Stream-wise Parallel Anomaly Detection in Computer Networks

Tomáš Čejka

Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague

October 19, 2018

Introduction

Topics of interest in this thesis

- Network **monitoring** and traffic **analysis**
- **Application layer (L7)** protocols and detection
- **Semantic relations** in flow data and **parallel processing**

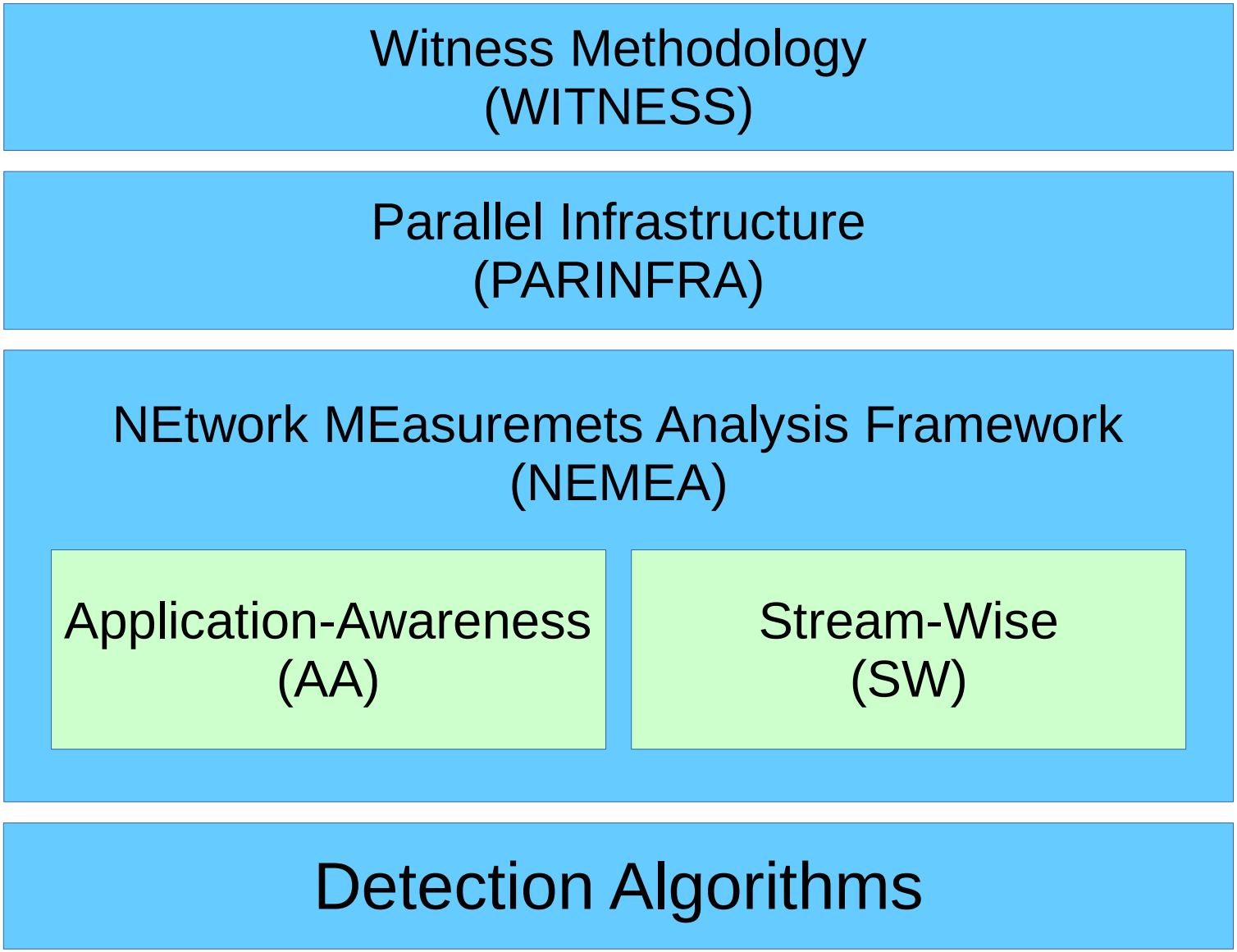
Motivation

- **Growing volume** of data in networks
- **Delay** of traditional batch flow data processing
- **Scalability** of the analysis and detection system
- **Parallel processing** without increased communication and without affecting detection results

