PhD Qualifying Exam

Dilum Bandara

Department of Electrical and Computer Engineering

July 9, 2009

Graduate Committee

- Prof. Anura P. Jayasumana (Advisor)
- Prof. V. Chandrasekar
- Prof. Daniel F. Massey
- Prof. Indrajit Ray

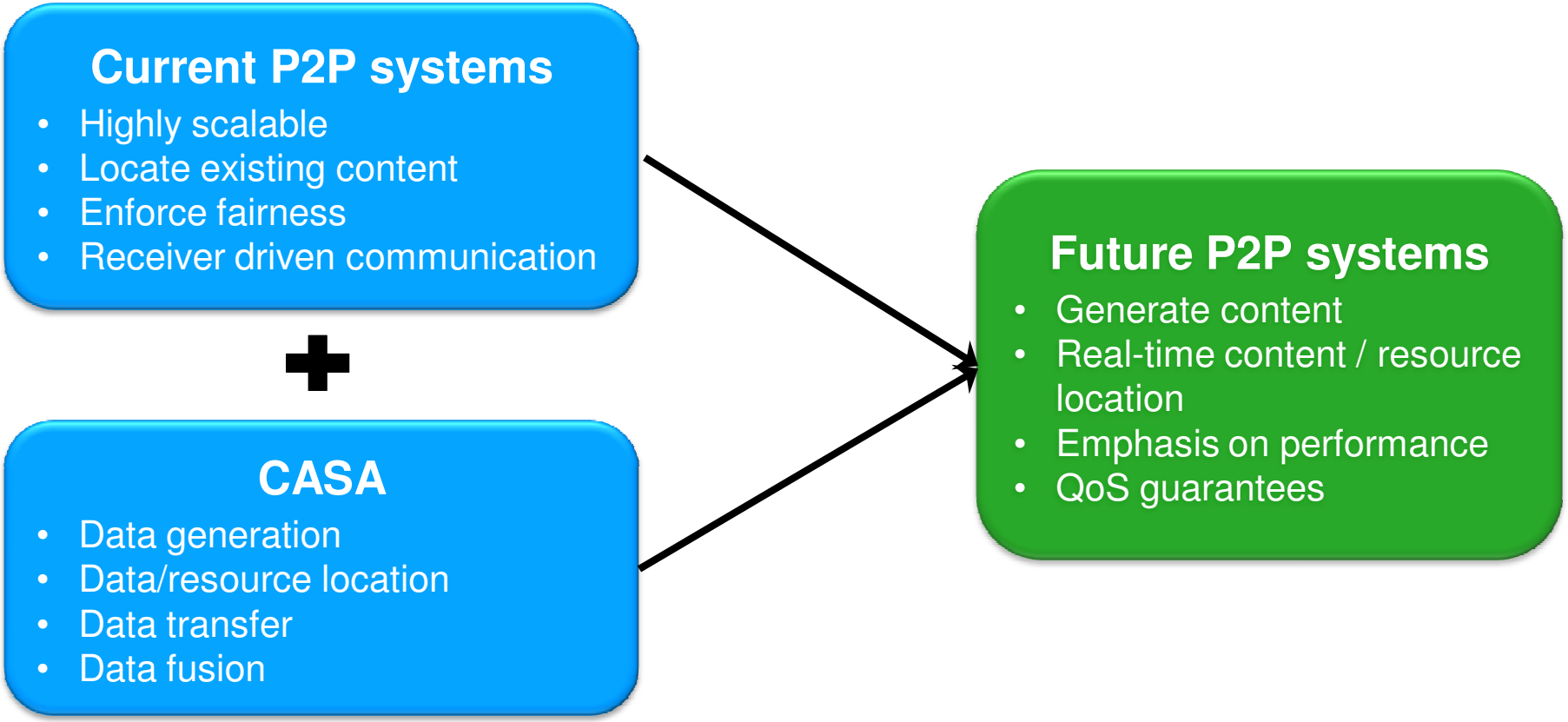
- This work is supported by the Engineering Research Center program of the National Science Foundation under NSF award number 0313747.

Exam papers

1. M. Al-Fares, A. Loukissas, and A. Vahdat, "**A scalable, commodity, data center network architecture**," In Proc. of the ACM SIGCOMM Conference, Seattle, WA, Aug. 2008.

2. Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal, "**Building low-diameter peer-to-peer networks**," IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, Aug. 2003, pp. 995-1002.

P2P in Collaborative Adaptive Sensing of the Atmosphere (CASA)



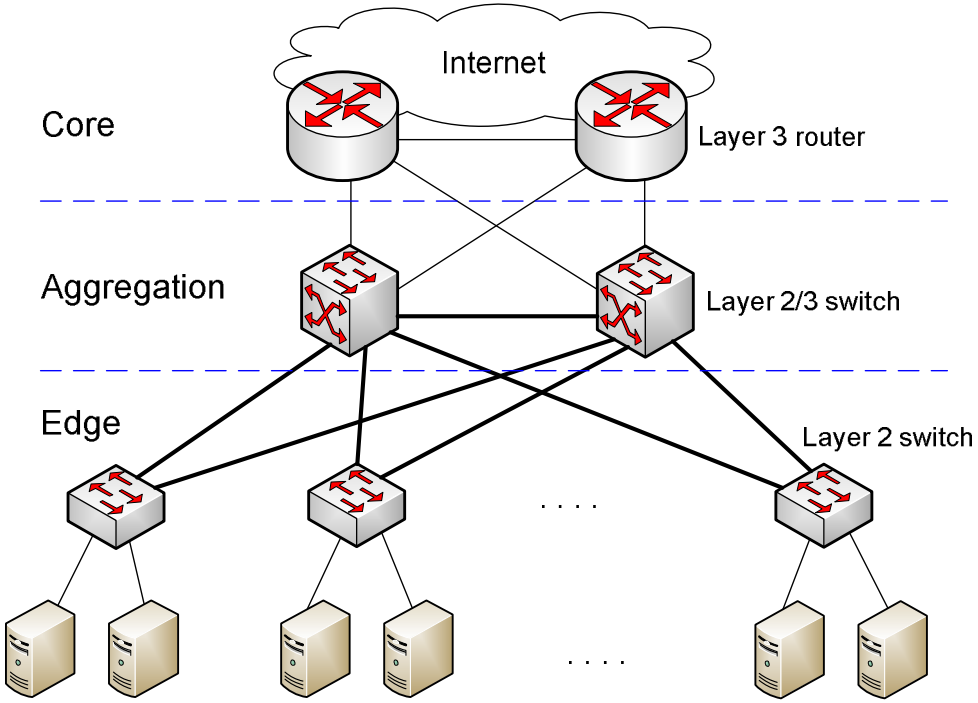
- 2 papers – Scalable topology construction without high bandwidth links
- Bounded node degree & lower diameter are desirable

