Scalable Flow-Based Networking with DIFANE

Minlan Yu
Princeton University

Joint work with Mike Freedman, Jennifer Rexford and Jia Wang
What’s DIFANE?

• Traditional networking
  – Easy to manage
  – Supported policies
  – Distributed
  – Scalability remains a challenge

DIFANE:
A scalable way to apply fine-grained policies in enterprises
Flexible Policies in Enterprises

• **Access control**
  – Drop packets from malicious hosts

• **Customized routing**
  – Direct Skype calls on a low-latency path

• **Measurement**
  – Collect detailed HTTP traffic statistics
Flow-based Switches

• Install rules in flow-based switches
  – Store rules in high speed memory (TCAM)
• Perform simple actions based on rules
  – Rules: Match on bits in the packet header
  – Actions: Drop, forward, count
Challenges of Policy-Based Management

• Policy-based network management
  – Specify *high-level policies* in a management system
  – Enforce *low-level rules* in the switches

• Challenges
  – Large number of hosts, switches and policies
  – Limited TCAM space in switches
  – Support host mobility
  – No hardware changes to commodity switches
Pre-install Rules in Switches

- Problems:
  - No host mobility support
  - Switches do not have enough memory
Install Rules on Demand (Ethane, NOX)

- Buffer and send packet header to the controller
- Install rules
- Forward
- First packet misses the rules

• Problems:
  – Delay of going through the controller
  – Switch complexity
  – Misbehaving hosts
DIFANE: Combining Proactive & Reactive

<table>
<thead>
<tr>
<th>Features</th>
<th>Install rules</th>
<th>Proactive</th>
<th>Reactive(Ethane)</th>
<th>DIFANE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DIFANE Architecture
(two stages)

Distributed Flow Architecture
for Networked Enterprises
Stage 1

The controller *proactively* generates the rules and distributes them to authority switches.
Partition and Distribute the Flow Rules

Controller

Distribute partition information

Authority Switch A

Authority Switch B

Authority Switch C

Flow space

accept

reject

Ingress Switch

Authority Switch A

Authority Switch B

Authority Switch C

Egress Switch
Stage 2

The authority switches keep packets always in the data plane and *reactively* cache rules.
A slightly longer path in the data plane is faster than going through the control plane.
Locate Authority Switches

• Partition information in ingress switches
  – Using a small set of coarse-grained wildcard rules
  – ... to locate the authority switch for each packet

• Distributed directory service but not DHT
  – Hashing does not work for wildcards
  – Keys can have wildcards in arbitrary bit positions

<table>
<thead>
<tr>
<th>Authority Switch A</th>
<th>AuthoritySwitch B</th>
<th>Authority Switch C</th>
</tr>
</thead>
<tbody>
<tr>
<td>X:0-1 Y:0-3</td>
<td>X:2-5 Y:0-1</td>
<td>X:2-5 Y:2-3</td>
</tr>
</tbody>
</table>

A ➔ B ➔ C
Packet Redirection and Rule Caching

- Ingress Switch
- Cache Rules
- Partition Rules
- Feedback: Cache rules
- Redirect
- Authority Switch
- Auth. Rules
- Forward
- Egress Switch
- Hit cached rules and forward
Three Sets of Rules in TCAM

<table>
<thead>
<tr>
<th>Type</th>
<th>Priority</th>
<th>Field 1</th>
<th>Field 2</th>
<th>Action</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Rules</td>
<td>110</td>
<td>00**</td>
<td>001*</td>
<td>Forward Trigger cache manager</td>
<td>Infinity</td>
</tr>
<tr>
<td>Authority Rules</td>
<td></td>
<td></td>
<td></td>
<td>DROP, Trigger cache manager</td>
<td></td>
</tr>
<tr>
<td>Partition Rules</td>
<td>15</td>
<td>0***</td>
<td>000*</td>
<td>Redirect to auth. switch</td>
<td></td>
</tr>
</tbody>
</table>

- **In ingress switches** *reactively* installed by authority switches
- **In authority switches** *proactively* installed by controller
- **In every switch** *proactively* installed by controller
Cache Rules

DIFANE Switch Prototype
Built with OpenFlow switch

Control Plane

Recv Cache Updates
Only in Auth. Switches

Send Cache Updates

Cache Manager

Just software modification for authority switches

Notification
Caching Wildcard Rules

• Overlapping wildcard rules
  – Cannot simply cache matching rules
Caching Wildcard Rules

- **Multiple authority switches**
  - Contain independent sets of rules
  - Avoid cache conflicts in ingress switch
Partition Wildcard Rules

• Partition rules
  – Minimize the TCAM entries in switches
  – Decision-tree based rule partition algorithm

Cut B is better than Cut A

Cut B

Cut A

R2
R1
R3
R4
## Handling Network Dynamics

<table>
<thead>
<tr>
<th>Network dynamics</th>
<th>Cache rules</th>
<th>Authority Rules</th>
<th>Partition Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy changes at controller</td>
<td>Timeout</td>
<td>Change</td>
<td>Mostly no change</td>
</tr>
</tbody>
</table>

| Topology changes at switches | No change | No change | Change |
| Host mobility               | Timeout   | No change | No change |
Prototype Evaluation

• Evaluation setup
  – Kernel-level Click-based OpenFlow switch
  – Traffic generators, switches, controller run on separate 3.0GHz 64-bit Intel Xeon machines

• Compare delay and throughput
  – NOX: Buffer packets and reactively install rules
  – DIFANE: Forward packets to authority switches
Delay Evaluation

• Average delay (RTT) of the first packet
  – NOX: 10 ms
  – DIFANE: 0.4 ms

• Reasons for performance improvement
  – Always keep packets in the data plane
  – Packets are delivered without waiting for rule caching
  – Easily implemented in hardware to further improve performance
Peak Throughput

- One authority switch; Single-packet flow

DIFANE further increases the throughput linearly with the number of authority switches.
Scaling with Many Rules

• How many authority switches do we need?
  – Depends on total number of rules
  ... and the TCAM space in these authority switches

<table>
<thead>
<tr>
<th></th>
<th>Campus</th>
<th>IPTV</th>
</tr>
</thead>
<tbody>
<tr>
<td># Rules</td>
<td>30K</td>
<td>5M</td>
</tr>
<tr>
<td># Switches</td>
<td>1.7K</td>
<td>3K</td>
</tr>
<tr>
<td>Assumed Authority</td>
<td>160 KB</td>
<td>1.6 MB</td>
</tr>
<tr>
<td>Switch TCAM size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>5 (0.3%)</td>
<td>100 (3%)</td>
</tr>
<tr>
<td># Authority Switches</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stepping back ...
Distributed or Centralized?

- **Distributed amongst the network elements**
  - All functions in switches

- **logically-centralized in the management system**
  - OpenFlow/NOX
  - DIFANE
    - Controller is still in charge
    - Switches host a distributed directory of the rules
Thanks!