An Analysis of the Timing of Traffic Restoration in Wide Area Communication Networks *

K.Balakrishnan, D.Tipper and J.Hammond
Electrical and Computer Engineering Department
Clemson University, Clemson, SC 29634, USA.

In this paper, we study the timing of virtual circuit traffic restoration after a link failure in wide area communication networks. The commonly used scheme of almost simultaneously rerouting the virtual circuits is compared with a scheme wherein the rerouting of the virtual circuits is staggered, thus ensuring that they do not congest the network node at the same time. We determine which approach is better by minimizing the time taken by a network node to reach steady state after the traffic restoration.

1. Introduction

The growing commercial dependence on communication networks has led to an increased focus on the reliability and survivability of these networks. The critical importance of network reliability/survivability is discussed in detail in [l]-[3]. For example, in [1], the author notes that the loss of revenue in high exposure industries due to a network outage may exceed six million dollars in unrecoverable revenue per hour of downtime.

While there have been great strides in increasing the reliability of physical network components, some rate of failure is inevitable. A network failure, such as the loss of a link or a node, can occur due to a variety of reasons causing service disruptions ranging in length from seconds to weeks. Typical events that cause failures are accidental cable cuts, hardware malfunctions, software errors, natural disasters (e.g.,fire), and human error (e.g., incorrect repair) [1], [3]. Since many of the causes of failures are outside the control of the network providers, there has been increasing interest in the design of survivable networks [l]-[3]. This work has largely focused on planning the network to reduce the impact of failures when they occur. Several techniques [3] have been proposed to minimize the effect of failures, common ones being multiple ingressing/egressing of users, trunk diversity, digital cross connect systems, and self healing ring architectures.

Note that the majority of the survivability literature concentrates on network design issues. Recently, we have begun a research effort into developing algorithms which make optimum use of network resources after a link/node failure rather than planning redundancy into the network. This effort has concentrated on virtual circuit based packet switched wide area networks such as IBM's proposed plaNET (formerly PARIS) network architecture [4], [5], which is a private high speed integrated network supporting a wide variety of traffic types. After a link failure, several network controls come into play such as

'The research reported here was supported in part by a grant from IBM, Research Triangle Park, NC.

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 which can have a significant effect on the quality of network service. In this paper, we report a study of how one may vary the time between successive reroutings from a node in order to reduce network congestion.

2. An Analysis of the Effect of Timing of Rerouting

Consider an arbitrary packet switched wide area network. We assume that the network uses virtual circuit service to transport packets and source node routing of the virtual circuits as in the plaNET network [4]. 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. After a link failure, in many virtual circuit based networks, the reconnection will be done on an individual virtual circuit basis rather than attempting facility restoration of the entire lost bandwidth. Specifically, the source nodes for the virtual circuits that were traversing the link which failed are responsible for the restoration of the affected virtual circuits. In the framework proposed in [6] for studying link failures, such source nodes are called 'primary nodes'.

After a failure, a primary node will typically have many virtual circuits to reconnect. In order to provide uninterrupted service to the affected virtual circuits, the primary node must restore the virtual circuit within approximately 2 seconds for a voice connection and within 10 seconds for a data connection [1]. Several issues in the traffic restoration process at the primary node can be critical to the network performance after restoration, namely: 1) the criterion for ordering the virtual circuits that need to be restored (e.g., highest bandwidth calls first); 2) the call admission algorithm (should it be modified to admit more or less connections?); 3) the route selected for restoration (should traffic be spread out to take advantage of the residual bandwidth or concentrated to the physical area of the failure?) and 4) the timing of the reconnection. The first three issues have been considered in [8], [6] and [7] respectively. Here, we concentrate on the problem of the timing of the rerouting.

After a failure, congestion can occur at a primary node due to virtual circuits being rerouted across a particular link at that node. Note that 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, a backlog of packets will accumulate at the source of each virtual circuit. As each virtual circuit is rerouted, it starts transmitting its entire backlog along its access link into the primary node. The link buffer at the primary node, being of a finite size, can quickly become congested. Any packet arriving at the network link queue and finding the buffer full is dropped. These packets need to be retransmitted from the source. These retransmissions add a positive feedback to the source, further worsening the congestion. Thus, the packet loss rate at the network node can become high, exceeding the grade of service (GOS) level, possibly until the backlogs of each of the restored virtual circuits is completely transmitted. As noted in [6], congestion control schemes are not entirely effective in preventing congestion after a failure since the overload at the network node

is mainly due to the rerouted virtual circuits needing to simultaneously work off their backlogs.

