Latte: Improving the Latency of Transiently Consistent Network Update Schedules

Mark Glavind, Niels Christensen, Jiri Srba

Aalborg University, Denmark

Stefan Schmid

University of Vienna, Austria

Funding:

DANMARKS FRIE FORSKNINGSFOND

W|W|T|F
VIENNA SCIENCE AND TECHNOLOGY FUND

Motivation: Two Trends in Networking

Networks become more flexible and „adaptable“

- ❑ Enablers: SDN, virtualization, reconfigurable optical topologies
- ❑ Vision of more dynamic, demand-aware, self-adjusting and „self-driving networks“: improve resource efficiency and performance

Networks are critical infrastructure of digital society

- ❑ Increasingly stringent dependability requirements

Motivation: Two Trends in Networking

Networks become more flexible and „adaptable“

- ❑ Enablers: SDN, virtualization, reconfigurable optical topologies
- ❑ Vision of more dynamic, demand-aware, self-adjusting and „self-driving networks“: improve resource efficiency and performance

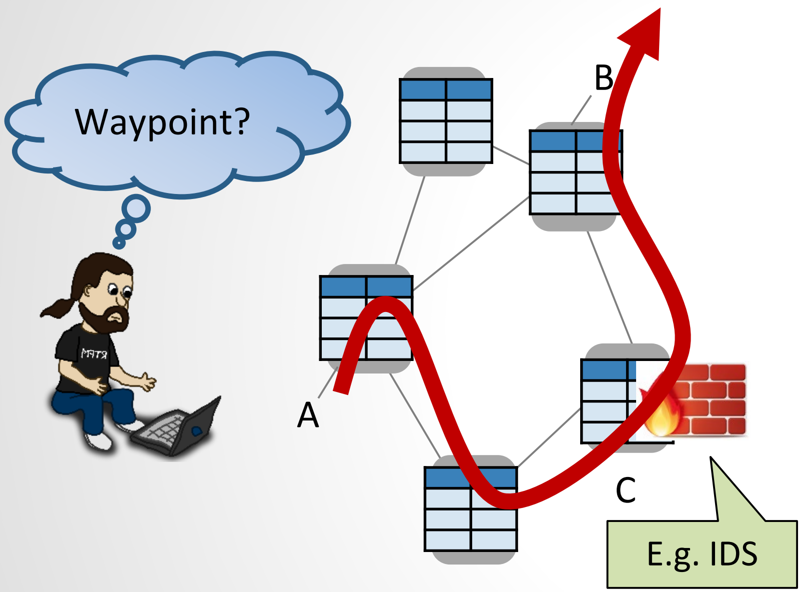
VS

Networks are critical infrastructure of digital society

- ❑ Increasingly stringent dependability requirements

A contradiction?
Performance-reliability tradeoff?

Responsible for Reliability: Network Operator



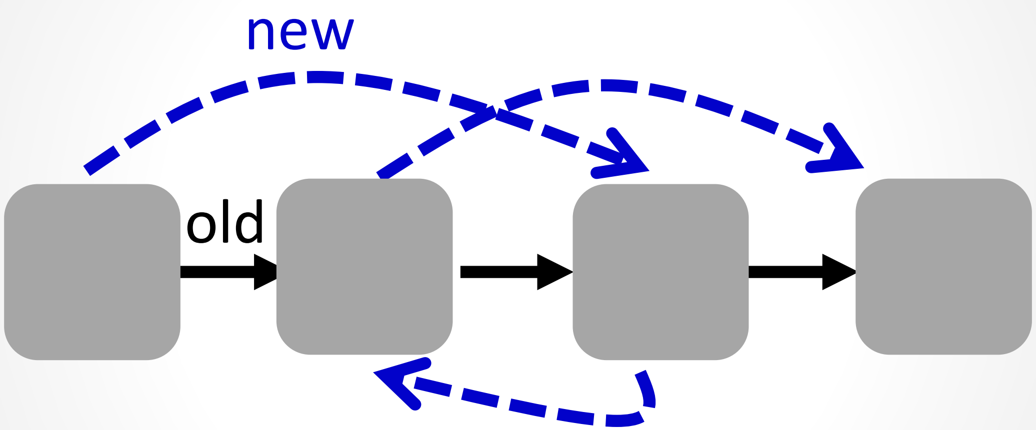
Operator responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?
- **Policy:** Is it ensured that traffic from A to B never goes via C?
- **Waypoint enforcement:** Is it ensured that traffic from A to B is always routed via a node C (e.g., intrusion detection system or a firewall)?

Even more challenging in dynamic network!

This Paper: Providing Efficiency and Reliability in the Context of Dynamic Routing

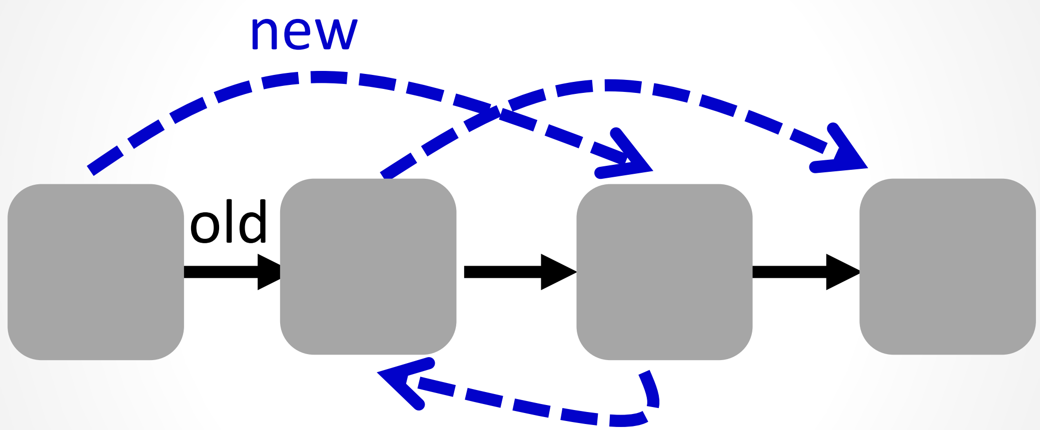
- How to quickly and correctly change from an old route to a new route?



