NASA Technical Memorandum 101056

Queuing Theory Models for Computer Networks

David C. Galant
Ames Research Center, Moffett Field, California

February 1989

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

SUMMARY

A set of simple queuing theory models which can model the average response of a network of computers to a given traffic load has been implemented using a spreadsheet. The impact of variations in traffic patterns and intensities, channel capacities, and message protocols can be assessed using them because of the lack of fine detail in the network traffic rates, traffic patterns, and the hardware used to implement the networks. A sample use of the models applied to a realistic problem is included in appendix A. Appendix B provides a glossary of terms used in this paper.

The Ames Research Center computer communication network is an evolving network of local area networks (LANs) connected via gateways and high-speed backbone communication channels. Intelligent planning of expansion and improvement requires understanding the behavior of the individual LANs as well as the collection of networks as a whole. Questions that need answers include:

- what is the response time of the network over every path?

- what are the effects of traffic pattern changes on the response and capacity of the network?

- what is the impact of changing one or more of the facilities?

- given a specified response (or more generally turnaround) time, what facilities are required to achieve it?

INTRODUCTION

The following properties characterize the kinds of networks that are considered and the behavior that can be studied by queuing theory.

- **Flow balance**—the number of job arrivals to the network during a time interval equals the number of jobs serviced by the network during the interval.

- **Homogeneity**—routing and servicing of a job at a node is independent of the local queue lengths and the mean time between service completions at a given node must not depend on the queue lengths of other nodes. This assumption appears shaky because a packet passing from one queue to another certainly affects the latter strongly because a packet has a fixed length and requires a fixed (and predetermined) amount of service when it enters a queue. Nonetheless, homogeneity is observed in practice. Because gateways service several links in the network and must respond to the traffic of these queues, its response usually is randomized. This gives the assumption operational validity.

- **Fair share service**—at a node service to a packet is either on a first-come, first-served basis or at random. The same is assumed true within a priority class in a network with access based upon a nonpreemptive, class-priority scheme and service is first-come first-served within a priority class.

- **Sufficiency of resources**—all nodes have infinite buffers and all resources necessary for service are instantly available. There are neither model effects for bottlenecks at a node nor is packet retransmission ever necessary.

- **Poisson arrivals**—the interarrival times of jobs are random and have exponential distribution. For these models, this is a standard assumption.

- **Persistence**—every node that has a packet to transmit is persistent in its effort to acquire service and transmits as soon as service is available (some arbitration scheme is required to break ties).

- **Loopless flow**—no job requires service at a node more than once.

- **Open network**—once a job has arrived at its destination, it disappears from the network.

- **Reliability**—hardware never fails. These effects are not part of the model

No assumptions about the service time distribution have been made, nor are they necessary; queuing theory is sufficiently developed to handle general service distributions.

Because the network is open and loopless, the end-to-end delay for a packet is simply the sum of the individual delays in each subnetwork and the various gateway systems. In other words, the network decomposes completely, allowing us to simply model pieces in relative isolation. The only effect of one piece of the network upon another is the induced traffic load.

Predicting the behavior of a network requires knowledge of the traffic load. The size distribution of packets, the rate of packet transmission, and the service time required to send a packet are sufficient to model the **average** behavior of the network under the given traffic load.

Once the isolated behavior of the individual links in the network and the flow of traffic through the network are known, the transmission delay between two nodes for a file many packets long can be calculated.

TERMINOLOGY

Parameters used to characterize the bulk behavior of the network links are

- U—the fraction of the time during which the network is busy. This is called the **utilization.**

- Q—the average length of the queue of packets awaiting transmission.

- R—the **response time** (seconds) of the network. This is the end to end delay time for a packet. It includes both the waiting time and the service time for the packet.

In the following, the subscript a on a symbol denotes its average value. The numbers that we need to derive from the work load are

- J_a, the **average packet size** (bytes)

- S^2, the **second moment** of the packet size distribution
- A, the **arrival rate** of packets (packets/sec)

The information about the communication channel required is

- C, the **channel capacity**, (bytes/sec)
- F, the **fixed overhead time** for channel usage (seconds)

S, the **service time**, for a packet L bytes long is

$$S = T + F$$

where T, the **transmission time** for the L byte packet, is

$$T = L / C$$

Finally, derived data that are of interest to both an engineer and the user of a network are

- **Effective transfer rate** which is the capacity of the cable with adjustment for the overhead time for transmission

$$C\, T_a / S_a$$

- The **actual traffic** over the network (number of bits transmitted)

$$A\, J_a$$

- The **average waiting time** for network access, W_a (seconds)
- The **marginal transfer rate**

$$C\ S_a / (S_a + W_a)$$

This marginal rate is the perceived transfer rate of a newly arrived packet. In other words, it is the transfer rate of data diluted by the time waiting for the channel to become available.

STANDARD EQUATIONS FOR THE MODELS

The fundamental interrelationships among the variables defined above can be found in reference 1 and are included here to make this report self-contained. For specific networks, we only need to apply the details of access protocol and other overheads to have a model.

