CHAPTER I

INTRODUCTION

1. Background

Considerable research in communication networks has recently focused on techniques appropriate for the implementation of very high speed networks (i.e., with transmission rates exceeding 1 gigabit/sec.). The demand for higher transmission rates has grown with the rapid expansion of the number of devices connected to networks, most notably personal computers and workstations. The demand for increased communication rates has also been partly created by proposed services requiring high bandwidth [1]. Such services include multimedia conferencing, digital transmission of high definition television, medical imaging, interactive video education, process and data sharing between tightly coupled computers, real time visualization, customer selectable broadcast video entertainment, etc.

Advances in lightwave communications technologies have also motivated the study of very high speed networks. Lightwave communication uniquely offers the bandwidth required to implement the range of desired services in a cost efficient manner. Lightwave systems also offer other desirable features such as noise immunity, relative security from eavesdropping, and high reliability. But applications of lightwave technology to date have been largely limited to point-to-point communications links, often merely replacing their existing electronic counterparts. Hence, a key technology which promises to enable network designers to achieve their applications' objectives is available, and efficiently tapping its capacity is the objective.

In the past 25 years, a huge amount of research and development effort has been dedicated to the implementation and analysis of low and medium speed communications networks. Unfortunately, it is often neither desirable nor possible to use conventional networking techniques when building very high speed networks. There are two reasons for this. First, conventional networking techniques have become extraordinarily complex. As transmission speed increases, message arrival rates generally increase, and there is less time available to perform necessary network functions (e.g., error checking) on arriving messages. As one illustration, the amount of processing time needed to execute many existing communication protocols would be prohibitive in a high speed network. This has led to proposals for fast transport protocols, such as Chesson's eXpress Transfer Protocol (XTP) [2].

The second reason why conventional networking techniques are no longer appropriate is that as transmission speed increases certain other parameters, such as propagation delay, remain fixed. The design of lower speed network functions often assumed a relationship between these parameters which may no longer hold in high speed networks. For instance, windowing is a popular admission (flow) control technique in low speed networks. Here the relationship between window size and propagation delay can dramatically affect overall network performance. It remains unclear whether windowing is a viable technique for high speed networks. This question is being addressed by Mitra [3].

While research on high speed networks has initiated a reexamination of many classical network techniques, there is also a search for entirely new approaches to network design. Examples of new approaches include novel switching architectures [4], streamlined protocols [2], wavelength-division multiplexing [5], optical packet compressors and expanders [6], and the use of communication links as a temporary storage media [7]. Central to many new proposals is to fully exploit optical communications technologies. To date lightwave communication has been used primarily in conjunction with electronic switching technology operating at much lower speeds. While "all optical" networks have been proposed [8], these proposals are generally still limited by electronic switching speeds, or the slower speeds required to change optical properties of materials. Yet advances in optical switch technologies are quite promising, and a breakthrough in this technology appears quite possible. There is relatively little that can be said about the form that optical switches and "all optical" networks may take. However, it is reasonable to assume that these switches will be rudimentary when judged by the complex standards of present day electronic switches.

Hence, it appears that in the short term high speed networks will continue to mix optical and electronic technologies, while in the longer term we strive for "all-optical" implementations. It is judicious to investigate design alternatives which will be appropriate in both mixed electronic and optical and entirely optical network implementations. Simplicity is then a primary design objective to guarantee that the network design will be applicable to all optical implementations. In this thesis, we will examine one necessary component of a multihop data network, the routing function. Of particular interest is how to implement this function in a simple yet efficient manner.

1.1. Routing

Consider a fixed communication network of N nodes with a corresponding strongly connected graph. Each node represents a processor or subnetwork which is both a potential source and destination for messages or *packets.* We take communication links (i.e., edges) to be unidirectional, and assume that each link is capable of transmitting only one packet at a time (bidirectional links are modeled as 2 unidirectional links with opposite sense, and links capable of transmitting multiple packets simultaneously are modeled as parallel unidirectional links). We are then interested in the problem of *routing* packets from their sources to their destinations in a multi-hop network.

A *routing algorithm* specifies the path a packet follows from its source to its destination. A particularly simple form of routing is *source* routing. In source routing a packet's entire route is determined at its source, and these routing instructions are carried through the network by the packet. Hence the routing of an admitted packet at an intermediate node simply reduces to "reading" a packet's routing "tag" and assigning the packet to the specified outgoing communication link.

A route might otherwise be determined by a sequence of routing decisions made at nodes visited by a packet. At a given node the algorithm specifies the outgoing communication link on which a packet exits the node. It is often the case that this specification is nondeterministic. Since each node may receive packets on each of its incident communication links, multiple packets may arrive at a node and simultaneously contend for access to an outgoing link. The conventional means of resolving this contention without loss of packets is *store-and-forward* routing; packets contending for a given communications link are queued and transmitted serially, according to some queueing discipline.

There are a variety of ways in which routing algorithms are classified. Routing algorithms may be static (i.e., oblivious) or dynamic. A *static* algorithm routes each packet with a given source and destination along a fixed path (i.e., time homogeneous). A *dynamic* algorithm may route packets destined for the same node along different paths. The particularly promising class of dynamic routing algorithms we study in this thesis are *adaptive*. Adaptive algorithms are those which may use information about prevailing conditions to determine a given packet's route. The most commonly proposed form of adaptive algorithms make routing decisions on the basis of current or recent traffic and congestion conditions. Remarkably few theoretical results exist for adaptive routing schemes, though it is conjectured that they efficiently route packets [9]. A key objective of this dissertation is to explore the efficiency of such schemes.

Routing algorithms may be further classified as being *local* (i.e., distributed). Such algorithms use only information locally available at a node to determine the assignment of a packet to a communication link. In our study we will exclude information that may be collected by packets as they pass though the network. Routing algorithms may or may not use memory of past events that have occurred on either a packet, node, or network basis. In this thesis, with one notable exception (the *slot count* rule of Chapter II), we will consider *memoryless* algorithms.

In this thesis we consider the performance of a novel local, adaptive routing algorithm known as shortest path *deflection routing*. In deflection routing, each node nominally attempts to route each arriving packet on a shortest path to its destination. Contention for communication link access might arise, and is resolved by purposefully misrouting or *deflecting* packets. That is, communication link contention causes some packets to be routed along longer paths to their destinations. Hence, in marked contrast to store-and-forward networks, packets are not held at a node to await access to a desired communication link. Indeed, in many proposed implementations of deflection routing there are no available buffers for storing packets for later forwarding. Hence, to avoid the loss of packets, we must insure that each packet arriving to a node can be transmitted on some outgoing link in the following time slot. Therefore we restrict our attention to *regular* topologies (i.e., networks of nodes with an identical number of incoming and outgoing communication links).

Intentionally misrouting packets to attain a degree of tolerance to communication link failure was first suggested by Baran [10]. His proposal, known as *hot potato* routing, was however introduced in the context of store-and-forward routing. That is, there was no explicit attempt to limit communications buffers. The focus of Baran's investigation concerned the reliability of a communications network. He suggested that keeping packets "moving," regardless of the optimality of the path, was a desirable property of a survivable network.

Network communications without store-and-forward buffers next appeared in the literature of interconnection networks for multiprocessors. Here a collection of tightly-coupled processors in close geographical proximity share data. Lawrie and Padua [11] analyzed the performance of routing with limited buffers in the folded shuffle-exchange network. In this analysis, packets were not routed along shortest paths; a slightly longer path was sometimes chosen to simplify implementation. Since this time, various forms of routing with limited buffering have been proposed in interconnection networks [12].

Maxemchuk first proposed deflection routing as a suitable technique for a very high speed Metropolitan Area Network [13, 14] and has since published several seminal papers solving a variety of related problems [15, 16, 17]. Since the appearance of these works, a large number of research papers have appeared on this subject. We refer the interested reader to the annotated bibliography in [18] and to recently published manuscripts including [19, 20, 21, 22, 23]. Despite the recent interest, many important problems remain unsolved. The analysis of networks with deflection routing is difficult, and there are only inadequate solutions to certain serious operational problems. Yet these difficulties are being addressed and overcome; we believe that solutions to certain analytical and operational problems are found in this thesis.

Deflection routing has a number of attractive properties. We now briefly mention some of its most compelling attributes.

- *Low transit delay* - Transit or transport delay is the difference in time between when a packet reaches its destination and the time it is admitted to the network (i.e., the time required for an admitted packet to reach its destination). Since nodes always attempt to route packets on shortest (or least time) paths, delay is kept low. Further, since the routing is adaptive, nodes can route arriving packets in a way that is jointly beneficial to the set of arriving packets. Also, forwarding packets in each time slot removes the problem of packets getting "stuck" in buffers for excessively long periods.

- *Congestion diffusion* - A network is congested if an increase in offered load produces a decrease in throughput. The offered traffic pattern can cause congestion to occur in either a network-wide or a localized fashion. If localized congestion occurs in in a network with deflection routing, deflections will occur relatively frequently, forcing packets "away" from the congested region. Hence, congestion is diffused by spreading packets over a wider region with more excess capacity. This property of diffusing congestion automatically causes otherwise underutilized links to share the burden of routing packets.

- *Simple node implementation* - The routing function to be implemented at each network node is simple. First, each node has to route arriving packets along shortest paths. Shortest path determinations can be easily accomplished by table lookup. Second, the absence of store-and-forward buffers frees nodes from complex buffer management tasks. In particular, no packets need be lost due to (transit) buffer overflow.

- *Fault tolerance* - Since the routing algorithm is distributed and can be locally implemented, no single point of network failure exists. On many topologies multiple edge-disjoint paths exist between each source and destination. The presence of alternate routes provides a degree of tolerance to link and node failures, since a packet can circumnavigate these failures. The reliability of such a network has been examined in [24].

- *Worst case transit delay* - As we will show in this thesis, proper routing algorithm design will guarantee that each admitted packet will reach its destination within a certain time. Satisfying such a "hard" delay requirement is particularly important in many applications, such as real-time "packetized voice" transmission. An excessively delayed voice sample which is not available for signal reconstruction may as well be discarded.

Of course, networks using deflection routing also enjoy all of the benefits of multihop networks. For example, adding additional nodes to an existing network also adds communication links, and hence increases cumulative communication capacity. But deflection routing also suffers from the following disadvantages.

- *Packet resequencing* - A message exceeding the data capacity of 1 packet is partitioned into multiple packets at the source node. Deflections may cause an ordered sequence of admitted packets to arrive at their destination in a different order. A reordering of packets is a necessary responsibility of the destination node.

- *Propagation delay* - One measure of the "cost" of a deflection is the additional time traversing a longer path adds to a packet’s transit delay. In networks of nodes which are widely geographically separated, the propagation delay suffered by a misdirected packet may make deflection routing unacceptably expensive.

- *Admission fairness* - Admission fairness describes the ability of each node to obtain "service" (i.e., to admit new packets) which is, in some sense, balanced. Most distributed control networks are susceptible to various forms of unfairness. An extreme unfairness condition, known as *lockout*, occurs when a node is completely inhibited from admitting new packets. Networks with severe buffer limitations are likely to be subject to this problem.
- *Livelock* - Under certain deflection routing algorithms it is possible (though highly unlikely) that packets will circulate indefinitely without ever reaching their destinations. This condition is known as *livelock*. In this thesis we will show how this problem can be avoided through proper routing algorithm design.

Additional discussions of the advantages and disadvantages of deflection routing can be found in [17]. It is notable that most of the disadvantages of deflection routing are also found in other routing proposals. A considerable part of this thesis will be spent analyzing ways in which these problems can be eliminated or mitigated.

1.2. Contents of this Thesis

We now briefly describe the organization of this thesis. In the second chapter we present our model of a node. We then describe the routing algorithms which will be analyzed in later chapters. Finally, we introduce a useful characterization of topologies which will be helpful in determining their suitability for deflection routing.

In the third chapter we present some results that are generally applicable to all regular topologies with deflection routing. In particular, we derive an exact expression for mean packet transit delay in any regular network with a uniform offered load. We then develop a collection of upper bounds on worst case transit delay for any regular network with an *arbitrary* offered traffic load.

In Chapter IV we look at the special case of the *Manhattan Street Network* (MSN), the topology proposed by Maxemchuk as a Metropolitan Area Network. We derive approximate delay distributions for the network operating with a uniform traffic load and several priority based contention resolution rules. We then consider how the delay performance can be improved in 2 ways. First, we add limited capacity store-and-forward buffers to each node. Second, we consider a routing "optimization" conjectured to decrease delay. Simulation results are presented and compared with our analytical results. We then analyze an MSN with a *nonuniform* offered load. Here we also study communication links on which packets suffer propagation delay. Again we derive approximate delay distributions and throughput measures. The study of nonuniform traffic enables us to investigate the ability of deflection routing to diffuse localized congestion.

In the fifth chapter we consider a novel way of combining *trunk* or *channel* grouping and deflection routing. Trunk grouping is a classical form of statistical multiplexing commonly used in telephony. We first introduce the Statistical Data Fork, a trunk group switch proposed in [25]. We then show how networks of these switches can be analyzed. Again we derive approximate performance measures for networks with arbitrary offered traffic loads. In summary, we argue that the combination of channel grouping and deflection routing offers many of the advantages of both techniques, and solves certain problems inherent in each.

CHAPTER II

NETWORK OPERATION AND CHARACTERIZATION

2. A Network Model

In this chapter we introduce a simple model of our network and network nodes. This model will be used for the entirety of our study. We will then introduce a collection of routing algorithms which will be analyzed in later chapters. Finally, we present several ways of characterizing network topologies. These characterizations will be one measure of the suitability of deflection routing for a given topology. Our intent here is to lay the groundwork for studying classes of topologies rather than individual topologies.

Our network is packet switched and time slotted, with time slots taken to be of unit duration. Nodes operate synchronously, and are not subject to loss of synchronization. A node may transmit at most one packet on each of its outgoing communication links in each time slot. Each packet is assumed to be of fixed length. The communication links are constructed from ideal communication channels; transmitted packets are neither corrupted nor lost. Each communication link operates at identical transmission rate. For now we also assume that propagation is instantaneous; in Chapter IV we will consider links on which packets suffer propagation delay. In summary, our network model is an idealized representation of an actual network. From time to time we will comment on how actual network behavior will deviate from this idealized model.

2.1. A Model of a Node

In this section we describe the assumed components and operation of a network node of *degree* 2 (a higher degree node can be constructed by interconnecting nodes of degree 2 [26]). A node has exactly 2 incoming and 2 outgoing links. In each time slot a node receives 0, 1 or 2 packets from directly connected nodes. Each node has a separate packet buffer for each input and output link.

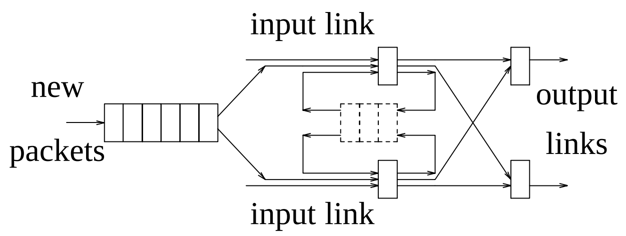


Figure 2.1 - A model of a node. The dashed box represents an optional transit queue.

The link buffer capacities are each exactly 1 packet. Nodes maintain a separate queue for new packets to be admitted to the network. Later, we will introduce an additional queue for temporarily storing packets in transit (i.e., a store-and-forward buffer). For now, this queue is disabled.

2.2. Routing Algorithms and Contention Resolution Rules

At the start of a time slot, each node examines each of its input link buffers for a packet that has reached its destination. The node removes any such packet from the network. If a node has new packet(s) to admit, and the authority to do so, new packet(s) are placed in unoccupied input link buffer(s). The details of packet admittance need not be fully specified for our analysis; we require only that the number of packets in the network in each time slot is a stationary random process. We will assume that new packets which can not be admitted (i.e., blocked packets) are discarded (in an actual network such packets would likely remain queued in a separate admission buffer). After any new packets are admitted the routing algorithm is executed: each packet in an input link buffer is "moved" to an output link buffer to be transmitted on the corresponding output link in the next time slot. Hence, each packet arriving in a time slot to a node other than its destination is forced to exit (i.e., is transmitted) in the next time slot.

Each node attempts to route continuing packets along shortest paths to their destinations. A node is an *option point* for a packet if there exists a shortest path from the node to the packet's destination node using either outgoing edge. If a shortest path to the packet's destination node exists only along 1 outgoing edge, the node is a *critical point*. If only 1 of a node's 2 outgoing links is along a shortest path (i.e., the packet is at a critical point), that link is chosen as the packet's *routing preference.* If both outgoing links are along shortest paths (i.e., the packet is at an option point), neither link is preferred. If only one packet arrives to a node and has a routing preference, it exits via its preferred link. If only one packet arrives and has no routing preference, a fair coin toss assigns the packet to an outgoing link. If two packets arrive and have different routing preferences, they each exit via their preferred links. Otherwise, a contention resolution rule determines packet-to-link assignments.

We study 4 contention resolution rules. Each rule is a *deferring* rule; a packet at an option point *defers* to a second, critical point arrival by permitting the second arrival to exit on its preferred outgoing link. The first 2 rules are due to Greenberg and Goodman [27].

- **Random -** If two packets arrive and both are at critical points or both are at option points, a fair coin toss assigns the arrivals to links.
- **Straight -** Label the incoming and outgoing links at a node upper or

Essentially Greenberg and Goodman's *priority/deferring* rule [27].

lower. Call this labeling the link *type*. We say that a packet passes *straight* through a node if it arrives and departs on links of identical type. If two packets arrive at a node and both packets are at option points, or both are at critical points and have identical routing preference, the packets are passed *straight* through the node.

- **Slot count -** Each packet admitted to the network carries a time slot count, initialized to 0 and incremented in each time slot. If two packets arrive and both packets are at option points, they are passed straight through the node. If both arrivals are at critical points, the packet with the larger slot count is assigned the outgoing link matching its routing preference, and the "younger" packet is deflected. If the arriving packets' slot counts are equal, a fair coin toss assigns packets to links.

- **Destination Distance -** If two packets arrive to a node and both packets are at option points, they are passed straight through the node. If neither arrival is at an option point, the packet closest to its destination is assigned the outgoing link matching its routing preference, and the packet furthest from its destination is deflected. If the arriving packets are equally distant from their respective destinations, a fair coin toss assigns packets to links.

The last 2 contention rules have been intentionally left simple. Resolving contention on the basis of either time in network or destination distance is a common technique [11, 19, 28, 29]. We do not claim that either rule is optimal in any sense. However, simulation and analysis suggest they perform well.

We have also studied other contention rules, whose descriptions we have omitted. One contention rule studied in [28] gives highest priority to an arriving packet that had been deflected the largest number of times. We have studied a similar rule and have found that it offers essentially nothing more (and perhaps less) than that obtained by the slot count rule. For a given application one could likely conceive of a contention rule, based on some heuristic, which produces some desired performance (e.g., minimizing maximum delay). But, in general, it is desirable to find rules which are optimal with respect to some objective function. Unfortunately, a rigorous derivation of an optimal rule appears to be a very difficult problem for most network topologies. Only one such optimal rule has been derived. Krishna and Hajek [18, 28] have shown that a contention rule giving priority to packets closest to their destinations minimizes expected delay in a shuffle-exchange network. They assumed that packet destinations were uniformly distributed and arrivals at each node in each time slot were independent. Their result is immediately extensible, with identical assumptions, to a subset of a class of networks we will introduce in Section 2.4. We will return to a discussion of routing optimization in Chapter IV.

Finally, when we study any routing rule it is often helpful to consider how a packet would be routed in a network otherwise free of packets. Clearly, such a packet could not be deflected. We will refer to this situation as a *zero-deflection* model.

2.3. Deflection Routing with Limited Store-and-Forward Buffering

We next consider node operation and routing when a packet transit buffer is included at each network node. Node operation is similar to operation without transit buffers, and again we study the contention resolution rules described in the previous section. However, a packet unsuccessful in capturing its desired output link is placed in the transit buffer, if it is not full. If full, the packet is placed in the buffer of the unused outgoing link, to be transmitted (i.e., misdirected) in the next time slot. Hence, only packets that lose in link contention and would otherwise be misdirected can be admitted to the transit queue. If either or both output link buffers are empty and a packet is in the transit queue, the node checks the routing preference of the packet at the head of the queue. If its preferred output link buffer is empty, the queued packet is moved into that output link buffer for transmission in the next time slot. Otherwise, the packet remains in the queue.

2.4. "Go-back" Networks

We begin by defining classes of networks in terms of the structure of their corresponding graphs. Consider a graph $G = (V, E)$ with three distinct nodes $a, b, c \in V$ with edge $(a, b) \in E$. Let d_{xy} denote the shortest path distance (in hops) between node *x* and node *y*. Suppose that node *b* is not along a shortest path from node *a* to node *c* (i.e., $d_{ac} \leq d_{bc}$). Thus, the length of a shortest path *abc* (i.e., from node *a* to node *c* via node *b*) is $1 + d_{bc} - d_{ac}$ hops longer than the shortest path *ac*.

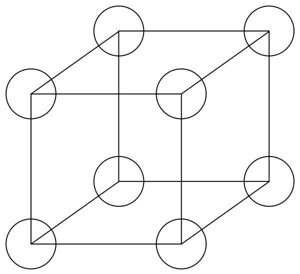
Now consider the paths taken by packets traversing the network corresponding to *G*. We say a packet is *deflected* if in a time slot it does not travel to a node which is closer to its destination. We say that packet *A* is deflected by packet *B* if all of the following are true:

1) Both packets *A* and *B* are located at the same node at the beginning of a time slot.

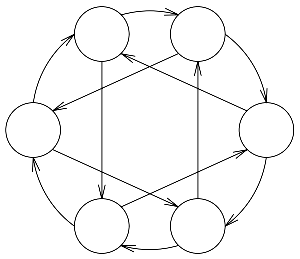
2) Packet *A* is deflected in the time slot.

3) If during the time slot packet *A* instead traveled on the link *B* traveled on, then packet *A* would not have been deflected.

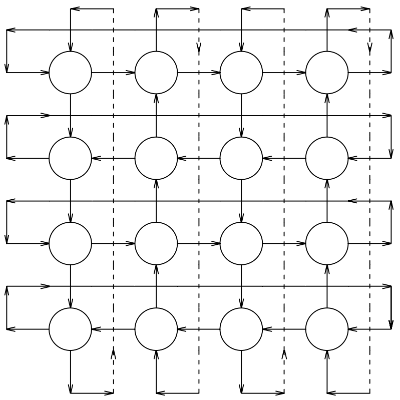
We define the *distance* of a deflection as the difference in a deflected packet's destination distance (in hops) immediately before and after a deflection. A deflection from node *a* into node *b* (w.r.t. destination node *c*) has distance $d_{bc} - d_{ac}$. We define the deflection *index* (D_I) of the network (or graph) as



a) The binary 3-cube is a "go-back-1" network.



b) This chordal ring is a "go-back-2" network.



c) This 16 node Manhattan Street Network is a "go-back-3" network.

Figure 2.2: Some common "go-back-d" networks with $d = 1, 2, 3$.

$$D_I \triangleq 1 + \max_{a,b,c \in V} d_{bc} - d_{ac} , \qquad (a,b) \in E. \qquad (2.1)$$

The index equals the maximum number of additional hops a deflection can add to a packet's path length.

We next define classes of networks in terms of the deflection distances which can be realized by traversing packets. A "go-back-d" network is defined by a *constant* deflection distance d for any realizable deflection. Hence, $D_I = d + 1$. Examples of "go-back-1" networks include the binary n-cube and n-dimensional mesh ($n \geq 2$). A Manhattan Street graph with an integer multiple of 4 nodes in each row and column is a "go-back-3" network [30]. One can easily construct a chordal ring of degree 2 to form a "go-back-d" network for arbitrary values of $d \geq 1$.

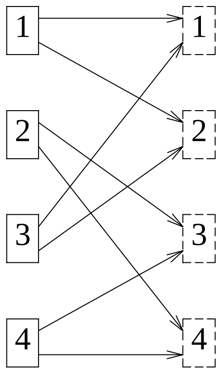


Figure 2.3 - An 8 input folded shuffle-exchange network can implement a "go-back-all" network.

A "go-back-all" network with diameter L is defined by the following property: a deflection of a packet at destination distance i (i.e., exactly i hops from its destination) has deflection distance $L - i$. That is, a deflected packet finds itself at the maximum possible distance from its destination in the time slot immediately follow-

A more direct graph theoretic classification can be developed, however little insight appears to be gained.

ing the deflection. An example of a network that has been studied as a "go-back-all" network is the folded shuffle-exchange [11]. Strictly speaking, however, the folded shuffle exchange network is not a "go-back-all" network; the routing algorithm studied in [11] rather than the network topology imposed the maximum deflection distance. Finally, we define a general "go-back" network as one where deflection distance depends on the particular choice of vertices $a, b, c \in V$. Examples of general "go-back" networks include many graphs of "irregular" topology.

CHAPTER III

DEFLECTION ROUTING ON ARBITRARY REGULAR TOPOLOGIES

3. Introduction

In this chapter we derive a collection of results that are valid for all regular topologies with deflection routing. We begin by deriving an exact expression for mean transit delay in an arbitrary network operating with a uniform traffic demand. We then derive an approximate equation for mean delay when a 1 packet capacity store-and-forward buffer is added to each network node. The performance improvements that can be realized by the inclusion of such a small capacity transit buffer have been explored for specific topologies in [16, 22].

