Building Low-Diameter P2P Networks

Eli Upfal

Department of Computer Science
Brown University

Joint work with Gopal Pandurangan and Prabhakar Raghavan.

P2P Network

- A distributed network of computers;
  no distinction between a server and a client.
- A dynamic network: nodes (peers) and edges (currently established connections) appear and disappear over time.
- Nodes communicate using only local information.
- Advantages: decentralized computing (e.g., search), sharing data and resources.
- Real-life Systems: Gnutella, Freenet.

Gnutella

Joining:

- Nodes contact a (central) host server to get entry-points to the network.

Search:

- A node sends query to its neighbors.

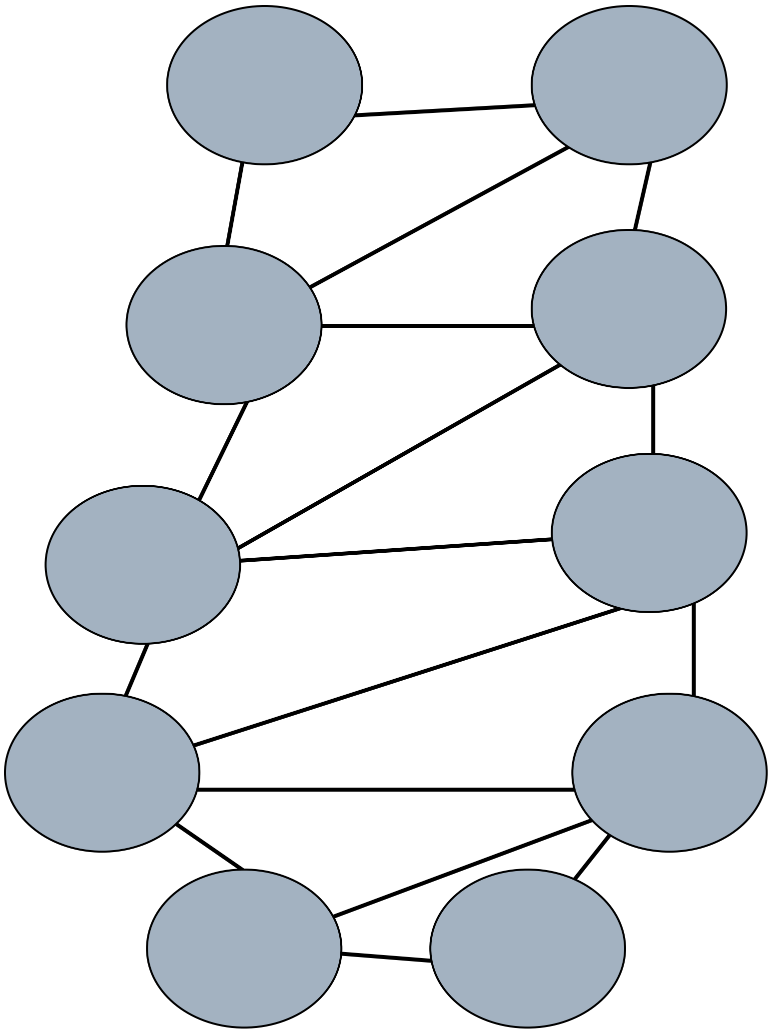
- They in turn forward it to their neighbors:
  - decrement "Time to Live (TTL)" for query;
  - query dies when TTL = 0.

- Search answers sent back along requested path.

Desired Global Network Properties

- Connectivity and low-diameter are two global properties crucial for doing search.
- Maintaining (even) global connectivity under a dynamic setting is a non-trivial issue.
- Current real-life systems (e.g., Gnutella):
  - take an ad hoc approach
  - results in partitioning of the network into disconnected pieces
- Challenge is to design distributed protocols which operate with only local knowledge.

A Bad Scenario



Each incoming node attaches to two recently joined nodes

- Long chains
- prone to disconnections

Our P2P Protocol

Pandurangan, Raghavan, and Upfal;
IEEE Symposium on Foundations of Computer Science (FOCS), 2001.

A distributed protocol to build P2P networks with provable guarantees under a reasonable model:

- connectivity.
- logarithmic diameter.
- constant degree.
- low overhead.
- operates with no global knowledge.
- can be easily implemented with local message passing.

To our knowledge, this is the first P2P protocol with provable guarantees on connectivity and diameter under a realistic dynamic setting.

