Simulation and Performance of PNNI ATM Networks

Yunming Song, David Cypher, and David Su

High Speed Network Technologies Group
National Institute of Standards and Technology
100 Bureau Drive, Stop 8920
Gaithersburg, MD 20899-8920
yunming.song@nist.gov
david.cypher@nist.gov
david.su@nist.gov

Abstract

This paper presents the simulation and performance of PNNI ATM networks by using an ATM PNNI Routing Protocol Simulator (APRoPS). This simulator was developed based on the ATM Forum's PNNI routing protocol specification. With this simulator, one can design and test almost any network configuration and evaluate its performance within the constraints of the physical machine running the simulation. Therefore one can evaluate the scalability, robustness and maintainability of PNNI routing status.

Using a small ATM network test bed, we conducted a performance study of various vendors' implementations of PNNI to obtain some measurements and to test the ability of the simulator. The output of the numerous simulation runs and test bed performance measurements are used to make suggestions for PNNI parameters, and to provide observations that may be useful in designing ATM networks using PNNI.

1. Introduction

Designers and administrators of networks who want to use ATM technology must decide on a signaling system. One such choice is the signaling system defined in the ATM Forum's Private Network-Network Interface (PNNI) specification [4][5]. This specification defines a signaling system complete with a dynamic routing system. However, the complexity and size of this specification are a deterrent for using this system. Some factors that are applicable for any network are the organization of, number of, and distance between ATM switches. Specific factors for PNNI alone are relevant protocol timers, packet sizes, number of levels of hierarchy and many others. All factors relate to the scalability, robustness, and maintainability of the network.

With the number of variables ever increasing and given the cost of ATM equipment, large-scale test beds are cost prohibitive. Another factor is the number of unknown values for PNNI parameters. These are due, in part, to the lack of knowledge about their interactive effects. Therefore simulators have become mandatory for investigating network performance and design. Still small test beds are important.

Currently there are a few papers that study the performance of PNNI ATM networks. Sivabalan and Mouftah present a discrete-event simulator, called QUARTS-II, for route calculations for call establishments [1]. Tunpan primarily focuses on the adaptability of the PNNI routing protocol to a single change that occurs in the ATM networks [3].

We have developed a simulator (APRoPS) for evaluating PNNI ATM networks. This simulator allows us to design and test almost any network configuration and evaluate its performance within the constraints of the physical machine running the simulation. The goal of this simulator is to provide a tool, which can be used to evaluate performance of the PNNI routing protocols within any network configuration. The performance evaluation may be made using fixed PNNI parameters while varying the network configuration, or by varying the parameters with a fixed network configuration. A combination of both can be done also to achieve optimal performance and design evaluations.

Many evaluations are possible when using this simulator. Evaluations include the time it takes for the PNNI routing protocols to reach stability and the amount of data required for PNNI operations alone, both in reaching stability and for maintaining stability. Other evaluation concerns are the amount of time and data needed for PNNI routing protocols to discover a failure and to restore from that failure. We will be investigating the following items:

- How much time is needed for the PNNI routing protocol to reach stability?
- How much data must be exchanged for the PNNI routing protocol to reach stability?
- How much data must be exchanged to maintain the PNNI routing status, once initial stability is reached?
- How quickly does the PNNI routing protocol recover from link or node failures?

Note that stability is defined as the state where each of the PNNI routing protocols has reached an operating mode, which is considered a final state.

Using a small ATM network test bed, we did a performance study of the various vendors' implementations of PNNI. From this performance study we obtained values that are used as input parameters into the simulator. We did this to test the ability of the simulator to produce real world output when provided with real world input.

The output of the numerous simulation runs and test bed performance measurements are used to make suggestions for PNNI parameters, which are currently undefined in the specification, and to provide observations which may be useful in designing ATM networks using PNNI.

This paper is organized as follows. The next section introduces the PNNI routing protocol. The third section describes the ATM PNNI routing protocol simulator and presents some assumptions. The fourth section describes our small test bed and measurements. The fifth section reports the results of simulation and performance study of PNNI ATM networks using the simulator.

2. PNNI Routing Protocol

In a PNNI routing domain, a lowest-level node (i.e., switch) belongs to a certain *peer group* (PG). A peer group consists of nodes having the same pre-configured peer group identifier (ID). A link connecting two nodes can be classified as either a *horizontal link* or an *outside link*, depending on whether the link between the two nodes is within the same peer group or not, respectively. If two connected nodes have different peer group IDs, they are considered *border nodes* with respect to their peer group. Each node runs the Hello protocol on its links to determine the status of the links and the status of neighbor nodes. The Hello protocol also serves as a way to exchange the node’s state information. By exchanging Hello packets, a node learns about its neighboring links and their type.

Once a node has learned that its neighboring node is in the same peer group, the node synchronizes its *topology database* with that node. The result is that the two nodes have identical topology databases. Then each node distributes, via a mechanism called flooding, the *PNNI Topology State Elements (PTSEs)* to the other nodes in the same peer group to which it has already synchronized. Information in PTSEs provides topology state parameters describing the characteristics of nodes and links, and thus allows all nodes in the same peer group to share the same view of the peer group. The flooding mechanism is done reliably by encapsulating a number of PTSEs in a *PNNI topology state packet (PTSP)* and requiring an acknowledgment of each PTSP. PTSEs are subject to aging; a PTSE will expire after a given period, so the node must periodically send updated PTSEs to the other nodes.

