy π:

$$DR_{JSQ}(M) \leq DR_{\pi}\left(\frac{M}{N} - L_{max}\frac{N+1}{N}\right)$$

where $DR_{\pi}(V)$ represents the drop rate in the multi-queue system with the same input stream but with any arbitrary (possibly anticipatory) routing policy π, for the case when all queues have buffer size V.

B.5 Joint Routing and Power Allocation

Suppose that the transmission rates $(\mu_1, \ldots, \mu_N)$ of the system can be controlled by adjusting power levels P_i allocated to each server. Specifically, suppose that each channel $i \in \{1, 2, \ldots, N\}$ has an associated channel state S_i which takes values on a finite set of states ($S_i \in \{C_1^i, C_2^i, \ldots, C_{M_i}^i\}$). Let $\mu_i(P_i, S_i)$ represent a concave rate-power curve for each channel state (see Fig. B-4). We assume that the individual queues belong to a collection of J sub-units, and let V_j represent the set of queues $i \in \{1, \ldots, N\}$ belonging to sub-unit j. The sub-units could represent distinct satellites of a satellite network, or different bases-tations in a wireless network. Each sub-unit j has its own power resource with total power $P_{tot}^{(j)}$.

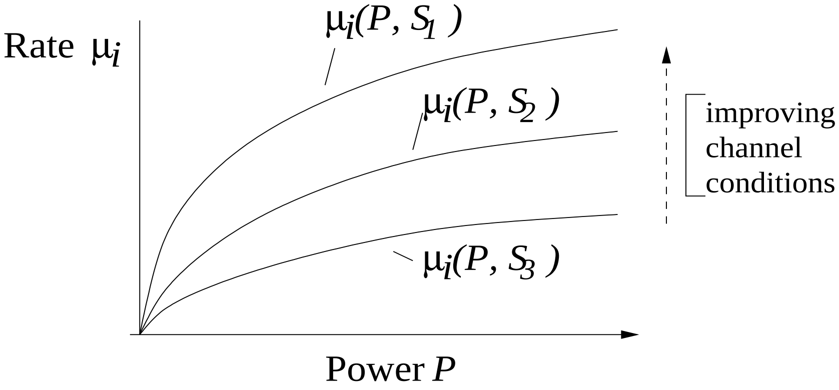


Figure B-4: A set of concave power curves $\mu_i(P_i, S)$ for channel states S_1^i, S_2^i, S_3^i.

Let $\vec{S}(t) = (S_1(t), S_2(t), \ldots, S_N(t))$ represent the vector of channel states at time t, and assume that $\vec{S}(t)$ varies according to a discrete time Markov chain with timeslots of length T. As before, packets enter the system according to an input process $X(t)$ and routing decisions are made immediately upon arrival. In addition to making routing decisions, a controller must choose a power allocation $\vec{P}(t) = (P_1(t), \ldots, P_N(t))$ for each instant of time, subject to the total power constraints of each sub-unit: $\sum_{i \in V_j} P_i(t) \leq P_{tot}^{(j)}$ for all $j \in \{1, \ldots, J\}$. The problem is to design a joint routing and power allocation strategy that maximizes throughput and stabilizes the system whenever the system is stabilizable. Such a policy makes decisions using the observable system state vectors $\vec{U}(t)$ and $\vec{S}(t)$.

In general, both state vectors $\vec{U}(t)$ and $\vec{S}(t)$ are important in both the routing and power allocation decisions. For example, clearly any power allocated to an empty queue is wasted and should be re-allocated to improve processing rates amongst the non-empty queues. Likewise, a router is inclined to place packets in faster queues, especially if the rates of those queues are guaranteed to operate at high levels for one or more timeslots. However, below we show that the routing and power allocation problem can be decoupled into two policies: a routing policy which considers only $\vec{U}(t)$, and a power allocation policy which considers only $\vec{S}(t)$. The power allocation policy is distributed, so that each sub-unit makes independent control decisions using only the local channel state information for each queue it contains. The resulting strategy maximizes total system throughput even when the underlying Markov chain describing $\vec{S}(t)$ is unknown.

Let $\beta_{\vec{S}}$ represent the steady-state probability that the channel vector is in state $\vec{S}$. Define the following rate $\overline{\mu}$:

$$\overline{\mu} = \sum_{\vec{S}} \beta_{\vec{S}} \left[\max_{\sum_{i \in V_j} P_i = P_{tot}^{(j)} \, \forall j} \sum_i \mu_i(P_i, S_i) \right] \tag{B.20}$$

The value of $\overline{\mu}$ is the average total rate offered by the system when power is allocated to maximize total rate at every instant of time. Assume that the input process $X(t)$ generates packets according to a fine state, ergodic Markov chain, and let λ represent the total bit rate.

Theorem 30. *The capacity of the multi-queue system with joint routing and power allocation is $\overline{\mu}$, i.e., a necessary condition for stability is $\lambda \leq \overline{\mu}$, and a sufficient condition is $\lambda < \overline{\mu}$.*

The fact that $\lambda \leq \overline{\mu}$ is necessary follows by application of Theorems 28 and 29. Indeed, suppose a stabilizing algorithm exists, and let $\{p_i(t)\}$ represent the stabilizing power functions. (Note that these functions are not necessarily ergodic). The sum rate of all servers in the multi-queue system is hence $\mu(t) = \mu_1(P_1(t), S_1(t)) + \ldots + \mu_N(P_N(t), S_N(t))$. By Theorem 28, the system is stable only if a single queue with the same inputs and server rate $\mu(t)$ is stable. But $\mu(t) \leq \mu^*(t)$ for all t, where $\mu^*(t)$ is the result when power is allocated to maximize instantaneous sum rate. Thus, a system with input $X(t)$ and server process $\mu^*(t)$ is also stable. But $X(t)$ is ergodic with rate λ and $\mu^*(t)$ is ergodic with rate $\overline{\mu}$, so $\lambda \leq \overline{\mu}$ must hold.

