Autonomous Routing Algorithms for Networks with Wide-Spread Failures: A Case for Differential Backlog Routing

by

Wajahat Faheem Khan

B.S. in Electrical Science and Engineering
B.S in Mathematics with Computer Science
Massachusetts Institute of Technology, 2007

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2008

Signature of Author: _____
Department of Electrical Engineering and Computer Science
August 7, 2008

Certified by: _____
Eytan H. Modiano
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: _____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Autonomous Routing Algorithms for Networks with Wide-Spread Failures: A Case for Differential Backlog Routing

by

Wajahat Faheem Khan

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2008, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

We study the performance of a differential backlog routing algorithm in a network with random failures. Differential Backlog routing is a novel routing algorithm where packets are routed through multiple paths to their destinations based on queue backlog information. It is known that Differential Backlog routing maximizes the throughput capacity of the network; however little is known about practical implementations of Differential Backlog routing; and its delay performance. We compare Differential Backlog to Shortest Path and show that Differential Backlog routing outperforms Shortest Path when the network is heavily loaded and when the failure rate is high. Moerover, a hybrid routing algorithm that combines principles of both Shortest Path and Differential Backlog routing is presented, and is shown to outperform both. Finally, we demonstrate further improvements in delay performance of the aforementioned hybrid routing algorithm through the use of Digital Fountains.

Thesis Supervisor: Eytan H. Modiano
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

Many factors contribute to a finished piece of work. It would be unfair to take sole responsiblity for this thesis even beyond the bibliograpy.

First and foremost, I want to thank my advisor Professor Eytan Modiano. His constructive feedback guided the thesis to its present form. His ideas and vision have provided the foundation and roadmap for the research that is the subject of this thesis. I learned a great deal from his methodoloy to analyze results and solve problems through the use of simple and elegant insights and principles. Even outside the scope of research, he has helped me to keep my priorities right in the bigger picture of academic and personal development.

Computer simulations play a signficant role in this thesis and required significant computer resources. I would like to thank my advisor for investing in a server suitable for simulations. Many of the results presented herein have just been possible because of faster simulation execution speeds thanks to the new server. Kayi Lee, a research colleague, also contributed by running his simulations at a lower priority to enable me to get some important results towards the end.

I am thankful for getting the opporunity to work in a wonderful research group. I got up-to-speed on many trades and tricks of the game under the mentorship of Gil Zussman who is now a Professor at Columbia University, NY. He made research enjoyable and shared his valuable philosophical perspectives on life. I am also delighted to have worked alongside Guner Celik, Murtaza Zafer, Jun Sun, Andrew Brzezinski, Krishna Jagannathan, Ryan Kingsbury, Anand Srinivas and Seb Neumayer.

Masoud Akbarzadeh, a graduate student in Department of Architecture, has been a great friend and roommate. Asad Kalantarian and Adnaan Jiwaji shared the same class year, major, and dormitory as me through four years of undergraduate life. I was lucky to have them around for Masters as well since they made classes and homework fun. Adnaan helped me a great deal in finishing this thesis, including help with some diagrams.

I cannot thank my loving mother enough for the inspiration and my hardworking

father for the energy which keeps me going. And I pray that the God Almighty keeps it the same way for times to come.

Finally, this work was supported by the Defense Threat Reduction Agency (DTRA) under grant number HDTRA1-07-1-0004.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Communication Systems

The field of Communications has been around since the evolution of mankind. The earliest examples include the use of smoke signals, mirrors and messengers. These primitive modes of communication were very limited in terms of message content as well as number of users. The need for greater versatility and higher volume of communications prompted the emergence of a service industry. The cycle of demand and innovation fueling each other, led to better quality and larger scale of communication services such as telegraph, telephone, wireless telegraph, radio, television, satellites, cellular phones and Internet.

Towards the end of twentieth century, a concerted effort to promote collaboration between numerous heterogeneous communication networks was initiated. The approach taken to solve the problem can be best summarized in the words of one my engineering professors, "When there is a problem, just lump it". The result was the standardization of a layered architecture called *The Basic Reference Model for Open Systems Interconnection* [13] for communication systems. The seven layers of OSI model are shown in Fig. 1-1. Rather than being a strict reference, the OSI model serves more as a guideline and an educational tool for design and analysis.

OSI model can be best explained through example. The Physical layer is responsible for providing the medium for transmission of messages; horses, for example in

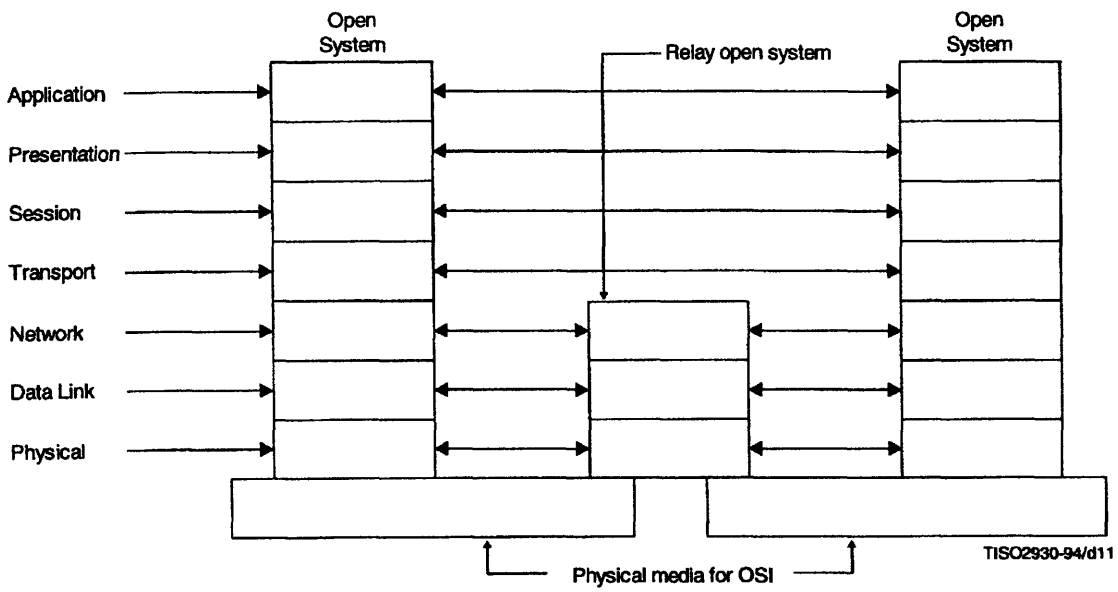

Figure 1-1: OSI layered reference model [13]

the Pony Express service (1860-1861). The setting of Pony Express network will be continued henceforth, to illustrate the other layers of OSI model. Data Link Control layer controls the transmission of messages across a single physical link. The stations which provided new horses for riders every 10-15 miles can be thought of as providing the functionality of this layer. The network layer provides routing information for messages between points which are not linked directly. For example, the maps which Pony Express riders used to navigate around, symbolize this layer. The next higher layer, Transport, controls dispatch and receipt of messages from sources to destinations. The closest resemblance to the functionality of this layer in the Pony Express network would be the dispatcher who decides when enough mail has arrived that a new horse be sent from one coast. Session layer is responsible for initiating and concluding individual correspondences; for example, it would be equivalent of a clerk who issues receipts for mailings. Presentation layer specifies format for a mailing such as a scroll, encrypted message etc. The highest layer, Application, governs essentially different types of available services that a communication network supports, e.g. letters, post cards, money orders etc. Table 1.1 summarizes examples of these layers in the Pony Express service and lists their loose equivalents in the modern day Internet.

OSI Layer	Pony Express	Internet
Physical	horses	wires, radio
DLC	checkposts	Aloha, 802.11
Network	maps	Internet Protocol(IP)
Transport	dispatcher	Transport Control Protocol(TCP), User Datagram Protocol(UDP)
Session	clerk	...
Presentation	scroll, encrypted message, cash	HTML, JPEG
Application	letter, money	Internet Explorer, Windows Media Player

Table 1.1: Examples of OSI model layers in Pony Express and the Internet

1.2 Optical Communications

Modern day communication systems can be classified into three main categories based on physical layer and therefore other higher layers, because of the trickling effect of physical layer on the design of higher layers. These are namely satellite, wireless and optical fibers. Electrical wires are sometimes also used in land-line networks but only to provide access at the end-user level. Optical fibers are by far the most widely used communication networks in terms of proportion of total network traffic. The popularity of optical fibers stems from their high capacity (bandwidth), high speed (latency) and low costs. Transcontinental fiber-optic submarine cables support practically all of the world-wide digital traffic including Internet, corporate communication lines and digital telephony. Fig. 1-2 shows a world-wide map showing the distribution of

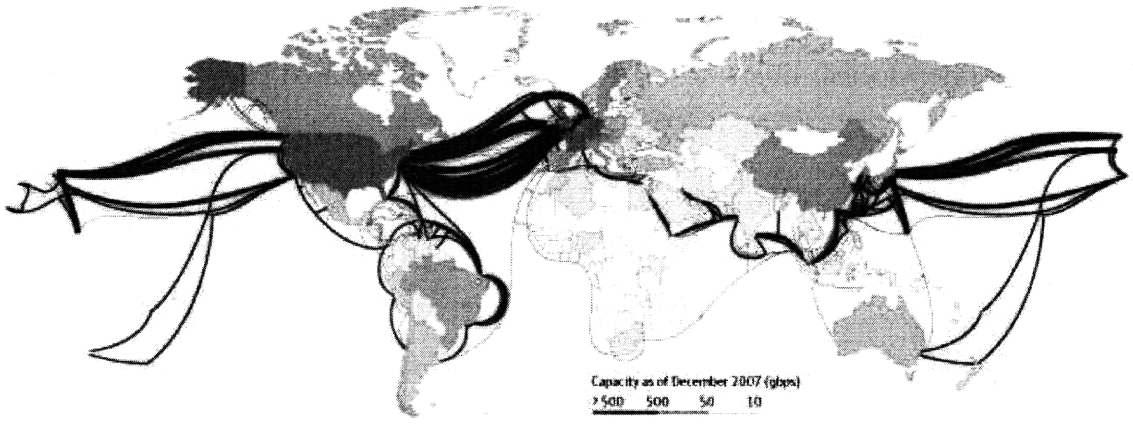

Figure 1-2: Map of transcontinental fiber-optic submarine cables [1]

transcontinental fiber-optic cables. Thus optical communications play an important role in keeping the world connected.

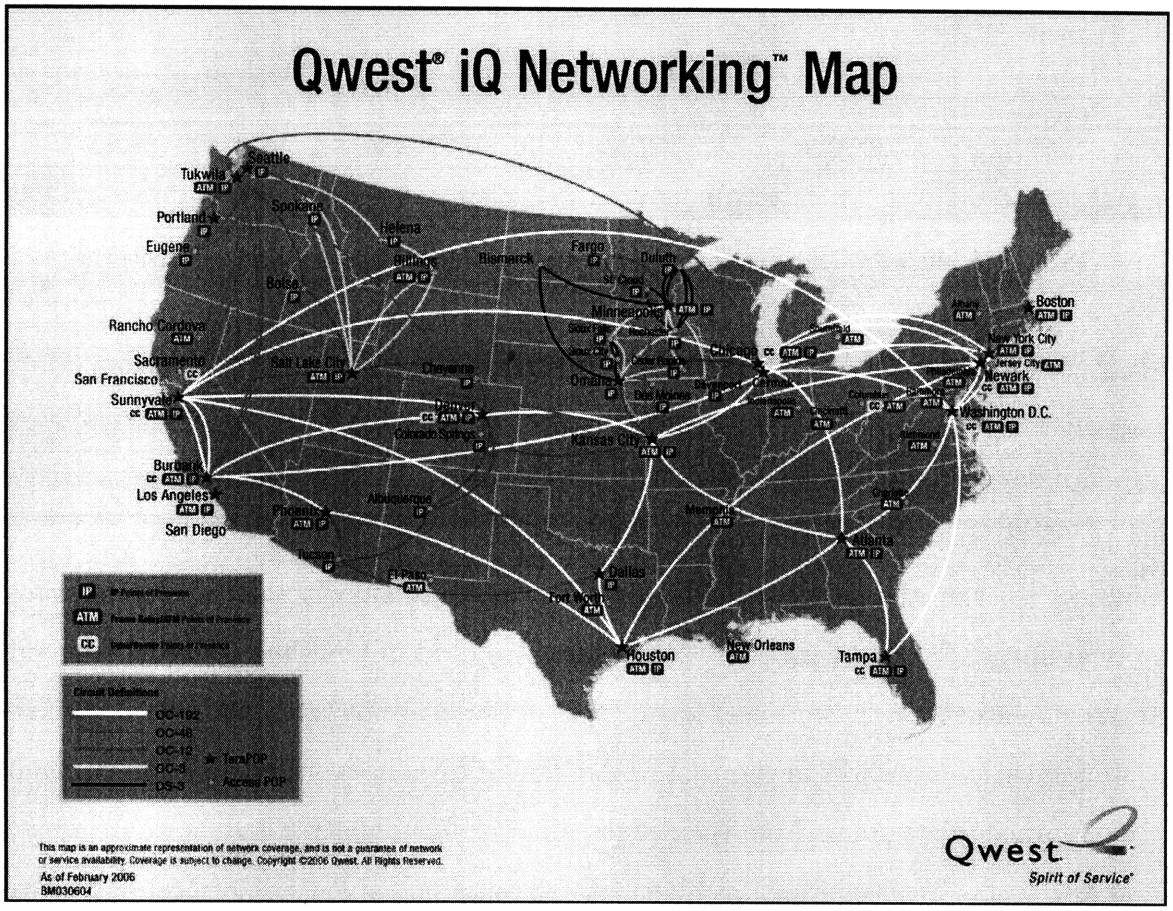

Figure 1-3: Qwest fiber-optic back-bone [8]

Several optical fiber backbone networks support almost all the digital traffic in the continental United States. Fig. 1-3 and Fig. 1-4 show two commercial backbone networks. Optical fibers also support almost all of the domestic digital traffic. Both military and commercial users subscribe heavily to optical networks. Apart from the Internet, the commercial applications include landline and cellular/mobile phones. Although mobile phones communicate with a cellular base station, the base stations use optical networks to connect to other base stations which in turn connect to the receiver's cellphone. Many economic services also rely heavily on optical communications. These include dedicated lines for banks, ATMs, stock exchanges, etc. Nowadays, cable companies dominate the industry for digital High-Definition

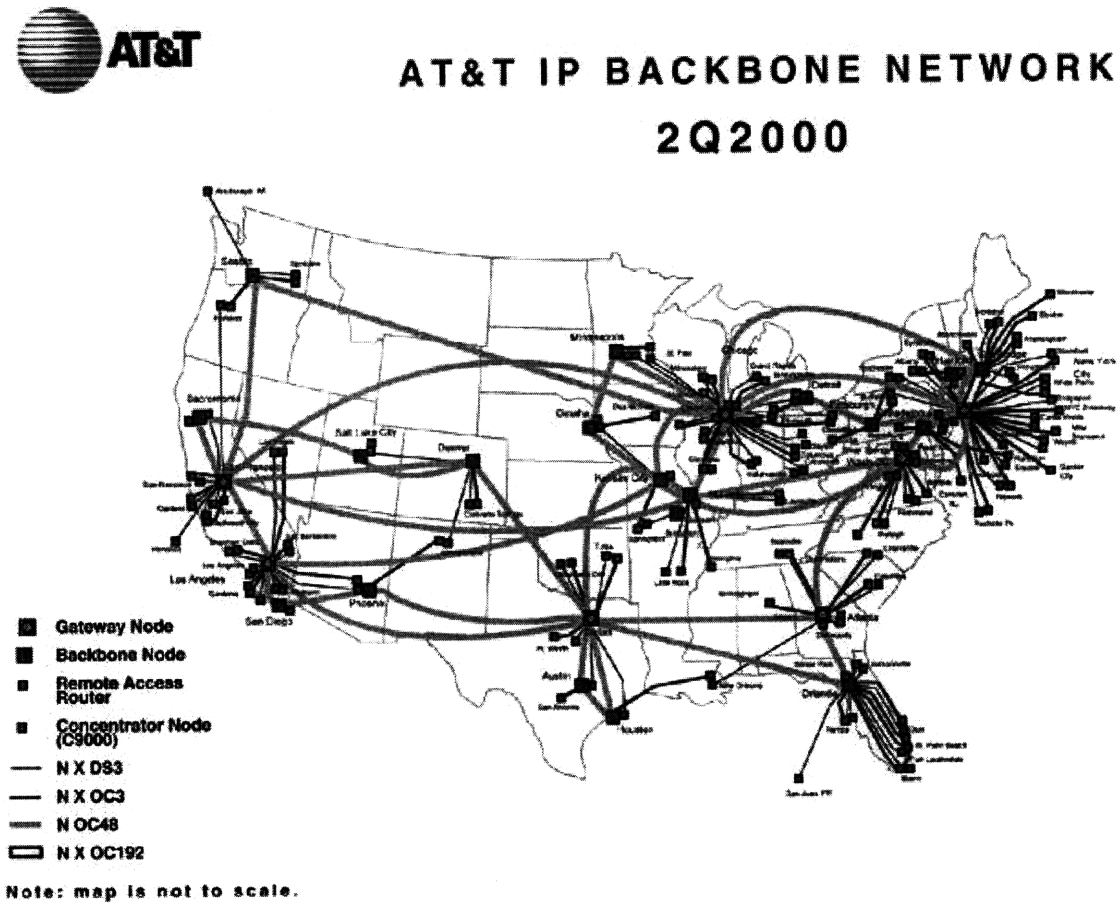

Figure 1-4: AT&T fiber-optic back-bone [8]

(HD) television channels and play-on-demand multi-media content delivery. Cable companies usually bundle television services with telephone and Internet. The traffic demand of average Internet user is increasing due to the popularity of video sharing websites, Internet radios, TV streams and other bandwidth-intensive applications. Furthermore, a cost-effective ubiquitous 4G service can only be possible if some of the celluar traffic is diverted to land-line networks. Electric cables (1 Gigabit/sec.Cable) are inferior to optical fibers (100 Terabit/sec.Fiber:one wavelength) in terms of theoretical maximum achievable capacity. As a result, many experts believe that optical fibers supporting TV, land-line and cell phones, in addition to Internet traffic, is the only way forward to cope with the immense pressure on digital traffic bandwidth and latency.

In addition to the consumer services, emergency response systems can also benefit

a great deal from a robust optical network. Because of their low-costs, it is easier to provision extra capacity using optical fibers to survive failures and meet unusual surges in traffic demand. Such optical networks can withstand catastrophes and help various government departments co-ordinate rescue and relief operations, sparing costlier satellite and wireless networks for better uses. In emergency situations where communication networks are stressed both in terms of failures and higher loads, robust optical networks can mitigate mass communication cut-offs typical of September 11 and North-East blackout in August, 2003.

1.3 (Un)Robustness of Optical Networks

From a military perspective, optical communications have an edge over other means of communications such as satellite and wireless, in addition to their high bandwidth, low latency and economy. Satellite and wireless networks use airwaves as a medium for carrying signals. Hence they are very susceptible to eavesdropping by third parties and encryption techniques are used to keep sensitive information private. Optical fibers need to be tapped physically for a third-party to be able to intercept communications. Moreover, there exist mechanisms to detect any physical tampering of optical fibers by examining the electromagnetic profile of received signal [19]. Satellite and Wireless networks are also prone to electronic jamming whereas optical networks are resilient to such measures. Optical fiber links are also less affected by Electromagnetic Pulse (EMP) attacks in comparison with other communication networks [23].

Optical networks, however, are at an extreme disadvantage in terms of their vulnerability to physical damage- be it intentional, un-intentional or natural. A significant amount of wireless infrastructure is mobile and can be reconfigured and/or camouflaged to make it less susceptible to an attack. Similarly, targeting satellites requires high sophistication. Optical networks are however static in nature and at most times fibers are laid in obvious locations such as along rail-road tracks and major highways. To make matters worse, optical fibers take the longest to recover from a failure. Wireless units tend to be field replaceable and the notion of spare

satellites parked in orbits [6] can help mitigate a satellite failure. Optical fibers in a Wide Area Network (WAN) such as a national backbone typically span hundreds if not thousands of miles and thus a disruption takes days if not weeks to resolve. It took almost two weeks to repair the under-sea fiber cuts in Mediterranean which severely affected 75 million Internet users in Middle East and India [14] in early 2008. Thus, it is of prime importance to identify ways in which optical networks can be made robust to physical failures.

1.3.1 Single-Link Failures

Un-intentional fiber cuts are not rare in optical networks. Insufficient care while digging for maintenance or laying of utility networks such as electricity, roads, natural gas etc. causes fiber links to be cut. Resultantly, a lot of research [15], [26], [27], [35], [39], [40], [10] and [31] has been done to make networks in general and optical networks in particular recover in the event of a single physical link failure. Nowadays, commercial network companies provide services which are guaranteed against single link failures, labelled as 1:1 or 1+1. In 1+1 protection, the primary traffic and its copy is simultaneously sent over two disjoint paths and the receiver can tune to the second path in case of a failure. For 1:1 protection, the network provider keeps a spare path which is used if the primary path fails. Nowadays, each optical cable or *trench* carries many (typically tens of) optical fibers because it does not make sense to justify digging hundreds of miles for just one optical fiber. Each optical fiber, in turn, transmits signals at many (usually 100-200) different wavelengths, a technique known as Wavelength Division Multiplexing (WDM). In this context, recovering from single link failures becomes significantly harder because a physical link failure might cause multiple transmissions to be disrupted.

1.3.2 Multiple Failures

Intentional or natural failures can cause yet greater damage to optical networks than isolated single link or node failures. An example of intentional failure is a Weapons of

Mass Destruction (WMD) attack which can cause wide-spread node and link failures. Natural causes of failures include hurricanes such as Katrina in summer of 2005, earthquakes such as those that occur frequently in California and floods. Fig. 1-5 depicts NSFNET which was the primary Wide Area Network (WAN) backbone in the United States until mid-90s. It shows example of a mass-scale failure in Western U.S. resulting in several link failures. In the absence of recovery paths, these link failures would have serious consequences for communications between the East and West Coasts. One example is the original path between Pittsburgh, PA and Palo Alto, CA, which has been disrupted. However, as one might observe, there are sufficient unaffected links in the network to reconnect the two stations by re-routing the traffic between them. A single re-routed path might not have sufficient capacity to support all of the re-directed traffic so having multiple recovery paths to share the load of the primary failed path would prove more promising in dealing with large-scale failures. Examples of a couple of re-routed paths are also shown in Fig. 1-5. This example illustrates just one of the desirables of an optical network that is robust against large-scale failures. In general, recovering from multiple failures in context of WDM is expected to be much more complex than recovering from single failures.

1.4 Design of Optical Networks robust against large-scale failures

We can break down the problem of robustness against multiple failures into four primary subparts.

1. **Robust network architecture design**

   First, a robust network must have enough spare capacity to handle failures. If a node is connected to the network with just one link and that link fails, it would not be possible to restore communications with the affected node. Hence it is essential that a network remains physically connected after failures occur. [15] and [27] are concerned with the design of robust network architectures in the

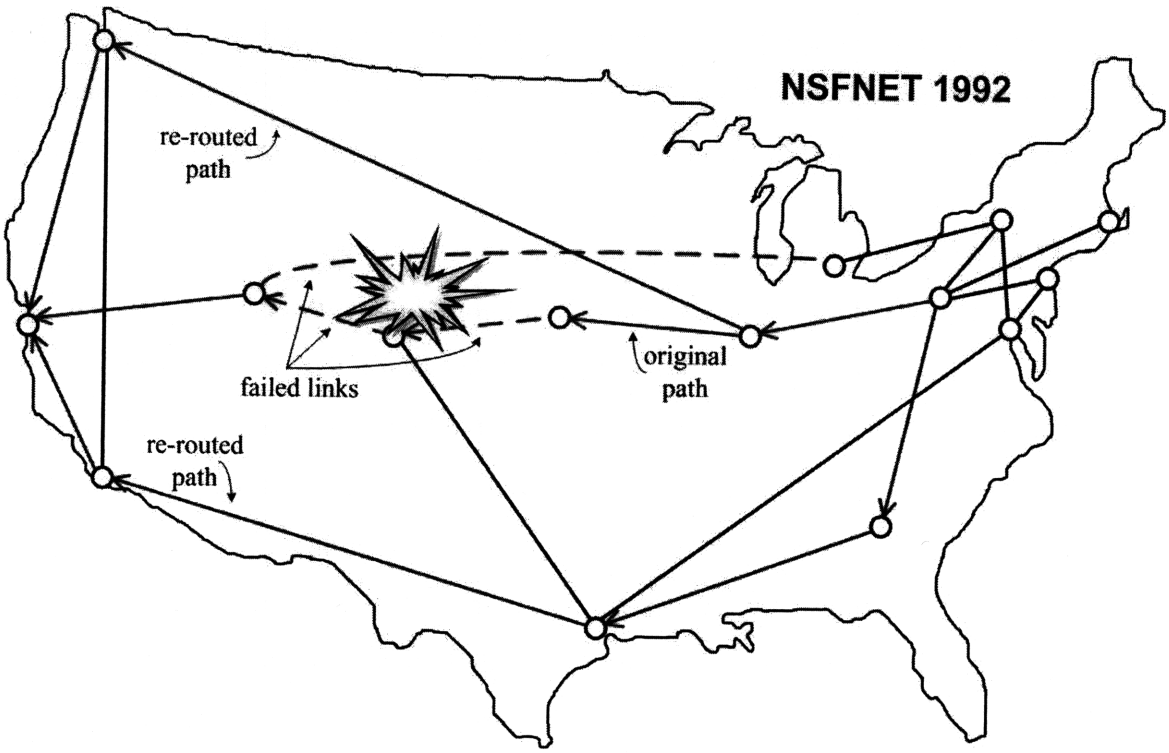

Figure 1-5: Example of a network failure and re-routing along alternative paths

event of a single link failure. The prevailing definition of *survivability* as the property of a network to recover from a single physical link failure needs to be generalized to *k-survivability* wherein a network can recover from k link failures. [40] analyzes connectivity of different graphs in terms of the probability of individual link failures but assumes that nodes are invulnerable. In general, a better framework for analyzing network robustness against multiple failures would be to characterize robustness in terms of the proportion of network component failures below which a network remains connected.

2. **Topology inference**

Designing a robust network architecture is an issue that needs to be dealt with before a network is operational. However, the first step in dealing with failure(s) in an operational network is to obtain an accurate picture of the extent of damage. For the Internet, this is made possible through employing Open Shortest Path First (OSPF) algorithm which is also responsible for updating routes. In [17], Kleinberg et al. treated the problem of detecting (ϵ, k)-failures, where upto

k network elements (either links or nodes) fail causing disconnectivity of two sets, each atleast ϵ by fraction of total number of nodes, by placement of D detectors. [5] presents a practical algorithm to infer the topology of a network.

3. **(Re-)Routing**

   Once the up-to-date state of network is known, affected communications need to be re-routed over alternative paths in the network. For small networks, these backup paths can be pre-computed against all possible failures. However, for networks on the scale of a nation, possible failure scenarios become too many to be tractable. For example, the number of possible failures for a graph with just thirty links is more than a billion. [26] studies the problem of routing optical signals in a way which is tolerant of any single physical link failure. The paper also proves that finding these desired routes in a network topology is an NP-complete problem. Dealing with large-scale failures using the same approach can reasonably be expected to be yet harder. The second sub-problem of topology inference can be avoided or discounted a great deal depending upon the information needs of the routing algorithm. For example, if a routing algorithm requires knowledge of the complete topology of a network, recovery can take too long to start.

4. **Quality of Service(QoS) support**

   It is impossible to design an invincible network regardless of the amount of spare capacity built into physical network architecture. From a commercial point of view, it is not always cost-effective to pay for spare capacity. Therefore, a network which can support Quality of Service (QoS) for different types of traffic in case of a large-scale failure is highly desirable. For example, government or military communications in a national crisis must take priority over commercial traffic. [10] talks about providing multiple degrees of reliability for different services in a network in a cost-effective manner. In [31], Ou and Mukherjee have approached the QoS problem from the perspective of differentiated quality of restoration times.

We have broken down the problem of robustness against large-scale failures, which formed the initial motivation for this research effort, into sizeable sub-parts. Herein, we will treat one of the important subparts which will have consequences on the design of other subparts as well.

1.5 Area of Focus

In this thesis, we will focus on the third subpart: the autonomous routing problem. The second and fourth sub-problems depend largely on routing because they are determined by the information needs and ability of the routing algorithm respectively.

An effective routing problem can be likened to an experienced and alert traffic sergeant. Like a traffic sergeant, it should try to do the best job possible given road (lane closures) and traffic conditions (congestion) etc. including accommodating vehicles of varying priority levels (ambulances).

More specifically, we are concerned with scheduling the use of available resources, which are link capacities in a network, to transmit packets from a source node to a specified destination node. Our main interest lies in scenarios of wide-spread, large-scale and arbitrary failures in a network. Since a general approach is desired to accommodate arbitrary network topologies, failure rates and load patterns, we seek an *algorithmic* solution. The sought algorithm needs to be self-fulfilling for its information needs and implementation; hence it must be *autonomous*. Ideally, the *autonomous algorithm* must also be able to provide fully differentiated quality of service (QoS), as specified earlier. The merits of an effective routing algorithm are:

1. restoration time in event of a failure
2. delay
3. throughput/achievable capacity
4. fairness according to a QoS specification
5. practicality

1.6 Thesis Outline

Section 2 provides academic context to the problem of autonomous routing by discussing previous work, introduces Differential Backlog routing and describes our approach for its analysis. In section 3 we present experimental results on the performance of Differential Backlog routing and its comparison with the existing paradigm of routing, namely Shortest Path routing. In section 4, we suggest and examine variants of Differential Backlog routing which exhibit better performance. In section 5, we provide discussion of considerations for a realistic version of Differential Backlog routing. Section 6 talks about topics for future research and we finish with concluding remarks in Section 7.

