PPRL

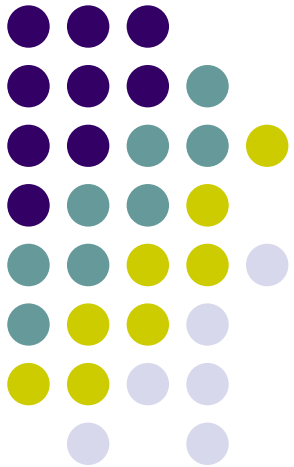
Panasonic ideas for life

Tork: A Variable Hop Overlay for Heterogeneous Networks

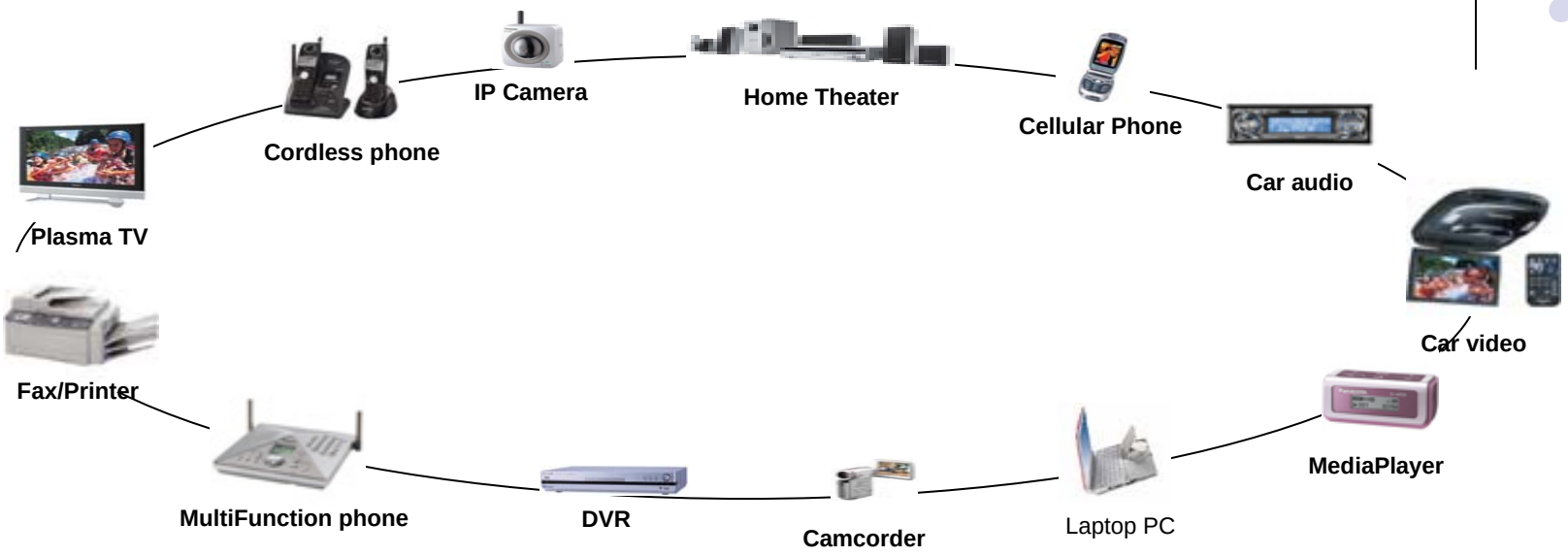
Alan Brown
University of Stirling

John Buford
(now at Avaya Labs Research)
Panasonic Princeton Lab

Mario Kolberg
University of Stirling



Motivation: ubiquitous wireless broadband



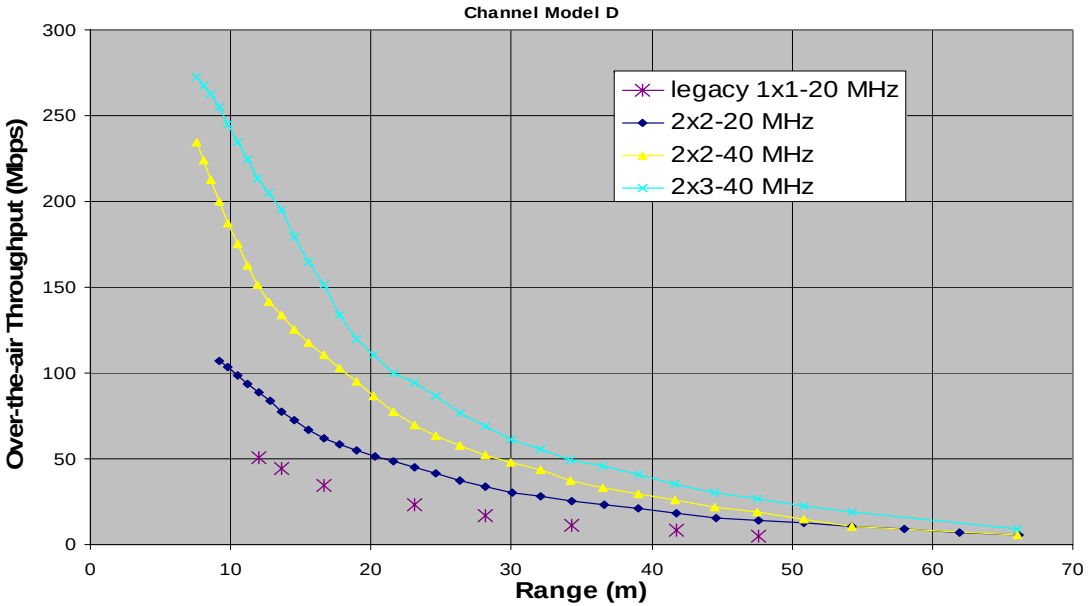
- Next generation CE devices will be broadband wireless capable
- Devices will be IP-enabled, mobile, and consumers will own many more such devices than PCs
- This will enable many new applications involving content sharing, social networking, location-based and context-based interactions.
- Can we design a general way for devices to share content, services, and context without requiring server infrastructure?

