Performance Prediction of Smart Permanent Virtual Circuits in ATM Networks with CACI Comnet

Tibor Gyires
Applied Computer Science Department
Illinois State University, Normal, Illinois 61790-5150

ABSTRACT

As networks becomes larger and more complex the design and management of systems become more and more difficult. New technologies, such as Asynchronous Transfer Mode, new algorithms, and bandwidth demanding applications are implemented every day. How can network designers decide which combinations of connections, communication speeds, algorithms, and applications are the most appropriate for their business? A widely accepted answer is performance prediction through simulation. A model can be used to

- evaluate various design alternatives or various operational policies,
- explore the behavior of proposed systems and connections before actually building them,
- pre-test modifications.

We apply simulation techniques for ATM networks. In an ATM network the basic data units, called cells, are routed through switched or permanent virtual circuits. A Smart Permanent Virtual Circuit is a connection that looks like a Permanent Virtual Circuit at the local and remote endpoints with a Switched Virtual Circuit in the middle. In an earlier paper we presented a search algorithm [3] for managing Smart Permanent Virtual Circuits. Our goal is to predict the performance of this search algorithm with the Distributed Software Module of the simulation modeling system CACI Comnet [4].

1. INTRODUCTION

In an ATM network the basic data units, called cells, are routed from a source to a destination through switched or permanent virtual circuits (or virtual channels) that can be bundled into virtual paths. A Switched Virtual Circuit is used to transport information between two locations and lasts only for the duration of the transfer. Permanent

Virtual Circuits (PVCs) are used for dedicated long-term information transport between locations. Establishment of virtual circuits and paths are based on a negotiation between the calling party and the ATM network. The cell routing is controlled by the ATM switches along the path between two ATM endpoints. A Smart PVC [1] is a connection that looks like a PVC at the local and remote endpoints with a Switched Virtual Circuit in the middle. Smart PVCs are more robust than PVCs. If a link carrying a PVC goes down, then the circuit goes down. If a link carrying a Smart PVC goes down and there is an alternate route, then the originating switch automatically reroutes the Smart PVC around the link. As a result of the rerouting, the network may not be able to deliver the promised quality of service as it was negotiated in the contract. The network may have to change the quality of service parameters negotiated for other connections.

One of the current research directions solves the rerouting problem by using Distributed Artificial Intelligence approaches to select between alternative rerouting plans [2]. We also presented a rerouting algorithm in [3] using a search algorithm. The objective of the search algorithm is to find a restoral path with the least cost. Although the performance of the algorithm was analyzed in [3] by estimating its space and time complexity, we are going to measure and predict its performance in an environment that is close to a real network. The paper presents a modeling environment by using the Comnet simulation modeling tool by CACI that is appropriate to validate the theoretical results in [3] and other distributed network algorithms. The goal of simulation modeling of Smart PVCs is to

- represent the topology and functions performed by the switches,
- obtain statistical results about the algorithm's and network's performance, and
- build a tool set for modeling other problems related to the Fast Resource Management issues in ATM networks.

The rest of this paper is organized as follows: Section 2 contains an overview of the search algorithm. Section 3 describes the simulation techniques. The results of the algorithm's and network's performance are presented in Section 4 followed by the conclusion.

2. OVERVIEW OF THE SEARCH ALGORITHM

For illustration we use the topology of Figure 1.

