Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement

Petr Velan, Tomáš Jirsík, Pavel Čeleda

{velan|jirsik|celeda}@ics.muni.cz

19th EUNICE Workshop on Advances in Communication Networking
28-30 August 2013, Chemnitz, Germany

Part I

Introduction

Motivation and R&D Goals – I

Is NetFlow still sufficient?

FTP	⟶	**20/21**
SSH	⟶	**22**
SMTP	⟶	**25**
HTTP	⟶	**80**
POP3	⟶	**110**
IMAP	⟶	**143**
HTTPS	⟶	**443**

Well-known Ports Applications

Motivation and R&D Goals – I

Is NetFlow still sufficient?

FTP	→	20/21
SSH	→	22
SMTP	→	25
HTTP	→	80
POP3	→	110
IMAP	→	143
HTTPS	→	443

Well-known Ports Applications

Today Applications

Motivation and R&D Goals – I

Is NetFlow still sufficient?

FTP	→	20/21
SSH	→	22
SMTP	→	25
HTTP	→	80
POP3	→	110
IMAP	→	143
HTTPS	→	443

Well-known Ports Applications

Today Applications

HTTP - *"new Transmission Control Protocol"* - new TCP

Motivation and R&D Goals – II

How to add application visibility to flow?

- Application labeling (protocol recognition)
- Application data (deep packet inspection)

Use the best DPI parsers to extend the flow

- Speed and accuracy is the most important factor
- We set out to find the best parser for HTTP protocol

Part II

HTTP Parser

General HTTP Parser Design

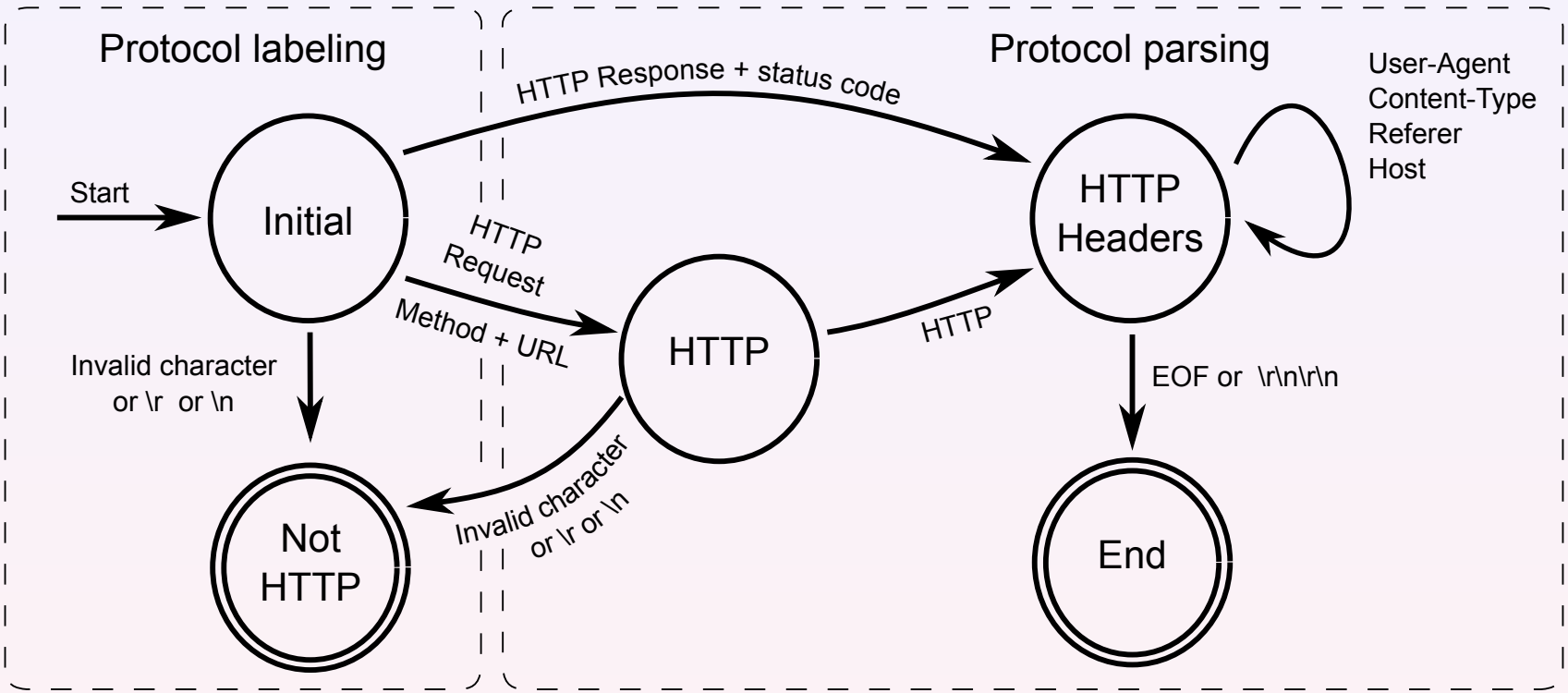
```
GET /wiki/Hypertext_Transfer_Protocol HTTP/1.1\r\n
Host: en.wikipedia.org\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:23.0) Gecko/20100101 Firefox/23.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: cs,en-us;q=0.7,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Referer: http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol\r\n
Connection: keep-alive\r\n
If-Modified-Since: Sat, 22 Jun 2013 17:32:12 GMT\r\n
Cach-Control: max-age=0\r\n
\r\n
```

