END TO END DELAYS IN A CATENET ENVIRONMENT

by

H.G. Perros
Center for Communications and Signal Processing
Computer Science Department
North Carolina State University

Serge Fdida*
Laboratoire MASI, Université Pierre et Marie Curie
4, Place Jussieu, 75252 Paris Cedex 05, France

Ulf Körner*
Dept. of Communication System, Lund Institute of Technology
Box 118, S-222 48 Lund, Sweden

Gerald Shapiro
Operation Research Program
North Carolina State University, Raleigh, N.C. 27895

October 1986

CCSP-TR-86/23

*Drs. Fdido and Körner visited the Center for Communications and Signal Processing and the Computer Science Department in the Summer of 1986.
END-TO-END DELAYS IN A CATENET ENVIRONMENT

ABSTRACT

This paper presents a model for a catenet environment. In particular, the model, which could be seen as a hierarchical model consisting of three levels of models, focuses on the end-to-end delay between two host computers connected to different LANs separated via gateways by a WAN. The model incorporates a basic flow control mechanism, standardized local area network behaviour, as well as gateway functions in terms of packet fragmentation and reassembly.

1. INTRODUCTION

Throughout the last five years, we have witnessed the computer network evolution grasping the field of local area covering. The initial investments, that have to be done to build up a local area network are comparably small and they often pay off whenever the number of connected computers is bigger than one. Today, many computers, even medium sized, are delivered with local area network hardware connections, as well as with communications software, whether the consumer has asked for it or not. The growth of local area networks, which a couple of years ago

\[\text{Supported by Rome Air Force Development Center through grant E-21-669-S8 George Institute of Technology}\]
was not as big as many had predicted, has now begun to become significant. In the majority of cases, local area networks, LAN, will be connected in some fashion to other local area networks as well as to wide area networks, WAN.

However, the rapid response times we may be used to when staying within our own local network, will be dramatically increased in most catenet (concatenation of networks) environments. File transfers as well as interactive communication tend to be tedious. End to end delays could in many cases be increased by a factor of 1000 or more not at least in those cases where heterogeneous networks are concatenated and protocol conversions have to take place in gateways between the different networks.

In order to increase the LAN efficiency, many LANs tend to be kept small, i.e. when the number of attached stations grows, "another bus" is installed, and a filtering bridge will connect the two parts. Thus, common LAN resources are frequently accessible via a bridge, which further increases the end to end delay.

In most cases, gateway performance contributes to a major part of the end to end delay. A filter bridge between two homogeneous LANs has often a throughput that is just a couple of percentage points less than those of the individual LANs. A gateway that couples two heterogeneous networks could often perform worse! This is not only due to the processing times of the gateway functions. The packet lengths in a WAN are usually much shorter than those of a LAN (at least for standardized CSMA/CD and Token Bus LANs). Packet fragmentation and reassembly are often major tasks for gateways. Those functions are of great importance to incorporate in a heterogeneous catenet end to end delay analysis. A general description of the issues involved in a catenet environment can be found in Cerf and Kirstein [1].

It is of great importance to have realistic models for an end to end delay analysis in such heterogeneous catenet environments. These models should incorporate the basic mechanisms that significantly contribute to the end to end delay, such as, end to end flow control mechanisms, protocol conversions, addressing functions in the gateways, and packet fragmentation and
reassembly. To the best of our knowledge, no analytic studies which incorporate all of these factors have been reported in the literature. However, some work has been done in this direction. Mitchell and Lide [2] reported on a hierarchical modeling technique for predicting the end to end delay in a single local area network. This study models the three lowest layers of the OSI protocol hierarchy, with multiple layers of flow control. These authors also report on other studies attacking similar problems. Bernard [3] examines the problem of end to end delay in a catenet environment, under the assumptions of no packet fragmentation. In this study the time to access the local area network is assumed to be exponentially distributed, and each station may have only one unacknowledged message at any time. Varakulsiripunth, et al [4] examine a catenet with a single gateway handling all inter-net traffic. The proportion of traffic from each net which must be rejected in order to achieve a desired throughput on each net is determined. Queuing theoretic results about flow-control systems can also be used in modeling a catenet environment. References to these results are in section 3 of this paper.

