Jump, Crawl, Attract, Propagate:
Security Challenges in Emerging Communication Networks

Stefan Schmid (Faculty of Computer Science, University of Vienna)

About Networks: Critical Infrastructure

NETFLIX

Source: Facebook

Digital society relies on networks, especially connectivity to, from, and in **datacenters**, but also more “exotic” networks such as **in-cabin networks**, **cryptocurrency** networks, etc.

Dependability on networks also because more and more “things” produce data: e.g., car sensors >6GB/h.

AI-enabled car features:

- collision risk prediction
- eight on-board cameras
- six radar emitters
- twelve ultrasonic sensors
- IMU sensor for autonomous driving
- computer power of 22 Macbook Pros

© Ivona Brandic

We‘ll see: Networks examplify what we discussed…

- *New technology needed* and automation to meet more stringent dependability requirements

- But: standardization and innovation (used to be) *slow*, deploying new security features takes time

- And: new technologies also introduce *new threats*

Panel: Cyber Security – Where Does Technology Stop and Where Should We Stop It?

Tuesday, February 25th, 2020

09:15 - 10:30

Roadmap

- To what extent can we trust our networks today?
- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation
- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing
- Another uncharted security landscape: cryptocurrency networks

Roadmap

- **To what extent can we trust our networks today?**

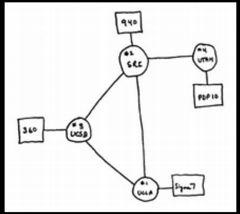
- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation

- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing

- Another uncharted security landscape: cryptocurrency networks

Slide credit:
Pedro Casas

The Internet 50 Years Ago



- Connectivity between fixed locations / "super computers"
- For researchers : Simple applications like email and file transfer

The Internet: A Success Story

Today:
- Supports connectivity between **diverse "users"** : humans, machines, datacenters, or even **things**
- Also supports wireless and **mobile** endpoints
- **Heterogeneous** applications: e-commerce, Internet telephony, VoD, gaming, etc.

Yet:
- ***Technology hardly changed! But now: mission-critical infrastructure***

But how secure are our networks?

The Internet at first sight:

- Monumental
- Passed the “Test-of-Time”
- Should not and cannot be changed

But how secure are our networks?

The Internet at first sight:

- Monumental
- Passed the “Test-of-Time”
- Should not and cannot be changed

The Internet at second sight:

- Antique
- Brittle
- More and more successful attacks

Slide credit: Adrian Perrig

Challenge: Security Assumptions Changed

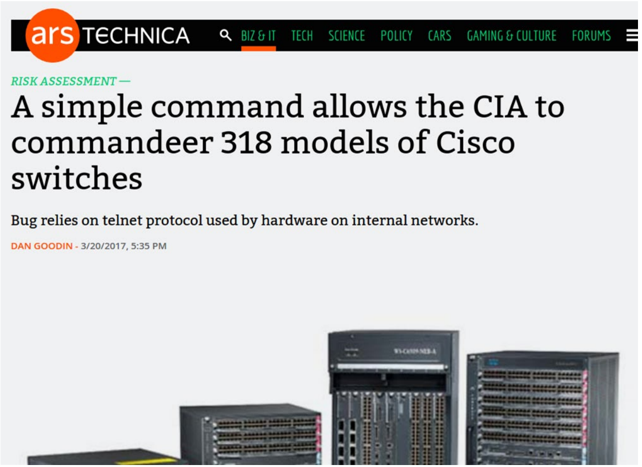
- Internet in 80s: based on **trust**
- Danny Hillis, TED talk, Feb. 2013, “There were two Dannys. *I knew both.* Not everyone knew everyone, but there was an atmosphere of trust.”

Slide credit: Adrian Perrig

More and Novel Exploits

(TS//SI//NF) Such operations involving **supply-chain interdiction** are some of the most productive operations in TAO, because they pre-position access points into hard target networks around the world.

(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A "load station" implants a beacon



ars TECHNICA | BIZ & IT | TECH | SCIENCE | POLICY | CARS | GAMING & CULTURE | FORUMS

RISK ASSESSMENT —

A simple command allows the CIA to commandeer 318 models of Cisco switches

Bug relies on telnet protocol used by hardware on internal networks.

DAN GOODIN - 3/20/2017, 5:35 PM

- **Hardware backdoors** and exploits
- The problem seems fundamental: how can we *hope to build a secure network* if the underlying hardware can be insecure?!
- E.g., *secure cloud for the government*: no resources and expertise to build own "trustworthy" high-speed hardware

More Recent Examples…

Vulnerabilities in **VPNs**


PART OF A ZDNET SPECIAL FEATURE: CYBERWAR AND THE FUTURE OF CYBERSECURITY

Iranian hackers have been hacking VPN servers to plant backdoors in companies around the world

Iranian hackers have targeted Pulse Secure, Fortinet, Palo Alto Networks, and Citrix VPNs to hack into large companies.

Let us help unlock the potential of your small business. Learn More

By Catalin Cimpanu for Zero Day | February 16, 2020 -- 20:53 GMT (20:53 GMT) | Topic: Cyberwar and the Future of Cybersecurity

Let us help unlock the potential of your small business.
FedEx.
Learn More

NEWSLETTERS

Vulnerabilities in **IoT**


Forbes Billionaires Innovation Leadership Money Business Small Business

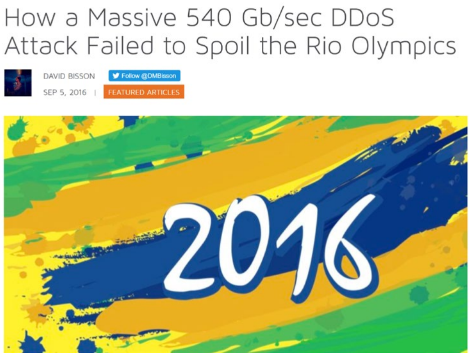
Hi!Magenta Hi Freiheit!
Jetzt nur für kurze Zeit: 9 GB um € 9 im Tarif Hi!Magenta ohne Bindung

12,571 views | Sep 14, 2019, 02:42am

Cyberattacks On IOT Devices Surge 300% In 2019, 'Measured In Billions', Report Claims

Zak Doffman Contributor
Cybersecurity
I write about security and surveillance.

DDoS attacks often in the news (e.g. **olympics**)


How a Massive 540 Gb/sec DDoS Attack Failed to Spoil the Rio Olympics

DAVID BISSON Follow @DMBisson
SEP 5, 2016 | FEATURED ARTICLES

A Major Issue: Complexity

Many outages due to misconfigurations and human errors.

Entire countries disconnected…

Data Centre ▸ Networks

Google routing blunder sent Japan's Internet dark on Friday

Another big BGP blunder

By Richard Chirgwin 27 Aug 2017 at 22:35 40 SHARE ▼

Last Friday, someone in Google fat-thumbed a border gateway protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

The trouble began when The Chocolate Factory "leaked" a big route table to Verizon, the result of which was traffic from Japanese giants like NTT and KDDI was sent to Google on the expectation it would be treated as transit.

… 1000s passengers stranded…

British Airways' latest Total Inability To Support Upwardness of Planes* caused by Amadeus system outage

Stuck on the ground awaiting a load sheet? Here's why

By Gareth Corfield 19 Jul 2018 at 11:16 109 SHARE ▼

… even 911 services affected!

Officials: Human error to blame in Minn. 911 outage

According to a press release, CenturyLink told department of public safety that human error by an employee of a third party vendor was to blame for the outage

Aug 16, 2018

Duluth News Tribune

SAINT PAUL, Minn. — The Minnesota Department of Public Safety Emergency Communication Networks division was told by its 911 provider that an Aug. 1 outage was caused by human error.

Even Tech-Savvy Companies Struggle to Provide Reliable Networks

We discovered a misconfiguration on this pair of switches that caused what's called a "bridge loop" in the network.

A network change was [...] executed incorrectly [...] more "stuck" volumes and added more requests to the re-mirroring storm

Service outage was due to a series of internal network events that corrupted router data tables

Experienced a network connectivity issue [...] interrupted the airline's flight departures, airport processing and reservations systems

Another Major Issue in Networks: Lack of Tools
Anecdote “Wall Street Bank”

- Outage of a data center of a Wall Street investment bank
- Lost revenue measured in USD 10^6 / min
- Quickly, an emergency team was assembled with experts in compute, storage and networking:
  - **The compute team:** soon came armed with ***reams of logs***, showing how and when the applications failed, and had already written experiments to reproduce and ***isolate the error***, along with candidate prototype programs to workaround the failure.
  - **The storage team:** similarly equipped, showing which file ***system logs*** were affected, and already progressing with ***workaround programs***.
  - “All the **networking team** had were ***two tools invented over twenty years ago*** to merely test end-to-end connectivity. Neither tool could reveal ***problems with the switches***, the ***congestion*** experienced by individual packets, or provide any means to create experiments to identify, quarantine and resolve the problem. Whether or not the problem was in the network, the ***network team would be blamed*** since they were unable to demonstrate otherwise.”

Source: «The world’s fastest and most programmable networks»
White Paper Barefoot Networks

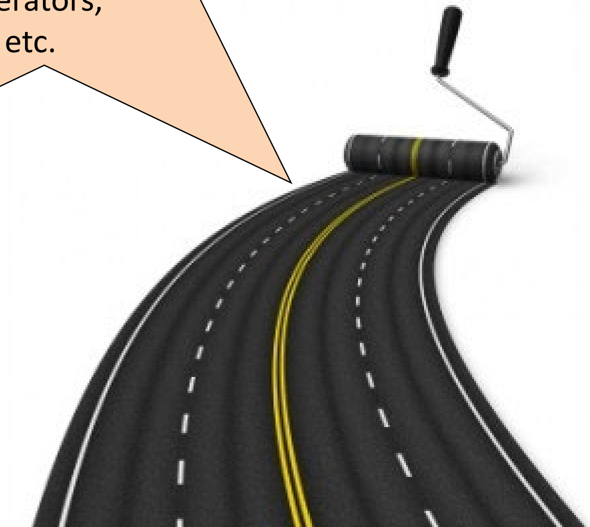
Roadmap

- To what extent can we trust our networks today?
- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation
- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing
- Another uncharted security landscape: cryptocurrency networks

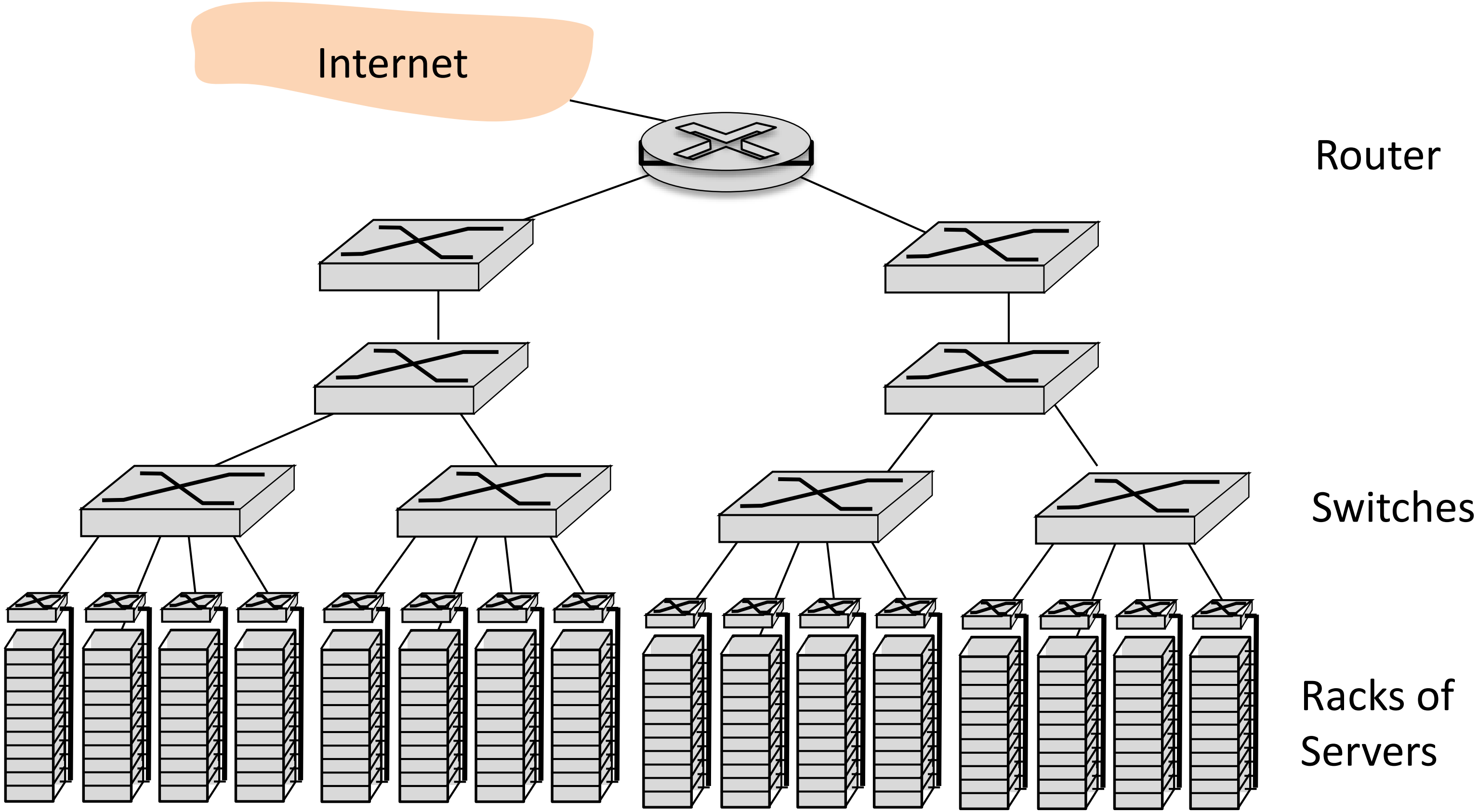
Roadmap

- To what extent can we trust our networks today?
- **Opportunity: emerging network technologies**
  - **Programmability and virtualization**
  - **„Self-driving networks“ and automation**
- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing
- Another uncharted security landscape: cryptocurrency networks

Exciting period: new tools, simple abstractions, disburdening human operators, etc.



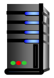
Case Study: Datacenter Network Virtualization

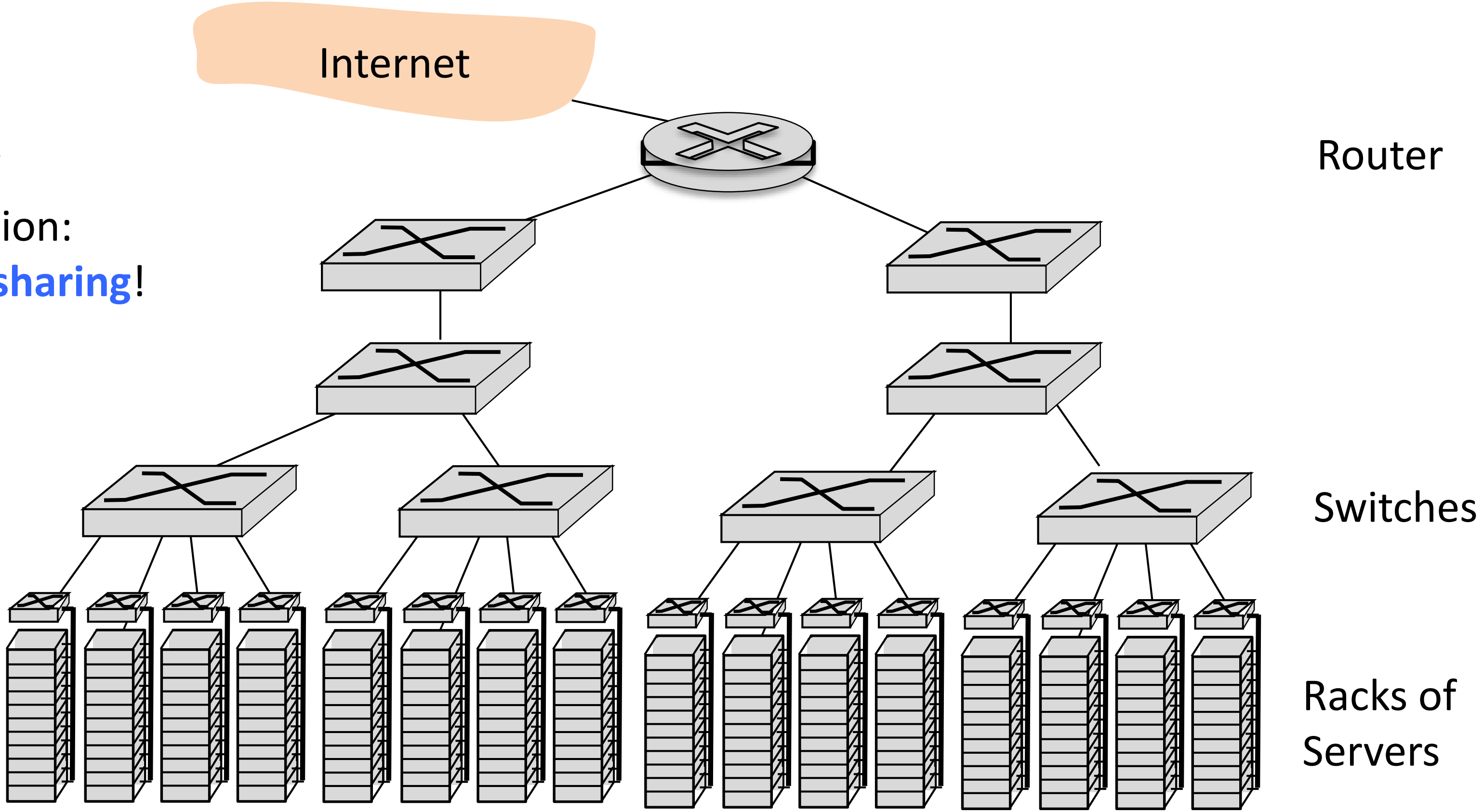


Case Study: Datacenter Network Virtualization

- Benefit of virtualization: **resource sharing**!

