Reducing the Effects of Large Propagation Delays on High Speed IEEE 802.3 CSMA/CD Networks using Collision Truncation

Mart L. Molle

Computer Science Department
University of California
Riverside, CA 92521

Boaz Yeger

Computer Science Department
University of California
Riverside, CA 92521

Abstract

A half-duplex repeater is the least-cost method for connecting multiple hosts to an IEEE 802.3 Ethernet LAN. However, if we increase the data rate from 10 Mb/s Ethernet to 100 Mb/s Fast Ethernet, shared access via CSMA/CD becomes less efficient because of the resulting increase in size of the collision fragments. For example, in a typical star-wired single-repeater network with a diameter of 200 meters, upgrading from 10BASE-T to 100BASE-T increases the round-trip delay (including bit budget for the network cards and repeater) from about 100 bit times to almost 500 bit times. Moreover, a further upgrade to 1000 Mb/s Gigabit Ethernet would increase the round-trip delay to about 4000 bit times—which is eight times larger than the minimum frame size. In this paper, we introduce a simple change to the handling of incoming signals at the repeater ports known as **collision truncation**, *and show that it reduces the average duration of a collision by more than 50 percent for networks with large round-trip delays. For the standard BEB Ethernet protocol operating at moderate load, collision truncation reduces the mean end-to-end network delay by more than 25% for 100 Mb/s Fast Ethernet and by more than 50% for 1000 Mb/s Gigabit Ethernet. For BLAM, collision truncation increases the maximum throughput by more than 10%, and reduces the mean end-to-end delay substantially under high load. Collision truncation requires no changes to the host network adapters, and its use is completely transparent to the hosts. The only behavioral change that is visible to the attached hosts is the illusion that the network diameter appears to have been reduced. The relative cost of adding collision truncation to a repeater is very small compared to upgrading to a frame-level device, such as a bridge.*

1 Introduction

All recent versions of Ethernet—starting from 10BASE-T [1], and including the 100 Mb/s Fast Ethernet systems defined in the IEEE 802.3u standard [2] as well as the proposed 1000 Mb/s Gigabit Ethernet systems in the IEEE 802.3z draft standard [4]—use a hierarchical star-wired architecture that consists of a collection of host interfaces and repeaters connected by point-to-point link segments. That is, each link segment consists of a dedicated point-to-point communications channel, which connects one active electronic device (i.e., a host computer interface or a repeater port) to another. Thus, in these designs, the shared medium is created entirely by the digital logic in the core of the repeaters.

Despite the change in architecture from passive buses to star wiring, the operating rules for a repeater have not unchanged. However, star-wired Ethernet architectures offer a new opportunity for controlling collisions, since the colliding signals are now combined inside a repeater under the control of some digital logic. In this paper, we show that some small changes to the repeater logic can reduce the time wasted on each collision significantly on networks for which the bandwidth-delay product is large. Since the time spent on collisions is unavailable for delivering valid data, the proposed changes will increase the efficiency of CSMA/CD networks and reduce the message delivery times for their users. This reduction in collision time is particularly important for networks using the Binary Logarithmic Arbitration Method (BLAM) [5], which is an enhanced version of the Ethernet CSMA/CD algorithm now under consideration as an option to the Binary Exponential Backoff (BEB) algorithm as specified in the IEEE 802.3 standard [3].

The rest of the paper is organized as follows. In section 2, we summarize the basic operating principles for a star-wired network with a standard IEEE

802.3 repeater. In section 3, we introduce the concept of collision truncation. and we show that it is compatible with the CSMA/CD algorithm. That is, the hosts in the network would not conclude that anything was wrong with the network, even if they performed a careful timing analysis of events, except that transmitters appeared to be closer together than they really are. Section 4 presents the results of simulation experiments for both 100 Mb/s Fast Ethernet and 1000 Mb/s Gigabit Ethernet systems, to demonstrate the benefits of collision truncation on network performance. And, finally, section 5 gives our conclusions.

2 Collision Handling in Ethernet

2.1 Role of the MAC Layer Host Interface

Under the IEEE 802.3 standard, collision handling is done entirely by the Medium Access Control (MAC) layer protocol in each of the attached host's network interfaces. The repeaters have no role other than to copy bits blindly from each port to all others. Thus, a host wishing to transmit a frame first waits until the network is silent, then waits for a fixed 96 bit-time interframe gap, and finally starts transmitting. During the attempt, the host is simultaneously checking for the presence of other traffic on the network, which would indicate that a collision is taking place. If it detects a collision, the host:

1. finishes transmission of the first 64 bits of the frame (known as the preamble and start-frame delimiter), if it has not already done so,
2. abandons the remainder of this frame transmission attempt, and
3. transmits a 32-bit jamming sequence.

Having finished with this attempt, the host now pauses for a randomly generated backoff delay before trying again. However, even if it chooses a zero backoff delay, it cannot initiate another attempt until other collision fragments stop arriving from the rest of the network and the medium is silent once more.

