Scaling CSMA/CD to 1Gb/s with Frame Bursting

Mart Molle
University of Cal. Riverside
mart@cs.ucr.edu

Mohan Kalkunte
Advanced Micro Devices
mohan.kalkunte@amd.com

Jayant Kadambi
Advanced Micro Devices,
jayant.kadambi@amd.com

Abstract

In Gigabit Ethernet, the round-trip propagation delay can be much greater than the transmission time for a minimum length frame. In this paper, we describe some changes to the Ethernet CSMA/CD medium access control algorithm that allow CSMA/CD to be used in this case. First, carrier extension is used to increase the slot time without requiring a corresponding increase to the minimum frame length. Second, frame bursting is introduced so that a host may transmit more than one frame without releasing control of the channel, in a manner that increases the efficiency for small frames without changing its one-frame-at-a-time service interface. Using simulation, we show that CSMA/CD with carrier extension and frame bursting operating on 1 Gb/s links provides a significant performance increase over 100 Mb/s Fast Ethernet. These changes are being adopted by the IEEE 802.3z task force, which is currently defining the standard for Gigabit Ethernet.

1. Introduction

Under the current IEEE 802.3 standard, a host can be connected to the network by either a full-duplex point-to-point link leading directly to a dedicated port on a switch, bridge, router, or another host, or by a half-duplex link leading to a "collision domain" that is shared among several hosts. Shared Ethernet uses the well-known Medium Access Control (MAC) algorithm called Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [1].

Although switched full-duplex networks can support much higher total throughput than the equivalent shared CSMA/CD system, full-duplex connections are also more costly than half-duplex connections because switches are more complex devices than repeaters. Thus, Gigabit CSMA/CD may be more cost effective 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 the network. CSMA/CD also avoids the flow control problem in switched networks. And, finally, shared CSMA/CD networks can make use of physical layer technologies that are only capable of half-duplex operation, such as the coaxial cable used in 10BASE-5 and 10BASE-2 and the four-pair signaling used in 100BASE-T4. Thus, CSMA/CD may have some applications at very high data rates, and the IEEE 802.3z task force is defining a standard for 1Gb/s CSMA/CD [3].

CSMA/CD is a distributed algorithm that allows several active hosts to serialize their transmissions on the same 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 network activity. Once the frame transmission begins, the host continues to look for other traffic on the network using *collision detection*, in which case it abandons this attempt and schedules 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. The slot time is also used as the discrete delay quantum in the backoff algorithm as well as the minimum frame length. Restricting the backoff delay to integral multiples of the slot time leads to a topology-independent fairness property where, in the absence of other activity, two colliding hosts won't collide with each other again if they choose different backoff delays, independent of the collision event timing and their relative positions in the network.

Ensuring that the duration of each successful frame transmission is at least one slot time is important for both the sender and receiver in CSMA/CD. The sender can use the absence of a collision during transmission as an unreliable acknowledgment, since sender would have detected a collision, if there were one, within one slot time. Conversely it allows receivers to filter out incoming collision fragments using a length threshold.

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 by an increase in the slot time. When the IEEE 802.3u standard [2] raised the

0742-1303/97 $10.00 © 1997 IEEE

Ethernet 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. The resulting reduction in network diameter to 205 meters was deemed to be an acceptable compromise because current wiring standards [4] specify that the distance from an office to the nearest wiring closet should be less than 100 meters. Obviously, however, there is no room for a further distance reduction as the Ethernet data rate is raised from 100 Mb/s to 1 Gb/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 1 Gb/s networks.

2. CSMA/CD Extensions to Handle a Larger Slot Time

Simply increasing the CSMA/CD slot time parameter for 1 Gb/s operation would be unacceptable because it serves several functions, and some of them cannot be changed without compromising the utility of the standard. In particular, the most common application for Gigabit Ethernet is likely to be a backbone network for interconnecting various 100 Mb/s switches. Thus, an increase in the minimum frame size for transmission over the 1 Gb/s backbone would lead to longer frames, and hence greater congestion, on the network periphery. Therefore, IEEE 802.3z has adopted a technique known as *carrier extension* to decouple the minimum frame length from the slot time for half-duplex 1 Gb/s operation [5].

