A New Hybrid MAC-layer Protocol for Real-Time Bus Networks

V.D. KAPSALIS, S.A. KOUBIAS and H.C. HARALABIDIS
Applied Electronics Laboratory, Department of Electrical Engineering,
School of Engineering, University of Patras, Patras 26500, Greece

Indexing terms: Local area network, Hybrid MAC Protocol, CSMA/CD, Token passing, Performance evaluation.

Abstract: A hybrid Group Implicit Token - CSMA/CD (GIT-CSMA/CD) access protocol for use in real-time bus networks is introduced. Network control is distributed over all nodes, thus eliminating the need for central switching. Operating features of the hybrid protocol are described together with its computer simulated performance evaluation. The results showing the delay-throughput characteristics are compared with CSMA/CD 1-persistent and token passing MAC-layer protocols. Optimum values for the number of stations consisting a group are obtained according to the requirements and the specific conditions of each application. The implementation of the proposed protocol is based on the capabilities and the features of several existing CSMA/CD controllers and requires minor additional hardware and software.

1 Introduction

Local Area Networks have been widely used in data processing applications either to meet the communication requirements of large organizations or to enable distributed access by a great number of terminals to centralized computing systems. Recently, much interest has been focused on applications of LANs to real-time environments [1-3]. LANs of this type impose stricter performance criteria than their office counterparts which are mainly due to the critical response times of several time-constrained packets transmitting in a real-time environment.

The research in real-time communications can be grouped into two categories: best effort communication (or soft real-time) and hard real-time communication. The primary performance objective of the first category is to maximize the percentage of packets successfully transmitted within a predetermined time constraint. Applications of this category include packetized voice/video communication, in which a certain amount of packet loss is usually tolerable and a packet which is not successfully delivered within a certain time limit is considered lost [2].
The protocols belonging to the second category aim at improving the throughput while ensuring all time-constrained packets to meet their deadlines (bounded packet delay). Classical applications of hard real-time networks are the industrial networks in which several packets, such as alarm and control signals, must reach their destination before a certain time.

This paper is concerned with the design of a new hybrid Group Implicit Token-CSMA/CD (GIT-CSMA/CD) access protocol for the MAC sublayer of the OSI Reference Model. The protocol aims at combining effectively the advantages of the CSMA/CD and token passing protocols (immediate access during idle periods, maximum throughput and bounded packet delay during peak load periods), avoiding their major drawbacks without much control at user access policy. The property of bounded packet delay makes it suitable for hard real-time networks. An objective of this work was the feasibility of implementation of the proposed scheme by using the capabilities and the features of currently available CSMA/CD network controllers.

2 Background and previous work

There have been numerous publications on the subject of investigating the performance of MAC protocols for real-time applications. A non-exhaustive list of these studies illustrating several approaches follows.

Most of the CSMA/CD-based networks [4] have been mainly aimed at commercial rather than in real-time applications, due to their non-deterministic nature. However, in the literature, several studies examined the feasibility of using CSMA/CD (or Ethernet) for real-time applications, either without any kind of modification [3], [5,6], or by introducing (implicit or explicit) priorities for the high priority packets [7-13]. A general conclusion of these studies is that, under certain assumptions, the requirements of several soft real-time applications are met by the CSMA/CD protocol.

Various schemes which are based on deterministic retransmission delays for the collided packets of a CSMA/CD protocol [14-18], result in an upper bounded delay for all the transmitted packets. However, this is achieved at the expense of inferior performance than CSMA/CD at low to moderate channel utilization in terms of delay-throughput.

On the other hand, token passing protocols such as the IEEE 802.4 Token Passing Bus [19] have been widely
accepted as the standard for industrial LANs [20-22]. Token passing proves more suitable for the industrial environment because at high loads it is both efficient and deterministic. However at low channel traffic, its performance is not as good as of contention protocols (e.g. CSMA/CD). This lack of efficiency becomes more serious as the number of nodes and the (bandwidth) x (distance) product are increased.

