Degree-based Outlier Detection within IP Traffic Modelled as a Link Stream

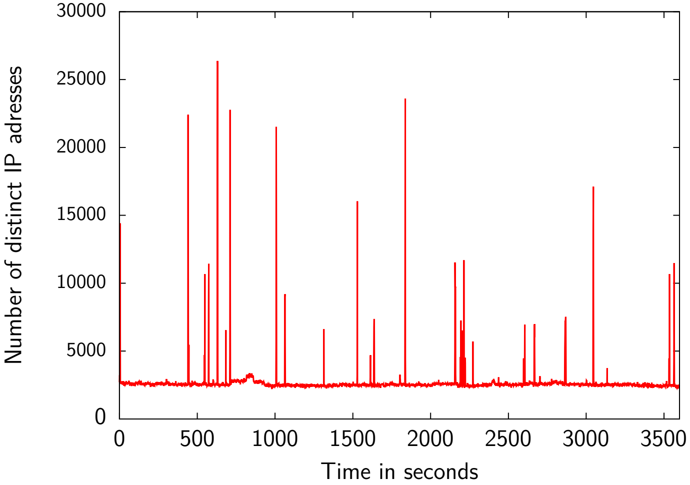
Audrey Wilmet[1], Tiphaine Viard[1],
Matthieu Latapy[1], Robin Lamarche-Perrin[2]

[1]Laboratoire d'informatique de Paris 6 (LIP6) - Complex Networks Team http://www.complexnetworks.fr/
[2]Institut des Systèmes Complexes Paris Île de France (ISC-PIF)

TMA Conference 2018, Vienna, Austria
June 26-29, 2018

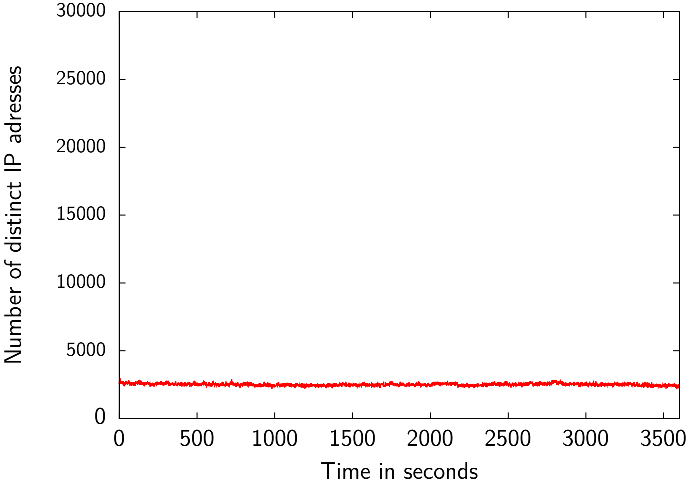
Context and Goals

Detect outliers, identify their cause, remove them from IP traffic:



Context and Goals

Detect outliers, identify their cause, remove them from IP traffic:



IP Traffic as a Link Stream

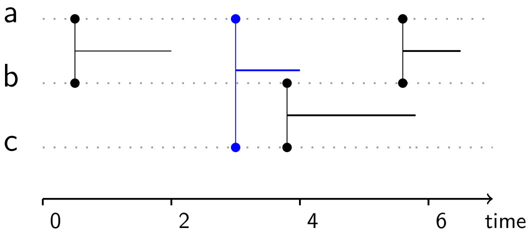
Link stream constructed from 1h of IP Traffic (MAWI):

– Nodes = IP addresses

– Interactions = packet exchanges

– Link stream construction:

Two nodes are linked together from time t_1 to time t_2 if they exchanged at least one packet every second within this time interval.



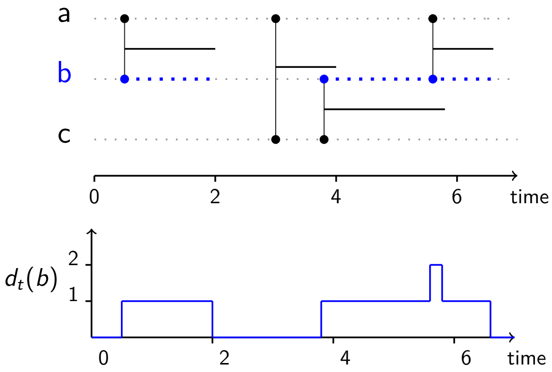
ex: nodes a and c interact from $t_1 = 3$ to $t_2 = 4$

→ M. Latapy *et al.*, 2017 ; T. Viard *et al.*, 2017.

Degree of (v, t)

$d_t(v)$ = Number of neighbours of node v at time t

Example: degree profile of b



Our Approach

Introduction

Our Approach

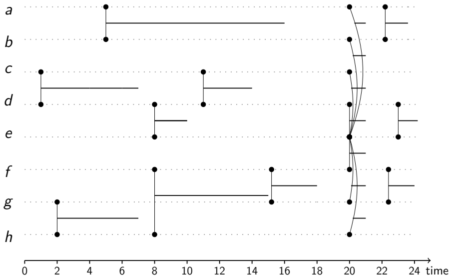
Our Method

Conclusion

① Detection : example

- Detection:

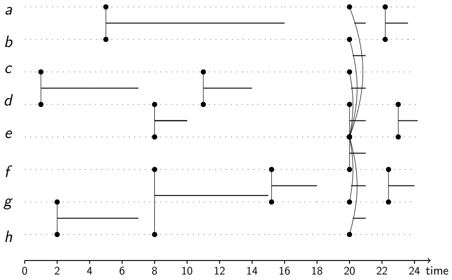
Find observations of the degree which deviate statistically from others.



