CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

GOSSIP

Lecture C

GOSSIP ANALYSIS

Properties

Claim that the simple Push protocol

- Is lightweight in large groups
- Spreads a multicast quickly
- Is highly fault-tolerant

Analysis

From old mathematical branch of *Epidemiology* [Bailey 75]

- Population of ($n+1$) individuals mixing homogeneously
- Contact rate between any individual pair is β
- At any time, each individual is either uninfected (numbering x) or infected (numbering y)
- Then, $x_0 = n, y_0 = 1$

  and at all times $x + y = n + 1$
- Infected–uninfected contact turns latter infected, and it stays infected

Analysis (contd.)

- Continuous time process
- Then

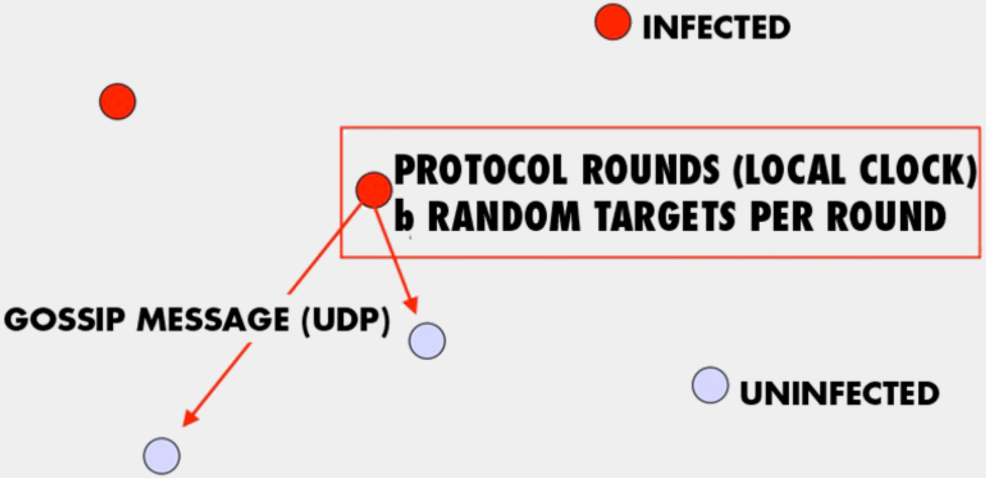
$$\frac{dx}{dt} = -\beta xy \qquad \text{(why?)}$$

with solution:

$$x = \frac{n(n+1)}{n + e^{\beta(n+1)t}}, y = \frac{(n+1)}{1 + ne^{-\beta(n+1)t}}$$

(can you derive it?)

Epidemic Multicast



Epidemic Multicast Analysis

$$\beta = \frac{b}{n} \qquad \text{(why?)}$$

Substituting, at time *t=clog(n)*, the number of infected is

$$y \approx (n+1) - \frac{1}{n^{cb-2}}$$

(correct? can you derive it?)

Analysis (contd.)

- Set *c,b* to be small numbers independent of *n*
- Within *clog*(*n*) rounds, [**low latency**]
  - all but $\frac{1}{n^{cb-2}}$ number of nodes receive the multicast

    [**reliability**]

  - each node has transmitted no more than *cblog*(*n*)gossip messages [**lightweight**]

Why is log(N) low?

- Log(N) is not constant in theory
- But pragmatically, it is a very slowly growing number
- Base 2
  - Log(1000) ~ 10
  - Log(1M) ~ 20
  - Log(1B) ~ 30
  - Log(all IPv4 address) = 32

Fault-tolerance

- Packet loss
  - 50% packet loss: analyze with *b* replaced with *b/2*
  - To achieve same reliability as 0% packet loss, takes twice as many rounds
- Node failure
  - 50% of nodes fail: analyze with *n* replaced with *n/2* and *b* replaced with *b/2*
  - Same as above

Fault-tolerance

- With failures, is it possible that the epidemic might die out quickly?
- Possible, but improbable:
  - Once a few nodes are infected, with high probability, the epidemic will not die out
  - So the analysis we saw in the previous slides is actually behavior *with high probability*

  [Galey and Dani 98]
- Think: Why do rumors spread so fast? Why do infectious diseases cascade quickly into epidemics? Why does a virus or worm spread rapidly?

