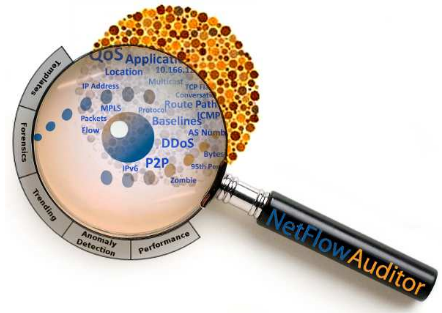
Pedro Casas

Telecommunications Research Center Vienna – FTW

Network Traffic Anomaly Detection

IIE – FING – ARTES

1–5 September 2014



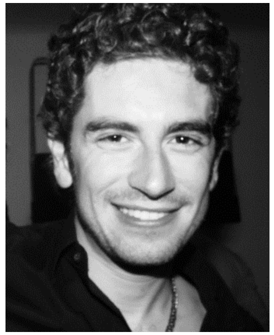
Thanks giving to many colleagues

- *The material presented in these slides is partiallty taken from the work done by Dr. Alessandro D'Alconzo @FTW*

Marco Mellia
Politecnico di Torino

Raimund Schatz
FTW

Arian Bär
FTW



Pierdomenico Fiadino
FTW

Ernst Biersack
EURECOM

Alessandro D'Alconzo
FTW

Tobias Hossfeld
Würzburg Universoty

Mirko Schiavone
FTW

Philippe Owezarski
CNRS

Alessandro Finamore
Politecnico di Torino

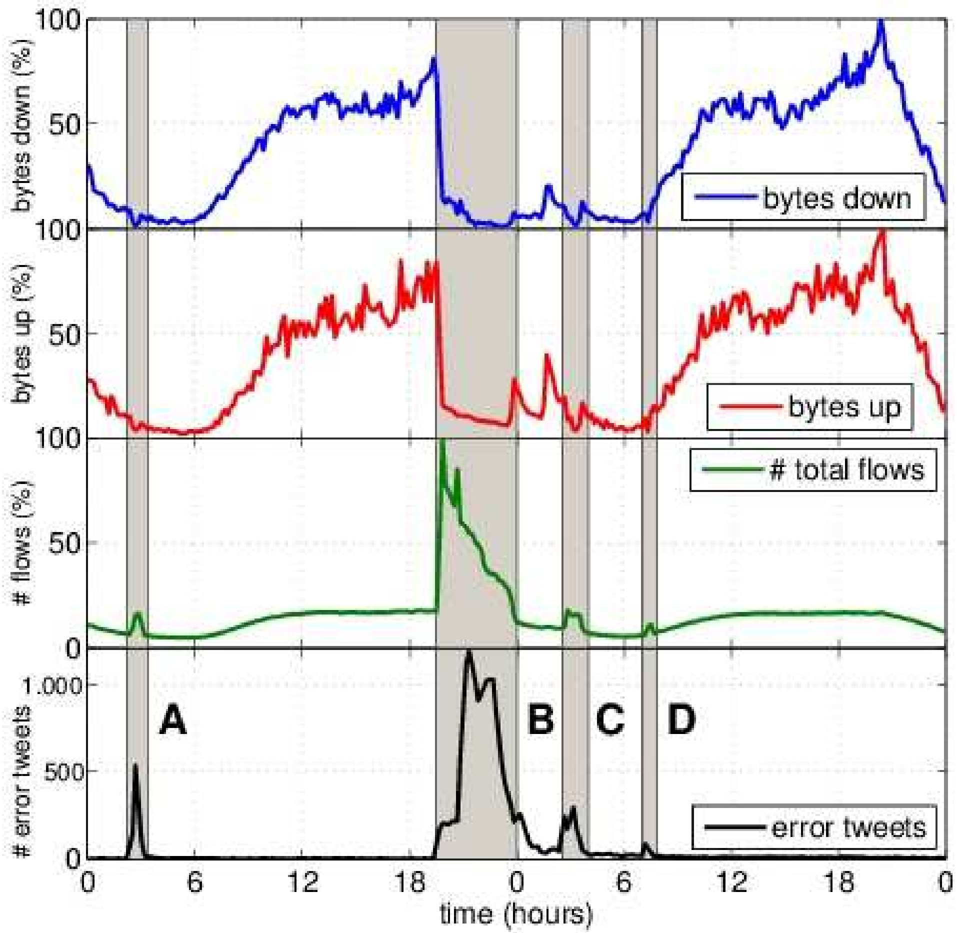
Why Anomaly Detection

- Fast-changing environment → production of new errors
  - example of network errors: congestions, failures, equipment misfunctioning...
- Connection to the Internet → exposed to attacks
  - including novel mobile-specific attacks
- Our focus: "anomalies" that (might) impact performance of the network infrastructure and the end users
  - events involving multiple mobile terminals (**macro-anomalies**)

The big outage (Feb. 22nd, 2014)
press reaction

ftw Creating Communication Technologies

The big outage (Feb. 22nd, 2014)
as seen from passive measurements and social feeds



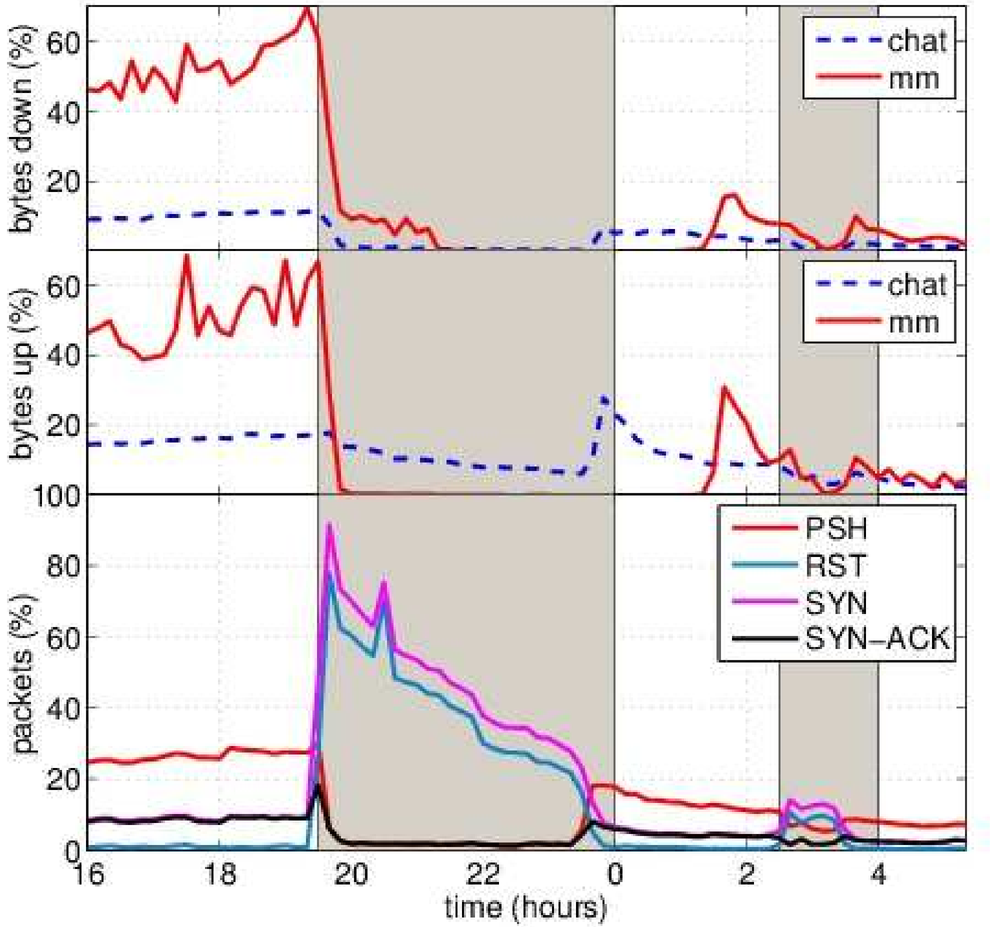
drop in volume down

drop in volume up

ramp-up on flow counts

#whatsappdown

The big outage (Feb. 22nd, 2014)
as seen from passive measurements and social feeds

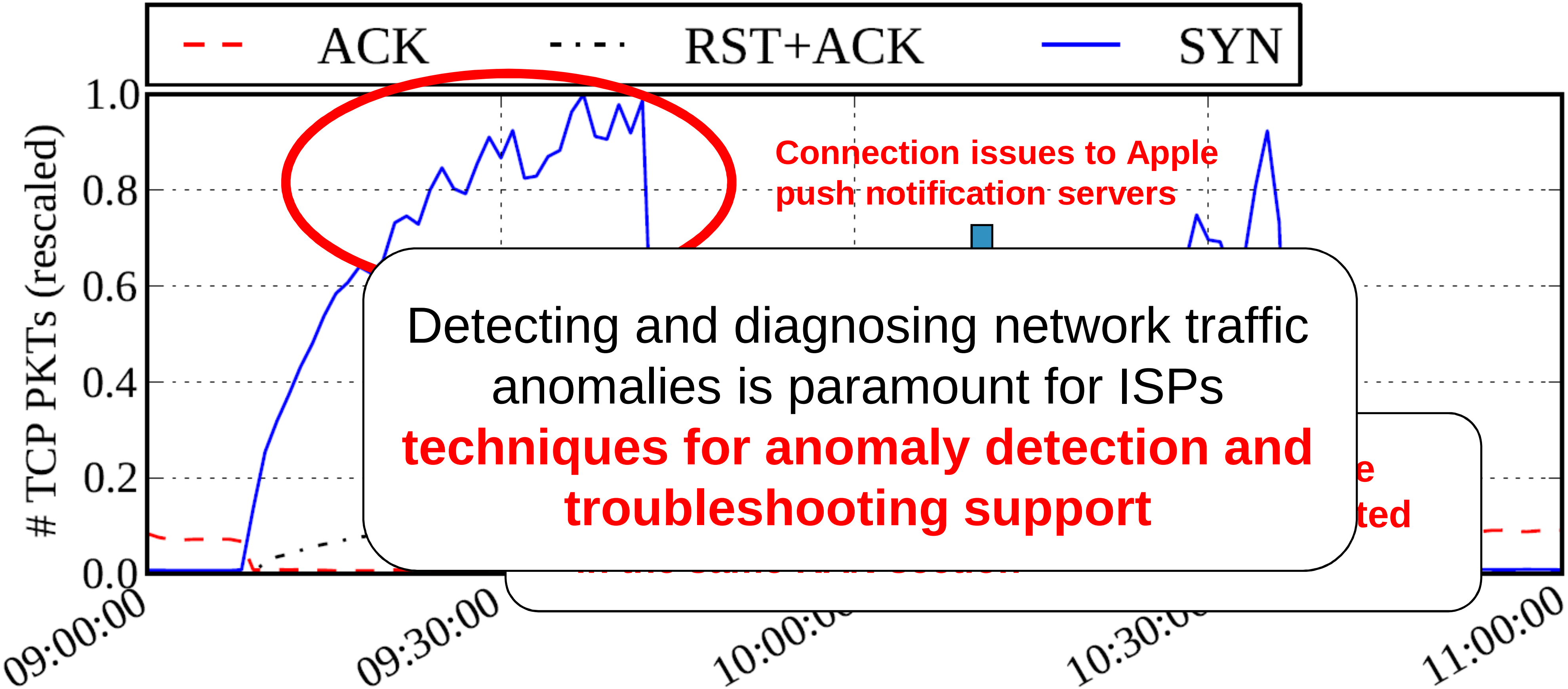


residual volume down (mm)

residual volume up (chat)

TCP flags counters

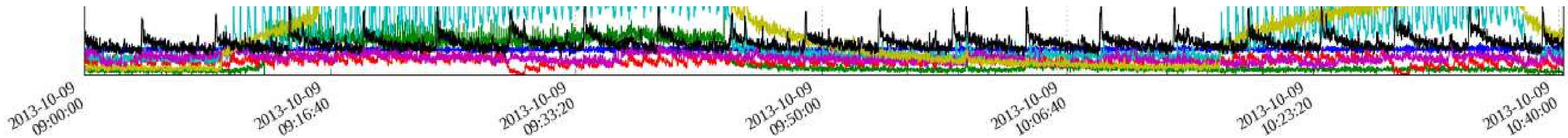
Network Traffic Anomaly Detection



Detecting and diagnosing network traffic anomalies is paramount for ISPs

techniques for anomaly detection and troubleshooting support

uplink/downlink TCP packets

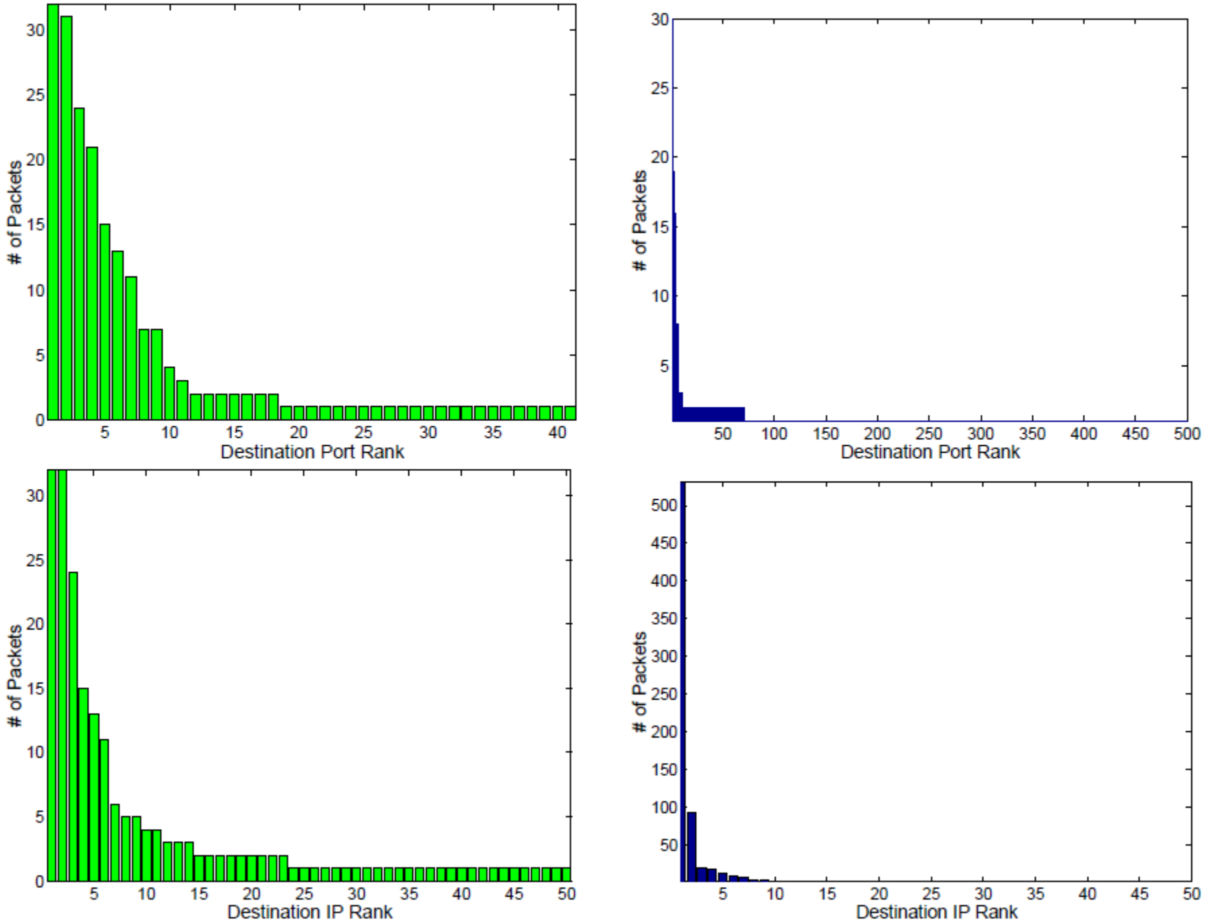


iOS7, iOS<7

Detecting Network Attacks (1/2)

- *The first stage of a DDoS attack is contaminating the devices to create a BOTNET*
- *The contamination is done through the propagation of a WORM*
- *The worm looks first for a backdoor to infect the victim*
- *→ PORT SCAN attack to find open ports*

Features distributions change during an anomalous event



(a) Normal (b) During Port Scan

Detecting Network Attacks (2/2)

- *Entropy-based detector → uses the empirical entropy of the monitored features as a summarization tool of the distribution*

$$H(X) = -\sum_{i=1}^{n} p(x_i) log(p(x_i))$$

$$x_1, \ldots, x_n$$

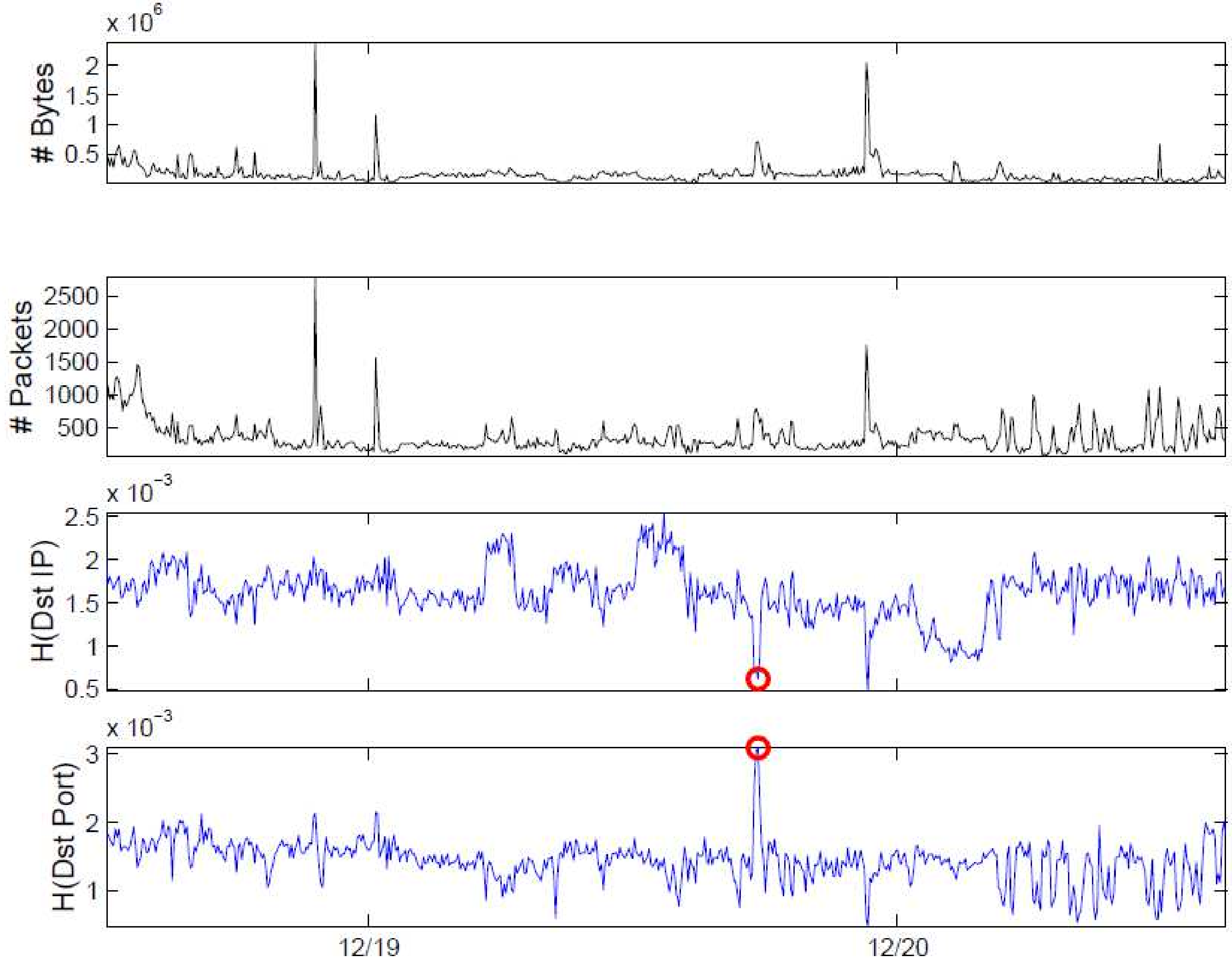
empirical (from measurements)

Detecting Network Attacks (2/2)

- *Entı*
  feat

$H($

$x_1, .$



Port scan anomaly viewed in terms of traffic volume and in terms of entropy.