VMs allocated dynamically, multiplexing

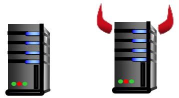


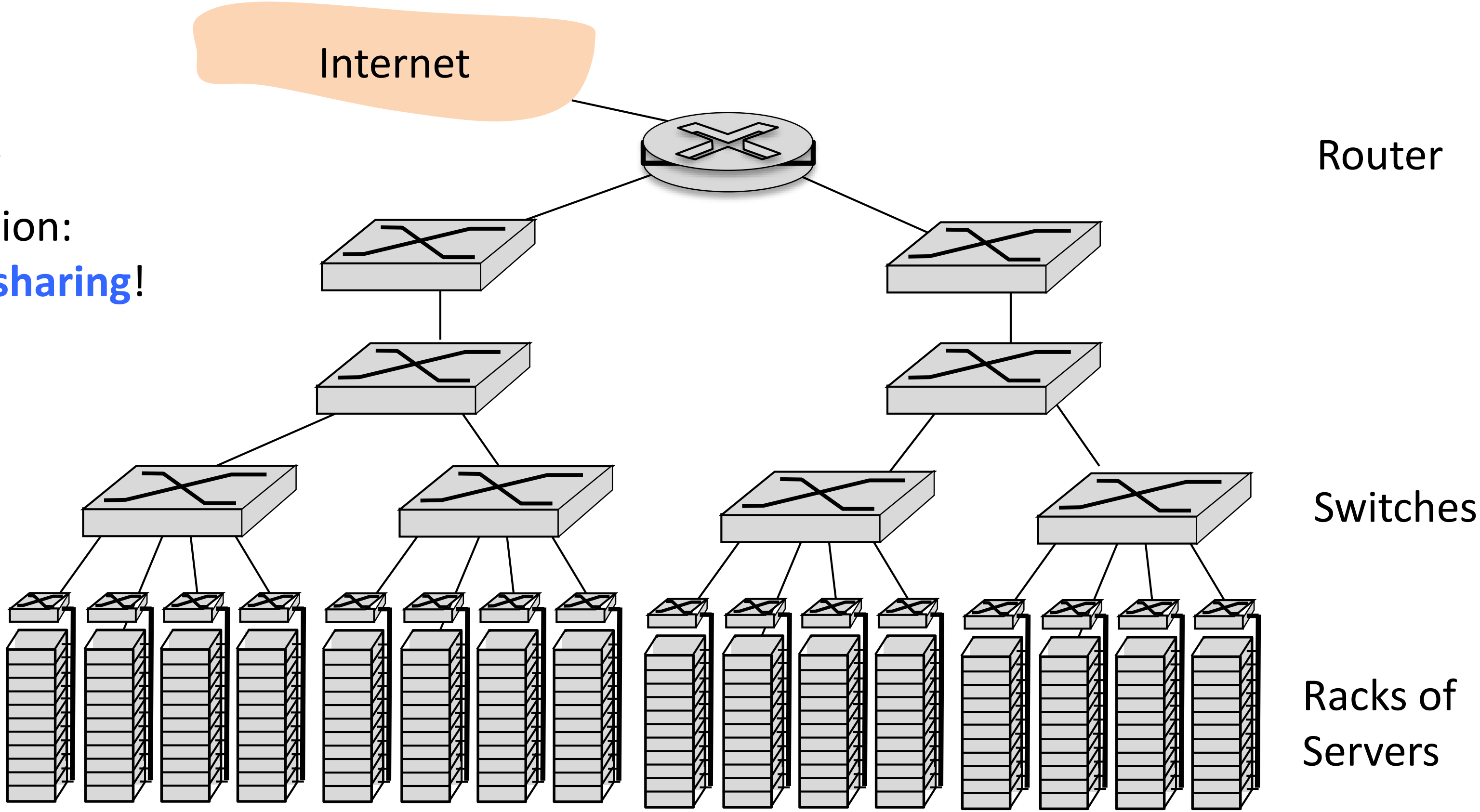


Case Study: Datacenter Network Virtualization

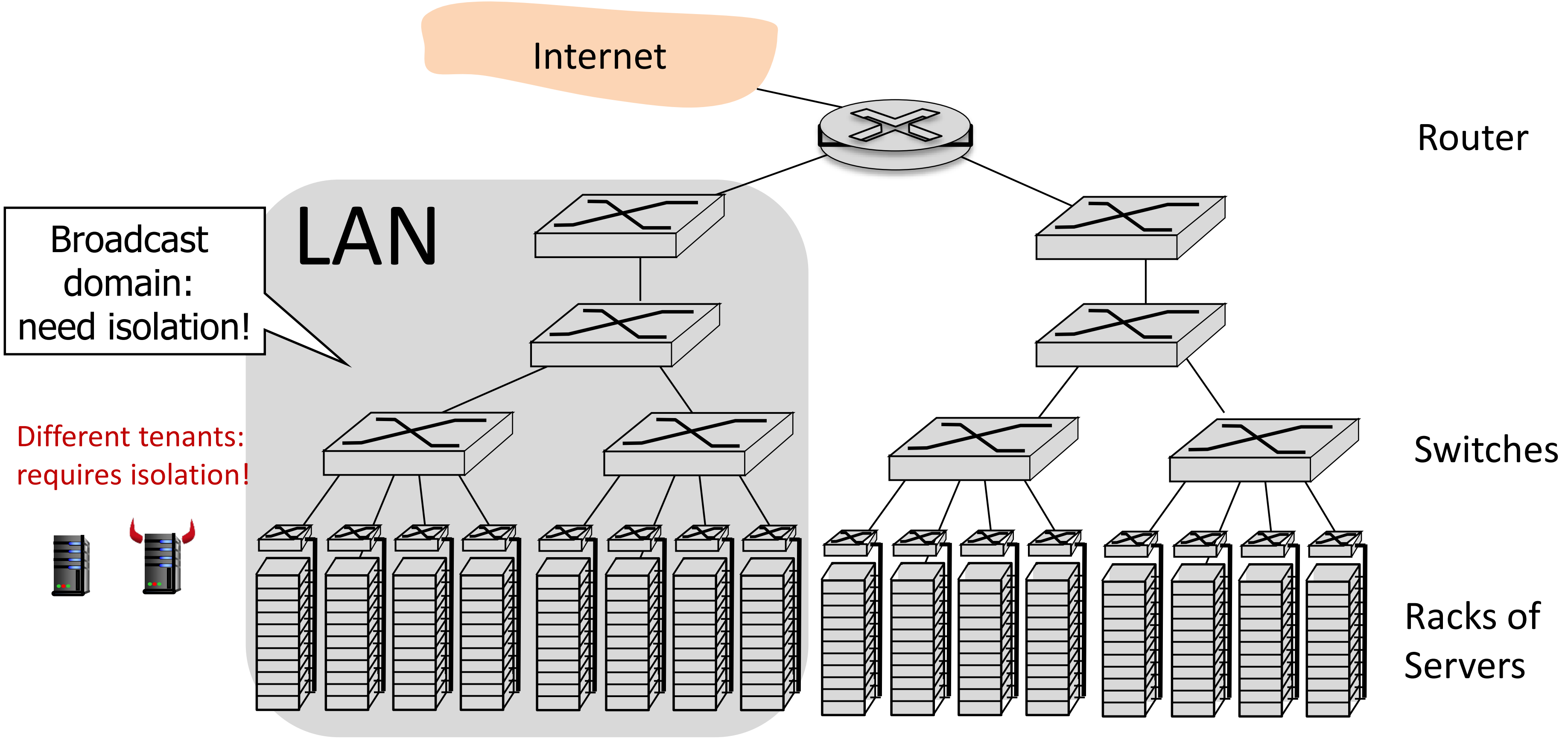
- Benefit of virtualization: **resource sharing**!

Different tenants: requires isolation!

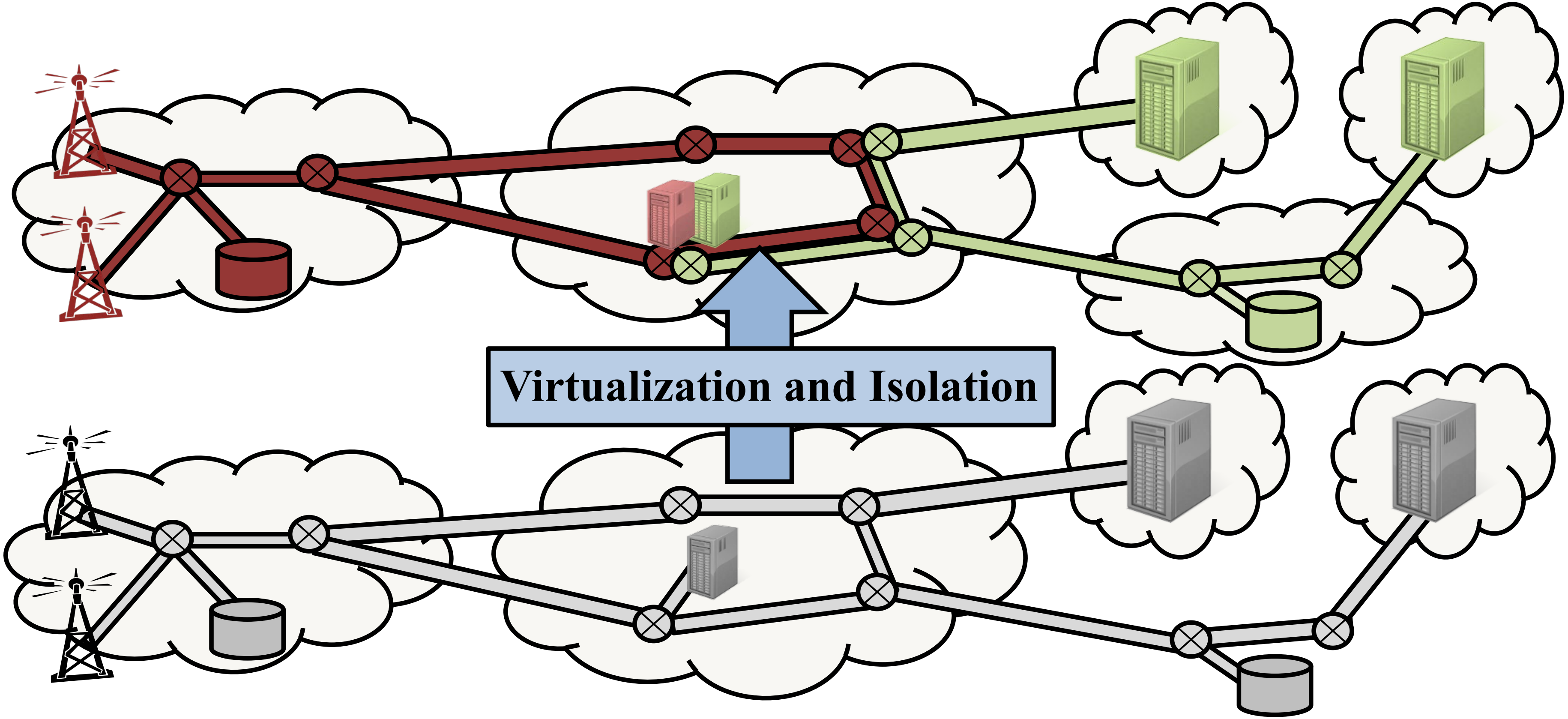




Case Study: Datacenter Network Virtualization

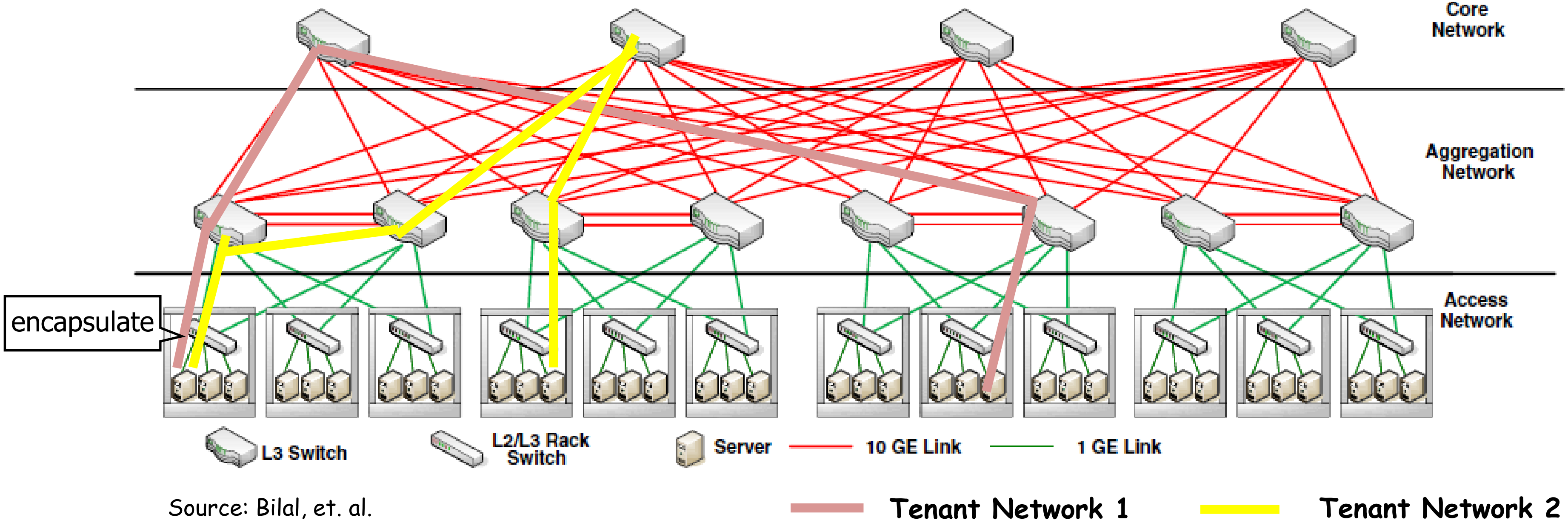


Security Requires Isolation on *All Levels*



State-of-the-Art Datacenter Networks

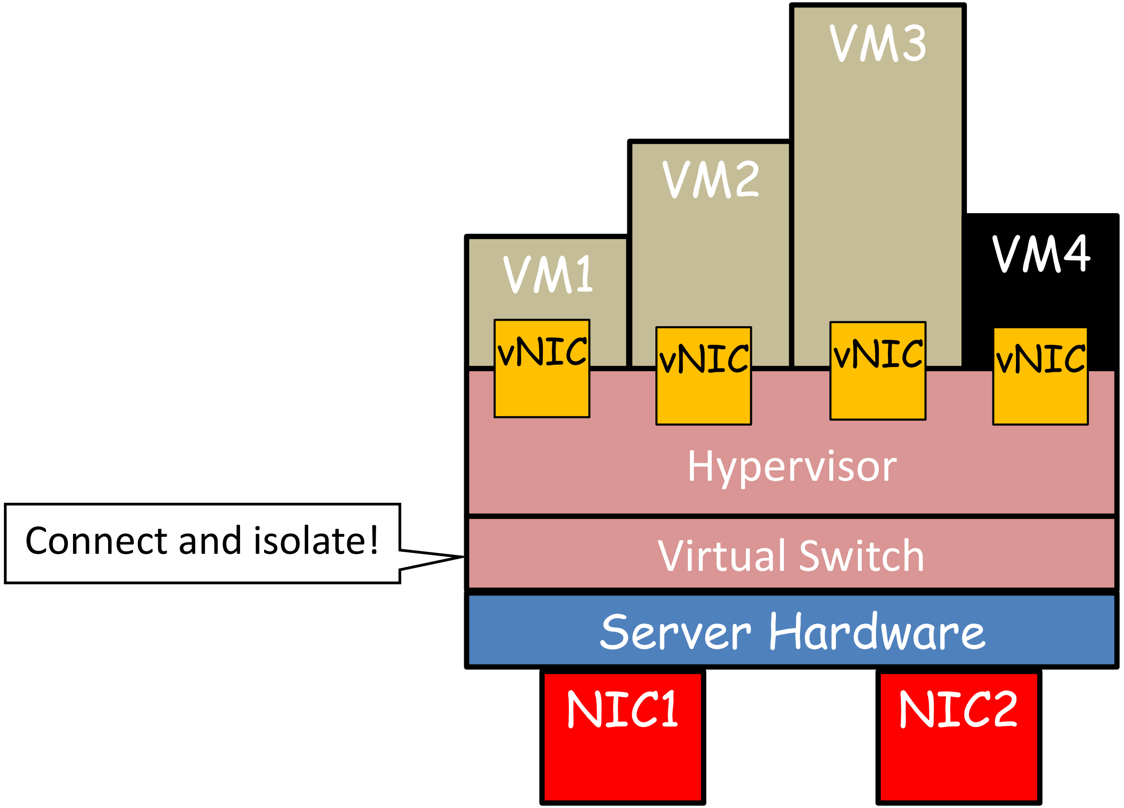
Network Virtualization Today: Tunneling



Source: Bilal, et. al.

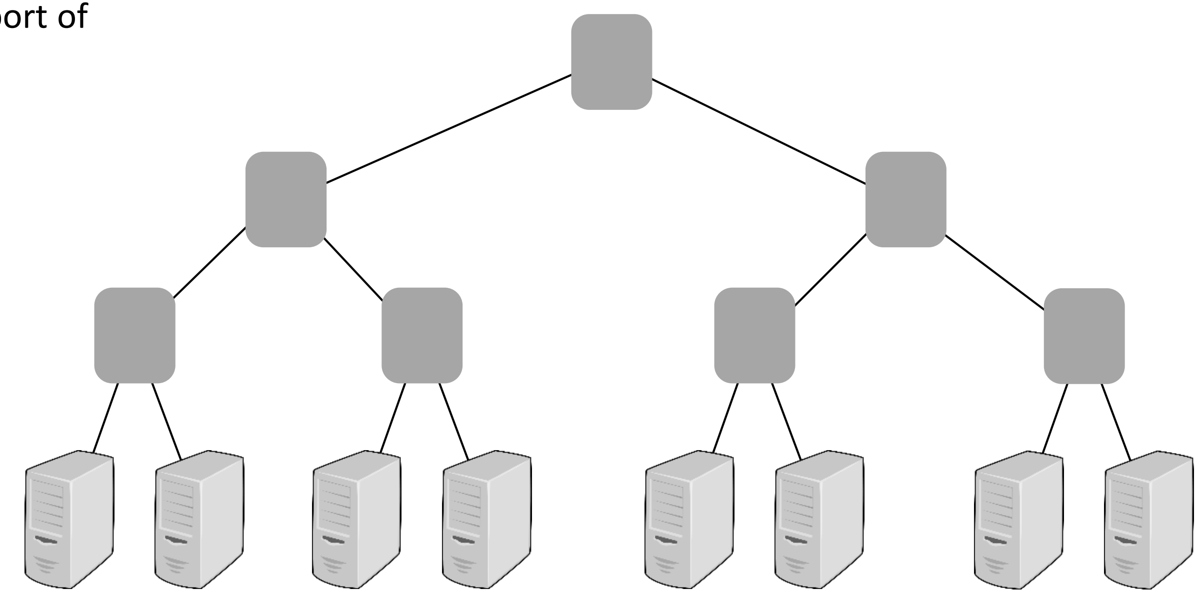
State-of-the-art: overlays, **tunneling** (e.g., **VxLAN**, VLAN, MPLS, …)

At the Heart: Virtual Switches, Networking VMs



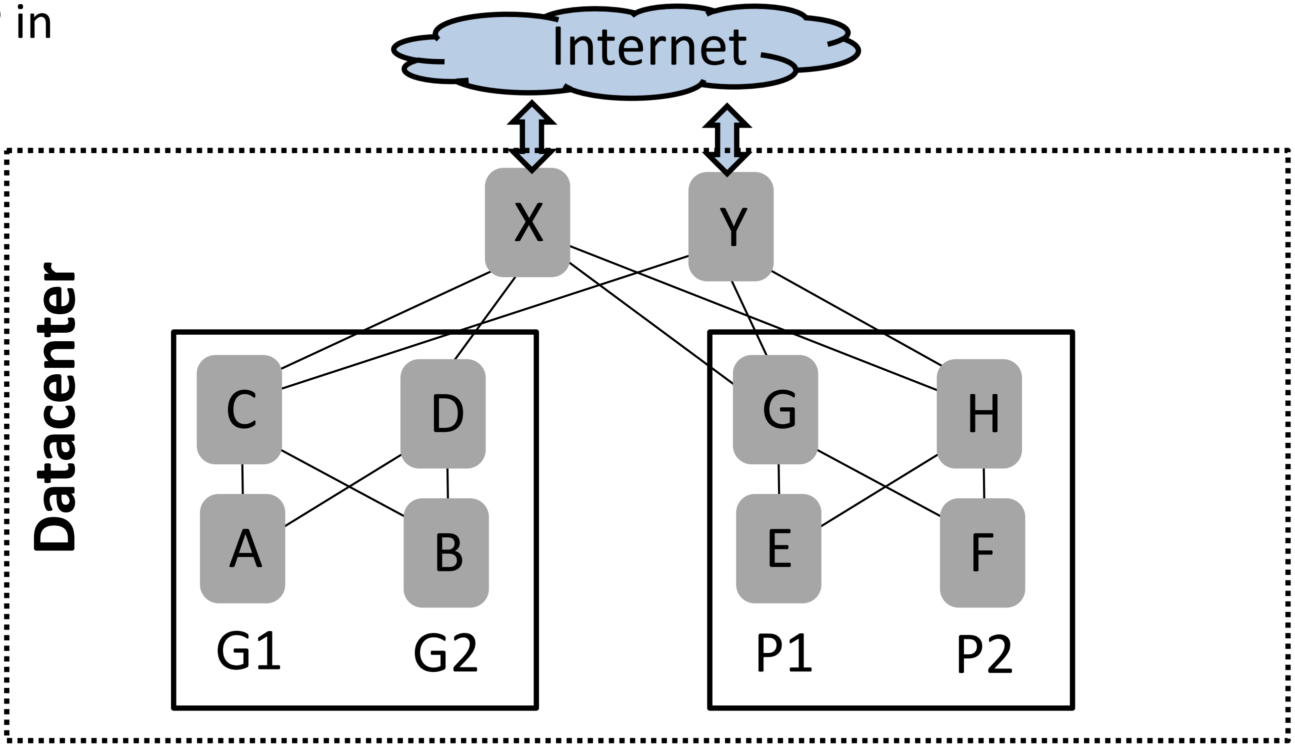
However, Today: Network Virtualization Complex and Inflexible