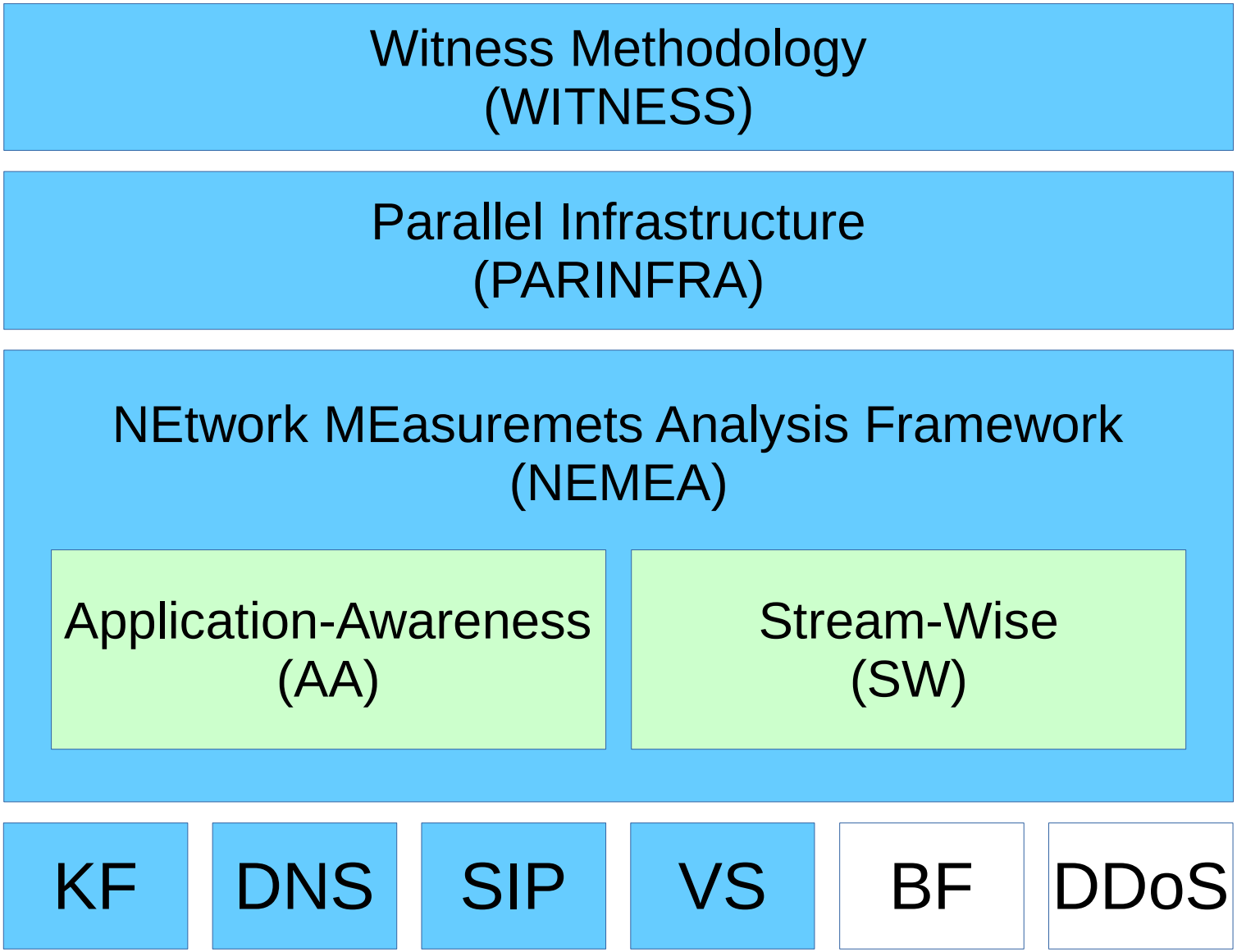
Note: "flow data" means a sequence of "flow records"

Note #2: (IP) Flow record is an aggregated information about all packets within a time interval having the same features (such as *srcip*, *dstip*, *srcport*, *dstport*, *proto*).

Parts of the Thesis



Parts of the Thesis



Scopes of My Contributions

1. **Stream-wise** flow data processing
2. **Application aware** detection algorithms
3. **Formal symbolic description** of detection algorithms
4. **Parallel flow data processing** based on **semantic relations**
5. **Experiments** and deployment in real networks

Stream-wise Processing

Current state-of-the-art

- Batch analysis of flow data divided into fixed time windows
- Delayed processing

Improvement of my approach

- Processing immediately "on-the-fly" (continuously) without delay
- No (long-term) flow data storage
- Aggregation per context, e.g., per IP or IP pair

$$S_t = A(F_t, S_{t-1}), \quad S_{\leq 0} = s_0.$$

T. Čejka, Z. Rosa, H. Kubátová: *Stream-wise Detection of Surreptitious Traffic over DNS*. IEEE CAMAD, Greece, 2014.
T. Čejka, V. Bartoš, M. Švepeš, Z. Rosa, H. Kubátová: *NEMEA: A Framework for Network Traffic Analysis*. CNSM, Canada, 2016.

