Distributed Algorithmic Foundations of Dynamic Networks

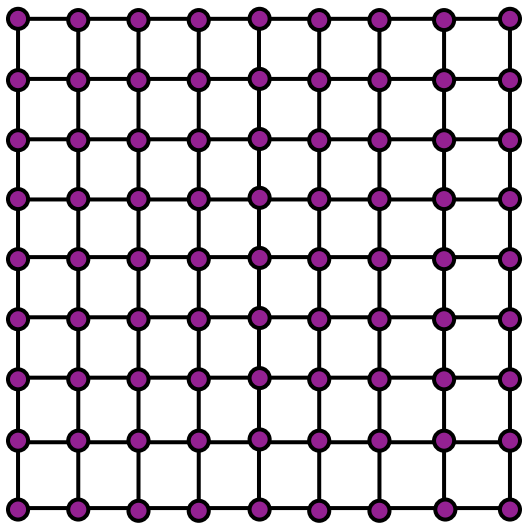
Gopal Pandurangan

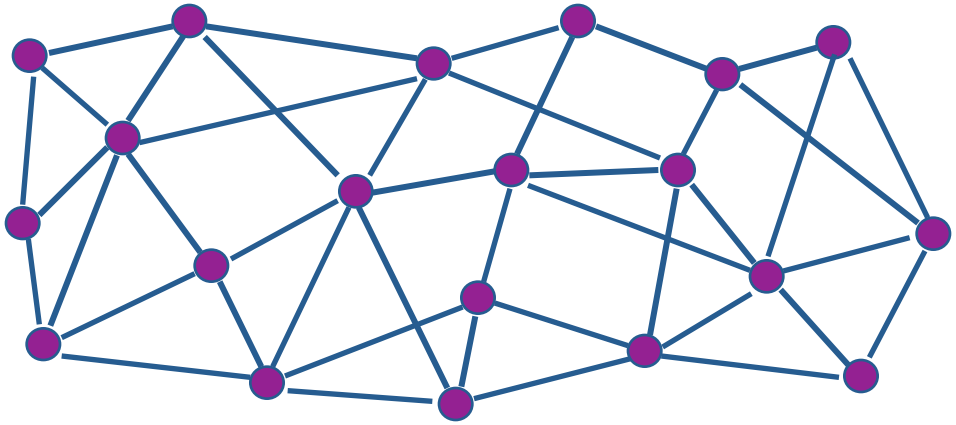
NANYANG TECHNOLOGICAL UNIVERSITY

BROWN

Stochastic Graph Models Workshop, ICERM, March 2014

Distributed Network





Undirected network/graph.

Nodes --- computers/hosts/agents.

Edges --- communication links.

Neighbors communicate via message passing.

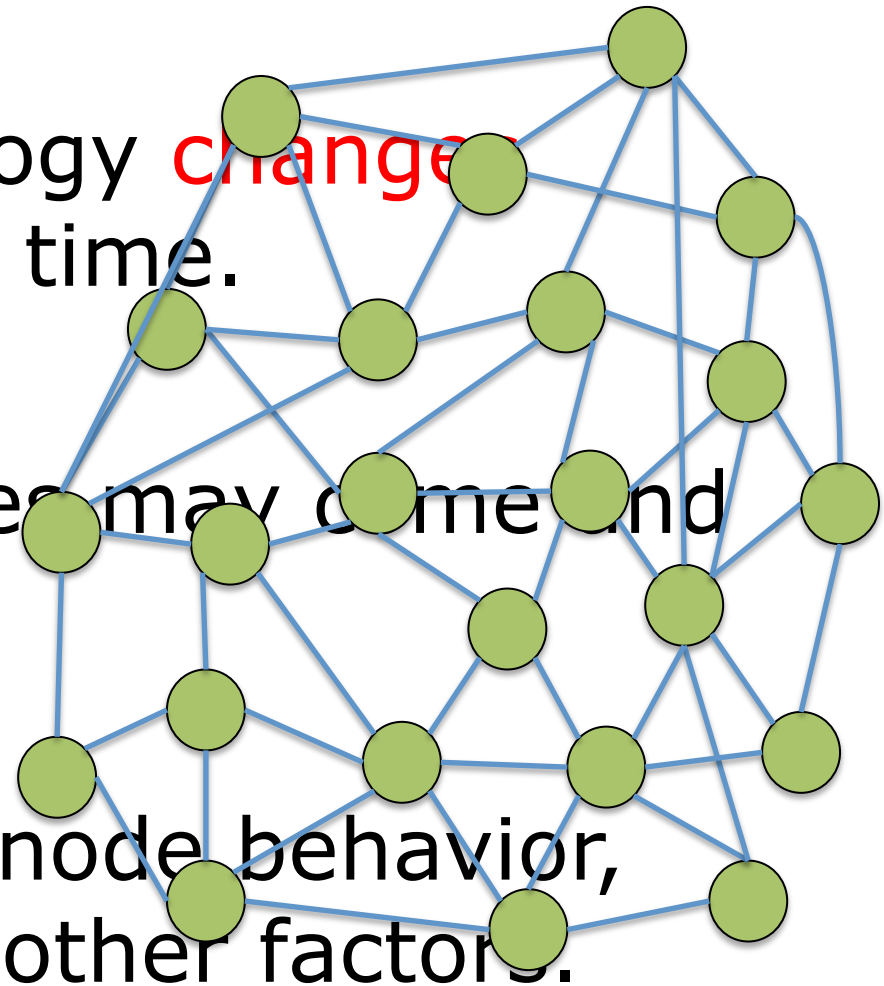
Nodes initially only have local knowledge.

Static Networks

- Nodes never crash.
- Edges maintain operational status forever.
- Focus of much of the established theory.

Dynamic Networks

- The network topology changes continuously over time.
- Nodes and/or edges may come and go.
- Dynamism due to node behavior, faults, traffic, and other factors.



Dynamic Networks Galore

- Peer-to-Peer (P2P) Networks
  - Large-scale, highly dynamic.
  - ~ 50% node churn.
  - Examples: Skype, BitTorrent, …
  - Significant fraction of Internet traffic.

- Mobile, ad hoc wireless, sensor, vehicular networks
  - Moving nodes.
  - Changing environmental conditions.
  - Failures of nodes/links.

- Social networks
  - Examples: Facebook, Twitter…
  - Nodes (people) and connections appear and disappear.

Fundamental Problems in Distributed Computing

- Agreement: Nodes have to agree on a common value.
- Lea twork shc
- Inf some nodes to all nodes.
- Storage and Search: Find data and resources in the network; store data reliably and securely.
- Byzantine problems: Tolerate malicious nodes.

Building blocks of distributed computing.

