Detecting network outages using different sources of data

The 2nd ACEMS Workshop on Challenges of Data and Control of Networks (ACDCN), Adelaide, Australia

Cristel Pelsser
University of Strasbourg

November, 2018

Overview

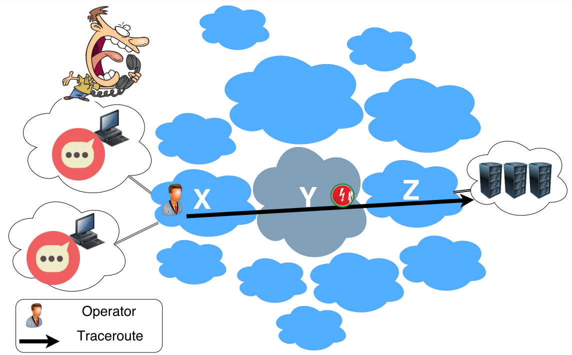
- From unsollicited traffic

  Detecting Outages using Internet Background Radiation. Andréas Guillot (U. Strasbourg), Romain Fontugne (IIJ), Pascal Mérindol (U. Strasbourg), Alberto Dainotti (CAIDA), Cristel Pelsser (U. Strasbourg). Under submission.
- From large-scale traceroute measurements

  Pinpointing Anomalies in Large-Scale Traceroute Measurements. Romain Fontugne (IIJ), Emile Aben (RIPE NCC), Cristel Pelsser (University of Strasbourg), Randy Bush (IIJ, Arrcus). IMC 2018.
- From highly distributed permanent TCP connections

  Disco: Fast, Good, and Cheap Outage Detection. Anant Shah (Colorado State U.), Romain Fontugne (IIJ), Emile Aben (RIPE NCC), Cristel Pelsser (University of Strasbourg), Randy Bush (IIJ, Arrcus). TMA 2017.

Understanding Internet health? (Motivation)



- To speedup failure identification and thus recovery
- To identify weak areas and thus guide network design

Understanding Internet health? (Problem 1)

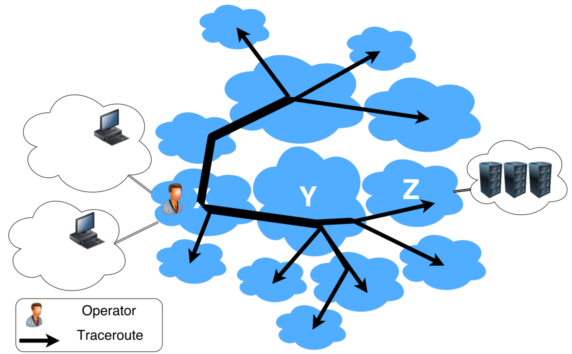
Manual observations and operations

- Traceroute / Ping / Operators' group mailing lists
- Time consuming
- Slow process
- Small visibility

→ **Our goal: Automaticaly pinpoint network disruptions (i.e. congestion and network disconnections)**

Understanding Internet health? (Problem 2)

A single viewpoint is not enough



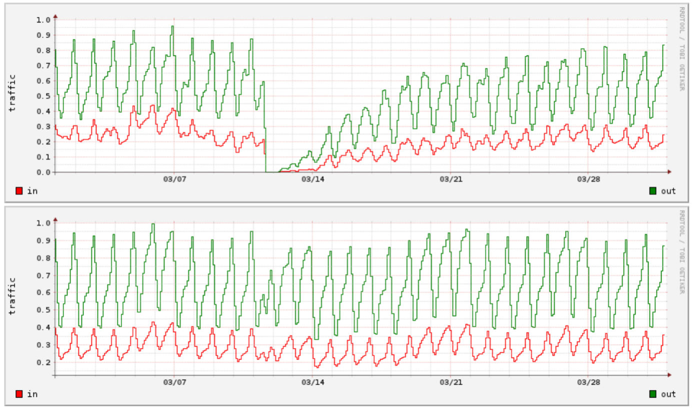
→ **Our goal: mine results from deployed platforms**

→ Cooperative and distributed approach

→ Using existing data, no added burden to the network

Understanding Internet health? (Problem 3)

Identify the right granularity

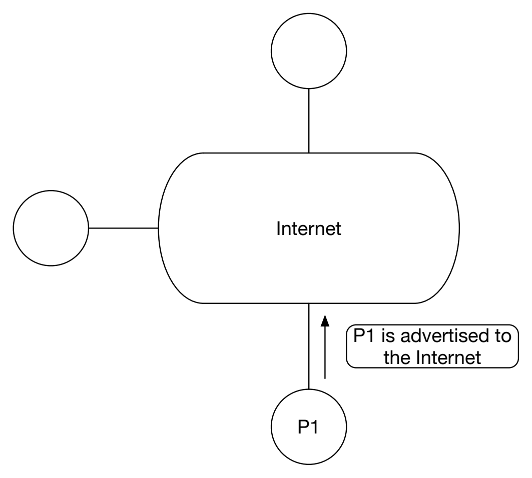


Japanese traffic for the March 2011 earthquake, Miyagi prefecture (top) and nationwide (bottom)

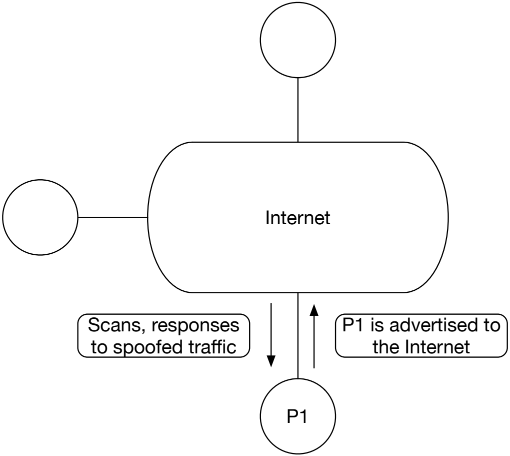
Cho et al., CoNext 2011

Outage detection from unsollicited traffic

Dataset: Internet Background Radiation

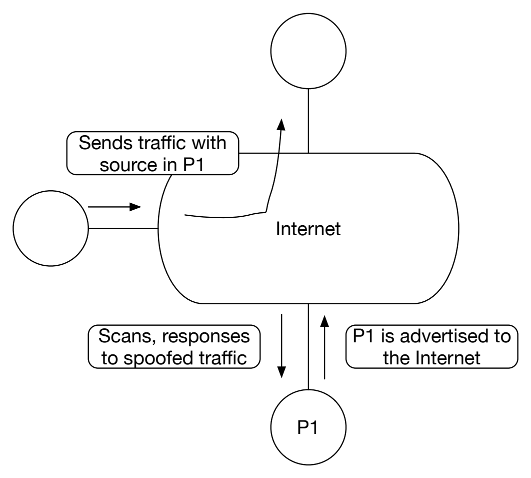


Dataset: Internet Background Radiation

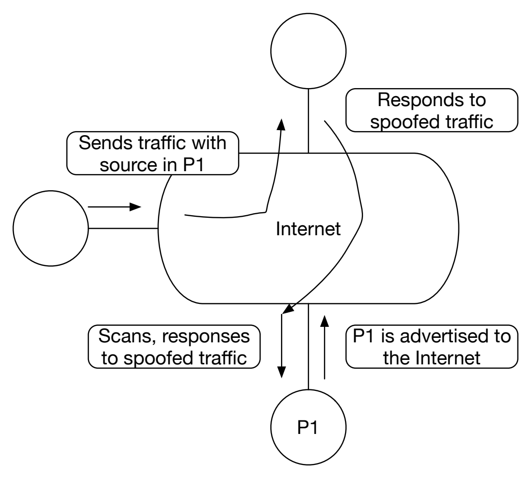


Dataset: Internet Background Radiation

Spoofed traffic



Spoofed traffic



Dataset: IP count time-series (per country or AS)