The complementary behaviour of the token passing and CSMA/CD access protocols has led to the suggestion that they be combined into a hybrid access protocol. Protocols which use hybrid access techniques have been proposed and evaluated by several authors both for unidirectional channels [1], [23-29] (e.g. fiber optic network) and for bidirectional bus networks [30-33]. The protocols described in [29,30] control the channel access by using reservation and information subchannels, while the protocol proposed in this paper is suitable for bidirectional single bus networks; therefore, a brief overview of a few representative hybrid protocols for bidirectional single bus follows.

In [31], a self-adaptive protocol (ATP-1) is analyzed which switches dynamically between two states, the random-access state and the fixed-access state, approaching S-ALOHA for small values of channel load, while it becomes a TDMA when the load is heavy. A feature of ATP-1 is that accurate synchronization must be maintained by all the stations; thus its implementation requires more complicated circuits than those of pure contention protocols.

In [32], the authors present an attractive hybrid protocol which combines the features of control token and CSMA/CD. Although, the protocol implements a token passing scheme, there is no waste of bandwidth, because token passing is implicit, i.e. the action is caused by the end of a transmission period, and not by the explicit transmission of a control packet. However, the procedures for initializing the logical ring, insertion and deletion of stations and handling of duplicate tokens are similar to that of the IEEE 802.4 standard, requiring several control packets which degrade the effective throughput. Moreover, the protocol doesn't guarantee an upper bound to the transmission delay of all the packets.

Another study [33], proposed a MAC-level protocol (Virtual Token) for Ethernet-type networks which is a variation of [15] and [32]. A number of backoff algorithms which guarantee bounded packet delay were tested and the result was that Virtual Token performs better than Token Bus in terms of mean delay-throughput; its performance is also comparable to CSMA/CD for low and moderate loads, being much better for heavy traffic conditions.
Section 3 describes the proposed hybrid MAC protocol and some critical times (for the implementation) are calculated. A simulation study and a comparison of its performance with a number of protocols is presented in section 4, where the results are also discussed. Finally section 5 gives some concluding remarks.

3  Hybrid Group Implicit Token-CSMA/CD (GIT-CSMA/CD) Protocol

The proposed protocol is a combination of group implicit token passing and CSMA/CD 1-persistent. It can be seen as consisting of the Virtual-Token [33], a deterministic contention resolution protocol [14] and the Token-Skipping protocol proposed in [34]. Token-Skipping is an access scheme for token bus networks, which allows the token to pass along certain dynamically determined chords of the logical ring, bypassing the idle stations. This results in a better performance for a wide range of moderate traffic loads (up to 75% of load for asymmetric traffic) compared to IEEE 802.4.

For the description of GIT-CSMA/CD protocol, the following notation is assumed:

- $N$ number of stations
- $m$ number of groups
- $G_i$ group identity ($i=1,2,...,m$)
- $S_j$ station address ($j=1,2,...,N$)
- $k$ number of stations in each group ($k=N/m$)
- $G_i^T$ identity of the group possessing the token
- $G_{G_i/S_j}$ identity of the group $G_a$ containing station $S_j$
- $T_g$ minimum interframe spacing between frames
- $T_t$ deferring period for stations possessing the token
- $T_e$ deferring period for stations without the token
- $T_{max}$ maximum time of no activity on the bus (at the deterministic contention resolution phase) after which the token is passed to the next group
- $T_s$ slot time for the deterministic resolution of collisions
\( \tau \)  
end-to-end propagation delay  

\( T_f^m \)  
maximum transmission delay for a packet  

\( P_s^m \)  
maximum length of a valid packet  

The diagram which presents the operations of the hybrid protocol is shown in Fig.1; an analytical description of these operations is given at the following subsections.  

