CS-438
Decentralized Systems Engineering

Week 3

Broadcast medium: UseNet, mailing lists, Twitter...

Goals

- Send msg to specific topic group(s)
- latency: reasonable delay
- timeshifting / buffering: read old msgs
- reliable delivery
- authenticity of sender
- integrity of content
- bandwidth efficiency

Challenges

quality of content: avoiding spam, abuse

Consensus? ordering?

network topology changes

Scalability

Consistency
availability
partition-tolerance

CAP

Broadcast algorithms

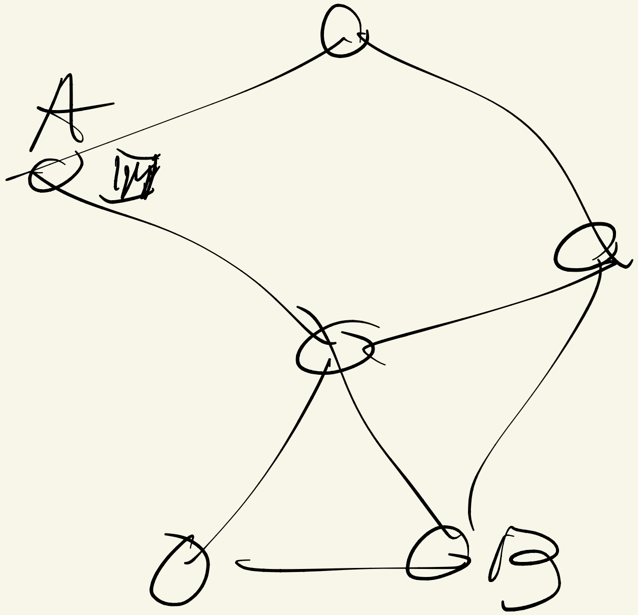
Naive broadcast:

- Send copies of M to "everyone"
  - high BW cost to A
  - who is "everyone"? master list?
  - reliability: B offline? A offline?

Reliable broadcast:

- A sends M to all
- other nodes echo

High bandwidth $O(n^2)$



UseNet needed:

- scalability
- adaptivity

Gossip algorithms

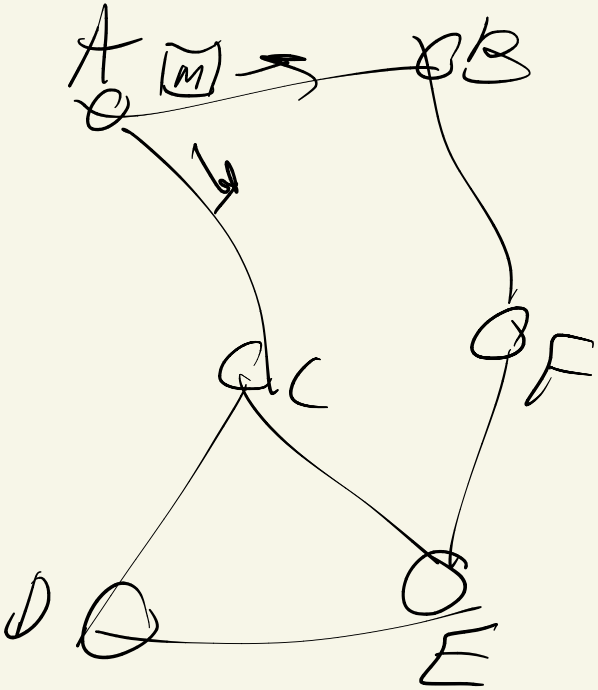
- Naive gossip
  - A sends M to neighbors
  - other nodes on receiving M, resend it to (other) neighbors

Problem: loops, exponential explosion

Solutions:

- Duplicate suppression
- force tree-structured topology (STP)

very delicate!



Message hdr

From:
Path:
Newsgroups:
Subject:
Message-ID:
<1234@foo.com>

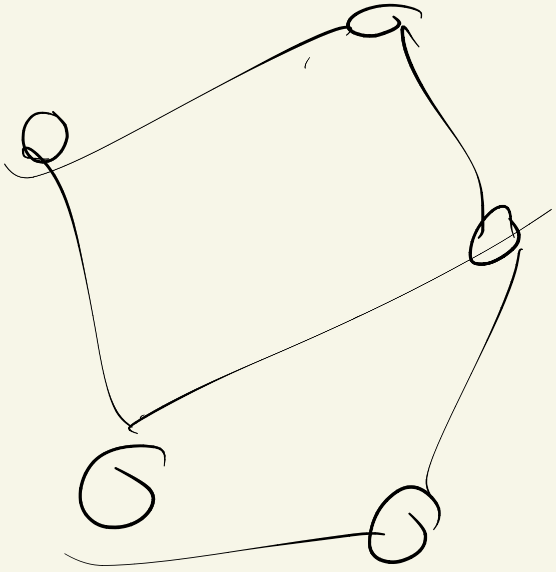
Body

Gossip efficency

- Basic:
  - A sends M
  - other nodes on reciving M the first time, resend to neighbors

  Total cost: $n(d-1) \Rightarrow O(nd)$

  Ideal: $O(n)$



degree d

- Randomized gossip ("epidemic" algs)
  - Rumor-mongering
  - Anti-entropy

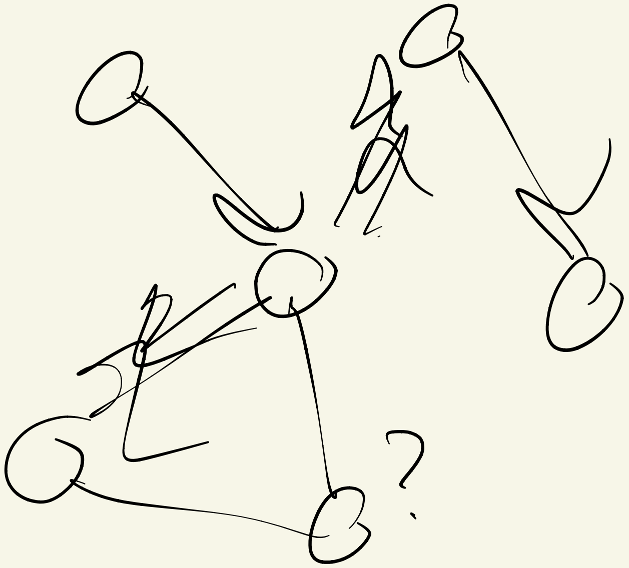
Rumor mongering

- on receiving new msg M,
  pick 1 random neighbor, send
  if new to friend
      stay excited, repeat
  else
      flip a coin. heads
      Tails → stop

Effectiveness:

- early stage
- late stage

O(n)



Anti-entropy

- each node wakes up occasionally, picks 1 random neighbor, "compares notes"
  $O(n)$ cost
- bad (slow) at early stage
- but very effective in late stage (after rumor mongering)