Self-driving/monitoring networks

in the age of deep network programmability

Slide 1

Self-driving/monitoring networks

in the age of deep network programmability

Laurent Vanbever

nsg.ee.ethz.ch

TMA

June 19 2019

Slide 2

1962

Self-driving/monitoring networks

in the age of deep network programmability

Slide 3

Paul Baran (1926–2011)

American electrical engineer

created the notion of a distributed network

which could maintain communications

in the face of a thermonuclear attack

inventor of packet switching
(with Donald Davies)

Slide 4

On distributed communication networks

Paul Baran—Sept. 1962

Introduces the concept of

- packet switching
- distributed routing
- reliable transport

among many others…

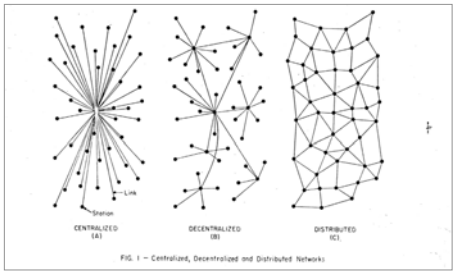


FIG. 1 – Centralized, Decentralized and Distributed Networks

Self-driving/monitoring networks

in the age of deep network programmability

Slide 5

Thanks to Paul Barlan et al.

the Internet infrastructure is extremely resilient

ability to route packets around failures
(not necessarily fast though)

Slide 6

Thanks to Paul Barlan et al.

the Internet infrastructure is extremely resilient

… at least when it comes to thermonuclear wars…

Self-driving/monitoring networks

in the age of deep network programmability

Slide 7

Thanks to Paul Barlan et al.

the Internet infrastructure is extremely resilient

… at least when it comes to thermonuclear wars…

but how does it fare against, say…

Slide 8

Thanks to Paul Barlan et al.

the Internet infrastructure is extremely resilient

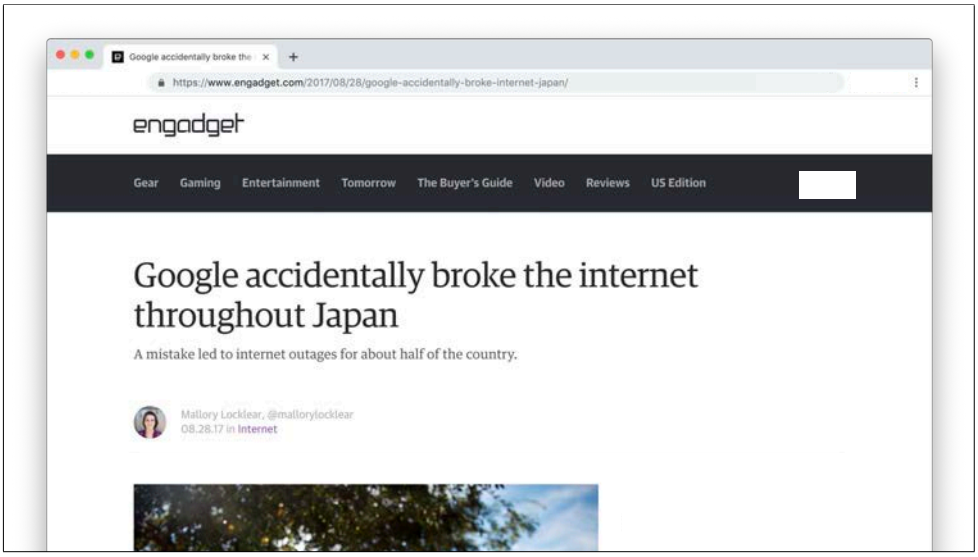
… at least when it comes to thermonuclear wars…

but how does it fare against, say… us humans?

Self-driving/monitoring networks

in the age of deep network programmability

Slide 9



Slide 10

27 August 2017

Google made a configuration mistake which caused their Chicago point-of-presence to wrongly advertise 160k IP prefixes to its neighbors.

Slide 11

Google made a configuration mistake which caused their Chicago point-of-presence to wrongly advertise 160k IP prefixes to its neighbors.

These advertisements propagated in the Internet and got picked up by Japanese giants (IIJ and KDDI) which started to direct local Japanese traffic to Google.

Slide 12

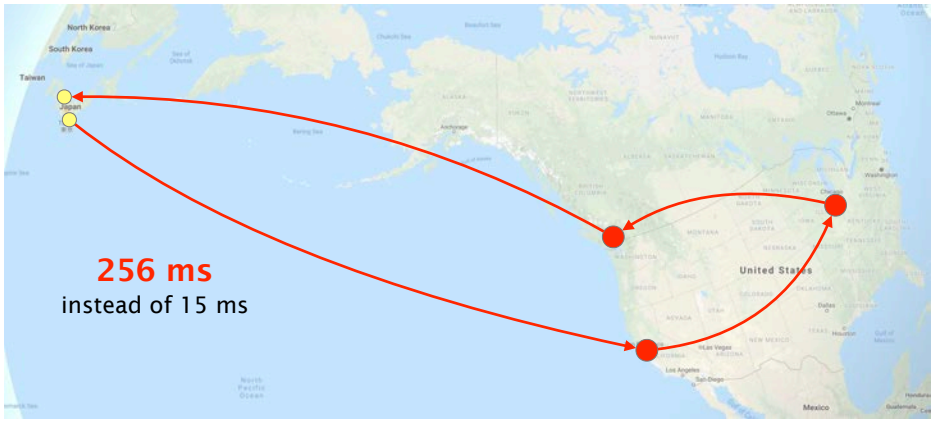
Tokyo to Nagoya
(normally)

Self-driving/monitoring networks

in the age of deep network programmability

Slide 13

Tokyo to Nagoya via... Chicago?!
A **17x** increased latency!



Slide 14

Google made a configuration mistake which caused their Chicago point-of-presence to wrongly advertise 160k IP prefixes to its neighbors.

These advertisements propagated in the Internet and got picked up by Japanese giants (NTT and KDDI) which started to direct local, Japanese traffic to Google.

The outage in Japan *only* lasted a couple of hours but was so severe that the country's ministries wanted carriers to report on what went wrong.

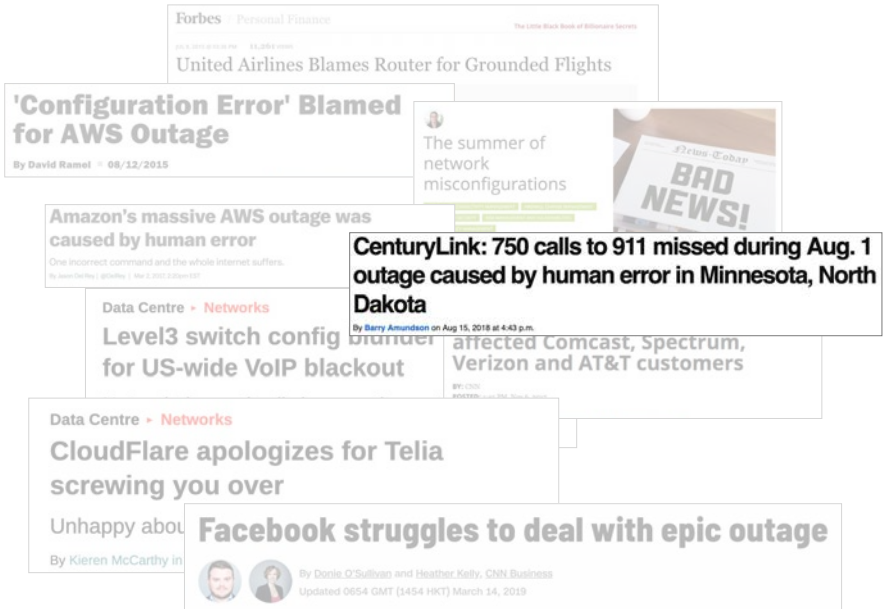
Self-driving/monitoring networks

in the age of deep network programmability

Slide 15

This is far from being an isolated event…

Slide 16



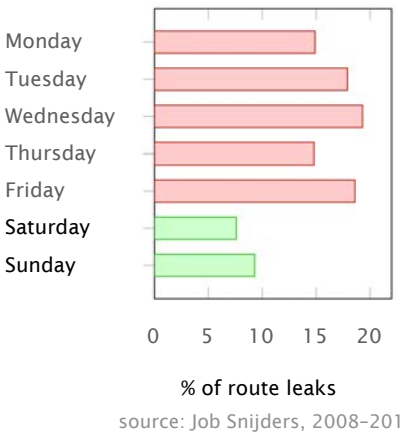
Slide 17

"Human factors are responsible
for 50% to 80% of network outages"

Juniper Networks, *What's Behind Network Downtime?*, 2008

Slide 18

A perhaps ironic consequence is that the
Internet works better during the week-ends



source: Job Snijders, 2008–2016

Slide 19

"Human factors are responsible
for 50% to 80% of network outages"

Juniper Networks, *What's Behind Network Downtime?*, 2008

Slide 20

"Human factors are responsible
for [redacted] of [redacted]"

Self-driving/monitoring networks

in the age of deep network programmability

Slide 21

"Human factors are responsible

for >90% of car accidents"

NHTSA, National Motor Vehicle Crash Causation Survey, February 2015

Slide 22

Enters…

Self-Driving Car

Self-driving/monitoring networks

in the age of deep network programmability

Slide 23

Enters…

Self-Driving Car

Wikipedia A vehicle that is capable of monitoring its environment and moving with little or no human input

Slide 24

Enters…

Self-Driving Car

Wikipedia A vehicle that is capable of monitoring its environment and moving with little or no human input

Promise A drastic reduction in the number of (fatal) accidents

Self-driving/monitoring networks

in the age of deep network programmability

Slide 25

Enters…

Self-Driving Car

Slide 26

Enters…

Self-Driving

Self-driving/monitoring networks

in the age of deep network programmability

Slide 27

Enters…

Self-Driving Network

Slide 28

Enters…

Self-Driving Network

Definition

A network that is capable of monitoring its environment and adapting its behavior accordingly with little or no human input

Self-driving/monitoring networks

in the age of deep network programmability

Slide 29

Enters…

Self-Driving Network

Definition

A network that is capable of
monitoring its environment and
adapting its behavior accordingly
with little or no human input

Questions

How do we build and deploy such networks?

Slide 30

Enters…

Self-Driving Network

Definition

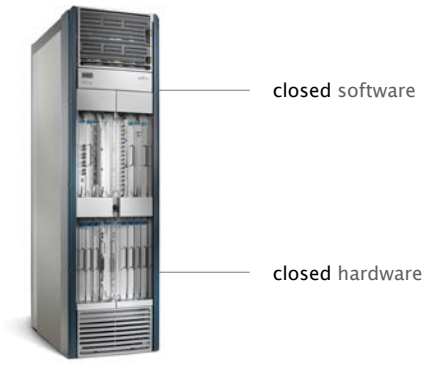
A network that is capable of
monitoring its environment and
adapting its behavior accordingly
with little or no human input

Questions

How do we build and deploy such networks?

Slide 31

Until recently, innovating in networks was hard because devices were completely locked down



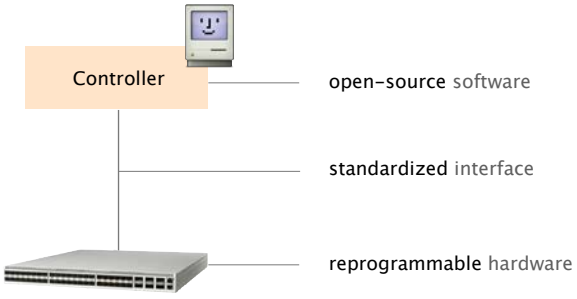
Cisco™ device

Slide 32

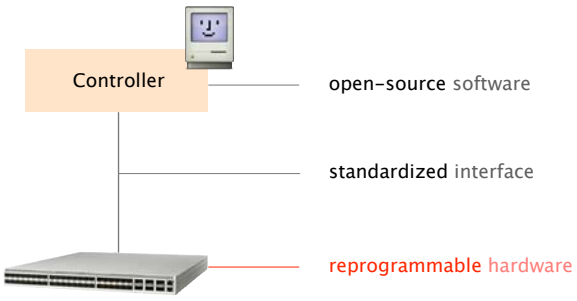
Things are changing though!

