Dependable and Secure Networks: Trends and Challenges

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

@ CERT.at Stammtisch
Communication Technologies @ Uni Wien

We aim at the investigation of future communication networks and future applications offered through these networks:

- **Algorithms** and mechanisms to design and operate communication networks
- Network architectures and protocols for future communication technologies
- **Performance** evaluation of networked and distributed systems
- Network security
- **Wireless** and cellular networks

Our vision is that networked systems should become *self-* (i.e., self-optimizing, self-repairing, self-configuring).

Accordingly, we are currently particularly interested in automated and data-driven approaches to design, optimize, and verify networked systems.
We aim at the investigation of future communication networks and future applications offered through these networks:

- **Algorithms** and mechanisms to design and operate communication networks
- Network **architectures** and **protocols** for future communication technologies
- **Performance** evaluation of networked and distributed systems
- Network **security**
- **Wireless** and cellular networks

Our vision is that networked systems should become **self-** (i.e., self-optimizing, self-repairing, self-configuring).

Accordingly, we are currently particularly interested in **automated** and **data-driven** approaches to design, optimize, and verify networked systems.
We aim at the investigation of future communication networks and future applications offered through these networks:

- **Algorithms** and mechanisms to design and operate communication networks
- **Network architectures** and **protocols** for future communication technologies
- **Performance** evaluation of networked and distributed systems
- **Network security**
- **Wireless** and cellular networks

Our vision is that networked systems should become **self-*** (i.e., self-optimizing, self-repairing, self-configuring).

Accordingly, we are currently particularly interested in **automated** and **data-driven** approaches to design, optimize, and verify networked systems.

But why?? Networks are working well today! Internet is huge success: hardly any outages!
Rewinding the clock of the Internet to a decade ago...

Slide credit: Pedro Casas
The Internet 50 Years Ago

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

Kudos to: Pedro Casas
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

Internet today: millions of users and billions of “things”, e.g., babyphones, webcams, cars (>6GB/h).

© Ivona Brandic
The Internet Is A Huge 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.
- “One of the complex artefacts created by mankind” (Christos H. Papadimitriou)

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.”
Indeed: More and More Exploits in the News

Vulnerabilities in **VPNs**

Iranian hackers have been hacking VPN servers to plant backdoors in companies around the world. Iranian hackers have targeted Pulsar Secure, Fortinet, Palo Alto Networks, and Citrix VPNs to hack into large companies.

Vulnerabilities in **IoT**


DDoS attacks often in the news (e.g. “babyphone attack”, **Olympics**)

How a Massive 540 Gb/sec DDoS Attack Failed to Spoil the Rio Olympics
How much can we trust technology?

(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

- 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
How much can we trust tech companies?

February 2020: For more than half a century, governments all over the world trusted a single company to keep the communications of their spies, soldiers and diplomats secret. But: Crypto AG was secretly owned by the CIA.
Awareness is Rising: First Creative Efforts for Self-Protection

February 2020: Wearable microphone jamming.
(https://www.mirror.co.uk/tech/alexa-owners-can-stop-eavesdropping-21539032)
Another Example: Wearable Camera Jamming

Glasses developed by Scott Urban reflect infrared light from security cameras to blur out the wearer’s face.
Another Major Issue: Complexity

Many outages due to misconfigurations and human errors.

Entire countries disconnected...

Google routing blunder sent Japan's Internet dark on Friday

Another big BGP blunder

By Richard Chirgwin 27 Aug 2017 at 22:35

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 11:35

... 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 15, 2018

Duluth News Tribune

SKYNT 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
And: *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 20y ago* to merely test end-to-end connectivity. Neither tool could reveal *problems with 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 **networking team would be blamed** since they were unable to demonstrate otherwise.”

Source: «The world’s fastest and most programmable networks»
White Paper Barefoot Networks
Complexity and human errors: we **need technology** and the networks should be **programmable**. However, this technology needs to be highly **dependable**.

