Application Layer Multicast

Instructor: Hamid R. Rabiee

Spring 2013
Outline

✧ Introduction
  ✧ IP Multicast vs. Application-Layer Multicast
  ✧ Limitations of IP Multicast
  ✧ Deployment level in ALM

✧ Multicast Tree Formation
  ✧ Tree-first approach
  ✧ Mesh-first approach
  ✧ Hybrid approach
    ✧ LayeredCast
    ✧ P2P Applications

✧ Routing mechanism in ALM
✧ Control operation in ALM
Multicast – Overlay Networks & Video Streaming

- Multiple Unicast
- IP Multicast
- Application Layer Multicast (ALM)
  - Content Distribution Networks (CDN)
  - Overlay Multicast
Limitation of IP Multicast

✧ Complexity and overhead at routers
  ✧ The routing and forwarding table at the routers need to maintain an entry corresponding to each unique multicast group address.
  ✧ Unlike unicast addresses, these multicast group addresses are not easily aggregatable.
  ✧ Requires routers to maintain per-group state; violates the stateless principle of the router construction

✧ Supporting higher level functionality is difficult
  ✧ IP multicast provides (best-effort) multi-point delivery service
  ✧ Reliability and congestion control for IP multicast is complicated
Limitation of IP Multicast (cont.)

- Extremely difficult to deploy efficiently on many research groups, companies, and Internet service providers (ISP) at a large scale

- Security issues
  - Vulnerable to flooding attacks without complex network management
  - Unauthorized reception of data from a multicast session
  - Preventing allocation of same multicast address for two sessions
  - The difficulty of setting up firewalls while allowing multicasting

Provide IP multicast functionality above the IP layer -> Application Layer Multicast
Application Layer Multicast (ALM)

- Application-layer (or end-system) multicast
  - End systems communicate through an overlay structure
  - Assuming only unicast paths provided by underlying network

Figure 1 - Comparing ALM with IP multicast

(a) A sample network
(b) Data distribution through IP Multicast
(c) Data distribution through ALM
Application Layer Multicast (ALM) (cont.)

- In ALM end-hosts are responsible for
  - *Group membership*
  - *Multicast delivery structure construction*
  - *Data forwarding*

- No requirement for the support of routers

- Joining the network:
  - *New members find out about the topology from a common bootstrap point called a Rendezvous Point (RP) or Landmark Point (LP)*
  - *Find the best path for exchanging data to a subset of members already part of the topology*
    - *Important to have a cost-aware, efficient, and scalable topology with minimum delay and low control overhead*
    - *Join the topology by exchanging control messages with the members in an application-specific manner*
Application Layer Multicast (ALM) (cont.)

✧ Advantages

✧ No need to change routers
✧ Allow features to be easily incorporated
✧ Immediate deployment on the Internet
✧ Easier maintenance and update of the algorithm
✧ The ability to adapt to a specific application

✧ Disadvantages

✧ End-hosts in ALM has little or no knowledge about the underlying network topology, thus it results in performance penalty in term of
  ✧ Less efficient network usage
  ✧ Longer end-to-end latency
1. Infrastructure level (or proxy-based ALM protocols)

- Requires the deployment of dedicated servers/proxies on the Internet which provides a transparent multicast service to the end-user

- **Advantages**
  - High efficiency: represent IP multicast groups as an overlay node
  - Greater bandwidth availability to the proxy nodes (compared with end-hosts)
  - Longer life cycle of overlay nodes (compared with end-hosts)
  - Relieve end-hosts from any forwarding responsibility => multicast is transparently made available to end-hosts => reduce application complexity

- **Disadvantages**
  - Incurring the cost for deployment proxies in the inter-network
  - Less adaptable and less organized for specific applications

Figure 2- A sample proxy-based ALM network
2. **End system level**

- Assume a unicast service from the infrastructure and expect end-hosts to participate in providing the multicasting functionality

**Advantages**
- Has more flexibility and adaptability to specific application domains
- Immediate deployment over the Internet
- No need for changes to IP or routers
- No need for ISP cooperation
- End hosts can prevent other hosts from sending
- Easy to implement reliability: use hop-by-hop retransmissions

**Disadvantages**
- Must deal with limited bandwidth of end systems
- Require end-hosts to take on some of the forwarding responsibility
- Increase application software development complexity
Group Management in ALM

✧ Responsibilities of a group manager

✧ Whether a mesh-first, a tree-first, or a hybrid approach is taken?