The P2P Protocol: Preliminaries

- A set of rules applicable to various situations a node may find itself in:
  - How to join the network ?
  - What happens if a neighbor drops out ?
  - How to maintain bounded number of connections ?
- A central host server:
  - a gateway mechanism to enter the network.
  - maintains a cache - a list of K (= constant) nodes (i.e., their IP addresses) at all times.
  - is reachable by all nodes at all times.
  - need not know the network topology nor the identities of all the nodes in the network.

The P2P Protocol

On Arrival:

Connect to D ($D < K$) random nodes chosen from the cache.

Cache Replacement Rule:

When a cache node reaches degree C ($> D$) it is replaced from the cache by a node having degree D.

(Degree of all nodes bounded by a constant.)

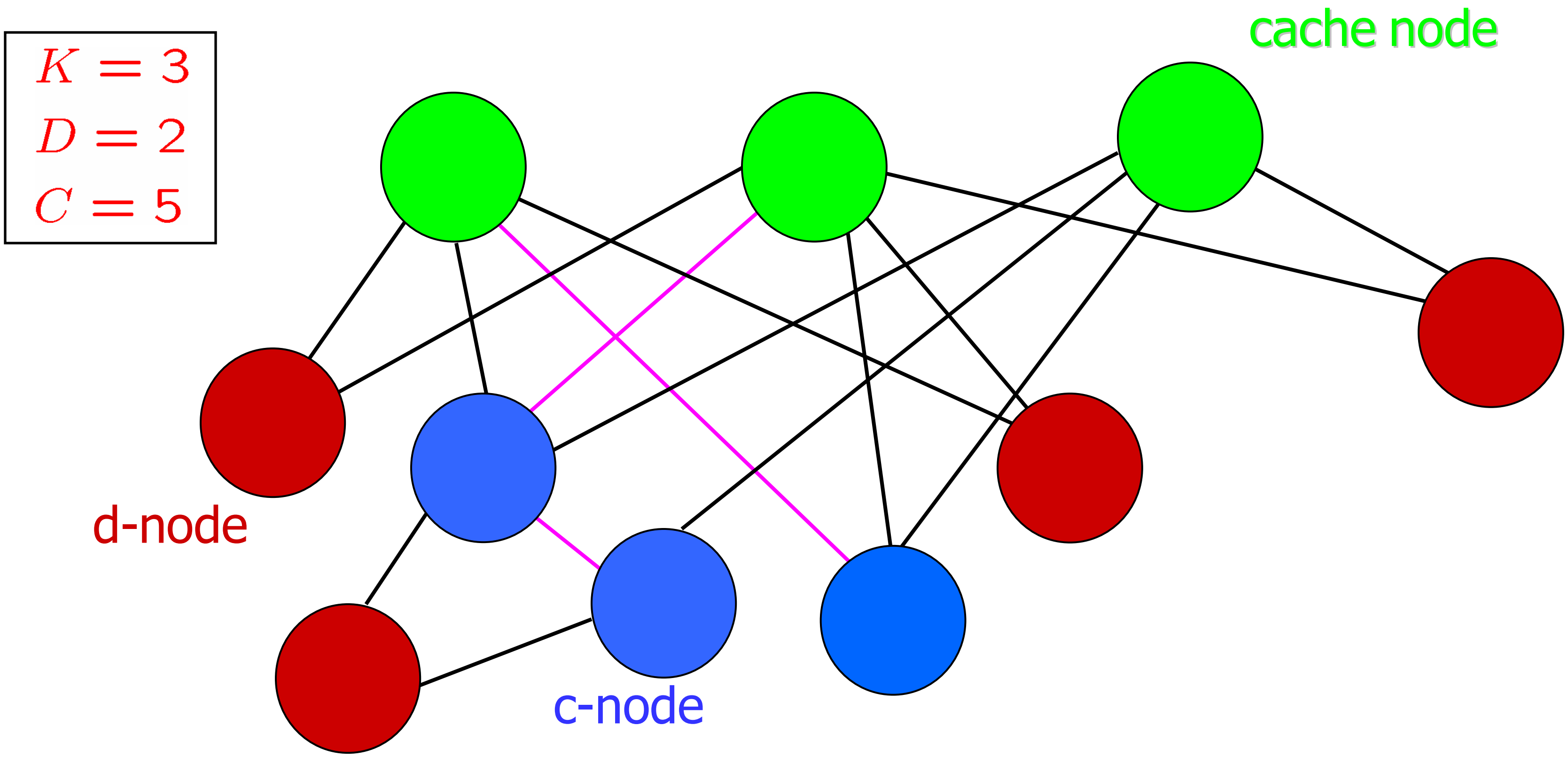
Reconnect Rule:

If a node (say v) loses its neighbor it (re-)connects to a random node in cache with probability $D/d(v)$.