Creating Communication Technologies

A Statistical-based Approach for Anomaly Detection

A statistical-based approach to AD

- The approach: **distributional change-detection**
  - maintain individual counters per-mobile station
  - for each feature ϕ, count occurrences in timebins of length τ
  - extract (timeseries of) empirical distributions $\mathbf{X}_{\phi,\tau}(\mathbf{t})$

- Qualitative definitions:

  to be defined

  to be defined

  - **Anomaly** = a statistically-relevant deviation from the "typical behaviour" of the traffic distributions

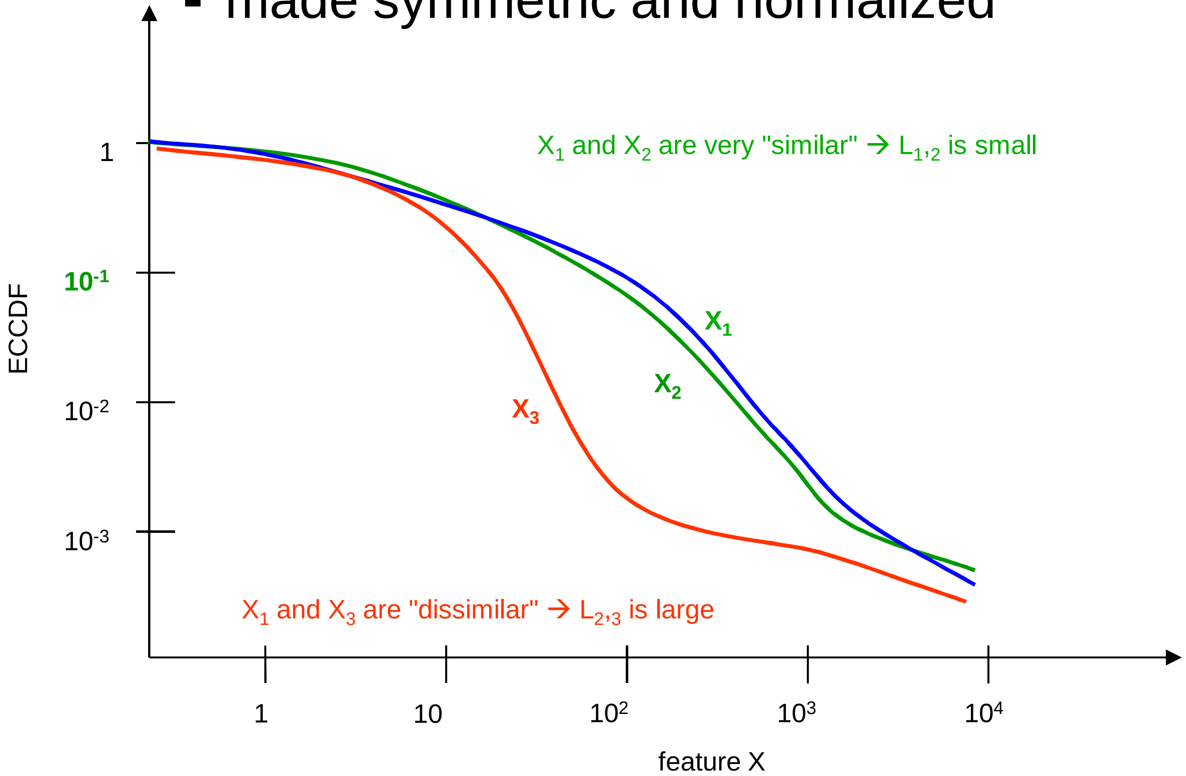
  - **Typical behaviour** = the dominant traffic pattern that was observed in the past

  to be defined

  to be defined

Measuring statistically-relevant deviations

- Define a **divergence metric** between empirical distributions
  - $\mathbf{L_{\phi,\tau}(t_1,t_2)=f(X_{\phi,\tau}(t_1),\ X_{\phi,\tau}(t_2))}$
  - we used a metric derived from Kullback-Leibler
  - made symmetric and normalized



KL divergence:

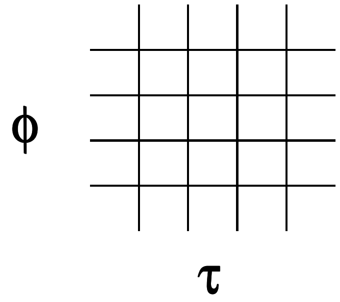
$$D(p\,\|\,q)=\sum_{\omega\in\Omega} p(\omega)\log\frac{p(\omega)}{q(\omega)}$$

ENKL metric:

$$L(p,q)=\frac{D(p\,\|\,q)}{H_p}+\frac{D(q\,\|\,p)}{H_q}$$

A distributional change-detection algorithm

- We look at aggregate traffic as a grid of feature/timescale combinations $\mathbf{X}_{\phi,\tau}(\mathbf{t})$
  - at different features ϕ → multi-dimensional
  - at different aggregation timescales τ → multi-resolution



- Use the divergence metric to compare the current observations with "**the past**"
  - need to define a **baseline** representative of "the past"

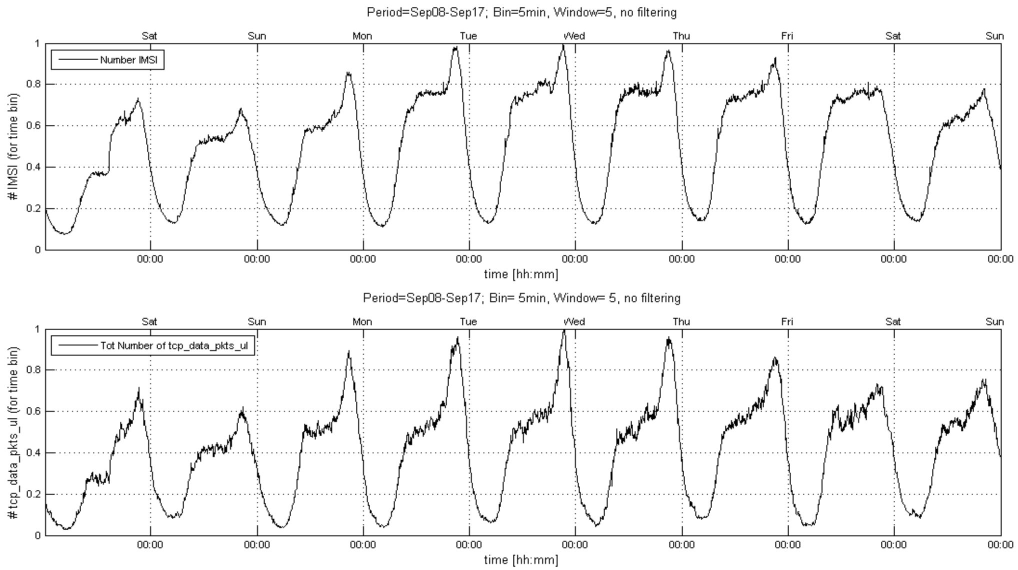
- Need first to understand what is the **"typical"** behaviour of distribution timeseries
  - **temporal patterns, (ir)regularities**

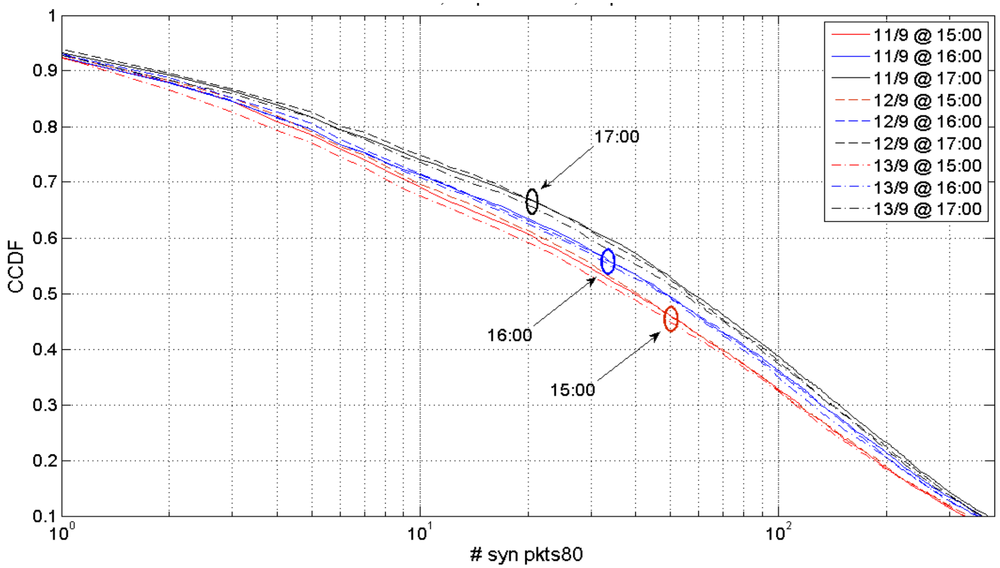
Temporal characteristics of feature distributions

- Marked 24h pseudo-seasonality + (slow) long-term trends
  - steep variations at morning and evening shaped by human activity cycle
  - time of day variations due to changes of terminal/application mix
  - distributions exhibit larger fluctuations during night (due to much lower number of active terminals)

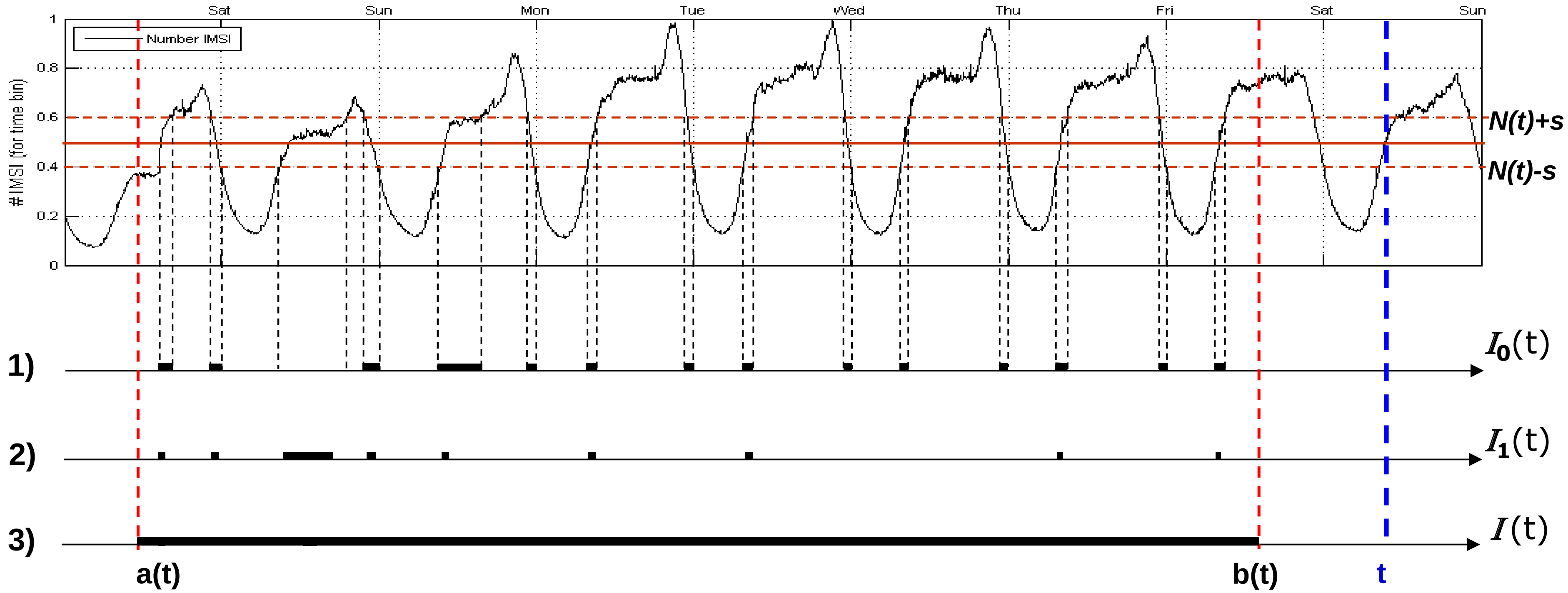
- Marked differences (for some features) between working days and weekend/festivities

- Distributions at the same hour of different days tend to be pretty similar





Constructing a dynamic reference baseline

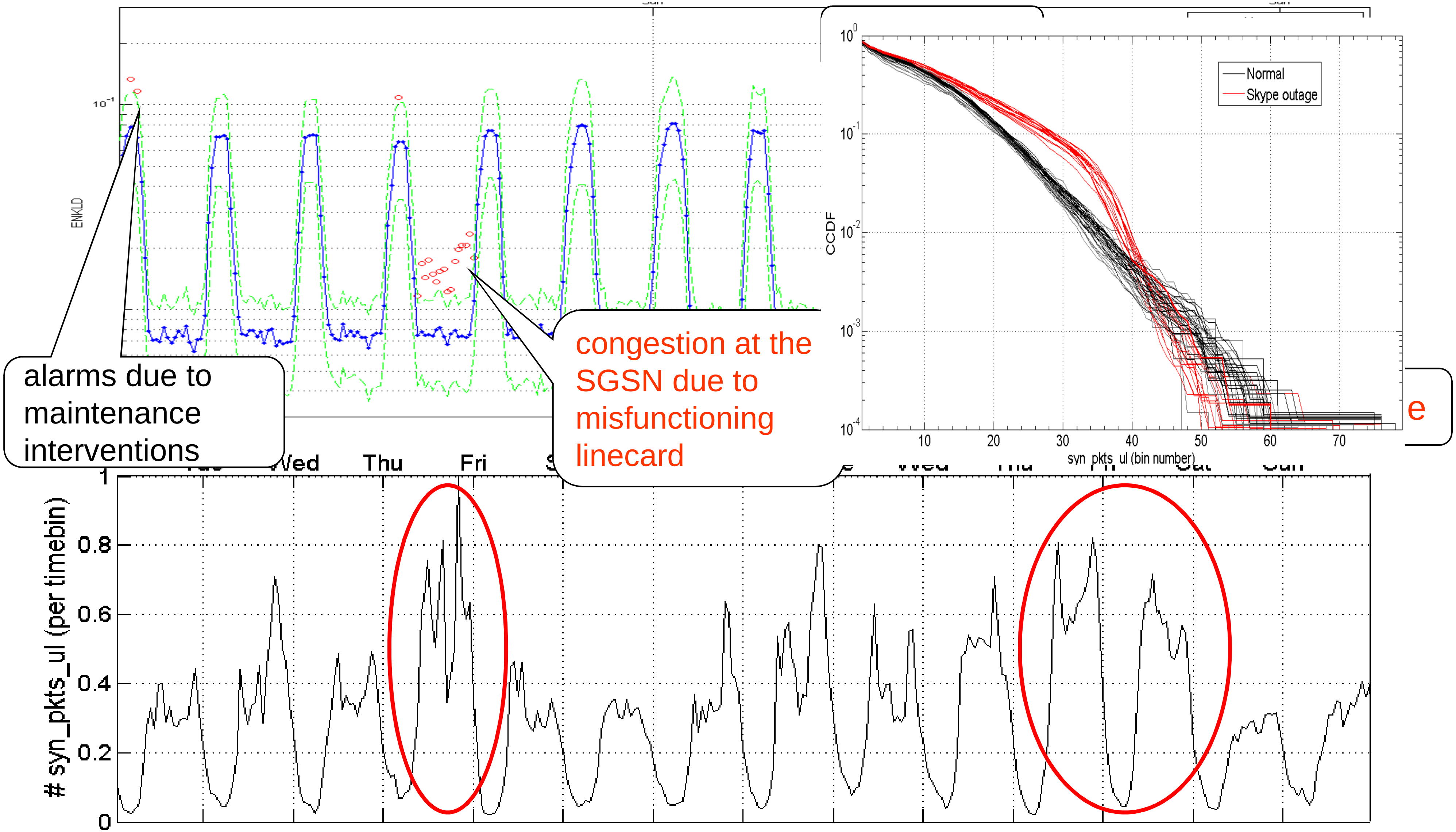


To evaluate sample at time **t**, with N(t) active users and ditribution X(t)

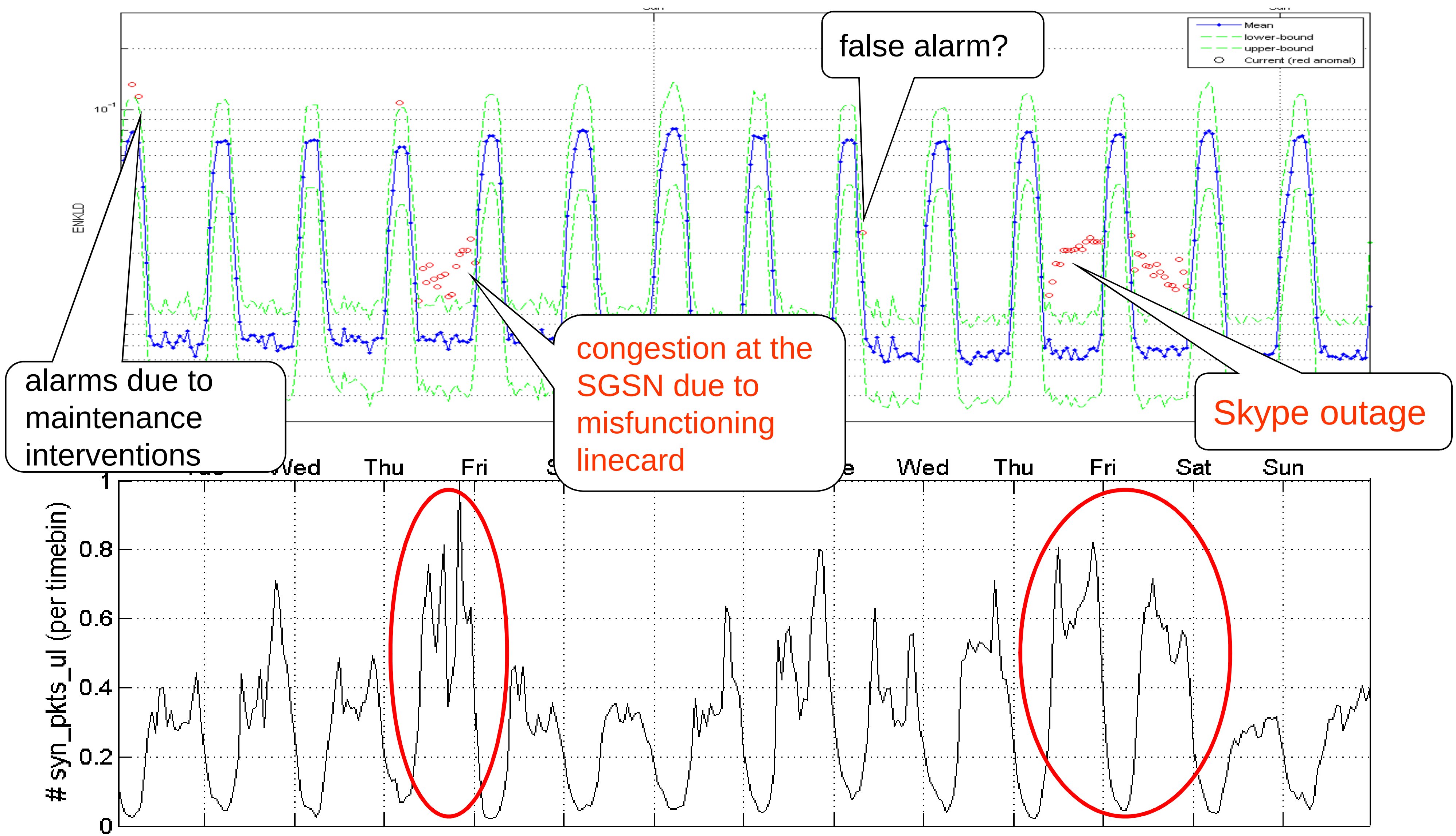
Construction of baseline (samples not older than 2-3 weeks):
1. Consider the past samples with N~N(t)
2. Pick the "closer" to X(t), based on the ENKL distance (for filtering out different times of day)
3. Reduce the reference set using a "pruning" heuristic

