# Scheduling Nonuniform Traffic in a Packet Switching System with Large Propagation Delay

Bruce Hajek, *Fellow, IEEE,* and Timothy Weller

*Abstract*—**Transmission algorithms are introduced for use in a single-hop packet switching system with nonuniform traffic and with propagation delay that is large relative to the packet transmission time. The traffic model allows arbitrary traffic streams subject only to a constraint on the number of data packets which can arrive *at* any individual source in the system or *for* any individual destination in the system over time periods of specified length. The algorithms are based primarily on sending transmission schedules to the receivers immediately before transmitting each data packet multiple times so that the receiver can maximize the number of packets it captures. An algorithm based on matchings in a random graph is shown to provide mean total delay divided by mean propagation delay arbitrarily close to one, as the propagation delay tends to infinity.**

*Index Terms*—**Switching, scheduling, large propagation delay, nonuniform traffic.**

## I. Introduction

RECENTLY, there has been demand for the multiplexing of many classes of data onto a single network. The resulting integrated data network must provide the same or better quality of service than is provided by a single-class network. The recent and widely accepted Asynchronous Transfer Mode (ATM) standard [1] for integrated networks is illustrative of the solutions proposed to meet the demands. Four key features of ATM deserve a close look because they differ from those features of most classical protocols. First, data are transmitted in small packets, primarily in a connection-oriented fashion. Second, because of high data rates and small packet sizes, the relative propagation delay expressed in terms of number of packets may be quite large, especially in some wide-area networks. Third, because diverse classes of data can be transported via ATM, nonuniform ("bursty") arrival traffic often arises. Finally, guaranteed quality of service is required for some classes of data using ATM, such as real-time video or voice. These applications may require a guarantee on the minimum throughput or a bound on the maximum delay.

In this paper, a model of a packet-switching system which contains these four key features is used. Propagation delay is a key parameter in the model. Most related work (see [2] for an overview) focuses on uniform traffic patterns. The goal of this paper is to study methods for transmitting nonuniform traffic in the type of communication networks described above. A new model of nonuniform traffic is introduced which allows a diverse class of arrival sample paths and facilitates determination of quality of service.

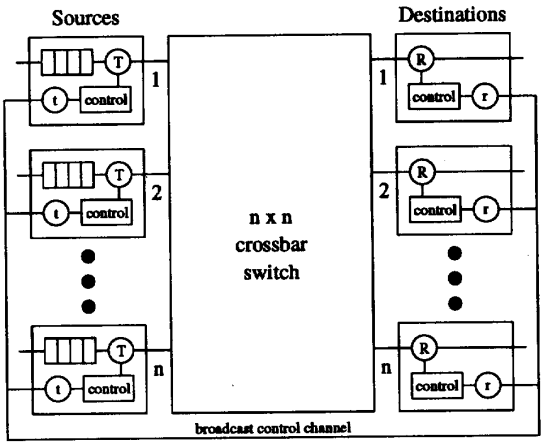


Fig. 1. The basic model.

The *basic model* of a packet-switching system considered in this paper has many stations with a fully connected topology (see Fig. 1). This multiple-access communication model was used previously to analyze optical networks [3], [4], but its application can be broader. Each station has a data transmitter and receiver. The fully connected topology is realized by dedicating a data channel to each source and allowing a receiver to receive from any data channel. Time is slotted. Data are transmitted synchronously in fixed-size packets. Each source can transmit at most one packet during a given slot, and each destination can receive at most one packet during a given slot. A *transmission algorithm* produces a *slot assignment,* which maps each transmission slot on each data channel to a particular virtual queue (source–destination pair).

Each receiver has *capture* capability, which is the ability of a receiver to receive one packet when multiple sources send packets to the receiver at the same time. Capture allows higher throughput than is possible in multiple-access systems without this feature. All transmissions are assumed to be completely reliable.

Manuscript received September 13, 1993; revised August 1, 1994. This research was supported by the National Science Foundation under Contract NSF NCR 9004355 and an AT&T Graduate Fellowship.

B. Hajek is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

T. Weller was with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801. He is now with Donaldson, Lufkin & Jenrette, New York, NY USA.

IEEE Log Number 9408643.

The propagation delay $d_{\text{prop}}$ is the same between any source and destination. Each source has a buffer to hold the queue of packets which have not yet been sent. More than one packet can arrive at a source queue during a single slot. Each source queue can be viewed as many separate *virtual queues* by partitioning traffic by destination. The relationships among the arrival traffic model, the propagation delay, and other system parameters are considered in the analysis.

There exists a low-bandwidth broadcast control channel available to all stations for the exchange of source queue state information and future transmission information. This channel might, for example, share the physical medium with the data channels. The control channel bandwidth should be a small fraction of the total bandwidth. Each station has a separate transmitter and receiver for the control channel. Unless otherwise noted, the control channel is shared using time-division multiplexing with each station allowed to transmit during one control channel minislot in each data slot. The broadcast control channel is assumed to have the same propagation delay as the data channel. For all but the most simple algorithm in this paper (TDMA), the broadcast channel is used in a *tell-and-go* fashion, defined as follows. The transmitter sends information on the broadcast control channel about when packets are to be transmitted, and shortly thereafter sends the packets. At the time of transmission, the receivers need not know about the transmission, or even about the existence of the packets. A propagation time later, the receivers first receive information about the transmissions, and then the transmitted data packets arrive. The information about the packets can be used by the receiver to decide which packet to capture in each slot.

An example of a communication system which fits the basic model is a broadcast network using a passive optical star as in the RAINBOW project at IBM [4]. The data transmission occurs using wavelength-division multiplexing. Each transmitter sends data on a fixed unique wavelength and each receiver has a tunable filter to receive from any one particular transmitter per slot. The control channel is a single unique wavelength, shared among the stations using time-division multiplexing. Each station needs a fixed transmitter and a fixed receiver on this wavelength. RAINBOW is designed primarily for use in metropolitan area networks.

Let $\mathcal{N} = \{1, \cdots n\}$ index the set of stations. The virtual queues of source $i$ are labeled $(i, j)$ for $j \in \mathcal{N}$. Slot $k$ refers to the time period $[k, k+1)$. The sequence of consecutive integers (slots) $k, \cdots, l-1$ is written $[k{:}\, l)$. Let $Z$ be the set of integers and $Z^+$ be the set of nonnegative integers.

