Token-Based Protocols for High-Speed Optical-Fiber Networks

MARIO GERLA, MEMBER, IEEE, PAULO RODRIGUES, AND C. W. YEH, FELLOW, IEEE

Abstract—The local network medium is a pair of unidirectional fiber-optic busses to which stations are connected via passive taps. For this configuration, we present several protocols which provide round-robin, bounded delay access to all stations, and are particularly suited for high-speed transmission. The common characteristic of the protocols is the use of the token as the synchronizing event to schedule transmission. The token may be explicit (as in U-Net) or implicit (as in Tokenless Net). It may be used all the time, or it may be used simply to resolve collisions (as in Buzz-Net). The protocols are shown to be cost effective at very high (bandwidth) x (length) products that are the unique characteristic of high-speed single-mode fiber networks. Furthermore, they are robust to failures because of the passive interfaces and the totally distributed control. The implementation of these protocols on fiber-optic busses is also discussed in the paper.

I. INTRODUCTION

RECENT years have witnessed a rapid growth in local area communications needs corresponding to the increasing sophistication of users and the emergence of new applications such as real-time voice and video, high-speed printers, graphics, etc. These environments require systems able to handle very high throughput among users several kilometers apart, while satisfying delay constraints which become very tight when real-time traffic is involved. The answer to these demands is a suitable choice of transmission medium, topology, and access protocol.

Among the various transmission media, optical-fiber appears to be the most promising technology. Fiber offers an extremely high bandwidth x length product, up to 10 GHz x km for a single-mode operation. That is, a 10-GHz signal transmitted over 1-km of fiber without intermediate regeneration can be properly received using state-of-the-art transmitters and receivers. Other attractive features of the optical fiber include immunity against electromagnetic interference and protection against signal leakage. Fiber is easy to install due to its light weight and small size, and has lower attenuation than coaxial cable.

Fiber-optics topologies can be configured in three basic ways: ring, star, and bus. Ring implementations require active repeaters and substantial logic working at channel speed (e.g., address recognition and flag setting) at each station. Their cost and reliability, in spite of some failsafe node proposals [1], may set a limit to the use of ring interfaces at very high speed. Furthermore, special procedures are needed to recover from token loss or duplication and to avoid nonrecognized packets circulating endlessly in the ring.

Star topologies can be built by connecting several stations via a star coupler. Star configured networks have been implemented using either passive [9] or active [10] components. Most star networks use the CSMA-CD access protocol, which was successfully experimented in ETHERNET. However, CSMA-CD performance becomes very poor when the end-to-end propagation delay on the bus is comparable to (or larger than) the packet transmission time [16]. Consequently, the effort required to push the technology to higher channel bandwidth is rewarded only by a marginal improvement in network throughput.

The third alternative is the linear bus to which stations are coupled via passive interfaces. Since optical couplers are intrinsically unidirectional, a unidirectional bus system (UBS) must be used.

In a unidirectional bus signals propagate in only one direction. Thus full connectivity among stations is achieved by using either two separate busses with signals propagating in opposite directions (twin bus implementation), or one bus folded to visit all stations twice.

Fig. 1 Twin unidirectional bus.

Fig. 2 Folded unidirectional bus (Express-Net).
If two separate pairs of unidirectional busses are used, two independent transmitters and receivers (one for each bus) are required for each interface (see Fig. 1). If a folded bus is used (as in Figs. 2 and 3), one section of the bus is used for transmission and another is used for reception. Therefore, only one transmitter and one receiver are required. In addition, a sensor is usually installed in front of the transmitter to permit carrier sensing. As we shall see, carrier sensing is an essential feature in UBS protocols. In a twin bus implementation, carrier sensing is provided directly by the receiver, making a separate sensor unnecessary.

In recent years, extensive research has been conducted on UBS local area networks, and several efficient protocols have been proposed. Express-net [2], based on a folded unidirectional bus (Fig. 2), achieves conflict-free transmissions and bounded delay by means of a roundrobin oriented access protocol. C-Net [8], based on the topology shown in Fig. 3, achieves high throughput efficiency and bounded delay by employing a protocol which combines implicit token and random access schemes. D-Net [18] also uses the topology shown in Fig. 3. In D-Net, the right most station has the responsibility for generating explicit tokens, while in the first two schemes the implicit token (or, better, the synchronizing event) can be generated by any station.

Several schemes have been proposed for the twin bus topology. Fasnet [6] is a synchronous slotted network with end stations (the right-most and the left-most station) responsible for slot generation and bit synchronization. DCR Net [14] employs a deterministic contention resolution scheme. The normal mode of operation is random access CSMA-CD. Once a collision occurs, it is resolved using an implicit token passing scheme.

In this paper, we will focus on three protocols for twin unidirectional bus systems which were recently developed at UCLA, namely, U-Net, Buzz-Net, and Token-less Net. The protocols are described in detail, and their performance is compared to that of similar systems. Implementation aspects regarding the optical/electrical interfaces, the optical coupler ratio’s, and the dynamic range of the receivers are also addressed.

II. U-Net Protocol

A. Basic Operation

Unidirectional Network (U-Net) is a local network consisting of two unidirectional busses [5]. Stations are connected to the busses via passive taps (see Fig. 1). Each tap includes a receiver and a transmitter.

The receiver detects presence/absence of carrier. When carrier is present, the receiver attempts to acquire bit synchronization from the preamble. After acquisition, the receiver copies bus data into private memory.

The transmitter sends a preamble followed by the data packet after it has received the go-ahead by the access protocol. If the station senses carrier coming from upstream while transmitting, it aborts its own transmission and tries again at the end of the incoming packet.

We assume a reaction delay of $d$ seconds between the time a station senses end of carrier on the bus and the time it can start transmission on the same bus. Likewise, there is a $d$ second delay between the sensing of carrier coming from an upstream station and the interruption of an ongoing transmission.