Comparison
4. compute all divergence pairs within the baseline, extract α-percentile
5. compute average divergence between current sample and baseline elements
6. compare them

Examples of real alarms

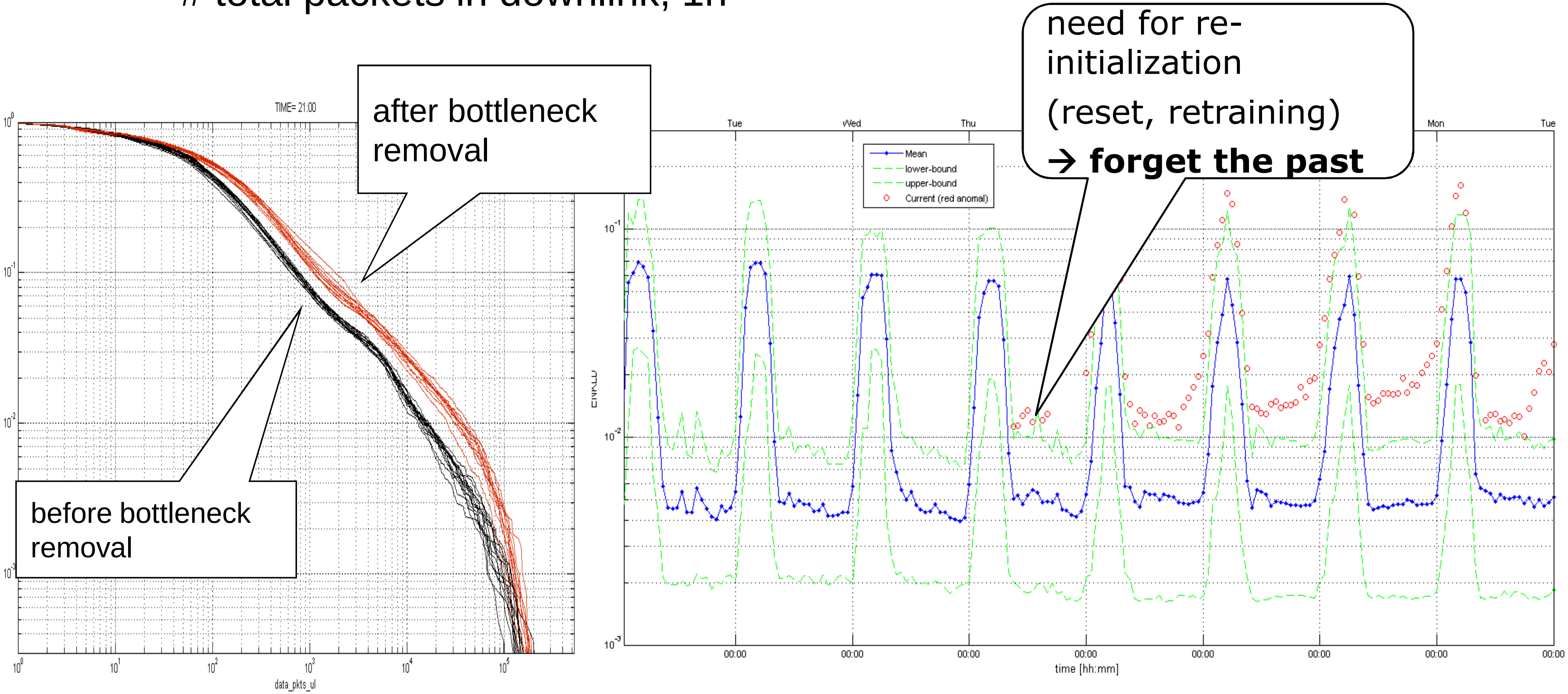


Examples of real alarms



An example of persistent change

- Capacity upgrade at a SGSN because of a bottleneck link
  - \# total packets in downlink, 1h



Detecting Traffic Shifts in CDNs

The Case of Facebook

Why Detecting CDNs Traffic Shifts?

- CDN perform load-balancing among multiple servers (FEs, content replica)
- Complex and undisclosed time/space variant policies
- Understanding CDN traffic patterns is challenging

CDNs policies have:

- impacts on traffic routed by underlying transport network
- influences on achieved latency/throughput (end-user's QoE)

It's important for ISPs to rapidly and automatically detect the occurrence of macroscopic changes in how CDNs serve traffic...

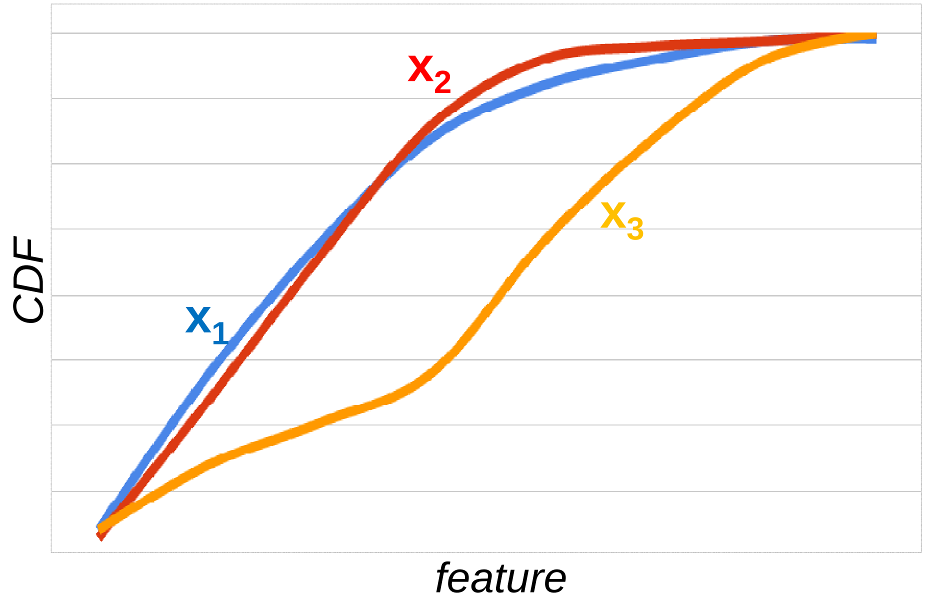
…especially when ISPs themselves and their users are negatively affected (i.e. anomalies)

ADTool
A statistical Anomaly Detection (AD) Tool

- **AD algorithm** consists of two phases for each iteration (time batch):

(1) **Reference-Set identification**: find past traffic distributions which are a suitable reference of normality (sliding window)

(2) **AD test**: use a normalized variant of the *Kullback-Leibler divergence* to decide if current distribution is compatible with the reference-set



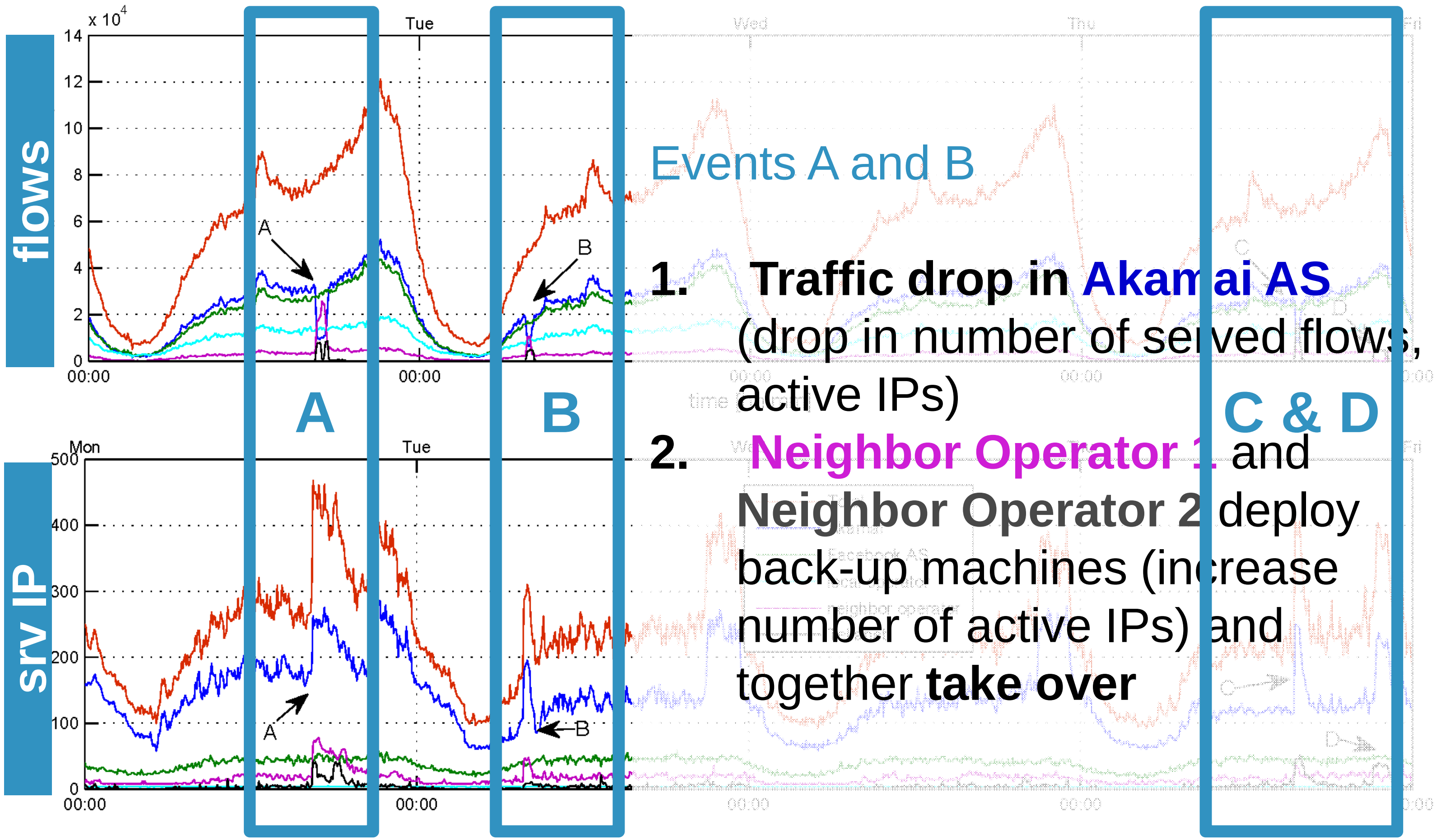
$$D(P\,||\,Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

$$L(P,Q) = \frac{D(P\,||\,Q)}{H(P)} + \frac{D(Q\,||\,P)}{H(Q)}$$

x_1 and x_2 are *similar* $\rightarrow L(x_1,x_2)$ is small
x_1 and x_3 are *dissimilar* $\rightarrow L(x_1,x_3)$ is large

Akamai macroscopic traffic shifts [1/3]

Time Series (4 days)



Events A and B

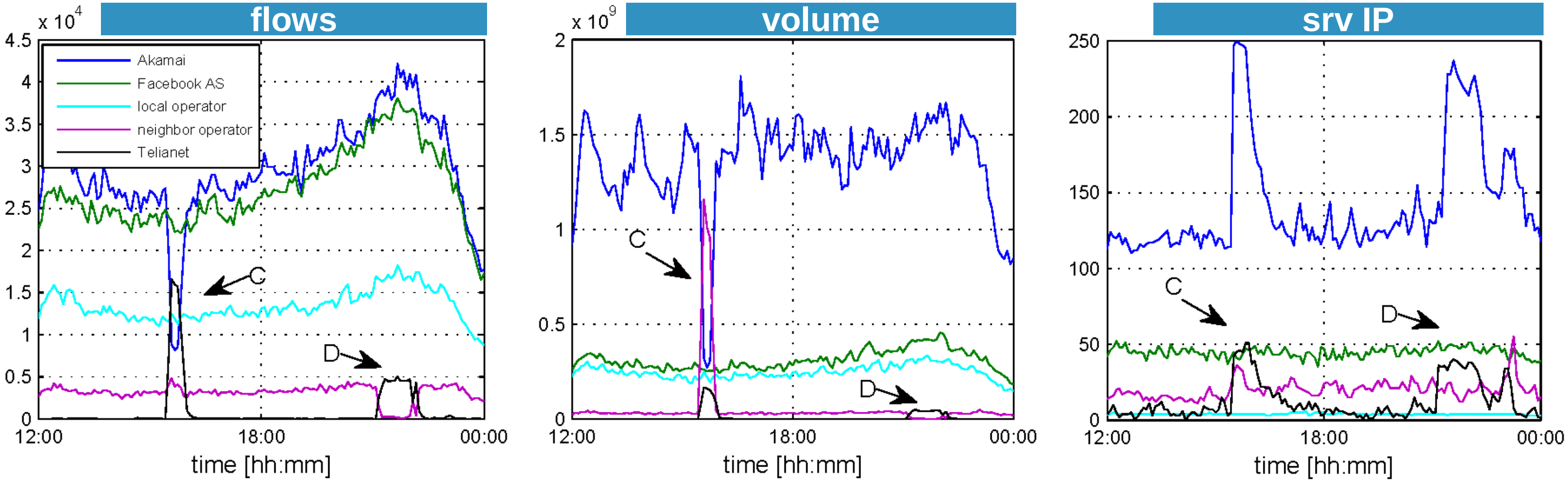
1. **Traffic drop in Akamai AS** (drop in number of served flows, active IPs)
2. **Neighbor Operator 1** and **Neighbor Operator 2** deploy back-up machines (increase number of active IPs) and together **take over**

- CDNs have ~*constant* share of deployed IPs and number of flows
- Facebook AS and Akamai lead the number of served flows
- Akamai employs largest share of active IPs per time-bin

Akamai macroscopic traffic shifts [2/3]

Time Series (12 hours zoom-in)

- Zoom on last 12 hours:



- Event C
  1. Akamai: drop in number of flows, served volume but NOT active IPs
  2. Neighbor Operator 2 increases number of active IPs, number of flows and volume
  3. Neighbor Operator 1 keeps same number of active IPs, but increase served volume (takes over Akamai's larger flows)
- Event D
  - Akamai not involved
  - Swap between NO1 and NO2 w.r.t. number of flows

Akamai macroscopic traffic shifts [3/3]

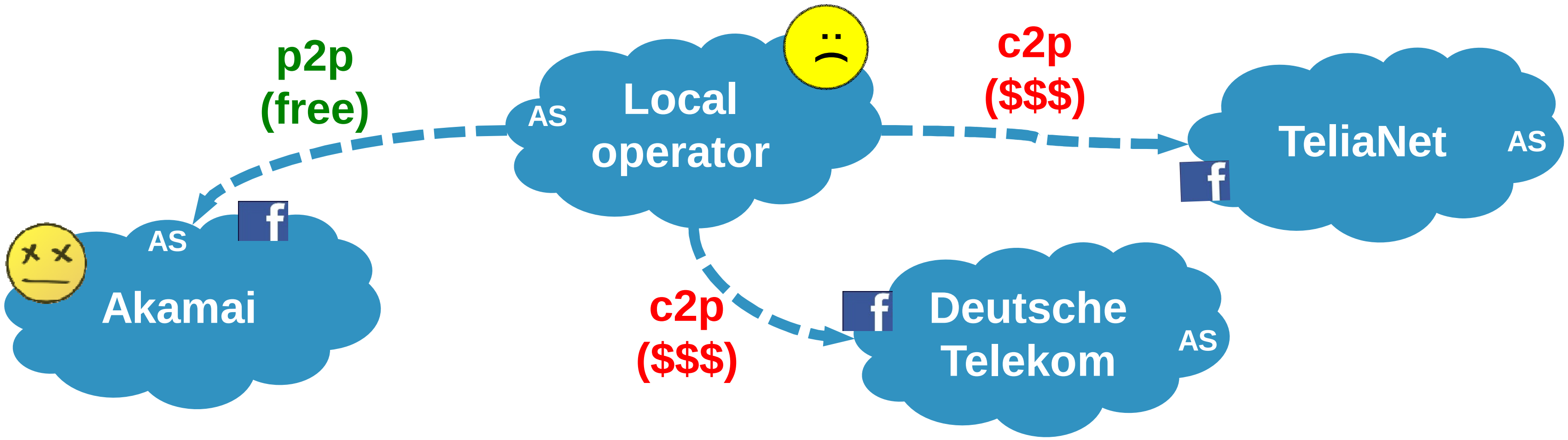
Some considerations

Events A-D reveal chain of agreements in serving contents

Depending on the nature of **commercial agreements** for peering, it is possible that huge shifts of traffic volumes from one AS to another imply an **economical loss** for the ISP

No performance impact from user perspective (normal RTT, throughput, number of erroneous HTTP response codes)

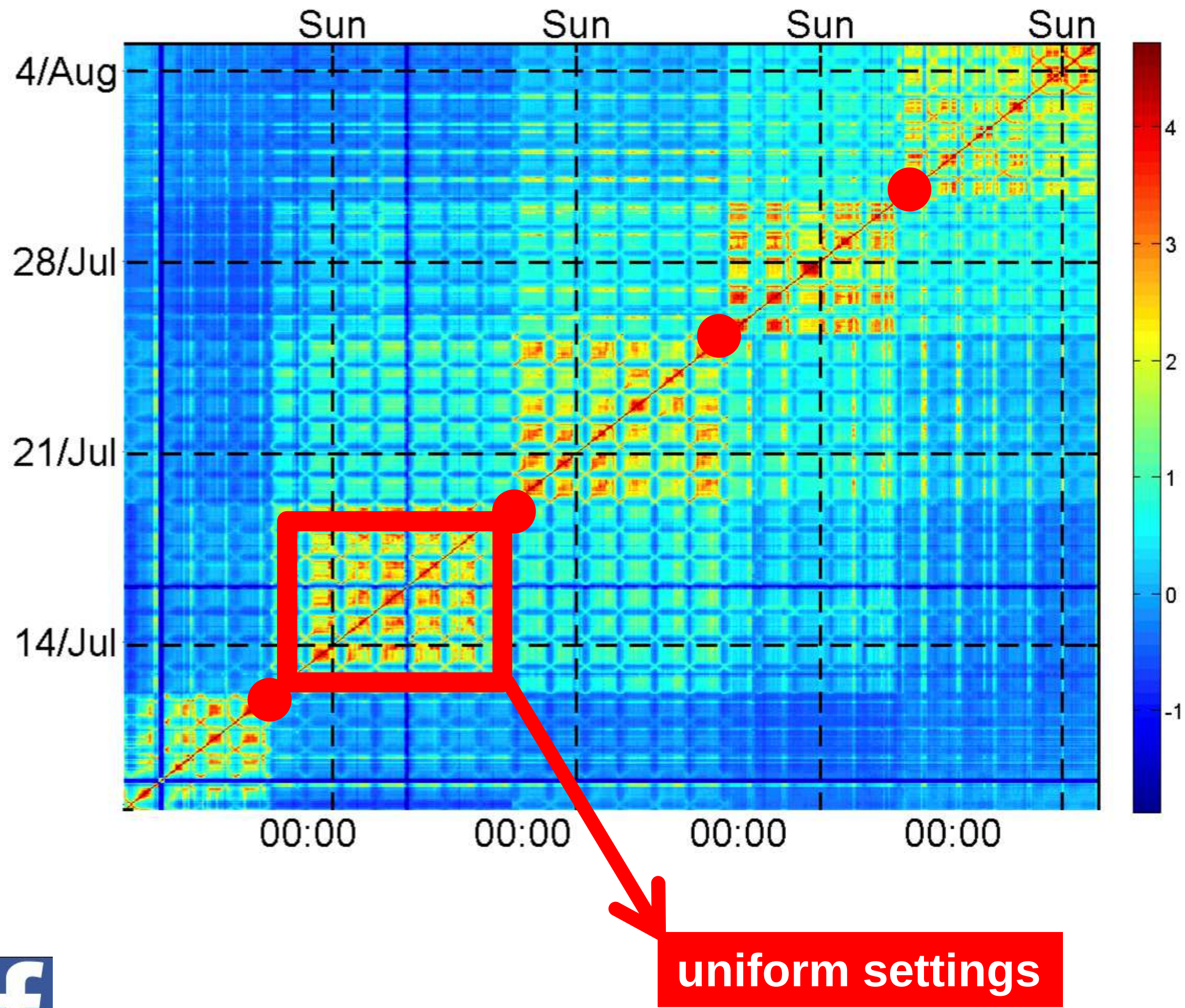
But different commercial agreements for peering:



