Technical Report

Performance Evaluation of PNNI using the KUPNNI Simulator

Pradeepkumar Mani and David Petr

October 2002

ITTC-FY2003-TR-22735-02

Project Sponsor:
Sprint

Chapter 1

Introduction

This technical report is a compilation of the results of various simulations conducted using the **KU-PNNI** simulator during the academic year 2001 – 2002. Analyses of the results are primarily based on evidence collected during the course of the simulation, in the absence of which the analysis is based on a plausible hypothesis drawn after extensive discussion. Pictorial representations of the test topologies and plots of results are provided wherever deemed necessary. For more information on the KU-PNNI simulator, refer to the KU-PNNI user's manual **[1].**

Chapter 2 provides a brief explanation of the meaning of each performance metric that is monitored during the simulations. Chapter 3 deals with the results and analysis of simulations conducted by scaling the topology in size. Chapter 4 deals with the analysis of the results of simulations conducted by varying load, while chapter 5 deals with analysis of the effect of peergrouping. Chapter 6 deals with the effects of changing the number of hierarchical levels, while chapter 7 deals with the effects of varying the crankback counts. Chapter 8 deals with the simulator performance (time taken to complete the simulation) measured as a function of load and size of the topology.

All tests were conducted on a Linux box running RedHat Linux 6.1, with 2GB RAM and an Intel P-III 533 MHz processor.

Chapter 2

Terminologies

2.1 Performance metrics

The performance of the network under consideration is assessed through various metrics as deemed appropriate. The following are all the performance metrics that are monitored during the entire set of simulations, and a brief explanation to each one of them:

Mean Call setup time

A non-zero time is required to setup a call between the caller and callee. The time elapsed from the moment the caller placed a request for the setup of a virtual circuit between the caller and the callee, to the moment the caller receives confirmation from the callee that a path was successfully setup, is called *Call setup time.* The average of call setup times of all the calls is called the "mean call setup time".

Call Success percentage

A call is considered successfully setup if and only if a virtual path is successfully established between them, meeting all QoS requirements of the call. Hence, the call success percentage is the ratio of successful calls of all the nodes to the sum total of calls placed by all the nodes.

Convergence time

Convergence time is defined as the time taken for all the nodes in the network to exchange topological information, and eventually share the same view of the network. *Convergence time – low* refers to the least value of convergence time among all nodes, while *Convergence time – high* refers to the maximum value of convergence time among all the nodes

Average Database size

Each node maintains a database of topological information of all the nodes in the network. The average database size is the average of database sizes of all the nodes.

Average number of hops

The number of hops for a call can be defined as the number of links the call has to traverse before it reaches the destination. Hence the average number of hops is the ratio of the total number of hops over all successful calls, to the total number of successful calls.

Average Utilization percentage

Utilization percentage is the ratio of consumed bandwidth at a link to the total available bandwidth in the link at any instant, expressed as a percentage. Average utilization percentage is the ratio of the sum of all samples of utilization percentage collected over a period of time to the total number of such samples.

Total floods

The nodes in the network exchange PNNI control information through the mechanism of reliable flooding. The total floods metric is the count of the total number of floods initiated during the entire duration of the simulation.

PNNI data / Flood

PNNI data per flood is the total PNNI data exchanged via flooding (in bits) normalized over the total number of floods, or PNNI data normalized over the number of nodes and floods (per node). This is done to remove dependency of PNNI data over the number of nodes or the length of simulation (or number of floods). *PNNI data – high* refers to the maximum value of PNNI data flooded by any node, among all the nodes, while *PNNI data –low* refers to the least value of PNNI data flooded by any node, among all the nodes.

Percentage Wasted Floods

A flood is considered “wasted” if it conveys no new information to the node concerned. Hence percentage total wasted floods is the total number of wasted floods, normalized over the total number of floods and expressed as a percentage.

Chapter 3

Scalability test

3.1 Introduction

The aim of this test is to study the effect of scaling the topology (in size) on the various identified metrics. This is an important test for any topology, because it gives a rough idea on how much the topology can be expanded within acceptable limits of values of the performance metrics. For all topologies, calls were generated with exponential ***mean call inter-arrival times*** of *20 sec, 15 sec, 10 sec, 5sec, and 2 sec*.

3.2 Test topology

3.2.1 Edge-core with 20 nodes

This topology (see fig 3.1) has 16 edge nodes and 4 core nodes. The edge nodes are grouped in 4 peergroups of 4 nodes each, and core nodes are grouped in one peergroup. Each node has one host connected to it with 120 calls generated per host.

The hosts are connected to the nodes via 10Mbps links; the border edge node in each peergroup connects to the three other edge nodes in the peergroup via DS3 (approximately 45Mbps) links, while the border edge nodes connect to the core nodes via OC3 (approximately 155.6 Mbps) links. Finally, the core nodes interconnect among themselves via OC12 (approximately 622 Mbps) links. One point to be noted is that the core nodes do not have any hosts attached to them.

A two-level hierarchical structure is followed while designing the topology. The border edge nodes are numbered as ‘1’ in each peergroup; the core nodes are contained in the ‘E’ peergroup.

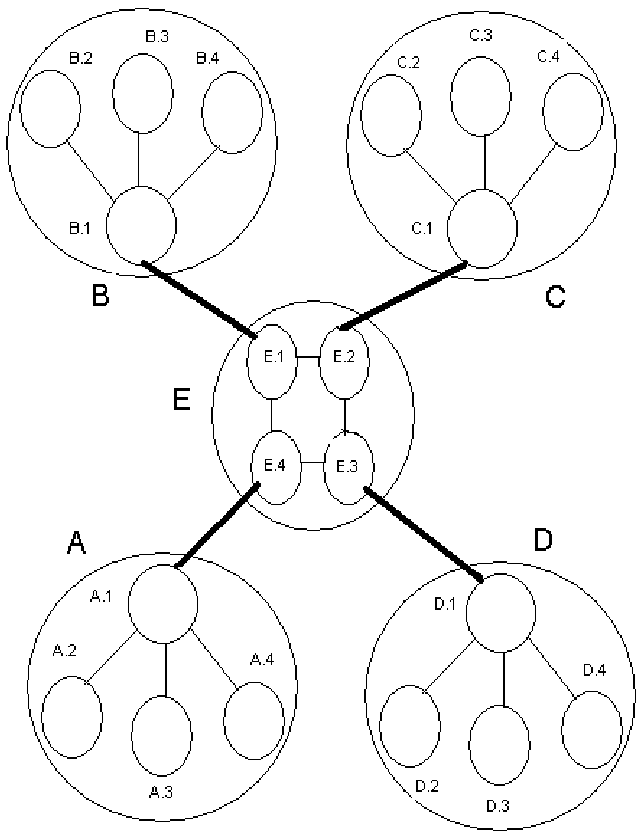

Fig 3.1 Edge-Core 20 nodes

3.2.2 Edge-core with 40 nodes

This topology (fig 3.2) is composed of 32 edge nodes and 8 core nodes. The edge nodes are grouped in 8 peergroups of 4 nodes each, and core nodes are grouped in one peer group ‘Core’. Each node has one host connected to it and the number of calls generated is 60 calls per host.

The links used to interconnect the nodes to nodes, and nodes to hosts follow the same pattern as in the edge-core with 20 nodes topology. The core nodes, instead of simply being in connected in a ring-like fashion, have a couple of other inter-connections, allowing any core node to reach any other code node in one or two hops.

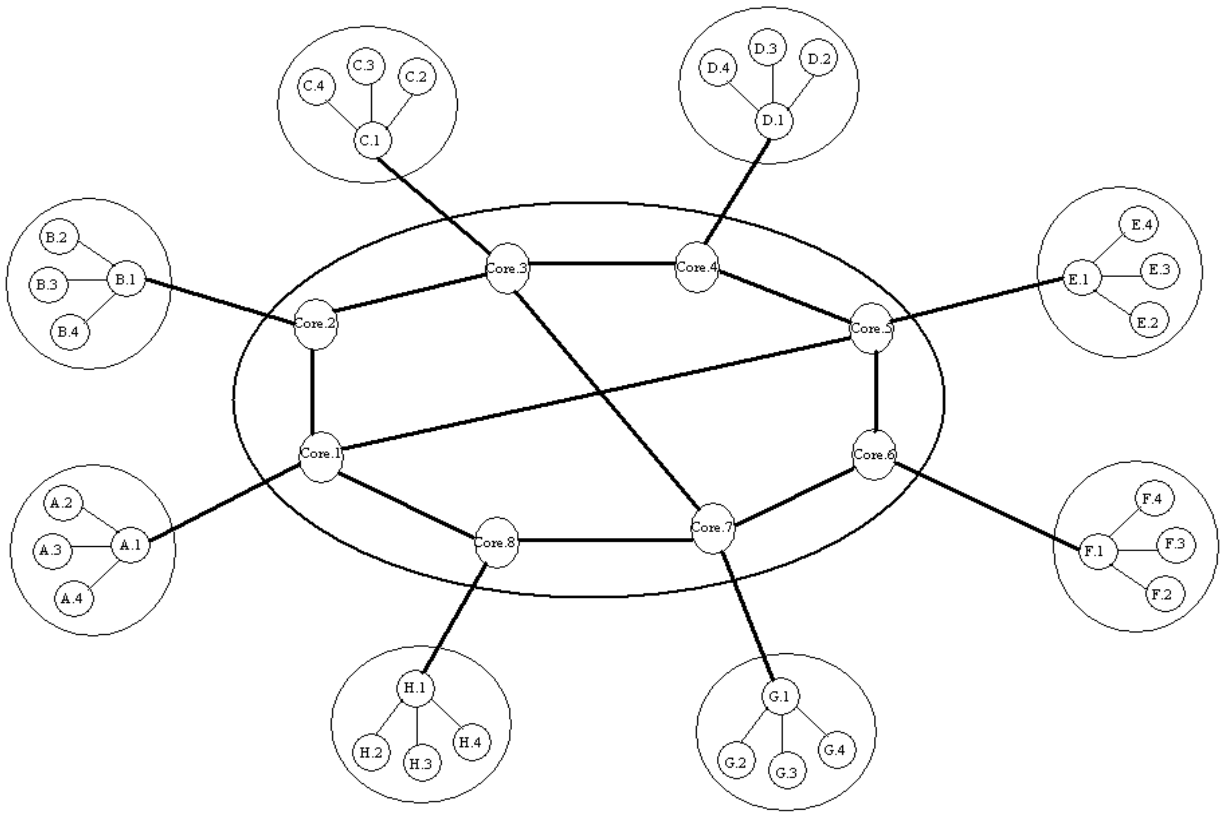

Fig 3.2 Edge-Core 40 nodes

3.2.2 Edge-core with 80 nodes

This topology (fig 3.3) is composed of 64 edge nodes and 16 core nodes with the edge nodes grouped in 16 peergroups of 4 nodes each, and core nodes in one peergroup. Each node has *one host* connected to it and the number of calls generated is *30 calls per host.* The interconnecting links follow the same pattern as in the Edge-core 40 node topology.

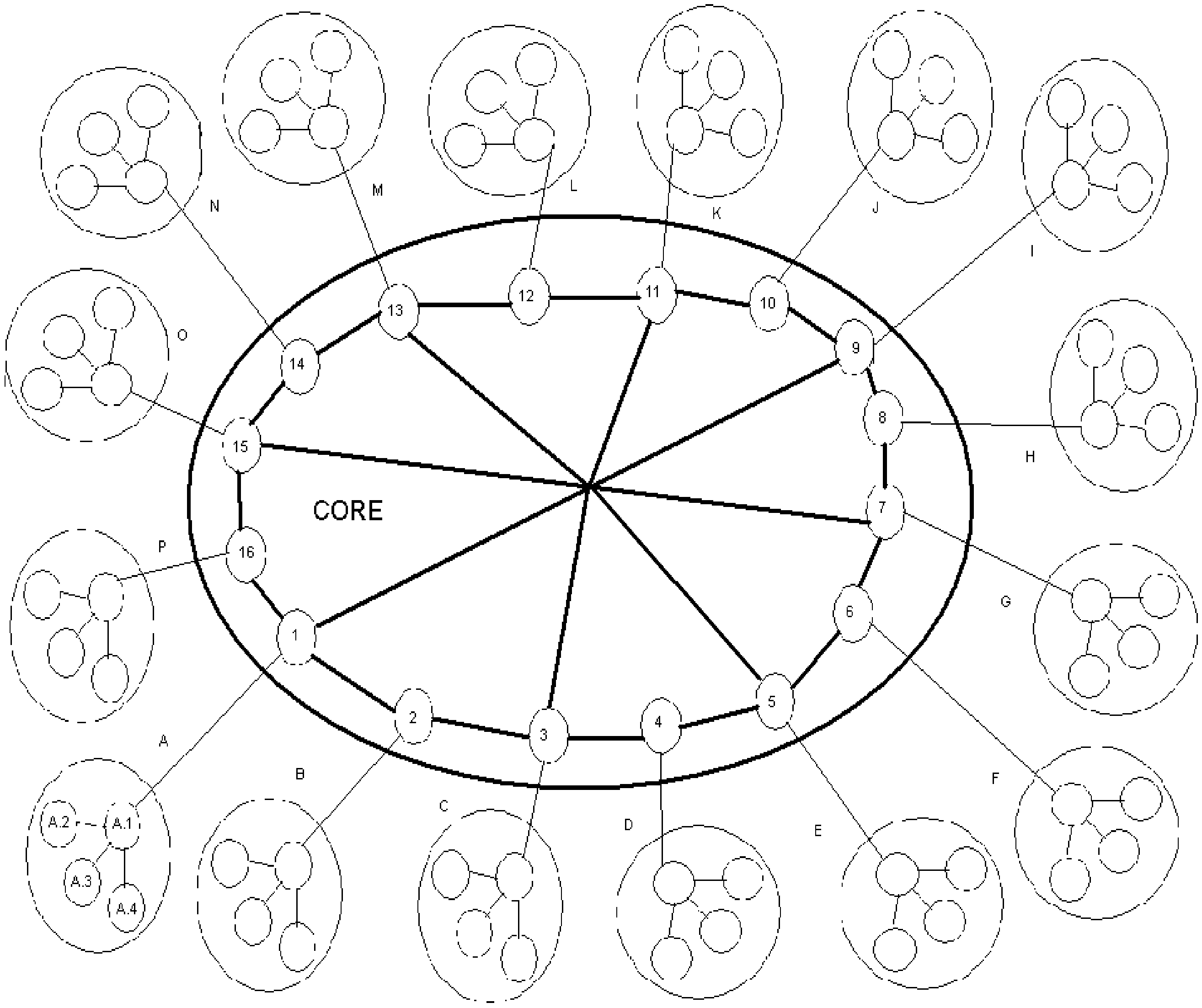

Fig 3.3 Edge-Core 80 nodes

3.3 Performance Metrics

The metrics of *primary* interest in this set of simulations are:

- Mean Call setup time
- Call success percentage
- Average Database size
- Convergence time

Along with these metrics, the other metrics such as Average number of hops, Average utilization percentage and PNNI data / flood are also monitored.

3.4 Results and Analysis

3.4.1 Average Database size

Figure 3.4 gives the variation of average database size with increase in the number of nodes in the topology. The average Database size increases as the number of nodes is increased in the topology, as expected.

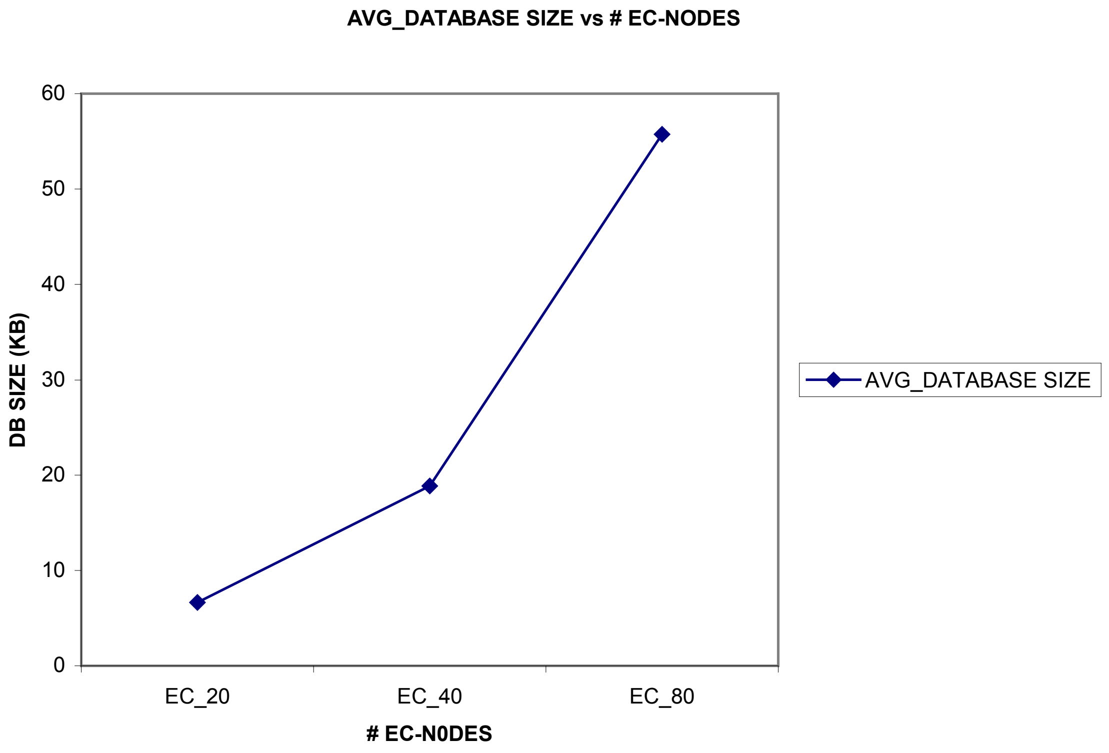

Fig 3.4 Average database size vs. Topology size

This is because each node has to record more connectivity information to other peergroups (more the number of nodes, more is the number of peergroups).

3.4.2 Call success percentage

Figure 3.5 gives the variation of call setup success rate as a function of the number of nodes in the edge-core topology. For any given load, the call success percentage tends to decrease with an increase in the size of the topology. The test topologies have been designed such that a call never fails due to lack of bandwidth; it fails only due to time outs (8 seconds) that occur when the link is down or when the peer node is very busy processing other calls.

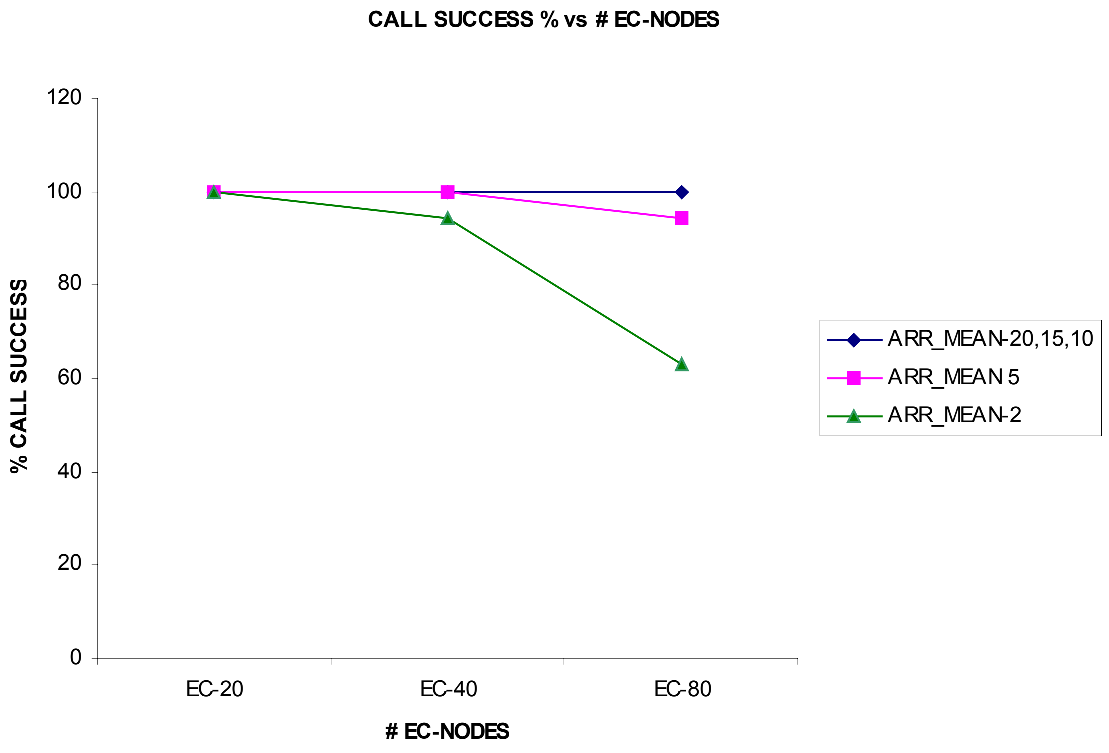

Fig 3.5 Call success rate vs. Topology size

The call success rate is fairly constant (nearly 100 percent) at light load for all topologies, but decreases as the load increases. The effect is more pronounced in a bigger topology. This is because the nodes receive calls at a higher rate than they can process the calls. For example, the core nodes serve as gateways for edge nodes, and hence for a larger topology the rate at which the core nodes receive calls is much higher than the rate at which core nodes in a smaller topology receives calls. Also, flood messages also increase call delay. This is because the flood messages are also queued and they consume processing resources. If the queue size were sufficiently large so that the waiting time of any call setup request exceeds 8 seconds, then the caller would be timed out because timer T303 would expire twice and the call would be cleared.

3.4.3 Mean Call setup time

Figure 3.6 gives the variation of mean call setup time as a function of the topology size. For any given load, the mean call setup time increases with an increase in the size of the topology. The term *light load* refers to call inter-arrival times of 20 secs, 15 secs and 10 secs, while *heavy load* refers to call inter-arrival times of 5 secs and 2 secs.

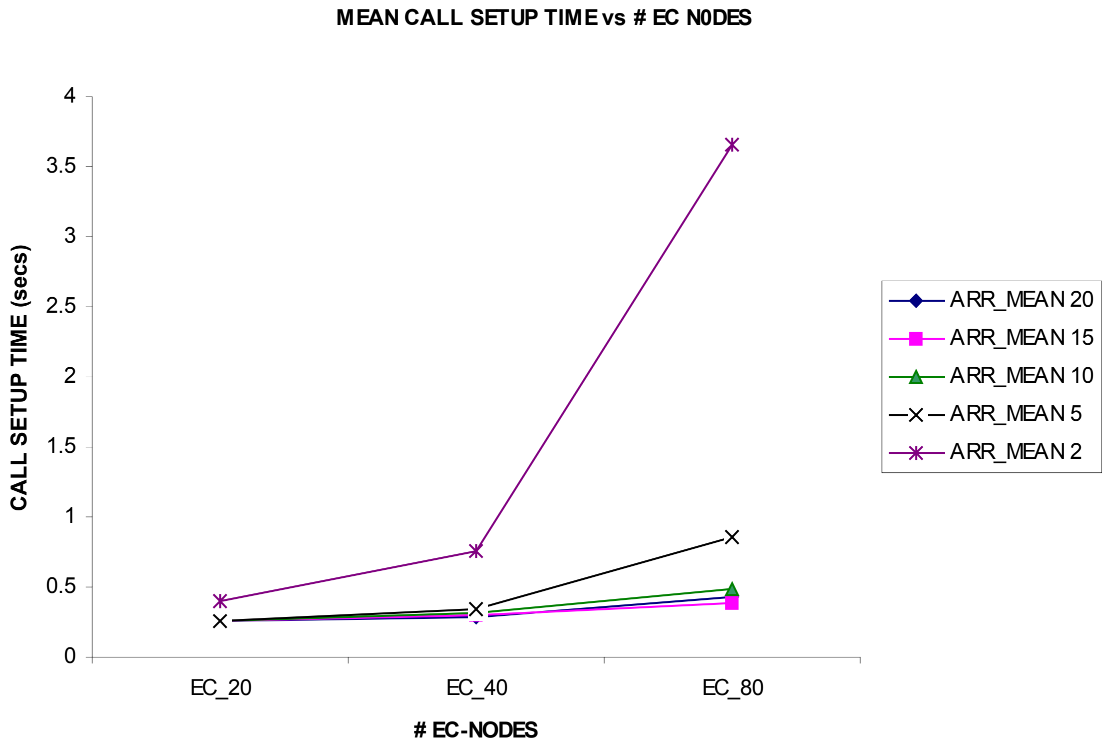

Fig 3.6 Mean Call setup time vs. Topology size

Under light load, the call setup time is mainly composed of the time taken to find a route; hence at light load, the call setup time is fairly a constant for all topologies. Under heavy load, the time a call has to wait in the queue before it gets processed becomes dominant. The time a call has to wait for service is much higher with a larger topology, because of the higher arrival rate to the core nodes (as seen above).