During a given slot $k$, new packets arrive at a source and are placed in the buffer at the beginning of the slot. Next, the source makes a decision about which (if any) packet to send. Any packet in the buffer is eligible for transmission. Finally, the selected packet is transmitted. Those transmitted packets which are captured $d_{\text{prop}}$ slots later are said to *depart* in slot $k$. The transmission of packets which are guaranteed to be captured is called *scheduling*. A given receiver can capture at most one packet per slot.

The *access delay* of a packet is defined to be the number of whole slots that the packet is present in the system before its departure. Note that the access delay does not include the propagation delay $d_{\text{prop}}$ suffered by all packets in transit to a receiver after they depart. Because a packet can arrive and depart in the same slot, the minimum possible access delay is zero. Given an arrival sequence and a packet $p$ in the sequence, let $d_p$ denote the access delay of $p$, and let $d_{\max} = \max_p (d_p)$ denote the maximum access delay for the sequence.

The backlog matrix for slot $k$ is the $n \times n$ matrix with $ij$th entry given by the number of packets in virtual queue $(i, j)$ *after* the arrivals but *before* the departures in slot $k$. The term *line* is used to refer to either a row or column of a matrix. A *line sum* is the sum of all the elements in a line of a matrix. *Line backlogs* refer to the line sums of the backlog matrices, which are simply the numbers of queued packets at particular sources, and numbers of queued packets bound for particular destinations.

The delay analysis in this paper is for $(\alpha, S)$ traffic arrival sequences, defined as follows. An $(\alpha, S)$ traffic sequence is simply one such that, over any time period of length $S$ no more than $\alpha S$ packets arrive at a given source and no more than $\alpha S$ packets bound for a given destination arrive at all sources. A random sequence is said to be $(\alpha, S)$ if its sample paths meet the $(\alpha, S)$ constraint. Arrival sequences with sources exhibiting on/off bursts or "hot spots"—commonly studied as representative nonuniform traffic—are easily accommodated with the $(\alpha, S)$ model. Periodic, multiplexed traffic is also easily accommodated by the $(\alpha, S)$ model. For convenience, $\alpha S$ is *taken to be an integer*. An example of a similar single stream model is found in [5].

In this paper, transmission algorithms are presented for use with $(\alpha, S)$ traffic in networks with large propagation delay. Upper bounds on the maximum access delay and maximum line backlog are desired. In a companion paper [6], algorithms are given for scheduling $(\alpha, S)$ traffic in networks with small propagation delay. Those algorithms can be adapted to networks with large propagation delay by making each packet wait a period $d_{\text{prop}}$ while its arrival is announced to all the stations. The main idea in this paper is to avoid such an initial waiting period. This implies that packets are transmitted before the transmitters have a chance to coordinate their transmissions, with the goal of achieving access delay that is small compared to $d_{\text{prop}}$.

In Section II algorithms which transmit each packet only once are considered. It is shown that variations of TDMA permit close to the maximum values of $\alpha$ in this case, but the allowable values of $\alpha$ are rather small for a large number of stations. A considerable improvement is obtained if a single packet can be transmitted more than once, as shown in Section III. All the algorithms in Sections II and III do not require the transmitters to listen to the control channel in order to decide when to transmit packets. Thus the access delay is small no matter how large the propagation delay.

Still, these algorithms all suffer from the fact that the maximum throughput per station decreases as the number of stations increases. Therefore, a final, more sophisticated transmission algorithm is presented which can achieve a fixed throughput, independently of the number of stations, and still provide mean access delay which is negligible compared to the

propagation delay $d_{\text{prop}}$. This is a random algorithm (which is why the delay of a given packet is random) which uses a small amount of feedback information. It is based on a result about matchings in random graphs, similar to results in [7]–[9].

## II. Algorithms Using Single Transmissions

In this section, transmission algorithms which are allowed to transmit each packet only once are considered. The capture capability cannot be used by such algorithms, for *all* packet transmissions must be successful. First, a lower bound on the maximum access delay for a given throughput is developed for such transmission algorithms. Then two transmission algorithms are presented. The first is a standard time-division multiple-access (TDMA) algorithm applied to the basic model. See [10] for a survey of TDMA algorithms. The second algorithm, TDMA-2, achieves nearly twice the throughput of TDMA. Both of the algorithms have maximum access delay, $d_{\max}$, which is bounded as $d_{\text{prop}}$ tends to infinity.

The lower bound on maximum-access delay, given next, is also an implicit upper bound on $\alpha$ for fixed values of $n$ and transmission algorithms which are collision-free. For fixed $n$, the bound implies that unless $\alpha \leq 2/n$, the maximum access delay for any transmission algorithm which transmits each packet only once grows at least linearly with $d_{\text{prop}}$ as $d_{\text{prop}}$ and $S$ tend to infinity. This result is nearly tight, for Theorem 2.3 below implies that if $\alpha \leq 2/(n+1)$ and if $S/d_{\text{prop}} \to 0$, then access delay that grows less than linearly with $d_{\text{prop}}$ can be achieved by a collision-free algorithm.

*Theorem 2.1:* Let $0 < \epsilon < 1$ and $d_{\text{prop}} > 0$ and suppose $\alpha$, $S$, and $n$ are such that $\alpha > 2/((1-\epsilon)n) + 1/S$. Any transmission algorithm which transmits each packet only once has maximum access delay at least $\epsilon d_{\text{prop}}$.

*Proof:* The proof is of the contrapositive statement of the theorem. Suppose there exists a transmission algorithm with $d_{\max} < \epsilon d_{\text{prop}}$ for any $(\alpha, S)$ arrival stream. It is shown that $\alpha \leq 2/((1-\epsilon)n) + 1/S$. Consider one such algorithm and fix a source $i$ and a destination $j$. Consider an arrival stream at source $i$ where the only arrivals are $\lfloor \alpha S/2 \rfloor$ packets at each of the times $0, S, 2S, \cdots, \lfloor d_{\text{prop}}(1-\epsilon)/S \rfloor S$, all bound for destination $j$. These packets must depart strictly before $d_{\text{prop}}$ and avoid collisions with packets from any other sources for the maximum access delay assumption to hold.

