The Effect of Detail on Ethernet Simulation*

Alefiya Hussain, Aman Kapoor and John Heidemann
USC/Information Sciences Institute
{hussain, kapoor, johnh} @isi.edu
To appear: PADS-04, Kufstein, Austria, 16-19 May 2004

Abstract

An important question for network simulation is what level of detail is required to obtain a desired level of accuracy. While in some networks, the level of detail is an open research issue (for example, radio propagation models in wireless networks), it has long been assumed that wired networks could be accurately modeled by fairly simple queues with a bandwidth limit and propagation delay. To our knowledge this assumption has not been widely tested. In this paper we evaluate different levels of detail for an Ethernet simulation. We consider two models for Ethernet simulation; a detailed, CSMA/CD protocol based model and a more abstract model using a DropTail, shared queue. Using web traffic with two different TCP simulation models, we evaluated the accuracy of these Ethernet models as compared to testbed measurements. We observed the DropTail Ethernet model requires significantly less execution time and can accurately model performance using a bandwidth normalization factor.

1. Introduction

Network simulation is widely used to develop new protocols for the Internet. Even with the existence of public testbeds, simulation remains the only way to study very large models, and as software-only, simulation is much more accessible to researchers and students than a physical testbed. When designing simulations to study a protocol, a researcher needs to make choices regarding details in the protocol that should be exercised or implemented. Although there are a number of network simulation packages available, the onus of choosing the right level of detail for simulation still lies on the researcher.

Network protocols can be simulated at many levels of detail (or, alternatively, the level of abstraction)—from the physics of electromagnetic propagation, to bit-, packet-, and flow-level simulations. In some domains such as wireless networking, identifying appropriate levels of detail remain an open research challenge [13], and recent work suggests that radio propagation vagaries can greatly influence network protocol operation [6, 38], demonstrating a complexity beyond traditional analytic models (see for example [25]) that affects protocol operation.

In *wired* networks, however, fairly simple abstractions have long been assumed to be appropriate. For example, wide-area links are usually simulated as a queue, with some amount of fixed propagation delay, and a fixed bandwidth or bit-rate. Internet-like networks are then formed from collections of these abstract links. We know that networks in reality are much more complex, with multi-link routers with complex internal structure, often meeting at exchange points joined by local-area networks such as Ethernet or FDDI. While these abstractions are readily accepted, we are not aware of any systematic study about their validity—do these abstractions affect the accuracy of network simulations?

This paper begins that kind of systematic study, to understand what level of detail is needed in simulation models of wired networking protocols. In addition to common use at exchange points, Ethernet networks are often present as the last hop of most networks on the Internet today. With the fast growing Internet it has become important to revisit these lower layers and understand their behavior with respect to the changing traffic patterns and to understand network dynamics when a new protocol is deployed. We can only begin to explore such questions in this paper because ultimately the answer depends on the design question being investigated through simulation. Studies of capture effect might require a detailed model of Ethernet, while a trivial model is probably sufficient to simulate the Ethernet connecting a single home computer to a DSL modem.

Although Ethernet networks have been extensively studied in the past, yet little is known about their behavior. Several analytical models for throughput have been formulated [1, 3, 5, 8, 9, 11, 16, 17, 21, 32, 34, 33, 31, 30]. However due to the complexity of the CSMA/CD protocol, analytical approaches often make many simplifying as-

*The research is based on work supported by DARPA via the Space and Naval Warfare Systems Center San Diego under Contract No. N66001-00-C-8066 ("SAMAN"), by NSF under grant number ANI-9986208 ("CONSER").

sumptions about different parameters, for example packet size, load distribution, population size, to make the problem tractable. These assumptions sometimes lead to contradictory results. For example, research has shown maximum achievable CSMA/CD utilization of 60% with one set of assumptions [30], while later work showed utilization up to 97% with large packet sizes [4].

To complement analysis, several simulation models of Ethernet have been presented [1, 14, 19, 23, 22, 24, 36]. Many of the simulation models represent the Ethernet as a point-to-point link with special characteristics. It is not clear if abstractions in these networks are appropriate for different simulation scenarios. Simulation of the Ethernet with accurate representation of the CSMA/CD mechanisms such as binary exponential backoff, inter-frame spacing and transmission of a jamming signal after a collision are expensive in computational power and memory requirements. The complexity added to the simulation model by these mechanisms may or may not be justified during the investigation of a particular protocol. In particular it may not be worth the extra effort to model the physical layer interactions with the MAC protocol or the computational overhead at the interface. Increasing the level of detail in a simulation model makes it difficult to quickly explore alternate network scenarios and protocol interactions.

