CS 425/ECE 428/CSE 424 Distributed Systems (Fall 2009)

Lecture 14

Peer-to-Peer systems (Part III) and

Midterm Review Session

Acknowledgement

- The slides during this semester are based on ideas and material from the following sources:
  - Slides prepared by Professors M. Harandi, J. Hou, I. Gupta, N. Vaidya, Y-Ch. Hu, S. Mitra.
  - Slides from Professor S. Gosh's course at University o Iowa.

Administrative

- **MP1 scores posted**
- **HW2 solutions posted**

Administrative

- **MP2** posted **October 5, 2009**, on the course website,
  - **Deadline November 6 (Friday)**
  - **Demonstrations** , 4-6pm, 11/6/2009
  - You will need to **lease** one Android/Google Developers Phone per person from the CS department (see lease instructions)!!
  - **Start early** on this MP2
  - **Update groups** as soon as possible and let TA know by email so that she can work with TSG to update group svn
  - **Tutorial for MP2** planned for **October 28** evening if students send questions to TA by **October 25**. Send requests what you would like to hear in the tutorial.
  - During October 15-25, Thadpong Pongthawornkamol (tpongth2@illinois.edu) will held office hours and respond to MP2 questions for Ying Huang (Ying is going to the IEEE MASS 2009 conference in China)

Administrative

- **MP3 proposal instructions**
  - You will need to submit a proposal for MP3 on top of your MP2 before you start MP3 on November 9, 2009
  - Exact Instructions for MP3 proposal format will be posted **October 9, 2009**
  - Deadline for MP3 proposal: **October 25, 2009, email proposal to TA**
  - At least one representative of each group meets with **instructor or TA during October 26-28** during their office hours) watch for extended office hours during these days.
    - **Instructor office hours: October 28 times 8:30-10am**

Administrative

- To get Google Developers Phone, you need a **Lease Form**
  - You must pick up a **lease form** from the instructor during **October 6-9** (either in the class or during office hours) since the lease form must be signed by the instructor.
  - Fill out the lease form; bring the lease form to **Rick van Hook/Paula Welch** and pick up the phone from **1330 SC**
- **Lease Phones**: phones will be ready to pick up starting **October 20, 9-4pm** from room 1330 SC (purchasing , receiving and inventory control office)
- **Return Phones**: phones need to be returned during **December 14-18**, **9-4pm** in 1330 SC

Plan for Today

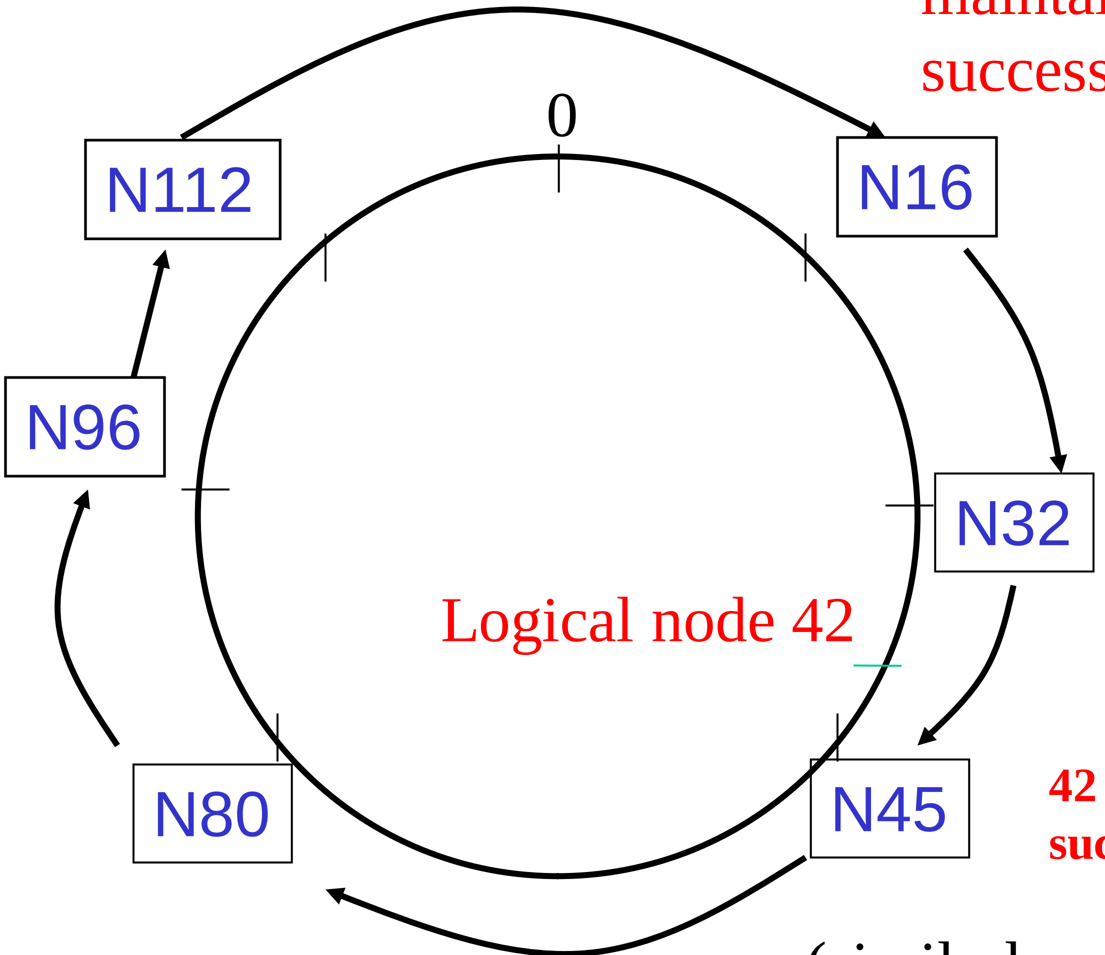
- Chord
- Review material for midterm

Peer pointers (1): *successors*

Say *m=7*

SHA-1(140.45.3.12, 1245) = 42 (0101010)