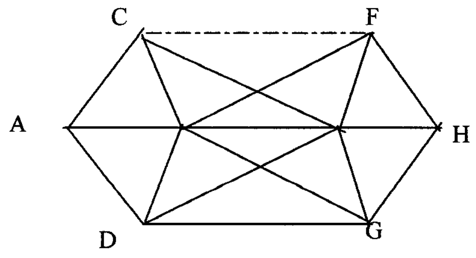


Figure 1. Network topology and link failure between switches B and E

A, B, C, D, E, F, G, and H denote the switches. A and H are the two endpoints of a PVC {A, B, E, H}. We assume a link failure between B and E. The rerouting process is initiated by switch A that established the circuit before the failure. A sends a request to restore the PVC to adjacent switches along with an upper bound on the rerouting cost and the circuit characteristics, such as service categories, traffic parameters, and quality of service parameters. A switch is qualified if it has been able to restore similar circuits in previous cases. The switches compute the estimated cost of each of the activities associated with the rerouting process, such as the allocation or reallocation of network resources, rerouting existing but lower priority connections, and reestablishing interrupted connections. These cost estimates are sent back to A, which will select the route with the smallest cost estimate, e.g., via D. The selection of a switch is based on a performance parameter P which is a measure of a switch's inefficiency (as calculated by its predecessor switch along the rerouting circuit) based on its previous reroutings. Small (close to zero) values of P indicate high quality and efficiency, and high values indicate that the switch is unreliable and unstable in rerouting the circuit. A switch's P value is recalculated after each circuit rerouting by that adjacent switch which sent a rerouting request to it. The recalculation is based on the statistical evaluation of the switch's estimated and actual restoration costs [12]. The motivation for using P is to enable the search process to learn from past performances and use this knowledge to select the most efficient rerouting switches.

The process continues recursively. At each recursive step a switch uses three arguments: the circuit number, the subsequent site, and the upper bound on the restoral cost. Each step expands the path by those children through which the estimated restoral costs do not exceed the upper bound. Each step returns the estimated cost along the path to a child, replacing parent values with the minimum of the estimated costs via the last children expanded, going back along the path, until a better cost estimate is reached. Then, the procedure continues along that path. After a recursive step this cost value will be equal to the minimum estimated cost path to the last child along the expanded subtree.

Once the destination H is reached the process stops. The rerouted circuit is along the return path of the recursive steps. Selection of the rerouted circuit by the participating switches is based on cost estimates and information about the switches' previous performance. Since at each step the estimated least-cost path is selected, the resulting restoral path is the expected least-cost path or close to it. It can be proved that the above algorithm always finds the least-cost rerouting circuit if some conditions are met [7].

3. SIMULATION TECHNIQUES

We constructed the following simplified model in Comnet:

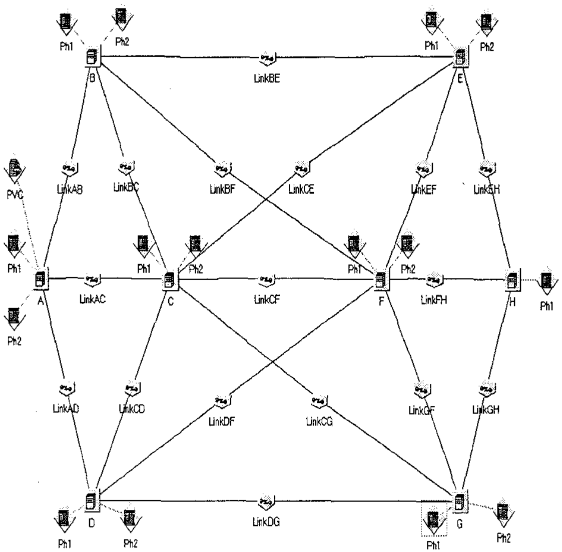
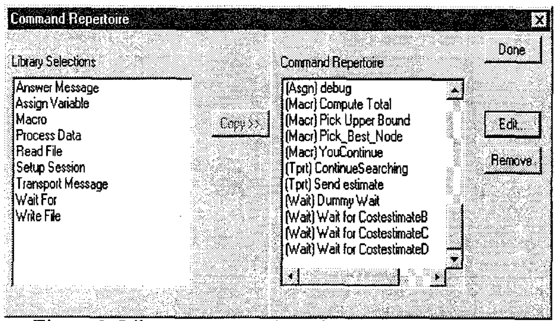


Figure 2. Model of the ATM network

In Figure 2, A, B, C, D, E, F, G, and H denote the switches of the ATM network connected by OC-3 links LinkAB, LinkBE, etc. A is the origin, H is the destination of an ATM Smart PVC. The circuit is represented by a session modeling construct called PVC attached to switch A. (In Comnet a session establishes a connection between two endpoints.) Each switch can execute commands and define variables. Commands and variables can be local ones accessed by a switch locally or global ones used by any node it the network. A macro is a named collection of commands. The following screen shows the available library commands and the ones defined for switch A:



Command Repertoire

Library Selections

Answer Message
Assign Variable
Macro
Process Data
Read File
Setup Session
Transport Message
Wait For
Write File