Distributed Computation in Dynamic Networks

- Dynamic networks pose non-trivial challenges in solving even basic distributed problems.
- Models and algorithmic techniques developed for static networks are not applicable to dynamic networks.
- Need for rigorous models and algorithmic techniques for dynamic networks.
- The rest of the talk is a step in this direction…

Road Map

- A model for dynamic networks.
- Agreement problem.
- Byzantine agreement.
- Storage and Search.
- Information Spreading – Rajmohan Rajaraman's talk on Wednesday.
- Summary.

Dynamic Network Model

Augustine, Pandurangan, Robinson, Upfal, SODA, *2012*.

- Sequence of graphs: $G^0, G^1, G^2, \ldots$
  --- A graph process

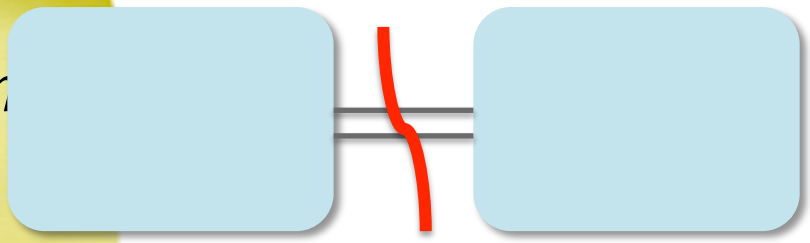
Initial network

Chosen by an **adversary --- can be oblivious or adaptive**

$n = |G^0| = |G^1| = \cdots$

Node churn up to $\leqslant \epsilon n$ nodes per round. Can target all nodes in $1/\epsilon$ rounds!

- $G^0, G^1, G^2, \ldots$ are **expanders**
- Constant degrees and $O(\log n$ diameter
- Expansion α depends on ϵ



Distributed Computing Model

- Computation proceeds in a sequence of synchronous rounds.
- Nodes initially have only local knowledge.
- In each round, each node can communicate with its (current) neighbors by exchanging messages of small (polylogarithmic) size.
- Goal: Fast algorithms --- terminate in polylogarithmic (in network size) number of rounds.

Road Map

- A model for dynamic networks.
- Agreement problem.
- Byzantine agreement.
- Storage and Search.
- Summary.

Stable Agreement (SA)

Adversary assigns nodes in G^0 either 0 or 1.

Node can (irrevocably!) decide

Some nodes can remain undecided but no conflicting decisions!

Problem Definition

① **Almost Everywhere Agreement**: After some rounds, almost all nodes decide on common value x.

② **Validity**: Some node in G^0 must have started with x.

③ **Stability**: Maintain Almost Everywhere Agreement ad infinitum.

Results

Augustine, Pandurangan, Robinson, Upfal, SODA, *2012*

Oblivious Adversary
We can reach Stable Agreement despite $\leqslant \epsilon n$ churn per round; $O(\log^2 n)$ rounds and $O(\log^c n)$ size messages whp.

Adaptive Adversary
We can reach SA despite $\epsilon\sqrt{n}$ churn; $O(\log^c n)$ rounds and $O(n^c)$ size messages whp.

Next: Focus on result for **oblivious adversary**.

Simple flooding-based algorithm

Doesn't work!

Code for every node *u*:

Set *x* := input bit (assigned to *u* by adversary)

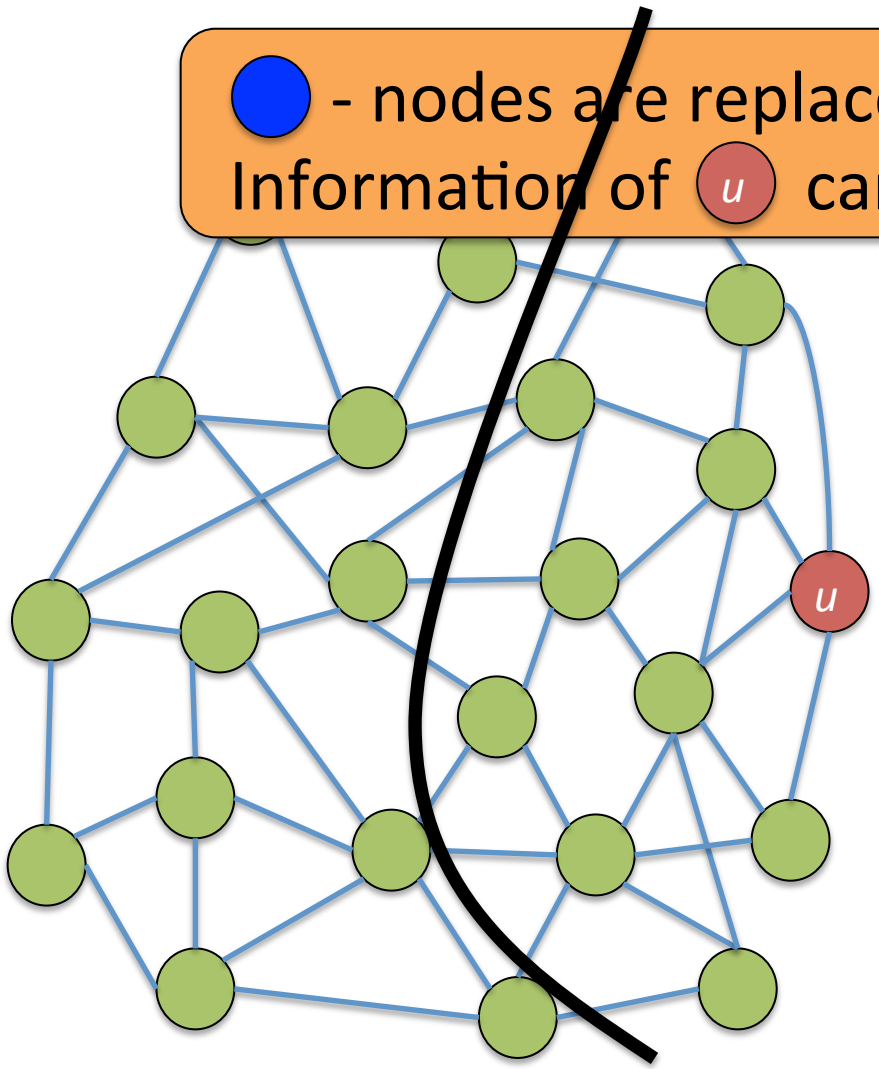
Forever flood message <*x*>

Upon receiving message <*y*> from neighbor:

Set $x = \max(y, x)$

After $O(\log n)$ steps decide on *x*.

How the adversary can ruin the day...



- nodes are replaced in every step.
Information of u cannot spread!

signs inputs:
all - nodes have 0;
node u starts with 1

Adversary **suppresses** node u (has constant degree!)

Shortly before decision, adversary allows u to spread value 1.

Our approach

Use randomization to defeat adversary

Techniques for building SA Protocol

1. Leverage graph expansion to show information spreading

2. Estimate support of input values using exponential distribution

Dynamic Distance & Influence

Dynamic Distance (DD)

Suppose node $u \in G^s$ floods message m.

$DD_s(u \to v)$ = number of rounds until m reaches v.