3.4.4 Convergence time

Figure 3.7 gives the variation of convergence time as a function of the number of nodes in the topology. Convergence time increases as the topology is increased in scale. This is because the number of nodes that has to converge is higher in a bigger topology.

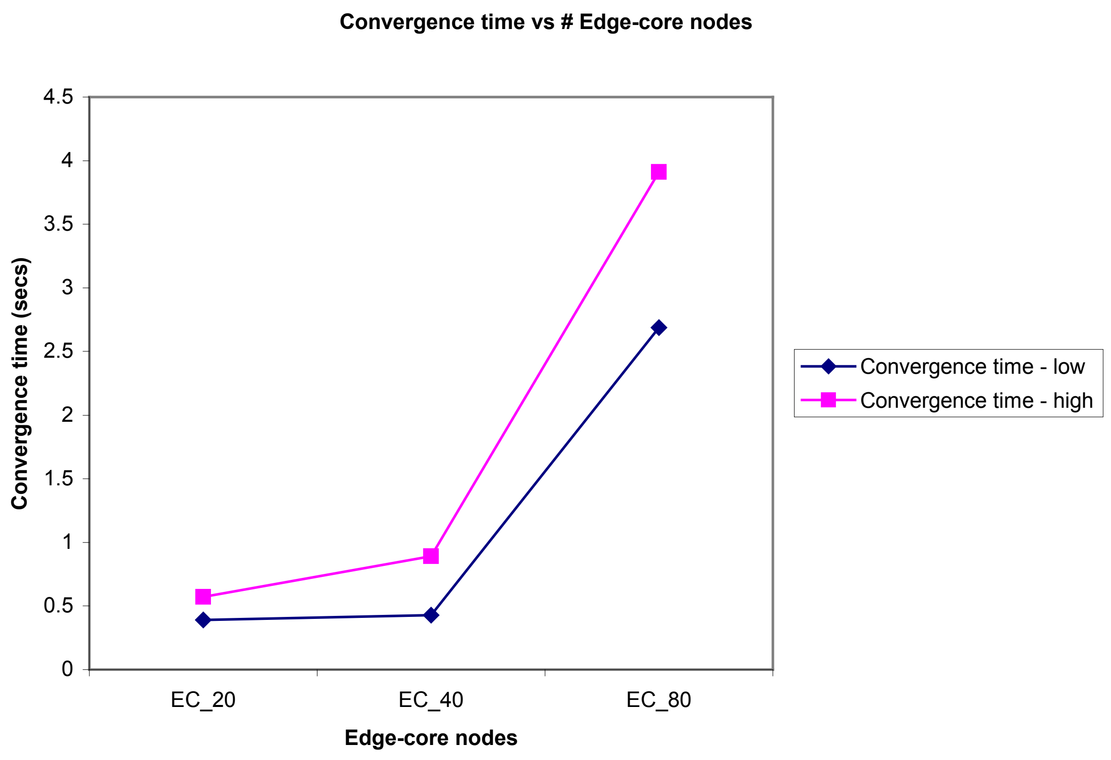

Fig 3.7 Convergence time vs. Topology size

3.4.5 PNNI data

Figure 3.8gives the variation of PNNI data / flood as a variation of topology size As the number of nodes increases, so does the average database size; hence the average PNNI data/flood also increases.

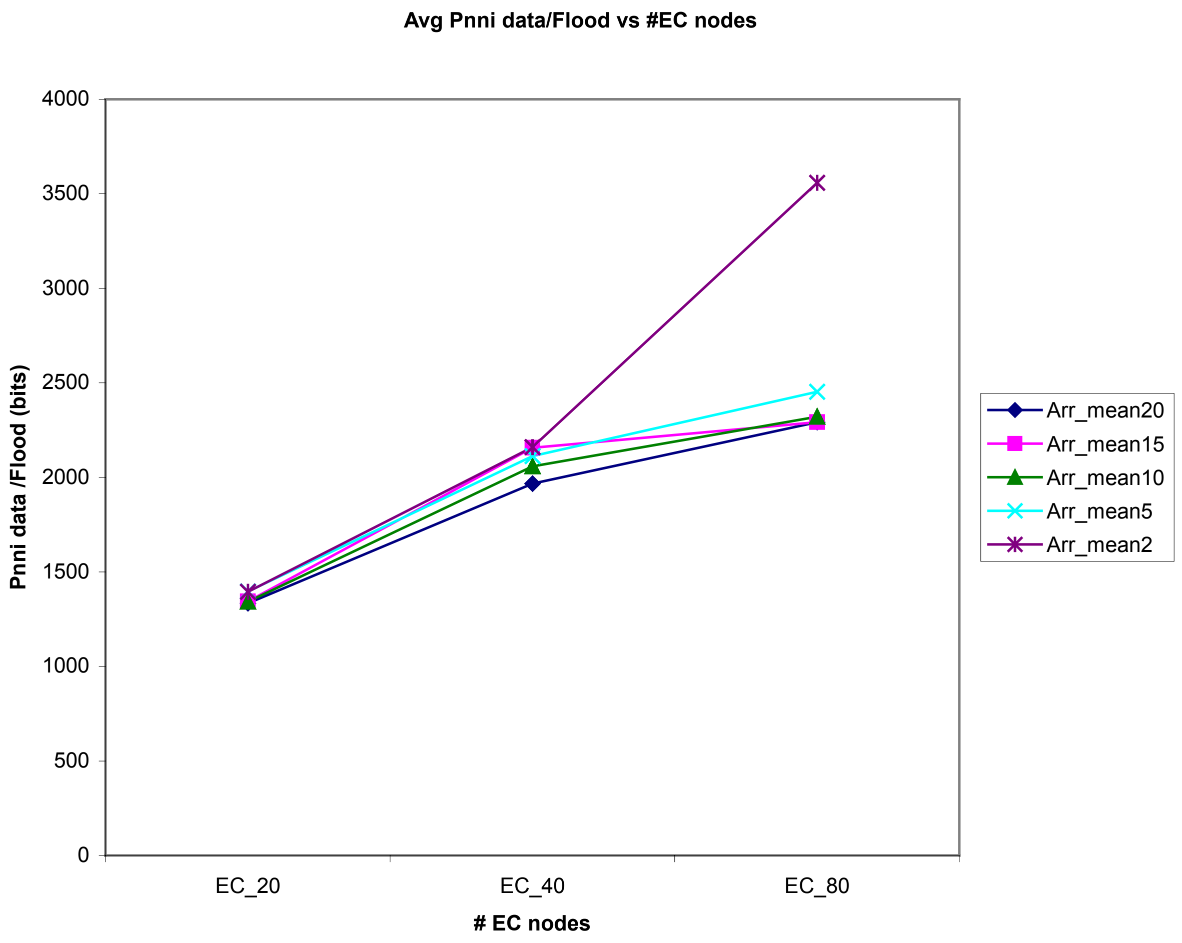

Fig 3.8 PNNI data / flood vs. Topology size

As load increases, *significant changes* occur at a faster rate, which results in the *PNNI Topology State Packets (PTSPs)* carrying more *PNNI Topology State Elements (PTSEs)*. Hence, the average value of PNNI data /flood increases with an increase in load.

3.4.6 Average Bandwidth Utilization percentage

Figure 3.9 shows average bandwidth utilization as function of both load and topology. For any given load, for similarly scaled topologies, the average utilization would be almost equal. The pattern as exhibited by the graph is due to the fact that the topologies exhibit significant differences.

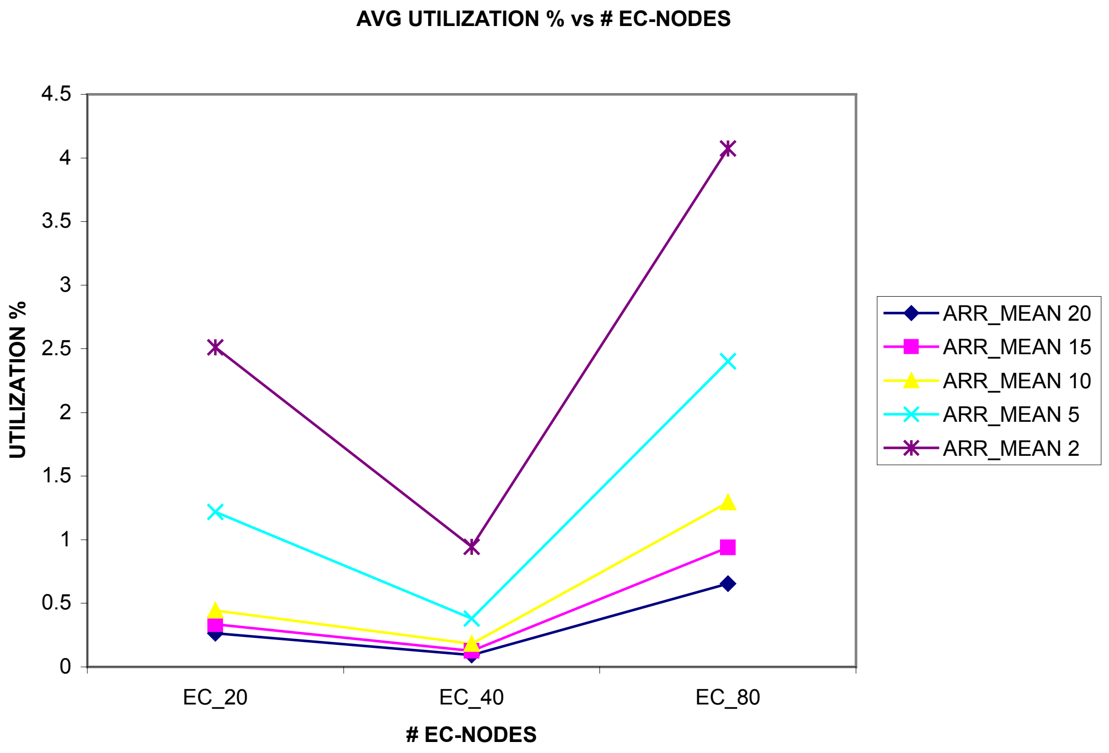

Fig 3.9 Average Utilization percent vs. Topology size

For example, EC-40 has a semi-full mesh core, all inter-connected through OC-12 links, which tends to decrease utilization percentage considerably. On the contrary, the EC-80 has a relatively less-densely interconnected core and hence the average utilization for EC-80 is higher when compared to that of EC-40.

3.4.7 Average number of hops

Figure 3.10 gives the variation of average number of calls with respect to topology size. In exactly similarly scaled topologies, the average number of hops per call would be the same. But, for practical purposes, exactly similar scaling of topologies is very difficult. Hence, the average number of hops can be expected to vary with the *maximum number of* *hops* that any successful call has to take. In the test topologies, the possible values for hops are 2, 5 and 6 (for 20 nodes and 40 nodes), and 2, 5, 6 and 8 (for 80 nodes).

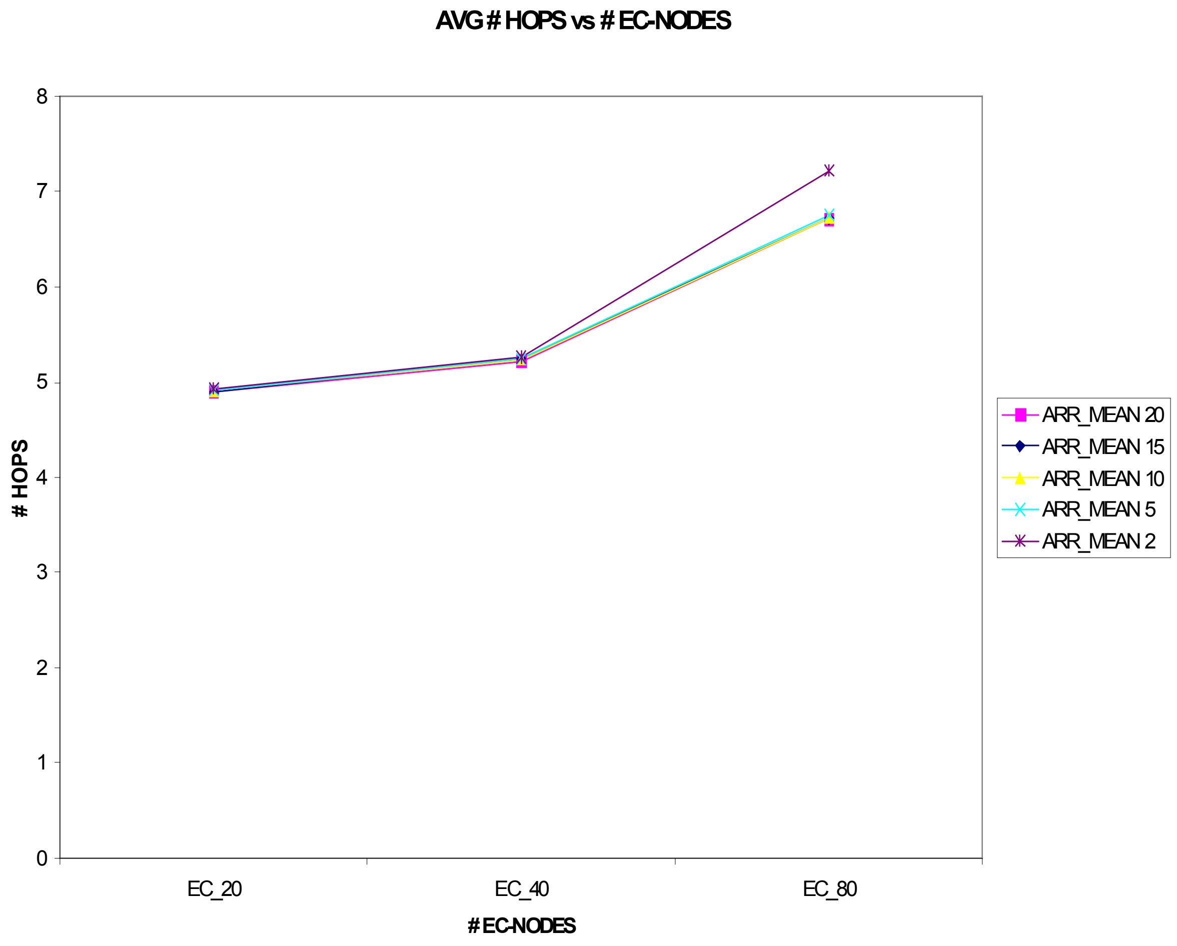

Fig 3.10 Average hops vs. Topology size

The variation in the number of hops for various loads is purely due to randomness in choosing the destination to setup a call.

3.5 Problems faced

The following were the problems faced during the course of conducting the simulations listed above:

1. The topology could not be scaled beyond 80 nodes because the simulator core dumped even at light load beyond a topology size of 80 nodes. The simulations could have been run to completion at bigger topologies, but with a lesser number of calls/host. This would have given only a lesser (and insufficient) number of samples of the performance metrics for analysis to be conducted, the results of which would be biased due to the effect of variance in the sampled values. Hence, as a compromise, it was decided that the simulations would be based on roughly 2000 calls (64 hosts x 30 calls/host, 32 hosts x 60 calls/host, 16 hosts x 120 calls/host). Moreover, 30 calls/host was the maximum the simulator could be strained at high loads with 80 nodes.

3.6 Conclusion

It can be thus concluded that as the topology is scaled in size for bandwidth-rich networks,

- Mean Call setup time increases
- Call success percentage decreases
- Average Database size increases
- Convergence time increases
- Average PNNI data / flood increases
- Average number of hops increases (in general, but depends on topology)

Chapter 4

Load test

4.1 Introduction

The aim of this test is to study the effect of varying the load (call arrival rate) on the various identified metrics. This is an important test for any topology, because it gives a rough idea on how much the network can be loaded before the values of performance metrics begin to degrade to unacceptable levels. The simulation results in this chapter are identical to those in chapter 3, but the results are presented to highlight the effects of load variations rather than network size.

4.2 Test topology

The simulations were run for mean call inter-arrival times of 20 seconds, 15 seconds, 10 seconds, 5 seconds and 2 seconds over Edge core topologies with:

- 20 nodes (same as in scalability test, fig 3.1)
- 40 nodes (same as in scalability test, fig 3.2)
- 80 nodes (same as in scalability test, fig 3.3)

4.3 Performance Metrics

The metrics of *primary* interest in this set of simulations are:

- Mean Call setup time
- Call success percentage
- Average Utilization percentage
- PNNI data / flood
- Average number of hops

Other metrics (like average data base size and convergence time) are load independent, and hence are not considered here

4.4 Results and Analysis

4.4.1 Mean Call setup time

Figure 4.1 gives the variation of mean call setup time with respect to variation in load.

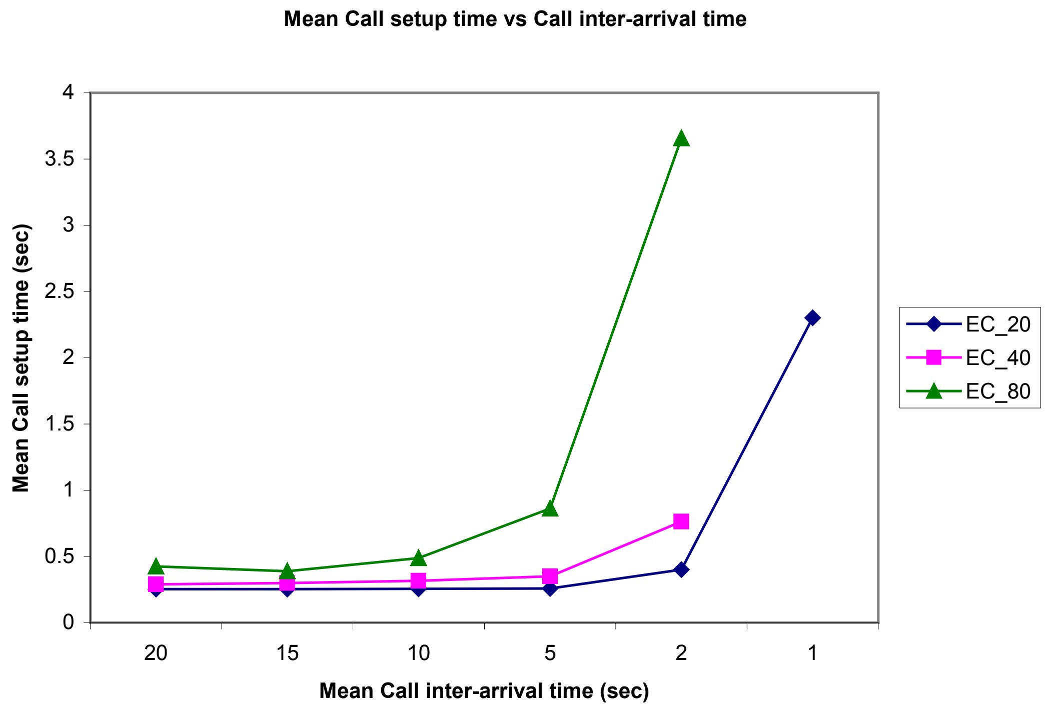

Fig 4.1 Mean Call setup time vs. Load

The mean call setup time increases with increase in load because the call arrival rate to each node keeps increasing, and hence the time each call has to wait (in the queue) for service also increases. This effect is more pronounced in topologies with a larger number of nodes. The mean call setup time is purely a function of call processing rate of the nodes.

4.4.2 Call success percentage

Figure 4.2 gives the variation of call success percentage as a function of load. The call success percentage decreases as load increases. This is because the rate of call arrival to any node is so high, that some calls have to wait for a very long time before they get timed out and the call fails.

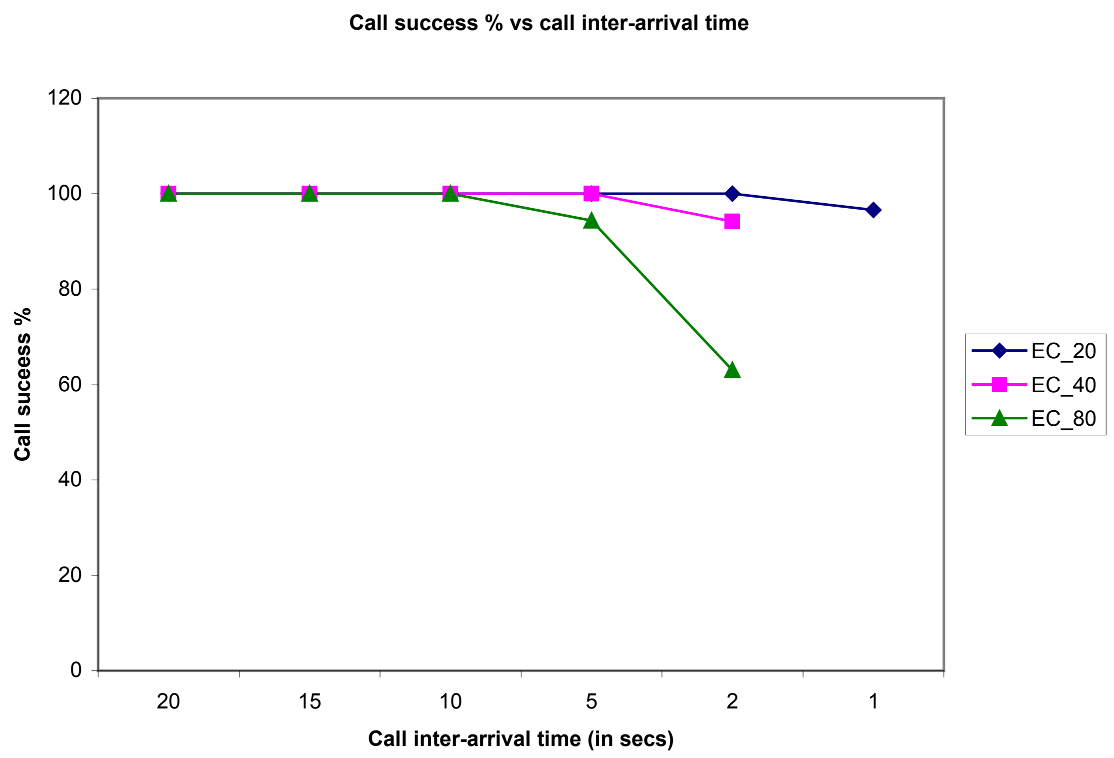

Fig 4.2 Call success rate vs. Load

The call success rate is again purely a function of the call-processing rate of the nodes; the link bandwidth is always sufficient to handle all the calls generated, and hence lack of link bandwidth never occurs to cause a call failure.