No information about other source queues is available to source $i$ until time $d_{\text{prop}}$. Thus source $i$ must transmit its packets without such information. Therefore, to avoid collisions, the slots in which source $i$ transmits its packets to $j$ must be distinct from the slots in which another source would transmit the same arrival stream of packets for destination $j$. Because any one other source could have the same arrival stream and still be in compliance with the $(\alpha, S)$ constraint, these departure slots must be unique for each source. Therefore, since there are $n$ sources

$$n\lfloor \alpha S/2 \rfloor \{ \lfloor d_{\text{prop}}(1-\epsilon)/S \rfloor + 1 \} \leq d_{\text{prop}}$$

which implies that

$$n\lfloor \alpha S/2 \rfloor \{ d_{\text{prop}}(1-\epsilon)/S \} \leq d_{\text{prop}}$$

or that

$$(\alpha S - 1)/2 \leq S/(n(1-\epsilon))$$

which is equivalent to the desired inequality $\alpha \leq 2/(n(1-\epsilon)) + 1/S$.

Advance scheduling of any $(\alpha, S)$ traffic sequence is possible if $\alpha \leq 1/n$, using the TDMA strategy defined as follows. In slot $k$, station $i$ can transmit a packet to the station $j$ such that $j - i \equiv k \pmod{n}$.

*Theorem 2.2:* For $\alpha \leq 1/n$, TDMA has the following properties: The maximum access delay is no more than $S$ and the maximum line backlog is no more than $\alpha S$.

*Proof:* Fix $k > 0$. Suppose for the sake of argument by induction that the maximum line backlog for all slots up to and including slot $k-1$ is less than $\alpha S$. Consider a particular line. The backlog of the line in slot $k$ is equal to its backlog in slot $k - S$ plus the number of arrivals at the line during $[k - S + 1 : k + 1)$ minus the number of departures from the line during $[k - S : k)$. By the induction hypothesis the line backlog in slot $k - S$ is less than or equal to $\alpha S$. There are at least $\lfloor S/n \rfloor$ opportunities for departures from each virtual queue in the line during $[k - S : k)$, and because $\alpha S \leq \lfloor S/n \rfloor$, all packets in the line backlog in slot $k - S$ depart before time $k$. Therefore, because there are no more than $\alpha S$ arrivals during $[k - S + 1 : k + 1)$, the backlog of the line is less than or equal to $\alpha S$ in slot $k$. Also, $d_{\max}$ is at most $S$ because after its arrival a packet is transmitted within $\lfloor S/n \rfloor$ frames of length $n$ slots.

By using the fact that the input streams to the source queues are correlated through the $(\alpha, S)$ constraint, an additional factor of two over TDMA can be achieved for the maximum throughput. The new two-phase algorithm, TDMA-2, is now described. This is a batch algorithm. Let batch interval $k$ be $[kS : (k+1)S)$. Packets arriving during batch interval $k$ depart during $[(k+1)S - 1 : (k+2)S - 1)$. Packets depart either in phase 1 or phase 2.

*Phase 1 Transmission* (The first $\lceil \alpha S/2 \rceil$ slots of every $S$ slot batch interval): Any station with more than $\alpha S/2$ packets for a single destination at the beginning of a batch interval transmits $\lceil \alpha S/2 \rceil$ of these packets immediately in consecutive slots. No other station has more than $\alpha S/2$ packets for the same destination because no more than $\alpha S$ packets in a batch are bound for a single destination. Therefore, no collisions occur. The control channel is used to notify the receivers about packets sent in phase 1, in a tell-and-go fashion as described in the Introduction.

*Phase 2 Transmission* (The last $n\lfloor \alpha S/2 \rfloor$ slots of every $S$ slot batch interval): In the current batch after phase 1, no station has more than $\alpha S/2$ packets for a given destination, and $n\lfloor \alpha S/2 \rfloor$ slots using TDMA can evacuate the remainder of the batch. In all, $\lfloor \alpha S/2 \rfloor$ TDMA frames of length $n$ slots are used.

Suppose $\alpha \leq 2/(n+1)$. Then $\lceil \alpha S/2 \rceil + n\lfloor \alpha S/2 \rfloor \leq S$, and both phases fit within a batch interval. The following theorem follows immediately.

*Theorem 2.3:* For $\alpha \leq 2/(n+1)$ and $n \geq 2$, TDMA-2 has the following properties. The maximum access delay is no more than $2S$ and the maximum line backlog is no more than $2\alpha S$.

## III. A Deterministic Algorithm Using Multiple Transmissions

Transmitting some packets more than once is useful at low throughput. Collisions can occur, in the sense that more than one source can transmit to a given destination in the same slot. Any time a packet is not received due to a collision, a particular transmission algorithm must guarantee that a duplicate of that packet is received during the same batch interval. This guarantee is provided by repeated transmissions of any packet that can possibly collide with others.

The algorithm TDMA-L introduced here, is a two-phase batch algorithm similar to TDMA-2. Phase 1 has $L$ TDMA frames of length $n$ slots. We will show later how to find the optimal $L$. Phase 2 has multiple transmissions of those packets which do not depart in phase 1. Transmission of a fixed batch of packets with maximum line sum $\alpha S$ is now described.

Fix a destination (receiver). For the algorithm description and proof of the following theorem, *all packets are bound for this destination.* The analysis holds for any destination. The receiver is notified of arrivals via the control channel in a tell-and-go fashion so that the transmission times of all packets are known to the receiver before phase 2 reception. Let $M_i$ be the number of packets at source $i$.

*Phase 1 Transmission* (The first $nL$ slots of every $S$ slot batch interval): Each source $i$ transmits min $\{L, M_i\}$ packets using $L$ TDMA frames. Thus no more than $L$ packets can be sent by a single source. Note that no line has more than $\alpha S - L$ packets after phase 1.

