Ethernet for Short Packets

(As appeared in International Journal of Modelling and Simulation.)

Paweł Gburzyński*

*Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G2H1

Ziao You Zhou**

**The School of Computer Science
Acadia University
Wolfville, Nova Scotia, Canada BOP 1XO

Abstract: This paper proposes a new Ethernet topology, together with a communication protocol suitable for short-packet environments. The performance of the proposed scheme has been investigated by modeling the network and the protocol in LANSF. Empirical results show that our solution improves the performance of CSMA/CD in situations when packets are substantially shorter than the round-trip propagation delay of the bus.

Key Words: Local Area Networks, CSMA/CD protocols, Ethernet, Network performance.

1. Introduction

Commercial Ethernet [9, 11] is a bus-type multi-access network. The Ethernet protocol is based on a technique called Carrier Sense Multiple Access with Collision Detection (CSMA/CD). According to this protocol, a station willing to transmit a packet senses the channel before it decides to start a transfer. In addition, the station is able to recognize the collision of its packet by monitoring the carrier during transmission and detecting activities originated by other stations. The commercial success of Ethernet is due to a number of its advantages, such as low cost, simplicity, flexibility and reliability [10]. In terms of average performance measures, the Ethernet protocol is unbeatable: there is practically no synchronization overhead. Unfortunately, Ethernet suffers from the following two major drawbacks:

- As the workload increases, the maximum observed packet delay grows indefinitely. Thus, the network may not be applicable in a real-time environment.

- The Ethernet protocol imposes some boundary on the minimum length of a packet. This boundary grows with the increasing length of the bus and/or the transmission rate.

The first of the two problems mentioned above has been tackled by many authors. Several variations of CSMA/CD have been proposed [1, 2, 4, 5, 8] that, without giving up the good properties of Ethernet for light and moderate loads, improve its behavior for heavy traffic conditions. The second problem, which becomes painful for high transmission rates and short-packet applications, is addressed in this paper.

1.1 The Problem Statement

The Ethernet protocol only works under the assumption that if a packet is ever to collide, the collision must be detected while the packet is still being transmitted. In other words, no packet can be shorter than $2t_a$, where t_a is the propagation length of the bus expressed in bits.[8] For the commercial network, this length is 576 bits including the packet frame information, which corresponds to the maximum bus length of 288 bits.[9] Consequently, any packet whose total length is less than 576 bits has to be "inflated" to that minimum size before the packet can be handled by the protocol. As a result, in a short-packet environment, the performance of an Ethernet-type network depends heavily on two factors: the length of the channel and the actual size of the (uninflated) packets. If the packet size is small and the cable is long, the network performance will be poor.

The short-packet environment is characterized by applications in which the packet size is small, say a few bytes of data. Electronic mail, remote sensor sample data, character-oriented traffic between computers and unintelligent terminals, acknowledgement and status packets are just a few examples of such applications.

The problem of the minimum packet length for collision protocols becomes more serious when the network is based on a fast communication

[8] E.g., for 10Mb/s Ethernet, 1 bit=10^{-7} seconds$\approx$25m.

[9] It is assumed that the maximum length of commercial Ethernet is 256 bits (the minimum packet length includes a safety margin).

medium, such as fiber optic. Even for a relatively slow (by modern standards) network such as commercial Ethernet, the problem of poor performance in short-packet applications is serious. It is commonly felt that collision protocols are generally not applicable for high transmission rates. In our paper we demonstrate that it need not be the case.

1.2 Previous Work

The problem of short packets has been investigated by many authors—not only in the context of collision protocols (e.g., [6]). To improve the throughput of Ethernet under heavy traffic conditions, Marson and Roffinella [7] proposed a multichannel CSMA/CD protocol in which the network is divided into M subchannels. Unfortunately, the protocol is rather costly to implement: M separate receivers are required for each station and the status of all the subchannels must be sensed before every packet transmission.