Sufficiency is established by design of the following decoupled policy π^* which stabilizes the system whenever $\lambda < \overline{\mu}$:

Power Allocation: At every new timeslot, each sub-unit j observes the entries of the channel state vector $\vec{S}(t)$ corresponding to the queues it contains (given by the set V_j). The sub-unit then allocates power $\{P_i(t)\}$ (for $i \in V_j$) to maximize $\sum_{i \in V_j} \mu_i(P_i, S_i(t))$ subject to $\sum_{i \in V_j} P_i = P_{tot}^{(j)}$. This power allocation is held constant until the next timeslot.

Routing: Whenever a new packet enters the system, we observe the value of $\vec{U}(t) = (U_1(t), \ldots, U_N(t))$ and route to the shortest queue.

Note that this strategy is simply an application of *JSQ* routing in reaction to the rates determined by the power allocation decisions. Thus, from Theorem 29 we know that the multi-queue system is stable whenever the single queue system is stable, which is ensured when $\lambda < \overline{\mu}$. This establishes Theorem 30.

B.6 Summary

The problem of routing packets from an arbitrary stream over a set of parallel queues has been considered in the context of constant and time-varying processing rates. Using sample path analysis, a simple work conserving strategy was developed to provide fundamental performance bounds on the unfinished work in the system at any instant of time. In time varying systems, this strategy can be implemented with a pre-queue and guarantees that performance closely follows the performance of a superior single-queue system with an aggregated data rate.

The pre-queue was shown to be unnecessary when service rates are constant. In the case of time-varying rates, removing the pre-queue precludes the design of a routing strategy which meets the tight performance bounds on unfinished work guaranteed by a work conserving policy. However, a general stability result was established for the *Join-the-Shortest-Queue* policy, and the result was extended to treat a joint problem of routing and power allocation. This analysis was performed using a new and useful notion of stability defined in terms of finite buffer systems. Performance bounds for the *JSQ* strategy were given by showing that if all queues have buffer size $M + NL_{max}$, the drop rate is less than or equal to the drop rate of a single queue with an aggregate rate and buffer size M.

Our approach differs significantly from other approaches in that we provide tight, worst case bounds on system performance with arbitrary input and linespeed processes, rather than analyzing systems with particular stochastic inputs and linespeeds. We believe this approach can be applied to more complex queueing structures in satellite and wireless networks to provide performance bounds and stability guarantees for systems with very general input processes and control laws.

Appendix B.A — Proof of Finite Buffer Multiplexing Inequality

Here we prove the finite buffer multiplexing inequality (Theorem 28), which compares a single queue with a time varying server rate $\mu(t)$ and a buffer of size M to a parallel system of N queues with server rates $\mu_1(t), \mu_2(t), \ldots, \mu_N(t)$ with buffer sizes $M_1, M_2, \ldots, M_N$.

Theorem: If $\sum_i \mu_i(t) = \mu(t)$ and if $L_{max} + \sum_i M_i \leq M$, then for an arbitrary input stream $X(t)$:

(a) Departures satisfy: $D_{single}(t) \geq D_{multi}(t)$ for all t

(b) Packet drops satisfy: $G_{single}(t) \leq G_{multi}(t)$ for all t

Proof. The single-queue and multi-queue systems satisfy the following bit conservation equalities for all time:

$$U_{single}(t) = X(t) - D_{single}(t) - G_{single}(t) \tag{B.21}$$

$$U_{multi}(t) = X(t) - D_{multi}(t) - G_{multi}(t) \tag{B.22}$$

Claim 1: If $D_{single}(t) \geq D_{multi}(t)$ for all $t \in [0, t^*]$ for some time t^*, then $G_{single}(t) \leq G_{multi}(t)$ on the same interval.

Pf: It suffices to check times t when the single-server system loses a packet, and the multi-server system retains that same packet (otherwise, $G_{single}(t) - G_{multi}(t)$ cannot increase). At such times, $U_{single}(t) > M - L_{max}$, and $U_{multi} \leq M_1 + M_2 + \ldots + M_N \leq M - L_{max}$. Furthermore, we have:

$$\begin{aligned} G_{single}(t) &= X(t) - D_{single}(t) - U_{single}(t) \\ &\leq X(t) - D_{multi}(t) - U_{single}(t) \\ &< X(t) - D_{multi}(t) - (M - L_{max}) \\ &= U_{multi}(t) + G_{multi}(t) - (M - L_{max}) \\ &\leq G_{multi}(t) \end{aligned}$$

which proves the claim. □

Claim 2: $D_{single}(t) \geq D_{multi}(t)$ for all time $t \geq 0$.

Pf: For simplicity, assume $\mu(t) < \infty$ so that packets drain continuously from the queues.

The departure inequality is true at time 0. If it is ever violated, there must be some first crossing time t^* where $D_{multi}(t^*) - D_{single}(t^*)$. At such a time, from (B.21) and (B.22) we have:

$$U_{multi}(t^*) + G_{multi}(t^*) = U_{single}(t^*) + G_{single}(t^*)$$

However, from Claim 1, we know $G_{single}(t^*) \leq G_{multi}(t^*)$, and hence $U_{multi}(t^*) \leq U_{single}(t^*)$. Thus, if the single-server system is empty at time t^*, the multi-server system is also empty, no bits are being processed, and the departure function cannot overtake the bound. Otherwise, the single-server system is busy and departures are draining from it at the fastest possible rate $\mu(t^*)$—so again the departure function for the multi-server system cannot cross the $D_{single}(t^*)$ bound. □

Appendix C

The Jitter Theorem

Consider a queue in continuous time with an input process $A(t)$ and a time varying server process $\mu(t)$ with an average rate $\mu_{av} = \mathbb{E}\left\{\mu(t)\right\}$. In this appendix, we show that if the server process is stationary and independent of the arrival process, then any moment of unfinished work in the queue is lower bounded by the corresponding moment if the time varying server is replaced by a constant rate server of rate μ_{av}.