- Configuring tunnels/overlays today is *complex*, requiring *manual* work
- *Inflexible*, e.g., limited support of VM migration



Configuring Today's Networks is Hard: Case Study Microsoft Datacenter

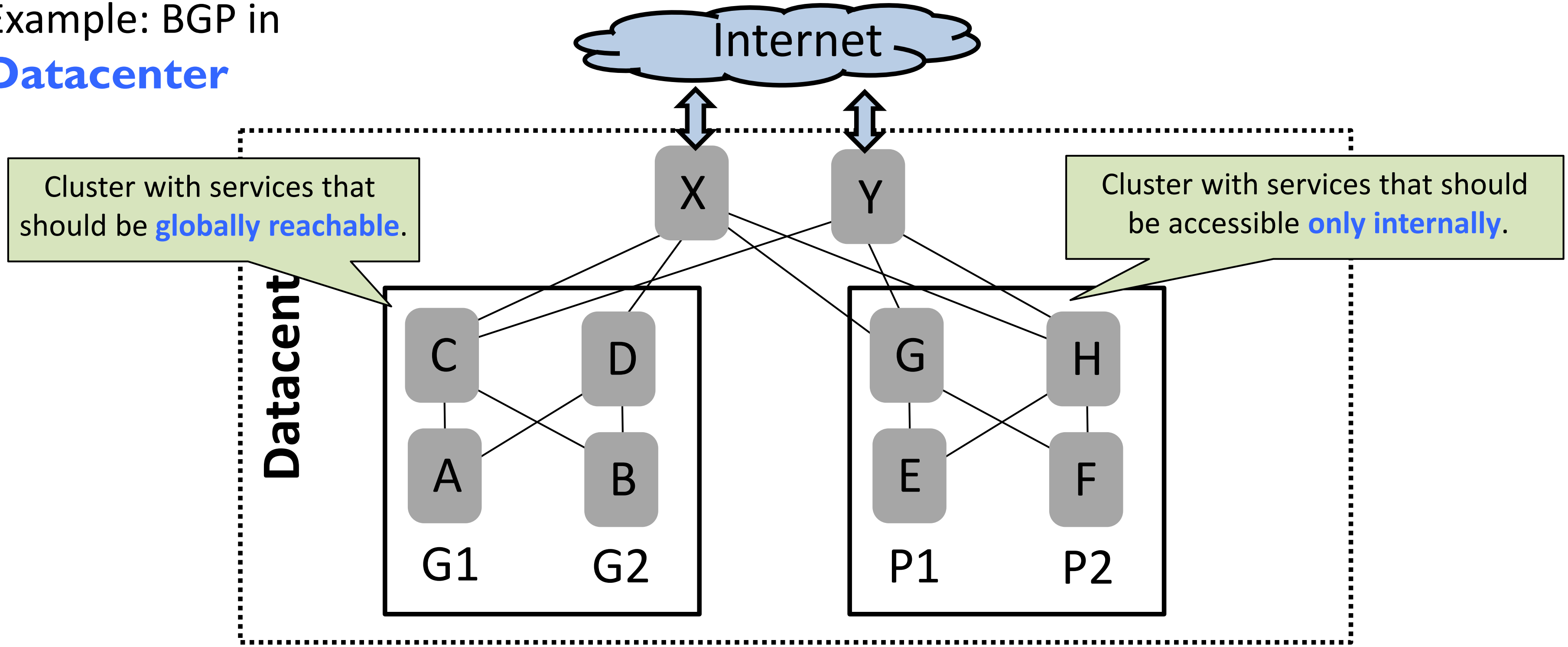
Example: BGP in **Datacenter**



Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.

Configuring Today's Networks is Hard: Case Study Microsoft Datacenter

Example: BGP in **Datacenter**

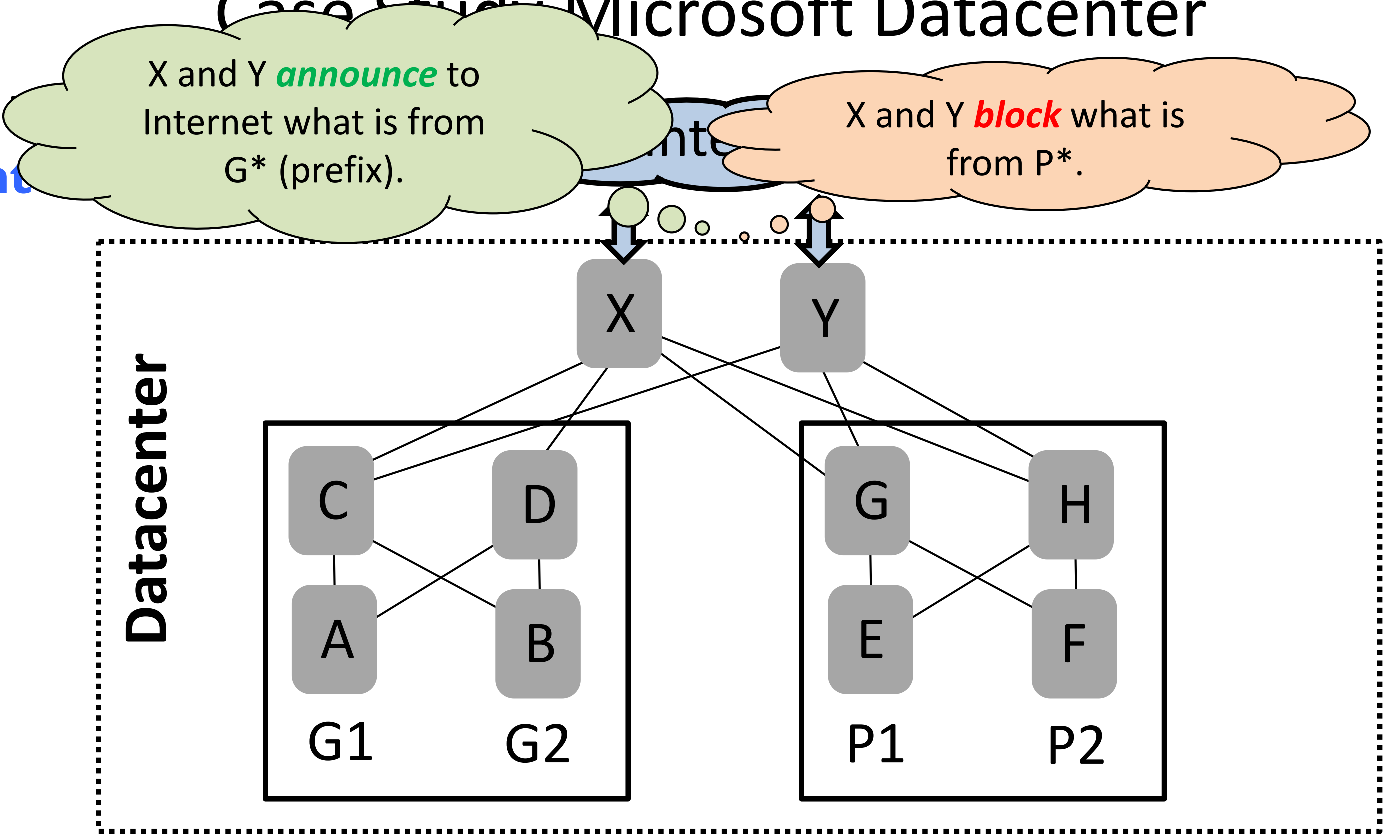


Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.

Configuring Today's Networks is Hard: Case Study: Microsoft Datacenter

Example:

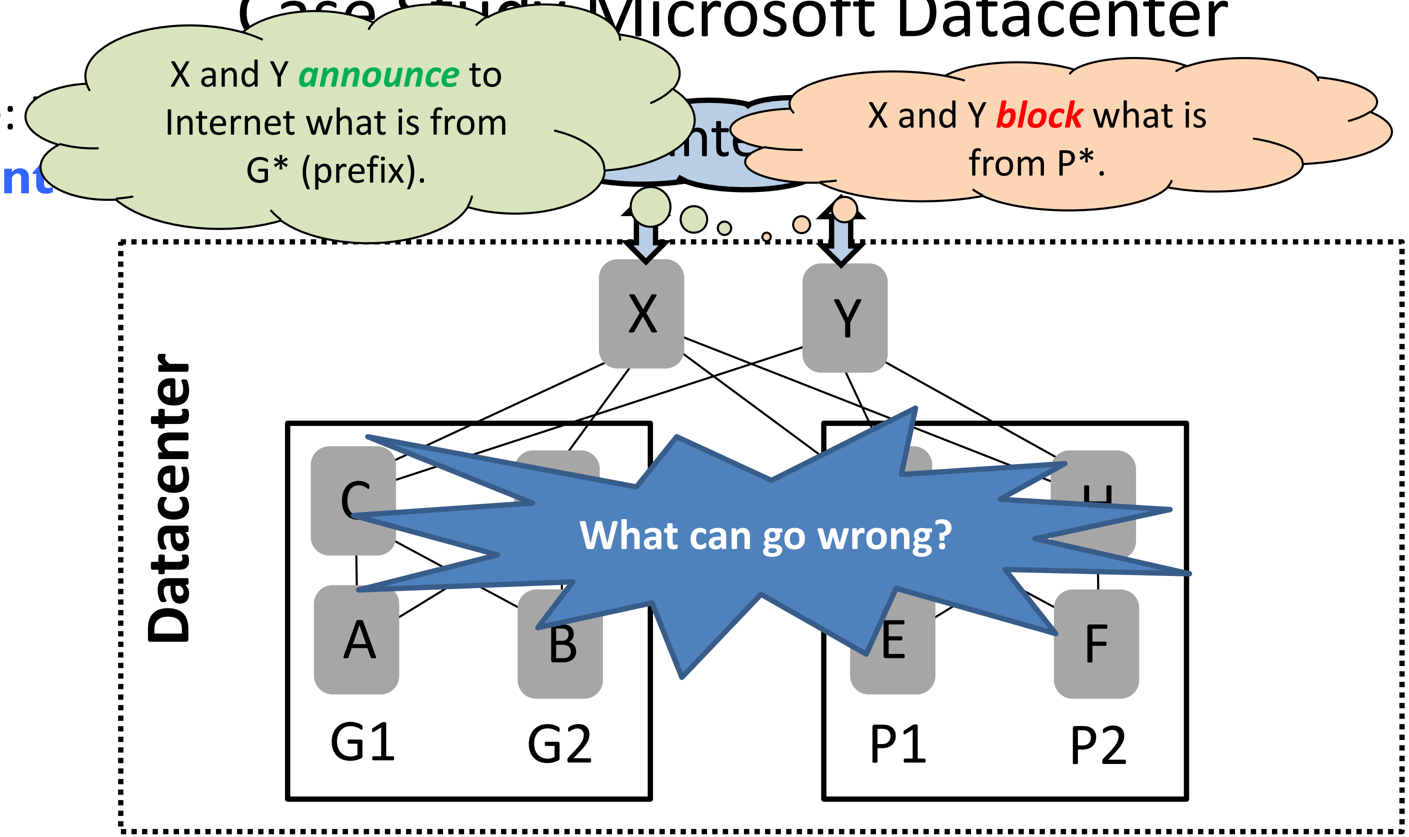
Datacent



Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.

Configuring Today's Networks is Hard: Case Study: Microsoft Datacenter

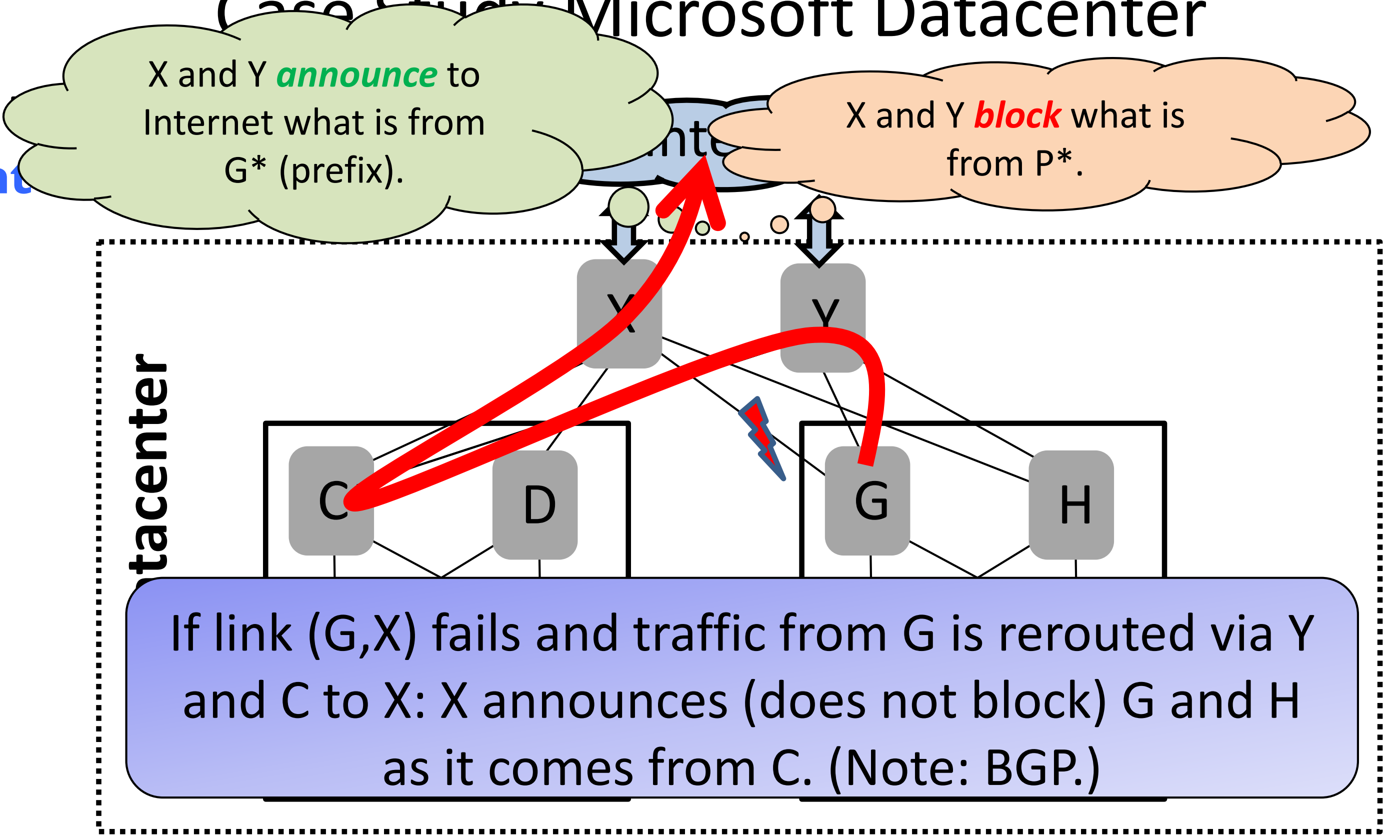
Example:
Datacent



Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.

Configuring Today‘s Networks is Hard: Case Study: Microsoft Datacenter

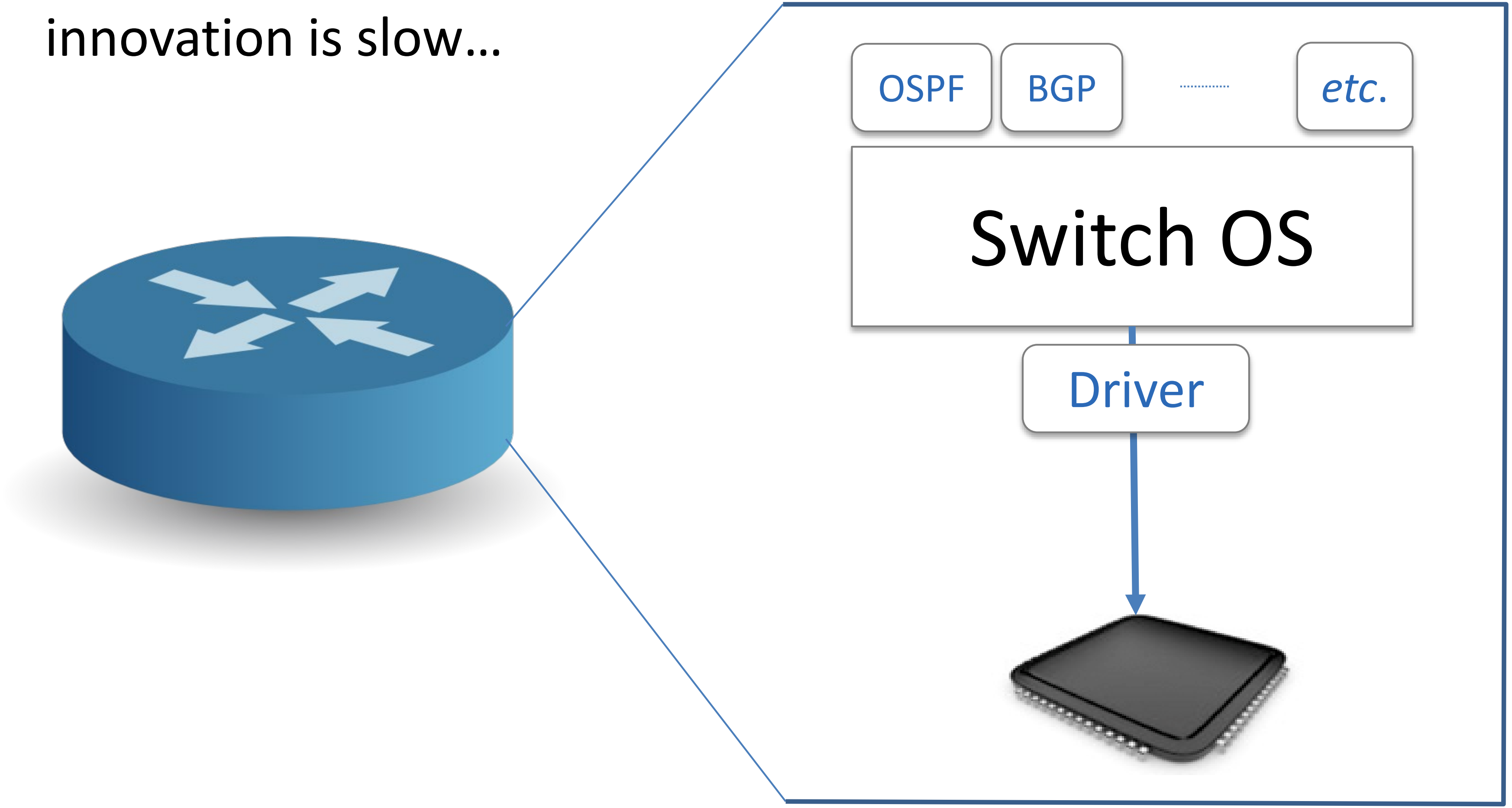
Example:
Datacent



If link (G,X) fails and traffic from G is rerouted via Y and C to X: X announces (does not block) G and H as it comes from C. (Note: BGP.)