Problem Statement

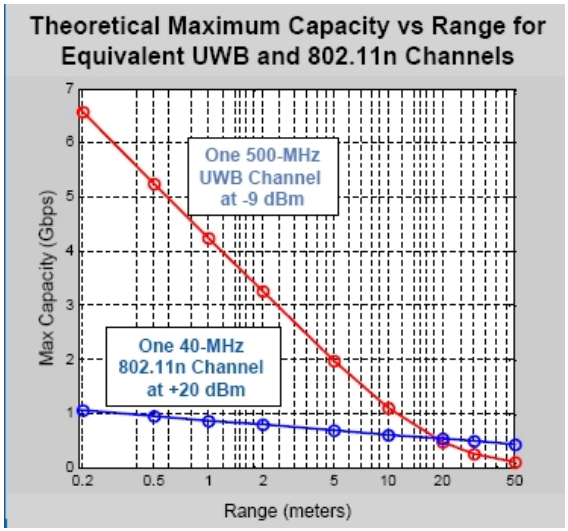
- Peer-to-peer overlays enable
  - End-to-end connectivity
  - Adaptive
  - Highly scalable architecture
  - Low barrier of entry deployment
  - General purpose
    - Search, wide-area service discovery, application layer multicast
- Issues
  - How to improve latency performance (versus multi-hop)
  - How to support heterogeneous peer population

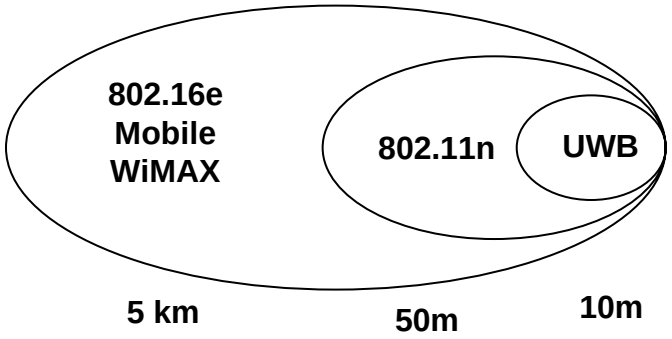
Distance vs Throughput Examples

802.11n (Source: R. Stacey, Intel)

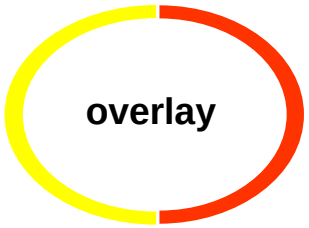


UWB vs 802.11n (Source: D. Leeper, Intel)

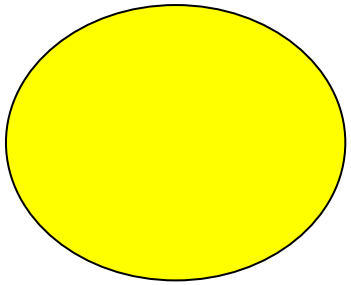




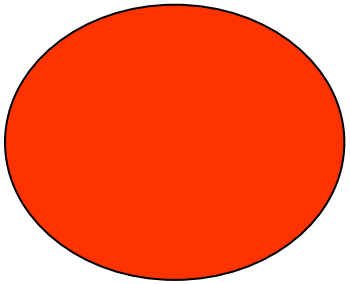
Distribution and Density



Certain regions of the network may be heavily M or H



What is the performance over the range?

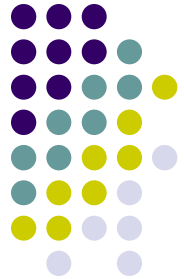


Density of M and H peers may vary

Heterogeneity vs Adaptivity

- Heterogeneity
  - Different devices in the overlay have different capacities for CPU, storage, access BW
  - These capacities for each device are relatively stable over time
- => There are few transitions between H-, M-, L- states.

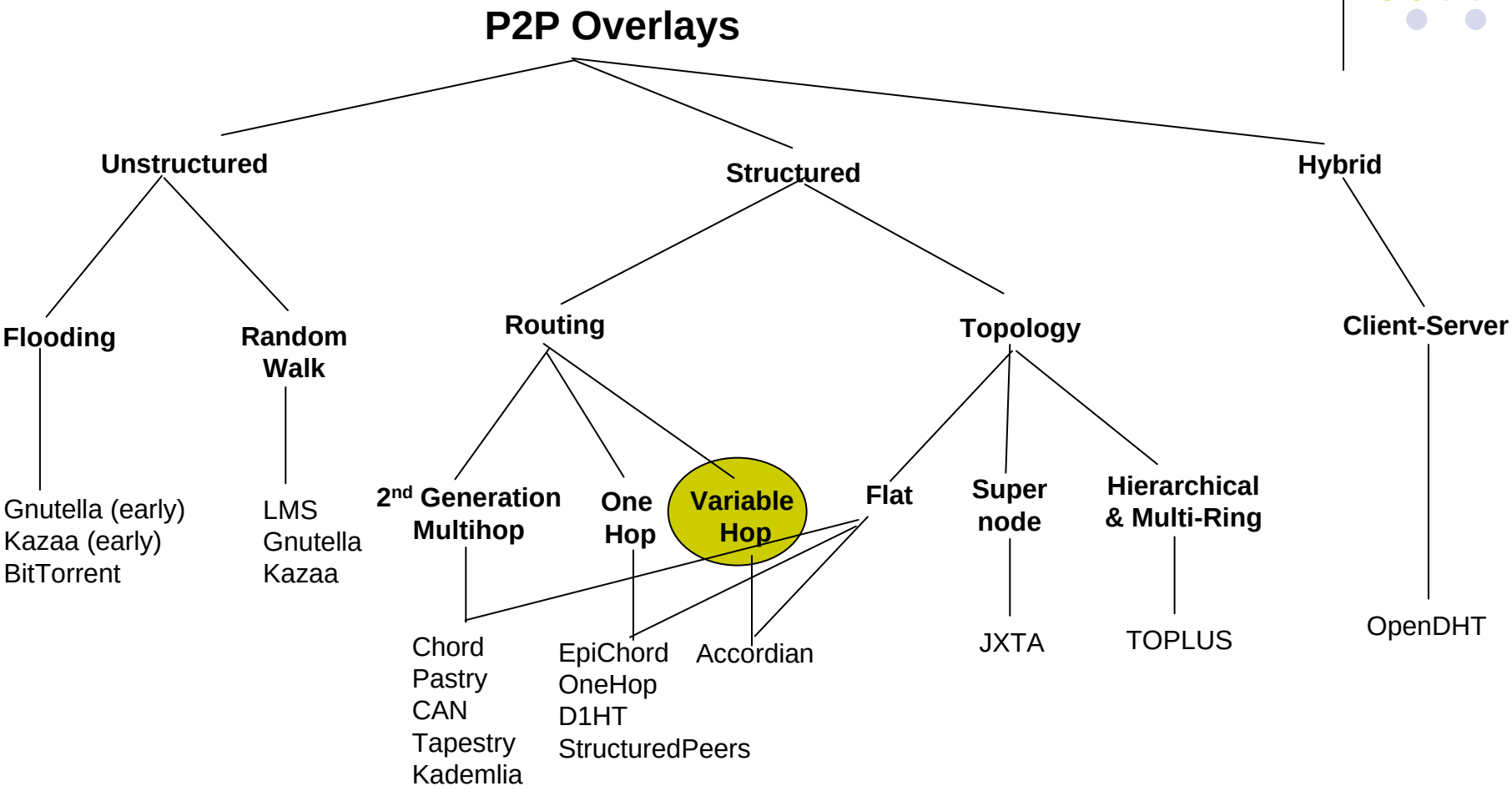
- Adaptivity
  - In a given access network, access BW varies
    - Distance from access point / base station
    - Interference
    - => could be frequent transitions between H-, M-, and L- states
  - Devices support multiple network interfaces
    - Devices roam and encounter different access BW
    - => If multi-homed, transitions could be masked by high bandwidth interface



