

Lecture 33 – The Dark Web

Ryan Cunningham

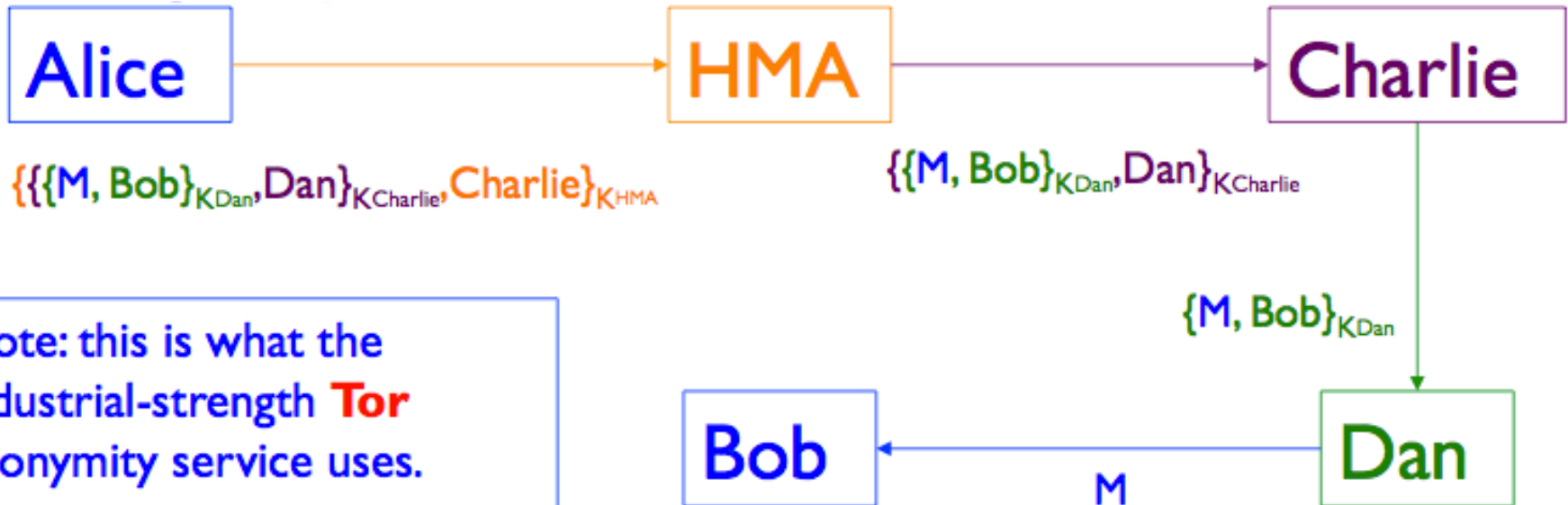
University of Illinois

ECE 422/CS 461 – Fall 2017

Security News

- Amazon Key camera can be remotely disabled
- 17 year old RCE in Microsoft Equation Editor patched

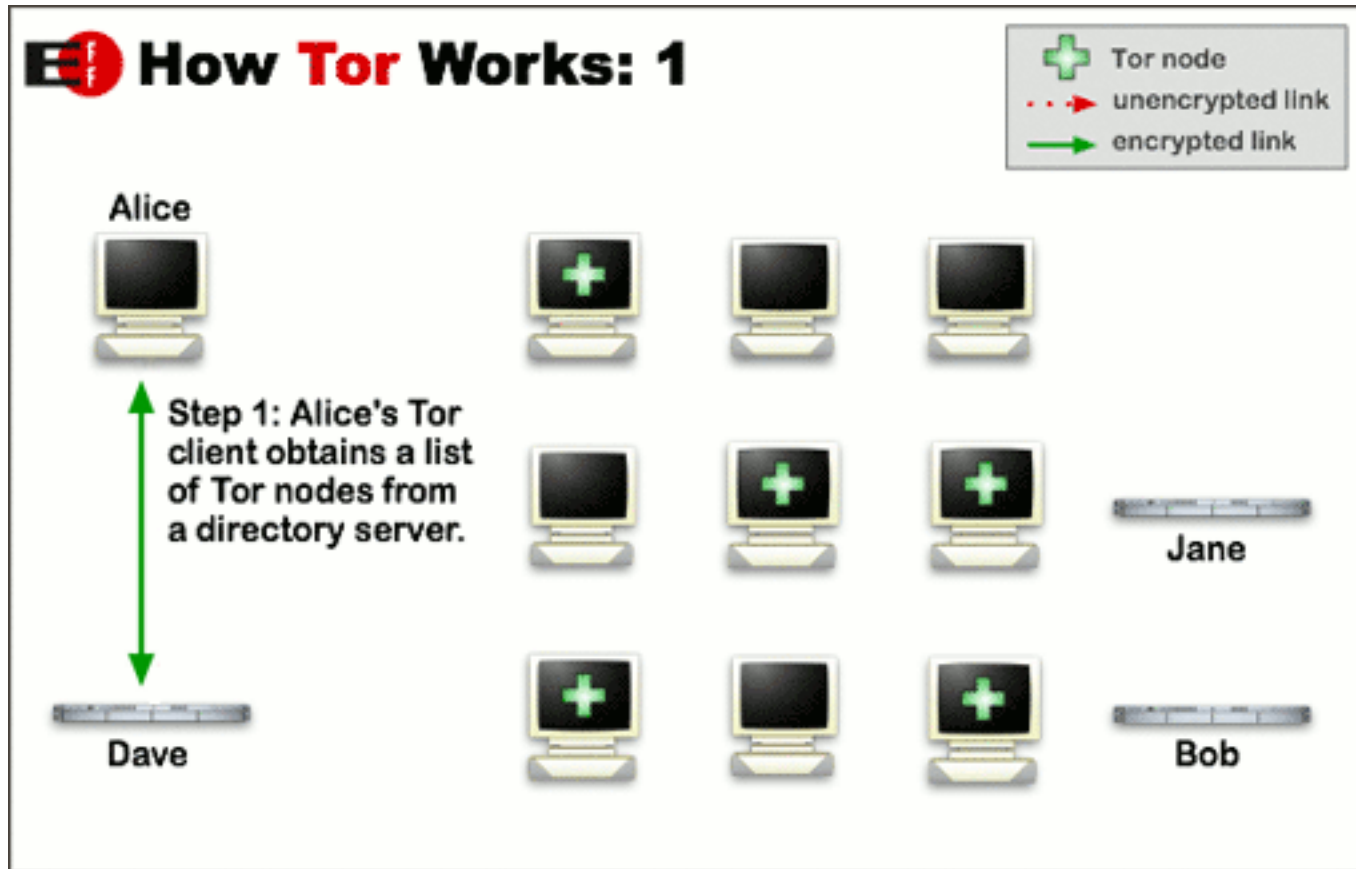
Onion Routing



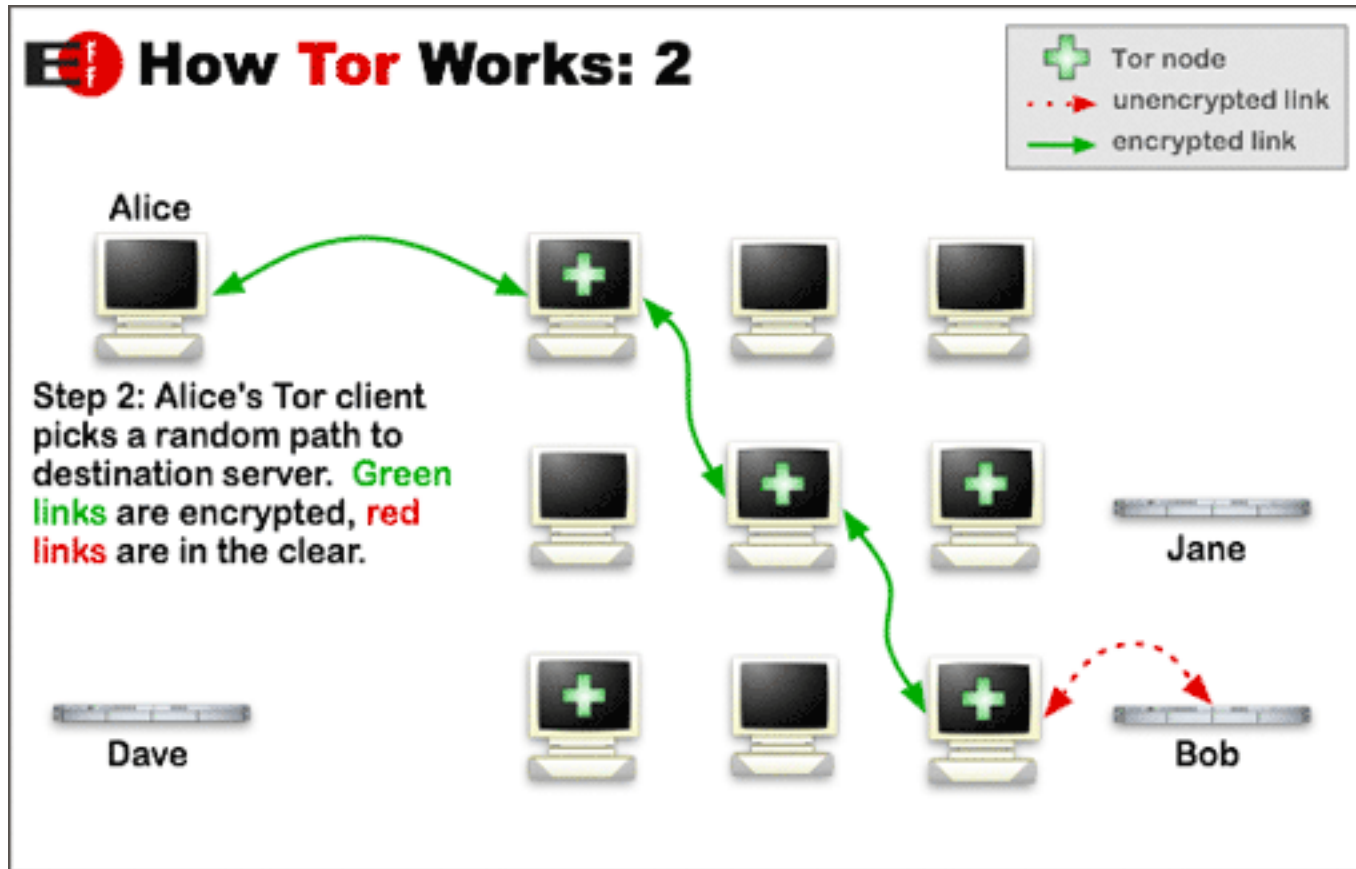
Note: this is what the industrial-strength **Tor** anonymity service uses.
(It also provides bidirectional communication)

Key concept: No one relay knows both you and the destination!

