Chapter 5
Multicast and P2P

A note on the use of these pt slides:
All material copyright 1996-2007
J.F Kurose and K.W. Ross, All Rights Reserved

In-network Duplication
- Flooding: when node receives brdcst pckt, sends copy to all neighbors
  - Problems: cycles & broadcast storm
- Controlled flooding: node only brdcsts pckt if it hasn’t brdcst same packet before
  - Node keeps track of pckt ids already brdcsted
  - Or reverse path forwarding (RPF): only forward pckt if it arrived on shortest path between node and source
- Spanning tree
  - No redundant packets received by any node

Broadcast Routing
- Deliver packets from source to all other nodes
- Source duplication is inefficient:

Spanning Tree
- First construct a spanning tree
- Nodes forward copies only along spanning tree

(a) Broadcast initiated at A
(b) Broadcast initiated at D
Spanning Tree: Creation
- Center node
- Each node sends unicast join message to center node
  - Message forwarded until it arrives at a node already belonging to spanning tree

Multicast Routing: Problem Statement
- **Goal:** find a tree (or trees) connecting routers having local mcast group members
  - tree: not all paths between routers used
  - source-based: different tree from each sender to rcvs
  - shared tree: same tree used by all group members

Approaches for Building Mcast Trees
- **Approaches:**
  - **Source-based tree:** one tree per source
    - Shortest path trees
    - Reverse path forwarding
  - **Group-shared tree:** group uses one tree
    - Minimal spanning (Steiner)
    - Center-based trees

You can read the details about the above approaches in the textbook.

IP Multicast – Related Works
- **Seminal work by S. Deering in 1989**
- **Huge amount of follow-on work**
  - Research
    - 1000s papers on multicast routing, reliable multicast, multicast congestion control, layered multicast
  - Standard: IPv4 and IPv6, DVMRP/CBT/PIM
  - Development: in both routers (Cisco etc.) and end systems (Microsoft, all versions of Unix)
  - Deployment: Mbone, major ISP’s
IP Multicast – Problems

- Problems
  - Scalability
    - Large number of multicast groups
  - Requirement of dynamic spanning tree
    - Practical problem under dynamic environment
  - System complexity
    - Routers maintain state information of multicast groups – deviated from stateless router design
    - Bring out higher level features, e.g. error, congestion control.
  - Autonomous
    - Difficult across different domain for consistent policies

Content Distribution Networks (CDN)

- Push content to servers at network edge close to users
- Support on-demand traffic, but also support broadcast
- Reduce backbone traffic
- CDNs like Akamai places ten of thousands of servers

Content Delivery Networks (CDN)

- Push content to servers at network edge close to users
- Support on-demand traffic, but also support broadcast
- Reduce backbone traffic
- CDNs like Akamai places ten of thousands of servers

The Scale Problem

- The aggregate capacity
  - To reach 1M viewers with MPEG-5 (1.5 Mbps) TV quality video, it requires 1.5 Tbps aggregate capacity
  - CBS NCAA tournament (March 2006), video at 400 Kbps with 268,00 users, the aggregate capacity is 100 Gbps
  - Akamai, the largest CDN service provider, reports at the peak 200 Gbps aggregate capacity

- Implication
  - Self-scaling property
Overlay Multicast

- Application layer multicast or Overlay Multicast
- Build multicast trees at the application end
  - A virtual topology over the unicast Internet
  - End systems communicate through an overlay structure
- Existing multicast approaches
  - Swarming-based (tree-less or data-driven)
  - Tree-based (hierarchical-based)
- Examples:
  - End system multicast (ESM) – Hui Zhang et al.
  - Yoid – Paul Francis et al.
  - ...

Overlay Multicast – Discussion

- Major advantages
  - Efficient multicast service deployment without the need of infrastructure support
  - Feasibility of implementing multicast function at the end of system
  - Easy to apply additional features (metrics)
- Issues
  - Limited topological information at end user side?
  - How to find/determine an ideal topology?
  - Lack of practical system and experiment?

Ideal Overlay

- Efficiency:
  - Routing (delay) in the constructed overlay network is close to the one in the underlying network
  - Efficient use of bandwidth
    - Less duplicated packets on the same link
    - Proper number of connections at each node
  - Support node locality in overlay construction
- Scalability:
  - Overlay remains tractable with the increasing number of hosts and data traffic
  - Small overlay network maintenance cost
  - Overlay constructed in a distributed way and support node locality
**Locality-aware and Randomly-connected Overlay**

- Randomly-connected overlay
- Locality-aware overlay

**Locality-aware Unstructured Overlay**

- Objective of mOverlay [1]
  - The ability to exploit local resources over remote ones when possible
  - Locate nearby object without global communication
  - Permit rapid object delivery
  - Eliminate unnecessary wide-area hops for inter-domain messages
  - Eliminate traffic going through high latency, congested stub links
  - Reduce wide-area bandwidth utilization