Topology from: http://irl.cs.ucla.edu/topology/

Temporal Similarity Plots (TSP)
A powerfull tool to visualize temporal patterns

- Discover **temporal patterns** and ***(ir)*regularities** in distribution timeseries

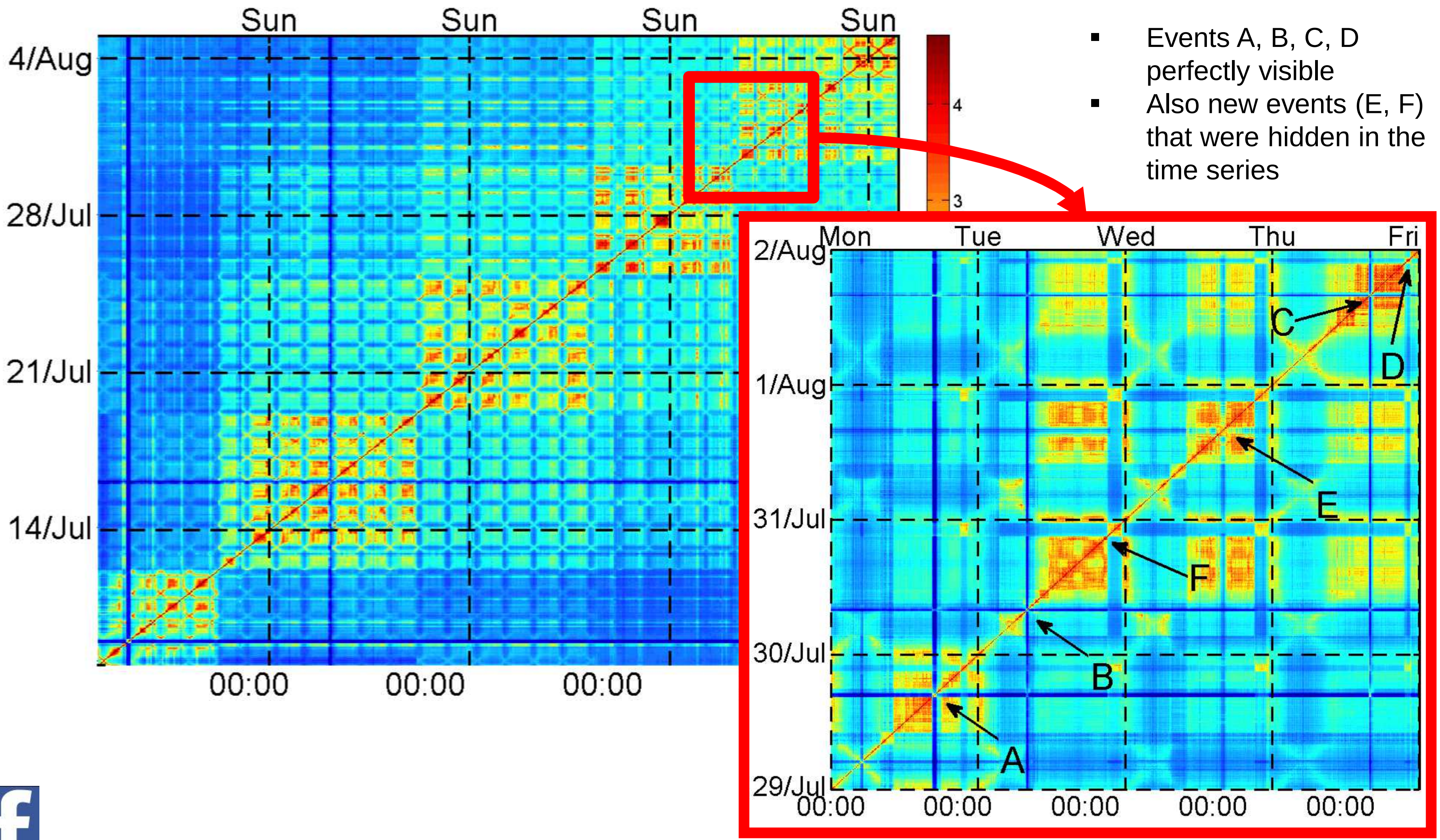


1. *For every IP: counters of flows number and volume*
2. *Counters cumulated over different time scale (eg. 1hour)*
3. *For every time-bin: distribution of counters across IPs*
4. *Distribution compared with Kullback-Leibler metric*
5. *Comparisons plotted on heatmap (logscale)*

Temporal Similarity Plots (TSP)

A powerfull tool to visualize temporal patterns

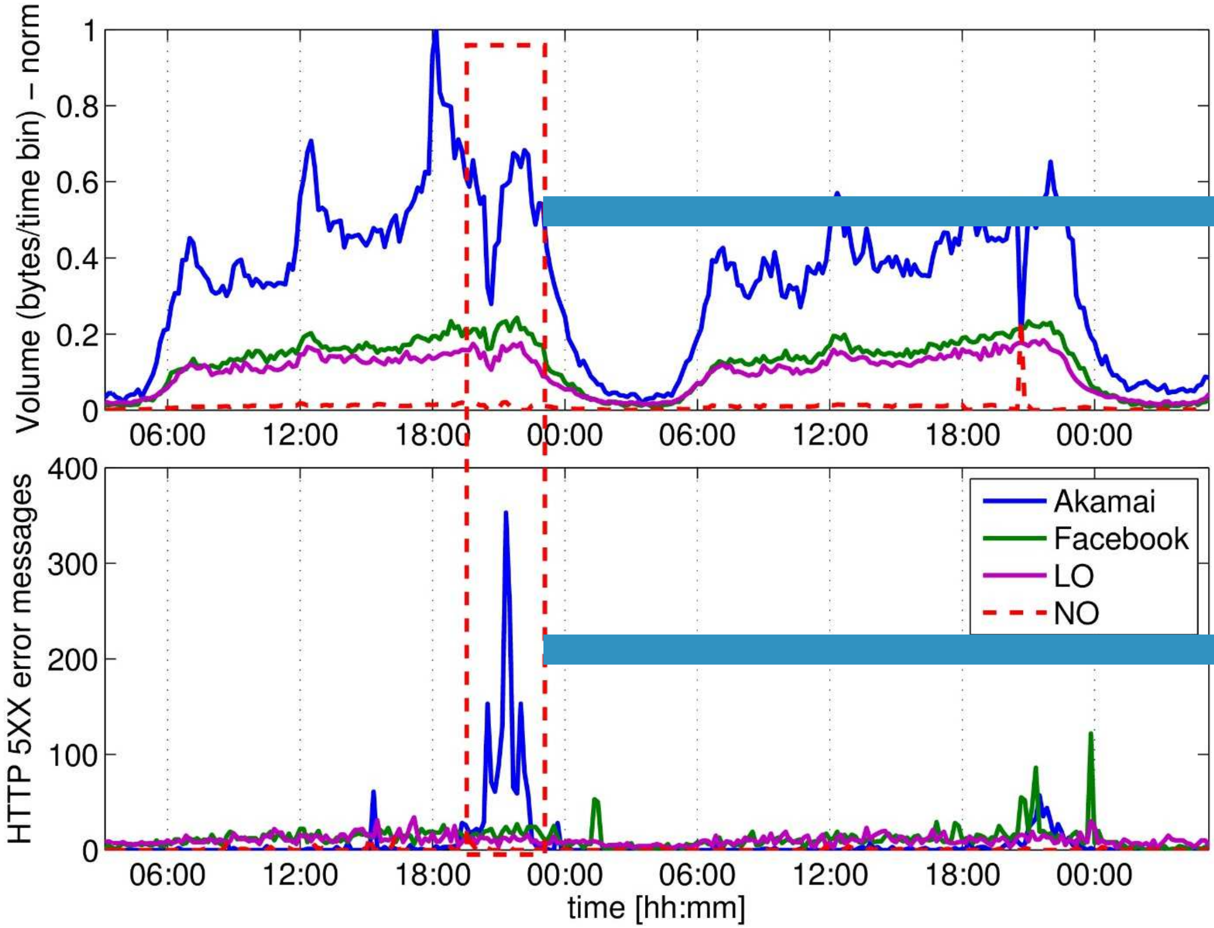
- Zoom on unexpected events



- Events A, B, C, D perfectly visible
- Also new events (E, F) that were hidden in the time series

Detecting Facebook Outages
September 2013

- Outages are typically not linked to CDN load-balancing policies
- Nevertheless, they may involve different ASes provinding the service

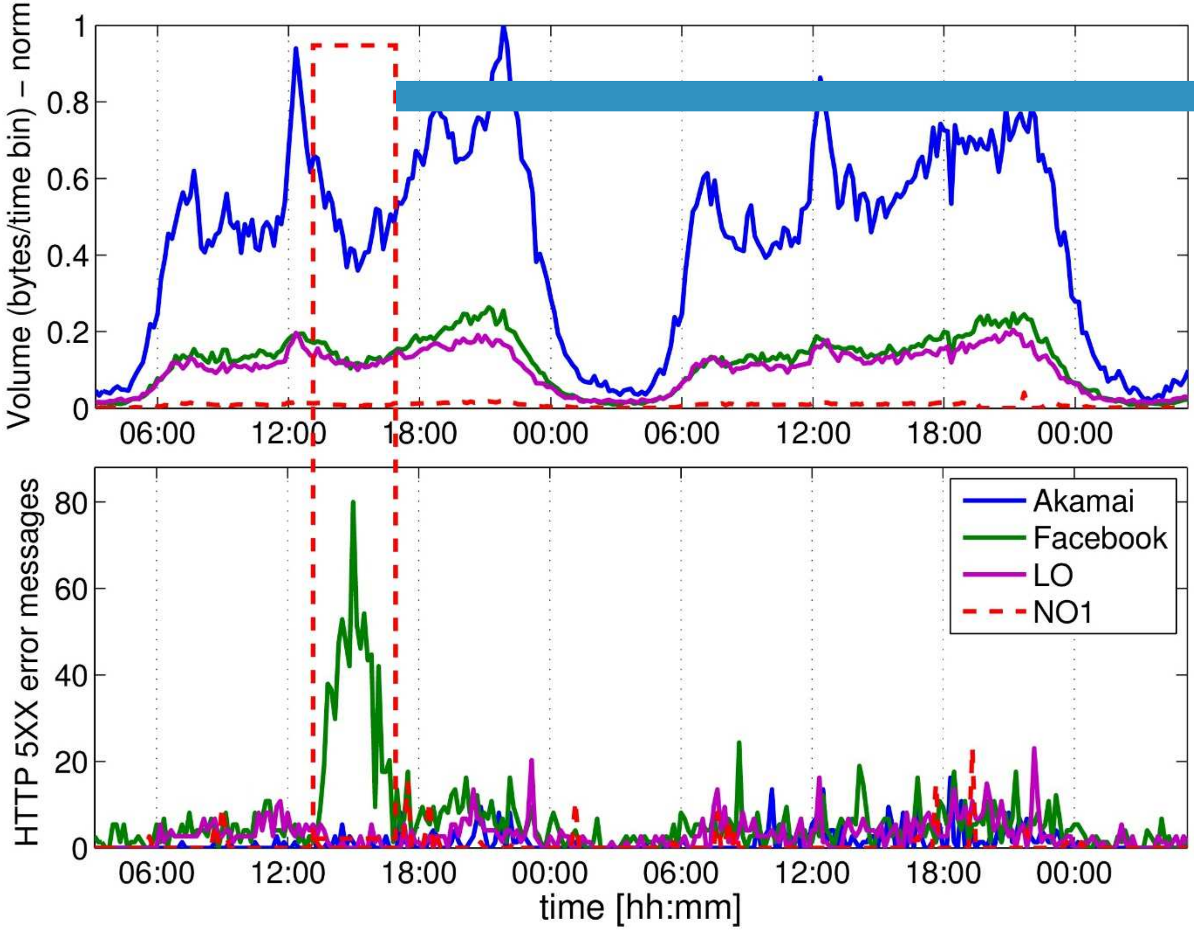


- Drop in Akamai's traffic volume
- Other ASes do not take over

- Increased number of 5XX server errors
- Users experience service impairment

Detecting Facebook Outages
October 2013

- Similar outages after exactly 1 month
- Officialy reported by Facebook



- Same outage characterisics as before

QoE degradations in YouTube

Detecting and Diagnosing QoE–based Anomalies

Another testbed
Youtube

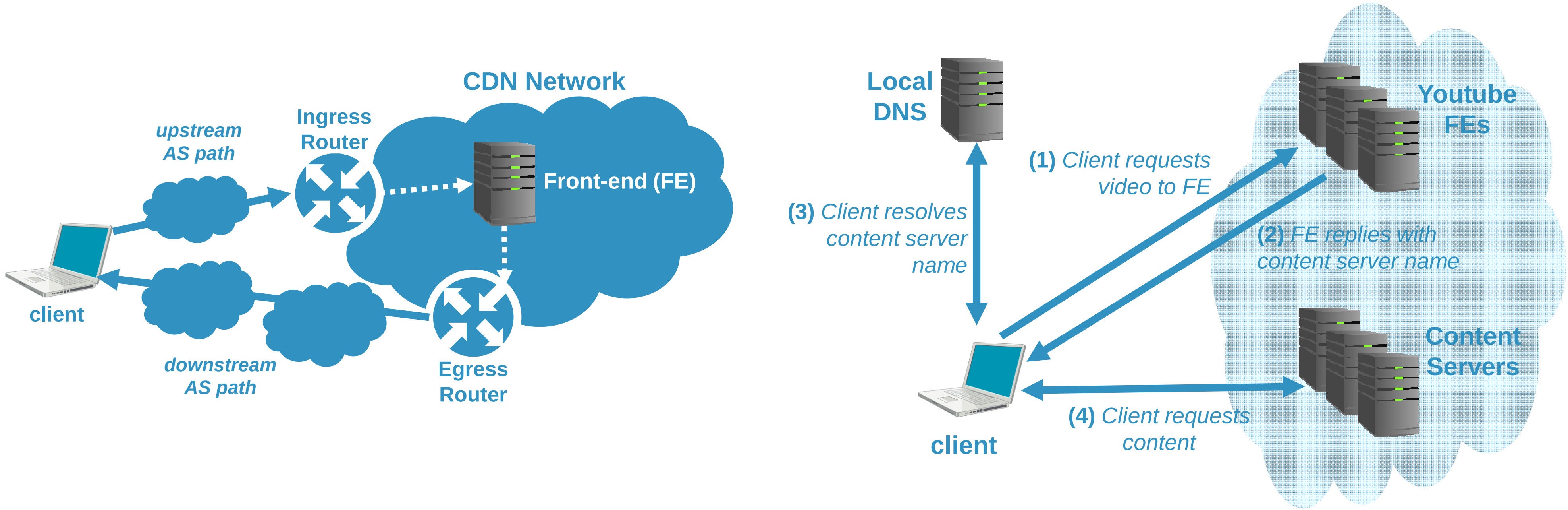
- Largest content provider

- Very complex hosting infrastructure:
  - Load balancing
  - Optimal QoE

Research questions:
1. How Youtube traffic looks like as seen from our passive Vantage Point?
2. Where its traffic is coming from?
3. Do users always get optimal QoE?

A typical CDN architecture
Google CDN for Youtube

- Google CDN employes a complex server selection strategy for:
  - load balancing
  - optimize client-server latency
  - increase QoE in general
- DNS used for re-direction based on content popularity and location.



Youtube load-balancing

- DNS-driven users redirection
- Goals:
  - Load balancing
  - Optimize choice of content servers aimed at reduce latency for clusters of users (cluster: <AS,country>)
- Is it always optimal? Look at the next example...

Anomalies/Changes impacting QoE

Requestes served by different /24 subnets

..which correspond to different data centers

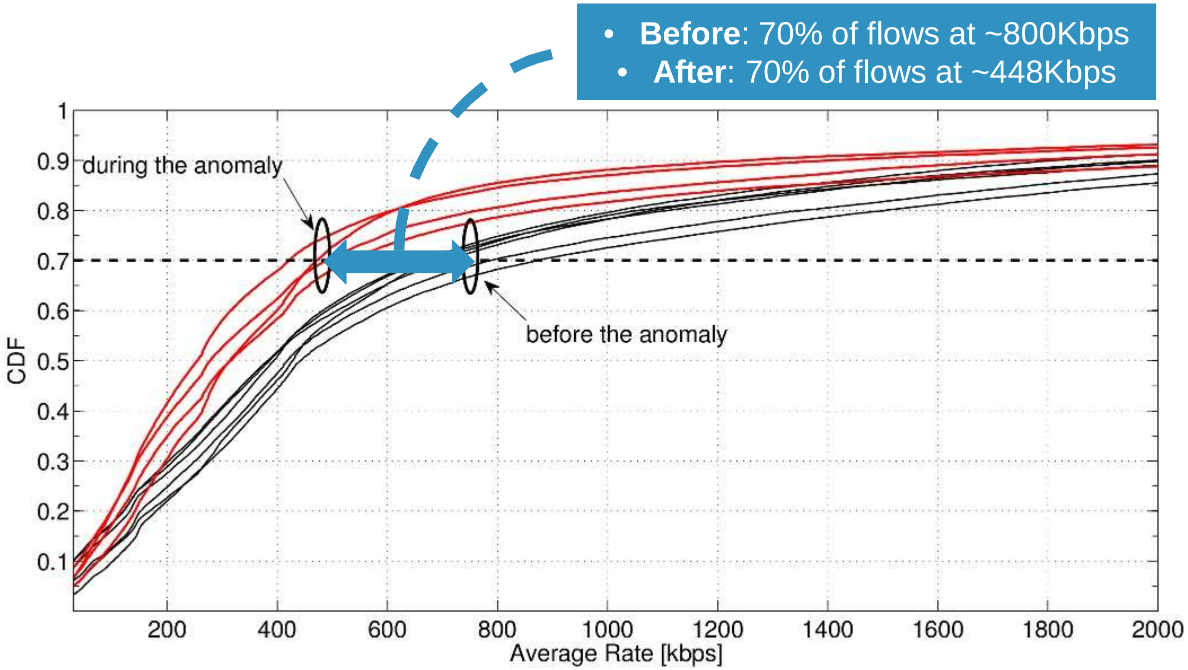
- Measurement … om a single vantage point (European fixed-line ISP)