Suppose the queue is empty at time zero, when the input stream $A(t)$ is applied. Note that the unfinished work $U(t)$ in the queue at any instant of time is given by

$$U(t) = \max_{\tau \geq 0} \left[A(t) - A(t-\tau) - \int_{t-\tau}^{t} \mu(v)dv \right] \tag{C.1}$$

The above equation is easily verified by noting that the unfinished work at any time instant t is at least as large as the total arrivals minus the total offered service over any interval ending at time t, and the bound is met with equality over the interval $[t_b, t]$, where t_b is the start of the current busy period.

Define $\hat{U}(t)$ as the unfinished work in the system with the same arrival process but with a constant server rate of μ_{av}. An expression for $\hat{U}(t)$ in terms of the input $A()$ is given by using $\mu(v) = \mu_{av}$ in the above equality (C.1).

Theorem 31. *(Jitter Theorem) For any convex, non-decreasing function $f(u)$, we have at every time instant t:*

$$\mathbb{E}f\left(U(t)\right) \geq \mathbb{E}f\left(\hat{U}(t)\right)$$

The following proof uses convexity of the max[] operator in a manner similar to the technique used in [64] to show that fixed length packets minimize delay over all packet input sequences with a given mean packet length.

Proof. For any convex increasing function $f(u)$, we have:

$$\begin{aligned}
\mathbb{E}f\left(U(t)\right) &= \mathbb{E}f\left(\max_{\tau\geq 0}\left[A(t)-A(t-\tau)-\int_{t-\tau}^{t}\mu(v)dv\right]\right) && \text{(C.2)}\\
&= \mathbb{E}_{A()}\mathbb{E}\left\{f\left(\max_{\tau\geq 0}\left[A(t)-A(t-\tau)-\int_{t-\tau}^{t}\mu(v)dv\right]\right)|A()\right\} && \text{(C.3)}\\
&\geq \mathbb{E}_{A()}f\left(\max_{\tau\geq 0}\left[\mathbb{E}\left\{A(t)-A(t-\tau)-\int_{t-\tau}^{t}\mu(v)dv\,|A()\right\}\right]\right) && \text{(C.4)}\\
&= \mathbb{E}_{A()}f\left(\max_{\tau\geq 0}\left[A(t)-A(t-\tau)-\int_{t-\tau}^{t}\mathbb{E}\left\{\mu(v)\,|A()\right\}dv\right]\right) && \text{(C.5)}\\
&= \mathbb{E}_{A()}f\left(\max_{\tau\geq 0}\left[A(t)-A(t-\tau)-\int_{t-\tau}^{t}\mu_{av}dv\right]\right) && \text{(C.6)}
\end{aligned}$$

where (C.3) follows because we have broken the original expectation into an iterated expectation, (C.4) holds by Jensen's inequality together with the fact that the $f(\max[])$ operator is convex,[1] and (C.6) holds because the server process is stationary and independent of the arrival process. The final expression is by definition equal to $\mathbb{E}\left\{f(\hat{U}(t))\right\}$, proving the theorem. □

Thus, any time varying jitter in the linespeed process creates extra queue congestion. We indirectly apply this result in [105] to show any $N \times N$ packet switch scheduler that does not consider queue backlog has an average delay of at least $O(N)$.

C.1 Upper Bound Conjecture

Define $\Phi_X(\mu)$ as the steady state expectation of unfinished work in a queue with an input process $X(t)$ and a constant server rate of μ, and let $\mu(t)$ represent a time varying server process that is stationary and independent of $X(t)$. Let $p(\mu)$ represent the steady state distribution of the $\mu(t)$ process, and let μ_{av} be the average rate. Let $\overline{U}$ represent the average unfinished work in a queue with input $X(t)$ and server process $\mu(t)$.

[1] Indeed, using the non-decreasing convex property of the $f(u)$ function, it is not difficult to show that $f\left(\max_{\tau\geq 0}\left[p_1g_1(\tau)+p_2g_2(\tau)\right]\right) \leq p_1f\left(\max_{\tau\geq 0}\left[g_1(\tau)\right]\right)+p_2f\left(\max_{\tau\geq 0}\left[g_2(\tau)\right]\right)$ for any functions $g_1()$, $g_2()$, and for any probabilities p_1, p_2 such that $p_1+p_2=1$.

Conjecture: $\Phi_X(\mu_{av}) \leq \overline{U} \leq \mathbb{E}_{p(\mu)}\left\{\Phi_X(\mu)\right\}$

The lower bound follows as a special case of the Jitter Theorem, so the conjecture only concerns the upper bound. Thus, we conjecture that the average unfinished work in a queue with a time varying server process that is stationary and independent of the inputs is less than or equal to the expectation of $\Phi_X(\mu)$, where μ is treated as a random variable with a distribution equal to the steady state distribution of the $\mu(t)$ process.

To give intuition about why we expect the upper bound to hold, we note that it trivially holds in the case when the $\mu(t)$ process has a positive steady state probability of being strictly larger than λ, as $\mathbb{E}_{p(\mu)}\left\{\Phi_X(\mu)\right\} = \infty$ in this case. Further note that both the upper and lower bounds are tight when the speed of server variation is decreased to zero or increased to infinity, respectively, while the same steady state distribution is maintained (i.e., for a fixed $\mu(t)$ process, we can consider $\mu(Vt)$ where $V \to 0$ or $V \to \infty$). Indeed, in the case when server variations are very rapid, the average server rate converges to μ_{av} over the course of just a single packet transfer, so that effective service rates are constant and $\overline{U} \to \Phi_X(\mu_{av})$. Alternately, when variations are very slow, the queue reaches steady state behavior for each different channel state, and hence $\overline{U} \to \mathbb{E}_{p(\mu)}\left\{\Phi_X(\mu)\right\}$.