✧ How they join or leave a session?

✧ Whether the management is done in centralized or in distributed way?

✧ Which design is taken; minimizing the length of the path (source-specific tree) or minimizing the total number of hops (shared-tree)?
Distributed Hash Tables (DHT)

Operations:
- insert(k,v)
- lookup(k)

P2P overlay maps keys to nodes
- completely decentralized and self-organizing
- robust, scalable

k1,v1, k2,v2, k3,v3
k4,v4
k5,v5, k6,v6
Structure of a P2P Video Streaming Protocol

✧ Four basic category
  ✧ Topology
  ✧ Send & Receive Data
  ✧ Incentive
  ✧ Group Management

✧ Characteristics of a P2P overlay
  ✧ Distribution
  ✧ Decentralized control
  ✧ Self-organization

Figure 4- structure of P2P video streaming protocol
## Design Choices

1. **Topology**
   - Tree (*Push-Based*)
   - Mesh (*Pull-Based*) (*Data Driven*)
   - Hybrid
     - Separated Data/Control Overlays
     - Compensatory Overlays
     - Multiple Primary Data Delivery Overlays

2. **Video Codec**
   - Single Layer
   - Scalable Video Codec
     - Multi-description Video
     - Layered Video
       - **SVC**: Scalable Video Coding
       - **FGS**: Fine Granularity Scalability
Problem Definition

 Tree-based
 Mesh-based

 Problems
  ✉ Heterogeneous bandwidth
  ✉ Reliability and fairness in Tree-based protocols
  ✉ Delay in Mesh-based protocols
Common issues
- Organize, maintain overlay network
  - Node arrivals
  - Node Failures
- Resource allocation
- Balancing
- Resource location
- Network proximity routing

Idea: provide a generic P2P substrate
Mesh, Tree, and Hybrid Approaches

Tree Approach
- The tree is built directly without any mesh
- The members’ parent are selected from the known members in tree
  - Require running an algorithm to detect and avoid loops and to ensure the structure is a tree.
- Direct control over the tree to
  - maintain strict control over the fan-out
  - select a best parent neighbor that has enough resources
  - respond to the failed members with a minimum impact to the tree
- Sample Tree protocols
  - Overcast: Build a single source multicast tree that maximize the bandwidth from the source to the receivers
  - Yoid: A tree is constructed for data delivery, while a mesh is constructed for control messages exchanging.
  - Jungle Monkey: Build a single source multicast tree for file transferring
  - ALMI: Build a single source multicast tree in single server and then distributes it.
Mesh, Tree, and Hybrid Approaches (cont.)

✧ Tree Approach (cont.)

✧ Advantages
  ✧ Lower communication overhead
  ✧ Simple architecture
  ✧ Delay reduction for the peers at the bottom levels => low delay

✧ Disadvantage
  ✧ Single point of failure problem: If the Root peer crashes => its sub-tree is disconnected for a while => may cause in high loss rate
  ✧ Performance bottleneck => low network throughput
  ✧ High recovery time
  ✧ Leaf nodes not contribute their uploading bandwidth => decreasing bandwidth utilization efficiency
Mesh, Tree, and Hybrid Approaches (cont.)

