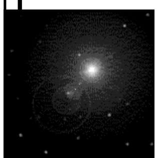


A Course on Foundations of
Peer-to-Peer Systems & Applications

CS 6/75995
Foundation of Peer-to-Peer

Applications & Systems

Kent State University

Dept. of Computer Science

www.cs.kent.edu/~javed/class-P2P08

PASTRY

[mechanics]

- Update overview
- 1 class start+routing+node failure

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry [update.. Old]

- Overview
  - Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, Antony Rowstron and Peter Druschel, 2001
- Topology
  - Consistant Hashing
  - Key Space
- Routing
  - Leaf Set
  - Numerically Closest Set
  - Physically Closest Set
- Node Arrival
  - Bootstrapping
  - Finding a Zone
  - Joining the Routing (Route Table Updates)
- Node Departure
  - Identification of Takeover Node
  - Recovery Algorithm
- Performance Analysis
- Evaluation
  - Stability
  - Robustness
  - Load balancing

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry
Topology

Pastry

- An overlay network that provides a self-organizing routing and location service (like *Chord*).
- Seeks to minimize the "distance" (scalar proximity metric like *routing hops*) messages travel.
- Expected number of routing steps is **O(log N)**; N=No. of Pastry nodes in the network

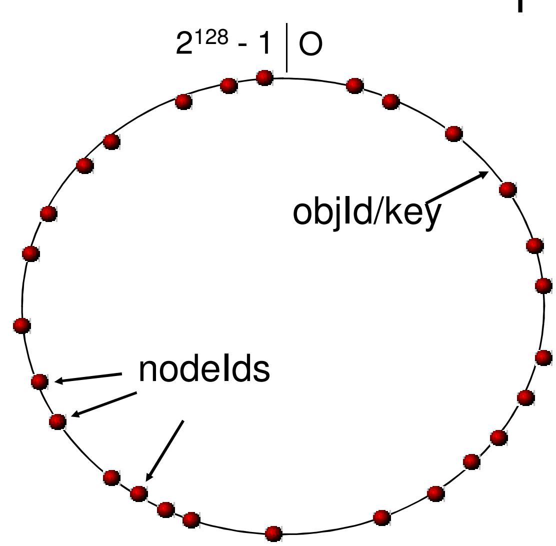
Pastry Topology

- Nodes are organized in a circular ID space, using consistent DHT hashing.
- *NodeId* randomly assigned from {0, .., 2^{128}-1}
- A pastry node can route to the numerically closest node to a given key in less than $\boldsymbol{log_{2b} N}$ steps. (*b,* |*L*| are configuration parameters)
- Despite concurrent node failures, delivery is guaranteed unless more than **|*L*|/2** nodes with adjacent NodeIds fail simultaneously
- Each node join triggers **O(*log*$_{2b}$ *N*)** messages

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Object distribution

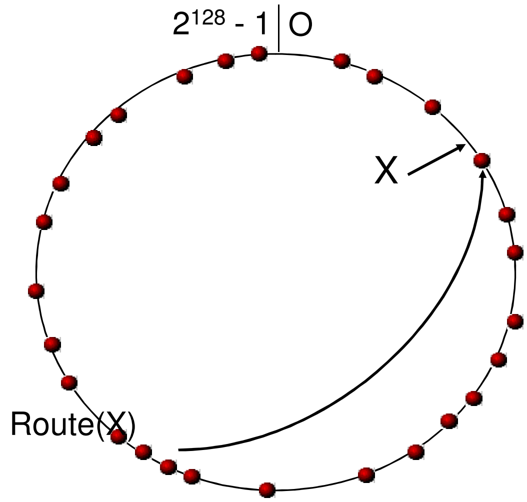
- **Consistent hashing**
- 128 bit circular id space
- *nodeIds* (uniform random)
- *objIds/keys* (uniform random)
- **Invariant:** node with numerically closest nodeId maintains object



FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Object insertion/lookup

- Msg with key *X* is routed to live node with nodeId closest to X
- **Problem:** complete routing table not feasible

