Utah Distributed Systems Meetup and Reading Group - Kademlia

JT Olds

Space Monkey
Vivint R&D

January 20, 2015

Outline

Outline

Introduction

Distributed Hash Tables

- Why?
- What options are out there?

Distributed Hash Tables

- Why?
- What options are out there?

Distributed Hash Tables

- Why?
- What options are out there?

Outline

Kademlia Basics

Kademlia Basics

Ids

Every node in the network gets a unique, random id of the same length (160 bits).

```
001110100001...1100100101111
```

Values also get ids of the same length.

Kademlia Basics

Distance

- Every node id has a defined distance from every other node and value id.
- We store values on the nearest k nodes.

Distance

- Every node id has a defined distance from every other node and value id.
- We store values on the nearest k nodes.

Distance

- Every node id has a defined distance from every other node and value id.
- We store values on the nearest k nodes.

Interlude about metrics

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Interlude about metrics

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Interlude about metrics

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Interlude about metrics

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Interlude about metrics

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Interlude: Standard distance metric

1. $|x - y| \geq 0$
2. $|x - y| = 0 \iff x = y$
3. $|x - y| = |y - x|$
4. $|x - z| \leq |x - y| + |y - z|$

Interlude: Standard distance metric

1. $|x - y| \geq 0$
2. $|x - y| = 0 \iff x = y$
3. $|x - y| = |y - x|$
4. $|x - z| \leq |x - y| + |y - z|$

Interlude: Standard distance metric

1. $|x - y| \geq 0$
2. $|x - y| = 0 \iff x = y$
3. $|x - y| = |y - x|$
4. $|x - z| \leq |x - y| + |y - z|$

Interlude: Standard distance metric

1. $|x - y| \geq 0$
2. $|x - y| = 0 \iff x = y$
3. $|x - y| = |y - x|$
4. $|x - z| \leq |x - y| + |y - z|$

Interlude: Standard distance metric

1. $|x - y| \geq 0$
2. $|x - y| = 0 \iff x = y$
3. $|x - y| = |y - x|$
4. $|x - z| \leq |x - y| + |y - z|$

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$
   Why?

Interlude: XOR metric

1. $x \oplus y \geq 0$
2. $x \oplus y = 0 \iff x = y$
3. $x \oplus y = y \oplus x$
4. $x \oplus z \leq x \oplus y + y \oplus z$
   Why? Because:
   - $a \oplus b \leq a + b$
   - $x \oplus z = (x \oplus y) \oplus (y \oplus z)$
   - so $(x \oplus y) \oplus (y \oplus z) \leq x \oplus y + y \oplus z$

id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

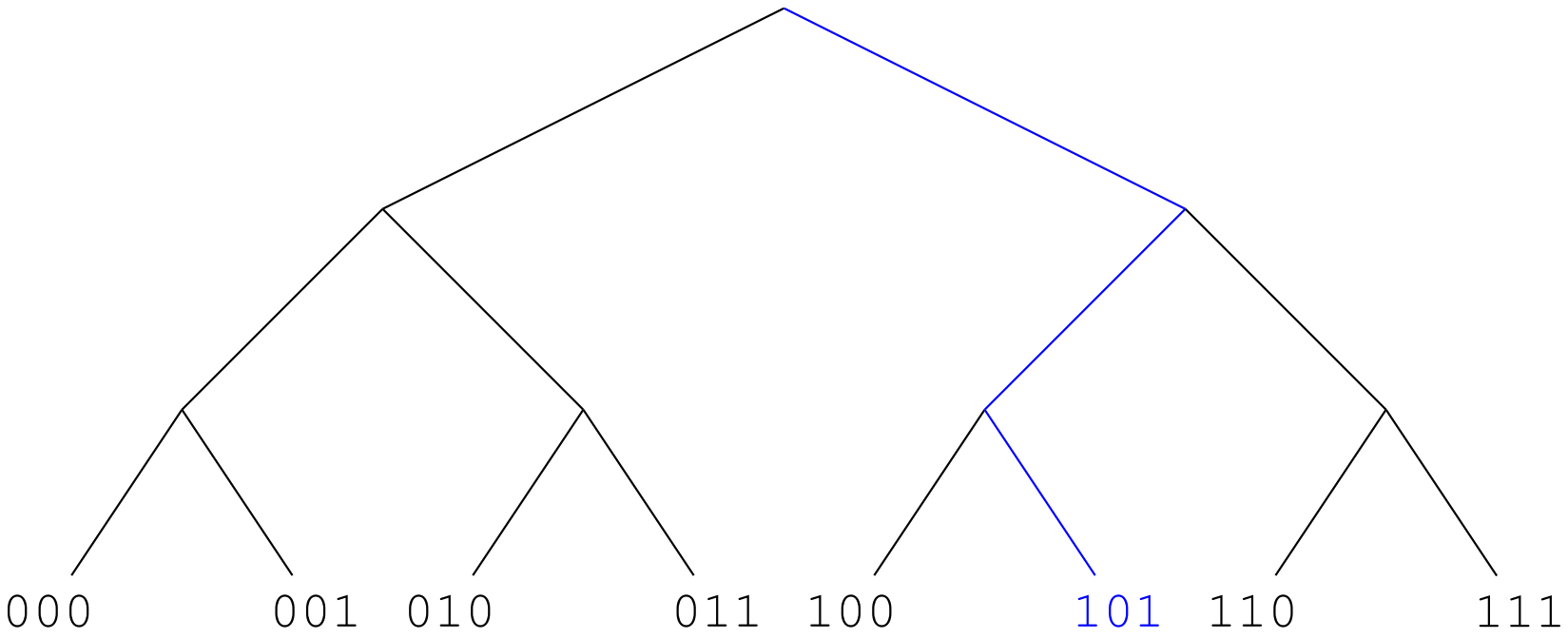
id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