C.2 Alternate Proof of Jitter Theorem

Here we develop an alternate and more intuitive proof of the Jitter Theorem that is based on the multiplexing inequality for queueing systems (Lemma 25 of Appendix B). Consider an input stream $X(t)$ (representing the amount of bits that arrive during the interval $[0, t]$), and a time varying server process $\mu(t)$, representing the instantaneous server rate in units of bits/second. Let $U(t)$ represent the unfinished work in the queue as a function of time (assuming the queue is initially empty). Further let $U_{mutli-server}(t)$ represent the sum unfinished work in a multi-server system with rates $\mu_1(t), \mu_2(t), \ldots, \mu_N(t)$ such that $\sum_i \mu_i(t) = \mu(t)$. We make use of the following basic properties of all queueing systems:

- *Multiplexing Inequality:* $U(t) \leq U_{multi-server}(t)$ for all time $t \geq 0$.

- *Unit Scaling Equality:* For any constant $V > 0$, $VU(t) = \tilde{U}(t)$ for all time $t \geq 0$, where $\tilde{U}(t)$ represents the unfinished work process in a single queue system with an input stream $VX(t)$ and a server rate $V\mu(t)$. That is, $\tilde{U}(t)$ is the resulting unfinished work when both the input and server rate are scaled by V.

The multiplexing inequality is proved in Appendix B (Lemma 25). The unit scaling property holds because scaling both the input and server process by a constant V scales the $U(t)$ sample path by V at every instant of time.[2]

The Jitter Theorem follows directly from these two properties together with the picture shown in Fig. C-1. For simplicity of exposition, we illustrate only the fact that $\mathbb{E}\left\{U(t)\right\} \geq \mathbb{E}\left\{\hat{U}(t)\right\}$, proving Theorem 31 for the special case where $f(u) = u$.

From the figure, we note by the unit scaling property that the unfinished work $U(t)$ at every instant of time is equal to the sum unfinished work when the system is duplicated M times, with scaled inputs and server rates $\frac{1}{M}X(t)$ and $\frac{1}{M}\mu(t)$. Next, note that the expected unfinished work in each duplicate system m is equal to the expected unfinished work in a modified system m' with the same input process $\frac{1}{M}X(t)$ but with the server process replaced by $\frac{1}{M}\mu_m(t)$, where $\mu_m(t)$ is an independent but identically distributed version of the original $\mu(t)$ process. This holds because the original $\mu(t)$ process is independent of the input stream $X(t)$. However, applying the multiplexing inequality, we find that the sum unfinished work in all modified queues is greater than or equal to the unfinished work in a single queue with an input stream $X(t) = \frac{1}{M}\sum_{m=1}^{M} X(t)$ and a server process $\frac{1}{M}\sum_{m=1}^{M}\mu_m(t)$. Thus, the expected unfinished work in the original system is greater than or equal to the expected unfinished work in a new system with the original input stream but with a server process consisting of a sum of M i.i.d. processes $\mu_m(t)$. This holds for all positive integers M, and hence we can take limits as $M \rightarrow \infty$. Because all processes are stationary and independent, it follows by the law of large numbers (applied to processes) that $\frac{1}{M}\sum_{m=1}^{M}\mu_m(t) \rightarrow \mu_{av}$.

[2] The operation of scaling both $X(t)$ and $\mu(t)$ by the same constant $V > 0$ can be viewed simply as expressing the backlog in units other than bits.