- A.k.a. the Consistent Network Update Problem
- Motivation for changing routes:
  - Traffic engineering, changes in the demand, security policy changes, service relocation, maintenance work, link/node failures, ...

This Paper: Providing Efficiency and Reliability in the Context of Dynamic Routing

- How to quickly and correctly change from an old route to a new route?



- A.k.a. the Consistent Network Update Problem
- Motivation for changing routes:
  - Traffic engineering, changes in the demand, security policy changes, service relocation, maintenance work, link/node failures, ...

This paper focuses on **Software-Defined Networks (SDNs)**: programmable networks managed by a centralized controller.

An Active Research Area

- Recent survey* discusses >100 related papers
  - A classic problem
  - Recent interest due to SDN and more stringent transient dependability requirements
  - E.g., keynote by Nate Foster at ACM CoNEXT 2018

Abstractions for Network Update

Mark Reitblatt, Cornell; Nate Foster, Cornell; Jennifer Rexford, Princeton; Cole Schlesinger, Princeton; David Walker, Princeton

Efficient Synthesis of Network Updates

Jedidiah McClurg, CU Boulder, USA, jedidiah.mcclurg@colorado.edu; Hossein Hojjat, Cornell University, USA, hojjat@cornell.edu; Pavol Černý, CU Boulder, USA, pavol.cerny@colorado.edu; Nate Foster, Cornell University, USA, jnfoster@cs.cornell.edu

Good Network Updates for Bad Packets

Waypoint Enforcement Beyond Destination-Based Routing Policies

Arne Ludwig[1], Matthias Rost[1], Damien Foucard[1], Stefan Schmid[1,2]

[1] TU Berlin, Berlin, Germany; [2] T-Labs, Berlin, Germany

Timed Consistent Network Updates

Tal Mizrahi, Efi Saat, Yoram Moses*

Technion — Israel Institute of Technology

{dew@tx, efisaat@tx, moses@ee}.technion.ac.il

Enforcing Customizable Consistency Properties in Software-Defined Networks

Wenxuan Zhou*, Dong Jin**, Jason Croft*, Matthew Caesar*, and P. Brighten Godfrey*

On Consistent Updates in Software Defined Networks

Ratul Mahajan, Microsoft Research; Roger Wattenhofer, ETH Zurich

* Foerster et al., Survey of Consistent Software-Defined Network Updates, IEEE Communications Surveys and Tutorials (COMST), 2018.

An Active Research Area

- Recent survey* discusses >100 related papers
  - A classic problem
  - Recent interest due to SDN and more stringent transient dependability requirements
  - E.g., keynote by Nate Foster at ACM CoNEXT 2018

* Foerster et al., Survey of Consistent Software-Defined Network Updates, IEEE Communications Surveys and Tutorials (COMST), 2018.