Under carrier extension, the minimum frame size remains 512 *bits* (as it is for 10 Mb/s and 100 Mb/s networks), but the minimum length of any transmission over the network is increased to 512 *bytes* in the following way. If the bit 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, then the bit transmitter returns a *transmit done* status code to the MAC layer. However, if the length is less than one slot time, then the bit transmitter withholds the status code and continues transmitting a sequence of a special *extended carrier* symbols until the end of the slot time, at which time it returns the *transmit done* status code to the MAC layer.

Should the MAC layer detect a collision at any time during this process, it truncates the remainder of the outgoing frame transmission (or extended carrier), and waits for the bit transmitter to finish sending the jam signal. Meanwhile, the bit receiver processes at the other hosts are 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, then the incoming frame is discarded as a collision fragment.[1] Otherwise, the receive buffer is passed to the MAC layer for checksum and address verification.

Carrier extension represents a very minor change to the existing CSMA/CD algorithm, and it solves the problem of increasing 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 to 1 Gb/s for a host that generates only minimum-length (64 byte) frames would allow it to send bits ten times faster than before, while requiring it to send eight times as many bits per frame—resulting in only a 25% 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 improvements if we can find a way to add *pipelining* to the frame transmission process in CSMA/CD.

Pipelining is widely used in automatic repeat request (ARQ) algorithms at the data-link layer [6, §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 [7, 8], only the most basic approach, known as *packet packing* [9, 10], received serious consideration by the Gigabit Ethernet group. The idea in packet packing was to boost efficiency by allowing a transmitter to send additional frames, separated by small amounts of extended carrier, while waiting for the slot time to end. However, if the host runs out of data during this time, the remainder of the slot time is simply filled with extension bits. In this case, the sender might transmit several short frames without knowing if a collision was taking place. If a collision is *not* detected before the end of the slot time, however, then the host can assume it has acquired the network and just send the remainder of its sequence. Otherwise, it must assume that *the entire sequence* has been destroyed by a collision, and retransmit them all after waiting for a random backoff delay.

On a busy network, packet packing can recover essentially all of the efficiency lost because of carrier extension, assuming the time between consecutive frames packed in the same burst is the same as the normal interframe spacing. Unfortunately, however, this proposal also requires substantial

[1] The MAC layer must not duplicate frames, so the receiver discards the incoming fragment—even if the collision took place during the extended carrier beyond the frame—since the transmitter may wish to retransmit the frame.

changes to the service interface between the CSMA/CD MAC layer and its client. On the transmit side, the MAC layer can no longer return a status code to its client for one frame before requesting the next. (Indeed, the MAC layer may have as many as eight unacknowledged minimum-length frames under its control at any given time.) Similarly, on the receive side, the MAC layer must "quarantine" the incoming frames, 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 burst. 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 standard. However, the inefficiencies of carrier extension remained, so work continued on finding a compromise that would 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* [11], 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 is based on the maximum frame size instead of the slot time, like the Binary Logarithmic Arbitration Method (BLAM) [12, 13]. 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 [8]. This feature ensures that collisions can only affect the first frame of any burst, so that both the sender and receiver can retain their familiar one-frame-at-a time service interfaces.

More precisely, frame transmissions under CSMA/CD with frame bursting work in the following way:

1. To send the first (or only) frame in a new burst, the transmitter follows the normal rules for CSMA/CD—deferring to other activity, backing off after collisions, etc.—except that at the start of each attempt it sets a flag to indicate that this is the *first frame* in a burst and starts a *burst timer.*
2. If the attempt fails because of a collision, the transmitter clears its *first frame* flag and *burst timer,* and returns to step 1. Otherwise, it sends extended carrier until the *burst timer* reaches one slot time, if necessary, then returns the *transmitOK* status code to its client for this frame, clears the *first frame* flag and goes on to step 3.
3. At this point, the transmitter still has control of the channel and must decide whether or not to extend its burst. Thus, if the *burst timer* indicates that there is no more time available to initiate new transmissions in this burst, the burst is over and the next attempt will begin at step 1. Otherwise, it sends 96 bits of extended carrier (which serves as the interframe space within the burst) and goes on to step 4.
4. If no more frames are available, then the burst is over even though more time was available, so the transmitter clears its *burst timer* and waits for the next frame (which will begin at step 1). Otherwise, it starts sending the new frame immediately, without creating a gap in the outgoing bit stream, and returns to step 3 when it is finished.