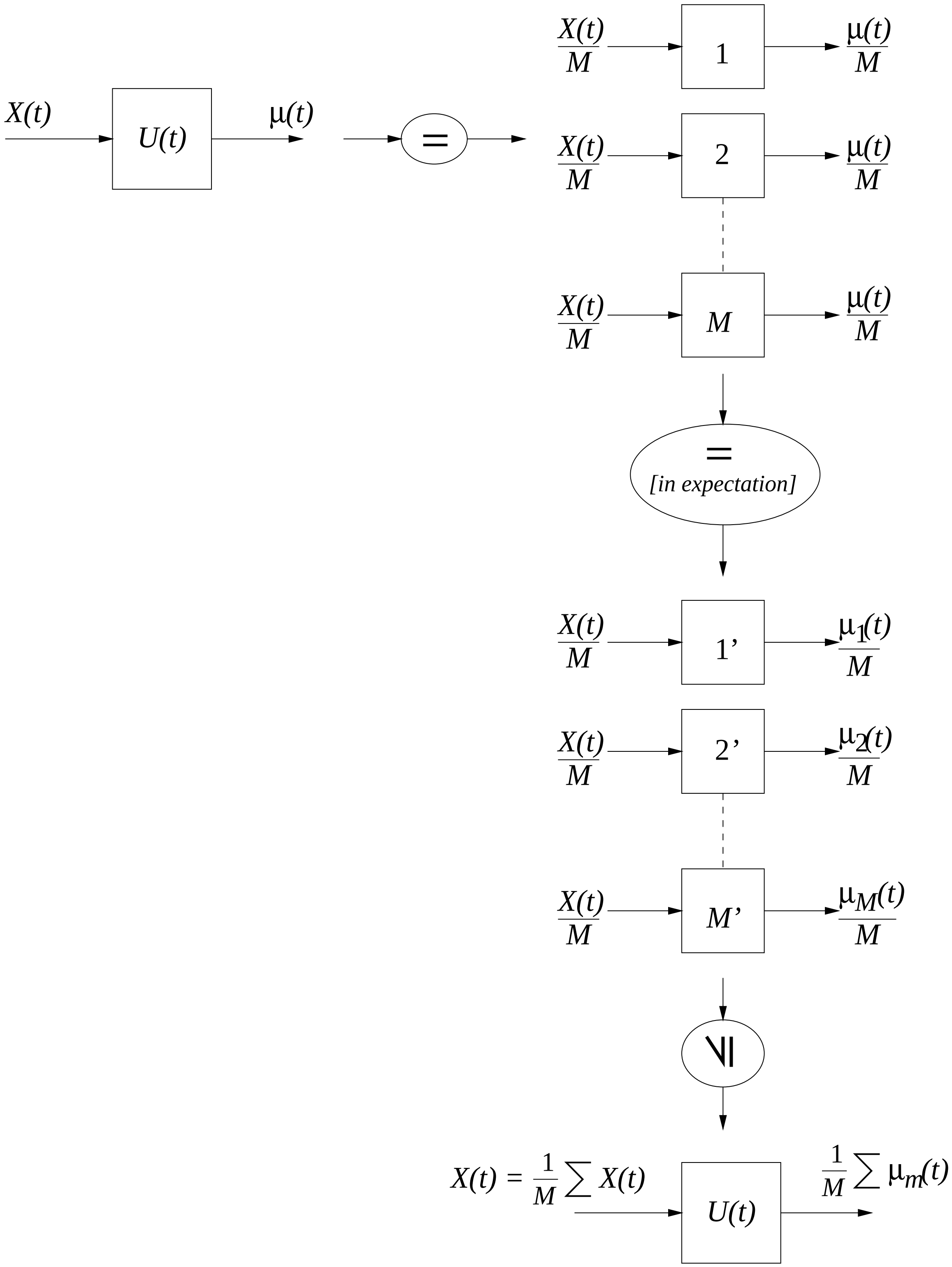


Figure C-1: An illustration proving the Jitter Theorem.

Bibliography

[1] M. S. Alouini, S. A. Borgsmiller, and P. G. Steffes. Channel characterization and modelling for ka-band very small aperture terminals. *Proc. IEEE*, June 1997.

[2] E. Altman, B. Gaujal, and A. Hordijk. Admission control in stochastic event graphs. *IEEE Trans. on Automatic Control*, Vol. 45, no. 5, pp. 854-867, 2000.

[3] E. Altman, B. Gaujal, and A. Hordijk. Balanced sequences and optimal routing. *JACM*, Vol. 47, is. 4, pp. 752-775, 2000.

[4] E. Altman, B. Gaujal, and A. Hordijk. Multi-modularity, convexity, and optimization properties. *INRIA Tech. Report 3181*, June 1997.

[5] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, 2001.

[6] Søren Asmussen. *Applied Probability and Queues, Second Edition.* New York: Spring-Verlag, 2003.

[7] F. Baccelli and P. Brémaud. *Elements of Queueing Theory.* Berlin: Springer, 2nd Edition, 2003.

[8] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad-hoc networks. *IEEE Proceedings of INFOCOM*, April 2003.

[9] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra. New algorithms for an ancient scheduling problem. *Proc. of the 24th Annual ACM Symp. on the Theory of Computing*, 1992.

[10] R. Berry. Power and delay optimal transmission scheduling: Small delay asymptotics. *IEEE Proceedings of the International Symposium on Information Theory*, July 2003.

[11] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135-1149, May 2002.

[12] R. Berry, P. Liu, and M. Honig. Design and analysis of downlink utility-based schedulers. *Proceedings of the* 40*th Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[13] D. P. Bertsekas. *Nonlinear Programming.* Athena, Belmont, MA, 1995.

[14] D. P. Bertsekas and R. Gallager. *Data Networks.* New Jersey: Prentice-Hall, Inc., 1992.

[15] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization.* Boston: Athena Scientific, 2003.

[16] E. Biglieri, J. Proakis, and S. Shamai. Fading channels: Information-theoretic and communications aspects. *IEEE Transactions on Information Theory*, vol. 44(6), pp. 2619-2692, Oct. 1998.

[17] R. K. Boel and J. H. Van Schuppen. Distributed routing for load balancing. *Proceedings of the IEEE*, Vol. 77, no.1, Jan. 1989.

[18] G. Bongiovanni, D. T. Tang, and C. K. Wong. A general multibeam satellite switching algorithm. *IEEE Transactions on Communications*, vol. COM-29, pp. 1025-1036, July 1981.

[19] F. Bonomi and A. Kumar. Adaptive optimal load balancing in a nonhomogeneous multiserver system with a central job scheduler. *IEEE Transactions on Computers*, Vol. 39, no.10, Oct. 1990.

[20] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE Proc. of INFOCOM*, 2003.

[21] O.J. Boxma. Static optimization of queueing systems. *Recent Trends in Optimization Theory and Applications*, ed. R.P. Agarwal (World Scientific Publ. Singapore, pp.1-16, 1995.

[22] J. P. Buzen and P. P-S Chen. Optimal load balancing in memory hierarchies. *Information Processing 74, Proceedings of IFIP*, North-Holland, ed. J. Rosenfeld, pp. 271-275, 1974.

[23] M. Carr and B. Hajek. Scheduling with asynchronous service opportunities with applications to multiple satellite systems. *IEEE Transactions on Automatic Control*, vol. 38, pp. 1820-1833, Dec. 1993.

[24] C-S. Chang. A new ordering for stochastic majorization: Theory and applications. *Adv. Appl. Prob.*, Vol.23, pp.210-228, 1991.

[25] C. S. Chang, X. Chao, and M. Pinedo. A note on queues with bernoulli routing. *IEEE Proc. of 29th Conference on Decision and Control*, 1990.