Influence Set of u after R rounds starting at round s:

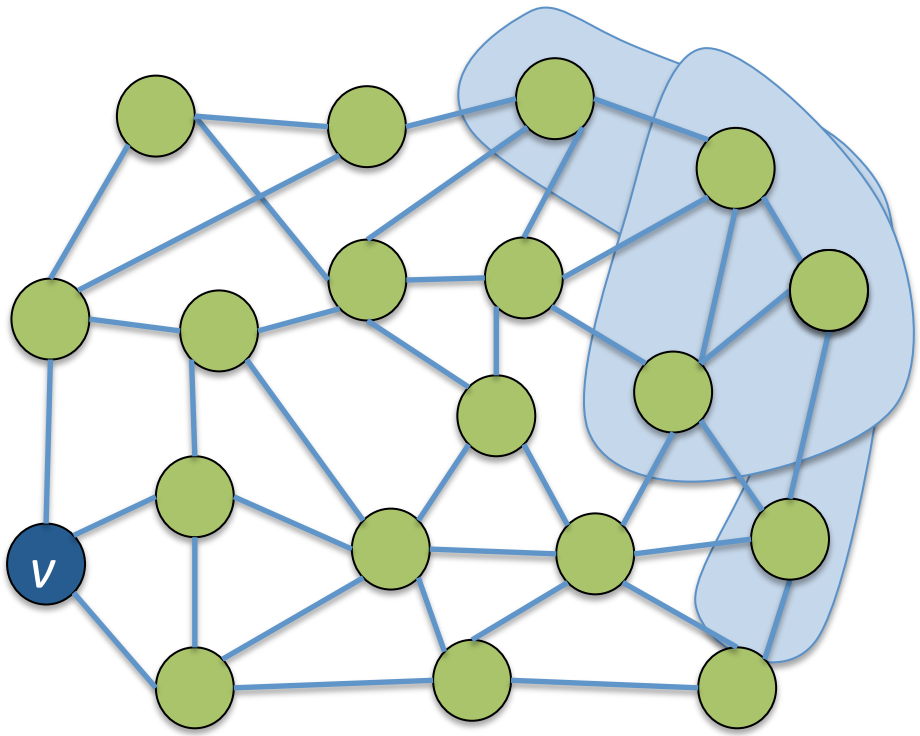
$$\mathrm{Infl}_s(u, R) = \{v \in G^{s+R} : DD_s(u \to v) \leqslant R\}$$

In **static graphs**:

$DD_r(u \to v)$ = graph distance between u, v

$\mathrm{Infl}_r(u, R)$ = causal influence set of u

Dynamic Distance (DD) & Influence Sets



DD can be larger than "static" distance

DD(*u*,*v*) is infinite if:

1. *v* is prematurely churned out
2. *u* is shielded by churn nodes

How many nodes do we need to start with and how long does it take to influence almost all nodes?

Influence of a set of nodes

Graphs $G^0, G^1, G^2, \ldots$ have expansion $\geqslant \alpha$.

Churn is $\leqslant \epsilon n$ per round.

$\mathrm{Infl}(U, T) := \bigcup_{u \in U} \mathrm{Infl}(u, T)$

Lemma (Growth of Influence Sets)

Fix any $\theta > \frac{\epsilon(1+\alpha)}{\alpha}$ and consider set U of θn nodes.
After $T \in O(1)$ rounds:

$$|\mathrm{infl}(U, T)| \geqslant (1 - \theta)n$$

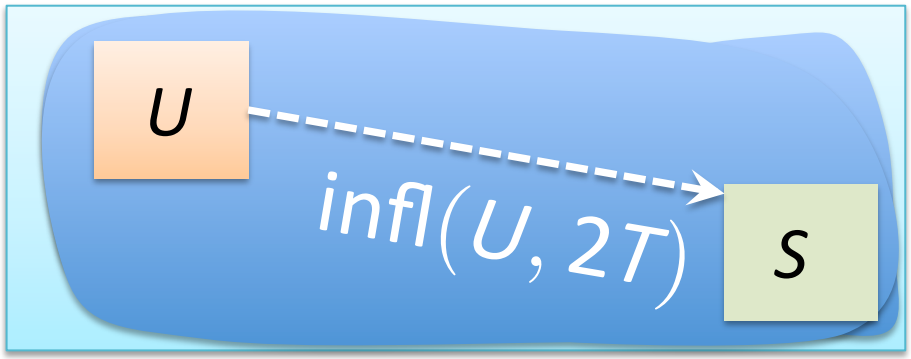
Proof idea

1. Show that U influences $n/2$ nodes within T rounds:

$$|\mathrm{infl}(U,1)| \geqslant (\theta-\epsilon)n(1+\alpha) = \theta n(1+\alpha) - \epsilon n(1+\alpha)$$

$$|\mathrm{infl}(U,i)| \geqslant \theta n(1+\alpha)^i - \epsilon n \textstyle\sum_{k=1}^{i}(1+\alpha)^k \geqslant n/2$$

$i \in O(1)$



2. Jump to round $2T$. Consider set S of size θn and use expansion argument in reverse back to round T. Shows: S is influenced by at least $n/2 + 1$ nodes.

$\Longrightarrow$ U influences S within $2T$ rounds.

How many nodes can the adversary suppress?

