

# An Analysis of the Congestion Effects of Link Failures in Wide Area Networks

D.Tipper, J.Hammond, S.Sharma, A.Khetan, K.Balakrishnan and S.Menon  
Electrical and Computer Engineering Department  
Clemson University, Clemson, SC-29634-0915.

## Abstract

*Here we present the results of a study to determine the effects of link failures on network performance. The network studied is a virtual circuit based packet switched wide area network. A generic queueing framework is developed to study the effect of failures, and the subsequent traffic restoration, on network performance. In general, the congestion resulting after a failure is a transient phenomenon. Hence a numerical methods based nonstationary queueing analysis is conducted in order to quantify the effects of failures in terms of the transient behavior of queue lengths and packet loss probabilities. A bounding relationship is developed whereby a network node can determine whether or not congestion will occur as the result of traffic restoration after a failure.*

## 1 Introduction

Due to the rapidly growing demand for information transfer across communication networks, the need for reliable communication service has become increasingly important. It has been noted [1] that high exposure businesses such as airlines, mail order retail and banking can lose up to 6 million dollars per hour in unrecoverable revenue due to a communication network failure. The potentially drastic effects of network failures has been demonstrated by the recent series of highly publicized failures [1, 2]. Such accidents have clearly shown the need for reliable networks that provide service which is robust to failures.

A network failure, such as the loss of a link, can occur due to a wide variety of reasons causing service disruptions ranging in length from seconds to weeks. Typical events that cause failures [1, 2] are accidental cable cuts, hardware malfunctions, software errors, natural disasters (e.g., fires), and human error (e.g., incorrect maintenance). Since many of the causes of failures are outside the control of network providers, there has been increasing interest in the design of survivable networks. This work has largely focused on planning the network to reduce the impact of failures

when they occur. Several techniques have been proposed to minimize the effect of failures, some common ones being multiple ingress/egress of users, trunk diversity, digital cross connect systems, and self-healing ring architectures. A survey of current survivability techniques is given in [8].

The survivability literature primarily concentrates on network design issues and relatively little literature exists on quantifying the performance impact of a failure. Here we present the results of a study of the effects of link failures on network performance for a virtual circuit, packet switched, wide area network. The network studied is modeled after IBM's proposed plaNET (formerly PARIS) network architecture [3, 4], which is a high speed integrated packet switched network supporting a wide variety of traffic types. After a link failure, several network controls come into play, such as congestion control, call admission control and routing, in order to restore the lost traffic. The restored connections can result in a transient period of congestion occurring which can have a significant effect on the quality of network service. In this paper a general framework is developed for studying the congestion that can occur in a network as the result of a failure. The nonstationary network analysis techniques developed in [5] and [6] have been applied within the general framework developed here in order to model and quantify network performance after link failures.

## 2 Modeling the effects of failures

Consider the arbitrary packet switched wide area network shown in Figure 1. We assume that the network uses virtual circuit service to transport packets and source node routing of the virtual circuits. In source node routing each network node maintains a database of the network topology and determines the route through the network for all virtual circuits originating at the node.

The impact of a link or node failure on the network performance will depend on the complex interaction of several factors, some of which are the location of the

failure, the network topology, the network loading, the routing algorithm, the error control procedures, and the congestion control. Note that most of these factors will be specific to the network under study and determining generic effects of failures on network performance is difficult. However, certain aspects of the behavior of the network can be modeled by studying the general steps of the network in reacting to failures. In the event of a link failure all the virtual circuits using the failed link are disrupted and need to be reconnected if possible. The reconnection of the virtual circuits takes place only after a time delay which consists of the time taken to detect the link failure, plus the time for the affected source nodes to get the relevant information and the time taken to determine the new route and set up the connection. During the time delay taken to restore a connection, a backlog of packets will accumulate at the source of each virtual circuit. Hence, when these virtual circuits are reconnected, the backlog of packets from all of the virtual circuits must be transmitted and congestion can occur at various points in the network.

One would expect that congestion control mechanisms would come into play after traffic restoration to prevent the occurrence of congestion. Congestion control schemes proposed for broadband networks are based on either end-to-end windowing schemes or rate based policing mechanisms [7] (e.g., virtual leaky bucket, leaky bucket). In the event of a failure while using the windowing schemes, the source nodes keep transmitting the packets belonging to the virtual circuits that use the affected link until their corresponding window runs out. These packets are lost and hence must be retransmitted. Further packets, however accumulate at the access nodes awaiting an opening in the window. In broadband networks, the window size will typically be quite large [7] so as not to interrupt the packet flow when the network is operating in a normal fashion. Thus, windowing schemes do not help in controlling the congestion as the burst of packets that are generated pass through the window of the source, the window being set to a large value to optimize the steady state flow in the network.

The virtual leaky bucket scheme would also not prevent congestion as the packets would be tagged and still admitted to the network. The leaky bucket algorithm would allow a burst of packets at the peak rate to enter the network and then packets would enter at the leaky bucket token generation rate until the backlog is worked off. Observe that the congestion at the network node is mainly due to the rerouted virtual circuits needing to simultaneously work off their back-

logs. This is, in effect, a correlated burst at the network node, yet the current congestion control schemes are designed to prevent congestion from occurring due to an individual user rather than a group. Also, note that current congestion control schemes are designed on the basis of a steady state analysis and the congestion after a failure will be transient in nature. Thus, the current congestion control schemes cannot entirely prevent congestion after a failure.

