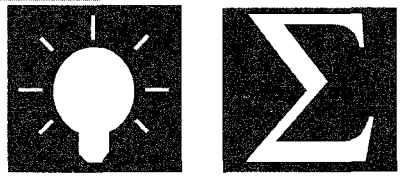


Frame Bursting: A Technique for Scaling CSMA/CD to Gigabit Speeds

Mart Molle, University of California, Riverside
Mohan Kalkunte and Jayant Kadambi, Advanced Micro Devices, Inc.

Abstract

Gigabit Ethernet supports the transmission of ordinary Ethernet frames at a data rate of 1000 Mb/s. Both flow-controlled full-duplex point-to-point links and half-duplex shared collision domains are included in the IEEE 802.3z draft standard. The parameters for half-duplex operation were chosen to align with the requirements of current generic building cabling standards, rather than to match the natural way that network size scales inversely with speed, so a star-wired single repeater topology with a maximum diameter of 200 m is permitted. Thus, gigabit Ethernet is the first time that the CSMA/CD medium access control algorithm has been applied to networks in which the round-trip propagation delay can be much greater than the transmission time for a minimum length frame. In this article, we describe the changes made to CSMA/CD that allow it to support large propagation delays without increasing the minimum frame length or changing its existing one-frame-at-a-time service interface. First, carrier extension is used to decouple the slot time from the minimum frame length, so the slot time can be increased without changing the Ethernet frame format. Second, frame bursting is used to reduce the overhead for transmitting small frames by allowing a host to transmit more than one frame without ever releasing control of the channel. Using simulation, we show that CSMA/CD with carrier extension and frame bursting operating on 1000 Mb/s links provides a significant performance increase over 100 Mb/s Fast Ethernet.

The development of gigabit Ethernet began in late 1995. The IEEE 802.3 working group formed a higher-speed study group at the November 1995 IEEE 802 Plenary Meeting and, after extensive discussions, the IEEE 802.3z task force was granted final approval in June 1996. Since that time, work on developing a standard for gigabit Ethernet has been moving ahead rapidly. The November 1996 meeting was established as the deadline for submitting new technical proposals so that a set of them could be selected to form the initial basis for the proposed standard. Draft D1 was completed in time for review at the January 1997 meeting. By the March 1997 meeting Draft D2 was ready for review, and a consensus was reached on a stable list of features. Further refinements to the proposed standard were incorporated into Draft 2.1, which was reviewed at the May 1997 meeting. Draft D3 should be ready for a final task force review at the July 1997 meeting before the document begins the formal balloting process—first by the voting members of the IEEE 802.3 Ethernet working group, and then by a larger sponsor ballot pool organized by the 802 LAN-MAN (local/metropolitan area network) Standards Committee — that will lead to an approved standard in early 1998.

Gigabit Ethernet supports both full-duplex point-to-point links (running directly from one host, switch, bridge, or router to another), as well as shared collision domains (using star-wired half-duplex links running between multiple hosts and a single repeater). Initially, fiber optic (both multimode and single-mode) and short-haul copper (150 Ω shielded cable) links will be supported. Later on a low-cost transmission scheme known as 1000BASE-T is planned, which will support gigabit speeds over Category 5 unshielded cable, widely used in commercial buildings.

A major goal for gigabit Ethernet was compatibility with the existing standards for 10 Mb/s and 100 Mb/s operation. It needed to be easy to forward frames between segments running at different speeds to simplify the design of the multi-speed bridges and switches from which large campus networks will be built. It also needed to be easy for users to manage their networks without a lot of retraining. Thus, no changes were made to the existing Ethernet frame format, to the minimum and maximum frame lengths, or to the familiar Truncated Binary Exponential Backoff (BEB) algorithm used for scheduling retransmissions in half-duplex networks. However, because of the necessity of handling very large propagation

0890-8044/97/$10.00 © 1997 IEEE

delays in shared half-duplex networks, some changes to the carrier sense multiple access with collision detection (CSMA/CD) medium access control (MAC) layer protocol were required, as described in the second section.

Initially, gigabit Ethernet will focus on switched full-duplex networks for backbone applications, since a switched network can support much higher aggregate throughput than the equivalent shared CSMA/CD system. However, switched full-duplex connections are inherently more costly than half-duplex connections because the core electronics for a switch are much more complex than for a repeater. An interesting compromise is the unintelligent bridge architecture being proposed by some vendors as a "full-duplex repeater." In this design, the hosts are connected to the device by tightly flow-controlled full-duplex links, with a MAC-layer controller and a small amount of input buffering on each port. However, unlike a conventional bridge or switch, the frames arriving on each port are broadcast to all other ports to avoid the need to perform address filtering or provide a higher-speed internal backplane. In terms of both cost and performance, a "full-duplex repeater" falls somewhere between a conventional half-duplex repeater and a switch. At this time, however, the optical transceivers represent the majority of the cost for all of these devices, so these differences probably do not matter until the arrival of 1000BASE-T.

Eventually, gigabit CSMA/CD could become more cost-effective for some applications than switched Ethernet running at a lower speed, especially for bursty traffic sources that can take advantage of intermittent access to the full bandwidth of a gigabit Ethernet link. Shared CSMA/CD networks can also make use of physical-layer technologies that are only capable of half-duplex operation, such as the four-pair signaling used in 100BASE-T4. Thus, CSMA/CD may have some applications even at very high data rates, and the IEEE 802.3z task force has decided to include it in the proposed Gigabit Ethernet standard [1].

Timing Issues in CSMA/CD

CSMA/CD is a distributed algorithm that defines a method for allowing several active hosts to serialize their transmissions on a shared network. Thus, before starting to transmit a frame, each host uses *carrier sensing* to see if the network is currently available and, if not, to *defer* its own attempt until the end of the current carrier event. Once the frame transmission begins, the host continues to look for other traffic on the network using *collision detection*, which triggers the host to abandon this attempt and schedule another after a suitable backoff delay has expired.

The *slot time* is a critical parameter for the CSMA/CD algorithm. It is derived from the worst-case round-trip delay in a network, expressed in bit transmission times (BT). The slot time is also used as the discrete delay quantum in the backoff algorithm as well as defining the minimum frame length. Restricting the backoff delay to integral multiples of the slot time leads to a useful topology-independent fairness property. If two colliding hosts choose different backoff delays, in the absence of other activity they will not collide with each other again, independent of the collision event timing and their relative positions in the network.

The relationship between the slot time and the minimum frame length is also very important to the proper functioning of a half-duplex Ethernet system. Although Ethernet offers only a "best effort" service without any guarantee of delivery, users still expect the frame loss probability to be very small under normal circumstances, and also that there will be no duplication of frames by the network. Ensuring that the duration of each successful frame transmission is at least one slot time is important for achieving both of these requirements. The sender can use the absence of a collision during transmission as an implicit acknowledgment, since the sender would have detected a collision, had there been one, within the first slot time. Similarly, the receivers can use this requirement to filter out incoming collision fragments based on a length threshold. Conversely, if shorter frame transmissions were allowed, two frames that collide at the receiver may not collide at the transmitters, and vice versa. Therefore, some frames might get lost because the sender is unaware that a collision occurred at the receiver, and some frames might get duplicated if the receiver accepted a frame that was later retransmitted because the sender thought a collision occurred.