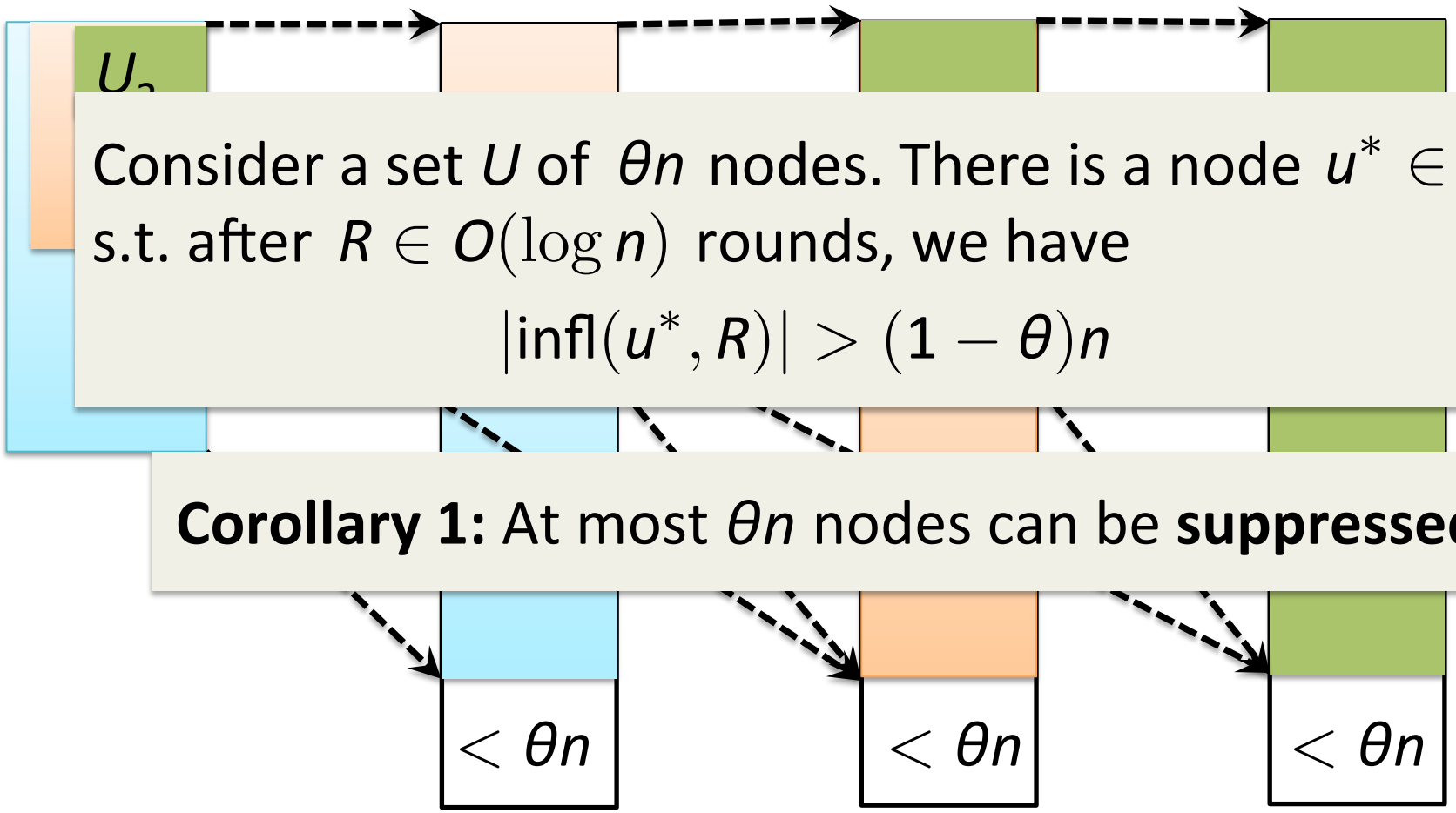
- Growth of influence sets not strong enough.
- Doesn't say anything about influence of *individual* nodes

u^* is **unsuppressed** node.

Consider a set U of θn nodes. There is a node $u^* \in U$ s.t. after $R \in O(\log n)$ rounds, we have

$$|\text{infl}(u^*, R)| > (1 - \theta)n$$

Finding a node with high influence



Consider a set U of θn nodes. There is a node $u^* \in U$ s.t. after $R \in O(\log n)$ rounds, we have

$$|\mathrm{infl}(u^*, R)| > (1 - \theta)n$$

Corollary 1: At most θn nodes can be **suppressed**!

We get a node $u^* \in U$ with $|\mathrm{infl}(u^*, R)| > (1 - \theta)n$ in $O(\log n)$ rounds.

Estimating the support of a value

Adversary assigns k nodes value 1 and $n - k$ nodes value 0.

Goal: For any $\delta > 2\theta$, at least $(1-\theta)n$ nodes estimate m = max(k,n-k) to be within $[(1-\delta)m, (1+\delta)m]$ whp.

Property of Exponential Distribution:
Consider $Y_1, \ldots, Y_j$ independent RVs of rate λ.
The minimum of all RVs is exponentially distributed with rate $j\lambda$.

Protocol for Support Estimation

Code f

Ge rate 1.

Init

For $O(\log n)$ rounds, keep flooding the *smallest* encountered sample s.

After $O(\log n)$ rounds, output estimate $k = 1/s$.

Accurate only on expectation; to get concentration, we perform $O(\log n)$ parallel iterations and take average

Intuition for correctness:

- Oblivious adversary does not know random choices.
- Cannot reliably suppress small samples.

Protocol for Stable Agreement

Node u starts with bit b:

At every checkpoint R (= $O(\log n)$ rounds) do:

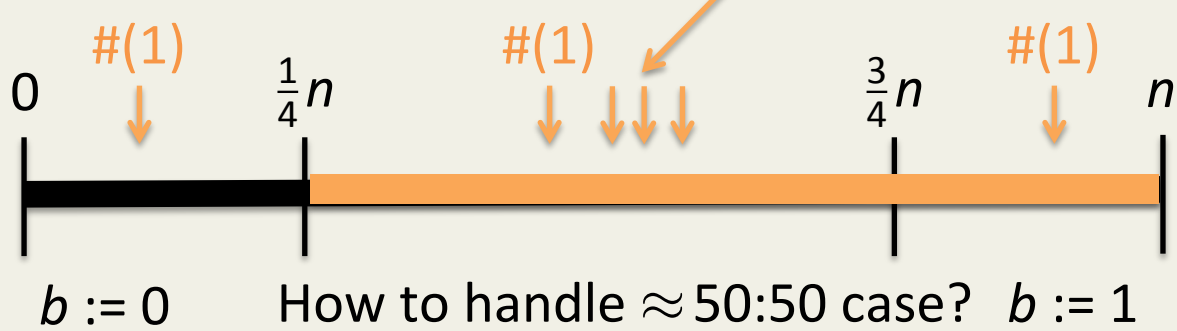
Initiate support estimation of # of 1s

Generate ra

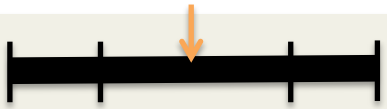
Whp, nodes that estimate #(1) to be above $1/4\ n$ is $\leqslant \theta n$.

Flood smallest received (r_v,b) pair; initially (r_u,b)

Use estimation of # of 1s (initiated R rounds ago):



How to handle $\approx$ 50:50 case?

If  then:

Let (r_v, b_v) be flooding message with smallest r_v that u received since last checkpoint (R rounds).

Set $b := b_v$

Why does this help?

- Adversary tries to suppress node with smallest r_v.
- It fails with constant probability.
- In $O(\log n)$ iterations fails once whp and sways decision one way.