PS: We *cannot stop* technology. And with IoT we already lost anyway. 😊
A 2nd Takeaway

Our digital society relies on *all sorts of networks*, e.g., increasingly on the networks to, from, and in *datacenters*, but also more “exotic” networks such as *in-cabin* and car *networks*, *cryptocurrency* networks, etc.
Roadmap

• Opportunity: emerging networking technologies
  – Programmability and virtualization
  – „Self-driving networks“ and automation
  – Case study P-Rex: Automated what-if analysis of MPLS networks

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

• Another uncharted security landscape: cryptocurrency networks
Roadmap

• Opportunity: emerging networking technologies
  – Programmability and virtualization
  – „Self-driving networks“ and automation
  – Case study P-Rex: Automated what-if analysis of MPLS networks

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

• Another uncharted security landscape: cryptocurrency networks
Roadmap

- **Opportunity: emerging networking technologies**
  - Programmability and virtualization
  - “Self-driving networks” and automation
  - Case study P-Rex: Automated what-if analysis of MPLS networks

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

- Another uncharted security landscape: cryptocurrency networks
Case Study: Datacenter Network Virtualization

Benefit of virtualization: resource sharing ("economy of scale")

VMs allocated dynamically, multiplexing
Case Study: Datacenter Network Virtualization

Benefit of virtualization: resource sharing ("economy of scale")

Different tenants: requires isolation!
Case Study: Datacenter Network Virtualization

- Internet
- LAN
- Broadcast domain: need isolation!
- Different tenants: requires isolation!
- Router
- Switches
- Racks of Servers
Security Requires *Isolation on All Levels*
State-of-the-Art Datacenter Networks
Network Virtualization Today: Tunneling

State-of-the-art: overlays, **tunneling** (e.g., VxLAN, VLAN, MPLS, …)
At the heart: Virtual Switches Networking the VMs

Connect and isolate!
However: Today Network Virtualization is Complex and Inflexible

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

Example: BGP in Datacenter

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

Cluster with services that should be globally reachable.

Cluster with services that should be accessible only internally.

Credits: Beckett et al. (SIGCOMM 2016): Bridging Network-wide Objectives and Device-level Configurations.
X and Y *announce* to Internet what is from $G^*$ (prefix).

X and Y *block* what is from $P^*$.

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

Example: BGP in Datacenter

X and Y announce to Internet what is from G* (prefix).

X and Y block what is from P*.

What can go wrong?

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

Example: BGP in Datacenter

X and Y announce to Internet what is from G* (prefix).

X and Y block what is from P*.

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.
Besides Complexity, Innovation is Slow: Example VxLAN

VxLAN: In principle, addition of a *simple function* to be added to switches and routers

- Defined 2010 by Cisco and VMware
Besides Complexity, Innovation is Slow: Example VxLAN

VxLAN: In principle, addition of a *simple function* to be added to switches and routers
- Defined 2010 by Cisco and VMware

At heart: devices running an OS (e.g. based on Linux or UNIX)
Below: driver communicating to add and delete entries into a forwarding chip
On top: user space processes implementing control

© Nick McKeown
Besides Complexity, Innovation is Slow: Example VxLAN

Needed steps to add VxLAN:

- **Add control** of VxLAN protocol
- **Change driver** to add/remove entries into VxLAN table in switch ASIC
- **Update ASIC**

© Nick McKeown
Besides Complexity, Innovation is Slow: Example VxLAN

Needed steps to add VxLAN:
- *Add control* of VxLAN protocol  
  Doable in weeks!
- *Change driver* to add/remove entries into VxLAN table in switch ASIC  
  Doable in weeks!
- *Update ASIC*  
  Took 4 years to add feature to ASIC!

At heart: devices running an OS (e.g. based on Linux or UNIX)

Below: driver communicating to add and delete entries into a forwarding chip

On top: user space processes implementing control
Why Does It Take So Long?

Network Owner

Network Equipment Vendor

Years

Need feature!

Vendors get together at IETF: which feature exactly?

Weeks

Features

Years

Software Team

ASIC Team

We can do that!
Why Does It Take So Long?

Vendors get together at IETF: which feature exactly?

In the meantime, owners probably figured out a workaround making network more complex and brittle.
Besides Slow Innovation: Process is Inflexible and Expensive

Operator says:

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

Vendor's answer:

Buy one of these!
I need something better than STP for my data-center...

We don't have that!
Programmable Networks

OSPF  BGP  

Switch OS

Driver

This is how I process packets!

