Pastry Advanced Idea: Proximity Routing

Teaching Note

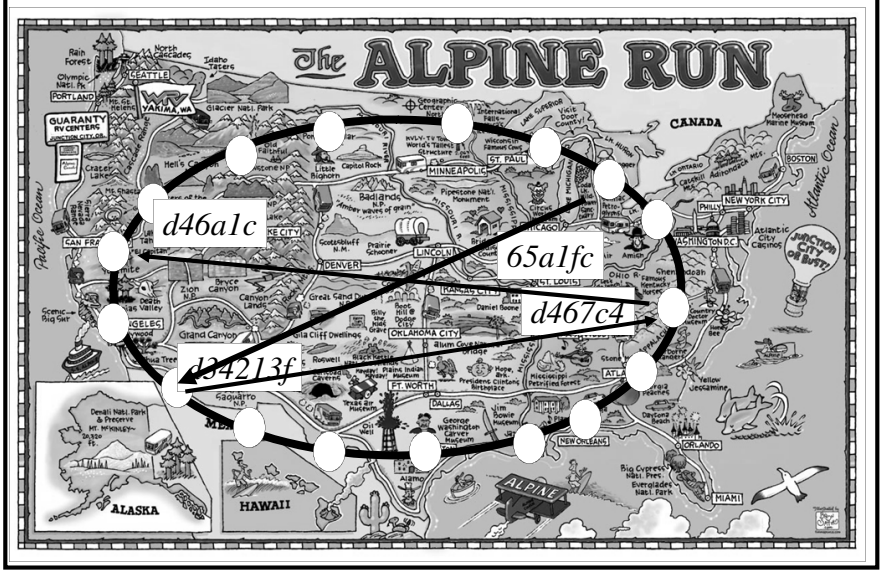
- [Euclidian distance]
- [notion of progressive distance- the rare is the address the far away we have to travel]
- [village, earth, orbit, solar system, constellation analogy]

Pastry: Proximity Routing

- The basic routing of Pastry is based on the notion of numerical proximity of the source and destination nodes ids and expected pastry routing hops.
- The demonstrated routing is complete because it will find the closest peer. The expected number of pastry hops on O(log n).
- But still it may be non-optimum in terms of physical routing hops and distances.
- However, Pastry's routing efficiency can be improved according to second notion of proximity.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Internet Distance vs. Identifier Distance



FOUNDATION OF PEER-TO-PEER SYSTEMS

Scalar Proximity Metric

- Various measures of distance can be used as a scalar proximity metric which is Euclidean or which obeys the triangulation inequality.
- Useful metric in hand is internet distance which can be approximated by quantities such as ping delay, # of IP hops, etc.
- A node can probe internet distance to any other node.
- Note: such internet distance does not strictly obey triangulation inequality, yet they are close that can take advantage of proximity routing.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Analysis of Proximity Routing

- Generally multiple nodes shares the same prefix with a given node.
- Thus, for each routing table entry there are multiple choices of nodes.
- Though it cannot be guaranteed that a node will always find out the closest node for a particular prefix, but over time it can be always improved as it keeps in touch with more node and keeps replacing nodes with closer nodes.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Proximity Invariant

- **Ideally, let us assume that following proximity property holds for each routing table:**
- **Proximity Invariant:** ***Each routing table entry refers to a node which is close to the local node in the proximity space, among all nodes with the appropriate nodeId prefix.***

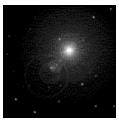
FOUNDATION OF PEER-TO-PEER SYSTEMS

Two Questions?

- **Proximity Invariant:** ***Each routing table entry refers to a node which is close to the local node in the proximity space, among all nodes with the appropriate nodeId prefix.***
- *Question 1:* Can this invariant be preserved by any practical incremental route table construction and maintenance process?
- *Question 2:* If this invariant is maintained in each routing can a packets be forwarded completely to the right node ID yet efficiently in scalar proximity metric space?

