On the Influence of Graph Density on Randomized Gossiping

Robert Elsässer Dominik Kaaser

May 27, 2015

UNIVERSITÄT SALZBURG

The Random Phone Call Model

A Simple Communication Model

Model

- n players, connected in a communication network
- parallel, synchronous rounds
- select a communication partner uniformly at random from neighbors
- bi-directional communication over this channel

The Random Phone Call Model

A Simple Communication Model

Model

- n players, connected in a communication network
- parallel, synchronous rounds
- select a communication partner uniformly at random from neighbors
- bi-directional communication over this channel

Problem

- information dissemination in this communication network

The Random Phone Call Model

A Simple Communication Model

Model

- n players, connected in a communication network
- parallel, synchronous rounds
- select a communication partner uniformly at random from neighbors
- bi-directional communication over this channel

Problem

- information dissemination in this communication network

Goal

- time-efficient algorithms
- small communication overhead

Motivation

Information Dissemination

- maintenance of replicated databases
  - propagation of updates at individual nodes
- peer-to-peer networks
  - efficient, simple, and robust broadcasting algorithms for overlays
- exascale computing

Motivation

Information Dissemination

- maintenance of replicated databases
  - propagation of updates at individual nodes
- peer-to-peer networks
  - efficient, simple, and robust broadcasting algorithms for overlays
- exascale computing

Motivation

Information Dissemination

- maintenance of replicated databases
  - propagation of updates at individual nodes
- peer-to-peer networks
  - efficient, simple, and robust broadcasting algorithms for overlays
- exascale computing

Motivation

Information Dissemination

- maintenance of replicated databases
  - propagation of updates at individual nodes
- peer-to-peer networks
  - efficient, simple, and robust broadcasting algorithms for overlays
- exascale computing

Motivation

Information Dissemination

- maintenance of replicated databases
  - propagation of updates at individual nodes
- peer-to-peer networks
  - efficient, simple, and robust broadcasting algorithms for overlays
- exascale computing

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides robustness, simplicity, and scalability.

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides **robustness**, simplicity, and scalability.

- robustness
  - comparison with deterministic communication schemes
  - single node failure results in many uninformed nodes
  - require either substantially more time or tolerate only a small number of node failures
  - random phone-call model: F node failures – $O(F)$ uninformed players

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides **robustness**, simplicity, and scalability.

- robustness
  - comparison with deterministic communication schemes
  - single node failure results in many uninformed nodes
  - require either substantially more time or tolerate only a small number of node failures
  - random phone-call model: F node failures – $O(F)$ uninformed players

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides **robustness**, simplicity, and scalability.

- robustness
  - comparison with deterministic communication schemes
  - single node failure results in many uninformed nodes
  - require either substantially more time or tolerate only a small number of node failures
  - random phone-call model: F node failures – $O(F)$ uninformed players

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides **robustness**, simplicity, and scalability.

- robustness
  - comparison with deterministic communication schemes
  - single node failure results in many uninformed nodes
  - require either substantially more time or tolerate only a small number of node failures
  - random phone-call model: F node failures – $O(F)$ uninformed players

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides robustness, **simplicity**, and **scalability**.

- simplicity and scalability
  - do not need to maintain communication structures and fail-safe procedures
  - lightweight clients can be used
  - algorithm easy to describe
  - no bottleneck nodes

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides robustness, **simplicity**, and **scalability**.

- simplicity and scalability
  - do not need to maintain communication structures and fail-safe procedures
  - lightweight clients can be used
  - algorithm easy to describe
  - no bottleneck nodes

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides robustness, **simplicity**, and **scalability**.

- simplicity and scalability
  - do not need to maintain communication structures and fail-safe procedures
  - lightweight clients can be used
  - algorithm easy to describe
  - no bottleneck nodes

Randomized Communication

Robustness, Simplicity, and Scalability

- Randomized communication provides robustness, **simplicity**, and **scalability**.