From time to time, the nodes within the same peer group elect a *Peer Group Leader (PGL)* to function as a *logical group node (LGN)* representing the peer group as *a single logical node* in a higher hierarchical level. In a peer group, the node with the highest leadership priority will become a PGL; if two or more nodes have the same leadership priority, then the node with the highest node identifier wins. An LGN plays a key role in building the hierarchy by summarizing the information in the peer group to which it belongs, and by distributing the topology state information to the next higher hierarchical level.

LGNs at the same hierarchical level are connected by *logical links*. A logical link connecting two LGNs is an accumulation of lower-level horizontal links and outside links (i.e., to provide a path between two PGLs of lower-level peer groups). An LGN also runs the Hello protocol to determine the status of its logical links and neighbors, and to find the LGN's peer group neighbors. An LGN also belongs to a peer group at its level. An LGN proceeds in the same manner as the lower level node: synchronizes topology database with its peer group neighbors, reliably floods PTSEs to the other LGNs in the same peer group, and elects a PGL for this LGN's peer group to function as a LGN at the next higher hierarchical level. This process is repeated until the whole network becomes one highest-level logical peer group.

Information can flow in the hierarchy in different manners. A PGL collects information within its peer group, summarizes the peer group information and feeds the summarized version up the hierarchy (to the corresponding LGN). PTSEs of nodes at a level in a peer group can flow in the following two ways. One, those PTSEs can be horizontally flooded within the peer group. Two, the PTSEs are sent downward through the hierarchy by the LGN (also functioning as a PGL at one level below) to be flooded within the PGL's peer group (one level under the LGN).

3. ATM PNNI Routing Protocol Simulator

The ATM PNNI Routing Protocol Simulator (APRoPS) was developed at the National Institute of Standards and Technology (NIST) to provide a flexible test bed for studying and evaluating the performance of PNNI ATM networks. This simulator is a tool that gives the user an interactive modeling environment with a user-friendly interface. With this simulator the user may create different network topologies, control component parameters, and measure network performance.

This simulator is written in the "C" programming language to execute on a Sun SPARCstation using SunOS Release 5.5.1. The simulator software is designed in a modular fashion using a number of building blocks, including an initialization module, a user interface module, a control module, an event manager module, and a number of protocol modules. In order to implement the ATM PNNI simulator, we made some assumptions. For example, a failure at a node or a link at any hierarchical level will always cause a detectable time-out at the other node waiting for processing with the failed node/link. The complete details are described in Song [2]).

For our performance study there are two main steps that we are concerned with. These are the following test steps:

- "**Initial Connection (Startup)**" means to observe the simulation results from starting the physical connection until the PNNI routing protocol reaches stability. The physical connection means first link or node up (or even all links and nodes are active at the same time) with the corresponding "Link Up" event of the Hello Finite State Machine (FSM). The stability is defined to be the state where each of the PNNI routing protocols has reached a final operating mode. In more detail, with one peer group at the lowest level the stability state means that the Neighboring Peer states of all nodes are "Full", if no PGL exists, or that PGL election is completed, if a PGL exists; with two or more peer groups at the lowest level, the stability state means that the PNNI routing hierarchy is established, if it exists.

- "**Maintainability**" means to observe the maintainability of the PNNI routing status after initial stability is reached.

4. Real Implementation Performance Study with Measurements

Using a small test bed, we obtained some basic input parameters for our simulator. Our test bed consists of four ATM PNNI capable switches. The two logical configurations are shown in Figure 1. Using a protocol analyzer, we monitored the network links and made some measurements.

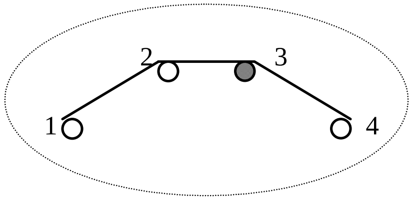

Figure 1a. Network Configuration A

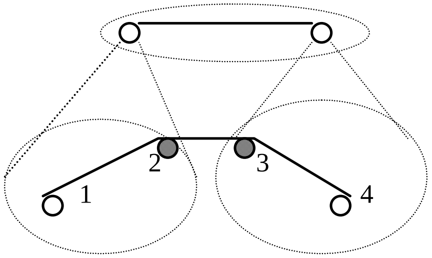

Figure 1b. Network Configuration B

Network configurations A and B have the same physical connections. Configuration A has a single PGL elected (i.e., node 3) within a single peer group with no routing hierarchy. Configuration B has two PGLs elected (i.e., node 2 and node 3), one for each of the two different peer groups, and a routing hierarchy is configured.

We monitored these configurations for various time periods. From these observations of this test bed, we get the following measured values:

1) The measured PTSE refresh interval ranges from about 1,500 seconds to 2,100 seconds.
2) The measured node processing time for different packet types is approximately:

Hello packet	0.1 second
Database Summary (DBS) packet	0.3 second
PTSE Request (REQ) packet	0.5 second
PTSP packet	0.5 second

3) The measured Hello interval is between 12~18 seconds, which is within the +/- 25% of the default Hello Interval of 15 seconds.
4) It takes approximately two hours to observe the periodic PNNI exchanges to maintain routing stability.

5. Simulation and Performance Study of the PNNI ATM networks

We use two pre-defined network topologies with the same physical connections, but with different logical configurations to study the performance of the PNNI routing protocols. The first network configuration (C), shown in Figure 2a, has a PGL elected (i.e., node 5) within a single peer group and with no routing hierarchy. The second network configuration (D), shown in Figure 2b, has two PGLs elected (i.e., node 3 and node 5), one for each of the two different peer groups, and the routing hierarchy is configured. The former configuration is typical of today's implementations (a flat network). The latter configuration is the first step towards building future implementations (a hierarchical PNNI network).