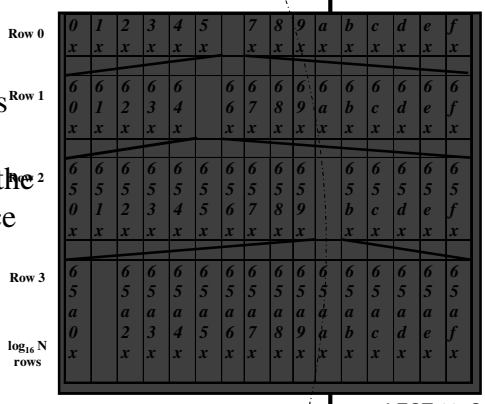
FOUNDATION OF PEER-TO-PEER SYSTEMS

Concept: Prefix Length & Euclidian Distance



FOUNDATION OF PEER-TO-PEER SYSTEMS

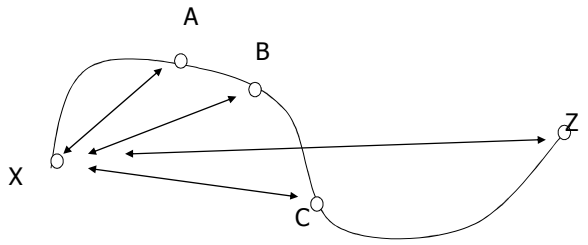
- A node which has larger prefix match with current node is further away from the current node.
- With each extra prefix match the nodes becomes more rare. Thus, with the assumption of uniform distribution of the nodeID in proximity space the distance increases exponentially.



Flashback: Route Table Construction

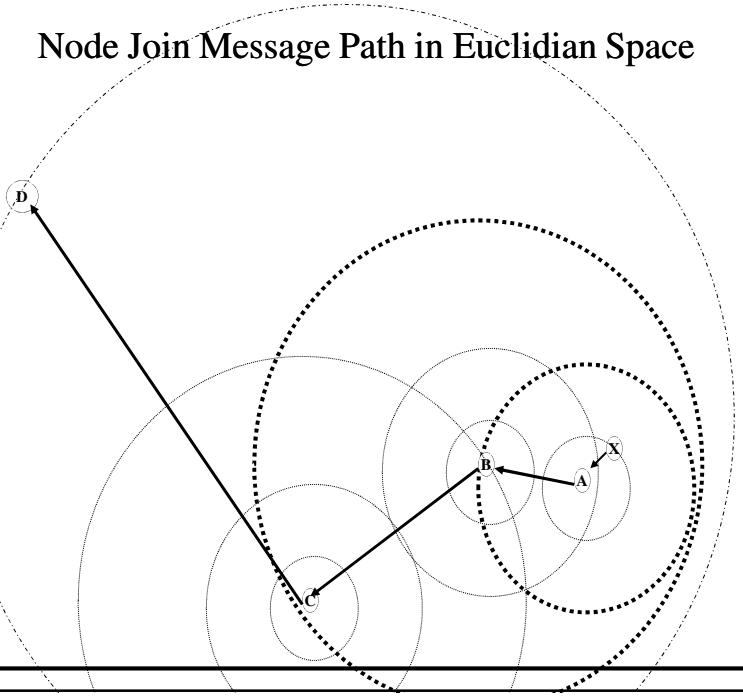
FOUNDATION OF PEER-TO-PEER SYSTEMS

- X borrows A's Neighborhood Set
- X's leaf set derived from Z's leaf set
- X_0 set to A_0
- X_1 set to B_1, X_2 set to C_2, ...