A Scalable, Commodity, Data Center Network Architecture

M. Al-Fares, A. Loukissas, and A. Vahdat

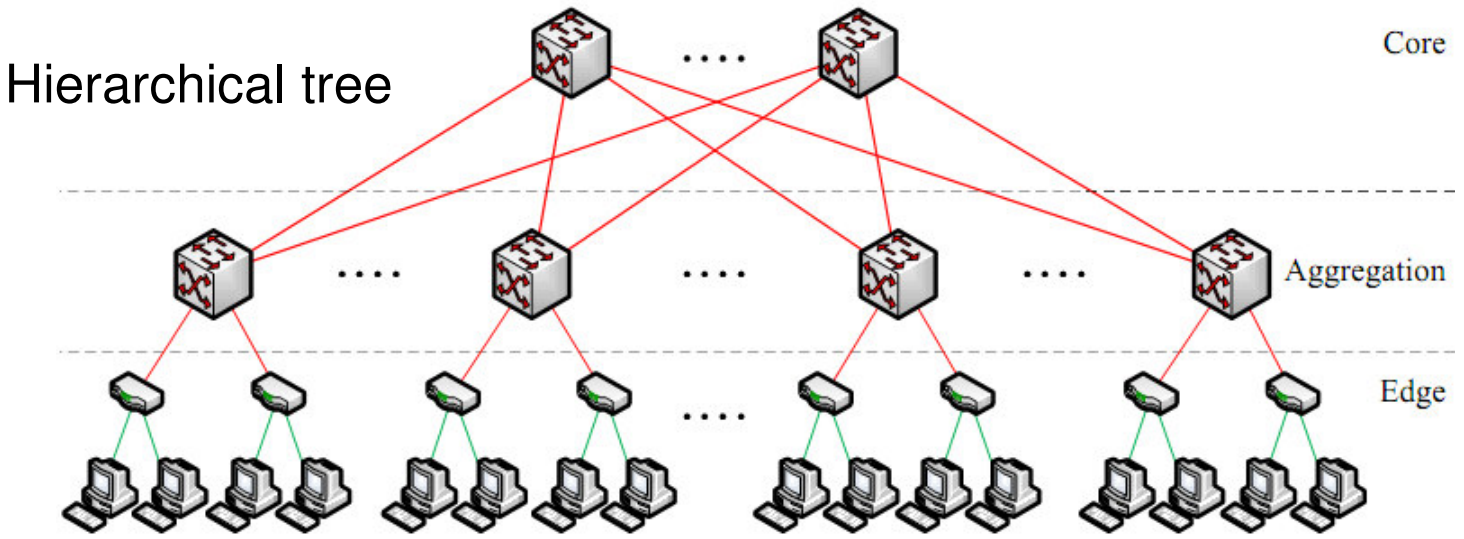
In Proc. of the ACM SIGCOMM Conference, Seattle, WA, Aug. 2008.

Common data center architecture



- 10s of 1000s of servers in a cluster
- External traffic
- Internal traffic
- Inter-node communication bandwidth is the bottleneck

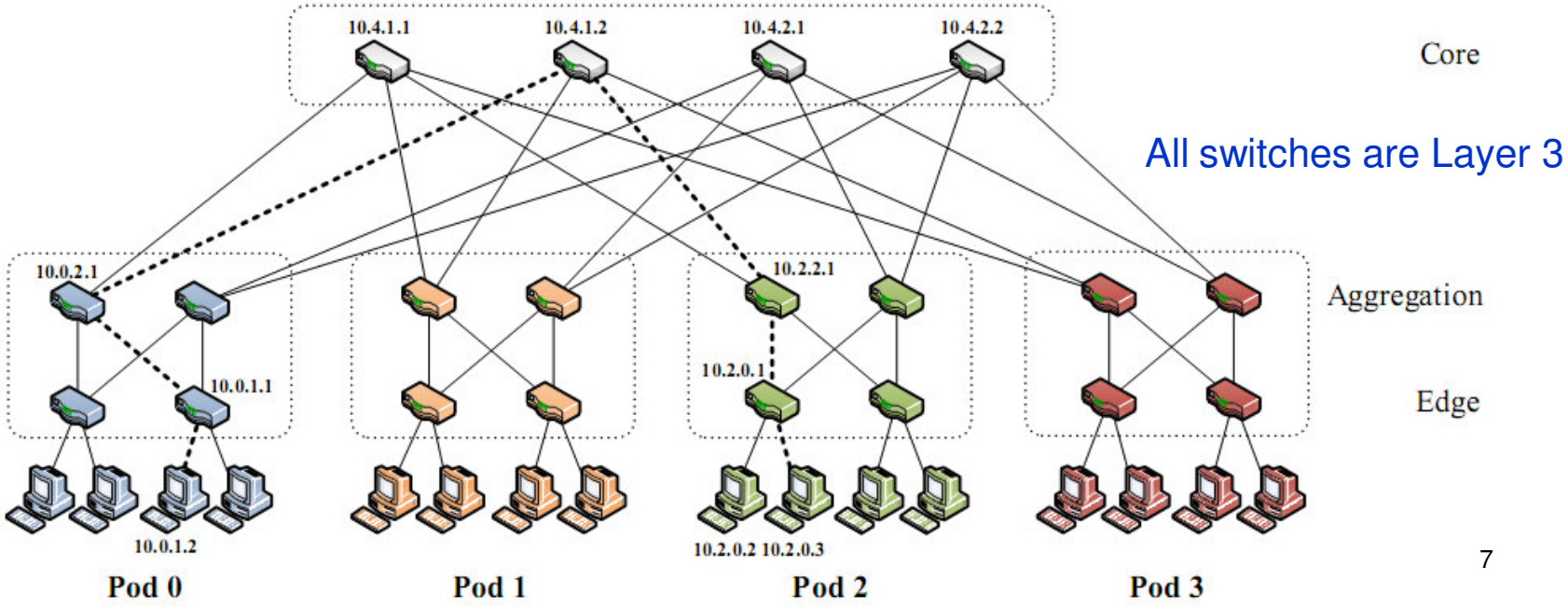
Building communication fabric



- Commodity hardware
  - Lower cost
  - No changes to TCP/IP applications
  - Poor bandwidth scalability
- Specialized hardware + protocols
  - High-end switches/routers as you go up the hierarchy
    - High cost per port
  - Changes to applications
  - Oversubscription to reduce cost
    - Lower bisection bandwidth