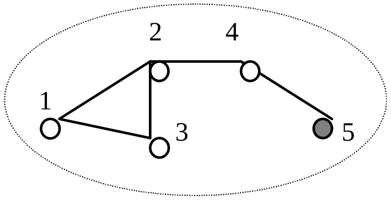

Figure 2a. Network Configuration C

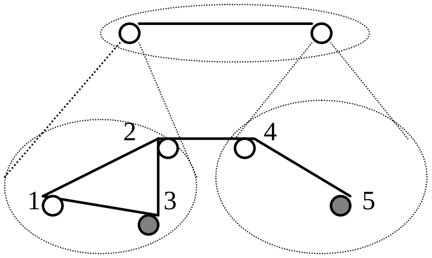

Figure 2b. Network Configuration D

We assume the following for both network configurations:

- Each node has the same node processing time for the same packet type, and each physical link has the same delay. The default node processing time for Hello packet, Database Summary packet, PTSE Request packet, PTSP packet are 0.1, 0.3, 0.5 and 0.5 second, respectively. The default link delay is 0.0001 second. The default Hello interval value is 15 seconds.
- One PTSP bundles all database information stored in a node; each node has the same PTSE refresh interval. PTSE refresh interval is 1,800 seconds.
- All nodes and links are active at once, and the starting global clock is 0.0 second.
- The simulated clock time is 9,000 seconds, i.e., two and a half hours, for studying the maintainability of the PNNI routing status.

We have different stability definitions for each of the different logical configurations:

1) In configuration C, the stability is considered to be the state that the PNNI routing protocol has completed the database synchronization and a PGL has been elected.
2) In configuration D, the stability is considered to be the state that the PNNI routing protocols have completed the database synchronization for the two peer groups, each has elected a PGL, and one routing hierarchy has been established.

From these two basic logical network configurations we studied the PNNI performance. First by using default values obtained from our small test bed measurements and then by changing various parameters to see what effect each of the parameters play. Our observations are presented next.

5.1 Default

Our base references are the two logical network configurations using the default values. The following results, presented in Table 1, describe the observations for these two configurations: C and D.

	Time (seconds)	**Data 1 (bytes)**	**Data 2 (bytes)**
Network Configuration C			
Database Synchronization	4.6013	37,630	34,450
PGL Election	50.458	52,218	44,626
Maintain PNNI Routing status	9000	1,076,377	122,218
Network Configuration D			
Database Synchronization	3.6011	19,610	16,218
Routing Hierarchy	94.456	46,958	29,362
Maintain PNNI Routing status	9000	1,736,121	86,337

Table 1. Observation Results for Default Values

The “Time” column represents the time to complete a certain function or simulated duration of the simulation run. “Data 1” represents the amount of data that was exchanged by the PNNI routing protocols. “Data 2” is the same as “Data 1” only without the Hello packets.

The time to complete database synchronization for configuration D is less than that for configuration C. This is due to the fact that there are fewer nodes in the peer group. Since the stability definition is different for configuration C (PGL Election) and configuration D (Routing Hierarchy), more time is needed by configuration D to complete its function. Looking at “Data 2” (i.e., without Hello packets), we observe that it takes less data to complete functions when the network is configured with a routing hierarchy (D) than without (C). When looking a “Data 1” (i.e., with Hello packets), there is an increase in the amount of data for configuration D because there are more logical links on which Hello packets must be sent.

5.2 Modify the Hello Interval

From the observation based on the two default configurations, we concluded that the Hello packets make the bulk of the PNNI data. The value of the Hello Interval determines the frequency of these Hello Packets. Therefore, by varying this parameter over the range of 10 ~ 20 seconds we observed the following.

1) Database Synchronization

The time and the amount of data required to complete the database synchronization remain unchanged from the default configuration.

2) PGL Election/Routing Hierarchy

The time and the amount of data required to complete the PGL election or routing hierarchy are shown in Figure 3.

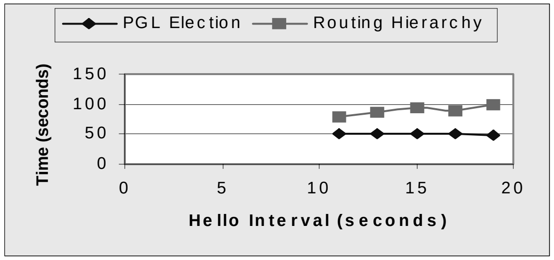

Figure 3a. Time

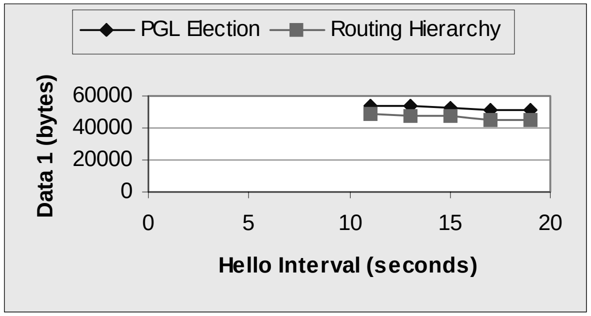

Figure 3b. Data 1

Increasing the Hello Interval increases the “Time” and decreases the “Data 1”. The “Data 2” (not shown) is the same as those observed by using the default values in configurations C and D. This is to be expected since “Data 2” does not contain Hello packets.

3) Maintain PNNI Routing status

The amount of data required to maintain the PNNI routing status is shown in Figure 4.