We then move to the analysis of worst case delay. We derive several upper bounds on the longest possible transit delay of a packet admitted to any regular network with deflection routing.

3.1. Mean Delay - Unbuffered Networks

Let $G = (V,E)$ be a graph corresponding to a regular network with diameter P. Suppose the destinations of packets admitted to the network are uniformly distributed over the set of $|V| - 1$ nodes other than the admitting node. Let L_0 be the average shortest path length between two distinct nodes. We write the shortest path distance from node i to node j as $sp(i,j)$. Let $R_m(i)$ be the set of nodes reachable from node i in no less than m hops (i.e., $R_m(i) = \{j \; ; \; sp(i,j) = m\}$, $0 < m \le P$). In a symmetric network we have

$|R_m(i)| \equiv |R_m|$ and the average shortest path length between two distinct nodes is

$$L_0 = \frac{1}{|V| - 1} \sum_{k=1}^{P} k\, R_k . \tag{3.1}$$

In a network of nodes without store-and-forward buffers, transit delay (in unit time slots) is identical to path length (in hops). We will now show that expected transit delay is a simple function of the average probability a packet is deflected upon arrival to a node.

To begin we count the following events in the interval $[0,\ T = n)$, where n is an integer number of time slots:

$$N^T \triangleq \textit{total number of packets admitted to the network}, \tag{3.2}$$
$$D^T \triangleq \textit{total number of deflections of all packets}, \tag{3.3}$$
$$L^T \triangleq \textit{total number of hops by all packets}, \tag{3.4}$$
$$N^T(i,j,x,y) \triangleq \textit{number of new packets admitted by node } (i,j) \textit{ destined for node } (x,y). \tag{3.5}$$

We next partition the event count L^T into 2 components, L_0^T and L_D^T, satisfying

$$L^T = L_D^T + L_0^T , \tag{3.6}$$

where we define L_0^T as

$$L_0^T \triangleq \sum_{i,j,x,y} N^T(i,j,x,y)\ sp(i,j,x,y). \tag{3.7}$$

The 2 components have the following intepretations. Suppose that packets were admitted to the network one at a time (i.e., under our *zero-deflection model*). Each packet would follow a shortest path to its destination. L_0^T is the total number of hops all admitted packets *would have taken* if admitted one at a time. Since the packets are not admitted in this fashion, L_D^T is the number of additional hops that must be taken due to deflections.

The event counts we have defined are used to form the random variables $\frac{L^T}{N^T}$, $\frac{L_0^T}{N^T}$, $\frac{L_D^T}{D^T}$, and $\frac{D^T}{L^T}$.

Assumption 3.1: (Mean Ergodicity) The random variables formed by event count ratios are mean ergodic, with well defined limiting values given by

$$L \triangleq \lim_{T\to\infty} \frac{L^T}{N^T}, \qquad L_0 \triangleq \lim_{T\to\infty} \frac{L_0^T}{N^T}, \qquad D \triangleq \lim_{T\to\infty} \frac{L_D^T}{D^T}, \qquad P_D \triangleq \lim_{T\to\infty} \frac{D^T}{L^T} \tag{3.8}$$

Each limit has a simple interpretation. L and L_0 are the average path length (i.e., expected delay) and average shortest path length (i.e., *zero-deflection* delay), respectively. P_D is the average probability a packet is deflected upon arrival to a node in a time slot, and D is the average number of additional hops a deflection adds to a packet's path. Note that L_0 satisfies (3.1).

Lemma 3.1 - Let G be a regular network operating with shortest path deflection routing. Under a uniform traffic demand, the expected packet transit delay is

$$L = \frac{L_0}{1 - D\, P_D}. \tag{3.9}$$

Proof: The total number hops taken by all packets in the network equals the zero-deflection number plus the hops resulting from deflections:

$$L^T = L_0^T + L_D^T = L_0^T + \frac{L_D^T}{D^T}\, D^T \tag{3.10}$$

$$\frac{L^T}{N^T} = \frac{L_0^T}{N^T} + \frac{L_D^T}{D^T}\, \frac{D^T}{L^T}\frac{L^T}{N^T} \tag{3.11}$$

Taking limits yields

$$L = L_0 + D\, P_D\, L \tag{3.12}$$

and (3.9) follows.□

This simple expression for mean delay holds without independence assumption for all regular networks with shortest path deflection routing. We now present several specific examples. Consider a binary n-cube with $|V| = 2^n$ nodes ($n \geq 2$). Recall that the n-cube is a "go-back-1" network; each deflection causes the deflected packet to take exactly $D = 2$ extra hops to reach its destination. Hence, the lemma indicates that the expected transit delay of packets admitted with uniform destinations in a binary n-cube is

$$L = \frac{\dfrac{1}{2^n - 1} \sum_{k=1}^{n} k \begin{bmatrix} n \\ k \end{bmatrix}}{1 - 2 P_D}. \tag{3.13}$$

If we permit the admitting node to also be a packet destination, (3.13) reduces to

$$L = \frac{n}{2(1 - 2 P_D)}. \tag{3.14}$$

The number of additional hops each deflection adds to a square Manhattan Street Network with an integer multiple of 4 rows and 4 columns is exactly 4. Hence, (3.9) becomes

$$L = \frac{L_0}{1 - 4 P_D}. \tag{3.15}$$

Khasnabish [31] has given simple expressions for L_0 in the square MSN.

From (3.9) and Little’s Result we can obtain an equation for network throughput. In any regular network of $|V|$ nodes of degree δ, the expected number of packets reaching their destination in each time slot is

$$\frac{\delta |V| p}{L_0} (1 - D P_D), \tag{3.16}$$

where p is the link utilization. Finally, it is easy to show that (3.9) can be extended to express expected delay even when the "initial" paths are longer than shortest paths, as long as the average initial path length is given by L_0. Similarly, (3.9) can be used to express expected delay under nonuniform traffic loads, as long as L_0 represents the average initial path length under the nonuniform loading.

3.2. Mean Delay - Buffered Networks

We now derive an approximate expression for expected transit delay in a network of nodes each with a 1 packet capacity transit buffer. Again we assume that the offered load is uniform. Note that, in buffered networks, transit delay is not identical to path length. We continue to use the variable L to designate expected transit delay (in unit time slots). Adding buffers enables more packets to be admitted to the network. But for comparison with results in unbuffered networks, we will limit the average number of packets in the network to $K = |E|\ p\ ,\ 0 < p \le 1$. At the start of each time slot in steady state, some number of admitted packets will occupy transit buffers, and the remaining packets will occupy link buffers. Let q denote the probability that a packet arrives on a link to a node in a time slot. The parameter q may be interpreted as the link utilization for the buffered network, reduced from the utilization p corresponding to the same number of packets in an unbuffered network. Under our assumption of stationarity, q is independent of time slot l. This definition allows us to specify the expected number of packets occupying transit buffers.

Lemma 3.2: Let $G = (V,E)$ be a regular network of degree $\delta = |E|/|V|$ Suppose each node contains a 1 packet capacity transit buffer, and the expected number of all admitted packets is $K = |E| p\ ,\ 0 < p \le 1$. In steady state, each buffer contains on average

$$\delta\ (p - q) \tag{3.17}$$

packets.

Proof: Let $\beta^l(v)$ be the number of packets in the transit buffer of node v at the start of the l^{th} time slot, and let $\zeta^l(e)$ be the number of packets traversing link e at the start of the l^{th} time slot. Let K^l be the number of packets in the network at the start of the l^{th} time slot. Then

$$K^l = \sum_{v \in V} \beta^l(v) + \sum_{e \in E} \zeta^l(e). \tag{3.18}$$

By assumption of stationarity we have $K = E[K^l]$. Taking expectations we have

$$E[K^l] = \sum_{v \in V} E[\beta^l(v)] + \sum_{e \in E} E[\zeta^l(e)], \tag{3.19}$$

$$K = |V|\ E[\beta^l] + |E|\ E[\zeta^l], \tag{3.20}$$
$$|E|\ p = |V|\ E[\beta] + |E|\ q, \tag{3.21}$$

and (3.17) follows.□

Instead of attempting an exact analysis of transit queue statistics, we introduce a simpler, approximate model based on the following assumptions.

Assumption 3.2 - In each time slot, buffer occupancy is independent of the state of packets arriving on the incoming links, i.e.,

$$P[\textit{buffer full}] \approx P[\textit{buffer full} \mid \textit{arriving packet deflected}\ ].$$

Assumption 3.3 - Packets arriving and departing on the links of a node are independent from slot to slot.

The justification for such a model is based on the assumption that both arrivals and the routing preferences of arrivals on the incoming links to a node in time slot are "well mixed." Consequently, deflections will be relatively infrequent, and arrivals to the transit queue in consecutive slots will be nearly independent. Similarly, opportunities for a packet in the transit queue to exit will be nearly independent from slot to slot, and we define this probability to be

$$e \triangleq P[\textit{packet exits buffer} \mid \textit{packet in buffer}].$$

We now use this transit queue model to derive an approximate expression for expected transit delay. We will say that a *generalized* deflection occurs whenever link contention occurs, regardless of whether the "deflected" packet is admitted to the transit queue or is misdirected. Let P_D be the conditional probability a packet is "deflected" upon its arrival to a node in a time slot, i.e.,

$$P_D \triangleq P[\textit{packet is deflected} \mid \textit{packet arrives}]. \tag{3.22}$$

Define the unconditional probability a packet is deflected at a node in a time slot to be

$$P_{D_{NODE}} \triangleq P[\textit{packet is deflected at a node in a time slot}]. \tag{3.23}$$

Let the state of the transit queue indicate the queue occupancy. Using Assumptions 3.2-3 it is easy to show that the Markov chain corresponding to the transit buffer has steady-state solution

$$p_0 \triangleq P[\textit{ empty buffer }] = \frac{e}{e + P_{D_{NODE}}}, \tag{3.24a}$$

$$p_1 \triangleq P[\textit{ full buffer }] = \frac{P_{D_{NODE}}}{e + P_{D_{NODE}}}. \tag{3.24b}$$

Clearly $E[\beta] = p_1$, so the effective utilization q must satisfy both (3.17) and (3.24b). Applying Little's Result and using Assumption 3.2 reveals that the expected number of slots a packet spends in the transit queue (*T*) is

$$T \approx \frac{E[\beta]}{p_0 P_{D_{NODE}}}. \tag{3.25}$$

Recall that the expected number of extra hops due to a *misdirection* is given by *D.* Expected transit delay is the sum of the expected number of hops taken by an admitted packet plus the expected number of time slots a packet spends in transit buffers. The expected number of hops taken is

$$H \triangleq E[\textit{hops taken by an admitted packet}] \approx L_0 + DP_D H p_1. \tag{3.26}$$

The first term is the expected number of hops taken by packets in the absence of misdirections, and the second term is the average number of additional hops due to misdirections. The average time an admitted packet spends in transit queues is given by

$$\begin{aligned} W &\triangleq E[\textit{total time slots an admitted packet spends in transit buffers}] \\ &\approx P_D H T p_0. \end{aligned} \tag{3.27}$$

Combining (3.26-27) indicates that the total expected transit delay is

$$L = H + W \approx \frac{L_0}{1 - Dp_1 P_D} \left[1 + p_0 T P_D\right]. \tag{3.28}$$

Eq. 3.28 is particularly valuable since it is valid for a network operating with any contention resolution rule. The details of the contention rule are embedded entirely in the value of deflection probability P_D.

We can further simplify these expressions by focusing on a network of degree $\delta = 2$. Under Assumption 3.3, the unconditional probability that a packet is deflected at a node is simply

$$P_{D_{NODE}} = 2qP_D. \tag{3.29}$$

The transit queue state transitions can be derived from the following facts:

a) An arrival to an empty transit queue occurs with rate $P_{D_{NODE}}$.

b) A packet exits a non-empty queue *w.p.* 1 given 0 packets arrive to the node.

c) A packet exits a non-empty queue *w.p.* ½ given 1 packet arrives to the node (i.e., the routing preference of the queued packet differs from that of the link arrival half of the time).

Totaling these departure probabilities indicates that $e \approx 1 - q$. The steady state occupancies (3.24) reduce to

$$p_0 = \frac{1 - q}{1 - q + 2qP_D}, \tag{3.30a}$$

$$p_1 = \frac{2qP_D}{1 - q + 2qP_D}. \tag{3.30b}$$

Applying Little's Result and using Assumption 3.2 reveals that the expected number of slots a packet spends in the transit queue (*T*) is

$$T \approx \frac{1}{1 - q}. \tag{3.31}$$

A similar analysis can be used to find mean queueing delay for a network of nodes with larger capacity transit buffers. For the common case when deflection probability is small (i.e., $P_D << q < 1$), the resulting expression is nearly equal to (3.31). But for a given offered load, more packets on average are held in the larger capacity transit buffers, producing a lower link utilization q. Eq. 3.31 indicates that a lower value of q will result in lower mean waiting time, and (3.28) indicates that packets will suffer less transit delay.

3.3. Maximum Delay

In this section we develop upper bounds on worst case packet transit delay in several classes of networks using deflection routing. We make no assumptions regarding the distribution of destinations of admitted packets. We examine 2 distinct types of offered load. We first find the *evacuation* time of packets admitted in a batch in a single time slot, with no subsequent admissions. We refer to this loading as *batch admissions.* We also study the maximum transit delay of any packet when packets are admitted continuously over multiple time slots (i.e., *continuous admissions*). We know of no published results on worst case delay under continuous admissions. However it is known that *livelock* can be avoided in nonminimal routing schemes by the introduction of a priority mechanism [32]. Hajek [33] has derived a collection of upper bounds on the evacuation time of batch admissions in certain networks operating with deflection routing. We summarize these results in the following propositions.

Proposition H1 (Hajek): Suppose a batch of p packets is admitted to a network with diameter L and deflection index D_I. Then if all packets share a *single destination,* the evacuation time (T) satisfies

$$T \le L + D_I(p - 1) \qquad \textit{single destination}. \tag{H1}$$

□

Proposition H2 (Hajek): Suppose a batch of p packets is admitted to a network with diameter L and deflection index D_I. Suppose that arriving packets are assigned to links by a *"one-pass" priority* rule, described as follows. At each node in every time slot, order arriving packets by nondecreasing destination distance. Then, starting with the packets closest to their destinations, assign each packet to an outgoing link along a shortest path, if available. If such a link is unavailable, assign the packet to a randomly chosen available outgoing link.

Under the one-pass priority rule, evacuation time satisfies

$$T \le L\,p \qquad \textit{arbitrary topology, multiple destinations}, \quad \text{(H2)}$$
$$T \le n + 2(p - 1) \qquad \textit{binary n-cube, multiple destinations}. \quad \text{(H3)}$$

□

(H1) is proved easily by showing that at least 1 packet must reach the destination in the first L time slots, and at least 1 packet must exit in each of the subsequent periods of D_I time slots, until the network empties. (H2) is apparently a weak bound for certain highly connected networks, and (H3) represents a significantly improved bound for the binary n-cube.

3.3.1. Analysis

Let i denote a packet's remaining distance to destination. A packet reaching its destination is taken to be at distance $i = 0$, and is immediately removed from the network. Let time slot t refer to the interval $[t - 1, t)$. Define the following variables:

$$N^t(i) \triangleq \textit{total number of packets at distance } i \textit{ in time slot } t,$$
$$N^t \triangleq \textit{total number of packets in network in time slot } t = \sum_{i=1}^{L} N^t(i),$$
$$D^t(i, l) \triangleq \textit{total number of packets at distance } i \textit{ deflected to distance } l \textit{ in time slot } t,$$
$$D^t(i) \triangleq \textit{total number of packets deflected at distance } i \textit{ in time slot } t = \sum_{l=i}^{L} D^t(i, l),$$
$$D^t \triangleq \textit{number of deflections in network in time slot } t = \sum_{i=1}^{L} D^t(i).$$

The evolution of packet destination distances in a "go-back-d" network with diameter L is described by the following system (or *evolution*) equations:

$$N^{t+1}(i) = \begin{cases} N^t(i+1) - D^t(i+1) & i = 0, 1, \ldots, d \\ N^t(i+1) - D^t(i+1) + D^t(i-d+1) & i = d+1, \ldots, L-d \\ N^t(i+1) + D^t(i-d+1) & i = L-d+1, \ldots, L-1 \\ D^t(i-d+1) & i = L. \end{cases} \qquad (3.32a)$$

The system equations for a "go-back-all" network are

$$N^{t+1}(i) = \begin{cases} N^t(i+1) - D^t(i+1) & i = 0, 1, \ldots, L-1 \\ \sum_{j=1}^{L} D^t(j) & i = L. \end{cases} \qquad (3.32b)$$

A general "go-back" network has the following system equations:

$$N^{t+1}(i) = \begin{cases} N^t(i+1) - D^t(i+1) + \sum_{j=(i-D_I+1)^+}^{i} D^t(j,i) & i = 0, 1, \ldots, L-1 \\ \sum_{j=(i-D_I+1)^+}^{i} D^t(j,i) & i = L. \end{cases} \qquad (3.32c)$$

Here the notation $(x)^+$ indicates the larger of x and 0. We begin with a useful result bounding the evacuation time for a batch of N^0 packets admitted to the network immediately prior to the beginning of the first time slot (i.e., $t = 0^-$).

Lemma 3.3: Suppose a batch of N^0 packets is admitted to a "go-back" network with diameter L and deflection index D_I. If

$$N^t(i+1) \geq 1 \qquad N^{t+1}(i) \geq 1 \qquad i = 0, 1, \ldots, L-1, \qquad (3.33)$$

then the network evacuates in time

$$T \leq L + D_I\,(N^0 - 1). \qquad (3.34)$$

Discussion: Note that we have not specified a rule for assigning arriving packets to outgoing links. Equation 3.34 is identical to (H1), but is not restricted to packets sent to a single destination. The condition (3.33) states that at least 1 packet among those found at *every* destination distance *i* must move 1 hop closer to its destination in each time slot. This guarantees that at least 1 packet found at distance *i* at time *t* will reach its destination at time $t + i$, i.e.,

$$N^t(i) \geq 1 \qquad\qquad N^{t+i}(0) \geq 1. \tag{3.35}$$

Proof: Following the proof of Proposition H1, we first show that at least 1 packet reaches its destination in the interval $t \in [1, L]$. We then show that for each time $t_j = L + jD_I$, $j = 0,1,...$, at least 1 packet reaches its destination in $[t_j + 1, t_j + D_I]$, until the network evacuates.

By assumption the network is not empty initially, and hence $N^0(i) \geq 1$ for some $i \in \{1,2,...,L\}$. Let $Q \triangleq \{ i : N^0(i) \geq 1 \}$. Then, (3.35) indicates that for $q \in Q$ we have

$$N^0(q) \geq 1 \qquad\qquad N^q(0) \geq 1. \tag{3.36}$$

Hence, at least one packet reaches its destination in the interval $[1, L]$.

Prior to beginning the second part of the proof, we introduce several definitions. Let $i^t_{\max}$ represent the destination distance of the packet furthest from its destination at time *t*, i.e.

$$i^t_{\max} \triangleq max\{ i ; N^t(i) \geq 1\}. \tag{3.37}$$

We may then define the evacuation time (T) as

$$T \triangleq \min\{ t ; i^t_{\max} = 0\}. \tag{3.38}$$

Finally, we say that a *type A* deflection occurred in time slot *t* iff

$$i^{t+1}_{\max} \geq i^t_{\max}. \tag{3.39}$$

That is, in the time slot immediately following a type A deflection, at least one packet is as far from its destination as was the furthest packet in the previous time slot. Note that a packet deflected in a type A deflection at time *t* must have been at some destination distance $i \in [\,(i^t_{\max} - D_I)^+ + 1,\ i^t_{\max}\,]$ at time *t*. Note that if no type A deflection occurs at time *t*, then

$$i^{t+1}_{\max} = i^t_{\max} - 1. \tag{3.40}$$

Hence only *type A* deflections can postpone a network's evacuation.

We next show the second part of the proof; if the network is not evacuated at a time $t_j = L + jD_I$, $j = 0,1,...$, then at least 1 packet reaches its destination in $[t_j + 1,\ t_j + D_I]$. Consider some fixed time t_j. Since the system is not empty by assumption, a type A deflection must have occurred at some time prior to t_j. Let t_d be the time slot of the first type A deflection which prevents the network from evacuating prior to time t_j, i.e.,

$$t_d \triangleq \min\,\{t : i^{t+1}_{\max} \geq t_j - t\}. \tag{3.41}$$

At time $t_d + 1$, at least 1 packet is sufficiently far from its destination that it can not be reached by time t_j, i.e.,

$$i^{t_d+1}_{\max} \geq t_j - t_d. \tag{3.42}$$

Since $D_I - 1$ is the maximum increase in a deflected packet's destination distance, we have the following bound on the deflected packet's destination distance in the time slot following the deflection:

$$t_j - t_d \leq i^{t_d + 1}_{\max} \leq t_j - t_d + D_I - 1. \tag{3.43}$$

By (3.35), at least 1 packet at distance *i* at time *t* must reach its destination at time $t+i$. So (3.43) indicates that at least 1 packet must reach it destination in the interval $[t_j + 1,\ t_j + D_I]$. □

3.3.2. Admissions to a Single Destination

We first consider the evacuation time of a batch of packets all destined for the same node. With a single destination batch demand, all packets visiting a node are equidistant from the destination node. Consequently, all contending packets have identical destination distance. In the following Proposition, we state that the single destination demand itself is sufficient to satisfy condition (3.33), so Lemma 3.3 may be used to bound network evacuation time.

Proposition 3.1: Suppose a batch of N^0 packets is admitted to a network with diameter L and deflection index D_I. If all packets are destined for the same node, then the evacuation time satisfies

$$T \le L + D_I(N^0 - 1). \tag{3.44}$$

Proof: Label the destination node *d*. In any time slot prior to *T*, select any other node holding at least one packet. All packets at the chosen node are equidistant from destination node *d*. At least one outgoing edge from the chosen node must be incident upon a node exactly one hop closer to the destination node *d*. Hence, at any nonempty node at least 1 packet must move one hop closer to the destination. It follows that at least 1 packet among those at every destination distance moves 1 hop closer to the destination in each time slot. Hence, condition (3.33) is satisfied, allowing us to apply Lemma 3.3. The lemma indicates that the network evacuates in time given in (3.44). □

Discussion: Proposition 3.1 is identical to Proposition H1. In the remainder of this section, we show that this result can be improved and extended in several ways.

Batch and Continuous Admissions in "Go-back-d" Networks

We next examine the special case of admissions to a single destination "go-back-d" network. We will show that the batch evacuation time bound (H1) can be reduced by a factor of $d + 1$, approximately. Further, this bound will be shown to apply to certain cases of continuous admissions as well.

Suppose packets are admitted continuously, and let $i(t)$ denote a marked packet's remaining destination distance at time t. Define the *admission class* of a packet admitted at time t_0 with initial distance $i(t_0)$ as

$$\{\ t_0 + i(t_0)\ \}\ mod\ D_I \quad \in \quad \{0, ..., D_I - 1\}. \tag{3.45}$$

There are exactly $D_I = d + 1$ admission classes. Recall that each deflection in the "go-back-d" network has distance *d*. A packet admitted at time t_0 at distance $i(t_0)$ can only reach the destination at some time $t = t_0 + i(t_0) + jD_I,\ \ j = 0,1,...$.

Fact: If all packets admitted to a "go-back-d" network are sent to a single destination, then any contending packets belong to the same admission class.

Proof: Consider any 2 packets *a* and *b* contending for an outgoing link at a node at time t. Suppose the packets were admitted at times t_a, t_b, at initial distances $i_a(t_a)$, $i_b(t_b)$, respectively. At time *t*, the packets are at distances

$$i_a(t) = i_a(t_a) + \delta_a D_I - (t - t_a), \tag{3.46a}$$

$$i_b(t) = i_b(t_b) + \delta_b D_I - (t - t_b), \tag{3.46b}$$

where $\delta_j \geq 0$ is the number of times packet *j* is deflected in $[t_j,\ t)$. Two contending packets are found at the same node, and if destined for a single node, must have identical destination distance, i.e.

$$i(t_a) + \delta_a D_I - (t - t_a) = i(t_b) + \delta_b D_I - (t - t_b).$$

Thus

$$\left|\ (t_a + i(t_a)\ ) - (t_b + i(t_b)\ )\ \right| = \left|\ \delta_b - \delta_a\ \right| D_I,$$

hence packets *a* and *b* must belong to the same admission class. □

The significance of the above fact is that for a single destination demand on a strongly connected network, the destination distance of at least one packet in *each nonempty admission class* decreases in each time slot. This observation allows us to apply the evacuation time bound of *Lemma 3.3* to each admission class separately. While the above fact was developed for continuous packet admissions, it applies to batch admissions as well.