The occurrence of congestion and its spreading in the network can be studied by noting that a link failure partitions the network nodes into four categories namely: primary nodes, secondary nodes, tertiary nodes, and quaternary nodes. Primary nodes are those nodes which are the source nodes for virtual circuits that were traversing the link which failed. Thus the primary nodes are responsible for the restoration of the affected virtual circuits. Secondary nodes are nodes which are relay nodes for the rerouted virtual circuits. Tertiary nodes are defined as those nodes which handle traffic that share a link in common with the rerouted traffic. Quaternary nodes are those nodes whose traffic is not affected by the traffic of the reconnected virtual circuits.

Congestion starts at the primary nodes, which can become directly congested by the backlogged packets at the attached sources. If links at a primary node become heavily congested the congestion can spread downstream to secondary nodes. Tertiary nodes can also become congested in an indirect way caused by increased packet loss rates at primary or secondary node links that the tertiary node traffic shares with the restored traffic.

We examine the modeling of the primary, secondary, and tertiary nodes by studying a representative portion of the network, specifically the region circled in Figure 1. Figure 2 is a magnification of this region of the network, illustrating a typical set of network nodes. Note, that Figure 2 also shows a representative loading pattern for the network nodes. Let us focus on node 3 and study the effects of a link failure at this node. One can see from the figure that node 3 is the source node for virtual circuits VC#1 - VC#4 whereas, it is a relay node for virtual circuits VC#5 and VC#6. Consider the effect of a link failure, in this case link 3-2. The link failure results in node 3 being a primary node since it must reroute virtual circuits VC#1 - VC#3. In general, the location of the link failure and the network loading pattern at the time of the failure will specify a set of primary nodes. Here we focus on modeling a generic primary node such as node 3.

After a time delay involved in the detection of the failure a primary node will try to restore its affected virtual circuits. Note that some of the virtual circuits may be blocked due to insufficient available bandwidth. However these blocked circuits will not affect the level of congestion in the network and are ignored here. After the primary nodes have completed traffic restoration, a new loading pattern will exist in the network as illustrated in Figure 3. After traffic restoration, the traffic at an arbitrary primary node can be grouped into the following three categories: *i*) Normal traffic - traffic not on the failed link, *ii*) Source rerouted traffic - traffic on the failed link for which this is the source node and *iii*) Transit rerouted traffic - traffic on failed link for which this is a relay node. This categorization of the traffic is illustrated for node 3. From figure 3 we have the following groupings for node 3: VC#4 and VC#5 are normal traffic, VC#1 - VC#3 are source rerouted traffic, and VC#6 is transit rerouted traffic.

Focusing on a single output link at the primary node, in this case link 3-4, we can develop the generic queueing model of a primary node output link shown in Figure 4(a). In Figure 4(a),  $C_{ij}$  denotes the capacity of the output link  $i-j$ , and the node buffer represents the finite buffer at the output link under consideration. We have assumed that the nodal buffer space is completely partitioned among the output links. The normal traffic has been grouped together and is shown to have aggregate mean packet arrival rate  $\lambda_{nt}$ . The source rerouted traffic is represented by the individual virtual circuit's source queues feeding into the nodal buffer. Note, that each source queue is modeled as an infinite capacity queue and the mean packet arrival rate of the  $i$  th source rerouted traffic is denoted by  $\lambda_i$ . Also in the figure, the transit rerouted traffic is represented as an aggregate traffic stream with mean rate  $\lambda_{trt}$ , shown flowing through several queues.

Congestion can occur at the nodal buffer due to the backlog of packets at the source rerouted traffic queues and due to congestion propagating from the primary nodes of the transit rerouted traffic. The primary node only has direct control over the source rerouted traffic and this traffic is the most likely cause of congestion at the node. Thus we simplify the queueing model by combining the transit rerouted traffic and normal traffic into a single traffic stream termed the background traffic. The resulting queueing model is shown in Figure 4(b) with the mean packet arrival rate of the background traffic defined as  $\lambda_{bg}$ . The queueing model of Figure 4(b) can be used to study congestion at the primary node that results from a link failure.

In a fashion similar to the analysis for the primary node, we have developed queueing models to study the spread of congestion to secondary and tertiary nodes and these are given in [9], where it is shown that the general primary, secondary, and tertiary model provides a overall framework for the study of failures and the issues associated with them.

### 3 Quantifying the congestion effects of failures

Utilizing the approach of Section 2 one can determine the impact of link failures on the performance of the network. We focus on analyzing congestion at a generic primary node and determining when congestion will occur. The network studied is a simplified variant of IBM's plaNET. The plaNET network is an experimental high speed packet switching system for integrated voice, video, and data communications. The plaNET network architecture is based on the previous PARIS system and both network architectures are described in a number of papers [3, 4].

