DE CIX

PCAP BGP Parser

DENOG 8, Darmstadt

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

[1] Research and Development, DE-CIX

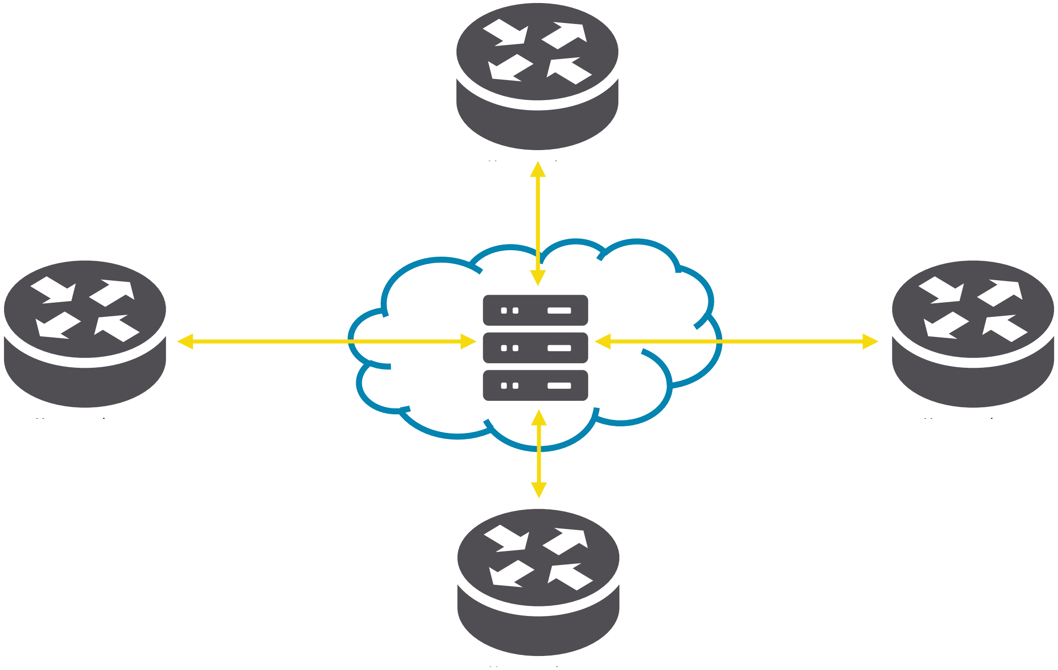
[2] INET, TU Berlin

Where networks meet

www.de-cix.net

IXPs’ Route Servers