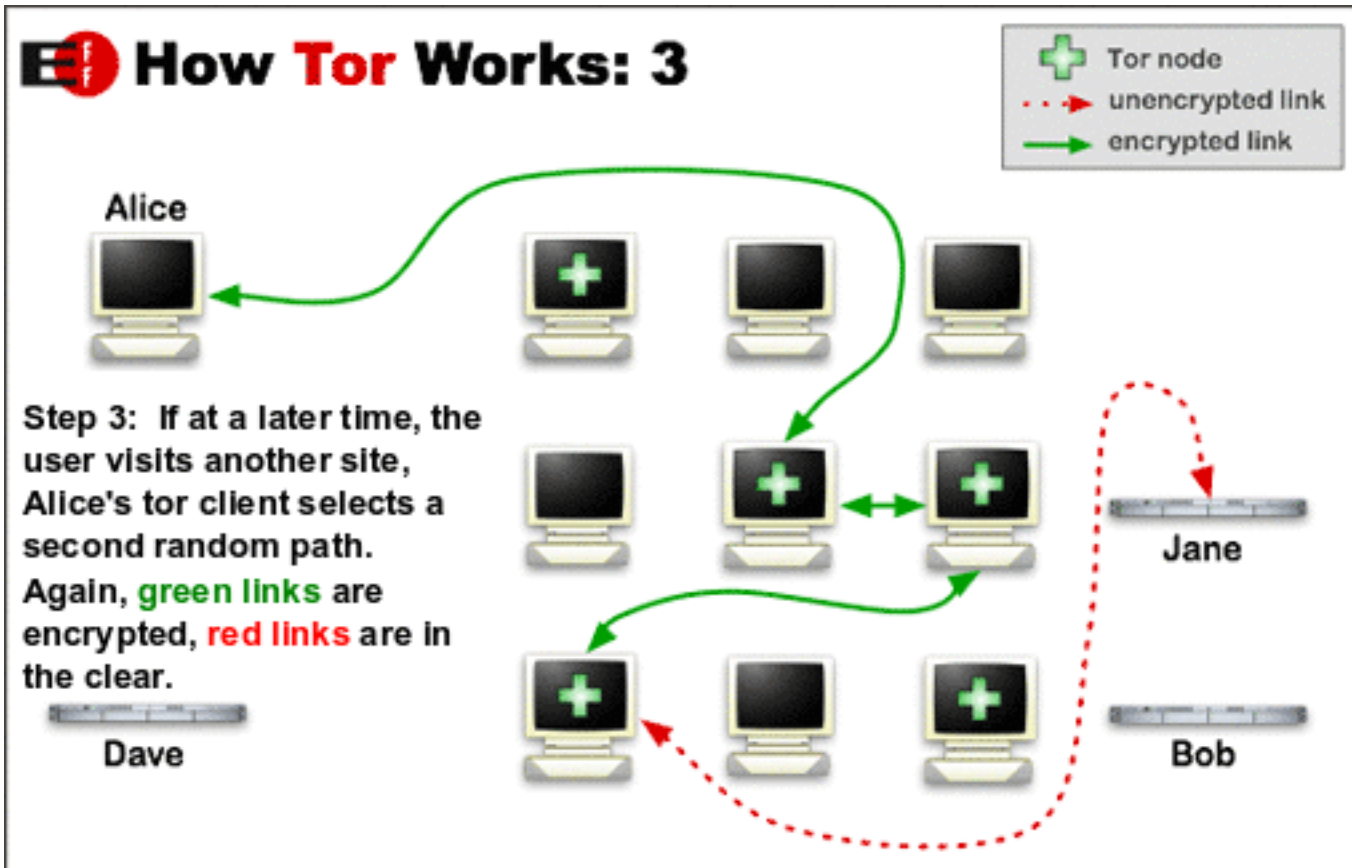
Tor



Tor



Tor



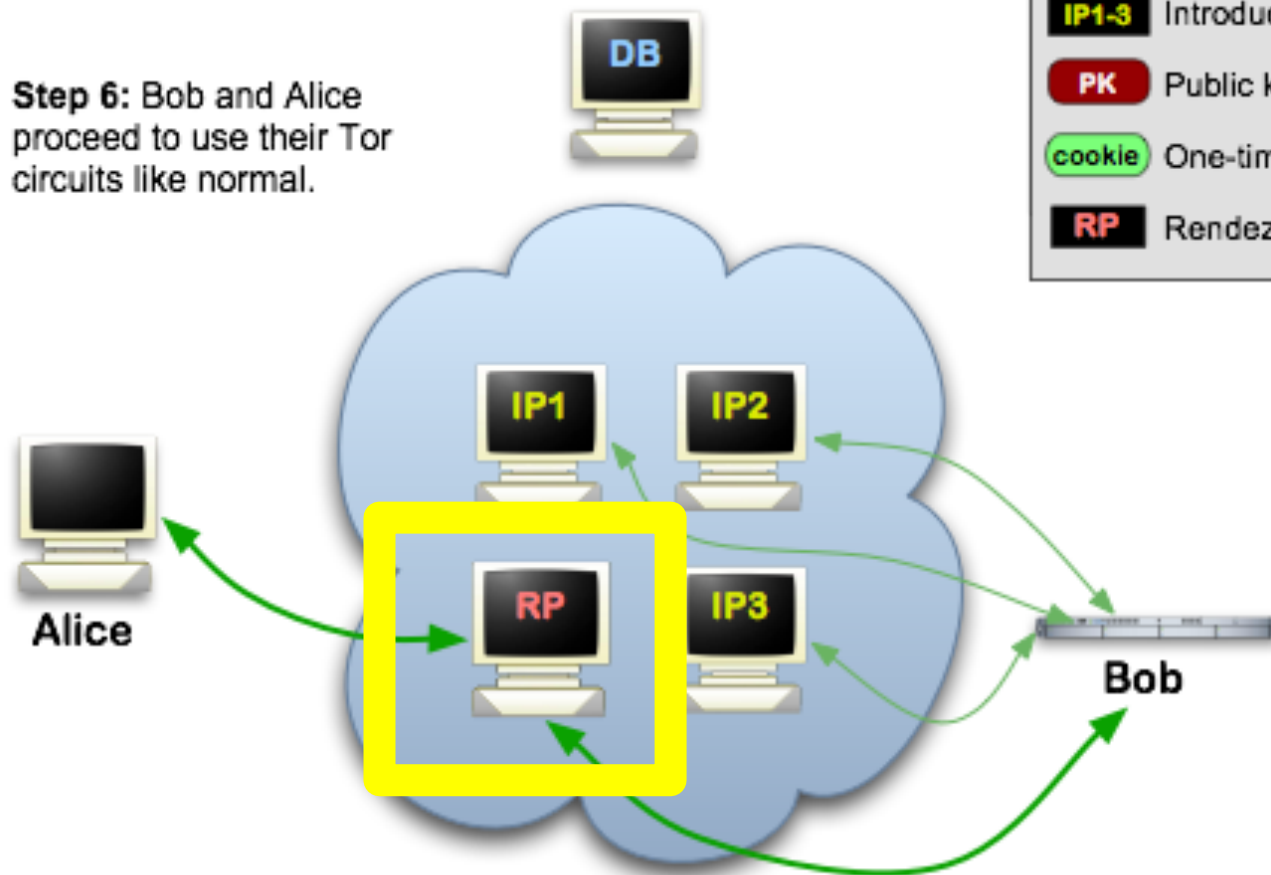
Trust in Tor

- Entry node: knows Alice is using Tor, and identity of middle node, but not destination
- Exit node: knows some Tor user is connecting to destination, but doesn't know which user
- Destination: knows a Tor user is connecting to it via the exit node
- Important to note that Tor does not provide encryption between exit and destination! (e.g., use HTTPS)

Tor Hidden Services

Hidden Services: 6

Step 6: Bob and Alice proceed to use their Tor circuits like normal.



How to get Tor

- Tor Browser bundle available (built on modified version of firefox)
- ☺ optional exercise: download and use it!
- <https://www.torproject.org/>
- ...or volunteer to be a part of the Tor network.

Onion Routing Issues/Attacks?

- Performance: message bounces around a lot
- Attack: rubber-hose cryptanalysis of mix operators
 - Defense: use mix servers in different countries
- Attack: adversary operates all of the mixes
 - Defense: have lots of mix servers (Tor today: ~6,500)
- Attack: adversary observes when Alice sends and when Bob receives, links the two together
- A side channel attack – exploits timing information
 - Defenses: pad messages, introduce significant delays
 - Tor does the former, but notes that it's not enough for defense

Onion Routing Issues, cont.

- Issue: traffic leakage
- Suppose all of your HTTP/HTTPS traffic goes through Tor, but the rest of your traffic doesn't
- How might the operator of sensitive.com deanonymize your web session to their server?

The traffic leakage problem

- Answer: they inspect the logs of their DNS server to see who looked up sensitive.com just before your connection to their web server arrived
- Hard, general problem: anonymity often at risk when adversary can correlate separate sources of information

Confirmed: Carnegie Mellon University Attacked Tor, Was Subpoenaed By Feds



JOSEPH COX

Feb 24 2016, 8:05am

Update 25 Feb: *In a statement, the Tor Project told Motherboard that "the Tor network is secure and has only rarely been compromised. The Software Engineering Institute ("SEI") of Carnegie Mellon University (CMU) compromised the network in early 2014 by operating relays and tampering with user traffic. That vulnerability, like all other vulnerabilities, was patched as soon as we learned about it. The Tor network remains the best way for users to protect their privacy and security when communicating online."*


Metadata

- If
- When
- How much
- Who
- What

Metadata

- If
- When
- How much
- Who
- What ← TLS/PGP/OTR/Signal

Metadata

- If
- When
- How much
- Who ← 
- What ← TLS/PGP/OTR/Signal

Pond

- "Pond is not email. Pond is a forward secure, asynchronous messaging system for the discerning"
- Seeks to protect against leaking traffic info against all but a global passive adversary
 - forward secure
 - no spam
 - messages expire automatically after a week

Pond



User

Private Key

Public Key



Pond
Server

Messages? Pubkey=A padding=XXXX..

None. padding=XXXXXXXXXXXXXXXXX...



Messages? Pubkey=A padding=XXXX..

Message=M padding=XXXXXXXXX...



Pond



User

Private Key

Public Key



Pond
Server

Messages? Pubkey=A padding=XXXX..

None. padding=XXXXXXXXXXXXXXXXX...



Messages? Pubkey=A padding=XXXX..



Message=M padding=XXXXXXXXX...



Private key



Metadata summary

- If
- When ← 
- How much ←
- Who ← 
- What ← TLS/PGP

DARK WEB

Darkweb \neq Deep Web

- Deep web – websites not indexed by search engines (hard to find)
- Dark web – websites accessible via specialized software and protocols (overlay networks)
- Darkweb \subset Deep Web

The Deep Web

An iceberg floating in a blue ocean. The tip of the iceberg is above the water line, representing the public web. The much larger part of the iceberg is submerged below the water line, representing the deep web. The iceberg is composed of various shades of blue triangles.

THE PUBLIC WEB

A donut chart showing 4% of the total. The chart is mostly white with a small blue segment representing 4%.

4%

of web content (~8 billion pages)
is available via search engines
like Google

THE DEEP WEB

7.9 Zettabytes

A donut chart showing 96% of the total. The chart is mostly blue with a small white segment representing 4%.

96%

of the digital universe
is on “deep websites”
protected by passwords

Source: The Deep Web: Semantic Search Takes Innovation to New Depths

FREENET



Freenet

- Peer to peer (no central authority)
- Distributed data store (platform for publishing and sharing files)
- Hide the uploader and downloader of content
- Fight censorship, protect free speech

Freenet

- Each participant is a "node" in the network
- Data is distributed in a network
- The network is a distributed cache:
 - Files broken up into blocks (redundant)
 - Each block is stored independently
 - Contents of file are copied and spread across many nodes
- Achieved by distributed hash table (DHT)

Distributed Hash Table

- Search for blocks using a hash value
 - Key-value pairs
- Key-value table can't belong to one node
 - violates P2P principles
 - could be too big for one node
- Values must be distributed through the network in a balanced way