① Detection : example

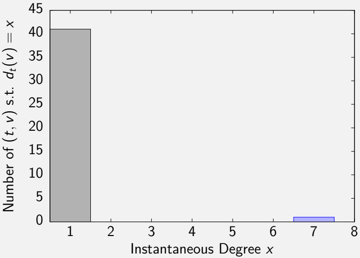
- Detection:

Find observations of the degree which deviate statistically from others.



Degree distribution

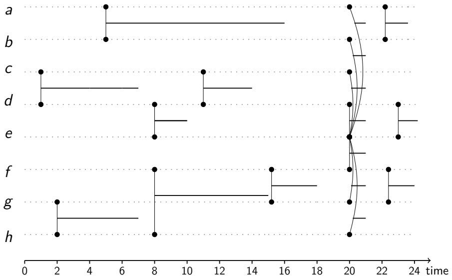
on all couples (v, t):



① Detection : example

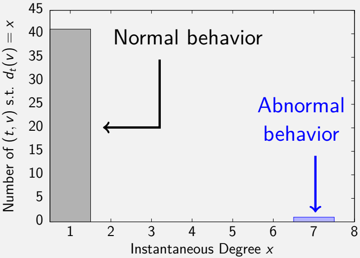
- Detection:

Find observations of the degree which deviate statistically from others.



Degree distribution on all couples (v, t):

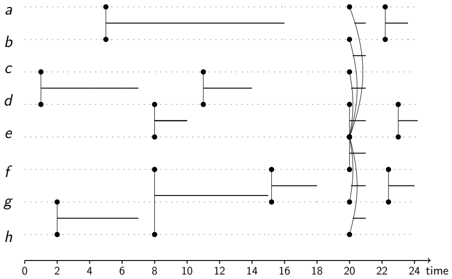
Detected outlier:
$\Rightarrow$ *outlying observation* $d_t(v) = 7$



② Identification : example

- Identification:

Find entities which are responsible for the outlying degree observation.



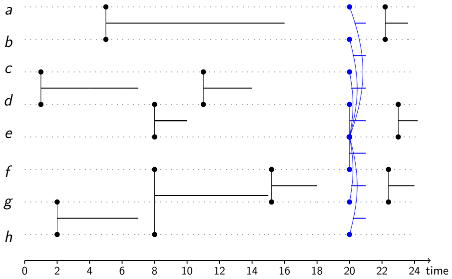
Detected outlier:
⇒ *outlying observation* $d_t(v) = 7$.

Identified outlier:

② Identification : example

- Identification:

Find entities which are responsible for the outlying degree observation.



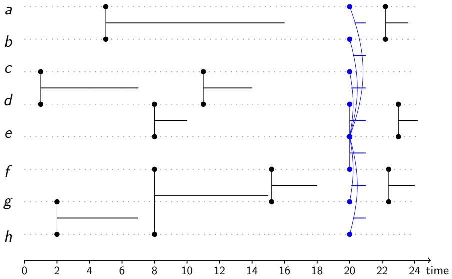
Detected outlier:
$\Rightarrow$ *outlying observation* $d_t(v) = 7$.

Identified outlier:
$\Rightarrow$ *the set :* $\{(e, t) \mid t \in [20, 21[\}$

③ Removal : example

- Removal:

Remove identified entities from the link stream.



Detected outlier:
⇒ *outlying observation* $d_t(v) = 7$.

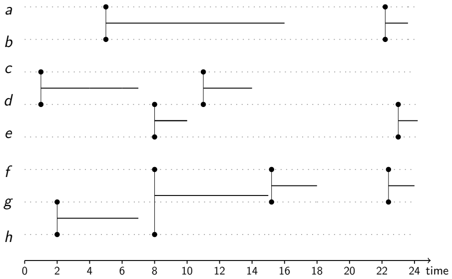
Identified outlier:
⇒ *the set :* $\{(e,t) \mid t \in [20, 21[\}$

Removed outlier:

③ Removal : example

- Removal:

Remove identified entities from the link stream.



Detected outlier:
$\Rightarrow$ *outlying observation* $d_t(v) = 7$.

Identified outlier:
$\Rightarrow$ *the set* : $\{(e,t) \mid t \in [20,21[\}$

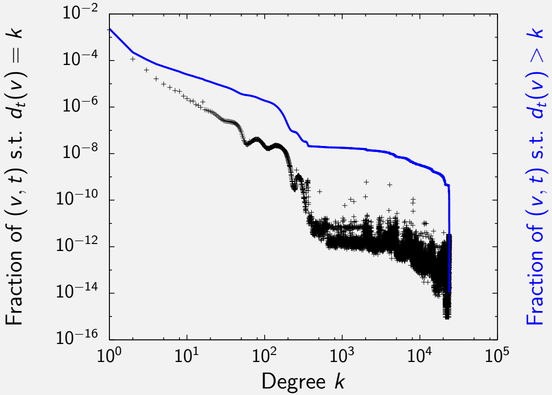
Removed outlier:
$\Rightarrow \cancel{\{(e,t) \mid t \in [20,21[\}}$

① Detection in our data

Link stream constructed from 1h of IP Traffic (MAWI)

↓

Degree distribution on all couples (v, t):

