Distributed Partial Inference Under Churn

Sriharsha Gangam, Sonia Fahmy
Purdue University

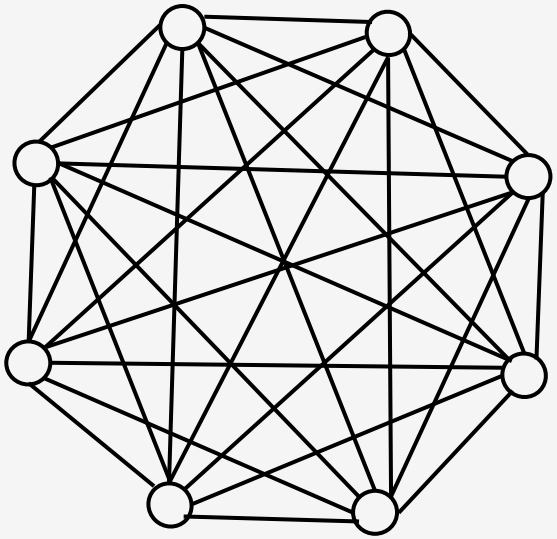
http://www.cs.purdue.edu/homes/fahmy/

This work is sponsored in part by BBN Technologies and by HP.

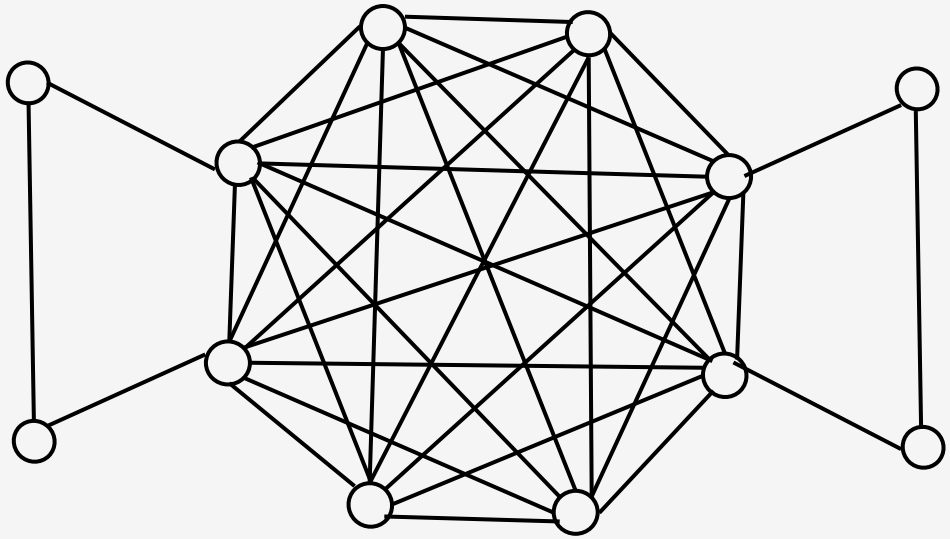
PURDUE UNIVERSITY

A Measurement Service…

- Accepts measurement requests from applications
  - Scalable Sensing Service (***S*^{3}**), ScriptRoute, iPlane
  - Example: Azureus BitTorrent
- Measurement request graph
  - A measurement request between two nodes is an edge