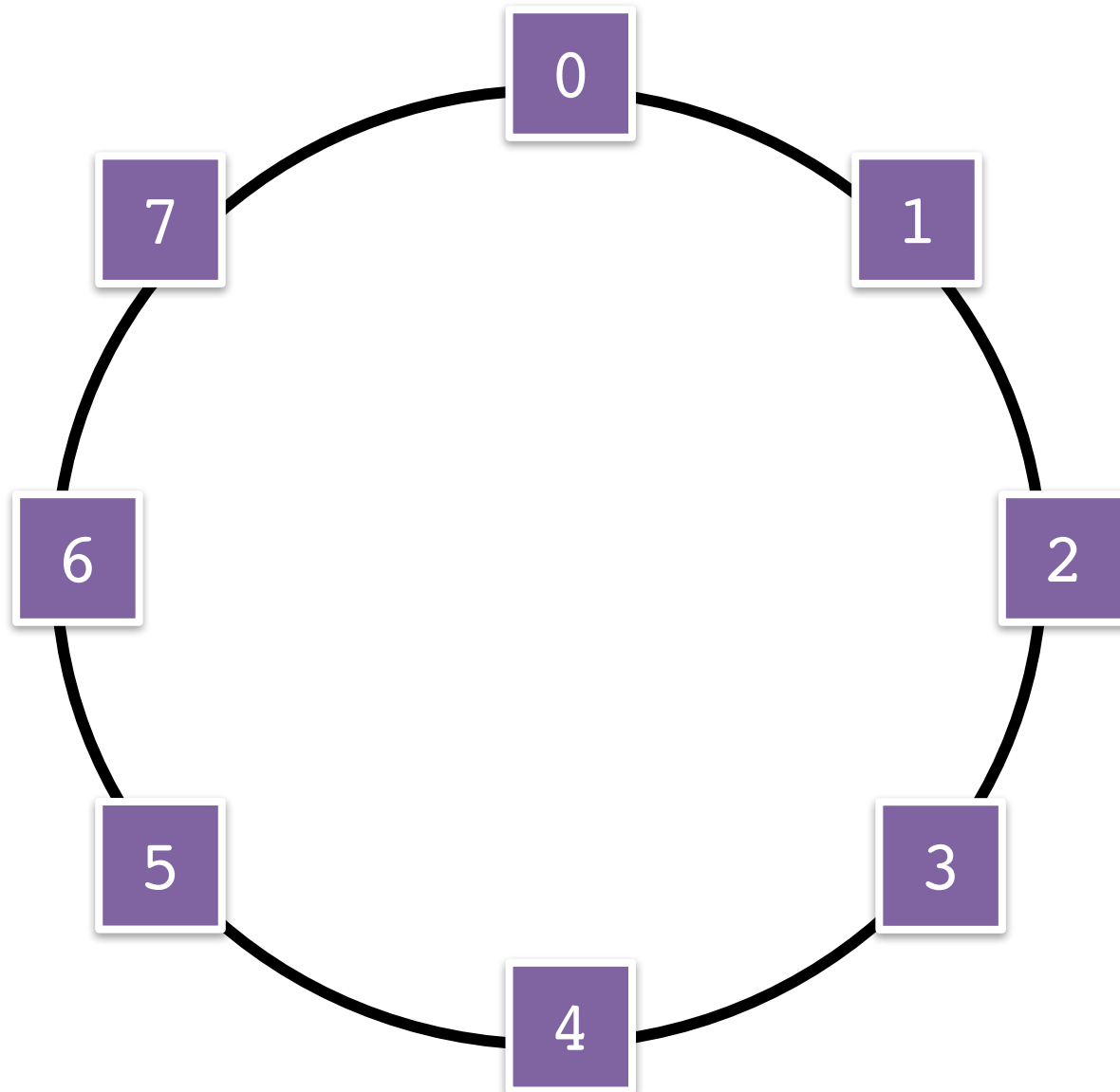
Distributed Hash Table

- Choose a hash function with m bit output
 - MD5 $m=128$ bits
 - SHA-1 $m=160$ bits
 - Example $m=6$
 - Output from 0 to 63
 - SHA-256 = 256 bits

Chord

- One DHT implementation: Chord
- Similar, but not used in Freenet
- Imagine output space (range, image) of hash as a circle
 - Like a clock, starts at 0 and proceeds clockwise to 2^m-1

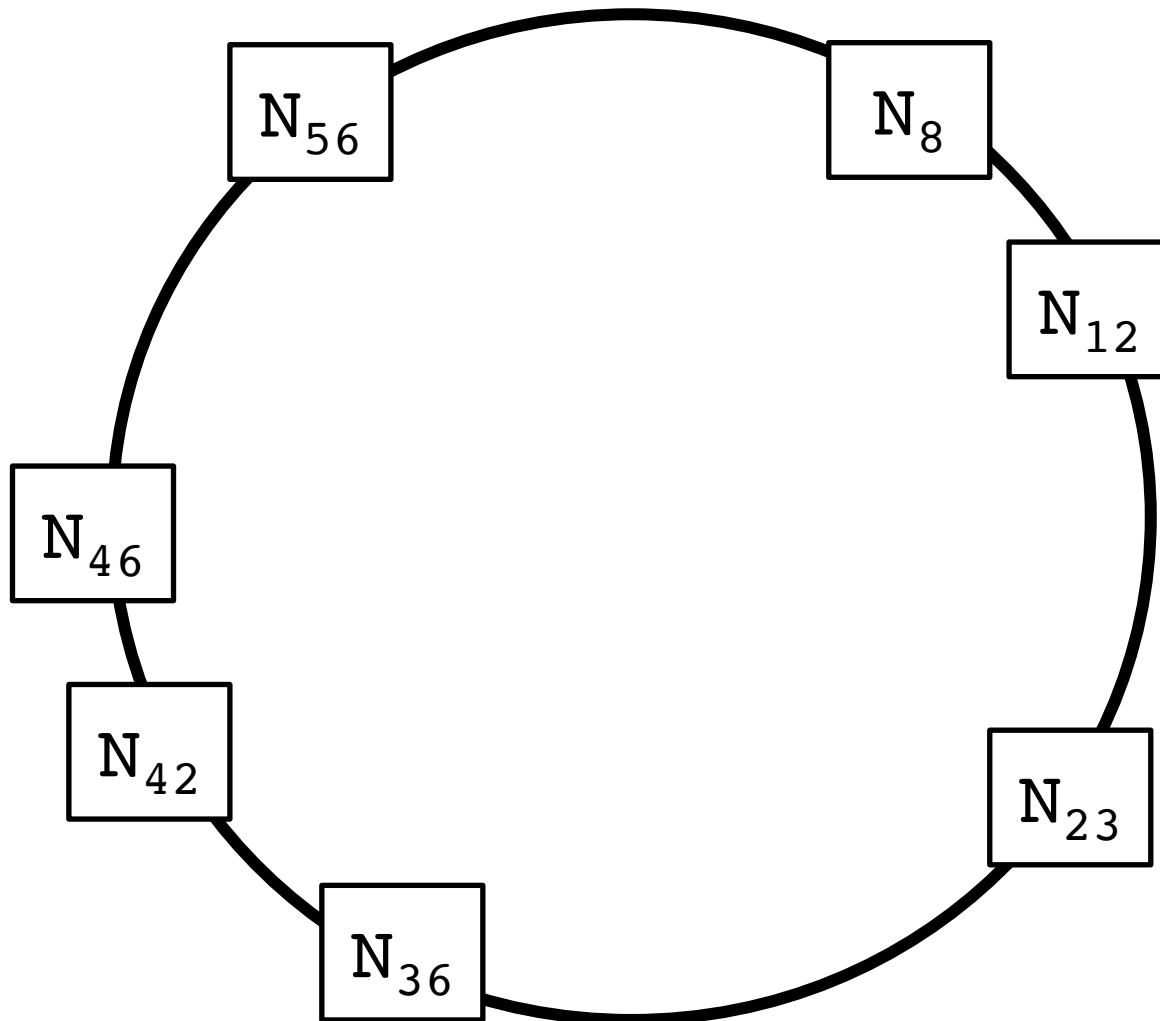
Circle ($m=3$)



Chord Circle

- Nodes are distributed on this hash circle:
 - Hash unique identifier (e.g. IP address, username, etc.)
 - Track predecessor and successor
 - predecessor - first node with a lower hash
 - successor - first node with a higher hash
 - Predecessor and successors “wrap around” the origin (just like a clock)

Chord Circle (m=6)



Chord Circle

- Nodes organized
- What about data? $\langle K, V \rangle$
 - Use the same hash function!
 - Find smallest node where $h(K) \leq h(N)$
 - Store $\langle K, V \rangle$ there

Example

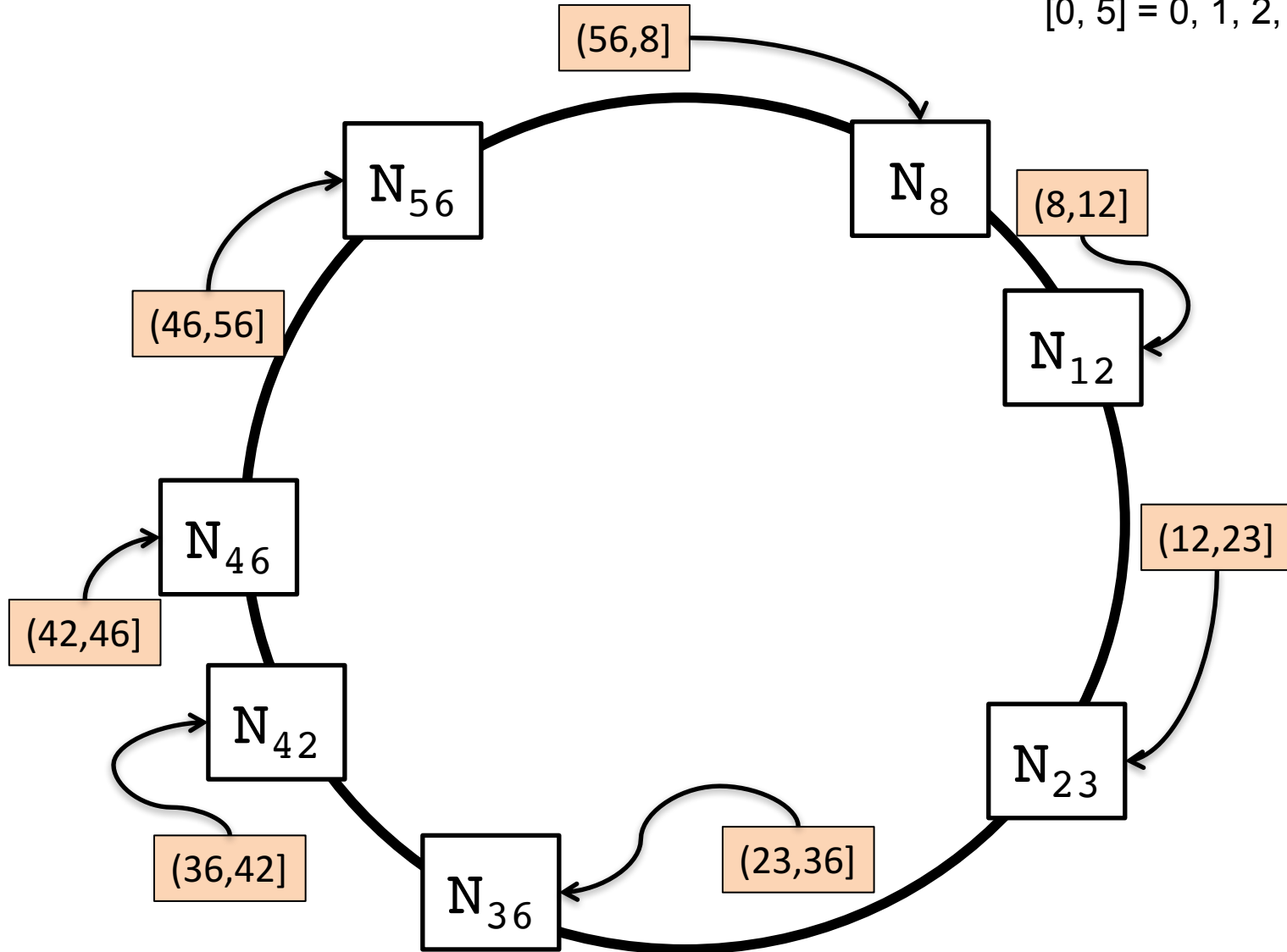
Example (mhasan11):

