Abstractions for Network Update

Nate Foster
Mark Reitblatt
Jen Rexford
Cole Schlesinger
Dave Walker
Updates Happen

Network Updates
- Maintenance
- Failures
- ACL Updates
Updates Happen

Network Updates
- Maintenance
- Failures
- ACL Updates

Desired Invariants
- No black-holes
- No loops
- No security violations
Network Updates Are Hard
Network Updates Are Hard
Prior Work

Consensus Routing

Reliable BGP

Graceful Migration

Seamless Migration
Prior Work

Consensus Routing

Graceful Migration

Reliable BGP

Seamless Migration
Network Update Abstractions

Goal
• Tools for whole network update

Our Approach
• Develop update abstractions
• Endow them with strong semantics
• Engineer efficient implementations
Example: Distributed Access Control

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Traffic Flow Diagram:
- Traffic flows from F1 to F2 to F3.
- Security Policy:
  - Allow Web traffic.
  - Drop Non-web traffic.
  - Allow Any traffic.
Example: Distributed Access Control

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎩 Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>🌧 Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>🎯 Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:
- Web: ✓, ✗
- Non-web: ✗
- Any: ✓

Traffic:
- F1 ➔ F2 ➔ F3

Clouds:
- 🎩 ➔ F1, F2
- 🎩 ➔ F3

Thursday, August 16, 12
Example: Distributed Access Control

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Traffic

Source

F1, F2

F3

Web: ✓
∗: ✗
Example: Distributed Access Control

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Security Policy

Traffic
Naive Update

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Order

F1
F2
F3

Traffic

Thursday, August 16, 12
Naive Update

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td>Allow</td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td>Drop</td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>

Order

F2
F3
I

Traffic

7

Thursday, August 16, 12
Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Order

F3

Traffic
Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Naive Update

Order

Traffic
Naive Update

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Order

Traffic 7
Naive Update

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>🇺🇸</td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td>🇧🇷</td>
<td>Non-web</td>
<td>Drop</td>
</tr>
<tr>
<td>🇮🇳</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

Order

Traffic

Thursday, August 16, 12
Naive Update

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Order

Traffic: F1 ➔ F3, F1 ➔ F2
Use an Abstraction!
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

I ➔ !F1, F2 ➔ !F3

Web: ✓ *: ✗

Web: ✓ *

*: ✓

Thursday, August 16, 12
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>🕵️‍♂️</td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td>🕵️‍♂️</td>
<td>Non-web</td>
<td>Drop</td>
</tr>
<tr>
<td>🕵️‍♂️</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

Diagram:
- F1 connects to F2 and F3.
- F2 connects to F3.
- Web traffic is allowed from F1 to F2 and F3.
- Non-web traffic is dropped from F1 to F2.
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Traffic
Atomic Update?

Security Policy

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Traffic:
- Web: ✓, ✚:
- Non-web: ✖:
- Any: ✓, ✚:

F1 ➔ F1
F1 ➔ F2, F3
F2 ➔
F3 ➔ F3

Thursday, August 16, 12
Per-Packet Consistent Updates

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Allow</td>
<td></td>
</tr>
<tr>
<td>Non-web</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>

Each packet processed with old or new configuration, but not a mixture of the two.
Universal Property Preservation

**Theorem:** Per-packet consistent updates preserve all trace properties.

**Trace Property**
Any property of a *single* packet’s path through the network.

**Examples of Trace Properties:**
Loop freedom, access control, waypointing ...

**Trace Property Verification Tools:**
Anteater, Header Space Analysis, ConfigChecker ...
**Corollary:** To check an invariant, verify the old and new configurations.

**Verification Tools**
- Anteater [SIGCOMM ’11]
- Header Space Analysis [NSDI ’12]
- ConfigChecker [ICNP ’09]
MECHANISMS
2-Phase Update

Overview
• Runtime instruments configurations
• Edge rules stamp packets with version
• Forwarding rules match on version

Algorithm (2-Phase Update)
1. Install new rules on internal switches, leave old configuration in place
2. Install edge rules that stamp with the new version number
2-Phase Update in Action
2-Phase Update in Action
2-Phase Update in Action
2-Phase Update in Action
2-Phase Update in Action

Traffic

15
Optimized Mechanisms

Optimizations

- Extension: strictly adds paths
- Retraction: strictly removes paths
- Subset: affects small # of paths
- Topological: affects small # of switches

Runtime

- Automatically optimizes
- Power of using abstraction
Subset Optimization

![Diagram showing traffic flow and subset optimization]
Subset Optimization

Traffic

Thursday, August 16, 12
Subset Optimization
Subset Optimization

![Diagram showing traffic flow through F1, F2, and F3 with subset optimization symbols indicating Web, *, and ✓.](attachment:image.png)
Correctness

**Question:** How do we convince ourselves these mechanisms are correct?

**Solution:** We built an operational semantics, formalized our mechanisms and proved them correct.

**Example:** 2-Phase Update

1. Install new rules on internal switches, leave old configuration in place
2. Install edge rules that stamp with the new version number

} Unobservable

} One-touch
Correctness

**Question**: How do we convince ourselves these mechanisms are correct?

**Solution**: We built an operational semantics, formalized our mechanisms and proved them correct.

**Example**: 2-Phase Update

1. Install new rules on internal switches, leave old configuration in place
2. Install edge rules that stamp with the new version number

\{ \}

**Unobservable**

\{ \}

**One-touch**

**Theorem**: Unobservable + one-touch = per-packet.
IMPLEMENTATION & EVALUATION
Implementation

Runtime
• NOX Library
• OpenFlow 1.0
• 2.5k lines of Python
• `update(config, topology)`
• Uses VLAN tags for versions
• Automatically applies optimizations

Verification Tool
• Checks OpenFlow configurations
• CTL specification language
• Uses NuSMV model checker
**Evaluation**

**Question:** How much extra rule space is required?

**Setup**
- Mininet VM

**Applications**
- Routing and Multicast

**Scenarios**
- Adding/removing hosts
- Adding/removing links
- Both at the same time

**Topologies**
- Fattree
- Small-world
- Waxman
Results: Routing Application

- Fattree
- Small-world
- Waxman

Full
Subset

22
Results: Routing Application

Worst-Case Rule Overhead

- Fattree
- Small-world
- Waxman

Thursday, August 16, 12
Results: Routing Application

Worst-Case Rule Overhead

- Full
- Subset

WRAP UP
Conclusion

Update abstractions
• Per-packet
• Per-flow

Mechanisms
• 2-Phase Update
• Optimizations

Implementation
• Runtime
• Verifier

Formal model
• Network operational semantics
• Universal property preservation
Thank You!

Collaborators
Shrutarshi Basu (Cornell)
Arjun Guha (Cornell)
Stephen Gutz (Cornell)
Rob Harrison (West Point)
Nanxi Kang (Princeton)
Naga Praveen Katta (Princeton)
Chris Monsanto (Princeton)
Josh Reich (Princeton)
Cole Schlesinger (Princeton)
Robert Soulé (Cornell)
Alec Story (Cornell)
Nate Foster (Cornell)
Mike Freedman (Princeton)
Jen Rexford (Princeton)
Emin Gün Sirer (Cornell)
Dave Walker (Princeton)

http://frenetic-lang.org
BACKUP SLIDES
Beyond Per-Packet

**Per-flow consistent update**
Each set of related packets processed with old or new configuration, but not a mixture of the two.

**Use Cases**
- Load balancer
- Flow affinity
- In-order delivery

**Mechanism**
- 2-Phase Update + “flow tracking”