As we shall see, the above functions are common to all UBS interfaces. Actual UBS protocols differ from each other in the way they use these basic functions to provide access scheduling and synchronization.

The U-Net protocol consists of two procedures. The first procedure, described in this section, defines access to the bus after the end stations have been elected and the token mode has been established. The second procedure, introduced in the next section, defines the election of end stations at network initialization and/or network configuration change.

The following describes the token mode of operation used in U-Net. The two end stations are defined as $L$ (left) and $R$ (right). Protocol operation can be viewed as a sequence of cycles. Each cycle is initiated by one end station, for example, $R$ station. $R$ sends a special bit pattern, called token, on the $R$-to-$L$ bus. This token is followed by a data packet from $R$ (if $R$ has data to send). Token and packet are separated by a gap on the order of $d$ seconds.

Each station continuously monitors both busses for a token. Once the token is heard on a bus (henceforth referred to as the token bus), the station is allowed to transmit one packet on both busses. More precisely, immediately after hearing the token, the station begins transmitting the preamble on the token bus. If, after an interval $d$ from the beginning of its transmission, the station does not hear conflict on the bus (conflict may occur if an upstream station on the token bus is also attempting to transmit), it proceeds transmitting the preamble on the token bus as well as on the reverse bus (i.e., the bus in the opposite direction). If conflict is detected (i.e., the station hears another preamble coming in from upstream while it is transmitting its own), the station aborts its transmission on the token bus and does not even attempt to transmit on the reverse bus. The station will restart transmission after the oncoming packet has passed by. This procedure is called “probing” the token bus.

On the token bus, packets are appended to the token in the same way as cars in a train are appended to a locomotive. Each station has the chance to transmit in the train, and can transmit at most one packet. Packets on the bus are separated by gaps of size $d$. On the reverse bus, a similar train is also formed. However, packets are not preceded by a token; rather they are separated by larger gaps than the packets on the token bus. The size of the gap
between two packets on the reverse bus is equal to twice the propagation delay between the two sending stations, plus 2$d$. Fig. 4 shows the space-time diagram for a possible sequence of packets on the token bus and on the reverse bus. Each point in the diagram represents the position of the packet in one of the busses at a given time. A snapshot of the system is also shown.

Another difference between the token bus and the reverse bus is that on the token bus the initial $d$ seconds of the preamble may be damaged by conflicts. In fact, if the train carries $N$ packets, the first $k$ bits of the preamble (where $K = dC$, and $C =$ bus speed) in the first packet correspond to the superimposition of $N - 1$ preambles. The preamble must be large enough to allow bit sync to be acquired despite initial garbage. To simplify preamble acquisition, the beginning of the packet could be simply masked off by the receiver.

It is important to note that each packet transmission is heard by all stations exactly once. Assuming the R-to-L bus is the token bus (see Fig. 1), the packet transmitted by station $i$ is received by station $i + 1, i + 2, \ldots$, and $N$ on the token bus, and by station $i - 1, i - 2, \ldots, 1$ on the reverse bus. The transmission mode is implicitly a broadcast mode; specific knowledge of the destination station is unnecessary to properly route the packet.

The cycle terminates when the train terminates, i.e., when all the stations, including $L$ and $R$, have had the opportunity to send their packets. The $L$ station detects the end of the train from the absence of carrier for more than $2d$ s at the end of a packet (or token). After detecting the end of the train and (possibly) transmitting its own packet, the $L$ station declares the cycle closed and starts a new cycle in the reverse direction by injecting a token in the reverse bus, which becomes the new token bus. The operation is the same with the roles of token bus and reverse bus interchanged.

B. End Station Election

U-Net is equipped with a dynamic procedure for electing end-stations. This procedure provides automatic recovery from station failure from token loss, without operator intervention. It also permits smooth insertion of new stations in the system.

$R$ is defined as the round trip propagation delay on the fiber cable. $T_{MAX}$ is the maximum size packet transmission time. $t_0$ is the time required by an end station to “turn around” the token (read if from one bus and inject it onto the other bus).

Next, some observations. During normal token mode operation there are short gaps between packets within each train, and larger gaps between trains. The distance between gaps is $T_{MAX}$, by definition. If a continuous data stream of duration $> T_{MAX}$ is detected, it is interpreted as an anomaly. This property is exploited in the election procedure. As a second observation, the maximum duration of a silence gap at a station (the time during which both busses are sensed idle) during token mode operation is $R + t_0$. A larger silence gap denotes an abnormal situation (e.g., a failure or token loss).

The following describes the end station election procedure. During this procedure each station moves through the states shown in the state diagram in Fig. 5.

During normal operation each established (as opposed to new entering) station is found in the token mode state. Operation in this state was described in Section II-A. From this state a station moves to the buzz mode state if it ob-
serves a silence gap $\geq R + t_0$, or it senses continuous signal for an interval $> T_{\text{MAX}}$.

In the buzz mode state a station issues a buzz tone on both busses. As a possible implementation, this buzz tone could consist of a preamble repeated continuously without gaps. During buzz mode a station defers to upstream stations by aborting its buzz tone when a buzz tone arrives from upstream.

After an interval $R + T_{\text{MAX}}$ from the time the first station entered buzz mode, all stations are necessarily in buzz mode. At this point, there are three possible conditions in which a station can be found.

a) The station has deferred on both buses. In this case, the station is an intermediate station (i.e., not an end station.) It moves thus to the wait for token state. In this state, the station remains silent, awaiting for the token.

b) The station is still transmitting on one bus (and has deferred on the other because a busy tone was detected or the bus is busy). The station is an end station and moves to the end station selection state, where one of the two end stations is selected to start the token cycle.

c) The station is transmitting on both busses. This implies that there is only one station on the bus! The station moves to the new station state (to be defined later).

