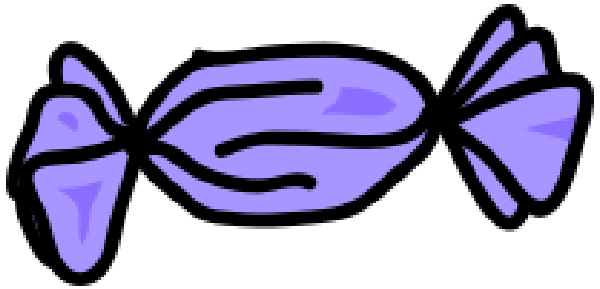


TRAFFIC TAFFY
TEMPORAL ANALYSIS OF FLUCTUATING FLOWS

EXPLORING THE BUMPS IN THE INTERNET HIGHWAY

Wes Hardaker
<hardaker@isi.edu>

2024-02-09

Network operators are plagued with odd anomalies

Current solutions

- Human search for obvious repetition using **tcpdump or wireshark**
  - Works well for huge spikes
  - Seeing the “obvious” decreases in smaller anomalies
  - Prone to missing subtle secondary signals
  - Requires significant knowledge of protocol details
- Automated **traffic analysis** tools
  - Diagnosis common components in an anomalies
  - Prone to false positives
    - it may report about normal traffic too

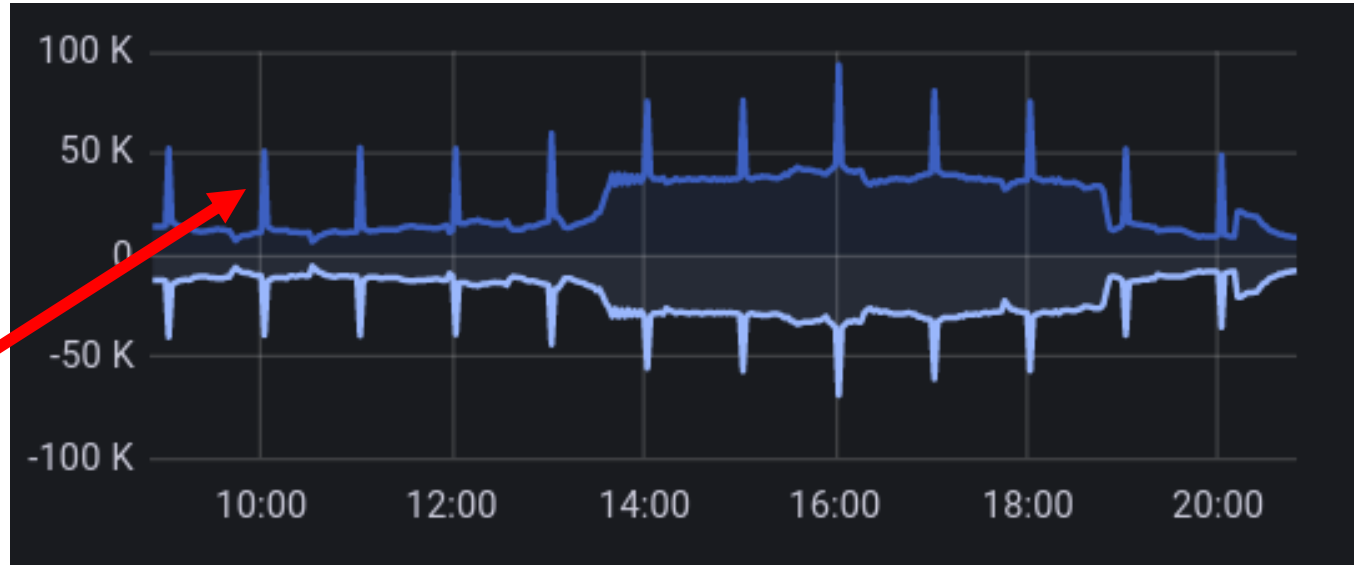

There's gotta be a better way!

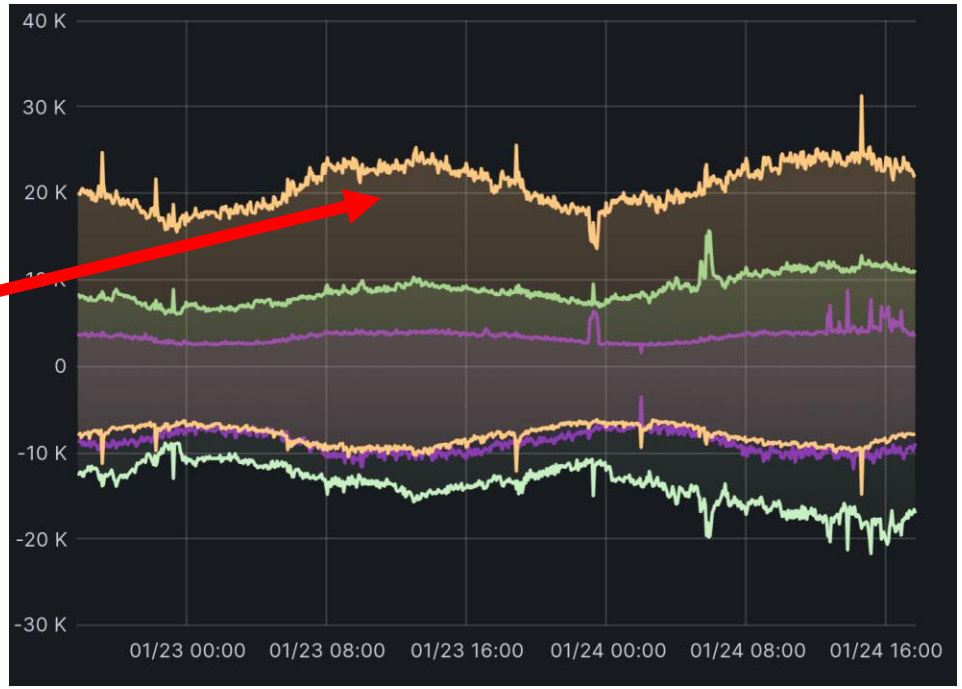
A specific problem space

Assumptions:

