Gossip-based computation of Aggregate Information

An Evaluation on different network topologies

Basic Idea

- Node Aggregation problems
  - In a network of n nodes, each node i holds a value xi (or a set Mi of values)
  - The goal is to compute some aggregate function of these values (such as sums, averages, quantiles, etc.) in a decentralized and fault-tolerant fashion

- A simple and natural protocol Push-Sum for computing sums or averages of values at the nodes of a network.
  - At all times t, each node i maintains a sum s_t_i, initialized to s_0_i := xi, and a weight w_t_i, initialized to w_0_i := 1.
  - At time 0, it sends the pair (s_0_i;w_0_i) to itself, and in each subsequent time step t, each node i follows the following protocol

Protocol Push-Sum

Algorithm 1 Protocol Push-Sum

1: Let $\{(\hat{s}_r, \hat{w}_r)\}$ be all pairs sent to i in round $t-1$
2: Let $s_{t,i} := \sum_r \hat{s}_r$, $w_{t,i} := \sum_r \hat{w}_r$
3: Choose a target $f_t(i)$ uniformly at random
4: Send the pair $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ to $f_t(i)$ and i (yourself)
5: $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step t

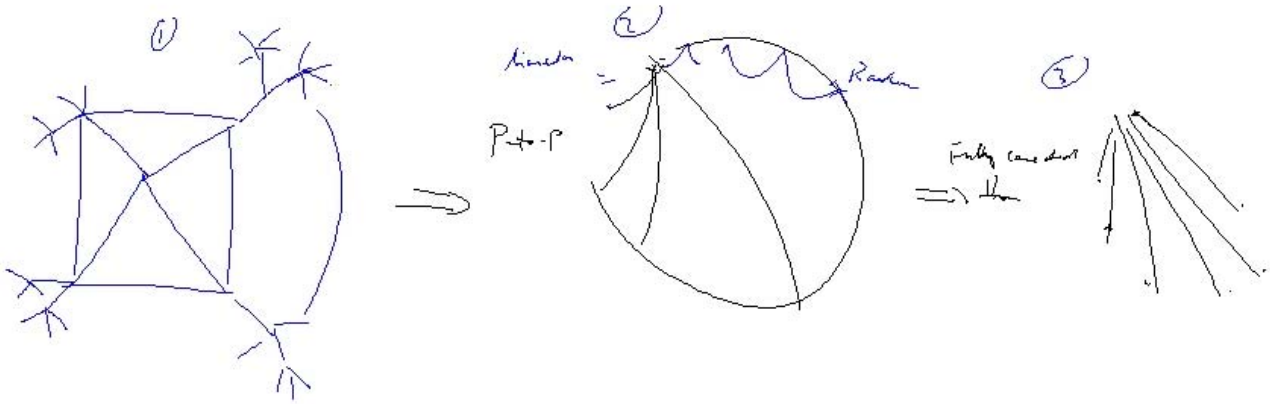
Protocol Push-Sum

- The Theoretical Result
  - The paper [gossip-focs2003] proves statistically that the aggregate values computed by the nodes in the system converge exponentially fast to the true value
  - It also makes some probabilistic guarantees about how good the approximation is

Evaluation on network topologies

- The following types of network topologies were considered for this experiment
  - Internet topology
  - P2P overlays
  - Fully Connected overlay network (using P2P overlays)
- Diagrammatically ..