Since we are primarily interested in the effects of failures, several simplifying assumptions are made in the modeling of the plaNET network. Consider the primary node queueing model shown in Figures 4b. In this model, all links are assumed to be full duplex with the capacity of the primary node output link denoted by  $C$  and the capacity of the  $i$  th source rerouted access link denoted by  $C_i$ . As in plaNET, variable length packets are transported via virtual circuit connections. Packets are assumed to arrive to the network according to independent Poisson processes with fixed mean rate  $\lambda_i$  for the  $i$  th virtual circuit being reconnected. Furthermore, we assume exponentially distributed packet lengths with mean  $1/\mu$  and that the service rate of a packet at a link is proportional to the link capacity. The buffer space at the network node output link is finite with system size  $K$ . Packets which are dropped in the network are retransmitted from the traffic source using a selective repeat mechanism.

In order to generalize the results so that they can be scaled to any channel capacity, the transmission rate of the primary node link was normalized to have a value of 1.0 (i.e.,  $C = 1.0$ ). Also, the mean packet length  $1/\mu$  is normalized to a value of 1.0. Hence, the mean packet service rate of the output link is  $\mu C = 1.0$  and one can unnormalize quantities by relating them to the mean service rate of the actual link  $\tilde{\mu}\tilde{C}$ . For example, consider a link which is a T1 line (i.e.,  $\tilde{C} = 1.544$  Mbps) in a network where the average packet length is  $1/\tilde{\mu} = 2000$  bits/packet. Thus in the actual network the service rate is  $\tilde{\mu}\tilde{C} = 772$  packets/sec. Hence, a virtual circuit of mean rate  $\lambda_i = .025$

in the normalized network translates to a virtual circuit of  $\tilde{\lambda}_i = \lambda_i \tilde{\mu} \tilde{C} = 38.6$  kbps in the actual network.

The plaNET network provides a guaranteed steady state grade of service (GOS) for each virtual circuit that has been allowed to enter the network. The maximum flow on any network link is controlled by link congestion thresholds  $TH_{ij}$  in the call set up procedure. Here we assume the congestion threshold is  $TH_{ij} = .85$ . Hence, under the assumptions that each queue can be represented by a finite M/M/1 queue a worst case grade of service at any network link can be determined using the maximum flow which is equivalent to the maximum link utilization. Specifically, performance measures, such as the maximum allowable packet loss rate at a network queue  $PB$ , can be found using the standard M/M/1/K queueing formula. For example, in the networks modeled here the maximum link utilization is  $\rho = TH_{ij} = .85$  and the buffer system size is  $K = 21$  which results in a value of  $PB = 5.08 \times 10^{-3}$ . Similarly, one can determine the average number in the system  $NS$  at any queue. As above, to determine the worst case the values of  $\rho = .85$  and  $K = 21$  were used, which yields  $NS = 5.03$ . Thus, the guaranteed grade of service at each network link is a average number in the system  $NS_{GOS} = 5.03 \approx 5$  and a packet loss rate  $PB_{GOS} = 5.08 \times 10^{-3}$ . The network is assumed to be performing satisfactorily only when the performance parameters are less than or equal to the GOS values.

The failure of a network link would be detected only after a certain amount of time. This detection would be done by one of the nodes to which the failed link is attached. This node would then transmit the information to all the other nodes as a control message. The virtual circuits that had been using the failed link continue transmitting packets until their source nodes get the information about the failure. These packets are assumed to be lost in the network and have to be retransmitted upon reconnection of the virtual circuit. Typical times for a node to detect a failure can range from a few milliseconds to two seconds. The downtime [1] allowed before the user will disconnect the session is on the order of two seconds for a voice call and ten seconds for data connections. Let  $\tilde{\tau}$  denote the time taken by a node to detect the failure in the actual network, then the backlog of packets at the traffic source of the  $k$  th session at the node will be  $\tilde{\lambda}_k \tilde{\tau}$ . Thus for the same number of packets to be lost in the normalized network where  $\lambda_k = \tilde{\lambda}_k / \tilde{\mu} \tilde{C}$  the time to detect the failure  $\tau$  must be given by  $\tau = \tilde{\tau} \tilde{\mu} \tilde{C}$ . Here we assume that the time to detect the failure in the actual network is  $\tilde{\tau} = 1$  second. Thus in this second,

a network operating at standard T1 (1.54Mbps) rates with the average packet size of 2 Kbytes would have allowed the transmission of a maximum of 772 packets. Directly extending this assumption, any virtual circuit transmitting at an average rate of  $\lambda_k$  would have a backlog of  $(772 \times \lambda_k)$  packets to be retransmitted. This backlog of packets arriving at the network nodes from reconnected calls can lead to a violation of the grade of service. This condition can exist until the entire backlog has been transmitted and the network reaches steady state once again.

Consider the queueing model of a primary node output link shown in Figure 4b. Focusing on the network primary node output link queue, we assume that customers arrive to the network queue according to non-stationary Poisson processes with mean rate  $\lambda(t)$  at time  $t$ . Furthermore, we assume that the link can be approximately modeled as a M/M/1/K queue with mean service rate  $\mu C$ , where  $1/\mu$  is the average packet length and  $C$  is the link capacity. Defining  $p^j(t)$  as the probability of  $j$  packets being in the queueing system at time  $t$ , then the Chapman-Kolmogorov differential equation model for the queue can be derived as :

$$\begin{aligned} \frac{dp^0(t)}{dt} &= -\lambda(t)p^0(t) + \mu C p^1(t) \\ \frac{dp^j(t)}{dt} &= \lambda(t)p^{j-1}(t) - [\lambda(t) + \mu C]p^j(t) \\ &\quad + \mu C p^{j+1}(t); \quad 0 < j < K \\ \frac{dp^K(t)}{dt} &= \lambda(t)p^{K-1}(t) - \mu C p^K(t). \end{aligned} \quad (1)$$

Given the time-varying link load  $\lambda(t)$  and initial conditions, one can solve (1) using standard numerical integration techniques.