In this paper we present a model for a catenet environment. In particular, the model focuses on the end to end delay between two hosts connected to different networks within a catenet. The model incorporates a basic flow control mechanism, standardized local area network behaviour, as well as, gateway functions in terms of packet fragmentation and reassembly. In section 2, we describe the catenet environment under study. The model is given in section 3, and the performance measures obtained using this model are given in section 4. Finally, the conclusions are given in section 5.

2 THE CATANET ENVIRONMENT

In this paper we have focused on the end to end delay in a heterogeneous catenet environment, and especially on the communication between two host computers, each connected to a separate LAN. The two LANs, the first a Token Bus and the second an Ethernet, are connected via gateways to a WAN as shown in figure 1. In general, there are different catenets. The one
treated here could be seen as a typical concatenation of heterogeneous networks. The WAN, a packet switched network, provides a virtual circuit connection, which gives sequencing, error recovery, etc.

As we take neither application nor presentation layer functions into consideration, the host to host communication could be seen as a virtual connection at the transport layer, i.e. a TCP-type of protocol is implemented. This protocol incorporates a window-type flow control procedure between the peer transport protocols in the two hosts.

After a transport set-up procedure, transport frames are delivered from a CEP (Connection End Point) seen by the transport protocol at the sending host to a CEP seen by the peer transport protocol at the receiving host. The transport layer at the sending host uses services provided by the IP-protocol in the same host, a protocol that finds its peer at the gateway in front of the WAN. The information exchange between those two peer IP-protocols relies on services delivered by the 802.2 LLC-protocol (Logical Link Control), which in turn relies on the 802.4 MAC-protocol (Media Access Control). Figure 2 shows the total layered architecture. The encapsulation process is schematically shown in figure 3. Transport frames, i.e. "user data" and a transport header (TH), are exchanged between entities in the peer protocols in the two hosts. At the sending host, a transport frame is passed as a unit to layer 3 and its IP-protocol. This protocol treats the whole unit as data, and appends its own header (a second encapsulation). This IP-frame is then
delivered to the LLC and is seen by the latter as just data. The LLC appends its small header (another encapsulation) and passes it to the MAC, which appends both a header and a trailer. This last frame represents the stream of data that is actually forwarded on the communication channel, though the Physical layer could use different techniques for coding.

3 THE MODEL

3.1 Some Basic Assumptions

In order to develop a somewhat general model, which could be applicable to other catenet environments, we have, where possible, made assumptions that mirror standardized system behaviour regarding maximum network packet lengths, channel capacities, media access principles and packet fragmentation and reassembly.
Let us assume that transport packets are generated at the sending host according to a Poisson process. These packets, consisting of a transport header of constant length and "user data", vary in length. We have assumed a distribution for the user data part, that could be described by the sum of two stochastic variables. The first is a truncated exponential distribution with mean $\frac{1}{\mu_1}$ equal to 100 bytes, the second one is deterministic with $\frac{1}{\mu_2}$ equal to 8000 bytes as shown in figure 4. This should mirror a mix of interactive and bulk data traffic. This packet distribution will be referred to as the "user data" packet length distribution. Let $b(x)$ be its pdf, then for $x \leq 1/\mu_2$

$$b(x) = \frac{\lambda_1}{\lambda_1 + \lambda_2}(\mu_1 e^{-\mu_1 x} + \delta(x - \frac{1}{\mu_2})e^{-\mu_1/\mu_2}) + \frac{\lambda_2}{\lambda_1 + \lambda_2}(\delta(x - \frac{1}{\mu_2}))$$  \hspace{1cm} (1)

where $\frac{\lambda_1}{\lambda_1 + \lambda_2}$ and $\frac{\lambda_2}{\lambda_1 + \lambda_2}$ describe the proportion of interactive packets and bulk data packets respectively.