We now introduce a model for a single destination *batch* demand in a "go-back-d" network. Consider a regular network of degree *k*. Suppose that each node has a 1 packet capacity buffer for each incident link (i.e., each node can hold *k* packets). Note that the buffers at all nodes (other than the destination) may be viewed collectively as a "distributed queue" with combined capacity of $k(|V| - 1)$ packets. We may logically partition this "queue" into D_I parallel queues of packets, one corresponding to each admission class. The capacity of the l^{th} queue is

$$C_l \triangleq k \sum_{j: l+1+jD_I \leq L} R_{l+1+jD_I} \qquad l = 0,\ldots,D_I - 1, \tag{3.47}$$

where

$$R_i \quad \triangleq \quad \textit{total number of nodes at destination distance } i.$$

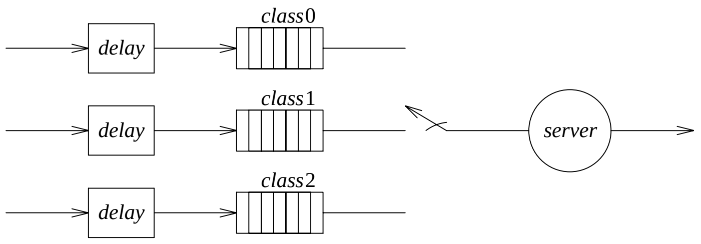


Figure 3.1: The "parallel queue" model for a regular "go-back-2" network of degree *k*. Newly admitted packets suffer a variable delay equal to their initial destination distance, and are then admitted to the queue corresponding to their admission class. At least one packet from each nonempty queue is serviced (i.e., reaches the destination) every D_I time slots.

In our model, a packet admitted to the network at time t_0 with destination distance $i(t_0)$ is initially delayed for $i(t_0)$ time slots before being admitted to the "queue" for its admission class. The D_I queues are served in round robin fashion by a "server," with service corresponding to absorption at the destination node. At least 1 packet in each nonempty queue is serviced every D_I time slots. A deflection represents a reordering of service to packets *within a single queue*. Thus, the network evacuation time is determined by the distribution of packets among the queues. In general, evacuation time will be decreased by insuring that admitted packets are distributed evenly among the queues. We summarize our result for batch demands in the following proposition.

Proposition 3.2: Suppose a batch of N^0 packets is admitted to a "go-back-d" network, with all packets destined for a single node. Let N_j^0 be the number of class *j* packets initially admitted. The evacuation time of class *j* packets (T_j) satisfies

$$T_j \leq L + D_I \,(N_j^0 - 1), \qquad j = 0,\ldots,D_I - 1. \tag{3.48}$$

The maximum evacuation time of any admitted packet is

$$T \leq L + D_I \,(\max_j N_j^0 - 1) \leq L + D_I \,(\max_j C_j - 1). \tag{3.49}$$

Discussion: In many regular networks for which deflection routing has been proposed [13, 34] each of the D_I "queues" in our model has approximately the same capacity. For example, the binary n-cube can admit a batch of $N^0 \leq n(2^n - 1)$ packets destined to a single node. This batch may be partitioned among the 2 "queues" having capacities $C_0 = n\,2^{n-1}$ and $C_1 = n(2^{n-1} - 1)$ packets. It follows that the evacuation time of a fully loaded binary n-cube satisfies

$$T \leq n - 2 + n2^n, \tag{3.50}$$

which is approximately half of (H1).

The "parallel queue" model for "go-back-d" networks enables us to extend our results to include *continuous* admissions to a single destination. However, in the case of continuous admissions we must ensure that our choice of packet-to-link assignment rule avoids livelock [9]. We now specify a rule for assigning packets of different ages to links. We describe a *longest-in-network* priority packet-to-link assignment rule, as follows. At each node, arrange packets in nondecreasing order by the time each has spent in the network. Beginning with the packet(s) longest in the network, assign each packet to a preferred outgoing link (i.e., along a shortest path) if available. If unavailable, assign the packet to a randomly chosen available outgoing link.

In terms of our network model, a "reordering" of service to packets due to a deflection does not permit "younger" packets to take precedence over "older" packets. Note, however, that it is possible for "younger" packets to reach the destination before older packets. Again we use the fact that at least one packet in each nonempty queue is serviced (i.e., reaches the destination) every D_I time slots. We summarize our result in the following proposition.

Proposition 3.3: Suppose packets are admitted continuously and destined for a single node in a "go-back-d" network operating with the *longest-in-network* link assignment rule. Let N^τ (N_j^τ) denote the number of packets (of class *j*) in the network at the end of time slot τ. The maximum transit delay ($T_j(\tau)$) for a class *j* packet admitted during time slot τ satisfies

$$T_j(\tau) \leq L + D_I\ (N_j^\tau - 1) \leq L + D_I\ (C_j - 1), \qquad (3.51)$$

and the maximum transit delay ($T(\tau)$) for any packet admitted during time slot τ satisfies

$$T(\tau) \leq L + D_I\ (\max_j N_j^\tau - 1) \leq L + D_I\ (\max_j C_j - 1). \qquad (3.52)$$

Discussion: In summary, Proposition 3.2 indicates that the evacuation time for fully loaded single destination "go-back-d" networks is approximately $D_I = d + 1$ times smaller than (H1) or (3.44). Proposition 3.3 indicates that maximum transit delay for continuous admissions satisfies the same bound, given a particular choice of assignment rule.

The "parallel queue" model also provides some insight into packet delay in general "go-back" networks. The following discussion may apply to either batch or continous admissions. Suppose admitted packets are destined for a single node, and the deflection distances suffered by packets are not constant. In terms of our model, a deflected packet may now possibly move from one "queue" to another (i.e., a deflection can effectively change the "admission class" of a packet). If the deflections distances suffered by packets are varied, there will be a tendency for packets to distribute evenly among the queues (i.e., "random" jockeying). Indeed, this "randomization" can generally reduce evacuation time or maximum transit delay if certain queues are heavily loaded while others are relatively empty. In the next section we consider one such situation.

Batch Admissions in the "Go-back-all" Network

Consider a batch of N^0 packets admitted to a "go-back-all" network and destined to a single node. Proposition 3.1 indicates that the network evacuates in time

$$T \leq L + L\ (N^0 - 1)\,. \tag{3.53}$$

We now argue that the evacuation time is "likely" to be considerably less. As we will show, a shorter evacuation time is due to randomization introduced by variable packet deflection distances.

In the time slot immediately following a deflection, at least 1 packet is found at destination distance $i = L$. Hence, (3.35) indicates that at least one packet must exit L time slots later. In summary, a deflection at time t guarantees that a packet exits at time $t + L + 1$. Now suppose that in each time slot where there are 2 or more packets in the network the following 2 conditions are met: **a)** at least 1 packet at each destination distance moves one hop closer to its destination, and **b)** some other packet is deflected. Satisfying these conditions would cause the network to evacuate in time

$$T \leq L + (N^0 - 1)\,, \tag{3.54}$$

approximately L times faster than (3.53). Paradoxically, very long delays might be incurred by having too *few*deflections. This suggests that evacuation time can be reduced by implementing a policy where some packets are voluntarily deflected. Of course such a policy might have undesirable properties, such as a tendency to increase mean delay.

Unfortunately, it does not seem possible to guarantee that both conditions can be met in a distributed fashion. However, in time slots where the network is heavily occupied (i.e., $N^t >> 0$) both conditions are likely to be met, assuming the use of a conventional packet-to-link assignment rule. Further, in time slots where the network is lightly loaded, deflections should be relatively few in number, and packets should quickly reach the destination.

3.3.3. Admissions to Multiple Destinations

We next study the evacuation time of a batch of packets, each of which may have a different destination. In a network with a multiple destination demand, packets located at the same node in a time slot can be at different destination distances. In general, evacuation time will be affected by the choice of the rule which assigns arriving packets to outgoing links.

Batch Admissions in the Hypercube

In this section we show that the evacuation time for a batch of N^0 packets admitted to a binary n-cube is bounded by $n + 2(N^0 - 1)$. This result is identical to (H3), but we use a different link assignment rule, and the proof is considerably simpler. We believe that our assignment rule results in higher evacuation times than the rule used to derive (H3). This suggests that (H3) might be a loose evacuation time bound for the rule described in Proposition H3.

Call the following distributed, locally implementable assignment rule the *guaranteed minimum progress* priority rule. At each node in each time slot, let *group i* be the set of packets at destination distance *i*. There are at most *n* such groups. In order from the smallest numbered group to the largest, assign exactly 1 packet from each nonempty group to an outgoing link along a shortest path to its destination. Following this, the assignment of the remaining packets to links is arbitrary. Indeed, all such packets may be purposefully deflected!

Proposition 3.4: Let a batch of N^0 packets be admitted to a binary n-cube. If packets are routed under the *guaranteed minimum progress* priority rule, the evacuation time satisfies

$$T \le n + 2(N^0 - 1). \tag{3.55}$$

Proof: In the binary n-cube, a packet at destination distance *i* has exactly *i* outgoing links from the current node along a shortest path to its destination. Under the ordering of link assignments in the *guaranteed minimum progress* priority rule, the packet chosen from each group to be assigned to an outgoing link will have at least 1 unassigned, shortest path link available. Hence, one packet from each group, or equivalently from each destination distance, moves 1 step closer to its destination. Hence

$$D^t(i) \le N^t(i) - 1 \qquad \textit{when} \qquad N^t(i) \ge 1, \qquad\qquad i = 1,2,\dots,n. \tag{3.56}$$

By inserting (3.56) into the system equations for the hypercube (Eqs. 2.1a with $d = 1$), it is easy to see that condition (3.33) is met. Applying *Lemma 3.3* yields (3.55). □

Batch Delay on Arbitrary Topology

We next derive an evacuation time bound for a batch of N^0 packets, each destined for one of *M* nodes in an arbitrary topology with diameter *L* and deflection index D_I. We begin by ordering the M destinations in an arbitrary fashion, and assume that this ordering is known to each node. Let N_j^0, $j = 1, 2, \ldots, M$, be the number of packets initially admitted and destined for the j^{th} destination, so

$$N^0 = \sum_{j=1}^{M} N_j^0 .$$

We refer to packets destined for node j as *class j* packets. Consider the following priority link assignment rule, which we call the *ordered destination* rule. Let each node order arriving packets by class from the lowest to the highest numbered class. Starting from the lowest numbered class, packets are assigned to a preferred output link, if available. The choice of ordering for packets within a class is arbitrary.

Let T_j be the evacuation time of class j packets, with $T_0 \equiv 0$. Let τ_j denote the time by which *all* classes up to and including the j^{th} class have evacuated, i.e.

$$\tau_j = \min \{ t : N_i^t = 0 , i = 1,2,\ldots,j \}.$$

It follows that $\tau_1 \le \tau_2 \le \cdots \le \tau_M$ and $T_j \le \tau_j$. Let *phase j* refer to the interval $[\tau_{j-1}, \tau_j)$. Note that during phase j, all packets of class $k < j$ have been evacuated. Further, under the *ordered destination* rule, packets of class j can not be deflected by packets of class $k > j$. Hence, we may view each phase as a separate evacuation period for packets sent to a single destination. We may apply the single destination evacuation time bound (3.44) to upper bound the length of each phase:

$$\tau_j \le \tau_{j-1} + L + D_I(N_j^{\tau_{j-1}} - 1) \qquad j = 1,2,\ldots,M, \tag{3.57}$$

$$\le \tau_{j-1} + L + D_I(N_j^0 - 1). \tag{3.58}$$

It follows that the network evacuates in time

$$T = \tau_M = \sum_{j=1}^{M} \tau_j - \tau_{j-1} \tag{3.59}$$

$$\le \sum_{j=1}^{M} L + D_I(N_j^0 - 1)$$

$$= LM + D_I(N^0 - M). \tag{3.60}$$

Remark: Eq. 3.60 is useful since in represents an evacuation time bound which is a function of the number of packet destinations, not just the topology parameters L and D_I. For networks with $D_I = L$ (e.g., "go-back-all" network), (3.60) is identical to (H2).

We next show that by choosing the ordering of destinations appropriately, the multiple destination evacuation time bound (3.60) can be considerably reduced. Indeed, the following proposition indicates that the smallest evacuation time bound is achieved when a walk connecting destinations in order has minimum length.

Proposition 3.5: Let a batch of N^0 packets be admitted to a "go-back" network with deflection index D_I and diameter L. Suppose each packet is destined for one of M destination nodes, and packets are assigned to links by the *ordered destination* rule. Then the N_j^0 packets of the j^{th} class evacuate in time

$$T_1 = \tau_1 \le L + D_I(N_1^0 - 1), \tag{3.61a}$$

$$T_j \le \tau_j \le \max \left[L, \left\{ \max_{k:1 \le k \le j-1} (\tau_k + \rho_{k,j} + D_I) \right\} + D_I(N_j^0 - 1) \right] \quad j = 2,\ldots,M, \tag{3.61b}$$

where

$$\rho_{k,j} \triangleq \textit{distance from } k^{th} \textit{ to } j^{th} \textit{ destination node}.$$

If the destinations are ordered so a walk connecting destinations in order has length P, then the network evacuation time satisfies

$$T \le L + P + D_I\ (N^0 - 1). \tag{3.62}$$

Proof: See Appendix A.

Remarks: To minimize the bound on evacuation time, we choose the ordering of destinations to minimize the walk length P. Note that in many regular networks of $|V|$ nodes it is possible to span all nodes in a walk of length $P = |V| - 1$. For networks with $D_I << L$, (3.62) is a considerable improvement over (H.2). For the n-cube with a single destination demand, (3.62) is identical to (H.3). For multiple destination demands on a fully loaded n-cube with $P = |V| - 1$, (3.62) is approximately a factor of $\frac{n + ½}{n}$ larger than (H.3).

We note that it is somewhat nonintuitive to assign adjacent priority classes to destination nodes which are "close together." We would expect that this strategy might lead to congestion in the vicinity of the highest priority destination node at any time. Alternatively, we would expect few deflections to occur between packets simultaneously moving to nodes which are relatively "far apart". The following simple modification to the *ordered destination* rule exploits this fact, and can reduce the evacuation time (3.62) for a class of "self-routing" networks including the hypercube and the folded shuffle-exchange.

Call 2 destination nodes *complementary* if their binary representations are bit complementary in each bit position (e.g. destinations 11000 and 00111 on a 5-cube). Note that packets destined for complementary destinations will not seek to use the same outgoing link at a node in following shortest paths to their destinations. Hence, packets headed to complementary destinations can not deflect one another. We can exploit this fact by modifying the *ordered destination* rule as follows. Suppose that each destination node has a complement which is also a destination node. Partition the M destination nodes into a set of $M/2$ nodes and a set of their $M/2$ complements. Then assign each node in the second set the same priority class value as its complement in the first set. Hence, the j^{th} phase ($j = 1, 2, ..., M/2$) will now correspond to the evacuation of packets headed for the j^{th}destination node in the ordering and its complement j^*.

Using a single phase to evacuate packets for a node and its complement reduces the bound on evacuation time. Under certain demands (e.g., a permutation), the decrease can be as much as a factor of 2, approximately.

$$T \le L + P + D_I\,(N^0 - 1). \tag{3.62}$$

Continuous Admissions to the Hypercube

In this section we study the maximum transit delay of packets admitted continuously to a binary n-cube. Once again we route packets according to the *guaranteed minimum progress* priority rule, with a small modification to avoid livelock under continuous admissions. As before, at each node in each time slot let *group i* be the set of packets at destination distance *i*. In order from the smallest numbered group to the largest, assign the **oldest** packet (i.e., longest in the network) from each nonempty group to an outgoing link along a shortest path to its destination. (If several packets in a group are equally "old" one is picked arbitrarily). As before, the assignment of all remaining packets to links is arbitrary.

Proposition 3.6: Suppose packets are admitted continuously to a binary n-cube, with packets routed under the *modified* guaranteed minimum progress priority rule (as described above). Let N^τ denote the number of packets in the network at the end of time slot τ. The maximum transit delay ($T(\tau)$) of a packet admitted during time slot τ is

$$T(\tau) \le n + 2(N^\tau - 1). \tag{3.63}$$

The maximum delay of a packet admitted in any time slot (*T*) is bounded by

$$T \le n + 2(n2^n - 1). \tag{3.64}$$

Proof: Consider the N^τ packets in the network at the end of time slot τ. Under our routing scheme, a packet from each nonempty group (i.e., at each destination distance) from among the N^τ moves closer to its destination distance in every time slot. We may view the initial collection of N^τ packets as a "batch" of packets effectively admitted at time τ. Since routing these packets under the guaranteed minimum progress rule assures that we meet condition (3.56), all packets in this "batch" must be routed within a time given by (3.55). Hence, the delay of the longest delayed packet admitted at time τ is bounded in (3.63). □

Discussion: Suppose all packets in a fully populated n-cube are destined for a single node. Then each of nearly $n\ 2^n$ packets must traverse one of the n links incident to the destination node. This indicates that a lower bound on worst case delay is $O(2^n)$. Note that the upper bound on worst case delay (3.64) is only $O(n)$ times larger than this lower bound.

3.4. Summary

We have derived several new bounds on the delay of packets admitted to networks operating with deflection routing. We have considered both single and multiple destination demands. For multiple destination demands, no assumptions were made which limit the number of packets to be routed, nor the distribution of destinations. Indeed, the bounds are derived without independence assumption, and are valid even for demands which may represent a malicious cooperation between network nodes. We have also derived delay bounds for packets admitted continuously to a network. To the author's knowledge, these appear to be the only such results published. These results confirm [32] that with the proper choice of contention rule livelock can be eliminated from deflection routing.

We note that a number of authors have published simulation results for evacuation time and maximum delay for deflection routing on topologies as varied as the shuffle-exchange network [28], the Manhattan Street Network [27, 30] and the hypercube [29, 34]. These simulations all indicate that deflection routing yields remarkably low *mean* transit delay for heavily loaded networks with uniform offered traffic. Indeed, observed worst case transit delays under uniform loads are often only a fraction of the bounds presented here. Less has been reported in the literature regarding worst case delays produced by highly nonuniform demands [9, 17, 20, 21]. It is under highly nonuniform loads that we would expect to observe the most severe transit delays. It can be shown that the bounds presented here are tight when a relatively small number of packets are admitted to a network. It is unknown how tight these bounds are when the number of admitted packets is large, though we would conjecture that they are quite loose.

There appear to be several opportunities for further research on worst case delay. One is to consider limits on the number of admitted packets or the distribution of packet destinations. As an example, Borodin and Hopcroft conjecture that some adaptive routing rule exists which can route a permutation on a binary n-cube in $O(n)$ time [35]. It is unknown whether this can be achieved under the severe buffer restrictions we have assumed. With the exception of such special cases, however, it seems likely that significant improvements to nonprobabilistic worst case bounds on transit delay will be difficult to achieve.

A second opportunity is to consider probabilistic bounds. As an example, Greenberg and Hajek [34] have studied uniform loads on the hypercube. Using an independence assumption, they have developed an analytical bound on packet deflection probability as a function of destination distance. Use of such bounds may be particularly helpful in developing practical estimates of the required capacity of resequencing buffers [21].

CHAPTER IV

THE MANHATTAN STREET NETWORK

4. Introduction

In this chapter we study deflection routing on a particular topology, the *Manhattan Street Network* (MSN). We begin by considering an unbuffered network operating with a uniform offered load. That is, we assume that packets are admitted independently from node to node, and uniformly destined for one of the nodes other than the admitting node. We derive an approximate transit delay distribution for a network of nodes using each of the contention resolution rules introduced in Chapter II. We also consider a routing "optimization" which attempts to reduce deflections by increasing a packet's routing flexibility in future time slots. We then turn to the analysis of buffered networks. In particular we derive approximate delay distributions for a network of nodes each with a 1 packet capacity store-and-forward buffer. Network simulation results are reported and compared with analytical results.

We next consider the challenging problem of the analysis of an MSN with a nonuniform offered load. Here we also permit communications links to have an arbitrary but known propagation delay. Once again we derive approximate delay distributions. In the next section we state several properties of the Manhattan graph which both motivate the design of routing rules and facilitate their analysis.

Properties of the Manhattan Topology

The *NxM* MSN has a corresponding graph $G = (V,E)$ comprising *NM* vertices (nodes) and $2NM$ edges (directed communication links). Let *P* be the diameter of the graph. The Manhattan graph is defined when both N and M are even integers (i.e., $2f \ x \ 2g$ MSN). We identify each node $v \in V$ by a row and column number (i,j). The shortest path distance (in hops) from node (i,j) to node (x,y) is written $sp(i,j,x,y)$. The incoming links to node (i,j) are labeled with the triples (i,j,k) , $k \in \{0,1\}$. Let $R_m(i,j)$ be the set of nodes reachable from node (i,j) in no less than m hops (i.e., $R_m(i,j) = \{(x,y) \ ; \ sp(i,j,x,y) = m \}$, $0 < m \le P$). By symmetry we have $|R_m(i,j)| \equiv |R_m|$. The average shortest path length between two distinct nodes (L_0) is given by (3.1) with $|V| = N \ M$.

We now introduce terminology to describe paths between nodes in the graph; the same terminology is used to describe packets traversing paths. A path *bends* at a node if it follows an incoming row (column) edge and an outgoing column (row) edge. Each node is assigned a *point type* with respect to a destination node as follows: a node is an *option point* if there exists a shortest path from the given node to the destination node using either outgoing edge. If a shortest path to the destination node exists only along 1 outgoing edge, the node is a *critical point.* A *critical bend (straight)* for a shortest path is a critical point (w.r.t. the destination node at the path end) where the path bends (does not bend).

We now state several properties of the MSN graph which have been introduced by other authors.

Property 4.1: Every shortest path between any 2 distinct nodes in G has either 0, 1, 2 or 3 critical bends [27].

We next consider a relationship between directly connected nodes. Consider any node in G which is a critical point with respect to some destination node. The outgo-

Note that we can construct a torus for which N or M are odd integers, however a pair of adjacent rows or columns will be in the same direction.

ing links of the critical point node terminate at 2 distinct nodes. By definition, only one of these 2 nodes is along a shortest path from the critical point node to the destination node. Call this node the *preferred* node.

> **Property 4.2:** The preferred node is exactly one hop closer to the destination node than the critical point node. The node terminating the other outgoing link from the critical point node is either exactly 1 or 3 hops further from the destination node, and is an option point [14].

These observations can be verified rigorously by inspection of the distance function found in the appendix of [15]. Property 4.2 has the following two useful consequences for shortest path deflection routing.

> **Fact 4.1:** Every deflected packet is deflected into an option point [16].

> **Fact 4.2:** A deflection adds either 2 or 4 hops to the deflected packet's path.

A deflection adding 2 hops to a packet's path occurs relatively infrequently under a uniform traffic demand. Two hop deflections are introduced by the MSN wraparound (i.e., the closure of the finite mesh to form a torus) and can occur only for packets that are at least $min\ [\tfrac{1}{2}N,\ \tfrac{1}{2}M]$ hops from their destinations.

Finally, we mention the special case of MSNs having an integer multiple of 4 rows and 4 columns (i.e., $4fx4g$ MSN). These networks have the special property that every deflection adds exactly 4 hops to the deflected packet's path (i.e., no deflection adds 2 hops). Khasnabish [31] has developed simple expressions for $|R_m|$ and L_o for the square MSN.

4.1. Uniform Traffic: Unbuffered MSN

4.1.1. Mean Delay

In Chapter III we showed that the mean transit delay for an MSN with a uniform offered traffic load is given by

$$L = \frac{L_0}{1 - D\, P_D}. \tag{4.1}$$

From (4.1) and Little's result we can easily write the network throughput as

$$\frac{2\, N\, M}{L_0}\, p\, (1 - D\, P_D). \tag{4.2}$$

where *p* is the link utilization. Note that an equation similar to (4.1) holds for a network operating with a *nonuniform* offered load. In this case the numerator would equal the average shortest path distance under the applied traffic demand.

Fact 4.2 indicates that

$$D = \frac{2\,(P_D - P_D^{(4)}) + 4\, P_D^{(4)}}{P_D}, \tag{4.3}$$

where $P_D^{(4)}$ is the probability that an arriving packet suffers a deflection which forces the packet to travel 4 extra hops. For the special case of the $4 fx 4 g$ MSN the probability $P_D^{(4)} = P_D$, and (4.1) reduces to

$$L = \frac{L_0}{1 - 4\, P_D}. \tag{4.4}$$

Eq. 4.4 represents a good approximation to mean delay in an MSN of any size since, as noted earlier, deflections adding 2 hops to a packet's path occur infrequently.

Hence, finding expected delay reduces to finding the average deflection probability P_D. If P_D is unknown, bounds on P_D can be written, and (4.1) or (4.4) can then be used to bound mean delay. We will now demonstrate this for an MSN operating with the *straight* contention rule. To do so, we introduce an additional assumption.

> **Assumption 4.1** (Independence approximation) - Consider the two incoming links to a node in a time slot: **a)** The arrival of a packet on one link is independent of the arrival of a packet on the second link. **b)** The "state information" (i.e., point type, routing preference, remaining destination distance, time in network) of an arrival on one incoming link is independent of the state information of an arrival on the second incoming link.

A deflection occurs when 2 packets arrive to a node in a time slot, both packets are at critical points, and one packet prefers to "bend" while the other prefers to move "straight." Define

$$p \triangleq P[\textit{packet arrives on an incoming link to a node in a time slot}], \tag{4.5}$$
$$P_{cb} \triangleq P[\textit{packet seeks to bend at a critical point upon arrival}], \tag{4.6}$$
$$P_{cs} \triangleq P[\textit{packet seeks to pass straight at a critical point upon arrival}], \tag{4.7}$$
$$p_A \triangleq P[\textit{packet at an option point upon arrival}]. \tag{4.8}$$

Under the *straight* or *random* rules the conditional deflection probability is given by

$$P_D \approx p\ P_{cb}\ P_{cs} = p\ P_{cb}\ (1 - P_{cb} - p_A). \tag{4.9}$$

Eq. 4.9 follows easily by considering 2 "typical" packets arriving to a node in a time slot, and using Assumptions 3.1, 4.1. By symmetry, each of the $2NM$ links has identical utilization. Consequently $p = \frac{K}{2NM}$, where K is the expected number of packets in the network in each time slot.

