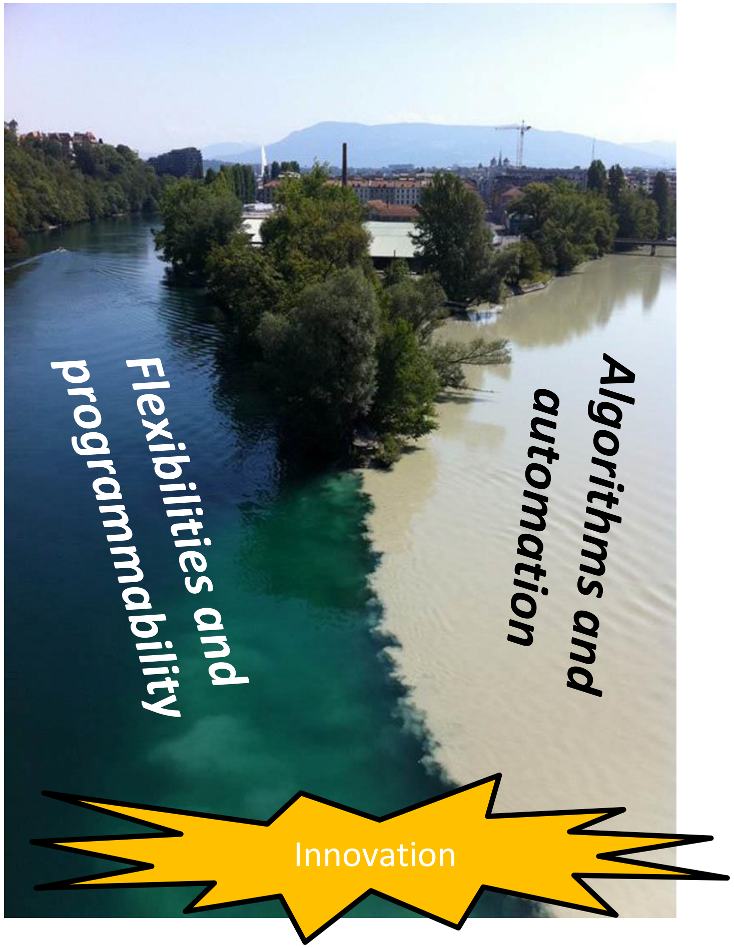
Self-Adjusting Networks: Vision, Solutions, Challenges

Stefan Schmid

"We cannot direct the wind,
but we can adjust the sails."
(Folklore)

Acknowledgements:

It`s a Great Time to Be a Networking Researcher!



Credits: George Varghese

It`s a Great Time to Be a Networking Researcher!

Enables and motivates **self-adjusting networks**!

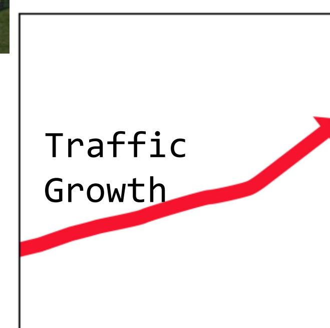
Credits: George Varghese

It’s High Time!

Explosive Traffic

Datacenters (“hyper-scale”)

Interconnecting networks:
a **critical infrastructure**
of our digital society.



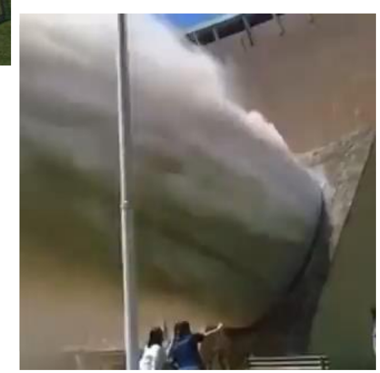
It’s High Time!

Explosive Traffic

Datacenters (“hyper-scale”)

+network

Interconnecting networks:
a **critical infrastructure**
of our digital society.



Credits: Marco Chiesa

It’s High Time!

Reality vs Requirements

Today, dependability requirements stand in contrast with reality:

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.

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 ▼

BA flights around the world were grounded as a result of the Amadeus outage

Even 911 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:

Go Daddy.com github SOCIAL CODING amazon web services

Credits: Laurent Vanbever, Nate Foster

It’s High Time!

Reality vs Requirements

Today, dependability requirements stand in contrast with reality:

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.

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 ▼

BA flights around the world were grounded as a result of the Amadeus outage

Even 911 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:

GoDaddy.com github SOCIAL CODING amazon web services

Mainly: human errors!

Credits: Laurent Vanbever, Nate Foster

Agenda

Three Use Cases

Passau, Germany

Agenda

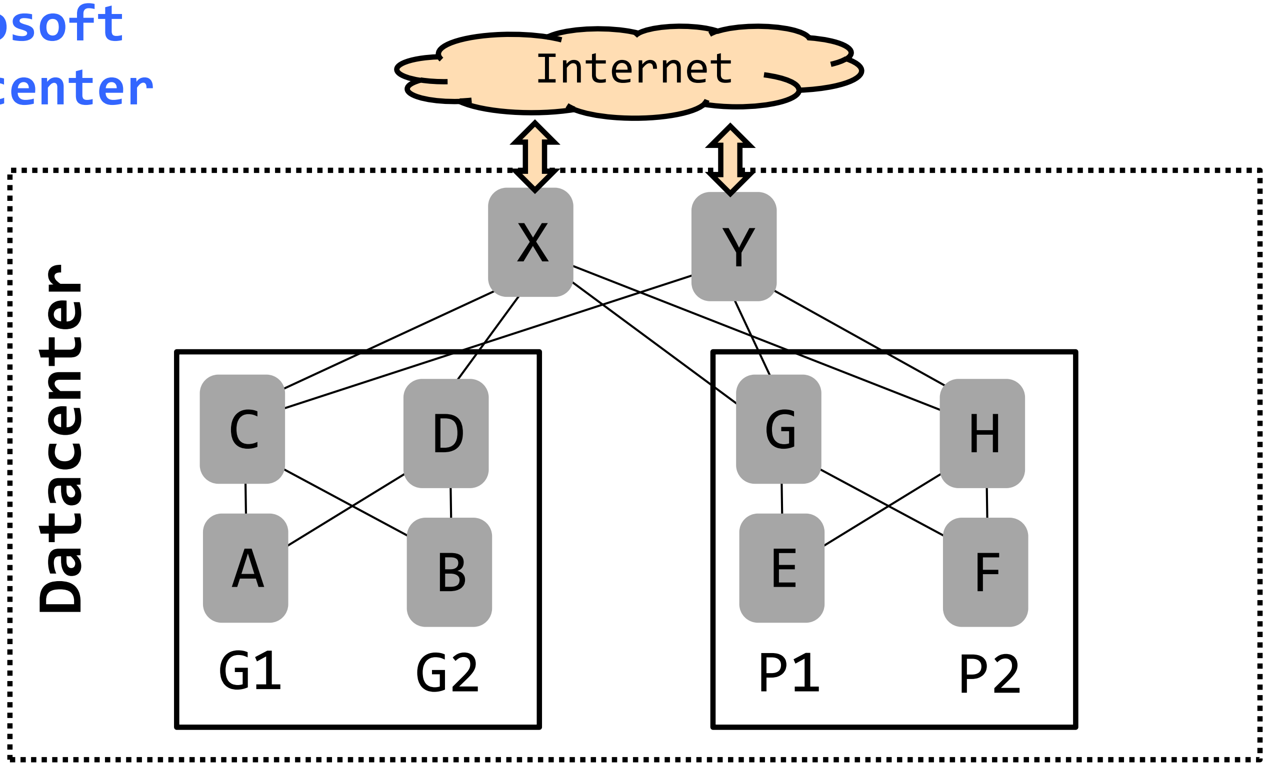
Three Use Cases

Formal methods as a tool!

Motivation: Complexity

Especially Under Failures (Policy Compliance)

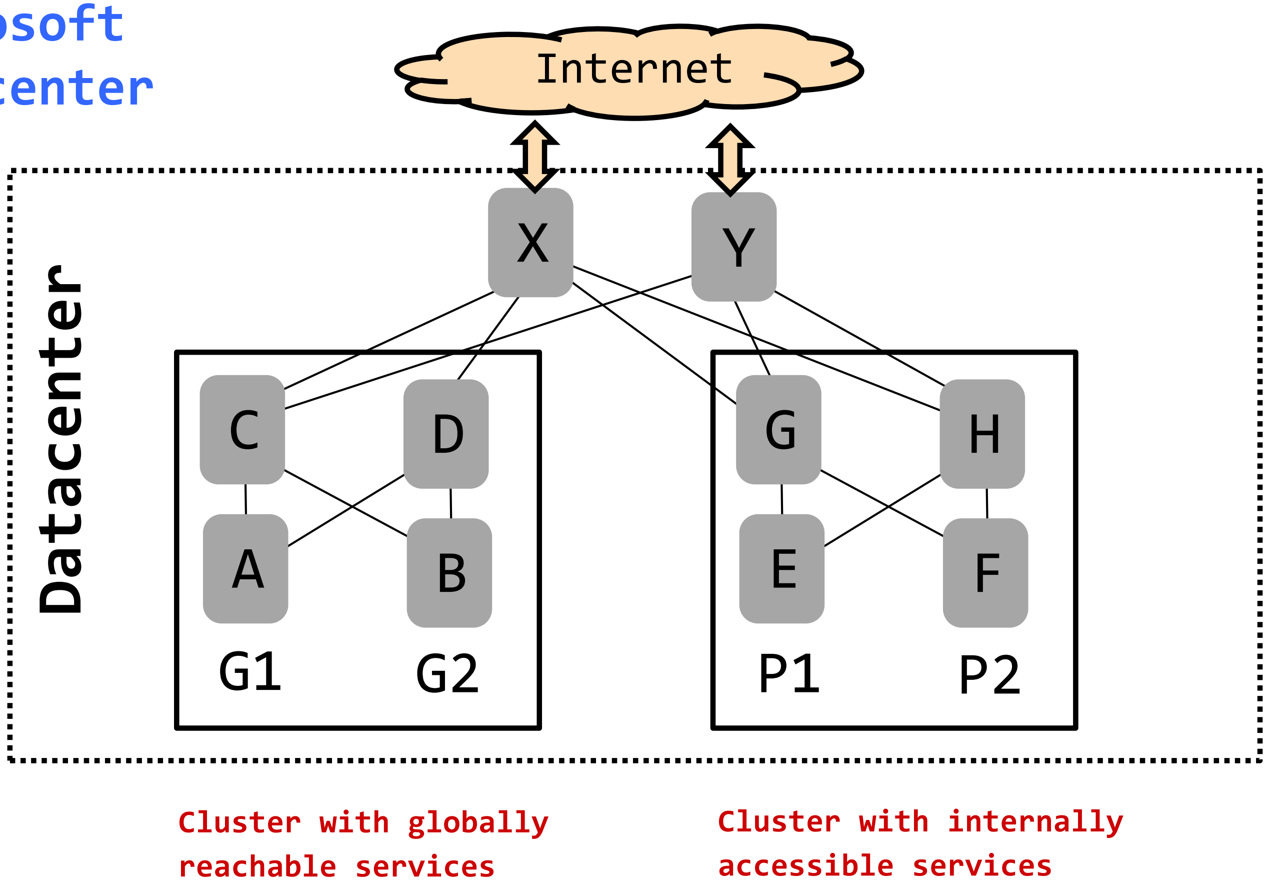
Example: BGP in **Microsoft datacenter**



Motivation: Complexity

Especially Under Failures (Policy Compliance)

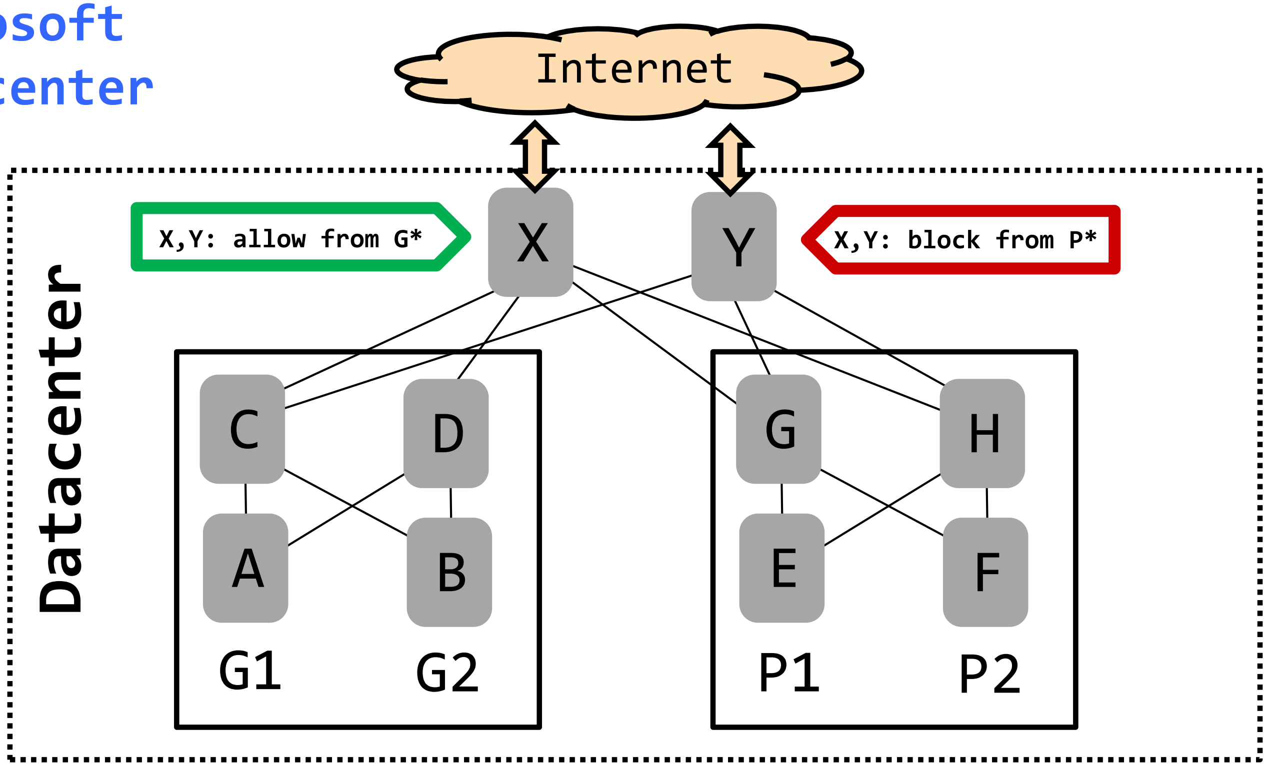
Example: BGP in **Microsoft datacenter**



Motivation: Complexity

Especially Under Failures (Policy Compliance)

Example: BGP in **Microsoft datacenter**



Cluster with globally reachable services

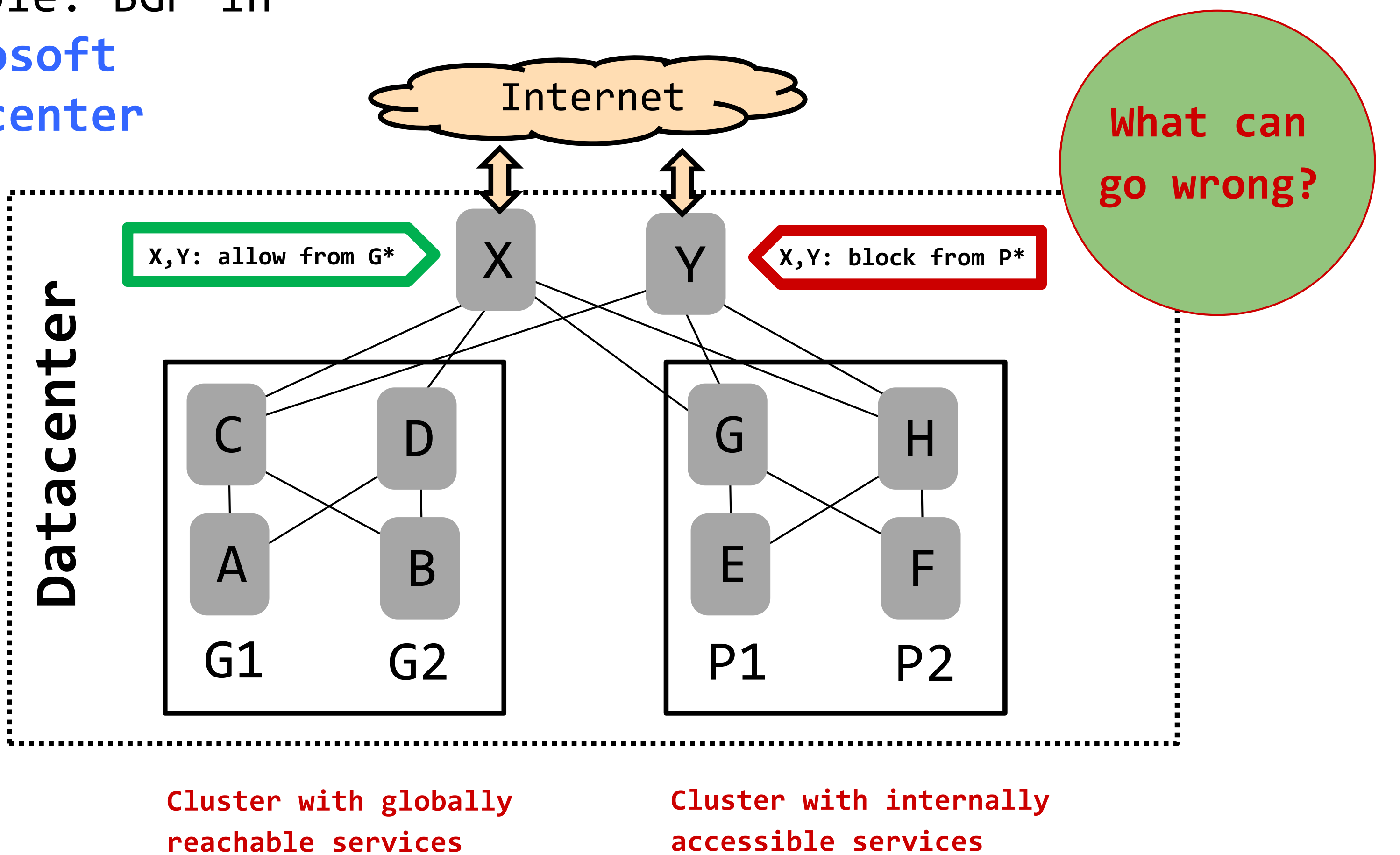
Cluster with internally accessible services

Credits: Ratul Mahajan

Motivation: Complexity

Especially Under Failures (Policy Compliance)

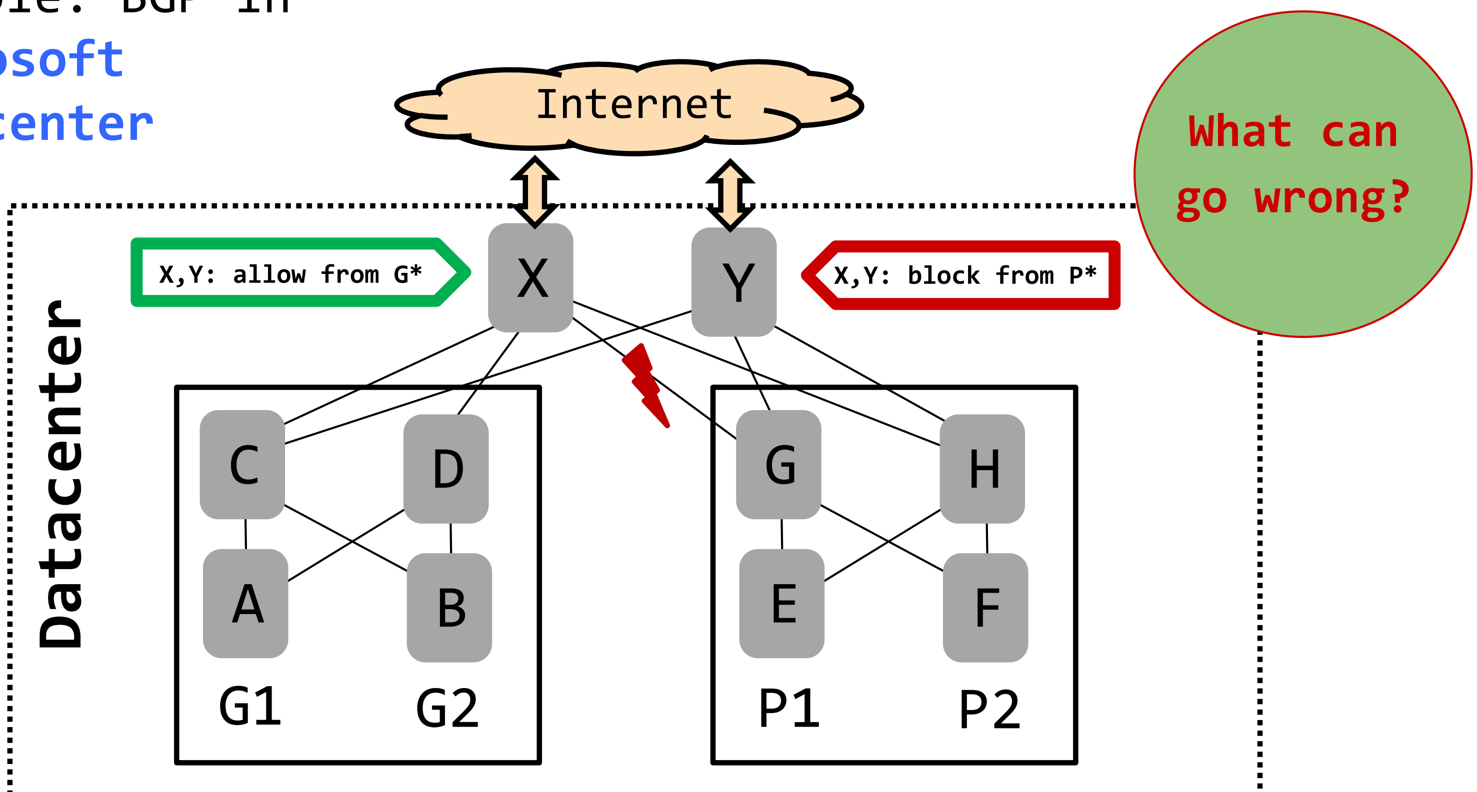
Example: BGP in **Microsoft datacenter**



Motivation: Complexity

Especially Under Failures (Policy Compliance)