We assume the work load is a mixture of packets of sizes $s_1, s_2, \ldots, s_n$ bytes, with relative frequencies of occurrence $f_1, f_2, \ldots, f_n$ with

$$\sum_{i=1}^{n} f_i = 1$$

and

$$J_a = \sum_{i=1}^{n} f_i s_i$$

The **average service time** is given by

$$S_a = (J_a \ / C + F)$$

and the second moment of the service time is given by

$$S^2 = \sum_{i=1}^{n} f_i \left(s_i/C + F\right)^2$$

$$= \frac{1}{C^2} \sum_{i=1}^{n} (f_i s_i^2) + 2 F J_a / C + F^2$$

The utilization U is given by

$$U = A S_a$$

The average length of the queue is

$$q_a = U + \frac{A^2 S^2}{2(1 - U)}$$

the **average time in the system** is

$$R = (q_a / A) + \text{any other fixed overheads}$$

and the **average waiting time** is

$$W_a = \frac{(q_a - U)}{A}$$

These formulas model the average behavior of a single server queuing system. In particular, q_a is a measure of the average congestion.

SOME REAL NETWORKS

Some common communication protocols are token ring networks, slotted token ring networks, and the Ethernet. We will examine only those effects associated with the use of the network and explicitly ignore effects like queuing within a node, as occurs at a gateway, and the cost of packaging a packet for transmission which will be considered later. The descriptions below give more specific details about the variables in the standard equations in the specific network types.

Token Ring Network In a token ring, the simplest network to model, access to the transmission channel is controlled by passing a permission token around the ring. A free token travels around the ring until a node, ready to transmit, changes the token to busy and puts its packet onto the channel. The packet can, in principle, be of arbitrary length. The sending station is responsible for removing its own packet from the ring. At the end of its transmission, it passes the access permission to the next station by generating a new free token. The service time of a packet is modeled by the transmission time plus a fixed overhead for passing the token plus additional latency within a node for token handling, etc.

This model allows one node to acquire the transmission network for an indefinite period of time. Such behavior is not particularly desirable and, in practice, a maximum packet size is enforced. Any message longer than that is broken into an appropriate number of packets of the largest size. There is a minimum packet size as well. The transmission time for a packet of length L is

$$T = (L+H) / C$$

where H is the length of the header. In the standard equations for this kind of network, the fixed overhead time term, F, for a packet is the time to acquire the token, which is, on the average, half the time for a token to traverse a ring of idle nodes.

A slotted token ring has a constant number of fixed length slots continually circulating around the ring. This network acts as if the token is passed at fixed time intervals. Hence, the transmission time is not part of the service time and the service time for any packet is constant.

As an example of token ring behavior, we consider a Proteon-like ring (ref. 2). This network uses an 80-megabit/sec transmission channel to which access is controlled by passing a token. A node with a packet to transmit acquires the token and begins to send its packet. The maximum packet size is 4000 bytes. The overhead for sending a packet is 11 bytes plus a small amount of variable overhead for synchronization purposes which we shall assume is 2 bytes for a total of 13 bytes. Assuming that each node delays a packet it does not read, but instead passes it on to the next is 1 bit time and the delay of 5.77 μsec/ km of cable, the average transmission time for a packet J bytes long on a 2-km ring with N stations attached is

$$(J + 13) / 10 + 11.54 + N / 160 \ \mu\text{sec}$$

of which the last two terms are fixed overheads. A byte is 8 bits.

Sending only the largest packets on a network of 10 nodes connected to 2 km of cable, almost 70 million user bits/sec can be transmitted. However, one must be aware that, in practice, the largest packets will seldom be sent because a packet coming from an Ethernet (see below) is, at most, 1518 bytes long and will be transmitted "as is." Furthermore, a machine running a POSIX-like operating system is more likely to choose 2048 (plus some overhead for the transmission protocol) byte packets to send. Transmitting these will degrade the effective capacity of the channel.

Another shortcoming of this simple model is that it does not account for the capacity of the channels connecting nodes to the network. They typically do not exceed 24 million bits/sec for a minicomputer or 8 million bits/sec for workstations. This creates a mismatch of capacities. The nodes are unable to absorb data at the rate at which the network can send them and the practical capacity of the network may be severely degraded.

The Ethernet Network The Ethernet is a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol network. It consists of a broadcast channel whose capacity is specified to be 10 MHz (1,250,000 byte/sec) and a protocol for a node to access the channel. Access to the channel is asynchronous and random.

When a terminal has a packet to send, it monitors the state of the cable. If it is busy, the node defers transmission until the channel becomes quiescent for a period of time and then begins to transmit its packet. Because of the finite speed of transmission, two nodes can start to transmit simultaneously and mutually corrupt one another's packets. This is called a collision. When a collision occurs, a "jam" signal is sent and transmission by the nodes is aborted. To prevent another collision, each node involved is set to back off for a random amount of time before beginning to retransmit.

The smallest packet that can be transmitted is 46 bytes (368 bits) of data (including the surrounding frame, this is 512 bits); a smaller packet will be padded to this length, if necessary. The largest packet size is 1500 bytes. These sizes are for user data and do not include the data added by the network protocol.

The service time for a packet consists of three elements, transmission time, fixed overheads for transmission imposed by the details of the protocol, and the additional overhead imposed by collision resolution. To each packet transmitted, an 18-byte frame is added that includes the source and destination addresses of the packet and a cyclical redundancy check. Also, an 8-byte synchronization signal is sent immediately before the packet is transmitted. This is equivalent to a 26-byte transmission overhead so the transmission time of a packet L bytes long is

$$T = (L + 26) \times 0.8\ \mu\text{sec}$$

The fixed overheads for propagation delay in the physical channel are detailed in the Ethernet reference document (ref. 3) and, for a cable 1.5 km long, these sum to 46.38 μsec. Also required is an obligatory listening time of a node to the inactive channel before a packet may be transmitted.