In the end station selection state the newly elected end stations must decide which one should start the token cycle. To accomplish this, each station replaces the buzz tone with its ID number. The elected end stations compare the respective ID numbers. Using the rule that low ID starts the cycle, the high ID station moves to the wait for token state, and the low ID station moves to the starting end station state, waiting for the reverse bus to become idle. It then issues a token and moves to the token mode state.

Upon hearing the token, all other stations move from the wait for token state to the token mode state.

A new entering station finds itself initially in the new station state. From this state, it must detect the token on both busses before moving to the token state. If a token is heard twice on the same bus, but not on the other bus, the station is the new end station. Thus, it moves to buzz mode to trigger a new selection. Likewise, the station moves to buzz mode if a silence gap $> R + t_0$ is detected. This may happen at system initialization.
The election procedure may appear somewhat elaborate, but it is quite efficient. It requires approximately an interval \(3R + T_{\text{MAX}} + t_o\) to recover from failures. Typically, this is in the order of fractions of a millisecond for channel speeds over 100 Mbps. The procedure is robust to any sort of failure. Even failures that occur during the recovery procedure are detected and recovered from.


One of the drawbacks of U-Net is the latency delay incurred by a station waiting for the token to come by. This delay is particularly annoying if there is only one station sending traffic (say, a large file) while all the other stations are idle. In this case U-Net forces the transmitting station to send only one packet at a time for each round-trip interval—a fairly inefficient proposition at very high bandwidth \(x\) length products.

This inefficiency can be overcome using a hybrid random access/implicit token protocol, which operates in random access mode when the system is lightly loaded (or a single station in transmitting) and reverts to implicit token mode when multiple source traffic builds up. This scheme we called Buzz-Net, because of the "buzz" pattern used to synchronize the system and drive it to token mode [3].

A. The Buzz-Net Algorithm

The network can operate in either of two modes: random access, or controlled access mode. Initially, a station starts in the idle state of the random access mode (see the state diagram in Fig. 6). When a packet arrives, the station moves to the backlogged state. From this state, transmission of the packet is attempted in random access mode.

a) If both busses are sensed idle, the station moves to the Random Access Transmission state. In this state, packet transmission immediately begins on both busses (it is assumed that the sender does not know the relative position of the destination on the bus).

b) If one bus is idle and one is busy, the station moves to the Wait for EOC state. Here, the station waits for end-of-carrier (EOC) on the busy bus.

c) If both busses are sensed busy or a buzz pattern is sensed, the station moves to the Buzz-I state, which is part of the controlled access procedure.

In the Random Access Transmission state the station proceeds transmitting on both busses. If, while transmitting, it is interfered by an upstream station (that is, it hears a Begin-Of-Transmission (BOT) on one of the busses) it aborts its transmission and moves to Buzz-I state. The upstream transmission is allowed to proceed intact. If the transmission is successfully completed, the station moves to Idle state.

In the Wait for EOC state, when EOC is sensed, the station moves to Random Access Transmission state. If, while in Wait state, the station senses a buzz pattern or it senses both busses busy, it moves to Buzz-I state.

While in the random access mode a station with several packets ready for transmission may attempt to send them all in a single train, cycling between Backlogged and Random Access Transmission states, and thus capturing the channel and locking out the other stations. To avoid capture, a minimum interpacket gap must be observed between any consecutive packet transmissions. This minimum gap, on the order of a station reaction time interval \(d\), allows downstream stations in the Wait for EOC state to detect EOC inside a train and, upon collision, force the system to controlled access mode, thus breaking capture.

In Buzz-I state a station transmits the buzz pattern on both busses (deferring, of course, to upstream transmissions) for \(R\) seconds, where \(R = \text{round-trip delay}\). Because of deferrals, a station in the buzz state may actually buzz the busses only intermittently (or it may not buzz them at all). After \(R\) seconds, the station moves to Buzz-II state.

In Buzz-II state the station buzzes only the Left-to-Right bus, deferring as usual to upstream stations, until it hears no more buzzing on either bus. At this point, the station moves to Controlled Access Transmission state. The intermediate Buzz-II state guarantees that the leftmost (and only the leftmost) station starts the controlled access cycle when all the right-to-left buzzing has ceased.

In Controlled Access Transmission state each station is allowed to transmit its backlogged packet, and to move to Hold state thereafter. Controlled mode transmission is carried out much in the same way as in U-Net. As already discussed in U-Net, the left-to-right bus must be probed.

![Fig. 6. Buzz-Net state diagram.](image-url)
before transmission. That is, a station waits for the left-to-right bus to become free. Then it probes this bus by starting transmission of the preamble on it. If it does not hear upstream interference within a reaction time interval \( d \), it will then proceed to transmit the remaining preamble also on the right-to-left bus, followed by the data packet. If interference is sensed, the station aborts its transmission and retries when the left-to-right bus is free again (i.e., after EOC is sensed). If a buzz is heard, the station moves back to Buzz-I state.

Clearly, in the controlled access mode, a “train” of packets is formed from left to right. Backlogged stations are allowed to append their packets to the train in a left to right order. At the end, all stations end up in the Hold state.

A time-out \( T_0 \) from Controlled Access Transmission state to Idle state is provided to prevent a new station entering the system from locking up other stations finding themselves in Controlled Access Transmission state. A time-out \( T_0 \) is also provided for the Hold state for similar reasons. A more detailed description of the recovery procedures when a new station joins the network is given in Section III-C below.

B. Buzz Signal Implementations

The buzz signal is a signal (or event) clearly distinguishable from regular packet flow. If the preamble pattern is uniquely distinguishable even when embedded in other data, then a possible buzz implementation consists of sending a prolonged preamble pattern. In this case, bit stuffing may be required to maintain data transparency within each packet.