Fixed-function switch

Traditionally: features defined by chip designers, defines what can be done.
Programmable Networks

Traditionally: features defined by chip designers, defines what can be done.

This is how I process packets!

Future? Features defined by operator, tells switch what we really want!

This is how I want to process packets!
Networking is Catching Up: Happening in Other Domains

Domain specific processors are a trend:

- **Computers**
  - CPU
  - Java Compiler

- **Graphics**
  - GPU
  - OpenCL Compiler

- **Signal Processing**
  - DSP
  - Java Compiler

- **Machine Learning**
  - TPU
  - TensorFlow Compiler

- **Networking**
  - PISA/Tofino
  - P4 Compiler
What About Performance?

• Are programmable switches not much *slower* than fixed-function switches?
  – And *cost* more and consume more *power*?
• As data models, ASIC technology etc. are evolving: no!
• Tofino chip: operates at 6.5 Tb/s (fastest in world!)
  – Can switch entire Netflix catalogue in **20sec**
  – While running a **4000 line program** on any packet...
  – ... and not being more costly or consume more power
What About Performance?

• Are programmable switches not much *slower* than fixed-function switches?
  – And *cost* more and consume more *power*?
• As data models, ASIC technology etc. are evolving: no!
• Tofino chip: operates at *6.5 Tb/s* (fastest in world!)
  – Can switch entire Netflix catalogue in *20sec*
  – While running a *4000 line program* on any packet...
  – ... and not being more costly or consume more power
A 3rd Takeaway

Programmable networks can enable faster innovation without decreasing performance or increasing cost.
A 4th Takeaway

Not only the **data plane** becomes *programmable* but also the **control plane**.
A 4th Takeaway

Not only the **data plane** becomes **programmable** but also the **control plane**.

*Local* functions, e.g., *forward* packet from incoming interface to outgoing interface.

*Network-wide* functions such as *routing*!
A 4th Takeaway

Not only the data plane becomes programmable but also the control plane.

Local functions, e.g., forward packet from incoming interface to outgoing interface.

Network-wide functions such as routing!

Analogy: minister of education (Heinz Fassmann)

Analogy: teacher in classroom
Traditionally:
• Distributed control plane
• Blackbox, not programmable

Software-defined Networks (SDN):
• Logically centralized control
• Programmable, match-action
Traditionally:
• Distributed control plane
• Blackbox, not programmable

Software-defined Networks (SDN):
• Logically centralized control
• Programmable, match-action

**Benefit 1:** simple and fast innovation in control plane!
Centralized view, and can implement own control plane algorithms (e.g., routing).

**Benefit 2:** secure communication channels.

**Benefit 3:** more flexible API (match-action).
Example Application for SDN: Detecting Misbehavior
Allows to Deal with New Threat Vectors: Secure Trajectory Sampling

Monitor packets, traditionally:

*trajectory sampling*

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

*mirror, exfiltrate, modify, drop, insert, ... and misreport: knows what is currently sampled!**
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!

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!*

---

A 5th Takeaway

Programmable control planes (SDN) enable fast innovation in the control plane and can help improve network security.
Another and Related Trend Motivated by Network Complexity: Automation
Responsibilities of a Sysadmin

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

A

B

C

• Reachability: Can traffic from ingress port A reach egress port B?
• Loop-freedom: Are the routes implied by the forwarding rules loop-free?
• Non-reachability: Is it ensured that traffic originating from A never reaches B?
• Waypoint ensurance: Is it ensured that traffic from A to B is always routed via a node C (e.g., a firewall)?
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

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?

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

Router configurations, Segment Routing etc.

Compilation

\[ pX \Rightarrow qXX \]
\[ pX \Rightarrow qYX \]
\[ qY \Rightarrow rYY \]
\[ rY \Rightarrow r \]
\[ rX \Rightarrow pX \]

Interpretation

Pushdown Automaton and Prefix Rewriting Systems Theory
Vision: Automation Methods

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

Compilation

pX ⇒ qXX
pX ⇒ qYX
qY ⇒ rYY
rY ⇒ r
rX ⇒ pX

Interpretation

What if...?!

Router configurations, Segment Routing etc.