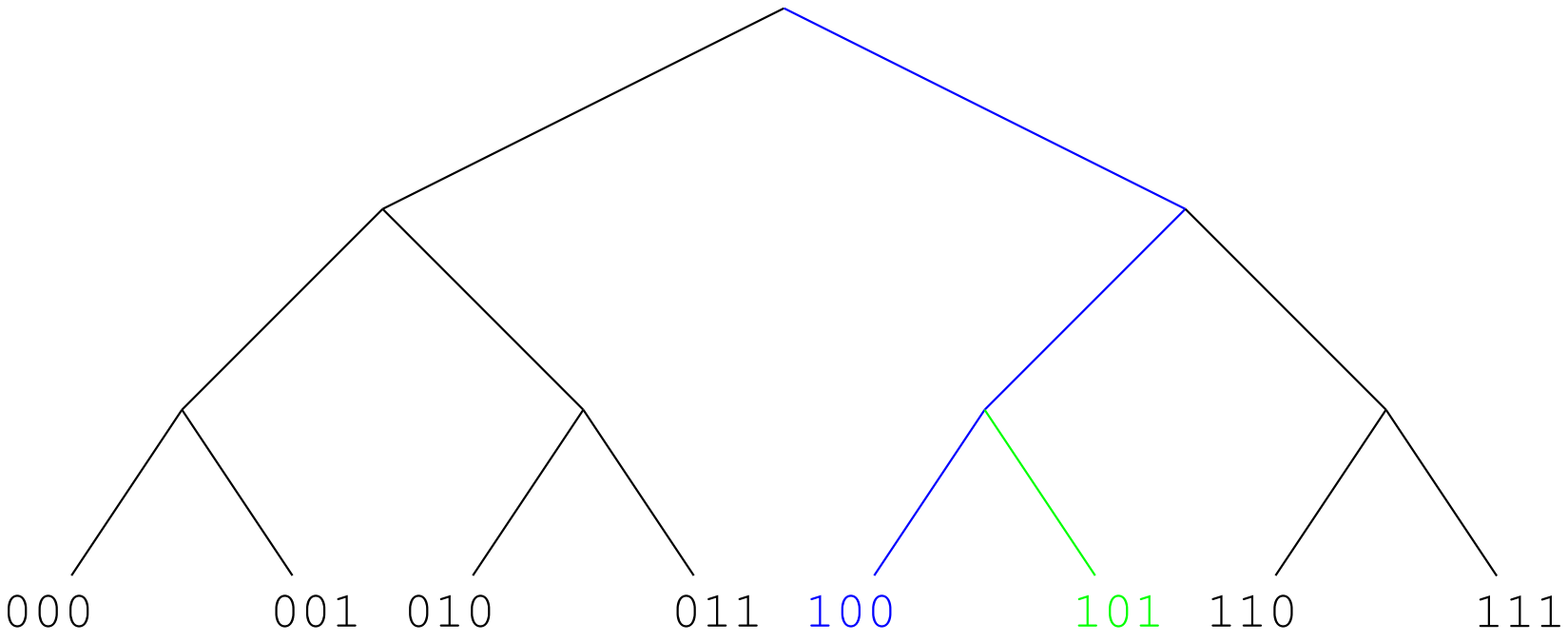
id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

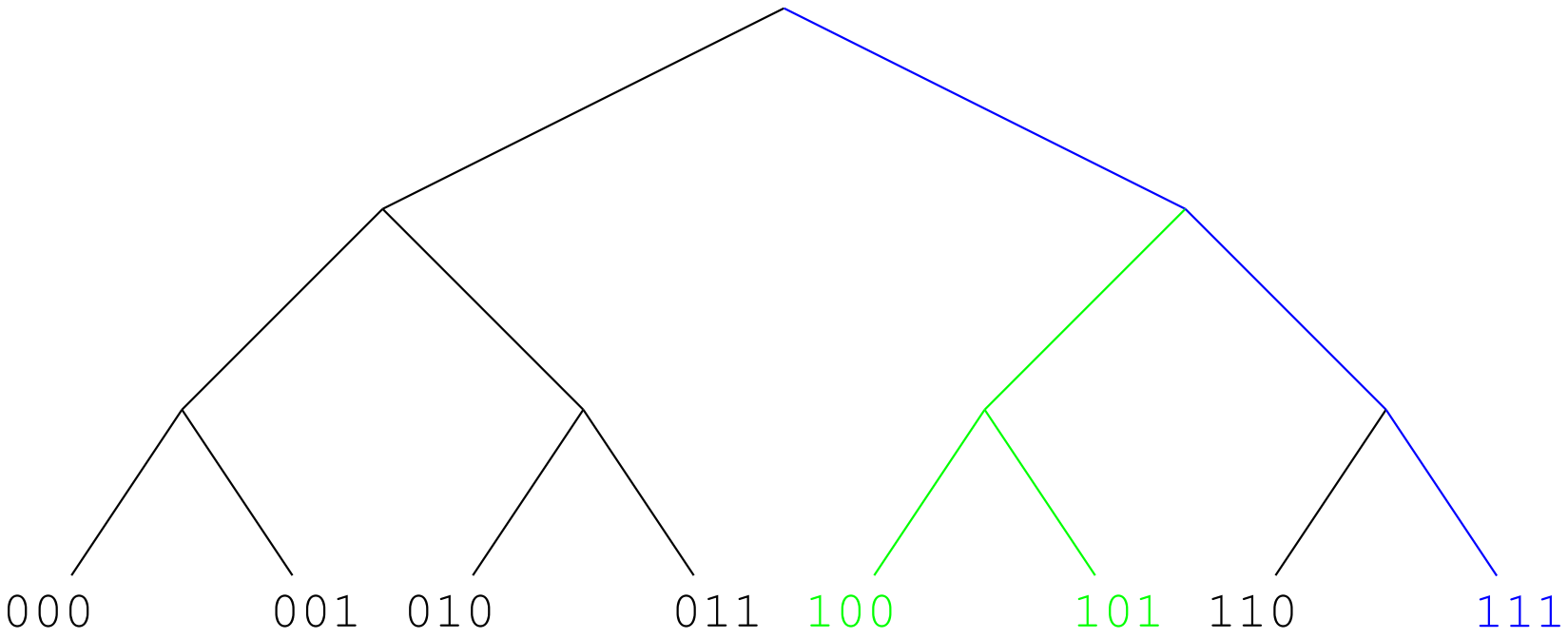
id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

