Consistent Updates for Software-Defined Networks: Change You Can Believe In!

Mark Reitblatt, Nate Foster, Jen Rexford, and Dave Walker
“[A] network change was performed as part of our normal AWS scaling activities... This change disconnected both the primary and secondary network simultaneously, leaving the affected nodes completely isolated from one another.”
Prior Work

Seamless IGP migration

Avoiding transient loops during the convergence of link-state routing protocols

Consensus routing

Graceful state migration
Example

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>Other</td>
<td>Drop</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>Allow</td>
<td></td>
</tr>
</tbody>
</table>
Initial Configuration

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Web hat]</td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td>![Other hat]</td>
<td>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>![Any hat]</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

1-2: F1  
3: F2  
4: F3  

1-2: Web: ✓  
*: ✗  

*: ✓  

*: ✓  

1-2: F1  
3: F2  
4: F3
Redistribute Configuration

Web: ✓ ✱:
[*: ✗]

F1
F2
F3

1-2: F1
3: F2
4: F3

I

Web: ✓ ✱:
[*: ✗]

F1
F2
F3

1: F1
2: F2
3-4: F3

I

Redistribute Con
fguration

6
Software Defined Networks (SDN)
void main() {
  ... monitor ...
  Conf = balance_load();
  install(F1, Conf[F1]);
  install(I, Conf[I]);
  ...
}
## Initial Configuration

<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>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>🧣</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

**Diagram:**
- **Web:** Allow
- **Other:** Drop
- **Any:** Allow

**Routing:**
- 1-2: F1
- 3: F2
- 4: F3
Initial Configuration

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="hat.png" alt="Hat" /></td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td><img src="hat.png" alt="Hat" /></td>
<td>Other</td>
<td>Drop</td>
</tr>
<tr>
<td><img src="hat.png" alt="Hat" /></td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

- Web: ✓
- *: ✗
- *: ✓
- *: ✓

1-2: F1
3: F2
4: F3

10
## Updating Configuration

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![hat1]</td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td>![hat2]</td>
<td>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>![hat3]</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

**F1**
- Web: ✓
- ✗

**F2**
- ✗
- ✓

**F3**
- ✓

1: F1
2: F2
3-4: F3

11
Updating Configuration

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Web Hat]</td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td>![Other Hat]</td>
<td>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>![Any Hat]</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

- **Web**: ✓
- **Other**: ✗
- **Any**: Allow

1-2 Web: Allow
1-2 Other: Drop
3-4 Any: Allow

Updating Configuration 12
Updating Configuration

<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>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>🎩</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

Web: ✓ ✗
*: ✓ ✗

1: F1
2: F2
3-4: F3

Updating Configuration
Updating Configuration

<table>
<thead>
<tr>
<th>Src</th>
<th>Traffic</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="black" alt="hat" /></td>
<td>Web</td>
<td>Allow</td>
</tr>
<tr>
<td><img src="black" alt="hat" /></td>
<td>Other</td>
<td>Drop</td>
</tr>
<tr>
<td><img src="white" alt="hat" /></td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>

- Web: ✓, *: ✗
- *: ✓
- *: ✓

- F1
- F3

- 1: F1
- 2: F2
- 3-4: F3

14
Updating Configuration

<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>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>🧢</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>
Updating Configuration

<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>Other</td>
<td>Drop</td>
</tr>
<tr>
<td>🧢</td>
<td>Any</td>
<td>Allow</td>
</tr>
</tbody>
</table>
Bad Update Order

Safe       Broken Connectivity       Broken Security

Diagram showing a network of nodes labeled F1, F2, and F3 with arrows indicating connectivity and update order.
Towards a Solution

Updating individual switches doesn’t work!

What’s the solution?

• Don’t implement updates rule-by-rule and switch-by-switch!

• Leverage the run-time system to handle tedious, low-level details
void main() {
  ...
  monitor ...
  Conf = balance_load();
  install(Conf);
}
An update from configuration A to configuration B is **per-packet consistent** if each packet is routed according to either configuration A or B.
A path property $\phi$ specifies the legal paths that a packet can take through a network $N$.

Formally: $\phi \subseteq \text{Packet} \times \text{Paths}(N)$.

- Loop-freedom
- “Block SSH traffic from 10/8”
- “All Web traffic waypoints through switch 5”
void main() {
  ...
  monitor ...
  Conf = balance_load();
  install(perpacket, Conf);
}
Beyond Path Properties

Not path properties:

- In-order delivery
- Flow affinity

An update from configuration A to configuration B is per-flow consistent if each packet in the same flow is routed according to either configuration A or B.

See paper for details
2-Phase Implementation

1. Instrument new configuration with version
2. Install instrumented configuration, leaving all old ingress rules active
3. Activate new ingress rules
4. Wait for old version packets to leave
5. Uninstall old configuration
Future Work

Implementation

• Naive prototype running
• Exploring performance optimizations

Unplanned Change

• Highly responsive
• Weaker consistency

Formal Verification

• Specification language for path properties
• Configuration verifier
Ongoing Work

• This paper
  Network write abstraction

• PRESTO ’10, ICFP ’11
  Network read abstraction

• POPL ’12
  Rich policy abstraction
Questions?

Thank You

http://frenetic-lang.org
## Database Analogy

<table>
<thead>
<tr>
<th>Network</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully routed packet</td>
<td>Read Transaction</td>
</tr>
<tr>
<td>Single hop routed packet</td>
<td>Read</td>
</tr>
<tr>
<td>Network update</td>
<td>Write Transaction</td>
</tr>
<tr>
<td>Single switch update</td>
<td>Write</td>
</tr>
<tr>
<td>Per-Packet Consistency</td>
<td>Snapshot Isolation</td>
</tr>
</tbody>
</table>