*Phase 2 Transmission* (The last $S - nL$ slots of every $S$ slot batch interval): Let

$$T_i = \left\lfloor \frac{1}{L+1}(\alpha S - M_i) \right\rfloor + 1.$$

Each source $i$ transmits each remaining packet exactly $T_i$ times in any order. The control channel is used to notify the receivers about packets sent in phase 2.

*Phase 1 Reception:* Every packet sent in phase 1 is received without conflict because TDMA slots are all reserved.

*Phase 2 Reception:* The reception of phase 2 packets is coordinated by the use of maximum matchings. Consider the following bipartite graph $G = (U, V, E)$ where

$U = \{\text{nodes representing all phase 2 packets}\}$

$V = \{\text{nodes representing phase 2 slots}\}$

$E = \{(u, v): u \in U, v \in V, \text{ packet } u \text{ is sent during slot } v\}$.

It is shown next that the receiver can find a (maximum) matching on $G$ that covers $U$, so that all phase 2 packets can be received. The transmission schedule is given by the matching. Let $\eta$ be the number of sources with at least one packet after phase 1. Note that the degree of any node in $V$ is less than or equal to $\eta$. Let $u$ be a given node in $U$. The degree of $u$ is $T_i$, where $i$ is the source corresponding to $u$. The arrival constraint (no more than $\alpha S$ packets for the fixed destination) requires that $M_i + (\eta - 1)(L+1) \leq \alpha S$. To see this note that source $i$ has $M_i$ packets and the other $\eta - 1$ sources have at least $L+1$ packets before phase 1. Rearranging this inequality and using the definition of $T_i$ gives $\eta \leq T_i$. Thus the degree of any node in $V$ is less than or equal to the degree of any node in $U$. This, by a well-known consequence of Hall's Marriage Theorem [11], implies that there is a maximum matching for $G$ that covers $U$.

TABLE I
Values of $L$ which Minimize $\lfloor \psi(L) \rfloor$ Over Integers $L$

| | $n = 10$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
|---|---|---|---|---|---|
| $\alpha S = 10$ | 2 | 0 | 0 | 0 | 0 |
| $10^2$ | 26 | 8 | 2 | 0 | 0 |
| $10^3$ | 267 | 82 | 28 | 9 | 2 |
| $10^4$ | 2679 | 822 | 289 | 94 | 30 |
| $10^5$ | 26794 | 8221 | 2900 | 943 | 307 |

Next, the maximum length of phase 2 is determined, which implies a constraint on $S$, because a batch must be completed in $S$ total slots. Focus on a source $i$ which has $M_i$ packets in a batch for the given destination. The worst value of $M_i$ and the best value of $L$ are considered. Assume that $L+1 \leq M_i \leq \alpha S$ because if $M_i \leq L$ then phase 2 is not needed. Let $\zeta(M_i, L)$ be the average number of phase 2 slots needed per packet for a given batch at source $i$. Note that (dropping the subscripts on $T$ and $M$ for convenience),

$$\zeta(M, L) = \frac{(M-L)T}{M} \tag{1}$$

$$= \frac{M-L}{M}\left\lfloor \frac{\alpha S - M}{L+1} + 1 \right\rfloor. \tag{2}$$

The average number of phase 2 slots needed per packet at any source $i$ (even if the source has packets for more than one destination) is thus at most $\zeta(M^*(L), L)$, where $M^*(L)$ is obtained by maximizing $\zeta(M, L)$ with respect to $M$ over the range $L+1 \leq M \leq \alpha S$. Careful examination of $\zeta$ shows that either $M^*(L) = L+1$ or $\alpha S - M^*(L)$ is a multiple of $L+1$ so that the $\lfloor \cdot \rfloor$ in (2) can be removed before maximization if the maximization is restricted to the set $\{L+1\} \cup \{\alpha S - q(L+1): q = 0, 1, \cdots, \lfloor \alpha S/(L+1) \rfloor\}$. If the $\lfloor \cdot \rfloor$ is removed, then the function $\zeta(M, L)$ is convex in $M$, and the derivative $\partial\zeta(M, L)/\partial M$ has its only root at $M = \sqrt{L(\alpha S + L + 1)}$. Therefore, the maximizer $M^*(L)$ can be found by checking only three values: $L+1$ and the two integers of the form $\alpha S - q(L+1)$ nearest to $\sqrt{L(\alpha S + L + 1)}$.

Define $\psi(L) = nL + \alpha S \zeta(M^*(L), L)$. Then $\lfloor \psi(L) \rfloor$ phase 1 and phase 2 slots are sufficient to transmit a batch of packets with line sums at most $\alpha S$ using TDMA-L, because source $i$ has at most $\alpha S$ packets for all destinations combined. Hence, if $S \geq \lfloor \psi(L) \rfloor$ (for given $n$ and $\alpha S$), TDMA-L works properly, so that the maximum access delay is at most $2S$. Table I lists values of $L$ which minimize $\lfloor \psi(L) \rfloor$, and Table II lists the corresponding values of $\alpha S / \lfloor \psi(L) \rfloor$, which is the throughput achievable if $S = \lfloor \psi(L) \rfloor$.

TABLE II
MAXIMUM VALUES OF $\alpha S/\psi(L)$ (ACHIEVABLE VALUES OF $\alpha$)

| | $n = 10$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
|---|---|---|---|---|---|
| $\alpha S = 10$ | 0.2857 | 0.1000 | 0.1000 | 0.1000 | 0.1000 |
| $10^2$ | 0.2564 | 0.0694 | 0.0220 | 0.0100 | 0.0100 |
| $10^3$ | 0.2536 | 0.0667 | 0.0190 | 0.0058 | 0.0020 |
| $10^4$ | 0.2533 | 0.0664 | 0.0187 | 0.0055 | 0.0017 |
| $10^5$ | 0.2533 | 0.0664 | 0.0187 | 0.0055 | 0.0017 |

For example, if there are $n = 100$ stations and $\alpha S = 100$, then $(\alpha, S)$ traffic is supported by TDMA-L with $L = 8$ if $\alpha = 0.0694$ and $S = \lfloor \psi(L) \rfloor = 1442 \approx 100/0.0694$. Of the 1442 slots per batch in this example, 800 are used by the phase 1 TDMA. The next theorem presents achievable values of $\alpha$ for large $n$ and $S$.