An additional burden to the Ethernet service time of a packet is the effect of collisions. The Ethernet standard specifies the time to resolve a collision as 97.58 μsec. The burden on a packet-by-packet basis is the average collision resolution time. If N is the probability of a collision, the service time for a packet L bytes long is

$$(L + 26) \times 0.8 + NT \ \mu\text{sec}$$

The probability of a collision is derived below.

Some comments on the capacity of an Ethernet are appropriate. Its theoretical capacity is 10 million bits/sec, but a considerable debate exists over the actual capacity of an Ethernet. Some estimate actual capacity as low as 3 million bits/sec, while others argue for numbers much nearer the theoretical capacity. Under the proper conditions, the actual capacity is remarkably close to the theoretical capacity. If only one node transmits packets of maximum size, the capacity of an Ethernet is 9.6 million bits/sec; more than 9.4 million of them are data.

The following table shows the actual maximum capacity versus the number of transmitting nodes when only the largest (1500-byte) packets are sent.

Number of nodes	Maximum capacity, Mbits/sec	Data Mbits/sec
1	9.60	9.4
2	9.36	9.2
3	9.16	9.0
5	9.05	8.9
20	8.85	8.7
infinity	8.65	8.5

However, the pessimists must be given their due. If nodes transmit only the smallest (46 byte) packets, the capacity of the channel is drastically reduced as is seen in:

Number of nodes	Maximum capacity, Mbits/sec	Data Mbits/sec
1	4.8	3.4
2	3.9	2.8
5	2.5	3.5
20	3.2	2.3
infinity	3.2	2.3

The actual maximum capacity under mixed traffic is clearly somewhere between these two widely separated extremes.

The Hyperchannel Network This is a high-speed network designed to support communication among large computers of disparate architectures. Each computer is connected to the hyperchannel by a device called a network adapter that can be connected to as many as four channels. Each channel has bandwidth of 50 million bits/sec. The adapters, connected to a communication channel of the host computer, act as the channel nodes. Access is asynchronous, but collisions are almost completely eliminated by a slotting scheme that acts as a strict priority access scheme when packets at two nodes await transmission.

The largest packet has 65,536 bytes which is broken up into 16 equally sized subpackets. The smallest packet is a single subpacket. The service time for a packet is somewhat complicated. It depends upon

- the rate at which the sending host can transmit data to the adapter
- the rate at which the receiving host can accept data from its network adapter
- the length of time needed to get the attention of the receiving host computer

In symbols,

T = maximum (receiver delay time, s / sender channel rate)
+ H(k-1) s / receiver channel rate
+ (k-2)H(k-2)4096 / minimum (sender channel rate, receiver channel rate)
+ (492 + 266k)/cable capacity

where

s = minimum (4096, packet length),
k = number of 4096-byte subpackets in the packet,

$$H(k) = \begin{cases} 0 \text{ if } k < 0 \\ 1 \text{ if } k \geq 0 \end{cases}$$

We assume that each host computer has as many network adapters as there are available cables. To assume otherwise makes queuing theory models of the average behavior impossible because the objects in the system that provide service are distinguishable. However, with this assumption, a hyperchannel network can be modeled as a queuing system to which access is asynchronous, nonpreemptive, strict priority, and multiserver; the theory of this kind of queuing systems can be found in reference 1.

GLUING NETWORKS TOGETHER

Every node in the network must be able to send information to any other node, so the local area networks (LANs) must be interconnected. Intermediary devices, called bridges and gateways, do this. Bridges connect logically separate networks of the same kind, such as two Ethernet networks. Gateways connect networks of dissimilar kind, for example, an Ethernet network and a token ring network.

A bridge acts as a surrogate receiver and then a surrogate transmitter for communication between nodes on interconnected networks. Typically, it adds no overhead, or at least very little.

A gateway is a far more elaborate device. It must reformat packets for transmission over the network with the receiving node, and then send the new set of packets over the next network. A gateway is, of necessity, a computer of some sort and the transmission delay it causes to a packet is difficult to determine in detail. For the purposes of this report, we will assume that, on the average,

– there is a fixed overhead of F seconds

– I machine instructions are necessary to receive, process, and transmit 1 byte of information

– packets are retransmitted in random order

Thus a packet of length L is delayed by

$$I\,(L + L_a\,K)\,/\,M + F \text{ sec}$$

where M is the instruction rate of the gateway, L_a is the average length of an enqueued packet, and K is the average number of packets enqueued within the gateway. This estimate for the delay is crude, but is probably effective with a reasonable choice of I. The advantage of the estimate is relating the delay within a gateway to the speed of the machine. The average number of packets enqueued at a gateway is the sum of all incoming traffic plus all of the outgoing traffic and the service time scales with the speed of the gateway.

A detail that has not been considered, except for the hyperchannel, is the delay incurred because of lower capacities of the channels connecting the nodes to the actual computers. This is done from a lack of specification in both how the network connects to these channels and the capacities of these channels. In any case, the inclusion of a term like the one in the calculation of the time for transmission of a hyperchannel packet should be considered to approximate the effects of this capacity mismatch when the information becomes available.

