Network Security and Measurement

- Data Plane Measurements -

Prof. Dr. Thomas Schmidt

http://inet.haw-hamburg.de | t.schmidt@haw-hamburg.de

Agenda

How to obtain data plane measurements?

Passive measurements:

    Monitoring Flows

    IPFIX (IP Flow Information Export)
    Network Telescopes

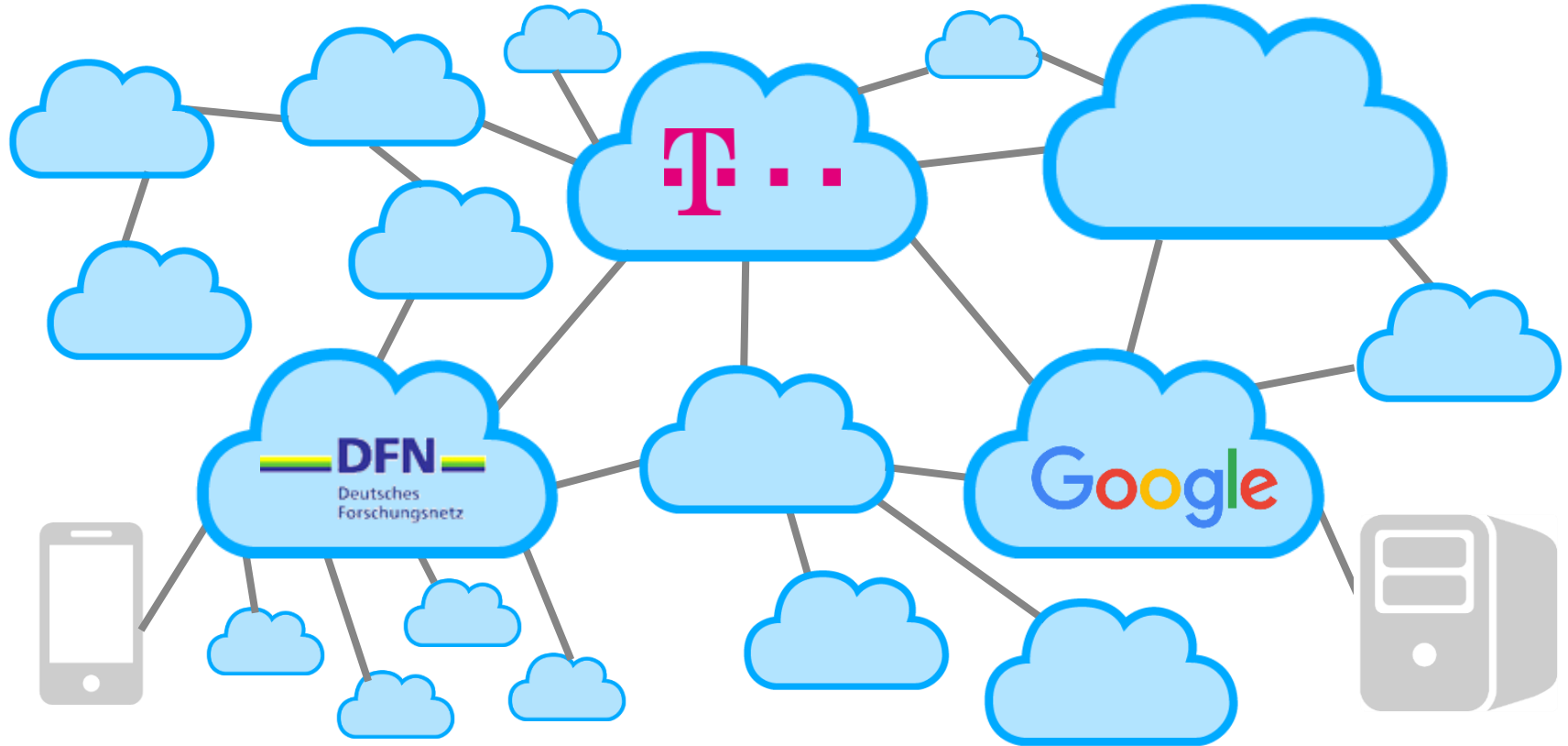Active measurements:

    Challenges and good practice

    Traceroute measurements are not trivial
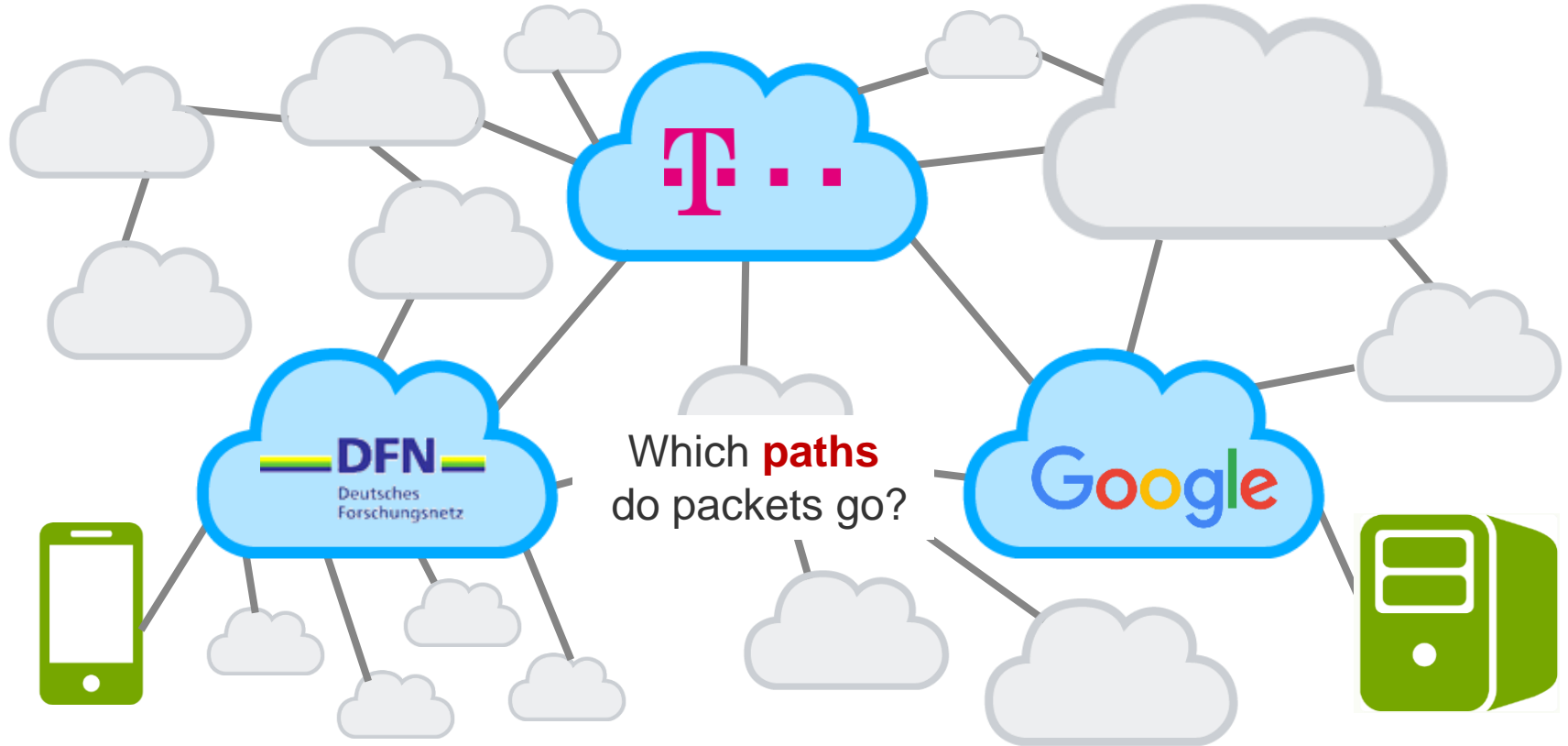
    Active measurement infrastructures

Technical Challenge

MEASURING THE DATA PLANE

From **control** to data plane

From control to data plane

Which **paths** do packets go?

From control to data plane

Which **service** is available?

Which **traffic** is exchanged?

Why should we measure the data plane?

Protocol deployment

Network provisioning

Security

…

How to measure the data plane?

Active

Passive

Examples

Ping, traceroute, scanning, …

Traffic monitoring, log files, …

Listen and Record

PASSIVE DATA PLANE MEASUREMENTS

Passive data measurement introduces two questions

How to select traffic?

Sampling vs. full capture

How to classify the captured traffic?

Port-based vs. application payload

Full packet captures are not always achievable

Privacy requirements

Scalability challenges

Select only a subset of data, either in terms of packets or packet headers.

Filtering

"Filtering is the deterministic selection of packets based on the Packet Content, the treatment of the packet at the Observation Point, or deterministic functions of these occurring in the Selection State." [RFC 5475]

Sampling

"Sampling is targeted at the selection of a representative subset of packets. The subset is used to infer knowledge about the whole set of observed packets without processing them all. The selection can depend on packet position, and/or on Packet Content, and/or on (pseudo) random decisions." [RFC 5475]

Two basic sampling policies

Systematic sampling
Deterministic selection of every 1-out-of-k elements

k=4

Random sampling
Probabilistic selection of elements

Random p=1/4

Composite sampling strategies

Stratified sampling
Leverage a priori information and group k consecutive elements, select one randomly within the group
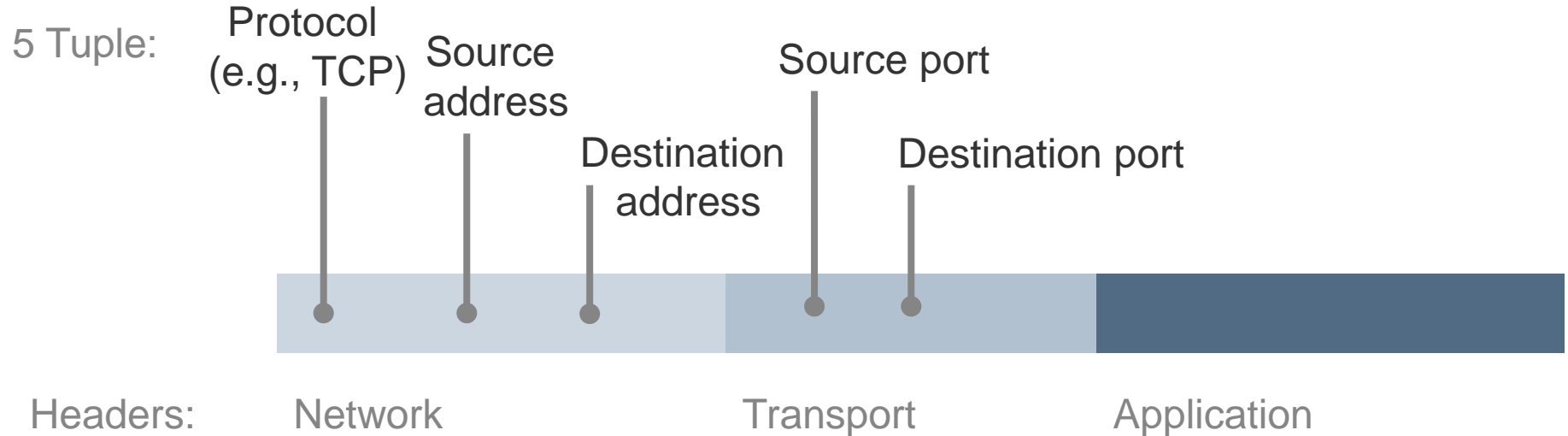
Systematic SYN sampling
Filter all SYN packet and sample k packets

Sampling can be applied on a **per packet** base or **per flow** base.

A flow is typically defined by a 5 tuple

5 Tuple:

Protocol (e.g., TCP)

Source address

Destination address

Source port

Destination port

Headers: Network   Transport   Application

Packet sampling: Example

Packet sampling uses randomness in the sampling process to prevent synchronization with any periodic patterns in the traffic.

Consider a link with 1,000,000 packets.

You sample 2,500 packets uniformly randomly (sampling rate 0,25%).

1,000 of the sampled packets belong to voice traffic.

How many of the 1M packets are most likely voice packets?

Packet sampling: Example

Packet sampling uses randomness in the sampling process to prevent synchronization with any periodic patterns in the traffic.

Consider a link with 1,000,000 packets.

You sample 2,500 packets uniformly randomly (sampling rate 0,25%).

1,000 of the sampled packets belong to voice traffic.

How many of the 1M packets are most likely voice packets?

400,000 packets, or 40% (1,000/2,500 = 0,4).

Sampling error

Measurement accuracy does not depend on the number of packets but on the number of samples.

Accuracy can be improved by (i) increasing the sampling rate or (ii) look at the data over longer time.

TRAFFIC CLASSIFICATION

Which packet belongs to which application?

Which packet belongs to which application?

How to classify systematically?
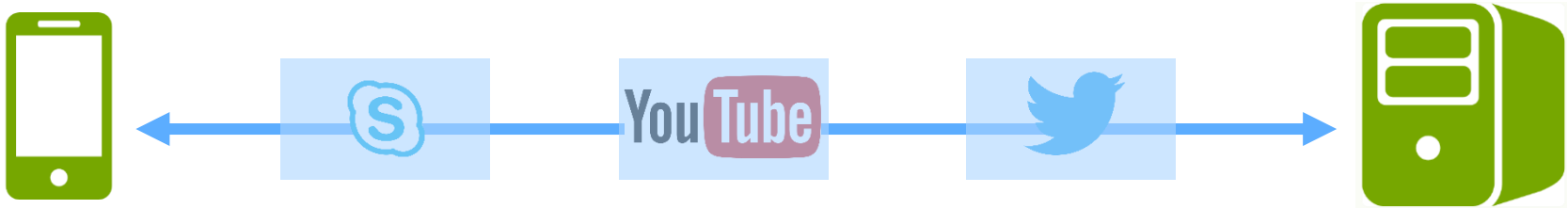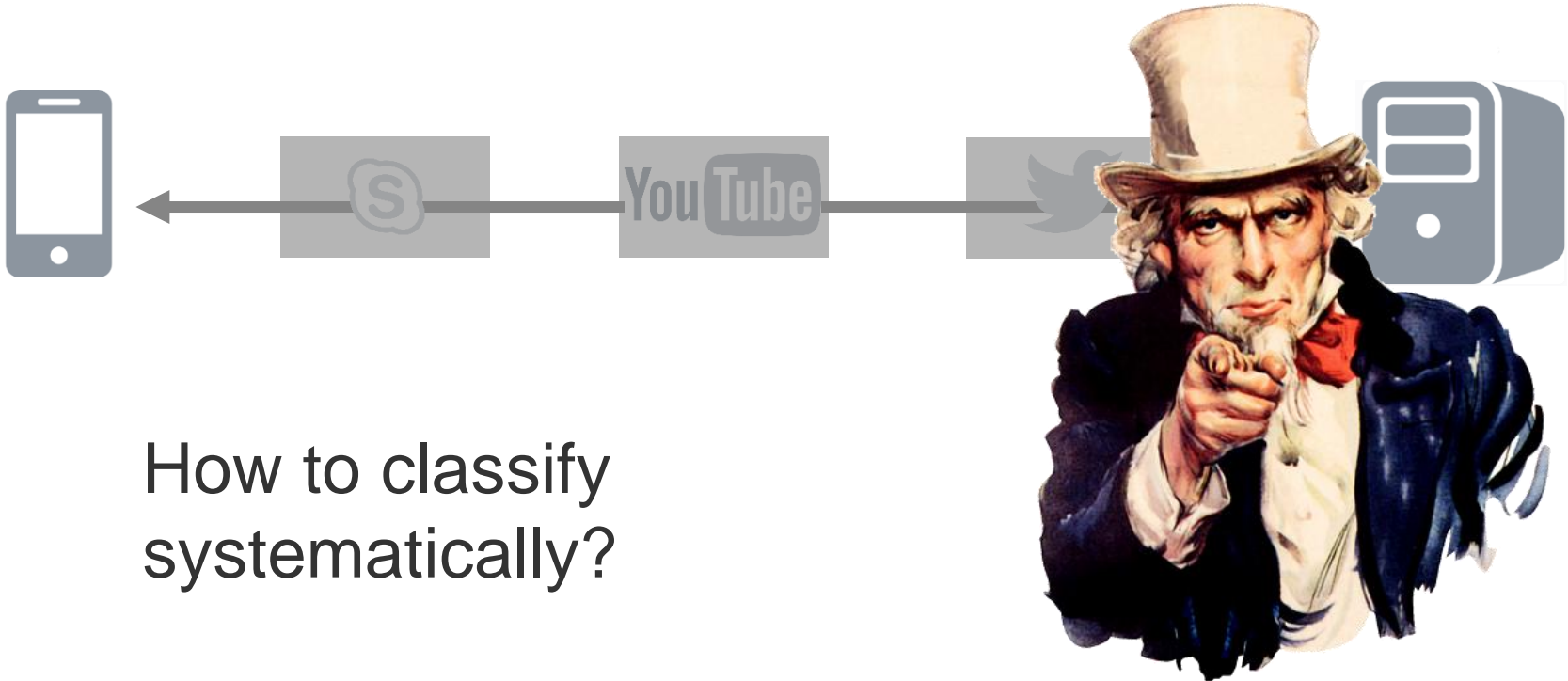
Traffic classification approaches