The variable hop overlay approach

- What?
  - Each peer in the overlay has bandwidth budget that is allocated to routing table maintenance
  - Higher budget means more routing table updates are exchanged, leading to higher routing table accuracy
  - Each peer manages its budget independently
- Why?
  - Devices have heterogeneous resources and access network capacity
  - Latency matters
  - Many nodes have the capacity for more routing table accuracy
  - Doesn’t penalize the low bandwidth nodes

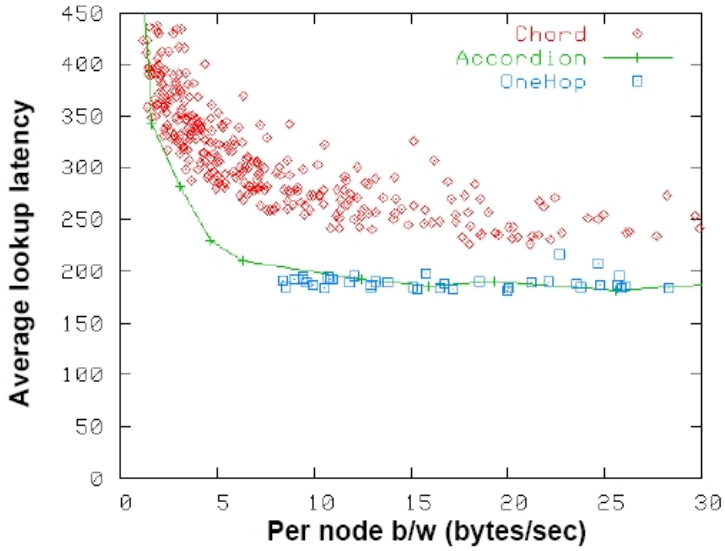
Taxonomy



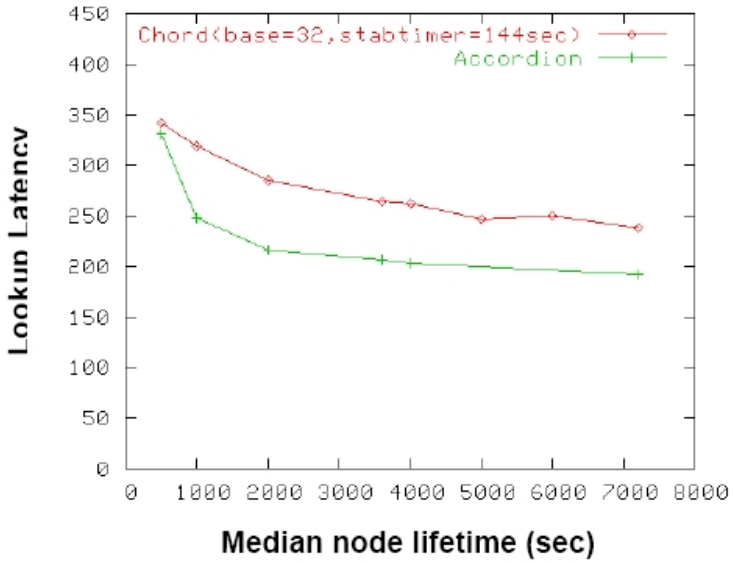
From: J. Buford & K. Ross, P2P Overlay Design Overview. P2P SIP Ad Hoc, Nov 2005

Accordion – 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

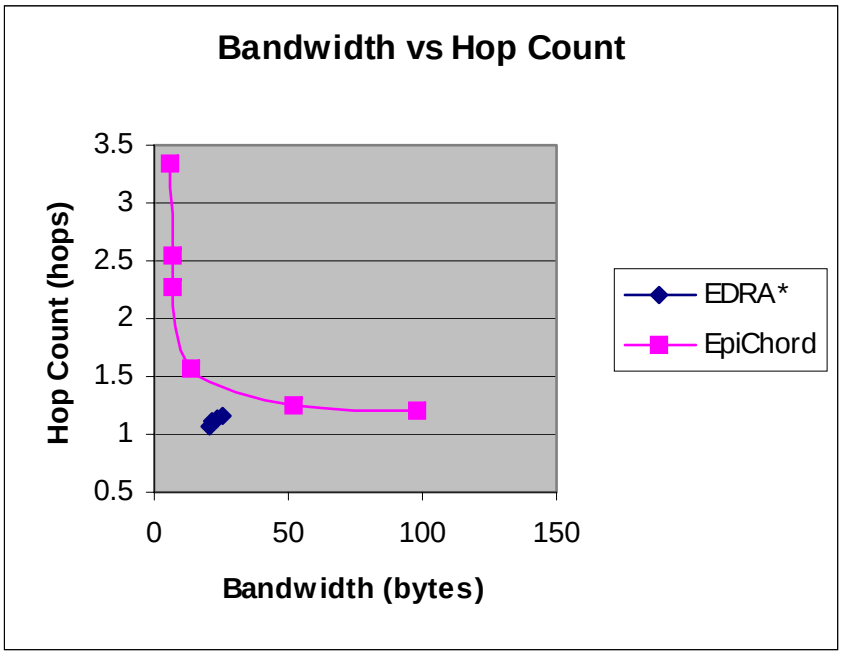
Jinyang Li, Jeremy Stribling, Robert Morris and M. Frans Kaashoek. Bandwidth-efficient management of DHT routing tables. In the Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05), Boston, MA, 2005.

