Backup Slides

Building Low-Diameter Peer-to-Peer Networks

Theorem III.1

1) For any $t = \Omega(N)$, w.h.p. $|V_t| = \Theta(N)$
2) If $t/N \to \infty$ then w.h.p. $|V_t| = N \pm o(N)$.

- Proof
  - Consider a node v that arrives at $\tau \leq t$
  - P{v stays in system after t } = $P(X \geq t - \tau)$
    - Where X is the departure time
  - $P(X \geq t - \tau) = 1 - P(X \leq t - \tau) = 1 - F_x(t - \tau)$
  - $1 - (1 - e^{\mu(t-\tau)}) = e^{\mu(t-\tau)} = e^{(t-\tau)/N}$
  - Let $p(t)$ be the probability that a node arriving during $[0, t]$ stay in system after t
  - $p(t) = P\{$ arriving by τ $\} \times P\{$ stay in system at t $\}$

$$p(t) = \frac{1}{t}\int_0^t e^{-(t-\tau)/N} d\tau = \frac{1}{t} N \left(1 - e^{-t/N}\right)$$

Theorem III.1 (cont.)

- E[no of peers in system at t] = $E[|V_t|] = \lambda p(t)t$
- $= p(t)t = N(1 - e^{-t/N})$
- $t = \Omega(N)$, $t \geq aN$
  - After some initial time t that is sufficient to have N arrivals
- $E[|V_t|] = N(1 - e^{-a})$, $\Theta(N)$
- When $t/N \rightarrow \infty$
- $E[|V_t|] = N - o(N) = N + o(N)$
- We can now use a tail bound for Poisson distribution to show that for $t = \Omega(N)$ $$\Pr\left(||V_t| - E[|V_t|]| \leq \sqrt{bN \log N}\right) \geq 1 - \frac{1}{N^c}$$
- **Corollary 4.2.3.** *Let $X_1, \ldots, X_n$ be independent Poisson trials such that* $\text{Prob}[X_i = 1] = p_i$. *Let* $X = \sum_{i \in [n]} X_i$ *and* $\mu = \mathbf{E}[X]$. *For* $0 < \delta < 1$,

$$\text{Prob}\left[|X - \mu| \geq \delta \cdot \mu\right] \leq 2 \cdot e^{-\mu \cdot \delta^2 / 3}.$$

Theorem III.2

Theorem III.2: 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 $N' \pm o(N')$ nodes.

- Proof
  - Suppose M nodes were in system at τ
  - E[no of peer at t] = $M \times P${ a peers remains at t that were there by τ } + no of new pears remain at t that arrived at τ
    - Because of memoryless property Part 1 is like starting at τ

$$Me^{-\frac{(t-\tau)}{N'}} + N'(1-e^{-\frac{t-\tau}{N'}}) = N' + (M-N')e^{-\frac{(t-\tau)}{N'}}$$

  - As $(t - \tau)/N \rightarrow \infty$
  - $= N' \pm o(M - N')$

Lemma III.1

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

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

- Assume $t \geq N$
- No of new nodes arriving in $[t - N, t]$
  - For a Poisson process no of arrivals by $\Delta t = \lambda \Delta t + o(\Delta t)$
  - $= (t - (t - N)) + o(t - (t - N)) = N + o(N) = N(1 + o(1))$
- Hence, no of new connections to cache nodes = $DN(1 + o(1))$
- E[no of connections arriving in a unit time] = $1 + o(1)$
- System has $N + o(N)$ nodes at any time, Theorem III.1
- Therefore, E[no of peers leaving at unit time] = $1 + o(1)$

Lemma III.1 (cont.)

- Consider reconnections
- E[no of reconnections to cache nodes in unit time] =
  - # of nodes leaving × P{ neighbor leaving} × P{ reconnection } + # of nodes leaving × P{ preferred connection leaving } × P{ reconnecting }

$$\sum_{v \in V} \left((1+o(1)) \frac{d(v)}{N} \frac{D}{d(v)} + (1+o(1)) \frac{1}{N} \right) \quad = (D+1)(1+o(1))$$

  - Above is an upper bound as we assume a peer leave in every time unit
  - E[no of nodes leaving during time interval] $\leq N + o(N)$
- Total no of reconnections to cache nodes in $[t - N, t]$
- $= (t - (t - N))(D + 1)(1 + o(1)) = N(D + 1)(1 + o(1))$
- Let $u_1, u_2, \ldots, u_l$ be the nodes that left the network
- Let $X_{v,ui} = 1$ when v makes a reconnection when u_i left network

Lemma III.1 (cont.)

- Actual no of reconnections = $E\left[\sum_{i=1}^{\ell}\sum_{v} X_{v,u_i}\right] \leq N(D+1)(1+o(1))$
- Maximum no of new & reconnections to cache nodes
  - $DN(1 + o(1)) + (D + 1)N(1 + o(1)) = (2D + 1)N(1 + o(1))$
- Each cache node is capable of accepting $C - D$ connections
- No cache nodes need in $[t - N, t] = \{(2D + 1)N(1 + o(1)\}/(C - D)$
- All these nodes will become c-nodes
- We have $N + o(N)$ nodes in network at any time
- So, no of remaining d-nodes

$$N(1+o(1)) - \frac{(2D+1)N(1+o(1))}{C-D} = \left(1 - \frac{2D+1}{C-D}\right)N(1+o(1))$$

  - For above to satisfy our requirement $2D+1 < C - D \Rightarrow C > 3D+1$

Lemma III.2

Lemma III.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 $\frac{(2D+1)c}{(C-D)K} \log N(1 \pm \delta)$. ■

- $Z(v)$ – Set of nodes that occupied v's slot in $[t - c \log N, t]$
- From Lemma III.1 E[total no of connections to cache nodes]
  - $(2D + 1)(c \log N)(1 + o(1))$
- E[no of connections to a cache node] = $E[X]$
  - $(2D + 1)(c \log N)(1+o(1))/K$
- No of cache nodes needed $= \dfrac{(2D+1)(c\log N)\big(1+o(1)\big)}{K(C-D)}$

$$= \frac{(2D+1)(c\log N)\big(1+o(1)\big)}{K(C-D)} = \frac{(2D+1)(c\log N)(1\pm\delta)}{K(C-D)} = d\log N$$

Lemma III.2 (cont.)

- $E[X] = (2D + 1)(c \log N)(1+o(1))/K$, with high probability

For any $\delta > 0$ we have the following large deviation bounds (also known as *Chernoff bounds*):

$$\Pr(X > (1+\delta)\mu) < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu} \tag{2.5}$$

For $0 < \delta < 1$ we have the following bounds:

$$\Pr(X < (1-\delta)\mu) \leq e^{-\mu\delta^2/2} \tag{2.6}$$

$$\Pr(X > (1+\delta)\mu) \leq e^{-\mu\delta^2/3} \tag{2.7}$$

- For sufficiently large $E[X] = \mu$ above probability is low
  - For sufficiently large $c > 0$

Lemma III.3

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

- Let $v_1, v_2, \ldots, v_k$ be the set of cache nodes at time t
- From Lemma III.2 $|v_i| = d \log n$
  - Where $d = \frac{(2D+1)(1 \pm \delta)}{K(C-D)}$
- Consider time interval $[t - c \log n, t]$
- $P\{$ node doesn't leave by t $\}$
  - $P\{$ departure time $\geq c \log n\} = e^{-(c \log N)/N}$
- There are K cache nodes & each will be replaced by $|Z(v_i)|$
- $P\{$ All cache nodes don't leave $\} = \left(e^{-c \log N/N}\right)^{K|Z(v_i)|}$

$$\left(e^{-c \log N/N}\right)^{Kd \log N} = e^{-Kcd \log^2 N/N}$$

Lemma III.3 (cont.)