Chapter 2

Previous Work

2.1 Routing Algorithms

Almost all realistic networks are *incomplete* graphs. Hence, typically messages need to travel through intermediate nodes to get to their destinations. A routing algorithm is responsible for directing messages from their origins all the way to their respective destinations. There are two sides to the problem of routing in general: first, how are the paths calculated; and second, how are they enforced. Obviously, the first distinction is broader and we use it to distinguish between different routing mechanisms.

2.1.1 Shortest Path Algorithms

Many different routing algorithms exist in practice. Shortest Path algorithms are by far the most popular class of algorithms employed in modern networks. As it might be implied from their name, Shortest Path algorithms consider all the possible paths between a source and destination and choose the one with minimum cost. The costs are typically a function of length, congestion, monetary costs etc. of a link. Bellman-Ford and Dijkstra [9] are two widely used shortest path algorithms. The best-known running times of these algorithms are $\theta(|V| \log V + E)$ and $\theta(|V||E|)$ respectively, where V represents the number of nodes and E represents the numer of edges in a

network. As it can be seen, these algorithms grow quickly with network size.

Apart from their running time, Shortest Path algorithms also have many other issues:

- **Distributed Implementation**

  The decentralized nature of most networks requires distributed implementation of these algorithms giving rise to a set of issues [2]. Propagation of network state information and synchronization among nodes are major issues which become worse in the event of a change of topology before a pass of algorithm is finished. Border Gateway Protocol (BGP) [32], [11], and Open Shortest Path First (OSPF) [12] are well-documented and well-studied routing *protocols* employed in modern networks such as the Internet.

- **(In)Efficiency**

  Shortest path algorithms come under the class of greedy algorithms and thus do not always achieve optimality. While each *session* is routed over its shortest path, network resources might not be utilized to their full potential as a whole. A link might become a bottle-neck while other links might be left unused at the same time. This means that either some sessions are blocked or that they get lower-than-achievable rates. For example consider the example presented in Fig. 2-1.

  In Fig. 2-1, 2-1(a) and 2-1(b) depict what happens when two sessions try to take place simultaneously in a network employing shortest path routing. In Fig. 2-1(a), the second session gets blocked while going over its second hop because all of the hop's capacity is dedicated for the first session. Another possibility is to share the limited capacity between the two sessions which would allow them to take place simultaneously but at half the rate that can be supported by a hop, as illustrated in Fig. 2-1(b). Ideally, the second transmission can also achieve full capacity if it is routed through a longer but unutilized path as shown in Fig. 2-1(c). This is an example of *optimal routing* for the given topology and traffic conditions.

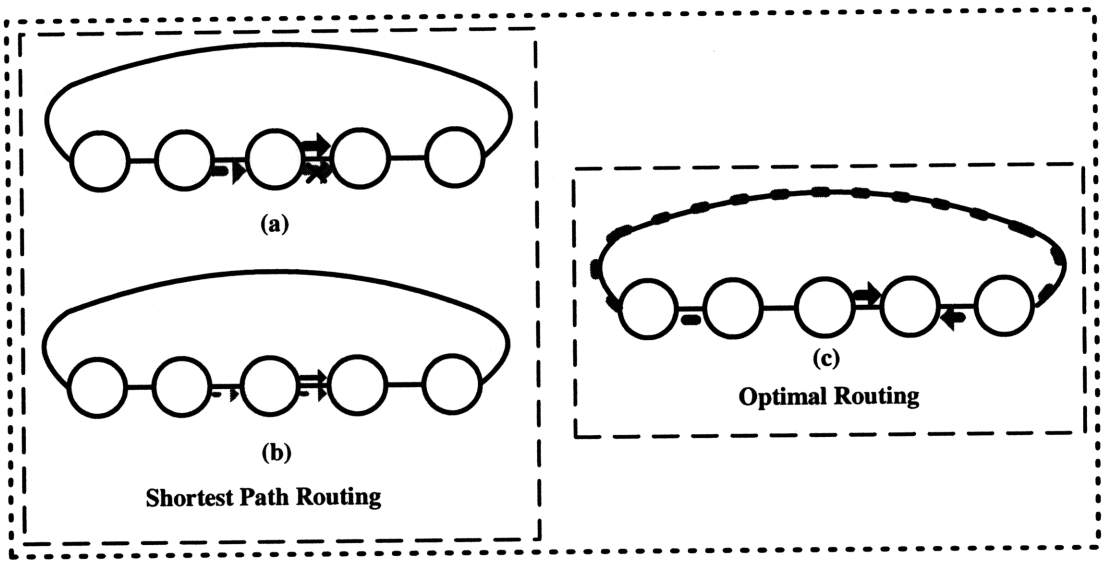

Figure 2-1: Sub-optimality of Shortest Path routing in terms of throughput

- **Network Congestion**

  Traffic demand in most networks is random and inconsistent. This means that various parts of a network are stressed at different times. Routing algorithms can be *static* or *adaptive*. A *static* routing algorithm only updates computed paths in response to a link or node failure. An *adaptive* algorithm also takes congestion on different parts of the network into account. Most major networks employ some sort of adaptive routing based on shortest path to respond well to a variety of traffic conditions. The congestion information cannot be fully taken into account by all parts of the network because real-time updates of traffic situation will not only give rise to more congestion but probably will be useless by the time they reach all nodes in a network. In experience, a compromise is made between the transmission requirements of congestion information and its usefulness.

- **Stability**

  Realistic networks tend to be dynamic in terms of traffic conditions and network resource availability. The time taken to respond to a network change is an important criterion for evaluating the performance of a routing protocol. A routing protocol which takes congestion information into account might divert

traffic from a busy link to an idle link. This will cause the idle link to become busy and the routing protocol might divert traffic back to the original link and so the cycle starts all over again. This leads to an *oscillatory* behavior and far worse kinds of oscillations are possible [2].

Although stability issues are made prominent by the need for distributed implementation, they are specially exacerbated in *incremental* routing protocols. In an *incremental* protocol, only changes in network topology and routing policies are propagated through the network. The relative locale in time and place of these updates might make them outdated or redundant. Routers which act on outdated updates also send more updates to their neighbors triggering more outdated updates and giving rise to increased network congestion and rapid fluctuations in routing policies. For example, Border Gateway Protocol (BGP), the routing protocol used to route packets among Autonomous Systems (ASs) in Internet, has convergence issues and has been a subject of due monitoring and academic investigation [21], [22].

- **Survivability**

  Because of the reasons mentioned in Section 1.3, it is desirable for a network to be tolerant of failures. As mentioned in Section 1.5, the routing problem in context of large-scale failures is the main focus of this thesis. Distributed implementations of shortest path algorithms take significant time to re-route traffic for large networks using respond-on-the-fly approach. [20] shows that it could take upto 15 mins in some cases for BGP to converge in event of a single failure while packet loss and delay can increase manifold during recovery. One can reasonably expect the failure response to be worse yet for large-scale failures and [33], [34] explore some techniques for its improvement.

  The response time to failures can be significantly improved by pre-planning against all the failure scenarios. For instance, [4] presents algorithms for finding shortest pairs of disjoint paths- both link and vertex- to accommodate single-link failures. However, preplanning against all possible failures becomes increasingly

impractical with increasing size of networks and possibility of large scale failures.

Inspite of all the above-mentioned issues, Shortest Path algorithms are very popular because of their excellent delay performance. Since packets are routed directly to their destinations, they tend to take the shortest time. Shortest path can therefore meet stringent Quality of Service(QoS) specifications of delay for time-sensitive services such as voice, video and tele-surgery. It is possible to achieve better delays than those of Shortest Path algorithms, however. The trick lies in realizing that a session typically consists of one or more file transfers whereas a file contains a number of packets. The file delay may be improved by using multiple paths to send packets from the same file. Figure 2-2 illustrates this phenomenon.

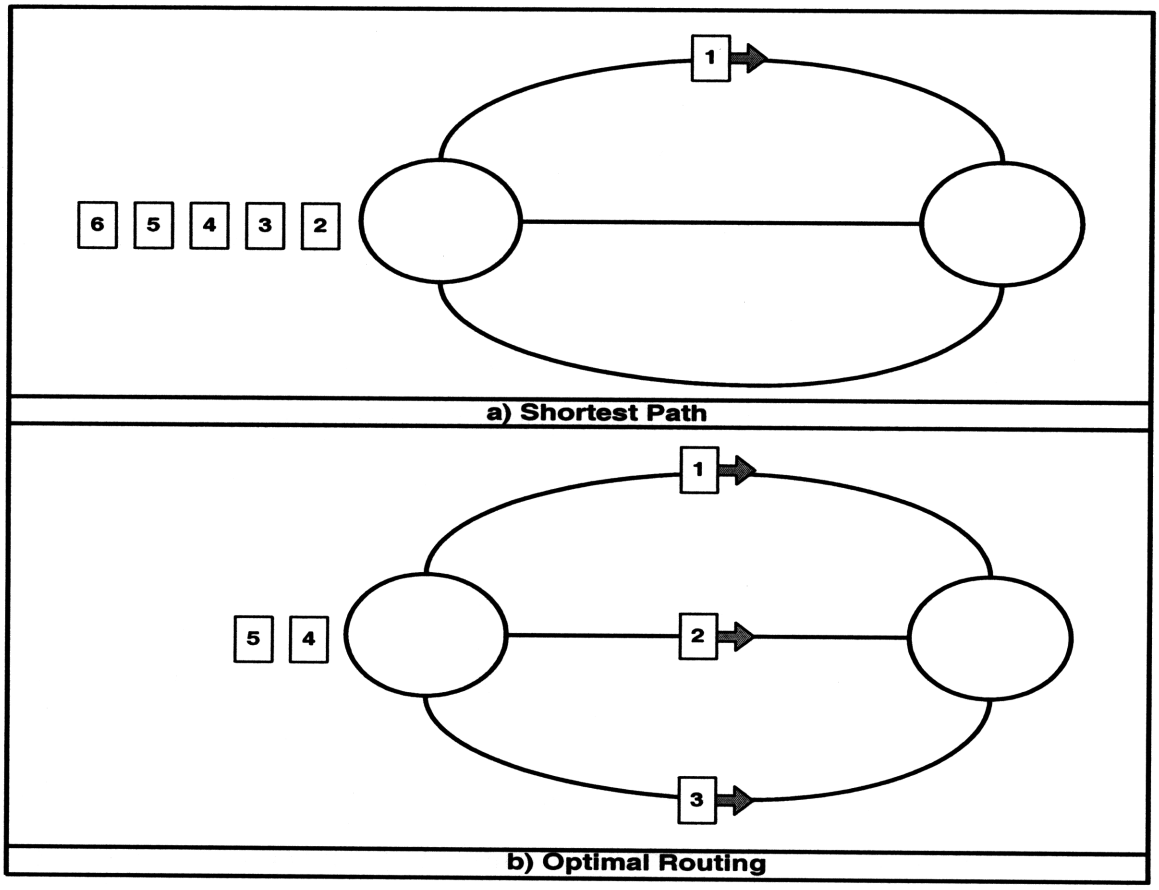

Figure 2-2: Sub-optimality of Shortest Path routing in terms of delay

We will now turn to a different paradigm of routing which is suitable for distributed implementation and tries to utilize the full capacity of a network.

2.1.2 Differential Backlog Routing

Our goal is to develop novel approaches to the autonomous re-routing problem for rapid and efficient failure recovery. Tassiulas and Ephremides [38] proposed an algorithm that solves the problem of maximum flow for a static network with traffic of a single class of priority in the context of multi-hop wireless networks. They proved that the algorithm is optimal in achieving the capacity of a network with stochastic multi-class traffic. This is in comparison to the Ford-Fulkerson algorithm [9] which solves the problem of single commodity maximum flow in a static network. We refer to Tassiulas's algorithm as the *Diffusion Routing* or *Differential Backlog* algorithm since it actively routes traffic to remove heterogeneity and keeps queue backlogs minimized in a network. Traffic is classified based on its destination(s) and kept in different queues. The algorithm tries to keep queue sizes for different classes of traffic balanced.

Making the algorithm of Tassiulas and Ephremides more specific to optical networks, the network consists of N nodes and L links. Let $s(i)$ and $d(i)$ represent the source and destination respectively, of link i. Packets are distinguished by the destination node j that they are headed to. Each node keeps a queue of packets awaiting service, except the packets destined to the node itself which are removed upon arrival at the node. Let $X_{nj}(t)$ be the number of packets at node n destined for node j at the end of time slot t. At each slot t, the routing decision is made as follows: for link i, let

$$\hat{j}_i = \arg \max_{j \in \{1,2,\ldots,N\}} (X_{s(i)j} - X_{d(i)j}).$$

If there are multiple classes of packets which happen to achieve the maximum, one is chosen arbitrarily. Assuming all links are of unit capacity, a packet of class $\hat{j}_i$ is transferred over link i in slot t. To prevent a situation where there might not be enough packets at a node to be served by the links, if number of packets of any class at a node are less than or equal to the number of outgoing links at the node, the outgoing links are left unutilized.

Since the introduction of Tassiulas's algorithm, many further works have specialized or generalized the notion of achieving increased throughput using differen-

tial backlog information in various settings. In [37], Tassiulas presented Differential Backlog algorithm in context of networks with varying topologies. Maximum weight matching in an input-queued switch to attain 100% throughput [24] is just a special case of the Differential Backlog algorithm. Neely et al. generalized the Differential Backlog algorithm taking into account transmitter power constraints in [30] and fairness in [29]. Tassiulas suggested the use of randomized algorithms for the implementation of Differential Backlog algorithm in [36]. However, little practical work has been done to explore difficulties in practical implementation and measure other important metrics of performance beyond capacity of Differential Backlog algorithm. Increased requirements for computing at each node and expected worse delay performance have been the primary concern for practical circles concerned with routing.

As far as the information needs of Differential Backlog routing are concerned, it can be easily implemented in a distributed manner since each node only needs to know about the state of its neighbors. The theoretically proven performance and ease of implementation of Differential Backlog routing makes it an obvious candidate for autonomous re-routing in context of failures. The expected delay of Differential Backlog routing is higher than that of shortest path algorithms because packets can travel in loops. However, it still does not deter us to take a deeper look at some of the practical issues of Differential Backlog routing and compare its performance with Shortest Path routing.

2.2 Approach

Since a theoretical treatment of the delay of Differential Backlog routing algorithm appears unwieldy at first sight, we decide to take the route of computer simulations. Developing simulations will not only allow us to focus on issues that might arise in implementation of Differential Backlog routing but will provide us with very practical metrics of performance such as end-to-end packet and file delays, queue-sizes and computational complexity. We can also evaluate the performance of Differential Backlog routing under a variety of traffic conditions, failure situations and real

networks to find out how it compares with Shortest Path. Lastly, one can also try to tweak Differential Backlog routing algorithm to overcome any apparent flaws and improve its performance.

Chapter 3

Performance Evaluation of Differential Backlog Routing

3.1 Methodology

3.1.1 Assumptions

We assume that time is slotted. All packet transmissions are completed by the end of a slot. In a slot, events take place in the following order: link statuses change, files arrive and packets get transmitted.

Arrivals

There has been a history of using Poisson traffic in network modeling and analysis [18]. Karagiannis et al. have observed similar trends in the traffic patterns experienced by an Internet babckbone [16]. We choose Poisson arrivals in our simulations also for their simplicity of analysis and implementation.

Files arrive at the beginning of each slot at each node according to a Poisson process with an average of λ files per slot. A file is equally likely to be destined to any node in a network except the source node. The number of packets in each file is a geometric random variable with parameter p.

Failures

As we seek to investigate the problem of routing in context of wide-spread network failures, we need a framework for evaluating the performance of Differential Backlog routing under failures. Herein, we concentrate on studying the effect of link failures and ignore node failures which are less frequent. At the beginning of each slot, each live link has a probability p_f of failing. Similarly each failed link becomes re-activated with probability p_s. All links start in the live state. The *steady state probability* [3], π_f, of being in the failed state is given by $\frac{p_f}{p_f+p_s}$.

3.1.2 Network Topologies

An important question that comes up during treatment of a concept based on simulation is the choice of background scenarios. Our primary focus in making these decisions has been on practicality, suitability for Differential Backlog routing and computational complexity. We have considered two network topologies as we analyze the performance of Differential Backlog routing along with its different variants and Shortest Path routing.

10-node 4-connected symmetric topology

The first is a 10-node symmetric topology as shown in Fig. 3-1 where each node is connected to four other nodes.

Qwest OC-192 Backbone

Secondly, we use the network of high capacity OC-192 links from the Qwest backbone presented earlier in Fig. 1-3 to analyze the performance of Differential Backlog in a representative real setting. The topology of Qwest OC-192 backbone used in our simulations is shown in Fig. 3-2.

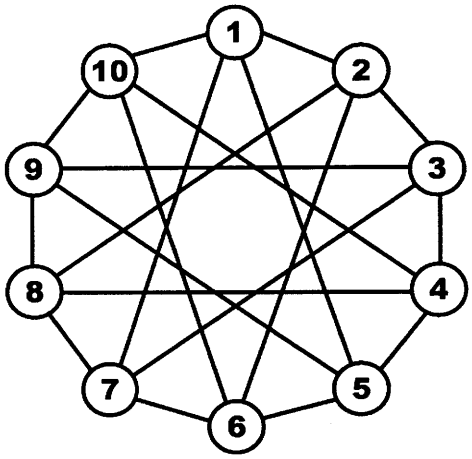

Figure 3-1: 10-node 4-connected symmetric topology

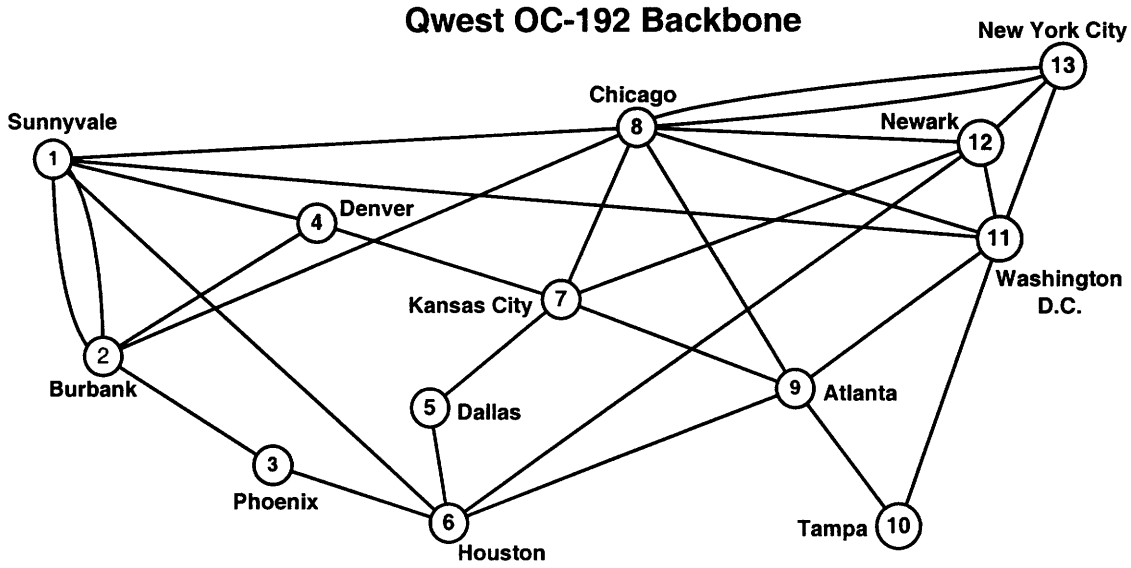

Figure 3-2: Qwest OC-192 backbone

3.1.3 Implementation Details

Since links have been implemented just as conduits for traffic flow and all the buffering is done at the nodes, no packets are lost when a link fails. Hence no packets are dropped or lost and assuming links will be re-activated eventually, there is no need for error correction or transport control.

3.1.4 Performance Metrics

There are several metrics for evaluating the performance of a routing algorithm as described in Section 1.5. Our simulations allow us to extract data on file and packet delays, computational complexity and buffer sizes. In our analysis, we present quantitative results on *end-to-end delays* and use them as the primary criterion for comparing the performance of different routing schemes. *End-to-End delay* is the time a file or packet takes to reach its destination after its arrival at the source node . End-to-End delay constraints play a vital role in smooth functioning of most real world applications such as email, internet browsing, voice and video conferencing etc. and thus can genuinely capture the utility of a routing algorithm. Nevertheless, we do occasionally make qualitative comparisons of buffer sizes and computational complexity in performance analysis of different routing schemes.

3.1.5 Simulation Execution

Simulations can complement theoretical analysis of a phenomenon but can rarely serve as a substitute. The limitations of simulations lies in the finiteness of their life times and specificity of their results. For instance, it is hard to conclude whether a particular traffic load is stable even if buffer sizes do not increase significantly over long intervals. Similarly, insights gained from simulation study of a few topologies cannot be generalized. We, however, take utmost care to make sure that simulations converge to a stable point by comparing results from repetitive runs. Also, buffer sizes must become relatively stable for a simulation to be deemed useful.

3.1.6 Results Presentation

End-to-End file and packet delay results are extracted for each possible *session* in a network. However, due to large number of these sessions and difficulty to differentiate between them in terms of their usefulness, we take their averages. The average end-to-end file and packet delays over all sessions, thus, convey a wholistic view of delays experienced by a network.

As will become apparent, there are several parameters that govern a simulation: network topology, λ, p, p_f, p_s, for instance. We present selected results that capture a general trend or convey an interesting point. The latter required careful adjustment of parameters and do not convey trends that hold in general. These cases will be identified as such and the emphasis is placed on the existence of these regions of operation.

3.2 Differential Backlog Routing

3.2.1 Algorithm Description

The underlying idea behind Differential Backlog routing is to use all of the network resources to distribute data, differentiated by destination, evenly throughout the network. Since a destination acts as a traffic sink, the net flow of traffic is from all of the data sources to each of the destinations. In mathematical terms, borrowing Neely's notation [28], let $U_a^{(c)}(t)$ be the number of packets waiting at node a destined for node c at time t. For each pair of directly connected nodes, let's say a and b, the commodity $c_{ab}^*(t)$ with the highest differential backlog, i.e.

$$c_{ab}^*(t) = \arg \max_{c \in 1,\ldots,N} \{U_a^{(c)} - U_b^{(c)}(t)\} \tag{3.1}$$

is transmitted over the link (a, b) at time t.

After all arrivals for a slot take place, each link is marked with the commodity that has the maximum differential backlog across that link, with ties broken randomly. For

links with a positive differential backlog, as many packets of the marked commodity as the capacity of the concerned link are transmitted across the link. To simplify preliminary analysis, we assign a capacity of one packet per slot to all links in our simulations. It is possible that there are not enough packets of a commodity for transmission over all the outgoing links marked with it at a node. In case of such a shortage, the outgoing links marked with the commodity are served in a random order as long as packets of the commodity remain at the node. The remaining links are left unutilized for the slot. All packet transmissions are completed by the end of the slot.

Like the approach used in [37] for applying Differential Backlog routing to networks with time-varying topologies, the links are further constrained under failures. A failed link cannot transmit any packets from its source to its destination in a slot.

3.2.2 Implementation Details

The queue at each node is implemented as $n-1$ FIFO buffers to hold packets destined to each of the other nodes in the network. The computer memory held by these buffers is managed dynamically and hence there are virtually no limits (apart from hardware constraints on memory) to buffer sizes as long as the loading is experimentally stable. Consequently, no transport control or error-recovery mechanism is needed to ensure loss-less delivery of packets. Each link can transmit up to one packet in every slot from the link's source to its destination. In each slot links are processed in a random order to achieve random breaking of ties in the case of multiple links competing for a limited commodity at a node.

3.2.3 Results

Symmetry in 10-node 4-connected symmetric topology

Since each node can receive a file request for any other node in the network, there are a total of ninety different sessions, namely $(1,2), (1,3), (1,4), ..., (2,1), (2,3), 2,4), ..., (10,9)$. However, due to the inherent symmetry of the network, many of these sessions

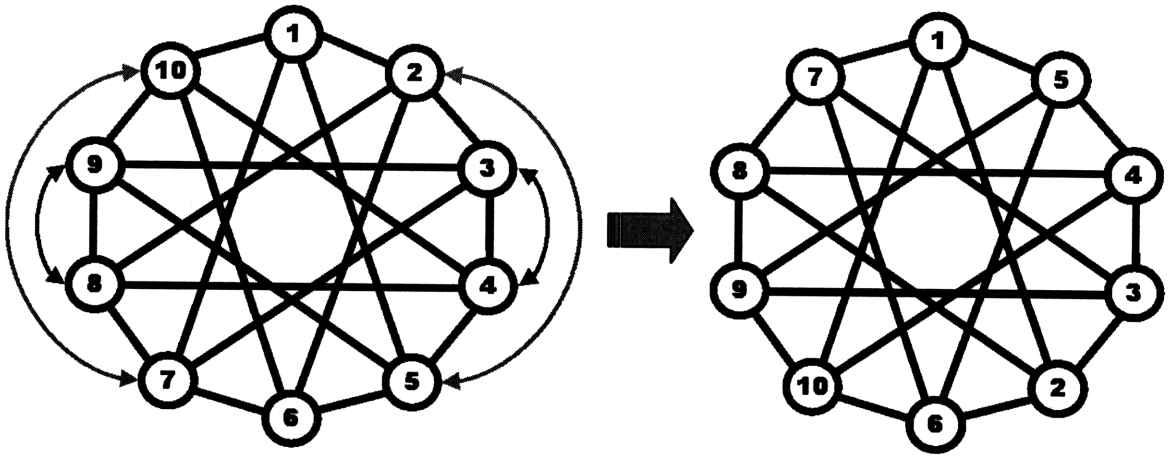

Figure 3-3: Reconfigurable symmetry in the 10-node 4-connected symmetric topology

are symmetric. Since the links are bi-directional, the reciprocal sessions such as $\{(1,2),(2,1)\}$ and $\{(1,3),(3,1)\}$ must be symmetric. Rotational symmetry leads to sessions such as $\{(1,2),(2,3),...,(10,1)\}$ and $\{(1,3),(2,4),...,(10,2)\}$ being identical. Also, due to the reflexive symmetry around any line joining two opposite nodes, sessions such as $\{(1,2),(1,10)\}$ and $\{(1,3),(1,9)\}$ are identical as well. Although, it might not be obvious from the topology, a closer examination of the topology reveals that sessions such as $\{(1,2),(1,5)\}$ and $\{(1,3),(1,4)\}$ are also symmetric. Fig. 3-3 helps to recognize this reconfigurable symmetry. The node pairs: $\{2,5\}$,$\{3,4\}$,$\{7,10\}$ and $\{8,9\}$, can be graphically swapped all-at-once without changing the underlying graph. After the swap every node would still be connected to the same set of nodes as before the swap. The new graphical layout shown in Fig. 3-3 clearly demonstrates why the above-mentioned sessions must be symmetric. Hence, the number of essentially different sessions is three.

It is good practice to perform sanity checks for ensuring correctness of simulations. Simulation results for individual sessions were checked against the above-mentioned symmetries. Infact, the reconfigurable symmetry was discovered through simulation results shown in Fig. 3-4. Here, sessions have been grouped according to their degrees of separation along the rim of the decagon which takes into account the first three kinds of symmetries namely reciprocal, rotational and reflexive. The classification yields five groups of sesssions and the average file and packet delays of all sessions

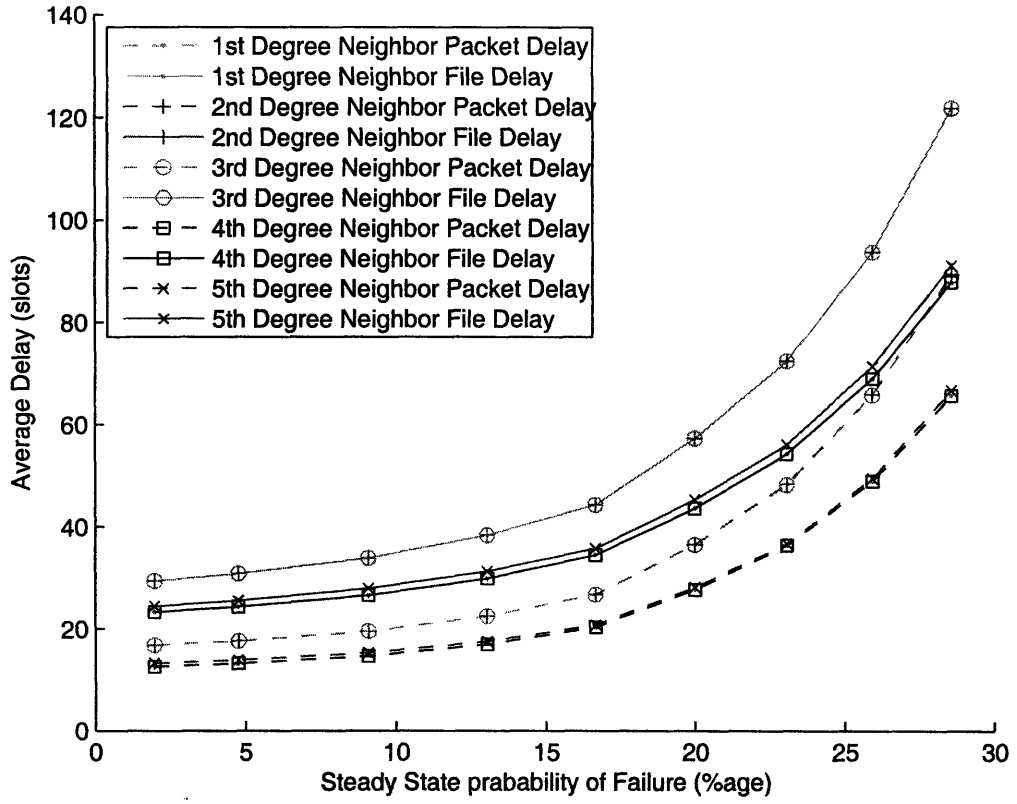

Figure 3-4: Evidence of the reconfigurable symmetry in the 10-node 4-connected symmetric topology

in a class are plotted against increasing rate of failures. However, the results show existence of just three fundamentally different classes. The delays of first and fourth degree neighbors are equal; and the delays of second and third degree neighbors are equal. This observation confirms reconfigurable symmetry in 10-node 4-connected symmetric topology.