Pushdown Automaton and Prefix Rewriting Systems Theory
Vision: Automation and Formal Methods

Router configurations, Segment Routing etc.

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

Compilation

\[ p_X \Rightarrow q_{XX} \]
\[ p_X \Rightarrow q_{YX} \]
\[ q_Y \Rightarrow r_{YY} \]
\[ r_Y \Rightarrow r \]
\[ r_X \Rightarrow p_X \]

Interpretation

Compilation

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!)
Demo of P-Rex / AalWiNes Tool

Tool: https://demo.aalwines.cs.aau.dk/, Youtube: https://www.youtube.com/watch?v=mvXAn9i7_Q0
Roadmap

• Opportunity: emerging networking technologies
  – Programmability and virtualization
  – „Self-driving networks“ and automation
  – Case study P-Rex: Automated what-if analysis of MPLS networks

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

• Another uncharted security landscape: cryptocurrency networks
Example 1: Data Plane
New Types of Attacks: Security of Compiler?

- Bugs in compiler not easy to catch
  - New attack surface?

- P4Fuzz: compiler fuzzer

- Further reading:

Example 2: Control Plane
New Types of Attacks: Via SDN Controller

- **Controller** may be attacked or exploited
New Types of Attacks: Via SDN Controller

- **Controller** may be attacked or exploited
  - By design, *reacts* to switch events, e.g., by packet-outs
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
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 unintended ways!
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 Stefan Schmid.
EuroS&P, Paris, France, April 2017 + [CVEs](#).
Example 3: Virtual Switch
Trend in Datacenter Networks: Virtual Switches

**Simple in core:** hardware switches.

**Clever at edge:** virtualized and Programmable.
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...
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
Further Reading

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,
ACM Symposium on SDN Research (SOSR), Los Angeles, California, USA, March 2018.
Challenge: How to provide better isolation efficiently?

- Idea for better *isolation*: put vSwitch in a VM
- But what about *performance*?
- Or container?
Example 4: 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.
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?!
Example 5: 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.
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

- **Opportunity: emerging networking technologies**
  - Programmability and virtualization
  - „Self-driving networks“ and automation
  - Case study P-Rex: Automated what-if analysis of MPLS networks

- **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 \textbf{scalability of Bitcoin} and other cryptocurrencies
- E.g., Lightning, Raven, Ripple, ...
- But also \textit{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
Toward Active and Passive Confidentiality Attacks
On Cryptocurrency Off-Chain Networks

Utz Nisslmeier1, Klaus-Tycho Foerster1, Stefan Schmid1, and Christian Decker2
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 easily 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

Thank you
Further Reading

Toward Active and Passive Confidentiality Attacks On Cryptocurrency Off-Chain Networks
Utz Nisslmueller, Klaus-Tycho Foerster, Stefan Schmid, and Christian Decker.

NetBOA: Self-Driving Network Benchmarking
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.

Taking Control of SDN-based Cloud Systems via the Data Plane (Best Paper Award)
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.

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
14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Heraklion, Greece, December 2018.

And

Hijacking Routes in Payment Channel Networks: A Predictability Tradeoff
Samuel Jahan and Anirudh Jha
The Hebrew University of Jerusalem
samuel.jahan@huji.ac.il
Stefan Schmid
Faculty of Computer Science, University of Vienna
stefan.schmid@tuwien.ac.at

Advances in on-chain transactions networks can mitigate the availability issues of today’s lightning electronic cash systems such as Bitcoin and Ethereum. In this paper, we describe a new attack surface which is not well-understood today. This paper describes and analyses a novel financial-hijacking attack which is based on chain hijacking, i.e., which exploits the way transactions are spent and verified along the channel. The attack leverages the open-transaction view which is the result of a limited attacker that manipulates the topology through the creation of new channels can manipulate channels related to the new

Direction—Off-chain transactions networks can mitigate the availability issues of today’s lightning electronic cash systems such as Bitcoin and Ethereum. In this paper, we describe a new attack surface which is not well-understood today. This paper describes and analyses a novel financial-hijacking attack which is based on chain hijacking, i.e., which exploits the way transactions are spent and verified along the channel. The attack leverages the open-transaction view which is the result of a limited attacker that manipulates the topology through the creation of new channels can manipulate channels related to the new