$$e^{-\frac{(Kcd\log^2 N)}{N}} \geq 1 - O\left(\frac{\log^2 N}{N}\right)$$

- Suppose v leave cache at t
- Replace v by a d-node neighbor in $Z(v)$
- Z(v) received at least $Dc \log N(1 + o(1))/K$ connections
  - From Lemma III.1
- Among these no more than $|Z(v)|$ could enter cache & become c-nodes
- So there are $Dc\ log\ N(1 + o(1))/K$ - $|Z(v)|$ remaining d-nodes
  - $Dc\ log\ N(1 + o(1))/K - d \log N = \log N\{Dc(1 - o(1))/K - d\}$
  - So we need to examine $O(\log N)$ nodes

Lemma III.4

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

- A d-node is always connected to a c-node
- Hence we only need to consider connectivity of c-nodes
- A c-node is either in cache or it's connected to a cache node through preferred connection
  - v's preferred cache node u may become a c-node. Still v maintains a preferred connection to u. similarly u (after leaving cache) maintains a connection to it's preferred cache node w
  - These links continue unless a node leaves
  - If a node leave, neighbor(s) that had the preferred connection initiate another connection to a cache node

Lemma III.5

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

- Let 2 cache nodes be u & v
- $Z(v)$ – Set of nodes that occupied v's slot in $[t - c \log N, t]$
- From Lemma III.2 $|Z(v)| = d \log N$
- $P\{$ node doesn't leave by $t\ \}$
  - $P\{$ departure time $\geq c \log n\} = e^{-(c \log N)/N}$
- $P\{$ All $Z(v)$ nodes don't leave by $t\ \}$ = $\left(e^{-c \log N/N}\right)^{d \log N} = e^{-cd \log^2 N/N}$

$$\geq 1 - O\left(\log^2 N \big/ N\right)$$

Lemma III.5 (cont.)

- Because of preferred connections
  - If no node in $Z(v)$ leave, all of them are connected to v, same for u
  - Hence, $P\{$ $Z(v)$ is connected to a cache node $\}$ $\geq 1 - O\left(\log^2 N / N\right)$
- $P\{$ A new node not connecting $Z(u)$ & $Z(v)$ $\}$ = $1 - (D/K)^2$
  - $P\{$ connecting to a $Z(u)$ $\}$ = $P\{$ connecting to a $Z(v)$ $\}$ = D/K
  - $P\{$ connecting to a $Z(u)$ & $Z(v)$ $\}$ = $(D/K)^2$
- No of new nodes during $[t - c \log N, t] = c \log N$
- P { All new nodes don't connect to $Z(u)$ & $Z(v)$ } = $\left(1 - D^2/K^2\right)^{c \log N}$
  - $= O(1/N^c)$
- Hence there is a path between u & v

Theorem III.3

Theorem III.3: There is a constant c such that at any given time $t > c \log N$

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

- From Lemma III.4 & III.5 all the nodes are connected w.h.p
- Hence, graph G_t is connected w.h.p
- This theorem doesn't depend on the state of the network at time $t - c \log N$
- Hence, show that network can rapidly recover

Corollary III.1: There is a constant c such that if the network is disconnected at time t

$$Pr(\mathrm{G}_{\mathrm{t+c\log N}} \text{ is connected}) \geq 1 - O\left(\frac{\log^2 N}{N}\right).$$

Theorem III.4

Theorem III.4: 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 $N(1 - o(1))$.

- By Lemma III.4 all nodes are connected to some cache node
- From Theorem III.3, $P\{$ that network may not be connected$\}$
  - $O(\log^2 N/N)$
  - This is the probability that some cache node has fewer than $d \log N$ neighbors
- E[No of disconnected cache nodes] = $K\ O((\log^2 N)/N)$
- No of connected nodes = $N(1 + o(1)) - K\ O((\log^2 N)/N)$
  - $= N(1 + o(1))$

Theorem III.4 (cont.)

- P{ A new node is not connected to both $Z(u)$ & $Z(v)$}
  - $1 - D^2/K^2$
- P{ All new nodes don't connect $Z(u)$ & $Z(v)$}
  - $(1 - D^2/K^2)^{c \log N}$
- Possible no of connections between cache nodes
  - $K(K - 1)/2 = (K^2 - K)/2$
- Graph is disconnected if one of these pairs is disconnected
  - Each pair is independent
  - P{ graph disconnected } $= (K^2 - K)(1 - D^2/K^2)^{c \log N}/2$
- Hence, P{ graph is connected } $= 1 - (K^2 - K)(1 - D^2/K^2)^{c \log N}/2$
  - $= 1 - 1/N^c$

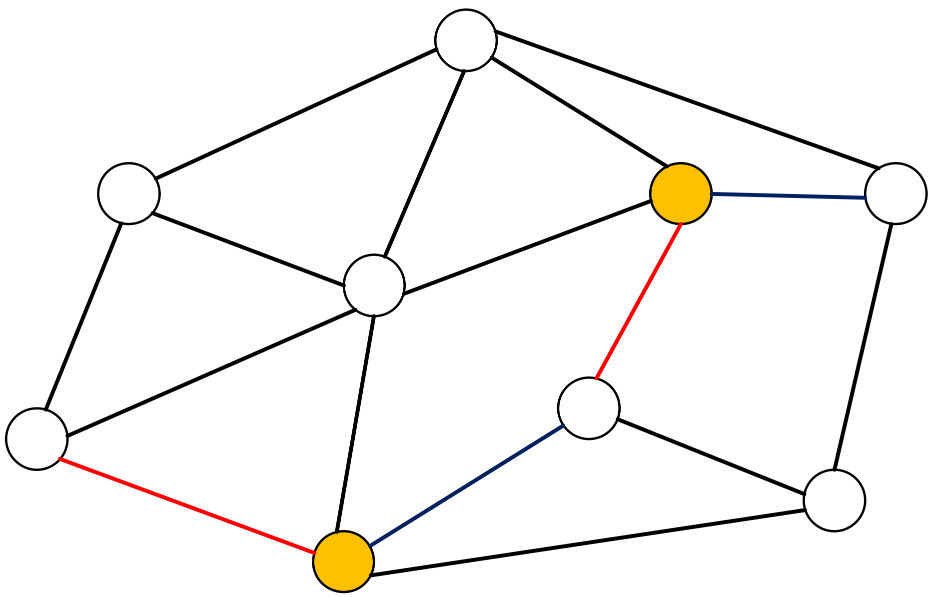
Theorem III.5

Theorem III.5: For any t, such that $t/N \to \infty$, w.h.p., the largest connected component of G_t has diameter $O(\log N)$. In particular, if the network is connected (which has probability $1 - O(\log^2 N/N)$), then w.h.p., its diameter is $O(\log N)$.

- A d-node is always connected to a c-node
- Hence, it's sufficient to consider connectivity of c-nodes
- Let f be a constant
- A cache node is called good, if it receives $r \geq f$ connections
  - All r connections are reconnection requests
  - All r connections are not preferred connections
  - r connections result for departure of r different nodes

Theorem III.5 (cont.)