3.1 Network Configuration  
The stations of the network \((1, 2, \ldots, N)\) are connected in a bus topology and are organized in groups \((G_1, G_2, \ldots, G_m)\) consisting a logical ring in cyclic order. Each group contains \(k\) stations. There is not any separation between active and inactive stations. Token passes along the groups under certain conditions (subsection 3.5) after the channel state changes from busy-to-idle (implicit token passing). These channel state changes are detected at the end of transmission periods (successful transmissions or collisions between stations without the token). In the case of a collision between stations having the token, a different algorithm is followed; an analytic description of this algorithm will be given below.  

3.2 Priority Mechanism  
Each time, only one of the groups possesses the token. The possession of the token by a group of stations, \(G_i^T\), means that this group is assigned the highest priority over all others. This priority (taking place in a cyclic order) is achieved by a modification of a transmission parameter, the interframe space (time between successive frames). Analytically, after the last bit of the frame currently in the communication channel, the stations with the token, continue to defer for the interval \(T_t (=T_g)\) before initiating a transmission (in token mode) irrespective of the value of the carrier-sense. This deferring period is different for the stations not having the token; they defer for a longer time interval \(T_c\) in order to have enough time for the detection of any possible activity by the stations of \(G_i^T\) and thus cause no interference to them. The relation between \(T_t\) and \(T_c\) depends on various parameters and will be calculated below. Thus, stations ready to transmit but without possessing the token, wait until the end of carrier and if none of the stations belonging to \(G_i^T\)
transmits, they will be able to gain access to the channel (Fig.1). Transmission in token mode, means that the transmitting station belongs to the group currently possessing the token \((G_i^T)\). Transmission in CSMA/CD mode may be initiated by any station which does not possess the token, provided that all the stations belonging to \(G_i^T\) are inactive. Note that a packet is transmitted in token mode only if its arrival has taken place before the station becomes the token possessor; a packet arrival after that instant is transmitted in CSMA/CD mode. This protocol's feature avoids collisions between stations belonging to \(G_i^T\) with stations not having the token and thus achieves a simple (and robust) collision resolution algorithm and a completely collision-free operation at very high traffic conditions.

### 3.3 Collision Resolution Algorithms

If a collision between the stations without the token is detected, all stations involved terminate their transmissions, send jam signals and schedule retransmissions after a random number of slots (backoff interval); these stations get in the backoff state. The number of slot times to delay before the \(n\)th retransmission attempt is chosen as a uniformly distributed random integer \(r\) in the range:

\[
0 \leq r < 2^{(n-bv)}
\]

where \(v\) is an integer that corresponds to the upper limit of the backoff interval \(2^v\) and \(b\) is the integer part of the \((n-1)/v\) ratio. This backoff algorithm guarantees a bounded distribution of backoff time. In the case that one station gets the token while it is still in the backoff state, it transmits its packet in token mode. It is virtually a truncation of backoff interval that takes place, as soon as the station gets the token; thus, an upper bounded delay is guaranteed for all the packets (Fig.1).

A different procedure is followed in the case where a collision between the stations of the group possessing the token is detected; a deterministic contention resolution takes place [14], allowing any station of the group to transmit its previously collided packet at its preassigned slot, without experiencing any interference. At any time, each of the stations belonging to \(G_i^T\), owns a distinct slot of duration at least equal to the maximum round-trip propagation delay \((T_s = 2\tau)\). Each station defers for a number of slots before initiating a retransmission. If a station belonging to \(G_i^T\)
detects no transmission on the channel (by the stations of its group that own a preceding slot) upon its scheduled transmission point, then it transmits. Otherwise, if any packet transmission is detected, the station stops counting (slots) and waits for the end of carrier, where it continues the above procedure (deferring for the rest of slots) until its own preassigned slot, where it transmits. During the deterministic resolution of a collision between the stations of $G_i^T$, no transmission is allowed to be initiated by any station not having the token. This is achieved by having all the other stations deferring for $T_c$ time between successive frames, while stations possessing the token defer for $T_t$ time. An example, showing a collision between $S_1$ and $S_5$ (among $k=5$ stations belonging to the group currently possessing the token, $G_i^T$) and the deterministic resolution algorithm that follows is illustrated in Fig.2.