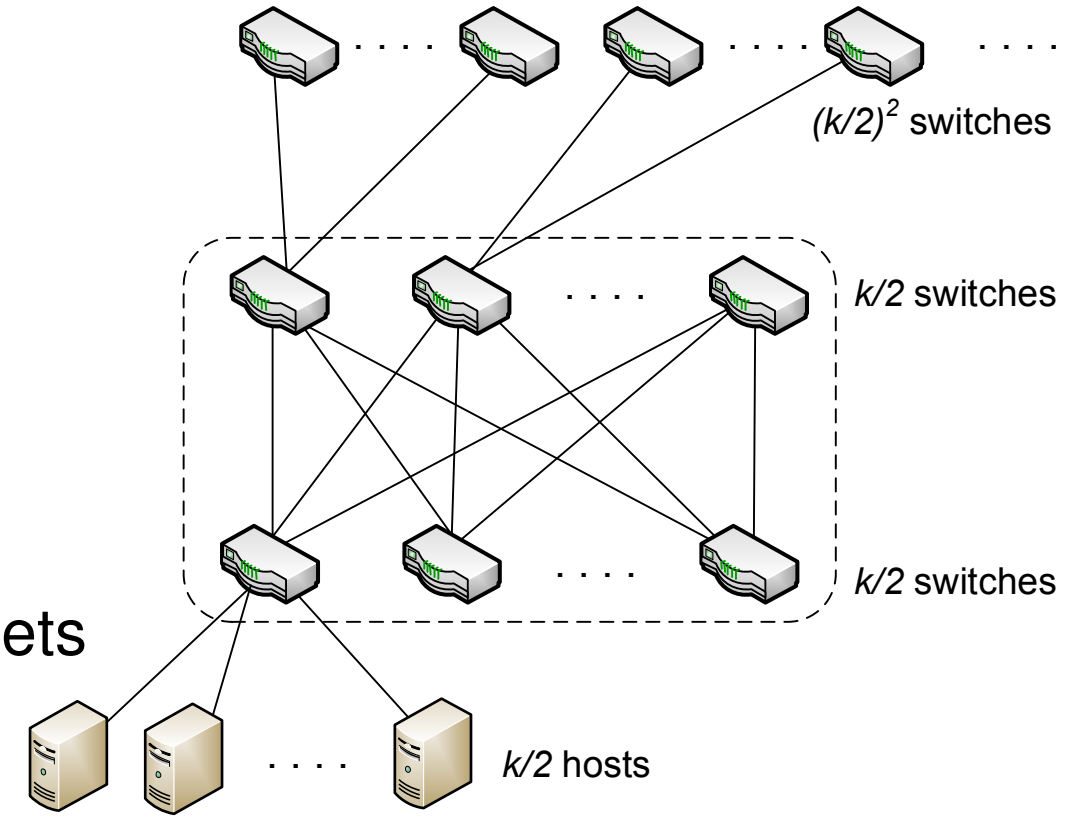
Fat tree

- Goals
  1. Lower the cost by utilizing commodity hardware
  2. Scalable interconnection bandwidth
  3. No changes to applications
- Build a fat tree (Leiserson, 1985)



Fat tree (cont.)

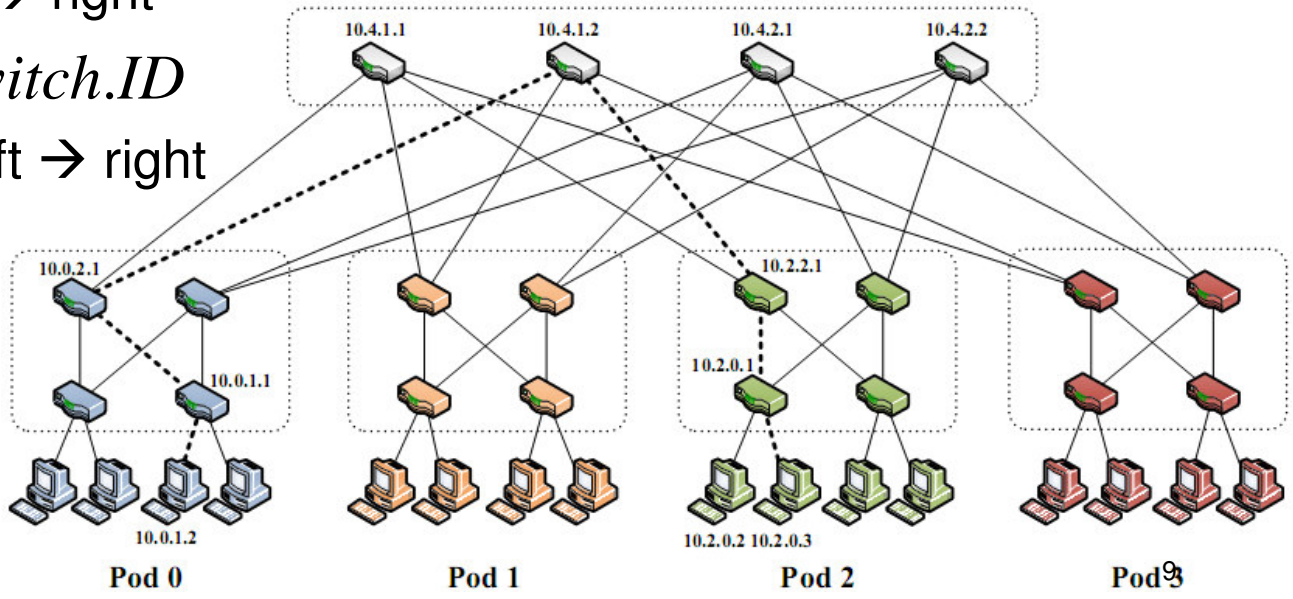
- All switches are k-port
- k pods
- $k^3/4$ hosts
- $(k/2)^2$ core switches
- k^2 switches in all pods
- $1.25k^2$ total switches
- Servers are in $k^2/2$ subnets
- $3k^3/4$ links
- $(k/2)^2$ equal cost paths



How can we make use of multiple paths?

Addressing

- Assume $10.0.0.0/8$ IP block
- Pod switches – $10.pod.switch.1$
  - $pod \in [0, k-1]$
  - $switch \in [0, k-1]$, left → right, bottom → top
- Core switches – $10.k.j.i$
  - $j, i \in [1, k/2]$, left → right
- Hosts – $10.pod.switch.ID$
  - $ID \in [2, k/2+1]$, left → right



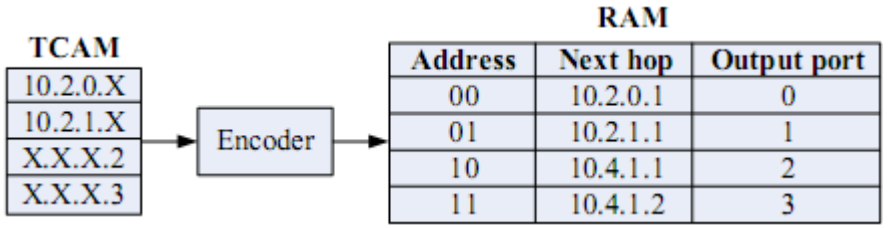
Routing table

- 2 level routing table
  - Prefix table – Reflect subnet
  - Suffix table - Reflect host *ID*
    - Allow traffic to be evenly spread & follow the same path
- $k/2 + k/2$ table entries
- Could implement as a TCAM
  - TCAM - Ternary Content Addressable Memory
  - Can store don't care terms

