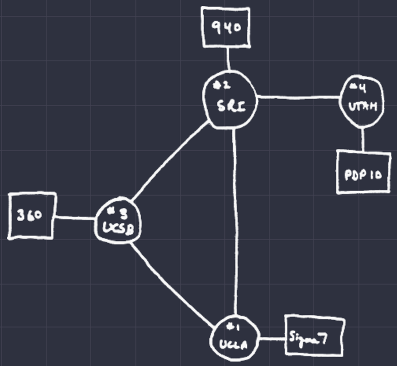
AGREE TO DISAGREE

On the current state of BGP parsing

Johann Schlamp | RIPE89 | Prague



THE ARPA NETWORK, DEC 1969

0:0:742e:2401:4900::/79 | AS2936451170

ANYONE HERE FROM THEIR NOC?

DISCLAIMER

⚠ This talk is not about **»the best«** MRT parser

⚠ This talk is not about the **»performance«** of MRT parsers

ℹ All analyses are based on **full MRT data sets** (first RIB and all UPDATEs) from RouteViews, RIPE RIS and Packet Clearing House (PCH) ≈ **3B routes**

> For single-day analyses, we use recentish data from **January 1st 2024**, for time series the **first day** of each month from Jan'2022 till Jan'2024

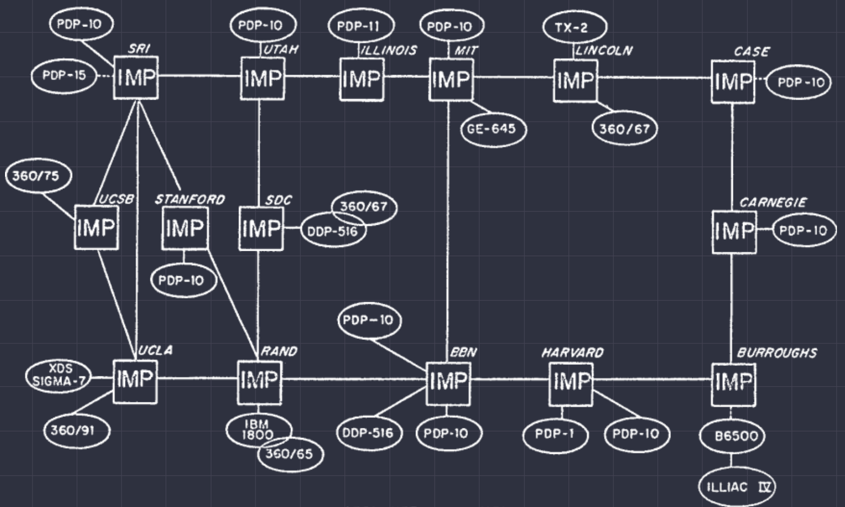
Challenges in parsing BGP/MRT data

Many moving parts

- **Error chaining** BGP standard ⇔ BGP speaker ⇔ BGP exporter ⇔ BGP parser
- **Conflicting goals** be conformant with standard ⇔ preserve the most information
- **Differing use cases** interactive/bulk, standalone/ecosystem, research/operations
- **Implementation** pointer arithmetic, algorithmic decisions (e.g. AS_PATH length > 255?)

Selected problems

RFC791 | RFC904 | RFC1112 | RFC1997 | RFC2042 | RFC2373 | RFC2460 | RFC2545 | RFC2858
RFC2918 | RFC3392 | RFC4271 | RFC4291 | RFC4360 | RFC4364 | RFC4456 | RFC4486 | RFC4493
RFC4684 | RFC4724 | RFC4760 | RFC4761 | RFC5065 | RFC5195 | RFC5291 | RFC5492 | RFC5512
RFC5543 | RFC5701 | RFC5747 | RFC6037 | RFC6052 | RFC6368 | RFC6396 | RFC6397 | RFC6513
RFC6514 | RFC6608 | RFC6666 | RFC6793 | RFC6938 | RFC7117 | RFC7267 | RFC7311 | RFC7313
RFC7432 | RFC7447 | RFC7534 | RFC7752 | RFC7911 | RFC8050 | RFC8092 | RFC8093 | RFC8190
RFC8205 | RFC8215 | RFC8277 | RFC8538 | RFC8654 | RFC8669 | RFC8810 | RFC8950 | RFC8955
RFC9003 | RFC9012 | RFC9015 | RFC9026 | RFC9072 | RFC9234 | RFC9384 | RFC9552



ARPA NET, APRIL 1971

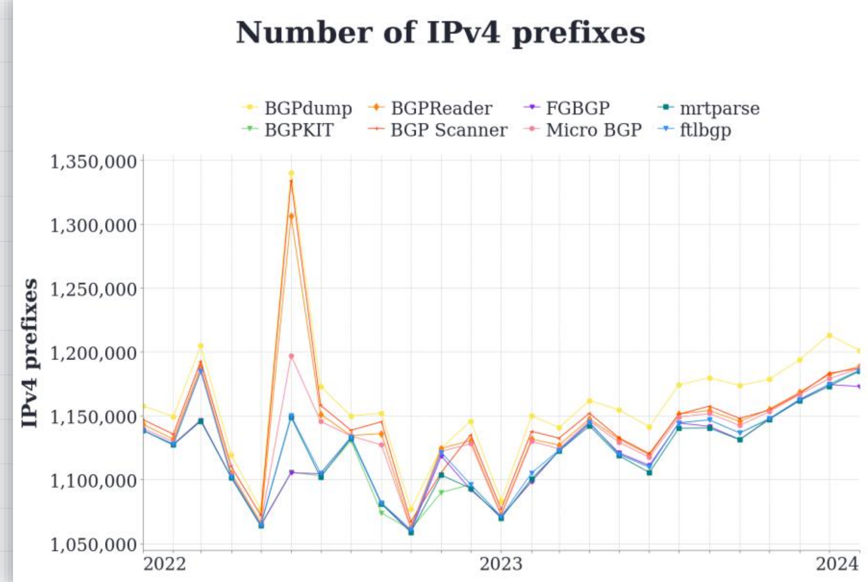
Address space characteristics

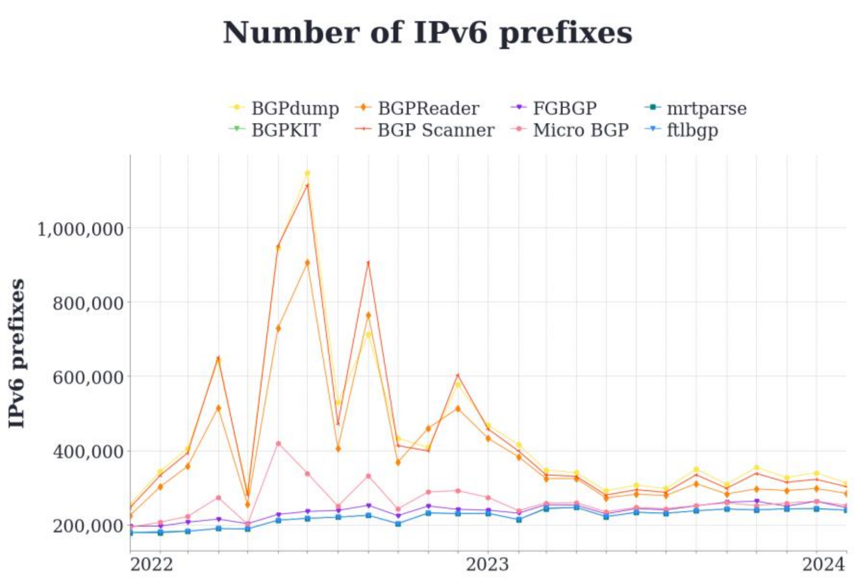
HOW BIG IS THE INTERNET?

How big is the Internet? (I)

Routed address space

Are you able to define an acceptable margin of error?