Slide 33

Networks are on the verge of a paradigm shift towards deep programmability



Slide 34



Self-driving/monitoring networks

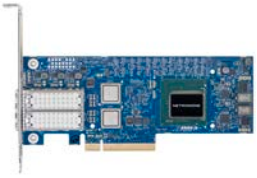
in the age of deep network programmability

Slide 35

Barefoot Tofino 12.8 Tbps

programmable network switch

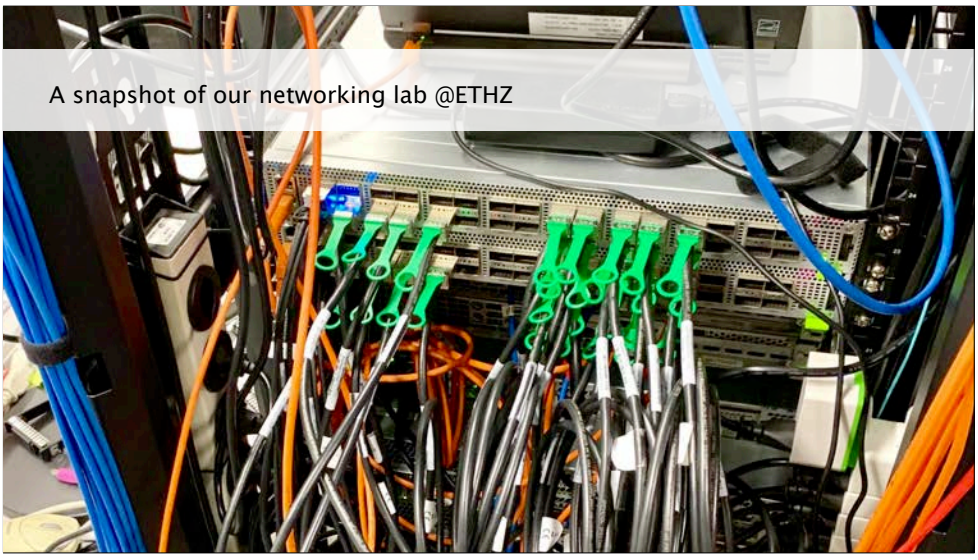
NetFPGA SUME



Netronome Agilio CX

100 Gbps programmable network card

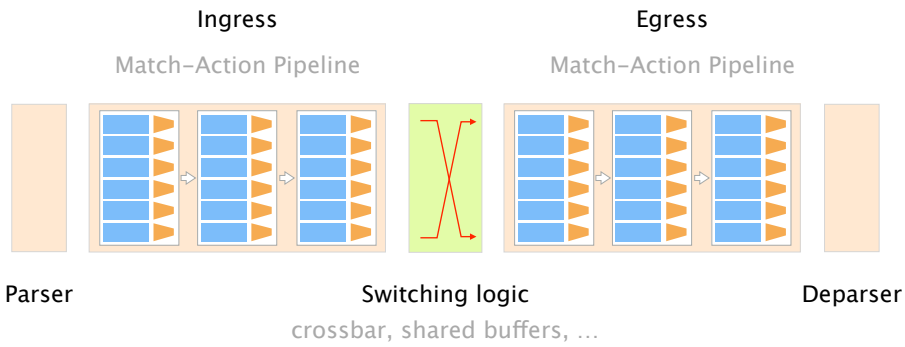
Slide 36



A snapshot of our networking lab @ETHZ

Slide 37

Reprogrammable network hardware allows to completely redefine the forwarding logic



Slide 38

Programmable networks can be made "self-driving"

Programmable devices can

- measure
- perform statistical inference
- adapt their forwarding decisions

at line rate, on a per-packet basis

Self-driving/monitoring networks

in the age of deep network programmability

Slide 39

Enters…

Self-Driving Network

Definition

A network that is capable of
monitoring its environment and
adapting its behavior accordingly
with little or no human input

Questions

How do we build and deploy such networks?

Slide 40

Levels of autonomy in self-driving cars

level 0	no automation	
1	driver assistance	human monitors the environment
2	partial automation	
3	conditional automation	system monitors human as fallback
4	high automation	
5	full automation	no more human

Self-driving/monitoring networks

in the age of deep network programmability

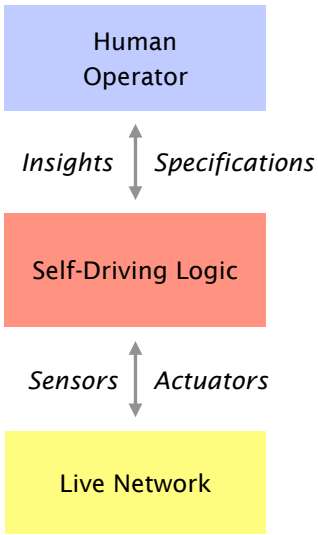
Slide 41

Self-driving/monitoring networks

in the age of deep network programmability

1	operator assistance	Part I assisting operators
2	partial automation	
3	conditional automation	Part II taking control
4	high automation	
5	full automation	too futuristic? ☺