- Major shifts in traffic are from a **single cause**
  - Something has clearly changed. *What?*

- *"more of the same" and ramp-ups are out of scope*
  - *AKA diurnal patterns*





Insight: let's compare the oddities against a baseline

Defining the problem space

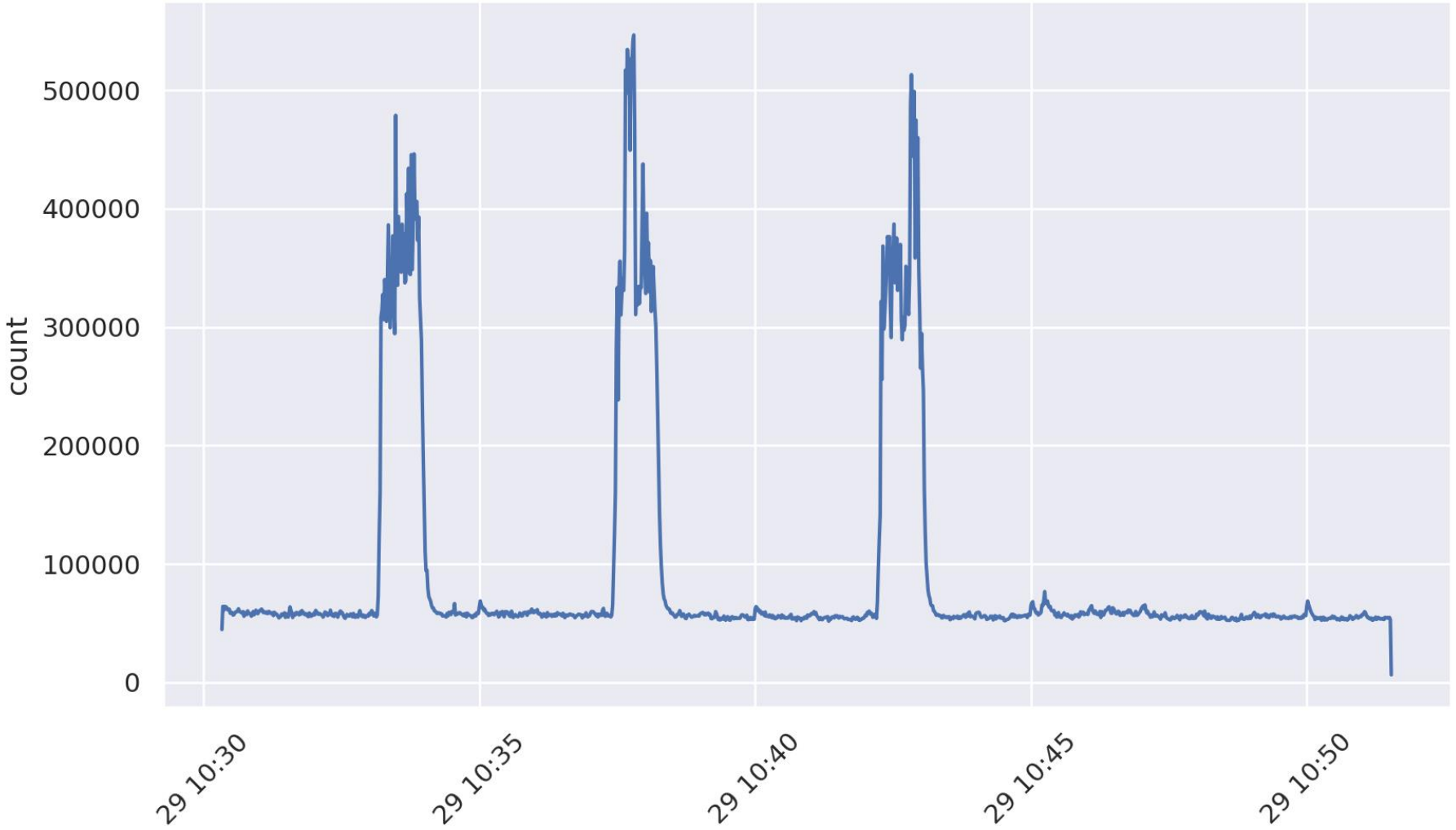
- Goal: analyze **single-cause changes**
  - Show ***what has changed***
  - *Show* ***when it changed***
- Need a "traffic diff" tool to compare "left" and "right":
  - "Left": a sample of regular traffic
  - "Right": a sample from an anomaly
  - "Delta": what is different between them
- "left" and "right" samples can be:
  - different files
  - different time ranges within files

Introducing traffic-taffy

- Approach:
  - Perform deep packet inspection of a **base-line**
  - Perform deep packet inspection of an **anomaly**
  - Compare levels for ***each value*** of ***each protocol field***
  - **Sort, Filter and Report** based on findings

- Some of the tools:
  - taffy-dissect: enumerates field counts in a pcap file
  - taffy-compare: compares one file/time-range against another
  - taffy-graph: graphs enumerated fields in pcap files
- Easy to install: ***pip install traffic-taffy***

Case Study 1: three large bumps seen at b.root-servers.net



Dataset:

- Three 5x spikes
- At 1 anycast site
- What are they?

- Can we find the
- root cause?