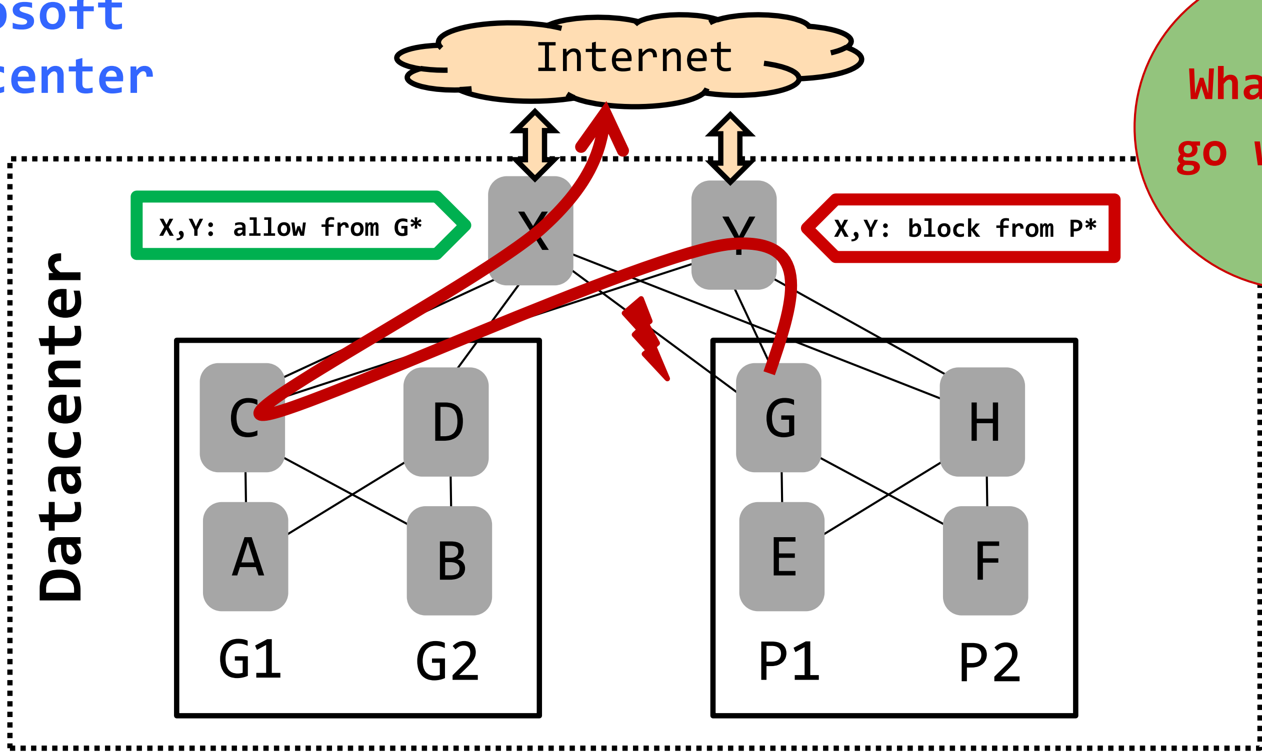
Example: BGP in **Microsoft datacenter**



Motivation: Complexity

Especially Under Failures (Policy Compliance)

Example: BGP in **Microsoft datacenter**

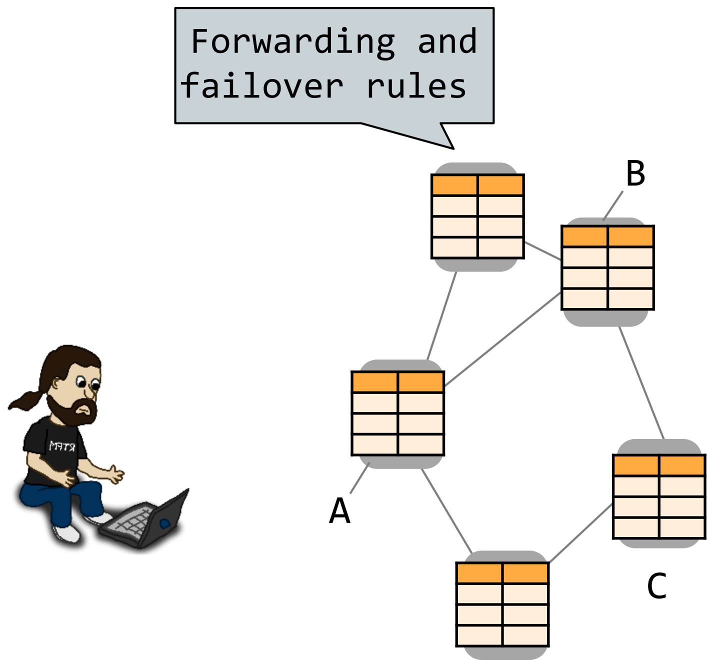


What can go wrong?

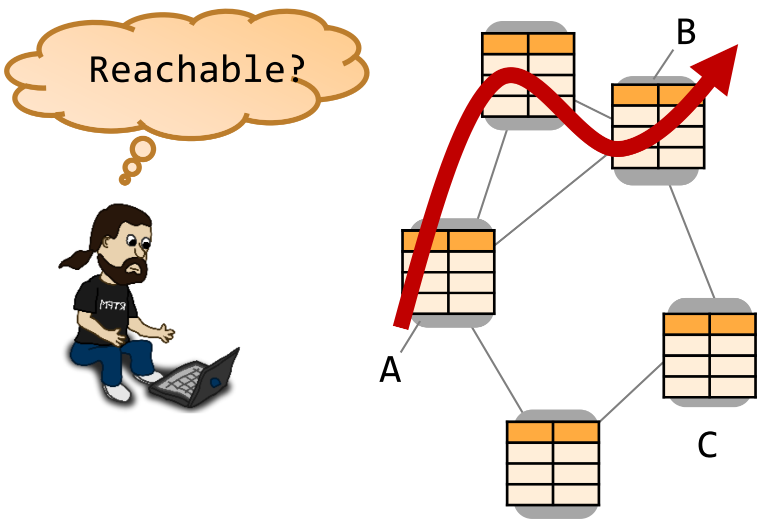
If link (G,X) fails and traffic from G is rerouted via Y and C to X: X announces (does not block) G and H as it comes from C. (Note: BGP.)

Credits: Ratul Mahajan

Admin’s Responsibilities

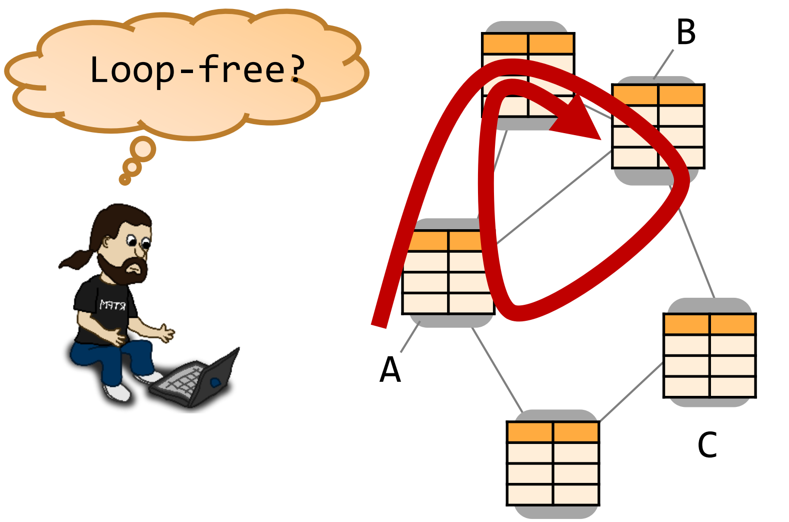


Admin's Responsibilities



⇢ Reachability: Can traffic from ingress port A reach B?

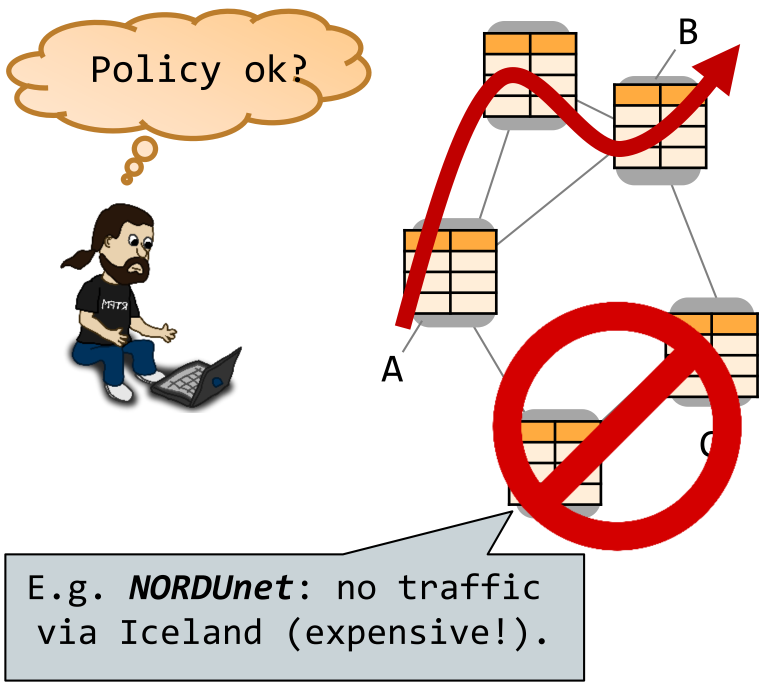
Admin's Responsibilities



⇢ Reachability: Can traffic from ingress port A reach B?

⇢ Loop-freedom: Do forwarding rules imply loop-free routes?

Admin's Responsibilities

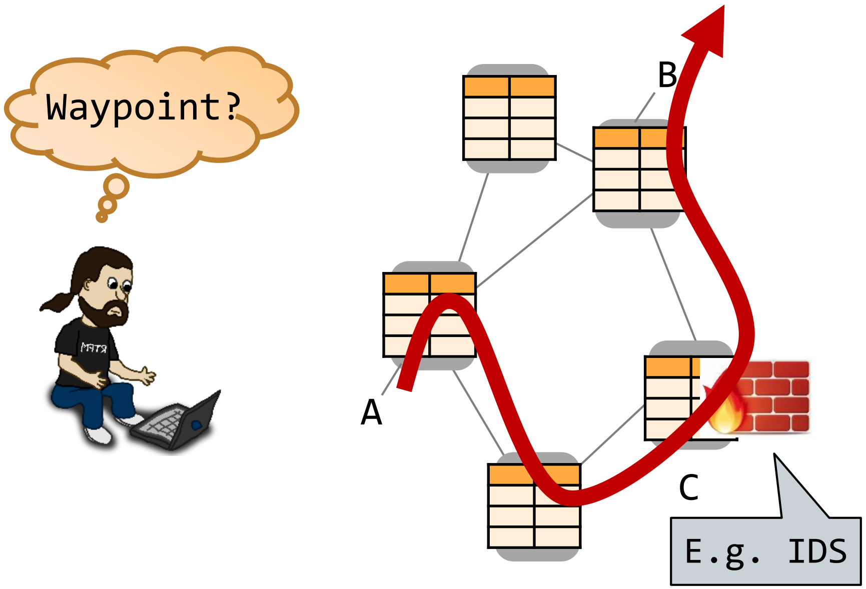


⇢ Reachability: Can traffic from ingress port A reach B?

⇢ Loop-freedom: Do forwarding rules imply loop-free routes?

⇢ Policy: Does traffic from A to B never go via C?

Admin's Responsibilities



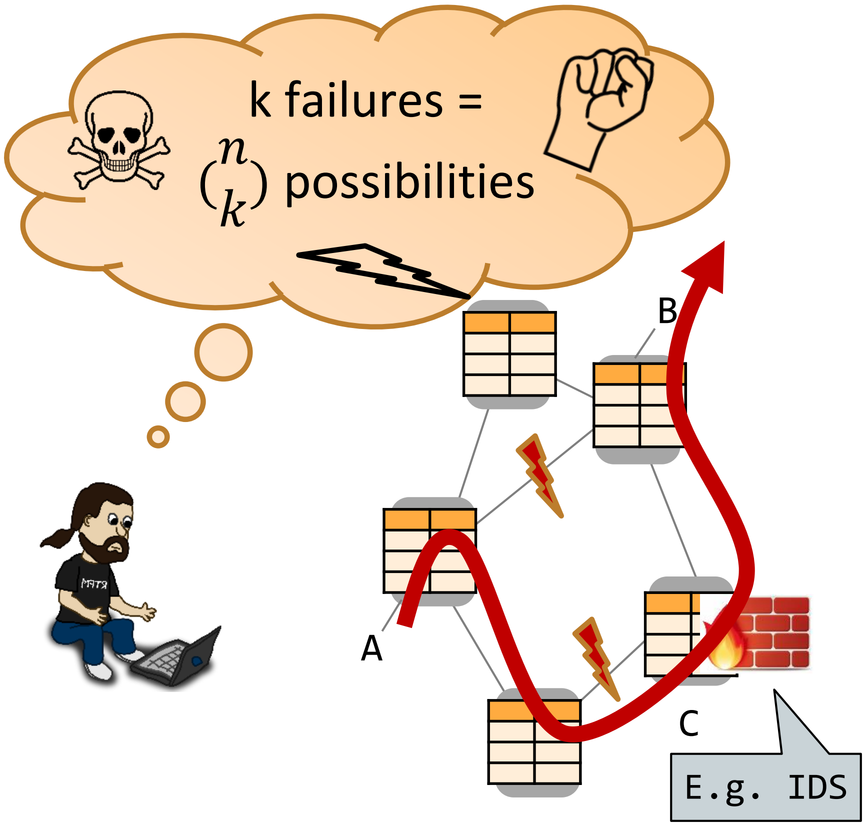
⇢ Reachability: Can traffic from ingress port A reach B?

⇢ Loop-freedom: Do forwarding rules imply loop-free routes?

⇢ Policy: Does traffic from A to B never go via C?

⇢ Waypoint enforcement: Is traffic from A to B always routed via a node C (e.g., an IDS)?

Admin's Responsibilities



⇢ Reachability: Can traffic from ingress port A reach B?

⇢ Loop-freedom: Do forwarding rules imply loop-free routes?

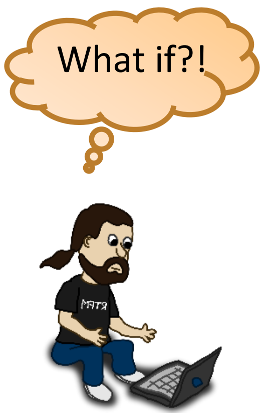
⇢ Policy: Does traffic from A to B never go via C?

⇢ Waypoint enforcement: Is traffic from A to B always routed via a node C (e.g., an IDS)?

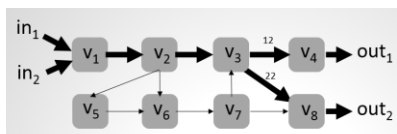
… and everything even under failures?!

A Modern Approach:

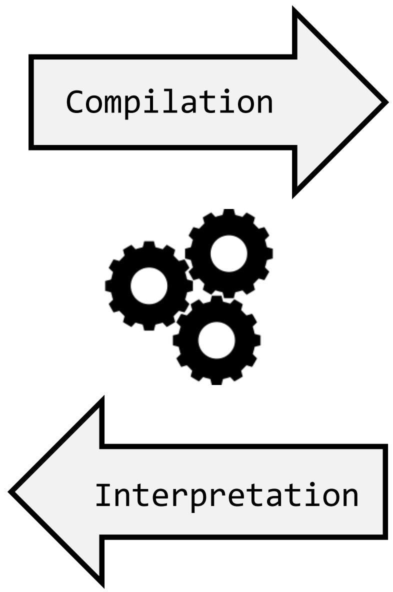
Automated Whatif Analysis



FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\perp$	(v_1, v_2)	$push(10)$
	in_2	$\perp$	(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)	$push(40)$



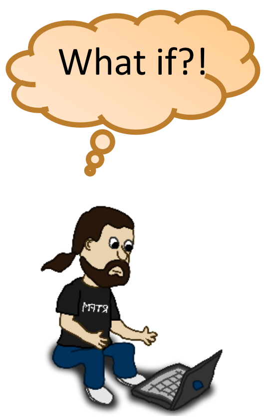
$$pX \Rightarrow qXX$$
$$pX \Rightarrow qYX$$
$$qY \Rightarrow rYY$$
$$rY \Rightarrow r$$
$$rX \Rightarrow pX$$

Router **configurations**
(Cisco, Juniper, etc.)

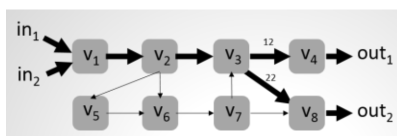
Formal language
which supports
automated analysis

A Modern Approach:

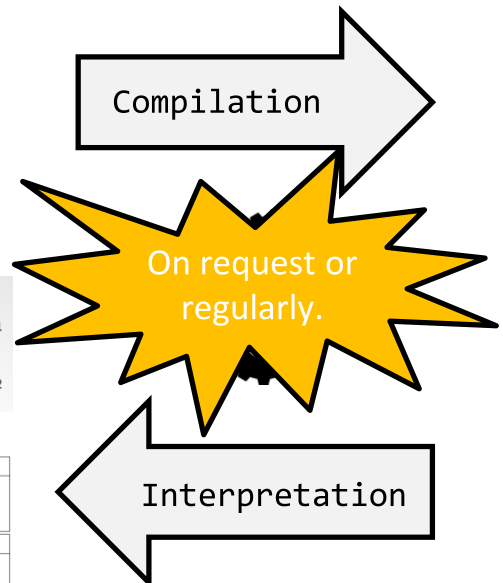
Automated Whatif Analysis



FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\perp$	(v_1, v_2)	$push(10)$
	in_2	$\perp$	(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)	$push(40)$



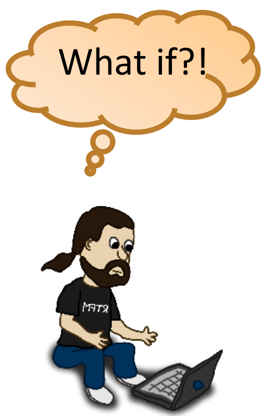
$$pX \Rightarrow qXX$$
$$pX \Rightarrow qYX$$
$$qY \Rightarrow rYY$$
$$rY \Rightarrow r$$
$$rX \Rightarrow pX$$

Router **configurations**
(Cisco, Juniper, etc.)

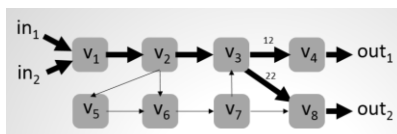
Formal language
which supports
automated analysis

A Modern Approach:

Automated Whatif Analysis



FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\perp$	(v_1, v_2)	$push(10)$
	in_2	$\perp$	(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**
(Cisco, Juniper, etc.)

Compilation

On request or regularly.

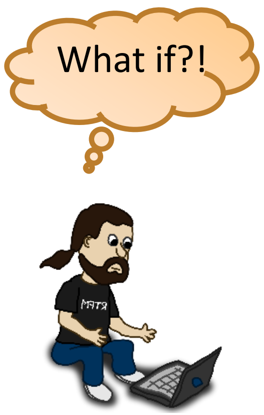
Fix/**synthesize**

$$
\begin{aligned}
pX &\Rightarrow qXX \\
pX &\Rightarrow qYX \\
qY &\Rightarrow rYY \\
rY &\Rightarrow r \\
rX &\Rightarrow pX
\end{aligned}
$$

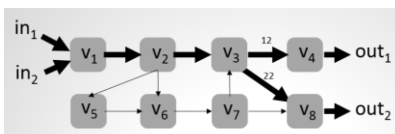
Formal language
which supports
automated analysis

A Modern Approach:

Automated Whatif Analysis

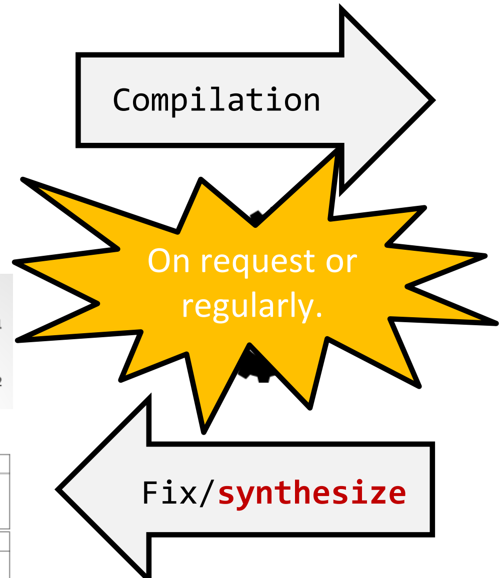


FT	In-I	In-Label	Out-I	op
τ_{v_1}	in_1	$\perp$	(v_1, v_2)	$push(10)$
	in_2	$\perp$	(v_1, v_2)	$push(20)$
τ_{v_2}	(v_1, v_2)	10	(v_2, v_3)	$swap(11)$
	(v_1, v_2)	20	(v_2, v_3)	$swap(21)$
τ_{v_3}	(v_2, v_3)	11	(v_3, v_4)	$swap(12)$
	(v_2, v_3)	21	(v_3, v_8)	$swap(22)$
	(v_7, v_3)	11	(v_3, v_4)	$swap(12)$
	(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)	[illegible]



$$
\begin{aligned}
pX &\Rightarrow qXX \\
pX &\Rightarrow qYX \\
qY &\Rightarrow rYY \\
rY &\Rightarrow r \\
rX &\Rightarrow pX
\end{aligned}
$$

Router **configura**
(Cisco, Juniper,

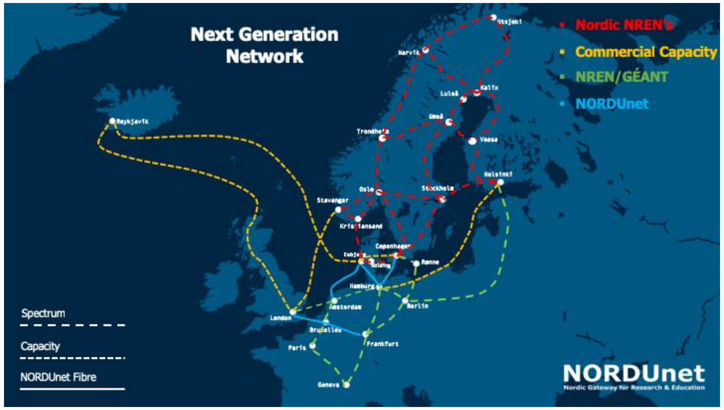
l language
supports
ed analysis

Challenge:

Hard Even for Computers?

⇢ NORDUnet: provider for Nordic countries

⇢ 24 MPLS routers, running Juniper OS, >30,000 labels!