$(0, 5) = 1, 2, 3, 4$

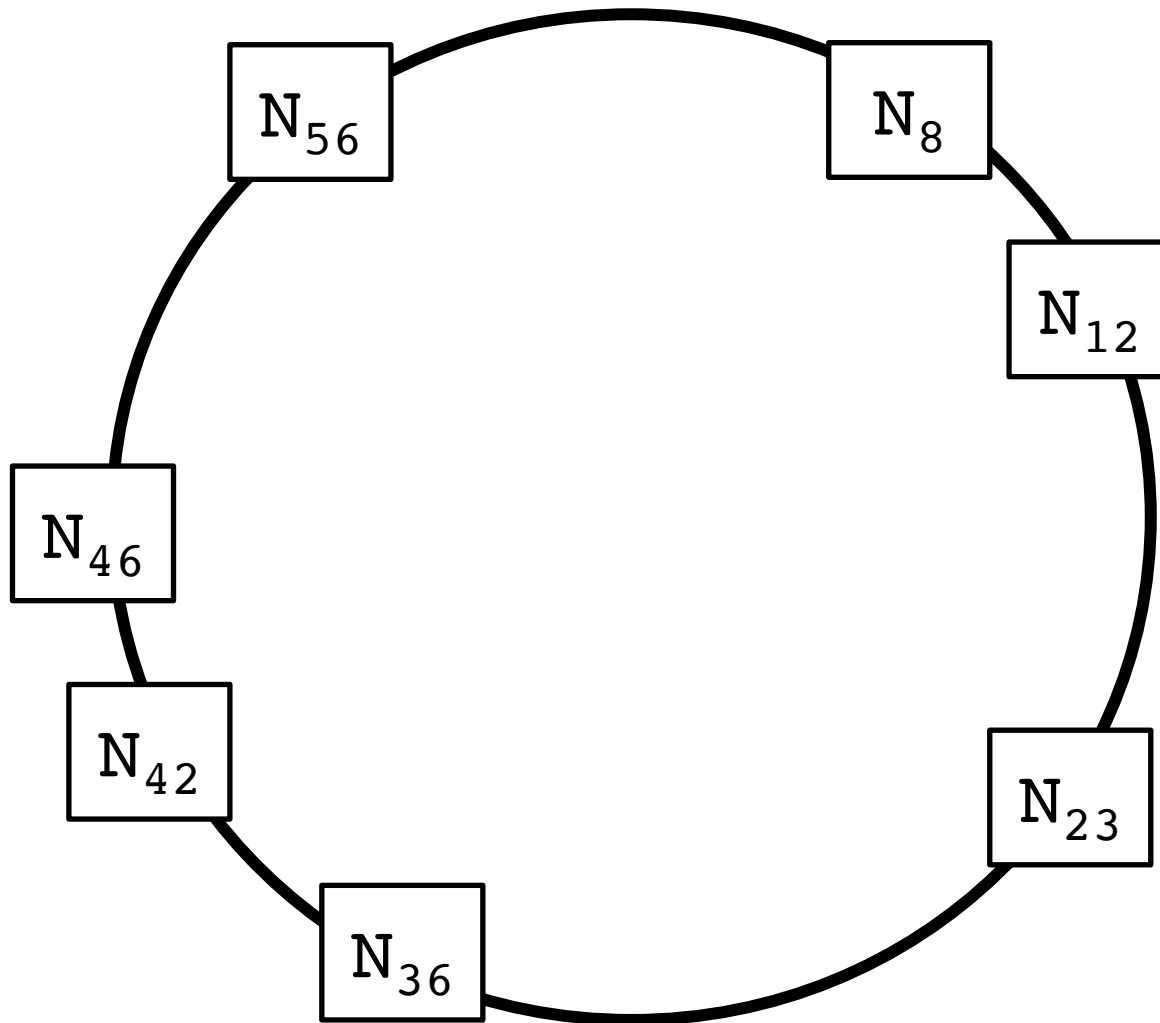
$(0, 5] = 1, 2, 3, 4, 5$

$[0, 5) = 0, 1, 2, 3, 4$

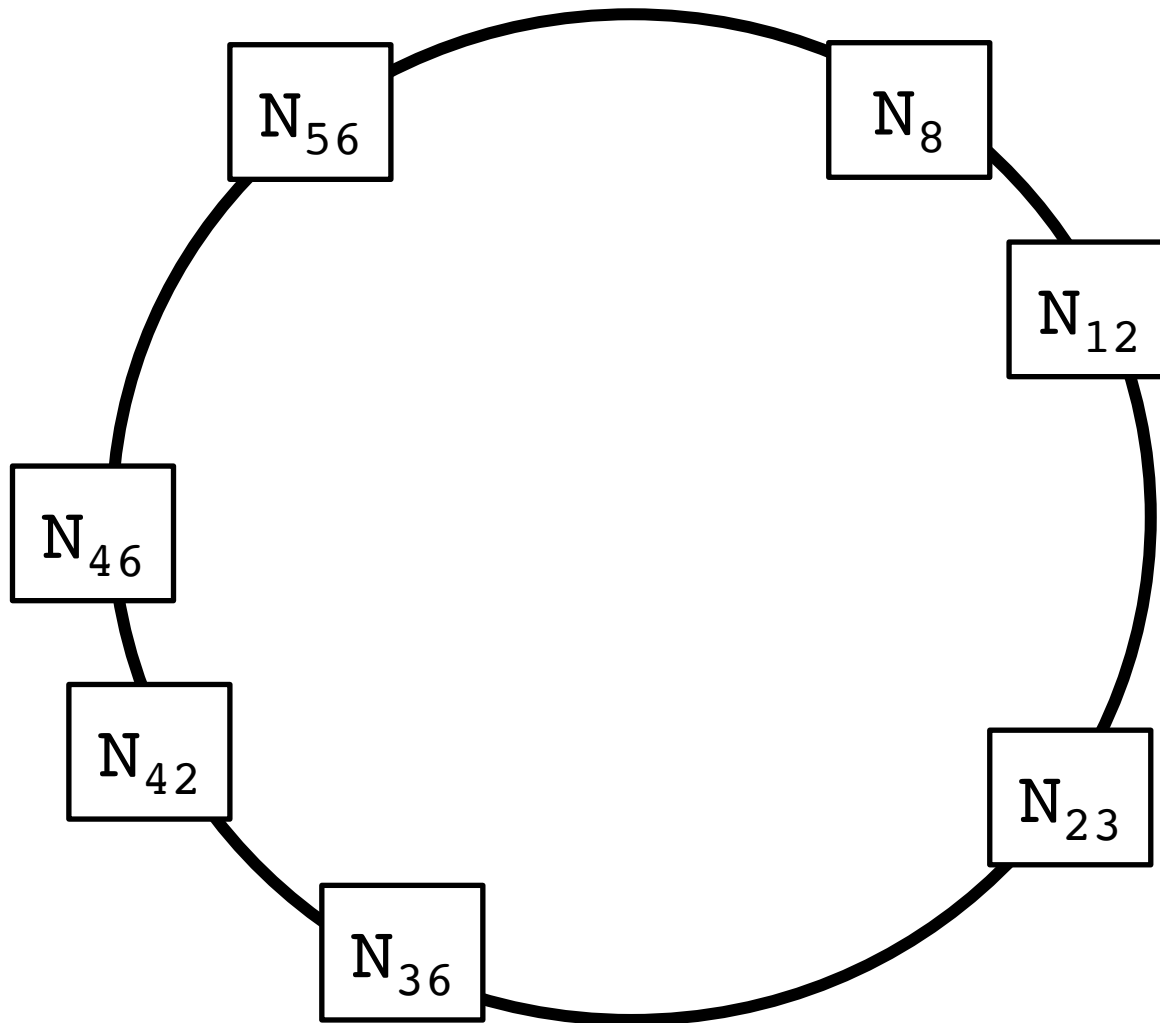
$[0, 5] = 0, 1, 2, 3, 4, 5$



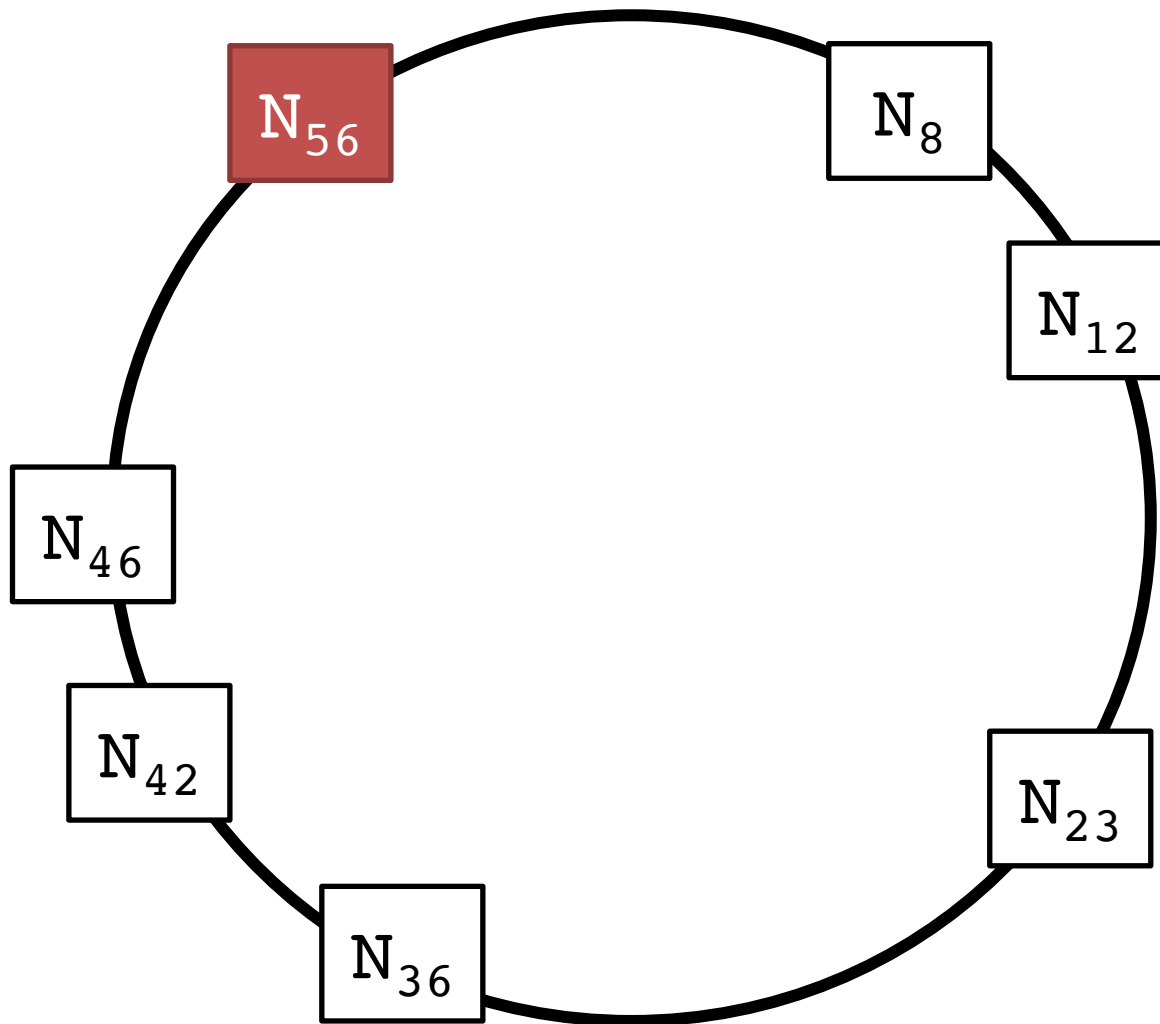
<“Electric Feel, MGMT”,electricfeel.mp3>