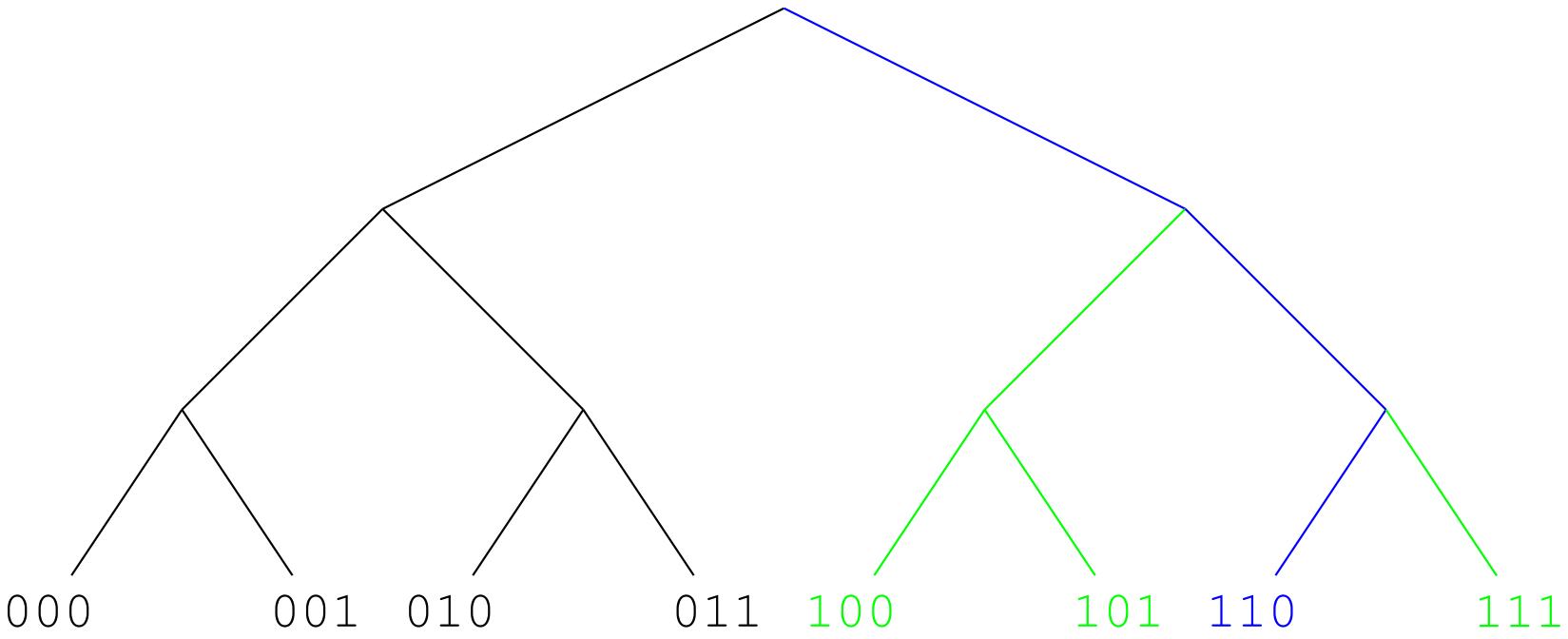
id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