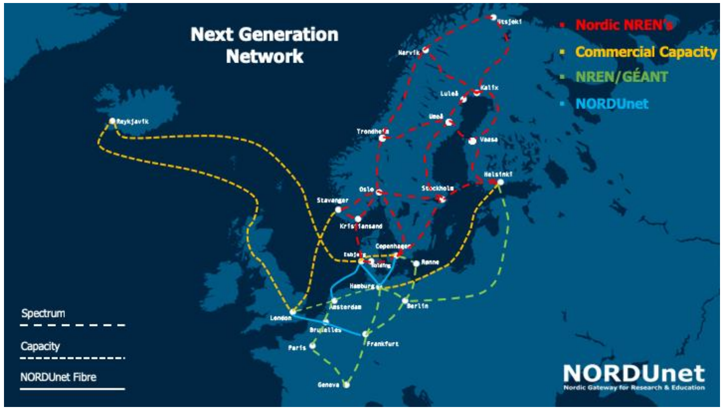


Challenge:

Hard Even for Computers?

⇢ NORDUnet: provider for Nordic countries

⇢ 24 MPLS routers, running Juniper OS, >30,000 labels!



For specific networks such as MPLS: feasible and fast!
Tools such as P-Rex or AalWiNes do it in secs for MPLS: reduction to automata theory, polynomial-time.

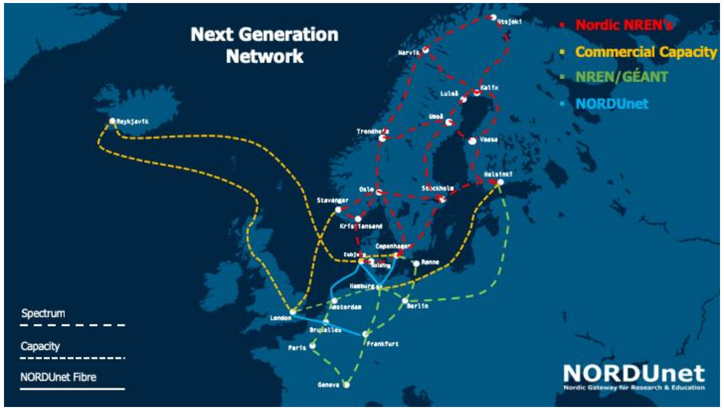
Formal methods!

Challenge:

Hard Even for Computers?

⇢ NORDUnet: provider for Nordic countries

⇢ 24 MPLS routers, running Juniper OS, >30,000 labels!



For specific networks such as MPLS: feasible and fast!
Tools such as P-Rex or AalWiNes do it in secs for MPLS: reduction to automata theory, polynomial-time.
But general networks more challenging.

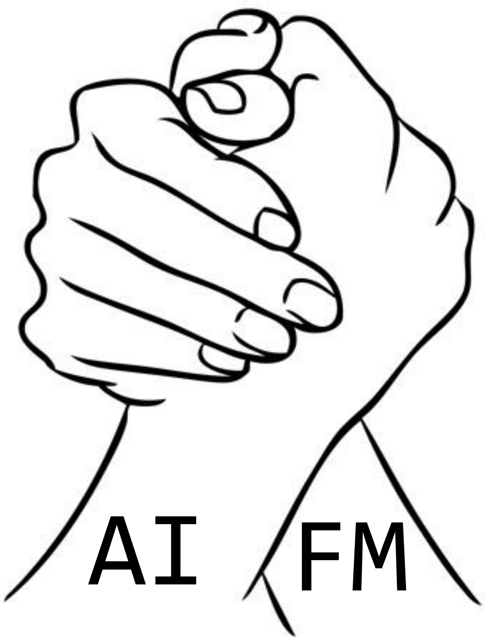
Formal methods!

Fixing&Synthesis: Harder

- Approaches: ***Petri games***, Stackelberg games, UPPAAL Stratego…
- But ***synthesis slower*** than verification

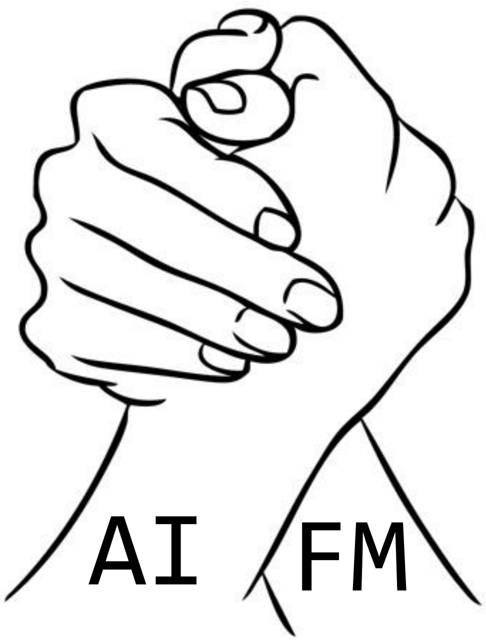
Fixing&Synthesis: Harder

- Approaches: ***Petri games***, Stackelberg games, UPPAAL Stratego…
- But ***synthesis slower*** than verification
- An opportunity for using AI!
- ***Ideally AI+FM***: guarantees from formal methods, performance from AI
- For example: synthesize with AI then verify with formal methods
- Examples: DeepMPLS, DeepBGP, …



Fixing&Synthesis: Harder

- Approaches: ***Petri games***, Stackelberg games, UPPAAL Stratego…
- But ***synthesis slower*** than verification
- An opportunity for using AI!
- ***Ideally AI+FM***: guarantees from formal methods, performance from AI
- For example: synthesize with AI then verify with formal methods
- Examples: DeepMPLS, DeepBGP, …



Fixing&Synthesis: Harder

- Approaches: ***Petri games***, Stackelberg games, UPPAAL Stratego…
- But ***synthesis slower*** than verification
- An opportunity for using AI!
- ***Ideally AI+FM***: guarantees from formal methods, performance from AI
- For example: synthesize with AI then verify with formal methods
- Examples: DeepMPLS, DeepBGP, …

… and what about quantitative properties?

A Possible Starting Point:
The AalWiNes Tool

Query: regular expression

Witness

Dozens of networks

Online demo: https://demo.aalwines.cs.aau.dk/
Source code: https://github.com/DEIS-Tools/AalWiNes
Paper: https://schmiste.github.io/conext20.pdf

Agenda

Three Use Cases

Passau, Germany

Agenda

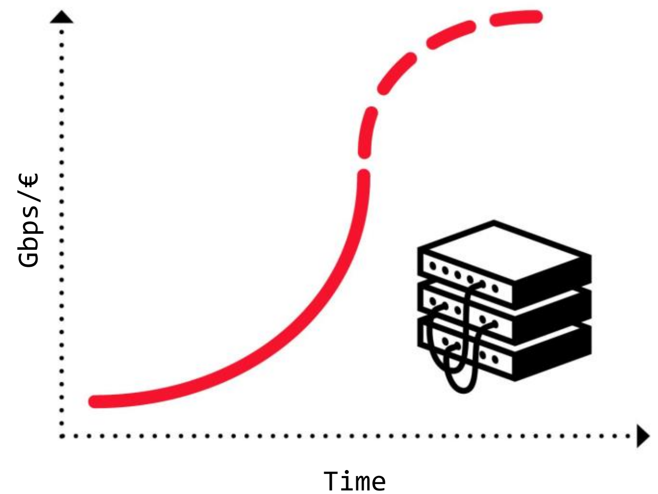
Three Use Cases

On lower layers!

Let’s go back to datacenter use case:

Moore’s Law of Datacenters

- Recall: **explosive growth** of demand
- Problem: network equipment reaching capacity limits
  - → Transistor density rates stalling
  - → “End of **Moore‘s Law** in networking”
- Hence: more equipment, larger networks
- Resource intensive and: **inefficient**



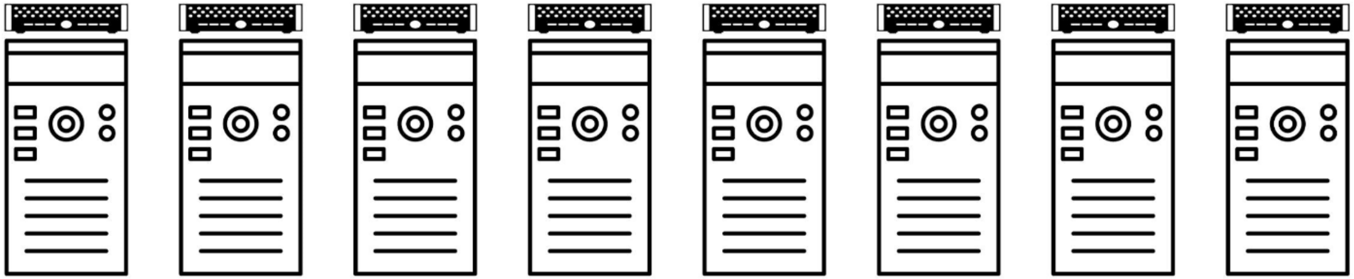
[1] Source: Microsoft, 2019

Annoying for companies, **opportunity** for researchers

Root Cause

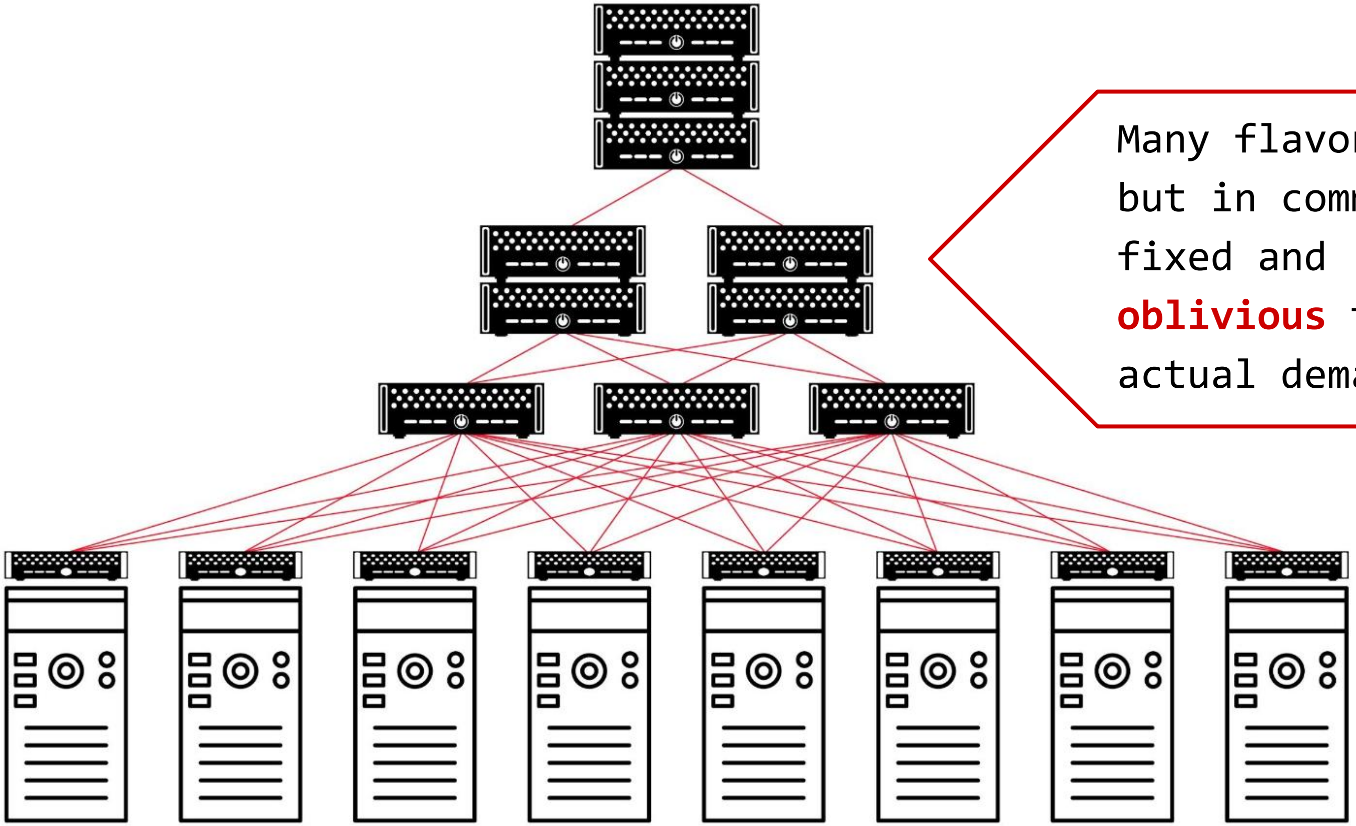
Fixed and Demand-Oblivious Topology

How to interconnect?



Root Cause

Fixed and Demand-Oblivious Topology

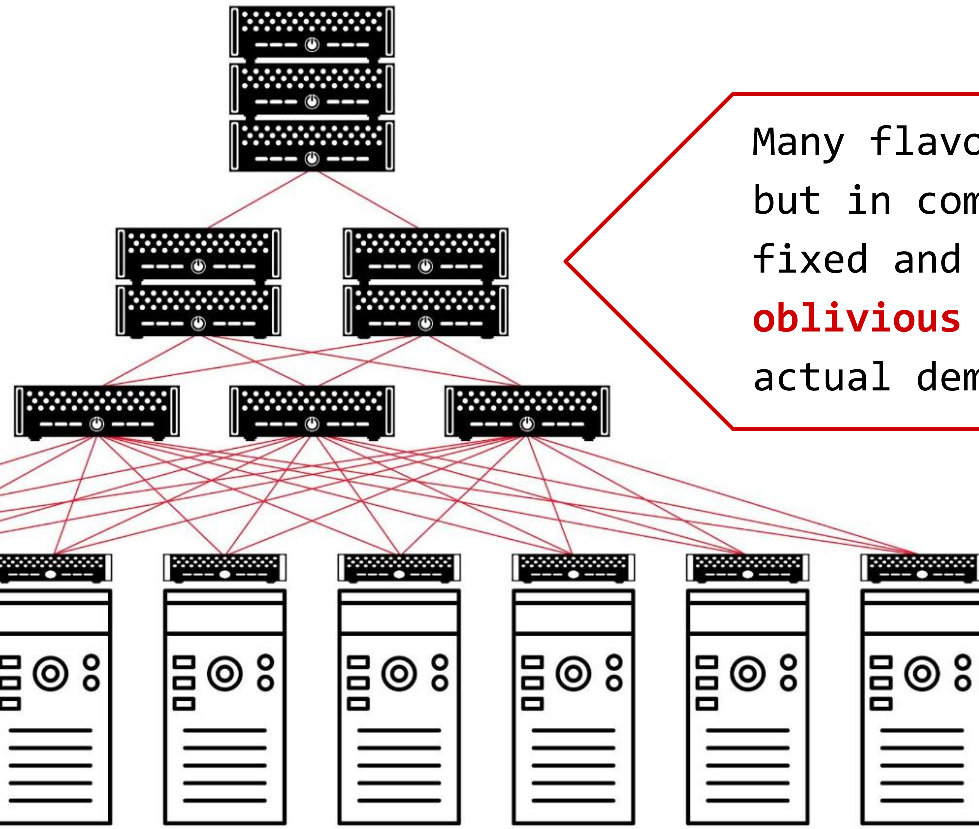


Many flavors,
but in common:
fixed and
oblivious to
actual demand.

Root Cause

Fixed and Demand-Oblivious Topology

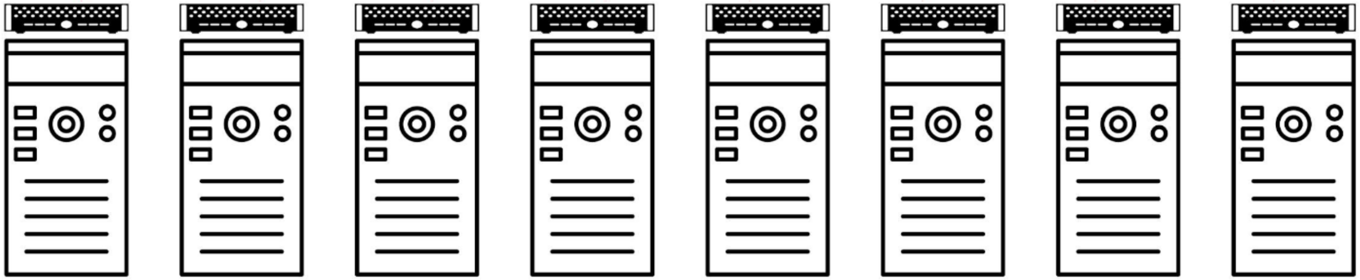
Highway which ignores actual traffic: frustrating!



Many flavors, but in common: fixed and **oblivious** to actual demand.