Each physical node maintains a successor pointer



42 stored in the successor N45

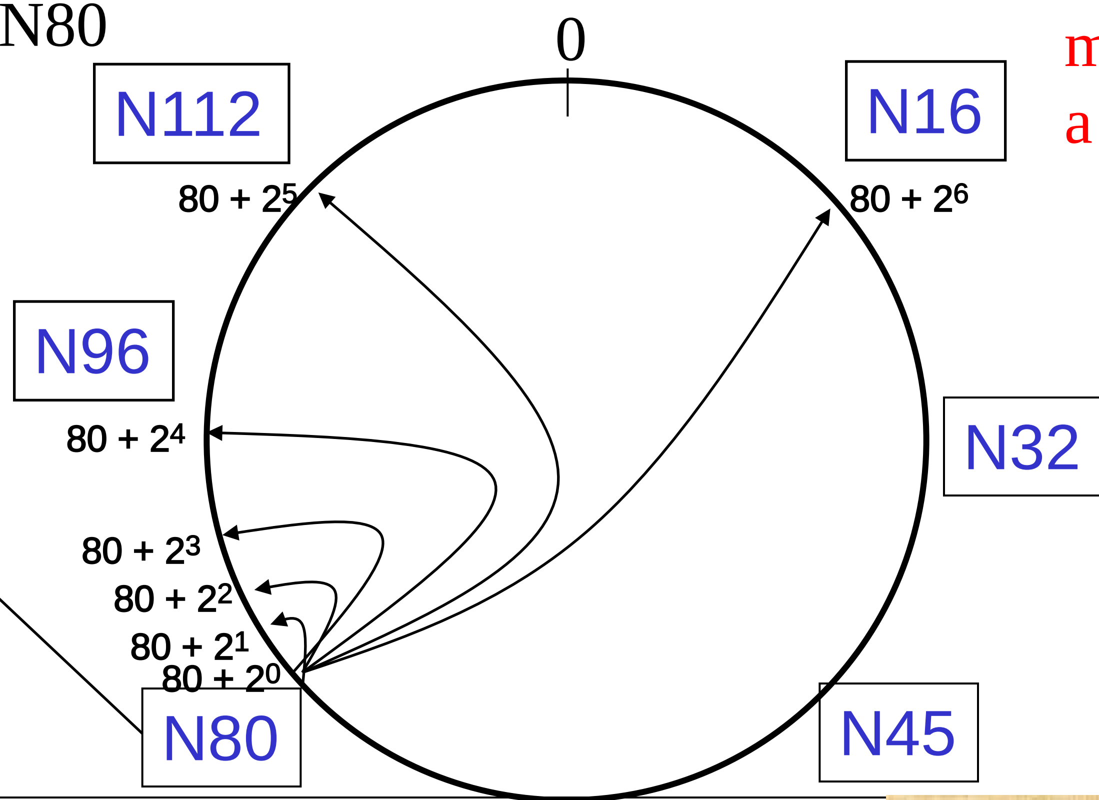
(similarly predecessors)

Peer pointers (2): *finger tables*

(Scalable Key Location)