Examples of possible event sequences in a two-host collision are shown in Figures 1 and 2. The reader is referred to [8] for a detailed explanation of how such space-time diagrams are used for reasoning about the performance of CSMA-type protocols. In Figures 1 and 2, we assume that the difference in starting times between the two hosts is as small and as large as possible, respectively. The Figures hide numerous inessential details about how the round-trip bit budget is allocated amongst the various components of the network. The key point is simply that the maximum network size is limited by an important parameter of the CSMA/CD algorithm called the *slot time*, which is expressed in "bit-times" and is related to the maximum round-trip delay $2D$ by the following set of constraints:

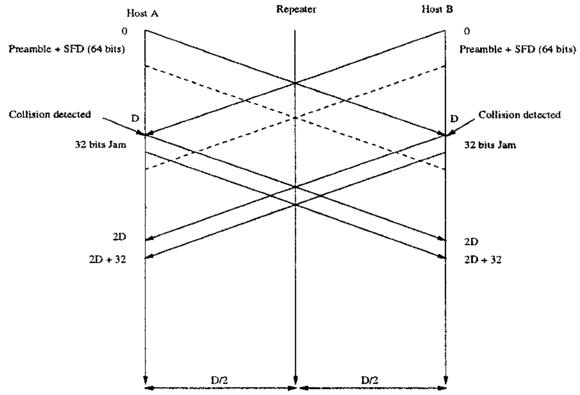

Figure 1: Space-time diagram of a collision with minimum starting time offset

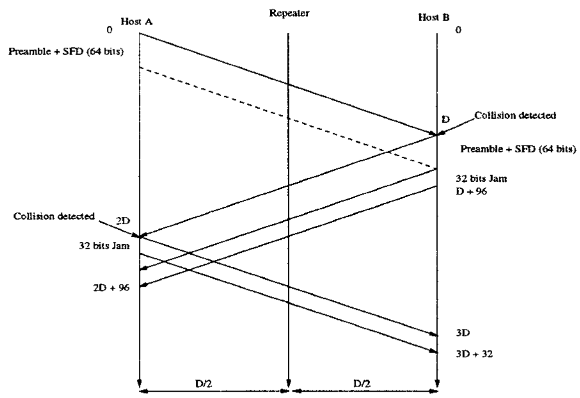

Figure 2: Space-time diagram of a collision with maximum starting time offset

1. the minimum frame size must be at least one slot time in length,
2. after stripping off the 64-bit preamble and start-frame delimiter, the maximum length of all collision fragments received by any host must be shorter than a slot time,
3. each of the involved transmitters must detect the occurrence of a collision before they have transmitted for a slot time, and
4. hosts that collide once should not collide again if they choose different values for their respective backoff delays, which are integer multiples of a slot time.

The slot time is defined as 512 bit-times for both 10 Mb/s and 100 Mb/s operation [1][2], and as 4096 bit-times for 1000 Mb/s operation [4].

It can be shown that if two hosts, A and B, both begin transmitting at time 0 (Figure 1), then both will sense activity on the network from time 0 until time $2D + 32$. However, as host B delays the start of its transmission, the duration of host A's transmission time keeps getting longer while host B's transmission keeps getting shorter until we reach the limit where B is about to defer to A's transmission, as shown in Figure 2. In this case, host A senses activity on the network from time 0 until time $2D + 96$, whereas host B senses activity for a duration of $2D + 32$ bit-times, during the interval from time D until time $3D + 32$. Note that no other host located anywhere else in the network will see a longer collision fragment than these hosts. The start and end of each fragment travels away from its source at a constant velocity, to create a convex "V" shaped busy region in space-time. Since the boundary of the busy region on the start-of-carrier side is defined by the *union* of the start-of-carrier events created by each fragment, whereas the boundary of the busy region on the end-of-carrier side is defined by the *intersection* of the end-of-carrier events created by each fragment, it is clear that the duration of combined collision event created by all transmitters must reach a local maximum at the position of each transmitter.

The strongest constraint occurs when we apply the second condition to an observer located next to host A in Figure 2, from which we can determine that $2D+32$ must be less than one slot time in the worst case. Thus, the duration of the collision event as seen by each host in Figure 1 can as much as 511 bit-times for 10 Mb/s and 100 Mb/s operation, and as much as 4095 bit-times for 1000 Mb/s operation. Similarly, for Figure 2 the duration of the collision events as seen by hosts A and B can be as much as 575 bit-times and 511 bit times, respectively, for 10 Mb/s and 100 Mb/s operation, and as much as 4159 bit-times and 4095 bit-times, respectively, for 1000 Mb/s operation. The durations of these collision events is significant compared to the size of a minimum length Ethernet frame, namely 512 bits, especially for Gigabit Ethernet.

2.2 Role of a Conventional Repeater

Half-duplex collision domains that contain more than two hosts in a 100 Mb/s Fast Ethernet or 1000 Mb/s Gigabit Ethernet system cannot be created without using a repeater. However, a repeater was not explicitly shown in either of the above Figures. This is because the role of an N-port repeater, as specified in the IEEE 802.3 standard, is essentially just to emulate a direct physical layer signaling path between the different media segments connected to each port. In other words, as far as the protocol dynamics is concerned, a repeater is characterized by its port-to-port latency and hence can be equated to an equivalent length of cabling. Thus, whenever a signal is received at any of its ports, the repeater state machine ensures that the signal is copied to each of the other $N - 1$ ports. (The repeater is also responsible for regenerating any preamble bits lost in the receiver, but this detail is not relevant to our current discussion.) Similarly, throughout the duration of the collision state (i.e., an interval of time during which there are signals arriving at more than one port), the repeater state machine simply instructs all ports to transmit a jam signal. In other words, the repeater has no knowledge of the CSMA/CD medium access control protocol, and plays no part in its collision handling. And, more importantly, the exact time at which the beginning or end of a specific transmission by host A reaches host B depends only on the actions of host A and on the propagation delay along the path from A to B.

