CAPACITY ASSIGNMENT IN NON-SWITCHING MULTICHANNEL NETWORKS

by

SOUNG CHANG LIEW

S.B., Massachusetts Institute of Technology (1984)
S.M., Massachusetts Institute of Technology (1986)
E.E., Massachusetts Institute of Technology (1986)

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

February 1988

Signature of Author ________________________________

Department of Electrical Engineering and Computer Science
January 8, 1988

Certified by ________________________________

Professor Robert S. Kennedy
Thesis Supervisor

Accepted by ________________________________

Professor Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

CAPACITY ASSIGNMENT IN NON-SWITCHING MULTICHANNEL NETWORKS

by

SOUNG CHANG LIEW

Submitted to the Department of Electrical Engineering and Computer Science on January 8, 1988 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

ABSTRACT

Fundamental properties of non-switching multichannel communication networks are studied. We assume optical fiber is used, implying that bandwidth is abundant. Because of the speed limitations of electronic transmitters and receivers, a concurrency scheme is needed to exploit the high-bandwidth capability. One could divide the bandwidth into many frequency channels so that the bandwidth of each channel is compatible with the transmitters' and receivers' speed.

Two obvious ways to establish connectivity between the nodes in a multichannel network are using switches and using tunable transmitters and receivers. However, to appreciate when and why these networks are desired, it is essential to first understand non-switching networks.

A flow model is used to describe the network traffic. The Capacity Assignment Problem (CAP) deals with mapping the traffic flows between the nodes onto the channels such that the total numbers of channels, transmitters and receivers (N_c, N_t and N_r, respectively), are minimized. Approaches to this problem include precise math programming formulation, heuristics and block designs. The inherent difficulty of the problem is established by relating it to the class of NP-complete problems. Using lower and upper bounds, it is shown that the trade-off between N_t and N_r is strong , but that between these two parameters and N_c is weak. Furthermore, N_t and N_r are necessarily very large for a high-traffic network with many nodes. This problem can be alleviated by partitioning the network into a hierarchical structure. For completeness, the relationship between CAP and access control schemes is also investigated. On the whole, this work suggests that in studying a switching or a tuning network, a focus should be on reducing N_r and N_t.

Thesis Supervisor: Dr. Robert S. Kennedy
Title: Professor of Electrical Engineering

ACKNOWLEDGEMENTS

This is it! Finally, I am leaving school. So many people have contributed to my education at M.I.T. that I don't even know where to begin my note of appreciation. On the other hand, this is not a thesis on *acknowledgements*. So, please don't get offended if I purposefully "forget to mention your name inadvertently" here. You see, I believe that acknowledgements, like proofs of mathematical theorems, are more elegant if they are shorter. Let's start before I decide to write the acknowledgements in matrix form.

First and foremost, I would like to thank M.I.T. for providing such a great learning environment. I would not have come to the United States for my undergraduate study if not for the generous financial aids given to me. Very few colleges are that magnanimous to a foreign student. I can still remember interviewing with an Harvard alumnus for admission and being told that I would stand a better chance if I were to be a son of an ambassador. I didn't ask him about financial aids.

Anyway, if I did not come here, I would not have met my wife, So Kuen, during our freshman year. And if I had not met her, who knows whether this thesis would have existed. It definitely would not have been what it is now, for better or worse. So Kuen, thank you for your support and the diversions that make my life so much more interesting. By the way, for the curious, I typed this thesis myself.

Professor Kennedy is instrumental in shaping my graduate study here. I learned a lot, in academic subjects as well as non-academic matters, while doing my master's and PhD dissertations under him. Besides Professor Kennedy, I would also like to thank the other two thesis readers of mine, Professor Gallager and Professor Tsitsiklis. They all did a good job in catching the mistakes "intentionally" introduced into the draft copy of this thesis. Professor Humblet has also given many valuable comments to this work.

My colleagues of the Local Communication Networks (LCN) research group have all been wonderful. I enjoyed our discussions and learned much from their research results and interesting ideas.

My parents deserve the highest recognition from me. I owe my achievements to their teachings and their encouragement to excel. This thesis is dedicated to them.

Best wishes to the LCN group! I am sure the research group will prosper under the leadership of Professor Kennedy and Professor Humblet.

Contents

Glossary of Notation

a_{pq}	Channel assigned to traffic (p, q).	**63**
α	Absolute traffic from one node to another for the uniform traffic CAP.	**156**
BPP	Bin Packing Problem.	**49**
c	Cost of a CAP solution.	**63**
c_d	Fraction of channel capacity devoted to data messages.	**145**
c_v	Fraction of channel capacity devoted to voice messages.	**145**
$\underline{c_d}$	Lower limit on c_d.	**145**
$\underline{c_v}$	Lower limit on c_v.	**145**
c_{pq}	Local cost of traffic (p, q).	**63**
C	The set of available channels in network.	**21**
CAP	Capacity Assignment Problem.	**21**
g	The number of groups or level-1 subnetworks in a hierarchical network.	**118**
G_i	The set of nodes in group i of a hierarchical network.	**118**
γ	Normalized traffic from one node to another for the uniform traffic case: $\alpha/\rho r$.	**62**

k	Number of objects in a block of a block design.	98
LP	Linear Program.	32
λ	Total normalized traffic in system, i.e. $\sum_{(p,q)} \lambda^{(p,q)}$.	36
$\lambda^{(p,q)}$	Normalized traffic from node p to node q.	21
Λ	Arrival rate per channel.	135
Λ_{P_b}	The arrival rate per channel at which $P_{block} = P_b$.	136
m	The number of minichannels in a channel.	135
MILP	Mixed Integer Linear Program.	32
MNFP	Multicommodity Network Flow Problem.	28
n	Connectivity requirement of network: two nodes must have at least n different channels for communication purposes.	95
n_{rt}	The number of blocks in which an object occurs in a block design.	98
$n^{(k)}$	The number of different traffic types assigned to channel k.	63
$n_r^{(p,q)}$	The largest integer not greater than $\lambda^{(p,q)}$.	40
$n_r^{(k)}$	The number of receivers receiving on (or the local receivability of) channel k.	65
$n_t^{(k)}$	The number of transmitters transmitting on (or the local transmittability of) channel k.	65
$n_{p,\cdot}^{(k)}$	Total number of traffic types $(p,\cdot)$ assigned to channel k.	63
$n_{\cdot,q}^{(k)}$	Total number of traffic types $(\cdot,q)$ assigned to channel k.	63
N	Total number of nodes in network.	20
N_c	Total number of channels in network.	20

N_{con}	Number of constraints in the MILP formulation of CAP.	33
N_r	Receivability of communication network.	20
NSTCAP	Non-splitting traffic capacity assignment problem.	35
N_t	Transmittability of communication network.	20
N_{var}	Number of variables in the MILP formulation of CAP.	33
P_b	Blocking probability requirement; i.e. $P_{block} \leq P_b$.	136
P_{block}	Blocking probability.	136
r	Channel transmission rate.	156
ρ	Offered load to a channel: load = input rate/maximum output rate.	136
ρ_{P_b}	Offered load to a channel at which $P_{block} = P_b$.	136
ρ_d	Offered load of data messages.	145
ρ_v	Offered load of voice messages.	145
SBPP	Splitting Bin Packing Problem.	50
S_n	The set of arcs belonging to stage n in the multicommodity network of CAP.	31
SPP	Set Partition Problem.	51
STCAP	Splitting traffic capacity assignment problem.	35
$\sigma^{(p,q)}$	Absolute traffic from node p to node q.	136
T	Average delay of a message.	142
$v_r(x)$	Cost to a node that receives a traffic flow of x on some channel.	21

v_r	Step size of $v_r(x)$ if it is a step function.	**21**
$v_t(x)$	Cost to a node that sends a traffic flow of x on some channel.	**21**
v_t	Step size of $v_t(x)$ if it is a step function.	**21**
$w(x)$	Cost of having a traffic flow of x on some channel.	**21**
w	Step size of $w(x)$ if it is a step function.	**21**
$\mathbf{x}$	Vector consisting of $x_{(i,j)}$ for all (i,j).	**31**
x_k	Traffic on channel k.	**21**
$x_k^{(p,q)}$	Traffic flow from node p to node q that is carried on channel k.	**21**
$x_{(i,j)}$	Aggregate commodity flow on arc (i,j) in Multicommodity Network Flow Problem.	**31**
$\mathbf{x}^{(\cdot,q)}$	Vector consisting of $x_{(i,j)}^{(\cdot,q)}$ for all (i,j).	**31**
$x_{(i,j)}^{(\cdot,q)}$	Flow of commodity $(\cdot,q)$ on arc (i,j) in Multicommodity Flow Problem.	**30**
$\mathbf{y}$	Vector consisting of $y_{(i,j)}$ for all (i,j).	**32**
$y_{(i,j)}$	An integer variable which is 1 if $x_{(i,j)} > 0$ and 0 otherwise.	**32**

Chapter 1

Introduction

1.1 Background

Recent advances in fiber-optic technologies have prompted the use of optical fiber as an information transmission medium in many modern communication systems. One of the advantages of optical fiber is its high bandwidth. A key question, however, is how and to what extent this high bandwidth allows us to design better and more capable communication systems.

Most traditional communication systems have been designed on the premise that channel bandwidth is a scarce and expensive resource. A network designer will therefore concentrate on optimizing the use of the channel, often leaving other network costs out of consideration. This approach needs to be reassessed in a fiber-optic system since the costs of peripheral devices, such as transmitters and receivers, are not negligible compared with that of the channel bandwidth. This is doubly so in short-distance communication systems where the fiber lengths are likely to be short.

The ultimate goal of this research is to develop analytical tools and methods that will help identify and study important issues related to high-bandwidth networks.

Through this work, it is hoped that the fundamental trade-offs between different network design considerations can be understood more concretely and systematically. Conceding that an all-encompassing study of the subject is probably not realistic, we limit the scope of this thesis to a non-switching network for the most part. Nevertheless, this work serves as a foundation upon which more sophisticated networks can be studied. Furthermore, it is essential to first understand the simple non-switching network in order to appreciate the alternatives of switching and tuning in networks.

The motivation for partitioning the capacity in a fiber-optic network into multiple frequency channels is that the overall bandwidth of a fiber far exceeds the bandwidths of practical electronic transmitters and receivers. Concurrency (the capability of having more than one message in a network simultaneously) is therefore desired if the fiber bandwidth is to be more fully utilized [38]. Frequency multiplexing is simply a straight-forward way of achieving concurrency.

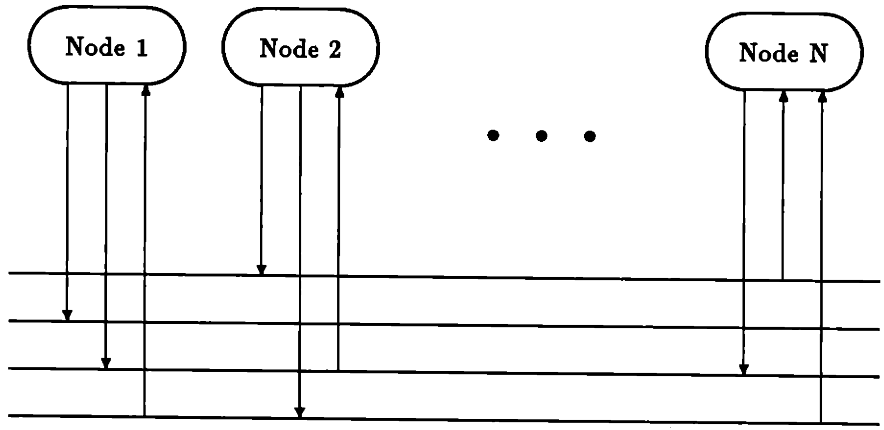
At present, fiber-optic technologies are predominantly applied to long-distance communication systems. In most of these systems, the fiber is used as a point-to-point link. Typically, the nodes are far apart and the traffic between two nodes is very high because of the aggregation of the individual traffic streams from end users. In other words, each node actually represents a group of users. There have also been substantial efforts devoted to fiber-optic Local Area Networks (LAN's) [31,33,41]. Here, the nodes usually correspond to the actual end-users and they are located in a small geographical region. In a LAN, the average traffic between two nodes is comparatively small and the traffic rate viewed over a long time period is rather bursty, whether the network supports data services or voice services — two users do not talk to one another constantly throughout the day. The statistical multiplexing, or aggregation effect which "smooths" out traffic generally does not apply to LAN's.

This thesis focuses on Metropolitan Area Networks (MAN's) in which the traffic of a node is the aggregate traffic of many end users. For example, the traffic between two nodes may correspond to the traffic between two buildings. Specifically, a metropolitan network here refers to any network within a city limit in which each node represents, say more than ten end users. Compared with a long-distance network, the nodes in a MAN are densely populated in a small geographical region. Compared with a LAN, on the other hand, the system traffic is relatively high, say on the order of tens to hundreds of gigabits per second. This, of course, assumes there are services demanding such a high traffic rate in the future. Otherwise, the motivation for such a wide-band network will be lacking. As in a long-haul network, many simultaneous services are going on between two nodes in a MAN; thus, there is some degree of "traffic smoothing" due to aggregation.

1.2 Preliminary Discussion and Problem Statement

1.2.1 Preliminary Discussion

Basically, the capacity assignment problem deals with assigning the capacity in a non-switching multichannel network to the communicating nodes in order to satisfy their communication requirements. Here, switching is defined to be the capability of dynamically directing traffic from one channel to another; this functionality may resides in a central switch, or in the peripheral nodes, as in a multihop network. One may think of the network as an extension of the single-channel multiaccess network. Fig. 1.1 provides a mental picture for such a network; the actual physical networks may look very different, as long as Fig. 1.1 is its logical equivalent. The



Note: Node 1 cannot transmit to Node N

Figure 1.1: A Multidrop Multichannel Network

problem is static in the sense that once a node is given the capability of transmitting (or receiving) on a particular channel, it retains that capability permanently. For example, if the channels correspond to different frequency bands, being able to transmit on a channel means having the hardware to do so. An underlying assumption is that the nodes are capable of extracting messages destined to them out of the information streams passing by. It must be stressed that this selection capability is fundamentally different from switching since switching is the capability of directing messages from one channel to another.

Many previous works on "Capacity Assignment" integrate scheduling or access control results and the capacity assignment problem into a single framework. For example, the trunk assignment problem in telephone networks [4] deals with the problem of assigning trunk capacity between two nodes in order to achieve certain minimum blocking probability, a parameter value readily available from the Erlang B Formula based on the $M/M/m/m$ queueing discipline. The capacity assignment problem in data communication networks [6,18,22,36] concerns assigning link capacities between nodes to minimize the overall expected delay, which again is a result obtained by assuming certain queueing discipline.

In this thesis, since the channels are multiaccess rather than point-to-point, and there is no switching, not only is it necessary to supply enough capacity to accommodate the total network traffic, the channel capacities must be assigned to the nodes in such a way that at least a channel is available for transmission of messages from one node to another. In Fig. 1.1, for instance, the solution is not acceptable if there is some traffic from node 1 to node N. One way to achieve full connectivity trivially is to let every node transmit and receive on all channels. The problem then reduces to a situation wherein one does not have to worry about the connectivity issue when dealing with the capacity assignment problem. Such an arrangement may be too expensive, especially if there is a large number of channels. Therefore it is desirable to reduce the number of channels a node receives and transmits on. These numbers will be called the *transmittability* and *the receivability* of the node. Another extreme is to devote an entire channel to each traffic stream from one node to another, resulting in a network that is logically equivalent to a mesh network. This solution is undesirable if the traffic between two nodes is very small compared to the channel capacity.

We assume that the channel capacities are equal and that each channel has enough capacity to support many services simultaneously. If there is a diversity of the services with very different service requirements, it is difficult, if not undesirable, to adopt one common scheduling scheme. To get around this problem at the capacity assignment level, we assume associated with a channel is an "effective" capacity[1] which is lower than the true capacity, or the transmission rate, of the channel —

[1] To avoid confusion, we emphasize that the capacity here is definitely not the *information-theoretic* capacity: an assumption is that the channels are reliable, or the error rates are negligible.

to account for the overhead needed to achieve the performance requirements of the services (e.g. blocking probability and delay). Thus, the performance issue will be neglected in the beginning. How scheduling aspect comes into play will be addressed in a later chapter after we have understood the capacity assignment problem better. But, on an intuitive level and from many queueing-theoretic results, keeping the performance requirements fixed, the effective capacity of a channel approaches the true channel capacity as we increase the channel capacity. In other words, a large channel can be used more efficiently than a small one. This is due to the higher degree of channel sharing, or the implicit statistical multiplexing, between services. In this limit, the scheduling problem can be justifiably ignored.

To describe the traffic between nodes, a flow model is adopted. If the total traffic flow assigned to a channel is less than the effective capacity, then we assume the performance requirements are automatically satisfied by some underlying dynamic access control or scheduling mechanisms. The effective capacity is assumed to be the only coupling between the capacity assignment problem and the access control problem. This is not to say these two problems are independent. In fact, it is conceivable that capacity assignments of certain structures will lead to a better performance in scheduling control, and this in turn means the effective capacity is actually higher than that originally assumed. For illustration, consider a voice network. If there are more than one alternative channel for the traffic between two nodes, the blocking probability will be smaller than that of a network with only one such channel, assuming the same channel load. This means the channels in a *2-connectivity* network can support a higher load, or have a higher effective capacity, given some fixed blocking probability. Nevertheless, the "independence" assumption is desired, at least at the preliminary stage, so as to simplify matters. Subsequent modifications can be included when considering a particular access control scheme.

In summary, the flow model is a good model if the following situations apply:

- The capacity of a single channel is large enough so that many services are in progress on the channel at the same time.
- Each node actually corresponds to many end users in that messages originating from it come from many end users.

Then, the traffic of a particular service is just a small part of the overall network traffic. Furthermore, because of the averaging effect, the aggregate traffic "pouring" out of a node appears more or less like a constant flow even though the individual messages may be bursty. Again, if the channel is small, the burstiness of messages will surface and the scheduling aspect of the network must be taken into account. With our formulation, this amounts to adjusting the effective channel capacity, and this can be done quite simply in certain situations, as will be seen in Chapter 7.

1.2.2 Problem Statement

Roughly, the Capacity Assignment Problem (CAP) is the problem of mapping the traffic flows between N nodes onto the channels such that the total number of channels N_c, the system transmittability N_t (i.e. the sum of transmittabilities over all nodes) and the system receivability N_r are minimized. Thus, an assumption is that it is desirable to decrease N_c, N_t and N_r, and some motivating examples will be discussed in the next section. The physical fiber lengths are not included as a cost since this will entail considering the actual physical network topologies and we are more interested in the logical constructs of non-switching networks. A brief discussion on fiber lengths, however, is given in Chapter 6.

For a more precise definition, let's first assume a more general cost structure. To that end, we let the effective capacity of a channel be 1 (or equivalently, all traffic

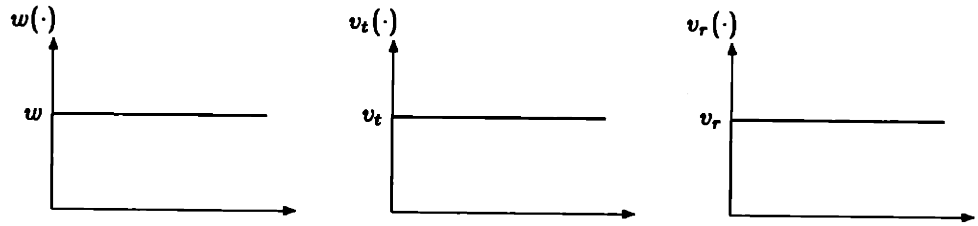


Figure 1.2: Step Cost Functions

flows are normalized by the effective capacity of a channel) and adopt the following notation:

$$\begin{aligned}
\lambda^{(p,q)} &= \text{traffic from node } p \text{ to node } q, \\
x_k &= \text{total traffic on channel } k, \\
x_k^{(p,q)} &= \text{traffic from node } p \text{ to node } q \text{ that is carried on channel } k, \\
C &= \text{the set of available channels in the network}, \\
w(x) &= \text{cost of having a traffic of } x \text{ on a channel}, \\
v_t(x) &= \text{cost to a node that transmits a traffic of } x \text{ on some channel}, \\
v_r(x) &= \text{cost to a node that receives a traffic of } x \text{ on some channel}.
\end{aligned}$$

A CAP is the following optimization problem:

$$\begin{aligned}
\min \quad & \sum_k \left[w(x_k) + \sum_p v_t\left(\sum_q x_k^{(p,q)}\right) + \sum_q v_r\left(\sum_p x_k^{(p,q)}\right)\right] \\
\text{s.t.} \quad & \sum_k x_k^{(p,q)} = \lambda^{(p,q)} \quad \forall\ (p,q), \\
& x_k = \sum_{(p,q)} x_k^{(p,q)} \leq 1 \quad \forall\ k \in C.
\end{aligned} \tag{1.1}$$

If we are only interested in a linear combination of the total number of channels, the system transmittability and the system receivability, then $w(\cdot)$, $v_t(\cdot)$ and $v_r(\cdot)$ are all step functions (see Fig. 1.2) with step sizes w, v_t and v_r respectively. Except in the beginning of Chapter 2, these will be the assumed cost functions throughout this report. Note that $|C|$ is the maximum number of channels available and some of the channels may not be assigned any traffic at all in a feasible solution (i.e $N_c \leq |C|$). In general, the system parameters, N_t, N_r and N_c cannot be minimized simultaneously. How they interact with each other is studied in detail in this research.

1.3 Examples of Related Situations

We now discuss a few situations to which the study here applies.

Example 1.1 This example is the motivation behind this study, and as such the formulation fits naturally into this setting. Consider a high bandwidth network in which the bandwidth is frequency-divided into many channels, say hundreds of them. Furthermore, there is no frequency translator within the network and the transmitters and receivers are not tunable. In this case, a separate physical transmitter(receiver) is needed for the transmission(reception) on a particular channel. In addition, a traffic flow from one node to another must be assigned to at least one explicit channel since there is no internal frequency switching; with switching this is not necessary since traffic transmitted on one channel may be received on another channel through frequency translation. Thus, the transmittability and the receivability of a node correspond to the total numbers of transmitters and the receivers at the node. Given the traffic flows between nodes, can we reduce the total numbers of channels, transmitters and receivers in the system? This is a multiobjective optimization problem. One way to approach this problem is to assign relative weights to the costs of a channel, a transmitter and a receiver. Relating this to our formulation, these are the step costs of w, v_t and v_r respectively.

Example 1.2 Suppose the transmitters and receivers are tunable. Then, we may be able to reduce the numbers of transmitters and receivers at each node since a transmitter or a receiver can be shared by several channels. However it is still desirable to reduce the transmittability and the receivability since they now correspond to the tunabilities of the transmitters and the receivers, and hardware that tunes over a smaller number of channels may be cheaper. What is not captured here is the tuning range, which in practice may be a more important factor than the absolute number of tunable frequencies.

Example 1.3 The multiple channels are not necessarily frequency-divided channels. For a Code-Division-Multiaccess (CDMA) system, they correspond to different codewords. Recently, there has been much research on optical codes for CDMA purposes [8,13,20]. A codeword is a string consisting of 0's and 1's where a 1 corresponds to an optical pulse and a 0 corresponds to the absence of such a pulse. By minimizing the overlappings of pulses in the codewords, semi-orthogonality is achieved. The signals at the detection ends are extracted by performing a correlation operation. Optical tapped delay lines have been proposed for the generation and the detection of codewords and a set of delay lines with unique tap positions is needed for each codeword. Since the physical lengths of the delay lines are long and they are likely to be expensive, we would like to reduce the sets of optical delay lines at each node. This translates to reducing the transmittability and the receivability in our formulation. Another problem of certain optical CDMA schemes (e.g. those of [8]) is the limited number of codewords that can be constructed due to the orthogonality requirement. Hence, it is desirable to reduce the total number of codewords required in the system. This is equivalent to reducing the number of channels in our formulation. In short, reducing the transmittability and the receivability of a node corresponds to reducing the number of codewords to be generated and detected by the node, and reducing the number of channels corresponds to reducing the size of the set of codewords in the whole system.

1.4 Overview

Fig. 1.3 shows how the different chapters of this thesis are interrelated. An arrow from A to B means the discussions in A lead to the discussions in B.

Chapter 2 shows how to solve CAP *exactly* as a multicommodity network flow problem, which can be formulated as a mixed integer linear program. The resulting size of the problem is large and a fast algorithm is not available. This leads us to suspect that CAP is a difficult algorithmic problem in general.

In Chapter 3, we study the relationship between CAP and other optimization problems. Some simple theorems and false conjectures are presented to familiarize the reader with the problem. This chapter is concluded with a proof showing that CAP is indeed an intractable problem in that it is NP-hard. It is therefore unlikely to be fruitful to attempt to solve CAP exactly.

Chapter 4 explores heuristics that look for suboptimal CAP solutions. Various lower bounds and upper bounds on the system parameters are obtained; and the trade-offs between the parameters are studied based on these bounds. Essentially, there is a strong fundamental trade-off between the system transmittability and the system receivability. The number of channels, on the other hand, is largely determined by the system traffic and is independent of these two parameters. We also show that when the total system traffic is large but the individual traffic between any two nodes is small, simple heuristics yield very good solutions approaching those of an exact method. Ironically, in this limit, the system transmittability and receivability also necessarily become very large; and this is a fundamental property of a non-switching network. General rules of thumb on how to devise good heuristics are also discussed. The chapter is concluded with a discussion on solving an n-connectivity CAP, a version of CAP in which there must be at least n alternative

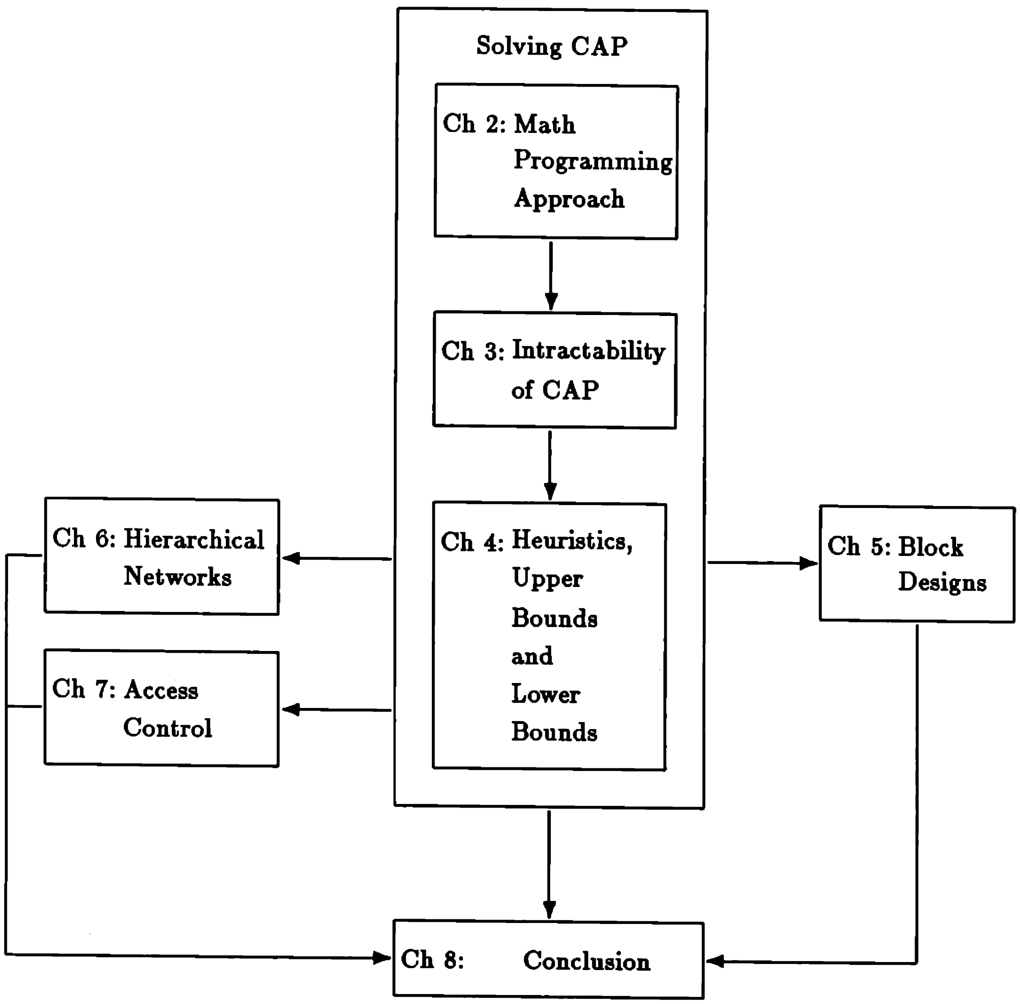


Figure 1.3: Logical Relationship between Different Chapters of Thesis

channels for the traffic from one node to another.

In Chapter 5, the combinatorial study of Block Designs is related to a version of CAP with more symmetry requirements. For example, one of the additional constraints is that the transmittable channels and the receivable channels of a node must be the same. Another constraint is that the number of channels on which two nodes may communicate is the same for all node-pairs. It is shown that under the two additional constraints, the number of channels is bounded below by the number of nodes regardless of the actual traffic requirements. This may not be desirable if the system traffic is small, but the only way to avoid this situation is to drop at least one of the constraints. An implication is that any access control scheme that depends on (or assumes) the two constraints may lead to poor channel utilization. As will be established more concretely, poor channel utilization results in poor transmitter and receiver utilization.

A way to reduce the system transmittability and receivability is to introduce hierarchical structures into the network. Based on a logical definition of hierarchical networks, we show how this can be accomplished in Chapter 6. Intuitive arguments explaining the results are also presented and they point to some directions for further research. Basically, the reduction of system transmittability and receivability is brought about by the implicit introduction of switching in a hierarchical network; a hierarchical network can be considered as a type of distributed switching network. Thus, a function of switching is to decrease the transmitter and receiver costs at the peripheries of the network. Indeed, it can be shown that hierarchical structures can reduce the system transmittability and receivability very significantly when the total system traffic is very large and the individual traffic between two nodes is small. This is also the situation in which the two parameters get very big in a non-switching network. A conclusion is that switching, whether implemented distributedly, as in a hierarchical network, or centrally, is desirable under such a condition.

Chapter 7 is devoted to the study of the interaction between access control and CAP. The implications for the system parameters is investigated based on various analytical access control models. The results support our claim that the flow model is a good model and that scheduling effects can be ignored in a large channel situation. If the channel capacity is small, a counter-intuitive result is that the system transmittability and receivability actually decreases as we increase the network's connectivity (or the number of channels for the traffic from one node to another). In this situation, for some fixed performance requirements, we actually need fewer channels, fewer transmitters and fewer receivers if we increase the connectivity requirements. There seems to be no obvious engineering trade-off.

Chapter 8 concludes this thesis by exploring the implications of the results for fiber-optic networks. The chapter also summarizes the insights gained from this work and discusses a few topics that call for further research. An especially interesting topic is the study of the trade-off between the amount of switching in a network and the system transmittability and receivability. Preliminary discussions on the approaches to these problems and some important issues to be addressed are also presented.

Chapter 2

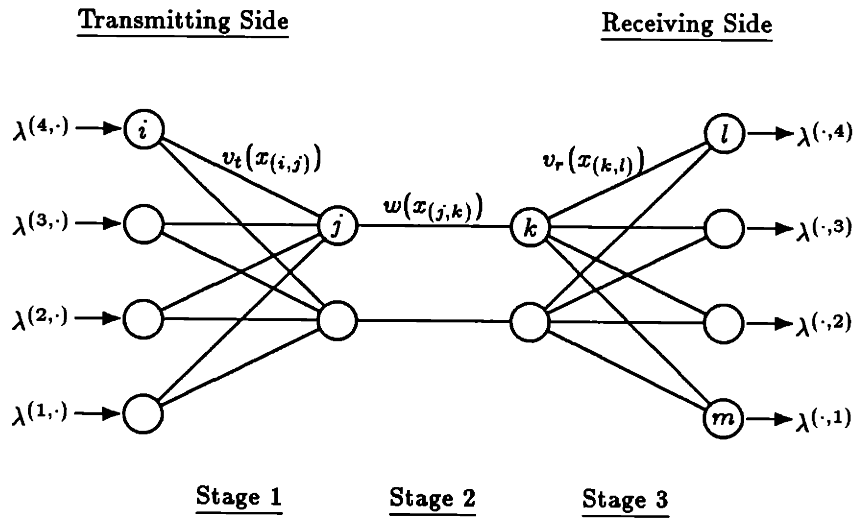
Mathematical Programming Approach

The capacity assignment problem can be formulated formally as a nonlinear-cost Multicommodity Network Flow Problem (MNFP) [37]. We will show that the size of the resulting problem is very large; therefore, unless the exact optimal solution is required, it is not an attractive method. However, this is the only method that finds the optimal solution in this work.

2.1 Formulation

An example where there are four nodes and two channels is depicted in Fig. 2.1. As shown, there are three stages of arcs. The first stage and the third stage model the transmittability and receivability cost. The middle stage models the cost corresponding to the total traffic flow on each channel. The arcs here correspond to the channels in the communication network. Each communication node is represented by two multicommodity-network nodes, one on the transmitting side and one on the receiving side.

Before dwelling into further details, a subtle point needs clarification. That is, the number of channels is fixed at the outset in this approach. One may argue that



Direction of arcs is from left to right.

$$\lambda^{(p,\cdot)} : \{\lambda^{(p,q)} \quad \forall \quad q \neq p\}$$
$$\lambda^{(\cdot,q)} : \{\lambda^{(p,q)} \quad \forall \quad p \neq q\}$$

Figure 2.1: Multicommodity Network Flow Model of CAP

in practice there is always a natural limit on $|C|$ and we may simply set $|C|$ at this limit when solving the MNFP. This natural limit may be overly large compared with N_c, the actual number of channels with non-zero flows in the final solution. This unnecessarily increases the size of the MNFP.[1] This can be compromised by fixing $|C|$ at a value higher than the total system traffic, say $|C| = \sum_{(p,q)} \lambda^{(p,q)} + \delta$ where $\delta \geq 1$. If there are unoccupied channels in the final solution then the solution after eliminating the unused channels will be accepted. Otherwise, we increase the value of $|C|$ and solve the multicommodity flow problem again. This solution may not be optimal with respect to the original CAP. In other words, there exist cost functions such that although there are unused channels in the solution of the MNFP for some fixed $|C|$, by increasing $|C|$ further we obtain a lower-cost solution without any idle channel. However, based on the heuristic studies in Chapter 4, when the

[1] Recall that we are interested in a high bandwidth environment where the bandwidth of the communication medium is much higher than the system traffic.

individual node-to-node traffic flows are small relative to the capacity of a single channel and the cost functions are the step costs as shown in Fig. 1.2, the optimal N_c should not be too much greater than the total system traffic.

In this chapter, the word "nodes" is used to refer to both the communication nodes and the multicommodity network's nodes. Where their distinction is not implicit by context, the explicit terminologies will be used. In Fig. 2.1, the transmitting aspect of communication node 4 is represented by node i and the receiving aspect by node l. For exactness, one may define two functions, $f_t(p)$ and $f_r(p)$, which map communication node p to a node on the transmitting side and a node on the receiving side respectively. The inverse mappings will be denoted by f_t^{-1} and f_r^{-1}. Thus, $f_t(4) = i$, $f_r(4) = l$ and $f_t^{-1}(i) = f_r^{-1}(l) = 4$.

Corresponding to each ordered communication node-pair, say (p, q), there is a flow of $\lambda^{(p,q)}$. Identifying the flows into a terminal node on the receiving side as a single commodity type, we have a total of N commodities, where N is the number of communication nodes. The notation $(\cdot, q)$ denotes the commodity destined for communication node q.

Let $x_{(i,j)}^{(\cdot,q)}$ be the flow of commodity $(\cdot, q)$ along the directed arc (i, j). If the effective channel capacity is 1 then without loss of generality, the aggregate flow

$$x_{(i,j)} = \sum_{\forall\ (\cdot,q)} x_{(i,j)}^{(\cdot,q)} \leq 1 \quad \forall\ (i,j). \tag{2.1}$$

Referring to Fig. 2.1, by necessity, the aggregate flow on each and every arc does not exceed 1 if the aggregate flow on every channel, or every Stage 2's arc, is constrained by 1. Therefore, generalizing the constraint to all the arcs as above does not change the problem.