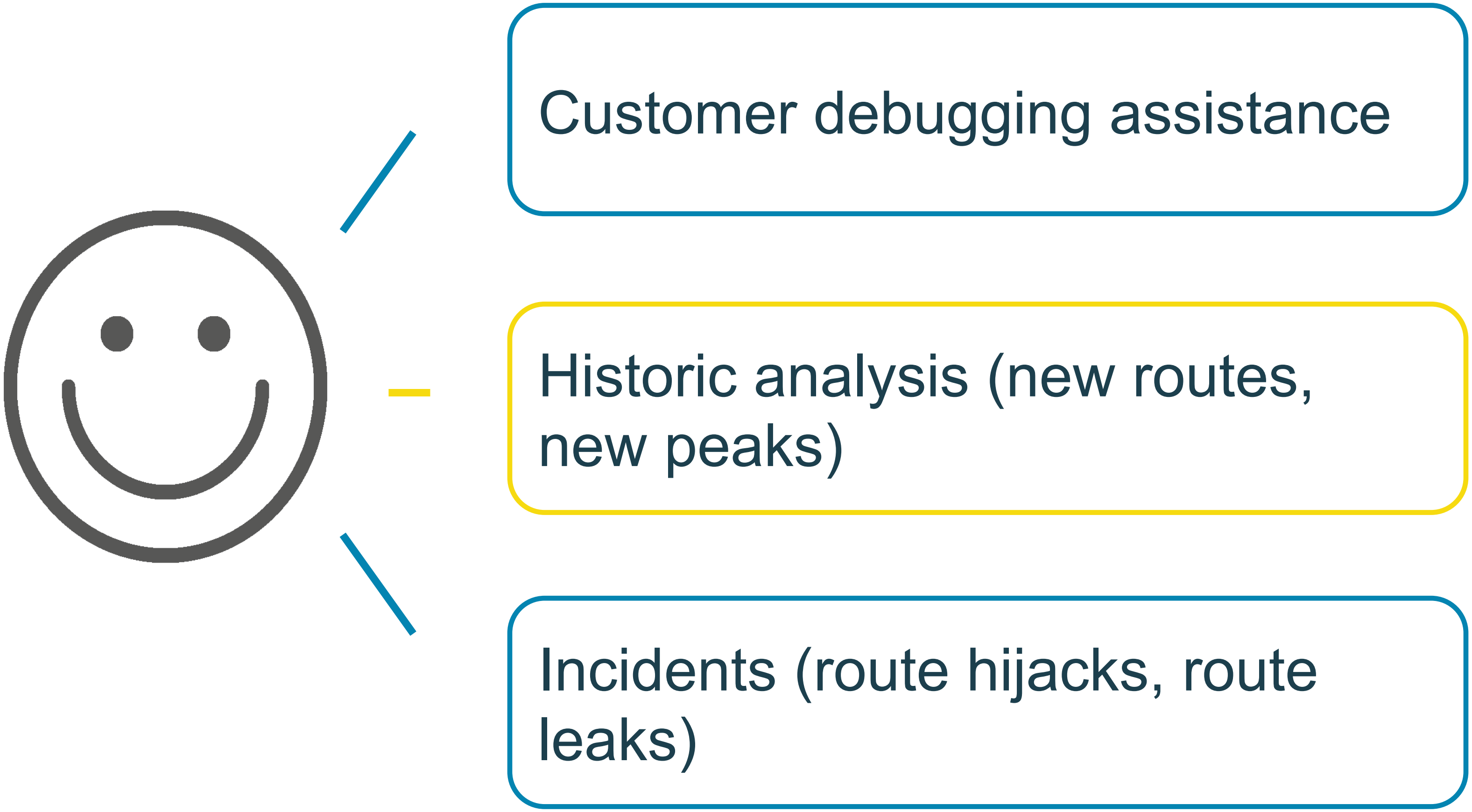
- They exist (yees!)
- Process a significant amount of data
- Crucial information for IXPs



Where networks meet

www.de-cix.net

Route Server as BGP Speaker at IXPs



Customer debugging assistance

Historic analysis (new routes, new peaks)