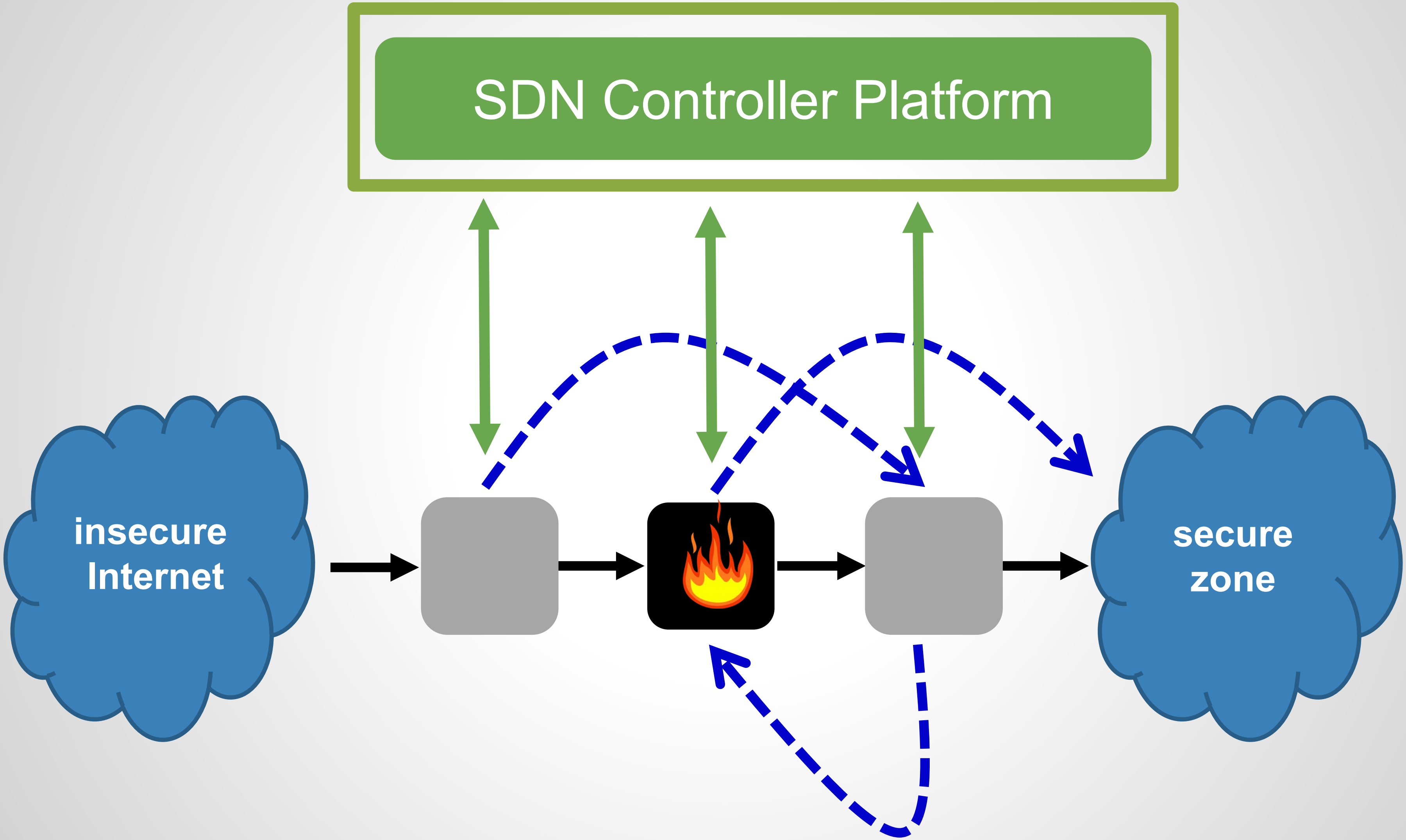
Roadmap of This Talk

- ❑ Background and Model
- ❑ Motivation and Contribution
- ❑ Approach
- ❑ Evaluation
- ❑ Demo

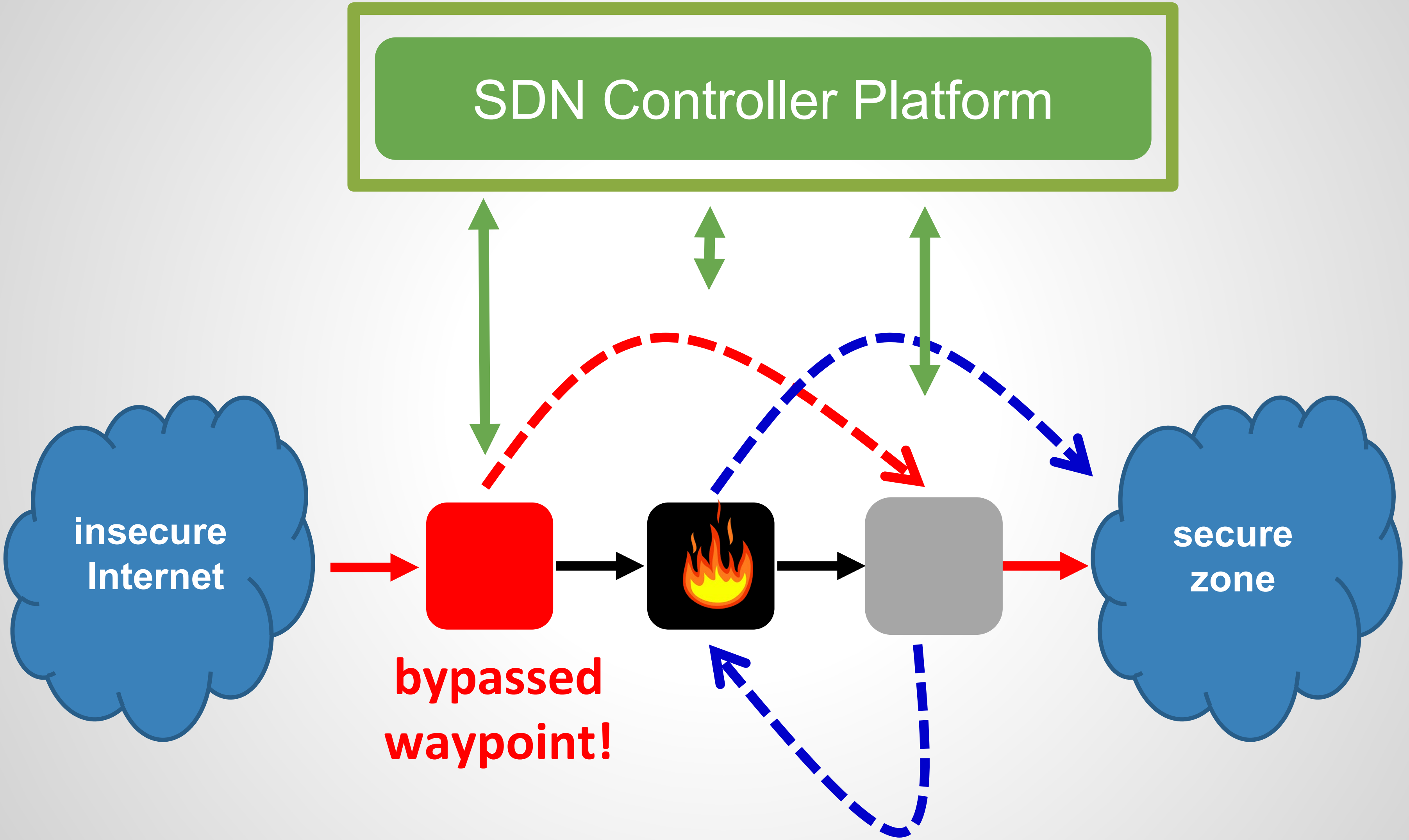
Roadmap of This Talk

- ❑ **Background and Model**
- ❑ Motivation and Contribution
- ❑ Approach
- ❑ Evaluation
- ❑ Demo

