

Peer-to-Peer and GRID Computing, 2G1526 Lecture 06

**Seif Haridi and Ali
Ghods**
ECS, KTH



5/23/07

S. Haridi, 2G1526, Lecture 06

1

Epidemics and Gossip

- Important technique to solve problems in dynamic large scale systems
 - Scalable
 - Simple
 - Robust to node failures, message loss and transient network disruptions (network partitions ...)



5/23/07

S. Haridi, 2G1526, Lecture 05

2



Gossip Intro.

- Suppose that I know something
- I'm sitting next to Ali, and I tell him
 - Now 2 of us "know"
- Later, he tells Cosmin and I tell Tallat
 - Now 4
- This is an example of a *push* epidemic
- Pull happens if Ali asks me instead
- *Push-pull* occurs if we exchange data

5/23/07

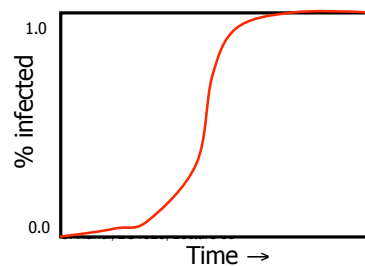
S. Haridi, 2G1526, Lecture 05

3



Gossip scales very nicely

- Participants' loads independent of size
- Network load linear in system size
- Information spreads in $\log(\text{system size})$ time



5/23/07

4

So what is a gossip protocol (1/2)?



- Cyclic/Periodic, pair-wise interaction between peers
- The amount of information exchanged is of (small) bounded size per cycle
- The state of each peer is bounded (small)
- During interaction the state of one of both peers changes in a way that reflects the state of the other peer

5/23/07

S. Haridi, 2G1526, Lecture 05

5

So what is a gossip protocol (2/2)?



- Random peer selection
 - The full peer set, or
 - Small set of neighbors
- Reliable communication is not assumed
- The protocol cost is negligible
 - The frequency of interaction is much lower than message round-trip times

5/23/07

S. Haridi, 2G1526, Lecture 05

6

Gossip protocols in distributed systems



- Information Dissemination Protocols: gossip to spread information in a manner that produces bounded worst-case loads
 - Event dissemination protocols use gossip to perform multicast. They report events periodically
 - Background data dissemination protocols gossip about information associated with nodes

5/23/07

S. Haridi, 2G1526, Lecture 05

7

Gossip protocols in distributed systems



- *Anti-entropy protocols for repairing replicated data*, which operate by comparing replicas and reconciling differences
 - “I have 6 updates from Cosmin”
- If we aren't in a hurry, gossip to replicate data too
- Typical use (bimodal Multicast)
 - Use a best effort multicast
 - Then gossip to fill the gaps

5/23/07

S. Haridi, 2G1526, Lecture 05

8

Gossip about membership



- Start with a *bootstrap protocol*
 - For example, processes go to some web site and it lists a dozen nodes where the system has been stable for a long time
 - Pick one at random
- Then track “processes I’ve heard from recently” and “processes other people have heard from recently”
- Use push gossip to spread the word

5/23/07

S. Haridi, 2G1526, Lecture 05

9

Gossip protocols in distributed systems



- *Protocols that compute aggregates*
- These compute a network-wide aggregate by:
 - Sampling information at the nodes in the network
 - Combining the values to arrive at a system-wide value
 - Like wave algorithms computing average, max, min, ...
- Example: the number of nodes in the system?

5/23/07

S. Haridi, 2G1526, Lecture 05

10

Gossip protocols in distributed systems



- *Protocols that arrange network topology*
- Example: *rings (T-man)*
 - Starting from local view of fixed size in a random network
 - Do bidirectional exchange of the view with a random peer \Rightarrow 2 views
 - Keep peers with ids near to you \Rightarrow 1 view
 - λ repeat

5/23/07

S. Haridi, 2G1526, Lecture 05

11

Gossip protocols in distributed systems



- Gossip has been used for many other things
 - Global failure detection
 - Global clock synchronization
 - Reputation dissemination
 - ...

5/23/07

S. Haridi, 2G1526, Lecture 05

12

Information Dissemination



- Starting with one peer that wants to disseminate some message
- Every peer does the following:
 - Buffers every message (information unit) it receives up to a certain *buffer capacity* b
 - Forwards that message a limited number of *times* t
 - Forwards the message each time to f randomly selected set of processes, f called the *fanout* of the dissemination

5/23/07

S. Haridi, 2G1526, Lecture 05

13

Information Dissemination ?