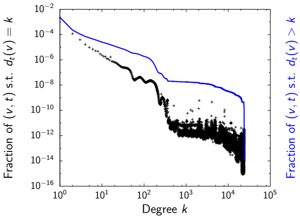


Difficulties

Outlier = Activity that deviates from the usual one

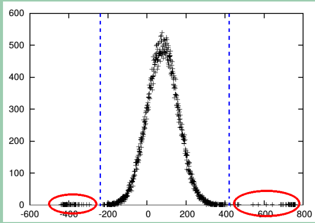
Find an outlier $\Longleftrightarrow$ Find the normality

✗



Heterogeneous

✓

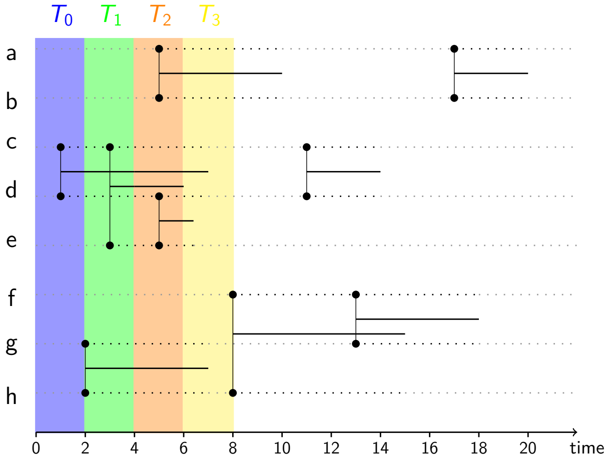


Homogeneous with outliers

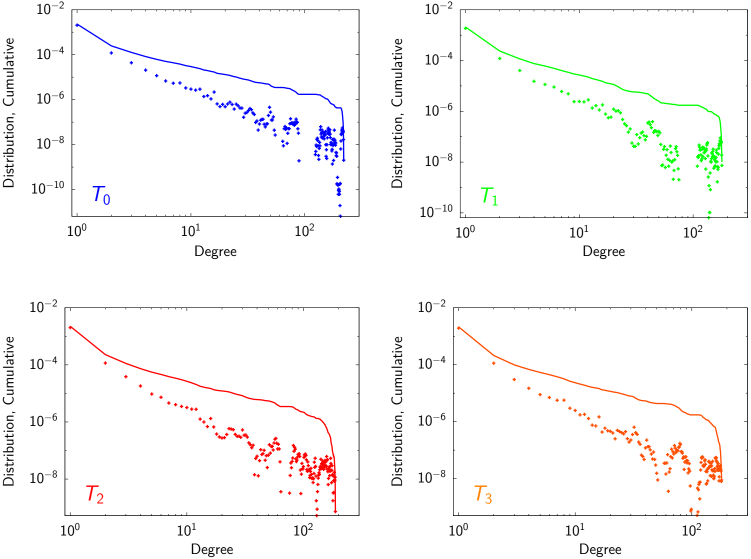
Our Method

Local Degree Distributions

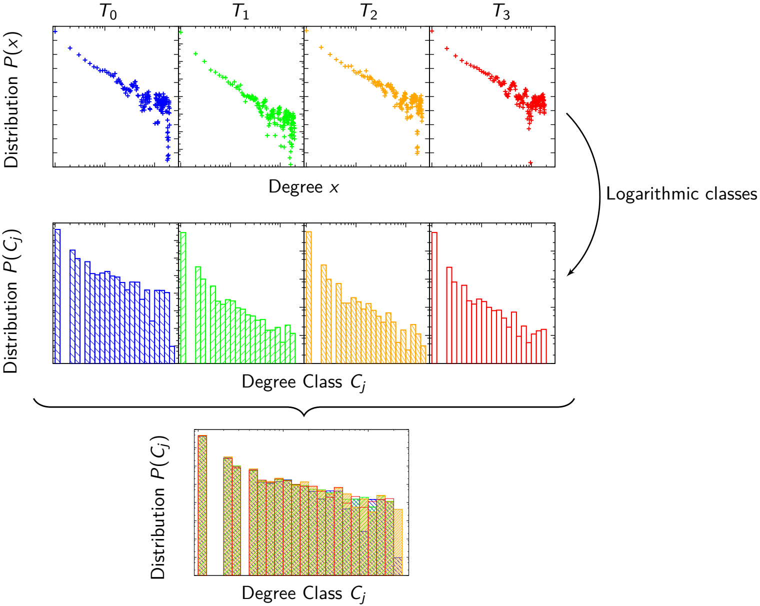
Degree observation on substreams with a duration of 2 seconds.