- Find one of **HTTP**, **POST**, **GET**, **CONNECT**, **PUT**, **DELETE**, **HEAD**, **TRACE** method
- Parse **status code** or **URI**
- Try to find matching header fields for **User-Agent**, **Content-Type**, **Host**, **Referer**
- End when double end of line (’\r\n’) is encountered

Evaluated Parser Types

- **No application parser** - L2 through L4 flow exporters
  - **No HTTP** - no special parser, reference measurement
- **String compare** - nProbe, FlowMon
  - **strcmp** - hand-written parser standard version
  - **optimized strcmp** - highly optimized hand-written parser
- **Regular expression** - YAF
  - **pcre** - parser using Perl Compatible Regular Expressions
- **Finite automaton** - our approach
  - **flex** - parser using flex generated finite automaton
  - **optimized flex** - optimization of flex parser

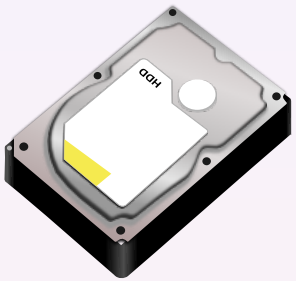
Flex Parser Schema



Part III

Experiment

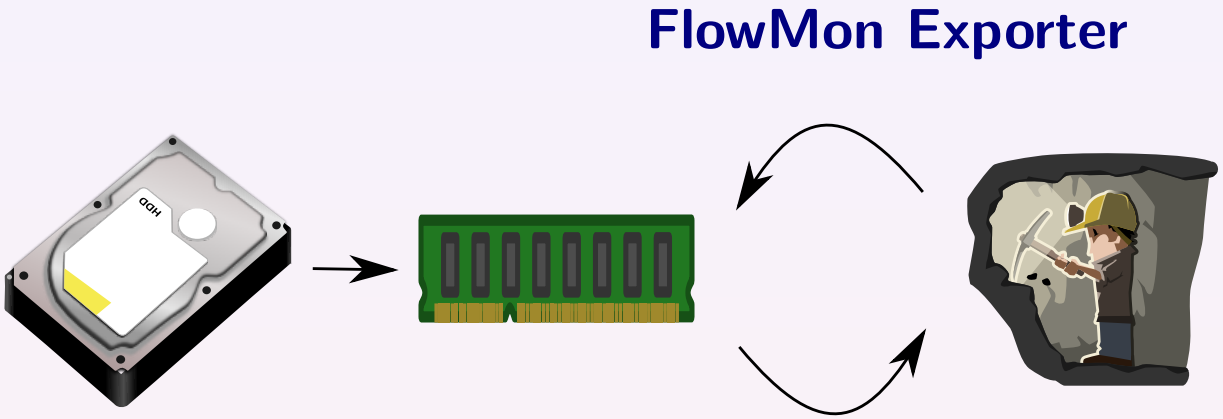
Measurement Setup



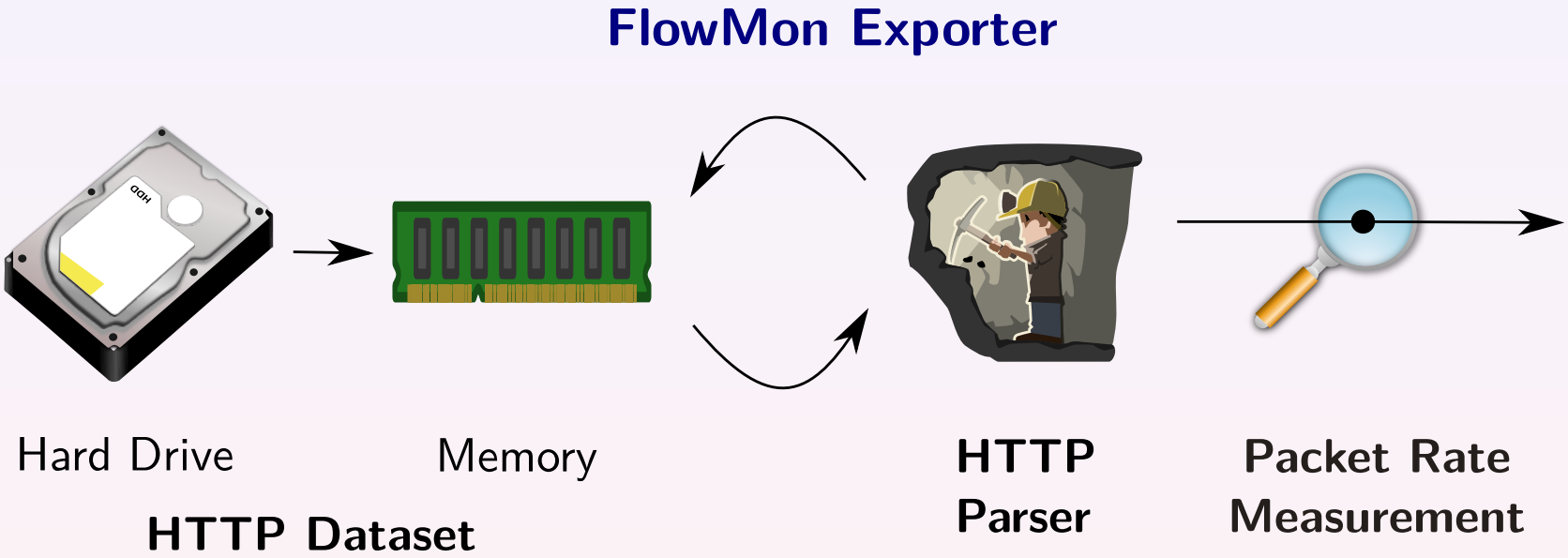
Hard Drive

HTTP Dataset

Measurement Setup



Measurement Setup



Measurement Setup II

Dataset

- HTTP request and response packets
- Data packets with binary payload
- Created data sets containing 0 - 100 % of HTTP packets
- Modified data packets with End of Line only at start and end