Wong and Yum [12] proposed a scheme aimed at reducing the cost of the Marson's and Roffinella's solution. Essentially, their approach is based on a multichannel network with local and global transceiving media and a group of local transceiving channels. Each station uses its local channel to communicate with other stations on the same local channel. If a packet is to be sent to a station connected to another local channel, it is transmitted via a global channel, using the so-called Contention-based Look-Ahead Reservation protocol (CLAR). Since CLAR involves expensive synchronization mechanism, Wong and Yum's approach suffers from losing many of the attractive features of Ethernet. Moreover, performance degradation in short-packet environments still exists, especially when many stations want to communicate with stations on other local channels.

2. The Proposed Protocol

One *ad hoc* solution to the short-packet problem in Ethernet, dubbed the *naive* solution in the sequel, is as follows. Assume that each station knows its propagation distance to every other station in the network. This information can be kept in the station's address table where it can be naturally associated with the list of destinations. Assume that station s_1 sends a packet p addressed to station s_2. If the length of p (denoted by $S(p)$) is less then $2dis(s_1, s_2)$ (where $dis(s_1, s_2)$ denotes the propagation distance between s_1 and s_2), the protocol operates exactly as in the commercial network. Otherwise, having completed the packet transmission, s_1 monitors the bus for the amount of time equal to $2dis(s_1, s_2) - S(p)$. If no intruding activity is detected during that time,[10] the transmitter assumes that the packet has made it to the destination. Otherwise, the packet is retransmitted. With this protocol, it is possible that the receiver may successfully receive a packet while the transmitter "thinks" that the packet has been destroyed. In other words, the receiver may get the same packet more than once. This problem can be solved by labeling packets with serial numbers. Such labeling is typically performed by higher protocol layers.

The performance study of the naive approach (in a number of slightly different flavors) indicates that this solution is not very useful in dealing with short packets (see Table 1 and Figure 4).

2.1 The New Ethernet Topology for Short-Packet Environments

The main cause of the failure of the above-mentioned scheme is the inherent gap between the cable length and the packet size. In a bus network based on a uniform, single, broadcast-type medium, the whole bus is "flooded" by each packet transmission. This implies that for a given amount of information to be transmitted, it is generally better to organize it into a sequence of bigger packets, as this way channel utilization is improved. On the other hand, by using longer packets we increase the average message delay for the whole network. It is therefore commonly accepted that an "ideal" packet size in Ethernet is $c2t_a$, where $c \geq 1$ is a not-too-big constant which can be chosen heuristically.

Based on this assumption, we propose the following network topology (see Figure 1). The network consists of a global transceiving medium and a group of local transceiving channels. Each local channel is equipped with two buffers: one buffer for the leftmost station and the other for the rightmost station connected to the channel. The size of all these buffers is the same and proportional to $2t_a$.

The objective of the proposed topology is to offer the following advantages over standard Ethernet:

1. Higher throughput: local channels can be used for simultaneous packet transfers. Moreover, as an extra path is provided, the traffic in the network is split into two streams (i.e., a packet can be transmitted either

[10] In a realistic implementation this time should include a short tolerance overhead.

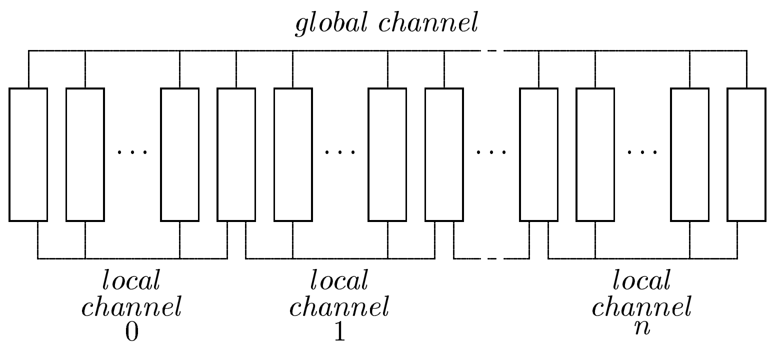

Figure 1: The proposed network topology.

via a sequence of local channels or via the global channel). Therefore, contention in both the global channel and the local channels is reduced.

2. Flexibility and expandability: stations and local channels can be added or deleted easily, and the global channel can be extended when network expansion is called for.

3. Suitability for short-packet environments: the local channels lend themselves conveniently to short-packet communication. Furthermore, with the appropriate organization of the protocol (discussed later), the performance of the network can be made independent of the network length.