An example of the sequence of items in a frame burst is shown below in Figure 1. Notice that extended carrier symbols are used for both the extension part of the first frame and the interframe spacing between consecutive frames in the burst, and that the extension part of the first frame is not used as interframe spacing. 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.

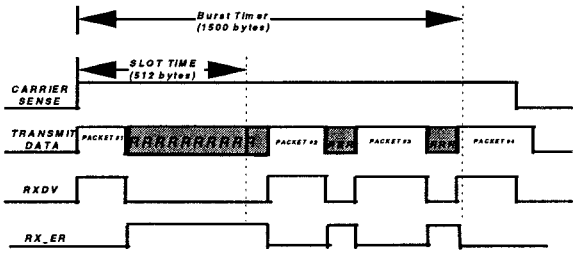

Figure 1. Frame burst sequence

Notice that the condition for including another frame into an ongoing burst is based on two tests. First, the starting time for the next frame must occur before the *burst timer* reaches a certain cutoff value. Second, the next frame must be available for transmission by the end of the interframe gap period. Initially, the burst limit was set at 12,000 bit times, which was deliberately chosen to be just below the maximum frame length to promote fairness. Thus, no matter what mix of frame lengths it has, a host could keep extending its burst until it had transmitted for at least the equivalent of one maximum-length frame, and for no more longer than (roughly) twice the maximum frame size in the worst case. However, there was considerable discussion about using a larger value for the burst limit to improve efficiency, and it was later changed to 8K bytes (i.e., 65,536 bit times).

The maximum burst length is defined by a starting time threshold rather than a finishing time threshold because the logic is simpler, and because some implementations may not even know the length of an outgoing frame at the moment they start its

transmission. And we set the threshold just below the maximum frame length to avoid favoring hosts doing bulk data transfer, where consecutive maximum-size frames are a common case.

Frame receptions under CSMA/CD with frame bursting use 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 any incoming data until it has found a valid preamble. At that time, it sets an *extending* flag to indicate this will be the first frame in a burst, and hence is subject to the carrier extension rule. Thereafter, it counts the incoming bits from this frame (gathering those bits that are not extended carrier to form the incoming frame) until it finds the end of the frame, as indicated by *either*: a) end-of-carrier, *or* b) the appearance of an extension bit after the end of the slot time has been reached, and goes on to step 2.
2. The receiver now decides whether or not to pass the incoming frame to its client by checking that its length is at least equal to the minimum frame length and, if the *extending* flag is still set, that the total number of incoming bits including any extension bits is at least equal to the slot time. In addition, it makes sure that the *extending* flag is off, and goes on to step 3.
3. If the frame ended because of the first condition, or if we reach end-of-carrier instead of another preamble and start-frame delimiter as we look at more of the incoming bit stream, this burst is over and the receiver returns to step 1 in preparation for the next burst. Otherwise, the receiver gathers incoming bits to form the next incoming frame until it is terminated by *either*: a) end-of-carrier, *or* b) the first extension bit, and returns to step 2 with *extending* off.

Notice the significance of the *extending* flag in this modified algorithm. First, it is used in an obvious way to ensure that the extended transmission time requirement is only applied to the first frame in each burst. In addition, its value is used to change the meaning of an extension bit into one of two "meta-symbols," i.e., if *extending* is on, then extension bits are treated like data bits, whereas if *extending* is off they are treated like idle time.

3. Modeling Approach

A simulation model was developed in OPNET [14] 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 modeled. The frame generation process at each host is independent of the current state of the network (in particular, there is no "flow control" to throttle the arrivals if the network is already busy), and there is no upper bound on the size of the transmit queue at each host.