- Color edges (links) of the graph using A, B_1, B_2
  - Randomly pick $^f/_2$ of the reconnection links of a good cache node & color them as B_1
  - Color another $^f/_2$ of reconnection links of a good cache node as B_2
  - Color all other links with A



Theorem III.5 (cont.)

- Theorem III.3 gives the probability that the network is connected using only A colored links
  - $1 - O(\log^2 N / N)$
  - Proof uses preferred connections & newly joined nodes
- Theorem III.4, size of the connected network is $N(1 + o(1))$
- A connections could grow arbitrary long
  - Reconnections (B_1, B_2) allow a way to reduce the distance to a cache node

Lemma III.6

Lemma III.6: Assume that node v enters the cache at time t, where $t/N \to \infty$. Then, for a sufficiently large choice of the constant C, the probability that v leaves the cache as a good node is at least $\gamma > 1/2$. Further, the f recolored edges of a good cache node are distributed uniformly at random among the nodes currently in the network. Furthermore, the probability that a c-node is good is independent of other c-nodes.

- E[no of connections to v from a new node] = D/K
- E[no of reconnections due to departure of a node] =

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

  - This imply all reconnections are for departure of different nodes
  - Each connection has a constant probability of being triggered by a unique node leaving the network

Lemma III.6 (cont.)

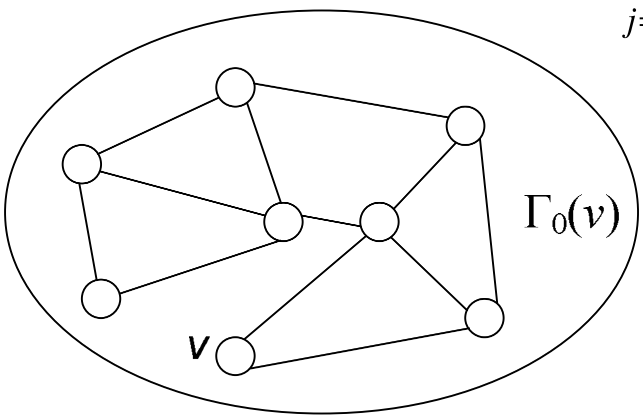
- E[conn. from a new node] = E[conn. from an old node]
- A cache node can accept $C - D$ new/reconnections
  - ½ of the connections are from old nodes
  - In minimum it will accept $(C - D)/2$ reconnections
- If C is sufficiently large, it could easily handle $r \geq f$ reconnections
  - In minimum, with probability ½, a cache node could node becomes a good node
  - If C is large, probability would further increase
  - Hence v would leave the cache as a good node with probability $\geq$ ½

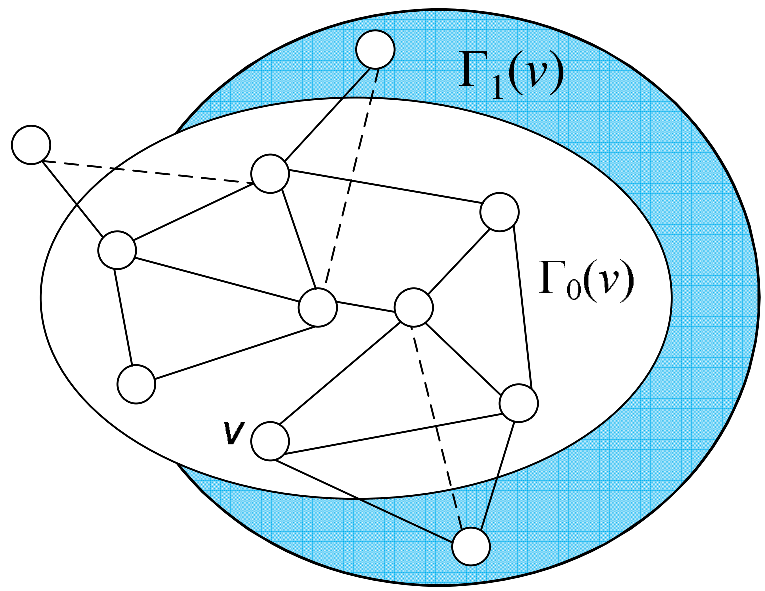
Lemma III.6 (cont.)

- E[no of connections form old node u to v] = $\frac{d(u)}{N}\frac{D}{d(u)}=\frac{D}{N}$
  - This needs to be divided by K ???
  - Each node leaves independently with identical ~ $exp(\mu)$
  - Each node in the network has equal probability of connecting to v
  - Independent of node degree
- A cache node stays in cache until it accept C connections
  - This behavior is independent of other cache nodes
  - Hence, whether a given cache node becomes a good node is independent of others

Lemma III.7

- Given a node v
- Let $\Gamma_0(v)$ be an arbitrary cluster of $d \log N$ c-nodes
- $v \in \Gamma_0(v)$
- This cluster has a diameter of $O(\log N)$ using only A edges
- Let $\Gamma_i(v)$ be all c-nodes in G_t that are connected to $\Gamma_{i-1}(v)$ using B_1 links & not in $\bigcup_{j=0}^{i-1} \Gamma_j(v)$

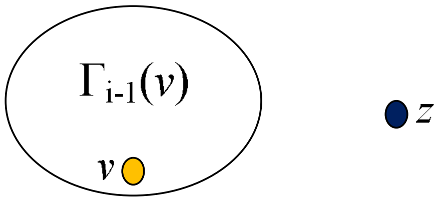




Lemma III.7 (cont.)

Lemma III.7: If $|\Gamma_{i-1}(v)| = o(N)$

$$Pr\{|\Gamma_i(v)| \geq 2|\Gamma_{i-1}(v)|\} \geq 1 - \frac{1}{N^5}.$$



- Let $W = \Gamma_i(v)$ & $w = |W|$
- Let z be a c-node such that $z \notin W \bigcup \left(\bigcup_{j=0}^{i-1} \Gamma_j(v) \right)$
  - Need to be a good cache node
- $P\{$ z is connected to W using B_1 edges $\}$
  - $P\{z$ being a good node$\}$ × $P\{$selecting a node$\}$ × no of connections used × no of nodes to connect to

$$\geq \frac{1}{2} \frac{1}{N(1+o(1))} \frac{f}{2} w = \frac{fw}{4N}(1+o(1))$$

Lemma III.7 (cont.)

- Let $Y = |\Gamma_i(v)|$ be number of nodes (like z) that are outside W & connected to W by B_1
- $$E[Y] = \sum_{|V|} \frac{fw}{4N}\left(1 + o(1)\right) = \frac{fw}{4}\left(1 + o(1)\right)$$
- Let $w_1, w_2, \ldots$ be an enumeration of nodes in W
- Let $N(w_i)$ be set of neighbors of w_i that are connected by B_1
- $N(w_i)$ are not independent, so use martingale based analysis
- Define exposure martingale such that $Z_0, Z_1, \ldots$ such that
- $Z_0 = E[Y], Z_i = E[Y \mid N(w_1), N(w_2), \ldots, N(w_i)]$
  - Above reflects no of outside c-nodes connected, given subset of nodes in W by B_1 links

Lemma III.7 (cont.)