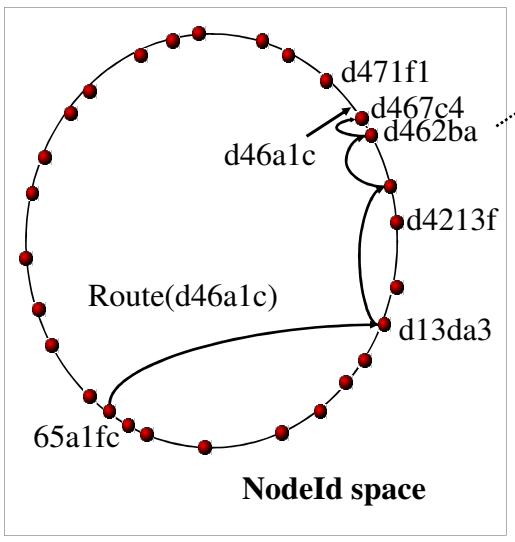


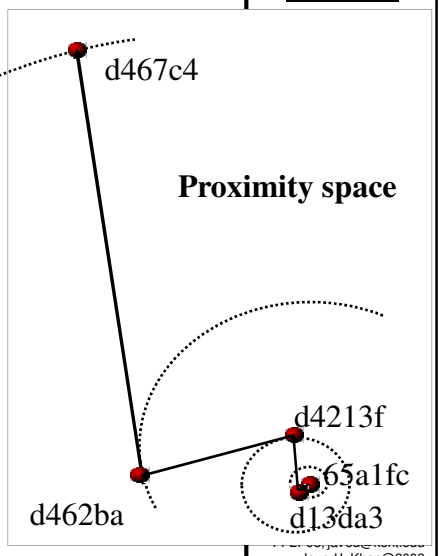
Node Join Message Path in Euclidian Space

FOUNDATION OF PEER-TO-PEER SYSTEMS



Routes in NodeID Space vs. Proximity Space





Q1: Preservation of Invariant (approximately)

- Let us consider row one of X's routing table, which is obtained from node B. The entries in this row are near B, however, it is not clear how close B is to X.
- Intuitively, it would appear that for X to take row one of its routing table from node B does not preserve the desired property, since the entries are close to B, but not necessarily to X. Right? Not exactly!
- In reality, the entries tend to be reasonably close to X. Recall that the entries in each successive row are chosen from an exponentially decreasing set size. Therefore, the expected distance from B to its row one entries (B1) is much larger than the expected distance traveled from node A to B. As a result, B1 is a reasonable choice for X1.
- This same argument applies for each successive level and routing steps.

Q1: Preservation of Invariant (Refinement)

- After X has initialized its state in this fashion, its routing table and neighborhood set approximate the desired locality property.
- However, the quality of this approximation must be improved to avoid cascading errors that could eventually lead to poor route locality.
- For this purpose, there is a second stage in which X requests the state from each of the nodes in its routing table and neighborhood set.
- It then compares the distance of corresponding entries found in those nodes' routing tables and neighborhood sets, respectively, and updates its own state with any closer nodes it finds.
- Also note, the neighborhood set contributes valuable information in this process, because it maintains and propagates information about nearby nodes regardless of their nodeId prefix.

Which Rows to Use?

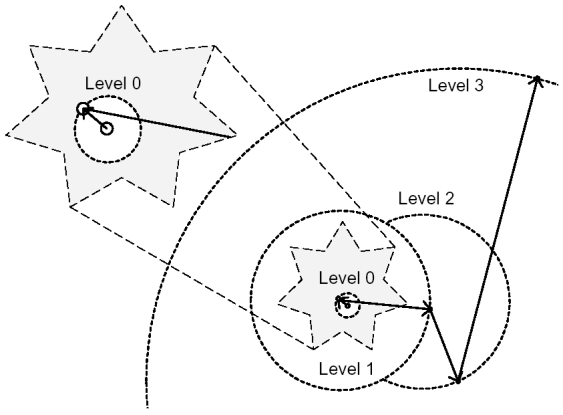
- [All nodes down the path has more and more prefix match with X. Thus, X can use not only i-th row but i, i-1, i-2, ..0 th rows from node at ith hop, nodes where will have at least one digit prefix map with X. –javed]
- Intuitively, why incorporating the state of nodes mentioned in the routing and neighborhood tables from stage one provides good representatives for X?
- The circles show the average distance of the entry from each node along the route, corresponding to the rows in the routing table. Observe that X lies within each circle, albeit off-center. In the second stage, X obtains the state from the entries discovered in stage one, which are located at an average distance equal to the perimeter of each respective circle.
- These states must include entries that are appropriate for X, but were not discovered by X in stage one, due to its off-center location.

Q2(a): Is a Route Optimum in Proximity Space?