Accordion

- Routing table density decreases with distance from the node
- Lookups
  - Recursive, since each node has more knowledge about local region of address space
  - Each node forwards lookups with parallelism determined by its available bandwidth
- Routing table maintenance
  - Learns routing table entries from incoming lookups and from responses to lookups
  - Estimates node liveness based on the node's lifetime and when it last heard from the node
  - Excess budget used to explore regions of routing table
    - Exploration lookups biased towards nodes with higher bandwidth budgets
- Bandwidth budget
  - Has an average budget and a burst budget
  - Each node counts (in time window) only outgoing requests and responses for its budget
- Has performance comparable to OneHop algorithm

Tork Approach

- Combine reactive and active stabilization maintenance
  - Based on two O(1)-hop overlays, EpiChord and D1HT/EDRA
  - Our simulations show EpiChord and EDRA are complementary
- Modify EDRA to operate over a wide bandwidth range

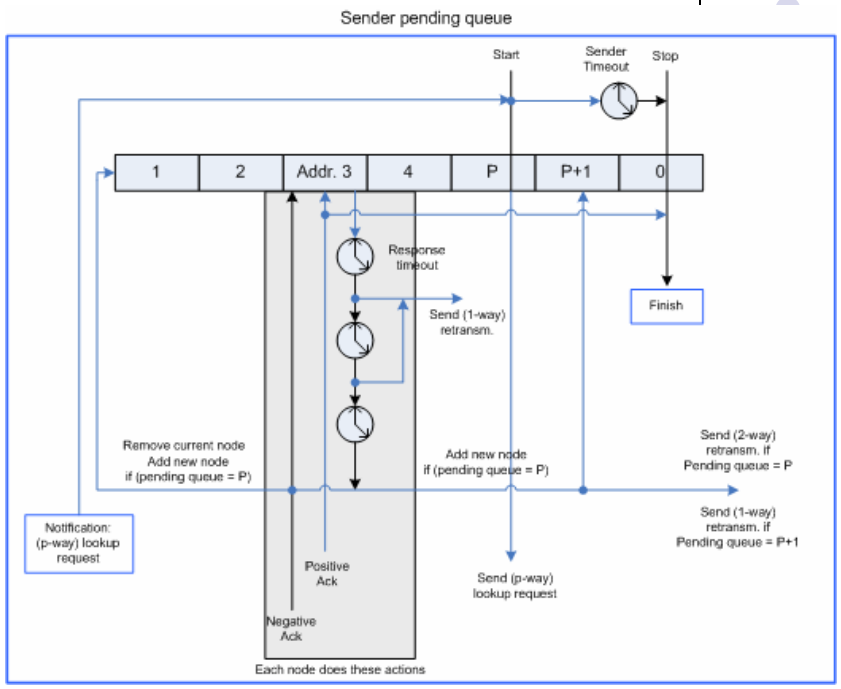


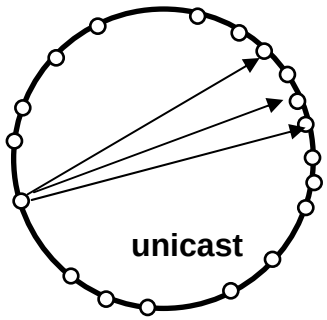
EpiChord – O(1)-Hop Overlay

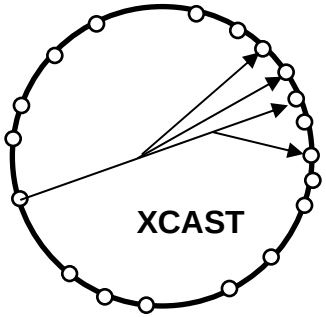
- *Reactive* routing state management strategy where routing state maintenance costs are amortized into the lookup costs.
- Nodes piggyback network information on query replies to keep their routing state up-to-date under reasonable traffic conditions.
  - Only sends probes as a backup mechanism if lookup 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

Simulation: EpiChord O(1)-hop overlay

- Routing table is organized in slices
- Slice density is highest in region near peer
- Each slice must have at least 2 entries
- DHT lookups and slice maintenance use parallel unicast requests
  - Failed responses are used iteratively to update routing table and narrow the search