An alternative buzz implementation which does not require bit stuffing consists of enforcing a minimum gap \( \Delta T \) between any two consecutive packets on the bus. \( \Delta T \) should be large enough so that a station in buzz mode can fill the gap with a burst of (arbitrary) data. The buzz implementation consists of sending one (or more, for reliability purposes) short burst(s) of unmodulated carrier where the length of a burst is less than the smallest packet size, but large enough to be safely detected by a station. A station recognizes the buzz condition when it detects on either bus the presence of a burst shorter than the minimum packet length. This scheme provides a faster detection of the buzz signal.

In general any method which permits some form of in-band signaling is a feasible buzzing method. The best method will most likely depend on interface implementation considerations, and may vary from application to application.

C. New Stations Joining The Network

A newly activated station may join the network at any time. In some cases, the joining process occurs transparently. In other situations, activity of the new station forces a transient phase which adds extra delay to the transmission in process. However, whatever the case, the new station does not cause permanent disruption of network operation, and the access algorithm automatically absorbs the external interference.

If the network is operating in random access mode, the new station is absorbed transparently. If, on the other hand, the network is in controlled access mode the new station will either move to Buzz-I state and participate in a new buzzing phase, or it will successfully transmit its packet after both busses are sensed idle. The first situation occurs because a buzz is detected by the new station, or its transmission collides with a transmission of a backlogged station. In the worst case (stations 1 and \( N \) participating in the new buzzing phase), an extra \( 2R \) interval may be necessary before transmissions are resumed. The second situation develops when the new station senses the right-to-left bus busy due to packet transmission by the next backlogged station situated upstream on that bus. Upon detecting end of transmission, the new station transmits on both busses and moves to Idle state. Other backlogged stations do not perceive this intrusion and behave normally.

Theoretically, there is still a chance that the new station will keep transmitting in random access mode after reaching the Idle state. In this case, any station that has previously moved to Hold state will eventually time out and move back to Idle. Nevertheless, if these stations do not have any packet to transmit, the new station will continue to lock the remaining stations in controlled access mode. The time out \( T_0 \) in the Controlled Access Transmission state prevents this capture effect.

As we have shown, the new station joins the set of active stations gracefully. The extra delay added to controlled access mode delay is in the best case 0, between 0 and \( 2R \) in most cases, and of the order of \( T_0 \) in very unlikely worst case situations.

IV. Token-Less Protocol

The Buzz-net protocol is quite successful in eliminating token latency at light load. When load becomes substantial, however, the delay performance degrades, and in fact becomes worse than the performance observed in U-Net, due to the overhead caused by the continuous switching between random access and controlled access mode.

The attempt to retain the efficiency of U-Net at heavy loads and, at the same time reduce token latency and eliminate the need for special token initialization procedures, led to the creation of the Token-less protocol [11]. This protocol, as the name indicates, does not use an explicit token. Instead, it supports a round robin access mode (of the type seen in U-Net) by means of a synchronizing event which can be viewed as an “implicit” token. The synchronizing event is represented by the ceasing of activity on one of the busses. This event when detected authorizes a station to “probe” the other bus and append its packet at the end of the “train” that has been forming
on that bus. The details of the protocol are described below.

A. Principles Of Operation

In a tokenless network each station is connected to each bus with two passive taps, a receiver tap and a transmit tap. Through the receiver tap a station can receive all packets flowing in that bus and can monitor channel activity. More specifically it can observe presence or absence of activity (i.e., signal) and detect events as End of Activity EOA and Beginning of Activity BOA. As usual, the detection of these events occurs within the delay \( d \) which is the station reaction time.

It is also assumed that when a station detects a BOA event on either bus it stops transmission if any. That is, a station engaged in transmission always defers to an upstream transmission by aborting its own. The upstream transmission proceeds with only the first bits of the preamble corrupted no matter how many downstream stations are attempting to transmit. This guarantees that if the preamble is long enough a packet which has been completely transmitted by a station is correctly received by all (downstream) stations. We assume that an interface is able to correctly receive a packet even when this packet is immediately preceded by some truncated transmission. The underlying assumption is that the beginning-or-packet flag cannot be replicated within the packet data nor is contained in the activity signal (to be described below). Flags can be implemented as reserved bit patterns (in which case bit stuffing is required to preserve data transparency) or as code violations on the bit encoding level.

The transmit tap is used by a station to transmit either (data) packets or activity signal. As we shall see later, the purpose of the activity signal is to keep the downstream part of the channel busy. Its implementation (modulated or unmodulated carrier, random bits, continuous sequence of 1's, etc.) can be chosen according to the level encoding utilized for transmission on the channel.

The goal of the protocol described in the following section is to guarantee collision-free transmissions among all stations with backlogged packets and to achieve good throughput/delay performance. Furthermore the need for special packets (e.g., token) is avoided and no central control is introduced. These characteristics are achieved by having EOA's events propagating alternatively in the two busses. One advantage of controlling the channel only through EOA events is to provide simple, reliable and low cost implementation even at very high speed. Another advantage, as we shall see later, includes easy implementation of initialization and recovery procedures for the protocols.

The EOA events can be seen as virtual tokens which allow the stations to transmit their packets in a round-robin fashion. These EOA events, detected simply by sensing the channel, propagate from left to right in one bus and from right to left in the other, thus minimizing the silent gap between the end of a round, and the beginning of the next.

B. The Basic Token-Less Protocol

The protocol basically consists of three procedures. The first procedure enables a station to recognize when it is its turn to transmit in a round. The second procedure enables a station to determine whether it is an extreme (left most or right most) active station. The third procedure enables a station which has been just powered-on to synchronize with other active stations, if any, or to initialize the round-robin cycle in an empty net. Part of this procedure also allows for recovery from failures caused by stations transmitting at the wrong time because of detection of false events.

There are different parameters and options which may be chosen when specifying the full protocol and in the following we will present in details three different implementations. Each one has some distinctive advantages depending on the traffic environment. Before getting into the details of the different implementations, we present the common foundation of the various versions of the Token-Less protocol.