[26] C-S. Chang, X.L. Chao, and M. Pinedo. Monotonicity results for queues with doubly stochastic poisson arrivals: Ross's conjecture. *Adv. Appl. Prob.*, Vol. 23, pp.210-228, 1991.

[27] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. *IEEE Proceedings of INFOCOM*, March 2000.

[28] J. P. Choi. Channel prediction and adaptation over satellite channels with weather-induced impairments. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[29] J.P. Choi and V.W.S. Chan. Predicting and adapting satellite channels with weather-induced impairments. *IEEE Trans. on Aerospace and Electronic Syst.*, July 2002.

[30] E. G. Coffman, M. R. Garey, and D. S. Johnson. An application of bin-packing to multi-processor scheduling. *SIAM J. of Comput.*, vol. 7, no. 1, Feb. 1978.

[31] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *ACM Proceedings of SIGCOMM*, August 2002.

[32] M. B. Combe and O. Boxma. Static traffic allocation policies. *Theoretical Computer Science*, Vol. 125:17-43, 1994.

[33] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* New York: John Wiley & Sons, Inc., 1991.

[34] R. Cruz and A. Santhanam. Optimal routing, link scheduling, and power control in multi-hop wireless networks. *IEEE Proceedings of INFOCOM*, April 2003.

[35] R. L. Cruz. A calculus for network delay. i. network elements in isolation. *IEEE Transactions on Information Theory*, vol. 37:1, pp. 114-131, Jan. 1991.

[36] R. L. Cruz. A calculus for network delay. ii. network analysis. *IEEE Transactions on Information Theory*, vol. 37:1, pp. 132-141, Jan. 1991.

[37] R.L. Cruz and A. V. Santhanam. Hierarchical link scheduling and power control in multihop wireless networks. *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[38] H. Daduna. *Queueing Networks with Discrete Time Scale.* Springer, 2001.

[39] S. Das, H. Viswanathan, and G. Rittenhouse. Dynamic load balancing through coordinated scheduling in packet data systems. *IEEE Proceedings of INFOCOM*, March 2003.

[40] D. Dutta, A. Goel, and J. Heidemann. Oblivious aqm and nash equilibria. *IEEE Proceedings of INFOCOM*, April 2003.

[41] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. *IEEE Proceedings of INFOCOM*, 2002.

[42] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *Bell Syst. Tech. J.*, Sep. 1963.

[43] S.M. Elnoubi, R. Singh, and S.C. Gupta. "a new frequency channel assignment algorithm in high capacity mobile communication systems". *IEEE Trans. on Vehicular Technology*, Vol. VT-31, no. 3, August 1982.

[44] A. Ephremides and B. Hajek. Information theory and communication networks: An unconsummated union. *IEEE Transactions on Information Theory*, Oct. 1998.

[45] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, Aug. 1980.

[46] B. Fox. Discrete optimization via marginal analysis. *Management Science*, 13, 1966.

[47] A. Fu, E. Modiano, and J. Tsitsiklis. Optimal energy allocation for delay-constrained data transmission over a time-varying channel. *IEEE Proceedings of INFOCOM*, 2003.

[48] R. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, vol. COM-25, pp. 73-85, 1977.

[49] R. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Boston, 1996.

[50] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi. Energy-efficient scheduling of packet transmissions over wireless networks. *IEEE Proceedings of INFOCOM*, 2002.

[51] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. J.*, Sep. 1960.

[52] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math*, vol. 17, no. 2, March 1969.

[53] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *Proceedings of IEEE INFOCOM*, 2001.

[54] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *IEEE/ACM Transactions on Networking, vol. 10, no. 4*, August 2002.

[55] M. Grossglauser and M. Vetterli. Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion. *IEEE Proceedings of INFOCOM*, April 2003.

[56] L. Gün, A. Jean-Marie, A. M. Makowski, and Tedijanto. Convexity results for parallel queues with bernoulli routing. *ISR Tech. Report, University of Maryland*, 1990.

[57] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity in wireless networks. *IEEE Conference on Decision and Control*, 1998.

[58] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, Vol. 46:388–404, March 2000.

[59] B. Hajek. Extremal splittings of point processes. *Math. Oper. Res.*, Vol. 10, No. 4., pp. 543-556, 1985.

[60] B. Hajek. Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control*, vol. AC-29, pp. 491-499, June 1984.

[61] S. V. Hanly and D. Tse. Power control and capacity of spread-spectrum wireless networks. *Automatica*, vol. 35 (12), pp. 1987-2012, Dec. 1999.

[62] J. M. Harrison and J. A. Van Mieghem. Dynamic control of brownian networks: State space collapse and equivalent workload formulations. *The Annals of Applied Probability*, Aug. 1997.

[63] M. Huang, P. E. Caines, and R. P. Malhame. Individual and mass behaviour in large population stochastic wireless power control problems: Centralized and nash equilibrium solutions. *IEEE Conference on Decision and Control*, Dec. 2003.

[64] P. Humblet. Determinism minimizes waiting time. *MIT LIDS Technical Report P-1207*, May 1982.

[65] W. C. Jakes, editor. *Microwave Mobile Communications.* IEEE Press, 1993.

[66] A. Jean-Marie and L. Gün. Parallel queues with resequencing. *Journal of the ACM*, Vol.40,no.5, Nov. 1993.

[67] C. Jin, D. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh. Fast tcp: From theory to experiments. *Submitted for Publication*, April 2003.

[68] N. Jindal and A. Goldsmith. Capacity and optimal power allocation for fading broadcast channels with minimum rates. *IEEE Transactions on Information Theory*, vol. 49, no. 11, Nov. 2003.

[69] R. Johari and J. N. Tsitsiklis. Network resource allocation and a congestion game. *Submitted to Math. of Oper. Research*, 2003.