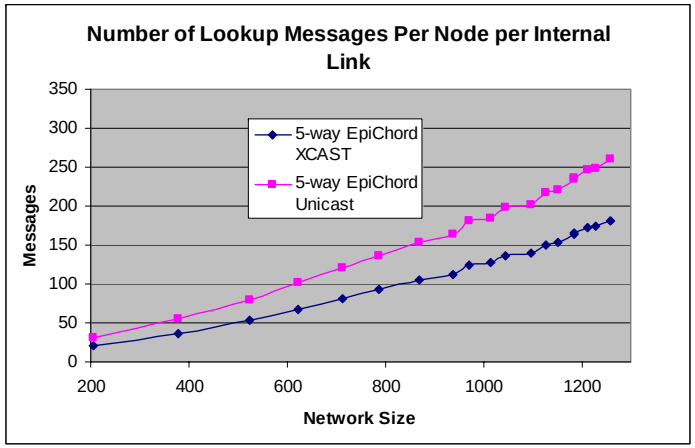


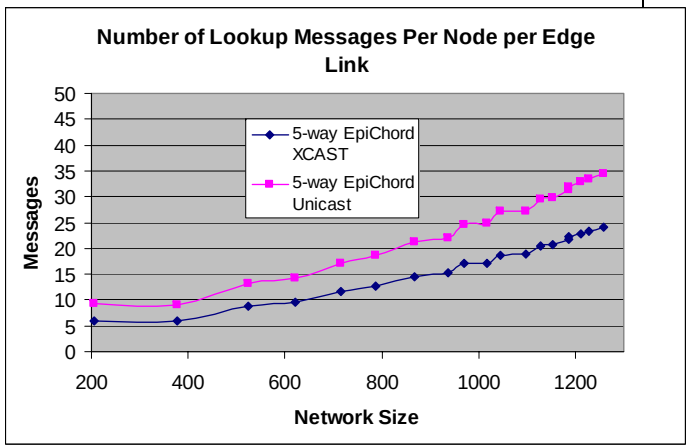




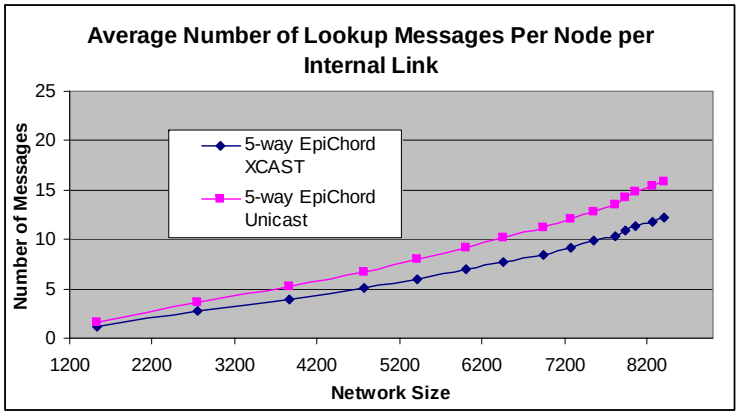
Unicast vs XCAST EpiChord

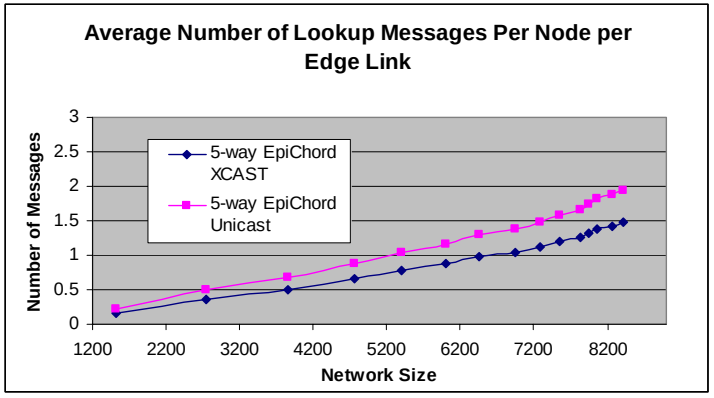
Lookup intensive workload, 1K peers





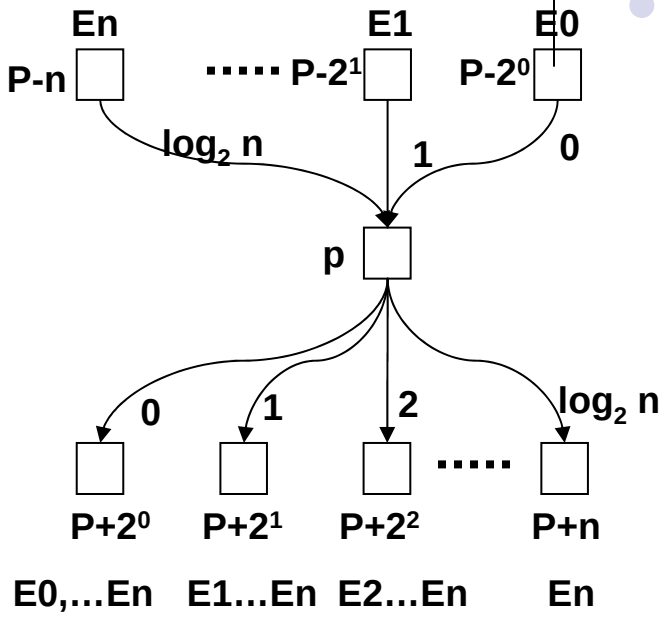
Churn intensive workload, 9K peers



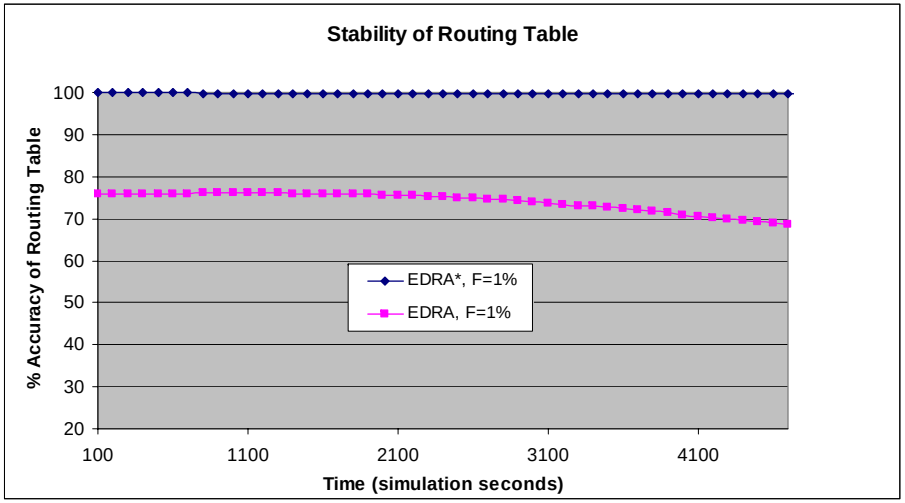


Simulation: EDRA (used in D1HT)

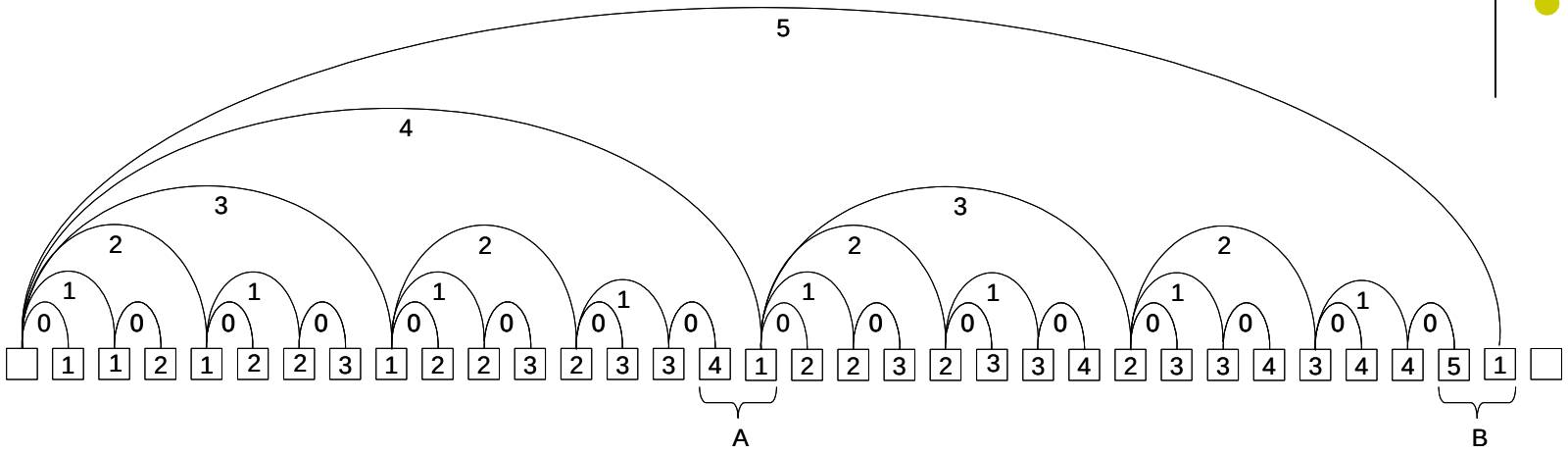
- EDRA (Event Detection and Recording Algorithm)
- Each peer collects join and leave events
- Propagates events to (log n) successors
- No peer receives duplicate events