- simplicity and scalability
  - do not need to maintain communication structures and fail-safe procedures
  - lightweight clients can be used
  - algorithm easy to describe
  - no bottleneck nodes

Information Dissemination

The Broadcasting Problem

- one to all communication

- distribute one message to all other nodes in the network
- efficient algorithms for complete graphs

  Karp et al., FOCS 2000: $O(\log n)$ *time,* $O(n \log \log n)$ *messages for Broadcasting*

- do not extend to sparse graphs

Information Dissemination

The Broadcasting Problem

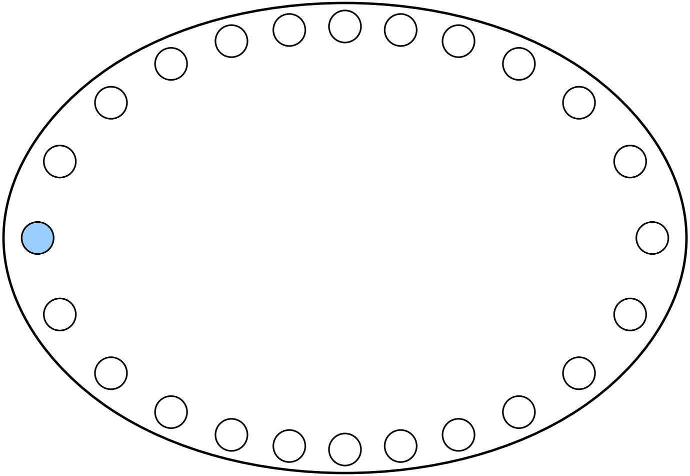
- one to all communication

- distribute one message to all other nodes in the network
- efficient algorithms for complete graphs

  Karp et al., FOCS 2000: $O(\log n)$ *time,* $O(n \log \log n)$ *messages for Broadcasting*
- do not extend to sparse graphs