- $d(v)$: degree of v before losing the neighbor.

(The above probabilistic reconnection is crucial to maintaining bounded degree. Degree of any node $\geq D$.)

Illustration



- c-node: node that was a cache node at some time.
- d-node: all other nodes.

The P2P Protocol (contd.)

These rules turn out to be crucial for maintaining connectivity.

Preferred Connection Rule*:*

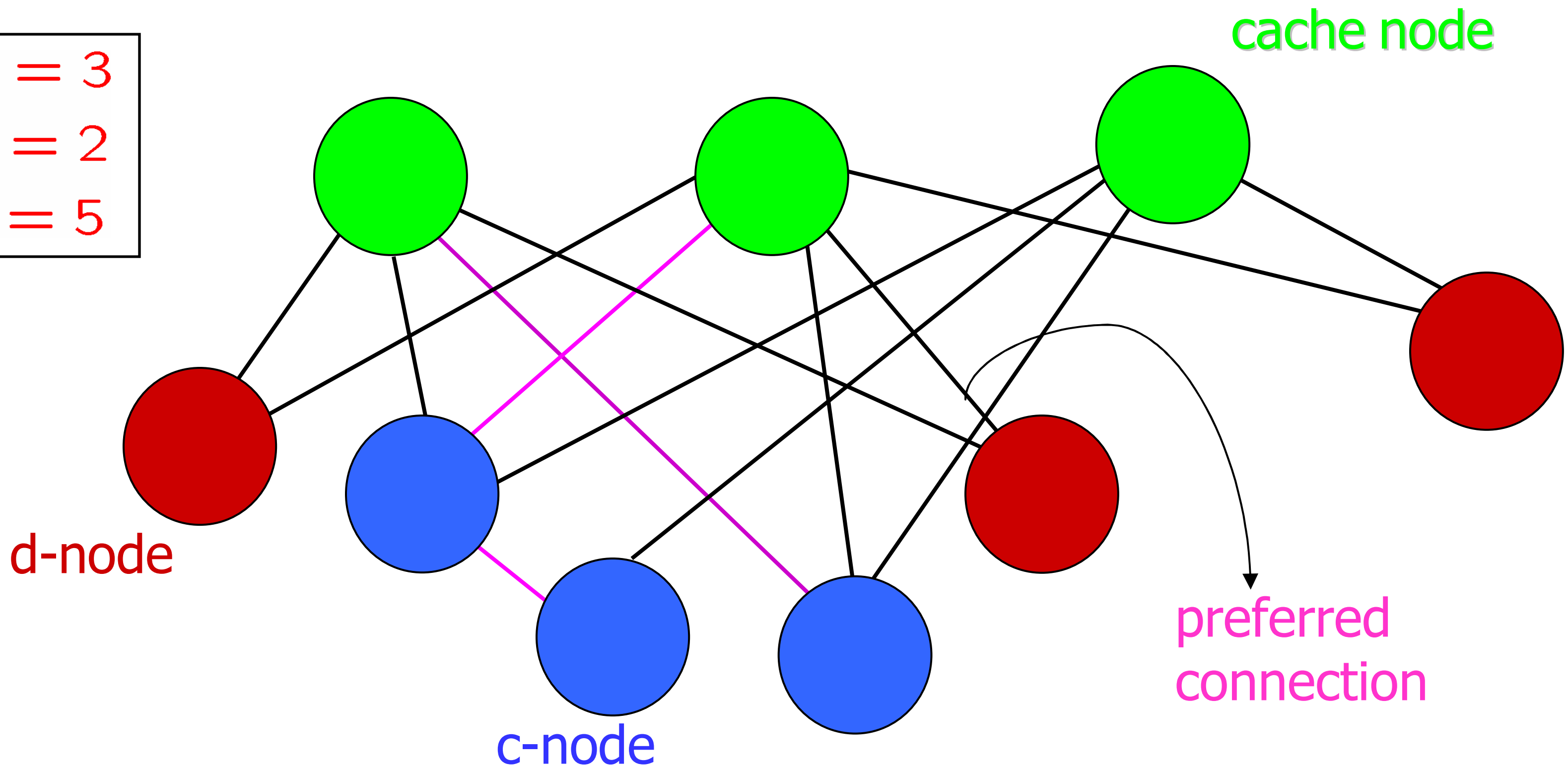
When a cache node leaves the cache it maintains a preferred connection to the node that replaced it in the cache.