Finger Table at N80

i	*ft[i]*
0	96
1	96
2	96
3	96
4	96
5	112
6	16

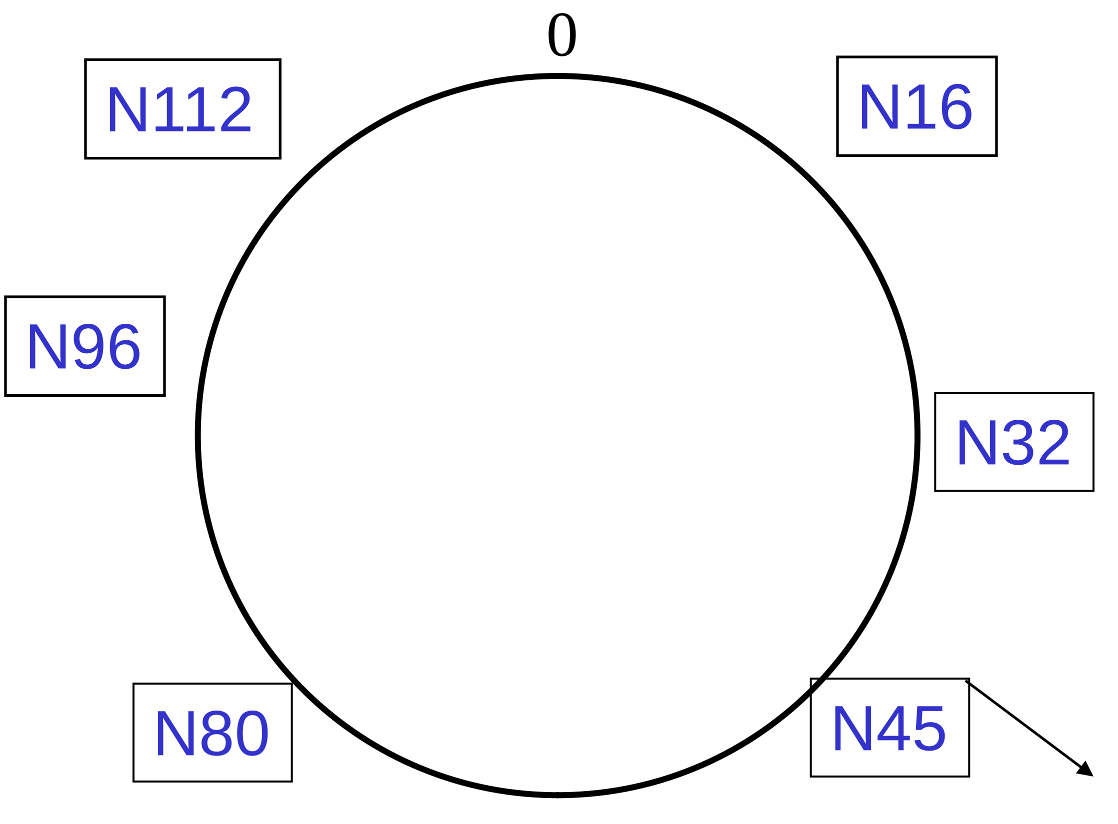


Each node maintains a finger table

*i*th entry at peer with id *n* is first peer with id >= $n + 2^i \pmod{2^m}$

Mapping Files

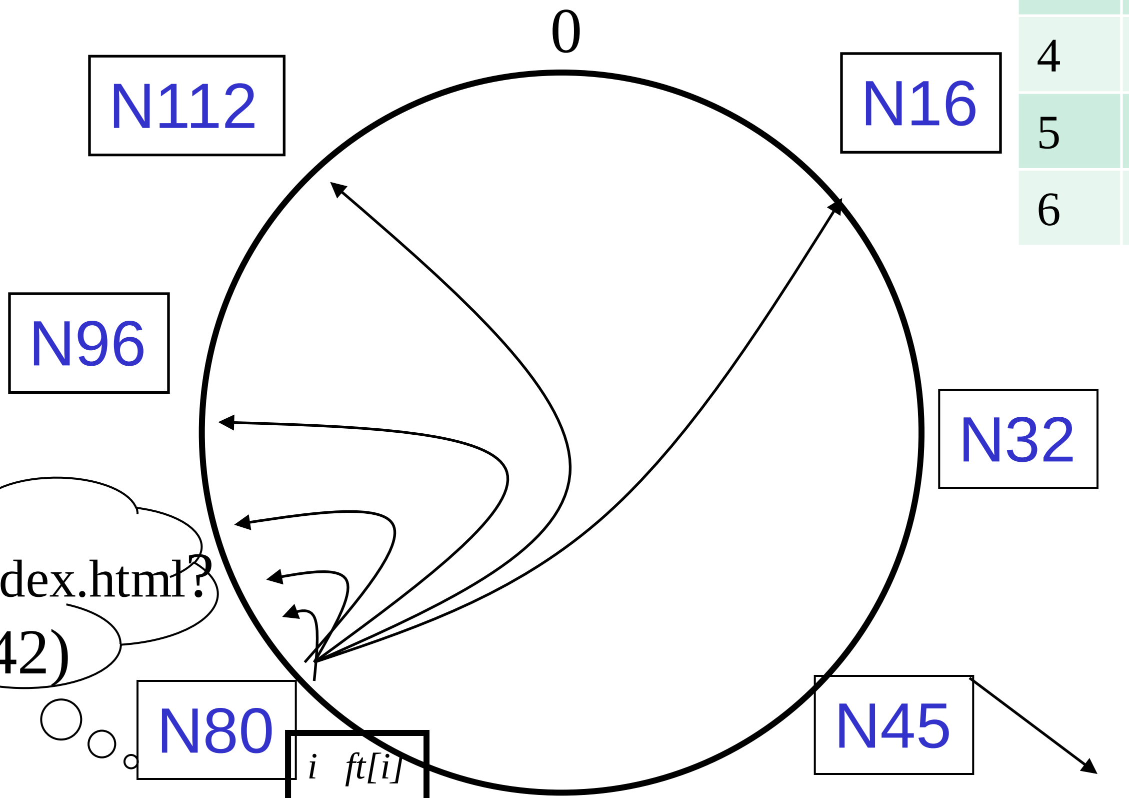
Say *m=7*

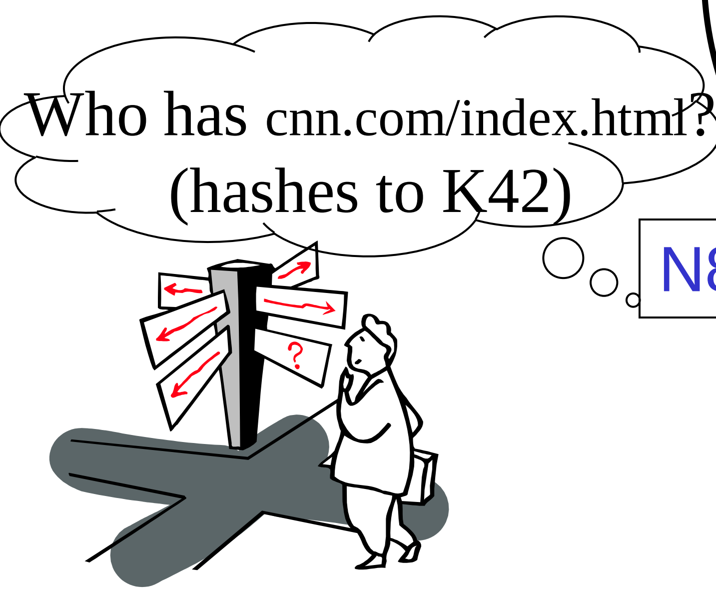


File cnn.com/index.html with key K42 stored here

Search

Say $m=7$





i	*ft[i]*
0	96
1	96
2	96
3	96
4	96
5	112
6	16

0	**32**
1	32
2	32
3	32
4	32
5	80
6	80

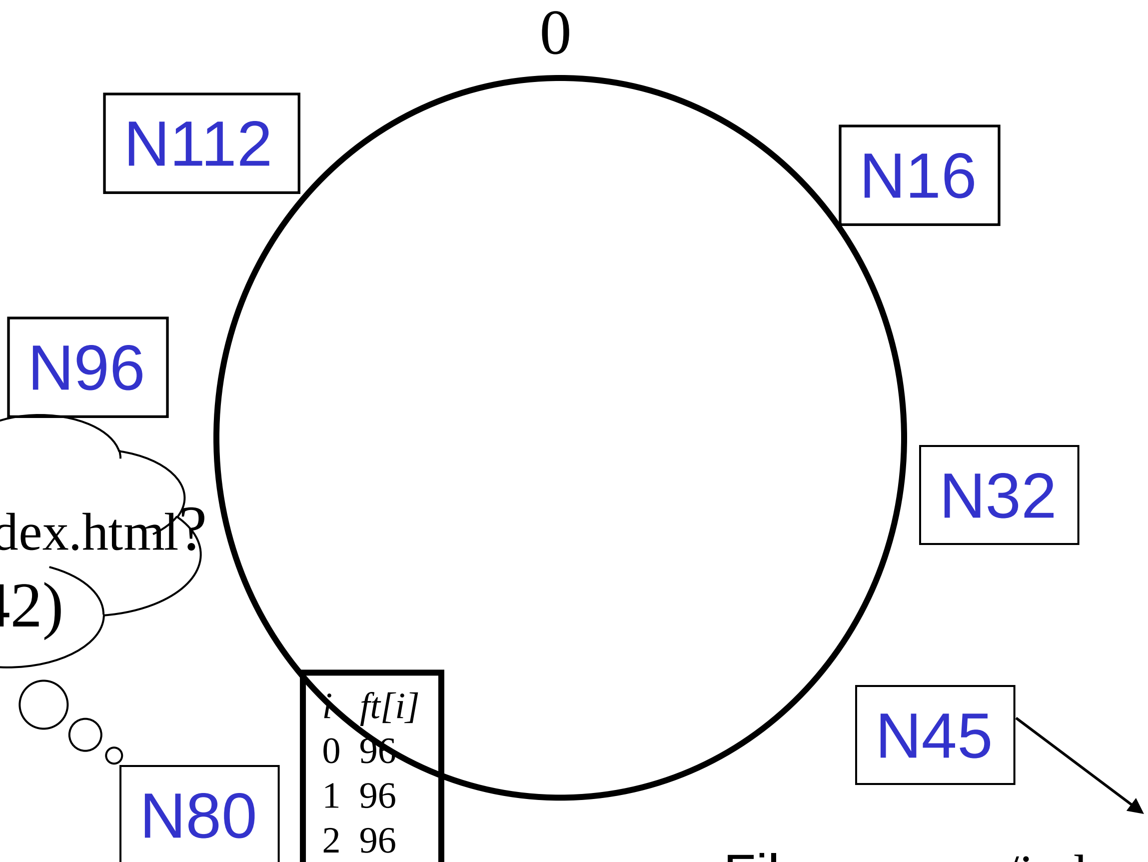
0	**45**
1	45
2	45
3	45
4	80
5	80
6	96

File cnn.com/index.html with key K42 stored here

Search

At node *n*, send query for key *k* to largest successor/finger entry < *k* (all modulo *m*) if none exist, send query to *successor(n)*