- Degree of all nodes are bounded by *C*
- $|Z_i - Z_{i-1}| < C$
  - At least 1 connection is already inside
- Using Azuma's inequality

Then *Azuma's inequality* gives for all $t \geq 0$ and any $\lambda > 0$,

$$\Pr(|X_t - X_0| \geq \lambda) \leq 2e^{-\frac{\lambda^2}{2\sum_{k=1}^{t} c_k^2}}$$

$$Pr\left\{|Y - E[Y]| \geq \frac{f}{8}\frac{\sqrt{w}}{C}C\sqrt{w}\right\} \leq 2e^{-(f^2/128C^2)w} \leq \frac{1}{N^5}.$$

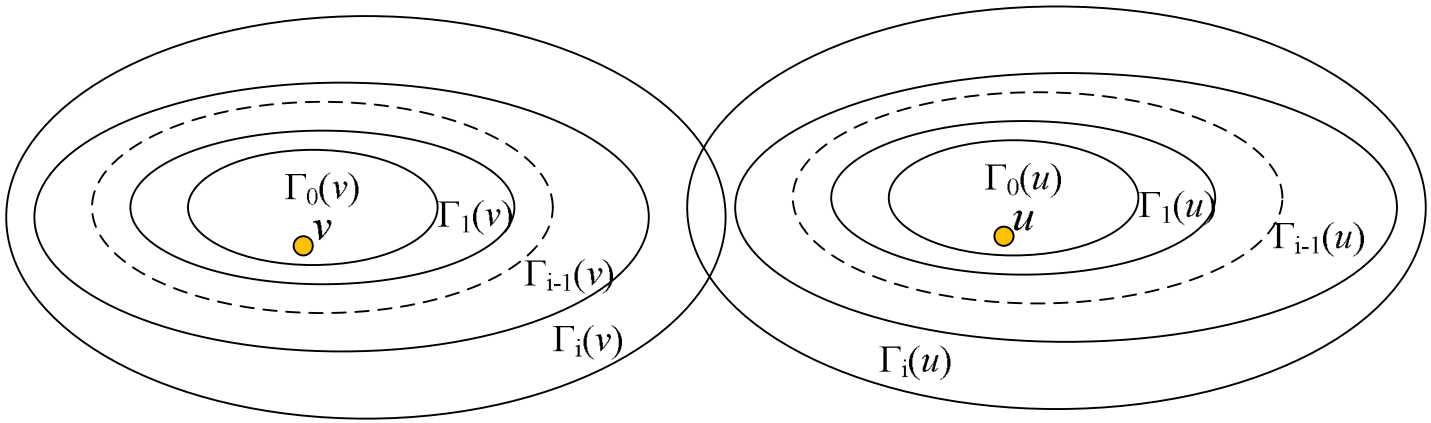
- This imply that Y is concentrated around ½ of mean w.h.p

Lemma III.7 (cont.)

$$P\left\{ \left| Y - E[Y] \right| \le {}^{fw}\!/_{8} \right\} \ge 1 - {}^{1}\!/_{N^5}$$

- $fw/8 \approx E[Y]/2$
- Therefore, $Y \in [\ E[Y] - E[Y]/2, E[Y] + E[Y]/2\ ]$ w.h.p
- $Y \in \left[\frac{fw}{4}\left(1 + o(1)\right) - \frac{fw}{8}, \frac{fw}{4}\left(1 + o(1)\right) + \frac{fw}{8} \right]$
- $|Y| \ge \frac{fw}{2}$, ≥ is because $|Y|$ could be above the given range
- For above to be satisfied $f \ge 4$

Theorem III.5 (cont.)



- Let u & v be any 2 c-nodes in the network
- Let $\Gamma_0(v)$ & $\Gamma_0(u)$ be the clusters they form by connecting c-nodes using A colored links
  - Each has a diameter of $O(\log N)$
- Our goal is to show that distance between any 2 c-nodes is $O(\log n)$
  - Expand the cluster by connecting nodes using B1
  - Then show that 2 cluster would overlap

Theorem III.5 (cont.)

- From Lemma III.7 $|\Gamma_i(v)| \geq |\Gamma_{i\text{-}1}(v)|$, w.h.p
  - $|\Gamma_1(v)| \geq 2|\Gamma_0(v)|$
  - $|\Gamma_2(v)| \geq 2|\Gamma_1(v)| \geq 4|\Gamma_0(v)|$
  - $|\Gamma_3(v)| \geq 2|\Gamma_2(v)| \geq 8|\Gamma_0(v)|$
  - ….
  - $|\Gamma_n(v)| \geq 2|\Gamma_{n\text{-}1}(v)| \geq 2^n|\Gamma_0(v)|$
- Apply Lemma III.7 $O(\log N)$ times, i.e., $c \log N$ times
  - $|\Gamma_{c \log N}(v)| \geq 2^{c \log N}|\Gamma_0(v)|$
- $P\{$ that $|\Gamma_i(v)|$ is not $2\times$ as $|\Gamma_{i\text{-}1}(v)|$ $\} \leq 1/N^5$
  - $P\{$ that a $c \log N$ hop neighborhood does not satisfy $2\times$ requirement$\}$
  - $\leq (c \log N)(1/N^5) = O(\log N/N^5)$
    - If at least 1 of the circles are not $2\times$ as previous one our goal fails
  - $P\{$ $2\times$ requirement hold for a $d \log n$ neighborhood$\}$ $= 1 - O((\log N)/N^5)$

Theorem III.5 (cont.)

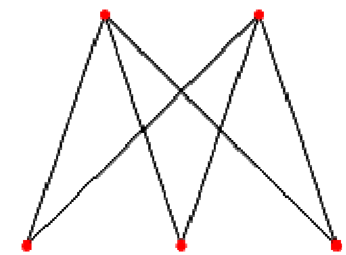
- From Lemma III.7 it can be shown that $|\Gamma_i(v)| \geq \frac{fw}{2}$
  - Where w is $|\Gamma_{i-1}(v)|$
- If $|\Gamma_0(v)| = d \log N$
  - $|\Gamma_{c \log N}(v)| \geq 2^{c \log N}|\Gamma_0(v)| = 2^{c \log N} d \log N \approx N^{1/2} \log N$
- $P\{$ that 2 nodes are connected using B_1 links$\} = f/(2N)$
  - Only ½ of the connections are considered
- $P\{$ that 2 nodes are disconnected using B_1 links$\} = 1 - f/(2N)$
- $P\{$ that all nodes in $\Gamma_{c \log N}(v)$ & $\Gamma_{c \log N}(u)$ are disconnected$\}$

$$\left(1-\frac{f}{2N}\right)^{\left(\sqrt{N}\log N\right)^2} = \left(1-\frac{f}{2N}\right)^{N\log^2 N}$$

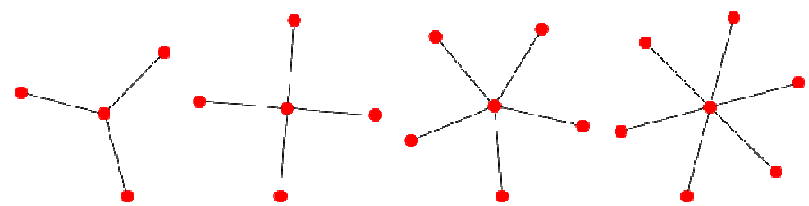
Therefore, with probability $1 - O(\log N/N^5)$ any 2 c-nodes are connected by a path length $O(\log N)$

Lemma IV.1

Lemma IV.1: At any time $t \geq c$, where c is a sufficiently large fixed constant, there is a constant probability (i.e., independent of N) that there exists a subgraph of type H in G_t.