Use cases: Attacks, Censorship, Local outages detection

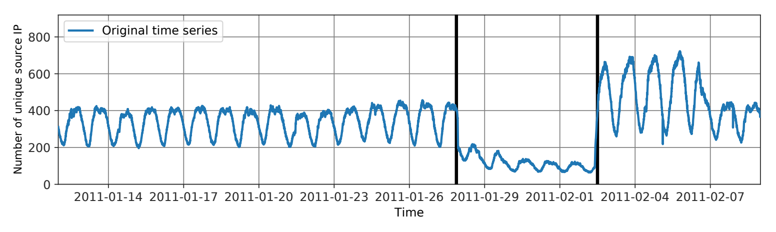
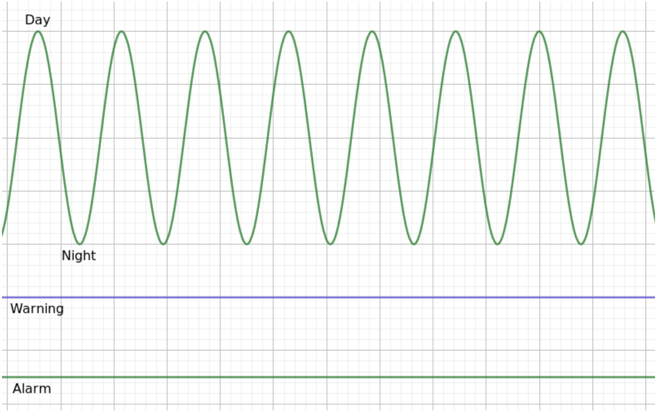


Figure 1: Egyptian revolution

$\Rightarrow$ More than 60 000 time series in the CAIDA telescope data.

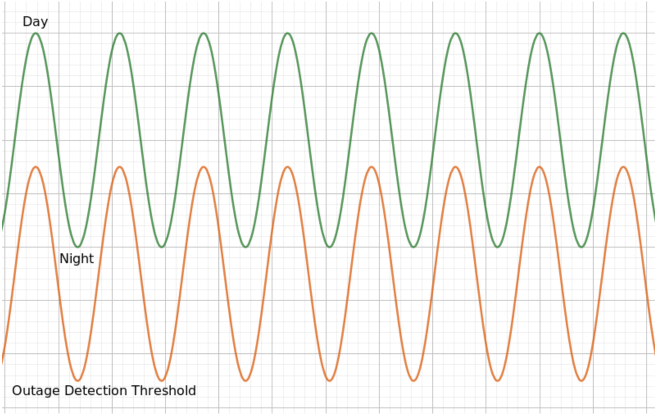
We use drops in the time series are indicators of an outage.

Current methodology used by IODA



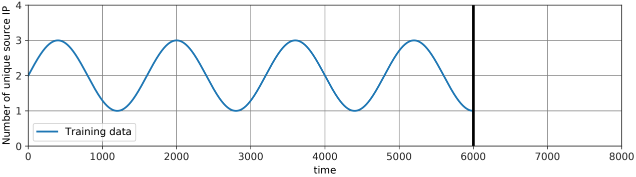
Detecting outages using **fixed thresholds**

Our goal

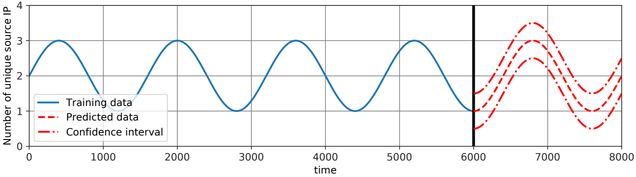


Detecting outages using **dynamic thresholds**

Outage detection process

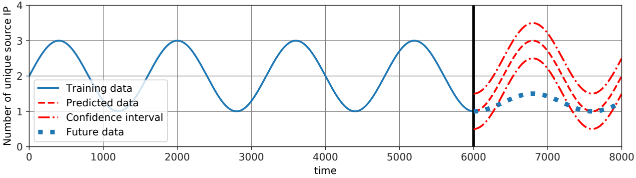


Outage detection process



Prediction and confidence interval

Outage detection process



- When the real data is outside the prediction interval, we raise an alarm.
- We want a prediction model that is robust to the seasonality and noise in the data.

The SARIMA model

S : Seasonal → Remove *trends*

AR : AutoRegressive (p)

I : Integrated → Normalize *mean and variance*

MA : Moving Average (q)

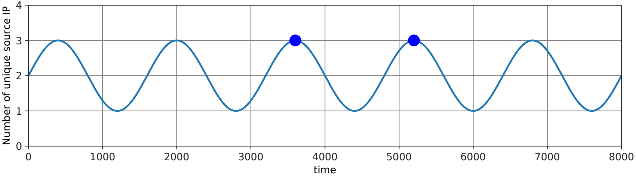


Figure 2: Original time series

The SARIMA model

S : Seasonal → Remove *trends*

AR : AutoRegressive (p)

I : Integrated → Normalize *mean and variance*

MA : Moving Average (q)

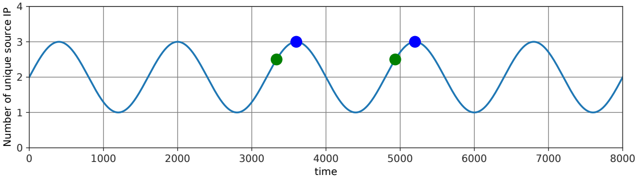


Figure 2: Our original time series

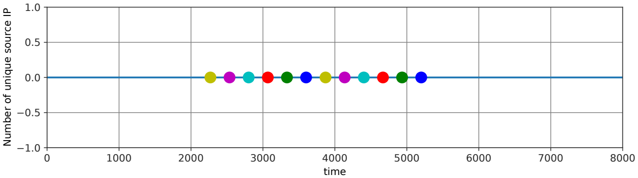


Figure 3: Differentiated time series → removed non-stationarity

The SARIMA model

S : Seasonal

AR : AutoRegressive (p) $\rightarrow$ Predict based on *past values*

I : Integrated

MA : Moving Average (q) $\rightarrow$ Predict based on *past errors*

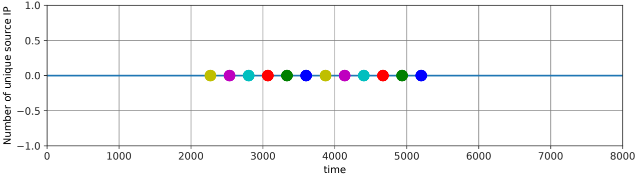


Figure 2: Differentiated time series $\rightarrow$ removed non-stationarity

Our approach

1. **Splitting the data set**
   - Training with different p and q to predict the validation set
2. Finding the best parameters
3. Detection on the test set