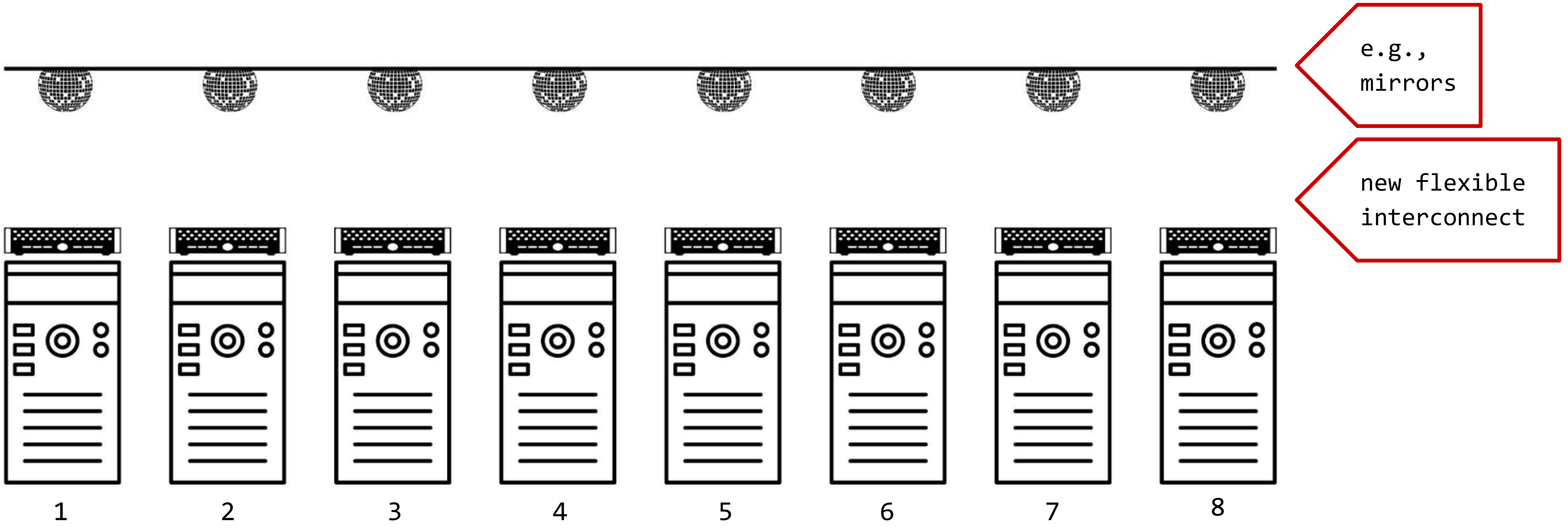
A Vision

Flexible and Demand-Aware Topologies



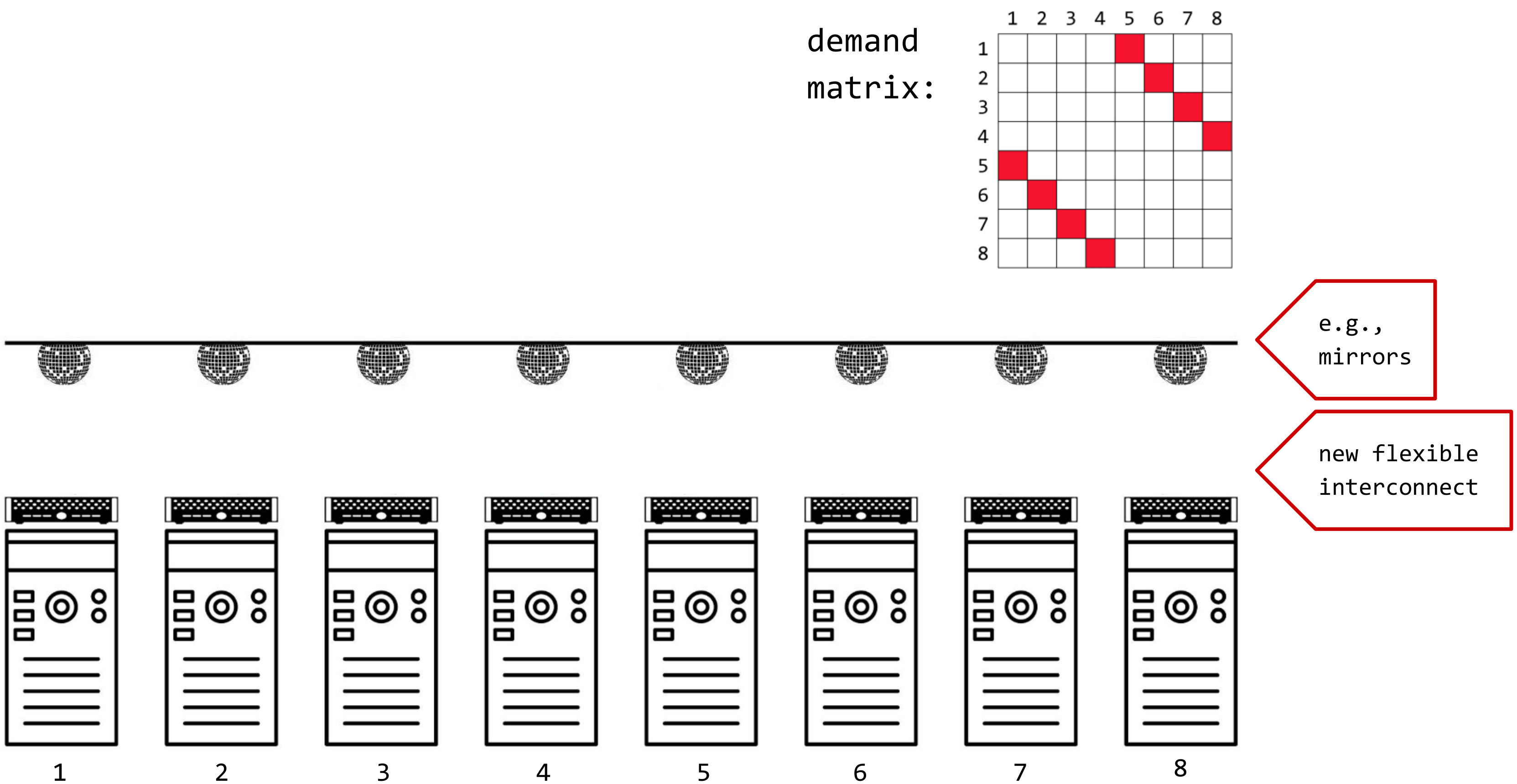
A Vision

Flexible and Demand-Aware Topologies



A Vision

Flexible and Demand-Aware Topologies

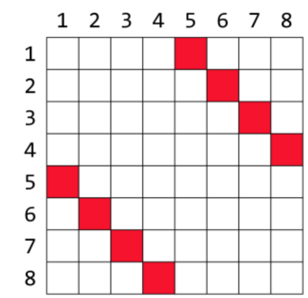


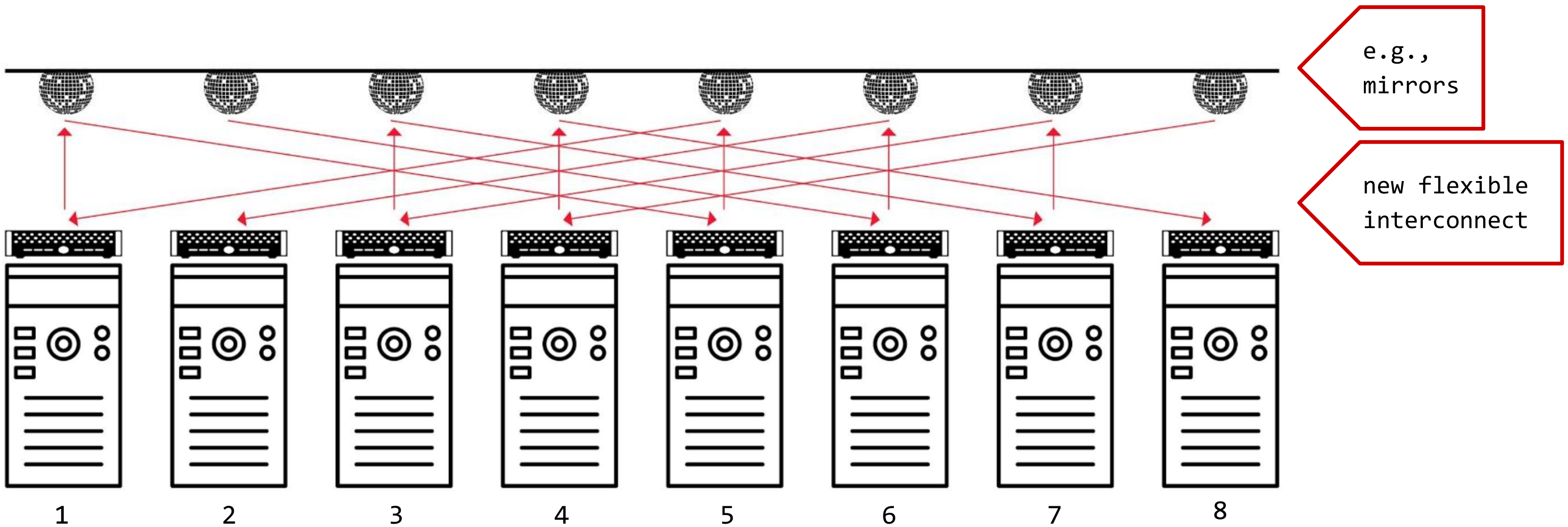
A Vision

Flexible and Demand-Aware Topologies

Matches demand

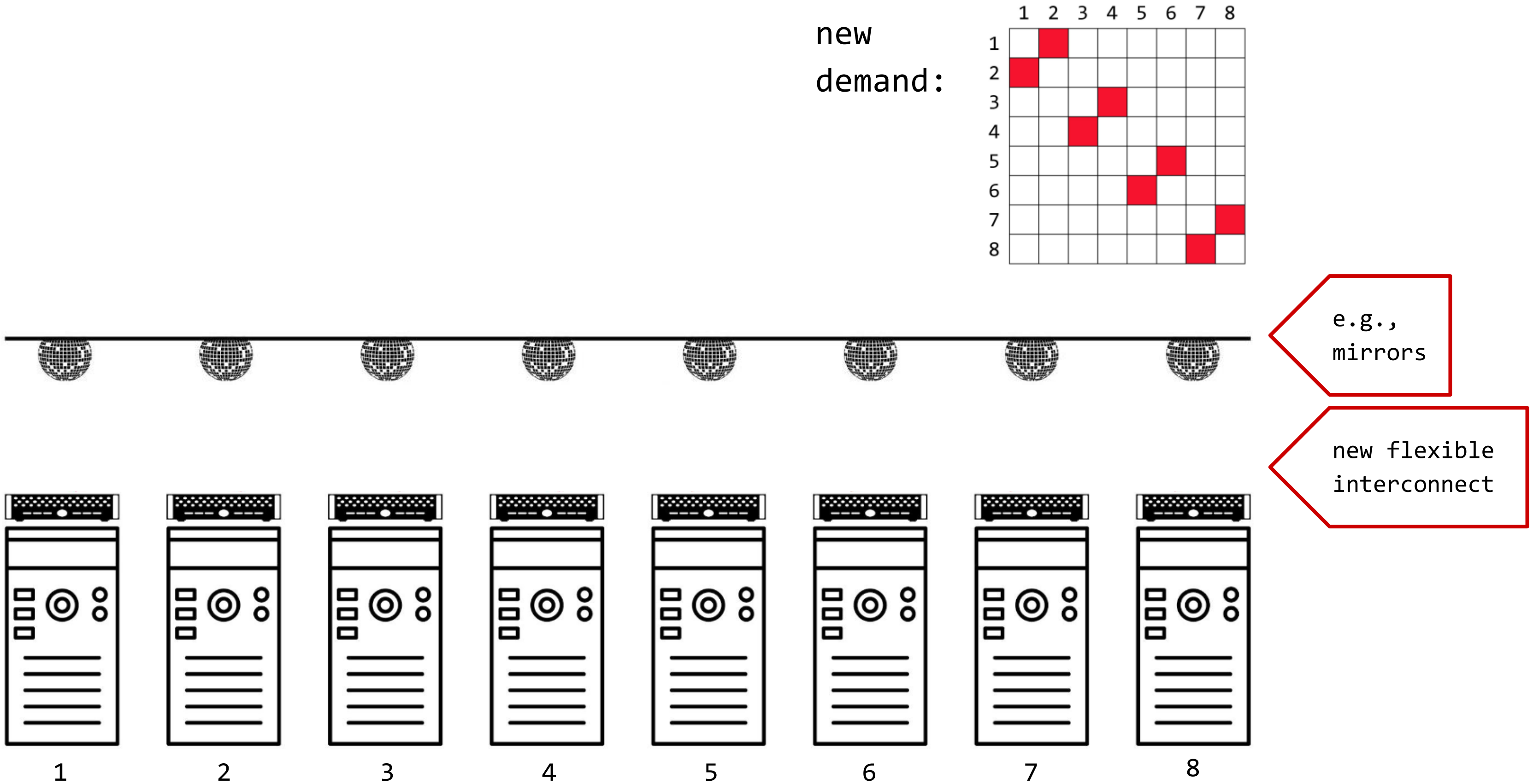
demand matrix:





A Vision

Flexible and Demand-Aware Topologies

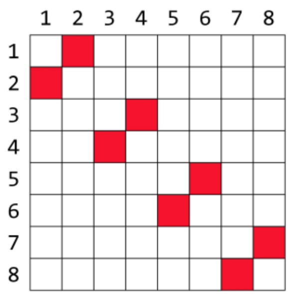


A Vision

Flexible and Demand-Aware Topologies

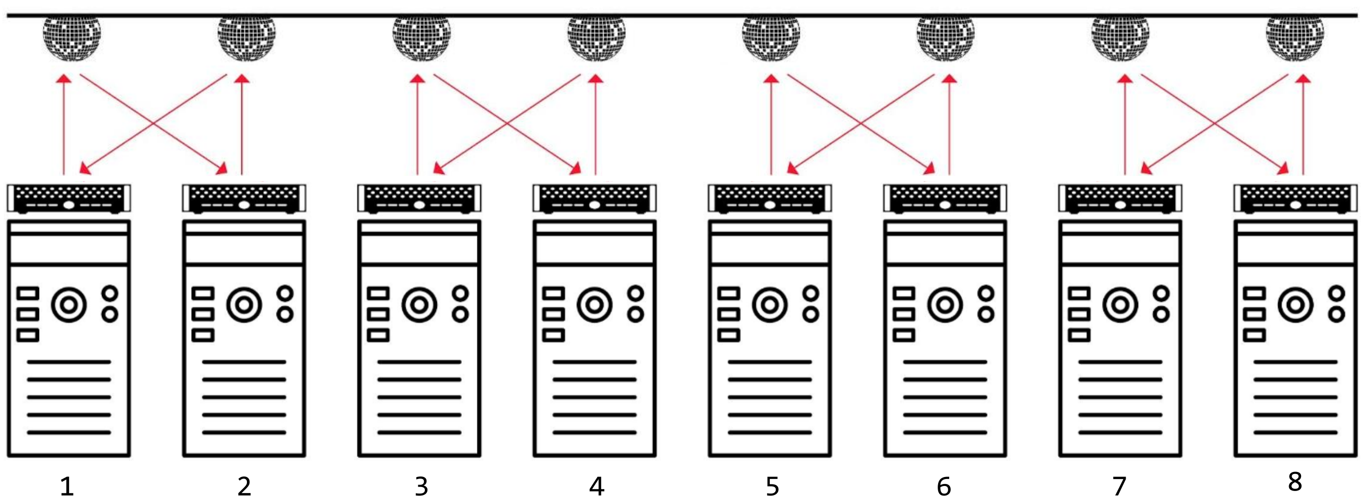
Matches demand

new demand:



e.g., mirrors

new flexible interconnect

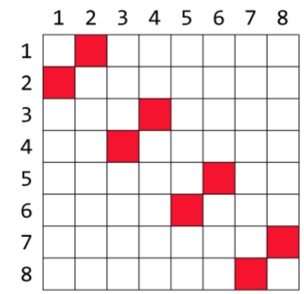


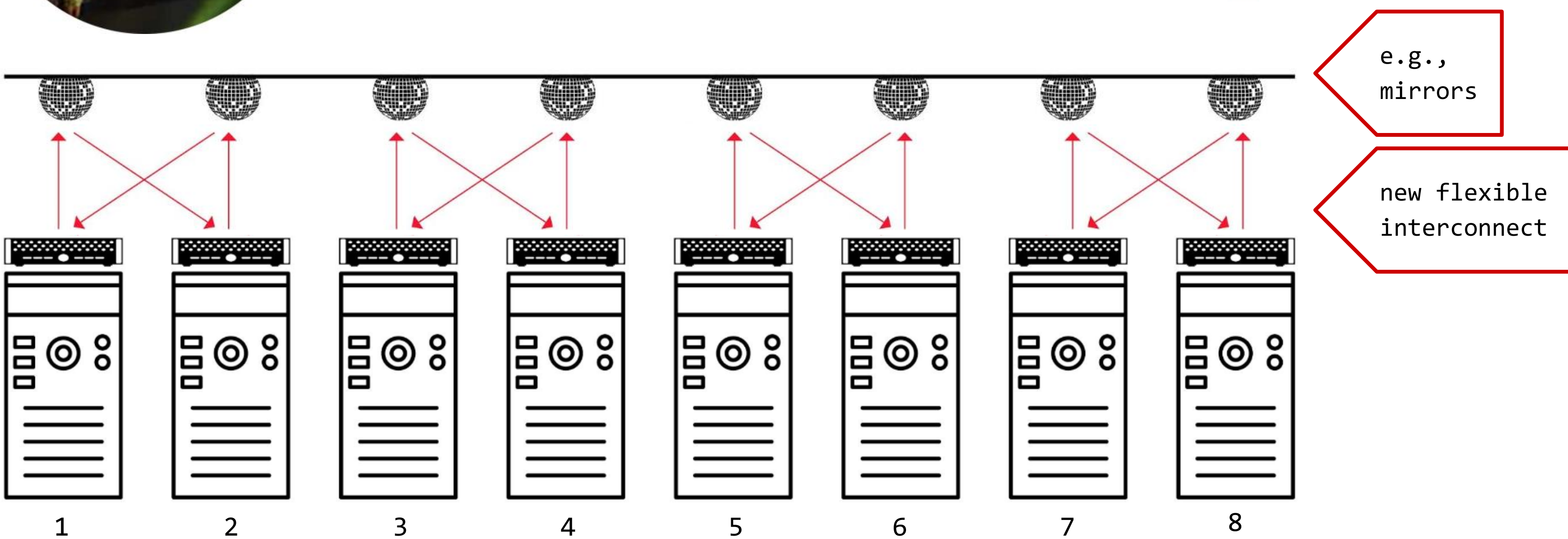
A Vision

Flexible and Demand-Aware Topologies

Demand-Aware Networks

new demand:



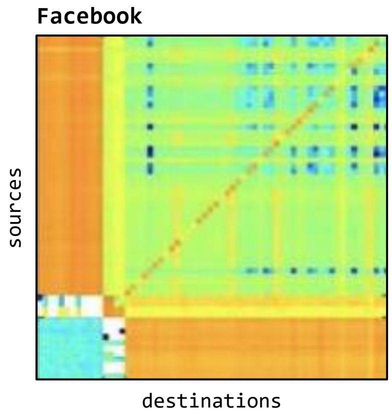


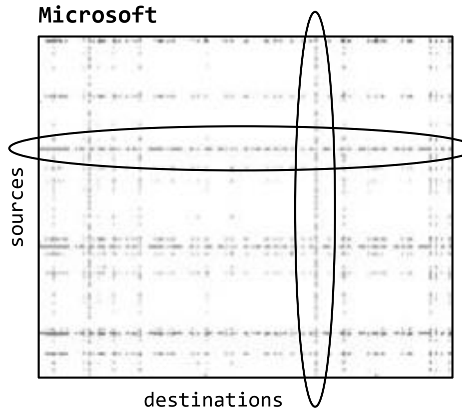
The Motivation

Much Structure in the Demand

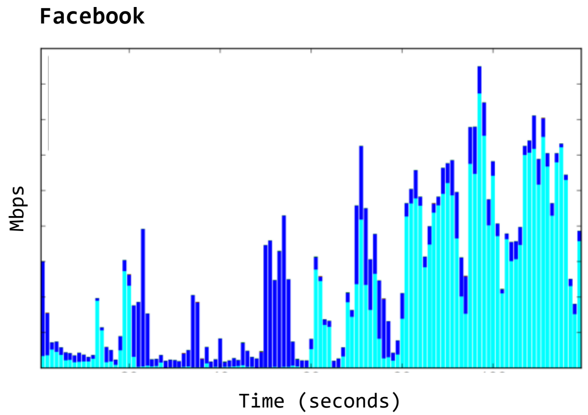
Empirical studies:

traffic matrices sparse and skewed



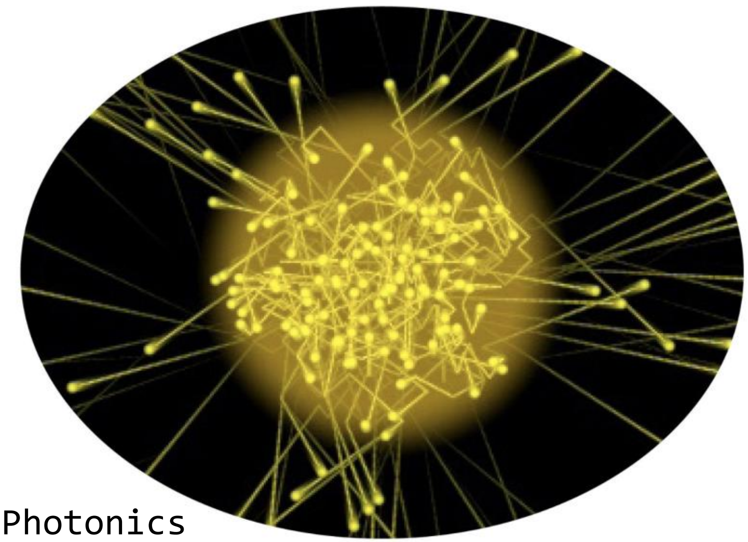


traffic bursty over time



Hypothesis: this can be exploited.

Sounds Crazy? Emerging Enabling Technology.



Photonics

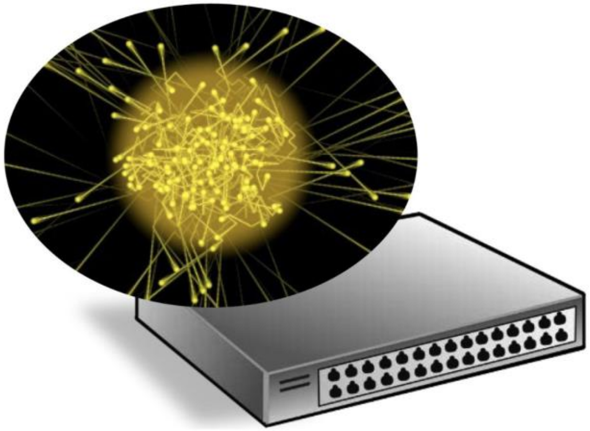
H2020:
"Photonics one of only five key enabling technologies for future prosperity."

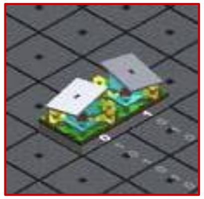
US National Research Council:
"Photons are the new Electrons."

Enabler

Novel Reconfigurable Optical Switches

- **Spectrum** of prototypes
  - Different sizes, different reconfiguration times
  - From our last years' ACM **SIGCOMM** workshop OptSys





Prototype 1

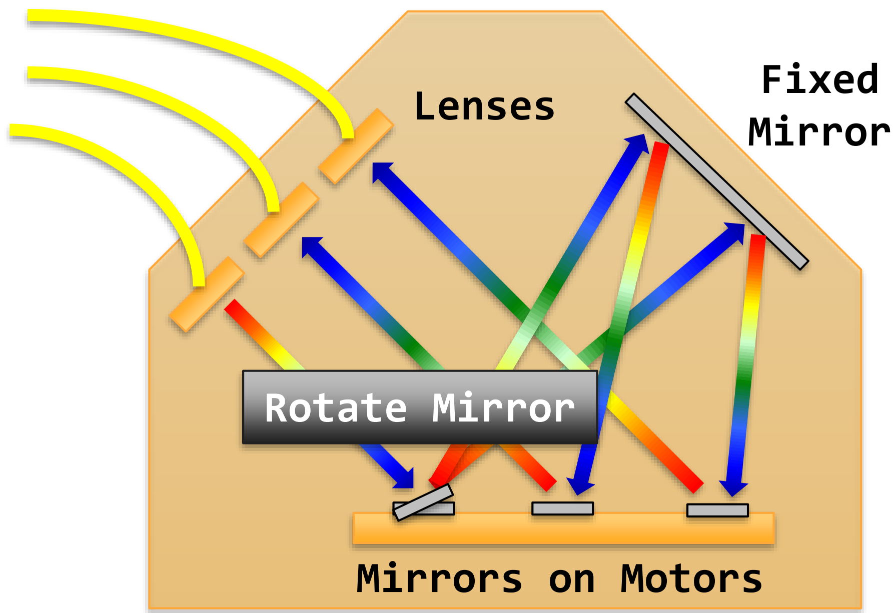
Prototype 2

Prototype 3

Example

Optical Circuit Switch

- Optical Circuit Switch rapid adaption of physical layer
  - Based on rotating mirrors



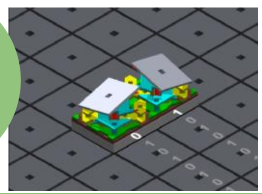
Optical Circuit Switch

By Nathan Farrington, SIGCOMM 2010

The Big Picture

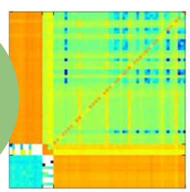
New!

Flexibility



More!

Structure



Self-Adjusting Networks

Efficiency

Now is the time!

Unique Position

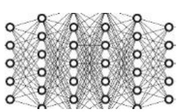
Demand-Aware, Self-Adjusting Systems

Everywhere, but mainly in software

Algorithmic trading

NETFLIX

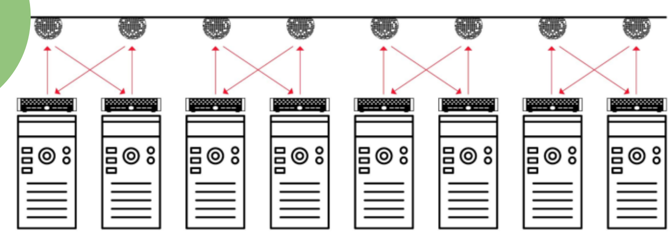
Recommender systems



Neural networks

VS

Our focus: in hardware

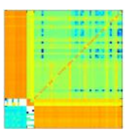


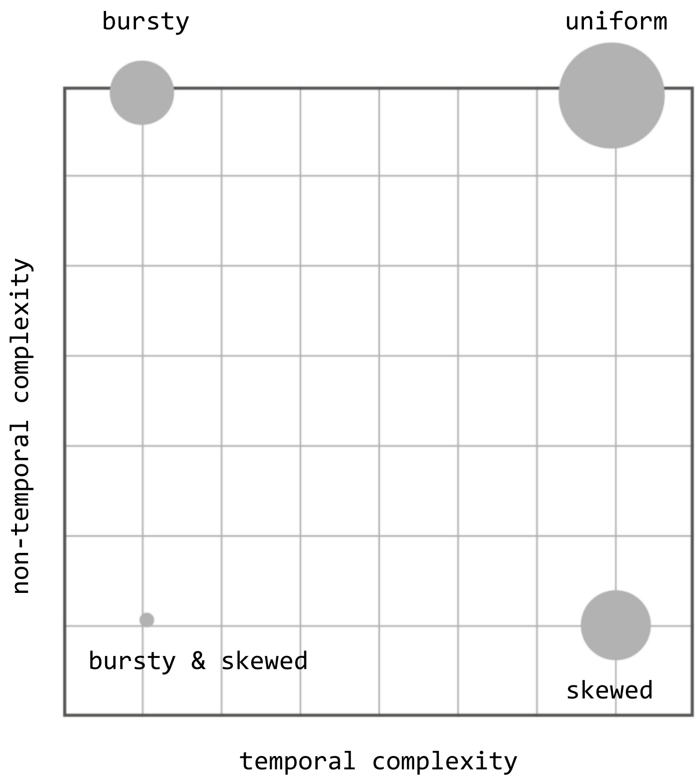
Question 1:

How to Quantify such “Structure” in the Demand?