4.4.3 Average bandwidth utilization percentage

Figure 4.3 shows the variation of average bandwidth utilization percentage with change in load. The average utilization increases as the load increases, since more calls are active and consuming link bandwidth.

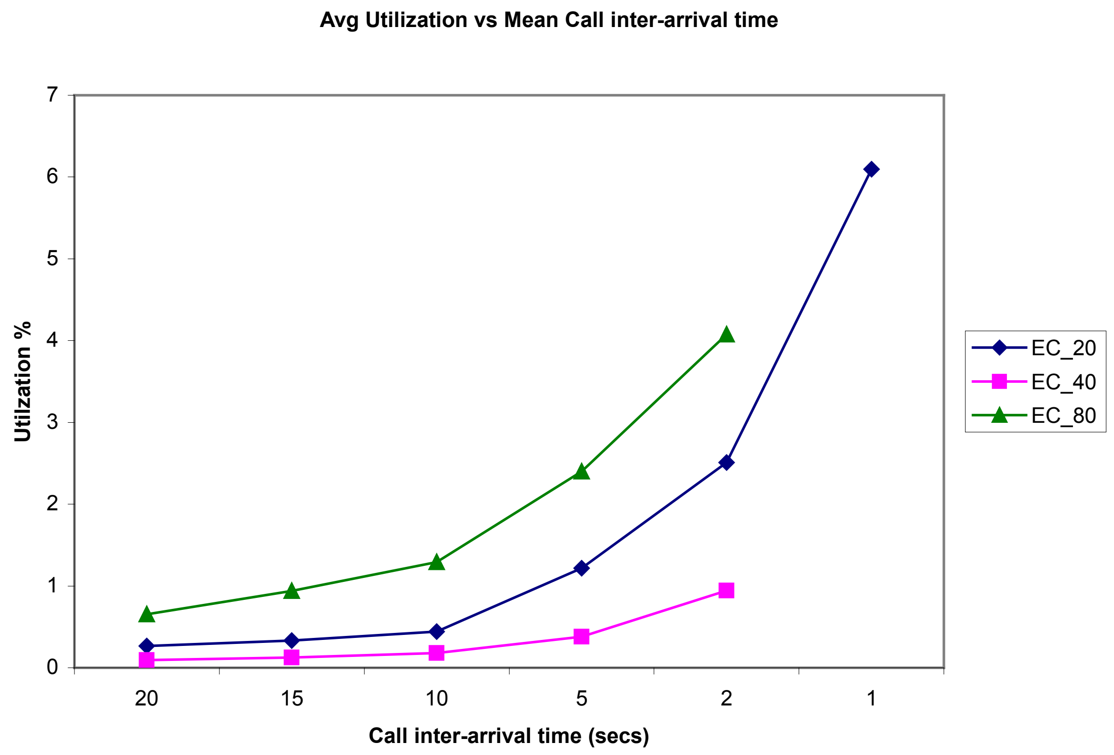

Fig 4.3 Average Utilization percentage vs. Load

It can be seen (and has already been shown – figure 3.9) that the variation of the utilization percentage with respect to topology size does not show a monotonic trend due to topological dissimilarities during scaling from one size to another.

4.4.4 PNNI data/ Flood

Figure 4.4 (a)-(c) give the variation of the average PNNI data / flood with respect to variation in load for the three test topologies – 20 nodes, 40 nodes and 80 nodes. The PNNI data (low, high and average) have been normalized to PNNI data /flood (bits /flood). This gives a better picture of how much PNNI data is flooded on an average.

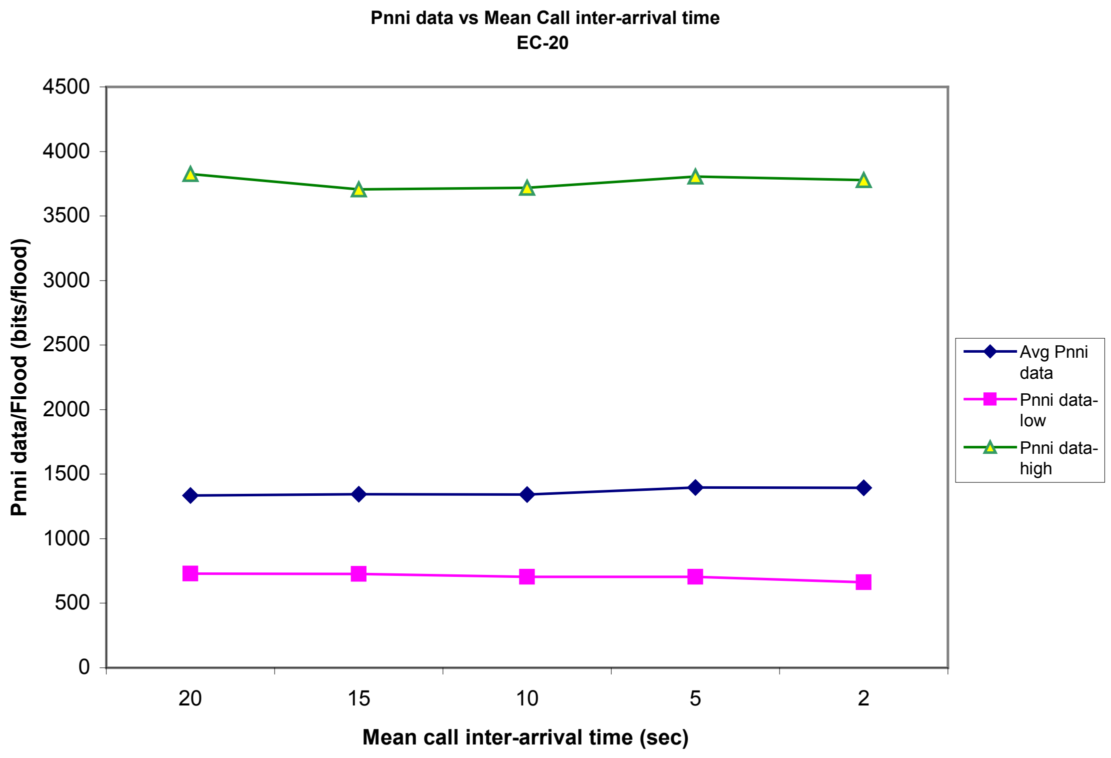

Fig 4.4(a) PNNI data/Flood vs. Load

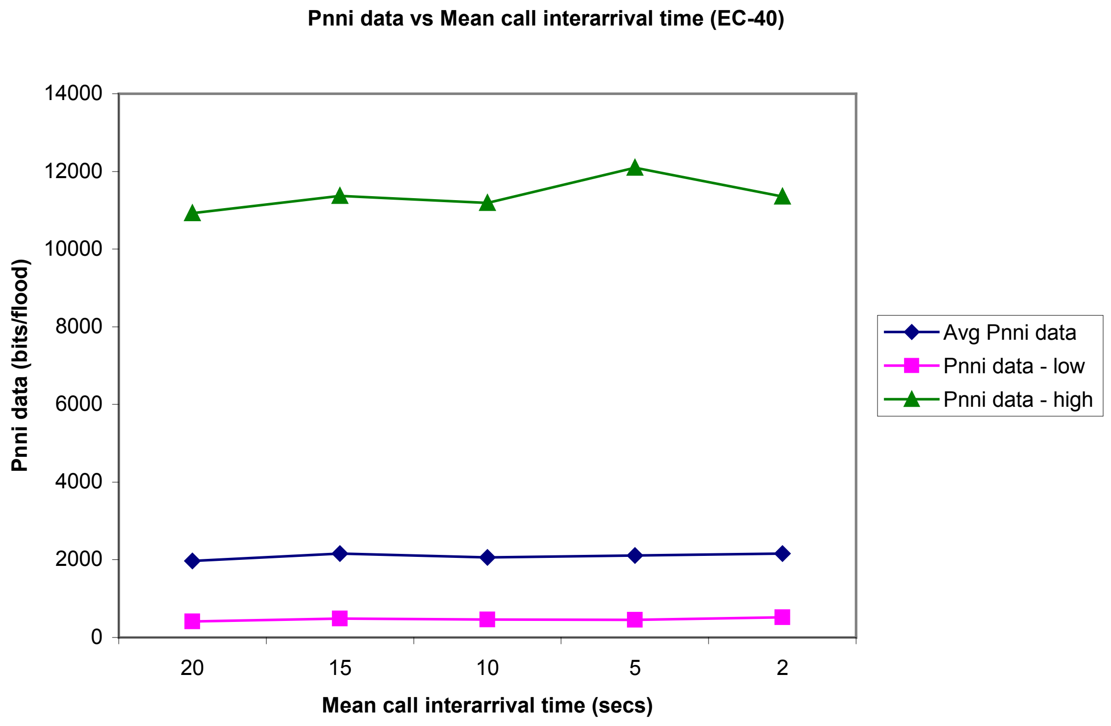

Fig 4.4(b) PNNI data/Flood vs. Load

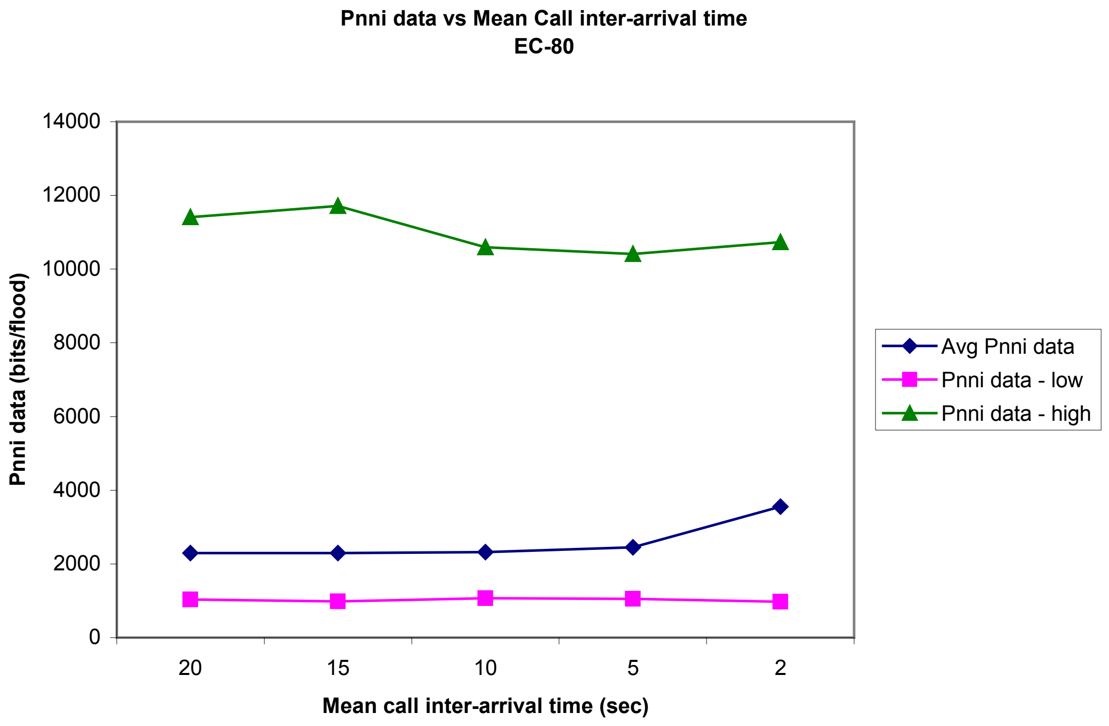

Fig 4.4(c) PNNI data/Flood vs. Load

The average PNNI data/flood for both EC-20 and EC-40 is almost constant for all loads, with a very small increase with increase in load. This is because the higher rate of significant changes at higher loads causes the nodes to pack a larger number of PTSEs into the PTSPs before flooding the PTSPs. The increase of average PNNI data with increasing load is very prominent in EC-80 for a mean call inter-arrival time of 2 sec

4.4.5 Average number of hops

Figure 4.5 depicts the variation of average number of hops with respect to load. The average number of hops can be expected to increase with an increase in load. This is because, as the load is increased, the *shortest path link* may not be able have sufficient bandwidth capacity to support all the calls. The call is then routed through another link, which will not be the shortest path. Thus the average number of hops can be expected to increase.

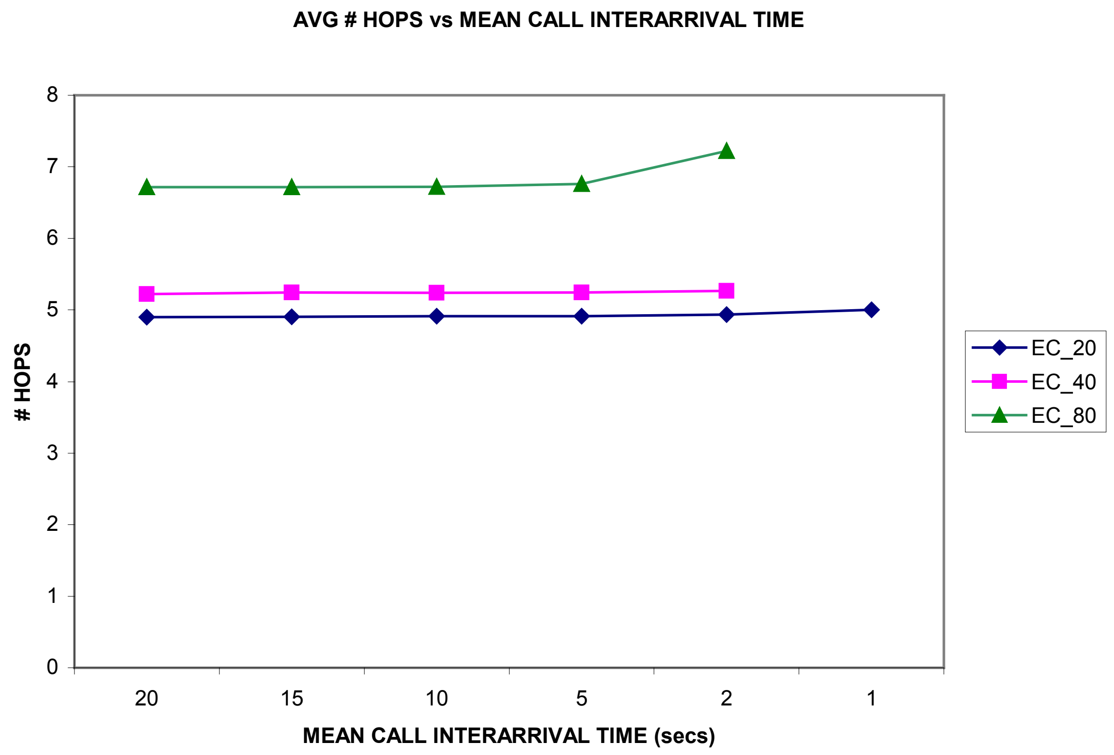

Fig 4.5 Average hops vs. Load

Surprisingly, the average number of hops seem to be almost unaffected by the variation in load. This is because that the topologies have been constructed such that link bandwidth is more than adequate to support all the calls. The slight variation that is seen for 80 nodes at high load is because more calls fail at higher loads in bigger topologies, which affects the value of the average number of hops.

4.5 Problems faced

The following were the problems faced during the course of these sets of simulations:

1. The simulator encountered out-of-memory problems and timer related problems at very high loads (call arrival rate of 1 call per second or more) for topology sizes with 40 nodes or more. Hence it was decided to strain the simulator only up to a call arrival rate of 0.5 calls per second (call inter-arrival time of 2 seconds) for all the test topologies.

4.6 Conclusion

It can be thus concluded that as the load in a bandwidth-rich network is increased,

- Mean Call setup time increases
- Call success percentage decreases
- Average link utilization percentage increases
- Average PNNI data / flood increases (only in large-sized networks)
- Average number of hops remains unchanged (in general, but depends on topology)

Chapter 5

Effect of Peergrouping test

5.1 Introduction

The aim of this test is to study the effect of peergrouping on the performance of the network. This test enables us to determine the topology that gives the best performance of the desired metrics, for a given fixed number of nodes.

5.2 Test topology

All the test topologies constructed for this set of simulations are Cluster topologies with 160 nodes and 2 levels of hierarchy. Inter-Peergroup links are all OC-12 links, Intra-Peergroup links are all OC-3 links, and the hosts are connected to the nodes through 10 Mbps links. In the figures for the topologies, a filled circle represents the leader of each peergroup. The hosts in the network generate 25 calls each.

5.2.1 Cluster 4x40

This topology consists of 160 nodes, grouped into 4 clusters (Peer groups), each with 40 nodes. The 40 nodes are further grouped into 10 *mini-clusters* of 4 nodes each. In figure -5.1 the entire intra-peergroup connections are not shown; instead, the interconnections from only one mini-cluster to all other mini-clusters are shown. It can be seen that *not all* mini-clusters are connected to one another. Each mini-cluster has 7 links emanating from it. The remaining intra-peergroup connections are constructed in a similar fashion.

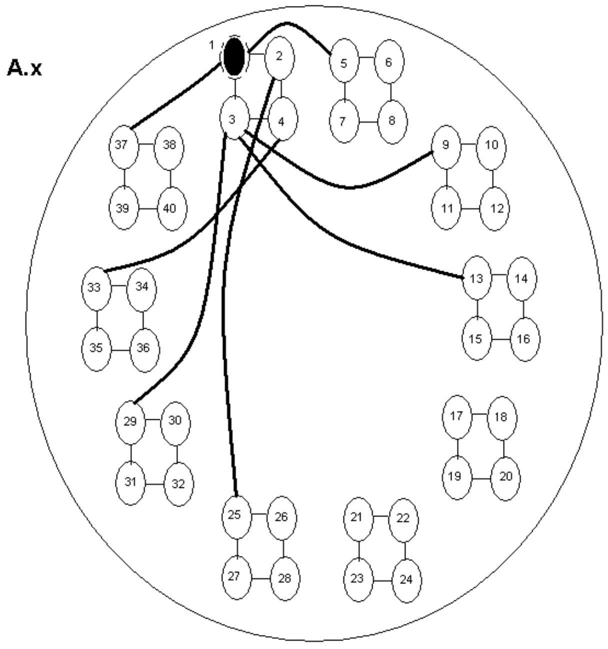

Fig 5.1 4x40 Intra-Peergroup connections

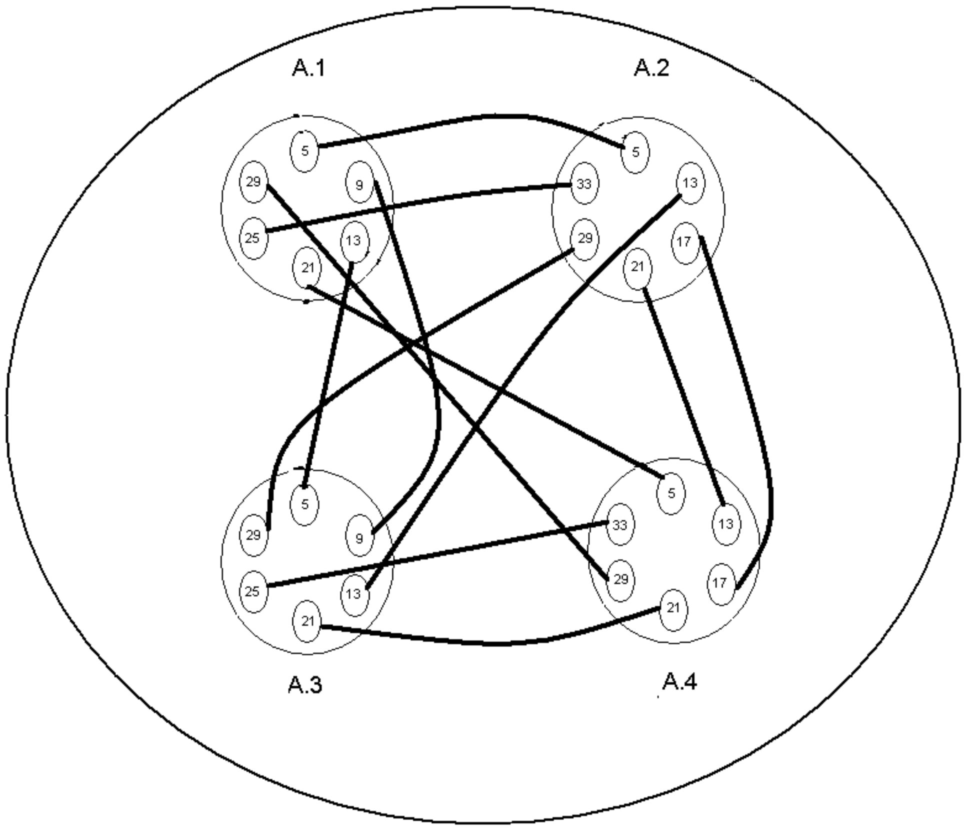

Fig 5.2 4x40 Inter-Peergroup connections

As shown in figure 5.2, the border nodes in each peergroup are identified as 5,9,13,21,23 and 29 (for Peergroups A.1 and A.3) and 5,13,17,21,29 and 33 (for Peergroups A.2 and A.4). Each border node has one link originating from it.

5.2.2 Cluster 5x32

This topology consists of 5 clusters (peergroups) each with 32 nodes in it. There are 8 mini-clusters in each peergroup. Figure 5.3 is a partial representation of the intra-peergroup connections. The entire interconnection can be constructed following the same pattern as shown in fig 5.3.