Since the signal propagation velocity in a link is set by the laws of physics, any increase in the data rate in a CSMA/CD network must be accompanied by either a decrease in the maximum distance spanned by the network or an increase in the slot time. When the IEEE 802.3u Fast Ethernet standard [2] raised the data rate from 10 Mb/s to 100 Mb/s, the slot time was left unchanged at 512 bit times, and the maximum distance spanned by the network was reduced accordingly.[1] The resulting reduction in network diameter to 205 m was deemed an acceptable compromise because of the trend toward increasing network segmentation [4]. Thus, for most installations, a network span that can handle the ports connected to a single wiring closet is sufficient, and current generic cabling standards — both Electronics Industry Association/Telecommunications Industry Association (EIA/TIA) and International Organization for Standardization (ISO) [5] — specify that the distance from an office to the nearest wiring closet should be less than 100 m. Obviously, however, there is no possibility of imposing a further distance reduction to compensate for the next speed increase from 100 Mb/s to 1000 Mb/s. Thus, the IEEE 802.3z task force has specified that the slot time will increase from 512 *bits* to 512 *bytes* (i.e., 4096 bit times) for 1000-Mb/s networks.

CSMA/CD Extensions to Handle a Larger Slot Time

Carrier Extension

Defining a larger value for the slot time parameter for half-duplex 1000-Mb/s operation was just the first step in developing a usable version of CSMA/CD for gigabit Ethernet. Fortunately, an increase in the slot time represents a very simple change to the backoff algorithm, which causes no unwanted side effects. The resulting proportional increase in the retransmission delays merely allows the other hosts to transmit more frames between each attempt, and hence reduces the likelihood of more collisions. However, something must be done to bring the transmission time for a minimum-length frame back in line with the slot time.

Maintaining a compatible frame format over all operating speeds was a very high priority, so simply increasing the minimum frame length for 1000-Mb/s operation was not an acceptable solution. Otherwise, bridged multispeed networks would not work very well. First, bridges would be forced to reformat every short frame before it could be forwarded from a lower-speed link to a gigabit Ethernet link. Second, if servers were

[1] *In addition to signal propagation at about 200 m/s, the minimum round-trip delay for a single-repeater network includes a considerable bit budget allocated to the circuitry in the host adapters and repeater, which is about 80 BT for 10BASE-T [3, Table 13-2], about 180 BT for 100BASE-T [2, Table 29-3], and about 1600 BT for gigabit Ethernet [1, Table 42-3].*

given gigabit Ethernet connections, each of their short acknowledgment frames would be eight times longer than necessary, needlessly burdening the low-speed client networks. Therefore, the IEEE 802.3z task force has adopted a technique known as *carrier extension* to decouple the minimum frame length from the slot time for half-duplex 1000-Mb/s operation [6].

Under carrier extension, the minimum frame length remains 512 *bits* (as it is for 10-Mb/s and 100-Mb/s networks), but the minimum length of the associated carrier event for every successful transmission is increased to 512 bytes in the following way. If the transmitter reaches the end of an outgoing frame without detecting a collision, it looks at the frame length. If the length was at least one slot time, the transmitter returns a *transmit done* status code to its client in the usual way. However, if the length is less than one slot time, the transmitter withholds the status code and continues transmitting a sequence of special *extended carrier* symbols until the end of the slot time, when it returns the *transmit done* status code. Note that carrier extension takes place *after* the checksum that marks the end of the frame — it is not part of the frame itself, and is handled locally within each collision domain. Should the transmitter detect a collision at any time during this process, however, it truncates the remainder of the outgoing frame (or extended carrier) transmission, and then sends a 32-bit jam signal in the usual way.

Note that the jam signal always looks like data to the receivers, even if the collision happens during the extension. Thus, a collision in the extension will cause the frame to be dropped by the receivers. In particular, great care was taken to avoid the possibility of duplicating a very short frame if a collision occurs late in the extension. The concern was that the receiver could have gotten the entire frame before the start of the collision, so it passes the checksum test, and if the collision occurred late enough, the addition of the 32-bit jam might extend the fragment enough to pass the slot time test.

Carrier extension also affects the actions of the receivers. Normally, the bit receiver process at each host searches the incoming bitstream for the preamble and start-frame delimiter that marks the start of a new frame. Thereafter, it starts counting incoming bits and accumulating those bits that are not extended carrier symbols into a receive buffer until the frame ends. At that point, if the total number of incoming bits is below one slot time, the incoming frame is discarded as a collision fragment, even if the receive buffer contains a perfectly valid frame; otherwise, the receive buffer is passed to the MAC layer for checksum and address checking. Carrier events shorter than the slot time indicate that the transmitters thought there was a collision, and may have chosen to retransmit this frame. Thus, even if the collision occurred during the extension and left the actual frame unharmed, it must be discarded to avoid duplication.

On one hand, carrier extension represents a very minor change to the existing CSMA/CD algorithm, and it solves the problem of how to increase the slot time without altering the minimum frame length or any other properties of the algorithm. However, carrier extension also increases the transmission time for short frames significantly, which reduces the benefit of the increased data rate. In the worst case, upgrading the network connection from 100-Mb/s fast Ethernet to 1000-Mb/s gigabit Ethernet for a host that generates only minimum-length (64-byte) frames would allow it to send bits 10 times faster than before, while requiring it to send eight times as many bits per frame, resulting in only a 25 percent net increase in throughput! Of course, network traffic rarely consists of just minimum-length frames, so the overhead caused by carrier extension is generally much smaller. Nevertheless, there is the potential for significant performance improvement if we can find a way to add *pipelining* to the frame transmission process in CSMA/CD.

Frame Bursting

Pipelining is widely used in automatic repeat request (ARQ) algorithms at the data-link layer [7, §6.4], and at first glance the basic go-back-*N* algorithm seems appropriate. Although this feature was included as part of more radical proposals by several authors [8, 9], only the most basic block-oriented approach, known as *packet packing* [10, 11], received serious consideration by the Gigabit Ethernet group. A continuous ARQ, controlled by a sliding window, would be completely out of the question: if frames can be shorter than a slot time, normal frame transmission in shared Ethernet is as complex as the reliable multicast problem.

The idea in packet packing was to boost efficiency by allowing a transmitter to combine several frames into a single block, to which carrier extension would be applied if it were still too short to fill up the slot time. Packet packing can recover essentially all of the efficiency lost because of carrier extension. Unfortunately, however, this proposal also requires substantial changes to the service interface between the CSMA/CD MAC layer and its client, since the transmission unit is now a sequence of frames instead of a single frame. Thus, the MAC-layer transmitter would no longer be able to return a status code to its client for each frame before starting to work on the next request. (Indeed, the MAC layer could have as many as eight unacknowledged minimum-length frames under its control at any given time.) Similarly, the MAC-layer receiver must "quarantine" the incoming frames within a given block, withholding them from its client until the end of the slot time to avoid duplicating frames in case a collision causes the sender to retransmit the entire block. Furthermore, the possibility that the sender may back up and retransmit several frames also affects the management interface, where various activity counters can no longer be updated after every frame transmission or reception event. In the end, the implementation of packet packing was deemed too complex, and it was not included in the IEEE 803.3z draft standard.