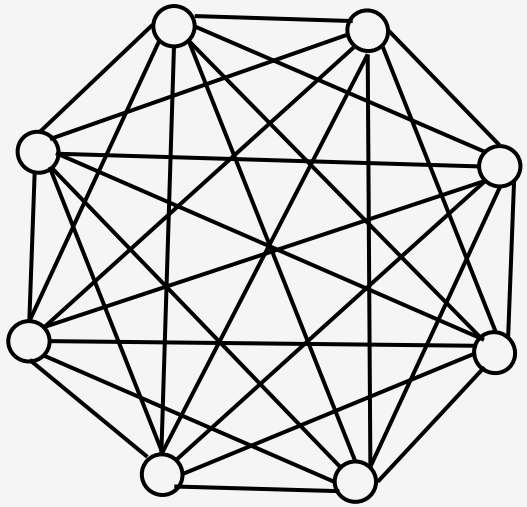
Number of measurements = |E| = 28



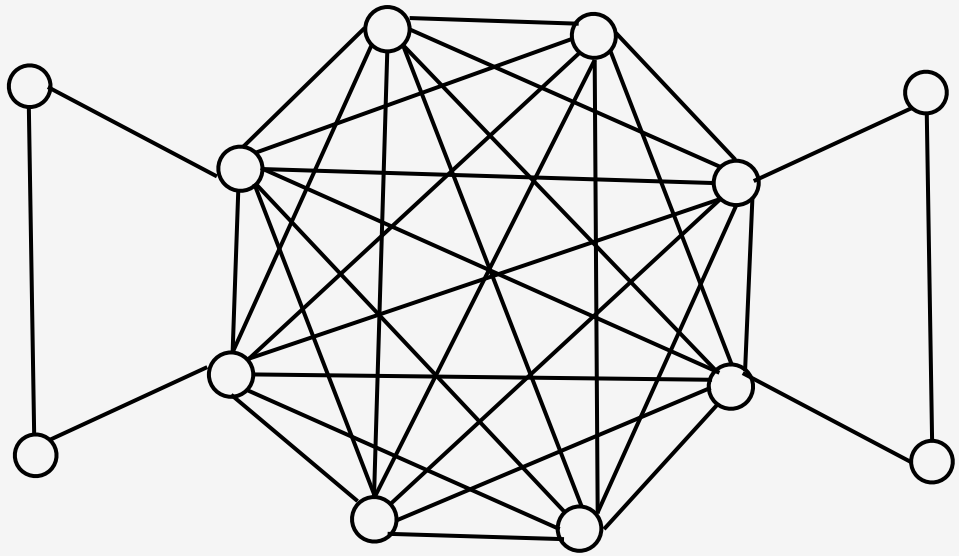
Number of measurements = |E| = 34

Inference Mechanisms…

- May be employed by measurement services
- For *N* nodes, an inference mechanism takes ***kN*** measurements on average
  - Result: All pair end-to-end measurement data
  - Vivaldi $k \approx 32$; GNP $k \approx 15$
- Example: $k = 3$



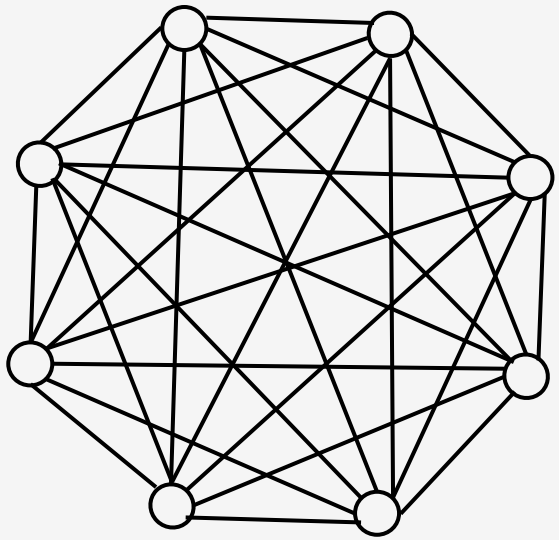
Number of measurements = |E| = 28
Using inference = k|V| = 3x8 = 24



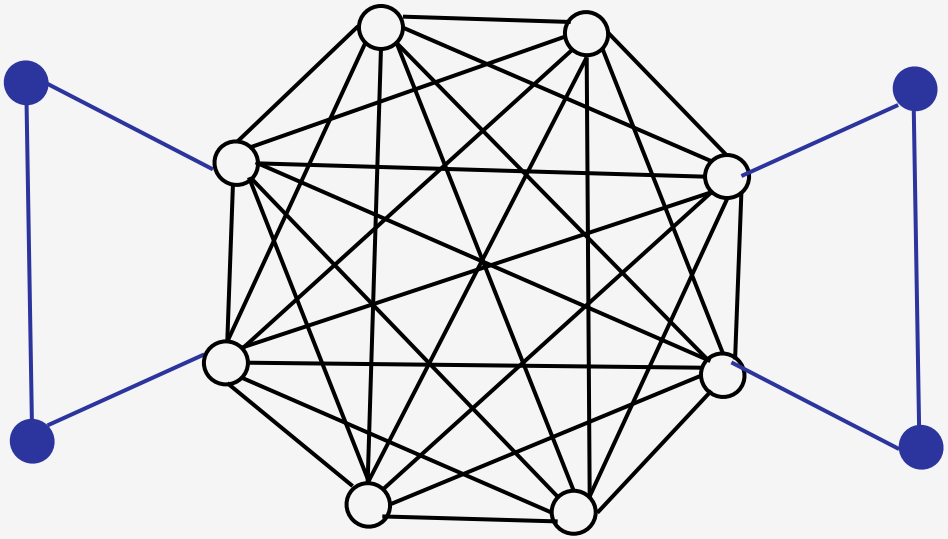
Number of measurements = |E| = 34
Using inference = 3x12 = 36

Partial Inference

1. We do not always need all pair end-to-end measurement data
2. *k* may be large
3. Inference reduces accuracy



Number of measurements = |E| = 28
Using inference = k|V| = 3x8 = 24
Inference over all vertices is good



Number of measurements = |E| = 34
Using inference = 3x12 = 36
Using inference only among white vertices = 6 + 3x8 = 30 < 34
6 direct measurements benefit from higher accuracy

This Work…

- Goals
  - Make best use of inference mechanisms
  - Reduce the number of measurements
  - Improve accuracy when possible.

