CS 268: Overlay Networks: Introduction and Multicast

Ion Stoica
April 15-17, 2003

Definition

- Network
  - defines addressing, routing, and service model for communication between hosts
- Overlay network
  - A network built on top of one or more existing networks
  - adds an additional layer of indirection/virtualization
  - changes properties in one or more areas of underlying network
- Alternative
  - change an existing network layer

A Historical Example

- Internet is an overlay network
  - goal: connect local area networks
  - built on local area networks (e.g., Ethernet), phone lines
  - add an Internet Protocol header to all packets

Benefits

- Do not have to deploy new equipment, or modify existing software/protocols
  - probably have to deploy new software on top of existing software
  - e.g., adding IP on top of Ethernet does not require modifying Ethernet protocol or driver
  - allows bootstrapping
    - expensive to develop entirely new networking hardware/software
    - all networks after the telephone have begun as overlay networks
### Benefits

- **Do not have to deploy at every node**
  - Not every node needs/wants overlay network service all the time
  - e.g., QoS guarantees for best-effort traffic
- **Overlay network may be too heavyweight for some nodes**
  - e.g., consumes too much memory, cycles, or bandwidth
- **Overlay network may have unclear security properties**
  - e.g., may be used for service denial attack
- **Overlay network may not scale (not exactly a benefit)**
  - e.g. may require $n^2$ state or communication

### Costs

- **Adds overhead**
  - Adds a layer in networking stack
  - Additional packet headers, processing
  - Sometimes, additional work is redundant
    - E.g., an IP packet contains both Ethernet (48 + 48 bits) and IP addresses (32 + 32 bits)
    - Eliminate Ethernet addresses from Ethernet header and assume IP header?
- **Adds complexity**
  - Layering does not eliminate complexity, it only manages it
  - More layers of functionality → more possible unintended interaction between layers
  - E.g., corruption drops on wireless interpreted as congestion drops by TCP

### Applications

- **Mobility**
  - MIPv6: pretends mobile host is in home network
- **Routing**
- **Addressing**
- **Security**
- **Multicast**

### Applications: Routing

- **Flat space**
  - Every node has a route to every other node
  - $n^2$ state and communication, constant distance
- **Hierarchy**
  - Every node routes through its parent
  - Constant state and communication, $\log(n)$ distance
  - Too much load on root
- **Mesh (e.g., Content Addressable Network)**
  - Every node routes through $2d$ other nodes
  - $O(d)$ state and communication, $n^2$ distance
- **Chord**
  - Every node routes through $O(\log n)$ other nodes
  - $O(\log n)$ state and communication, $O(\log n)$ distance
Applications: Increasing Routing Robustness

- Resilient Overlay Networks [Anderson et al 2001]
  - Overlay nodes form a complete graph
  - Nodes probe other nodes for lowest latency
  - Knowledge of complete graph → lower latency routing than IP, faster recovery from faults

Applications: Security (VPN)

- Provide more security than underlying network
- Privacy (e.g., IPSEC)
  - Overlay encrypts traffic between nodes
  - Only useful when end hosts cannot be secure
- Anonymity (e.g., Zero Knowledge)
  - Overlay prevents receiver from knowing which host is the sender, while still being able to reply
  - Receiver cannot determine receiver exactly without compromising every overlay node along path
- Service denial resistance (e.g., FreeNet)
  - Overlay replicates content so that loss of a large set of node does not prevent content distribution

Problems with IP Multicast

- Scales poorly with number of groups
  - A router must maintain state for every group that traverses it
- Supporting higher level functionality is difficult
  - IP Multicast: best-effort multi-point delivery service
  - Reliability and congestion control for IP Multicast complicated
  - Scalable, end-to-end approach for heterogeneous receivers is very difficult
  - Hop-by-hop approach requires more state and processing in routers
- Deployment is difficult and slow
  - ISP’s reluctant to turn on IP Multicast

Overlay Multicast

- Provide multicast functionality above the IP layer
  → overlay or application level multicast
- Challenge: do this efficiently
- Narada [Yang-hua et al, 2000]
  - Multi-source multicast
  - Involves only end hosts
  - Small group sizes <= hundreds of nodes
  - Typical application: chat
**Narada: End System Multicast**

- A distributed protocol for constructing efficient overlay trees among end systems
- Caveat: assume applications with small and sparse groups
  - Around tens to hundreds of members