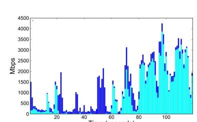
An Information-Theoretic Approach

Complexity Map





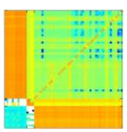
No structure

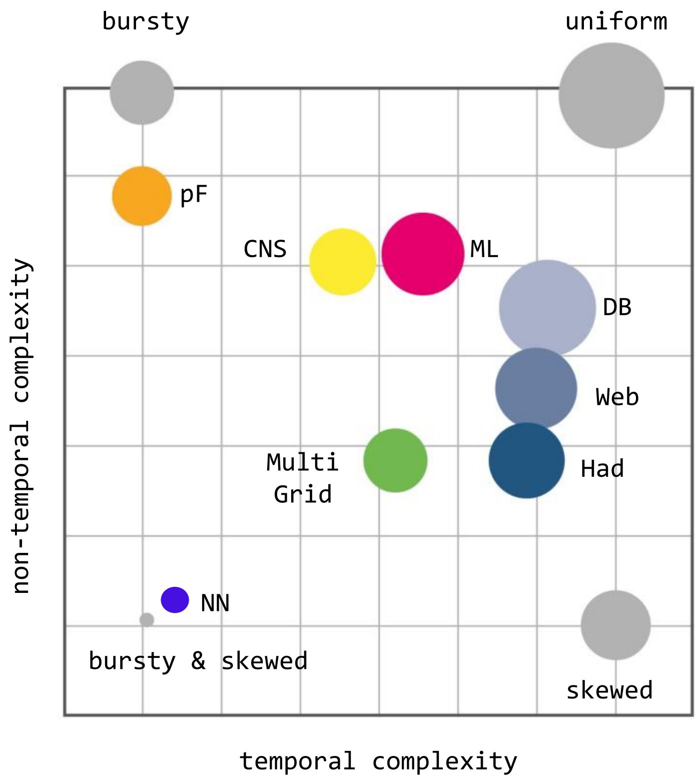


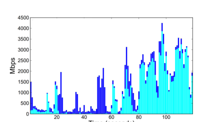
Our **approach:** iterative **randomization and compression** of trace to identify dimensions of structure.

An Information-Theoretic Approach

Complexity Map







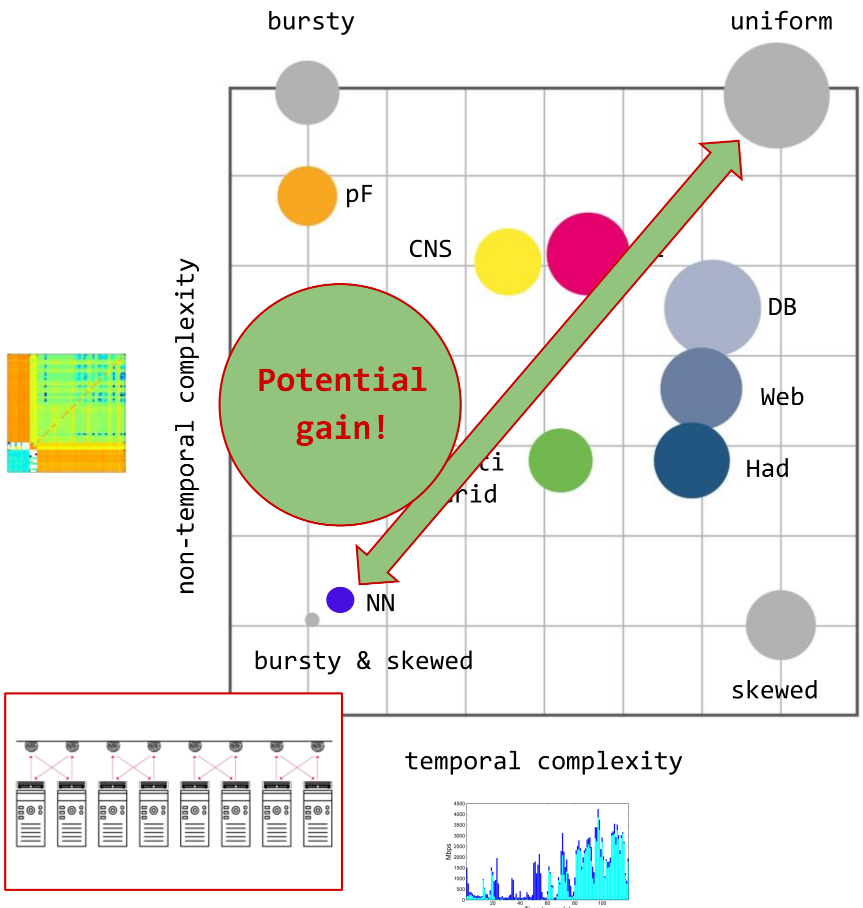
No structure

Our **approach**: iterative **randomization and compression** of trace to identify dimensions of structure.

Different structures!

An Information-Theoretic Approach

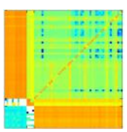
Complexity Map

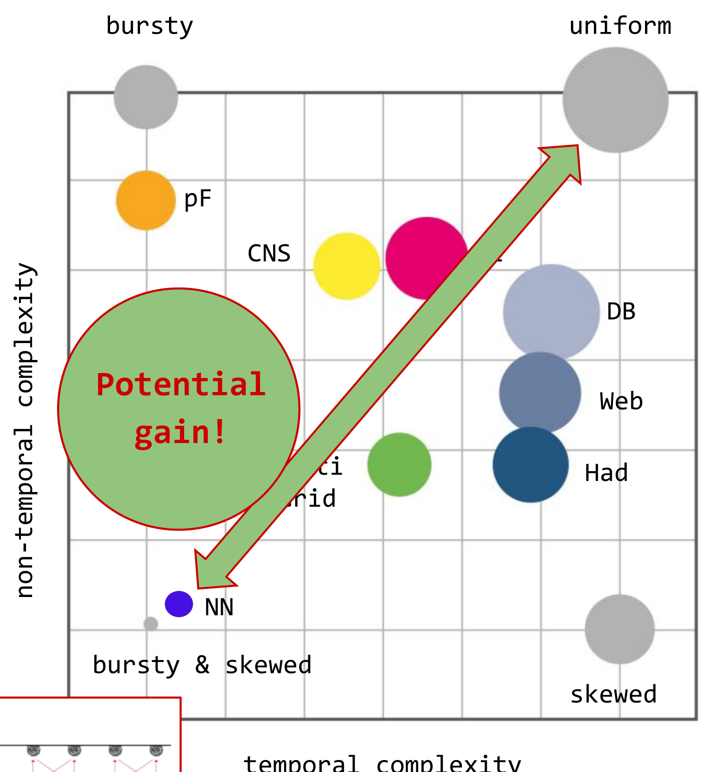


Our **approach**: iterative **randomization and compression** of trace to identify dimensions of structure.

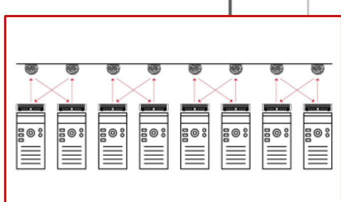
An Information-Theoretic Approach

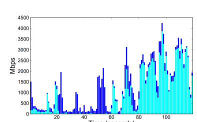
Complexity Map





Our **approach**: iterative **randomization and compression** of trace to identify dimensions of structure.





Griner et al., Sigmetrics 2020

Question 2:

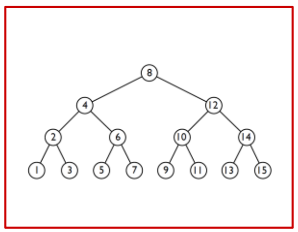
Given This Structure, What Can Be Achieved? Metrics and Algorithms?

A first insight: entropy of the demand.

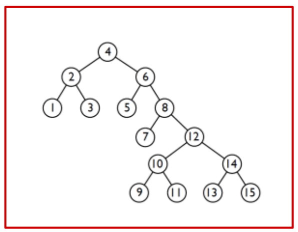
Interesting Perspective:

Connection to Datastructures

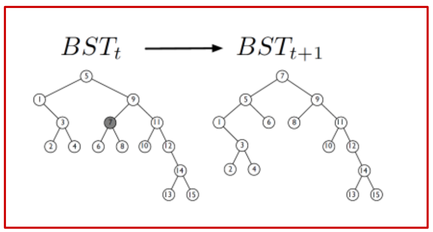
Traditional BST



Demand-aware BST



Self-adjusting BST

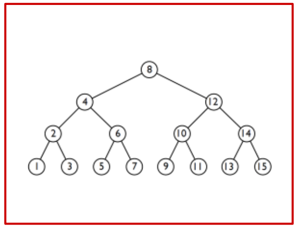


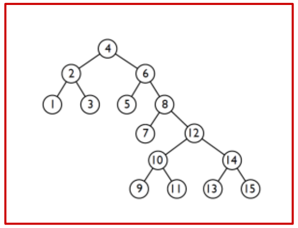
More structure: improved **access cost**

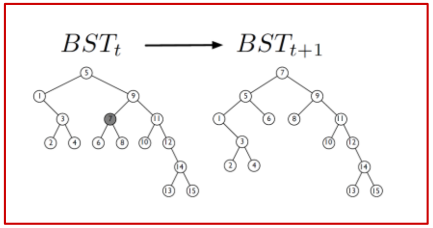
Interesting Perspective:

Connection to Datastructures & Coding

Traditional BST (Worst-case coding)	Demand-aware BST (Huffman coding)	Self-adjusting BST (Dynamic Huffman coding)





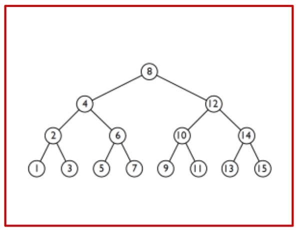


More structure: improved **access cost** / shorter **codes**

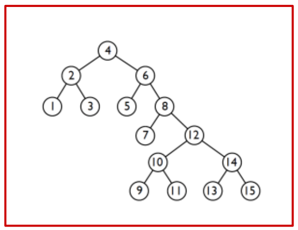
Interesting Perspective:

Connection to Datastructures & Coding

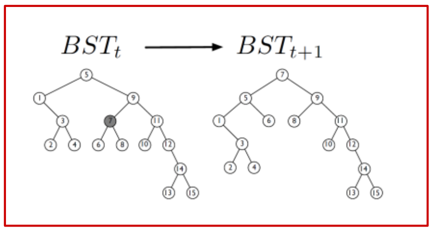
Traditional BST
(Worst-case coding)



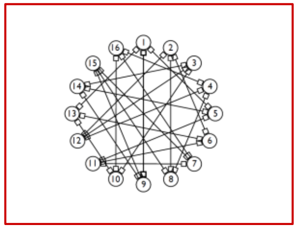
Demand-aware BST
(Huffman coding)

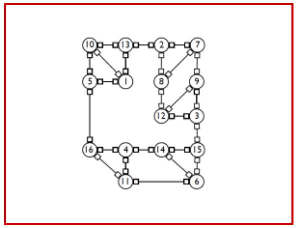


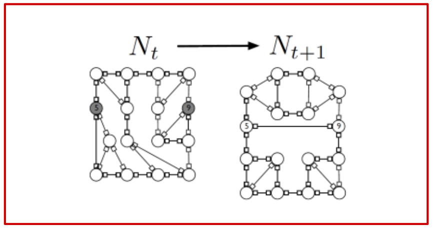
Self-adjusting BST
(Dynamic Huffman coding)



More structure: improved **access cost** / shorter **codes**







Similar **benefits**?

Interesting Perspective:

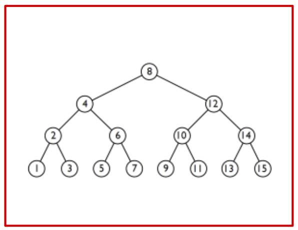
Connection to Datastructures & Coding

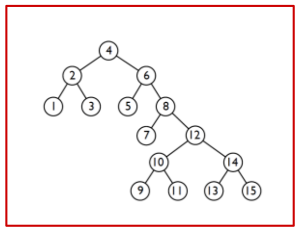
Traditional BST
(Worst-case coding)

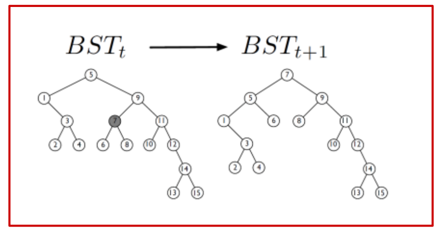
Demand-aware BST
(Huffman coding)

Self-adjusting BST
(Dynamic Huffman coding)

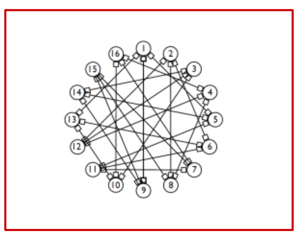
More than
an analogy!

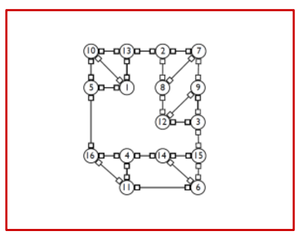


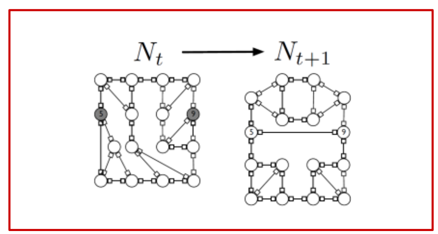




More structure: improved **access cost** / shorter **codes**







Similar **benefits**?

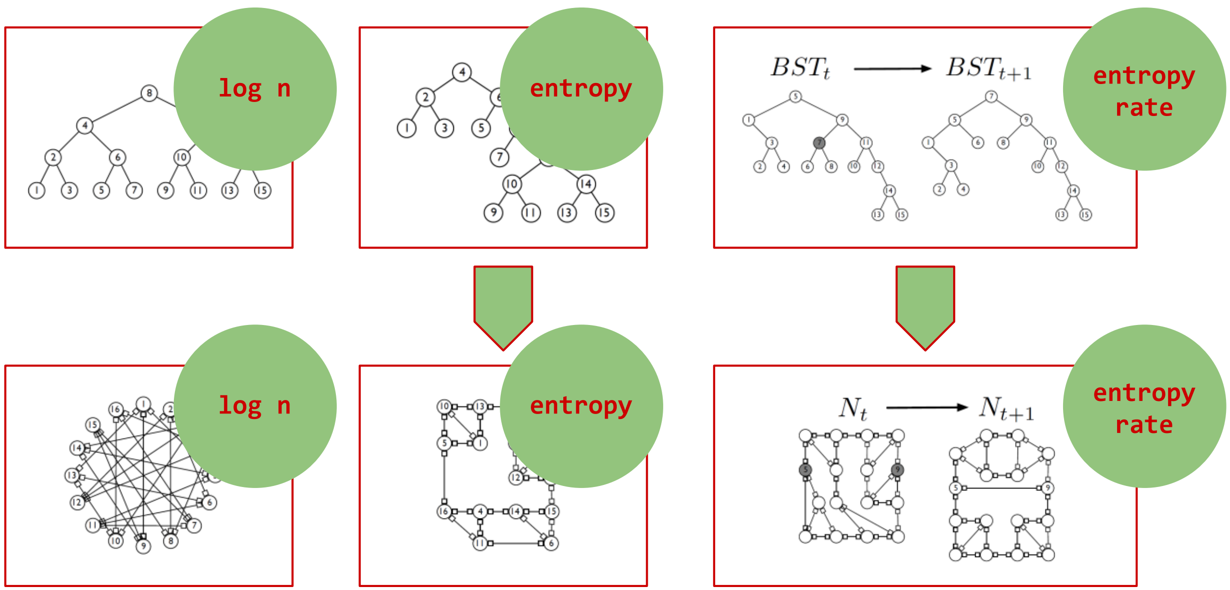
Interesting Perspective:

Connection to Datastructures & Coding

Traditional BST
(Worst-case coding)

Demand-aware BST
(Huffman coding)

Self-adjusting BST
(Dynamic Huffman coding)



Reduced expected **route lengths**!

More than
an analogy!

<u>Generalize methodology:</u>
... and transfer entropy bounds and algorithms of data-structures to networks.

<u>Results, e.g.:</u>
Demand-aware networks of asymptotically optimal route lengths.

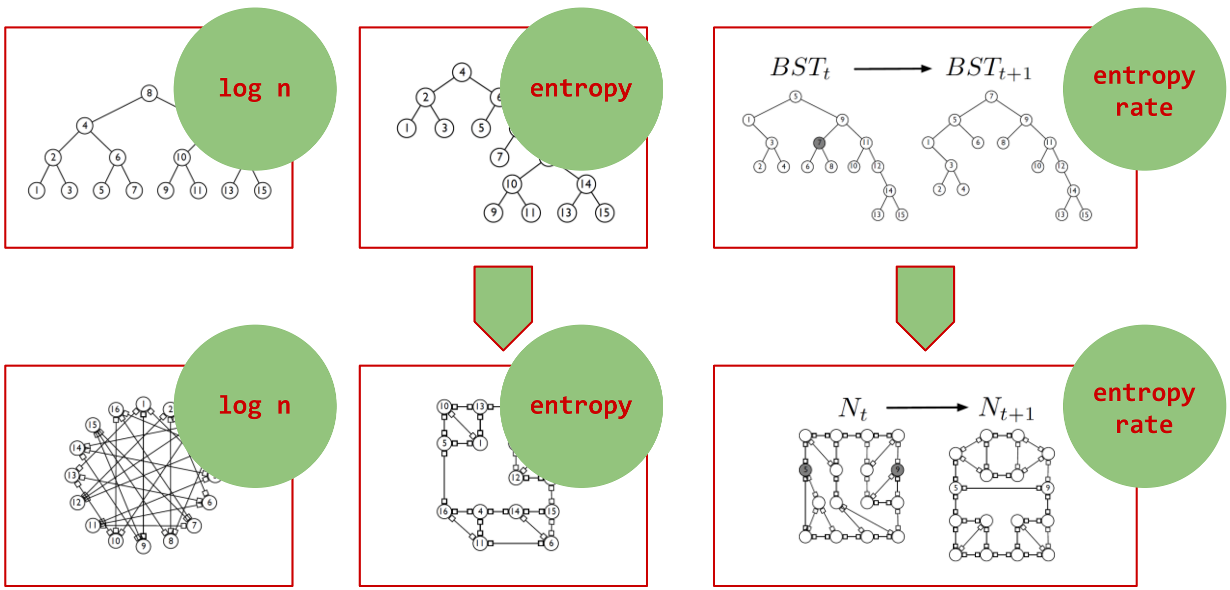
Interesting Perspective:

Connection to Datastructures & Coding

Traditional BST (Worst-case coding)

Demand-aware BST (Huffman coding)

Self-adjusting BST (Dynamic Huffman coding)



More than an analogy!

Generalize methodology:

... and transfer entropy bounds and algorithms of data-structures to networks.

Results, e.g.:

Demand-aware networks of asymptotically optimal route lengths.

Reduced expected **route lengths**!

Avin et al., SIGCOMM CCR 2018.

Agenda

Three Use Cases

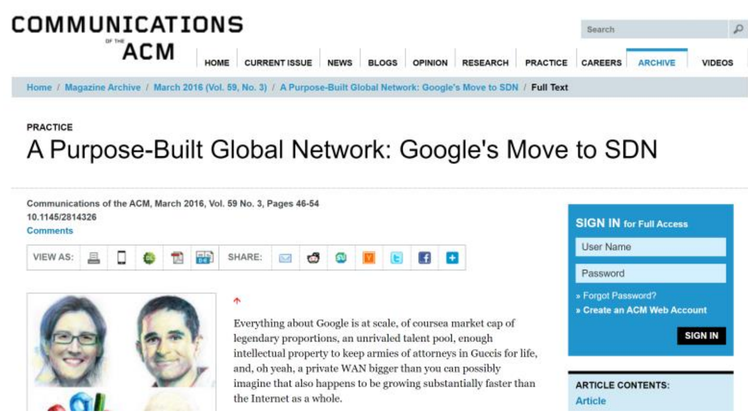
Passau, Germany

Another Benefit of Automation:

More Adaptive Operation

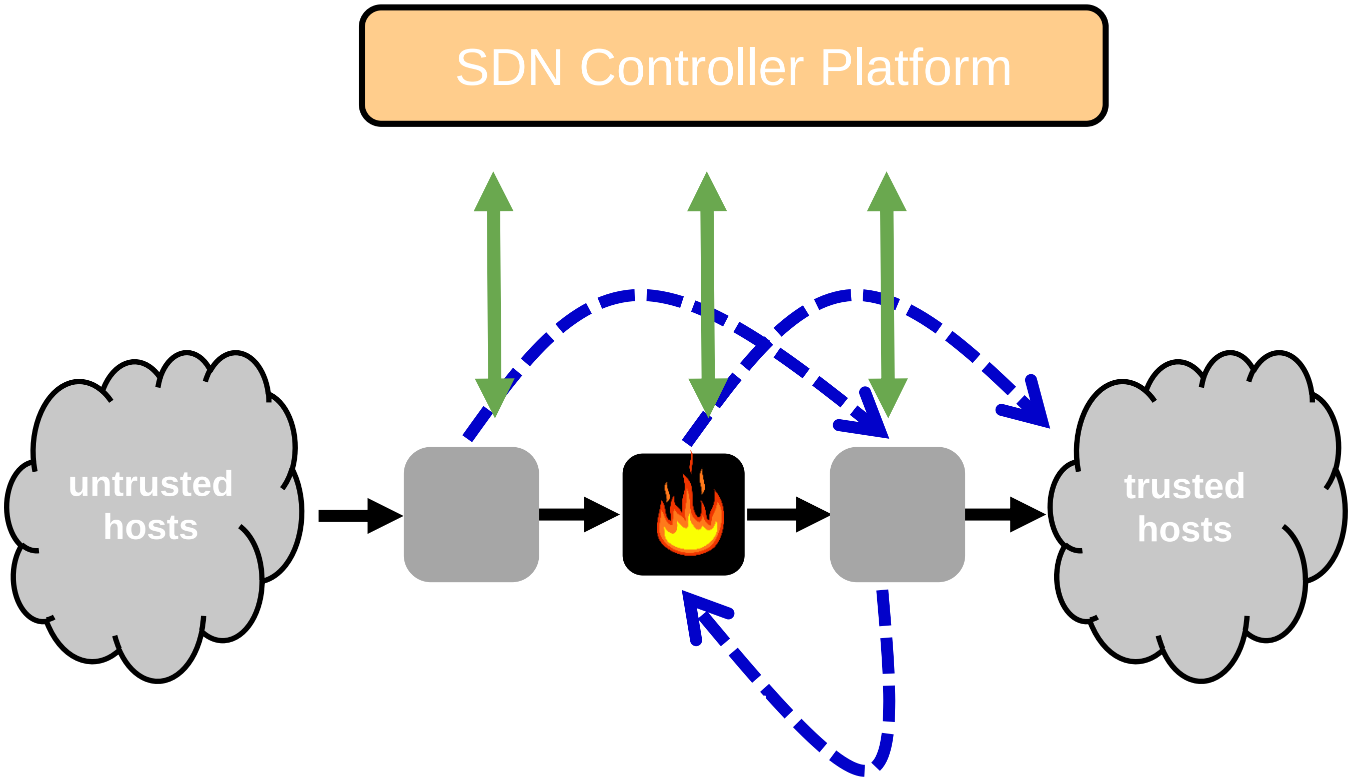
- Automation and programmability: enables more **adaptable networks**