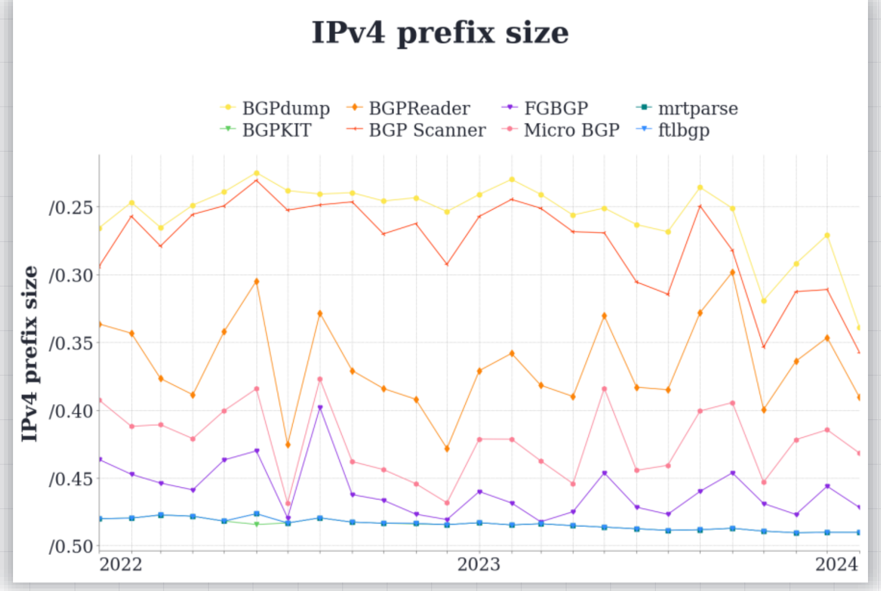


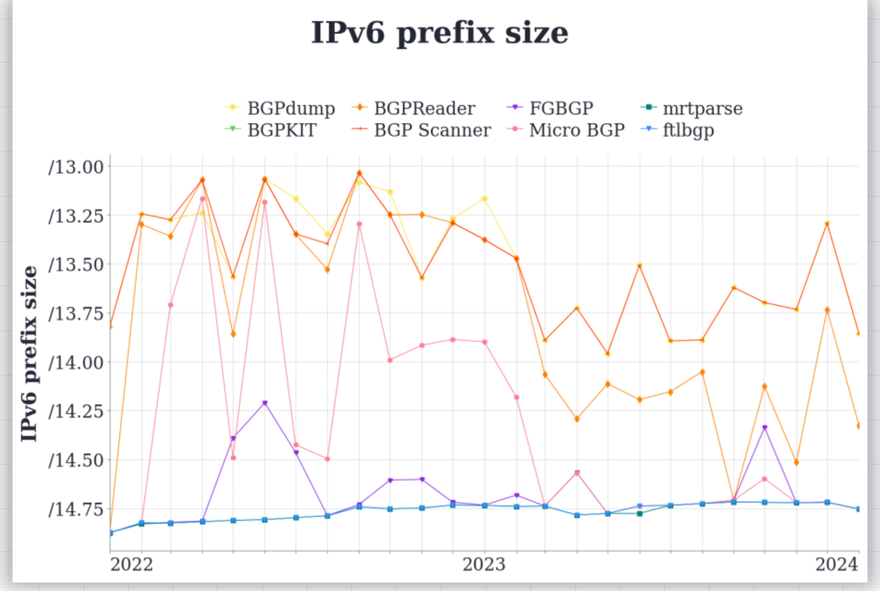


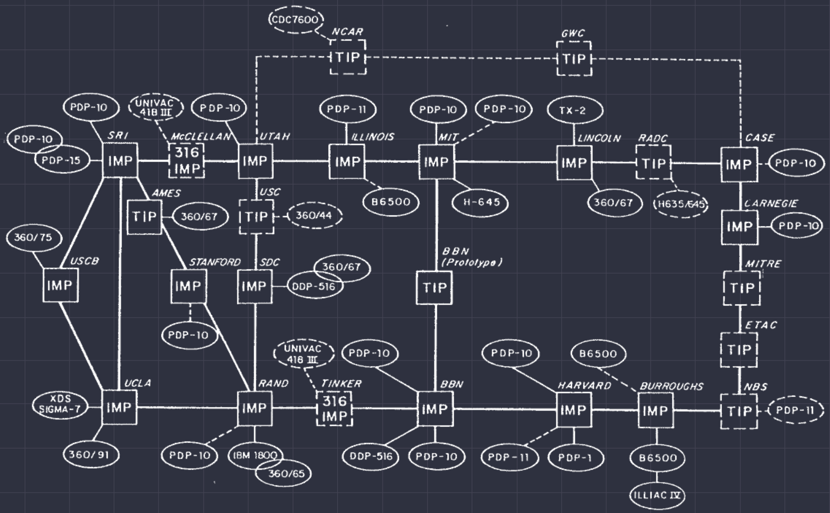
How big is the Internet? (II)

Aggregated size of the routed address space

Does it matter if the Internet is 20% bigger or smaller? What about 100%?







ARPA NET, AUGUST 1971

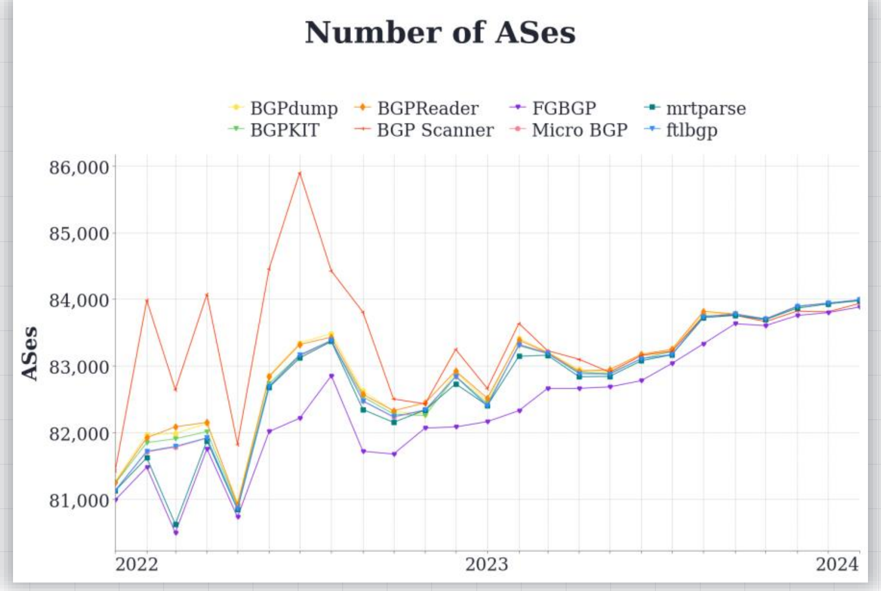
Topological characteristics

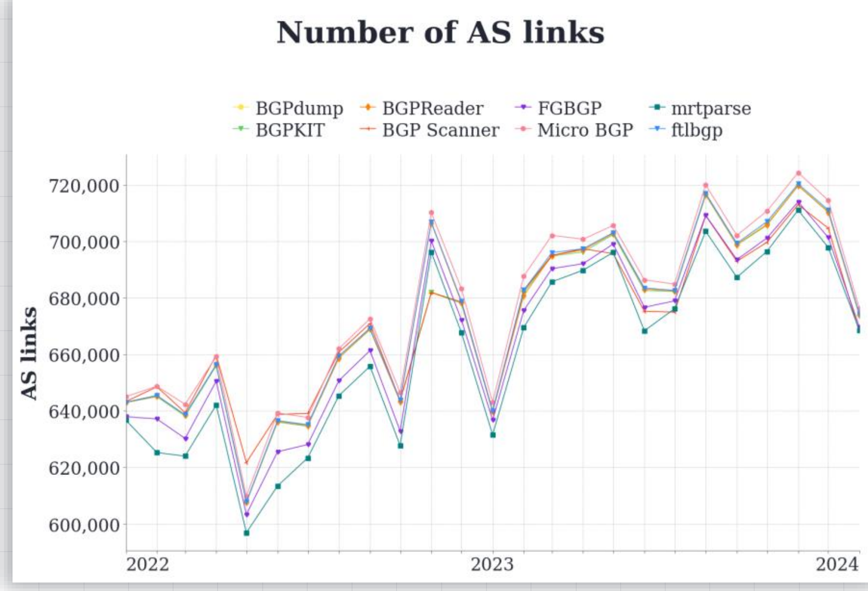
HOW ARE ASES INTERCONNECTED?