Port-based

Payload-based

Host behavior-based

Flow feature-based

Port-based traffic classification

```
If
   TCP/SRC or TCP/DST == 80
Then
   HTTP;
```

NW   TP   App
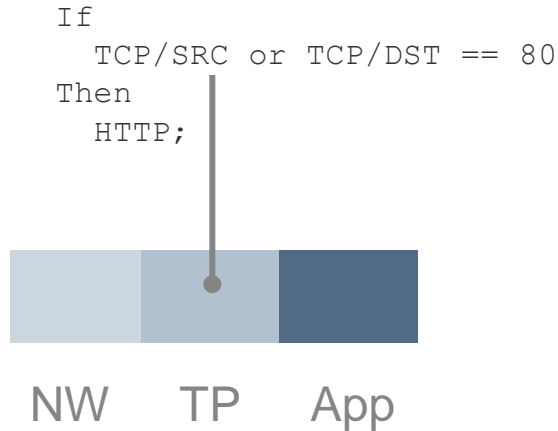
Assumption

Many applications run on fixed service ports

Advantage

Simple and fast

Drawback

Assumption holds only in some scenarios
P2P apps use random ports, apps use well-known ports to obfuscate traffic etc.

High probability of misclassification

Payload-based traffic classification (or DPI)

```
If
   GET followed by HTTP/2.0
Then
   HTTP;
```

NW   TP   App
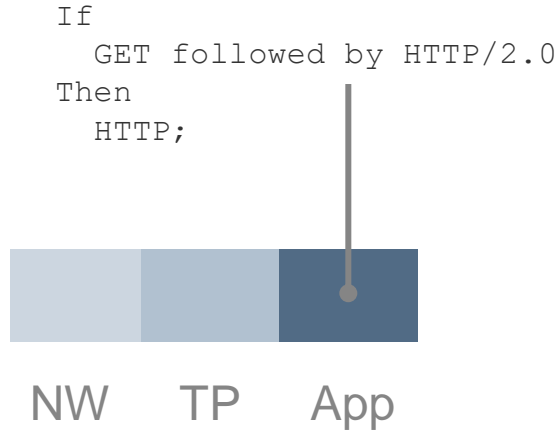
Assumption

Application layer protocol known

Advantage

Very accurate

Drawback

Signatures available only for common protocols

Challenging when traffic is encrypted

Usually needs first packet(s) of handshake

Host behavior-based traffic classification

```
If
    IP==8.8.8.8 & Port==443
Then
    DNS over HTTPS;
```

NW   TP   App
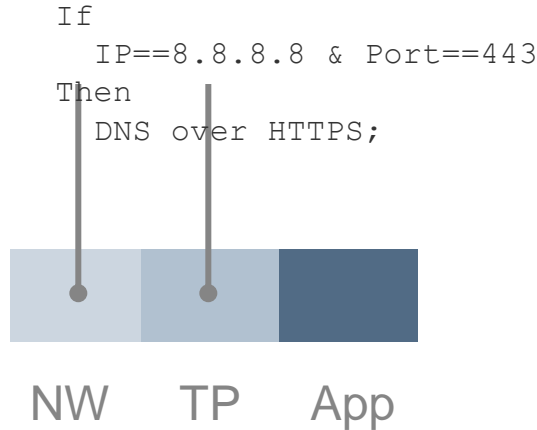
Assumption

Network interaction and host context represent the protocol

Advantage

Works well for P2P applications and encrypted traffic

Drawback

Complex profiles needed

Flow feature-based traffic classification (or DPI)

```
If
  <# of packets/s> = 50
Then
  Voice traffic;
```

NW   TP   App
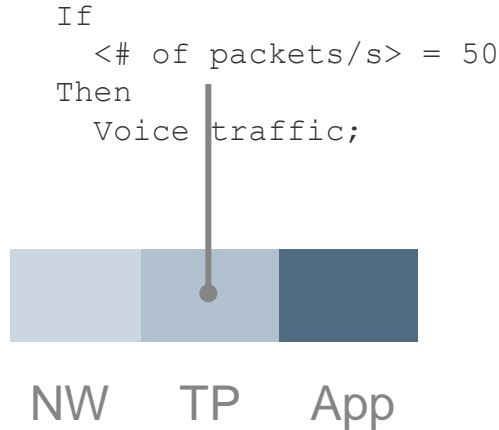
Assumption

Flow properties (average packet frequency, size etc.) describe application

Advantage

Flexible

Drawback

Needs per flow characteristics

Metrics to assess the performance of classification approaches (2)

HAW HAMBURG

Precision

Ratio of True Positives over the sum of True Positives and False Positives or the percentage of flows that are properly attributed to a given application

Recall

Ratio of True Positives over the sum of True Positives and False Negatives or the percentage of flows in an application class that are correctly identified

Metrics to assess the performance of classification approaches (2)

Example

Input
4 packets

Output
2 packets correctly identified,
1 packet incorrectly identified

Precision: 2/3

Recall: 2/4

Precision
Ratio of True Positives over the sum of True Positives and False Positives or the percentage of flows that are properly attributed to a given application

Recall
Ratio of True Positives over the sum of True Positives and False Negatives or the percentage of flows in an application class that are correctly identified

Comparison of different classification schemes

Based on seven (complete) packet traces from different sources from 2004 and 2006.

Details see: Kim et al.: "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," Proc. of ACM CoNEXT 2008.
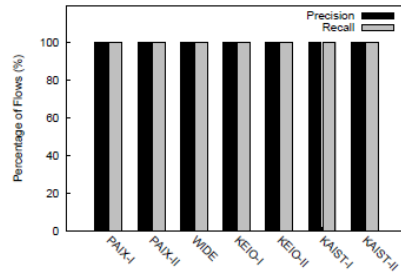
We will not focus on flow feature-based machine learning.

Port-based classification
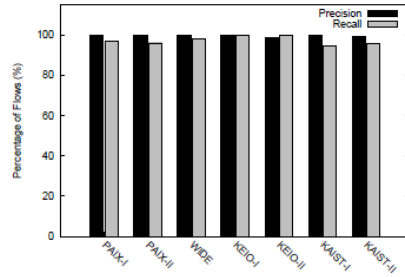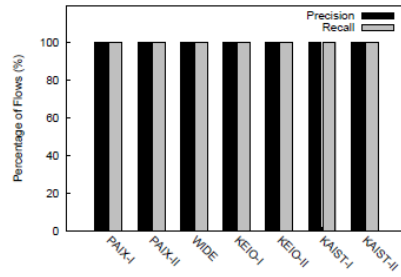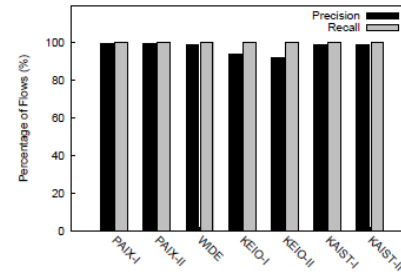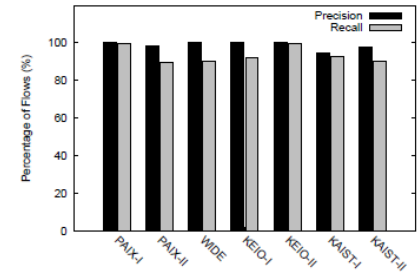
(a) WWW
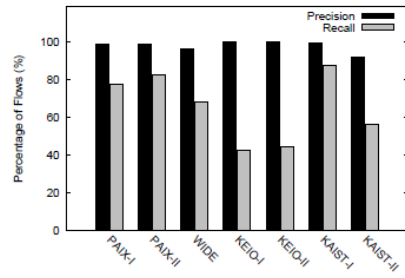
(b) DNS

(c) Mail

(d) Chat

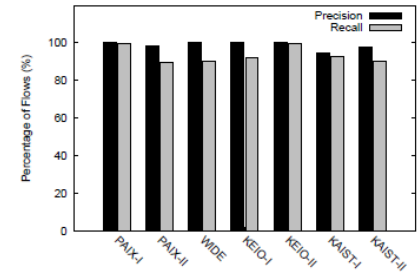(e) FTP
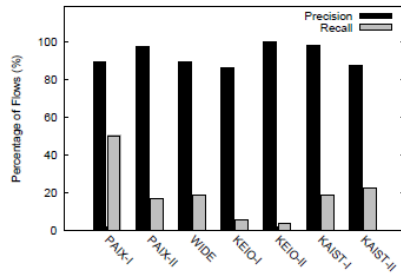
(f) P2P
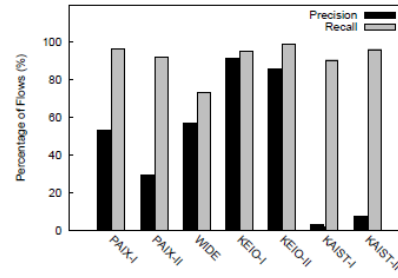
(g) Streaming
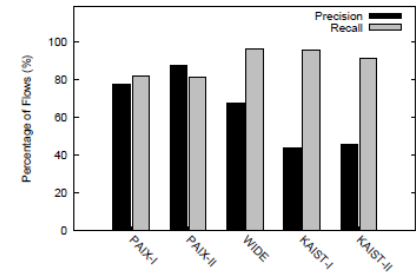
(h) Game

Port-based classification
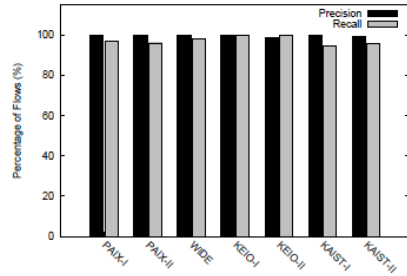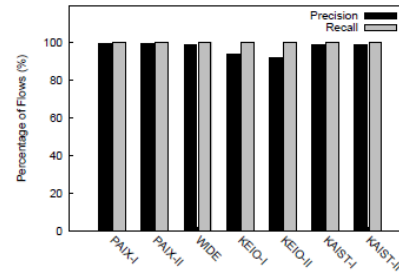
(a) WWW (b) DNS (c) Mail (d) Chat

(1) High precision of a port-based classifier implies that its default ports are seldom used by other applications

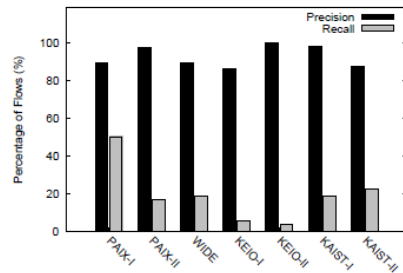(2) High recall implies that corresponding application mostly uses its default ports.

Port-based classification
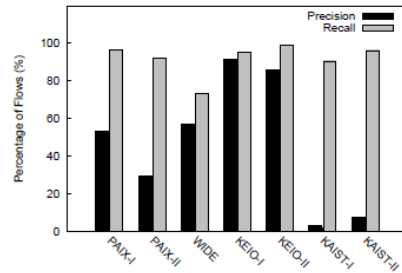
Port-based classification fails to yield accurate classification results

(1) When applications use ephemeral ports

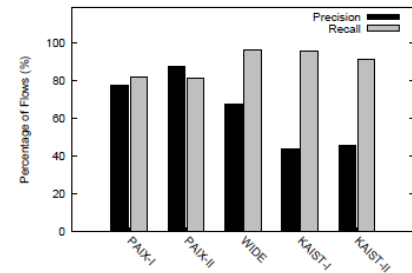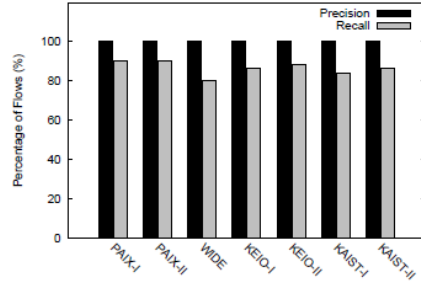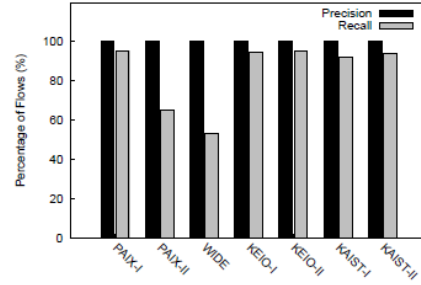(2) When default ports coincide with port masquerading
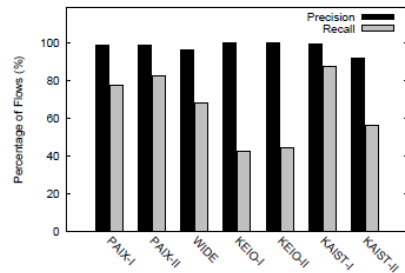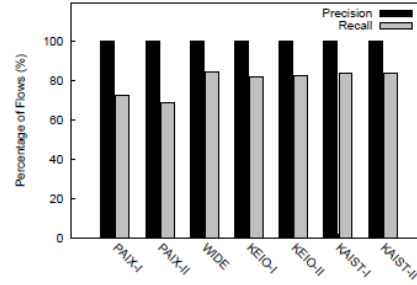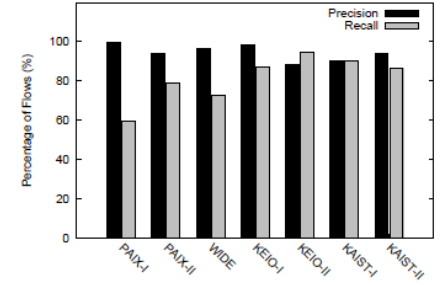