3 Collision Truncation

3.1 Approach

The key ideas behind collision truncation are the following. First, it is only the *leading edge* of a collision fragment that triggers a response from the CSMA/CD algorithm: once a transmitter detects the start of a collision, the entire sequence of collision recovery steps follows a fixed timetable and thereafter the host just waits for silence. Second, since the *receivers* in an IEEE 802.3 CSMA/CD network can easily anticipate the future actions of the *transmitters* during a collision event, they can apply some very simple logic to "accelerate" the termination of the collision. Consider an observer, O, situated at one end of a point-to-point link segment (i.e., adjacent to some repeater port or host interface). Whenever O sees signals traveling in both directions over the segment, O knows that a collision is taking place—just like the "adjacent device." Furthermore, since the most general topology for a "star-wired" Ethernet is an undirected tree, O also knows that at least one of the transmitters, say X, must be part of the subtree rooted in the "adjacent device" next to O and at least one of the transmitters, say Y, must be part of the subtree rooted in the "remote device" at the end of the link segment away from O.

Because the path from X to Y crosses the link segment where O is watching the network, O finds out about every transmission by X before Y does. Similarly, O finds out about every transmission by Y no later than[1] X does. Thus, O has some advance knowledge of every collision between X and Y. What could it do with this information? Clearly, O finds out about the transmission from X ahead of any "remote device" including Y. Thus, under collision truncation O acts as an *agent* for the "remote device" and *masks* the remainder of the incoming signal at the point where Y, had it been located closer to O, would have stopped transmitting according to the rules of CSMA/CD described in Section 2.1. Since O sees the same event timing as the "adjacent device," there is no point in having O act as an agent for the "adjacent device."

Now consider what happens if O applies collision truncation to the signal arriving from the "remote device." If the signal from X arrived first, then O would allow only the first 96 bits from Y's incoming signal to pass through (i.e., a minimum length collision fragment, consisting of a 64 bit preamble and a 32 bit jam) and then mask the rest. Conversely, if the signal from Y arrived first then O would allow either an additional 32 bits beyond the start of the collision or a total of 96 bits, whichever is greater, before masking the remainder of the incoming signal from Y. As a result, the "adjacent device," and all other hosts in the same subtree including X, see an earlier end to the collision fragment originating from Y, and possibly also an earlier end to the entire collision event. Thus, from the perspective of all hosts in the "adjacent" subtree, including X, incoming collision truncation at O frees up some of the network bandwidth lost to collisions and may also reduce the network delays experienced by these hosts by allowing them to begin their next attempts a little sooner.

Obviously, however, this incoming collision truncation rule can only be used if it does no harm on the *other* half of the network, which contains all hosts that belong to the subtree rooted in the "remote device," including Y. The danger here is that O has tricked X into believing that Y's collision fragment ended when O began masking the incoming signal, instead of at some later time when Y actually stopped transmitting. In particular, if X made another transmission attempt right away, the *new* signal from X might enter the link segment while O is still masking the *previous* signal from Y, and O sees a collision that is hidden from all hosts in the "adjacent" subtree. Moreover, such a collision would be inconsistent with the sequence of events seen by X, since X knows that it waited for Y's fragment to end before transmitting and therefore, because of the triangle inequality, the start of its new attempt cannot overtake the end of Y's fragment as it is broadcast to the rest of the network.

Fortunately, however, the conditions that created the "hidden collision" at O no longer apply by the time the new from X signal reaches the other end of the link segment. We consider two cases. First, suppose the "remote device" is the host Y itself. In this case, we know:

1. Y must have started to transmit before it heard the previous signal from X, or it would have deferred rather than participating in the collision;
2. because of the collision handling rules for CSMA/CD described in Section 2.1, Y must stop transmitting no later than 96 bit-times after the beginning of the previous signal from X arrived; and
3. X must have transmitted for at least 96 bit-times in the previous collision, waited for the network to become idle, and then waited again for a further 96-bit interframe gap before starting its new transmission.

Therefore, the new signal from X does not arrive at Y for *at least* 96 bit-times after Y stopped its previous transmission, as required.

The other possibility is that the "remote device" is a repeater, and host Y is connected to one of its other ports. But in this case, the above host-based timing analysis can be applied directly to the repeater state machine if we assume that the same collision truncation rule is used at each repeater port. In other words, the repeater state machine (and hence the outbound signals from each repeater port) would see no overlap between the previous transmission from Y and the new transmission from X because Y's incoming signal must have been masked at its input port within 96 bit-times of the arrival of the previous signal from X. Therefore, we are finished, because the situation on the next link segment from the repeater to Y itself is identical to the host-based timing analysis that was treated in case I.