		5-May		6-May		7-May	
SUBNET	NAME with AIRPORT code	#flow	Tru avg	#flow	Tru avg	#flow	Tru avg
173.194.18	fra02s08.c.youtube.com	-1	-1	-1	-1	-1	-1
173.194.19	fra02s15.c.youtube.com	-1	-1	-1	-1	-1	[illegible]
173.194.2	mil01s12.c.youtube.com	17054	1333.46	15470	1276.31	13655	**125[illegible].63**
173.194.20	par08s06.c.youtube.com	-1	-1	-1	-1	-1	-1
173.194.208	par08s06.c.youtube.com	-1	-1	-1	-1	-1	-1
173.194.5	lhr14s08.c.youtube.com	449	1819.57	283	1658.45	-1	-1
173.194.6	fra07s13.c.youtube.com	-1	-1	-1	-1	-1	[illegible]
173.194.62	fra07s19.c.youtube.com	-1	-1	-1	-1	-1	-1
173.194.9	par03s06.c.youtube.com	-1	-1	-1	-1	-1	-1
208.117.236	par03x04.c.youtube.com	179	164.18	4250	540.16	957	496.91
208.117.248	mia02s11.c.youtube.com	-1	-1	77	552	-1	-1
208.117.250	ams09x06.c.youtube.com	41430	679	49437	656.39	57675	**653.81**
208.117.252	dfw06x02.c.youtube.com	-1	-1	51	285.63	-1	-1
208.117.254	fra07x03.c.youtube.com	838	667.29	2130	852.53	-1	-1
74.125.105	lhr22s16.c.youtube.com	1829	1551.78	1655	1185.94	3957	[illegible]42.47
74.125.13	zrh04s03.c.youtube.com	719	1074.15	499	2264.09	82	[illegible]3[illegible]2.03
74.125.14	mil02s01.c.youtube.com	48366	1234.82	37968	1253.01	37182	**1162.[illegible]**
74.125.216	bru02t11.c.youtube.com	-1	-1	-1	-1	-1	-1
74.125.218	fra07t13.c.youtube.com	8697	1355.33	12579	1338.71	8560	[illegible]1239
74.125.4	lhr22s11.c.youtube.com	1496	1846.25	2488	1034.78	4146	**1363.63**
74.125.99	fra07s03.c.youtube.com	-1	-1	-1	-1	-1	-1

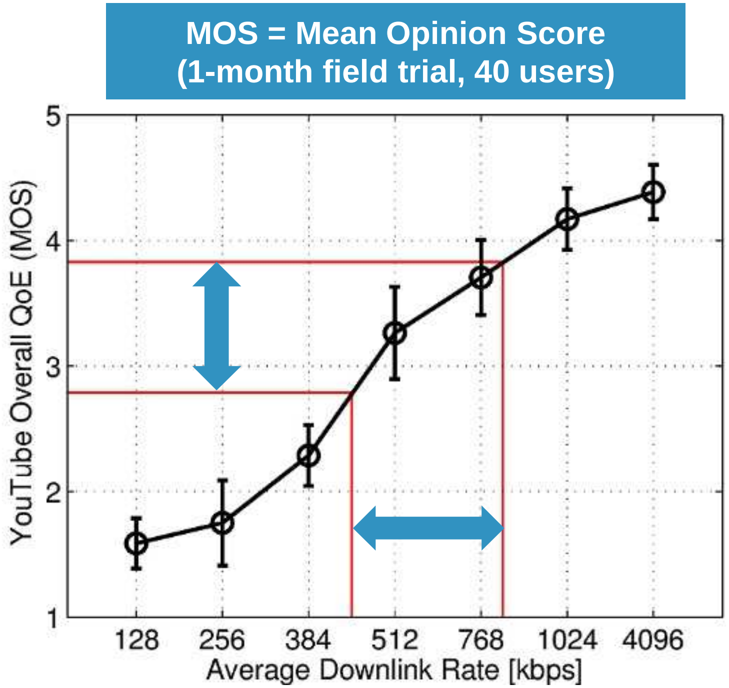
YouTube

Anomalies/Changes impacting QoE
Degradation of the Average Download Rate

- Google's cache selection policies might be sub-optimal!
- Let's consider a **real example**:



CDFs at peak hours (9pm – 11pm)
before and after change in cache selection policy



Avg. Downlink Rate and **QoE**

Loss of 1 point (MOS from 4 to 3) in Quality of Experience

customers noticed and **complained** with the operator

Anomalies/Changes impacting QoE

Correlating Throughput and Video Bit Rate

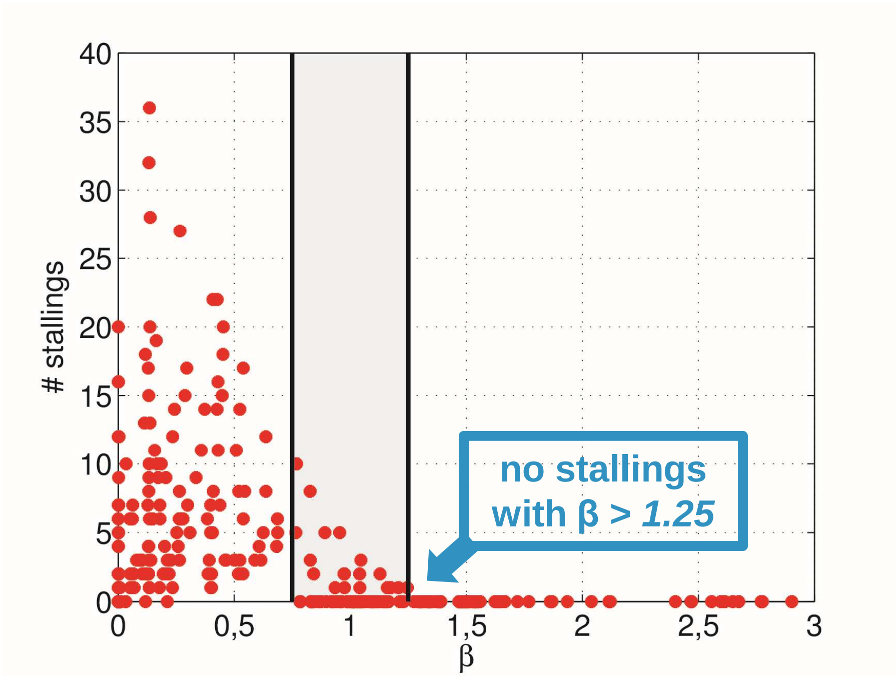
ADT = Average Download Throughput

VBR = Video Bit Rate

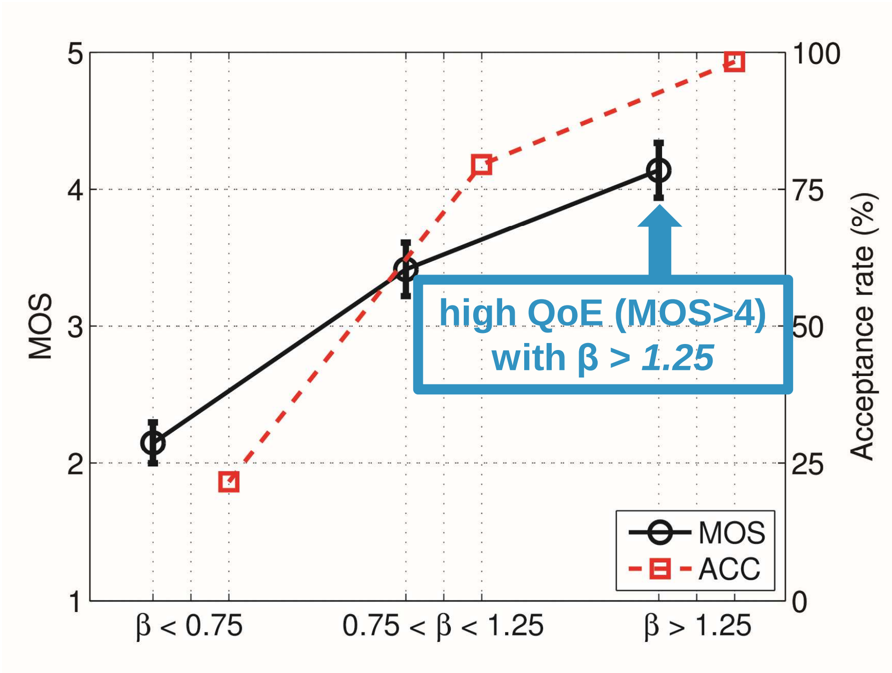
β = ADT/VBR

(metric reflecting user experience)

Idea: if **ADT<VBR** → low **β** and *video stallings* (=low QoE)



β-parameter vs. video stallings



MOS and acceptance vs. **β**-parameter

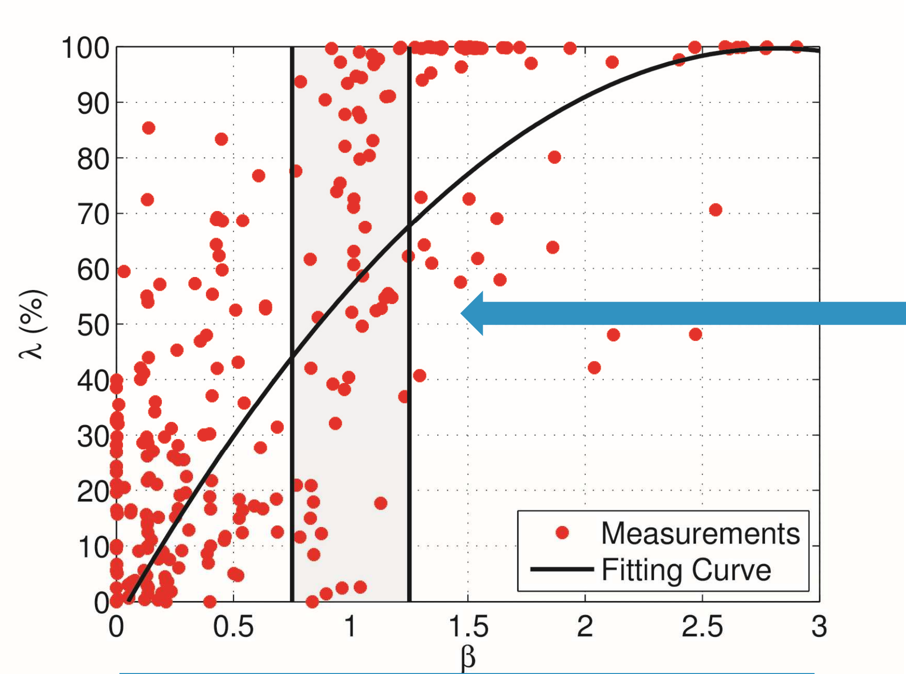
Anomalies/Changes impacting QoE
User engagement

- Optimal user experience with **β**>1.25 (VBR=360p, ADT=750kbps)
- Any relation between **β** and **user engagement**?

VPT = Video Played Time
VD = Video Duration

λ = VPT/VD

(fraction of video actually viewed)



β-parameter vs. **λ-parameter**

High user engagement (λ) with optimal value of β

Anomalies/Changes impacting QoE
Diagnosis

Multiple possilble **root causes** for the anomaly:

- Problems at the access network

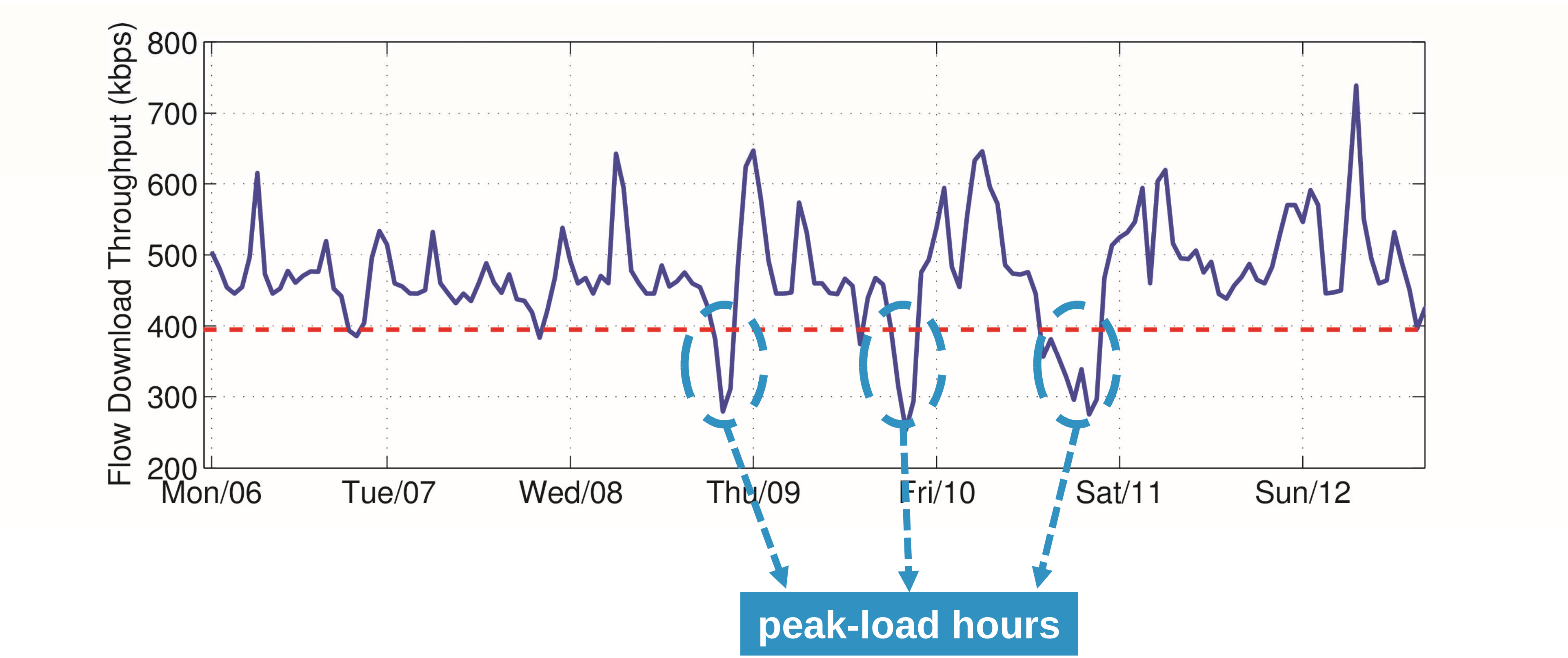
operator domain

- Faulty cache selection strategy by Google
- Youtube content servers are overloaded
- Path between users and servers suddenly changes
- Path is congested

outside operator boundaries

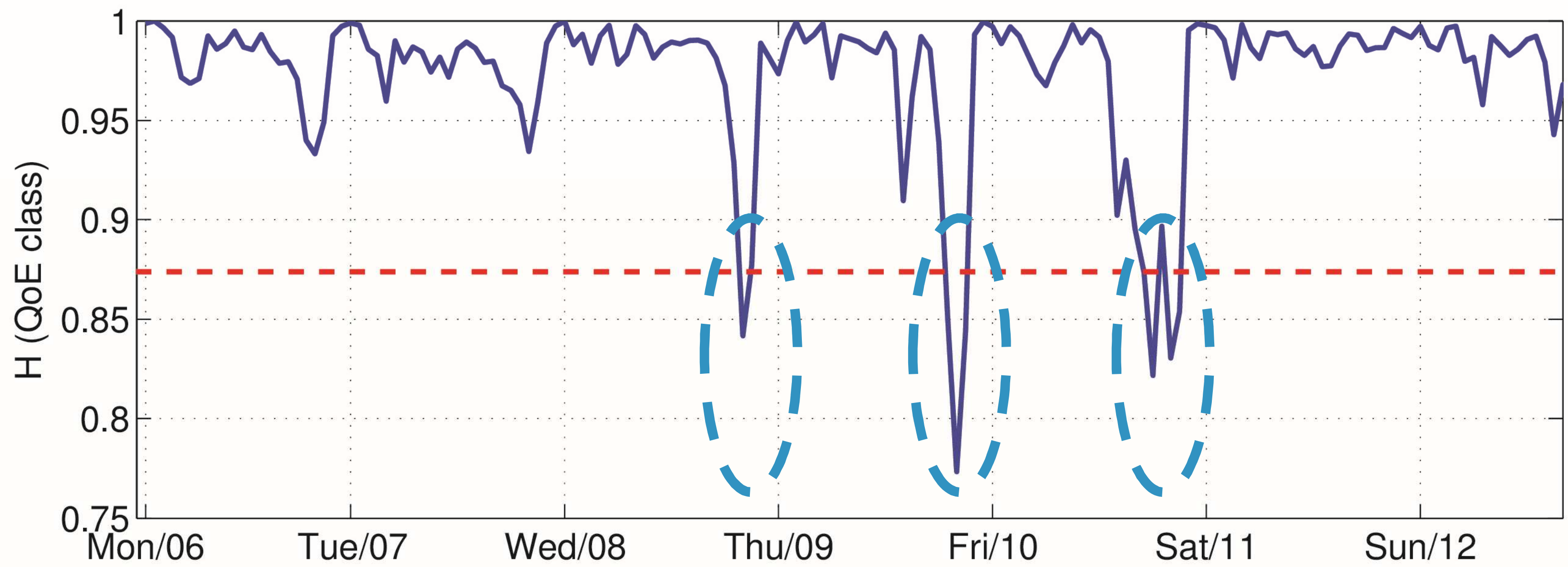
Detection
threshold based: throughput loss

- Clear degradation of achieved throughput from Wednesday afternoon



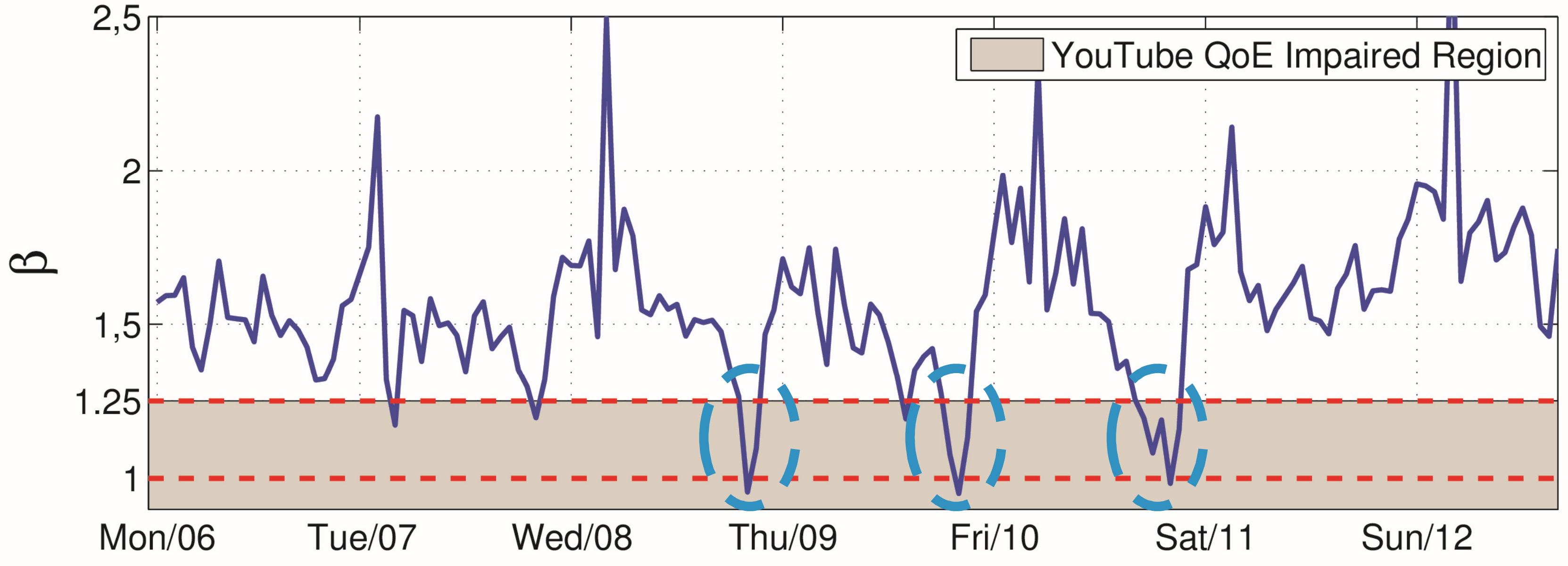
Detection
entropy based

- Drop in the entropy of the QoE (MOS) classes
  - i.e. fewer classes become predominant



Detection
threshold based: β-parameter loss

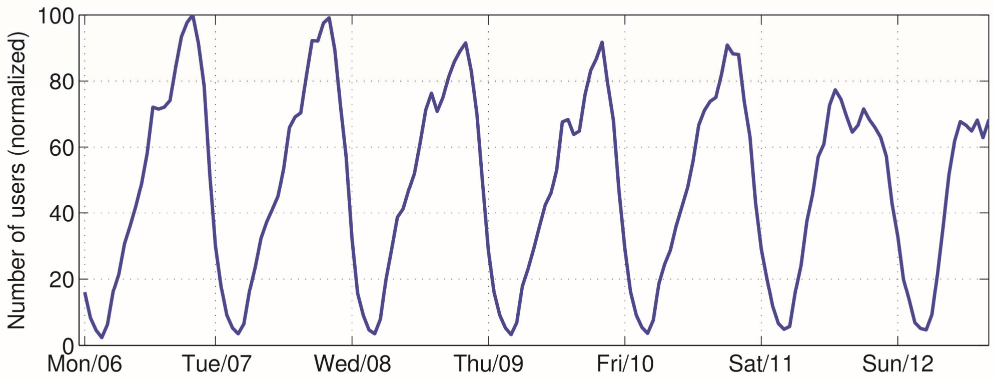
- Drop in the entropy of the QoE (MOS) classes
  - i.e. fewer classes become predominant



Remember: $\beta<1.25$ means video stallings and low QoE

Diagnosis #1
drops in the number of users or downlink bytes?

