The SAFE Network from First Principles

1. XOR Distance and Basic Routing

Erick Lavoie
McGill University,
October 14th, 2014
Motivation
Motivation

• Ubiquitous Surveillance by the NSA as revealed by Edward Snowden
Motivation

• Ubiquitous Surveillance by the NSA as revealed by Edward Snowden
• Insecurity of Cloud Service Providers
  • Apple iCloud Celebrity Photo Leaks
Motivation

• Ubiquitous Surveillance by the NSA as revealed by Edward Snowden

• Insecurity of Cloud Service Providers
  • Apple iCloud Celebrity PhotoLeaks

• Lack of Ownership of Private Data
  • Facebook changing Terms of Service
Motivation

• Ubiquitous Surveillance by the NSA as revealed by Edward Snowden
• Insecurity of Cloud Service Providers
  • Apple iCloud Celebrity Photo Leaks
• Lack of Ownership of Private Data
  • Facebook changing Terms of Service
• Impermanence of Cloud Services
  • Google shutdown of Google Reader even though it had a million users
SAFE Network Proposition

- Protect data and communication with cryptography
- Remove single point of failures (servers) with a self-managing decentralized network made of client-resources
- Handle failure by automatic replication of data
- Enforce cooperation of nodes with decentralized consensus
MaidSafe

• Worked on the problem for 8 years now
• Everything is open source
• Transparent and inclusive dev. process
• Raised over 6 M$US in Bitcoins in April
• Aim to launch the beta version of the network “soon”
My Own PhD

• What?
  • Leverage end-user computational resources for scientific numerical computing

• How?
  • Building a peer-to-peer system with web technologies to run distributed MATLAB code
This Serie of Talks

• Introduce all the core ideas and algorithms behind the system in a technical yet accessible and intuitive manner

• Build an easy to understand and to deploy implementation using web technologies (JavaScript + WebRTC)

• Non-goals:
  • Guaranteed security
  • Performance
General Problem
Data

Machines

Bob

Joe

Alice

Internet

Reliable
Private
Efficient
Scalable
Unstructured Peer-to-Peer System
Unstructured Peer-to-Peer System

Data Machines

Internet

Bob 

Joe

Alice
Unstructured Peer-to-Peer System

Data Machines

Bob

Joe

Alice

Internet

No guarantee on lookup time
Structured Networks
Core Idea #1: Uniform ID Space
Core Idea #1: Uniform ID Space

0
0x00...00

$2^{512} - 1$
0xff...ff

Data

Machines

Bob

Joe

Alice

Internet
Core Idea #1: Uniform ID Space

Data

Machines

Bob

Joe

Alice

0x00...00

0x14...a8

0x71...f2

0xc1...d0

0x07...23

0x51...98

0x92...e4

0xc1...d0

0

2^{512} - 1

0xff...ff
Core Idea #1: Uniform ID Space

Data
- 0x00...00
- 0x07...23
- 0x51...98
- 0x92...e4
- 0xc1...d0

Machines
- Bob
- Joe
- Alice

Range: 0 to $2^{512} - 1$
Core Idea #2: Closeness

0x00...00 0x07...23 0x51...98 0x92...e4 0xc1...d0

2^{512} - 1
0xff...ff

Bob Joe Alice
Core Idea #2: Closeness

- Bob
- Joe
- Alice
Core Idea #2: Closeness

$$d_{euclidean}(a,b) = |a - b|$$
Core Idea #2: Closeness

\[ d_{\text{euclidean}}(a,b) = | a - b | \]
Core Idea #2: Closeness

\[ d_{euclidean}(a,b) = | a - b | \]
Core Idea #2: Closeness

\[ d_{\text{euclidean}}(a,b) = | a - b | \]
Core Idea #2: Closeness

\[ d_{\text{xor}}(a,b) = \text{bitwise\_xor}(a,b) \]
Core Idea #2: Closeness

\[ d_{\text{xor}}(a, b) = \text{bitwise}_\text{xor}(a, b) \]
Core Idea #2: Closeness

\[ d_{\text{xor}}(a,b) = \text{bitwise}_x\text{or}(a,b) \]

\[
\begin{array}{c}
5 \\
^\land 3 \\
\hline
6
\end{array} = \begin{array}{c}
0101 \\
^\land 0011 \\
\hline
0110
\end{array}
\]
Message Routing