Pull Gossip: Analysis

- In all forms of gossip, it takes O(log(N)) rounds before about N/2 gets the gossip
  - Why? Because that's the fastest you can spread a message – a spanning tree with fanout (degree) of constant degree has O(log(N)) total nodes
- Thereafter, pull gossip is faster than push gossip
- After the *i*th, round let p_i be the fraction of non-infected processes. Then

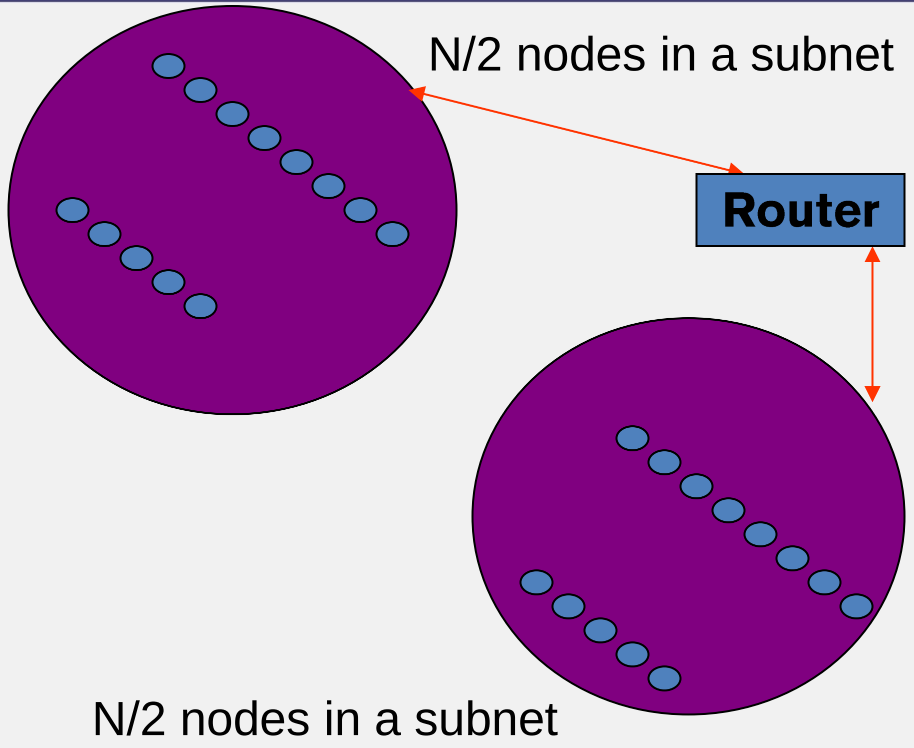
(k=number of gossip pulls per round per process)

$$p_{i+1} = \left(p_i\right)^{k+1}$$

- This is super-exponential
- Second half of pull gossip finishes in time O(log(log(N))

Topology-Aware Gossip

- Network topology is hierarchical
- Random gossip target selection => core routers face O(N) load (Why?)
- Fix: In subnet *i*, which contains n_i nodes, pick gossip target in your subnet with probability $1/n_i$
- Router load=O(1)
- Dissemination time=O(log(N))
    - Why?



Answer – Push Analysis (contd.)

Using: $$\beta = \frac{b}{n}$$

Substituting, at time *t=clog(n)*

$$y = \frac{n+1}{1+ne^{-\frac{b}{n}(n+1)c\log(n)}} \approx \frac{n+1}{1+\frac{1}{n^{cb-1}}}$$

$$\approx (n+1)(1-\frac{1}{n^{cb-1}})$$

$$\approx (n+1) - \frac{1}{n^{cb-2}}$$