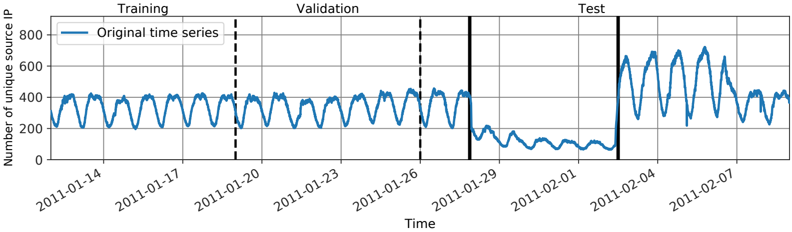


Figure 3: Different data sets

Our approach

1. Splitting the data set
2. **Finding the best parameters**
   - Minimizing the regression error (best p and q)
3. Detection on the test set

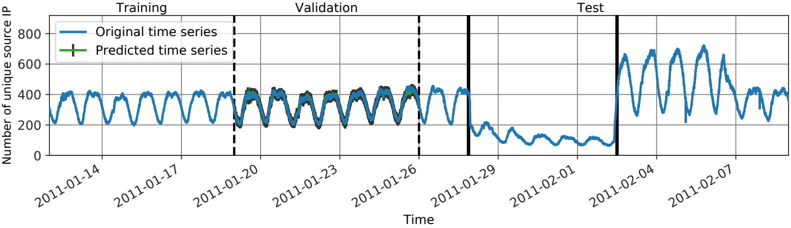


Figure 3: Making predictions on the validation set ($AR = 4, MA = 1$)

Our approach

1 Splitting the data set

2 Finding the best parameters

3 **Detection on the test set**

- Detecting and correcting outages

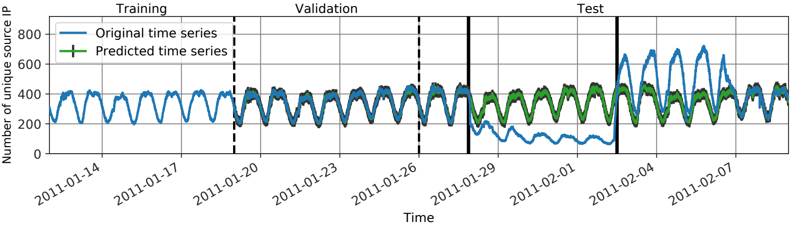


Figure 3: Predicting and inpainting the test set to preserve the integrity of the model

Definition of an outage

- Points *below* the *prediction interval*

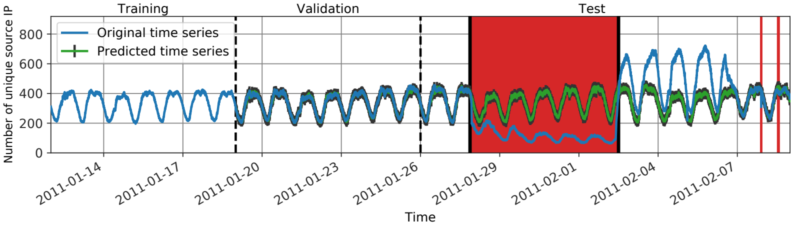


Figure 4: Analyzing the test set with the best model ($AR = 4, MA = 1$)

Validation: ground truth

Characteristics

- 130 known outages
- Multiple spatial scales
  - Countries
  - Regions
  - Autonomous Systems
- Multiple durations (from an hour to a week)
- Multiple causes (intentional or non intentional)

Evaluating our solution

Objectives

- Identifying the minimal number of IP addresses
- Identifying a good *threshold*

Threshold

- TPR of 90% and FPR of 2%

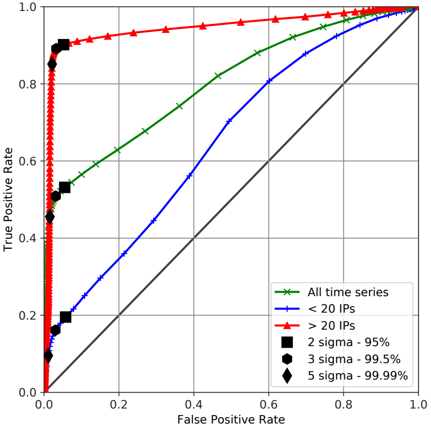


Figure 5: ROC curve

Summary

Goal

- Detecting worldwide Internet outages

Data Source

- Internet background radiation, a passive source with global coverage

Solution used

- SARIMA, a time series forecasting technique

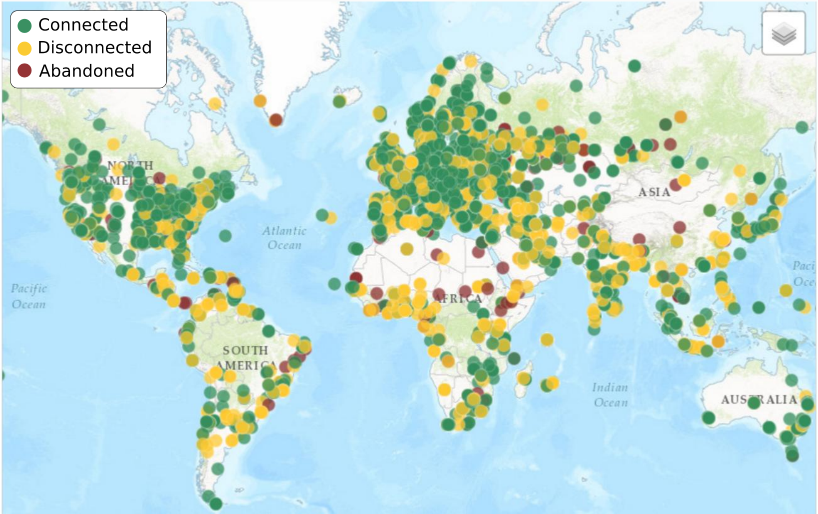
Outage detection from large-scale traceroute measurements

Actively measures Internet connectivity

- Ethernet port
- Automatically perform active measurements: ping, **traceroute**, DNS, SSL, NTP and HTTP
- All results are collected by RIPE NCC

RIPE Atlas: coverage

9300+ active probes!



RIPE Atlas: traceroutes

Two repetitive large-scale measurements

- *Builtin*: traceroute every 30 minutes to all DNS root servers ($\approx$ 500 server instances)
- *Anchoring*: traceroute every 15 minutes to 189 collaborative servers

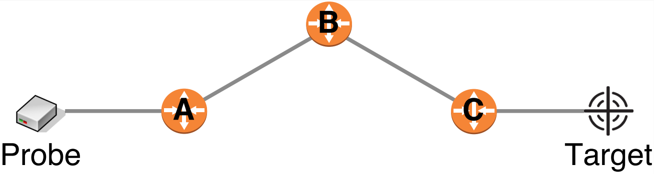
Analyzed dataset

- May to December 2015
- 2.8 billion IPv4 traceroutes
- 1.2 billion IPv6 traceroutes

Monitor delays with traceroute?