<“Electric Feel, MGMT”,electricfeel.mp3>
 $h(\text{“Electric Feel, MGMT”})=54$

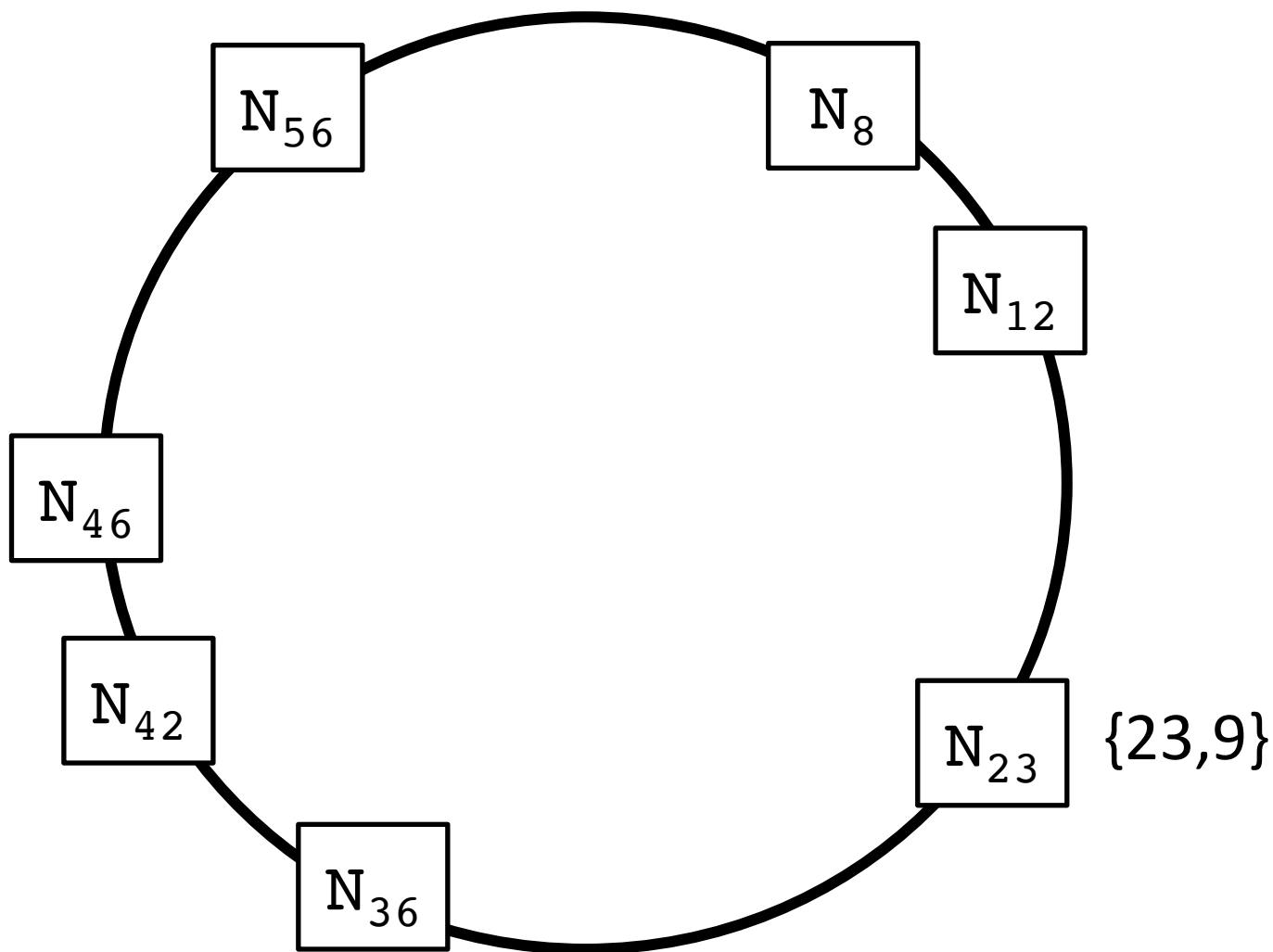


<“Electric Feel, MGMT”,electricfeel.mp3>
 $h(\text{“Electric Feel, MGMT”})=54$



Naïve Search

- Send a message around the circle until we find the right node
 - Send out our hash and hash of the search key
 - e.g. {23,9} “node 23 is looking for file with hash 9”
 - Correct node directly sends back result



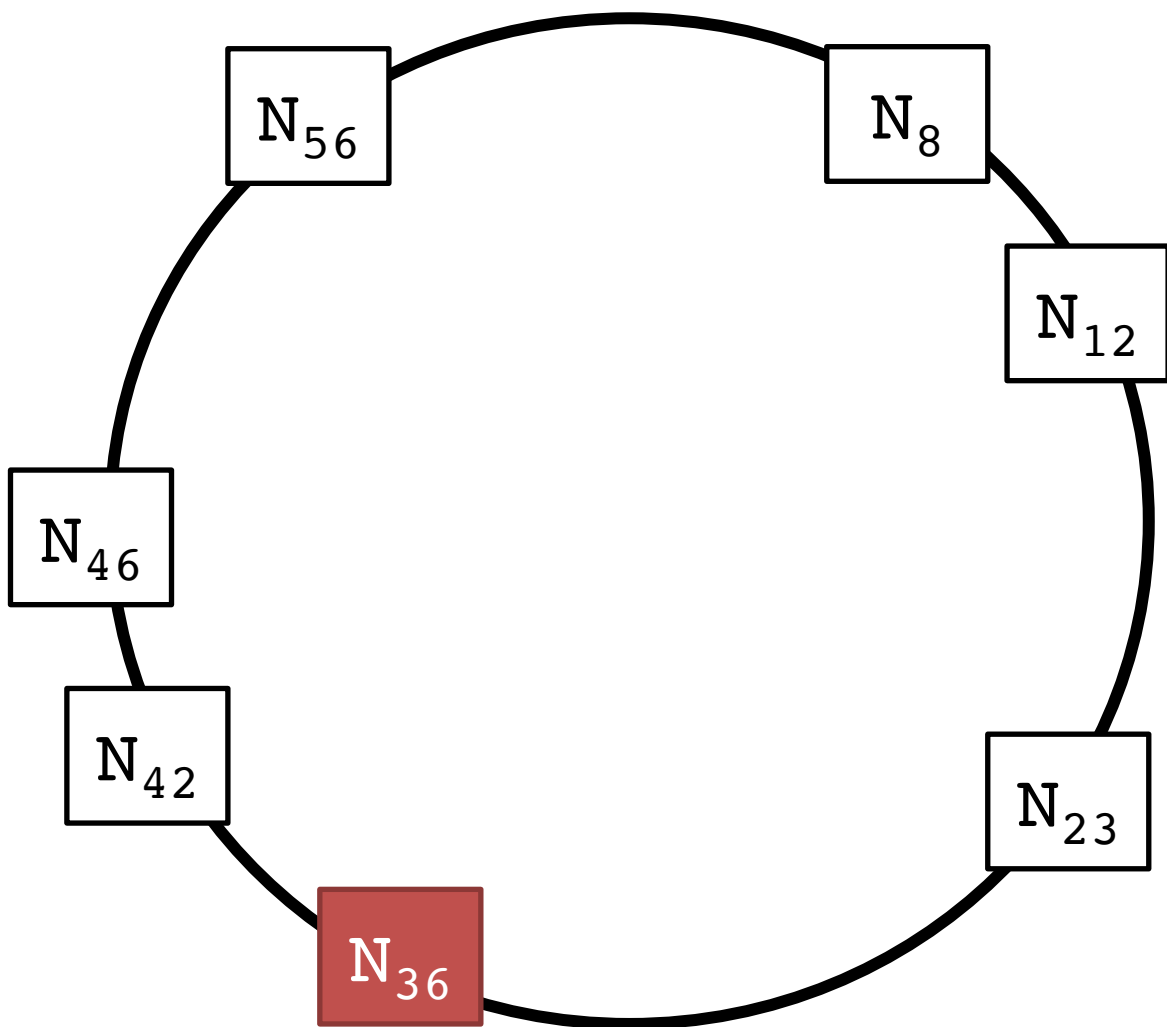
Naïve Search

- Too expensive
 - Have to send, on average $O(\text{Nodes})$ messages
 - Too slow!
- Keep track of all other nodes?
 - Have to store $O(\text{Nodes})$ addresses
 - Too big!
 - Too fiddly!