How are ASes interconnected?

Countless analyses on the BGP AS_PATH attribute

Graph metrics, AS rankings, customer cones, peering/transit relations, routing policies, RPKI/ASPA

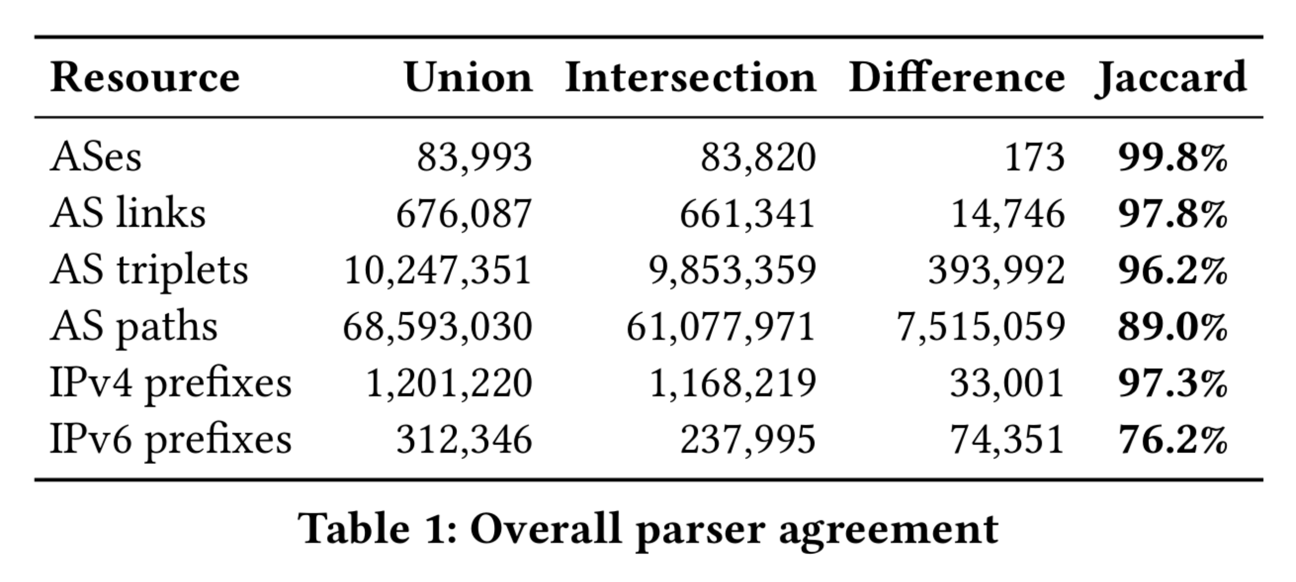




MRT parsing: current situation is best (!) so far

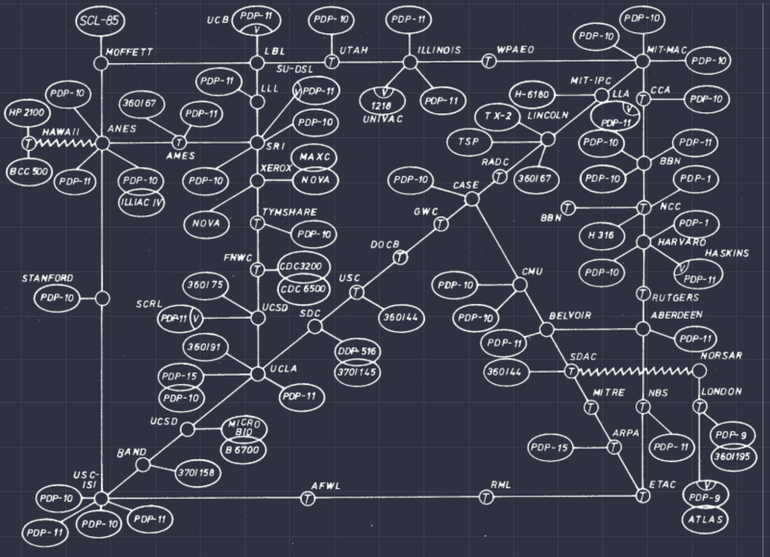
»Today, we commemorate those 173 poor souls that were lost in translation«

So is it really that hard to agree on the information content of a few BGP updates?



Resource	Union	Intersection	Difference	Jaccard
ASes	83,993	83,820	173	**99.8%**
AS links	676,087	661,341	14,746	**97.8%**
AS triplets	10,247,351	9,853,359	393,992	**96.2%**
AS paths	68,593,030	61,077,971	7,515,059	**89.0%**
IPv4 prefixes	1,201,220	1,168,219	33,001	**97.3%**
IPv6 prefixes	312,346	237,995	74,351	**76.2%**

Table 1: Overall parser agreement



ARPA NETwork, June 1974.

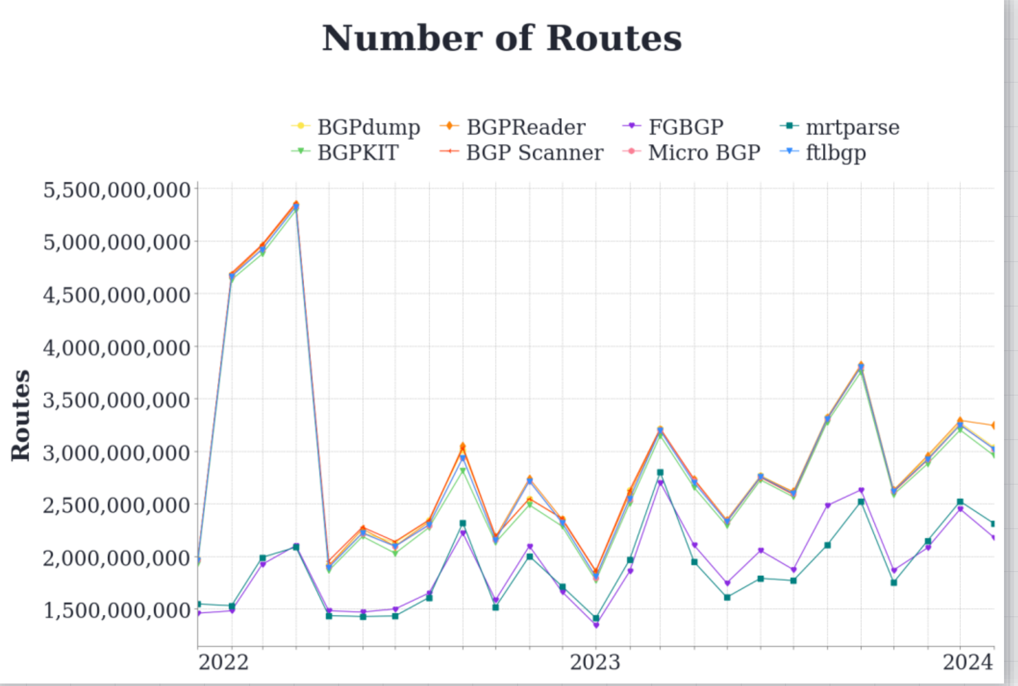
A matter of **taste**?