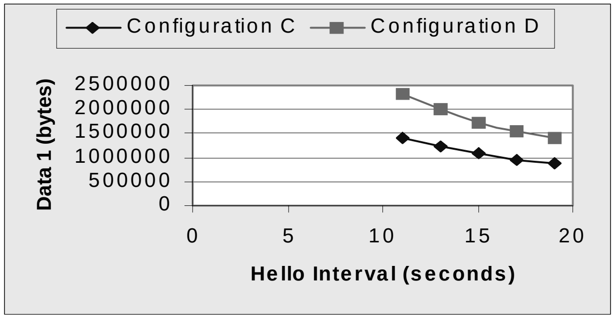

Figure 4. Maintain PNNI routing status

Increasing the Hello Interval decreases the “Data 1”, because the Hello Interval affects the frequency for sending Hello packets. The “Data 1” for configuration D is greater than configuration C, since more Hello packets are needed to establish the routing hierarchy. The “Data 2” remains unchanged from the default configurations.

As expected, increasing the Hello Interval decreases the amount of PNNI data transmitted and increases the time of completion for a network with routing hierarchy.

5.3 Modify the link delay

Since PNNI may be used over satellite links, we next investigated the effect of varying link delays on the PNNI routing protocols.

1) Database Synchronization

The time required to complete the database synchronization is shown in Figure 5.

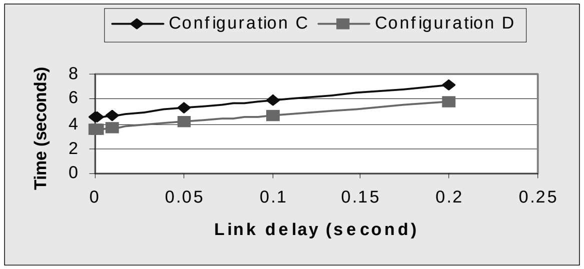

Figure 5. Database Synchronization

Only “Time” is shown in Figure 5, since the “Data 1”and “Data 2” remain unchanged from the results obtained by the default configurations. With the physical link delay increasing, the “Time” increases. The “Time” for a network with routing hierarchy is less than one without.

2) PGL Election/Routing Hierarchy

The time and the amount of data required to complete the PGL Election or Routing hierarchy are shown in Figure 6.

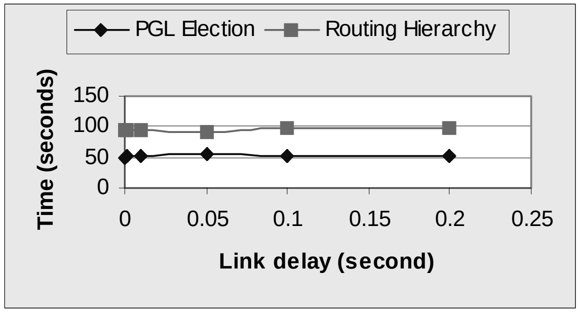

Figure 6a Time

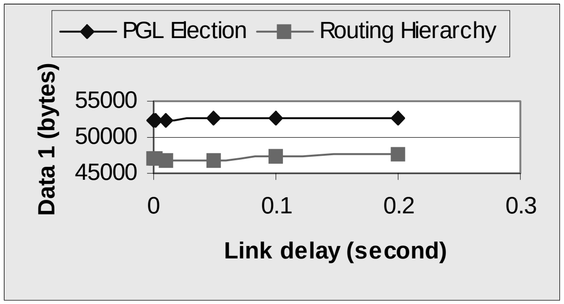

Figure 6b. Data 1

With the physical link delay increasing, the “Time” and “Data 1” both slightly increase. For the network with a routing hierarchy the “Time” increases and the “Data 1” decreases. The “Data 2” remains unchanged from the default configurations.

3) Maintain PNNI Routing status

The amount of data required to maintain the PNNI routing status is the same as those observed using the defaults.

5.4 Modify the processing time for various packet types

Since it is expected that the faster a system (i.e., node) can process the data it receives, the better it should perform. We investigate the results by varying the processing times for various packet types; Hello, DBS, PTSE request, and PTSP.

1) Database Synchronization

The time required to complete the database synchronization function based on various packet types are shown in Figure 7.

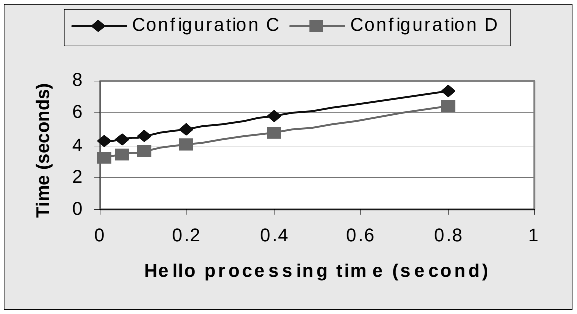

Figure 7a. Database Synchronization (Hello)

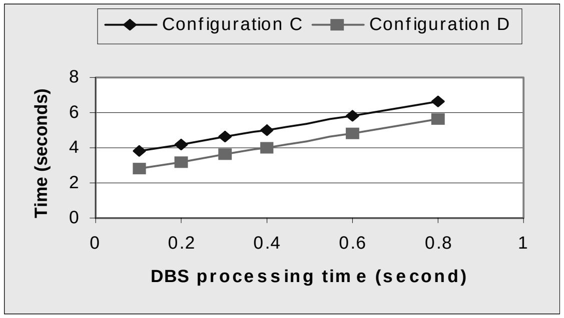

Figure 7b. Database Synchronization (DBS)