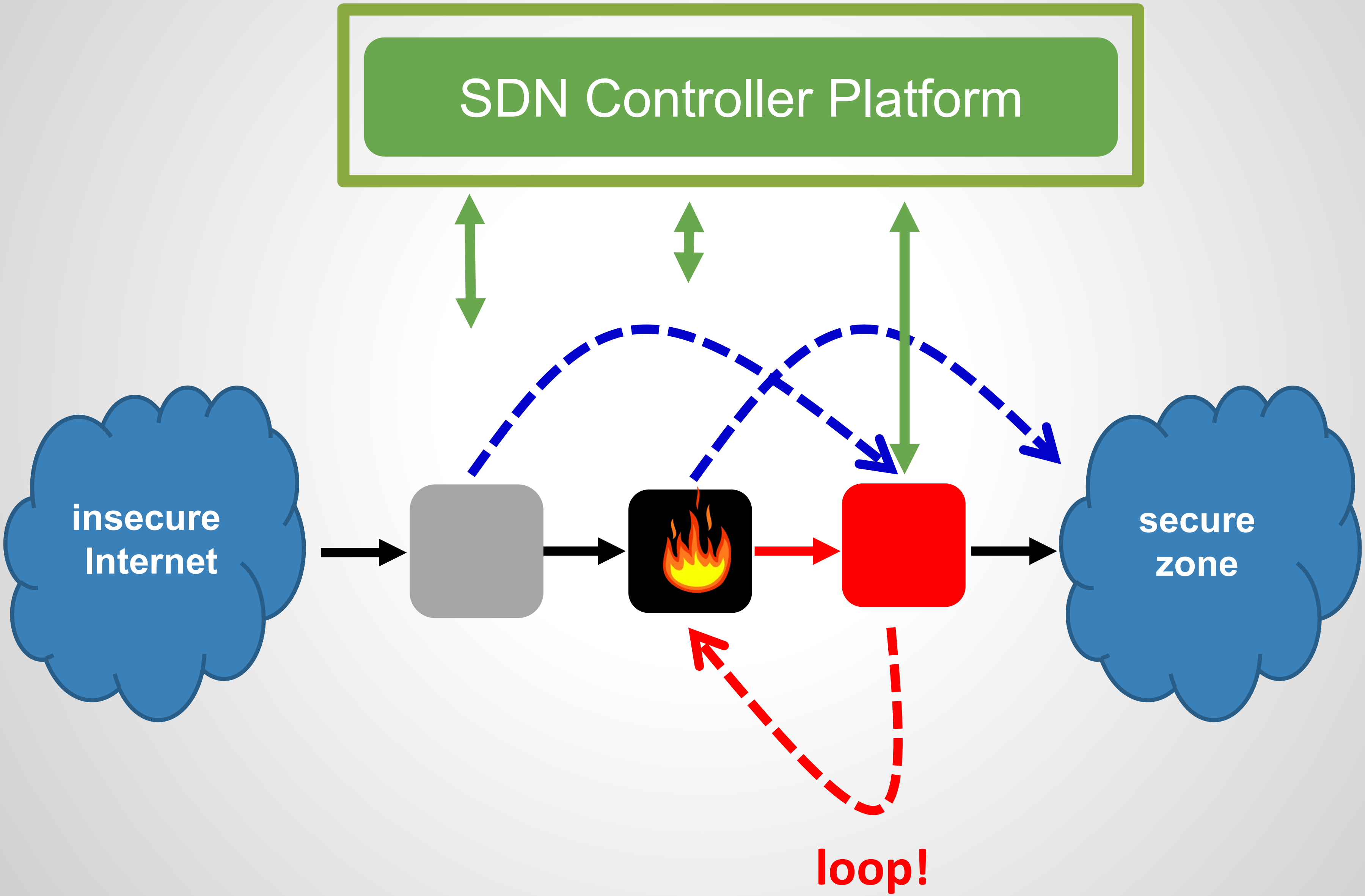
The Challenge: Asynchrony



The Challenge: Asynchrony

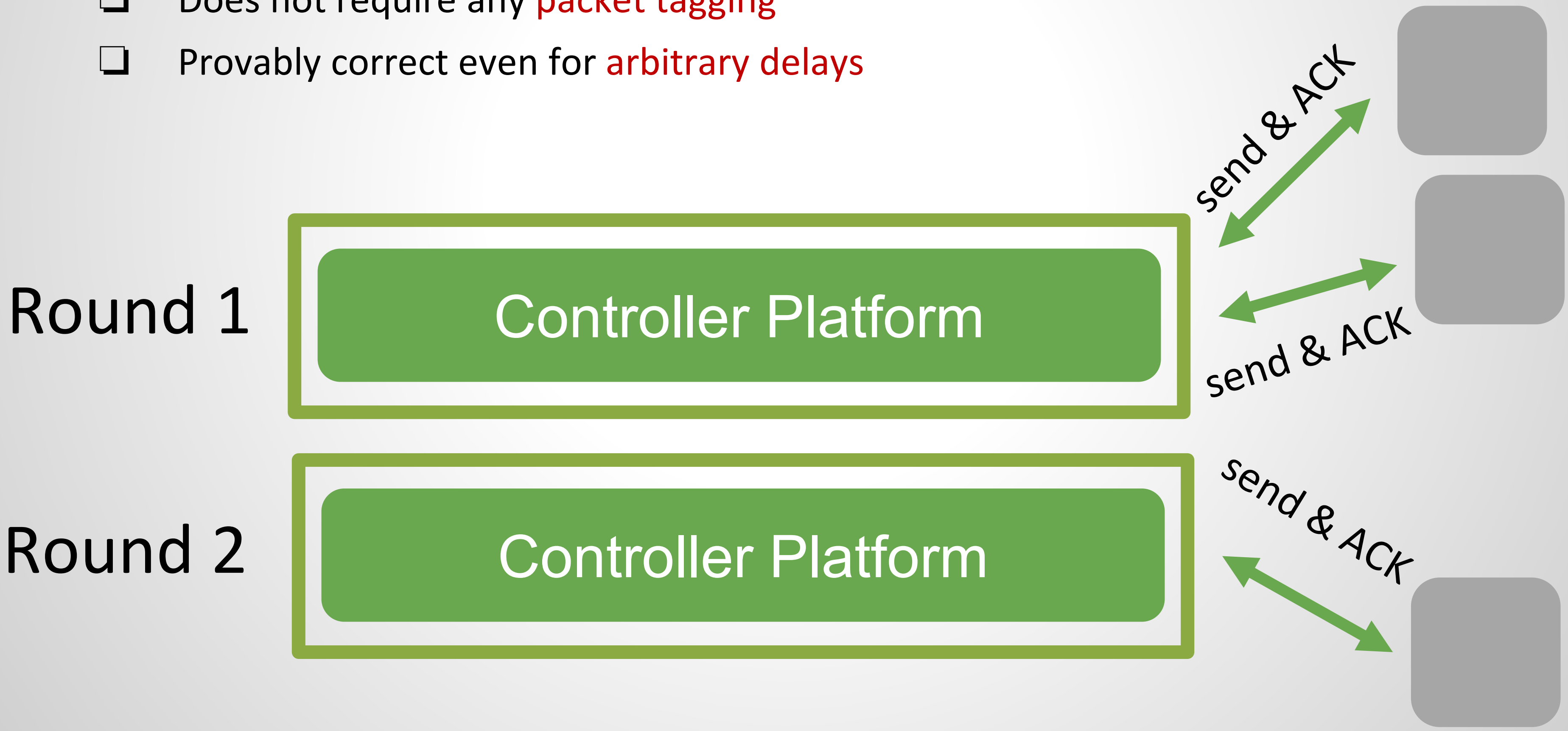


The Challenge: Asynchrony



Popular Approach to Ensure Transient Consistency

- Proceed in multiple rounds
  - Proceed to next round when ACK received
  - Does not require any packet tagging
  - Provably correct even for arbitrary delays



Roadmap of This Talk

- ❑ Background and Model
- ❑ **Motivation and Contribution**
- ❑ Approach
- ❑ Evaluation
- ❑ Demo

Motivation for Our Paper

Existing consistent network update mechanisms:

- Often based on hand-crafted algorithms } ☹ Complex
- Either overly pessimistic: underlying network may be arbitrarily asynchronous } ☹ Unnecessarily slow
- Overly optimistic model where updates can be timed precisely } ☹ Requires special hw

Our Paper

Latte: Improving the Latency of Transiently Consistent Network Update Schedules

Niels Christensen, Mark Glavind, Jiří Srba, Stefan Schmid

- Fully automated approach to optimize the performance of network update schedulers
