A Case For End System Multicast

Yang-hua Chu, Sanjay Rao and Hui Zhang
Carnegie Mellon University

IP Multicast

•No duplicate packets
•Highly efficient bandwidth usage

Key Architectural Decision: Add support for multicast in IP layer

Key Concerns with IP Multicast

• Scalability with number of groups
  - Routers maintain per-group state
  - Analogous to per-flow state for QoS guarantees
  - Aggregation of multicast addresses is complicated
• Supporting higher level functionality is difficult
  - IP Multicast: best-effort multi-point delivery service
  - End systems responsible for handling higher level functionality
  - Reliability and congestion control for IP Multicast complicated
• Deployment is difficult and slow
  - ISP’s reluctant to turn on IP Multicast

The billion dollar question...

• Can we achieve efficient multi-point delivery, without support from the IP layer?

End System Multicast

Overlay Tree

CMU
Gatech
Stanford
Stan1
Stan2
Berk1
Berk2
Berk1
Berk2
Potential Benefits

- Scalability
  - Routers do not maintain per-group state
  - End systems do, but they participate in very few groups
- Easier to deploy
- Potentially simplifies support for higher level functionality
  - Leverage computation and storage of end systems
  - For example, for buffering packets, transcoding, ACK aggregation
  - Leverage solutions for unicast congestion control and reliability

What I hope to convince you of ...

- End System Multicast is a promising alternative approach for multi-point delivery
  - Narada: A distributed protocol for constructing efficient overlay trees among end systems
  - Simulation and Internet evaluation results to demonstrate that Narada can achieve good performance
- Consider applications with small and sparse groups
  - Around tens to hundreds of members

What is an efficient overlay tree?

- The delay between the source and receivers is small
- Ideally,
  - The number of redundant packets on any physical link is low
- Heuristic we use:
  - Every member in the tree has a small degree
  - Degree chosen to reflect bandwidth of connection to Internet

Why is self-organization hard?

- Dynamic changes in group membership
  - Members may join and leave dynamically
  - Members may die
- Limited knowledge of network conditions
  - Members do not know delay to each other when they join
  - Members probe each other to learn network related information
  - Overlay must self-improve as more information available
- Dynamic changes in network conditions
  - Delay between members may vary over time due to congestion

Narada Design

Step 1

- "Mesh": Richer overlay that may have cycles and includes all group members
  - Members have low degrees
  - Shortest path delay between any pair of members along mesh is small

Step 2

- Source rooted shortest delay spanning trees of mesh
- Constructed using well known routing algorithms
  - Members have low degrees
  - Small delay from source to receivers
Narada Components

- Mesh Management:
  - Ensures mesh remains connected in face of membership changes
- Mesh Optimization:
  - Distributed heuristics for ensuring shortest path delay between members along the mesh is small
- Spanning tree construction:
  - Routing algorithms for constructing data-delivery trees
  - Distance vector routing, and reverse path forwarding

Optimizing Mesh Quality

- Members periodically probe other members at random
- New Link added if Utility Gain of adding link > Add Threshold
- Members periodically monitor existing links
- Existing Link dropped if Cost of dropping link < Drop Threshold

The terms defined

- Utility gain of adding a link based on
  - The number of members to which routing delay improves
  - How significant the improvement in delay to each member is
- Cost of dropping a link based on
  - The number of members to which routing delay increases, for either neighbor
- Add/Drop Thresholds are functions of:
  - Member’s estimation of group size
  - Current and maximum degree of member in the mesh

Desirable properties of heuristics

- Stability: A dropped link will not be immediately readded
- Partition Avoidance: A partition of the mesh is unlikely to be caused as a result of any single link being dropped

Narada Evaluation

- Simulation experiments
- Evaluation of an implementation on the Internet
Performance Metrics

- Delay between members using Narada
- Stress, defined as the number of identical copies of a packet that traverse a physical link

Factors affecting performance

- Topology Model
  - Waxman Variant
  - Mapnet: Connectivity modeled after several ISP backbones
  - ASMap: Based on inter-domain Internet connectivity
- Topology Size
  - Between 64 and 1024 routers
- Group Size
  - Between 16 and 256
- Fanout range
  - Number of neighbors each member tries to maintain in the mesh

Simulation Details

- Simulator
  - Packet-level and event-based
  - Models propagation delay of physical links
  - Does not model queuing delay and packet loss
- Individual Experiment Description
  - All group members join in random sequence in first 100 seconds
  - No change in group membership after 100 seconds
  - One sender picked at random and multicasts data at constant rate

Delay in typical run

- Waxman model: 1024 routers, 3145 links
- Group Size: 128
- Fanout Range: <3-6> for all members

Stress in typical run

- Native Multicast
- Narada: 14-fold reduction in worst-case stress!
- Naive Unicast

Variation with group size

- Waxman model: 1024 routers, 3145 links
- Fanout Range: <3-6>
Variation with topology model

- Waxman
- ASMap
- Mapnet

Implementation Status
- Implemented and ported to Linux and Sun
- Available as library that can be compiled with applications
- Examining how applications written with IP Multicast API can be used without source-code modification

Internet Evaluation
- 13 hosts, all join the group at about the same time
- No further change in group membership
- Each member tries to maintain 2-4 neighbors in the mesh
- Host at CMU designated source

Narada Delay Vs. Unicast Delay
- Internet Routing can be sub-optimal

Related Work
- Yoid (Paul Francis, ACIRI)
  - More emphasis on architectural aspects, less on performance
  - Uses a shared tree among participating members
  - More susceptible to a central point of failure
  - Distributed heuristics for managing and optimizing a tree are more complicated as cycles must be avoided
- Scattercast (Chawathe et al, UC Berkeley)
  - Emphasis on infrastructural support and proxy-based multicast
  - To us, an end system includes the notion of proxies
  - Also uses a mesh, but differences in protocol details

Conclusions
- Proposed in 1989, IP Multicast is not yet widely deployed
  - Per-group state, control state complexity and scaling concerns
  - Difficult to support higher layer functionality
  - Difficult to deploy, and get ISP’s to turn on IP Multicast
- Is IP the right layer for supporting multicast functionality?
- For small-sized groups, an end-system overlay approach
  - is feasible
  - has a low performance penalty compared to IP Multicast
  - has the potential to simplify support for higher layer functionality
  - allows for application-specific customizations