Each data point in the figures was obtained by running the program for 30 seconds of simulated time. Depending on the network load, more than 6 million frames transmissions may be simulated in a single run.

4. Performance

4.1. Maximum Throughput

Depending on the traffic mix, frame bursting may provide a significant performance increase. Though increasing the burst length does provide a modest increase in performance, if all frames are at least one slot time in length, these improvements will be minor. At the opposite extreme, with short frames they can lead to dramatic differences in the maximum throughput on a heavily loaded network. To see this, let us compare the respective worst-case efficiencies for:

- ordinary 100 Mb/s CSMA/CD (512 bit slot time and no carrier extension);
- baseline 1 Gb/s CSMA/CD (4096 bit slot time with carrier extension only); and
- 1 Gb/s CSMA/CD with frame bursting (4096 bit slot time, also including carrier extension),

when a single busy source is attempting to transmit large numbers of frames over an otherwise-quiet network.[2]

In general, the loss of efficiency for each method due to framing overhead can easily be calculated by finding the ratio of the number useful bits sent by a single active host per "cycle" divided by 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 cycle lasts for $(96+64+P)$ bit times, where we have added the inter-frame spacing, preamble and start-frame delimiter. Since this ratio is clearly an increasing function of the frame length, P, the worst-case overhead occurs at the minimum frame size, where P=512 and the normal-

[2] This situation is not as unrealistic as it appears because of the dynamics of the capture effect in heavily loaded CSMA/CD networks, as described below.

ized efficiency is 76%. If we now increase the speed to 1 Gb/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%.

The worst-case normalized efficiency for packet packing is the same as for ordinary 100 Mb/s CSMA/CD, i.e., 76%, since there is no extra overhead under high load. The worst-case normalized efficiency calculation for 1 Gb/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. The efficiency is worst with minimum length frames, and is given by:

$$(n*512)/(4096 +(n-1)*672)$$

in which n is the number of minimum sized frames in the burst. The worst case normalized efficiency occurs when the burst timer does not expire until after we have started $\lceil(12{,}000–4096)/672\rceil$=12 extra frames, therefore efficiency for 13 minimum-length frames per cycle (n=13) is,

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

This is more than 70% of the worst-case normalized efficiency for ordinary CSMA/CD, and more than 4.5 times higher than the baseline proposal for 1 Gb/s CSMA/CD with carrier extension. If we increase the burst limit to 8K bytes, then the burst timer does not expire until after we have started 92 additional minimum-length frames, giving a worst-case efficiency of

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

which is within 5% of the 100 Mb/s CSMA/CD result.

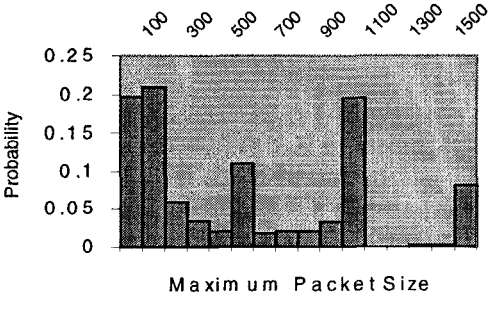

Figure 2. Workgroup packet size distribution

Since the efficiency calculations given above do not account for the effects of collisions and do not reflect a realistic mix of frame sizes, we constructed event-driven simulation models of the three options, which are briefly described in section 3. Each model was run with a simple empirical traffic model, known as the *workgroup average distribution* [5, p.21], which is also shown in Figure 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 the spring of 1996.

It is interesting to note that the average frame length in the workgroup average distribution is about 600 bytes, but the average rises to about 750 bytes when we expand the short frames using carrier extension. As we will see below, frame bursting improves performance on heavily loaded networks enough to eliminate the effects of this 25% traffic increase due to carrier extension overhead. That is, under high network load, all of the performance measures shown below using frame bursting are as good or better than the results without frame bursting at 600/750 of the network load, i.e., a 20% penalty in the load.