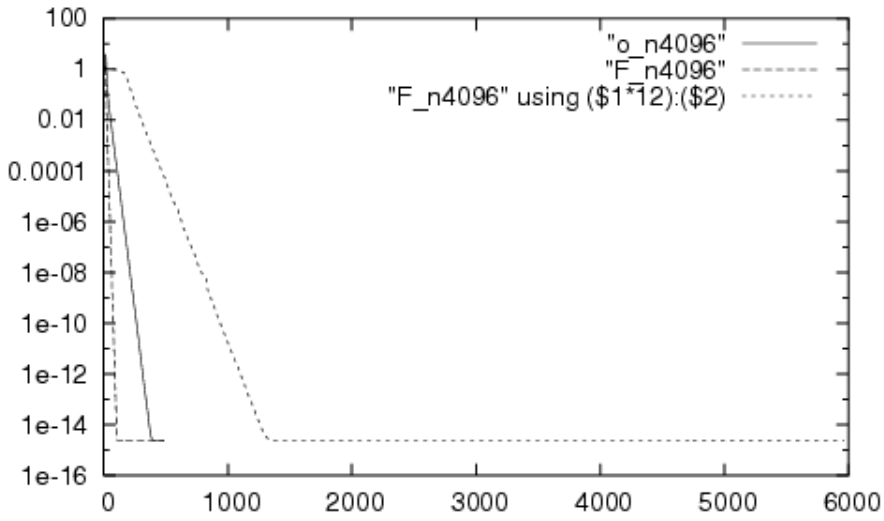
Evaluation on network topologies

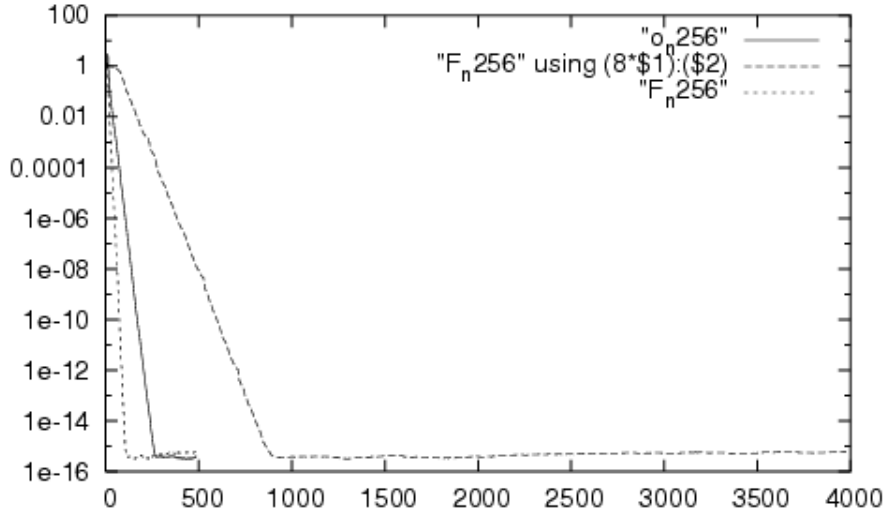


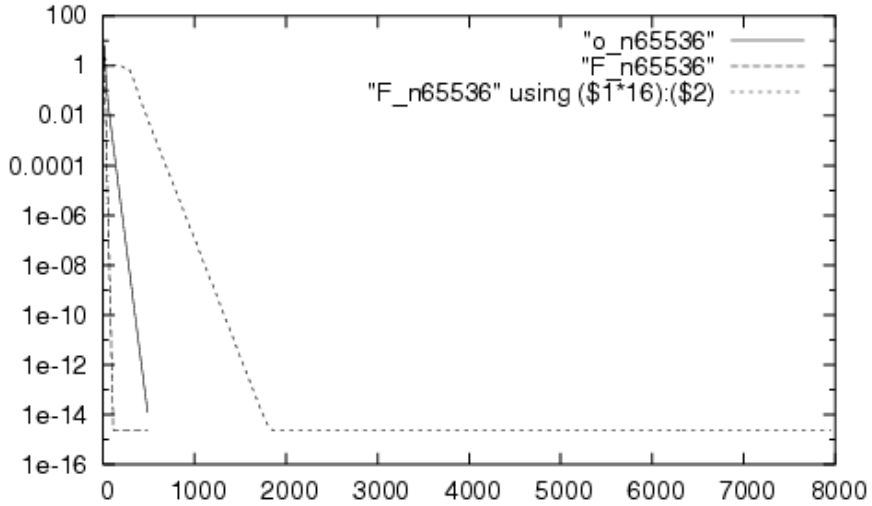
Network Topologies

- The P2P overlay network is maintained by keeping routing information for a few "randomly" selected neighbors for each node
- The fully connected overlay will need storage space in the routing table for ALL the nodes in the network
- The fully connected overlay can be simulated using a P2P overlay. The minimum number of additional messages sent in this case would be (log N)