Application Awareness

Current state-of-the-art

- Flow-based analysis uses up to transport layer (L4) info

Improvement of my approach

- Extend flow records with L7 information
- Developed new detection algorithms for several L7 threats
- Evaluation of the feasibility in real network

Note: Naturally, this feature can work only with unencrypted traffic.

T. Čejka, V. Bartoš, L. Truxa, and H. Kubátová: *Using Application-Aware Flow Monitoring for SIP Fraud Detection*. AIMS, Belgium, 2015.

Network Measurements Analysis (NEMEA) framework

- Open source, stream-wise, application-aware, modular, flow-based
- Evaluation of detection algorithms
- Deployment in:
  - CESNET2 — Czech national research and education network (NREN)
  - SWITCH (NREN in Switzerland)
  - Linköping university (Sweden)
  - commercial data center/ISP in Prague
- Significant source of detected security events in CESNET2
- Foreign contributors at github — established community

T. Čejka, V. Bartoš, M. Švepeš, Z. Rosa, H. Kubátová: *NEMEA: A Framework for Network Traffic Analysis*. CNSM, Canada, 2016.

Developed Algorithms and Experiments

Set of developed detection modules

- Suspicious VoIP traffic (SIP)

  T. Čejka, V. Bartoš, L. Truxa, and H. Kubátová: *Using Application-Aware Flow Monitoring for SIP Fraud Detection*. AIMS, Belgium, 2015.

  T. Jánský, T. Čejka, V. Bartoš *Hunting SIP Authentication Attacks Efficiently*. AIMS, Switzerland, 2017.

- Port Scans (VS)

  T. Čejka, M. Švepeš *Analysis of Vertical Scans Discovered by Naive Detection*. AIMS, Germany, 2016.

- Real-time filtering based on known features (KF)

  T. Čejka, R. Bodó, and H. Kubátová: *Nemea: Searching for Botnet Footprints*. PESW, Czech Republic, 2015.

- Covert Channels — DNS tunnels (DNS)

  T. Čejka, Z. Rosa, H. Kubátová: *Stream-wise Detection of Surreptitious Traffic over DNS*, IEEE CAMAD, Greece, 2014.

+ about 25 supervised bachelor/master theses related to NEMEA

Formal symbolic description

Current state-of-the-art

- Missing standard approach of readable description of algorithms
- Algorithms described in the literature usually use free-form text description or some kind of pseudocode

Improvement of my approach

- Description of algorithms using own defined symbols and operators
- Description is based on specification of relevant subsets of flow data analyzed by algorithm
- Analysis of the formally described algorithms allowed identification of semantic relations

Semantic Relations

- Random splitting flow data breaks semantic relations
- Thesis defines *witness*: data subset that must remain together
- Preserving *witnesses* — without affecting detection results

T. Čejka, M. Žádník: *Preserving relations in parallel flow data processing*. AIMS, Switzerland, 2017.

Example: Detection of DDoS Attacks (1/2)

Sequence of flow records

$F = \{F_0, ..., F_n\}$

Symbol for grouping flow records

$I \underset{dstip}{\subset} F$ creates groups $I_0, ..., I_m$, where each group contains all flow records with the same *dstip*.

