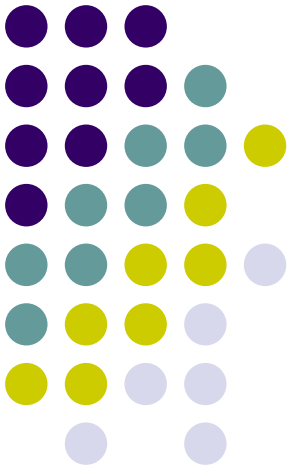
P2P Overlay Design Overview

John Buford, Panasonic Digital Networking Laboratory
IRTF P2P RG Core Subgroup
buford@research.panasonic.com

Keith Ross, Polytechnic University
ross@poly.edu



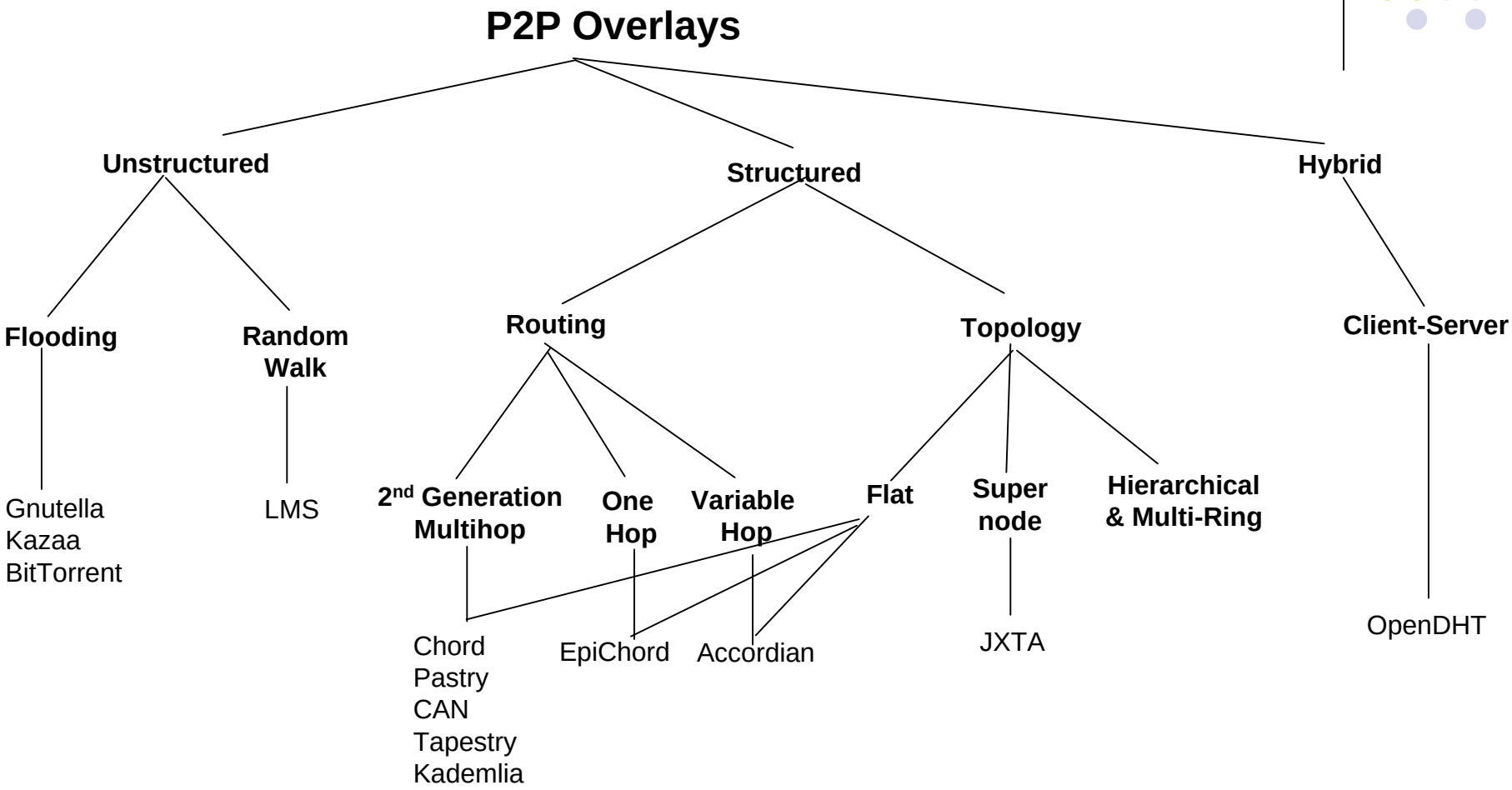
November 2005

Definition

- Overlay network
  - An overlay network is a virtual network of nodes and logical links that is built on top of an existing network with the purpose to implement a network service that is not available in the existing network
    - *I. Stoica*