Thus, to summarize, collision truncation is applicable to both host interfaces and repeater ports. Notice, however, that in proving that collision truncation at O did not lead to inconsistencies in the "remote" subtree, we needed to assume that the "remote device" did collision truncation if it was a repeater, but not if it was a host. Thus, collision truncation may optionally be applied to any or all of the hosts. However, whenever collision truncation is used, it *must* be applied to *all repeater ports*. Moreover, it can also be

[1] X and O will find out at the same time if O is adjacent to X; otherwise, O will find out earlier.

shown [6] that collision truncation can be used in networks with more than one repeater as long as we have a star-wired network topology and every repeater does collision truncation.

3.2 Application

Figures 3 and 4 show the effect of inserting a collision truncating repeater into the signal path for the same two situations as were shown in Figures 1 and 2, respectively.

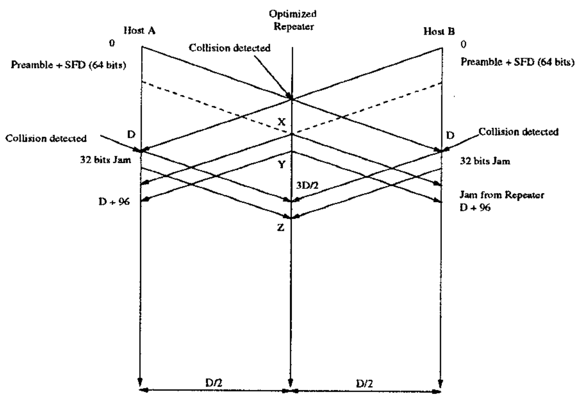

Figure 3: Repeat of Figure 1 with the addition of collision deletion

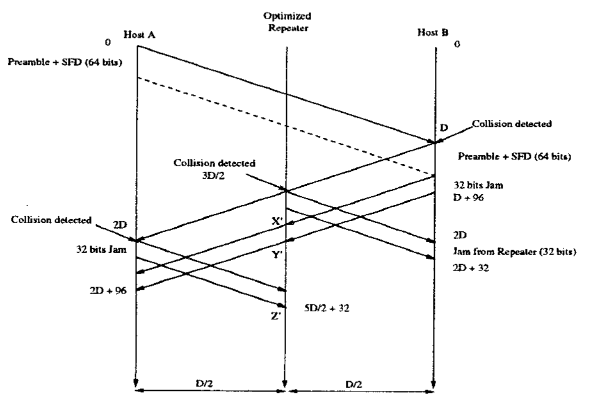

Figure 4: Repeat of Figure 2 with the addition of collision deletion

Figure 3 shows the event timing when we assume that hosts A and B transmit simultaneously, and that a collision truncating repeater has been added at the midpoint of the network. Notice that the collision is detected at time $D/2$ at the repeater ports instead of at time D at the hosts. Thereafter, the each repeater port continues to supply incoming bits to the repeater core for the duration of a preamble and start-frame delimiter (ending at time $X = D/2 + 64$), then supplies a further 32 bits of jam to the repeater core (ending at time $Y = D/2+96$), and finally ignores the remainder of the incoming collision fragment. Thus, the repeater core stops receiving all input at time Y, at which point it stops forwarding an outbound signal to the ports. Hence no further transmissions occur on the outgoing links beyond time Y and the hosts see the end of the incoming signal from the repeater at time $D + 96$. Meanwhile, the two hosts each finish their respective jam signals and stop transmitting at time $D + 32$, so the corresponding input ports on the collision truncating repeater continue to receive incoming collision fragments until the end of transmission event reaches them at time $Z = 3D/2+32$. In this case, collision truncation reduces the duration of the collision as seen by the two transmitters from $2D+32$ to $D+96$ bit-times. For 10 Mb/s and 100 Mb/s operation, this represents a reduction from at most 511 bit-times to at most 335 bit-times, or 34% for a maximum topology. For 1000 Mb/s operation, this represents a reduction from at most 4095 to at most 2127 bit-times, or 48% for a maximum topology. If the repeater has any other ports, then those other ports will start transmitting an outbound signal at time $D/2$, as before, but will now stop at time Y, instead of at time Z for a conventional repeater, which is 96 bit-times instead of $D + 32$ bit-times. For a maximum topology, this is a reduction of 175 bit-times, or 65%, at 10 Mb/s or 100 Mb/s, and a reduction of 1967 bit-times, or 95% at 1000 Mb/s.