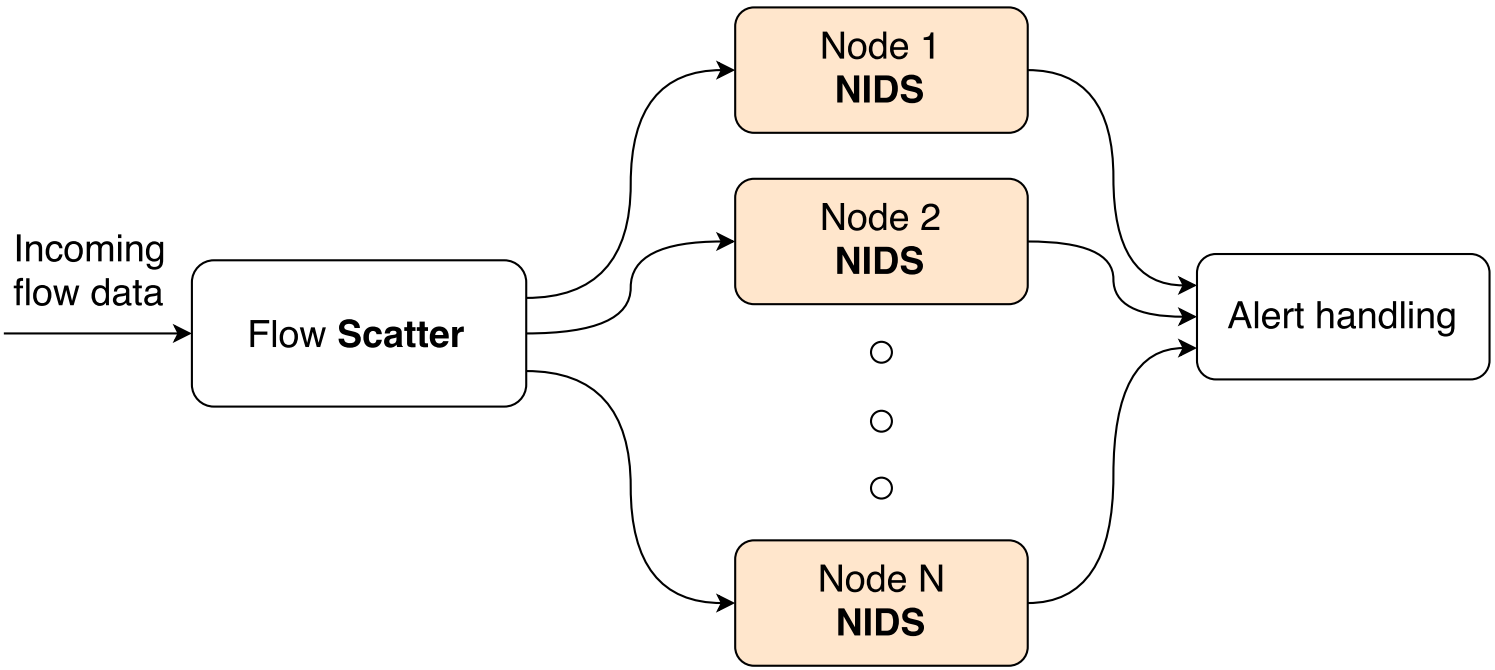
Example: Detection of DDoS Attacks (2/2)

Simplified Algorithm Description

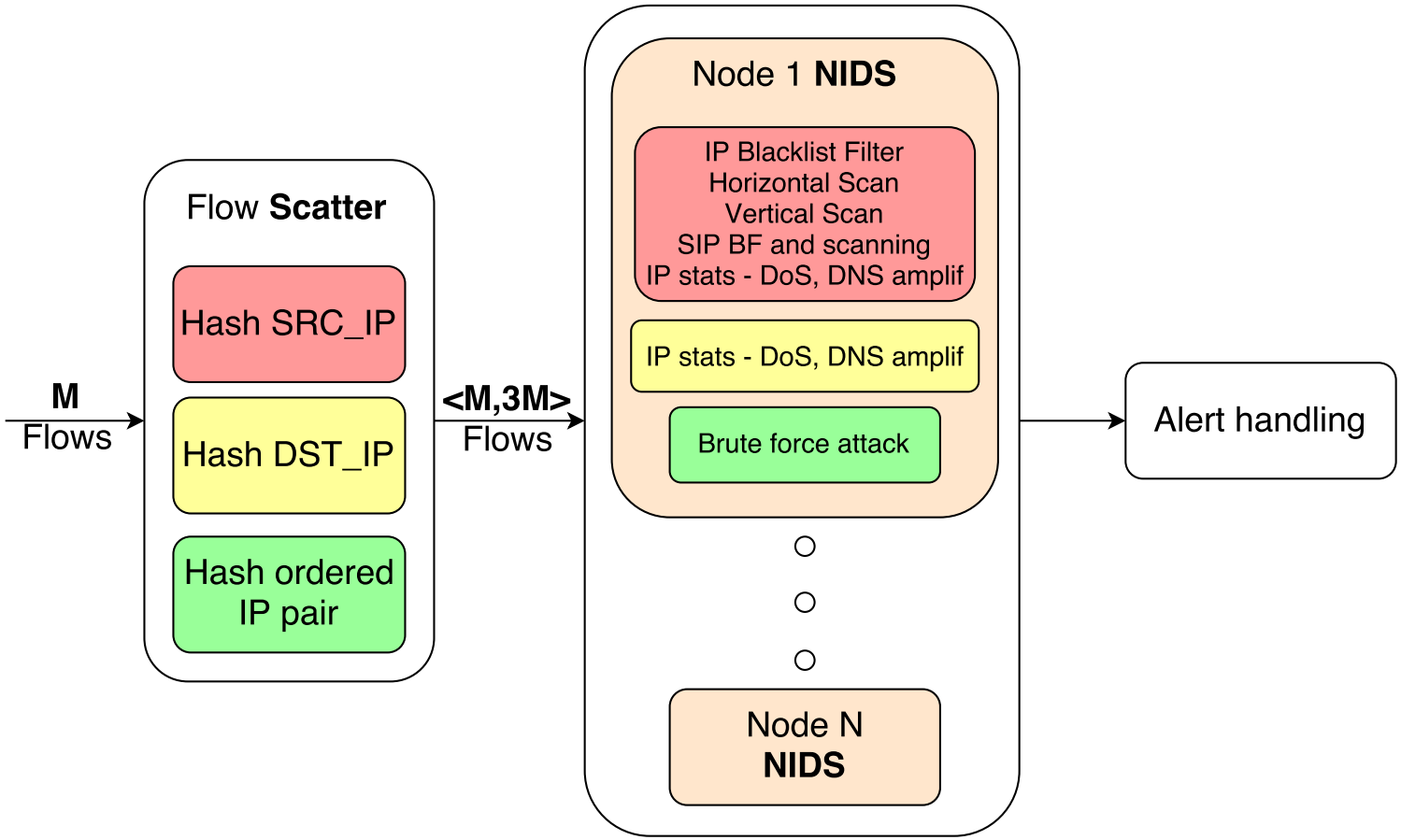
for all $I \underset{dstip}{\subset} F$ **do**
  $S_1 = \text{COUNT}(\text{DISTINCT}(I[srcip])$
  $S_2 = \text{COUNT}(I[bytes])$
  if $S_1 \geq Thr_1$ and $S_2 \geq Thr_2$ **then**
    $\text{Alert}_{\text{ddos}}$
  end if
end for