From now onwards, for sake of better presentation and in the interest of studying more useful delay trends, we just plot average end-end file and packet delays over all sessions. Results for mutliple values of a control variable are often plotted for better utilization of the graphing space.

Delay response to variations in Network Loading

We will study all the governing parameters of simulations one by one. First, we focus on variables in the loading model. The average loading at a node in terms of number of packets received per slot is given by:

E[number of packets arriving at each node]

= E[number of file arrivals] x E[number of packets in a file],

Since file arrivals and number of packet arrivals are assumed to be independent processes.

$$= (\lambda\ ) \left(\frac{1}{p}\right) = \frac{\lambda}{p}.$$

We obtain results for different levels of network loading. For each network load, we plot results for three $\{\lambda, p\}$ pairs which yield the same network load. The analysis allow us to study the behavior of increased network loading and average file sizes.

10-node 4-connected symmetric topology The average file and packet delays over all sessions in the 10-node 4-connected symmetric topology, are plotted against increasing network load in Fig. 3-5. Netwok loading is varied through increases in average file arrival rate values. As expected the delays increase with average file arrival rate λ. The growth can be best described as exponential.

Fig. 3-5 also shows the variation of file delays with respect to the parameter p which governs the number of packets in a file. We see that as average number of packets in a file $\frac{1}{p}$ decreases, the average end-to-end delays tend to improve. This is expected because, decreasing average number of packets per file and increasing file arrival rate λ proportionately (to keep network loading constant) leads to a more uniform traffic arrival with respect to time. Hence sizes of input queues at nodes remain smaller and packets see less queueing delays at the source nodes.

Fig. 3-6 plots the average packet delays for better comparison. Average packet delays are typically less than average file delays but exhibit the same trend.

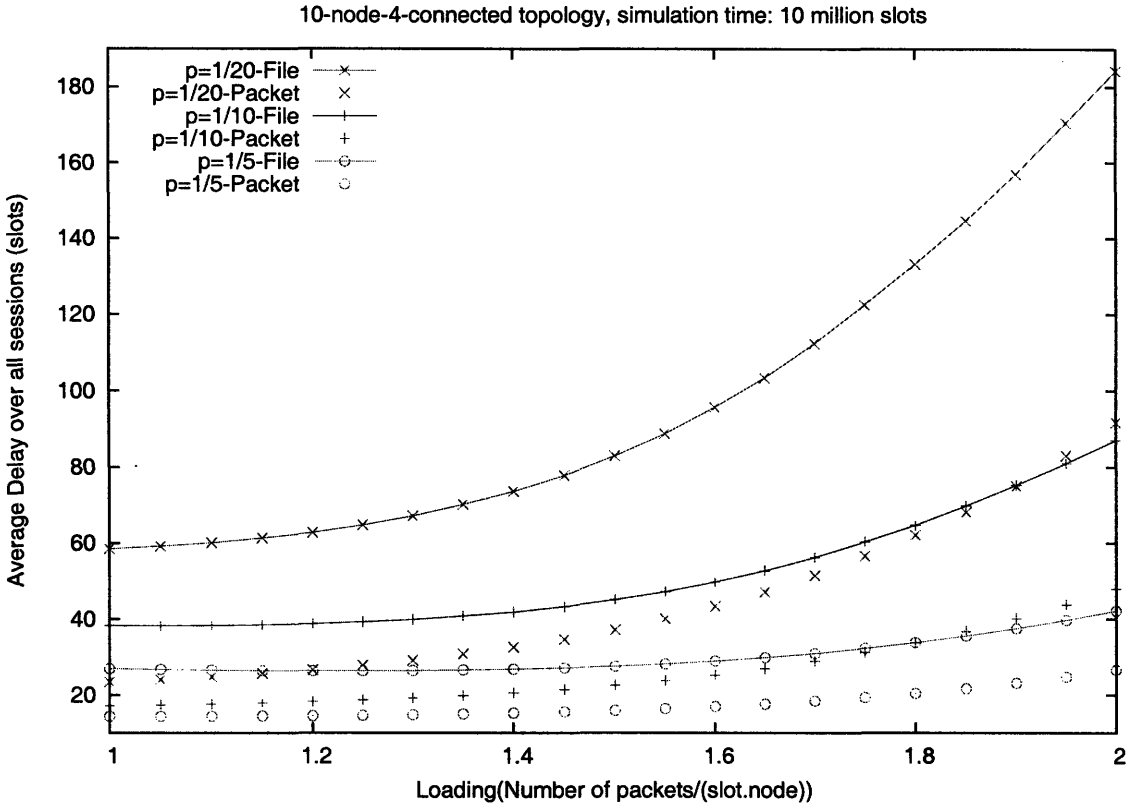

Figure 3-5: File and packet delays under variation in network loading for different average file sizes in the 10-node 4-connected symmetric topology

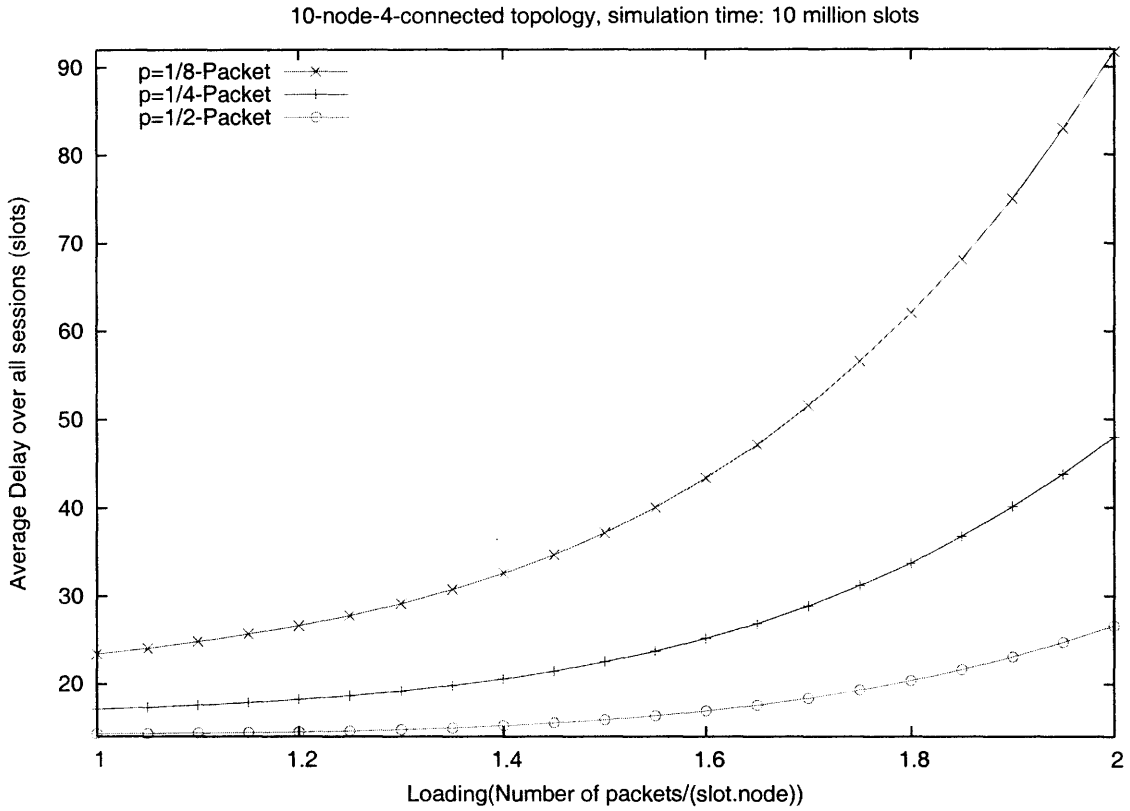

Figure 3-6: Packet delays under variation in network loading for different average file sizes in the 10-node 4-connected symmetric topology

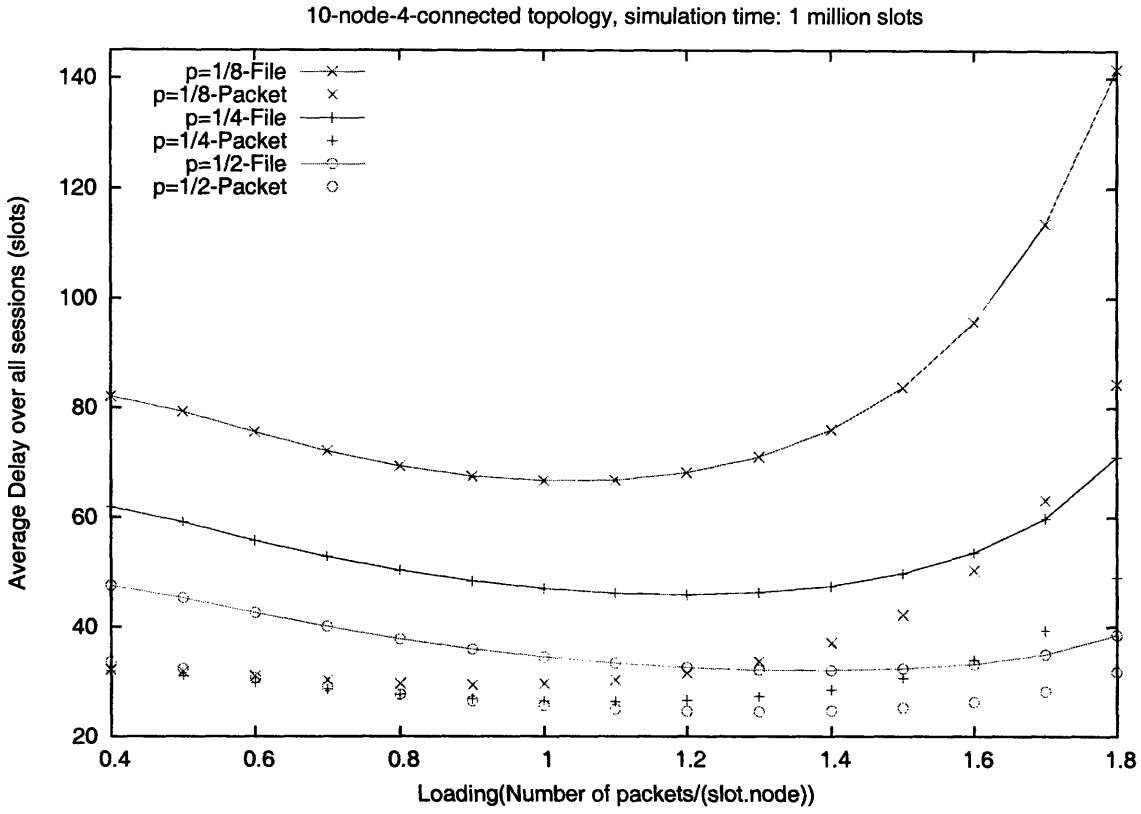

Figure 3-7: File and packet delays under variation in network loading for different average file sizes in Qwest OC-192 Backbone

Qwest OC-192 Backbone The results for average file and packet delays for Qwest OC-192 backbone are presented in Fig. 3-7. The results confirm the trends observed for 10-node topology, namely higher delays for higher average file sizes and general increase in delays with increased loading. However, a notable difference is the observed increase in delays as loading decreases. This phenomenon has been predicted by Neely in his doctoral thesis [28]. If a network is lightly loaded, the absence of backlog pressures can contribute to packets taking random walks. The delays increase because of the increased time taken to reach a destination using a random walk approach.

In Fig. 3-8 we plot average packet delays which, as in the 10-node topology case, follow file delays but are smaller.

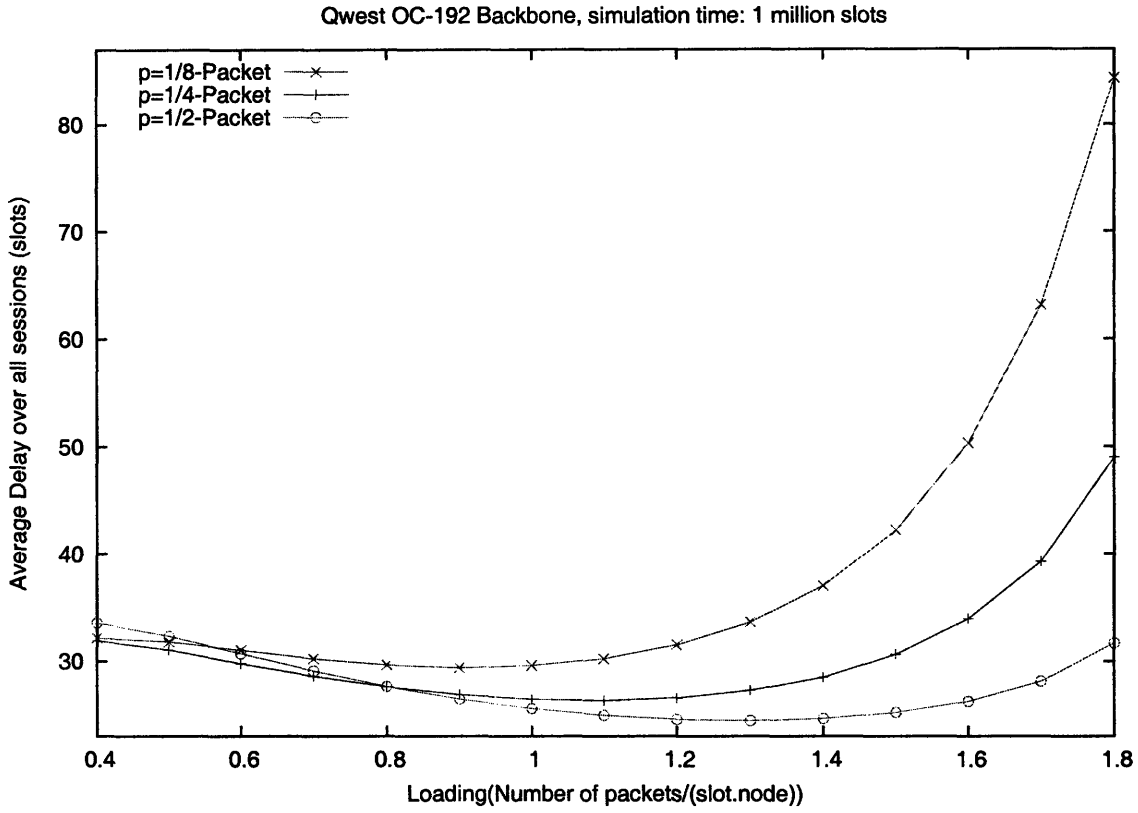

Figure 3-8: Packet delays under variation in network loading for different average file sizes in Qwest OC-192 Backbone

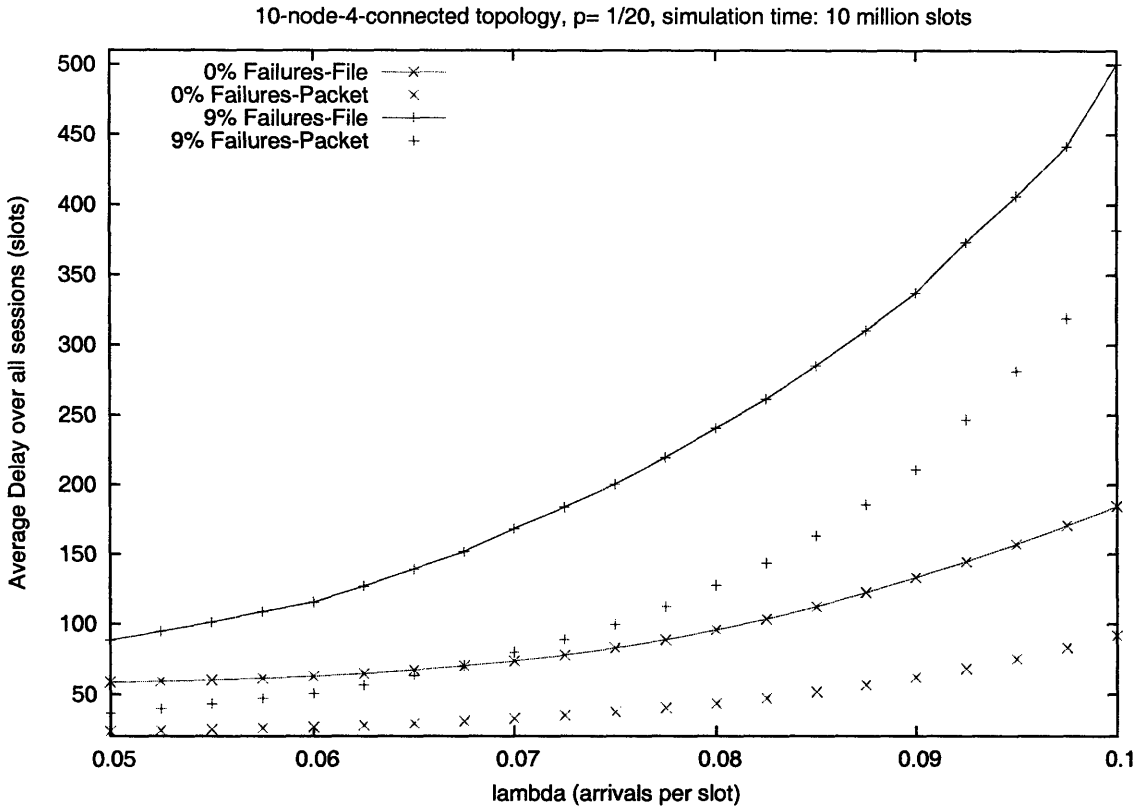

Figure 3-9: File and packet delays under variation in network loading with and without failures in the 10-node 4-connected symmetric topology

Delay response to variations in Failure Rate, p_f

After identifying the delay trend with respect to average file sizes in the previous section, we pick a value for average file size, p, and use average file arrival rate λ to vary network loading. We present delay results for variations in average file arrival rate, λ, for different values of failure rate.

10-node 4-connected symmetric topology The delay results for 10-node 4-connected symmetric topology are presented in Fig. 3-9 against increasing values of average file arrival rate λ in the backrop of two scenarios: one without failures and the other one where 9% of links are in the failed state on average. The delays rise exponentially with λ and as expected, delays in the failure setting are larger and appear to grow at a faster rate than the setting without any failures.

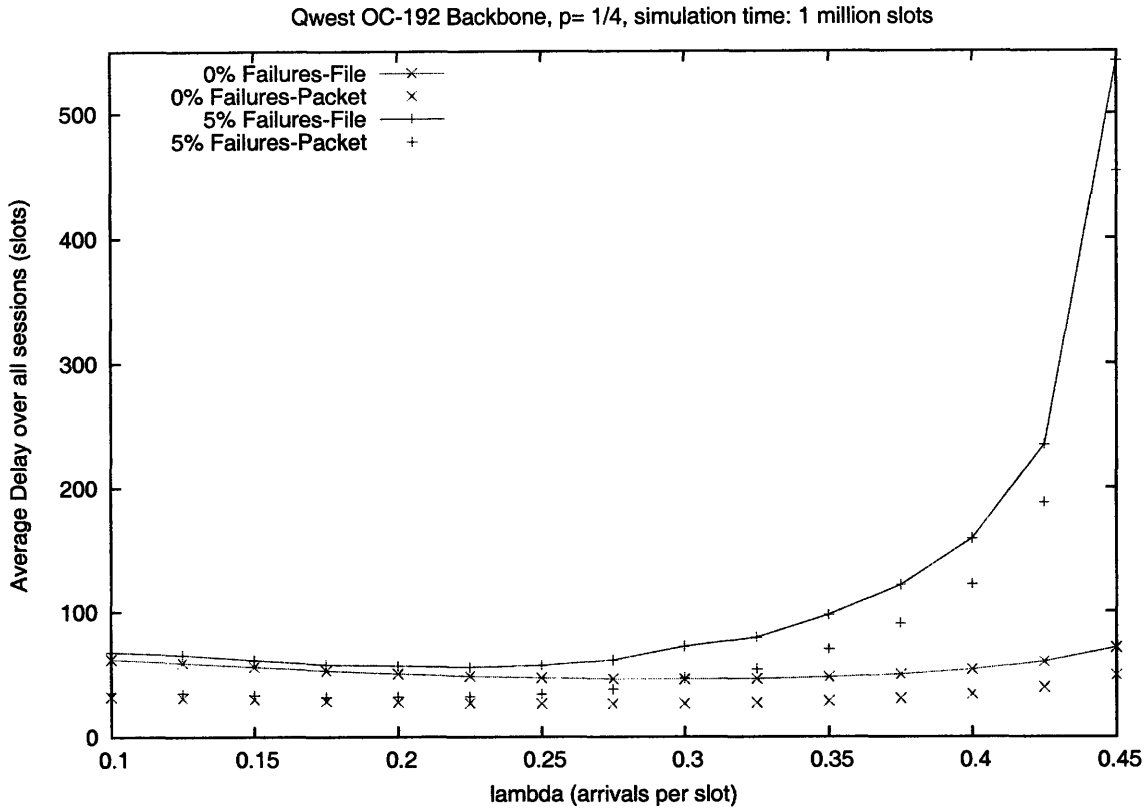

Figure 3-10: File and packet delays under variation in network loading with and without failures in Qwest OC-192 Backbone

Qwest OC-192 Backbone The delays for Qwest OC-192 Backbone are presented in Fig. 3-10. The same trends are observed as those observed for the 10-node 4-connected symmetric topology. In addition, the familiar trend of increasing delays with decreasing λ is observed again to hold for failure settings as well. Fig. 3-11 plots just average packet delays for a better comparison which again shows an exponential increase in packet delays with λ and slight increase in delays for low values of λ.

After studying the behavior of a network with failures over a range of loading conditions, we are inclined to analyze the delay behavior with respect to changes in failure rate.

10-node 4-connected symmetric topology We plot delays with increasing failure probability for the case of 10-node 4-connected symmetric topology in Fig. 3-12. As expected, delays become worse exponentially with increasing probability of fail-

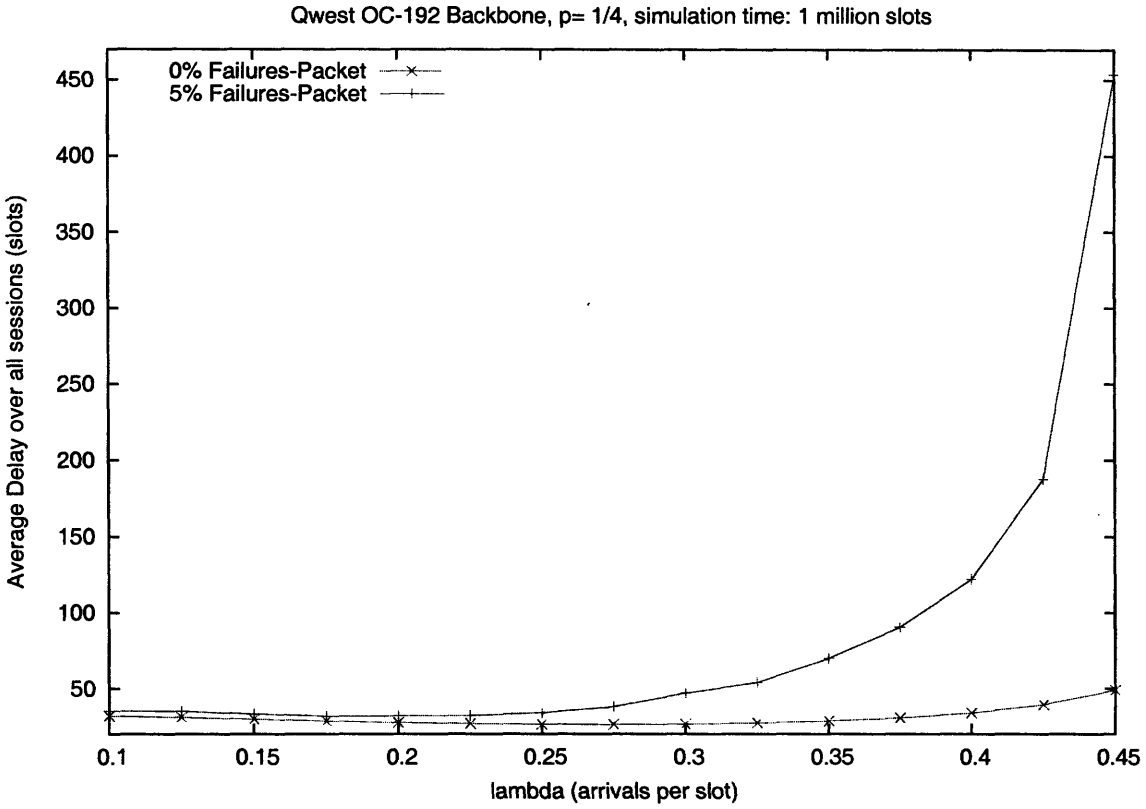

Figure 3-11: Packet delays under variation in network loading with and without failures in Qwest OC-192 Backbone

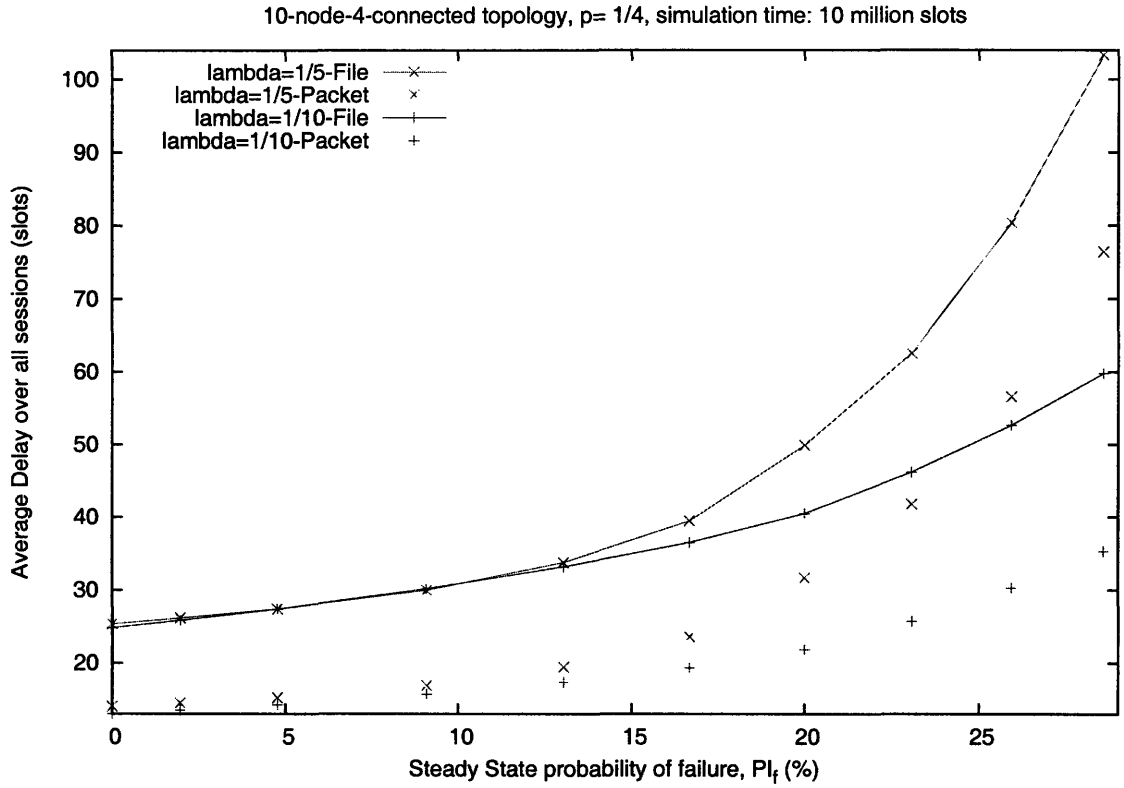

Figure 3-12: File and packet delays under variation in failure rate for different values of network loading in the 10-node 4-connected symmetric topology

ures and are larger for higer values of average file arrival rate λ. Interestingly, there is a region of failure rates where file delays for different values of loading are approximately equal. This shows that Differential Backlog routing can accomodate low failures without significant degradation in delays.