A station, say \( S_i \) with a backlogged packet waits for the first of two events: EOA on either channel or time-out ND (Network Dead).

If an EOA (\( A \)) on channel \( A \) occurs first \( S_i \) starts transmitting activity signal on channel \( A \). If BOA(\( A \)); occurs, stop transmission and wait for next EOA(\( A \)), otherwise after time-out \( d \), start packet transmission on both channels. When packet transmission is completed \( S_i \) keeps transmitting activity signal on channel \( A \) until either a BOA(\( A \)) is detected or time-out Extreme Station (ES) occurs. If BOA(\( A \)) is detected then \( A \leftarrow \bar{A} \) and repeat the above procedure. Otherwise if ES is reached, station \( S_i \) realizes that it is an extreme station. \( S_i \) then undertakes the round restart procedure.

If time-out ND occurs (meaning that no other station is active in the network) \( S_i \) undertakes the Initialization procedure.

The flow diagram of the Basic Token-Less protocol is shown in Fig. 7.

C. Variations On The Basic Theme

As previously mentioned there are several ways to specify Round Restart and Initialization procedures and to choose the parameters ND and ES. Some of these are described in the following.

The state diagram shown in Fig. 8 defines the operation of TLP-1 (Token-Less protocol, Version 1). Explanation of the notation used in state diagrams is given in Table 1. The states in the left side of the figure represent the Initialization procedure. A station is in these states only when it is powered-on. The time-out ND is set equal to a round-trip delay \( R \). If the time-out ND is not reached, either because one channel is sensed busy or a BOA is detected, the station leaves the initialization state and reaches the WFT state. The Round Restart procedure is performed in state ES. In this state, the station transmits activity signal...
Fig. 7. Flow diagram of the basic token-less protocol.

Fig. 8. State diagram of TLP-1.
in the direction opposite to the one where the virtual token is propagating. If the activity transmission lasts \( ES = R \) s the station knows it is an extreme station and thus behaves accordingly. The dotted portion of the state diagram of Fig. 8 represents the recovery procedure which is undertaken any time an illegal event occurs.

An example of the operation of TLP-1 for a network with 10 stations is given in the space-time diagram shown in Fig. 9. The time intervals \( A, B, C, D, \) and \( E \) represent rounds. In round \( A \) the virtual token propagates from left to right. Stations 1, 3, 4, 5, 7, 8, 9, and 10 are powered-on, and stations 1, 7, and 8 have backlogged packets. In the next round, \( B \), the virtual token propagates from right to left. A new station, station 6, is powered-on, and stations 6, 4, and 1 transmit a packet, and so on.

In TLP-1 a powered-on station always performs some activity on the channel even if it has no packets to transmit. This implies that the virtual token at each round goes back and forth between the two extreme powered-on stations. Thus any station can transmit a packet within a round. If traffic load is unbalanced and only a few stations are actually sending traffic this mode of operation introduces an unnecessary delay due to the fact that the virtual token must sweep the entire bus, rather than the section of the bus containing the stations involved in transmissions.

This inconvenience is eliminated in TLP-2 where the virtual token sweeps only the stations which have a packet to transmit. This is achieved, as shown in the state diagram of Fig. 10, by forcing a station with no backlogged packets back to the \textit{Idle} state. All the other parameters are the same as in TPL-1. The behavior of TLP-2 for the same traffic pattern as in the previous example is shown in the space-time diagram of Fig. 11.

A substantial contribution to the overhead in both TLP-1 and TLP-2 is given by the time-out \( R \) between rounds. One way to reduce this delay is to set two time-outs \( ES_1 \) and \( ES_2 \) at each station equal to the round-trip delays from the station to either end of the busses. The appropriate time-out (\( ES_1 \) or \( ES_2 \)) is then used when performing the \textit{Round Restart} procedure. This method, however, introduces some additional complications when the network is installed and requires \( ES \) parameter updating at each station if extension are added to the bus.

Another method which is solely based on the protocol and not on the physical layout of the network is to take advantage of the fact that, in TLP-1, the extreme stations do not change very frequently. Thus, if a station is the right most station, say, in a round, it will very likely be again the right most station in the next round. Therefore, as soon as it has been granted access to the channel in one round it starts right away a new round in the opposite direction. This scheme, called TLP-3, eliminates the \( R \) seconds gap between rounds and works properly until a new station external to the present extreme stations is powered-on. In this case a collision is generated and the initialization procedure is invoked to restore the correct operation with the new powered-on station as an extreme station. Details of the algorithm TLP-3 are given in the state diagram of Fig. 12 where a flag \( E(A) \) is used to signal if a station is an extreme one or not. The space-time diagram in Fig. 13 shows how this version works for the same example previously considered.

\section{V. Performance Analysis}

In this section we evaluate the performance of the various access protocols introduced so far and compare it with that of two other schemes that have been extensively studied, namely, Express Net and CSMA-CD [17].

The following performance measures are considered in this analysis:

1) bus utilization \( U(i) \) at heavy load, defined as the net bus utilization when \( i \) stations are active and have infinite packet backlog;

2) average insertion delay defined as the interval between the time when the packet moves to the head of the transmitting queue, and the time when successful transmission begins. Note that insertion delay is equivalent to queueing delay when there is only one buffer per station. We distinguish between insertion delay at light load (IDL), and insertion delay at heavy load (IDH).

The above measures, albeit simple, provide us with useful criteria to decide whether a bus protocol is suitable or not for a given application. For example, interactive and real time applications are particularly sensitive to insertion delays. Batch data transfer on the other hand is mostly affected by bus throughput efficiency.
In our analysis, we assume that stations are uniformly spaced on the bus. For simplicity we also assume that detection time \( d \) is much smaller than the end-to-end propagation delay \( \tau \), and that the preamble is much shorter than the packet.