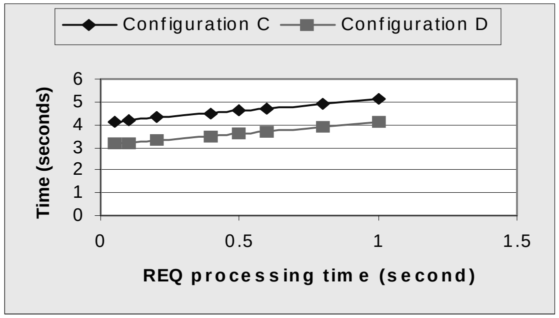

Figure 7c. Database Synchronization (REQ)

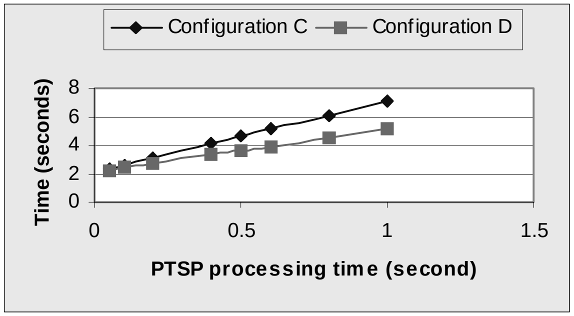

Figure 7d. Database Synchronization (PTSP)

As expected, as processing times increase, so does the time to complete the functions. For all packet types the "Data 1" and "Data 2" remain unchanged from the default.

2) PGL Election/Routing Hierarchy

The time and the amount of data required to complete the PGL Election or Routing hierarchy for the various packet types are shown in Figure 8.

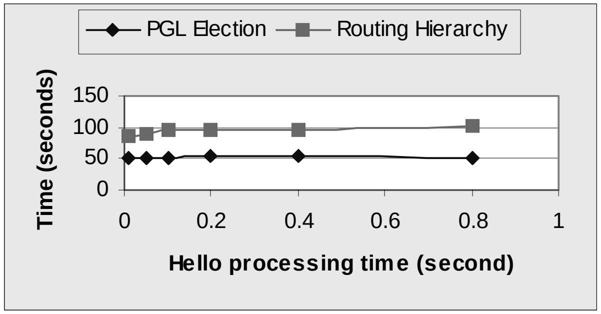

Figure 8a. Time (Hello)

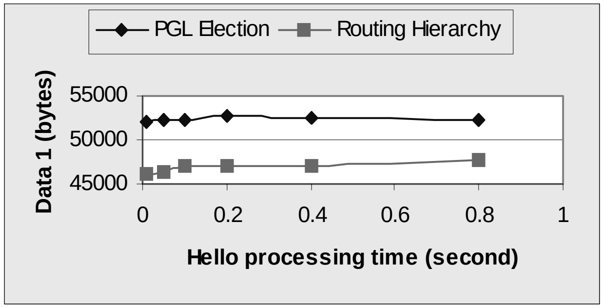

Figure 8b. Data 1 (Hello)

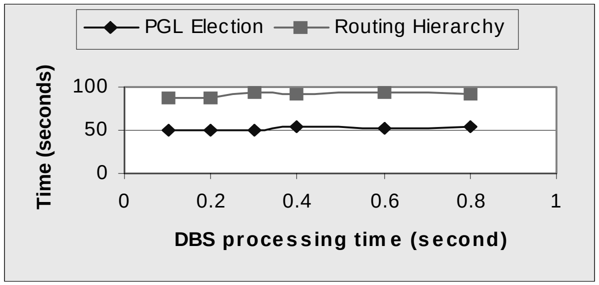

Figure 8c. Time (DBS)

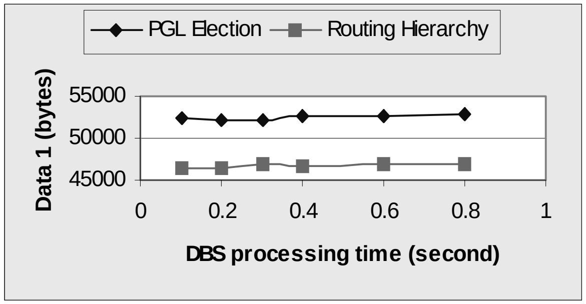

Figure 8d. Data 1 (DBS)

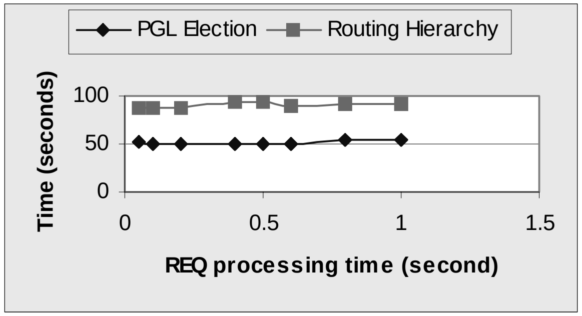

Figure 8e. Time (REQ)

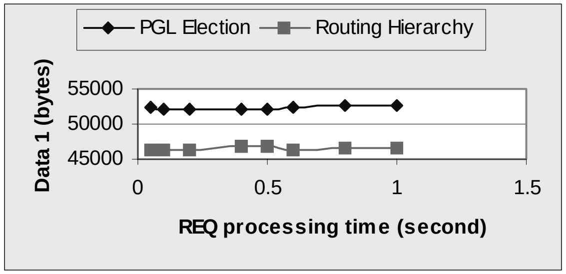

Figure 8f. Data 1 (REQ)