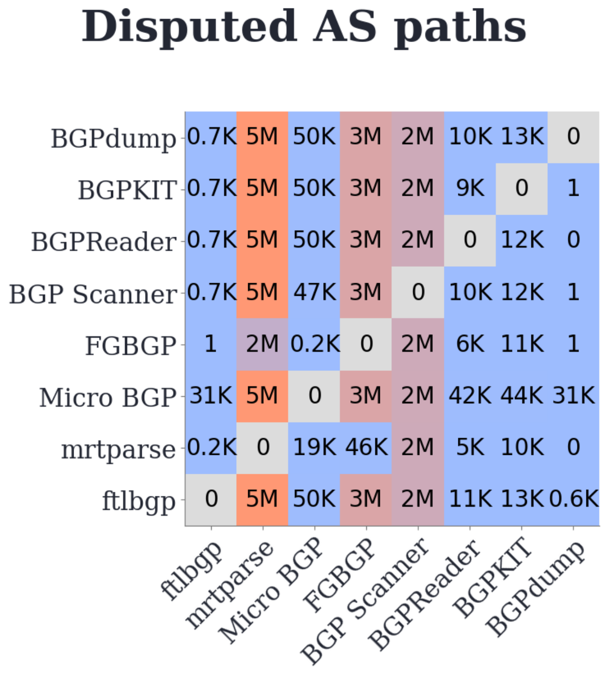
MRT PARSING STRATEGIES

MRT parsing strategies (I)

We observe three different types of parsers

There are clearly **distinguishable sets of results**, but we also looked at the source code



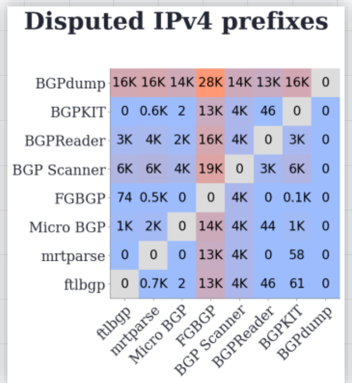

Disputed AS paths

	ftlbgp	mrtparse	Micro BGP	FGBGP	BGP Scanner	BGPReader	BGPKIT	BGPdump
BGPdump	0.7K	5M	50K	3M	2M	10K	13K	0
BGPKIT	0.7K	5M	50K	3M	2M	9K	0	1
BGPReader	0.7K	5M	50K	3M	2M	0	12K	0
BGP Scanner	0.7K	5M	47K	3M	0	10K	12K	1
FGBGP	1	2M	0.2K	0	2M	6K	11K	1
Micro BGP	31K	5M	0	3M	2M	42K	44K	31K
mrtparse	0.2K	0	19K	46K	2M	5K	10K	0
ftlbgp	0	5M	50K	3M	2M	11K	13K	0.6K

MRT parsing strategies (II)

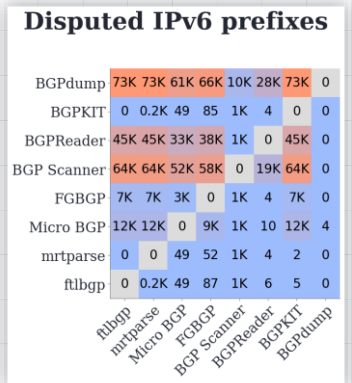
We observe three different types of parsers

- A – Parsing exactly as standardized, even if it means crashing on faulty input
- B – Recover from errors where possible, filter out prohibitive anomalies (e.g. IPv4 >/32)
- C – Try to reconstruct original information as best as possible, even heuristically if need be



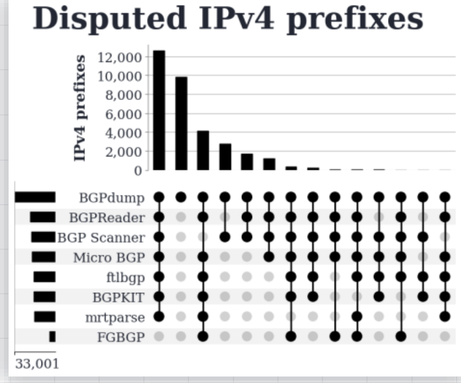
Disputed IPv4 prefixes

	ftlbgp	mrtparse	Micro BGP	FGBGP	BGP Scanner	BGPReader	BGPKIT	BGPdump
BGPdump	16K	16K	14K	28K	14K	13K	16K	0
BGPKIT	0	0.6K	2	13K	4K	46	0	0
BGPReader	3K	4K	2K	16K	4K	0	3K	0
BGP Scanner	6K	6K	4K	19K	0	3K	6K	0
FGBGP	74	0.5K	0	0	4K	0	0.1K	0
Micro BGP	1K	2K	0	14K	4K	44	1K	0
mrtparse	0	0	0	13K	4K	0	58	0
ftlbgp	0	0.7K	2	13K	4K	46	61	0

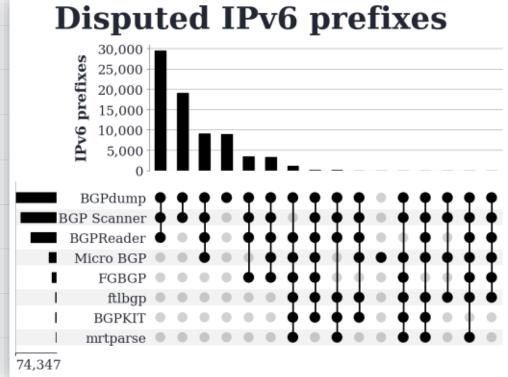


Disputed IPv6 prefixes

	ftlbgp	mrtparse	Micro BGP	FGBGP	BGP Scanner	BGPReader	BGPKIT	BGPdump
BGPdump	73K	73K	61K	66K	10K	28K	73K	0
BGPKIT	0	0.2K	49	85	1K	4	0	0
BGPReader	45K	45K	33K	38K	1K	0	45K	0
BGP Scanner	64K	64K	52K	58K	0	19K	64K	0
FGBGP	7K	7K	3K	0	1K	4	7K	0
Micro BGP	12K	12K	0	9K	1K	10	12K	4
mrtparse	0	0	49	52	1K	4	2	0
ftlbgp	0	0.2K	49	87	1K	6	5	0



Disputed IPv4 prefixes

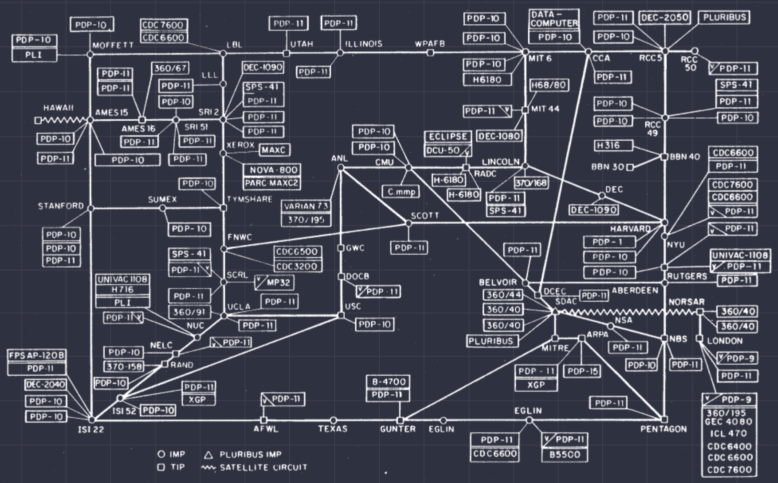


Disputed IPv6 prefixes

Which ~~strategy~~ use case is »the best«?

None – if you want to see if someone is announcing a /129, you should **not** filter it out

ARPANET LOGICAL MAP, MARCH 1977



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

Let's change topic

DO WE LIKE MRT?