[70] D. Julian, M. Chiang, D. O'Neill, and S. Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. *IEEE Proceedings of INFOCOM*, 2002.

[71] N. Kahale and P. E. Wright. Dynamic global packet routing in wireless networks. *IEEE Proceedings of INFOCOM*, 1997.

[72] I. Kang and R. Poovendran. Maximizing static network lifetime of wireless broadcast adhoc networks. *IEEE Proceedings of the International Conference on Communications*, 2003.

[73] Y. Karasawa, M. Yamada, and J. E. Alnett. A new prediction method for tropospheric scintillation on earth-space paths. *IEEE Trans. Antennas Propagat.*, Nov. 1988.

[74] D. R. Karger, S. J. Phillips, and E. Torng. A better algorithm for an ancient scheduling problem. *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1996.

[75] M. Karol, M. Hluchyj, and S. Morgan. Input versus output queueing in a space division switch. *IEEE Transactions on Communication*, vol. COM-35, pp. 1347-1356, Dec. 1987.

[76] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 1997.

[77] F. P. Kelly. *Reversibility and Stochastic Networks.* Wiley, Chichester, 1984.

[78] F.P. Kelly, A.Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journal of the Operational Research Society*, 49, p.237-252, 1998.

[79] C.E. Koksal. *Providing Quality of Service over Electronic and Optical Switches.* PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), Sept. 2002.

[80] G. Koole. On the pathwise optimal bernoulli routing policy for homogeneous parallel servers. *Mathematics of Operations Research*, Vol. 21:469-476, 1996.

[81] P.R. Kumar and S.P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Transactions on Automatic Control*, Feb. 1995.

[82] K. Kumaran and M. Mandjes. The buffer-bandwidth trade-off curve is convex. *Queueing Systems*, Vol. 38, pp. 471-483, 2001.

[83] K. Kumaran, M. Mandjes, and A. Stolyar. Convexity properties of loss and overflow functions. *Operations Research Letters*, Vol. 31, pp. 95-100, 2003.

[84] K. Kumaran and L. Qian. Uplink scheduling in cdma packet-data systems. *IEEE Proceedings of INFOCOM*, April 2003.

[85] E. A. Lee and D. G. Messerschmitt. *Digital Communication (Second Edition)*. Boston: Kluwer Academic Publisher, 1994.

[86] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class cdma wireless networks. *IEEE Proceedings of INFOCOM*, 2002.

[87] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Sciences*, 50(2) pp. 228-243, April 1995.

[88] E. Leonardi, M. Melia, F. Neri, and M. Ajmone Marson. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. *Proceedings of IEEE INFOCOM*, 2001.

[89] Z. Liu and R. Righter. Optimal load balancing on distributed homogeneous unreliable processors. *Journal of Operations Research*, Vol. 46:563-573, 1998.

[90] Z. Liu and D. Towsley. Optimality of the round robin routing policy. *J. Appl. Probab.*, 31, pp. 466-475., 1994.

[91] S. H. Low. A duality model of tcp and queue management algorithms. *IEEE/ACM Trans. on Networking*, Vol. 11(4), August 2003.

[92] P. Marbach. Priority service and max-min fairness. *IEEE Proceedings of INFOCOM*, 2002.

[93] P. Marbach and R. Berry. Downlink resource allocation and pricing for wireless networks. *IEEE Proceedings of INFOCOM*, 2002.

[94] C. E. Mayer, B. E. Jaeger, R. K. Crane, and X. Wang. Ka-band scintillations: Measurements and model predictions. *Proc. IEEE*, June 1997.

[95] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *Proceedings of IEEE INFOCOM, pp. 296-302*, 1996.

[96] M. Médard. Channel uncertainty in communications. *IEEE Information Theory Society Newsletter*, vol. 53, no. 2, June 2003.

[97] A. Mekkittikul and N. McKeown. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. *IEEE Proceedings of INFOCOM*, 1998.

[98] S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability.* Springer-Verlag, NY.

[99] J. A. Van Mieghem. Dynamic scheduling with convex delay costs: The generalized c/mu rule. *The Annals of Applied Probability*, Aug. 1995.

[100] J. R. Munkres. *Topology.* NJ: Prentice Hall, Inc., 2000.

[101] M. J. Neely. Queue occupancy in single server, deterministic service time tree networks. Master's thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), March 1999.

[102] M. J. Neely. *Dynamic Power Allocation and Routing in Satellite and Wireless Networks with Time Varying Channels.* PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), 2003.

[103] M. J. Neely and E. Modiano. Convexity and optimal load distributions in work conserving $*/*/1$ queues. *IEEE Proceedings of INFOCOM*, 2001.

[104] M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *Submitted to IEEE Transactions on Information Theory*, 2003.

[105] M. J. Neely and E. Modiano. Logarithmic delay for $n \times n$ packet switches. *Submitted to IEEE Proceedings of INFOCOM*, 2004.

[106] M. J. Neely and E. Modiano. Improving delay in ad-hoc mobile networks via redundant packet transfers. *Proceedings of the Conference on Information Sciences and Systems*, Johns Hopkins University: March 2003.

[107] M. J. Neely, E. Modiano, and C. E. Rohrs. Power and server allocation in a multi-beam satellite with time varying channels. *IEEE Proceedings of INFOCOM*, 2002.

[108] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multibeam satellites with time varying channels. *IEEE Transactions on Networking*, Feb. 2003.

[109] M. J. Neely, E. Modiano, and C. E. Rohrs. Tradeoffs in delay guarantees and computation complexity for $n \times n$ packet switches. *Proceedings of the Conference on Information Sciences and Systems, Princeton University:*, March 2002.