Traceroute to "www.target.com"

```
~$ traceroute www.target.com
traceroute to target, 30 hops max, 60 byte packets
 1   A        0.775 ms   0.779 ms   0.874 ms
 2   B        0.351 ms   0.365 ms   0.364 ms
 3   C        2.833 ms   3.201 ms   3.546 ms
 4   Target   3.447 ms   3.863 ms   3.872 ms
```



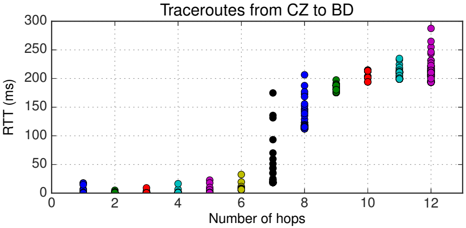
Round Trip Time (RTT) between B and C?

Report abnormal RTT between B and C?

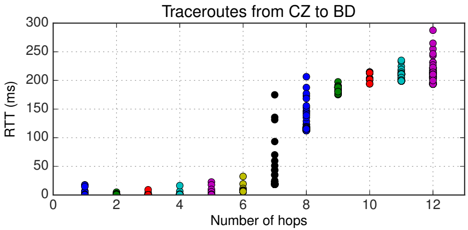
Monitor delays with traceroute?

Challenges:

- Noisy data

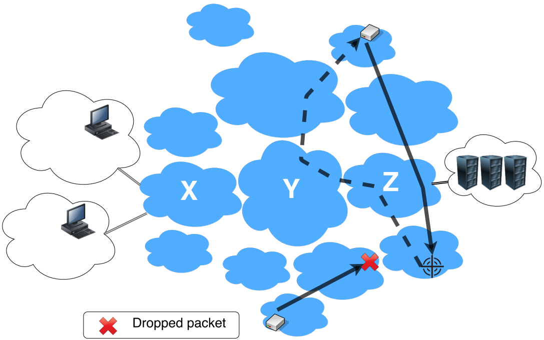


Monitor delays with traceroute?



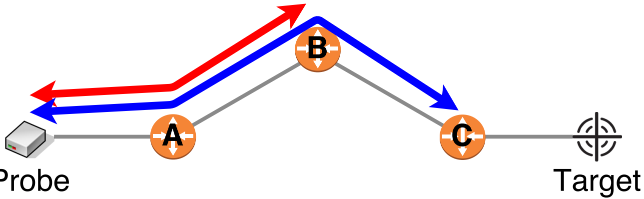
Challenges:

- Noisy data
- Traffic asymmetry
- Packet loss



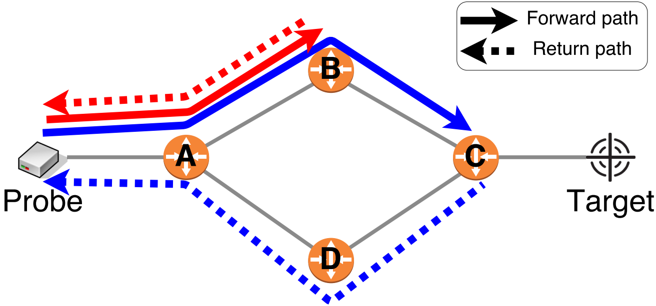
What is the RTT between B and C?

```
~$ traceroute www.target.com
traceroute to target, 30 hops max, 60 byte packets
 1  A        0.775 ms  0.779 ms  0.874 ms
 2  B        0.351 ms  0.365 ms  0.364 ms
 3  C        2.833 ms  3.201 ms  3.546 ms
 4  Target   3.447 ms  3.863 ms  3.872 ms
```



RTT_C - RTT_B = RTT_{CB}?

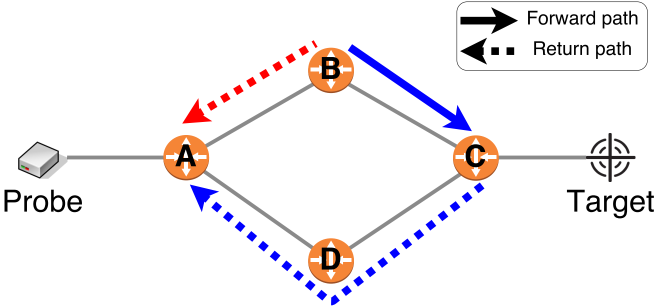
What is the RTT between B and C?



RTT_C - RTT_B = RTT_{CB}**?**

- No!
- Traffic is asymmetric
- RTT_B and RTT_C take **different return paths!**

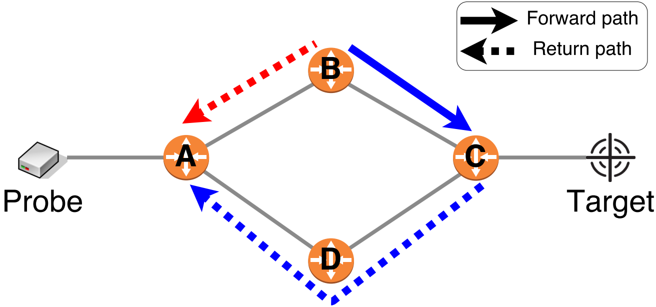
What is the RTT between B and C?



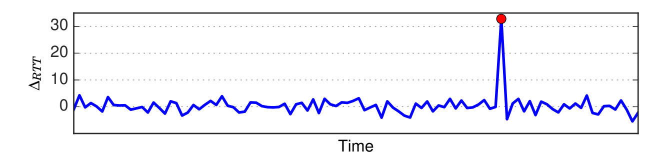
RTT_C - RTT_B = RTT_{CB}**?**

- No!
- Traffic is asymmetric
- RTT_B and RTT_C take **different return paths!**
- **Differential RTT**: $\Delta_{CB} = RTT_C - RTT_B = d_{BC} + e_p$

Problem with differential RTT



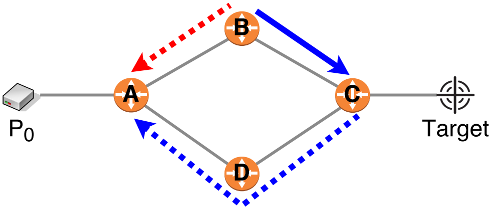
Monitoring Δ_{CB} over time:



$\rightarrow$ Delay change on BC? CD? DA? BA???