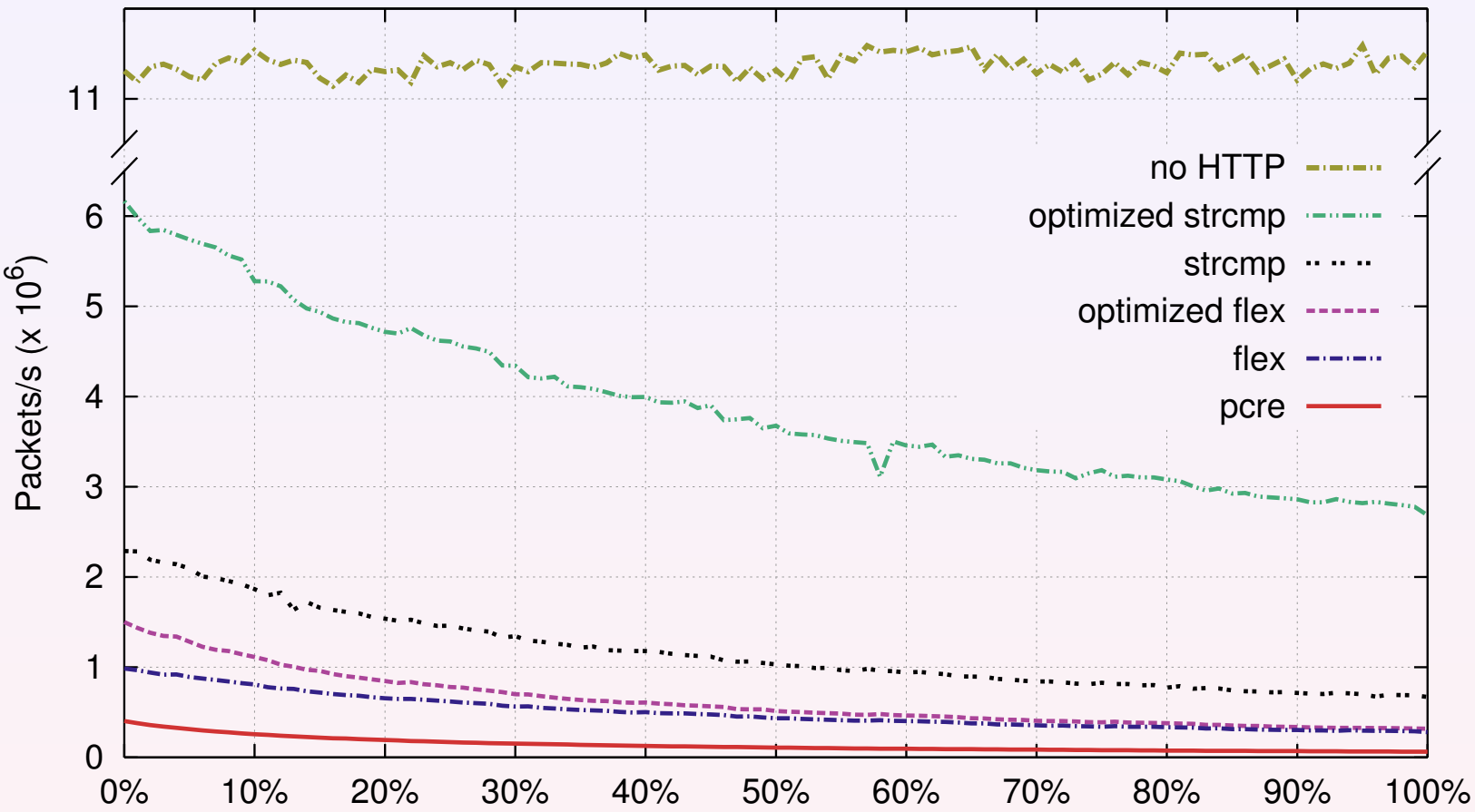
Measurement

1) Throughput measurement
2) Parsed HTTP header fields impact
3) Packet content effect