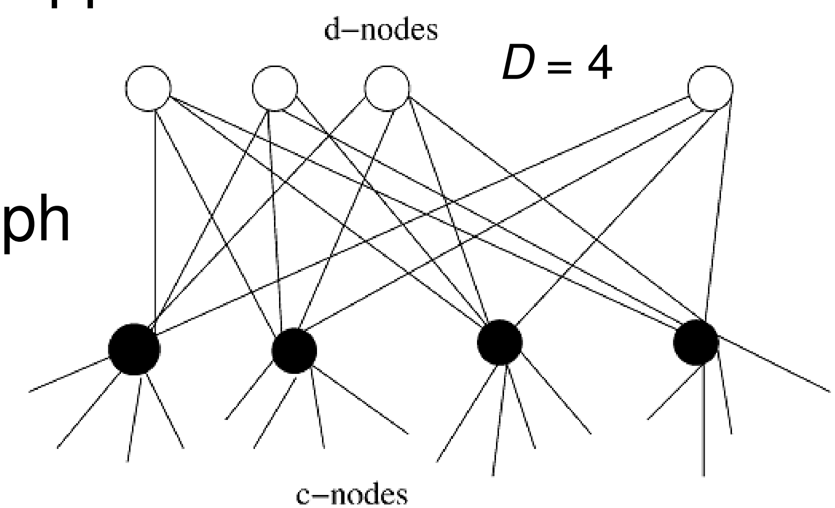


- Let H be a complete bipartite network
  - Graph with 2 disjoint sets of vertices
  - Elements in 2 sets are directly connected
  - Each element in 1 set connect to every element in another
- P2P network could have sub graph of type H
  - Between D d-nodes & D c-nodes
  - Could occur when D new nodes join D cache nodes that become c-nodes



Lemma IV.1 (cont.)

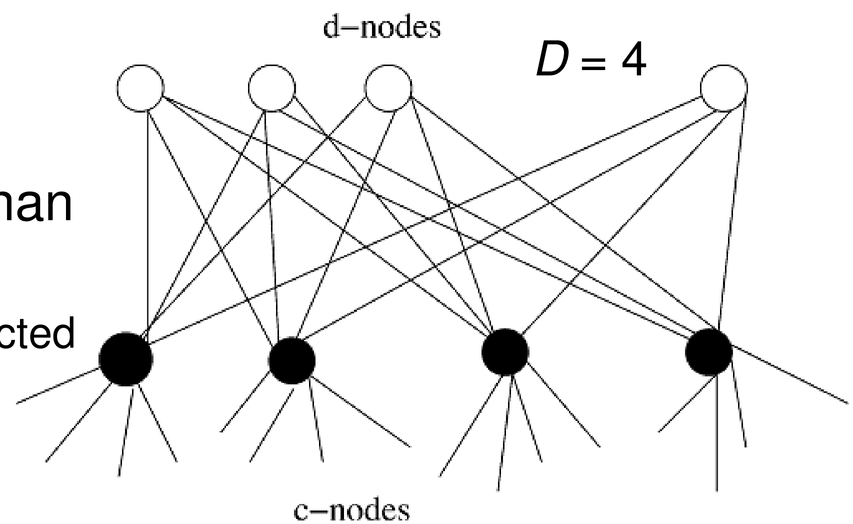
- Conditions for formation of a complete bipartite network
  1. There is a set (S) of D cache nodes each having degree D at time $t - D$
     - These are new nodes in cache & yet to accept connections
  2. There are no deletions in the network during the interval $[t - D, t]$
  3. A set (T) of D new nodes arrive during interval $[t - D, t]$
  4. All incoming nodes of T choose to connect to D cache nodes in S
- Each of the above events could happen with constant probability (> 0)
  - Independent of N
- Network could form a type H graph



Lemma IV.2

Lemma IV.2: Consider the network G_t, for $t > N$. There is a constant probability that there exists a small (i.e., constant size) isolated component.

- From Lemma IV.1 it's possible to have a complete bipartite network H
- Let sub graph F of type H occur at $t - N$
- F will be isolated if
  - All its $2D$ nodes stay in system by t
  - All c-nodes loose neighbors other than new d-nodes
    - At most $D(C - D)$ such nodes are connected
  - c-nodes don't try to reconnect



Lemma IV.2 (cont.)

- $P\{$ all $2D$ nodes survive interval $[t - N, t]$ $\}$ $= (e^{-N/N})^{2D} = e^{-2D}$
- $P\{$ a neighbor retains after interval $[t - N, t]$ $\} = e^{-N/N} = e^{-1}$
- $P\{$ a neighbor leave after interval $[t - N, t]$ $\} = 1 - e^{-1}$
- $P\{$ all neighbors leave after interval $[t - N, t]$ $\} = (1 - e^{-1})^{D(C - D)}$
- $P\{$ Reconnection$\} = D/d(v)$
- Maximum $P\{$ Reconnection $\} = D/(D + 1)$
  - Has a minimum of D connections as they are connected to D new nodes
- $P\{$No reconnection $\} = 1 - D/(D + 1)$
- $P\{$No reconnection for loss of all neighbors$\} = (1 - {}^{D}/_{(D + 1)})^{D(C-D)}$

$$e^{-2D}\left(1-e^{-1}\right)^{D(C-D)}\left(1-{}^{D}\!/_{D+1}\right)^{D(C-D)} = \Theta(1)$$

Theorem IV.1

Theorem IV.1: The expected number of small isolated components in the network at any time $t > N$ is $\Omega(N)$, when there are no preferred connections.

- Let S be set of new nodes arrived between $[t-N, t-N/2]$
- Let $v \in S$ be a node that arrived at t'
- From Lemma IV.1 & IV.2, there is a nonzero probability that $v \in F$
  - F is a complete bipartite network
  - From Lemma IV.2, F has a constant probability of being isolated at t
- Let indicator variable X_v denote whether v is in F or not