The probabilities P_{cb} and p_A are well defined, but unknown. However, simple bounds can be developed. For example, under the *straight* rule

$$P_D < P_{cb} < \frac{3}{L} + 2P_D. \tag{4.10}$$

The lower bound follows from the fact that under the *straight* rule, only packets at critical bend points can be deflected. The upper bound holds since a packet can never be more than 3 critical bends from its destination *(Property 4.1),* and a deflection can not introduce more than 2 critical bends in a packet's path.

The parameter p_A is the fractional number of nodes visited that are option points. By Fact 4.1 we have

$$p_A \geq P_D . \tag{4.11}$$

Approximate bounds on p_A in large networks can be produced informally, as follows. On average, a packet following a shortest path spends between half (approximate) and all (approximate) of its hops reaching its destination row or column (or a neighboring row and column with correct sense). The bulk of the remainder of its hops are following that row (column) to the destination node. Due to the alternating directions of neighboring row and column links, the packet visits an option point at nearly every other node visited on the first leg of its trip. No option points are visited on the second leg. Now consider the effect of deflections. At least ¼ of the extra hops due to deflections will be option points (Fact 4.1). Consequently, deflections can not force p_A below ¼. Combining these informal arguments suggests that

$$\tfrac{1}{4} \lesssim p_A \lesssim \tfrac{1}{2} \qquad \textit{(large networks).} \tag{4.12}$$

Equations (4.1, 4.4, 4.9-12) combine to form a variety of bounds for mean transit delay in a network operating with the *straight* contention rule. For instance, by maximizing the RHS of (4.9) with p_A and p fixed we find an upper bound on mean delay given by

$$L \leq \frac{L_0}{1 - p\,(1-p_A)^2} . \tag{4.13}$$

Simulations indicate that this is a tight bound for a small network (e.g., $N = M = 8$) however it requires knowledge of p_A. If p_A is unknown, we can apply our approximate lower bound of ¼. It has been conjectured that modifications to routing that increase the value of p_A will further decrease mean delay. Indeed, increases to p_A do tighten the bound (4.13). However, we will shortly show that increasing the value of p_A does not necessarily decrease delay, and can in fact increase delay in sufficiently small networks.

4.1.2. Delay Distribution

We next present an approach for solving for an approximate packet transit delay distribution for an MSN operating with each of the contention rules described in Chapter II. We begin by characterizing the state of packet arrivals. Let

$$h(n,m) \triangleq \textit{P[packet arrives with slot count = n, distance to destination = m]}. \qquad (4.14)$$

By symmetry the utilization of each link is given by

$$p \triangleq \textit{P[packet arrives]} = \sum_{i,j} h(i,j). \qquad (4.15)$$

Recall that new packets are admitted independently from node to node, and new packet destinations are drawn uniformly from the $NM - 1$ nodes other than the admitting node. Alternatively, new packet destination distances are drawn from the distribution

$$\textit{P[destination distance = m} \mid \textit{new packet]} = \frac{|R_m|}{\sum_{i=1}^{P} |R_i|}. \qquad (4.16)$$

We adopt the convention of assigning newly admitted packets an age (i.e., slot count) of 0 time slots. By stationarity, new packets are admitted at the same rate that packets reach their destinations (and are removed), which equals $\sum_i h(i,0)$. Hence, the probability that a packet arriving to a node is new and at destination distance *m* is

$$h(0,m) = \frac{|R_m|}{\sum_{k=1}^{P} |R_k|} \sum_i h(i,0), \qquad m=1,\ldots,P. \qquad (4.17)$$

Define the conditional deflection probability

$$P_D(n,m) \triangleq P[\textit{packet is deflected} \mid \textit{packet arrives with slot count} = n, \textit{distance to destination} = m\,].$$

We can further partition deflections into those costing 2 and 4 hops, with corresponding conditional deflection probabilities $P_D^{(2)}(n,m)$ and $P_D^{(4)}(n,m)$ satisfying

$$P_D(n,m) = P_D^{(2)}(n,m) + P_D^{(4)}(n,m). \tag{4.18}$$

All possible transitions of age and distance for packets admitted to the $4x6$ MSN are depicted in Figure 4.1.

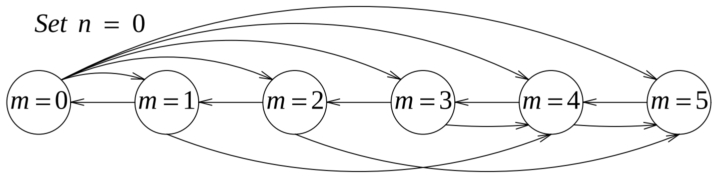


Figure 4.1 - Age and destination distance transitions for the $4x6$ MSN. A packet's age *n* is incremented in each time slot after admission.

These transitions enable us to write a simple recursion for the probability $h(n,m)$ by considering each event that can occur to a packet in a time slot, i.e.,

$$\begin{aligned} h(n,m) = {} & \Big[1 - P_D(n-1,m+1)\Big]\ h(n-1,m+1)\ u[P-1-m] \\ & + \Big[P_D^{(2)}(n-1,m-1)\Big]\ h(n-1,m-1)\ u[m-2] \\ & + \Big[P_D^{(4)}(n-1,m-3)\Big]\ h(n-1,m-3)\ u[m-4], \qquad n=1,2,\ldots \qquad m=0,1,\ldots,P, \end{aligned} \tag{4.19}$$

where $u[x]$ denotes the unit step sequence

$$u[x] \triangleq \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases}$$

The first term on the RHS of (4.19) describes the event that a packet is not deflected, while the following 2 terms describe the events corresponding to a packet deflected distances of 1 and 3 hops, respectively. For the special case of the $4fx4g$ MSN, (4.19) reduces to

$$h(n,m) \;=\; \Big[1 - P_D(n-1,m+1)\Big]\; h(n-1,m+1)\;\; u[P-1-m] \qquad (4.20)$$
$$+\;\Big[P_D(n-1,m-3)\Big]\; h(n-1,m-3)\;\; u[m-4], \qquad n=1,2,... \qquad m=0,1,...,P.$$

If $P_D(n,m)$ is known, we can solve (4.20) for $h(n,m)$, from which we can derive a variety of useful performance measures such as the transit delay distribution

$$P[\text{transit delay} = i] \;\triangleq\; P[\text{packet has slot count } i \text{ given packet arrives to its destination }],$$
$$= \frac{h(i,0)}{\sum_j h(j,0)}, \qquad i=1,2,... \;. \qquad (4.21)$$

Note that (4.19-20) hold without independence assumption, and that a closed form solution for $h(n,m)$ can be obtained for the special case where $P_D(n,m) \equiv$ *constant*. In general, however, $P_D(n,m)$ will be a nontrivial function of *n* and *m*. Yet it is of interest to find a constant *C* for which $P_D(n,m) \le C$, from which we can derive a closed form transit delay distribution which stochastically dominates the actual delay distribution.

We can again use Assumption 4.1 to derive an approximate equation for $P_D(n,m)$ under each of the contention rules specified in Chapter II. For a deflection to occur, two packets must arrive to a node in a time slot, and both packets must be at critical points and have identical routing preference. Define the following events:

$\mathbf{A}\ (\mathbf{A_C})$: *packet A arrives (to critical point) with slot count = n and destination distance = m,*

$\mathbf{B}\ (\mathbf{B_C})$: *packet B arrives (to critical point) with slot count = k and destination distance = l.*

The conditional probability that packet *A* is deflected is

$$P_D(n,m) \;=\; P[\text{packet A is deflected} \mid \mathbf{A}\,] \qquad (4.22)$$
$$=\; \sum_k \sum_l P[\text{packet A is deflected} \mid \text{link contention}, \mathbf{A_c}, \mathbf{B_c}]$$

$$P[\text{link contention} \mid \mathbf{A_c}, \mathbf{B_c}]P[\mathbf{A_c}, \mathbf{B_c} \mid \mathbf{A}].$$

If we define

$$p_A(n,m) \triangleq P[\text{ packet at option point} \mid \text{packet arrives with slot count} = n \text{ , distance to destination} = m], \tag{4.23}$$

$$P_{b|c}(n,m) \triangleq P[\text{ packet seeks to bend} \mid \text{packet arrives at critical point with slot count} = n \text{ , distance to destination} = m], \tag{4.24}$$

then Assumption 4.1 enables us to write

$$P[\mathbf{A_c}, \mathbf{B_c} \mid \mathbf{A}] \approx P[\mathbf{A_c} \mid \mathbf{A}]\, P[\,\mathbf{B_c}] = (1-p_A(m,n))\,h(k,l)(1-p_A(k,l)), \tag{4.25}$$

$$P[\text{ link contention} \mid \mathbf{A_C},\mathbf{B_C}] \approx \Big[P_{b|c}(k,l)(1-P_{b|c}(n,m)) + (1-P_{b|c}(k,l))\,P_{b|c}(n,m)\Big]. \tag{4.26}$$

Eq. (4.22) then becomes

$$P_D(n,m) \approx \sum_k \sum_l (1-p_A(n,m))(1-p_A(k,l))\,h(k,l)\,\Big[P_{b|c}(k,l)(1-P_{b|c}(n,m))\,\delta_s(k,l,n,m) + (1-P_{b|c}(k,l))\,P_{b|c}(n,m)\,\delta_b(k,l,n,m)\Big], \tag{4.27}$$

where we have inserted the conditional probabilities

$$\delta_b(k,l,n,m) \triangleq P[\text{packet A is deflected} \mid \text{contention, } \mathbf{A_c}, A \text{ prefers bend, } \mathbf{B_c}] \tag{4.28}$$

$$\delta_s(k,l,n,m) \triangleq P[\text{packet A is deflected} \mid \text{contention, } \mathbf{A_c}, A \text{ prefers straight, } \mathbf{B_c}] \tag{4.29}$$

The expressions for δ_s, δ_b under each of the contention rules of Chapter II are

$$\begin{array}{ll} \delta_b(k,l,n,m) \equiv \delta_s(k,l,n,m) \equiv \tfrac{1}{2} & \textit{random} \\ \delta_b(k,l,n,m) \equiv 1 \quad \textit{and} \quad \delta_s(k,l,n,m) \equiv 0 & \textit{straight} \\ \delta_b(k,l,n,m) \equiv \delta_s(k,l,n,m) \equiv I_{\{k>n\}} + \tfrac{1}{2} I_{\{k=n\}} & \textit{slot count} \\ \delta_b(k,l,n,m) \equiv \delta_s(k,l,n,m) \equiv I_{\{m>l\}} + \tfrac{1}{2} I_{\{l=m\}} & \textit{destination distance.} \end{array} \tag{4.30}$$

The probabilities $P_{b|c}(n,m)$ and $p_A(n,m)$ are unknown, in general. One approach that may be used is to measure these probabilities in a simulation. Alternatively, one may seek either approximations or bounds. Unfortunately, it appears that tight bounds and good approximations are a function of several parameters, including network size and utilization. Nonetheless, we have had success by fixing those parameters and using bounds and approximations.

We now briefly illustrate one such approach. Suppose we introduce the following 3 simplifying assumptions.

Assumption 4.2 - A packet *m* hops from its destination node is equally likely found at any of the nodes that are *m* hops from the destination node.

Assumption 4.3 - The probability that a packet is at an option point depends only on its destination distance and not its slot count, i.e.,

$$p_A(n,m) \approx P[\textit{ packet at option point} \mid \textit{packet arrives with distance to destination} = m] \equiv p_A(m).$$

Assumption 4.4 - The probability that a packet prefers to bend upon arrival to a critical point is independent of its destination distance and its slot count (i.e., $P_{b|c}(n,m) \equiv P_{b|c}$).

Discussion: The above assumptions enable us to approximate the probability that link contention occurs. It is difficult to judge the accuracy of these approximations, though simulations (see Section 4.3) suggest that the shape of the delay distributions is largely determined by the details of the contention resolution rule (i.e., δ_b, δ_s) and is less affected by approximations to the link contention probability.

Assumptions 4.2-3 permit us to determine the probability that a packet is at an option point given its destination distance. To do so, we select any node and call it our destination node. Then we simply count the fractional number of nodes at distance *m* from our destination node which are option points. That is,

$$\begin{aligned} p_A(m) &= P[\textit{packet at option point} \mid \textit{packet arrives to a node m hops from its destination}], \\ &= \frac{1}{|R_m|} \sum_{(i,j) \in R_m} I_A[(i,j)], \end{aligned} \tag{4.31}$$

where we define

$$I_A[(i,j)] \triangleq \begin{cases} 1 & \textit{if } (i,j) \textit{ is an option point} \\ 0 & \textit{otherwise.} \end{cases}$$

We may then approximate (4.27) as

$$P_D(n,m) \approx 2P_{b|c}(1-P_{b|c})(1-p_A(m))\,\psi(n,m), \qquad n=0,1,... \quad m=1,...,P, \tag{4.32}$$

where

$$\psi(n,m) = \begin{cases} \tfrac{1}{2}p(1-p_A) & \textit{straight or random,} \\ \sum\limits_j (1-p_A(j)) \left[\sum\limits_{i>n} h(i,j) + \tfrac{1}{2}\,h(n,j)\right] & \textit{slot count,} \\ \sum\limits_i \left[\sum\limits_{j<m} (1-p_A(j))\,h(i,j) + \tfrac{1}{2}\,(1-p_A(m))\,h(i,m)\right] & \textit{destination distance,} \end{cases} \tag{4.33}$$

and

$$p_A = \sum_{m=1}^{P} p_A(m) \frac{\sum\limits_n h(n,m)}{\sum\limits_m \sum\limits_n h(n,m)}. \tag{4.34}$$

Approximations to $P_{b|c}$ can be inserted in (4.32) to approximate deflection probability. One simple, useful bound is $P_{b|c}(1 - P_{b|c}) \le \tfrac{1}{4}$. This bound is fairly tight for a small network (e.g., $N = M = 8$), and substituting it in (4.32) yields

$$P_D(n,m) \le \tfrac{1}{2}\,(1-p_A(m))\,\psi(n,m). \tag{4.35}$$

The value of $P_{b|c}$ decreases with increasing network size, so alternate bounds on $P_{b|c}$ must be developed for larger networks (e.g., derived as was done for (4.10)).

We found that the system of equations comprising (4.15-17), (4.20), (4.30-31), and (4.33-35) are easily solved iteratively for $h(n,m)$, and then the approximate delay distribution (4.21). In Section 4.3 we will compare one such numerical result to a simulation result.

4.1.3. A Routing "Optimization"

Recall that p_A is the fractional number of nodes that are option points visited by packets. Several researchers have conjectured that transit delay can be decreased by modifying routing rules so that the value of p_A increases. The accompanying reasoning is that a "typical" packet frequenting more option points will have less opportunity to be deflected.

The desired routing modification is as follows. If a packet arrives to an option point, route the packet along the outgoing link which will maximize the probability that the packet will visit option points in future time slots. Unfortunately, using only local information, it is not possible in general to determine the output link which will increase the likelihood a given packet visits option points in future time slots. Hence, heuristics have been suggested to accomplish this objective, on the average.

In mesh networks such as the MSN, the commonly proposed heuristic reduces to forcing a packet to follow a "diagonal." Badr and Podar [36] have formalized this notion into the Z^2 or *zig-zag* routing, and have shown this to be optimal on certain mesh networks. Under their "one packet" model, a packet takes a random walk to its destination and with each hop is subject to an independent deflection w.p. p. Weller and Hajek [37] have noted that this scheme is not quite optimal for the MSN. Nontheless, since an optimal scheme has not been identified, we consider the effect of this "diagonal" routing optimization, which we describe precisely as follows.

In each time slot consider each packet which is at an option point and is not deferring to a second packet. Let T be a transformation which relabels all network nodes so that the given packet's destination is relabeled to node $(0,0)$, with the desti-

Of the nodes at a given distance to a chosen destination node in a bidirectional mesh, nodes on the diagonal have the maximum number of distinct shortest paths.

nation node's outgoing links pointing in direction of increasing row and column number. Suppose that T relabels the given packet's current location to (i_c, j_c). Then, route the packet along the outgoing column link if $i_c < j_c$. If $i_c > j_c$, route the packet along the outgoing row link. If $i_c = j_c$, use a fair coin toss to assign the packet to one of the outgoing links.

We will report the result of a simulation of this routing optimization in Section 4.3. We will now argue, however, that increasing the value of p_A does not necessarily reduce mean delay. From (4.6-8) we have

$$p_A + P_{cb} + P_{cs} = 1. \tag{4.36}$$

Hence, an increase in p_A must be offset by a decrease in either P_{cb} or P_{cs}. However, recall that deflection probability is approximately given by

$$P_D \approx p\, P_{cb}\, P_{cs} = p\, P_{cb}\, (1 - P_{cb} - p_A). \tag{4.37}$$

Eqs. 4.36-37 indicate that deflection probability is not necessarily monotonically decreasing with an increase in p_A. Hence, it might be possible to visit more option points on average, yet not decrease deflection probability. In this case, though the number of critical points visited by packets is lessened, it is more likely that packets will be deflected when visiting critical points. Simulations show that this does indeed happen for small networks.

4.2. Uniform Traffic: Buffered MSN

We next consider the performance of the buffered MSN with a 1 packet capacity transit buffer at each node. Under the assumed independence of transit buffer arrivals and departures (Assumptions 3.2-3), expected delay is approximately given by (3.28) with $D = 4$.

We now find an approximate solution for the transit delay distribution, under Assumptions 3.2-3 and 4.1. Define the following probabilities:

$$g(n,m) \triangleq P[\textit{packet in buffer with slot count = n, distance to destination = m }], \quad (4.38)$$

$$z \triangleq P[\textit{packet in buffer}] = \sum_n \sum_m g(n,m), \quad (4.39)$$

$$e \triangleq P[\textit{packet exits buffer} \mid \textit{packet in buffer}]. \quad (4.40)$$

We continue to define $h(n,m)$ and $P_D(n,m)$ as in (4.14) and (4.18). The effective utilization is then

$$q = P[\textit{packet arrives on link }] = \sum_n \sum_m h(n,m), \quad (4.41)$$

and by Lemma 3.2 we have

$$q = p - \tfrac{1}{2} z. \quad (4.42)$$

Analogous to our development of a recursive equation for an unbuffered network, we describe the possible transitions of an admitted packet's age and distance in a buffered network. We write the following 2 equations by considering each event that can happen at a node in a time slot:

$$\begin{aligned} h(n,m) \approx\ & \Big[1-P_D(n-1,m+1)\Big]\ h(n-1,m+1)\ u[P-1-m] & (4.43)\\ &+ \Big[P_D^{(2)}(n-1,m-1)\Big]\ h(n-1,m-1)\ z\ u[m-2] \\ &+ \Big[P_D^{(4)}(n-1,m-3)\Big]\ h(n-1,m-3)\ z\ u[m-4] \\ &+ \tfrac{1}{2}\ \Big[g(n-1,m+1)\Big]\ e\ u[P-1-m], \qquad n=1,2,\ldots \quad m=0,1,\ldots,P, \\ g(n,m) \approx\ & 2\ \Big[P_D(n-1,m)\Big]\ h(n-1,m)\ (1-z) & (4.44)\\ &+ \Big[g(n-1,m)\Big]\ (1-e), \qquad n=1,2,\ldots \quad m=1,2,\ldots,P. \end{aligned}$$

The first term on the RHS of (4.43) describes a packet that is not deflected. The second and third terms describe a packet misdirected 1 or 3 hops further from its destination, respectively, due to a full transit buffer when link contention occurred. The final term describes the event in which a packet in the transit buffer exits. The first term on the RHS of (4.44) is the probability a deflected packet is admitted to the (empty) transit buffer, and the second term is the probability a packet does not exit a transit buffer. The factor of 2 in the first term, as well as the factor of ½ in the final term of (4.43), are normalizing factors, since there are twice as many links as buffers.

Once again we can numerically solve the system of equations comprising (4.17), (4.30-31), (4.33-35), and (4.41-44) for an approximate transit delay distribution.

4.3. Simulation Results

A Monte Carlo simulation was written to measure network statistics in both buffered and unbuffered networks operating under a uniform traffic load. We simulated the operation of unbuffered Manhattan networks under each contention rule of Chapter II. We also simulated each contention rule supplemented with the routing optimization described in Section 4.1.3. For buffered networks, identical capacity transit buffers were added to each node. We implemented only the *straight* contention resolution rule, supplemented with the buffer management described in Chapter II. Experiments were performed for buffer capacities of $C = 1, 2,$ 4 and 20 packets. The 20 packet capacity was chosen to approximate an infinite capacity buffer; such a buffer was never observed to fill.

We simulated several network sizes ranging from $4x4$ to $64x64$. For each network size, link utilization was set at $p = .125i$, $i = 1,2,...,8$. For each utilization, the number of packets in the network was kept constant at $K = p2NM$. In addition, a "zero-deflection model" case was also run by admitting exactly 1 packet ($K = 1$). Statistics were collected on both a network and a packet level. Each statistic we report is an average of 20 separately-seeded, premixed batches of 10,000 time slots.

4.3.1. Results: Unbuffered Network

As reported in [27], shortest path deflection routing with each contention resolution rule produces low expected delay for all link utilizations $0 < p \leq 1$. Simulation results for an $8x8$ network operating with *straight* and *slot count* contention resolution are compared in Table 4.1.

straight						
p	p_A	*L*	σ^2	P_{cb}	P_D	*Max. delay*
0.0078	0.248	5.02	3.59	0.302	0.000	9
0.1250	0.257	5.35	5.19	0.290	0.016	23
0.2500	0.268	5.72	7.33	0.278	0.031	34
0.3750	0.280	6.17	10.61	0.265	0.047	42
0.5000	0.293	6.67	15.44	0.251	0.063	45
0.6250	0.303	7.17	19.72	0.239	0.075	54
0.7500	0.315	7.72	25.97	0.227	0.088	65
0.8750	0.326	8.33	33.78	0.214	0.100	76
1.0000	0.337	8.99	43.19	0.203	0.111	86

slot count						
p	p_A	*L*	σ^2	P_{cb}	P_D	*Max. delay*
0.0078	0.246	5.00	3.53	0.303	0.000	9
0.1250	0.263	5.44	4.98	0.300	0.020	15
0.2500	0.276	5.81	6.18	0.299	0.034	17
0.3750	0.287	6.14	7.22	0.298	0.046	19
0.5000	0.300	6.60	8.70	0.297	0.060	20
0.6250	0.312	7.06	10.12	0.296	0.072	20
0.7500	0.324	7.57	11.76	0.294	0.084	23
0.8750	0.336	8.13	13.58	0.291	0.096	23
1.0000	0.347	8.71	15.31	0.288	0.106	25

Table 4.1 - A comparison of simulation results for an 8x8 MSN operating with the *straight* and *slot count* contention resolution rules. The columns labeled σ^2 and *Max. delay* report the delay variance and the delay of the longest delayed packet observed during the simulation run of 146778 packets, respectively.

The table shows that the fractional number of nodes visited that are option points (p_A) increases with utilization. This behavior was observed with every contention rule we studied without routing optimization. The table also indicates that delay moments grow monotonically with link utilization. To simplify reporting results, we will frequently present results only for fully loaded networks (i.e., $p = 1$).

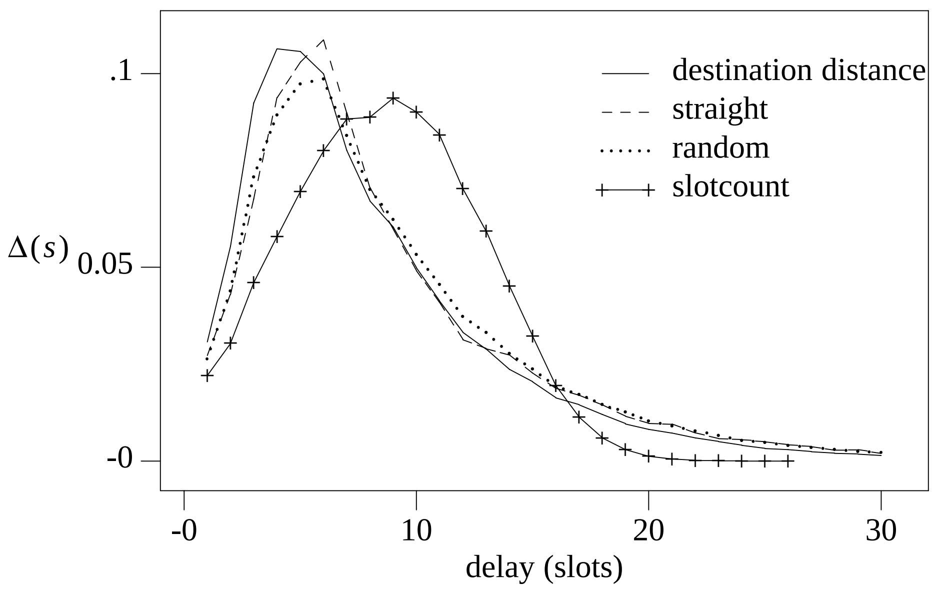


Figure 4.2 - Delay histograms for an unbuffered $8x8$ MSN operating with each contention rule.

