Network Science: Aggregation

Ozalp Babaoglu
Dipartimento di Informatica — Scienza e Ingegneria
Università di Bologna
www.cs.unibo.it/babaoglu/

Processes on networks

- Networks serve as the underlying transport mechanism for processes that are being carried out *on top* of them
- Interested in studying the effects of the underlying network on the dynamics of the higher-level process
- Processes already studied
  - Gossiping
  - Heartbeat synchronization
  - Formation creation
- Today we will study
  - Aggregation

Aggregation (collective computation)

- Initially, each node has a (numeric) local state
- Want to compute in a *decentralized manner* a (global) aggregate function over the initial values
- In the end, the aggregate value must be known (locally) at each node
- Examples of aggregate functions:
  - Average
  - Min-max
  - Geometric mean
  - Variance
  - Network size

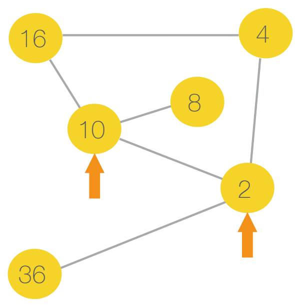
Aggregation Gossip framework instantiation

- 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.)

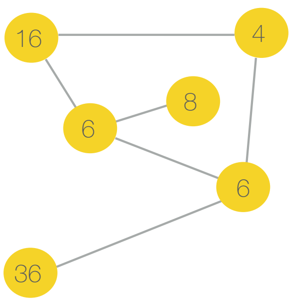
Aggregation
Gossip framework instantiation

- Local variable S_p contains current estimate of the aggregate
- Need to give implementations for
  - `SelectPeer()`
  - `Update(`S_p, S_q`)`
- `SelectPeer()` picks a random neighbor
  - `Update(`S_p, S_q`)` $= \frac{(S_p + S_q)}{2}$ (average)
  - `Update(`S_p, S_q`)` $= \sqrt{(S_p S_q)}$ (geometric mean)
  - `Update(`S_p, S_q`)` $=$ `max(`S_p, S_q`)` (maximum)
- More complex functions built by combining the above