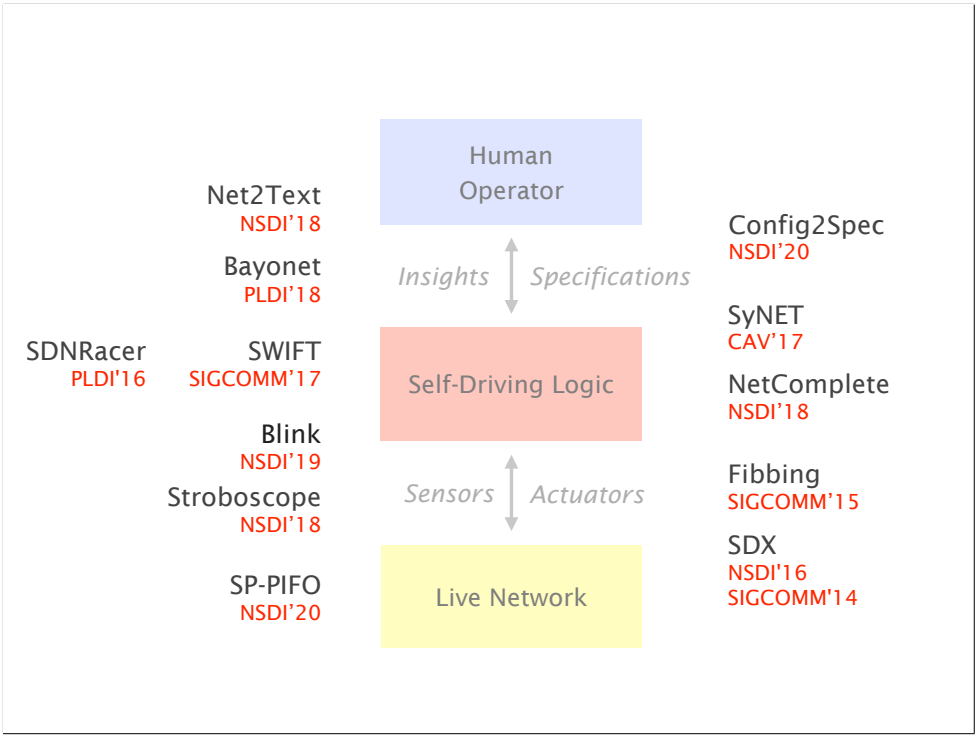
Slide 42



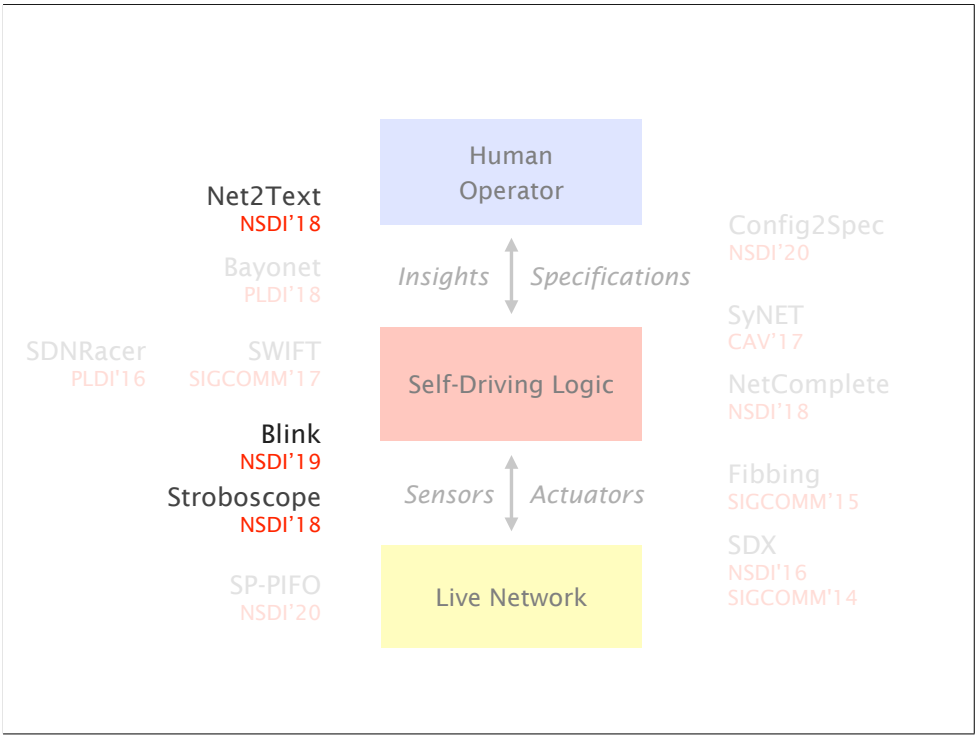
Self-driving/monitoring networks

in the age of deep network programmability

Slide 43



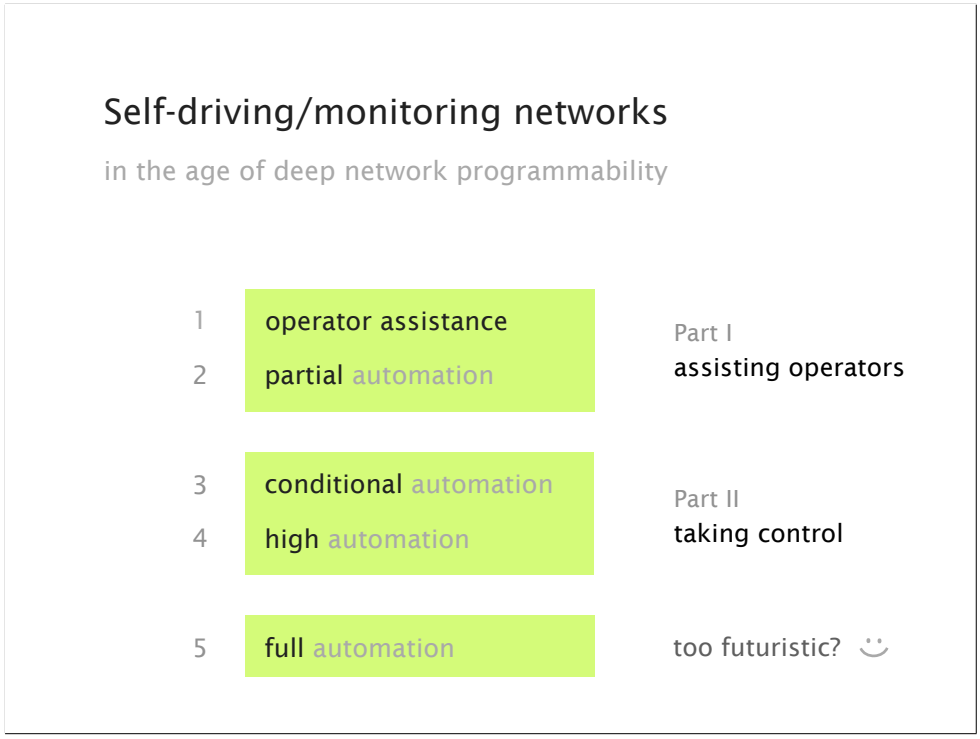
Slide 44



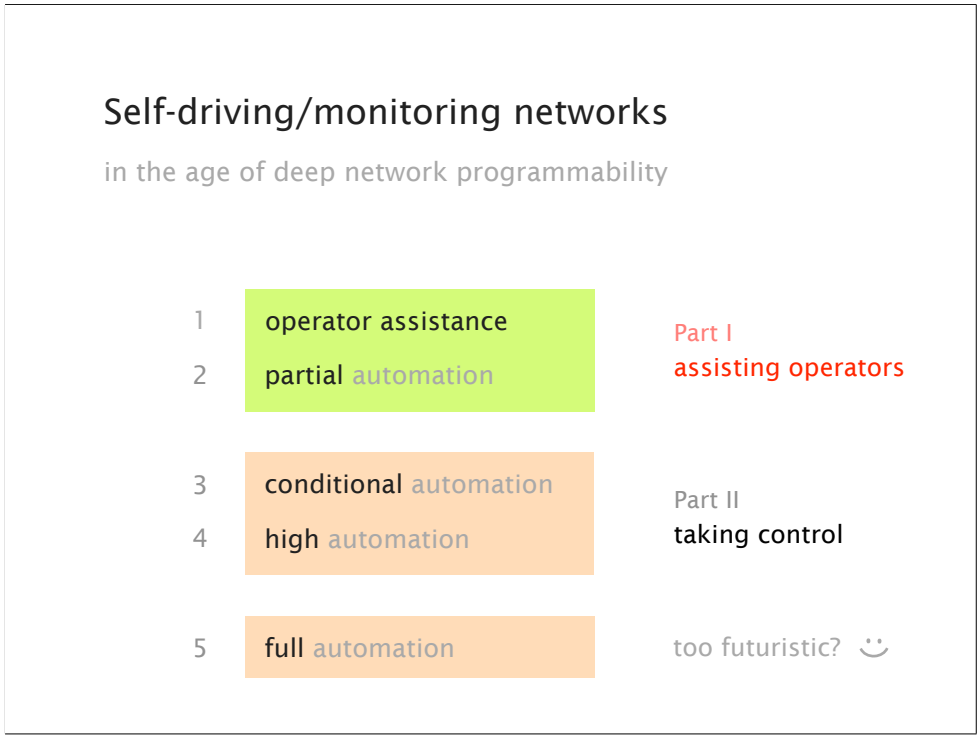
Self-driving/monitoring networks

in the age of deep network programmability

Slide 45



Slide 46



Self-driving/monitoring networks

in the age of deep network programmability

Slide 47

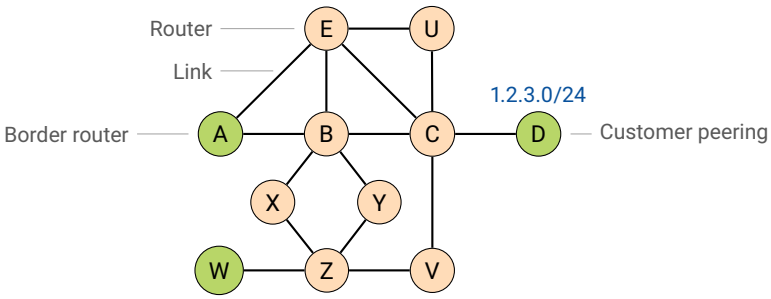
Stroboscope: Declarative Network Monitoring on a Budget

Olivier Tilmans Tobias Bühler Ingmar Poese Stefano Vissicchio Laurent Vanbever

USENIX Symposium on Networked Systems Design and Implementation. April 2018.

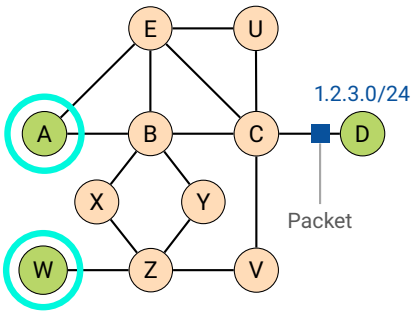
Slide 48

Consider this example ISP network topology



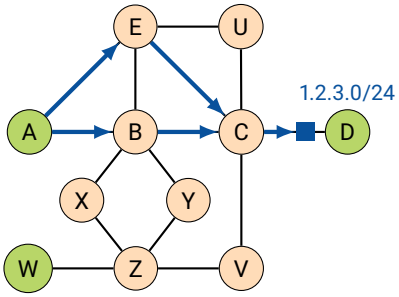
Slide 49

What is the ingress router for this packet arriving at router D?



Slide 50

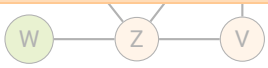
Which paths does the traffic follow?



Slide 51

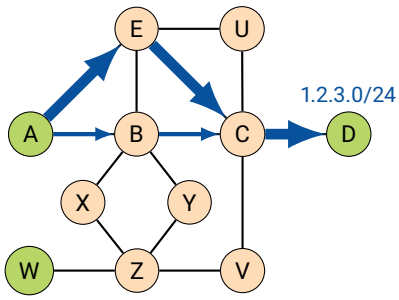
Which paths does the traffic follow?

Tracking flows network-wide requires to
match measurements across multiple vantage points

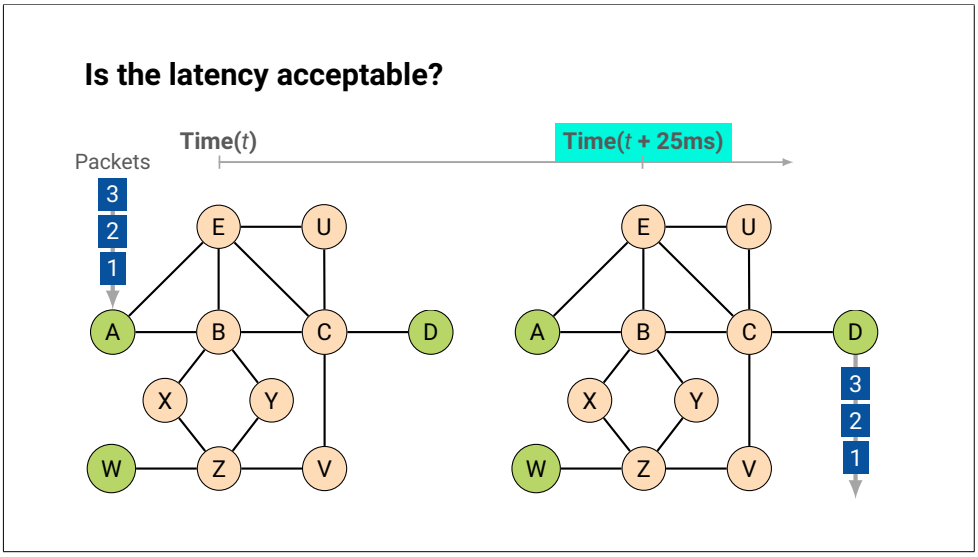


Slide 52

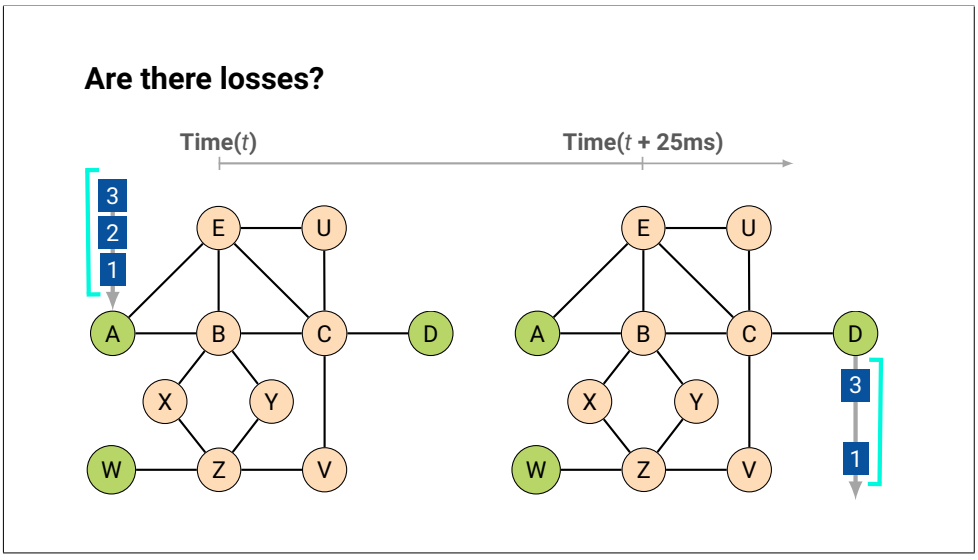
Is traffic load-balanced as expected?



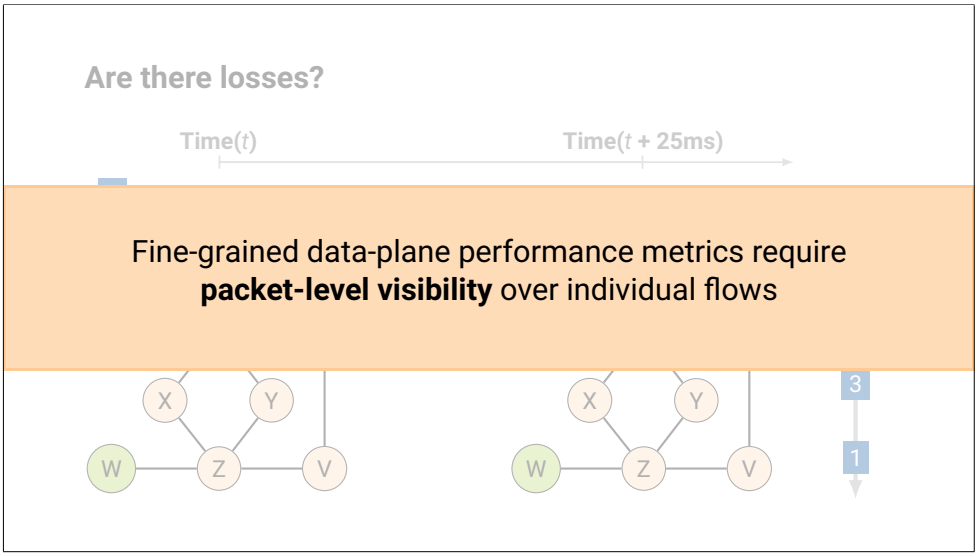
Slide 53



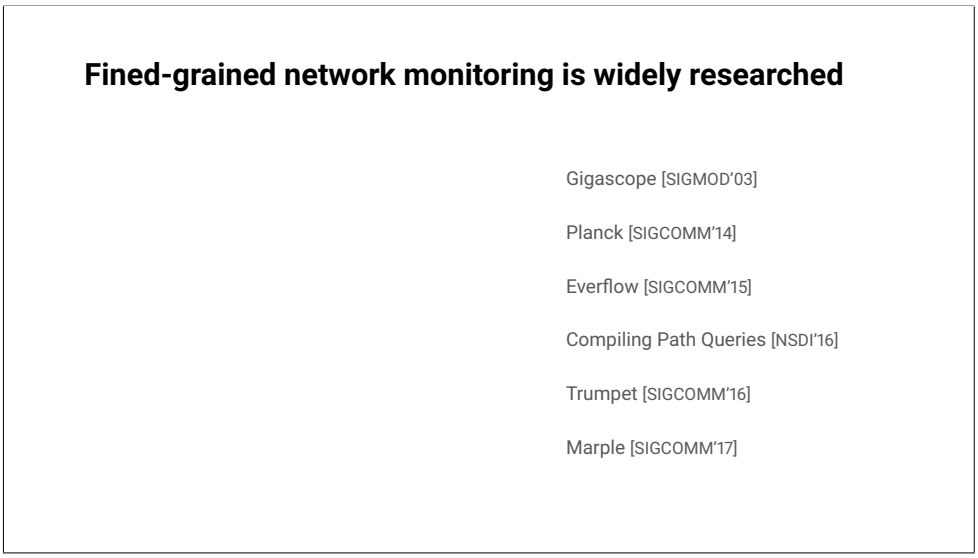
Slide 54



Slide 55



Slide 56



Slide 57

Fined-grained ISP network monitoring poses unique and unmet challenges

- No control over end hosts

Gigascope [SIGMOD'03]

Planck [SIGCOMM'14]

Everflow [SIGCOMM'15]

Compiling Path Queries [NSDI'16]

~~Trumpet [SIGCOMM'16]~~

Marple [SIGCOMM'17]

Slide 58

Fined-grained ISP network monitoring poses unique and unmet challenges

- No control over end hosts
- Limited data-plane flexibility

Gigascope [SIGMOD'03]

Planck [SIGCOMM'14]

Everflow [SIGCOMM'15]

~~Compiling Path Queries [NSDI'16]~~

~~Trumpet [SIGCOMM'16]~~

~~Marple [SIGCOMM'17]~~

Slide 59

Fined-grained ISP network monitoring poses unique and unmet challenges

- No control over end hosts
- Limited data-plane flexibility
- Limited monitoring bandwidth

~~Gigascope [SIGMOD'03]~~
~~Planck [SIGCOMM'14]~~
~~Everflow [SIGCOMM'15]~~
~~Compiling Path Queries [NSDI'16]~~
~~Trumpet [SIGCOMM'16]~~
~~Marple [SIGCOMM'17]~~