Preferred Reconnect Rule:

If a node's preferred connection is lost, then it reconnects to a random node in the cache which becomes its new preferred connection.

Illustration: Preferred Connection

$K = 3$
$D = 2$
$C = 5$



- c-node: node that was a cache node at some time.
- d-node: all other nodes.

Our Stochastic Model

- Nodes arrive and depart in an uncoordinated and unpredictable fashion.
- A stochastic model for the dynamic setting:
  - Arrival of nodes: Poisson process with rate λ.
  - Duration of nodes: Independently and Exponentially distributed with parameter μ.
- A reasonable model:
  - Used in modeling similar scenarios e.g., the classical telephone trunking model in queuing theory.
  - Approximates real-life data fairly well; [*Sariou et al. MMCN 2002*] study of real P2P systems.
  - Insight into real-life performance.

A Stochastic Graph Process

- Let $G_t = (V_t, E_t)$ be the network at time t.
- We analyze the evolution in time of the graph process $\mathcal{G} = (G_t)_{t \geq 0}$.
- Network size depends only on the ratio $\frac{\lambda}{\mu} = N$.

Theorem 1.

a) For any $t = \Omega(N)$, with high probability (w.h.p.)

$|V_t| = \Theta(N)$.

b) If $\frac{t}{N} \to \infty$ then w.h.p. $|V_t| = N(1 \pm o(1))$.

(w.h.p. = $1 - N^{-\Omega(1)}$)

w.l.o.g, we let $\lambda = 1$, thus $\mu = \frac{1}{N}$.

Connectivity

Theorem 2.