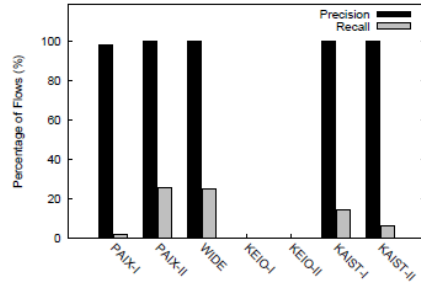(e) FTP (f) P2P (g) Streaming (h) Game
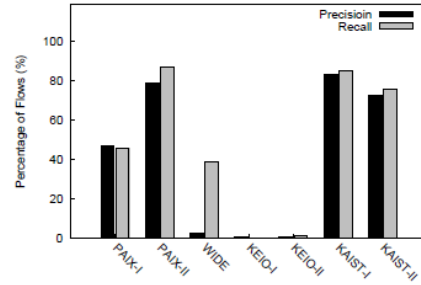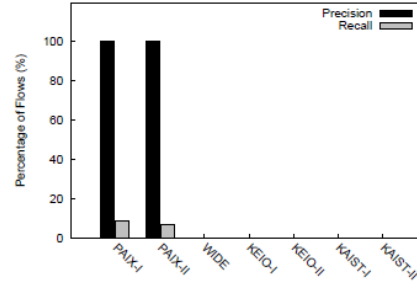
Host behavior-based classification

(a) WWW
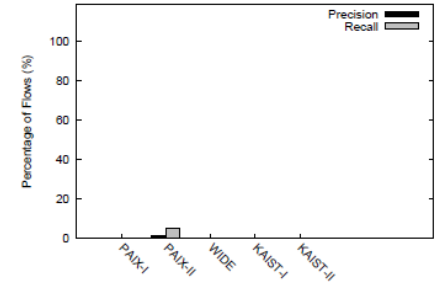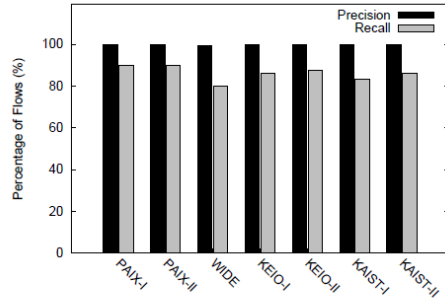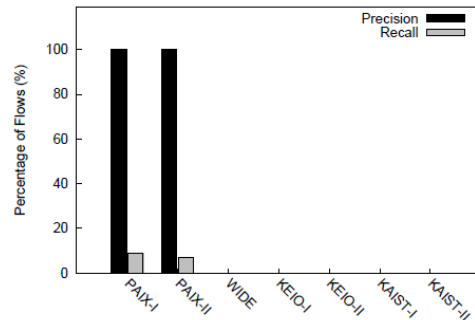
(b) DNS

(c) Mail

(d) Chat

(e) FTP

(f) P2P

(g) Streaming

(h) Game

Flow-based Classification

(a) WWW

(g) Streaming
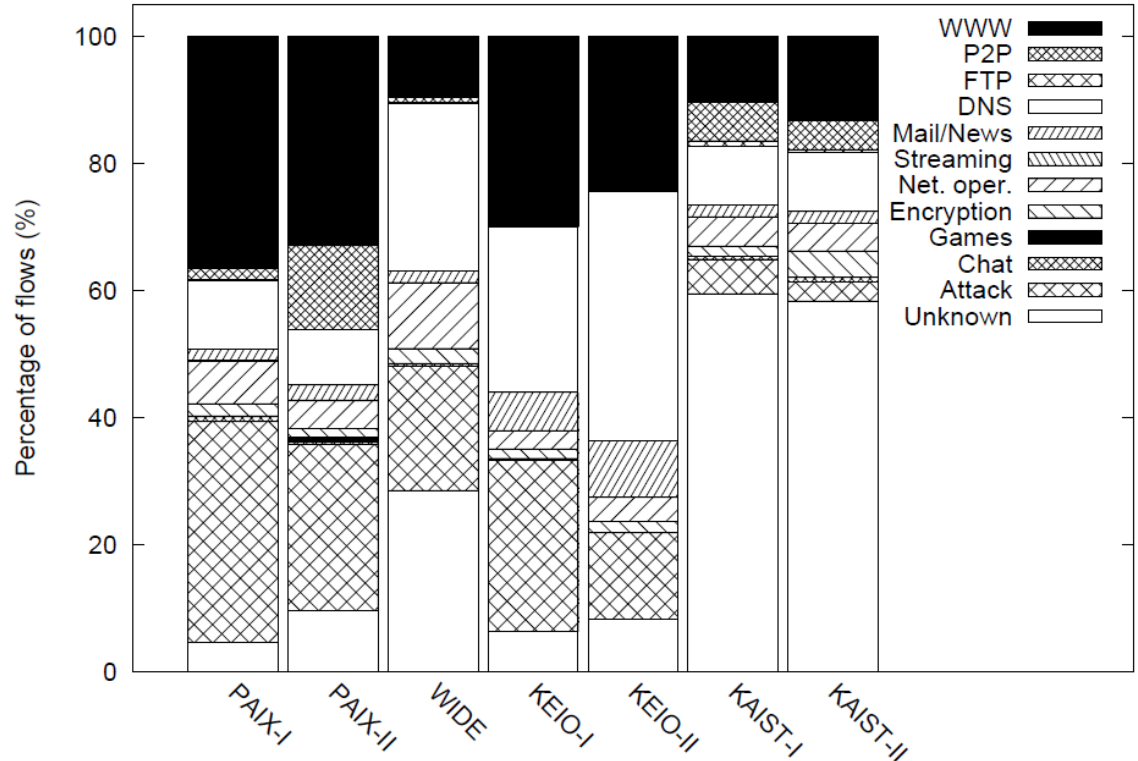
Successful classification needs (1) fine tuning and (2) traffic needs to include enough behavioral information about each host.

Best place to use such classification approach: border link of a single-homed edge network

Backbone links are *not* suitable because where (1) only a small portion of behavioral information is collectable of each host and (2) often one direction of traffic flow is missed

(g) Streaming

PAIX-I   PAIX-II   WIDE   KEIO-I   KEIO-II   KAIST-I   KAIST-II

Now, we change the observation perspective and data collection approach.

Observation point: Large European IXP

Data collection: Random packet sampling, data from 2011 – 2013

More details: Richter et al.: "Distilling the Internet's Application Mix from Packet-Sampled Traffic," Proc. of PAM 2013.

Dataset characteristics

Name	Timerange	Sampling	Packets	Bytes	IPv4 / IPv6	TCP / UDP
09-2013	2013-09-02 to 2013-09-08	1/16K	9.3B	5.9TB	99.36 / 0.63	83.7 / 16.3
12-2012	2012-12-01 to 2012-12-07	1/16K	8.5B	5.5TB	99.64 / 0.36	83.1 / 16.9
06-2012	2012-06-04 to 2012-06-10	1/16K	7.3B	4.6TB	99.80 / 0.20	80.7 / 19.3
11-2011	2011-11-28 to 2011-12-04	1/16K	6.4B	4.2TB	99.93 / 0.07	79.8 / 20.2
04-2011	2011-04-25 to 2011-05-01	1/16K	5.3B	3.5TB	99.94 / 0.06	79.2 / 20.3

Dataset characteristics

86% of sampled TCP flows: only one packet sampled
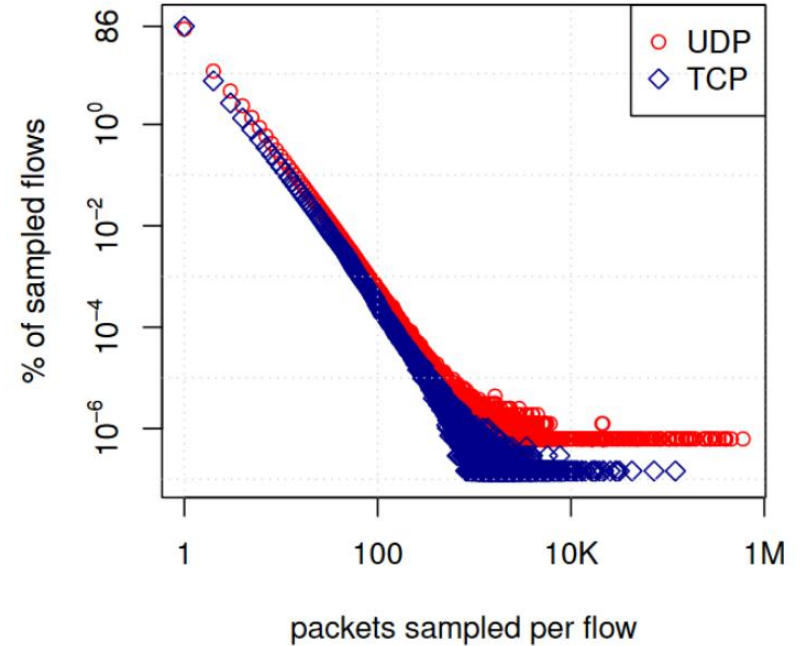
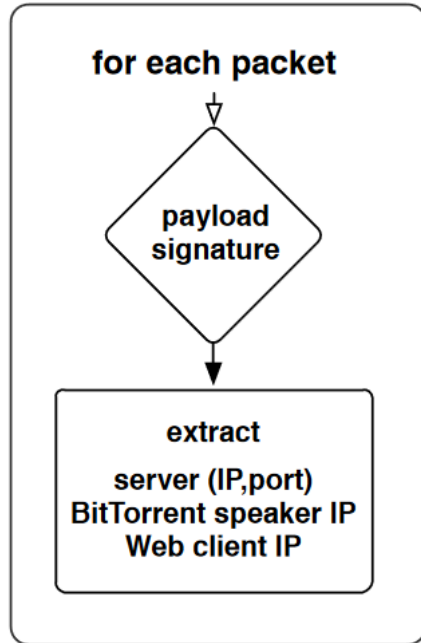(a) Samples per flow (1200s timeout).

Sampling limits

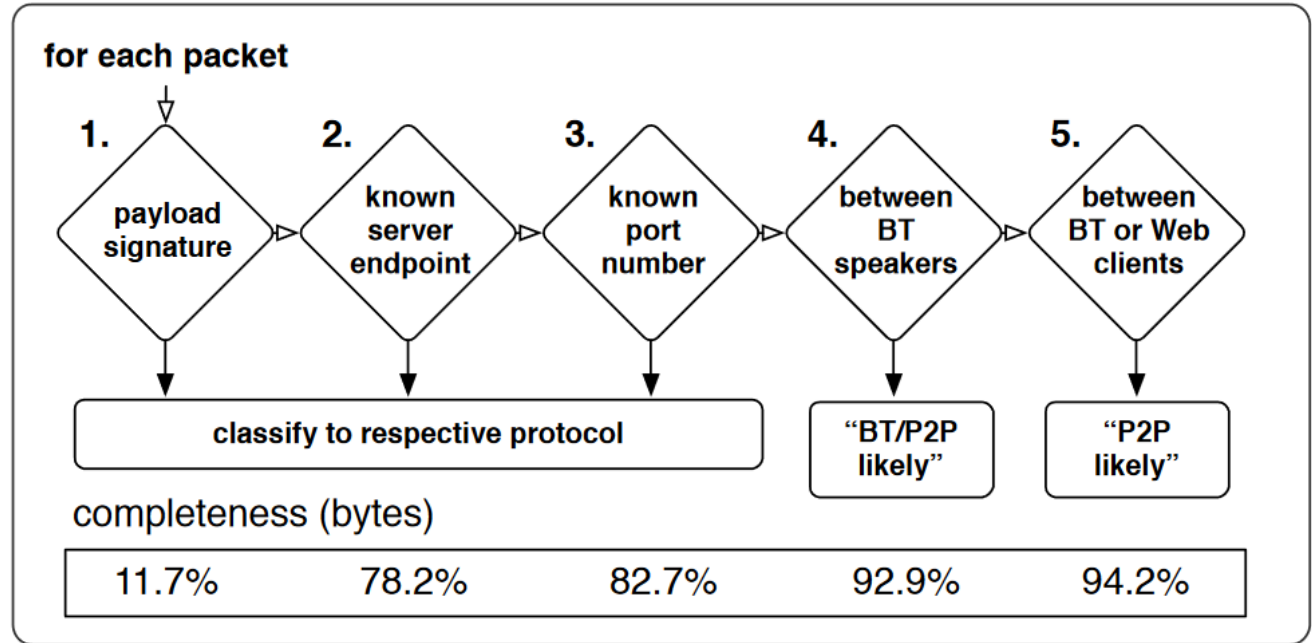Only limited amount of payload was captured (details depend on IP and TCP options)

Flow feature-based approaches not applicable
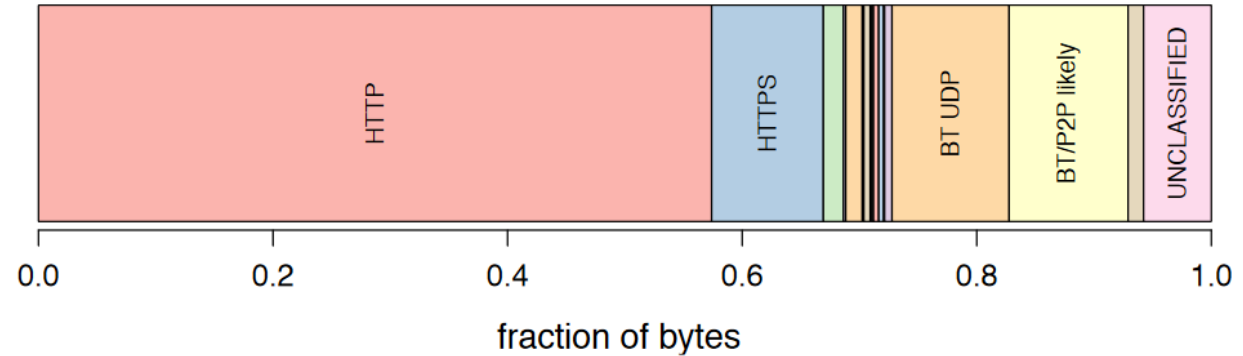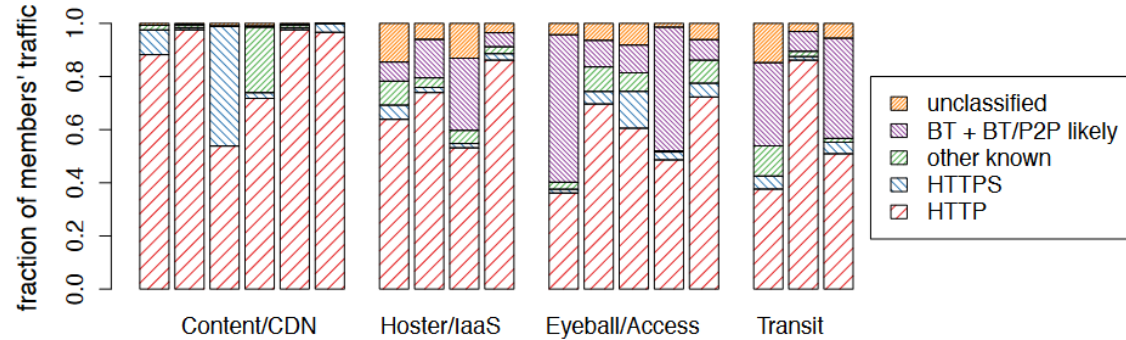
Classification pipeline

Application mix: Aggregate

fraction of bytes

- HTTP(S) dominates ~67%
- other applications (e.g., RTMP, mail, news) ~6%
- BitTorrent/BT/P2P likely ~22%
- unclassified ~5%