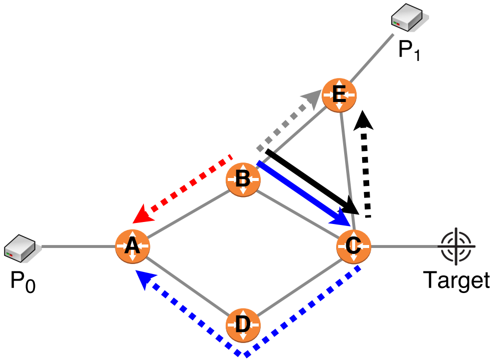
Proposed Approach: Use probes with different return paths

Differential RTT: $\Delta_{CB} = x_0$



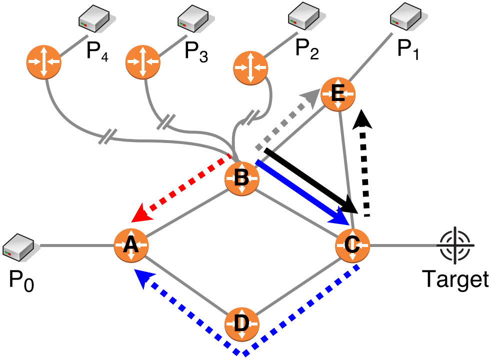
Proposed Approach: Use probes with different return paths

Differential RTT: $\Delta_{CB} = \{x_0, x_1\}$



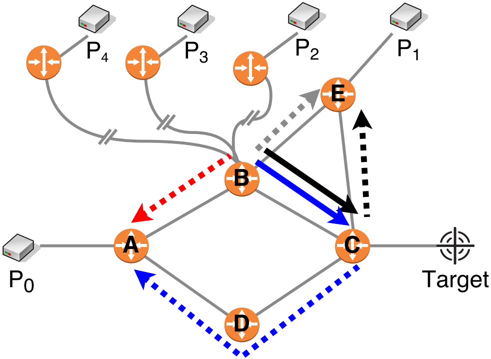
Proposed Approach: Use probes with different return paths

Differential RTT: $\Delta_{CB} = \{x_0, x_1, x_2, x_3, x_4\}$



Proposed Approach: Use probes with different return paths

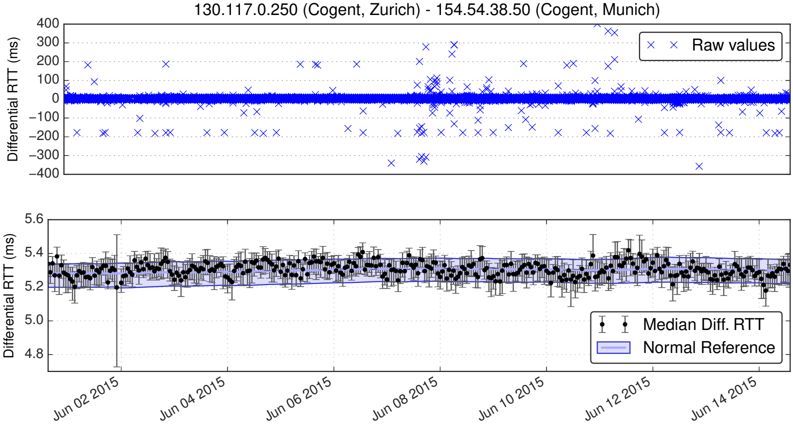
Differential RTT: $\Delta_{CB} = \{x_0, x_1, x_2, x_3, x_4\}$



Median Δ_{CB}:

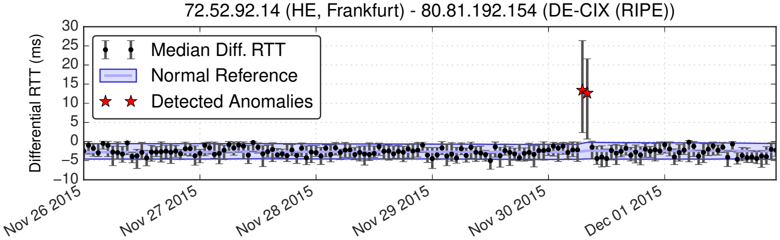
- Stable if a few return paths delay change
- Fluctuate if delay on BC changes

Median Diff. RTT: Tier1 link, 2 weeks of data, 95 probes



- **Stable** despite noisy RTTs (not true for average)
- Normally distributed

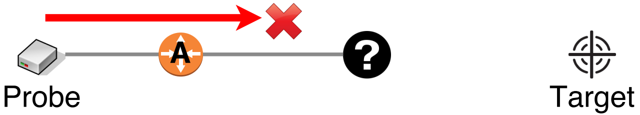
Detecting congestion



Significant RTT changes:

Confidence interval not overlapping with the normal reference

Packet loss

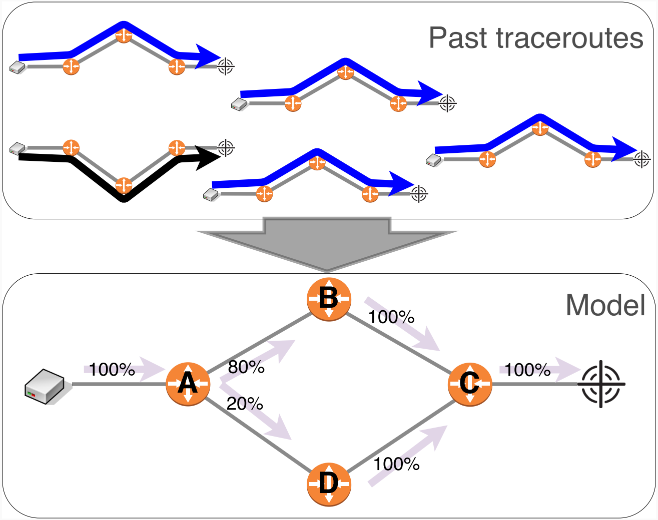


Worst case: router is not responding

- Cannot obtain RTT values
- Need to identify the faulty link

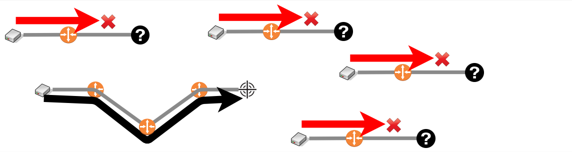
Packet forwarding model

Learn usual paths from past traceroutes:

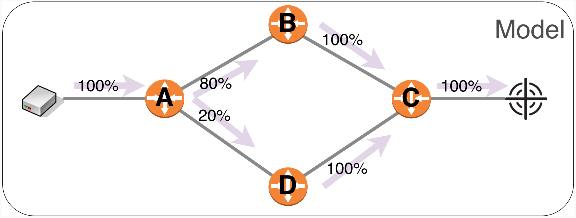


Identifying faulty links

In case of packet loss:



Query the model for the expected next hop



→ Link AB is dropping packets!

Analyzed dataset

- Atlas *builtin/anchoring* measurements
- From May to Dec. 2015
- Observed 262k IPv4 and 42k IPv6 links

We found a lot of congested links!
Let's see only two significant examples

Study case: DDoS on DNS root servers

Two attacks:

- Nov. 30th 2015
- Dec. 1st 2015

Almost all servers are anycast

- Congestion at the 531 sites?
- Found 129 instances altered by the attacks


The Register®
Biting the hand that feeds IT

DATA CENTRE SOFTWARE NETWORKS SECURITY INFRASTRUCTURE DEVOPS BUSINESS HARDWARE

Networks

Internet's root servers take hit in DDoS attack

Who's testing the limits of the DNS system?

8 Dec 2015 at 23:10, Kieren McCarthy

The internet's root servers came under a c
effectively knocked three of the 13 critical pi

The attack came just days before the Janet

According to a first analysis of the root serv
attack occurred on November 30, 2015 bet

Many, but not all, of the root servers receive
flood network connections and cause timec
messages for a single domain name; the s

Ultimately, the operators affected by the att
proper analysis is now underway to discove

Of perhaps most concern is the fact that ev
deal with such an attack, a number of the s

The root servers themselves make up the
as a sort of global directory for all the other

Due to the internet's design, the servers the
you compare it to what companies like Goo
immediate problems for the wider internet,
thousands of other servers.