*Theorem 3.1:* Given any $\epsilon > 0$, for large enough $n$ and $\alpha S$ with $n \leq (\alpha S)^{2-\epsilon}$, if $\alpha \leq (1-\epsilon)/(2\sqrt{n})$, TDMA-L has the following properties: The maximum access delay is no more than $2S$ and the maximum line backlog is no more than $2\alpha S$.

*Proof:* Suppose that $\alpha S$ and $n$ approach infinity with $n \leq (\alpha S)^{2-\epsilon}$ and $L \to \infty$ such that $L/\alpha S \to 0$. For large $\alpha S$, $n$, and $L$, $M^*(L) \approx \sqrt{L\alpha S}$ and $\zeta(M^*(L), L) \approx \alpha S/L$ so that $\psi(L) \approx nL + (\alpha S)^2/L$, which is minimized at $L = \alpha S/\sqrt{n}$. This value of $L$ is consistent with the assumptions $L \to \infty$ and $L/(\alpha S) \to 0$. Substituting this value of $L$ into $\psi(L)$ gives $\psi(L) \approx 2\sqrt{n}\alpha S$. The inequality $\lfloor \psi(L) \rfloor \leq S$ thus holds for large $n$ if $\alpha \leq (1-\epsilon)/(2\sqrt{n})$ and the theorem is proved.

To see that the approximation is good, note that the values in Table II for $\alpha S \geq n$ are close to $1/(2\sqrt{n})$. An open problem is to develop a companion result to Theorem 2.1 for transmission algorithms which are allowed to send packets multiple times, providing a converse to Theorem 3.1. In particular, it is conjectured here that $\alpha \geq \Omega(1/\sqrt{n})$ with $n = \alpha S$ causes any transmission algorithm to have maximum access delay which grows at least linearly with $d_{\text{prop}}$ for sufficiently large $d_{\text{prop}}$.

## IV. A RANDOMIZED ALGORITHM USING MULTIPLE TRANSMISSIONS

This section suggests how to transmit $(\alpha, S)$ traffic for large propagation delay with $\alpha \leq \frac{1}{4}$. This is significant because the algorithms described above require a value of $\alpha$ which is decreasing in the number of stations. The algorithm itself introduces randomization, so that although a fixed deterministic input sequence is assumed, the access delay of a given packet is random. Roughly speaking, it is shown that, using randomization and a small amount of feedback, the mean access delay can be made small compared to $d_{\text{prop}}$ if $\alpha \leq \frac{1}{4}$.

As noted in the Introduction, another strategy, appropriate for any $\alpha \leq 1$, is to first announce the arrival of packets on the control channel. After a delay of at least $d_{\text{prop}}$, the packets can then be scheduled in a conflict-free fashion [6]. This leads to mean access delay at least as large as $d_{\text{prop}}$, whereas here and elsewhere in this paper the goal is to make the access delay small compared to $d_{\text{prop}}$. The algorithm of this section can be viewed as the notify-and-schedule algorithm, but with three early transmissions of each packet inserted, in an attempt to get packets to their destinations sooner.

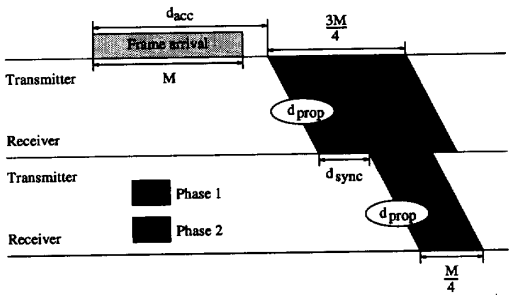


Fig. 2. Timing of algorithm.

The transmission algorithm is first described, and then the delay is analyzed. The delay of any fixed packet is random because the algorithm is random. *Frame* $f$ refers to slots $[fM:(f+1)M)$ where $M$, the *frame length*, is a multiple of $\alpha S$ ($M = l\alpha S$). For brevity, the transmission algorithm is described for packets arriving during frame 0. These are referred to as frame 0 packets. The description applies to packets arriving during frame $f$ by adding $fM$ to all of the times.

The transmission algorithm attempts to minimize collisions by randomly spreading out transmissions over a long period of time. Packet transmission occurs in two phases. During phase 1, each packet is transmitted three times in randomly chosen slots. Before phase 1 transmission begins, messages are transmitted on the broadcast control channel which describe when transmissions will occur during the phase. The algorithm uses the tell-and-go mechanism, so that phase 1 transmission can begin before the control channel information is received.

The beginning of phase 2 transmissions is at least $d_{\text{prop}}$ time units after the beginning of phase 1 transmissions. Phase 2 may need to begin somewhat later (by a time interval called $d_{\text{sync}}$ below) in order to avoid the phase 1 transmission interval of another batch of packets. By that time, the stations all know when the phase 2 packets will be received. They can all separately compute which packets will be captured, and which will not. During phase 2, those packets which are not captured (or will not be captured) in phase 1 are scheduled for transmission in phase 2 using conflict-free scheduling. Phase 1 thus uses 75% of the total transmission slots and phase 2 uses the other 25%. Under the assumption that $\alpha \leq 1/4$, the phase 2 slots are sufficient to transmit all packets that are not captured in phase 1, even if no packets are successful in phase 1.

See Fig. 2 for the timing of this two-phase algorithm. The three parallel axes each represent the time axis, though the second axis has a fixed offset from the first, and the third has the same fixed offset from the second, because the algorithm is designed for $d_{\text{prop}}$ substantially larger than $M$.

*Phase 1 Transmission:* Frame 0 packets arriving at a particular source are transmitted in an interval of length $\frac{3}{4}M$. Before any transmission, the whole frame must be accumulated and all destinations must be notified of the packet transmission

slots. This requires an accumulation delay $d_{\text{acc}}$ from the start of frame 0 until the first packet of the frame can be transmitted. A frame 0 packet which arrives at a particular source during batch interval $b$ is assigned three distinct slots chosen uniformly at random from among all slots in $[d_{\text{acc}}: d_{\text{acc}} + \frac{3}{4}M)$ which have not already been assigned by the source to other frame 0 packets. These are the packet's phase 1 *transmission slots* $s_1$, $s_2$, and $s_3$. If the packet arrives during batch interval $b$ (within frame 0), then during batch interval $b+1$, the vector (source, destination $s_1$, $s_2$, $s_3$) is sent on the control channel to announce the future transmissions. Because a batch contains no more than $\alpha S$ packets, these control information vectors can be accommodated one per slot. Finally, the packet is transmitted during slots $s_1$, $s_2$, and $s_3$. Set $d_{\text{acc}} = M + \alpha S$. This assignment of $d_{\text{acc}}$ leaves adequate time for the sources to accumulate and announce transmission times of packets before actual transmission occurs, as described. Note that $d_{\text{acc}} \leq 2M$.