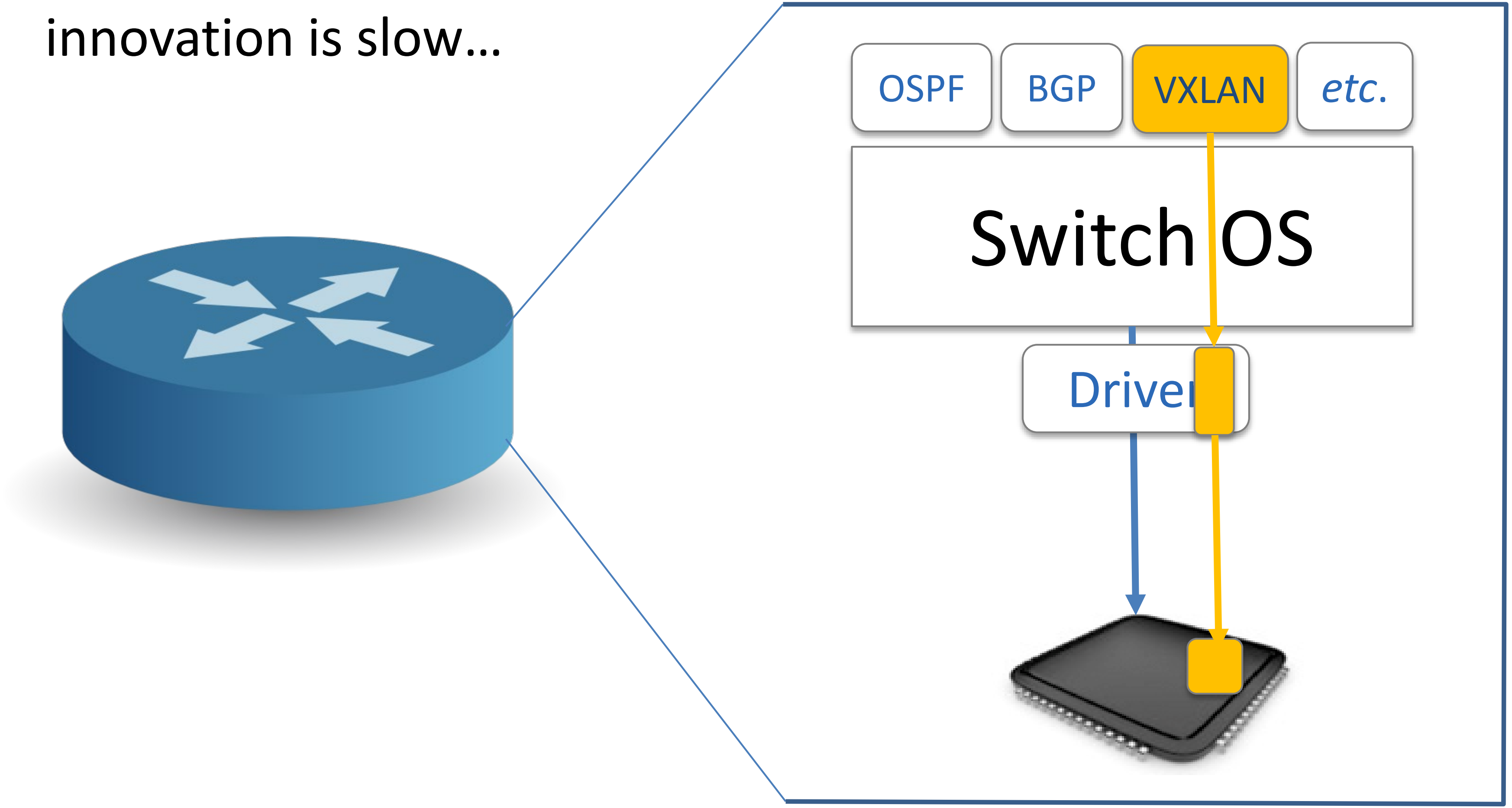
Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.

Another problem: innovation is slow…



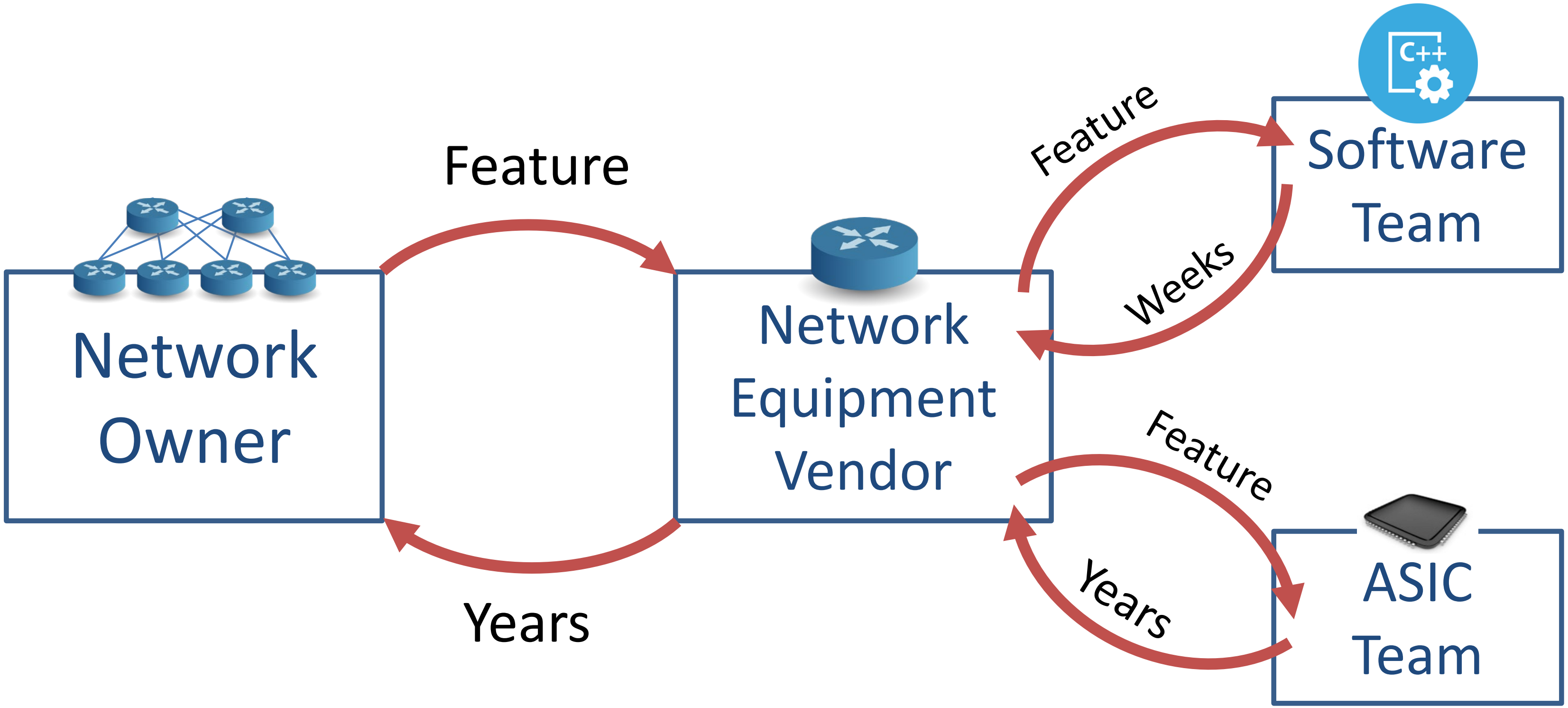
© Nick McKeown

Another problem: innovation is slow…



© Nick McKeown

VxLAN: Took Years…



© Nick McKeown

Slow Innovation…

Operator says:

I need extended VTP (VLAN Trunking Protocol) / a 3rd spanport etc. !

Vendor's answer:

Buy one of these!

Slow Innovation…

Operator says:

I need something better than STP for my data-center…

Vendor's answer:

We don't have that!

Opportunity: ?

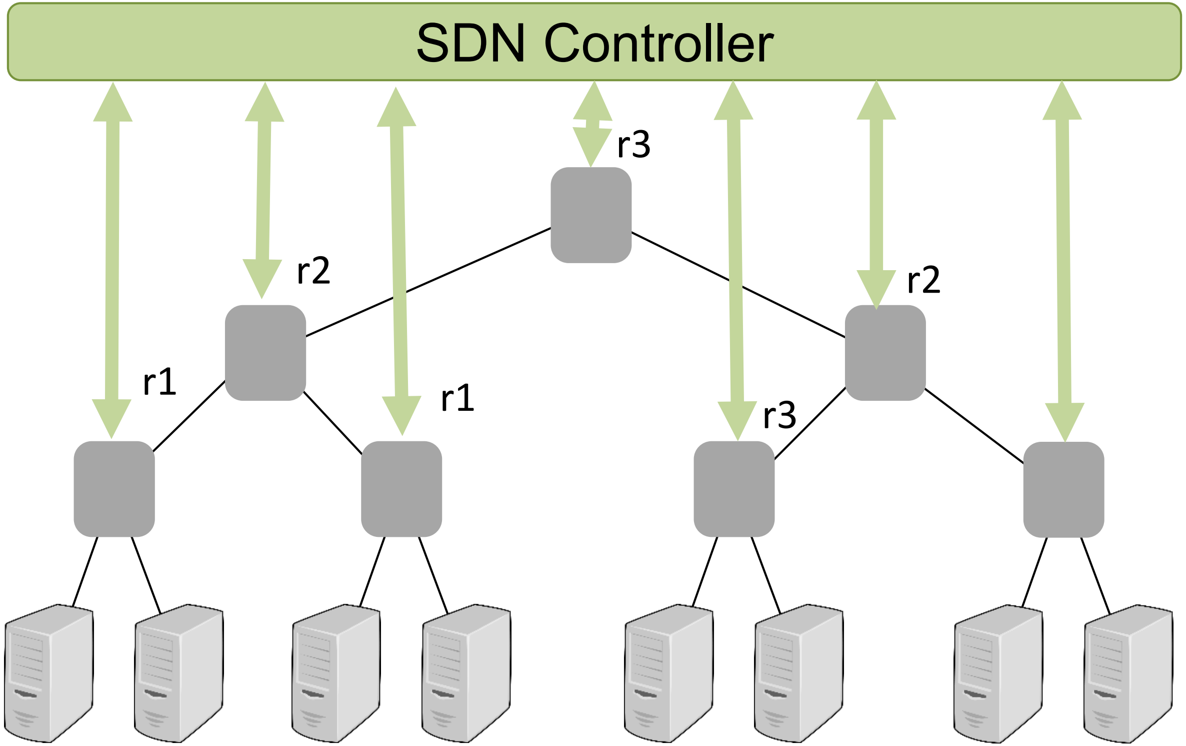
Introducing VxLAN:
matter of weeks

Opportunity: Programmability

Introducing VxLAN:
matter of weeks

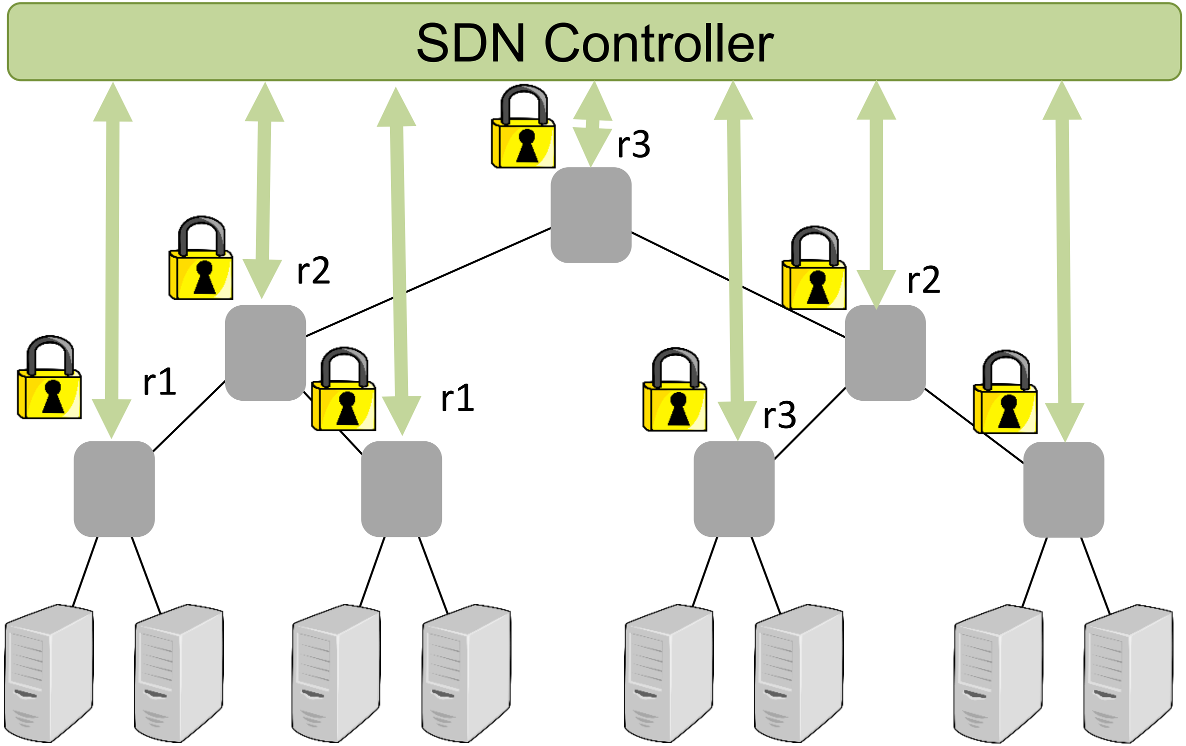
Software-Defined Networks (and Dataplanes)

- SDN = "The **Linux** of Networking"
  - *Open* interfaces
- **Centralized** and programmatic control
- Fine-grained control, lots of **flexibilities**
- ***Killer application***: network virtualization



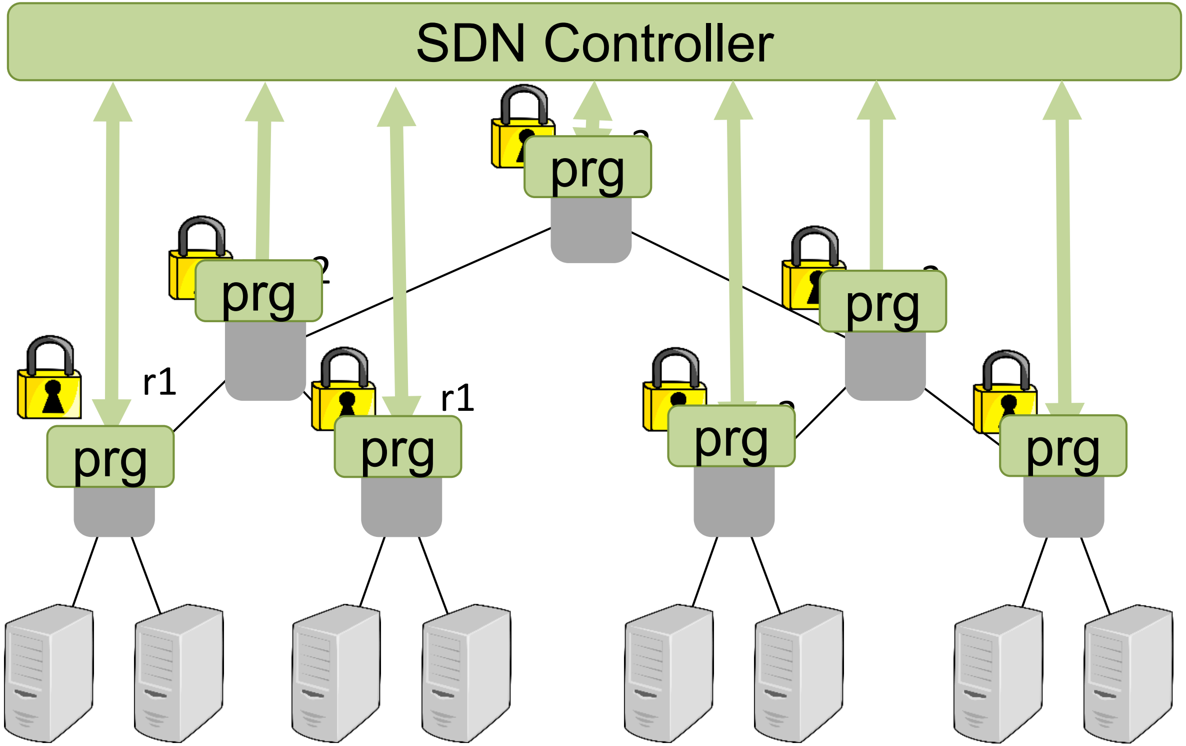
Software-Defined Networks (and Dataplanes)

- SDN = "The **Linux** of Networking"
  - *Open* interfaces
- **Centralized** and programmatic control
- Fine-grained control, lots of **flexibilities**
- ***Killer application***: network virtualization
- Secure communication



Emerging Software-Defined Networks

- SDN = "The **Linux** of Networking"
  - *Open* interfaces
- **Centralized** and programmatic control
- Fine-grained control, lots of **flexibilities**
- ***Killer application***: network virtualization
- Secure communication
- Also *programmable dataplane*
  - Packet processing pipeline
  - Introducing **VxLAN** easy!

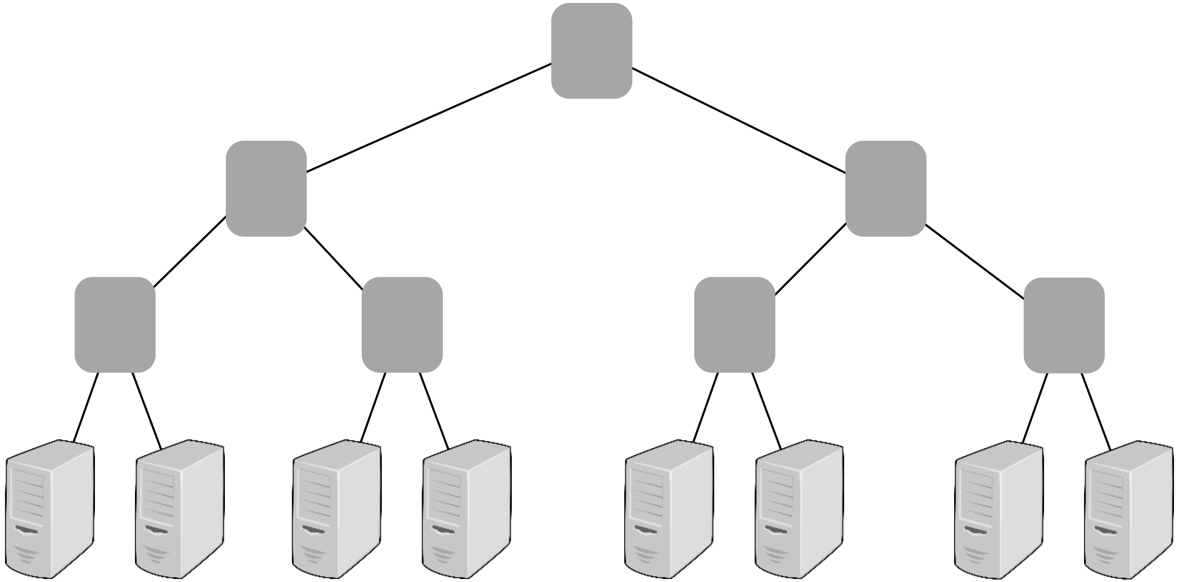


Allows to Deal with New Threat Vectors: Secure Trajectory Sampling

Monitor packets, traditionally:

trajectory sampling

- *Globally* sample packets with ***hash(imm. header)∈[x,y]***
- See full routes *of some packets*

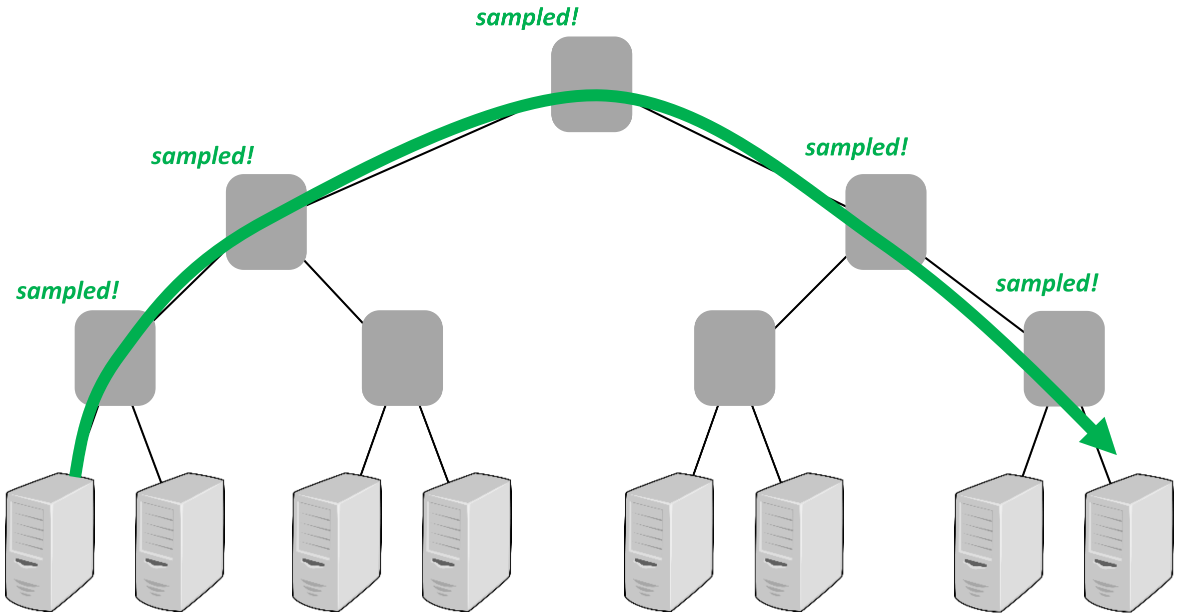


Allows to Deal with New Threat Vectors: Secure Trajectory Sampling

Monitor packets, traditionally:

trajectory sampling

- *Globally* sample packets with ***hash(imm. header)∈[x,y]***
- See full routes *of some packets*