That said, any attack on the DNS' infrastruc
longer than a day, it would start causing sig


Home Hacking Tech Cyber Attacks Vulnerabilities Malware Spying

The Hacker News™
Security in a serious way

Someone Just Tried to Take Down Internet's Backbone with 5 Million Queries/Sec

Wednesday, December 09, 2015 Swati Khandelwal

Someone just DDoSed one of the most critical organs of the Internet anatomy – **The Internet's DNS Root Servers**.

Early last week, a flood of as many as 5 Million queries per second hit many of the Internet's DNS (*Domain Name System*) Root Servers that act as the authoritative reference for mapping domain names to IP addresses and are a total of 13 in numbers.

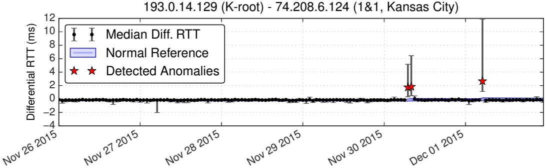
The attack, commonly known as **Distributed Denial of Service** (DDoS) attack, took place on two separate occasions.

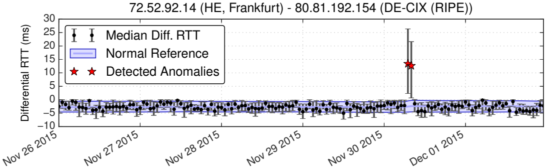
The first DDoS attack to the Internet's backbone root servers launched on *November 30* that lasted 160 minutes (*almost 3 hours*), and the second one started on *December 1* that lasted almost an hour.

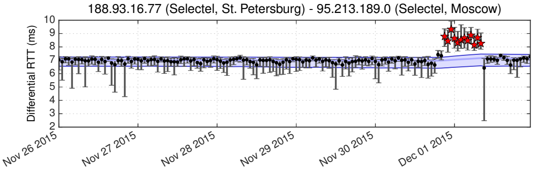
Massive Attacks Knocked Many of the 13 Root Servers Offline

The DDoS attack was able to knock 3 out of the 13 DNS root servers of the Internet offline for a