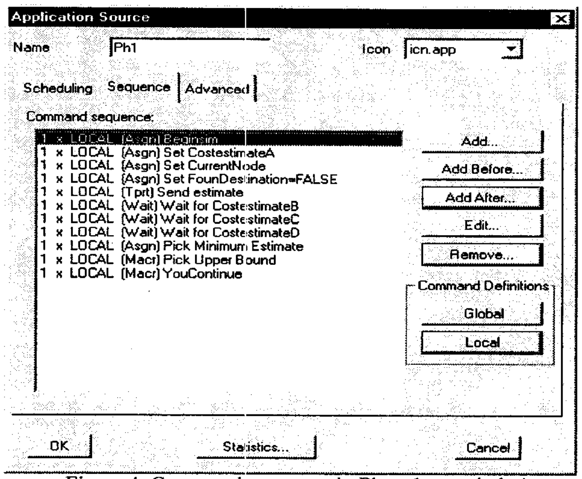
Command Repertoire

(Asgn) debug
(Macr) Compute Total
(Macr) Pick Upper Bound
(Macr) Pick_Best_Node
(Macr) YouContinue
(Tprt) ContinueSearching
(Tprt) Send estimate
(Wait) Dummy Wait
(Wait) Wait for CostestimateB
(Wait) Wait for CostestimateC
(Wait) Wait for CostestimateD

Figure 3. Library commands and some commands of A

Commands are executed by modeling constructs called applications. Ph1 and Ph2 connected to a switch are applications. They model two phases of the rerouting algorithm corresponding to two recursive steps.

The next screen for instance, depicts the order of Ph1's commands at switch A:


Application Source

Name Ph1

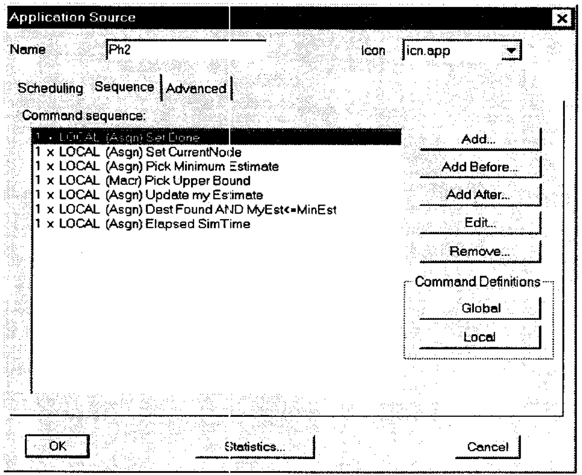
Icon icn.app

Command sequence:

```
1 x LOCAL (Asgn) Beginsim
1 x LOCAL (Asgn) Set CostestimateA
1 x LOCAL (Asgn) Set CurrentNode
1 x LOCAL (Asgn) Set FounDestination=FALSE
1 x LOCAL (Tprt) Send estimate
1 x LOCAL (Wait) Wait for CostestimateB
1 x LOCAL (Wait) Wait for CostestimateC
1 x LOCAL (Wait) Wait for CostestimateD
1 x LOCAL (Asgn) Pick Minimum Estimate
1 x LOCAL (Macr) Pick Upper Bound
1 x LOCAL (Macr) YouContinue
```

Figure 4. Command sequence in Phase1 at switch A

We give a brief explanation of some commands. For specific details please see [4]. The "Set CostestimateA" command assigns the initial estimated value of A to the global variable "CostestimateA". "Set CurrentNode" sets the global variable "CurrentNode" to "A" to find the backward path from the destination H. "Send Estimate" transport command multicast the "Send Estimate" message to A's neighbors. "Wait for CostestimateB" command waits for an "Answer with CostestimateB" message from switch B. "Pick Minimum Estimate" computes the smallest estimated costs of the neighbors. The "Pick Upper Bound" macro computes the upper bound in the recursive step of the search algorithm. "YouContinue" will redirect the search path to the switch with the minimum estimated cost. The next screen displays the commands executed in phase two by the Ph2 application:


Application Source

Name Ph2

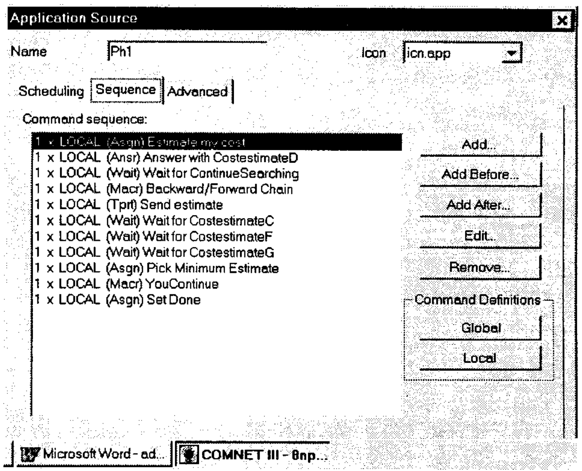
Icon icn.app

Command sequence:

```
1 x LOCAL (Asgn) Set Done
1 x LOCAL (Asgn) Set CurrentNode
1 x LOCAL (Asgn) Pick Minimum Estimate
1 x LOCAL (Macr) Pick Upper Bound
1 x LOCAL (Asgn) Update my Estimate
1 x LOCAL (Asgn) Dest Found AND MyEst<=MinEst
1 x LOCAL (Asgn) Elapsed SimTime
```

Figure 5. Command sequence in Phase2 at switch A

Ph2 is scheduled to run upon the receipt of the message "Answer with Total CostEstimate" sent by the neighbors after the first recursive step in the search algorithm. Ph2 either redirects the search path or terminates the algorithm. The "Dest Found AND MyEst<=MinEst" implements this decision. Phase 2 terminates if the destination has been found and A's estimate (MyEst) is still the least estimated cost among others. Otherwise the search algorithm continues via the switch with the least-cost estimate. Intermediate switches act similarly. They execute commands corresponding to the rest of the search algorithm. There are few commands specific to intermediate switches as it is shown in Figure 6. For instance, the "Backward/Forward Chain" macro in Ph1 of switch D maintains a list of neighboring switches to keep track of the search path. The "Answer with CostestimateD" command in switch D answers to the message "Send Estimate" sent earlier by one of its neighbors.


Application Source

Name Ph1

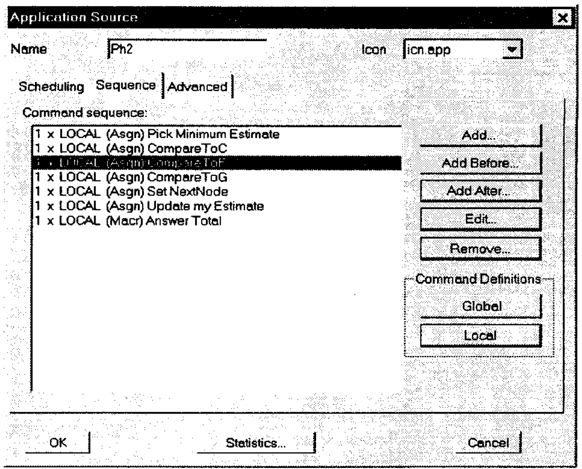
Icon icn.app

Command sequence:

```
1 x LOCAL (Asgn) Estimate my cost
1 x LOCAL (Ansr) Answer with CostestimateD
1 x LOCAL (Wait) Wait for ContinueSearching
1 x LOCAL (Macr) Backward/Forward Chain
1 x LOCAL (Tprt) Send estimate
1 x LOCAL (Wait) Wait for CostestimateC
1 x LOCAL (Wait) Wait for CostestimateF
1 x LOCAL (Wait) Wait for CostestimateG
1 x LOCAL (Asgn) Pick Minimum Estimate
1 x LOCAL (Macr) YouContinue
1 x LOCAL (Asgn) Set Done
```

Figure 6. Commands of Phase 1 at intermediate switch D

The macro "Answer Total" in the next figure returns the total estimated cost of the current search path.