3.4 High Traffic Conditions

The proposed protocol adapts dynamically to CSMA/CD at low channel traffic and to group token at moderate loads. However, at a heavy loaded channel, the maximum throughput is achieved by using a collision-free operation, in which each station transmits without experiencing any interference (individual token mode). GIT-CSMA/CD protocol uses an efficient algorithm (executed by all the stations in a distributed manner) in order to achieve a completely collision-free operation at high traffic conditions, while retaining the benefits of group token operation at moderate channel loads.

Therefore, each station has the capability of detecting any activity on the channel (bus topology) as well as to distinguish valid transmissions from collisions. Furthermore, collisions between stations belonging to $G_i^T$ can be distinguished from collisions between stations transmitting in CSMA/CD mode (without the token). This can be achieved easily by having each station counting the time elapsed from the end of a transmission and until a collision has occurred; if this time lies between $T_t$ and $(T_t + 2\tau)$, it is assumed that a collision between stations possessing the token has taken place. The term $2\tau$ is added in order to enable any station detecting the same channel activity, irrespective of its position. The time intervals between successive possessions of the token by a group are called operation cycles. Operation at individual token mode means that there is no grouping of stations; the particular group consists of one station only. Note that different groups may operate in different modes simultaneously (CSMA/CD, group token or individual token mode).
The transition from group token mode \((\text{Group Token flag}=1)\) to individual token mode \((\text{Group Token flag}=0)\) takes place each time a collision between stations of the group currently possessing the token, \(G_i^T\), occurs (Fig.1). At the first operation cycle, after this collision, \(G_i^T\) will contain the first station (among \(k\) stations of the group), at the next cycle, it will contain the second and so on. Thus, each successive cycle for this group, belongs to one station only (individual cycles), in a cyclic order \((1,2,\ldots,k,1,\ldots)\). The transition back to group token mode takes place, if only one (or none), among \(k\) stations belonging to the particular group, transmits in token mode at its individual cycle; this event (detected by all the stations of the group monitoring the channel) initiates a group mode of operation, at which all the \(k\) stations belonging to this group will be able to access the channel at the next cycle (group token cycle). Note that, for simplicity purposes, the transition from individual to group token mode is not shown in Fig.1.

3.5 Implementation of Implicit Token Passing Mechanism

Only one state variable is required to be tracked by each station for the implementation of the implicit token passing. This variable is \(t\); it is the number of channel state changes (from busy-to-idle) before the station possesses the token. All the stations decrement their values of \(t\) by 1 (modulo \(m\)), when certain conditions have fulfilled and after the channel state changes from busy-to-idle. The station with \(t=0\) possesses the token. The above mentioned conditions are the following:

\(i\) End of a valid transmission by (only) one station belonging to \(G_i^T\). In this case, only one station (among \(k\)) belonging to the group currently possessing the token is active; thus it initiates a transmission (in token mode) without experiencing any interference.

\(ii\) End of a valid transmission by any station not belonging to \(G_i^T\). This corresponds to the case in which none of the stations belonging to the group with the token has a packet waiting for transmission and a station without the token transmits a packet in CSMA/CD mode.

\(iii\) End of collision between stations not belonging to \(G_i^T\). None of the stations belonging to \(G_i^T\) is active and two or more stations not possessing the token try to transmit their packets. Obviously, a collision takes place; at the end of this collision the token is passed to the next group.