In the model described in this paper, we study the end to end delay for packets generated at a host connected to a Token bus destined for a host connected to an Ethernet. Thus Token bus packets will consist of user data packets to which a header is added. This header represents a total overhead consisting of a transport- and an IP-header, the LLC-header, the MAC-header and trailer. For a 10 Mbit token bus the LLC- and the MAC-headers and trailer sums up to 26 bytes. This gives us a total header of around 90 bytes. For the Ethernet that has a smaller maximum packet size, user data packets to which transport-, IP- and LLC-headers have been added, are split up into smaller packets to which each a MAC-802.3 header is added. The same principle is used for WAN packets.
3.2 The Flow Control Model

The flow control problem has been studied by several authors, see Gerla and Kleinrock [5], Labetoulle and Pujolle [6], and Reiser [7],[8]. In this paper, we model the end to end flow control procedure, implemented as a window-type control, using the queueing network shown in figure 5. A transport packet arrives at the customer queue and it may pass if there is a token available in the token queue. If the token queue is empty, the packet is forced to wait until a token arrives. The total number of tokens is equal to C, the maximum allowable number of unacknowledged sent packets.

When a packet is allowed to pass, the packet and the token join to form a single customer, which enters the Communication Network. The number of tokens in the token queue is reduced by one. The communication network consists of all the communications resources between the peer protocols in the two hosts, that contribute to the end to end delay. That is, it consists of the lower layer protocols in the sending host, the token bus, a gateway, the WAN, another gateway, the Ethernet and finally the lower layer protocols at the receiving host.

When a customer leaves the communication network, it splits into the original transport packet and a token. The packet is delivered to a user application in the receiving host, and the token, carrying the acknowledgement signal, is sent back. In our model, we assume that the token is returned to the token queue through an acknowledgement channel. This channel is modeled as an infinite server. We assume that the service time of this station is exponentially distributed, with a mean of 300ms. In fact, we can suppose that the main response time component of a
The number of customers in the catenet plus the number of tokens in both the acknowledgement channel and the token queue is equal to \( C \), the maximum window size.

In figure 5, we used symbols commonly used in Petri Nets, in order to depict the join and the fork operations. In particular, the join symbol (figure 6) depicts the following operation. At the instance that each queue contains a customer, the two customers instantaneously depart from their respective queues and merge into a single customer. The fork symbol (figure 7) depicts the following operation. A customer arriving at this point (i.e. departing from the catenet) is split into two siblings.

The queue model described above can be seen as a queueing network in which some of its resources are shared by several customers. The resource sharing is managed by the token queue. The sharing of several resources is a phenomenon that arises commonly in computer and communications systems. This problem has been studied by several authors, see, for instance, Jacobson and Lazowska [9], Freund and Bexfield [10], Sauer [11], Smith and Browne [12], Thomasian [13] and Agrawal and Buzen [14]. For a detailed literature review see Fdida, Wilk and Perros [15].

An exact analytic solution of the above queueing network is rather difficult. Also techniques, relying on the numerical solution of the underlying rate matrix, are limited to small size problems. In view of this, we analyze this queueing network approximately using a decomposition method based on Norton's Theorem (see Chandy, Herzog, and Woo [16]).

Arrivals occur at the customer queue in a Poisson fashion at the rate \( \lambda \). We first analyze the communication network as a closed queueing network with \( n \) customers in it, \( n = 1, 2, \ldots, C \). Let \( \gamma(n) \) be the throughput of this system, for \( n = 1, 2, \ldots, C \). Now, the queueing network shown in
Figure 8: The reduced system

Figure 5 can be reduced approximately to the queueing system shown in Figure 8. The arrival process into the token queue is exponentially distributed with a rate \( \gamma(C - n) \), where \( n \) is the number of tokens in the token queue. This queueing system depicts the join operation described above.

Let \( p(i, j) \) be the probability that there are \( i \) customers in the customer queue and \( j \) tokens in the token queue. The rate diagram is as follows:

We note that this system is identical to an \( M/M/1 \) queue, with an arrival rate \( \lambda \) and a state-dependent service rate \( \gamma(n) \), if \( n \leq C \), and \( \gamma(C) \), if \( n \geq C \). We can easily establish that

\[
\begin{align*}
    p(i, 0) &= \rho^i p(0, 0) \\
    p(0, j) &= (\Pi(j)/\lambda^j) p(0, 0)
\end{align*}
\]  

(2)

where \( \rho = \lambda/\gamma(C) \) and

\[
\Pi(j) = \begin{cases} 
    \Pi_{k=0}^{i-1} \gamma(C - k); & j > 0; \\
    1; & j = 0;
\end{cases}
\]

(3)

The quantity \( p(0, 0) \) can be obtained by normalization, i.e.

\[
p(0, 0) = 1/(1 - \rho + \sum_{j=1}^{C} (\Pi(j)/\lambda^j))
\]

(4)

From the above equations, we can obtain the marginal queue-length probability distributions.
The above analysis was carried out assuming that $\gamma(n), n = 1, 2, ..., C$, are known. These quantities are obtained in the following section.