In addition to Constraint (2.1), each commodity must satisfy the conservation of flows into and out of a node, that is,

$$\sum_j \left(x_{(i,j)}^{(\cdot,q)} - x_{(j,i)}^{(\cdot,q)}\right) = \begin{cases} \lambda^{(f_i^{-1}(i),q)} & \text{if } i = \text{source node of commodity } (\cdot,q); \\ -\sum_p \lambda^{(p,q)} & \text{if } i = \text{sink node of commodity } (\cdot,q); \\ 0 & \text{otherwise.} \end{cases} \quad (2.2)$$

Let $\mathbf{x}^{(\cdot,q)} = (\, x_{(i,j)}^{(\cdot,q)} \,)$ (i.e. a vector consisting of all the arc flows of commodity $(\cdot,q)$) and let $\mathbf{A}$ be the node-arc incidence matrix of the network and $\mathbf{d}^{(\cdot,q)}$ be the vector obtained from the right-hand side of (2.2). Finally, let $\mathbf{e} = (1 \quad 1 \quad \ldots)^T$ and N be the number of nodes in the communication network. We can write the above constraints in vector form as

$$\begin{aligned} \mathbf{I}\mathbf{x}^{(\cdot,1)} + \mathbf{I}\mathbf{x}^{(\cdot,2)} + \ldots + \mathbf{I}\mathbf{x}^{(\cdot,N)} + \mathbf{I}\mathbf{s} &= \mathbf{e} \\ \mathbf{A}\mathbf{x}^{(\cdot,1)} \qquad\qquad\qquad\qquad &= \mathbf{d}^{(\cdot,1)} \\ \mathbf{A}\mathbf{x}^{(\cdot,2)} \qquad\qquad\quad &= \mathbf{d}^{(\cdot,2)} \\ \ddots \qquad\qquad &\;\;\vdots \\ \mathbf{A}\mathbf{x}^{(\cdot,N)} \qquad &= \mathbf{d}^{(\cdot,N)} \\ \mathbf{x}^{(\cdot,q)} \geq 0 \quad \forall\, (\cdot,q). & \end{aligned} \quad (2.3)$$

where $\mathbf{s}$ is a vector of slack variables and $\mathbf{I}$ is the unit matrix. We wish to minimize the total cost,

$$c(\mathbf{x}) = \sum_{(i,j)\in S_1} v_t(x_{(i,j)}) + \sum_{(i,j)\in S_3} v_r(x_{(i,j)}) + \sum_{(i,j)\in S_2} w(x_{(i,j)}) \quad (2.4)$$

where

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^{(\cdot,1)} + \mathbf{x}^{(\cdot,2)} + \ldots + \mathbf{x}^{(\cdot,N)}, \\ x_{(i,j)} &= x_{(i,j)}^{(\cdot,1)} + x_{(i,j)}^{(\cdot,2)} + \ldots + x_{(i,j)}^{(\cdot,N)}, \\ S_n &= \text{the set of arcs belonging to stage } n \text{ in Fig. 2.1.} \end{aligned}$$

If the cost is linear, then the structures in (2.3) and (2.4) suggest that the linear program can be tackled by the Dantzig-Wolfe Decomposition Method.[7,10,30]

Here, we are interested in situations where $v_t(x_{(i,j)})$, $v_r(x_{(i,j)})$ and $w(x_{(i,j)})$ are step functions, as depicted in Fig. 1.2. The problem can be transformed into a Mixed Integer Linear Program (MILP) [7,9,30] that can be solved by the branch-and-bound method in which a sequence of LP's is solved [27]. For such a formulation, we introduce an integer quantity $0 \leq y_{(i,j)} \leq 1$ on arc (i,j): $y_{(i,j)}$ is 1 if and only if $x_{(i,j)} \geq 1$. This is equivalent to the constraint

$$y_{(i,j)} \geq x_{(i,j)}. \tag{2.5}$$

$x_{(i,j)} > 0$ implies $y_{(i,j)} = 1$ since $x_{(i,j)}$ cannot be greater than 1 and $y_{(i,j)}$ must be an integer value. The converse, $x_{(i,j)} = 0$ implies $y_{(i,j)} = 0$ is guaranteed by considering the new cost function,

$$c(\mathbf{x},\mathbf{y}) = \sum_{(i,j)\in S_1} v_t y_{(i,j)} + \sum_{(i,j)\in S_3} v_r y_{(i,j)} + \sum_{(i,j)\in S_2} w y_{(i,j)} \tag{2.6}$$

where $\mathbf{y} = (y_{(i,j)})$; if $x_{(i,j)} = 0$ then to minimize cost, $y_{(i,j)} = 0$ since $v_t, v_r, w \geq 0$.

2.2 Size of Problem

Let's consider the size of the first relaxed LP in the branch-and-bound method of MILP. As written in (2.3), there are many redundant variables and constraints which must be excluded from analysis if we are only interested in the "inherent" size of the problem. For instance, in Fig. 2.1, the traffic terminating in node m does not flow through node l and arc (k,l) at all. This is not reflected in (2.3) since there we unnecessarily include the variables $x_{(k,l)}^{(\cdot,1)}$ and use the node-arc incidence matrix of the whole network to describe the conservation of flows into the nodes. To calculate the total number of non-redundant constraints, we note from (2.1) that there are altogether $|C|(2N+1)$ arc flow constraints. From (2.2), there are $N(2|C|+N+1)$ node conservation equations since each commodity may pass through at most $2|C|+$

$N+1$ nodes. The integer variables $\mathbf{y}$ result in another $4N|C|+2|C|$ constraints (i.e. twice the number of arcs in the multicommodity network since $x_{(i,j)} \leq y_{(i,j)} \leq 1$). Hence, the total number of constraints, other than the nonnegativity constraints of $\mathbf{x}$ and $\mathbf{y}$, and the integer constraints of $\mathbf{y}$, is

$$N_{con} = |C|(8N+3) + N(N+1). \tag{2.7}$$

Enumerating the variables in a similar way, we get

$$N_{var} = |C|(N^2+4N+1). \tag{2.8}$$

Here, N_{var} does not include the slack variables required to put the problem in standard form. From above, we see that even with a CAP of moderate size, the dimensions of the resulting mathematical program are rather large. For instance, $N = 100$ and $|C| = 20$ results in a problem with $N_{con} = 26160$ and $N_{var} = 208020$. Coupled with the fact that this is an MILP, it is clear that the problem as formulated is quite formidable.

We also lack a closed form solution, and consequently, it is difficult to study the relationships and the trade-offs between the different parameters. Furthermore, the mathematical formulation may not represent the physical situation precisely. It could be that the costs cannot be conveniently expressed as simple functions, or even worse they may not be known exactly.

In Chapter 4, we will investigate CAP through heuristic approaches. Although the resulting solutions may not be optimal according to the criteria defined in this chapter, the steps taken toward finding a good solution give us some insights into the "feasible region" of the problem. This in turn allows us to study the trade-offs between the system parameters.

Chapter 3

Intractability of CAP

This chapter addresses the question of whether CAP is inherently difficult. In particular, we would like to relate CAP to the class of NP-complete combinatorial problems [30]. A crucial question is whether formulating CAP as a mixed integer linear program as in the preceding chapter is unnecessarily complicating the problem, since MILP is NP-hard. Is it possible that the symmetry of CAP, as shown by the corresponding multicommodity flow network, lends itself to easier algorithms? It will be shown that CAP is indeed intractable in terms of its algorithmic complexity.

3.1 Relationship between CAP and Other Problems

It is clear that CAP cannot be formulated as a convex nonlinear program because of the step cost functions. Furthermore, because of the drastic discontinuity of the cost functions, one cannot even hope to approximate the solution by solving, say, several easier linear programs. Hence, looking at it from the *traditional* nonlinear programming's point of view may not be very fruitful. On the other hand, the general CAP is not strictly combinatorial in nature since there are uncountably infinite number of feasible solutions. For instance, a new solution can be obtained by exchanging infinitesimal amounts of traffic between two channels. It is therefore difficult to relate this problem with known combinatorial optimization problems directly. Nevertheless, two different directions can be explored:

1. Study a restricted but combinatorial version of CAP. Specifically, we can impose an additional requirement that $\lambda^{(p,q)}$ for each (p,q) must be carried on one and only one channel in its entirety. An assumption here is that $\lambda^{(p,q)} < 1$, and this will be justified later in this chapter. This version of CAP will be called Non-Splitting Traffic CAP (NSTCAP) and the original version Splitting Traffic CAP (STCAP). Since any solution to an instance of NSTCAP is also a solution to the corresponding STCAP version, the optimal cost of the STCAP is bounded above by that of the NSTCAP. A criticism about this approach is that the optimal solution of one may not be readily transformed to that of the other. Nevertheless, establishing the intractability of NSTCAP may give us some insights into the complexity of STCAP. Furthermore, a good solution to one version may be modified to yield a good solution to the other.

2. Attempt to combinatorialize the problem by concentrating only on a subset consisting of a finite or countably infinite number of feasible solutions. It is known that a linear program in general may have an infinite number of solutions. However, if the program is feasible and bounded, there is an optimal solution among the finite number of basic feasible solutions. In this respect, LP can be considered as a combinatorial problem. A key question is whether there is an analogous situation in CAP.

The author has not found a finite or countably infinite subset of feasible solutions which is guaranteed to have an optimal solution. Therefore, this chapter will be devoted mainly to Approach 1, as far as a concrete statement about the intractability of CAP is concerned.

There are a few properties that can be identified with special cases of STCAP and they are stated as theorems in the next section.

3.2 Basic Properties of CAP and False Conjectures

If only one of the step costs is nonzero, an optimal solution can be found trivially.

Theorem 3.1 *There is a feasible solution to CAP such that* $N_c = \lceil \lambda \rceil$ *where* $\lambda = \sum_{(p,q)} \lambda^{(p,q)}$.

Comment: Since the total system traffic is λ and the capacity of a channel is 1, the minimum number of channels required is bounded from below by $N_c = \lceil \lambda \rceil$. The above theorem states that this bound can be achieved and therefore the minimum number of channels required depends only on the total traffic volume. Furthermore, if $v_t, v_r = 0$ and $w > 0$, the solution is also optimal.

Proof: A trivial solution that achieves the bound is to simply let all the nodes transmit and receive on all the $\lceil \lambda \rceil$ channels. For each traffic type, say (p, q), a fraction of $\lambda^{(p,q)}/\lceil \lambda \rceil$ traffic volume is assigned to each and every one of the channels.

□

Theorem 3.2 *There is a feasible solution to CAP with* $N_t = \sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$.

Comment: Since the total traffic originating from node p is $\sum_q \lambda^{(p,q)}$, the minimum number of transmitters, or transmittability, node p requires is $\lceil \sum_q \lambda^{(p,q)} \rceil$ because each transmitter can output a unit of traffic at most. The above theorem simply states that the lower bound of the overall system can be achieved. The solution is optimal if $w, v_r = 0, v_t > 0$.

Proof: A trivial solution is to assign $\lceil \sum_q \lambda^{(p,q)} \rceil$ exclusive channels to the traffic originating from each node p. A fraction of $\lambda^{(p,q)} / \lceil \sum_q \lambda^{(p,q)} \rceil$ of each traffic (p,q) is assigned to each and every one of the channel. All nodes can receive on all channels in the network.

□

Theorem 3.3 *There is a feasible solution to CAP such that* $N_r = \sum_q \lceil \sum_p \lambda^{(p,q)} \rceil$.

Proof: By symmetry, similar to the previous proof.

□

Corollary 3.1 *From* **Theorems 3.1, 3.2** *and* **3.3,** $\min N_t, \min N_r \geq \min N_c$.

The above proofs construct trivial solutions. These solutions are not unique in achieving the lower bounds. For example, it is possible to reduce N_r while maintaining the values of N_t in the proof of Theorem 3.2, as will be shown in the next chapter.

Theorem 3.4 *There is an optimal solution to CAP such that given any pair of traffic types, say* (p,q) *and* (r,s), *they co-exist on at most one channel.*

Proof: Suppose traffic (p,q) and traffic (r,s) co-exist on channel i and channel j. Let $\rho_i \lambda^{(p,q)}$ $(0 < \rho_i < 1)$ and $\sigma_i \lambda^{(r,s)}$ $(0 < \sigma_i < 1)$ be the fractions of traffic (p,q) and traffic (r,s) assigned to channel i respectively. Without loss of generality, assume $\rho_i \lambda^{(p,q)} \leq (1 - \sigma_i) \lambda^{(r,s)}$ (i.e. traffic (r,s) on channel j is equal to or greater than traffic (p,q) on channel i). To obtain a potentially better solution, shift the traffic flow $\rho_i\lambda^{(p,q)}$ from channel i to channel j and in its place, an amount of $\rho_i\lambda^{(p,q)}$ of traffic (r,s) is moved from channel j to channel i. N_c remains the same, N_t decreases if there is no traffic originating from node p remaining on channel i after the above operation (i.e. (p,q) is the only traffic of type $(p,\cdot)$ on channel i in the beginning), and similarly N_r decreases if there is no traffic terminating in node q remaining on channel i. In all cases, the system cost does not increase. Performing the above procedure over all traffic types co-existing on more than one channel, we obtain a solution which is no worse than what we start out with.

□

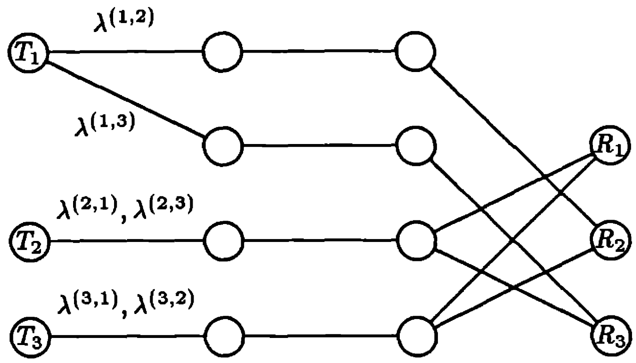
Theorem 3.4 narrows down the "interesting" solution space somewhat. However, the number of feasible solutions in this solution space is still uncountably infinite. Furthermore, it is not clear how we can devise a scheme for improving a solution within this solution space. The following conjectures attempt to narrow down the interesting solution space further, but they are false.

False Conjecture 3.1 *If $\lambda^{(p,q)} < 1$ for all (p,q) then there exists an optimal solution such that $\lambda^{(p,q)}$ for each (p,q) is carried on one and only one channel*

Comment: This conjecture essentially says that solving STCAP is equivalent to solving NSTCAP. It is not surprising that it is not true.

Counterexample: Suppose $v_t, v_r = 0$ and $w > 0$. Consider the case where there are three nodes and traffic from one node to another is uniform with value $2/3$. The objective is to minimize the number of channels used. It is $2/3 \times 6 = 4$ from Theorem 3.1. Without traffic splitting, 6 channels are required.

□



Unoccupied arcs are not shown, i.e. the number of arcs on the transmitting side is the transmittability and the number of arcs on the receiving side is the receivability.

Figure 3.1: Illustration of the Counterexample to False Conjecture 3.2.

False Conjecture 3.2 *There exists an optimal solution such that* $N_c = \lceil \lambda \rceil$.

Comment: The above attempts to strengthen the statement in Theorem 3.1.

Counterexample: Consider a network consisting of three nodes and having cost functions, $v_r, w = 0$ and $v_t > 0$. In addition, the traffic flows are

$$\lambda^{(1,2)} = \tfrac{2}{3} \quad \lambda^{(2,1)} = \tfrac{1}{3},$$
$$\lambda^{(1,3)} = \tfrac{3}{4} \quad \lambda^{(3,1)} = \tfrac{1}{4},$$
$$\lambda^{(2,3)} = \tfrac{1}{2} \quad \lambda^{(3,2)} = \tfrac{1}{2}.$$

An optimal solution is one with the minimum transmittability, and from Theorem 3.2 this is 4. An optimal solution is given in Fig. 3.1. It is a routine exercise to verify that in order to put all the traffic onto $\lceil \lambda \rceil = 3$ channels, the transmittability must increase.

□

Definition 3.1 Consider instances of CAP where there exists (p, q) such that $\lambda^{(p,q)} \geq 1$. It is natural to propose the following ***residual method*** for solving the problems: For every (p, q) where $\lambda^{(p,q)} \geq 1$, $n_r^{(p,q)} = \lfloor \lambda^{(p,q)} \rfloor$ channels are assigned exclusively to this traffic. Consequently, the residual traffic, $\lambda'^{(p,q)} = \lambda^{(p,q)} - \lfloor \lambda^{(p,q)} \rfloor$, is smaller than 1. We may then proceed with solving a ***residual CAP*** in which $\lambda'^{(p,q)} < 1$ for all (p, q). The ***best*** solution obtained with this method will be called a ***residual solution.***

Using the residual method is the "mathematical" justification for assuming $\lambda^{(p,q)} < 1$ in CAP. In the following, we will argue that this is a good method. However, it may not yield an optimal solution in general (see False Conjecture 3.3).

Definition 3.2 In a ***pseudo-residual method***, for every (p, q) where $\lambda^{(p,q)} \geq 1$, $n^{(p,q)} \leq n_r^{(p,q)}$ channels are "filled" up exclusively by traffic (p, q) (i.e. the number of exclusive channels may not equal to $n_r^{(p,q)}$). We then proceed to solve the ***pseudo-residual CAP*** based on the residual traffic, $\lambda'^{(p,q)} = \lambda^{(p,q)} - n^{(p,q)}$, with the requirement that no additional channel be entirely filled up by a single type of traffic (i.e. $x_k^{(p,q)} < 1$ for every additional channel k). The ***best*** solution obtained with this restriction is called a ***pseudo-residual solution.***

Definition 3.3 Pseudo-Residual Solution(CAP) A is of a higher order than Pseudo-Residual Solution(CAP) B if there exists (p, q), $\lambda^{(p,q)} \geq 1$ such that $n_A^{(p,q)} > n_B^{(p,q)}$ (i.e. $n^{(p,q)}$ of A is greater than that of B) and $n_A^{(i,j)} \geq n_B^{(i,j)}$ for all $(i, j) \neq (p, q)$ and $\lambda^{(i,j)} \geq 1$.

From the above definition, it is clear that the highest-order pseudo-residual solution is the residual solution. Also, two solutions do not necessarily have any natural ordering, e.g. $n_A^{(p,q)} > n_B^{(p,q)}, n_A^{(r,s)} < n_B^{(r,s)}$; this is not essential to the following discussion.

In principle, the original CAP can be tackled by a decomposition scheme in which each and every one of the possible pseudo-residual CAP is solved. From the above definitions, this covers all the solution space of the original CAP. Thus, choosing the best pseudo-residual solution among all of the solutions yields an optimal solution to the original CAP. The number of pseudo-residual CAP's to be solved is

$$\prod_{\{(p,q):\lambda^{(p,q)}\geq 1\}} (n_r^{(p,q)} + 1). \tag{3.1}$$

Note that $n^{(p,q)}$ can be zero and therefore there are $n_r^{(p,q)} + 1$ different possible values for it. Also, the total number of different pseudo-residual CAP's may be prohibitively large considering the fact that there is a maximum of $N(N-1)$ terms in the above product. As a result, it would be nice if we could establish some result to the effect that it is not essential to solve all the pseudo-residual CAP's. Let's guess that a residual solution is an optimal solution to the original CAP and see if there is any intuitive argument supporting this conjecture. The following theorem narrows down the interesting solution space within each pseudo-residual CAP when the decomposition scheme is used to solve the original CAP.

Theorem 3.5 *There is an optimal solution to CAP such that if* $0 < x_k^{(p,q)}, x_j^{(p,q)} < 1$, $j \neq k$, *then* $x_k^{(p,q)} + x_j^{(p,q)} < 1$.

Proof: Suppose there is a solution in which $x_k^{(p,q)} + x_j^{(p,q)} \geq 1$ and without loss of generality assume $x_k^{(p,q)} \geq x_j^{(p,q)}$. We may attempt to improve the solution as follows: Transfer an amount $(1 - x_k^{(p,q)})$ of traffic (p,q) from channel j to channel k and in its place, move all traffic other than (p,q) from channel k to channel j. This can be done because by hypothesis, $1 - x_k^{(p,q)} \leq x_j^{(p,q)}$. It is easy to check that N_t, N_r and N_c

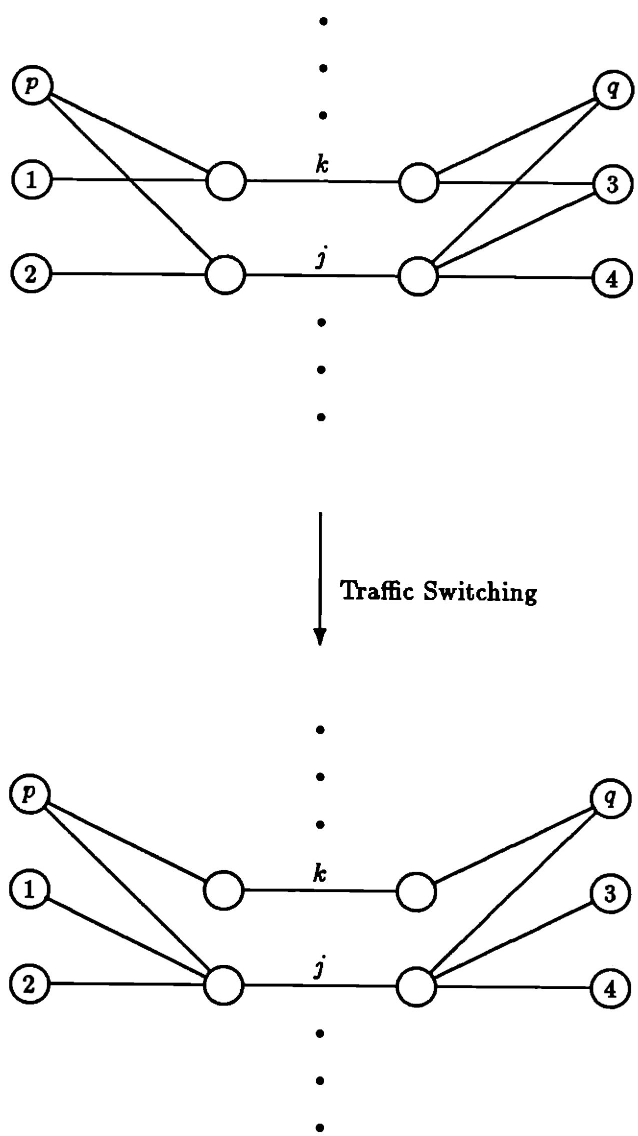
do not increase after the traffic switching (see Fig. 3.2 for an example). Performing the switching for all $k, j, (p,q)$ such that $x_k^{(p,q)} + x_j^{(p,q)} \geq 1$, we obtain a solution no worse than the original one. Notice that $x_k^{(p,q)} = 1$ in the end, and channel k is an exclusive channel devoted to traffic (p,q).

□

Theorem 3.5 implies that if there is a pseudo-residual CAP that contains a solution optimal with respect to the original CAP, then this pseudo-residual CAP has a pseudo-residual solution such that there is at most one channel k such that $1 > x_k^{(p,q)} \geq 1/2$. This immediately suggests that the amount of traffic splitting is large if this is a pseudo-residual CAP of a lower order since the residual traffic is large. Consider the implications for our decomposition scheme. In solving a particular pseudo-residual CAP, we only need to concentrate on solutions that satisfy the condition in Theorem 3.5. Consider the residual traffic $\lambda'^{(p,q)} = \lambda^{(p,q)} - n^{(p,q)}$, which increases as $n^{(p,q)}$ decreases. Accordingly, because of the inequality $x_k^{(p,q)} + x_j^{(p,q)} < 1$, $\lambda'^{(p,q)}$ is carried on at least $2\lfloor \lambda'^{(p,q)} \rfloor + 1$. As a rule-of-thumb, traffic splitting tends to increase N_t and N_r since this increases the number of channels on which a single traffic type is carried. Without further information about the traffic distribution, we conclude that it is probably unlikely to find a pseudo-residual solution that is better than all the higher-order pseudo-residual solutions, assuming v_t and v_r are not negligible relative to w. The steps of the argument are summarized as follows:

1. There is a pseudo-residual CAP containing an optimal solution such that the residual traffic is split at least $2\lfloor \lambda'^{(p,q)} \rfloor + 1$ ways. We only need to look for solutions of this type when *sweeping* through the different pseudo-residual CAP's in our decomposition scheme.

2. Traffic splitting, heuristically, increases N_t and N_r.



Switch as much traffic (p, q) as possible from channel j to channel k. Traffic other than (p, q) on channel k is "pushed" onto channel j.

Figure 3.2: Illustration of the Proof of Theorem 3.5

3. In looking for this solution, low-order pseudo-residual's traffic is split more. Therefore, it is unlikely that a low-order pseudo-residual solution is an optimal solution or a good solution to the original CAP.

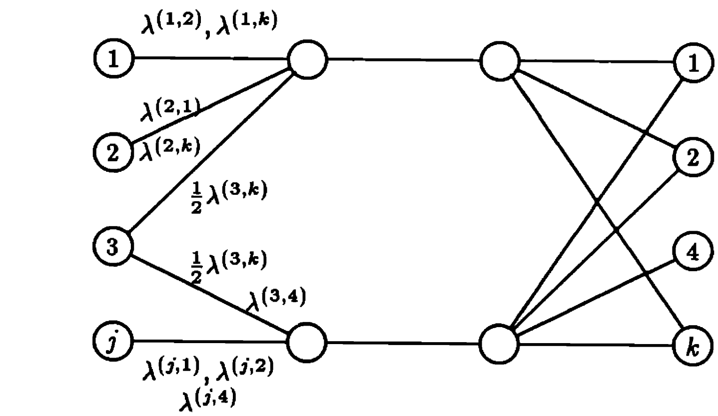
The above argument implies that it is quite likely that there is an optimal residual solution. Even if a residual solution is not optimal, it is likely to be good, based on the procedure in the proof of Theorem 3.5. Unfortunately, this statement is not concrete and we cannot extend our conclusion further to state that the residual CAP yields an optimal CAP. The steps in our argument are precise except the statement: "*Traffic Splitting suggests large values for N_t and N_r,*" which is not always true as attested by the counterexample to the following false conjecture.

False Conjecture 3.3 *Suppose there exists (p,q) such that $\lambda^{(p,q)} \geq 1$. The residual solution is optimal with respect to the original CAP.*

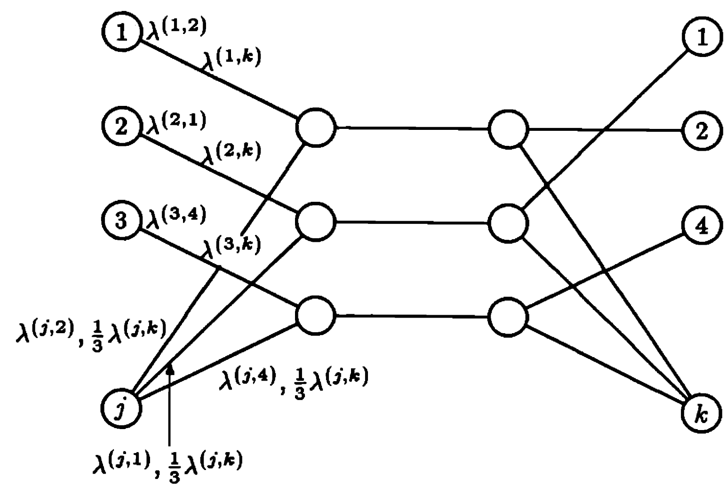
Counterexample: Consider a network with N communication nodes and the sizes of the cost functions are lopsided, $w \gg v_t \gg v_r > 0$. $\lambda^{(p,q)} = 1$ for all $p \neq q$ except the following

$$\begin{array}{llll} & \lambda^{(1,2)} = \frac{2}{9} & & \lambda^{(1,k)} = \frac{2}{9}, \\ \lambda^{(2,1)} = \frac{2}{9} & & & \lambda^{(2,k)} = \frac{2}{9}, \\ & & \lambda^{(3,4)} = \frac{2}{9} & \lambda^{(3,k)} = \frac{2}{9}, \\ \lambda^{(j,1)} = \frac{2}{9} & \lambda^{(j,2)} = \frac{2}{9} & \lambda^{(j,4)} = \frac{2}{9}. & \end{array}$$

For the residual method, each traffic (p,q) with $\lambda^{(p,q)} = 1$ is assigned an exclusive channel in the beginning. Fig. 3.3(a) shows a solution after solving the residual CAP and Fig. 3.3(b) is a feasible solution of a lower-order pseudo-residual CAP which is better than the residual solution. To show that the solution given in Fig. 3.3(a) is indeed the residual solution (i.e. the best solution of the residual CAP), we note



(a)



(b)

Figure 3.3: Illustration of the Counterexample to False Conjecture 3.3.

from the lopsided cost functions that the optimization problem can be tackled as follows. First we set the number of channels at its minimum. In this case, it is

$$\sum_{\{(p,q):\lambda^{(p,q)}<1\}} \lambda^{(p,q)} = 2. \tag{3.2}$$

Now, we find the minimum transmittability given that the number of channels is 2. By Theorem 3.2, the minimum transmittability without any restriction is 4, each source node having a transmittability of 1. However, this can not be done with 2 channels since the values

$$\begin{aligned} \sum_{\{(1,q):\lambda^{(1,q)}<1\}} \lambda^{(1,q)} &= \frac{4}{9} \\ \sum_{\{(2,q):\lambda^{(2,q)}<1\}} \lambda^{(2,q)} &= \frac{4}{9} \\ \sum_{\{(3,q):\lambda^{(3,q)}<1\}} \lambda^{(3,q)} &= \frac{4}{9} \\ \sum_{\{(j,q):\lambda^{(j,q)}<1\}} \lambda^{(j,q)} &= \frac{6}{9} \end{aligned} \tag{3.3}$$

cannot be partitioned into two sets, the elements of each summing up to 1. Hence, a transmittability of at least 5 is needed with only 2 channels. Next, we find the minimum receivability given that the number of channels is 2 and the transmittability is 5. It remains a routine exercise to exhaustively enumerate all the possibilities and conclude that the minimum receivability with this restriction is indeed 7. Combining the parameters of the residual solution with the parameters due to the exclusive channel assignment in the beginning yields

$$N_c^{(a)} = N(N-1) - 7, \quad N_t^{(a)} = N(N-1) - 4, \quad N_r^{(a)} = N(N-1) - 2 \tag{3.4}$$

The solution in Fig. 3.3(b) is a feasible solution of the pseudo-residual CAP of a lower order than the residual CAP. In this case, $n^{(j,k)}$ is 0 rather than 1 and therefore there is one fewer channel assigned in the beginning. Note that $\lambda^{(j,k)}$ is split equally 3 ways and assigned to all the 3 channels. Furthermore, there is no traffic splitting other than this. It is easy to derive the resulting system parameters by combining the result here with the result due to the assignment in the beginning.

$$N_c^{(b)} = N(N-1) - 7, \quad N_t^{(b)} = N(N-1) - 4, \quad N_r^{(b)} = N(N-1) - 4 \qquad (3.5)$$

Comparing (3.5) with (3.4), we see that the receivability here is 2 units lower.

□

Despite the counterexample, for the rest of this report, we will assume the residual method is employed in cases of CAP where there is (p, q) where $\lambda^{(p,q)} \geq 1$. Hence, an implicit assumption throughout is that $\lambda^{(p,q)} < 1$ for all (p, q). The justifications for this approach are listed as follows:

- The argument before False Conjecture 3.3 suggests that a reasonably good solution can be obtained using the residual method. As will be seen, CAP is computationally intractable and if a heuristic scheme is to be used, it does not make sense to insist on exact optimality at this stage.

- The counterexample is artificially contrived to illustrate that the residual method does not yield an optimum solution in general. Specifically, the traffic distribution is such that there is an "exact fit" of the traffic into the channels in solution of Fig. 3.3(b). In practice, it is "improbable" such a situation will occur.

- The number of pseudo-residual CAP's to be solved may simply be formidable, as suggested by (3.1), even if individual pseudo-residual CAP's could be solved easily. As we will see, solving the residual CAP is itself a difficult problem.

- In practice, there may be motivations to devote exclusive channels to a single type of traffic, should the traffic volume justifies it. The exclusive channels are essentially point-to-point channels and this tends to simplify the lower-level control issues.

Based on the last reason above, we may in fact argue that CAP is not a very meaningful or significant problem if $\lambda^{(p,q)} \gg 1$ for a large majority of (p, q) since the "optimizable part" of the total cost is negligible with respect to the total cost, and we might as well consider a network in which the capacity is partitioned into many point-to-point links. It will be shown in the next chapter that when $\lambda^{(p,q)} \ll 1$ for all (p, q), very good heuristic solutions can be obtained. Two intuitive reasons why this is so are:

1. Each traffic type occupies only a small part of the total available capacity on a channel and we do not have to split too many traffic types to create a "good fit". This implies small N_t and N_r.

2. The problem of fitting traffic into the channels become easy and very tight fitting, in which the unassigned capacity on any channel is very small, can be achieved. This suggests small N_c.

The difficult instances of CAP's lie between the two extremes described above. Specifically, an instance of CAP tends to be difficult the if the values of $\lambda^{(p,q)}$ are distributed somewhat uniformly between 0 and 1.

3.3 CAP is NP-Hard

In this section, we show that CAP in general is an intractable problem. First, we will show that NSTCAP is intractable by showing special cases of it are equivalent to the Bin Packing Problem (BPP), which is NP-complete. Since NP-completeness theory deals with the recognition versions of optimization problems, the following definitions of the recognition versions of CAP and BPP are given. It can be shown that for a large number of combinatorial problems, the recognition versions and the optimization versions are equivalent as far as their complexities are concerned [30].

Definition 3.4 [Recognition Version of CAP] Given the traffic matrix $(\lambda^{(p,q)})$ and three integers , $\overline{N_t}, \overline{N_r}$ and $\overline{N_c}$, , is there a traffic assignment such that $N_t \leq \overline{N_t}, N_r \leq \overline{N_r}$ and $N_c \leq \overline{N_c}$?

Definition 3.5 [Recognition Version of BPP] Given an integer m, and n rational numbers, $\lambda_1, \lambda_2 \cdots \lambda_n$; $\lambda_k \leq 1$ for all k, can these numbers be partitioned into m sets, the elements of each summing up to a value not greater than 1?

For the optimization version of BPP, the objective is to minimize the total number of partitions.

Theorem 3.6 *NSTCAP is NP-complete.*

Proof: Many special cases of NSTCAP are equivalent to BPP. For simplicity, consider a special case of NSTCAP in which $v_r, w = 0$ and $v_t > 0$ and $\lambda^{(p,q)} = 0$ for all (p, q) except when $p = i$ [1]. The NSTCAP is equivalent to the bin packing problem with $\lambda_k = \lambda^{(i,k)}$ for all $1 \leq k \leq n$. In particular, $n = N - 1$, $m = \overline{N_t}$, $\overline{N_r} = \infty$ and $\overline{N_c} = \infty$ and the traffic types belonging to the same bin are assigned to the same channel.

[1]Alternatively, $\sum_q \lambda^{(p,q)} = 1$ for all p except $p = i$ so that $\sum_q \lambda^{(p,q)}$ fits exactly onto 1 channel.

To show that the general CAP is intractable, we introduce the following Splitting Bin Packing Problem (SBPP).

Definition 3.6 [Recognition Version of SBPP] Given an integer v, and n rational numbers, $\lambda_1, \lambda_2 \cdots \lambda_n$; $\lambda_k \leq 1$ for all k. Suppose each λ_k is allowed to split into v_k numbers such that the sum of the numbers, $\sum_{i=1}^{v_k} \lambda_{k_i} = \lambda_k$. Is there a way of splitting the original numbers such that the total number of splittings, $\sum_k v_k \leq v$ and the resulting numbers can be "bin-packed" into $\lceil \sum_k \lambda_k \rceil$ sets?

For the optimization version, the objective is to minimize the total number of splittings. Intuitively, if we can obtain a good solution to an instance of the optimization version BPP, it is easy to obtain a good solution to the corresponding version of SBPP and vice versa. For example, from a solution to SBPP, we can obtain a solution to the BPP version by considering the packing problem of only the numbers that have been split up. There should be only a few of such numbers if the solution is good. An important point to note is that if $v = n$ and $m = \lceil \sum_k \lambda_k \rceil$ then SBPP and BPP are equivalent. In other words, the question of whether there is a packing with no splitting in SBPP is the same question as whether there is a packing achieving the lower bound on the number of bins in BPP. Hence, intuitively, SBPP is as difficult as BPP. In the following, the NP-hardness of SBPP is established through the transformation of Set Partition Problem (SPP), an NP-complete problem, to a special case of SBPP. That SBPP is not in the NP Class is due to the fact that there may not be a *concise* certificate given a yes instance to SBPP [30]. More precisely, splitting the input rational numbers may result in real numbers that cannot be represented concisely, i.e. an infinite number of bits is required to represent a general real number. This can be viewed as a consequence of SBPP not being a strictly combinatorial problem.

Definition 3.7 [SPP] Let $\{a_1, a_2, \cdots, a_n\}$ be a set consisting of n rational numbers. Is there a way of partitioning the set into two subsets, S_1 and S_2, such that $\sum_{a_i \in S_1} a_i = \sum_{a_i \in S_2} a_i$?

Lemma 3.1 *SBPP is NP-hard.*

Proof: SPP can be transformed to an instance of SBPP as follows: $\lambda_i = 2a_i / \sum_i a_i$, $1 \le i \le n$. Clearly, there is a solution with no splitting iff there is a solution to SPP.

