DE-CIX

PCAP BGP Parser

NANOG 69, Washington D.C.

Christoph Dietzel[1,2], Tobias Hannaske[1]

[1] DE-CIX, Research & Development

[2] INET, TU Berlin

Where networks meet

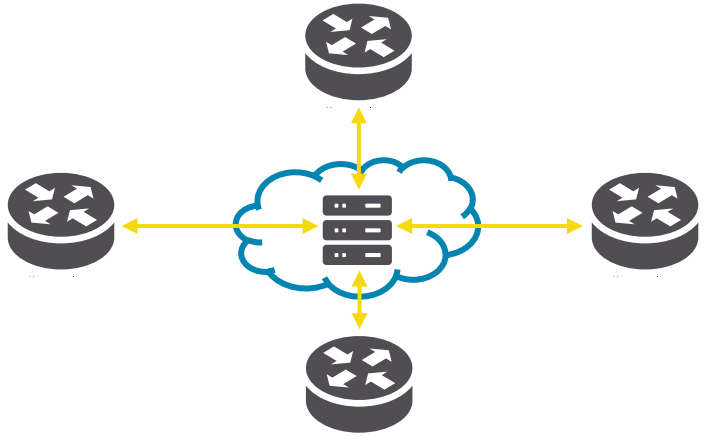
www.de-cix.net

IXPs‘ Route Servers

- Processing a significant amount of data

- Crucial information for IXPs

- What to do with those route server data?
  - Customer debugging assistance
  - Historic analysis (new routes, new peaks)
  - Incidents (route hijacks, route leaks)



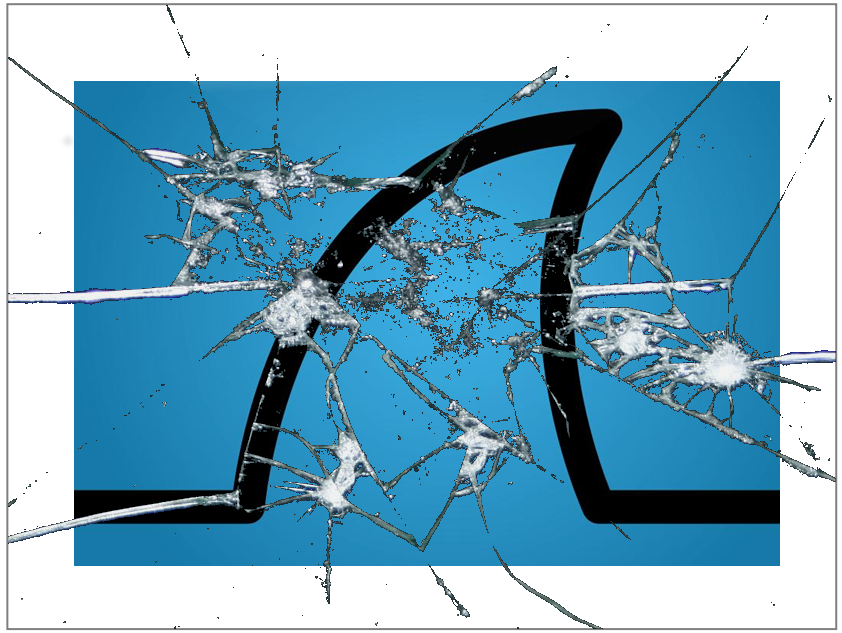
Data & Information Export Limitations of BIRD

- Limited long-term export of BGP information
- No continous export of MRT for BIRD
- No simple filtering before MRT exports
- No insights into incoming BGP advertisments

Why NOT Wireshark / tshark?

- Complex and cumbersome
- Output hard to process in automated fashion
- BGP support, but not built for BGP

```
cat file.pcap | tshark -i - -Y 'bgp.type == 2' \
-T fields \
-e frame.time \
-e bgp.nlri_prefix \
-e bgp.prefix_lenght \
-e bgp.path_attribute.community_as \
-e bgp.path_attribute.community_value
```



Solution: PCAP BGP Parser (pbgpp)

- Python 2.7 and 3.x
- Open Source (github.com/de-cix/pbgp-parser)
- PyPI package (pypi.python.org/pypi/pbgpp)
- Apache License 2.0

```
cat file.pcap | tshark -i - -Y 'bgp.type == 2' \
-T fields \
-e frame.time \
-e bgp.nlri_prefix \
-e bgp.prefix_lenght \
-e bgp.path_attribute.community_as \
-e bgp.path_attribute.community_value
```

```
cat file.pcap | pbgpp - -f LINE --fields \
timestamp, prefixes, communities
```