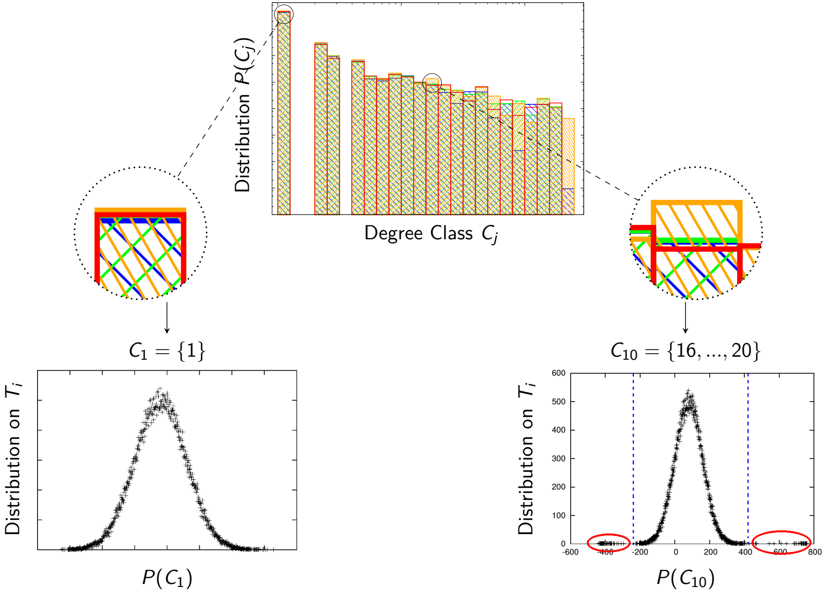
Local Distributions Similarity



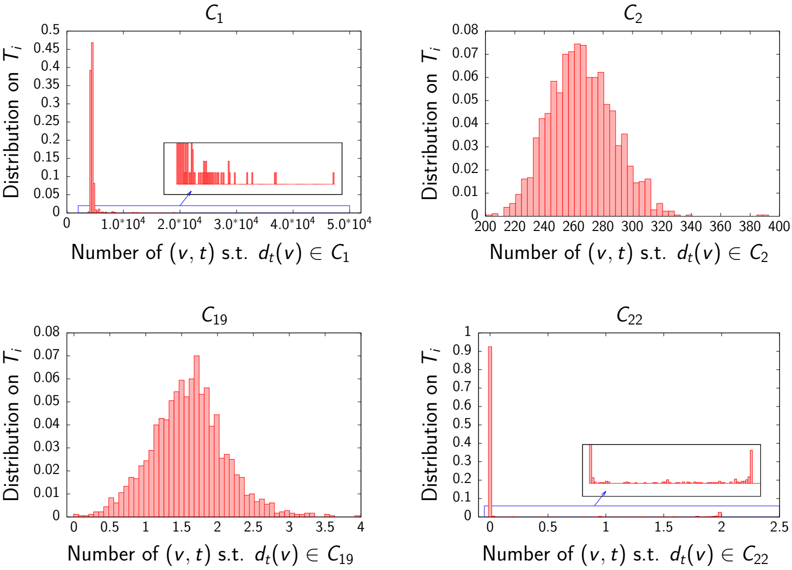
Comparison of Local Distributions



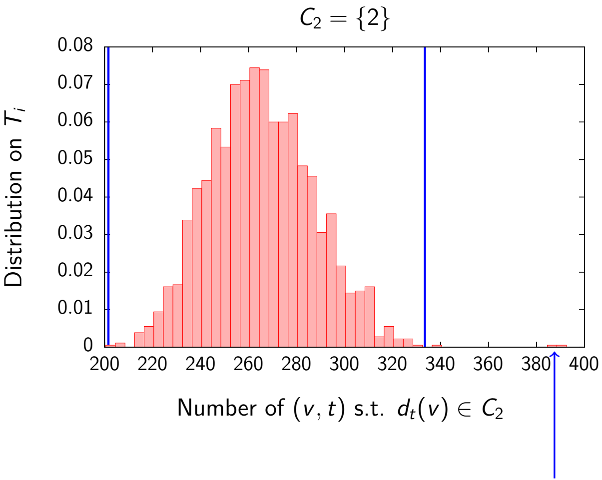
Comparison of Local Distributions



Results: Homogeneous Distributions



Results: Homogeneous Distributions



There are a lot more couples (v, t) for which $d_t(v) \in C_2$ during T_9 than during the majority of other time slices.

Overview of our method

Comparison of local distributions

↓

Detection of outliers

Identification: difficulties

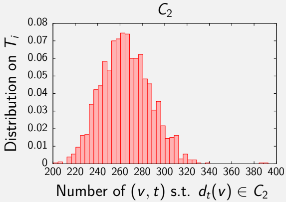
Detected Outlier $= 2$ informations
$\Rightarrow$ time slice T_i + degree class C_j

How to find responsible entities ?

How to identify detected outliers ?

Previous example:

Detected Outlier $\Rightarrow$ T_9 and C_2



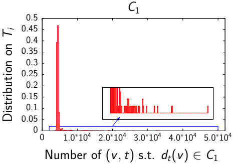
Difficulty: there are numerous (v, t) within C_2 during T_9

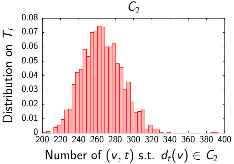
Which of them are abnormal ?

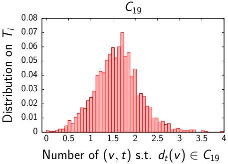
Identification

- Low degree classes:

outlier = normal + abnormal traffic

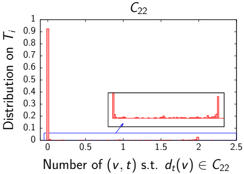


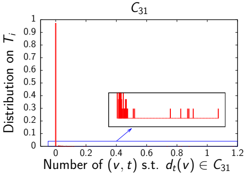


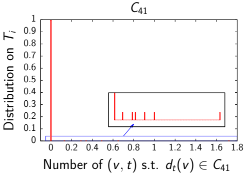


- High degree classes:

outlier = abnormal traffic only



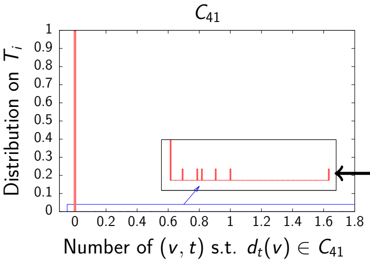




→ Direct identification possible in high degree classes only

Introduction
Context and Goals
Link Stream
Degree
Our Approach
In theory
In Practice
Difficulties
Our Method
Distributions Similarity
Detection
Identification
Removal
Validation
Conclusion