□

Theorem 3.7 *CAP is NP-hard.*

Proof: Consider the special case where $v_t, w \gg v_r > 0$ and $\lambda^{(p,q)} = 0$ for all (p, q) except when $p = i$. Let's consider first the implication of this on the optimization version of CAP. The way to solve the problem is to minimize on N_t and N_c and then find the lowest value of N_r for which the minimum values of N_t and N_c are attained. From Theorem 3.2 and 3.1, the minimum values of N_t and N_c are $\lceil \sum_q \lambda^{(i,q)} \rceil$. The way to minimize N_r is to minimize the traffic splitting required to achieve the minimum N_t and N_c. For example, carrying $\lambda^{(i,1)}$ over 3 channels means node 1 must have a receivability of 3 to access all of the 3 channels (see Fig. 3.4). It is therefore desired to avoid traffic splitting. Next, consider the recognition version where $\overline{N_t}$ and $\overline{N_c}$ are $\lceil \sum_q \lambda^{(i,q)} \rceil$ and $\overline{N_r}$ is finite. This is easily shown to be equivalent to SBPP, following an outline analogous to the proof of Theorem 3.6. Hence, CAP is NP-hard.

□

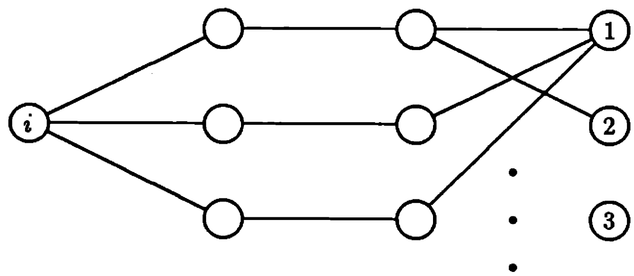


Figure 3.4: Illustration of Increase in N_r due to Splitting: Theorem 3.7

In the above proof, the special case of CAP considered is "relatively easy" because of the lopsided cost functions. This enables us to decompose the optimization process into a series of optimization problems; the minimization of N_t and N_c is followed by the minimization of N_r. Nonetheless, this special case is already NP-hard. Hence, one would expect the general CAP to be very difficult. Also, in this chapter, several attempts have been made to narrow down the the interesting solution space. These attempts fail in guaranteeing an optimal solution. One would suspect that some sort of exhaustive enumeration scheme would be needed to locate an optimal solution. Barring formulating the problem as a MILP, it is not clear how to proceed with the enumeration since the number of feasible solutions is uncountably infinite.

Given the substantial evidence that it is unlikely to find an exact and polynomial algorithm to CAP, let alone a closed form expression describing the interactions between the different parameters, heuristic approaches that yield suboptimal solutions will be explored in the next chapter. In addition, the trade-offs between the parameters will be studied based on the results of these approaches. Finally, before proceeding to the next chapter, it is worth pointing out that even though the conclusion of this chapter is a negative one, the results are nonetheless of theoretical interest. Furthermore, some of the results and frameworks in this chapter do give us some insights into how to approach this problem heuristically. In short, this chapter may be viewed as a stepping stone to the material in the next chapter.

Chapter 4

Heuristic Methods, Upper Bounds and Lower Bounds

In this chapter, we discuss two heuristic approaches to CAP. In addition, the fundamental trade-offs between the system parameters will also be investigated.

The first approach deals with situations in which the step sizes of the cost functions are lopsided; e.g. $w \gg v_r \gg v_t$. The method to find a good solution can be decomposed into several steps. For instance, if v_t is substantially greater than the two other costs, then, one possible approach is to first concentrate on solutions yielding small N_t. We then choose among these solutions the solutions that exhibit small N_c, assuming $w \gg v_r$. Finally, a solution with small N_r will be chosen. As will be seen, the result points to a strong trade-off between N_t and N_r and a weak dependence between N_c and these two parameters.

The trade-off between N_t and N_r will be studied more formally by setting up a minimization problem in which the objective function is N_tN_r and the constraints are a subset of the constraints of CAP. This gives us a lower bound on N_tN_r and reveals the trade-off between the two parameters.

The second approach, which addresses more general cases with arbitrary cost sizes, is based on identifying desirable mapping patterns of the traffic matrix, an $N \times N$ matrix in which entry (p, q) is $\lambda^{(p,q)}$. Here each entry is mapped onto one or more channels, and we want to identify mapping methods that yield good solutions. By addressing the uniform traffic case, an upper bound on the optimal cost will be derived using a simple heuristic. This is an upper bound because the optimal solution must have at least as good a solution as the heuristic used. We will also show that in the limit that the total system traffic is large but the individual traffic between two nodes is small, the ratio between the upper bound and the lower bound approaches 1. This implies our heuristic yields a very good solution in the asymptotic limit.

The n-connectivity CAP, in which any two nodes must have at least n alternative channels for communication, is studied at the end of this chapter.

4.1 Transmittability-Based Approach

In this approach, the main objective is to minimize N_t. The secondary objective is to find the smallest N_r and N_c possible, given that N_t must retain its minimum value, $\sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$.

Definition 4.1 A solution is ***transmittability-based*** if N_t achieves its lower bound, $\sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$.

There are many transmittability-based solutions to an instance of CAP, some achieving low values of N_r and N_c while others do not. The trivial solution in the proof of Theorem 3.2, for example, gives rise to a very large N_r. The transmittability-based solutions with small values of N_r and N_c will be loosely called *good* transmittability-based solutions.

From the discussions in the preceding chapter, we know that traffic splitting, as a rule-of-thumb, is likely to increase N_t and N_r. One may therefore think of first finding a good solution to the NSTCAP and then modify this solution to get an even better solution to the original CAP. Note that the solution to the NSTCAP is already a feasible and, by its non-splitting characteristic, potentially good solution to the original CAP.

A heuristic to finding a good transmittability-based solution is outlined in Fig. 4.1. First, we concentrate on achieving the minimum local transmittability $\lceil\sum_q \lambda^{(p,q)}\rceil$ for each node p separately. For each node p we solve a bin-packing problem with $\lambda_k = \lambda^{(p,k)}$. It is not necessary to solve the BPP optimally since by splitting some of the traffic later, we can always achieve $N_t = \sum_p\lceil\sum_q \lambda^{(p,q)}\rceil$. One may use some heuristic method, for example, the First Fit method or choose the best solution out of those obtained from a group of several methods, say First Fit, Best Fit, First Fit Decreasing and Best Fit Decreasing [16]. The better the solution found here, the less splitting is necessary later, and this leads to a smaller value of N_r. The next step is to achieve the minimum local transmittability by splitting the traffic originating from node p such that the number of splittings is minimized and the resulting traffic can be packed into the minimum number of channels. Again, the optimal way of doing this is not obvious and we leave out the exact specification of this part. For example, one may choose to split only the traffic among the least tightly-packed channels. In essence, first tackling the BPP and then considering the splitting problem, as in Steps (1) and (2) in Fig. 4.1 is really one way of approaching the underlying SBPP. One may also conceive of algorithms in which these two steps are integrated. After the SBPP for every node p has been solved to satisfaction, the resulting N_c can be further reduced by considering the BPP problem in which $\lambda_i = \sum_{(p,q)} x_i^{(p,q)}$, i.e. as long as their total traffic can be accommodated by only one

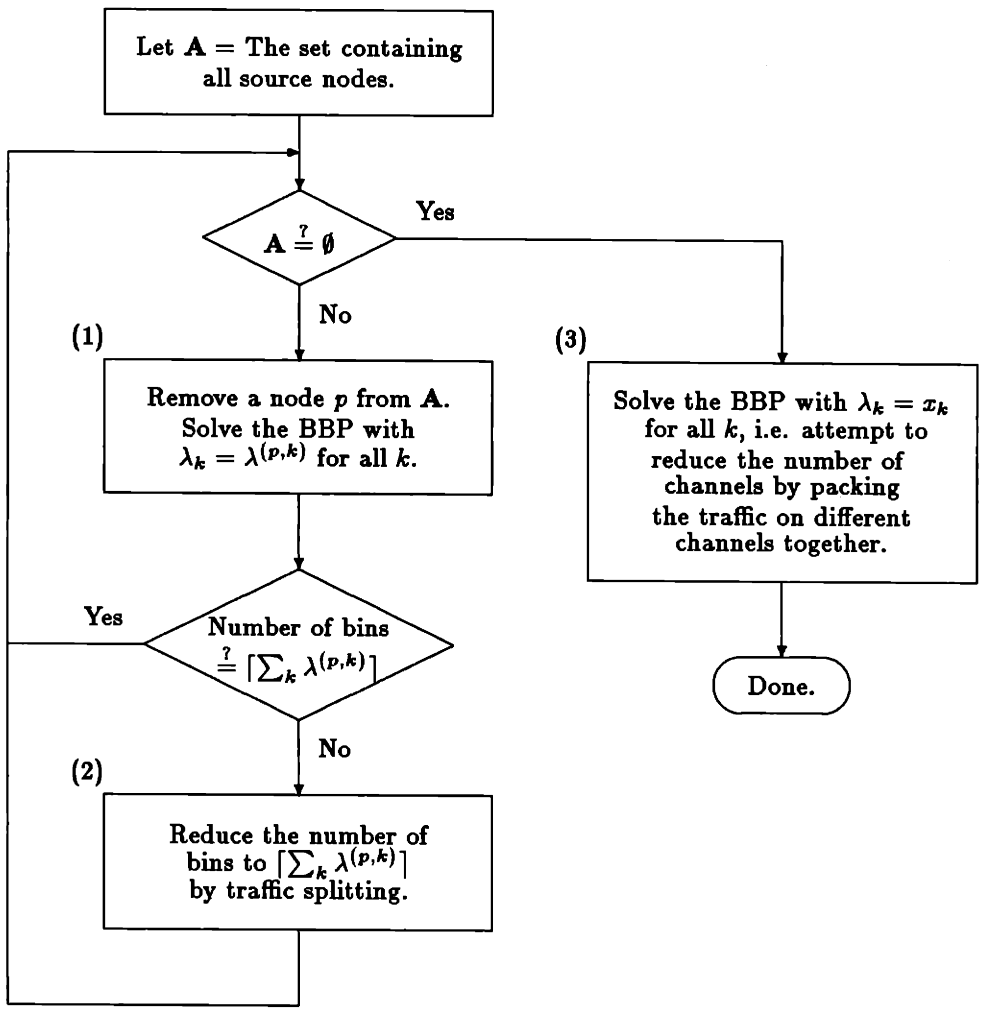


Figure 4.1: A Method for Finding Good Transittability-Based Solutions

	R_1	R_2	R_3	R_4	R_5
T_1	×	0.6	0.6	0.8	0.5
T_2	0.6	×	0.6	0.8	0.5
T_3	0.6	0.6	×	0.8	0.5
T_4	0.6	0.6	0.8	×	0.5
T_5	0.6	0.5	0.6	0.8	×

Entry (i, j) in the matrix is $\lambda^{(i,j)}$.

Figure 4.2: A Traffic Matrix for the Illustration of Finding a Good Transmittability-Based Solution

channel, we may condense or combine together traffic of several channels.

For illustration, consider the traffic matrix in Fig. 4.2. Step (1) is trivial; each of the source nodes needs 4 channels, or a transmittability of 4, since this is the tightest non-splitting packing. Consider node 1 in Step (2). Traffic $(1,4)$ can be divided into two equal fractions and assigned to the channels occupied by traffic $(1,2)$ and $(1,3)$. In this way, the local transmittability of node 1 is decreased by 1. Performing a similar procedure to all the other source nodes, the solution depicted in Fig. 4.3 is obtained. For Step (3), we note that the channels carrying traffic $(1,5), (2,5), (3,5), (4,5)$ and $(5,2)$ are only half full. So any two of these traffic types can be carried on a single channel alone. Combining $(1,5)$ with $(2,5)$ and $(3,5)$ with $(4,5)$, N_c and N_r each decreases by 2. Hence, in the resulting solution, $N_c = 13$, $N_t = 15$ and $N_r = 23$.

By symmetry, one may also obtain a "receivability-based" solution using an analogous method. From the results here alone, the trade-off between N_t and N_r can be seen. The intuitive reasoning is simple, and for illustration let's consider

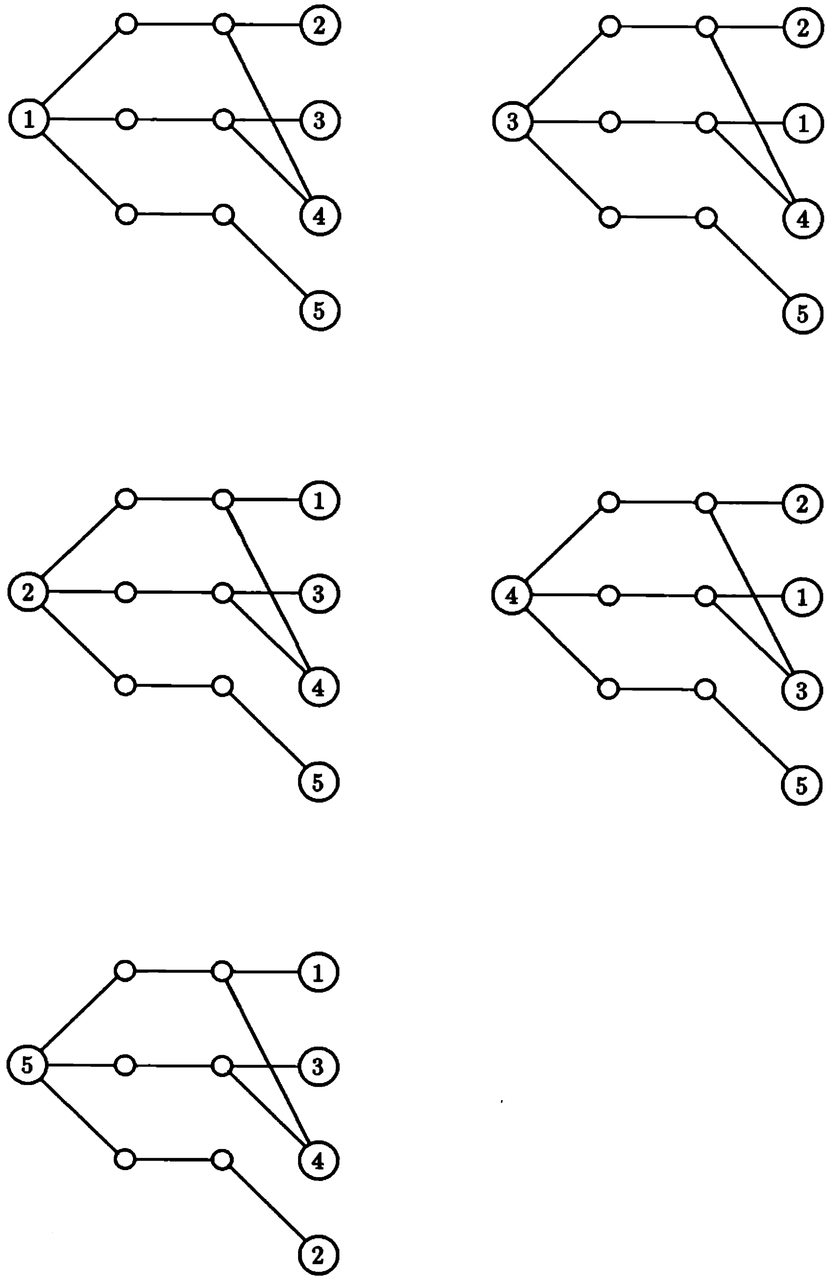


Figure 4.3: The Intermediate Solution after Steps (1) and (2) in Fig. 4.1

the traffic originating from node p, $\lambda^{(p,\cdot)} = \{\lambda^{(p,q)}$ for all $q\}$. In the transmittability-based approach, we want to concentrate this traffic on as few channels as possible. Ideally, we do not want to mix the traffic from a different source node onto the same channels unnecessarily as this may increase the local transmittability of node p; note that this traffic mixing is done in Step (3) because there it does not matter anymore as far as N_t is concerned. As a result, after Steps (1) and (2), each source node contributes at least $N-1$ to the resulting value of N_r. The contribution would be more if there is traffic splitting. This means N_r would be approximately $N(N-1)$ in the final solution and this could be significantly greater than the lower bound, $\sum_q \lceil \sum_p \lambda^{(p,q)} \rceil$, if $\lambda^{(p,q)}$ is small compared with 1 for all (p,q). The trade-off between N_t and N_r will be established more concretely in Section 4.3.

4.2 Channel-Based Approach

The aim here is to minimize N_c. After this, we can either optimize on N_t or N_r.

Definition 4.2 A *channel-based* solution is a solution in which N_c achieves its lower bound, $\lceil \sum_{(p,q)} \lambda^{(p,q)} \rceil$.

A channel-based solution with small N_t, as opposed to small N_r, can be obtained by modifying Step (3) of the algorithm for finding a transmittability-based solution. Instead of solving a BPP, the corresponding SBPP is considered. A crucial question is how splitting should be done. Again, we may tackle the SBPP by first solving the BPP and then splitting the traffic in the least tightly-packed channel. Also, suppose it is decided that the traffic on channel k should be split, then to reduce N_r it is desired to assign traffic, say (p,q), on channel k to another channel which is already carrying some traffic to node q. In other words, it is not necessary to set up a new "receiving path" in such an assignment.

We have presented some general ideas on which many different algorithms may be developed. A significant shortcoming here is the assumption that the costs are lopsided, so that it is justified to consider the optimization of the different parameters separately. The main intuition we gain is that despite False Conjecture 3.2, N_c does not depend very strongly on the exact traffic distribution so long as each traffic type is small compared with 1. N_c that is a little larger than $\lceil \lambda \rceil$ can be easily achieved simultaneously with the minimization of either N_t or N_r. To see this, consider the fact that minimizing the local transmittability in Step (1) of Fig. 4.1 requires tight-packing of traffic onto channels, and therefore a low value of N_c is achieved simultaneously. Further, $N_c = \lceil \lambda \rceil$ can always be achieved without too much traffic splitting after an algorithm in which the minimization of N_r and N_t is considered. Hence, one would expect there is no strong fundamental trade-off between N_c and N_t or N_r. This will be explored further in the next section.

4.3 Traffic-Matrix Mapping Method

The Traffic-Matrix Mapping Method is a more general, and yet simpler, approach than those discussed previously. Essentially, we map each entry of a traffic matrix, in which entry (p, q) corresponds to traffic (p, q), to one or more channels. The quest is to identify good "mapping patterns".

Mapping an entry onto more than one channel is equivalent to traffic splitting, and therefore it is to be avoided. But traffic splitting is sometimes necessary, for example to allow more efficient use of the channels through tight packing. It is also conceivable that a system in which two nodes may communicate through more than

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
T_1	×	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1	$\frac{1}{9}$ 1
T_2	$\frac{1}{9}$ 2	×	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2
T_3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	×	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3
T_4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	×	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4	$\frac{1}{9}$ 4
T_5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	×	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5	$\frac{1}{9}$ 5
T_6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	×	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6	$\frac{1}{9}$ 6
T_7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	×	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7	$\frac{1}{9}$ 7
T_8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8	×	$\frac{1}{9}$ 8	$\frac{1}{9}$ 8
T_9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	$\frac{1}{9}$ 9	×	$\frac{1}{9}$ 9
T_{10}	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	$\frac{1}{9}$ 10	×

$\lambda^{(p,q)}$ a_{pq} — Channel assigned to Traffic (p,q)

Figure 4.4: A Transmitter-Based Solution

one channel is desired. Such systems will be discussed later. For now, we consider only "1-connectivity" problems where there is no need to establish more channels than necessary to carry traffic from one node to another.

4.3.1 Preliminary Discussion

For simplicity, let's consider a uniform traffic CAP in which

$$\lambda^{(p,q)} = \gamma < 1 \quad \text{for all } (p,q), p \neq q.$$

For $N = 10$ and $\gamma = \frac{1}{9}$, the transmitter-based approach will yield the mapping in Fig. 4.4. The resulting parameters are

$$N_t = 10, \quad N_r = 90, \quad N_c = 10.$$

Let $v_r = v_t$ so that the cost is

$$c = v_t(N_t + N_r) + wN_c. \tag{4.1}$$

Suppose that $N_c = \lceil \lambda \rceil$ could be achieved. Then the objective would be to reduce $N_t + N_r$. A solution better than the transmitter-based solution is given in Fig. 4.5. In this solution, all nodes except node 10 have a local transmittability of 3 and a local receivability of 4. For example, node 1 transmits on channels 1, 2 and 3 and receives on channels 1, 4, 7 and 10. The system parameters' values are

$$N_t = 28, \quad N_r = 39, \quad N_c = 10.$$

Comparing the two results, the increase in N_t in Fig. 4.5 is more than offset by a larger decrease in N_r. To see the underlying reason, we concentrate on the "local cost" rather than the overall system cost. Adopting the notations

a_{pq} = channel assigned to traffic (p, q), assuming nonsplitting traffic mapping; otherwise, a_{pq} would be a set of channels,

$n^{(k)}$ = total number of traffic types assigned to channel k,

$n^{(k)}_{p,\cdot}$ = total number of traffic types $(p, \cdot)$ assigned to channel k,

$n^{(k)}_{\cdot,q}$ = total number of traffic types $(\cdot, q)$ assigned to channel k,

the local cost of traffic (p, q) is defined as

$$c_{pq} = \frac{w}{n^{(a_{pq})}} + \frac{v_t}{n^{(a_{pq})}_{p,\cdot}} + \frac{v_r}{n^{(a_{pq})}_{\cdot,q}}. \tag{4.2}$$

It is easy to see that the system cost is

$$c = \sum_{(p,q):p \neq q} c_{pq}. \tag{4.3}$$

To simplify the minimization of c, one may then concentrate on the minimization of groups of c_{pq} separately and in succession. This is an "approximation" since the resulting c_{pq} for different (p, q)'s are interrelated. Consequently, (p, q) considered later tends to have a higher local cost because of the constraints imposed by the minimization of the local costs of the preceding traffic. Nevertheless, this approximation is useful in that it makes the problem very easy to tackle .

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
T_1	×	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$
T_2	$\frac{1}{9}\ _{1}$	×	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$
T_3	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{1}$	×	$\frac{1}{9}\ _{1}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{2}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$	$\frac{1}{9}\ _{3}$
T_4	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	×	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$
T_5	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{5}$	×	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$
T_6	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{4}$	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{5}$	×	$\frac{1}{9}\ _{5}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$	$\frac{1}{9}\ _{6}$
T_7	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	×	$\frac{1}{9}\ _{9}$	$\frac{1}{9}\ _{9}$	$\frac{1}{9}\ _{9}$
T_8	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{9}$	×	$\frac{1}{9}\ _{9}$	$\frac{1}{9}\ _{9}$
T_9	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{7}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{8}$	$\frac{1}{9}\ _{9}$	$\frac{1}{9}\ _{9}$	×	$\frac{1}{9}\ _{9}$
T_{10}	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	$\frac{1}{9}\ _{10}$	×

Figure 4.5: A Better Solution

Returning to the transmitter-based solution in Fig. 4.4, we see that

$$c_{pq} = \frac{w}{9} + \frac{v_t}{9} + v_r \quad \text{for all } (p,q), p \neq q. \tag{4.4}$$

Each traffic entry (p,q) shares the same channel and the same transmitter with eight other traffic types, and it has an exclusive receiver to itself. For the assignment in Fig. 4.5, the traffic originating from node 10 has the same local cost as in (4.4). All the other traffic types have a local cost of either $w/9+v_t/3+v_r/3$ or $w/9+v_t/3+v_r/2$. For example,

$$c_{1,9} = \frac{w}{9} + \frac{v_t}{3} + \frac{v_r}{3} \tag{4.5}$$

and

$$c_{1,2} = \frac{w}{9} + \frac{v_t}{3} + \frac{v_r}{2}. \tag{4.6}$$

If $v_t = v_r$, it is easily seen that the local costs in Fig. 4.5 are less than those in Fig. 4.4.

It is clear that the key to minimizing the local costs is to find a mapping a_{pq} such that sharing of channels, receivers and transmitters is maximized. But sharing of channels is limited by the finite channel capacity. Two traffic types may share a transmitter (receiver) only if they are also assigned to the same channel and they have the same source (destination) node. To increase transmitter sharing and receiver sharing, we would like to increase $n_{p,\cdot}^{(a_{pq})}$ and $n_{\cdot,q}^{(a_{pq})}$ respectively. However, these are two conflicting objectives since increasing either $n_{p,\cdot}^{(a_{pq})}$ or $n_{\cdot,q}^{(a_{pq})}$ increases the traffic in the assigned channel and the channel can only carry a finite amount of traffic. More Specifically, we have

$$n_t^{(a_{pq})} n_r^{(a_{pq})} \geq n^{(a_{pq})} \tag{4.7}$$

where

$$n_t^{(k)} = \text{the number of transmitters attached to channel } k,$$
$$n_r^{(k)} = \text{the number of receivers attached to channel } k.$$

Inequality (4.7) is satisfied with equality if

$$a_{ij} = a_{pq} = k \iff a_{iq} = a_{pj} = k. \tag{4.8}$$

The interpretation of (4.8) is that by switching rows and columns in the traffic matrix, the traffic types assigned to channel k can be arranged so that they are adjacent to each other and form a rectangular submatrix (see Fig. 4.6 in which the asterisked entries are assigned to the same channel). It is also obvious why (4.8) implies a good solution: if (i, j) and (p, q) are already assigned to channel k, barring exceeding the channel capacity, we may as well assign (i, q) and (p, j) to channel k since no extra transmitter or receiver is needed.

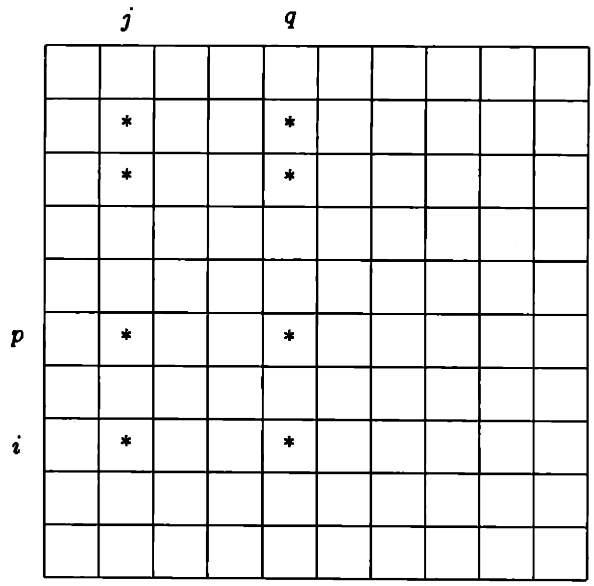


Figure 4.6: An Example of a Rectangular Submatrix Channel Assignment

4.3.2 Lower Bounds and Trade-Offs between Parameters

We now proceed to a more formal study of the trade-off between N_t and N_r. For simplicity, let's consider the uniform traffic case with $\lambda^{(p,q)} = \gamma$ for all (p, q). Further, let's assume nonsplitting traffic assignment. We have the following

$$N_t N_r = \sum_{k=1}^{N_c} n_t^{(k)} \sum_{k=1}^{N_c} n_r^{(k)}, \tag{4.9}$$

$$\sum_{k=1}^{N_c} n^{(k)} = N(N-1), \tag{4.10}$$

$$n^{(k)} \leq \left\lfloor \frac{1}{\gamma} \right\rfloor \quad \forall\, k, \tag{4.11}$$

$$n_t^{(k)} n_r^{(k)} \geq n^{(k)} \quad \forall\, k. \tag{4.12}$$

Equations (4.9) and (4.10) are obtained directly from the definitions of the notation. Inequality (4.12) is the same as (4.7), and (4.11) simply expresses the finiteness of the capacity of a channel.

We wish to find a lower bound on $N_t N_r$. To do so, one may minimize (4.9) subject to Constraints (4.10), (4.11) and (4.12). Since these are only a subset of the constraints to the original CAP, the resulting lower bound may not be as tight as possible. But the least we know is that the result is a legitimate lower bound. To minimize $N_t N_r$, let's first neglect the integrality of $n_t^{(k)}$, $n_r^{(r)}$ and $n^{(k)}$. This again may "loosen" the resulting lower bound. One way to show the resulting lower bound is tight is to present an example which achieves it, and this will indeed be done after the following analysis. Let's denote the lower bound by y. The optimization problem is

$$\begin{aligned}
\min_{\substack{N_c, n^{(k)} \\ n_t^{(k)}, n_r(k)}} \quad & y = \sum_{k=1}^{N_c} n_t^{(k)} \sum_{k=1}^{N_c} n_r^{(k)} \\
\text{s.t.} \quad & (4.10) \\
& (4.11) \\
& (4.12) \\
& n^{(k)}, n_t^{(k)}, n_r^{(k)} \geq 0 \text{ and real} \\
& N_c \geq 0 \text{ and integral.}
\end{aligned} \tag{4.13}$$

Fixing N_c, Constraints (4.10) and (4.11) correspond to convex sets while (4.12) does not. The objective function is neither convex nor concave in the region of interest. In addition, we have the integer constraint of N_c. So, at first sight, the problem may seem to be quite formidable from the nonlinear programming point of view [5]. It turns out that the problem is rather easy. As a first step, let's assume $n^{(k)}$'s and N_c are fixed and that (4.10) and (4.11) are satisfied (note: N_c must be an integer not less than $N(N-1)/\left\lfloor\frac{1}{\gamma}\right\rfloor$). By inspection, we know that for optimality, (4.12) must be satisfied with equality. The problem then becomes

$$\min_{n_t^{(k)},n_r^{(k)}} \quad y = \sum_{k=1}^{N_c} n_t^{(k)} \sum_{k=1}^{N_c} n_r^{(k)}$$
$$\text{s.t.} \quad n_t^{(k)} n_r^{(k)} = n^{(k)} \quad \forall\, k \tag{4.14}$$
$$n_t^{(k)}, n_r^{(k)} \geq 0 \text{ and real.} \tag{4.15}$$

Define two vectors,

$$\mathbf{u} = \left(\sqrt{n_t^{(k)}}\right), \quad \mathbf{v} = \left(\sqrt{n_r^{(k)}}\right). \tag{4.16}$$

We know that

$$\| \mathbf{u} \| \| \mathbf{v} \| \geq \mathbf{u}^T \mathbf{v}. \tag{4.17}$$

It follows that

$$\begin{aligned} y &= \sum_{k=1}^{N_c} n_t^{(k)} \sum_{k=1}^{N_c} n_r^{(k)} \\ &\geq \left[\sum_{k=1}^{N_c} \sqrt{n_t^{(k)} n_r^{(k)}}\right]^2 \\ &= \left[\sum_{k=1}^{N_c} \sqrt{n^{(k)}}\right]^2. \end{aligned} \tag{4.18}$$

The minimum value of y, $\left[\sum_{k=1}^{N_c} \sqrt{n^{(k)}}\right]^2$, is obtained when $n_t^{(k)} = n_r^{(k)} d$ where d is a constant independent of k. Putting this into (4.14), $n^{(k)}/{n_r^{(k)}}^2 = d$ for all k.

Now we come to the second step of the original problem.

$$\min_{N_c, n^{(k)}} \quad w = \sqrt{\min_{n_t^{(k)},n_r^{(k)}} y} = \left[\sum_{k=1}^{N_c} \sqrt{n^{(k)}}\right]$$
$$\text{s.t.} \quad \text{(4.10) and (4.11),}$$
$$N_c \geq 0 \text{ and integral}$$
$$n^{(k)} \geq 0 \text{ and real.} \tag{4.19}$$

The task now is to minimize w on N_c and $n^{(k)}$ with constraints given by (4.10) and (4.11). Suppose N_c is fixed at an arbitrary large value much greater than

$N(N-1)\gamma$. Since the objective function is concave and the constraints correspond to a convex set, the optimal solution must lie on the boundary [5], where the actual number of occupied channels is minimum. Alternatively, the minimum w can be intuitively obtained with the following straightforward reasoning. The problem is analogous to filling $N(N-1)$ units of water into containers, each with capacity $\lfloor \frac{1}{\gamma} \rfloor$. The cost associated with filling a container with an amount of $n^{(k)}$ water is $\sqrt{n^{(k)}}$. Since $\sqrt{n^{(k)}}$ is a strictly decreasing function of $n^{(k)}$, the optimal strategy to is use the minimum number of containers and fill up $\lfloor N(N-1)/\lfloor \frac{1}{\gamma} \rfloor \rfloor$ of the containers completely and the remaining one partially [1]. Let β be the amount of water in the partially-filled container. Then,

$$\beta = \left\{ \frac{N(N-1)}{\lfloor 1/\gamma \rfloor} - \left\lfloor \frac{N(N-1)}{\lfloor 1/\gamma \rfloor} \right\rfloor \right\} \lfloor 1/\gamma \rfloor \tag{4.20}$$

This yields

$$\begin{aligned} \min_{N_c, n^{(k)}} w &= \frac{(N(N-1)-\beta)\sqrt{\lfloor 1/\gamma \rfloor}}{\lfloor 1/\gamma \rfloor} + \sqrt{\beta} \\ &= \frac{N(N-1)}{\sqrt{\lfloor 1/\gamma \rfloor}} + \sqrt{\beta} - \frac{\beta}{\sqrt{\lfloor 1/\gamma \rfloor}} \\ &\geq \frac{N(N-1)}{\sqrt{\lfloor 1/\gamma \rfloor}}, \end{aligned} \tag{4.21}$$

since $\beta \leq \lfloor 1/\gamma \rfloor$. Hence, the following theorem is obtained

---

Theorem 4.1 *Consider the uniform traffic CAP*

$$\begin{aligned} N_t N_r &\geq \max\left[\frac{N^2(N-1)^2}{\lfloor 1/\gamma \rfloor}, N^2 \right] \\ &\geq \max[N^2(N-1)^2\gamma, N^2]. \end{aligned} \tag{4.22}$$

[1]This suggests that N_c is minimized in the process of minimizing $N_t N_r$, strengthening our earlier intuition that there is no strong trade-off between N_c and N_t or N_r. It should be noted, however, that we are not dealing with the original CAP here since the integrality of various parameters have been ignored. As demonstrated by False Conjecture 3.2, we can indeed construct special cases in which the two objectives are conflicting.

Comment: N^2 is a lower bound because each node must have at least a transmitter and a receiver.

Inequality (4.22) reveals the trade-off between N_t and N_r. To show the tightness of the bound, it is interesting to note that the mapping in Fig. 4.4 satisfies the inequality with equality. For the nonuniform traffic case, a simple expression like this is difficult to come by. Nevertheless, there seems to be no fundamental reason why this trade-off does not carry over to the nonuniform traffic case.

An interesting question is whether a nonuniform case with the same total traffic has a lower or a higher lower bound on N_tN_r. There is no general answer to this question, as will be seen from the following examples. Since the total traffic is $\lambda = N(N-1)\gamma$, (4.22) becomes

$$N_tN_t \geq N(N-1)\lambda. \tag{4.23}$$

Let's create a nonuniform traffic situation by shifting all traffic in a uniform traffic matrix to the upper left-hand square submatrix of dimensions $N/2 \times N/2$ in such a way that we have a uniform traffic submatrix. In this extreme case, only half the nodes need to communicate with each other and we essentially have a lower-dimension problem. Clearly,

$$N_tN_t \geq \frac{N}{2}\frac{(N-1)}{2}\lambda, \tag{4.24}$$

and the lower bound is lower than before. Note, however, that N_tN_r is close to this lower bound only if tight and "rectangular" packing can be achieved, as in Fig. 4.4 and Fig. 4.5. As the size of the individual traffic entry increases, this becomes more difficult; and this fact is not reflected in (4.24).

For an example showing the reverse result that a nonuniform traffic case may give rise to a worse situation, modify the traffic matrix in Fig. 4.4 by shifting the traffic

$\lambda^{(1,2)}$ to $\lambda^{(8,9)}$. It is straight-forward to check that tight and rectangular packing is impossible and the lower bound in (4.23) cannot be achieved. From the two examples, we therefore conclude that given a fixed total traffic, there is no general answer to the question of whether the nonuniform traffic case will have a better optimal solution than the uniform traffic case.

The above statement, however, does not deter us from looking for some rules of thumb to deal the nonuniform traffic CAP. Consider the derivation of (4.22), the uniform traffic characteristic is contained only in (4.11) and this constraint is not considered until the second step of the derivation in (4.19). For the nonuniform traffic case, (4.10) can be replaced by

$$n^{(k)} \leq \frac{1}{\phi_k}, \tag{4.25}$$

where ϕ_k is the average value of a traffic entry assigned to channel k (i.e. total traffic divided by the number of entries). There are two possibilities:

1. ϕ_k should be more or less uniform across all channel k. One way to achieve this is to intermix large traffic entries with small traffic entries. The resulting $n^{(k)}$'s will also be more or less equal.

2. ϕ_k should be as nonuniform as possible. One way to achieve this is to group traffic of comparable magnitudes together. Depending on the traffic distribution, the resulting $n^{(k)}$ may vary widely across k.

