Application Layer
Multicast

11/28/2001

Some slides on Narada are from Yang-hua Chu

Review: IP Multicast

- Service model
  - a group identified by a class D IP address
  - open group: anyone can send to the group; anyone can join the group
  - best-effort service model

- Implementation
  - LAN: map from IP multicast to Ethernet multicast
  - WAN:
    - source tree: e.g. DVMRP
    - shared tree: e.g. CBT

- Extended service model
  - EXPRESS
Issues with IP Multicast

- Many years of development, but deployment is difficult and slow
  - ISP’s reluctant to turn on IP Multicast
  - scalability with number of groups
    - routers need to maintain per-group state
    - aggregation of multicast addresses is complicated
  - open group model, lack control
  - address allocation

Application Layer/End System Multicast (ESM)
Discussion

- What are the benefits of ESM?
- What are the potential issues of ESM?

Many Projects

- ALMI (Washington U., 2001)
- Bayeux (UC Berkeley, 2001)
- Narada [Chu et al., CMU, 2000]
- NICE (U. Maryland, 2000)
- Overcast [Jannotti et al., Cisco, 2000]
- Scalable Self-Organizing Overlay (U. Washington, 2001)
- Scattercast (UC Berkeley, 2000)
- Yoid (Fastforward/ACIRI, 2000)
Two Representative Examples

- **Narada [Chu et al, 2000]**
  - multi-source multicast
  - involves only end hosts
  - small group sizes ≤ hundreds of nodes
  - typical application: chat, conferencing

- **Overcast [Jannotti et al, 2000]**
  - single source tree
  - assume an infrastructure; end hosts are not part of multicast tree
  - large groups ~ millions of nodes
  - typical application: content distribution

---

**Narada: A Two-Step Design**

- Step 1: build a mesh that includes all participating end-hosts (why build a mesh?)
- Step 2: multicast over the mesh
Step 1: Mesh Management

- Mesh connection: connect a new host to the mesh
  - members have low degrees
- Mesh optimization: distributed heuristics for improving bandwidth and delay between members

Mesh Optimization

- Measuring the properties of the overlay network
  - how to measure latency?
  - how to measure bandwidth?
- Link restructuring
Optimizing Mesh Quality

- Desired properties
  - 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

- Add a new link
  - members periodically probe other members at random
  - new link added if
    - Utility_Gain of adding link > Add_Threshold

- Drop a current link
  - members periodically monitor existing links
  - existing link dropped if
    - Cost of dropping link < Drop Threshold

Example: Add a New Overlay Link

Delay improves to Stan1, CMU but marginally.
Do not add link!

Delay improves to CMU, Gatech1 and significantly.
Add link!
Example: Drop a Current Overlay Link

Used by Berk1 to reach only Gatech2 and vice versa: Drop!!

Step 2: Multicast Over the Mesh

- Shortest widest path routing
  - how to consider both bandwidth and delay?
  - how to avoid oscillation?
- Reverse path forwarding
Simulation Results

- Simulations
  - group of 128 members
  - delay between 90% pairs < four 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

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

Maintain Tree Membership

- 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 is 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
Fault Tolerance of Tree Root

- Problem: root \(\rightarrow\) single point of failure
- Solution: replicate root to have a backup source
- Problem: only root maintains complete info about the tree; also need 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

Nodes maintaining full Status info about tree

Some Results

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

- Will ESM fly?
- Will ESM fragment the protocol space:
  - is every application going to come with its multicast suite?
  - are we going to end up with very few de facto standards for different categories of applications?

Backup
The Terms Defined [CRZ00]

- 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