✧ **Mesh Approach**
  ✧ **Nodes constructed a mesh-based topology**
  ✧ **Source-specific Tree for multicasting:**
    ✧ The source is chosen as a root and a routing algorithm is run over the mesh to build the multicast tree
  ✧ **Advantages**
    ✧ High resiliency against peer departures
    ✧ More suitable for multi-source applications
    ✧ No single point of failure problem => low loss rate, High throughput
    ✧ optimized by performing end-to-end latency measurements and adding and removing links to reduce multicast latency
  ✧ **Disadvantages**
    ✧ No pre-defined and simple architecture
  ✧ **Sample mesh ALM protocols**
    ✧ Narada: Creates a mesh and then build multicast trees with DVMRP algorithm.
    ✧ Scattercast: Proxy servers are placed at strategic location. These proxy servers self-organize into multicast trees.

![Figure 8- A Mesh-first ALM network](image)
Mesh-first, Tree-first, and Hybrid Approaches (cont.)

✧ **Tree-Mesh Hybrid Approach** ➔ Best approach (specially in terms of QoS)
  ✧ Builds a tree from IP multicast groups (each with a unique ID) by application layer multicast
  ✧ Dynamically map ALM path to underlying IP multicast path where available to optimize performance
  ✧ Within a region, dynamically transition multicast groups and flows between multicast protocols/mechanisms in response to changes in traffic characteristics, group properties, and network topology
  ✧ Sample hybrid protocols
    ✧ Borg: It has implemented on top of Pastry protocol which is implemented for ALM

![Figure 9- A Hybrid ALM network](image)
Mesh, Tree, and Hybrid Approaches (cont.)

✧ Tree-Mesh Hybrid Approach (cont.)

✧ Advantages
  ✧ Enables end-to-end multicast with incremental native multicast roll-out
  ✧ Have better performance specially in searching process (compared to tree and mesh)
    => higher Network Throughput, lower delay
  ✧ Avoiding replicating group management functions across multiple trees
  ✧ Providing more resilience to failure of members => low loss rate
  ✧ Leveraging on standard routing algorithms => simplifying overlay construction and maintenance (e.g. loop avoidance)

✧ Disadvantages
  ✧ Complexity and performance loss due to
    ✧ Mapping different join/leave and routing protocols
    ✧ Brokering different group management mechanisms
  ✧ Application sensitivity to performance variations
Mesh, Tree, and Hybrid Approaches (cont.)

Summary of Tree, Mesh, Hybrid approaches in terms of QoS:

- **Mesh**: low loss rate, High throughput
- **Tree**: low network throughput, High loss rate, low delay
- **Hybrid**: higher Network Throughput, lower delay, lower loss rate

*The most efficient approach for ALM = Hybrid*
LayeredCast: A Hybrid Mesh-Tree Protocol

- The hybrid video streaming architecture
- Tree construction and improvement algorithm
- Mesh construction, suggestion, and improvement
- Bandwidth reservation mechanism

Diagram:
- Multicast Manager
  - Overlay Buffer
  - Overlay Manager
  - Packetizer
- Tree
  - Topology Manager
  - Leaky Buckets
- Mesh
  - Token Assigner
  - Data Advertiser
  - Request Scheduler
  - Topology Manager
- Weighted Fair-Queue Scheduler
- Network
LayeredCast: A Hybrid Mesh-Tree Protocol (cont.)

- **Uses:**
  - Layered video (FGS)
  - Tree structure for pushing base layer
  - Mesh structure for pulling enhanced layer and retransmit base layer