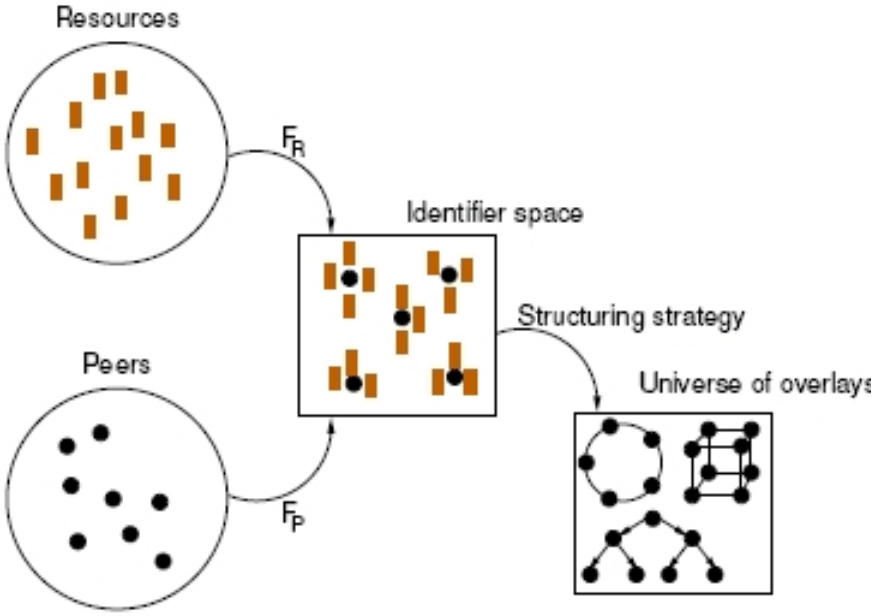
- Peer-to-peer
  - A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P,.) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers,.). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers
    - *Rüdiger Schollmeier: A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. Peer-to-Peer Computing 2001*

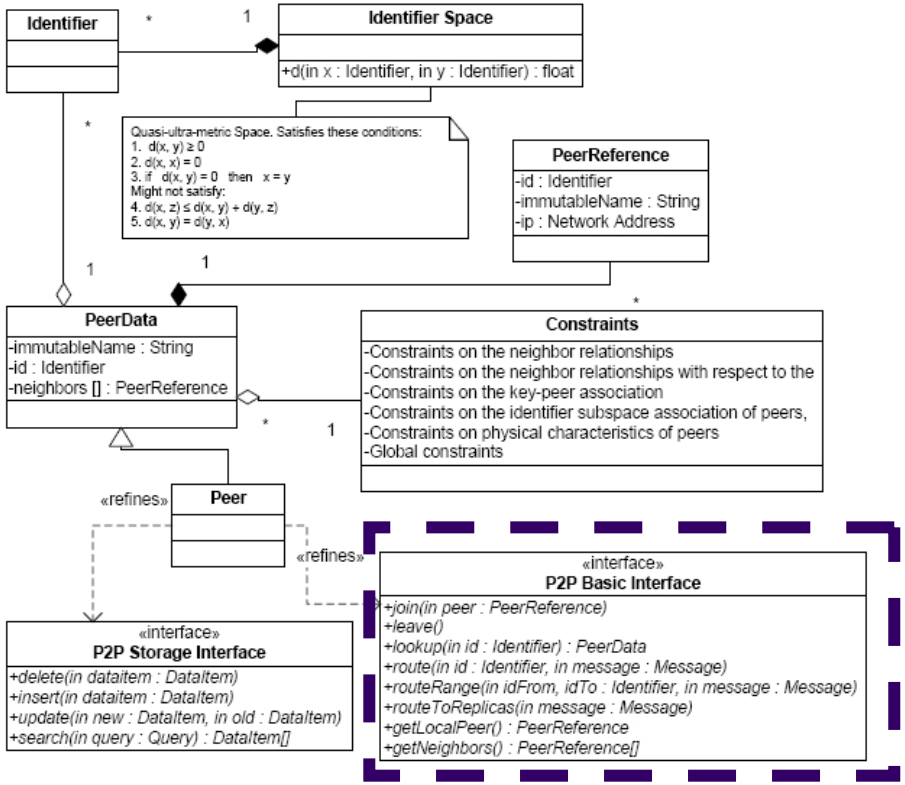
Taxonomy


P2P Overlays
Unstructured
Structured
Hybrid
Flooding
Random Walk
Routing
Topology
Client-Server
Gnutella
Kazaa
BitTorrent
LMS
2nd Generation Multihop
One Hop
Variable Hop
Flat
Super node
Hierarchical & Multi-Ring
OpenDHT
Chord
Pastry
CAN
Tapestry
Kademlia
EpiChord
Accordian
JXTA

Overlay Design Space

1. **Choice of an identifier space**
2. **Mapping of resources and peers to the identifier space**
3. **Management of the identifier space by the peers**
4. **Graph embedding (structure of the logical network)**
5. **Routing strategy**
6. **Maintenance strategy**





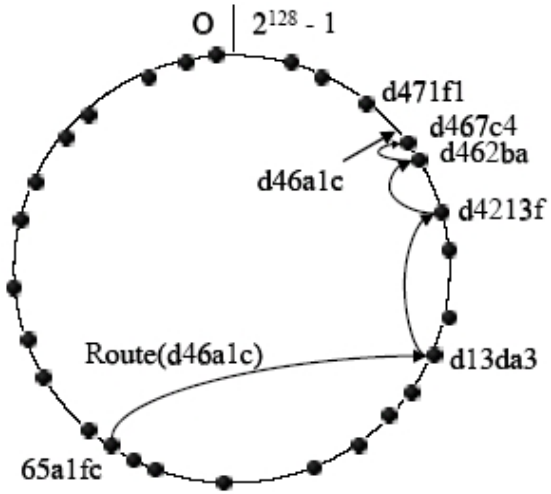
K. Aberer, et al. The essence of P2P: a reference architecture for overlay networks. Fifth IEEE International Conference on Peer-to-Peer Computing, Sep 2005, Konstanz, Germany.