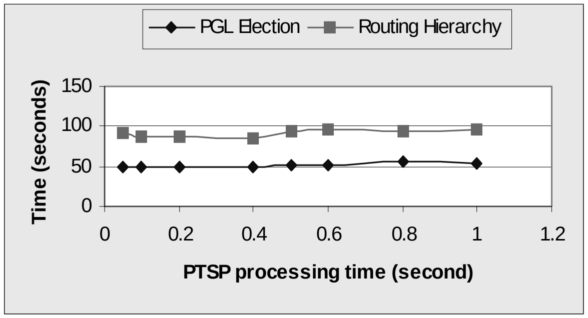

Figure 8g. Time (PTSP)

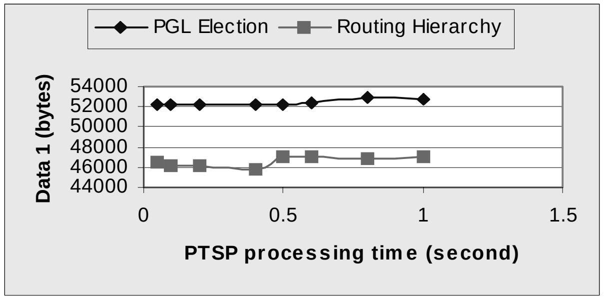

Figure 8h. Data 1 (PTSP)

Increasing the processing times for the various packet types, only show a very slight increase in the completing of the PGL Election or Routing hierarchy. The processing times have a random effect on the amount of data needing to complete the PGL Election and Routing hierarchy. This is due to the fact that there are different functions for each packet type.

3) Maintain PNNI Routing status

For all packet types the amount of data required to maintain the PNNI routing status remains unchanged from those observed using the default values in configurations C and D.

5.5 Scalability of PNNI routing

So far, we have simulated the performance of PNNI routing by modifying the node and link parameters and Hello Interval within one and two small peer groups, and observed the scalability of PNNI routing from a flat network to a hierarchical network. In order to study the scalability better, we use a larger network topology.

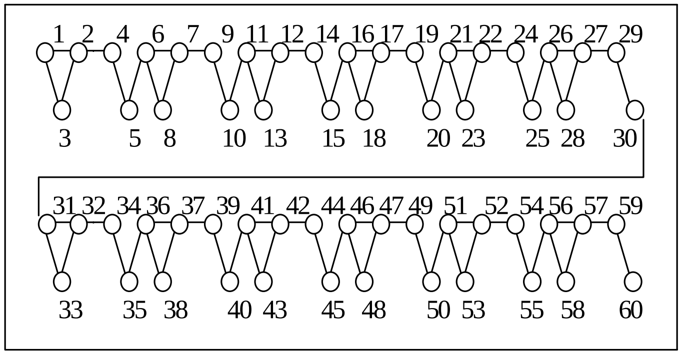

Figure 9. Topology for scalability study

Figure 9 shows a network with 60 nodes and their physical connections. All nodes are setup with a leadership priority and node ID in turn from 1 to 60. All other node parameters (processing times and PTSE refresh interval), link delay and Hello Interval use the default values. We consider the following logical configurations based on this fixed physical topology:

1) One peer group of 60 nodes. Stable state is indicated by completion of PGL Election.
2) Two peer groups of 30 nodes each (1-30 and 31-60). For this configuration and the remaining, the stable state is indicated by establishment of the routing hierarchy.
3) Three peer groups of 20 nodes each (1-20, 21-40, and 41-60).
4) Four peer groups of 15 nodes each (1-15, 16-30, 31-45, and 46-60).
5) Six peer groups of 10 nodes each (1-10, 11-20, 21-30, 31-40, 41-50, 51-60).
6) Twelve peer groups of five nodes each.

The amount of data required to complete the PGL election or routing hierarchy and to maintain the PNNI routing status are shown in Figure 10.

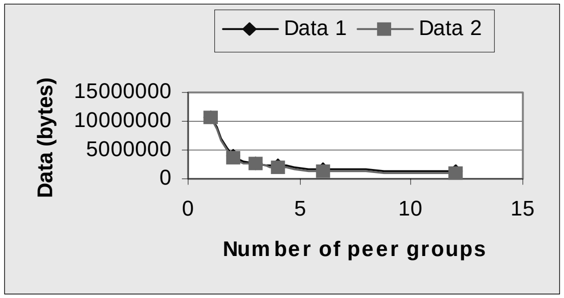

Figure 10a. PGL Election/Routing Hierarchy

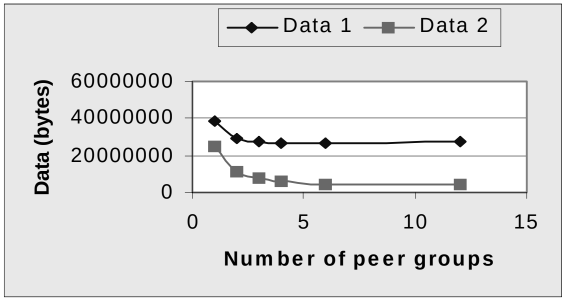

Figure 10b. Maintain PNNI routing status

The change in network configuration from no routing hierarchy to one with routing hierarchy shows a tremendous decrease in the amount of data. With the number of peer groups increasing, the number of nodes per peer group decreases and the amount of data decreases. In Figure 10b, the “Data 2” is much smaller than that of the “Data 1”, meaning that the majority of data transmitted consists of Hello packets.

Looking only at the configurations with a routing hierarchy (i.e., 2-6), Figure 11 shows the results based on the number of nodes within a single peer group, instead of the number of peer groups (i.e., Figure 10).