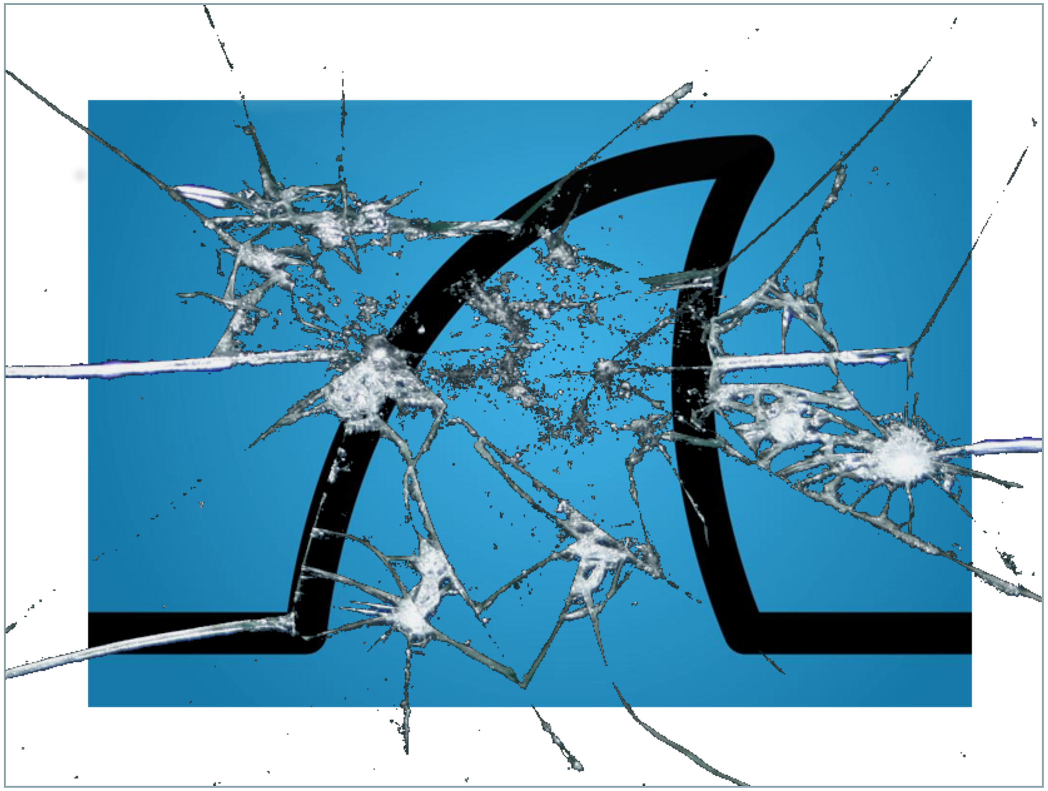
Incidents (route hijacks, route leaks)

BIRD’s Information Export Limitations

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

Solution? - tcpdump & tshark!(?)

- Complex / cumbersome
- Output hard to process in automated fashion
- Not build for BGP



PCAP BGP Parser (pbgpp)

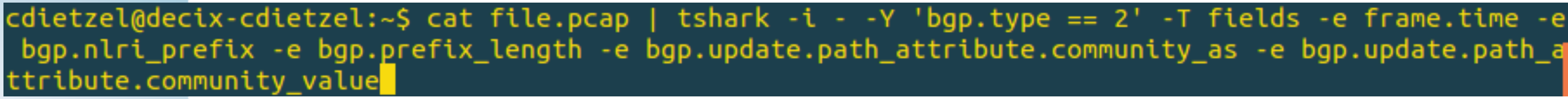
» Python 2.7 and 3.x

» Open Source (github.com/de-cix/pbgp-parser)

» PyPi package (https://pypi.python.org/pypi/pbgpp/0.2.3)

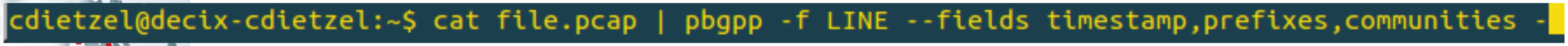
» License Apache 2.0

tshark vs. pbgpp



```
cdietzel@decix-cdietzel:~$ cat file.pcap | tshark -i - -Y 'bgp.type == 2' -T fields -e frame.time -e
 bgp.nlri_prefix -e bgp.prefix_length -e bgp.update.path_attribute.community_as -e bgp.update.path_a
ttribute.community_value
```

vs.



```
cdietzel@decix-cdietzel:~$ cat file.pcap | pbgpp -f LINE --fields timestamp,prefixes,communities -
```