MODELING THE NETWORK TRAFFIC

One of the major concerns in modeling the response of a network is describing the work load imposed upon it. This is more difficult when traffic flows from one kind of network to another through a gateway because there is a tendency, in studying a LAN, to examine the packets flowing through a network as if they existed without a context. Much of the traffic that flows through the network we are concerned with comes from packaging a file into packets because of the size limitations on a packet imposed by the network protocol. This means that a file can suffer the LAN delays for each packet that constitutes part of the file and the gateway cost of repackaging the file for transmission over the next LAN.

The problem of estimating end-to-end delays for transmitting a packet involves not just the nature of each segment of its path, but the underlying packet size distributions as well. To estimate the total delay within each link of the network, we must know the total delay for a file in each link of the transmission path. Because we know the maximum size of a packet for each link, these calculations are direct. If the *ith* link of the transmission path has a maximum packet length of L_i bytes and a delay per packet of D_i seconds, and the gateway delay for the *jth* gateway computer for processing a file is G_j seconds then the end-to-end delay of a packet B bytes long over the path is

$$E(B, \text{path}) = \sum_{i \in \text{path}} \left\lceil \frac{B}{L_i} \right\rceil D_i + \sum_{j \in \text{path}} G_j(B)$$

where the stylized square brackets denote the first integer larger than the argument.

Knowledge of the file size distribution and traffic intensity between nodes, makes possible calculation of the average end-to-end delay for the entire network. Since network response can be roughly related to traffic intensity and equipment capability (and both relate to the network cost), the end-to-end delays can be used to drive cost analyses of equipment needed to meet given performance requirements over any path in the network.

COPING WITH COLLISIONS IN AN ETHERNET

Calculating the probability of a collision on an Ethernet under the assumptions that all of the terminals generate the same traffic is direct.

A collision can occur only when a packet departing a queue leaves two or more packets waiting in the queue. The (mutually exclusive) events that lead to this are

– a packet enters service leaving an empty queue behind it, and during service two or more packets arrive.

– a packet enters service leaving exactly one awaiting service, and during the service time one or more packets arrive.

– a packet enters service with two or more packets awaiting service behind it.

Letting P[V] denote the probability that the event V occurs, q denote the length of the queue of remaining customers, and n denote the number of arrivals during an average service period

$$P[\text{collision}] = \{P[q=0]P[n\geq 2] + P[q=1]P[n\geq 1] + P[q\geq 2]\}$$

From elementary probability theory,

$$\begin{aligned} P[n\geq 1] &= 1 - P[n=0] \\ P[n\geq 2] &= 1 - P[n=0] - P[n=1] \end{aligned}$$

reference 1 gives the results

$$\begin{aligned} P[q=0] &= 1 - U \\ P[q=1] &= 1/B^*(A) \end{aligned}$$

where B*(A) is the Laplace transform of the service time probability density function. Because the arrival of customers is a Poisson process,

$$\begin{aligned} P[n=0] &= \exp(-U) \\ P[n=1] &= U \exp(-U) \end{aligned}$$

Substitution of these into the formula for the probability of a collision gives

$$\begin{aligned} P[\text{collision}] = 1 + (1 - U)\,(1 - (1 + U)\exp(-U) \\ + (1 - \exp(-U)(1/B^*(A) - 1) - 1/B^*(A)) \end{aligned}$$

A somewhat similar model which treats collisions in a different and less flexible fashion was proposed by Lam (ref. 4).

Accounting for Effects of Traffic Asymmetry Models for collisions in CSMA/CD networks, including ours, assume complete equivalence of the nodes in the network. This assumption is certainly false for a network of dumb terminals connected to a terminal server or a system of terminals connected to a file server. Traffic is highly asymmetrical in these situations because one node generates most of the traffic. Also, any protocol guarantees that traffic from a single source cannot collide with itself. When the system consists of terminals and a file server, the asymmetry is even greater because the file server generates a far greater portion of the traffic. In either of these two situations, the terminals do not, by and large, communicate among themselves, but almost always communicate with the server. Because of these asymmetries, the probability of a collision is greatly overestimated.

Yet another major traffic asymmetry is found in a network of smart terminals. Usually, they do not send a packet, but a file so large that it must be transmitted in pieces that show up on the network as packets. That is, the traffic from any given node on a network is "bursty," many packets at once interspersed with long periods of silence with little contention taking place.

To account partially for these asymmetries, we simply replace U with $U'=(1 - b)U$, $0 \leq b \leq 1$ in the formula for the probability of a collision. The parameter, b, deflates the probability of a collision by removing the fraction of the traffic that cannot cause collisions. For instance, for a system of dumb terminals with a terminal server, b certainly exceeds 0.5 because half the traffic originates with the server.

Appendix A. A sample traffic analysis

The models discussed in general terms in this paper were implemented on an Apple Macintosh™ computer using the spreadsheet program Trapeze 2.0™. To show how the models are used, an analysis of a network is done. We use a hypothethical network consisting of (fig A-1) five` Ethernets, each connected to the same Proteon backbone by a five-mips processor and a supercomputer complex, consisting of a mass storage system (MSS) and a high speed computer (HSC), internally connected by a two-cable hyperchannel which is also connected to the Proteon backbone by a five-mips processor.

On each of the Ethernets, there are two nodes that during each hour randomly transmit a 65,536-byte file over the network to the supercomputer complex. The job this file represents causes a 335,544,320-byte file to be transmitted over the hyperchannel from a mass storage system to the supercomputer. Five minutes of waiting and 5 min of supercomputer processor time are consumed in some computation, and 3,145,728 bytes of data are then transmitted over the network from the supercomputer to the originating node. Given a level of background traffic in the network, some questions are

- How much time passes from the start of the transmission of the first file to the completion of the transmission of the result file?

