Application Layer Multicast

RVS Seminar HS08, Oct 22\textsuperscript{nd}, 2008

Marc Brogle
Overview


> Multicasting
  — Unicast vs. Multicast and Link costs

> IP Multicast vs. Application Layer Multicast
  — Trees among routers vs. among end systems
  — Efficiency of IP Multicast vs. ALM
  — Deployment issues

> ALM protocol design
  — Application domain
  — Deployment level
  — Group management
  — Routing mechanisms

> Comparison & Conclusion
Multicasting

> Multicast is more efficient than multiple unicast connections

> Multicast example scenario considering link costs (RTT, $, ...)

topology with costs

multicast tree

[1]
IP Multicast vs. Application Layer Multicast: Trees among Routers vs. among End Systems

> IP Multicast [2] optimal regarding tree structure (routers in tree)
> ALM has overhead due to tree built among end systems

all figures from [1]
IP Multicast vs. Application Layer Multicast: Efficiency of IP Multicast vs. ALM

- IP Multicast is efficient but needs deployment of routers
- ALM hosts have little information about underlaying network
- ALM tree building can be optimized (link / tree stretch) to incur only low penalties compared to IP Multicast [3]

**topology**

**IP Multicast tree**

**ALM tree**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>20</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**total costs = 37**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>20</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**total costs = 39**

**all figures from [1]**
IP Multicast vs. Application Layer Multicast: Deployment issues

> No widespread deployment of IP Multicast in the Internet

> Technical, administrative and business related issues [4]
  — IP Multicast capable routers at all levels of network required
  — Tendency to install simple, unintelligent (= very fast) routers
  — Managing and security issues (flooding attacks)
  — Billing and charging

> MBONE [5] project (mid 90’s)
  — Unicast connections between (IP Multicast) subnetworks
  — IP tunneling between these “IP Multicast islands”
  — Problems: receiver authentication, group management, flooding
  — Static setup of unicast tunnels = growth problem
  — Not available for home Internet users through their ISPs
Protocol design depends on the application domain [4]

- Audio/video streaming
  - single source
  - large number of receivers
- Audio/video conferencing
  - small to medium group size
  - interactive multiparty conferencing session
  - multiple sources
- Generic multicast service
  - based on specific metrics (delay, bandwidth, fan-out, ...)
- Reliable data broadcast and file transfer
  - large data sets (distributed DB, file sharing)
  - bandwidth as only metric

Typically focus on optimizing for a single application domain
Proxy-based (infrastructure-level) ALM:
- requires dedicated server/proxies in the Internet
- creates overlay only among proxies
- provides a transparent multicast service to end-users (IP Multicast)
- is typically generic multicast service
- may expect a service charge

End-system ALM:
- assumes only unicast infrastructure
- expects users (end-systems) to take part in the forwarding
- is “free” as of peer-to-peer nature (independent and cost-free)
- enjoys more flexibility, optimized for specific application domains
ALM Protocol Design
Group Management

> Key decisions regarding group / node management
  — How to find out about / join / leave groups?
  — Sending allowed when not joined?
  — Centralized or decentralized management?
  — Support existing IP Multicast Islands?
  — Support refinement during group life-time?
  — Use mesh-first or tree-first approach?

> Typically ALM use
  — Rendez-vous points for discovery
  — Source-specific trees for video streaming 1:n
  — Mesh-first constructed shared trees for conferencing
ALM Protocol Design
Group Management

> Mesh-first
  — builds P2P “mesh” without the multicast tree in mind
  — limits multicast tree quality (depends on quality of the mesh)
  — is more robust and better for multi-source applications

> Tree-first
  — builds the multicast tree directly without any mesh
  — gives direct control over the tree (e.g. control fan-out)
  — changes cause change for all descendants in tree
  — has lower communication overhead (simpler)

> Source-specific trees vs. shared trees
  — Two conflicting design goals:
    - minimize individual path length (hops/end-to-end delay)
    - minimize $\sum$ hops (cumulative end-to-end delay) to all destinations
ALM Protocol Design
Group Management

> Source-specific trees
  — optimizes the tree for a single source
  — has a limited efficiency for multiple sources on same tree

> Shared trees
  — supports efficiently multiparty-communications
  — has better maintenance costs than source-specific trees

> Distributed vs. centralized (balance simplicity vs. robustness)
  — Distribute workload for tree maintenance among root nodes
    (robust, synchronization issues, large-scale applications)
  — Central group management for small-scale applications
    (single-point of failure, simple & easy deployment)

> Refinement: optimize tree performance because of new joins and leaves (can cause interruptions & stability issues)
ALM Protocol Design
Routing Mechanism

> Shortest path trees: use RTT measurements to build the shortest path tree from source to end hosts

> Minimum spanning trees: construct “low cost” trees

> Clustering structures: build hierarchical clusters

> Peer-to-peer structures: typically use reverse-path forwarding
## Comparison & Conclusion

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Routing</strong></td>
<td>Peer-to-peer</td>
<td>clustering</td>
<td>Peer-to-peer</td>
<td>Peer-to-peer</td>
</tr>
<tr>
<td></td>
<td>structure</td>
<td>structure</td>
<td>structure</td>
<td>structure</td>
</tr>
<tr>
<td><strong>Refinement</strong></td>
<td>Yes (hop delay)</td>
<td>Periodically</td>
<td>No</td>
<td>Periodically</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(reclustering)</td>
<td></td>
<td>(fix fingers)</td>
</tr>
<tr>
<td><strong>Out-degree</strong></td>
<td>typical bounds</td>
<td>fixed bounds</td>
<td>none (~N, ~d, ~t)</td>
<td>typical bounds</td>
</tr>
<tr>
<td><strong>constrains</strong></td>
<td>(~N)</td>
<td></td>
<td></td>
<td>(~N)</td>
</tr>
<tr>
<td><strong>Application</strong></td>
<td>generic multicast</td>
<td>“single source”</td>
<td>“generic multicast”</td>
<td>“generic multicast”</td>
</tr>
<tr>
<td><strong>domain</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tree-first /</strong></td>
<td>mesh-first</td>
<td>“mesh-first”</td>
<td>mesh (duplicates)</td>
<td>mesh-first</td>
</tr>
<tr>
<td><strong>mesh-first</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> common: distributed, hop delay metric, source-specific trees
References


References


Questions

...and so, in conclusion, the proposed method...

THANK GOODNESS, ALMOST OVER... HOPEFULLY I DIDN'T BORE THEM TO TEARS.

...thank you, you've been a great audience...

OK, THE OBLIGATORY CALL FOR QUESTIONS AND I AM DONE...

...tions?

OR NOT.

www.phdcomics.com