---

**Key Concepts for mOverlay**

- Two-level hierarchical network
  - A group consists of a set of hosts close to each other
  - For ANY position P in the underlying network, the distance between P and hosts within a group could be considered as equal
  - Neighbor groups in this overlay are the groups nearby in the underlying network
  - A desirable overlay structure is that most links are between hosts within a group and only a few links between two groups

- Approximation
  - Use neighbors of a group as dynamic landmarks

**Locating Process**

1. Return boot host B from Group 1
2. Measurement and information exchange
3. Contact RP to fetch boot host
4. Measure the distance to boot host and its neighbor groups
5. Determine the closest group with group criterion checking
6. Terminate with group criterion or stop criterion meet

---

Popular Deployed Systems

- Live P2P streaming has become increasingly popular approach
- Many real deployed systems. Just name a few...
  - Coolstreaming: Cooperative Overlay Streaming
    - First release: May 2004
    - Till Oct 2006
      - Download: > 1,000,000
      - Average online users: 20,000
      - Peak-time online user: 80,000
      - Google entries (CoolStreaming): 370,000

CoolStreaming is the base technology for Roxbeam Corp., which launched live IPTV programs jointly with Yahoo Japan in October 2006

Popular Deployed Systems (Cont.)

- PPlive: well-known IPTV system
  - 3.5 M subscribers in 2005
  - 36.9 M subscribers in 2009 predicted
  - May 2006 ~over 200 distinct online channels
  - Revenues could up to $10 B
  - Need to understand current system to design better future systems

- More to come ...

Pull-based Streaming

- Almost all real-deployed P2P streaming systems are based on pull-based protocol
  - Also called “data-driven”/”swarming” protocol

- Basic idea
  - Live media content is divided into segments and every node periodically notifies its neighbors of what packets it has
  - Each node explicitly requests the segments of interest from its neighbors according to their notification
  - Very similar to that of BitTorrent

- The well-acknowledged advantages
  - Robustness and simplicity

Hybrid Pull-Push Protocol

- Pull-based protocol has the tradeoff between control overhead and delay
  - To minimize the delay
    - Node notifies its neighbors of packet arrival immediately
    - Neighbors should also request the packet immediately
    - Result in a remarkable control overhead
  - To diminish the overhead
    - Node can wait until dozens of packets arrived before inform its neighbors
    - Neighbors can also request a bunch of packets each time
    - Leads to a considerable delay
Push-Pull Streaming Mechanism

- How to reduce the delay of pull mechanism while keeping the advantages of pull mechanism?
  - Use the pull mechanism as a startup to measure the partners’ ability to provide video packets
  - Use the push mechanism to reduce the delay
  - Partition the video stream according to the video packets received from the partners in last interval
  - Packets lost during push time interval will be recovered by pull mechanism

GridMedia

- GridMedia is designed to support large-scale live video streaming over world-wide Internet
- The first generation: Gridmedia I
  - Mesh-based multi-sender structure
  - Combined with IP multicast
  - First release: May 2004
- The second generation: Gridmedia II
  - Unstructured overlay
  - Push-pull streaming mechanism
  - First release: Jan. 2005

Real Deployment

- Gala Evening for Spring Festival 2005 and 2006
  - Streaming server: double-core Xeon server
  - Video encoding rate = 300 kbps
  - Maximum connections from server:
    - 2005: 200
    - 2006: 800
  - Partners number = about 10
  - Buffer Deadline = 20s

For the largest TV station in China (CCTV)

Performance Analysis

- Gala Evening for Spring Festival 2005
  - More than 500,000 person times in total, maximum concurrent users 15,239
  - Users from 66 countries, 78.0% from China
  - Enabled 76 times (15,239/200 = 76) in terms of capacity amplification to bounded server outgoing bandwidth
**Performance Analysis (Cont.)**

- Gala Evening for Spring Festival 2006
  - More than 1,800,000 person times in total, maximum concurrent users 224,453
  - Users from 69 countries, 79.2% from China
  - Enabled 280 times (224,453/800 \(\approx\) 280) in terms of capacity amplification to bounded server outgoing bandwidth

**Deployment Experience**

- In 2005, about 60.8% users were behind different types of NATs while at least 16.0% users (in China) accessed Internet via DSL connections
- In 2006, about 59.2% users were behind different types of NATs while at least 14.2% users (in China) accessed Internet via DSL connections

An effective NAT traversal scheme should be carefully considered in the system design of P2P-based live streaming applications

- The average request rate always kept at a record of hundreds in 2005 while thousands in 2006
- Occasionally the request rate rushed to a peak

The high request rate and sporadic flush-crowd essentially pose great challenge on the reliability and stability of RP server and system
Future Directions

- Throughput improvement should not be the only key focus
- Interesting future directions
  - Minimize ISP core network and cross-ISP traffic
    - Use proxy cache and locality-aware technique to relieve the link stress
  - Server bandwidth reduction
    - How to let home users broadcast video with high quality?
  - Real Internet environment
    - Connections across the peer link bridge between ISPs have low rate
    - NAT/firewall prevent end-host from connecting with each other