By using this workgroup average distribution to select the frame lengths in our study, the traffic will be a mixture of frame sizes, most of which are either quite short (i.e., a control message or perhaps a few keystrokes of interactive data) or quite long (i.e., large data segments for some popular protocols). Similar bimodal frame size distributions have also been reported in other network measurement studies. However, we did not attempt to reproduce all the temporal features of real network traffic in our model, such as the burstiness of the arrivals and the correlation between the 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 size distribution.

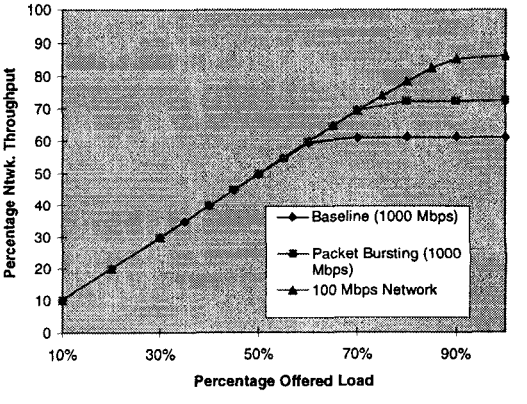

Figure 3. Network throughput

Figure 3 shows percentage throughput as a function of percentage offered load for a 15-host network with the workgroup average traffic model for the three alternative versions of CSMA/CD described above, namely an ordinary 100 Mb/s system, a baseline 1 Gb/s system using only carrier extension, and a 1 Gb/s system using frame bursting. The same experiments were also run on a 4-host network, but the results are qualitatively similar and are not 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 rises in lock-step with the load to form a straight 45° line until we approach its maximum, where the throughput is limited by the efficiency of the protocol, the queues saturate and the curve turns horizontal. The two parts of each curve do not meet as a sharp corner because some frames are dropped because of excessive collision errors. However the corner is visibly more sharp for the 1 Gb/s curves because the longer backoff delays (relative to the frame transmission time) reduce the proportion of frames that experience excessive collisions. From this Figure, the performance penalty from increasing the slot time for 1 Gb/s operation using carrier extension is roughly a 30% reduction in the maximum percentage throughput with the workgroup average frame size distribution (i.e., from about 86% to 61% for the 15-host system). However, adding frame bursting lets us recover almost half of this loss, raising the maximum throughput to about 72%, i.e., close to the 25% reduction in overhead we expected. The corresponding throughput values for a 4-host system were about 4% higher in each case.

While packet packing was deemed to be too complicated from an implementation perspective, we show its throughput performance in comparison with baseline carrier extension and frame bursting with burst limits of 1500 bytes and 8K bytes, respectively. Table 1 shows the network throughput at 100% offered load for a 15-host system for fixed packet sizes of 64 bytes (minimum) and 1518 bytes (maximum), and for the workgroup average distribution. All three of the pipelined approaches provide substantial improvements over baseline carrier extension, especially for 64 byte frames. As expected, packet packing is more efficient than frame bursting with a burst limit of 1.5K bytes, offering 15% higher throughput with 64 byte frames and 5% higher throughput with the work group average distribution. However, what is more surprising is that frame bursting with a burst limit of 8K bytes is actually more efficient than packet packing. The explanation for this result is quite simple. Packet packing stops adding frames to a burst as soon as its length exceeds one slot time (4096 bits), whereas frame bursting keeps going until the burst length exceeds 65,536 bits. Because of this 16-fold increase in the burst length, collisions will be less frequent and hence less time will be wasted on contention.