8
iv) Transmission of $k$ packets in token mode during the deterministic contention resolution between the stations belonging to $G_i^T$ or bus idle for a time interval greater than $T_{\text{max}}$ during this period. Analytically, at the time instant that a group gets the token, there are two or more stations belonging to this group, having packets for transmission; thus a collision takes place, followed by a deterministic collision resolution period. In this case, the token is passed to the next group after the end of the $k^{th}$ transmission in token mode, or (if less than $k$ stations had involved in the previous collision) after a certain time of no channel activity, $T_{\text{max}}$:

$$T_{\text{max}} = T_i + \left[ T_i(k - 2) \right] + 2\tau k_2 \tag{2}$$

### 3.6 Calculation of some Critical Times and the Maximum Throughput

As it has already mentioned above, stations not belonging to the group with the token, must defer (for $T_c$) any possible transmission, until the time that any activity by the stations having the token has finished. This activity may be a single transmission or a number of retransmissions (up to $k$) during the deterministic resolution of collisions. The stricter restriction is imposed by the contention resolution period, where all the collided stations belonging to $G_i^T$ have to retransmit their packets without any interference. Thus, the stations without the token must defer any possible transmission for a time interval, $T_c$, at least longer than $T_{\text{max}}$. This time is generally different for each LAN controller. For example, 80C152, [35] is able to detect any activity on the channel (and subsequently to defer any possible transmission), if it takes place during the first half of its deferring interval; thus the relation between $T_c$ and $T_{\text{max}}$ for this controller becomes (see Fig.2):

$$T_c > 2T_{\text{max}} \tag{3}$$

One important property of the GIT-CSMA/CD protocol is that bounded transmission delay is guaranteed for all the packets. It is due to the bounded distribution of the backoff time in conjunction with the token mode operation. If the cost (time delay) of switching the token by the backoff algorithm is never greater than the corresponding cost implied by the token mode operation [33], then the maximum delay, $T_{f_{\text{m}}}$, from the moment that a packet gets on the top of the queue buffer until its successful transmission is given by:

$$T_{f_{\text{m}}} = \left\{ (m - 1)\left[ (k + 1)T_r + kP_n + P_s \right] \right\} + \left\{ m(k - 1) + 1 \left( P_n + T_r \right) \right\} \tau_2 \tag{4}$$
This delay stands for the worst case. This is the case when a packet gets on the top of a station's queue just after this station has lost the individual token. Also, all the rest groups operate in group token mode and a collision between the stations of each group takes place, followed by $k$ deterministic transmissions (first term enclosed in braces). The second term enclosed in braces includes the delays imposed by the individual mode transmissions of the network stations preceding the particular station and the transmission delay of the station's packet. If there is no grouping of stations ($k=1$), the maximum delay is equal to:

$$T_{ij}^m = N(P_i^m + T_i)k = 1$$

The maximum achievable throughput ($S_{max}$) by the GIT-CSMA/CD is given by:

$$S_{max} = \frac{P_i^m}{P_i^m + (T_i + \tau)}$$

### 3.7 System Initialization and Node Insertion-Deletion

The logical ring consists of all the stations irrespective of their current state (active or inactive). Thus, no procedures are needed for the initialization of the logical ring, as well as for any deletion of stations. The only procedure that must be followed by a station wishing to enter the ring, is to be informed about the group currently possessing the token. It is achieved, by having the station monitoring the channel, until a transmission is initiated at a time instant lying between $T_i$ and $T_i + 2\tau$ after the end of the previous transmission. This event can trigger a promiscuous mode [35-37] for the new station's controller (receiving packets irrespective of their destination addresses). The group currently possessing the token will be determined by the source address of the received packet (each station knows the addresses of the stations consisting a group). The new station will calculate the value of busy-to-idle channel state changes before the new station is in possession of the token, $t$ as:

$$t = \text{mod}_n(G_i^T - G_n\{S_j\} - 1)$$

where $G_i^T$ is the group identity of the station which transmitted and $G_n\{S_j\}$ is the identity of the new station's group. After the determination of $t$, the network controller will return in normal mode (receiving a packet only if the packet's destination address coincides with its individual address). In the case that, a station is the first which enters the network.
(a predetermined number of busy-to-idle channel state changes has taken place without any transmission in token mode) it gets the token by setting $t=0$. Note that each station is able to transmit in CSMA/CD mode, even if it has not determined the group currently possessing the token yet; thus achieving immediate access to the channel and retaining in this way the advantages of the pure contention protocols.