Prefix	Output port
10.2.0.0/24	0
10.2.1.0/24	1
0.0.0.0/0	

Suffix	Output port
0.0.0.2/8	2
0.0.0.3/8	3

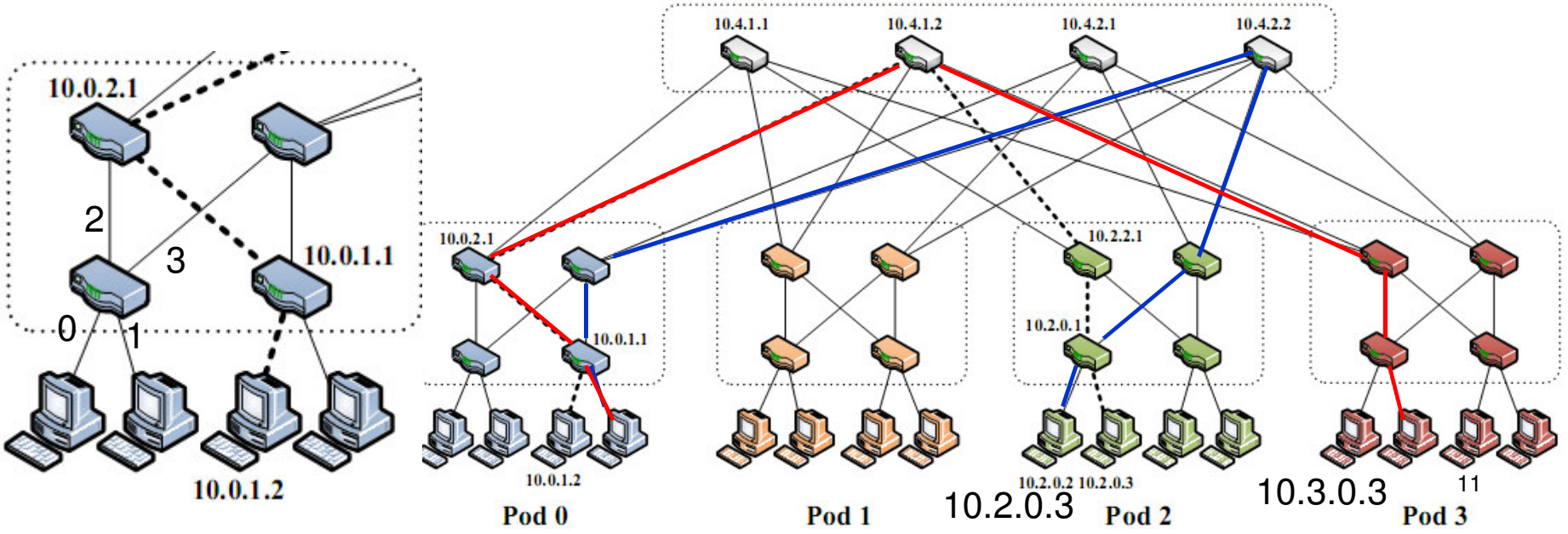
At switch 10.2.2.1



Implementation

Routing table (cont.)

- Central entity assigns routing table for each switch
- Routing table in 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 $(ID - 2 + switch) \bmod (k/2) + k/2$



Dynamic routing

- It's possible that 2 flows still use the same path

1. Flow classification – Local knowledge
   - Dynamic port reassignment of flows by switches
   - Periodically reassign flows in few of the ports
2. Flow scheduling – Global knowledge
   - Switches assign each flow to least loaded port
   - If flow exceeds a certain threshold, notify central scheduler
   - Central scheduler tracks all flows & try to reassign them into non-conflicting paths
     - Pick a non-congested path from a core switch
   - Inform selected switches (lower & upper) in source pod

Performance analysis - Implementation

- TCAM based routing table implementation on NetFPGA
  - NetFPGA – IPv4 router implementation with TCAM
- Large-scale evaluations
  - Prototype using Click
    - Modular software router architecture
  - *TwoLevelTable*, *FlowClassifier*, *FlowReporter*, *FlowScheduler*
  - Build a 4-port fat tree
    - Simulated 20 switches & 16 hosts using 10 machines
    - Used 48-port 1 Gbps switch to interconnect 10 machines
  - Each host generates traffic at 96 Mbps
    - To make the comparison with hierarchical tree easy

Performance analysis – Bandwidth

Test	Tree	Two-Level Table	Flow Classification	Flow Scheduling
Random	53.4%	75.0%	76.3%	93.5%
Stride (1)	100.0%	100.0%	100.0%	100.0%
Stride (2)	78.1%	100.0%	100.0%	99.5%
Stride (4)	27.9%	100.0%	100.0%	100.0%
Stride (8)	28.0%	100.0%	100.0%	99.9%
Staggered Prob (1.0, 0.0)	100.0%	100.0%	100.0%	100.0%
Staggered Prob (0.5, 0.3)	83.6%	82.0%	86.2%	93.4%
Staggered Prob (0.2, 0.3)	64.9%	75.6%	80.2%	88.5%
Worst cases:				
Inter-pod Incoming	28.0%	50.6%	75.1%	99.9%
Same-ID Outgoing	27.8%	38.5%	75.4%	87.4%

As a % of bisection bandwidth (1.536 Gbps) of fat tree

* $Strid_x\,(i) = (x + i) \bmod 16$ * $Staggered\,Prob(p_{sub}, p_{pod})$

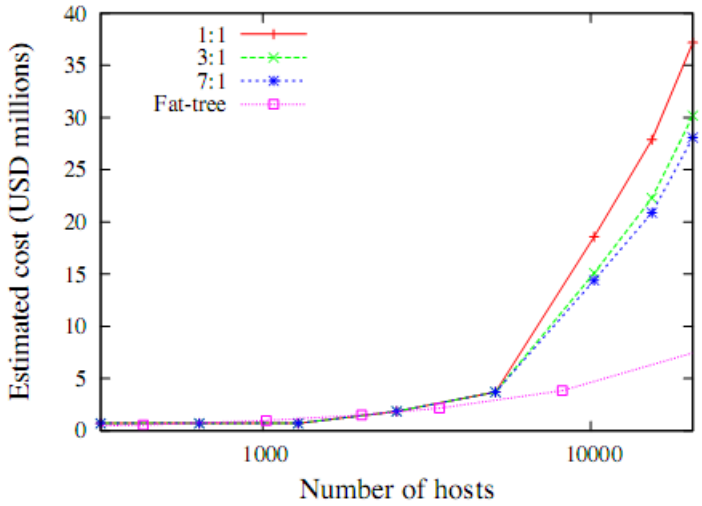
* *Inter-pod Incoming* – Multiple pods send to different hosts in same pod, same core switch