A. Bus Utilization

We begin by evaluating bus utilization \( U(N) \) at heavy load. For token based protocols under equilibrium conditions, \( U(N) \) is given by the ratio between the time in a round spent for packet transmissions and the round duration, given that \( N \) stations are active and always transmitting in each round. For CSMA-CD (which is not a token based protocol), the utilization will be computed separately.

The total packet transmission time during a round is the same for all schemes and is given by \( NT \) (where \( T = \) packet transmission time) since there are \( N \) backlogged stations and each station transmits in each round.

As for the round duration \( d \), this varies from scheme to scheme. In U-Net we have

\[
D_{U-Net} = NT + \tau.
\]

In Buzz-Net, under heavy load conditions, the active stations will always conflict again at the end of a controlled phase. Therefore, the activity in the network is a succession of cycles where active stations are served in a round-robin way, lowest numbered stations first. The diagram in Fig. 14 portrays the cyclic pattern when all \( N \) stations are active. From Fig. 14 we can see that

\[
D_{Buzz-Net} = NT + 6\tau
\]

In TLP-1 and TLP-2, we have

\[
D_{TLP-1,2} = NT + 3\tau.
\]

In TLP-3 we have the same results as in U-Net, namely,

\[
D_{TLP-3} = NT + \tau.
\]
Taking now the ratios of packet transmission time during the round and round time we obtain the following expressions for bus utilization:

**U-NET:** \[
\frac{1}{1 + a/N}
\]

**Buzz-NET:** \[
\frac{1}{1 + 6a/N}
\]

**TLP-1,2:** \[
\frac{1}{1 + 3a/N}
\]

**TLP-3:** \[
\frac{1}{1 + a/N}
\]

**Express-Net:** \[
\frac{1}{1 + 2a/N}
\]

where \(a = \tau/T\).

For CSMA-CD, bus utilization can be approximated by [17]

\[
U = \begin{cases} 
\frac{1}{1 + 5a} & \text{for } a < 0.5 \\
1/7a & \text{for } a > 0.5 
\end{cases}
\]

Note that CSMA-CD maximum utilization is not affected by the number of stations, as long as this number is fairly large.

The results are reported as a function of \(a\) in Fig. 15 for \(N = 15\). The best performance is shown by TLP-3 and U-Net. TLP-1 and TLP-2 perform slightly worse than Express Net, but better than Buzz-Net. CSMA-CD performs very poorly for larger \(a\), as expected.

It should be pointed out, however, that there is one important case in which Buzz-Net outperforms all other token protocols. This is the case of single station with heavy backlog. In Buzz-Net, the single station can transmit an uninterrupted sequence of packets thus yielding \(U = 1\). In the other schemes, subsequent packets from a single station are spaced by at least one round-trip delay.

**B. Insertion Delay at Light Load**

For light load conditions we assume that all stations are powered-on, and that the probability of two or more stations transmitting in a cycle is negligible.

It should be noticed that for U-Net and TLP protocols the token latency varies with the station position on the bus. It is minimum at the center, and it is maximum at the extremes. Correspondingly, there is a best and worst value.
of IDL for each network. In the following, we report both values.

The values of average insertion delay for the various schemes are shown below:

- **U-Net** \[ \frac{\tau}{2} \leq IDL \leq \tau \],
- **Buzz-Net** \[ IDL = 0 \],
- **TLP-1** \[ \frac{3}{2} \tau \leq IDL \leq \frac{5}{3} \tau \],
- **TLP-2** \[ IDL = 6\tau \],
- **TLP-3** \[ \frac{\tau}{2} \leq IDL \leq \tau \],
- **Express-Net** \[ IDL = \tau \],
- **CSMA-CD** \[ IDL = 0 \].

We note that the best performance is obtained with Buzz-Net and CSMA-CD, as expected. We also note that TLP-2 displays a very high insertion delay. This is due to the fact that at very light load, the protocol must be reinitialized for each transmission. This requires on the order of three round-trip delays (i.e., 6\tau), as can be seen from the diagram in Fig. 10.

**C. Insertion Delay at Heavy Load**

In heavy-load conditions we assume that a station has always a packet to transmit. The insertion delay is then obtained by computing the time required for the token to come back to the station, given that all the intermediate stations will transmit whenever they get the token. Again, an asymmetry is noticed in U-Net and TLP, in that each station will alternatively see a long cycle and a short cycle. However, one discovers that the two cycle times average out so that the insertion delay is the same for all stations.

The expressions for insertion delay at heavy load follow:

- **U-Net** \[ IDH = NT + \tau \],
- **Buzz-Net** \[ IDH = NT + 6\tau \],
- **TLP-1, 2** \[ IDH = NT + 3\tau \],
- **TLP-3** \[ IDH = NT + \tau \],
- **Express-Net** \[ IDH = NT + 2\tau \],
- **CSMA-CD** \[ IDH = \text{Unbounded} \].

From the results we notice that the delays of all the token based schemes are bounded—a key property due to the round robin scheduling. In contrast, the delay of the random access scheme, CSMA-CD, is unbounded.
Fig. 16 summarizes delay and utilization results in a single diagram, where delay is shown as a function of utilization for various access schemes. The network under study had the following characteristics:

- \( N = 15 \)
- \( \tau = 5 \, \mu s \)
- \( T = 1 \, \mu s \) (i.e., packet length = 1000 bits at 1 Gbps, or 100 bits at 100 Mbps)
- Poisson arrivals (heavy load)
- Exponential packet length distribution
- Infinite buffers at each station.

The data points in Fig. 16 were obtained via simulation. Exact analytic results are available only for light and heavy utilization. Thus approximate analytic models must be developed for intermediate values of utilization. One can verify that simulation results are in good agreement with the analytic results reported earlier for light and maximum utilization.