- **Pros:**
  - Support heterogeneous bandwidth
  - Provide adaptive quality in video
  - Support low delay video transmission
  - Fairness
  - Bandwidth reserve to avoid congestion
  - Reliability for base layer
  - An overlay broadcast structure
# P2P Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Delivery Method</th>
<th>Content Awareness</th>
<th>Video Codec</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>RaDiO</td>
<td>Tree</td>
<td>+</td>
<td>Single Layer</td>
<td>Push in tree</td>
</tr>
<tr>
<td>CoDiO</td>
<td>Tree</td>
<td>+</td>
<td>Single Layer</td>
<td>Push in tree</td>
</tr>
<tr>
<td>CoolStreaming</td>
<td>Mesh</td>
<td>-</td>
<td>Codec Irrelevant</td>
<td>Pull, Data Adv. in mesh</td>
</tr>
<tr>
<td>LSONet</td>
<td>MultiTree</td>
<td>+</td>
<td>Layered</td>
<td>Data Adv. in mesh; Form Multi-tree</td>
</tr>
<tr>
<td>AnySee2</td>
<td>Hybrid</td>
<td>-</td>
<td>Single Layer</td>
<td>Push through tree; Pull, Data Adv. in mesh</td>
</tr>
<tr>
<td>mTreeBone</td>
<td>Hybrid</td>
<td>-</td>
<td>Single Layer</td>
<td>Backbone tree; Pull in mesh</td>
</tr>
<tr>
<td>New CoolStreaming</td>
<td>MultiTree</td>
<td>-</td>
<td>Codec Irrelevant</td>
<td>Data Adv. in mesh; Form Multi-tree</td>
</tr>
<tr>
<td>SWaF</td>
<td>Mesh</td>
<td>-</td>
<td>Codec Irrelevant</td>
<td>Waterfilling BW allocation; Centralized Scheduling</td>
</tr>
</tbody>
</table>
P2P Applications

- **Pioneers**: Napster, Gnutella, FreeNet

- **File sharing**: CFS, PAST \([SOSP’01]\)

- **Network storage**: FarSite \([Sigmetrics’00]\), Oceanstore \([ASPLOS’00]\), PAST \([SOSP’01]\)

- **Web caching**: Squirrel \([PODC’02]\)

- **Event notification/multicast**: Herald \([HotOS’01]\), Bayeux \([NOSDAV’01]\), CAN-multicast \([NGC’01]\), Scribe \([NGC’01]\), SplitStream \([submitted]\)

- **Anonymity**: Crowds \([CACM’99]\), Onion routing \([JSAC’98]\)

- **Censorship-resistance**: Tangler \([CCS’02]\)
Structured P2P Overlays

✧ Characteristics of a structured P2P overlay
  ✧ Leverage pooled resources (storage, bandwidth, CPU)
  ✧ Leverage resource diversity (geographic, ownership)
  ✧ Leverage existing shared infrastructure
  ✧ Scalability
  ✧ Robustness
  ✧ Self-organization

One primitive:

✧ \text{route}(M, X): route message \textit{M} to the live node with nodeID closest to key \textit{X}
  ✧ nodeIDs and keys are from a large, sparse id space
Pastry

- Generic p2p location and routing substrate
  - Self-organizing overlay network
  - Lookup/insert object in $< \log_{16} N$ routing steps (expected)
  - $O(\log N)$ per-node state
- Network proximity routing
- Consistent hashing [Karger et al. ‘97]
  - 128 bit circular id space
  - nodeIDs (uniform random)
  - Obj-IDs (uniform random)
  - Invariant: node with numerically closest nodeID maintains object
SCRIBE: Large-scale, decentralized multicast

✧ Characteristics

✧ Infrastructure to support topic-based publish-subscribe applications

✧ Scalable:
  ✧ Large numbers of topics, subscribers, wide range of subscribers/topic

✧ Efficient:
  ✧ Low delay, low link stress, low node overhead

✧ Advantages

✧ Scribe achieves reasonable performance when compared to IP multicast
  ✧ Scales to a large number of subscribers
  ✧ Scales to a large number of topics
  ✧ Good distribution of load
QoS for Application Level Multicast

✧ Development of a concept to support QoS in structured P2P networks

✧ Modifications of Scribe/Pastry to build QoS-aware multicast tree

✧ Pastry P2P network per active multicast group

✧ QoS-related Pastry ID assignment

✧ Root node with highest QoS requirements / capabilities → largest possible Pastry ID

✧ child QoS requirements / capabilities ≤ parent QoS requirements / capabilities

Figure 10 - QoS in a structured P2P network
QoS aware Multicast Trees with Scribe/Pastry

Joining nodes get IDs dependent on QoS requirements / capabilities:

- **Pastry default:** random ID
- **Higher QoS → higher ID**
- **Scribe constructs multicast trees with required structure to support QoS**