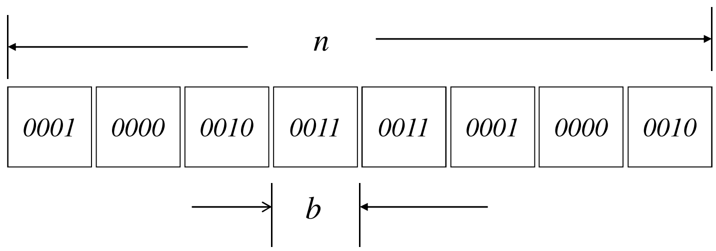


FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry Routing

Node ID

- *NodeIds* are in base 2^b



NodId#10233102

FOUNDATION OF PEER-TO-PEER SYSTEMS

Three Concept of Proximity

Leaf set

	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Set of nodes with |L|/2 smaller and |L|/2 larger numerically closest NodeIds

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Prefix-based routing entries

Neighborhood set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

|M| “physically” closest nodes

FOUNDATION OF PEER-TO-PEER SYSTEMS

Routing Table Dimensions

L nodes in leaf set (typical $L = 2^b$)

$\log_{2^b} N$ Rows (actually $\log_2 b\ 2^{128} = 128/b$)

2^b columns

M neighbors (typical $M = 2 \times 2^b$)

NodeId 10233102

Leaf set

SMALLER		LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-13
FP2P08, javed@kent.edu
Javed I. Khan@2008

How to select b?

- *NodeIds* are in base 2^b
- One row for each prefix of local NodeId ($Log_{2b}\ N$ populated on average)
- One for each possible digit in the NodeId representation $2^b - 1$ columns

b defines the tradeoff:

(Log_{2b} N) x ($2^b - 1$) entries Vs. Log_{2b} N routing hops

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-14
FP2P08, javed@kent.edu
Javed I. Khan@2008

Pastry: Prefix Table (# 65a1fc*x*)

Row x0	0x	1x	2x	3x	4x	5x		7x	8x	9x	ax	bx	cx	dx	ex	fx
Row x1	60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex	6fx
Row x2	650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex	65fx
Row x3	65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex	65afx

$\log_{16}$ N rows

FOUNDATION OF PEER-TO-PEER SYSTEMS

A Hypothetical Pastry node with ID 10233102

NodeId 10233102

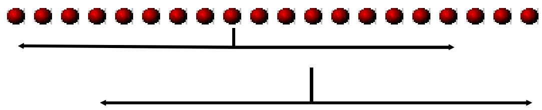
Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

- Values: b = 2, and l = 8. All numbers are in base 4.
- The top row of the routing table is row zero.
- The entries are *common prefix with 10233102 - next digit - rest of nodeId*.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Leaf Sets



NodeId 10233102

Leaf set	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

- In leaf set each node maintains IP addresses of the nodes with the |L|/2 numerically closest larger |L|/2 |L|/2 smaller numerically closest nodeIds.

- Routing efficiency/robustness
- Fault detection (keep-alive)
- Application-specific local coordination

FOUNDATION OF PEER-TO-PEER SYSTEMS

Neighborhood Set

- The neighborhood set M contains nodeIDs and IP addresses of |M| nodes those are physically closest (or as per some other proximity metric) to the local node.

- Its use will be discussed in “proximity routing” discussion.

NodeId 10233102

Leaf set	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood set

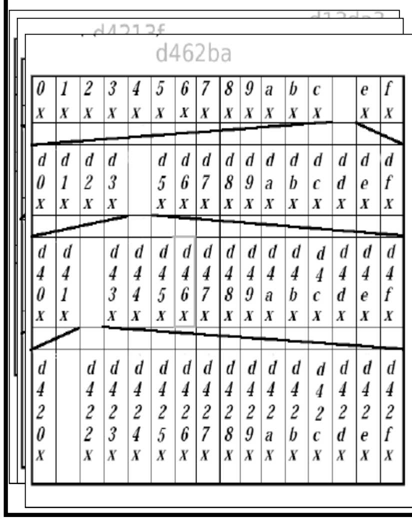
13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