Say *m=7*



i	*ft[i]*
0	96
1	96
2	96
3	96
4	96
5	112
6	16

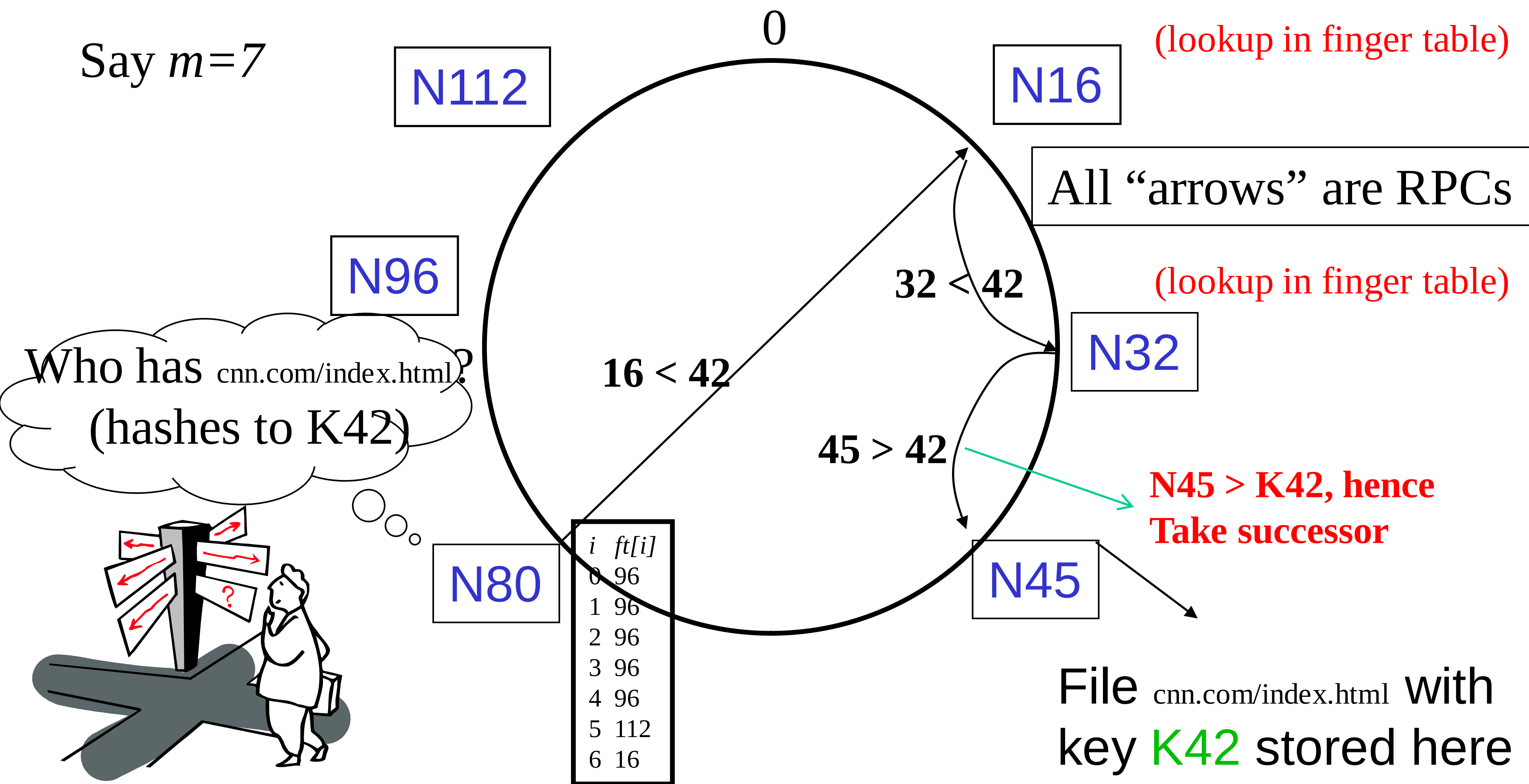
Who has cnn.com/index.html?
(hashes to K42)

File cnn.com/index.html with key K42 stored here

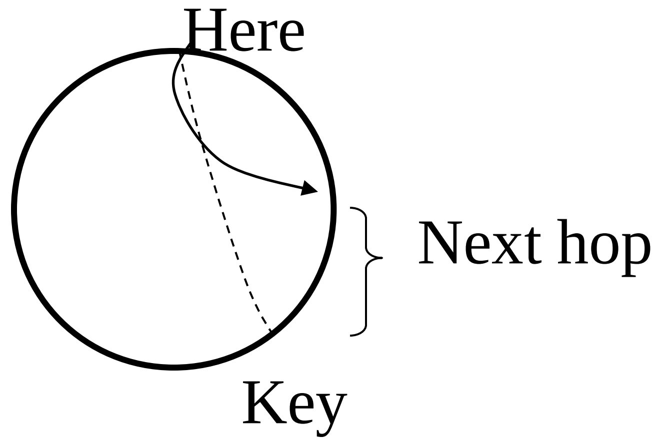
Search

At node *n*, send query for key *k* to largest successor/finger entry < *k* (all mod *m*)
if none exist, send query to *successor(n)*

Say *m=7*



Analysis



Search takes *O(log(N))* time

Proof

– (intuition): *at each step, distance between query and peer-with-file reduces by a factor of at least 2* (why?)

Takes at most *m* steps: 2^m is at most a constant multiplicative factor above *N*, lookup is *O(log(N))*

– (intuition): after *log(N)* forwardings, distance to key is at most $2^m / N$ (why?)

- Number of node identifiers in a range of $2^m / N$ is *O(log(N))* with high probability
- So using *successors* in that range will be ok

Analysis (contd.)

- *O(log(N))* search time holds for file insertions too (in general for *routing to any key*)
  - "Routing" can thus be used as a building block for other applications than file sharing [can you name one?]
- *O(log(N))* time true only if finger and successor entries **correct**
- When might these entries be wrong?

Analysis (contd.)

- *O(log(N))* search time holds for file insertions too (in general for *routing to any key*)
  - "Routing" can thus be used as a building block for other applications than file sharing [can you name one?]
- *O(log(N))* time true only if finger and successor entries **correct**
- When might these entries be wrong?
  - When you have failures

Stabilization Protocol

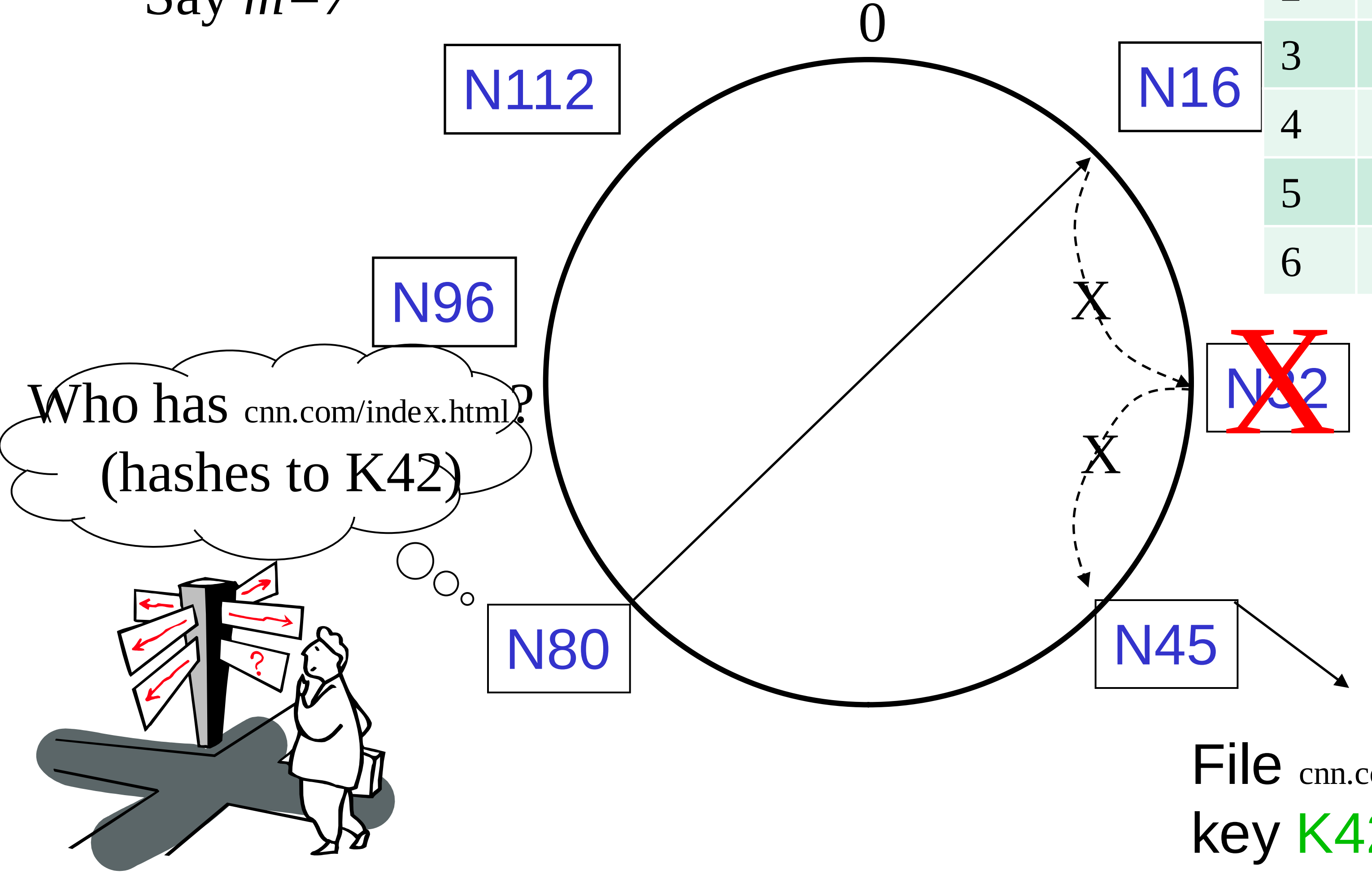
- Maintaining finger tables only is expensive in case of dynamic joint and leave nodes
- Chord therefore separates correctness from performance goals via stabilization protocols
- Basic stabilization protocol
  - Keep successor's pointers correct!
  - Then use them to correct finger tables