- Is there sufficient capacity to support the traffic induced by the load?

- How much "think" time is available between transmissions (the time from the end of the returned file transmission to the end of the hour)?

We begin by considering the traffic over the various subnetworks.

Hyperchannel Traffic

Within the supercomputer complex each job generates

- 5120 65,536-byte packets transferred from the MSS to the HSC
- 1 65,536-byte packet from the front end to the HSC
- 48 65,536-byte packets to be sent from the HSC to the front end

Backbone Traffic

We must make some assumptions about how the gateways connecting the various subnetworks operate. Assume that they have a POSIX-like operating system. It is very likely that large files will be broken into packet lengths that are a power of two. The largest such multiple on the backbone is 2048 bytes in length. Under this assumption, each job generates

– 1536	2048-byte packets
– 1536	64-byte packets (for acks)
– 43	1500-byte packets
– 1	1036-byte packet

on the backbone.

Ethernet Traffic

Each job generates

– 1536	1500-byte packets
– 1536	548-byte packets
– 1536	64-byte packets (acks for the above)
– 43	1500-byte packets
– 1	1036-byte packets
– 44	64-byte packets

on the Ethernet when we assume that the gateway breaks a backbone packet into larger than 1500 bytes into a series of 1500-byte packets plus a smaller packet for the remainder (hence the 548-byte packets).

In addition to all of the traffic above, we assume background traffic distributed as given in the Ames study (Grams, NASA, unpublished report) using 10% of the Ethernet capacity. We also assume that the job is monitored by the user while it is in the supercomputer complex. This causes a 1000-byte packet to be sent from the supercomputer complex to the user each second. The receipt of this packet is acknowledged by the receiver. The traffic per job this causes is

– 1	1000-byte packet/sec on the Ethernet
– 1	1000-byte packet/sec on the backbone
– 1	64-byte packet/sec on the Ethernet
– 1	64-byte packet/sec on the backbone

The standard mix of background jobs on the Ethernet is

Ethernet traffic	
Packet size	**Fraction of traffic**
1500	0.210625
1000	0.275679
600	0.017864
64	0.495828

and on the backbone, the standard traffic mix is

Backbone traffic	
Packet size	**Fraction of traffic**
1500	0.007500
1000	0.526428
600	0.040835
64	0.425236

This is the pattern of traffic activity on the network, neglecting the traffic induced by sending the the two hourly jobs from each Ethernet. Assuming that the number of background jobs uses 10% of the Ethernet capacity, we have the number of packets transmitted per second as 173 (fig. A-2). Part of the traffic analysis in Grams' report is that 35% of the Ethernet traffic is off net, so with five Ethernets, we have 302.75 jobs generated on the backbone each second. This is 2% of the backbone capacity (fig. A-3.) In terms of packets, the background traffic is

Ethernet traffic	
Packet size	Number of packets
1500	36.438125
1000	47.692467
600	3.090472
64	85.778244

Backbone traffic	
Packet size	Number of packets
1500	2.270625
1000	159.37608
600	12.37608
64	128.74019

For the background on the hyperchannel (fig. A-4), we use the model numbers from a NAS study (Bruce Young, personal communication). For the loaded systém (fig. A-5), we find the total average delay in the Hyperchannel network is 2 min, 17.5 sec.

Since we assume that once a job enters the supercomputer complex, it waits 5 min and spends 5 min in execution, the total time in the supercomputer complex is 12 min, 17.5 sec. In that time, 737.5 X-window packets are sent through the network by each of the 10 jobs: a total of 7375 1000-byte packets and 7375 64-byte packets each hour, or an average of 2.0486/sec of each. We can now calculate the increase in the number of packets on each of the backbone and the Ethernet networks.

We are going to assume that all of the Ethernet traffic induced by the jobs sent to the supercomputer complex are dumped onto the network in a period of 5 min (rather than in 1 hr) to examine the network response under a high load condition that would likely occur from the bursty behavior of the load. The traffic load from these assumptions and from the assumption of the X-window traffic and the assumed background load is

Loaded Ethernet traffic	
Packet size	Number of packets
64	98.3601733
548	10.2400000
600	3.0904700
1000	59.9810600
1036	0.0066667
1500	46.9647867

The loaded Ethernet response to this load is found in figure A-6. Notice that the collision deflator, the parameter used to remove asymmetric traffic effects, is set to 0.2. Because 35% of the traffic originates from the gateway, such a number is certainly reasonable to assume.

The number of packets sent over the Ethernet for one job is 6171 (including the X-window traffic). With an Ethernet average response time of 0.75 msec per packet, the time spent on the Ethernet is about 4.6 sec. Even if the traffic intensity is quintupled, the delay is only about 11 sec.

On the backbone, under the same assumptions upon the arrival rate of packets, the loaded traffic is

Loaded Backbone Traffic	
Packet size	Number of Packets
64	181.988580
600	12.368000
1000	161.424676
1036	0.033333
1500	3.703958
2048	51.200000

The response of the backbone to this traffic load is given in figure A-7. From these results, and the fact that each job causes 3116 packets to be sent over the backbone, we infer that the delay from this part of the network is 0.3 sec. This remains true, even if the intensity of traffic is quadrupled.