FOUNDATION OF PEER-TO-PEER SYSTEMS

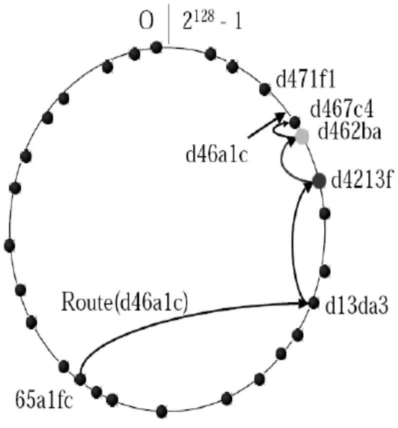
Find (d46a1c)

- Route Table of A 65a1fc
- Route Table of B d13da3
- Route Table of C d4213f

- 65a1fc find B (d13da3)
- d13da3 finds C (d4213f)
- d4213f finds D(d462ba)

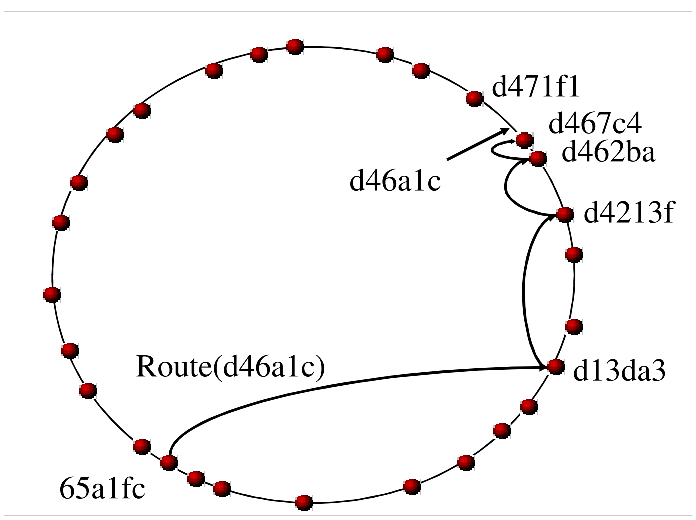


0 x	1 x	2 x	3 x	4 x	5 x	6 x	7 x	8 x	9 x	a x	b x	c x		e x	f x
d 0 x	d 1 x	d 2 x	d 3 x		d 5 x	d 6 x	d 7 x	d 8 x	d 9 x	d a x	d b x	d c x	d d x	d e x	d f x
d 4 0 x	d 4 1 x		d 4 3 x	d 4 4 x	d 4 5 x	d 4 6 x	d 4 7 x	d 4 8 x	d 4 9 x	d 4 a x	d 4 b x	d 4 c x	d 4 d x	d 4 e x	d 4 f x
d 4 2 0 x		d 4 2 2 x	d 4 2 3 x	d 4 2 4 x	d 4 2 5 x	d 4 2 6 x	d 4 2 7 x	d 4 2 8 x	d 4 2 9 x	d 4 2 a x	d 4 2 b x	d 4 2 c x	d 4 2 d x	d 4 2 e x	d 4 2 f x



FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Routing



Properties
- $\log_{16}$ N steps
- O(*log N*) state

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry Routing Algorithm

if ($L_{-\lfloor |L|/2 \rfloor} \le D \le L_{\lfloor |L|/2 \rfloor}$) {
    // D is within range of our leaf set
    forward to L_i, s.th. $|D - L_i|$ is minimal;
} else {
    // use the routing table
    Let $l = shl(D, A)$;
    if ($R_l^{D_l} \ne null$) {
        forward to $R_l^{D_l}$;
    }
    else {
        // rare case
        forward to $T \in L \cup R \cup M$, s.th.
            $shl(T, D) \ge l$,
            $|T - D| < |A - D|$
    }
}

(1) Single hop

(2) Towards better prefix-match

(3) Towards numerically closer *NodeId*

D: Message Key
L_i: i^{th} closest NodeId in leaf set
shl(A, B): Length of prefix shared by nodes A and B
R^i_j: (j, i)th entry of routing table

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Routing Procedure

if (destination is within range of our leaf set)
    forward to numerically closest member
else
    let *l* = length of shared prefix
    let *d* = value of *l*-th digit in *D*'s address
    if (R_l^d exists)
        forward to R_l^d
    else
        forward to a known node that
        (a) shares at least as long a prefix
        (b) is numerically closer than this node