In Figure 4.2 we present the transit delay histograms observed in an $8x8$ network operating with each contention rule. For this network size, the *destination distance* rule was found to have the lowest mean delay. As shown, the tail of the delay distribution for a network using the *slot count* rule is dramatically reduced in comparison to the other rules. Both delay variance and the maximum observed delay are lowest under this rule.

Figure 4.3 presents a comparison of a simulation transit delay histogram and an analytically derived dominating transit delay distribution. The results are for a fully utilized $8x8$ network operating with the *slot count* rule. The analytical distribution was derived using the system of equations comprising (4.11), (4.13), (4.24), and (4.26-27).

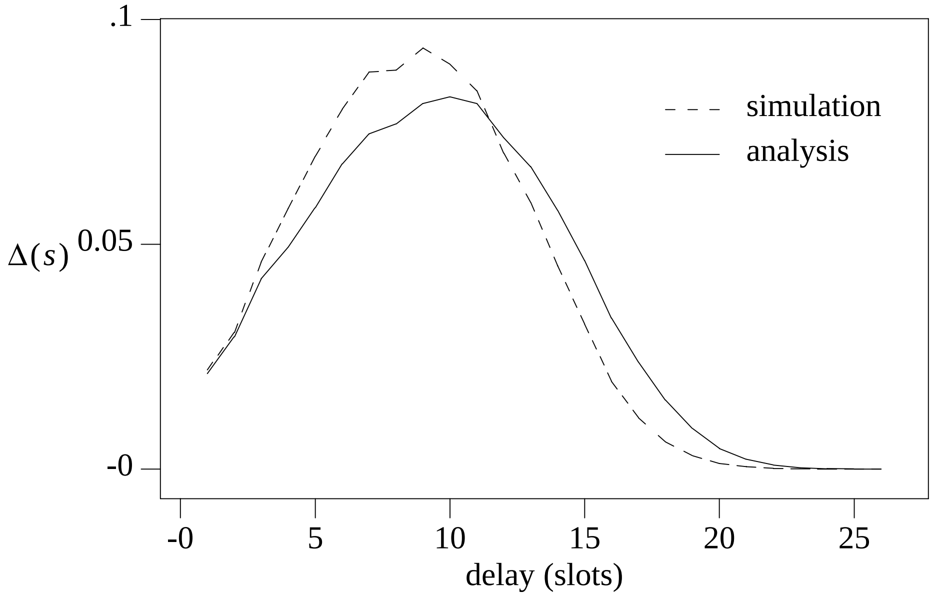


Figure 4.3 - Transit delay histogram in a fully utilized $8x8$ MSN operating with the *slot count* contention rule and an analytically derived dominating distribution.

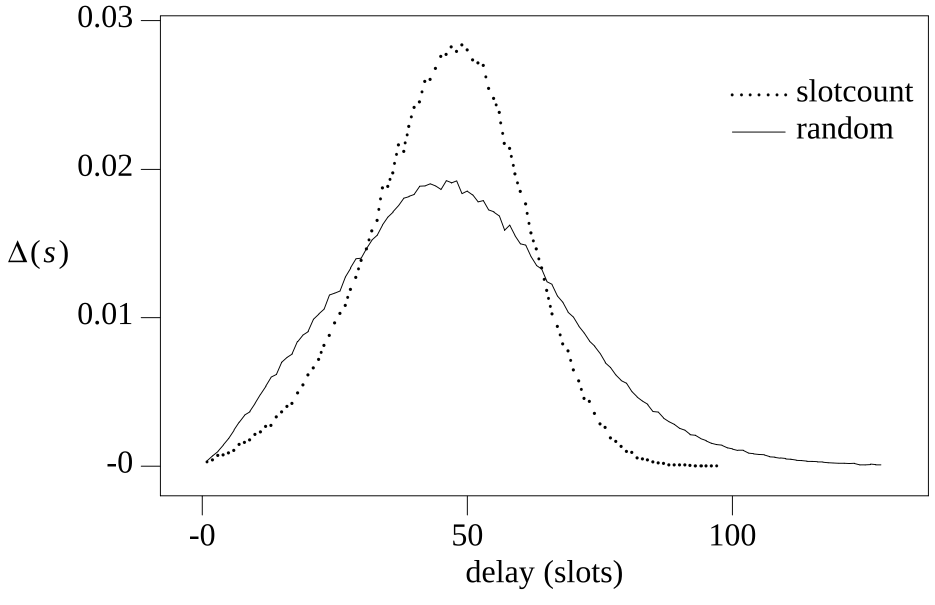


Figure 4.4 - Simulation transit delay histograms for an unbuffered $64x64$ MSN operating with the *random* and *slot count* rules.

In Table 4.2 we present simulation data for several size networks using the *random* contention rule with and without routing optimization. We see that routing optimization does indeed help packets visit a higher fraction of option points. However, relatively little improvement in mean delay or delay variance accompanies this increase. Indeed, in the $8x8$ MSN the routing optimization actually increases delay. As network size increases the benefit of the routing optimization increases.

		w/o optimization				w/ optimization			
NxM	p	L	P_D	σ^2	p_A	L	P_D	σ^2	p_A
8x8	z.d.	5.01	0.0000	3.56	0.246	5.01	0.0000	3.55	0.341
	0.5	6.65	0.0613	14.34	0.291	6.69	0.0626	14.79	0.355
	1.0	9.06	0.1117	41.71	0.335	9.10	0.1123	42.32	0.367
16x16	z.d.	9.04	0.0000	11.45	0.250	9.02	0.0000	11.65	0.408
	0.5	11.76	0.0584	33.71	0.284	11.58	0.0556	31.75	0.374
	1.0	16.53	0.1138	105.34	0.321	16.26	0.1116	100.03	0.354
64x64	z.d.	32.98	0.0000	174.31	0.300	33.12	0.0000	172.00	0.527
	0.5	37.24	0.0292	229.46	0.317	36.21	0.0229	213.65	0.459
	1.0	47.78	0.0781	452.55	0.352	45.33	0.0688	388.58	0.399

Table 4.2 - Simulation results for 3 different size networks operating with the *random* contention rule with and without routing optimization. Performance statistics are compared when a single packet is admitted (i.e., *zero deflection* model) and for link utilizations $p = 0.5,\ 1.0$.

4.3.2. Results: Buffered Network

A summary of simulation results for an $8x8$ MSN using the *straight* rule with transit buffers of different capacities is shown in Table 4.3.

As expected, the lowest delay is associated with the largest transit buffer capacity. However, the mean delay performance of the unity capacity buffer case is only comparable to that obtained with an equally loaded, unbuffered network using the *destination distance* contention rule.

Worst case delay for even the largest capacity buffer case exceeds that of an unbuffered network using the *slot count* contention rule. This observation suggests that it may not be worth the additional complexity to implement

C	p	q	L	σ^2	T_b	Max. Delay	Max. Buffer Stay
1	1.000	.781	7.89	32.35	5.29	82	72
2	1.000	.730	7.26	23.09	4.39	68	66
4	1.000	.709	7.11	19.91	4.33	56	54
20	1.000	.706	7.10	19.47	4.35	45	37

Table 4.3 - Simulation results for an 8x8 MSN with buffers and the *straight* rule. The column labeled C indicates each node's transit buffer capacity. The number of packets in the MSN was set at $2NMp = 128$ ($p = 1.0$, $N = M = 8$). The effective link utilization, labeled q, is reduced from the utilization p that would correspond to the same number of packets admitted to an unbuffered MSN. T_b indicates the expected queueing time spent by a packet admitted to a transit buffer. The longest observed transit delay and transit buffer visit are also reported.

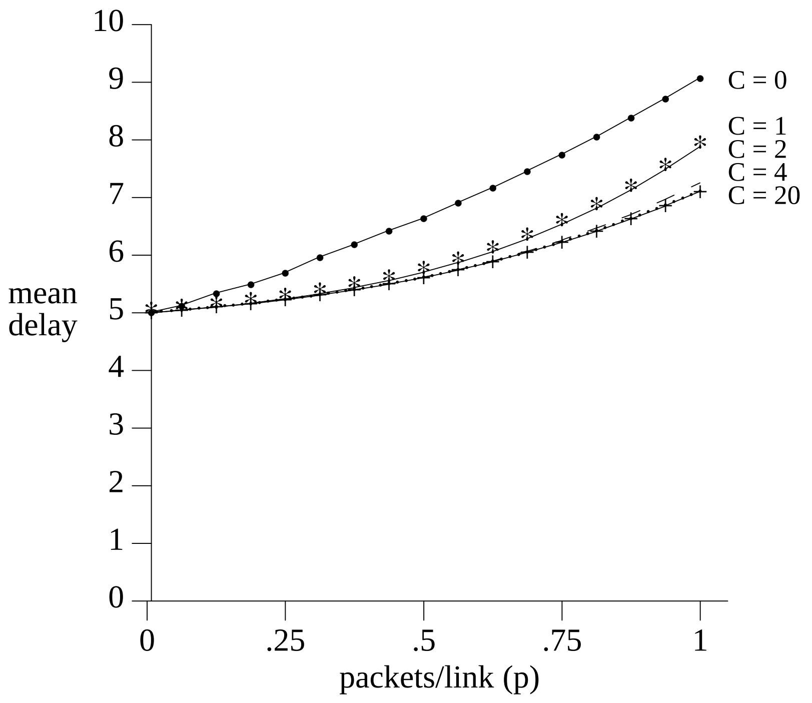


Figure 4.5 - Mean delay for an $8x8$ MSN operating with the straight rule and different capacity transit buffers.

buffers in an environment where propagation delay is negligible. At high load, the expected time a packet spends in a transit queue (if deflected and not misdirected) is observed to exceed 4 time slots. This indicates that a more complex buffer management strategy (e.g., one where certain packets are

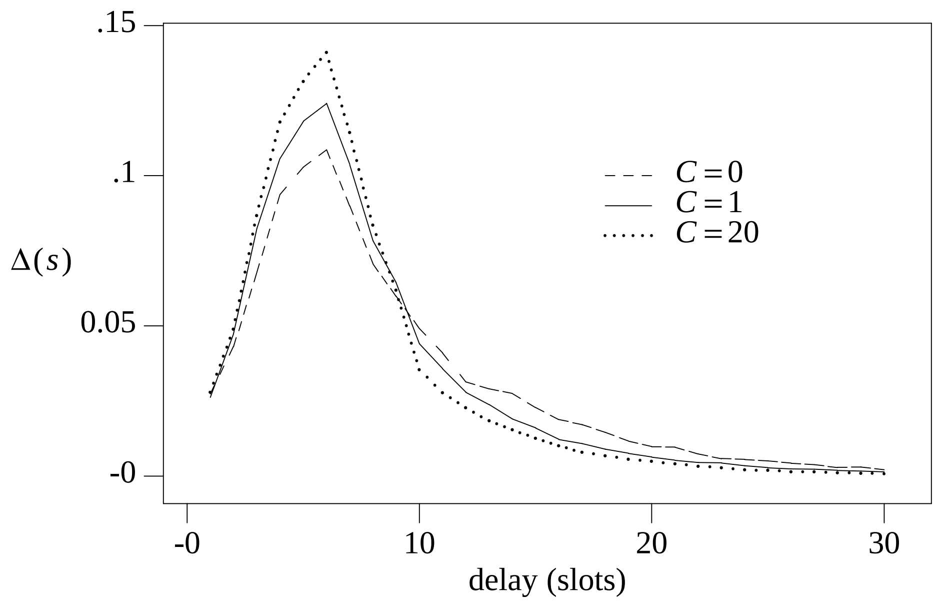


Figure 4.6 - Simulation delay histograms for an $8x8$ MSN operating with the straight rule and transit buffers of capacity $C = 0,$ 1, and 20.

misdirected rather than queued) can likely further reduce delay.

4.4. Nonuniform Traffic

In this section we analyze the performance of an MSN operating with a general, time-homogeneous offered traffic load. We also consider a less idealized node and network model than that introduced in Chapter II. Again we consider a time slotted network with fixed length slots, however we now take slots to have duration τ_s. We continue to assume that communication links are error-free, however we will assume that these links are characterized by an arbitrary but known propagation delay.

We begin by discussing node operation in the absence of propagation delay. At the start of a time slot, each node independently generates a new packet to be admitted to the network in each time slot with probability

$$t_{f,g}(x,y) \triangleq P[\textit{node } (f,g) \textit{ generates a new packet to admit destined for } (x,y)] \quad (4.45)$$

A node generates at most 1 packet in each time slot. We assume that a node does not generate packets destined for itself (i.e., $t_{f,g}(f,g) \equiv 0$). As before a node attempts to place a newly generated packet in an input link buffer. The new packet is *blocked* (and lost) if both input link buffers contain packets (i.e., link arrivals). If only one input link buffer contains a packet and a new packet is generated, the new packet is placed in the unoccupied link buffer. If neither input link buffer is occupied and a new packet is generated, a fair coin toss assigns the new packet to either input link buffer. Hence, top priority is assigned to routing packets in transit rather than admitting new packets. The routing algorithm is then executed. We again consider the contention resolution rules introduced in Chapter II. The desirable performance characteristics of the *slot count* rule suggest that it should be our focus.

We now consider modifications to node operation required in the presence of propagation delay. If propagation delay causes a packet to arrive after the start of a time slot, the packet is held in the input link buffer until the beginning of the next slot. Hence, the required input link buffer capacities must be increased to at most 2 packets. Otherwise, node operation is as described above, with the packets to be "switched" in the current time slot being those packets at the head of the input link buffers.

4.4.1. Delay Analysis

Define the following probabilities for packets arriving on the incoming links and departing on the outgoing links of node (f,g) in a time slot:

$$r_{f,g}(x,y,s) \triangleq P[\textit{packet with age } s \textit{ destined for } (x,y) \textit{ arrives on the input row link of } (f,g)], \tag{4.46a}$$

$$c_{f,g}(u,v,l) \triangleq P[\textit{packet with age } l \textit{ destined for } (u,v) \textit{ arrives on the input column link of } (f,g)], \tag{4.46b}$$

$$r^{o}_{f,g}(x,y,s) \triangleq P[\textit{packet with age } s \textit{ destined for } (x,y) \textit{ departs} \tag{4.46c}$$

$$\text{\textit{on the output row link of} } (f,g)\,],$$

$$c^o_{f,g}(u,v,l) \triangleq P[\text{\textit{packet with age} } l \text{ \textit{destined for} } (u,v) \text{ \textit{departs on the output column link of} } (f,g)\,]. \tag{4.46d}$$

We now present the key simplifying independence approximation: *at each node in each time slot the states of arrivals on the incoming row and column links are independent, and are independent of the generation of any new packets, i.e.*

$$\begin{aligned} &P[\text{ \textit{packet with age} } s \text{ \textit{destined for} } (x,y) \text{ \textit{arrives on input row link of} } (f,g),\\ &\quad \text{\textit{packet with age} } l \text{ \textit{destined for} } (u,v) \text{ \textit{arrives on input column link of} } (f,g),\\ &\quad \text{\textit{packet destined for} } (a,b) \text{ \textit{is generated at} } (f,g)\,]\\ &\quad = r_{f,g}(x,y,s)\; c_{f,g}(u,v,l)\; t_{f,g}(a,b). \end{aligned}$$

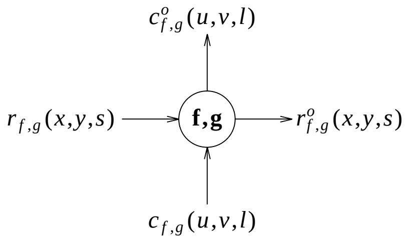


Figure 4.7 - Packet arrival and departure probabilities at node (f,g).

We may easily write the transit delay distribution for packets destined to node (x,y):

$$\begin{aligned} \Delta_{x,y}(s) &\triangleq P[\text{\textit{packet has age} } s \mid \text{\textit{packet arrives to destination} } (x,y)\,]\\ &= \frac{r_{x,y}(x,y,s) + c_{x,y}(x,y,s)}{\sum\limits_s r_{x,y}(x,y,s) + c_{x,y}(x,y,s)}. \end{aligned} \tag{4.47}$$

Other performance measures are also easily written. We next develop a recursion to solve for the link arrival probabilities. By considering each event that can occur at a node in a time slot and using the above independence approximation, we may write equations for the output link probabilities of node (f,g) in terms of its input link and new arrival probabilities:

$$r^o_{f,g} = F(r_{f,g},\, c_{f,g},\, t_{f,g}), \tag{4.48a}$$

$$c^o_{f,g} = G(c_{f,g},\, r_{f,g},\, t_{f,g}). \tag{4.48b}$$

The specific expressions for Eq. 4.40 for nodes operating under each of the contention rules we studied are found in Appendix B.

Next we relate the output link probabilities of a node and the input link probabilities of directly connected nodes. Let $(f,\tilde{g})$ and $(\tilde{f},g)$ be the nodes terminating the outgoing row and column links of node (f,g), respectively. Let $\{\chi\}_K$ denote the integer χ modulo *K*. The Manhattan graph adjacency matrix can be determined from Table 4.4.

(f,g)	$(\tilde{f},g)$	$(f,\tilde{g})$
even, even	$(\{f+1\}_N,g)$	$(f,\{g+1\}_M)$
even, odd	$(\{f-1\}_N,g)$	$(f,\{g+1\}_M)$
odd, even	$(\{f+1\}_N,g)$	$(f,\{g-1\}_M)$
odd, odd	$(\{f-1\}_N,g)$	$(f,\{g-1\}_M)$

Table 4.4 - The adjacency table for the *NxM* MSN.

Let the propagation delay on the output row and column links of (f,g) be designated $\delta^r_{f,g}$ and $\delta^c_{f,g}$. Non-zero propagation delay causes the input link probabilities of downstream nodes to be a delayed version of the output link probabilities of upstream nodes:

$$r_{f,\tilde{g}}(x,y,s) = r^o_{f,g}(x,y,s - \left\lceil \delta^r_{f,g} / \tau_s \right\rceil), \tag{4.49a}$$

$$c_{\tilde{f},g}(x,y,s) = c^o_{f,g}(x,y,s - \left\lceil \delta^c_{f,g} / \tau_s \right\rceil). \tag{4.49b}$$

Network behavior is approximately described by Eqs. 4.48-49, which may be solved iteratively. In the following sections we solve these equations for simple but interesting traffic demands.

4.4.2. Example: Updating a Shared Resource

Consider a $4x4$ MSN with zero propagation delay. Let each node admit packets destined for node $(2,2)$, except for the destination node itself. Each of the other 15 nodes independently generate packets at rate 1/15. Solving (4.48-49) iteratively for a network operating with the *slot count* rule produces the approximate transit delay distribution and link utilizations found in Figures 4.8 & 4.9.

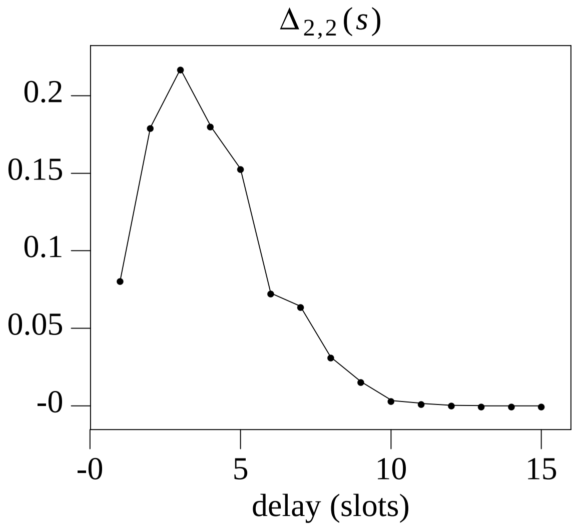


Figure 4.8 - Transit delay distribution (*mean* = 3.91, *standard deviation* = 1.95) at the destination node (2,2).

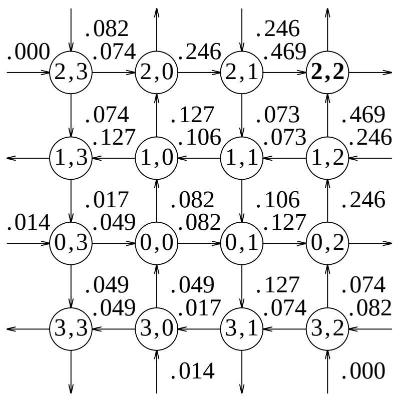


Figure 4.9 - Row and column link utilizations yield an average blocking probability of $8.3x10^{-4}$.

4.4.3. Example: Routing around a "hot spot"

Deflection routing naturally forces packets to circumvent congested regions of the network. We now demonstrate how packets are redirected around a congested link (i.e., "hot spot"). A similar example is considered in [17]. Consider the simple demand of Figure 4.10. Exactly 2 sources are active, nodes $(0,0)$ and $(7,2)$. Each generates 1 packet in each time slot to send to a distinct destination in an $8x8$ MSN with zero propagation delay. Under this traffic demand, solving (4.48-49) for nodes operating with the *slot count* rule yields the approximate link utilizations of Figure 4.11. The utilizations presented are those obtained after 32 iterations, starting with an initial condition for Eq. 4.48 corresponding to an empty network (i.e., $r_{f,g} \equiv 0$, $c_{f,g} \equiv 0$ for all (f,g)).

A number of observations are noteworthy. First, in steady state, all packets destined for node $(0,4)$ are deflected at least once. This occurs since each newly admitted packet from source node $(7,2)$ is deflected at node $(0,2)$ by an "older" packet arriving on the node’s incoming row link. Second, deflections cause some fraction of admitted packets to eventually traverse the incoming links to the source nodes. This "feedback" causes some offered packets to be blocked. One may view this mechanism as a limited form of backpressure which can automatically throttle sources as congestion develops.

In our model, we have chosen to have nodes place top priority on routing packets in transit rather than admitting new packets. Several authors have noted that such a policy can lead to severe unfairness for certain traffic demands. In our example, node $(0,2)$ is entirely occupied routing transit packets, and would be unable to admit any new packets without discarding arriving packets.

At first glance, the "spreading" of packets over a large number of network links may seem to be an inefficient use of bandwidth. However, we now show that this is not so. For the demand of Figure 4.10, an "optimal" routing scheme would

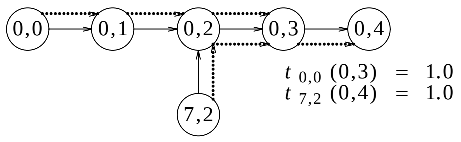


Figure 4.10 - Two continuously transmitting sources produce localized congestion along the unique shortest paths (dotted lines) to the packets' destinations.

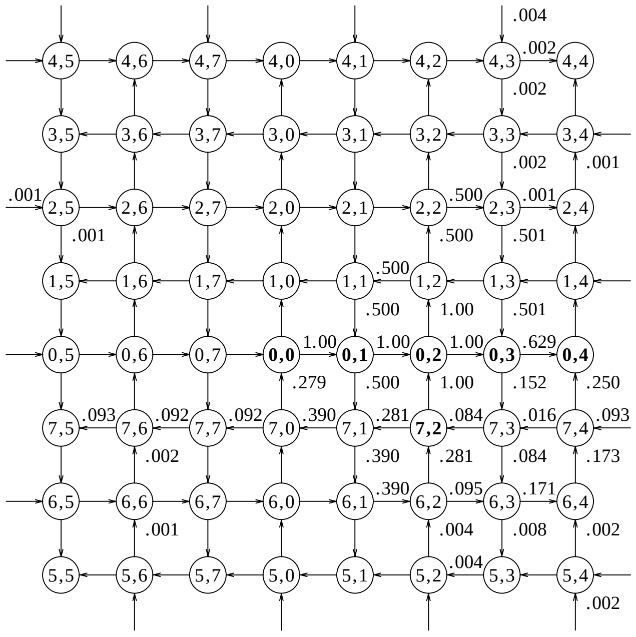


Figure 4.11 - The resulting link utilizations larger than $5.0 \ x \ 10^{-4}$. The mean delay of packets reaching destinations $(0,4)$ and $(0,3)$ are 8.86 and 6.46 slots, respectively.

route 100% of all offered traffic, fully utilizing 10 links. Yet, the "spreading" of packets shown in Figure 4.11 combines to consume the bandwidth equivalent of 12.58 links, with only 2.4% of offered packets blocked.

4.4.4. Discussion of Independence Assumption

The validity of our model is crucially dependent on the validity of the independence approximation. For example, in Figure 4.12 we present a comparison of the transit delay distribution taken from our analysis here and a simulation [30] of a fully utilized $8x8$ MSN operating with the *slot count* rule and a *uniform* traffic demand. Indeed, the approximation is quite accurate.