- We determine:
  - When to use partial inference?
  - How to handle churn in measurement request graph?

Problem

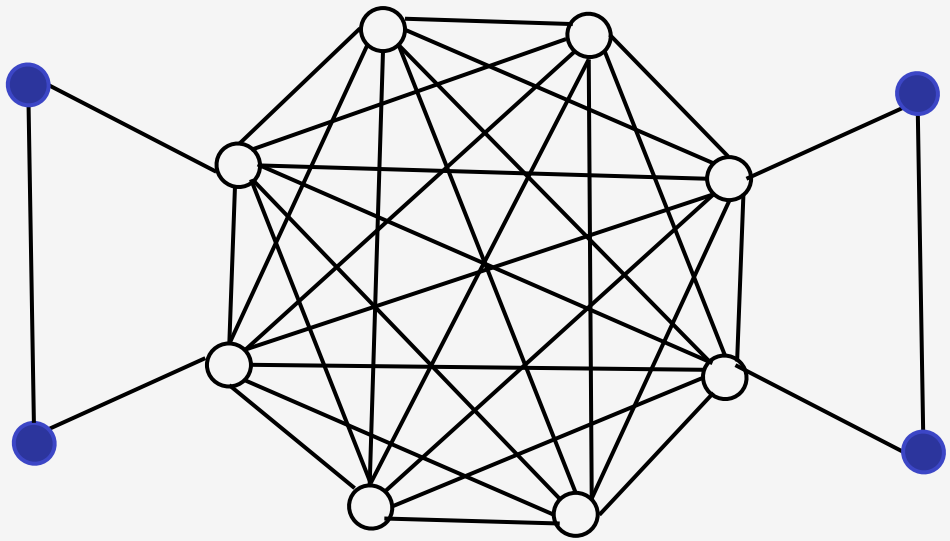
- Given a measurement request graph $G(V,E)$, identify a sub-graph $G_i(V_i,E_i)$ such that the number of measurements taken M is minimized: $M = k|V_i| + (|E| - |E_i|)$
  - $k|V_i|$ = Measurements taken by nodes in inference mechanism
  - $(|E| - |E_i|)$ = Direct measurements

- NP-hard problem [Blanton09]

- E. Blanton, S. Fahmy, G. Frederickson, *On the Utility of Inference Mechanisms*, In Proc. of IEEE ICDCS, June 2009.

k-Core Algorithm

- ***Intuition:*** A vertex with degree ≤ *k* should take direct measurements
- Repeatedly remove all vertices with degree ≤ *k* from the measurement request graph
- Only the remaining vertices participate in inference

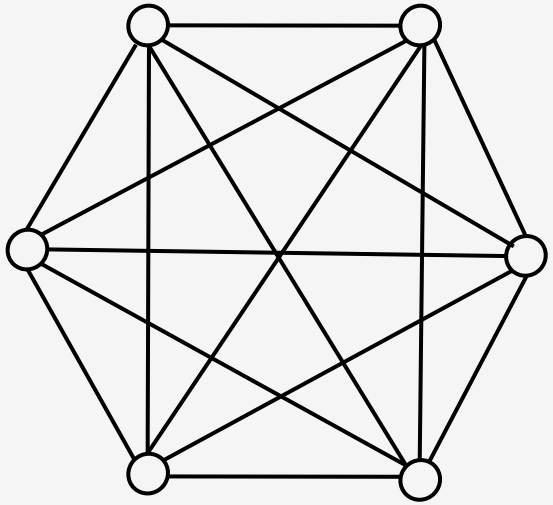


Suppose *k*=*3*
Blue vertices have degree ≤ *3* and are removed
k-Core identifies white vertices for inference

When is k-Core Sub-optimal?