- Attractive for:
  - Fine-grained traffic engineering (e.g., at Google)
  - Accounting for changes in the demand (spatio-temporal structure)
  - Security policy changes
  - Service relocation
  - Maintenance work
  - Link/node failures



Another Benefit of Automation:

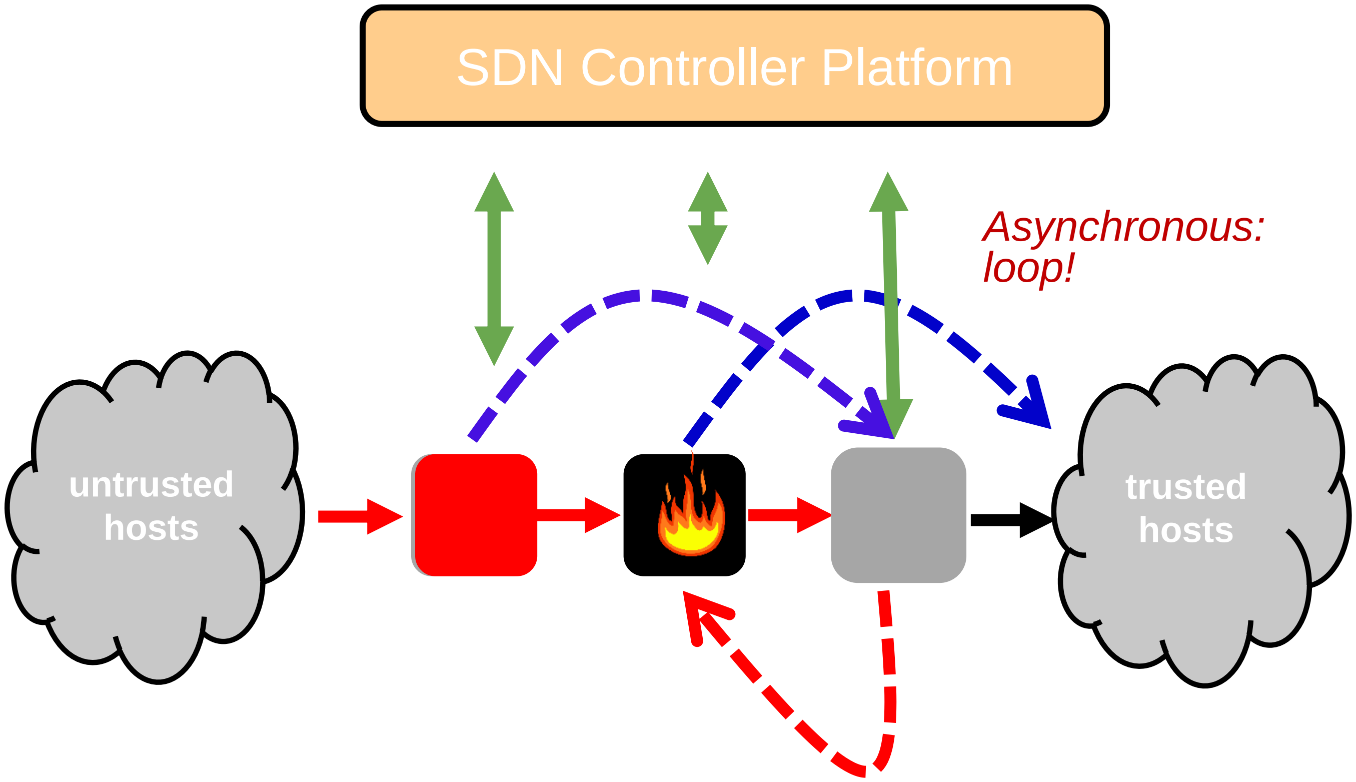
More Adaptive Operation



Enabled by SDN, it has become „easy“ to quickly change route to blue route.

Another Benefit of Automation:

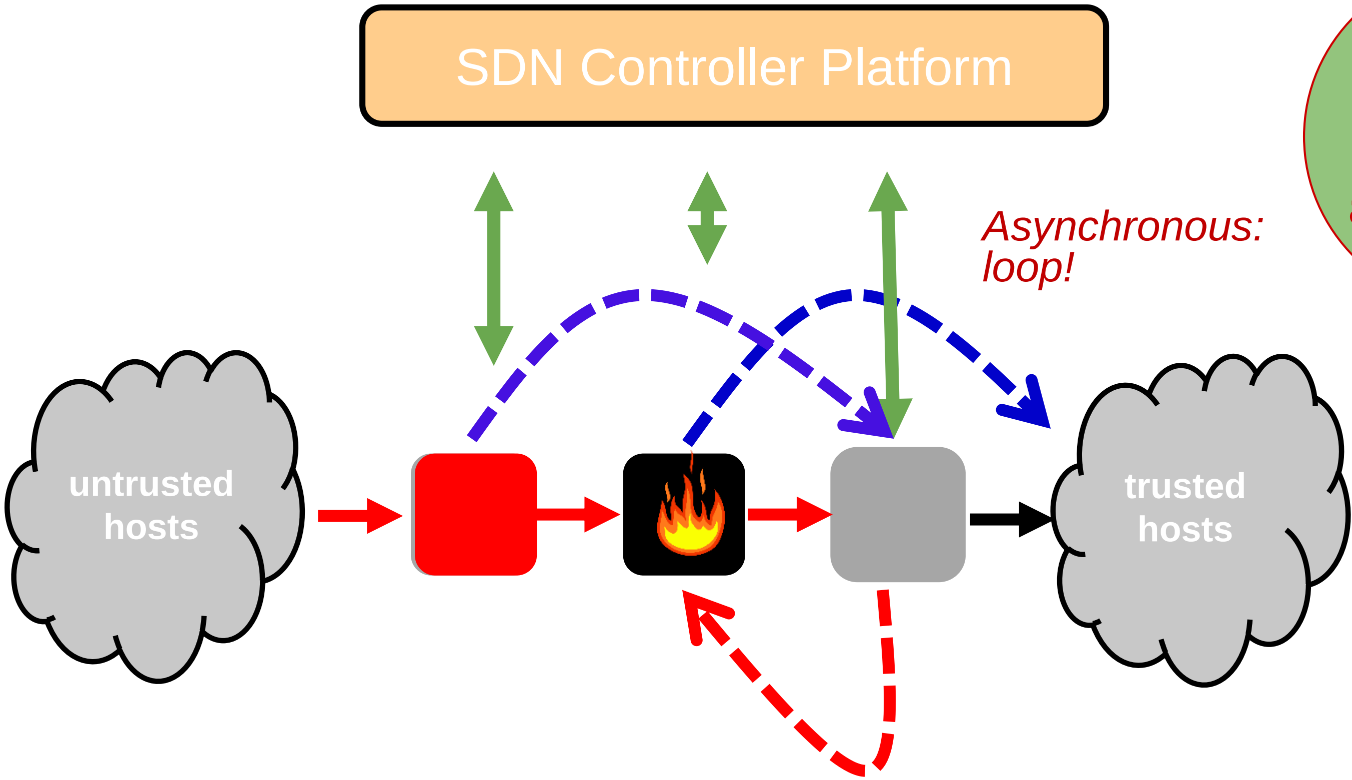
More Adaptive Operation



But still need clever algorithms! Updates are asynchronous, may lead to temporal inconsistencies.

Another Benefit of Automation:

More Adaptive Operation

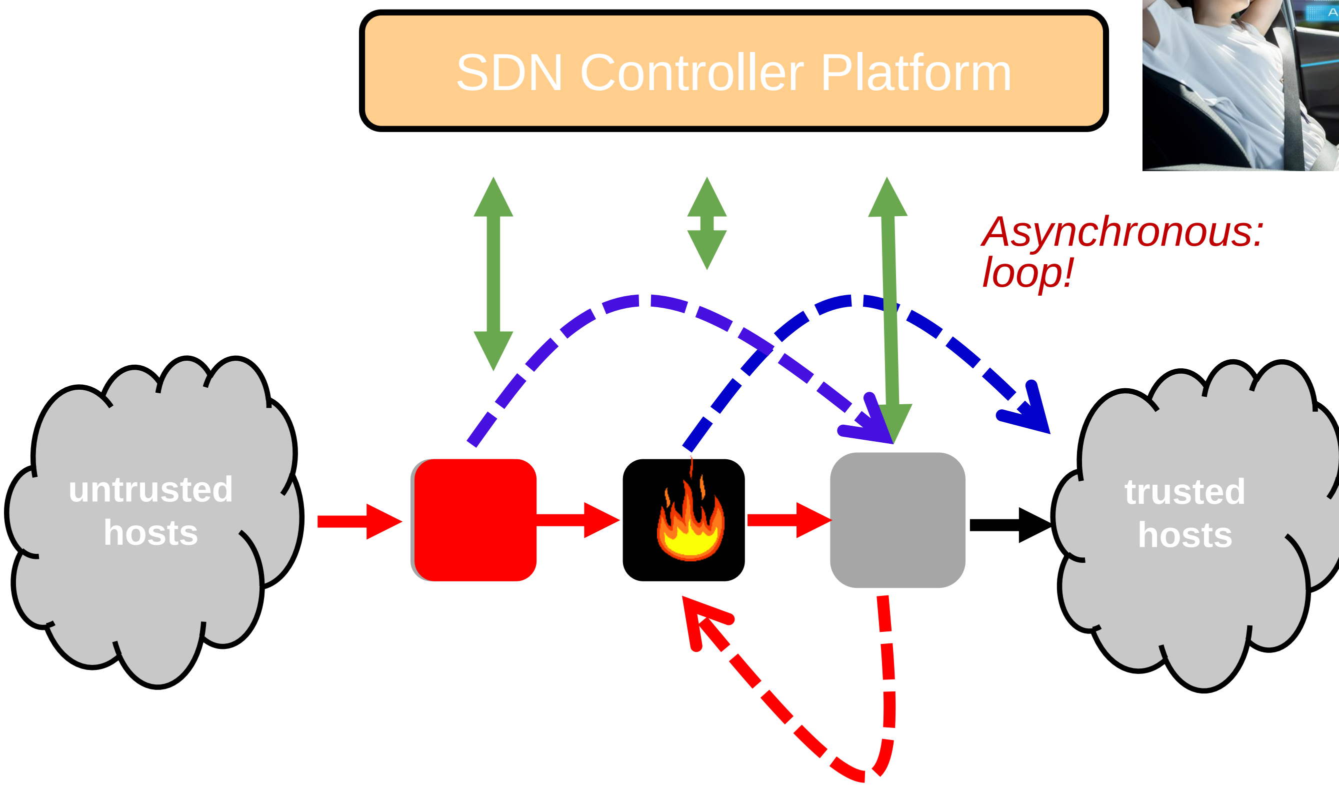


Survey: 100s of algorithms

But still need clever algorithms! Updates are asynchronous, may lead to temporal inconsistencies.

Again: Formal Methods for

Self-Adjusting Updates

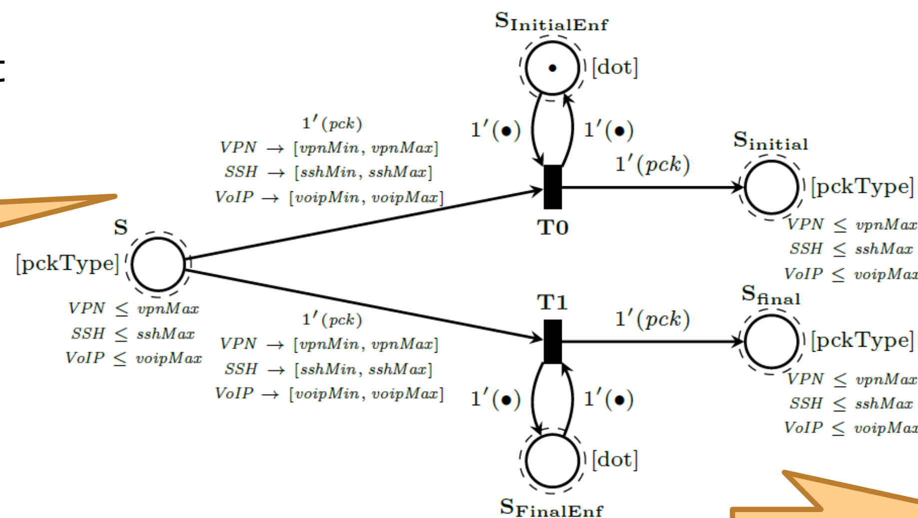


Vision: self-adjusting networks could synthesize even their algorithms! „Ex machina“: e.g., parametrized.

Examples: NetSynth, Latte

- Already “***in the making***”!
- NetSynth (PLDI’15): supports **any LTL property** and hence **operator preferences**. Then: standard framework to synthesize schedule.
- Latte (PER’20): fast Petri net model and ***synthesis***
- Example: Gadget

If token up here: packets go old path



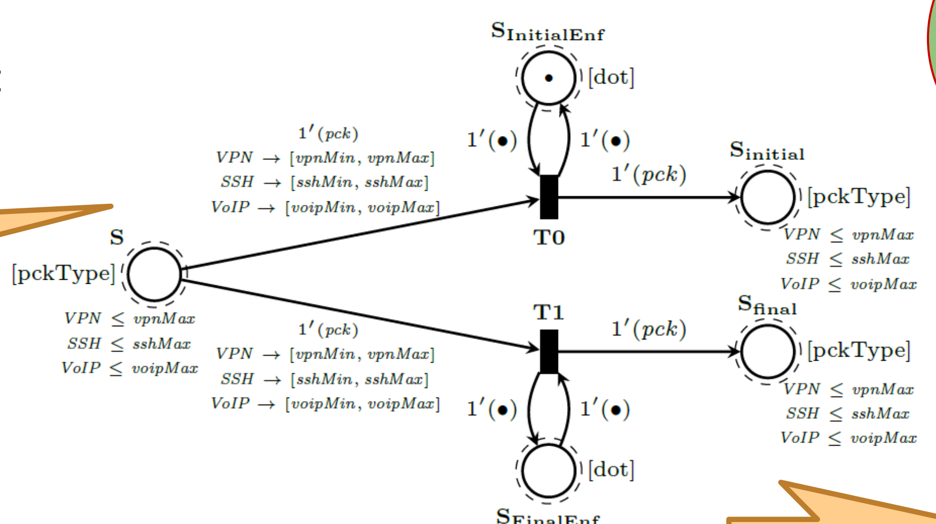
If token down here: switch updated to new path

Examples: NetSynth, Latte

- Already "*in the making*"!
- NetSynth (PLDI'15): supports **any LTL property** and hence **operator preferences**. Then: standard framework to synthesize schedule.
- Latte (PER'20): fast Petri net model and *synthesis*
- Example: Gadget

Formal methods!

If token up here: packets go old path



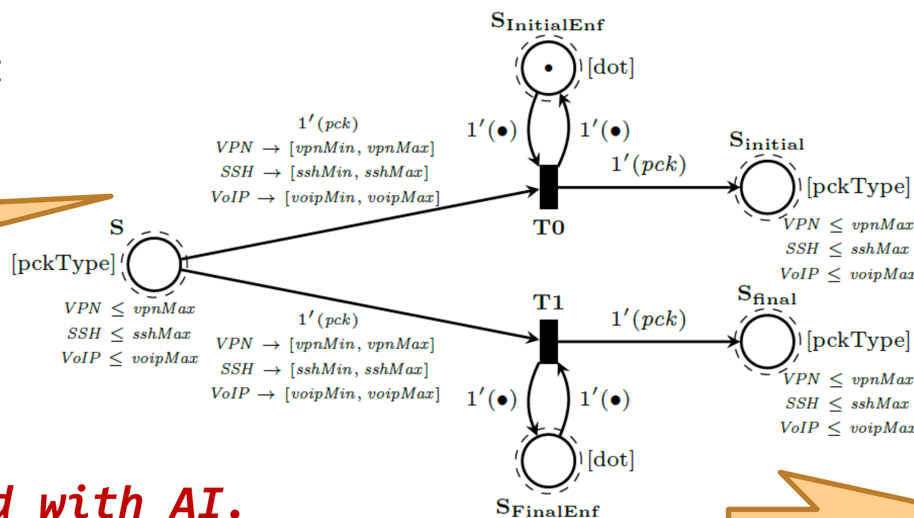
If token down here: switch updated to new path

Examples: NetSynth, Latte

- Already “*in the making*”!
- NetSynth (PLDI’15): supports **any LTL property** and hence **operator preferences**. Then: standard framework to synthesize schedule.
- Latte (PER’20): fast Petri net model and *synthesis*
- Example: Gadget

Formal methods!

If token up here: packets go old path



May be enhanced with AI.

If token down here: switch updated to new path

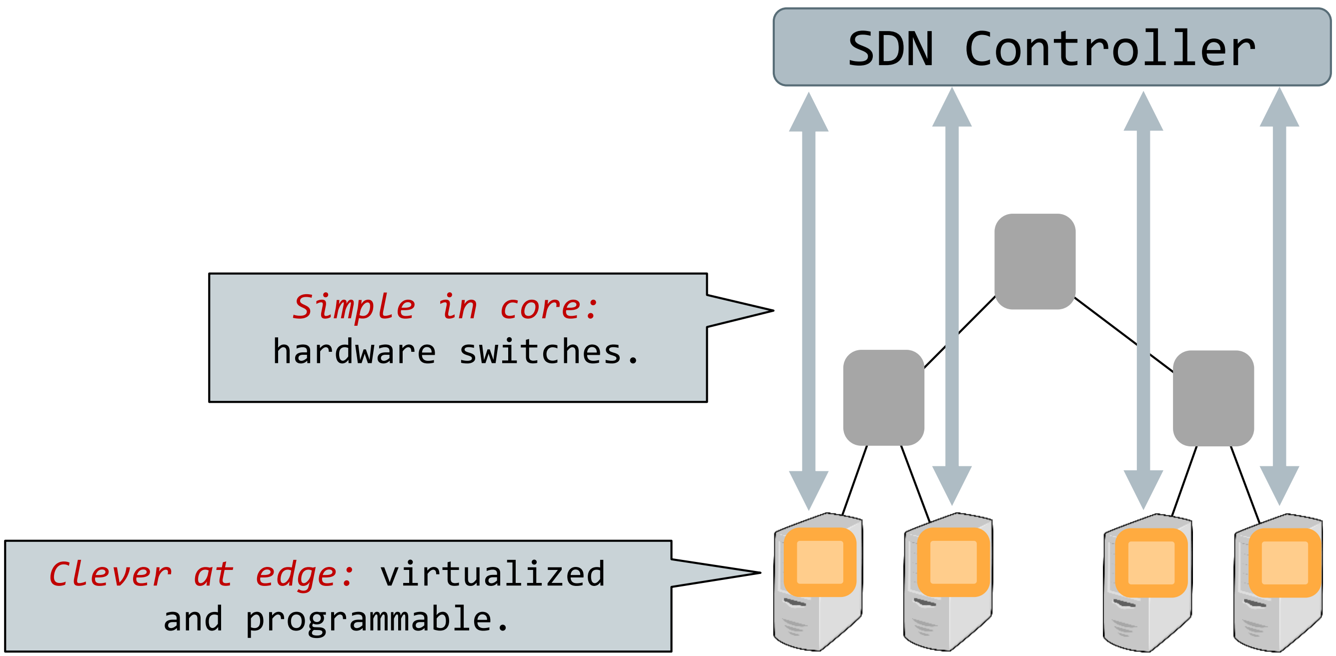
Challenges of Self-Adjusting Networks

Challenges

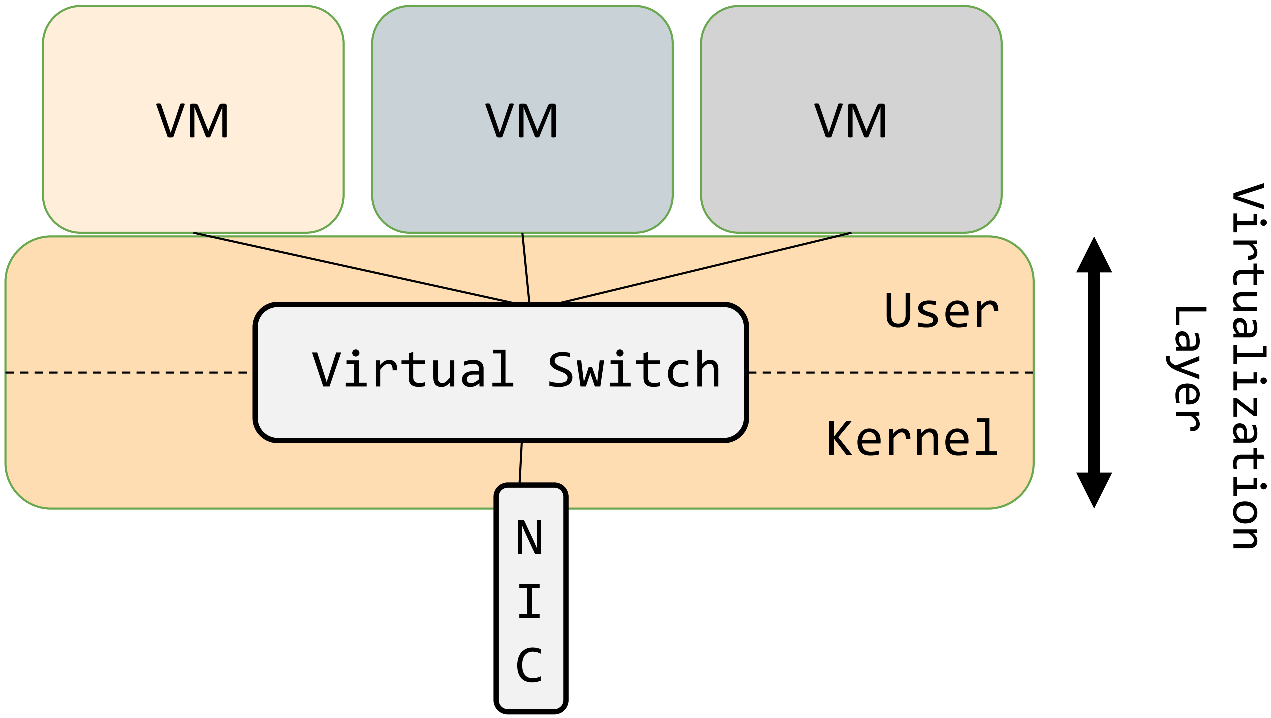
- ⇢ ***Performance*** of formal methods? Opportunity: ***algorithm engineering***!
- ⇢ Use of ***AI***: to speed up synthesis and deal with complexity?
- ⇢ Limitations of automation: can networks detect themselves, when the need “help from the operator”?
- ⇢ ***Data:*** How to learn about and/or predict demand? Telemetry?
- ⇢ Programmability vs ***security***?