0

1

5

9

13

Bob

Joe

Alice

3

7

13

15
Message Routing

3  7  13
Bob Joe Alice
IP Routing

19.24.67.3 -> Bob

19.73.234.7 -> Joe

198.65.3.3 -> Alice
IP Routing

From: 3
To: 13

13

Alice

13

Bob

From: 3
To: 13

3

Joe

7

Diagram showing IP routing with nodes representing individuals (Bob, Joe, Alice) and edges indicating the routing path and numbers indicating the route choice.
IP Routing
IP Routing

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>7*</td>
</tr>
</tbody>
</table>

From: 3  
To: 13  

Bob

Joe

13

Alice
IP Routing

Dest | Node
13 : 7
7  : 7*

From: 3
To:  13

Bob

Joe

Alice
IP Routing
IP Routing

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3*</td>
</tr>
<tr>
<td>13</td>
<td>13*</td>
</tr>
</tbody>
</table>

From: 3
To: 13
IP Routing

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3*</td>
</tr>
<tr>
<td>13</td>
<td>13*</td>
</tr>
</tbody>
</table>

From: 3
To: 13
IP Routing

Bob

Joe

3

7

Alice

From: 3
To: 13
Route summarization

Dest | Node
--- | ----
13  | 13
3   | 3
... | ...

Dest | Node
--- | ----
13  | 7
7   | 7
... | ...

- Bob
- Joe
- Alice
Route summarization

Bob

Joe

5

Bob

John

2

5

8

12

13

Alice
Route summarization

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Bob
- Joe
- John
- Alice

3 → Joe → 7
5 → Joe
2 → John → 8
12 → John
13 → Alice
8 → Alice
Route summarization
Route summarization

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>&gt;10</td>
<td>13</td>
</tr>
</tbody>
</table>

Bob → Joe (7)
Bob → John (5)
John → Joe (2)
Joe → Alice (13)
John → Alice (12)
John → Bob (2)

Route summarization
Route summarization

Dest | Node
--- | ---
24.2.x.x  : 24.2.34.4
...

```
```

```
```
Core Idea #3: Local View

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.2.x.x</td>
<td>24.2.34.4</td>
</tr>
</tbody>
</table>

Route summarization

Bob → Joe

Bob → John

John → 24.2.45.1

Alice → 24.2.34.3
Core Idea #3: Local View

Diagram:

- Bob
- Joe (7)
- Alice (13)
- John (12)
- 3, 5, 2, 8
Core Idea #3: Local View

XOR Distance Structure
Core Idea #3: Local View

XOR Distance Structure

Diagram showing a tree structure with nodes labeled Bob, Joe, John, and Alice, and XOR distances.
Core Idea #3: Local View

XOR Distance Structure
Core Idea #3: Local View

Routing Table Structure
Core Idea #3: Local View

Routing Table Structure

*Joe*

Dest | Node
--- | ---
0111 | 7 (self)
01lx | _
01xx | 5
0xxx | 2
xxxx | 12

Routing Table Structure:

```
0111 7  (self)
01lx  _
01xx  5
0xxx  2
xxxx  12
```
Core Idea #3: Local View

Routing Messages

*Joe*

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>8</td>
</tr>
<tr>
<td>0001</td>
<td>5</td>
</tr>
<tr>
<td>0002</td>
<td>12</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>7</td>
</tr>
<tr>
<td>0012</td>
<td>_</td>
</tr>
<tr>
<td>0100</td>
<td>9</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0102</td>
<td>12</td>
</tr>
<tr>
<td>0110</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>0112</td>
<td>_</td>
</tr>
<tr>
<td>1000</td>
<td>9</td>
</tr>
<tr>
<td>1001</td>
<td>5</td>
</tr>
<tr>
<td>1002</td>
<td>12</td>
</tr>
<tr>
<td>1010</td>
<td>2</td>
</tr>
<tr>
<td>1011</td>
<td>7</td>
</tr>
<tr>
<td>1012</td>
<td>_</td>
</tr>
<tr>
<td>1100</td>
<td>9</td>
</tr>
<tr>
<td>1101</td>
<td>5</td>
</tr>
<tr>
<td>1102</td>
<td>12</td>
</tr>
<tr>
<td>1110</td>
<td>2</td>
</tr>
<tr>
<td>1111</td>
<td>7</td>
</tr>
<tr>
<td>1112</td>
<td>_</td>
</tr>
</tbody>
</table>