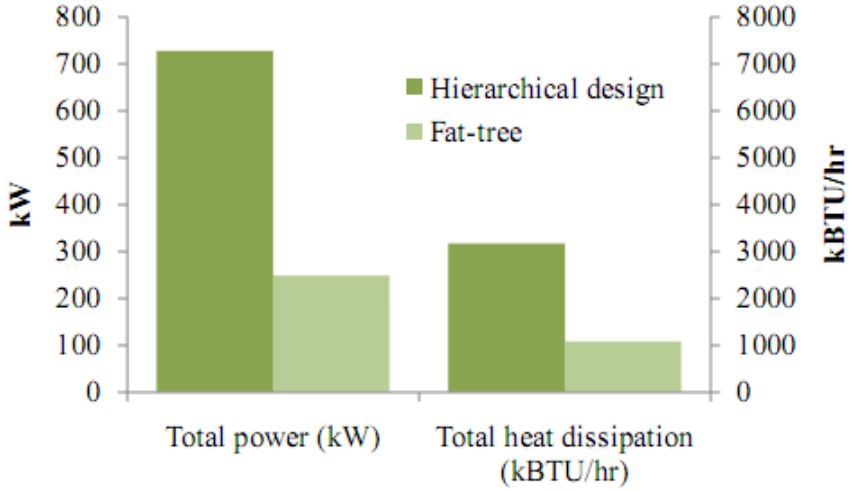
* *Same-ID Outgoing* – Destination host have the same ID, congestion at aggregation layer

- Inter-pod – flow scheduler is best, hierarchical tree is worst
- 2 level table use same outgoing path if destination *ID* is same
- Lack of global knowledge hinder performance of flow classification
- With global knowledge flow scheduling is the best

Performance analysis (cont.)

k	Hosts	Avg Time/ Req (μs)	Link-state Memory	Flow-state Memory
4	16	50.9	64 B	4 KB
16	1,024	55.3	4 KB	205 KB
24	3,456	116.8	14 KB	691 KB
32	8,192	237.6	33 KB	1.64 MB
48	27,648	754.43	111 KB	5.53 MB



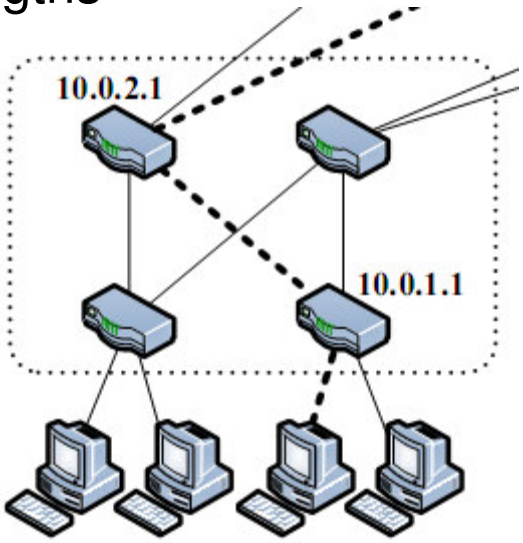


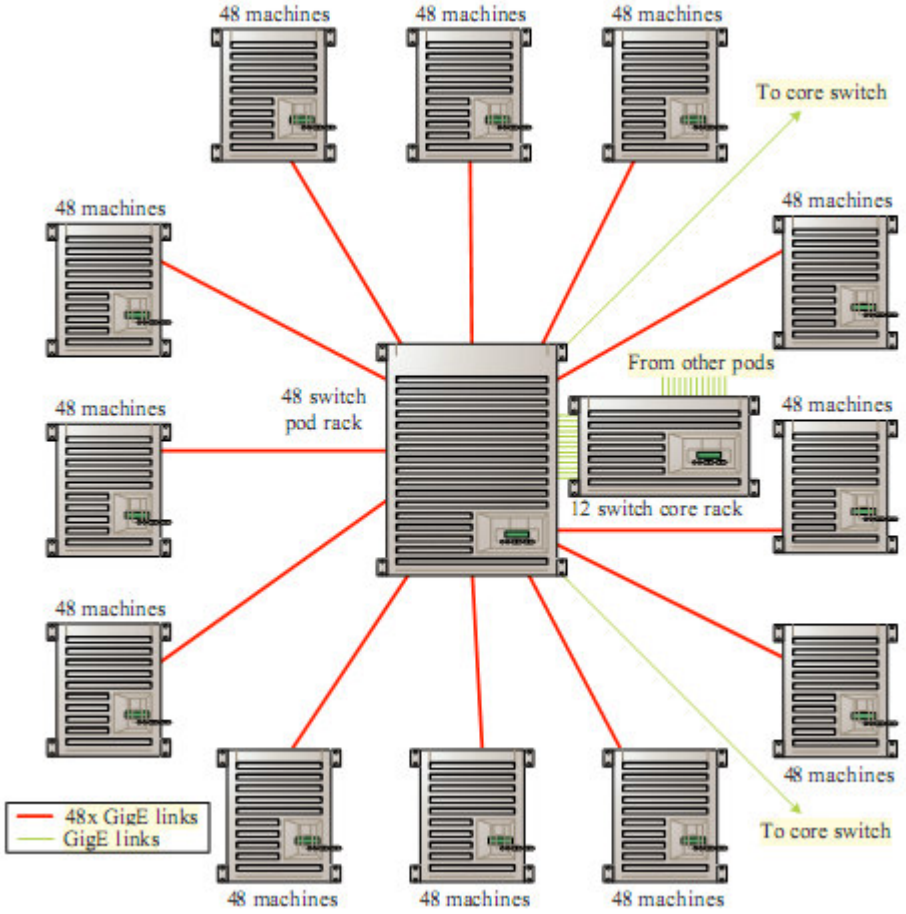
48-port 1 Gbps at edge @ $7000
128-port 10 Gbps at aggregate & edge @ $700,000

- Overhead of central scheduler is low, relatively low resources
- Cost of conventional hierarchical design is prohibitive

Packaging

- Design for $k = 48$
  - 48 switch rack
  - 576 hosts in 12 racks
  - 576 core switches
    - 12 switches in a pod
  - 2D design to reduce cable lengths





576 wires going out to hosts

Critique

- Pros
  - Well written
  - Maintain full bandwidth for most communication patterns
  - Relatively low cost
  - No changes to TCP/IP applications
- Cons
  - Need modification to routers
  - Need a flow scheduler
  - Excessive wiring – even with specially designed racks
  - Configuration of many switches (Mysore, 2009)
  - VM migration needs reassignment of IP address (Mysore, 2009)
  - VLANs could restrict redundant links
- Authors are working on a layer 2 solution (Mysore, 2009)

Building Low-diameter Peer-to-Peer Networks

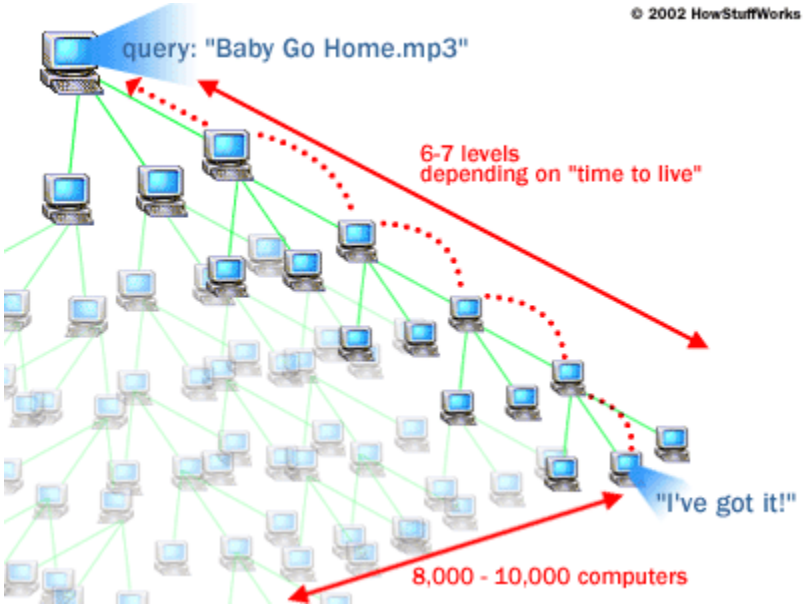
Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal

IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, Aug. 2003, pp. 995-1002.