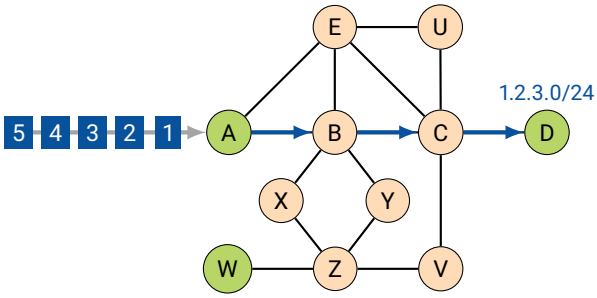
Slide 60

Stroboscope: Declarative Network Monitoring on a Budget

- Collecting traffic slices to monitor networks
- Adhering to a monitoring budget
- Using Stroboscope today

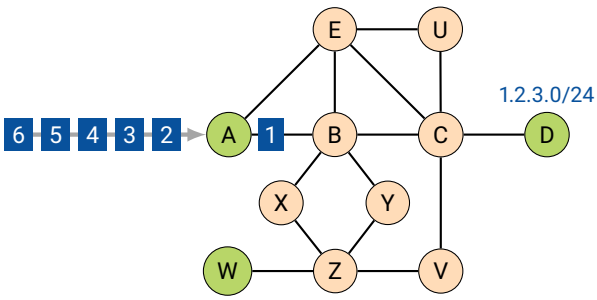
Slide 61

Consider the following flow of packets



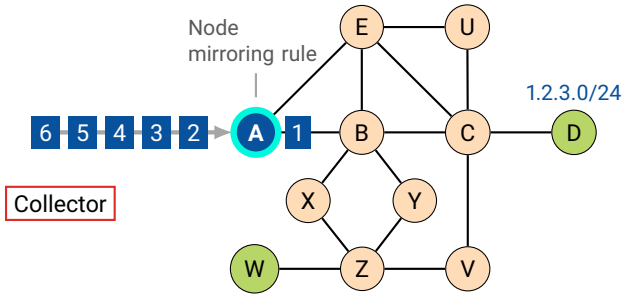
Slide 62

Consider the following flow of packets



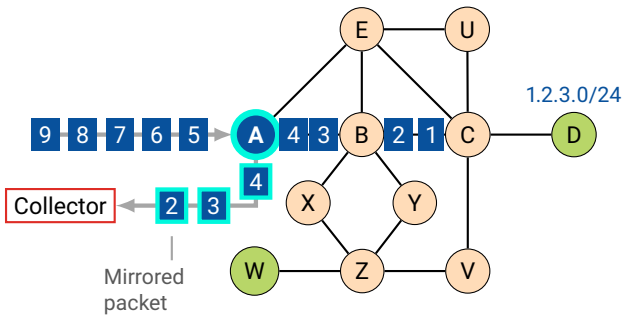
Slide 63

Stroboscope activates mirroring for the flow



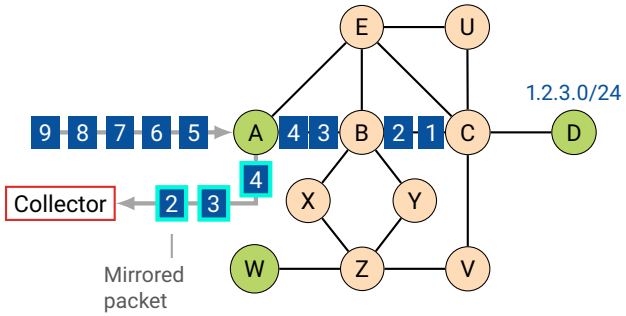
Slide 64

Packets are copied and encapsulated towards the collector



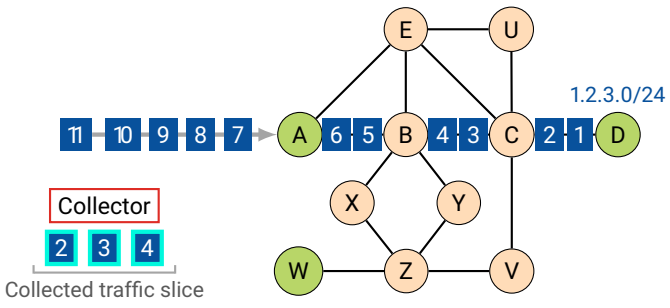
Slide 65

The mirroring rule is deactivated after a preset delay



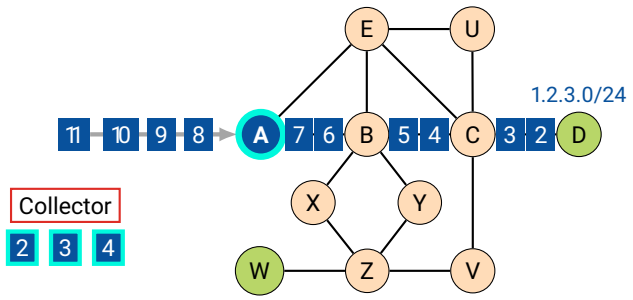
Slide 66

Stroboscope stores the traffic slice for analysis



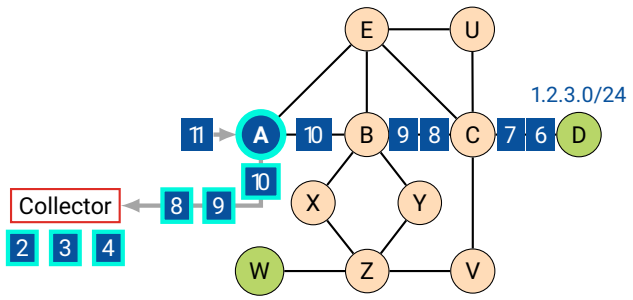
Slide 67

Stroboscope periodically toggles the mirroring rule



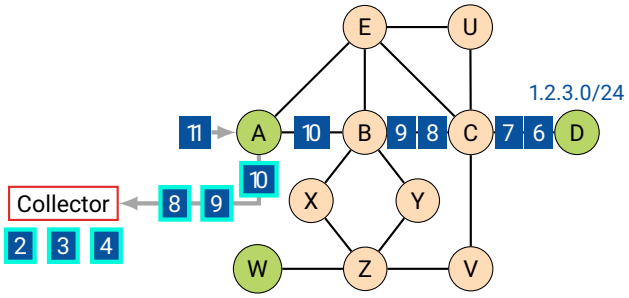
Slide 68

Stroboscope periodically toggles the mirroring rule



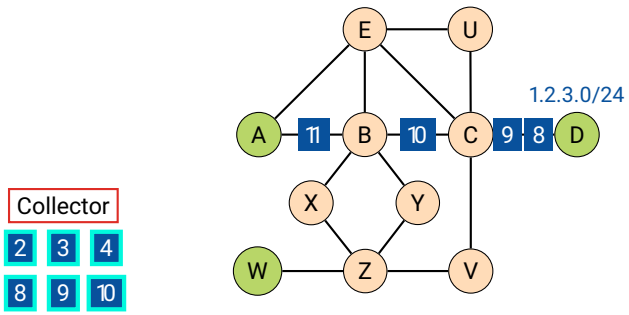
Slide 69

Stroboscope periodically toggles the mirroring rule



Slide 70

Stroboscope collects multiples traffic slices over time



Self-driving/monitoring networks

in the age of deep network programmability

Slide 71

Analyzing matching packets across traffic slices
enables fine-grained measurements at scale

Slide 72

Analyzing matching packets across traffic slices
enables fine-grained measurements at scale

Forwarding paths discovery, timestamp reconstruction, payload inspection, . . .

Slide 73

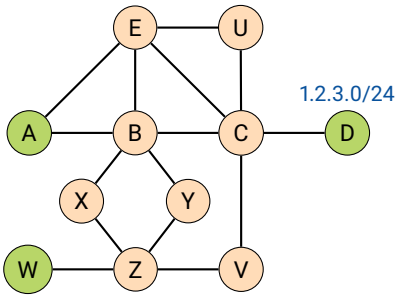
Stroboscope defines a declarative requirement language

```
MIRROR  1.2.3.0/24 ON [A B C D], [A E C D]
MIRROR  1.2.3.0/24 ON [A -> D]

CONFINE 1.2.3.0/24 ON [A B E C D]
CONFINE 1.2.3.0/24 [A -> D]

MIRROR  1.2.3.0/24 ON [ -> D]
CONFINE 1.2.3.0/24 ON [ -> D]

USING 15 Mbps DURING 500 ms EVERY 5 s
```



Slide 74

Stroboscope defines two types of queries

MIRROR

CONFINE

Slide 75

Stroboscope defines two types of queries

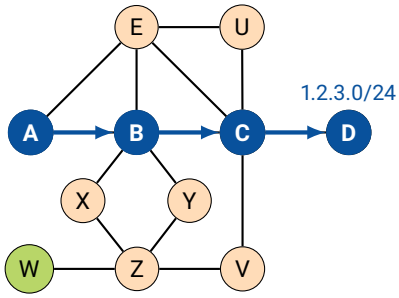
MIRROR

CONFINE

Slide 76

MIRROR queries reconstruct the path taken by packets

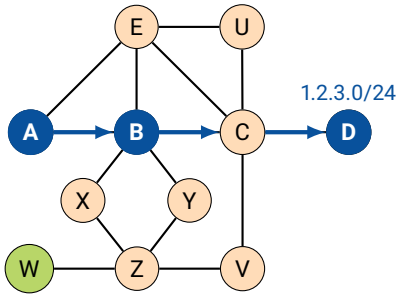
`MIRROR` 1.2.3.0/24 `ON` [A B C D]



Slide 77

Fewer mirroring rules reduces bandwidth usage

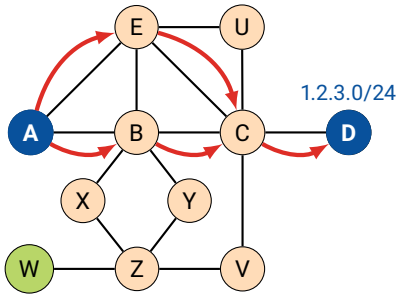
`MIRROR` 1.2.3.0/24 `ON` [A B C D]



Slide 78

Too few mirroring rules creates ambiguity

`MIRROR` 1.2.3.0/24 `ON` [A B C D]

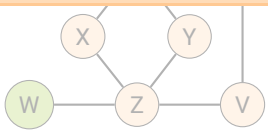


Slide 79

Too few mirroring rules creates ambiguity

MIRROR 1.2.3.0/24 ON [A B C D]

The **Key-Points Sampling** algorithm minimizes mirroring rules and guarantees non-ambiguous reconstructed paths



Slide 80

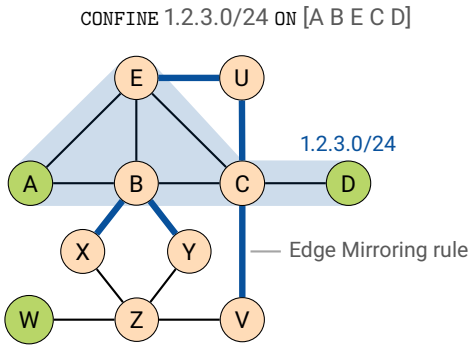
Stroboscope defines two types of queries

MIRROR

CONFINE

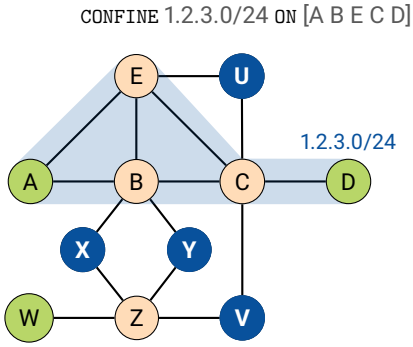
Slide 81

CONFINE queries mirror packets leaving a confinement region



Slide 82

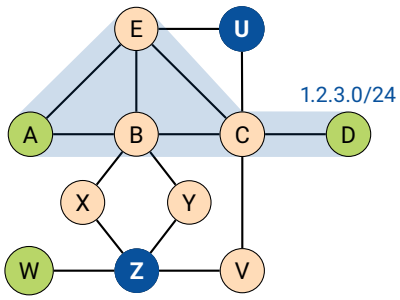
Fewer mirroring rules minimizes control-plane overhead



Slide 83

The lower bound is a multi-terminal node cut

`CONFINE` 1.2.3.0/24 `ON` [A B E C D]

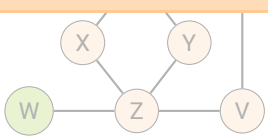


Slide 84

The lower bound is a multi-terminal node cut

`CONFINE` 1.2.3.0/24 `ON` [A B E C D]