Parallel analysis of flow data

- Flow Scatter and a set of IDS nodes

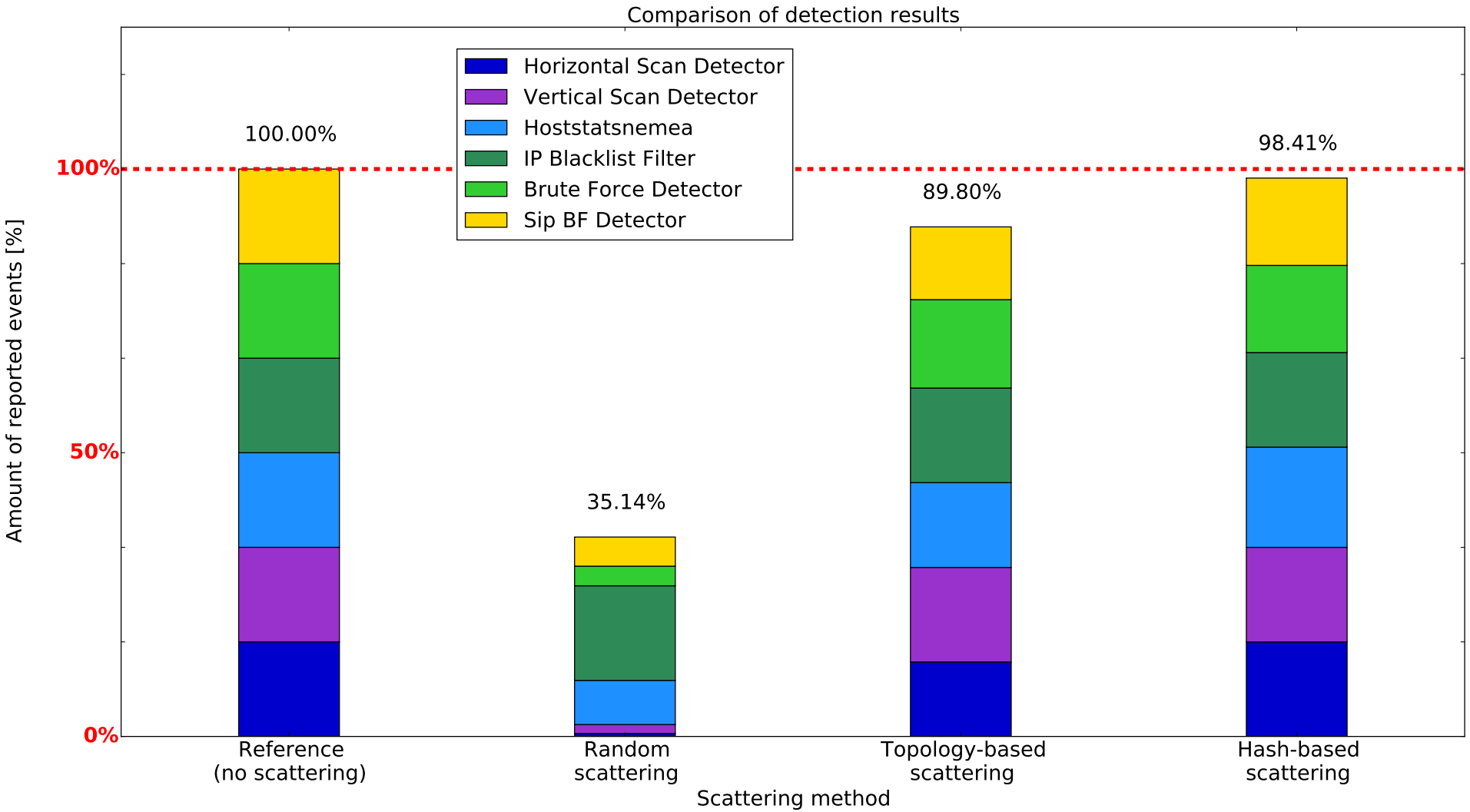


Experiments with Parallel Processing



M. Švepeš, T. Čejka: *Overload-resistant Network Traffic Analysis*. PESW, Czech Republic, 2016.
M. Švepeš, T. Čejka: *Making flow-based security detection parallel*. AIMS, Switzerland, 2017.

Experiments with Parallel Processing



Summary and Main Contributions

- My research was focused on real-time analysis of flow data
- Developed open source NEMEA framework and a set of detection algorithms
- Defined stream-wise approach that is used in NEMEA
- Use of application layer (L7) information for more reliable detection
- Described semantic relations in flow data— witnesses
- Parallel processing based on splitting a stream of data with respect to witnesses

My Selected Reviewed Publications (1/3)

[P.1] P. Benáček, V. Puš, H. Kubátová, T. Čejka: *P4-To-VHDL: Automatic generation of high-speed input and output network blocks*. Microprocessors and Microsystems, Vol. 56, Elsevier, 2018.

The paper has been cited in:

- Garcia, Luis Fernando Uria, et al.: *Introdução à Linguagem P4-Teoria e Prática.* Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (Minicursos_SBRC) 36 (2018).
- H. Zolfaghari, et al.: *An Explicitly Parallel Architecture for Packet Parsing in Software Defined Networks*. IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), 2018.

[P.2] J. Havránek, P. Velan, T. Čejka, P. Benáček: *Enhanced Flow Monitoring with P4 Generated Flexible Packet Parser*. AIMS, Germany, 2018.

[P.3] T. Jánský, T. Čejka, M. Žádník, V. Bartoš: *Augmented DDoS Mitigation with Reputation Scores*. 13th International Conference on Availability, Reliability and Security, ARES, 2018.