Observed congestion

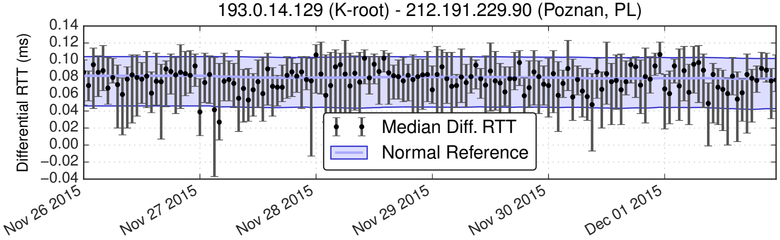






- Certain servers are affected only by one attack
- Continuous attack in Russia

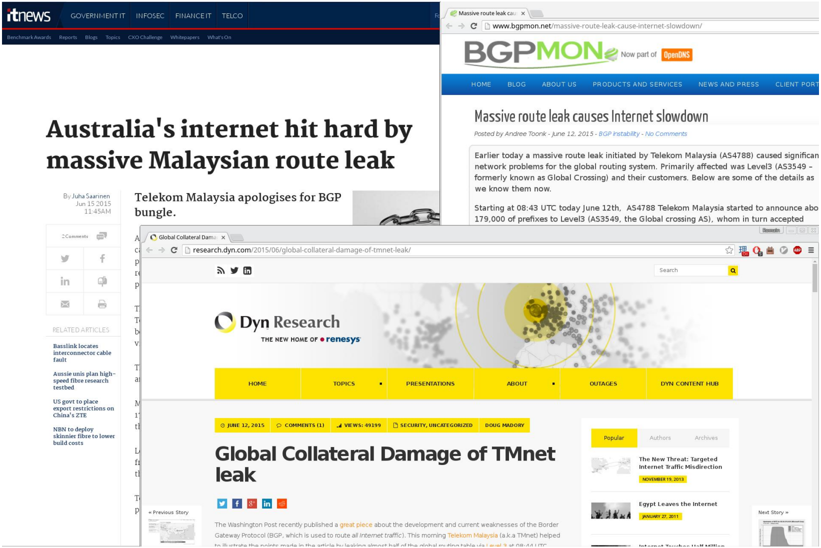
Unaffected root servers



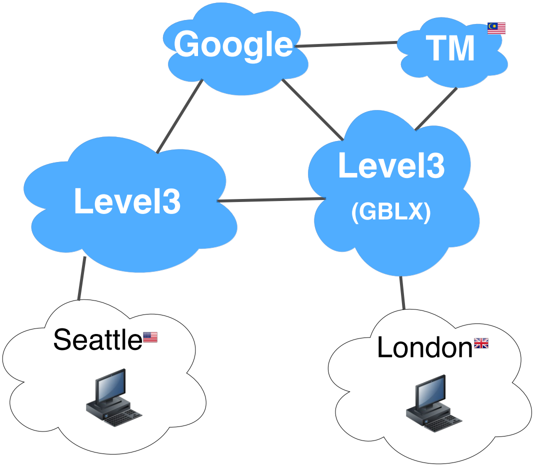
Very stable delay during the attacks

- Thanks to anycast!
- Far from the attackers

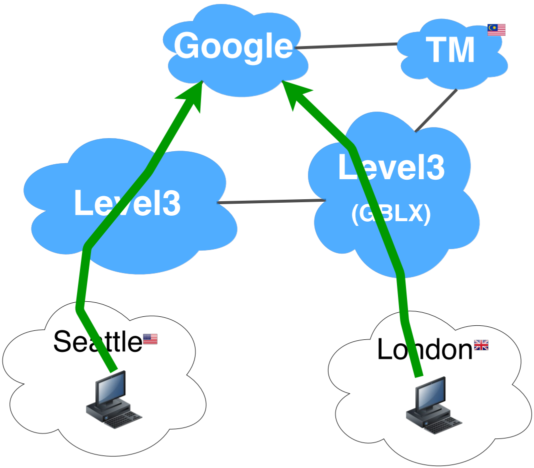
Study case: Telekom Malaysia BGP leak

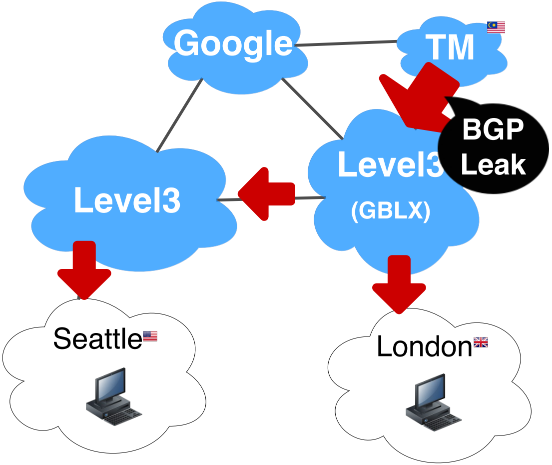


Study case: Telekom Malaysia BGP leak

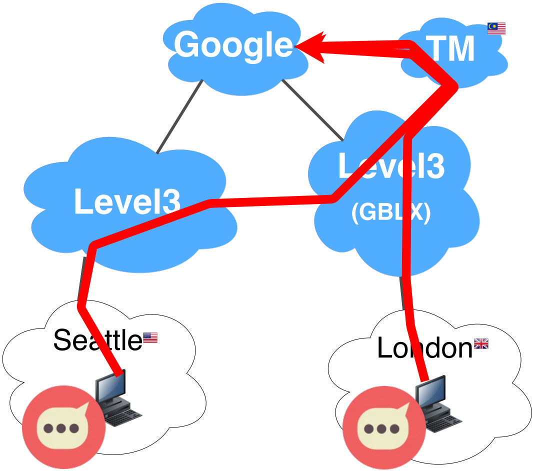


Study case: Telekom Malaysia BGP leak




Google
TM
BGP Leak
Level3
Level3
(GBLX)
Seattle
London

Study case: Telekom Malaysia BGP leak

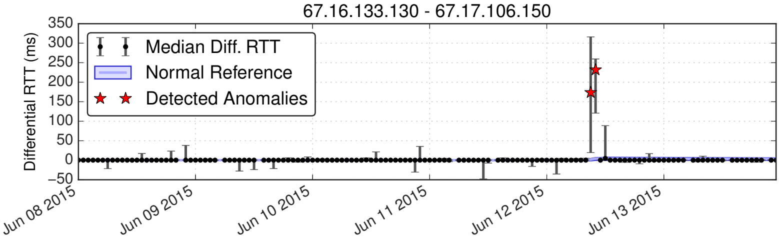


Not only with Google... but about **170k prefixes!**

Congestion in Level3

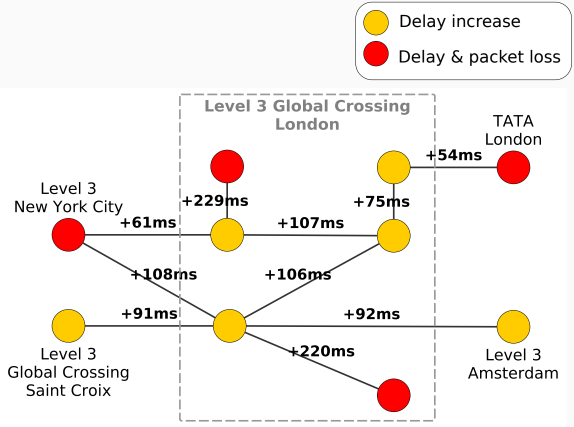
Rerouted traffic has congested Level3 (120 reported links)

- Example: 229ms increase between two routers in London!



Congestion in Level3

Reported links in London:



→ Traffic staying within UK/Europe may also be altered

Per-AS alarm for delay

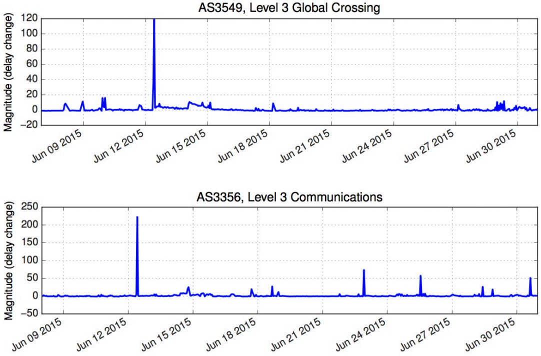


Figure 8: Delay change magnitude for all monitored IP addresses in two Level(3) ASs.

And forwarding too!

Per-AS alarm for forwarding

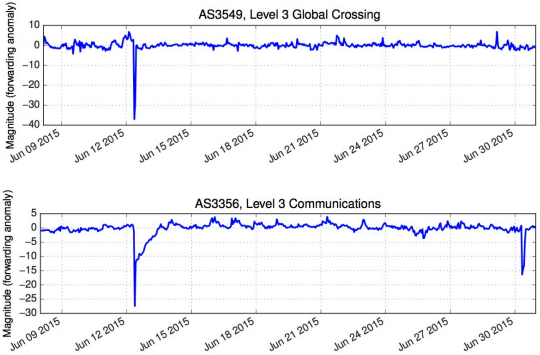


Figure 9: Forwarding anomaly magnitude for all monitored IP addresses in two Level(3)ASs.

Monitor delays with the Atlas platform

- Billions of (noisy) traceroutes

Detect and locate Internet congestion

- Robust statistical analysis
- Diverse root causes: remote attacks, routing anomalies, etc...
- Give a lot of new insights on reported events

On going work with RIPE NCC:

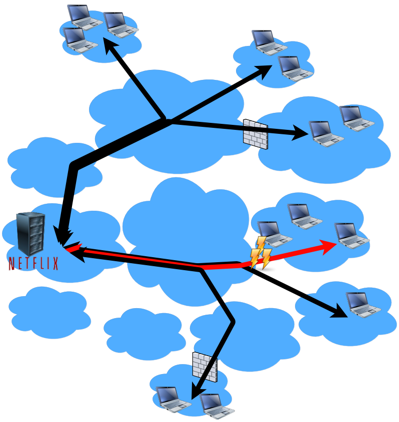
- Online detection and reports for network operators

Outage detection from highly distributed permanent TCP connections

Proposed Approach

Disco:

- Monitor long-running TCP connections and synchronous disconnections from related network/area
- We apply Disco on RIPE Atlas data, where probes are widely distributed at the edge and behind NATs/CGNs providing visibility Trinocular may not have



→ Outage = synchronous disconnections from the same topological/geographical area

Assumptions / Design Choices

Rely on TCP disconnects

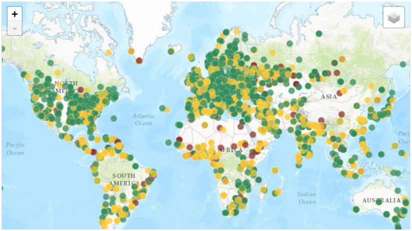
- Hence the granularity of detection is dependent on TCP timeouts

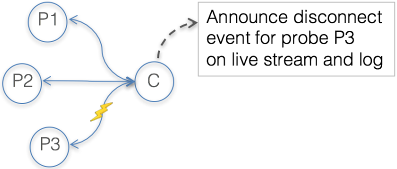
Bursts of disconnections are indicators of interesting outage

- While there might be non bursty outages that are interesting, Disco is designed to detect large synchronous disconnections

RIPE Atlas platform

- 10k probes worldwide
- Persistent connections with RIPE controllers
- Continuous traceroute measurements (see outages from inside)