Suppose tight and rectangular packing (i.e. traffic entries assigned to the same channel form an approximately rectangular submatrix) is possible in both cases, which of the above is better in terms of minimizing N_rN_t? The answer lies in the objective function of (4.19), $\sum_{k=1}^{N_c}\sqrt{n^{(k)}}$. Note that $\sqrt{n^{(k)}}$ is a concave (convex $\cap$) function of $n^{(k)}$. Using an analogy to the Jensen's Bound — $\overline{f(x)} \leq f(\overline{x})$ if $f(x)$ is a concave function of a random variable x, we can establish that Approach 2 is better to Approach 1. Let $f(x) = \sqrt{x}$ and $P[x = n^{(k)}] = 1/N_c$ for all k. Suppose the set of $n^{(k)}$ is obtained using Approach 2. Then, the objective function

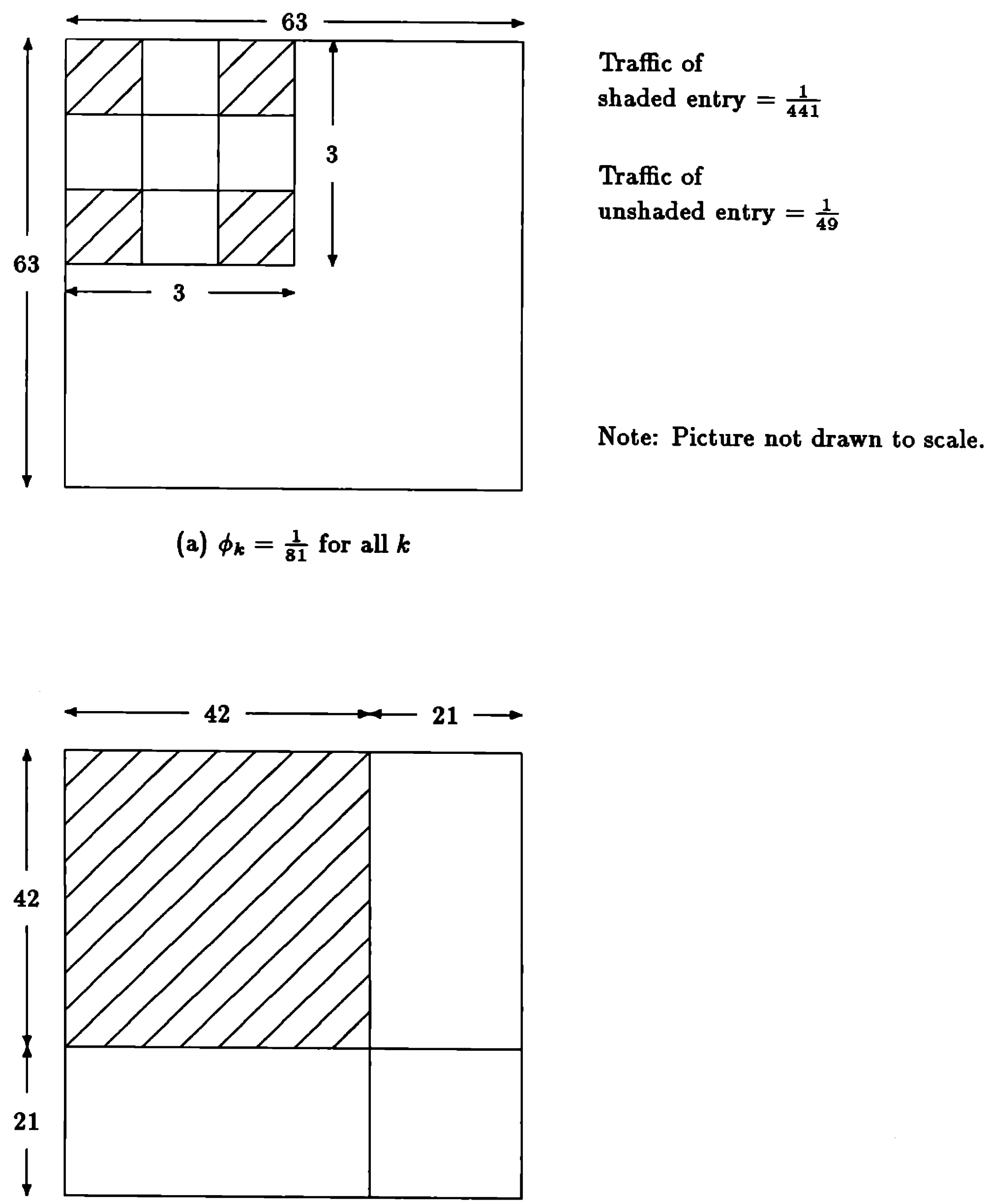
$$\begin{aligned} \sum_{k=1}^{N_c} \sqrt{n^{(k)}} &= N_c \overline{f(x)} \\ &\leq N_c f(\overline{x}). \end{aligned} \tag{4.26}$$

The right side of the inequality is given approximately by the solution of Approach 1. In general, if the traffic sizes differ widely, so will $n^{(k)}$ and the solution of Approach 2 will be much better. It must be stressed that the above argument assumes rectangular and tight packing can be achieved in both cases.

For a concrete, albeit artificial, example, consider Fig. 4.7. Fig. 4.7(a) is a 63×63 matrix with two types of traffic entries. It is made up of repetitions of the 3×3 cell shown at the upper-left corner, and for simplicity, we ignore the fact that the diagonal entries are zero. The shaded entry has value $\frac{1}{441}$ and the unshaded one has value $\frac{1}{49}$. Through interchanging of rows and columns (moving all the purely unshaded rows and columns to the bottom and to the left, respectively), the traffic matrix can be arranged into the form shown in Fig. 4.7(b). This corresponds to relabeling of some of the nodes.

In (a) the total traffic of each cell is $\frac{1}{9}$. A 3 cells $\times$ 3 cells arrangement, or 9×9 entries, forms a square with total traffic of 1. There are altogether $\frac{63^2}{9^2} = 49$ such squares, and to each of them we assign a channel. It is clear that $n_r^{(k)} = n_t^{(k)} = 9$ for all channel k. Hence, $N_t = N_r = 49(9) = 441$. Notice that $\phi_k = \frac{1}{81}$ for all k.

In (b), for the 42×42 submatrix, we form four 21×21 squares and assign a channel to each of them. It is easy to calculate that the shaded region needs a transmittability and a receivability of $4(21) = 84$. For the unshaded region, we form forty-five 7×7 squares and assign a channel to each of them. Here, the transmittability or the receivability is $45(7) = 315$. Hence, for the whole system,



(a) $\phi_k = \frac{1}{81}$ for all k

(b) $\phi_k = \frac{1}{49}$ or $\frac{1}{441}$

Figure 4.7: Two Mapping Methods of a Nonuniform Traffic Matrix.

$N_t = N_r = 84 + 315 = 399$. This is less than that in (a). Notice that here, $\phi_k = \frac{1}{441}$ for the channels assigned to the shaded squares and $\phi_k = \frac{1}{49}$ for the channel assigned to the unshaded squares.

In summary, a good heuristic to CAP should attempt to obtain solutions with the following characteristics:

- The traffic types assigned to the same channel approximates a rectangular submatrix.
- The traffic types assigned to the same channel are of comparable sizes.

We now investigate the relation between the dimensions of the rectangular submatrices and the costs, v_t and v_r. Consider the following optimization problem.

$$\begin{aligned}
\min_{\substack{n_r^{(k)}, n_t^{(k)} \\ N_c, n^{(k)}}} \quad & v_r \sum_{k=1}^{N_c} n_r^{(k)} + v_t \sum_{k=1}^{N_c} n_t^{(k)} + wN_c \\
\text{s.t.} \quad & n_r^{(k)} n_t^{(k)} \geq n^{(k)} \quad \forall\, k \\
& n^{(k)} \leq \frac{1}{\gamma} \quad \forall\, k \\
& \sum_{k=1}^{N_c} n^{(k)} = N(N-1) \\
& N_c, \text{ positive integer} \\
& n_r^{(k)}, n_t^{(k)}, n^{(k)} \geq 0 \text{ and real.}
\end{aligned} \tag{4.27}$$

This is analogous to the minimization of $N_t N_r$ as before. To minimize the above objective function, we first fix N_c and $n^{(k)}$ and consider the problem

$$\begin{aligned}
\min_{n_r^{(k)} n_t^{(k)}} \quad & v_r \sum_{k=1}^{N_c} n_r^{(k)} + v_t \sum_{k=1}^{N_c} n_t^{(k)} + wN_c \\
\text{s.t.} \quad & n_r^{(k)} n_t^{(k)} \geq n^{(k)} \\
& n_r^{(k)}, n_t^{(k)} \geq 0 \text{ and real.}
\end{aligned} \tag{4.28}$$

The above can be easily solved and the optimizing parameters are

$$n_r^{(k)} = \sqrt{\frac{v_t}{v_r}}\sqrt{n^{(k)}}, \quad n_t^{(k)} = \sqrt{\frac{v_r}{v_t}}\sqrt{n^{(k)}} \tag{4.29}$$

Substituting (4.29) into the objective function, the second step of the optimization process can be written as

$$\begin{aligned} \min_{N_c, n^{(k)}} \quad & 2\sqrt{v_r v_t}\sum_{k=1}^{N_c}\sqrt{n^{(k)}} + wN_c \\ \text{s.t} \quad & n^{(k)} \leq \frac{1}{\gamma} \qquad (4.30) \\ & \sum_{k=1}^{N_c} n^{(k)} = N(N-1) \\ & N_c, \text{ positive integer} \\ & n^{(k)} \geq 0 \text{ and real.} \qquad (4.31) \end{aligned}$$

This is analogous to the problem in (4.19) and shares the same feature that in the optimal solution, N_c is minimized and there is at most one k where (4.30) is not satisfied with equality.

Returning to the algorithmic study, the important insight to be gained from the above analysis is contained in (4.29), from which we obtain

$$\frac{n_r^{(k)}}{n_t^{(k)}} = \frac{v_t}{v_r}. \tag{4.32}$$

In general, (4.32) may not be achieved in the original CAP. Nevertheless, it is a relationship governing the relative magnitudes of $n_r^{(k)}$ and $n_t^{(k)}$ in a good solution. One may then exploit this fact and try to approximate the relationship when exploring a heuristic scheme for solving the CAP.

Substituting $n^{(k)} = \frac{1}{\gamma}$ into (4.29) and summing over all channels, we have

$$\begin{aligned} N_t &= N_c\sqrt{\frac{v_r}{v_t}}\sqrt{\frac{1}{\gamma}} \geq \frac{\lambda}{\sqrt{\gamma}}\sqrt{\frac{v_r}{v_t}} \quad , \\ N_r &= N_c\sqrt{\frac{v_t}{v_r}}\sqrt{\frac{1}{\gamma}} \geq \frac{\lambda}{\sqrt{\gamma}}\sqrt{\frac{v_t}{v_r}}. \end{aligned} \tag{4.33}$$

Hence, a good heuristic should attempt to achieve values of N_t and N_r as close to the above bounds as possible. An alternative and more direct approach to obtaining the above bounds is solving the problem:

$\min v_t N_t + v_r N_r + w N_c$ s.t. $N_t N_r \geq N^2(N-1)^2\gamma, \quad N_c \geq \lceil N(N-1)\gamma \rceil$.

The following theorem is obtained.

Theorem 4.2 *Consider the uniform traffic CAP.* $N_c \geq \lceil N(N-1)\gamma \rceil$ *and* $v_t N_t + v_r N_r \geq \max[2N(N-1)\sqrt{v_t v_r \gamma}, (v_r + v_t)N]$.

4.3.3 A Traffic Mapping Heuristic – An Example

We are now ready to explore a traffic-matrix mapping heuristic based on the ideas developed. The method, outlined in Fig 4.8, successively maps some traffic in the traffic matrix to a channel, splitting the traffic entries if necessary. The input to the algorithm is a traffic matrix $\mathbf{A} = (a_{ij})$ and costs v_t and v_r. It produces a mapped traffic matrix $\mathbf{B}$, whose component b_{ij} is a set of ordered pairs describing how a_{ij} is partitioned and assigned to different channels. Specifically, suppose a_{ij} is split into n partitions; the size of partition k is $a_{ij}^{(k)}$. Let $c_{ij}^{(k)}$ be the channel $a_{ij}^{(k)}$ is assigned to. Then $b_{ij} = \{b_{ij}^{(k)}, 1 \leq k \leq n : b_{ij}^{(k)} = (a_{ij}^{(k)}, c_{ij}^{(k)}), \sum_k a_{ij}^{(k)} = a_{ij}\}$. Obviously, there is only one ordered pair in b_{ij} if there is no splitting in traffic. Roughly, the algorithm attempts to achieve a rectangular mapping for each channel, and it does so by expanding and maintaining the rectangular structure of the traffic already assigned to the channel, starting with a single traffic entry. The traffic that has been mapped is subtracted from $\mathbf{A}$ and the algorithm terminates when all traffic entries of $\mathbf{A}$ are reduced to zero. More explanations come after the outline below.

```
HEURISTIC TRAFFIC-MATRIX MAPPING METHOD FOR CAP
Input:    An N × N traffic matrix A = (a_ij), a_ij < 1 ∀ i,j; v_t; v_r.
Output:   An N × N mapped traffic matrix B = (b_ij);
          b_ij = {b_ij^(k) : b_ij^(k) = (a_ij^(k), c_ij^(k)), Σ_k a_ij^(k) = a_ij}.
begin
    cc := 1; cp := 1; flag := 0; sw := 0; R := ∅; C := ∅; b_ij = ∅ ∀ i,j;   (comment: initialization)
****************************************************************************
mn: findmax(A, {1,2,···,N}, {1,2,···,N}, r, s ,cp, flag);
    if flag = 0
    then  begin
          b_rs := b_rs ∪ {(a_rs, cc)}; a_rs := 0;
          cp := cp − a_rs; R := R ∪ r; C := C ∪ s;
          goto ex:
    else if flag = −1
    then  begin
          if cp ≠ 0 then b_rs := b_rs ∪ {(cp, cc)}; a_rs := a_rs − cp;
          cp := 1; cc := cc + 1; R := ∅; C := ∅; flag := 0
          goto mn: ;
          end
    else  begin
          if sw = 1 then B := B^T;   (comment:  correct "transposing" before output)
          stop;
          end
****************************************************************************
       ex: if abs[(|R| + 1)v_t − |C|v_r] ≤ abs[|R|v_t − (|C| + 1)v_r]
           then goto rw:
           else goto cl: ;
****************************************************************************
       rw:    begin
              findmax(A, R^c, C, r, s, cp, flag);
              if flag = 0
              then  begin
                    b_rs := b_rs ∪ {(a_rs, cc)}; a_rs := 0;
                    cp := cp − a_rs; R := R ∪ r;
          rwexp:    findmax(A, {r}, C, r, s, cp, flag)
                    if flag = 0
                    then        begin
                                b_rs := b_rs ∪ {(a_rs, cc)}; a_rs := 0;
                                cp := cp − a_rs;
                                goto rwexp:;
                                end
                    else if flag = −1
                    then        begin
                                if  cp ≠ 0 then b_rs := b_rs ∪ {(cp, cc)}; a_rs := a_rs − cp;
                                cp := 1; cc := cc + 1; R := ∅; C := ∅; flag := 0
                                goto mn:;
                                end
                    else        begin
                                flag := 0 ;
```

```
                              goto ex: ;
                              end
                  end
            else if flag = −1
            then  begin
                  if cp ≠ 0 then b_rs := b_rs ∪ {(cp, cc)}; a_rs := a_rs − cp;
                  cp := 1; cc := cc + 1; R := ∅; C := ∅; flag := 0
                  goto mn:;
                  end
            else if flag = 1 then goto cl:;
            else  begin
                  R := ∅; C := ∅; flag := 0;
                  goto mn:;
                  end
            end
***************************************************************************
      cl:   begin   (comment: can use the same steps as rw:)
            A := A^T;
            B := B^T;
            tempset := R; R := C; C := tempset;
            temp := v_t; v_t := v_r; v_r := temp;
            if sw = 0 then sw := 1 else sw := 0;
            goto rw:;
            end
***************************************************************************
end


***************************************************************************
procedure findmax(A, RI, CI, r, s, cp, flag)
   if a_ij = 0 ∀ i ∈ RI, j ∈ C
   then   flag := flag + 1;
   else if (a_ij = 0) or (a_ij > cp) ∀ i ∈ RI, j ∈ CI
   then   begin
          flag := −1;
          r, s := arg max_ij {a_ij : i ∈ RI, j ∈ CI}
          end
   else   begin
          flag := 0;
          r, s := arg max_ij {a_ij : i ∈ RI, j ∈ CI, a_ij ≤ cp};
   end
```

Figure 4.8: A Traffic-Matrix Mapping Heuristic to CAP

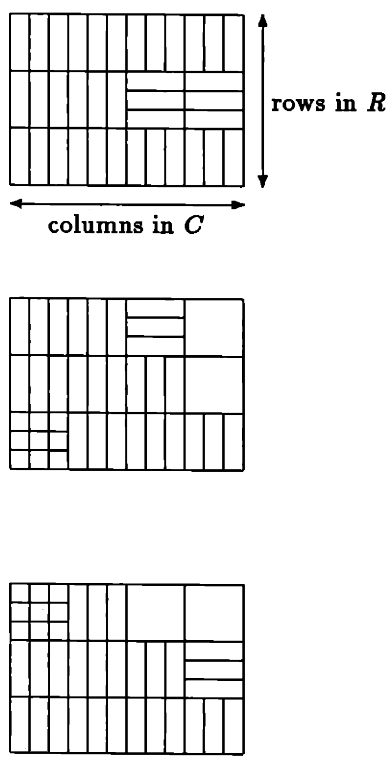
In Fig. 4.8, superscript c denotes the complementary set and superscript T denotes the transpose of a matrix. R is a set consisting of row indices of the traffic already assigned to cc, the channel whose mapping is currently considered by the algorithm. In other words, $R = \{i : \text{ part or the whole of traffic } (i,j) \text{ is assigned to } cc\}$. C is the corresponding set of column indices. Each time some traffic is assigned to cc, that amount of capacity is deducted from the cp, the remaining capacity of cc.

The procedure "findmax", which is defined at the end of Fig. 4.8, accepts a matrix $\mathbf{A}$, sets of row and column indices RI, CI and cp. It attempts to find the $\max a_{ij} : i \in RI, j \in CI, 0 < a_{ij} \leq cp$. If this is successful, $flag$ will be set to 0 and r, s will be set to the arguments of such an a_{ij}. However, if all non-zero a_{ij} are greater than cp, $flag$ will be set to -1. Finally, if all $a_{ij} = 0$, then 1 will be added to $flag$. Each time after "findmax" is used in the main algorithm, $flag$ is tested so that the appropriate action can be taken.

At any time, the algorithm is in one of four states, three of which are depicted in Fig. 4.9. The other possible state is the Starting State where the algorithm looks for an entry as the "base" of a rectangle to be expanded. The base entry chosen is the largest entry in $\mathbf{A}$ allowed by cp. In Fig. 4.8, "mn:" corresponds to the Starting State.

In the Complete-Rectangle State, all the traffic entries specified by R and C have already been mapped to either cc or some previously considered channels. We can either expand the rectangle horizontally by adding a column or vertically by adding a row. The expansion is determined by approximating (4.32) in Stage "ex:" in Fig. 4.8. Recall that it is desirable to have $v_r n_r^{(k)} - v_t n_t^{(k)} \approx 0$.

Suppose a vertical expansion is desired, we then consider which row is to be added in Stage "rw:". This algorithm chooses the row with the largest possible



(a) Complete-Rectangle State.

(b) Incomplete-Column State.

(c) Incomplete-Row State.

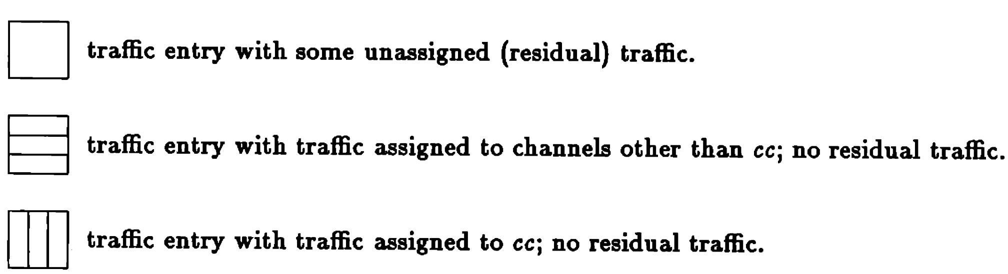


Figure 4.9: Three Possible States in a Traffic-Matrix Mapping Heuristic

entry allowed by *cp*. A vertical expansion is attempted by calling "findmax" with $RI = R^c$, the indices not in R, and $CI = C$, and it is successful when the $flag$ returned by "findmax" is either 0 or -1. Otherwise, we are forced to consider horizontal expansion in cl: even though (4.32) dictates otherwise. However, $flag$ greater than 1 signifies that a horizontal expansion has already been attempted and failed (note: each failure adds 1 to $flag$); and therefore neither vertical nor horizontal expansion is possible. This means $a_{ij} = 0 \; \forall \; i \in R$ or $j \in C$. Upon this, the algorithm goes back to the Starting State in "mn:", where instead of expanding an already established rectangle, it looks for a new entry as the base of a new rectangle to be expanded. Note that *cc* and *cp* remain unaltered since there may still be some unassigned capacity, *cp*, in *cc*.

Now suppose the vertical expansion is successful with $flag = -1$. This means all the traffic entries considered is larger than *cp*. The algorithm chooses the largest entry and assigns *cp* of it to *cc*. The intuitive reason is that the large traffic entry makes packing difficult and by reducing it here, channel packing will be easier later. After that, since all capacity in *cc* has been assigned, the algorithm enters the Starting State and considers the next channel's mapping.

If the vertical expansion is successful with a $flag = 0$ then the maximum entry a_{rs} is assigned to *cc*, a_{rs} in $\mathbf{A}$ is set to zero, and we enter the Incomplete-Row State, or the row expansion stage "rwexp:" in Fig. 4.8. The algorithm then tries to assign all the row entries with row and column indices confined by r and C to *cc*. However, since this may exceed *cp*, it considers the addition of the entries one at a time starting with the largest. For this purpose, "findmax" is used again. Note that any new entries added during the row expansion can be considered as enjoying a "free ride" in that no extra transmitters or receivers are needed because they are are furnished by traffic already assigned to *cc*. If the whole row can be added, we come to the Complete-Rectangle State again in "ex:". Otherwise, cp is reduced to zero and the algorithm will be in the Starting State.

Because of the symmetry of the problem, the part of the algorithm that deals with vertical expansion and row expansion can be used for horizontal expansion and column expansion after transposing the two matrices $\mathbf{A}$ and $\mathbf{B}$, switching R with C and v_t with v_r. This is done at "cl:". To keep track of the these switching and transposing, a switch sw is used. If $sw = 0$ then the notations have their usual meanings, and $sw = 1$ if columns and rows have been switched.

The algorithm is written in such a way as to ease presentation and it is not very efficient because of many redundancies. By introducing more data structures, it can certainly be improved further. These details will be left out here.

The algorithm is polynomial in terms of its computation complexity. To see this, note that "findmax" is where most of the computation is done and it is $O(N^2)$. Let's consider the maximum number of times "findmax" can be called from the main program. Since the algorithm terminates when all entries in $\mathbf{A}$ are reduced to zero, the key is to identify the maximum number of calls to "findmax" before an entry in $\mathbf{A}$ is reduced to zero. If "findmax" returns $flag = 0$, we know that an entry in $\mathbf{A}$ will be reduced to zero immediately after that. If it returns $flag = -1$, a new channel with $cp = 1$ is considered the next time "findmax" is called. Since we assume $a_{ij} < 1 = cp$, this next call of "findmax" is guaranteed to reduce an entry in $\mathbf{A}$ to zero if $\mathbf{A} \neq 0$. The situation is more complicated if $flag \neq 0$, or -1. An example of the maximum number of times "findmax" will be called before an entry is reduced to 0 is as follows: The "findmax" at rwexp: returns $flag = 1$. After that, $flag$ is set to 0 and the algorithm goes to ex:. A vertical expansion is needed and the "findmax" immediately after "rw:" returns $flag = 1$, whereupon we goto cl:.

The “findmax” immediately after “cl:” returns $flag = 2$. So we go to “mn:”. The “findmax” at mn: returns $flag = -1$. The next “findmax” is guaranteed to reduce an entry of $\mathbf{A}$ to 0. Altogether, “findmax” is called five times. From above, we deduce a maximum of five calls to “findmax” is needed before an entry is reduced to 0 and this is independent of N. The number of calls to ‘findmax” is therefore of order $O(N^2)$. Hence, at worst, the algorithm is of order $O(N^4)$ since “findmax” is $O(N^2)$.

For illustration, consider the traffic matrix in Fig. 4.10. Fig. 4.11 is the solution (i.e. the $\mathbf{B}$ matrix) generated by the algorithm for $v_t = v_r$ and Fig. 4.12 is the solution for $v_t = 2v_r$. Notice that some traffic entries in the latter case are split.

One may think of many heuristics of different variations based on the ideas presented in this section. For example, an improvement to the algorithm above is to consider the addition of a whole row or a whole column at a time rather than an entry at a time. For vertical expansion, we may choose the row with the largest sum of entries rather than the largest entry. By taking a more “global” view, it is likely that a better solution can be found. The trade-off is that more computation is necessary. These details will not be pursued further as the main purpose of this chapter, i.e. to explore general rules for devising heuristics, has been achieved.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
T_1	×	0.05	0.03	0.20	0.15	0.04	0.08	0.03	0.01	0.09
T_2	0.05	×	0.01	0.03	0.02	0.04	0.03	0.03	0.01	0.06
T_3	0.10	0.09	×	0.01	0.03	0.07	0.08	0.01	0.06	0.05
T_4	0.20	0.03	0.10	×	0.03	0.06	0.09	0.07	0.13	0.09
T_5	0.01	0.25	0.15	0.02	×	0.03	0.01	0.05	0.02	0.13
T_6	0.02	0.01	0.23	0.01	0.02	×	0.06	0.08	0.14	0.06
T_7	0.03	0.14	0.20	0.20	0.03	0.06	×	0.07	0.02	0.14
T_8	0.09	0.17	0.12	0.01	0.04	0.02	0.08	×	0.06	0.14
T_9	0.13	0.04	0.16	0.04	0.21	0.08	0.15	0.03	×	0.01
T_{10}	0.12	0.10	0.05	0.18	0.09	0.19	0.09	0.01	0.08	×

Figure 4.10: A Traffic-Matrix with 10 Nodes.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
T_1	×	(0.05,6)	(0.03,3)	(0.20,2)	(0.15,3)	(0.04,6)	(0.08,3)	(0.03,7)	(0.01,6)	(0.09,6)
T_2	(0.05,6)	×	(0.01,7)	(0.03,7)	(0.02,7)	(0.04,7)	(0.03,6)	(0.03,7)	(0.01,6)	(0.06,6)
T_3	(0.10,5)	(0.09,5)	×	(0.01,5)	(0.03,7)	(0.07,5)	(0.08,6)	(0.01,7)	(0.06,6)	(0.05,6)
T_4	(0.20,4)	(0.03,7)	(0.10,3)	×	(0.03,3)	(0.06,7)	(0.09,3)	(0.07,4)	(0.13,4)	(0.09,4)
T_5	(0.01,7)	(0.25,1)	(0.15,1)	(0.02,7)	×	(0.03,7)	(0.01,7)	(0.05,7)	(0.02,7)	(0.13,1)
T_6	(0.02,6)	(0.01,6)	(0.23,2)	(0.01,2)	(0.02,7)	×	(0.06,6)	(0.08,7)	(0.14,2)	(0.06,6)
T_7	(0.03,4)	(0.14,5)	(0.20,2)	(0.20,2)	(0.03,7)	(0.06,7)	×	(0.07,7)	(0.02,2)	(0.14,4)
T_8	(0.09,6)	(0.17,1)	(0.12,1)	(0.01,7)	(0.04,7)	(0.02,7)	(0.08,6)	×	(0.06,6)	(0.14,1)
T_9	(0.13,4)	(0.04,1)	(0.16,3)	(0.04,5)	(0.21,3)	(0.08,5)	(0.15,3)	(0.03,7)	×	(0.01,4)
T_{10}	(0.12,4)	(0.10,5)	(0.05,7)	(0.18,5)	(0.09,7)	(0.19,5)	(0.09,6)	(0.01,7)	(0.08,4)	×

$$v_t = v_r$$

$$N_r = 32$$

$$N_t = 33$$

Figure 4.11: The Solution to Fig. 4.10 for $v_t = v_r$.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
T_1	×	(0.05,3)	(0.03,7)	(0.20,3)	(0.15,3)	(0.04,3)	(0.08,5)	(0.03,7)	(0.01,7)	(0.09,3)
T_2	(0.05,7)	×	(0.01,7)	(0.03,7)	(0.02,7)	(0.04,7)	(0.03,7)	(0.03,7)	(0.01,7)	(0.06,7)
T_3	(0.10,5)	(0.09,5)	×	(0.01,6)	(0.03,7)	(0.07,6)	(0.08,5)	(0.01,6)	(0.06,6)	(0.05,5)
T_4	(0.20,4)	(0.03,7)	(0.10,4)	×	(0.03,7)	(0.06,4)	(0.09,4)	(0.07,7)	(0.13,4)	(0.09,4)
T_5	(0.01,7)	(0.25,1)	(0.15,1)	(0.02,7)	×	(0.03,7)	(0.01,7)	(0.05,7)	(0.02,1)	(0.13,1)
T_6	(0.02,7)	(0.01,1)	(0.23,1)	(0.01,7)	(0.02,7)	×	(0.06,7)	(0.07,7) (0.01,1)	(0.14,1)	(0.06,1)
T_7	(0.03,4)	(0.13,6) (0.01,2)	(0.20,2)	(0.20,2)	(0.03,2)	(0.06,6)	×	(0.07,6)	(0.02,4)	(0.14,4)
T_8	(0.09,5)	(0.17,5)	(0.12,5)	(0.01,7)	(0.04,7)	(0.02,7)	(0.08,5)	×	(0.06,7)	(0.14,5)
T_9	(0.13,4)	(0.04,6)	(0.16,2)	(0.04,2)	(0.21,2)	(0.08,6)	(0.15,2)	(0.03,6)	×	(0.01,4)
T_{10}	(0.12,4)	(0.09,6) (0.01,3)	(0.05,6)	(0.18,3)	(0.09,3)	(0.19,3)	(0.09,6)	(0.01,6)	(0.08,6)	×

$$v_t = 2v_r$$

$$N_r = 44$$

$$N_t = 23$$

Figure 4.12: The Solution to Fig. 4.10 for $v_t = 2v_r$.

4.4 Upper Bounds and Asymptotic Behaviors

This section addresses the upper bounds of N_tN_r and $v_tN_t + v_rN_r + wN_c$. Unlike the study of lower bounds, in order to show the validity of the upper bounds, it is necessary to present the actual methods that achieve them. We will show that the differences between the upper and lower bounds are negligible if the system traffic is large and the individual traffic is small. This means the heuristics used to derive the upper bounds are very good methods in this asymptotic situation.

For an upper bound on N_tN_r, two obvious candidates that have the desired asymptotic characteristic are

$$N_tN_r \leq N^2(N-1)^2\gamma + \epsilon(\gamma), \tag{4.34}$$

$$N_tN_r \leq N^2(N-1)^2\gamma\,(1+\delta(\gamma))\,. \tag{4.35}$$

$\epsilon(\gamma)$ and $\delta(\gamma)$ are functions which approach 0 as γ goes to 0. It is clear that (4.34) is a stronger condition than (4.35) since the former means the upper bound actually approaches the lower bound asymptotically while the latter only implies the difference between the upper and lower bounds are negligible compared with the values of the bounds. We will show that while an upper bound as in (4.35) can be found, it is not possible to achieve (4.34). This is not surprising considering that the lower bound can be achieved only under perfect packing situation where all channels are fully filled.

First of all, for a meaningful discussion, it is necessary to specify the functions $\epsilon(\cdot)$, $\delta(\cdot)$ and the way γ approaches 0 more precisely. Consider the trivial case where N is fixed as $\gamma \to 0$, but $\gamma > 0$. The lower bound $N^2(N-1)^2\gamma \to 0$ and only one channel is needed to support all the traffic. But, $N_t = N_r = N$ since each node must have a minimum local transmittability or receivability of 1. Consequently there is no way the upper bound can come close to $N^2(N-1)^2\gamma$. Therefore, to avoid the trivial single-channel case, we must let $N \to \infty$ as $\gamma \to 0$: there are many ways in which this can be done, for example, we could fix $N\gamma$. As a result, ϵ and δ must be functions of both γ and N.

Let's fix the total system traffic

$$N(N-1)\gamma = k \tag{4.36}$$

where $k \geq 3$ and it is an integer constant. To show (4.34) is impossible, we identify a sequence of γ, say $\{\gamma_i, i = 1, 2, \ldots\}$, which approaches 0, and that

$$N_t N_r \geq N^2(N-1)^2\gamma_i + c \tag{4.37}$$

where c is a strictly positive constant. In other words, we want to show that for $\gamma = \gamma_i$, a lower bound better than $N^2(N-1)^2\gamma$ can be found. Examining the two lines in (4.22), this is not too difficult a problem if $1/\gamma$ is not an integer value. Simply look for a sequence $\{\gamma_i\}$ where $1/\gamma_i = I_i + \epsilon'$, $0 < \epsilon' < 1$ and I_i is an integer. For this purpose, we fix $N_i = ik+2$. Then $1/\gamma_i = (i^2k^2+3ik+2)/k$, giving $\epsilon' = 2/k$ and $I_i = i^2k + 3i$. Now, $\lfloor 1/\gamma_i \rfloor = I_i$ and

$$\begin{aligned} N_t N_r &\geq \frac{N^2(N-1)^2}{\lfloor 1/\gamma_i \rfloor} \\ &\geq N^2(N-1)^2\gamma_i(1+\epsilon'\gamma_i) \\ &= N^2(N-1)^2\gamma_i + 2k \end{aligned} \tag{4.38}$$

Thus, c is $2k$, and therefore (4.34) cannot be achieved in general. The argument considers the nonperfect channel packing situation where $1/\gamma$ is not an integer; i.e. a channel cannot be fully packed without traffic splitting. It can be shown that even if $1/\gamma_i$ is restricted to be integer value, (4.34) is still not achievable in general. The details will not be discussed here. The basic idea is to resort to the tighter bound given by (4.21) and fixing $N(N-1)\gamma + \gamma = k$ to avoid perfect packing of the whole system.

We now try to obtain the less stringent upper bound of (4.35). Let's first consider the more general nonuniform traffic situation with the restriction

$$\lambda^{(p,q)} \leq \lambda^u \qquad \forall \quad (p,q), \tag{4.39}$$

i.e. all traffic entries are bounded by λ^u. Furthermore, γ is redefined to be the average traffic,

$$\gamma = \frac{\sum_{p,q:p\neq q} \lambda^{(p,q)}}{N(N-1)}. \tag{4.40}$$

It is enough to show a heuristic which achieves the upper bound desired.

Consider a transmitter-based heuristic without traffic splitting. For each node p, we solve the bin-packing problem with $\lambda_q = \lambda^{(p,q)}$ for all $q \neq p$. As previously discussed, traffic entries assigned to the same bin belong to the same channel. Suppose the First-Fit algorithm for BPP is employed. This heuristic does not put a traffic entry on a new channel unless the entry is greater than the remaining capacity of every channel already occupied. Since $\lambda^{(p,q)} \leq \lambda^u$, the resulting solution has at most a channel with unused capacity greater than or equal to λ^u. It follows that the local transmittability required by node p cannot be greater than $\lceil \sum_{q:q\neq p} \lambda^{(p,q)}/(1-\lambda^u) \rceil$. Summing over all nodes yields

$$\begin{aligned} N_t &\leq \sum_p \left\lceil \frac{\sum_{q:q\neq p} \lambda^{(p,q)}}{1-\lambda^u} \right\rceil \\ &\leq \sum_p \left[\frac{\sum_{q:q\neq p} \lambda^{(p,q)}}{1-\lambda^u} + 1 \right] \\ &= \frac{N(N-1)\gamma}{1-\lambda^u} + N. \end{aligned} \tag{4.41}$$

After solving N BPP as above, we may further reduce N_c and N_r by packing the traffic of different channels into one channel (see Section 4.1). Whether this further optimization is performed, it is easy to see that each node needs a maximum local receivability of $(N-1)$ provided $\lambda^u \leq 1$ and there is no traffic splitting. Thus,

$$N_r \leq N(N-1). \tag{4.42}$$

From (4.41) and (4.42),

$$N_t N_r \leq N^2(N-1)^2\gamma \left[\frac{1}{1-\lambda^u} + \frac{1}{(N-1)\gamma}\right] \tag{4.43}$$

Comparing the above to (4.35), we see that

$$\delta = \frac{1}{1-\lambda^u} + \frac{1}{(N-1)\gamma} - 1 \tag{4.44}$$

Suppose $(N-1)^x\gamma \geq k$ where $0 < x < 1$ and $k = constant$. Letting $\lambda^u, \gamma \to 0$,

$$\delta \leq \frac{1}{1-\lambda^u} - 1 + \frac{1}{k(N-1)^{1-x}} \to 0 \tag{4.45}$$

and we get the desired result.

Keeping the quantity $(N-1)^x\gamma \geq k$ implies that the average traffic from a single source, $(N-1)\gamma \to \infty$, as $\gamma \to 0$. Intuitively, this implies that the channel with unused capacity greater than λ^u, if there is one, is negligible relative to the total number of channels assigned to a source. The second term in (4.41) becomes negligible and the result follows immediately without detailed analysis.

