Performance Evaluation of Multiple Criteria Routing Algorithms in Large PNNI ATM Network

Phongsak Prasithsangaree
Master’s Thesis Defense
May 22, 2000

What is PNNI?

- Private Network to Network Interface
  - A protocol for ATM networks
- PNNI is composed of two protocols
  - PNNI Routing Protocol
  - PNNI Signaling Protocol

Presentation Flow

- What is PNNI?
- Problem statement
- Our solution
- Implementation
- Performance metrics
- Test scenarios
- Performance evaluation
- Conclusions

PNNI Routing Protocol

- Hello Protocol
  - Hello Packets are exchanged between neighbor nodes
  - To discover and verify the identity of the neighbor nodes.
- Flooding Mechanism
  - A reliable hop-by-hop propagation of topology information.
  - Topology information, PTSE, and PTSP
  - PTSE is subject to aging and is removed after a pre-defined duration.
  - An updated PTSE is sent when topology information is significantly changed.
  - A significant change is determined by configuration parameters.
PNNI Routing Protocol (continued)

- PNNI Topology Metrics and Attributes
  - Metric: delay or administrative weight
  - Attribute: bandwidth, CLR, or CDV

- Routing Mechanism
  - Router gets a route request with requirements.
  - Router retrieves topology information from its database.
  - Router finds a possible path according to the requirements.
  - Return the path in the DTL format.

PNNI Signaling Protocol

- A subset of UNI 4.0 signaling standard.
- Call Setup Procedure
- Call Admission Control (CAC)
- Crankback and Alternate Routing

Problem Statement

- The popular routing algorithm used is Dijkstra’s algorithm, which can find a path based on a single cost.
- Need something better than Dijkstra’s Algorithm
- Multiple QoS Routing
  - A routing algorithm that can find a route with more than one constraint at the same time.
- However...
  - The problem of deciding if there is a path which satisfies more than one additive constraints is $NP$-complete.
Our Solution

- Heuristic Multiple Criteria Routing Algorithms

<table>
<thead>
<tr>
<th>Primary Criterion</th>
<th>Widest</th>
<th>Strict</th>
<th>Minimum Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widest</td>
<td>∘</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Strict</td>
<td>×</td>
<td>∘</td>
<td>×</td>
</tr>
<tr>
<td>Minimum Bit Rate</td>
<td>×</td>
<td>×</td>
<td>∘</td>
</tr>
</tbody>
</table>

○ Single QoS Routing Algorithm
× Multiple QoS Routing Algorithms

Implementation

- Generic Call Admission Control (GCAC)
  - Standardized by ATM forum to be used for call admission control (CAC).
  - CAC is vendor-specific.
  - GCAC is used to reduce the routing computational time.
  - It prunes links and nodes that cannot support the call.
Routing Algorithms

- Dijkstra’s algorithm has cost and distance as parameters.
- Widest Shortest Algorithm
  - Modified Dijkstra’s algorithms to consider two costs and two distances.
- D_widest algorithm
  - Modified relaxation method of Dijkstra’s algorithm
- Shortest Widest Algorithm (has two routing passes)
  - The first pass used D_widest algorithm and the second pass used the modified Dijkstra’s algorithm

Implementation (continued)

- Routing Computation Flow Chart

Performance Metrics
Performance Metrics

- Average Call Failure Rate
  \[ \frac{\text{Total number of rejected calls}}{\text{Total number of requested calls}} \]

- Average Call Setup Time
  \[ \frac{\text{Total call setup time}}{\text{Total number of successful calls}} \]

Performance Metrics (continued)

- Routing Inaccuracy
  \[ \frac{\text{Number of Crankback events}}{\text{Total number of call requests}} \]

- Link Utilization
  \[ \frac{\text{Used Link Bandwidth}}{\text{Link Bandwidth Capacity}} \]

Test Scenarios

- Edge-Core Networks

<table>
<thead>
<tr>
<th>Topology</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>2-core</td>
</tr>
<tr>
<td>Dense</td>
<td>2-core</td>
</tr>
</tbody>
</table>