The rejection of the packet packing proposal did not make the inefficiency of carrier extension go away, so work continued on finding other ways to add pipelined frame transmission to CSMA/CD without changing the MAC layer's familiar single-frame-at-a-time service interface. The resulting method, known as *frame bursting* [12], includes features from several sources. Like packet packing, the sender can transmit several frames, separated by extended carrier, in a single burst. However, the maximum burst length was based on the maximum frame length instead of the slot time, like the binary logarithmic arbitration method (BLAM) [13, 14]. In addition, the transmission time for the first frame in each burst is padded to a full slot time using extended carrier, if necessary, which was also used in [9]. This feature ensures that collisions can only affect the first frame of any burst, so both the sender and receiver can retain their familiar one-frame-at-a time service interfaces.

Frame bursting can be implemented by adding some new static variables to the MAC transmitter, creating a "trap door" that allows some calls to bypass the normal CSMA/CD access rules, provided that this frame can be added to an already established burst. More precisely, frame transmission is done in the following way:

1. Before attempting to send a frame, the transmitter checks to see if its *burst timer* is running. If not, the transmitter must follow the normal rules for CSMA/CD — deferring to other activity, backing off after collisions, and so on —

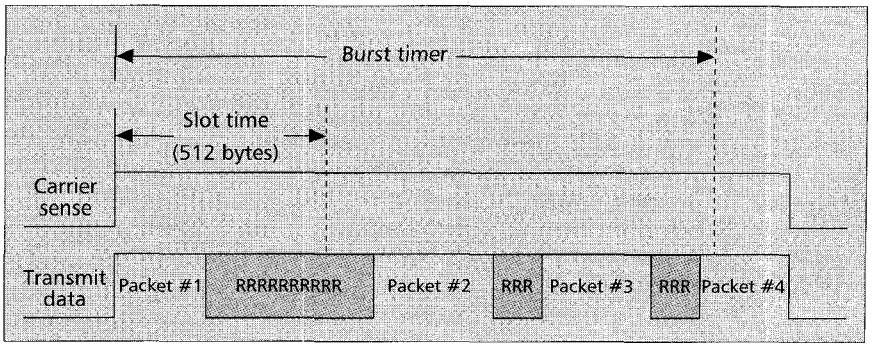
except that at the start of each attempt it sets a flag to indicate that this will be the *first frame* in a burst and starts its *burst timer*. Otherwise, it takes the "trap door" option and initiates the transmission immediately.

2. If the attempt succeeds, the transmitter looks at the *first frame* flag. If the flag is set, the transmitter keeps sending extended carrier until the *burst timer* reaches one slot time, if necessary, then clears the *first frame* flag. If there is still no collision, the transmitter returns the *transmitOK* status code to its client, indicating that this frame has been successfully delivered, and goes on to step 3 for some post-processing. Otherwise, there must have been a collision, in which case the transmitter clears the *first frame* flag and the *burst timer* and returns to step 1 to make another attempt according to the usual rules for CSMA/CD.
3. At this point, the transmitter has just finished a successful transmission and therefore still has control of the channel. Thus, if the *burst timer* has not yet expired, it sends another 96 bits of extended carrier (which serves as the interframe space within the burst) and goes on to step 4. Otherwise, the burst is over, so nothing more happens until the next request arrives.
4. At this point, the last call to the transmitter has kept the burst alive for the 96-bit interframe space to allow some time for the MAC client layer to make another request. If another frame is available, its transmission begins now; otherwise, the host must forfeit the remainder of its burst time allotment, so the *burst timer* is cleared, and nothing more happens until the next request arrives.

An example of the sequence of items in a frame burst is shown in Fig. 1. Notice that even if the first frame in a burst is short enough to require carrier extension, the frame will still be followed by a separate interframe spacing field which begins at the slot time boundary. This is done to give the receivers some time to process one frame before facing the arrival of the next. Figure 1 also demonstrates that the last frame of a burst must start before the burst timer expires, but its transmission may extend beyond the burst limit.

Under frame bursting, the decision to allow another frame into an ongoing burst is based on the outcome of two tests. First, the transmission of the next frame must begin before the *burst timer* reaches a certain cutoff value. Second, the next frame must be available for transmission before the end of the 96-bit interframe spacing period. In Draft D2, the burst limit was set at 12,000 BT, which was chosen to be just below the maximum frame length to promote fairness. Thus, no matter what mix of frame lengths it has, a host can keep extending its burst until it has transmitted for at least the equivalent of one maximum-length frame, and for no more than (roughly) twice the maximum frame length in the worst case. However, further investigation showed that increasing the burst limit led to significant performance benefits, so at the May 1997 meeting the burst limit was increased to 8 kB (or 65,536 BT). This fivefold increase in the burst limit also serves to restore the balance between the "cost" of contention (measured in slot times) and the associated "reward" (measured in the amount of data transmitted) that was present in 10-Mb/s and 100-Mb/s systems. Thus, in our simulations we have included results for both of these choices.

The difference between using a starting time threshold rather than a finishing time threshold to limit the burst length may seem small, but it can make a significant difference to the designer of a MAC-layer device. For example, suppose that the burst timer is implemented as a simple counter that shuts off when it counts down to zero. In that case, the logic to support a starting time threshold is just a test to see if the timer is still running, whereas an ending time threshold involves comparing the next frame length to the current timer value. Furthermore, some MAC-layer implementations may not even know the length of an outgoing frame at the moment they receive a transmission request,[2] but may wish to start transmitting anyway in order to reduce store-and-forward latency in the network adapter.



■ Figure 1. *The structure of a burst.*

Adding frame bursting to the MAC layer receiver in CSMA/CD requires a similar set of modifications:

1. To receive the first (or only) frame in a new burst, the receiver follows the normal rules for CSMA/CD, skipping across the incoming data stream until it finds a valid preamble and start-frame delimiter. At that time, the receiver sets the *extending* flag to indicate that it is looking for the first frame in a burst, which is the only one subject to the carrier extension rule. Thereafter, it starts counting the incoming bits from this frame (and gathering those bits that are not extended carrier to form the incoming frame) until it finds the end, indicated by either end-of-carrier or the appearance of the first extension bit after the extending flag has been cleared, and then goes on to step 2. Meanwhile, the receiver compares the number of incoming bits with the slot time after each bit, and clears the *extending* flag as soon as they are equal.
2. At this point, the receiver has found an incoming frame to which it must apply the collision filtering rules. First, it checks to see if the *extending* flag is set and, if so, throws the frame away as a collision fragment. (This flag should have been cleared during reception of the first frame of a burst, which is extended to a slot time unless it is a collision fragment, and thereafter the flag remains off for the rest of the burst.) Next, the receiver applies the normal collision filtering rules to discard the frame if its length is less than the Ethernet minimum frame length or it has an invalid checksum. Finally, it passes the frame to its client, if it has not already been thrown away, and then goes on to step 3.
3. If the frame ended because of end-of-carrier (i.e., the sender did not try to make another attempt), or the interframe period is followed by end-of-carrier instead of another preamble and start-frame delimiter (i.e., the sender forfeited the rest of its burst), then this burst is over and the receiver returns to step 1 and starts looking for the next burst. Otherwise, the receiver starts gathering incoming bits again to form the next incoming frame until it is terminated by either end-of-carrier or the first extension bit, and returns to step 2 with *extending* off.