Peer-to-peer networks

- A distributed system without any central control
- Typically peers are identical in functionality
- Tit-for-tat strategy
- Tremendous scalability
  - Millions of peers around the world
- Many application domains
  - File sharing, IPTV, VoIP, CPU cycle sharing
- Heterogeneous peers
- Peer churn & failure

Motivation – Gnutella protocol



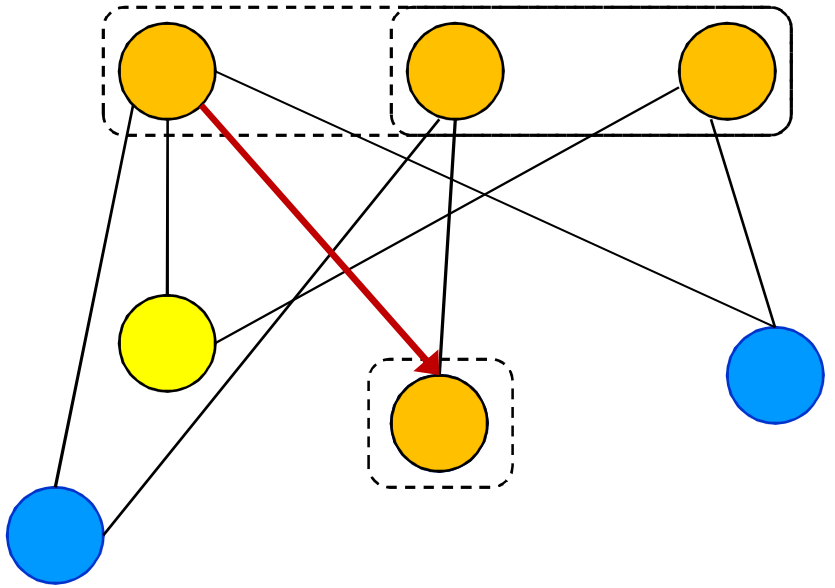
- Content discovery is based on broadcasts
- Random walk with TTL
  - Content discovery is not guaranteed
- Diameter of the graph could grow arbitrarily
  - Specification is open on how to & how many neighbors to maintain

Solution

- Goals
  - Construct the P2P topology such that
    - It is connected
    - Bounded node degree
    - Logarithmic diameter
- Approach
  - A centralized node to enable new & lost connections
  - Distributed algorithm to maintain node connectivity
  - Define set of parameters that achieve above goals

P2P protocol

- Central host server
  - Maintains K cache nodes
- New node
  - Connects to D cache nodes
  - Called d-node
- Cache node
  - Go out of cache after getting C connections
  - Called a c-node
  - Add a d-node neighbor to cache
  - Maintains a preferred connection
- Node degree $\in [D, C+1]$



$K = 3, D = 2, C = 4$

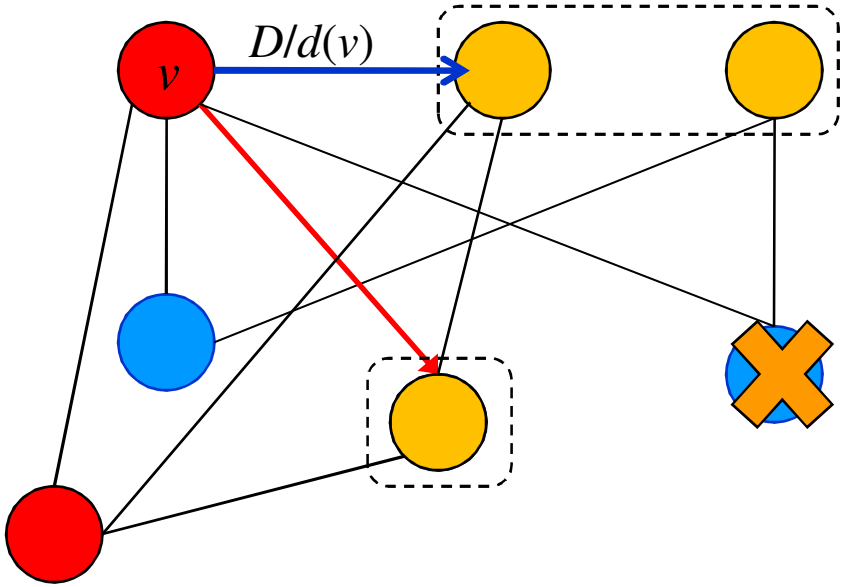
Cache node New node

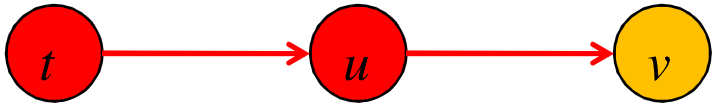
d-node c-node

Preferred connection

P2P protocol (cont.)

- If a connection is lost
  - Reconnect to a cache node
  - With probability $D/d(v)$
  - d-nodes will always reconnect
- If preferred connection is lost
  - Reconnect to a cache node
- Cache replacement rule
  - v is a cache node that is ready to go out of cache
  - v may not have any d-node neighbors
  - Then check its predecessor cache nodes





Assumptions

- Arrival & departure processes
  - New node arrival ~ Poisson(λ)
  - Time that a node stay in system ~ Exponential(μ)
  - Assume most peers are using dial-up links
  - Similar behavior conformed by measurements (Saroiu, 2002) & (Guillemin, 2008) for protocols without sharing incentives
- Size of the network $N = \lambda/\mu$
- Assume $\lambda = 1$
- Let the network at time t be $G_t = (V_t, E_t)$

Key theorems & lemmas

- Theorem III.1 – Size of the network $N - o(N)$
- Lemma III.1 – Bounded node degree
  - A replacement d-node can be found w.h.p when $C > 3D + 1$
- Theorem III.3 – G_t is connected w.h.p
  - Lemma III.4 – Each node is connected to a cache node
  - Lemma III.5 – 2 cache nodes are connected w.h.p
- Corollary III.1 – Network rapidly recovers from disconnections
- Theorem III.5 – Diameter of the network is $O(\log N)$
- Lemma IV.1 – There is a constant probability of some nodes are isolated

Bounded node degree

- A leaving cache node needs to find a replacement d-node
  - Else network can't accept new connections
  - Or has to break node degree bound $C + 1$
- A replacement node needs to be found with high probability
- Lemma III.1
  - Let $C > 3D + 1$, then at any time $t \geq a \log N$ $(a > 0)$ no of d-nodes in network (with high probability) is

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

- Consider interval $[t - N, t]$

Bounded node degree (cont.)