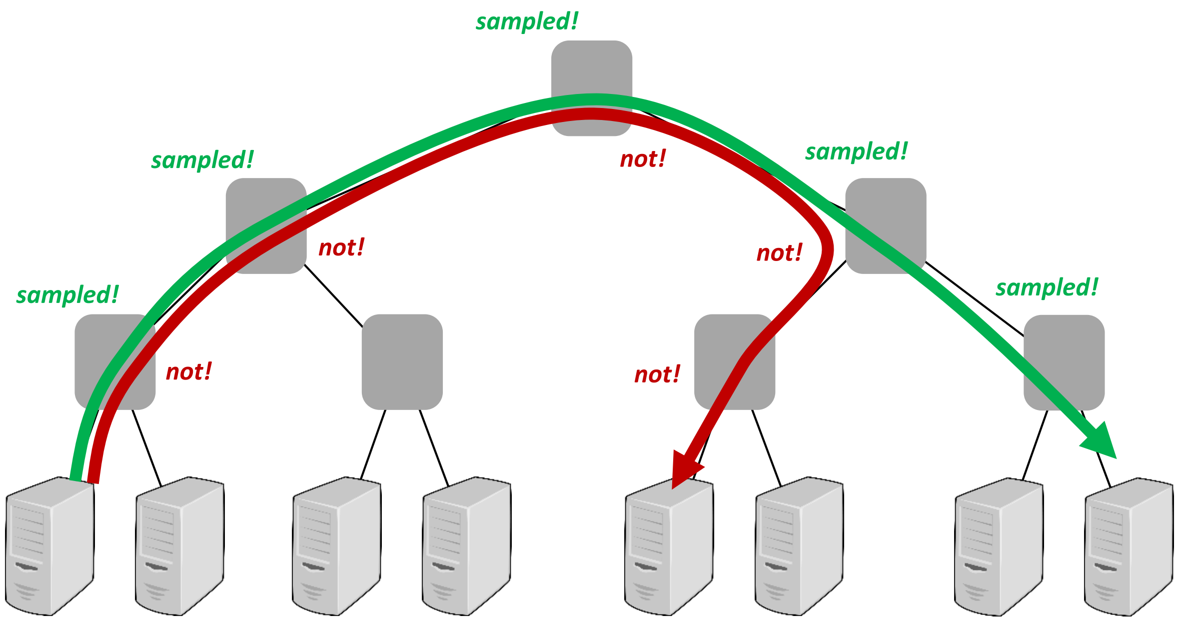


Allows to Deal with New Threat Vectors: Secure Trajectory Sampling

Monitor packets, traditionally:

trajectory sampling

- *Globally* sample packets with ***hash(imm. header)∈[x,y]***
- See full routes *of some packets*
- But *not others!* (resp. later)

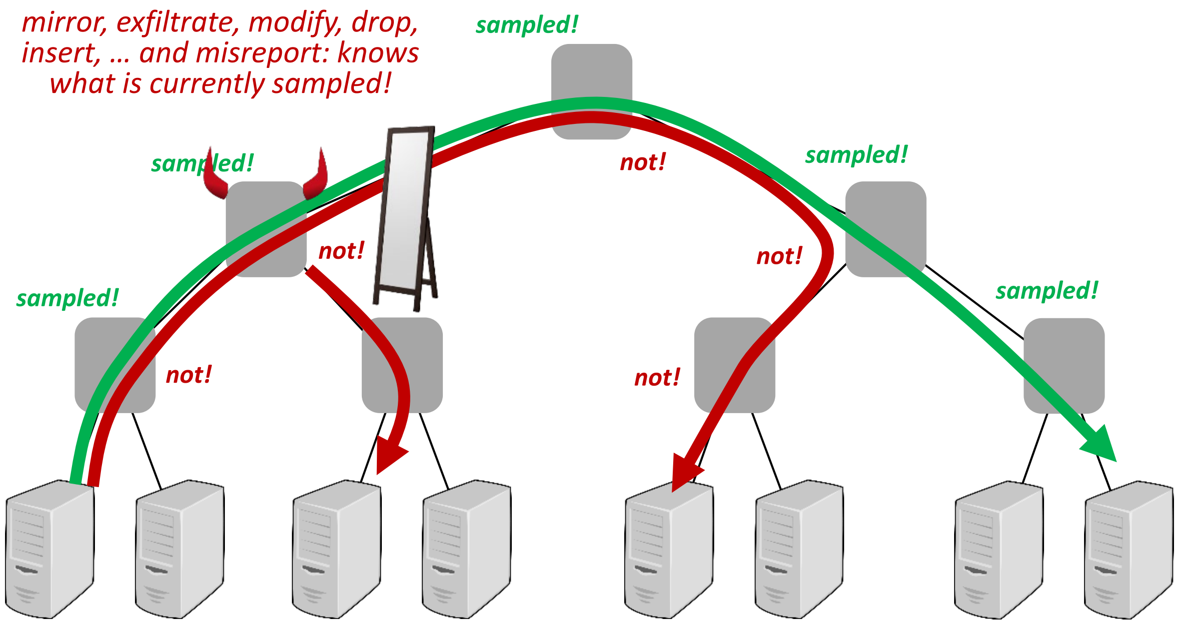


Allows to Deal with New Threat Vectors: Secure Trajectory Sampling

Monitor packets, traditionally:

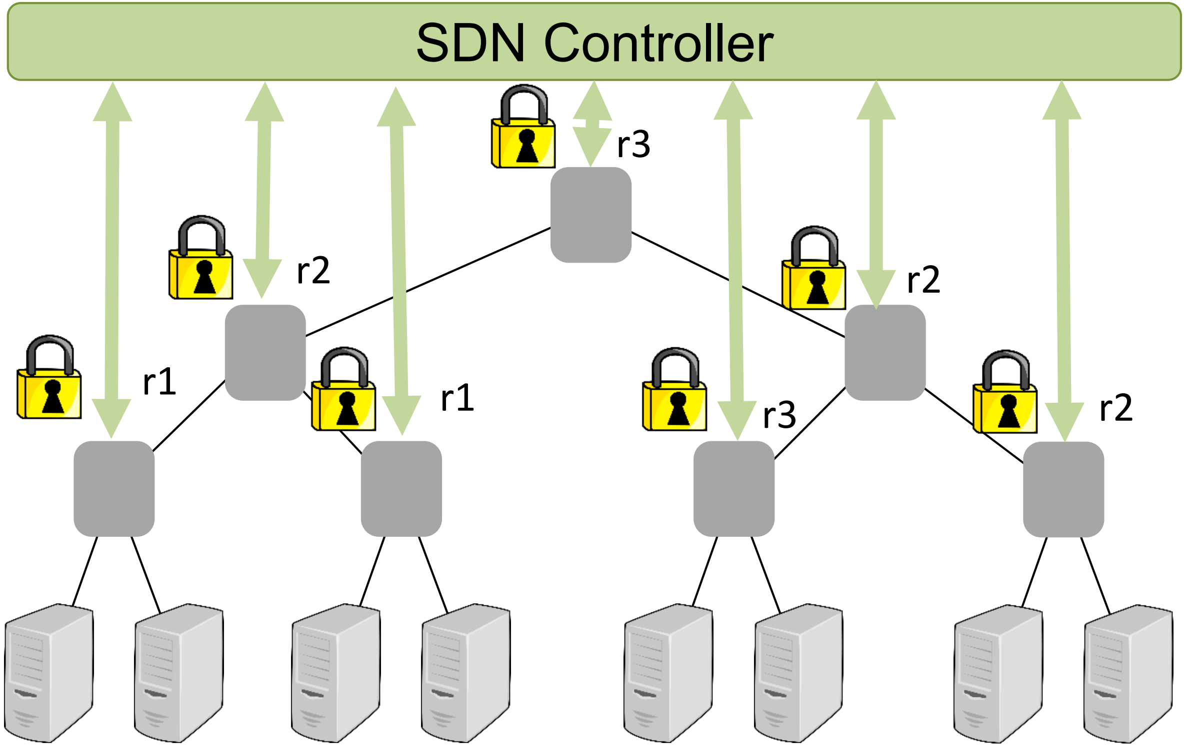
trajectory sampling

- *Globally* sample packets with ***hash(imm. header)∈[x,y]***
- See full routes *of some packets*
- But *not others!* (resp. later)



Solution: Use SDN for *Secure* Trajectory Sampling

- Idea:
  - Use ***secure*** channels between controller and switches to distribute hash ranges
  - Give ***different hash ranges*** hash ranges to different switches, but add some ***redundancy***: risk of being caught!

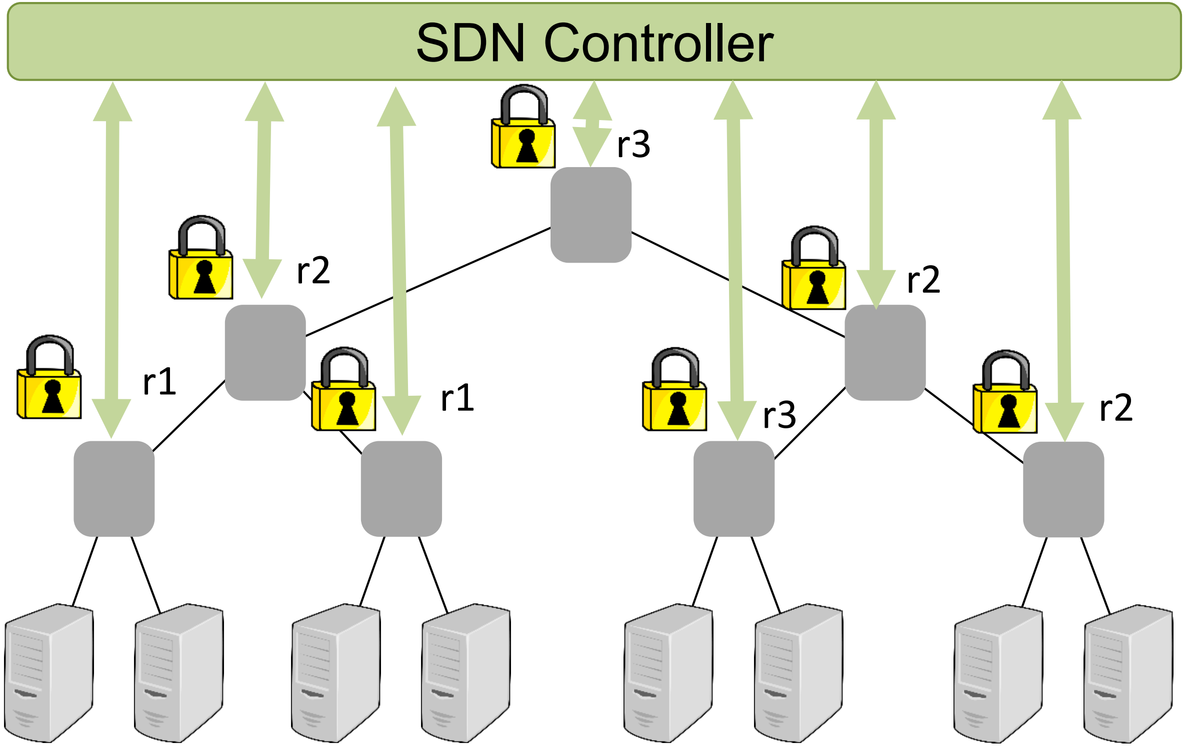


Network Policy Checker for Adversarial Environments.
Kashyap Thimmaraju, Liron Schiff, and S. SRDS 2019.

Solution: Use SDN for *Secure* Trajectory Sampling

- Idea:
  - Use ***secure*** channels between controller and switches to distribute hash ranges
  - Give ***different hash ranges*** hash ranges to different switches, but add some ***redundancy***: risk of being caught!

- In general: obtaining live data from the network ***becomes easier!***

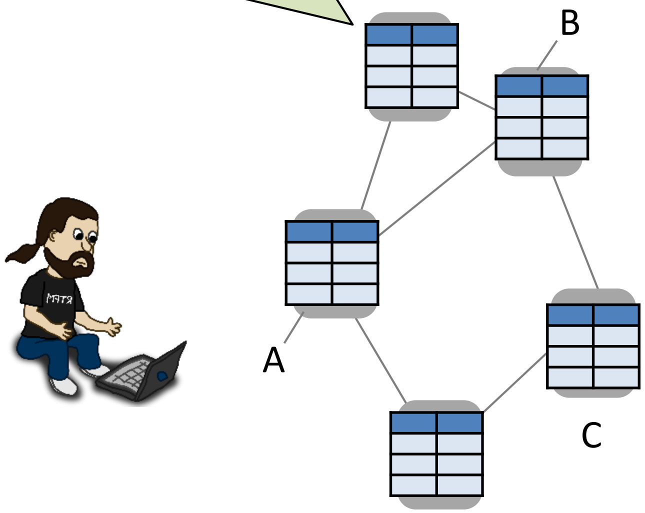


Network Policy Checker for Adversarial Environments. Kashyap Thimmaraju, Liron Schiff, and S. SRDS 2019.

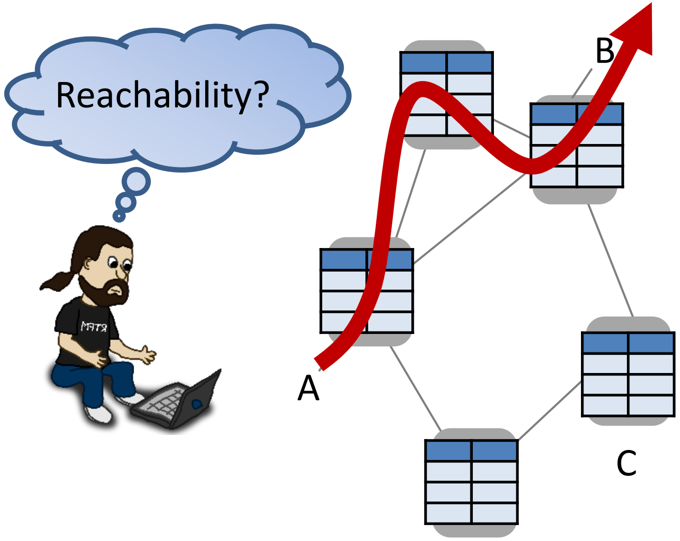
Opportunity: Automation

Responsibilities of a Sysadmin

Routers and switches store list of forwarding rules, and conditional failover rules.



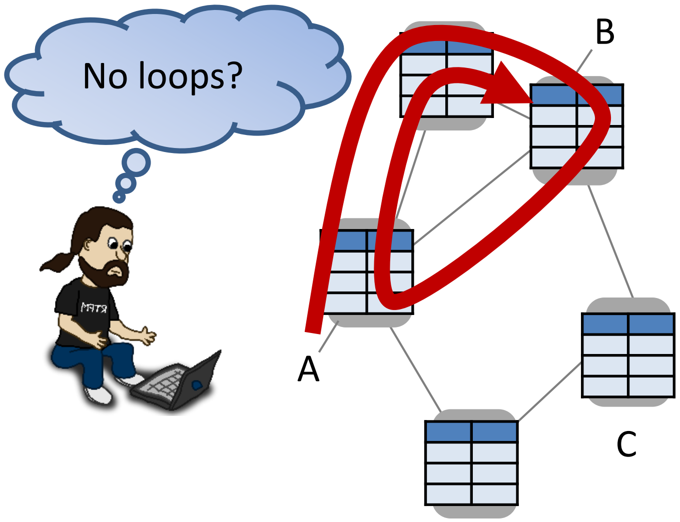
Responsibilities of a Sysadmin



Sysadmin responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?

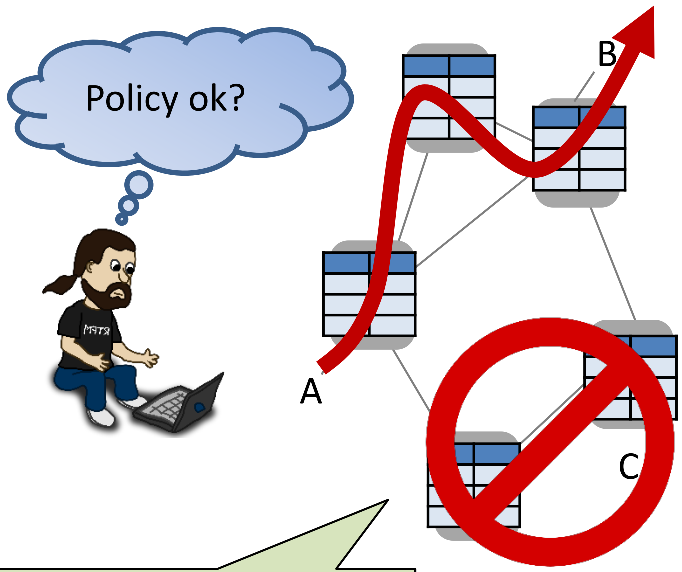
Responsibilities of a Sysadmin



Sysadmin 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?

Responsibilities of a Sysadmin

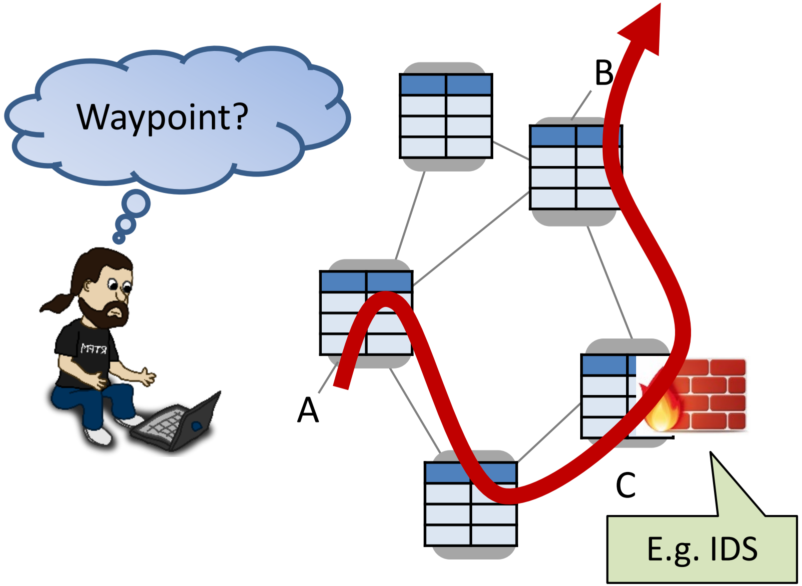


E.g. ***NORDUnet***: no traffic via Iceland (expensive!).

Sysadmin 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?

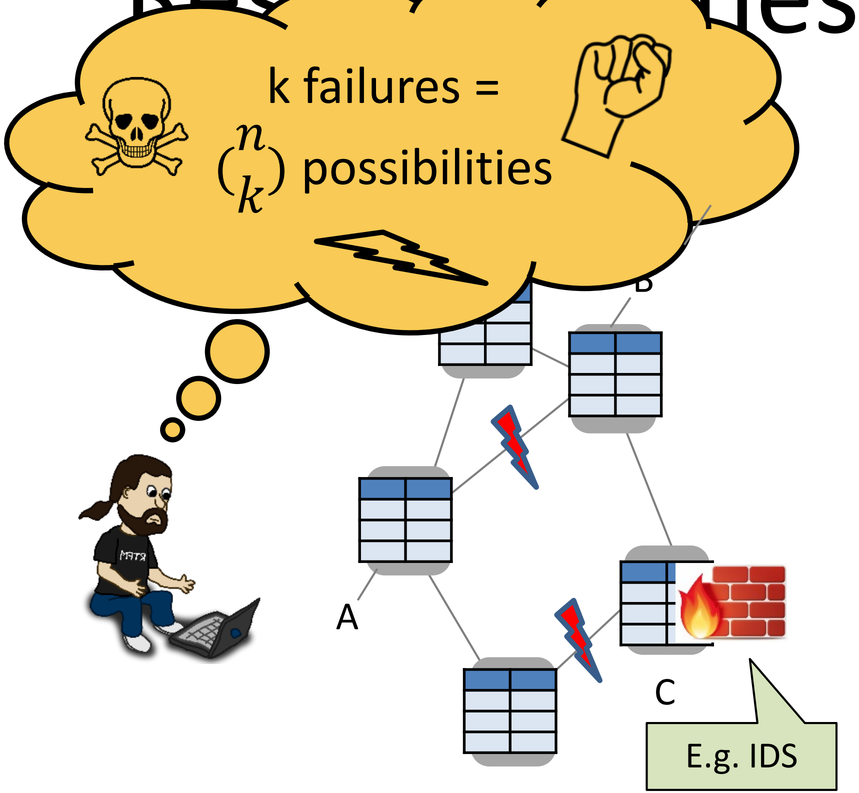
Responsibilities of a Sysadmin



Sysadmin 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)?

