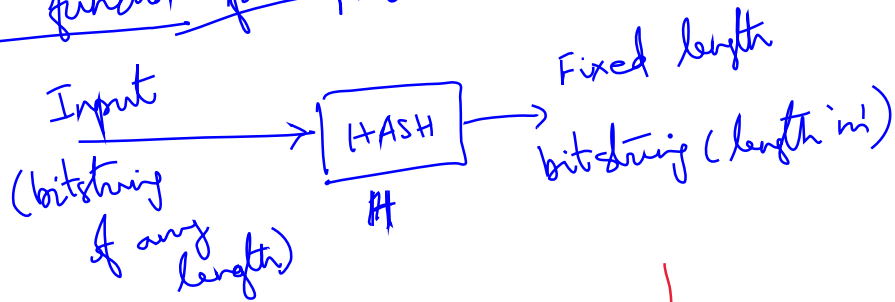
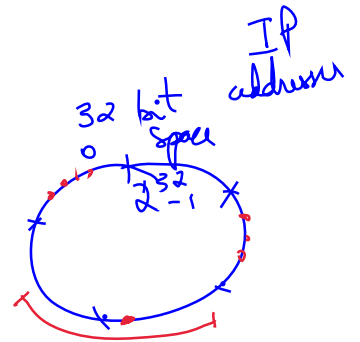