Graph produced with taffy-graph

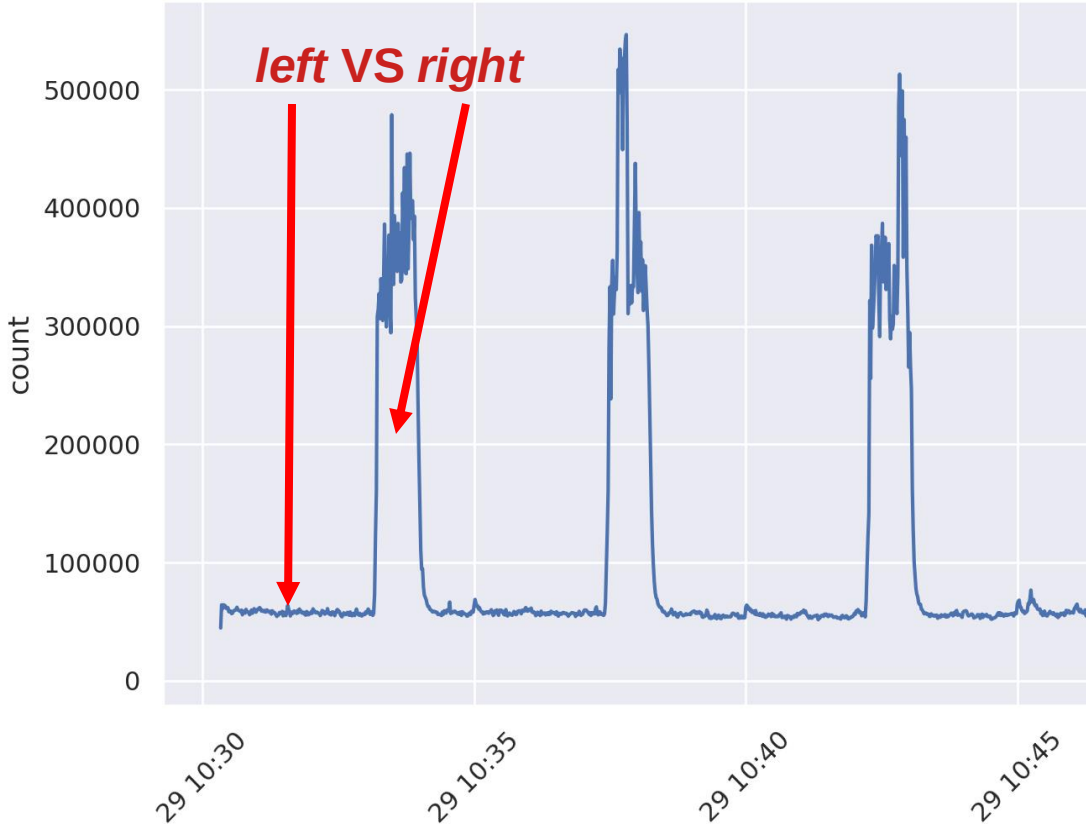
taffy-compare: find differences between points in time

Taffy-compare:

- Takes PCAP data from two points in time
- Uses the *left* side as a “normal” profile
- Identifies major shifts in the *right* side

Output: colorized results to the console

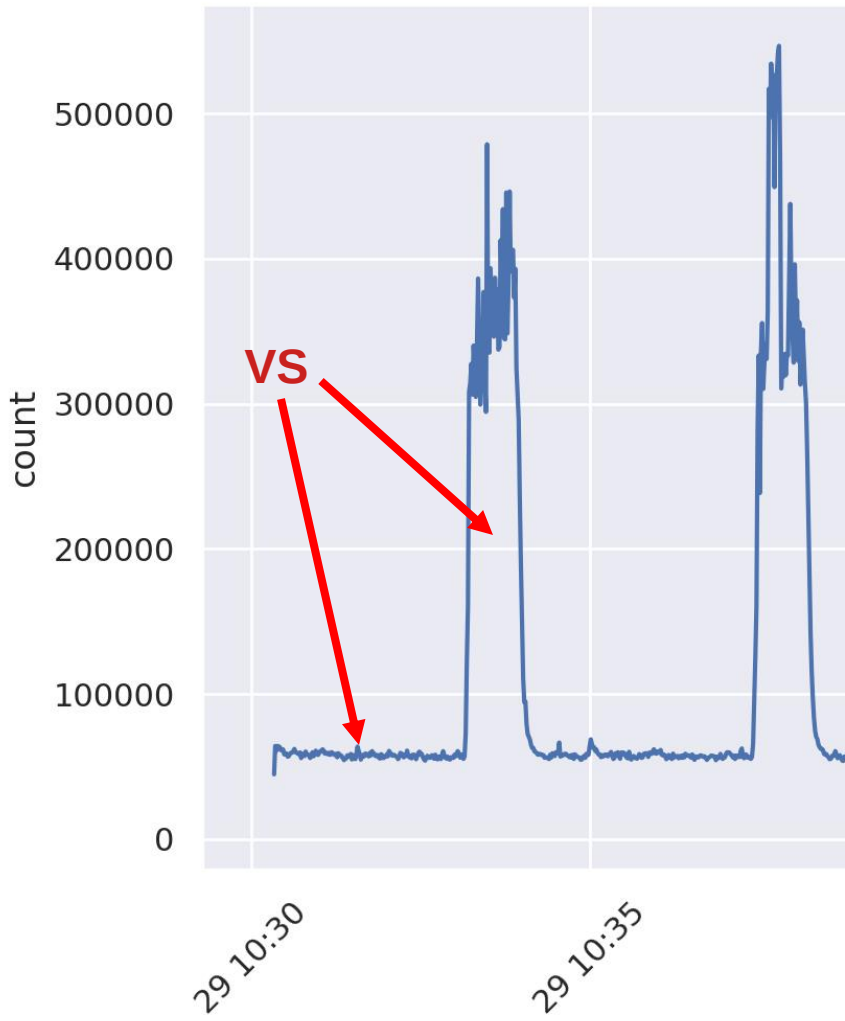
- Total counts per protocol field
  - Left and right
- Percentage of traffic for each field value
- Deltas between both values and percentages
- All filterable by threshold values