- The entries in the routing table of each Pastry node are chosen to be close to the present node, according to the proximity metric, among all nodes with the desired nodeId prefix.
- As a result, in each routing step, a message is forwarded to a relatively close node with a nodeId that shares a longer common prefix or is numerically closer to the key than the local node.
- That is, each step moves the message closer to the destination in the nodeId space, while traveling the least possible distance in the proximity space.
- Since only local information is used, Pastry minimizes the distance of the next routing step with no sense of global direction. This procedure clearly does not guarantee that the shortest path from source to destination is chosen.
- However, it does give rise to relatively good routes.

Q2(a): Is a Route Good in Proximity Space?

- Fact#1: First, given a message was routed from node A to node B at distance d from A, the message cannot subsequently be routed to a node with a distance of less than d from A.
- Fact#2: Second, the expected distance traveled by a messages during each successive routing step is exponentially increasing.
- Implication: (No return to a past node) Jointly, these two facts imply that although it cannot be guaranteed that the distance of a message from its source increases monotonically at each step, a message tends to make larger and larger strides with no possibility of returning to a node within d_i of any node i encountered on the route, where d_i is the distance of the routing step taken away from node i. (diagram)
- Therefore, the message has nowhere to go but towards its destination.



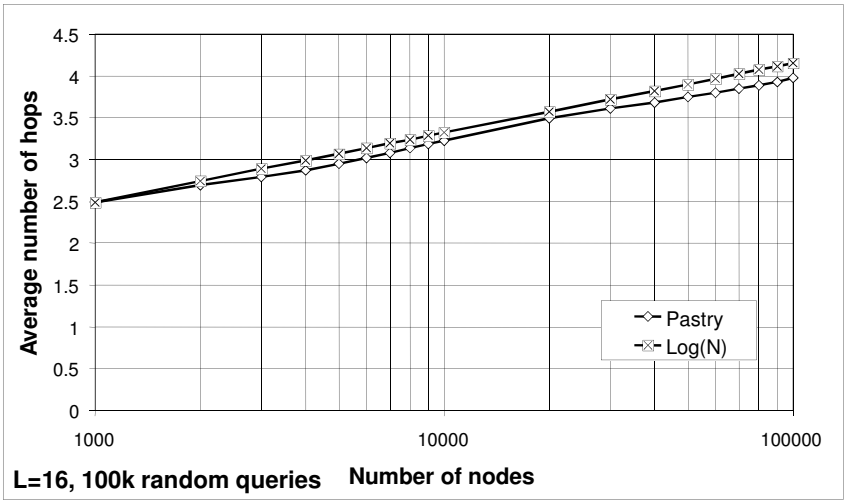
- Sample trajectory of a typical message in the Pastry network, based on experimental data. The message cannot re-enter the circles representing the distance of each of its routing steps away from intermediate nodes. Although the message may partly "turn back" during its initial steps, the exponentially increasing distances traveled in each step cause it to move toward its destination quickly.

Q2(b): Is the Route Complete?

- Pastry routes messages towards the node with the nodeId closest to the key, while attempting to travel the smallest possible distance in each step. Therefore, among the k numerically closest nodes to a key, a message tends to first reach a node near the client.
- But there are two approximations.
  - Firstly, Pastry makes only local routing decisions, minimizing the distance traveled on the next step with no sense of global direction.
  - Secondly, since Pastry routes primarily based on nodeId prefixes, it may miss nearby nodes with a different prefix than the key.
- Based on this estimation, a heuristic detects when a message approaches the set of k numerically closest nodes, and then it must switche to numerically nearest address based routing to locate the nearest replica (target).