Let's rephrase: do we like BGP?

Example: RFC7911 (BGP-ADDPATH)

»The **only explicit indication** that the encoding described in Section 3 is in use in a particular BGP session **is the exchange of Capabilities** described in Section 4. (…) However, if, for example, a packet analyzer is used on the wire to examine an active BGP session, it **may not be able to properly decode** the BGP UPDATES because it **lacks prior knowledge** of the exchanged Capabilities.«

MRT has no concept of peer capabilities

- There is a peer index table for table dump v2 MRT entries, which does not store capabilities
- BGP UPDATE streams are written into short-interval MRT files (usually 1-15 minutes)
- Peer capabilities are sometimes encoded in MRT entry types

Without proper peer knowledge, (strict) parsing will fail

Do we like this switching between BGP and MRT?

RFC6396 (MRT) and RFC8050 (MRT-ADDPATH)

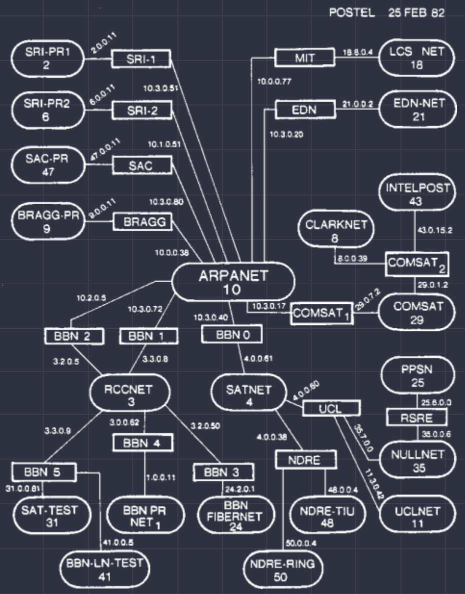
- BGP4MP_MESSAGE
- BGP4MP_MESSAGE_AS4
- BGP4MP_MESSAGE_ADDPATH
- BGP4MP_MESSAGE_AS4_ADDPATH

Imagine some new feature (MRT-FEAT2025)

- BGP4MP_MESSAGE
- BGP4MP_MESSAGE_AS4
- BGP4MP_MESSAGE_ADDPATH
- BGP4MP_MESSAGE_AS4_ADDPATH
- BGP4MP_MESSAGE_**FEAT2025**
- BGP4MP_MESSAGE_AS4_**FEAT2025**
- BGP4MP_MESSAGE_ADDPATH_**FEAT2025**
- BGP4MP_MESSAGE_AS4_ADDPATH_**FEAT2025**

2^n

REALLY?



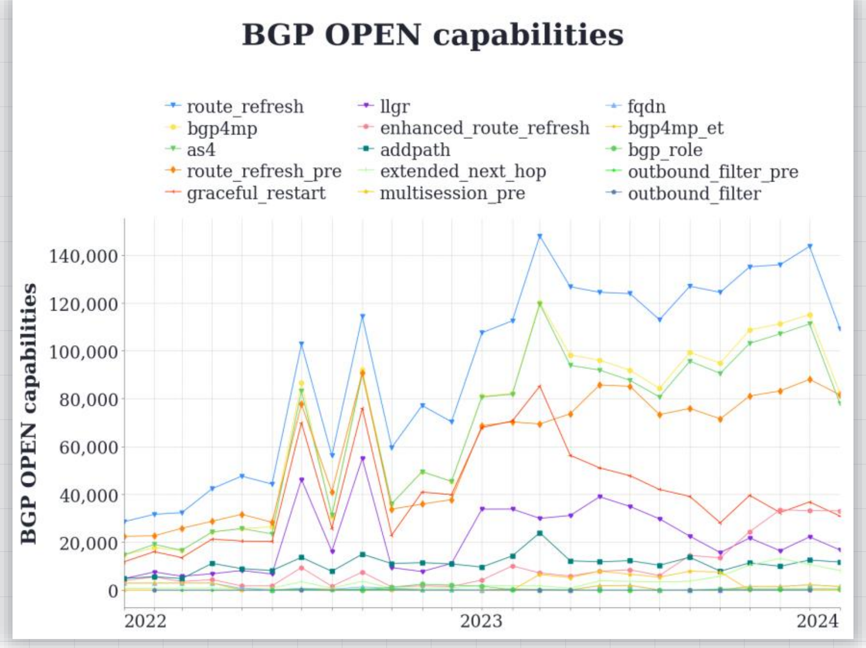
Case Study [RFC6793]

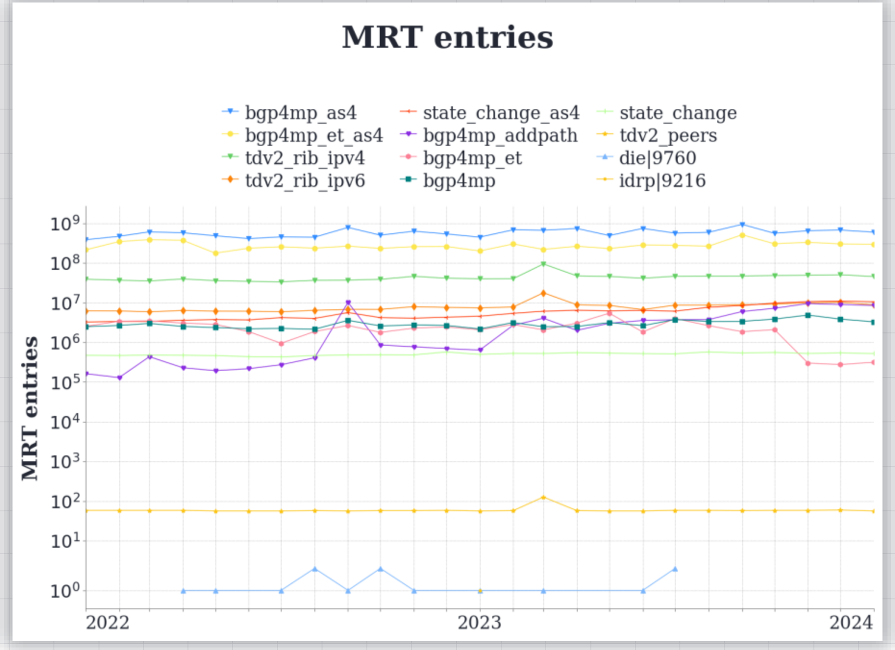
FOUR-OCTET AS NUMBERS

Case Study: Four-octet AS numbers (I)

We still observe BGP negotiations without AS4 support (<5.2%)

- A total of 0.1% of all BGP4MP MRT entries are typed with the wrong ASN octect length