Examples

- Structured Overlays
  - Multi-Hop
    - Chord
  - One-Hop
    - Epichord
  - Variable Hop
    - Accordian
  - Client-Server
    - OpenDHT
- Hierarchical / Multi-Ring
- Unstructured Overlays
  - LMS

2nd Generation Multihop Structured Overlay

	Pastry	CAN	Chord	Tapestry
Source	**Microsoft Research**	**ICSI**	**MIT**	**UC Berkeley**
Overlay network	Yes	Yes	Yes	Yes
DHT	Yes	Yes	Yes	Yes
Flat / Layered	Flat	Flat	Flat	Flat
Routing	Prefix based Multihop	Cartesian routing in N-dimensional space	Finger table	Longest-prefix Multihop
Routing performance	O(log N)	O(log N)	O(log N)	O(log N)
Routing table size	O(log N)	O(log N)	O(log N)	O(log N)



Chord

Scope

Initializing routing table of new node when joining overlay

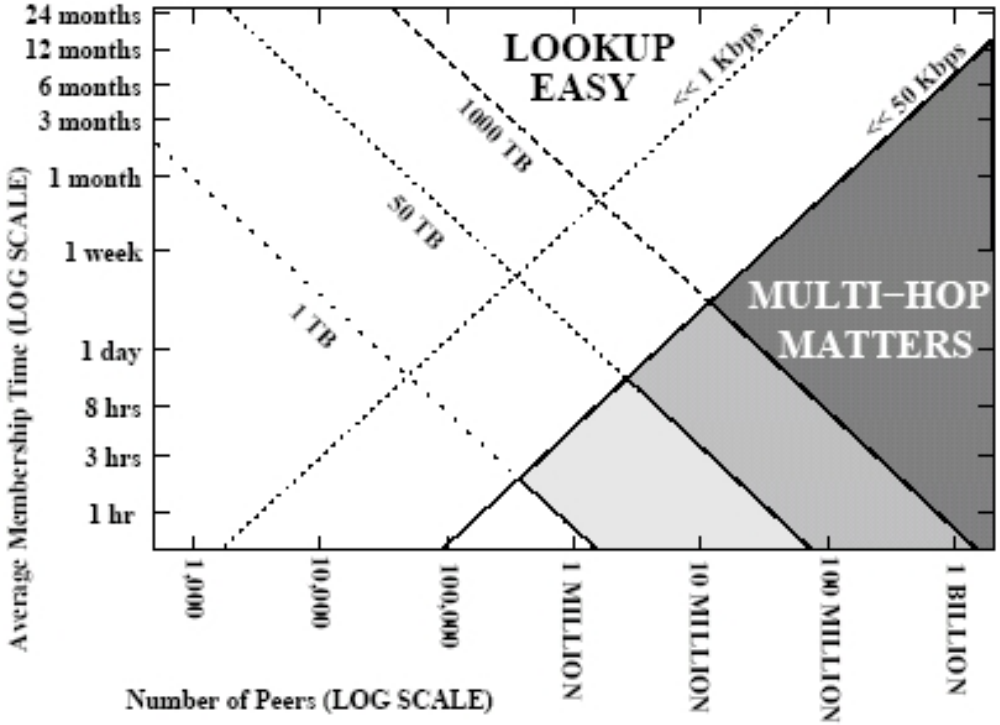
Updating routing table of other nodes in routing table when node joins or leaves

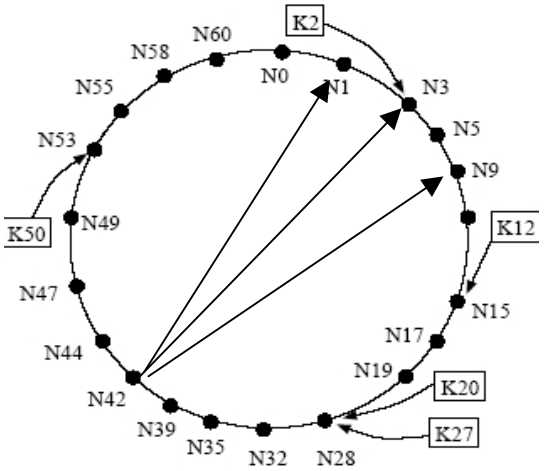
Re-arranging entries in the index to different nodes when nodes join or leave

Replicating index entries at multiple nodes

- A. Rowstron, P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. IFIP/ACM Intl Conference on Distributed Systems. Nov. 2001.
- S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker. A Scalable Content-Addressable Network. Proc of ACM SIGCOMM, ACM 2001
- I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan. Chord:A scalable peer-to-peer lookup service for internet applications. Proc. Of the ACM SIGCOMM 01 Conf., San Diego, Aug 2001
- B Y Zhao, J D Kubiatowicz, A. D Joseph. Tapestry: An infrastructure for fault-resilient wide-area location and routing. TR UCB/CSD-01-1141. UC Berkeley, April 2001.

Multi-Hop vs One-Hop





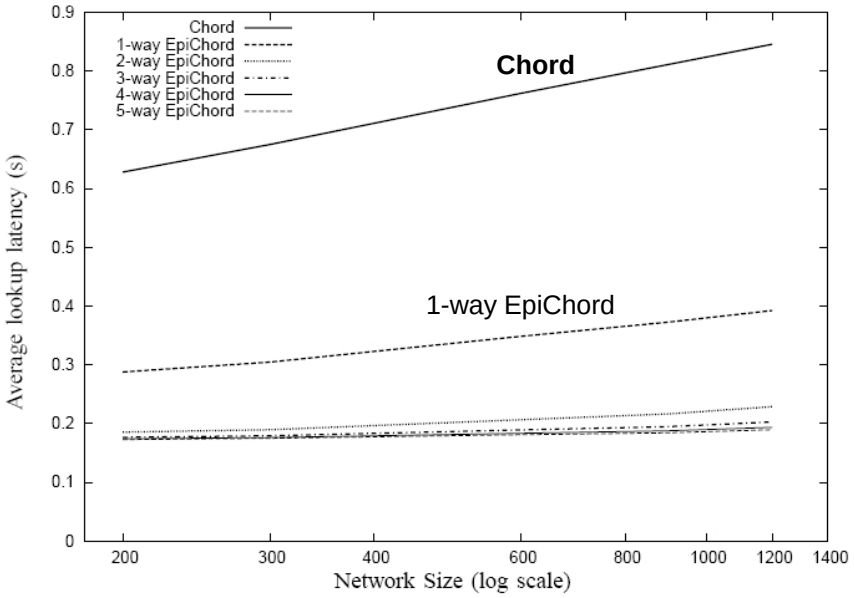
Anjali Gupta, Barbara Liskov, and Rodrigo Rodrigues. One Hop Lookups for peer-to-peer overlays Proceedings of the 9th Workshop on Hot Topics in Operating Systems (HotOS-IX), Lihue, Hawaii, May 2003.