As a way of reducing the length of the congestion period, we propose that the virtual circuits be restored at staggered time intervals, one after the other. The basic idea is that the congestion at the network node could be reduced by restoring a virtual circuit and then waiting until the network node reaches an acceptable GOS before restoring the next virtual circuit. The virtual circuits need to be ordered in a decreasing order of their input rates. This is due to the fact that the virtual circuit to be restored last would suffer an additional backlog, directly related to its input rate, while awaiting its turn to enter the network. However, it may be necessary to give certain connections priority during the restoration phase irrespective of their input rates.

A study was carried out to determine when the staggered restoration is superior to simultaneous restoration in terms of the time for the network link at the primary node to reach GOS states. The generic primary node queueing model developed in [6] and shown in Figure 1 for the case of two virtual circuits was used for the study. Note that each virtual circuit is assigned a source queue which is modeled as an infinite buffer. In Figure 1, C represents the capacity of the primary node link, λ the aggregate mean packet arrival rate to the link and N the buffer size. We define λ_i, to be the mean arrival rate of virtual circuit i, C, as the capacity of the corresponding access link and $1/\mu$, as the mean packet length. The traffic that existed on the primary node link before the failure and any rerouted traffic from other nodes is represented by the background traffic stream with mean rate λ_{bg}. Also, we define TRC_i as the total time for the primary node to become aware of the failure and to reconnect the ith virtual circuit. Denoting the backlog of packets to be retransmitted by source i as X_i, we have $X_i = \lambda_i \times TRC_i$.

Consider the transmission of a sample backlogged packet for the *i th* virtual circuit at its source queue. The probability that the packet gets blocked on its first attempt at the primary node is P_B, the loss probability at the primary node. Assuming independence of retransmissions and that the blocking probability remains constant over the period of interest, the probability that the same packet gets blocked on its *n th* attempt is P_B^n. Thus, the total number of retransmissions is given by the equation

$$\text{Number Retransmissions} = P_B + P_B^2 + P_B^3 + \bullet\bullet\bullet = P_B \sum_{i=0}^{\infty} P_B^i = \frac{P_B}{1-P_B}.$$

Hence, the total number of transmissions NT, required for the tagged packet is $NT = \frac{P_B}{1-P_B} + 1 = \frac{1}{1-P_B}$ Therefore, the average time T_i, required for one packet to successfully be transmitted to the primary node is given by

$$T_i = (\frac{1}{1-P_B})\frac{1}{\mu C}, \tag{2}$$

where μC_i = service rate of source queue i. The time required for the *i th* source to successfully transmit the entire backlog of packets to the primary node TB_i is

$$TB_i = \frac{X_i T_i}{1-\lambda_1 T_i} = \frac{X_i}{\mu C_i (1-P_B) - \lambda_i}. \tag{3}$$

This is the time when the access node for virtual circuit i has reached steady state. However, the primary node network queue follows the behavior of the access node in a delayed fashion and we add its settling time.

A first order approximation for the number in the system varying over time for a single server queue having the proper exponential dependence for a large t is given in [9]

$$QL(t) \approx QL_{ss} + (QL(t_0) - QL_{ss})e^{-\frac{1}{\partial}(t-t_0)}. \quad (4)$$

Here QL_{ss} is the number in the system at steady state, $QL(t_0)$ is the initial value of the number in the system at time t_Q and $\frac{1}{\partial}$ is the time constant. Defining the relaxation time of the queue ΓR *as* the time taken for the number in the system to reach within 2% of the steady state value, from (4) we get

$$\Gamma R = -\partial \times \ln\left|\frac{0.02QL_{ss}}{QL(t) - QL_{ss}}\right|. \quad (5)$$

This settling time, T_R, is added to ΓB_i to model the effect of the delay at the primary node network queue. Thus the time taken for the backlog of the *i th* rerouted virtual circuit to be worked off at the network queue, TVC_i, can be determined by $TVC_i = TB_i + \Gamma_R$. Using the analysis above, we can determine the total time for the network queue to reach steady state, *TSS,* for both staggered and simultaneous restoration schemes.