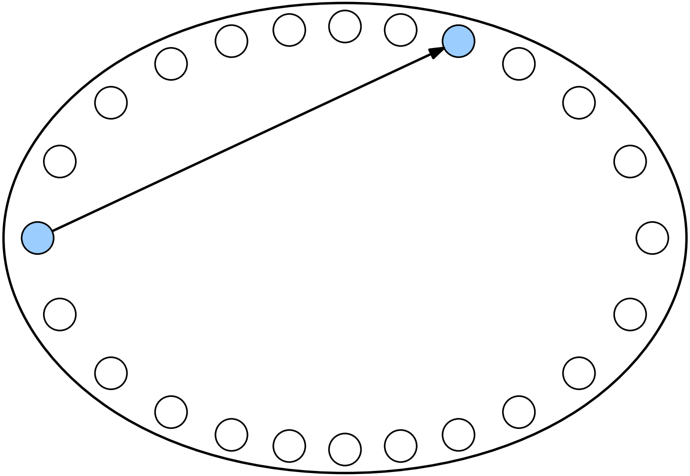
Animation

Broadcasting



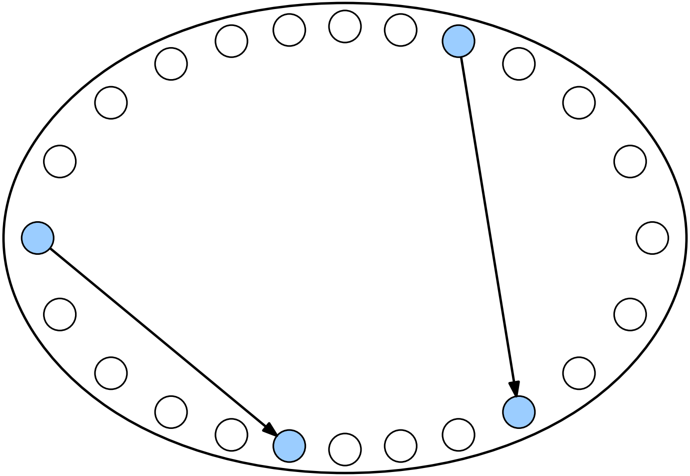
Animation

Broadcasting



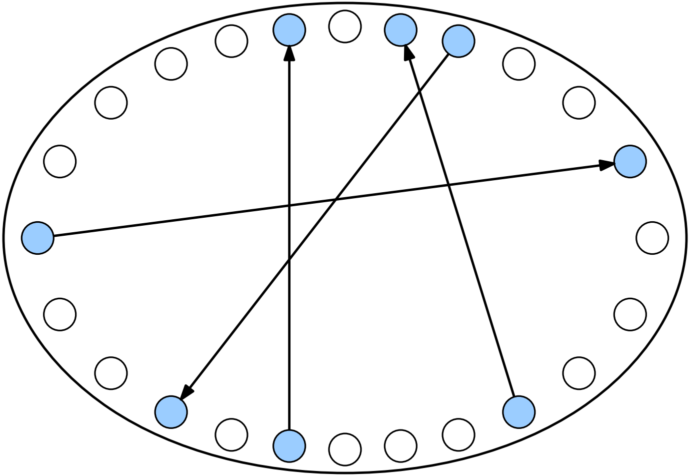
Animation

Broadcasting



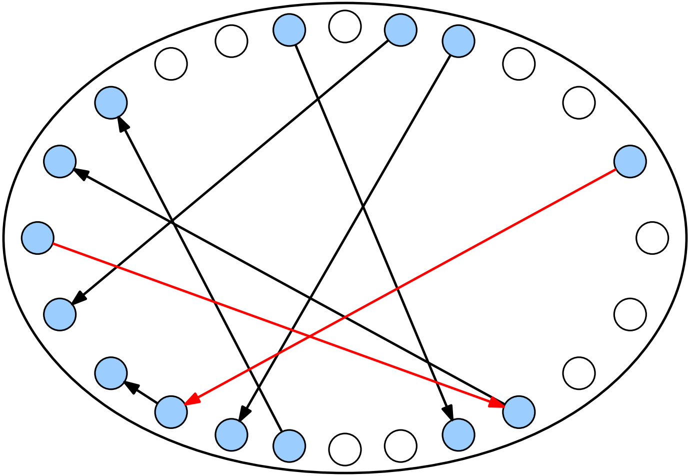
Animation

Broadcasting



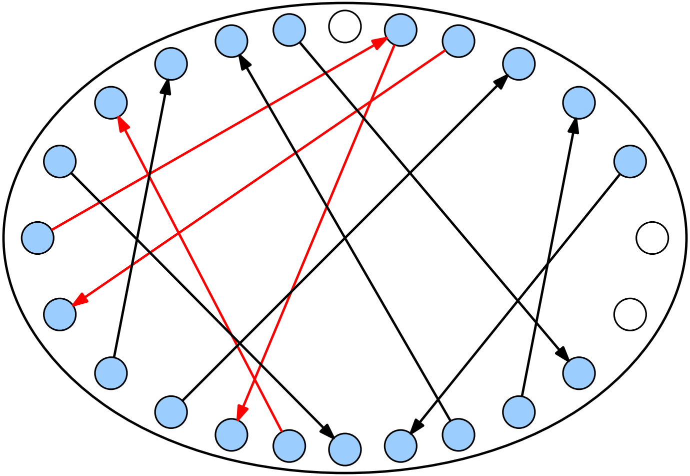
Animation

Broadcasting



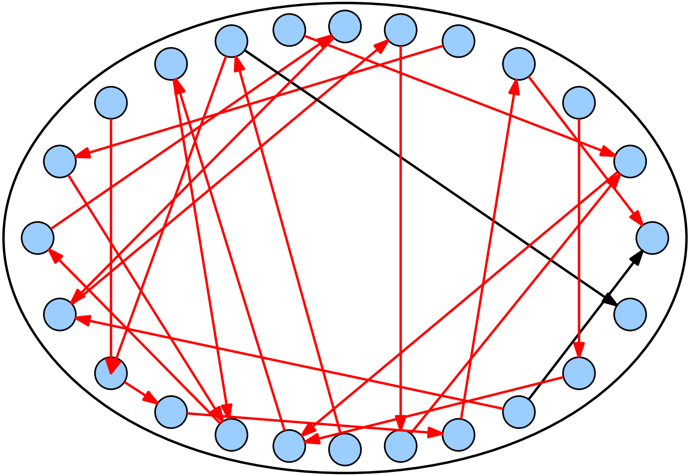
Animation

Broadcasting



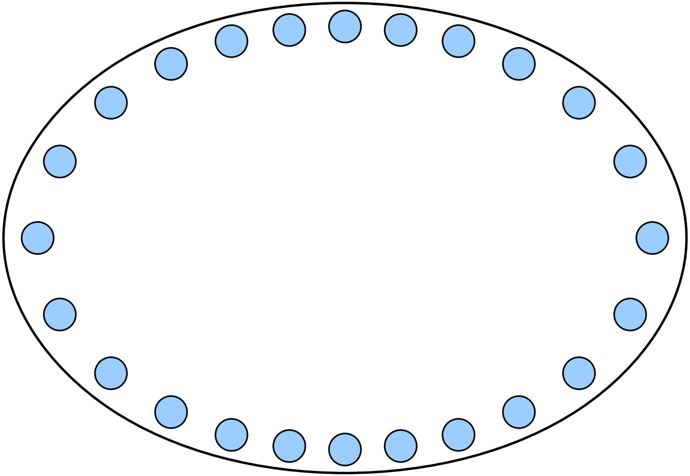
Animation

Broadcasting



Animation

Broadcasting



Information Dissemination

The Broadcasting Problem

- one to all communication

- distribute one message to all other nodes in the network
- efficient algorithms for complete graphs

  Karp et al., FOCS 2000: $O(\log n)$ time, $O(n \log \log n)$ messages for Broadcasting

- do not extend to sparse graphs

Information Dissemination

The Broadcasting Problem

- one to all communication

- distribute one message to all other nodes in the network
- efficient algorithms for complete graphs

  Karp et al., FOCS 2000: $O(\log n)$ *time,* $O(n \log \log n)$ *messages for Broadcasting*
- do not extend to sparse graphs

Information Dissemination

The Gossiping Problem

- all to all communication

- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ time, $\Omega(n \log n)$ messages for Gossiping

- algorithms can be extended to sparse graphs

- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Information Dissemination

The Gossiping Problem

- all to all communication
- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ time, $\Omega(n \log n)$ messages for Gossiping
- algorithms can be extended to sparse graphs
- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Information Dissemination

The Gossiping Problem

- all to all communication
- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ *time,* $\Omega(n \log n)$ *messages for Gossiping*
- algorithms can be extended to sparse graphs
- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Information Dissemination

The Gossiping Problem

- all to all communication

- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ time, $\Omega(n \log n)$ messages for Gossiping

- algorithms can be extended to sparse graphs

- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Information Dissemination

The Gossiping Problem

- all to all communication

- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ *time,* $\Omega(n \log n)$ *messages for Gossiping*
- algorithms can be extended to sparse graphs

- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Information Dissemination

The Gossiping Problem

- all to all communication

- each node starts with its own initial message
- gossiping cannot be done as efficiently as broadcasting

  Berenbrink et al., ICALP 2010: $O(\log n)$ time, $\Omega(n \log n)$ messages for Gossiping

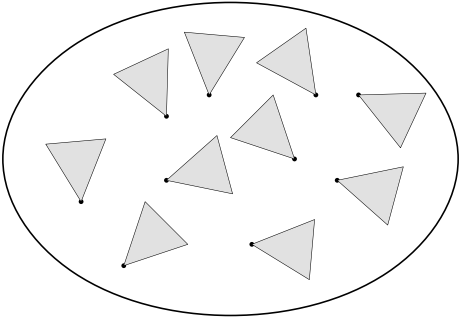
- algorithms can be extended to sparse graphs

- runtime-message complexity trade-off
- $O(\log^2 n)$ time, $O(n \log n / \log \log n)$ messages