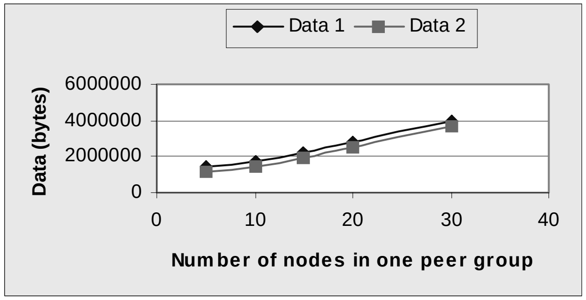

Figure 11a. PGL Election/Routing Hierarchy

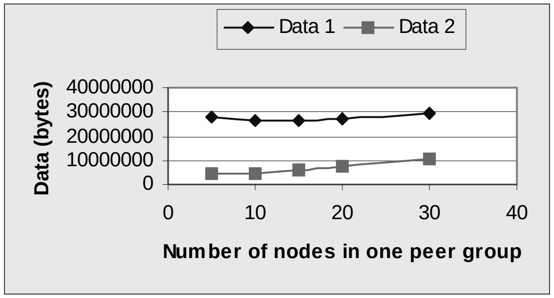

Figure 11b. Maintain PNNI routing status

It is obvious that as the number of nodes in one peer group increase, so does the amount of data needed to reach stability (PGL Election/Routing Hierarchy). However, for maintaining the PNNI routing status there is a trade-off. Looking at the Data 2, we see there is an increase, but looking at Data 1 there is a curve. This curve is due to the Hello packets exchanged on the new logical links connecting the peer groups.

5.6 Processing of Link failure or Node failure

The error time is defined as the time period when the failure is flooded to all nodes. The restore time is defined as the time period over which the failure is restored and the database is completely synchronized. We use the network topology for configuration C, Figure 2a.

We examine two cases: 1) a link failure takes place between node 4 and node 5, which is the same as when node 5 has a physical node failure and 2) node 5 has a logical node failure. We simulate this configuration by modifying the Hello Interval, Link delay, Hello processing time, DBS processing time, REQ processing time and PTSP processing time in turn, and obtain the results in Figure 12.