Rodrigo Rodrigues and Charles Blake.When Multi-Hop Peer-to-Peer Routing Matters. Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS04), San Diego, CA, February 2004.

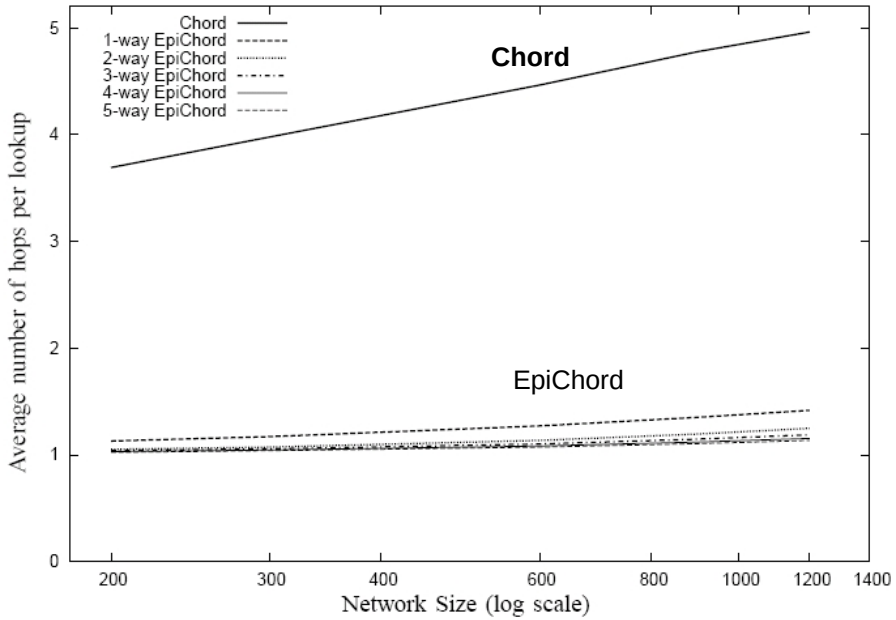
EpiChord – One Hop Overlay

- *Reactive* routing state management strategy where routing state maintenance costs are amortized into the lookup costs.
- Nodes piggyback network information on query replies to keep their routing state up-to-date under reasonable traffic conditions.
  - Only sends probes as a backup mechanism if lookup traffic levels are too low to support the desired level of performance.
- Can issue parallel queries without generating excessive amounts of lookup traffic only because its large routing state reduces the number of hops per lookup and thereby the number of lookup messages.
- EpiChord divides its routing table into slices, and maintains *j* entries per slice. Key performance results of EpiChord include:
  - 1. For j = 1, EpiChord gets the same worst-case lookup performance as Chord
  - 2. For j = 2, EpiChord lookup path lengths are 1/3 of Chord's
  - 3. For network sizes of n>1M nodes, at least 25% of the background traffic for maintaining routing information is eliminated due to piggybacking on lookups

EpiChord vs Chord



p-way EpiChord lookup latency vs Chord under lookup intensive workload

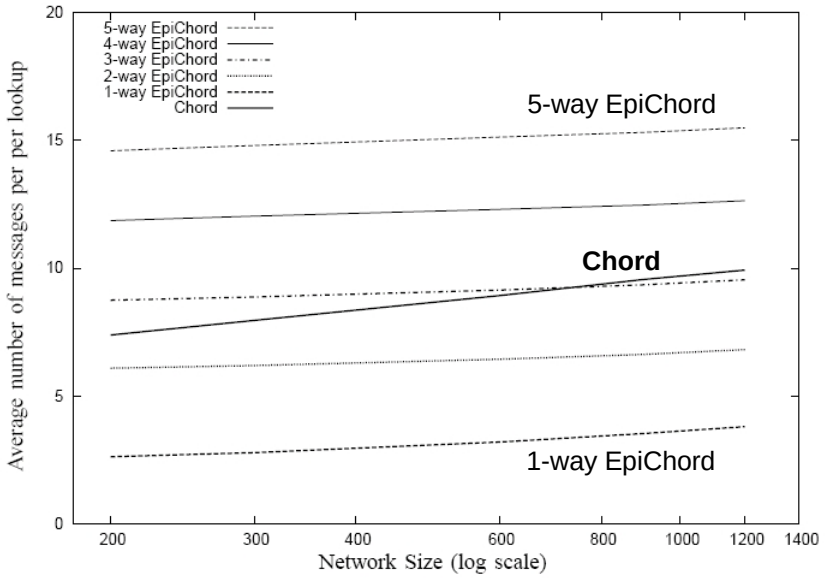


Average number of hops per lookup, comparing *p*-way EpiChord and Chord under lookup intensive workload

Ben Leong, Barbara Liskov, and Erik D. Demaine. EpiChord: Parallelizing the Chord Lookup Algorithm with Reactive Routing State Management". 12th International Conference on Networks (ICON), (Singapore), Nov. 2004.