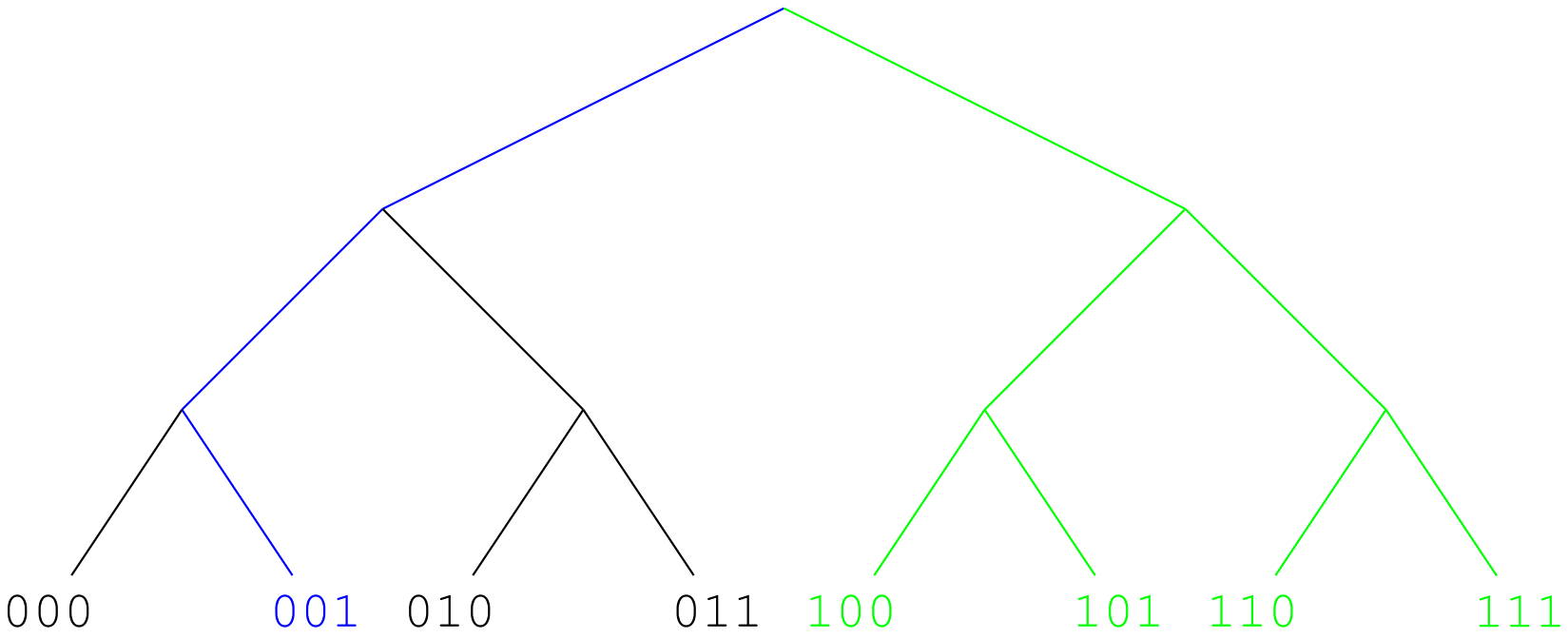
id	euclidian	xor
101	0	0
000	5	5
001	4	4
010	3	7
011	2	6
100	1	1
101	0	0
110	1	3
111	2	2