- We fixed 5 problems with published EDRA algorithm and simulated EDRA* with XCAST



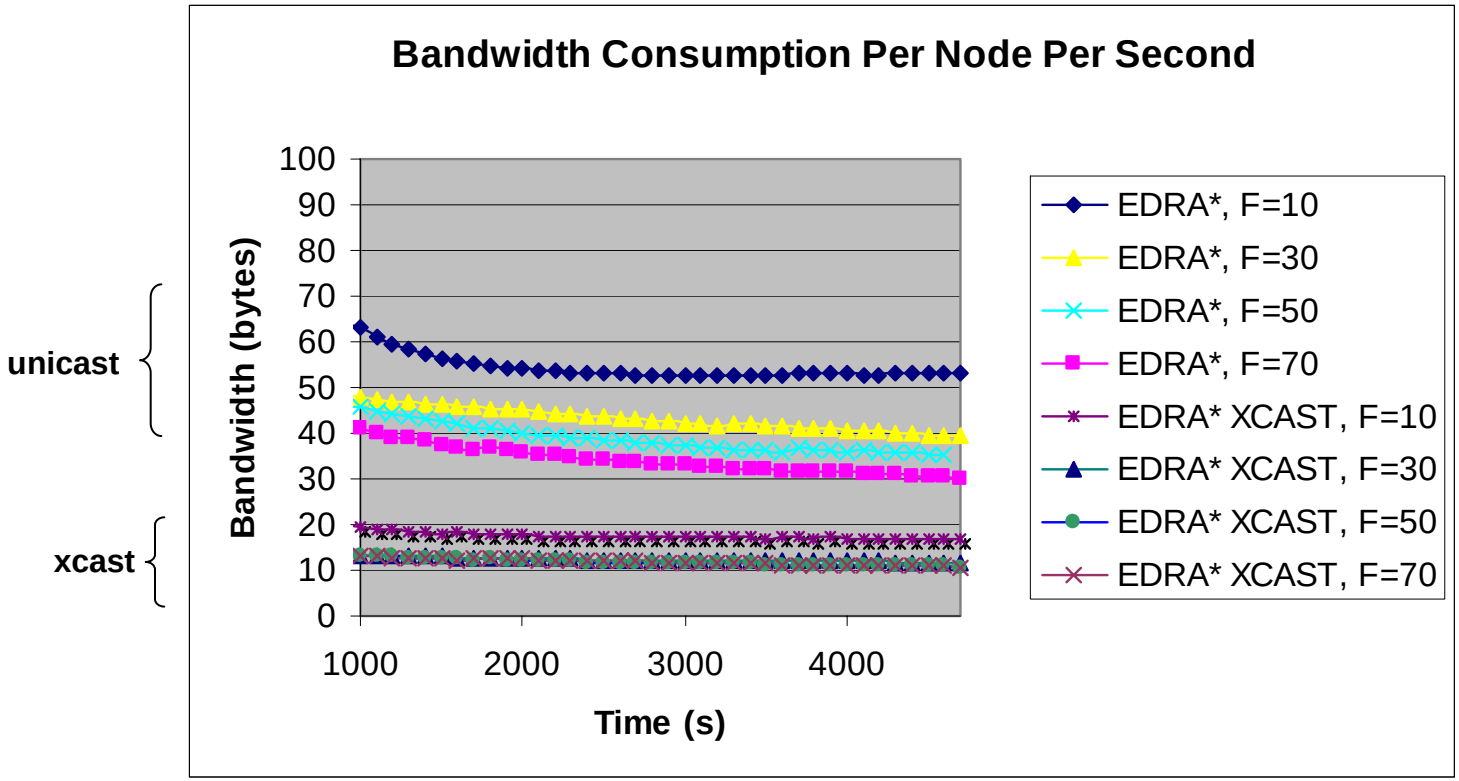
EDRA*



EDRA* Technique	Summary
Explicit join interval	Joining node gets events from node which provided copy of routing table
Correct join point	Successor node checks new predecessor is in correct position
Forwarding of un-acknoweldged events	Events are forwarded to successors of peer
Handling of duplicate events	Forwards duplicate events that occur due to routing table errors
Detecting concurrent adjacent events	New nodes contact both successor and predecessor nodes
Event cache propagation	Events are cached and forwarded as routing table changes reach to new nodes in overlay

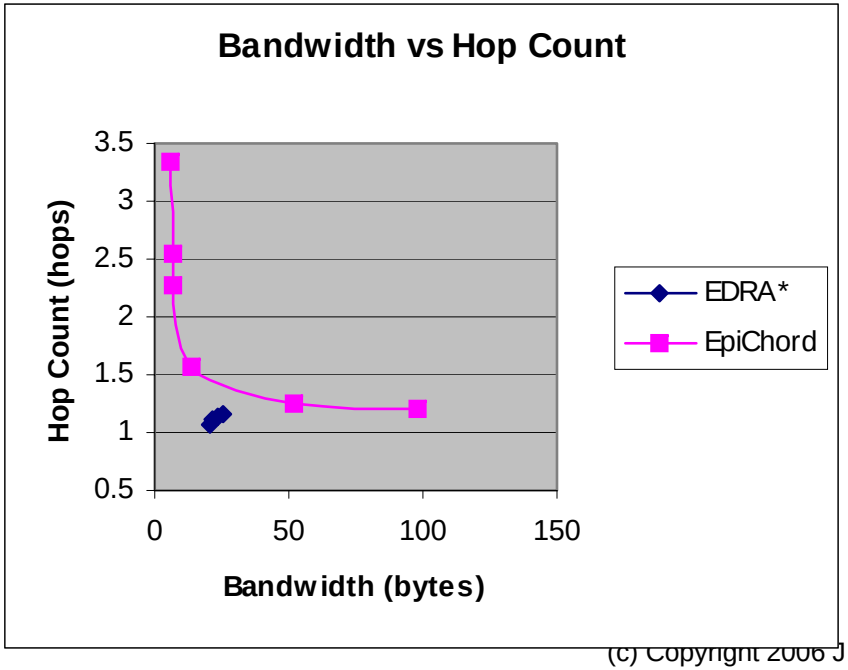
EDRA* vs EDRA*-XCAST

Overall savings about 33% for n = 1024



Hop Count versus Bandwidth Use

- EpiChord - opportunistic maintenance, achieves < 2.5 hops with around 50% routing table accuracy (churn intensive workload)
- EDRA* - active stabilization, achieves < 1.5 hops with more than 95% routing table accuracy