Consider the source queueing model of the  $i$  th connection at a primary node. We assume the source queues can be modeled as M/M/1. After a link failure the source queues will have a large number of backlogged packets to transmit. Thus using a finite capacity detailed Chapman-Kolmogorov model, to represent each queue is computationally difficult due to the large number of equations required. Hence, a approximate fluid flow model approach is adopted.

Following [5], we define  $C_i$  as the capacity of the link,  $NS_i(t)$  as the number in the system (i.e., queue + server) at time  $t$  and  $x_i(t)$  as the state variable representing the average number in the system at time  $t$ . Note that the state variable is the ensemble average of the number in the system at time  $t$  (i.e.  $x_i(t) = E\{NS_i(t)\}$ ). Assuming that the queue can be modeled as M/M/1, it was shown in [5] that by matching the steady state equilibrium point of the fluid flow

model with that of the queuing model leads to

$$\dot{x}_i(t) = -\mu C_i \left( \frac{x_i(t)}{1 + x_i(t)} \right) + \lambda_i(t) \quad (2)$$

with initial condition  $x_i(0)$ . This was shown to provide an accurate approximation to the time-varying mean number in the system. The initial condition for the fluid flow model  $x_i(0)$  denotes the number in the system of the  $i$ th source queue at the time of reconnection of the virtual circuit. Let  $T_i$  denote the time to reconnect the  $i$ th connection at the primary node in question. The initial condition for the  $i$ th source queue is the backlog of packets that accumulates during the reconnection time  $x_i(0) = \lambda_i \times T_i$ .

In order to have an accurate model after a failure, the effects of packet dropping at the primary node queue must be included. Consider a sample backlogged packet for the  $i$ th virtual circuit, the probability that the packet is dropped on its first attempt is  $PB$ , the blocking probability at the primary node output link. Assuming that the blocking probability remains constant over the period of interest, the probability that the packet gets blocked on its  $n$ th attempt is  $PB^n$ . Thus, the total number of retransmissions  $NR$  is given by  $NR = PB \sum_{i=0}^{\infty} PB^i = \frac{PB}{1-PB}$ . Hence, the total number of transmissions  $NT$  required for the tagged packet can be given as  $NT = \frac{1}{1-PB}$ .

Therefore, the average time  $TT_i$  required for one packet to successfully get transmitted at source queue  $i$ , is given by  $TT_i = \left( \frac{1}{1-PB} \right) \frac{1}{\mu C_i}$ . Note, that the increase in the average time for the packet to be transmitted is equivalent to decreasing the capacity of the source queue access link by a factor of  $(1 - PB)$ . Let  $\hat{C}_i$  denote the adjusted link capacity, that is  $\hat{C}_i = C_i(1 - PB)$ . Then the fluid flow model of the  $i$ th source queue (2) becomes

$$\dot{x}_i(t) = -\mu \hat{C}_i \left( \frac{x_i(t)}{1 + x_i(t)} \right) + \lambda_i(t). \quad (3)$$

The fluid flow model (3) can be used to model the source queues taking into account packet retransmissions. Note that  $PB$  must be known and this can be determined from (1) as  $PB = p^K(t)$ . Observe that the inclusion of retransmission effects provides a feedback coupling between the source queue and primary node link models.

In addition to the fluid flow models of the source queues and the detailed model (1) of the primary node output link, the arrival rate to the primary node output link  $\lambda(t)$  must be known. This can be found using the flow balance principal. Let  $\lambda_{bg}(t)$  represent the

mean arrival rate of the Poisson background traffic. It is important to note that though the effective service time at the source queues is reduced by retransmissions occurring, the flow of packets out of the source queues into the primary node link is unaffected. Then the arrival rate at the primary node output link queue at time  $t$  is given by

$$\lambda(t) = \lambda_{bg}(t) + \sum_{i=1}^N \mu C_i \left( \frac{x_i(t)}{1 + x_i(t)} \right) \quad (4)$$

where the second term on the right hand side represents the arrival rate of traffic from the source queues. This equation provides a coupling between the queues, since the output rate from each source queue is given as a function of the number in the system at the source queue. Note that when steady state conditions prevail at all source queues (i.e.,  $\dot{x}_i(t) = 0 \forall i$  and constant loads  $\lambda_i(t) = \lambda_i \forall t$  and  $\forall i$ ) the arrival rate of packets to the primary node link becomes  $\lambda(t) = \lambda_{bg}(t) + \sum_{i=1}^N \lambda_i$ .

Hence, the primary node queuing model of Figure 4b can be modeled by a coupled differential equation model consisting of  $n$  equations of the form (3) representing the source queues along with the system of equations (1) to model the primary node link and equation (4) to determine the primary node queue arrival rate. One can apply standard numerical integration techniques in an iterative fashion over the time intervals of interest to solve (1) together with (3) and (4) for the state probabilities  $p^j(t)$  of the primary node queue and the number in the system  $x_i(t)$  at each source queue. The steps in the solution technique are described in [5, 9]. From the time-dependent state probabilities  $p^j(t)$  and state variables  $x_i(t)$ , one can study the performance of the queuing system. For example,  $NS(t)$ , the expected number of packets in the system at the primary node link at time  $t$ , is given by  $NS(t) = \sum_{j=1}^K j p^j(t)$ .