Bob

John

Alice

Joe
Core Idea #3: Local View

Routing Messages

*Joe*

Dest | Node
---|---
0111 | 7 (self)
011x | _
01xx | 5
0xxx | 2
13 (1101) | xxxx | 12

Routing Messages

Bob
Joe
John
Alice
Core Idea #3: Local View

Routing Messages

*Joe*
Dest | Node
0111 | 7 (self)
011x  |
01xx  | 5
0xxx  | 2
13 (1101) | xxxx | 12

Routing Messages Diagram: joe node 13 (1101) to john alice (12 13 15) through bob (2 3 5 7 8)
Core Idea #3: Local View

Routing Messages

*Joe*
Dest | Node
---|---
0111 | 7 (self)
011x | _
01xx | 5
0xxx | 2
13 (1101) | xxxx | 12

Routing Messages:

```
Dest  | Node
---|---
0111  | 7 (self)
011x  | _
01xx  | 5
0xxx  | 2
13 (1101) | xxxx | 12
```
Core Idea #3: Local View

Routing Messages

<table>
<thead>
<tr>
<th><em>Joe</em></th>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0111</td>
<td>7 (self)</td>
</tr>
<tr>
<td></td>
<td>011x</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>01xx</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>0xxx</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>xxxx</td>
<td>12</td>
</tr>
</tbody>
</table>

Routing Messages
### Core Idea #3: Local View

**Routing Messages**

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0111</td>
<td>7 (self)</td>
</tr>
<tr>
<td>011x</td>
<td>_</td>
</tr>
<tr>
<td>01xx</td>
<td>5</td>
</tr>
<tr>
<td>0xxx</td>
<td>2</td>
</tr>
<tr>
<td>xxxx</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dest</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1100</td>
<td>12 (self)</td>
</tr>
<tr>
<td>110x</td>
<td>13</td>
</tr>
<tr>
<td>11xx</td>
<td>_</td>
</tr>
<tr>
<td>1xxx</td>
<td>8</td>
</tr>
<tr>
<td>xxxx</td>
<td>7</td>
</tr>
</tbody>
</table>

![Routing Message Diagram](image)
Core Idea #3: Local View

Routing Messages

*Joe*
Dest | Node
0111 7 (self)
011x _
01xx 5
0xxx 2
xxxx 12

*John*
Dest | Node
1100 12 (self)
110x 13
11xx _
lxxx 8
xxxx 7

Routing Messages
Core Idea #3: Local View

Routing Messages

*Joe*
Dest | Node
0111 7 (self)
011x _
01xx 5
0xxx 2
xxxx 12

*John*
Dest | Node
1100 12 (self)
110x 13
11xx _
1xxx 8
xxxx 7

Routing Messages
Core Idea #3: Local View

Routing Messages

*Joe*

Dest | Node
---|---
0111 | 7 (self)
011x | _
01xx | 5
0xxx | 2
xxxx | 12

*John*

Dest | Node
---|---
1100 | 12 (self)
110x | 13
11xx | _
1xxx | 8
xxxx | 7

Routing Messages
Core Idea #3: Local View

Data Ownership
Conclusion

- Uniform ID Space for both machine and data
- XOR distance metric
- Routing table that exploits the implicit hierarchy in the XOR distance between nodes
- --> Simple and efficient way of distributing ownership of resources and allowing communication at the scale of the Internet, without introducing any privileged nodes
Next Episode(s)

• Building the routing tables
• Handling new nodes joining and old nodes leaving (Churn)
• Failures
$d_{circular}(a,b) = (b - a) \mod 15$
\[ d_{\text{circular}}(a, b) = (b - a) \mod 15 \]