To overcome the limitation posed by computational power and memory, some simulation models use parallel and distributed Ethernet simulations [37]. This technique however requires extensive coordination among a set of machines and at times requires specialized hardware that may not be available. Hence a more viable option is to reduce the level of detail in the simulation model. Accessing the appropriate level of detail required to accurately simulate the Ethernet networks is a difficult problem and varies depending on the network scenario. A high level of detail in the model allows accurate simulation of the Ethernet, but increases the time required to implement and debug the design as well as takes longer to run simulations on different network scenarios. Making use of detailed simulations do not allow quick evaluation of network scenarios to see which ones are most interesting. On the other hand low level of detail may produce distorted and misleading results.

In this paper, we present an abstract model of the Ethernet that replaces the complex CSMA/CD mechanisms by a drop tail queue. Reducing the level of detail in the model allows quick implementation, debugging and testing cycles within the simulation framework and has lower runtime and comparable memory costs permitting quick studies of the Ethernet and its interactions with other components of the network. The abstract model can be normalized to achieve the accuracy of the detailed model.

The contribution of this paper is to study the effects of detail in Ethernet simulations on application performance. We compare qualitatively the Ethernet specification against two simulation models (Section 3). We then examine the quantitative performance difference on a web traffic workload, comparing testbed experiments against these models for both 10Mb/s and 100Mb/s networks (Section 5). We also consider the run-time simulation costs of these models (Section 5.3). We will show that, in this case, abstraction provides runtime performance improvements, and that it can provide high accuracy provided that bandwidth is normalized by a scaling factor.

2. Related Work

Ethernet behavior has been extensively studied both analytically [1, 5, 8, 9, 11, 16, 17, 3, 21, 32, 34, 33, 31, 30] and through simulation models [1, 14, 19, 23, 22, 24, 36]. Most analytical models developed, make assumptions about one or more parameters: packet size, cable length, population size etc. to make their calculations tractable. Metcalfe provided the first detailed measurement analysis of a 3Mbit/s Ethernet calculating throughput as a function of packet length [21]. Their results showed a near 100% throughput for large sized packets and a drop to about 37% for smaller packets. Almes studied the Ethernet in terms of response time as a function of offered load [1]. They showed that the response time stays under 1msec for an offered load of about 75% and grows asymptotically with any further increase in load. They developed a simulation model to evaluate their analytical results. The simulation results matched with their analysis for fixed packet sizes but were worse in case of variable packet sizes. Many researcher groups [30, 34, 33] analyzed CSMA/CD networks under finite population size. They measured throughput against the offered load and also studied the effect of the time taken to detect a collision on the throughput. Coyle studied the Ethernet under a finite population generating exponentially distributed inter-arrival times for packets [8, 9]. Due to the assumptions made in the analytical models, they sometimes provide incorrect/biased results. Takagi [30] showed that maximum throughput achievable by CSMA/CD is 60% when infact [4] implementation proved that for large packet size Ethernet throughput could be as high as 97% The Ethernet behavioral analysis is dependent on the functional requirements from the network and choosing an appropriate analytical model may be difficult. This has resulted in simulations being the prime source of analysis of the Ethernet.

Several simulation models have been presented for the Ethernet [1, 14, 19, 23, 22, 24, 36]. Many models represent the Ethernet as a point-to-point link with special characteristics. These assumptions in simulation can bias the results providing incorrect results. Tsui el al [36] models the Ethernet by dividing the link into sections of equal length, where each section is attached to at most one node. Other groups [24, 23] represented the Ethernet with a single server that kept count of number of the collisions seen by a packet and adjusted the packet delay accordingly. The REAL simulator [26] is based on a similar technique with a centralized master process to simulate CSMA/CD on point-to-point links. Armyros [2] developed a simulator for a equi-distant star topology of the Ethernet. Wang proposed a distributed simulation technique for complex Ethernet topologies [37].

The Ethernet performance has been evaluated using measurement studies [4, 12, 27, 29] by varying the packet lengths, network topologies and offered load. The Ethernet measurement studies indicate that the performance is high and very few packets are lost under normal offered load. In this paper we attempt to evaluate the amount of detail required to accurately simulate the Ethernet. We present two models for Ethernet simulation and perform measurement studies to validate our results.

3. Abstraction at different levels

We study the effect of abstraction at two layers in the network; the medium access control (MAC) layer and the transport layer. At the MAC layer, we propose two models of abstraction for studying the Ethernet, one a CSMA/CD model and the other a droptail model while at the transport layer, we consider a full TCP and a simple TCP model.