[110] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic routing to parallel time-varying queues with applications to satellite and wireless networks. *Proceedings of the Conference on Information Sciences and Systems*, Princeton: March 2002.

[111] M. J. Neely, E. Modiano, and C.E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Proceedings of INFOCOM*, April 2003.

[112] M. J. Neely and C. E. Rohrs. Equivalent models and analysis for multi-stage tree networks of deterministic service time queues. *Proceedings of the* 38*th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2001.

[113] M. J. Neely, J. Sun, and E. Modiano. Delay and complexity tradeoffs for routing and power allocation in a wireless network. *Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[114] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability and fairness in general networks. *Submitted for Publication*, April 2003.

[115] A. K. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, Vol. 1, pp. 344-357, 1993.

[116] E. Perevalov and R. Blum. Delay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategy. *IEEE Proceedings of INFOCOM*, April 2003.

[117] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2001.

[118] J. W. Roberts and J. T. Virtamo. The superposition of periodic cell arrival streams in an atm multiplexer. *IEEE Transactions on Communications*, Feb. 1991.

[119] S. Ross. *Stochastic Processes.* John Wiley & Sons, Inc., New York, 1996.

[120] S. Ross. *Introduction to Probability Models.* Academic Press, 8th edition, Dec. 2002.

[121] T. Roughgarden. *Selfish Routing.* PhD thesis, Cornell University, May 2002.

[122] S. Sarkar and K.N. Sivarajan. "channel assignment algorithms satisfying cochannel and adjacent channel reuse constraints in cellular mobile networks". *IEEE Proceedings of INFOCOM*, Vol. 1, 1998.

[123] M. Shaked and JG. Shanthikumar. *Stochastic Orders and their Applications.* San Diego, CA: Academic Press, 1994.

[124] S. Shakkottai and R. Srikant A. Stolyar. Pathwise optimality and state space collapse for the exponential rule. *IEEE International Symposium on Information Theory*, 2002.

[125] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, vol. 86 no. 2 pp. 447-457, Feb. 1998.

[126] D. B. Shmoys, J. Wein, and D. P. Williamson. Scheduling parallel machines on-line. *IEEE 32nd Ann. Symp. on Foundations of Comp. Sci.*, 1991.

[127] T. Small and Z. J. Haas. The shared wireless infostation model – a new ad hoc networking paradigm (or where there is a whale, there is a way). *IEEE Proceedings of Mobihoc*, June 2003.

[128] S. L. Spitler and D. C. Lee. Optimization of call admission control for a statistical multiplexer allocating link bandwidth. *To appear in IEEE Trans. on Automatic Control*, 2003.

[129] D. Stoyan. *Comparison Methods for Queues and other Stochastic Models.* John Wiley & Sons: Chichester, 1983.

[130] A. N. Tantawi and D. Towsley. Optimal static load balancing in distributed computer systems. *Journal of the Association for Computing Machinery*, Vol. 32,No.2, pp.445-465, April 1985.

[131] L. Tassiulas. Scheduling and performance limits of networks with constantly changing topology. *IEEE Transactions on Information Theory*, May 1997.

[132] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, Vol. 37, no. 12, Dec. 1992.

[133] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, pp. 466-478, March 1993.

[134] S. Toumpis and A. Goldsmith. Some capacity results for ad-hoc networks. *8th Annual Allerton Conference Proceedings*, 2000.

[135] D. Towsley, P. D. Sparaggis, and C. G. Cassandras. Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control*, Vol. 37:1446-1451, 1992.

[136] D. Tse and S. Hanly. Multi-access fading channels: Part 1: Polymatroid structure, optimal resource allocation, and throughput capacities. *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796-2815, Nov. 1998.

[137] D. Tse and S. Hanly. Multi-access fading channels: Part ii: Delay-limited capacities. *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2816-2831, Nov. 1998.

[138] D. N. Tse. Optimal power allocation over parallel broadcast channels. *Proceedings of the International Symposium on Information Theory*, June 1997.

[139] V. Tsibonis, L. Georgiadis, and L. Tassiulas. Exploiting wireless channel state information for throughput maximization. *IEEE Proceedings of INFOCOM*, April 2003.

[140] A. Tsirigos and Z. J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine*, Nov. 2001.

[141] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. Energy-efficient packet transmission over a wireless link. *IEEE/ACM Trans. Networking*, vol. 10, pp. 487-499, Aug. 2002.

[142] I. Viniotis and A. Ephremides. Extension of the optimality of the threshold policy in heterogeneous multiserver queueing systems. *IEEE Transactions on Automatic Control*, vol. AC-33, pp. 104-109, Jan. 1988.

[143] H. Viswanathan and K. Kumaran. Rate scheduling in multiple antenna downlink wireless systems. *Proc. of Allerton Conference*, Oct. 2001.

[144] J. Walrand. *An Introduction to Queueing Networks*. Prentice-Hall, Englewood Cliffs: NJ, 1988.

[145] R. W. Weber. On optimal assignment of customers to parallel servers. *J. Appl. Probab.*, Vol. 15, pp. 406-413, 1978.

[146] W. Winston. Optimality of the shortest line discipline. *J. Appl. Probab.*, Vol. 14, pp. 181-189, 1977.

[147] P. Wu and M. J. M. Posner. A level-crossing approach to the solution of the shortest-queue problem. *Operations Research Letters*, 1997.

[148] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation for wireless networks. *Proceedings of the 39th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2001.

[149] E. M. Yeh and A. S. Cohen. Throughput and delay optimal resource allocation in multiaccess fading channels. *Proceedings of the International Symposium on Information Theory*, May 2003.

[150] W. H. Yuen, R. D. Yates, and S-C Mau. Exploiting data diversity and multiuser diversity in noncooperative mobile infostation networks. *IEEE Proceedings of INFOCOM*, April 2003.