CS590R - Algorithms for communication networks
Lecture 19
Peer-to-Peer Networks (continued)

Lecturer: Gopal Pandurangan
Scribe: Ossama M. Younis
Department of Computer Sciences
Purdue University

1. Introduction

In this lecture, we continue our discussion on building low-diameter peer-to-peer networks. In the previous class, we introduced the basic definitions of a peer-to-peer network, introduced a proposed protocol for building low-diameter p2p networks [2], and started stufying the analysis of this protocol. We showed that the network size is bounded with respect to the arrival and departure rate of nodes. These bounds can be proven to prevail with high probabilities using Chernoff bounds [1]. Today, we continue studying the network size, capacity, and connectivity according to the proposed protocol.

2 Network Size

Theorem 1
Suppose that the ratio between arrival and departure rates in the network changed at time τ from N to N'. Suppose that there were M nodes in the network at time τ, then if $\frac{t-\tau}{N'} \rightarrow \infty$ w.h.p G_t has N'+o(N') nodes

Proof.
The proof is straigtforward. Compute the expected number of nodes in the network at time t, then apply the tail bound for the Poisson distribution. For more details, refer to [2].

3 Network Capacity

In this section, we show that there *always* exist a d-node in the network to replace any cache node that reaches a degree of C. The following lemmas show that (1) w.h.p., there are a large number of d-nodes in the network at any time t, and (2) the algorithm can replace any cache node efficiently, i.e., by examining only a small number of nodes ($O(log\ N)$). These lemmas will give some insight on the protocol, and help in our following discussion on network connectivity and diameter.

Lemma 1
Let $C > 3D$; then at any time $t \geq a \log N$, (for some fixed constant $a > 0$), w.h.p. there are

$$(1 - \frac{2D}{C-D}) \min[t, N](1 - o(1))$$

d-nodes in the network.

Proof.

- Let $t \geq N$ (similar proof for $t < N$).
- Consider the interval $[t - N, t]$.
- New connections to cache during this interval are contributed by (1) newly arriving nodes, and (2) old nodes trying to reconnect.
- Arrival of new nodes is Poissonian with rate ($\lambda = 1$)
- W.h.p. the number of new d-nodes arriving during this interval is $N(1 \pm o(1))$. Thus, the number of connections to cache nodes from the new arrivals is $DN(1 \pm o(1))$.
- For node v, the expected number of neighbors leaving the network in unit time is $(1 \pm o(1))\frac{d(v)}{N}$, probability that node v tries to reconnect because of a leaving neighbor is $\frac{D}{d(v)}$
- The expected number of connections to the cache nodes in from old nodes in this interval is bounded by
$$N \sum_{v \in V} (1 + o(1)) \frac{d(v)}{N} \frac{D}{d(v)} = ND(1 + o(1)).$$
- Thus, w.h.p. the number of connections to the cache from old nodes in this interval is bounded by $ND(1 + o(1))$.
- Summing up the two sources of new connections, the total number of new connections to the cache in this interval is $\leq 2ND(1 + o(1))$.
- Since a node receives $C - D$ connections while in the cache, w.h.p. no more than $\frac{2D}{C-D}N(1+o(1))$ d-nodes convert to new c-nodes in the interval.
- Thus, w.h.p we are left with $(1 - \frac{2D}{C-D})N(1-o(1))$ d-nodes that joined the network in this interval. Since $C > 3D$, it follows that we are left with *sufficient* d-nodes in the network.

Note that, the above proof assumes absolute independence of adding new connections. This is not completely true, since a neighbor leaving the network may trigger many many nodes to reconnect. However, this dependence is also still bounded by the maximum permissible degree.

Lemma 2
Suppose that the cache is occupied at time t by node v. Let $Z(v)$ be the set of nodes that occupied the cache (in v's slot) during the interval $[t - c \log N, t]$. For any $\delta > 0$ and sufficiently large constant c, w.h.p. $|Z(v)|$ is in the range $\frac{2Dc}{(C-D)K} \log N(1 \pm \delta)$.

Proof

- The expected number of connections to a given cache node in an interval $[t - c\log N, t]$ is $\frac{2Dc\log N}{K}(1 + o(1))$.
- Applying the Chernoff bound, we deduce that w.h.p. the number of connections is in the range $\frac{2Dc}{K}\log N(1 \pm \delta)$.
- Since a cache node receives $C - D$ connections while in the cache the result follows.

3.1 Efficient Cache Replacement

The following lemma comments on the efficiency of cache replacement in the proposed protocol.