Ben Leong, Ji Li. Achieving One-Hop DHT Lookup and Strong Stabilization by Passing Tokens. 12th International Conference on Networks (ICON), (Singapore), Nov. 2004.

EpiChord vs Chord

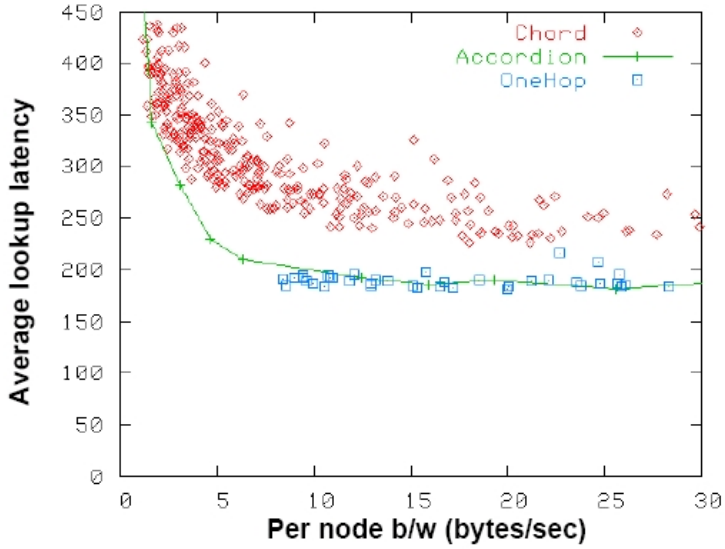


Average number of messages per lookup, *p*-way EpiChord vs Chord, in lookup intensive workload

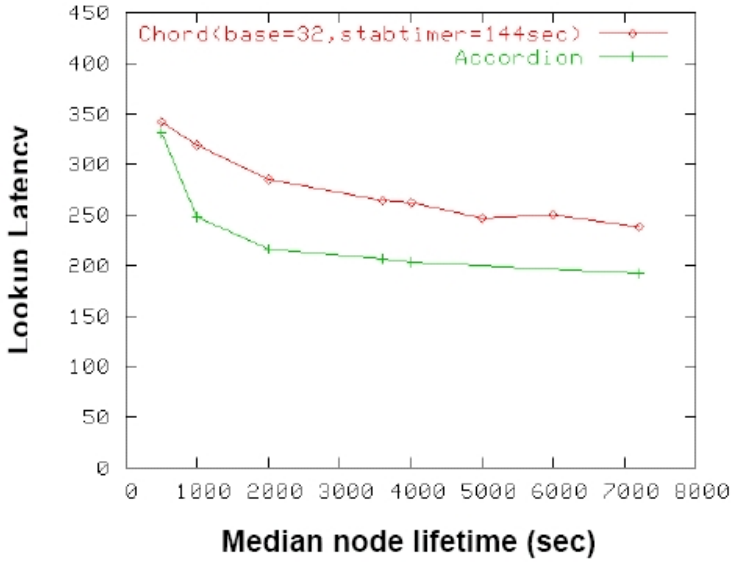
In simulations comparing *p*-way EpiChord vs Chord under lookup intensive and churn-intensive workload, EpiChord exhibits lower latency and lower number of hops per lookup at the cost of higher message traffic in the lookup intensive workload case.

Accordian – adaptive routing table size

- Goal: routing table that minimizes latency
- Use b/w budget to search for new nodes
  - To reduce average lookup hops
- Evict nodes likely to be dead
  - To reduce lookup timeouts
- Table size is determined by the equilibrium of acquisition and eviction process



1024 nodes



1024 nodes

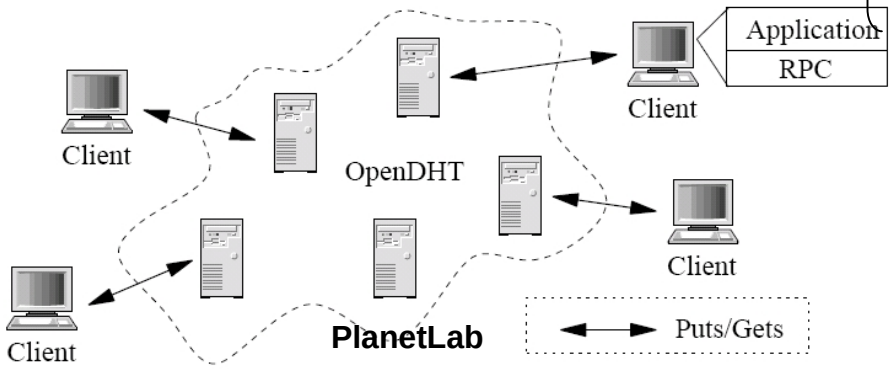
Jinyang Li, Jeremy Stribling, Robert Morris and M. Frans Kaashoek. Bandwidth-efficient management of DHT routing tables. In the Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05), Boston, MA, 2005.