Search under peer failures

Lookup fails
(N16 does not know N45)

0	32
1	32
2	32
3	32
4	32
5	80
6	80

Say $m=7$

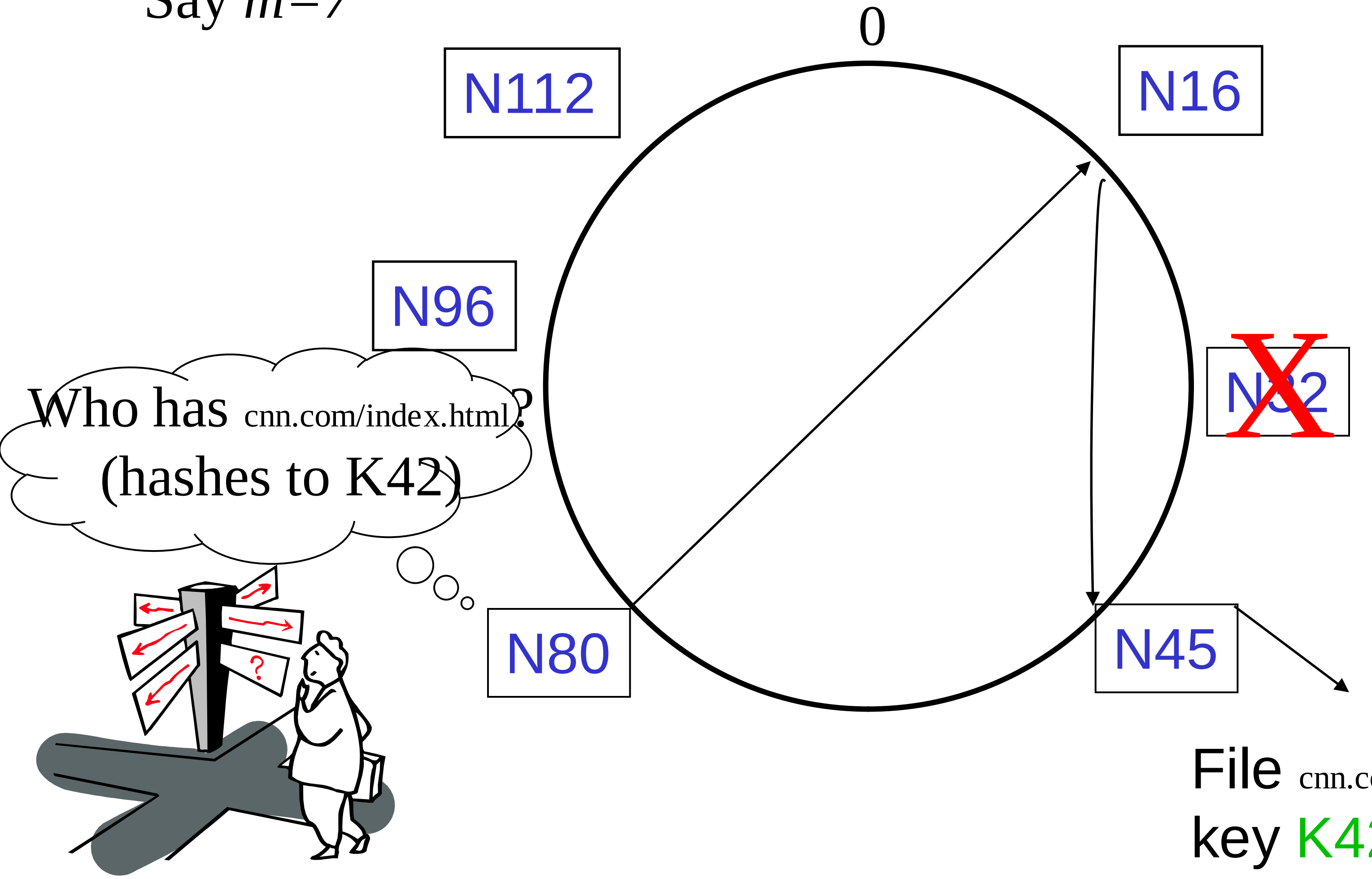


File cnn.com/index.html with key K42 stored here

Search under peer failures

One solution: maintain *r* multiple *successor* entries
in case of failure, use successor entries

Say *m=7*



File cnn.com/index.html with key K42 stored here

Search under peer failures

- Let *r* be the successor list length
- Choosing *r=2log(N)* suffices to maintain correctness with high probability
  - Say 50% of nodes fail
  - Pr(for given node, at least one successor alive)=

$$1-(\frac{1}{2})^r = 1-(\frac{1}{2})^{2\log N} = 1-\frac{1}{N^2}$$

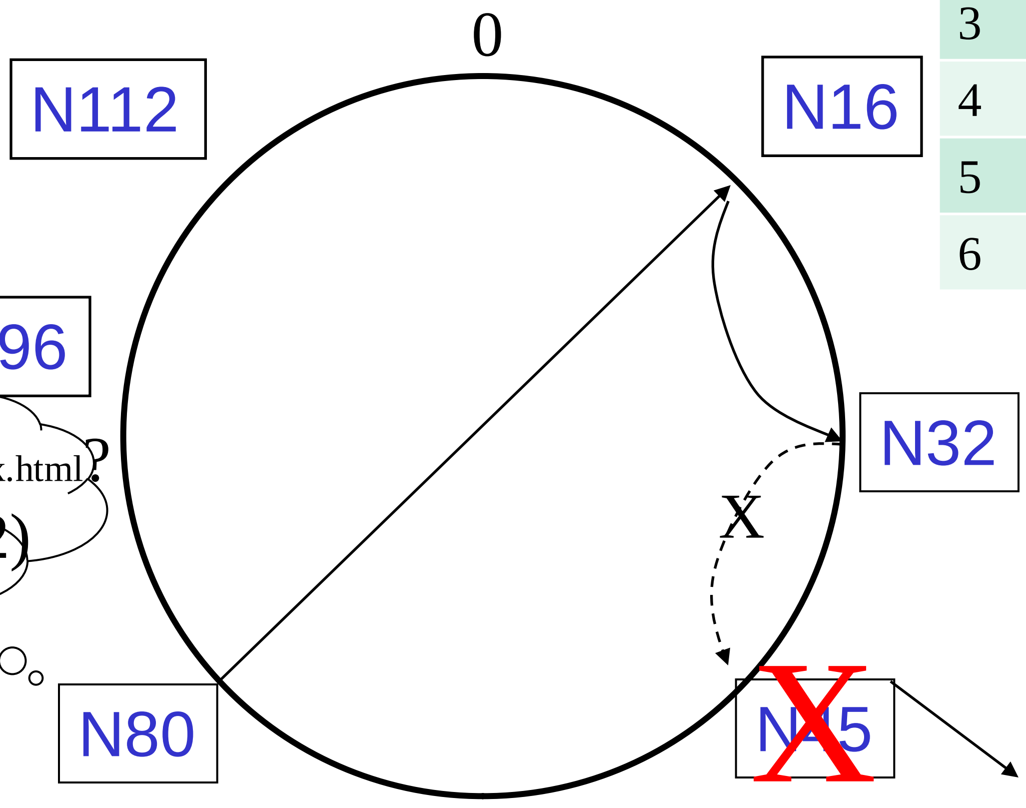
  - Pr(above is true for all alive nodes)=