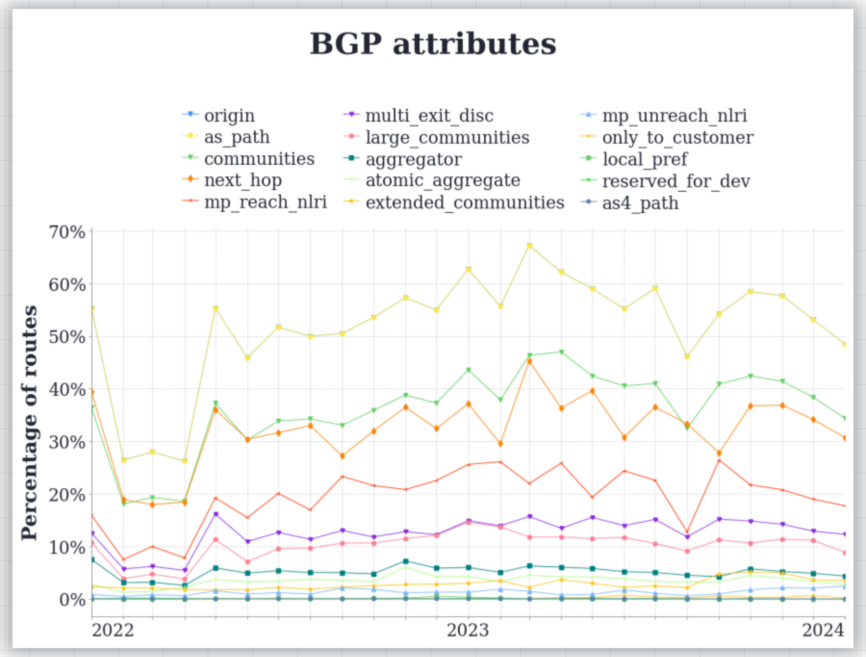


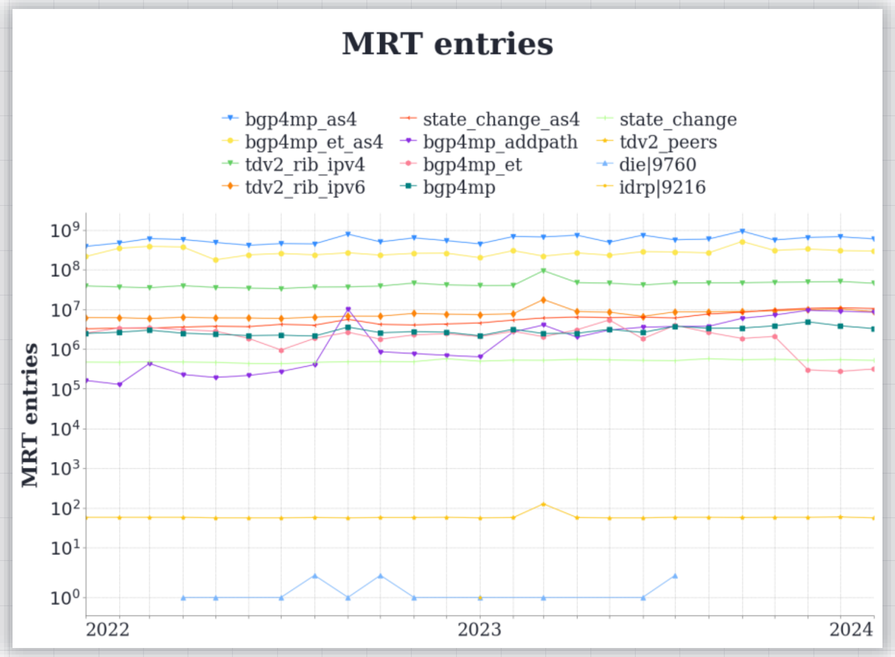


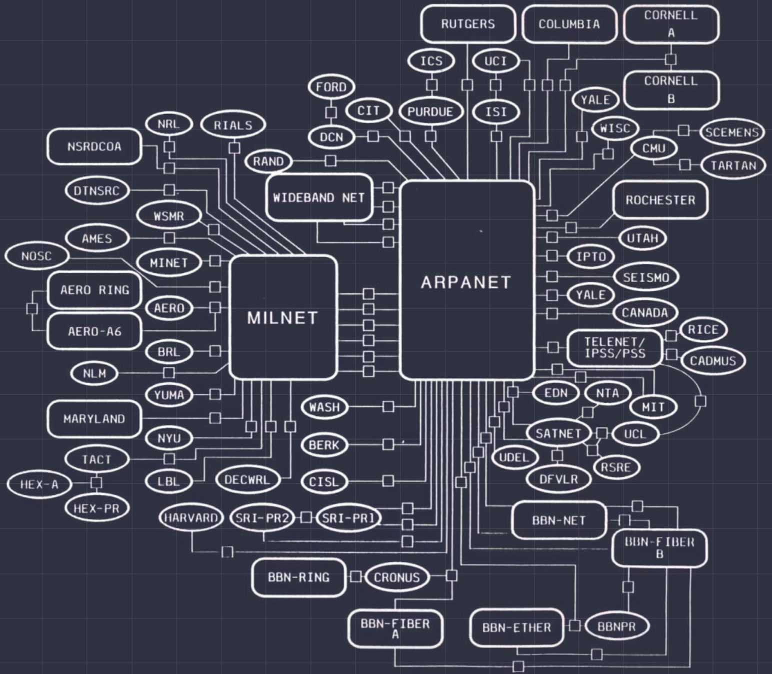
Case Study: Four-octet AS numbers (II)

We still see the transitional AS4_PATH attribute (on 0.06% of routes)

- However, most unknown-AS problems (c.f. **AS2936451170**) arise from ill-typed MRT entries







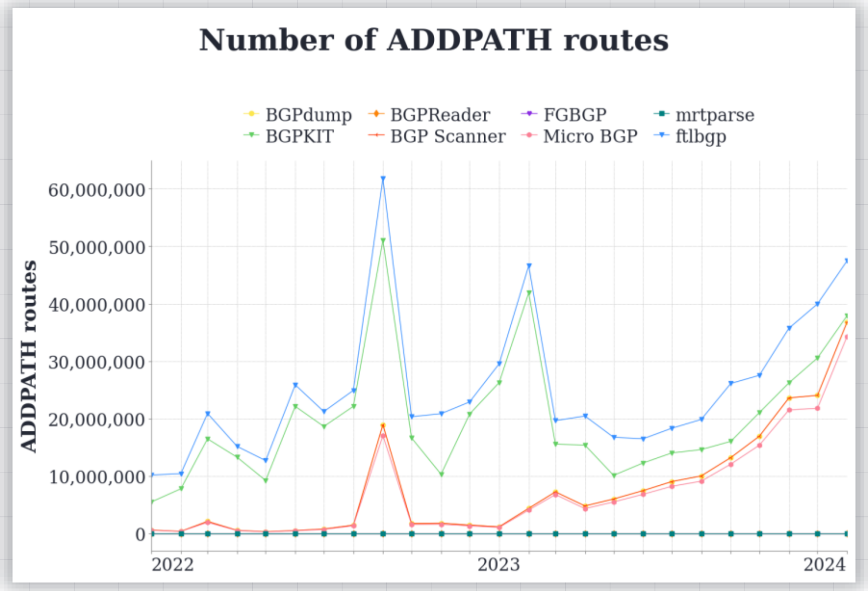
Case Study [RFC7911]

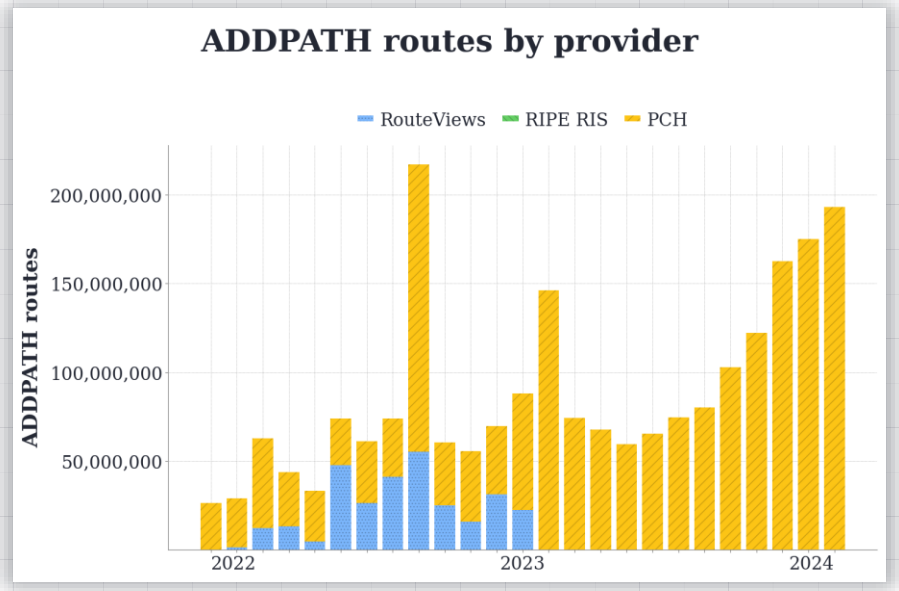
MULTIPLE PATHS WITH BGP-ADDPATH

Case Study: Multiple paths in BGP-ADDPATH (I)