Experimental results







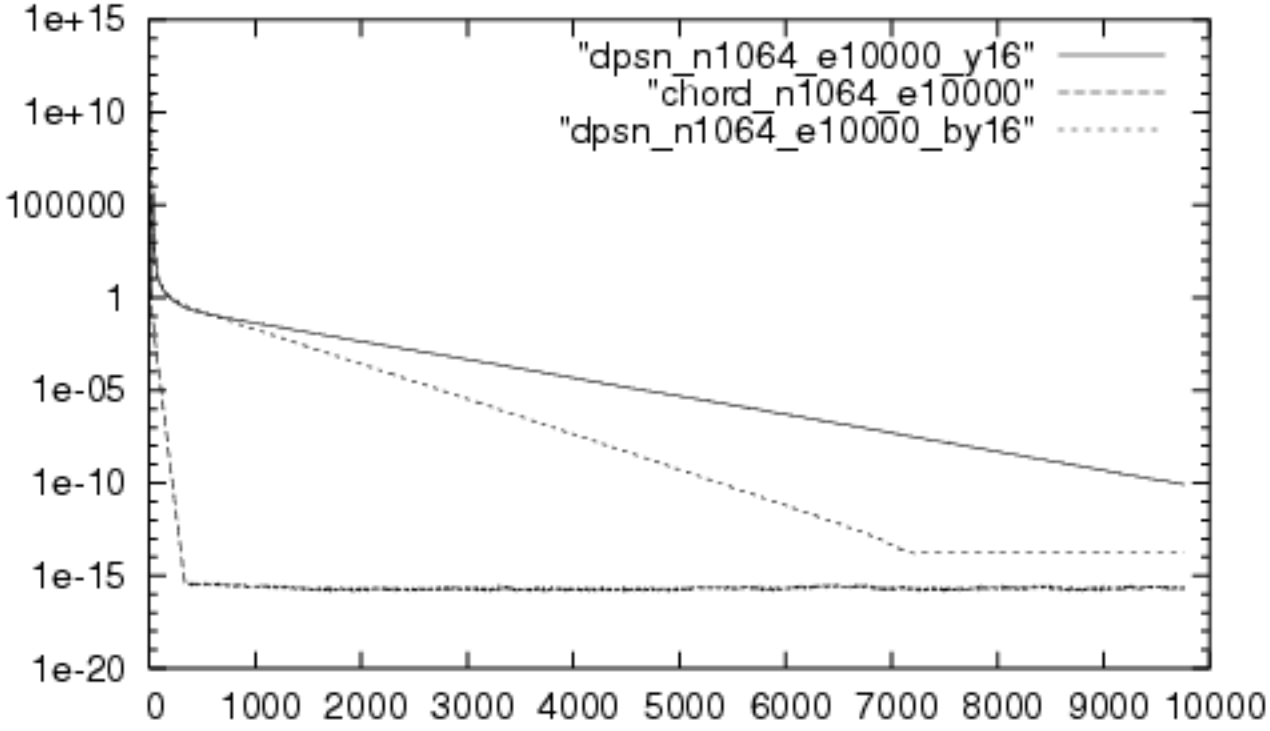
Observations

- The convergence to true result is exponential
- Is scalable
- The Chord network converges faster to the true result
  - Even assuming a lower-bound factor of (log N) for the number of messages required to implement fully connected overlays using P2P overlays (chord network here)

Sensor Networks

- Dynamic planar sensor networks
- Intuition
  - In a sensor network, each sensor can regulate its radio broadcast power to encompass a constant number or fixed percentage of the total sensors as its neighbors
  - Neighbors here imply the list of nodes that a node can directly talk to

Comparative study

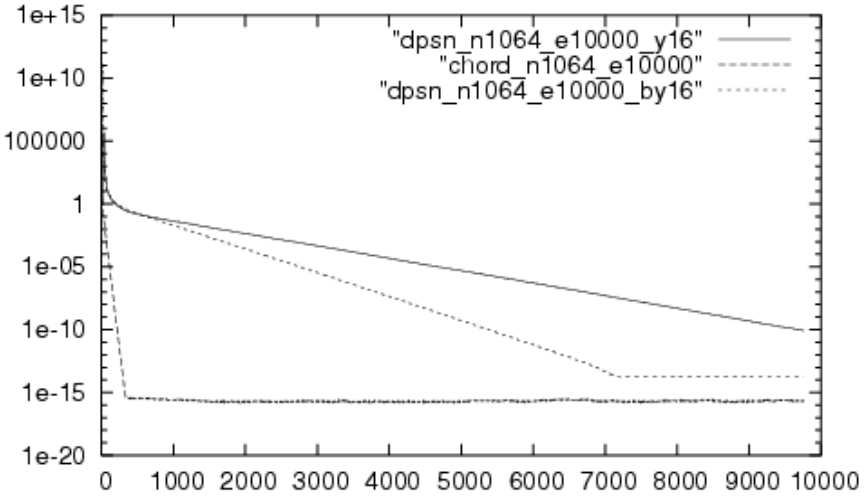


Comparative study

- The cases studies here compare the convergence rate for a dynamic planar sensor network and a chord (P2P) network
  - in each case a node talks to same number of nodes
  - in dpsn all are geographical neighbors (within a dynamically ranged radius)
  - in a chord, each neighbor is at an exponentially increasing distance (distance here is the (pseudo) random number assigned to each node)

Comparative study

- In the DPSN, the aggregate still does converge but it takes a lot more time
- Instead, in case of the chord network, it converges faster.
  - Comparison is based on approximately same number of neighbors



Conclusion

- It is better to use P2P overlays over Internet topology
  - Simulation of fully connected network using P2P overlays results in degradation of performance
- The algorithm converges slowly for dynamic planar sensor networks
  - Selection of uniformly random neighbors is better than using Euclidean distance as a measure of determining neighbors

Research directions

- Approximation Knowledge
  - How does a node know that it has a stable value which is close to the true value?
  - And how close?
  - Can it do this without having any knowledge of the network topology?

Research directions

- Vector of independent aggregations
  - What if we don't always gossip a value?
  - i.e. depending on the node we are talking to, that node might not be participating in THIS aggregation, and so in THIS time epoch, we do not speak to any node regarding this value
  - What effect does this have on the diffusion speed?

Research directions

- Boosting
  - Will it help if nodes not “contributing” in an aggregation “participate” in it?
  - How?
- Stream/Channel selection
  - Aggregation of different streams
  - Subscription of nodes to different channels