The **Surrounding** algorithm minimizes mirroring rules and guarantees to mirror any packet leaving the confinement region



Slide 85

Stroboscope: Declarative Network Monitoring on a Budget

- Collecting traffic slices to monitor networks
- Adhering to a monitoring budget
- Using Stroboscope today

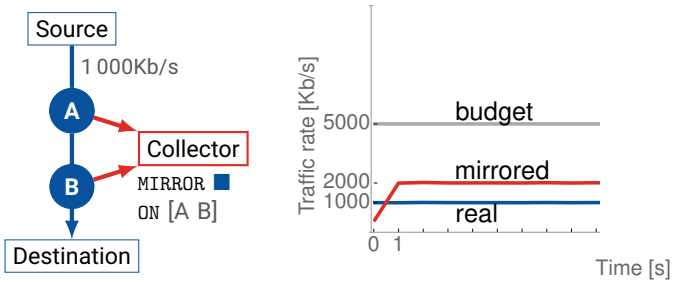
Slide 86

Stroboscope works with currently deployed routers

- Most vendors provide traffic mirroring and encapsulation primitives
- The collector activates mirroring for a flow by updating one ACL
- Routers autonomously deactivate mirroring rules using timers
- Traffic slices can be as small as **23 ms** on our routers (Cisco C7018)

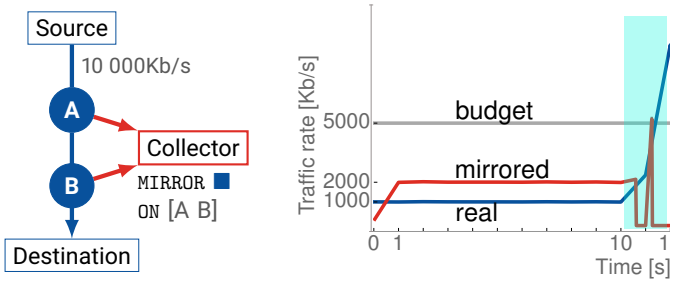
Slide 87

Stroboscope tracks the rate of mirrored traffic in real time



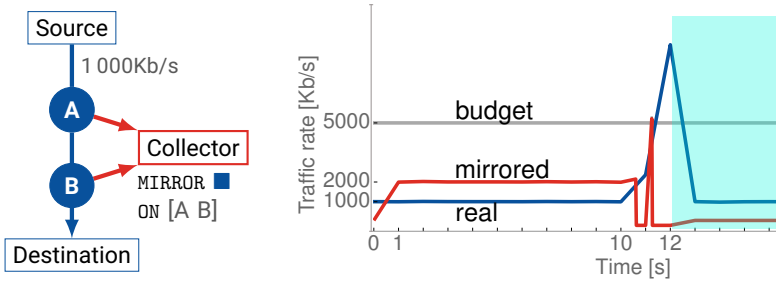
Slide 88

Measurement campaigns are stopped early if the estimated demand are exceeded



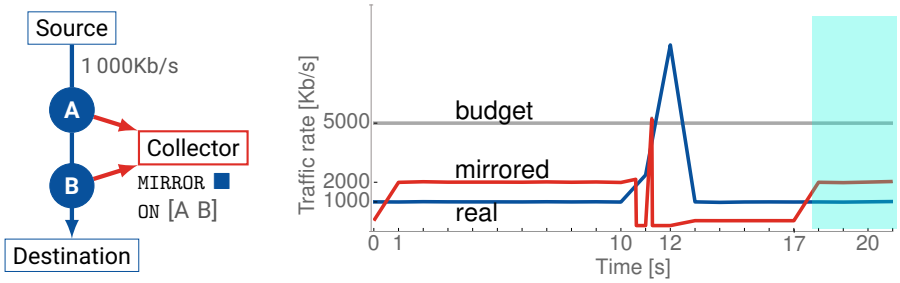
Slide 89

Exceeding the total budget schedules the query once per measurement campaign



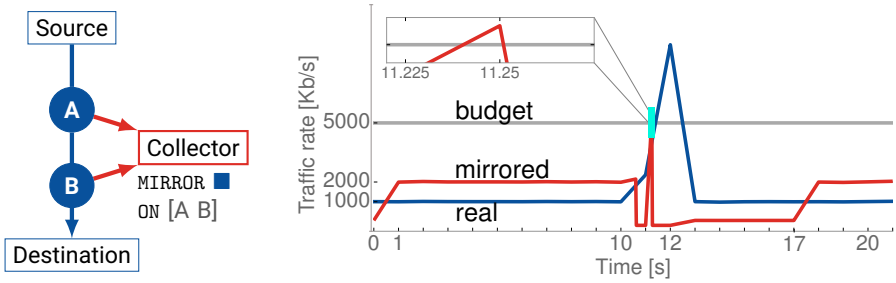
Slide 90

Stable recorded traffic rates are used for future estimations



Slide 91

Stroboscope exceeds the monitoring budget for at most one timeslot



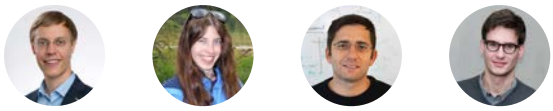
Slide 92

Stroboscope: Declarative Network Monitoring on a Budget

- Traffic slicing as a first-class data-plane primitive
- Strong guarantees on budget compliance and measurement accuracy
- Measurement analysis decoupled from measurement collection

Slide 93

Net2Text: Query-guided Network Captioning



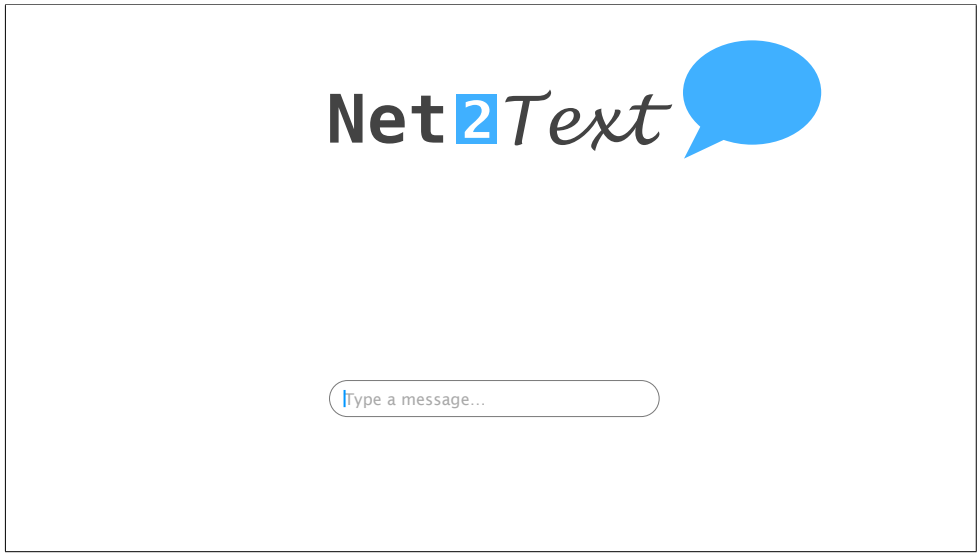
Rüdiger Birkner Dana Drachsler-Cohen Martin Vechev Laurent Vanbever

USENIX Symposium on Networked Systems Design and Implementation. April 2018.

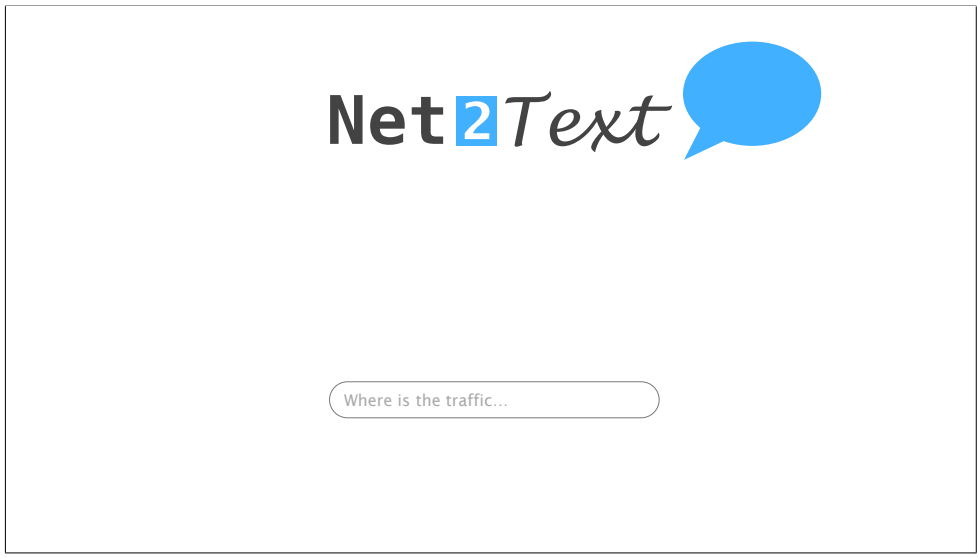
Slide 94

Net2Text

Slide 95



Slide 96



Slide 97

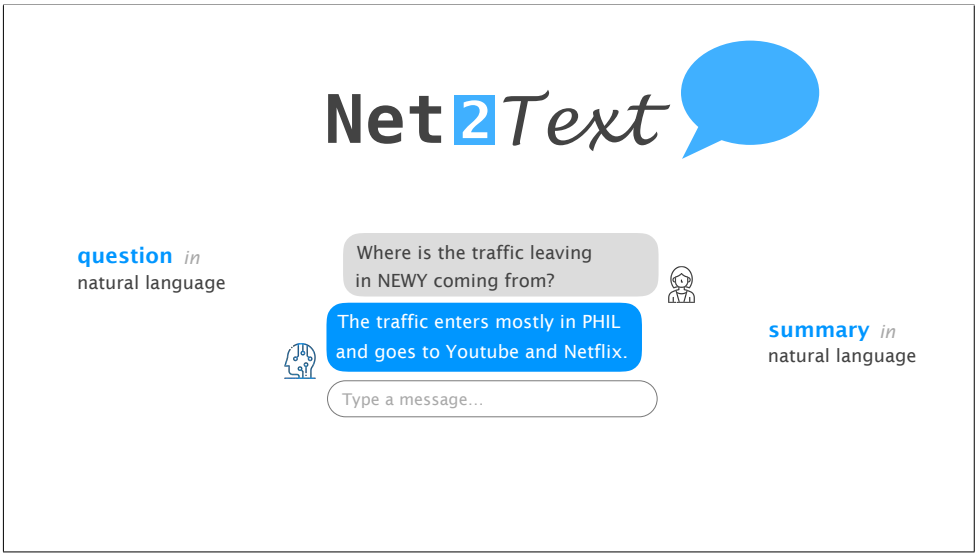
Slide 98

Self-driving/monitoring networks

in the age of deep network programmability

Slide 99

Slide 100



Slide 101

Finding *a* summary of the
network-wide forwarding state is simple

Slide 102

Finding *a* summary of the
network-wide forwarding state is simple

Traffic is forwarded.

Slide 103

Finding *a* summary of the
network-wide forwarding state is simple

Traffic from LOSA to 35.184.0.0/19,
which is owned by Google,
is leaving the network in CHIC
and takes the path
SUNV, DENV, KSCY, INDI to CHIC.