*Phase 2 Transmission:* At time $d_{\text{acc}} + d_{\text{prop}}$, a source knows whether a particular frame 0 packet will be captured during phase 1 reception. The packets which are not captured in phase 1 are scheduled in slots $[d_{\text{acc}} + d_{\text{prop}} + d_{\text{sync}}: d_{\text{acc}} + d_{\text{prop}} + d_{\text{sync}} + \frac{1}{4}M)$, where $d_{\text{sync}}$ is a synchronization delay of at most $M$ slots, chosen to avoid overlap of phase 1 and phase 2 transmissions. If $d_{\text{prop}}$ has the form $(k + \frac{3}{4})M$ for some integer $k$, then $d_{\text{sync}}$ can be taken to be 0. The scheduling can be accomplished by a transmission algorithm based on maximum matching because there are no more than $\alpha M$ frame 0 packets for any destination and at any source. Such algorithms are well-known for applications in satellite transmission scheduling [12].

*Phase 1 Reception:* Focus on a particular receiver, which during slots $[d_{\text{acc}} + d_{\text{prop}}: d_{\text{acc}} + d_{\text{prop}} + \frac{3}{4}M)$ receives frame 0 packets sent in phase 1 from all sources. Let $R$ denote the number of such packets. Note that $R \leq \alpha M$ by the assumption that the traffic sequence is $(\alpha, S)$. The locations of each frame 0 packet's three transmission slots are known to the receiver by time $d_{\text{acc}} + d_{\text{prop}}$. This allows the receiver to construct a directed bipartite graph $G = (U, V, E)$ where

$U = \{$nodes representing frame 0 packets destined for the receiver$\}$, $|U| = R$

$V = \{$nodes representing the slots to receive frame 0 packets$\}$, $|V| = 3M/4$

$E = \{(u, v): u \in U, v \in V,$ packet $u$ is sent during slot $v\}$.

Consider any matching on $G$. Each slot in $V$ is matched to at most one packet in $U$ and each packet in $U$ is matched to at most one slot in $V$. Unmatched slots in $V$ are unused for reception of frame 0 packets, and unmatched packets in $U$ are not successful in phase 1. To maximize throughput in phase 1, the choice of which packet to receive in each slot is made by finding a maximum matching on $G$. An example is shown in Fig. 3.

*Phase 2 Reception:* Even though most frame 0 packets succeed in phase 1, some packets may fail. During slots $[d_{\text{acc}} + 2d_{\text{prop}} + d_{\text{sync}}: d_{\text{acc}} + 2d_{\text{prop}} + d_{\text{sync}} + M/4)$, these leftover packets are received without conflict, due to the scheduling in phase 2 transmission.

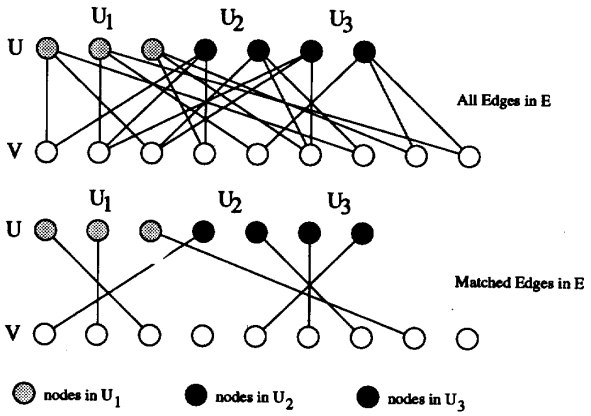


Fig. 3. Phase 1 reception.

This completes the description of the transmission algorithm. The mean access delay is close to the minimum possible, as shown by the next theorem which is proved in the Appendix.

*Theorem 4.1:* Choose any $\alpha \leq \frac{1}{4}$, integer $S \geq 1$, and $\epsilon > 0$. If $d_{\text{prop}}$ is sufficiently large and the above randomized transmission algorithm is used, then the following is true: Given any $(\alpha, S)$ arrival sequence and any fixed packet $p$ in that sequence, the expected access delay of the packet $d_p$ is at most $\epsilon d_{\text{prop}}$. The maximum access delay is at most $(1 + \epsilon)d_{\text{prop}}$.

The algorithm is conservative in the sense that each packet is guaranteed success on its first transmission in phase 2. Less conservative transmission algorithms may provide smaller mean access delay but larger maximum access delay.

The algorithm yields bounded access delay for any $M$, but the mean access delay may be larger than necessary if too small a value for $M$ is chosen. There are two ways to make $M = l\alpha S$ large. First, $l$ can be made large. Second, each batch interval can be extended by making $S$ large, although delay also increases for finite $d_{\text{prop}}$. In either case, $M$ must be $o(d_{\text{prop}})$ as $d_{\text{prop}}$ tends to infinity in order to achieve the goal of making the mean access delay $o(d_{\text{prop}})$.

The control information required is relatively small, but computation can be prohibitively large without some modification. The algorithm as given requires each source to execute the maximum matching algorithm for each destination to which it sends packets in order to determine phase 1 succcesses. In practice, it is probably desirable to wait an additional $d_{\text{prop}}$ before phase 2 to allow feedback information from phase 1 to propagate back to the sources, and then the algorithm can send explicit phase 1 acknowledgments over the control channel. The maximum matching at each receiver can proceed incrementally as each net control information vector is received. Because a packet seldom waits until phase 2, the additional delay does not affect the result in Theorem 4.1. If $\alpha \leq 1/5$, greedy scheduling based on maximal matchings suffices for the transmission scheduling in phase 2, with a 60%/40% allocation of slots between phases 1 and 2. This lowers the computational burden for phase 2 considerably.