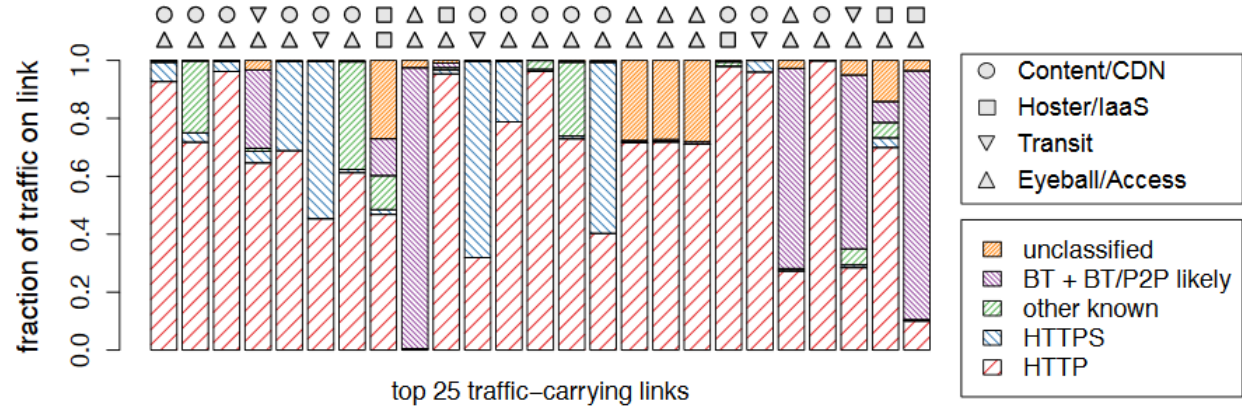# Application mix: Per network type

- Content/CDN almost 100% HTTP

- HTTPS increase driven by only a few networks

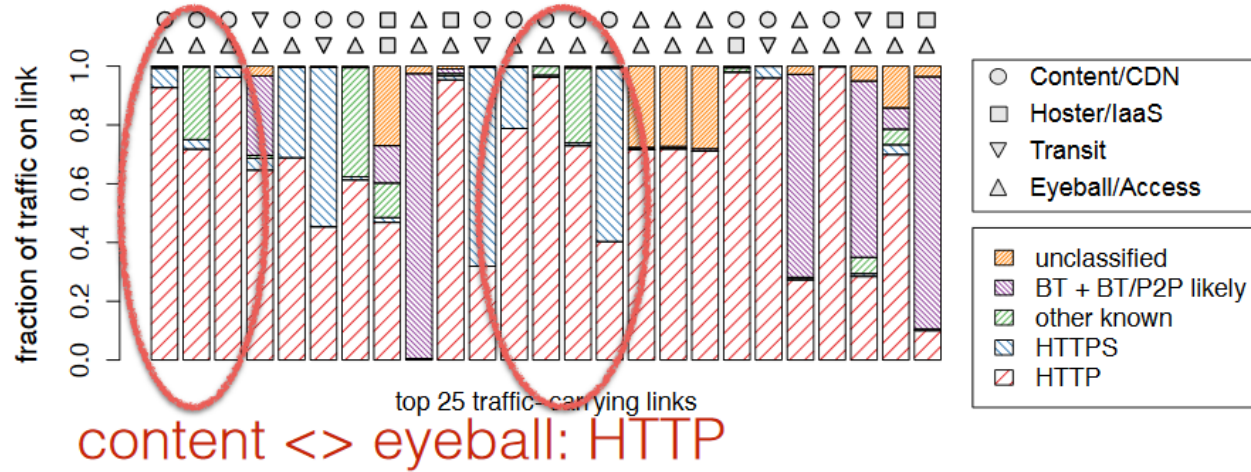- P2P not only between Eyeballs! Hoster/IaaS too!

Dissecting per network shows a different appmix!

Application mix: Per link

- Aggregate mix by no means representative of single link
- Many links just have one dominant protocol
- The business type of the ASes gives hints on app mix

Application mix: Per link (content – eyeball)

content <> eyeball: HTTP

- Aggregate mix by no means representative of single link
- Many links just have one dominant protocol
- The business type of the ASes gives hints on app mix

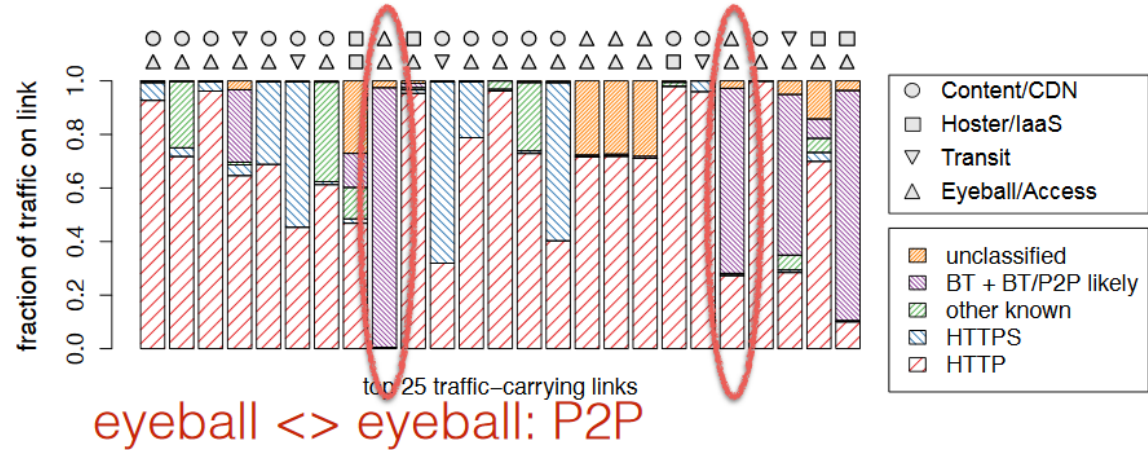# Application mix: Per link (eyeball – eyeball)

eyeball <> eyeball: P2P

- Aggregate mix by no means representative of single link
- Many links just have one dominant protocol
- The business type of the ASes gives hints on app mix

Application mix: Per link (hoster/IaaS)

top 25 traffic-carrying links

Content/CDN, Hoster/IaaS, Transit, Eyeball/Access

unclassified, BT + BT/P2P likely, other known, HTTPS, HTTP
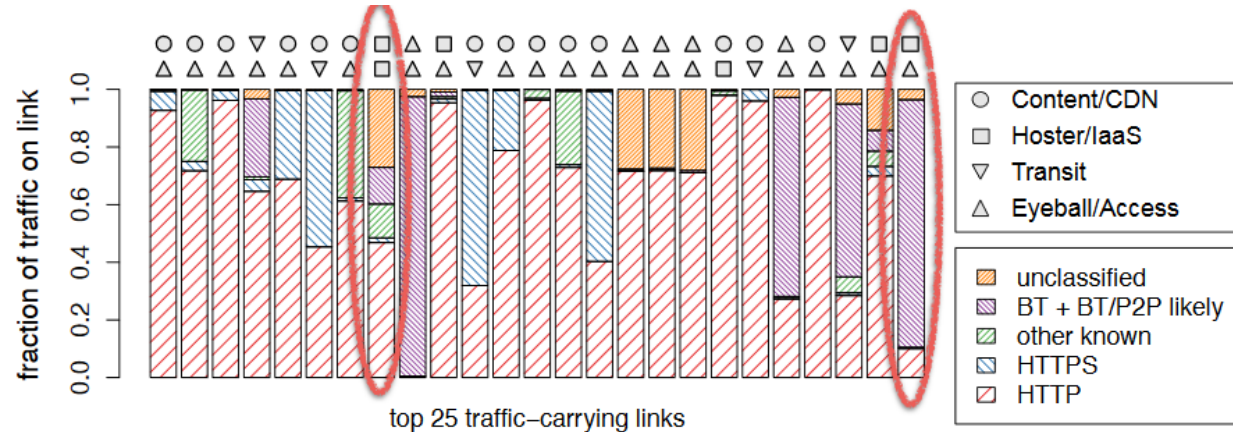
hoster/IaaS: diverse application mix

- Aggregate mix by no means representative of single link
- Many links just have one dominant protocol
- The business type of the ASes gives hints on app mix

Insights

A stateful approach can overcome limitations of random packet sampling

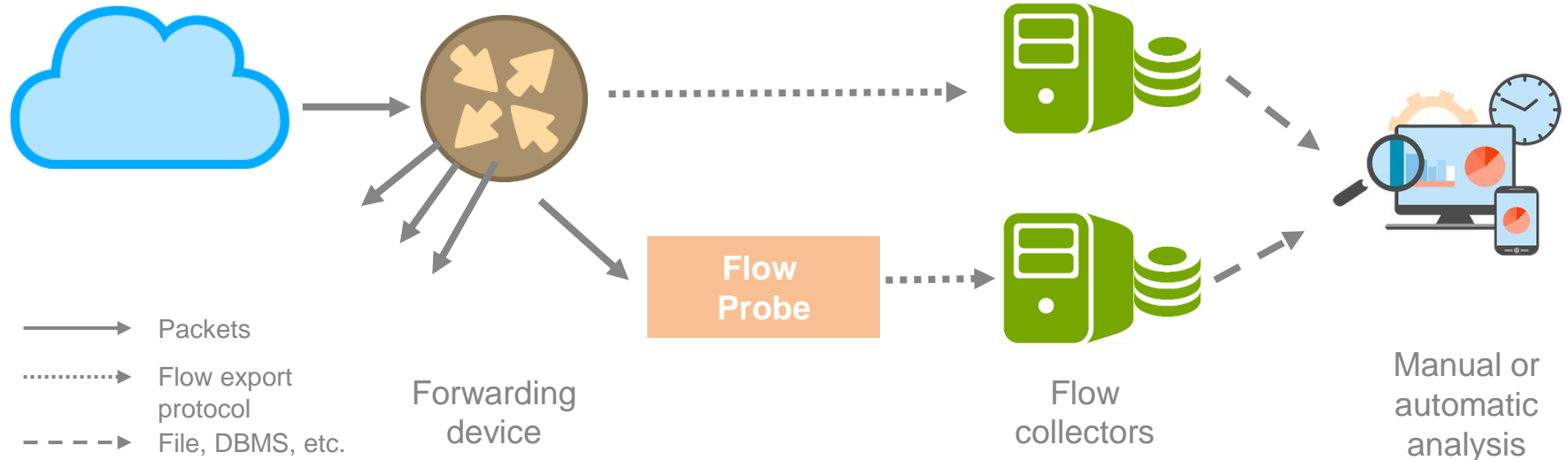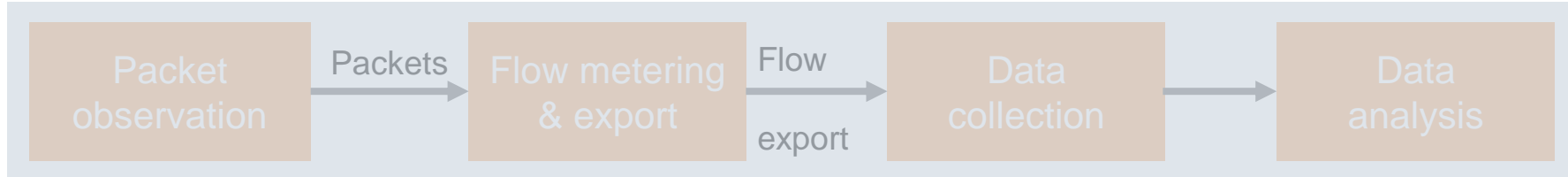Dissecting network types reveals different application mix

Measuring Packets in Context
MONITORING FLOWS

Typical flow monitoring setups

| Packet observation | → Packets → | Flow metering & export | → Flow export → | Data collection | → | Data analysis |

Typical flow monitoring setups

Packet observation → Packets → Flow metering & export → Flow export → Data collection → Data analysis

Packets

Flow export protocol

File, DBMS, etc.

Forwarding device
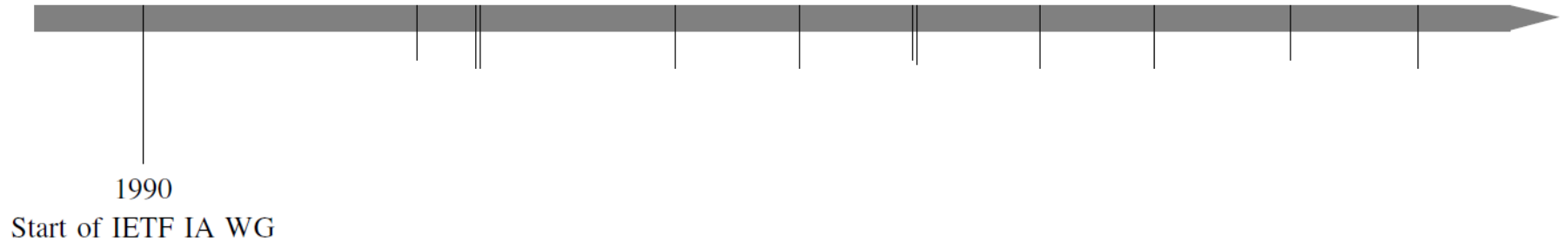
Flow Probe

Flow collectors

Manual or automatic analysis

Requirements

Vendor independent

Support different deployments

Handle large data

Evolution of flow export technologies and protocols
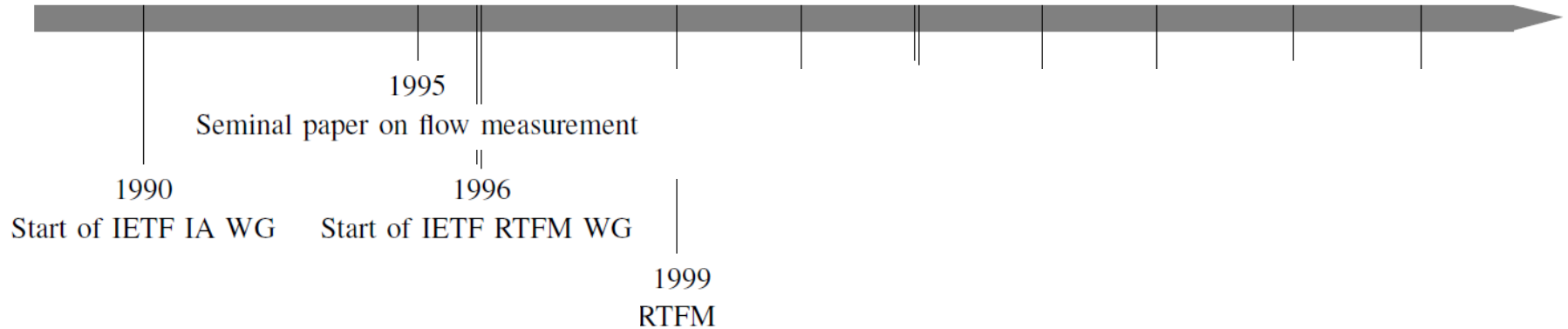