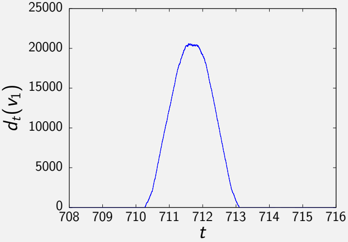
Identification in High Degree Classes



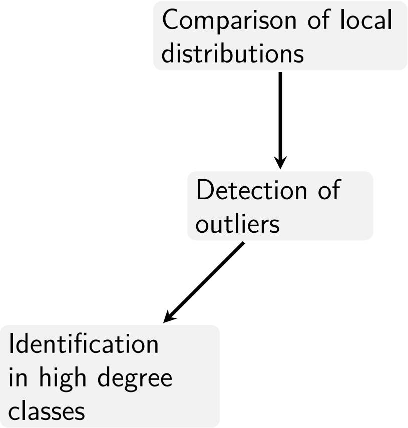
Detected Outlier:
$\Rightarrow \; T_{335} = [710, 712[$ and C_{41}

Identified outlier:
$\{(v_1, t) \mid t \in [710.3, 713.1[\}$

v_1 is the only node having a degree within C_{41} during T_{335}

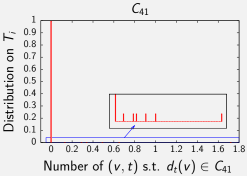


Overview of our method

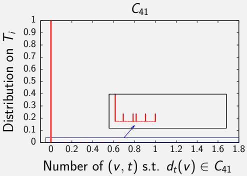


Removals of identified outliers

Disappearance of the detected outlier in the C_{41} distribution:

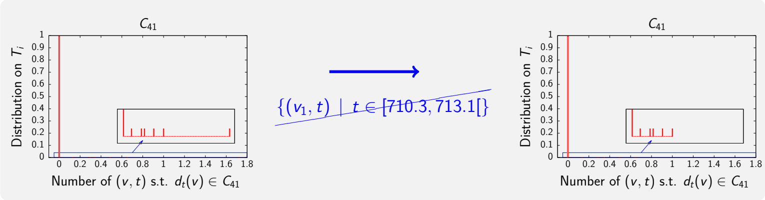


$\cancel{\{(v_1, t) \mid t \in [710.3, 713.1[\}}$

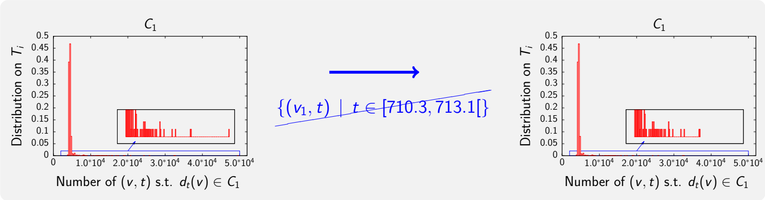


Removals of identified outliers

Disappearance of the detected outlier in the C_{41} distribution:

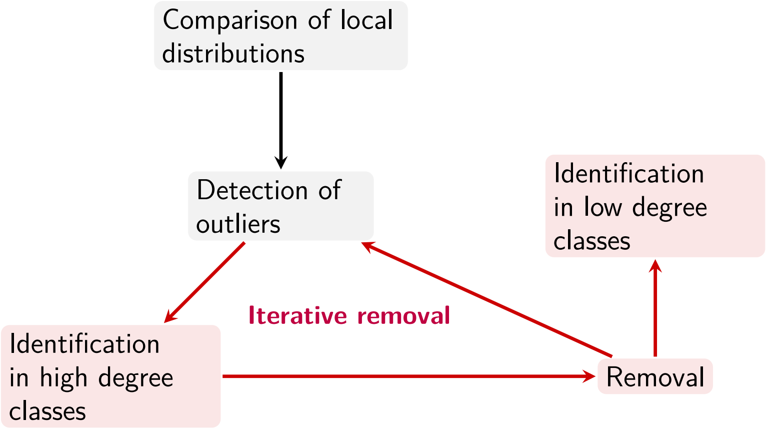


... as well as in a smaller degree class distribution:

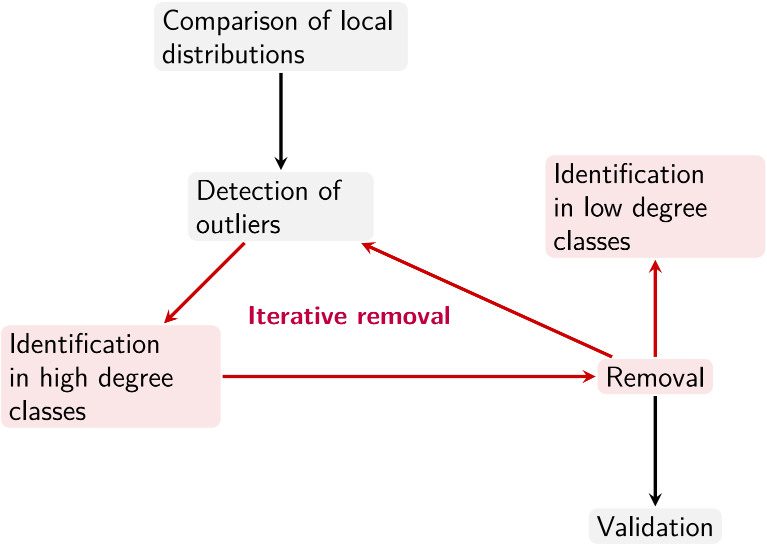


$\Rightarrow$ Allows to identify low degree classes outliers among neighbours of the removed node.

