NOX

A Network Operating System for OpenFlow
The Genesis of Nox

SANE
- Network As A FileSystem
- Host tracking
- routing
- topology
- Openflow v -2.0

Ethane
- Policy Compiler
- Host tracking
- routing
- topology
- Openflow v -1.0

Nox
- Host tracking
- routing
- topology
- Openflow
New functions require new hardware (e.g. VNTag, TrustSec)
Innovation happens at hardware design speeds
No support for network-wide control or high-level abstractions
The Goal of Nox

Network-wide “OS”

API

Control

Datapath

\[ F_1, F_2, \ldots, F_n \]
Nox Components

- **Network Application Services**
  - New functions as software services

- **Nox Controller - Network OS**
  - Open platform for network control
  - Provides system-wide abstractions
  - Turn networking into a software problem

- **OpenFlow**

Diagram:
- Diagram showing the components of Nox Components:
  - Nox Controller
  - Network Application Services
  - Nox Controller - Network OS
  - OpenFlow Switches

Legend:
- $F_1$, $F_2$, ..., $F_n$
Objectives (with retrospective)

- Generality
- Performance
- Robustness
- Simplicity
A new switch has joined!

A packet has arrived!

Port status Has changed!

C++

Python

Openflow stack

Core

Threading/IO/Event Harness

A new switch has joined!

A packet has arrived!

Port status Has changed!
A new link was detected!

LLDP packet received!

Event

Send out LLDP packet

Core

Threading/IO/Event Harness

Openflow stack

Topology Discovery

C++

Python
NOX Overview

Core
- Openflow stack
- Threading/IO/Event Harness

Routing
- C++

Topology Discovery
- Python

Link Detected

Packet in Switch join

Switches connected by links.
Component Architecture

Components
- Statistics Tracking
- Policy Engine
- Routing
- Host tracking
- Storage
- Python
- Directory Management
- Web Services
- Topology discovery

CORE
- Cooperative threading
- Event Harness
- OpenFlow API
- Asynchronous I/O
- Socket I/O
- File I/O

Component API
- UI
Example Uses
- Declare security policy in high-level language
- Enforce policy network-wide (per-flow)
- Supports
  - Admission/Access control policy
  - Directory integration
  - Network monitoring and logging

Basic Idea: Use dynamic policy compiler to manage all connectivity on network
Distributing Policy Lookup

Controller

Policy File

Policy Namespace (address Bindings)

Compiler

Packet Classification

User Login

Host Join

Switch Join

Authentication Traffic

packets
Network Access Control

Communication Rules

- allow() <= protocol('arp')
- automatic authentication rule
- http_redirect() <= usrc('discovered;unauthenticated') ^ tpdst(80) ^ hsrc('Built-in;gobi')
- http_redirect() <= usrc('discovered;unauthenticated') ^ tpdst(80) ^ hsrc('Built-in;badwater')
- http_redirect() <= usrc('discovered;unauthenticated') ^ tpdst(80) ^ in_group('Built-in;laptops', 'HSRC')

- allow() <= True

Protected Rule

This rule is vital to system integrity. To edit, you must first unselect the protected toggle below.

Metadata

Description: 
Comment: 
Exception: 

NOX Network | 3 Active Admins
SNAC Deployments

- Stanford CS (~400 devices)
- Government (multiple deployments)
- Medical University (~200 users)
- Small business (~50 devices)
Example: Network Virtualization

- DMZ Servers
- IPS
- AAA
- 802.1X
- Sniffer
**Example: Network Virtualization**
Example: Network Virtualization
Open vSwitch + DVS Controller

- **Open vSwitch** (http://vswitch.org)
  - Full featured virtual network switch
  - Handles dynamics of virtual environment
  - Exports standard visibility and control interfaces

- **DVS Controller**
  - Central switch controller
  - Provides
    - *Distributed CLI*
    - *Policy declaration*
    - *Network visibility*
    - *QoS/RSPAN policy*
    - *Support VM Mobility*

---

**I proactively push rules on host join!**

**I speak OpenFlow!**

**DVS Controller**

**openflow**

**Distributed vSwitch**
## Virtual Network Console

### Server Statistics
- **vNetManager**
  - **Up Time:** 15 min
  - **CPU Load:** 0%

### Network Statistics
- **Network**
  - **Resource Pools:** 1
  - **Xen Servers:** 2
  - **Networks:** 7
  - **VMs:** 6

### Recent Network Events
<table>
<thead>
<tr>
<th>P#</th>
<th>Date/Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>2</td>
<td>06/10/09 22:53:16</td>
<td>switch 'authenticated' joined the network.</td>
</tr>
<tr>
<td>3</td>
<td>06/10/09 22:53:22</td>
<td>host 'Demo Etcsh 4.0 (1)' joined the network.</td>
</tr>
<tr>
<td>3</td>
<td>06/10/09 22:53:22</td>
<td>host 'Demo Etcsh 4.0 (1)' authenticated 46:fe:97:83:57:86 on location 'authenticated'.</td>
</tr>
<tr>
<td>3</td>
<td>06/10/09 22:53:22</td>
<td>host 'Demo Controller' joined the network.</td>
</tr>
<tr>
<td>3</td>
<td>06/10/09 22:53:22</td>
<td>host 'Demo Controller' authenticated 02:44:a7:4d:a7:54 on location 'authenticated'.</td>
</tr>
</tbody>
</table>

### Packet Rate (packets/sec) of the Top 5 VMs

### Connection Rate (connections/sec) of the Top 5 VMs
Other Projects Using Nox

- Home networking
- Control virtualization
- Datacenter policy
- Network-wide monitoring
- TE/WAN
- Mobility

- Many others ...
Nicira’s Role Going Forward

• Will continue to support Nox
  • Questions, bugfixes

• Will continue to develop tools useful for deployment

• Working on next-gen Nox (4th gen system)
Thank You

http://noxrepo.org