### 3.3 A Model for the Communication Network

The queue flow model (QFM) for the communication network is given in figure 9. Transport packets arrive at the first queue, named HOSTA after they have been given a grant to pass the end to end transport flow control mechanism. HOSTA mirrors the service given by protocol functions of layer 3 and 4 in our sending host, i.e. TCP/IP like services. We assume the service times to be exponentially distributed, with a mean 100 ms. This service time can be seen as the completion time in a time sharing system environment. The service includes the communication protocol functions that are needed until a Token bus packet is formed and is ready to enter the queue at the NIU (Network Interface Unite).

System LAN1AS (Local Area Network number 1, Access Scheme) is a model of the Token bus behaviour. We let access times as well as transmission times for the Token bus to be incorporated into a single service time. Packets in this queue belong to the same host to host communication process and are waiting to access the Token bus. We have modeled this service time as a Coxian-2
distribution. Its three unknown parameters were obtained using the method of moments. The values of the first three moments were obtained from a Token bus simulation program. The simulation program models a total Token bus according to the entire 802.4 standards. All the stations attached to the bus were assumed to offer a symmetric load. The first three moments of the access and transmission time were estimated for various loads. For a 20% load they were $E \{ x \} = 1.33 \text{ms}$, $E \{ x^2 \} = 7.40(\text{ms})^2$ and $E \{ x^3 \} = 73.86(\text{ms})^3$. Under a 30% load the values were set to $E \{ x \} = 1.59 \text{ms}$, $E \{ x^2 \} = 9.78(\text{ms})^2$ and $E \{ x^3 \} = 95.81(\text{ms})^3$.

System GW1 (Gateway number 1) models the total gateway functions. This gateway should be seen as a standard personal computer totally dedicated to gateway functions. In view of this, the service times are rather long and do not vary much in time. We assume the service times to be Erlang-5 distributed, with a mean equal to 200 ms. The service includes the fragmentation of arriving packets into smaller WAN-packets to which X.25-headers are attached. Thus, we are faced with bulk arrivals depending on the user data length to the next system, i.e. the transmission channel to the WAN. Those WAN-packets will be reassembled at the gateway at the other side of the WAN.

The next system, GW1TRANS, reflects the transmission channel to the WAN. We have assumed it to be a standard 38.4 kbit link over which WAN-packets are transmitted. We assume that WAN-packets have a constant length equal to 131 bytes (128 bytes of data and an X.25 overhead of 3 bytes). This implies that the last packet in a bulk is always regarded to be filled. We model bulk arrivals by simply keeping track of the last WAN-packet within a bulk. This is possible due to the fact that the WAN, which follows immediately after GW1TRANS, is modeled as an infinite server and as reassembly takes part in the server after the WAN, namely in GW2. These two systems have been modeled, so that the concept of “modeling the last packet in a bulk” reflects the effects of splitting as well as transmitting a number of smaller packets. In view of this, the service times of the systems between the two gateways, i.e. GW1TRANS, WAN and GW2 (Gateway number 2), are adjusted to reflect the service of all WAN-packets which make
up a bulk. System GW1TRANS is modeled as a single server queue served by a Coxian-2 server. The Coxian-2 distribution is obtained using the method of moments. The first three moments are set so that to reflect the total bulk transmission. In particular, the Laplace transform for the pdf of the length of packets split in GW1 is given by expression (5), where $e^{-sTH}$ reflects the transport header added to user packets.

$$B^*_T(s) = e^{-sTH} \left( \frac{\lambda_1}{\lambda_1 + \lambda_2 \mu_1} + \frac{\mu_1}{\mu_1 + s} e^{-\mu_1/\mu_2} + \frac{\lambda_2}{\lambda_1 + \lambda_2} e^{-s/\mu_2} \right)$$

(5)

The bulk size distribution, $g_l = P(\text{bulk size} = l)$ is given by

$$g_l = B_T((l-1)d) - B_T(ld)$$

(6)

where

$$B_T(x) = \int_x^\infty b_T(z)dz; \quad B_T^*(s) = \int_0^\infty e^{-sz}b_T(z)dz;$$

(7)

and $d$ is equal to the WAN-packet length in bits, excluding X.25-headers.