Packet Size	*64*	*1518*	*Workgroup Average*
Carrier Extension	90.7	853.2	562.2
Packet Packing	339.8	853.2	740.8
1.5K Frame Bursting	286.7	853.2	700.3
8K Frame Bursting	360.7	885.0	790.2

Table 1. Comparison of Network Throughput (Mbps) at 100% offered load

4.2 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 system starts higher but extends further to the right because offered load has been normalized to a percentage, whereas delay is expressed in absolute units (i.e., msec). Once again, the results show that increasing the slot time causes a substantial loss in normalized performance. However, one must keep in mind the different scales in the x-axis—for a fixed one msec end-to-end delay, the throughput of the 100 Mb/s network is actually about five times lower instead of twice as high. Note, also, that the results with frame bursting are uniformly better than without it:

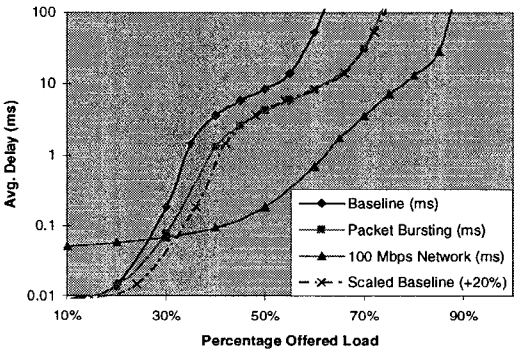

Figure 4. Average end-to-end packet delay

the mean delay drops by at least a factor of two when the offered load is over 30%, and by a factor of ten at 60% load.

The curves also demonstrate the distinctive non-convex shape that is characteristic of Ethernet systems, where the delay rises quite early as the network starts to become congested (and deferrals, collisions and backoff delays become more common), but then levels off again as we move towards higher loads and we start seeing evidence of the capture effect. In this case, once a queue of outgoing frames starts to develop at a host, its transmission efficiency increases because additional frames sent in rapid succession are far less likely to experience collisions than the first frame. The effect is less visible at 100 Mb/s because of the smaller value for the slot time, which reduces the length of a capture period.

4.3. The Effect of Capture on Access Latency

Even though the *average* access latency may be quite small, the latencies experienced by individual frames vary greatly at high loads because of the capture effect. In particular, a small percentage of the frames experience extremely large delays. 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[3] of the access latency distribution (in μsec), demonstrates how significant the burstiness of the access latency can become under heavy load. Referring back to Figure 4, it is evident that the given range for the percentage offered load falls entirely below the congestion point (just before the flat spot in the delay curve) for the 100 Mb/s CSMA/CD system. Conversely, the baseline 1 Gb/s CSMA/CD is way beyond congestion and has almost reached total saturation. As

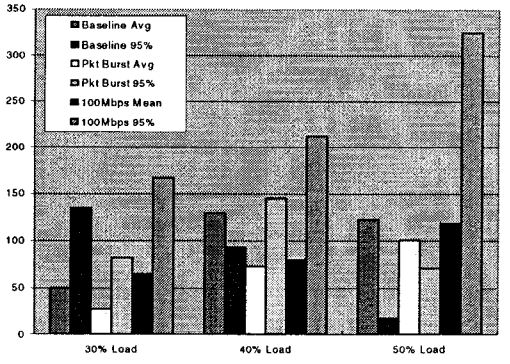

Figure 5. Mean & 95th percentile of access delay

[3] Such percentiles are used to show the variability of the delay.

expected, increasing load causes both measures of access latency to increase with increasing load in the 100 Mb/s system, and in each case the mean is slightly below half the 95th percentile. A similar relationship holds for the baseline 1 Gb/s system at 30% load, and for the 1 Gb/s system with frame bursting at both 30% and 40% load. However, the results for the baseline 1 Gb/s system at higher loads are very different, with the 95th percentile falling to *less than 15% of the mean* at the highest load! In other words, the average access latency experienced by the *worst* 5% of the frames must be 120 times higher than the average of the *remaining* 95%.

These counterintuitive results for the access latency in congested systems are caused by the capture effect [12, 15]. 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, a host on a busy CSMA/CD network will find itself alternating between long periods where it finds itself unable to transmit anything, and then suddenly is able to transmit a large burst of consecutive frames. These frames in the burst experience minimum access latency while the first frame in the burst may potentially experience large access latency. This would result in a skewed distribution for access latency, with the skewness increasing as offered load increases.