Algorithm

The FastGossiping Algorithm

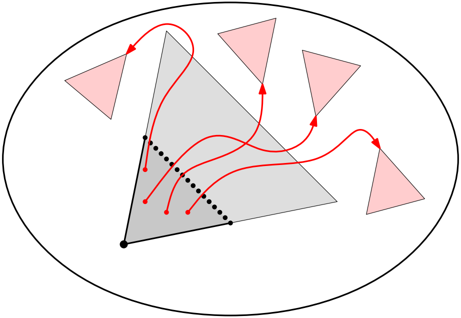
- runs in three phases
  1. distribution procedure
  2. start random walks to collect and distribute messages
  3. broadcasting phase to inform remaining nodes



Algorithm

The FastGossiping Algorithm

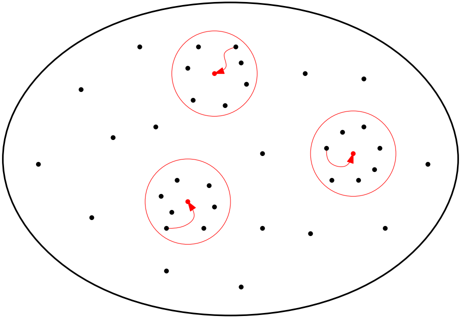
- runs in three phases
  1. distribution procedure
  2. start random walks to collect and distribute messages
  3. broadcasting phase to inform remaining nodes



Algorithm

The FastGossiping Algorithm

- runs in three phases
  1. distribution procedure
  2. start random walks to collect and distribute messages
  3. broadcasting phase to inform remaining nodes



Our Result

Theorem

The gossiping problem can be solved in the random phone call model on a random graph in the configuration model with expected node degree $\Omega\left(\log^{2+\epsilon} n\right)$ *in*

- $O\left(\log^2 n / \log\log n\right)$ *time using*
- $O\left(n \log n / \log\log n\right)$ *message transmissions,*

with high probability.

Analysis

Random Walks Phase

- $O(\log n / \log\log n)$ rounds, $O(\log n)$ steps
- start $\Theta(n/\log n)$ random walks
- run for $\Theta(\log n)$ steps
- The informed set I_m is visited at least $\Omega(|I_m|)$ times.
- It may happen that a single random walk visits I_m multiple times.

Analysis

Random Walks Phase

- $O(\log n / \log\log n)$ rounds, $O(\log n)$ steps
- start $\Theta(n/\log n)$ random walks
- run for $\Theta(\log n)$ steps
- The informed set I_m is visited at least $\Omega(|I_m|)$ times.
- It may happen that a single random walk visits I_m multiple times.

Analysis

Random Walks Phase

- $O(\log n / \log \log n)$ rounds, $O(\log n)$ steps
- start $\Theta(n/\log n)$ random walks
- run for $\Theta(\log n)$ steps
- The informed set I_m is visited at least $\Omega(|I_m|)$ times.
- It may happen that a single random walk visits I_m multiple times.

Analysis

Random Walks Phase

- $O(\log n / \log\log n)$ rounds, $O(\log n)$ steps
- start $\Theta(n/\log n)$ random walks
- run for $\Theta(\log n)$ steps
- The informed set I_m is visited at least $\Omega(|I_m|)$ times.
- It may happen that a single random walk visits I_m multiple times.

Analysis

Random Walks Phase

- $O(\log n / \log \log n)$ rounds, $O(\log n)$ steps
- start $\Theta(n/\log n)$ random walks
- run for $\Theta(\log n)$ steps
- The informed set I_m is visited at least $\Omega(|I_m|)$ times.
- It may happen that a single random walk visits I_m multiple times.

Analysis

Local Structure of Sparse Random Graphs

- Consider only random graphs (configuration model) with node degree $d \le \log^k n$.
- Let v denote an arbitrary but fixed vertex.
- The graph induced by vertices of distance to v at most $O(\log\log n)$ forms *almost* a tree.