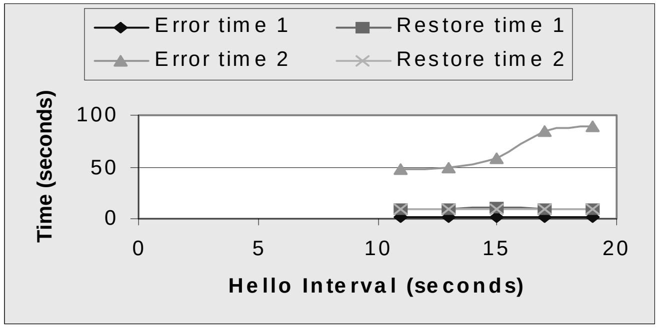

Figure 12a. Modify Hello Interval

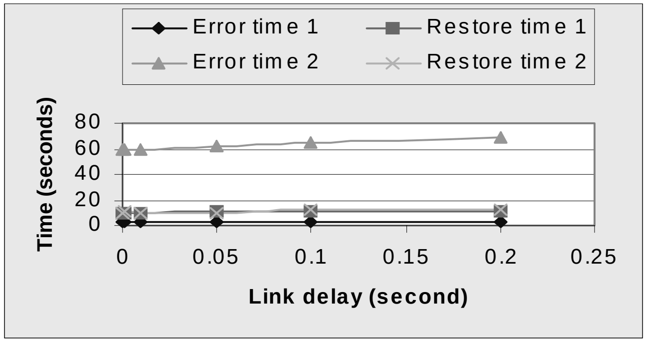

Figure 12b. Modify Link Delay

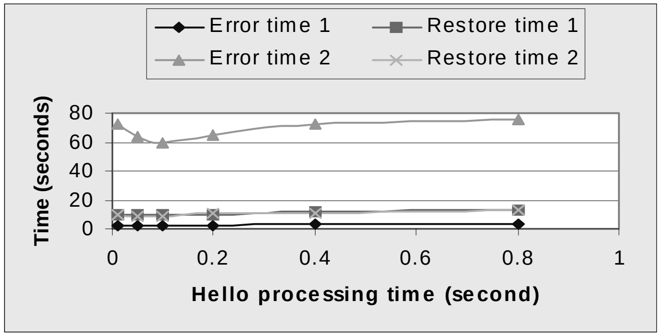

Figure 12c. Modify Hello processing time

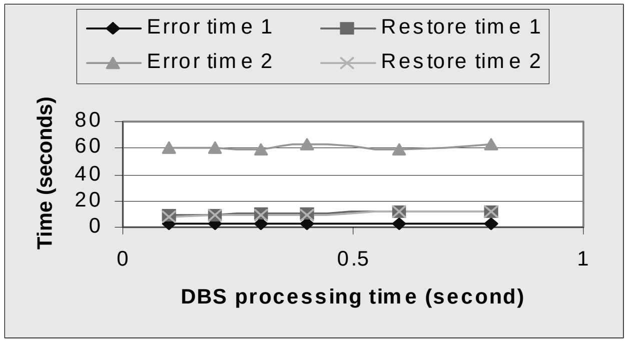

Figure 12d. Modify DBS processing time

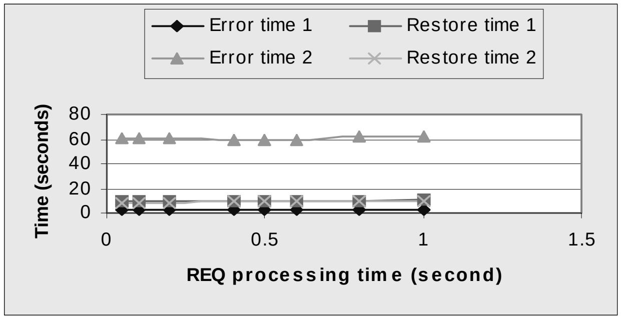

Figure 12e. Modify REQ processing time

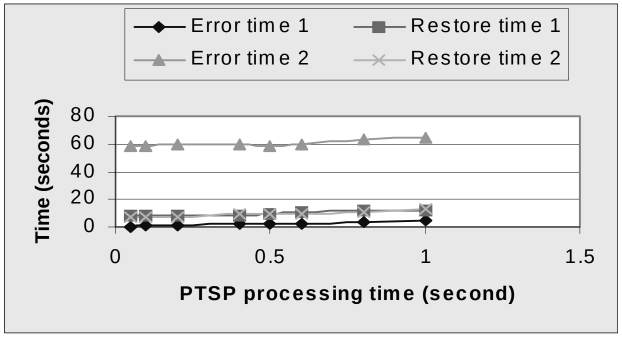

Figure 12f. Modify PTSP processing time

In Figure 12, "Error time 1"and "Restore time 1" are the error time and restore time for link failure and physical node failure. Since the link failure and physical node failure for node 5 have the same influence on all other nodes, the result of either error is the same. "Error time 2" and "Restore time 2" are the error time and restore time for a logical node failure.

We get the following observations:

- The "Restore time 1" is almost the same as the "Restore time 2", meaning that the restore time is independent of the failure type.
- The "Restore time 1" is bigger than the "Error time 1", since the database synchronization function must occur again and not just flooding when an error is discovered.
- The "Error time 2" is much greater than the "Error time 1", meaning that it takes longer for the PNNI routing protocols to discover a logical node failure than for a physical or link failure.
- For link delay, Hello processing time and PTSP processing time modification, the "Error time 1" increases as the relevant parameter increases. For other cases, the "Error time 1" remains unchanged.
- For Hello Interval and Link delay modification, the "Error time 2" increases as the relevant parameter increases. For other cases, the "Error time 2" does not simply increase or decrease its value with the relevant parameters' modification, because the "Inactivity timer" is expired after "Hello timer" of the same link is expired 3, 4, or 5 times. Therefore, the value ranges from 45 to 80 seconds.
- Except for the "Restore time 1" for Hello interval modification, the "Restore time 1 & 2" slightly increases as the relevant parameter increases.

5.7 Real & Simulated Results

Finally we used the same network configurations as the small test bed with the measured values as the input configuration and parameters to the simulator. This was done to verify the ability of the simulator to simulate real network behavior when provided with real network values.

Because all these switches are real implementations with different booting times, it is difficult for us to make them active at the same time. We assume that all these switches are already active, then we connect the physical link between these nodes in turn. Beginning from the time of the initial connection of the switches and ending two hours later, we monitored these switches. This two hour experiment period is enough time for us to observe the necessary functions that ensure the maintainability of the PNNI routing status. To simulate this, we define the physical link connection times to be 0 seconds between node 1 and node 2, 30 seconds between node 3 and node 4, and 60 seconds between node 2 and node 3. The simulation duration time is two hours.

The results are summarized in Tables 2 & 3.

	Test bed	Simulation 1	Simulation 2
Stable time (seconds)	70	75.3	75.3
Total data for PGL Election (bytes)	18550	18709	18709
Total data for PGL Election without Hello packets (bytes)	14734	14416	14416
Total data for maintainability (bytes)	523375	515743	530053
Total data for maintainability without Hello packets (bytes)	74677	57346	71656

Table 2. The comparison results for configuration A

	Test bed	Simulation 1	Simulation 2
Stable time (seconds)	120	120	120
Total data for Routing Hierarchy (bytes)	18126	17649	17649
Total data for Routing Hierarchy without Hello packets (bytes)	7791	8268	8268
Total data for maintainability (bytes)	903491	738714	745286
Total data for maintainability without Hello packets (bytes)	38743	27984	34556

Table 3. The comparison results for configuration B

Simulation 1 and Simulation 2 use different PTSE refresh intervals 1800 seconds and 1500 seconds, respectively. As you can see the simulation results are close to the measured results of the test bed.

6. Conclusions

Simulation plays an important role in the performance study of PNNI ATM networks, as do test beds. APRoPS is one such simulator. Using this simulator and our test bed, we conclude the following:

- The trade-off for the Hello Interval effects the amount of data sent to maintain link information and responsiveness to discover failures. An increase in the Hello Interval decreases the amount of data, but increases the time to discover a failure.
- Processing times within the nodes for packets, and link delays affect network stability times, but do not affect the amount of data needed to reach stability.
- Using a hierarchical network configuration reduces the amount of data and time required to reach network stability and maintainability.

The PNNI ATM networks can be a scalable, robust, and maintainable, provided an appropriate hierarchy is defined with the corresponding PNNI routing protocol parameters.

References

[1] Sivabalan, M. and Mouftah, H. T., "QUARTS-II: A Routing Simulator for ATM Networks," IEEE Communications Magazine, May 1998, pp.80-87.

[2] Song, Y., "The NIST ATM PNNI Routing Protocol Simulator (APRoPS), Operations and Programming Guide, Version 1.0, December 1998"

[3] Tunpan, A., "Performance Study of the ATM PNNI Routing Protocol," Dept. of Computer Science, University of Maryland, August 8, 1997.

[4] Private Network-Network Interface Specification Version 1.0 (PNNI 1.0), af-pnni-0055.000, March 1996.

[5] PNNI v1.0 Errata and PICS, af-pnni-0081.000, May 1998.