4. Cost-effectiveness: smaller bandwidth media can be chosen for local channels in order to reduce the overall cost. On the other hand, a fast medium can be used for the global channel to increase throughput.

2.2 Basic Concepts of the New Protocol

Below we outline a few concepts that constitute the foundation of the protocol for the network topology introduced above. We assume that stations in the network are arranged in an increasing order of their addresses from left to right. Furthermore, every station s is assumed to have a complete knowledge of its local channel geometry, including the propagation distance

$dis(s, s_i)$ to every other station s_i connected to the same local channel. Note that a bridge station, which is connected to two local channels, must know the geometry of both channels. For any station s_j on a different local channel, s must know the number of local channels separating s and s_j, as well as the propagation distance $dist(s, s_j)$ on the global channel. We assume that the propagation length of the global channel (denoted by L_g, $L_g = t_a$) is known by all stations. Moreover, for the sake of simplicity, we assume that all local channels are of the same (or almost the same) length denoted by L_l. The protocol can be easily modified to account for different lengths of the local channels.

$Type$	$T(p)$	$R(p)$	$S(p)$	$Payload$	$Error$ $check$ $bits$

Figure 2: The packet format.

Two station addresses are associated with each packet p, namely, the transmitter $T(p)$ and the receiver $R(p)$. Assume that $S(p)$ denotes the size of p in bits. Then, according to Figure 2, a complete packet p is defined by 6 components: $T(p)$, $R(p)$, $S(p)$, type, the payload part and the error check bits. In addition, a set of packets, say $p_1, \ldots, p_n$, can be grouped together to form a *batch* b_n. The structure of a batch is depicted in Figure 3.

$Type$	n	$R(p_1)\ \ldots\ R(p_n)$	$Packet$ p_1 $\ldots$ $Packet$ p_n	$Error$ $check$ $bits$

Figure 3: The batch format.

Assume that a packet p arrives at station s. The *remaining local distance* of p from its destination, denoted by $rdis(p, s)$, is defined as the number of local channels separating s and $R(p)$—the packet's destination. For a batch b_n being sent to the left (right), the *remaining local distance* $rdis(b_n, s)$ is determined by the *remaining local distance* of the packet with the lowest (highest) destination address.

We define two cost functions of a packet (or batch) p. These functions will be evaluated by a station s deciding whether p should be sent via the global or the local channel. By $lcost(p, s)$ we denote the cost of sending the packet via local channels and by $gcost(p, s)$ we denote the cost of sending p via the global channel. We postulate the following properties of the cost functions:

- Assume that $T(p_1) = T(p_2)$, $R(p_1) = R(p_2)$, and $S(p_1) > S(p_2)$. Then:
$$\begin{array}{ll} lcost(p_1, s) > lcost(p_2, s) & \text{if } S(p_2) > C_l 2L_l \\ lcost(p_1, s) \leq lcost(p_2, s) & \text{if } S(p_1) < C_l 2L_l \end{array}$$
If a packet is shorter than the ideal size for a local channel, then the longer the packet the lower the cost of its transfer. Otherwise, the cost of sending the packet over the local channels increases with the packet length. In such case, the global channel may become a good choice for the transmission medium.

- Assume that $S(p_1) = S(p_2)$ and $rdis(p_1, s) > rdis(p_2, s)$. Then $lcost(p_1, s) > lcost(p_2, s)$. In simple words, the cost of sending a packet via local channels should increase with the number of local channels to be traveled by the packet. This seems natural as the overhead for buffering and switching increases with the number of local channels to be traveled.

- The first of the above properties is also postulated for the global channel cost function $gcost$. However, $gcost$ should be independent of the distance traveled by the packet as each packet inserted into the global channel travels across the entire network.

Let N_l denote the number of local channels. It is not difficult to see that the following cost functions fulfill our postulates:

$$lcost(p, s) = \begin{cases} \frac{K_l rdis(p,s) S(p)}{2C_l N_l L_l} & \text{if } S(p) \geq 2C_l L_l \\ \frac{2K_l C_l L_l rdis(p,s)}{N_l S(p)} & \text{otherwise} \end{cases}$$

$$gcost(p) = \begin{cases} \frac{K_g S(p)}{2C_g L_g} & \text{if } S(p) \geq 2C_g L_g \\ \frac{2K_g C_g L_g}{S(p)} & \text{otherwise} \end{cases}$$

Constants K_l and K_g are network parameters which should be tuned properly to optimize network performance. The role of these two constants will be discussed later.