Responsibilities of a Sysadmin

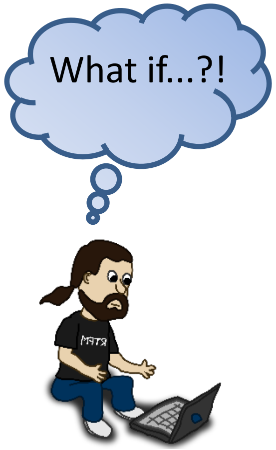


Sysadmin responsible for:

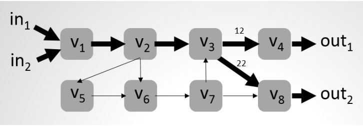
- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?
- **Policy:** Is it ensured that traffic from A to B never goes via C?
- **Waypoint enforcement:** Is it ensured that traffic from A to B is always routed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

… and everything even under multiple failures?!

Vision: Automation and Formal Methods

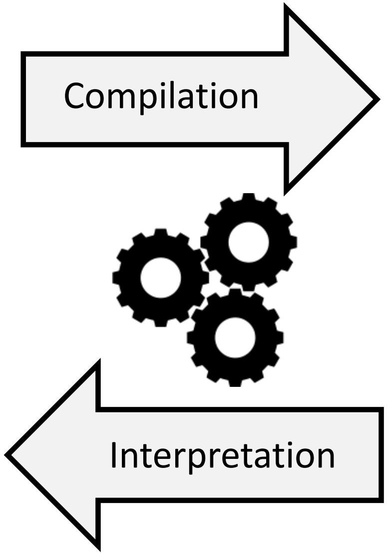


FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\bot$	(v_1, v_2)	$push(10)$
	in_2	$\bot$	(v_1, v_2)	$push(20)$
τ_{v_2}	(v_1, v_2)	10	(v_2, v_3)	$swap(11)$
	(v_1, v_2)	20	(v_2, v_3)	$swap(21)$
τ_{v_3}	(v_2, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_2, v_3)	21	(v_3, v_8)	$swap(22)$
	(v_7, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_7, v_3)	21	(v_3, v_8)	$swap(22)$
τ_{v_4}	(v_3, v_4)	12	out_1	pop
τ_{v_5}	(v_2, v_5)	40	(v_5, v_6)	pop
τ_{v_6}	(v_2, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	61	(v_6, v_7)	$swap(62)$
	(v_5, v_6)	71	(v_6, v_7)	$swap(72)$
τ_{v_7}	(v_6, v_7)	31	(v_7, v_3)	pop
	(v_6, v_7)	62	(v_7, v_3)	$swap(11)$
	(v_6, v_7)	72	(v_7, v_8)	$swap(22)$
τ_{v_8}	(v_3, v_8)	22	out_2	pop
	(v_7, v_8)	22	out_2	pop



local FFT	Out-I	In-Label	Out-I	op
τ_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$push(30)$
	(v_2, v_3)	21	(v_2, v_6)	$push(30)$
	(v_2, v_6)	30	(v_2, v_5)	$push(40)$

global FFT	Out-I	In-Label	Out-I	op
τ'_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$swap(61)$
	(v_2, v_3)	21	(v_2, v_6)	$swap(71)$
	(v_2, v_6)	61	(v_2, v_5)	$push(40)$
	(v_2, v_6)	71	(v_2, v_5)	$push(40)$



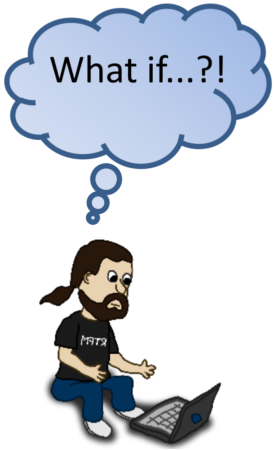
$$pX \Rightarrow qXX$$
$$pX \Rightarrow qYX$$
$$qY \Rightarrow rYY$$
$$rY \Rightarrow r$$
$$rX \Rightarrow pX$$

Router **configurations**, Segment Routing etc.

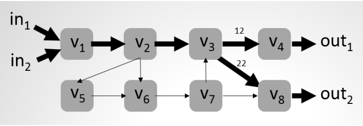
Pushdown Automaton and **Prefix Rewriting Systems** Theory

Vision: Automati ods

Use cases: Sysadmin ***issues queries*** to test certain properties, or do it on a ***regular basis*** automatically!

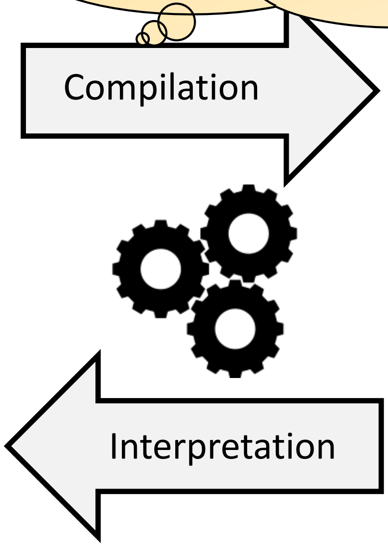


FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\bot$	(v_1, v_2)	$push(10)$
	in_2	$\bot$	(v_1, v_2)	$push(20)$
τ_{v_2}	(v_1, v_2)	10	(v_2, v_3)	$swap(11)$
	(v_1, v_2)	20	(v_2, v_3)	$swap(21)$
τ_{v_3}	(v_2, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_2, v_3)	21	(v_3, v_8)	$swap(22)$
	(v_7, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_7, v_3)	21	(v_3, v_8)	$swap(22)$
τ_{v_4}	(v_3, v_4)	12	out_1	pop
τ_{v_5}	(v_2, v_5)	40	(v_5, v_6)	pop
τ_{v_6}	(v_2, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	61	(v_6, v_7)	$swap(62)$
	(v_5, v_6)	71	(v_6, v_7)	$swap(72)$
τ_{v_7}	(v_6, v_7)	31	(v_7, v_3)	pop
	(v_6, v_7)	62	(v_7, v_3)	$swap(11)$
	(v_6, v_7)	72	(v_7, v_8)	$swap(22)$
τ_{v_8}	(v_3, v_8)	22	out_2	pop
	(v_7, v_8)	22	out_2	pop



local FFT	Out-I	In-Label	Out-I	op
τ_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$push(30)$
	(v_2, v_3)	21	(v_2, v_6)	$push(30)$
	(v_2, v_6)	30	(v_2, v_5)	$push(40)$
global FFT	**Out-I**	**In-Label**	**Out-I**	**op**
τ'_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$swap(61)$
	(v_2, v_3)	21	(v_2, v_6)	$swap(71)$
	(v_2, v_6)	61	(v_2, v_5)	$push(40)$
	(v_2, v_6)	71	(v_2, v_5)	$push(40)$

Router **configurations**, Segment Routing etc.

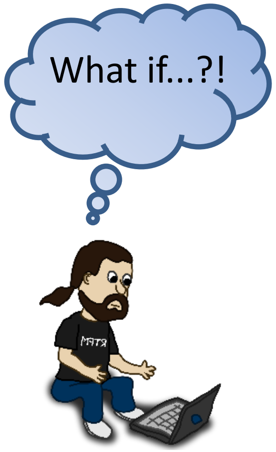


$$pX \Rightarrow qXX$$
$$pX \Rightarrow qYX$$
$$qY \Rightarrow rYY$$
$$rY \Rightarrow r$$
$$rX \Rightarrow pX$$

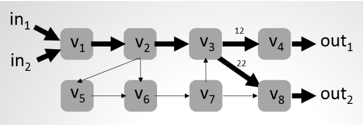
Pushdown Automaton and **Prefix Rewriting Systems** Theory

Vision: Automati… …ods

Use cases: Sysadmin ***issues queries*** to test certain properties, or do it on a ***regular basis*** automatically!



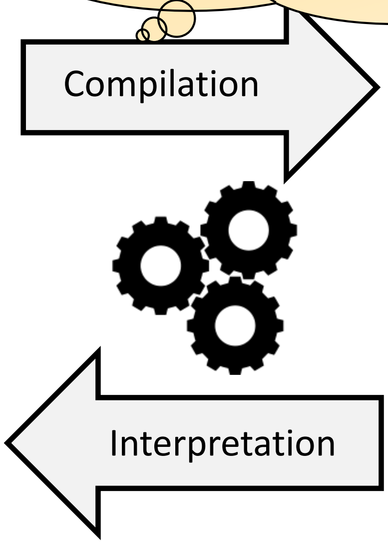
FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\bot$	(v_1, v_2)	$push(10)$
	in_2	$\bot$	(v_1, v_2)	$push(20)$
τ_{v_2}	(v_1, v_2)	10	(v_2, v_3)	$swap(11)$
	(v_1, v_2)	20	(v_2, v_3)	$swap(21)$
τ_{v_3}	(v_2, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_2, v_3)	21	(v_3, v_8)	$swap(22)$
	(v_7, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_7, v_3)	21	(v_3, v_8)	$swap(22)$
τ_{v_4}	(v_3, v_4)	12	out_1	pop
τ_{v_5}	(v_2, v_5)	40	(v_5, v_6)	pop
τ_{v_6}	(v_2, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	30	(v_6, v_7)	$swap(31)$
	(v_5, v_6)	61	(v_6, v_7)	$swap(62)$
	(v_5, v_6)	71	(v_6, v_7)	$swap(72)$
τ_{v_7}	(v_6, v_7)	31	(v_7, v_3)	pop
	(v_6, v_7)	62	(v_7, v_3)	$swap(11)$
	(v_6, v_7)	72	(v_7, v_8)	$swap(22)$
τ_{v_8}	(v_3, v_8)	22	out_2	pop
	(v_7, v_8)	22	out_2	pop



local FFT	Out-I	In-Label	Out-I	op
τ_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$push(30)$
	(v_2, v_3)	21	(v_2, v_6)	$push(30)$
	(v_2, v_6)	30	(v_2, v_5)	$push(40)$

global FFT	Out-I	In-Label	Out-I	op
τ'_{v_2}	(v_2, v_3)	11	(v_2, v_6)	$swap(61)$
	(v_2, v_3)	21	(v_2, v_6)	$swap(71)$
	(v_2, v_6)	61	(v_2, v_5)	$push(40)$
	(v_2, v_6)	71	(v_2, v_5)	$push(40)$

Router **configurations**, Segment Routing etc.



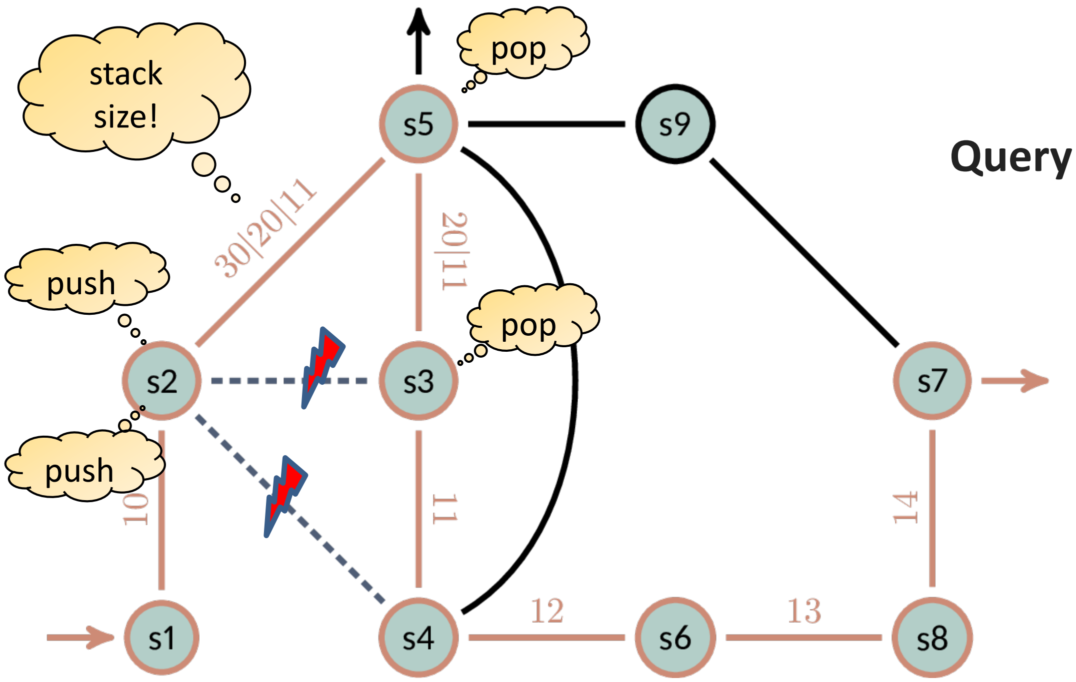
$$pX \Rightarrow qXX$$
$$pX \Rightarrow qYX$$
$$qY \Rightarrow rYY$$
$$rY \Rightarrow r$$
$$rX \Rightarrow pX$$

Pushdown Automaton and **Prefix Rewriting Systems** Theory

P-Rex: Fast Verification of MPLS Networks with Multiple Link Failures. Jensen et al., ACM CoNEXT, 2018.

Example: P-Rex for MPLS Networks

Can traffic starting with [] go through s5, under up to k=2 failures?



Query: k=2 [] s1 >> s5 >> s7 []

YES

(Polynomial time!)

Or Even: “Self-Driving Networks”?

- Networks could even automatically **troubleshoot** and fix themselves completely independently
- **Synthesis** of policy-compliant network configurations or even **self-optimize**: a case for *machine learning*?
- Disburdens human but *we give away control*: when to hand over back to human? Or **fall back** to „safe/oblivious mode“?
- Can we learn from self-driving **cars**? 

Or Even: “Self-Driving Networks”?

- Networks could even automatically **troubleshoot** and fix themselves completely independently

- **Synthesis** of policy-compliant network configurations or even **self-optimize**: a case for *machine learning*?

- Disburdens human but *we give away control*: when to hand over back to human? Or **fall back** to „safe/oblivious mode“?

- Can we learn from self-driving **cars**?

DeepMPLS: Fast Analysis of MPLS Configurations Using Deep Learning. Fabien Geyer and Stefan Schmid. IFIP Networking, Warsaw, Poland, May 2019.

Roadmap

- To what extent can we trust our networks today?

- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation

- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing

- Another uncharted security landscape: cryptocurrency networks

Roadmap

- To what extent can we trust our networks today?

- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation

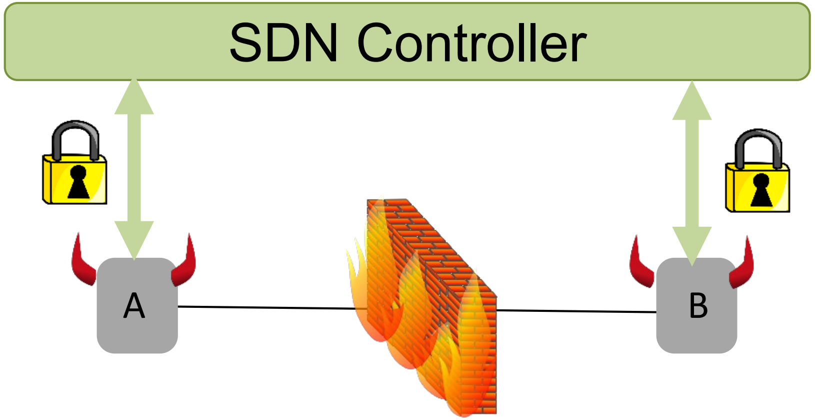
- **Challenge: emerging network technologies**
  - **New threat models**
  - **Algorithmic complexity attacks**
  - **AI-driven attacks and performance fuzzing**

- Another uncharted security landscape: cryptocurrency networks

Example 1: SDN

New Types of Attacks: Via SDN Controller

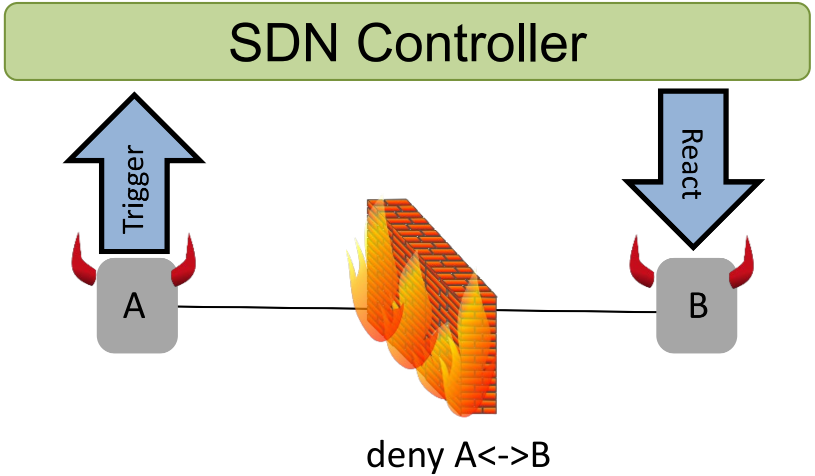
- **Controller** may be attacked or exploited



Outsmarting Network Security with SDN Teleportation
Kashyap Thimmaraju, Liron Schiff, and S.
EuroS&P, Paris, France, April 2017.

New Types of Attacks: Via SDN Controller

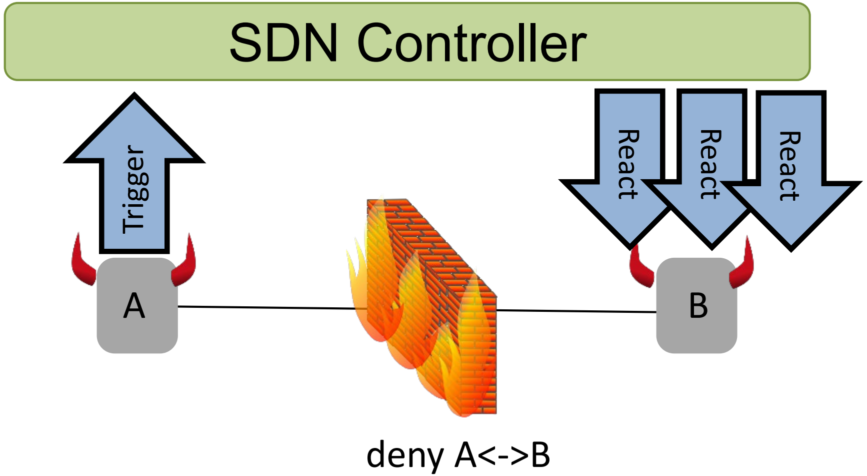
- **Controller** may be attacked or exploited
  - By design, *reacts* to switch events, e.g., by packet-outs



Outsmarting Network Security with SDN Teleportation
Kashyap Thimmaraju, Liron Schiff, and S.
EuroS&P, Paris, France, April 2017.