Qwest OC-192 Backbone Results for Qwest OC-192 Backbone are presented in Fig. 3-13. They show the same trend as described earlier for 10-node 4-connected symmetric topology. Again, we see the trend familiar for this topology where lower stress on the network might prove counter-productive. We observe that at lower failure rates, a heavier network load yields better delay performance which might be counter intuitive in context of conventional shortest path algorithms. However, we once again appeal to the explanation offered by Neely [28] citing light loadings to justify this behavior. Fig. 3-14 plots only packet delays which are smaller than file

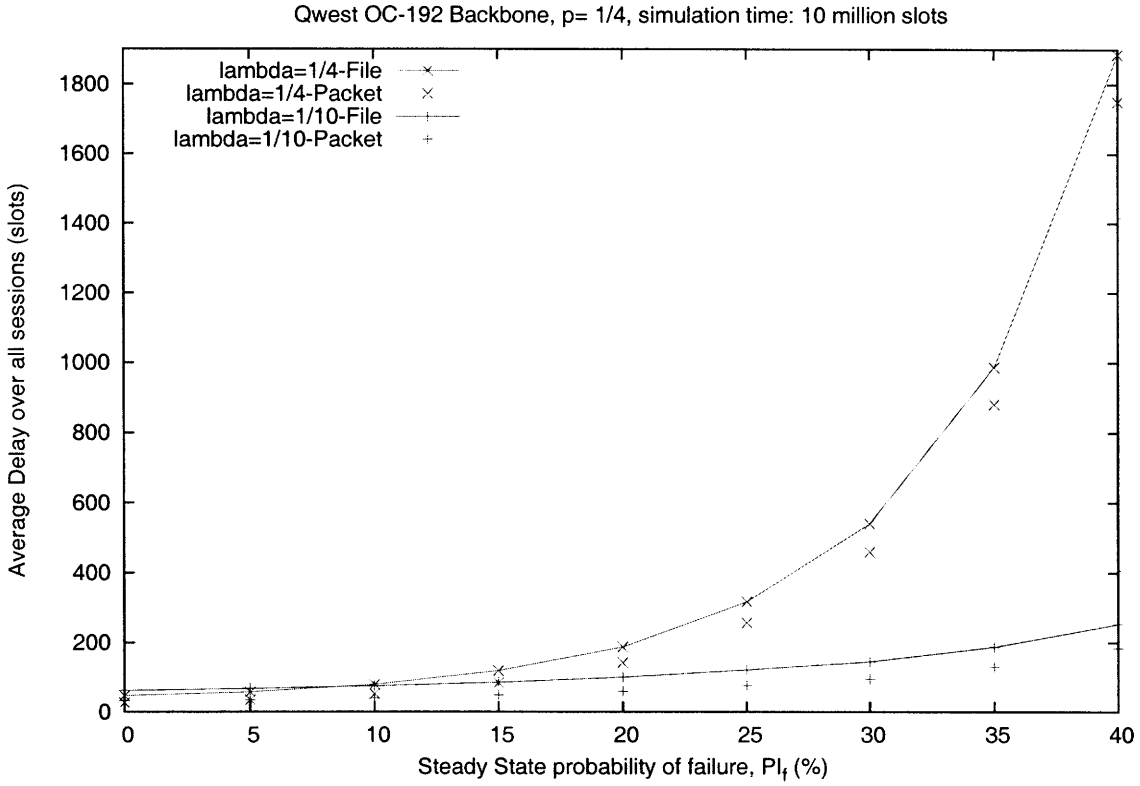

Figure 3-13: File and packet delays under variation in failure rate for different values of network loading in Qwest OC-192 Backbone

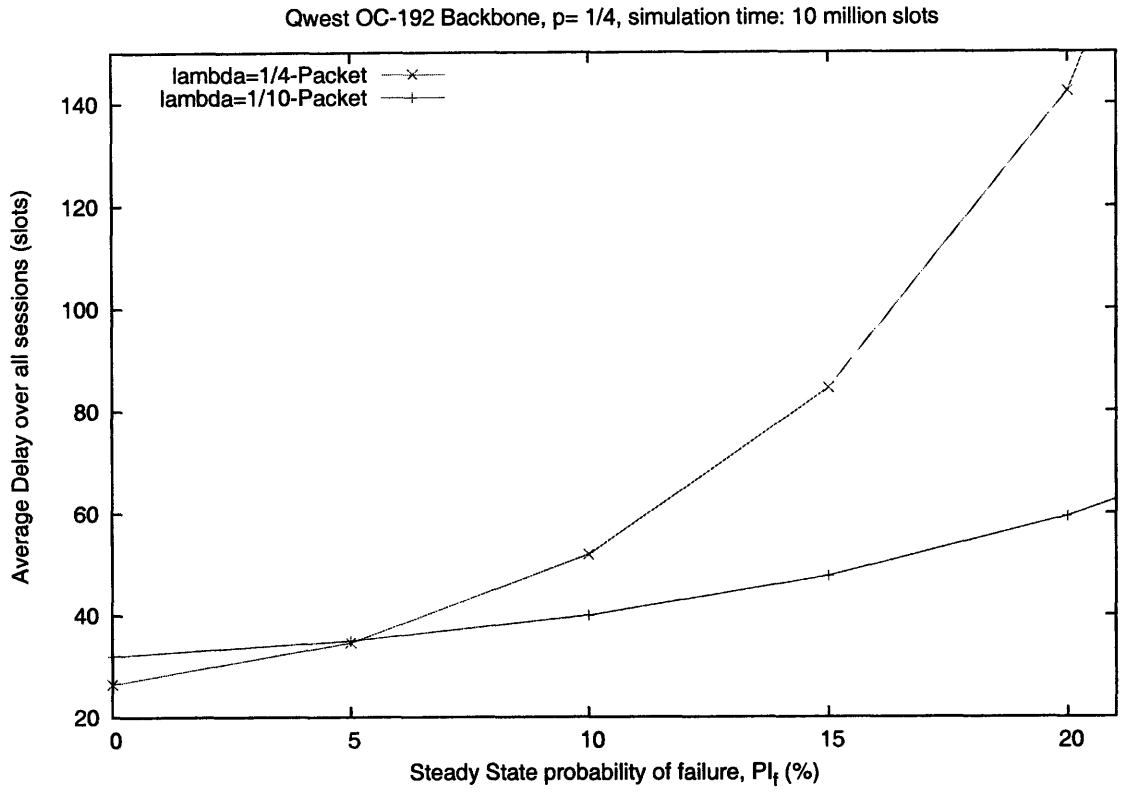

Figure 3-14: Packet delays under variation in failure rate for different values of network loading in Qwest OC-192 Backbone

delays as usual and show the same trends as file delays. We make special notice of better performance of network at higher network loading and lower failure rates once again.

3.3 Comparison with Shortest Path

Most of the routing in Internet is based on shortest path algorithms such as OSPF. Here we compare Differential Backlog routing to a model shortest path algorithm.

3.3.1 Algorithm Description

Each link in the network has a cost associated with it that depends on its capacity. The cost of a path is equal to the sum of the costs of links that form the path. In each slot, a node computes minimum-cost paths to all other nodes in the network based

on link statuses in the slot. When more than one path achieves the minimum, one is chosen arbitrarily. A node directs each received packet, except those destined to it, to the link that constitues the shortest path to the packet's destination. As long as there are packets awaiting, each link transfers as many packets as its capacity to its destination. As we did in the case of Differential Backlog Algorithm, we will assign a capacity of one packet per slot to all links in our simulations. All packet transfers are considered complete by the end of the slot.

3.3.2 Implementation Details

We use Dijkstra's All Pairs Shortest Path algorithm to implement Shortest Path routing. As each link can transfer up to one packet in each slot, the cost of all links are set to be equal. Hence, path costs are equal to hop counts and Shortest Path routing amounts to Minimum Hop routing in our case. Each node maintains an address classifier to route packets to outgoing links based on their destinations. Incoming packets to a link arrive at a FIFO queue with virtually unlimited capacity. In addition to queues at the links, there are input queues at the nodes which buffer new arrivals to the nodes as well as routed packets from other parts of the network. These node queues are served fully only once per slot. Doing so restricts the number of links a packet can traverse in one slot to a maximum of one, removing any dependency on the order in which queues at links are processed. The address classifiers are updated every t slots to account for any changes in network topology resulting from any link failures or activations. When a link fails, all packets residing in its queue get redirected to the link source.

3.3.3 Results

Delay response to variations in Network Loading

10-node 4-connected symmetric topology Fig. 3-15 compares the performance of Differential Backlog routing and Shortest Path routing. In the Shortest Path case, packets are routed deterministically strictly towards their destinations; hence Shortest

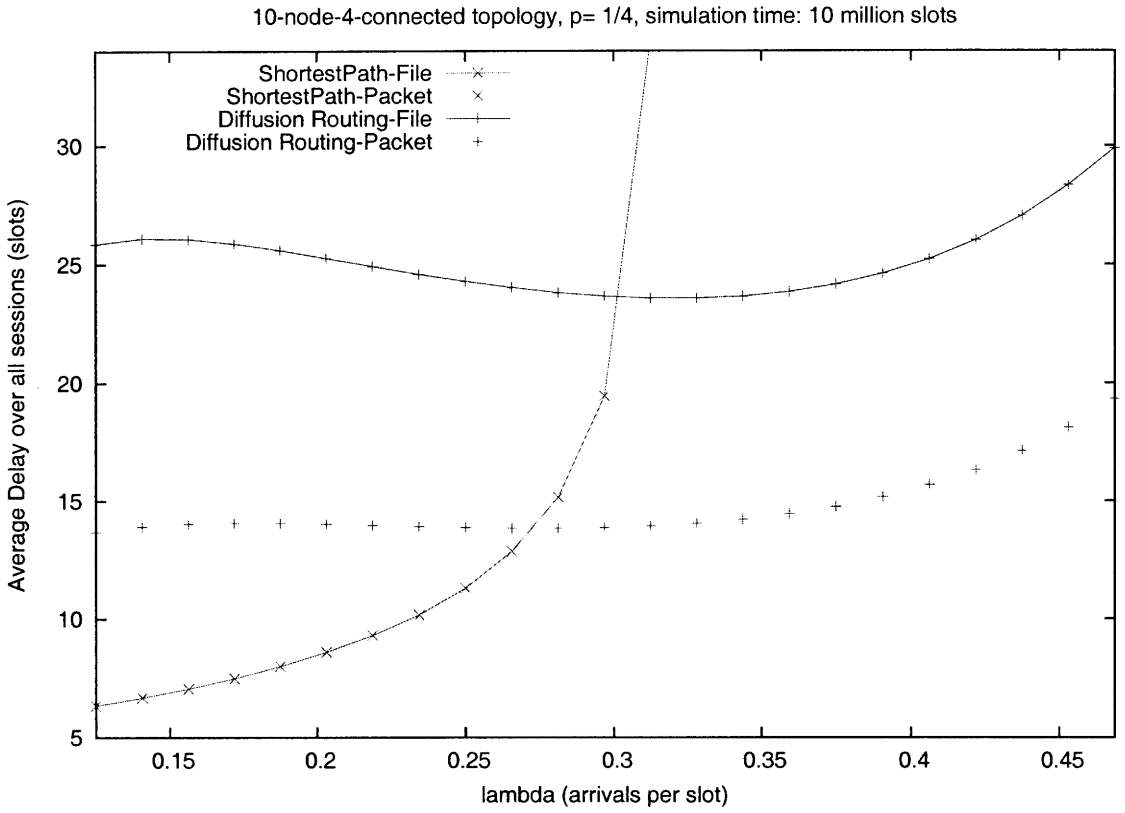

Figure 3-15: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in network loading in the 10-node 4-connected symmetric topology

Path routing behaves better in terms of delays as long as the network load is stable. However, since Shortest Path does not utilize alternative paths which might exist in a topology such as the 10-node 4-connected symmetric topology shown in Fig. 3-1, it does not achieve the full capacity of a network topology. In contrast, Differential Backlog routing incurs larger delays but can support far greater network loads than those supported by Shortest Path. Fig. 3-16 plots the same quantities for an extended x-axis and serves to illustrate the capacity of Differential Backlog routing. Differential Backlog routing becomes unstable at roughly twice the loading at which shortest path becomes unstable. Hence, Differential Backlog routing shows a capacity gain of 100 percent over Shortest Path routing in this example.

Qwest OC-192 Backbone Qwest OC-192 Backbone is subjected to a similar treatment as that of the 10-node 4-connected symmetric topology above in Fig. 3-17.

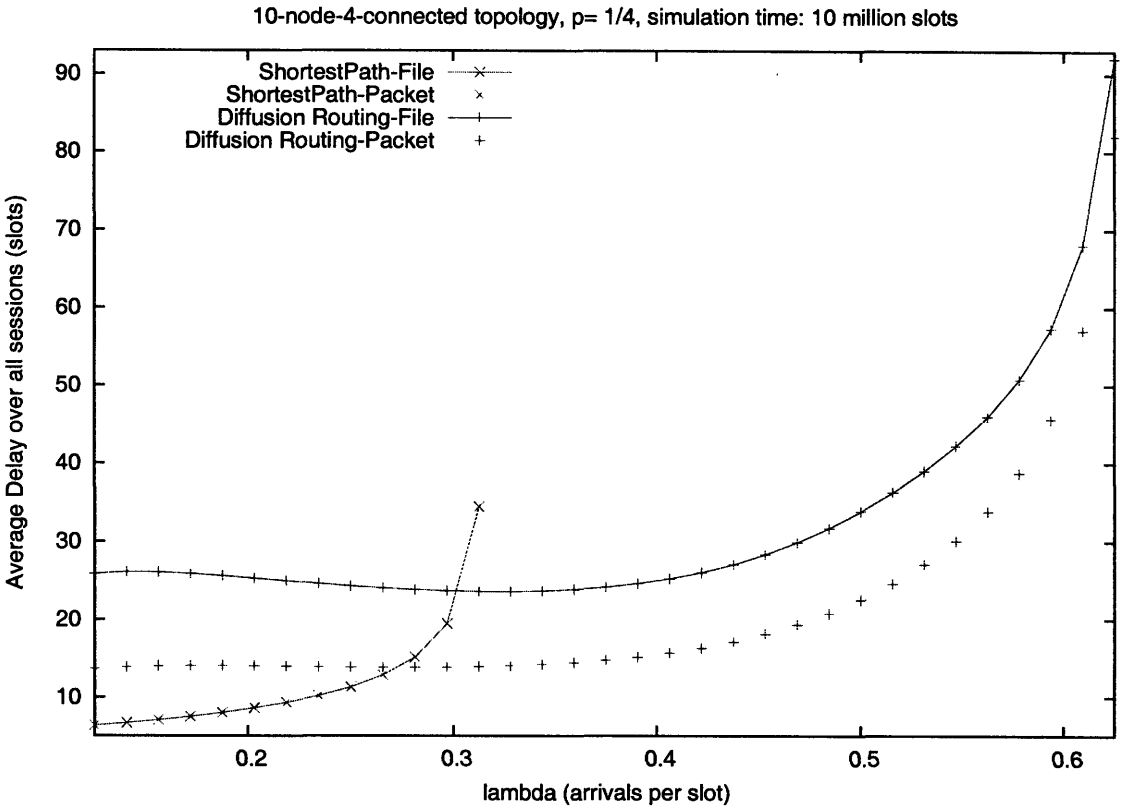

Figure 3-16: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in network loading in the 10-node 4-connected symmetric topology- extended

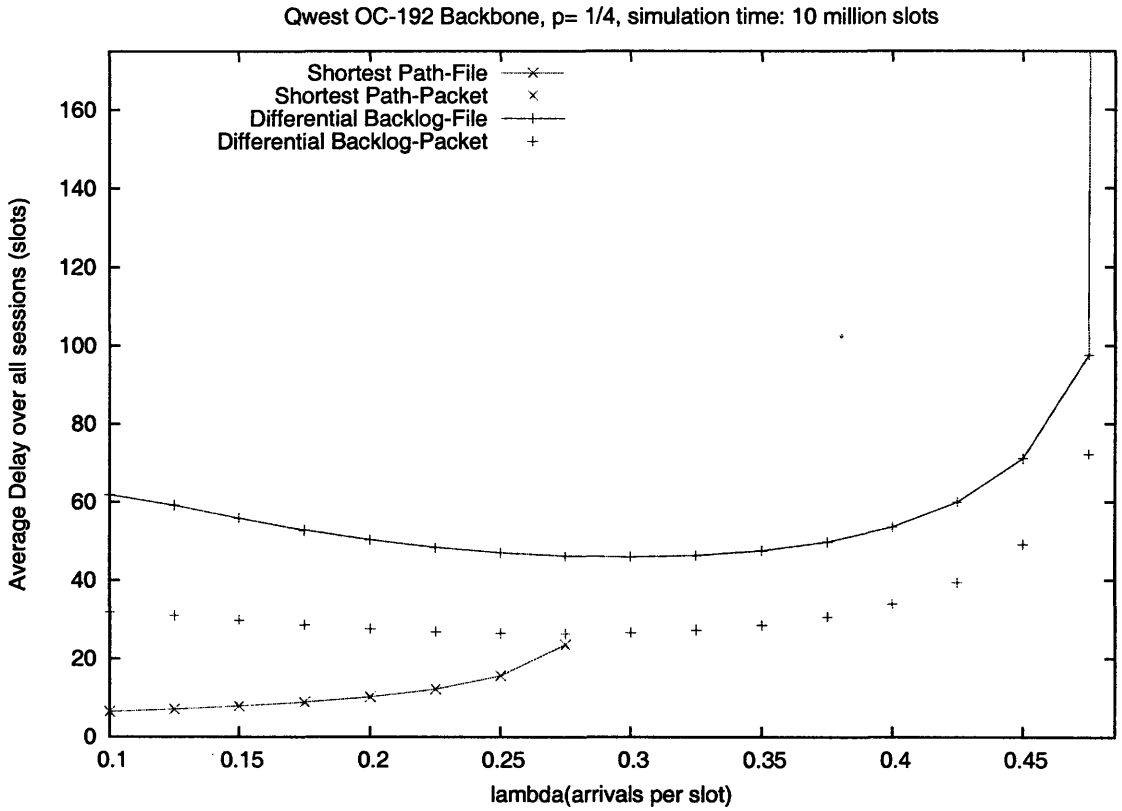

Figure 3-17: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in network loading in Qwest OC-192 Backbone

The results corroborate the trends observed in the 10-node 4-connected symmetric topology.

Delay response to variations in Failure Rate, p_f

Next, we plot delays against increasing failure probability for two values of network loading. The examination allows us to infer general behavior of Shortest Path routing to failures in comparison with Differential Backlog routing. Results for different values of loading can offer insights on how network loading affects the delay behavior in the presence of failures.

10-node 4-connected symmetric topology Fig. 3-18 confirms the notion that shortest path has lower delays for most of the region where it converges. Fig. 3-19 plots the same quantities for a higher value of loading. Shortest Path becomes worse

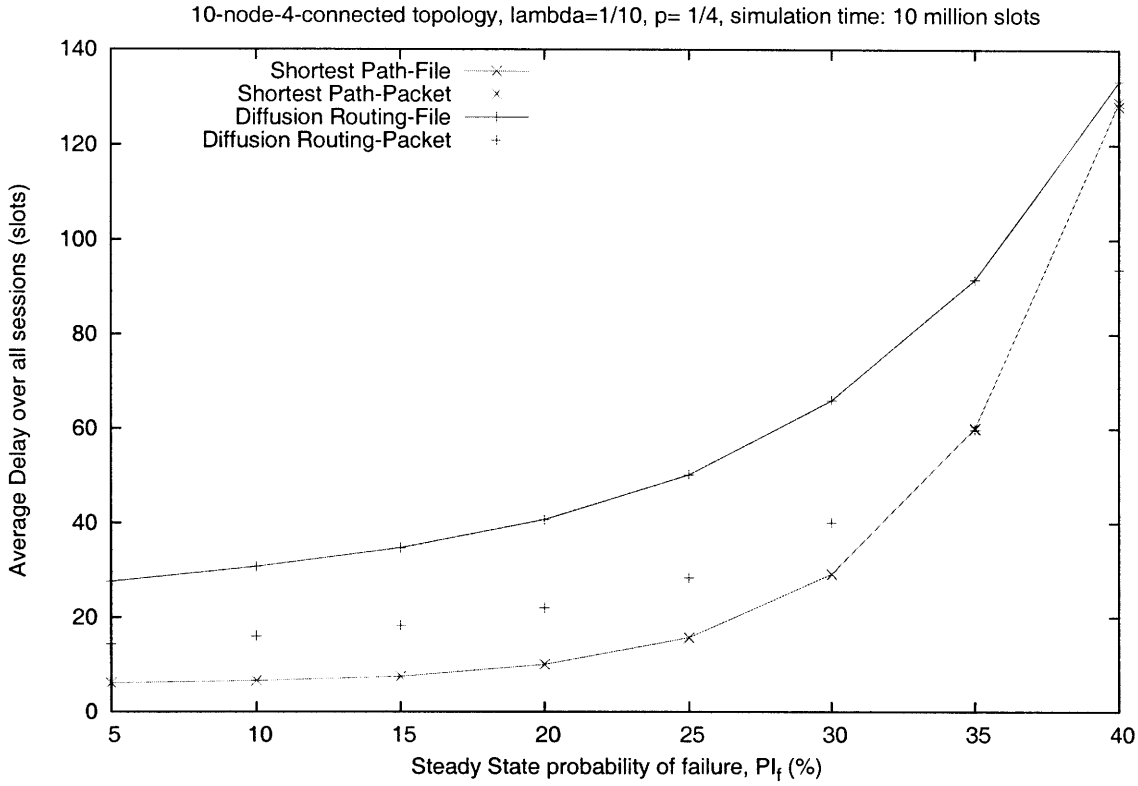

Figure 3-18: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in failure rate in the 10-node 4-connected symmetric topology at low network loading

in delay performance at lower levels of failures than those in Fig. 3-18 because of the increased overall load on the network.

Qwest OC-192 Backbone Fig. 3-20 and Fig. 3-21 plot delays for different rates of failures in Qwest OC-192 Backbone at low and high network loading, respectively. Although the results differ quantitatively from those obtained for the 10-node 4-connected symmetric topology, they are remarkably similar, and therefore enforce our previous analysis.

We conclude from analysis of results in this section that whenever Shortest Path converges, it promises much better delay performance than Differential Backlog most of the time. However, at higher loads, Shortest Path routing becomes unstable whereas Differential Backlog routing continues to function smoothly. The choice between Differential Backlog routing and Shortest Path routing is very clear given

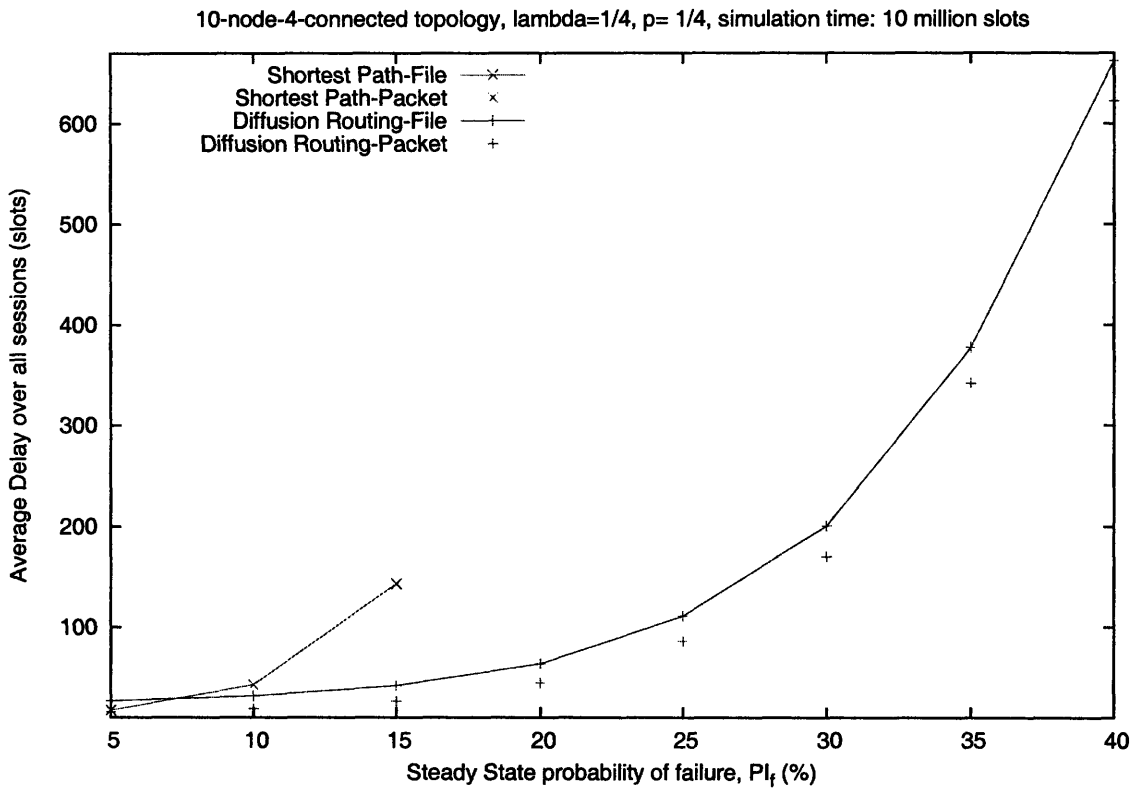

Figure 3-19: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in failure rate in the 10-node 4-connected symmetric topology at high network loading

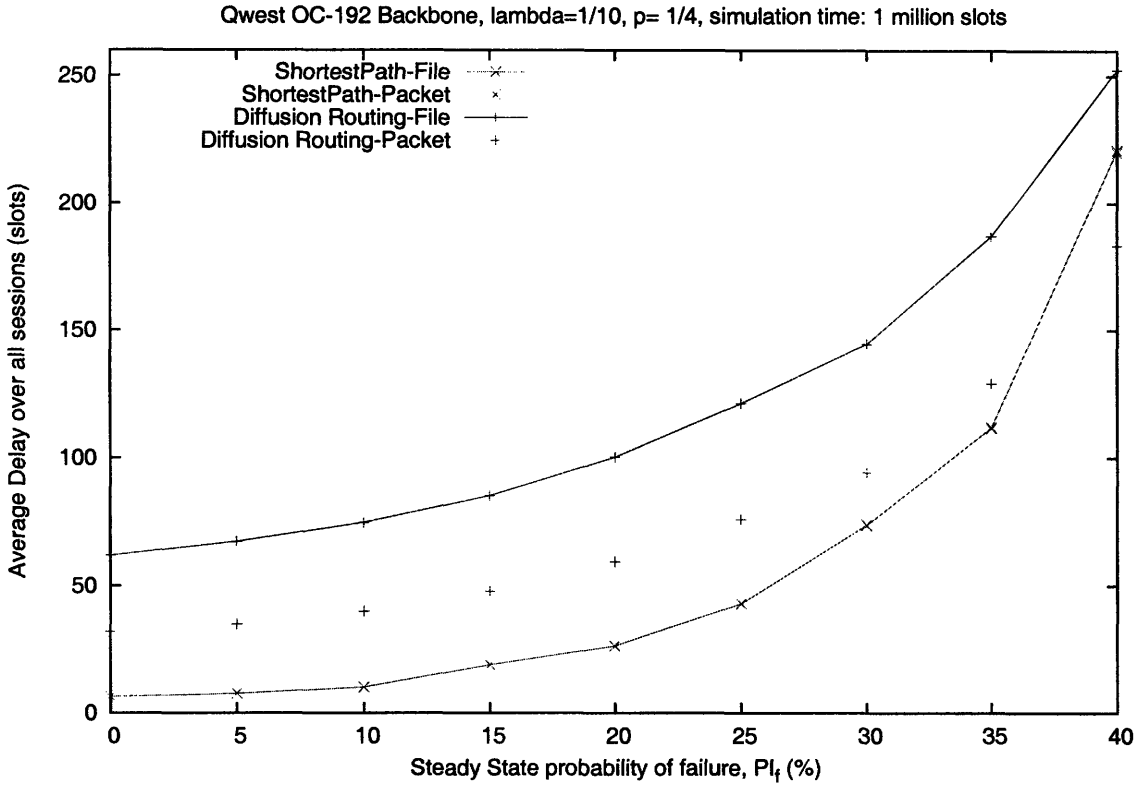

Figure 3-20: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in failure rate in Qwest OC-192 Backbone at low network loading

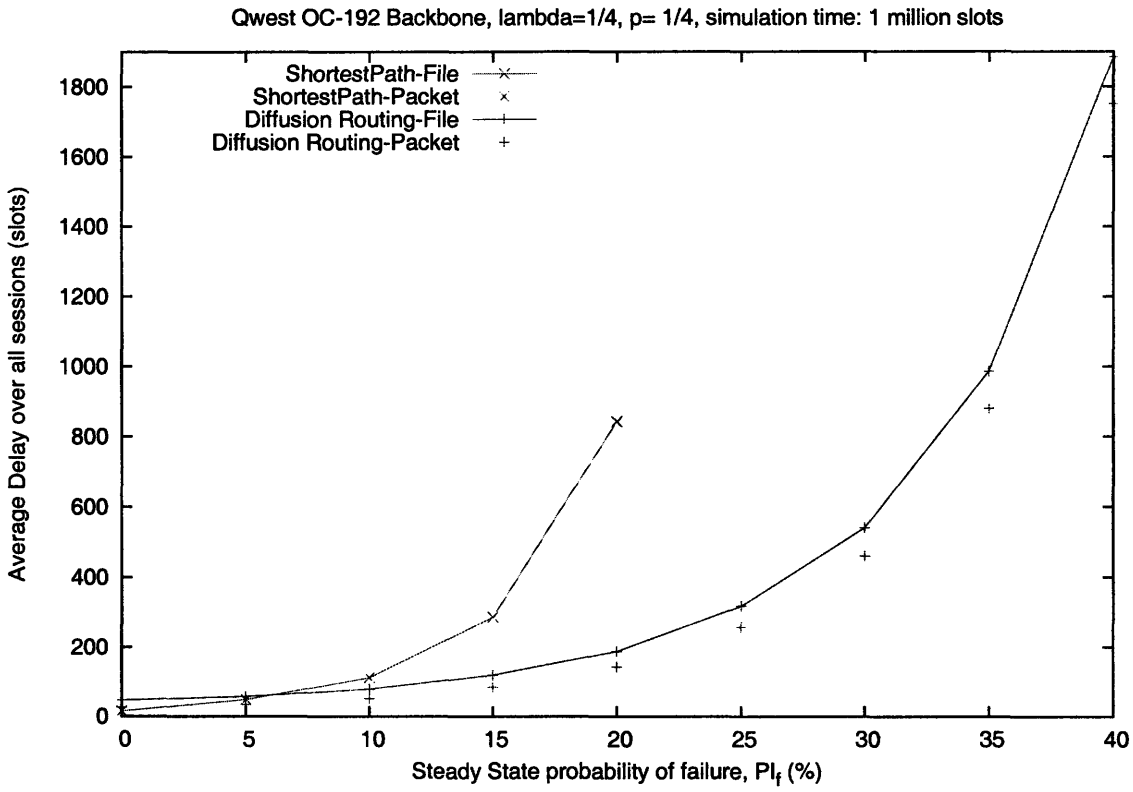

Figure 3-21: File and packet delays of Shortest Path routing as compared to Differential Backlog routing under variation in failure rate in Qwest OC-192 Backbone at high network loading

the amount of stress a network is subjected to, both in terms of traffic loads and network failures.