Part IV

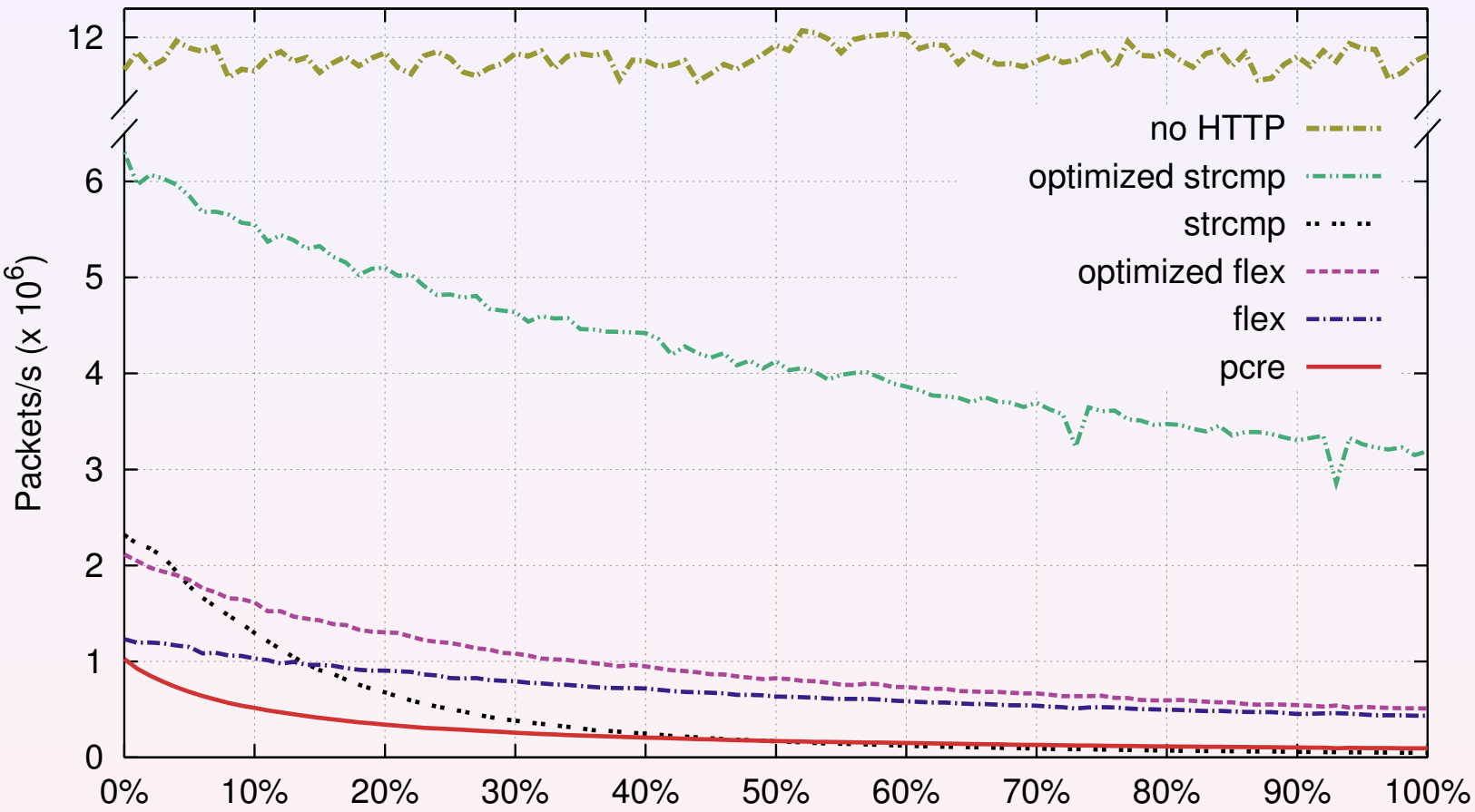
Results

Throughput – 1500 B Snaplen