As for the gateways, one must speculate. Assume that there is a 10-msec delay in the gateways for overhead (like fielding the interrupt, looking up addresses, etc.) and that one byte is transferred for each instruction, then the delay in the gateways for each job is about 65 sec almost all of which is caused by the length of the overhead delay.

From the above analysis we draw the following conclusions.

- At the traffic intensity internal to the supercomputer complex, assumed, the delays in the Ethernet and backbone networks are negligible, even at considerably higher traffic intensities within the latter two subnetworks.

- The delay from the beginning of one of the jobs considered to the end of the transmission of the results is between 13 and 14 min, more than 12 of which are spent within the supercomputer complex.

- The most significant delay is internal to the supercomputer complex, and the next most significant delay is in the gateways.

- The "thinking time" in a job is more than 45 min.

Note that the X-window protocol, which is something of a standard, is designed to operate with a 20- to 50-msec delay. Under the assumptions we have used, this time frame is easily met.

Appendix B. Glossary

The terms included in this glossary are taken from reference 5.

Collision: The event when two nodes in a network try to transmit at the same time on the same channel, causing the transmitted data to be useless.

Congestion: A condition when the demands on the network exceed the network's capacity to handle them within a particular time. The condition that arises when arriving traffic exceeds the capability of servers handling the traffic.

CSMA/CD: Carrier Sense Multiple Access with Collision Detection, a method of having multiple stations access a transmission medium (multiple access) by listening until no signals are detected (carrier sense), then transmitting and checking to see if more than one signal is present (collision detection).

Gateway servers: Communication servers that provide access from one type of network to another (networks having different access protocols).

Network: A group of computers interconnected to share resources and information.

Node: Any intelligent device that communicates with other devices in the network. A point in the network where service is provided or used, or communication channels are interconnected.

Packet: A group of bits, including user data and control data, that is transferred as a unit through a network.

Path: A possible route for a unit of data from a source node to a destination node. The path can consist of a sequence of connected nodes between the source and destination.

Propagation time: Part of the transmission time; it is the time necessary for a signal to travel from one point on a circuit, or network, to another. Its starting point is when the first byte of a message enters the transmission medium and its ending point is when the first byte of a message reaches its destination.

Protocol: A basic procedure or set of rules that controls communication between computers. Also, a set of conventions between communicating processes regarding the format and contents of messages to be exchanged. The rules of "etiquette" determine the behavior that the communicating devices can expect of each other.

Response time: The time it takes a system or network to respond to a given input; the interval between an event and the response to that event.

Ring: A network topology in which nodes are connected to one another in a closed, logical circle.

Token Passing: A scheme used by networks for controlling access to the network. Usually used in ring networks. Permission to use the network is given in the form of a token that is passed from node to node around a ring. When a node has a message to send, it grabs the token. Possession of the token gives the node exclusive access to the network. When the node finishes transmitting, it reinserts the token into the ring so other nodes may have a turn.

Transmission time: The time required to transmit a unit of data from the source node to the destination node. Its starting point is when the first byte of a message enters the network medium and its ending point is when the last byte of that packet reaches its destination.