The above discussion is summarized in the following theorem:

Theorem 4.3 *Let γ be the average traffic and λ^u be the maximum traffic entry. Suppose $(N-1)^x\gamma \geq k$ where k is a constant and $0 < x < 1$. Then, $N_t N_r \leq N^2(N-1)^2\gamma(1+\delta(\lambda^u,\gamma,N,x))$ where $\delta \to 0$ as $\lambda^u, \gamma \to 0$.*

The argument leading to the above theorem will break down if $x \geq 1$. It is not clear whether there are better heuristics with the desired asymptotic behavior but less stringent requirements. However, consideration of the simple uniform traffic situation allows us to enlarge the range of x to $0 < x < 2$.

A simple rectangular mapping strategy is used, and it is illustrated in Fig. 4.13. Notice that the dimensions of a "regular" rectangle is $n_t \times n_r$ whereas the rectangles

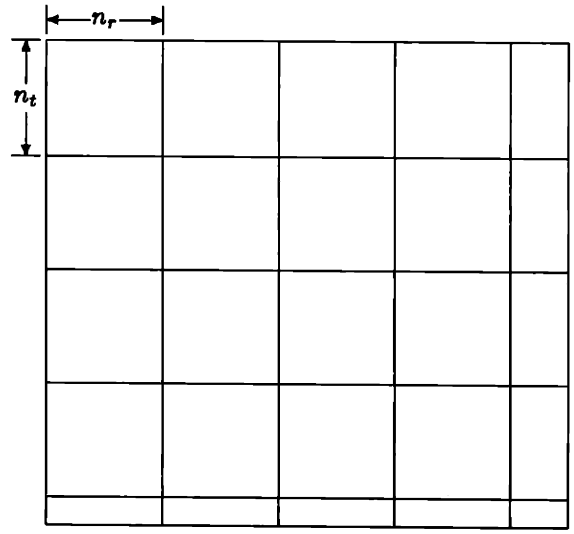


Figure 4.13: A Rectangular Traffic-Matrix Mapping Scheme

at the bottom and right edges of the matrix are smaller because of the finite size of the matrix and that N may not be divisible n_t or n_r. For node p, the local transmittability is $\lceil N/n_r \rceil$, corresponding to the total number of rectangles occupied by traffic of the types $(p, \cdot)$. Similarly, the local receivability of each node is $\lceil N/n_t \rceil$.

$$\begin{aligned} N_t &= N\left\lceil\frac{N}{n_r}\right\rceil \le N\left(\frac{N+n_r}{n_r}\right), \\ N_r &= N\left\lceil\frac{N}{n_t}\right\rceil \le N\left(\frac{N+n_t}{n_t}\right). \end{aligned} \tag{4.46}$$

For feasibility

$$n_t n_r \le \frac{1}{\gamma}. \tag{4.47}$$

For efficient packing, we require

$$n_t(n_r+1)\gamma > 1, \tag{4.48}$$

$$(n_t+1)n_r\gamma > 1. \tag{4.49}$$

The above essentially says that if a row or a column can be added to the regular rectangle without exceeding the channel capacity, it should be done. From (4.46),

$$\begin{aligned} N_t N_r &\leq N^2(N+n_t)(N+n_r)\frac{1}{n_t n_r} \\ &\leq N^2\left[(N-1)+(n_t+1)\right]\left[(N-1)+(n_r+1)\right]\frac{\gamma}{1-n_r\gamma} \\ &= N^2(N-1)^2\gamma\left[\frac{1}{1-n_r\gamma}\right]\left[1+\frac{2+n_t+n_r}{(N-1)}+\frac{(n_t+1)(n_r+1)}{(N-1)^2}\right] \end{aligned} \quad (4.50)$$

where the second line is obtained by substitution from (4.49). Now, the rectangle's dimensions n_t and n_r are further restricted by the following scheme. Roughly, we try to obtain "square" assignments. For efficient assignments satisfying (4.48) and (4.49), an exact square may not be possible. In such a case, we expand the rows rather than the columns. More exactly, to determine n_t and n_r, find n such that $n^2\gamma \leq 1$ and $(n+1)^2\gamma > 1$. Set $n_r = n$, $n_t = n+m$ where m is the largest integer such that (4.47) is satisfied. Clearly, $m \leq 2$ since otherwise $(n+1)^2\gamma \leq n(n+m)\gamma \leq 1$, and the hypothesis that $(n+1)^2\gamma > 1$ will not be valid. Hence,

$$n_r + 2 \geq n_t \geq n_r. \quad (4.51)$$

Using the above in (4.50) yields

$$N_t N_r \leq N^2(N-1)^2\gamma\left[\frac{1}{1-n_r\gamma}\right]\left[1+\frac{4+2n_r}{(N-1)}+\frac{(n_r+3)(n_r+1)}{(N-1)^2}\right]. \quad (4.52)$$

Now $n_r^2 \leq 1/\gamma$ so that

$$\frac{1}{1-n_r\gamma} \leq \frac{1}{1-\sqrt{\gamma}} = 1+\delta'(\gamma) \quad (4.53)$$

where $\delta' \to 0$ as $\gamma \to 0$. Suppose $(N-1)^x\gamma \geq k$ for some constant k and $0 < x < 2$. Then $n_r \leq (N-1)^{\frac{x}{2}}/\sqrt{k}$.

$$\frac{n_r}{(N-1)} \leq \frac{1}{(N-1)^{1-\frac{x}{2}}\sqrt{k}}, \quad (4.54)$$

which goes to 0 as $N \to \infty$. Using these facts in (4.52), we obtain the following theorem.

Theorem 4.4 *Consider the uniform traffic situation. Suppose* $(N-1)^x\gamma \geq k$ *where* k *is a constant and* $0 < x < 2$. *Then,* $N_t N_r \leq N^2(N-1)^2\gamma(1+\delta(\gamma, N, x))$ *where* $\delta \to 0$ *as* $\gamma \to 0$.

Let's derive an upper bound for $v_t N_t + v_r N_r + w N_c$. A modified version of the above strategy may be used. It will be shown that in the limit $\gamma \to 0$, $N(N-1)\gamma \to \infty$, the difference between this upper bound and the lower bound derived previously becomes negligible. The scheme can certainly be improved further to obtain a tighter bound. The primary aim here, however, is to demonstrate the desired asymptotic behavior without complicated analysis. The heuristic can also be used to derive Theorem 4.4 although the resulting bound is not as good.

Instead of approximating "square" assignment as before, we want to obtain rectangular assignments where the dimensions n_t and n_r are related by

$$\frac{n_t}{n_r} \approx \frac{v_r}{v_t}. \tag{4.55}$$

To approximate the relation, we let

$$n_t = \left\lceil n_r \frac{v_r}{v_t} \right\rceil. \tag{4.56}$$

The strategy is to find n_r such that $n_r n_t \leq 1/\gamma$, n_t satisfies (4.56) and n_r is as large as possible. This means

$$\begin{aligned} n_r \left\lceil n_r \frac{v_r}{v_t} \right\rceil &\leq \frac{1}{\gamma}, \\ (n_r+1)\left\lceil (n_r+1)\frac{v_r}{v_t} \right\rceil &> \frac{1}{\gamma}. \end{aligned} \tag{4.57}$$

Simplifying the above yields

$$\begin{aligned} n_r &\leq \sqrt{\frac{1}{\gamma}\frac{v_t}{v_r}}, \\ (n_r+1) &> \frac{1}{2}\frac{v_t}{v_r}\left[-1+\sqrt{\frac{4v_r}{\gamma v_t}+1}\right]. \end{aligned} \tag{4.58}$$

Now, (4.46) will still be valid. Using the inequalities and noting from (4.56) that $\frac{v_r}{n_t} \leq \frac{v_t}{n_r}$ and $n_t \leq n_r \frac{v_r}{v_t} + 1$, we obtain

$$v_r N_r + v_t N_t \leq \frac{N v_t}{n_r} \left(2N + n_r + \frac{n_r}{v_t / v_r} + 1 \right). \tag{4.59}$$

Substituting n_r from (4.58) yields

$$v_r N_r + v_t N_t \leq \frac{N v_t}{\frac{1}{2} \frac{v_t}{v_r} \left[-1 + \sqrt{\frac{4 v_r}{\gamma v_t} + 1} \right] - 1} \left(2N + \left(1 + \frac{v_r}{v_t} \right) \sqrt{\frac{v_t}{\gamma v_r}} + 1 \right). \tag{4.60}$$

Rearranging the above gives

$$v_r N_r + v_t N_t \leq \frac{2\sqrt{v_t v_r} N (N-1) \sqrt{\gamma}}{\sqrt{1 + \frac{\gamma v_t}{4 v_r}} - \sqrt{\frac{\gamma v_r}{v_t}} - \sqrt{\frac{\gamma v_t}{4 v_r}}} \left[1 + \frac{3\sqrt{\gamma} + (1 + v_r/v_t)\sqrt{v_t/v_r}}{2(N-1)\sqrt{\gamma}} \right]. \tag{4.61}$$

It is easy to see that the denominator of the first term of the above product is $1 - \delta_1(\gamma)$ where $\delta_1 \geq 0$ and $\delta_1 \to 0$ as $\gamma \to 0$. The second term of the product is $1 + \delta_2(\gamma, N)$ where $\delta_2 \to 0$ as $\gamma \to 0$ if $(N-1)^x \gamma \geq k$, a constant, and $0 \leq x \leq 2$. Hence, we conclude that

$$v_r N_r + v_t N_t \leq 2\sqrt{v_t v_r \gamma} N (N-1)(1 + \delta(\gamma, N, x)) \tag{4.62}$$

where $\delta \to 0$ as $\gamma \to 0$ so that the difference between the upper bound and the lower bound becomes insignificant.

For the bound on the channel cost, we have

$$N_c = \left\lceil \frac{N}{n_r} \right\rceil \left\lceil \frac{N}{n_t} \right\rceil = \frac{N_t N_r}{N^2}. \tag{4.63}$$

We note from the above that the procedure used to bound N_c can actually be used to obtain a bound for $N_t N_r$ too. Substituting (4.46) into the above,

$$\begin{aligned} N_c &\leq \frac{1}{n_r n_t} (N + n_r)(N + n_t) \\ &\leq \frac{1}{n_r (n_r \frac{v_r}{v_t})} (N + n_r) \left(N + \left(n_r \frac{v_r}{v_t} \right) + 1 \right) \end{aligned} \tag{4.64}$$

Substituting n_r from (4.58) and arranging, we get

$$N_c \leq \frac{N(N-1)\gamma\left[1+\sqrt{\frac{v_t}{v_r\gamma N^2}}\right]\left[1+\frac{\sqrt{\frac{v_r}{v_t}}+2\sqrt{\gamma}}{(N-1)\sqrt{\gamma}}\right]}{\left[\sqrt{1+\frac{\gamma v_t}{4v_r}}-\sqrt{\frac{\gamma v_t}{4v_r}}-\sqrt{\frac{\gamma v_r}{v_t}}\right]^2}. \tag{4.65}$$

Using the same argument as before,

$$N_c \leq N(N-1)\gamma(1+\epsilon(\gamma,N)) \tag{4.66}$$

where $\epsilon \to 0$ as $\gamma \to 0$ provided $(N-1)^x\gamma \geq k$ for $0 < x < 2$. The discussion is summarized in the following theorem.

Theorem 4.5 *Consider the uniform traffic situation. Suppose* $(N-1)^x\gamma \geq k$ *where* k *is a constant and* $0 < x < 2$. *Then,* $v_rN_r + v_tN_t + wN_c \leq 2N(N-1)\sqrt{v_tv_r\gamma}(1+\delta(\gamma,N)) + wN(N-1)\gamma(1+\epsilon(\gamma,N))$ *where* $\delta, \epsilon \to 0$ *as* $\gamma \to 0$.

The requirement $(N-1)^x\gamma \geq k$ implies the total traffic

$$N(N-1)\gamma \approx (N-1)^2\gamma \geq (N-1)^{2-x}k \to \infty$$

as $\gamma \to 0$. The interpretation of the theorem is that under conditions with large total traffic but small individual traffic between two nodes, simple heuristics yield very good results. It seems, at least intuitively, this will hold even for the nonuniform traffic case. However, the analysis may be much more involved because of the increase in the "degrees of freedom" of the traffic values.

4.5 CAP with n-Connectivity Requirement

The last subject in this chapter is CAP with n-connectivity ($n > 0$) requirement. That is, there are at least n alternative channels for the transmission of each traffic type. Among the motivations for such an arrangement are reliability, flexibility in access control, and perhaps an improvement in performance.

An easy heuristic that deals with this new requirement is as follows. Form a new traffic matrix by dividing each traffic entry in the original matrix by n;

$$\lambda'^{(p,q)} = \frac{\lambda^{(p,q)}}{n}. \tag{4.67}$$

We then proceed by applying an algorithm that solves the 1-connectivity problem on this new traffic matrix: n replicas of the resulting mapping constitute a solution to the n-connectivity case, with the understanding that the same channel labels on different copies of the map are actually different channels. This heuristic has the appeal that $\lambda'^{(p,q)}$ are likely to be small for large n, and therefore the resulting packing problem of the 1-connectivity problem is easy and a good solution can be obtained. Thus, the resulting n-connectivity solution must be reasonably good.

Let's consider the implication of n for N_t and N_r for the uniform-traffic CAP. Assuming no traffic splitting, we know from (4.22) that for each individual map

$$N_t N_r \geq N^2(N-1)^2\gamma/n. \tag{4.68}$$

But for the whole system, N_t and N_r are n times that given by the above. Hence

$$N_t N_r \geq N^2(N-1)^2 n\gamma. \tag{4.69}$$

Therefore, we should expect in the results obtained, N_t (or N_r) are roughly proportional to $\sqrt{n}$ in the large total traffic and small individual traffic limit. If $(N-1)^2\gamma < n$, however, a better bound is n^2N^2 — each node needs at least n transmitters and n receivers.

Theorem 4.6 *Consider the uniform traffic CAP with n-connectivity requirement. Using the n-replica strategy,* $N_t N_r \geq \max[N^2(N-1)^2 n\gamma, n^2N^2]$.

Comment: Unlike the 1-connectivity case, this bound is scheme dependent. It is conceivable that other heuristics will yield better bounds.

A slight modification to the above scheme is as follows. Suppose we use the 1-connectivity heuristic given at the end of Section 4.3 repeatedly to solve the n-connectivity problem. Notice that for the example in Fig. 4.12, some traffic entries are split (e.g. traffic $(7,2)$). As a result, using the above method, these entries would have $2n$ accessible channels. It is likely that a better solution can be obtained by eliminating these redundancies. As before, for the first mapping, we consider a traffic matrix with entries as in (4.67). However, whenever there is a split in traffic, instead of keeping it for the same mapping, the remaining traffic will be "transferred" to the traffic matrix of the subsequent mappings (e.g. divide the residual traffic equally among the $(n-1)$ subsequent traffic matrices.). Also, the last channel considered may not be fully packed, as in channel 7 in Fig. 4.12. To make packing easier in the subsequent mappings, we try to pack this channel completely by transferring some traffic from the same traffic entries in the subsequent traffic matrices to the current matrix until this last channel is completely full. Of course, we have to be careful not to reduce any entry in the subsequent traffic matrices to zero, else the n-connectivity requirement will not be satisfied. Note that in this method, a modification of the 1-connectivity heuristic is applied for each resulting map and a total of n applications are needed. Instead of an $O(N^4)$ algorithm as in the first method, we now have an $O(nN^4)$ algorithm.

Yet another scheme is to use a 1-connectivity solution for the original traffic matrix. In addition, we introduce $(n-1)$ extra channels on which all nodes can transmit and receive. It can be easily shown that this yields a better solution if the total system traffic is much greater than $(n-1)$.

The above analysis assumes the effective capacity is independent of the mapping schemes. Chapter 7 reexamines this assumption and shows that it is not always justified.

Chapter 5

Block Designs

The problem of block designs is a combinatorial problem well studied by mathematicians and papers on the subject can be traced back to the nineteenth century. The particular form of block designs that are relevant to the study here is the so-called balanced incomplete block designs and it is defined as follows:

Definition 5.1 A balanced incomplete block design $D(N_c, N, n_{rt}, k, n)$ is an arrangement of N distinct objects into N_c blocks such that each block contains k distinct objects, each object belongs to n_{rt} blocks and every pair of distinct objects occurs together in exactly n blocks.

Fig. 5.1 is an example of a block design with $N_c = 7$, $N = 7$, $n_{rt} = 3$, $k = 3$, $n = 1$. Section 5.1 relates block designs to CAP. Section 5.2 discusses some known results of block designs and Section 5.3 shows some applications and extensions of results to frameworks closer to the original CAP.

5.1 Relating Block Designs to CAP

In terms of CAP, a block corresponds to a channel and an object to a communication node. The problem can be viewed as a uniform-traffic capacity assignment problem with the additional constraints:

(1, 2, 4)	(1, 5, 6)
(2, 3, 5)	(0, 2, 6)
(3, 4, 6)	(0, 1, 3)
(0, 4, 5)	

(i, j, k) = A block containing objects i, j and k.

Figure 5.1: A Block Design $D(7, 7, 3, 3, 1)$.

Constraints 5.1

(a) A node's transmittable channels are the same as its receivable channels, i.e. n_{rt} corresponds to both node transmittability and node receivability.

(b) All nodes access the same number of channels.

(c) Any two nodes share exactly n common channels (i.e n-connectivity).

(d) Every channel is accessed by the same number of nodes (i.e. k).

Because of the symmetries implied by these constraints, the block design problem is more akin, but not similar, to the capacity assignment problem with uniform traffic between nodes. Another way to look at it is that the traffic intensity aspect of CAP is ignored in the block design formulation.

Consider an instance of CAP with 7 nodes and $\lambda^{(p,q)} = 1/6$ for all (p, q), $p \neq q$. Then the block design in Fig. 5.1 gives a solution with $N_c = 7$, $N_t = N_r = n_{rt}N = 21$ and there are 3 transmitters and 3 receivers attached to each channel. The total traffic within a block (channel) is exactly 1. The assignments given in Fig. 5.2, on the other hand, yield $N_c = 7$, $N_r = 20$, $N_t = 19$, a better solution. This is

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
T_1	×	$\frac{1}{6}$ 1	$\frac{1}{6}$ 1	$\frac{1}{6}$ 2	$\frac{1}{6}$ 2	$\frac{1}{6}$ 3	$\frac{1}{6}$ 3
T_2	$\frac{1}{6}$ 1	×	$\frac{1}{6}$ 1	$\frac{1}{6}$ 2	$\frac{1}{6}$ 2	$\frac{1}{6}$ 3	$\frac{1}{6}$ 3
T_3	$\frac{1}{6}$ 1	$\frac{1}{6}$ 1	×	$\frac{1}{6}$ 2	$\frac{1}{6}$ 2	$\frac{1}{6}$ 3	$\frac{1}{6}$ 3
T_4	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4	×	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4
T_5	$\frac{1}{6}$ 5	$\frac{1}{6}$ 5	$\frac{1}{6}$ 6	$\frac{1}{6}$ 6	×	$\frac{1}{6}$ 7	$\frac{1}{6}$ 7
T_6	$\frac{1}{6}$ 5	$\frac{1}{6}$ 5	$\frac{1}{6}$ 6	$\frac{1}{6}$ 6	$\frac{1}{6}$ 7	×	$\frac{1}{6}$ 7
T_7	$\frac{1}{6}$ 5	$\frac{1}{6}$ 5	$\frac{1}{6}$ 6	$\frac{1}{6}$ 6	$\frac{1}{6}$ 7	$\frac{1}{6}$ 7	×

Figure 5.2: A CAP Solution Better than that Derived from Fig. 5.1

not surprising since there are more constraints in the block design formulation. On the other hand, it is conceivable that the symmetries in block designs may lend themselves to facilitating other aspects of network design. It is also reasonable to use the block design formulation when the capabilities of transmitting and receiving on a channel come in a single package, say a transceiver.

The optimization issue of the capacity assignment is not addressed in the block design problem. This is understandable since CAP-like optimization problems are not the original motivation for the study of block design. For instance, an intended application of block designs is the symmetrical arrangement of sets of experiments in order to diffuse unwanted correlations, as applied in statistics [11]. For our purpose, we would like to have small values of N_c and n_{rt} given some N, n and the traffic intensity. It is difficult to incorporate the optimization aspect since the construction of block designs is itself a very difficult problem: only block designs with certain sets of parameters are known. The emphasis of this chapter will be on the implications of the additional constraints for the capacity assignment problem.

5.2 Some Theorems and Known Results

This section discusses some important known mathematical properties of block designs to prepare for the next section, in which we extend and apply these results in the framework of CAP. The reader may choose to read the next section first, and come back to this section only when necessary. Some theorems will be stated without proofs, but examples and comments will be added to put the meanings into the proper context. The construction methods of block designs will only be briefly overviewed here.

The simplest and perhaps the most fundamental relationship between the parameters (N_c, N, n_{rt}, k, n) are given in Theorem 5.1.

Theorem 5.1

$$N_c k = N n_{rt}, \quad (5.1)$$

$$n_{rt}(k-1) = n(N-1). \quad (5.2)$$

Proof: (5.1) counts the total number of the "occurrences" of objects in two ways. Each block contains k objects and there are N_c blocks. Hence, there are $N_c k$ occurrences. Also, each object belongs to n_{rt} blocks and there are N objects. Hence, there are $N n_{rt}$ occurrences. (5.2) counts the number of pairs containing a particular object. The object occurs in n_{rt} blocks and in each of these blocks, it pairs with $(k-1)$ other objects. Hence, the number of pairs containing the object is $n_{rt}(k-1)$. But the object occurs n times together with each of the other $(N-1)$ objects. Hence, the number of pairs containing the object is also $n(N-1)$.

□

Theorem 5.1 implies at most three of the five parameters are free. The fact that only integral values are allowed further restricts the choice of parameter values. Also, as will be seen, Theorem 5.1 is necessary but not sufficient for the existence of a block design.

Theorem 5.2 *If $n_{rt} > n$, then $N_c \geq N$ and $n_{rt} \geq k$.*

Comment: Notice that $n_{rt} \geq n$ by definition, i.e. two objects cannot be paired more than n_{rt} times if an object belongs to only n_{rt} blocks. In the case $n_{rt} = n$, we have a trivial design in which all the objects belong to every block.

Proof: Define an $N \times N_c$ incidence matrix $\mathbf{A}$; $a_{ij} = 1$ if object i belongs to block j and $a_{ij} = 0$ otherwise. Let

$$\mathbf{B} = \mathbf{A}\mathbf{A}^T = \begin{pmatrix} n_{rt} & n & \cdots & n \\ n & n_{rt} & \cdots & n \\ \vdots & \vdots & \ddots & \vdots \\ n & n & \cdots & n_{rt} \end{pmatrix} \tag{5.3}$$

where the second equality is implied by the definition of block design; i.e. b_{ij} = the number of blocks in which Objects i and j occurs together. Let

$$\mathbf{C} = \begin{pmatrix} n_{rt} + (N-1)n & 0 & 0 & \cdots & 0 \\ n & n_{rt} - n & 0 & \cdots & 0 \\ n & 0 & n_{rt} - n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & 0 & 0 & \cdots & n_{rt} - n \end{pmatrix}. \tag{5.4}$$

$\mathbf{C}$ can be obtained from $\mathbf{B}$ by subtracting the first column of $\mathbf{B}$ from all the other columns and then adding all the other rows to the first row. Hence,

$$\det \mathbf{B} = \det \mathbf{C} = \left(n_{rt} + (N-1)n\right)\left(n_{rt} - n\right)^{N-1}. \tag{5.5}$$

Since $n_{rt} > n$, $\det \mathbf{B} \neq 0$. Therefore, rank $\mathbf{A} \geq$ rank $\mathbf{A}\mathbf{A}^{\mathbf{T}}$ = rank $\mathbf{B} = N$, implying $N_c \geq N$. By (5.1), $n_{rt} \geq k$.

□

We see that the number of channels in the corresponding CAP must be at least the number of communication nodes, regardless of the total system traffic. This may not be desirable in low traffic situations. In particular, by removing some of the symmetry requirements (e.g. allowing different pairs of objects to occur together different numbers of times, i.e. letting the connectivity $\geq n$ instead of fixing it), it is possible to remove the restriction of Theorem 5.2. This will be discussed in the next section.

Definition 5.2 A *symmetric block design* $D(N, k, n)$ is a design in which $N_c = N$. By (5.1), this also means $n_{rt} = k$.

Theorem 5.3 *If $D(N, k, n)$ exists, then*

1. *N is even implies $(k - n)$ is an integer square;*
2. *N is odd implies there exist integers x, y, z, not all zero, such that* $z^2 = (k - n)x^2 + (-1)^{\frac{1}{2}(N-1)} n y^2$.

Proof: Omitted; see [17].

□

Theorem 5.3 applies to symmetric block designs. There are also nonexistence theorems on asymmetric block designs. Basically, they rely on the concept of completion and embedding. It can be shown that asymmetric block designs can be completed to form symmetric block designs by adding more object and blocks. If the resulting symmetric block designs do not exist, then the corresponding asymmetric designs do not exist. Thus, Theorem 5.3 is a rather useful result.

Theorem 5.4 *Let* $\mathbf{A}$ *be a real nonsingular* $N \times N$ *matrix satisfying either*

$$\mathbf{A}\mathbf{A}^T = (k-n)\mathbf{I} + n\mathbf{J} \tag{5.6}$$

or

$$\mathbf{A}^T\mathbf{A} = (k-n)\mathbf{I} + n\mathbf{J} \tag{5.7}$$

and either

$$\mathbf{A}\mathbf{J} = k\mathbf{J} \tag{5.8}$$

or

$$\mathbf{J}\mathbf{A} = k\mathbf{J}, \tag{5.9}$$

where $\mathbf{I}$ *is the identity matrix and* $\mathbf{J}$ *a matrix with all elements 1's. Then* $\mathbf{A}$ *satisfies all four relations and* N, k *and* n *satisfy*

$$k(k-1) = n(N-1). \tag{5.10}$$

Proof: Omitted; see [17].

□

Basically, Theorem 5.4 implies the dual property of symmetric block design: if the block labels and node labels are interchanged, a symmetric block design of the same parameters is obtained; a corollary is that two blocks must have n common nodes. The theorem is also employed in some block design construction method.

The above theorems all deal with the necessary conditions on block designs. Other than those of known block designs, there are no general sufficiency tests

---

Theorem 5.5 *The conditions in* Theorem 5.1 *are necessary and sufficient for a block design with* $4 \geq k \geq 2$ *and any* n.

Proof: Omitted; see [17].

□

Block design construction methods can be categorized into two classes. Direct construction methods construct block designs in a number of ways, including using finite fields, finite geometries and finite groups. There are also methods based on number theory and algorithmic approaches. Indirect methods construct new designs from known ones. This can either be recursive methods that construct a larger design based on smaller ones or methods that derive smaller designs by eliminating a subset of objects or blocks in a larger design.

Perhaps the most elegant and the most general method is the direct construction method based on finite geometries. In this method, the points and hyperplanes in a finite geometric construct are mapped in a *one-to-one* fashion *onto* the objects and blocks of a block design. Two objects occur in at least a common block since two points belong to at least a common hyperplane. Hence, the problem of block designs is transformed to the problem of constructing finite geometries and identifying the resulting hyperplanes.

Because of the extensive amount of material, the reader is referred to [17,32] for further details of Block Designs. A more introductory treatment is given in [34]. Applications of Block Designs on Statistics are discussed in [11]. For a list of known designs, refer to Appendix I of [17].

5.3 Application and Extension of Results to CAP

We now apply and extend the results of the preceding section in the context of CAP.

5.3.1 Traffic Consideration

Consider an instance of CAP with $N = 7$. If $42\gamma \leq 1$, then the problem is trivial and we simply assign a single channel to all the nodes. Now if $42\gamma = 6$, we can still assign 6 channels such that each is accessed by all nodes and a traffic type is split equally among the 6 available channels (i.e. connectivity $n = 6$). Thus, no capacity is wasted and $n_{rt} = 6$. An optimization question is whether we can minimize n_{rt} by formulating this problem in terms of a block design problem with $n = 1$. According to Theorem 5.2, $N_c \geq N \geq 7$ so that at least a capacity of 1 will be wasted. Fixing $N_c = 7$ in Theorem 5.1 yields

$$n_{rt}(n_{rt} - 1) = 6n. \tag{5.11}$$

Hence, if $n = 1$ then $n_{rt} = 3$. In fact, Fig. 5.1 is a block design with the above parameter values. In general, much channel capacity will be wasted if $N(N-1)\gamma \ll N$ since redundant channels must be added for the feasibility of the associated block design. Note that this is not so with the original CAP.

5.3.2 The Constraints Responsible for $N_c \geq N$

Let's investigate which are the essential constraints among Constraints 5.1 that result in $N_c \geq N$. To do so, we first relax each of the four listed constraints.

(i) Drop Constraint 5.1 (a). We must be careful about what the other constraints mean here. Constraints 5.1 (a), (b) and (c) are redefined as follows:

- Any given node can transmit on n_t channels and receive on n_r channels.

	R_1	R_2	R_3	R_4	R_5	R_6
T_1	$\times$	$\frac{1}{6}$ 1	$\frac{1}{6}$ 1	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2
T_2	$\frac{1}{6}$ 1	$\times$	$\frac{1}{6}$ 1	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2
T_3	$\frac{1}{6}$ 1	$\frac{1}{6}$ 1	$\times$	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2	$\frac{1}{9}$ 2
T_4	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\times$	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4
T_5	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{6}$ 4	$\times$	$\frac{1}{6}$ 4
T_6	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{9}$ 3	$\frac{1}{6}$ 4	$\frac{1}{6}$ 4	$\times$

$N_c = 4$, $N = 6$, $n_r = n_t = 2$, $k_r = k_t = 3$, $n = 1$.

Figure 5.3: An Example in whcih Transmittable Channels Different from Receivable Channels and $N_c < N$.

- Given any pair of nodes, say p and q, there are exactly n channels on which p can transmit and q can receive.

- There are exactly k_t nodes which can transmit and k_r nodes which can receive on any given channel.

We claim that the above constraints do not imply $N_c \geq N$. It suffices to give the example in Fig 5.3.

(ii) Drop Constraint 5.1 (c). Instead we require any pair of nodes to have at least, rather than exactly, n common accessible channels. It is easy to show that $N_c \geq N$ is not true anymore even if we keep the other constraints. Consider a case with $N_c = 3$, $N = 6$, $n_{rt} = 2$, $k = 4$ and $n \geq 1$. An example satisfying the above specifications is as follows: $(0,1,2,3)$, $(2,3,4,5)$, $(0,1,4,5)$. We see that there are two common channels between nodes 0 and 1, between nodes 2 and 3, and between nodes 4 and 5. However, there is only a common channel between nodes from different pairs as grouped above. Also, note that $N > N_c$.

(iii) Drop Constraint 5.1 (b). Instead of n_{rt}, node i can access $n_{rt}^{(i)}$ channel. If all the other constraints are kept, (5.2) becomes $n_{rt}^{(i)}(k-1) = n(N-1)$ for all i (recall that this is the number of pair occurrences that include node i). Consequently, $n_{rt}^{(i)}$ is the same for all i and dropping this constraint without dropping at least another constraint does not relax the problem at all.

(iv) Drop Constraint 5.1 (d). Let $k^{(j)}$ denote the number of nodes belonging to channel j. If we keep all the other constraints, we see from the proof of Theorem 5.2 that $N_c \geq N$ will still hold since the proof is independent of the assumption that $k^{(j)}$ is the same for all j.

(v) We see that cases (iii) and (iv) above do not relax the requirement $N_c \geq N$. Suppose we combine the two cases. In other words, we drop Constraints 5.1 (b) and (d) together. It can be shown that N_c is still $\geq N$. For clarity of presentation, this will be stated in a theorem and proved thereafter.

Definition 5.3 A *relaxed* block design $D_r(N_c, N, n)$ is an arrangement of N distinct objects into N_c blocks such that every pair of distinct objects occurs together in exactly n blocks.

As in the previous cases, $n_{rt}^{(i)}$ will denote the number of blocks containing object i and $k^{(i)}$ will denote the number of objects block i contains.

Theorem 5.6 *In a relaxed block design, unless every block contains all the objects,* $N_c \geq N$.

Proof: Suppose there exists i such that $n_{rt}^{(i)} = n$. Then all other nodes also occur in the n channels in which object i occurs. In particular, any $j \neq i$ and $k \neq i$ also occur together n times in these n blocks. This is the trivial design in which every single block contains all the objects. Therefore we require $n_{rt}^{(i)} > n$ for all i. Define an $N \times N_c$ incidence matrix $\mathbf{A}$; $a_{ij} = 1$ if object i belongs to block j and $a_{ij} = 0$ otherwise. Let

$$\mathbf{B} = \mathbf{A}\mathbf{A}^T = \begin{pmatrix} n_{rt}^{(1)} & n & \cdots & n \\ n & n_{rt}^{(2)} & \cdots & n \\ \vdots & \vdots & \ddots & \vdots \\ n & n & \cdots & n_{rt}^{(N)} \end{pmatrix}. \tag{5.12}$$

Using the same argument as in the proof of Theorem 5.1, it suffices to show $\det \mathbf{B} \neq 0$. Let

$$\mathbf{C_N} = \begin{pmatrix} n_{rt}^{(1)} & n - n_{rt}^{(1)} & n - n_{rt}^{(1)} & \cdots & n - n_{rt}^{(1)} \\ n & n_{rt}^{(2)} - n & 0 & \cdots & 0 \\ n & 0 & n_{rt}^{(3)} - n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & 0 & 0 & \cdots & n_{rt}^{(N)} - n \end{pmatrix}. \tag{5.13}$$

$\mathbf{C_N}$ can be obtained from $\mathbf{B}$ by subtracting the first column of $\mathbf{B}$ from all the other columns and therefore their determinants are equal. We will prove by induction that $\det \mathbf{C_N} > 0$. Assume $\det \mathbf{C_{N-1}} > 0$ and it is easy to show by direct verification that $\det \mathbf{C_2}$ and $\det \mathbf{C_3}$ are greater than 0. Expanding the determinant of $\mathbf{C_N}$ about the last row, we have

$$(-1)^{N+1} n \begin{vmatrix} n - n_{rt}^{(1)} & n - n_{rt}^{(1)} & \cdots & n - n_{rt}^{(1)} & n - n_{rt}^{(1)} \\ n_{rt}^{(2)} - n & 0 & \cdots & 0 & 0 \\ 0 & n_{rt}^{(3)} - n & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & n_{rt}^{(N-1)} - n & 0 \end{vmatrix} + (-1)^{2N}(n_{rt}^{(N)} - n)\,|\mathbf{C_{N-1}}|. \tag{5.14}$$

The second term above is positive by hypothesis. Expanding the first term about the last column yields

$$(-1)^{N+1}(-1)^N n(n - n_{rt}^{(1)})(n_{rt}^{(2)} - n)\cdots(n_{rt}^{(N-1)} - n), \tag{5.15}$$

which is also positive. Hence, $\det \mathbf{C_N} > 0$.

□

As shown above, the requirement $N_c \geq N$ in the block design formulation of CAP is due entirely to the combination of Constraints 5.1 (a) and (c) only. In particular, even in practical situation where the transmittable channels of a node are the same as its receivable channels, it is not necessary to have a larger number of channels than the number of nodes, as long as we do not insist on exactly n common channels between any pair of nodes.

On the other hand, one can easily think of situations in which exact connectivity is desired. An example is a multichannel broadcast system in which the messages originating from a node are broadcast to the other nodes by transmitting the messages on all its accessible channels. Suppose it is desired, because of some underlying access scheme, that there is one and only one channel through which a message can travel from one node to another. Barring letting all nodes access all channels, this cannot be done if a transmitter and a receiver comes in a single package as a transceiver operating on a single channel, and the number of available channels is less than the number of nodes.