Features

- Input: PCAP from file, stdin and live interface (beta)
- Output: human readable, JSON, line based (arbitrary fields)
- Easily extendable due to modular application structure

Filtering & Performance

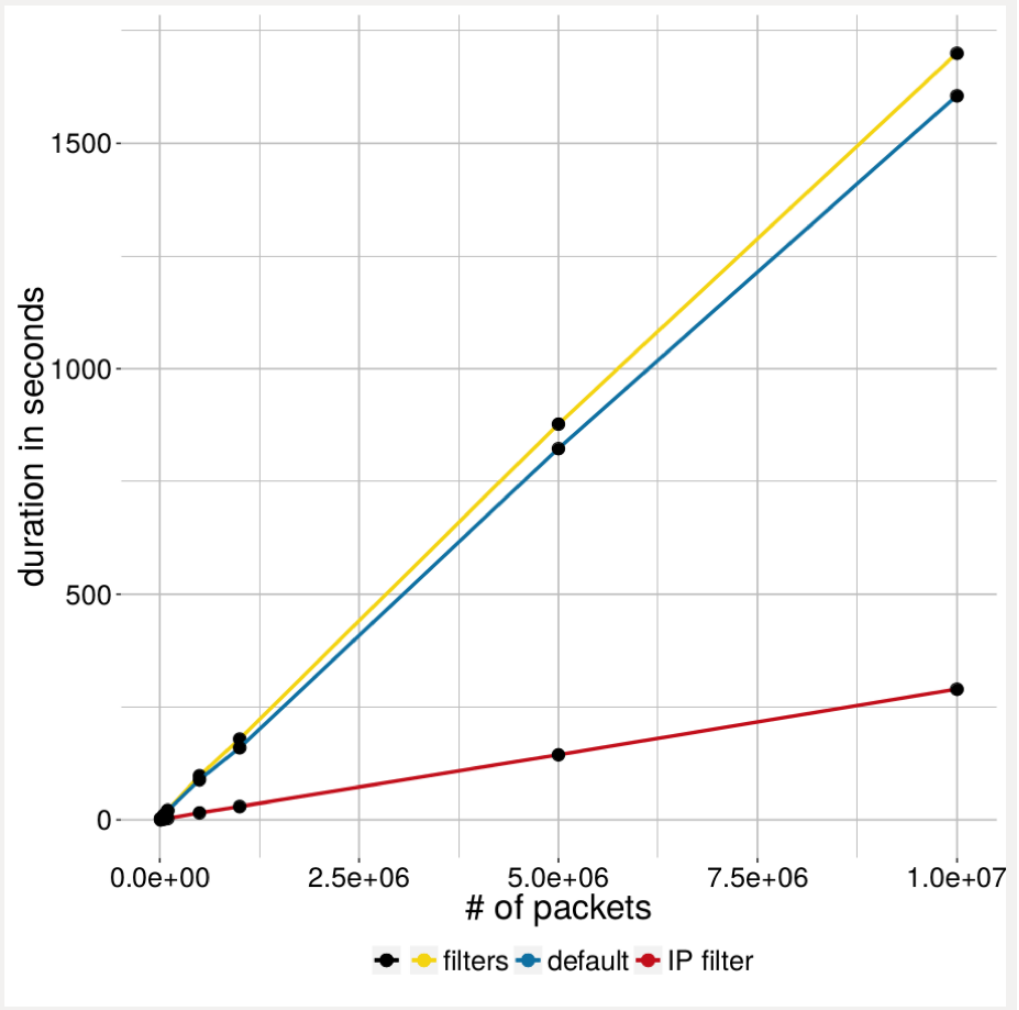
- Filtering in two steps (pre-parsing and post-parsing)
  - Filter by Layer 2 / 3 information and BGP specific fields

- Advanced filtering features
  - Combining filters as desired (logical AND and OR available)
  - Negative filtering (logical NOT)

```
--filter-nlri 127.0.0.0/8 --filter-nlri 192.168.1.0/32 --filter-next-hop ~1.1.1.1 --filter-message-type UPDATE
```

Evaluation: Correctness and Performance

- Performance evaluation with different settings
- Evaluated correctness with many hours of RS dumps
- Compared tshark output with pbgpp output