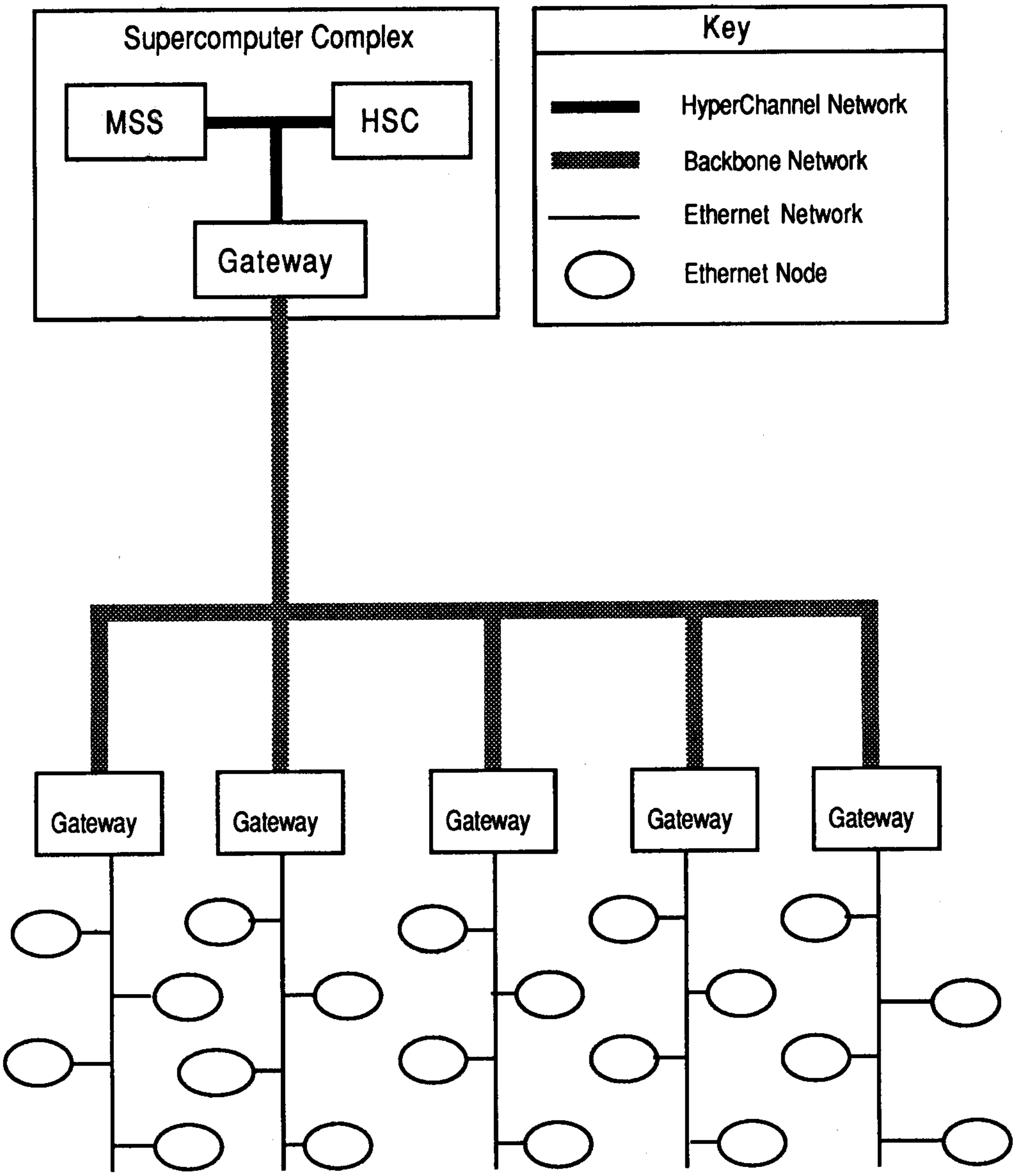

Figure A-1. Conceptual network.

Workload Characteristics

Note : These sizes do not include the 18-byte frame that is attached to each packet.

Traffic	
Packet Size	Percent
64	49.58
600	1.79
1000	27.57
1500	21.06

Cable	
Capacity (KBytes)	1250
Collision Vulnerability Time (msec)	0.01
Collision Resolution Time (msec)	0.0972

Workstation/Host Traffic		
Number Workstations	Collision Deflator	Average Input Rate (jobs/sec)
1	0	173
		174
		175
		176
		177
		178
		179
		180

Statistics	
Mean Job Time (sec)	Mean Job Size (bytes)
5.781E-4	634.07

Predictions

Packet arrival rate (per sec)	Channel utilization	Response time, msec	Average number of packets enqueued	Average collisions per packet	Effective transfer, Mbits/sec	Actual transfer (Mbits/s)
173.00	10.00%	0.69	0.19	0.02	8.74	0.88
174.00	10.06%	0.69	0.19	0.02	8.74	0.88
175.00	10.12%	0.69	0.20	0.02	8.74	0.89
176.00	10.17%	0.69	0.20	0.02	8.74	0.89
177.00	10.23%	0.70	0.20	0.02	8.74	0.90
178.00	10.29%	0.70	0.20	0.02	8.74	0.90
179.00	10.35%	0.70	0.20	0.02	8.74	0.91
180.00	10.41%	0.70	0.20	0.02	8.74	0.91

Figure A-2. Ethernet background traffic.

Workload Characteristics

Note: These sizes do not include the 13-byte frame

Traffic Mix	
Packet Size	Percent
64	42.52
600	4.08
1000	52.64
1500	0.75

Cable	
Capacity (KBytes)	10000
Propagation Delay (msec)	0.01154

Traffic Rates	
Number Nodes	Average Input Rate (jobs/sec)
1	300
	302.75
	305
	325
	450
	600
	1200
	1800

Statistics	
Mean Job Time (sec)	Mean Job Size (bytes)
6.601E-5	589.39

Predictions

Packet arrival rate (per sec)	Channel utilization	Response time, msec	Average number of packets enqueued	Effective transfer Mbits/sec	Actual transfer Mbits/sec
300.00	1.98%	0.08	1.19	71.43	1.41
302.75	2.00%	0.08	1.19		1.43
305.00	2.01%	0.08	1.19		1.44
325.00	2.15%	0.08	1.19		1.53
450.00	2.97%	0.08	1.20		2.12
600.00	3.96%	0.08	1.20		2.83
1200.00	7.92%	0.08	1.24		5.66
1800.00	11.88%	0.08	1.27		8.49

Figure A-3. Backbone background traffic.

Workload Characteristics

Host	Delay (msec)	Channel Rate (MB)
WKS	25.00	1.250
HSS	6.67	3.125
MSS	6.67	3.125
HSC	0.25	6.250

Raw Arrival Rates (/h)

	WKS	HSS	MSS	HSC
WKS	0	271	300	75
HSS	58	0	0	5744
MSS	634	0	0	31046
HSC	661	661	2566	0

Cable Parameters

Number Cables	Capacity (MB/sec)
2	6.25

Mean Queue Length

WKS	0.0102
HSS	0.0390
MSS	0.2148
HSC	0.0402
System	0.3042

Mean Sojourn Time (sec)

WKS	0.0568
HSS	0.0242
MSS	0.0244
HSC	0.0372
System	0.0260

Figure A-4. Hyperchannel background traffic.