- **QoS support based on**
  - **Reservations**
  - **Measurements**

- **Implementation with Freepastry**
QoS Estimation for Overlay Networks

✧ **Problem:** Mismatch of overlay and real network topology

✧ **Solution:** Topology and QoS aware overlay construction

✧ *Information about end-to-end QoS of potential overlay links required, e.g. by distance (round trip time, available bandwidth) estimation services*

✧ **Existing approaches often**

✧ *only support round trip time estimation*

✧ *require substantial additional infrastructure in the network*

✧ *estimate distances only between members of a peer-to-peer network*
QoS Estimation for Overlay Networks (cont.)

✶ Approach

✧ Nodes are organized in local groups.
✧ Each group stores end-to-end measurements in a distributed repository
✧ Clustering of hosts and groups
✧ Predictions for each cluster

✶ Advantages

✧ locally deployable
✧ no additional infrastructure needed
✧ predictions instead of estimates
✧ supports any type of “distance”

Figure 11- QoS prediction for P2P networks
Routing Mechanisms in ALM

1. Shortest Path Tree (SPT)
   - Constructing degree constraint minimum cost path spanning tree
   - Use RTT to find shortest paths from source to end-hosts -> minimize the time delay for each application while considering the degree constraint.
   - Shortest-path trees may not have the resources to support the quality requirement in terms of QoS.

2. Minimum Spanning Tree
   - Constraints of nodes aren’t important. A low-cost tree (Minimum Spanning Tree) is built
   - MSP = a tree with minimum total cost spanning all the members

Figure 12- (a) A graph with link costs (b) Shortest Path Tree (c) Minimum Spanning Tree
Routing Mechanisms in ALM (cont.)

3. Clustering Structure
   - Construct a hierarchical cluster of nodes with each cluster having a ‘head’ representing it in the higher level
   - Reduction in control overhead
   - Faster joining and management of the tree at the cost of sub-optimal tree
   - Example protocols: ZIGZAG, NICE

4. Peer-to-peer structure
   - The routing is simply done through reverse path forwarding (e.g. Gossamer) or forward-path forwarding (e.g. Bayeux) or a combination of both types (e.g. Borg).
   - Low control overhead
   - Distributed management of the multicast tree
   - Do not restrict the degree of each node => sub-optimal
ALM Control Operation

- ALM control messages tasks
  - Connectivity maintenance
    - Periodic message exchange among hosts is essential to maintain the connectivity of the overlay topology
  - Network condition measurement
    - Measuring the round-trip time and available bandwidth between hosts in order to reduce the stress and stretch & improve the network connectivity

- Overhead Ratio
  - For measuring control overhead
  - Amount of non-data traffic to that of data traffic
    - Non-data traffic: control packets for connectivity maintenance and network condition measurement
Application Level multicast Infrastructure (ALMI)

- ALMI = an application level group communication middleware
  - Allows accelerated application deployment and simplified network configuration, without the need of network infrastructure support.
  - A tree-based topology

- ALMI consists of
  - Session controller -> handles member registration and maintains the multicast tree
    - Checks tree’s connectivity when members join/leave the tree
    - Ensures tree’s efficiency by calculating minimum spanning tree periodically
  - Session member -> receives and sends data & forwards it to designated adjacent neighbors

- ALMI relies on a control protocol for communication between session controller and session members. It handle tasks related to:
  - Membership management
  - Performance monitoring
  - Routing
Application Level multicast Infrastructure (ALMI) (cont.)

- Latency between members = link cost of the MST
- Support data delivery via both TCP and UDP
- Error recovery mechanism
  - *Out-of-band connection direct to the source for re-transmission*
  - *In cases where application has buffering capability, retransmission can happen locally*
- Control topology = unicast connections between members and the controller
  1. Central controller receive updates from each member and computer MST
  2. Routing data of MST sent to members
  3. Members keep a cache of different versions of routing tables -> a packet with new tree version is received
Next Session

Streaming