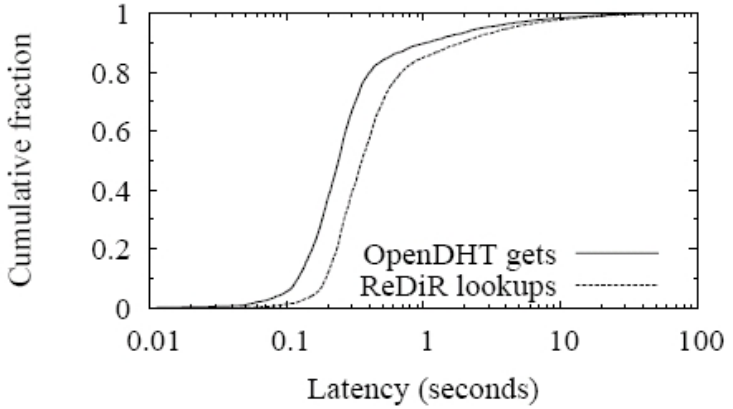
Open DHT

- OpenDHT operates on a set of infrastructure nodes, clients run on separate nodes
- No client node is concerned with DHT deployment
- Clients can not run application-specific code on infrastructure nodes.
- Based on Bamboo DHT/overlay

Procedure	Functionality
$put(k,v,H(s),t)$	Write (k,v) for TTL t can be removed with secret s
$get(k)$ returns $\{(v,H(s),t)\}$	Read all v stored under k returned value(s) unauthenticated
$remove(k,H(v),s,t)$	Remove (k,v) put with secret s $t >$ than TTL remaining for put
$put\text{-}immut(k,v,t)$	Write (k,v) for TTL t immutable $(k = H(v))$
$get\text{-}immut(k)$ returns (v,t)	Read v stored under k returned value immutable
$put\text{-}auth(k,v,n,t,K_P,\sigma)$	Write (k,v), expires at t public key K_P; private key K_S can be removed using nonce n $\sigma = \{H(k,v,n,t)\}_{K_S}$
$get\text{-}auth(k,H(K_P))$ returns $\{(v,n,t,\sigma)\}$	Read v stored under $(k,H(K_P))$ returned value authenticated
$remove\text{-}auth(k,H(v),n,t,K_P,\sigma)$	Remove (k,v) with nonce n parameters as for *put-auth*

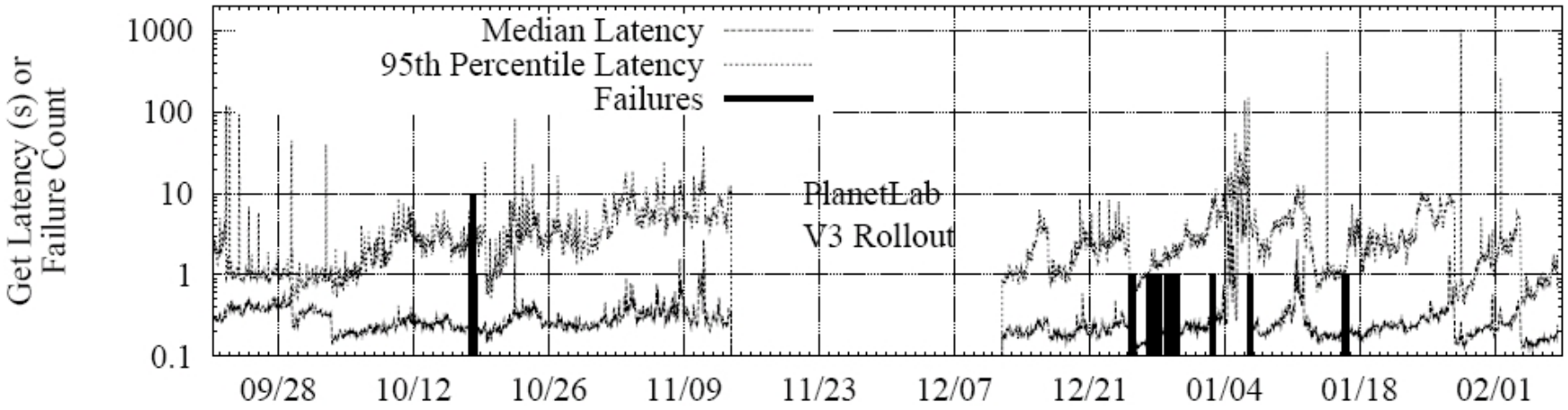
OpenDHT APIs





Rhea, S., Godfrey, B., Karp, B., Kubiatowicz, J., Ratnasamy, S., Shenker, S., Stoica, I., and Yu, H. 2005. OpenDHT: a public DHT service and its uses. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (Philadelphia, Pennsylvania, USA, August 22 - 26, 2005).

OpenDHT – 3.5 month test



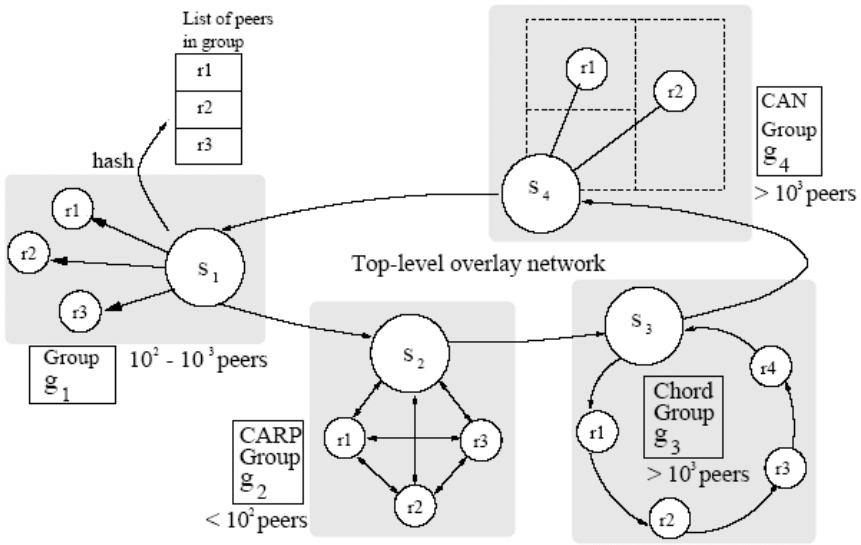
- ~ 200 PlanetLab hosts
- Client puts 1 value every second
- Value sizes are drawn randomly from {32, 64, 128, 256, 512, 1024} bytes, and TTLs are drawn randomly from {1 hour, 1 day, 1 week}.
- Client randomly retrieves a value every second
- “Latency ramps due to bugs, now fixed”
- 9M puts and gets each, only 28 lost values during test period
- See paper for other tests