In order to check the accuracy of the model, its solution was compared to a detailed simulation for several cases. A typical experimental study is reported here for the two virtual circuit case. In the experimental model, the capacity of the primary node output link was set at  $C = 1$ , the capacity of the source node access links were set at  $C_1 = C_2 = .45$  and the mean packet length was one ( $1/\mu = 1$ ). The system size of the primary node link was  $K = 21$ , and no background traffic was present ( $\lambda_{bg} = 0$ ). As a worst case the virtual circuits were assumed to be reconnected simultaneously with the time to reconnect being  $T_1 = T_2 = 772$  seconds, which as noted previously translates to a time to reconnect of one second in an

## 9a.2.5

actual network operating at T1 rates. An illustrative result of the model accuracy is shown in Figure 5 for virtual circuit arrival rates being  $\lambda_1 = \lambda_2 = .06$  and the primary node initially being empty. In Figure 5 the  $NS(t)$  starting at the time instantance that the virtual circuits are reconnected is plotted along with the results of a discrete event simulation. For the simulation results three curves are given showing the point estimate of the number in the system and the upper and lower bounds of the 95% confidence intervals. From the figure one can see that the analytical model is reasonably accurate. The simulation results were generated from 5000 independent runs using the ensemble averaging technique of [6].

#### 4 An analysis of congestion at primary nodes

A parameterized study of the primary node queuing model was conducted to determine when congestion occurs in the primary node. In the study the normalized network model was used for the primary node output link. As noted in Section 3 for the normalized model, the system size was  $K = 21$ , the link capacity was  $C = 1$ , the mean packet length was one  $1/\mu = 1$  and the call admission congestion threshold was  $TH_{ij} = .85$ . The resulting GOS at the network link was  $NS_{GOS} = 5.03$  and  $PB_{GOS} = 5.08 \times 10^{-3}$ . The primary node output link was considered congested when  $NS(t)$  exceeded the grade of service level (i.e.,  $NS(t) > NS_{GOS} = 5.03$ ). The number in the system was chosen as the metric rather than  $PB(t)$  since our numerical results showed that  $NS(t)$  was the more sensitive metric. The time to detect the link failure and reconnect the  $i$  th session was assumed to be  $T_i = 772 \forall i$ , which is equivalent to one second in an actual network operating at T1 link rates. The queue at the primary node link was assumed initially empty. Since we are interested in the effect of the rerouted traffic from the failed link it was assumed that no background traffic is present.

The purpose of the study was to determine what combination of parameters ( $C_i, \lambda_i$  etc.) would result in congestion at the primary node link. Note that, the steady state arrival rate to the primary node can never be greater than the congestion threshold of  $TH_{ij}$ . Congestion can occur only when  $\lambda(t) > TH_{ij} = .85$  for an amount of time sufficient for  $NS(t)$  to grow larger than  $NS_{GOS}$ . From (4) with  $\lambda_{bg}(t) = 0$ , one can see that an upper bound on the input arrival rate is  $\lambda(t) \leq \sum_{i=1}^n \mu C_i$ . Congestion can not occur if the upper bound is less than  $TH_{ij}$ , specifically a necessary condition for congestion to occur is that  $\sum_{i=1}^n \mu C_i > TH_{ij}$ . Thus, after a failure each primary

node can check this condition and determine whether congestion is possible. While this condition is necessary for congestion to occur it is not sufficient and we adopt a numerical approach to determine a curve which precisely indicates the occurrence of congestion.

Consider the case of one virtual circuit  $n = 1$  being reconnected through the primary node output link. Obviously for congestion to be possible,  $C_1 \geq TH_{ij} = .85$ . A numerical study was conducted by assigning particular values to the access link capacity  $C_1$  and the input rate  $\lambda_1$  then solving the differential equation model to find  $NS(t)$ . The values of  $NS(t)$  determined from the differential equation model were checked to see if  $NS(t) > NS_{GOS}$ . The value of  $\lambda_1$  was then varied and the experiment repeated until the bounding value of  $\lambda_1$  was found where the GOS is just met (i.e.,  $NS(t) \leq NS_{GOS} \forall t$ ). The experiment was then repeated for a different value of  $C_1$  until the bounding curve of Figure 7 was determined. From Figure 6, all points  $(\lambda_1, \mu C_1)$  lying below the bounding curve are those where the GOS is met, where as for those points above the curve the GOS will not be provided.

A similar study was conducted for the case of two virtual circuits being rerouted through the primary node link. The experimental model was the same except for the additional access link  $C_2$  and mean arrival rate  $\lambda_2$  of the second session. The condition for congestion to be possible in this case is  $C_1 + C_2 > .85$ . Note that for the two virtual circuit case there are four parameters to vary (i.e.,  $\lambda_1, \lambda_2, C_1$ , and  $C_2$ ) and this makes an exhaustive search for bounding conditions difficult. Hence, to reduce the number of parameters the values of  $C_1$ , and  $C_2$ , were fixed as well as the ratio  $\lambda_1/\lambda_2$  and the total load  $\lambda_1 + \lambda_2$  was varied to the point where the number in system exceeds the grade of service. This was repeated for different values of  $C_1$  holding  $C_2$  and  $\lambda_1/\lambda_2$  constant and the resulting values were plotted as a curve. The entire experiment was repeated for different values of the ratio  $\lambda_1/\lambda_2$  and the resulting family of curves are shown in Figure 7. Thus, in Figure 8 each curve is a bounding curve for a particular ratio of the arrival rates.