**Potential Benefits**

- **Scalability**
  - Routers do not maintain per-group state
  - End systems do, but they participate in very few groups
- **Easier to deploy**
  - Only requires adding software to end hosts
- **Potentially simplifies support for higher level functionality**
  - Use hop-by-hop approach, but end hosts are routers
  - Leverage computation and storage of end systems
  - E.g., packet buffering, transcoding of media streams, ACK aggregation
  - Leverage solutions for unicast congestion control and reliability

**End System Multicast: Narada**

**Performance Concerns**

- **Duplicate Packets:** Bandwidth Wastage
- **Delay from CMU to Berk1 increases**
Overlay Tree

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

Solution

- Two step design
  - Build a mesh that includes all participating end-hosts
    - What they call a mesh is just a graph
    - Members probe each other to learn network related information
    - Overlay must self-improve as more information available
  - Build source routed distribution trees

Overlay Construction Problems

- Dynamic changes in group membership
  - Members may join and leave dynamically
  - Members may die
- Dynamic changes in network conditions and topology
  - Delay between members may vary over time due to congestion, routing changes
- Knowledge of network conditions is member specific
  - Each member must determine network conditions for itself

Mesh

- Advantages:
  - Offers a richer topology & robustness; don’t need to worry much about failures
  - Don’t need to worry about cycles
- Desired properties
  - Members have low degrees
  - Shortest path delay between any pair of members along mesh is small
**Overlay Trees**

- Source routed minimum spanning tree on mesh
- Desired properties
  - Members have low degree
  - Small delays from source to receivers

**Narada Components/Techniques**

- 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
- 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

**Definitions**

- 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 re-added
- Partition avoidance: A partition of the mesh is unlikely to be caused as a result of any single link being dropped

Example

Simulation Results

- Simulations
  - Group of 128 members
  - Delay between 90% pairs < 4 times the unicast delay
  - No link carries more than 9 copies
- Experiments
  - Group of 13 members
  - Delay between 90% pairs < 1.5 times the unicast delay

Summary

- End-system multicast (NARADA): aimed to small-sized groups
  - Application example: chat
- Multi source multicast model
- No need for infrastructure
- Properties
  - Low performance penalty compared to IP Multicast
  - Potential to simplify support for higher layer functionality
  - Allows for application-specific customizations
Other Projects

- **Overcast** [Jannotti et al, 2000]
  - Single source tree
  - Uses an infrastructure; end hosts are not part of multicast tree
  - Large groups – millions of nodes
  - Typical application: content distribution
- **Scattercast** (Chawathe et al, UC Berkeley)
  - Emphasis on infrastructural support and proxy-based multicast
  - Uses a mesh like Narada, but differences in protocol details
- **Yoid** (Paul Francis, Cornell)
  - Uses a shared tree among participating members
  - Distributed heuristics for managing and optimizing tree constructions

Overcast

- Designed for throughput intensive content delivery
  - Streaming, file distribution
- Single source multicast; like Express
- Solution: build a server based infrastructure
- Tree building objective: high throughput

Tree Building Protocol

- Idea: Add a new node as far away from the route as possible without compromising the throughput!

```
Join(new, root) {
  current = root;
  do {
    B = bandwidth(new, current);
    B1 = 0;
    forall n in children(current) {
      B1 = bandwidth(new, n);
      if (B1 >= B) {
        current = n;
        break;
      }
    }
  } while (B1 >= B);
  new->parent = root;
}
```

Details

- A node periodically reevaluates its position by measuring bandwidth to its
  - Siblings
  - Parent
  - Grandparent
- The Up/Down protocol: track membership
  - Each node maintains info about all nodes in its sub-tree plus a log of changes
    - Memory cheap
    - Each node sends periodical alive messages to its parent
    - A node propagates info up-stream, when
      - Hears first time from a children
      - If it doesn’t hear from a children for a present interval
      - Receives updates from children
Details

- Problem: root → single point of failure
- Solution: replicate root to have a backup source
- Problem: only root maintain complete info about the tree; need also protocol to replicate this info
- Elegant solution: maintain a tree in which first levels have degree one
  - Advantage: all nodes at these levels maintain full info about the tree
  - Disadvantage: may increase delay, but this is not important for application supported by Overcast

Some Results

- Network load < twice the load of IP multicast (600 node network)
- Convergence: a 600 node network converges in ~ 45 rounds

Summary

- Overcast: aimed to large groups and high throughput applications
  - Examples: video streaming, software download
- Single source multicast model
- Deployed as an infrastructure
- Properties
  - Low performance penalty compared to IP multicast
  - Robust & customizable (e.g., use local disks for aggressive caching)