- Number of new connections = $DN(1 + o(1))$
- E[Number of reconnections in unit time] =

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

- Number of cache nodes need in $[t - N, t]$

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

  - All these nodes will become *c*-nodes
- $N + o(N)$ nodes in network at any time (Theorem III.1)

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

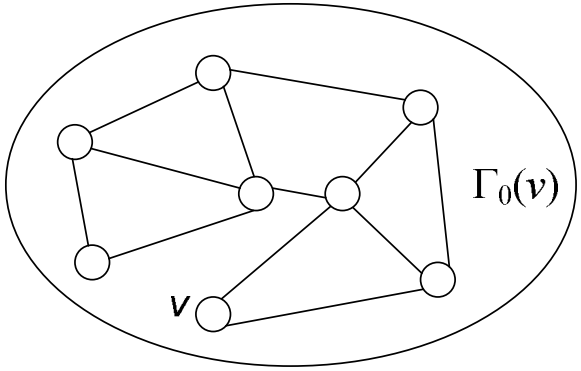
- To satisfy our requirement $C > 3D+1$

Diameter

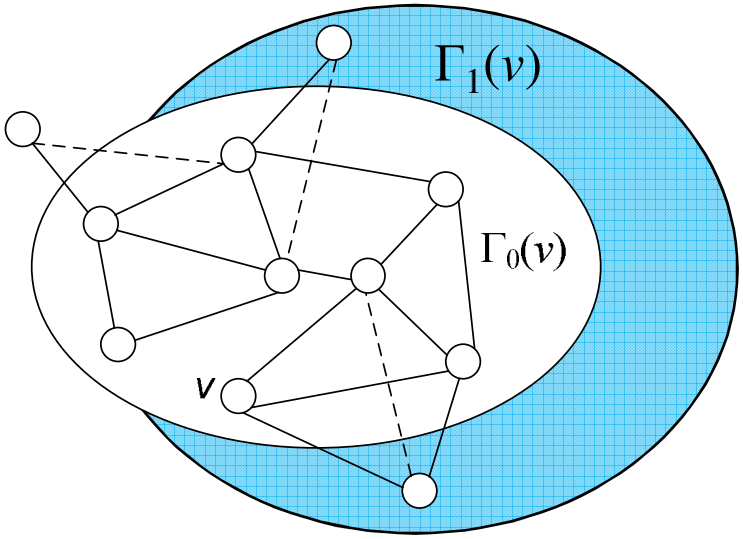
- A d-node is always connected to a c-node
  - It's sufficient to consider connectivity of c-nodes
- Let f be a constant
- A cache node is called good, if it receives $r \geq f$ connections
  - All r connections are reconnection requests
  - All r connections are not preferred connections
  - r connections result for departure of r different nodes
- 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 with B_1
  - Color another $f/2$ of reconnection links of a good cache node with B_2
  - Color all other links with A

Diameter (cont.)

- A connections could grow arbitrary long
- Reconnections (B_1, B_2) can reduce distance to a cache node

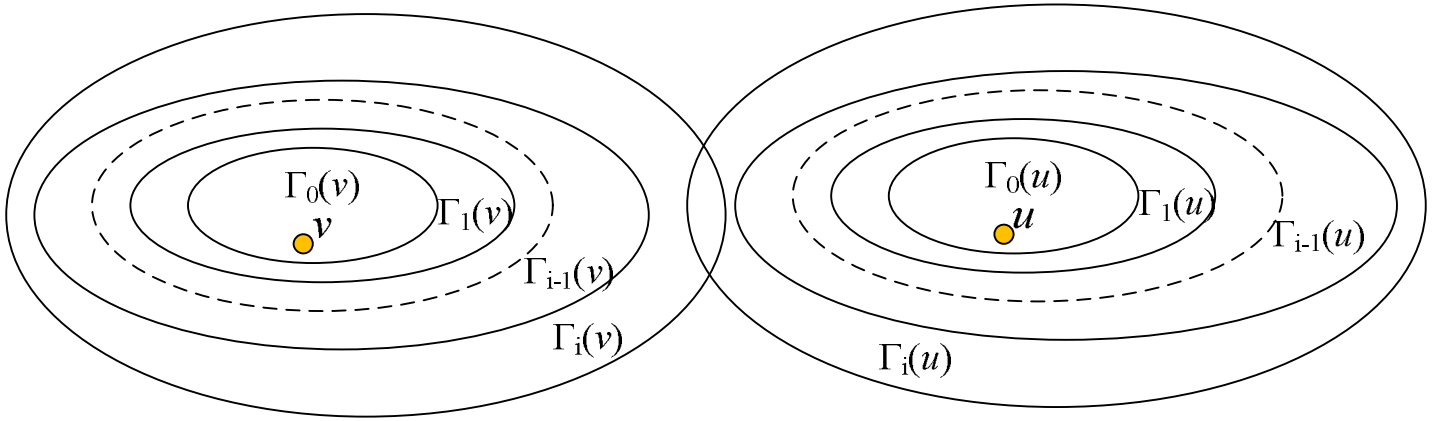


Diameter = 3



- Let $\Gamma_0(v)$ be an arbitrary cluster of $d \log N$ c-nodes, $v \in \Gamma_0(v)$
- Cluster has a diameter of $O(\log N)$ using only A edges
- Cluster expands by connecting nodes through B_1 connections
- From Lemma III.7 $Pr\{|\Gamma_i(v)| \geq 2|\Gamma_{i-1}(v)|\} \geq 1 - \frac{1}{N^5}$

Diameter (cont.)