Let $B^*_b(s)$ be the Laplace transform of the service time distribution for the "last" packet in a bulk. This service time distribution reflects the time it takes to give service to a whole bulk. Then

$$B^*_b(s/\text{bulk size} = l) = e^{-s(d+XH)}l$$

(8)

and

$$B^*_b(s) = \sum_{l=1}^N e^{-s(d+XH)}lg_l$$

(9)

which gives us

$$B^*_b(s) = G^*(e^{-s(d+XH)})$$

(10)

where $XH$ is the X.25 header and

$$G^*(z) = \sum_{l=1}^N z^lg_l; \quad N = \text{int}\left(\frac{1/\mu_2+TH}{d}\right) + 1;$$

(11)

Thus the $k$:th moment for the service times in GW1TRANS is given by

$$(-1)^k \frac{d^kG^*(e^{-s(d+XH)})}{ds^k}\bigg|_{s=0}$$

(12)
System WAN is to represent an X.25 packet switching network. As this network is not specified in detail, we model it as an infinite server with an exponential service time distribution with a mean of 300 ms. Upon completion of this service time, the last WAN-packet belonging to a bulk is forwarded to the gateway.

For system GW2, the second gateway, that connects the Ethernet and the WAN, we have assumed an exponential service time distribution with a mean equal to 50 ms. This service includes reception of WAN-packets, normal X.25 link handling, WAN-packet reassembly and finally forming IP-packets to be sent over a new IP to IP connection to the receiving host on the other side of the Ethernet. However, the IP-packets may well be too large to fit into Ethernet MAC-frames. So fragmentation may take place again.

This fragmentation is taken place in system GW2FRAG. For this system we have used an Erlang-5 distribution with a mean of 5 ms. The fragmentation leads to bulk departures of Ethernet-packets, packets which are now ready to access the Ethernet.

System LAN2AS (Local Area Network number 2, Access Scheme) is modeled as a single queue served by a Coxian-2 server. We use the same approach as for the gateway to gateway communication, i.e. we just model the last packet of a bulk. The service time, therefore, is the time required to access and transmit all the packets in a bulk. The three parameters of the Coxian-2 distribution are obtained using the method of moments. The three moments were estimated using a simulation model of an Ethernet. The simulation model reflects the 802.3 standards completely. As in the case of the Token bus simulation, we have estimated the first three moments for various loads. All stations attached to the Ethernet have the same load. We note that the estimated access plus transmission times show up very large variations due not only to the "normal" variations of Ethernet performance. As we model a whole bulk, a service time was estimated from the time the first packet in a bulk senses the channel until the last packet in a bulk has been transmitted. For a 20 % load we have $E\{x\} = 0.16\text{ms}$, $E\{x^2\} = 0.16(\text{ms})^2$ and $E\{x^3\} = 1.69(\text{ms})^3$ and for the 30 % load $E\{x\} = 0.31\text{ms}$, $E\{x^2\} = 1.55(\text{ms})^2$ and $E\{x^3\} = \ldots$
System HOST* BREC (Host B, Reception) receives the Ethernet packets and also handles the packet reassembly. An Erlang-5 distribution with a mean of 5 ms is set for this server.

The last system, HOSTB, is a model of the layer 3 and 4 protocol services in the receiving host. This system delivers transport packets to the user application and also sends acknowledgements to the transport protocol at the sending host.

Now, we obtain \( \gamma(n), n = 1, 2, ..., C \) by treating the queueing network shown in figure 9 as a closed queueing network. That is, customers departing from the last node, (Host B), are fed back into the first node (Host A) through the acknowledgement channel. This queueing network is not of the BCMP type seeing that some of the service distributions are coxian. In view of this, it can be only analyzed approximately. In this paper, we obtain \( \gamma(n), n = 1, 2, ..., C \) using an approximation algorithm due to Marie [17] and implemented in QNAP2 [18].