=

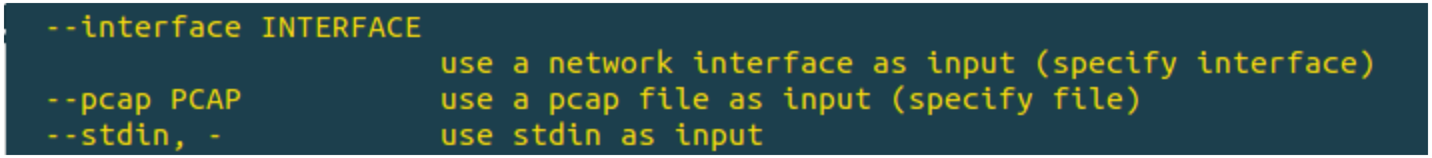
```
Nov 17, 2015 14:35:08.034535000 CET
145.120.16.0,194.53.0.0 23,24
286,286,286,286,6695,12859
286,3031,4516,4990,47541,4000
```

vs.

```
1447767308.34535
145.120.16.0/23
0:6695;286:286;286:3031;286:4516;286:4990;6695:47541;12859:4000
```

Features - Input

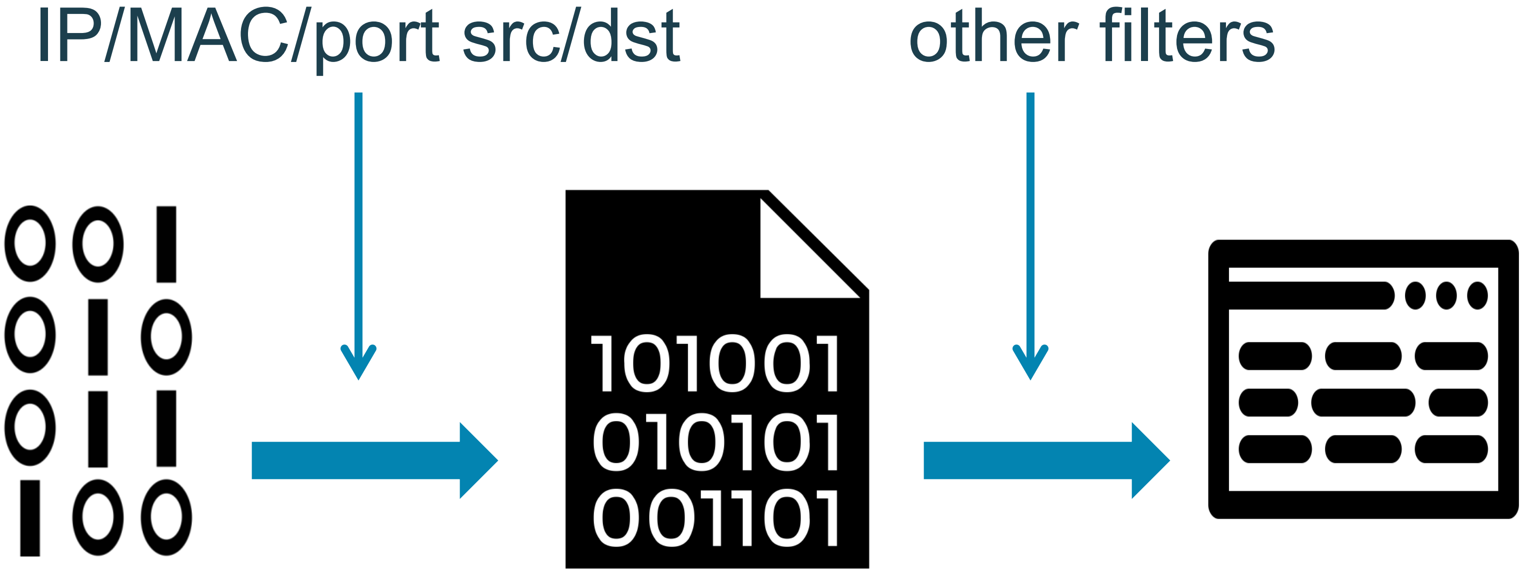
- Reads PCAP files (not PCAPng yet - would be easy to implement)
- BGP parser can read from stdin (PCAP format)
- Live reading from network interface (beta!)
- Extending is possible, as long as it relies on raw packet data



```
  --interface INTERFACE
                        use a network interface as input (specify interface)
  --pcap PCAP           use a pcap file as input (specify file)
  --stdin, -            use stdin as input
```