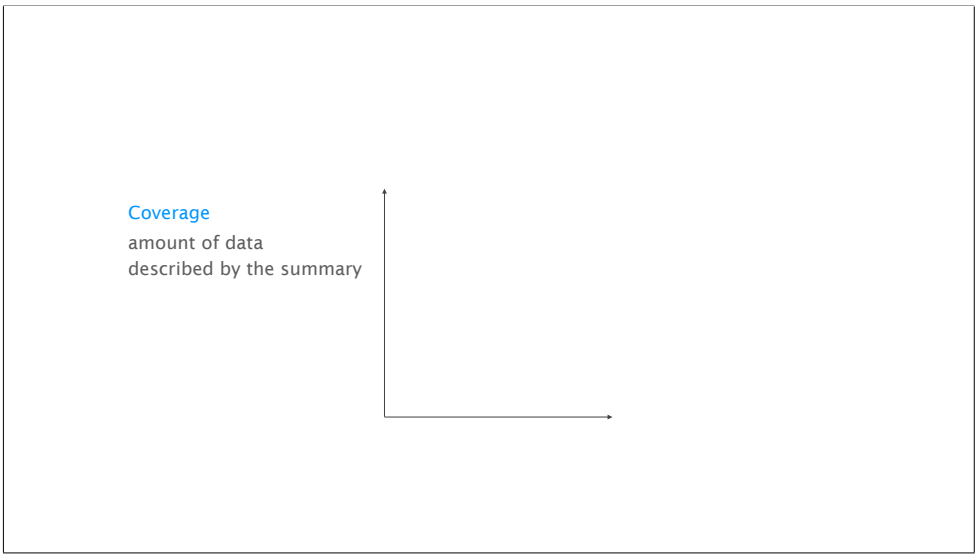
Slide 104

Finding a good summary of the
network-wide forwarding state is hard

Slide 105

Summarization is a multi-objective optimization problem

Slide 106

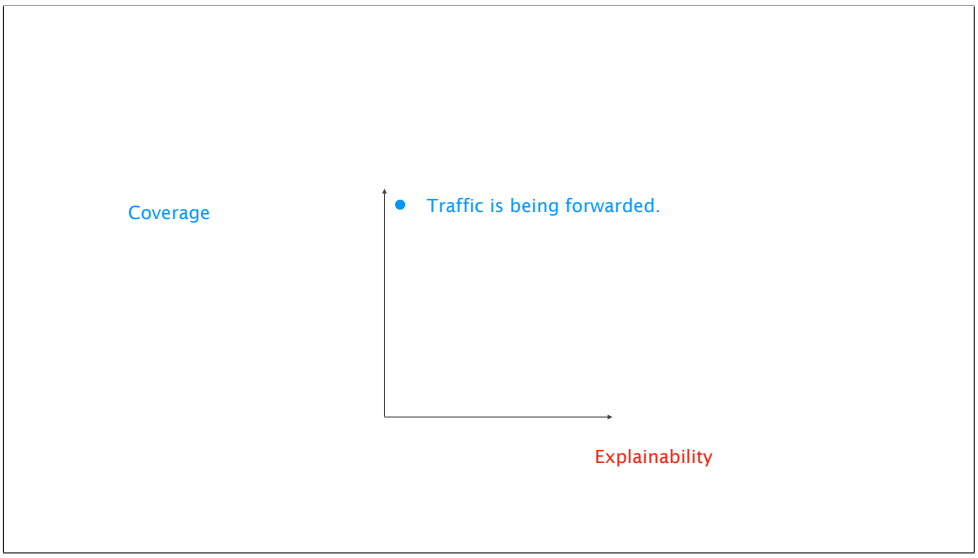


Self-driving/monitoring networks

in the age of deep network programmability

Slide 107

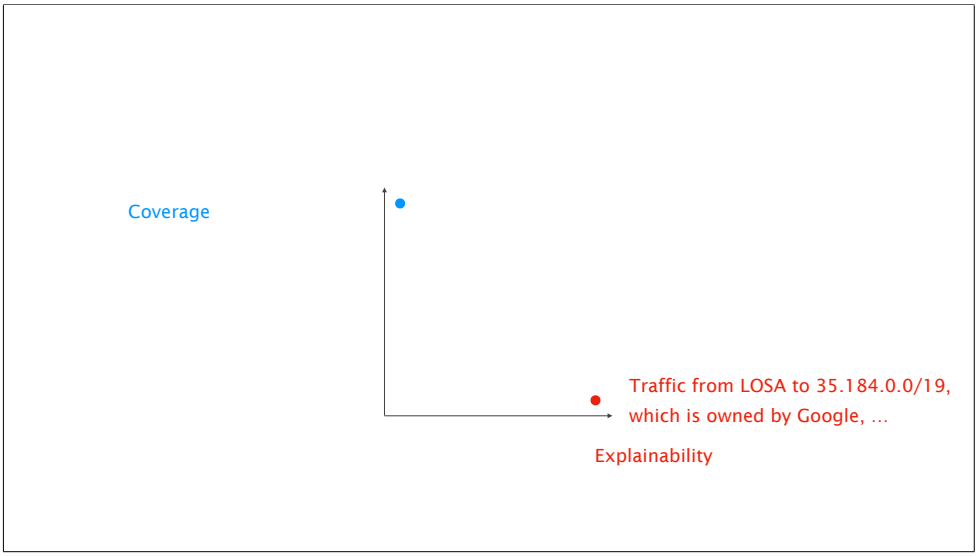
Slide 108



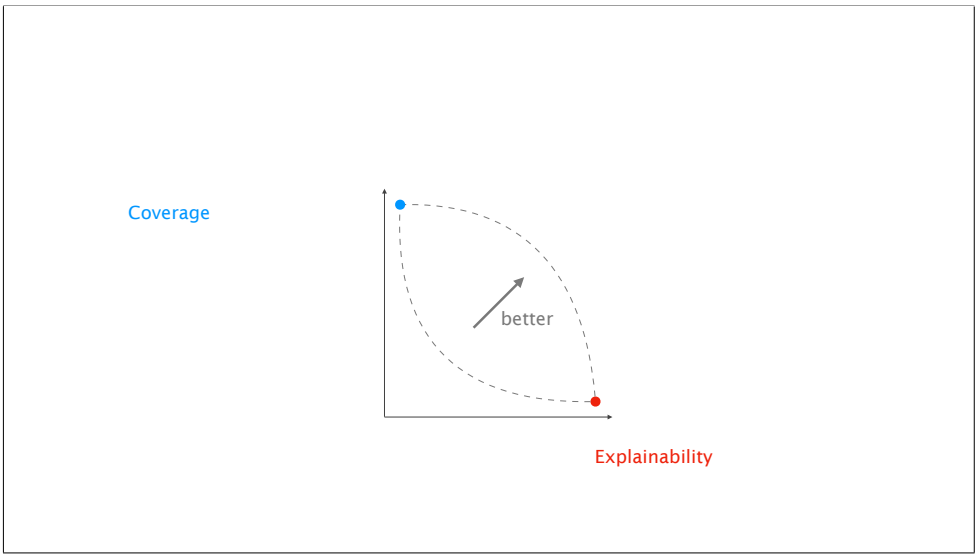
Self-driving/monitoring networks

in the age of deep network programmability

Slide 109



Slide 110



Slide 111

Summarization is a multi-objective optimization problem

Score Weighted sum of the amount of traffic covered by each path specification in the summary.

Slide 112

Summarization is a multi-objective optimization problem

Coverage

Score Weighted sum of the amount of traffic covered by each path specification in the summary.

Slide 113

Summarization is a multi-objective optimization problem

Explainability
weights based on level of detail
of the path specification

Score — Weighted sum of the amount of traffic covered by each path specification in the summary.

Slide 114

Summarization is a multi-objective optimization problem

Score — Weighted sum of the amount of traffic covered by each path specification in the summary.

Goal — Find path specifications that maximize the score.

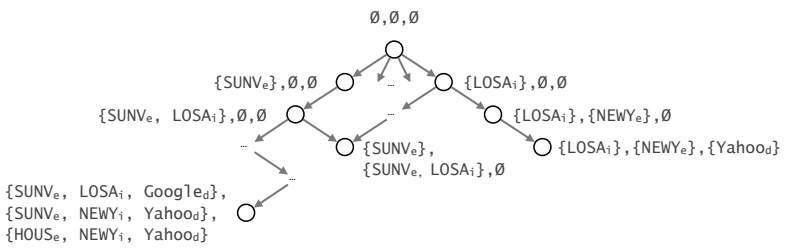
Slide 115

Summarization is a multi-objective optimization problem

Score	Weighted sum of the amount of traffic covered by each path specification in the summary.
Goal	Find *k* path specifications each of size at most *t* that maximize the score.

Slide 116

The search space is exponential
in the number of path specifications and feature values

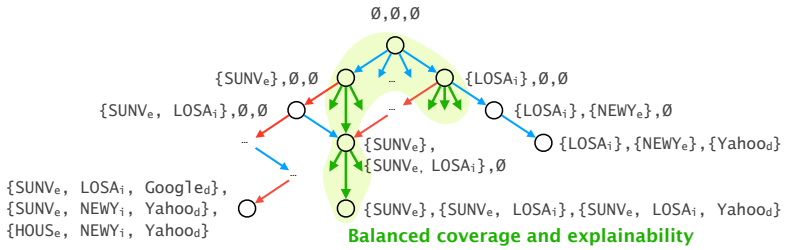


Self-driving/monitoring networks

in the age of deep network programmability

Slide 117

Net2Text reduces the search space to solutions
that balance coverage and explainability



Slide 118

Self-driving/monitoring networks

in the age of deep network programmability

1	operator assistance	Part I assisting operators
2	partial automation	
3	conditional automation	Part II taking control
4	high automation	
5	full automation	too futuristic? ☺

Slide 119

Part II: Taking control

applications	frameworks	methods
data-driven convergence	monitoring/inference	some of our challenges lying ahead

Slide 120

Part II: Taking control

applications	frameworks	methods
data-driven convergence		

Self-driving/monitoring networks

in the age of deep network programmability

Slide 121

Blink

Fast Connectivity Recovery Entirely in the Data Plane


Thomas Holterbach


Edgar Costa Molero


Maria Apostolaki


Alberto Dainotti

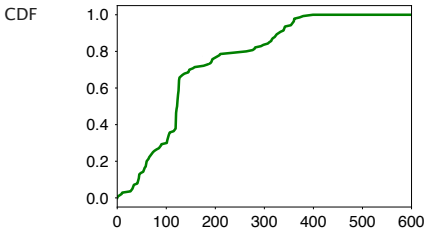

Stefano Vissicchio


Laurent Vanbever

USENIX Symposium on Networked Systems Design and Implementation. February 2019.

Slide 122

Internet control plane can take minutes to converge upon remote failures



Second elapsed between the data plane failure and the control plane learning about it

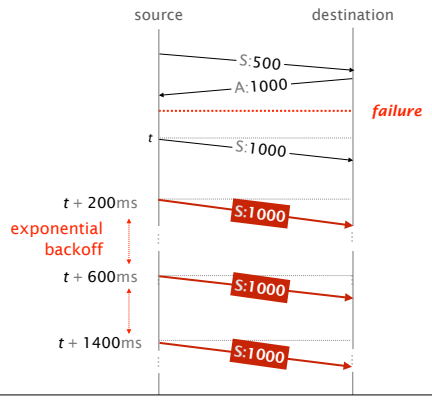
Slide 123

While the control plane can take minutes to learn about a failure,
the actual Internet traffic is affected almost instantaneously

How about we track this signal instead?

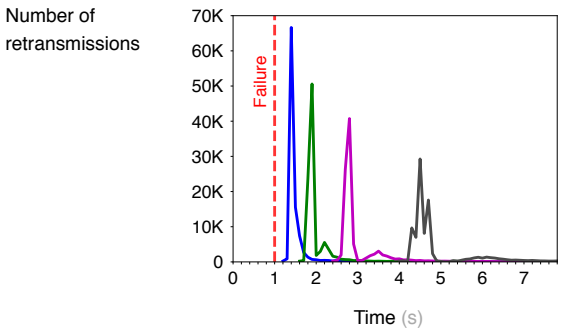
Slide 124

Internet traffic end-points retransmit packets
upon experiencing packet drops



Slide 125

Many end-points retransmitting leads to
waves of retransmission



Slide 126

Tracking this signal in the data plane is challenging

Challenges

Signal is noisy
packets loss are routinely observed

Signal fades away quickly
due to the exponential backoff

Signal is compounded over many small ones
requires per-connection tracking, which is hard to scale

Slide 127

We solve these challenges by considering a subset of the signal that we carefully craft for maximal signal-to-noise ratio

Solutions

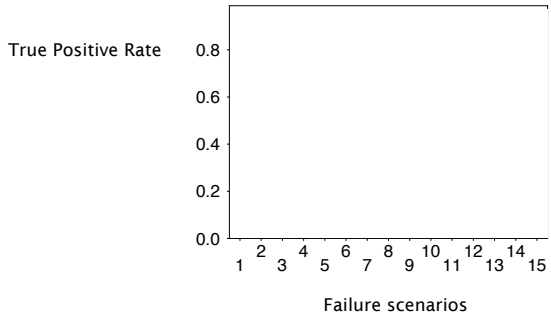
Signal is noisy
Focus on retransmissions caused by bursty losses

Signal fades away quickly
Focus on active flows (fast to retransmit after a failure)

Signal is compounded over many small ones
Rely on scalable data structures and sampling