3.3.4 Notes

The average end-end packet delays of Shortest Path routing are observed to be experimentally equal to the average end-end file delays.

Chapter 4

Adaptations of Differential Backlog Routing

4.1 Differential Backlog Routing augmented with Shortest Path

As we have observed, shortest path algorithm achieves better delay performance whereas Differential Backlog routing has a larger capacity region. Shortest path routes packets towards their destinations while Differential Backlog routing tries to keep network queues for different destinations balanced. A hybrid approach that combines principles that the two algorithms operate on might also buy benefits of both and seems worth while to explore.

4.1.1 Algorithm Description

Neely suggests a similar extension to the one proposed in his doctorate thesis [28]. However, his formulation is generalized to take power control and differentiated QoS into account. We qualify the utility function in (3.1) for maximization to include shortest path costs between two nodes in addition to queue lengths: Let $U_a^{(c)}(t)$ be the number of packets waiting at node a destined for node c at time t. For each pair of directly connected nodes, let's say a and b, the commodity $c_{ab}^*(t)$ to be chosen is

specified as:

$$c^*_{ab}(t) = \arg \max_{c \in 1,\ldots,N} \{U^{(c)}_a(t) - U^{(c)}_b + \alpha(V^c_a(t) - V^c_b(t))\} \tag{4.1}$$

where V^c_i denotes the path distance between node i and node c and α is a scalar constant. The maximum difference between shortest path costs of neighbors is equal to one assuming all links have unit cost. Therefore, we introduced the constant α to make shortest path matter more in situations where average queue size is large.

We refer to this extension as Hybrid Differential Backlog (HybridDB) as this approach combines Differential Backlog routing with Shortest Path.

4.1.2 Implementation Details

HybridDB is implemented in the same way as Differential Backlog routing with the addition of cost-tables at each node which store the shortest path costs of all the other nodes in the network. These costs are needed to calculate the modified utility function described in section 4.1.1.

4.1.3 Results

Delay response to variations in Network Loading

10-node 4-connected symmetric topology Fig. 4-1 compares HybridDB performance with Differential Backlog routing and Shortest Path routing for low realtively low network loadings. HybridDB performs far better than than Differential Backlog routing and appears not to have any inclination of instability. One phenomenon that is truly remarkable is the edge of HybridDB over even Shortest Path routing for the operating regimes shown. Although it does appear the Shortest Path routing might eventually outperform HybridDB given faster decrease in its delays with decreased loading. However, still if the capacity of Shortest Path routing is deemed to be at the loading level of 0.3 file arrivals for an average file size of 4 packets, we have at our hands an algorithm that can perform better than Shortest Path routing for more

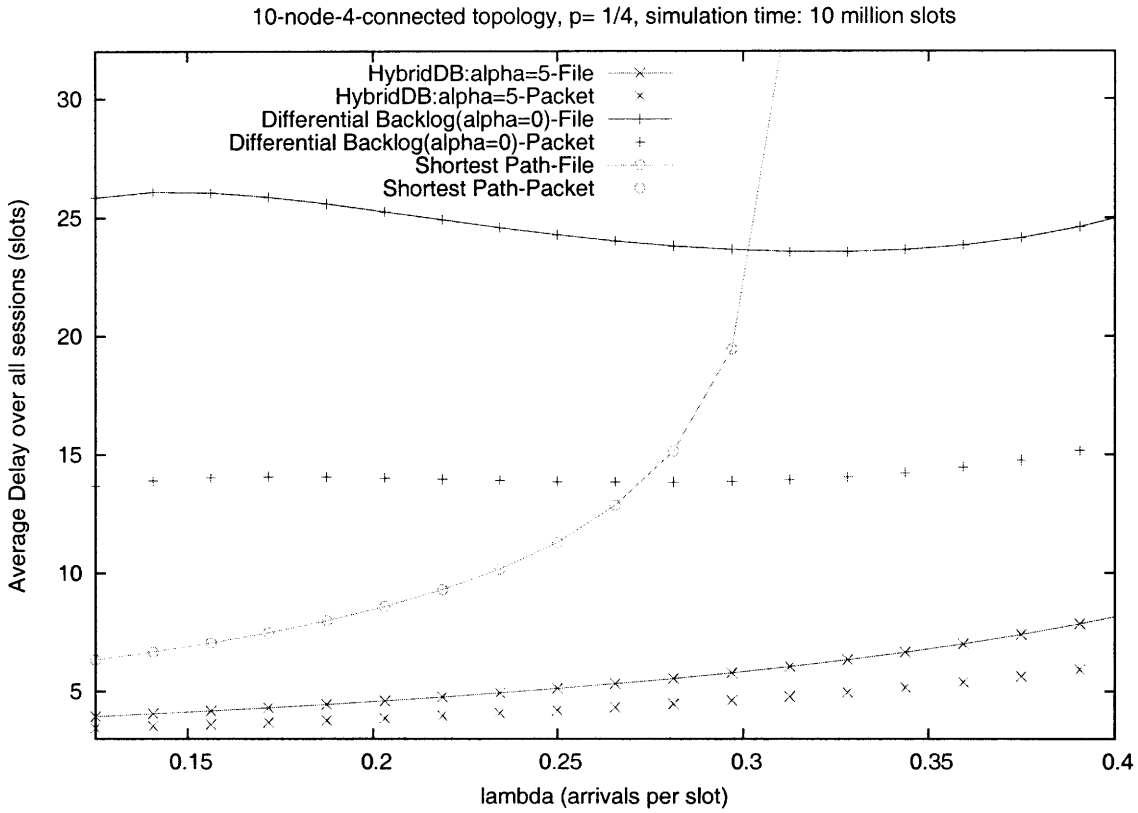

Figure 4-1: Delay performance of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in network loading for the 10-node 4-connected symmetric topology at low network loads

than half shortest path's capacity region. Fig. 4-2 focuses on higher network loads and enables us to compare the capacity regions of each routing algorithm. As can be seen, HybridDB has more or less the same capacity region as that of Differential Backlog routing. In a sense, HybridDB does achieve the best of both worlds by exhibiting superior delay performance which beats even Shortest Path routing and capacity that rivals that of Differential Backlog routing.

Qwest OC-192 Backbone Fig. 4-3 plots average file and packet delays for Qwest OC-192 Backbone. One sees the same trends as observed for the case of the 10-node 4-connected symmetric topology. Here, HybridDB again shows attributes of outpeforming both Differential Backlog routing and Shortest Path routing in capacity and delay performance. A subtle trend that might get overshadowed by the its superior routing performance is that HybirdDB does not seem to suffer from increased delays

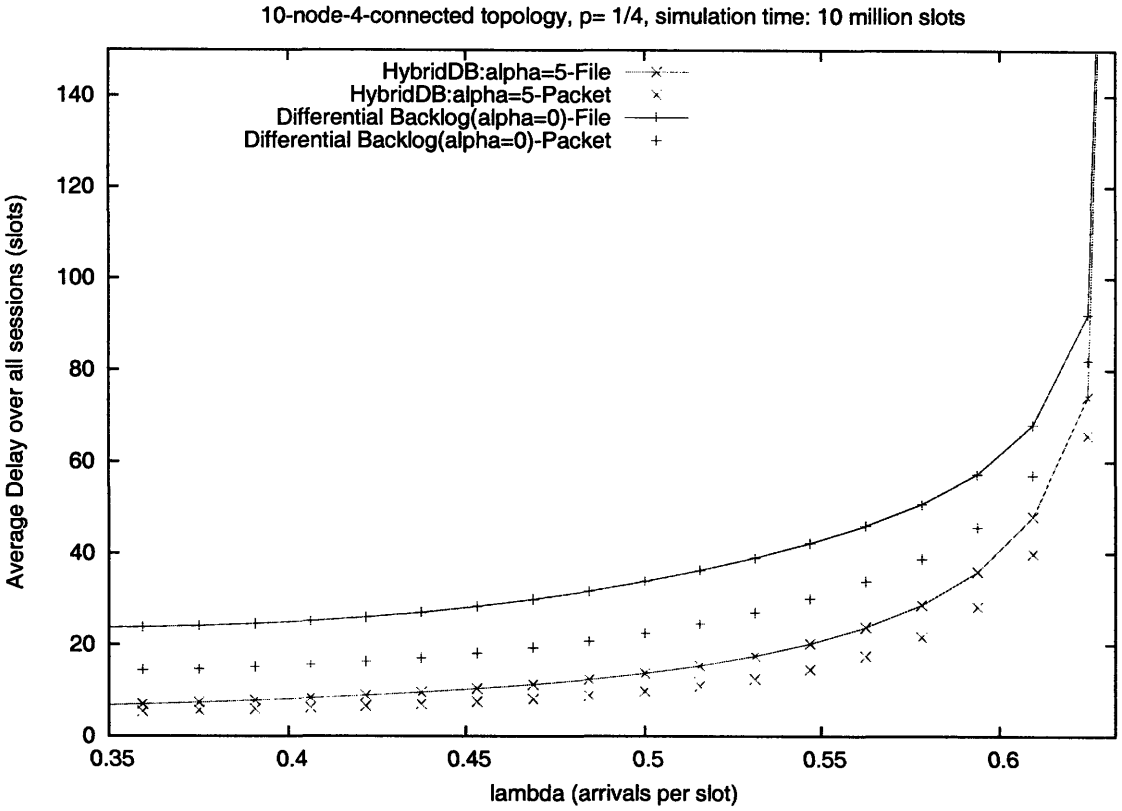

Figure 4-2: Delay performance of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in network loading for the 10-node 4-connected symmetric topology at high network loads

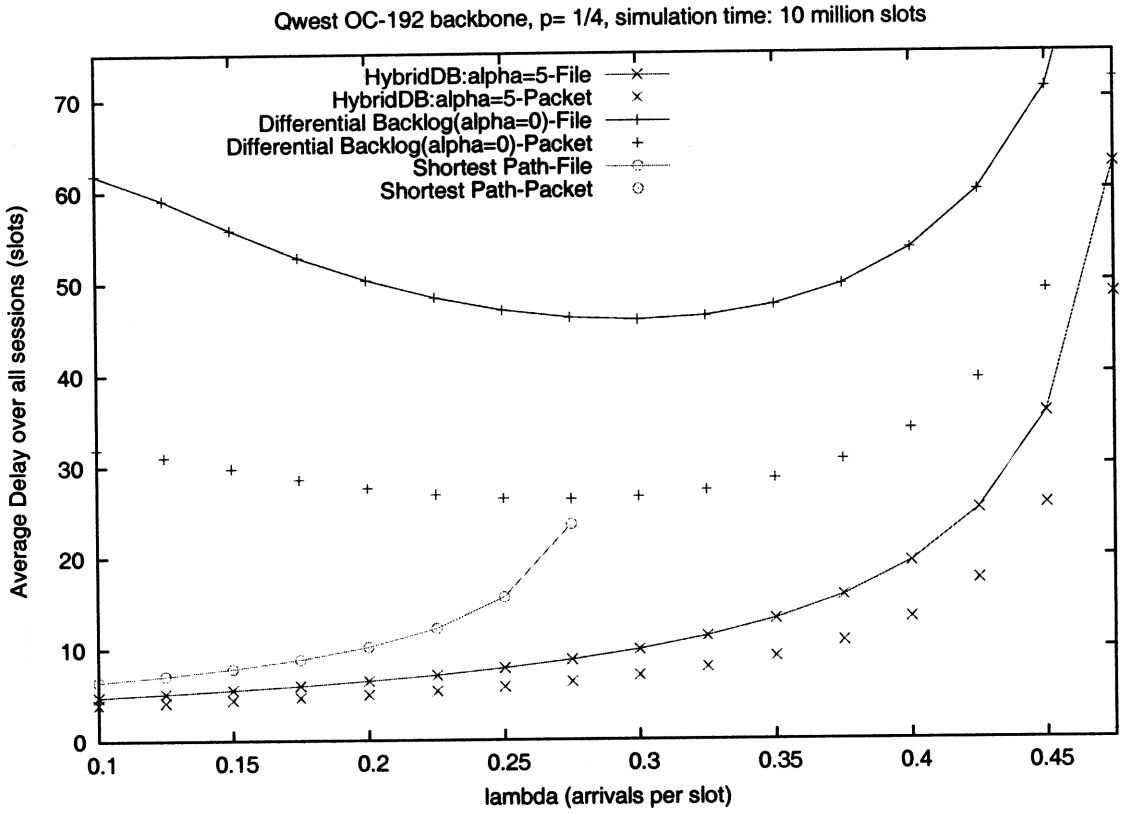

Figure 4-3: Delay performance of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in network loading for Qwest OC-192 Backbone at low network loads

for lower loads which was one of the prloblems which HybridDB was sought to solve.

Delay response to variations in Failure Rate, p_f

10-node 4-connected symmetric topology We subject HybridDB to increasing failures and analyze its performance. The file and packet delay results presented in Fig. 4-4 are not as stellar as previously observed with respect to increasing network loading but in the absence of failures. HybridDB behaves worse than Shortest Path routing for most levels of failures. There are exceptions as HybridDB delay profile drops more steeply than that of Shortest Path routing and shows slower growth than Shortest Path routing. Overall, HybridDB seems to perform better than Shortest Path routing either at very low failure rates due to its delays or very high rates due to its increased capacity. The above behavior makes sense because at lower loads,

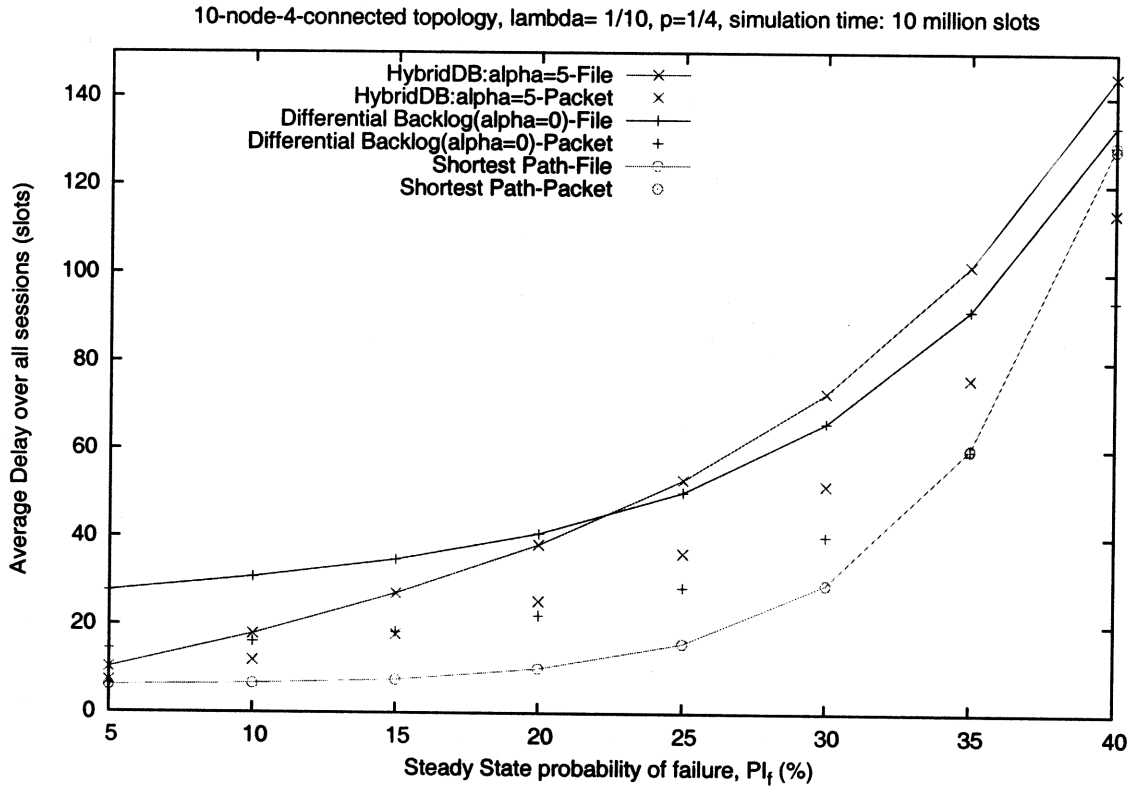

Figure 4-4: File and packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the 10-node 4-connected symmetric topology

HybridDB effectively operates as Shortest Path routing but can also use mutliple paths to route packets towards their destinations, making the average delays lower. At high failures, HybridDB starts to act more and more like a pure form of Differential Backlog routing. This is because queue occupancies grow significantly at higher loadings and/or failure rates and minimize shortest path bias. As far as comparison with Differential Backlog routing is concerned, HybridDB's effectiveness remains questionable. We will take into account more results on HybridDB before making a generalization in this respect.

Average packet delays are plotted in Fig. 4-5 and do not show any new trends as they follow file delays closely but are always smaller by a small margin.

We repeat the experiment for a higher value of loading as we have often done in the past and results are presented in Fig. 4-6. As one can see, HybridDB is shown

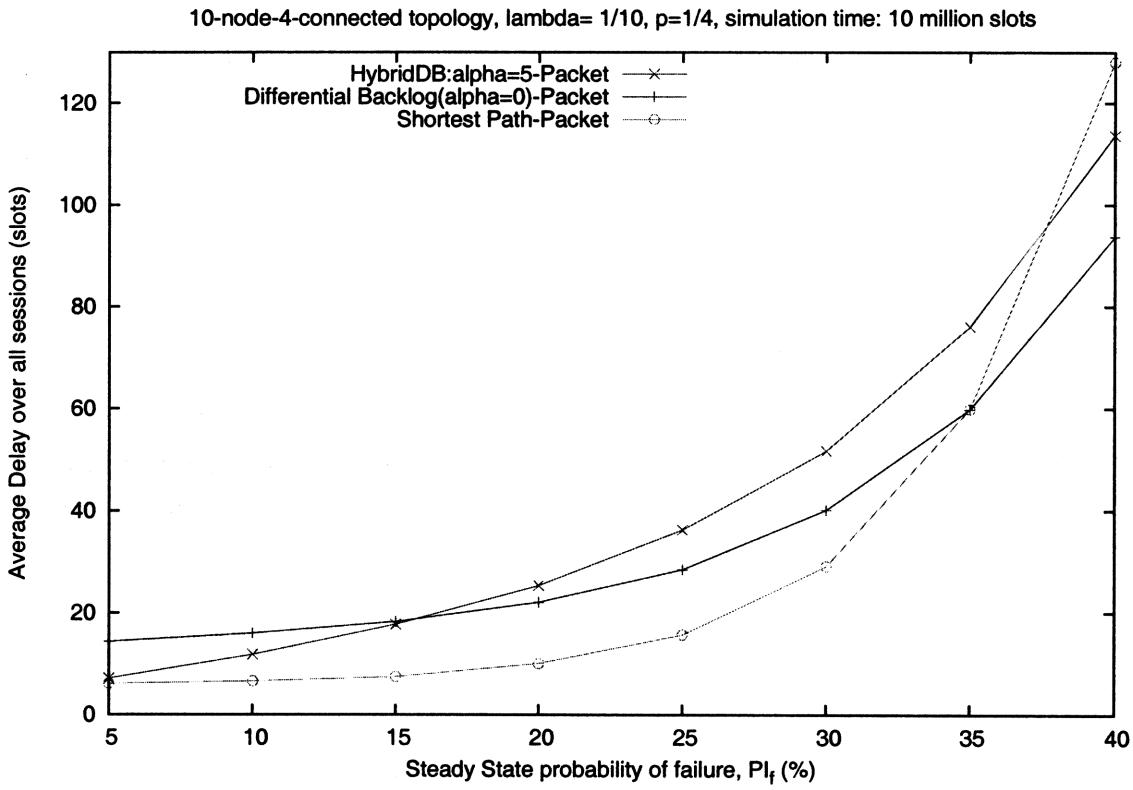

Figure 4-5: Packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the 10-node 4-connected symmetric topology

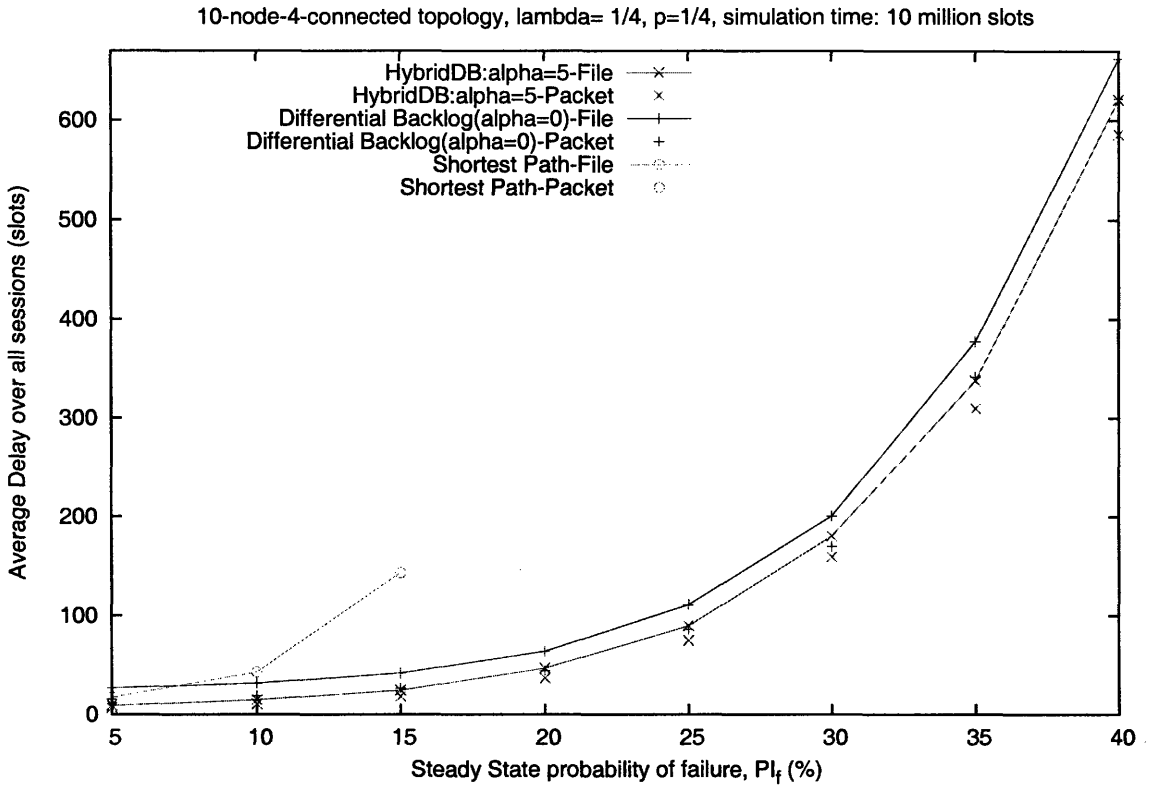

Figure 4-6: File and packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the 10-node 4-connected symmetric topology

to outperform both Shortest Path routing and Differential Backlog routing. The behavior is hard to explain but we can characterize it as the tendency of HybridDB to perform relatively better in comparison to Shortest Path routing and Differential Backlog routing at higher loads if failures are kept constant. Once again, we plot average packet delays under different routing paradigms separately in Fig. 4-7 for better visualization but they their analysis does not offer any new insights or trends.

Qwest OC-192 Backbone The results for the same traffic situations and algorithm parameters as those used for 10-node 4-connected symmetric topology are shown in Figs. 4-8, 4-9, 4-10 and 4-11 for the network setting of Qwest OC-192 Backbone. They enforce the trends observed and explained for the 10-node 4-connected symmetric topology earlier.