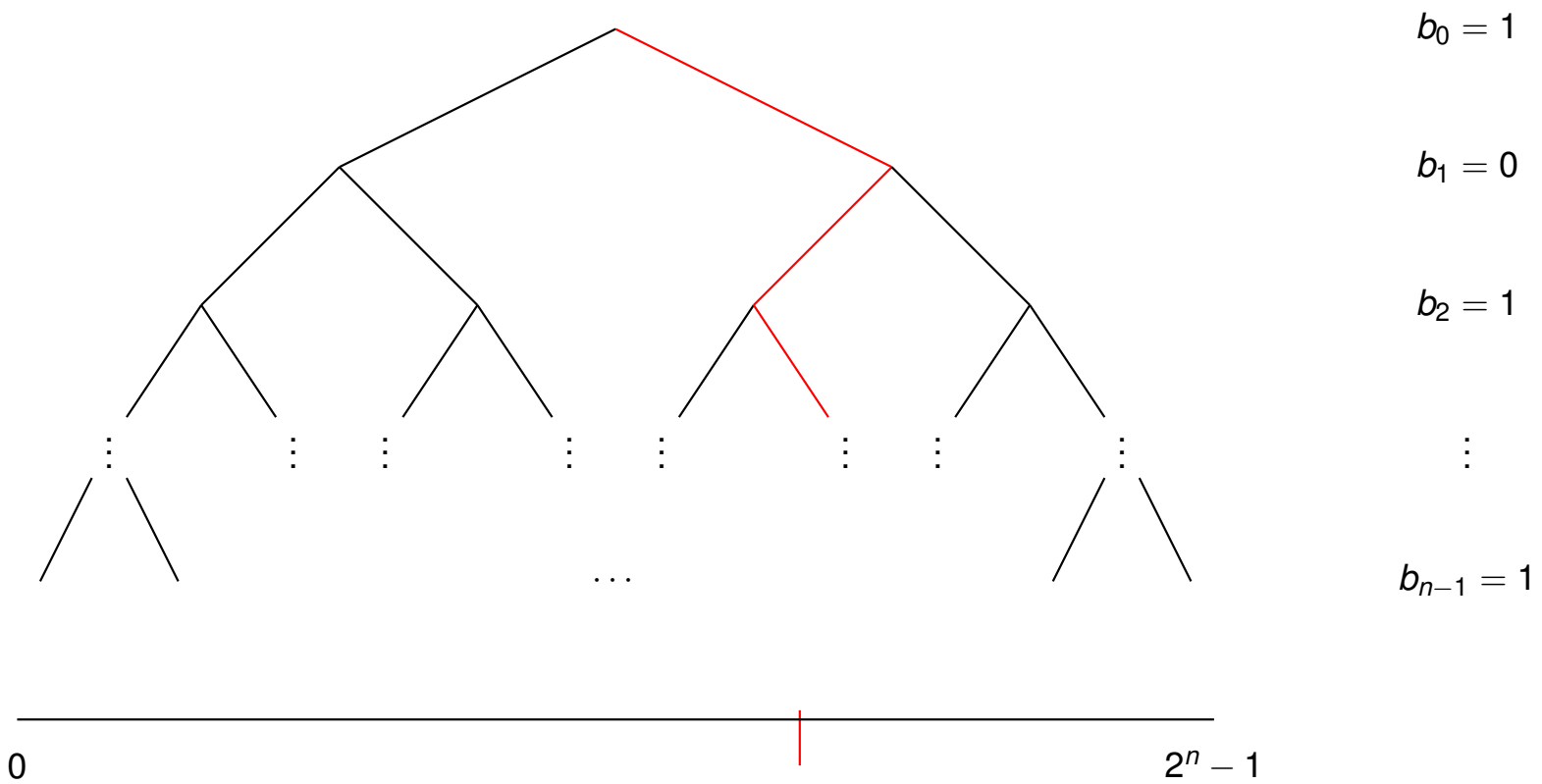

000
001
010
011
100
101
110
111

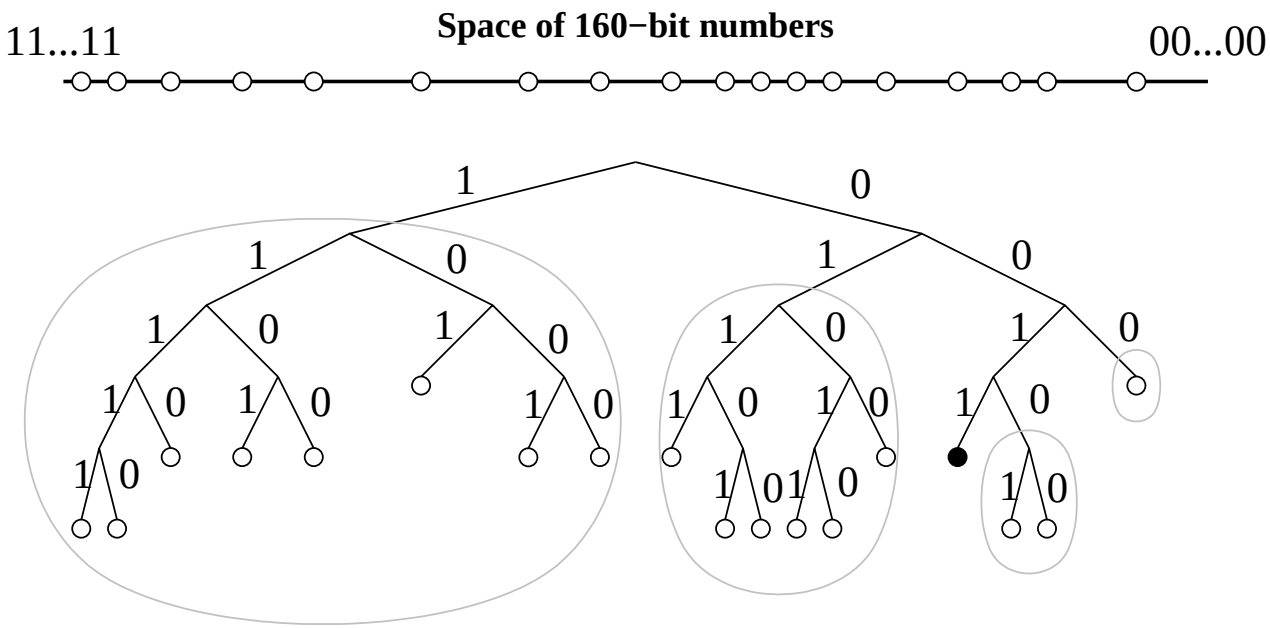

000
001
010
011
100
101
110
111


000
001
010
011
100
101
110
111


000
001
010
011
100
101
110
111


000
001
010
011
100
101
110
111


$b_0 = 1$
$b_1 = 0$
$b_2 = 1$
$b_{n-1} = 1$
0
$2^n - 1$


11...11
Space of 160−bit numbers
00...00

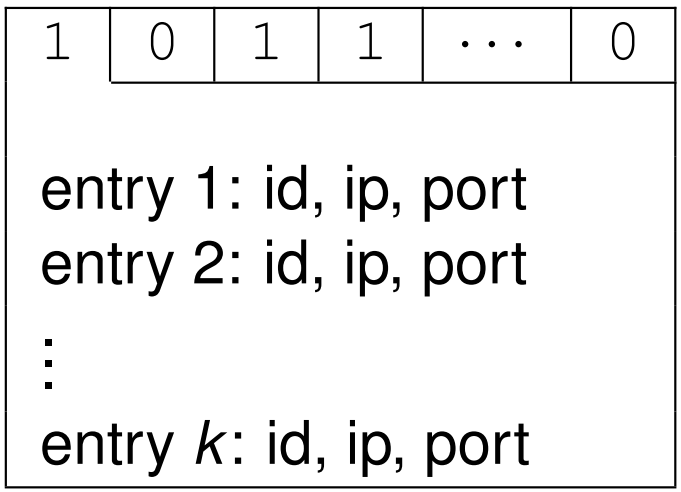
Distance

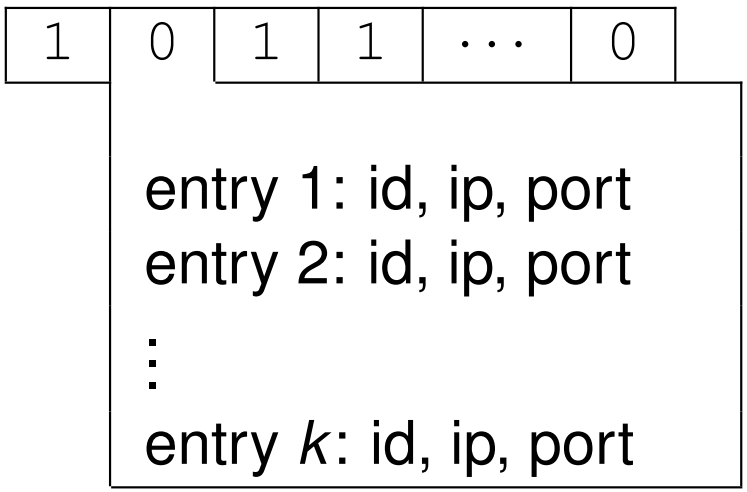
- Every node id has a defined distance from every other node and value id.
- We store values on the nearest k nodes.