Every vertex has a degree > k and average degree of the sub-graph < 2k

k=3
Optimal:15
k-Core:18



Theorem: Let M_o be the optimal total number of measurements value for a given measurement request graph $G(V, E)$ and inference parameter k. Let M_c be the number of measurements value using the k−Core algorithm. *Then* $M_o \leq M_c < 2M_o$

Distributed k-Core Algorithm

- Without churn and with initial states as inference for all nodes, it behaves like the k-Core
- Needs to react quickly to churn (change in measurement request graph)
- Each node decides for itself if it should participate in inference.
  - Scalable solution

Distributed k-Core Algorithm

- Each node maintains:
  - State: *Inference* or *Direct*
  - State and degree of neighbor vertices
- Main Idea: Participate in inference if either
  - Number of neighbors participating in inference > k
  - There are more than 2k neighbors with degree > 2k
- The second condition provides stability under churn
- State information (in the order of a few bits) is exchanged among nodes

Message Passing

Nodes exchange messages:

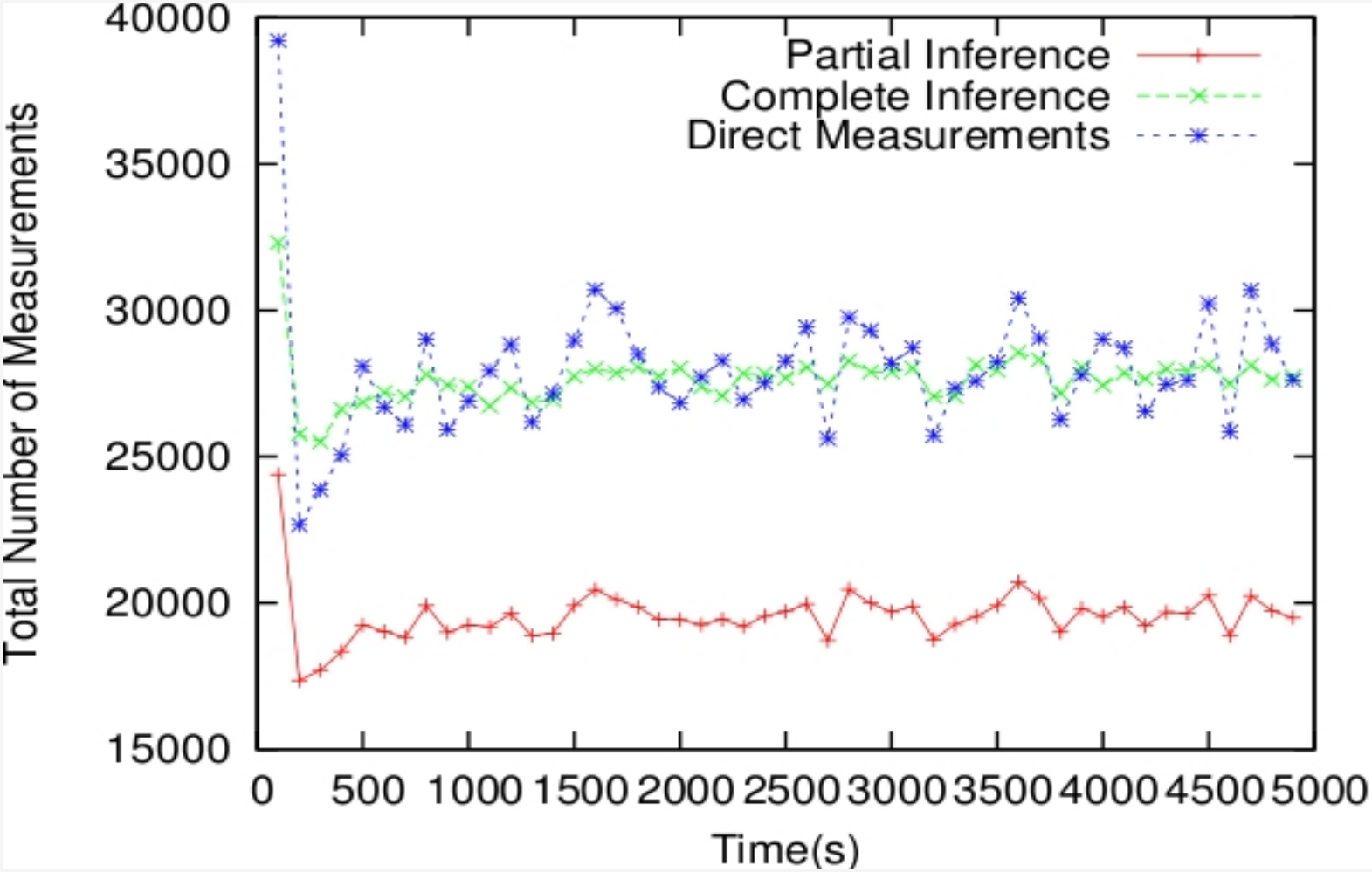
- Signaling based
  - Send messages as soon as the topology changes
  - Effectively deal with churn; more accurate
  - Expensive
- Time step based
  - Periodically contact neighbors at regular intervals
  - Less responsive to churn; hence less accurate
  - Less expensive
  - How to select time step?