Overview of our method

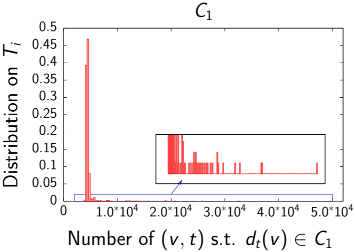


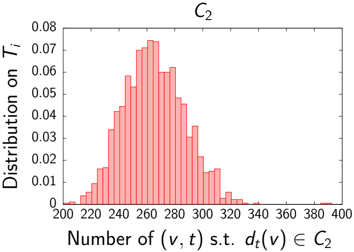
Overview of our method

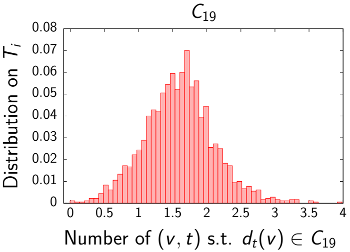


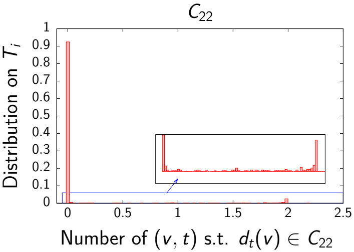
Degree Classes Distributions After Removals

Disappearance of most outliers without creation of negative outliers.



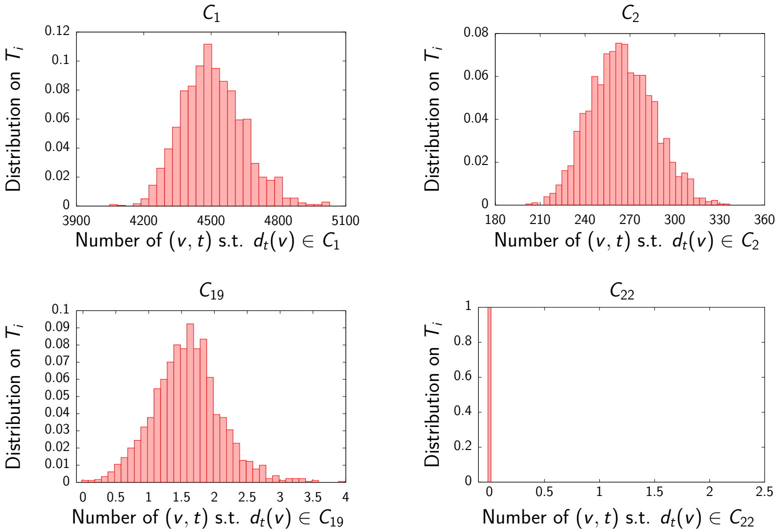






Degree Classes Distributions After Removals

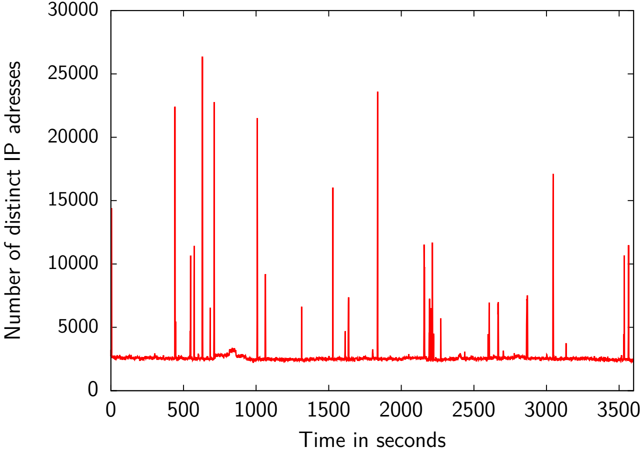
Disappearance of most outliers without creation of negative outliers.



Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163 =$ 85% of the detected outliers

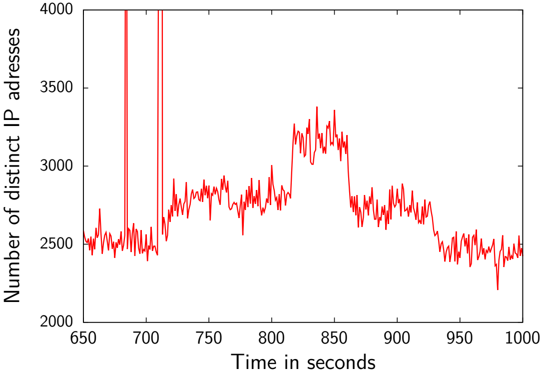
$\Rightarrow$ Consequence on the number of distinct IP addresses per second.

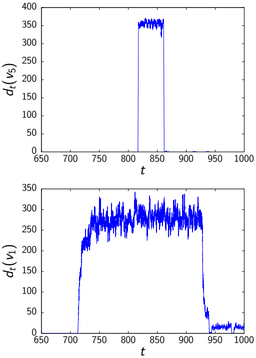


Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163$ = 85% of the detected outliers

$\Rightarrow$ Consequence on the number of distinct IP addresses per second.