The proposed cost functions are relatively simple and they can be used by the protocol to compare the cost of packets to be transmitted. Moreover, they are flexible and adaptable: the performance of the network can be improved by adjusting the system parameters: K_l, K_g, C_l, and C_g.

2.3 The Traffic Control Strategy

Each station uses its local channel to communicate with other stations on the same local channel. If a station wants to communicate with a station on another local channel, it has two options: to send the packet via the global channel or to use the local channel instead. The decision can be made by comparing the cost functions *lcost* and *gcost* introduced in the previous section. With these prerequisites, the traffic control strategy for a station s can be described as follows:

> For any packet (or batch) p, if $lcost(p, s) > gcost(p)$ then transmit the packet (or the batch) via the global channel; otherwise, transmit the packet (or the batch) via the local channel.

2.4 Channel Access Protocols

The proposed protocol consists of two parts that can be viewed as separate protocols for accessing the two types of channels. We discuss the local channels first.

2.4.1 The Protocol for Accessing Local Channels

As we have already mentioned, there are two buffers associated with each local channel. These buffers are used for storing packets that have to be relayed. The protocol for accessing a local channel operates as follows:

Transmitter

> If the station is neither the rightmost station nor the leftmost station on the channel, then the packet (or the batch) is sent according to the CSMA/CD protocol, using the control strategy

described above. If the station is the rightmost (or the leftmost) station on its local channel, then, besides transmitting its own packets, it must relay packets from other channels. In all cases, the station will do it by first grouping the packets in the buffer (if the buffer is not empty) together with its own packet (if there is one) to form a batch, then transmitting the batch, using the control strategy described in the previous section. If the packet is shorter than the round-trip propagation delay of the local channel, the naive strategy is used.

Receiver

The receiver is invoked upon the arrival of a new packet. First, the receiver checks the packet type. If the packet is a standard packet and it is addressed to the station, the packet is received and its processing is completed. If the packet is a batch, the receiver checks whether there are packets in the batch addressed to it. If so, these packets are received. For the leftmost (rightmost) station on a local channel, the receiver has the additional responsibility to buffer all packets whose destination addresses are smaller (larger) than its own address. If the buffer overflows, a jamming signal is inserted into the local channel to force the collision of the incoming packet.

2.4.2 The Protocol for Accessing the Global Channel

Since each station is equipped with one transmitter, the protocol for the transmitter is the same as described in the previous section. Thus, we only discuss here the receiver part of the protocol.

Receiver

The receiver is invoked upon the arrival of a new packet. First, the receiver checks the packet type. If the packet is a standard packet and it is addressed to the station, the packet will be received. If the packet is a batch, then the receiver checks whether there are packets in the batch addressed to it. If so, these packets are received. All other packets are ignored.

3. Performance

Due to the batching phenomenon, the queuing analysis of the proposed scheme is somewhat involved. The performance of our solution has been investigated by implementing the networks and the protocols in LANSF [3]. LANSF is a network-modeling package[11] oriented towards investigating low-level communication phenomena at the medium access control (MAC) level.

3.1 Simulation Models

The LANSF models of the networks and protocols discussed in this paper were very accurate and reflected practically all relevant aspects of realistic implementations. In particular, such phenomena as distance-dependent signal propagation time between two points of a channel, limited accuracy of distributed clocks, collision jams, etc., were all carefully accounted for. To illustrate the detail level of the model, let us describe briefly the collision-detection and processing mechanism.

Assume that station s starts transmitting at time t_0 and another station s_1 starts another transfer attempt in the same channel at time t_1. The collision is perceived by s at time $t_1 + dist(s, s_1)$ where $dist(s, s_1)$ is the propagation distance between s and s_1. At that moment, s aborts its transmission and sends a jamming signal.[12] Having completed the jamming signal, s waits for the amount of time determined by the *binary exponential backoff* algorithm and tries again. Similarly, s_1 notices the interfering transfer of s at time $t_0 + dist(s, s_1)$ and performs similar actions.