1990
Start of IETF IA WG

Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

1995
Seminal paper on flow measurement

1990
Start of IETF IA WG

1996
Start of IETF RTFM WG
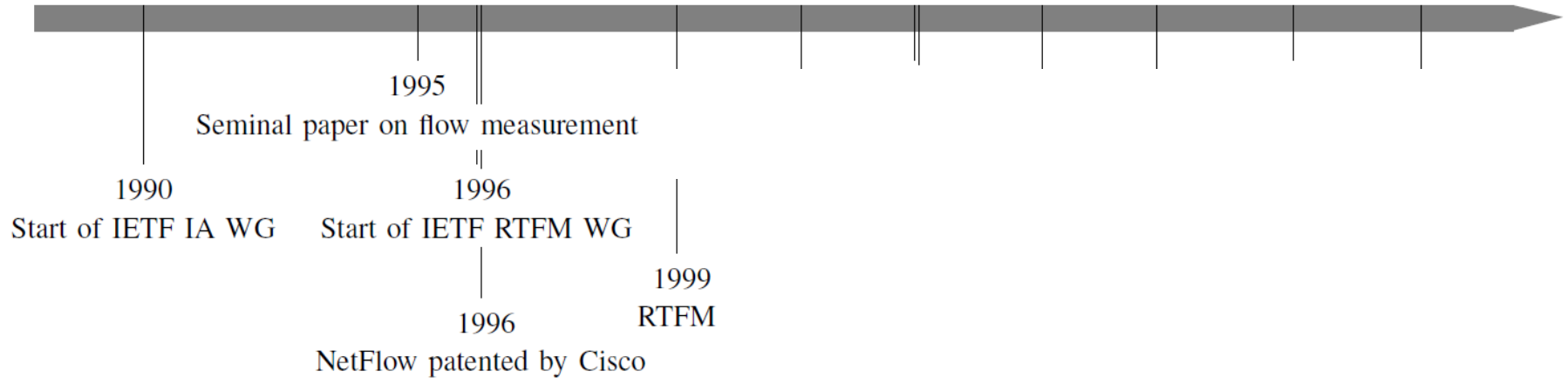
1996
NetFlow patented by Cisco

1999
RTFM

Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

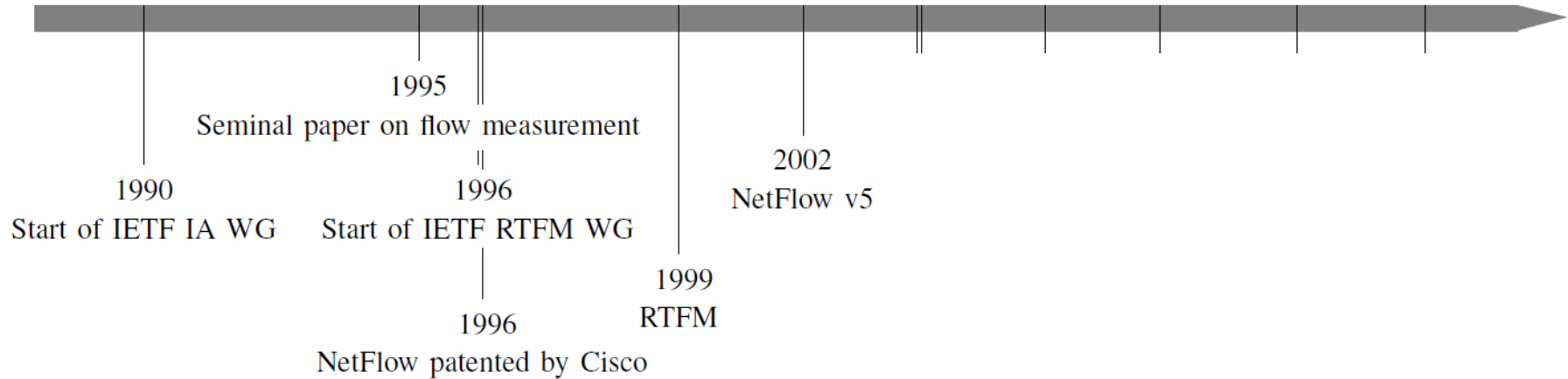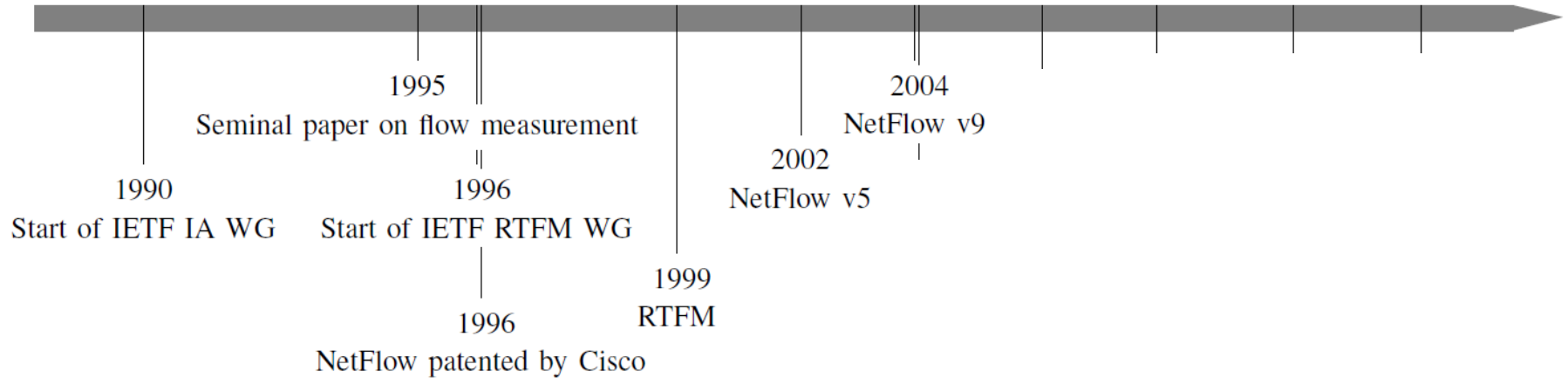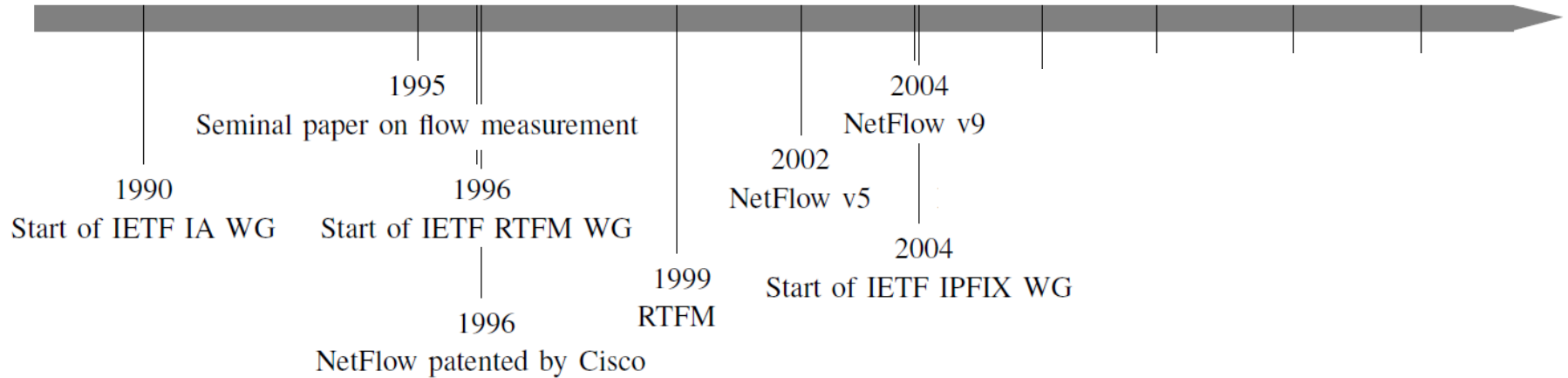# Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

Timeline of flow export technologies and protocols:

- **1990** — Start of IETF IA WG
- **1995** — Seminal paper on flow measurement
- **1996** — Start of IETF RTFM WG
- **1996** — NetFlow patented by Cisco
- **1999** — RTFM
- **2002** — NetFlow v5
- **2004** — NetFlow v9
- **2004** — Start of IETF IPFIX WG
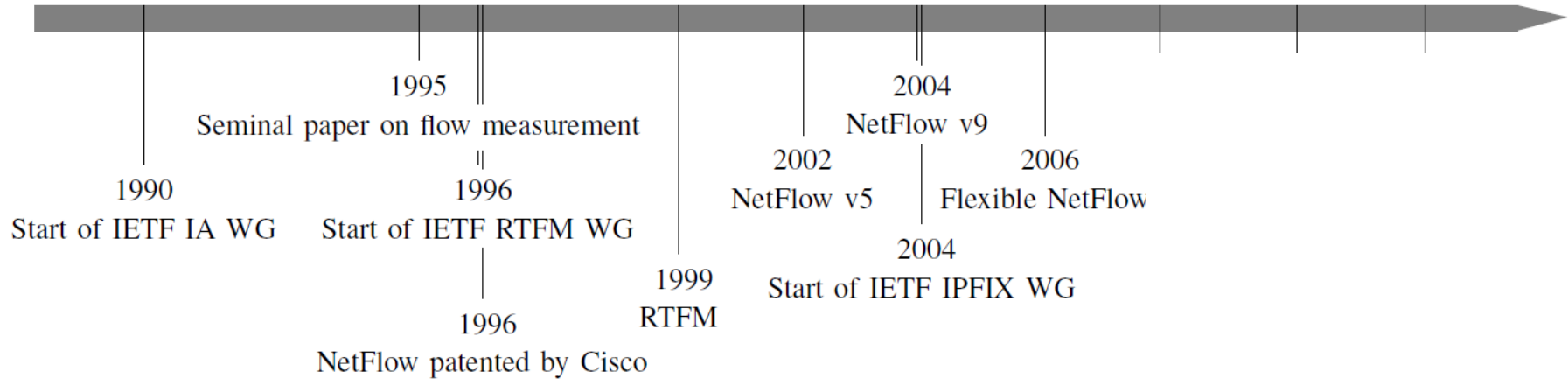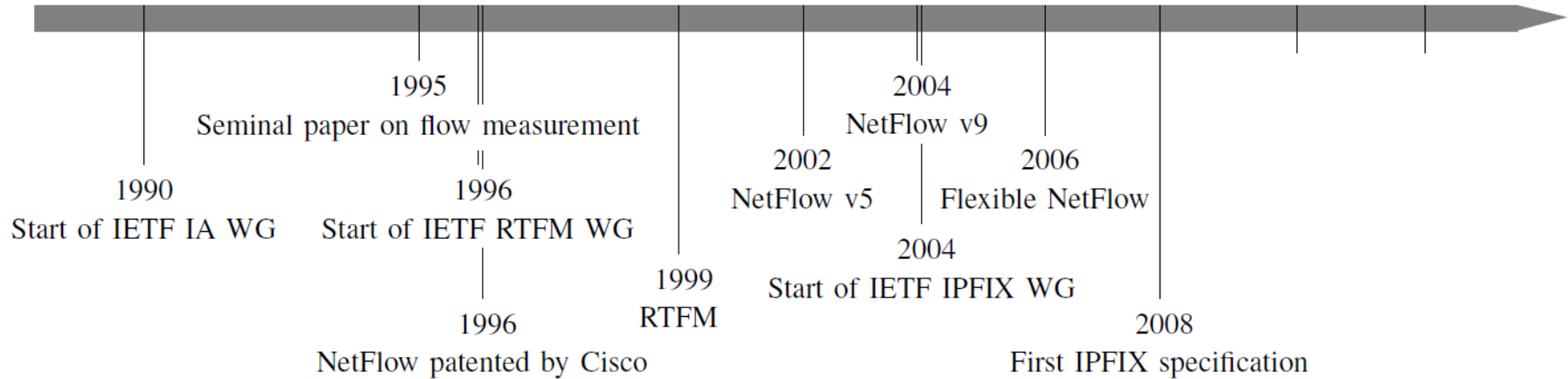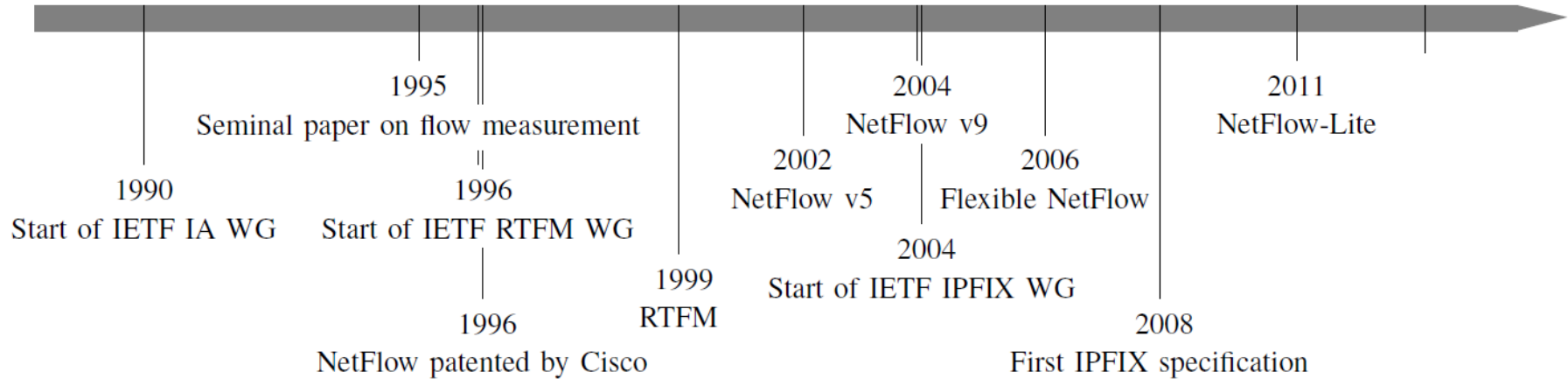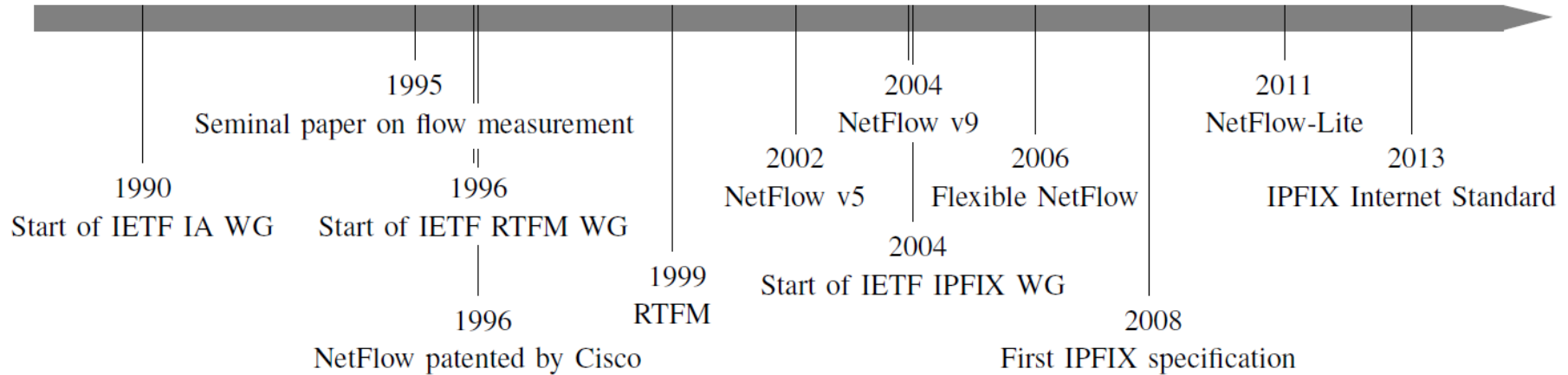- **2006** — Flexible NetFlow

Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

Evolution of flow export technologies and protocols

Related but not the same: sFlow

Industry standard

Integrated into many packet forwarding devices
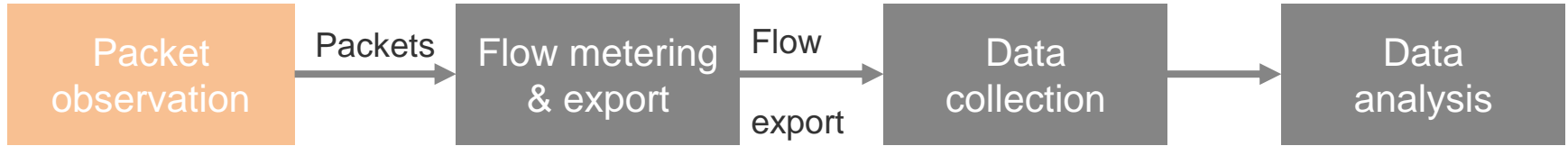
Samples packets and interface counters