- Synthesize waiting times between (ordered) updates
  - Accounting for update time characteristics
  - E.g., different packet types, such as VoIP, SSH, or VPN, entail different forwarding times at switches [1]
- Support wide range of consistency properties, e.g.:
  - (Sequence of) waypoint enforcement
  - Loop freedom
  - Blacklist enforcement
  - Blackhole freedom

[1] Bauer et al., Behind the scenes: What device benchmarks can tell us. Proc. ANRW, 2018

Roadmap of This Talk

- ❑ Background and Model
- ❑ Motivation and Contribution
- ❑ **Approach**
- ❑ Evaluation
- ❑ Demo

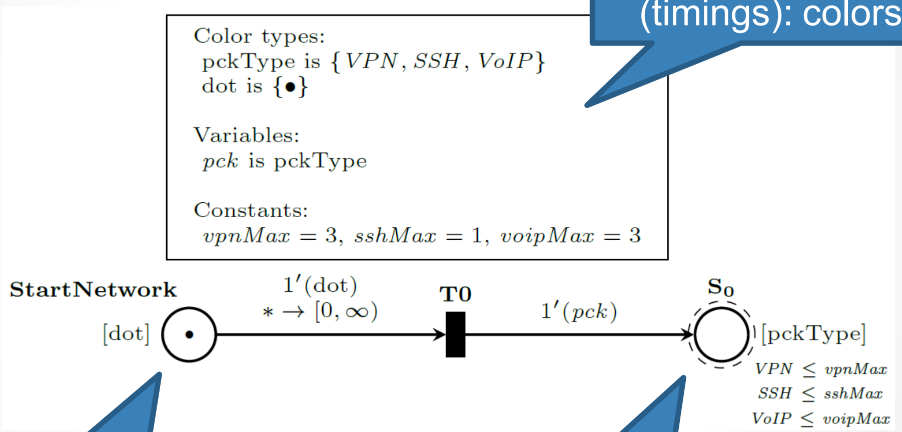
Novel Extension of Classic Petri Nets: Timed-Arc Colored Petri Nets (TACPNs)

- Petri nets: powerful modeling language for distributed systems
  - Configurations: tokens located at places
- In our extension: tokens also contain
  - Color information: e.g., modeling different packet types
  - Time information: e.g., modeling age
- Places and input arcs have time constraints for each color

Example: Encoding Network Updates in TACPNs

(1) Gadget to inject packets:

Packets can be of different types (timings): colors



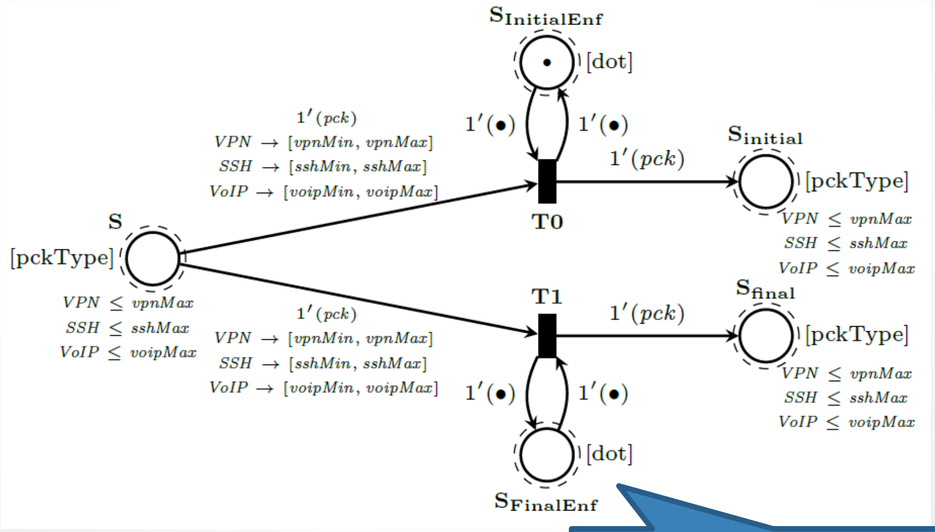
Initially: token at this place

Jump to place S_0 and generate packet of arbitrary type

Example: Encoding Network Updates in TACPNs

② Gadget to model switches:

If token up here: packets go old path

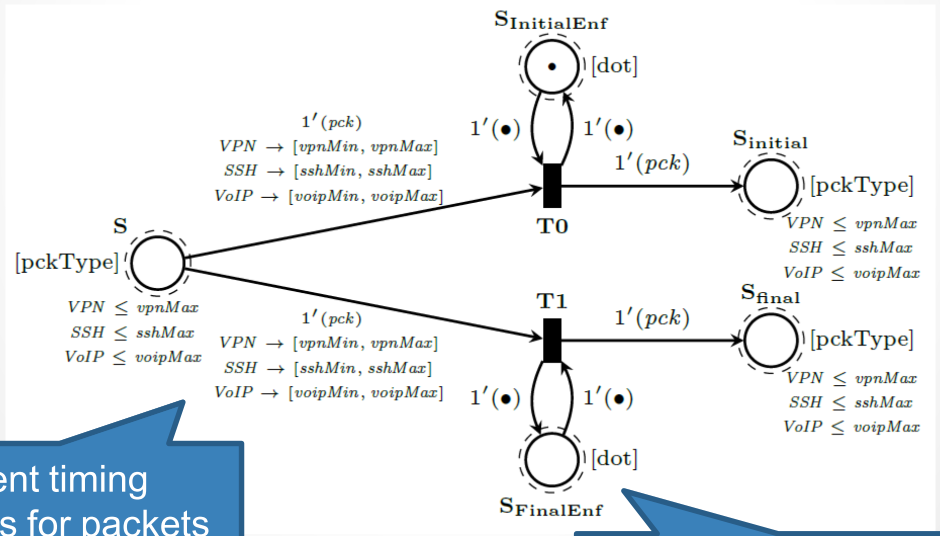


If token down here: switch updated to new path

Example: Encoding Network Updates in TACPNs

(2) Gadget to model switches:

If token up here: packets go old path



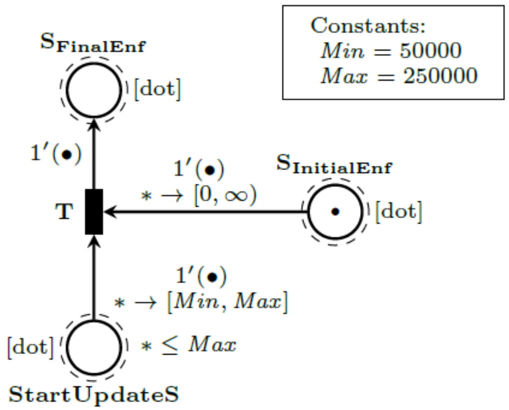
Different timing constraints for packets

If token down here: switch updated to new path

Example: Encoding Network Updates in TACPNs

③ Gadget to model switch update:

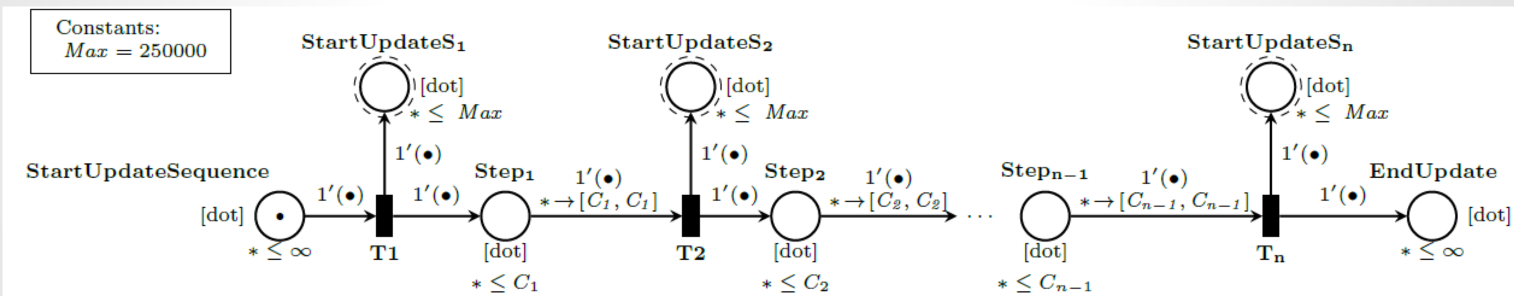
How to change between initial and final switch configuration