Consider the staggered restoration strategy for the case of two virtual circuits and no background traffic (i.e. $\lambda_{bg} = 0$). We assume that the calls are ordered in decreasing magnitude of bandwidth (i.e. $\lambda_1 \geq \lambda_2$) and compute the time for the primary node to work off the backlog TB_1 of the first circuit. Note that until the source queue has completely transmitted its backlog, it would be sending out packets at a rate equivalent to the channel capacity, μC_1, of the access link. The next virtual circuit to be rerouted is set up after time TVC_1, resulting in $TRC_2 = TRC_1 + TVC_1$. Thus, the input rate to the network queue λ can be determined as:

$$\lambda = \begin{cases} \mu C_1 & TRC_1 \leq t \leq TRC_1 + TB_1 \\ \lambda_1 & TRC_1 + TB_1 < t \leq TRC_1 + TVC_1 \\ \mu C_2 + \lambda_1 & TRC_2 < t \leq TRC_2 + TB_2 \\ \lambda_1 + \lambda_2 & TRC_2 + TB_2 < t. \end{cases} \quad (6)$$

Using the appropriate value of the input rate, λ, we can use steady state queueing formulae to calculate the probability of blocking P_B and QL_{ss} at the primary node during each time interval. These values are then used to calculate TB_i and T_R. Thus, the time *TSS* at which the network queue reaches steady state $TSS = TRC_2 + TVC_2$.

In a similar fashion, the time *TSS* can be calculated for the simultaneous restoration of the virtual circuits. All of the virtual circuits are assumed to be restored immediately and start working off their respective backlogs at their individual access channel capacities. Hence, $\lambda = \sum_{i=1}^{n} \mu C_i$ and P_B for the network queue can be calculated. This value of P_B is used to calculate the time required by each virtual circuit to successfully transmit a packet. It can be seen that the virtual circuit with the smallest access link utilization would be the first to work off its backlog and reach steady state. On reaching steady state at a time TB_i, this virtual circuit would send packets to the network queue at a rate equal to its input rate λ_i. At this point, appropriate changes are made to reflect the new input rate to the network queue, and P_B is recalculated. This procedure is repeated until all the virtual circuits have worked off their respective backlogs at time T_{BVC}. In order to ensure that the network queue reaches steady state, a settling time, ΓR, is calculated as before

and added to T_{BVC}. Note that the time of rerouting *TRCi* is assumed to be the same for all the virtual circuits which in effect ignores the small processing time (typically a few μsec - msec) for each virtual circuit. The arrival process λ for the two virtual circuit case is given by the following equation

$$\lambda = \begin{cases} \mu C_1 + \mu C_2 & t < TRC_1 + TD_1 \\ \left.\begin{array}{l} \mu C_1 + \lambda_2 \\ or \\ \mu C_2 + \lambda_1 \end{array}\right\} & TRC_1 + TD_1 < t < TRC_1 + TD_2 \\ \lambda_1 + \lambda_2 & TRC_1 + TD_2 \le t \end{cases} \quad (7)$$

where TD_i = time to deliver the total backlog of *i* virtual circuits. For the two virtual circuit case, this is given by $TD_1 = Min\{TB_1, TB_2\}$ and $TD_2 = Max\{TB_1, TB_2\}$. Thus. $T_{BVC} = TD_2$ and *TSS* is given by $TSS = TD_2 + \Gamma R$. Note that one can easily extend the analysis of both schemes to the case of nonzero background traffic.

3. Performance Evaluation

A numerical study was conducted to determine when the staggered restoration scheme is superior to the simultaneous restoration scheme. With reference to Figure 1, we assume that packets arrive to the network according to independent Poisson processes with mean rate λ_i, for the i *th* virtual circuit being restored. Furthermore, we assume exponentially distributed packet lengths with mean l/μ 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 is finite with system size *N*. We assume that there is no congestion control on the source nodes to provide a worst case scenario. Packets which are dropped in the network are retransmitted from the traffic source using a selective repeat mechanism.