Application Source

Name Ph2

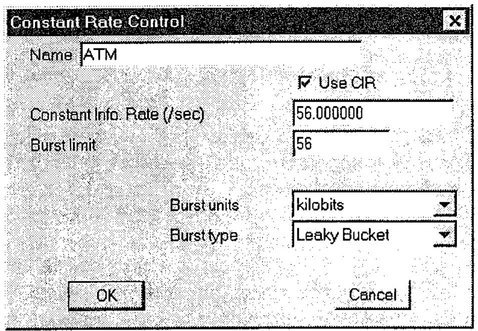
Icon icn.app

Command sequence:

```
1 x LOCAL (Asgn) Pick Minimum Estimate
1 x LOCAL (Asgn) CompareToC
1 x LOCAL (Asgn) CompareToF
1 x LOCAL (Asgn) CompareToG
1 x LOCAL (Asgn) Set NextNode
1 x LOCAL (Asgn) Update my Estimate
1 x LOCAL (Macr) Answer Total
```

Figure 7. Commands of Phase 2 at intermediate switch D

The links are constant rate ATM connections:



Constant Rate Control

Name ATM

☑ Use CIR

Constant Info. Rate (/sec) 56.000000

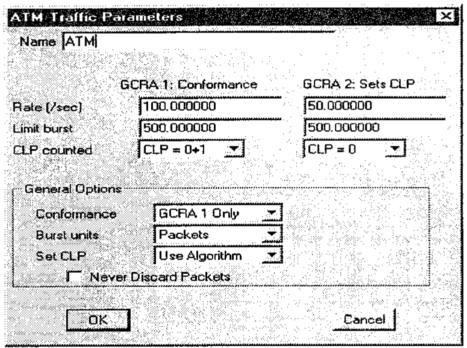
Burst limit 56

Burst units kilobits

Burst type Leaky Bucket

Figure 8. Constant Rate ATM connection

The traffic policing parameters are defined as follows:



ATM Traffic Parameters

Name ATM

	GCRA 1: Conformance	GCRA 2: Sets CLP
Rate (/sec)	100.000000	50.000000
Limit burst	500.000000	500.000000
CLP counted	CLP = 0+1	CLP = 0

General Options

Conformance GCRA 1 Only

Burst units Packets

Set CLP Use Algorithm

□ Never Discard Packets

Figure 9. The Traffic policing parameters

The simulation starts with establishing a Smart PVC between A and H. The small animated rectangle on LinkBE shows the data flow along the ABEH path:

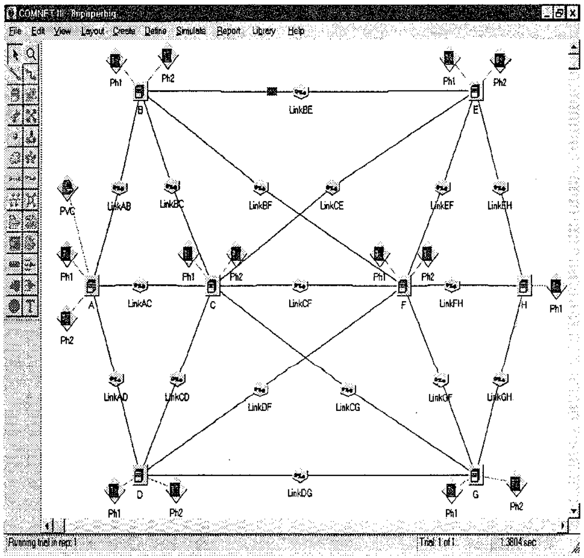


Figure 10. PVC data flow along the path ABEH

After a random period of time LinkBE fails:

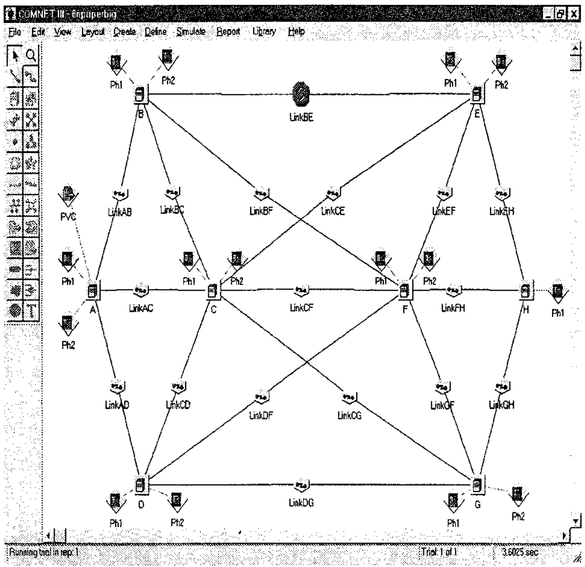


Figure 11. Link failure between switches B and E

The failure triggers the execution of the recursive steps of the rerouting algorithm implemented by applications Ph1 and Ph2 at switch A. The numbers in the following screen indicate the estimated least-cost rerouting path. The PVC is switched to path ADFH with total estimated-cost 11.