Evaluation: Parameters

- Inference parameter *k* (3-60)
- Measurement request graphs
  - Graphs representing peers connections in UUSee peer to peer streaming service |V|=2500, |E|=53000
- Synthetic churn
  - Poisson arrivals
    - Mean inter-arrival time per node 10-130 s
  - Pareto staying times
    - Min staying time 40-120 s
    - Pareto shape parameter 0.5-2.5
- Experimental churn
  - Arrival and departure of Skype peers
- Delays between nodes
  - A subset of the MIT king latency dataset (mean RTT 133 ms)

Number of Measurements

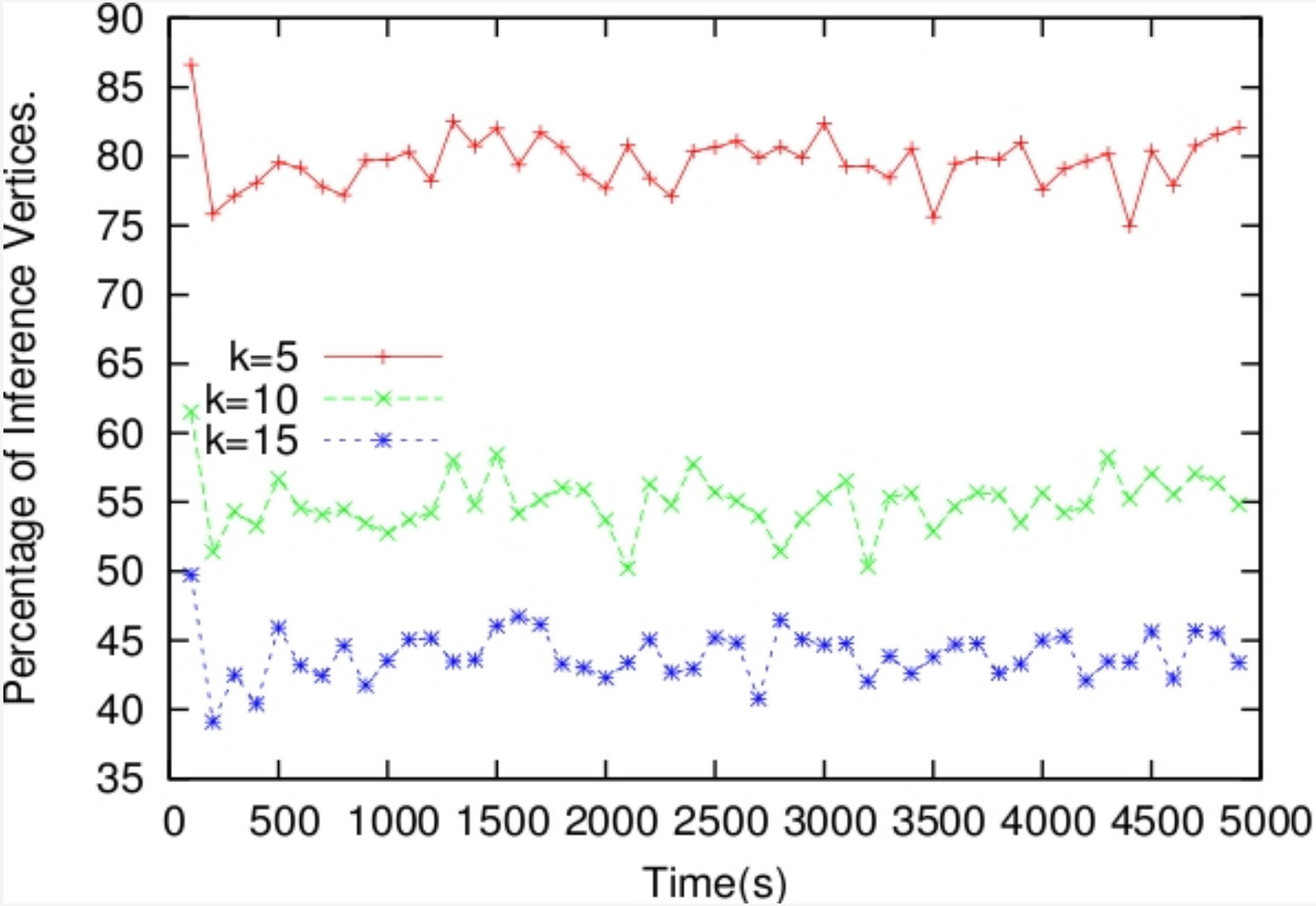
Partial inference is better than complete inference or direct measurements. k=15