Features - Filtering

» Filtering before and after parsing

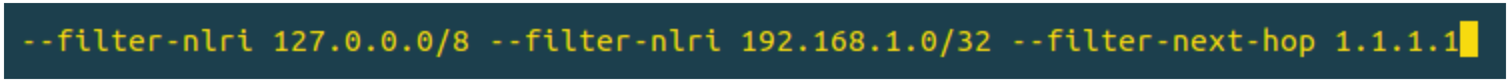


Features – Filtering

Filter field	Values; Description
Message type	OPEN, UPDATE, NOTIFICATION, ROUTE-REFRESH, KEEPALIVE
NLRI	Prefix, e.g., 80.81.82.0/24
Withdrawn route	Prefix, e.g., 80.81.82.0/24
Next hop	IP, e.g., 80.81.82.1
ASN in AS path	ASN, e.g., 6339
Last ASN in AS path	ASN of the neighbor AS
Community	BGP Community, e.g., 6993:666
Source IP	Neighbor router’s IP
Destination IP	Neighbor router’s IP
Source MAC	Neighbor router’s MAC
Destination MAC	Neighbor router’s MAC
NOT IP, MAC, …	ANY, e.g., ~192.168.0.10

Features - Filtering

- Filtering to display specific BGP messages – only messages that apply are displayed
- Combine any filters as desired
- Different values for same filter are chained with a logical *OR*
- Different filters are chained with a logical *AND*



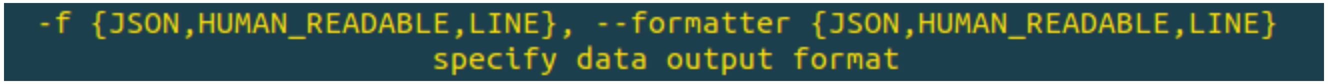
```
--filter-nlri 127.0.0.0/8 --filter-nlri 192.168.1.0/32 --filter-next-hop 1.1.1.1
```

- NLRI must contain either *127.0.0.0/8 OR 192.168.1.0/32 AND* next hop must be *1.1.1.1*

Where networks meet

www.de-cix.net

Features - Output



```
-f {JSON,HUMAN_READABLE,LINE}, --formatter {JSON,HUMAN_READABLE,LINE}
                        specify data output format
```

- Human readable
  - Basic information about BGP msgs
  - Easy to read
  - Includes all important fields such as NEXT_HOP, AS_PATH, NLRI and/or WITHDRAWALS, etc.
- JSON
- All BGP msgs + meta information (capture specific data such as timestamp, source/dest ip/mac/port)
  - RFC 7159 (see Python internal json-package)
  - One JSON string per line
- Line based
  - User can specify fields to be displayed
  - Not all fields supported, yet
  - Available fields for line based output are:
    - NLRI, AS_PATH, NEXT_HOP, Communities, Source/Destination IP, Timestamp, Message Types

Features – Community Contribution

- Standard Py dir structure & easy system tool installation
  - During my RIPE 73 talk
  - Thanks to mxxxc
- Large BGP Communities
  - draft-ietf-idr-large-community-06
  - Thanks to pierky

More to come, hopefully!