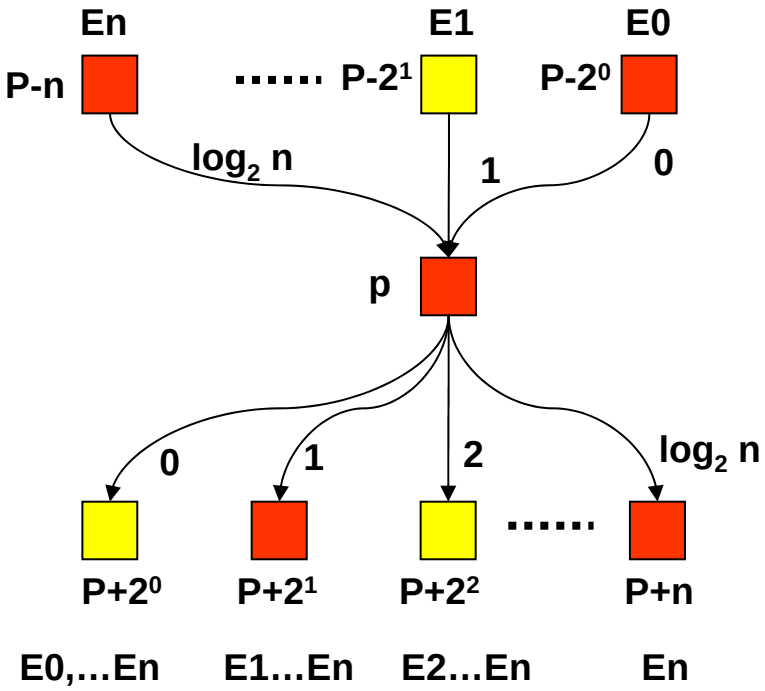


Churn intensive workload
1K to 10K overlay
Mean lifetime 1 hour

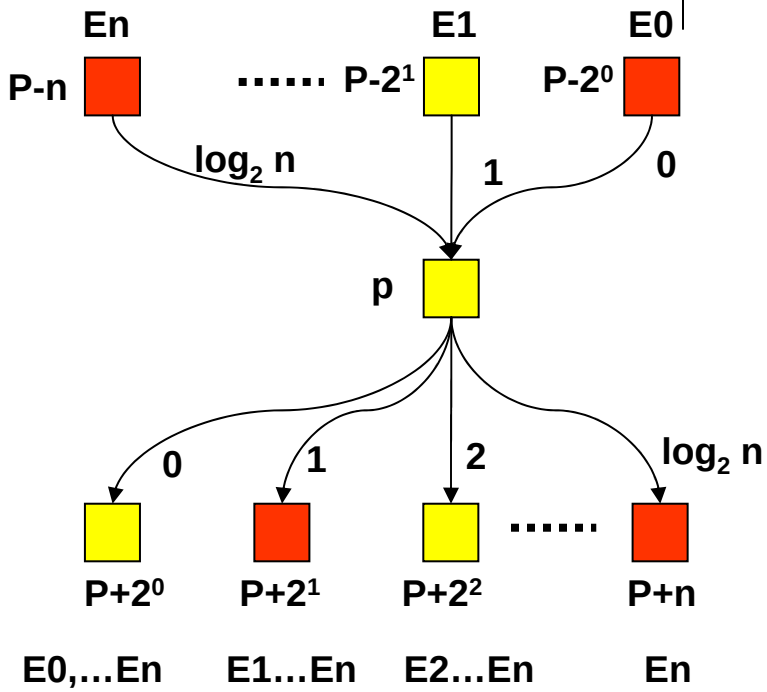
EDRA-EC Co-Existence

- BW allocation
  - Each peer has a maintenance bandwidth budget B
  - It divides that budget into EDRA* and EC maintenance messaging
  - The ratio may vary according to the value of B
    - i.e., if B is large, then EDRA* may run at normal theta value
  - Once EDRA* reaches normal theta value, it reaches limit
- Selection of θ
  - If there is sufficient bandwidth, chose θ according to usual EDRA* formulation
  - If not, each peer emits event reports as BW becomes available
- Rate control
  - Adjacent M-H and H-M combinations in the EDRA forwarding graph require adaptation, see next slides.

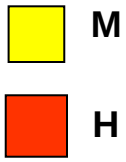
Rate Asymmetry #1



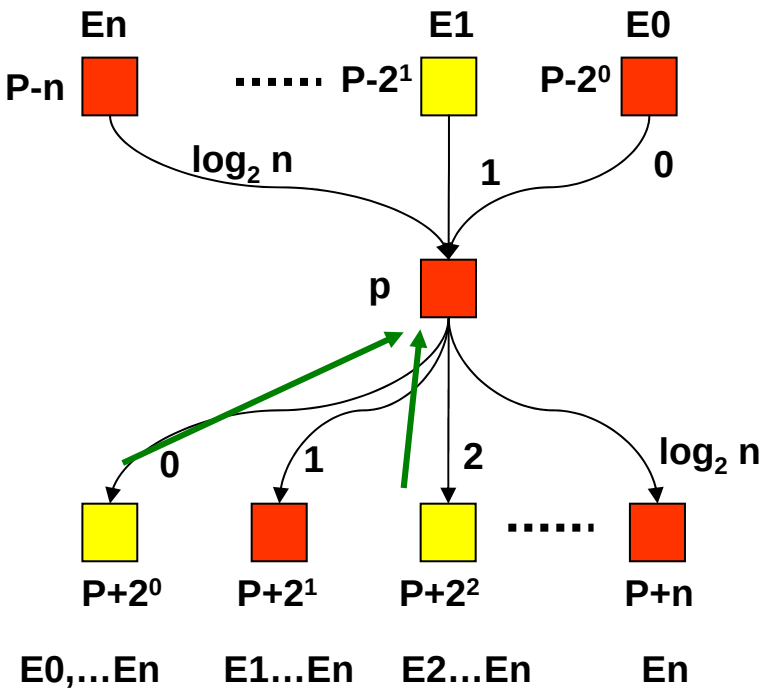
Node p is not a bottleneck,
but it may overwhelm downstream M peers