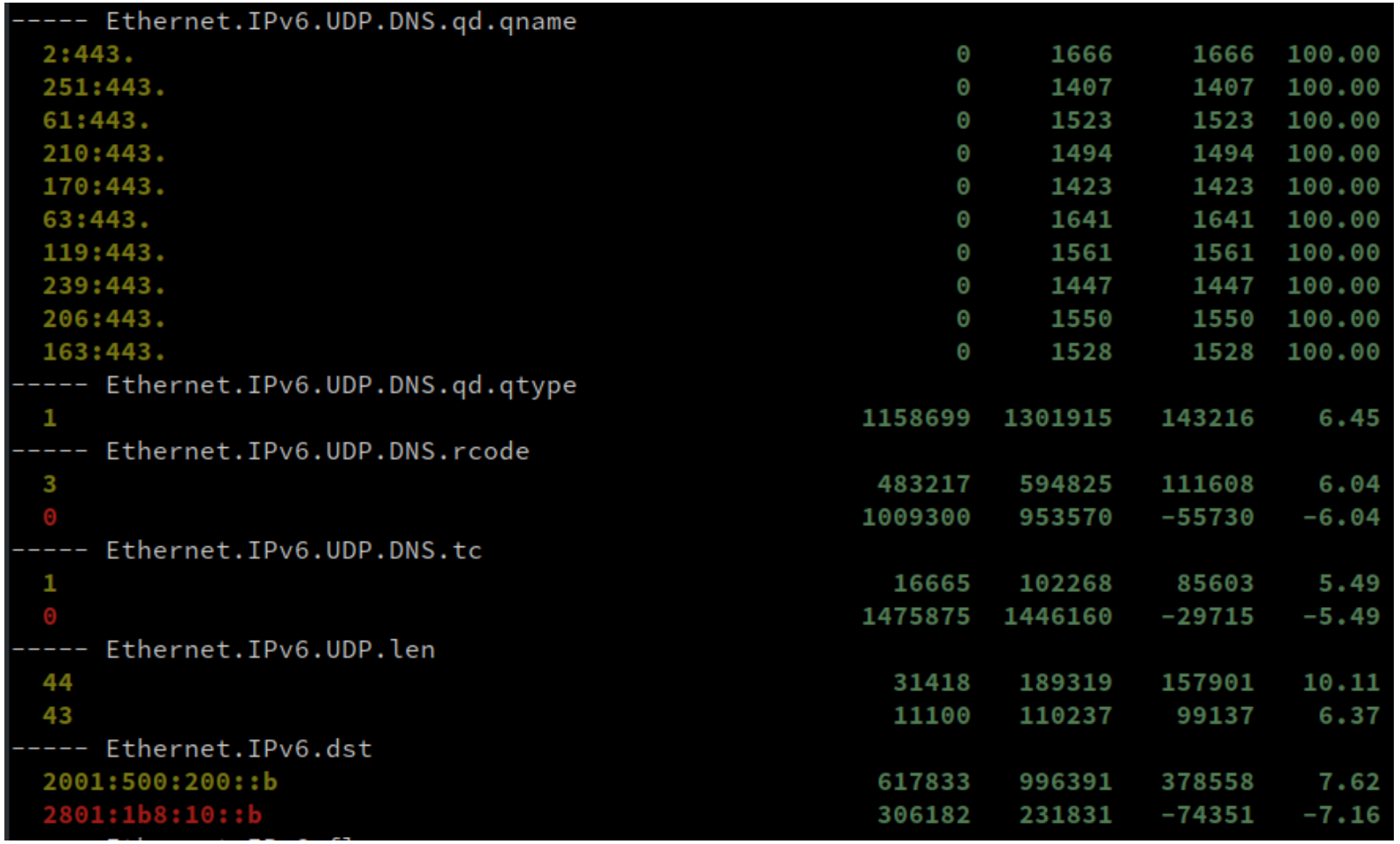
Value	Left	Right	Delta	Left %	Right %	Delta-%
----- Ethernet.IP.UDP.DNS.cd						
0	1961290	2668655	707365	39.43	54.29	14.86
1	3012629	2246678	-765951	60.57	45.71	-14.86

taffy-compare: find differences between points in time

```
----- Ethernet.IPv6.UDP.DNS.qd.qname
  2:443.                                0     1666     1666  100.00
  251:443.                              0     1407     1407  100.00
  61:443.                               0     1523     1523  100.00
  210:443.                              0     1494     1494  100.00
  170:443.                              0     1423     1423  100.00
  63:443.                               0     1641     1641  100.00
  119:443.                              0     1561     1561  100.00
  239:443.                              0     1447     1447  100.00
  206:443.                              0     1550     1550  100.00
  163:443.                              0     1528     1528  100.00
----- Ethernet.IPv6.UDP.DNS.qd.qtype
  1                               1158699  1301915   143216    6.45
----- Ethernet.IPv6.UDP.DNS.rcode
  3                                483217   594825   111608    6.04
  0                               1009300   953570   -55730   -6.04
----- Ethernet.IPv6.UDP.DNS.tc
  1                                 16665   102268    85603    5.49
  0                               1475875  1446160   -29715   -5.49
----- Ethernet.IPv6.UDP.len
  44                                31418   189319   157901   10.11
  43                                11100   110237    99137    6.37
----- Ethernet.IPv6.dst
  2001:500:200::b                  617833   996391   378558    7.62
  2801:1b8:10::b                   306182   231831   -74351   -7.16
```



taffy-compare example: colored console differences



```
----- Ethernet.IPv6.UDP.DNS.qd.qname
  2:443.                                    0     1666     1666  100.00
  251:443.                                  0     1407     1407  100.00
  61:443.                                   0     1523     1523  100.00
  210:443.                                  0     1494     1494  100.00
  170:443.                                  0     1423     1423  100.00
  63:443.                                   0     1641     1641  100.00
  119:443.                                  0     1561     1561  100.00
  239:443.                                  0     1447     1447  100.00
  206:443.                                  0     1550     1550  100.00
  163:443.                                  0     1528     1528  100.00
----- Ethernet.IPv6.UDP.DNS.qd.qtype
  1                                   1158699  1301915   143216    6.45
----- Ethernet.IPv6.UDP.DNS.rcode
  3                                    483217   594825   111608    6.04
  0                                   1009300   953570   -55730   -6.04
----- Ethernet.IPv6.UDP.DNS.tc
  1                                     16665   102268    85603    5.49
  0                                   1475875  1446160   -29715   -5.49
----- Ethernet.IPv6.UDP.len
  44                                    31418   189319   157901   10.11
  43                                    11100   110237    99137    6.37
----- Ethernet.IPv6.dst
  2001:500:200::b                      617833   996391   378558    7.62
  2801:1b8:10::b                       306182   231831   -74351   -7.16
```