In Figure 7 an interesting effect can be seen, notice that for a particular ratio of  $C_1/C_2$  the bounding curve that is the worst case ( i.e. lowest load to exceed the grade of service ) is always the curve resulting from  $\lambda_1/\lambda_2 = C_1/C_2$ . For example, at the value  $C_1/C_2 = .3$  the curve that is leftmost is the one resulting from  $\lambda_1/\lambda_2 = .3$ . The condition  $\lambda_1/\lambda_2 = C_1/C_2$  is equivalent to  $\rho_1 = \rho_2$ , that is the utilization of the two access links being equally balanced. In order to clearly demonstrate this worst case effect an experi-

mental study was conducted where the values of  $\mu C_1$ ,  $\mu C_2$ ,  $\lambda_1 + \lambda_2$  were fixed and ratio  $\lambda_1 / \lambda_2$  was varied. The results of a typical study of this type are shown in Figure 8, where  $NS(t)$  and  $PB(t)$  are plotted versus time for various ratios of  $\lambda_1/\lambda_2$ . In Figure 8 the horizontal lines represent the grade of service levels  $NS_{GOS}$ ,  $PB_{GOS}$ . One can clearly see from the figure that when  $\lambda_1/\lambda_2 = C_1/C_2 \rightarrow \rho_1 = \rho_2$  the maximum number in system and blocking probability occurs. As shown in [9], one can prove in general that balancing the utilization of the access links results in maximum packet loss probability at the primary node queue which is a worst case scenario.

Thus, by considering  $\rho_1 = \rho_2$  case in the experimental model we can construct a conservative bounding curve below which the  $GOS$  will always be met. The experimental study used to create Figure 7 was repeated to generate a worst case curve by maintaining  $\lambda_1/\lambda_2 = C_1/C_2$  at each point then determining the load where the  $GOS$  is just met. The resulting bounding curve is shown in Figure 9. A set of bounding curves for various values of  $T$ , the time to reconnect the virtual circuits, can easily be constructed by repeating the experiment. Bounding curves such as Figure 9 are potentially quite useful for network control and could be utilized for call admission or adjustment of flow control settings after a failure.

The approach above for constructing bounding curves should be extendable to the general  $n$  virtual circuit primary node case. However the number of parameters to vary becomes difficult to handle with increasing  $n$  and we propose an algorithm for determining if congestion occurs based on the results for the two virtual circuit case. Given  $\lambda_1, \lambda_2, \dots, \lambda_n$  and  $\mu C_1, \mu C_2, \dots, \mu C_n$ , compare them two at a time then combine the two together that we have just compared. The procedure can be written in algorithmic form as follows.

- 1) Let  $\lambda_{sum} = \lambda_1$ ,  $\mu C_{sum} = \mu C_1$ ,  $j = 2$
- 2) Check two call bounding curve for  $\lambda_{sum}$ ,  $\lambda_j$ ,  $\mu C_{sum}$ ,  $\mu C_j$
- 3) If congestion occurs (i.e. above curve) then stop congestion will occur for  $n$  call case else if  $j < n$  then
 
$$\lambda_{sum} = \lambda_{sum} + \lambda_j$$

$$\mu C_{sum} = \mu C_{sum} + \mu C_j$$

$$j = j + 1$$
 go to step 2
- else if  $j = n$  then
 
$$\text{stop congestion will not occur}$$
- end if

Thus using this algorithm we can determine whether

or not congestion will occur at an arbitrary primary node for any number of virtual circuits needing to be restored. Numerical illustrations of its use are given in [9]. The results of the algorithm above tend to be conservative due to the combining of the access queues after each iteration. Note that use of the bounding curves for call admission or to set congestion controls after a failure would lead to a conservative strategy. A possibly better approach would be to allow some level of congestion at primary nodes but not enough to spread congestion to secondary and tertiary nodes.

## 5 Conclusions

In this paper we have presented a study of the performance impact of link failures in a virtual circuit wide area network. It was shown that a link failure and the subsequent traffic restoration divides the network nodes in to four groups: primary, secondary, tertiary and quaternary. A generic queueing model framework for studying the performance impact of failures at the different types of nodes was presented. It was shown that congestion after a link failure starts at the primary nodes (i.e., the source nodes of virtual circuits that were on the failed link) and spreads to the other nodes. A nonstationary performance analysis of a generic primary node was conducted using a numerical methods differential equation model approach. A set of boundary curves which specify the conditions under which a primary node will be come congested was presented and an algorithm for its general use developed.