After $O(\log n)$ phas

Let #(1), #(0) =

If #(1) > $n/2$ the

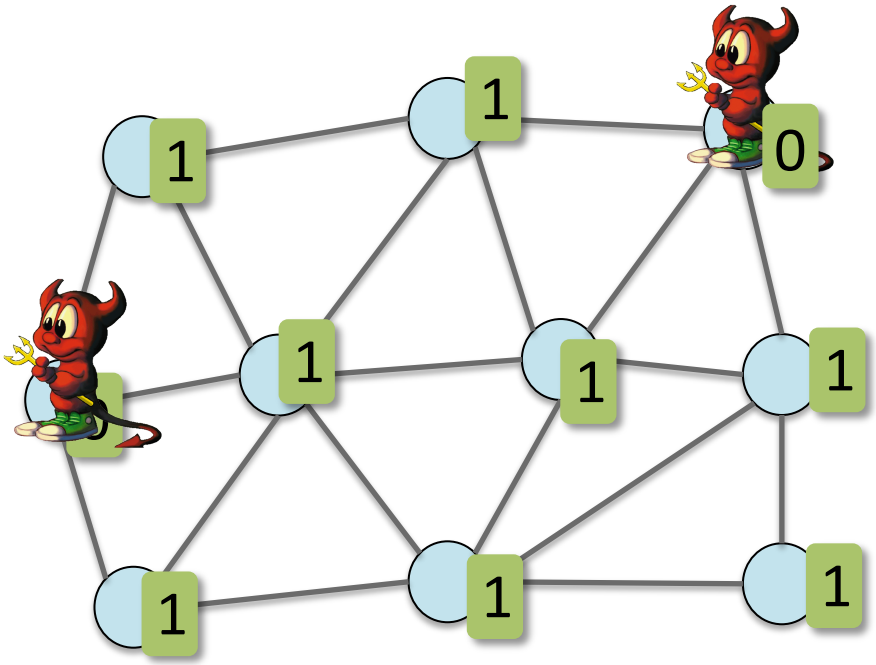
If #(0) > $n/2$ then

Otherwise remain undecided.

Road Map

- A model for dynamic networks.
- Agreement problem.
- Byzantine agreement.
- Storage and Search.
- Summary.

Byzantine Almost Everywhere Agreement (BAE)



Each node has input value.

Sparse network.

Some nodes are Byzantine.

Termination: Eventually every correct node decides on value.

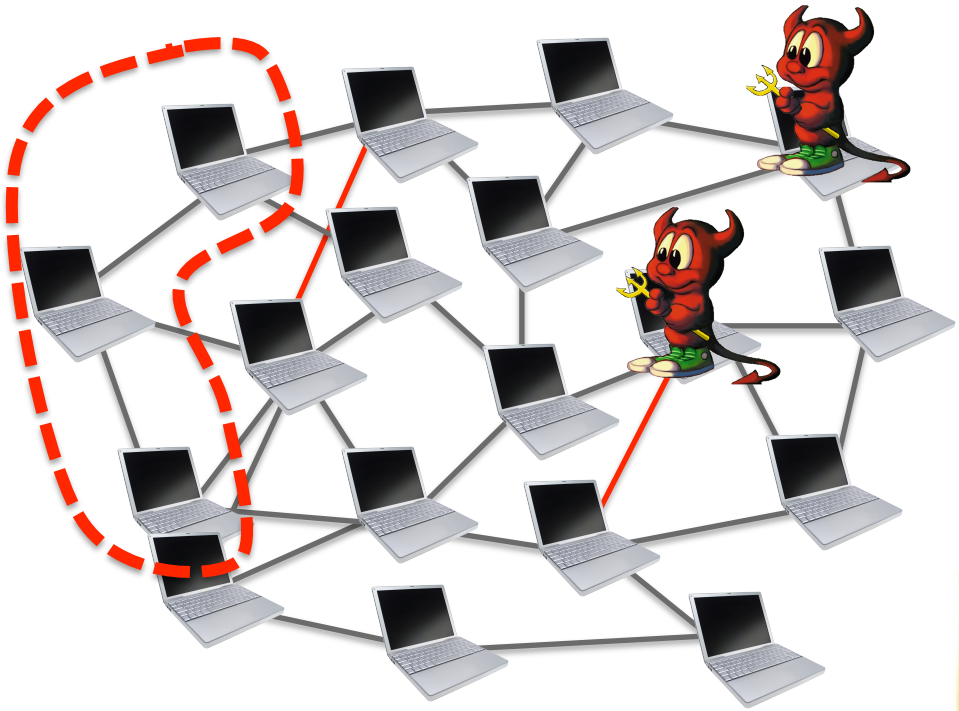
Almost Everywhere Agreement: All but $o(n)$ correct nodes decide on the same value.

Validity: If all correct nodes start with value v, their decision is v.

Solved for "static" networks.
[Dwork,Peleg,Pippenger,Upfal'88]

Dynamic Network Model with Byzantine Nodes

Augustine, Pandurangan, Robinson, PODC 2013



Synchronous Rounds

Sequence $G^1, G^2, \ldots$

Communication in round r determined by $G^r = (V^r, E^r)$

Stable network size n.

Adaptive adversary **churns** $O(\sqrt{n}/\log^c n)$ nodes per round.

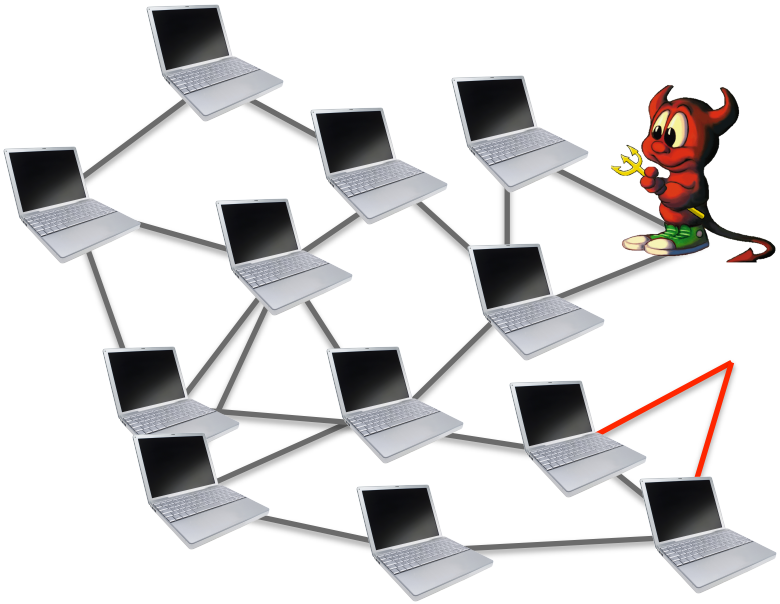
Oblivious adversary rewires edges in each round:

Each G^r is d-**regular expander**

$O(\sqrt{n}/\log^k n)$ **Byzantine nodes**:

Can deviate from protocol

The Power of the Adversary



Adversary controls edges:
$\Rightarrow$ Byzantine nodes are mobile.

No private channels:
Byz nodes know entire state.

''Worst case'': Same number of 0s and 1s.

Ways to keep network undecided:

1. inject fake values
2. remove strategic nodes

Our Results

Algorithm

Assume $O(\sqrt{n}/\log^{k+3} n)$ churn per round and $O(\sqrt{n}/\log^{k} n)$ Byzantine nodes. There is a BAE algorithm that takes $O(\log^3 n)$ rounds, using messages of $O(\log^2$ size. ($k \in \Theta(1)$)

Optimal up to log-factors.

Lower Bound

Assume $O(\sqrt{n \log n})$ churn per round.
plexity of a (B)AE algorithm is $\Omega(\sqrt{n})$.