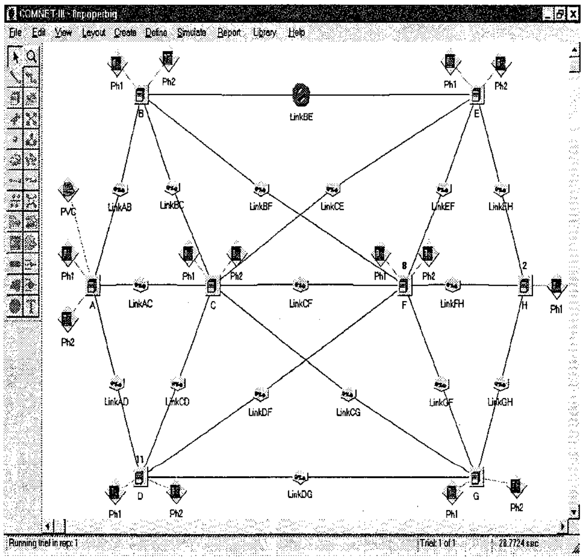


Figure 12. Estimated least-cost rerouting path ADFH

4. PERFORMANCE MEASUREMENT

In order to measure the performance of the rerouting algorithm we collect the following statistics for demonstration purposes:

- Application run time to find a new route. The rerouting time should be small relative to the message delay.
- Link utilization by the applications
- Number of messages generated during the rerouting process.

First, we compare the PVC setup time between the source and destination switches and the time needed to find a new route. The rerouting time should not be significantly greater than the setup delay. The figure below indicates that the setup time is around 0.003 second.

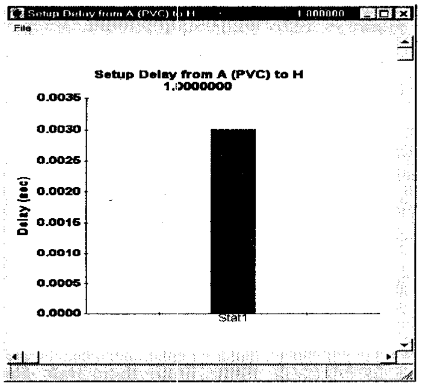
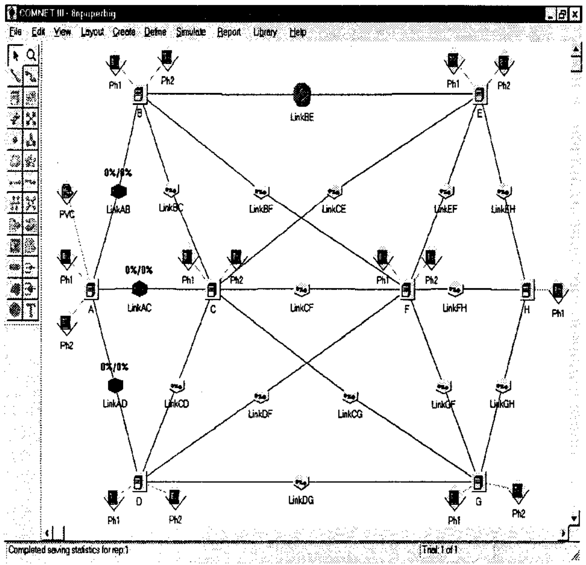


Figure 13. Setup delay between A and H

The execution of Ph1 and Ph2 took 0.007 seconds in simulation time.