- Dissemination is like epidemics infection
- Given a system size n (population size)
 - What is reliability of information delivery given b , t , f , n ?
 - Does it depend on n ?
 - How many cycles do we need to infect all peers?

5/23/07

S. Haridi, 2G1526, Lecture 05

14

Infect forever model



- Fixed size population n
- One infectious individual at round 1
- Infected individuals remain infectious throughout
- Y_r is the fraction of individuals infected at round r
- Assume that infectious individuals try to contaminate f other members in each round:

$$Y_r \approx \frac{1}{1 + n \cdot e^{-f \cdot r}}$$

- The ratio of number of infected individuals to number of uninfected ones increases exponentially fast on average, by a factor of e^f in each round.

5/23/07

S. Haridi, 2G1526, Lecture 05

15

Latency of infection Infect forever



- The number R of rounds necessary to infect the entire system respects the following equation:

$$R = \log_{f+1}(n) + \frac{1}{f} \log(n) + O(1)$$

- $f = 2$
- Round 1: 1, Round 2: 3, Round 3: 3+6 = 9

5/23/07

S. Haridi, 2G1526, Lecture 05

16



Infect and Die Model

- Each process will take action to communicate a message exactly once, namely after receiving that message for the first time
- No further action is taken, even if copies of the same message are received again

π is the proportion of processes eventually contaminated

$$\pi = 1 - e^{-\pi \cdot f}$$

5/23/07

S. Haridi, 2G1526, Lecture 05

17



Infect and Die Model

- If $f = 1$, then $\pi = 0$ satisfies the equation
- λ When $f > 1$, π becomes positive
- λ Example $f = 2$, $\pi = 0.5$
- λ π never becomes 1, for very large population

π is the proportion of processes eventually contaminated

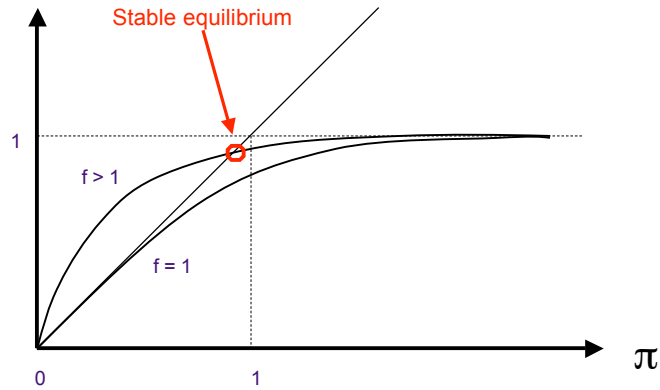
$$\pi = 1 - e^{-\pi \cdot f}$$

5/23/07

S. Haridi, 2G1526, Lecture 05

18

Infect and Die Model



5/23/07

S. Haridi, 2G1526, Lecture 05

19

Infect and Die Model finite size population n

- *When is all the population infected ?*
- *When $f/\log(n)$ crosses 1 all population can become infected*

5/23/07

S. Haridi, 2G1526, Lecture 05

20

Latency of infection “infect and die”



- f should be $O(\log(n))$
- The number R of rounds necessary to infect the entire system respects the following equation:

$$R = \frac{\log(n)}{\log(\log(n))} + O(1)$$

Issues in info dissemination



- *Membership*
 - How peers get to know each other, and how many do they need to know
- *Network awareness*
 - How to make the connections between peers reflect the actual network topology
- *Buffer management*
 - Which information to drop at a process when its storage buffer is full

Membership



- Each process has a partial view
- The view should have a random sample of node, even under churn
- Whenever a process forwards a message, it also includes in this message a set of processes it knows
- Hence, the process that receives the message can enhance the list of processes it knows by adding new processes

5/23/07

S. Haridi, 2G1526, Lecture 05

23

Membership protocols Requirements



- Uniformity
- Adaptively under churn
 - The parameters have to be tuned (t and f)
- Bootstrapping
 - To start with!
- Cyclone, Newscast, SCAMP

5/23/07

S. Haridi, 2G1526, Lecture 05

24

Network Awareness



- Most solutions proposed to address this issue rely on a hierarchical organization of processes which attempts to reflect the network topology

Buffer Management



- Depending on the broadcast rate, the buffer capacity of every process may be insufficient to ensure that every message is buffered long enough
- Messages are classified according to their *age* (the number of processes the message went through)
- Replace old messages
- Replace subsumed messages (depends on application semantics)

Specific gossip framework by Jelassy and Babaoglu



5/23/07

S. Haridi, 2G1526, Lecture 05

27

Proactive gossip framework



```
// active thread
do forever
  wait(T time units)
  q = SelectPeer()
  push S to q
  pull Sq from q
  S = Update(S, Sq)

// passive thread
do forever
  (p, Sp) = pull * from *
  push S to p
  S = Update(S, Sp)
```