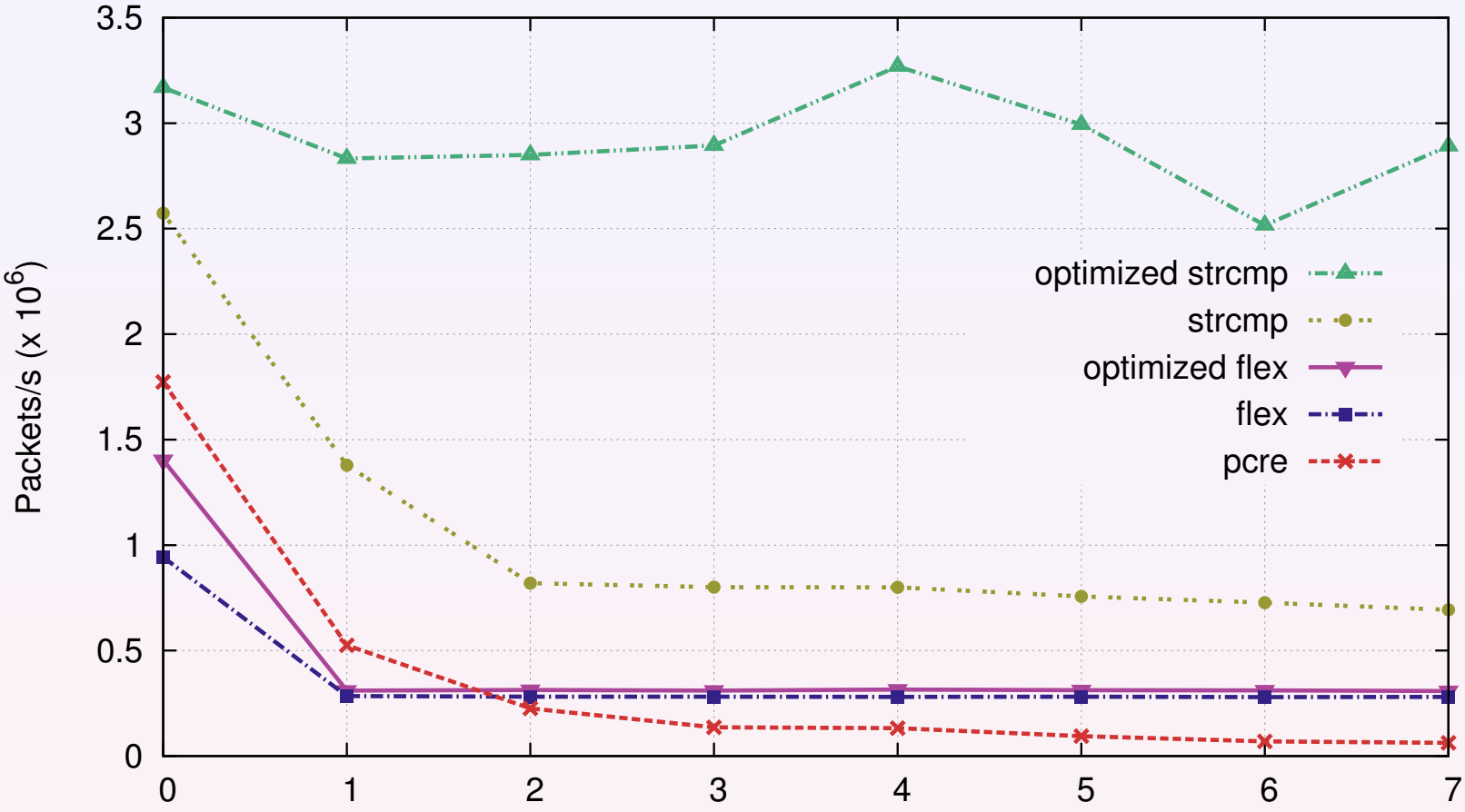
- Throughput for data with x % of HTTP header packets