Under these assumptions, the source queues are M/M/1 and the primary node queue can be approximately modeled as an M/M/l/N queue and standard steady state queueing formulae can be used to find P_B, QL_{ss} and $QL(t_O)$ in (2) and (5). Note that to compute the primary node queue relaxation time ΓR, the time constant 1/α must be known. As discussed in [9], the time constant is a function of the utilization of the queue $p = \lambda/\mu C$. In order to accurately determine 1/α, the Chapman Kolmogorov differential equation model of the M/M/l/N queue was numerically integrated to find the 2% relaxation time and settling time constant for various values of *ρ*. A curve fit was then performed to yield the relation between 1/α and ρ.

The experimental model used here is loosely based on the plaNET network architecture [5]. Specifically, we assume each link to be a Tl line (i.e., $C = C_1 = C_2 = 1.544$ Mbps) and the average packet length to be 1/μ = 2000 bits/packet which would result in a service rate of $\mu C = 772$ packets/sec at each link. Also, we assume $N = 21$ at the primary node and the time to detect a failure and restore the first virtual circuit is TRC_1=1 second (typical times range from milliseconds to seconds). Note that the numerical values reported in this paper for virtual circuit arrival rates λ_i, are normalized with respect to the link service rate μC. Thus,

a virtual circuit with $\lambda_i = 0.25$ used in the plot would translate to a virtual circuit having $\tilde{\lambda}_i = \lambda_i \times \mu C = 386$ Kbps in the actual model.

Also, note that the times reported here have been normalized with respect to packet service times $1/\mu C$. For example, a time to reach steady state of $TSS = 1158$ in a plot would correspond to $\tilde{TSS} = TSS/\mu C = 1.5$ seconds in the actual model. In order to quantify the time during which the network node is congested, we follow the approach defined for the plaNET network which provides for a guaranteed steady state GOS for each virtual circuit that has been allowed to enter the network. The maximum flow on any network link is controlled by the link congestion thresholds *TH* in the call set up procedure. Here we assume the congestion threshold is $TH = 0.85$. Hence, under the assumption that each queue can be represented by a finite M/M/1 queue, a worst case GOS at any network link can be determined. Specifically, in the network modeled here, the maximum link utilization is $\rho = TH = 0.85$ and the system size is $N = 21$ which results in values of $P_B = 5.08 \times 10^{-3}$ and $NS = 5.03$. Thus, the guaranteed GOS at each network link is a average number in the system $NS_{GOS} = 5.03$ 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 two restoration schemes were compared in a series of numerical studies as follows. A specific ratio of the input rates of the two virtual circuits λ_2/λ_1 was chosen and for various values of the total load $\lambda = \lambda_1 + \lambda_2$, the time to reach steady state, *TSS,* was calculated and plotted for both schemes. The ratio λ_2/λ_1 was then varied and a new plot was generated. Figure 2 shows the comparison between the two schemes for a ratio of 0.8. As can be seen from the curve, at low loads, the simultaneous rerouting scheme seems to be superior to the staggered rerouting scheme. Note that there exists a crossover point after which, for further values of λ, the staggered scheme takes less time than the simultaneous scheme to reach steady state. Similar experiments were conducted for different ratios of the input rates and are given in [7].

Figure 3 is a graph plotting the input rate of the first virtual circuit λ_1, against the ratios of the two input rates. The cross-over points for each ratio are displayed as points and connected into a curve. Notice that the cross-over curve divides the entire plane into two regions, the region where the simultaneous rerouting scheme is better than the staggered rerouting scheme and the region where the reverse is true. The region for the staggered scheme is bordered by the call admission threshold of the network. Note that for small values of the ratio, the input rate λ_1 needs to be large in order for the point chosen to fall into the region for the staggered scheme. On the other hand, for large ratios of the input rates, it takes a comparably small input rate λ_1 in order for the point chosen to fall into the region for staggered scheme.

In order to determine the accuracy of the analytical model used to develop Figure 3, two points A and B from Figure 3 were selected and a detailed simulation was conducted. Details of the simulation model and the steady state analysis to validate the model are given in [7]. Note that the simulation results presented here were collected using the ensemble averaging technique given in [10] and enough runs were made (typically 5000) to obtain 95 % confidence intervals with a relative precision of at least 0.05. The confidence intervals are very small and are not presented here to preserve the clarity of the plots.