VI. FIBER IMPLEMENTATION CONSIDERATIONS

The various classes of token based protocols described in this paper can be applied to any unidirectional bus system regardless of the medium used (twisted pair, coaxial cable or optical fiber). However, the full potential of these protocols can be exploited only with optical fiber, for the following reasons:

The token based protocols permit to operate efficiently at much higher data rates than conventional carrier sense protocols. These higher data rates (above 500 Mbps, say) are economically achievable only with optical fibers.

The protocols permit efficient operation over large distances (more precisely, at very high Bandwidth \( \times \) Length products). Only optical fiber permits the operation over several miles without intermediate repeaters. Cable requires signal regeneration every few miles, thus
defying the goal of totally, fault passive tolerant bus
design.

It is therefore apparent that the full advantages of the
token-based protocols can be obtained only in fiber-optics
media. It is then appropriate to ask, whether the fiber
implementation poses technical problems which are not
present in cable implementations. Basically, there are three
areas in which the cable bus design differs from the optical
bus design, namely:

- optical/electrical interface,
- optical coupler ratios and power budget; and
- transmit and receive power calibration.

These issues are discussed in the following sections.

A. Optical/Electrical Interface

At low speeds, the design of the interface is very
straightforward: The baseband electric signal, instead of
being injected directly in the cable, modulates a laser or an LED; on the receive side, the baseband electric signal is obtained from the p-i-n diode. The electronics is the same as in cable implementations.

The challenge arises when we operate at speeds above 100 Mbps. While the optical components can operate very efficiently at higher speeds, the electronic components show serious limitations. Without entering in details which are beyond the scope of this paper, we mention the fact that optical components can be used to compensate for the "inadequacy" of electronic components. On the one hand, optical logic can be used to perform the most time critical functions of the interface, such as beginning and end of carrier detection, preamble acquisition, clock acquisition, address recognition, etc; On the other hand, passive optical delay lines permit serial to parallel conversion of the bit stream traveling on the bus. Namely, this stream can be subdivided into \(N\) substreams which are out-of-phase by exactly one bit among each other. Thus, data can be written to (and read from) the bus at a speed which is \(N\) times less than the speed of the bus.

Akin to the optical delay line is the concept of "fiber buffer loop" proposed in [15]. The fiber loop provides a circulating buffer for the data in transfer between a low-speed computer device and a high-speed optical bus. Optical switches control the connection of loop to bus and of loop to computer device. By properly clocking the switches, data can be "strobed" in and out of the optical loop at a period which is multiples of the loop delay. A reduction in speed of several orders of magnitude can thus be obtained.

One should also note that the serial to parallel conversion solution is more effective in token busses than in token rings. In the bus situation, in fact, the only function that must be carried out serially, at bus speed, is the detection of beginning and end of carrier. This function is very straightforward to implement, even at extremely high speeds. All the other functions can be carried out through serial-to-parallel conversion. In the ring configuration, on the other hand, each interface must perform several logical functions on line, at bus speed, (e.g., clock acquisition, address recognition, etc.). This implies that the ring interface is generally more complex to implement than the bus interface at very high speeds.

B. Optical Coupling

One well-known drawback of optical receivers is the fact that they require a certain amount of optical power in order to operate properly (while electrical receivers can be built with very high input impedance so that power absorption from the line is minimal). Consequently, optical taps introduce a much higher power loss than electrical taps, posing a severe constraint on the number of stations that can be connected to the bus.

In a fiber bus, optical taps are generally implemented with biconical couplers, which allow to couple a fraction \(C\) of optical power from one fiber to the other. In addition to the "coupling loss" \(C\), we must also account for an "excess loss" \(\beta\) corresponding to the fraction of optical power irradiated in the air at the junction.

Assuming that in the system under consideration the transmitters have maximum output power \(P_T\) and the re-
receivers can detect reliably a minimum power $P_s$, then the ratio $P_t/P_s$ is defined as the power margin $M$ for the system. Clearly, for proper system operation, the sum of all tap and attenuator losses (in dB) along the bus must be less than $M$. One important problem in fiber bus design is therefore the determination of the maximum number of taps (i.e., stations) that can be installed on the bus. This number generally will depend on the coupling ratio $C$ as well as on excess loss and attenuation.

Recent advances in single-mode fiber technology have driven excess loss below $0.1$ dB and attenuation below $0.2$ dB/km. Furthermore, the coupling ratio $C$ can be tuned to match any desired value. Thus, the key parameter in the optimization is the coupling ratio $C$. Several different optimization problems can be formulated, depending on the constraints posed on $C$. The most common optimization assumes that $C$ is the same for all taps. In this case, one easily finds that the optimal $C$ is given by [13]

$$C = 1/(N - 1)$$

where $N$ is the number of stations. From the above expression, one can then compute the max value of $N$ for given power margin $M$ and for given excess and attenuation loss per tap, $\gamma$. For example, for $M = 45$ dB and $\gamma = 0.2$ dB one finds $N_{MAX} = 24$ [12].

A more sophisticated optimization involves the individual adjustment of the ratio $C$ for each tap. If this procedure is followed, one finds that for $M = 45$ dB and $\gamma = 0.2$ dB up to $N_{MAX} = 50$ stations can be installed on a bus [12]. Even better results can be obtained if within each interface the coupling ratio's of transmit and receive taps are individually optimized. In this case, the maximum number of stations becomes $N_{MAX} = 61$ for the above stated parameters [12].

C. Power Calibration

For proper signal detection at the p-i-n diode, the received power level should be approximately constant, independent of the transmitting station. In other words, the dynamic range allowed by the receiver is very limited, for proper operation.

If all the stations on the bus transmitted at the maximum power $P_t$, clearly the above condition would not be met. In fact, the signal received from a distant station may be as much as $40$ dB lower than the signal received from a near station. Even worse, since in the token protocol the first few bits of the preamble may be the superposition of signals from several stations, the large disparity in received powers may seriously impair the effectiveness of the AGC (automatic gain control) mechanisms. There is therefore a need to calibrate the output power of the transmitters along the bus so that a receiver sees the same power, regardless of the station that is transmitting. This calibration can be carried out with a variable optical attenuator placed immediately after the laser.