Throughput – 384 B Snaplen



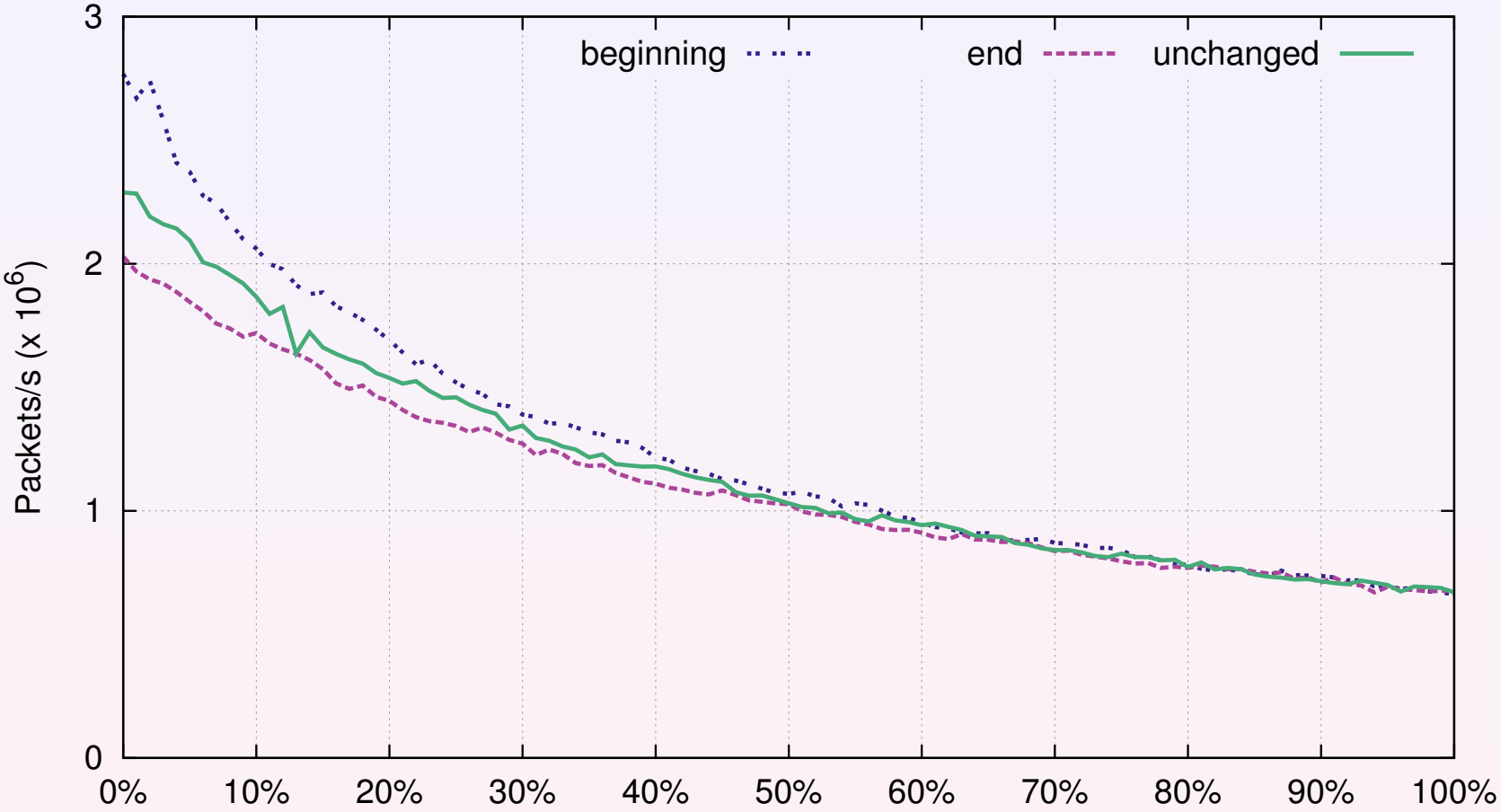
- Throughput for data with x % of HTTP header packets

Parsed HTTP Header Fields Impact



: An HTTP parser throughput for 1500 B packets; supported fields - (0) *none* - HTTP protocol labeling, (1) *+host*, (2) *+method*, (3) *+status code*, (4) *+request URI*, (5) *+content type*, (6) *+referer*, (7) *+user agent*

Packet Content Effect - Strcmp Parser



Packet content effect - packet length 1500 B.

Part V

Conclusion

Conclusion

Summary

- Application data is required to ensure high level of security
- Fast parsing algorithms, throughput deterioration
- Hand-written parsers vs. generated parsers

Future Work

- Extensibility - new protocols, more thorough inspection
- Increasing throughput - examine only necessary data
- Data processing - storage and evaluation

Thank You For Your Attention!

Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement

Petr Velan
velan@ics.muni.cz

Tomáš Jirsík
jirsik@ics.muni.cz

Pavel Čeleda
celeda@ics.muni.cz

Plugins for HTTP Monitoring
http://www.muni.cz/ics/920232/web/http-plugins