3.1. Medium Access Control

The proposed CSMA/CD and the droptail MAC simulation models are developed in ns [20], a discrete event driven simulator targeted at networking research. Each node in the network has its own buffer that stores packets before they are transmitted on the shared medium. Both MAC models in this paper focus on wired local area networks or the Ethernet. The CSMA/CD simulation model depicted in Figure 1 implements the complexities of the CSMA/CD multiple access protocol as specified by the IEEE 802.3 specification [7, 15]. The IEEE 802.3 specification proposes a 1-persistent Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to control access of the shared transmission medium. End hosts monitor the carrier sense signal on the medium and defer to the passing traffic by delaying transmission of the data frame. When the medium is subsequently sensed idle, indicating the completion of the previous transmission, the end host continues to defer for an additional inter-frame spacing before initiating the transmission of the data frame. The inter-frame spacing provides recovery time for the CSMA/CD sublayer and the physical medium.

To handle collisions created due to transmissions from multiple end hosts, the specification additionally outlines a collision detection procedure. A transmitting end host should continuously monitor the signal on the medium to detect collisions. When a collision is detected, the transmitting end host propagates additional bits specified by the jam size on the medium to ensure collision detection at all end hosts on the local area network. The transmitting end host then attempts retransmission after waiting for a random time interval. On failure to retransmit, the end host doubles the waiting time interval, and attempts retransmission after a random period chosen from the new time interval. If the packet is not successfully transmitted after 16 attempts, the packet is dropped. The randomization of the delay after collision is also called truncated binary exponential backoff.

The CSMA/CD model depicted in Figure 1 is a detailed emulation of the IEEE 802.3 specification observed in physical networks. The nodes in the simulated network defer to the passing data frames and delay their transmission for an additional inter-framespacing once the medium is idle. When a collision is detected, the transmitting node enforces a jamming signal and performs a binary exponential backoff. After sixteen retransmission attempts, an error is reported and the frame is dropped. Although the CSMA/CD model is a careful implementation of the IEEE 802.3 specification, a few small mechanisms are not implemented due to limitations of the programming environment. The complete difference between the IEEE 802.3 specification and CSMA/CD model is outlined in Table 1. The droptail MAC model simplifies the shared access protocol and the contention mechanism. The model replaces the complex contention mechanism by a single shared drop tail queue as shown in Figure 2. All nodes on the network send data frames to the common queue, which in turn transmits the data frames onto the shared medium on a first-come-first-served basis. Each node at any time can have only a single frame in the shared queue, additional frames are stored in the individual buffers present in the node. The model thus eliminates the need to perform complex carrier sensing and collision detection mechanisms. The deference delay and binary exponential backoff delay experienced in the CSMA/CD model and the physical network is replaced with a queuing delay at the shared queue. Further, similar to drop of frames during high load in the IEEE 802.3 specification, frame loss will occur in the abstract model when the queue is full. Table 1 summarizes the difference between the proposed models and the IEEE 802.3 specification.

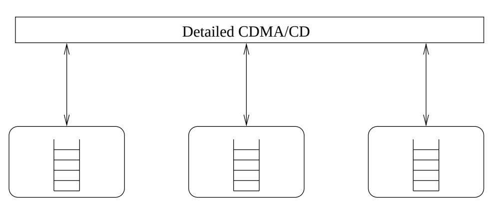

Figure 1. The CSMA/CD MAC model simulates the operation complexity of the IEEE 802.3 protocol

Table 1. Comparison: IEEE 802.3 specification, CSMA/CD and Droptail Ethernet simulation models

Feature	IEEE 802.3 Specification	CSMA/CD Ethernet Model	Droptail Ethernet Model
Deference Mechanism	Nodes defer to passing frames. Ensures at least one inter-frame spacing between frames. If carrier is sensed during the first $2/3^{rd}$ of the inter-frame spacing, reset inter-frame spacing timer.	Nodes defer to passing frames. Ensures at least one inter-frame spacing between frames. Does not reset timer.	A frame generated by a node, defers to all frames generated before it. All frames all queued in the shared queue and transmitted on the medium on a first-come-first-served basis.
Collision Detection and Enforcement	Collision is detected by the physical signaling unit. The collision signal is enforced on the medium by transmitting the additional bits called the jamming signal.	Collision is detected by a busy semaphore. The collision signal is enforced on the medium by transmitting the additional bits called the jamming signal.	No collisions occur since a shared queue ensures only one frame on the Ethernet.
Backoff and Retransmission	Uses truncated binary exponential backoff. A maximum of 16 retransmission attempts are made before reporting an error.	Uses truncated binary exponential backoff. A maximum of 16 retransmission attempts are made before reporting an error.	Since no collision occurs, a retransmission strategy is not needed.
Frame Size	Maximum: 1518 bytes Minimum 64 bytes Higher network layers modify frame if it does not fit the size specification via fragmentation and padding.	Print a warning message when frame size $>$ 1518 bytes. Adds padding if size $<$ 64 bytes	Print a warning message when frame size $>$ 1518 bytes. Adds padding if size $<$ 64 bytes.