Slide 128

Blink works well in practice,
with an inference accuracy above 80% in most cases



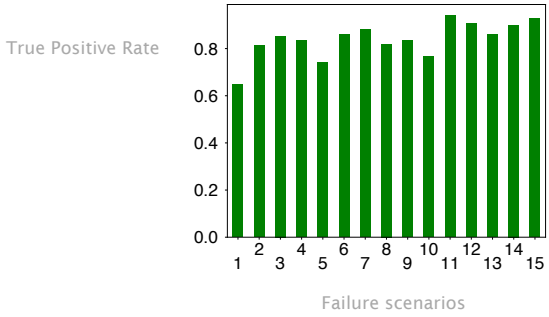
Self-driving/monitoring networks

in the age of deep network programmability

Slide 129

Blink works well in practice,

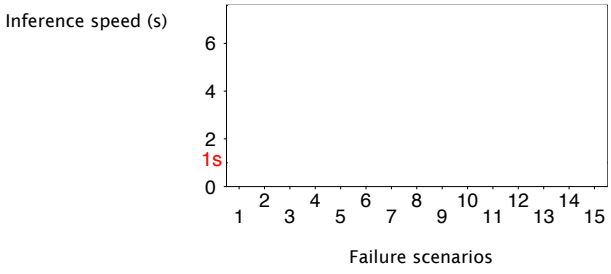
with an inference accuracy above 80% in most cases



Slide 130

Blink works well in practice,

with an inference speed below 1 second in most cases



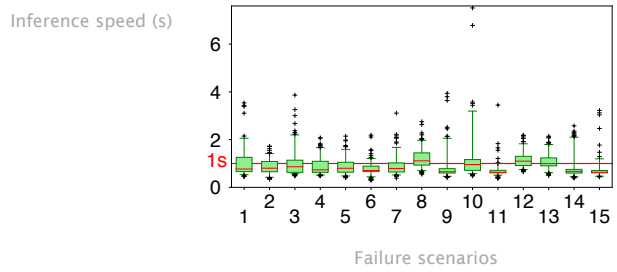
Self-driving/monitoring networks

in the age of deep network programmability

Slide 131

Blink works well in practice,

with an inference speed below 1 second in most cases

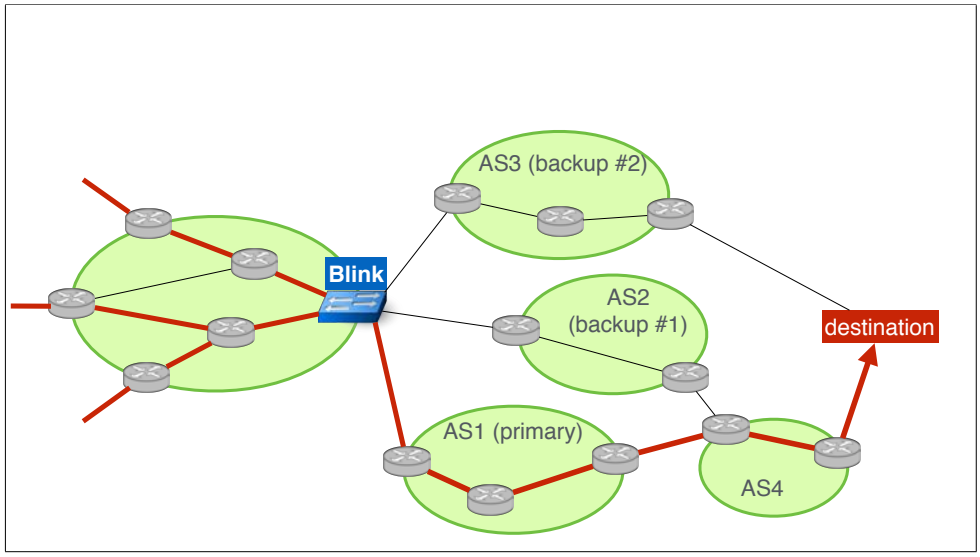


Slide 132

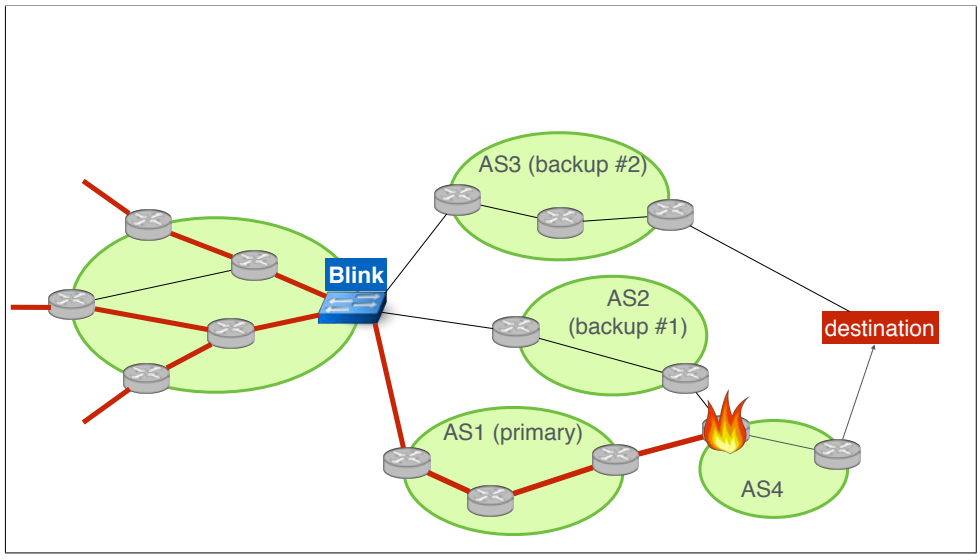
What about actuation?

Where do we send the traffic upon detecting a failure?

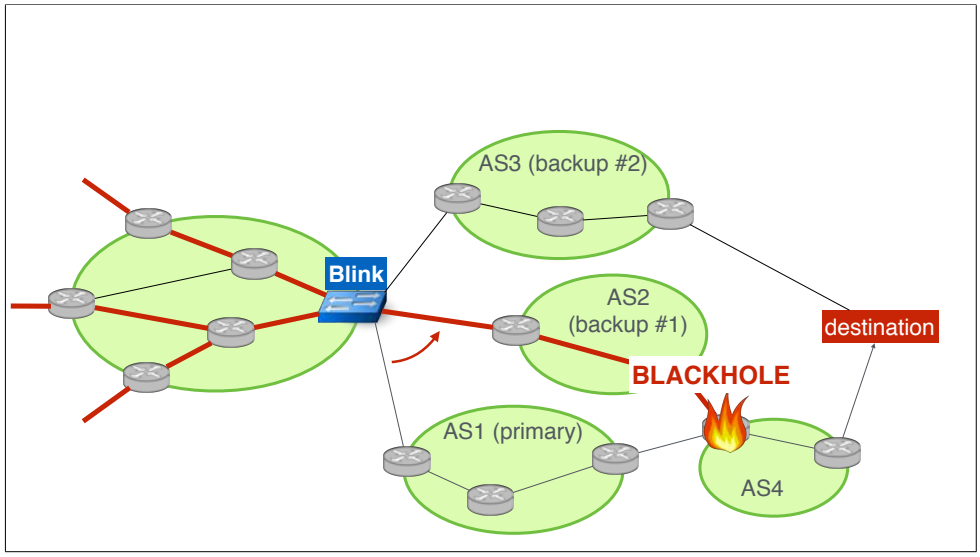
Slide 133



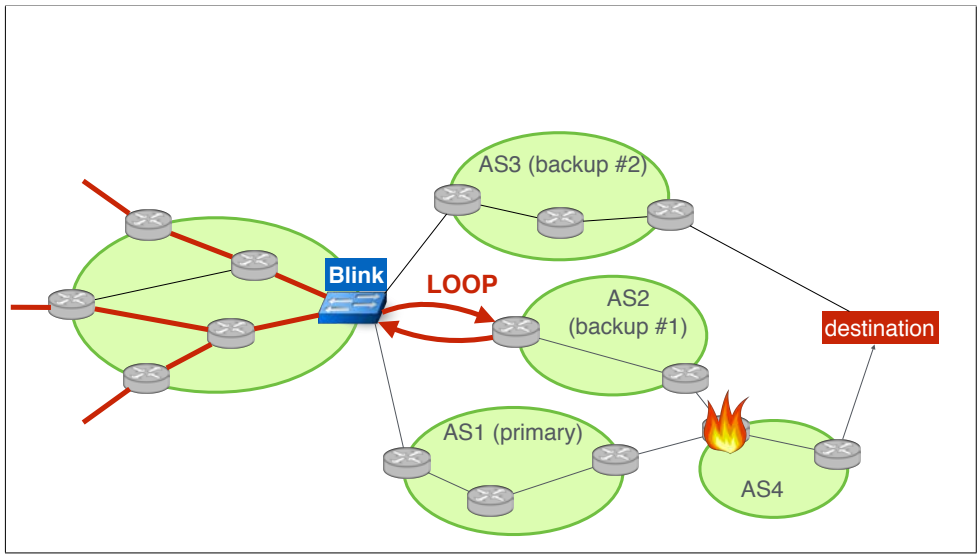
Slide 134



Slide 135



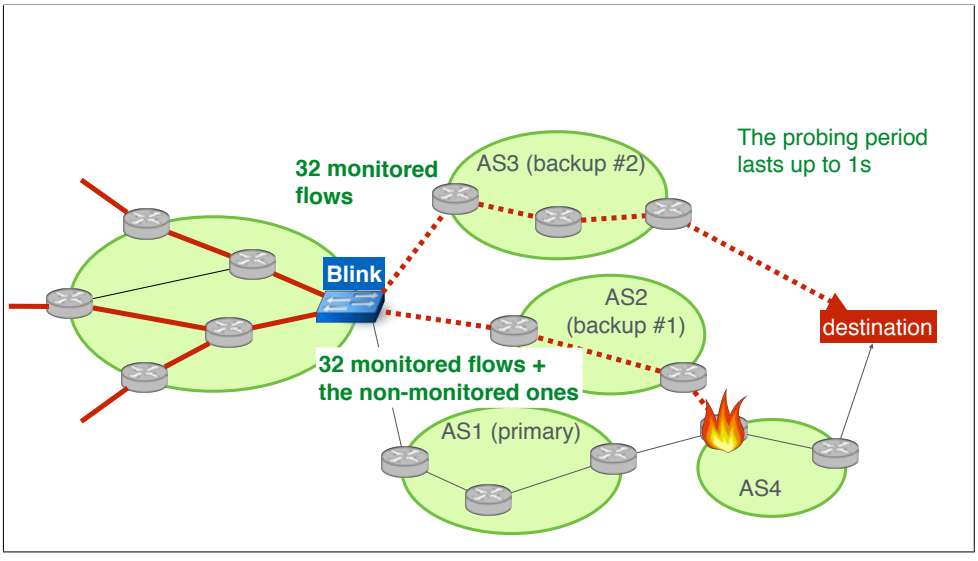
Slide 136



Slide 137

As for failures, Blink relies on data plane signals
to pick a working backup path

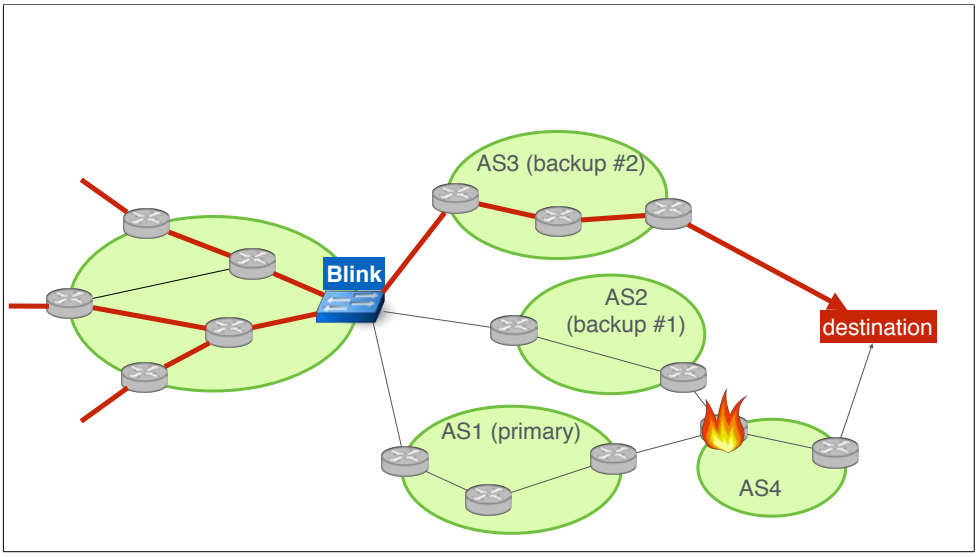
Slide 138



Self-driving/monitoring networks

in the age of deep network programmability

Slide 139



Slide 140

We intend to generalize the signals we track,
for a wider variety of applications

reliability	reroute around fine-grained failures
performance	steer traffic along the most performant path
security	detect unwanted traffic redirection