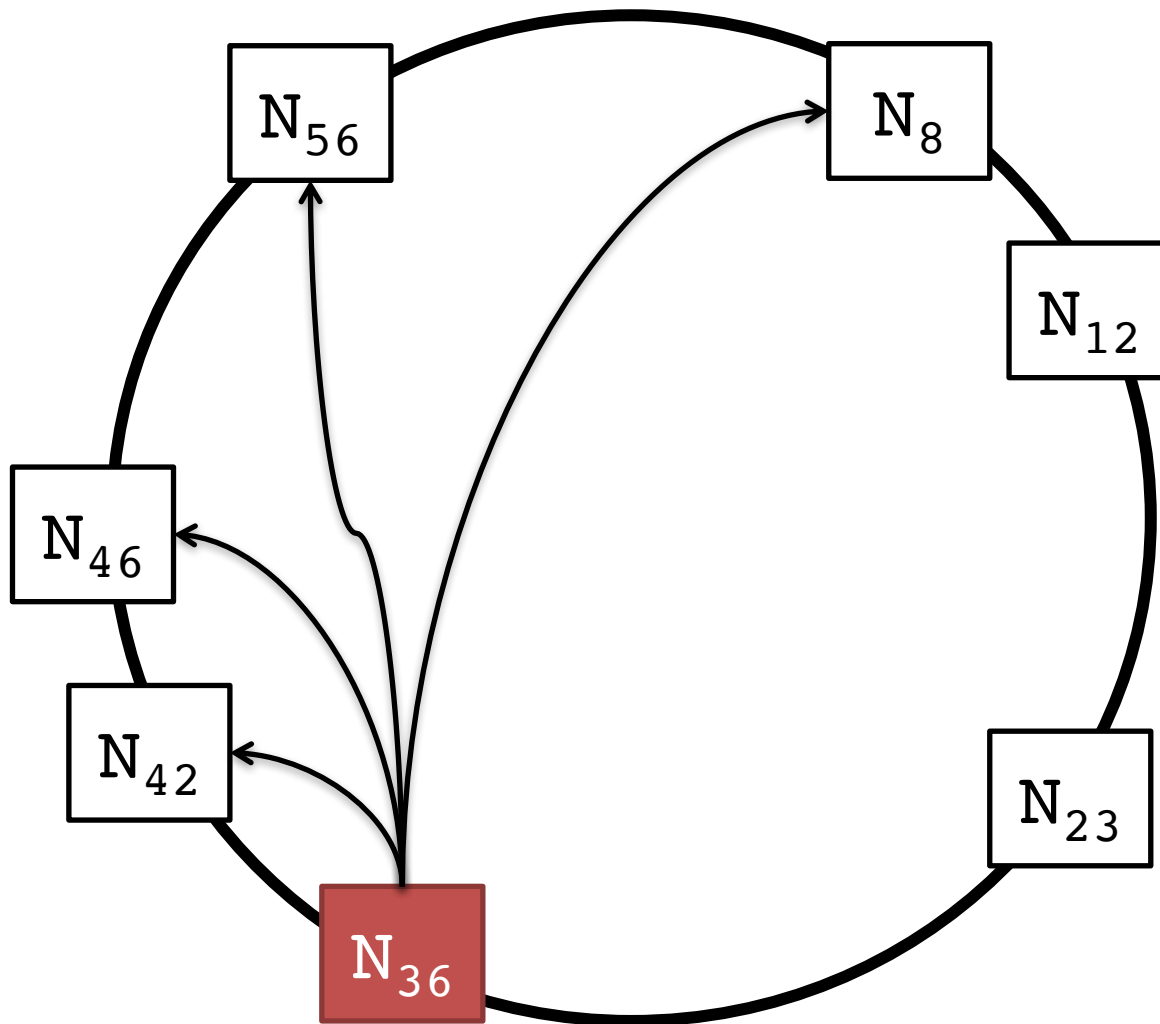
Finger Table

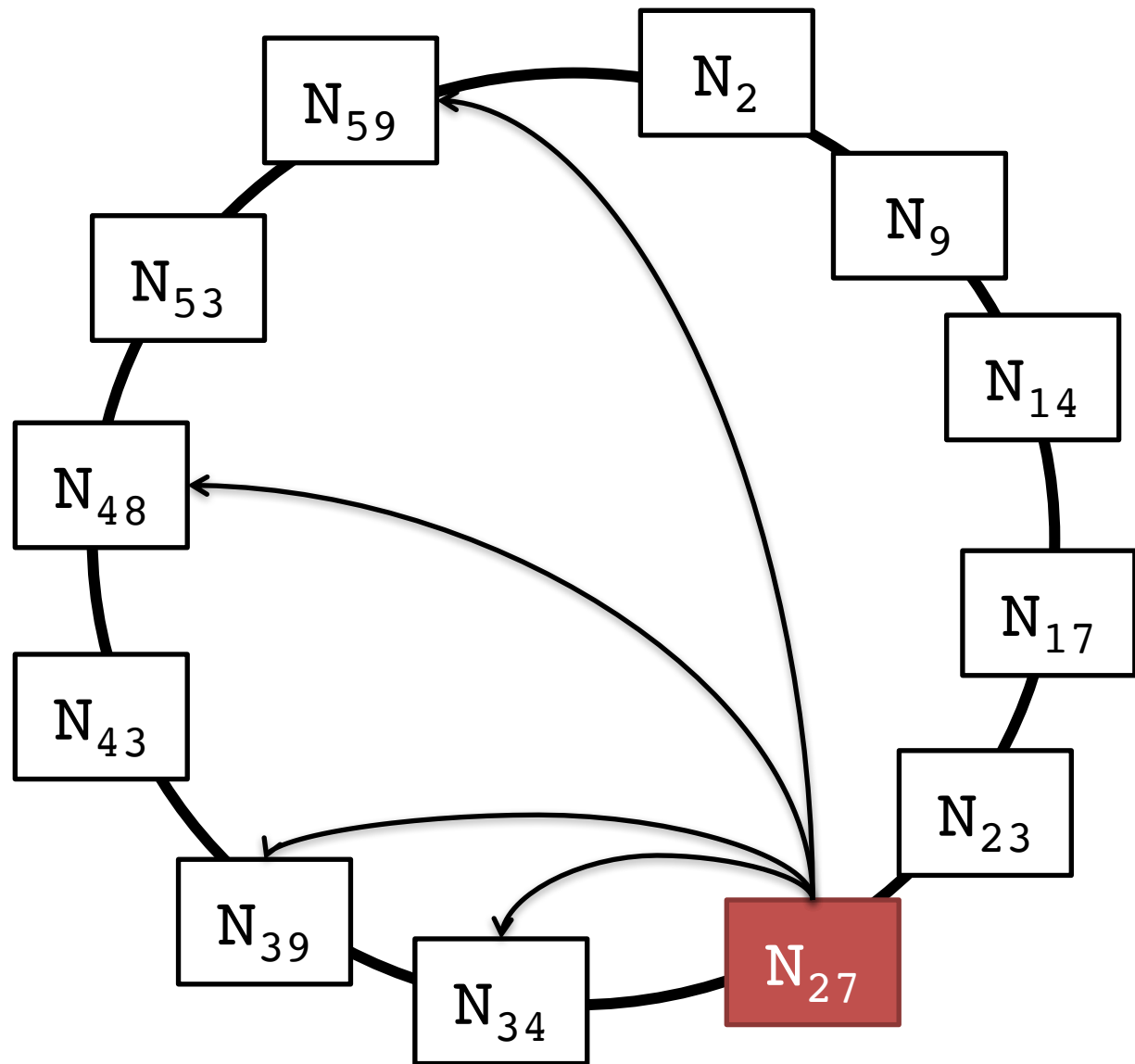
- Track all nodes at power of 2 distances
 - $1, 2, 4, 8, 16, \dots, 2^{m-1}$
 - Distance in hash codomain (range, image)
- Basically, look up value that distance away
 - Find $h(N) + \text{distance} \bmod 2^m$

Distance	1	2	4	8	16	32
Node	42	42	42	46	56	8



Distance	1	2	4	8	16	32
Node	42	42	42	46	56	8





Distance	1	2	4	8	16	32
Node	34	34	34	39	48	59

Finger Table

- How does this help?
 - Route our searches like a binary search
 - First node sends it halfway there
 - Next node sends it halfway again
 - Repeat until we find the right node

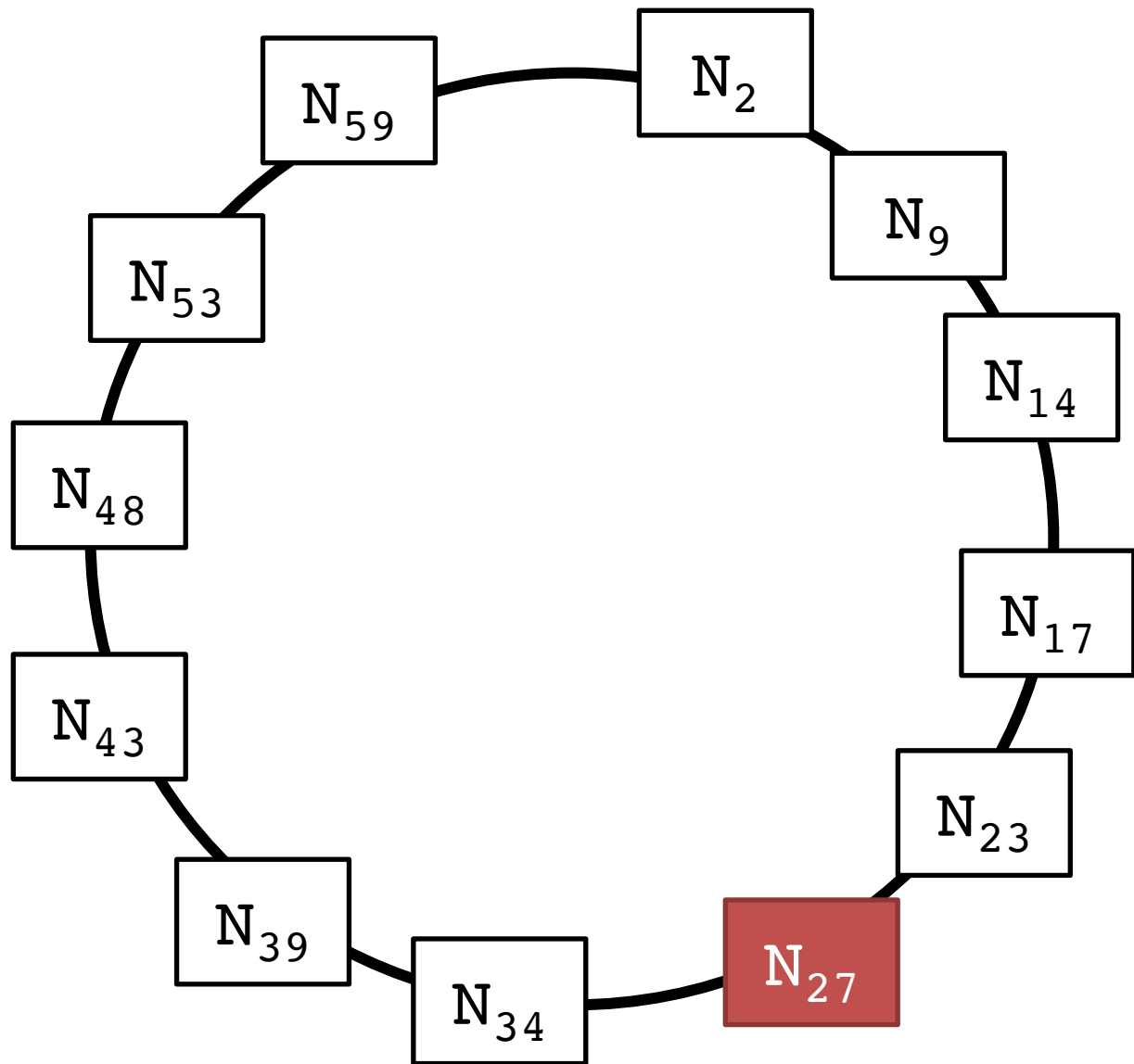
Finger Table Search

- Terminology
 - N_r is requesting node (initiated search)
 - N_c is current node (initially $N_c = N_r$)
 - x is the hash of search key ($h(K) = x$)

Finger Table Search

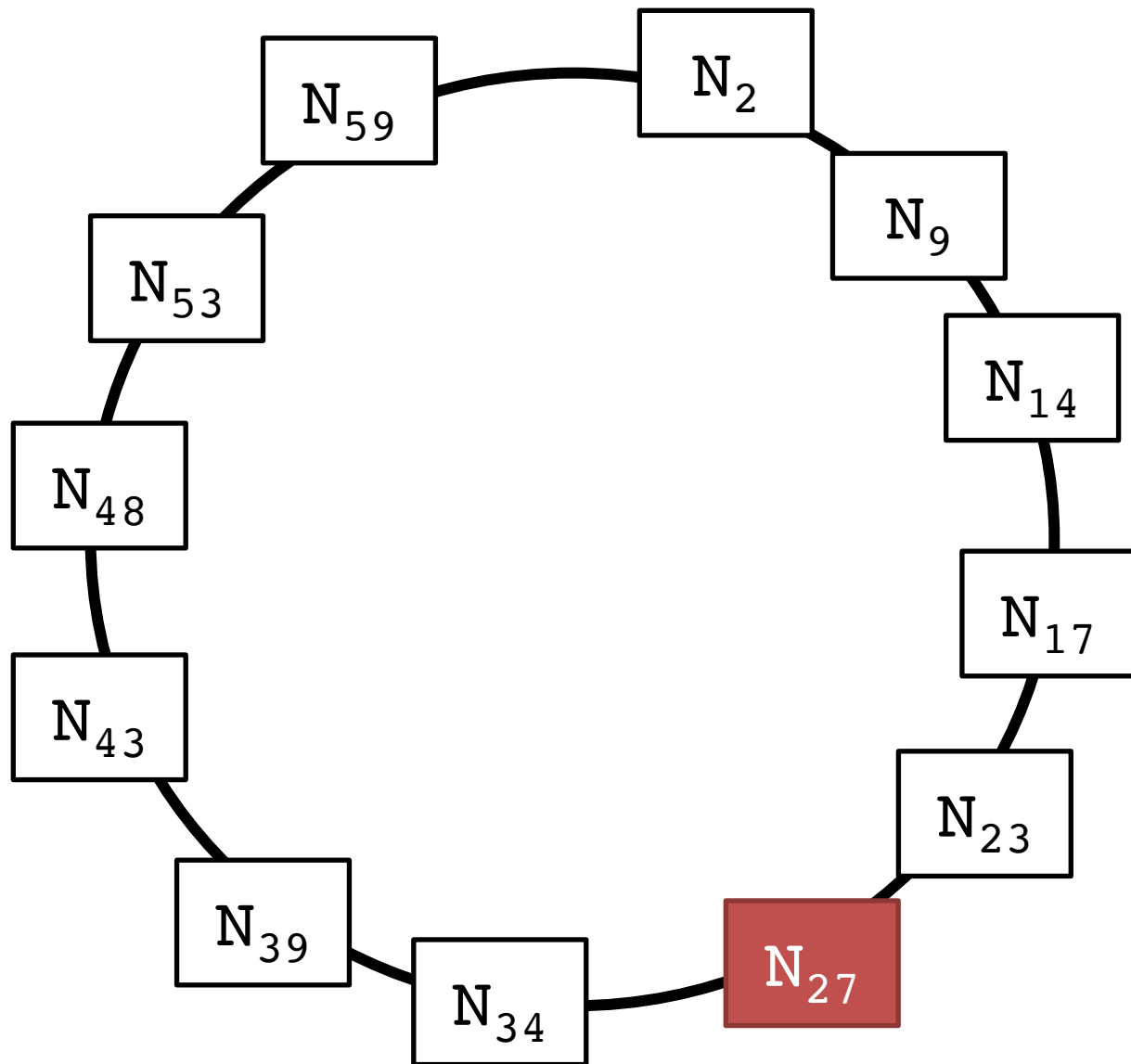
1. N_c checks if its successor N_s has search key $(c < x \leq s)$
 - If so, N_s looks up x and returns result to N_r .
 - Search is done.
2. N_c searches its finger table for highest numbered node N_h less than $h(x)$.
 - Tell N_h to search for x on behalf of N_r
 - Now $N_c = N_h$ and search continues