3.8 Error Handling Procedures

The problem of lost tokens is avoided by the implicit token passing mechanism. However, there is the possibility that "multiple" tokens may exist simultaneously. That is, two or more stations, belonging to different groups, have the same $t$; thus they become token possessors at the same time instant. This invalid situation may be detected by any station possessing the token, if it encounters a collision during its transmission in the deterministic resolution of a previous collision in group token mode, or during its individual token mode transmission. In this case, the stations involved in a collision perform the initialization procedure again (see Fig.1).

3.9 Calculation of the Maximum Allowable Propagation Delay

In a CSMA/CD network the number of busy-to-idle channel state changes that a station monitors, may differ from those of another remote station, due to the finite propagation delay of the channel. Fig.3 is a space-time diagram illustrating a typical collision between 3 stations, referred to as $A,B$ and $C$. Transmission for station $A$ starts at $t_{sa}$ and for $B$ at $t_{sb}$. Since they are very close, the detection of the collision takes place very quickly at time instances $t_{ca}$ and $t_{cb}$ for station $A$ and $B$ respectively. Due to the fact that jam signals are not allowed to be sent if the preamble ($t_p$) has not finished, stations $A$ and $B$ transmit their jam signals at $t_{ja}$ and $t_{jb}$ respectively ($t_{ja} - t_{sa} = t_{jb} - t_{sb} = t_p$). Station $C$ detects the collision at time $t_{c}$ and starts transmitting its jam signal at $t'_{c}$. As is clearly shown in Fig.3, stations $B$ and $C$ will detect one busy-to-idle transition of channel state at time $t'_{b}$ and $t'_{c}$ respectively, while station $A$ will detect two channel state changes (the first at $t'_{a}$ and the second at $t'^2_{a}$). This invalid situation may be avoided if the following inequality stands:

$$t_p + t_{j} \geq 2\tau$$
where $t_p$ is the preamble duration, $t_j$ is the jam duration and $\tau$ is the end-to-end propagation delay.

4 Simulation Results and Discussion

In this section, the performance of the proposed hybrid protocol is evaluated and compared with the CSMA/CD 1-persistent, the token passing and the Virtual Token [33] protocol, using simulation. Protocols with token passing mechanisms have been incorporated into a number of network standards for real-time applications which require bounded packet delay (hard real-time) [20-22]. Thus, the results of a comparison between GIT-CSMA/CD and the token passing protocol will highlight the benefits obtained by using the former protocol for the same range of applications. Since our aim was to investigate the behaviour of the new protocol in an implementable network, the assumptions of the simulated model were based on the capabilities and the features of an existing network controller (80C152, [35]):