Example: Security

- Enabling technology like SDN often deployed “***in software***”
- E.g., virtual switches in datacenters



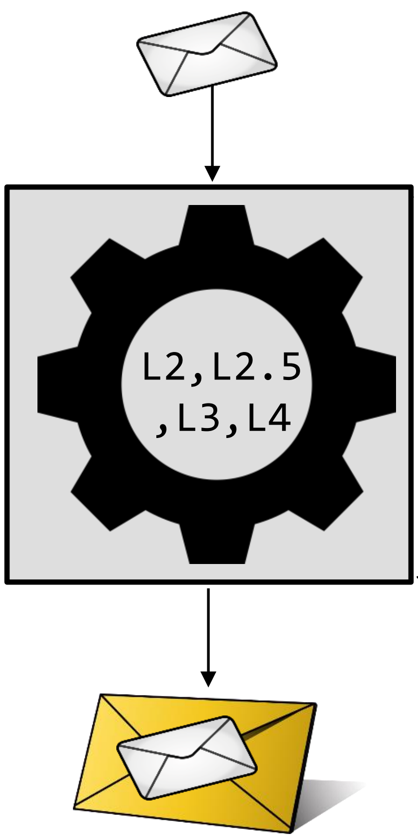
Virtual Switches

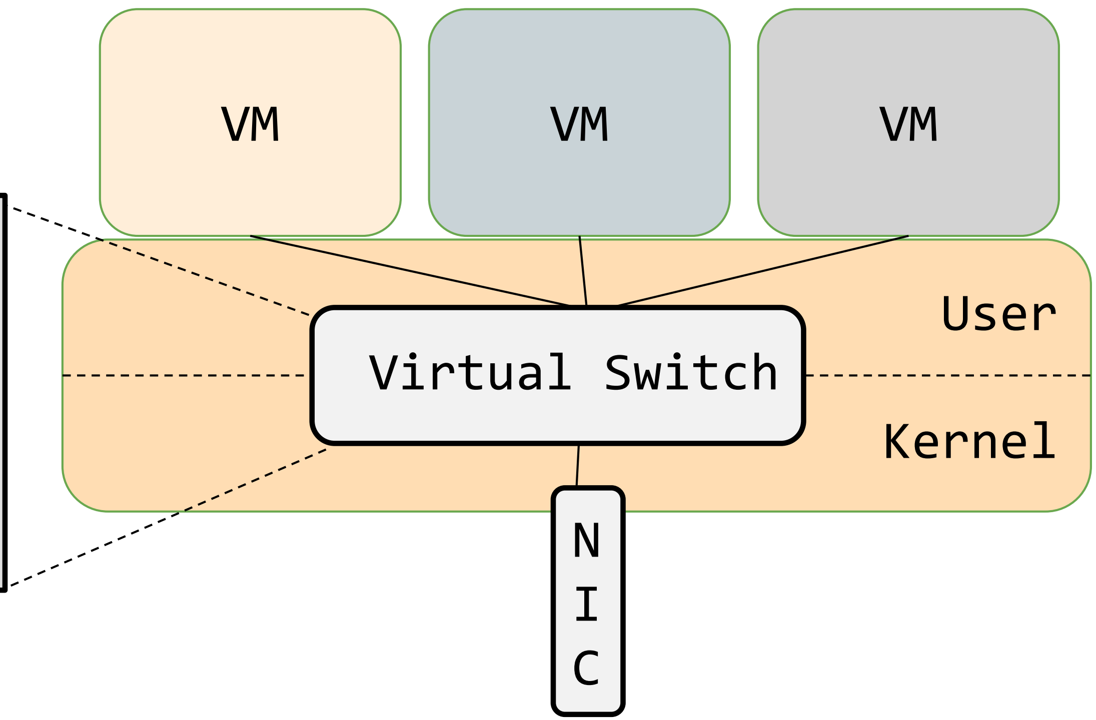


Virtual switches reside in the **server's virtualization layer** (e.g., Xen's Dom0).
Goal: provide connectivity and isolation.

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.

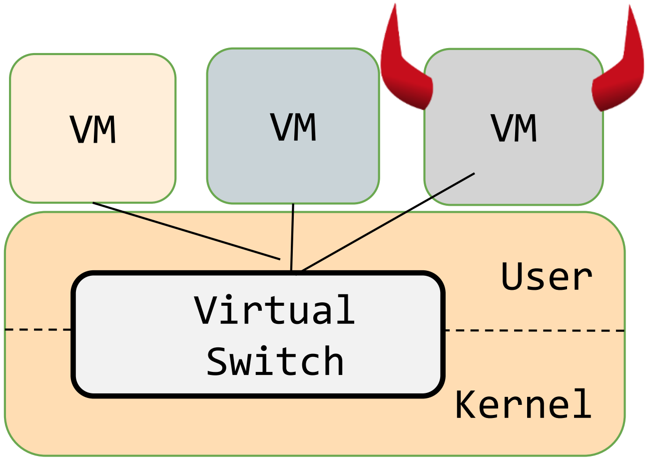
Parsing must be fast!

Unified packet parser is fast, but complex.

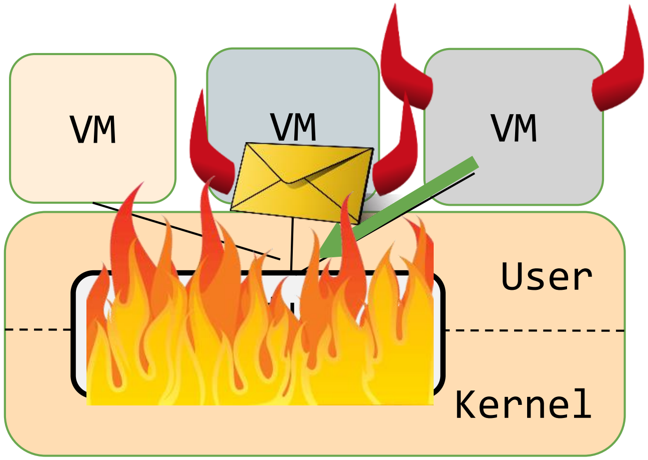
Credits:

Marco Chiesa

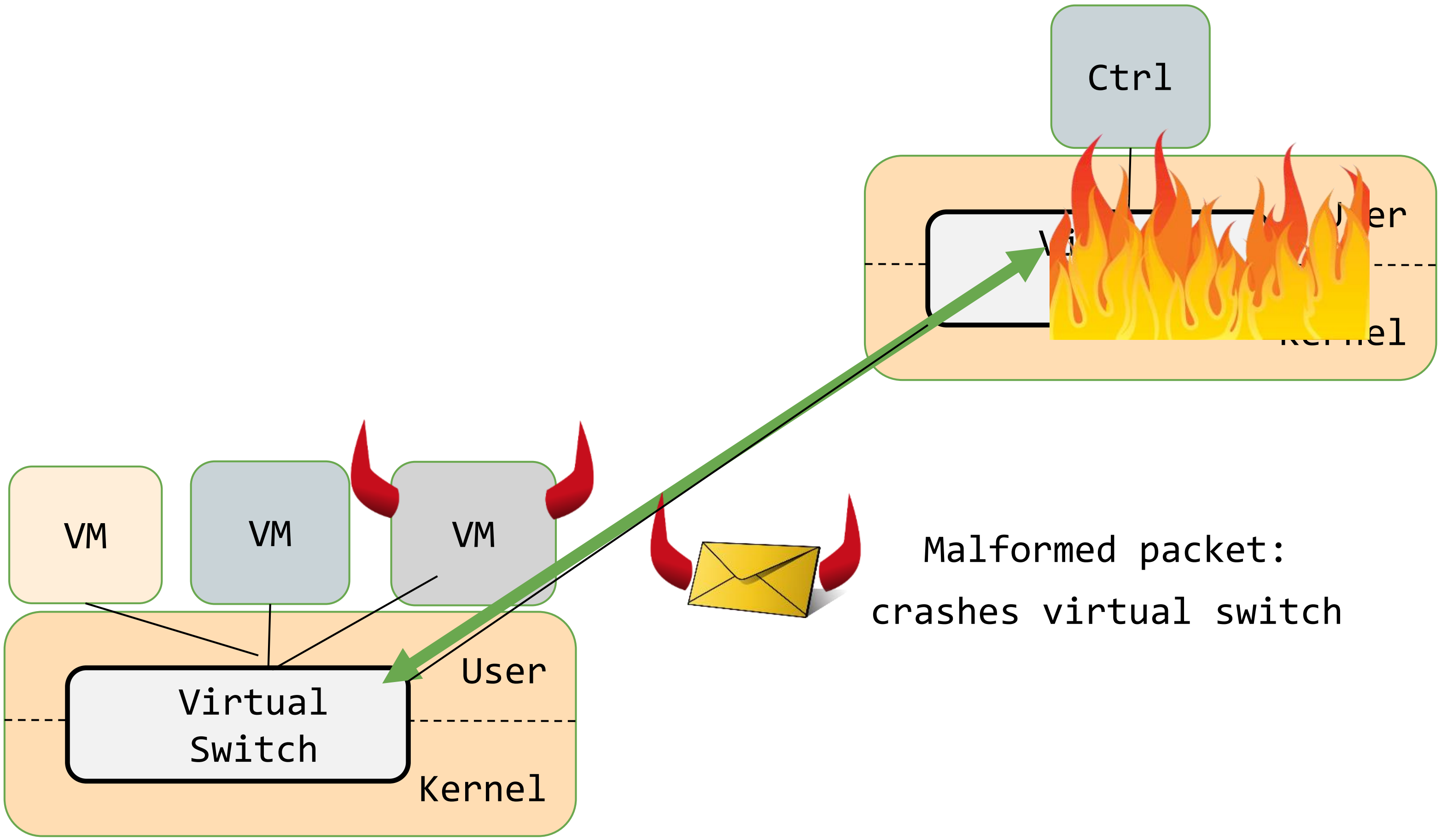
Bears Risks!



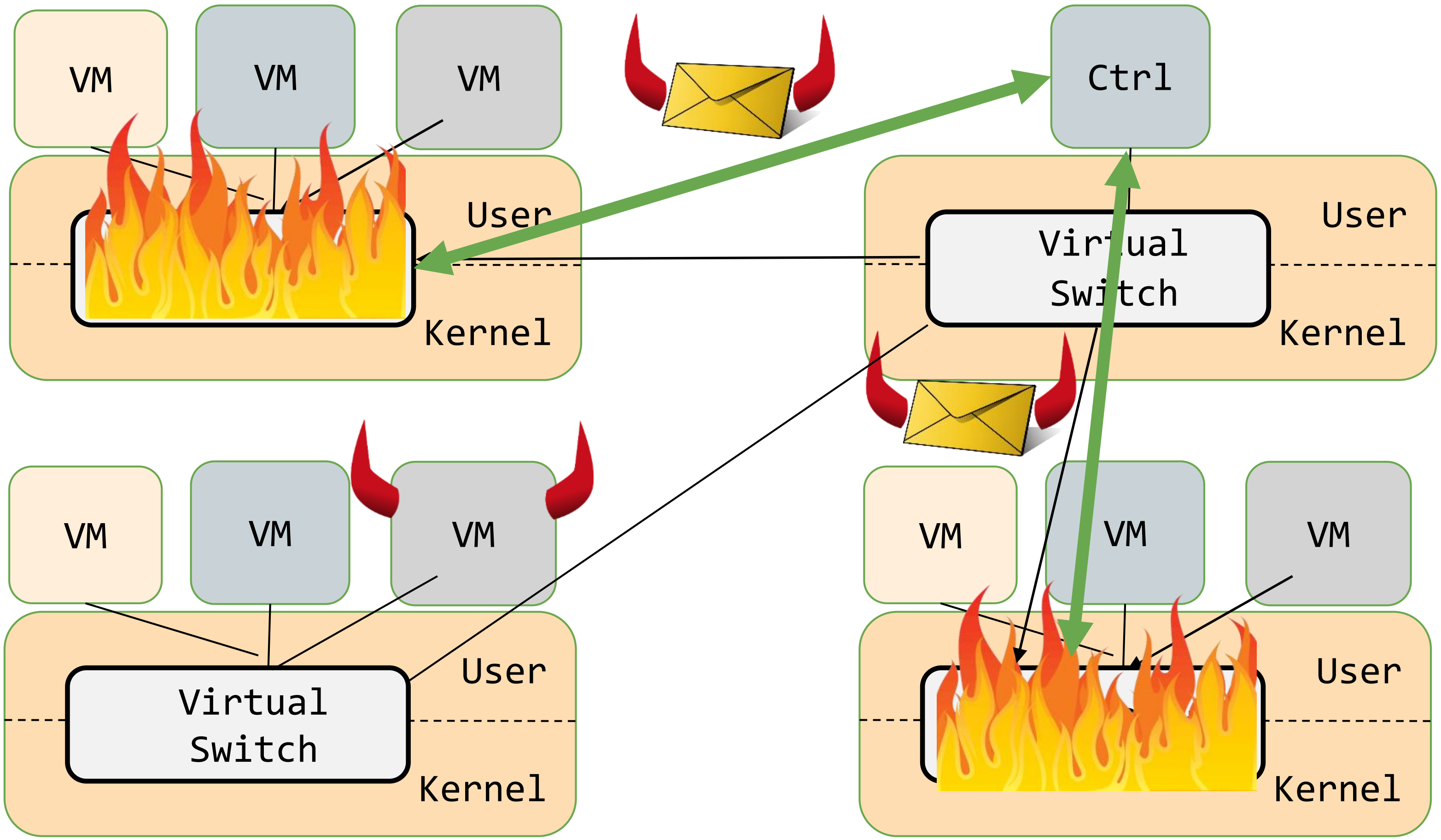
Bears Risks!



Bears Risks!


Ctrl
VM
VM
VM
Virtual
Switch
User
Kernel
Malformed packet:
crashes virtual switch

Bears Risks!



VM
VM
VM
User
Kernel
Ctrl
User
Virtual
Switch
Kernel
VM
VM
VM
User
Virtual
Switch
Kernel
VM
VM
VM
User
Kernel

Conclusion

- ⇢ A vision: self-adjusting networks
- ⇢ Example 1: policy-compliant networks
  - ⇀ self-verifying
  - ⇀ self-repairing
- ⇢ Example 2: demand-aware topologies
- ⇢ On both fronts: tip of the iceberg!

⇀ E.g., self-adjusting networks further supported by telemetry (data) and AI (e.g., prediction)

Thank you!

References

AalWiNes: A Fast and Quantitative What-If Analysis Tool for MPLS Networks
Peter Gjøl Jensen, Morten Konggaard, Dan Kristiansen, Stefan Schmid, Bernhard Clemens Schrenk, and Jiri Srba.
16th ACM International Conference on emerging Networking EXperiments and Technologies (**CoNEXT**), Barcelona, Spain, December 2020.

On the Complexity of Traffic Traces and Implications
Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Boston, Massachusetts, USA, June 2020.

Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks (Editorial)
Chen Avin and Stefan Schmid.
ACM SIGCOMM Computer Communication Review (**CCR**), October 2018.

Cerberus: The Power of Choices in Datacenter Topology Design (A Throughput Perspective)
Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin.
ACM **SIGMETRICS** and ACM Performance Evaluation Review (**PER**), Mumbai, India, June 2022.

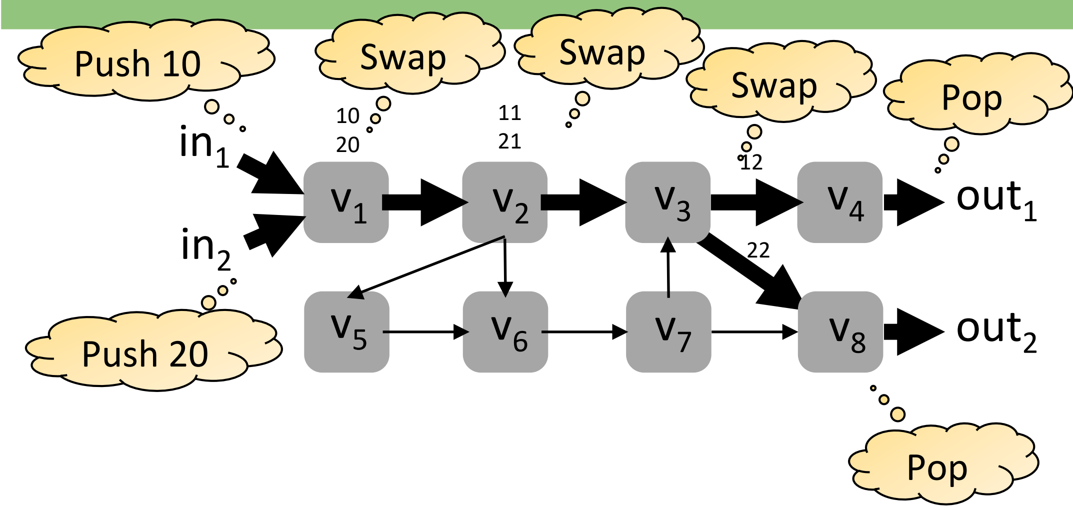
Latte: Improving the Latency of Transiently Consistent Network Update Schedules
Mark Glavind, Niels Christensen, Jiri Srba, and Stefan Schmid.
38th International Symposium on Computer Performance, Modeling, Measurements and Evaluation (**PERFORMANCE**) and ACM Performance Evaluation Review (**PER**), Milan, Italy, November 2020.

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.

Backup Slides

Case Study:

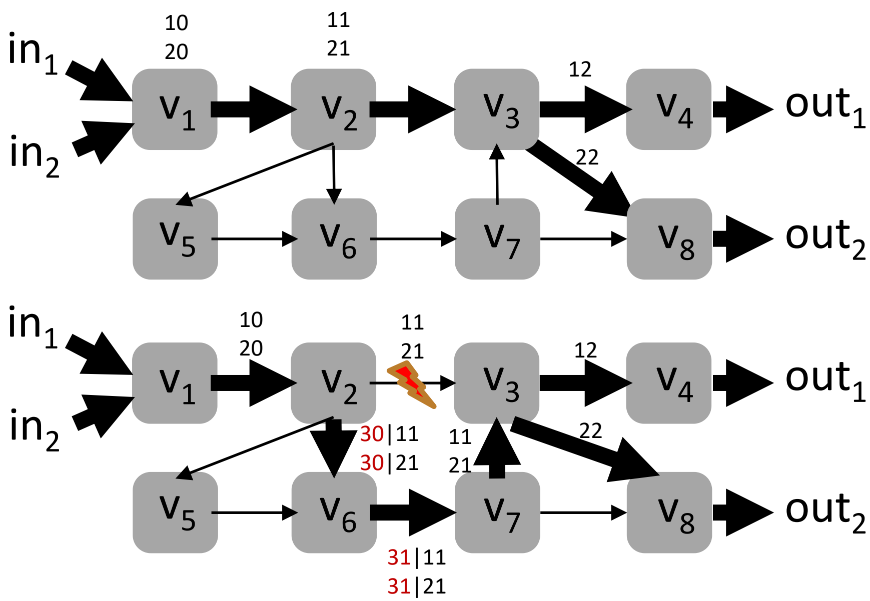
MPLS Networks



Original Routing

Case Study:

MPLS Networks

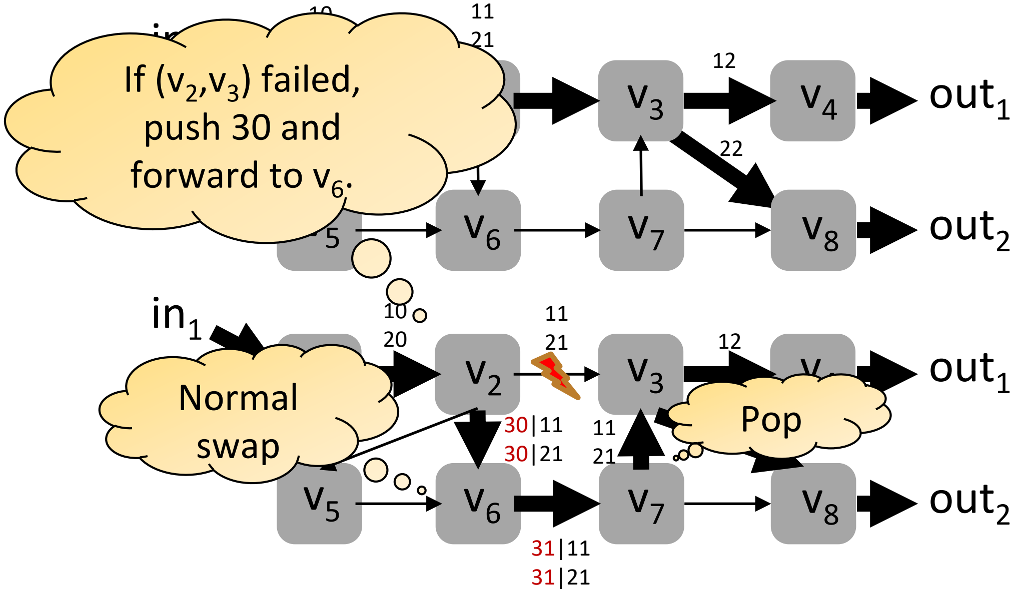


Original Routing

One failure: push 30: route around (v_2, v_3)

Case Study:

MPLS Networks

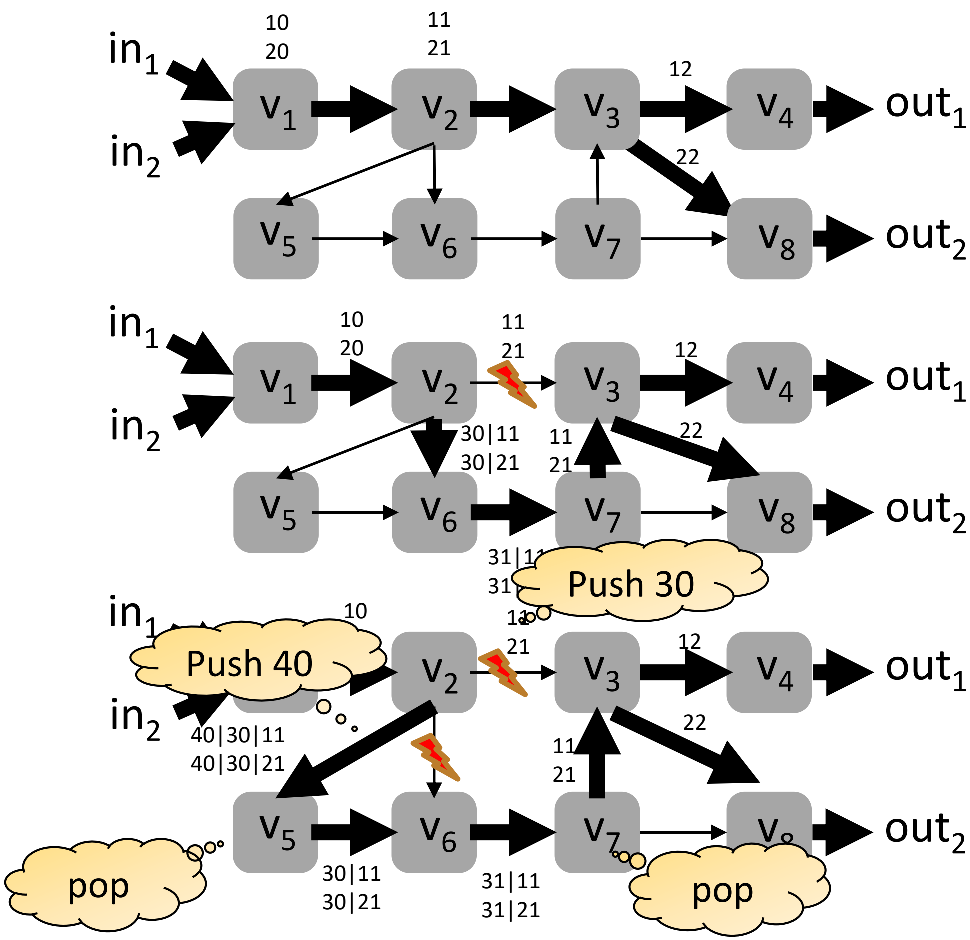


Original Routing

One failure: push 30: route around (v_2,v_3)

Case Study:

MPLS Networks



Original Routing

One failure: push 30: route around (v_2,v_3)

Two failures: first push 30: route around (v_2,v_3)

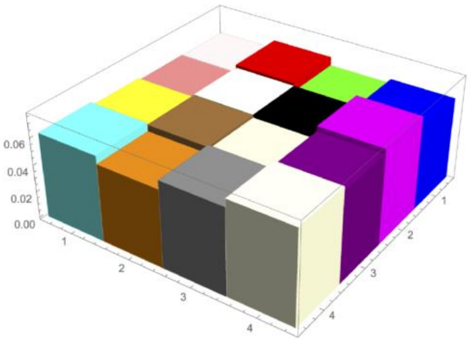
Push recursively 40: route around (v_2,v_6)

Intuition

Which demand has more structure?

