Enabling Conferencing Applications using an Overlay Multicast Architecture

Authors: Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zang @ CMU
Motivation

Alternative to active networks

- Goal of active networks: Provide network intelligence to data traveling over network.
- Is network intelligence required ??
Why conferencing as a test case

- Time critical
- Bandwidth
- Changing topology
Architectural Alternatives

- Multiple unicasts
- IP Multicast
- Active Networks based Multicast
- End System Multicast
Multiple unicast

The dumb choice
IP Multicast

Specific intelligence in router

A → R1 → R2 → C
B → R1 → R2 → D
Active Networks based Multicast

Really intelligent router
End System Multicast

The dumb network
Smart end points
## Conferencing Requirements v.s. End System Multicast Features

<table>
<thead>
<tr>
<th><strong>Conferencing</strong></th>
<th><strong>End system multicast features</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Lasts for longer periods compared to other network apps</td>
<td>- Overlays get better with time</td>
</tr>
<tr>
<td>- Small group sizes</td>
<td>- Allows end systems to interact among themselves</td>
</tr>
</tbody>
</table>

But what about performance??
Ideal end system overlay network

- Small latency
- Optimal bandwidth usage

Least Bandwidth

Efficient overlay

Least Latency
Issues Addressed

- Adapt to changes in group membership
- Adapt to changes in network conditions (and get knowledge of network)
Narada: End System Multicast

1. Create a mesh
   (Might have cycles.)
2. Optimize mesh to get good spanning tree at each node
   - Each node sends its routing table periodically
   - Nodes perform measurements
Heuristics

To make decision about a link (add/drop from tree)

1. Path bandwidths assigned to levels
2. At same level, lower latency link is used
3. To avoid oscillation among different links
   1. Base decisions on smoothed out readings (rather than just one)
   2. Go to lower level immediately but raise bandwidth level only if significantly different from current estimate.
4. Over estimate cost of dropping link
Experimental Evaluation

Goals

- Is ESM suitable for conferencing
  - Measurement of critical parameters (bandwidth and latency)
- Metrics for constructing overlays
  - Use different metrics see which delivers best values for parameters
- Resource usage
  - Estimate based on propagation delay for all links used
- Overheads of protocol
  - Measurement of control and probe traffic
Experimental methodology

- Networking experiments on real test beds are prone to changes in network conditions.
  - Carry out multiple times. Average results
  - Interleave different schemes
  - For different network loads, carryout on different times of the day
  - Consider heterogeneous host set
  - See the big picture
Parameters

• Different metrics: Sequential unicast, Bandwidth latency, latency only, bandwidth only, propagation delay only
• Different stream sizes 1.2 Mbps, 2.4 Mbps
• Different host sets: Primary and Extended
Primary set with 1.2Mbps (b/w)

Internet pathology

Graph from SIGCOMM presentation
Extended set with 2.4 Mbps (b/w)

Graph from SIGCOMM presentation

Latency based scheme gives poor b/w
Extended set with 2.4 Mbps (b/w)

Graph from SIGCOMM presentation

b/w based scheme gives poor latency
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Primary 1.2</th>
<th>Extended 2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>2.62</td>
<td>1.83</td>
</tr>
<tr>
<td>Random</td>
<td>2.24</td>
<td>1.97</td>
</tr>
<tr>
<td>B/w only</td>
<td>1.85</td>
<td>1.51</td>
</tr>
<tr>
<td>B/w latency</td>
<td>1.49</td>
<td>1.31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Primary 1.2</th>
<th>Extended 2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Overhead</td>
<td>10.79</td>
<td>14.20</td>
</tr>
<tr>
<td>B/w probes</td>
<td>92.24</td>
<td>94.30</td>
</tr>
<tr>
<td>Other</td>
<td>7.76</td>
<td>5.70</td>
</tr>
</tbody>
</table>
Adaptation to network congestion
Conclusions

- End systems approach to multicast is pretty good.
- Can be easily deployed
- Active networks still easier way to provide multicast functionality and customization for network conditions.