Use Hash function for mapping



$$f_1 \rightarrow [H] \rightarrow H(f_1)$$

$$f_2 \rightarrow [H] \rightarrow H(f_2)$$



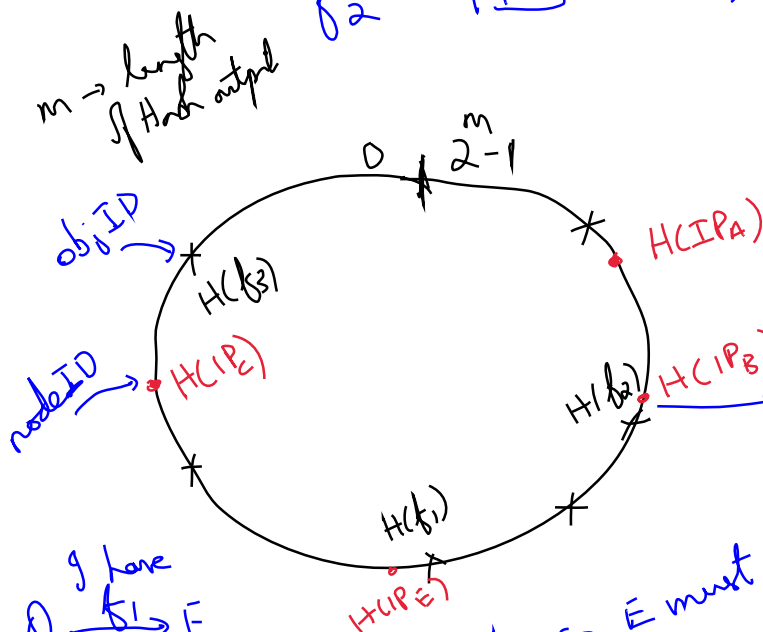
files: objects

peers: nodes

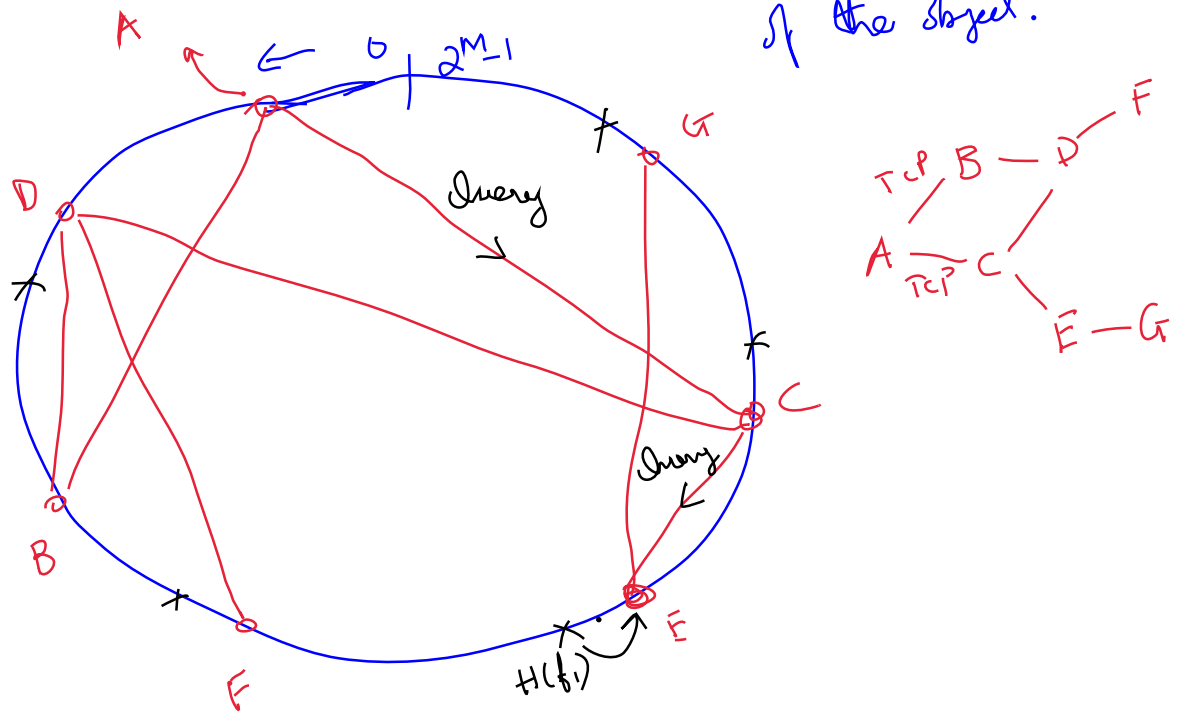
$m \rightarrow$  length  
of Hash output

obj ID

node ID



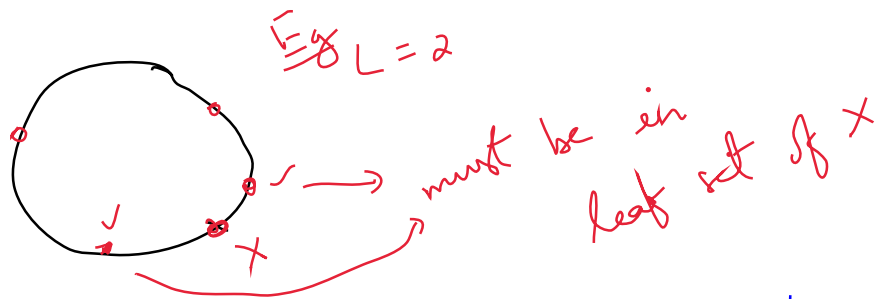
Want to store info. about where an object is stored  
at node whose nodeID is closest to objID  
of the object.



Idea: Route <sup>query</sup> messages closer to  $H(f_1)$  in this  
virtual space till we reach "E"

- Suppose 'A' wants to find out where  $f_1$  is stored
- A finds <sup>which of</sup> its connected peers is closest to E  
leaf set of A
- This repeats till the query reaches a node  
who has no connected peer closer to  $H(f_1)$

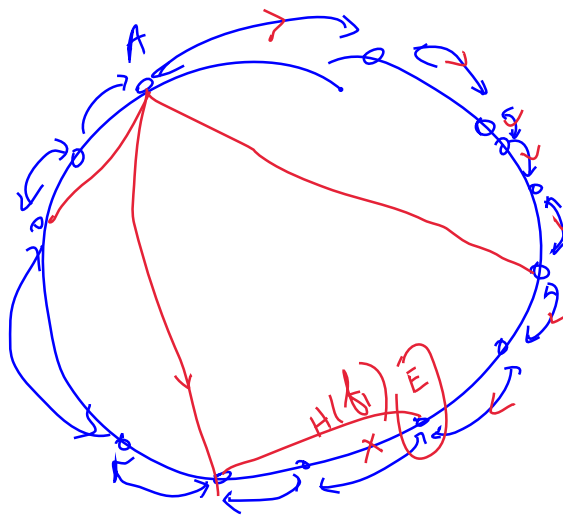
PASTRY: In leaf node of node X, include  $1/2$  closest  
nodes on either side of X in the  
virtual space