N₂₇ searching
for h(x)=19



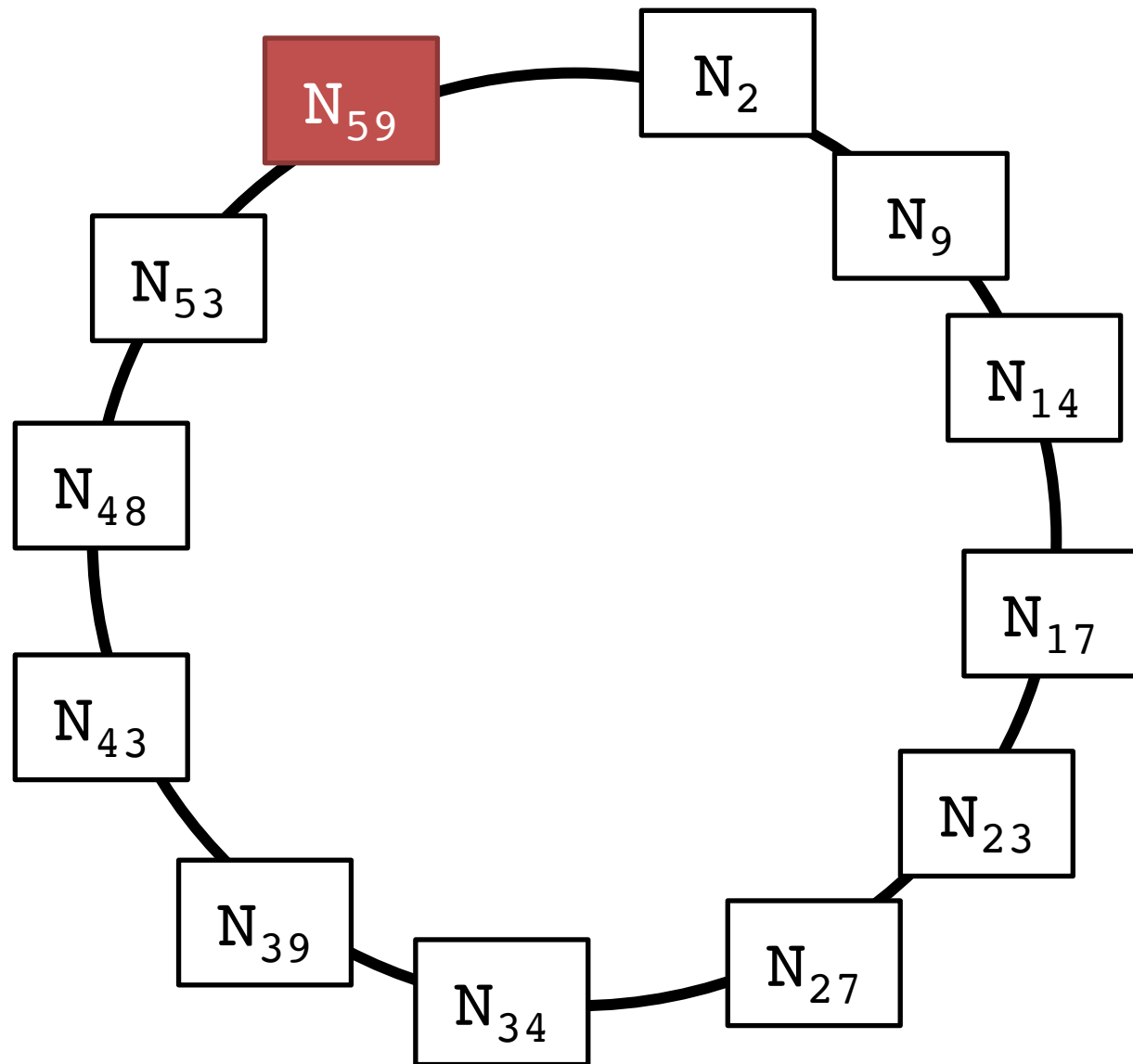
Distance	1	2	4	8	16	32
Node	34	34	34	39	48	59

N₂₇ searching
for h(x)=19



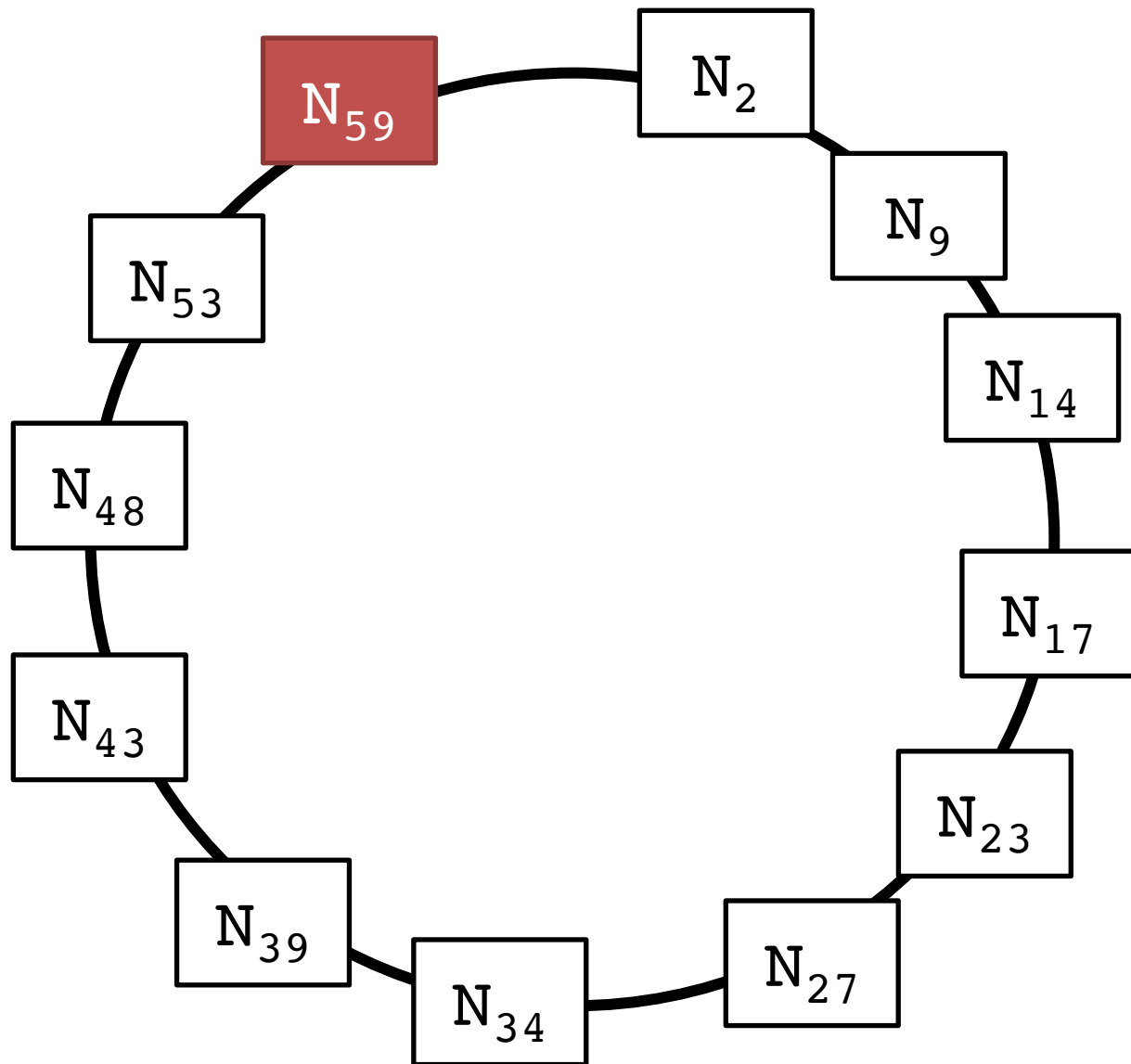
Distance	1	2	4	8	16	32
Node	34	34	34	39	48	59

N_{27} searching
for $h(x)=19$



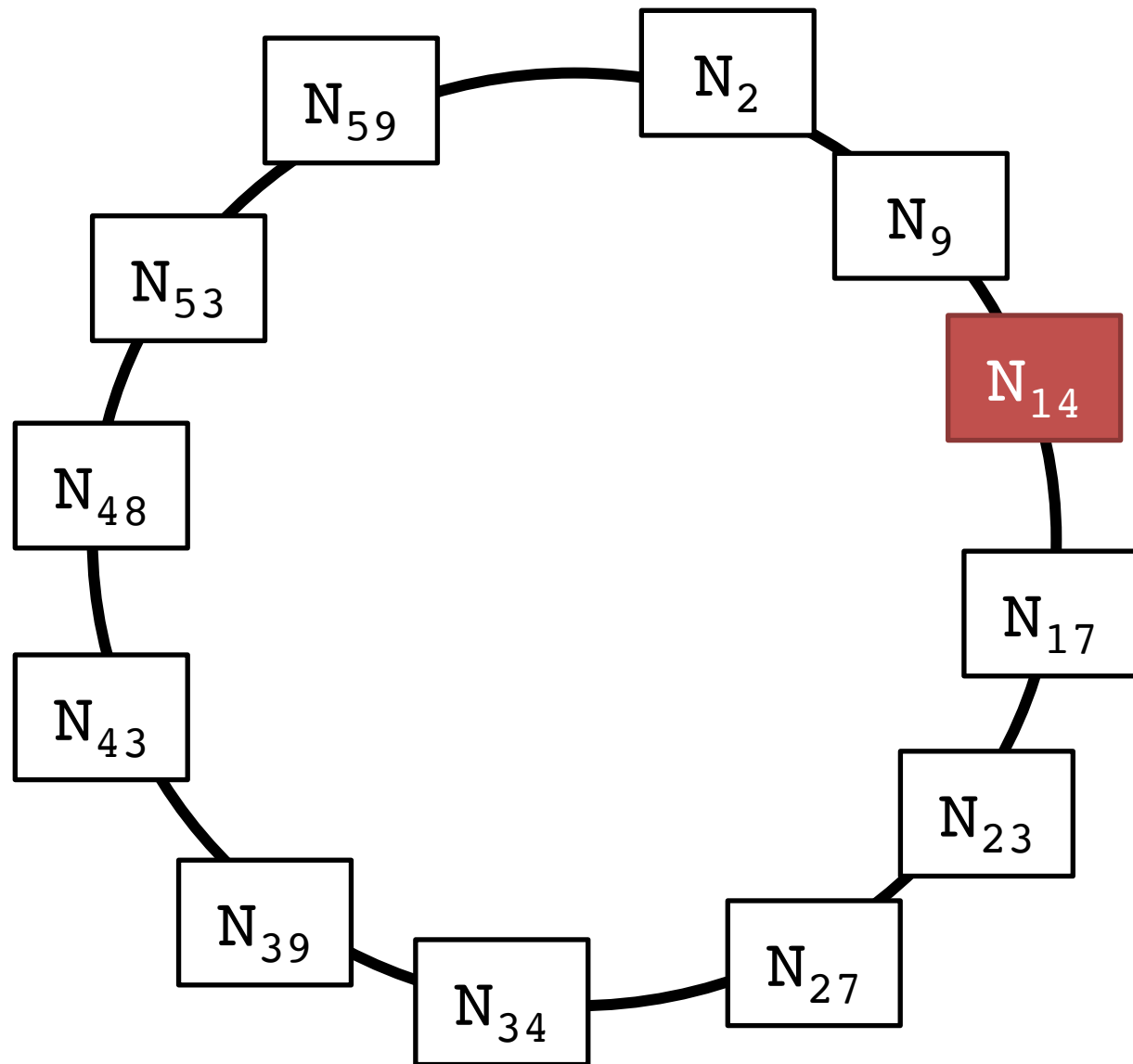
Distance	1	2	4	8	16	32
Node	2	2	2	9	14	27