[2] *Remember that Ethernet frames do not necessarily contain a length field, and looking inside the payload at the IP header, say, is not possible because the MAC layer has no idea what protocol stack is running above it.*

Notice the significance of the *extending* flag in this algorithm. First, it is used in an obvious way to ensure that the extended carrier length filter is only applied to the first frame in each burst. Second, its value is used in combination with an extension bit to generate two "meta-symbols"; that is, if *extending* is on, extension bits are treated like carrier, whereas if *extending* is off, they are treated like idle time, except for resetting the *extending* flag.

Modeling Approach

A simulation model was developed in OPNET [15] to study the effect of frame bursting on gigabit Ethernet performance. OPNET is a hierarchical object-oriented protocol simulation model developed by MIL3 Inc. The basic models provided in OPNET were enhanced to reflect the additions to CSMA/CD in gigabit Ethernet, namely carrier extension, a burst timer, and frame bursting. A maximum topology for the gigabit Ethernet network was assumed. Details such as collision detection, backoff, deference mechanism, and both cable and hub delays were also modeled. Each data point in the figures was obtained by running the program for 30 s of simulated time. Depending on the network load, somewhere between 1 and 6 million frames will be transmitted in a single run.

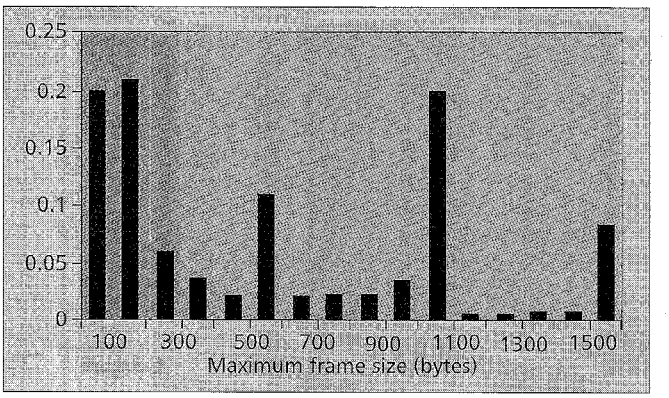
A simple "open" frame generation process was employed at each host, keeping in mind that gigabit Ethernet is being positioned initially as a backbone technology. The traffic generator is independent of the current state of the network (in particular, there is no "flow control" to throttle the arrivals to a host if the network is already busy), and there is no upper bound on the size of the transmit queue at each host. Clearly, a "closed" frame generation process which reduces the arrival rate for new traffic as the number of queued frames increases would be more representative of applications that use acknowledgment-based flow control schemes, such as Transmission Control Protocol/Internet Protocol (TCP/IP). However, we do not yet know what applications for shared gigabit Ethernet might look like, and in any case the network can certainly support a very large number of such flow-controlled sessions, which would tend to reduce the difference between the "open" and "closed" models. On the other hand, network traffic has been found to exhibit long-range dependencies that make the arrivals more "bursty" than the simple Poisson model we included in our experiments, which would tend to increase the benefits of frame bursting under light load. Traffic characterization is an important topic that is beyond the scope of this article.

Performance

Single-Transmitter Worst-Case Overhead

Depending on the traffic mix, frame bursting may provide a significant reduction in overhead by reducing the number of times carrier extension must be applied. Obviously, if all frames are at least one slot time in length, none of these modifications will have any effect. At the opposite extreme, with short frames they can lead to dramatic differences in their respective worst-case efficiencies. To see this, let us compare:

- Ordinary 100-Mb/s CSMA/CD (512-bit slot time and no carrier extension)
- Baseline 1000-Mb/s CSMA/CD (4096-bit slot time with carrier extension only)
- 1000-Mb/s CSMA/CD with 1500-frame bursting (4096-bit slot time, 12,000-BT burst limit and carrier extension)
- 1000 Mb/s CSMA/CD with 8000-frame bursting (4096-bit slot time, 65,536-BT burst limit, and carrier extension)



■ Figure 2. *Workgroup packet size distribution.*

In each case, we assume that a single busy source is attempting to transmit large numbers of frames over an otherwise quiet network. This situation is not as unrealistic as it may seem, because the dynamics of CSMA/CD under heavy load lead to a capture effect where one host can transmit large numbers of consecutive frames while blocking the rest of the hosts from transmitting anything. The capture effect is described in more detail later.

In general, the efficiencies for each method can easily be calculated by finding the ratio of the number of bits sent per cycle to the time taken for that cycle. For example, in ordinary 100-Mb/s CSMA/CD with a frame length of P bits, there are P bits sent in a cycle, and the duration of the cycle is (96 + 64 + P) BT, where we have added the interframe spacing, preamble, and start-frame delimiter. Since this ratio is clearly an increasing function of the frame length, P, its minimum occurs at the minimum frame length, where P = 512 and the normalized efficiency is 76 percent. If we now increase the speed to 1000 Mb/s and add carrier extension, the only change needed is to replace P by max{P, 4096} in the denominator to account for the time to transmit any extension bits, which lowers the worst-case normalized efficiency to only 12 percent.

The worst-case normalized efficiency calculation for 1000-Mb/s CSMA/CD with frame bursting is slightly more complex, because the host can transmit multiple frames in a cycle, and we must include both the total amount of data sent and total time required in our calculations. First, it is easy to see that if the first frame has a length P, it contributes P bits to the numerator and max{P, 4096} bits to the denominator, with P = 512 as the worst case. After that, the rest of the time until the burst timer expires will be filled with additional frame transmissions having the same amount of overhead as ordinary CSMA/CD. Thus, since the efficiency is worst with minimum-length frames, the worst-case normalized efficiency occurs when we send a burst that consists entirely of minimum-sized frames. In particular, if the burst limit is 12,000 BT, the timer does not expire until after we have started $\lceil (12{,}000 - 4096)/672 \rceil = 12$ extra frames, giving a worst-case efficiency of

$$(13 * 512)/(4096 + 12 * 672) = 55\%.$$

This is more than 70 percent of the worst-case normalized efficiency for ordinary CSMA/CD, and more than 4.5 times higher than the baseline proposal for 1000 Mb/s CSMA/CD with carrier extension. If we increase the burst limit to 65,536 BT, we can send 93 minimum-sized frames in a burst to achieve a worst-case efficiency of

$$(93 * 512)/(4096 + 92 * 672) = 72\%,$$

which is almost 95 percent of the worst-case normalized efficiency for ordinary CSMA/CD.