Figure 4 shows the same event timing as Figure 2, where we assume that host B waits as long as possible before starting to transmit, and we again assume that a collision truncating repeater has been added at the midpoint of the network. This time the collision is detected by the repeater at time $3D/2$, which is between the times $2D$ and D at which the collision is detected by hosts A and B, respectively. Since the repeater has already received D bits from host A before detecting the collision, it just needs to accept another 32 bits before masking the remainder of the input—even though incoming data continues to arrive from host A until time $Z = 5D/2 + 32$. Conversely, the repeater has just started to receive data from host B, so it accepts the next 96 bits before starting to mask the rest of the input from B at time $Y = 3D/2 + 96$, at which point the incoming signal ends immediately anyway. Thus, host A finishes transmitting at time $2D + 32$ and senses the end of all activity at time $2D + 96$, just as it did in Figure 2 without collision truncation. However, host B now senses the end of all activity much earlier than before, namely at time $2D+32$ instead of at time $3D+32$ in Figure 2. In this case, we have reduced the duration of the collision as seen by host B from $2D + 32$ to $D + 32$. For a maximum topology, this is a reduction of 240 bit-times or 47% at 10 Mb/s and 100 Mb/s, or 2048 bit-times or 50% at 1000 Mb/s. Once again, if the repeater has any additional ports, then they will begin transmitting an outbound signal at time $D/2$, but they will stop transmitting at time Y, instead of at time Z for a conventional repeater, which is $D + 96$ instead of $2D + 32$ bit-times. For a maximum topology, this is a reduction of 176 bit-times or 34% at 10 Mb/s and 100 Mb/s and 1984 bit-times or 48% at 1000 Mb/s.

Figure 4 also demonstrates another interesting feature that results from repeater-based collision truncation. By comparing Figures 3 and 4 (or Figures 1 and 2 for a conventional network), we see that the total elapsed time during a collision event is much smaller when the starting times of the transmitters are synchronized rather than offset. Unfortunately, the offset transmission scenario depicted in Figures 2 and 4 could easily arise in practice if the previous event on the network had been a successful frame transmission originating from host A. In this case, if host A had another frame to transmit, and host B became ready to transmit while A was transmitting its previous frame, then both hosts would attempt to transmit as soon as possible after they detected the end of the previous frame. Hence, after each host leaves the required 96 bit interframe spacing on the network, the offset in starting times in this first collision would be as large as possible. Figure 2 demonstrates that this large difference in starting times will persist across multiple collisions in a conventional CSMA/CD network, whereas Figure 4 shows that the actions of a collision truncating repeater eliminates almost all of this offset after one collision.

It is interesting to note that collision truncation at the repeater creates the illusion that an alternate, but still valid, sequence of events took place in a collision. In general, when two transmitters collide, each one can determine the location (in both space and time) for the start of the other transmission. The distance to the other host, measured in bit-times, is simply half the elapsed time from the start of its own transmission until the start of the incoming jam signal. The starting time for the other transmission is the observed starting time for the collision, minus the distance. Thus, in Figure 3 each host sees an incoming collision fragment that looks like it was generated from a point on the start-of-carrier line that is at most 32 bit times (i.e., half the length of the preamble and start-frame delimiter) beyond the repeater. Similarly, in Figure 4 the collision fragment from host A looks like it was generated by the repeater.

3.3 Collisions of Higher Multiplicity

If the distances between hosts is highly variable, then the space-time boundary for a collision can become quite complicated—even without collision truncation. The difficulty arises because the minimum collision fragment size generated by a transmitter is much smaller than the maximum round-trip propagation delay (i.e., the ratio is about 1 : 5 for 10 Mb/s or 100 Mb/s operation, and about 1 : 40 for 1000 Mb/s operation). Thus, two nearby hosts A and B could collide to form a carrier event that is much shorter than the end-to-end propagation delay—known as a "flying pygmy." Repeater-based collision truncation can also turn a "normal" collision between widely separated hosts into a "flying pygmy" if the signals happen to arrive at the repeater at approximately the same time.

In general, "pygmies" are a feature rather than a problem, because they consume less network bandwidth than larger collision fragments. However, the "pygmy" could later collide with a transmission by some distant host, C, that started long after both A and B were already silent. Even in this case, such a sequence of events is just a normal part of the CSMA/CD algorithm. The choice of whether two hosts will collide or one will defer to the other is based only on the timing of the *leading edges* of their respective transmissions, and not their lengths; all of these hosts would have collided with each other anyway, whether or not A and B created the "pygmy." On the other hand, such high-multiplicity collisions can lead to inconsistent views of the collision by different hosts, with those near A and B seeing two separate carrier events and those near C seeing only one. Such inconsistencies are not significant because of the robustness and simplicity of the CSMA/CD algorithm. And, finally, the gap in the carrier event may lead some additional host D near A and B to join the middle of the collision, which would not have happened if the signals from A and B had lasted longer. While it is unfortunate that D suffered a collision, it is not somehow "undeserved" since D would have transmitted and collided with C anyway if A and B had been silent. In other words, it should be clear that adding collision truncation may change the sequence of events in a network, but not in harmful way.

4 Experimental Results

4.1 Effect of Truncation on Collision Overhead

Figures 5–8 show how the minimum, maximum and average size of collisions, as seen at the repeater, vary as a function of the one-way propagation delay on each

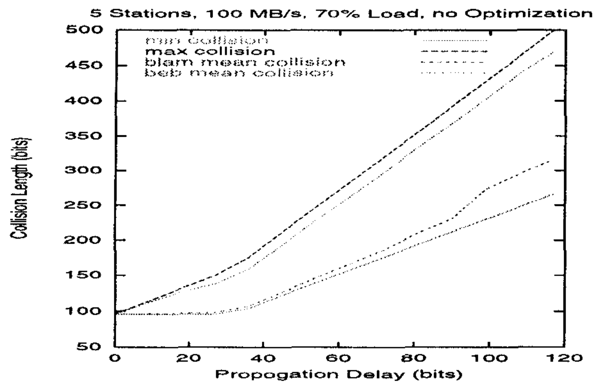

Figure 5: Sensitivity of collision sizes to the length of each link for BEB and BLAM on 100 Mb/s Fast Ethernet without collision truncation