Point A was chosen with $\lambda_2/\lambda_1 = 0.4$ and $\lambda_1 = 0.35$. Figure 4a shows the ensemble average number in the system at the primary network node versus normalized time for the two restoration schemes. One can see the effect of the staggered scheme from the figure,

that is, at time zero, the first virtual circuit is restored and the second virtual circuit is restored at time 480 after the first virtual circuit has worked off its backlog. It is seen that the curve validates the analytical model in that the simultaneous scheme takes less time to reach steady state. Specifically, the simultaneous scheme reaches steady state at approximately 790, whereas the staggered scheme requires approximately 840.

Point B was chosen in the staggered region (very close to the cross-over curve) with a ratio of $\lambda_2/\lambda_1 = 0.5$ and $\lambda_1 = 0.45$. Figure 4b is a comparison of the ensemble average number in the system at the network node due to the two rerouting schemes. We see that TSS = 1780 due to the simultaneous scheme as compared to $TSS = 1850$ for the staggered scheme. However, from Figure 4b, we notice that during this period, there is a vast difference in the number in the system which leads us to favor the staggered scheme as predicted from Figure 3. Specifically, if we look at the total time the number in the system exceeds the GOS level (5.036), the staggered scheme takes approximately 185 service times less than the simultaneous scheme. Furthermore, if we look at the time the node is heavily congested, (e.g., $QL(t) \geq 15$), then the staggered scheme seems to be significantly better by approximately 550 service times. Thus, the staggered scheme would be preferred in this case, since the time taken for the network to stabilize is large and it is important to keep the level of congestion at the smallest possible value.

The analysis for the two virtual circuit model can be extended for the case of an arbitrary number of K virtual circuits needing restoration. The approach is to consider the virtual circuits in groups, two at a time and dynamically apply the algorithm developed in the previous section.

Algorithm for Optimal Restoration of K Virtual Circuits

- Sort the virtual circuits in decreasing order of their bandwidth.
- $i = 1, j = 2$.
- While $i, j \leq K$
  - Compare VC_i and VC_j using the equations for the two virtual circuit generic model. Make a decision as to the optimal reconnection timing.
  - If (Simultaneous Restoration)
    - $\lambda_i = \lambda_i + \lambda_j$
    - $j = j + 1$
    - Replace λ_i and λ_j by a combined queueing model with an input rate = $\lambda_i + \lambda_j$
  - else if (Staggered Restoration)
    - $i = j + 1$
    - $j = j + 2$
    - Compare the next two virtual circuits
  - end if

This algorithm gives conservative results, since for simultaneous rerouting, it combines the input rates of the two virtual circuits into a single queue. Thus, using this algorithm gives us an approximate result for the timing of the restoration of all the virtual circuits. The use of the algorithm is illustrated in Figure 5. Two simulations were conducted,

each with three virtual circuits, with their respective input rates given in the figure. In Figure 5a, the simultaneous and staggered scheme were compared with an intermediate restoration scheme wherein the first two virtual circuits were restored simultaneously while the third virtual circuit was restored in a staggered fashion after the network had recovered from the initial restoration. From these comparisons, it is seen that for this combination of input rates, the simultaneous restoration scheme gives the best results. This is a validation of the algorithm developed above which leads to the same conclusion. In the experiment relating to Figure 5b, by applying the algorithm, we see that an intermediate scheme consisting of simultaneously restoring the first two virtual circuits followed by staggering the third circuit until the network has reached a steady state after the initial restoration, should be the optimum scheme. It can be seen that the simultaneous and the intermediate scheme reach steady state at approximately the same time. However, the choice of the analytical model (the intermediate restoration scheme) seems to be better in terms of how much the GOS levels are exceeded.

4. Conclusions

In this paper, an analytical model was developed to determine the optimum time to reroute virtual circuits after a link failure so as to reduce the congestion. The algorithm was first developed for a model with only two virtual circuits being restored and then extended to a case with an arbitrary number of K virtual circuits. The results were validated via a simulation model and it was seen that the congestion can be curtailed using this algorithm.