The assumed clock tolerance was 0.01%, which corresponds to the clock tolerance in the commercial network.

To reduce the impact of secondary factors on the observed performance of the investigated networks and protocols, we ignored the packet header, i.e., we assumed that the entire frame carries useful information that counts to the effective throughput.[13] The results can easily be scaled to any non-zero header length. Similarly, we reduced (but not ignored) the length of the

[11] The package is available via *anonymous ftp* from *menaik.cs.ualberta.ca*.

[12] The role of the jamming signal in the commercial network is to enforce the so-called *collision consensus*.

[13] Note that the packet header in the commercial network is rather long. It is reasonable to assume that it would be much shorter if the network were suitable for short-packet applications.

inter-packet space from 96 to eight bits. Collision jams were also shortened from 24 to eight bits.

The simulation experiments reported in this paper were performed for two network configurations denoted A and B. Experiments carried out for other configurations were consistent with the ones discussed here. Network A consisted of 51 stations equally spaced along the 1000-bit global channel. The number of local channels (N_l) was 10 and there were six stations attached to every local channel. The propagation distance between a pair of neighboring stations was 20 bits and the propagation length of a local channel (L_l) was 100 bits. In network B, the number of local channels was 40 and four stations were connected to each local channel. The distance between two neighboring stations was 10 bits; the propagation length of the global channel (L_g) was thus 1200 bits. The propagation length of a local channel (L_l) was 30 bits.

The value of $2t_a$ used by the *binary exponential backoff* algorithm was determined according to the channel in which the collision occurred. For example, for a collision occurring in a local channel of network A, the value of t_a assumed by the backoff algorithm was 208 bits. The eight bits were added as a "safety margin."

The traffic pattern was uniform, with exponentially distributed packet inter-arrival time and exponentially distributed packet length. A single point of a performance curve was determined as an average from four independent experiments. Each experiment started with a "warm-up" phase, during which network behavior was monitored for equilibrium, followed by the measurement phase, during which 100,000 packets were successfully received at their proper destinations.

3.2 Parameters of Cost Functions

Parameters of the cost functions *lcost* and *gcost* are determined by four constants: K_l, K_g, C_l, and C_g. C_l and C_g determine the threshold of suitability of a given packet to be transmitted on the corresponding channel. Assume that the global channel is the one in question. If the packet is longer than $2L_g$, it can be safely transmitted on the global channel in the same way as in standard Ethernet. Otherwise, the transmission is tricky and involves awaiting a possible "delayed" and uncertain interference. Thus it is clear that C_g should be set to $1 + \varepsilon$ where ε stands for a small safety margin. In our experiments both C_l and C_g were set to 1.001 and our attempts to use other values did not produce any interesting results.

The other two constants (K_l and K_g) can be combined into one. Note that the two cost functions will produce the same decisions if both K_l and K_g are multiplied by the same factor.

To get some insight into the impact of the ratio $r = K_g/K_l$ on the network performance, let us consider a packet (or batch) p which is transmitted from one end of the network towards the other end. This packet (or batch) must travel the entire network, irrespective of how it is transmitted. Let us calculate the threshold length of such a packet, for which the two cost functions return the same value. A packet with such length is equally suitable to be transmitted via the global channel and via local channels. From the definitions of the cost functions we have:

$$\frac{K_l S(p)}{2C_l L_l} = \frac{2K_g C_g L_g}{S(p)}$$

and consequently:

$$S(p) = 2\sqrt{r}\sqrt{C_l C_g L_l L_g}$$

The second factor in the above formula can be viewed as the geometric average of the channel lengths. The ratio r determines how the cost functions are biased in favor of a particular channel type.

Let us consider network A as an example and assume that r is 1. The threshold packet length is then about 640 bits. Note that this value is much too low for an easy transmission via the global channel. As a matter of fact, it is more than three times too low. Our experiments indicate that it is generally better to transmit such a packet via local channels. Thus, r should be bigger than 1.

3.3 Results

The performance of networks A and B was compared to the performance of Ethernet and the naive solution implemented on networks with the same numbers of stations and propagation lengths. In Ethernet it was assumed that short packets must always be inflated to $2t_a$.