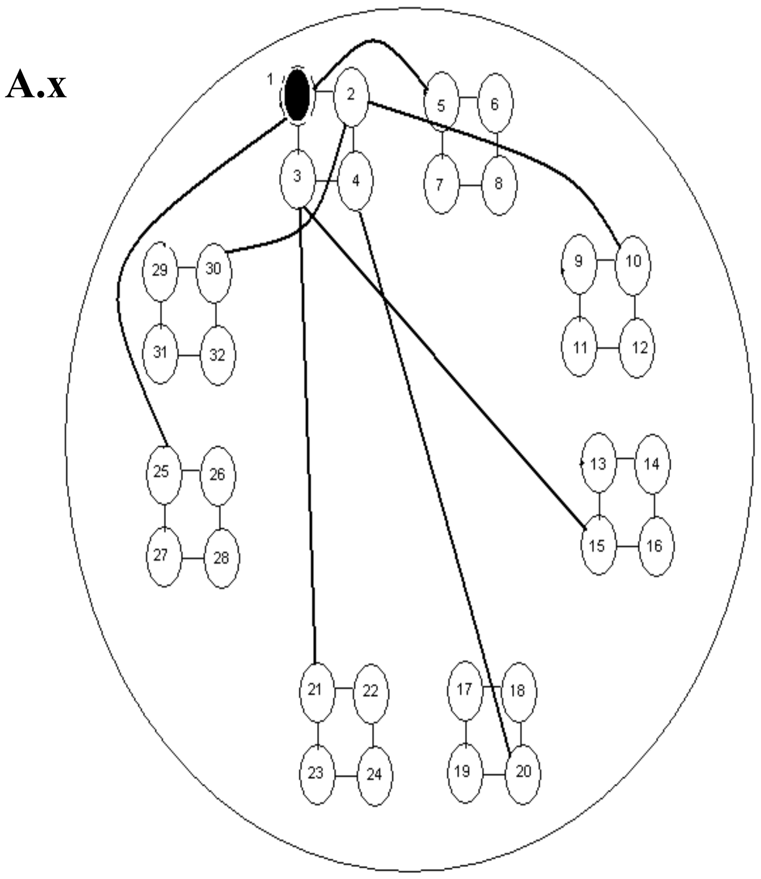

Fig 5.3 5x32 Intra-Peergroup Connections

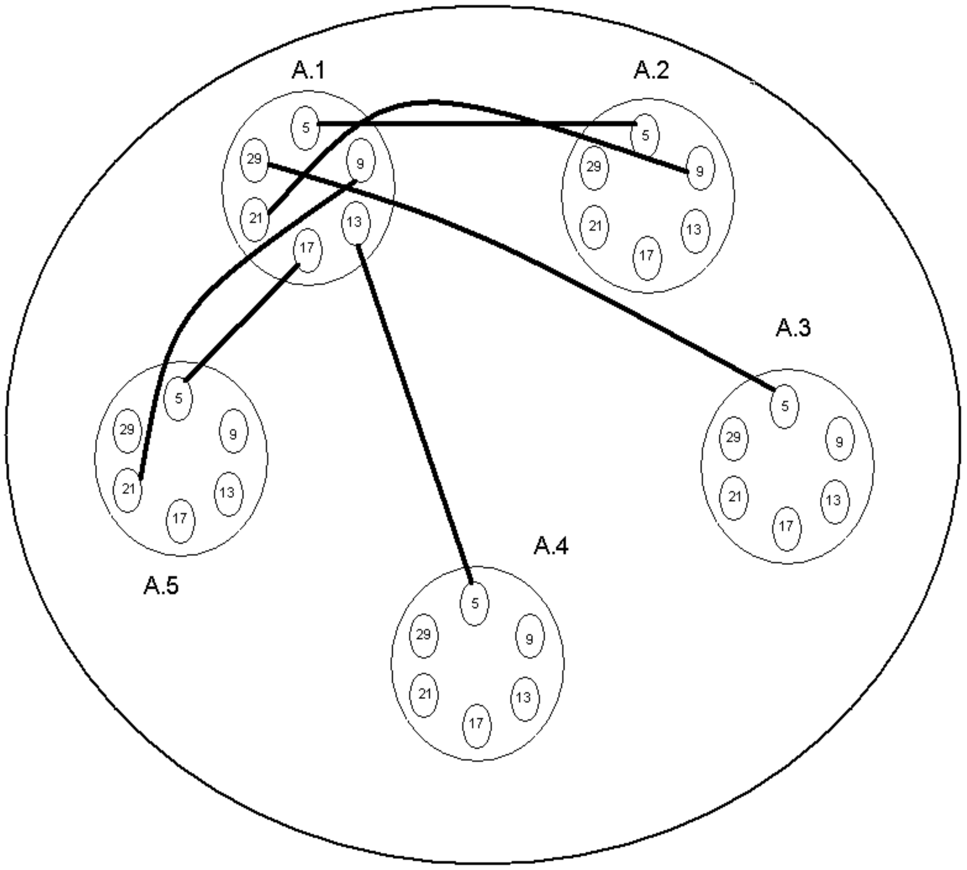

Fig 5.4 5x32 Inter-Peergroup connections

The border nodes are identified as 5,9,13,17,21 and 29. Fig 5.4 gives only the partial interconnections between various peergroups. The entire topology can be constructed by having 6 links originate from each peergroup and by following the same pattern as Fig-5.4.

5.2.3 Cluster 8x20

This topology consists of 160 nodes grouped in 8 clusters (Peer groups), each with 20 nodes in it. The 20 nodes in each peergroup are grouped as 5 mini-clusters of 4 nodes each. Each node has one intra-peergroup link originating from it. Fig 5.5 is a complete representation of the intra-peergroup interconnections in the topology.

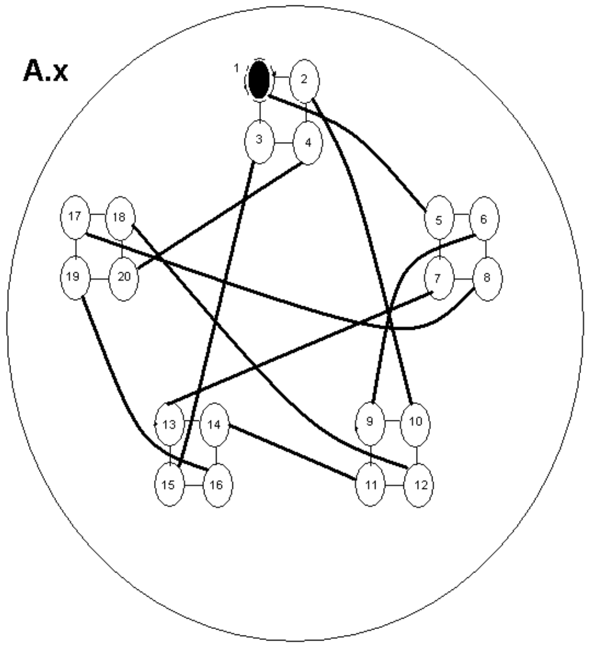

Fig 5.5 8x20 Intra-Peergroup connections

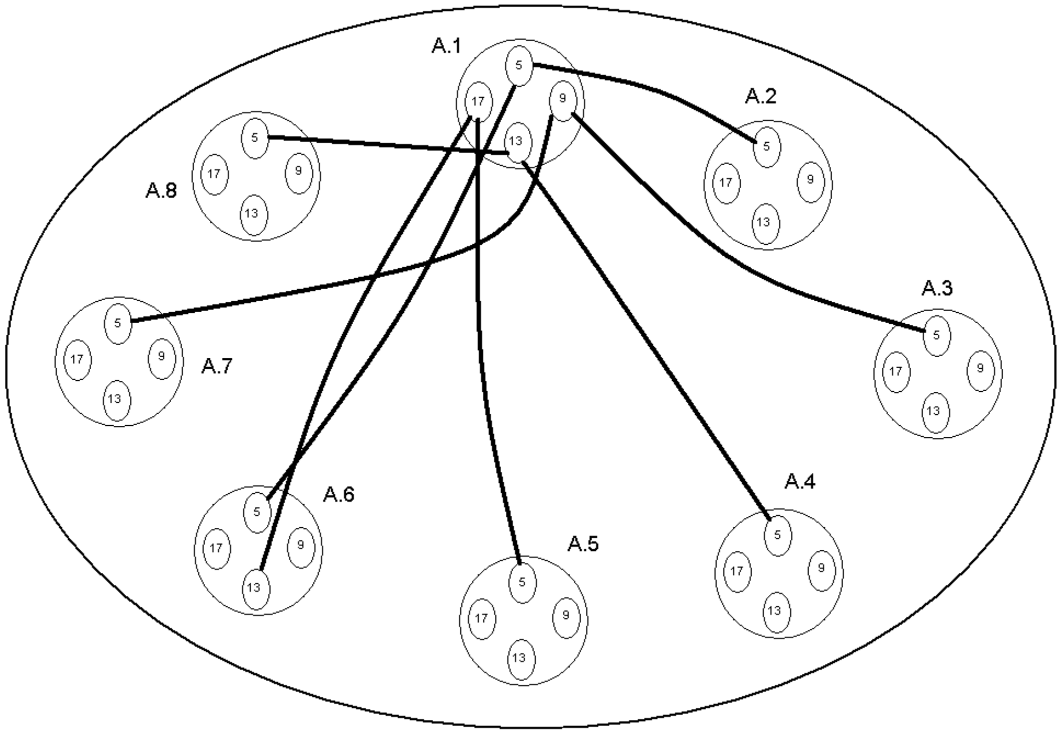

Fig 5.6 8x20 Inter-Peergroup connections

The border nodes are identified as 5,9,13 and 17. Fig 5.6 is an incomplete representation of the fully connected network. By configuring each border node to have two inter-peergroup links originate from it, and by following the same patter as in Fig 5.6, the topology can be completely constructed.

5.2.4 Cluster 10x16

This topology consists of 160 nodes grouped into10 clusters (Peer groups), each with 16 nodes in it. The 16 nodes are further grouped as 4 mini-clusters of 4 nodes each. Fig 5.7 gives complete details about the intra-peergroup connections.

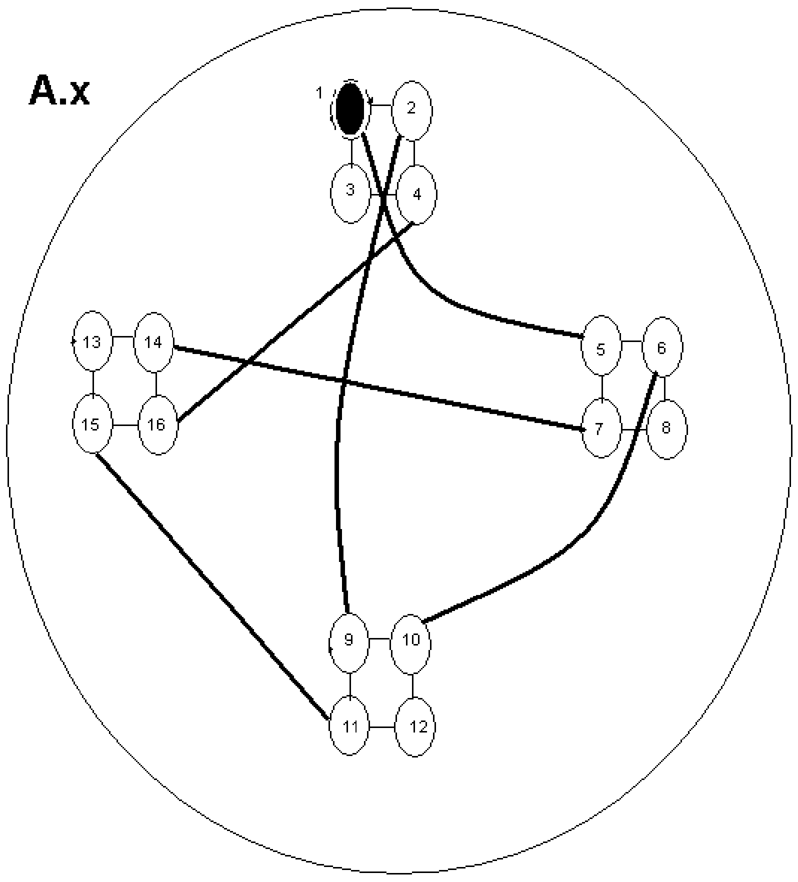

Fig 5.7 10x16 Intra-Peergroup connections

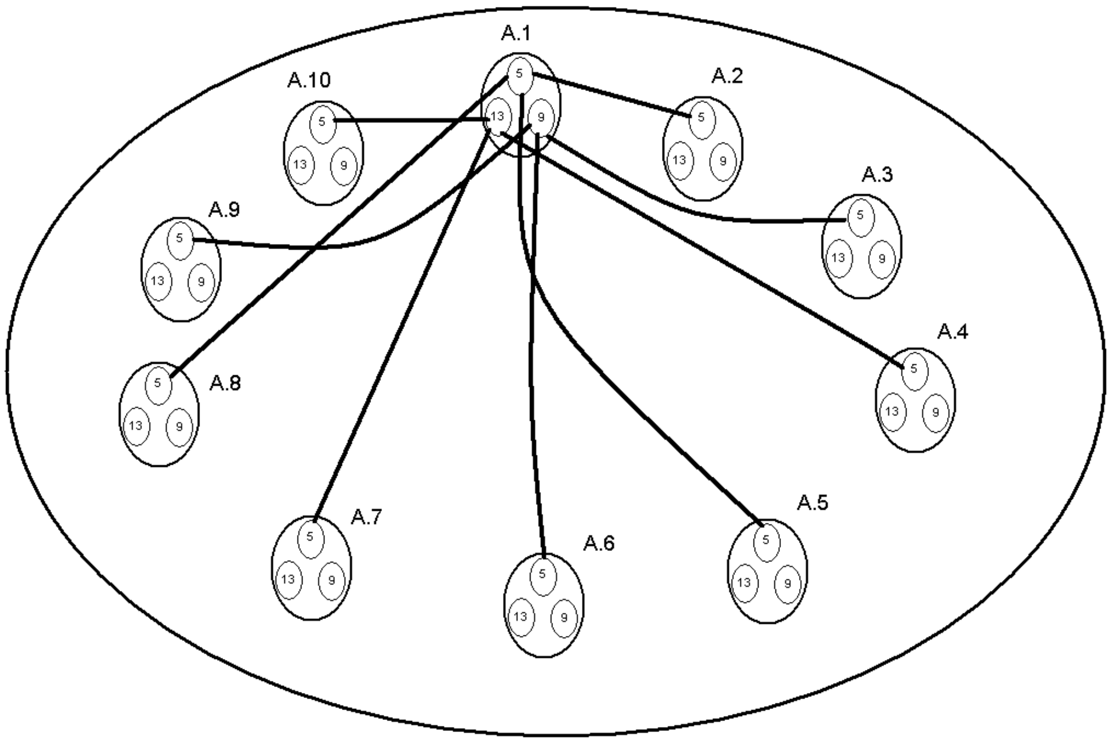

Fig 5.8 10x16 Inter-Peergroup connections

The border nodes are identified as 5, 9 and 13. As the number of peergroups increase, the inter-peergroup interconnections become more and more complex. Fig 5.8 is an incomplete representation of the inter-peergroup interconnections. The entire topology can be constructed by having one interconnection from one peergroup to all other peergroups.

5.2.5 Cluster 16x10

This topology consists of 160 nodes grouped as 16 clusters (Peer groups), each with 10 nodes. This topology is slightly different from the other test topologies in this set of simulations in that it has the 10 nodes in each peergroup grouped into 2 mini-clusters of *5 nodes* each. The mini-clusters are interconnected through 3 links. Fig 5.9 gives a complete view of the intra-peergroup interconnections.

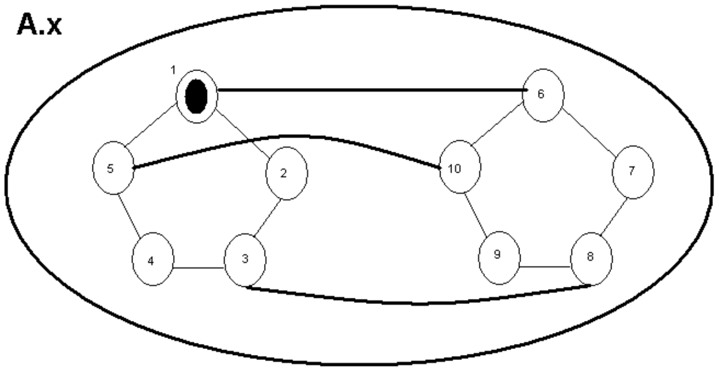

Fig 5.9 16x10 Intra-Peergroup connections

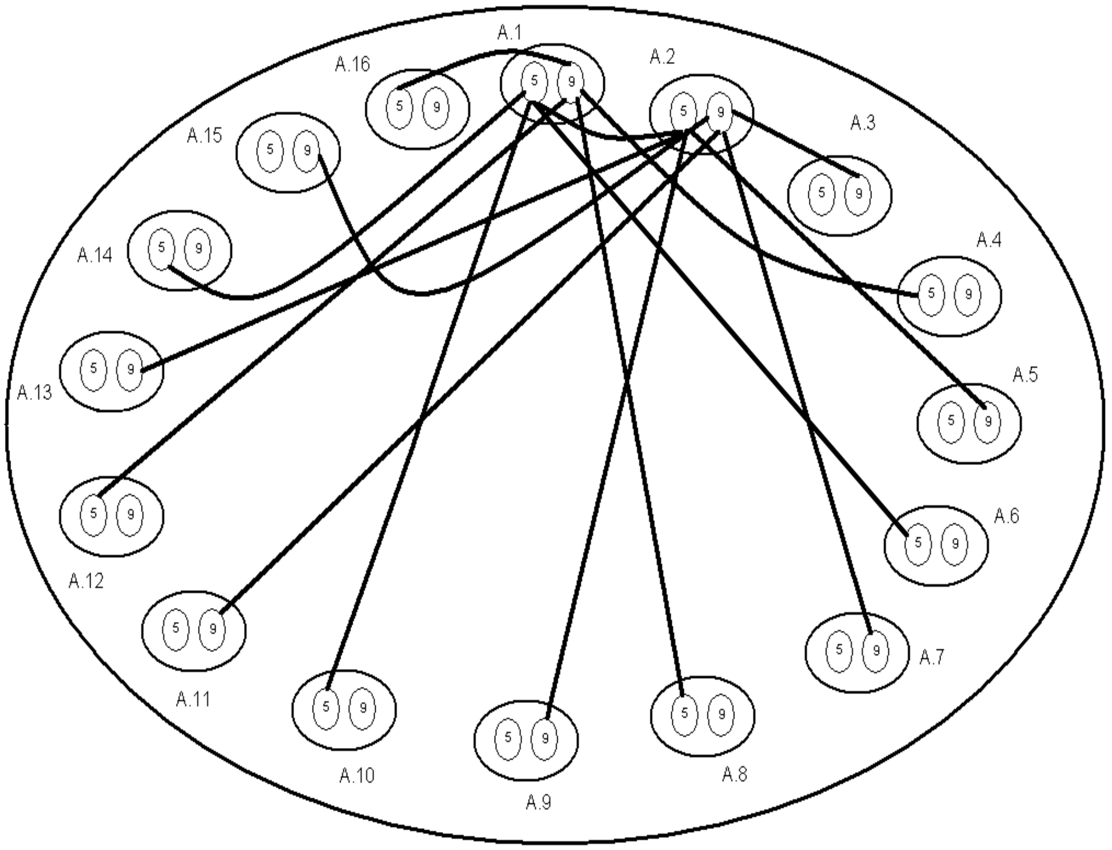

Fig 5.10 16x10 Inter-Peergroup connections

The border nodes are identified as 5 and 9. Fig 5.10 gives a partial representation of the actual inter-peergroup interconnections in the topology. The inter-peergroup interconnections can be completed by following the pattern exhibited by the partial interconnections.

5.3 Performance Metrics

The metrics of *primary* interest in this set of simulations are:

- Mean Call setup time
- Call success percentage
- Average Database size
- Convergence time

Along with these metrics, the other metrics such as Average number of hops, Average utilization percentage, total floods and PNNI data / flood are also monitored.

5.4 Results and Analysis

5.4.1 Convergence time

Figure 5.11 shows the variation of convergence time with respect to the number of peergroups. As the number of peergroups increases, we can expect the convergence time to decrease. There is a convergence number associated with each peergroup, which gives the number of nodes that each member in the peergroup has to converge with. This number is the sum of intra-peergroup links and the inter-peergroup links (links that the border nodes share with the border nodes of other peergroups); the number of links (and hence the convergence number) has a direct bearing on the convergence time. The convergence time is composed of 2 parts: intra-peergroup convergence time, and inter-peergroup convergence time. If the number of Peergroups increases (for a fixed total number of nodes in the topology), the number of intra-peergroup links decreases, while the number of inter-peergroup links increases; hence the convergence time keeps decreasing as the number of peergroups increases, until a point where the increase in inter-peergroup links more than offset the decrease in intra-peergroup links.

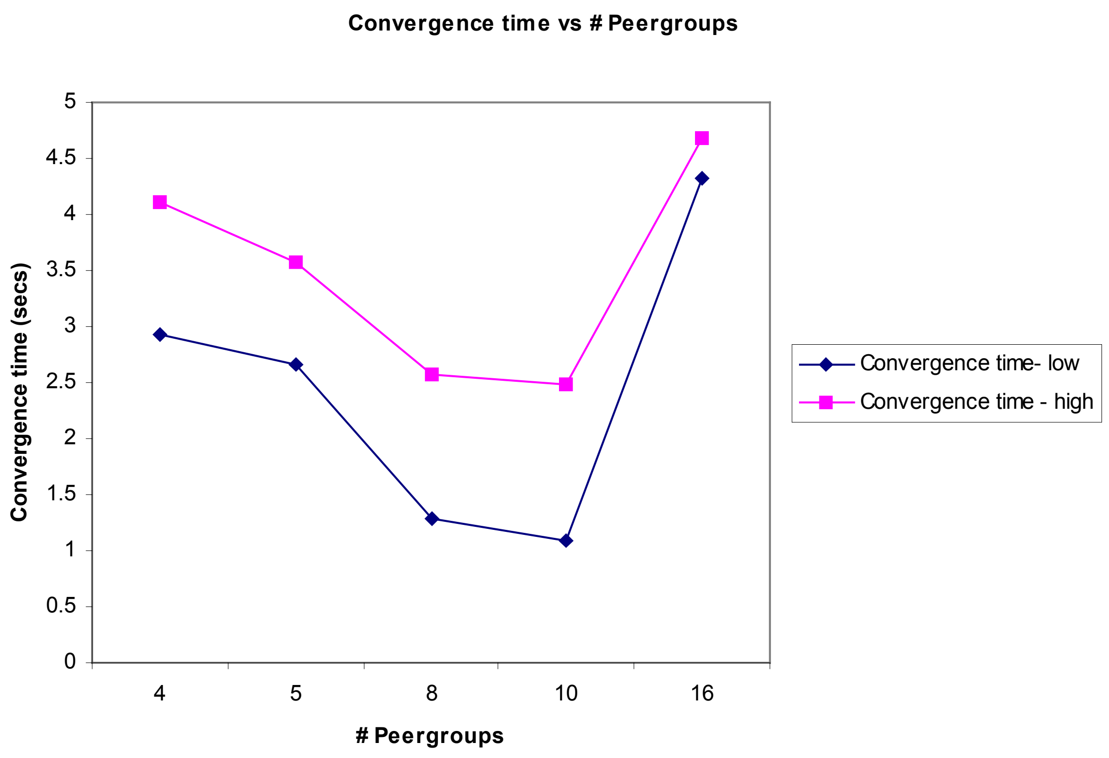

Fig 5.11 Convergence time vs. Peergroups

In the simulations conducted, the convergence time keeps decreasing as the number of topology moves from 4x40 to 10x16, but increases at 16x10. For example, in 10x16 each node has to converge with 16 other nodes within the same peergroup, while it needs to converge with only border nodes in 10 other peergroups. But in 16x10, the intra-peergroup convergence number in only 10, while the inter-peergroup convergence is 16. In this case, inter-peergroup convergence takes longer than intra-peergroup convergence. This validates the hypothesis just stated.

5.4.2 Mean Call setup time

Figure 5.12 shows the variation of mean call setup time as a function of number of peergroups. The mean call setup time decreases as the number of peergroups increases (for a fixed total number of nodes). The call setup time can again be broken down into three components: time taken for (source) intra-peergroup routing, time taken for inter-peer group routing and time taken for (destination) intra-peergroup routing (ignoring variation in the process time taken at each node, since the process time at each node is assumed to be the same). Since the destination and source intra-peergroup routing are similar events, they shall be referred as simply intra-peergroup routing.

