Good Network Updates for Bad Packets

Arne Ludwig,
Matthias Rost, Damien Foucard, Stefan Schmid
Updates happen

• Network updates happen
  – Changing security policies
• Network updates are challenging
  – Even with global view
• Potential high damage if fail
  – Security policy violation
Example
Example

Waypoint Enforcement (WPE)
Example

- Eventual consistency
Eventual consistency

Transient consistency?
Example

- Eventual consistency
- Transient consistency?
Example

- Eventual consistency
- Transient consistency?
Example

- Eventual consistency
- Transient consistency?
Example

- Eventual consistency
- Transient consistency?
Example

- Eventual consistency
- Transient consistency
Outline

- What could possibly go wrong?
- It's not a trivial thing!
- But we present an optimal solution.
Model and a Trivial Compression

Solid lines = current path

\( s_1 \) \( s_2 \) \( s_3 \) \( s_4 \)
Model and a Trivial Compression

- Solid lines = current path
- Dashed lines = new path
- Flow-specific path
Model and a Trivial Compression

Solid lines = current path
Dashed lines = new path
Flow-specific path
Model and a Trivial Compression

Solid lines = current path
Dashed lines = new path
Flow-specific path

Safe to be updated
Safe to be left untouched
Consistency Properties

- WPE = every packet traverses the waypoint at least once
- LF = loop freedom
Update all “simultaneously“?
Update all “simultaneously“?

Not possible in practice!

What could possibly go wrong?
Update all “simultaneously“?

Not possible in practice!

What could possibly go wrong?

Update times can vary significantly (up to 10x higher than median [Dionysus – SIGCOMM'14])
Update all “simultaneously“?
Update all “simultaneously“?

- Not waypoint enforced!
Delay $s_1$?
Delay $s_1$ ?

- Not loop free!
Update possible?
Update possible?
Update possible?

\[ s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \]
Update possible?

- Consistent transient states!
Rounds

- Round = set of parallel updates
- $R_1 = \{s_2\}, \quad R_2 = \{s_3\}, \quad R_3 = \{s_1\}$

→ Minimize number of rounds / communication overhead
Greedy Update Fails

- Greedy approach may:
  - take up to $\Omega(n)$ times more rounds
  - fail to find solution
Greedy Update Fails

• Greedy approach may:
  – take up to $\Omega(n)$ times more rounds
  – fail to find solution

See paper!
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
2. Switches < WP (new), < WP (old)
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
2. Switches < WP (new), < WP (old)
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
2. Switches < WP (new), < WP (old)
3. Remaining switches

\[ s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \]
WPE - Update Algorithm

1. Switches < WP (new), > WP (old)
2. Switches < WP (new), < WP (old)
3. Remaining switches

Constant in 3 rounds, but not LF!
LF and WPE Conflict
LF and WPE Conflict

- $s_1, s_2$ violate WPE; $s_3, s_4$ violate LF
Mixed Integer Program

Minimize Rounds

\[ \min R \]

\[ R \geq r \cdot x^r_v \quad r \in \mathcal{R}, v \in V \]  

\[ 1 = \sum_{r \in \mathcal{R}} x^r_v \quad v \in V \]  

\[ y^r_{u,v} = 1 - \sum_{r' \leq r} x^r_u \quad r \in \mathcal{R}, (u,v) \in E_{\pi_1} \]  

\[ y^r_{u,v} = \sum_{r' \leq r} x^r_u \quad r \in \mathcal{R}, (u,v) \in E_{\pi_2} \]  

\[ a^r_s = 1 \quad r \in \mathcal{R} \]  

\[ a^r_v \geq a^r_u + y^r_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E \]  

\[ a^r_v \geq a^r_u + y^r_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E \]  

\[ y^{r-1 \forall r}_{u,v} \geq a^r_u + y^{r-1}_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E \]  

\[ y^{r-1 \forall r}_{u,v} \geq a^r_u + y^{r-1}_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E \]  

\[ y^{r-1 \forall r}_{u,v} \leq \frac{l^r_v - l^r_u - 1}{|V| - 1} + 1 \quad r \in \mathcal{R}, (u,v) \in E \]  

\[ a^r_s = 1 \quad r \in \mathcal{R} \]  

\[ a^r_v \geq a^r_u + y^r_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E_{\text{WP}} \]  

\[ a^r_v \geq a^r_u + y^r_{u,v} - 1 \quad r \in \mathcal{R}, (u,v) \in E_{\text{WP}} \]  

\[ a^r_t = 0 \quad r \in \mathcal{R} \]
Mixed Integer Program

Mixed Integer Program

Optimal solution

Unclassified (stopped 600sec)

Not solvable (provably)
Solvability Analysis

- % of solvable instances?
- % of failed greedy?
- 1k random permutations per size
- Max duration 600 seconds
Solvability Analysis

Number of switches

Percentage of solvable instances

- Greedy
- MIP
- Unclear
- No solution
Solvability Analysis

Percentage of solvable instances

Number of switches

- Greedy
- MIP
- Unclear
- No solution
Conclusion

- Transient consistency is not easy to guarantee
- LF and WPE might even conflict
- Greedy can fail to find consistent updates

Dynamic WPE + LF updates are hard to find!
Backup Slides
Scaling of MIP – Solvable Instances
Scaling of MIP – Unsolvable Inst.
SDN: Tagging vs. Dynamic

Tagging
- Per packet consistency → Included security

Dynamic
- Load adaptive [Dionysus]
- Parts updated earlier
- Efficient partial updates

Partial update:
- Tagging: communication with all switches
- Dynamic: communication only with affected switches
SDN – Mind Map

- Dst based
- Greedy

On Consistent Updates in SDN [HotNets'13]

Dynamic Scheduling of Network Updates [SIGCOMM'14]

Abstractions for Network Update [SIGCOMM'12]

zUpdate: Updating Data Center Networks with Zero Loss [SIGCOMM'13]

WAN/Inter DCN

B4: Experience with a Globally-Deployed Software Defined WAN [SIGCOMM'13]

Towards Correct Network Virtualization [HotSDN'14]

Good Network Updates for Bad Packets [HotNets'14]

Dynamic Scheduling of Network Updates

- Flow based
- Min rounds

Extends to transient

Creates system + more dynamic

Add dependencies to other flows in DCN

Eventual/per packet consistency not enough

Adds security
Solution for Greedy Fail
Solution for Greedy Fail
Solution for Greedy Fail
Solution for Greedy Fail
Solution for Greedy Fail
Solution for Greedy Fail

\[ s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7 \]