$\Rightarrow$ For polylog time complexity we need churn $\in \tilde{O}(\sqrt{n})$.

Augustine, Pandurangan, Robinson, PODC 2013

Overview of the Algorithm

Providing almost uniform sampling to most nodes via **random walks**.

Converging to agreement via **majority rule**.

Random Sampling by Token Forwarding

A Phase of Random Walks: [Code for Node *u*]

Initially: Generate $h \log n$ tokens; store in *BUF* (FIFO buffer)

Token has fields $\langle id_u, TTL, \text{msg} \rangle$; $TTL = \tau = \Theta(\log n)$.

For $\Theta(\log^2 n)$ **rounds do:**

Send $\leqslant h \log n$ tokens of *BUF* to random neighbors and decrease their *TTL* fields.

Add received tokens to *BUF*.

Return tokens in *BUF* with *TTL*

u's current decision estimate

Nodes in CORE remain in network for $\Theta(\log^2 n)$ rounds.

Theorem – Dynamic Sampling

There exists set CORE of size $n - O(\sqrt{n}/\log^{k-4} n)$ such that:

1. All except $O(\sqrt{n}/\log^{k-6} n)$ nodes in CORE rcv $\Theta(\log n)$ tokens each;

2. received tokens originated with almost uniform ($\pm \frac{1}{n^3}$) probability from nodes in CORE.

The BAE Algorithm

Code for Node u:

Initially: Set x_u to input value $\in \{0, 1\}$.

In Phase $i = 1, \ldots, \Theta(\log n)$ **do:**

Run random walk implementation.

Pick 2 samples s_1, s_2 rcvd in this phase:
Set x_u to majority value of x_u, s_1, s_2.

Decide on x_u.

If $u \in$ CORE, then s_1, s_2 almost uniform among Core

Theorem – Dynamic Sampling

There exists set CORE of size $n - O(\sqrt{n}/\log^{k-4} n)$ such that:

1. All except $O(\sqrt{n}/\log^{k-6} n)$ nodes in CORE rcv $\Theta(\log n)$ tokens each;

2. received tokens originated with almost uniform $(\pm \frac{1}{n^3})$ probability from nodes in CORE.

Byzantine nodes can simulate churn

Suppose network $\mathcal{H} = (H^0, H^1, \dots)$ has $\epsilon_1 \sqrt{n} / \log^k n$ Byz nodes and $\epsilon_2 \sqrt{n} / \log^{k+3} n$ churn per round.
Suppose network $\mathcal{G} = (G^1, G^2, \dots)$ has $(\epsilon_1 + \epsilon_2) \sqrt{n} / \log^k n$ Byz nodes and 0 churn.

Sufficient to prove properties for $\mathcal{G}$ (only Byz)!

For every run α on $\mathcal{H}$, there is run β on $\mathcal{G}$ where **correct nodes receive same tokens**, for first $O(\log^3 n)$ rounds.

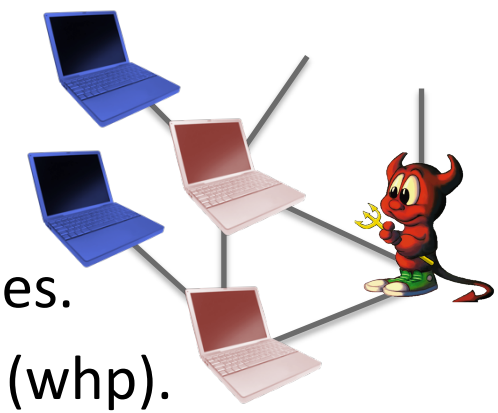
Bounding Impact of Byzantine Nodes

Random walk of length $\Theta(\log n)$ required for mixing on expander.
Show: 1. Most tokens complete $\Theta(\log n)$ steps in $O(\log^2 n)$ rounds.
2. $\geqslant n - O(\sqrt{n}/\log^{k-2} n)$ nodes "unaffected" by Byz nodes.

Assume Byzantine nodes generate "legit" tokens (otherwise ignored).

G^r constant deg $\Rightarrow$ $O(\log^2 n)|\text{Byz}|$ nodes affected by Byz nodes.

G^r regular $\Rightarrow$ $\forall r$: unaffected nodes receive
$\leqslant h \log n$ tokens (expectation), and
$\leqslant 2h \log n$ tokens (whp).



Token is **unaffected** if it only reached unaffected nodes.

In round r : unaffected token delayed for $\leqslant r$ rounds (whp).

$\Rightarrow$ Unaffected tokens complete $\Theta($ Recall: $O(\sqrt{n}/\log^k n)$ Byz nodes

$\geqslant n - O(\sqrt{n}/\log^{k-2} n)$ nodes unaffected

$\Rightarrow$ **most tokens unaffected.**

Theorem – Dynamic Sampling

There exists set CORE of size $n - O(\sqrt{n}/\log^{k-4} n)$ such that:

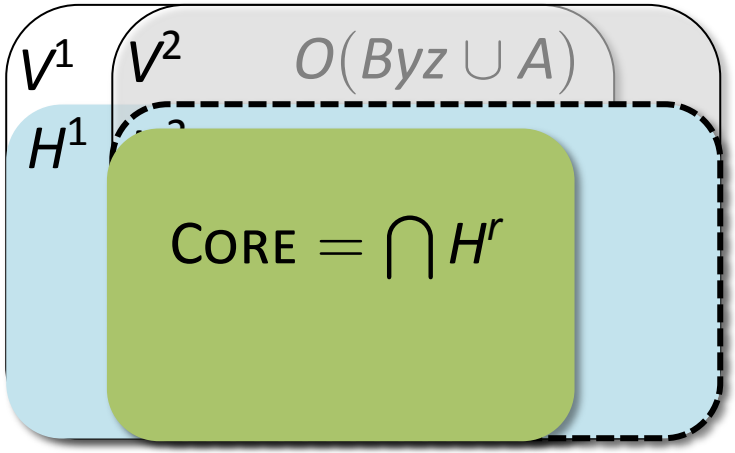
1. All except $O(\sqrt{n}/\log^{k-6} n)$ nodes in CORE rcv $\Theta(\log n)$ tokens each;
2. received tokens originated with almost uniform ($\pm\frac{1}{n^3}$) probability from nodes in CORE.

A = set of **affected nodes**; neighbors of Byz in some round

$$|H^r| = n - O(\sqrt{n}/\log^{k-2} n)$$

$$|\text{CORE}| = n - O(\sqrt{n}/\log^{k-4} n)$$

CORE has no Byzantine neighbors.



$\Rightarrow O(\sqrt{n}/\log^{k-6} n)$ tokens across cut in $\Theta(\log^2 n)$ rounds.

$\Rightarrow O(\sqrt{n}/\log^{k-6} n)$ nodes in CORE have "outside" tokens.