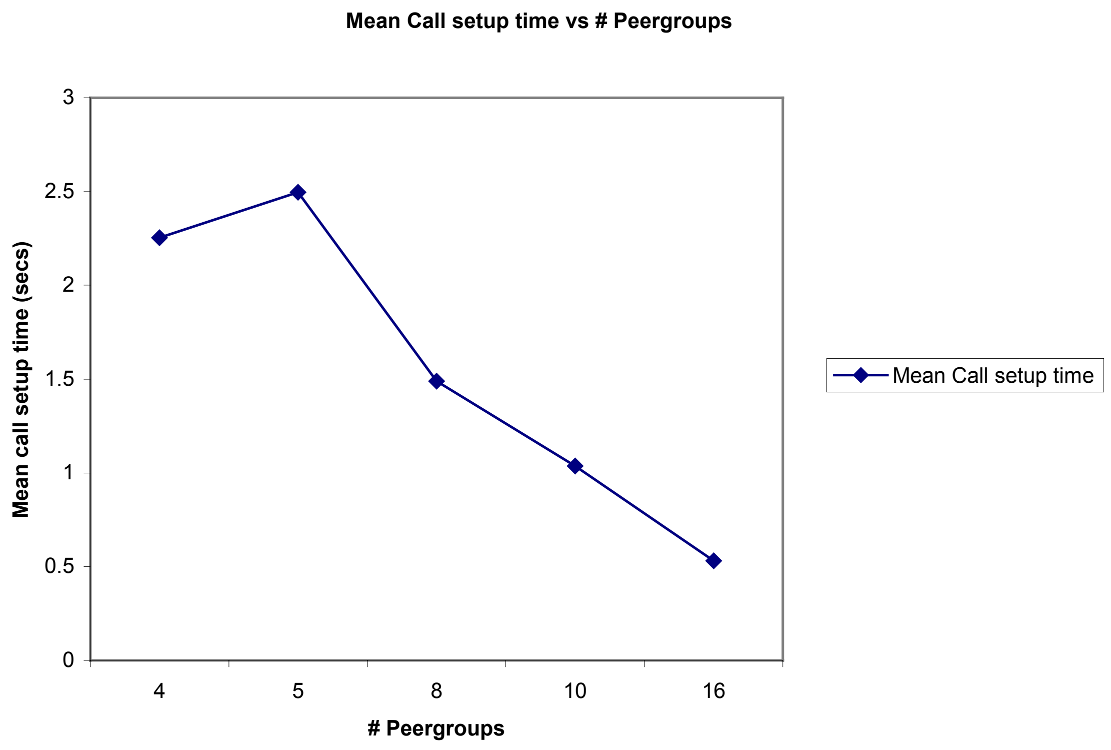

Fig 5.12 Mean Call Setup time vs. Peergroups

With a lesser number of peergroups, intra-peergroup routing is more complex than inter-peergroup routing. This is because with lesser number of peergroups, there are more nodes in each peergroup, thus making routing slightly more complex. The topology has been designed such that inter-peergroup routing is trivial i.e. every peergroup has at least one direct link to every other peergroup, and the inter-peergroup routing time is almost a constant. Hence the call setup time keeps decreasing as the number of peergroups increases, until it reaches a point, where the inter-peergroup routing times starts to dominate.

In the simulations conducted, the mean call setup time decreases with an increase in the number of peergroups. This validates the hypothesis just stated (the limitations of the simulator did not facilitate running a simulation with a very large number of peergroups, to show that the inter-peergroup routing time becomes a significant factor influencing the call setup time).

There is an apparent tradeoff between cost and call setup time - by increasing the number of peergroups, the mean call setup time does decrease; but by constructing a topology such that the inter-peergroup routing is trivial, the number of high bandwidth links has been increased considerably (at least one high bandwidth link between each pair of peergroups), which adds to the cost.

5.4.3 Call Success percentage

Figure 5.13 shows the variation of call success percentage as a function of number of peergroups. The call success rate is closely related to the mean call setup time; as the mean call setup time increases, the call success percent decreases - more calls are rejected since the nodes (either transit nodes or destination nodes) are busy. In these simulations, the calls do not fail because of lack of capacity in the links.

As the number of peergroups increases, the mean call setup time decreases; hence the call success percentage increases. Moreover, more the number of peergroups, lesser number of nodes are abstracted into a peergroup advertisement, which increases reachability, and hence the call success percentage increases for this reason as well.

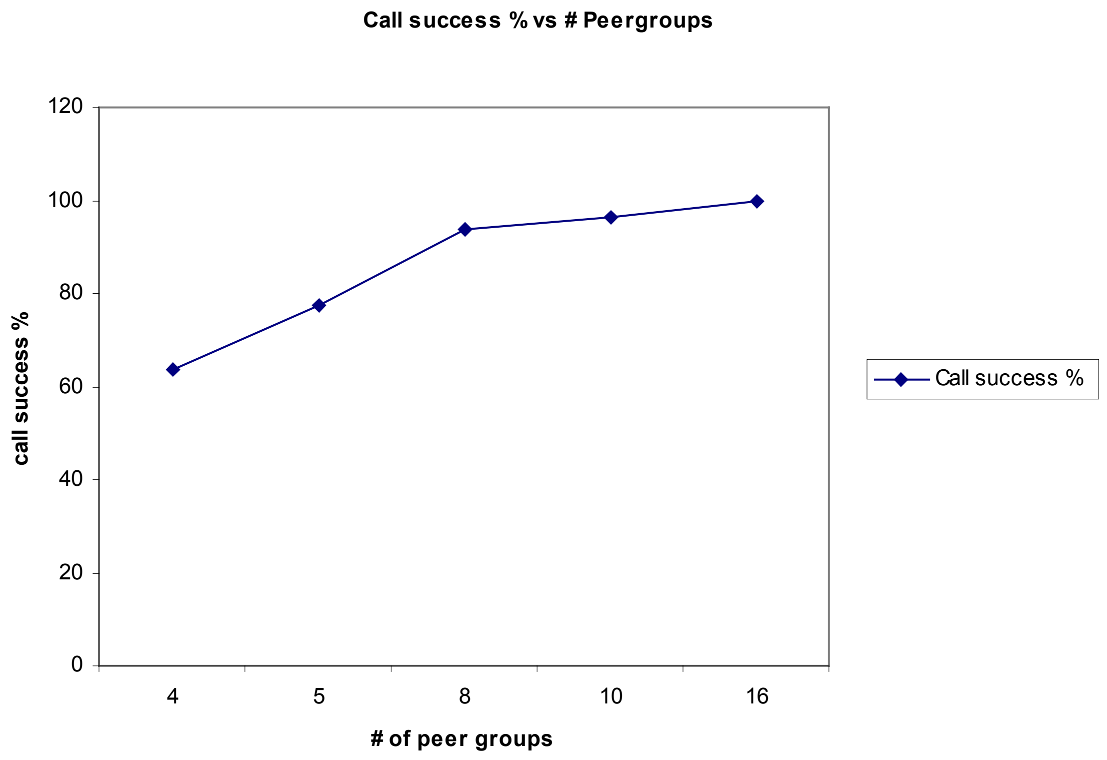

Fig 5.13 Call success rate vs. Peergroups

5.4.4 Average Database size

Figure 5.14 shows the variation of average database size as the number of peergroups is varied. The average database size decreases as the number of peergroups increases (up to a point). This is because as the number of peergroups increases, the database has to record (contain) only lesser information about its neighbors. This is closely related to convergence number. The information contained in the database can be broken down into two components - nodes within the same peergroup and border nodes.

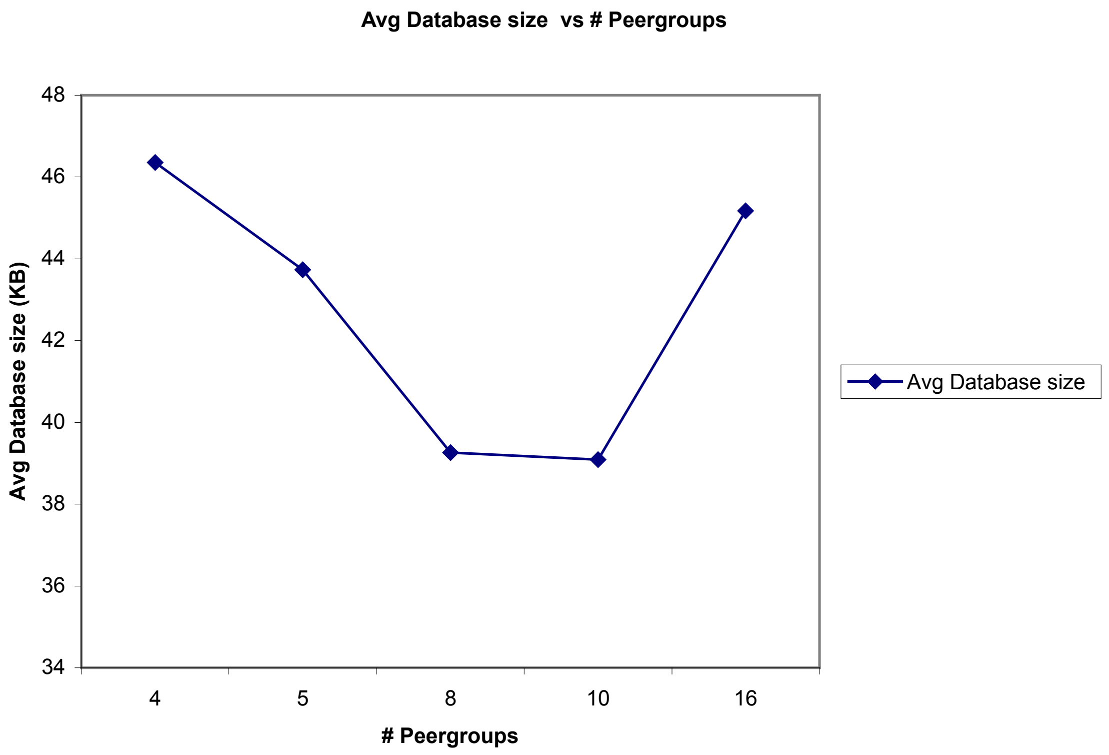

Fig 5.14 Average Database size vs. Peergroups

The database size keeps decreasing until a point after which the database size increases. This is because the number of border nodes becomes significant when compared to the number of nodes within the same peergroup. This trend can very well be seen from the results of the simulations

5.4.5 Average number of hops

Figure 5.15 shows the variation of average number of hops as a function of the number of peergroups. The average number of hops is simply a measure of how identical each topology is, which in turn is a measure of the confidence level of the results obtained.

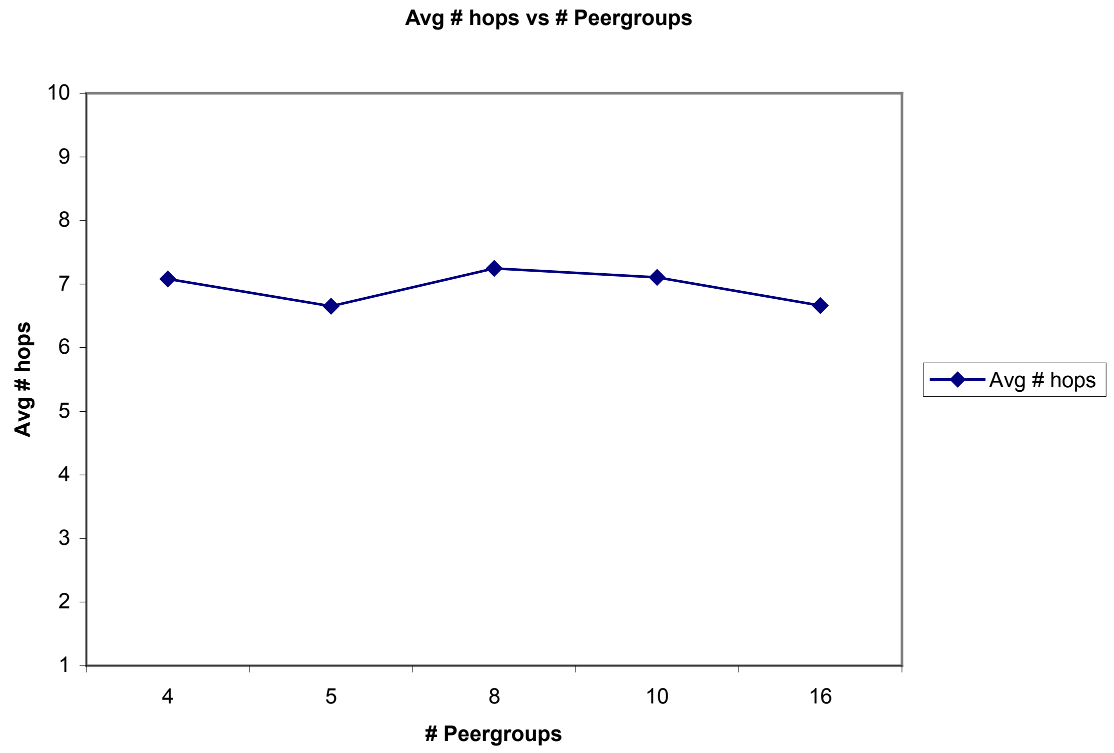

Fig 5.15 Average hops vs. Peergroups

In the simulation experiments conducted, the average number of hops in each case is almost the same, and thus it could be said that the topologies had no significant influence or bias on the results obtained. The slight variations in the average number of hops are due to the dissimilarities in the way inter-peergroup and intra-peergroup connections are made for each peergroup.

5.4.6 Floods per node

The total number of floods is composed of two components: floods when a significant change occurs in one of the node's database and the periodic flooding (hello packet, periodic PTSE refresh etc). The trend in the number of floods is dictated by the significant changes (floods, which are more frequent than the periodic floods)

Figure 5.16 shows the variation of floods per node with respect to a variation in the number of peergroups. With a lesser number of peergroups, there will be more significant changes within a peergroup; hence this will trigger intra-peergroup flooding; inter-peergroup flooding takes place mainly for two reasons: if there is a significant change in one of the border nodes or if a peergroup loses reachability to one of the other peergroups. With a lesser number of Peergroups, the intra-peergroup floods dominate; with the increase in the number of peergroups, the inter-peergroup floods start to dominate. A point is reached whereby the increase in inter-peergroup floods will be more pronounced than the decrease in intra-peergroup floods.

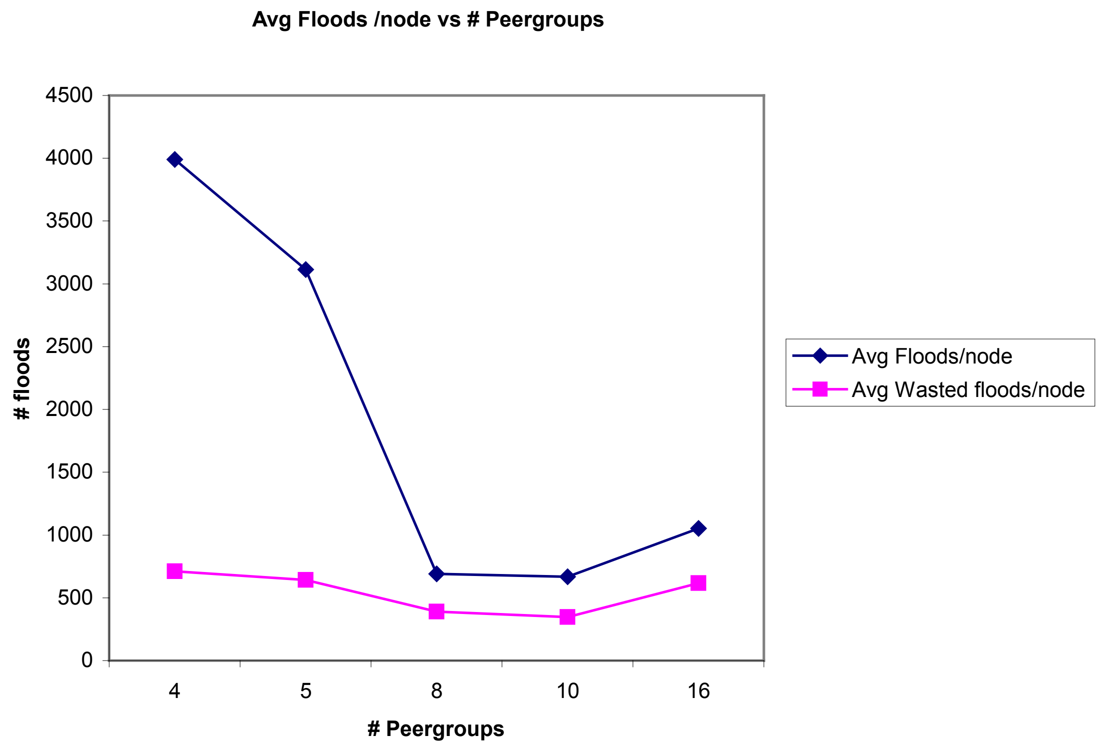

Fig 5.16 Average Floods/node vs. Peergroups

In this set of simulations, the number of nodes is maintained as a constant (160); hence the total floods and the floods per node both follow the same trend. The floods per node are taken as the metric of interest here. The results of the simulation behave exactly as described above. The number of floods decreases as the number of peergroups increases from 4 to 10, but again increases when the number of peergroups is 16. It can be seen that the wasted floods also follow the same trend as the total number of floods.

5.4.7 PNNI data per Flood

Figure 5.17 shows the variation of PNNI data per flood per node with respect to variation in the number of peergroups. The PNNI data per flood per node for similarly configured peergroups (for the same load) should remain unchanged.

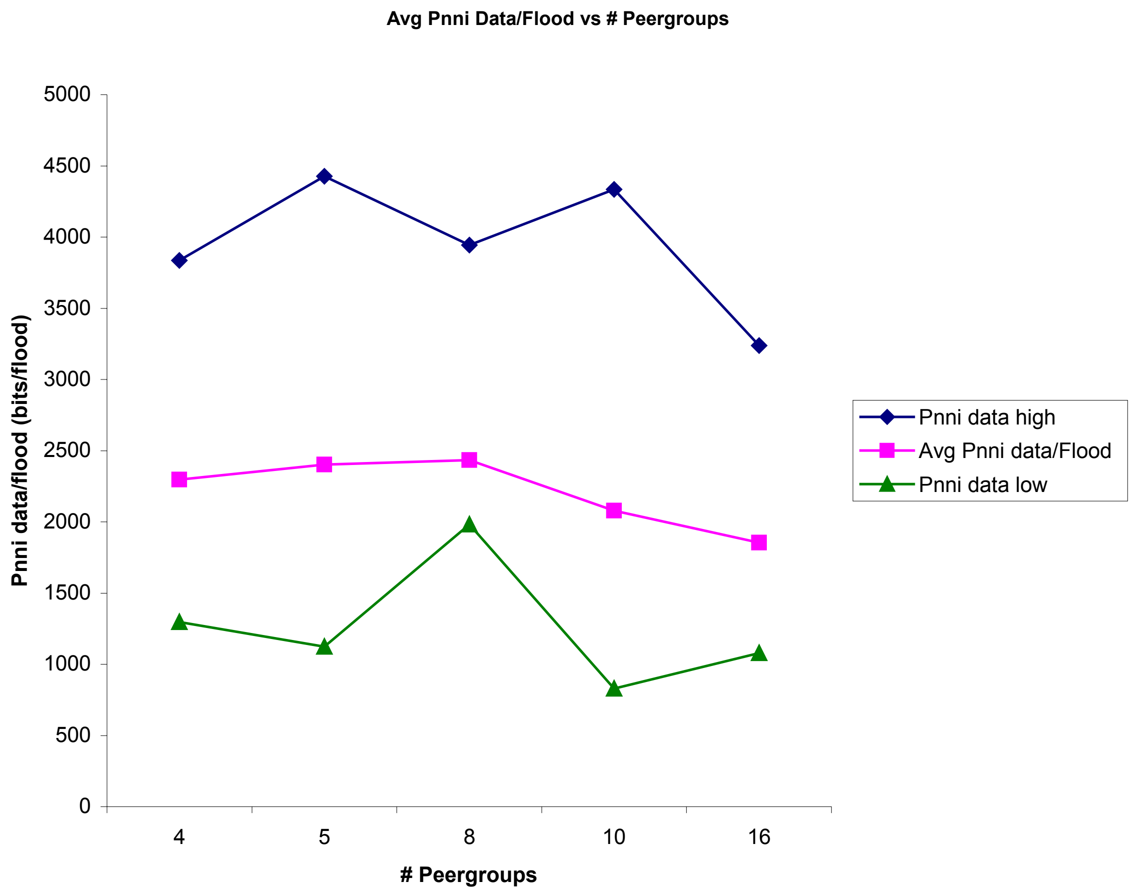

Fig 5.17 PNNI data per flood vs. Peergroups

The decrease in the average PNNI data per flood for 10x16 and 16x10 peergroups is due to the fact that, unlike the intra-peergroup connections in other peergroups, intra-peergroup links do not originate from all the nodes in the peergroup. One point to be noted here is that the 'Average PNNI data' metric is normalized over the *total number of floods*, while 'PNNI data – high' and 'PNNI data – low' have been normalized over *floods per node*.

5.6 Conclusion

It can be thus concluded that as the number of peergroups in a bandwidth-rich network is increased,

- Convergence time decreases initially, and then increases
- Mean Call setup time increases
- Call success percentage decreases
- Average Database size decreases
- Average PNNI data / flood remains unchanged
- Average number of hops remains unchanged (in general, but depends on topology)

Chapter 6

Effect of Hierarchy

6.1 Introduction

The aim of this set of simulations is to study the effect of hierarchical levels on the metrics that characterize the performance of the network.

6.2 Test topology

The topologies constructed for this set of simulations consist of a cluster topology with 96 nodes. The simulations are conducted on the same topology, but by varying the hierarchy from one-level (flat), two-levels, three-levels and four-levels. The topology has been constructed as follows. The 96 nodes are grouped as 6 'leaves' of 16 nodes each (see figure 6.1). Each leaf has 3 mini-clusters, of which 2 mini-clusters have 5 nodes each, while the third mini-cluster has 6 nodes in it. From each leaf, two nodes from the 6-node mini-cluster have links to interconnect to other leaves. The inter-leaf interconnections are carried out through OC-3 links, while the inter-mini-cluster connections are carried out through 20 Mbps links, and the nodes within a mini-cluster are connected through 10 Mbps links. The hosts are connected to the nodes via 5 Mbps links. Darkened circles represent the leader nodes in each topology. The number of calls generated is *50 per host*.

6.2.1 One-level (flat) hierarchy

This topology has been constructed such that it has just one-level hierarchy (or in effect no hierarchy at all); the topology is flat structured. All the nodes in the network fall are grouped under one single peergroup. Fig 6.1 is a representation of a leaf in the flat network. Fig 6.2 is shows partially the one-level hierarchical topology, with inter-leaf interconnections.