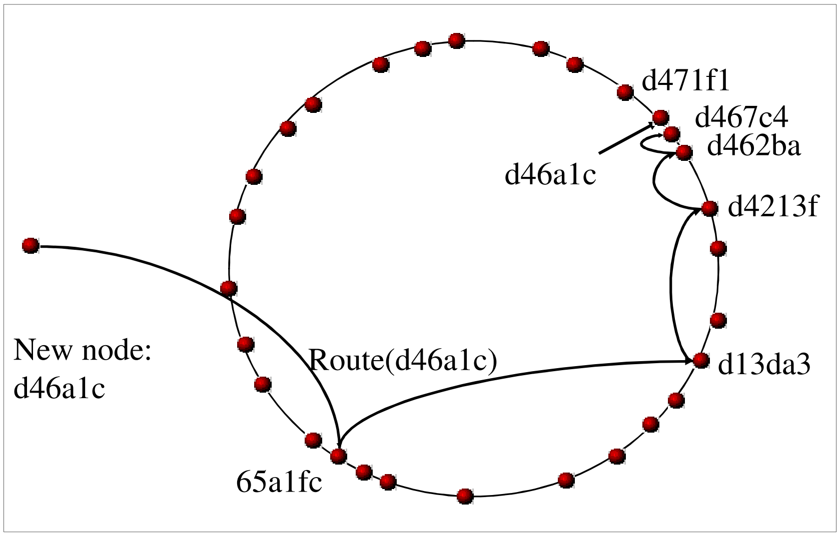
FOUNDATION OF PEER-TO-PEER SYSTEMS

Routing Performance: Intuition

- (1) – Single hop, termination
- (2) – No. of nodes which prefix-match the key upto current length reduces by 2^b
- (3) – Low probability, adds one hop

Pastry Self-Organization

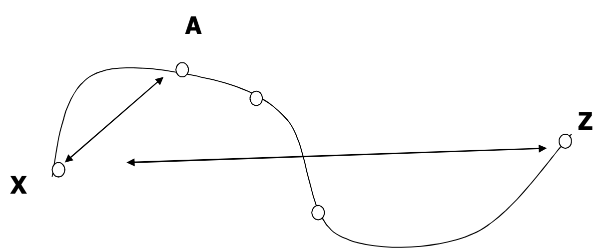
Pastry: Node Addition



FOUNDATION OF PEER-TO-PEER SYSTEMS

Self-organization: Node Arrival

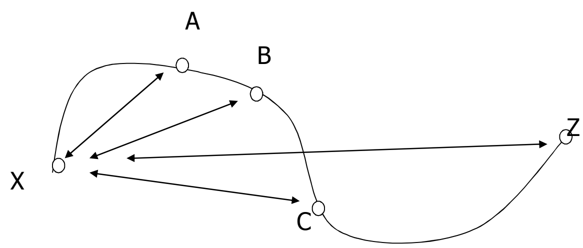
- Arriving Node X knows “nearby” node A.
- X asks A to route a “join” message with key = NodeId(X).
- Message is routed and finds Z, whose NodeId is numerically closest to NodeId(X)
- All nodes along the path A, B, …, Z send state tables to X
- X initializes its state using this information.
- X sends its state to “concerned” nodes



FOUNDATION OF PEER-TO-PEER SYSTEMS

State Initialization (1)

- X borrows A's Neighborhood Set
  - A is geographically closer to X so it is OK to borrow the set.

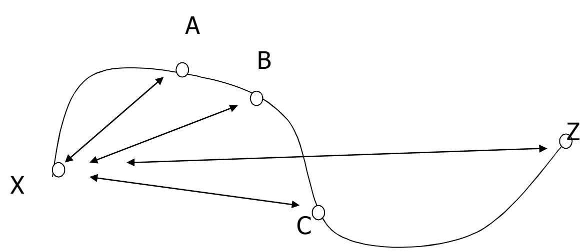


FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-27
FP2P08, javed@kent.edu
Javed I. Khan@2008

State Initialization (2)

- Z' ID is numerically closest to X's Therefore:
- X's leaf set is derived from Z's leaf set

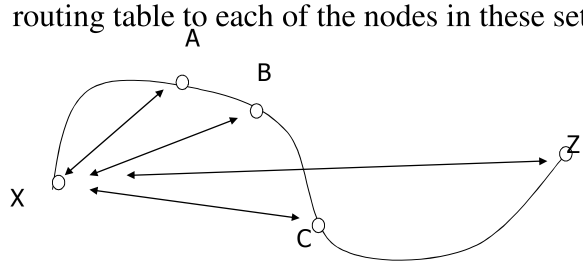


FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-28
FP2P08, javed@kent.edu
Javed I. Khan@2008

State Initialization (3)

- X_0 set to A_0
- X_1 set to B_1, X_2 set to C_2, …
- Finally, X transmits its leafset, neighborhood set and routing table to each of the nodes in these sets.