PURDUE UNIVERSITY

Extent of Inference

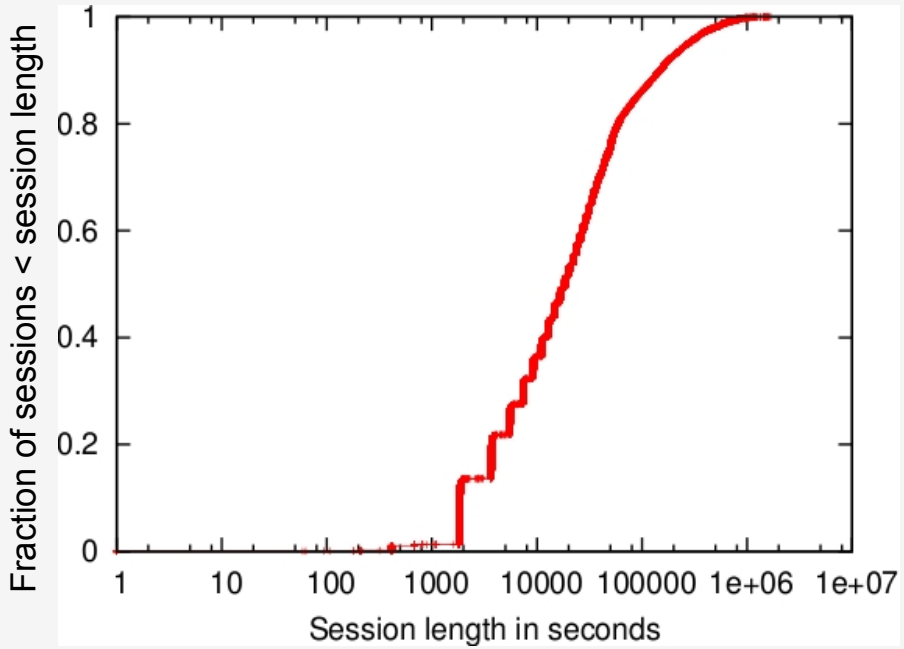
- Higher $k \rightarrow$ Less inference
- Lower $k \rightarrow$ More inference

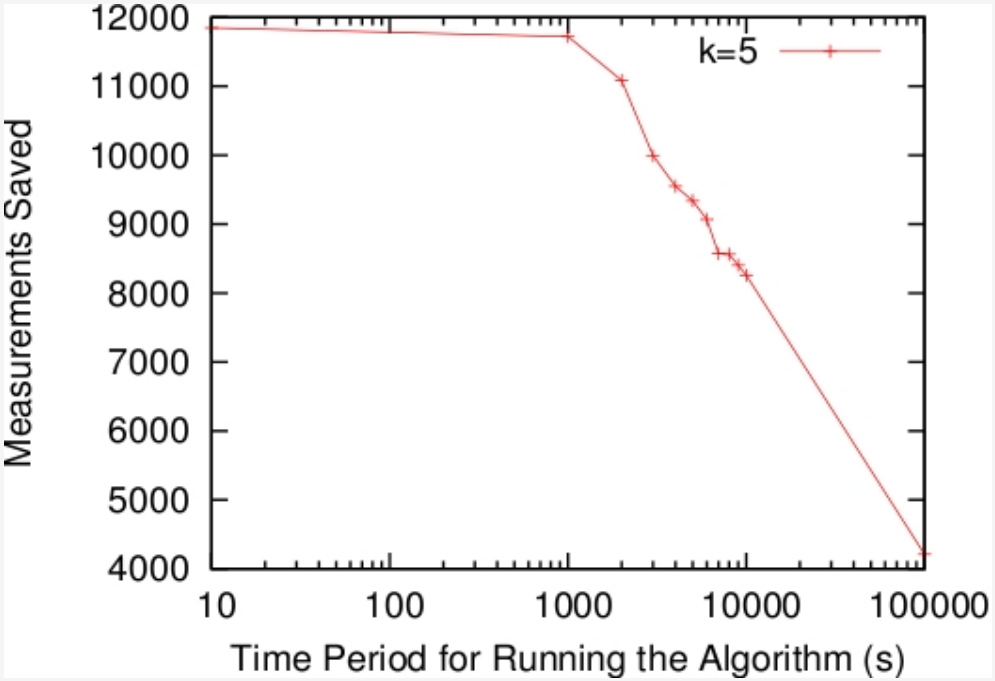


PURDUE UNIVERSITY

What Time Step to Use?