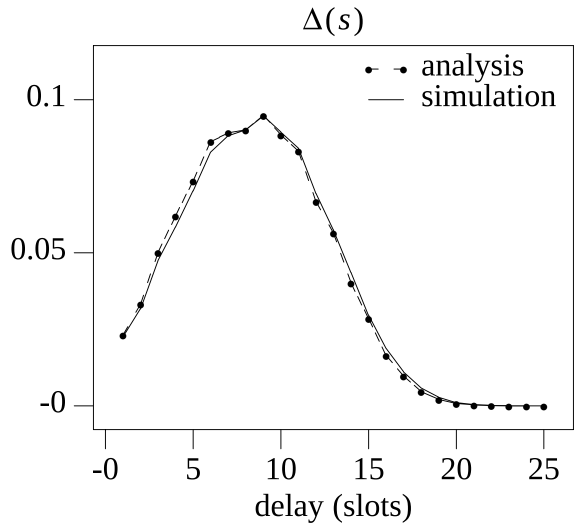


Figure 4.12 - A comparison of the delay distribution found by solving Eqs. 4.48-49 and a simulation transit delay histogram for a fully utilized network with a uniform traffic demand and no propagation delay.

On the other hand we now offer an example of a demand where the independence assumption is "strongly" violated. Consider the "streaming source" of Figure 4.13, where 1 source (a,b) generates packets destined for (x,y) with rate 1 packet/slot. No other sources are active.

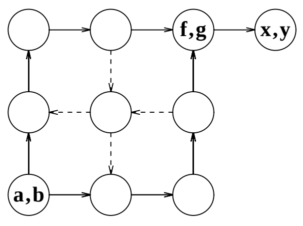


Figure 4.13 - A "streaming source" violates the assumed independence of arrivals at node (f,g).

Clearly no deflections are possible since, by definition, (a,b) admits at most one packet per time slot. Consider the node (f,g) found on a shortest path from (a,b) to (x,y), with disjoint shortest subpaths from (a,b) incident at (f,g). Arrivals at (f,g) in a time slot are strongly dependent since an arrival on one incoming link precludes an arrival on the other. In an $8x8$ MSN using the *slot count* rule, our model predicts that generated packets are blocked with probability $1.3x10^{-4}$, and 44.0% of admitted packets are deflected at least once. Indeed, this dependency exists to some degree in a large class of traffic demands.

4.5. Summary

In this chapter we have focused on a single, important network topology. We first derived approximate delay distributions in an MSN under a uniform offered load. It was shown that delay distributions can be shaped by the choice of contention resolution rule. Of particular importance in certain applications, we have found that the slot count rule dramatically reduces the maximum delay suffered by a packet admitted to a heavily loaded network.

We have also studied the addition of store-and-forward buffers at each network node. This type of node design is of particular importance in implementations where propagation delay is large. Simulation results were presented, and used to study a routing "optimization" conjectured to reduce delay.

We have also studied the MSN performance under a highly nonuniform traffic load. An independence assumption allowed us to derive an approximate delay distribution. This appears to be the first general analysis of deflection routing's ability to diffuse localized network congestion. Other authors are also studying nonuniform traffic in networks with deflection routing [23] and the related problems of packet resequencing [21] and congestion [17]. The advantage of our analysis is generality, though this comes at the cost of computational complexity. For a K node MSN supporting the maximum number of "conversations", it is necessary to solve $O(K^3)$ equations. While this number grows rapidly, it is smaller than the number needed in a direct solution via a Markov chain, where the number of states increases exponentially [27]. We note that Eqs. 4.48-49 may not have a unique solution, however no time-homogeneous demand we tested produced multiple solutions. An open question is if a more complex yet still tractable model exists which will produce a better approximation to the actual network behavior.

The approach we have introduced to study nonuniform traffic may be easily generalized to study deflection routing on arbitrary topologies. Indeed, for a topology of degree 2, all that must be done is to replace the adjacency matrix of the MSN (Table 4.4) with that of the topology under study. For topologies of higher degree, a similar analysis may be used. However, it is likely that the node equations corresponding to (4.48) will quickly become unmanageable. Indeed, even if the equations can be written, the computational complexity of their solution may be enormous.

CHAPTER V

Combined Trunk Group and Deflection Routing

5. Introduction

In this chapter we introduce networks which employ both trunk group and deflection routing. We argue that the combination of these 2 approaches enjoys the benefits of both. In particular we will show that trunk grouping can be used to dramatically reduce deflections. We then examine the performance of 2 network topologies with both uniform and nonuniform offered loads.

Space-division multiplexed packet switches have been proposed to implement broadband switching networks. Such designs can potentially exploit the very high bandwidth of lightwave communications, yet use electronic switching technology operating at much lower speeds. Several authors [38, 39, 40, 41] have proposed fast packet switch architectures that utilize *trunk group routing*. In a multihop network using trunk group routing, packets are routed to their destination by the selection of a sequence of trunk groups. The particular trunk utilized within a trunk group is irrelevant. As a result, switch performance is enhanced by the statistical smoothing of many users communicating on a single trunk group.

More recently Cruz [4, 25] introduced a new broadband switch called the *Statistical Data Fork* (SDF). The switch is well suited to implement trunk group routing on a large scale at very high speeds. The underlying switching fabric of the SDF is an unbuffered omega network [42]. Packets admitted simultaneously to an SDF contend for access to their desired outgoing trunk group. Deflection routing is used *within* the SDF to resolve contention. The switch can be designed so that the probability a packet is routed incorrectly (i.e., routed to an undesired trunk group) is quite small. In this sense routing through the switch is "soft." Bellur and Sasaki [43] have modified the SDF to incorporate internal buffering, and have shown that buffering can further reduce routing error probabilities.

In this chapter we consider the analysis of time-slotted networks of SDFs. To avoid the loss of misrouted packets, we let the network operate with deflection routing at the trunk *group* level. Each SDF attempts to route an arriving packet to an output trunk group along a shortest path to the packet's destination. A misrouted packet is forced to travel a longer path (through more SDFs) to reach its destination. Hence, deflection routing operates on 2 levels, both internal to the switching nodes and external to the switching nodes.

In the next section we introduce our node model. We then analyze the routing performance of an individual node. We next consider the analysis of networks of these nodes. We derive an approximate analytical transit delay distribution for a shuffle-exchange network of SDFs operating with a uniform offered load. We then study a Manhattan Street Network of SDFs with a general, nonuniform offered load. Finally, we propose some alternative node designs that can facilitate the implementation of these networks.

5.1. The Statistical Data Fork

In this section we introduce a model of a node based on the Statistical Data Fork. We begin by describing the simplest formulation of a node with N input trunks and N output trunks. A node is organized as a $2x2$ trunk *group* switch (i.e. 2 input trunk groups and 2 output trunk groups) with $N/2$ trunks assigned to each trunk group. Outgoing trunk groups are labeled 0 and 1.

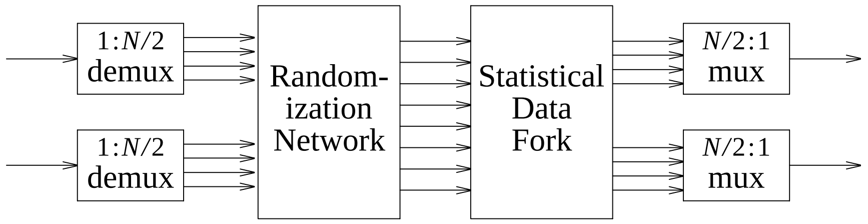


Figure 5.1 - A model of a node (a $2x2$ trunk group switch).

As before we consider a time-slotted network with unit duration time slots and packets of fixed length. Each switching cycle of $N/2$ consecutive time slots is called a *frame*. A node receives up to $N/2$ packets per frame on each of its 2 input trunk groups. All *continuing* packets (i.e., those not yet at their destination) received in a frame are held until the end of the current frame, then simultaneously switched. At most 1 packet is switched to each output trunk. Packets arriving to output trunks are then serially multiplexed on to 1 of the node's 2 output trunk groups, and exit the node in the next frame.

A node comprises 4 stages; a demultiplexing stage, a randomization network, a Statistical Data Fork, and a multiplexing stage (Figure 5.1). Two $1{:}N/2$ demultiplexers form the first stage; each input trunk group is demultiplexed to $N/2$ trunks. That is, each packet arriving in a frame is placed on a unique output trunk of the demultiplexer. This approach was called *input smoothing* in [44]. Similarly, the final stage of a node comprises 2 $N/2{:}1$ multiplexers. A set of exactly $N/2$ trunks exiting the previous stage is connected to each multiplexer.

The second stage is an idealized randomization network with N input trunks and N output trunks. Cruz [25] has suggested an extended omega network with randomized routing as a possible switching fabric for the randomization network. We will discuss an alternative implementation later. For now we assume that, in each frame in steady-state, departures from the randomization network satisfy the following property: *for an arbitrary packet arrival distribution on the N input trunks to the randomization network, the packet departures on each of the N output trunks are independent and identically distributed.* As will be evident in the next section, a randomization network is required in each node since the distribution of packet departures on the groups exiting upstream nodes is highly nonuniform.

The third stage in each node is an SDF with N input trunks and N output trunks. The SDF is an omega network with $\log_2 N$ stages of $N/2$ $2x2$ switching *elements*, with adjacent stages interconnected in a reverse shuffle pattern (Figure 5.2). In our formulation, exactly $N/2$ of the outgoing trunks are connected to each output multiplexer, and hence to each outgoing trunk group. We will introduce an outgoing trunk-to-group assignment rule in the next section. Each switching element executes the adaptive SDF routing algorithm, which is described as follows.

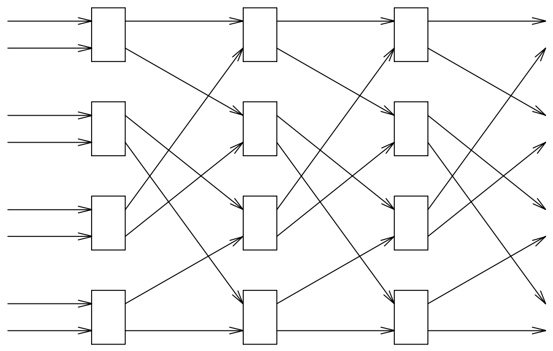


Figure 5.2 - An omega network (N = 8), the switching fabric of the Statistical Data Fork.

The SDF attempts to route each arriving packet to an output trunk *group* along a shortest path to the packet's destination. More precisely, a packet is to be routed to an SDF output trunk which is assigned to the desired trunk group. A packet's *routing preference* is its preferred outgoing trunk group at a node. A packet may prefer to exit the node on trunk group 0, trunk group 1, or prefer neither outgoing trunk group. To route arriving packets, each switching element examines the destination address of each arriving packet, and "moves" the packet to either its upper or lower outgoing link.

The choice of upper or lower link is made simply if we base the construction of the SDF on a repetition code with 2 codewords [25]. Each output trunk group corresponds to a unique "target codeword"; trunk group 0 corresponds to the all zero codeword ($\underline{\mathbf{0}}$), and trunk group 1 corresponds to the all one codeword ($\underline{\mathbf{1}}$). To route a packet to a trunk group, each switching element attempts to route the packet to the output trunk whose binary representation is the target codeword. To do so, each switching element attempts to place a packet destined for trunk group 0 on its upper outgoing link, and a packet destined for group 1 on its lower outgoing link.

When only 1 packet arrives to a switching element, it is switched to its preferred outgoing link. If the packet prefers neither link, a link is chosen by a fair coin toss. When 2 packets arrive to a switching element and both seek different groups, both packets are switched to their desired outgoing links. When 2 packets arrive and only 1 has a routing preference, that packet is switched to its preferred link, and the second packet assigned the alternate outgoing link. When 2 packets arrive and neither has a routing preference, a fair coin toss assigns the packets to outgoing links. When 2 packets arrive and both seek the same outgoing group, a contention resolution rule is invoked to assign the packets to outgoing links. We consider only the *random* contention resolution rule described in Chapter II. Cruz [25] has studied an SDF with an arbitrary priority contention resolution rule. The analysis we present is extensible to the study of the other priority contention rules of Chapter II.

Observe that as many as N arriving packets to be switched in a cycle may seek the same trunk group and hence the same target output trunk. Only one packet can be successful; all other packets will reach other output trunks. In general, packets of a given preference will be nonuniformly distributed over the output trunks.

Finally, each node also removes packets at their destinations and admits new packets to the network. We assume that packets are removed immediately prior to the randomization network, and assume that packets are admitted immediately following the randomization network. At the end of each frame, each of the N inputs to the randomization network is examined for a packet that has reached its destination. Such packets are removed. In each time slot, each node independently generates up to 2 packets. (i.e., up to N packets are generated in a frame). To admit a generated packet, an outgoing randomization network link is randomly selected. The generated packet is placed on that link, if unoccupied. Otherwise, the packet is blocked and lost. Hence, as many as N packets can be removed and/or admitted by each node in each switching cyle. We note that there are a variety of ways of admitting and removing packets. We will return to this discussion in Section 5.4.

5.2. Analysis of a Node

In this section we develop a proposition which enables us to determine routing error probabilities (i.e., deflection probabilities) at a node. In subsequent sections we show how the proposition can be used to analyze certain networks of SDFs.

We begin by introducing some useful notation. Let w be any link in the SDF. We say that link w is $type\,0$ (1) if it is the upper (lower) link exiting a $2x2$ switch element. Recall that all packets arriving to the SDF are continuing packets; packets having reached their destination have been removed, and any new packets have been admitted. An arriving packet has routing preference z if it prefers to exit the switch on trunk group z. We define an additional preference, which we call e, if a packet "does not care" which output group it exits on. Let $\underline{q}^w = (q_0^w, q_1^w, q_e^w)$

denote the triple of probabilities that a packet with preference $z \in \{0, 1, e\}$ arrives on link w. Let ϕw and ψw denote the upper and lower input links to a $2x2$ switching element with output link w.

Consider the j^{th} SDF in the fixed topology network of SDFs. For $z \in \{0,1,e\}$, define the following steady-state variables:

$$p_z(j) = \frac{1}{N} E[\textit{number of packets with preference } z \textit{ arriving at switch } j \textit{ in a frame }], \tag{5.1}$$

$$D_z(j) = P[\textit{ packet deflected } \mid \textit{ packet with preference } z \textit{ arrives at switch } j], \tag{5.2}$$

$$E_l(j) = P[\textit{ packet reaches output trunk group } l \mid \textit{packet of preference } e \textit{ arrives at switch } j], \qquad l \in \{0,1\}. \tag{5.3}$$

Due to the randomization network, a property enjoyed by the SDF is that routing performance at the group level is approximately determined only by $\overline{p}(j) = (p_0(j), p_1(j), p_e(j))$, the assignment of output trunks to trunk groups, and the number of trunks (N). Hence, for a fixed trunk-to-group assignment we have

$$D_z(j) \approx D_z(N, \overline{p}(j)), \tag{5.4a}$$

$$E_z(j) \approx E_z(N, \overline{p}(j)). \tag{5.4b}$$

For a given triple of packet arrival probabilities (i.e., $\overline{p}(j)$), the following proposition enables us to obtain the packet departure probabilities on each output trunk of the SDF.

Proposition 5.1: Suppose that packet arrivals on each input trunk i of an NxN SDF are independent and identically distributed (i.i.d.) with $\underline{q}^i = (p_0, p_1, p_e)$. Then the packet departure probabilities for any link w in the k^{th} stage satisfy

$$\underline{q}^w = \underline{f}^w \underline{f}^{\phi w} \underline{f}^{\phi^2 w} \cdots \underline{f}^{\phi^{k-1} w}(\underline{q}^{\phi^k w}) \qquad \textit{if } k > 0 \tag{5.5}$$

where $\underline{f}^w$ is given by

$$\underline{f}^w((x_0, x_1, x_e)) \triangleq (f_0^w((x_0, x_1, x_e)), f_1^w((x_0, x_1, x_e)), f_e^w((x_0, x_1, x_e))) \tag{5.6}$$

and

$$f_0^w((x_0,\, x_1,\, x_e)) = \begin{cases} 2x_0 - x_0^2 & \textit{type } w = 0 \\ x_0^2 & \textit{type } w = 1 \end{cases} \tag{5.7}$$

$$f_1^w((x_0,\, x_1,\, x_e)) = \begin{cases} x_1^2 & \textit{type } w = 0 \\ 2x_1 - x_1^2 & \textit{type } w = 1 \end{cases} \tag{5.8}$$

$$f_e^w((x_0,\, x_1,\, x_e)) = \begin{cases} x_e - x_0x_e + x_1x_e & \textit{type } w = 0 \\ x_e + x_0x_e - x_1x_e & \textit{type } w = 1. \end{cases} \tag{5.9}$$

□

Proof: The proof is a simple extension of the proof of Proposition 3.1 in [25]. There it was shown that if packet arrivals on each input trunk to the SDF are i.i.d., then the following facts hold:

> **a)** The arrivals on the two input links to any $2x2$ switch element in the SDF are i.i.d. (i.e., $\underline{q}^{\phi w} \equiv \underline{q}^{\psi w}$).
>
> **b)** The $2x2$ switches operate independently and identically, and this operation is described by the vector function $\underline{f}$ satisfying $\underline{q}^w = \underline{f}^w(\underline{q}^{\phi w})$. Eq. 5.5 follows by induction.

We need only show that switch operation is described by (5.7-5.9). Let x_z , $z \in \{0,\, 1,\, e\}$, be the probability that a packet of type z arrives on each input link to a switch. We then identify each packet arrival event that can occur on the inputs to a switch, and each event's probability. Using the definition of the *random* contention resolution rule, for each input event we write the corresponding output event and its probability. These events and probabilities are shown in Table 5.1. Then f_z , $z \in \{0,\, 1,\, e\}$, is the sum the probabilities of events for which a packet of type z reaches an output link of type 0 or 1. Summing theses probabilities produces (5.7-5.9). □

ϕw , ψw	*probability*	w, w'
$-$, 0	$x_0\ (1-x_0-x_1-x_e)$	0 , $-$
$-$, 1	$x_1\ (1-x_0-x_1-x_e)$	$-$, 1
$-$, e	$x_e\ (1-x_0-x_1-x_e)$	$-$, e ($w.p.$ ½)
		e , $-$ ($w.p.$ ½)
0 , $-$	$x_0\ (1-x_0-x_1-x_e)$	0 , $-$
1 , $-$	$x_1\ (1-x_0-x_1-x_e)$	$-$, 1
e , $-$	$x_e\ (1-x_0-x_1-x_e)$	e , $-$ ($w.p.$ ½)
		$-$, e ($w.p.$ ½)
0 , 0	x_0^2	0 , 0
0 , 1	$x_0\ x_1$	0 , 1
0 , e	$x_0\ x_e$	0 , e
1 , 0	$x_0\ x_1$	0 , 1
1 , 1	x_1^2	1 , 1
1 , e	$x_0\ x_e$	e , 1
e , 0	$x_0\ x_1$	0 , 1
e , 1	$x_1\ x_e$	e , 1
e , e	x_e^2	e , 1

Table 5.1 - Events at a switch with upper output link w and lower output w'.

The proposition enables us to determine the statistics of packet types on each output trunk of the SDF (i.e., stage $k = \log_2 N$). An example of packet departure probabilities on each output trunk of a switch is shown in Figure 5.3.

Hence, for any assignment of output trunks to trunk groups, we may simply determine the routing probabilities $D_z(j)$ and $E_z(j)$ by accumulating probabilities on the collections of output trunks corresponding to trunk groups. Before evaluating these routing probabilities, we introduce our trunk assignment rule.

There are a variety of approaches to assigning output trunks to trunk groups. We will consider the assignment that minimizes the average routing error probability $(p_0(j)D_0(j) + p_1(j)D_1(j))$ for a given steady-state arrival distribution, under the constraint that $N/2$ trunks are assigned to each of the 2 output trunk groups. Let $T_z(\overline{p}(j))$ be the set of $N/2$ trunks assigned to trunk group z. Let $\omega \in \{0,1,...,N-1\}$ denote an *output* trunk exiting the SDF. We assign trunks as follows. Order the set of output trunks $\omega \in \{0,1,...,N-1\}$ in nondecreasing order of

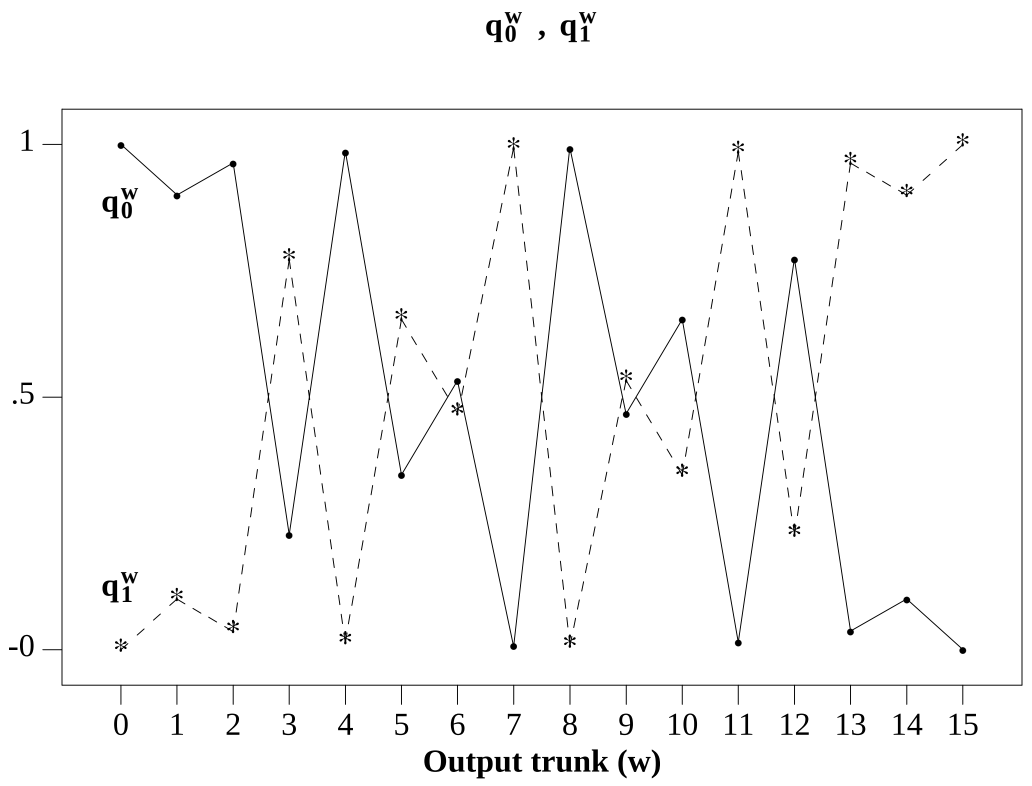


Figure 5.3 - The probability that a packet of each type arrives on each ouput link w of a $16x16$ SDF. In this example a packet arrives on each input trunk in each time slot, and arriving packets are equally likely destined for for each output trunk group (i.e., $p_0 = p_1 = 0.5,\ p_e = 0$).

the corresponding values of $q_1^\omega - q_0^\omega$. Hence, the m^{th} trunk in this ordering satisfies

$$q_1^l - q_0^l \leq q_1^m - q_0^m, \tag{5.10}$$

for $0 \leq l \leq m \leq N - 1$. Partition the "ordered" set of trunks into 2 sets, each of size $N/2$. Assign the first $N/2$ trunks of the ordered set to T_0, and the second $N/2$ trunks to T_1.

Routing probabilities at the j^{th} SDF are then determined as follows:

$$D_0(j) = \frac{1}{p_0(j)\ N} \sum_{w \in T_1} q_0^w, \tag{5.11a}$$

$$D_1(j) = \frac{1}{p_1(j)\ N} \sum_{w \in T_0} q_1^w, \tag{5.11b}$$

$$D_e(j) \equiv 0, \tag{5.11c}$$

$$E_0(j) = \frac{1}{p_e(j)\ N} \sum_{w \in T_0} q_e^w, \tag{5.11d}$$

$$E_1(j) = \frac{1}{p_e(j)\ N} \sum_{w \in T_1} q_e^w. \tag{5.11e}$$

We note that many other trunk-to-group assignments are possible. Of particular interest are robust assignments that are independent of the input trunk packet arrival probabilities (i.e., $T_0(\overline{p}(j)) \equiv T_0$, $T_1(\overline{p}(j)) \equiv T_1$) and perform well. Also of interest are assignments with unequal numbers of trunks assigned to different trunk groups. Indeed, Eqs. 5.11a-e and the remainder of our analysis are valid for any chosen assignment.

5.3. Examples

5.3.1. Uniform Traffic on the Folded Shuffle Exchange Network

Lawrie and Padua [11] first studied deflection routing on the folded shuffle exchange network with V nodes and a uniform offered load. Lawrie and Padua chose to have each packet traverse the unique $L = \log_2 V$ path to its destination, both when admitted and following a deflection, rather than have the packet follow a shortest path to its destination. This choice eases network implementation and increases initial average path length by less than 1 hop. We continue to make the same simplification. Under a uniform offered load, the deflection probabilities at each switch j are constant, i.e.

$$D_0(j) = D_1(j) \equiv P_d, \tag{5.12}$$

and the trunk arrival probabilities are

$$p_0(j) = p_1(j) \equiv \tfrac{1}{2}p, \tag{5.13a}$$

$$p_e(j) \equiv 0. \tag{5.13b}$$

The transit delay distribution is given by

$$h[n] \triangleq P[\textit{admitted packet transit delay is } n \textit{ frames}]$$

$$= \begin{cases} (1 - P_D)^L & n = L \\ P_D(1 - P_D)^L & n = L+1,\ L+2,\ \cdots,\ 2L \\ P_D(1 - P_D)^L \left[1 - \sum_{j=L}^{n-L-1} h[j] \right] & n = 2L+1,\ 2L+2,\ \cdots \end{cases} \qquad (5.14)$$

and the expected packet delay is $\dfrac{1 - (1 - P_D)^L}{P_D(1 - P_D)^L}$ frames.