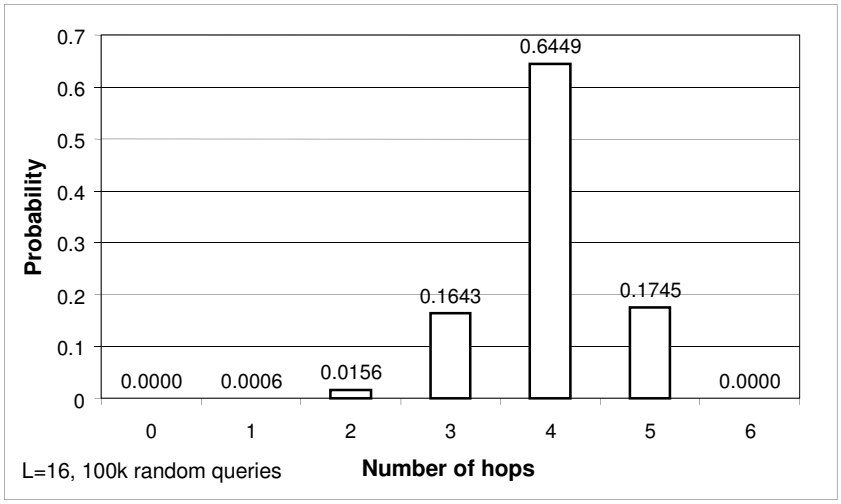
Pastry Performance

Pastry: Average # of Hops



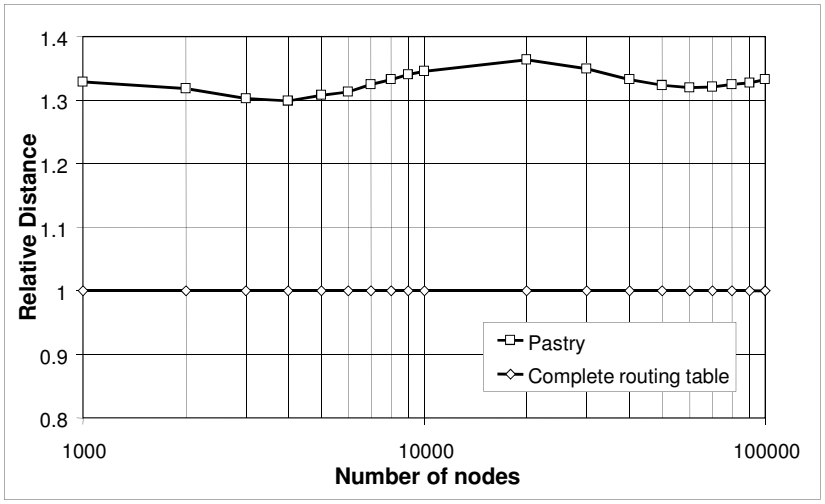
FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: # of Hops (100k nodes)



FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Distance traveled



b=4; |L|=16; |M|=32; 200,000 lookups; Random end points

L=16, 100k random queries, Euclidean proximity space

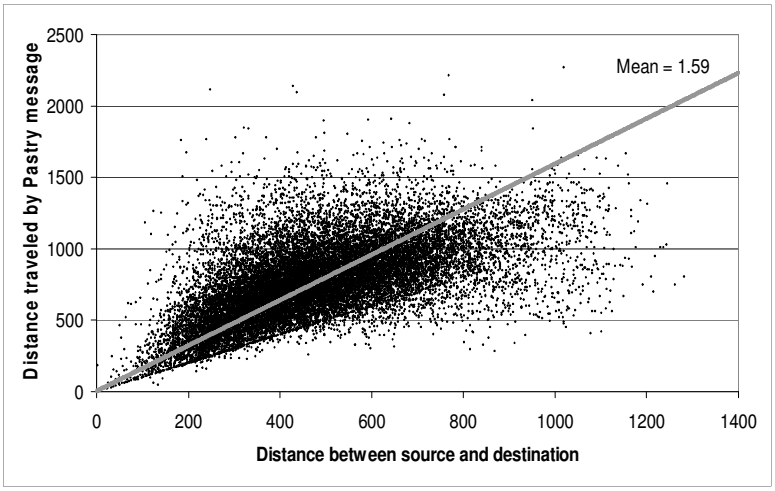
FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: Locality properties

- 1) Expected distance traveled by a message in the proximity space is within a small constant of the minimum.
- 2) Routes of messages sent by nearby nodes with same keys converge at a node near the source nodes.
- 3) Among k nodes with nodeIds closest to the key, message likely to reach the node closest to the source node first.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry Delay vs IP Delay