[P.4] T. Čejka, M. Žádník: *Preserving relations in parallel flow data processing*. IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS, Switzerland, 2017.

My Selected Reviewed Publications (2/3)

[P.5] T. Jánský, T. Čejka, V. Bartoš: *Hunting SIP Authentication Attacks Efficiently*. IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS, Switzerland, 2017.

[P.6] M. Švepeš, T. Čejka: *Making flow data analysis parallel*. IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS, Switzerland, 2017.

[P.7] T. Čejka, V. Bartoš, M. Švepeš, Z. Rosa, H. Kubátová: *NEMEA: A Framework for Network Traffic Analysis*. International Conference on Network and Service Management (CNSM), Canada, 2016.

[P.8] Z. Rosa, T. Čejka, M. Žádník, V. Puš: *Building a Feedback Loop to Capture Evidence of Network Incidents*. International Conference on Network and Service Management (CNSM), Canada, 2016.

[P.9] T. Čejka, M. Švepeš: *Analysis of Vertical Scans Discovered by Naive Detection*. IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS, Germany, 2016.

[P.10] T. Čejka, V. Bartoš, L. Truxa, and H. Kubátová: *Using Application-Aware Flow Monitoring for SIP Fraud Detection*. IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS, Belgium, 2015.

My Selected Reviewed Publications (3/3)

[P.11] T. Čejka, Z. Rosa, H. Kubátová: *Stream-wise Detection of Surreptitious Traffic over DNS*. IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (IEEE CAMAD), Greece, 2014.

The paper has been cited in:

- Nuojua, V., et al.: *DNS tunneling detection techniques – Classification, and theoretical comparison in case of a real APT campaign.* Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2017.

[P.12] P. Benáček, R. B. Blažek, T. Čejka, H. Kubátová: *Change-Point Detection Method on 100 Gb/s Ethernet Interface*. ACM/IEEE Symposium on Architectures for Networking and Communications Systems, USA, 2014.