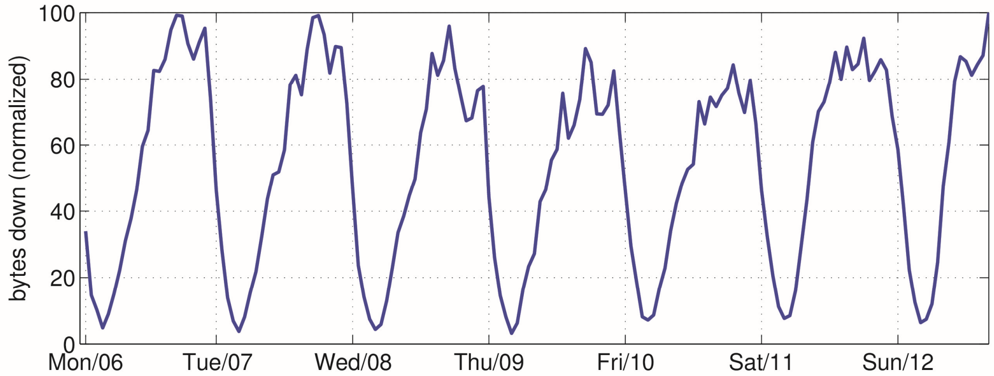
~NO



Time series: number of users

- No significant variations in the number of users (during working days)

Conclusion 1
Throughput/QoE variations **are not** tied to statistical variations of the sample size



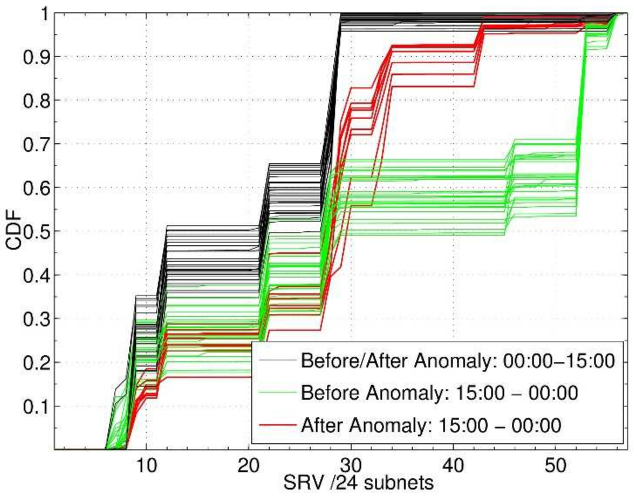
Time series: downlink bytes

- Slight decrease in number of bytes (from Wednesday)

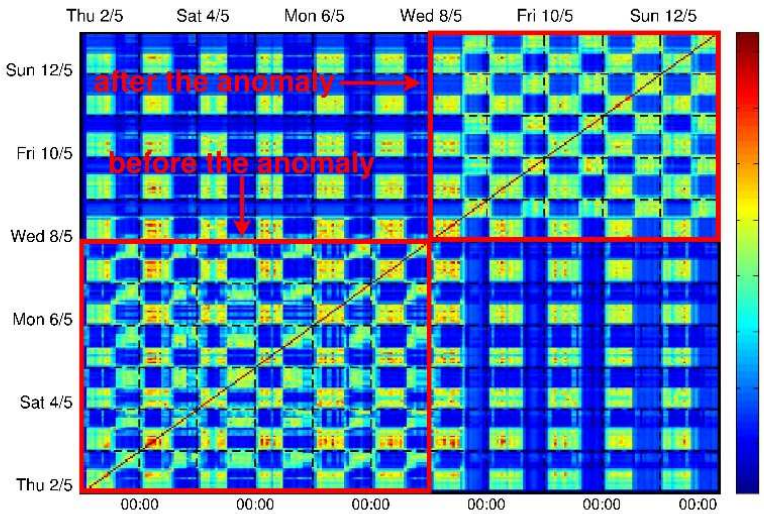
Conclusion 2
Maybe due lower user engagement
(remember β <1.25)

Diagnosis #2
change in the cache-selection policy?

YES



traffic distribution among /24 subnets



Temporal Similarity Plot (TSP)

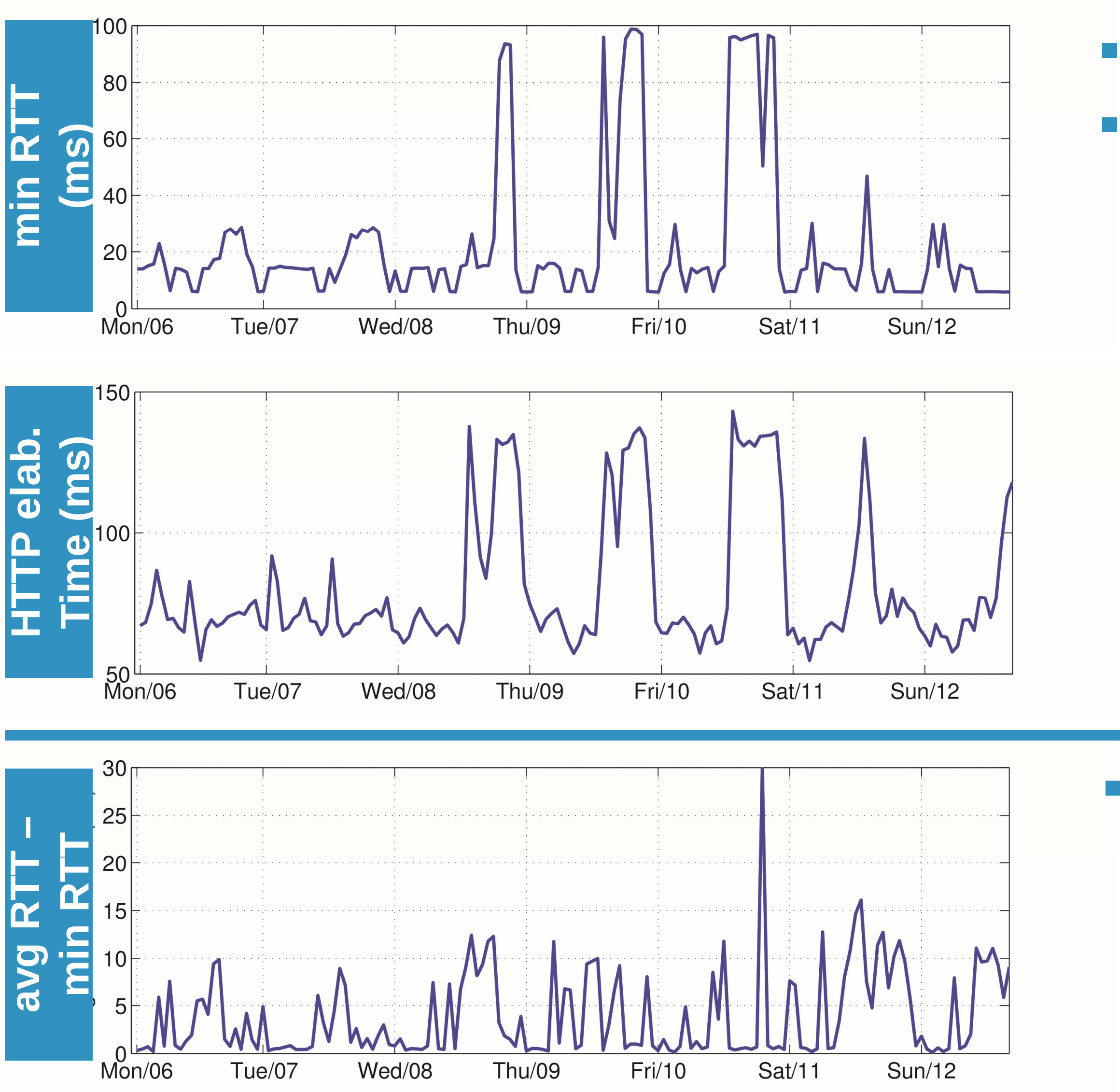
- Sharp shift from AS15169 to AS 43515 during peak hours
- Reduction of servers selected from AS43515 on the days of the anomlay

Conclusion

A **different server selection** policy is set up exactly on the same day when the anomaly occurs!

Diagnosis #3

given that change... who is to blame: new servers or path?



- Marked increase in min RTT
- As a consequence: marked increase in the HTTP eleboration time

Conclusion 1
Newly selected server are farther

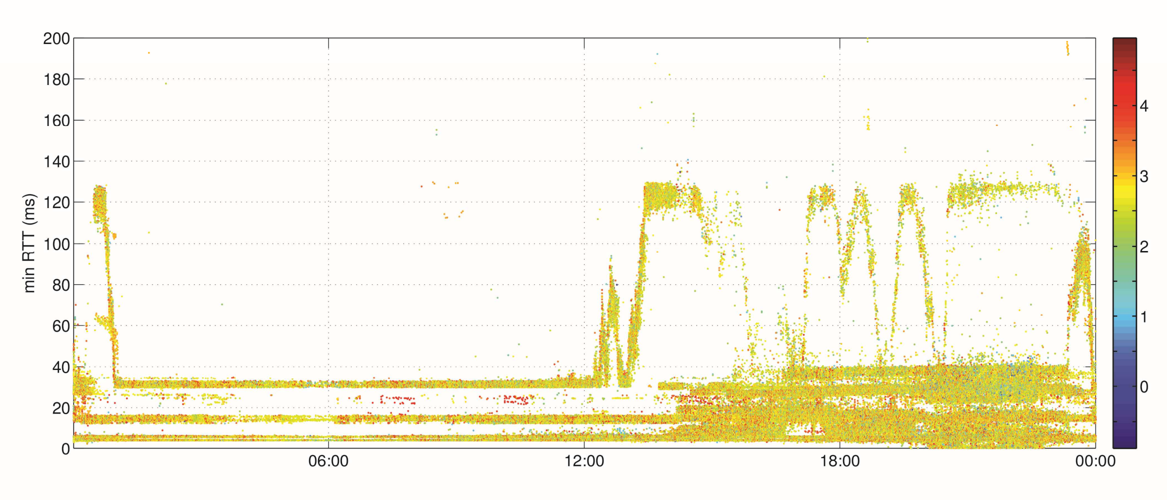
- No significant changes in avg RTT

Conclusion 2
No apparent path congestions (also, no increase of RTX number)

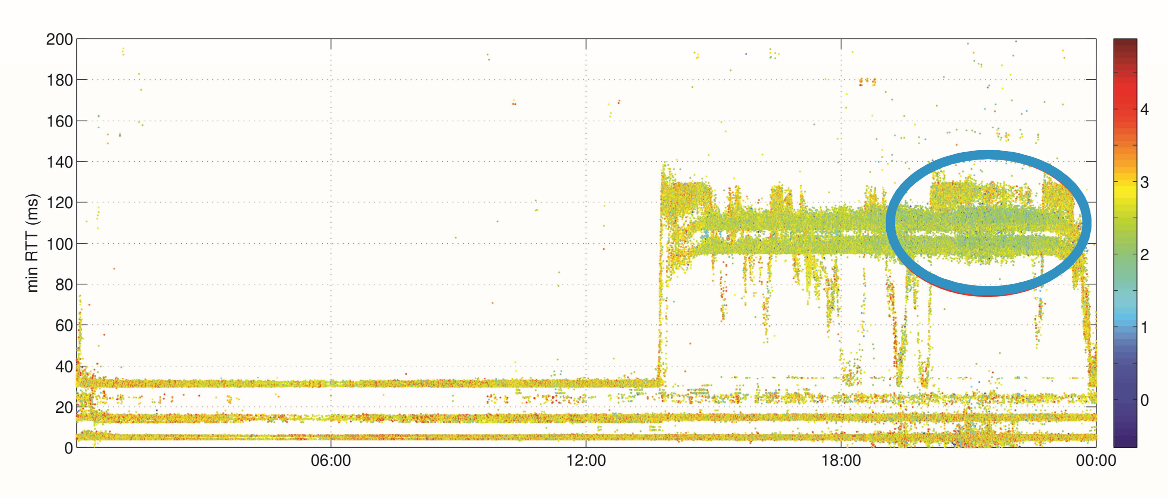
Diagnosis #4

is there a correlation between increased RTT and throughput?

NO



min RTT and avg. dowload rate (before)



min RTT and avg. dowload rate (after)

- Large min RTT does not imply low throughput

Conclusion

The increase of min RTT **is not** the root cause of the anomaly!

Youtube Anomaly Diagnosis Conclusion

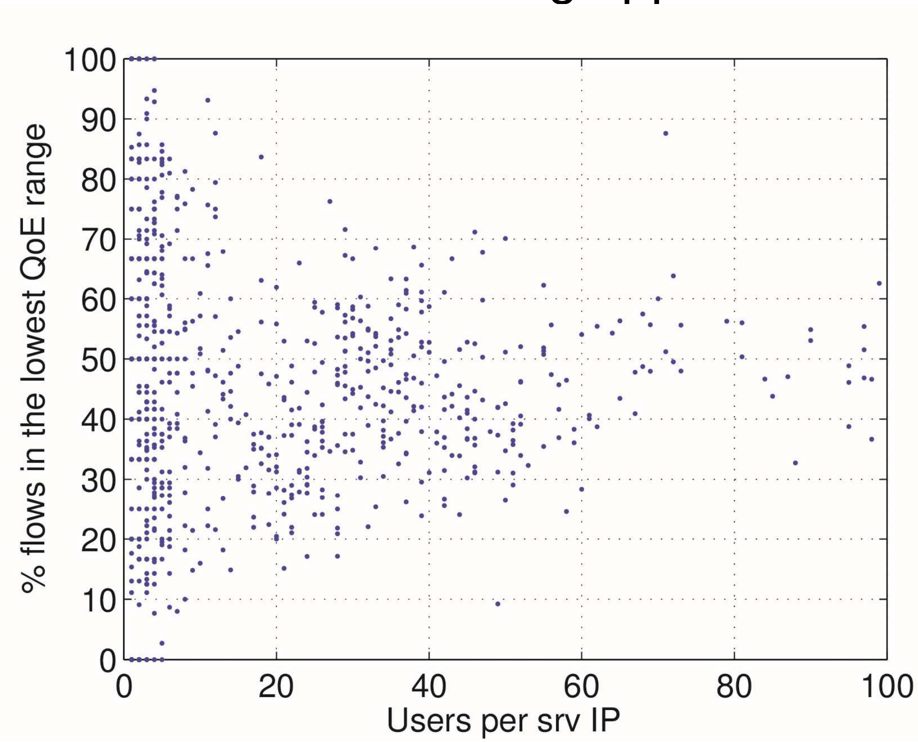
- The origin of the anomaly is the cache selection by Google
  - Additional selected servers from 15:00 to 00:00 are under dimensioned for peak hours (20:00 – 23:00)
  - Dynamics of Google selection policies might result in poor end-to-end experience
    1. Servers unable to handle load at peak hours
    2. Not considering end-to-end path performance

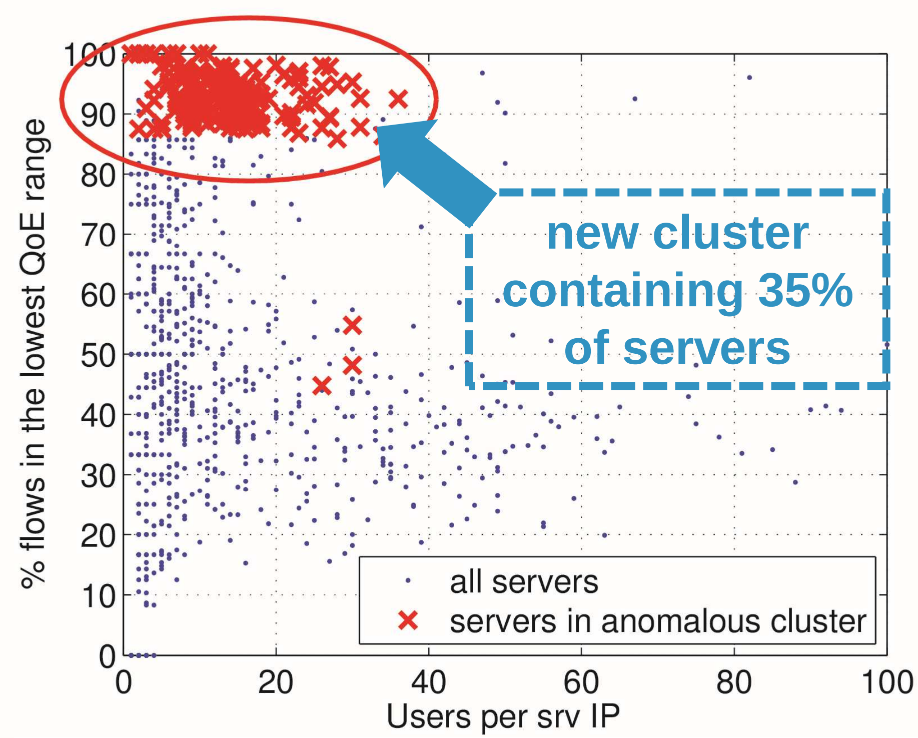
Make it unsupervised, please
Density-based clustering

- Characterize every Youtube server with a set of features (seen before)

#flows | **#bytes** | **#users** | **avg throughput** | **H(QoE classes)** | **min RTT** | **avg RTT**

- Track the evolution of the traffic structure over time through the DBSCAN clustering approach





ftw Creating Communication Technologies

Unsupervised Detection of Attacks

Current Detection of Network Attacks

Security is based on an **"acquired knowledge"** perspective:

Signatures-based: detect what I ALREADY KNOW

- (+) highly effective to detect what it is programmed to alert on.
- (−) can not defend the network against unknown attacks.
- (−) signatures are expensive to produce: human manual inspection.

Anomaly detection: detect what DIFFERS from WHAT I KNOW

- (+) it can detect new attacks out-of a baseline profile.
- (−) requires some kind of training for profiling.
- (−) robust and adaptive models are difficult to conceive, specially in an evolving context.

Unsupervised Detection of Network Attacks

- Unsupervised Detection based on CLUSTERING
- HYPOTHESIS: attacking flows are sparse and different from normal traffic....in some representation (traffic aggregation)!!