ADDPATH has been tested by RouteViews and is gaining traction at PCH

Some providers possibly still fear an explosion of data (which is only partially true)

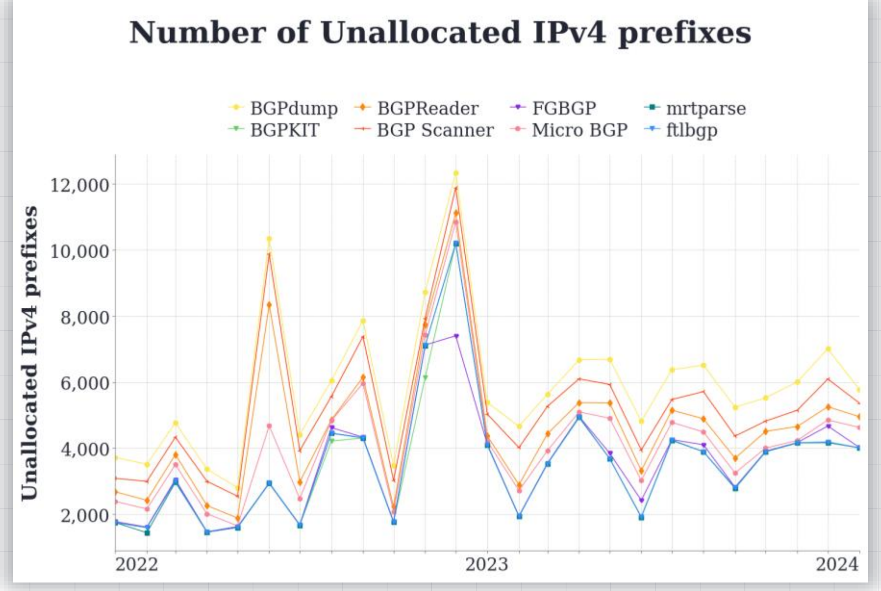


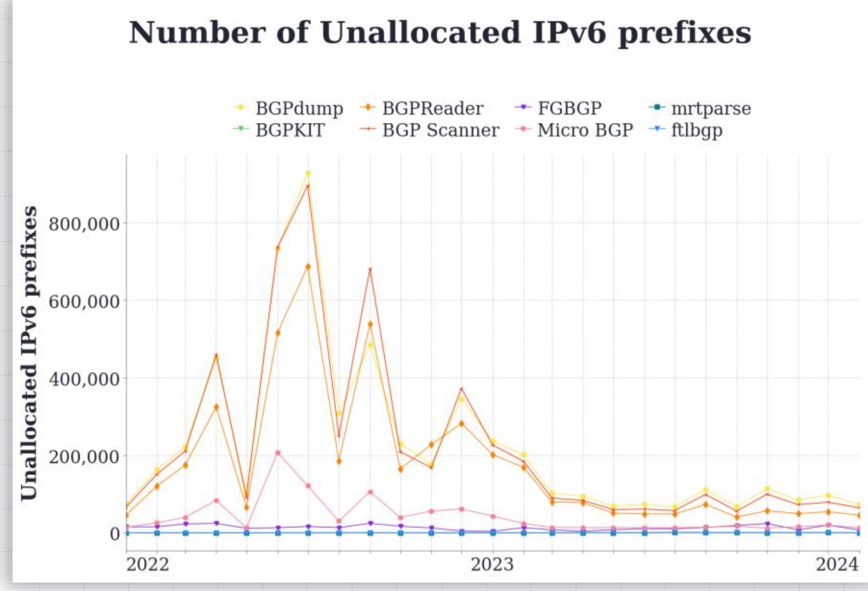


Case Study: Multiple paths in BGP-ADDPATH (II)

Enabling new BGP features can lead to data loss and/or corruption

It is necessary that both **exporter and parser** add support for new capabilities

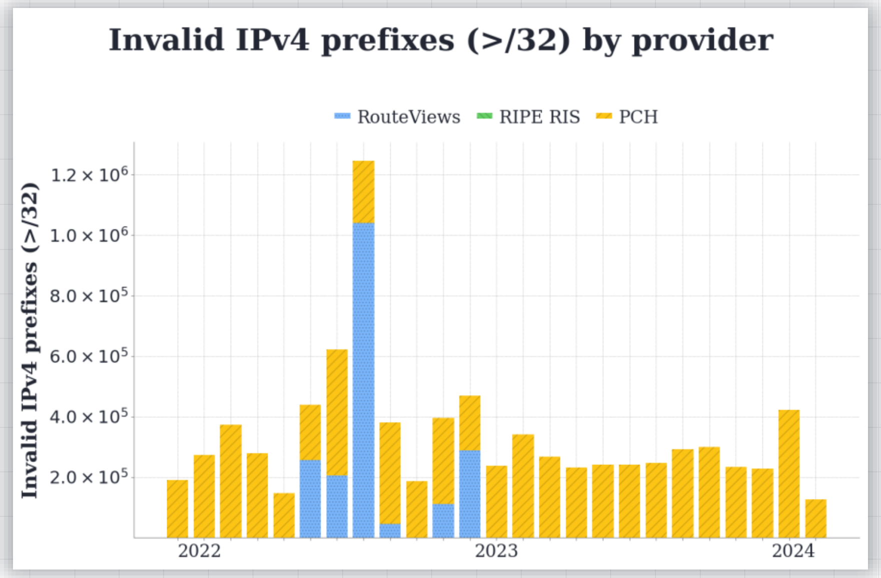


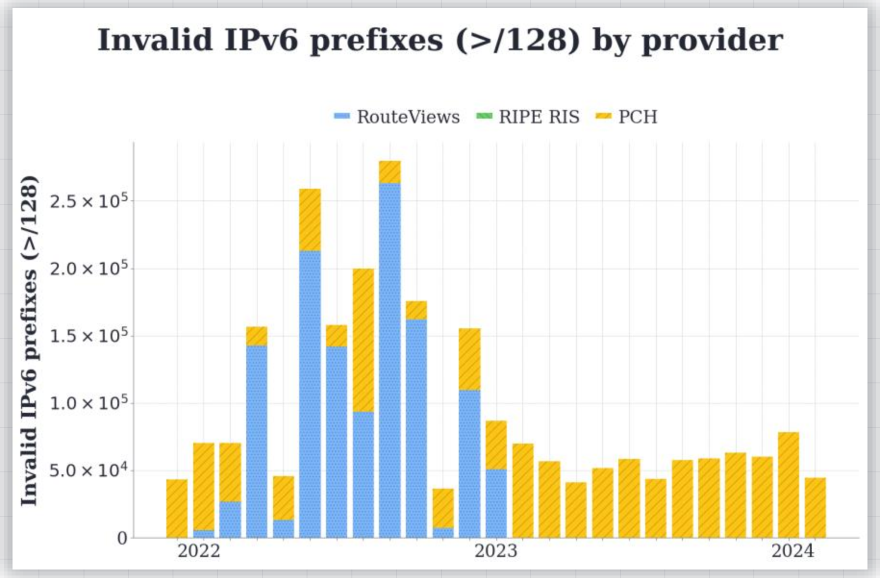


Case Study: Multiple paths in BGP-ADDPATH (III)