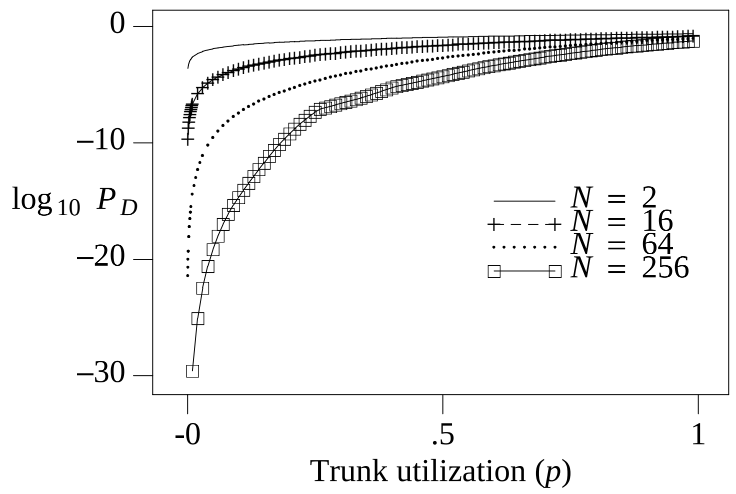


Figure 5.4 - Deflection probability as a function of trunk utilization *p* and SDF trunk size *N* in a shuffle-exchange network with uniform offered load.

Figure 5.4 depicts deflection probability (i.e., $\log_{10} P_D$ for all link utilizations p , $0 < p \leq 1$, as we vary the number of SDF trunks (or equivalently, frame size). The trunk-to-group assignments have been optimized for each value of utilization presented. Figure 5.4 and (5.12-14) enable us to engineer a shuffle-exchange network by selecting the appropriate SDF size (*N*) to achieve a desired delay distribution. The behavior of mean transit delay in a fully utilized network where we vary *N* is shown in Figure 5.5. The benefit of statistical smoothing as *N* increases is readily apparent.

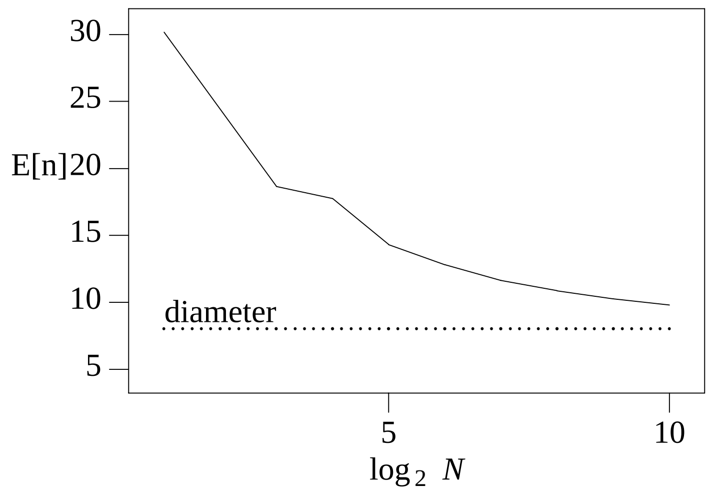


Figure 5.5 - Expected transit delay for a fully utilized 256 node shuffle-exchange network ($p = 1$, $L = 8$) as we vary the number of SDF trunks N.

As an example of the effectiveness of the above architecture at avoiding deflections, if $p = 0.5$, $L = 10$, and $N = 64$, the probability an admitted packet reaches its destination without being deflected is approximately $.99$. If the switching nodes were simple $2x2$ switches rather than SDFs, the probability an admitted packet would reach its destination without being deflected would be approximately $.26$.

5.3.2. Nonuniform Traffic on the Manhattan Street Network

We next consider a *Manhattan Street Network* of SDFs. It is convenient to label switching nodes by row and column number (e.g., (f,g)), and use the notation ‘r’ and ‘c’ (rather than 0 and 1) to indicate row and column trunk groups. To study nonuniform traffic, we extend the approach used to study a network of simple 2x2 switches which was introduced in the previous chapter and in [20]. Here we again consider a nonuniform offered load, with packets generated independently from node to node. Define the following variables:

$$t(f,g,x,y) \triangleq \frac{1}{N/2} E[\textit{number of packets node } (f,g) \textit{ generates in a frame destined for } (x,y)], \quad (5.15)$$

$$\alpha(f,g,x,y) \triangleq \frac{1}{N/2} E[\textit{number of packets node } (f,g) \textit{ admits in a frame destined for } (x,y)], \quad (5.16)$$

$$r(f,g,x,y,s) \triangleq \frac{1}{N/2} E[\textit{number of packets with age } s \textit{ destined for } (x,y) \textit{ arriving on the input row group of } (f,g)], \quad (5.17)$$

$$c(f,g,u,v,l) \triangleq \frac{1}{N/2} E[\textit{number of packets with age } l \textit{ destined for } (u,v) \textit{ arriving on the input column group of } (f,g)], \quad (5.18)$$

$$r^o(f,g,x,y,s) \triangleq \frac{1}{N/2} E[\textit{number of packets with age } s \textit{ destined for } (x,y) \textit{ departing on the output row group of } (f,g)], \quad (5.19)$$

$$c^o(f,g,u,v,l) \triangleq \frac{1}{N/2} E[\textit{number of packets with age } l \textit{ destined for } (u,v) \textit{ departing on the output column group of } (f,g)]. \quad (5.20)$$

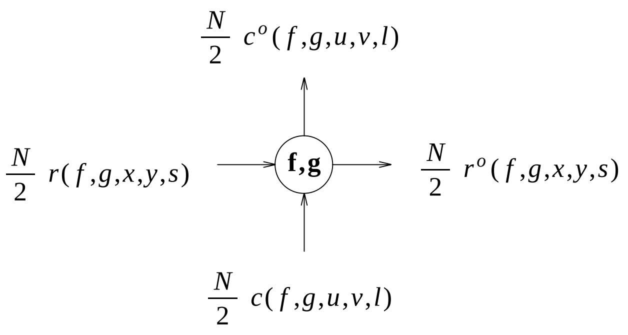

$\frac{N}{2} c^o(f,g,u,v,l)$
$\frac{N}{2} r(f,g,x,y,s)$
f,g
$\frac{N}{2} r^o(f,g,x,y,s)$
$\frac{N}{2} c(f,g,u,v,l)$

Figure 5.6 - Packet arrival and departure rates per frame at node (f,g).

We may then easily write the transit delay distribution for packets destined to node (x,y) as

$$\Delta(x,y,s) \triangleq P[\textit{packet has delay } s \mid \textit{packet arrives to destination } (x,y)]$$
$$= \frac{r(x,y,x,y,s) + c(x,y,x,y,s)}{\sum_s r(x,y,x,y,s) + c(x,y,x,y,s)}. \quad (5.21)$$

For a fixed node (f,g), define the routing preference indicator function

$$I_z(x,y) \triangleq \begin{cases} 1 & \textit{packet destined for } (x,y) \textit{ prefers output trunk group } z \\ 0 & \textit{otherwise,} \end{cases} \tag{5.22}$$

where $z \in \{r, c, e\}$. At node (f,g) packets reaching their destination are removed. Any new packets generated in the last frame are admitted or blocked. We may then write the utilization of continuing packets of type z on each input trunk to the SDF as

$$p_z(f,g) = \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} \tfrac{1}{2} \Big[r(f,g,u,v,l) + c(f,g,u,v,l) \Big] I_z(u,v) + \sum_{\substack{u,v \\ (u,v)\neq(f,g)}} \tfrac{1}{2}\, \alpha(f,g,u,v)\, I_z(u,v). \tag{5.23}$$

The utilization of each input trunk to the SDF at node (f,g) is

$$p(f,g) = p_r(f,g) + p_c(f,g) + p_e(f,g). \tag{5.24}$$

We next develop a recursion to solve for the output trunk group departures. By considering each event that can occur at node (f,g) in a time slot, we may write equations for departures on the output row trunk group in terms of packet arrivals:

$$r^o(f,g,x,y,1) = \beta_r(f,g,x,y)\, \alpha(f,g,x,y) \qquad (x,y) \neq (f,g), \tag{5.25}$$

$$r^o(f,g,x,y,s) = \beta_r(f,g,x,y) \Big[r(f,g,x,y,s-1) + c(f,g,x,y,s-1) \Big] \qquad (x,y) \neq (f,g),\ s > 0, \tag{5.26}$$

where

$$\beta_r(f,g,x,y) = I_r(x,y)\,(1 - D_r(f,g)) + I_c(x,y)\, D_c(f,g) + I_e(x,y)\, E_r(f,g). \tag{5.27}$$

Under our new packet admission model, generated and admitted packets are related by

$$\alpha(f,g,x,y) = t(f,g,x,y) \left[1 - \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} \tfrac{1}{2} \Big[r(f,g,u,v,l) + c(f,g,u,v,l) \Big] \right]. \tag{5.28}$$

The recursion for the output column trunk group departures is obtained by interchanging row and column labels in (5.25-27). The probabilities $D_z(f,g)$ and $E_z(f,g)$ are found using (5.11a-e) and (5.23).

Next we relate the packet departure probabilities of a node and the packet arrival probabilities of directly connected nodes. As in Chapter IV, let $(f,\tilde{g})$ and $(\tilde{f},g)$ be the nodes terminating the outgoing row and column links of node (f,g), respectively. The Manhattan graph adjacency matrix can be determined from Table 4.4. In the absence of propagation delay, trunk group arrivals and departures at adjacent nodes are related by

$$\begin{aligned} r(f,\tilde{g},x,y,s) &= r^o(f,g,x,y,s), \\ c(\tilde{f},g,x,y,s) &= c^o(f,g,x,y,s). \end{aligned} \tag{5.29}$$

Eqs. 5.11, 5.23-29 can be solved numerically to produce the desired delay distribution (5.7).

As a first example we consider a uniform traffic demand. In Figure 5.7 we show the delay distribution of a fully utilized $8x8$ MSN under a uniform offered load as we vary the SDF size.

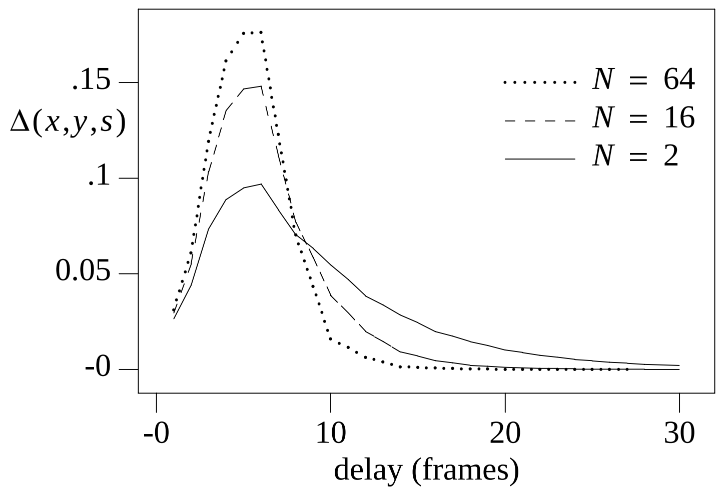


Figure 5.7 - Transit delay distributions for a fully utilized $8x8$ MSN under a uniform offered load where SDF size is varied.

We next study a nonuniform load. Consider the $8x8$ MSN of Figure 5.8 partitioned into 4 communities of interest, with each corresponding to a quadrant. A locally shared resource is found in each community (nodes (2,2), (2,7), (7,7), (7,2)). Every node in each quadrant (other than the resource node) generates a packet with rate 1/15 destined for its local resource node. Solving (5.11), (5.23-29) produce the resulting trunk group utilizations shown in Figure 5.8. The mean transit delay of admitted packets is 3.9 frames, with a standard deviation of 1.9 frames.

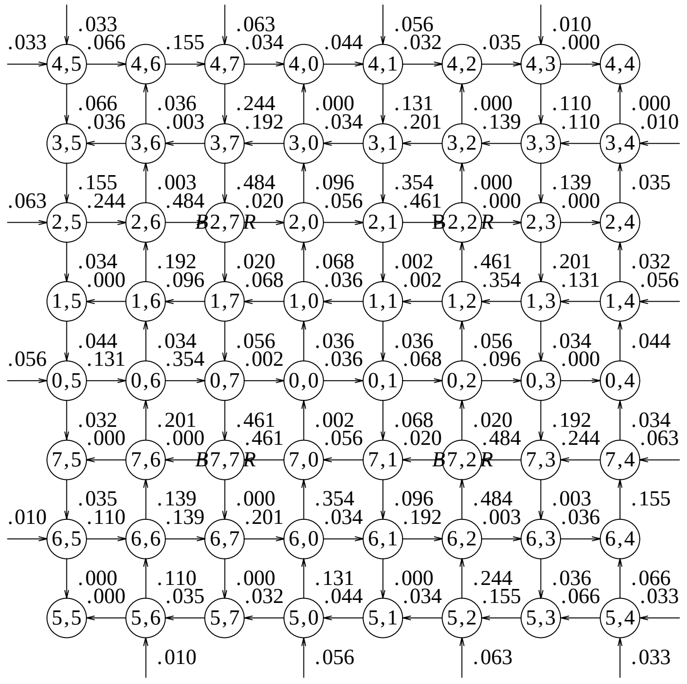


Figure 5.8 - Trunk group utilizations in an $8x8$ MSN of SDFs ($N = 32$) with 4 localized communities of interest.

5.4. Design Alternatives

We now briefly mention certain alternatives to the design of nodes presented in Section 5.1. In particular, we seek simplifications to the node implementation that reduce hardware/software complexity, or modifications that are well suited to the use of optical communication technology. We note that other researchers have recently proposed a similar network using a group of parallel physical channels rather than a single time-multiplexed channel [45].

The proposed switching fabric for both the SDF and the randomization network is the unbuffered omega network. An immediate possibility is to consider a single network capable of performing both tasks. For example, a single omega network can operate with a 2 phase process. In the first phase, packets are admitted to the omega network, and the switching elements use random routing (i.e., independent fair coin tosses assign packets to links). In the second phase, packets are readmitted to the network, but the SDF adaptive routing is employed.

It is also possible to consider a time-division rather than a space-division switch implementation. One interesting opportunity lies with the electro-optical time-domain permutation switch proposed by Ramanan, Jordan, and Sauer [46]. Here only $2\log_2 N - 1$ $2x2$ switches with fixed delay lines of different durations are needed to generate an arbitrary permutation of a sequence of length N. This network appears to be particularly well suited to implement a randomization network, since incoming packets need not be examined to determine switch settings.

Finally, we can look at alternative ways of admitting and removing packets at nodes. When admitting packets, we can capitalize on the nonuniform arrival distributions across the input trunks to the randomization network. Packets can be admitted on the most underutilized trunks, thus reducing blocking probabilities. Also, we can use the SDF as a concentrator to facilitate packet removal [47]. A new target trunk (or set of trunks) can be designated as a packet's routing preference at its destination. Packets reaching this target trunk will be at their destination (with high probability). Hence, it is no longer necessary for a node to examine all N trunks to remove packets at their destination.

5.5. Summary

The use of both trunk group and deflection routing brings many of the advantages of both approaches to a high speed packet switched network. In addition, the combined approach minimizes certain difficulties with either approach. For example, packet resequencing has been cited as a difficult problem to solve in networks with deflection routing [17]. But trunk grouping can decrease deflection probability, which reduces resequencing complexity. A known difficulty of certain proposed trunk group switch architectures is handling packets that fail to reach their intended trunk groups. Here use of adaptive deflection routing enables misrouted packets to successfully reach their destinations.

We have analyzed 2 networks using this combined routing scheme, and the analysis is extensible to more general network topologies. The presence of a randomization network, and the statistical smoothing of many users on a trunk group suggest that the analysis is accurate, particularly under a uniform offered load. Our analysis considered only the random contention rule of Chapter II. It is likely that use of other priority contention rules (e.g., the slot count rule) can further shorten the tails of the resulting delay distributions.

CHAPTER VI

Conclusion

In this dissertation we have focused on a single promising routing technique. We now highlight our most significant contributions. An examination of the existing literature on deflection routing reveals that most published results are topology specific. One goal of our work has been to derive results which are valid for *classes* of topologies, rather than individual topologies. We believe that we have been successful, in part, as is evidenced by the contents of Chapter III. In particular, our bounds on worst case delay for continuous packet admissions appear to be the only published results on this problem. These results demonstrate that the problem of *livelock* can be avoided by the proper design of a contention resolution rule.

The analysis of nonuniform traffic in the MSN also appears to be the first published work on the study of arbitrary traffic loading in a network with deflection routing. Though developed for a particular topology, the analysis immediately extends to the study of any network of degree 2. The significance of this work is that many operational problems (e.g., congestion, livelock, unfairness, etc.) are closely associated with highly nonuniform loads. Hence our analysis represents a first step toward a deeper understanding of a variety of interesting network phenomena.

Finally, the novel introduction of trunk grouping into a network with deflection routing was shown to dramatically reduce deflections. In addition, the architecture of the proposed system directly addresses the problem of tapping the bandwidth of optical communications technology while using lower speed electronic switching technology.

The careful reader has recognized a variety of topics for further research. These include finding optimal routing schemes, improving admission fairness, estimating resequencing buffer capacities, etc. At this stage of research, deflection routing has largely been studied separately from other techniques suitable for implementing high speed networks. Yet very high speed networks will require a combination of new techniques, combined with a comprehensive approach to design. It is in this combination of various approaches that there appears to be the most significant opportunity for additional research.

APPENDIX A

Proof of Proposition 3.3: The proof will proceed as follows. We begin by showing that (3.62) follows immediately from (3.61a) and (3.61b). We will then show that (3.61a) and (3.61b) must hold.

Proof of **(3.62):** Suppose that the evacuation time of the first phase satisfies (3.61a), i.e.

$$T_1 = \tau_1 \le L + D_I(N_1^0 - 1), \tag{A.1}$$

and the evacuation times of the remaining phases satisfy (3.61b), i.e.

$$T_j \le \tau_j \le \max\ [\ L,\ \{\max_{k:1 \le k \le j-1}(\tau_k + \rho_{k,j} + D_I)\} + D_I(N_j^0 - 1)\ ],\quad j = 2,...,M. \tag{A.2}$$

We claim that Eq. 3.62 follows by induction from (A.1) and (A.2). Starting with $j = 2$, evaluate (A.2) to find an upper bound on τ_2. To evaluate the RHS of (A.2), substitute the bound (A.1) for τ_1. If the number of destinations is greater than 2 (i.e., $M > 2$), then for each $j = 3,4,...M$, evaluate (A.2) to find an upper bound on τ_j. To evaluate the maximization on the RHS of (A.2), substitute the bound (A.1) for τ_1, and the previously evaluated bounds on τ_k for each $k = 2,3,4,...,j-1$. Use of the triangle inequality reveals that

$$T_j \le \tau_j \le L + \sum_{k=2}^{j} D_I + \rho_{k-1,k} + \sum_{k=1}^{j} D_I(N_k^0 - 1). \tag{A.3}$$

It follows that the network evacuates in time

$$T = \tau_M \le L + \sum_{j=2}^{M} D_I + \rho_{j-1,j} + \sum_{j=1}^{M} D_I(N_j^0 - 1) \tag{A.4}$$

$$\le L + (M-1)D_I + \sum_{j=2}^{M} \rho_{j-1,j} + D_I(N^0 - M) \tag{A.5}$$

$$\le L + \sum_{j=2}^{M} \rho_{j-1,j} + D_I(N^0 - 1). \tag{A.6}$$

By assumption, a walk connection destinations in order has length P, so

$$\sum_{j=2}^{M} \rho_{j-1,j} = P. \tag{A.7}$$

Substituting (A.7) into (A.6) yields (3.62). □

Proof of **(3.61a-b):** We begin by applying the single destination evacuation time bound (3.1) to the initial phase ($j = 1$), and $(3.61a)$ follows immediately. We next argue that each remaining phase evacuates in time satisfying $(3.61b)$. There are 2 cases to consider for each each phase $j=2,3....,M$:

i) $\tau_j \leq L,$
ii) $\tau_j > L.$

Eq. 3.61b is trivially true for case i; we next show that $(3.61b)$ holds under case ii. The argument has 2 parts:

a) For each phase $j = 2,3....,M$, at least 1 class j packet reaches its destination at or before time

$$\alpha_j \triangleq \max_{k:1\leq k\leq j-1} \tau_k + \rho_{k,j} + D_I. \tag{A.8}$$

b) If class j packets have not evacuated by time $t_{ji} = \alpha_j + i\, D_I$, $i = 0,1,2,...,$ then at least one class j packet reaches its destination in the interval $[t_{ji} + 1,\ t_{ji} + D_I]$.

Proof of part **a)**:

We begin by noting that each packet in the network after time $t = L$ has been deflected at least once. Call a deflection *type A* if a packet is deflected by another packet of the same class, or *type B* if deflected by a packet of lower class. Let $i^t_{\max}(j)$ ($i^t_{\min}(j)$) denote the destination distance of the class j packet furthest from (closest to) its destination at time t, i.e.

$$i^t_{\max}(j) \triangleq max\{\ i\ ;\ N^t_j(i) \geq 1\}, \tag{A.9a}$$
$$i^t_{\min}(j) \triangleq min\{\ i\ ;\ N^t_j(i) \geq 1\}. \tag{A.9b}$$

We now show that only type B deflections of class *j* packets can postpone the time when the first class *j* packet reaches its destination. If a class *j* packet at destination distance $i_{\min}^{t_d}(j)$ is deflected at time t_d, then

$$i_{\min}^{t_d+1}(j) \begin{cases} \leq i_{\min}^{t_d}(j) + D_I - 1 & \textit{if deflection at time } t_d \textit{ is type B} \\ = i_{\min}^{t_d}(j) - 1 & \textit{otherwise.} \end{cases} \quad \text{(A.10)}$$

Hence, we need only consider type B deflections to find an upper bound on the time when the first class *j* packet reaches its destination. Consider the time slot of the last type B deflection of a class *j* packet before the evacuation of class *j* packets. Let us say that a packet of class $k<j$ deflects a class *j* packet at time $t_d < \tau_k$. Such a deflection can increase the destination distance of the deflected packet by no more than $D_I - 1$ hops. It follows that

$$i_{\min}^{t_d+1}(j) \leq i_{\max}^{t_d}(k) + \rho_{k,j} + D_I - 1. \quad \text{(A.11)}$$

We also know that

$$i_{\max}^{t}(k) \leq \tau_k - t, \quad \text{(A.12)}$$

so

$$i_{\min}^{t_d+1}(j) \leq \tau_k - t_d + \rho_{k,j} + D_I - 1. \quad \text{(A.13)}$$

Thus, since t_d was the time slot of the last type B deflection, we have

$$i_{\min}^{\tau_k}(j) \leq \rho_{k,j} + D_I, \quad \text{(A.14)}$$

and the first class *j* packet reaches it destination at time not later than

$$\tau_k + \rho_{k,j} + D_I. \quad \text{(A.15)}$$

Hence, the first class *j* packet reaches it destination before time α_j, and the statement of part **a** is proved. □

Prior to the beginning the proof of part **b)**, we introduce several definitions and facts. Consider 3 packets labeled *A*, *B* and *C*. If packet *A* is deflected in a time slot by packet *B*, we say that packet *B* is packet *A*'s *nemesis*. Thus each time a packet is deflected, it is assigned a (potentially new) nemesis. It is convenient to designate a "current nemesis" (or simply "nemesis") in each time slot, as follows.

1) If *A* had no nemesis in the previous slot and is not deflected in the current slot, *A* continues to have no current nemesis.

2) If *A* is deflected by *C* in the present slot, then *A*'s current nemesis becomes *C*.

3) If *A* had nemesis *B* in the previous slot and neither *A* nor *B* are deflected in the current slot, *A* continues to have current nemesis *B*.

4) Suppose *A* had nemesis *B* in the previous time slot. Suppose that in the current time slot *A* is not deflected, but *B* is deflected by *C*. Then *A*'s current nemesis becomes *C*.

The above definition has the following consequence.

Consequence 1: Consider a network where all contending packets are destined for the same node, and all packets have been deflected at least once. Then a packet can never be more than D_I hops from its current nemesis (if its nemesis is still in the network) or more than D_I hops from the destination.

Recall that each packet found in the network after time $t = L$ has been deflected at least once. In each time slot $t > L$, partition all class *j* packets in the network into 2 classes:

$$M_j^t \triangleq \{ \text{ class } j \text{ packets whose most recent deflection (prior to time } t) \text{ was a type A deflection } \},$$
$$U_j^t \triangleq \{ \text{ class } j \text{ packets whose most recent deflection was a type B deflection } \}.$$

It follows from these definitions that all class *j* packets in the network at time *t* belong either to M_j^t or U_j^t, so

$$|M_j^t| + |U_j^t| = \sum_{i=1}^{L} N_j^t(i). \tag{A.16}$$

We say that a packet in M_j^t is *marked* and a packet in U_j^t is *unmarked.* Note that if a packet is not deflected in a time slot, it will retain its identity (marked or unmarked) from the previous time slot, and move one hop closer to its destination. We now consider how deflections can alter a packet's identity.