The paper has been cited in:

- Nakamura, K., et al.: *An FPGA-based low-latency network processing for spark streaming.* IEEE International Conference on Big Data, 2016.

[P.13] T. Čejka, L. Kekely, P. Benáček, R. B. Blažek, H. Kubátová: *FPGA Accelerated Change-Point Detection Method for 100Gb/s Networks*. Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS), Czech Republic, 2014.

Thank you for your attention!

Tomáš Čejka

cejkato2@fit.cvut.cz

Acknowledgement: This research was supported by the CTU grant No. SGS17/212/OHK3/3T/18 funded by the Ministry of Education, Youth and Sports of the Czech Republic and by several projects of CESNET, research organization and operator of the Czech national research and education network.

Remark 1

Equation (3.3) case for $i = 0$ is not necessary as the second case also includes the situation in which $F_0[attr] \notin \{\}$.

Comment

$$\text{DISTINCT}(F_i[attr]) = \begin{cases} 1 & \text{for } i = 0, \\ 1 & \text{for } F_i[attr] \notin \{F_0[attr], \ldots, F_{i-1}[attr]\}, \\ 0 & \text{otherwise.} \end{cases}$$

The intention was to explicitly mention the case $i = 0$ for easier readability.

Remark 2

Equation 3.6 does not probably express the author's intention. Reading the text I assume that I is an equivalence class on F. The provided definition rather express that I is any strict subset (possibly empty) of F such that its members have the same attribute *srcip*.

Comment

$$F = \{F_0, \ldots, F_n\},$$
$$I = \{I_0, \ldots, I_m\},$$
$$I \underset{srcip}{\subset} F : I \subset F, \quad \forall i, j : I_i[srcip] = I_j[srcip].$$

Intention:
For any non-empty F, the operator $\underset{attr}{\subset}$ should create a set of groups I of all suitable flow records with fulfilled characteristic $attr$.

Remark 3

In algorithms 3.2 – 3.5 there is a mismatch in a notation of flow subsets. For instance in 3.2 there should be $S_1 = COUNT(DISTINCT\_SUBDOMAIN(I[domain])$. Similarly in other algorithms.

Comment

I would like to apologize for the mismatches.
Recapitulation:
Alg. 3.2, Alg. 3.3, Alg. 3.5 suffer from the bad notation, as Reviewer stated.
Alg. 3.1, Alg. 3.4, Alg. 3.6, Alg. 3.7, Alg. 3.8, Alg. 3.9 are correct.

Remark 4

Equation 3.8 suffers from the same problem as Equation 3.6.

Remark 5

Algorithm 3.8: CHECKBRUTEFORCE function is not defined anywhere in the text.

Comment

The aim of the description was to show which related flow records are analyzed. Definition of CHECKBRUTEFORCE was not important since it does not affect the analyzed sequence of flow records.

Questions (1/5)

Have you evaluated these algorithms in terms of their accuracy?

Partially yes, evaluation of the algorithms was part of the development.
However, this kind of evaluation is complicated:

- lack of ground truth,
- generally, trained parameters work only locally,
- real traffic is not annotated.

Is it possible to include "history" in the detection methods to improve accuracy?

It was not considered yet but it is possible.
More preferred way in practice is some alert post-processing to decrease number of false positive alerts. This can be based on history.

Questions (2/5)

The method has an almost uniform distribution of flows. Each flow can be sent to up n-nodes, where n is the number of hash functions. What is the cost of flow distribution comparing to the cost of execution of detection algorithms?

Detection algorithms (modules) are significantly slower than the flow scatter, because detection modules are much more complex.

By construction, this method obeys "semantic" relations between flows. However, results presented in paper A.7 show that it misses some alerts. Why?

- non-deterministic variances of time delay during data replay,
- periodic "cleaning"
  $\rightarrow$ some "border" events might not reach thresholds,
- difference was caused of by one detection module.

Questions (3/5)