Lemma 3
Assume that $C > 3D$. At any time $t \geq c\log\ N$, with probability $1 - O(\frac{\log^2 N}{N})$ the algorithm finds a replacement d-node by examining only $O(\log N)$ nodes.

Proof

- Let $v_1, ..., v_K$ be the K nodes in the cache at time t.
- With probability $Ke^{-\frac{a\log^2 N}{N}} \geq 1 - O(\frac{\log^2 N}{N})$ no node in $Z(v_i)$, $i = 1, .., K$ leaves the network in the interval $[t - c\log N, t]$.
- Assume that node v leaves the cache at time t. The protocol tries to replace v by a d-node neighbor of a node in $Z(v)$.
- Thus, w.h.p. $Z(v)$ received at least $\frac{D}{K}c\log N$ connections from new d-nodes in the interval $[t - c\log N, t]$.
- Among these new d-nodes no more than $|Z(v)|$ nodes entered the cache and became c-nodes during this interval.
- Thus, w.h.p. there is a d-node attached to a node of $Z(v)$ at time t.

4 Network connectivity

The proof that at any given time the network is connected w.h.p. is based on two properties of the protocol:

(1) Steps 3-4 of the protocol guarantee (deterministically) that at any given time a node is connected through a “preferred connection” to a cache node;

(2) The random choices of new connections guarantee that w.h.p. the $O(\log N)$ neighborhoods of any two cache nodes are connected to each other.

The first property is essential for connectivity. Without it, there is a constant probability that the graph has a number of small disconnected components

Lemma 4
At all times, each node in the network is connected to some cache node directly or through a path in the network.

Proof.

- A d-node is always connected to some c-node. Thus, it is sufficient to prove the claim for c-nodes.
- A c-node (v) is either: (1) in the cache, or (2) connected through its preferred connection to some node that was in the cache after v left.
- Therefore, the path of preferred connections must lead to a node that is currently in the cache

Lemma 5
Consider two cache nodes v and u at time $t \geq c \log N$, for some fixed constant $c > 0$. With probability $1 - O(\frac{\log^2 N}{N})$ there is a path in the network at time t connecting v and u.

Proof.

- Let $Z(v)$ be the set of nodes that occupied the cache (in v's slot) during the interval $[t - c \log N, t]$. W.h.p. $|Z(v)| = d \log N$, for some constant d.
- The probability that no node in $Z(v)$ leaves the network during the interval $[t - c \log N, t]$ is

$$e^{-\frac{cd \log^2 N}{N}} \geq 1 - O(\frac{\log^2 N}{N}).$$

- The probability that no new node that arrives during the interval $[t - c \log N, t]$ connects to nodes in both $Z(v)$ and $Z(u)$ is bounded by $(1 - D^2/K^2)^{c \log N} = O(1/N^{c'})$.
- It follows that if no node in $Z(v)$ leaves the network during this interval, then all nodes in $Z(v)$ are connected to v by their chain of preferred connections.
- The probability that no new node that arrives during the interval $[t - c \log N, t]$ connects to both $c(v)$ and $c(u)$ is bounded by $(1 - D^2/K^2)^{c \log N} = 1/N^c$

Theorem 2
There is a constant c such that at any given time $t > c \log N$,

$$Pr(G_t \textit{ is connected}) \geq 1 - O(\frac{\log^2 N}{N}).$$

Note that the above theorem does not depend on the state of the network at time $t - c \log N$. Therefore, it shows that the network recovers rapidly from fragmentation, as explained in the following Corrolary.

Corrolary 1
There is a constant c such that if the network is disconnected at time t,

$$Pr(G_{t+c \log N} \textit{ is connected}) \geq 1 - O(\frac{\log^2 N}{N}).$$

Theorem 3
At any given time t such that $t/N \to \infty$, if the graph is not connected then it has a connected component of size $N(1 - o(1))$.

Proof

- We showed that all nodes in the network are connected to some cache node.
- The probability that some cache node is left with fewer than $d \log N$ nodes connected to it is $O(\frac{\log^2 N}{N})$.
- Excluding such cache nodes, all other cache nodes are connected to each other with probability $1 - K^2(1 - D^2/K^2)^{c \log N} = 1 - 1/N^c$, for some $c > 0$.

References

[1] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
[2] G. Pandurangan, P. Raghavan, and E. Upfal. Building Low-Diameter P2P Networks. In *IEEE FOCS*, 2001.