Proactive gossip framework



- To instantiate the framework, need to define
 - Local state **S**
 - Method **SelectPeer()**
 - Style of interaction
 - **push-pull**
 - **push**
 - **pull**
 - Method **Update()**

5/23/07

S. Haridi, 2G1526, Lecture 05

29

#1 Aggregation



5/23/07

S. Haridi, 2G1526, Lecture 05

30

Gossip framework instantiation



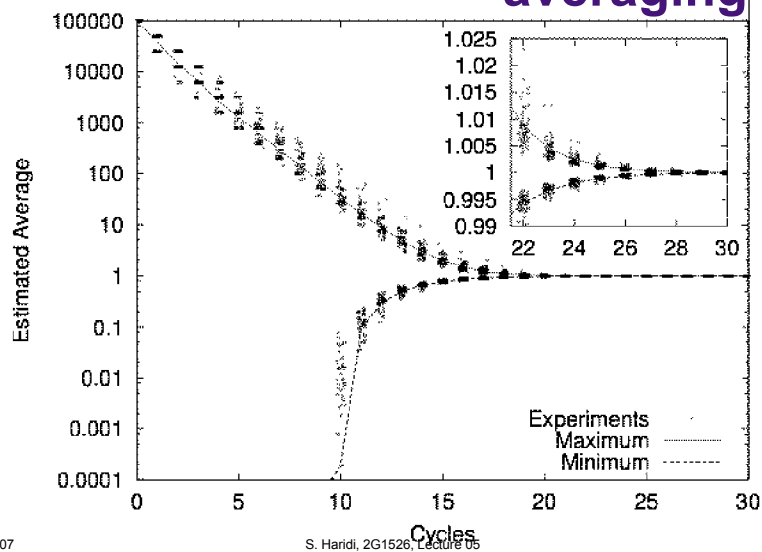
- Style of interaction: push-pull
- Local state **S**: Current estimate of global aggregate
- Method **SelectPeer()**: Single random neighbor
- Method **Update()**: Numerical function defined according to desired global aggregate (arithmetic/geometric mean, min, max, etc.)

5/23/07

S. Haridi, 2G1526, Lecture 05

31

Exponential convergence of averaging



5/23/07

S. Haridi, 2G1526, Lecture 05

32

Properties of gossip-based aggregation



- In gossip-based averaging, if the selected peer is a globally random sample, then the variance of the set of estimates decreases exponentially
- Convergence factor:

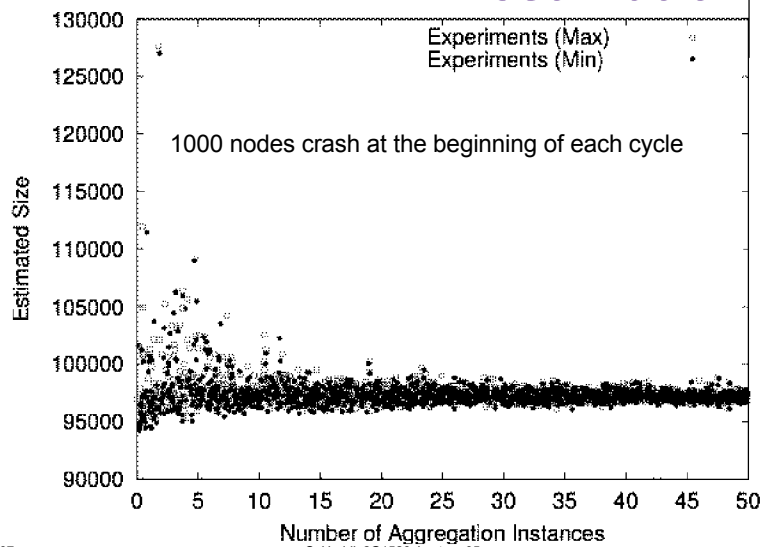
$$\rho = \frac{E(\sigma_{i+1}^2)}{E(\sigma_i^2)} \approx \frac{1}{2\sqrt{e}} \approx 0.303$$