- Synthetic churn: Performance decreases with increasing time step
- Experimental churn: Session length of Skype peers is a good estimate

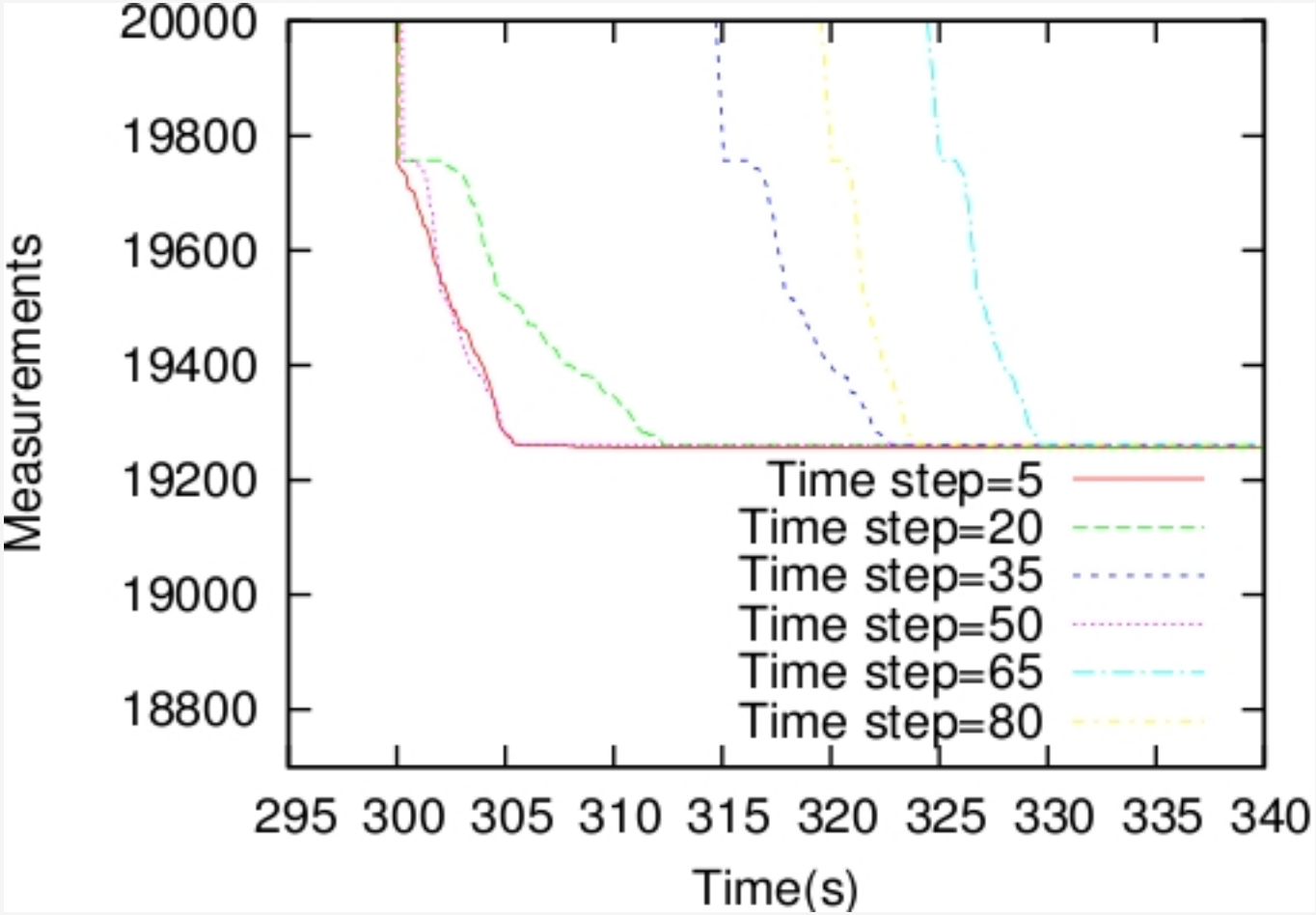




PURDUE UNIVERSITY

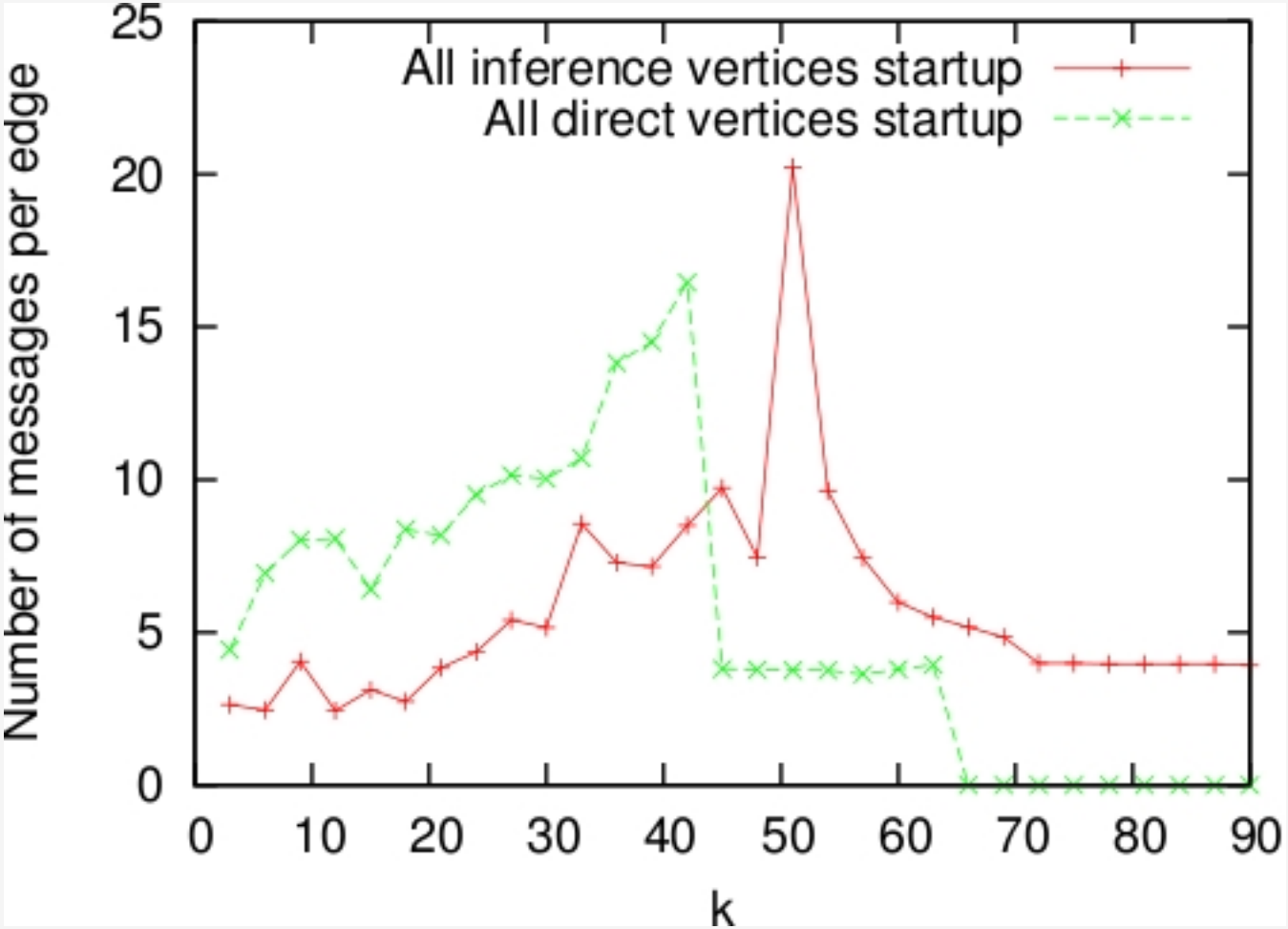
Convergence Time

- At t=300 s, 1/4th nodes leave
- Signaling based messaging: 2-3 seconds to converge
- Time step based messaging: About the same as time step



Messages to Converge

Initially states are all inference or all direct



PURDUE UNIVERSITY

Summary

- Identifying the nodes suitable for inference is an NP-hard graph problem
- A simple approximate solution exists → k-Core algorithm
  - Measurements taken are less than twice the optimal
- Distributed k-Core
  - Scalable solution that works well under churn
- We *save* measurements by using partial inference
  - Depends on *k*, measurement request graph, and churn
  - Time step based on session/staying time

Thank You

Questions?

PURDUE UNIVERSITY