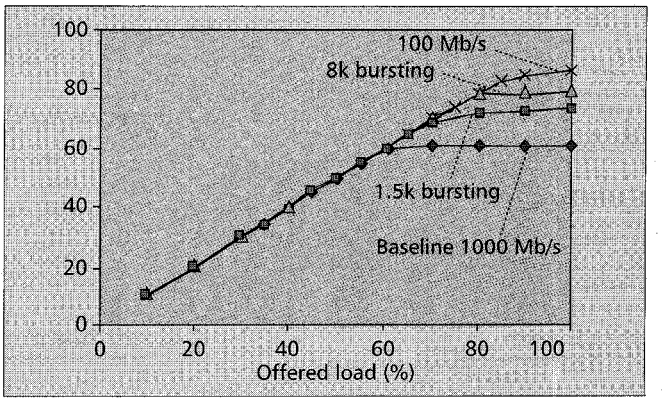
Workgroup Average Throughput

In order to get a more balanced view of the significance of these differences in efficiency, we ran a series of experiments with the event-driven simulation models described in the third section. Each model was run with a simple empirical traffic model, known as the *workgroup average distribution*, which is shown in Fig. 2. These data were derived from traffic measurements performed on several 10-Mb/s and 100-Mb/s networks at Sun Microsystems, Advanced Micro Devices, and 3Com, and presented to the IEEE 802.3z group in spring 1996 [16, p.21]. By using this workgroup average distribution to select the frame lengths in our study, the traffic in our experiments included a mixture of frame lengths, most of which were either very short (i.e., a control message or perhaps a few keystrokes of interactive data) or quite long (i.e., large data segments, with the peaks in the distribution attributed to the default Internet segment size, multicast video, and the maximum Ethernet frame size, respectively). Similar frame length distributions have also been reported in other network measurement studies.

It is interesting to note that the average frame length in the workgroup average distribution is slightly below 600 bytes, but when we expand the short frames using carrier extension, the average frame length rises to slightly more than 750 bytes. As we will see below, adding frame bursting improves performance on heavily loaded networks enough to *completely eliminate* this 20 percent overhead penalty. That is, for all of the metrics shown below, performance with frame bursting is better than that without frame bursting, even when the network load is reduced by 20 percent to compensate for the carrier extension overhead.

We did not attempt to reproduce all the temporal features of real network traffic in our model, such as the burstiness of arrivals and the correlation between lengths of successive frames. Instead, we simply generated frame arrivals according to a Poisson process and distributed them randomly among the hosts, and then chose a length for each frame using the workgroup frame length distribution. We expect that the performance measures obtained this way are likely to overestimate the throughput under heavy load because it allows large queues to develop at each host (which would not happen with an acknowledgment-based transport protocol like TCP/IP) as needed by the capture effect. Conversely, we expect our results to underestimate the benefits of frame bursting under light to moderate load because our model lacks the burstiness of long-range dependent traffic models that would allow the hosts to take advantage of frame bursting.

Figure 3 shows the percentage of throughput as a function of percentage of offered load for a 15-host network with the workgroup average traffic model and the four alternatives described above. The same experiments were also run on a four-host network, but the results are qualitatively similar and will not be shown here to save space. In all cases, the throughput curves have the same general form: initially the system is traffic-limited, so the throughput curves rise in lockstep with the load to form a straight 45° line until we approach their respective maximum efficiencies, at which point the queues saturate and the curve turns horizontal. The two parts of each curve do not form a sharp corner where they meet because some frames are dropped with excessive collision errors. However the corner is visibly more sharp for the 1000-Mb/s curves because the longer backoff delays (relative to the frame transmission time) reduce the collision rate and hence the number of excessive collisions.

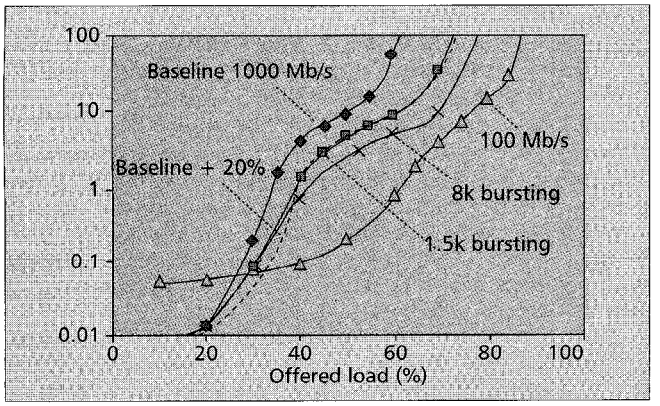


■ Figure 3. *Network throughput (percent).*

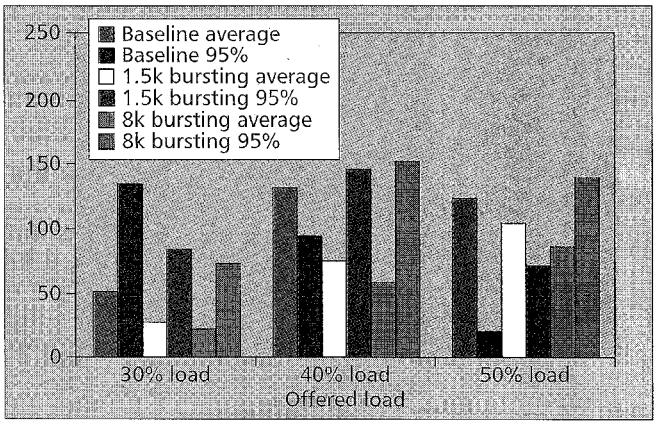
With the workgroup average frame length distribution, the performance penalty for adopting the baseline carrier extension scheme to handle the larger slot time for 1000-Mb/s operation is almost a 30 percent reduction in the maximum percentage throughput (i.e., about 61 percent versus 86 percent for the 15-host system). Part of this drop is caused by the extra overhead of carrier extension; part is just lost bandwidth due to larger collision fragments. Adding 1.5k-frame bursting lets us recover almost half this loss, raising the maximum throughput from 61 to 72 percent, which is very close to the 20 percent reduction in overhead we were hoping for. Increasing the burst limit to 8k raises the maximum throughput to almost 80 percent, which is within 8 percent of the 100-Mb/s fast Ethernet system.

Mean Delay

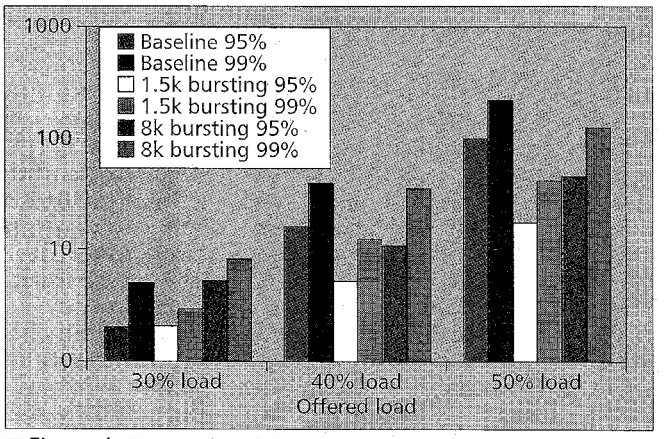
Figure 4 shows the mean end-to-end delay as a function of offered load for a 15-host network using the workgroup average traffic model. The end-to-end delay includes both the *waiting time* and *access latency*, from the moment a frame is generated at the sending host until it has been fully received at the destination. Notice that the delay curve for the 100-Mb/s fast Ethernet system starts higher but extends further to the right because offered load has been normalized (0–100 percent), whereas delay is expressed in absolute units (i.e., milliseconds rather than BT). Once again, the results show that increasing the slot time for 1000-Mb/s operation causes a substantial loss in normalized performance. However, one must keep in mind that the scale for the x-axis has been normalized—for a fixed 1-ms end-to-end delay, the *un*normalized throughput (in megabits per second) of the 100-Mb/s fast Ethernet system is actually about five times lower than the 1000-Mb/s networks, instead of twice as high. Note, also, that the results with frame bursting are uniformly better than without it: 1.5k-frame bursting drops the mean delay by at least a factor of two when the offered load is over 30 percent, and by a factor of 10 at 60 percent load. Raising the burst limit to 8k leads to even greater improvement.