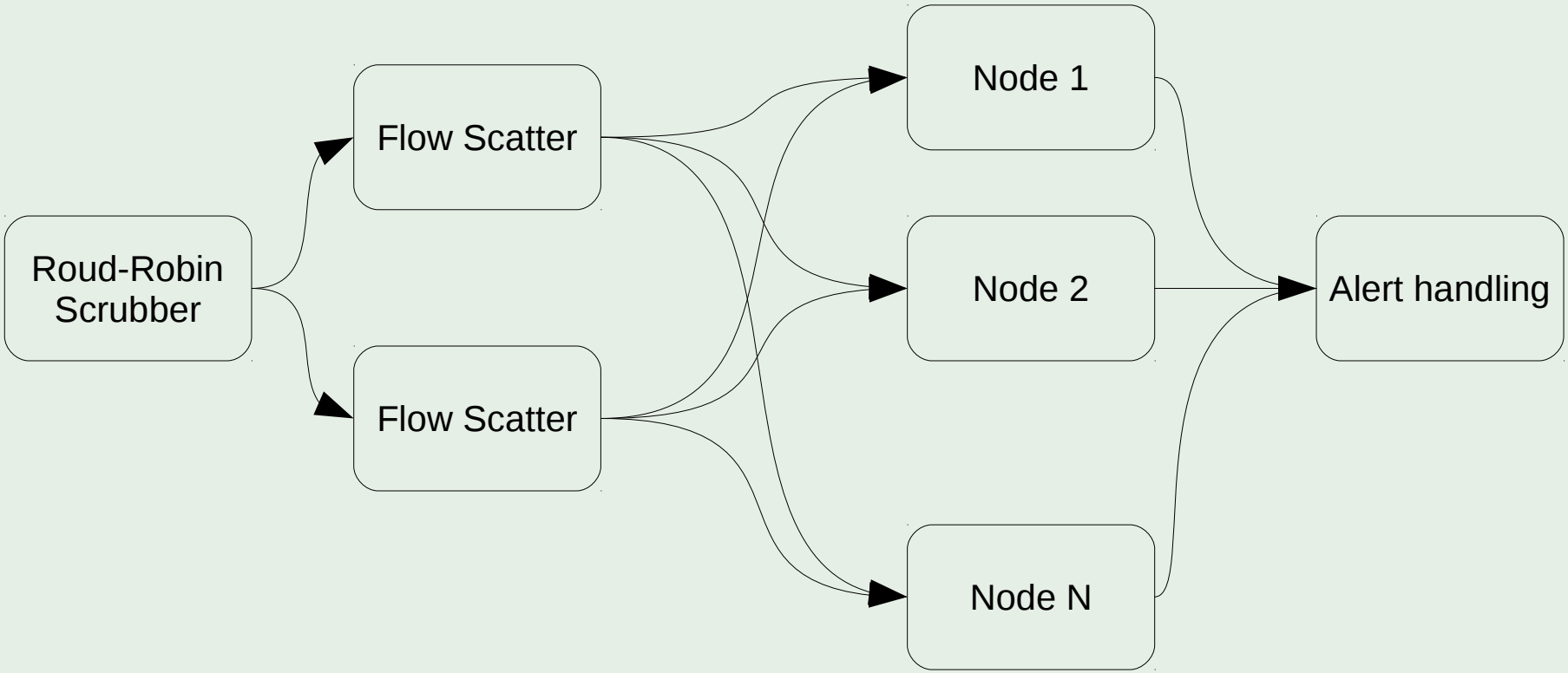
Have you measured some other performance indicators than the number of flows that the scatter can process?

No, because the throughput (number of flow records per second) is the most significant performance metric for deployment in practice.

What is possible speed up when adding new processing nodes?

It depends mainly on the slowest detection module.

Multiple Flow Scatters



Questions (5/5)

What are the benefits of UniRec in comparison to other serialisation formats?

Simplicity and focus on flow data representation.
However, any other data format can be used.

What is the performance gain of using UniRec comparing to other binary formats, e.g., ProtocolBuffers, Apache Thrift?

Unfortunately, we do not have such comparison based on experiments.
According to documentation, ProtocolBuffers and Apache Thrift are primarily designed for RPC that means a description/schema must be known apriori.
Contrary, UniRec allows statically defined fields as well dynamically (at runtime) defined fields.

Question 1

In the Chapter 1.1, Motivation, you mentioned Distributed on-the-fly anomaly detection. Please indicate the difficulty of this type of detection and how it should be addressed?

Comment

The issue is to analyze a continuous stream of flow data using as many detection algorithms as possible. The difficulty is in high volume of flow data that we need to process fast enough so we do not loose anything.

This issue was elaborated in my thesis:

- stream-wise approach is feasible,
- splitting a stream of flow data with respect to witnesses is possible,
- and it allows for parallel processing.

Question 2

Extended flow records increase the accuracy and reliability of detection of the anomalies, but this tool works with unencrypted traffic only. Do you have any idea of how to detect anomalies in encrypted communication?

Comment

Encrypted traffic is an inevitable future for transit infrastructure. There are two ways how to deal with it:
1) Additional features in flow data (e.g., inter-packet delays, histogram of sizes) and some Machine Learning techniques.
2) Deploy monitoring probes closer to the services (e.g., in data centers), where the data are unencrypted.