Demonstration: Example Use Case

- Task: visualize distribution of BGP Hold Time values

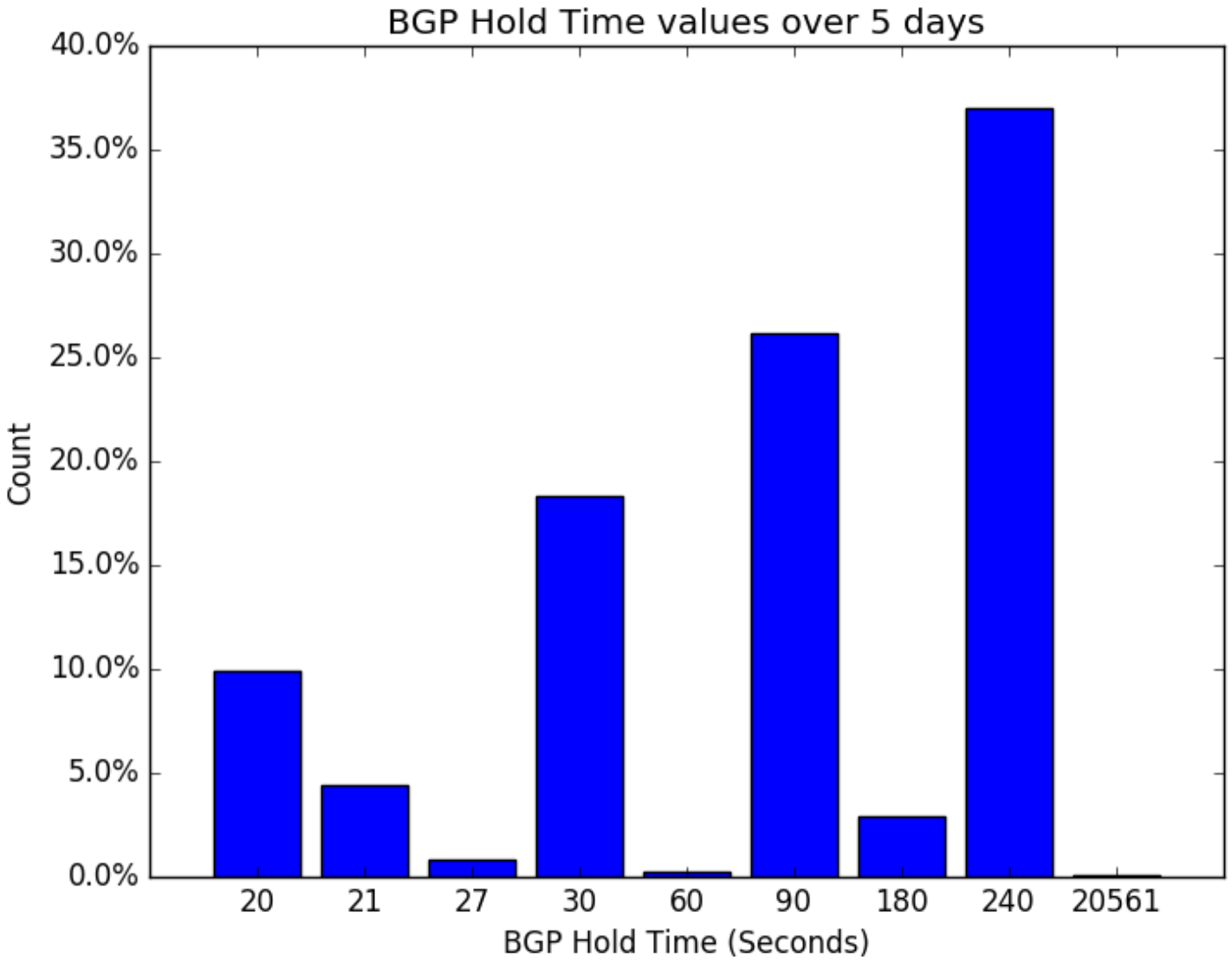
```
zcat dump.pcap.gz | pbgpp - --filter-message-type OPEN --fields hold_time \
-f LINE -p FILE -o output.txt
```

- Output: list of integer values separated by line break and writes it into a file

- Visualization: e.g., Python & matplotlib, R, ...

Demonstration: Example use case

- Single line to call pbgpp
- Saves time
- Ad-hoc analysis



PCAP BGP Parser (pbgpp)

DE-CIX

GitHub: github.com/de-cix/pbgp-parser
PyPI: pypi.python.org/pypi/pbgpp