CSCI 5105

Instructor: Abhishek Chandra

Today

- Multicast
- Flooding
- Gossiping

One-to-Many Communication

- Same data/message may have to be sent to multiple receivers
- How to find the multiple receivers?
- How to efficiently route data to each receiver?

Multicast

- Multicast group: A special address/id
- A node can join/leave a multicast group
  - Will receive data sent to the group
- Studied a lot in networking
- Deployment difficult
  - Lot of state to keep and exchange
  - Lack of standardization

Application-Level Multicast

- Use overlay network
- Multicast done on top of the overlay network
  - Multicast joins/leaves at application level
  - Multicast routing through the overlay network
  - Can differ substantially from real network

5

Multicast Routing

- How does data flow from a source to the group members?
- Multicast tree:
  - Source at the root
  - Members at the leaves
  - Members/forwarders in between

6

Multicast Tree

- How to form multicast trees?
- Use topology of overlay network
  - Neighbor relationship
  - E.g.: DHT network
- Incremental
  - Joining nodes attach to an existing node in the tree
- SuperPeer based:
  - Some nodes know about the topology of the tree

7

Robustness

- What if a node in the tree fails/leaves?
- Option 1: Reorganize the tree
  - Orphan nodes reattach to a different parent
  - Which parent?
- Option 2: Maintain a mesh instead of a tree
  - Have alternate paths for multicast routing

8

Efficiency

- Several possible multicast trees for a group
  - Which one is the best?
  - How to construct the best tree?

9

Efficiency Metrics

- Link Stress: How often does a packet cross a physical link?
  - Measured per-link
- Stretch: Delay in the overlay/Delay in the network
  - Measured per-node-pair
- Tree Cost: Global metric
  - Cost can be link stress or stretch

10

Optimal Multicast Tree

- Total tree cost is minimal
  - Minimum spanning tree
  - What is the minimum spanning tree for stretch?
- In practice, may have other constraints
  - E.g.: maximum number of children for each node
- How to construct such "optimal" trees dynamically?

11

Building "Good" Trees

- Centralized: Using rendezvous nodes
- Decentralized: Switch-trees

12

Rendezvous Nodes

- Keeps track of current members of group
- A joining node gets this list from R.N.
  - Can be linked to a node to minimize incremental cost

Switch Trees

- Construct some initial multicast tree
- Goal: Incrementally change to an optimal tree
  - Each node can switch its parent
- Each node probes other nodes
  - Switch parent to a better node if found
  - E.g.: minimize delay
  - Maintain other constraints such as max-children
  - Avoid race conditions
- Failure of parent is an extreme case
  - Attach to the root
  - Move to a different position in the tree gradually

Flooding-based Multicast

- Each node sends message to all its neighbors
  - Except from where it received the message
- Naïve flooding:
  - Number of messages = number of edges O(N^2)
- Probabilistic flooding:
  - Send message with probability *p_flood*
  - Number of messages decreases substantially
  - Risk: a node may not be reached
- Structured overlays: Can do a controlled flooding
  - E.g.: hypercubes, DHTs require (N-1) messages

Gossiping

- Information needs to be sent to "all" nodes eventually
  - E.g.: updates to a file
- What if there is a large number of nodes?
- Goals:
  - Scalability
  - Efficiency (low latency)
- Idea: Use non-deterministic data dissemination

Epidemic Propagation

- Based on epidemic theory
  - Infecting someone = sending data
  - Goal is to spread epidemics faster
- Types of nodes
  - Infective: Node that has data and is willing to spread it further
  - Susceptible: Node which has not seen the data
  - Removed: Node which has the data but not willing to spread it further

17

Anti-Entropy

- A node P chooses a node Q at random
- P and Q exchange updates:
  - Push, pull or push-pull
- Pure push-or-pull based approaches not as effective
  - Why?
- In general, number of rounds to update all nodes is O(log N)

18

Rumor Spreading

- Infective node P selects Q at random
  - P tries to push data to Q
  - If Q already has the data, P stops spreading further with a probability *p_stop*
- Good at spreading updates quickly
- Cannot guarantee all nodes will be updated
  - A fraction of nodes remains uninfected
- To update all nodes:
  - Combine gossiping and anti-entropy

19

Directional Gossiping

- Use network topology for faster propagation
  - Assign different probability of selection for different nodes
  - E.g.: Higher probabilities for nodes with few neighbors

20

Applications of Epidemic Protocols

- Data updates
- Data deletion
  - Death certificates: Keep track of deleted data
  - Throw away death certificates after a while
    - Can all nodes do this?
- Aggregating information
  - Computing averages
    - x_i, x_j = $(x_i+x_j)/2$
  - Can use this to compute number of nodes in the system

21