Leaked docker port mappings

For A records

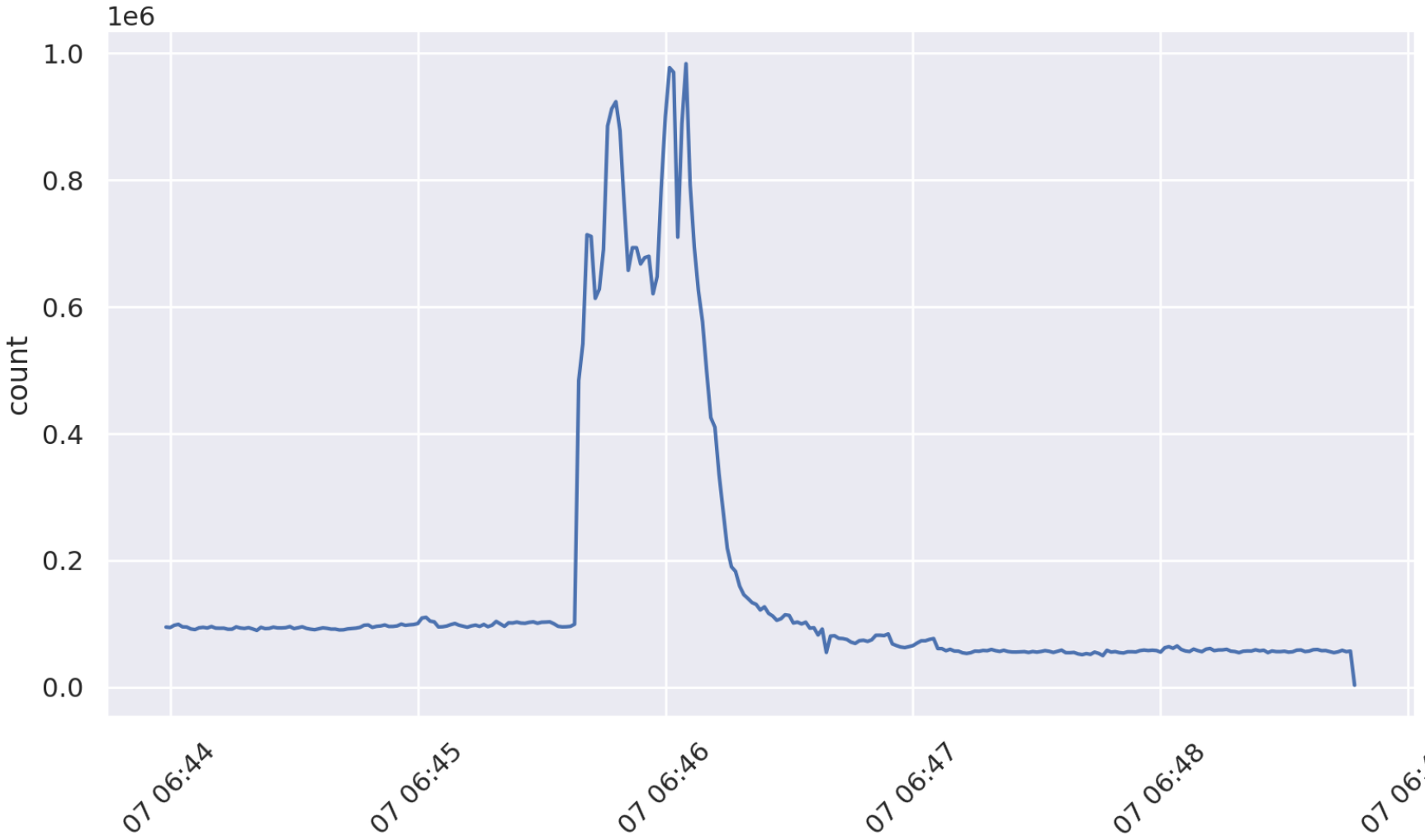
We return more NXdomains

RRL limits hit (increasing TC)

Heading to our older IPv6 address

Important note: I did not pick these fields to study – the tool did!

Case Study 2: A large DDoS attack against b.root-servers.net



A USC/ISI published dataset

Dataset Description

- No fixed query name
- 554 bytes requests
- Randomized sources

Traffic-Taffy findings

- Many previously unknown secondary effects

(Some) results from taffy-compare

Decrease in the Checking Disabled bit:

Value	Left	Right	Delta	Left %	Right %	Delta-%
----- Ethernet.IP.UDP.DNS.cd						
0	1961290	2668655	707365	39.43	54.29	14.86
1	3012629	2246678	-765951	60.57	45.71	-14.86

Increase in odd DNS operation codes:

Value	Left	Right	Delta	Left %	Right %	Delta-%
----- Ethernet.IP.UDP.DNS.opcode						
7	0	233422	233422	0.00	4.75	100.00
8	0	220067	220067	0.00	4.48	100.00
14	0	233096	233096	0.00	4.74	100.00
9	0	234615	234615	0.00	4.77	100.00
15	0	222026	222026	0.00	4.52	100.00