## References

- [1] W. Falconer, "Service Assurance in Modern Telecommunications Networks," *IEEE Communications Magazine*, June, 1990.
- [2] P. Neumann, "Risks to the Public in Computers and Related Systems," *ACM Software Engineering Notes*, Vol. 17, No. 1, pp. 3-32, Jan., 1992
- [3] I. Cidon, I. Gopal and R. Guerin, "Bandwidth Management and Congestion Control in plaNET," *IEEE Communications Magazine*, October, 1991.
- [4] I. Cidon, I. Gopal, and H. Heleis "PARIS: An approach to integrated private networks" in *Proceedings IEEE Globecom '87*, 1987.
- [5] D. Tipper and M. Sundareshan, "Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions," *IEEE Journal On Selected Areas in Communications*, December, 1990.
- [6] W. Lovegrove, J.L. Hammond, and D. Tipper, "Simulation Methods for Studying Non-stationary Behavior of Computer Networks," *IEEE Journal on Selected Areas in Communications*, Dec. 1990.

- [7] B. Doshi, P. Johri, A. Netravali, and K. Sabnani, "Retransmission Protocols and Flow Controls in High Speed Packet Networks," *Proceedings of the 13th ITC*, Copenhagen, Denmark, June 1991.
- [8] T.H. Wu, *Fiber Network Service Survivability*, Boston, MA: Artech House, 1992.
- [9] D. Tipper et al., "A Study of Network Recovery from Link and Node Failures with Emphasis on Transient Behavior of Routing Algorithms," Technical Report, ECE Dept. Clemson University, Clemson, SC., July, 1992.

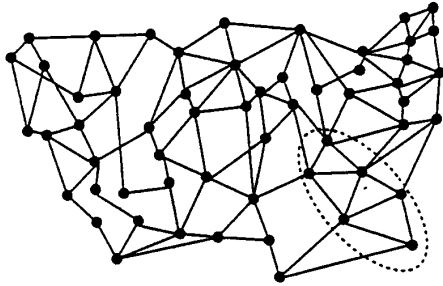


Figure 1. A Sample Packet Switched Wide Area Network Topology

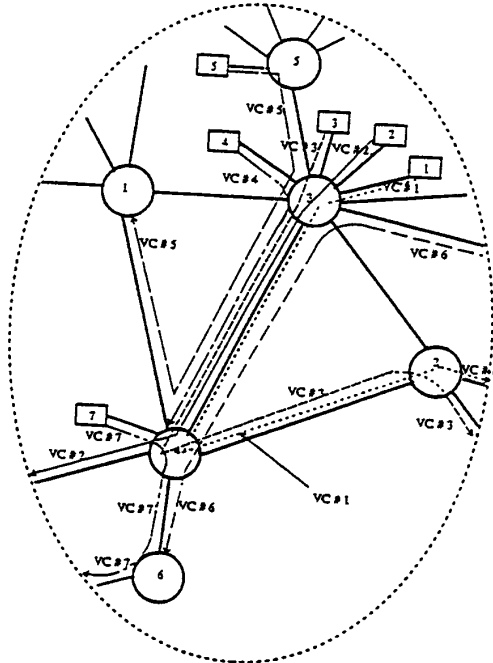


Figure 3. Traffic Pattern after Virtual Circuit Restoration due to Link 3-2 Failure

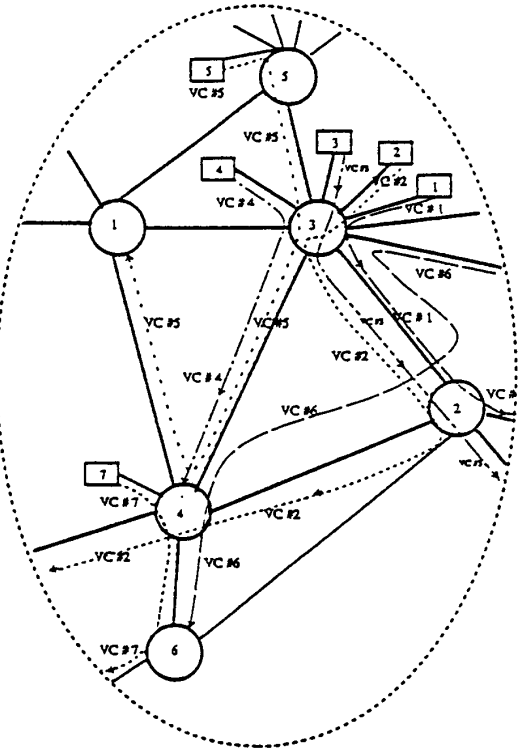
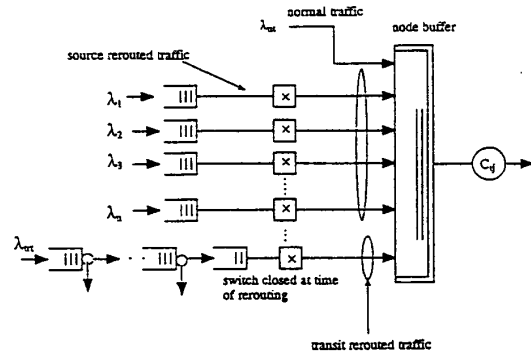
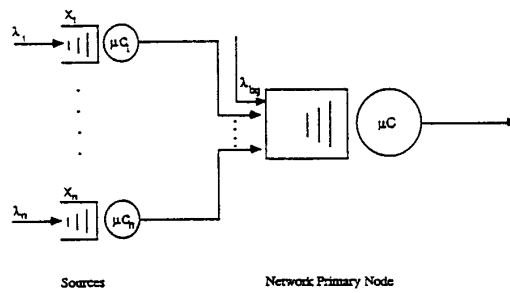