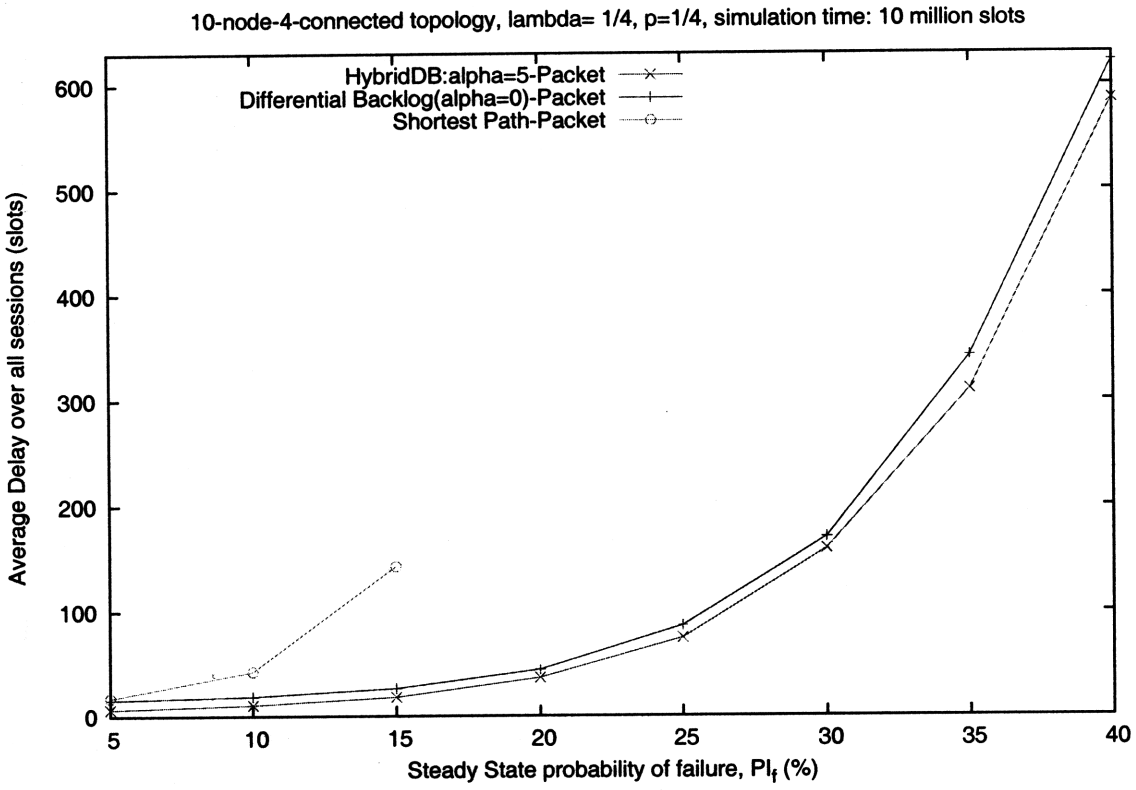

Figure 4-7: Packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the 10-node 4-connected symmetric topology

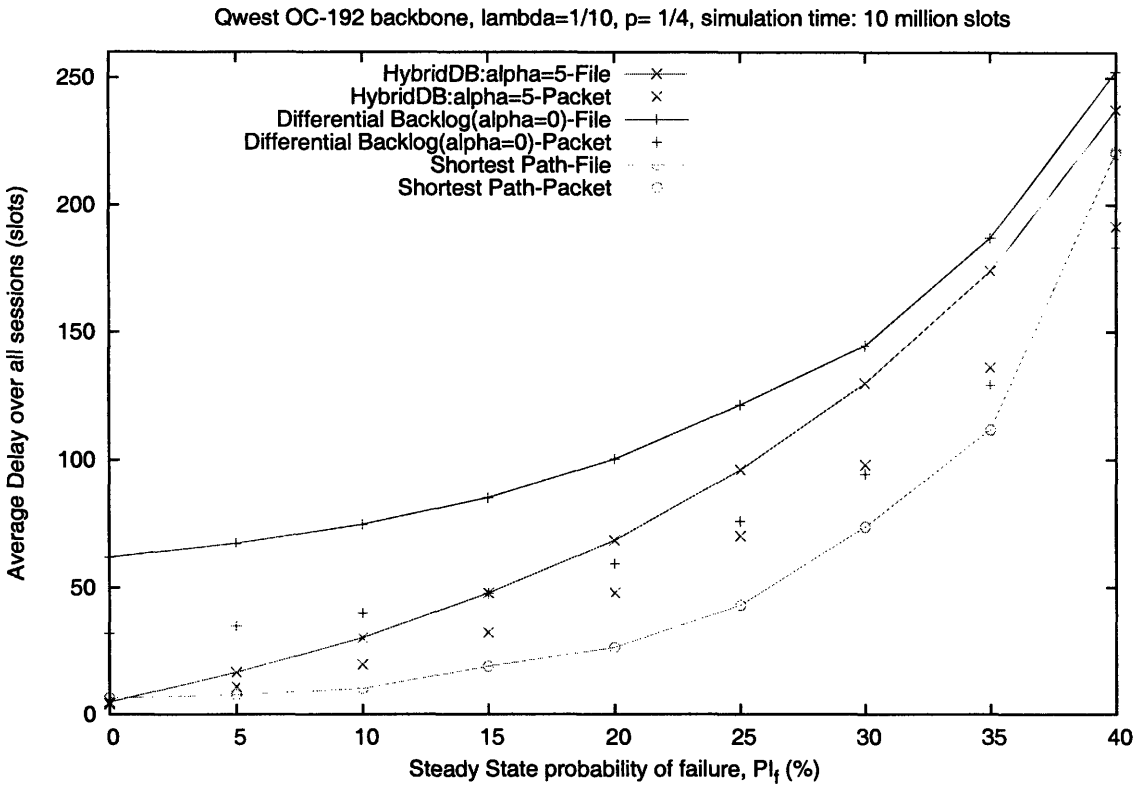

Figure 4-8: File and packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the Qwest OC-192 Backbone

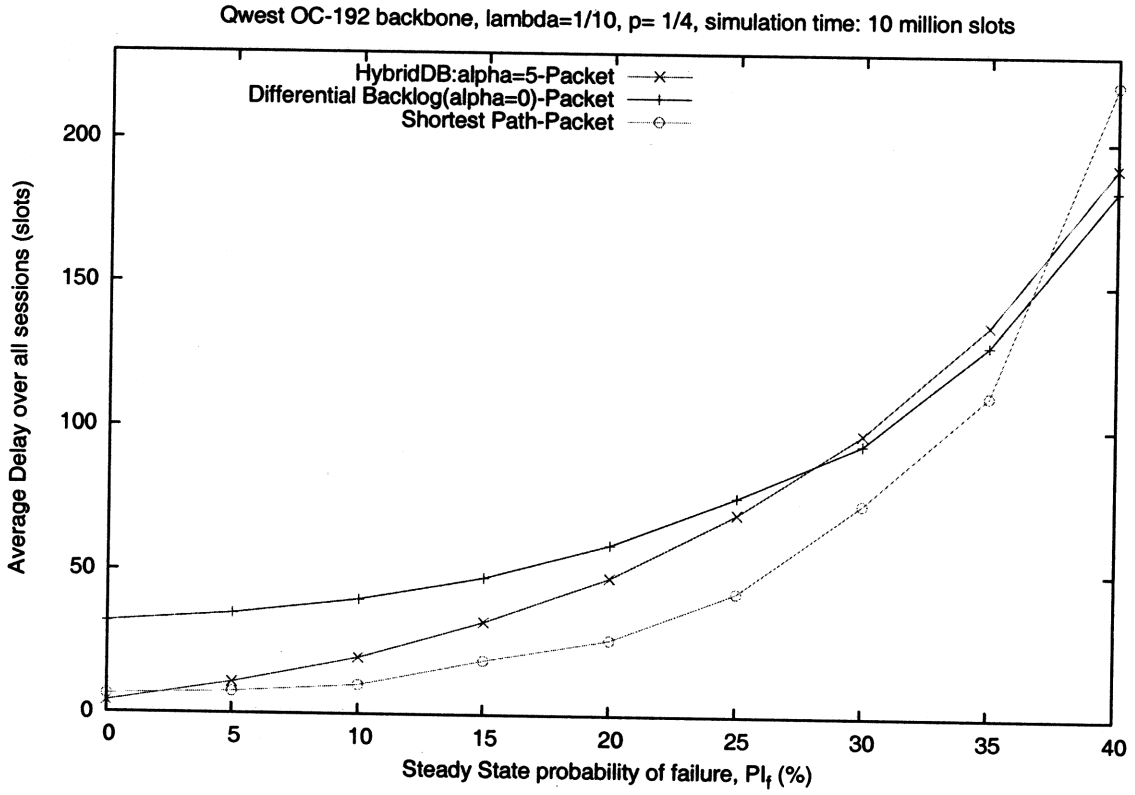

Figure 4-9: Packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the Qwest OC-192 Backbone

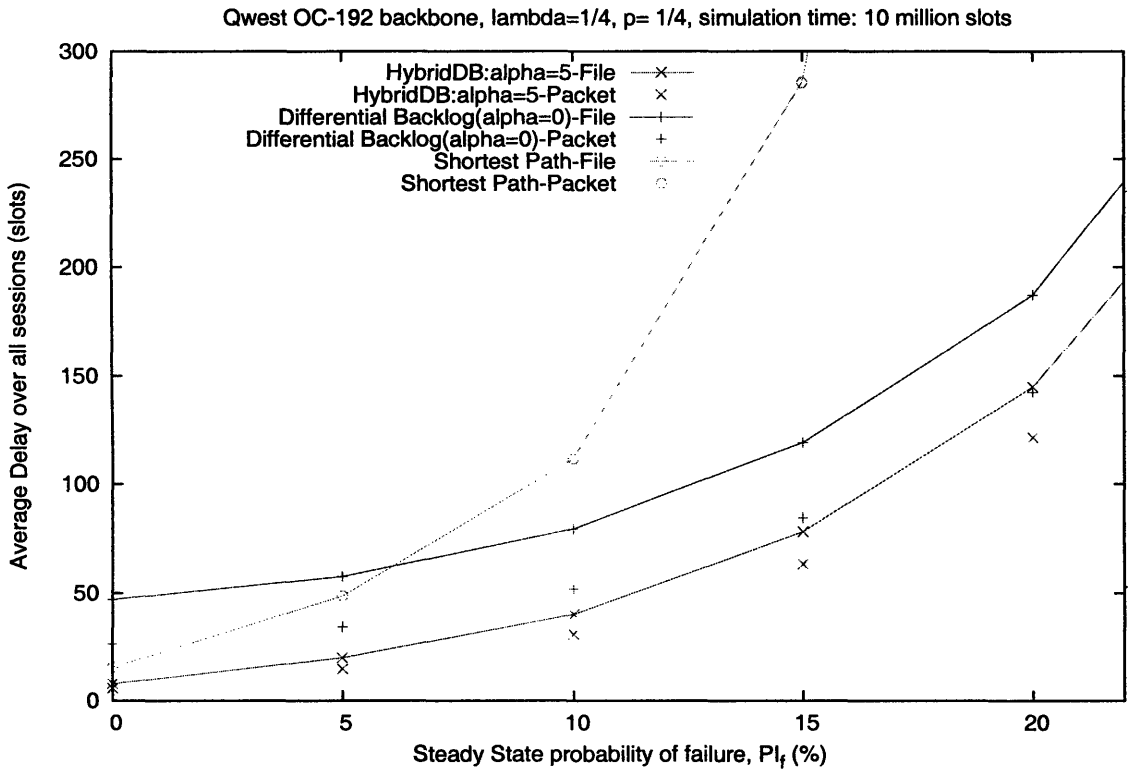

Figure 4-10: File and packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the Qwest OC-192 Backbone at low failure rates

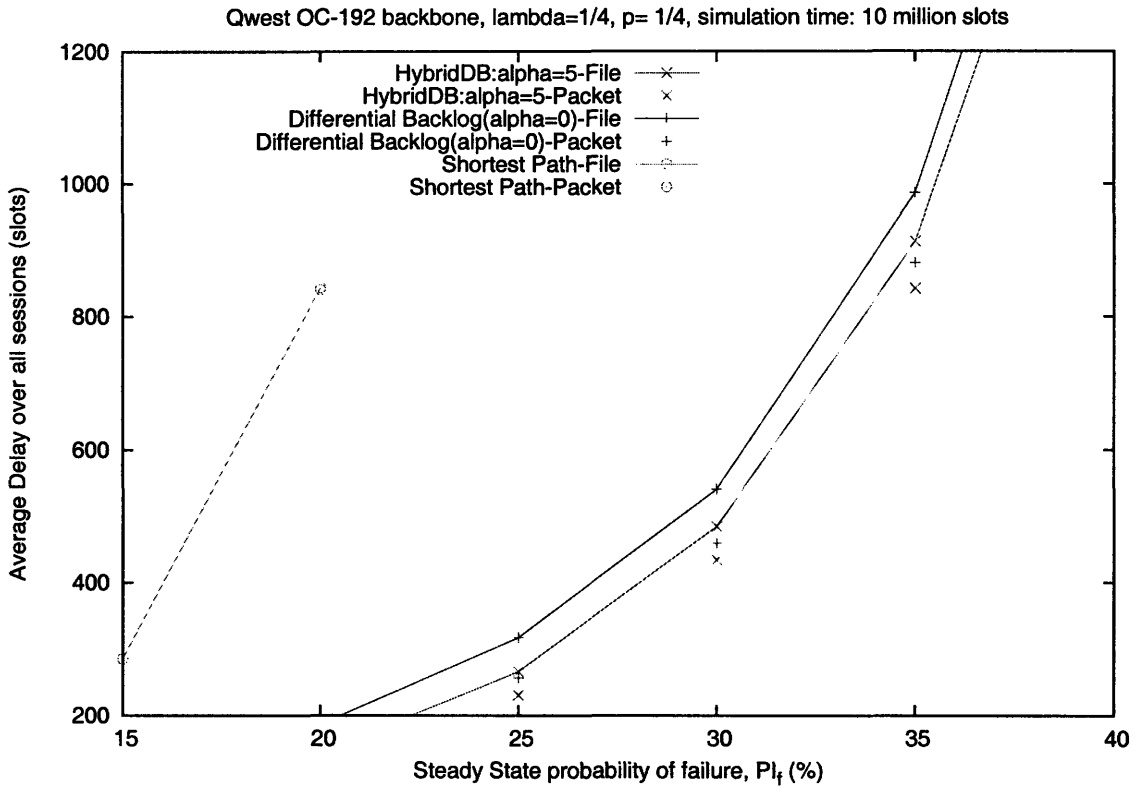

Figure 4-11: File and packet delays of HybridDB as compared to Differential Backlog routing and Shortest Path routing under variation in failure rates for the Qwest OC-192 Backbone at high failure rates

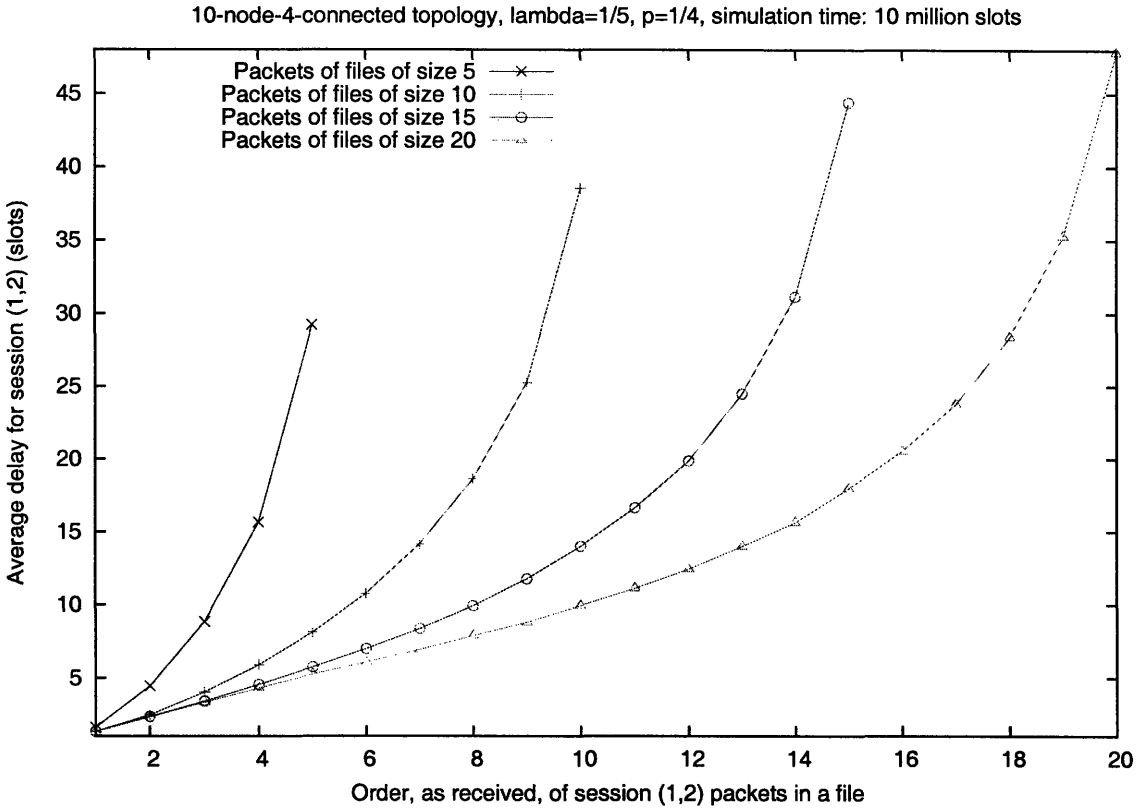

Figure 4-12: Average successive packet delays for different file sizes for 10-node-4-connected topology, $\lambda = 1/5, p = 1/4$

4.2 Digital Fountains

In a bid to investigate the delays experienced by individual packets in a file, we tracked the time it took for packets to be received at a destination regardless of their order in a file. Fig 4-12 plots the average delay for packets for session (1,2) in order of their arrival at the destination for different file sizes. The successive packet delays tend to grow exponentially. If the destination node does not have to wait for the last few packets in a file, the file delays can be improved significantly. For example, for a file of size 20, discounting the receipt of last two packets (10% loss) the average delay can be decreased by an average of approximately 20 time slots (40 % improvement in delay). This realization led us to the prospect of employing the paradigm of *digital fountains* to achieve better delay performance for Differential Backlog routing.

Mitzenmacher et al. introduced the idea of *digital fountains* [7] as an alternative to strictly ordered packet transfer as in TCP. The idea is to stretch information contained

in k source packets into $n \geq k$ encoding packets. The original information can be reconstructed by decoding *any* k of the encoded packets. [25] identifies several coding schemes, namely Reed-Solomon codes, Tornado codes, LT codes, Raptor codes, to construct a digital fountain. In Differential Backlog routing, packets in a file may arrive out of order because of the different paths they may use. Using a digital fountain approach, not only the order of arrival of packets in a file at the destination node becomes irrelevant but one can also improve the end-end file delays.

4.2.1 Digital Fountain Model

Files arrive according the Arrivals Model described in subsubsection 3.1.1. File sizes, however, have been fixed at x packets, so that evaluation of digital fountains is least affected by variations in file sizes. For a file of x packets, $\lceil \frac{x}{f} \rceil$ packets are generated where f is the coding rate. At the destination node, a file arrival is considered to be complete when any x of the $\lceil \frac{x}{f} \rceil$ packets originally transmitted by source node have been received.

4.2.2 Implementation Details

Nodes keep track of the number of packets they have received for each file destined towards them. The order in which packets are received does not matter. Since there is no packet loss, the redundant packets in a file do get to the nodes eventually whereby they are discarded. A file is termed *active* if its destination has not received all the packets generated for that file - including redundant packets for digital fountain. A list of all its *active* files, is maintained at each destination node.

4.2.3 Results

Differential Backlog routing and Digital Fountains

Delay response to variations in Network Loading In a bid to fully evaluate the usefulness of digital fountains, we decided not to keep network load constant when

code rate changes. As a result, as code rate decreases, network loading increases. Hence, increased network load has been considered as part of the problem that digital foutains were sought to treat.

10-node 4-connected symmetric topology The delay performance results - with and without failures - are presented in Fig. 4-13. One can observe that average packet delays always increase as coding rate decreases (and hence network load increases) because of increased congestion in the network. On the other hand, digital fountains show promise when it comes to average file delays. Average file delays decrease first and after reaching a minimum, start increasing. The minimum occurs at a coding rate which we call *optimal* for a specific file size, x, and network load. The concave average file delay profile results from the tradeoff between not having to wait for last packets in a file which arrive with exponentially larger delays and increased average packet delays because of network loading.

Qwest OC-192 Backbone Next, we investigate the supplemental use of using digitial fountains in a network with varying probabilities of failure operating at or close to the *optimal* value of code rate.

Delay response to variations in Failure Rate, p_f

10-node 4-connected symmetric topology The results which have been presented in Fig. 4-15 show again the interesting behavior where digital fountains result in higher average packet delays but lower average file delays. The increase in average packet delays is once again explained through increased congestion in the network due to extra coding packets. Since file delays are the ones of most practical interest, we conclude that digital fountains can effectively be used to supplement the performance of Differential Backlog routing.

Qwest OC-192 Backbone Fig. 4-16 plots delays for Qwest OC-192 Backbone which shows similar behavior as in the case of 10-node 4-connected symmetric topol-

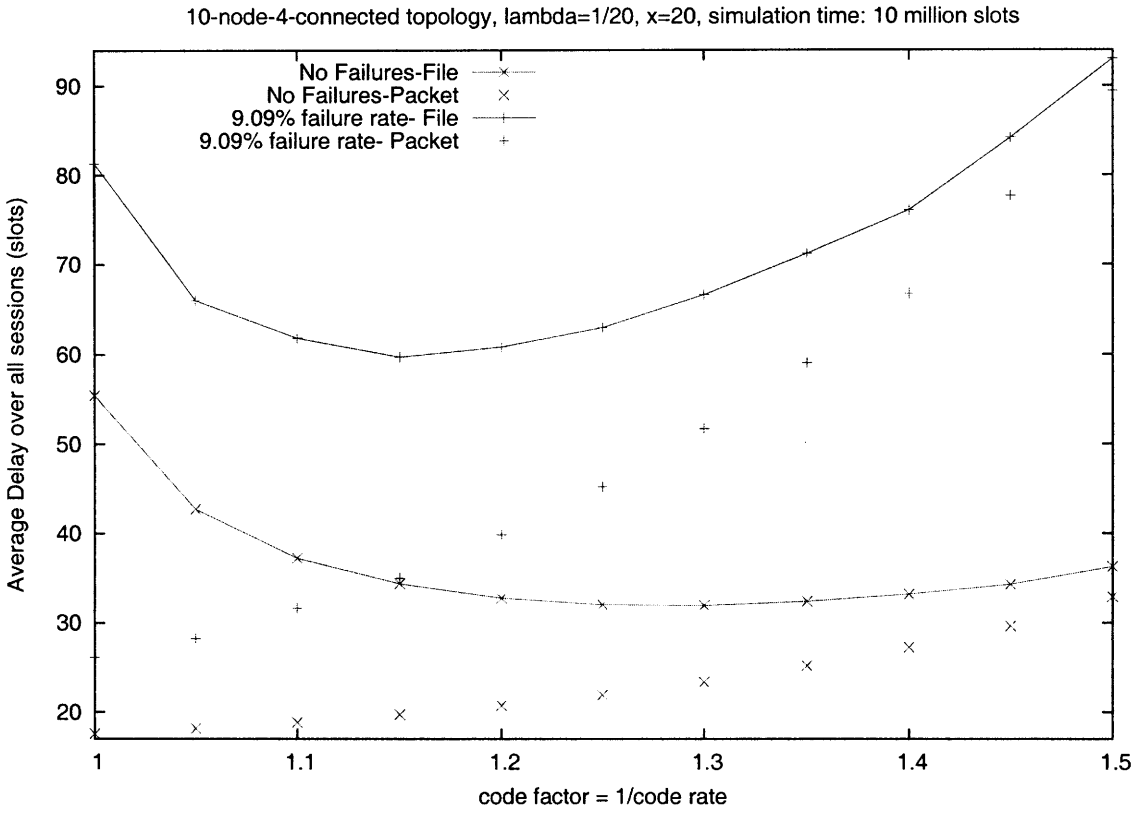

Figure 4-13: Delay performance of Digital Fountain approach in Differential Backlog routing as a function of code rate for 10-node 4-connected symmetric topology

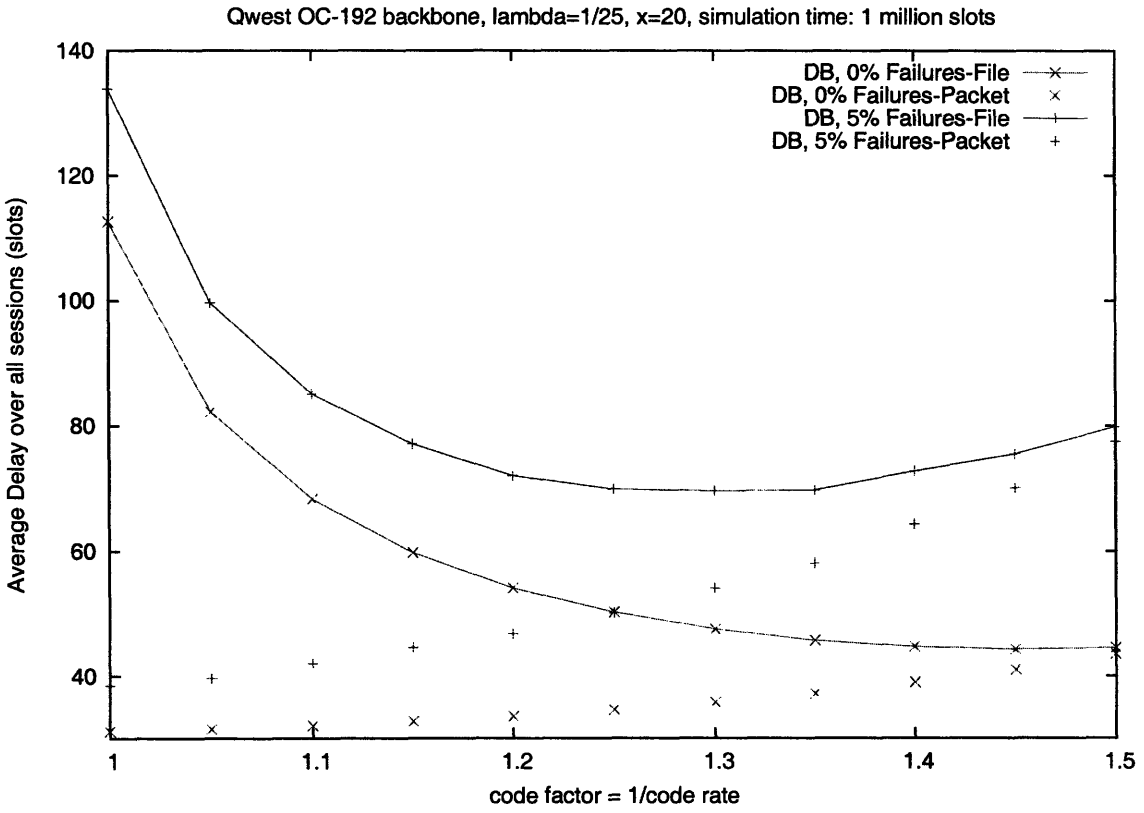

Figure 4-14: Delay performance of Digital Fountain approach in Differential Backlog routing as a function of code rate for Qwest OC-192 Backbone

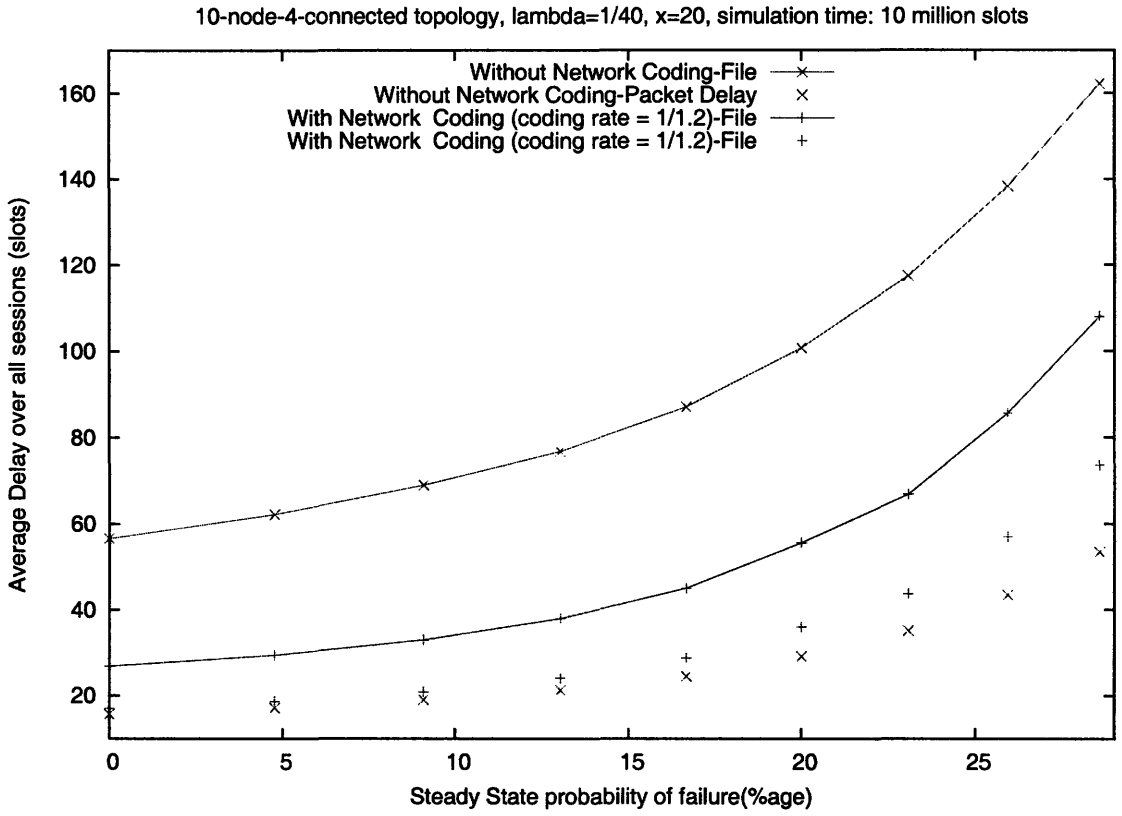

Figure 4-15: Delay performance of Digital Fountain approach in Differential Backlog routing with failure rate for 10-node 4-connected symmetric topology

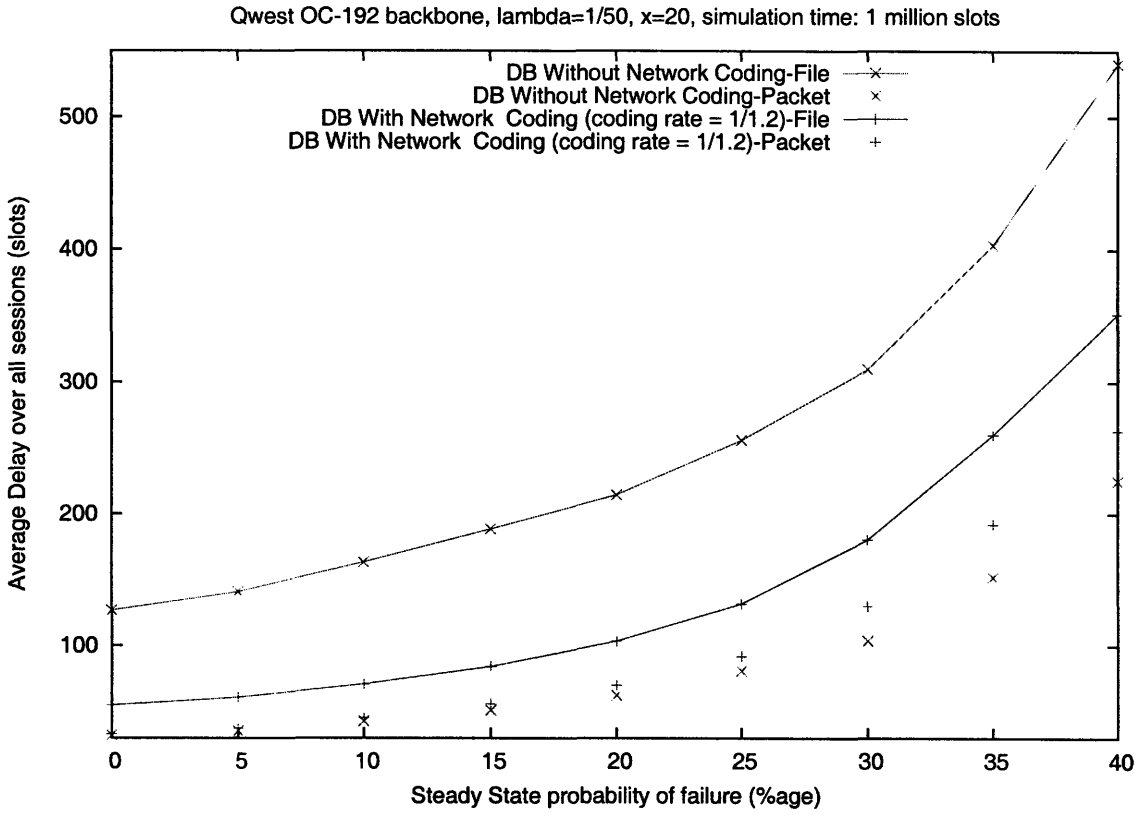

Figure 4-16: Delay performance of Digital Fountain approach in Differential Backlog routing with failure rate for Qwest OC-192 Backbone

ogy.