## V. Conclusions

As stated in Theorem 2.1, the successful transmission of $(\alpha, S)$ traffic in a system with large propagation delay yields large maximum-access delay, assuming each packet is transmitted only once. Significant improvement results if packets can be transmitted more than once, as shown in Section III. Finally, mean access delay of not much more than $d_{\text{prop}}$ can be achieved for very large $d_{\text{prop}}$ and throughput up to 1/3, as shown in Section IV, demonstrating that a globally known state is not necessary to capture most packets without feedback.

We leave open the question of whether $\alpha$ larger than $\Omega(1/\sqrt{n})$ can be supported with maximum delay smaller than $d_{\text{prop}}/2$, as $n$ tends to infinity. (We conjecture the answer is no.) Also unknown is how large $\alpha$ can be so that the mean access delay, divided by $d_{\text{prop}}$ for some random transmission algorithm tends to zero in the limit for large $d_{\text{prop}}$. We showed $\alpha \leq 1/4$ is sufficient and conjecture (see end of the Appendix) that $\alpha \leq 1/2$ is best possible.

## VI. Appendix
### Proof of Large Propagation Delay Theorem 4.1

Fix any receiver. The key to the proof of Theorem 4.1 is to show that phase 1 is successful with high probability. That is the purpose of the next lemma. Let $G$ be any possible bipartite graph constructed for phase 1 by the receiver as described above. Let $\mathcal{U} = (U_1, \cdots, U_n)$ be a partition of $U$, where $U_i$ is the set of nodes corresponding to packets transmitted by source $i$. The phase 1 transmission slot selection algorithm constructs $G$ from $\mathcal{U}$ and $V$. Let $P_{\mathcal{U}}$ be the probability that a maximum matching on $G$ does not cover $U$, given a partition $\mathcal{U}$.

*Lemma A.1:* For $\alpha \leq \frac{1}{4}$

$$\max_{\mathcal{U}:|U|\leq \alpha M} P_{\mathcal{U}} \to 0, \qquad \text{as } M \to \infty.$$

Lemma A.1 is proved after a technical lemma is presented. Fix any $A \subset U$. The edge set $E^A = E \cap (A \times V)$ contains only edges from $E$ which have a node in $A$. Analysis of the matching used in phase 1 reception is facilitated by the introduction of sets $E^0$ and $E^1$, where $E^1$ has the same distribution as $E^A$, and the sets are constructed as follows. Define $A_i$ to be the subset of $A$ corresponding to source $i$ packets. Each packet $u \in A$ chooses any three elements $v_1(u), v_2(u), v_3(u) \in V$, uniformly at random, independently of all other packets in $A$. Let all such selections determine $E^0$, so that $E^0 = \{(u, v_1(u)): u \in A, l = 1, 2, 3\}$. Note that the vectors $(v_1(u), v_2(u), v_3(u))$ and set $E^0$ are similar to the transmission slot vectors of $(s_1, s_2, s_3)$ and set $E^A$, but that *source conflicts* can occur in $E^0$. A source conflict occurs if for some $i$, there is more than one edge from $A_i$ to some $v$. This violates the basic model constraint of one packet per source per slot. The set $E^0$ is now modified for each $v \in V$ to remove all *source conflicts*. The heads of all but one of the conflicting edges are moved to new nodes in $V$. Each new node is chosen uniformly from all nodes, which does not create a new source conflict. The modified set with all source conflicts removed is labeled $E^1$. The construction of $E^0$ and $E^1$ is complete, and it is clear that $E^A$ and $E^1$ have the same distribution. Fig. 4 illustrates an example of the construction of $E^1$ from $E^0$.

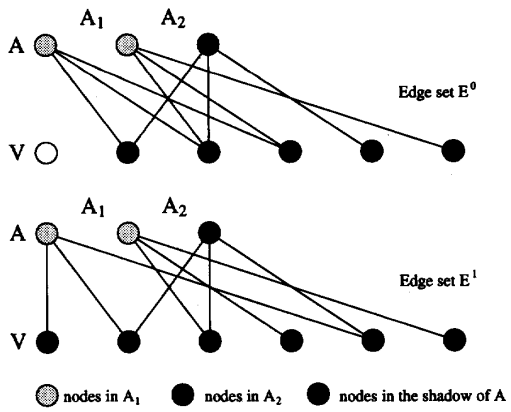


Fig. 4. Removing source conflicts.

Recall that for any edge set $E$, the shadow of $A$ is defined as $S_E(A) = \{v \in V: (u, v) \in E, u \in A\}$. Clearly, $S_{E^A}(A) = S_E(A)$. Fix any $B \subset v$. Because $E^A$ and $E^1$ have the same distribution, $P\{S_E(A) \subset B\} = P\{S_{E^1}(A) \subset B\}$. Also, because $S_{E^0}(A) \subset S_{E^1}(A)$ by construction, $P\{S_{E^1}(A) \subset B\} \leq P\{S_{E^0}(A) \subset B\}$. Therefore, the following lemma holds.

*Lemma A.2:*

$$P\{S_E(A) \subset B\} \leq P\{S_{E^0}(A) \subset B\}, \quad \forall A \subset U, \quad \forall B \subset V.$$

*Proof of Lemma A.1:* Because $P_{\mathcal{U}}$ is not decreased if additional packets are added, it is enough to show that

$$\max_{\mathcal{U}:|U|=\alpha M} P_{\mathcal{U}} \to 0, \qquad \text{as } M \to \infty.$$