N_{27} searching
for $h(x)=19$



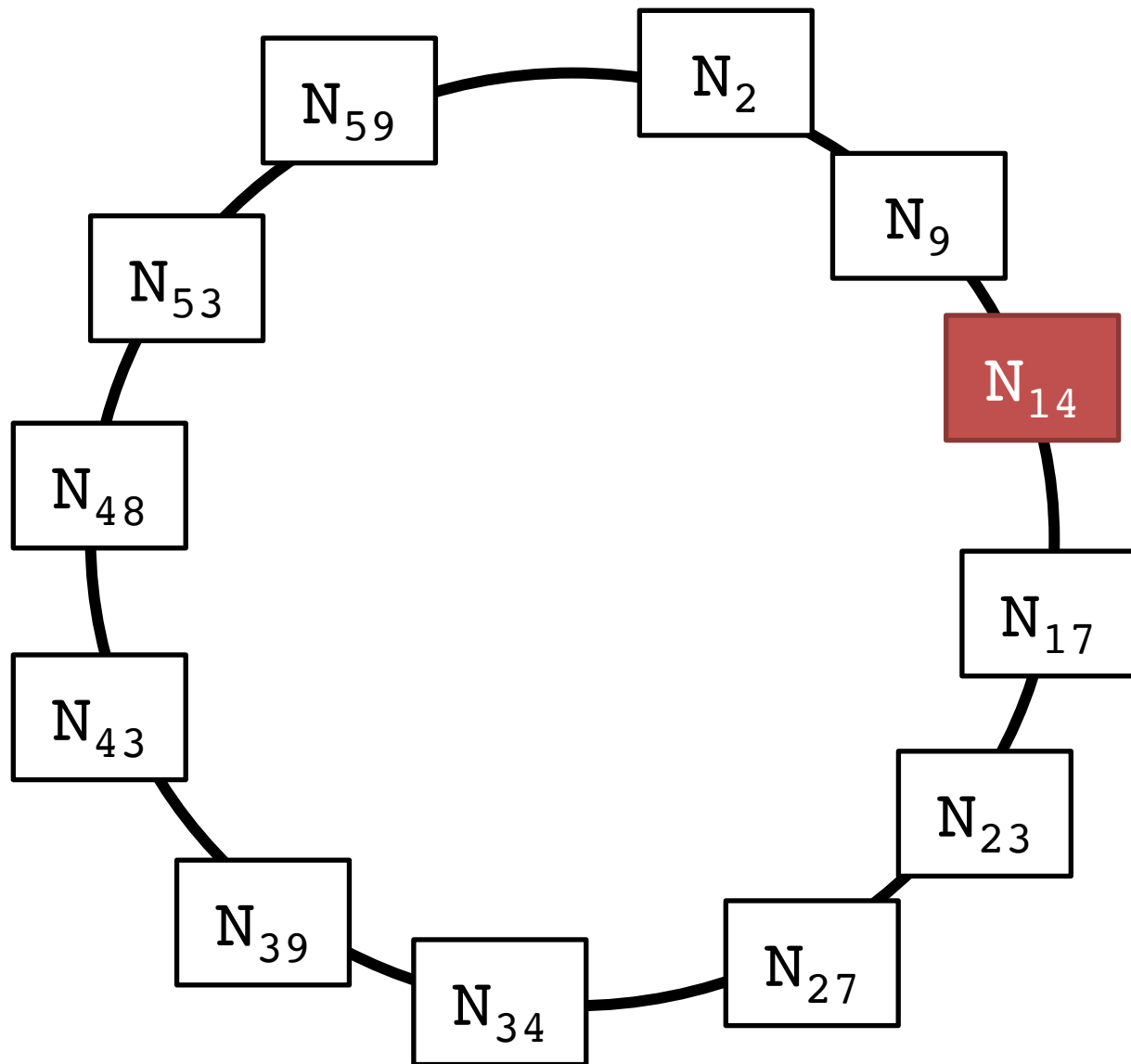
Distance	1	2	4	8	16	32
Node	2	2	2	9	14	27

N_{27} searching
for $h(x)=19$



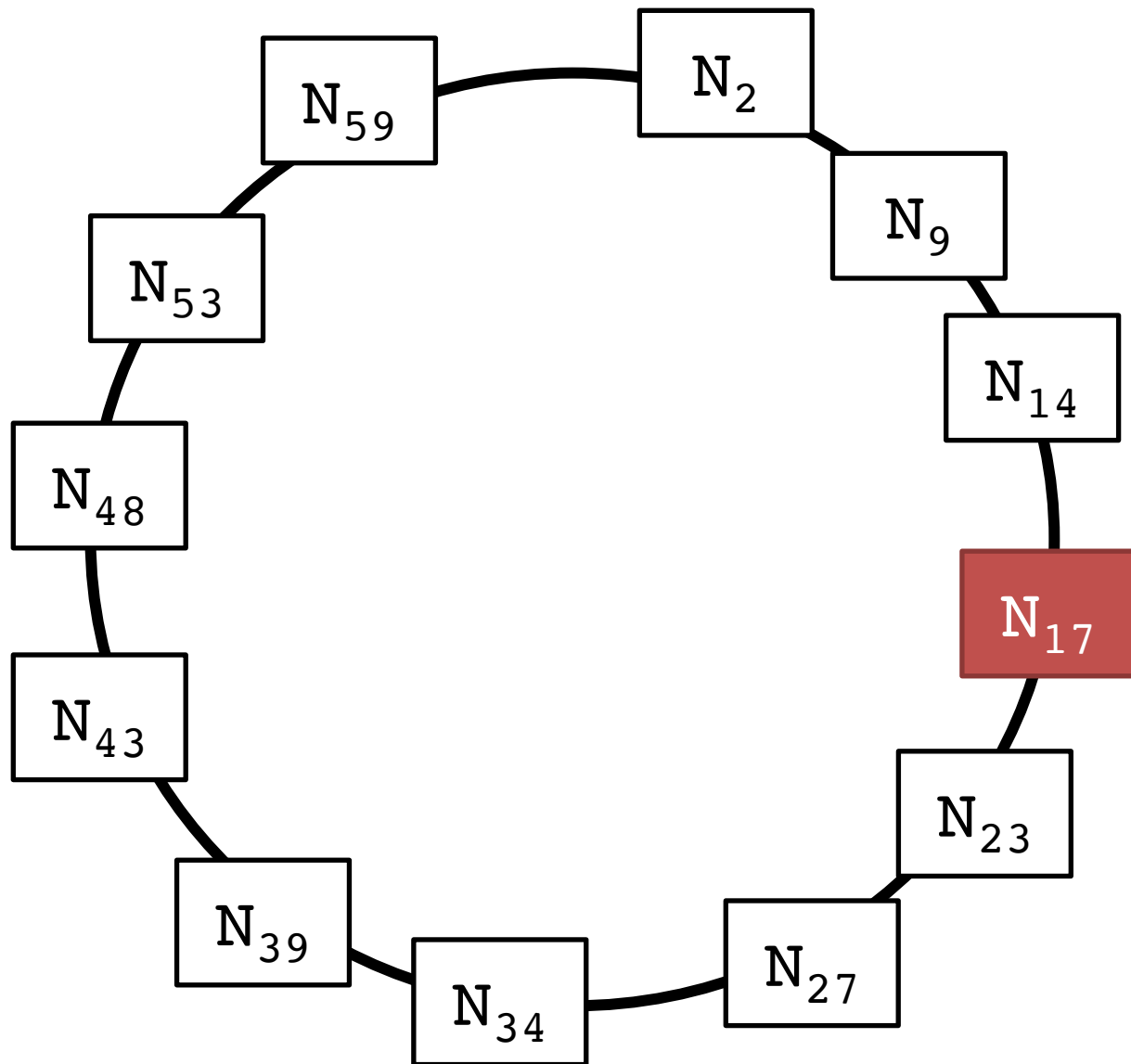
Distance	1	2	4	8	16	32
Node	17	17	23	23	34	48

N_{27} searching
for $h(x)=19$



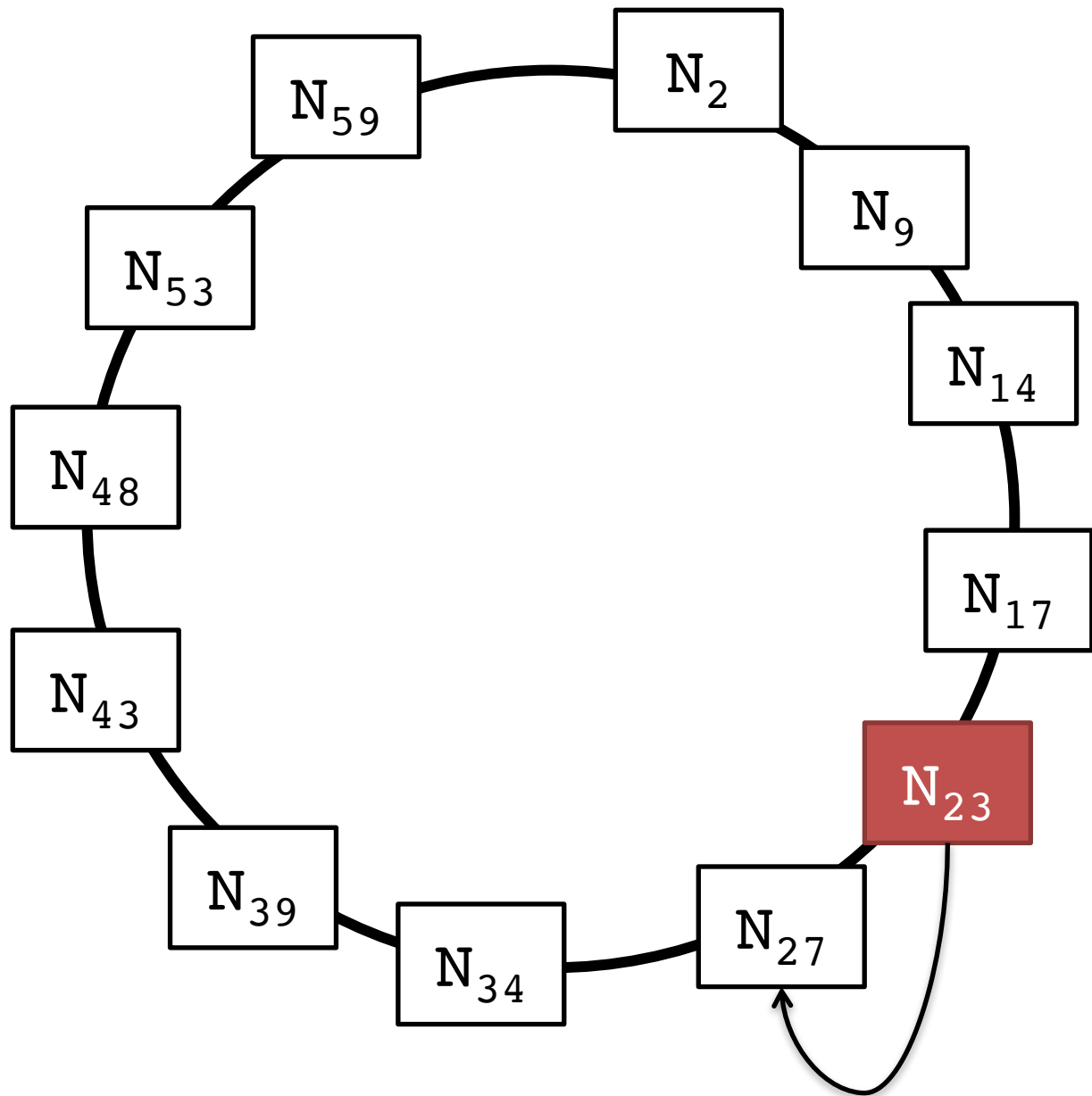
Distance	1	2	4	8	16	32
Node	17	17	23	23	34	48

N₂₇ searching
for h(x)=19



Distance	1	2	4	8	16	32
Node	17	17	23	23	34	48

N_{27} searching
for $h(x)=19$



Finger Table Search

- Number of entries in table: $m-1$
- Number of messages sent for search:
 $O(\log(\text{nodes}))$

So...

- Given a chord-circle
 - We can search for a key
 - We know where to store keys and values
- But wasn't this supposed to be an ad-hoc peer-to-peer network?
- How do we add a node?

Back to Freenet

- Freenet routing similar to but NOT exactly like Chord DHT
- Freenet builds a “small world” topology:
 - Most nodes are not directly connected
 - Most nodes can reach other nodes through a small number of hops
- Small world topology achieved through Markov Chain Monte Carlo (MCMC)

Freenet Insertion

1. Split file into blocks
2. Encrypt each block
3. Compute content hash key (CHK) of each block and insert them
4. Create manifest block containing CHKs for each block and decryption key
5. Insert manifest block

Freenet Lookup

- Key points to manifest
- Manifest contains file metadata and the key of all of the file's blocks
- Only key is used for search, so content of blocks being requested is obscured
- Manifest contains key to decrypt and reconstruct file

Darknet

- Previous description was mostly “OpenNet”
 - Connect to strangers through “seed nodes”
- Darknet operation is also possible:
 - Nodes in Freenet are connected only through real-life social networks (Friend-to-Friend)