■ Figure 4. *Average end-to-end packet delay (ms).*



■ Figure 5. *Mean and 95th percentile of access delay (μs).*

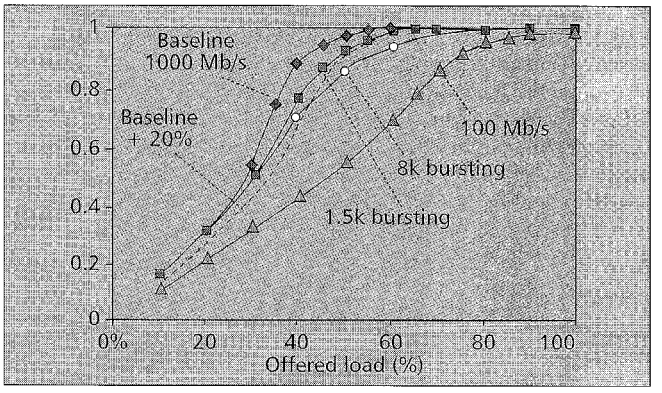


■ Figure 6. *Percentiles of the distribution of consecutive frame transmissions by a single host.*

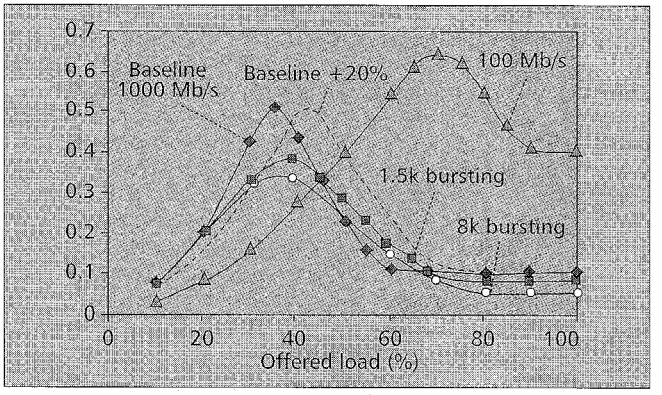
The curves also demonstrate the distinctive nonconvex shape characteristic of half-duplex Ethernet systems. Although the delay starts off small, it rises quite early as the network starts to become congested (and deferrals, collisions, and backoff delays become more common). However, once congestion sets in and the queues start to build up at each host, the slope of the delay curves starts to level off again before finally blowing up as they approach maximum efficiency.

Capture and the Burstiness of Access Latency

This flat spot in the throughput-delay curve is caused by the asymmetry of the Truncated Binary Exponential Backoff algorithm, which leads to the capture effect [13, 17]. Capture causes short-term unfairness in CSMA/CD systems, because a host making its first few attempts to transmit a new frame can be much more aggressive in trying to acquire the network than a host that has already made many attempts. As a result, hosts on a busy CSMA/CD network will experience long periods where they are unable to transmit anything, punctuated by an occasional capture period where they can transmit a large burst of consecutive frames without interference. Capture makes the CSMA/CD algorithm surprisingly efficient under high network loads. The effect of capture is less visible at 100 Mb/s because it uses a smaller value for the slot time, so the "winning" host has time to send fewer frames before having to contend for the channel again.



■ Figure 7. *Probability of transmission delay due to deferral.*



■ Figure 8. *Probability that a frame experiences at least one collision.*

The capture effect is a serious problem for time-sensitive applications. The problem is difficult to see by examining only the *average* access latency, and even its standard deviation does not tell the whole story. For example, in their well-known Ethernet measurement study [18], Boggs *et al.* found that the average access latency is only slightly higher than for an ideal round-robin scheme for a saturated 10-Mb/s Ethernet and that its standard deviation is about three times larger than the mean, neither of which seems problematic. However, when we tried reproducing their experiment via simulation [13], we found that on average the same host was transmitting more than 100 frames in a row each time it captured the network. Thus, any technique (such as frame bursting) that intentionally allows a host to transmit multiple frames without releasing control of the network must be checked to make sure that it does not aggravate the capture effect.

Figure 5, which shows the mean and 95th percentile of the access latency distribution (in microseconds), demonstrates the significance of the burstiness of the access latency under heavy load. Referring back to Fig. 4, it is evident that the given range for the percentage offered load covers the region in which the various 1000-Mb/s options are becoming congested (i.e., the flat spot in the delay curve). In general, we would expect the mean to be well below the 95th percentile of the access latency distribution, as we see in the baseline system at 30 percent load, with 1.5k-frame bursting at both 30 and 40 percent load, and with 8k-frame bursting at all three values for the load. However, the character of the access latency becomes very different under high load because of capture. In particular, notice that the 95th percentile of the access latency for the baseline system is *less than 15 percent of the mean* at the highest load! In other words, the average access latency experienced by the worst 5 percent of the frames must be at least 120 times higher than the average of the *remaining* 95 percent.

The capture effect is further demonstrated by Fig. 6, where we show the 95th and 99th percentiles of the distribution of the number of consecutive frame transmissions by a single host. Notice that the number of consecutive frame transmis-

sions for 1000-Mb/s CSMA/CD systems grows very large as we increase the load, especially in the baseline system without frame bursting. More important, the number of consecutive frames decreases when we add frame bursting. This happens because capture creates an access pattern that looks like exhaustive service in a polling system, and by reducing the single-transmitter overhead each host is able to clear its queue more quickly.

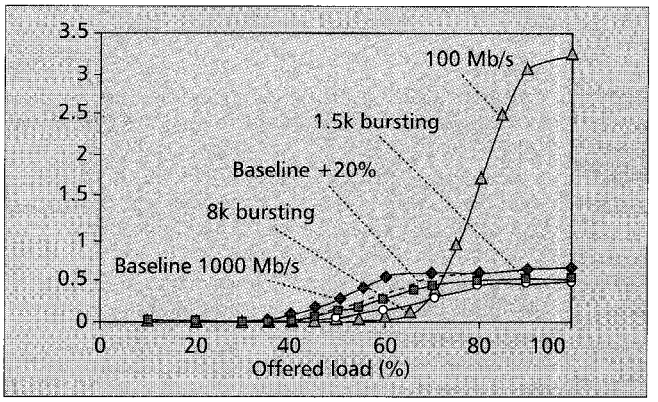
The Effect on Deferrals and Collisions

Since the purpose of CSMA/CD is to control access to a *shared* medium, an important consideration is how the other hosts interfere with a given host's requests for network usage. Basically, there are three levels of interference to consider:

- Using carrier sensing, the host is required to *defer* the transmission of a frame because of some earlier network activity.
- Using collision detection, the host is forced to abandon at least one attempt to transmit its frame and try again because of interference from other hosts' transmissions.
- Following 16 consecutive collisions while attempting to transmit the same frame, the host reports an *excessive collision error* and drops the frame.