3.2. Transport layer models

Simulation approximations can occur at multiple levels and hence it is important to study effect of abstractions at both the MAC level and network. Thus in this paper, in addition to the MAC models, we evaluate two transport layer TCP models; a simple TCP model and the full TCP model. The simple TCP model simulates all the details of the TCP congestion and error control behavior, however, the model does not simulate the connection establishment and tear down phases with SYN and FIN packets. Additionally during congestion and flow control, the simple model makes use of packet units (rather than byte units) for sequence numbers and acknowledgment numbers. This simple abstraction in TCP permits the transfer of data without initial connection setup and thus stores less state regarding the current source-destination transfer. On the other hand, the full TCP model is more detailed and simulates connections establishment and tear down phases using SYN and FIN packets. The full TCP model supports bidirectional data transfer and tracks sequence and acknowledgment numbers in packets. Although the simple TCP model is sufficient in many simulation scenarios where fidelity in the connection setup and teardown phases is not critical, the full TCP model is recommended when modeling a realistic TCP implementation. The connection establishment phase may be important in when trying to model short data transfers.

In this paper, we evaluate the effect of both layers of abstraction and validate our results with testbed measurements on both a 10Mbits/s and a 100Mbits/s local area network. The problem space exercised in the paper is summarized in Table 3.2. We describe the experimentation methodology in the next section.

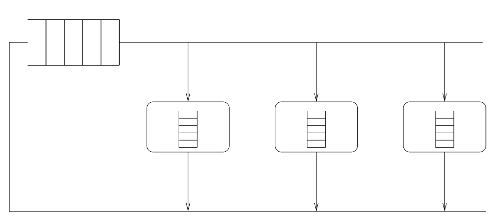

Figure 2. The droptail MAC model makes use of a shared queue to represent the contention mechanism

4. Methodology

We evaluate the abstraction models proposed in Section 3 by comparing the results with testbed measurements. We outline the simulation and the testbed configuration next whereas the experimental results are presented in Section 5.

4.1. Simulation Environment

We study the effect of abstraction in CSMA/CD and droptail MAC models by comparing the throughput measurements obtained in simulation with measurements for the similar environment on a testbed. All measurements, both in simulation and the testbed, are gathered by making use of the topology described in Figure 3. The local area network consists of a single web server with multiple web clients.

We consider a physical network consisting of 10Mb/s and 100Mb/s Ethernet LANs, both in simulation and testbed experimentation. The Ethernet configuration parameters applied for the models are given in Table 3. We make use of the same bandwidth and propagation delay parameters for the droptail MAC model.

For our traffic workload, the number of web clients is progressively increased from 1 to 30. Network traffic is generated by a client node when it instantiates a web session with the server node. Every web session is made up of multiple pages where each web page consists of many web objects. The web session emulates user browsing behavior on the Internet, where a web page may contain multiple images (or objects) [18, 28]. In this paper, to reproduce similar web traffic on the testbed and in simulation, each web page consists of a single object. The web object size follows a cumulative distribution proposed by the Webstone benchmark [35] and is given in Table 4.1. The web object size is heavy-tailed where nearly 99% web pages are less then 50Kbytes and 0.1% of the pages are more than 5Mbytes. We believe this emulates the web page distributions observed on the Internet [10].

Each simulation experiment considers 10 minutes of simulation time. We vary three variables: network speed (10 or 100Mb/s), LAN model (a simple simulation model, a more detailed model, and a real testbed), and number of clients (from 1 to 30). We then evaluate the mean throughput and simulation time and simulation memory usage.

4.2. Testbed setup

To evaluate our simulation models we created a testbed environment for 10Mbit/s and 100Mbit/s LANs. We use the Webstone benchmark software [35] to record the throughput, using the same object size distribution as the simulation models (Table 4.1). The setup consists of one web server machine and five client machines in a private non-switched network and all measurements are made after 10minutes of run time. Every client machine can support multiple web clients, each capable of a distinct web session with the web server. The web sessions follow the same methodology outlined in Section 4.1. To make sure the limited number of client machines do not change the measurement results, we ensured that the client machines saturate the link and also observed the collision rate. The number of collisions stay fairly constant on adding a new machine. Hence, we believe that the above setup shall produce similar results as by an extended 30 machine LAN. In future we plan to extend the testbed to 30 clients to validate our belief. The 10Mbit/s experiments used a 200Mhz Pentium Pro, 128MB of physical memory machine running FreeBSDv4.3 and an Apache web server. The 100Mbit/s experiments used a Pentium III Xeon, 532MB of physical memory machine running FreeBSD v4.7 and an Apache web server. We monitored the server utilization during the experiments to determine if the bandwidth measurements were being affected

Table 2. Problem space exercised in simulation and testbed.