REFERENCES

1. W.Falconer, "Service Assurance in Modern Telecommunication Networks", *IEEE Communications Magazine,* Vol. 28(6):32-39, June 1990.
2. J.Spragins, J.C.Sinclair, Y.J.Kung and H.Jafari, "Current Telecommunication Net work Reliability Models", *IEEE Journal on Selected Areas in Communications,* Vol. SAC-4(7):1168-1173, October 1986.
3. T.H.Wu,"Fiber Network Service Survivability", Artech House, Boston, MA, 1992.
4. I.Cidon, I.Gopal and R.Guerin, "Bandwidth Management and Congestion Control in plaNET", *IEEE Communications Magazine,* Vol. 29(10):54-64, October 1991.
5. I.Cidon and I.Gopal, "PARIS: An Approach to Integrated High Speed Private Net works", *International Journal of Digital and Analog Cabled Systems,* 1988.
6. D.Tipper, J.Hammond, S.Sharma, A.Khetan, K.Balakrishnan and S.Menon, "An Analysis of the Congestion Effects of Link Failures in Wide Area Networks", *IEEE Infocom'93,* April 1993 (also to appear in IEEE JSAC, 1993).
7. K. Balakrishnan, "An Analysis of Routing Strategies for Traffic Restoration in Wide Area Networks", M.S.Thesis, Clemson University, 1992.
8. S.K.Menon, "Petri-Net Models for Routing in Communication Networks with Appli cation to Rerouting after Failure", M.S.Thesis, Clemson University, 1992.
9. T.E.Stern, "Approximations of Queue Dynamics and their Applications to Adaptive Routing in Computer Communication Networks", *IEEE Transactions on Communi cations,* Vol. COM.27(9):1331-1335, September 1979.

10. 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,* Vol. 8. December 1990.

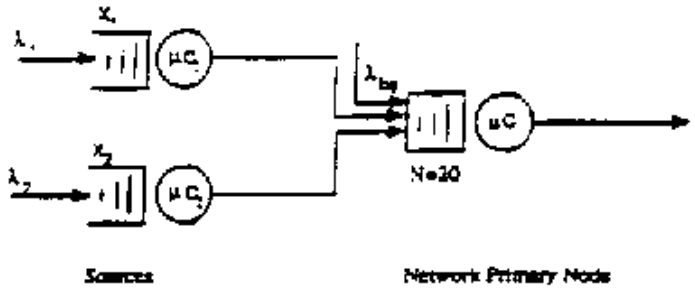


Figure 1. Generic Queueing Model with Two Virtual Circuits

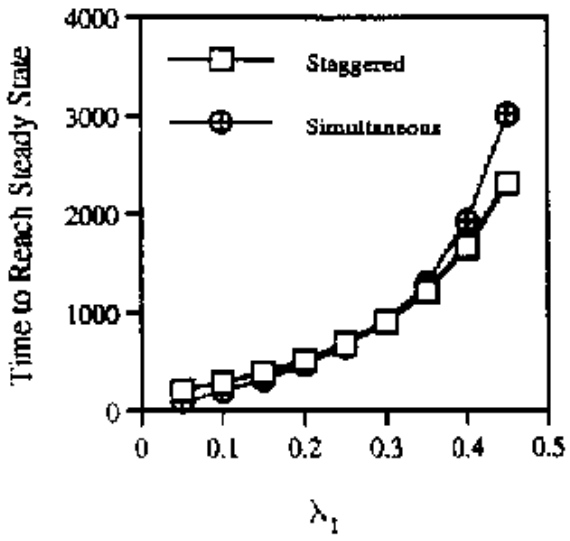


Figure 2. Comparison of Time to Attain Steady State.