5/23/07

S. Haridi, 2G1526, Lecture 05

33

Robustness of network size estimation

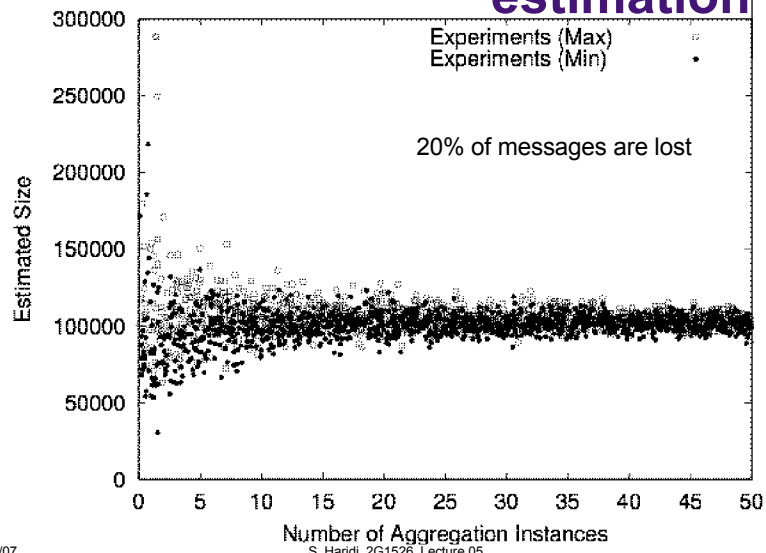


5/23/07

S. Haridi, 2G1526, Lecture 05

34

Robustness of network size estimation



#2 Topology Management

5/23/07

S. Haridi, 2G1526, Lecture 05

36

Gossip framework instantiation



- Style of interaction: push-pull
- Local state **S**: Current neighbor set
- Method **SelectPeer()**: Single random neighbor
- Method **Update()**: Ranking function defined according to desired topology (ring, mesh, torus, DHT, etc.)

5/23/07

S. Haridi, 2G1526, Lecture 05

37

Mesh Example



Mesh.mov

5/23/07

S. Haridi, 2G1526, Lecture 05

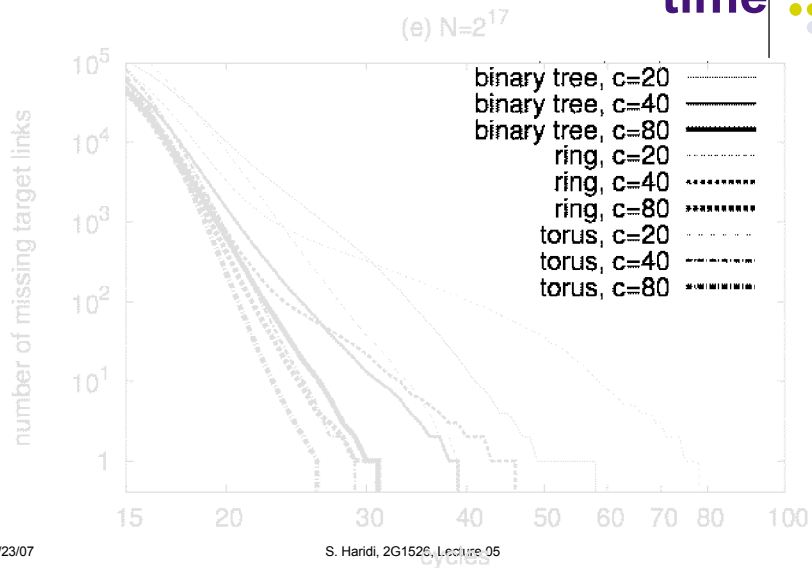
38

Sorting example

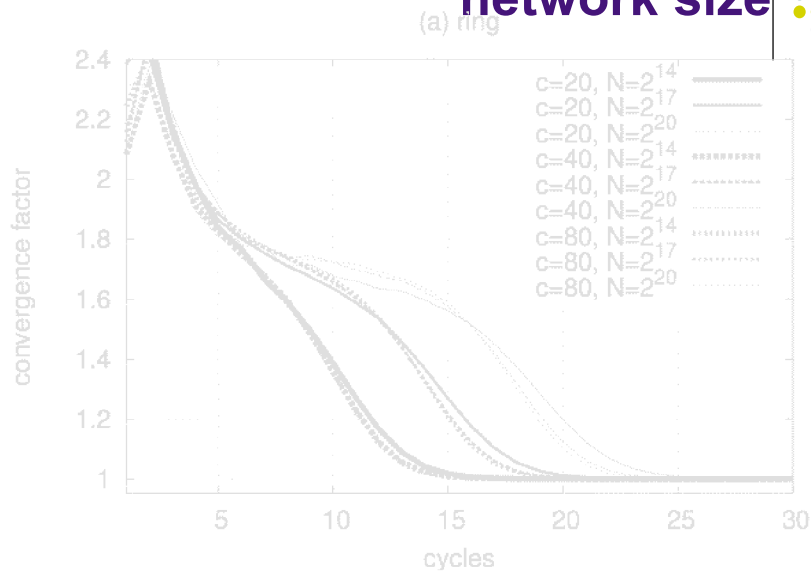


Line.mov

Exponential convergence - time



Exponential convergence - network size



#3

Heartbeat Synchronization



Synchrony in nature



- Nature displays astonishing cases of synchrony among independent actors
 - Heart pacemaker cells
 - Chirping crickets
 - Menstrual cycle of women living together
 - Flashing of fireflies
- Actors may belong to the same organism or they may be parts of different organisms