[1] A type A deflection of an unmarked packet marks the deflected packet.

[2] A type A deflection of a marked packet keeps the deflected packet marked.

[3] A type B deflection of an unmarked packet keeps the deflected packet unmarked.

[4] A type B deflection of a marked packet unmarks the deflected packet.

Two facts follow from the above definitions:

Fact 1: The set of unmarked class j packets is empty at all times $t \ge \alpha_j$, i.e. $|U_j^t| = 0$ for $t \ge \alpha_j$.

Proof: Follows from the definition of α_j and reasoning following the development of (A.10-15). □

Fact 2: At each time t in the interval $\alpha_j \le t \le \tau_j$, arrange the packets in the nonempty set M_j^t in order of nondecreasing destination distance. Let i_p be the destination distance of the p^{th} packet, so we have

$$1 \le i_1 \le i_2 \le i_3 \le \cdots \le i_m \le L \qquad m = |M_j^t|. \tag{A.17}$$

Then at any time $t \ge L$, no packets adjacent in the ordering differ in destination distance by more than D_I hops, i.e.

$$\max_{p:1<p\le m} \quad i_p - i_{p-1} \le D_I, \tag{A.18}$$

where $i_0 \equiv 0$.

Proof: We now show that (A.17-18) must hold at any time t in $\alpha_j \leq t \leq \tau_j$. Recall that each packet has been marked, and all contending packets are destined for the same node. Hence each packet has a nemesis, and (A.17-18) hold by Consequence 1. □

We now return to the proof of part **b)**. For $t \geq \alpha_j$, any remaining class j packets belong to M_j^t (Fact 1) and no 2 packets are separated in destination distance by more than D_I hops (Fact 2). Regardless of the number of type A deflections after $t = \alpha_j$, at least one class j packet will exit in each interval of D_I time slots. □

APPENDIX B

Development of Eq. 4.40:

We now provide the exact form of Eq. 4.40 for nodes operating under each of the 3 contention rules we studied. Each routing algorithm is symmetric w.r.t. input links, so we present only the equation for the output row link probabilities; the equations for the output column link probabilities can be obtained by interchanging all row and column labels in the corresponding equations below.

We begin by sketching the development of the equations for a network operating with the *slotcount* rule. Consider any node (f,g) in a time slot. Define the routing preference indicator function:

$$I_{x,y}^{r(c,e)} \triangleq \begin{cases} 1 & \textit{packet destined for } (x,y) \textit{ prefers the} \\ & \quad \textit{output row (column, either) link} \\ 0 & \textit{otherwise.} \end{cases} \tag{B.1}$$

Let **E** be the event that both input link buffers are occupied when the routing algorithm is executed; one input link buffer holds a packet of age $s-1$ destined for node (x,y), and the other input link buffer holds a packet of age l destined for node (u,v). Note that $(x,y) \neq (f,g)$ and $(u,v) \neq (f,g)$, since packets destined for node (f,g) have been removed prior to executing the routing algorithm. By convention, a newly generated packet placed in a input link buffer has age 0. Given event **E**, let $I_{s,l}(u,v,x,y)$ be the conditional probability that the packet of age $s-1$ destined for (x,y) exits on the output row link in the next time slot. By the definition of the *slotcount* rule, we have

$$I_{s,l}(u,v,x,y) \triangleq \begin{cases} I_{s<l}(u,v,x,y) & \textit{if } s-1<l \\ I(u,v,x,y) & \textit{if } s-1=l \\ I_{s>l}(u,v,x,y) & \textit{if } s-1>l \end{cases} \tag{B.2}$$

where we define

$$I(u,v,x,y) \overset{\Delta}{=} \tfrac{1}{2}\, I^c_{x,y}\, I^c_{u,v} + \tfrac{1}{2}\, I^r_{x,y}\, I^r_{u,v} + I^r_{x,y}\, I^c_{u,v} + I^r_{x,y}\, I^e_{u,v} + I^e_{x,y}\, I^c_{u,v} + \tfrac{1}{2}\, I^e_{x,y}\, I^e_{u,v} \qquad \text{(B.3)}$$

$$I_{s<l}(u,v,x,y) \overset{\Delta}{=} I^c_{u,v} + I^r_{x,y}\, I^e_{u,v} + \tfrac{1}{2}\, I^e_{x,y}\, I^e_{u,v} \qquad \text{(B.4)}$$

$$I_{s>l}(u,v,x,y) \overset{\Delta}{=} I^r_{x,y} + I^e_{x,y}\, I^c_{u,v} + \tfrac{1}{2}\, I^e_{x,y}\, I^e_{u,v}\,. \qquad \text{(B.5)}$$

Let $\alpha_{f,g}$ ($\beta_{f,g}$) be the probability that no continuing packet arrives on the input row (column) link of (f,g) in a time slot, i.e.

$$\alpha_{f,g} \overset{\Delta}{=} 1 - \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} r_{f,g}(u,v,l) \qquad \text{(B.6)}$$

$$\beta_{f,g} \overset{\Delta}{=} 1 - \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} c_{f,g}(u,v,l)\,. \qquad \text{(B.7)}$$

Let $\gamma_{f,g}$ be the probability that no new packet is generated at (f,g) in a time slot, i.e.

$$\gamma_{f,g} \overset{\Delta}{=} 1 - \sum_{u,v} t_{f,g}(u,v)\,. \qquad \text{(B.8)}$$

The specific expressions for Eq. 4.40 are as follows:

Slotcount rule

$$r^o_{f,g}(x,y,1) = t_{f,g}(x,y)\, \alpha_{f,g}\, \beta_{f,g} \left[I^r_{x,y} + \tfrac{1}{2}\, I^e_{x,y} \right] \qquad \text{(B.9)}$$
$$+\ t_{f,g}(x,y) \sum_{\substack{u,v,l \\ (u,v)\neq(f,g) \\ l>0}} \left[\alpha_{f,g} c_{f,g}(u,v,l) + \beta_{f,g} r_{f,g}(u,v,l) \right] I_{1,l}(u,v,x,y)$$

$$r^o_{f,g}(x,y,s) = \gamma_{f,g} \left[\alpha_{f,g} c_{f,g}(x,y,s-1) + \beta_{f,g} r_{f,g}(x,y,s-1) \right] \left[I^r_{x,y} + \tfrac{1}{2}\, I^e_{x,y} \right] \qquad \text{(B.10)}$$
$$+ \sum_{\substack{u,v \\ (u,v)\neq(f,g)}} \sum_{l=1}^{\infty} \left[r_{f,g}(x,y,s-1)\, c_{f,g}(u,v,l) + c_{f,g}(x,y,s-1)\, r_{f,g}(u,v,l) \right] I_{s,l}(u,v,x,y)$$
$$+ \sum_{u,v} \left[\beta_{f,g}\, r_{f,g}(x,y,s-1) + \alpha_{f,g}\, c_{f,g}(x,y,s-1) \right] t_{f,g}(u,v)\, I_{s,0}(u,v,x,y)$$

$$(x,y) \neq (f,g)\ ,\ s > 1$$

$$r^o_{f,g}(f,g,s) = 0. \tag{B.11}$$

The equation for $r^o_{f,g}(x,y,1)$ consists of 2 terms corresponding to the events where a new packet is generated and not blocked. The first term corresponds to the event where no continuing packet arrives, and the second term corresponds to the event where exactly 1 continuing packet arrives.

The equation for $r^o_{f,g}(x,y,s)$, $s > 1$ consists of 3 terms. The first term corresponds to the event that exactly 1 continuing packet arrives and no new packet is generated. The second and third terms correspond to both input link buffers being occupied immediately prior to executing the routing algorithm, with 2 link arrivals (i.e. no new packet) and 1 link arrival and 1 new packet, respectively. The final equation states that nodes do not emit packets destined for themselves.

The equations for the other rules we studied are developed similarly and stated below. Again it is helpful to define the following collections of routing preference indicator functions:

$$I_r(u,v,x,y) \triangleq I^r_{x,y} + I^e_{x,y} \left(I^c_{u,v} + ½\, I^e_{u,v} \right) + I^c_{x,y}\, I^c_{u,v} \tag{B.12}$$

$$I_c(u,v,x,y) \triangleq I^r_{x,y} \left(1 - I^r_{u,v} \right) + I^e_{x,y} \left(I^c_{u,v} + ½\, I^e_{u,v} \right). \tag{B.13}$$

Random rule

$$\begin{aligned} r^o_{f,g}(x,y,1) = {} & t_{f,g}(x,y)\, \alpha_{f,g}\, \beta_{f,g} \left[I^r_{x,y} + ½\, I^e_{x,y} \right] \\ & + t_{f,g}(x,y) \sum_{\substack{u,v,l \\ (u,v) \neq (f,g)}} \left[\alpha_{f,g} c_{f,g}(u,v,l) + \beta_{f,g} r_{f,g}(u,v,l) \right] I(u,v,x,y) \end{aligned} \tag{B.14}$$

$$\begin{aligned} r^o_{f,g}(x,y,s) = {} & \gamma_{f,g} \left[\alpha_{f,g} c_{f,g}(x,y,s-1) + \beta_{f,g} r_{f,g}(x,y,s-1) \right] \left[I^r_{x,y} + ½\, I^e_{x,y} \right] \\ & + \sum_{\substack{u,v,l \\ (u,v) \neq (f,g)}} \Big[r_{f,g}(x,y,s-1)\, c_{f,g}(u,v,l) \\ & \qquad + c_{f,g}(x,y,s-1)\, r_{f,g}(u,v,l) \Big] I(u,v,x,y) \\ & + \sum_{u,v} \left[\beta_{f,g}\, r_{f,g}(x,y,s-1) + \alpha_{f,g}\, c_{f,g}(x,y,s-1) \right] t_{f,g}(u,v)\, I(u,v,x,y) \\ & \qquad (x,y) \neq (f,g) \ , \ s > 1 \end{aligned} \tag{B.15}$$

$$r^o_{f,g}(f,g,s) = 0. \tag{B.16}$$

Straight rule

$$r^o_{f,g}(x,y,1) = t_{f,g}(x,y)\ \alpha_{f,g}\ \beta_{f,g} \left[I^r_{x,y} + \tfrac{1}{2}\ I^e_{x,y} \right] \tag{B.17}$$
$$+ t_{f,g}(x,y)\ \alpha_{f,g} \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} c_{f,g}(u,v,l)\ I_r(u,v,x,y)$$
$$+ t_{f,g}(x,y)\ \beta_{f,g} \sum_{\substack{u,v,l \\ (u,v)\neq(f,g)}} r_{f,g}(u,v,l)\ I_c(u,v,x,y)$$

$$r^o_{f,g}(x,y,s) = \gamma_{f,g} \left[\alpha_{f,g} c_{f,g}(x,y,s-1) + \beta_{f,g} r_{f,g}(x,y,s-1) \right] \left[I^r_{x,y} + \tfrac{1}{2} I^e_{x,y} \right] \tag{B.18}$$
$$+ r_{f,g}(x,y,s-1) \sum_{\substack{u,v \\ (u,v)\neq(f,g)}} \beta_{f,g}\ t_{f,g}(u,v)\ I_r(u,v,x,y)$$
$$+ r_{f,g}(x,y,s-1) \sum_{\substack{u,v,l \\ (u,v)\neq(f,g) \\ l>0}} c_{f,g}(u,v,l)\ I_r(u,v,x,y)$$
$$+ c_{f,g}(x,y,s-1) \sum_{\substack{u,v \\ (u,v)\neq(f,g)}} \alpha_{f,g}\ t_{f,g}(u,v)\ I_c(u,v,x,y)$$
$$+ c_{f,g}(x,y,s-1) \sum_{\substack{u,v,l \\ (u,v)\neq(f,g) \\ l>0}} r_{f,g}(u,v,l)\ I_c(u,v,x,y)$$
$$(x,y) \neq (f,g)\ ,\ s > 1$$

$$r^o_{f,g}(f,g,s) = 0. \tag{B.19}$$

1. W. F. Leung and G. W. R. Luderer, ‘‘The Network Operating System Concept For Future Services,’’ *AT&T Technical Journal* **68**(2), pp. 23-35 (March/April, 1989).

2. G. Chesson, ‘‘Protocol Engine Design,’’ *Proceedings of USENIX 1987* (1987).

3. D. Mitra, ‘‘Optimal Design of Windows for High Speed Data Networks,’’ *IEEE INFOCOM ’90 Proceedings* **3**, pp. 1156-1163 (June, 1990).

4. R. L. Cruz, ‘‘An Architecture for Very High Speed Packet Switching Systems,’’ *Book of Abstracts*, *IEEE International Symposium on Information Theory*, p. 116 (January, 1990).

5. M. G. Hluchyj and M. Karol, ‘‘ShuffleNet: An Application of Generalized Shuffles to Multihop Lightwave Networks,’’ *Proceedings of IEEE INFOCOM’88*, pp. 379-390 (March, 1988).

6. A. S. Acampora and S. Ali Shah, ‘‘A Packet Compression/Decompression Approach for Very High Speed Optical Networks,’’ *International Telecommunication Symposium*, Brasil (1990).

7. Z. Haas and D. Cheriton, ‘‘Blazenet: A High Performance Wide-Area Packet Switched Network using Optical FIbers,’’ *Proceedings of IEEE Pacific Rim Conference on Communication, Computers and Signal Processing* (June, 1987).

8. A. S. Acampora, M. J. Karol, and M. G. Hluchyj, ‘‘Terabit Lightwave Networks: The Multihop Approach,’’ *AT&T Technical Journal* **67**(1), pp. 21-34 (1987).

9. John Y. Ngai and Charles L. Sietz, ‘‘A Framework for Adaptive Routing in Multicomputer Networks,’’ *1989 ACM Symposium on Parallel Algorithms and Architectures* (1989).

10. P. Baran, ‘‘On Distributed Communication Networks,’’ *IEEE Trans. on Communications Systems* **CS-12**, pp. 1-9 (March, 1964).

11. D. H. Lawrie and D. A. Padua, ‘‘Analysis of Message Switching with Shuffle-Exchanges in Multiprocessors,’’ *Proceedings of Workshop on Interconnection Networks for Parallel and Distributed Processing*, pp. 116-123, ACM (April, 1980).

12. W. D. Hillis, *The Connection Machine*, M.I.T. Press, Cambridge, MA (1985).

13. N. F. Maxemchuk, ‘‘Regular Mesh Topologies in Local and Metropolitan Area Networks,’’ *AT&T Technical Journal* **64**(7), pp. 1659-1685 (September, 1985).

14. N. F. Maxemchuk, ‘‘The Manhattan Street Network,’’ *IEEE Globecom 1985*, New Orleans, La., pp. 255-261 (September, 1985).

15. N. F. Maxemchuk, ‘‘Routing in the Manhattan Street Network,’’ *IEEE Trans. on Communications* **COM-35**(5), pp. 503-512 (May, 1987).

16. N. F. Maxemchuk, ‘‘Comparison of Deflection and Store-and-Forward Techniques in the Manhattan Street and Shuffle-Exchange Networks,,’’ *IEEE Infocom 1989 Proceedings*, Washington, D.C. **3**, pp. 800-809, IEEE Computer Society Press (1989).

17. N. F. Maxemchuk, ‘‘Problems Arising from Deflection Routing: Live-lock, Lockout, Congestion and Message Reassembly,’’ *Proceedings of NATO*

Workshop on Architecture and High Performance Issues of High Capacity Local and Metropolitan Area Networks, France (June, 1990).

18. A. Krishna, ‘‘Communication with Few Buffers: Analysis and Design,’’ *Ph.D. Dissertation*, University of Illinois (December, 1990).

19. T. G. Robertazzi and A. A. Lazar, ‘‘Deflection Strategies for the Manhattan Street Network,’’ *Proceedings of ICC’91* **3**, pp. 1652-1658 (June, 1991).

20. J. T. Brassil and R. L. Cruz, ‘‘Nonuniform Traffic in the Manhattan Street Network,’’ *Proceedings of ICC’91* **3**, pp. 1647-1651 (June, 1991).

21. A. K. Choudhury and N. F. Maxemchuk, ‘‘Effect of a Finite Reassembly Buffer on the Performance of Deflection Routing,’’ *Proceedings of ICC’91* **3**, pp. 1637-1646 (June, 1991).

22. A. K. Choudhury and V. O. K. Li, ‘‘Performance Analysis of Deflection Routing in the Manhattan Street Network,’’ *Proceedings of ICC’91* **3**, pp. 1659-1665 (June, 1991).

23. J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, ‘‘A Performance Model of Deflection Routing and its Application to the Topological Design of Multichannel Networks,’’ *UCLA Computer Science Department Technical Report CSD-910002* (January, 1991).

24. Z. Chen, ‘‘Performance Analysis of a Metropolitan Area Network,’’ *Ph.D. Dissertation*, Cornell University (August, 1991).

25. R. L. Cruz, ‘‘The Statistical Data Fork: A Class of Broadband Multichannel Switches,’’ *to appear in* IEEE Trans. on Communications (submitted February, 1990).

26. V. E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York (1965).

27. A. G. Greenberg and J. W. Goodman, ‘‘Sharp Approximate Models of Adaptive Routing in Mesh Networks (Preliminary report),’’ *Teletraffic Analysis and Computer Performance Evaluation*, Elsevier Science Publishers, B. V. (1986).

28. A. Krishna and B. Hajek, ‘‘Performance of Shuffle-Like Switching Networks with Deflection,’’ *Proceedings of IEEE INFOCOM’90*, pp. 473-480 (June, 1990).

29. T. Syzmanski, ‘‘An Analysis of Hot-Potato Routing in a Fiber Optic Packet Switched Hypercube,’’ *IEEE INFOCOM ’90 Proceedings* **1**, pp. 918-925 (June, 1990).

30. J. T. Brassil and R. L. Cruz, ‘‘An Approximate Analysis of Packet Transit Delay in the Manhattan Street Network,’’ *submitted to* IEEE INFOCOM’92 (June, 1991).

31. B. Khasnabish, ‘‘Topological Properties of Manhattan Street Networks,’’ *IEE Proceedings on Communications* (January, 1990).

32. S. A. Felperin, L. Gravano, G. D. Pifarre, and J. L. C. Sanz, ‘‘Routing Techniques for Massively Parallel Communication,’’ *Proceedings of the IEEE* **79-1**(4), pp. 488-503 (April, 1991).

33. B. Hajek, ‘‘Bounds on Evacuation Time for Deflection Routing,’’ *Proceedings of CISS’89* (March, 1989).

34. A. G. Greenberg and B. Hajek, ‘‘Deflection Routing in Hypercube

Networks (3/89 Draft),'' *submitted to* IEEE Trans. on Communications (1989).

35. A. Borodin and J. E. Hopcroft, ''Routing, Merging and Sorting on Parallel Models of Computation,'' *Journal of Computer and System Sciences* **30**, pp. 133-145 (1985).

36. H. G. Badr and S. Podar, ''An Optimal Shortest-Path Routing Policy for Network Computers with Regular Mesh-Connected Topologies,'' *IEEE Trans. on Computers* **38**(10), pp. 1362-1371 (October, 1989).

37. T. Weller and B. Hajek, ''An Optimal Policy for Deflection Routing on a 2-Dimensional Grid,'' *Proceedings of Spring 1990 Network Topics Course (UILU-ENG-90-2221)*, pp. 33-45 (June, 1990).

38. A. Pattavina, ''Multichannel Bandwidth Allocation in a Broadband Packet Switch,'' *IEEE Journal on Selected Areas in Communications* **6**(9), pp. 1489-1499 (December, 1988).

39. S. Liew and K. Lu, ''Performance Analysis of Asymmetric Packet Switch Modules with Channel Grouping,'' *Proceedings of IEEE Infocom'90*, pp. 668-676 (June, 1990).

40. J. Hui, ''Resource Allocation for Broadband Networks,'' *IEEE Journal on Selected Areas in Communications* **6**(9), pp. 1598-1608 (December, 1988).

41. A. Y. Lin and J. A. Silvester, ''Fixed-Node Routing and its Performance in ATM Networks,'' *Proceedings of IEEE Infocom'90*, pp. 803-810 (June, 1990).

42. D. Mitra and R. Cieslak, ''Randomized Parallel Communications on an Extension of the Omega Network,'' *Journal of the ACM* **34**(4), pp. 802-824 (October, 1987).

43. B. Bellur and G. Sasaki, ''The Buffered Statistical Data Fork,'' *Technical Report C-90-2, Electrical Engineering Research Laboratory*, University of Texas, Austin (October, 1990).

44. M. G. Hluchyj and M. J. Karol, ''Queueing in High-Performance Packet Switching,'' *IEEE Journal on Selected Areas in Communications* **6**(9), pp. 1587-1597 (December, 1988).

45. K. Lee, A. K. Choudhury, and V. O. K. Li, ''The Multi-Channel Architecture for Deflection Routing in High-Speed Manhattan Street Networks,'' *to appear in* Proc. IEEE SICON (September, 1991).

46. S. V. Ramanan, H. F. Jordan, and J. R. Sauer, ''A New Time-Domain Multistage Permutation Algorithm,'' *IEEE Trans. on Information Theory* **36**(1) (1990).

47. Y. Yeh, M. G. Hluchyj, and A. S. Acampora, ''The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching,'' *IEEE Journal on Selected Areas in Communications* **5**(8), pp. 1274-1283 (October, 1987).

48. H. S. Stone, ''Parallel Processing with the Perfect Shuffle,'' *IEEE Trans. on Computers* **20**(2), pp. 153-161 (October, 1971).

49. A. M. Bignell and T. D. Todd, ''SIGnet: A New Ultra-High-Speed Lightwave Network Architecture,'' *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (June, 1989).

50. L. G. Valiant and G. J. Brebner, ''Universal Schemes for Parallel

Communication,'' *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pp. 263-277 (1981).

51. A. M. Bignell and T. D. Todd, ``Analytic Node Model for Deflection Routing Networks,'' *Electronic Letters* **26**(1), pp. 67-69 (January, 1990).

52. F. Borgonovo and E. Cadorin, ``HR4-NET: A Hierarchical Random-Routing Reliable and Reconfigurable Network for Metropolitan Area,'' *Proceedings of IEEE INFOCOM'87*, pp. 320-326 (March, 1987).

53. A. G. Greenberg and J. W. Goodman, ``Sharp Approximate Models of Adaptive Routing in Mesh Networks (12/88 Draft),'' *submitted to* IEEE Trans. on Communications.

54. F. Borgonovo and E. Cadorin, ``Locally Optimal Deflection Routing in the Bidirectional Manhattan Street Network,'' *Proceedings of IEEE INFOCOM'90*, pp. 458-464 (1990).

55. A. G. Fraser, ``Toward a Universal Data Transport System,'' *IEEE Journal on Selected Areas in Communications* **SAC-1**, pp. 803-816 (November, 1983).

56. B. Hajek and R. L. Cruz, ``On the Average Delay for Routing Subject to Independent Deflections,'' *submitted to* IEEE Trans. on Information Theory (June, 1990).

57. J. T. Brassil and R. L. Cruz, ``Broadband Networks Using Combined Trunk Group and Deflection Routing,'' *Proceedings of CISS'91* (March, 1991).

58. B. Hajek and R. L. Cruz, ``On the Average Delay for Routing Subject to Independent Deflections,'' *IEEE International Symposium on Information Theory* (June, 1991).

59. L. Kleinrock, *Queueing Systems - Volume 1: Theory*, 1975.

60. J. T. Brassil and R. L. Cruz, ``Bounds on Maximum Delay in Networks with Deflection Routing,'' *submitted to the* 29th Allerton Conference (June, 1991).

61. J. S. Turner, ``Design of an Integrated Services Packet Network,'' *IEEE Journal on Selected Areas in Communications* **4**(8), pp. 1373-1380 (November, 1986).

62. Z. Zhang and A. S. Acampora, ``Analysis of Multihop Lightwave Networks,'' *IEEE Globecom 1990*, San Diego, CA, pp. 1873-1879 (December, 1990).

63. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill Book Company, New York (1965).

64. A. Y. Lin and J. A. Silvester, ``On the Perfomance of an ATM Switch with Multichannel Transmission Groups,'' *Technical Report C-ENG-89-35, Computer Engineering Division, Dept. of EE-Systems, Univ. of Southern California* (December, 1989).

65. T. G. Robertazzi, ``Toroidal Networks,'' *IEEE Communications Magazine* **26**(6), pp. 45-50 (June, 1988).

66. M. J. Karol and S. Shaikh, ``A Simple Adaptive Routing Scheme for ShuffleNet Multihop Lightwave Networks,'' *Proceedings of IEEE Globecom 1988* (1988).

67. J. M. Jaffe, F. H. Moss, and R. A. Weingarten, ``SNA Routing: Past,

Present, and Possible Future,'' *IBM Systems Journal* **22**(4), pp. 417-434 (1983).

68. F. Kamoun, ``A Drop and Throttle Flow Control Policy for Computer Networks,'' *IEEE Trans. on Communications* **COM-29**(4), pp. 444-452 (April, 1981).

69. T. G. Robertazzi and A. A. Lazar, ``Information Based Deflection Strategies for the Manhattan Street Network,'' *CEAS Technical Report 603* (April, 1991).

70. C. Petitpierre, ``Meshed Local Computer Networks,'' *IEEE Communications Magazine* **Vol. 22**(4), pp. 36-40 (August, 1984).