GATech top., .5M hosts, 60K nodes, 20K random messages

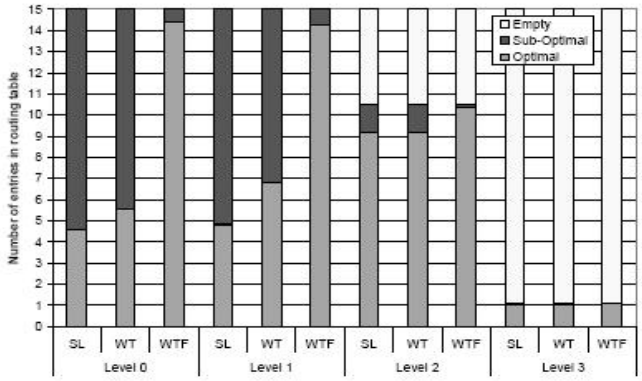
FOUNDATION OF PEER-TO-PEER SYSTEMS

Quality of Routing Entries

- Routing Effort
  - "SL" is a hypothetical method where the joining node considers only the appropriate row from each the route from itself to the node with the closest existing nodeId (see Section 2.4).
  - With "WT", the joining node fetches the entire state of each node along the path, but does not fetch state from the resulting entries. This is equivalent to omitting the second stage.
  - "WTF" is the actual method used in Pastry.
- Quality
  - Empty: Does a node get any IP for the prefix?
  - Optimum: Does the node get the closest node for that prefix?
  - Sub-Optimum: A node got a node- but which is not the best one.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Quality of Routing Tables



b=4; |L|=16; |M|=32; 5000 New Nodes

FOUNDATION OF PEER-TO-PEER SYSTEMS

Quiz: Can you compare the messaging complexity of the three schemes- SL, WT, & WTF?

Node Failure

- A 5000 node pastry network.
- 10% nodes fails silently.
- A key is chosen and routing is performed.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Impact of Node Failure

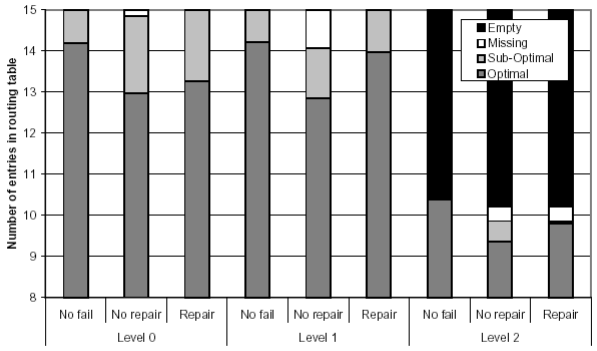
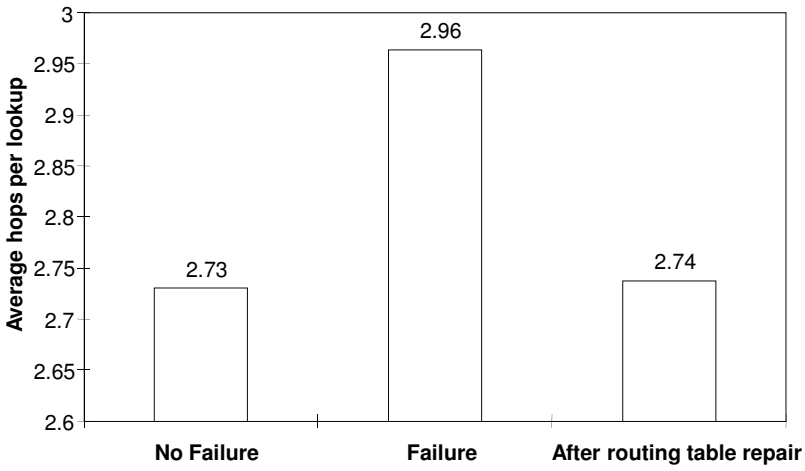


Fig. 9. Quality of routing tables before and after 500 node failures, $b = 4$, $|L| = 16$, $|M| = 32$ and 5,000 starting nodes.

FOUNDATION OF PEER-TO-PEER SYSTEMS

Pastry: # Routing Hops (failures)



L=16, 100k random queries, 5k nodes, 500 failures

FOUNDATION OF PEER-TO-PEER SYSTEMS

Next Class:
Presentations