It was consistently observed that the value of r maximizing the throughput achieved by our solution was close to 7. In particular, this value produced the best results for networks A and B. The maximum throughput achievable by networks A and B (with $r = 7$) for different mean packet length is listed in Table 1. For comparison, the table also includes the maximum throughput achieved by standard Ethernet and by the naive solution.

These results have been obtained by oversaturating the networks with packets: in each case, the offered load was just above the maximum that the network could handle without growing the packet queues at stations.

PS	A	Ethernet	Naive	B	Ethernet	Naive
16	1.24	0.01	0.02	1.10	0.01	0.02
32	1.44	0.01	0.04	1.17	0.01	0.03
64	2.00	0.02	0.07	1.40	0.02	0.05
128	2.25	0.04	0.13	1.66	0.04	0.10
256	2.35	0.09	0.23	1.24	0.08	0.19
512	2.50	0.17	0.38	0.95	0.15	0.32
1024	2.60	0.33	0.53	0.95	0.29	0.47
2048	2.60	0.53	0.67	0.98	0.48	0.60
4096	2.62	0.704	0.77	1.00	0.65	0.71
8192	2.65	0.818	0.84	1.01	0.78	0.80
16384	2.65	0.876	0.88	1.01	0.85	0.85

Table 1: Maximum throughput versus mean packet length.

The improvement over standard Ethernet is presented graphically in Figure 4. The curves represent the ratio of the maximum throughput achieved by networks A and B (and also by the corresponding naive solutions denoted by N_A and N_B) to the maximum throughput achieved by Ethernet.

Figures 5 and 6 show the "delay versus throughput" characteristics of network A (the corresponding characteristics for network B were very similar) for two packet sizes: 32 and 512 bits. The vertical axis represents the mean observed packet delay measured since a packet becomes queued for transmission until it is completely and successfully received at its final destination.

The common shape of all the curves in Figure 5 results from the batching phenomenon. For a relatively low traffic intensity, all packets are transmitted via local channels. When the traffic reaches the batch formation level, the capacity of the global channel becomes relevant and the network enters a different mode of operation (the plateau area). This mode reaches its limit when the global channel becomes saturated.

The impact of the batching phenomenon on network performance is also

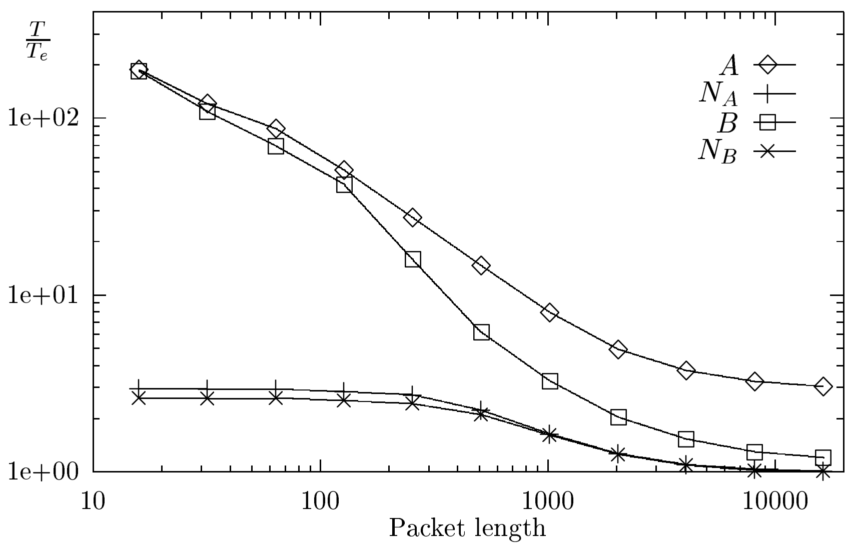

Figure 4: Improvement in the maximum achievable throughput versus packet length.