MAC layer model	Transport layer model			
	Simple TCP	*Full TCP*	*FreeBSD 4.3*	*FreeBSD 4.7*
CSMA/CD	X	X		
Droptail	X	X		
Testbed 10Mbit/s			X	
Testbed 100Mbit/s				X

Table 3. Configuration parameters: CSMA/CD model

Parameters	10Mbit	100Mbit
Bus Bandwidth	10Mbit/s	100Mbit/s
Max Propagation Delay	950ns	950ns
Jam time after Collision	3.2μs	0.32μs
Slot Size	51.2μs	5.12μs
Inter-frame Delay	9.6μs	0.96μs
Max Frame Size	1518 bytes	1518 bytes
Min Frame Size	64 bytes	64 bytes

Table 4. Web object distribution

Web Object Size	Access Probability	Cumulative Distribution
500B	0.35	0.35
5KB	0.50	0.85
50KB	0.14	0.99
500KB	0.009	0.999
5MB	0.001	1

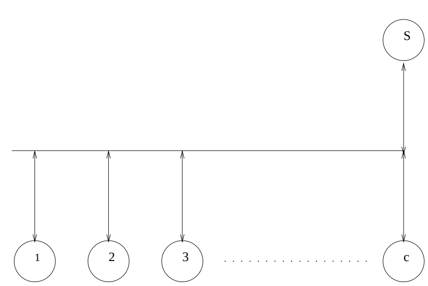

Figure 3. The Ethernet Simulation Topology

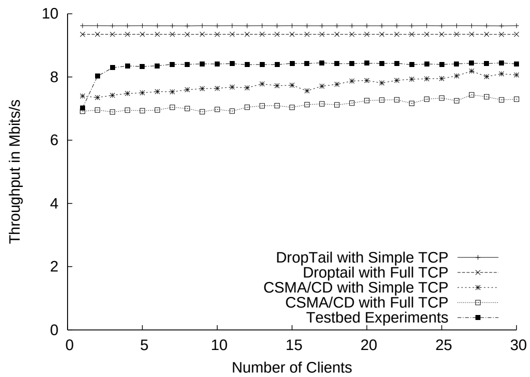

Figure 4. 10Mbit throughput against number of clients

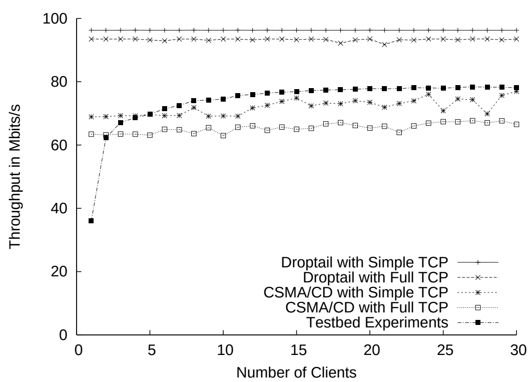

Figure 5. 100Mbit throughput against number of clients

by a bottleneck at the web server. The CPU utilization on the web server was always less than 60% indicating that the bandwidth in the experiments is limited by the network resources, not the server utilization. In the next section we present the analysis of our results.

5. Evaluation

Our first goal is to understand the relative accuracy of the MAC models as compared to each other and testbed experiments. We then consider the effect of TCP abstraction on accuracy. Finally, we evaluate the runtime and memory costs of detail in the simulation. We explore each of these issues in this section.

5.1. Effects of MAC Detail on Accuracy

To evaluate the relative accuracy of the simulation models, we measure and compare the throughput (the number of bytes transferred per unit time) obtained in each MAC model. Figure 4 shows observed throughput for the simulation and testbed measurements in a 10Mbit/s LAN and Figure 5 indicates the results for a 100Mbit/s LAN. We ran all experiments thrice and the variation in the throughput values was less than 2% and hence we did not plot errorbars.