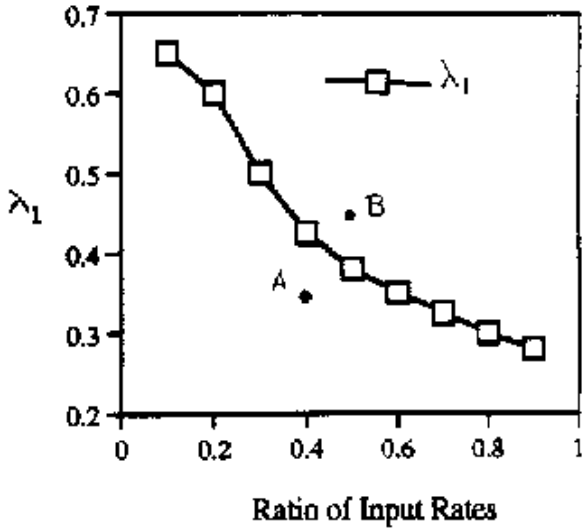
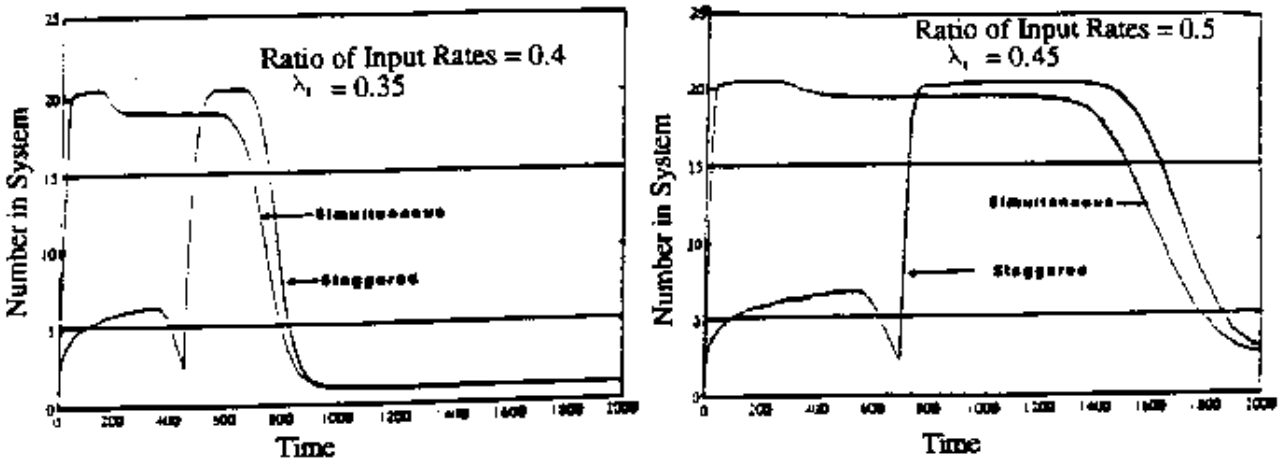


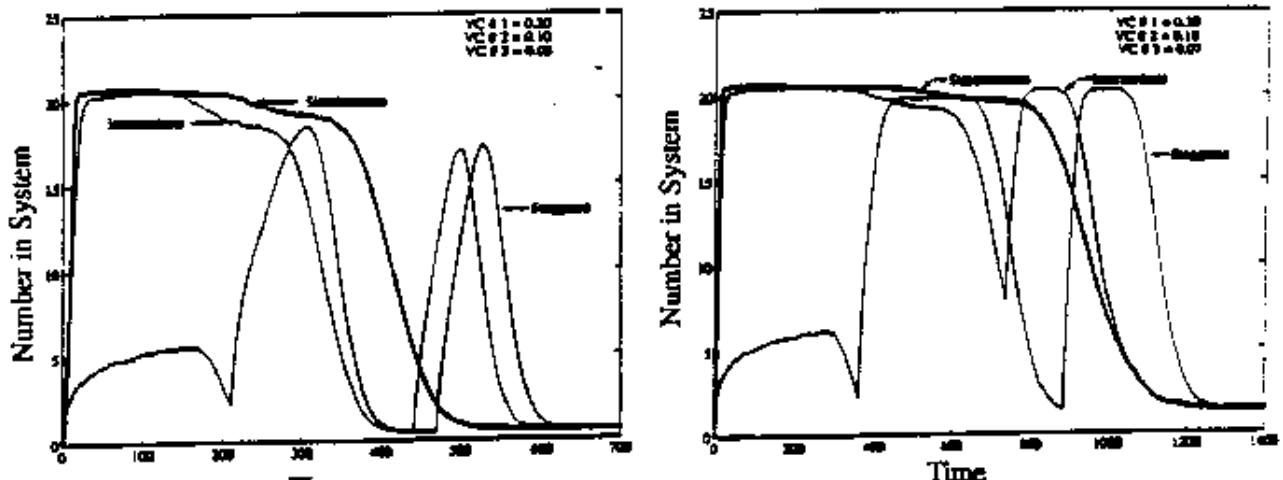
Figure 3. Boundary Curve for Selection of Time of Rerouting.



a). Validation of Point A

b). Validation of Point B

Figure 4. Validation of Points in Boundary Curve



a)

b)

Figure 5. Validation of Algorithm for 3 Virtual Circuits