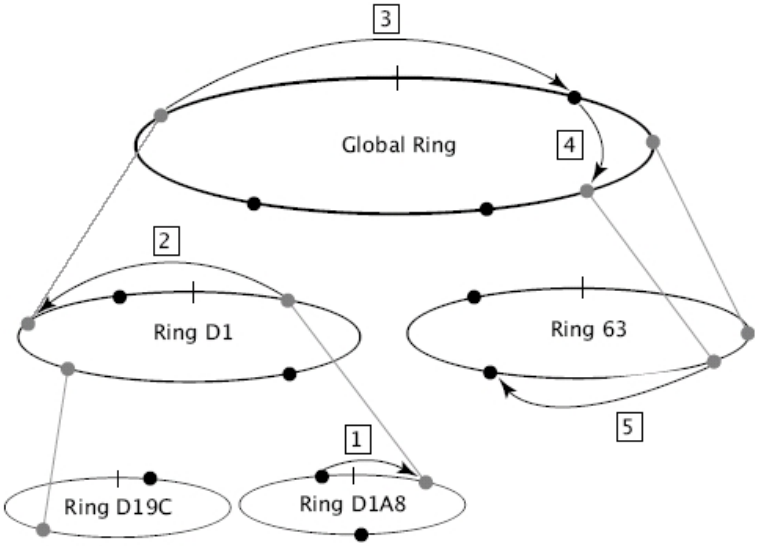
Rhea, S., Godfrey, B., Karp, B., Kubiatowicz, J., Ratnasamy, S., Shenker, S., Stoica, I., and Yu, H. 2005. OpenDHT: a public DHT service and its uses. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (Philadelphia, Pennsylvania, USA, August 22 - 26, 2005).

Hierarchical / Multi-Ring

- There are several research proposals for organizing P2P overlays in to a hierarchy. These proposals are motivated as follows:
  - Supporting sophisticated search requirements
  - Permitting different overlays to have separate administrative domains while still supporting wide area use
  - Separating categories of use (content, personal communications, …)
  - Scaling
  - Boot strapping of multiple overlay-based systems
- Generally these proposals maintain a common addressing model and are hierarchies of structured overlays

Hierarchical / Multi-Ring





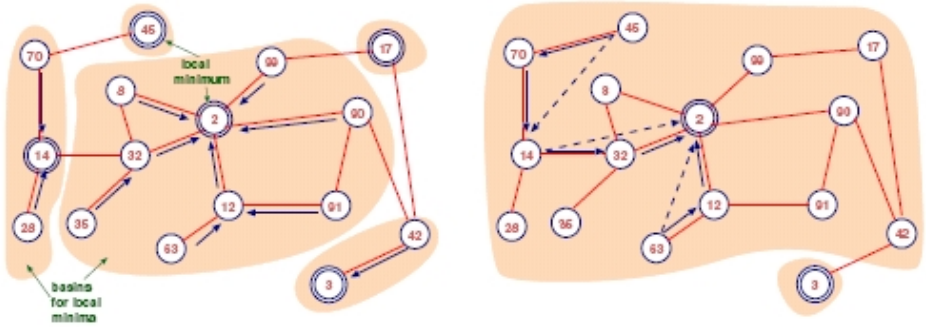
L. Garces-Erce, E. Biersack, P. Felber, K.W. Ross, G. Urvoy-Keller, Hierarchical Peer-to-Peer Systems, 2003, *Euro-Par 2003*, Klagenfurt, Austria.

A. Mislove, P. Druschel. Providing Administrative Control and Autonomy in Structured Peer-to-Peer Overlays. IPTS 2004.

M. Castro, P. Druschel, A.-M. Kermarrec, A. Rowstron. One Ring to Rule them All: Service Discovery and Binding in Structured Peer-to-Peer Overlay Networks. SIGOPS European Workshop 2002.

Unstructured Overlay : LMS

- Publishing an item involves storing replicas of the item at a number of randomly selected local minima; retrieving an item involves querying randomly selected local minima until one is found that holds a replica.
- "Its performance approaches that of DHTs on networks of similar size"



(a) 1-hop Neighborhoods (b) 2-hop Neighborhoods

Figure 1: Local minima, basins and deterministic forwarding. Nodes are labelled with their distances from the key. Concentric circles denote local minima, and shaded regions their basins. Arrows indicate the path from a node towards its local minimum. (a) All forwarding arrows for $h = 1$. (b) Forwarding arrows for two nodes with $h = 2$. A dashed arrow indicates the target towards which a node forwards a probe when different than the next hop.

Morselli, R., Bhattacharjee, B., Srinivasan, A., and Marsh, M. A. 2005. Efficient lookup on unstructured topologies. In *Proceedings of the Twenty-Fourth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing* (Las Vegas, NV, USA, July 17 - 20, 2005). PODC '05.