We can make a number of observations from these two graphs. First, we observe that the throughput obtained in the simulation models of CSMA/CD closely follows the measurements on the testbed. Since this model is the most detailed and hence closely resembles the testbed environment, the above results validate the accuracy of the model. The results for both LAN capacities indicate that the CSMA/CD model measurements are close to testbed measurements but are slightly more conservative, with a 7% lower throughput. We believe the results in simulation are more conservative as the number of collisions in the simulation environment are higher. While the simulation model delay and back-off, it does not model jitter. In the real network, randomized jitter helps packets avoid collisions.

Next we observe that the droptail model has consistently higher throughput measurements at both LAN speeds. Although the queuing delay in the droptail model emulates the carrier sensing characteristic in CSMA/CD model, it does

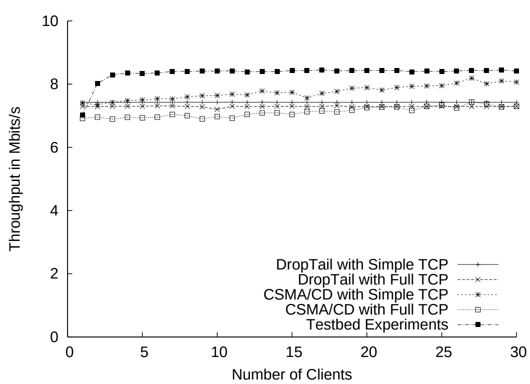

Figure 6. 10Mbit throughput measurements with 22% normalization of bandwidth for droptail model

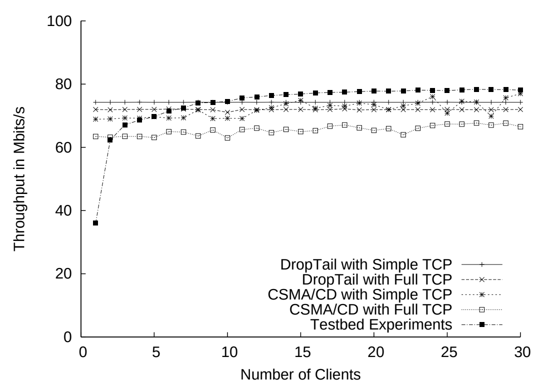

Figure 7. 100Mbit throughput measurements with 22% normalization of bandwidth for droptail model

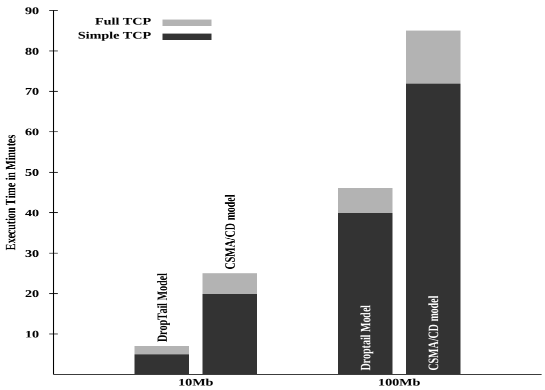

Figure 8. Real execution time corresponding to 10 minute Simulation time

not effectively capture the delay caused due to binary exponential back-off after a collision, and so it allows a higher utilization. Later, we discuss how the droptail model can be used in place of the CSMA/CD model to simulate the Ethernet. The throughput difference in the TCP models is discussed in the next section.

5.2. Effect of TCP abstraction

In addition to comparing MAC protocol detail, we considered two versions of TCP: ns's simpler one-way TCP, and full-TCP. The difference between these two TCP models is discussed in Section 3.2. We next compare the throughput results of simple and full TCP models. The Figures 4 and 5 indicate that the throughput measurements for simple TCP model are higher than the Full TCP model for both the droptail and the CSMA/CD models indicating that Ethernet performance is dependent on how the higher layers characterize the traffic. The cumulative distribution in Table 4.1 shows that about 85% of the generated web traffic consists of short duration flows that have fewer than 4 frames. Therefore connection establishment, tear down, and slow start mechanisms govern the data transfer and results in many small packets on the network. The lower throughput in the Full TCP model is due to the connection establishment and teardown phases absent in the simple TCP model.

5.3. Memory usage and Runtime measurements

A common motivation for abstraction is improved performance, so in addition to understanding the cost in accuracy we wanted to measure the differences in simulation runtime and memory usage. (The other common motivation of abstraction is simplicity of implementation; we do not consider that here because it is more difficult to quantify.) In this section, we compare the memory and runtime gains obtained by making simplifications in the MAC model.