New Types of Attacks: Via SDN Controller

- **Controller** may be attacked or exploited
  - By design, *reacts* to switch events, e.g., by packet-outs
  - Or even *multicast*: **pave-path technique** more efficient than hop-by-hop

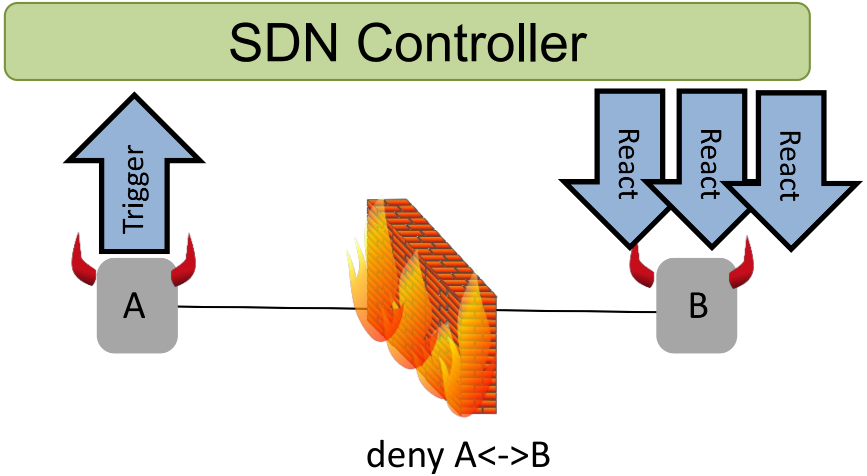


Outsmarting Network Security with SDN Teleportation
Kashyap Thimmaraju, Liron Schiff, and S.
EuroS&P, Paris, France, April 2017.

New Types of Attacks: Via SDN Controller

- **Controller** may be attacked or exploited
  - By design, *reacts* to switch events, e.g., by packet-outs
  - Or even *multicast*: **pave-path technique** more efficient than hop-by-hop

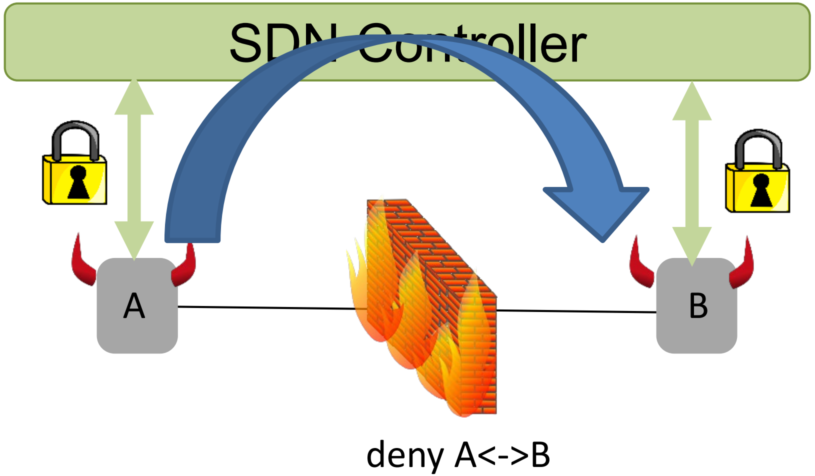
May introduce ***new communication paths*** which can be used in unintendend ways!



Outsmarting Network Security with SDN Teleportation
Kashyap Thimmaraju, Liron Schiff, and S.
EuroS&P, Paris, France, April 2017.

New Types of Attacks: Via SDN Controller

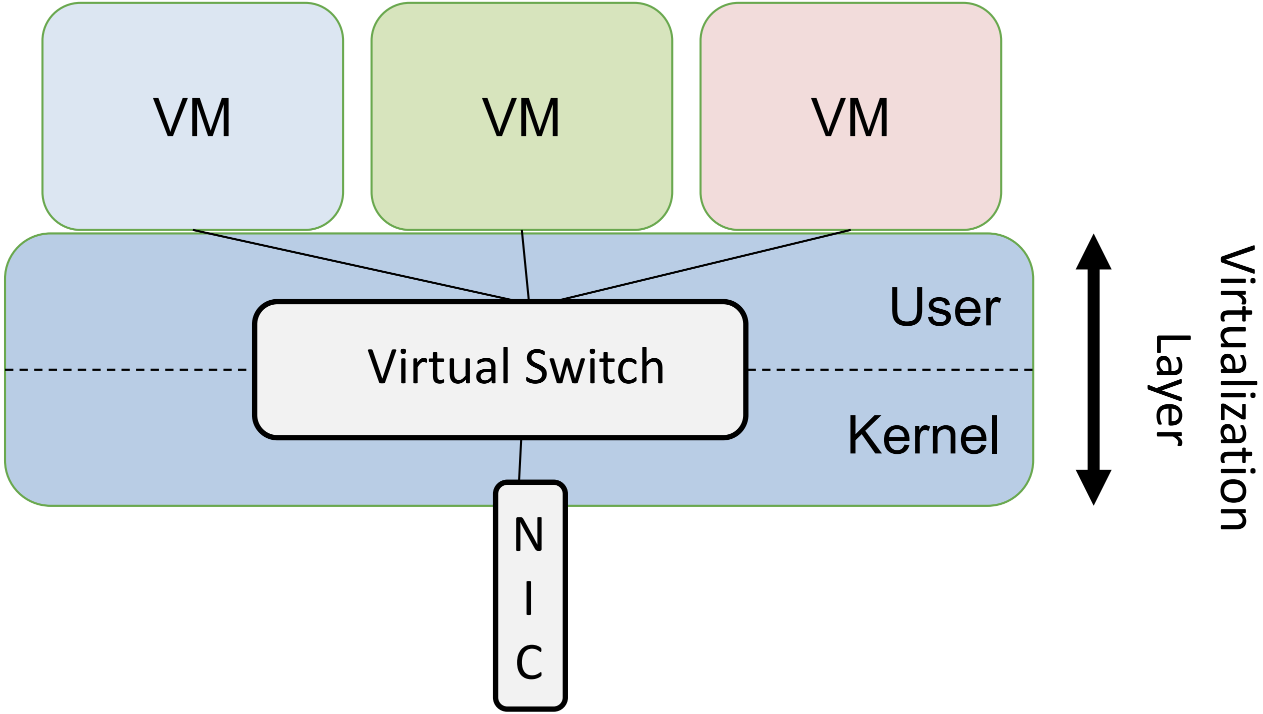
- In particular: new **covert communication** channels
  - E.g., exploit MAC learning (use codeword „0xBADDAD“) or modulate information with timing
- May *bypass security-critical elements*: e.g., firewall in the dataplane
- *Hard to catch*: along „normal communication paths“ and encrypted



Outsmarting Network Security with SDN Teleportation
Kashyap Thimmaraju, Liron Schiff, and S.
EuroS&P, Paris, France, April 2017.

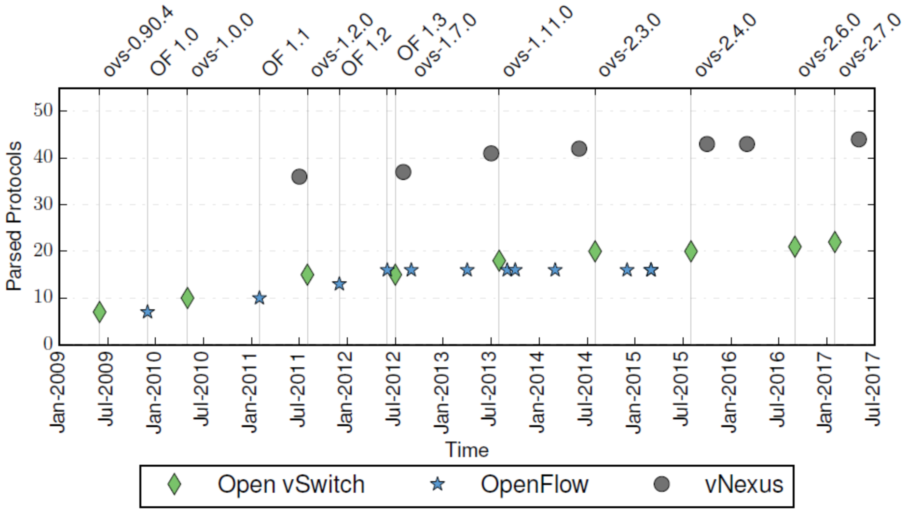
Example 2: Virtual Switch

Another New Vulnerability: Virtual Switch



Virtual switches reside in the **server's virtualization layer** (e.g., Xen's Dom0). Goal: provide connectivity and isolation.

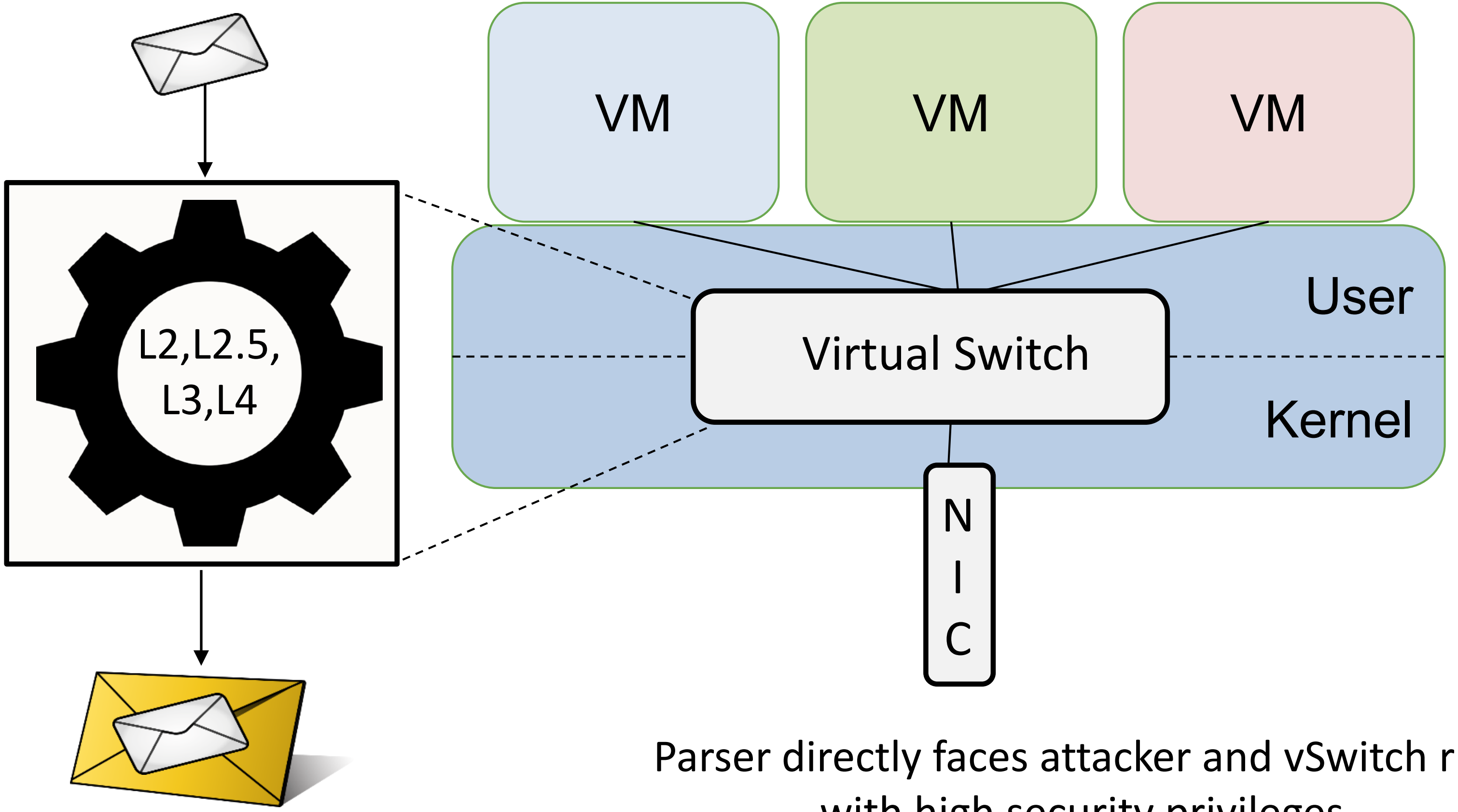
The Underlying Problem: Complexity



Number of parsed high-level protocols constantly increases…

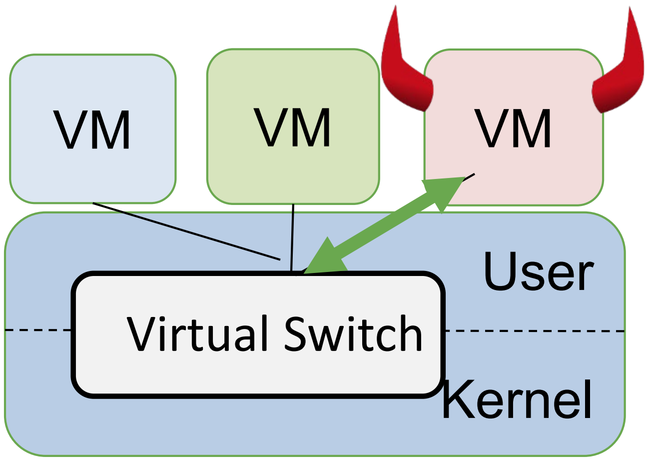
Complexity: Parsing

Ethernet
LLC
VLAN
MPLS
IPv4
ICMPv4
TCP
UDP
ARP
SCTP
IPv6
ICMPv6
IPv6 ND
GRE
LISP
VXLAN
PBB
IPv6 EXT HDR
TUNNEL-ID
IPv6 ND
IPv6 EXT HDR
IPv6HOPOPTS
IPv6ROUTING
IPv6Fragment
IPv6DESTOPT
IPv6ESP
IPv6 AH
RARP
IGMP

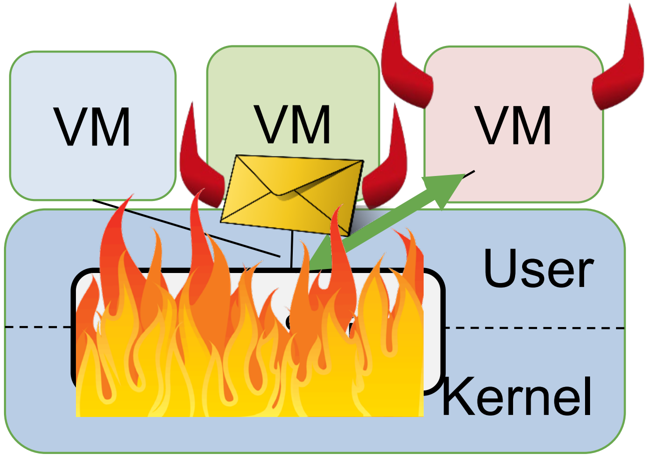


Parser directly faces attacker and vSwitch runs with high security privileges.

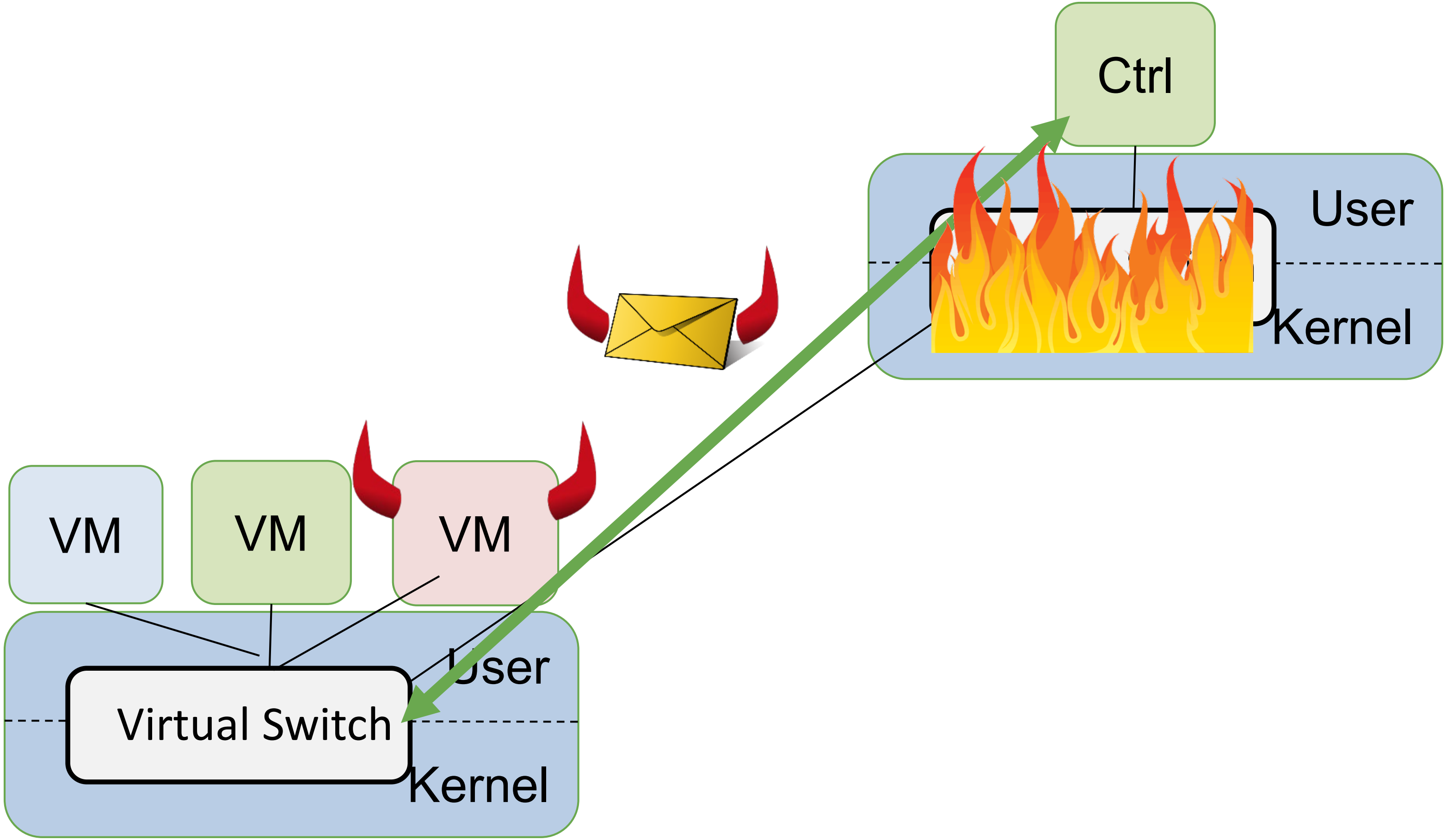
Enables Very Low-Cost Attacks



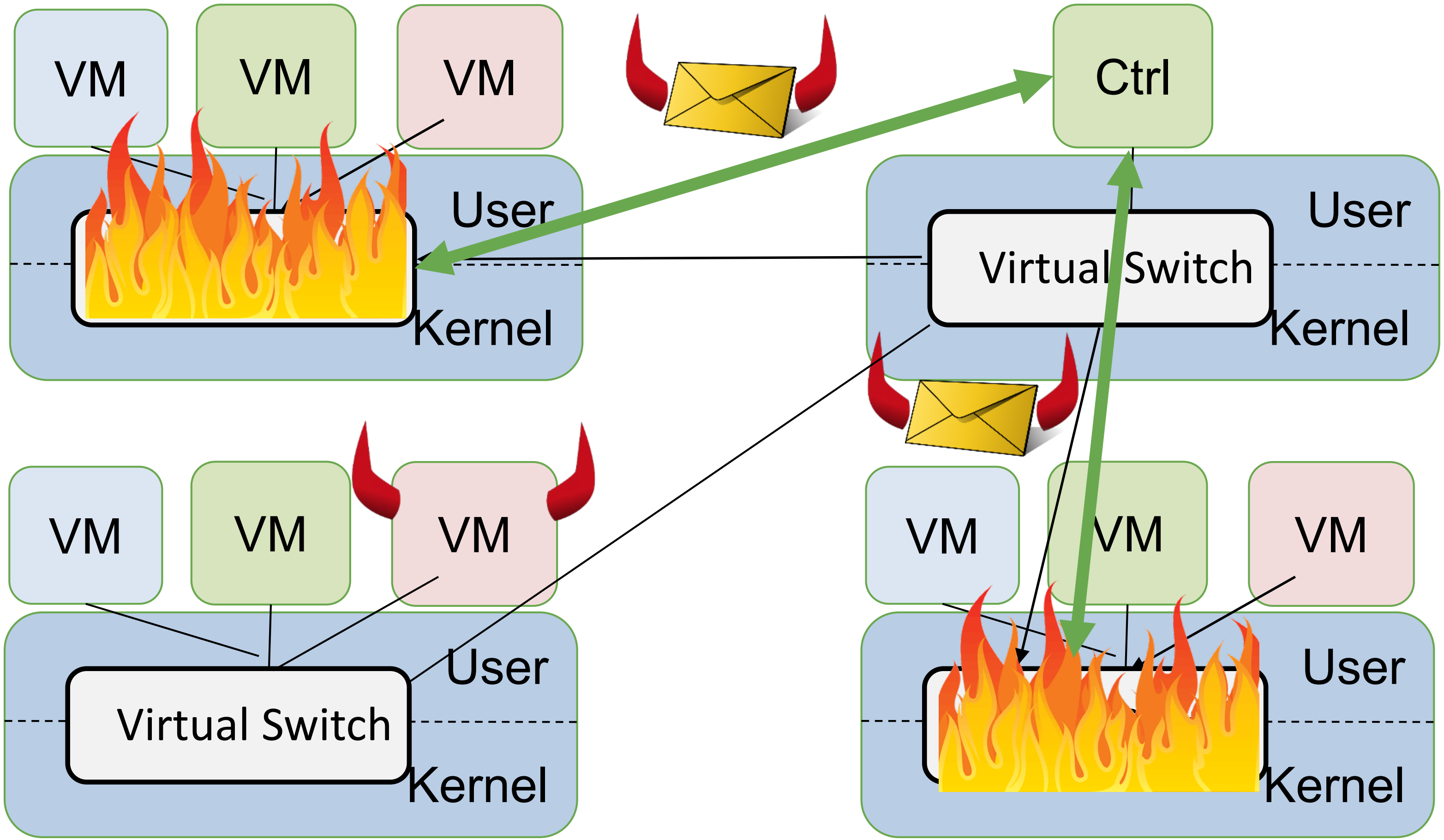
Enables Very Low-Cost Attacks



Enables Very Low-Cost Attacks

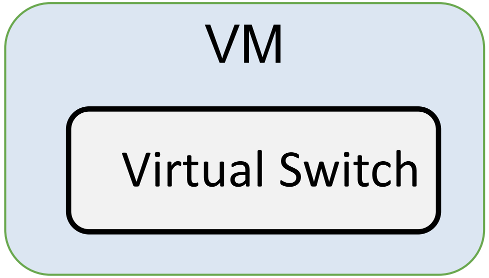


Enables Very Low-Cost Attacks



Challenge: How to provide better isolation *efficiently*?