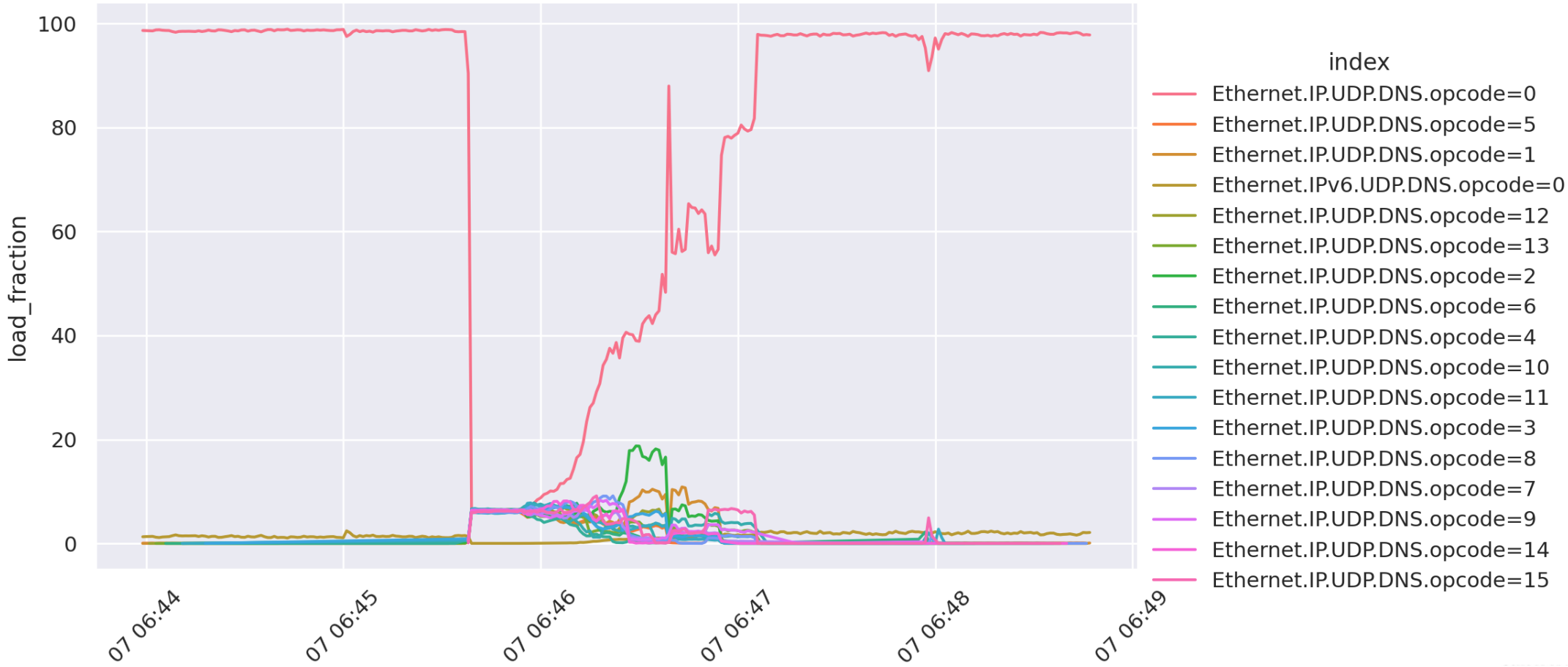
Increase in queries to example.com:

Value	Left	Right	Delta	Left %	Right %	Delta-%
----- Ethernet.IP.UDP.DNS.qd.qname						
www.example.com.	259	787526	787267	0.00	32.05	32.05

Emergency firewall filters can be built on these!