Evaluation Correctness

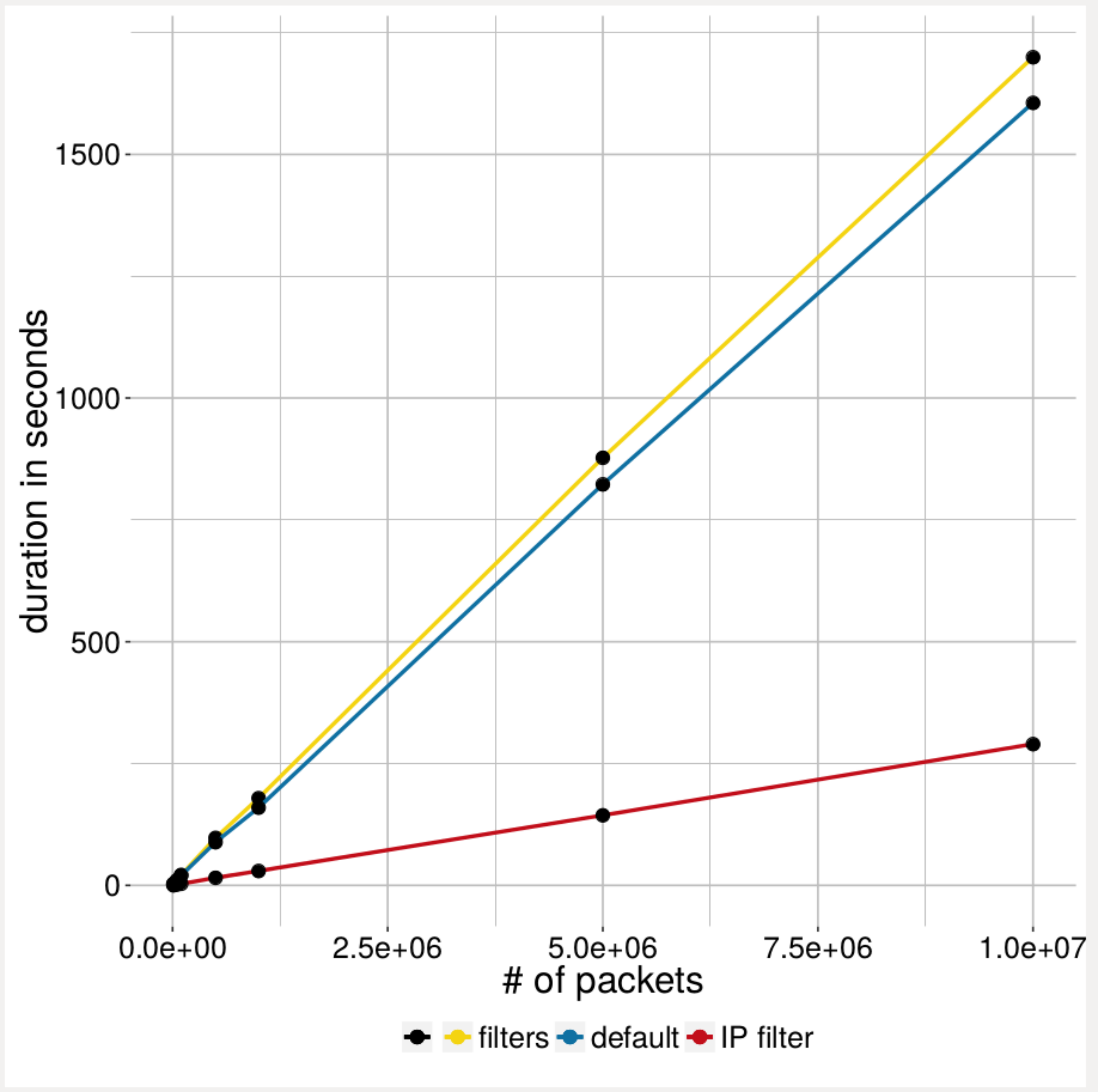
- Compared results of pbgpp and tshark
  - E.g., no. of packets after filtering, timestamps
  - DE-CIX RS dump of several hours

✓ Correct, but we keep looking

Limitations

- Packet reordering issue
- Not all features implemented yet

Evaluation Performance



Where networks meet

www.de-cix.net

Conclusion / Contribution

- Open source PCAP BGP Parser (pbgpp)
- Apache 2.0 license
- Wide range of flexible input/output parameters
- Strong filtering capabilities
- Nice to integrate in shell/bash/python toolchain
- Fast enough perform “live” parsing for RS dumps from large IXP

github.com/de-cix/pbgp-parser

PyPI package available! (https://pypi.python.org/pypi/pbgpp/0.2.3)