HybridDB and Digital Fountains

We extend the digital fountains paradigm to HybridDB to investigate whether the gains in delay observed by using digital fountains in Differential Backlog routing also carry over to HybridDB.

Delay response to variations in Network Loading

10-node 4-connected symmetric topology Indeed, we observe similar gains in delays observed in HybridDB as in DB through the use of digital fountains as shown in Fig. 4-17. Digital fountains not only help improve delays in HybridDB but also help it to mantain its edge over Differential Backlog routing over the range of coding rate used in simulations as shown in Fig. 4-18 which plots and compares delays for

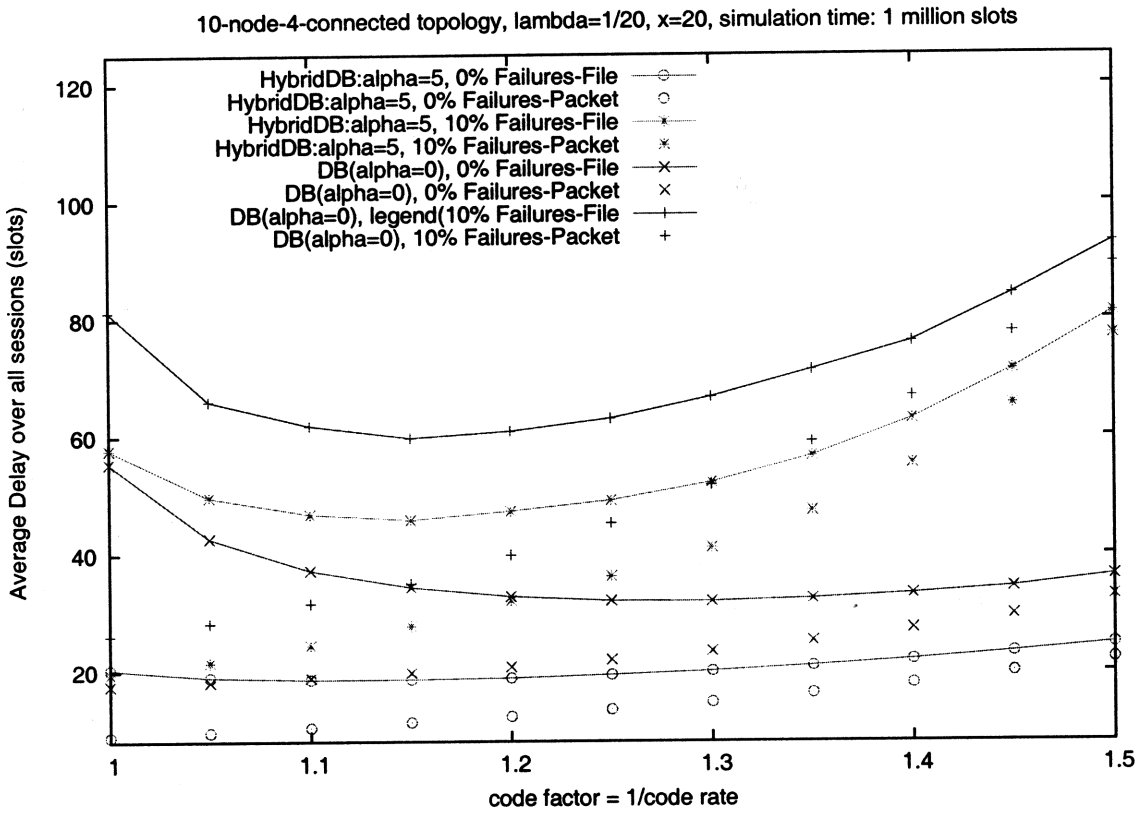

Figure 4-17: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for 10-node 4-connected symmetric topology with and without failures

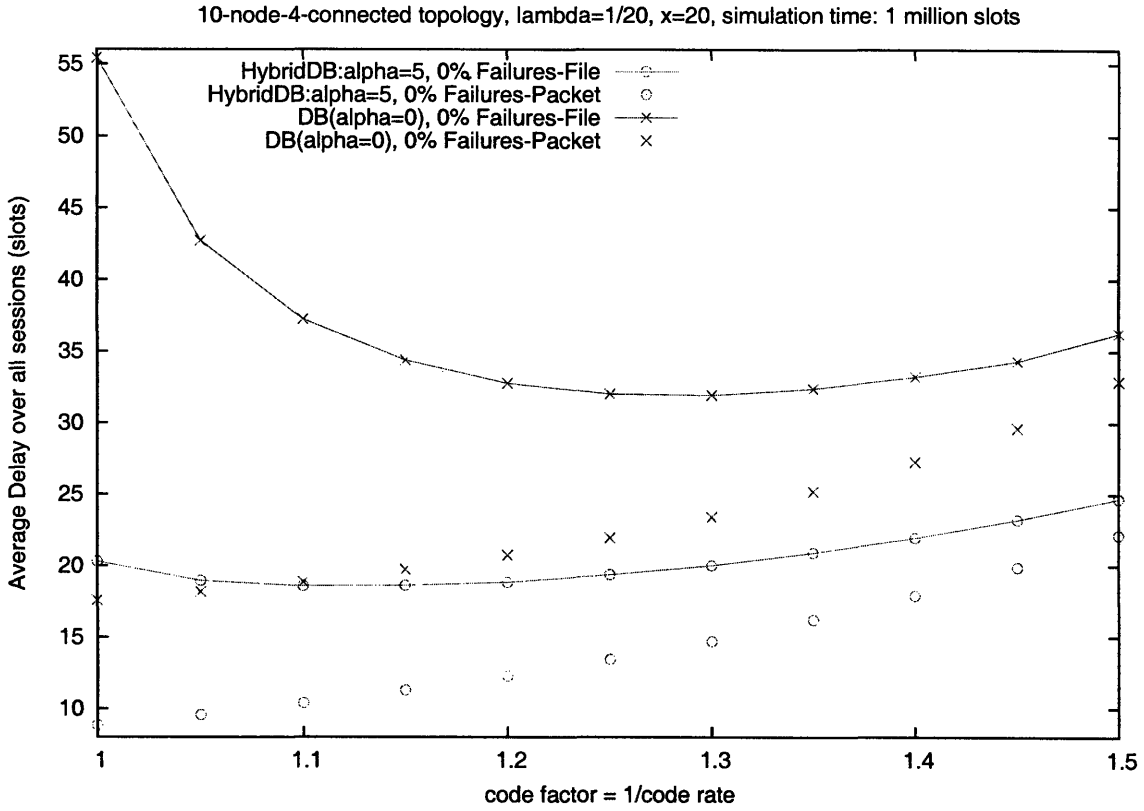

Figure 4-18: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for 10-node 4-connected symmetric topology without failures

HybridDB in networks without failures with Differential Backlog routing and Shortest Path routing. The improvement in delay for HybridDB and its comparatively better performance over DB continues for a range of coding rates is observed to hold for networks with failures as shown in Fig. 4-19. Lastly, we plot average packet delays for all routing protocols introduced so far, namely Differential Backlog routing, Shortest Path routing and HybridDB over a range of values of coding rate, and, with and without failures in Fig. 4-20. As the general trend observed in the case of Differential Backlog routing earlier, average packet delays are also seen to increase with increasing code rate for the case of HybridDB.

Qwest OC-192 Backbone We obtain plots for identical scenarios for Qwest OC-192 Backbone and they are presented in Figs. 4-21, 4-22, 4-23. and 4-24.

Delay response to variations in Failure Rate, p_f

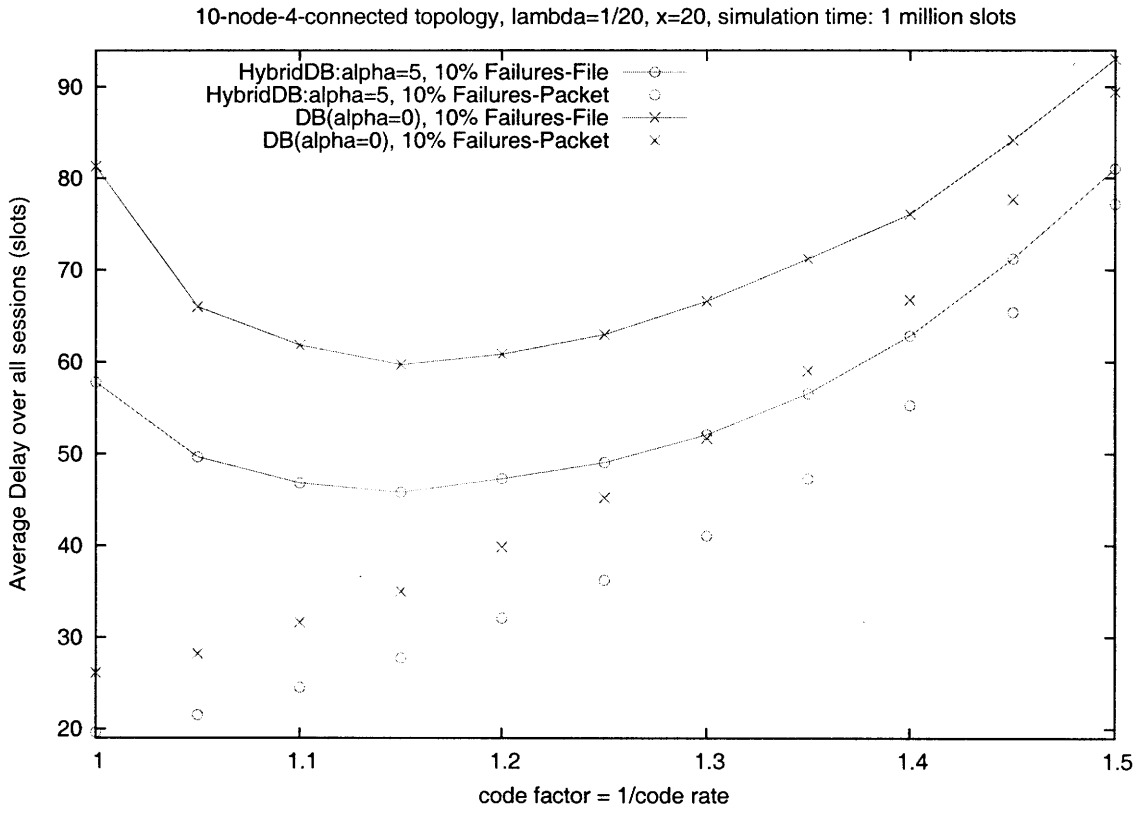

Figure 4-19: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for 10-node 4-connected symmetric topology with failures

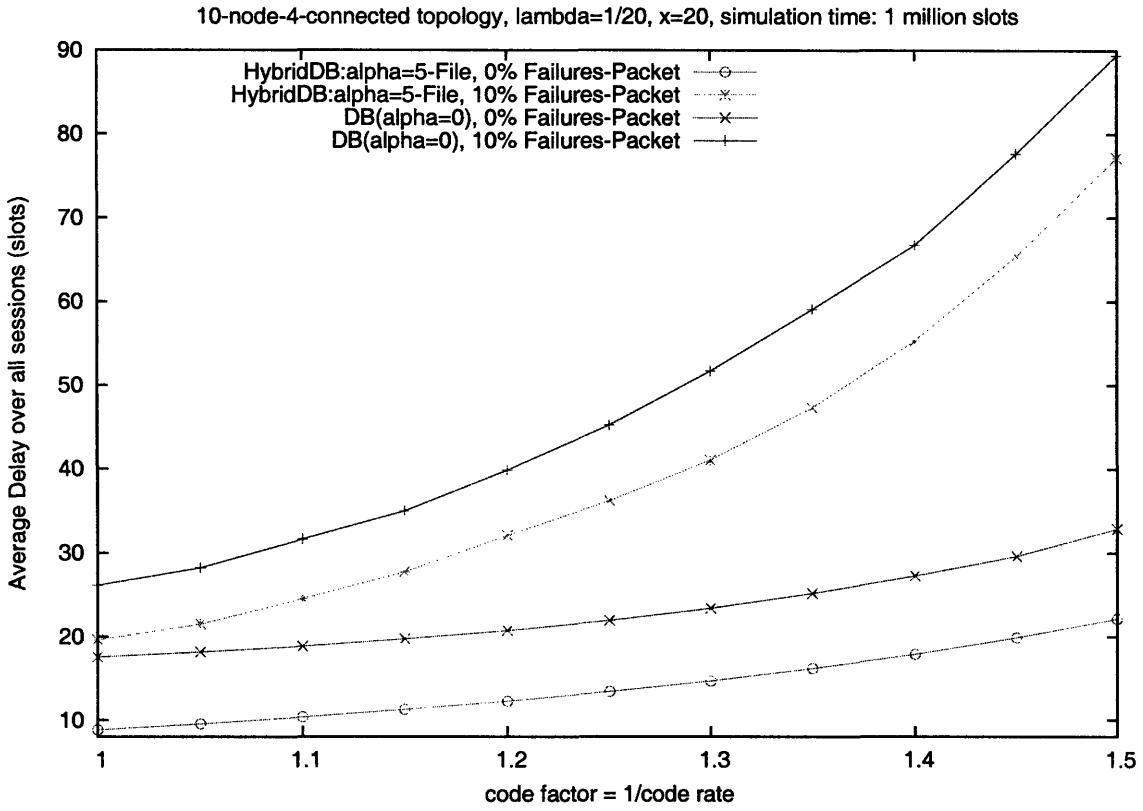

Figure 4-20: Packet delays of Digital Fountain approach in HybridDB as a function of code rate for 10-node 4-connected symmetric topology with and without failures

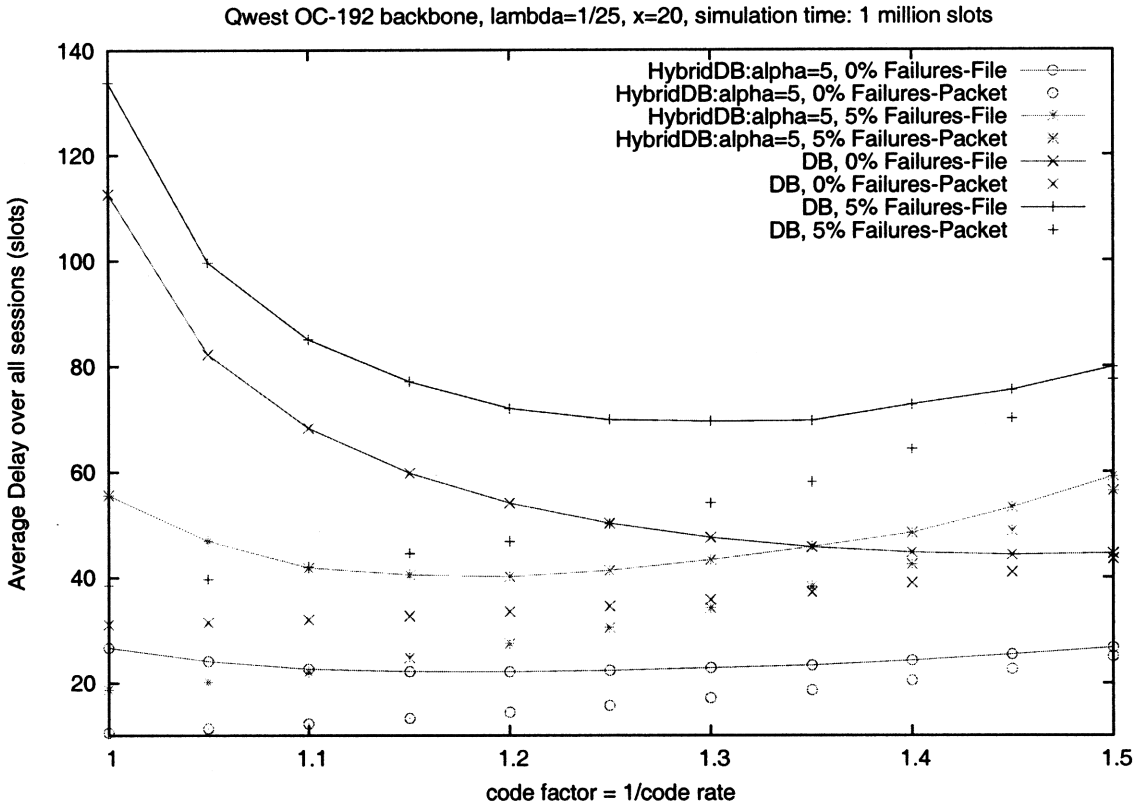

Figure 4-21: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for Qwest OC-192 Backbone with and without failures

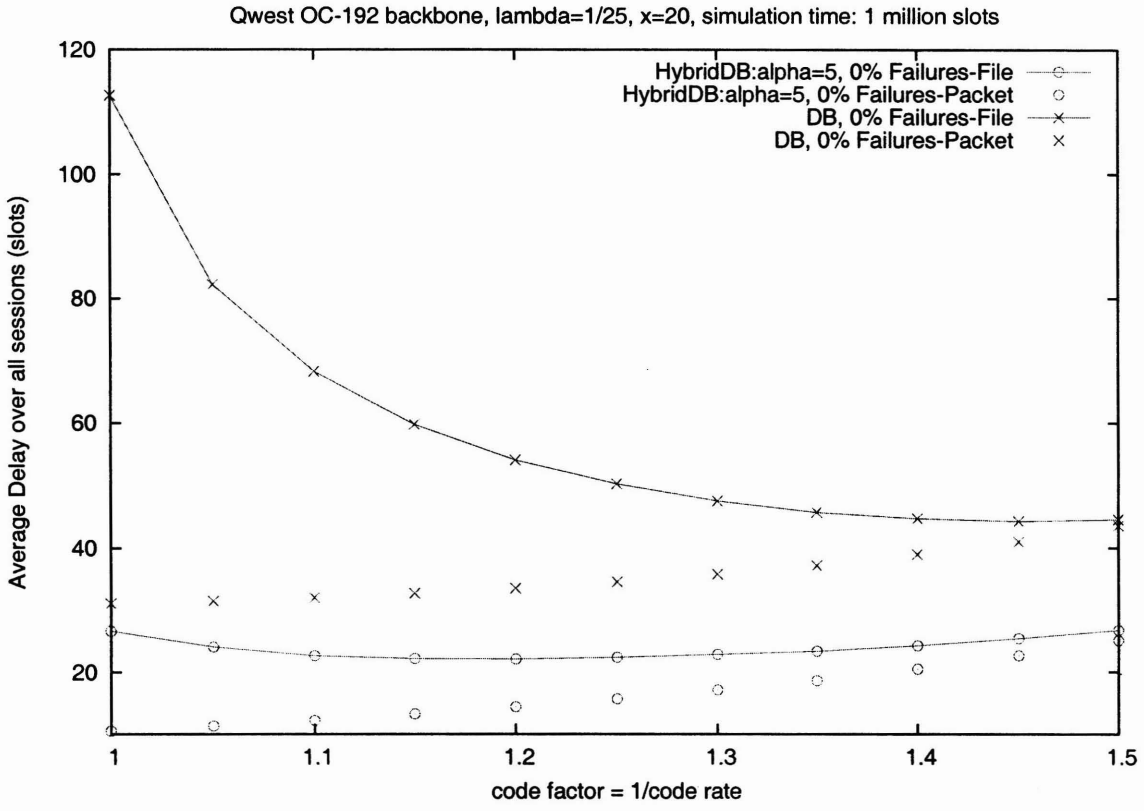

Figure 4-22: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for Qwest OC-192 Backbone without failures

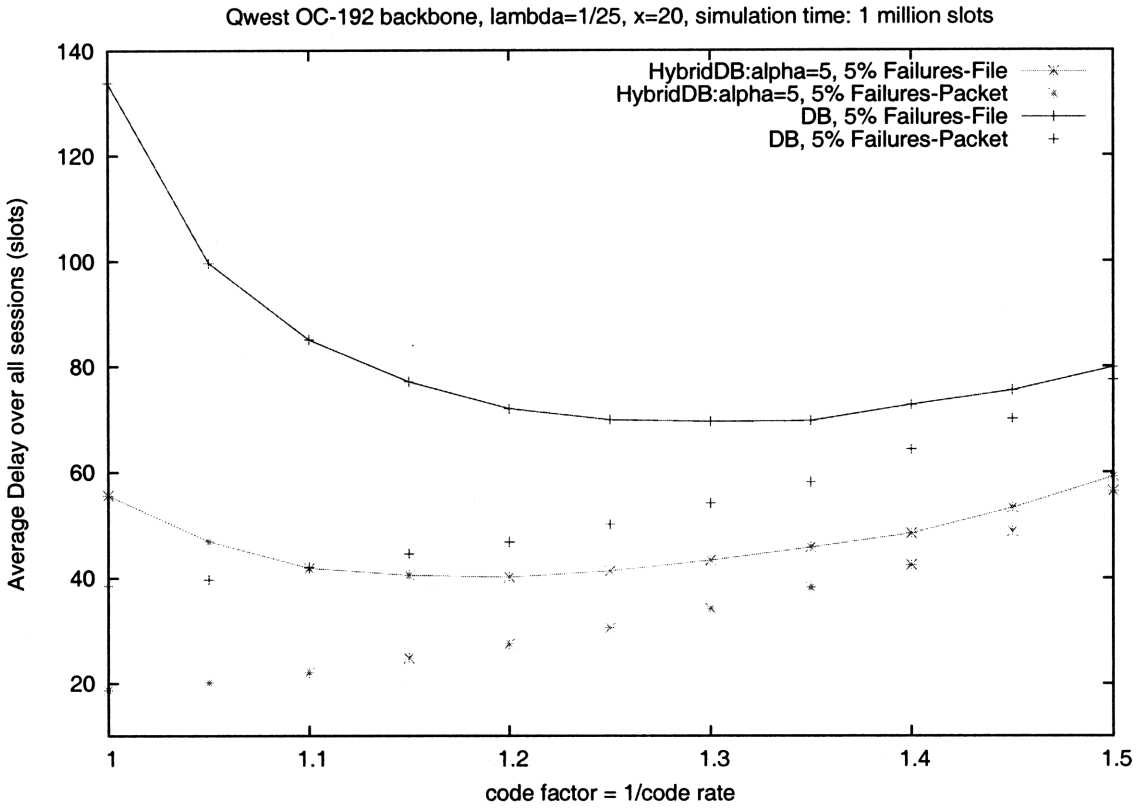

Figure 4-23: Delay performance of Digital Fountain approach in HybridDB as a function of code rate for Qwest OC-192 Backbone with failures

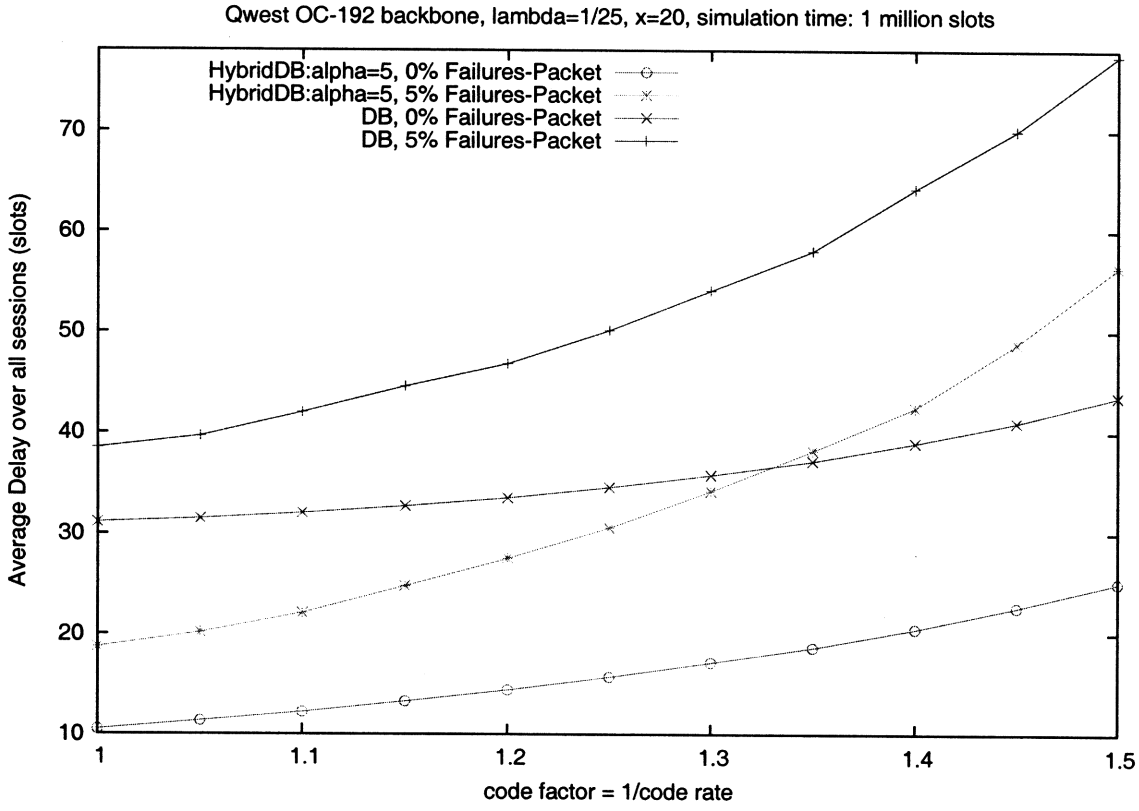

Figure 4-24: Packet delays of Digital Fountain approach in HybridDB as a function of code rate for Qwest OC-192 Backbone with and without failures

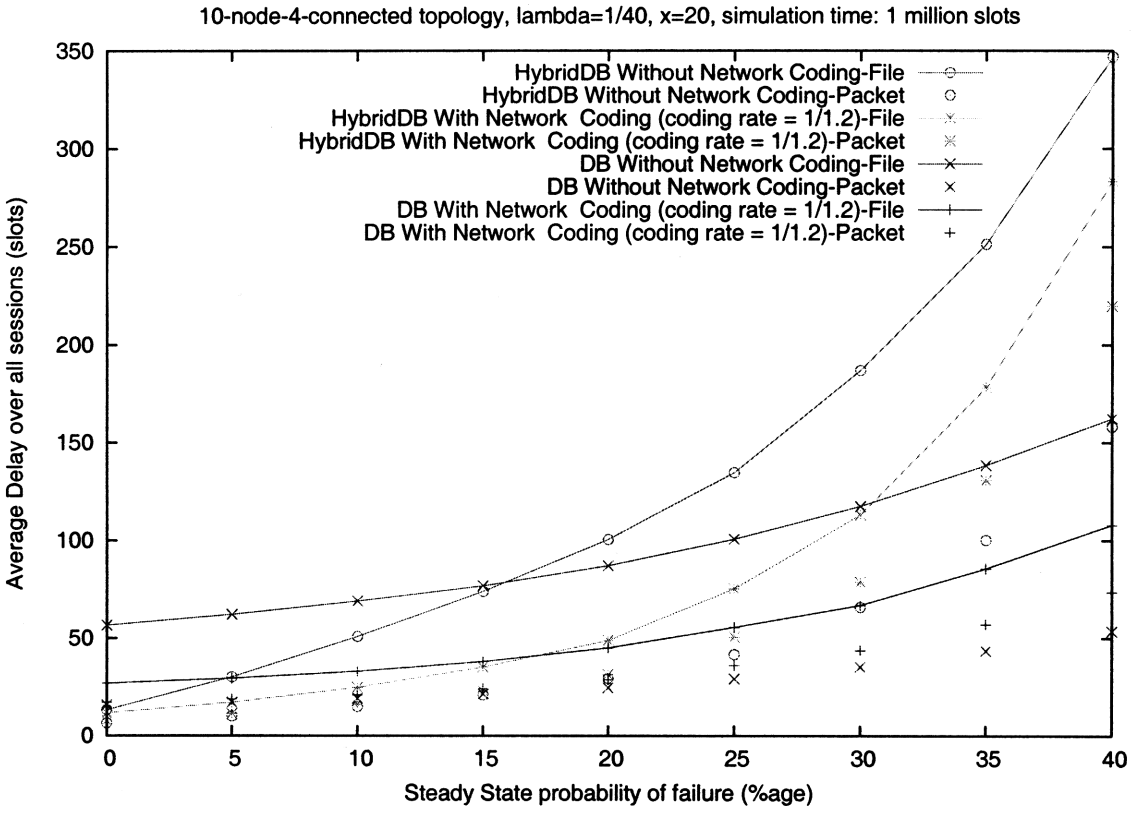

Figure 4-25: Delays of Digital Fountain approach in HybridDB as a function of failure rate for 10-node 4-connected symmetric topology with different values of code rate

10-node 4-connected symmetric topology The results presented in Figs. 4-25, 4-26 and 4-27, show that HybridDB performs better than DB only for low failures or low loads. Hence the choice bewteen HybridDB and DB is dependent on the operating region of the the network.