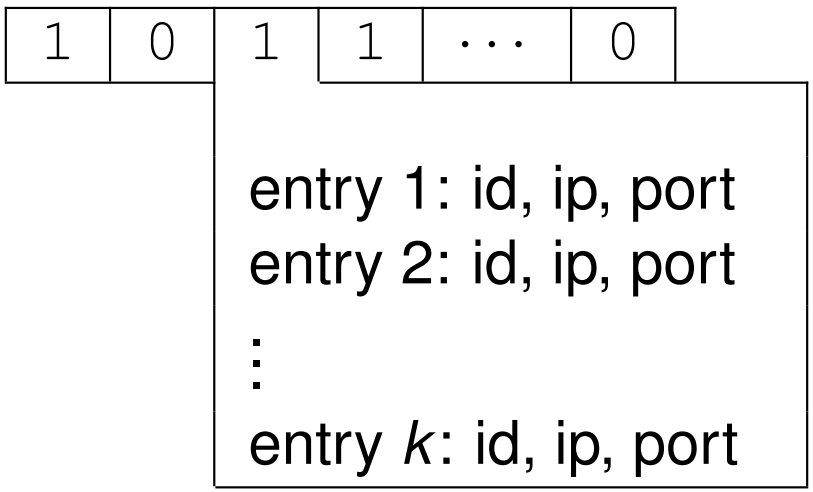
Outline

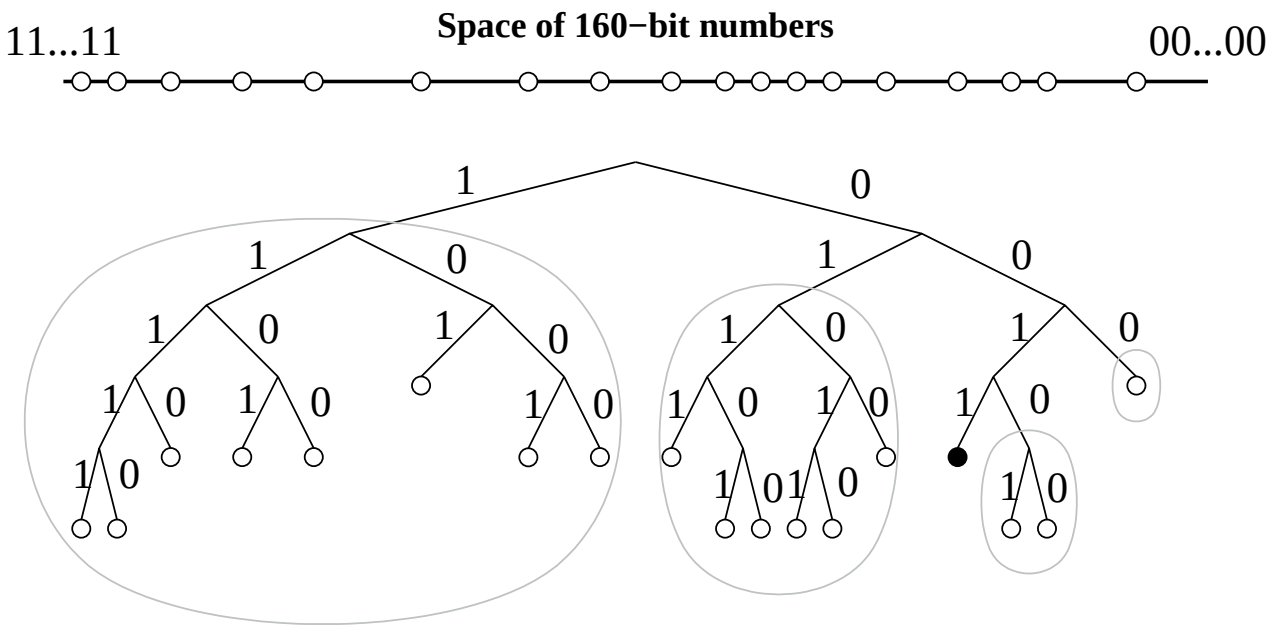
Routing Table

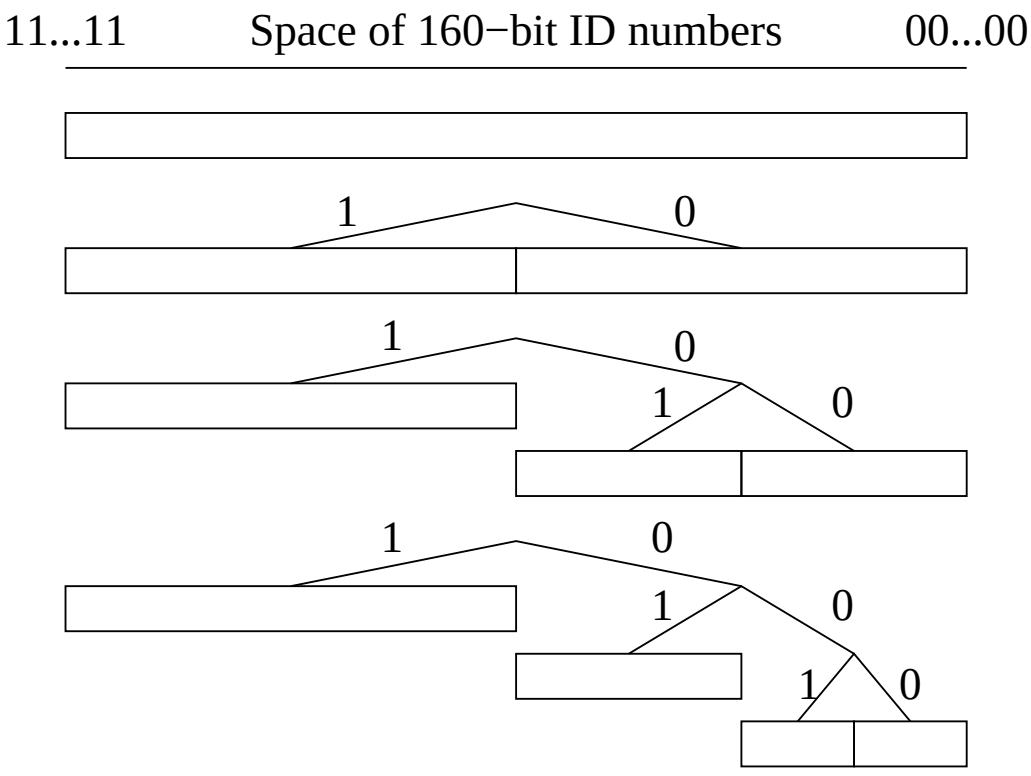
Routing Table


1
0
1
1
· · ·
0
entry 1: id, ip, port
entry 2: id, ip, port
entry k: id, ip, port

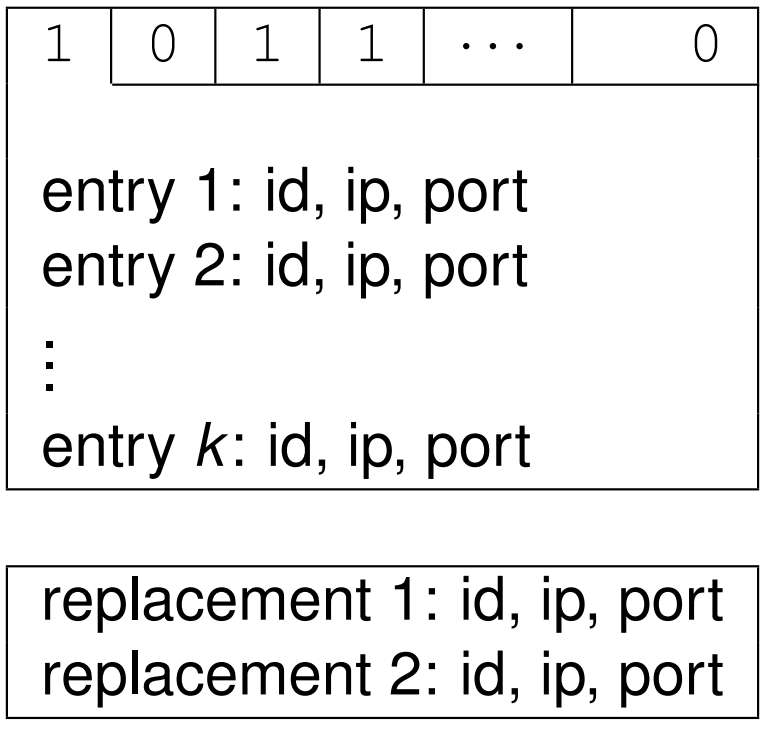

1
0
1
1
· · ·
0
entry 1: id, ip, port
entry 2: id, ip, port
entry k: id, ip, port


1
0
1
1
···
0
entry 1: id, ip, port
entry 2: id, ip, port
entry k: id, ip, port


11...11
Space of 160−bit numbers
00...00