The capture effect is further demonstrated by Figure 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 hosts in the 100 Mb/s system rarely have any consecutive frame transmissions, as we expect in a system that has not yet reached its congestion point. However, the number of consecutive frame transmissions for the 1 Gb/s CSMA/CD systems grows very large as we increase the load, especially in the baseline system without frame bursting.

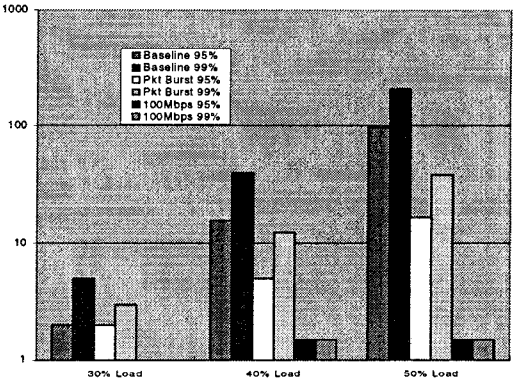

Figure 6. Percentiles of the distribution of consecutive frame transmission by a single host

The significance of these results is that they clearly show that frame bursting increases the efficiency of CSMA/CD without any negative side effects that might degrade the quality of service under heavy load. In particular, 1 Gb/s CSMA/CD with frame bursting provides the same quality of service at 50% offered load as the baseline 1 Gb/s CSMA/CD system at 40% load, in terms of end-to-end delay, access latency statistics, and even the number of consecutive frame transmissions.

4.4. 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:

1. Using carrier sensing, the host is required to *defer* the transmission of a frame because of some earlier network activity.
2. 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.
3. After suffering 16 collisions, the host is required to drop frame, in which case an *excessive collision error* is reported.

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
Thus, the deference likelihood should be no less than the percentage offered load, as is evident from Figure 7 for the case of a 100 Mb/s network under light traffic. The light traffic deference likelihood for Gb/s networks are about 25% 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 25% higher offered load.

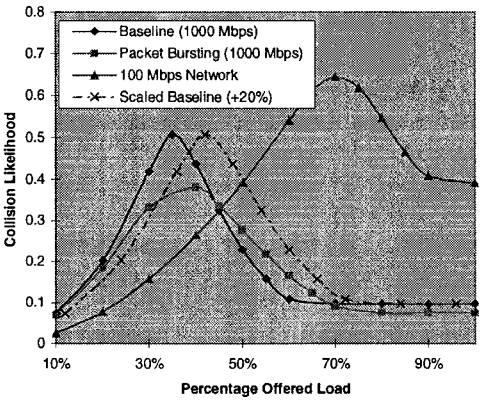

Figure 8. Probability that a frame experiences more than one collision

The probability that a transmitted frame experiences at least one collision is shown in Figure 8. The results shown are typical of most shared-media Ethernet systems, where the collision likelihood peaks

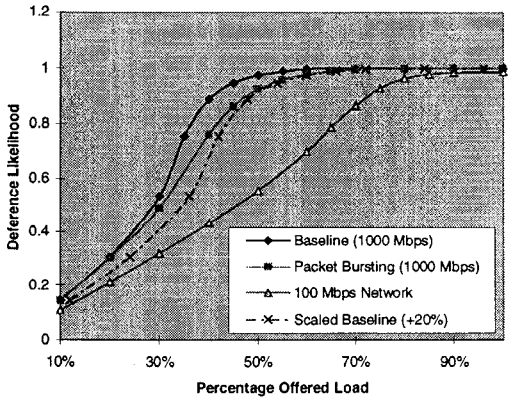

Figure 7. Probability of a transmission delay due to deferral

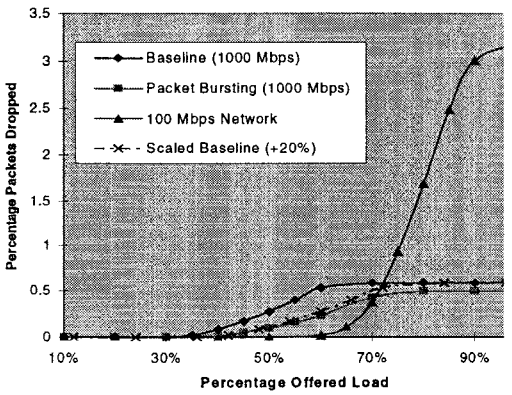

Figure 9. Percentage of frames with an excessive collision error

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 of the capture effect. The peak of the collision likelihood curves is much lower at 1 Gb/s than at 100 Mb/s because of the increase in the 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 1 Gb/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 Figure 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 25% higher offered load.