Architecturally similar to NetFlow and IPFIX but it is packet-oriented

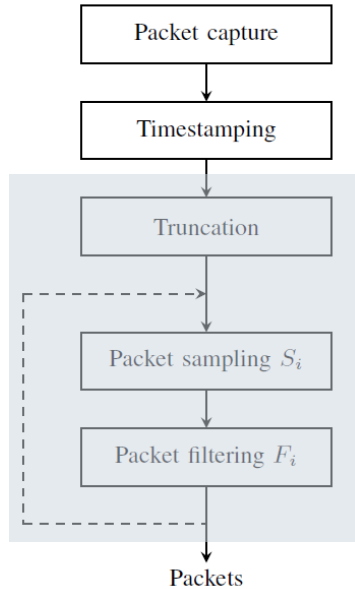Closer related to packet sampling techniques

Typical flow monitoring setups

```
┌─────────────────┐  Packets  ┌─────────────────┐  Flow    ┌─────────────────┐        ┌─────────────────┐
│     Packet      │ ────────► │  Flow metering  │ ───────► │      Data       │ ─────► │      Data       │
│   observation   │           │    & export     │  export  │   collection    │        │    analysis     │
└─────────────────┘           └─────────────────┘          └─────────────────┘        └─────────────────┘
```

Typical flow monitoring setups

Packet observation

Packet capture → Timestamping → Truncation → Packet sampling S_i → Packet filtering F_i → Packets

Truncation selects only those bytes that fit into a preconfigured snapshot length

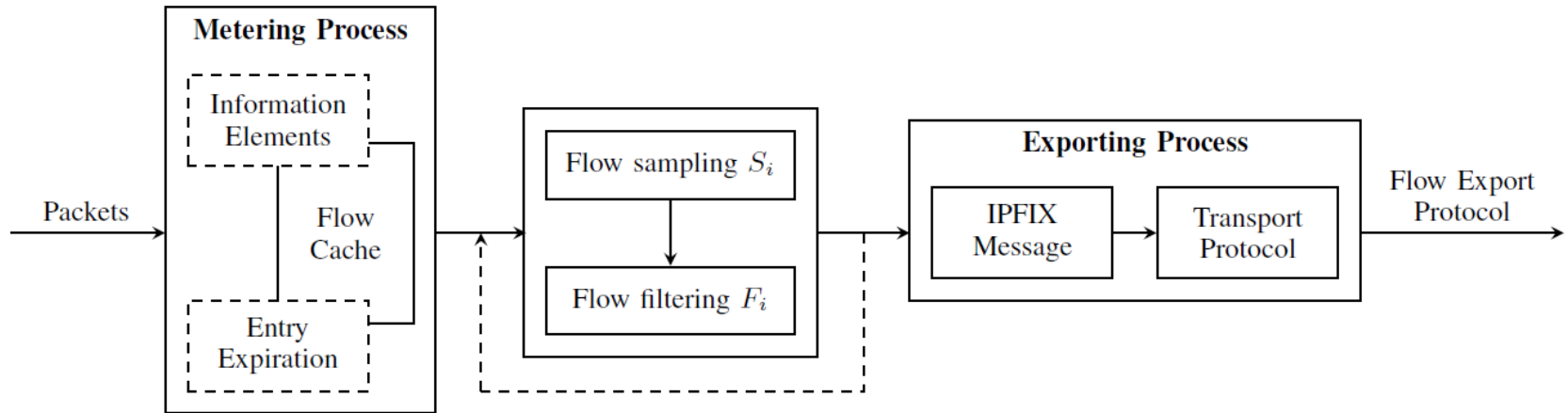Traffic capture can be implemented in in-line mode or mirroring mode

Software tools, e.g., libpcap
Network stacks are made for general-purpose networking, leading to suboptimal performance; improvements available (e.g., PF_RING)

Typical flow monitoring setups

```
┌─────────────┐  Packets  ┌─────────────┐  Flow    ┌─────────────┐         ┌─────────────┐
│   Packet    │ ────────→ │Flow metering│ ───────→ │    Data     │ ──────→ │    Data     │
│ observation │           │  & export   │  export  │ collection  │         │  analysis   │
└─────────────┘           └─────────────┘          └─────────────┘         └─────────────┘
```

Flow metering and export

Current Standard
IP FLOW INFORMATION EXPORT (IPFIX)

Information Elements (IE) describe the exported data in IPFIX

ID	Name	Description
152	flowStartMilliseconds	Timestamp of the flow's first packet.
153	flowEndMilliseconds	Timestamp of the flow's last packet.
8	sourceIPv4Address	IPv4 source address in the packet header.
12	destinationIPv4Address	IPv4 destination address in the packet header.
7	sourceTransportPort	Source port in the transport header.
11	destinationTransportPort	Destination port in the transport header.
4	protocolIdentifier	IP protocol number in the packet header.
2	packetDeltaCount	Number of packets for the flow.
1	octetDeltaCount	Number of octets for the flow.

Information Elements (IE) describe the exported data in IPFIX

ID	Name	Description
152	flowStartMilliseconds	Timestamp of the flow
153	flowEndMilliseconds	Timestamp of the flow
8	sourceIPv4Address	IPv4 source address header.
12	destinationIPv4Address	IPv4 destination address packet head
7	sourceTransportPort	Source port in the tra
11	destinationTransportPort	Destination port in header.
4	protocolIdentifier	IP protocol number header.
2	packetDeltaCount	Number of packets
1	octetDeltaCount	Number of octets

Maintained by IANA
Enterprise-specific IEs possible

Can be defined for any layer
But common focus on network and transport layer

Configuration of metering process not standardized

Allows for templates, variable-length encoding, and structured data

Flow Caches store information about active network traffic flows

Entries are composed of IEs

Flow key defines whether a packet defines a new flow or not

Flow caches may differ in cache layout

Cope with IE flexibility

… type

e.g., immediate caches, permanent cache

… and size

Cache entries usually require expiration timers

Cache entries are maintained in the flow cache until the corresponding flows are considered terminated

Active timeout, flow has been active for a specified period of time (120s – 30 min); cache entries are not removed but counters are reset

Idle timeout, no packets belonging to a flow have been observed (15s – 5 min)

Resource constraints, special heuristics

Natural expiration, TCP packet with a FIN or RST flag; depends on the exporter implementation

Idle and active timeout have impact on total # of recorded and exported flows

Longer timeout values result in higher aggregation of packets into flow records

Pros: Reduces load on flow collector

Cons: takes longer before a flow becomes visible in the data analysis

Experimental evaluation
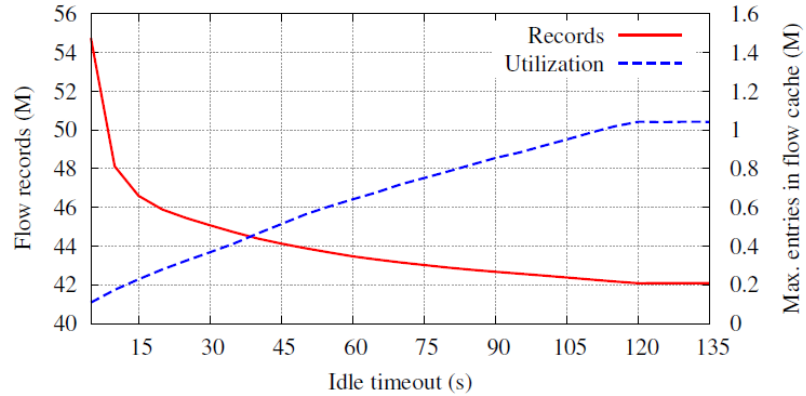
(a) Varying idle timeout values, active timeout = 120 seconds

Experimental evaluation
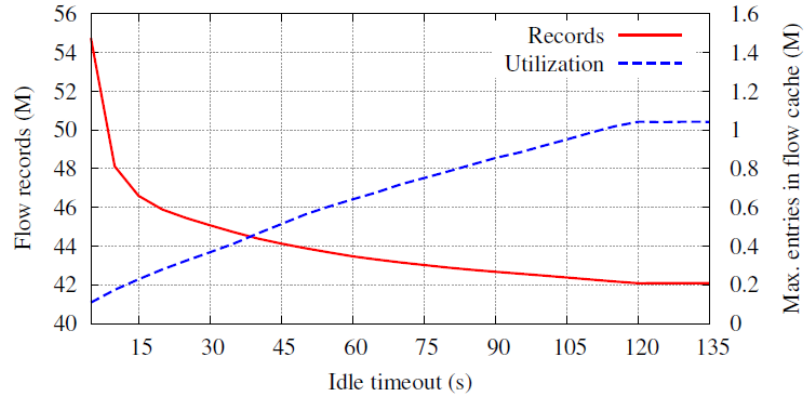
(a) Varying idle timeout values, active timeout = 120 seconds
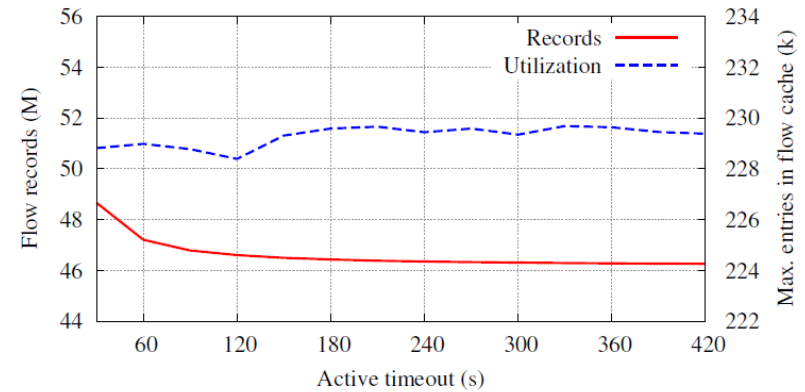
(b) Varying active timeout values, idle timeout = 15 seconds

IPFIX messages [RFC 7011]

Template Set describes the layout of Data Records

Data Set carries exported Data Records (i.e., flow records)

Options Template Set includes meta-data

Version number (2)		Length (2)	
Export time (4)			
Sequence number (4)			
Observation domain ID (4)			
Set ID (2)		Length (2)	
Record 1			
Record 2			
Record n			

Set
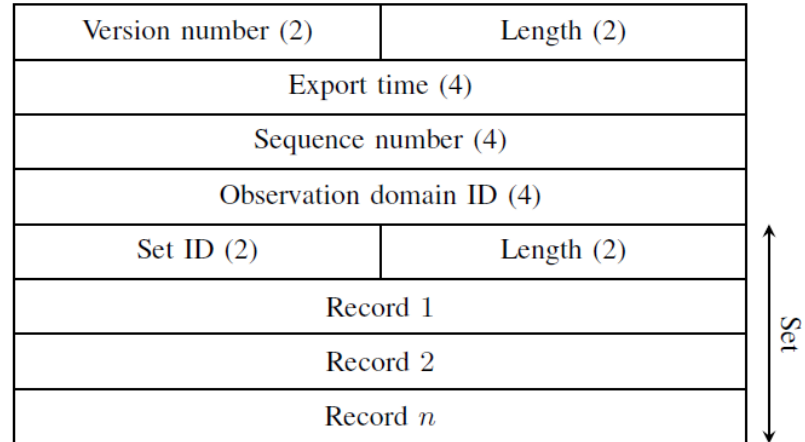
(simplified)

IPFIX messages [RFC 7011]

Template	
Template ID = 257	Length = 9 IEs
flowStartMilliseconds (ID = 152)	
flowEndMilliseconds (ID = 153)	
sourceIPv4Address (ID = 8)	
destinationIPv4Address (ID = 12)	
sourceTransportPort (ID = 7)	
destinationTransportPort (ID = 11)	
protocolIdentifier (ID = 4)	
packetDeltaCount (ID = 2)	
octetDeltaCount (ID = 1)	

Data Record
Set Header (Set ID = 257)
Record 1
Record 2
Record n

Flow Record	
flowStartMilliseconds = 2013-07-28 21:09:07.170	
flowEndMilliseconds = 2013-07-28 21:10:33.785	
sourceIPv4Address = 192.168.1.2	
destinationIPv4Address = 192.168.1.254	
sourceTransportPort = 9469	dstTransportPort[6] = 80
protocolIdentifier = 6	
packetDeltaCount = 17	
octetDeltaCount = 3329	

Version number (2)	Length (2)
Export time (4)	
Sequence number (4)	
Observation domain ID (4)	
Set ID (2)	Length (2)
Record 1	
Record 2	
Record n	

(simplified)

Which transport protocol to export flows?

Problems:

TCP - head-of-line blocking

UDP – unreliable, lack of
congestion control

SCTP – missing deployment

Potentials of SCTP:

- message oriented w/ boundaries

- multiple streams per connection

	SCTP	TCP	UDP
Congestion awareness	+	+	−
Deployability	−	+	+
Graceful degradation	+	−	−
Reliability	+	+	−

Typical flow monitoring setups

| Packet observation | →Packets→ | Flow metering & export | →Flow export→ | Data collection | → | Data analysis |
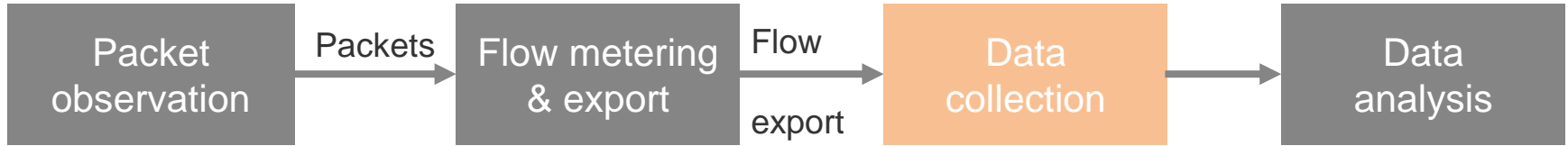
Storage formats

	Flat files	Row-oriented databases	Column-oriented databases
Disk space	+	−	0
Insertion performance	+	−	0
Portability	− (binary), + (text)	−	−
Query flexibility	−	+	+
Query performance	+ (binary), − (text)	−	+

Data anonymization

Even though flow data include no or very limited payload, individuals can be identified and tracked

Anonymization technique depends on the use case

Complete random, prefix-preserving, prefix anonymized

Typical flow monitoring setups

```
┌──────────────┐   Packets  ┌──────────────┐  Flow    ┌──────────────┐          ┌──────────────┐
│   Packet     │ ─────────▶ │ Flow metering│ ───────▶ │    Data      │ ───────▶ │    Data      │
│ observation  │            │  & export    │  export  │  collection  │          │   analysis   │
└──────────────┘            └──────────────┘          └──────────────┘          └──────────────┘
```
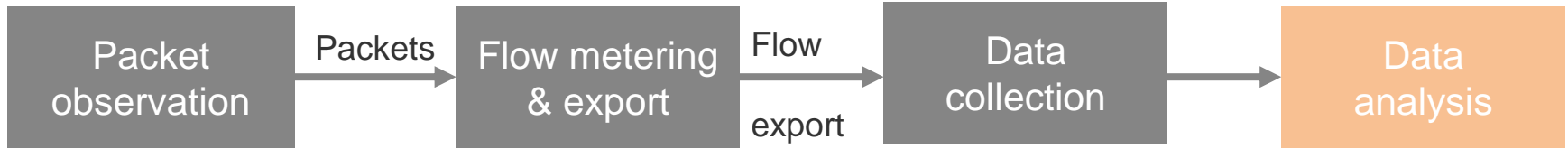
Example: Threat detection SSH

Frequently-used target of dictionary attacks

How would you detect those attacks, albeit SSH is encrypted?