- Idea for better *isolation*: put vSwitch in a VM
- But what about *performance*?
- Or container?

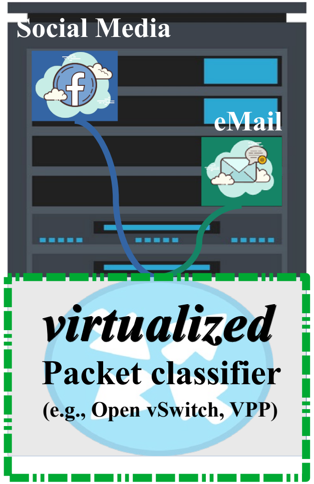


MTS: Bringing Multi-Tenancy to Virtual Switches
Kashyap Thimmaraju, Saad Hermak, Gabor Retvari, and S. USENIX ATC, 2019.

Example 3: Algorithmic Complexity Attacks

Algorithmic Complexity Attacks

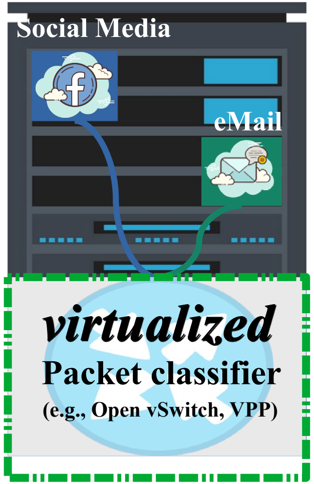
- Network dataplane runs many **complex algorithms**: may perform poorly under specific or *adversarial inputs*
- E.g., packet classifier: runs **Tuple Space Search** algorithm (e.g., in OVS)
- Can be exploited: adversary can *degrade performance* to ~10% of the baseline (10 Gbps) with only <1 Mbps (!) attack traffic
- Idea:
  - Tenants can use the Cloud Management System (CMS) to set up their **ACLs** to access-control, redirect, log, etc.
  - Attacker's goal: send some *packet towards the virtual switch* that when subjected to the ACLs will *exhaust resources*



Tuple Space Explosion: A Denial-of-Service Attack Against a Software Packet Classifier. Levente Csikor et al. ACM CoNEXT, 2019.

Algorithmic Complexity Attacks

- Network dataplane runs many **complex algorithms**: may perform poorly under specific or *adversarial inputs*
- E.g., packet classifier: runs **Tuple Space Search** algorithm (e.g., in OVS)
- Can be exploited: adversary can *degrade performance* to ~10% of the baseline (10 Gbps) with only <1 Mbps (!) attack traffic
- Idea:
  - Tenants can use the Cloud Management System (CMS) to set up their **ACLs** to access-control, redirect, log, etc.
  - Attacker's goal: send some *packet towards the virtual switch* that when subjected to the ACLs will *exhaust resources*



How to find such attacks?!

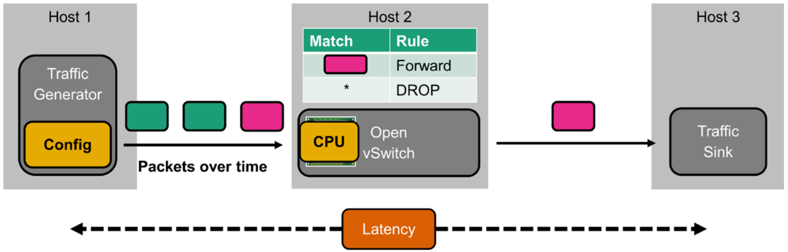
Tuple Space Explosion: A Denial-of-Service Attack Against a Software Packet Classifier. Levente Csikor et al. ACM CoNEXT, 2019.

Example 4: AI-Driven Attacks

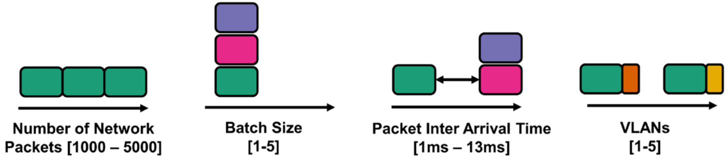
(Or: Automated Identification of Complexity Attacks)

NetBOA: Automated Performance Benchmarking

- Idea: *automate*! Generate different input, measure impact (e.g., latency)
  - Similar to *fuzzing*



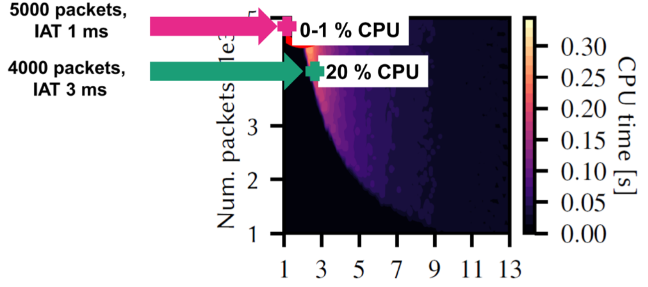
- Different dimensions:
  - Packet size, inter-arrival time, packet type, etc.



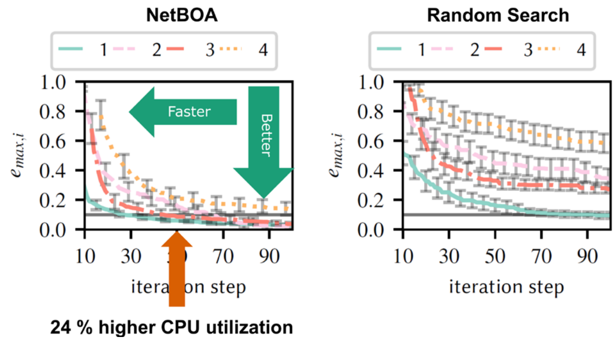
NetBOA: Self-Driving Network Benchmarking
Zerwas et al. ACM SIGCOMM Workshop on Network Meets AI & ML (NetAI), Beijing, China, August 2019.

Baysian Optimization Approach

- Complex systems (such as vSwitch) have complex behavior: e.g., sometimes sending less packets increases CPU load
  - Hard to find for humans



- Baysian optimization much faster than random baseline



Roadmap

- To what extent can we trust our networks today?
- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation
- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing
- Another uncharted security landscape: cryptocurrency networks

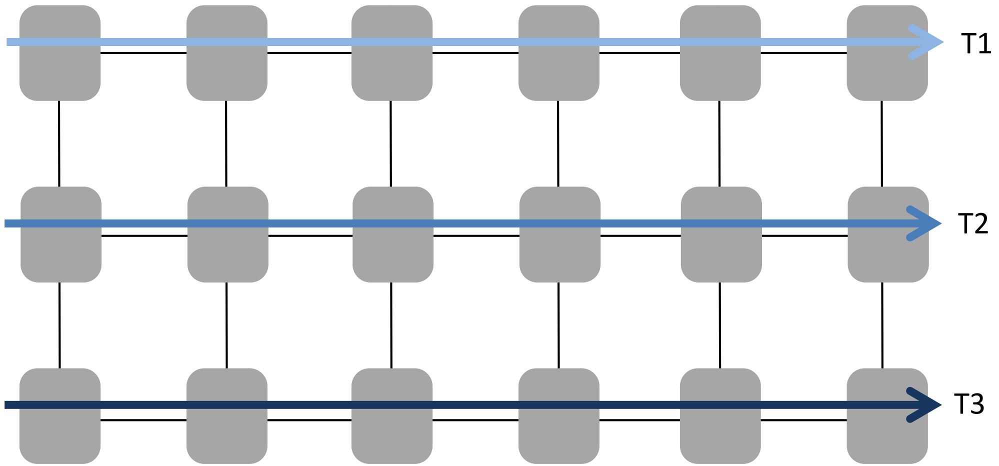
Roadmap

- To what extent can we trust our networks today?
- Opportunity: emerging network technologies
  - Programmability and virtualization
  - „Self-driving networks“ and automation
- Challenge: emerging network technologies
  - New threat models
  - Algorithmic complexity attacks
  - AI-driven attacks and performance fuzzing
- **Another uncharted security landscape: cryptocurrency networks**

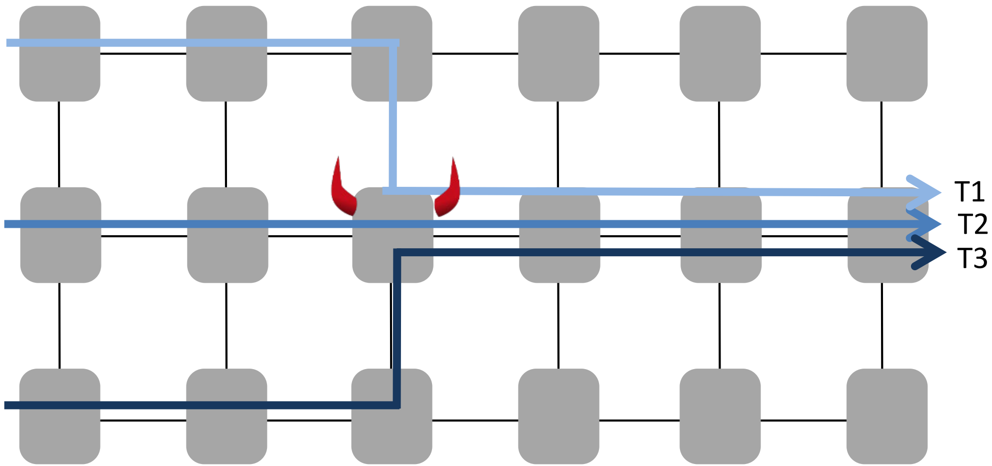
Example: Offchain Networks

- Novel networks to improve **scalability of Bitcoin** and other cryptocurrencies
- E.g., Lightning, Raven, Ripple, …
- But also *uncharted security landscape*

Attracting Transaction Routes

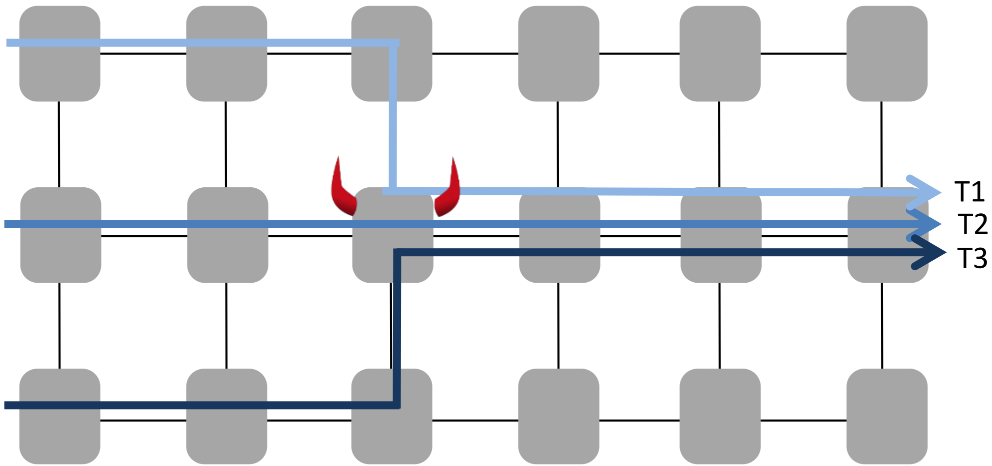


Attracting Transaction Routes



By *announcing low fees*, can attract and *stop* significant fraction of transactions on offchain networks!

Attracting Transaction Routes



By *announcing low fees*, can attract and *stop* significant fraction of transactions on offchain networks!

Hijacking Routes in Payment Channel Networks: A Predictability Tradeoff. Saar Tochner, Stefan Schmid, Aviv Zohar. Arxiv 2019

Or Attack Confidentiality (@ICISSP2020)

Toward Active and Passive Confidentiality Attacks On Cryptocurrency Off-Chain Networks

Utz Nisslmueller[1], Klaus-Tycho Foerster[1], Stefan Schmid[1], and Christian Decker[2]
[1] *Faculty of Computer Science, University of Vienna, Vienna, Austria*
[2] *Blockstream, Zurich, Switzerland*

Keywords: Cryptocurrencies, Bitcoin, Payment Channel Networks, Routing, Privacy

Abstract: Cryptocurrency off-chain networks such as Lightning (e.g., Bitcoin) or Raiden (e.g., Ethereum) aim to increase the scalability of traditional on-chain transactions. To support nodes to learn about possible paths to route their transactions, these networks need to provide gossip and probing mechanisms. This paper explores whether these mechanisms may be exploited to infer sensitive information about the flow of transactions, and eventually harm privacy. In particular, we identify two threats, related to an active and a passive adversary. The first is a *probing attack:* here the adversary aims the maximum amount which is transferable in a given direction of a target channel, by active probing. The second is a *timing attack:* the adversary discovers how close the destination of a routed payment actually is, by acting as a passive man-in-the middle. We then analyze the limitations of these attacks and propose remediations for scenarios in which they are able to produce accurate results.

1 INTRODUCTION

Blockchains, the technology underlying cryptocurrencies such as Bitcoin or Ethereum, herald an era in which mistrusting entities can cooperate in the absence of a trusted third party. However, current blockchain technology faces a scalability challenge, supporting merely tens of transactions per second, compared to custodian payment systems which eas-

in which the source of a payment specifies the complete route for the payment. If the global view of all nodes is accurate, source routing is highly effective because it finds all paths between pairs of nodes. Naturally, nodes are likely to prefer paths with lower per-hop fees, and are only interested paths which support their transaction, i.e., have sufficient channel capacity.

However, the fact that nodes need to be able to find routes also requires mechanisms for nodes to

Conclusion

- Can we trust our networks today? Challenges, due to complexity, **security assumptions** and lack of tools
- Opportunities of emerging network technologies
  - Programmability and virtualization: improved **network monitoring** and new tools, ***faster innovation***
  - „Self-driving networks“ and automation: case for **formal methods** and **AI**?
- Challenges of emerging network technologies
  - New threat models: e.g., ***jump*** firewall, ***propagate*** worm in datacenter
  - Algorithmic complexity attacks: e.g., make virtual switch ***crawl***
  - AI-driven attacks and performance fuzzing
- A new frontier: cryptocurrency networks
  - ***Attract*** transactions in Lightning

Further Reading

[Toward Active and Passive Confidentiality Attacks On Cryptocurrency Off-Chain Networks]
Utz Nisslmueller, Klaus-Tycho Foerster, Stefan Schmid, and Christian Decker.
6th International Conference on Information Systems Security and Privacy (**ICISSP**), Valletta, Malta, February 2020.

[NetBOA: Self-Driving Network Benchmarking]
Johannes Zerwas, Patrick Kalmbach, Laurenz Henkel, Gabor Retvari, Wolfgang Kellerer, Andreas Blenk, and Stefan Schmid.
ACM SIGCOMM Workshop on Network Meets AI & ML (**NetAI**), Beijing, China, August 2019.

[MTS: Bringing Multi-Tenancy to Virtual Switches]
Kashyap Thimmaraju, Saad Hermak, Gabor Retvari, and Stefan Schmid.
USENIX Annual Technical Conference (**ATC**), Renton, Washington, USA, July 2019.

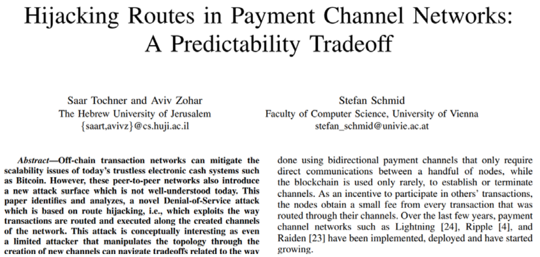
[Taking Control of SDN-based Cloud Systems via the Data Plane] (Best Paper Award)
Kashyap Thimmaraju, Bhargava Shastry, Tobias Fiebig, Felicitas Hetzelt, Jean-Pierre Seifert, Anja Feldmann, and Stefan Schmid.
ACM Symposium on SDN Research (**SOSR**), Los Angeles, California, USA, March 2018.

[Outsmarting Network Security with SDN Teleportation]
Kashyap Thimmaraju, Liron Schiff, and Stefan Schmid.
2nd IEEE European Symposium on Security and Privacy (**EuroS&P**), Paris, France, April 2017.

[Preacher: Network Policy Checker for Adversarial Environments]
Kashyap Thimmaraju, Liron Schiff, and Stefan Schmid.
38th International Symposium on Reliable Distributed Systems (**SRDS**), Lyon, France, October 2019.

[P-Rex: Fast Verification of MPLS Networks with Multiple Link Failures]
Jesper Stenbjerg Jensen, Troels Beck Krogh, Jonas Sand Madsen, Stefan Schmid, Jiri Srba, and Marc Tom Thorgersen.
14th International Conference on emerging Networking EXperiments and Technologies (**CoNEXT**), Heraklion, Greece, December 2018.

And



Hijacking Routes in Payment Channel Networks: A Predictability Tradeoff

Saar Tochner and Aviv Zohar
The Hebrew University of Jerusalem
{saart,avivz}@cs.huji.ac.il

Stefan Schmid
Faculty of Computer Science, University of Vienna
stefan_schmid@univie.ac.at

Abstract—**Off-chain transaction networks can mitigate the scalability issues of today's trustless electronic cash systems such as Bitcoin. However, these peer-to-peer networks also introduce a new attack surface which is not well-understood today. This paper identifies and analyzes, a novel Denial-of-Service attack which is based on route hijacking, i.e., which exploits the way transactions are routed and executed along the created channels of the network. This attack is conceptually interesting as even a limited attacker that manipulates the topology through the creation of new channels can navigate tradeoffs related to the way**

done using bidirectional payment channels that only require direct communications between a handful of nodes, while the blockchain is used only rarely, to establish or terminate channels. As an incentive to participate in others' transactions, the nodes obtain a small fee from every transaction that was routed through their channels. Over the last few years, payment channel networks such as Lightning [24], Ripple [4], and Raiden [23] have been implemented, deployed and have started growing.