There is a constant C such that at any given time $t > c \log N$,

$$\Pr(G_t \text{ is connected}) \geq 1 - O\left(\frac{\log^2 N}{N}\right)$$

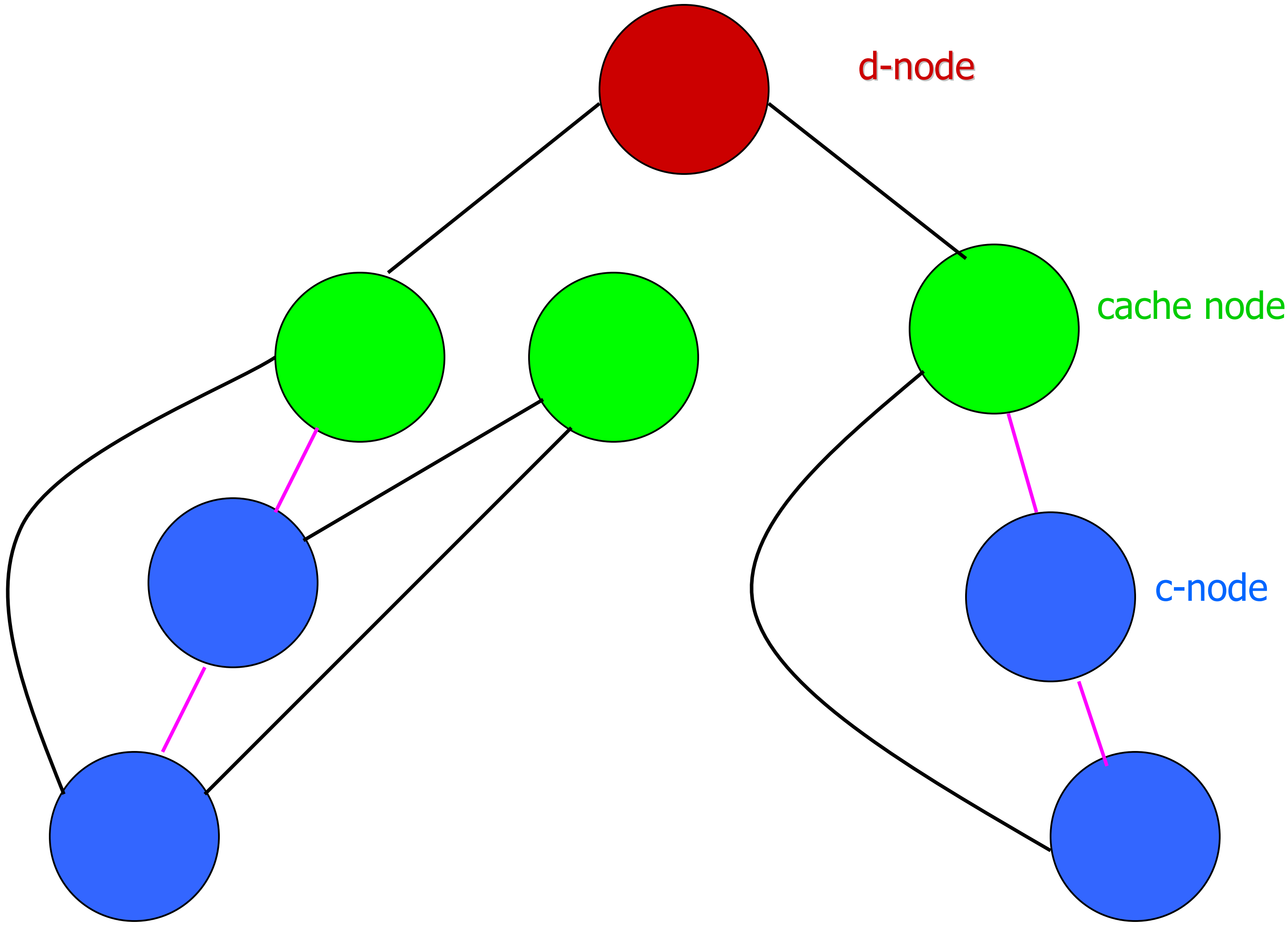
"The protocol maintains connectivity with large probability at any time after a short initial period."

Proof ideas:

- preferred connections (form a "backbone").
- random selection of cache nodes in the arrival rule.

Has a nice "self-correcting" property to recover from failures.

Connectivity: Intuition



Diameter

Theorem 3.

For any t, such that $\frac{t}{N} \to \infty$, G_t has diameter $O(\log N)$ with probability $1 - O\left(\frac{\log^2 N}{N}\right)$.

"The protocol maintains logarithmic diameter with large probability at any point of time after the network has "stabilized" (say, after $N \log N$ time from start)."

Proof Ideas:

- "reconnect" connections: "long-range", "random".
- "good" cache nodes: many reconnect connections.
- many good cache nodes.
- distance between any two nodes is $O(\log N)$.

Diameter: Intuition

- It is sufficient to analyze the distance between c-nodes.
- Let u and v be two c-nodes; then Pr(u and v are connected by a "reconnect" edge) $\approx \frac{d(u)}{N}\frac{D}{d(u)} = O\left(\frac{1}{N}\right)$
- Although the reconnect edges are "random" , their occurrence is not independent (unlike $G_{n,p}$ model of random graphs).
- More sophisticated analysis needed.

Good Cache Nodes

- A cache node is good if during its time in cache it receives at least f (= fixed constant) connections such that:
  - they are "reconnect" connections;
  - they are not preferred connections;
  - they resulted from different nodes leaving the network.
- Color the above connections blue; Color all other connections red.

Lemma 1.
Let node v enter the cache at time t, where $\frac{t}{N} \to \infty$. Then,

Pr(v leaves the cache as a good node) $\geq \frac{1}{2}$

- the f blue edges are distributed uniformly at random among the nodes in the current network.
- is independent of other c-nodes being good.

Proof Sketch of Lemma 1

- Nodes join the network according to a Poisson process with rate 1 . The expected number of connections to v from an incoming node is D/K .

- Nodes leave the network according to a Poisson process with rate ≈ 1. The expected number of connections to v as a result of a node leaving the network is:

$$\approx \sum_{u \in V} \frac{d(u)}{|V|} \frac{D}{d(u)} \frac{1}{K} \approx \frac{D}{K} < 1$$

D	$=$	lower bound on degree
K	$=$	number of cache nodes
C	$=$	max. number of connections a cache node can accept

- Each connection to v has a constant probability of being a reconnect connection.