Example: Threat detection SSH

Many credentials are tested
subsequently

SSH daemons close connections after a
fixed number of login attempts
Consequently: Many TCP connections
with similar size in terms of packets

Example: Threat detection SSH
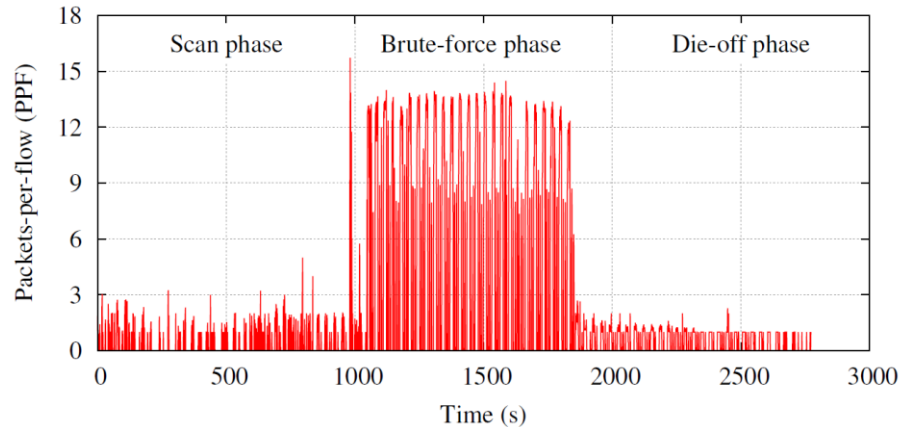
Many credentials are tested subsequently

SSH daemons close connections after a fixed number of login attempts
Consequently: Many TCP connections with similar size in terms of packets

Common pitfalls
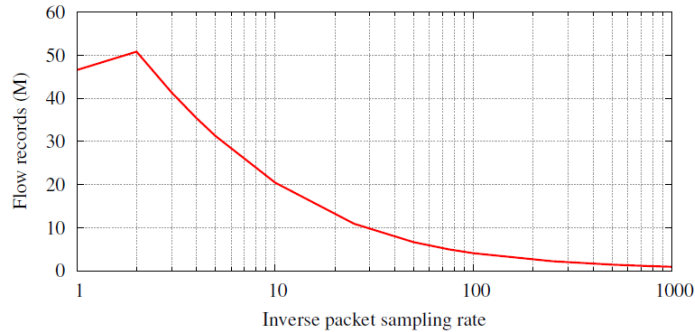
EXPORT VOLUMES FOR THE UT DATASET (2.1 TB)

Sampling rate	Protocol	Export packets / bytes
1:1	NetFlow v5	1.4 M / 2.1 G
1:1	NetFlow v9	3.5 M / 2.5 G
1:10		1.6 M / 1.1 G
1:100		314.9 k / 222.5 M
1:1000		72.2 k / 49.5 M
1:1	IPFIX	4.3 M / 3.0 G

Flow Exporter overload
Flow cache may exceed limits (check loss statistics, adapt timeouts, apply packet sampling)

Transport overhead

Flow Collector overload

Flow data artifacts (timing, data loss, inaccuracies)

Literature

R. Hofstede *et al.*, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037-2064, 2014.
https://dx.doi.org/10.1109/COMST.2014.2321898

Listening out of the Dark
NETWORK TELESCOPES

Concept of a Network Telescope

- Unused network address space (darkspace)
  - Absence of wanted/legitimate traffic
- Records all incoming traffic
  - None of the traffic relates to the operation of the telescope
- Viable method to measure and analyze Internet Background Radiation (IBR)
  - The fraction of Internet traffic that serves no primary Internet purpose – the amount is huge!

Traffic at a Network Telescope

Internet Background Radiation:

- Direct path transmission – scanners, attackers

- Backscatter – replies to randomly spoofed (attack) packets

- Rich source of studying Internet anomalies, e.g., attacks, outages, spoofing, botnets, censorship, …
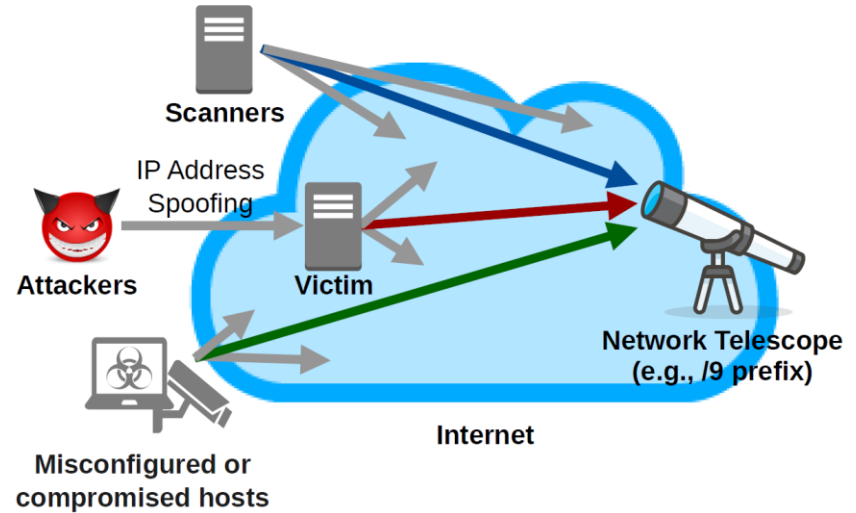
The UCSD/CAIDA Network Telescope

The UCSD Telescope is the largest IPv4 network telescope and a unique infrastructure for network researchers

- 1981 – Amateur Packet Radio Network (AMPRnet) received a /8 IP block
- UCSD agreed to provide upstream and support
- Since about 25 years, CAIDA operates the telescope and performs leading research
- Numerous researchers published results based on the telescope data over the last 20 years

Hilbert Curve of the UCSD/CAIDA Telescope

- Telescope: dark pixels
- Blue pixels: ARDC members w/ UCSD upstream
- Green pixels: Leased addresses to ARDC regions separately announced
- Red pixels: potential hijacks

Interactive View:
https://netd-tud.github.io/hilby/

Sold in 2019
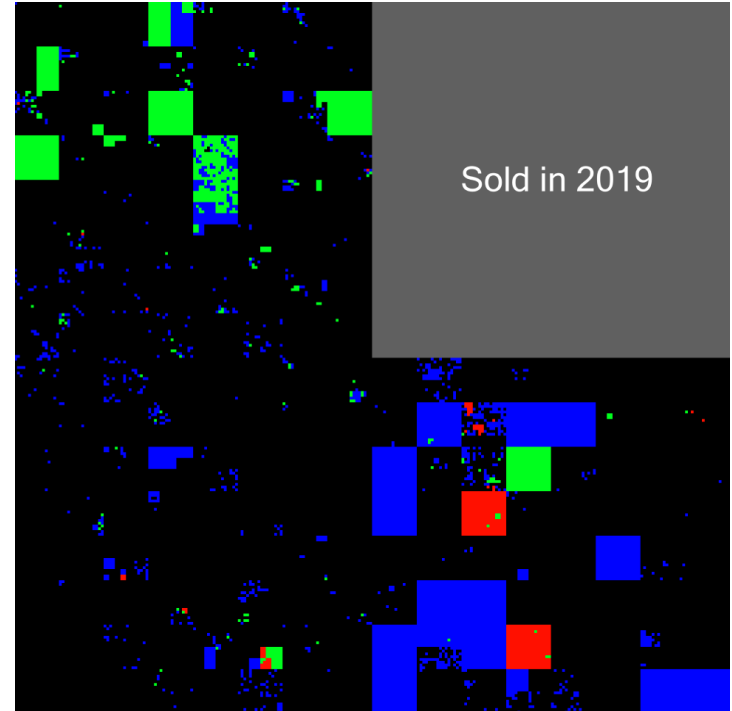
Literature

Alexander Männel, Jonas Mücke, kc Claffy, Max Gao, Ricky K. P. Mok, Marcin Nawrocki, Thomas C. Schmidt, Matthias Wählisch, **Lessons Learned from Operating a Large Network Telescope**,
In: *Proc. of ACM Special Interest Group on Data Communication (SIGCOMM),* ACM : New York, 2025.

Lessons learned from operating a large network telescope

Alexander Männel
alexander.maennel@tu-dresden.de
TU Dresden
Germany

Jonas Mücke
jonas.muecke@tu-dresden.de
TU Dresden
Germany

kc claffy
kc@caida.org
CAIDA/UC San Diego
USA

Marcin Nawrocki
marcin.nawrocki@netscout.com
NETSCOUT
USA

Thomas C. Schmidt
t.schmidt@haw-hamburg.de
HAW Hamburg
Germany

Matthias Wählisch
m.waehlisch@tu-dresden.de
TU Dresden
Germany

Abstract

Network telescopes (aka *darknets*) collect unsolicited Internet traffic (aka *Internet background radiation* or *IBR*), which includes benign and malicious scanning as well as artifacts of spoofed denial-of-service attacks and misconfigured software and networks. Analysis of this traffic has revealed macroscopic insights into security-related events and global network dynamics such as outages. Operating a large-scale network telescope is challenging but often taken for granted, unlike measurement infrastructures in physics. We offer the first study documenting our experiences operating the UCSD Network Telescope, the largest and longest-operating network telescope supporting scientific research. We provide background on the history of the telescope, and focus on increasing operational challenges as the underlying network evolves. We develop and apply techniques to leverage third-party scanning activity to validate the integrity of the data, and to discover misconfigurations in the instrumentation. These insights are crucial for understanding measurement results, which we illustrate using concrete examples. We discuss how our findings generalize to support the expanding ecosystem of other passive techniques, such as honeypots, to track security phenomena.

enabled security studies, e.g., characterizing distributed denial of service attacks (DDoS) [12, 13, 43, 78], network censorship [28, 58], and the spread of botnets and malware [8, 26, 36, 44, 52, 63].

As with other network traffic research infrastructure, the three biggest infrastructure challenges of network telescope instrumentation are collection and storage, efficient curation, and sharing large volumes of data. Some network telescopes use partially active (*live*, or *lit*) address space, which brings another challenge: co-existence with operational network traffic. In the case of the UCSD-NT, this coexistence requires careful filtering of traffic to subnets in the underlying address space that are legitimately assigned (leased) to users, and therefore not dark. In recent years, this co-existence has grown more complex, as ARDC has expanded its leasing of address space to its members, requiring more frequent updates of the filter list. This complexity demands careful attention to ensure correct filtering of traffic destined toward the telescope, in order to preserve the integrity of the data and research that uses it. We investigate this operational complexity and its implications, including how to detect misconfigurations that impair the resulting data. Our contributions are:

1 Introduction

Network telescopes collect unsolicited Internet traffic (called *Internet background radiation (IBR)*) sent to largely unused IP address space (called *darkspace*). Over the last two decades, CAIDA (at UCSD) has been operating the world's largest network telescope (UCSD-NT) to capture IBR from an address space owned by the Amateur Radio Digital Communications (ARDC). The data gathered by the UCSD-NT has

(1) We describe the historical and operational context of the UCSD-NT (§2), how it operates in cooperation with the owners of the address space, and associated operational challenges (§3).

(2) We introduce and validate a technique to monitor the address space and packet loss of a network telescope by leveraging third-party Internet-wide scanning campaigns (§4). Our method requires no special privileges beyond user access to the data and is effective in auditing the completeness of the data collected by UCSD-NT.

Probing It Ourselves
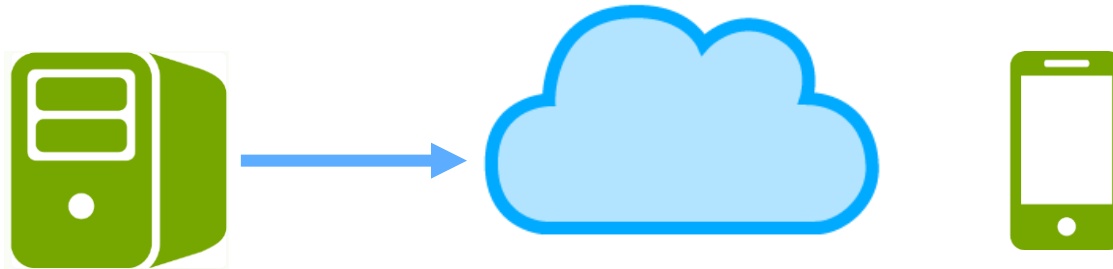# ACTIVE MEASUREMENTS

How to measure the data plane?

Active

Passive

Examples   Ping, traceroute, scanning, …   Traffic monitoring, log files, …

Active measurements on the data plane send packets from end host(s) to other host(s).

It involves the network, transport, and usually the application layer.

Typical examples for active measurements

Internet delay analysis (round trip time)

Deployment of application layer services

DNS ecosystem

Web ecosystem

Certificate ecosystem

+++

Challenges

Coverage
Which sources and which destinations do you select to prevent a bias?

Performance
Sending many packets takes time, may challenge system resources etc.

Ethics
Easier to inject packets on the data plane compared to control plane, easier to introduce unintended effects

Protection
Depending on the measurement objective, source IP addresses should be whitelisted

Good practices

Add `Whois` entries for measurement prefixes

Add reverse DNS entries for source IP addresses

Create a web page that explains your project and lists a point of contact

If something goes wrong, operators want to know what is going on & who is responsible ;)

Expand the set of measurement probes

Building a dedicated distributed measurement infrastructure, which involves the deployment of specific hardware probes

Recruit users to run software probes

Two simple examples and what might go wrong

Ping

Send ICMP echo requests, wait for ICMP reply

You measure the reachability of an end host, do you?