11...11
Space of 160−bit ID numbers
00...00
1
0
1
0
1
0
1
0
1
0

Routing Table


1
0
1
1
···
0
entry 1: id, ip, port
entry 2: id, ip, port
entry k: id, ip, port
replacement 1: id, ip, port
replacement 2: id, ip, port

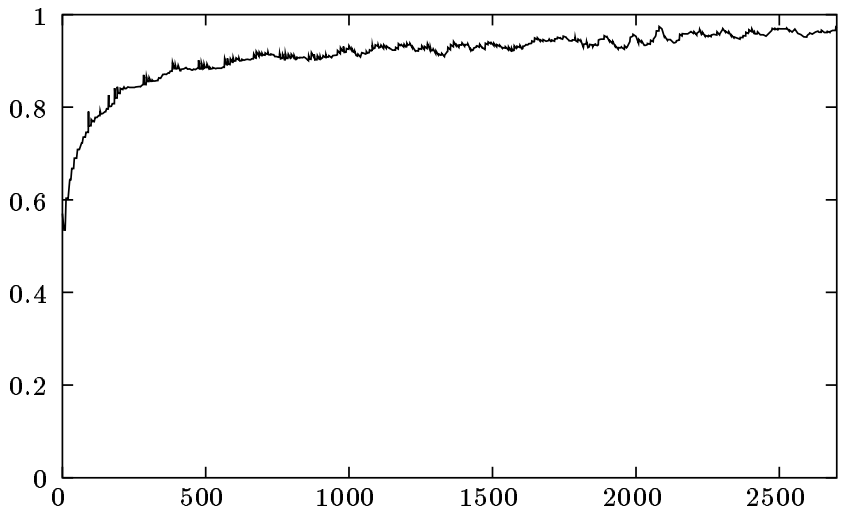


Fig. 3: Probability of remaining online another hour as a function of uptime. The x axis represents minutes. The y axis shows the the fraction of nodes that stayed online at least x minutes that also stayed online at least $x + 60$ minutes.