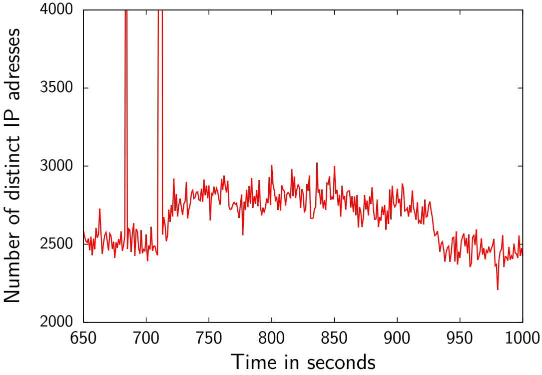


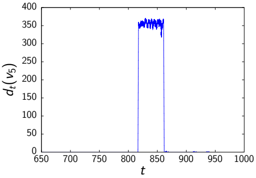


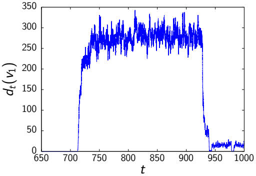
Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163 =$ 85% of the detected outliers

$\Rightarrow$ Consequence on the number of distinct IP addresses per second.



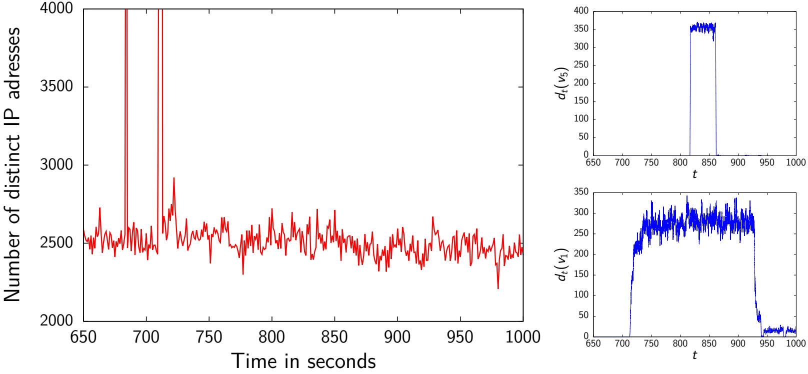




Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163 =$ 85% of the detected outliers

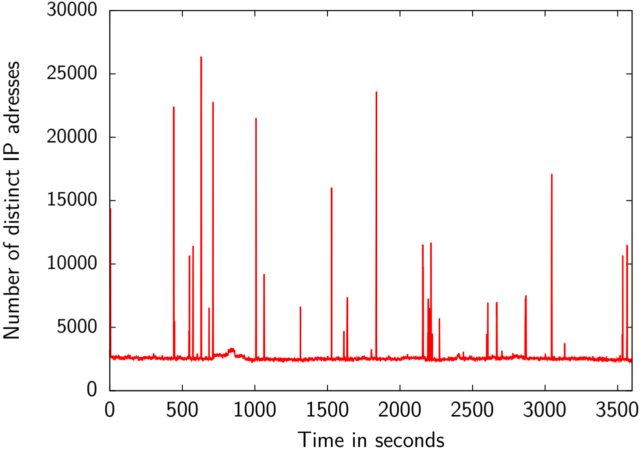
$\Rightarrow$ Consequence on the number of distinct IP addresses per second.



Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163 =$ 85% of the detected outliers

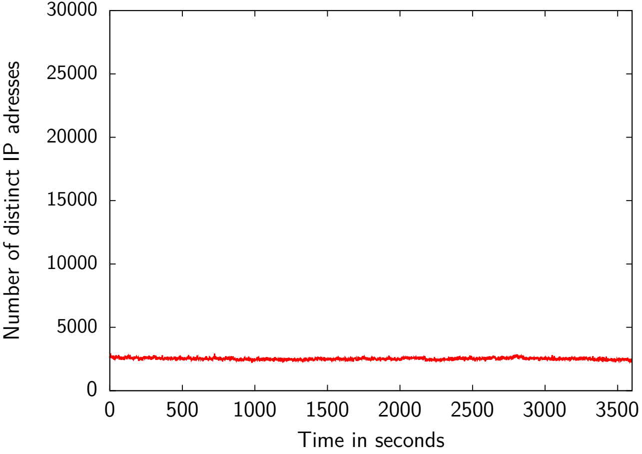
$\Rightarrow$ Consequence on the number of distinct IP addresses per second.



Creation of normal traffic

Number of detected outliers: $1,358$
Number of identified outliers: $1,163 =$ 85% of the detected outliers

$\Rightarrow$ Consequence on the number of distinct IP addresses per second.



Conclusion

Conclusion

Design of a method to detect and precisely identify outliers in heterogeneous distributions:

– Structural and temporal similarity evaluation of distributions.

– Modelling of IP traffic as a link stream.

$\rightarrow$ IP with anomalous degree profile, network scans.

Conclusion

Design of a method to detect and precisely identify outliers in heterogeneous distributions:

– Structural and temporal similarity evaluation of distributions.

– Modelling of IP traffic as a link stream.

$\rightarrow$ IP with anomalous degree profile, network scans.

Iterative removal of identified outliers

$\rightarrow$ Validation: Creation of normal traffic (w.r.t $d_t(v)$).

Conclusion

Design of a method to detect and precisely identify outliers in heterogeneous distributions:

– Structural and temporal similarity evaluation of distributions.

– Modelling of IP traffic as a link stream.

$\rightarrow$ IP with anomalous degree profile, network scans.

Iterative removal of identified outliers

$\rightarrow$ Validation: Creation of normal traffic (w.r.t $d_t(v)$).

$\Rightarrow$ Method applicable over temporal interactions in general.

Conclusion

Design of a method to detect and precisely identify outliers in heterogeneous distributions:

– Structural and temporal similarity evaluation of distributions.

– Modelling of IP traffic as a link stream.

$\rightarrow$ IP with anomalous degree profile, network scans.