$$(1-\frac{1}{N^2})^{N/2} = e^{-\frac{1}{2N}} \approx 1$$

Search under peer failures (2)

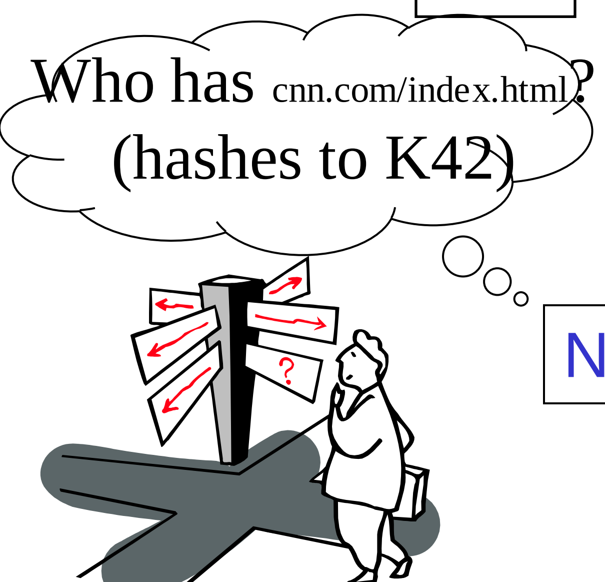
Lookup fails
(N45 is dead)

Say $m=7$



0	32
1	32
2	32
3	32
4	32
5	80
6	80

0	45
1	45
2	45
3	45
4	80
5	96
6	0

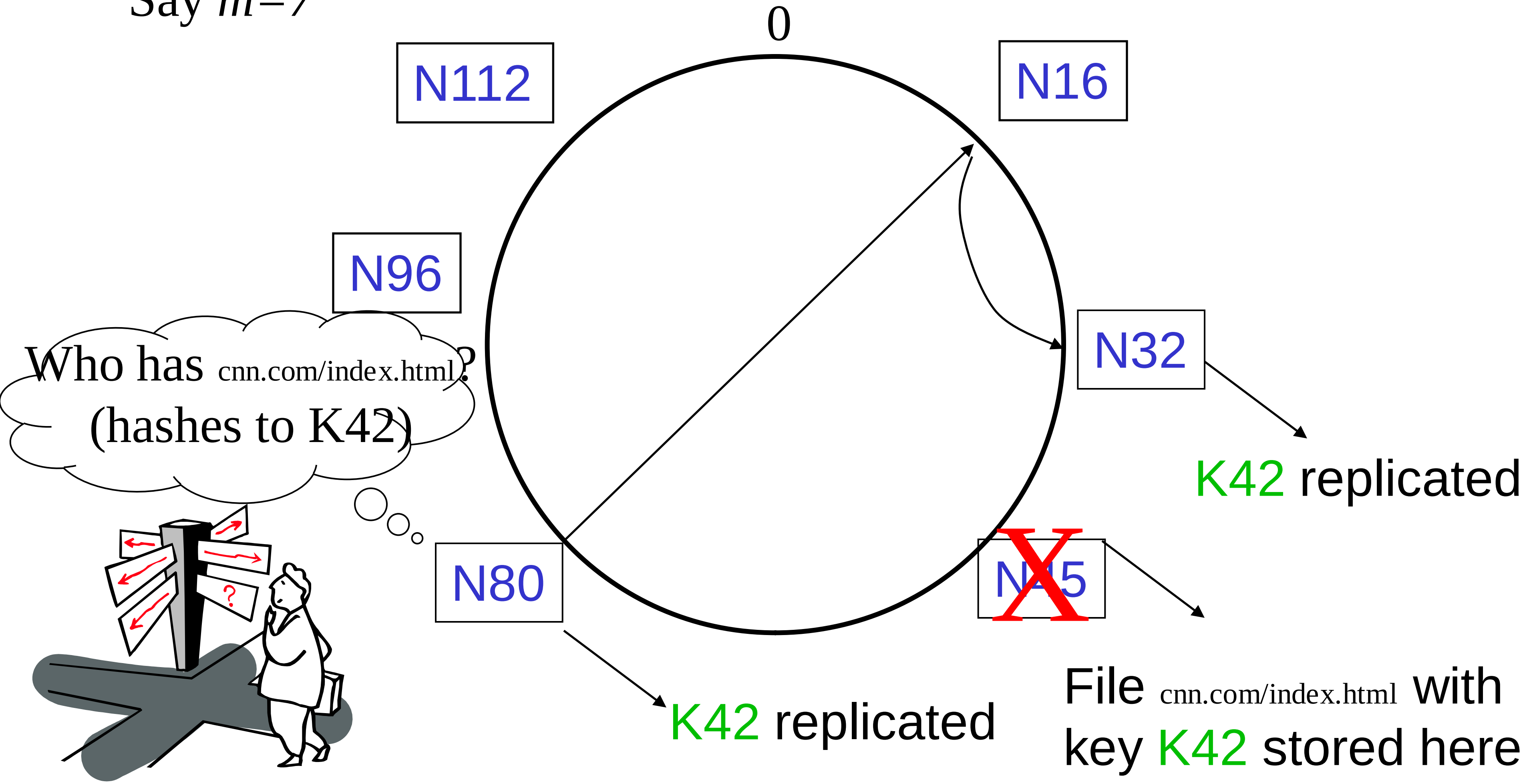


File cnn.com/index.html with key K42 stored here

Search under peer failures (2)

One solution: replicate file/key at *r* successors and predecessors

Say *m=7*



Need to deal with dynamic changes

- ✓ Peers fail
- New peers join
- Peers leave

All the time

→ Need to update *successors* and *fingers*, and ensure keys reside in the right places