Outline

Protocol

Protocol

Node endpoints

- `Ping`
- `Store`
- `FindNode`
- `FindValue`

Node endpoints

- `Ping`
- `Store`
- `FindNode`
- `FindValue`

Node endpoints

- `Ping`
- `Store`
- `FindNode`
- `FindValue`

Node endpoints

- `Ping`
- `Store`
- `FindNode`
- `FindValue`

Node endpoints

- `Ping`
- `Store`
- `FindNode`
- `FindValue`

Protocol

Lookups

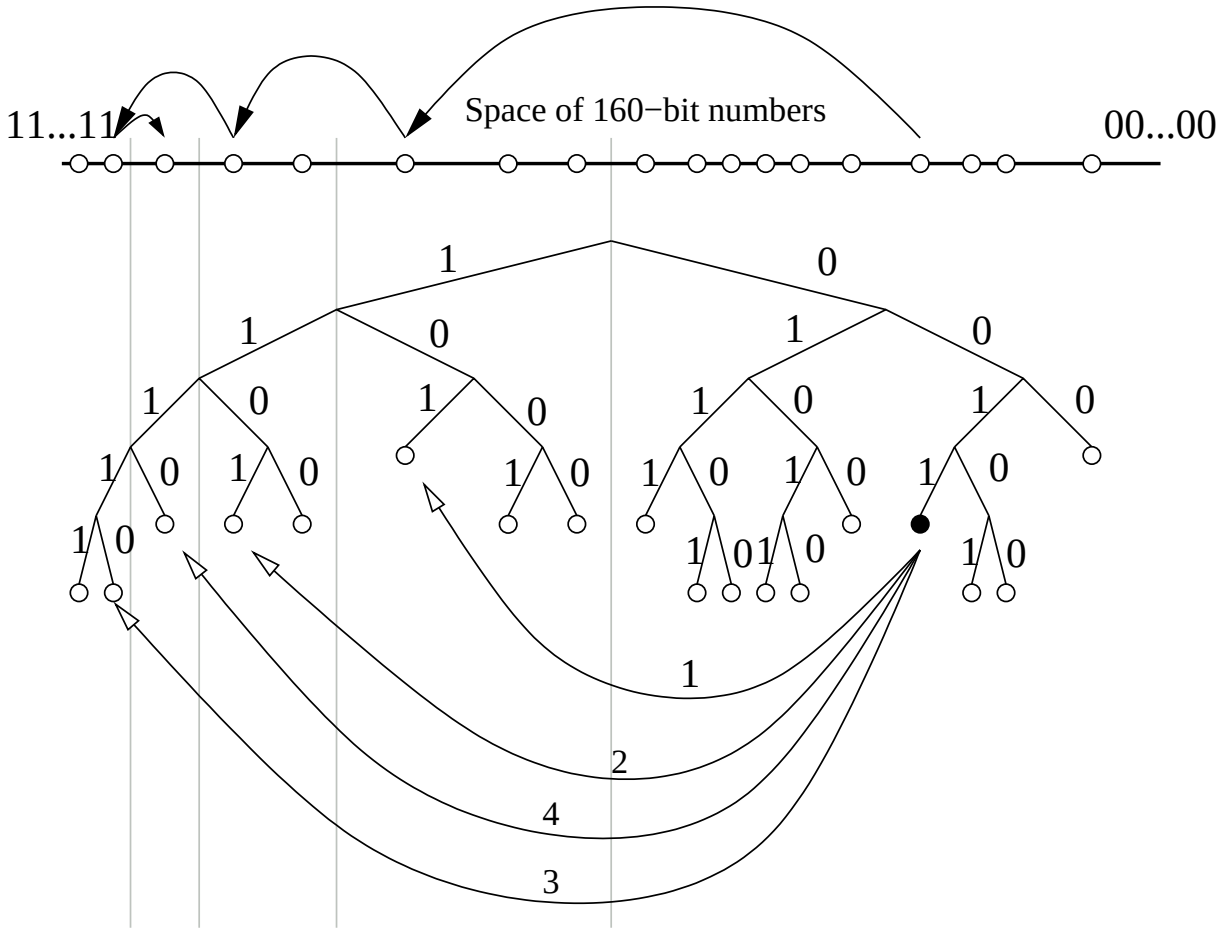
- Kick off α workers
- As long as a k-close node is uncontacted, contact, and add results.

Lookups

- Kick off α workers
- As long as a k-close node is uncontacted, contact, and add results.

Lookups

- Kick off α workers
- As long as a *k*-close node is uncontacted, contact, and add results.


11...11
Space of 160−bit numbers
00...00
1
2
4
3

Protocol

Maintenance

- Bucket refreshes
- Neighbor republishing
- Owner republishing

Maintenance

- Bucket refreshes
- Neighbor republishing
- Owner republishing

Maintenance

- Bucket refreshes
- Neighbor republishing
- Owner republishing

Maintenance

- Bucket refreshes
- Neighbor republishing
- Owner republishing

Outline

Field notes

Field notes

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.

- Address updates are not bidirectional.
- Data is mutable.
- No hot caching.
- No neighbor republishing.
- SSL identification proofs.
- Namespaces.


Space
MONKEY

Space Monkey!

- Distributed Hash Tables
- Consensus algorithms
- Reed Solomon
- Monitoring and sooo much data
- Security and cryptography engineering

Space Monkey!

Come work with us!