Slide 141

Part II: Taking control

applications | frameworks | methods

monitoring/inference

Slide 142

Adaptive Network Monitoring


Alexander Dietmüller


Albert Gran Alcoz


Laurent Vanbever

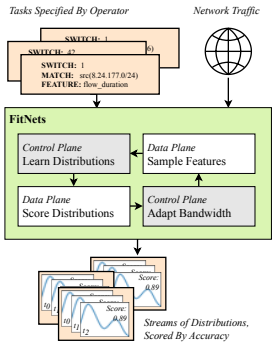
Self-driving/monitoring networks

in the age of deep network programmability

Slide 143

We're building an adaptive monitoring framework
that can learn high-quality traffic distributions

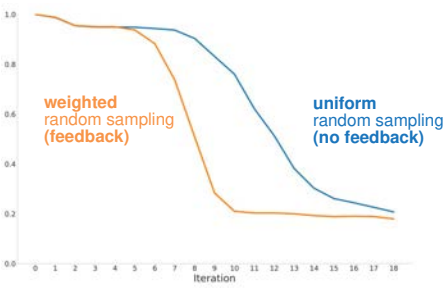
- computes traffic distribution
  in the control plane, on traffic samples
- scores these distributions
  in the data plane, on *all* traffic
- adapts the sampling rate
  according to the score



Slide 144

Our framework enables to learn distribution better and faster
than simply randomly sampling traffic

Distance from
true distribution
(Jensen-Shannon)



Slide 145

Part II: Taking control

applications | frameworks | methods

monitoring/inference

Slide 146

In-network inference


Coralie Busse-Grawitz


Roland Meier


Alexander Dietmüller

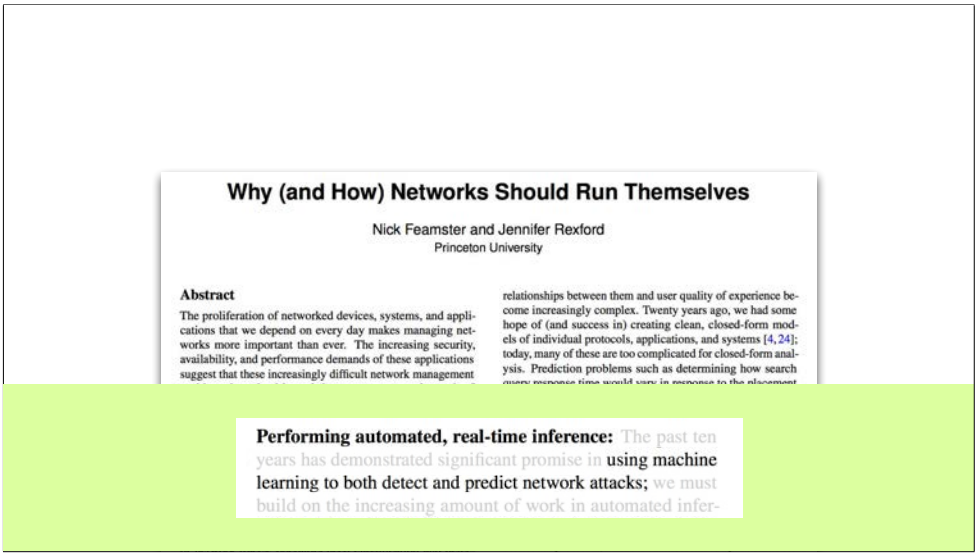

Tobias Bühler


Laurent Vanbever

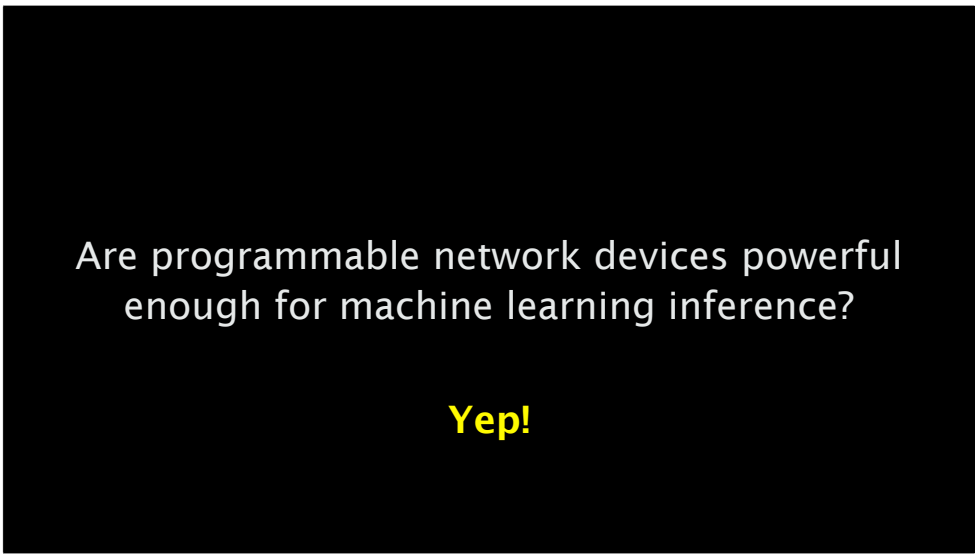
Self-driving/monitoring networks

in the age of deep network programmability

Slide 147



Slide 148

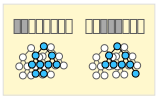


Self-driving/monitoring networks

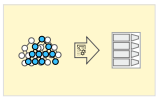
in the age of deep network programmability

Slide 149

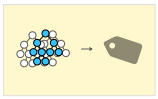
We're building a real-time,
in-network inference framework



Optimizing random forest models
for programmable network devices



Compiling random forest models
to programmable network devices



Performing runtime classification
at Tbps

Slide 150

Early, accurate & efficient classification of an ongoing flow
as an optimization problem

Self-driving/monitoring networks

in the age of deep network programmability

Slide 151

Given a labeled dataset $\mathcal{F}$ and a threshold τ

find a classifier C such that

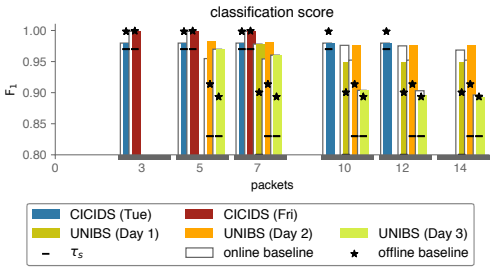
- accuracy(C) >= τ
- C fits in programmable network devices

while

- minimizing memory usage
- maximizing classification speed

Slide 152

Our prototype is already able to classify traffic after few packets, with an accuracy which is on-par with software-based solutions



Slide 153

Part II: Taking control

applications

frameworks

methods

some of our challenges lying ahead

Slide 154

Building and operating truly self-driving networks require us to overcome fundamental challenges

data

correctness

interpretability

Self-driving/monitoring networks

in the age of deep network programmability

Slide 155

data

correctness

interpretability

data is network-dependent

poor generalization capabilities

interesting events are rare

few interesting examples (if any)

data can easily be polluted

by anyone with a Internet connection…

Slide 156

how do we…

data

correctness

interpretability

ensure correctness?

network should be reachable

guarantee stability?

agents can clash with each other

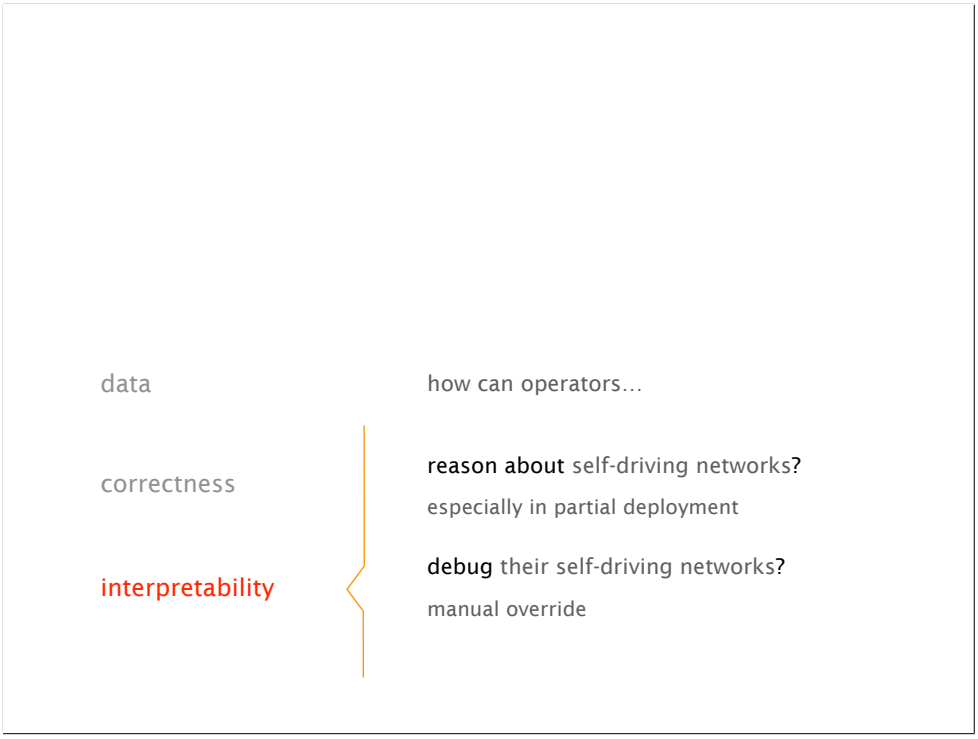
reason about optimality?

data-driven > non-data-driven (?)

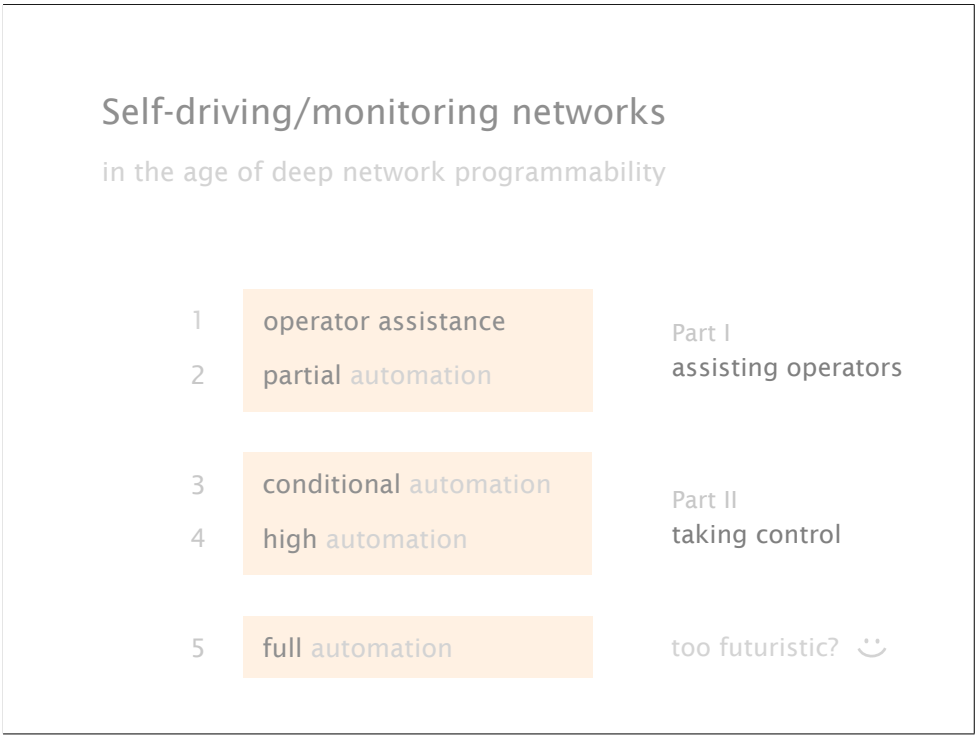
Self-driving/monitoring networks

in the age of deep network programmability

Slide 157



Slide 158



Self-driving/monitoring networks

in the age of deep network programmability

Slide 159

Slide 160

Self-driving/monitoring networks

in the age of deep network programmability

Laurent Vanbever
nsg.ee.ethz.ch

TMA
June 19 2019