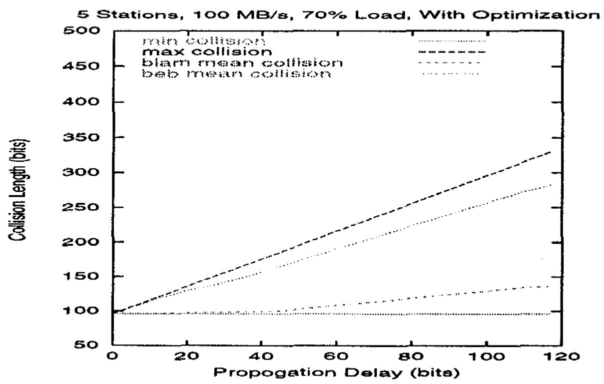

Figure 6: Sensitivity of collision sizes to the length of each link for BEB and BLAM on 100 Mb/s Fast Ethernet with collision truncation

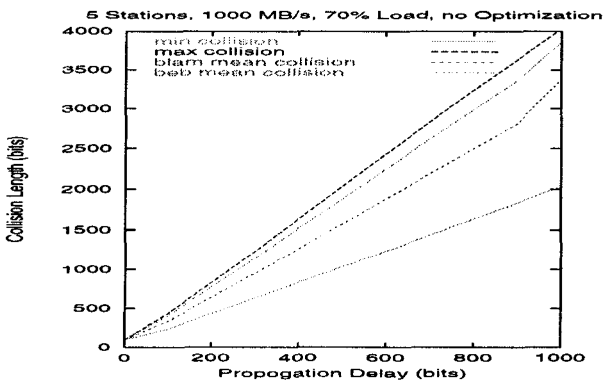

Figure 7: Sensitivity of collision sizes to the length of each link for BEB and BLAM on 1000 Mb/s Gigabit Ethernet without collision truncation

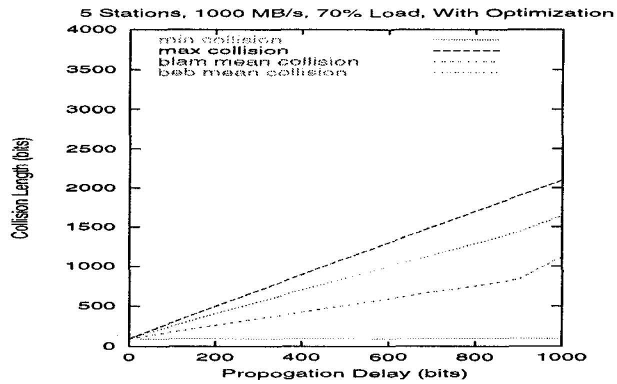

Figure 8: Sensitivity of collision sizes to the length of each link for BEB and BLAM on 1000 Mb/s Gigabit Ethernet with collision truncation

host-repeater link. The network load is fixed at 70% in each case. Because of the bit budget allocated to the circuitry in the host adapters and the repeater, the minimum round-trip delay for a single-repeater network is about 80 bit times for 10BASE-T ([1], Table 13-2[2]), about 180 bit times for 100BASE-T ([2], Table 29-3), and about 1600 bit times for Gigabit Ethernet ([4], Table 42-3). Thus when reviewing these figures one should keep in mind that the smallest possible values for the one-way propagation delay on each link are approximately 20 bit times for 10BASE-T, 45 bit times for 100BASE-T and 400 bit times for Gigabit Ethernet. A carrier event at the repeater is classified as a collision if the number of active input ports ever exceeds one. In that case, we assume that the the duration of the collision includes everything from the first start-of-carrier event until the last end-of-carrier event. This length may be shorter than the length of the collision as seen by the transmitters, but it is the same as the length seen by all passive receivers. In Figures 5–6, we consider a 100 Mb/s Fast Ethernet network, with a 512-bit slot time, and in Figures 7–8 we consider a 1000 Mb/s Gigabit Ethernet network with a 4096-bit slot time.

Collision truncation is not used in Figures 5 and 7. As expected, for small link lengths, the minimum collision sizes start off at 96 bits (the minimum fragment size) and then grow linearly in accordance with the value $D + 32$ derived in Figure 1. Similarly, the maximum collision sizes grow linearly in accordance with the value $2D + 32$ from Figure 2. It is interesting to note that the mean collision size for BEB is always quite close to the maximum, whereas for BLAM the

[2]Beware that the entries in this table only include an extra 96 bits to account for a minimum-length collision fragment.

mean collision size is much lower. It is well known that under high load, the dynamics of BEB leads to the capture effect [5][9], in which one host can transmit tens or even hundreds of consecutive frames while the rest of the hosts keep backing off again and again. Thus, most of the collisions under BEB will involve an immediate retransmission by the same host—a situation likely to create the timing shown in Figure 1. Conversely, under BLAM all active hosts are equally likely to start transmitting at each of the contention steps—a situation likely to create the timing shown in Figure 2.