In general, a host must defer a transmission if the network is either currently occupied with another frame transmission, or such a transmission has taken place so recently that the channel has not been idle for the required minimum 96-bit interframe spacing. As a result, a host may need to defer because of its own previous transmission. If hosts were to generate their transmission requests at random, then the probability of a deferral would be the same as the probability that the network is "busy" with other frame transmissions, collisions, or waiting for the interframe spacing to end. Thus, the deference likelihood should be no less than the percentage offered load, as is evident from Fig. 7 for the case of a 100-Mb/s network under light traffic. The light traffic deference likelihood for all of the Gb/s networks is about 20 percent higher than the offered load because of the overhead due to carrier extension. Under heavy traffic, the deference likelihood increases for all systems because of the time occupied by collisions. It is again evident that under heavy load, frame bursting can give the same performance while operating at a 20 percent higher offered load.

The probability that a transmitted frame experiences at least one collision is shown in Fig. 8. The results shown are typical of most shared-media Ethernet systems, where the collision likelihood peaks quite early, at the point where there is enough load to cause congestion but not to cause queues to form at each host, which would allow them to take advantage



■ Figure 9. *Percentage of frames with an excessive collision error.*

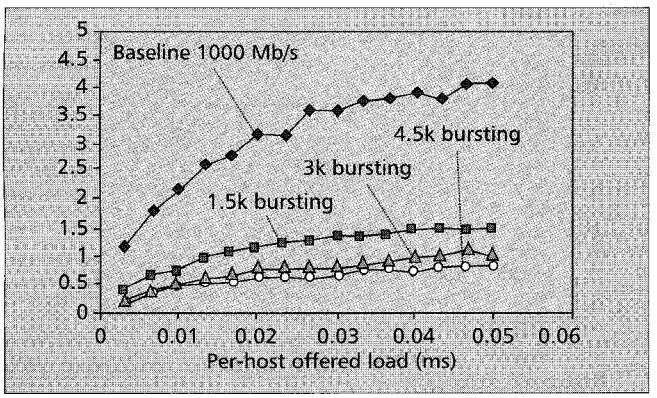
of the capture effect. The peaks of the collision likelihood curves happen much sooner at 1000 Mb/s than at 100 Mb/s because of the increase in slot time, which forces the hosts to retransmit less aggressively after each collision. Similarly, the limiting value of the collision likelihood under high load is much lower in the 1000-Mb/s curves than in the 100-Mb/s curve because the larger slot time allows a host to transmit more frames whenever it captures the network.

In Fig. 9 we show the likelihood that a frame will be dropped because of an excessive collision error. Once again, frame bursting gives us a significant improvement-even when handicapped by a 20 percent higher offered load.

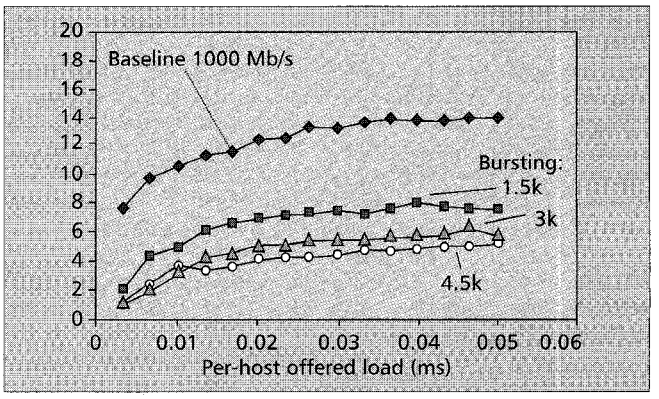
Fairness for Lightly Loaded Hosts

So far, all the experiments we presented have involved equal loading for each host; but what happens when the loadings are different? The concern with frame bursting is that heavily loaded hosts might have an unfair advantage over lightly loaded hosts because newly arriving frames can take the "trap door" path straight to transmission without experiencing any wait and also further increasing the wait for lightly loaded hosts. Thus, in Figs. 10 and 11 we looked for a bias in the end-to-end delays experienced by individual hosts as a function of their relative load. That is, we let the offered load for host i be $i/300$, giving a total offered load of 40 percent and a factor of 15 difference in load between host 1 and host 15. For this experiment, we show the mean end-to-end delay for each host in Fig. 10, and the 95th percentile of the delay for each host in Fig. 11.

It is evident from these figures that frame bursting offers a substantial improvement to all hosts over the baseline carrier



■ Figure 10. *Per-host average end-to-end delay, 40 percent load, nonuniformly assigned to the hosts.*



■ Figure 11. *Per-host 95th percentile of end-to-end delay, 40 percent load, non-uniformly assigned to the hosts.*

extension proposal. There is also a significant amount of additional improvement visible when we increase the burst limit from 12,000 BT to 24,000 BT and even to 48,000 BT. It is interesting to note how the end-to-end delay in the baseline system was an increasing function of the host's offered load, but this effect goes away when we add frame bursting.

An important lesson from these results is that increasing the burst limit does not seem to hurt the hosts with light traffic. If the entire system is lightly loaded, the burst limit has no effect, since the transmitters generally run out of traffic before reaching the limit. For a heavily loaded system, the data suggests that the improvement in transmission efficiency is the dominant factor. Thus, although the maximum deferral time for a light traffic source increases as we raise the burst limit (because the heavy traffic sources can block them from transmitting for longer periods of time), the network becomes less busy overall because the heavy traffic sources can transmit more efficiently. However, the results may be different if the load generated by the heavy users were more bursty (see, e.g., [19]).

And, finally, we note that the data suggests that increasing the burst limit does not increase the buffering requirement at each host interface. To see this, consider that all hosts (including the lightly loaded ones) experienced delay improvements at a given value of network load when we increased the burst limit. Thus, by applying Little's Law [20], we know that the queue lengths must be decreasing as well.

Conclusions

In this article, we have shown that with only a few minor changes, CSMA/CD can operate effectively at gigabit-per-second data rates. First, carrier extension decouples the minimum frame length from the slot time to accommodate the inevitable increase in the bandwidth-delay product without changing the Ethernet frame length. The resulting drop in efficiency when sending short frames is handled via frame bursting, which allows a host to pipeline multiple frame transmissions without changing the existing one-frame-at-a-time MAC-layer service interface.

By permitting a host to transmit a sequence of frames without ever relinquishing control of the medium, frame bursting allows CSMA/CD to achieve a high maximum throughput, even at gigabit-per-second data rates. In principle, carrier extension and frame bursting could be used to scale CSMA/CD to even higher speeds in the future by applying proportional increases to both the slot time and the burst limit. However, to take advantage of the extra bandwidth the queue sizes at each host would also need to grow by the same factor, and hence by applying Little's Law the average end-to-end delay would always rise quickly into the tens of milliseconds.

These changes could also be added to existing Ethernet systems to create 10-Mb/s Ethernet collision domains with a diameter greater than 2 km or 100-Mb/s fast Ethernet collision domains with a diameter greater than 205 m. This is unlikely to happen for several reasons. First, the installed base of existing Ethernet equipment is incompatible with carrier extension, and the length of a carrier event with frame bursting would trigger an excessive-length ("jabber") error condition. Furthermore, there is little incentive to develop a new "large-diameter" operating mode for the existing speeds: because of the trend toward network segmentation for other reasons (management, control of broadcasts, etc.), a 200-m collision domain diameter is enough.