Hall's Marriage Theorem allows $P_{\mathcal{U}}$ to be written in terms of the shadows of subsets of $U$. For arbitrary $\mathcal{U}$ with $|U| = \alpha M$,

$$\begin{aligned}
P_{\mathcal{U}} &= P\{\exists A \subset U, \exists B \subset V \text{ with } |B| = |A| - 1 \\
&\qquad \text{and } S_E(A) \subset B\} \\
&\leq \sum_{A,B:|B|=|A|-1} P\{S_E(A) \subset B\} \text{ by union bound} \\
&\leq \sum_{A,B:|B|=|A|-1} P\{S_{E^0}(A) \subset B\} \text{ by Lemma A.2} \\
&= \sum_{i=1}^{\alpha M} \binom{\alpha M}{i} \binom{\frac{3}{4}M}{i-1} \left(\frac{4i}{3M}\right)^{3i} \\
&\leq \sum_{i=1}^{\alpha M} \left(\frac{\alpha M e}{i}\right)^{i} \left(\frac{\frac{3}{4}Me}{i}\right)^{i} \left(\frac{4i}{3M}\right)^{3i} \\
&\qquad \text{by Stirling's bound} \\
&= \sum_{i=1}^{\alpha M} \left(\frac{16\alpha e^2 i}{9M}\right)^{i} \\
&= C/M + \sum_{i=2}^{\alpha M} \left(\frac{Ci}{M}\right)^{i}, \qquad \text{where } C = \frac{16\alpha e^2}{9}.
\end{aligned}$$

The ratio of consecutive terms in the last sum above is $(C/M)((i+1)/i)^i(i+1)$. This ratio is positive and increasing

in $i$. Thus the maximum term in the last sum is either the $i = 2$ term or the $i = \alpha M$ term. Therefore

$$P_{\mathcal{U}} \leq C/M + (\alpha M - 1) \max \{(2C/M)^2, (C\alpha)^{\alpha M}\}.$$

Because $C\alpha < 1$ and $\mathcal{U}$ is arbitrary, the lemma follows.

*Proof of Theorem 4.1:* Lemma A.1 shows that the probability of an arbitrary packet being successful in phase 1 can be made arbitrarily close to 1 for large enough $M$. If successful, a packet's access delay is no more than $d_{\text{acc}} + \frac{3}{4}M$. If not, then a packet is scheduled in phase 2 and suffers additional delay no more than $d_{\text{prop}} + d_{\text{sync}} - \frac{1}{2}M$.

Fix any $\epsilon > 0$. Choose $M$ large enough such that

$$\max_{\mathcal{U}:|U|=\alpha M} P_{\mathcal{U}} \leq \epsilon/3 \tag{3}$$

which is possible by Lemma A.1 and suppose that $d_{\text{prop}} \geq 9M/\epsilon$. Consider any $(\alpha, S)$ arrival sequence and let $p$ be an arbitrary packet in the sequence. Equation (3) implies that all packets that arrive during the same frame as $p$ (including $p$) are successful in phase 1 with probability at least $1 - \epsilon/3$. The expected access delay of packet $p$ is then

$$\begin{aligned} d_p &\leq E[\text{Phase 1 delay}] + (\epsilon/3)E[\text{additional phase 2 delay}] \\ &\leq d_{\text{acc}} + \frac{3}{4}M + (\varepsilon/3)\left(d_{\text{prop}} + d_{\text{sync}} - \frac{1}{2}M\right). \end{aligned}$$

Recall that $d_{\text{sync}} \leq 3M/4$ and $d_{\text{acc}} \leq 2M$, so that

$$d_p \leq d_{\text{prop}}\left(\frac{3M}{d_{\text{prop}}} + \frac{\epsilon}{3} + \frac{\epsilon M}{12 d_{\text{prop}}}\right).$$

The last three terms are smaller than or equal to $\epsilon/3$ and therefore $d_p \leq \epsilon d_{\text{prop}}$. The maximum access delay is at most $d_{\text{acc}} + d_{\text{prop}} + d_{\text{sync}} + M/4 \leq 3M + d_{\text{prop}} \leq (1 + \epsilon)d_{\text{prop}}$. This completes the proof of Theorem 4.1.

If source conflicts are ignored, then Lemma A.1 holds for $\alpha \leq 1/2$ according to [7]. The proof technique used there does not allow the application of Lemma A.2. It is conjectured here that Lemma A.1 is true for $\alpha \leq 1/2$ with source conflicts removed, because removal only increases the size of shadows. This yields $\alpha \leq 1/3$ as a sufficient condition for Theorem 4.1 with the conservative approach. It is also clear that $\alpha < 1/2$ is necessary for the two-phase approach followed here if each packet is to be sent twice in phase 1.

## REFERENCES


[1] M. de Prycker, *Asynchronous Transfer Mode.* New York: Ellis Horwood, 1991.

[2] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Commun. Mag.*, vol. 31, pp. 78–88, Feb. 1993.

[3] M. Chen, N. Dono, and R. Ramaswami, "A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks," *IEEE J. Selected Areas Commun.*, vol. 8, pp. 1048–1057, Aug. 1991.

[4] N. Dono, P. Green, K. Liu, R. Ramaswami, and F. Tong, "A wavelength division multiple access network for computer communication," *IEEE Selected Areas Commun.*, vol. 8, pp. 983–994, Aug. 1991.

[5] L. Zhang, "A new architecture for packet switching network protocols," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1989.

[6] T. Weller and B. Hajek, "Scheduling nonuniform traffic in a packet switching system with small propagation delay," in *IEEE INFOCOM 94*, June 1994. Full version to be submitted to *IEEE/ACM Trans. Networking.*

[7] R. Cruz and B. Hajek, "Global load balancing by local adjustments," in *Proc. 19th Annual Conf. on Information Sciences and Systems* (Johns Hopkins University, Baltimore, MD, 1986), pp. 448–454.

[8] D. Walkup, "Matchings in random regular bipartite digraphs," *Discrete Math.*, vol. 31, pp. 59–64, 1980.

[9] E. Shamir and E. Upfal, "One factor in random graphs based on vertex choice," *Discrete Math.*, vol. 41, pp. 281–286, 1982.

[10] I. Rubin and Z. Zhang, "Message delay analysis for TDMA schemes using contiguous-slot assignments," *IEEE Trans. Commun.*, vol. 40, pp. 730–737, Apr. 1992.

[11] P. Hall, "On representatives of subsets," *J. London Math. Soc.*, vol. 10, pp. 26–30, 1935.

[12] I. Gopal, D. Coppersmith, and C. Wong, "Minimizing packet waiting time in a multibeam satellite system," *IEEE Trans. Commun.*, vol. COM-30, pp. 305–316, Feb. 1982.