In each r: There is **expander** $H^r \subset G^r \setminus (\text{Byz} \cup \text{A})$ and $|H^r| = n - O(\text{Byz} \cup \text{Aff})$

$\Rightarrow 2h - O(\sqrt{n}/\log^{k-2} n)$ tokens end at unaffected nodes.

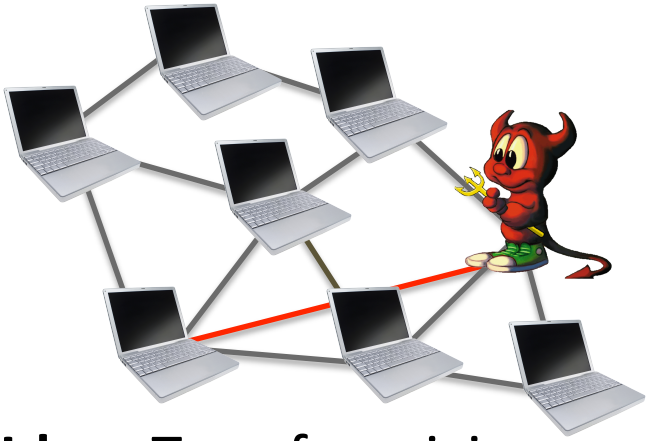
$\Rightarrow$ nodes "unaffected" by Byz nodes.

Theorem – Dynamic Sampling

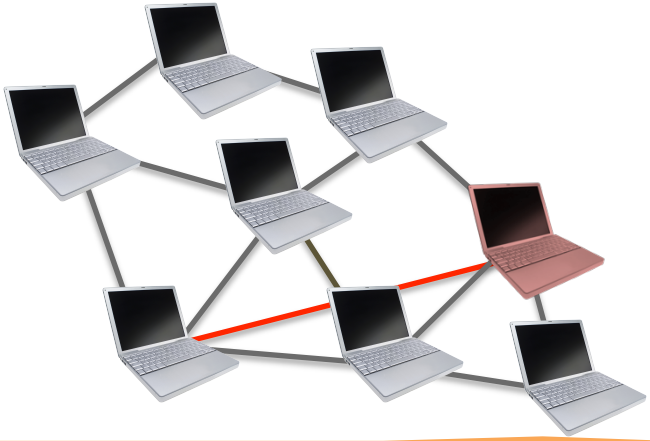
There exists set CORE of size $n - O(\sqrt{n}/\log^{k-4} n)$ such that:

1. All except $O(\sqrt{n}/\log^{k-6} n)$ nodes in CORE rcv $\Theta(\log n)$ tokens each;
2. received tokens originated with almost uniform ($\pm\frac{1}{n^3}$) probability from nodes in CORE.

$\mathcal{G}$: dynamic network with Byzantine



$\mathcal{P}$: preserving dynamic network



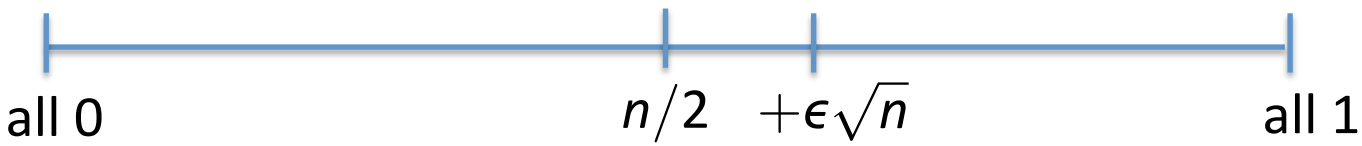
Idea: Transfer mixing properties of $\mathcal{P}$ to $\mathcal{G}$.

$\Rightarrow$ Distribution bounds hold

$\mathcal{P}$ is standard dynamic network: $\pm\frac{1}{n^3}$ close to uniform after $\Theta(\log n)$ steps; **[Das Sarma, Molla, Pandurangan, DISC 12]**

Convergence to AE Agreement

"Worst case": Same number of 0s and 1s in CORE.



Can we reach an **imbalance** $> \epsilon\sqrt{n}$?

Adversary can bias $\tilde{O}(\sqrt{n})$ nodes in CORE.

For each u define random var Y_u:
If $x_u = 0$, set $Y_u = 1$ iff x_u changed to 1.
If $x_u = 1$, set $Y_u = -1$ iff x_u changed to 0.

Let $Y = \sum_u Y_u$, Show Pr $\left[Y \geqslant c\sqrt{n}\right]$ is constant.

Define $Z = \frac{Y - \mathrm{E}[Y]}{\sqrt{\mathrm{Var}[Y]}}$. CLT $\Rightarrow$ Z converges to normal distribution.

$\Rightarrow$ Constant probability to achieve large imbalance.

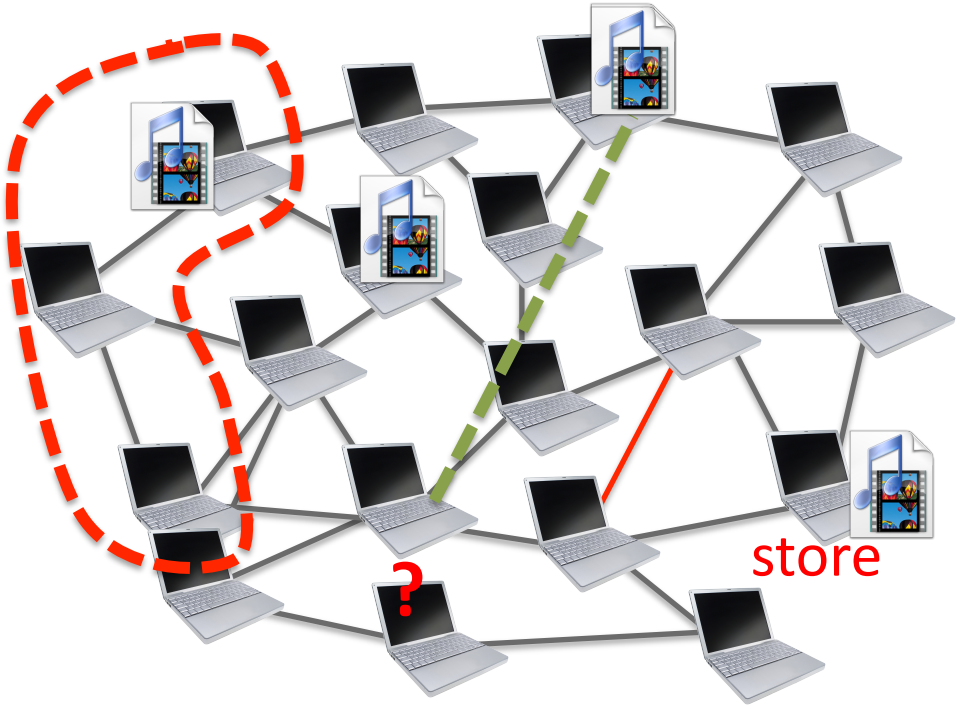
Y_u aren't identically distributed. CLT requires Lyapunov Condition.