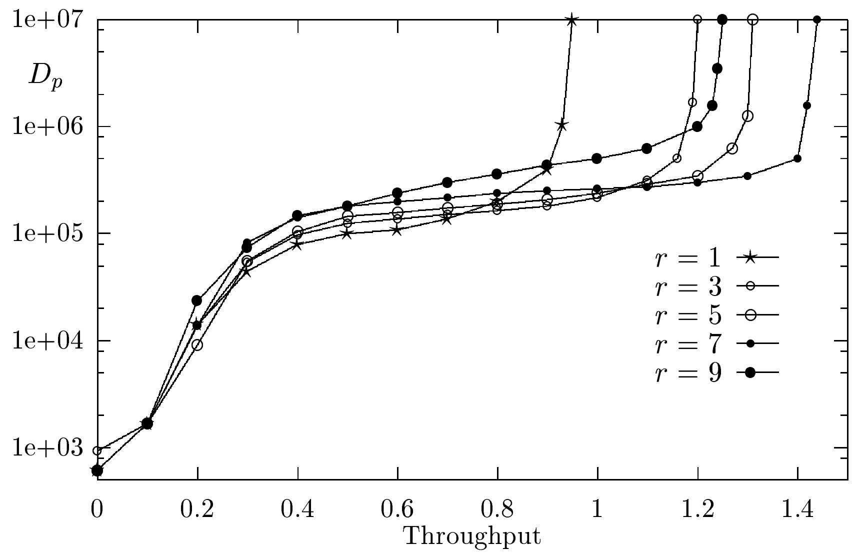

Figure 5: Delay versus throughput for the "A" network, mean packet length = 32 bits.

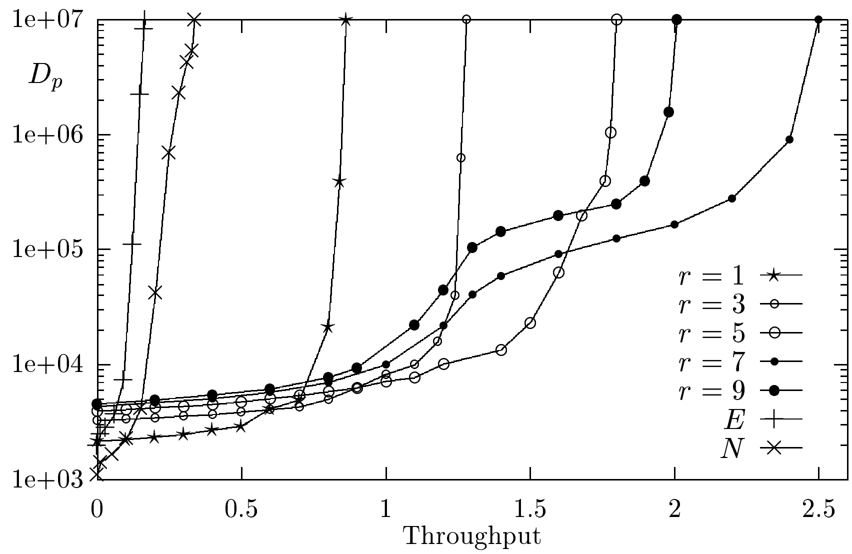

Figure 6: Delay versus throughput for the "A" network, mean packet length = 512 bits.

perceptible in Figure 6, although to a much lesser extent. One additional phenomenon visible there is the trade-off between the lower values of r, which tend to reduce delay for light load, and higher values of r, which offer better throughput for the price of a slightly increased delay under lighter traffic conditions. When r is low, more short packets are transmitted via the global channel. This happens until batches start to form; at that point transmitting short packets over the global channel has a detrimental impact on global channel utilization.

Figure 6 includes performance curves for Ethernet and the naive solution. No such curves are drawn in Figure 5; they would have been difficult to separate from the vertical axis (see Table 1).

3.4 Confidence of the Results

The confidence of the maximum throughput estimates in Table 1 and Figure 4 is very high. Starting from the moment when the network reaches a steady behavior, the observed throughput ceases to grow and remains practically the same, irrespective of the number of received packets. This equilibrium value exhibits zero variance across different simulation runs (with different random seeds).