$$E\left[\sum_{v \in S} X_v\right] = E[X_1] + E[X_2] + \ldots + E[X_{|S|}]$$

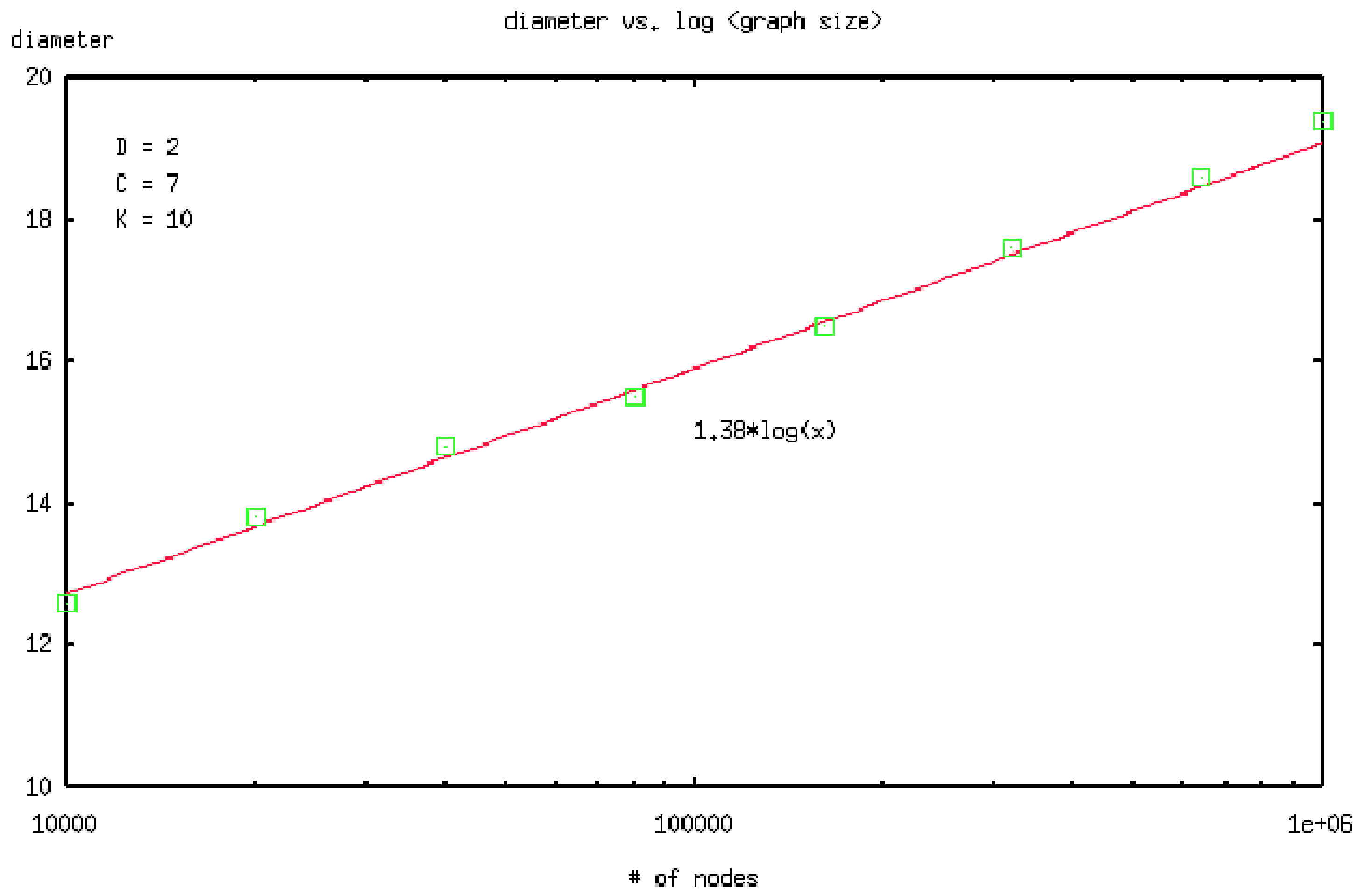
Theorem IV.1 (cont.)

- Let c be the constant probability of a node belonging to S
- $E[X_v] = 1 \times c + 0 \times (1 - c) = c$

$$E\left[\sum_{v \in S} X_v\right] = E[X_1] + E[X_2] + \ldots + E[X_{|S|}] = c\,|S|$$

- $|S| = N/2$
  - Length of time interval is $N/2$
- $$E\left[\sum_{v \in S} X_v\right] = cN/2$$
- There could be many more sub graphs $\geq cN/2$
  - $\Omega(N)$