4 SOME SAMPLE RESULTS

The analytic model described in the previous section was implemented on a computer using QNAP-2. The end to end delay model can be seen as a hierarchical model consisting of three levels of models. In particular, the simulation models for the token bus and ethernet can be seen as the first level of modelling. The results obtained from these simulation models are used in the second level model, the queueing network model of the communication network, in order to obtain the conditional throughput \( \gamma(n), n = 1, 2, ..., C \). Finally, these results are used in the third level model, the end to end window flow control model, in order to obtain relevant performance measures.

Here we will summarize some available results obtained using our hierarchical model. The parameters for those cases are given next.

The first three moments of the access and transmission time for both the Token bus and the Ethernet have been computed for a 20% load. For the station GW1TRANS, the first three
moments of the coxian service time distribution depend on the “user data” packet length distribution. This distribution which should mirror a mix of interactive and bulk data traffic is described by the s.v. $s$

$$s = \alpha s(1) + (1 - \alpha) s(2) + s(3) \quad 0 \leq \alpha \leq 1$$

(13)

where $\alpha$ is the proportion of interactive packets, and $(1 - \alpha)$ the proportion of bulk data packets. $s(1)$ denotes the packet length distribution of interactive packets, here assumed to be exponentially truncated with mean 100 bytes. $s(2)$ the packet length distribution of bulk data packets assumed to be deterministic and equal to 8000 bytes, and finally the constant $s(3)$ represents the total amount of headers.

In order to draw the end to end delay as a function of the “Mean Arrival Time” with $W$, the maximum window size, as a parameter (see figure 10), we have to compute, for each value of $W$, the stability condition $\lambda^*$ and the following parameters: $1/\lambda$ the mean user data entities arrival time; $\rho_s$ the busy utilisation of the semaphore queue; $N_r$ the mean number of customers in the customer queue; $N_t$ the mean number of customers in the token queue; $N_s$ the mean number of customers in the entire semaphore queue; $R_s$ the mean response time of the semaphore queue.

The maximum load that the system can handle as a function of the semaphore utilization is
desplayed graphically in figure 11. As predicted, for a constant semaphore queue utilization, we notice a system performance degradation (in terms of managable load), when the window size decreases.

We have also emphasized the influence of the characteristics of "user data" traffic on the end to end delay. If $\alpha$ is close to zero we have mainly bulk traffic and $\alpha$ close to one indicates a major proportion of interactive traffic. Here we use the expression "stability condition" as the maximum load that the system can manage. This load can be expressed as the maximum user data entities to be sent per time unit or as the minimum amount of time units between two successive user data entities to be sent. Our time unit is one millisecond and the maximum window size, $W$, was set to 7. Figure 12 shows the stability condition as a function of the parameter $\alpha$. Given the same total work load, we notice that the user data distribution is of great importance to the system performance. The results are improved as $\alpha$ is increased. Those observations are emphasized in figure 13, where we have plotted the end to end delay as a function the user load for different values of $\alpha$. We have also added the busy utilization of the semaphore queue ($\rho_s$) to show that for the same value of $\rho_s$, the performance degradation of the end to end delay can
Stability condition (ms) as a function of Alpha

Figure 12:

End to End Delay as a Function of the Mean Interarrival Time and the busy Utilization of the Semaphore Queue

Figure 13:
be very important, when the interactive part of the traffic ($\alpha$) decreases.

The model we have developed in this paper associates simulation measurements and analytical techniques and should be very useful to understand and study the behaviour of catenet environments including several splitting and reassembly mechanisms.

5 CONCLUSIONS AND AN OUTLINE FOR FURTHER RESEARCH

We have given an analysis for the end to end delay in a catenet environment. Our model, which could be seen as a three layer hierarchical model, incorporates all those system characteristics that have a great influence on the delay. Basic end to end flow control mechanisms as well as different packet fragmentation and reassembly functions are modeled.

This work calls for further research within this area. Efforts should be made to develop a more general approach, which should incorporate the possibilities of handling a number of flow control and packet fragmentation and reassembly functions belonging to different communication layers as well as tools to obtain higher moments of performance entities.

References