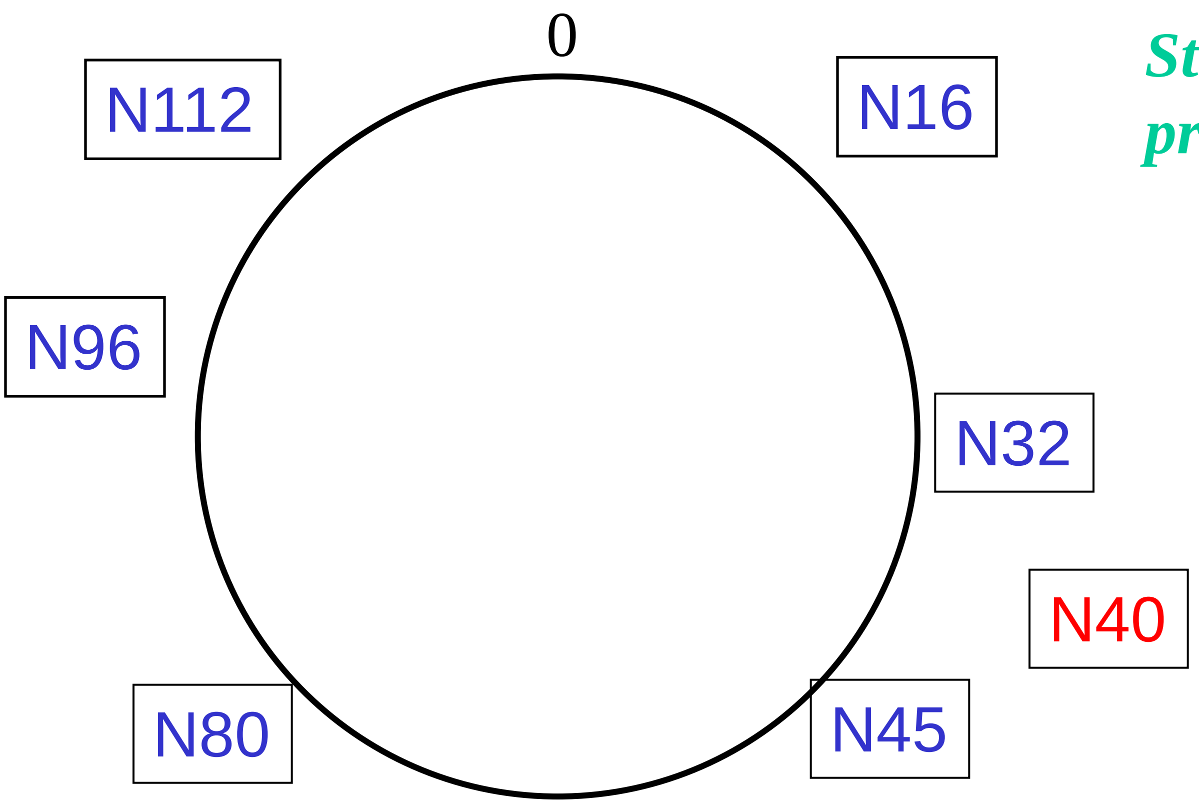
New peers joining

1. N40 acquires that N45 is its successor
2. N45 updates its info about predecessor to be N40
3. N32 runs stabilizer and asks N45 for predecessor
4. N45 returns N40
5. N32 updates its info about successor to be N40
6. N32 notifies N40 to be its predecessor
N40 periodically talks to neighbors to update own finger table

Peers also keep info about their predecessors to deal with dynamics

Stabilization protocol

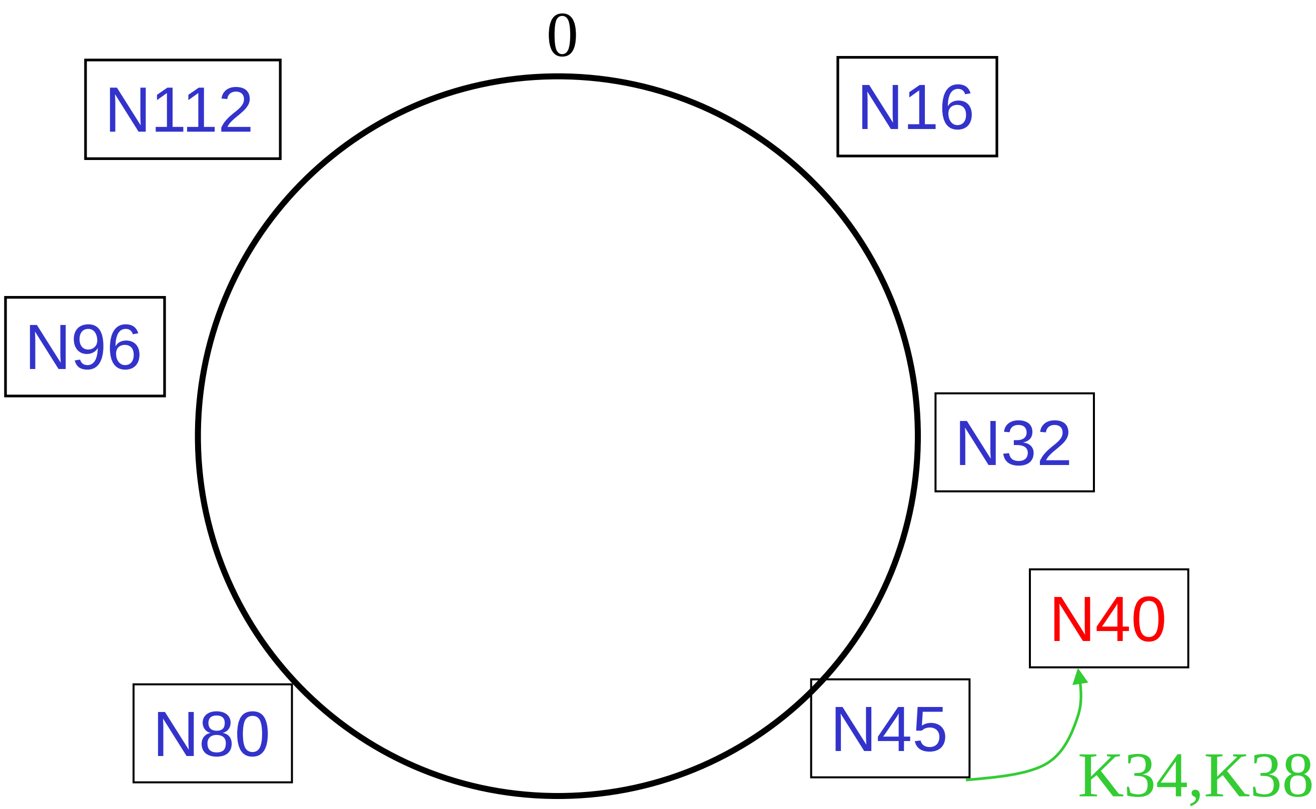
Say *m=7*



New peers joining (2)

N40 may need to copy some files/keys from N45
(files with fileid between 32 and 40)

Say *m=7*



New peers joining (3)

- A new peer affects $O(log(N))$ other finger entries in the system
- Consider the number of messages to re-establish the Chord routing invariants and finger tables
- Number of messages per peer join= $O(log(N)*log(N))$
- Similar set of operations for dealing with peers leaving