- Network data rate: 2 Mbps (1 bit = 500 ns).
- End-to-end propagation delay: $\tau =$ 1 and 10 bits.
- Minimum interframe space: 50 bits.
- Jam size ($t_j$): 16 bits.
- Packet size (including overhead): 800 bits.
- Simulation was carried out for $N =$ 12 and 48 stations.
- Group size ($k$): 1, 2, 3, 4 and 6.
- Poisson arrival of packets at each station is assumed.
- The upper limit of the backoff algorithm of GIT-CSMA/CD: $v=3$.
- The slot size for GIT-CSMA/CD must be greater than $T_c + 2\tau$ (imposed by the features of 80C152). $T_c$ is calculated according to equations 2 and 3.
- Slot size for the exponential backoff algorithm of CSMA/CD: 40 bits.
- The length of the token was 48 bits (equal to the minimum packet sent by the 80C152).
- Two types of traffic distributions were simulated: symmetric and asymmetric traffic.
The mean delay-throughput performance of the hybrid protocol for \( N=12 \) stations, \( P_s=800 \) bits and \( \tau=1 \) bit \((a=\tau P_s=1.25\times10^{-3})\) is compared to its component protocols in Fig.4a, where it is assumed that no grouping exists \((k=1)\) and the load is symmetric. Obviously, GIT-CSMA/CD operates similar to CSMA/CD (Fig.4a) at low traffic and much better at loads higher than about 0.65; also, it exhibits an improved performance over token passing (Fig.4a) for the whole range of traffic. Fig.4b illustrates the improvement achieved by the grouping of stations \((k=2,3 \text{ and } 6)\) compared to the hybrid protocol without grouping \((k=1)\). This improvement stands for a wide range of moderate (about 0.3) to high (about 0.9) channel traffic. At higher loads than 0.9, grouping of stations has no effect and all the versions of GIT-CSMA/CD \((k=1,2,3,...)\) achieve a quite similar performance. Fig.5a compares the performance of the three protocols for \( N=48 \) and \( \tau=10 \) \((a=12.5\times10^{-3})\). Note that the hybrid protocol without grouping \((k=1)\) suffers a degradation at the medium range of throughput compared to the case illustrated in Fig.4a. However, a comparison between Figs.4b and 5b illustrates that the improvement achieved by grouping the stations increases as \( N \) and \( \tau \) get higher. A general conclusion (Figs.4b,5b) is that under the assumption of symmetric load the best performance in terms of mean delay-throughput is achieved when \( k=2 \). It is due to the fact that in the case of a collision between stations belonging to \( G_i^T \), the number of stations involved may be less than \( k \), if \( k>2 \); thus, the deterministic resolution phase will contain unused slots which are responsible for a degradation of the performance at moderate loads. Furthermore, the algorithm controlling the transition from individual token mode to group token mode (subsection 3.4) achieves the best performance for \( k=2 \).

Fig.6a illustrates the mean delay-throughput performance of GIT-CSMA/CD, for \( N=12 \), \( P_s=800 \) bits and \( \tau=10 \) bits \((a=12.5\times10^{-3})\), under the assumption of asymmetric traffic distribution; two of the stations (not belonging to the same group) were responsible for the 60% of the load offered to the network. The results for a different kind of asymmetry (assuming the same \( N, P_s \) and \( \tau \)) are shown in Fig.6b. In this case, two stations were responsible for the 80% of the offered load. Simulations for both cases were carried out for \( k=1,2,3 \text{ and } 4 \). The best performance for the former case is achieved by using \( k=2 \); for the later case the optimum \( k=3 \). Thus, the optimum \( k \) (under asymmetric traffic) depends on the percentage load each station offers to the network. Another conclusion (from the results illustrated in Figs.4,5, and 6), is that grouping of stations achieves a more significant improvement of performance under the asymmetric traffic distribution than under the symmetric one, compared to the case where \( k=1 \); this improvement stands for all the loads
higher than 0.4. Note that the maximum throughput under symmetric traffic distribution is independent of $k$ (Figs.4 and 5), while grouping of stations ($k>1$) achieves a higher throughput than in the case of $k=1$, under asymmetric traffic (Fig.6).

A criterion of great interest for a hard real-time protocol is its performance in terms of maximum packet delay. Fig.7 illustrates a comparison of GIT-CSMA/CD ($k=1$ and 2), with token passing and Virtual Token (a hybrid protocol with bounded packet delay, [33]), in terms of maximum packet delay versus number of stations (the parameters have been taken from [33]). Obviously, GIT-CSMA/CD with $k=1$ outperforms all the other protocols and achieves the lowest packet delay. As long as the maximum packet delay is the primary performance criterion of an application, the GIT-CSMA/CD with $k=1$ is the best choice. In the case that the time constraints imposed by an application are also met by the GIT-CSMA/CD with $k=2$, the most appropriate choice is the GIT-CSMA/CD with $k=2$ because of its superior performance in terms of mean packet delay.