Benefits of Unsupervised-based Detection

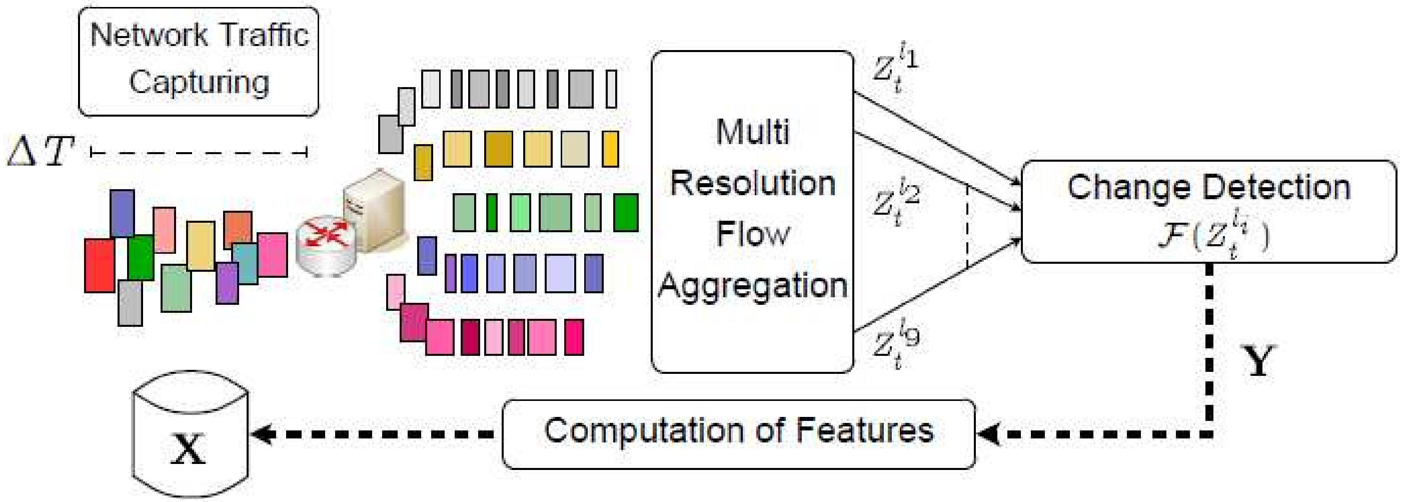
- no previous knowledge: neither signatures nor labeled traffic.
- no need for traffic modeling or profiling.
- can detect unknown attacks.
- a major step towards self-aware monitoring (0-day attacks).

Clustering for Unsupervised Detection is CHALLENGING

- lack of robustness: general clustering algorithms are sensitive to initialization, specification of number of clusters, etc.
- difficult to cluster high-dimensional data: structure-masking by irrelevant features, sparse spaces ("the curse of dimensionality").

UNIDS: Unsupervised Network Intrusions Detection

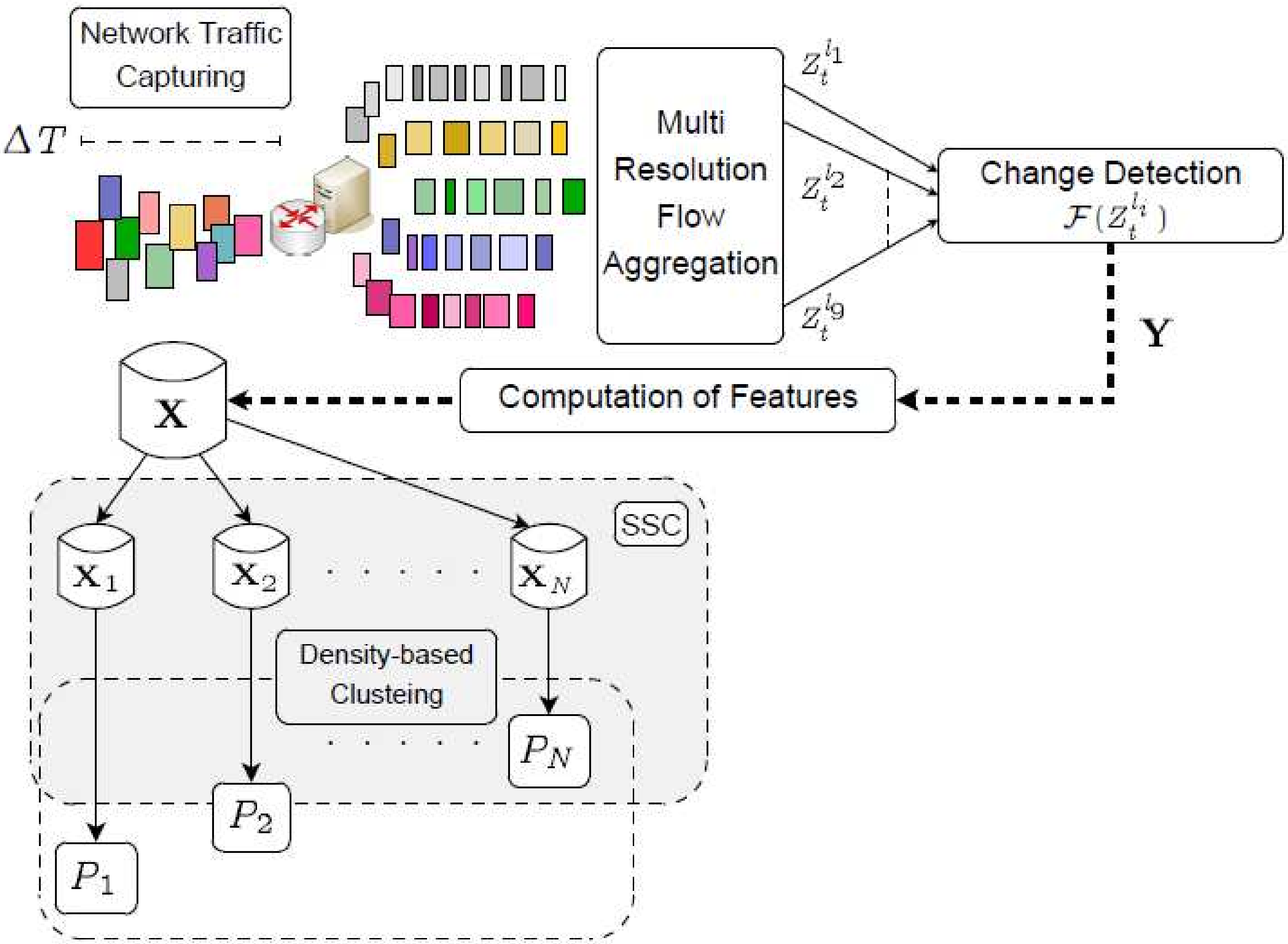
UNIDS is a 3-steps detection system:



(1) Multi-resolution change-detection & features computation.

UNIDS: Unsupervised Network Intrusions Detection

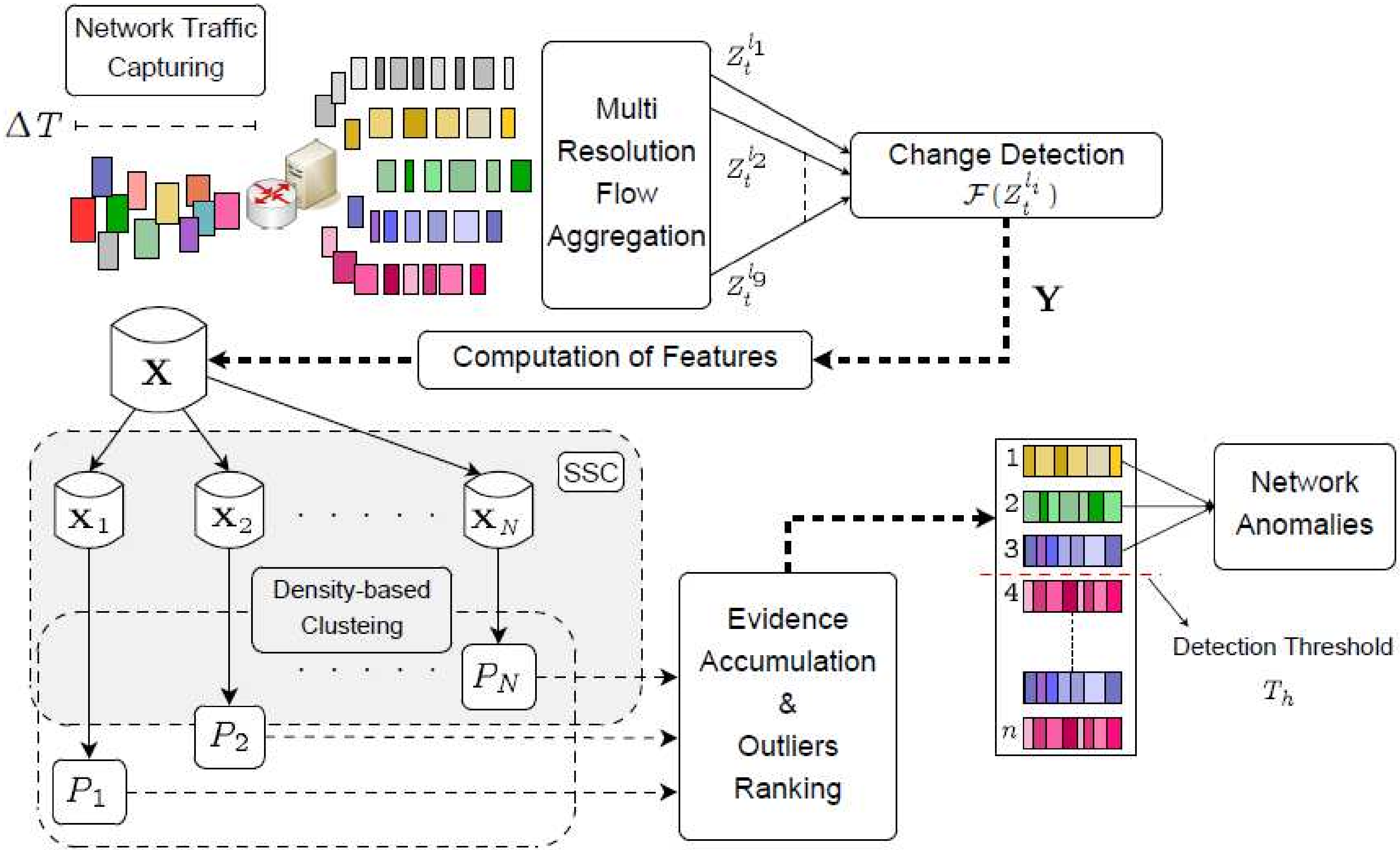
UNIDS is a 3-steps detection system:



(2) Sub-Space Clustering.

UNIDS: Unsupervised Network Intrusions Detection

UNIDS is a 3-steps detection system:



(3) Evidence Accumulation and Flow Ranking.

Change-detection in Multi-resolution Traffic Flows

Traffic Aggregation and Change-Detection

- Traffic is captured and aggregated in IP flows (5-tuples) every ΔT seconds, using a temporal sliding-window.
- Change-detection in simple traffic metrics to identify an anomalous time-slot (e.g., $\#pkts$, $\#bytes$, $\#IP\ flows$).

Multi-Resolution Analysis

- Analysis at different spacial resolutions, aggregating IP flows in *macro-flows*: hash-key $\{\mathrm{IPaddress}/\mathrm{netmask}\}$.
- Scan traffic from coarser to finer-grained macro-flows: $\mathrm{traffic\ per\ time\text{-}slot}$, $\mathrm{IP}/8$, $\mathrm{IP}/16$, $\mathrm{IP}/24$.
- Scan in both directions (IPsrc and IPdst) permits to detect $1\text{-to-}1$, $1\text{-to-}N$, and $N\text{-to-}1$ attacks of different intensities.

Clustering for Anomaly Detection

- Let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be the set of n macro-flows in the flagged time slot, aggregated at $\mathrm{IP}/32$.
- Each macro-flow $\mathbf{y}_i \in \mathbf{Y}$ is described by a set of m traffic features: $\mathbf{x}_i = (x_i(1), .., x_i(m)) \in \mathbb{R}^m$.
- Number of sources & destinations $(\mathrm{nSrcs}, \mathrm{nDsts})$, packet rate $(\mathrm{nPkts}/\mathrm{sec})$, fraction of SYN packets $(\mathrm{nSYN}/\mathrm{nPkts})$, etc.
- $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_n\}$ is the complete matrix of features, referred to as the *feature space*.

Clustering for Anomaly Detection

$\mathbf{X}$ is a black box

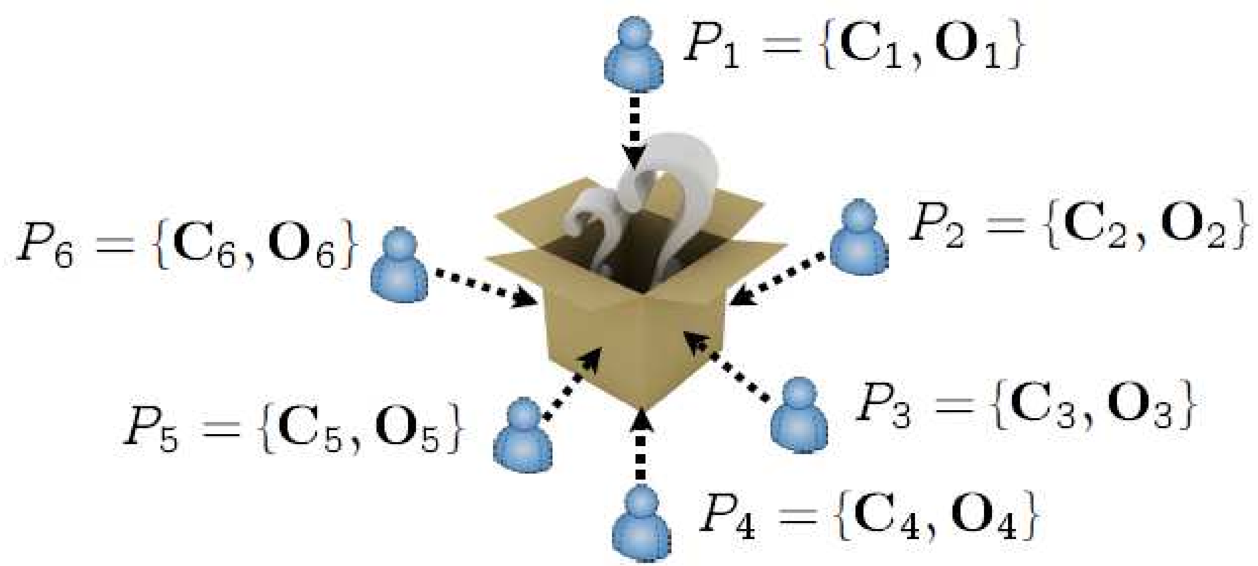
How to detect an anomalous macro-flow in $\mathbf{X}$ via clustering?

- "Simple idea": cluster $\mathbf{X}$, big-size clusters correspond to normal-flows, outliers are anomalies.

Sub-Space Clustering

How to Improve Robustness and Clustering Performance?

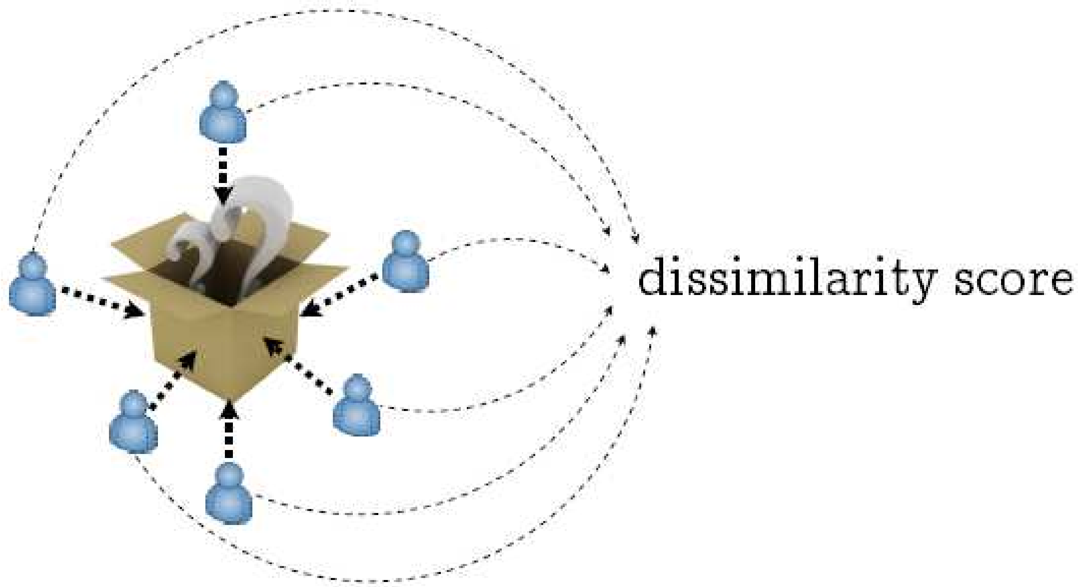
- Idea: combine the information provided by multiple partitions of $\mathbf{X}$ to “filter noise”, easing the discovery of outliers.
- How to produce multiple partitions? $\rightarrow$ Sub-Space Clustering.
- Each sub-space $\mathbf{X}_i \subset \mathbf{X}$ is obtained by projecting $\mathbf{X}$ in k out of the m original dimensions. Density-based clustering applied to $\mathbf{X}_i$.



Sub-Space Clustering

How to Improve Robustness and Clustering Performance?

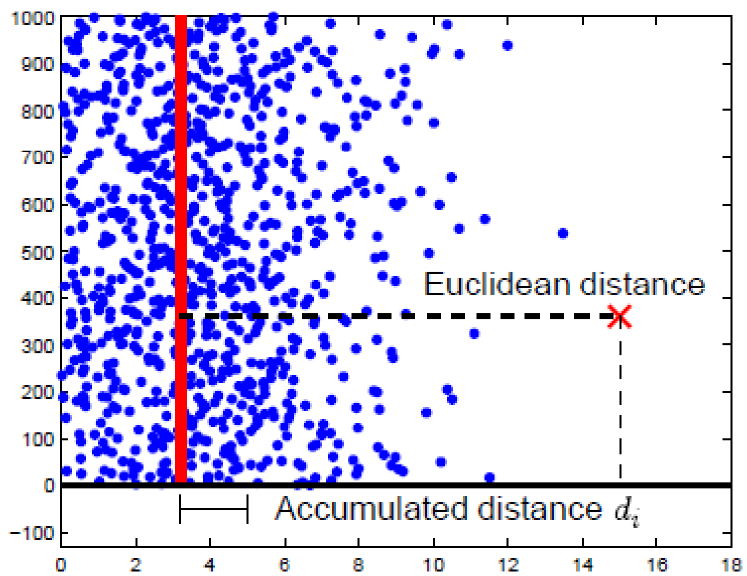
- Idea: combine the information provided by multiple partitions of $\mathbf{X}$ to “filter noise”, easing the discovery of outliers.
- How to produce multiple partitions? $\rightarrow$ Sub-Space Clustering.
- Each sub-space $\mathbf{X}_i \subset \mathbf{X}$ is obtained by projecting $\mathbf{X}$ in k out of the m original dimensions. Density-based clustering applied to $\mathbf{X}_i$.

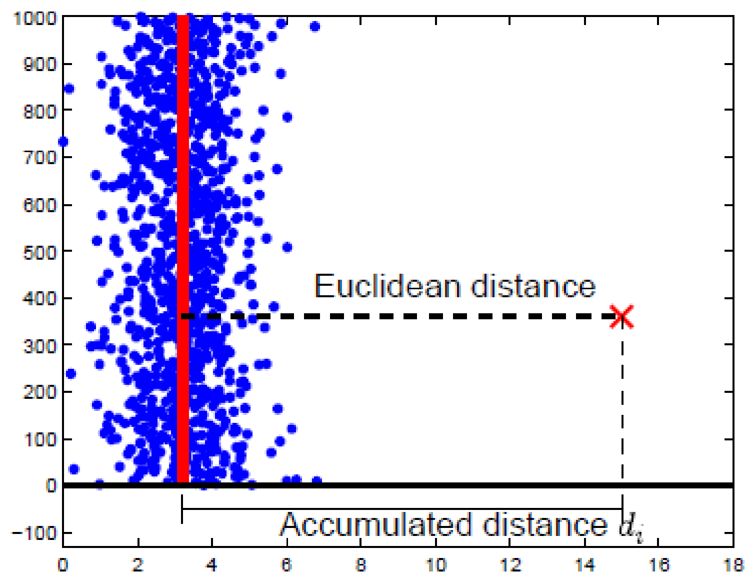


Evidence Accumulation for Outliers Ranking

Evidence Accumulation to combine the results of SSC:

- Build a new dissimilarity measure $D = \{d_1, d_2, \ldots, d_n\}$: d_i measures how different is flow i from the majority of the traffic.
- Accumulate in d_i the *weighted* distance from outliers to biggest cluster in each sub-space.
- Most dissimilar flows w.r.t. D are flagged as anomalies.