Figure 2. A Representative Set of Network Nodes



(a) Generic Model



(b) Simplified Model

Figure 4. Queuing Models of a Primary Node Output Link



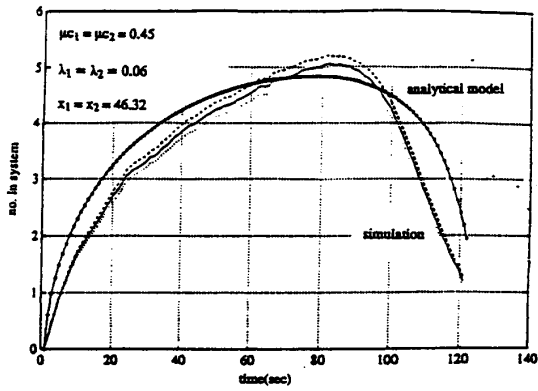


Figure 5. Validation of Analytical Model

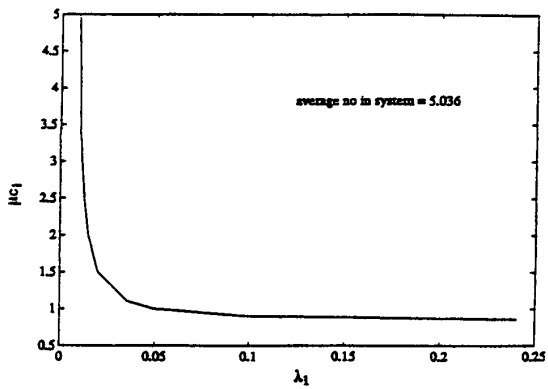


Figure 6. Boundary Curves for Congestion in Single Virtual Circuit Case

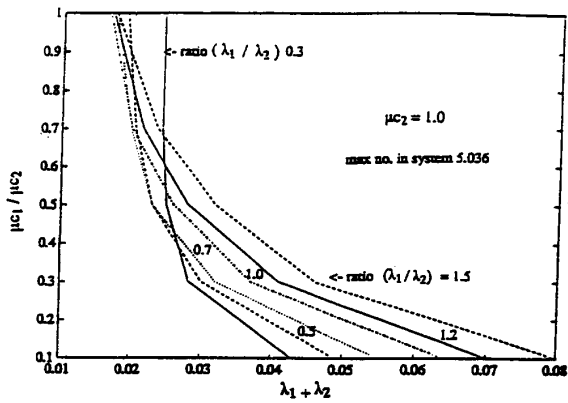
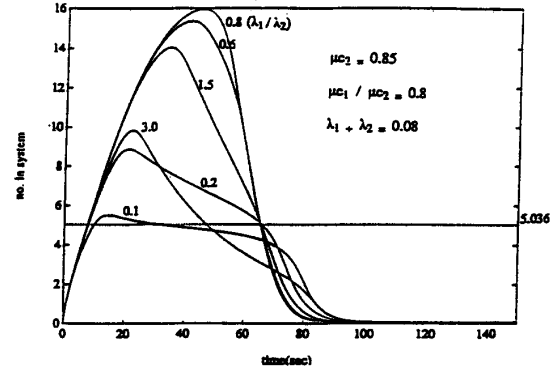
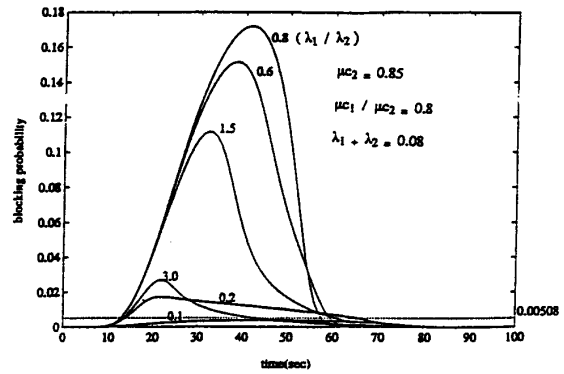


Figure 7. Grade of Service Boundary Curves



(a) Number in System at Primary Node Queue



(b) Packet Loss Rate at Primary Node Queue

Figure 8. Primary Node Queue Behavior as Function of Ratio  $(\lambda_1 / \lambda_2)$

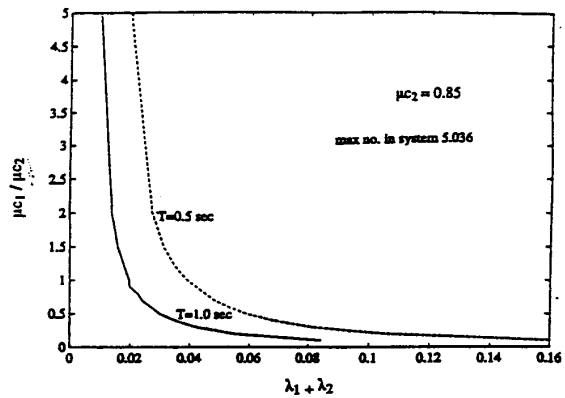


Figure 9. Boundary Curves for Congestion at Primary Node Queue

## 9a.2.9