Table 12: Link Metrics vs. Conventional Edge-Core Topologies

<table>
<thead>
<tr>
<th>Link</th>
<th>Capacity</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC-16</td>
<td>Unitree [5.3]</td>
<td></td>
</tr>
<tr>
<td>OC-32</td>
<td>Unitree [5.3]</td>
<td></td>
</tr>
<tr>
<td>OC-16</td>
<td>Unitree [5.3]</td>
<td></td>
</tr>
</tbody>
</table>

Table 13: Summary of Edge-Core Topologies
Dense Edge-Core Network

Light Edge-Core Network

Test Scenarios

<table>
<thead>
<tr>
<th>Cluster Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>5-cluster</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>6-cluster</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Metrics for Multiple Cluster Topologies

3-Cluster Network
Performance Evaluation

- Maximum bandwidth routing tests
- Minimum delay routing tests
- Link utilization tests
- Alternate routing tests
- Network density tests

Maximum Bandwidth Routing Tests

- Calls are CBR-typed with different bandwidth requests
- Call Arrival: 5 seconds between calls with Poisson distribution.
- Call duration: 60 seconds with Poisson distribution.
- Destination Hosts: uniformly selected from all other nodes.
- Total Calls: 2400 calls
Average Call Blocking Rate and Call Bandwidth

Average Call Setup Time and Call Bandwidth

Routing Inaccuracy and Call Bandwidth

Minimum Delay Routing Tests
Minimum Delay Routing Tests

- Calls are CBR-typed with different bandwidth requests
- Call Arrival: 5 seconds between calls with Poisson distribution.
- Call duration: varied with Poisson distribution.
- Destination Hosts: uniformly selected from all other nodes.
- Total Calls: 2400 calls

Average Call Blocking Rate and Call Holding Time

Average Call Failure Rate in Light Network (%)

Average Call Failure Rate in Dense Network (%)

Link Utilization Tests
Link Utilization Tests

- What is Link Utilization?
  
  \[
  \text{Link Capacity} = \frac{\text{Total BW used of the link}}{\text{Total BW of the link}}
  \]

- Calls are CBR-typed: an average of uniformly distributed call bandwidth: 10 Mbps.
- Call arrival: 5 seconds between calls with Poisson Distribution
- Call duration: 60 seconds with Poisson distribution
- One host makes 1000 calls
- Total 24,000 calls in the network.

Link Utilization in Edge-Core Network

- Shortest-minhop

- Widest-minhop
Alternate Routing Tests

- Calls are CBR-typed: an average of uniformly distributed call bandwidth: 30 Mbps.
- Call arrival: 5 seconds between calls with Poisson distribution.
- Call duration: 60 seconds with Poisson distribution
- Total number of calls: 2400 calls
- We increase the number of alternate routing retries.

Alternate Routing in Cluster Network

- Average call failure rate when using widest group algorithm

Network Core Density Tests
Network Core Density Tests

- Calls are CBR-typed: an average of uniformly distributed call bandwidth: 15 Mbps.
- Call arrivals: 5 seconds between calls with Poisson distribution.
- Call duration: 60 seconds with Poisson distribution.
- Total calls: 2400 calls
- Network density (or connectivity)

<table>
<thead>
<tr>
<th>Links</th>
<th>Low-dense</th>
<th>Medium-dense</th>
<th>High-dense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Links</td>
<td>18</td>
<td>27</td>
<td>36</td>
</tr>
<tr>
<td>Edge Links</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Nodes</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Connectivity</td>
<td>1.75</td>
<td>2.125</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Routing with Different Network Core Density

Conclusions

- In maximum bandwidth routing, the widest-minhop and the shortest-widest-minhop routing algorithms tend to perform better than others in the widest algorithm group.
- However, the minhop-widest and the shortest-widest routing algorithms tend to perform worse than others in the widest algorithm group.
- In minimum delay routing, those algorithms do not perform well because they do not consider the dynamic change of the network.
Conclusions (continued)

- Widest-minhop routing algorithm can improve the link utilization of the network.
- Increasing the number of alternate routing retries slightly improves the call success rate.
- Increasing the number of the core links in the edge-core network improves the call success rate, BUT not always.
- At a certain point, increasing the network density does not reduce the call failure rate. Instead, it increases the call setup time.
- The large amount of resource information can deteriorate the network performance.