- The total message cost is $O(\log_2{}^b N)$. The constant is $3x2^b$.
- To handle concurrent arrival, extensive timestamps are used.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-29
FP2P08, javed@kent.edu
Javed I. Khan@2008

Self-organization: Node Failure (1)

- Detected when a live node tries to contact a failed node
- Updating Leaf set – get leaf set from largest index on the side of the failed node.

  $L_{-|L|/2}$ or $L_{|L|/2}$ ⇐ **|L|/2 bound on failed nodes**

- This set partially overlaps the present nodes leaf set L and extra nodes not in L.
- It thus selects the appropriate one. Verifies that it is alive and adds.

FOUNDATION OF PEER-TO-PEER SYSTEMS

LECT-10, S-30
FP2P08, javed@kent.edu
Javed I. Khan@2008

Self-organization: Node Failure (2)

FOUNDATION OF PEER-TO-PEER SYSTEMS

- Updating routing table - To repair R^d_l , ask any R^i_l $i \neq d$ in the same row for its R^d_l
- If the unlikely case its' empty (no live node), with the right prefix then it contacts any R^i_{l+1} $i \neq d$. thereby casting a wider net.
- This process is highly unlikely to fail.

LECT-10, S-31
FP2P08, javed@kent.edu
Javed I. Khan@2008

Self-organization: Node Failure (3)

FOUNDATION OF PEER-TO-PEER SYSTEMS

- Updating neighborhood set
- This is not used in routing generally.
- – Ask any alive set-members for their neighbors

LECT-10, S-32
FP2P08, javed@kent.edu
Javed I. Khan@2008

Locality

- Application provides the “distance” function
- Invariant: “All routing table entries refer to a node that is near the present node, according to the proximity metric, among all live nodes with an appropriate prefix”
- Invariant maintained on self-organization

Handling Malicious Nodes

- Routing is deterministic
- Randomize choice between multiple suitable candidates – with a bias towards the best one

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry
Analysis

Routing Performance

- The expected number of routing steps is $\log_{2^b} N$ steps, assuming accurate routing tables and no recent node failures. Consider the three cases in the routing procedure.
- If a message is forwarded using the routing table (lines 6–8), then the set of nodes whose ids have a longer prefix match with the key is reduced by a factor of 2^b in each step, which means the destination is reached in $\log_{2^b} N$ steps.
- If the key is within range of the leaf set (lines 2–3), then the destination node is at most one hop away.
- The third case arises when the key is not covered by the leaf set (i.e., it is still more than one hop away from the destination), but there is no routing table entry. Assuming accurate routing tables and no recent node failures, this means that a node with the appropriate prefix does not exist (lines 11–14). The likelihood of this case, given the uniform distribution of nodeIds, depends on $|L|$.
- Analysis shows that with $|L| = 2^b$ and $|L| = 2x2^b$, the probability that this case arises during a given message transmission is less than .02 and 0.006, respectively. When it happens, no more than one additional routing step results with high probability.
- In the event of many simultaneous node failures, the number of routing steps required may be at worst linear in N, while the nodes are updating their state. This is a loose upper bound; in practice, routing performance degrades gradually with the number of recent node failures (shown experimentally). Eventual message delivery is guaranteed unless $|L|/2$ nodes with consecutive nodeIds fail simultaneously. The probability of such a failure can be made very low.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry
Extensions: API & Applications

The Pastry API

- Operations exported by Pastry
  - nodeId = pastryInit(Credentials,Application)
  - route(msg,key)

- Operations exported by the application working above Pastry
  - deliver(msg,key)
  - forward(msg,key,nextId)
  - newLeafs(leafSet)