- The probability of a node u connecting to v is

$$\approx \frac{d(u)}{N} \frac{D}{d(u)} = \frac{D}{N}$$

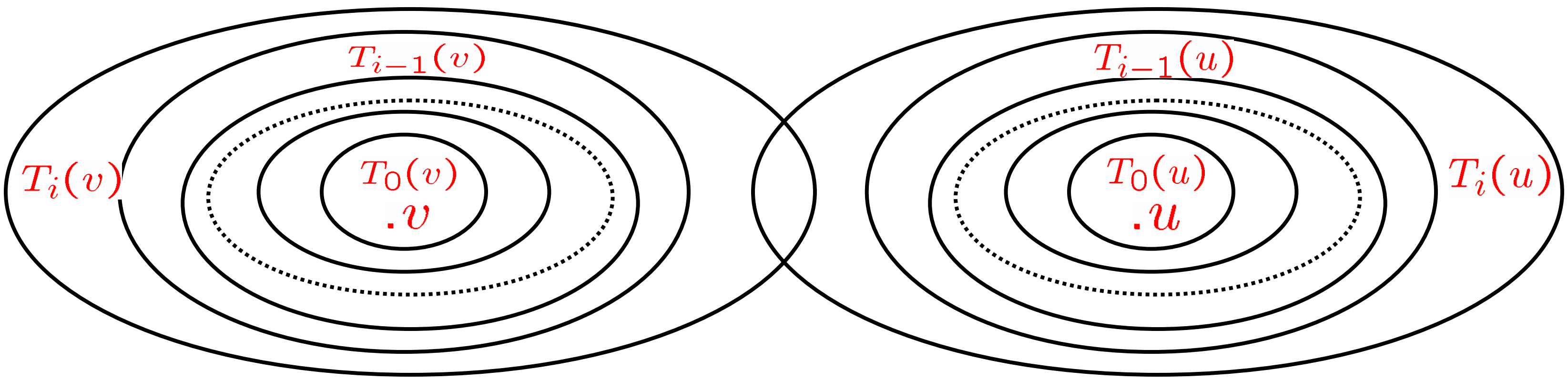
All nodes have equal probability of connecting to v.

For a sufficiently large C , the lemma holds.

Diameter: Expanding Neighborhoods

For a c-node v:

- $T_0(v)$: arbitrary connected cluster of $O(\log N)$ c-nodes including v, using red edges.
- $T_i(v)$: c-nodes that are connected by blue edges to $T_{i-1}(v)$, but are not in $T_0(v) \ldots T_{i-1}(v)$.

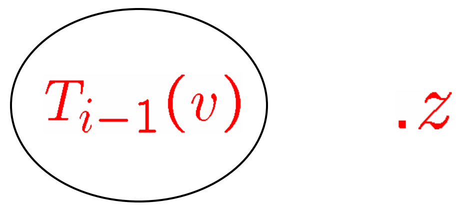


Expansion Lemma: Proof Sketch

Lemma 2.

If $|T_{i-1}(v)| = o(N)$ then w.h.p. $|T_i(v)| \geq 2|T_{i-1}(v)|$.

Proof sketch:



z is a good cache node with probability $\geq 1/2$ (Lemma 1)

$$\Pr(z \text{ is connected to } T_{i-1}(v)) \geq \frac{1}{2}\frac{f|T_{i-1}(v)|}{N}(1-o(1))$$

$$E[|T_i(v)|] \geq \frac{1}{2} f|T_{i-1}(v)|(1-o(1)) \geq 2|T_{i-1}(v)|$$

We use an exposure martingale to prove that $|T_i(v)|$ is concentrated around its mean with high probability.

Replacing Cache Nodes

Theorem 4.

There is a constant c such that at any time $t \geq c \log N$,

a) With high probability there is always a d-node in the network to replace a saturated cache node.

b) With probability $1 - O\left(\frac{\log^2 N}{N}\right)$ the protocol finds a replacement d-node by searching only $O(\log N)$ nodes.

Further Issues

- The network can maintain a bounded number of connections with high probability.
- The overhead involved per protocol step is constant.
- What if there are no preferred connections ?
  - Running the protocol without it leads to formation of many small disconnected components.