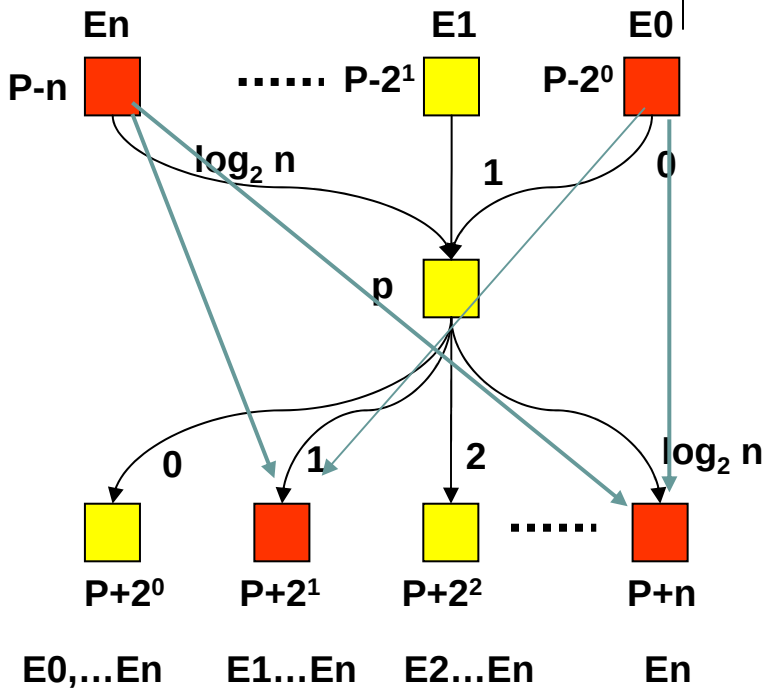


Node p is a bottleneck



Rate control of adjacent peers





Node p is not a bottleneck,
but it may overwhelm downstream M peers

Node p is a bottleneck

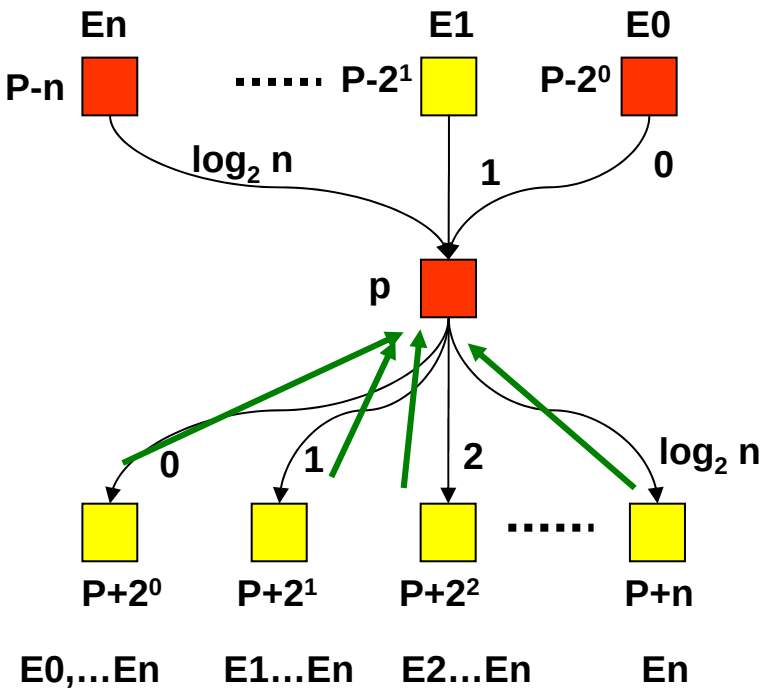
M

H

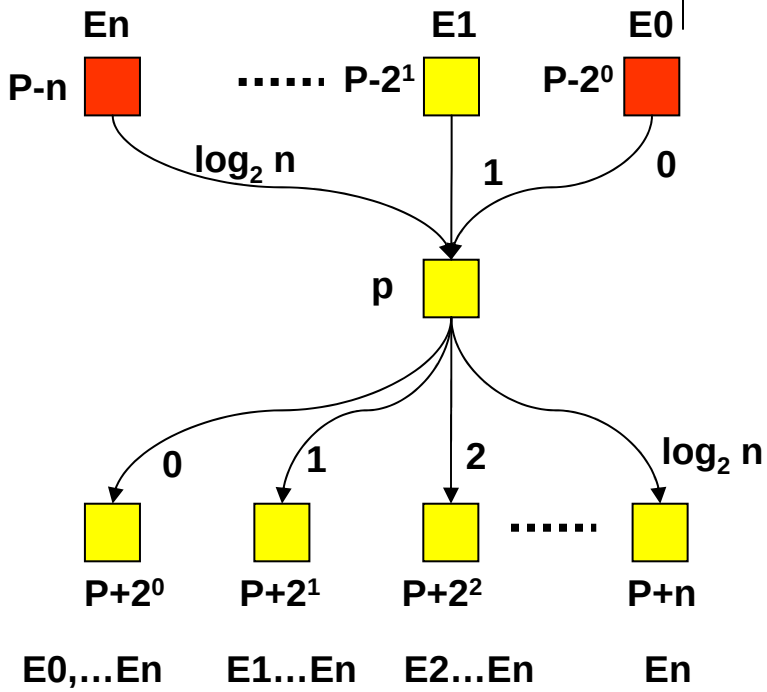
By-pass links from H to H

Rate control

Rate Asymmetry #2



Node p is not a bottleneck,
but if all neighbors are M
then update rate is reduced

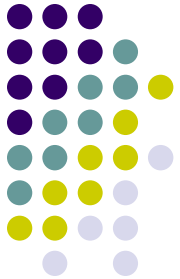


Node p is a bottleneck
and there are no by-pass links to P's neighbors

M

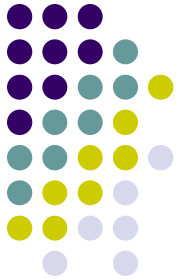
H

All H nodes arranged in contiguous predecessor-successor links
If p is blocked, it propagates events to its next H successor on the ring



Lost Events

- Lost events
  - EDRA* already loses some events due to routing table errors
  - Mixing M and H peers in EDRA* ring means that M peers may not be able to receive or forward events at a sufficient rate
- Approach
  - Priority scheme such that low priority events are dropped before high priority events (High) H-peer join
    - H-peer leave
    - L-peer join
    - L-peer leave
  - M peers will have reduced routing table accuracy but parallel lookup (as in EpiChord) will mask this to some degree



Rate of Incoming DHT Requests

- Peer type exchanged with routing table updates
- Request routing is biased toward H peers in a given region

Assessment

- Low latency overlay is important for many applications
- Variable hop overlays offer performance potential of O(1) hop overlays for capable nodes while accomodating less capable nodes
- Tork offers low hop-count for given bandwidth budget, providing a wider operating range
- Tork is targeted at overlays supporting heterogeneous devices, or course grained adaptivity scenarios
  - Granularity related to time to propagate events through out the overlay

Thank you!