Towards a richer set of services in Software-Defined Networking

Roberto Bifulco, Ghassan O. Karame
Roberto.Bifulco@neclab.eu, Ghassan.Karame@neclab.eu

NEC Europe Ltd.,
NEC Laboratories Europe
Heidelberg, Germany
Can we exploit SDN to build network services that are just too complicated to implement in traditional networks?

- Location proofs
- User-defined path
SDN: SOFTWARE-DEFINED NETWORKING
Software Defined Networking

Network OS

"Normal" switches / forwarding devices

© Nick McKeown (Stanford University)
Software Defined Networking

Network OS

1. Open interface to packet forwarding

1.2.3.0/24 Fwd to port 3
1.2.3.0/24 Fwd to port 2
1.2.3.0/24 Fwd to port 3
1.2.3.0/24 Fwd to port 2
This is not a future vision. It’s here

New industry forum assures interoperability
- Define OpenFlow Protocol
- Promote Software-Defined Networking (SDN)
- 7 board members, > 70 regular members

SDN in the Data Center:
NEC’s ProgrammableFlow

Board Members
- Deutsche Telekom
- Facebook
- Google
- Microsoft
- NTT Communications
- Verizon
- Yahoo!

SDN in the backbone:
Google's OpenFlow WAN
Domain Di is **Controlled** by Controller Ci;
Controller Ci is equipped with public/private **key pair**
Uj belonging to Di is equipped with public/private **key pair**
The network Controller and **network components are trusted**
Users want of course to **acquire** new services without being entitled to 😊
Location proof

“Location proofs” consist of a certificate that certifies the presence of a given entity at a certain location at some point in time.
Many services rely on location information
  - Maybe many more will come…
Audio/video streaming, banking, voting, etc.
Current solutions to acquire location proofs are either unreliable (e.g., IP Geolocation) or require ad hoc changes to the network
Exploit SDN to provide location proofs

We provide location proof by:
- guaranteeing that a given **IP address is present at a given location**;
- linking an **identity** to the IP address.

**Required steps**
- Discover network location
- Relate network location to physical location
- Relate network flows to user identity

Roberto
NPoL: overview

Dynamic

<table>
<thead>
<tr>
<th>Location</th>
<th>Entity</th>
<th>Switch</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address..</td>
<td>1.1.1.1</td>
<td>ABC</td>
<td>10</td>
</tr>
</tbody>
</table>

Identity

Roberto

Controller

Request location proof

Trusted location

Provide location proof to third party service

Static

<table>
<thead>
<tr>
<th>Location</th>
<th>Switch</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address, City...</td>
<td>ABC</td>
<td>10</td>
</tr>
</tbody>
</table>

ANTI-SPOOFING

DSLAM

HGW

S
NPoL: attacks

<table>
<thead>
<tr>
<th>Location</th>
<th>Entity</th>
<th>Switch</th>
<th>Port</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.1.1.1</td>
<td>ABC</td>
<td>10</td>
<td>Roberto</td>
</tr>
</tbody>
</table>

Roberto moved to location B

Controller

is it possible?

Roberto moved to location B
User-defined Path

A network path which obeys to user specific constraints
Why UdP?

- Untrusted ISPs
  - Some authoritarian countries hijacking traffic

- Improved QoS/dependability required by some applications
  - E.g., telemedicine

Hijacked traffic
UdP: overview

Untrusted network

Controller

Controller

Controller

Controller
UdP: packet forwarding

<table>
<thead>
<tr>
<th>L2 DST</th>
<th>L2 SRC</th>
<th>L3 SRC</th>
<th>L3 DST</th>
<th>ACTIONs</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>h(CERT)</td>
<td>IP-A</td>
<td>IP-B</td>
<td>Set header (Action list); Fwd-to: 2</td>
</tr>
</tbody>
</table>

**Action list**

<table>
<thead>
<tr>
<th>Action pointer</th>
<th>ACTIONs</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 1 4 2</td>
<td>Increment pointer; Based on pointer;</td>
</tr>
</tbody>
</table>

**Code | Action**
--- | ---
1   | Fwd-to: 11
2   | Fwd-to: 5
... | ...

*Restore original header*
UdP: Scalability
Implementation and evaluation

Controller

Control network

Server 1

Server 2

NEC ProgrammableFlow Switches
PF5240

Controller

C

A

B

Server 1

Server 2

RIT (ms)

Packet ID

static
dynamic
Conclusions

- SDN enables the creation of new services by exploiting the already deployed network infrastructure;
  - We implemented and evaluated two new network services: NPoL and UdP;

- Both services were implemented on a SDN testbed composed of hardware switches (OpenFlow-based)

- The solutions scalability has been validated using real traffic traces from both access and core networks

- NPoL and UdP are two examples, what’s next?
  - The network is not just a cloud (anymore)!
Empowered by Innovation

NEC
Asking for location proof

\[ U_j \]

\[ M_1 \leftarrow \{IP_j \parallel pk_j \parallel T1\} \]

\[ M_1 \parallel \text{Sig}(M_1, sk_j) \]

\[ \text{Verify}(M_1 \parallel \text{Sig}(M_1, sk_j)) \]

\[ L \leftarrow \text{Lookup}(IP_j) \]

\[ M_2 \leftarrow (IP_j \parallel pk_j \parallel T2 \parallel L) \]

\[ M_2 \parallel \text{Sig}(M_2, skc_i) \]
Asking for UdP

Uj

Controller i

M1 ← \{IP_a || pk_a || IP_b || pk_b || NPoL_b || CONSTR || T1\}

M1 || Sig(M1, sk_a) || Sig(M1, sk_b)

Verify(M1 || Sig(M1, sk_a) || Sig(M1, sk_b))

Verify(NPoL_b)

Check CONSTR applicability

M2 ← ('OK' || M1 || T2 || R || ACK_s)

M2 || Sig(M2, skc_i)
Software Defined Networking

1. Open interface to packet forwarding