Diameter vs. size

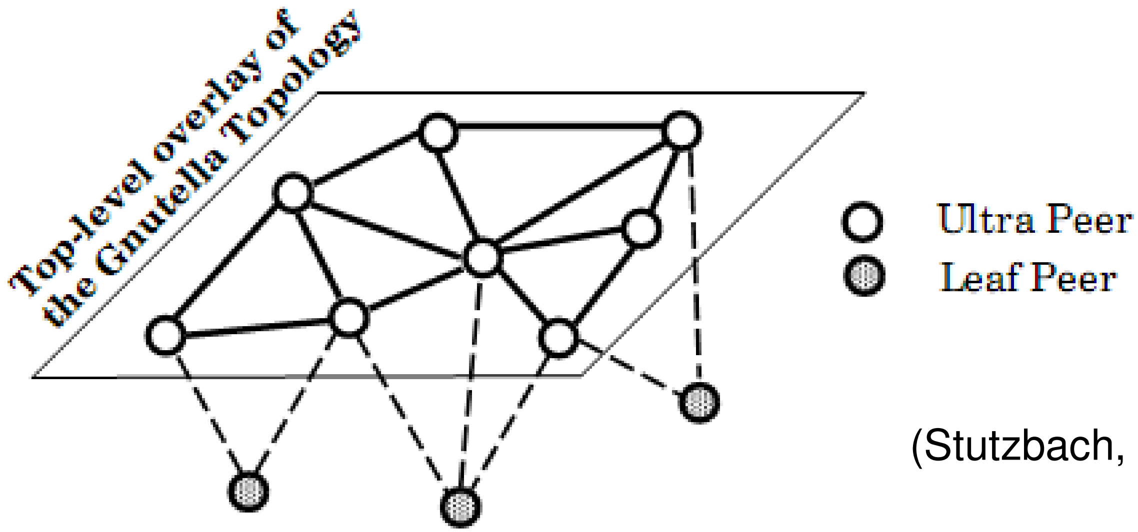


G. Pandurangan, “Protocol for building low-diameter P2P networks”

Backup Slides

A Scalable, Commodity, Data
Center Network Architecture

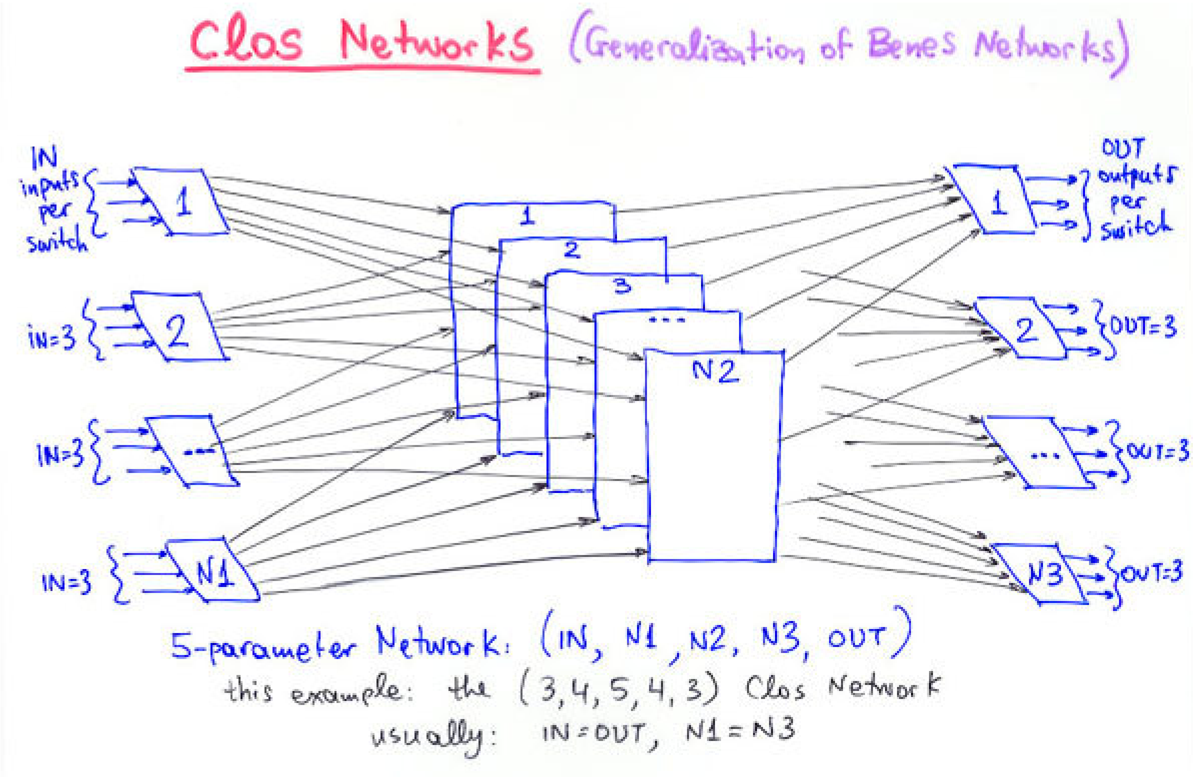
New Gnutella



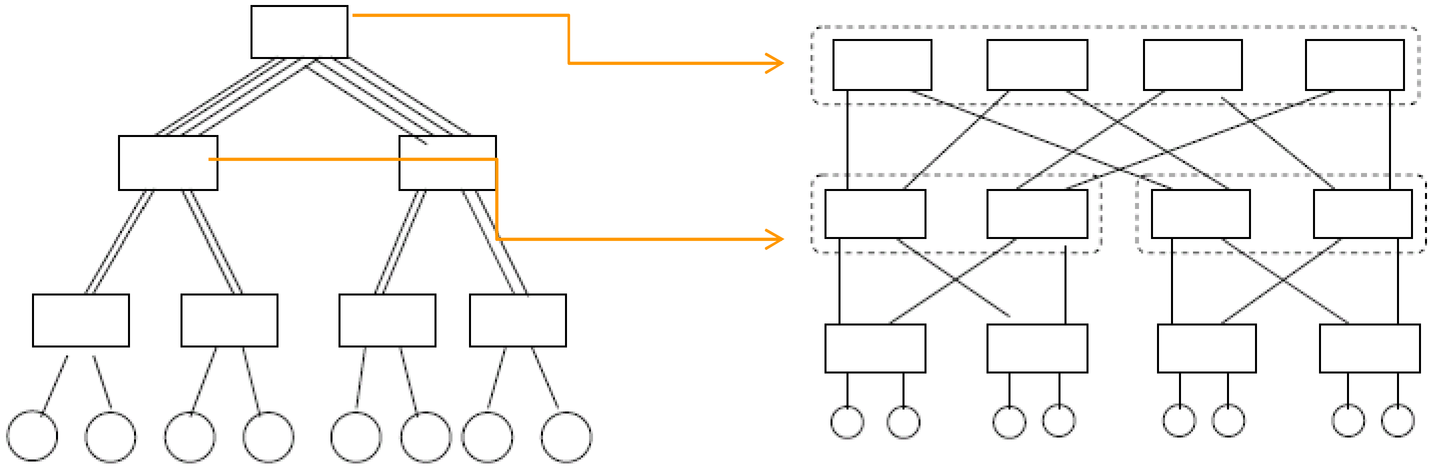
Gnutella V0.6

(Stutzbach, 2005)

Clos network

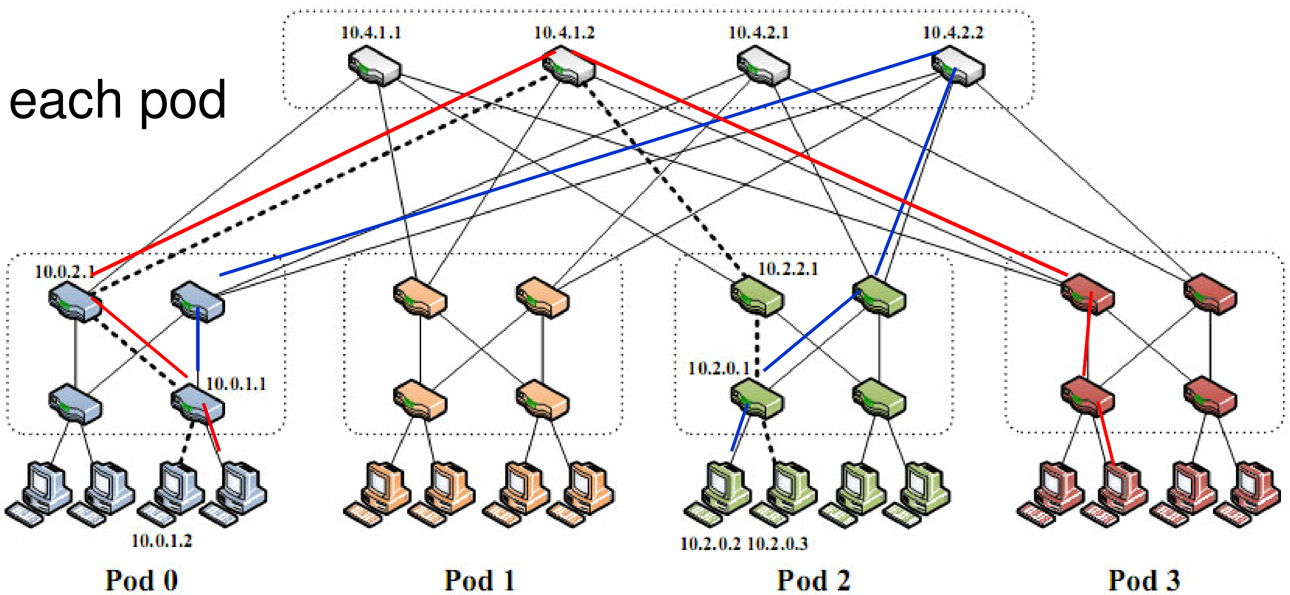


Fat tree



Routing table (cont.)

- Central entity assigns routing table for each switch
- Pod switches
  - $k/2$ prefixes for subnets in same pod
    - Only in top aggregation layer switches
  - $k/2$ suffixes for hosts in other pods/subnets
    - Output port is $(ID - 2 + switch) \bmod (k/2) + k/2$
- Core switches
  - k, /16 entries for each pod



Routing table fill up algorithms

```
foreach pod x in [0, k − 1] do
    foreach switch z in [(k/2), k − 1] do
        foreach subnet i in [0, (k/2) − 1] do
            addPrefix(10.x.z.1, 10.x.i.0/24, i);
        end
        addPrefix(10.x.z.1, 0.0.0.0/0, 0);
        foreach host ID i in [2, (k/2) + 1] do
            addSuffix(10.x.z.1, 0.0.0.i/8,
            (i − 2 + z)mod(k/2) + (k/2));
        end
    end
end
```

Algorithm 1: Generating aggregation switch routing tables. Assume Function signatures *addPrefix(switch, prefix, port)*, *addSuffix(switch, suffix, port)* and *addSuffix* adds a second-level suffix to the last-added first-level prefix.

```
foreach j in [1, (k/2)] do
    foreach i in [1, (k/2)] do
        foreach destination pod x in [0, (k/2) − 1] do
            addPrefix(10.k.j.i,10.x.0.0/16, x);
        end
    end
end
```

Algorithm 2: Generating core switch routing tables.