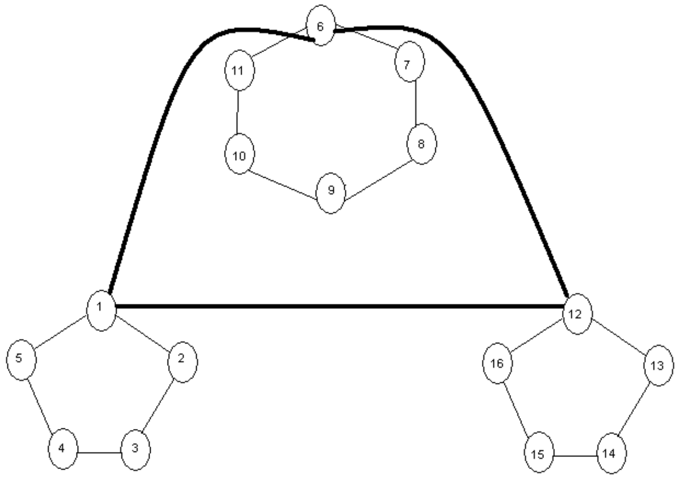

Fig 6.1 *Leaf* in the One-level (flat) network

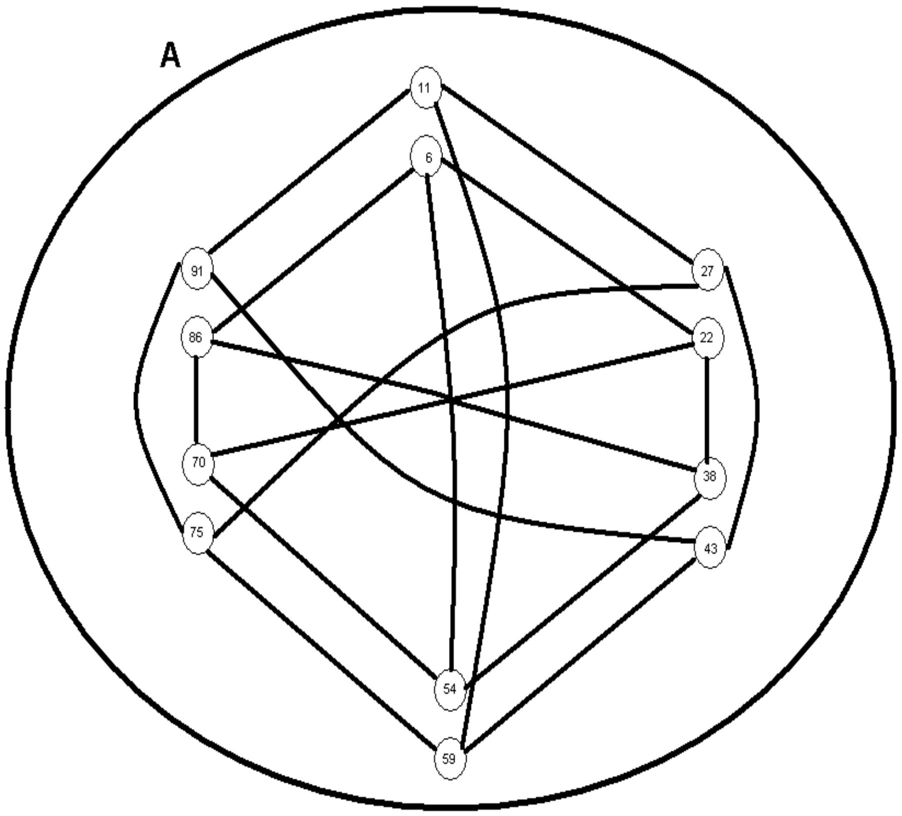

Fig 6.2 One-level Inter-leaf interconnections

6.2.2 Two-level hierarchy

This topology has been constructed such that there are two levels of hierarchy. Each leaf has been grouped under a separate peergroup as indicated in fig 6.3. Figure 6.4 gives the inter-peergroup connections, along with the logical-level of hierarchy (second-level).

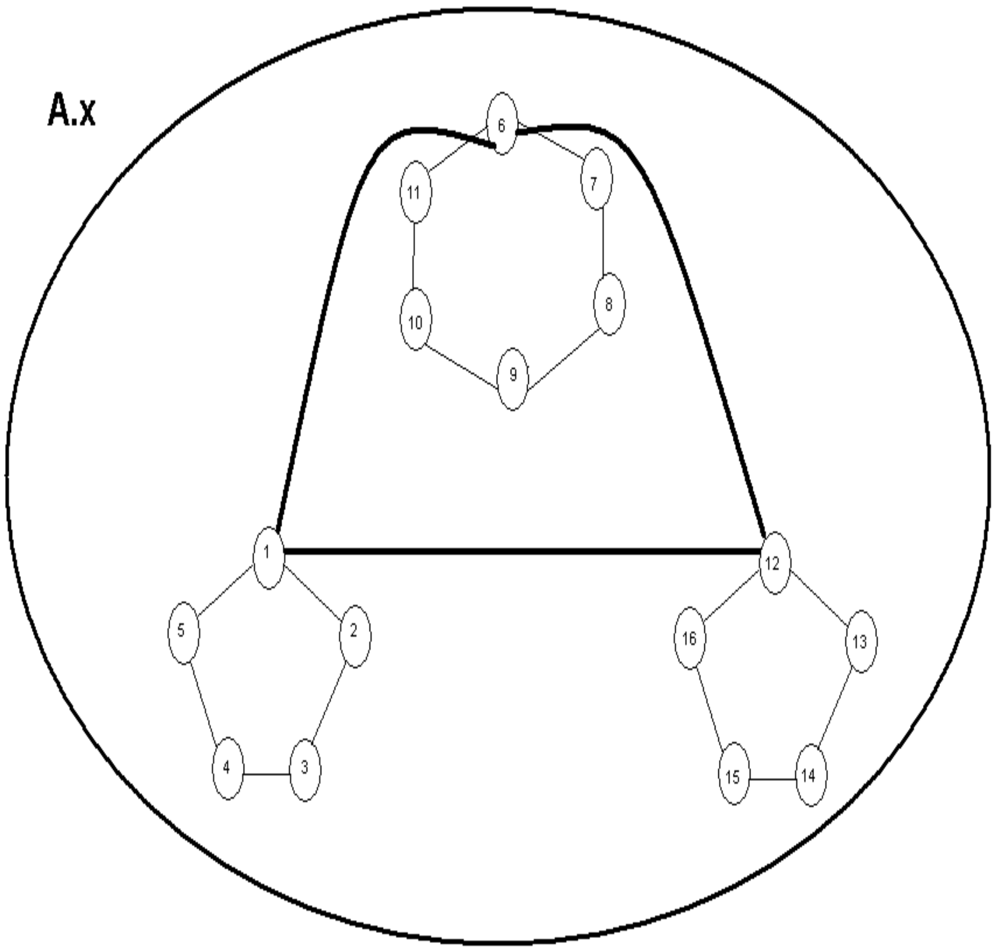

Fig 6.3 *Leaf* in a two-level hierarchical network

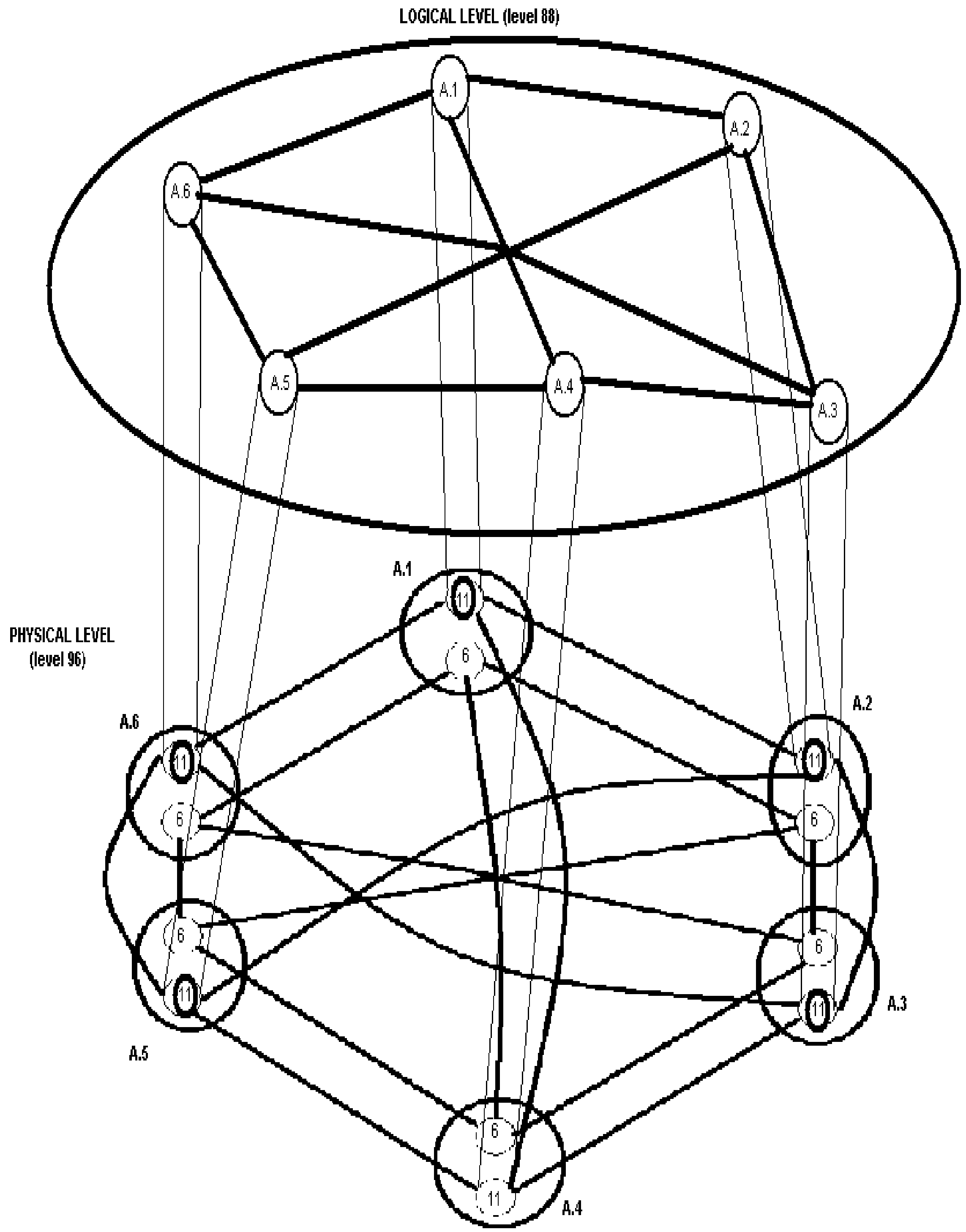

Fig 6.4 Inter-Peergroup connections in a 2-level hierarchical network

6.2.3 Three-level hierarchy

The next topology has been constructed with three-levels of hierarchy. Here, each mini-cluster is grouped into a peergroup, while the leaf serves as a logical level or a higher-level peergroup, which encompasses the mini-cluster peergroups. Fig 6.5 shows the representation of a leaf in a 3-level hierarchical network, along with its next higher-level (logical level) representation. Fig 6.6 shows the inter-peergroup interconnection at level-96 (lowest level of hierarchy), level-88 (second level of hierarchy) and level-80 (third level of hierarchy).

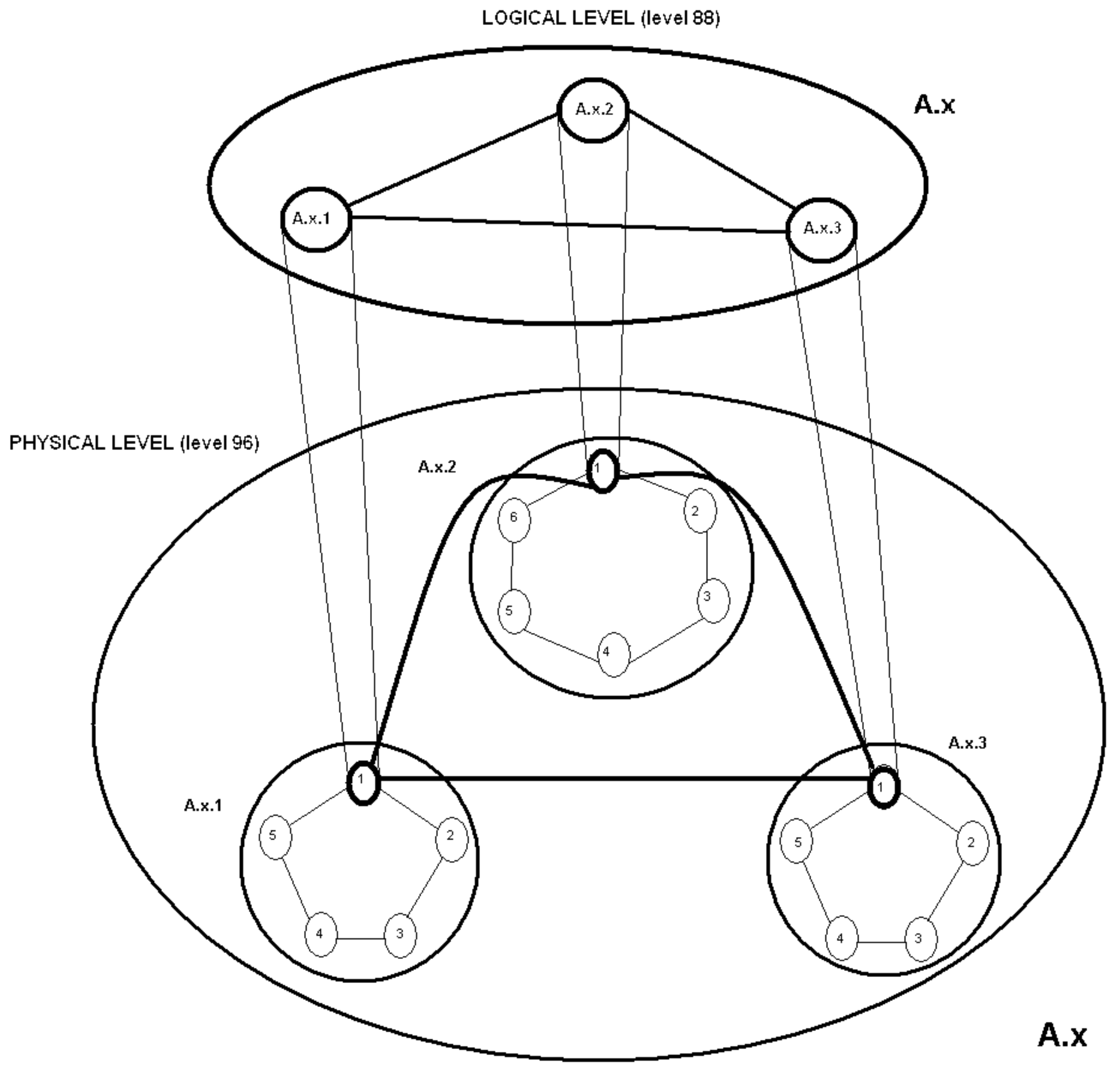

Fig 6.5 Leaf in a 3-level hierarchical network with its logical level representation

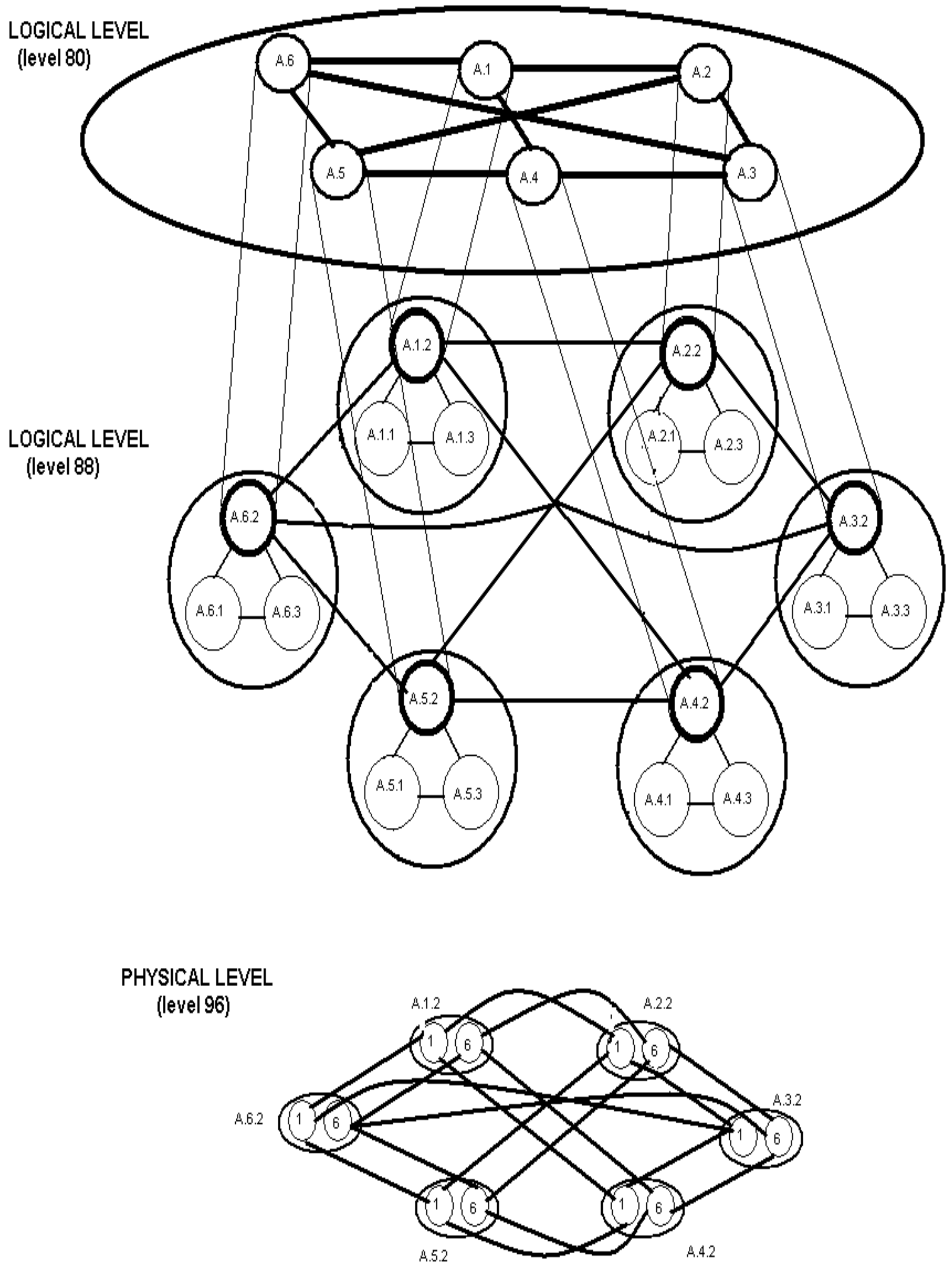

Fig 6.6 Complete 3-level hierarchical representation of the network

6.3 Performance metrics

The metrics of *primary* interest in this set of simulations are:

- Mean Call setup time
- Call success percentage
- Average Database size
- Convergence time
- Average number of hops

Along with these metrics, the other metrics such as Average utilization percentage, total floods and PNNI data / flood are also monitored.

6.4 Results and Analysis

6.4.1 Mean Call setup time

Figure 6.7 shows that the mean call setup time increases (not monotonically) with an increase in hierarchical levels. The call setup time is the sum of the queuing time at each node, the time to traverse the links and the routing time. The queuing time and the time to traverse the links can be assumed to be fairly constant. (The queuing time at each node is mainly dependent on the load in the network). With an increase in hierarchical levels, the number of decision making nodes increases and each decision-making node (the leader) does not have complete information about the entire network topology. Hence the routing decisions are not always optimal as the number of hierarchical levels increase; also, the routing decisions are made at each level, which increases the effective routing time.

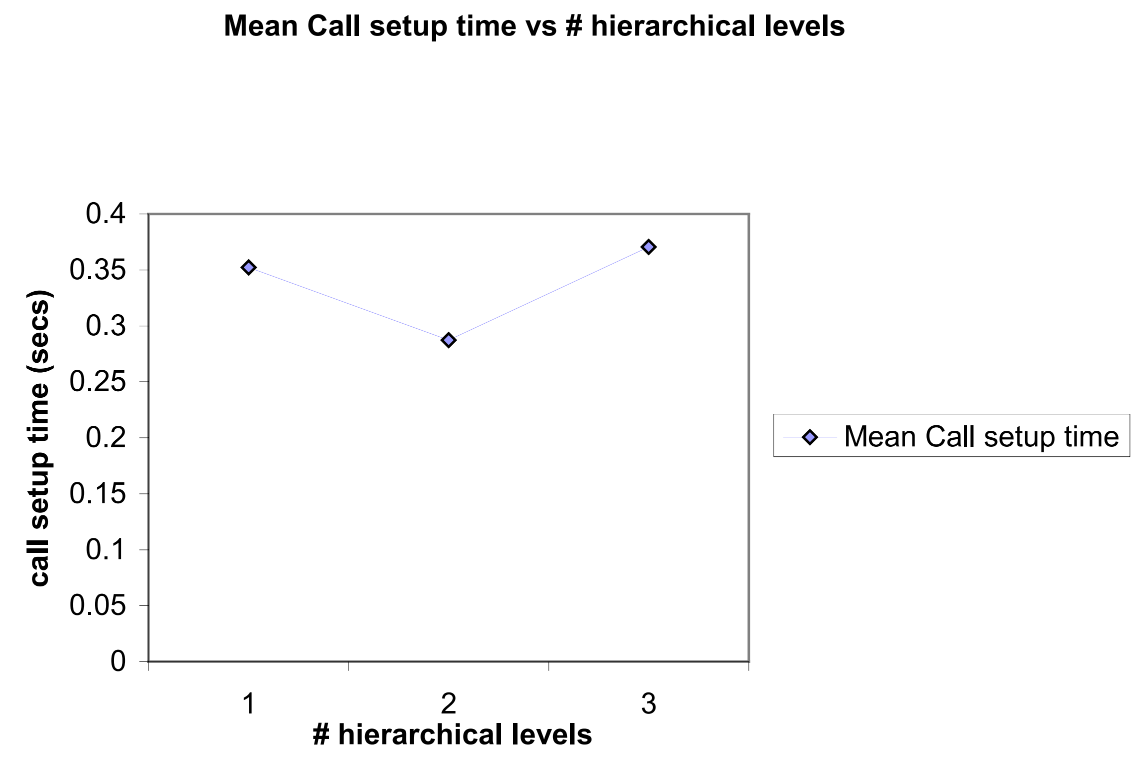

Fig 6.7 Mean Call setup time vs. levels of hierarchy

Thus the mean call setup time can be thought of as a function of the routing time of each call (with load and network topology remaining unchanged). The routing time can further be split as inter-peergroup routing time and intra-peergroup routing time. These results seem to contradict the results of "Effect of Peergrouping test". The time for flow of control data between hierarchical levels more than offsets the decrease in routing time due to peergrouping. Hence, an increase in the number of hierarchical levels increases the routing time, thus increasing the mean call setup time.

One exception is the results of 2-level hierarchy; here the increase in routing time due to flow of control data between hierarchical levels is less than the decrease in routing time due to the effect of peergrouping. It might be interesting to note that in Gowri Dhandapani's Master's thesis report (page 72) **[2]**, it was reported that the call setup time decreases with an increase in hierarchical levels; this was because the tests were conducted only on two topologies, and the results were affected by the above stated effect.

6.4.2 Average Database size

Figure 6.8 shows that the average database size decreases as the number of hierarchical levels increases. This is because, with an increase in the number of hierarchical levels, the routing decisions taken to setup a call are made by more than one node, and hence each node need not have a complete and accurate picture of the network topology; all the nodes need is complete reachability information at the highest level of hierarchy. This abstraction in reachability information considerably reduces the database size at each node. Stated otherwise, the reachability information is spread over the entire topology, rather than clustered at each node.