Iterative removal of identified outliers

$\rightarrow$ Validation: Creation of normal traffic (w.r.t $d_t(v)$).

$\Rightarrow$ Method applicable over temporal interactions in general.

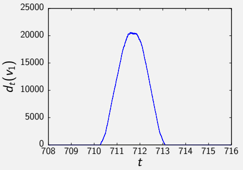
Thank you for your attention !

Identification: details

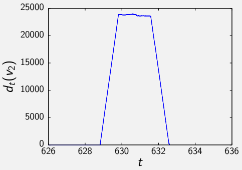
Detected Outlier $=$ time slice T_i $+$ degree class C_j

$$= \quad \{(v,t) : v \in C_{41} \text{ and } t \in T_{504} = [1008, 1010[\}$$

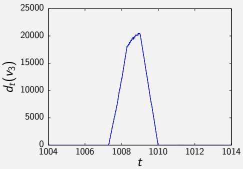
Nodes $\in C_{41}$:



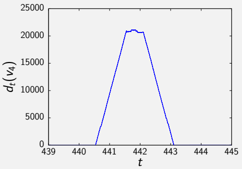
Abnormal on $I_{v_1} = [710.3, 713.1[$



Abnormal on $I_{v_2} = [628.8, 632.6[$



Abnormal on $I_{v_3} = [1007.3, 1010[$



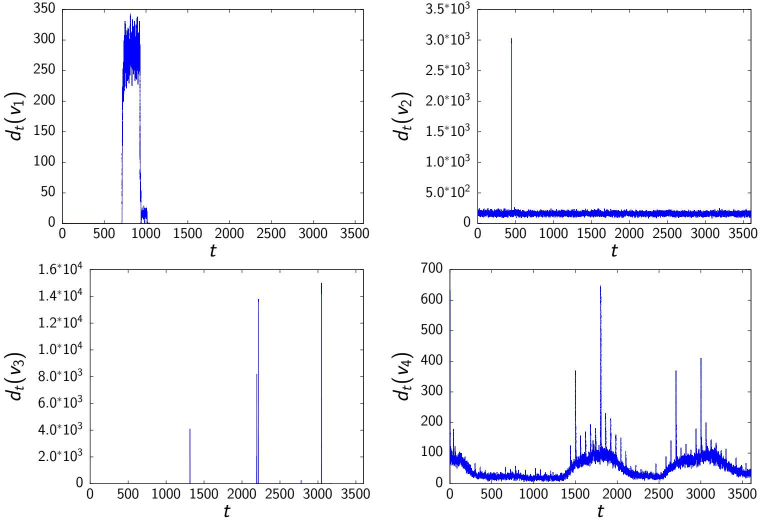
Abnormal on $I_{v_4} = [440.5, 443.2[$

$$I_{v_1} \cap T_{504} = \emptyset$$
$$I_{v_2} \cap T_{504} = \emptyset$$
$$I_{v_3} \cap T_{504} \neq \emptyset$$
$$I_{v_4} \cap T_{504} = \emptyset$$

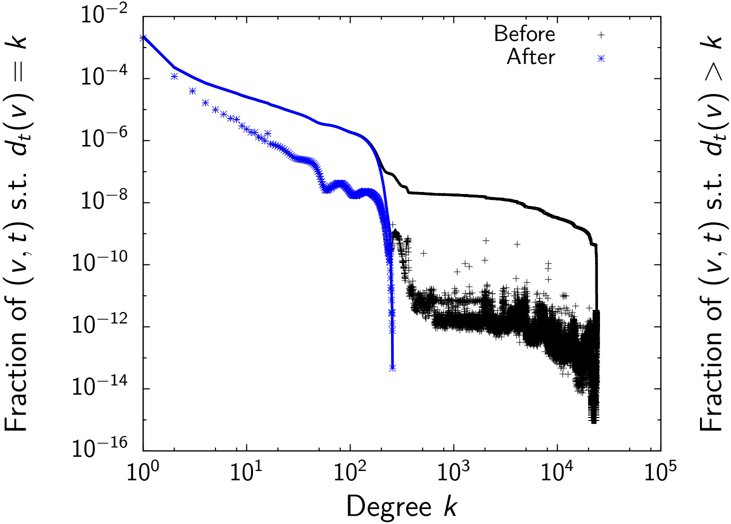
Identified outlier:

$\{(v_3, t) : t \in I_{v_3}\}$

Degree Profiles of 4 identified nodes



Degree Distribution : before and after



Classes construction

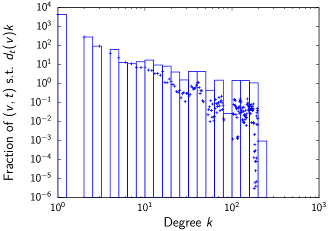
Need to respect the heterogeneous nature of the distribution:
- have low degree couples (v, t) which contains most of the traffic in isolated classes,
- take into account that the degree of nodes along time fluctuates and that generally: the larger the degree the larger the fluctuations.

$\Rightarrow$ logarithmic degree classes

In logarithmic scale: points spaced of the same distance represent values in the same ratio r

lin: $k_j \longrightarrow k_{j+1} = k_j + r$

log: $k_j \longrightarrow k_{j+1} = k_j \times r$ and $log(k_{j+1}) = log(k_j) + log(r)$



In our method:
$log(k_{j+1}) = log(k_j) + 0.1$

Other construction:
$\{1\}, \{2\}, ...\{9\}, \{10, ..., 19\}, \{20, ..., 29\}, ..., \{90, ..., 99\}, \{100, ...199\}$, etc.