Analysis

Local Structure of Sparse Random Graphs

- Consider only random graphs (configuration model) with node degree $d \leq \log^k n$.
- Let v denote an arbitrary but fixed vertex.
- The graph induced by vertices of distance to v at most $O(\log \log n)$ forms *almost* a tree.

Analysis

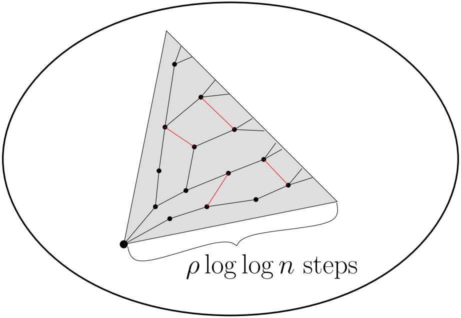
Local Structure of Sparse Random Graphs

- Consider only random graphs (configuration model) with node degree $d \leq \log^k n$.
- Let v denote an arbitrary but fixed vertex.
- The graph induced by vertices of distance to v at most $O(\log \log n)$ forms *almost* a tree.

Analysis

Local Structure of Sparse Random Graphs

- Consider only random graphs (configuration model) with node degree $d \leq \log^k n$.
- Let v denote an arbitrary but fixed vertex.
- The graph induced by vertices of distance to v at most $O(\log \log n)$ forms *almost* a tree.



Analysis

Distinct Random Walks in the Informed Set

Lemma

An arbitrary but fixed random walk leaves the set of informed vertices I_m to a distance of $\Omega(\log \log n)$ and does not return to I_m almost surely.

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ *random walks visit* I_m *at most a constant number of times.*

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$
- Therefore, $|Q| \in \Theta(|I_m|)$.

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$

- We now bound the number of random walks in P.
- Therefore, $|Q| \in \Theta(|I_m|)$. □

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$

- All random walk **steps** are independent from each other!
- Therefore, $|Q| \in \Theta(|I_m|)$.

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$
- We may apply Chernoff bounds
- Therefore, $|Q| \in \Theta(|I_m|)$. □

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$

$$\Pr\left[|P| \geq \left(1 + \frac{1}{\log^c n}\right)\mu\right] \leq e^{-O\left(\frac{n}{\log^{3c} n \cdot 2^{\log n / \log\log n}}\right)} \leq n^{-\alpha} \quad .$$

- Therefore, $|Q| \in \Theta(|I_m|)$. □

Analysis

Distinct Random Walks in the Informed Set

Lemma

$\Theta(|I_m|)$ random walks visit I_m at most a constant number of times.

Proof sketch.

- W: Number of hits in I_m
- Q: set of random walks that visit I_m at most c times
- P: all the other random walks
- $W \leq c \cdot |Q| + \Theta(\log n \cdot |P|)$

$$\Pr\left[|P| \geq \left(1 + \frac{1}{\log^c n}\right)\mu\right] \leq e^{-\mathrm{O}\left(\frac{n}{\log^{3c} n \cdot 2^{\log n / \log\log n}}\right)} \leq n^{-\alpha} .$$

- Therefore, $|Q| \in \Theta(|I_m|)$. □

Analysis

Exponential Growth

- We have $\Theta(|I_m|)$ distinct random walks carrying m.
- Each random walk performs a broadcasting procedure.
- Each round, we gain a growth by a factor of $\sqrt{\log n}$!
- This holds for one message with high probability.

Analysis

Exponential Growth

- We have $\Theta(|I_m|)$ distinct random walks carrying m.
- Each random walk performs a broadcasting procedure.
- Each round, we gain a growth by a factor of $\sqrt{\log n}$!
- This holds for one message with high probability.

Analysis

Exponential Growth

- We have $\Theta(|I_m|)$ distinct random walks carrying m.
- Each random walk performs a broadcasting procedure.
- Each round, we gain a growth by a factor of $\sqrt{\log n}$!

- This holds for one message with high probability.

Analysis

Exponential Growth

- We have $\Theta(|I_m|)$ distinct random walks carrying m.
- Each random walk performs a broadcasting procedure.
- Each round, we gain a growth by a factor of $\sqrt{\log n}$!