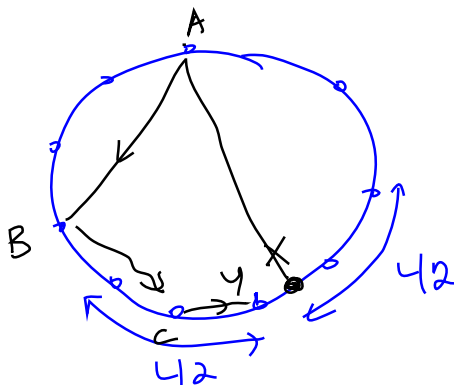
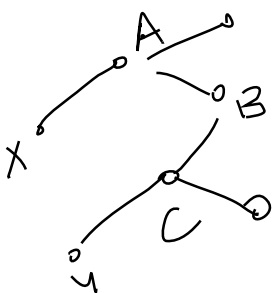
This ensures that the "routing" protocol gets to node closest to  $H(b_i)$

PASTRY ensures that leaf set of any node is of size  $L + O(\log N)$  #peers in network

---



Suppose P2P network already has the property that leaf set of all nodes contains  $4/2$  closest nodes on either side of the node



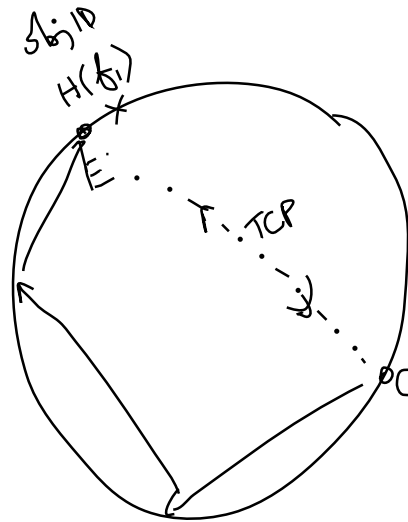
CHORD, TAPESTRY, PASTRY

- Peterson
- Druschel

( $L=4$ )

New node  $X$  routes a message to find node whose nodeID is closest to own ID

from  $Y$ ,  $X$  determines  $Y/2$  nodes ~~who~~ who are closest to it on left & right



• Query has  $Q$ 's IP address

•  $E$  contacts  $Q$

$Q \rightarrow$  has  $f_1$ .  $Q$  tells  $E$  that it has  $f_1$

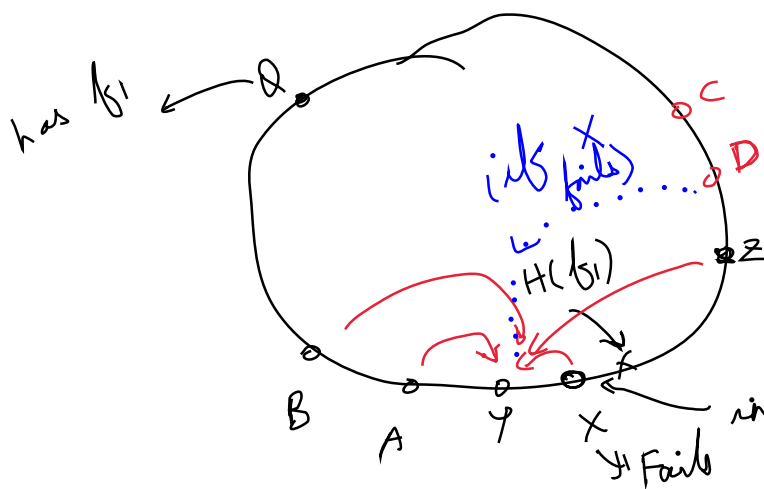
Similar procedure to find where  $f_1$  is stored

DHT Distributed Hash Table

$X$  can add  $A, B, C, Y$  to its leaf set

$O(\log n) \rightarrow$  this ensures leaf set is of size  $(O(\log N) + L)$

what if a node fails?



Y stores info which is present in 2 nodes  
on either side

All peers send Keep Alive messages to peers  
they are connected to (leaf set)

- if any node 2 nodes to right or left  
fails, then it finds out about  
nodes further away and ensures