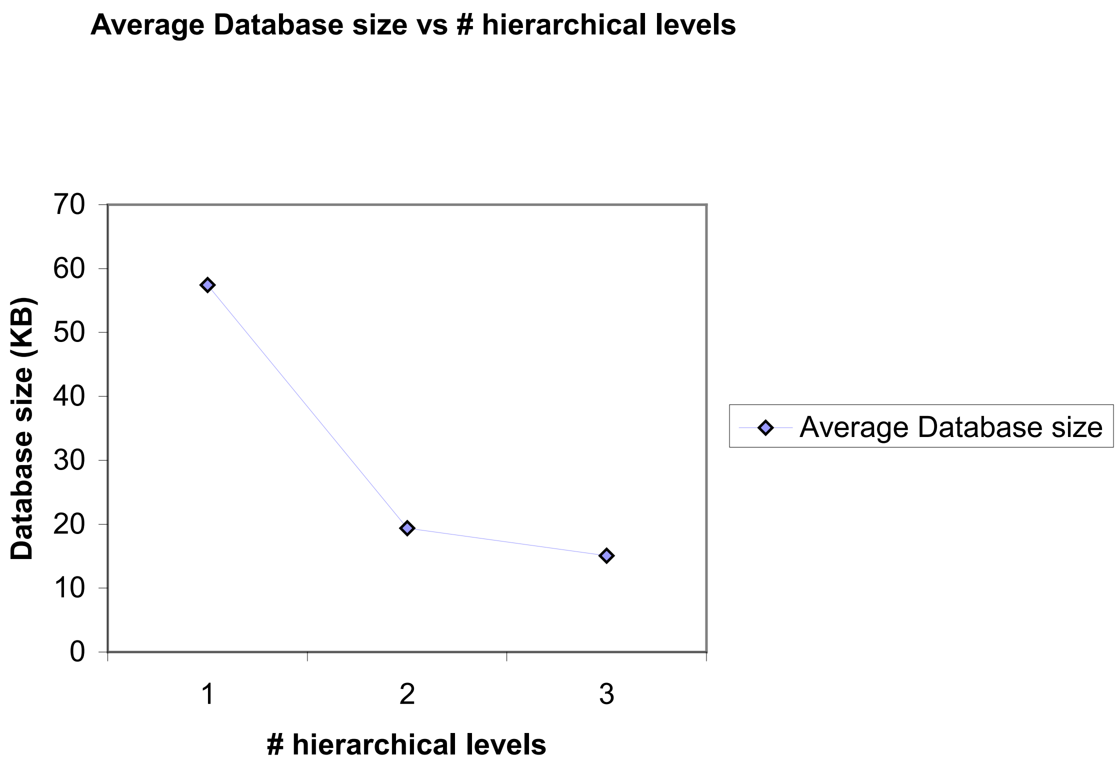

Fig 6.8 Average Database size vs. levels of hierarchy

The reduction in database size and the resulting inaccuracy in reachability information is actually *the* fundamental trade-off with hierarchical routing. As the topology scales in size, without hierarchical levels, each node needs to maintain state information about every other node in the topology and hence the database size increases to unmanageable proportions. With hierarchical levels, the database size is greatly reduced, but the trade-off for this reduction in size is inaccurate routing information caused as a result of abstraction of levels.

6.4.3 Mean Call setup success percentage

Figure 6.9 shows that the mean call setup percent decreases with an increase in the number of hierarchical levels.

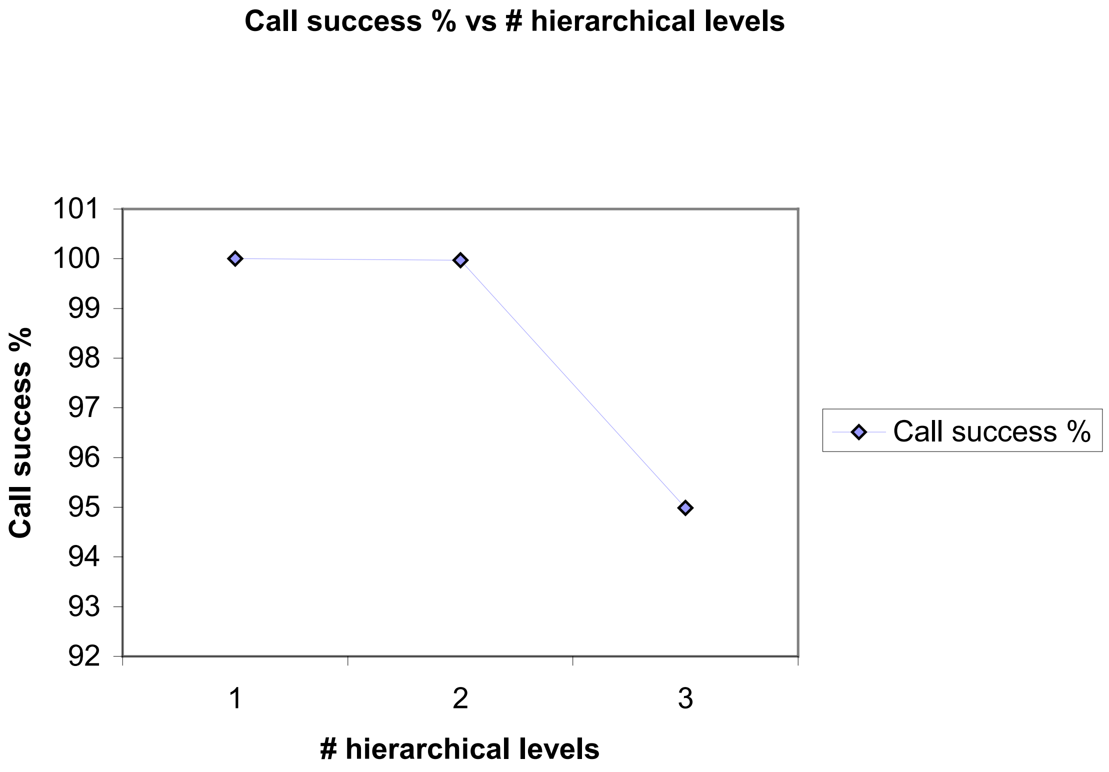

Fig 6.9 Mean call success rate vs. Levels of hierarchy

A call fails mainly because of routing inaccuracies and not because of unavailability of resources (computing and/or link bandwidth) when the number of hierarchical levels increases (for a given load and network topology). Routing inaccuracies are caused because of inaccurate and incomplete information present in the nodes' database as a result of topology abstraction.

6.4.4 Average number of hops

Figure 6.10 shows the variation of the average number of hops with respect to the variation in number of hierarchical levels. The average number of hops increases as the number of hierarchical levels increases. This is because, as the number of hierarchical levels increases, so do routing inaccuracies, and hence the path chosen for the call setup is not always the most optimal.

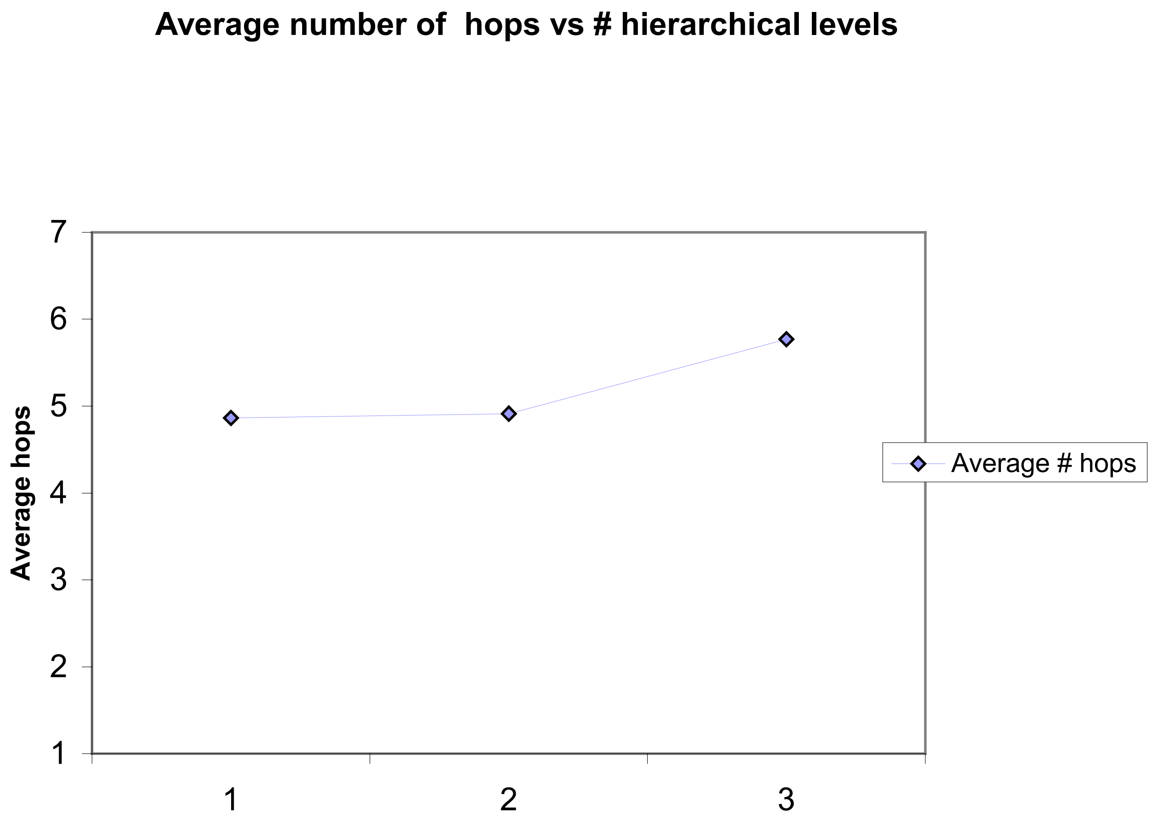

Fig 6.10 Average hops vs. Levels of hierarchy

The above explanation is further corroborated from a poorer call success percentage for an increased number of hierarchical levels, which is again due to inaccurate routing information in the database.

6.4.5 Convergence time

Figure 6.11 gives the variation of convergence time as a function of the number of hierarchical level. The convergence time decreases as the number of hierarchical levels increases.

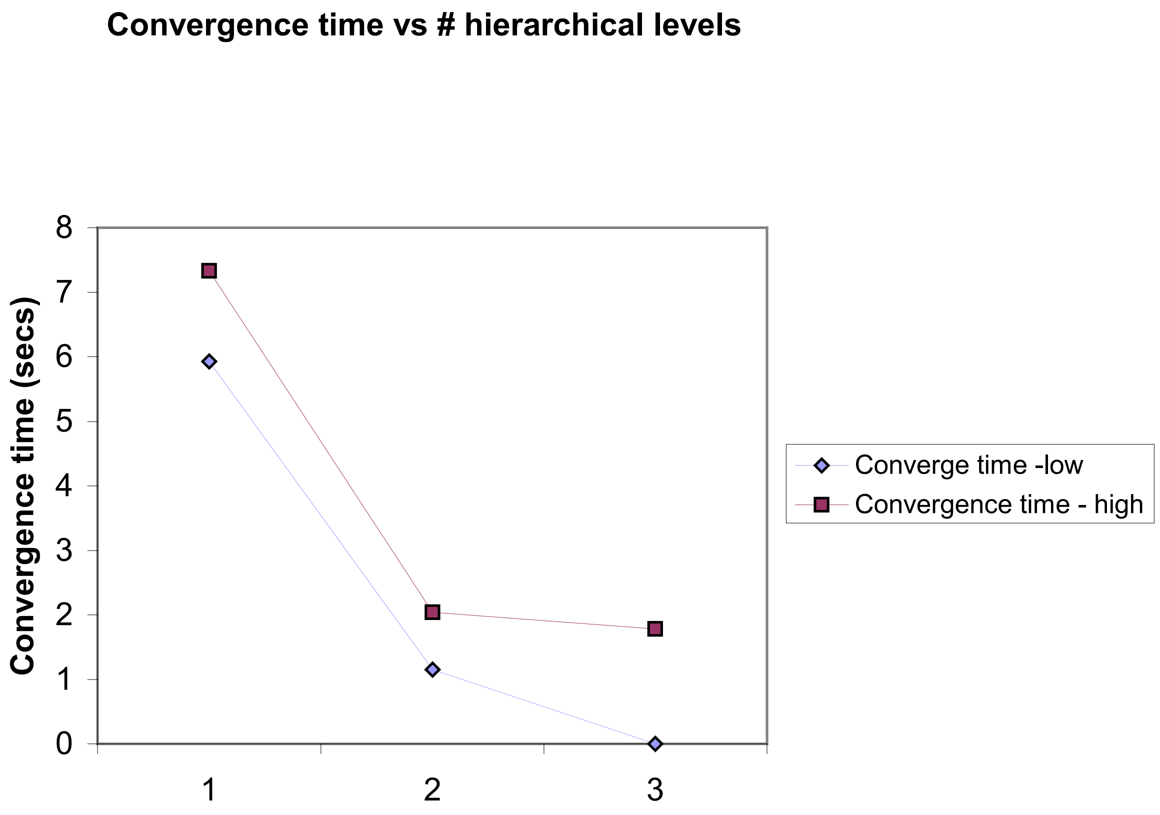

Fig 6.11 Convergence time vs. Levels of hierarchy

This is because the number of nodes that each node has to converge (in other words the convergence number) decreases as the number of hierarchical levels increases. This behavior was also seen in "Effect of Peergrouping" test.

6.4.6 Floods per node

Figure 6.12 shows that the number of floods per node increases as the hierarchical levels increases. This is because the PNNI control data is flooded to higher levels of hierarchy too; so more the number of hierarchical levels more are the number of floods.

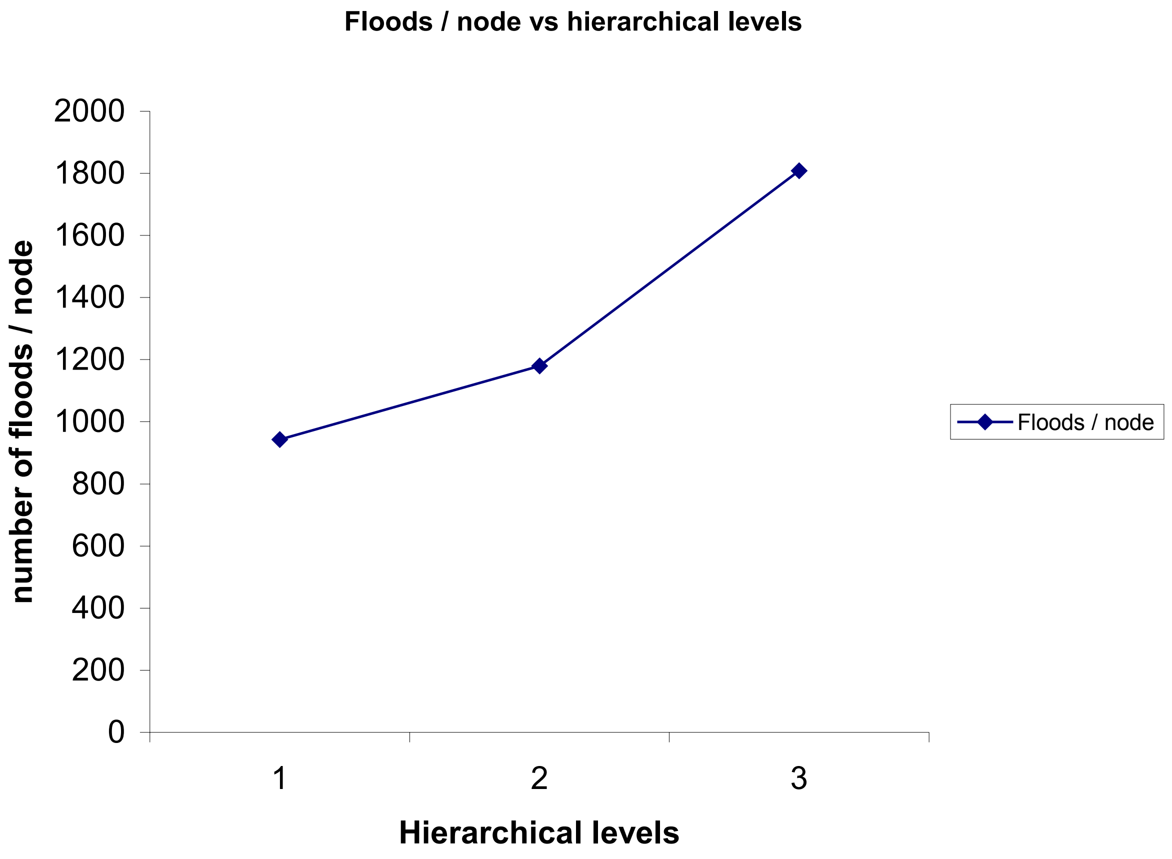

Fig 6.12 Floods per node vs. Levels of hierarchy

6.4.7 PNNI data per Flood

Figure 6.13 shows that the average PNNI data /flood is almost a constant. The topologies with more levels of hierarchy will have to flood PNNI data to its upper levels, too. Hence the number of floods in this case is higher but the PNNI data also increases proportionally. Hence the flat topology (without any hierarchical levels) has lesser number of floods and lesser PNNI data, but a similar PNNI data /flood measure as that of the topologies with higher levels of hierarchy.

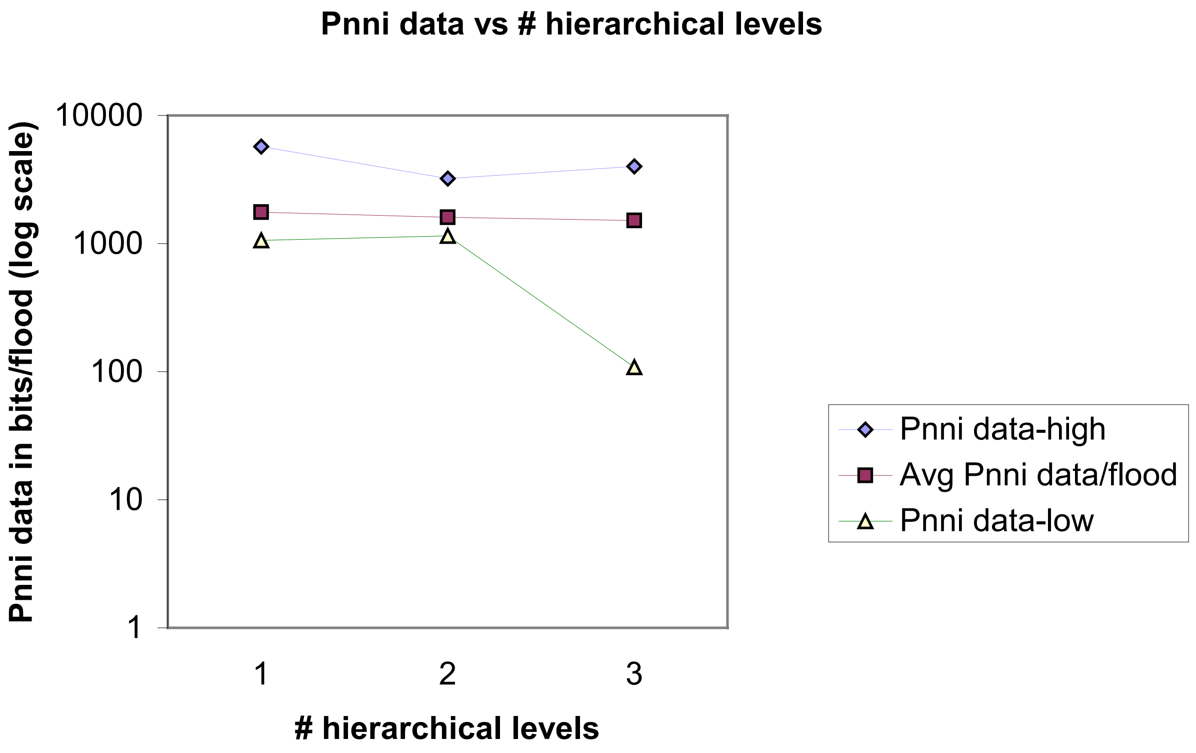

Fig 6.13 PNNI data per flood vs. Levels of hierarchy

6.4.8 Percentage Wasted Floods

Figure 6.15 shows the variation of percentage wasted floods with respect to variations in the number of hierarchical levels. The percentage wasted floods give a good measure of how robust the topology is to changes.

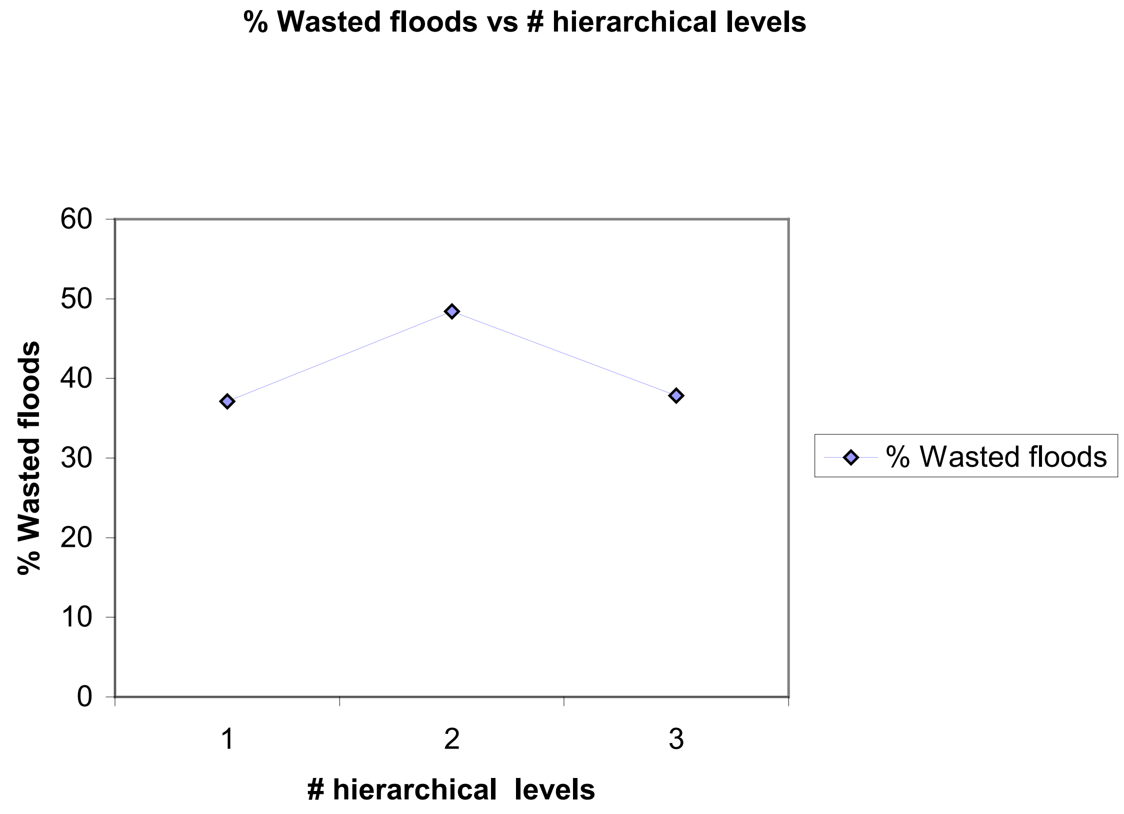

Fig 6.14 Wasted Floods percentage vs. Levels of hierarchy

The 2-level hierarchical topology has the maximum percentage of wasted floods. What the wasted floods indicate is that how often a significant change in one node has not affected another node's view of the entire topology. On the other hand, too much wasted floods is a waste of bandwidth. Hence, there is a trade-off between robustness (caused due to redundancy of data) and effective use of available bandwidth.

6.6 Conclusion

As the number of hierarchical levels increases, it can be concluded that

- Mean Call setup time increases
- Convergence time decreases
- Call success percentage decreases
- Average Database size decreases
- Average number of hops increases

Chapter 7

Effect of Crankback

7.1 Introduction

The aim of this test is to study the effect of varying the number of crankback retries on mean call setup time and call failure percentage, especially on the calls that fail due to lack of bandwidth, under fixed load and topology.

7.2 Test topology

The test topology is the 3-level hierarchical cluster topology with 96 nodes, as used in the "Effect of hierarchy" test. Refer to Fig 6.6 for the topology. The maximum number of crankback retries is varied as 0, 2, 5 and 8. The number of calls generated is 15 per host. Calls are generated with an exponential mean inter-arrival time of 40 seconds. A warm up ratio (similar to run in time) of 15% is used (the results of the first 15% of the calls generated by each host are ignored). The call bandwidth (Peak cell rate) of each call is changed from 64Kbps to 1250 Kbps.

7.3 Performance metrics

The metrics under consideration are

- Average Call failure percentage
- Mean Call setup time

7.4 Results and Analysis

7.4.1 Average Call failure percentage

Fig 7.1 gives the variation of call failure percentage with respect to the number of crankback retries. The average call failure percentage is expected to decrease as the number of crankback retries increases. But if the topology has not much scope for crankback i.e. if there is are not many routes to reach the destination, then crankback retries will only waste processing time, thus increasing the effective load on the system. This causes call failure percentage to increase due to processing time outs as the number of crankback retries increases.