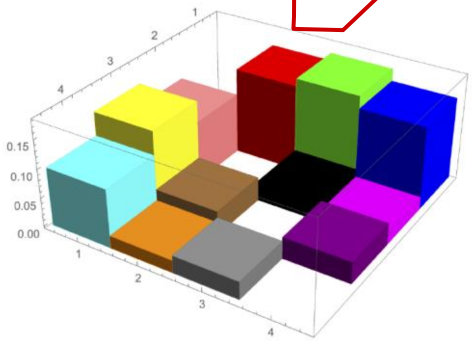
⇢ Traffic matrices of two different distributed ML applications

→ GPU-to-GPU



VS

Color = communication pair



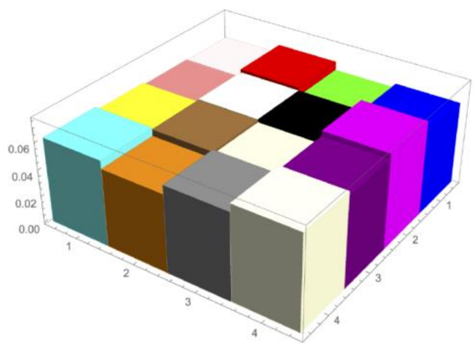
Intuition

Which demand has more structure?

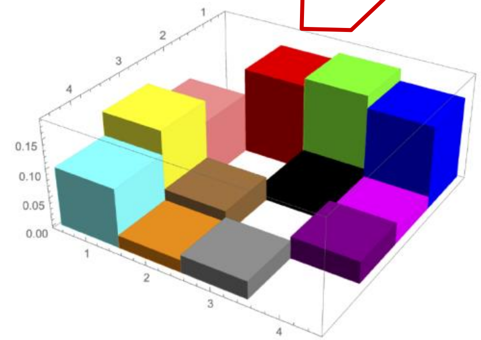
⇢ Traffic matrices of two different distributed ML applications

→ GPU-to-GPU

Color = communication pair



VS



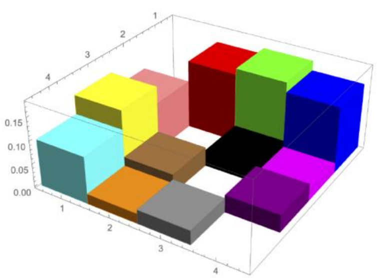
More uniform

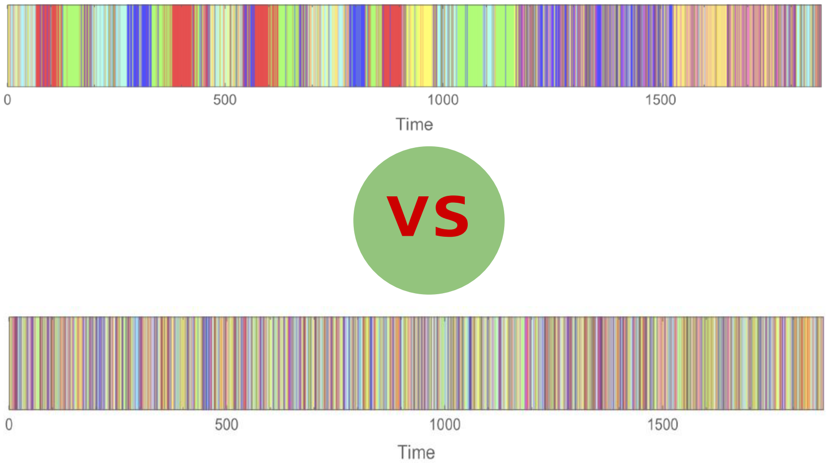
More structure

Intuition

Spatial vs temporal structure

- Two different ways to generate same traffic matrix:
  - Same non-temporal structure
- Which one has more structure?



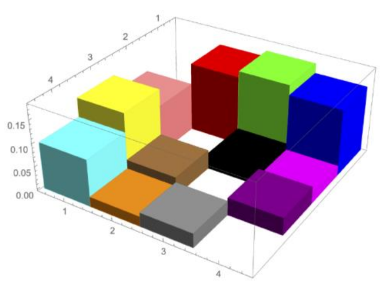


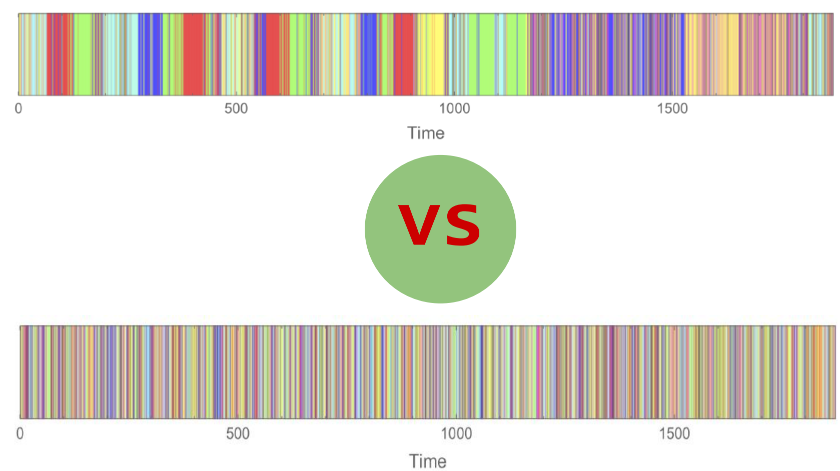
Intuition

Spatial vs temporal structure

⇢ Two different ways to generate same traffic matrix:
  ⇀ Same non-temporal structure

⇢ Which one has more structure?



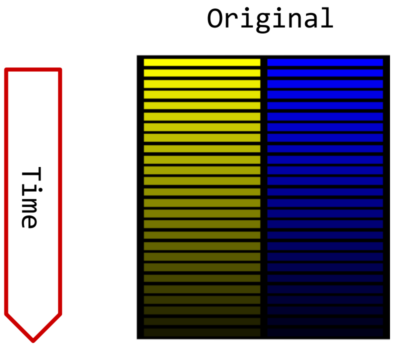


Systematically?

Trace Complexity

Information-Theoretic Approach

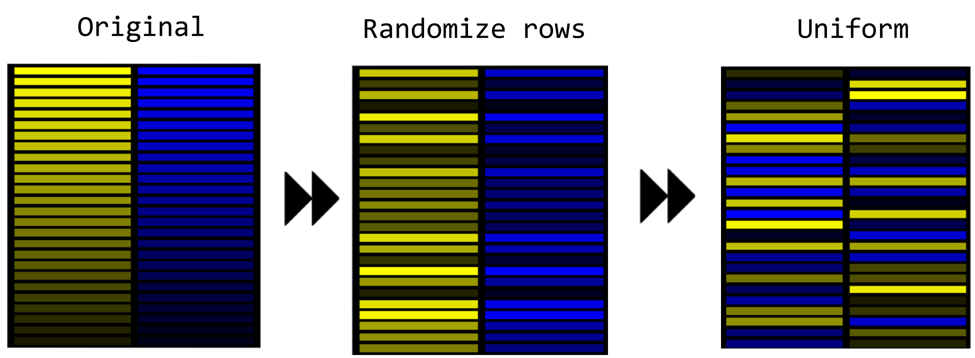
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

“Shuffle&Compress”



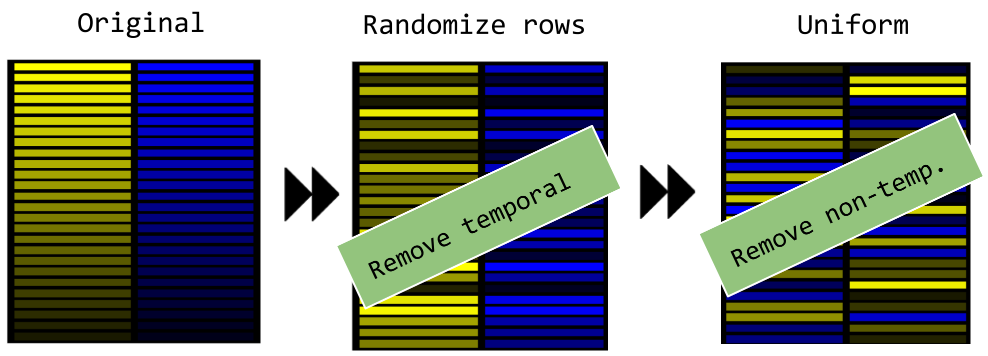
Increasing complexity (systematically randomized)

More structure (compresses better)

Trace Complexity

Information-Theoretic Approach

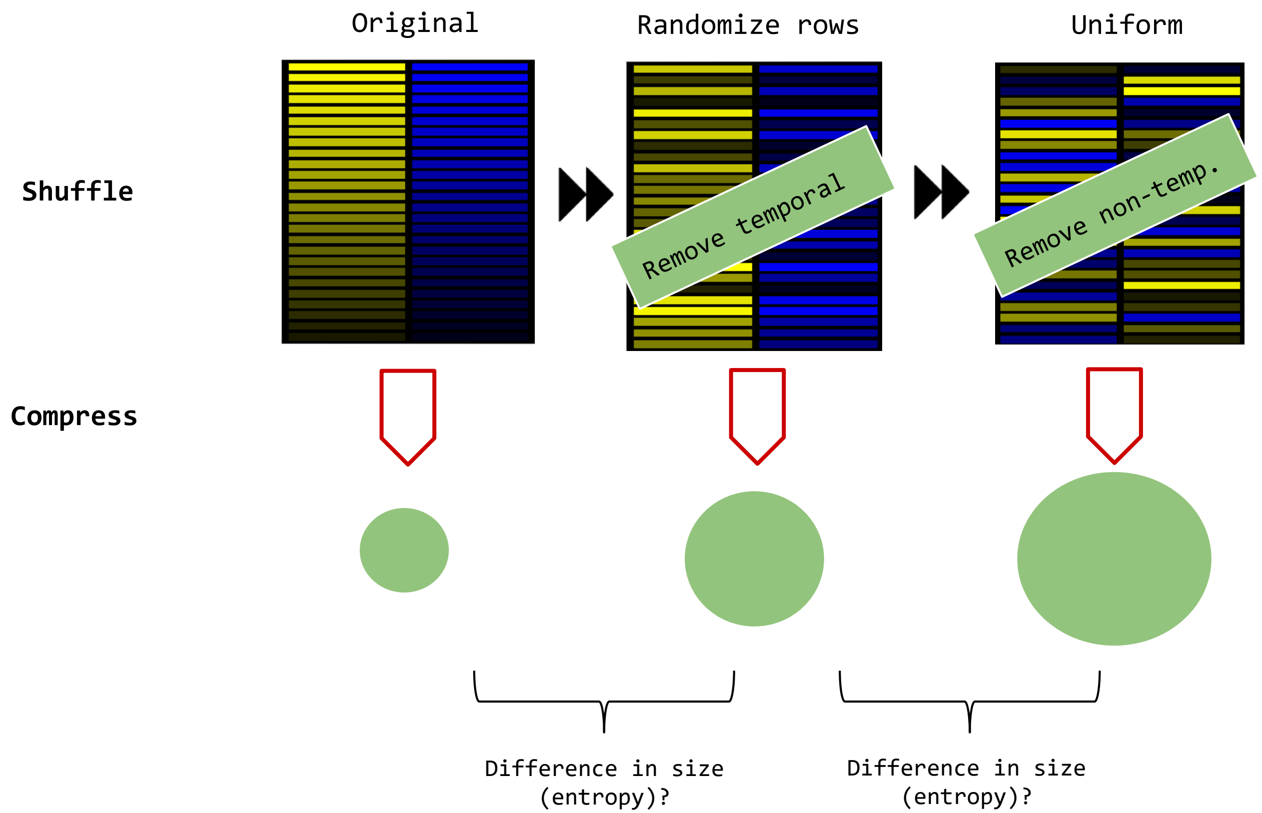
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

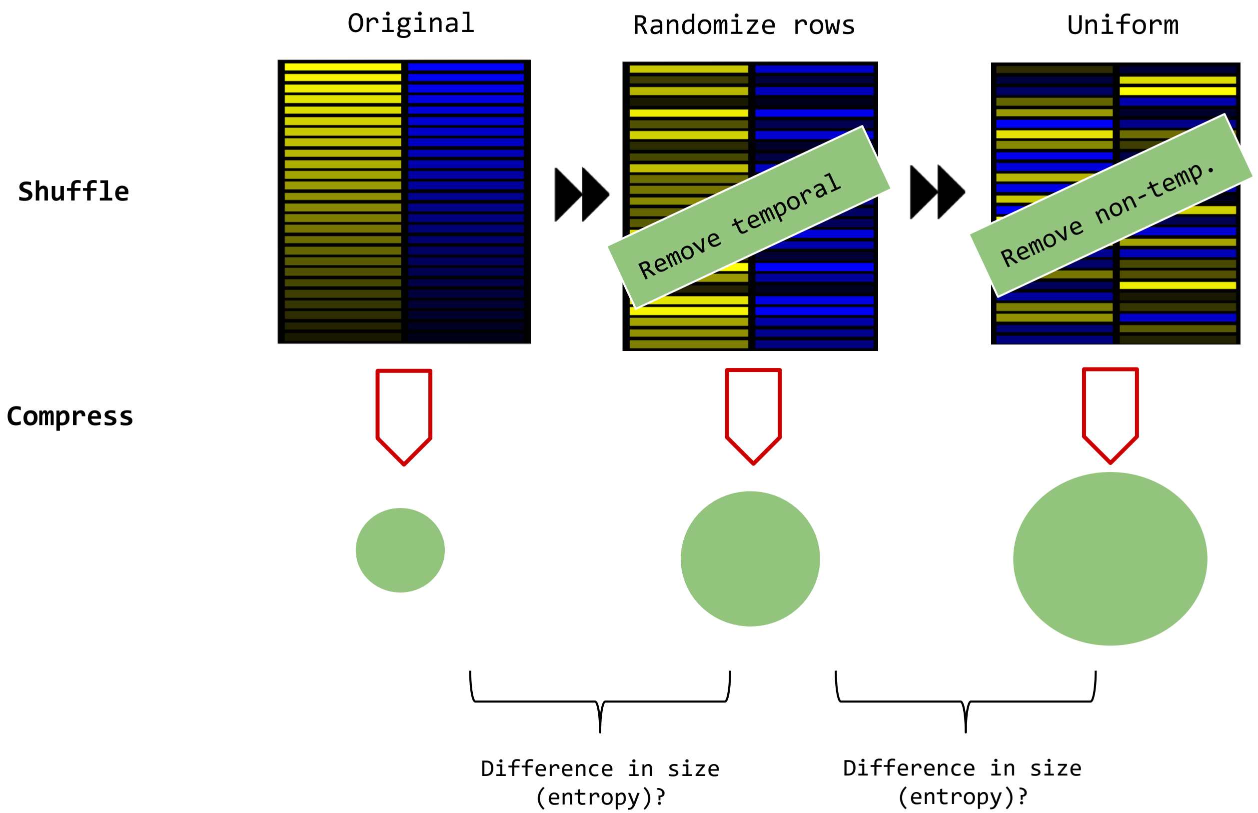
“Shuffle&Compress”



Trace Complexity

Information-Theoretic Approach

"Shuffle&Compress"

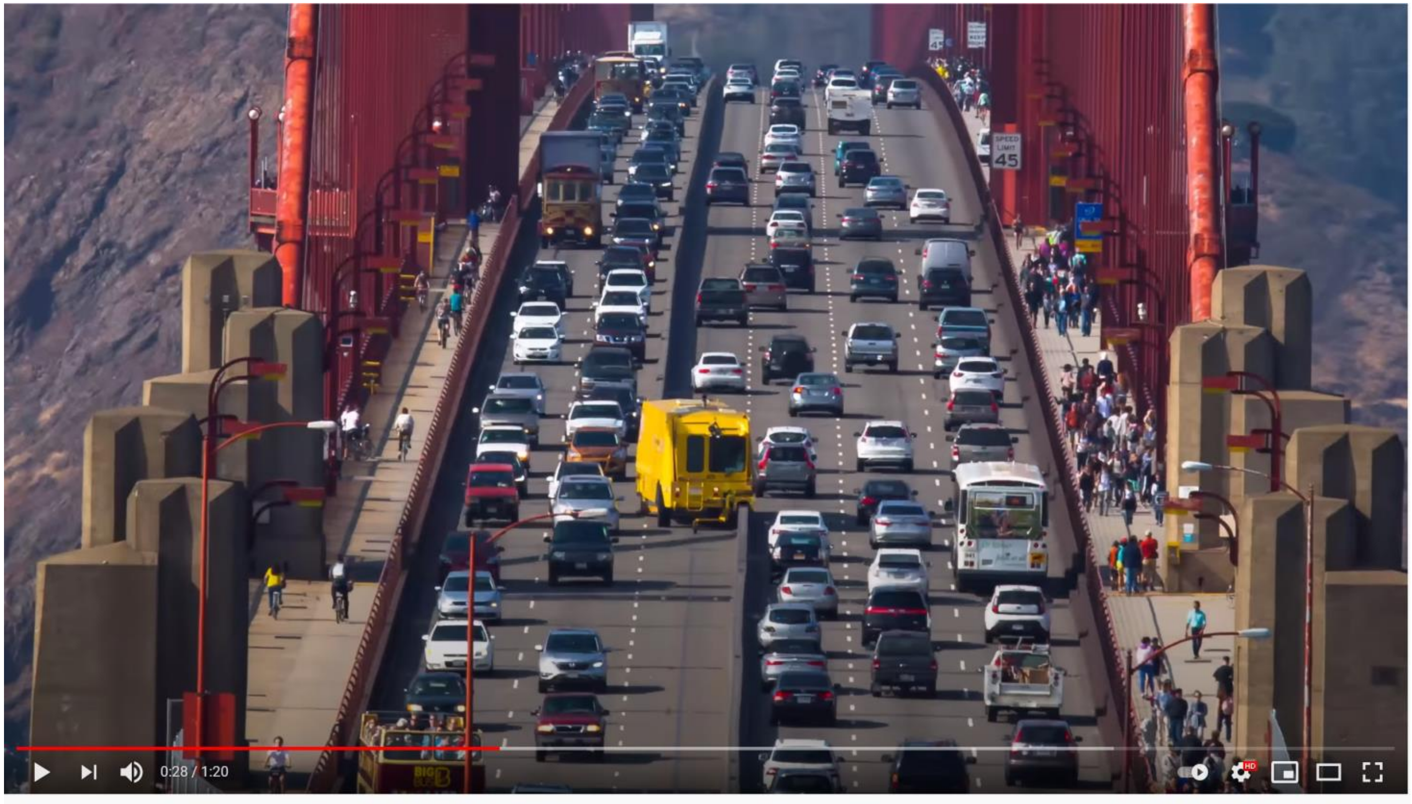


Can be used to define 2-dimensional complexity map!

Bonus Material

Hogwarts Stair

Bonus Material



Golden Gate Zipper

Bonus Material

IEEE SPECTRUM

Topics ▾ Reports ▾ Blogs ▾ Multimedia ▾ Magazine ▾ Resources ▾ Search ▾

Tech Talk | Computing | Hardware

07 May 2021 | 16:55 GMT

Reconfigurable Optical Networks Will Move Supercomputer Data 100X Faster

Newly designed HPC network cards and software that reshapes topologies on-the-fly will be key to success

By **Michelle Hampson**

Photo illustration: Shutterstock

In HPC