Starting here, the update can take time between min and max

Example: Encoding Network Updates in TACPNs

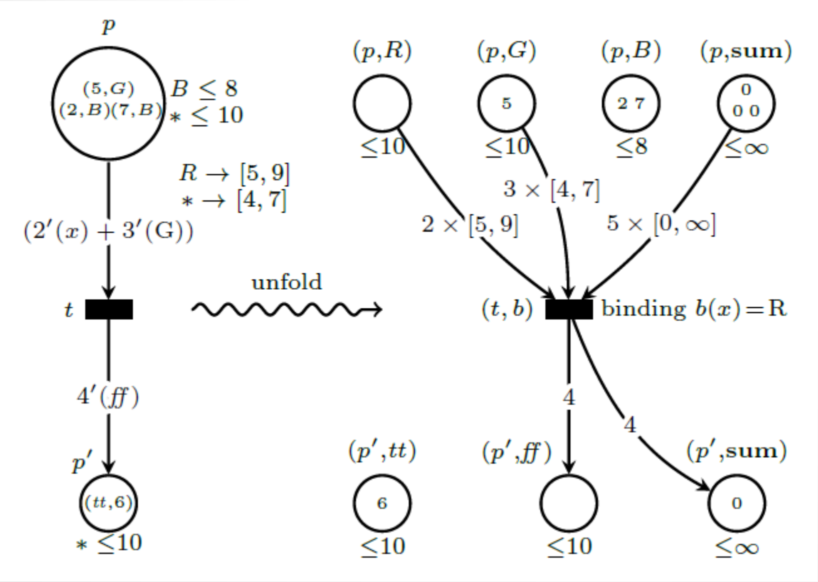
(4) Connecting the pieces: initialization of update sequence for all *n* switches



After updating Switch S_1 (delay C_1), go to Switch S_2, etc.

Analysis

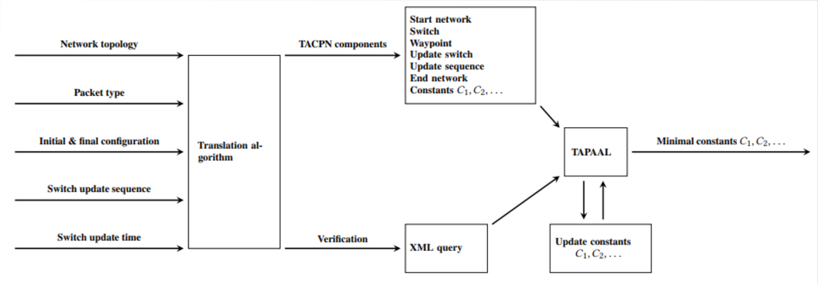
We show that the constructed nets can be analyzed efficiently via their unfolding into existing timed-arc Petri nets.

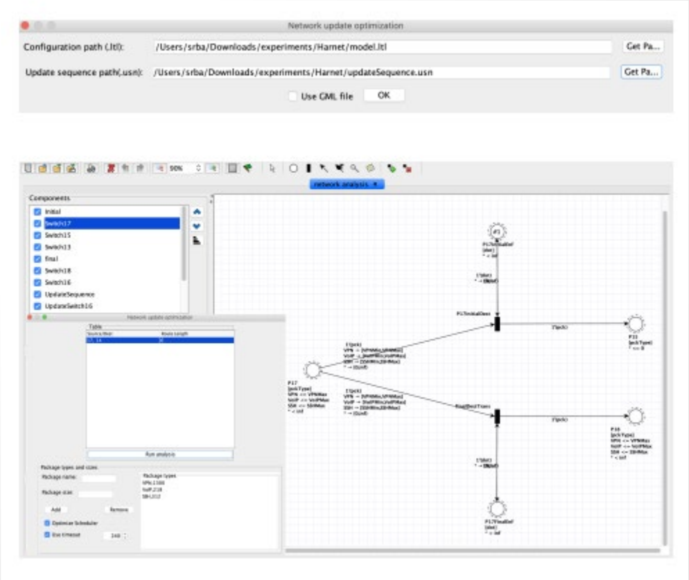


Preserves bisimilarity!

Tool Support

- Latte translates a given network update problem into a TACPN to compute minimal switch update delays
- Comes with strong tool support
- Integrated Latte plugin in open source tool TAPAAL
- Allows to draw networks graphically and issue CTL queries





Roadmap of This Talk

- ❑ Background and Model
- ❑ Motivation and Contribution
- ❑ Approach
- ❑ **Evaluation**
- ❑ Demo

Improved Latency of Update Schedules

Network	Route length	Verification time[s]	Default update time [s]	Optimized update time [s]	Improvement [%]
TLex	4	0.74	3.58	0.25	92.30%
HiberniaIreland	5	1.02	6.05	0.28	95.50%
Harnet	6	1.42	9.08	0.28	96.97%
UniC	7	1.49	12.65	0.28	97.83%
Oxford	8	2.02	16.78	0.28	98.36%
Xeex	10	5.86	26.68	0.28	98.97%
Sunet	11	10.23	32.45	0.28	99.15%
SwitchL3	12	18.88	38.78	0.28	99.29%
Psinet	14	89.67	53.01	0.28	99.48%
Uunet	15	211.86	61.05	0.28	99.55%
Renater2010	16	480.52	69.58	0.28	99.60%
Missouri	25	timeout	171.05	67.10	60.77%
Syringa	35	timeout	336.05	295.35	12.11%
VtlWavenet2011	35	timeout	336.06	295.35	12.11%

- Network topologies from the Topology Zoo
- Experiments run on a 64-bit Ubuntu 18.04 laptop

Improved Latency of Update Schedules

Compared to conservative delays as produced by NetSynth: over 90% improvement.