With transmitter calibration in place, a given receiver will receive the same signal level regardless of the position of the transmitter on the bus. However, the signal level will vary from receiver to receiver. For implementation and maintenance purposes it is convenient that the received signal level be the same at all stations. This would permit the replacement or the exchange of receiver units without requiring a cumbersome adjustment of operating levels each time. Again, the simplest solution to this problem is to place an optical variable attenuator in front of the receiver.

The receive and transmit attenuators can be adjusted at network initialization. One of the problems with this static adjustment approach, however, is the fact that whenever the system configuration is changed (e.g., a new tap is inserted, or the cable extended, etc.) a readjustment of the attenuators is required. To overcome this problem, an adaptive adjustment procedure must be used.

A possible adaptive adjustment scheme for the U-Net protocol is shown in Fig. 17. It is based on closed loop control. The optical attenuators are implemented with parallel optical guides with coupling ratio variable from zero to one depending on the voltage applied. We assume that the function of coupling ratio versus voltage is known for each attenuator. We let $P_i$ be the incoming power from the bus and $P_o$ the outgoing power to the bus.

First, we must reduce the input power from $P_i$ to $P_o$ so that the detector works at the target operating point and provides the desired output level $S_{REF}$. This can be accomplished by comparing the actual output $S$ with the reference $S_{REF}$ and driving the attenuator with the difference signal $\Delta V$.

Next, we must regulate the output $P_o$ so that this output generates on the bus the same signal strength as any other upstream station. From Fig. 17 we note that the following relationship must be satisfied (neglecting excess losses):

$$P_o C = P_i (1 - C^2)/C.$$  

Or

$$P_o = P_i (1 - C^2)/C^2.$$  

$P_o$ can be computed from $P_i$ and $\Delta V$ since the coupling vs voltage function of the attenuator is known. Likewise, $P_i$ can be computed from $P_o$ and $\Delta V$. The measurements of $P_o$, $P_i$, $\Delta V$ and $\Delta V$ are fed to a comparator node.
which computes \( P_I \) and \( P_0 \) and generates a voltage \( \Delta V \) proportional to the value \( D \) given by

\[
D = P_0 - P_f (1 - C^2/C^2).
\]

It should be noted that traffic on the U-Net bus is bursty, consisting of trains of packets separated by silence intervals. Thus, signal level and power measurements are also bursty. Since close loop controls require continuous invals, these protocols are suitable for high bandwidth allocation and control. Work is now in progress on the incorporation of such features in the access protocols.

**VII. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS**

We have described in this paper three different token-based protocols designed for twin fiber bus networks. These protocols are suitable for high bandwidth x length products. They provide bounded delay at heavy load, thus permitting the integration of data and real time traffic (voice, video, etc.) on the same network. The logic needed at line speed is kept simple. In fact, the protocols are driven solely by the detection of activity on the channels. The completely distributed control and the passive nature of the interface guarantee robust, fail-safe operation.

The various versions of the protocols offer different performance characteristics. The choice of the protocol should depend ultimately on the application at hand.

One of the well-known drawbacks of fiber bus networks is the limitation on the number of stations that can be connected to the bus due to tap insertion loss. In this respect, twin bus architectures offer a slight advantage over "folded" bus architectures (such as Express-Net and D-Net) in that the latter need to support twice as many taps as the former on a single bus segment. Nevertheless, the maximum number of stations that can be accommodated on a twin bus is in the order of 20 to 30 in the best case. This severely limits the use of the fiber bus in applications with large number of devices. In order to overcome this problem, we are currently pursuing at UCLA the investigation of multilevel network architectures which permit to interconnect several busses, thus expanding the number of station supported, and yet maintain the high bandwidth and fault tolerance properties of the original structure.

Another key issue in local networks supporting integrated data, voice, and video traffic is the ability to guarantee bandwidth to real time (i.e., voice and video) users. Token-based protocols appear to be ideally suited for bandwidth allocation and control. Work is now in progress on the incorporation of such features in the access protocols.

**References**


**Mario Gerla** (M’75) received the graduate degree in engineering from the Politecnico di Milano, in 1966, and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively.

From 1973 to 1976 he was with Network Analysis Corporation, New York, NY, where he was involved in several computer network design projects for both government and industry. From 1976 to 1977 he was with Tran Telecommunications, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Computer Science Department at UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks.
Paulo Rodrigues received the M.Sc. degree from COPPE, Rio de Janeiro, Brazil, in 1977, and the Ph.D. degree from UCLA in 1984, all in computer science.

He has been a Consulting Computer Analyst and Professor at the Federal University of Rio de Janeiro (UFJR), Brazil, since 1975, where he participated actively in the development of microsystems and peripheral devices. In 1978 he was director of EBC, a Brazilian computer industry, and in 1979 he joined the Ph.D. program at UCLA where he worked as a Research Assistant in high speed fiber optics local area network research.

He is presently in the research division of the Computer Center (NCE) and UFJR. His current technical interests are in design and performance analysis of computer networks, local area networking supporting integrated services, interconnection of computer networks and distributed operating systems.

Cavour Yeh (S'56-M'63-SM'82) was born in Nanking, China, on August 11, 1936. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, in 1957, 1958, and 1962, respectively.

He is presently professor of electrical engineering at the University of California at Los Angeles (U.C.L.A.). He joined U.C.L.A. in 1967 after serving on the faculty of U.S.C. from 1962 to 1967. His current areas of research interest are optical and millimeter-wave guiding structures, gigabit-rate fiber optic local area network, and scattering of electromagnetic waves by penetrable irregularly shaped objects.

Dr. Yeh is a member of Eta Kappa Nu, Sigma Xi, and the Optical Society of America.