Traceroute

Probes the IP path

Keeps very few states

Traceroute: Principle approach

S -> D, TTL=1

R1 -> S, TTL exceeded

S -> D, TTL=2

R2 -> S, TTL exceeded

…

The problem of load balancers

L

R1   R2

R3   R4

R5

Per-flow load balancer
Per-packet load balancer
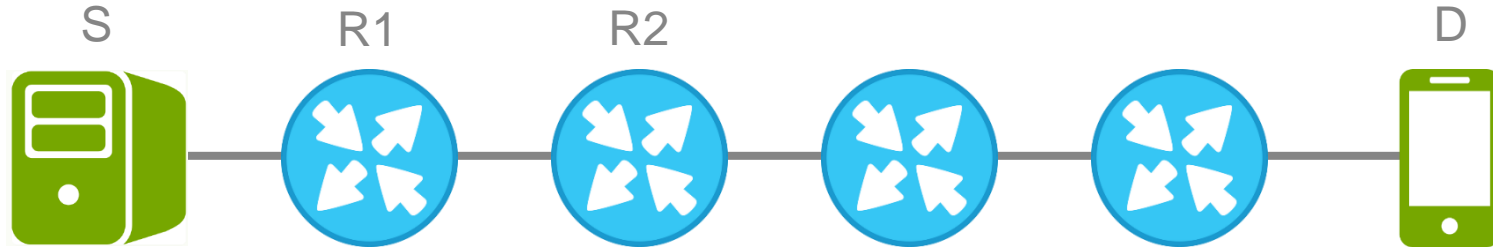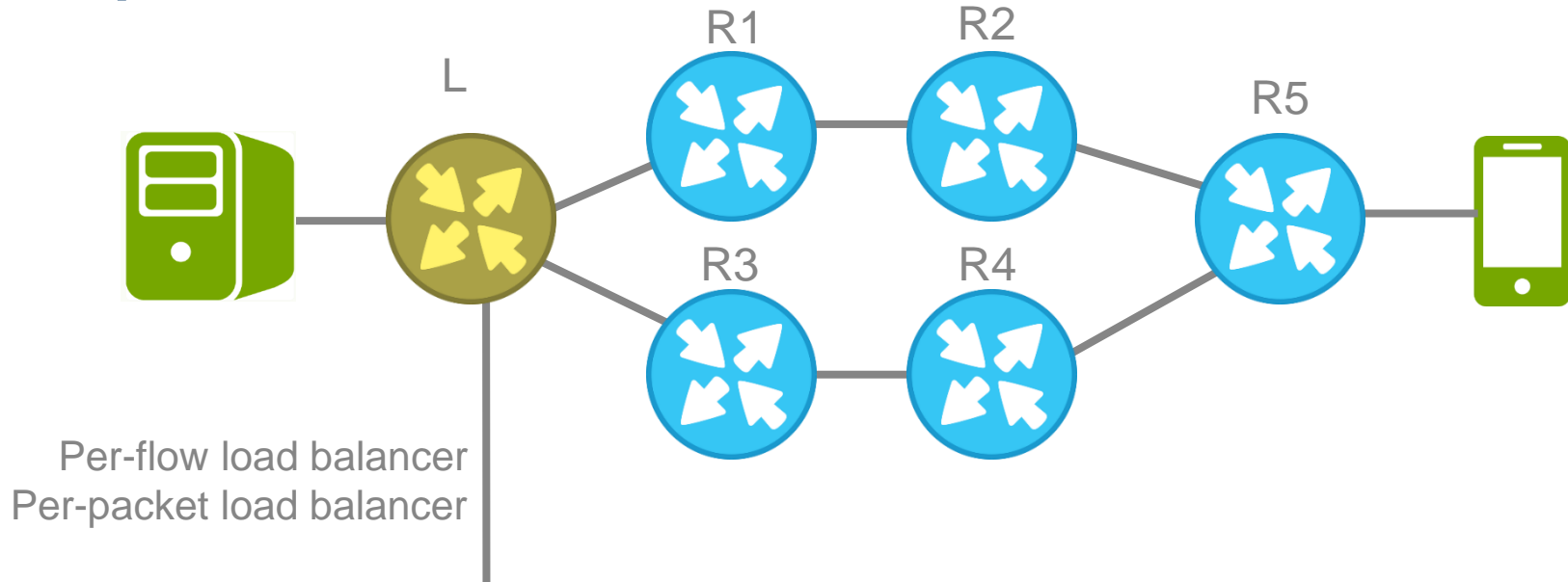
The problem of load balancers
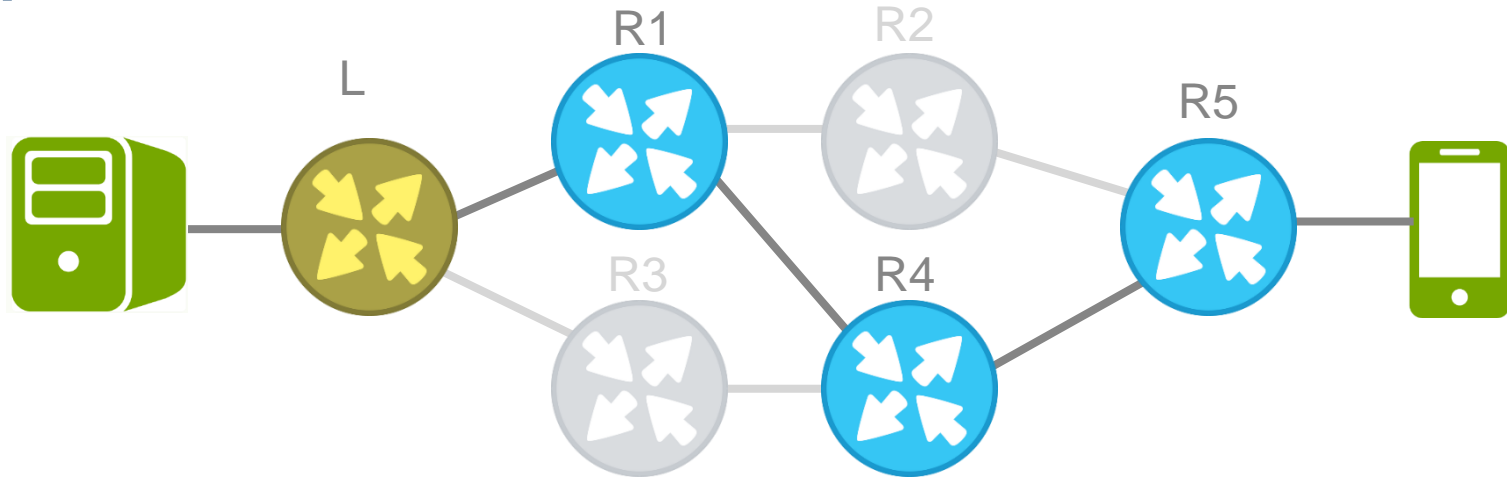
Missing nodes and links

False links

The core problem

Traceroute changes header fields

UDP traceroute: varies destination port

ICMP traceroute: varies sequence number

Many load balancers identify flows based on the first four octets of the transport header

Checksums cover even 'back' fields

The core problem & solution

Paris traceroute controls probe packet headers to overcome per-flow load balancing

Maintaining header fields is challenging because traceroute still needs to match request and reply

IP

Version	IHL	TOS	Total Length	
Identification (+)			Flags	Fragment Offset
TTL		Protocol	Header Checksum	
Source Address				
Destination Address				
Options and Padding				

UDP

Source Port	Destination Port (#)
Length	Checksum (#,*)

ICMP Echo

Type	Code	Checksum (#)
Identifier (*)		Sequence Number (#,*)

TCP

Source Port		Destination Port	
Sequence Number (*)			
Acknowledgment Number			
Data Offset	Resvd.	ECN	Control Bits	Window
Checksum		Urgent Pointer	
Options and Padding			

Key

☐ (grey) Used for per-flow load balancing   ☐ (white) Not encapsulated in ICMP Time Exceeded packets

\# Varied by classic traceroute   + Varied by tcptraceroute   * Varied by Paris traceroute

Based on common header fields you can gain more information to discover anomalies

Probe TTL is in the encapsulated IP header echoed in ICMP Time Exceeded message and should be 1

Response TTL is the TTL in the IP header of the Time Exceeded msg. and should reflect the length of the return path

IP ID field set by the router and incremented for each packet send, helps for de-aliasing

Anomalies in traceroute: Loops
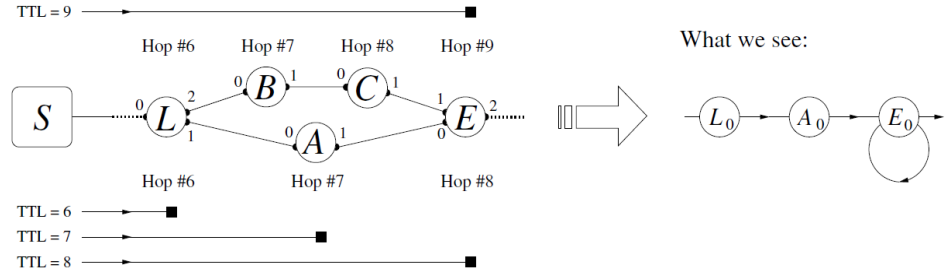
Loop because of load balancing

Anomalies in traceroute: Loops

Loop because of load balancing
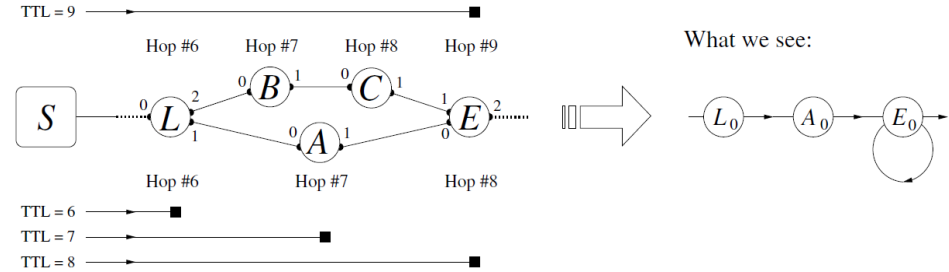
Loop because of zero-TTL forwarding
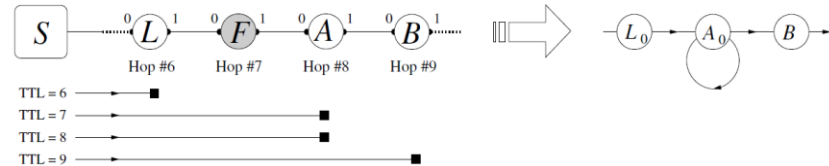
Anomalies in traceroute: Loops

Loop because of load balancing
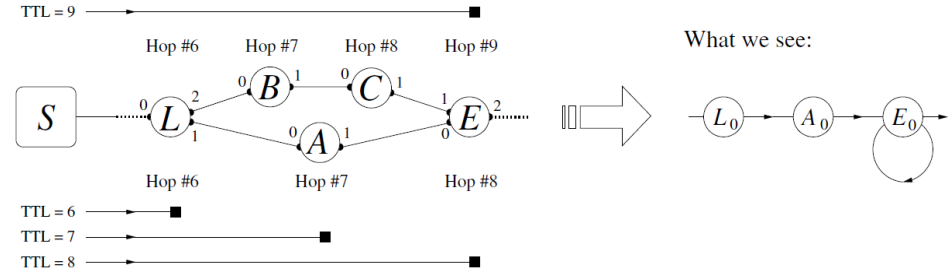
Loop because of zero-TTL forwarding
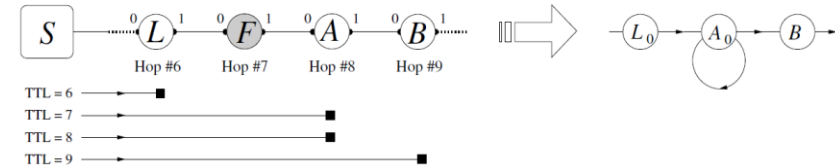
Loop because of address rewriting
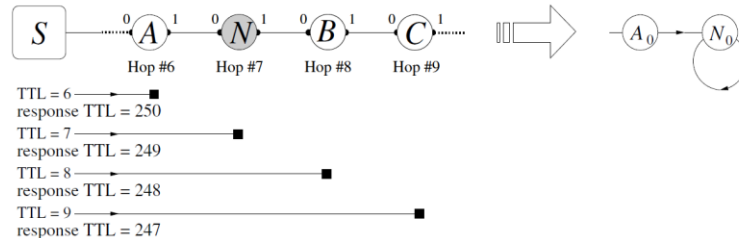
Anomalies in traceroute: Loops

Destination unreachable messages needs special consideration

Anomalies in traceroute: Loops

One month measurement study in 2006, to 5,000 randomly chosen nodes

Numbers to give you some idea

5% of the measured routes contained at least one loop

Loops because of load balancing: ~84%

Anomalies in classic traceroute: Cycles and Diamonds

Cycles

Load balancing and unreachability messages may lead to observed cycles, similar to loops
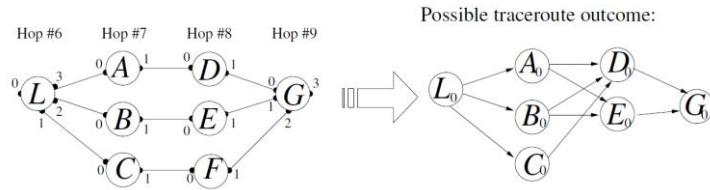
Diamonds

Arises only when multiple probes per hop are sent

Main cause: load balancing

Further challenges in traceroute

Routing path asymmetry

Routing policies, default routes, etc.

IP aliasing

How to distinguish multiple interfaces of the same router?

Literature

Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (IMC '06). ACM, New York, NY, USA, 153-158.
http://dx.doi.org/10.1145/1177080.1177100

What Researchers Do for Us

COMMON MEASUREMENT INFRASTRUCTURES

CAIDA Archipelago
(Ark, http://www.caida.org/projects/ark/)

Dedicated nodes that perform traceroutes and other measurements

Results are public
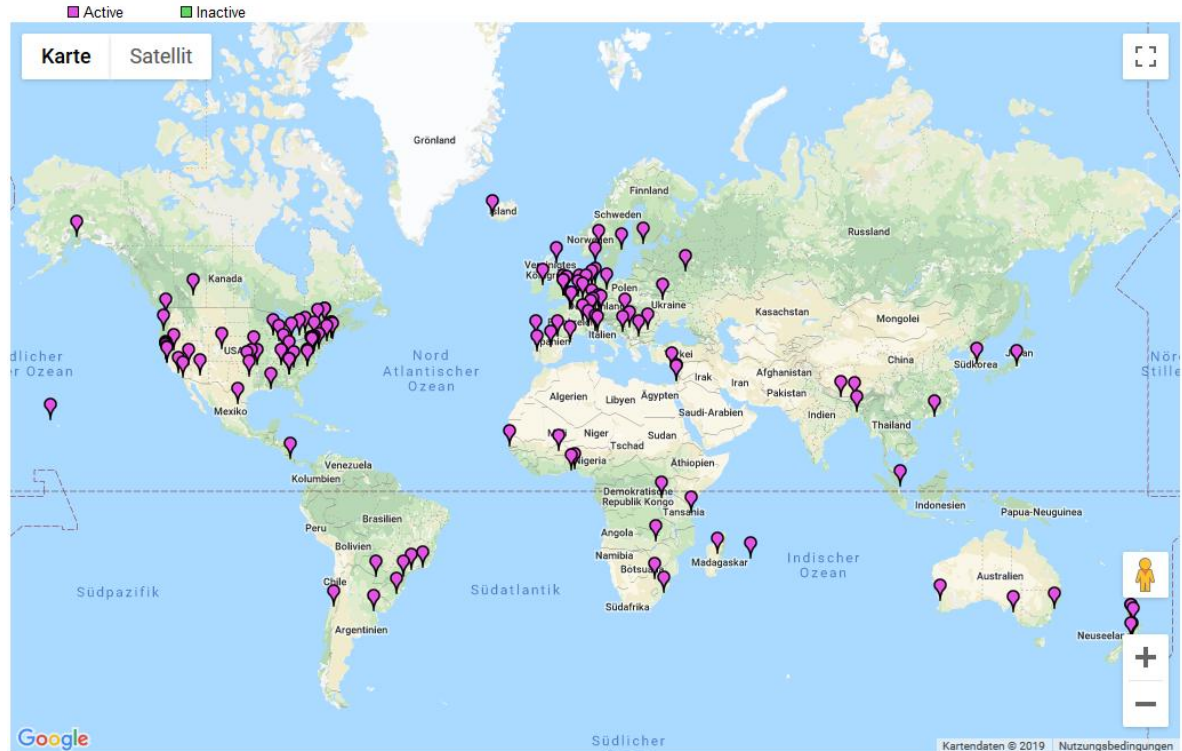
RIPE Atlas

Dedicated nodes common measurements

Credit-based system to perform own measurements

Results are public

RIPE Atlas in numbers

- 10,000 probes and 400 anchors connected worldwide
- 5.6% IPv4 ASes and 9% IPv6 ASes covered 181 countries covered
- 7,000 measurements per second

Most popular RIPE Atlas features

- Six types of measurements: ping, traceroute, DNS, SSL/TLS, NTP and HTTP (to anchors)

- APIs to start measurements and get results

- Powerful and informative visualisations: "Time Travel", LatencyMON, DomainMON, TraceMon

- CLI tools

- Streaming data for real-time results

- Roadmap shows what's completed and coming

Ethics design decisions (1)

- Active measurements only
  - probes do not observe user traffic
- Low barrier to entry
  - gratis probes, funded by LIRs and sponsors
- Hosted by volunteers
  - informed consent (accepting T&C)
  - personal data never revealed
- Data, API, source code, tools: free and open
- Measurements sets limited

Ethics design decisions (2)

- No bandwidth measurements
    - Other platforms provide that service
- HTTP measurements only towards RIPE Atlas anchors
    - Otherwise it would rely on hosts' bandwidth
    - And might put volunteer at risk