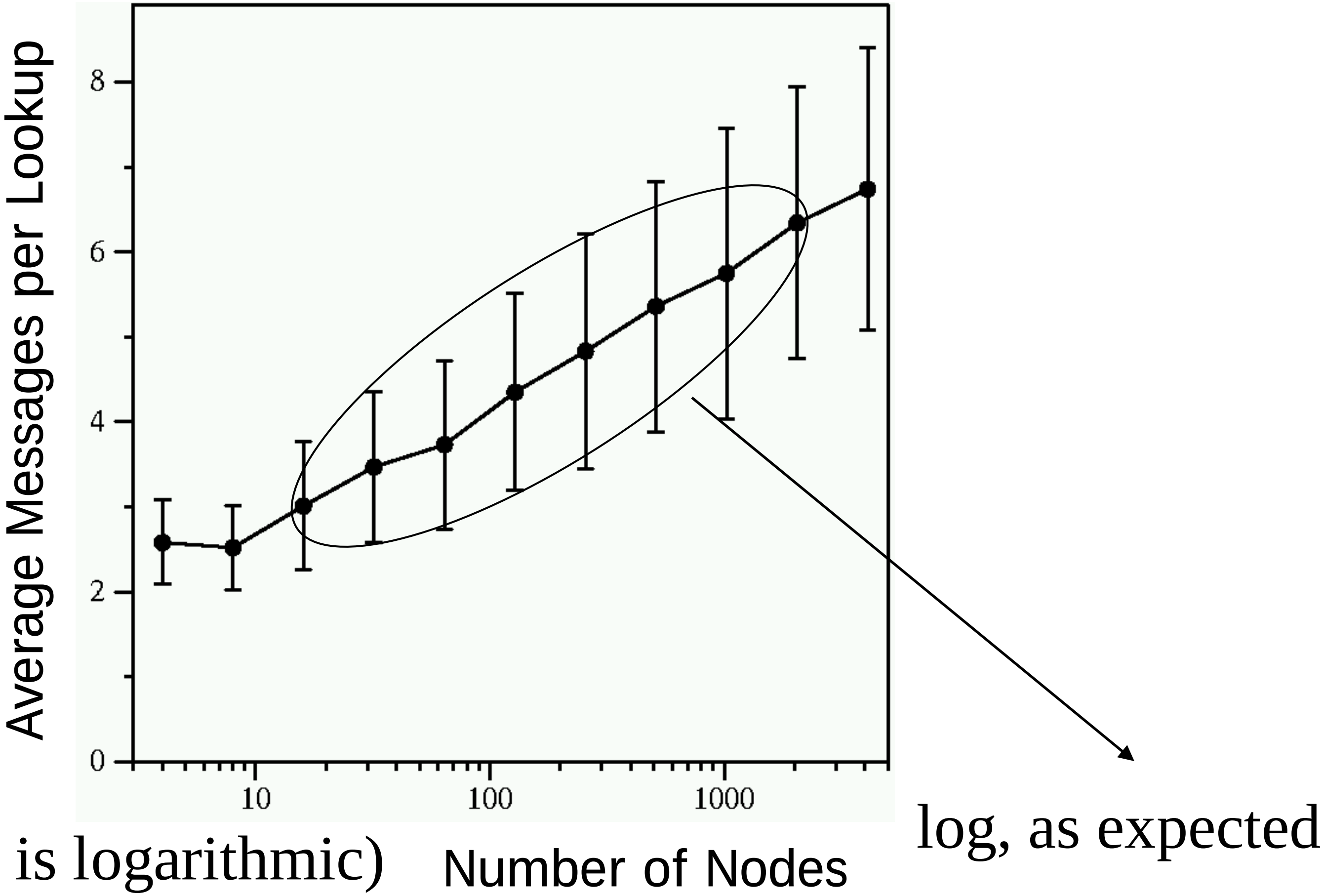
Stabilization Protocol

- Concurrent peer joins, leaves, failures might cause loopiness of pointers, and failure of lookups
  - Chord peers periodically run a *stabilization* algorithm that checks and updates pointers and keys
  - Ensures non-loopiness of fingers, eventual success of lookups and *O(log(N))* lookups
  - [TechReport on Chord webpage] defines weak and strong stability
  - Each stabilization round at a peer involves a constant number of messages
  - Strong stability takes $O(N^2)$ stabilization rounds (!)

Experimental Results

- Sigcomm 01 paper had results from simulation of a C++ prototype
- SOSP 01 paper had more results from a 12-node Internet testbed deployment
- We’ll touch briefly on the first set of results
- 10,000 peer system

Lookups



(X-axis is logarithmic)

Discussion

- Memory: $O(log(N))$ successor pointer, *m* finger entries
- Indirection: store a pointer instead of the actual file
- Does not handle **partitions** of the group (can you suggest a possible solution?)

Discussion (2)

- When nodes are constantly joining, leaving, failing
  - Significant effect to consider: traces from the Overnet system show *hourly* peer turnover rates (***churn***) could be *10-15%* of total number of nodes in system
  - Leads to excessive (unnecessary) key copying
    - further, remember that keys are replicated, so all these copies will need to be copied around as nodes join and leave
  - Stabilization algorithm may need to consume more bandwidth to keep up
  - There exist alternative DHTs that are churn-resistant

Discussion (3)

- Current status of project:
  - Protocol constantly undergoing change
  - File systems (CFS, Ivy) built on top of Chord
  - DNS lookup service built on top of Chord
  - Spawned research on many interesting issues about p2p systems

http://www.pdos.lcs.mit.edu/chord/

Summary

- Chord protocol
  - *Structured P2P*
  - *O(log(N))* memory and lookup cost
  - Simple lookup algorithm, rest of protocol complicated
  - Stabilization works, but how far can it go?

Review for Midterm

- **Midterm on October 13 (Tuesday)**
  - Exam will take place in class
  - You are allowed one cheat-sheet (one side only)
  - Exam will include all topics covered in HW1-HW2, plus P2P material (Lectures 1-13)
  - Closed book exam
  - **Instructor will hold office hours on Monday, October 12, 3-4pm in 3104 SC**

Midterm Topics

- What is distributed system
- Why distributed systems
- Design goals of distributed systems
- Time and Synchronization
  - Chapter 11.1-11.4
  - Physical clock/Time: skew, synchronization of physical clocks, internal and external synchronization , NTP
  - Logical clocks: happens before relation, Lamport clock, Vector logical clock

Midterm Topics

- Global state and global snapshots
  - Chapter 11.5
  - History of a process and cut, consistent cut, linearization, properties of a predicate: stable, safety, liveness; Chandy-Lamport Snapshot algorithm
- Multicast
  - Chapter 12.4
  - Communication models, B-multicast, reliable multicast, properties: integrity, validity, agreement; ordered multicast; Total ordering, FIFO ordering, causal ordering, FIFO-ordered multicast, causal multicast, total-ordered multicast

Midterm Topics

- Group Communication
  - Chapter 15.2.2
  - Group view, process view, view synchrony
- Distributed Mutual Exclusion
  - Chapter 12.2
  - Coordinator-based algorithm, token ring algorithm, Ricart&Agrawala, Maekawa algorithms, Raymond's Token-based algorithm
  - Make sure you understand analysis of these algorithms in terms of message overhead, bandwidth, client delay, synchronization delay

Midterm Topics

- Leader election
  - Chapter 12.3
  - Ring-based algorithm, modified ring-based algorithm, bully algorithm
- Consensus
  - Chapter 12.5
  - Failure models, consensus problem, byzantine general problem, interactive consistency problem, consensus in synchronous systems, byzantine generals in synchronous systems, impossibility of consensus in asynchronous systems

Midterm Topics

- Failure detectors
  - Chapter 12.1, 2.3
  - Heart-beating failure detector, ping-ack algorithm, distributed failure detection through heart-beating algorithms – centralized, ring-based, all-to-all; accuracy metrics, other types of failures
- P2P systems
  - Basic characteristics of P2P, Napster, Gnutella, Fast-track unstructured P2P systems