Another measurement that characterizes the algorithm's performance is link utilization. A low link utilization indicates that the rerouting algorithm doesn't generate excessive traffic on the network. On the next figure we printed only the utilization of the links connected to switch A. Close to 0% utilization indicates that the rerouting algorithm requires only insignificant portion of the bandwidth (The precise value is 0.01%):

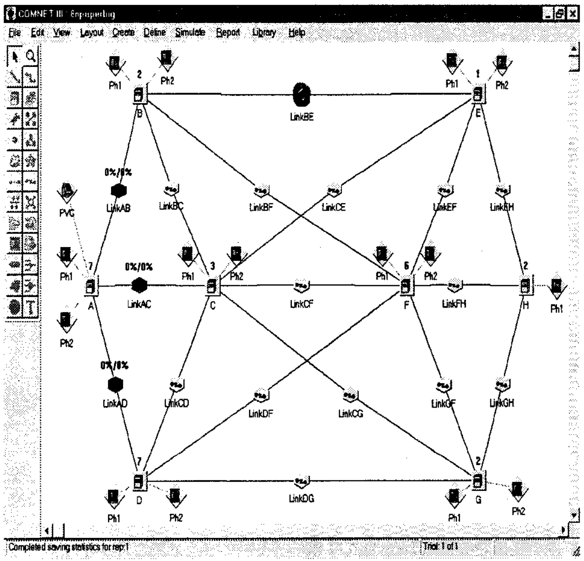


Completed saving statistics for rep:1
Trial: 1 of 1

Figure 14. Utilization of links connecting switch A

Similarly, the number of messages created during the rerouting algorithm can be considered as a measure of performance. Low number of messages is the indication of low algorithm's overhead.

The numbers above the switches in the following figure show the total number of messages created during the algorithm:



Completed saving statistics for rep:1
Trial: 1 of 1

Figure 14. Total number of messages created

5. CONCLUSION

We presented a modeling technique to predict the performance of Smart Permanent Virtual Circuits in ATM Networks. We applied the Distributed Software Module of the simulation modeling system CACI Comnet. The technique can be applied in other distributed systems where there is a need to measure the performance of network algorithms. The same technique can also be used to validate the theoretical results on performance parameters of distributed algorithms and systems.

6. REFERENCES

[1] ForeRunner ATM Switch Configuration Manual, FORE Systems, Inc., MANU0066, March, 1996

[2] Conry, S., Kuwabara, K., Lesser, V., "Multistage Negotiation for Distributed Constraint Satisfaction," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, November/December 1991, pp.1462-1477.

[3] Tibor Gyires, "Smart Permanent Virtual Circuits in Asynchronous Transfer Mode Networks," Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, October 12-15, Orlando, Florida, pp. 4460-4464.

[4] Comnet III, Planning for Network Managers, Release 1.4, 1997 CACI Products Company, 3333 North Torrey Pines Ct., La Jolla, California 92037

[5] Tibor Gyires, "Intelligent Routing Agents in Wide-Area Networks," in "Intelligent Agents for Telecommunications Applications," edited by: H. Velthuijsen (KPN Research, The Netherlands) and S. Albayrak (Technical University of Berlin, Germany), IOS Press, 1997.

[6] Leland, W., Taqqu, M., "On the Self-Similar Nature of the Ethernet Traffic," IEEE Transactions on Networking, Vol. 2, no. 1, February 1994, pp.1-15.

[7] Korf, R., E., "Iterative-Deepening-A*: An Optimal Admissible Tree Search," Proceedings of the Ninth International Joint Conference on Artificial Intelligence, UCLA, August 18-23, 1985, pp. 1034-1036.

[8] Dechter, R. and Pearl, J., "The optimality of A* revisited," Proceedings of the National Conference on Artificial Intelligence, Washington, DC, August, 1983, pp. 95-99.

[9] Durfee, E., "Coordination of Distributed Problem Solvers," Kluwer Academic Publishers, 1988.

[10] Korf, R., E., "Depth-First Iterative-Deepening: An Optimal Admissible Tree Search," Artificial Intelligence, Vol. 27, 19985, pp. 97-109.

[11] Bond, A., and Gasser, L., "Readings in Distributed Artificial Intelligence," Morgan Kaufmann Pub. Inc., 1988.

[12] Tibor Gyires and Krishnan Muthuswamy, "A Bidirectional Search Approach for Restoring Circuits in Communication Networks," Winter 1997 issue of The Journal of Computer Information Systems, pp. 85-95.

[14] Sycara, K., Roth, S., Sadeh, N., and Fox, M., "Distributed Constrained Heuristic Search," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, November/December 1991, pp. 1446-1461.