The droptail simulation model is very easy to implement since it does not require collision detection, back-off and retransmission mechanisms. Figure 8 shows the execution time for the CSMA/CD and the droptail models for both 10 and 100Mb/s networks. Execution time shown in the figure represents the average computation time during three independent simulation runs. We observe that eliminating the detail in the MAC models reduces the execution time considerably. The droptail model is 75% faster than the detailed CSMA/CD model, and we observe that using Full TCP increases execution time by as much as 17-25%. We conclude that the much larger amount of abstraction in the MAC models results in a corresponding performance benefit. It is also important to note that the execution time difference increases by more than 15% on a 100Mbit network as compared to the same experiment scenario on a 10Mbit network. We might expect that CPU savings would scale linearly with increase in bit rates, but it does not, presumably due to constant factors.

Memory usage for 10Mbit MAC models ranged between 1O-12MB of RAM for all the simulation experiments. Similar results were observed for 100Mbit simulations.

5.4. Correcting Throughput with Abstract MAC Models

The above results indicate using an abstract MAC and TCP model will provide considerable savings in execution time and memory usage, although at some cost in accuracy. In our experiments, the models overestimated throughput by a constant amount (about 22%). This observation suggests that, when this constant can be determined, it can be applied as a correction factor to provide simulations that are both accurate and faster. Figures 6 and 7 indicate that the normalized throughput simulations do provide better results as compared to previous runs. With this correction factor throughput for droptail is similar to the detailed CSMA/CD model and the testbed. Thus the droptail model provides a reasonable analysis of Ethernet behavior if we normalize the bandwidth along with lower run-time and implementation costs.

If this approach is to be generalized to other simulations we need to understand the factors that affect the normalization factor. Is it a constant value or does it change as a function of network type, traffic load, or other factors in the scenario? For our scenarios we were able to determine normalization factors that were basically constant across varying workloads and network type. We are currently exploring what set of simulation scenarios have consistent normalization factors.

6. Conclusion

This paper presented a study highlighting different levels of abstraction on the Ethernet. We presented an abstract simulation model the eliminates the overhead of complex

CSMA/CD mechanism by using a shared queue. The abstract model provides significant savings in computational time requirements and allows quick exploration of protocol integrations without getting lost in the details of the model. Especially at high loads, abstraction provides a significant advantage by giving comparable results at lower computation time. We also studied the affect of TCP level abstraction on both the proposed Ethernet models.

However while using an abstraction care must be taken to understand the impact of the abstraction on the observed results. It is essential to employ some validation techniques to ensure their accuracy.

Acknowledgements

We would like to thank Joe Touch and Yu-shun Wang for providing the USC/ISI Postel Center (www.postel.org) resource for our testbed measurements. We would also like to thank all the anonymous reviewers for their helpful comments on the earlier versions of this paper.

References

[1] G. T. Almes and E. D. Lazowska. The behavior of ethernet-like computer communications networks. In *Proceedings of the 7th ACM Symposium on Operating Systems Principles (SOSP)*, pages 66–81, 1979.

[2] S. Armyros. On the behavior of ethernet: Are existing analytic models adequate? Masters Thesis.

[3] E. Arthurs and B.W.Stuck. A theoritical performance analysis of three bus transmission medium access methods. In *IEEE Infocom*, pages 128–138, Las Vegas, Nevada, USA, 1982.

[4] D. R. Boggs, J. C. Mogul, and C. A. Kent. Measured capacity of an ethernet: Myths and reality. In *Proceedings of the ACM SIGCOMM Conference '88*, pages 222–234, Stanford, California, Aug. 1988. ACM.

[5] W. Bux. Local area subnetworks - A performance comparison. In *IEEE Trans. Commun., COM-29(10):1465–1473*, 1981.

[6] A. Cerpa, N. Busek, and D. Estrin. SCALE: A tool for simple connectivity assessment in lossy environments. Technical Report 21, University of California, Los Angeles, Center for Embedded Networked Sensing, Sept. 2003.

[7] I. S. Committee. Ieee standards for local area networks: carrier sense multiple access with collision detection (csma). Technical Report ANSI/IEEE Std 802.3-1985, IEEE, 1985.

[8] E. J. Coyle and B. Liu. Finite population CSMA/CD networks. *IEEE Transactions on Computers*, COM-31(11):1247–1251, Nov. 1983.

[9] E. J. Coyle and B. Liu. A matrix representation of CSMA/CD networks. *IEEE International Conference on Communications: Integrating Communication for World Progress*, 1:53–64, 1985.

[10] M. Crovella, M. Taqqu, and A. Bestavros. Heavy-tailed probability distributions in the world wide web. In *A Practical Guide To Heavy Tails, chapter 1, Chapman and Hall,*, pages 3–26, New York, 1998.

[11] T. A. Gonsalves and F. A. Tobagi. On the performance effects of station location and access protocol parameters in Etherent networks. *IEEE Transactions on Computers*, 36(4):441, Apr. 1988.