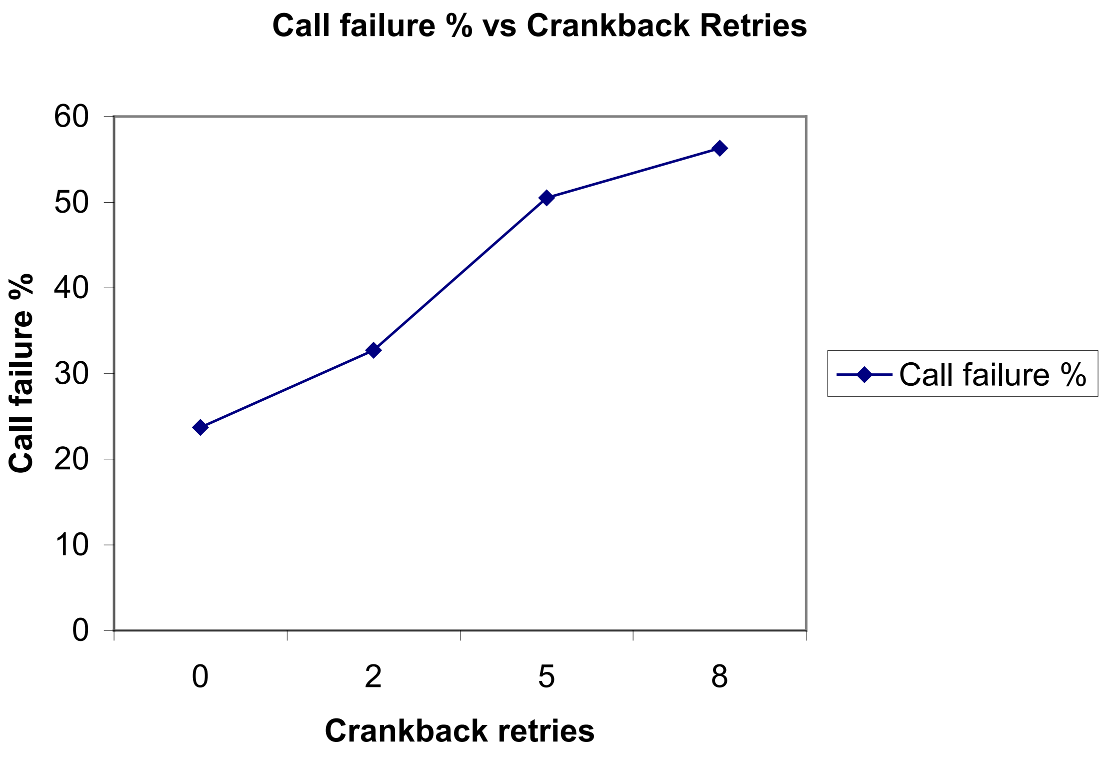

Fig 7.1 Call failure rate vs. Crankback retries

Figure 7.2 shows the variation of call failure rate (for calls failed due to lack of bandwidth) with a variation in the number of crankback retries. As expected, the call failure rate due to lack of bandwidth decreases initially with an increase in crankback retries, but eventually starts to increase for eight crankback retries. This behavior is seen because, with a large number of crankback retries, the effective load in the system is increased, and as a result a large number of calls are competing for bandwidth at the same time. Hence more calls fail because the effect of crankback is somewhat reduced at a higher effective load. In other words, higher crankback retries may indirectly be responsible for increase in load, and thus cause more call failures.

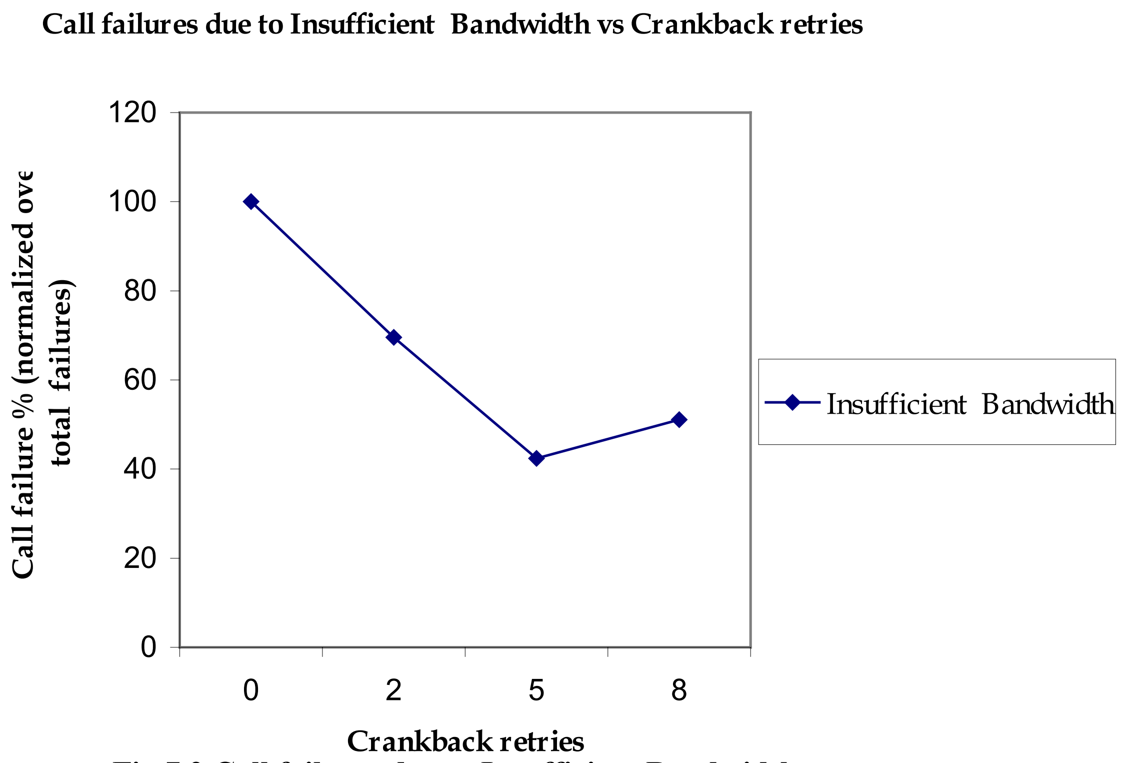

Fig 7.2 Call failures due to Insufficient Bandwidth vs. Crankback Retries

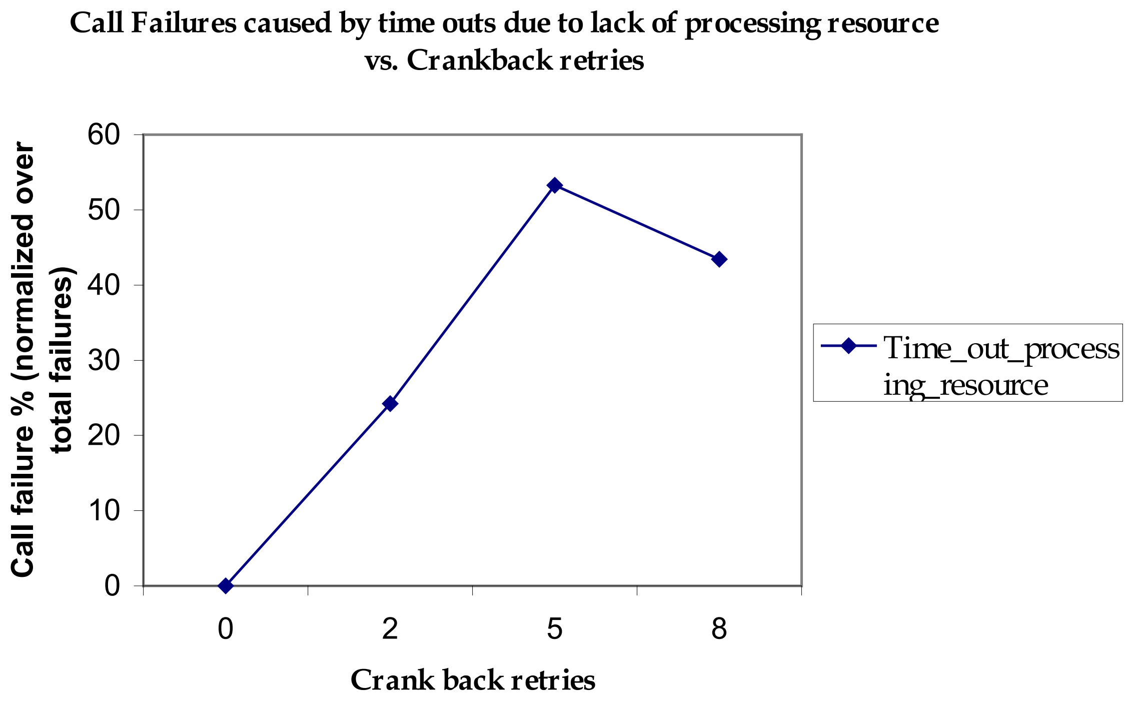

Fig 7.3 Call failures due to unavailability of processing resources vs. Crankback retries

Figure 7.3 shows the variation of call failure rate (for calls failed due to time outs caused by lack of processing resources) versus the variation in number of crankback retries. As seen above, crankback retries will increase effective load in the system. This causes an increased failure rate for calls that are timed out due to lack of availability of processing resources. Crankback will not help to alleviate the problem with lack of processing resources; in fact crankback will only serve to exacerbate the problem. For a maximum crankback retry number of 8, we see a lesser number of call failures due to time outs because more calls fail due to lack of bandwidth as well. Moreover, with 8 crankback retries, the effective load is very high, and many calls may fail at an earlier stage of the simulation itself, without progressing much into the network, and thus they reduce the number of failures due to processing time outs.

7.4.2 Mean Call Setup time

Figure 7.4 shows the variation of mean call setup time as a function of crankback retries. In general, more the number of crankback retries, more will be the call setup time.

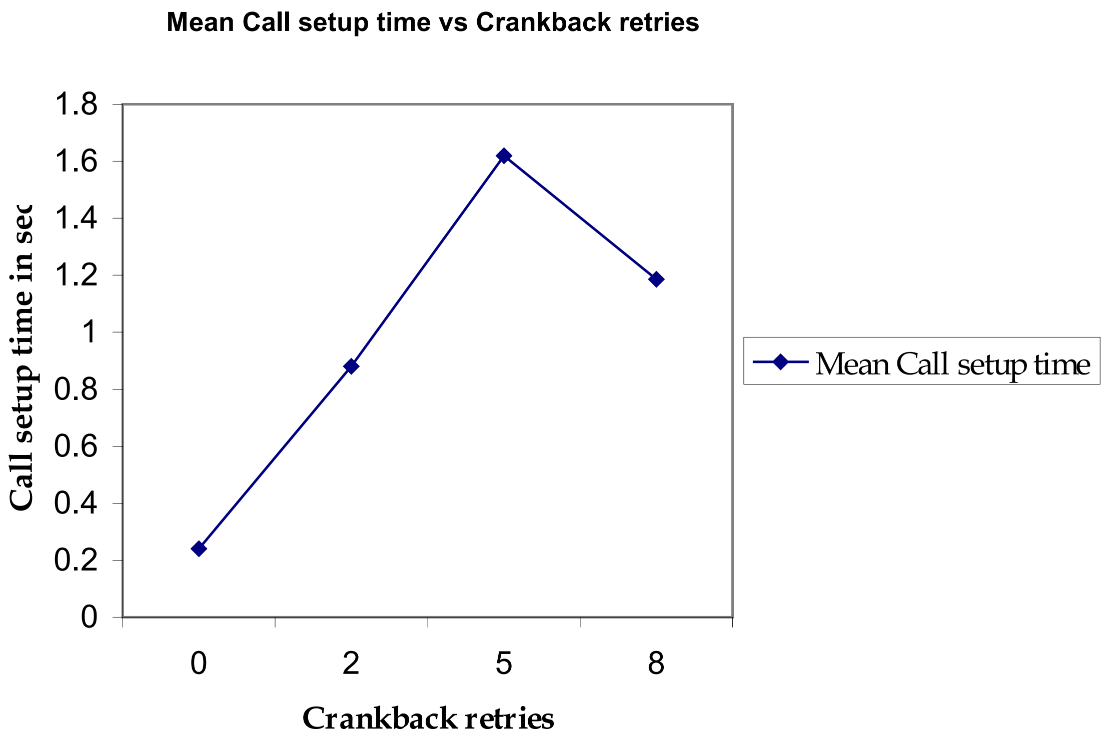

Fig 7.4 Mean Call setup time vs. Crankback retries

This is obvious because that crankback retries increase the effective load in the system, which causes an increase in the call queuing delay in the nodes. Hence, the calls experience a higher mean setup time. But, for a very high number of retries, many calls time out due to the excessive queuing delay, and the calls do not succeed. This causes a decrease in the call setup time; it also causes a decrease in the call setup percentage.

7.5 Conclusion

In a bandwidth-poor network, an increase in number of crankback retries

- Increases the call success percentage, until the call failure begins to be dominated by time outs associated with lack of processing resources. Even calls failing due to lack of bandwidth, if subjected to a large number of crankback retries, may eventually time out and fail, leading to decrease in call success percentage
- Increases call setup time irrespective of the type of network, until more calls fail due to lack of processing resources.

Hence it can be concluded that crankback retries are to be used only when a large percentage of calls fail due to lack of bandwidth (or some other QoS metric). In bandwidth-rich networks, the use of crankbacks will only decrease the call success rate. In a bandwidth-medium rich network, crankbacks could be used, but the number of retries should be chosen such that the crankbacks do not cause many call failures due to time outs before they could be processed. In a low-bandwidth network, it is advisable to use crankbacks, but again, the number of retries should be chosen so as not cause more failures associated with lack of processing power (due to crankbacks), than the number of calls that succeed due to crankbacks.

Chapter 8

Simulator Performance

8.1 Introduction

The scope of this set of simulations is to study the simulator's performance with change in load and size of the network – in other words to determine the effect of load and size of the network on the duration of the simulation.

8.2 Test topology

The test topologies used are the same as the ones used for the scalability and load tests. The simulations were run in each of the three topologies by varying the mean call inter-arrival time as 20 seconds, 10 seconds and 5 seconds. Each host generated the same number of calls as specified in the scalability test.

8.3 Parameter of interest

The duration of the simulation measured in real units of time is the parameter of interest. This gives a good idea of how the simulator behaves when load and or size of the network is increased.

8.4 Results and Analysis

8.4.1 Variation with change in size

The duration of simulation increases monotonically as the size of network is increased. This is obvious because the simulator will have to process more events generated as a result of increase in the number of calls (as a result of the number of nodes).

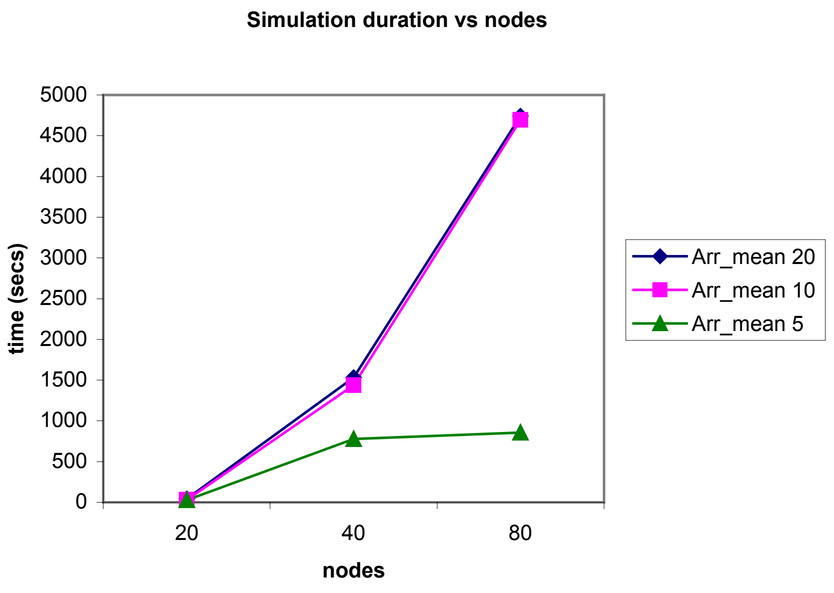

Fig 8.1 Variation of simulation time vs. size of network

8.4.2 Variation with change in load

The simulation time is expected to increase with increase in load, unless the heavy load causes calls to fail at an early stage. In such a case, the simulator runs out of events to process, and hence completes the simulation sooner.

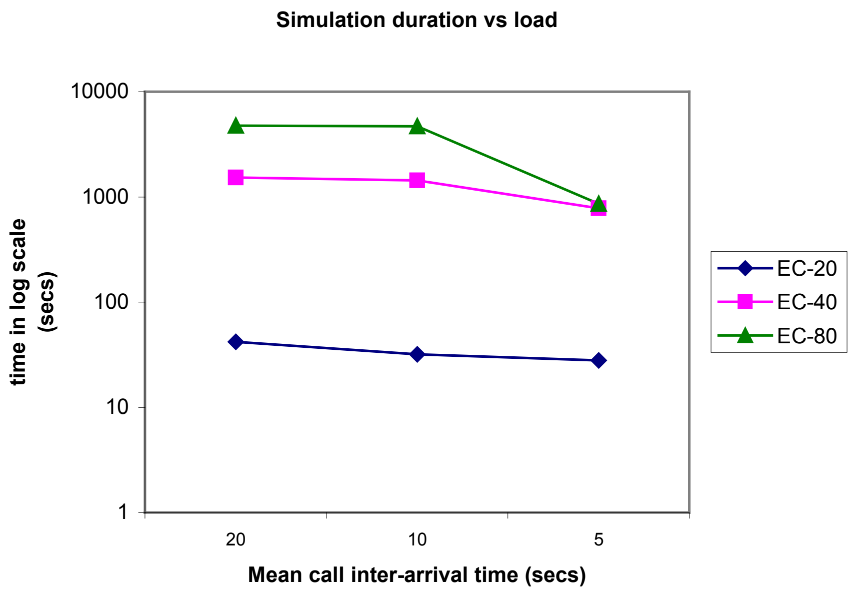

Fig 8.2 Variation of Simulation time vs. load

The results for variation of simulation time with changes in load (fig 8.2) are in direct contradiction with the hypothesis. For 20 nodes, changes in load hardly have any impact on simulation time. But for bigger topologies, simulation time decreases with increase in load. One possible explanation for this surprising behavior could be that the calls at high load in bigger topologies fail at an earlier stage, thus reducing the number of events the simulator has to process.

The call success rate in the case of 20 nodes is 100 percent for all loads. In case of 40 nodes, the call success rate is 100 percent for mean inter-arrival times of 20 seconds and 10 seconds, while it is 99 percent at heavy load. The duration of the simulation is almost the same at an inter-arrival mean of 20 seconds and 10 seconds, while the duration of simulation is lesser in case of a call inter-arrival mean of 5 seconds. One reason could be that, for the same call success rate, there tends to be more flooding (periodic floods, and hence more events) in case of lighter loads, since the simulated time is more.

The call success rate in the case of 80 nodes is 100 percent for mean inter-arrival times of 20 seconds and 10 seconds. But at heavy load, the call success rate is only 92 percent, which causes the duration of simulation to greatly reduce because of the reasons discussed in the preceding paragraphs.

8.4 Conclusion

As the size of the network is scaled upwards appropriately, the time to complete the simulation also increases.

As the load in the network is increased, the time to complete the simulation depends on the location of the failure of the call. If calls fail at the early stages of routing, then the simulation time might decrease for heavy loads; whereas if calls fail after progressing quite deep into the network, the simulation time might increase at heavy loads.

In the test topology, at 80 nodes, under heavy loads, the calls seem to fail at the early stages of routing itself.

Chapter 9 Summary

9.1 Summary of Conclusions

It can be thus concluded that:

- **As the topology is scaled in size in a bandwidth-rich network**,
  - Mean Call setup time increases
  - Call success percentage decreases
  - Average Database size increases
  - Convergence time increases
  - Total number of floods increases
  - Average PNNI data / flood increases
  - Average number of hops increases (in general, but depends on topology)

- **As the load in the network is increased in a bandwidth-rich network**,
  - Mean Call setup time increases
  - Call success percentage decreases
  - Average link utilization percentage increases
  - Average PNNI data / flood increases (only in large-sized networks)
  - Average number of hops remains unchanged (in general, but depends on topology)

- **As the number of peergroups in the network is increased in a band-width rich network,**
  - Convergence time decreases initially, and then increases
  - Mean Call setup time increases
  - Call success percentage decreases
  - Average Database size decreases
  - Average PNNI data / flood remains unchanged

- Average number of hops remains unchanged (in general, but depends on topology)

- **As the number of hierarchical levels increases in a bandwidth-rich network**,
  - Mean Call setup time increases
  - Convergence time decreases
  - Call success percentage decreases
  - Average Database size decreases
  - Average number of hops increases

For the test topology, it can be seen that we get the best performance in the 2-level hierarchy for almost all the performance metrics. Hence, for the chosen topology, the 2-level hierarchy is the most optimal.

- **As the number of crankback retries is increased in a bandwidth-poor network,**
  - Mean Call setup time increases and then decreases
  - Call success percentage increases initially and then decreases

- **As the number of nodes in the network increases,**
  - Simulation time increases

- **As the load in the network increases**
  - Simulation time depends on topology

.

Appendix - I

Simulation Parameters

The following simulation parameters are common for all simulations (except for the "Crankback Retries" test).

Node Parameters

ptse_age_offset	= uniform [-25 25]
fabric	= KU
numports	= 4
routing_policy	= max_bw
flooding_significance	= dynamic_threshold
flooding_threshold	= 2
prop_constant	= 25
default_flooding_period	= 1800 sec
default_flooding_factor	= 5
acac_policy	= call_packing
util_log_period	= 1
call_trace	= FALSE
queuesize	= 5000
crankback_retries	= 2
hello_timer	= 15
hello_inactivity_factor	= 4
summary_timer	= 20
ptsp_timer	= 20
ack_timer	= 5
request_timer	= 20
reaggregation_timer	= 0
process_time	= 2.0

Host Parameters

calls	=	25/30/50/60/120
arrival_distribution	=	exponential 20,15,10,5,2s
duration_distribution	=	exponential 100 sec
calltype1	=	cbr
pcr1	=	fixed 64 kbps
share1	=	5 %
ctd1	=	fixed 33 msec
cdv1	=	fixed 1 msec
clr1	=	fixed 8 (ie 10 ^ -8)
calltype2	=	vbr,
pcr2	=	fixed 64 kbps
pcr2scr2	=	fixed 2
share2	=	10 %
ctd2	=	fixed 43 msec
cdv2	=	fixed 11 msec
clr2	=	fixed 5

calltype3	=	vbr,
pcr3	=	fixed 64 kbps
pcr2scr3	=	fixed 1.78
share3	=	85 %
ctd3	=	fixed 43 msec
cdv3	=	fixed 11 msec
clr3	=	fixed 5

Scheduling info

duration	=	total # calls * mean call interarrival time
nodal_represent	=	complex
mpg	=	true

References

[1] *"KU-PNNI User's manual version 2.1"* by Phongsak Prasithsangaree, Kamalesh S Kalarickal, Gowri Dhandapani, Bhavani Shanmugam, Santosh Golecha, Pradeepkumar Mani, David W Petr and Douglas Niehaus, May 2002

[2] *"A Performance Evaluation Architecture for Hierarchical PNNI and Evaluation of Different Aggregation Algorithms in Large PNNI ATM Networks*" by Gowri Dhandapani, July 2000