5/23/07

S. Haridi, 2G1526, Lecture 05

43

Coupled oscillators



- The “Coupled oscillator” model can be used to explain the phenomenon of “self-synchronization”
- Each actor is an independent “oscillator”, like a pendulum
- Oscillators coupled through their environment
 - Mechanical vibrations
 - Air pressure
 - Visual clues
 - Olfactory signals
- They influence each other, causing minor local adjustments that result in global synchrony

5/23/07

S. Haridi, 2G1526, Lecture 05

44

Fireflies



- Certain species of (male) fireflies (e.g., *Luciola pupilla*) are known to synchronize their flashes despite:
 - Small connectivity (each firefly has a small number of “neighbors”)
 - Communication not instantaneous
 - Independent local “clocks” with random initial periods

5/23/07

S. Haridi, 2G1526, Lecture 05

45

Gossip framework instantiation



- Style of interaction: push
- Local state **S**: Current phase of local oscillator
- Method **SelectPeer()**: (small) set of random neighbors
- Method **Update()**: Function to reset the local oscillator based on the phase of arriving flash

5/23/07

S. Haridi, 2G1526, Lecture 05

46

Experimental results



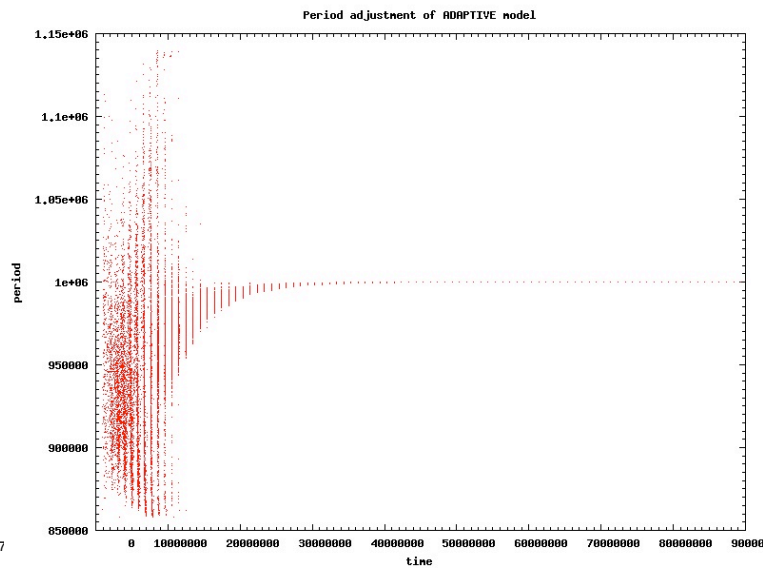
fireflies.mov

5/23/07

S. Haridi, 2G1526, Lecture 05

47

Exponential convergence



5/23/07

48



Assume $n = 64 = 2^6$, $f = 2$

$$Y_0 \approx \frac{1}{1 + 64/2^0} = \frac{1}{65}$$

$$Y_1 \approx \frac{1}{1 + 64/2^2} = \frac{1}{\frac{4 + 64}{4}} = \frac{4}{68}$$

Infect and Die Model

is the proportion of processes eventually
contaminated

$$\pi = 1 - e^{-\pi \cdot f}$$

$$\pi = 1 - e^{-\pi} = 1 - \frac{1}{e^\pi}, (f = 1)$$

$\pi = 0$ satisfies this equation

$$\text{for } f = 2, 1 - \frac{1}{e^{2\pi}} \approx 1 - \frac{1}{2^{2\pi}}, (\pi = 0.5), 1 - \frac{1}{2^{2/2}} = 0.5$$

$$\text{for } f = 1 + \delta, 1 - \frac{1}{e^{\pi(1+\delta)}} = 1 - \frac{1}{e^\pi \cdot e^{\pi\delta}}, (\text{small } \pi) \approx$$

$$1 - \frac{1}{1 + \varepsilon} \approx \pi$$