$\Rightarrow$ In $\Theta(\log n)$ phases (each of $\Theta(\log^2 n)$ rounds) agreement whp.

Road Map

- A model for dynamic networks.
- Agreement problem.
- Byzantine agreement.
- Storage and Search.
- Summary.

Storage and Search of Data Items



Augustine, Molla, Morsy, Pandurangan, Robinson, Upfal, SPAA 2013.

Node can request **storage** of data item among peers.

Nodes can **search** for item.

In real world P2P Networks

- high churn – all the time!
- topology changes

1. Small message complexity
2. Polylog(n) time

Goal: scalable search/storage algorithms for dynamic P2P networks with churn

Dynamic Network Model

Synchronous rounds:

compute-send-receive

Dynamic network
sequence $G^1, G^2, \ldots$
$G^r = (V^r, E^r)$

Oblivious adversary controls churn and edges

$O(n/\log^{1+\epsilon} n)$ churn per round

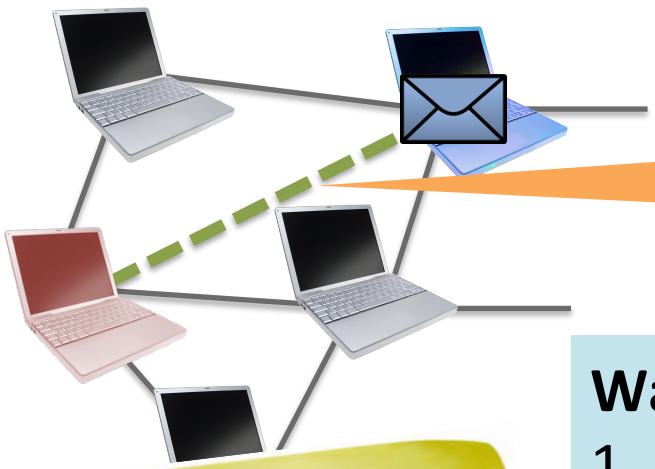
network size n is stable

Each G^r is d-**regular** expander

Communication Model

Node u entering in r can communicate with neighbors in G^r.

u can **learn ids** of other nodes.



Direct communication with "learnt" nodes

Ways to Communicate:

1. edges of G^r - controlled by adversary!
2. direct links to "learnt" nodes

Messages of $O(\log^2 n)$ size over each link

Search & Storage Revisited

Each G^r is **constant** degree expander

But $O(n/\text{polylog}(n))$ churn **per round**!

Adversary can isolate parts of the
ning

Storage

$n - o(n)$ nodes can store items for a polynomial number of rounds (whp).

Search

$n - o(n)$ nodes can retrieve available items in $O(\log n)$ rounds (whp).

Search & Storage Revisited

Storage

$n - o(n)$ nodes can store items for a polynomial number of rounds (whp).

Search

$n - o(n)$ nodes can retrieve available items in $O(\log n)$ rounds (whp).

A Simple Flooding Based Solution

Storage(key,item):

flood <STORE,key,item>
upon rcvng flooding msg,
locally store item

Search(key):

flood <id_u,SEARCH,key>
upon rcvng flooding msg,
send item to (direct link)

Problem 1: O(n) nodes

Problem 2: High

Goal: scalable search/storage algorithms for dynamic P2P networks with churn

Our Solution – Key Ingredients

Building Block - Committees

Forming and maintaining **committee** of $\Theta(\log n)$ randomly chosen nodes.

Building Block - Landmarks

Constructing **landmark tree** of randomly chosen nodes of size $\in [\Omega(\sqrt{n}), O(n^{1/2+\epsilon})]$.

Landmarks point to committee members.

Soup Theorem

There exists set MIX of size $n - O(n/\log^{1+\epsilon} n)$:
A **random walk** ending at any $u \in$ MIX, originates at any $v \in$ MIX with prob $\frac{1}{n} \pm O(\frac{1}{n})$.

A "soup" of mixing random walk tokens.

Similar to the "Dynamic Sampling" Theorem.

Storage and Search Algorithms

Building Block - Committees

Forming and maintaining **committee** of $\Theta(\log n)$ randomly chosen nodes.

Building Block - Landmarks

Constructing **landmark tree** of randomly chosen nodes of size $\in [\Omega(\sqrt{n}), O(n^{1/2+\epsilon})]$.

Storage request by *u* for (key,item):

u creates storage committee *SCom*.

Members of *SCom* store (key,item).

Every $\Theta(\log n)$ rounds:

SCom creates new landmarks (old landmarks discarded).

Retrieval of item by node *v*:

v creates search comm *Com*.

Nodes in *Com* create landmarks.

For $\Theta(\log n)$ rounds do:

If landmark *w* encounters storage landmark *z*:

z reveals *SCom* to *v* via direct link

Birthday paradox argument:

$\Omega(\sqrt{n})$ search landmarks vs $\Omega(\sqrt{n})$ storage landmarks

$\Rightarrow$ Search successful in $O(\log n)$ rounds (whp).

Summary

- Large-scale, highly dynamic networks are increasingly dominant in the real world.
- Distributed algorithms that are robust, efficient, and secure are required.
- Solve fundamental problems with provable guarantees, under strong models.

- We presented models, algorithms, and techniques that work even under a high amount of dynamism --- however it requires properties, such as good expansion.

- Future work: Build on current framework to design even stronger algorithms that work with minimal assumptions.

References

Towards Robust and Efficient Computation in Dynamic Peer-to-Peer Networks.
by John Augustine, Gopal Pandurangan, Peter Robinson, Eli Upfal.
*23rd ACM-SIAM Symposium on Discrete Algorithms (**SODA**), 2012.*

Fast Distributed Computation in Dynamic Networks via Random Walks.
by Atish Das Sarma, Anisur Rahaman Molla, Gopal Pandurangan.
*26th International Symposium on Distributed Computing (**DISC**), 2012.*

Fast Byzantine Agreement in Dynamic Networks.
by John Augustine, Gopal Pandurangan, Peter Robinson.
*32nd ACM Symposium on Principles of Distributed Computing (**PODC**), 2013.*

Search and Storage in Dynamic Peer-to-Peer Networks.
by John Augustine, Anisur Molla, Ehab Morsy, Gopal Pandurangan, Peter Robinson, Eli Upfal.
*25th ACM Symp. on Parallelism in Algorithms and Architectures (**SPAA**), 2013.*

On the Complexity of Information Spreading in Dynamic Networks.
Chinmoy Dutta, Gopal Pandurangan, Rajmohan Rajaraman, Zhifeng Sun, and Emanule Viola.
Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (**SODA**), 2013.

Near-Optimal Random Walk Sampling in Distributed Networks.
Atish Das Sarma, Anisur Rahaman Molla, and Gopal Pandurangan.
Proceedings of the 31st IEEE Conference on Computer Communications (**INFOCOM**), 2012.

Thank You !

Publications are available at:

http://www.ntu.edu.sg/home/gopal