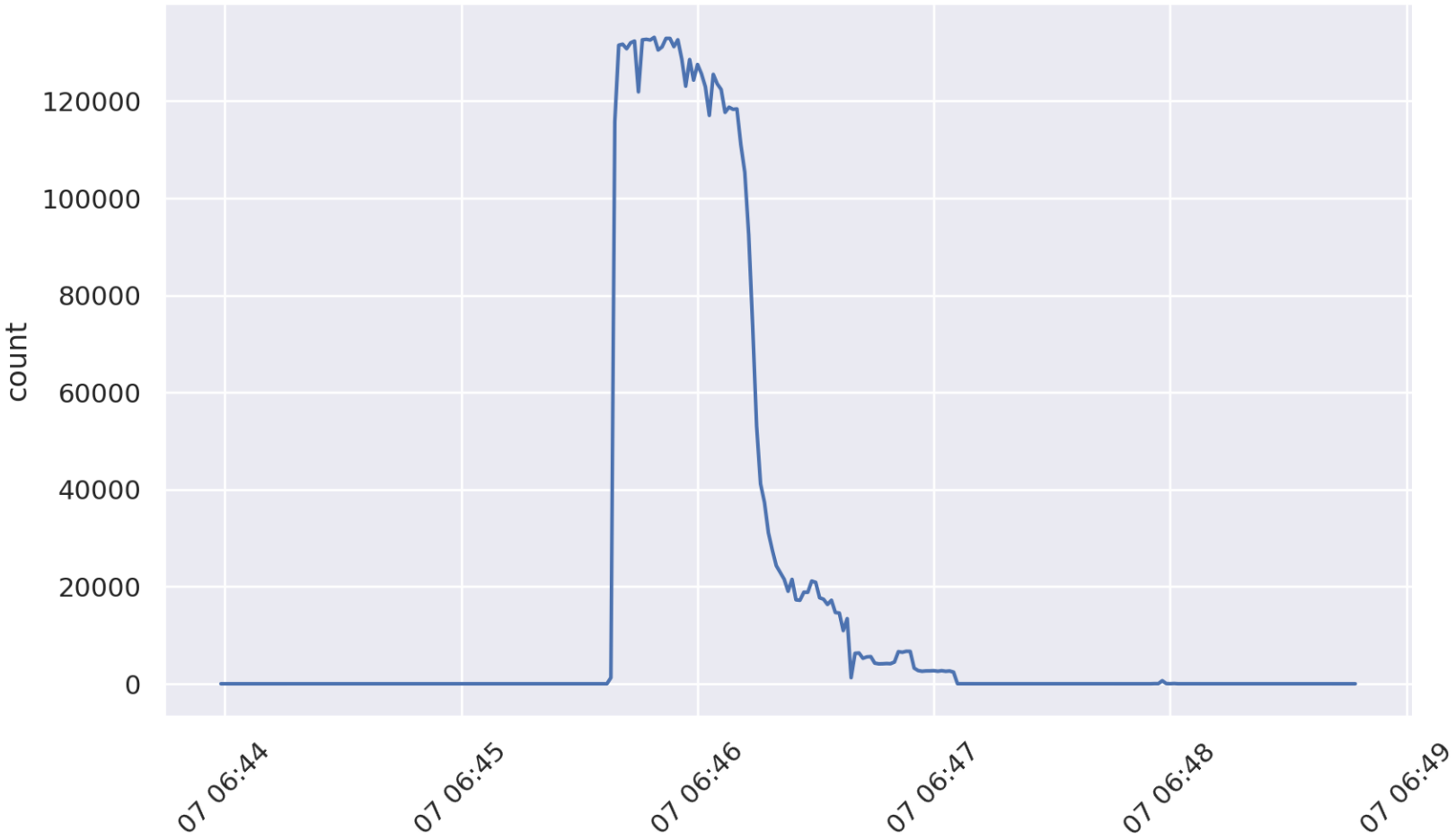
Emergence of unusual opcodes

Value	Left	Right	Delta	Left %	Right %	Delta-%
----- Ethernet.IP.UDP.DNS.opcode						
7	0	233422	233422	0.00	4.75	100.00
8	0	220067	220067	0.00	4.48	100.00
14	0	233096	233096	0.00	4.74	100.00
9	0	234615	234615	0.00	4.77	100.00
15	0	222026	222026	0.00	4.52	100.00



The count of queries for www.example.com went up

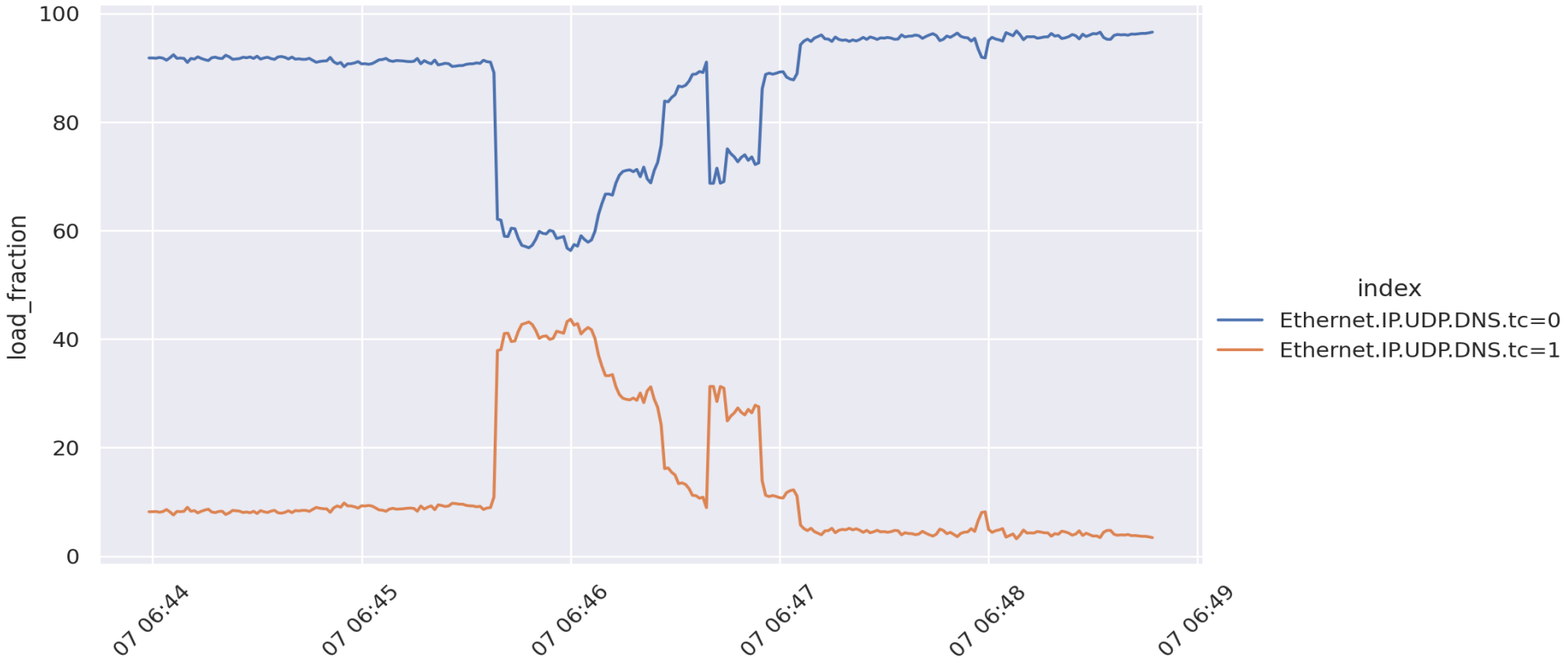
```
----- Ethernet.IP.UDP.DNS.qd.qname
 www.example.com.                    259    787526    787267    0.00    32.05    32.05
```