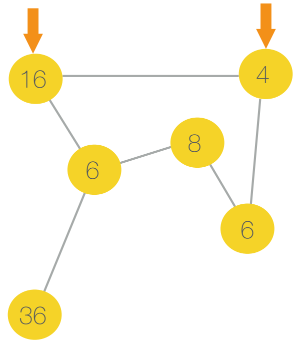
Aggregation
Averaging example



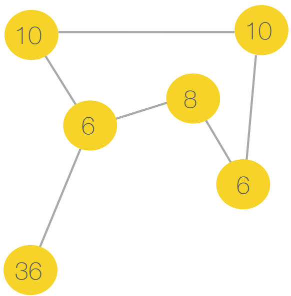
Aggregation
Averaging example



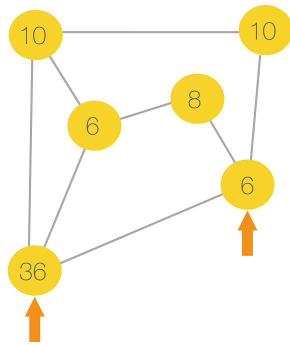
Aggregation
Averaging example



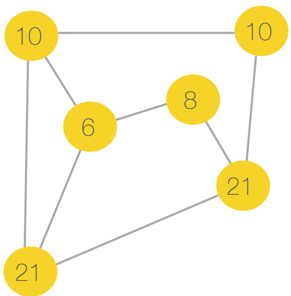
Aggregation
Averaging example



Aggregation
Averaging example

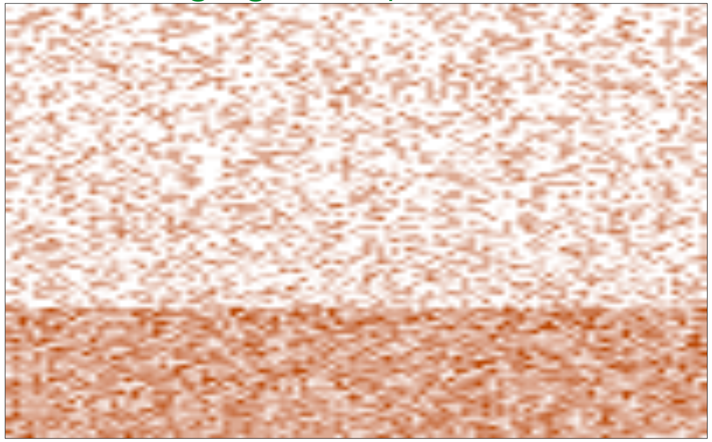


Aggregation
Averaging example

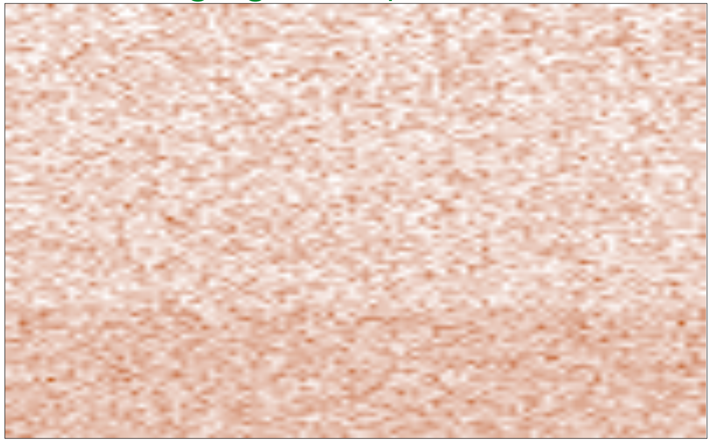


Aggregation
Averaging example: Round 0

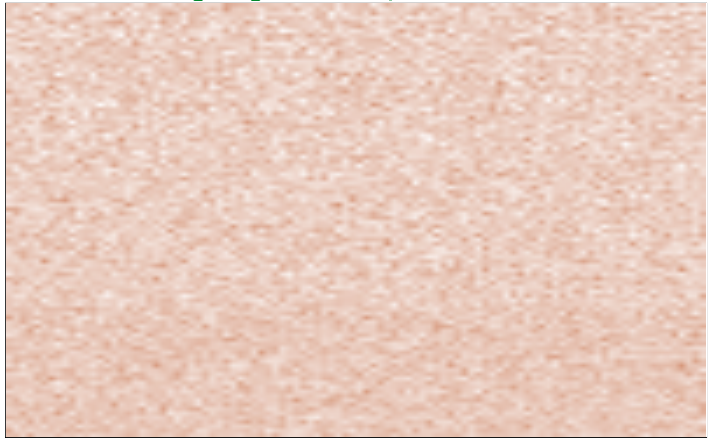
Aggregation
Averaging example: Round 1



Aggregation
Averaging example: Round 2



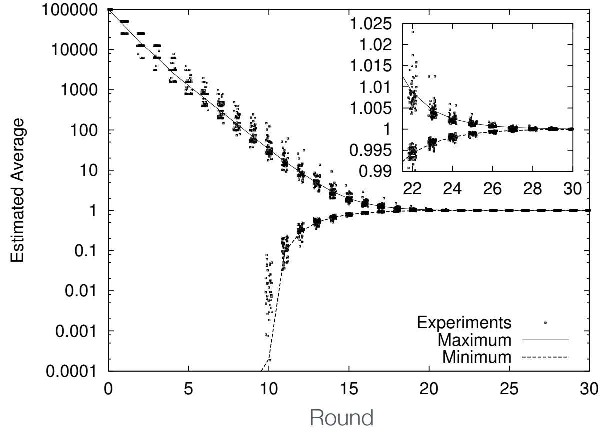
Aggregation
Averaging example: Round 3



Aggregation
Averaging example: Round 4

Aggregation
Averaging example: Round 5

Aggregation
Exponential convergence

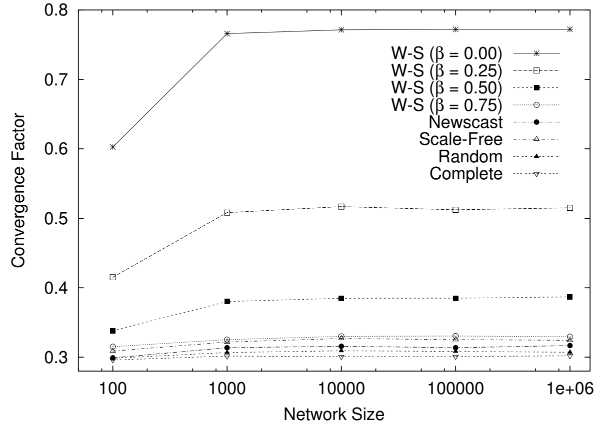


Aggregation
Exponential convergence

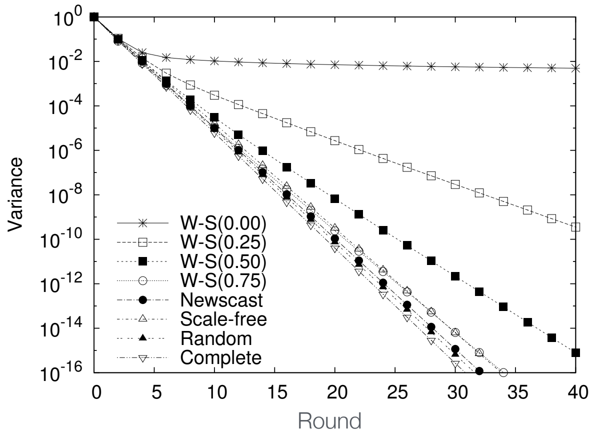
- In gossip-based averaging, if the selected peer is a *globally random* sample, then the expected variance among the estimates decreases exponentially
- Convergence factor:

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

Aggregation
Convergence factor vs topology



Aggregation
Convergence factor vs topology



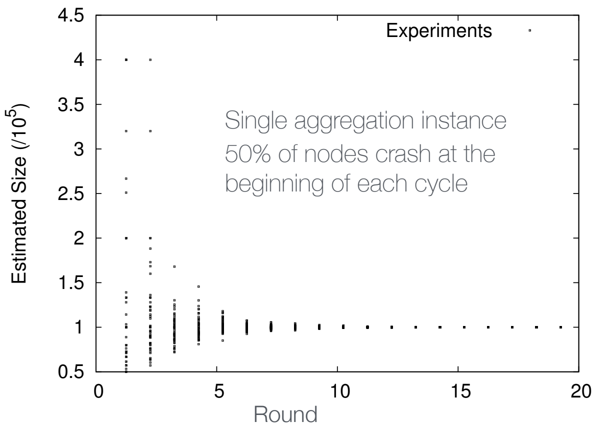
Aggregation
Network size estimation