- This holds for one message with high probability.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

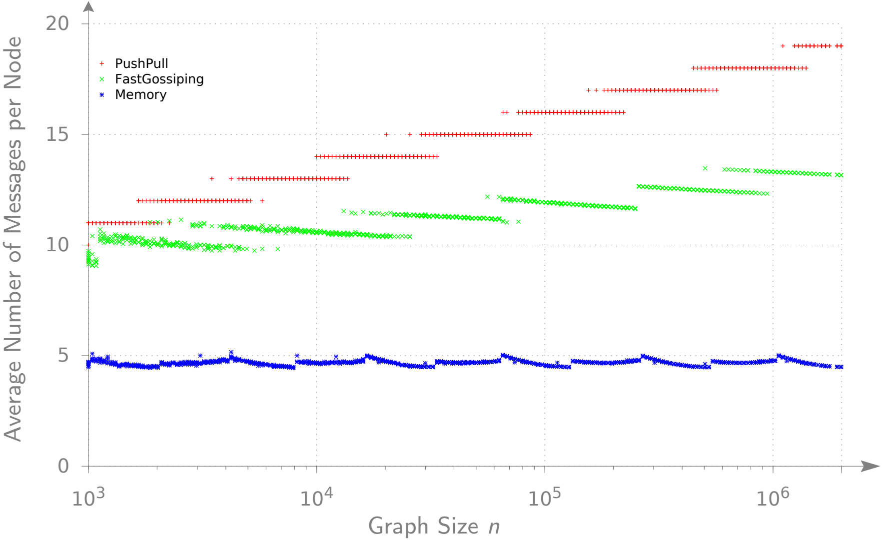
- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

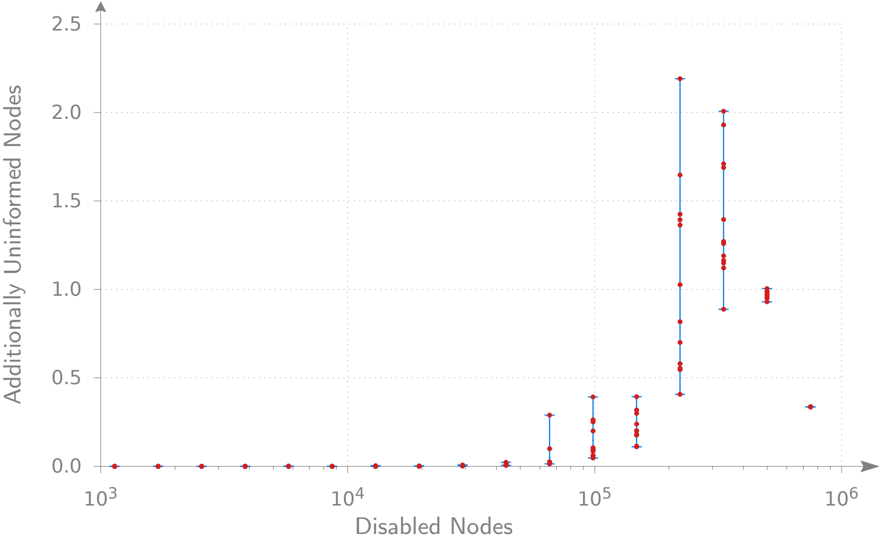
- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.

The Memory Model

- gossiping cannot be done as efficiently as broadcasting
- slightly modify the model
- equip every node with a constant size memory
- store the last 3 communication partners

- The memory can be used to construct a communication tree.
- This requires only a constant number of messages per node...
- provided a leader is known that acts as the tree's root.


PushPull
FastGossiping
Memory
Average Number of Messages per Node
Graph Size n
0
5
10
15
20
10^3
10^4
10^5
10^6


2.5
2.0
1.5
1.0
0.5
0.0
Additionally Uninformed Nodes
10^3
10^4
10^5
10^6
Disabled Nodes

Thank You for Your Attention — Questions Welcome!