For gigabit Ethernet, however, the benefits of frame bursting are clear. We found that it increases throughput while doing no harm to any of the other measures of performance. In other words, frame bursting improved every measurement we made (including end-to-end delay, access latency, and the likelihood that a frame must defer to other activity, experiences collisions, or gets dropped because of excessive collisions), and even provides equivalent performance even when handicapped by a 20 percent increase in the offered load.

Perhaps the most surprising result from our study is that frame bursting actually *reduces* the impact of the capture effect. The dynamics of capture are important for shared Ethernet because the network starts acting like an exhaustive service system under heavy load. By improving the efficiency with which a single host can transmit multiple frames, frame bursting allows a host to drain its queue more quickly each time it captures the network. In addition, since the other hosts were allowed fewer attempts during this capture period, their backoff delays will grow more slowly also.

There are still many open issues related to shared gigabit Ethernet that will require further study. First, we need to consider the effects of other traffic models on the performance of gigabit CSMA/CD and, conversely, the suitability of gigabit CSMA/CD for those classes of traffic. In particular, the "open" traffic generation model that we used allows large queues to form at each host, which may be appropriate for modeling the aggregation of desktop traffic leaving a workgroup switch, but not for the low-cost desktop connections that represent the most probable application for shared gigabit Ethernet. Second, although shared gigabit Ethernet has plenty of bandwidth to support multimedia data streams, such as multicast video, the quality of service needs to be carefully evaluated to see if it is adequate for these applications. The effects of using the enhanced CSMA/CD algorithm known as BLAM should also be investigated. Third, since the major attraction of gigabit CSMA/CD is low cost, we need to compare its performance with other low-cost options, such as shared gigabit Ethernet using full-duplex repeaters and switched 100BASE-T.

In the final analysis, the success of gigabit CSMA/CD depends critically on the availability of 1000BASE-T. At this point, we expect the complexity of a 1000BASE-T transceiver to be comparable to an Intel 80486 CPU, and that about two years from now they will start to become available as commodity parts. Once these parts become cheap enough, gigabit Ethernet to the desktop will become a high-volume, cost-competitive product, and users will need to carefully evaluate the cost-benefit tradeoffs between switched 100BASE-T and shared 1000BASE-T, and between half-duplex and full-duplex repeaters.

References

[1] IEEE Draft P802.3z/D2, "Media Access Control (MAC) Parameters, Physical Layer, Repeater and Management Parameters for 1000 Mb/s Operation," Feb. 1997.

[2] IEEE Std. 802.3u-1995, "Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T," 1995.

[3] ANSI/IEEE Std 802.3, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," 5th ed., 1996.

[4] M. Molle and G. Watson, "100BASE-T/IEEE 802.12/Packet Switching," *IEEE Commun. Mag.*, vol. 34, no. 8, Aug. 1996, pp.64–73.

[5] ANSI/TIA/EIA-568-A-1995, "Commercial Building Telecommunications Cabling Standard."

[6] H. Frazier, Jr., "Review and Update of Carrier Extension Proposal," IEEE 802.3z plenary mtg., Vancouver, BC, Canada, Nov. 1996; ftp://stdsbbs.ieee.org/pub/802_main/803.3/gigabit/presentations/nov1996/Hfcarext.pdf.

[7] J. Spragins, *et al.*, *Telecommunications Protocols and Design*, Reading, MA: Addison-Wesley, 1991.

[8] M. Molle, "Reducing the Effects of Propagation Delay on CSMA/CD Networks," IEEE 802.3 High-Speed Study Group plenary meeting, San Diego, CA, Mar. 1996; ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/mar1996/MMredpd.txt.

[9] M. Weizman, "HSSG CSMA/VCD Proposal," IEEE 802.3 High-Speed Study Group plenary meeting, Enschede, The Netherlands, July 1996, ftp://

stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/july1996/MWvcdprp.txt
[10] M. Kalkunte and J. Kadambi, "Packet Packing and mTBEB Simulation Results," IEEE802.3 High-Speed Study Group plenary meeting, Enschede, The Netherlands, July 1996, ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/july1996/MKsim.pdf.
[11] S. Haddock, "Carrier Extension Issues," IEEE 802.3 High-Speed Study Group plenary meeting, Enschede, The Netherlands, July 1996; ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/july1996/SHcarext.txt
[12] M. Molle *et al.*, "Packet Bursting," IEEE 802.3z plenary meeting, Vancouver, BC, Canada, Nov. 1996; ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/nov1996/MKpk_burst.pdf.
[13] M. Molle, "A New Binary Logarithmic Arbitration Method for Ethernet," Tech. Rep. CSRI-398, Univ. of Toronto, Apr. 1994 (revised July 1994).
[14] IEEE Draft 802.3w/D1.8, "Enhanced Media Access Control Algorithm for IEEE 802.3 Ethernet," Feb. 1997.
[15] OPNET Modeler - MIL3 Co., Washington DC
[16] H. Frazier, Jr., "Scaling CSMA/CD to 1000 Mb/s: An Update," IEEE802.3 High-Speed Study Group plenary meeting, Enschede, The Netherlands, July 1996; ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/july1996/MKsim.pdf.
[17] K. Ramakrishnan and H. Yang, "The Ethernet Capture Effect: Analysis and a Solution," *Proc. 19th IEEE Local Comp. Networks Conf.*, Minneapolis, MN, Oct. 1994, pp. 228–40.
[18] D. Boggs, J. Mogul, and C. Kent, "Measured Capacity of an Ethernet: Myths and Reality," *Proc. ACM SIGCOMM '88*, Aug. 1988, pp. 222-34.
[19] A. Erramilli, O. Narayan, and W. Willinger, "Experimental Queueing Analysis with Long-Range Dependent Packet Traffic," *IEEE/ACM Trans. Networking*, vol. 4, no. 2, Apr. 1996, pp. 209–23.
[20] L. Kleinrock, *Queueing Systems*, vol. I, Wiley-Interscience, 1975.

Biographies

MART MOLLE (mart@cs.ucr.edu) is a professor of computer science and engineering at the University of California, Riverside. His research interests include the performance evaluation of protocols for computer networks and of distributed systems. Mart also chairs the IEEE 802.3w task force, which is developing a standard for BLAM as an optional enhancement to CSMA/CD. Mart received the M.S. and Ph.D. degrees in computer science from UCLA in 1978 and 1981, respectively.

MOHAN KALKUNTE is a member of technical staff (MTS) in the Network Products Division at Advanced Micro Devices (AMD), Sunnyvale, CA. His interests are in the area of performance modeling/analysis of LANs. Prior to his current job, Mohan was an MTS at AT&T Bell Laboratories, specializing in performance/reliability modeling of signaling systems and voice-related services over circuit-switched networks. He received his Ph.D. degree in industrial and systems engineering from Ohio State University in 1988.

JAYANT KADAMBI is a manager in the Network Products Division at AMD. His interests are in the area of high-speed LAN network architecture. Prior to his current job, Jayant was an MTS at AT&T Bell Laboratories, specializing in LAN network architecture and design. Jayant received his M.S. degree in electrical engineering from Renssalaer Polytechnic Institute in 1986.