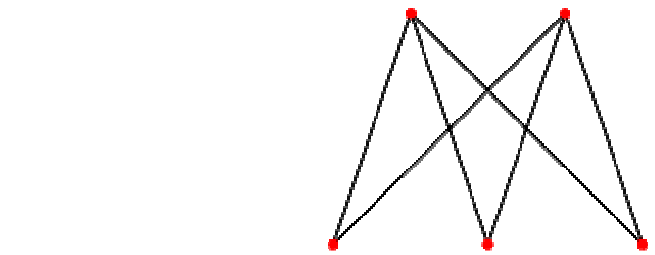


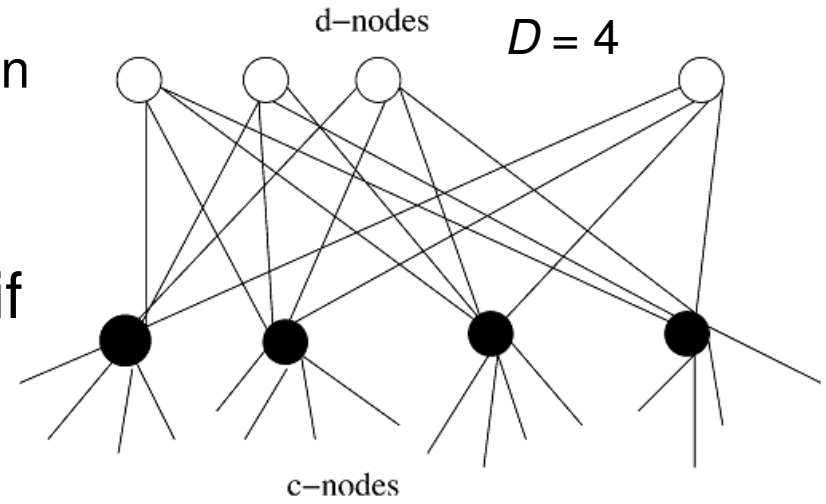
- Let $\Gamma_0(v)$ & $\Gamma_0(u)$ be clusters formed by any 2 c-nodes u & v
- Goal is to show that distance between 2 c-nodes is $O(\log n)$
  - Applying Lemma III.7 $O(\log N)$ times, i.e., $c \log N$ times
  - $= |\Gamma_{c \log N}(v)| \geq 2^{c \log N} |\Gamma_0(v)| = 2^{c \log N}\, d \log N \approx N^{1/2} \log N$
  - $P\{$ All nodes in $\Gamma_{c \log N}(v)$ & $\Gamma_{c \log N}(u)$ are disconnected using B_1 links$\}$

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

Why preferred connections

- There is a constant probability that the network may form a complete bipartite network (F)
- F will be isolated if
  - All its $2D$ nodes stay in system by t
  - All c-nodes loose neighbors other than new d-nodes
  - c-nodes don't try to reconnect
- Reconnection is guaranteed only if $d(v) = D$
  - Because $D/d(v)$
  - To ensure bounded node degree no reconnection every time a link is lost

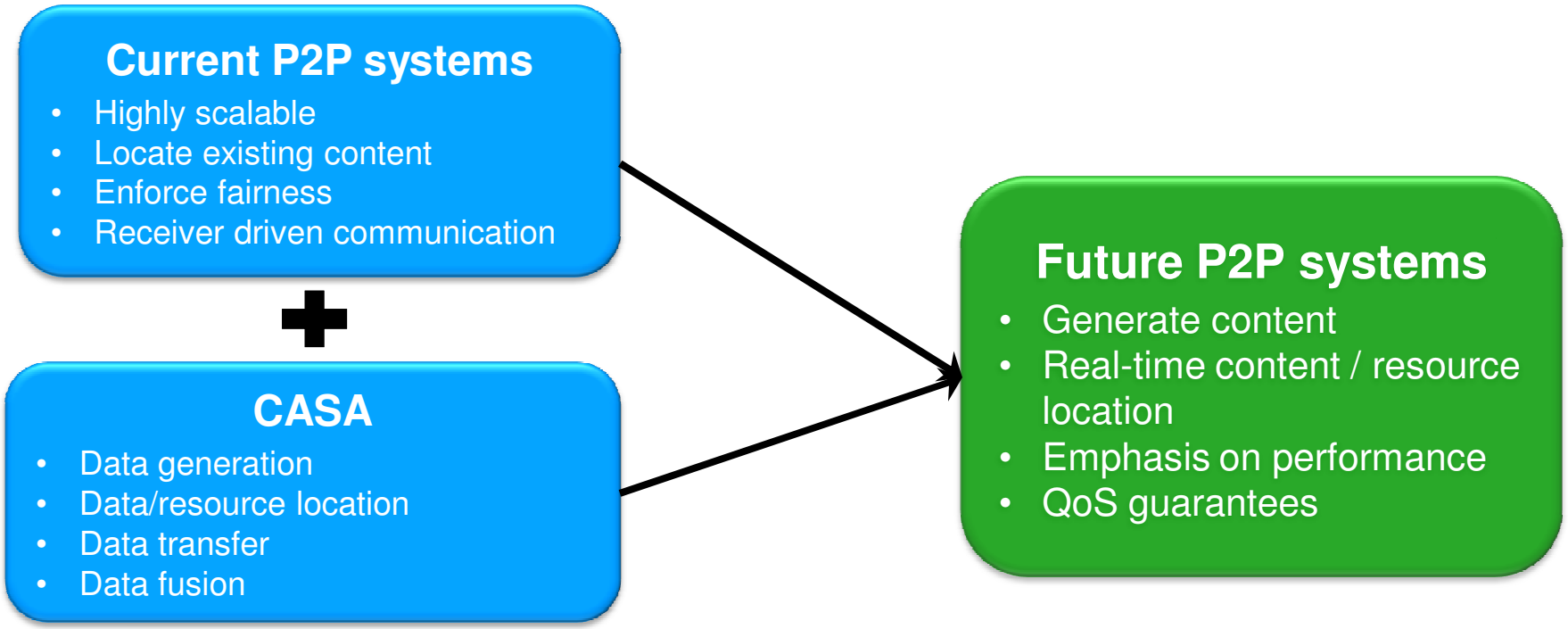




Critique

- Pros
  - Provable performance
  - Bounded node degree
  - Logarithmic network diameter, $TTL = O(\log N)$
- Cons
  - Host server could result in a single point of failure
  - No simulation/experimental results to confirm analysis
  - A c-node will have lower than $C + 1$ connections after a while
    - Node degree $[D, C + 1]$
    - Peers that stays in network for a long time are good candidates to have high node degree
      - Super peers
    - They could learn about the resources in the network

P2P in Collaborative Adaptive Sensing of the Atmosphere (CASA)



- Papers focus on scalable topology construction & maintenance without high bandwidth links
- Multiple paths to a destination – bound their distance
- Lower diameter & bounded node degree is important
- P2P is an alternative for some data center applications – e.g., BOINC, MOINC

References

1. F. Guillemin, C. Rosenberg, L. Le, and G. V. Brugier, “Peer-to-Peer Traffic: From Measurements to Analysis,” In Proc. Of IEEE Global Telecommunications Conference 2008 (GLOBECOM 2008), Dec. 2008, pp. 1-5.
2. C. E. Leiserson, “Fat-trees: universal networks for hardware-efficient supercomputing,” IEEE Transactions on Computers, vol. 34, no. 10, 1985, pp. 892–901.
3. R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, “PortLand: a scalable fault-tolerant layer 2 data center network fabric,” In Proc. of ACM SIGCOMM ‘09, Barcelona, Spain, Aug. 2009.
4. S. Nichols, “Streaming soars, but P2P traffic drops,” PC World, Sep. 03, 2008, Available: http://blogs.pcworld.com/staffblog/archives/007652.html
5. G. C. Pandurangan, “Stochastic analyses of dynamic computer processes,” PhD Dissertation, Brown University Providence, RI, USA, 2002.
6. S. Saroiu, P.K. Gummadi, and S.D. Gribble, “A measurement study of peer-to-peer file sharing systems,” in Proc. Multimedia Computing Networking (MMCN), San Jose, CA, 2002.
7. D. Stutzbach, R. Rejaie, and S. Sen, “Characterizing unstructured overlay topologies in modern p2p file-sharing systems,” in In Proc. of Internet Measurement Conference (IMC), 2005

Questions ?

Thank You…