Figures 6 and 8 show the same experiment, except for the addition of repeater-based collision truncation. This time, the minimum collision size remains fixed a 96 bits, independent of D, in accordance with Figure 3, while the maximum collision sizes grow linearly in accordance with the value $D + 96$ from Figure 4. As before, and for the same reasons, the average value for BEB remains close to the maximum, while the average for BLAM is much lower. The Figures confirm the reductions in minimum and maximum collision sizes that were derived in Section 3.2. In addition, the data shows that the reduction in the average collision size is almost 50% for Fast Ethernet and even larger for Gigabit Ethernet.

4.2 Effect of Truncation on Delay

Figures 9, 10 and 11 show the mean end-to-end delay (in milliseconds) for both 100 Mb/s Fast Ethernet and 1000 Mb/s Gigabit Ethernet. In each case, the

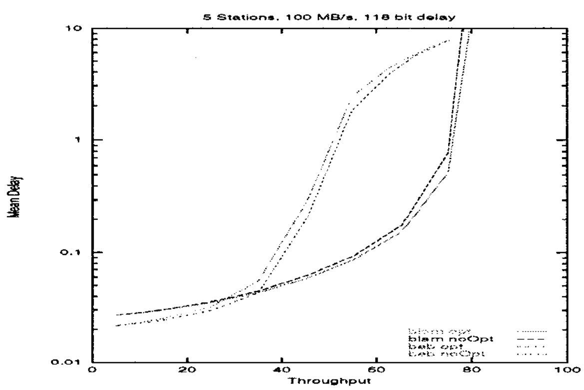

Figure 9: Effects of collision truncation on mean end-to-end delay for Fast Ethernet using BEB and BLAM

mean delay is plotted on logarithmic scale, since in each case it grows by a factor of a hundred as the load increases. As expected [5], the most striking effect is the difference between BLAM and BEB. For 100 Mb/s operation, BLAM reduces the mean end-to-end delay under moderate to high load conditions by a factor of 50, and for 1000 Mb/s operation with a burst limit of 8 kilobytes the improvement is more than a factor of 10. For Gigabit Ethernet with a burst limit of 12,000 bits, the difference between BLAM and BEB is very small until the system is completely saturated, at which point BEB can increase its throughput through the capture effect.

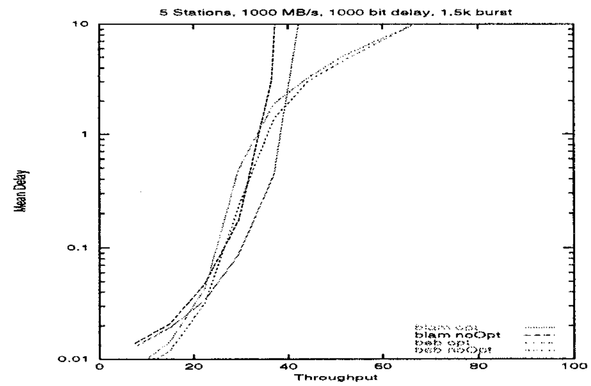

Figure 10: Effects of collision truncation on mean end-to-end delay for Gigabit Ethernet using BEB and BLAM, and a burst limit of 12,000 bits

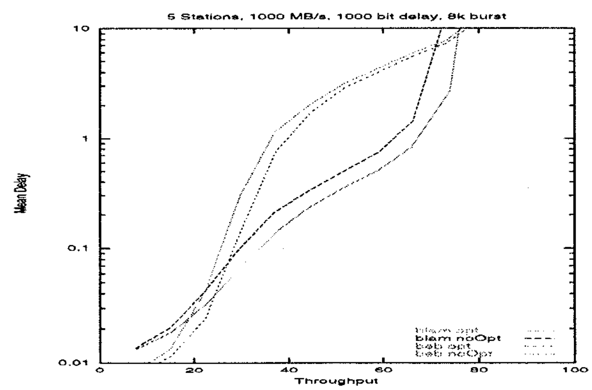

Figure 11: Effects of collision truncation on mean end-to-end delay for Gigabit Ethernet using BEB and BLAM, and a burst limit of 65,536 bits

Figures 12 and 13 show the percentage change in the delay for BEB and BLAM in the 100 Mb/s Fast Ethernet and 1000 Mb/s Gigabit Ethernet systems, respectively, when we add collision truncation. Even though the change in mean delay due to collision truncation looked quite small in Figures 9, 11 and 12, it can be seen that the percentage change is actually quite large, and is simply being masked by the huge range. In particular, for network loads in the range 20–40%,

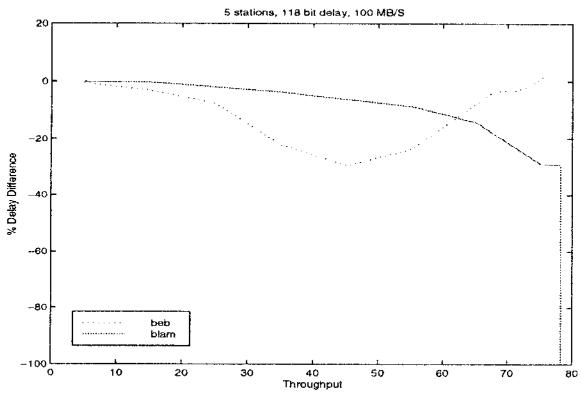

Figure 12: Percentage change in mean end-to-end delay from adding collision truncation to Fast Ethernet under BEB and BLAM