Qwest OC-192 Backbone Corresponding plots for Qwest OC-192 Backbone are presented in Figs. 4-28, 4-29 and 4-30. HybridDB seems to perform better than Differential Backlog routing for the particular level of network loading used in the plot. The packet delays shown in Fig. 4-30 show a different trend than those obained for 10-node 4-connected symmetric topology. It appears that the distinction between HybridDB and Differential Backlog routing is prominent for low failures whereas at higher failures, the code rate starts to take prominence over the choice of HybridDB and Differential Backlog routing.

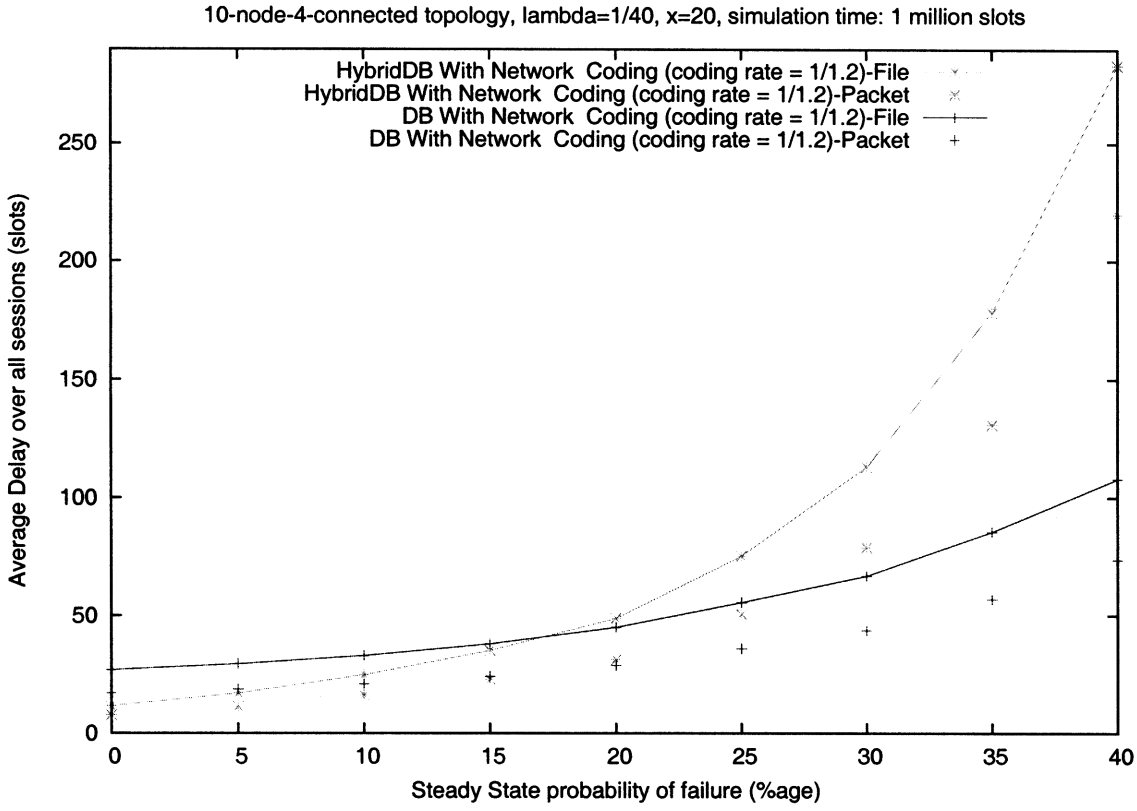

Figure 4-26: Delays of Digital Fountain approach in HybridDB as a function of failure rate for 10-node 4-connected symmetric topology with code rate of 1.2

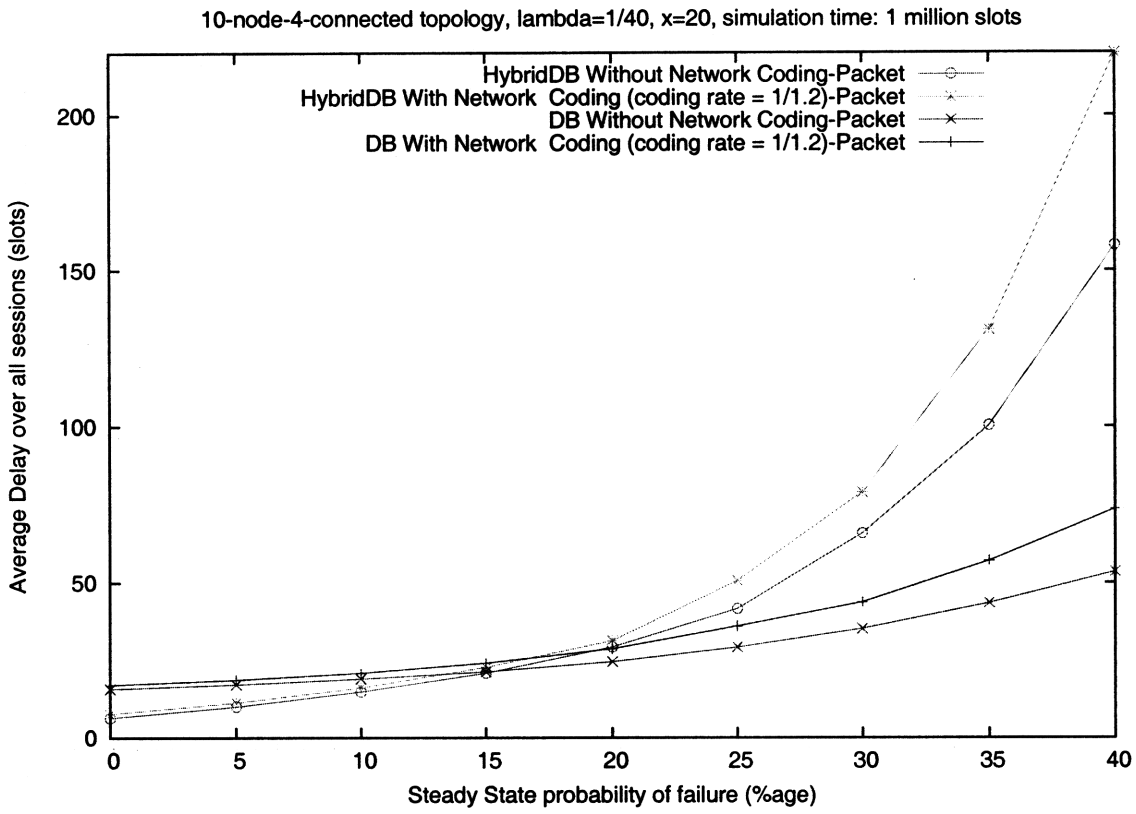

Figure 4-27: Packet delays of Digital Fountain approach in HybridDB as a function of failure rate for 10-node 4-connected symmetric topology with different values of code rate

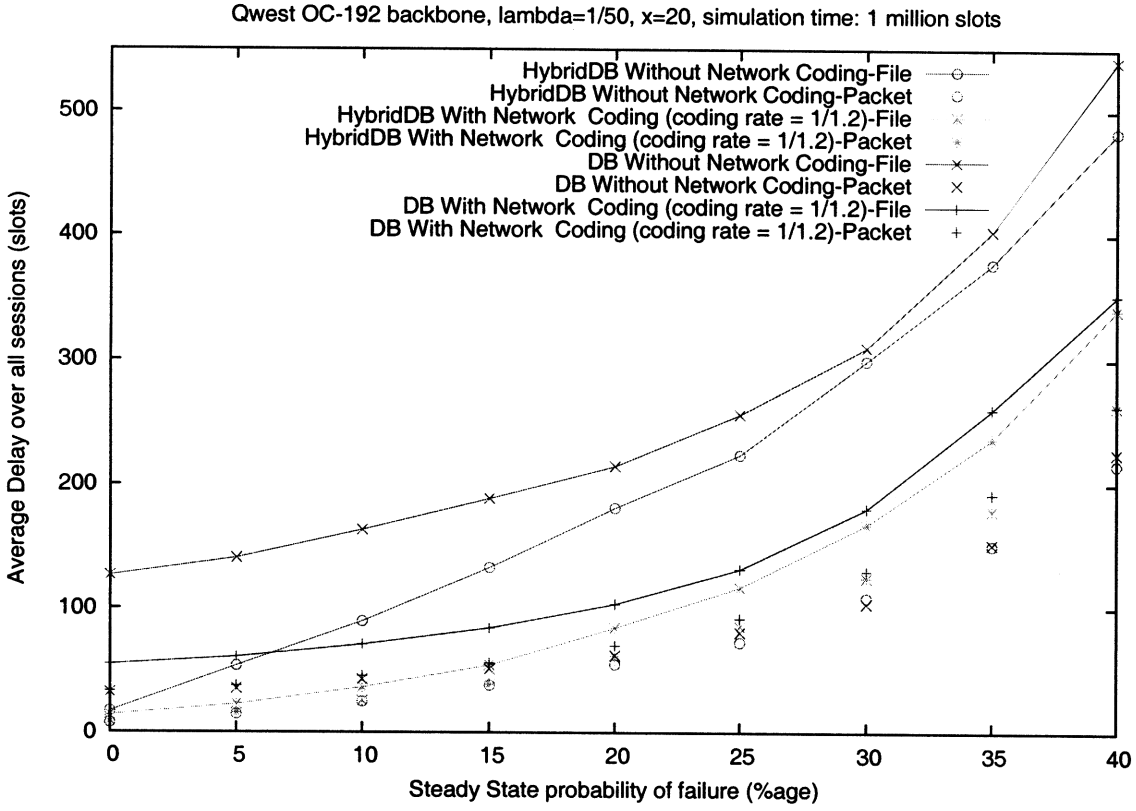

Figure 4-28: Delays of Digital Fountain approach in HybridDB as a function of failure rate for Qwest OC-192 Backbone with different values of code rate

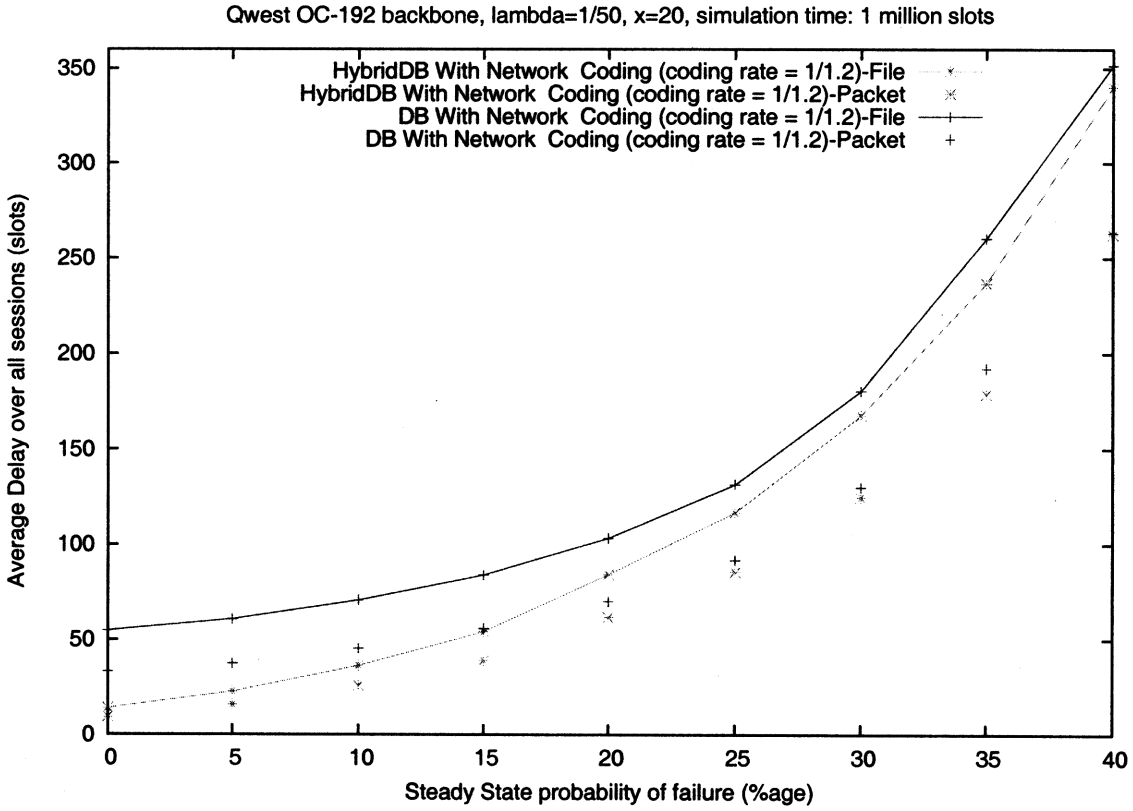

Figure 4-29: Delays of Digital Fountain approach in HybridDB as a function of failure rate for Qwest OC-192 Backbone with code rate of 1.2

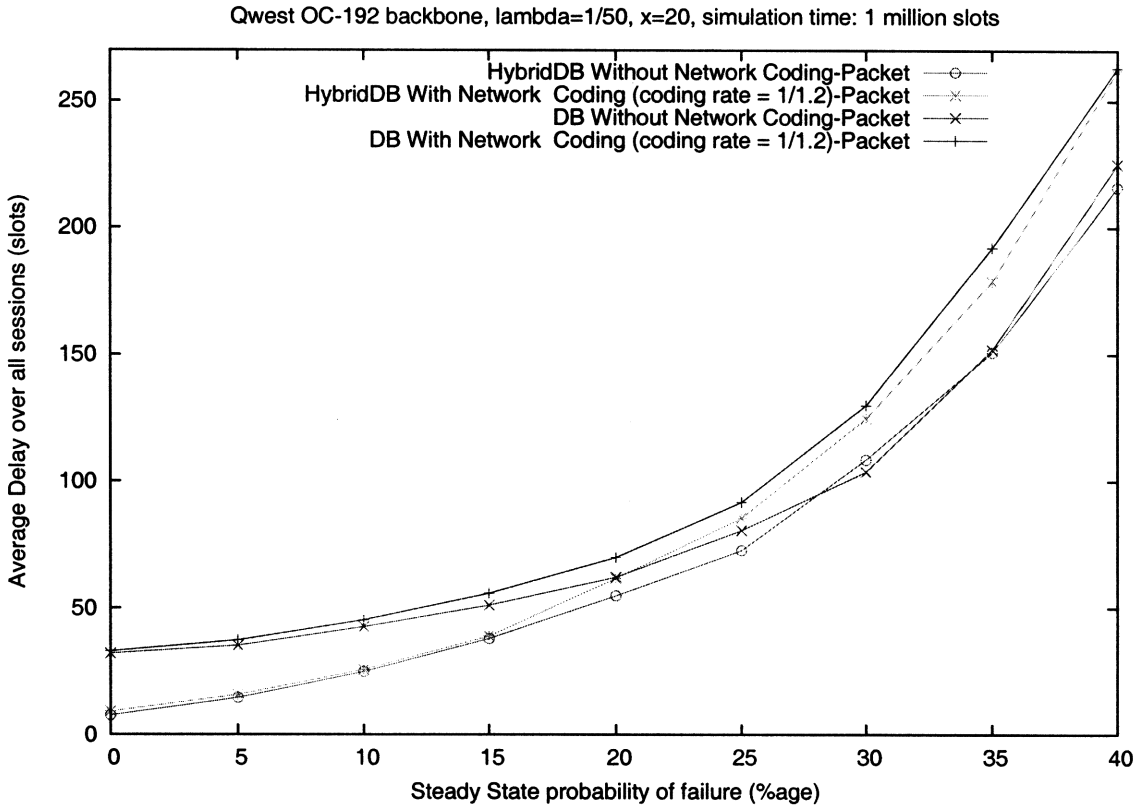

Figure 4-30: Packet delays of Digital Fountain approach in HybridDB as a function of failure rate for Qwest OC-192 Backbone with different values of code rate

Chapter 5

Discussion of Practical issues

No known real-world networks employ Differential Backlog algorithm as their underlying routing protocol. Differential Backlog routing requires quite a distinct set of hardware resources than those used in a conventional network using Shortest Path routing. Practical issues such as packet losses and rate control pose difficult questions for the viability of Differential Backlog routing. These practical hurdles also contribute to unpopularity of Differential Backlog routing, apart from its expected worse delay performance than Shortest Path routing at lower network loads.

Differential Backlog routing and its variants require implementation of input queues at each router in contrast to output-queueing used in modern day networks which use Shortest Path routing. In addition, the input queues must be maintained based on packet destinations rather than the incoming links used by packets. Additional computing power to sort packets based on their destinations, and additional storage to maintain queues for each destination, will be required at each node.

The number of input-queues needed at each router is equal to the number of total destinations in the network. In a nation-wide network with millions of users, it would be impractical to differentiate packets on the basis of their individual destinations. Hence, there would be a need to classify packets in broad destination classes. As we have observed through simulations, not only is Differential Backlog routing sensitive to loading but decreased loading might even be counter productive. Therefore, these broad classifications should not just depend upon geographical locations but must

also take into account relative traffic loads. This approach would be similar to the one used for routing based on closest prefix match used in the Internet. Naturally, this approach will also mandate a regulatory body which would oversee the broad classifications.

In our simulations, we did not take into account packet loss due to link failures, buffer overflow, internet blackholes etc. a phenomenon, which is very realistic in networks. Digital fountains with adaptive coding rates can be cleverly used to prevent retransmissions of lost packets similar to how TCP controls transmission rate. This way, one can trade delay gains for better response to increased failures and vice versa through digital fountains. Nevertheless, a transport protocol with basic features such as session initialization, acknowledgement of the number of packets that have been received at the receiver and session termination would be needed . In addition, special packet types would need to be instituted for distribution of Differential Backlog information and shortest-path costs in the network.

Increased computing will also be required both on the sender and receiver sides to encode and decode digital fountains and can also contribute to additional delays. These delays, which we have conveniently ignored in our simulations, will have to be compared with the delay gains of employing digital fountains in practice.

Another concern for Differential Backlog routing is privacy. Packets could visit any node in the network and an intermediate node can look at the contents of a packet at its discretion. This issue is also present in broadcast networks and can be removed by using shortest path routing with active components at the Local Area Network (LAN). The approach assumes that other higher level nodes in MAN and WAN are secure and trustworthy.

Lastly, it would be unrealistic to expect up-to-date and precise information about queue occupancies in a distributed implementation of a Differential Backlog algorithm. A separate channel on the lines of a control wavelength can be used to maintain information about queue backlogs across each link. The control channel can be used to periodically update each queue backlog in a fixed order. Keeping bandwidth the same, coarseness in backlog information can be traded for more frequent

updates to differential backlogs. Along the same lines, clever algorithms for finding the maximum differential backlog across a link can be implemented. For instance, an algorithm can calculate and keep track of the maximum differential backlog. It only needs to calculate the new differential backlog after an update and compare it to the previous maximum to find if the maximum needs to be changed. If it does need to be changed, the maximum differential backlog is updated and thus the algorithm runs in constant rather than linear time in terms of the total number of different destination classes.

Chapter 6

Future Directions

In our simulation study, we have not carried out an exhaustive analysis of different routing schemes in terms of the input space of all the parameters involved. Significant amount of time was spent on developing simulations and limited computer resources allowed simulations to be run only for carefully selected values of input parameters. With more computing resources and time at hand, one can carry out simulations which span all the interesting regions of input space of parameters. Three-dimensional plots can be generated to better understand the coupled effect of various parameters. It will also allow for identification of any regimes which are optimal given non-discretionnry parameters such as traffic rates.

An adaptive control mechanism which can dynamically select a routing paradigm in response to input traffic can enable a network to combine the best of both worlds. At low loads, one can get the performance gains of Shortest Path while at high loads or failures, one can utilize the capacity optimality of Differential Backlog. Alternatively, one can exhaustively simulate a network's performance under different routing schemes for all the governing parameters, pick design parameters that result in optimal behavior given a load profile and identify superior of the two routing schemes. Monitoring of traffic conditions then allows one to pick Differential Backlog or Shortest Path, which ever has been seen to behave better through simulations for the observed level of load.

Differential Backlog routing autonomously re-routes data along available network

capacity whereas HybridDB tries, in addition, to route data packets towards their destinations. HybridDB, with further modification, can also provide for different priority levels for traffic [28]. We bias the utility function in (4.1) to provide priority differentiation as follows:

$$c^*_{ab}(t) = \arg \max_{c \in 1,\ldots,N} \{\theta^c_a(U^{(c)}_a(t) + \alpha V^c_a(t)) - \theta^c_b(U^{(c)}_b(t) + \alpha V^c_b(t))\} \qquad (6.1)$$

where θ^c_i denotes the priority level for traffic destined for node c from node i. V^c_i denotes the path distance between node i and node c and α is a scalar constant as in

.

The simulation study of Differential Backlog routing can be extended in many ways. General network settings, distributed implementations of Differential Backlog routing algorithm and further detailed performance evaluation can more convincingly demonstrate the viability and effectiveness of using Differential Backlog routing or one of its variants. Examples of more general network settings include variable link capacities, directed links, scale-free graphs and richer traffic models. As far as distributed implementation is concerned, propagation, transmission and processing delays; asynchronous file and packet arrivals; distributed implementation of shortest path and Differential Backlog routing algorithms are some examples. Queue lengths and running times can provide for more detailed performance evaluation.

Theoretical treatment of delays in Differential Backlog routing and its suggested variants is also a challenging but equally insightful area to be explored.

Chapter 7

Conclusion

As one would suspect, Differential Backlog algorithm and its variants have a superior capacity region than Shortest Path algorithms. However, suspicions regarding worse delays have also proved true; Differential Backlog exhibits far higher delays than its counterpart for small loadings. Interestingly, the delay performance of Differential Backlog can improve with increased loading. As we have observed in the examples of the two topologies under simulation, the delays can be relatively constant for loadings of up to twice as much as those for which Shortest Path is stable. The ratio is even higher for networks with high failures.

The important question to ask is what loading region does a network operate in. If the loads are variable, as the case with most realistic networks, then important metrics would be the maximum, minimum and average loads. Network engineers are faced with a decision of whether they can leverage network utilization with worse delay performance. In practice, customers would not compromise delay performance even if it comes at a cost. Real world applications that are insensitive to delays are relatively few. Examples include peer-to-peer sharing services such as BitTorrent and advertisements. In comparison, frequently used services such as web surfing, digitial streaming and teleconferencing have very stringent QoS constraints on delays. Internet traffic studies [8] show that network resources are generally under-utilized except for large short-lived bursts in traffic. Conventioanl Shortest Path based routing mechanisms necessitate the under-utilization in a network to promise convergence

and QoS specifications. Opportunities for deployment of Differential-Backlog based routing might spring up if its delays are improved, and as customer base and bandwidth requirements per individual increase, requiring higher network capacities.

We can only make qualitative comparisons about relative delay performances of Differential Backlog and Shortest Path routing from our simulation study. The exact comparison can only be made through an experiment and if the delays for Differential Backlog fall within QoS constraints of most applications, it can be a real success. Contemporary routing schemes barely meet existing QoS specifications, so it is unlikely that Differential Backlog delays will be acceptable for prevalent traffic demands. It is quite possible, however, in future that advances in the design of underlying network hardware (links, routers) enable network delays to diminish a great deal. In that case, Differential Backlog routing could still have prospects in future assuming stricter delay-constrained applications do not arise.

There are situations where Differential Backlog algorithm proves to be the only effective mechanism for routing. The failure model used in our simulation study puts significant stress on a network and in turn the routing algorithm being used. We have observed Differential Backlog to be the only available choice for routing for high failures and network loads. Mutliple WMD attacks, earthquakes, floods, hurricanes are similar and very realistic situations which subject a network to high stresses but occur too rarely (hopefully) to provision spare capacity for. Even if spare capacity was built in, chances are that it would also get damaged and lost in these scenarios. In these circumstances, even normal loads would be unstable under Shortest Path routing because of decreased network capacity. Rather, emergency situations require higher than normal loads because of co-ordination of relief work and increased user activity due to mass hysteria. In these settings, delays can be less important than capacity since there is no available alternative as it is preferable to have something than nothing.

The variant of Differential Backlog- namely HybridDB along with optimized use of digtial fountains can improve the delay performance of Differential Backlog. These variants of Differential Backlog algorithms broaden its appeal to practical networking

circles and can prove instrumental in the field deployment of Differential Backlog. The observation that HybridDB can exhibit better delays than Shortest Path routing even at relatively low loads is a motivation that should be good enough for network engineers to start thinking about Differential Backlog routing as practical and viable routing algorithm.

Bibliography

[1] `http://www.telegeography.com/maps/`.

[2] D. Bertsekas and R. Gallager. *Data networks.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1987.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability.* Athena Scientific, 2002.

[4] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing.* Kluwer Academic Publishers, 1999.

[5] R. Black, A. Donnelly, and C. Fournet. Ethernet topology discovery without network assistance. *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 328–339, 2004.

[6] D.S. Bond. Method of storing spare satellites in orbit, December 7 1976. US Patent 3,995,801.

[7] JW Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *Selected Areas in Communications, IEEE Journal on*, 20(8):1528–1540, 2002.

[8] V.W.S. Chan. Optical communications. `web.mit.edu/6.442`.

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition.* The MIT Press, September 2001.

[10] A. Fumagalli, M. Tacca, F. Unghvary, and A. Farago. Shared path protection with differentiated reliability. *Communications, 2002. ICC 2002. IEEE International Conference on*, 4, 2002.

[11] B. Halabi, S. Halabi, and D. McPherson. *Internet Routing Architectures.* Cisco Press, 2000.

[12] Christian Huitema. *Routing in the Internet.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[13] ISO/IEC 7498-1:1994(E). *Information technology-Open Systems Interconnection - Basic Reference Model: The Basic Model.* ISO, Geneva, Switzerland, 1994.

[14] Bobbie Johnson. How one clumsy ship cut off the web for 75 million people. *The Guardian (International Edition)*, page 22, February 2008.

[15] Daniel Dao-Jun Kan. Design of survivable ip-over-wdm networks: Providing protection and restoration at the electronic layer. Master's thesis, Massachusetts Institute of Technology, 2003.

[16] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary Poisson view of Internet traffic. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 3, 2004.

[17] J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 76–85, 2004.

[18] L. Kleinrock. *Queueing systems. volume I, Theory.* New York: Wiley, 1974.

[19] W. Koechner and R.G. Buser. Optical fiber security system, May 27 1986. US Patent 4,591,709.

[20] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. *Networking, IEEE/ACM Transactions on*, 9(3):293–306, 2001.

[21] C. Labovitz, GR Malan, and F. Jahanian. Internet routing instability. *Networking, IEEE/ACM Transactions on*, 6(5):515–528, 1998.

[22] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. Bgp beacons. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14, New York, NY, USA, 2003. ACM.

[23] Ray G. McCormack and David C. Sieber. Fiber optic communications link performance in EMP and intense light transient environments. Technical report, Construction Engineering Research Laboratory (Army) Champaign ILL, October 1976.

[24] N. McKewon, A. Mekkittikul, V. Ananthram, and J. Walrand. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications*, 47(8):1260–1267, August 1999.

[25] M. Mitzenmacher. Digital fountains: a survey and look forward. *Information Theory Workshop, 2004. IEEE*, pages 271–276, 2004.

[26] E. Modiano and A. Narula-Tam. Survivable lightpath routing: a new approach to the design ofWDM-based networks. *Selected Areas in Communications, IEEE Journal on*, 20(4):800–809, 2002.

[27] A. Narula-Tam, E. Modiano, and A. Brzezinski. Physical topology design for survivable routing of logical rings in WDM-based networks. *Selected Areas in Communications, IEEE Journal on*, 22(8):1525–1538, 2004.

[28] M.J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels.* PhD thesis, Massachusetts Institute of Technology, 2003.

[29] MJ Neely, E. Modiano, and C. Li. Fairness and Optimal Stochastic Control for Heterogeneous Networks. *Networking, IEEE/ACM Transactions on*, 16(2):396–409, 2008.

[30] MJ Neely, E. Modiano, and CE Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *Selected Areas in Communications, IEEE Journal on*, 23(1):89–103, 2005.

[31] C. Ou and B. Mukherjee. Differentiated Quality-of-Protection Provisioning in Optical/MPLS Networks. *Proceedings of 3rd IFIP-TC6 Networking Conference*, pages 650–661.

[32] Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4), 1995.

[33] A. Sahoo, K. Kant, and P. Mohapatra. Improving BGP Convergence Delay for Large-Scale Failures. *Proceedings of the International Conference on Dependable Systems and Networks (DSN'06)-Volume 00*, pages 323–332, 2006.

[34] A. Sahoo, K. Kant, and P. Mohapatra. Speculative Route Invalidation to Improve BGP Convergence Delay under Large-Scale Failures. *Computer Communications and Networks, Proceedings-15th International Conference on*, pages 461–466, 2006.

[35] J. Sun and E. Modiano. Capacity provisioning and failure recovery for Low Earth Orbit satellite constellation. *Int. J. Satell. Commun*, 21:259–284, 2003.

[36] L. Tassiulas. Linear complexity algorithms for maximum throughput in radionetworks and input queued switches. *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2.

[37] L. Tassiulas. Scheduling and performance limits of networks with constantly-changing topology. *Information Theory, IEEE Transactions on*, 43(3):1067–1073, 1997.

[38] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radionetworks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

[39] H. Wang, E. Modiano, and M. Medard. Partial path protection for WDM networks: end-to-end recovery using local failure information. *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, pages 719–725, 2002.

[40] Guy E. Weichenberg. High-reliability architectures for networks under stress. Master's thesis, Massachusetts Institute of Technology, 2003.