5.3.3 A Lower Bound on $(Nn_{rt})^2$

Let's try to obtain a relationship similar to (4.22) for the block design approach. Suppose γ is the traffic from one node to another. two nodes share n common channels. Then the traffic between any two nodes within a single common channel is γ/n, if it is divided evenly among the n channels. Therefore, the total traffic carried on a channel is

$$k(k-1)\frac{\gamma}{n} \leq 1. \tag{5.16}$$

Substituting k with the value obtained from (5.2) yields

$$\frac{n}{\gamma} \geq \left(\frac{n}{n_{rt}}(N-1)+1\right)\left(\frac{n}{n_{rt}}(N-1)\right). \tag{5.17}$$

A lower bound on n_{rt} as shown below can be obtained from the above, and it is

$$n_{rt} \geq \frac{\gamma(N-1)}{2} + \sqrt{\left(\frac{\gamma(N-1)}{2}\right)^2 + n\gamma(N-1)^2}. \tag{5.18}$$

The number of transmitter (or receivers) in the system is Nn_{rt} and

$$(Nn_{rt}) \geq \frac{N(N-1)\gamma}{2}\left(1+\sqrt{1+\frac{4n}{\gamma}}\right) \tag{5.19}$$

$$> N(N-1)\sqrt{n\gamma}, \tag{5.20}$$

where the second inequality is obtained by discarding the two 1's on the right side of the first inequality. Letting $n = 1$, we obtain

$$(Nn_{rt})^2 > N^2(N-1)^2\gamma. \tag{5.21}$$

Comparing this to (4.22), we note that the inequality in (4.22) has been replaced by a strict inequality. This is not surprising since the block design formulation involves more constraints. Also, (5.20) is a close approximation to (5.19) if n/γ is large relative to 1. This means high connectivity or low individual traffic. Note that $n/\gamma \geq 1$ so that (5.19) is at most $(1+\sqrt{5})/2$ or approximately 1.618 times (5.20). This factor will be even smaller if $\gamma \ll 1$. So, under a perfect fitting condition when (5.16) is satisfied with equality (this is not a trivial condition that can be achieved easily), (5.19) is satisfied with equality; and we would expect the number of transmitters and receivers required by the block design approach to be not too much larger than that given by the heuristic traffic matrix mapping approach using "square" mapping. Of course, this assumes the cost of a transmitter is roughly the same as the cost of a receiver.

5.3.4 Solving CAP Using a Block Design: An Example

We now consider using the block design approach to solve an instance of CAP. Suppose $\gamma = 1/7$, $N = 20$ and the required n is 1. From (5.16), the largest possible k is 3. However, given the above values of N, k, n, (5.2) or $n_{rt}(k-1) = n(N-1)$ will not yield an integer n_{rt}. To solve this problem, a fictitious node is introduced so that $N = 21$. The resulting n_{rt} is 10. Finally, we look up a table of block designs to see if a design with the above parameters is available. According to Appendix I of [17], a solution is $[0,1,13]$ mod 21, $[0,4,10]$ mod 21, $[0,16,19]$ mod 21 and $[0,7,14]$ mod 21 period 7. In other words, the design is given by $[(0+n)$ mod $21, (1+n)$ mod $21, (13+n)$ mod $21]$, $[(0+n)$ mod $21, (4+n)$ mod $21, (10+n)$ mod $21]$, $[(0+n)$ mod $21, (16+n)$ mod $21, (19+n)$ mod $21]$ for $n = 0, 1, \ldots, 21$ and $[(0+3m)$ mod $21, (7+3m)$ mod $21, (14+3m)$ mod $21]$ for $m = 0, 1, 2, 3$. A solution to the original CAP is obtained by eliminating the fictitious node, say node 21, from the design. There are altogether 70 blocks. Since $n_{rt} = 10$, the resulting N_r or N_t is $20n_{rt} = 200$.

5.3.5 Existence of Block Designs: Examples

We now consider the existence problem of symmetric block designs. First, let's see whether a symmetric block design with $N_c = N = 56$, $n = 1$ exists. Using (5.1) and (5.2), we have $k(k-1) = 55$. But for integer k, $k(k-1)$ is even. Therefore, the block design does not exist.

Now suppose $N_c = N = 56$ but $n = 2$. Using Theorem 5.1 again, $k(k-1) = 110$ giving $k = 11$. $(k-n) = (k-2) = 9$ is an integer square, and this satisfies the necessary condition in Theorem 5.3. Therefore, it is possible, though not guaranteed, that such a design exists.

Consider $N_c = N = 211$ and $n = 1$. Theorem 5.1 yields $k = 15$. Applying

Theorem 5.3, a necessary condition for the existence of such a design is that integers x, y, z, not all zero, satisfying

$$z^2 + y^2 = 14x^2 \tag{5.22}$$

can be found. This is not possible and it can be seen as follows. Since the right side of (5.22) is even, z and y must be either both odd or both even. Consider the fact that x can be either even or odd, we have four cases altogether. If x, y and z are all even, say $x = 2x'$, $y = 2y'$, $z = 2z'$. Substituting into (5.22) and canceling out the factor of 4, we obtain an equation of the same form. Doing this repeatedly will finally yield an equation in which not all x, y and z are even. So, it is enough to consider just the three other cases.

(i) x odd, y even, z even.
Let $y = 2y'$, $z = 2z'$. Equation (5.22) becomes $z'^2 + y'^2 = \frac{7}{2}x^2$. The left side of the equation must be an integer but the right side cannot be since x^2 is odd.

(ii) y odd, z odd.
Let $z = 2z' + 1$, $y = 2y' + 1$ and $x = 2x' + \delta$ where $\delta = 1$ if x is odd and 0 otherwise. Substituting into (5.22) and arranging yields

$$z'^2 + y'^2 + z' + y' = 14(x'^2 + x'\delta) + \frac{7}{2}\delta - \frac{1}{2}. \tag{5.23}$$

The left side is always even. If $\delta = 0$, the right side is not an integer. If $\delta = 1$, the right side is odd.

Hence, the block design is not possible. The above argument can be generalized to show that there is no symmetric block design with parameter values, $n = 1$, $k = 4k' + 3$ for all k' odd. Note that N is determined by $k(k-1) = n(N-1)$ and it is not a free parameter once k is specified. Furthermore, for n odd, N is also odd.

Chapter 6

Hierarchical Networks

This chapter concerns the introduction of hierarchical structures into the multi-channel networks. We will not consider the constraints imposed by the block design formulation further. A logical description of hierarchical networks will be given. The purpose here is not to identify a fundamental definition that covers a wide range of hierarchical networks described in literature. Rather, the emphasis is on a simple definition that is relevant to the study here. In particular, the goal here is to show that it is possible to reduce N_t and N_r by adopting certain hierarchical structures.

6.1 Description of Hierarchical Networks

Thus far, the discussions in the preceding chapters have been limited to the network structure depicted in Fig. 6.1(a). Here, all nodes are "tapping" onto a network without any internal switches. Fig. 6.1(b) and (c) are networks of higher hierarchical orders. In the pictures, a rectangle represents a nonswitching subnetwork and a circle represents a communication node. A line is drawn between a node and a subnetwork if the node can receive and transmit on the subnetwork directly, and the node is said to belong to the subnetwork. Further, the nodes belonging to the same subnetwork can communicate with each other using the subnetwork alone. The collection of such nodes will be called a *group.*

Communication between subnetworks is achieved via bridges, the darkened circles in the figures. As with nodes, a line is drawn between a bridge and a subnetwork if it can transmit and receive on the subnetwork, and bridges with lines drawn to the same subnetwork can communicate directly through the subnetwork. Consider the level-1 subnetwork in Fig. 6.1(b). As far as it is concerned, the bridges are like communication nodes, although the traffic to and from a bridge is actually the aggregate traffic of many nodes at the level 0. Hence, the corresponding CAP at level 1 is exactly the same as that we discussed before and the traffic entries in the traffic matrix will simply be the inter-group traffic. The traffic in a subnetwork at level 0 consists of both intra-group traffic and inter-group traffic. Traffic with destinations outside the group will be directed to a bridge. Thus, if there are m nodes in a group, an $(m+1) \times (m+1)$ traffic matrix will be needed to describe the traffic in the subnetwork.

Fig. 6.1(c) is simply a higher order hierarchical network. The relationships between the entities are implicit from the above discussion. In general, a node can be referred to as a level-0 node and a bridge at level n as a level-n node since the distinction between nodes and bridges is not important as far as CAP is concerned. For the sake of a more precise definition, the following outline can be used to determine the levels of the entities in a network.

- We assume the level-0 nodes are known, implicit or predetermined.
- The level of a subnetwork is specified by the lowest level node connected to it.
- The level of a node, not at level 0, is 1 plus the level of the lowest level

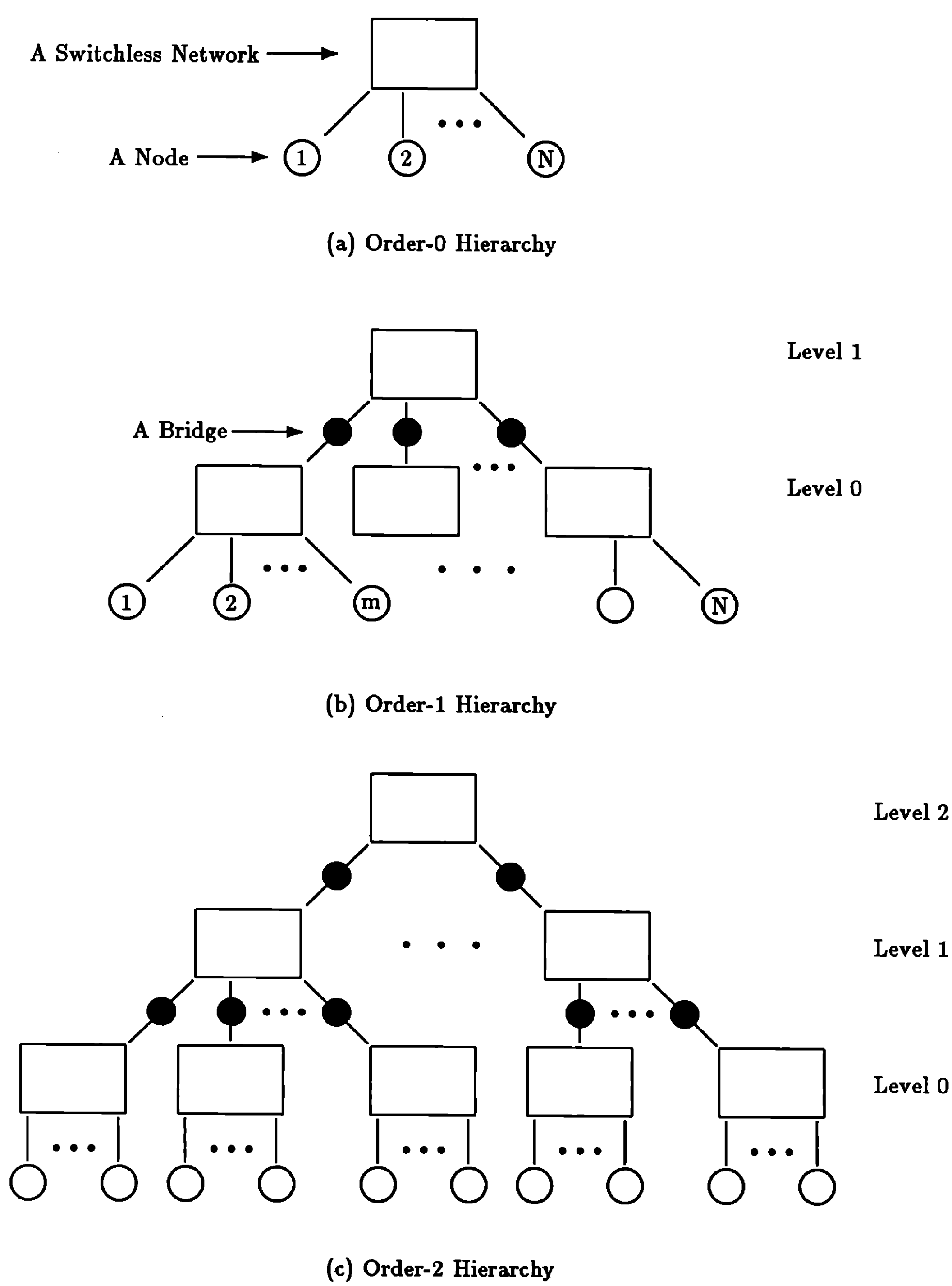


Figure 6.1: Hierarchical Networks of Different Orders

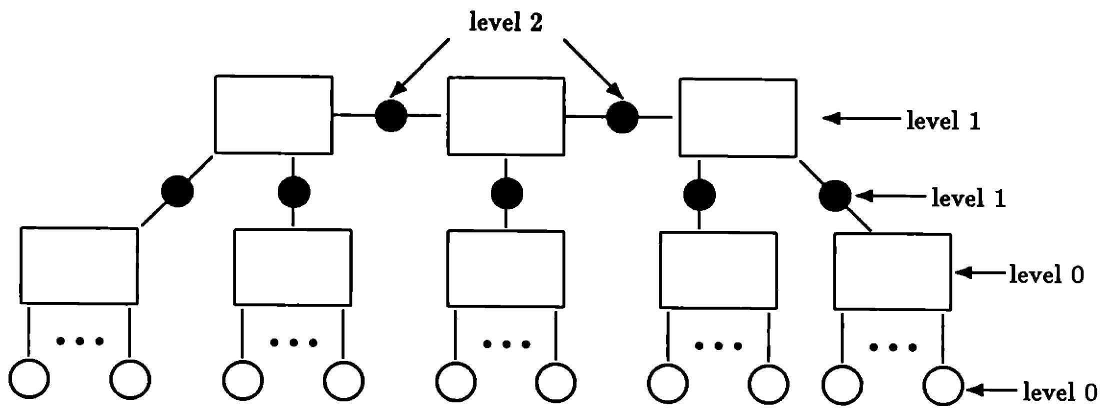


Figure 6.2: The Levels of the Entities of a Network

subnetwork connected to it.

- The order of a network is determined by the highest order subnetwork of the network.

Fig. 6.2 is an example of the application of the above procedure.

The hierarchical networks in Fig. 6.1 are the only type that will be studied. Here, there is a unique path between two entities: the path can be identified with the set of entities that a message passes through when traveling from one entity to another. Further, a level-n node ($n \neq 0$) always connects a level-$(n-1)$ subnetwork and a level-n subnetwork. A level-n subnetwork, except the one at the top level, is "sandwiched" between some level-n nodes and a level-$(n+1)$ node. Among the hierarchical structures that will be excluded from the study here are those similar to the network in Fig. 6.2, where two subnetworks of the same level may be connected via a bridge only. Yet another case is where there may be more than a path between two nodes. For example, a node may belong to two different groups.

The hierarchical networks in Fig. 6.1 are chosen because of their simplicity and the *vague* notion that they are the *pure* hierarchical networks. Obviously, when there are many paths between two entities, the network will look more like a mesh network than a hierarchical network. Note that the single-path feature of the hierarchical networks does *not* imply there is only one communication "route" between two nodes. A subnetwork can be an n-connectivity subnetwork so that a message has n alternative routes when passing through it.

In the following study, attention will be further restricted to the order-1 hierarchy depicted in Fig. 6.1(b). The results can be easily generalized to higher order networks. Since different subnetworks are isolated by bridges, the CAP's at different subnetworks can be tackled independently once the traffic matrices are known. In addition, given the system traffic matrix describing the flows between the N communication nodes, it is easy to derive the corresponding traffic matrices of the subnetworks. To see this, consider the leftmost level-0 subnetwork in Fig. 6.1(b). The bridge directly above it will be labeled b_1. If $(\lambda^{(p,q)})$ is the system traffic matrix, then the traffic matrix of this subnetwork, say $(\lambda'^{(p,q)})$, is obtained as follows:

$$\lambda'^{(p,q)} = \begin{cases} \lambda^{(p,q)} & \text{if } p,q \in \{1,2,\ldots,m\}, \\ \sum_{i=m+1}^{N} \lambda^{(i,q)} & \text{if } p = b_1,\ q \in \{1,2,\ldots,m\}, \\ \sum_{i=m+1}^{N} \lambda^{(p,q)} & \text{if } q = b_1,\ p \in \{1,2,\ldots,m\}. \end{cases} \tag{6.1}$$

The traffic matrices for the other subnetworks at level 0 are obtained similarly, i.e. the inter-group traffic is aggregated at the bridge above a subnetwork.

Let g denote the number of groups in the system, and label the bridges from left to right, $b_1, b_2, \ldots, b_g$. Then the traffic matrix of the level-1 subnetwork, $(\lambda^{*(p,q)})$, is simply given by

$$\lambda^{*(b_i,b_j)} = \begin{cases} \sum_{\substack{p\in G_i \\ q\in G_j}} \lambda^{(p,q)} & \forall \quad i \neq j \\ 0 & \text{otherwise,} \end{cases} \tag{6.2}$$

where G_i is the set of nodes in group i.

Let's define a ***hierarchical grouping*** as the process whereby the N nodes in an order-0 hierarchy are partitioned into groups, and bridges as well as subnetworks are introduced so that an order-1 hierarchy is obtained. Since the bridges can be viewed simply as nodes, once the traffic matrix is obtained, the CAP formulation at the level-1 subnetwork is exactly the same as that of an order-0 hierarchy. Furthermore, treating the level-1 subnetwork as a level-0 one, a hierarchical grouping on it yields an overall order-2 hierarchical network. In this way, different hierarchies are said to be separated by a sequence of hierarchical groupings. In addition, the results of a single hierarchical grouping, e.g. $N_c \uparrow$, $N_t, N_r \downarrow$, remain the same even for a sequence of hierarchical groupings, as long as the a priori conditions are the same throughout. This is the justification for limiting our attention to the order-1 hierarchy.

6.2 The Parameter Values of Hierarchical Networks

What are the effects of hierarchical groupings on the parameters, N_t, N_r and N_c? Obviously there are many ways a hierarchy can be introduced, and the effects will vary accordingly. A more specific question is whether there is a hierarchical grouping method which yields lower parameter values.

It is not practical to consider all the possible hierarchical grouping arrangements because of the large number of them. To determine the number of possible arrangements, consider the problem of partitioning N nodes into g groups, each having at least a node. Let $f_N(g)$ be the number of ways N distinct nodes can be partitioned into less than or equal to g distinct groups. Since there are g ways of assigning a group to each node,

$$f_N(g) = g^N.$$

Let $h_N(g)$ be the number of ways N distinct nodes can be partitioned into exactly g distinct groups, i.e. each group has at least a node. Then,

$$f_N(g) = \sum_{i=0}^{i=g-1} \binom{g}{i} h_N(g-i).$$

Using the fact $h_N(1) = 1$, $h_N(g)$ can be found recursively using the above expression. Since the exact labels assigned to the groups are not important in our problem, the number of ways the N nodes can be partitioned into exactly g groups, each having at least a node is $h_N(g)/g!$. It follows that the total number of ways a hierarchical grouping can be achieved in an N-nodes system is

$$\sum_{g=1}^{N} \frac{h_N(g)}{g!}.$$

For $N = 10$, this number is 115,464 and one would expect it to grow exponentially with N.

To simplify analysis, we consider the uniform traffic CAP in this chapter. Attention will also be restricted to partitions where the total intra-traffic within each group is not greater than 1. Scheme 1 below has the asymptotic behavior that N_t (or N_r) is a negligible fraction of $N(N-1)\sqrt{\gamma}$ when $N(N-1)\gamma \gg 1$ and $\gamma \ll 1$. Since $N(N-1)\sqrt{\gamma}$ is a lower bound on $\frac{1}{2}(N_t + N_r)$ (and $\sqrt{N_t N_r}$) of the original network, this implies there is a hierarchical grouping method which reduces N_t and N_r significantly.

SCHEME 1

Part 1: Partitioning the Nodes

1. Determine m where

$$m^2\gamma \;\leq\; 1,$$

$$(m+1)^2\gamma \;>\; 1. \tag{6.3}$$

If $N \leq m$, no hierarchical grouping will be needed since all traffic can fit into a single channel and $N_t = N_r = N$, which is absolutely the minimum achievable value: every node needs at least a transmitter and a receiver.

2. The nodes are partitioned into g groups where

$$g = \left\lceil \frac{N}{m} \right\rceil \leq \frac{N}{m} + 1. \tag{6.4}$$

$(g-1)$ groups, called the ***regular groups***, will have m nodes each. One group, called the ***special group***, will have $m_s = N - (g-1)m$ nodes.

There are three CAP's to solve; and the corresponding traffic matrices are given in Fig. 6.3. Since their traffic matrices are similar, the regular groups can use the same CAP solution, with the understanding that channels in different copies of the solution are distinct. There are also a CAP for the special group and a CAP for the level-1 subnetwork.

Part 2: Solving the CAP's

1. Solving the level-1 CAP — Each nondiagonal entry is either $m^2\gamma$ (i.e. traffic from a regular group to another) or $mm_s\gamma$ (i.e. traffic from a regular group the special group or vice versa), and neither exceeds 1. Simply assign a channel to each traffic entry. Superscripting the corresponding parameters with *, we have

$$N_c^* = N_r^* = N_t^* = g(g-1). \tag{6.5}$$

2. Solving the level-0 CAP's — Step (a) and (b) below outlines a method for solving the CAP of a regular group. The strategy for the special group is similar after replacing m with m_s.

	1	2	$\cdots$	m	b_1
1	$\times$	γ	γ	γ	σ_1
2	γ	$\times$	γ	γ	σ_1
$\vdots$	γ	γ	$\times$	γ	σ_1
m	γ	γ	γ	$\times$	σ_1
b_1	σ_1	σ_1	σ_1	σ_1	$\times$

$$\sigma_1 = (N - m)\gamma$$

(a) A Regular Group Traffic Matrix

			$\cdots$	N	b_g
	$\times$	γ	γ	γ	σ_2
	γ	$\times$	γ	γ	σ_2
$\vdots$	γ	γ	$\times$	γ	σ_2
N	γ	γ	γ	$\times$	σ_2
b_g	σ_2	σ_2	σ_2	σ_2	$\times$

$$\sigma_2 = (N - m_s)\gamma$$

(b) The Special Group Traffic Matrix

	b_1	b_2	b_3	$\cdots$	b_g
b_1	$\times$	$m^2\gamma$	$m^2\gamma$	$m^2\gamma$	σ_3
b_2	$m^2\gamma$	$\times$	$m^2\gamma$	$m^2\gamma$	σ_3
b_3	$m^2\gamma$	$m^2\gamma$	$\times$	$m^2\gamma$	σ_3
$\vdots$	$m^2\gamma$	$m^2\gamma$	$m^2\gamma$	$\times$	σ_3
b_g	σ_3	σ_3	σ_3	σ_3	$\times$

$$\sigma_3 = m m_s \gamma$$

(c) The Level-1 Subnetwork Traffic Matrix

Figure 6.3: Traffic Matrices of Hierarchical Networks

(a) The total intra-group traffic within a regular group is $m(m-1)\gamma \leq 1$. If $m = 1$, there is no intra-group traffic and no channel needs be assigned. Otherwise, an explicit channel is assigned.

$$N_c^{intra} = 1, \quad N_t^{intra} = N_r^{intra} = m. \tag{6.6}$$

(b) An intergroup traffic entry is $(N-m)\gamma$. This may be more than 1. Simply assign $\lceil (N-m)\gamma \rceil$ explicit channels to each of these entries. Since there are altogether $2m$ entries,

$$N_c^{inter} = N_t^{inter} = N_r^{inter} = 2m\lceil (N-m)\gamma \rceil. \tag{6.7}$$

(c) Since there are $(g-1)$ regular groups, the parameters associated with the regular groups are given by $(g-1)$ times the sum of (6.6) and (6.7).

$$\begin{aligned} N_c^{reg} &\leq (g-1)\{1 + 2m\lceil (N-m)\gamma \rceil\}, \\ N_t^{reg} = N_r^{reg} &\leq (g-1)\{m + 2m\lceil (N-m)\gamma \rceil\}. \end{aligned} \tag{6.8}$$

The inequalities above are satisfied with equality if $m > 1$.

(d) Using the same strategy for the special group,

$$\begin{aligned} N_c^{spe} &= \{1 + 2m_s\lceil (N-m_s)\gamma \rceil\}, \\ N_t^{spe} = N_r^{spe} &= \{m_s + 2m_s\lceil (N-m_s)\gamma \rceil\}; \end{aligned} \tag{6.9}$$

the inequalities are satisfied with equality if $m_s > 1$.

The parameters of the whole system is given by

$$\begin{aligned} N_c &= N_c^* + N_c^{reg} + N_c^{spe} \\ &\leq g(g-1) + (g-1)\{1 + 2m\lceil (N-m)\gamma \rceil\} + \{1 + 2m_s\lceil (N-m_s)\gamma \rceil\}, \end{aligned} \tag{6.10}$$

$$\begin{aligned} N_t = N_r &= N_t^* + N_t^{reg} + N_t^{spe} = N_r^* + N_r^{reg} + N_r^{spe} \\ &\leq g(g-1) + (g-1)\{m + 2m\lceil (N-m)\gamma\rceil\} + \{m_s + 2m_s\lceil (N-m_s)\gamma\rceil\}. \end{aligned} \tag{6.11}$$

The above are satisfied with equality iff $m, m_s > 1$.

□

We now show that when $N(N-1)\gamma \gg 1$ and $\gamma \ll 1$, N_t and N_r of the hierarchical network is a negligible fraction of $N(N-1)\sqrt{\gamma}$. Using (6.4) and

$$\begin{aligned} m_s &\leq m \\ \lceil (N-m)\gamma\rceil, \lceil (N-m_s)\gamma\rceil &\leq N\gamma + 1, \end{aligned} \tag{6.12}$$

Inequality (6.11) can be simplified to

$$\begin{aligned} N_t \text{ or } N_r &\leq \left(\frac{N}{m}+1\right)\frac{N}{m} + \left(\frac{N}{m}+1\right)\{m + 2m(N\gamma+1)\} \\ &= N(N-1)\sqrt{\gamma} \times \frac{(\frac{N}{m}+1)(\frac{N}{m} + 2m(N\gamma+\frac{3}{2}))}{N(N-1)\sqrt{\gamma}} \end{aligned} \tag{6.13}$$

Substituting the two inequalities of (6.3) and rearranging,

$$N_t \text{ or } N_r \leq (N(N-1)\sqrt{\gamma})\left(\frac{\sqrt{\gamma}}{1-\sqrt{\gamma}} + \frac{1}{N}\right)\left(\frac{N\frac{1}{1-\sqrt{\gamma}}}{(N-1)} + \frac{2(N\gamma+\frac{3}{2})}{(N-1)\gamma}\right) \tag{6.14}$$

The above is a product of three terms. The first term is the lower bound of $\frac{1}{2}(N_t+N_r)$ (or $\sqrt{N_tN_r}$) of an ordinary network. If $N \gg 1$ and $\gamma \ll 1$, the second term and the third term can be approximated by $\sqrt{\gamma} + 1/N$ and $3 + 3/(N-1)\gamma$, respectively. The product of these two terms are therefore approximately

$$3\sqrt{\gamma} + \frac{3}{N} + \frac{3}{(N-1)\sqrt{\gamma}} + \frac{3}{N(N-1)\gamma},$$

which tends to 0 as $\gamma \to 0$ and $(N-1)^2\gamma \to \infty$. Hence, we conclude that in the limit that the total traffic is very large and the individual traffic small, N_t is a small fraction of that of an ordinary network.

More generally, v_r and v_t may not be equal. The transmitting and receiving cost is

$$v_t N_t + v_r N_r \leq (v_r + v_t)(N(N-1)\sqrt{\gamma})\sigma(\gamma, N), \tag{6.15}$$

where $\sigma \to 0$ in the limit concerned. For ordinary networks,

$$v_t N_t + v_r N_r \geq 2\sqrt{v_t v_r}(N(N-1)\sqrt{\gamma}). \tag{6.16}$$

The requirement for (6.15) $\leq$ (6.16) is more stringent since $v_r + v_t \geq 2\sqrt{v_r v_t}$, with equality iff $v_r = v_t$. Nevertheless, the asymptotic result, (6.15) $\ll$ (6.16) if $(N-1)^2\gamma \gg 1$ and $\gamma \ll 1$, is valid since $\sigma \ll 1$. The theorem below summarizes the result.

Theorem 6.1 *Consider the uniform traffic situation. There exists a hierarchical grouping method which yields* $v_r N_r + v_t N_t \leq 2\sqrt{v_t v_r} N(N-1)\sqrt{\gamma}[\epsilon(\gamma, N, v_r, v_t)]$. *If* $v_t, v_r \neq 0$, *then* $\epsilon \to 0$ *as* $\gamma \to 0$, $(N-1)^2\gamma \to \infty$

It is easy to see that a similar asymptotic result, $N_c \ll N(N-1)\gamma$ cannot be obtained. In fact, a consequence of the hierarchical grouping is that $N_c \uparrow$.

Theorem 6.2 *Consider* $N_c = N_c^{hn}$ *resulting from solving the CAP's of a hierarchical network using any arbitrary scheme. There exists a solution to the CAP of the corresponding ordinary network with* $N_c \leq N_c^{hn}$.

Comment: The theorem is valid even for non-uniform traffic situation.

Proof: We show how to obtain the solution to the ordinary network based on the hierarchical network's solution. Consider the level-1 subnetwork. Each entry in the traffic matrix is the aggregate traffic of many entries in the ordinary network. The channel assigned to an entry here will be assigned to the corresponding entries in the ordinary network. That is, if $p \in G_i$, $q \in G_j$, the channel assigned to Traffic (p, q) in the ordinary network is the same as that assigned to traffic (b_i, b_j) in the hierarchical network. N_c does not increase (even though N_t and N_r do). This leaves us with only the intra-group traffic at level-0. Unlike in the hierarchical networks, we do not need to worry about the inter-group traffic entries here since it has been taken care of above. For a mental image, this means we do not have to worry about the last row and the last column in Fig. 6.3(a) and (b). Certainly, we can transfer the mappings of the hierarchical network to the ordinary network here. That is, channel assigned to traffic (p, q), p and q in the same group, is the same for both the hierarchical network and the ordinary network. N_c does not increase. Further, any channel assigned only to inter-group traffic at the level-0 subnetworks of the hierarchical network can be deleted. In this case, $N_c \downarrow$.

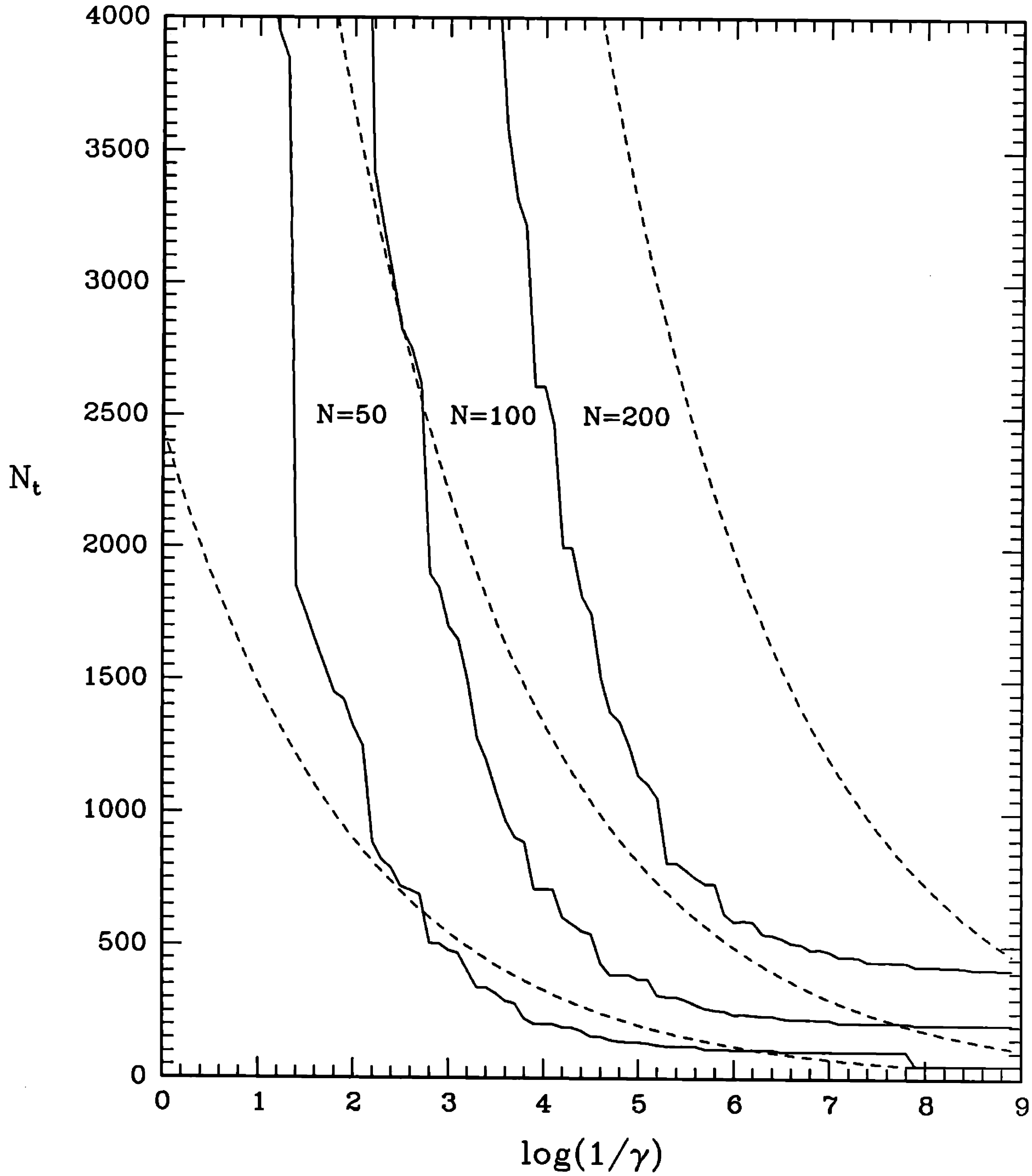
□

Let's illustrate Theorem 6.1 when $v_r = v_t$ with some calculated data. Scheme 2 below is an improvement on Scheme 1, and it has the same asymptotic characteristics. Scheme 1 was used to derive Theorem 6.1 because it is simpler analytically. Again, we assume the uniform traffic case. The graphs in Fig. 6.4 are the results of Scheme 2.

SCHEME 2

Step 2(b), Part 2 of Scheme 1 is modified :

- Consider an entry in the last column and the last row of Fig. 6.3. If $\sigma_1 = (N - m)\gamma \geq 1$, $\lfloor \sigma_1 \rfloor$ channels are assigned explicitly to this entry. The "residual" value of the entry is $\sigma_1' = \sigma_1 - \lfloor \sigma_1 \rfloor$.



The solid-line graphs are N_t of hierarchical networks.
The dash-line graphs are $\max[N(N-1)\gamma^{1/2}, N]$.

Figure 6.4: N_t of Hierarchical Networks.

- Calculate $\lfloor \frac{1}{\sigma_1'} \rfloor$. Partition the entries in the last column into groups of $\lfloor \frac{1}{\sigma_1'} \rfloor$ entries, with the possibility that at most a group may have fewer these entries. Each group is assigned a channel. Do the same thing for the last column.

- The parameter values result from the two steps above are:

$$\begin{aligned} N_c^{inter} &= 2m\lfloor \sigma_1 \rfloor + 2\left\lceil \frac{m}{\lfloor 1/\sigma_1' \rfloor} \right\rceil \\ N_t^{inter} \text{ or } N_r^{inter} &= 2m\lfloor \sigma_1 \rfloor + m\lceil \sigma_1' \rceil + \left\lceil \frac{m}{\lfloor 1/\sigma_1' \rfloor} \right\rceil, \end{aligned} \tag{6.17}$$

□

The values of N_t using Scheme 2 for $N = 50, 100$ and 200 are plotted in Fig. 6.4. The dash lines are the lower bounds on $\frac{1}{2}(N_t + N_r)$ of an ordinary network; and these lower bounds are not always achievable. Notice that for each graph, there is a region over which the hierarchical network yields better results. To the left of this region, γ is not small enough, and to the right, $N(N-1)\gamma$ is not large enough.

As N increases, the region of improvement also increases. It must be stressed that the "actual" regions of improvement for hierarchical networks may be even larger than those obtained here. First, there may be better schemes than Scheme 2. Second, the lower bounds are not always achievable in an ordinary network. In general, however, there are regions where Scheme 2's solution is not as good as that of an ordinary network.

We now present two simple examples where Scheme 2 fails to reduce $N_t + N_r$ as compared to to that of an ordinary network. Here, instead of using the lower bound, we must actually present assignments for the ordinary network which give better results. First consider an extreme case with $\gamma = 1, N = 3$. A level-0 subnetwork has a 2×2 traffic matrix, in which the nondiagonal entries are both 2. Each nondiagonal entry in the level-1 traffic matrix is 1 and the dimensions of the matrix is 3×3.

But this is the same traffic matrix as that of the ordinary network. In addition, we have three 2×2 matrices at level-0 to deal with. Obviously, $N_t + N_r$ is larger for the hierarchical network.

Next, consider a case where $N(N-1)\gamma$ is too small. Let $\gamma = \frac{1}{25}, N = 10$. Solve the CAP of the ordinary network as follows. The traffic matrix consists of four 5×5 submatrices, one for each conner, i.e. the top-left, the top-right, the bottom-left and the bottom-right. The total traffic of each of these submatrix does not exceed 1. A channel is assigned to each of them. This yields $N_t + N_r = 40$. For the hierarchical network, using the notations in Fig. 6.3, $m = 5$, $\sigma_1 = \sigma_2 = \frac{1}{5}$, $g = 2$, $m^2\gamma = \sigma_3 = 1$. From (6.5), $N_t^* + N_r^* = 2$. From (6.6), $N_t^{intra} = N_r^{intra} = 5$. From (6.17), $N_t^{inter} = N_r^{inter} = 6$. Hence, $N_t + N_r = N_t^* + N_r^* + 2(N_t^{intra} + N_r^{intra}) + 2(N_t^{inter} + N_r^{inter}) = 48$.