5. Conclusions

In this paper, we have shown that with only a few minor changes, CSMA/CD can operate effectively at 1Gbs/s data rates. First, carrier extension decouples the minimum frame size from the slot time, to accommodate the inevitable increase in the bandwidth-delay product without changing the Ethernet frame size. 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 reasonable maximum throughput, even at 1Gb/s data rates. More importantly, however, we found that this feature does no harm to any of the other measures of performance. In other words, frame bursting improved every measurement that we made (including end-to-end delay, access latency, and the likelihood that a frame defers to other activity, experiences collisions, or gets dropped because of excessive collisions), and provides equivalent performance even when handicapped by a 25% increase in the offered load.

Perhaps the most surprising conclusion from our study is that frame bursting actually *reduces* the impact of the capture effect; as the network becomes congested, frame bursting delays the onset of the capture effect by allowing each host to drain its queue more quickly.

References

1. ANSI/IEEE Std 802.3, "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", Fifth Edition, 1996.
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. IEEE Draft P802.3z/D2, "Media access control (MAC) parameters, physical layer, repeater and management parameters for 1000 Mb/s operation", February 1997.
4. ANSI/TIA/EIA-568-A-1995, Commercial Building Telecommunications Cabling Standard.
5. H. Frazier, Jr., "Review and update of carrier extension proposal", IEEE 802.3z plenary meeting, Vancouver BC, Nov. 1996, ftp://stdsbbs.ieee.org/pub/-802_main/803.3/gigabit/presentations/nov1996/-Hfcarext.pdf
6. J. Spragins, et al., *Telecommunications Protocols and Design*, Addision-Wesley, 1991.
7. 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
8. M. Weizman, "HSSG CSMA/VCD proposal", IEEE 802.3 High-Speed Study Group plenary meeting, Enschede NL, July 1996, ftp://stdsbbs.ieee.org/pub/-802_main/802.3/gigabit/presentations/july1996/MWvcdprp.txt
9. M. Kalkunte and J. Kadambi, "Packet Packing and mTBEB Simulation Results", IEEE802.3 High-Speed Study Group plenary meeting, Enschede, NL, July 1996, ftp://stdsbbs.ieee.org/pub/802_main/802.3/gigabit/presentations/july1996/MKsim.pdf
10. S. Haddock, "Carrier extension issues", IEEE 802.3 High-Speed Study Group plenary meeting, Enschede NL, July 1996, ftp://stdsbbs.ieee.org/pub/802_main/-802.3/gigabit/presentations/july1996/SHcarext.txt
11. M. Molle, J. Kadambi, M. Kalkunte and H. Frazier, Jr., "Packet bursting", IEEE 802.3z plenary meeting, Vancouver BC, Nov. 1996, ftp://stdsbbs.ieee.org/-pub/802_main/802.3/gigabit/presentations/nov1996/-MKpk_burst.pdf
12. M. Molle, "A new binary logarithmic arbitration method for Ethernet", Technical Report CSRI-398, University of Toronto, April 1994 (Revised July 1994).
13. IEEE Draft 802.3w/D1.8, "Enhanced media access control algorithm for IEEE 802.3 Ethernet", February 1997
14. OPNET Modeler – MIL3 Co., Washington D.C.
15. K. Ramakrishnan and H. Yang, "The Ethernet capture effect: analysis and a solution", *19th IEEE Local Computer Networks Conference.*, Minneapolis MN, Oct. 1994, pp. 228-240.
16. A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 2 (April 1996), pp. 209-223.
17. L. Kleinrock, *Queueing Systems, Vol. I*, Wiley-Interscience, 1975.