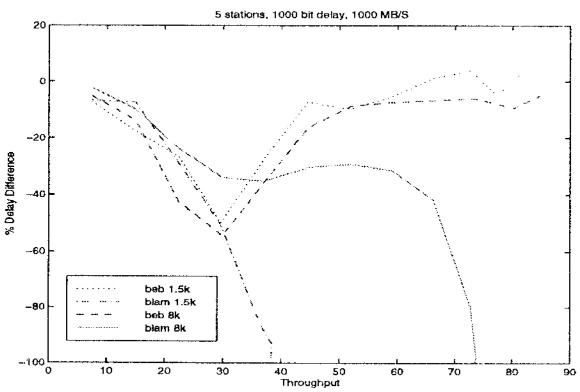

Figure 13: Percentage change in mean end-to-end delay from adding collision truncation to Gigabit Ethernet under BEB and BLAM

which is a fairly typical operating region for a shared Ethernet, collision truncation reduces the mean end-to-end delay by at least 30% in all cases. For 1000 Mb/s operation the percentage improvement is even larger, especially when the burst limit of 8 kilobytes is used. In general, BLAM seems to benefit more from the addition of collision truncation, which is to be expected because BLAM uses a lot of collisions as part of its arbitration procedure.

5 Conclusions

In this paper, we introduced the concept of collision truncation in half-duplex shared Ethernet systems. The technique is quite simple, requiring only a few minor changes to the finite-state machine that controls each port of a repeater. In particular, the changes do not require a new MAC layer entity to be added to the repeater, nor do we require a repeater to do any packet buffering, address-based and/or error filtering, or to have an internal backplane that runs faster than the signalling speed for one of its attached links. In short, we are not trying to turn the repeater into a bridge. Collision truncation can also be applied to a host interface, but only if the repeater(s) implement it too.

By describing how collision truncation affects the timing of events in the network, we showed that it cannot cause any correctness problems because one packet was able to overtake another, or because its actions created some unexpected collisions. Instead, the hosts simply conclude that the network diameter has become much smaller than it was before, and carry on in the normal way.

Although the effect of collision truncation on performance is much smaller than the difference between BEB and BLAM, we showed that it does provide a significant performance improvement. This is especially for Gigabit Ethernet, where collisions can be very large compared to the size of a frame. For 100 Mb/s Fast Ethernet, collision truncation reduces the mean delay for both BEB and BLAM by more than 25% at moderate values of network load. For 1000 Mb/s Gigabit Ethernet, the reduction in the mean delay is generally much larger and can reach more than 50%. In addition, collision truncation increases the maximum throughput for BLAM by 10%. Thus, considering its modest implementation cost, collision truncation looks like a very promising technique for reducing the effects of large propagation delays on shared Ethernet systems.

References

[1] ISO/IEC Standard 8802-3:1996 (ANSI/IEEE Std 802.3, Fifth Edition), *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*, Institute of Electrical and Electronic Engineers, New York, July 1996.

[2] IEEE Standard 802.3u-1995, *Media access control (MAC) parameters, physical layer, medium attachment units, and repeater for 100 Mb/s operation, type 100BASE-T*, Institute of Electrical and Electronic Engineers, New York, October 1995.

[3] IEEE Draft P802.3w/D1.8, *Enhanced media access control algorithm for IEEE 802.3 CSMA/CD*, Institute of Electrical and Electronic Engineers, New York, 1997.

[4] IEEE Draft P802.3z/D2, *Media access control (MAC) parameters, physical layer, medium attachment units, and repeater for 1000 Mb/s op-*

eration, Institute of Electrical and Electronic Engineers, New York, February 1997.

[5] M. L. Molle, *A New Binary Logarithmic Arbitration Method for Ethernet*, Technical Report CSRI-298, Computer Systems Research Institute, University of Toronto, 1994.

[6] M. L. Molle, "Reducing the effects of propagation delay on CSMA/CD networks." presented to the IEEE802.3 standards committee in March 1996. *URL: ftp://stdsbbs.ieee.org/pub/802_main/802.3-/gigabit/presentations/mar1996/MMredpd.txt.*

[7] M. L. Molle, "Making Half-Duplex Repeaters Attractive for Gigabit CSMA/CD," presented to the IEEE802.3 standards committee in March 1997. *URL: ftp://stdsbbs.ieee.org/pub/802_main/802.3-/gigabit/presentations/mar1997/MMrepeat.pdf.*

[8] M. L. Molle, K. Sohraby and A. N. Venetanopoulos, "Space-Time Models of Asynchronous CSMA Protocols for Local Area Networks," *IEEE Journal on Selected Areas in Communications*, July 1987

[9] K. Ramakrishnan and H. Yang, "The Ethernet capture effect: analysis and a solution," *19th IEEE Local Computer Networks Conference*, Minneapolis MN, October 1994, pp.228-240.