The contribution of this study is the demonstration of the possibility of decreasing N_t and N_r through a hierarchical grouping. There may be better hierarchical schemes yet to be found.

In the two schemes presented, the entries of the level-1 traffic matrix have values close to 1. Therefore, another application of hierarchical grouping on the level-1 subnetwork is unlikely to reduce N_t or N_r further. Suppose we have a nonuniform traffic situation where nodes can be divided into groups such that traffic between nodes of the same group is dense but not between nodes of different groups. This is likely to be a common practical situation, for example where the traffic between nearby nodes is large but that between far-away nodes is small. Then, it is natural to logically group the nodes accordingly in our hierarchical network. At the level-1 subnetwork, since the traffic between groups is small, the entries of the resulting level-1 traffic matrix may be small enough so that applying hierarchical grouping here will decrease N_t and N_r further.

6.3 Hierarchical Networks — A Qualitative Discussion

In the previous sections, we showed that the introduction of a hierarchical structure may reduce N_t and N_r. Parameter N_c , however, tends to increase (or at least nondecreasing) in general. What does this imply about the cost of a fiber-based network? This section discusses some of the relevant issues qualitatively. Because of the qualitative nature of the discussion, some parts may be vague and there may be unknown exceptions to some of the claims here. Further research, which is beyond the scope of this work, will be necessary in order to substantiate the claims more concretely.

First of all, the total amount of fiber in a network and the magnitude of N_c are two different aspects of the network. By multiplexing all the channels onto a single fiber, other than the additional bridges, the physical topologies of a hierarchical network can be the same as any ordinary network. For both types of networks, the physical topology can be a bus, a ring, a hierarchical structure made up of a connection of several buses, *etc.* For instance, when the logical hierarchical network is physically implemented as a multidrop bus, a bridge is simply another node tapping onto this bus. All the logical subnetworks are embedded in the same bus. Thus, a message from one node to a node in a different group first travels to the bridge belonging to the source node's level-0 subnetwork. This bridge then redirects the message back to the same physical medium, but on a different logical channel of the level-1 subnetwork. The message then travels through another bridge and is redirected back to the channel before reaching the destination node. Indeed, this is a multihop network in which a message hops from node to node in order to reach its destination.

When the product of N_c and the channel capacity exceeds the total capacity afforded by a fiber, more than one fiber will be needed. But this does not necessarily translate to larger total fiber length. It is possible to have a physical hierarchical network where the subnetworks are physically isolated. So, if the channels are frequency channels, the same frequencies can be used on different subnetworks since they are isolated through the use of bridges. This means a physical hierarchical network can afford a larger N_c than an ordinary network. Furthermore, we do not need parallel fibers running along each other.

To complete the above argument, we show the total fiber length of a physical hierarchical network can be made to be close to that of an ordinary network in certain situations. Suppose there are groups of nodes such that the nodes within each group are physically close together, but the distances between groups are large. To provide a mental image to the following discussion, the reader may picture the ordinary network as a ring threading through all the nodes in the system. The discussion applies to many networks, however. The total fiber length consists of two parts: the fibers running between different groups and the fibers within groups. To obtain the desired physical hierarchical network, consider an arbitrary group of nodes. We do *not* change the physical topology connecting the nodes within the group. There are inter-group fibers connecting this topology with the "outside" portion of the network at several points. We "break" the fibers just before these connection points, bring the open ends together to a bridge near these open ends. From this bridge, we make connections back to the broken points. Doing the same thing over all groups of nodes, we then have a physical hierarchical network. The added fiber length is only a small fraction of the original network since the added length for each connection is only a small fraction of the inter-group fiber. This is because the bridge is close to the connection points and the groups are far apart. If the ordinary network we start out with is a ring, the resulting hierarchical network consists of many "small" rings, one for each group. One of the nodes of each ring is actually the bridge of the subnetwork. In addition, there is a "large" ring connecting all these bridges, and this is the level-1 subnetwork.

Next, we consider the complexity issues. Because of the larger N_c, the access control problem of a hierarchical network may be more complex, since it is necessary to keep track of more channels. Each individual subnetwork, however, has an easier access control problem because of the smaller number of channels. Additional complexity is also introduced at the bridges. Intuitively, N_r and N_t are large in an ordinary network because of the absence of switches. In fact, the bridges of a hierarchical network function partly as switches. Consider three nodes i, j and k in three different groups. traffic (i,j) and (i,k) both go to the same bridge belonging to the subnetwork of i, and there is no distinction between these two types of traffic at the subnetwork. The two different traffic types, however, may have been assigned to two different channels at level-1. Therefore, the bridge must distinguish between them and transmits them on the correct channels. Similar situation occurs when the traffic from another group travels to node i. But what is described above is exactly the function of a switch! On hindsight, therefore, it is not difficult to understand why a hierarchical network may have smaller values of N_t and N_r.

More research into quantifying the fundamental control complexity and switching complexity of a network is necessary for more meaningful comparison. For this purpose, we may need to further specify the underlying classes of access control, which we have left out in the network model of this research to avoid losing generality. These issues will not be studied here. The next chapter, however, will relate CAP to some access control schemes so as to illustrate the meaning of "capacity" as used here.

Chapter 7

Access Control and CAP

This chapter investigates the relationship between CAP and Access Control. A node cannot transmit on an accessible channel arbitrarily since this may result in conflicts with other nodes accessing the channel at the same time. Therefore, a control scheme is needed to coordinate or schedule the use of channels. For efficiency, the control scheme is usually dynamic and based on the immediate demand. This is different from CAP because CAP is a static problem and the traffic is the average traffic. In fact, if the traffic varies widely from time to time, instead of the average traffic, the worst-case traffic must be used in CAP. We assume the temporal variations of traffic input is not too big that it can be modeled as a stationary stochastic process.

We ignored the access control aspect of the problem in formulating CAP so as to simplify matters. The "effective" capacity of a channel is lower than the channel transmission rate in order that certain predefined performance requirements are satisfied. This notion will be made more concrete in this chapter. In so doing, we also show that the effective capacity may vary according to the connectivity in a network. Consequently, the expression obtained from solving the n-connectivity problem, where we assumed the effective channel capacity is independent of n,

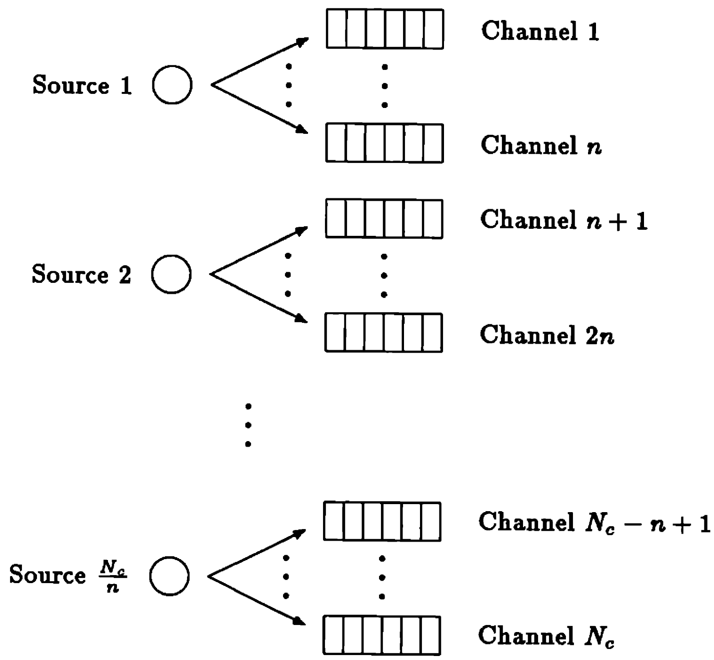


Figure 7.1: Channel Access Structure of an n-connectivity CAP solution

needs to be modified. A perhaps counter-intuitive result is that under certain circumstances, $N_r N_t$ can be reduced by increasing the connectivity.

In this chapter, we consider the simple heuristic scheme (see Chapter 4) for solving the n-connectivity CAP: the 1-connectivity problem with traffic entry $\lambda'^{(p,q)} = \lambda^{(p,q)}/n$ is solved without any further traffic splitting, and n replicas of this solution is taken to be the solution to the n-connectivity CAP. Fig. 7.1 shows the structure of the corresponding channel access model. Each source represents the traffic of many node-pairs, and in this simple CAP solution, the channels accessed by different sources are nonoverlapping. We will consider a few simple access control schemes. More realistic and sophisticated frameworks than those presented is certainly possible. The aim here, however, is to illustrate qualitatively the interaction between Access Control and CAP without complicated analysis.

7.1 Access Control Models

7.1.1 A Time-Multiplexed Circuit-Switched System

Consider a circuit-switched system in which there is only one type of messages, say voice. Suppose each channel in our network is further time-divided into m smaller channels, called minichannels, each capable of carrying a single service. The performance requirement is that the blocking probability [4] must not exceed P_b.

Although $\lambda^{(p,q)}$ has been divided and assigned to n separate channels, this is for solving the CAP only. This does not mean the access control mechanism will divide the incoming services between two nodes into n equal portions and transmit them on n separate channels without first looking at the network state. For example, it might be better to assign a service to the currently least congested channel among the n alternatives. Immediately, we see the potential of increasing the effective capacity with large n.

For the circuit-switched system, a service has at least nm minichannels for transmission. For simplicity, let's assume this is exactly nm for all services. Now, further assume a service is blocked if and only if all the nm minichannels are already occupied.[1] Implicitly, there is a central controller for scheduling and controlling the admission of a service and that it has perfect information about the network state. Also, we ignore the overheads incurred in setting up a circuit, that is, one may assume there is a separate logical or physical control network for carrying the control signals.

Suppose the arrivals of services are independent and Poisson distributed with ar-

[1] One can think of a number of control mechanisms with this feature, e.g. the n accessible channels are given n different priorities and a service searches from the highest priority channel to the lowest priority channel in looking for an unoccupied channel.

rival rate per channel Λ (number of arrivals per second per channel). Let the service duration be exponentially distributed, averaging $1/\mu$ (secs). As far as the blocking probability of a service is concerned, what we have is an $M/M/nm/nm$ queue with arrival rate $n\Lambda$ and average service duration $1/\mu$. The blocking probability is[2] [23]

$$P_{block} = \frac{(n\Lambda/\mu)^{nm}}{(nm)! \sum_{i=0}^{nm} \frac{(n\Lambda/\mu)^i}{i!}}. \tag{7.1}$$

Fig. 7.2 shows the graphs of P_{block} versus the load $\rho = \Lambda/m\mu$, for $m = 50$ and $n = 1$ to 5.

The performance requirement is specified by the blocking probability, P_b. One can then define the effective capacity of a channel to be $\Lambda_{P_b}^{(n)}$, the Λ at which $P_{block} = P_b$ in (7.1).[3] Hence, if $\sigma^{(p,q)}$ is the arrival rate of traffic (p,q), then the normalized traffic is $\lambda^{(p,q)} = \sigma^{(p,q)}/\Lambda_{P_b}^{(n)}$. Note from Fig. 7.2 that $\Lambda_{P_b}^{(n)}$ increases with n. Consequently, the normalized traffic decreases with n. This is not taken into account when solving the n-connectivity CAP in Section 4.5!

Let the load per channel be $\rho_{P_b}^{(n)} = \Lambda_{P_b}/m\mu$. The normalized channel throughput or utility is

$$(1 - P_b)\rho_{P_b}^{(n)} \leq 1 \quad \forall \quad n. \tag{7.2}$$

We see that given some $P_b < 1$, $\rho_{P_b}^{(n)}$ is bounded.

In Section 4.5, assuming the simple scheme for uniform n-connectivity CAP, we showed that for the uniform-traffic n-connectivity CAP,

$$N_r N_t \geq N^2(N-1)^2 n\gamma. \tag{7.3}$$

Solutions close to this bound are achievable in the asymptotic case where $N(N-1)\gamma \gg 1$ and $\gamma \ll 1$. In the following, we assume this situation and that the

[2] Strictly speaking, this formula is insensitive to the exponential service duration and it is valid for the more general $M/G/nm/nm$ queue [19].

[3] An alternative definition is effective capacity $= \Lambda_{P_b}^{(n)}/\mu \times$ tranmission rate of channel (bits per sec). That is, this is the effective capacity expressed in bits per second.

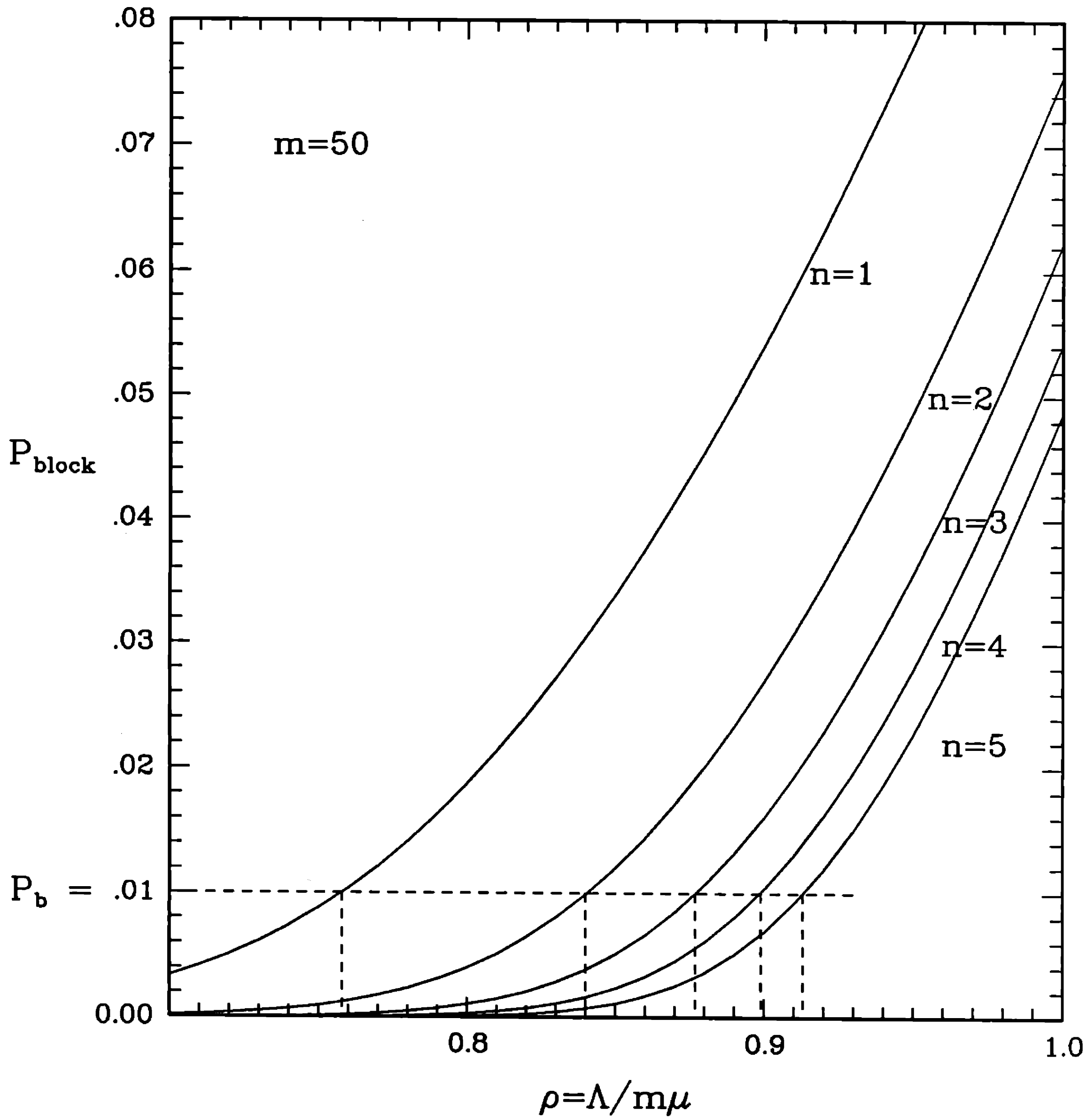


Figure 7.2: P_{block} for the Circuit-Switched System of Section 7.1.1

inequality is more or less satisfied with equality. Now, taking the access control into account, we have

$$N_r N_t \approx N^2 (N-1)^2 n\gamma^{(n)}, \tag{7.4}$$

where

$$\gamma^{(n)} = \frac{\gamma^{(1)} \rho_{P_b}^{(1)}}{\rho_{P_b}^{(n)}}, \tag{7.5}$$

and it decreases as n increases. For illustration, consider Fig. 7.2. For $P_b = 0.01$, since $\rho_{P_b}^{(1)} = 0.758$ and $\rho^{(n)} \leq 100/99$ from (7.2), $\rho_{P_b}^{(n)}/\rho_{P_b}^{(1)}$ is less than 2 for all n. Hence, $n\gamma^{(n)}$ for $n \geq 2$ is larger than $\gamma^{(1)}$, and the qualitative result that $N_t N_r$ is larger for $n > 1$ still holds although it is not as large as in the unmodified model of Section 4.5. We also see from the graph that $\rho_{P_b}^{(n)}$ increases more and more slowly as n increases.

What if we have a low-load operating point? This will be the case if the performance requirement is very stringent (i.e. P_b us small) or if the channel size is small (i.e. m is small). In particular, if $\rho_{P_b}^{(1)} \ll 1$, can it be that $N_t N_r$ actually decreases with n for a range of n? This seems unlikely because $\rho_{P_b}^{(n)}/\rho_{P_b}^{(1)}$ must increase faster than n in that range. Moreover, it would be surprising since this means $N_r N_t$ can be reduced by increasing the connectivity in the network. But this is exactly what happens when $\rho_{P_b}^{(n)}$ is small enough! To show this, we need to derive a situation where $n\gamma^{(n)} < (n-1)\gamma^{(n-1)}$ or equivalently, $\Lambda_{P_b}^{(n)}/n > \Lambda_{P_b}^{(n-1)}/(n-1)$.

Define a new variable

$$\beta^{(n)} = \frac{\Lambda}{n}.$$

Equation (7.1) becomes

$$P_{block} = \frac{\left(n^2\beta^{(n)}/\mu\right)^{nm}}{(nm)! \sum_{i=0}^{nm} \frac{(n^2\beta^{(n)}/\mu)^i}{i!}}. \tag{7.6}$$

For the $(n-1)$-connectivity case,

$$P_{block} = \frac{\left((n-1)^2\beta^{(n-1)}/\mu\right)^{(n-1)m}}{((n-1)m)!\sum_{i=0}^{(n-1)m}\frac{((n-1)^2\beta^{(n-1)}/\mu)^i}{i!}}. \tag{7.7}$$

We want to show for some blocking probability requirement, P_b, it is possible that $\beta^{(n)}$ increases as n increases. One way to do this is to equate the right-sides of (7.6) and (7.7) find a condition leading to $\beta^{(n)} > \beta^{(n-1)}$. Since the resulting expression is complicated, the following alternative approach is used. Instead of fixing P_{block}, we fix $\beta^{(n)} = \beta^{(n-1)} = \beta$ for some β initially. Label the resulting P_{block}'s, $P_{block}^{(n)}$ and $P_{block}^{(n-1)}$, respectively. If we can show that under certain circumstances, $P_{block}^{(n)} < P_{block}^{(n-1)}$, then we are done. Specifically, fix the blocking probability requirement $P_b = P_{block}^{(n-1)}$. $\beta^{(n-1)}$ remains the same, i.e. $\beta^{(n-1)} = \beta$. Since $P_{block}^{(n)} < P_b$ for $\beta^{(n)} = \beta$, in order to make the probabilities equal, we must increase the input rate of the n-connectivity case to $\Lambda_{P_b}^{(n)}$; and this means $\beta^{(n)} = \beta$ *increases* to some greater $\beta^{(n)}$. Hence, if $P_{block}^{(n)} = P_{block}^{(n-1)} = P_b$ then $\beta^{(n)} > \beta^{(n-1)}$.

We now show $P_{block}^{(n)} < P_{block}^{(n-1)}$ for some fixed β in (7.6) and (7.7) if $\beta/m\mu \leq (1-\frac{1}{n})^{2n}/(n-1)$. Since the summation series of the denominator of (7.6) is strictly greater that of (7.7), $P_{block}^{(n)} < P_{block}^{(n-1)}$ if

$$\frac{((n-1)m)!}{(nm)!}\left(\frac{n^2}{\mu}\beta\right)^{nm} \leq \left(\frac{(n-1)^2}{\mu}\beta\right)^{(n-1)m}.$$

This is easily reduced to

$$\left[\frac{((n-1)m)!}{(nm)!}\right]^{\frac{1}{m}}\frac{\beta}{\mu} \leq \frac{\left(1-\frac{1}{n}\right)^{2n}}{(n-1)^2}. \tag{7.8}$$

Now, $[((n-1)m)!/(nm)!]^{1/m} \leq 1/(n-1)m$. Therefore, the above condition is satisfied if the following is

$$\frac{\beta}{m\mu} \leq \frac{(1-\frac{1}{n})^{2n}}{(n-1)}; \quad n \geq 2. \tag{7.9}$$

Alternatively,

$$\rho_{P_b}^{(n-1)} \leq \left(1-\frac{1}{n}\right)^{2n-1}; \quad n \geq 2. \tag{7.10}$$

Note that the larger the n, the more difficult it is to meet this requirement since the right side is small and the left side is larger. In fact, if $\rho_{P_b}^{(n-1)}$ is large enough, as in the example of Fig. 7.2, it is possible that the condition cannot be met at all for all $n \geq 2$. Also, (7.10) is an over-specification of the required condition since there is some simplification in the analysis. So the exact condition is less stringent and has a higher right side. Suppose the exact condition is available. Let this condition be

$$\frac{\beta}{m\mu} \leq C(n, m, \beta). \tag{7.11}$$

In principle we can determine the range of n in which $\beta^{(n)}$ increases with n, somewhat clumsily, as follows. For some blocking probability requirement P_b, calculate $\beta^{(1)}$. Substitute $\beta = \beta^{(1)}$ and $n = 2$ in (7.11) and see if the condition is met. If it is, there will be an improvement by increasing n to 2. Fix $P_{block} = P_b$, $n = 2$ and calculate $\beta^{(2)}$, which is guaranteed to be greater than $\beta^{(1)}$. Next, substitute $\beta = \beta^{(2)}$ and $n = 3$ in (7.11) and see if the condition is met. Keep doing this yields a range of n values for improvement in $N_t N_r$ where $\beta^{(1)} < \beta^{(2)} < \cdots$. This range of n must terminate somewhere. Otherwise, the fact that $\beta^{(n)}$ is strictly increasing for all n implies $\Lambda_{P_b}^{(n)}$ and $\rho_{P_b}^{(n)}$ are unbounded. But the throughput, $\rho_{P_b}^{(n)}(1 - P_b) \leq 1$ for all n.

For a concrete example, the reader can verify that for $m = 1$ and $P_b = 0.01$, $\rho_{P_b}^{(1)} \approx 0.01$ and $\rho_{P_b}^{(2)} \approx 0.08 > 2\rho_{P_b}^{(1)}$.[4] Moreover, substituting $n = 2$ in (7.10), we see that as long as $\rho_{P_b}^{(1)} \leq 0.125$, there will be an improvement in $N_r N_t$ by increasing the network connectivity to 2. Again, the "actual" condition for improvement may be less stringent.

[4]Queueing interpretation: It is not surprising a 2-server queue can support a higher load per serve than (i.e. more than double the input rate of) a 1-server queue for the same blocking probability. It is somewhat curious that the example shows each server of the queue is 8 times as busy as that in the 1-server queue. One may argue intuitively that since the load is small, the servers idle most of the time and there is room for improvement. Nonetheless, the result remains peculiar and it carries over to the work here.

In a fiber-optic network supporting voice services only, it is likely we will not operate in the improvement region due to the possibility of large channels. To get a rough idea, let the transmission rate of the channels be 9.6 Mbps. Since this can accommodate 150 64-kbps voice services simultaneously, $m = 150$. From results of queueing theory, for fixed P_b, $\rho_{P_b}^{(1)}$ increases with m. If P_b is not unreasonably small, $\rho_{P_b}^{(1)}$ is likely to be close to 1. For instance, for $P_b = 0.001$, $\rho_{P_b}^{(1)} \approx 0.8$. But the throughput $(1 - P_b)\rho_{P_b}^{(n)} \leq 1$ implies $\rho_{P_b}^{(n)} \leq 1/(1 - P_b) \approx 1$ for all n. Hence, $\rho_{P_b}^{(n)}/n\rho_{P_b}^{(1)} < 1$ for all n and an improvement in $N_r N_t$ is not possible. On the other hand, if it turns out that it is better to have many smaller channels rather than a few large ones, then the discussion here will be relevant. The merits of small channels will be discussed in the next chapter.

The fact that it is possible to decrease both $N_r N_t$ and N_c by increasing the connectivity is a peculiar result from the engineering viewpoint. For this means all the parameters in our model improve: the bandwidth required is less, the number of channels a node accesses decreases and on top of that, the connectivity increases. Further research in this direction to see if this is a fundamental result for a large class of access control models is certainly worthwhile.

7.1.2 A Packet-Switched System

For a purely packet-switched data network in which the performance is measured in terms of the average delay of a packet, the same qualitative results apply.

As before, for simplicity, we assume there is an implicit central controller and that scheduling delay is negligible. Unlike before, however, the channels will not be further time-divided since statistical multiplexing using one channel generally performs better than statistical multiplexing using m channels with $\frac{1}{m}$ the service rate [6]. Recall that the reason the available "bandwidth" is divided into N_c "large" channels is that the bandwidth window of optical fiber is so large that there are no practical electronic transmitters and receivers which operate at compatible speeds. In a way, we are forced to frequency-divide the bandwidth into channels with maximum rate of 1–10 Gbps [38]. It may be desirable to reduce this rate (increase N_c) further if slower transmitters and receivers are very much cheaper.

For an n-connectivity set-up, using an $M/M/n$ queueing model with arrival rate $n\Lambda$ and average packet duration $1/\mu$, the average packet delay is [6,23]

$$T = \frac{1}{\mu} + \frac{P_Q}{n\mu(1 - \frac{\Lambda}{\mu})} \tag{7.12}$$

where P_Q is the probability that a packet will queue and it is

$$P_Q = \left[1 + \left(1 - \frac{\Lambda}{\mu}\right) \sum_{i=0}^{n-1} \frac{n!}{i! \left(\frac{n\Lambda}{\mu}\right)^{n-i}}\right]^{-1} \tag{7.13}$$

For a fixed Λ, P_Q decreases as n increases; and therefore so does T. If T is fixed instead, the sustainable rate Λ increases with n. Hence, the effective capacity of an n-connectivity network also increases with n.

An analogous result to the circuit-switching case where $N_t N_r$ can be reduced by increasing the connectivity when the load is small will be derived. As before, define $\beta^{(n)} = \Lambda/n$. We want to show that for a fixed T, $\beta^{(n)}$ increases with n. But it is complicated to compare $\beta^{(n)}$ with $\beta^{(n-1)}$ given by a fixed T since the right-side of (7.12) is messy. Instead, we use the old "trick". Specifically, we show that for some β, if we let $\beta^{(n)} = \beta^{(n-1)} = \beta$, the resulting delays, $T^{(n)} < T^{(n-1)}$. Therefore, we can increase $\beta^{(n)}$ further so that $T^{(n)} = T^{(n-1)}$, giving the result that $\beta^{(n)} > \beta^{(n-1)} = \beta$.

For $\beta^{(n)} = \beta^{(n-1)} = \beta$, (7.12) can be written as

$$T^{(n)} = \frac{1}{\mu} + \left[n\mu\left(1 - \frac{n\beta}{\mu}\right)\left[1 + \frac{\left(1 - \frac{n\beta}{\mu}\right) n!}{\left(\frac{n^2\beta}{\mu}\right)^n} \sum_{i=0}^{n-1} \frac{\left(\frac{n^2\beta}{\mu}\right)^i}{i!}\right]\right]^{-1}. \tag{7.14}$$

Let

$$A(n) = n\mu\left(1 - \frac{n\beta}{\mu}\right),$$

$$B(n) = \frac{\left(1 - \frac{n\beta}{\mu}\right) n!}{\left(\frac{n^2\beta}{\mu}\right)^n}.$$

The reader can easily check that $T^{(n)} < T^{(n-1)}$ if $A(n) \geq A(n-1)$, $B(n) \geq B(n-1)$ and $1 - n\beta/\mu > 0$ (note: $n\beta/\mu = \Lambda/\mu$ must be smaller than 1 for stability of the queues anyway). Now, $A(n) \geq A(n-1)$ can be reduced to

$$\frac{\beta}{\mu} \leq \frac{1}{2n-1}. \tag{7.15}$$

Also,

$$\frac{B(n)}{B(n-1)} = \frac{A(n)}{A(n-1)} \frac{(n-1)\left(\frac{(n-1)^2\beta}{\mu}\right)^{n-1}}{\left(\frac{n^2\beta}{\mu}\right)^n}.$$

Suppose $A(n) \geq A(n-1)$. Then $B(n)/B(n-1) \geq 1$ iff $(n-1)((n-1)^2\beta/\mu)^{n-1} \geq (n^2\beta/\mu)^n$, which simplifies to

$$\frac{\beta}{\mu} \leq \frac{\left(1 - \frac{1}{n}\right)^{2n}}{(n-1)}; \quad n \geq 2. \tag{7.16}$$

Alternatively,

$$\rho^{(n-1)} \leq \left(1 - \frac{1}{n}\right)^{2n-1}; \quad n \geq 2. \tag{7.17}$$

Since Condition (7.15) in contained in the above, we can forget about (7.15). To see this, one can show $(2n-1)(1-1/n)^{2n}/(n-1) \leq 1$. Substituting $x = 1 - 1/n$ and using the inequality $\log_e x \leq x - 1$, it is easily seen that $(1 - 1/n)^{2n} \leq e^{-2}$. Also, $(2n-1)/(n-1) \leq 3$. Finally, $(2n-1)(1-1/n)^{2n}/(n-1) \leq 3e^{-2} \leq 1$.

It is interesting that (7.17) is exactly the same condition as that of the circuit-switched model! Whether this is a mere coincidence or the result of a more fundamental phenomenon remains to be studied.

7.1.3 Integrated Systems

In an integrated network supporting a diversity of services, the performance requirements of different service types may well be very different. It is therefore difficult to define the effective capacity. Also, in the CAP model, we do not distinguish between traffic generated by different service types. A question is under what condition the model is appropriate for an integrated network. Answering the question very precisely is beyond the scope of this work. Qualitatively, we expect the CAP model to be a good model under conditions which can sustain a high load, i.e. the effective capacity is close to the actual transmission rate of a channel. In this case, the access control aspect of the problem can be ignored, and we can arbitrarily set the effective capacity to some value slightly lower than the true capacity without sacrificing the performance of the services too much. To illustrate the above claim, we propose a simple way of dealing with the CAP in an integrated environment.

The basic strategy is to assume some a priori effective capacity. Without distinguishing between the different service types making up the traffic, the CAP is solved. We then come back to calculate the resulting performance of each service type assigned to a channel. It may be necessary to solve the CAP again based on a lower effective capacity if the performance requirement of some service type is not satisfied. Iterating this eventually yield a valid solution. A shortcoming of this strategy is that as long as the performance requirement of a single service type is not satisfied, the CAP is not a valid solution. Therefore, a final solution may be one in which capacities devoted to the other service types are poorly utilized. This notion is made more concrete in the following discussion.

A new set of parameters is introduced: for each service type on a channel there is a parameter describing the fraction of channel capacity dedicated to that service type. These parameters must achieve certain values for a CAP solution to be valid.

Consider a channel in a 1-connectivity network which supports two service types, voice and bursty computer data. Let's assume a static capacity partitioning scheme in which a fraction c_v of the channel capacity is devoted exclusively to the voice services. The fraction the data receive is therefore $c_d = 1 - c_v$. There is no sharing of capacity between different services.[5] We have the performance requirements: the voice blocking probability is no more than P_b and the average data delay is no more than T seconds. Basically, for either service type, we have the interaction and the trade-offs between three parameters: the assigned capacity, the load (or the utility of the assigned capacity) and the performance. For the data, for example, a load of ρ_d and a performance requirement of T are satisfiable only if the capacity devoted to data is high enough. This can be seen by considering the $M/M/1$ queueing result for the average delay: $T = (1/\mu)/(1-\rho_d)$ where $1/\mu$ is inversely proportional to the capacity dedicated to the data. In general, in order to satisfy the performance requirements and operate in some predefined load region, c_v and c_d must be greater than some lower bounds, $\underline{c}_v$ and $\underline{c}_d$, respectively.

Consider a uniform traffic example with a transmission rate of 1 Gbps per channel. Assume each voice service requires 64 kbps and the average data packet length is 5000 bits. Tables 7.1(a) and (b) show the values of $\underline{c}_v$ and $\underline{c}_d$ for various load and performance requirements. Table 7.1(a) is generated by fixing $P_{block} = P_b$ and $\Lambda/m\mu = \rho_v$ in (7.1) and finding the m that satisfy the equation; $\underline{c}_v$ is simply $m(6.4 \times 10^4)/10^9$. Table 7.1(b) is obtained using the $M/M/1$ formula : $T = 1/(\mu - \Lambda)$. If L is the average packet length, then ignoring the discreteness of bits, we have $\mu = \underline{c}_d \times 10^9/L$. This gives $T = L/(\underline{c}_d \times 10^9 \times (1-\rho_d))$ where $\rho_d = \Lambda/\mu$.

Suppose the blocking probability requirement of voice is $P_{block} \leq P_b = 0.001$ and let the predefined load region $\rho_v = 0.95$. Then, according to the table, $\underline{c}_v = 0.112$.

[5] There have been substantial efforts devoted to the study of dynamic schemes in which the capacity is shared by different services. [1,2,3,14,15,21,25,26,28,39]

ρ_v	$\underline{c}_v$ ($P_b = 0.01$)	$\underline{c}_v$ ($P_b = 0.001$)
0.80	4.42×10^{-3}	9.66×10^{-3}
0.81	4.80×10^{-3}	1.06×10^{-2}
0.82	5.25×10^{-3}	1.18×10^{-2}
0.83	5.76×10^{-3}	1.31×10^{-2}
0.84	6.34×10^{-3}	1.46×10^{-2}
0.85	7.04×10^{-3}	1.64×10^{-2}
0.86	7.87×10^{-3}	1.87×10^{-2}
0.87	8.83×10^{-3}	2.14×10^{-2}
0.88	9.92×10^{-3}	2.46×10^{-2}
0.89	1.13×10^{-2}	2.88×10^{-2}
0.90	1.31×10^{-2}	3.41×10^{-2}
0.91	1.52×10^{-2}	4.11×10^{-2}
0.92	1.80×10^{-2}	5.04×10^{-2}
0.93	2.16×10^{-2}	6.35×10^{-2}
0.94	2.66×10^{-2}	8.26×10^{-2}
0.95	3.36×10^{-2}	1.12×10^{-1}
0.96	4.41×10^{-2}	1.63×10^{-1}
0.97	6.12×10^{-2}	2.59×10^{-1}
0.98	9.25×10^{-2}	4.91×10^{-1}
0.99	1.62×10^{-1}	> 1

(a)

ρ_d	$\underline{c}_d$ ($T = 0.01$ s)	$\underline{c}_d$ ($T = 0.001$ s)
0.80	2.50×10^{-3}	2.50×10^{-2}
0.81	2.63×10^{-3}	2.63×10^{-2}
0.82	2.78×10^{-3}	2.78×10^{-2}
0.83	2.94×10^{-3}	2.94×10^{-2}
0.84	3.13×10^{-3}	3.13×10^{-2}
0.85	3.33×10^{-3}	3.33×10^{-2}
0.86	3.57×10^{-3}	3.57×10^{-2}
0.87	3.85×10^{-3}	3.85×10^{-2}
0.88	4.17×10^{-3}	4.17×10^{-2}
0.89	4.55×10^{-3}	4.55×10^{-2}
0.90	5.00×10^{-3}	1.00×10^{-2}
0.91	5.56×10^{-3}	5.56×10^{-2}
0.92	6.25×10^{-3}	6.25×10^{-2}
0.93	7.14×10^{-3}	7.14×10^{-2}
0.94	8.33×10^{-3}	8.33×10^{-2}
0.95	1.00×10^{-2}	1.00×10^{-1}
0.96	1.25×10^{-2}	1.27×10^{-1}
0.97	1.67×10^{-2}	1.67×10^{-1}
0.98	2.50×10^{-2}	2.50×10^{-1}
0.99	5.00×10^{-2}	5.00×10^{-1}

(b)

Table 7.1: (a) $\underline{c}_v$ as a Function of Load; (b) $\underline{c}_d$ as a Function of Load.

Similarly, let the average delay be bounded below by $T = 0.001$ and $\rho_d = 0.95$. Then, according to the table, $\underline{c}_d = 0.1$. Note that this is a realizable situation since the sum of $\underline{c}_v$ and $\underline{c}_d$ is less than 1. On the other hand, a voice load of 0.99 with $P_b = 0.001$ is not realizable since $\underline{c}_v > 1$.