Network	Route length	Verification time[s]	Default update time [s]	Optimized update time [s]	Improvement [%]
TLex	4	0.74	3.58	0.25	92.30%
HiberniaIreland	5	1.02	6.05	0.28	95.50%
Harnet	6	1.42	9.08	0.28	96.97%
UniC	7	1.49	12.65	0.28	97.83%
Oxford	8	2.02	16.78	0.28	98.36%
Xeex	10	5.86	26.68	0.28	98.97%
Sunet	11	10.23	32.45	0.28	99.15%
[illegible]	[illegible]	18.88	38.78	0.28	99.29%
[illegible]	[illegible]	89.67	53.01	0.28	99.48%
[illegible]	[illegible]	211.86	61.05	0.28	99.55%
[illegible]	[illegible]	480.52	69.58	0.28	99.60%
Missouri	25	timeout	171.05	67.10	60.77%
Syringa	35	timeout	336.05	295.35	12.11%
VtlWavenet2011	35	timeout	336.06	295.35	12.11%

Up to route length 16, optimal update time can be computed.

Too many updates can be performed concurrently: could be tackled with static analysis (future work).

- Network topologies from the
- Experiments run on a 64-bit U

Improved Latency of Update Schedules

Name	Route length	Verification time[s]	Default update [s]	Optimized update [s]	Improvement [%]
HiberniaIreland	6	4.37	4.68	0.45	90.70%
Oxford	12	4.71	7.99	0.45	94.42%
SwitchL3	8	4.67	5.78	0.47	91.95%
Psinet	16	4.67	10.18	0.45	95.63%
Renater2010	7	4.23	5.23	0.45	91.48%
Missouri	10	5.14	6.88	0.45	93.53%
Ans	13	5.73	8.52	0.43	94.90%
Bics	13	6.20	12.65	0.44	96.56%
Globalcenter	14	7.63	17.88	0.45	97.51%
Geant2009	13	11.72	16.78	0.45	97.35%

- More complicated scenario where concurrent updates are not possible
- Require minimal delays for waypointing

Improved Latency of Update Schedules

Improved verification times!

Still over 90%

Name	Route length	Verification time[s]	Default update [s]	Optimized update [s]	Improvement [%]
HiberniaIreland	6	4.37	4.68	0.45	90.70%
Oxford	12	4.71	7.99	0.45	94.42%
SwitchL3	8	4.67	5.78	0.47	91.95%
Psinet	16	4.67	10.18	0.45	95.63%
Renater2010	7	4.23	5.23	0.45	91.48%
Missouri	10	5.14	6.88	0.45	93.53%
Ans	[illegible]	5.73	8.52	0.43	94.90%
Bics	[illegible]	6.20	12.65	0.44	96.56%
Globalcenter	[illegible]	7.63	17.88	0.45	97.51%
Geant2009	[illegible]	11.72	16.78	0.45	97.35%

e.g. 67 switches within seconds!

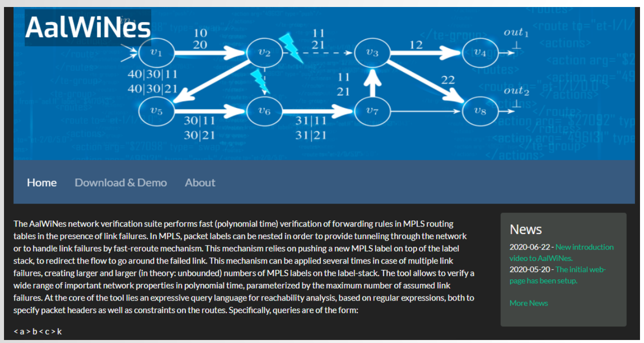
- More complicated scenario where concurrent updates are not possible
- Require minimal delays for waypointing

Roadmap of This Talk

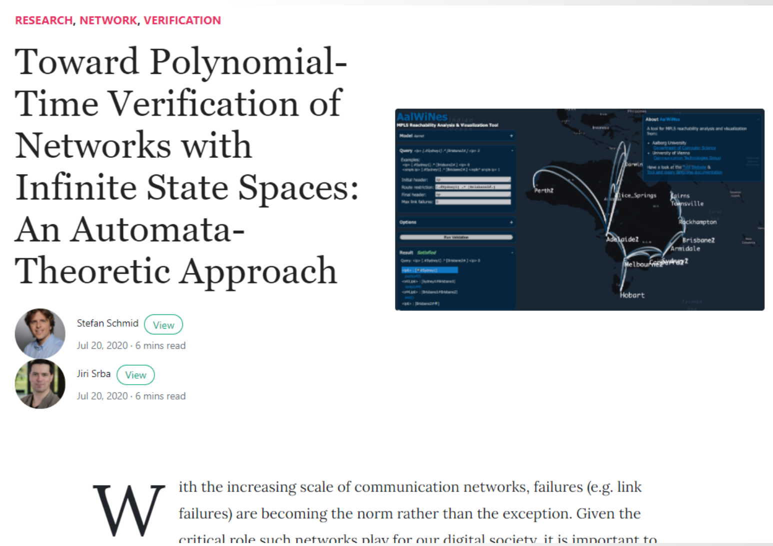
- ❑ Background
- ❑ Motivation and Contribution
- ❑ Approach
- ❑ Evaluation
- ❑ **Demo**

Further Reading

The AalWines project
https://aalwines.cs.aau.dk/



Netverify.fun



TAPAAL.net