Analyzing responses: Response Rate Limiting kicked in

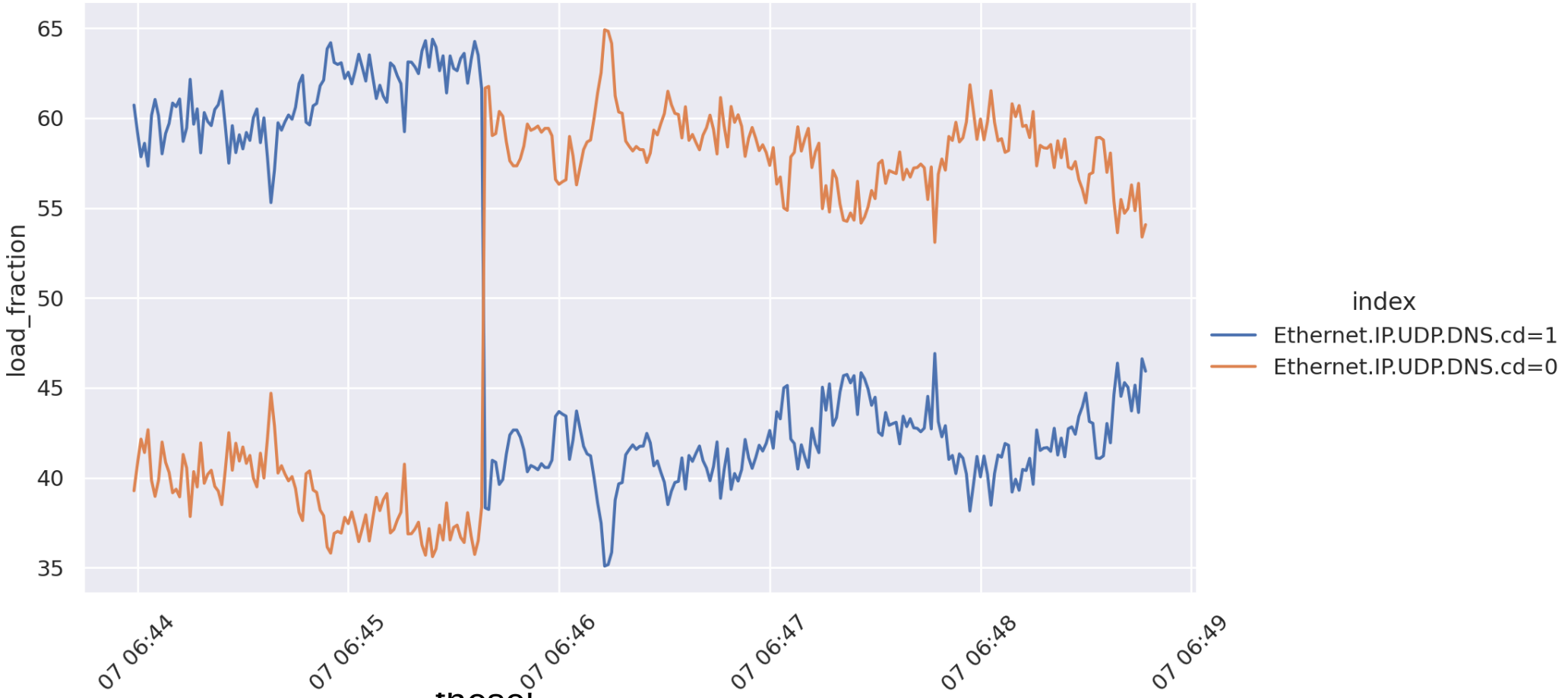
- Increased Truncation (TC) bits seen:

```
----- Ethernet.IP.UDP.DNS.tc
  1                                422630 1574127 1151497  8.50  32.02  23.53
  0                               4551289 3341206 -1210083 91.50  67.98  -23.53
```



Flipping of the CD bit -- Why does cd=1 stay high longer???

Value	Left	Right	Delta	Left %	Right %	Delta-%
Ethernet.IP.UDP.DNS.cd						
0	1961290	2668655	707365	39.43	54.29	14.86
1	3012629	2246678	-765951	60.57	45.71	-14.86



these!

Many planned features to come

taffy-explorer interactive interface

pre-alpha

Detailed graph

Total traffic graph

Browsable report

Future Features

My List

- Category sorting by likelihood
- Different comparison algorithms
- More documentation
- Large dataset improvements:
  - Memory improvements
  - Speed improvements
- Many taffy-explorer improvements
  - e.g. better graphing support with clickable time-ranges

YOUR LIST HERE

- This project is under very active development for another few months
- Looking for early adopters
- Please provide feedback (soon)!

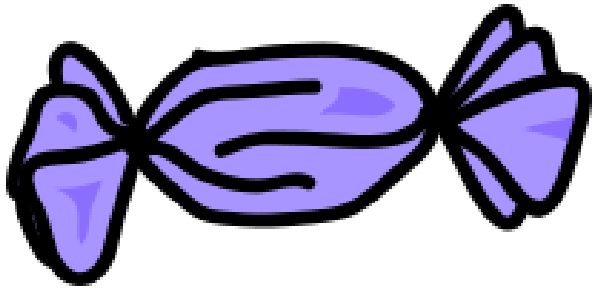
Note: there are also many more existing features not discussed in this presentation (see the documentation)

General Usage Tips

- Always turn on caching: -C
  - It's not a default because it creates files
- Start comparisons you need rapid responses for with small samples:
  - Limit to 10k packets: -n 10000 [cached]
  - Dissection level 2: -d 2 [cached]
    - Eventually you'll always want level 10, but it's CPU and memory intensive
- Start comparisons with large filtering thresholds:
  - Show only differences with at least 1000 counts: -c 1000
  - Show only differences with at least a 10% change: -t 10
  - Show only the top 10 differences: -x 10
- The graphing app supports these too
- *It HELPS human analysis – it doesn't replace it*

Warning… this is easy to recreate

Try it!



- Please test it! (soon)
  - https://traffic-taffy.readthedocs.io/
  - https://github.com/hardaker/traffic-taffy
  - **pip install traffic-taffy**
  - *Warning: it is very new – expect bugs and do e-mail me*

- Thank you to the **Comcast Innovation Fund** for sponsoring this work!

Wes Hardaker <hardaker@isi.edu>