A strategy for tackling the integrated network's CAP is to solve the CAP assuming effective capacity $= 0.95$(i.e. the load requirement)$\times$ true capacity, and ignoring the difference between the two different service types initially. After solving the CAP, we come back to check if the fractions of voice and data in each channel exceed $\underline{c}_v$ and $\underline{c}_d$. If so, the solution is valid. Otherwise, another solution is needed, and we may either relax the load requirements or the performance requirements, or simply look for another CAP solution. Now, suppose we have a solution in which $c_d = \underline{c}_d$ and $c_v > \underline{c}_v$. For the voice services, we have overdesigned and $P_b < 0.001$.

In principle, we should be able to support even more voice services than indicated by the solution. However, we are already operating in a high-load situation where $\rho = 0.95$ and there is unlikely that we can "squeeze" in significantly more voice services. Note that this is not the case with low load situations where the solution can be a very poor solution. Thus, intuitively, the high load situations are easy to deal with as far as a good solution is concerned. To have high load situations, we want the channel capacity to be as large as possible; i.e. large channels.

In the extreme that all service types are in the high load regions and a single service (message) of each service type occupies only a tiny fraction of the channel capacity, one would expect some "law-of-large-number" limit and the messages can be treated more or less as constant flows. In this case, CAP can be approached assuming negligible effect from access control. There remain many issues to be answered about integrated networks. But these are beyond the scope of this work.

7.2 Further Comments

7.2.1 Hierarchical Networks and Access Control

In general, N_r and N_t decrease and N_c increases as a result of introducing a hierarchical structure into a network. However, if access control is taken into account, N_r and N_t will decrease less and N_c increases more.

Consider a voice service. It can be blocked by any of the subnetworks it passes through. Thus, to maintain the same blocking probability as that of an ordinary network, the effective channel capacity in the hierarchical network must be reduced. The higher the order of the hierarchy, the lower the effective capacity since the voice passes through more subnetworks. The same argument applies to a data message, where the system delay is the sum of the delays in the subnetworks of the data's path.

Detailed analysis will *not* be undertaken here. But again, intuitively, ignoring the access control's effects when solving the CAP will be a good first approximation under a high load condition, since the effective capacity is close to the actual transmission rate.

7.2.2 More Sophisticated n-Connectivity CAP solutions

The channel access structure of Fig. 7.1 is a direct consequence of the simple CAP heuristic we assumed. If it is somehow possible to increase the effective capacity by going to another CAP solution, then N_r, N_t and N_c can be reduced further.

Fig. 7.3 shows an example of an alternative scheme for a 2-connectivity CAP. Like before, two CAP's are solved, each assuming a traffic matrix of $(\lambda^{(p,q)}/2)$. Unlike before, the second CAP solution is deliberately different from the first. As far as the resulting mappings are concerned, the traffic entries can be divided into 16 sources as shown, each containing a square submatrix of traffic entries. The traffic entries of the same source access the same set of channels. A key difference between this scheme and the simple scheme is that a single channel can be accessed by more than one source in this scheme. For instance, channel (1) is accessed by source 1, 2,5 and 6.

Intuitively, one would expect the system here to perform better. This is because channel sharing, although still partial, is across all the channels. Consider the simple CAP solution obtained by duplicating the first CAP mapping. Sources 1, 2, 5 and 6 are actually a single source in this simple scheme and it accesses two exclusive channels. It is possible that the channels of a group of sources are very

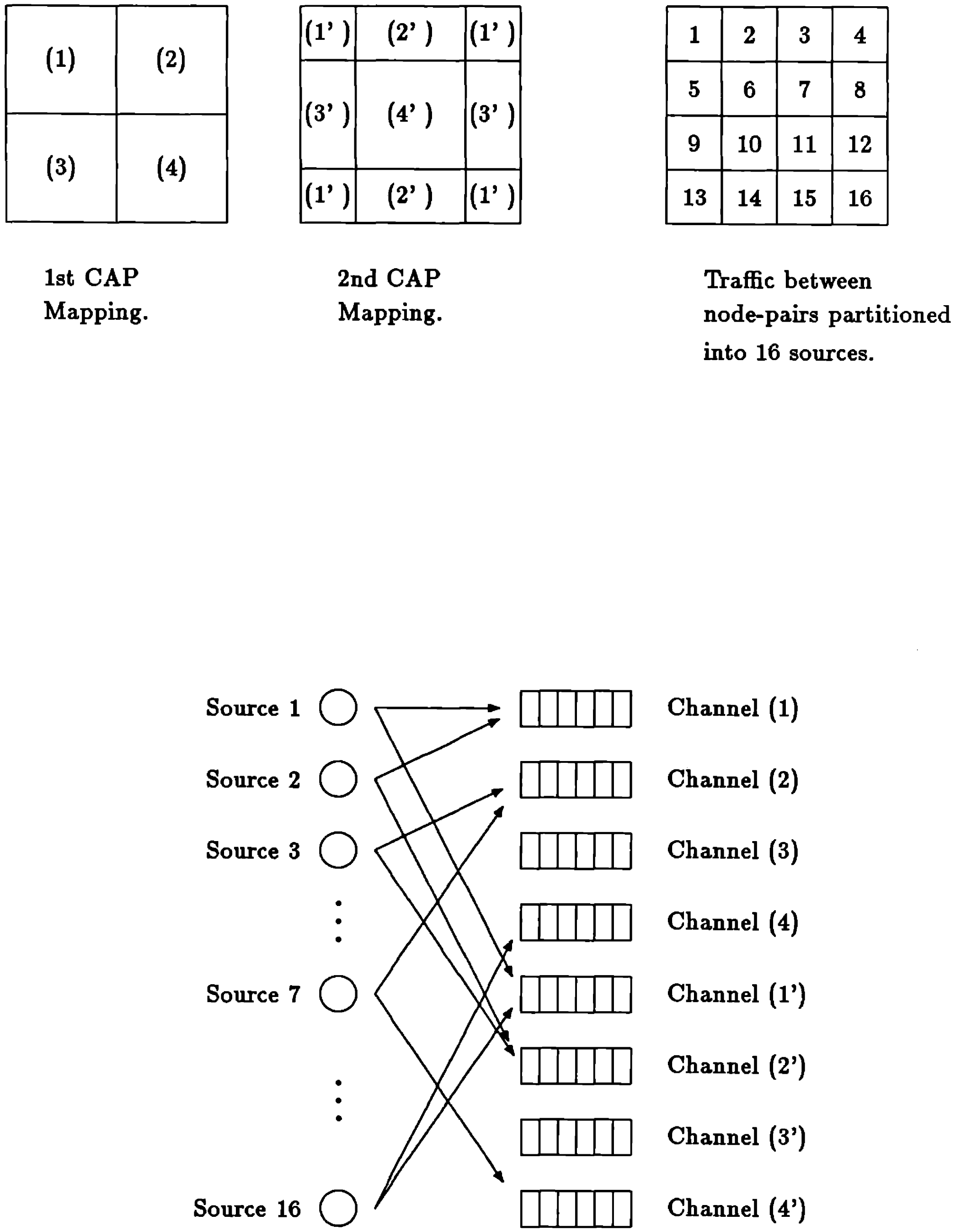


Figure 7.3: An Example of a 2-Connectivity CAP Solution.

congested while the other channels are relatively free. Nothing can be done since channels of different source groups are non-overlapping. On the other hand, for the situation in Fig. 7.3, by introducing a more sophisticated structure, it is possible to distribute congestion more evenly among the available channels. For instance, if channel (1) is congested, source 1 may opt to transmit on channel (1'), source 2 on channel (2'), source 5 on channel (3') and source 6 on channel (4'). The scheduling control issues of structures like this are studied in detail in [40]. Note that given all sources can access the same number of channels, there are many possible channel access configurations. However, not all of these configurations correspond to good CAP solutions in which N_r and N_t are small.

A consequence of using more complicated n-connectivity CAP solutions and access control schemes is that the effective channel capacity is higher, and therefore, N_r, N_t and N_c lower. Again, this improvement is negligible under a high load condition, which can be achieved by having large channels or less stringent performance requirements.

Chapter 8

Conclusion

This research addressed the problem of assigning the capacity of a non-switching multichannel network to its nodes in order to satisfy their traffic requirements. As far as the cost criteria is concerned, three parameters were singled out: the number of channels N_c, the system receivability N_r and the system transmittability N_t. In a sense, to include only the phrase "capacity assignment" in the title of this work is misleading since channel capacity is not considered to be the only precious resource. The assumption that fiber-optic technology is used implies the "peripheral" network costs, such as the cost of transmitters and receivers, are not negligible relative to bandwidth. It is also important to recognize that N_t and N_r do not relate to just the transmitters' and receivers' costs. Specifically, N_t/N and N_r/N are the average numbers of channels a node transmits and receives on. It is likely that these parameters are intimately tied to the control complexity at a node and other network design considerations, e.g. the number of couplers in a network.

This is an extended concluding chapter containing additional discussions on the implications of the previous chapters' results. A summary of this research and its important results is given in Section 8.1. Section 8.2 discusses some implications for fiber-optic networks. Conclusions and issues that need to be explored further are presented in the Section 8.3.

8.1 Summary

In Chapter 2, the capacity assignment problem was formulated as a nonlinear-cost multicommodity network flow problem. Although in principle the problem can be solved, this approach suffers from the following shortcomings:

- The nonlinearity of the problem is such that it is necessary to solve a very large mixed integer linear program.
- This approach gives us an algorithm, but not a closed form solution. Therefore, it is not possible to study the relationship between the parameters easily.
- CAP captures only certain aspects of network design. When other considerations are included, the important target may not be finding the exact solution to the problem, but rather using some rules of thumb to determine the parameter values that are possible.

The question of the computational intractability of CAP was explored in Chapter 3. In the same chapter, albeit not rigorously, we justified the assumption that the traffic $\lambda^{(p,q)} < 1$ for all (p,q) and suggested a "residual" method for dealing with cases with $\lambda^{(p,q)} \geq 1$. This scheme is not optimal, but if some suboptimal heuristic is to be used in the end anyway, there is no point in insisting on exact optimality here. A simple instance of CAP was shown to be NP-hard. Therefore, it is unlikely that a polynomial algorithm can be found for CAP, and this further motivated the heuristic approaches in Chapter 4.

Various heuristics were investigated in Chapter 4. In addition, the trade-offs between N_c, N_r and N_t were studied. Through a lower bound on N_rN_t, it was shown that small values of N_r and N_t cannot be achieved simultaneously. Although this bound assumes uniform traffic between all pairs of nodes, intuitively, there is no reason why the trade-off between N_r and N_t does not apply to the nonuniform traffic case. The trade-off between these two parameters and N_c, however, is not very strong, especially when the traffic entries are small relative to the channel capacity. Any such trade-off can be attributed to the integer constraints of the parameters and the corresponding "packing" problem. Typically, N_c is also small as a side-effect of minimizing N_t and N_r. In Section 4.3, an $O(N^4)$ traffic-matrix mapping heuristic was presented.

Upper bounds on $v_tN_t + v_rN_r$ and N_c were derived in Section 4.4 based on the results of some simple heuristics. Here, v_t and v_r are the cost of a transmitter and the cost of a receiver, respectively. We showed that under conditions with large total traffic but small individual traffic between two nodes, these upper bounds are close to the lower bounds in the sense that their ratios are roughly 1. Thus, the conclusion is that very good solutions can be found using simple heuristics in such a situation. Furthermore, the values of these bounds characterize the fundamental achievable parameter values in the above asymptotic limit. If $v_r = v_t$, a rule of thumb for a uniform traffic case is that $N_t/N = N_r/N \approx \sqrt{N_c}$. In other words, the number of transmitters (or receivers) per node is bounded below by, but roughly equal to, the square root of the system traffic.

n-connectivity networks in which any two nodes have at least n alternative channels for communication was discussed in Section 4.5. A scheme-dependent lower bound on N_tN_r was derived and this bound also has the characteristic that when the total traffic is large and the individual traffic small, simple traffic mapping heuristics yield results approaching it.

In Chapter 5, we departed from the strict definition of CAP and related the combinatorial study of block designs to our problem. Basically, the block design formulation is equivalent to adding more constraints and symmetries to CAP. A result is that if the transmittable channels of every node is the same as its receivable channels, and if the number of common accessible channels is the same for all node-pairs, then $N_c \geq N$, regardless of the actual traffic intensity. Whether this kind of networks is justified in an environment where bandwidth is cheap, such as fiber-optic networks, remains to be studied. The key question is whether the redundant channels (i.e. more than that dictated by the total traffic) imply a greater cost even if the channel bandwidth is cheap. Within the context of CAP here, the minimization of N_t and N_r is accompanied by the minimization of N_c. Therefore, it would be unwise to have N_c much larger than the total system traffic. On the other hand, one may argue that bandwidth, and correspondingly N_t and N_r as well, can be traded for simplicity in control. This will bring in the consideration of access control schemes and whether the symmetry in a block design lends itself to more structural and simple control schemes remains to be explored.

Chapter 6 studied the consequences of introducing hierarchical structures into networks. Based on a logical definition of hierarchical networks, it was shown that hierarchical networks have higher N_c but smaller N_t and N_r. When the total system traffic is large and the individual traffic small, N_t and N_r in a hierarchical network is negligible compared with those of the corresponding ordinary network. Intuitively, this is the result of the embedded switching functionality at the bridges. Thus, a hierarchical structure can be considered as a method of introducing distributed switches to reduce N_t and N_r.

In Chapter 7, the interaction between access control schemes and CAP is investigated. When scheduling overhead is taken into account, N_c decreases as connectivity n increases. This is due to the control flexibility provided by the larger connectivity. Whereas N_t (or N_r) is roughly proportional to $\sqrt{n}$ previously, when access control is considered in the analysis, N_t (or N_r) increases less dramatically with n. In fact, a perhaps counter-intuitive result is that N_t (or N_r) actually decreases with n if the system is operating in a low load situation, a consequence of either stringent performance requirements or small-size channels, or both. Typically, given some fixed performance requirements, a large channel can be better utilized than a small channel. Thus, intuitively, the above situation occurs when the channels are too small to allow a high load operating point. However, for most services' bandwidth requirements, the maximum allowable capacity (1–10 Gbps) of a channel in a fiber system is relatively large. With reasonable performance requirements and large channels, a high load operation is expected. Hence, the qualitative result that N_t (or N_r) increases with n still applies, unless, of course, small channels are desirable for other reasons. In this case, the physical speeds of transmitters and receivers are not a limiting factor on the channel size anymore.

8.2 Implications for Fiber-Optic Networks

8.2.1 The Size of the Channel Capacity

Consider a high-bandwidth fiber-optic environment in which the maximum channel capacity is 1–10 Gbps. Suppose we drop the viewpoint that the channel capacity is given as a fixed quantity. A question is whether the fiber bandwidth should be partitioned into frequency channels of capacity 1–10 Gbps (i.e. maximum rate of electronics) or into many smaller channels.

As with most practical situations, there is no simple general answer to this question if every aspect of network design is taken into account. To narrow down the scope, we assume the transmitters' and receivers' costs are the dominant costs and that N_c is *not* a significant cost factor. Also, to simplify our discussion, we assume $v_r = v_t = v(r)$ where r is the transmission rate of a channel in bits per second. Thus, the average transmitters' cost and receivers' cost per node are $v(r)N_t/N$ and $v(r)N_r/N$ respectively. Our problem is simply to find the minimum $v(r)[N_t/N + N_r/N]$.

We consider the uniform traffic situation with 1-connectivity requirement. Let

$$\begin{aligned} \alpha &= \text{absolute traffic from one node to another (bps) ;} \\ \gamma &= \text{normalized traffic from one node to another (unitless);} \\ \rho(p,r) &= \text{channel utility or load in a channel (unitless),} \end{aligned}$$

where p is some quantity describing the performance requirement: the smaller it is, the more stringent the requirement. Thus, ρ increases with p. As we have seen in the previous chapter, ρ also increases with r thanks to greater capacity sharing among services. By definition,

$$\gamma = \frac{\alpha}{r\rho(p,r)}. \tag{8.1}$$

To have a closed form expression for comparison purposes, suppose the total traffic is very large and the individual traffic γ is small. Then using the "square traffic matrix mapping" heuristic discussed in Chapter 4,

$$\frac{N_t}{N} \approx (N-1)\sqrt{\gamma} = (N-1)\sqrt{\frac{\alpha}{r\rho(p,r)}}. \tag{8.2}$$

Thus, the transmitter cost per node is

$$v(r)\frac{N_t}{N} \approx \frac{v(r)}{\sqrt{r\rho(p,r)}}(N-1)\sqrt{\alpha}. \tag{8.3}$$

Note that $v(r)$ can be used to model various cost components, other than the transmitter cost, that depend on N_t/N. For example, a coupler (or any interface device) may be needed for each accessible channel. N_t/N may also be related to the control complexity at a node. One possibility is $v(r) = a + v_t(r)$, where a represents costs that depend linearly on N_t/N but not on r, and v_t is the actual cost of a transmitter. Let's focus our attention on the case where $v(r)$ can *not* be separated in the above fashion.

In a high load region where $\rho(p, r) \approx 1$, what (8.3) says is that it will be worthwhile to use large capacity channels only if $v(r)$ does not go up as fast as $\sqrt{r}$ as r increases. Thus, if the transmitter cost increases linearly with r, then many smaller channels are better than a fewer very high capacity channels. This is not difficult to understand intuitively. If $v(r) = r$, then $v(r)N_t/N$ corresponds to the total "transmission capability" of a node. But for a large channel, the traffic originating from a single node constitutes only a small part of the total traffic assigned to a channel. But the node must transmit at the channel transmission rate. Therefore, a substantial part of the transmission capability of the node is "wasted". But there is a catch to this reasoning! As r decreases, ρ decreases and we may be in a low load situation. Nonetheless, if r is sufficiently larger than the individual service requirements, this is not a significant problem, and the above argument will still be valid qualitatively.

The conclusion is that we must use channels that are large enough so that channel utility ρ is close to 1 and good CAP solutions can be easily found[1], but not too large that a lot of the transmitting and receiving capabilities are wasted, unless their costs increase less rapidly than the square root of their speed. It must be

[1]Strictly speaking, a good CAP solution is sometimes possible even if γ is large. Suppose we let $r = \alpha/\rho$ so that each traffic entry occupies a channel. Then $v(r)N_t/N = v(r)(N-1) = \alpha(N-1)/\rho$ if $v(r) = r$. This will be a good solution if α is large and represents an aggregate of many services so that ρ is large.

stressed that this conclusion depends strongly on the assumptions we make. For example, in terms of the technical difficulty, it is easier to divide the bandwidth into a few large channels using wavelength multiplexing rather than many smaller channels using heterodyne detection [38]. Furthermore, the channel capacity must be greater than α if the traffic between every node-pair is not to be split across more than one channel.

8.2.2 To Switch or Not to Switch

From this work, we see that even without switching, bandwidth of the communication medium can be utilized quite efficiently in that N_c is approximately equal to the traffic requirements. On the other hand, N_t and N_r grow very fast as the system becomes large (i.e. N increases). Suppose the output rate of a node $(N-1)\gamma$ is fixed , say at λ^*. If $N_t \approx N_r$, then

$$\frac{N_t}{N} \approx (N-1)\sqrt{\gamma} = \frac{\lambda^*}{\sqrt{\gamma}}. \tag{8.4}$$

Since $(N-1)\gamma$ is fixed, as N increases, γ decreases and the above becomes very large. With internal switching (a frequency translator inside the network), potentially, $N_t/N \approx \lambda^*$, which is invariant to N. This is obtained assuming that each node needs only enough transmitters to send out the outgoing traffic and more discussion on its achievability will follow in the next section.

An irony is that although when N is large and γ is small, good suboptimal CAP solutions can be found easily, it is also in this case that N_t and N_r necessarily become very large, as can be seen from the lower bound of N_rN_t. They can be reduced by introducing switching. One method is using hierarchical structures discussed in Chapter 6. Thus, unless transmitters and receivers are cheap, in a large system, switching functionality, whether implemented inside or at the peripheries of networks, is necessary. Here, switching functionality refers to the capability of directing traffic from one channel to another.

Tuning can replace switching, but may require more complicated access control. In this case, a tunable transmitter at a node is shared by many accessible channels. If many services are going on at a node simultaneously, this means its transmitters and receivers must be able to tune from channel to channel very quickly. In other words, a transmitter or a receiver is time-shared by many services very much the same way a mainframe computer is time-shared by many users. Adding to the complication is that a transmitter and its intended receiver must be coordinated so as to tune onto a common channel. Extra overhead is likely to be encountered here as compared to a switching system, and this may result in requiring more transmitters and receivers, even if they are tunable. Whether this is really the case, of course, remains to be explored.

A final word is that the power division problem [29,38] in a passive optical network is a nontrivial problem. N_t and N_r will be important if the number of couplers sharing the power in the network increases with them. So, for a very large network, some switching or tunability is desired as a way to reduce the transmitters' and receivers' costs and other costs related to N_t and N_r, and as a way to solve the power division problem.

8.3 Concluding Remarks and Topics for Further Research

In this thesis, we assumed there is no switching in the network. For the most part, we also assume nontunable transmitters and receivers. With these assumptions, the system transmittability and receivability (N_t and N_r) naturally correspond to the total number of transmitters and receivers. It is shown that, without switching and tuning, these parameters necessarily grow quickly as the network size increases. For a network with tunable hardware, this is not so since a single transmitter or receiver can access many channels. In all cases, the lower bounds on the number of transmitters and receivers, $\sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$ and $\sum_q \lceil \sum_p \lambda^{(p,q)} \rceil$, must still hold since the operating speed of a transmitter or a receiver is limited by the transmission rate of the channel.

These lower bounds cannot be achieved simultaneously in the nontunable case. To what extent these limits can be approached by introducing tunability remains an open question. Technology is a factor to be concerned. As an example, consider the situation where a channel is time-divided into slots, which are then allocated dynamically to services based on demand. To share a single transmitter between several channels, the operating frequency of the transmitter must be tunable on a slot-by-slot basis. This is difficult with the current technology if the time scale is on the order of milliseconds.

For a network with internal switches (frequency translators), it can be shown that the lower bounds on N_r and N_t can be achieved simultaneously in principle, though we may need complex switching control modules in addition to mere frequency translation capability. This will be discussed in the section entitled "Networks with Internal Switching" that follows.

The hierarchical structures in Chapter 6 is a way of introducing switching capability into a network. The degree of switching in a network varies according to the underlying hierarchical structure. Thus, instead of categorizing networks into switching networks and non-switching networks, we could talk of the degree of switching incorporated in a network. In general, a hierarchical network is a "hybrid" network, with some of the nodes communicating directly, and others through switches. The bridges in a hierarchical network can either be active or passive. In the former, the subnetworks are isolated and the bridges are located inside the network. In the latter, the bridges are external and they can be considered as special nodes through which messages hop in order to get to their destinations. Specifically, a passive hierarchical network is a kind of multihop network — a bridge is a special message-passing node and it does not generate traffic; an ordinary node generates traffic but does not switch the traffic of other nodes.

On a conceptual level, whether using a large number of transmitters and receivers, or tunable hardware, or internal switches, or multihop strategy, the goal is to establish connectivity between the communicating nodes. Further investigation is needed so that we can compare more accurately the trade-offs between these different methods. This work points to *a fundamental trade-off between the required transmitting and receiving capabilities, and the complexity of switching in a network.* If the nodes are capable of selecting out their own messages, there is no essential trade-off between the required channel capacity and the switching complexity since it is already close to the total system traffic even without switching. Of course, a general measure of switching complexity is necessary in order to support the above conclusion on a firmer ground.

This research is a first attempt at understanding some of the important issues in a setting where bandwidth is not a primary concern to network design. It lays a foundation upon which more complicated networks can be studied systematically. We conclude this thesis with a discussion on some topics for further research.

Networks with Tuning Capability

The nodes in a network can be either single-user nodes in which there is at most a service going on at a node at one time, or multi-user nodes in which many services are going on simultaneously at a node. The underlying issues for these two kinds of networks are different.

Consider a tuning network with single-user nodes. In this case, each node needs a transmitter and a receiver, assuming their rates exceed the outgoing and incoming traffic. The main problem here is coordinating two communicating nodes so that the transmitting node and the receiving node are tuned onto the same channel. There is no optimization issue as far as the numbers of transmitters and receivers are concerned. This is the setting in Reference [1].

In a tuning network with multi-user nodes, on the other hand, there can be much saving if a transmitter or a receiver is shared by many services. To serve these services simultaneously in a time-shared manner, the hardware must be able to hop from one tuned channel to another very quickly. The problem of coordinating the transmission and the reception becomes very complex if both the transmitters and the receivers are allowed to tune. Thus, it is easier to approach this problem fixing the accessible channels of either the receivers or the transmitters.

Suppose the receivable channels are fixed. In principle, N_r and N_t can be limited to their lower bounds $\sum_q \lceil \sum_p \lambda^{(p,q)} \rceil$ and $\sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$. Let's divide the receiving time of node q is into time frames, each consisting of many time slices. The transmitters are served in a round-robin fashion, and the lengths of services depend on the proportions $\lambda^{(1,q)}, \lambda^{(2,q)}, \ldots, \lambda^{(N,q)}$; that is, the ratio of the numbers of time slices allocated to transmitter nodes i and j is $\lambda^{(i,q)} : \lambda^{(j,q)}$. But in order that $N_t = \sum_p \lceil \sum_q \lambda^{(p,q)} \rceil$, the sequences of times slices of the different receiving nodes must be so arranged that at no time more than $\lceil \sum_q \lambda^{(p,q)} \rceil$ receiving nodes are serving transmitting node p simultaneously. In addition, the transmitters and the receivers must be active most of the time. Specifically, for node p, the ratio of the transmitters' active time to their idling time is $\sum_q \lambda^{(p,q)} : \lceil \sum_q \lambda^{(p,q)} \rceil - \sum_q \lambda^{(p,q)}$; for node q, the ratio of the receivers' active time to their idling time is $\sum_p \lambda^{(p,q)} : \lceil \sum_p \lambda^{(p,q)} \rceil - \sum_p \lambda^{(p,q)}$. This is not a trivial problem! But to the extent that scheduling overhead can be ignored, the problem of which transmitters should be tuned to which receivers at what times is essentially the same problem as the central switching problem that will be discussed next. There the problem is which inputs should be connected, or switched, to which outputs at what times. It can be shown that the lower bounds can indeed be achieved. How this will change if scheduling overhead is taken into account remains to be studied.

There are also issues related to the tuning range of transmitters and receivers. For a system with limited tuning, would the resulting parameter values fall between those of the two extremes, the fully tunable case and the nontunable case? In addition, tuning speed is another important consideration in practice.

Networks with Internal Switching

In principle, once an internal switch is introduced, there is no problem with N_t and N_r achieving their lower limits. Fig. 8.1 depicts the situation with an active frequency translator in the network.

Obviously, the frequency translator needs to do more than mere frequency translation. At different times, the messages on a channel must be switched to different channels and no two channels must be switched to the same channel simultaneously. Reference [12] addresses the scenario in which each channel is divided into time frames consisting of many time-slotted channels. Although the motivating physical situations are different, the resulting problem frameworks are similar. In particular, the paper's main theorem, when applied to our situation, implies that there is a switching scheme in which N_t and N_r achieve their lower bounds. There

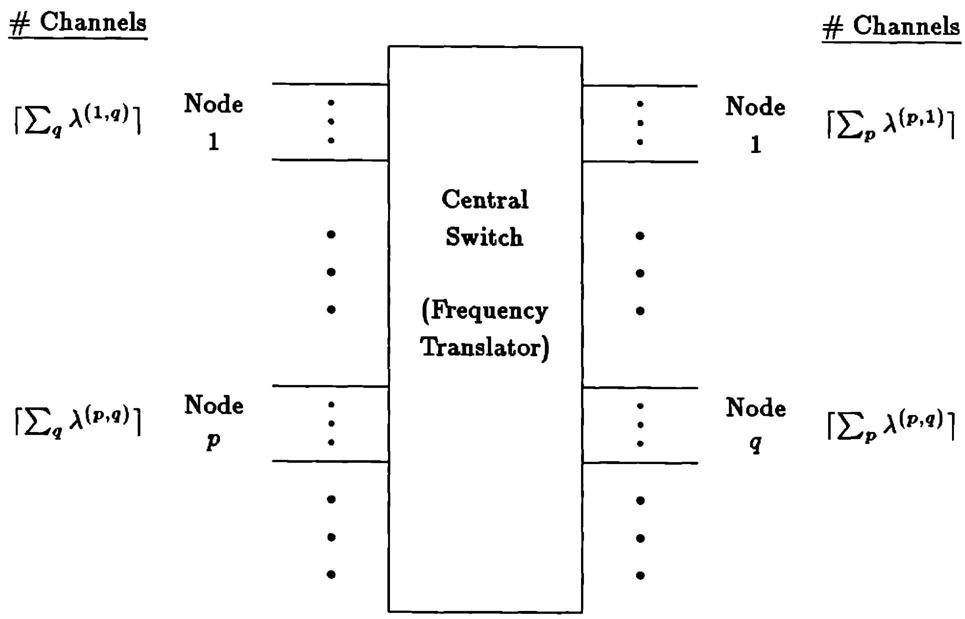


Figure 8.1: A Central Switch Configuration.

seems to be no conceptual difficulty in extending the result to the continuous time scenario since we can simply let the duration of the time slots go to zero and the number of time slots in a frame go to infinity.

An algorithm that can find the "optimal" switching scheme in a short time remains an open research topic. In practice, this is particularly difficult if there are many time slots in a frame. Another question is whether there is a satisfactory suboptimal but fast algorithm.

Multihop Networks

Switching functionality can be implemented centrally or distributedly. As mentioned earlier, the hierarchical networks in Chapter 6 can be regarded as distributed-switching multihop networks if the bridges are passive and external to the networks. An alternative scheme is to have the traffic-generating nodes share the switching responsibility. Thus, a node will receive its own messages as well as switch messages between other nodes. The different ways the switching functionality is distributed and assigned to the nodes, and their respective merits, are open to further research.

In general, for the same reasons discussed in Subsection 7.2.1, a multihop network will experience some degradation in performance compared with the ordinary network of this work. To compensate for this degradation, more channel capacity and devices of higher speed will be needed. On the other hand, switching in a multihop network may be simpler than in a central-switching network. This, of course, remains to be substantiated.

Finally, a measure of switching complexity is essential if different switching networks are to be studied and compared more systematically. Whether a fundamental measure for switching complexity is possible is worth exploring.

Bibliography

[1] G. L. Abbas, *Frequency Allocation for Fiber Optic Integrated Services Communication Networks*, M.I.T. Electrical Engineering Ph. D. Thesis, Feb 1988.

[2] J. M. Aein, "A Multi-User-Class, Blocked-Calls-Cleared, Demand Access Model," *IEEE Trans. Commun.*, Vol. COM-26, # 3, pp. 378-385, Mar 1978.

[3] E. Arthurs and J. S. Kaufman, "Sizing a Message Storage subject to Blocking Criteria," *Performance of Computer Systems*, Armsterdam, The Netherlands: North-Holland, pp. 547–564, 1979.

[4] J. Bellamy, *Digital Telephony*, Wiley (1982).

[5] D. P. Bertsekas, *Notes on Nonlinear Programming and Discrete-Time Optimal Control*, M.I.T. Lids Report LIDS-R-919, Jan 1983.

[6] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc. (1987).

[7] S. P. Bradley, A. C. Hax and T. L. Magnanti, *Applied Mathematical Programming*, Addison-Wesley, Reading Mass. (1977).

[8] F. R. K. Chung, J. A. Salehi and V. K. Wei, "Optical Orthogonal Code: Designs, Analysis, and Applications," submitted to *IEEE Trans. of Info. Theory*, 1986.

[9] R. J. Dakin, "A Tree-Search Algorithm for Mixed Integer Programming Problem," *Comp. J.*, Vol. 8, # 3, pp. 250–55, 1965.

[10] G. B. Dantzig and P. Wolfe, "Decomposition Principle for Linear Programming," *OR*, Vol. 8, # 1, pp. 101–111, Jan–Feb 1960.

[11] A. Dey, *Theory of Block Designs*, John Wiley & Sons (1986).

[12] K. Y. Eng and A. S. Acampora, "Fundamental Conditions Governing TDM Switching Assignments in Terrestrial and Satellite Networks," *IEEE Trans. of Communications*, Vol. COM-35, pp. 755–761, July 1987.

[13] J. Escobar, *Maximum Likehood Detection for Optical Code Division Multiple Access System Models*, M.I.T. Electrical Engineering Ph. D. Thesis, Feb 1988.

[14] M. J. Fischer and T.C. Harris, "A Model for Evaluating the Performance of an Integrated Circuit-and-Packet-Switched Multiplex Structure," *IEEE Trans. Commun.*, Vol. COM-24, pp. 195-202, 1976.

[15] G. Y. Fletcher, *et. al.*, "A Queueing Network Model of a Circuit Switching Access Scheme in an Integrated Services Environment," *IEEE Trans. Commun.*, Vol. COM-34, # 1, pp. 25–29, Jan 1986.

[16] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnoy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," *Annals of Discrete Mathematics 5*, pp. 287–326, 1979.

[17] M. Hall, *Combinatorial Theory*, Blaisdell Publishing Company (1967).

[18] J. H. Hayes, *Modeling and Analysis of Communications Networks*, Plenum, New York (1984).

[19] D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research, Vol. I*, Mc-Graw-Hill (1982)

[20] J. Y. Hui, "Pattern Code Modulation and Optical Decoding – A Novel Code-Division Multiplexing Technique for Multifiber Networks," *IEEE Journal on Selected Areas in Comm.*, Vol. SAC-3 NO. 6, Nov 1985.

[21] J. K. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Trans. Commun.*, Vol. COM-29, pp. 1474–1481, Oct 1981.

[22] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, Mc Graw-Hill, New York (1964), out of print; reprinted Dover, New York (1972).

[23] L. Kleinrock, *Queueing Systems Vol. I: Theory*, Wiley (1967)

[24] R. W. Klessig, "Overview of Metropolitan Area Networks," *IEEE Communications Magazine*, Vol. 24, pp. 9–15, 1986.

[25] B. Kraimeche amd M. Schwartz, "Analysis of Traffic Access Control Strategies in Integrated Service Networks," *IEEE Trans. Commun.*, Vol. COM-33, # 10, pp. 1085–1093, Oct 1985.

[26] R. H. Kwong and A. L, Garcia, "Performance Analysis of an Integrated Hybrid-Switched Multiplex Structure," *Performance Evaluation*, Vol. 4, pp. 81–91, 1984.

[27] E. L. Lawler and D. E. Wood, "Branch-and-Bound Methods: A Survey," *OR*, Vol. 14, # 4, pp. 699–719, Jul–Aug 1966.

[28] S. -Q. Li and J. W. Mark, "Performance of Voice/Data Integration on a TDM system," *IEEE Trans. Commun.*, Vol. COM-33, # 12, pp. 1265–1273, Dec 1985.

[29] S. C. Liew, *Topologies and Power Division Problem of Fiber Optic Networks*, M.I.T. Electrical Engineering Master's Thesis, Feb 1986.

[30] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, NJ (1982).

[31] E. G. Rawson, R. M. Metcalfe, R. E. Norton, A. B. Nafaratte and D. Cronshaw, "Fibernet: A Fiber Optic Computer Network Experiment," *Proc. Fourth European Conf. on Optical Communication*, Genova, Italy, Sep 1978.

[32] H. J. Ryser, *Combinatorial Mathematics*, MAA Publication (1973).

[33] R. V. Schmidt *et. al.*, "Fibernet II: A Fiber Optic Ethernet," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, #5, pp. 702–710, Nov 1983.

[34] A. P. Street and W. D. Wallis, *Combinatorial Theory: An Introduction*, Published by the Charles Babbage Research Center, Canada (1977).

[35] Daniel T. W. Sze, "A Metropolitan Area Network," *IEEE Journal on Selected Areas in Communications*, pp. 815–824, Nov 1985.

[36] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, Eaglewood Cliffs, NJ (1981).

[37] J. A. Tomlin, "Minimum-Cost Multicommodity Network Flow," *OR*, Vol. 14, # 1, pp. 45–51, Feb 1966.

[38] S. S. Wagner, *Multiplexing Methods for Fiber Optic Local Communication Networks*, M.I.T. Electrical Engineering Ph. D. Thesis, June 1985.

[39] G. F. Williams and A. L. Garcia, "Performance Analysis of Integrated Voice and Data Hybrid-Switched Links," *IEEE Trans. Commun.*, Vol. COM-32, pp. 695–706, 1984.

[40] A. Wong, *Channel Scheduling for Optical Communication Networks with Frequency Concurrency*, M.I.T. Electrical Engineering Ph. D. Thesis, Feb 1988.

[41] C. Yeh and M. Gerla, "High Speed Fiber Optic Local Networks," *Proc. of the Tenth Anniv. Meeting of the NSF Grantee-User Group in Optical Communications Systems*, pp. 69–80, June 1982.

[42] N. Zadeh, "On Building Minimum Cost Communication Networks," *Network*, 3: pp. 315–331, Wiley (1973).

[43] N. Zadeh, "On Building Minimum Cost Communication Networks over time," *Network*, 4: pp. 19–34 (1974).