- How can nodes within a network obtain an estimate for its current size?
- Just “freezing” the network and “counting” does not work
  - Not decentralized
  - Cannot deal with churn
  - Not scalable (think of the Internet)
- Idea: Base the size estimate on an aggregate value that can be computed through a decentralized algorithm
- Compute the arithmetic mean starting from zeroes at all nodes except one that holds 1
- The mean value is $1/n$ and the network size n can be obtained simply by taking the inverse of the mean

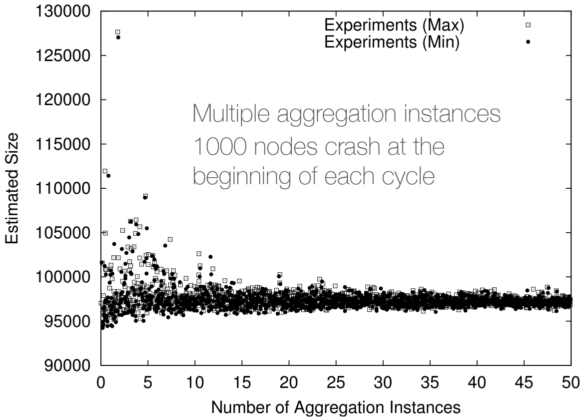
Aggregation
Network size estimation

- Idea works if there are no failures — no nodes fail and no messages are lost
- The estimate will be poor if failures occur during the early phases of the algorithm when the variance is greater
- Failures become less disruptive in later phases of the algorithm
- Worst-case failure scenario: the node with local value 1 fails immediately before exchanging local value with any node
- Idea: start multiple instances of the algorithm with different nodes holding the initial 1 value in each instance and average the results of the different instances to obtain the final estimate

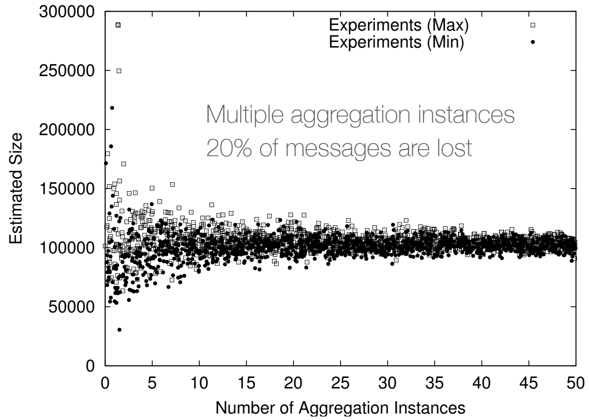
Aggregation
Robustness — node failures



Aggregation
Robustness — node failures



Aggregation
Robustness — communication



Aggregation
Robustness — churn