Workload Characteristics

Host	Delay (msec)	Channel Rate (MB)
WKS	25.00	1.250
HSS	6.67	3.125
MSS	6.67	3.125
HSC	0.25	6.250

Raw Arrival Rates (/h)

WKS	0	271	300	75
HSS	58	0	0	5754
MSS	634	0	0	82246
HSC	661	1141	2566	0
	WKS	HSS	MSS	HSC

Cable Parameters

Number Cables	Capacity (MB/sec)
2	6.25

Mean Queue Length	
WKS	0.0110
HSS	0.0459
MSS	0.6091
HSC	0.0461
System	0.7121

Mean Sojourn Time (sec)	
WKS	0.0612
HSS	0.0284
MSS	0.0265
HSC	0.0380
System	0.0270

Figure A-5. Loaded hyperchannel traffic.

Workload Characteristics

Note : These sizes do not include the 18-byte frame that is attached to each packet.

Traffic	
Packet Size	Number
64	98.36
548	10.24
600	3.09
1000	59.98
1036	0.01
1500	46.96

Cable	
Capacity (KBytes)	1250
Propagation Delay (msec)	0.04
Collision Resolution Time (msec)	0.0972

Workstation/Host Traffic	
Number Workstations	Average Input Rate (jobs/sec)
1	210
Collision Deflator	218
0.2	250
	436
	654
	872
	900
	1000

Statistics	
Mean Job Time (sec)	Mean Job Size (bytes)
5.984E-4	659.50

Predictions

Packet arrival rate (per sec)	Channel utilization	Response time, msec	Average number of packets enqueued	Average collisions per packet	Effective transfer Mbits/sec	Actual transfer Mbits/sec
210.00	12.57%	0.75	0.24	0.02	8.79	1.11
218.00	13.05%	0.75	0.25	0.02	8.78	1.15
250.00	14.96%	0.78	0.29	0.03	8.77	1.32
436.00	26.09%	0.94	0.55	0.09	8.70	2.30
654.00	39.14%	1.18	0.92	0.18	8.57	3.45
872.00	52.18%	1.53	1.44	0.29	8.42	4.60
900.00	53.86%	1.59	1.53	0.30	8.40	4.75
1000.00	59.84%	1.83	1.90	0.36	8.33	5.28

Figure A-6. Loaded Ethernet network traffic.

Workload Characteristics

Note: These sizes do not include the 13-byte frame

Traffic Mix	
Packet Size	Number
64	181.99
600	12.37
1000	161.42
1036	0.03
1500	3.70
2048	51.20

Cable
Capacity (KBytes)
10000
Propagation Delay (msec)
0.01154

Traffic Rates	
Number Nodes	Average Input Rate (jobs/sec)
1	400
	411
	800
	1200
	1600
	2000
	2055
	2466

Statistics	
Mean Job Time (sec)	Mean Job Size (bytes)
7.791E-5	708.37

Predictions

Packet arrival rate (per sec)	Channel utilization	Response time, msec	Average number of packets enqueued	Effective transfer Mbits/sec	Actual transfer Mbits/sec
400.00	3.12%	0.09	1.18	72.74	2.27
411.00	3.20%	0.09	1.18		2.33
800.00	6.23%	0.09	1.20		4.53
1200.00	9.35%	0.10	1.24		6.80
1600.00	12.47%	0.10	1.27		9.07
2000.00	15.58%	0.10	1.31		11.33
2055.00	16.01%	0.10	1.31		11.65
2466.00	19.21%	0.11	1.35		13.97

Figure A-7. Loaded backbone traffic.

REFERENCES

1. Kleinrock, L.: Queueing Systems, 2 volumes. Wiley, Interscience Press, vol. 1, 1975, vol. 2, 1976

2. Salween, H.; Marshall, A. C.; and Salween, N. K.: Pronet, An 80 MBIT/S Token Ring for High-Speed LAN Applications, Proteon Corporation, 1985.

3. Digital Equipment Corporation, Intel Corporation, and Xerox Corporation, The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, AA-K759B-TK, November 1982.

4. Lam, S.: A Carrier Sense Multiple Access Protocol for Local Networks. Comput. Networks, vol. 4, 1980, pp. 245-259.

5. Digital Equipment Corporation: Introduction to Network Performance, 1987.

National Aeronautics and
Space Administration

Report Documentation Page

1. Report No. NASA TM-101056	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Queuing Theory Models for Computer Networks	5. Report Date February 1989
	6. Performing Organization Code
7. Author(s) David C. Galant	8. Performing Organization Report No. A-89013
	10. Work Unit No. 505-65
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: David C. Galant, Ames Research Center, MS 244-3, Moffett Field, CA 94035
(415) 694-6013 or FTS 464-6013

16. Abstract

A set of simple queuing theory models which can model the average response of a network of computers to a given traffic load has been implemented using a spreadsheet. The impact of variations in traffic patterns and intensities, channel capacities, and message protocols can be assessed using them because of the lack of fine detail in the network traffic rates, traffic patterns, and the hardware used to implement the networks. A sample use of the models applied to a realistic problem is included in appendix A. Appendix B provides a glossary of terms used in this paper.

This Ames Research Center computer communication network is an evolving network of local area networks (LANs) connected via gateways and high-speed backbone communication channels. Intelligent planning of expansion and improvement requires understanding the behavior of the individual LANs as well as the collection of networks as a whole.

17. Key Words (Suggested by Author(s)) Computer networks Queuing theory Network models Local area networks	18. Distribution Statement Unclassified-Unlimited Subject Category – 66

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 34	22. Price A03

For sale by the National Technical Information Service, Springfield, Virginia 22161