[12] R. Gusella. A measurement study of diskless workstation traffic on an ethernet. *IEEE Trans. on Communications, Vol. 38, No. 9*, pages 1557–1568, Sept 1990.

[13] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. Effects of detail in wireless network simulation. In *Proceedings of the SCS Multiconference on Distributed Simulation*, pages 3–11, Phoenix, Arizona, USA, January 2001. USC/Information Sciences Institute, Society for Computer Simulation.

[14] H. Hughes and L. Li. Simulation model of an ethernet. In *Computer Performance, Vol. 3, No. 4*, pages 210–217, Dec. 1982.

[15] IEEE. Ieee std 802.3, 1998 edition. Technical Report ANSI/IEE Std. 802.3-1988, IEEE, 1998.

[16] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: part I carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE trans. on commun.*, COM-23:1400–1416, 1975.

[17] S. S. Lam. Carrier Sense Multiple Access Protocol for local networks. *Computer Networks: The International Journal of Distributed Informatique*, 4(1):21–32, 1980.

[18] B. Mah. An empirical model of HTTP network traffic. In *Proceedings of the IEEE Infocom*, pages 592–600, Kobe, Japan, Apr. 1997. IEEE.

[19] P. Marino and A. Rio. Microprocessing and microprogramming, 1993.

[20] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of the ACM SIGCOMM Conference '96*, pages 117–130, Stanford, CA, Aug. 1996. ACM.

[21] R. Metcalfe and D. R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, July 1976.

[22] P. J. P. O'Reilly and J. L. H. Jr. An efficient simulation technique for performance studies of csma/cd local networks. *IEEE Journal on Selected Areas in Communications, Vol. SAC-2, No. 1*, pages 238–249, Jan. 1984.

[23] K. Prasad and R. Patel. Performance analysis of ETHERNET based on an event driven simulation algorithm. In *Proc. 13th Conference on Local Computer Networks*, pages 253–267, Minneapolis, Minn., 10-12 1988. IEEE Computer Society.

[24] K. Prasad and A. Singhal. Simulation of ethernet performance based on single server and single queue model. In *Proceedings of the 12th Conference on Local Computer Networks*, pages 74–85, Minneapolis, Minnesota, USA, Oct. 1987.

[25] T. S. Rappaport. *Wireless Communications, Principles and Practice*. Prentice-Hall, 1996.

[26] RealEditv1.0. http://www.cs.cornell.edu/skeshav/real/.

[27] J. F. Shoch and J. A. Hupp. Measured performance of an ethernet local network. *Communications of the ACM*, 23(12):711–721, 1980.

[28] F. Smith, F. H. Campos, K. Jeffay, and D. Ott. What tcp/ip protocol headers can tell us about the web. In *ACM SIGMETRICS 2001/Performance 2001*, pages 245–256, Cambridge, MA, 2001.

[29] W. Smith and R. Kain. Ethernet performance under actual and simulated loads. In *In 16th Conference on Local Computer Networks*, pages 569–581. IEEE Computer Society Press, 1991., 1991.

[30] H. Takagi and L. Kleinrock. Throughput analysis for persistent CSMA systems. *IEEE Transactions on Computers*, COM-33(7):627–638, July 1985.

[31] X. Tan, S. Jue, E. Lo, and R. Hardy. Transient performance of a csma system under temporary overload conditions. In *Proceedings of 15th Conference of Local Computer Networks*, 1990.

[32] S. Tasaka. Dynamic behavior of a CSMA-CD system with a finite population of buffered users. *IEEE Transactions on Communications*, COM-34(6):576–586, June 1986.

[33] F. Tobagi and N. Gonzalez-Cawley. On csma-cd local networks and voice communication. In *IEEE Infocom*, pages 122–127, Las Vegas, Nevada, USA, 1982.

[34] F. Tobagi and V. Hunt. Performance analysis of carrier sense multiple access with collision detection. In *Computer Networks, Vol. 4*, pages 245–259, October/November 1980.

[35] G. Trent and M. Sake. WebSTONE: The first generation in HTTP server benchmarking. http://www.mindcraft.com/WebStone/, Feb. 1995.

[36] L. Tsui and O. M. Ulgen. On modeling local area networks. In *SCS Winter Simulation Conference*, pages 842–849, 1988.

[37] J. Wang and S. Keshav. Efficient and accurate ethernet simulation. In *Proceedings on 24th International Conference in Local Area Networks*, pages 182–191, Oct. 1999.

[38] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the First ACM SenSys Conference*, page to appear, Los Angeles, CA, USA, Nov. 2003. ACM.