Fault tolerance

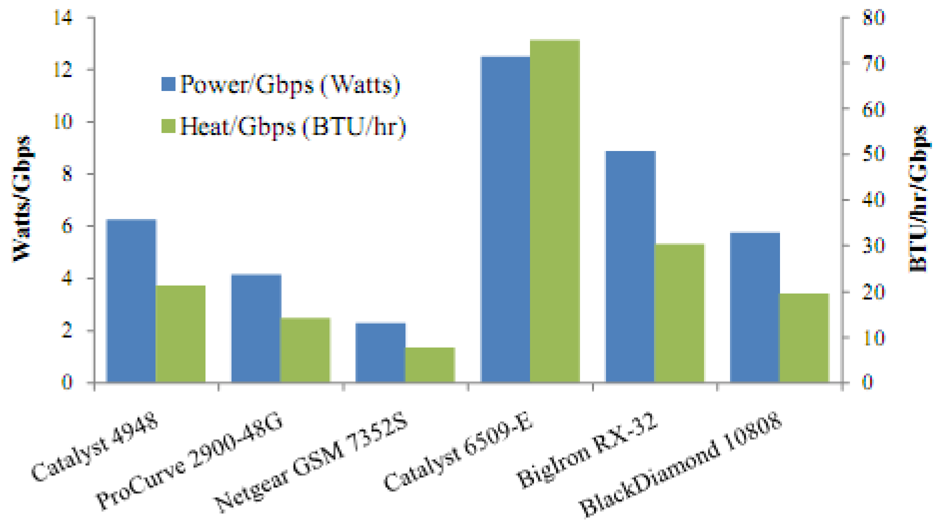
- Redundant links allow routing around a failure
- Need to keep track of state of each link
- Could withstand
  - Between lower-upper layer switches in a pod
    - Outgoing inter-pod & intra-pod – skip the link
    - Intra-pod using top layer – source skip top layer switch if possible
    - Inter-pod coming into top layer – ask the core switch to change → core layer ask top-layer of sender to change
  - Between upper & core layer switches
    - Outgoing inter-pod – select another core switch
    - Incoming inter-pod – core switch ask sending pods top layer to change
  - Failure between lower layer & PCs can't be handle without redundant switches/ports
- Flow scheduling make these problems easy to handle

Flow classifier heuristic

```
// Call on every incoming packet
IncomingPacket(packet)
begin
    Hash source and destination IP fields of packet;
    // Have we seen this flow before?
    if seen(hash) then
        Lookup previously assigned port x;
        Send packet on port x;
    else
        Record the new flow f;
        Assign f to the least-loaded upward port x;
        Send the packet on port x;
    end
end
// Call every t seconds
RearrangeFlows()
begin
    for i=0 to 2 do
        Find upward ports p_max and p_min with the largest and
        smallest aggregate outgoing traffic, respectively;
        Calculate D, the difference between p_max and p_min;
        Find the largest flow f assigned to port p_max whose size
        is smaller than D;
        if such a flow exists then
            Switch the output port of flow f to p_min;
        end
    end
end
```

Algorithm 3: The flow classifier heuristic. For the experiments in Section 5, t is 1 second.

Power & heat



- Last 3 switches have all 10 Gbps ports

Other

Comparison of 2 papers

- 2 different application domains
- Both focus on scalable topology construction & maintenance without high bandwidth links
- Multiple paths to a destination
  - How to connect to peers such that effective bandwidth is high
  - Paper 1 shows this for a static network
- Lower diameter & bounded node degree is important
  - Ability to reach majority of peers, no hot spots
- P2P is an alternative for some of the data center applications – e.g., BOINC, MOINC

Properties of a Poisson process

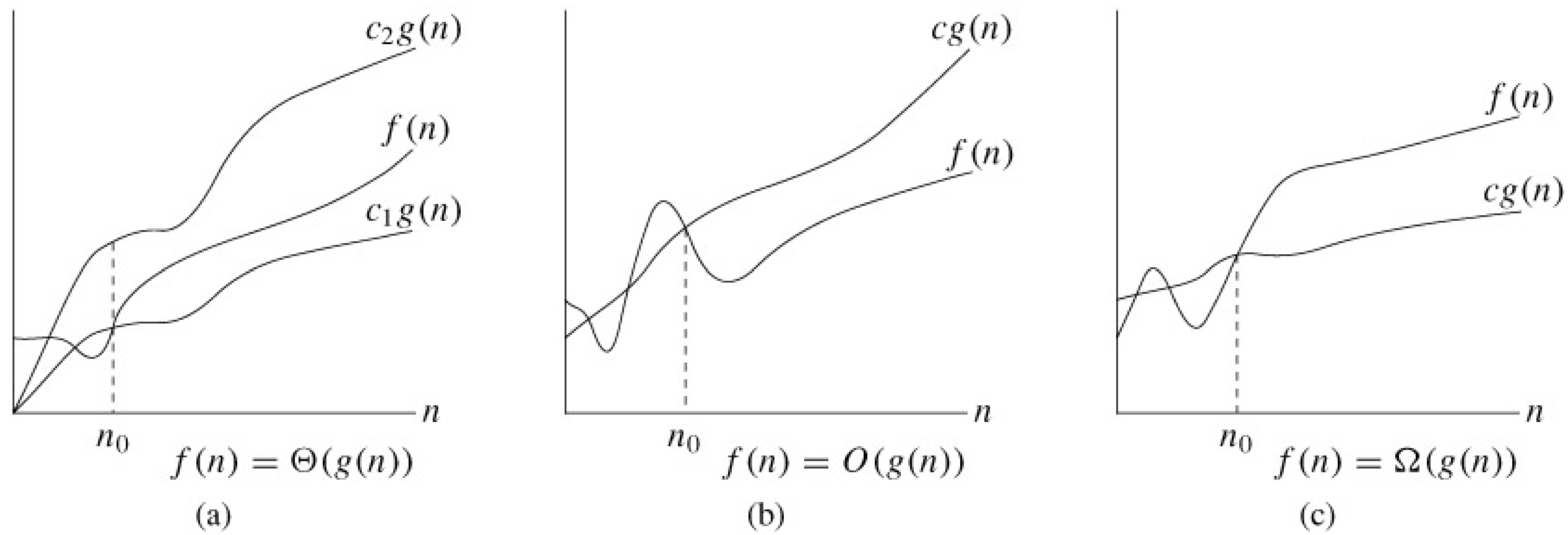
- A counting process $\{N_t, t \geq 0\}$ is a Poisson process if
  - $N_0 = 0$
  - $\{N_t, t \geq 0\}$ has stationary independent increment
    - N_{t1}-N_{s1} is independent from N_{t2}-N_{s2}
    - Memoryless
  - $P\{N_{\Delta t} = 1\} = \lambda \Delta t + o(\Delta t)$
  - $P\{N_{\Delta t} = 2\} = o(\Delta t)$
  - Inter arrival times are independently & identically distributed set of exponentially distributed random variables

$$\lambda = \lim_{\Delta t \to 0} \frac{P\{N_{t+\Delta t} = N_t + 1\}}{\Delta t}$$

$$\lim_{\Delta t \to 0} \frac{P\{N_{t+\Delta t} = N_t + 2\}}{\Delta t} = 0$$

- $o(\Delta t)$ is such that $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$

O, Θ, & Ω



$c_2g(n)$
$f(n)$
$c_1g(n)$
n
n_0
$f(n) = \Theta(g(n))$
(a)
$cg(n)$
$f(n)$
n
n_0
$f(n) = O(g(n))$
(b)
$f(n)$
$cg(n)$
n
n_0
$f(n) = \Omega(g(n))$
(c)