As far as the delay versus throughput characteristics are concerned (Figures 5 and 6), the situation is slightly more complicated. Generally, as the points get closer to the saturation threshold, their confidence decreases. This happens for two reasons. First, the standard deviation of the message delay tends to grow faster than the mean. Second, batches tend to be longer which increases the correlation of the message delay measured for different packets. To get some insight into the problem, let us consider the curve $r = 7$ from Figure 5 and the throughput value 1.4. The part of the curve located to the right of this point is very steep and its confidence is not very interesting. On the other hand, all the points located to the left of throughput $= 1.4$ will have shorter confidence intervals. Assume that the probability $\alpha = 0.95$. The observed mean message delay at our point is $m = 505,000$ bits and the observed standard deviation $\sigma = 350,000$. The total number of samples from four experiments is $N = 400,000$; thus, the length of the confidence interval for $\alpha = 0.95$ is:

$$\Delta = 2\frac{\sigma z_\alpha}{\sqrt{N}}$$

where z_α is the area under the normal probability curve between 0 and $\alpha/2$. Substituting the numerical values for the parameters, we get $\Delta \approx 2170$, which is slightly more than 0.4% of the observed mean. To be on the safe side, we should divide N by the mean observed batch length (assuming that the delay of packets from the same batch is strongly correlated). The mean batch length in this case was about 7 which gives us a slightly worse interval $\Delta = 5740$. This interval is below 1.2% of the observed mean and the confidence of the results is still quite high.

4. Conclusions

We have presented two solutions to the problem of short packets in CSMA/CD networks. The first (naive) solution is a simple modification of standard Ethernet. Our experiments demonstrate a significant superiority of the second, refined, protocol over the first, naive, solution. This improvement is achieved at the cost of more channels and buffer space.

The major contribution of this work can be summarized as follows:

- A new Ethernet topology: the proposed architecture shares many good features of standard Ethernet, which are augmented by additional features such as higher throughput, suitability for short-packet environments and cost-effectiveness.

- New communication protocols: the proposed CSMA/CD protocols take into account many factors, including the statistical nature of CSMA/CD, the packet size, the geometry of the network and certain hardware parameters (such as the channel capacity).

There are some interesting topics for further investigation. It is believed that the queuing analysis of the proposed scheme will reveal important relationships between the system parameters.

References

[1] P. Gburzyński and P. Rudnicki. A better-than-Token protocol with bounded packet delay time for Ethernet-type LAN's. In *Proc. of Symposium on the Simulation of Computer Networks*, pages 110–117, Colorado Springs, Co., Aug. 1987. IEEE.

[2] P. Gburzyński and P. Rudnicki. Bounded packet delay protocols for CSMA/CD bus: Modeling in LANSF. In *Proceedings of the 19th Annual*

Pittsburgh Conference on Modeling and Simulation, volume 19, pages 1185–1194. Instrumentation Society of America, 1988.

[3] P. Gburzyński and P. Rudnicki. LANSF: a protocol modelling environment and its implementation. *Software Practice and Experience*, 21(1):51–76, Jan. 1991.

[4] P. Gopal and J. Wong. Analysis of a hybrid Token-CSMA/CD protocol for bus networks. *Computer Networks and ISDN Systems*, 9:131–141, 1985.

[5] W. Kiesel and P. Küehn. A new CSMA/CD protocol for local area networks with dynamic priorities and low collision probability. *IEEE Journal on Selected Areas in Communications*, 1(5):869–876, 1983.

[6] W. Loucks et al. Short-packet transfer performance in local area rings. In *Proceedings of IEEE INFOCOM'84*, pages 174–181, 1984.

[7] A. Marson and D. Rofinella. Multichannel local area network protocols. *IEEE Journal on Selected Areas in Communication*, 1:885–897, Jan. 1983.

[8] R. Pickholtz and M. Abdelaziz. Performance analysis of a priority user protocol embedded in a CSMA/CD channel. In *Proceedings of IEEE INFOCOM'85*, pages 419–425, 1985.

[9] J. Shoch et al. Evolution of the Ethernet local computer network. *IEEE Computer*, 15(8):10–26, Aug. 1982.

[10] W. Stallings. Local network performance. *IEEE Communications Magazine*, 22(2):27–36, Feb. 1984.

[11] W. Stallings. *Local Network Technology*. IEEE Computer Society Press, 1985. second edition.

[12] P. Wong and T. Yum. A multichannel computer network with local and global transceiving media. In *Proceedings of IEEE INFOCOM'85*, pages 142–150, 1985.