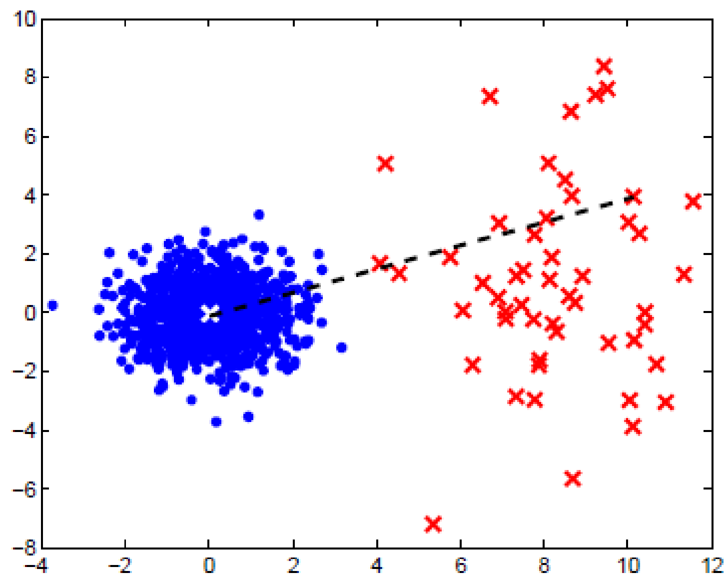


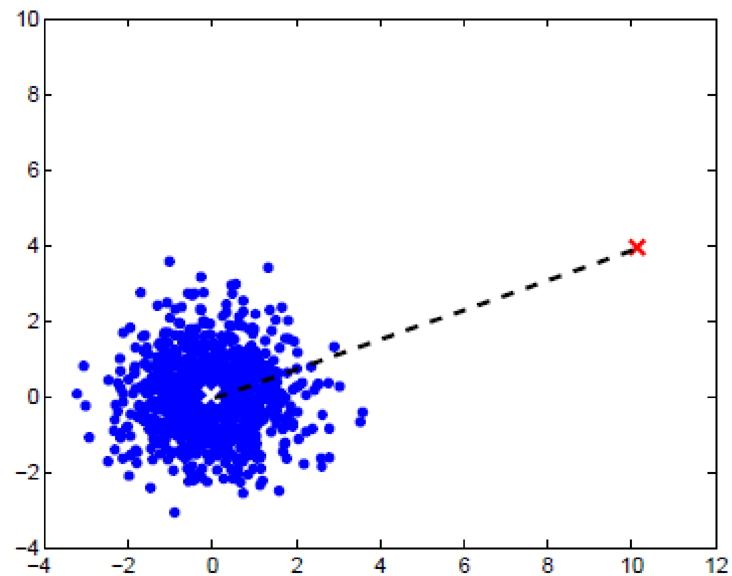


Evidence Accumulation for Outliers Ranking

Evidence Accumulation to combine the results of SSC:

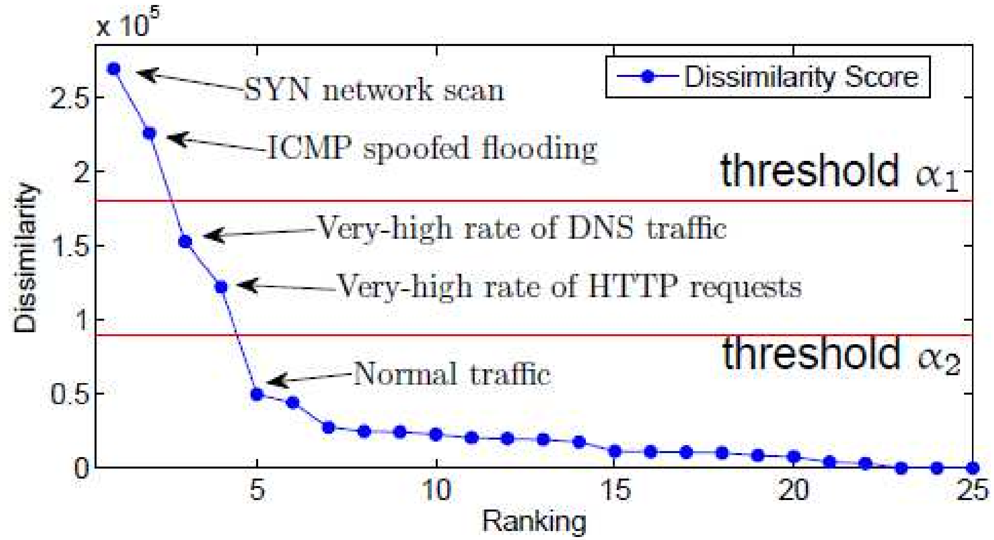
- Build a new dissimilarity measure $D = \{d_1,\ d_2, \ldots, d_n\}$: d_i measures how different is flow i from the majority of the traffic.
- Accumulate in d_i the *weighted* distance from outliers to biggest cluster in each sub-space.
- Most dissimilar flows w.r.t. D are flagged as anomalies.

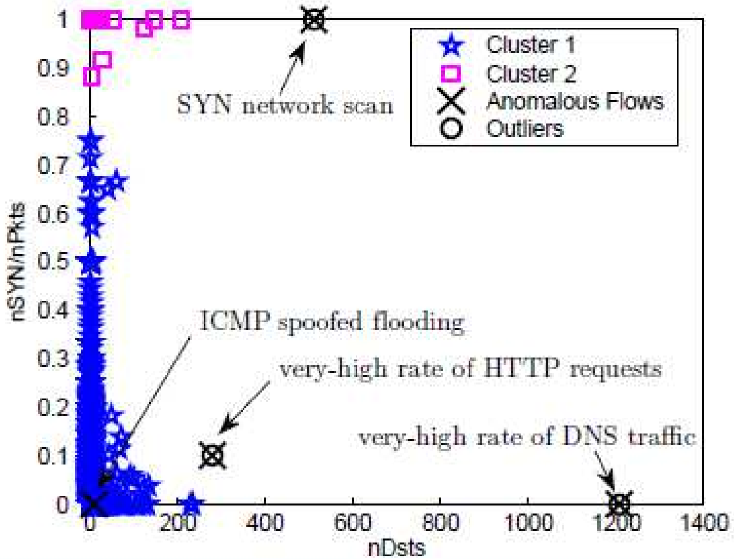


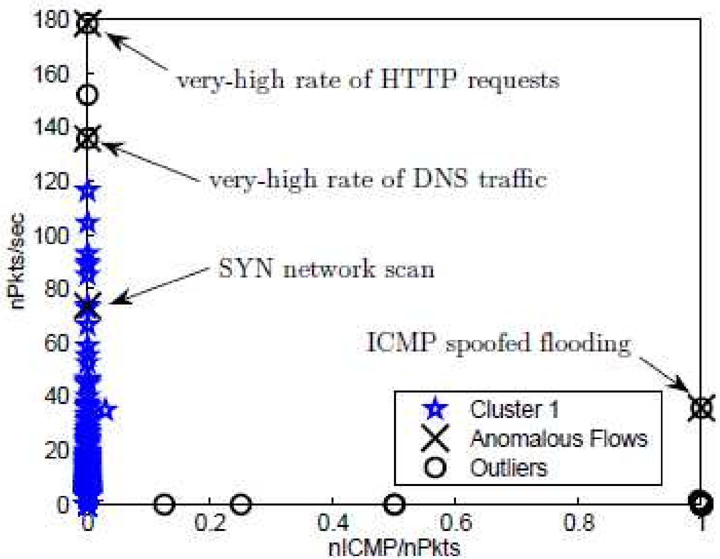


Attacks Detection in MAWI Traffic

- MAWI: packet traces from link Japan-U.S.A. of the WIDE network.
- Ex: worm scanning, ICMP flooding attack, $\mathtt{IPsrc}/32$ macro-flows.

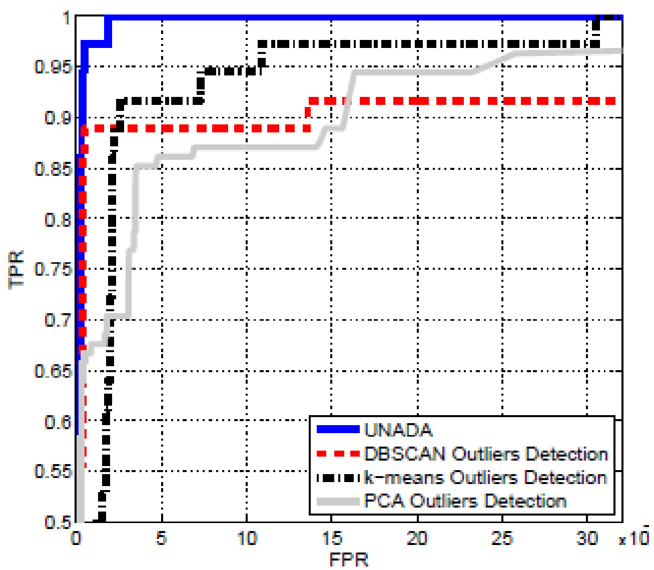




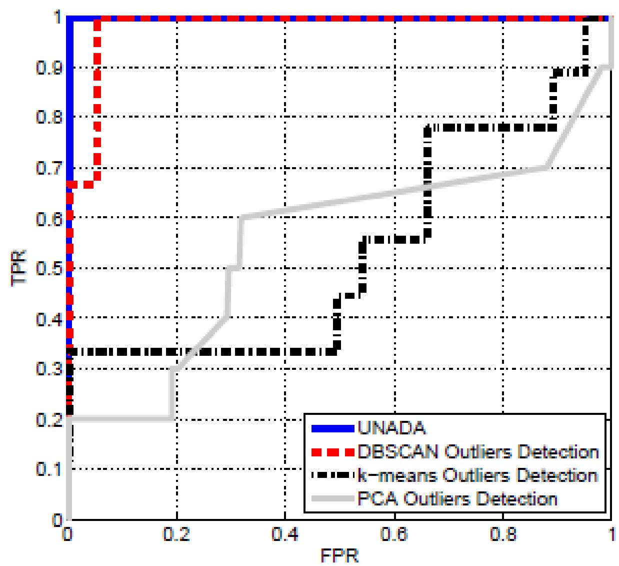


Ground-Truth (GT) Attacks in METROSEC & MAWI

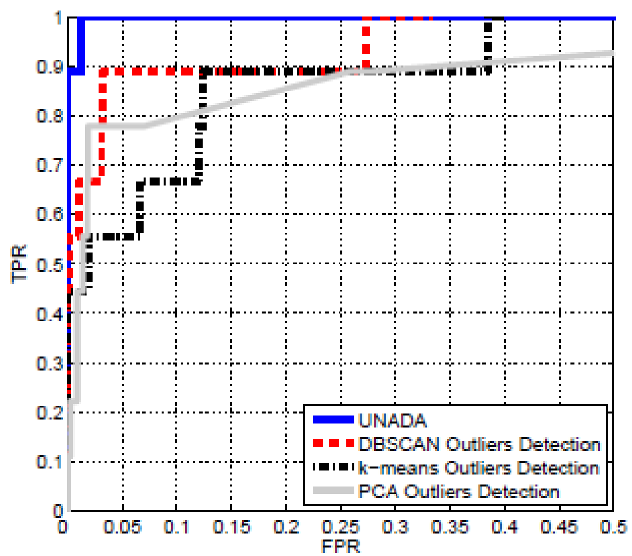
- METROSEC, DDoS attacks of different intensities (70% to 4%), IPdst/32 macro-flows.
- MAWI, worm scanning (Sasser and Dabber), DoS/DDoS attacks, GT attacks detected by signatures + Anomaly Detection.
- Compared against traditional unsupervised approaches: DBSCAN based, k-means based, and PCA based outliers detection.



(a) MAWI, IPsrc key.



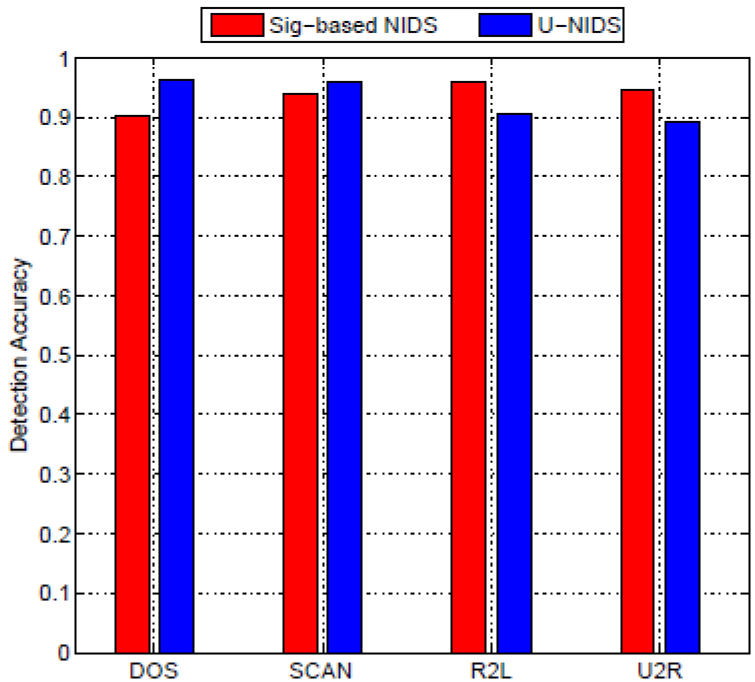
(b) MAWI, IPdst key.

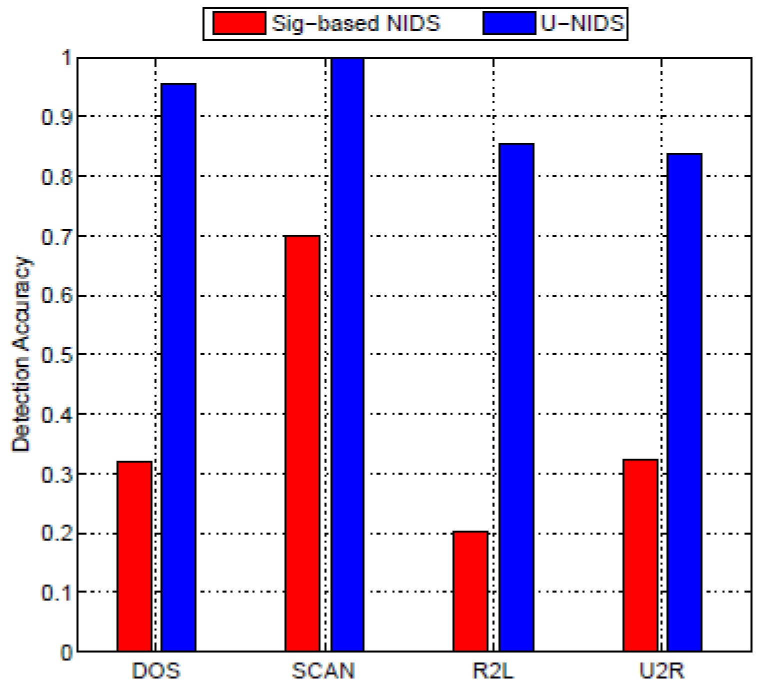


(c) METROSEC, IPdst key.

Detectiong Attacks in KDD99

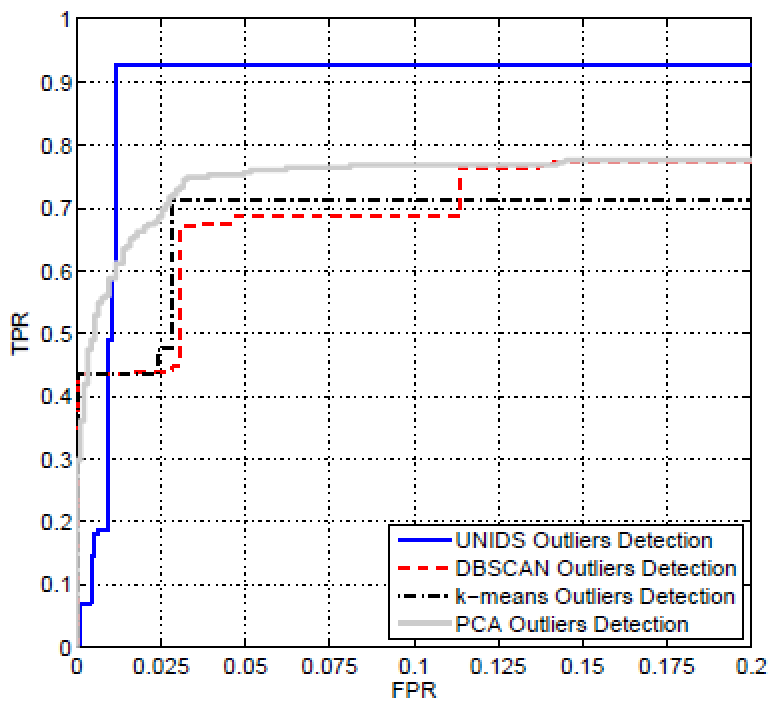
- DARPA - KDD99 dataset, DoS (udp storm, pod, appache flooding, etc.), scans (port, net), Remote-2-Local attacks (guess password, imap, http tunnel, etc.), User-2-Root (buffer overflows).
- Compared against signature-based detection $\rightarrow$ NIDS based on decision trees.
- Trees are constructed for a set of known attacks, and tested with *unknown* attacks.



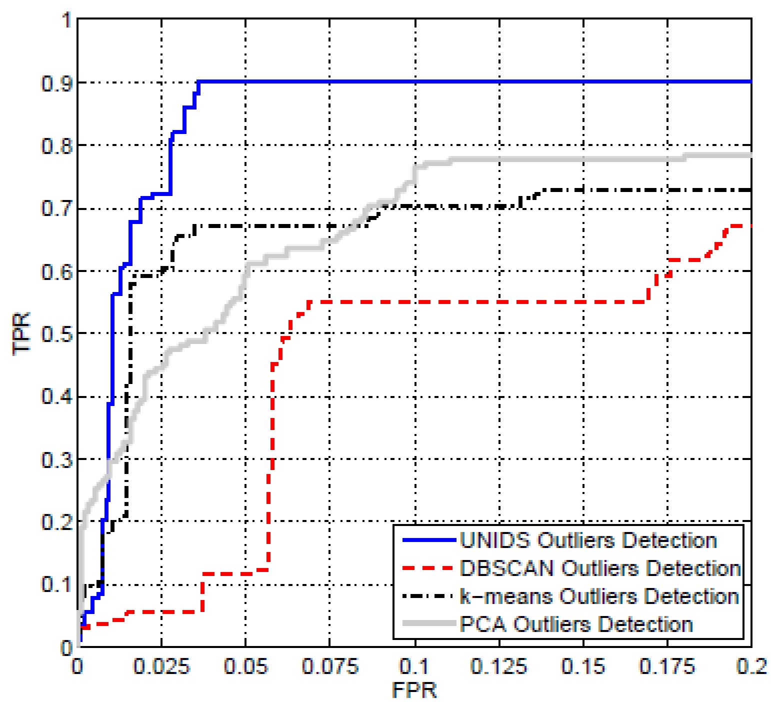


ROC Curves in KDD99

- True Positives Rate vs False Alarms in KDD99.
- Training and testing datasets contain different types of attacks.
- Compared against DBSCAN based, k-means based, and PCA based outliers detection.



(a) Training dataset.



(b) Test dataset.

Thanks You for Your Attention!

Pedro Casas, casas@ftw.at