Finally, Fig.8 shows the number of packets as a function of the collisions experienced before transmission (for $N=12$), taking as a parameter the throughput ($S=0.1, 0.3,...,0.9$). Obviously, none of the packets experienced more than 9 collisions (for the whole range of throughput values) which is less than the upper theoretical limit of collisions guaranteed by the hybrid protocol ($N$).

5 Conclusion

A hybrid MAC protocol, named GIT-CSMA/CD, has been described along with simulation results. The proposed protocol guarantees bounded delay for all the transmitted packets. This property makes it suitable for hard real-time applications. The concept of grouping the stations has been introduced and according to the specific application requirements, optimum values for the number of stations consisting a group have been obtained. The performance of the hybrid protocol in terms of mean delay-throughput is comparable to the CSMA/CD at low and moderate loads, being much better at heavy traffic conditions. Also, our protocol performs better than token passing for the whole range of offered loads. Under the assumption of symmetric traffic, the best performance in terms of mean delay-throughput is achieved if each group contains two stations ($k=2$). In terms of maximum packet delay, GIT-CSMA/CD with $k=1$
achieves the best performance compared to token passing, GIT-CSMA/CD \((k=2)\) and Virtual Token (a high performance hybrid protocol with bounded delay). In the case of asymmetric traffic, the optimum group size \((k)\) depends on the kind of the particular asymmetry.

Token passing and initialization of the network is achieved without the transmission of control messages. Insertion and deletion of any station is not necessary; thus there is no waste of bandwidth and any station is capable to achieve immediate access on the channel. All the algorithms required are executed in a distributed manner, thus achieving robustness and simplicity. GIT-CSMA/CD can be easily implemented based on the capabilities of several currently available CSMA/CD LAN controllers in conjunction with additional hardware and software.

6 References


30 KOUBIAS, S.A., and PAPADOPOULOS, G.D.: "Dynamic behaviour of a new adaptive multiple-access protocol and comparison with the


37 NATIONAL SEMICONDUCTOR, Local Area Network Databook, 1992.
7 List of Captions

Fig.1 State diagram of the hybrid GIT-CSMA/CD protocol

Fig.2 Collision resolution between stations possessing the token

Fig.3 Space-time diagram

Fig.4 Mean Delay-throughput characteristics under symmetric load, N=12, τ=1 bit, P_s=800 bits (a=1.25*10^{-3})
   a) CSMA/CD, token bus (token length=48 bits) and GIT-CSMA/CD (k=1) and
   b) GIT-CSMA/CD (k=1,2,3,6)

Fig.5 Mean Delay-throughput characteristics under symmetric load, N=48, τ=10 bits, P_s=800 bits (a=12.5*10^{-3})
   a) CSMA/CD, token bus (token length=48 bits) and GIT-CSMA/CD (k=1) and
   b) GIT-CSMA/CD (k=1,2,3,4).

Fig.6 Mean Delay-throughput characteristics of the GIT-CSMA/CD under asymmetric load, N=12, τ=10 bits, P_s=800 bits
   a) two stations offer the 60% of the load and
   b) two stations offer the 80% of the load.

Fig.7 Maximum Delay characteristics of the GIT-CSMA/CD (k=1,2), token bus and Virtual Token.

Fig.8 Packet throughput characteristics
Fig. 1  V.D. KAPSALIS et al
Fig. 2  V.D. KAPSAISIS et al
Fig. 3 V.D. KAPSALIS et al
Fig. 4. V.D. KAPSALIS et al
Fig. 5 V.D. KAPSA LIS et al

Fig. 6 V.D. KAPSA LIS et al

[Graph showing the relationship between Maximum Delay and Number of Stations with different lines representing different protocols: GIT-CSMA/CD (K=1), TOKEN BUS, GIT-CSMA/CD (K=2), and VIRTUAL TOKEN (VT0).]
Fig. 7  V.D. KAPSALIS et al
Fig. 8  V.D. KAPSALIS et al