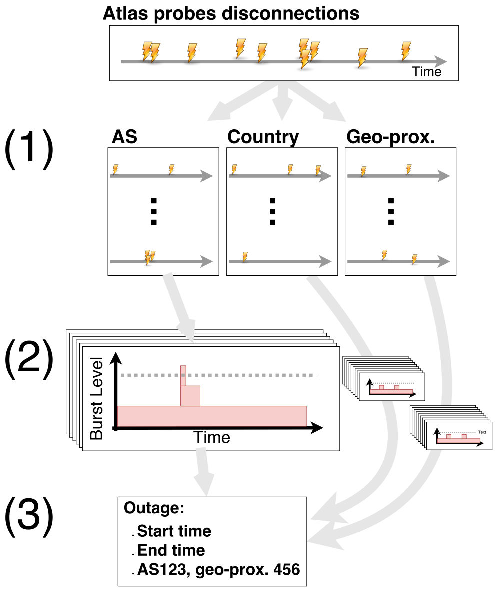
→ Dataset: Stream of probe connection/disconnections (from 2011 to 2016)

Disco Overview

1. Split disconnection stream in sub-streams (AS, country, geo-proximate 50km radius)

2. Burst modeling and outage detection

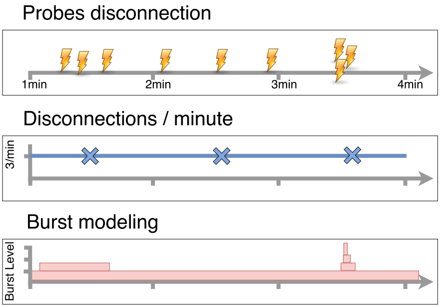
3. Aggregation and outage reporting



Why Burst Modeling?

Goal: How to find synchronous disconnections?

- Time series conceal temporal characteristics
- Burst model estimates disconnections arrival rate at any time

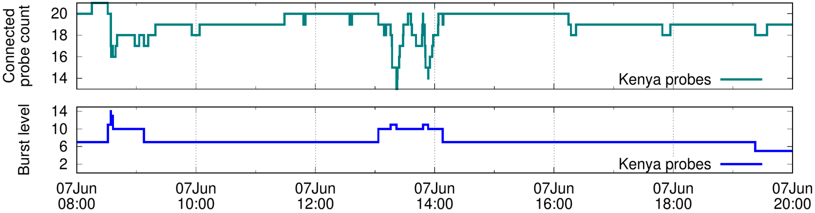


Implementation: Kleinberg burst model[1]

[1] J. Kleinberg. “Bursty and hierarchical structure in streams”, Data Mining and Knowledge Discovery, 2003.

Burst modeling: Example

- Monkey causes blackout in Kenya at 8:30 UTC June, 7th 2016
- Same day RIPE rebooted controllers



Results

Outage detection:

- Atlas probes disconnections from 2011 to 2016
- Disco found 443 significant outages

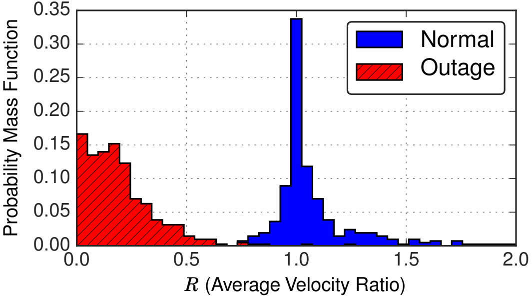
Outage characterization and validation:

- Traceroute results from probes (buffered if no connectivity)
- Outage detection results from Trinocular

Validation (Traceroute)

Comparison to traceroutes:

- Probes in detected outages can reach traceroutes destination?
  $\rightarrow$ Velocity ratio: proportion of completed traceroutes in given time



$\rightarrow$ **Velocity ratio ≤ 0.5 for 95% of detected outages**

Validation (Trinocular)

Comparison to Trinocular (2015):

- Disco found 53 outages in 2015
- Corresponding to 851 /24s (only 43% is responsive to ICMP)

Results for /24s reported by Disco and pinged by Trinocular:

- 33/53 are also found by Trinocular
- 9/53 are missed by Trinocular (avg time of outages $<$ 1hr)
- Other outages are partially detected by Trinocular

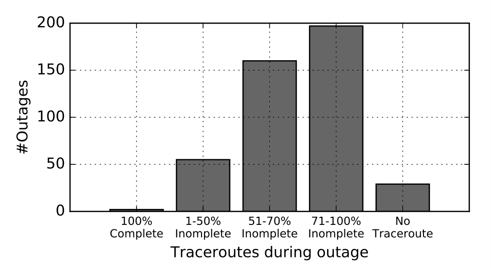
23 outages found by Trinocular are missed by Disco

- Disconnections are not very bursty in these cases

$\rightarrow$ **Disco's precision: 95%, recall: 67%**

Outage Characterization (1)

Percentage of traceroutes reaching their target:



- In most cases probes lost complete connectivity
- For cases in 1-70%, probes have limited connectivity to local targets (e.g. anycasted services)
- Complete lack of traceroute in case of power outage

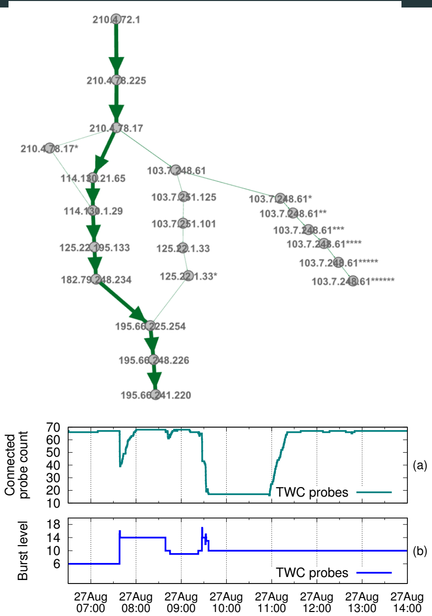
Outage Characterization (2)

Traceroutes allow us to identify faulty hops

- Learn typical paths and identify expected hop
- Found several forwarding loop

Example: TWC outage in 2014

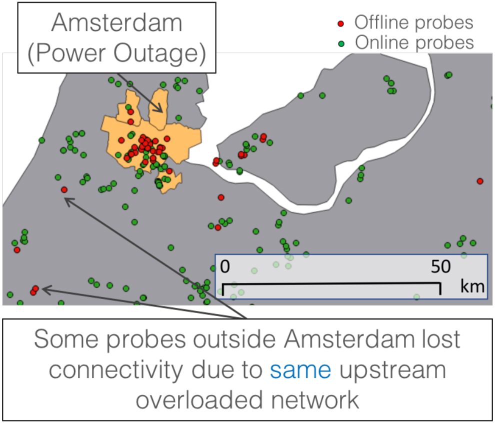
- 73% of traceroutes revealed a forwarding loop



Example of geo-proximate outage

Amsterdam outage (2017)

- Disco's detection correlated with network problems between two network elements of a large provider



Summary:

Disco: Outage Detection using long-lived TCP connections

- **Fast:**
  - Passive monitoring
  - Processed 6 years of data in 103 minutes
- **Good:**
  - Precise location of outages in space and time
  - 95% precision, 67% recall
- **Cheap:**
  - Generates no measurement packet
  - Monitor beyond NATs

Conclusions and perspectives

We proposed 3 different techniques to detect outages for 3 different sources of data

- Each source of data has its own coverage, noise, properties
- Identifying the suitable model is a challenge
- There is no subsequent ground truth to validate the results

Turn this

Into this

http://ihr.iijlab.net