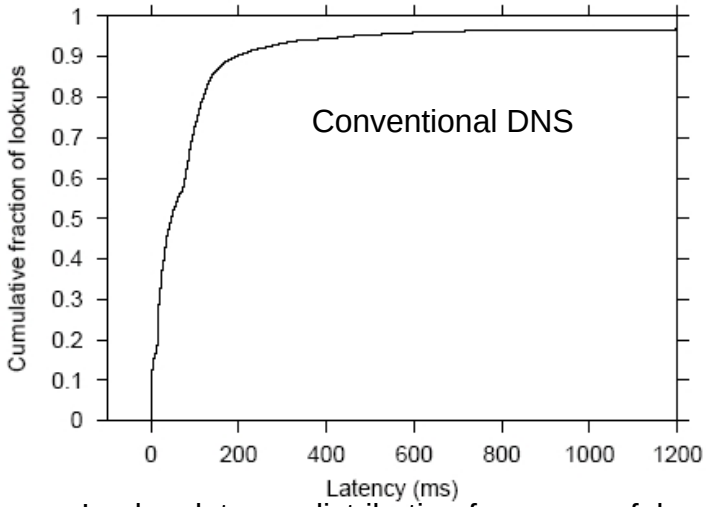
Structured vs Unstructured overlays:

Miguel Castro, Manuel Costa, and Antony Rowstron, Debunking Some Myths About Structured and Unstructured Overlays. Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05), Boston, MA, 2005.

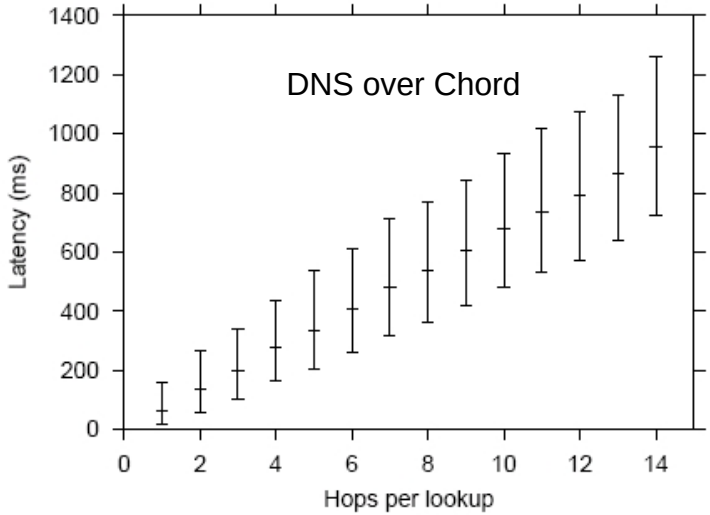
DDNS – P2P Overlay for DNS

- R. Cox, A. Muthitacharoen, and R. T. Morris, "Serving DNS using a Peer-to-Peer Lookup Service," in IPTPS, Mar. 2002 [MIT]
  - Prototype at http://distributeddns.sourceforge.net [U. Zurich]
- Simulation
  - Data taken from a study of DNS records for 1000 DNS nodes
  - Inserted 120K RRs into 1K Chord nodes, then ran 1 day equivalent set of queries (~260K queries)
- Conclusions
  - lookups in DDNS take longer than lookups in conventional DNS
    - median response time is **350ms** vs conventional DNS's **~ 43ms**.
    - increased the median in favor of removing the large tail: lookups taking 60s simply cannot happen in our system.
  - "In our judgement, using DDNS would prove a worse solution for serving DNS data than the current DNS."

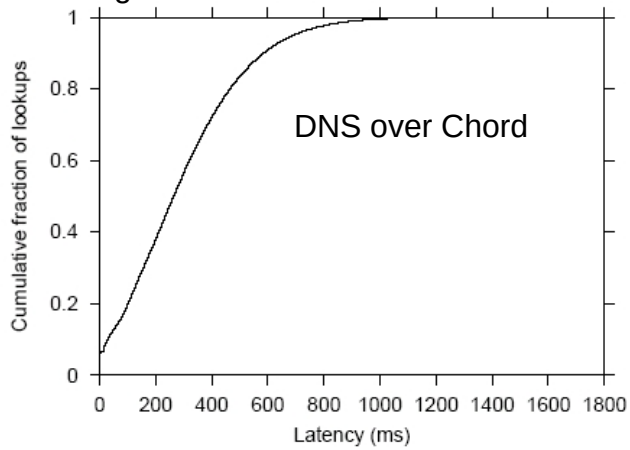
DNS Over Chord



Lookup latency distribution for successful oneserver queries over conventional DNS in the Jung. *et al.* data.



Hops per lookup in the simulation of the Jung. *Et al.* traces over Chord.

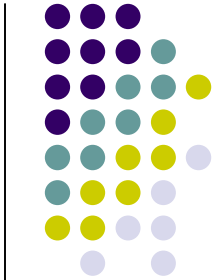


Imaginary latency to perform DNS lookups over Chord, assuming the latency distribution from the Jung. *et al.* trace

R. Cox, A. Muthitacharoen, and R. T. Morris, "Serving DNS using a Peer-to-Peer Lookup Service," in IPTPS, Mar. 2002

Research Issues

- Topological awareness
  - M. Castro, P. Druschel, Y. C. Hu, A. Rowstron. Topology-aware routing in structured peer-to-peer overlay networks. In *FuDiCo 2002: International Workshop on Future Directions in Distributed Computing*. University of Bologna Residential Center Bertinoro (Forli), Italy, June 2002.
- P2P-specific vulnerabilities
  - Doucer, J. *The Sybil Attack*. In 1st Intl. Workshop on Peerto -Peer Systems (2002).
  - Atul Singh, Miguel Castro, Peter Druschel and Antony Rowstron Defending against Eclipse attacks on overlay networks SIGOPS European Workshop, Leuven, Belgium, Sept. 2004
  - J. Liang, N. Naoumov, and K.W. Ross, The Index Poisoning Attack in P2P File-Sharing Systems, submitted.
- NAT Traversal
  - B. Ford, P. Srisuresh. Peer-to-Peer Communication Across Network Address Translators.



Further Information

- P2PRG Bibliography
  - http://www.cs.umd.edu/projects/p2prg/bib/
- P2P Reading List
  - http://cis.poly.edu/~ross/p2pTheory/P2Preading.htm