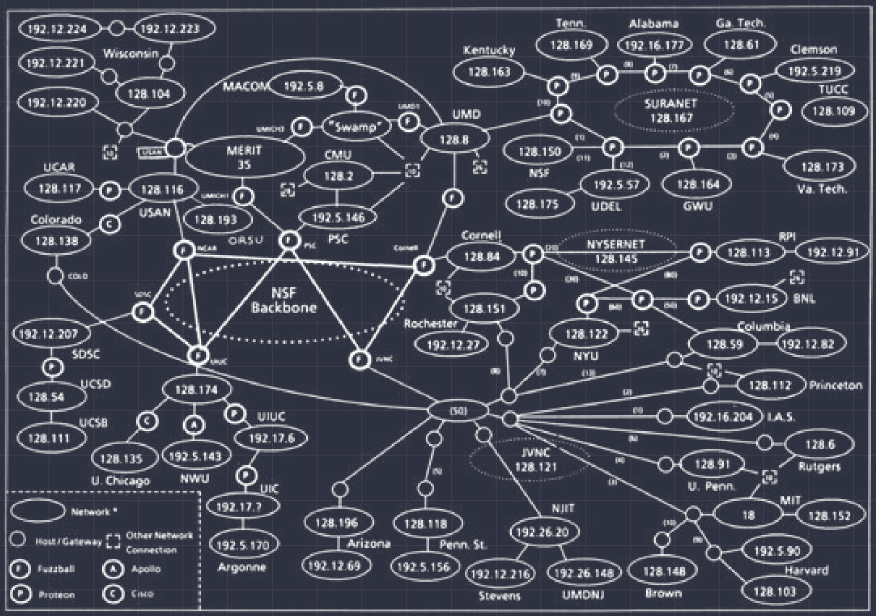
Enabling new BGP features can lead to data loss and/or corruption

It is necessary that both **exporter and parser** add support for new capabilities





NSFNet Physical Connectivity -- April 87



* For some networks internal structure (e.g. subnets) is suppressed.

ISI 4/17/87

Lessons learned

MRT != MRT

We »had« to implement our own MRT parser (I)

Core feature requests

- Support for **all** MRT entries/BGP messages and attributes
- Customizable in terms of **selecting record/attribute types**
- **Raw values** and human-readable output (integers vs. strings)
- Native processing of BGP records (+JSON/CSV serialization)

Nice-to-have features

- Transparent support for looking glass text formats (**show bgp** output)
- Rapid prototyping and high-performance modes (namedtuple vs. tuple)
- Built-in statistics and **flexible error handling** (no unexpected aborts)

ftlbgp

- Implemented in Python3 / PyPy3 (fast)
- Zero-Copy operations on all data items (really fast)
- Released as **open-source software** today ☺

We »had« to implement our own MRT parser (II)

```
from ftlbgp import BgpParser

with BgpParser(named_records=True, human_readable=True, serialize=False) as parse:

    for record in parse("rib.20240101.0000.bz2"):
        print(record)

BgpRouteRecord(type=, source=, sequence=, timestamp=, peer_protocol=, peer_bgp_id=, peer_as=,
peer_ip=, nexthop_protocol=, nexthop_ip=, prefix_protocol=, prefix=, path_id=, aspath=, origin=,
communities=, large_communities=, extended_communities=, multi_exit_disc=, atomic_aggregate=,
aggregator_protocol=, aggregator_as=, aggregator_ip=, only_to_customer=, originator_id=, cluster_list=,
local_pref=, attr_set=, as_pathlimit=, aigp=, attrs_unknown=)

BgpPeerTableRecord(…)         BgpStateChangeRecord(…)    BgpKeepAliveRecord(…)
BgpRouteRefreshRecord(…)      BgpNotificationRecord(…)   BgpOpenRecord(…)
BgpStatsRecord(…)             BgpErrorRecord(…)
```

Summary and future work

Lessons learned

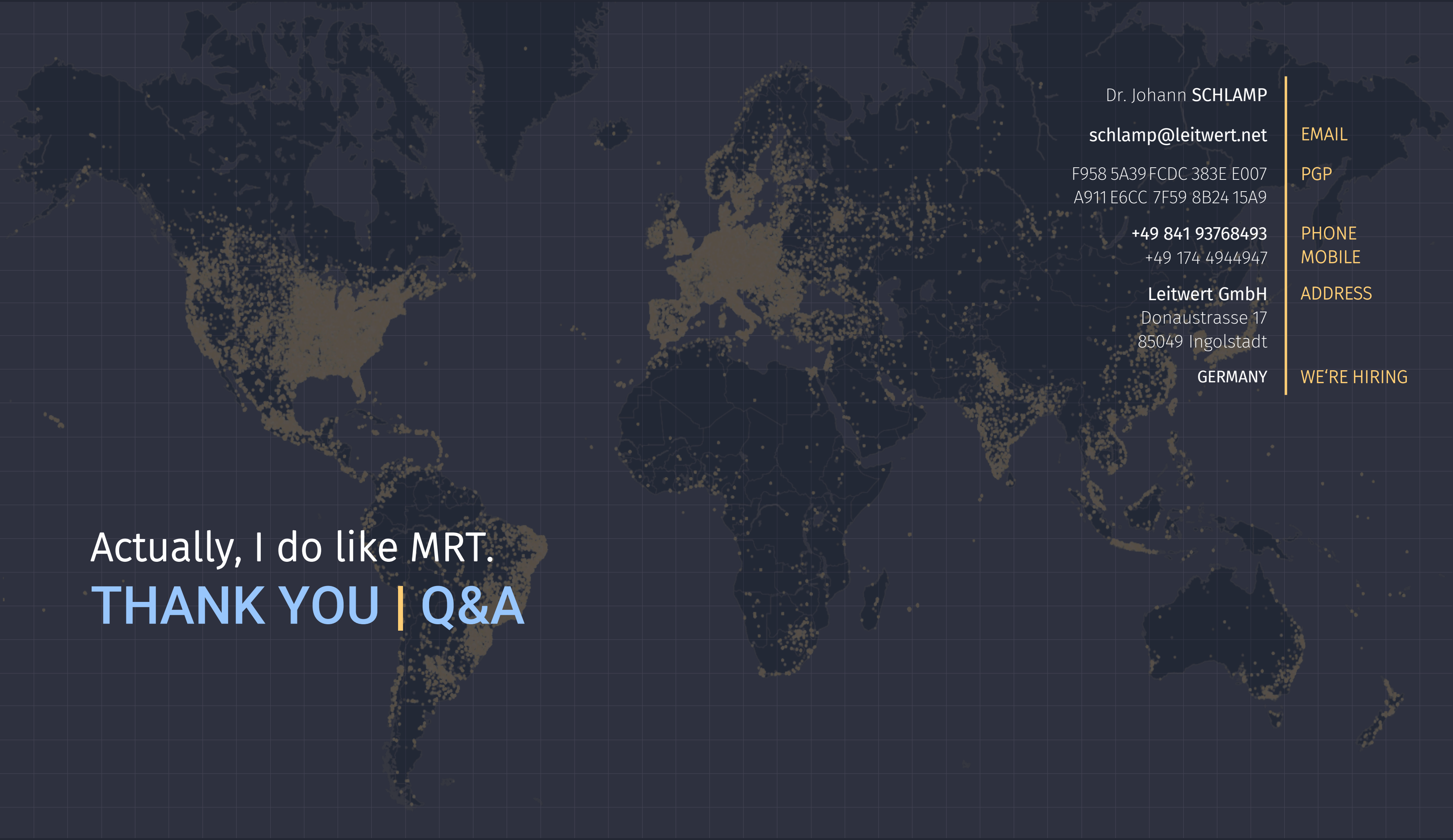
- Raw BGP data requires interpretation and interpolation – we have **dialects** and **artifacts**
- Knowledge of **peer capabilities** would be paramount – but there is no way for direct access
- Adding new features to the BGP/MRT standard can lead to **data corruption** (c.f. ADDPATH)
- The situation improves with better exporters – but **historic analyses remain problematic**
- Crafting BGP messages with certain attributes may **conceal routes** or even **crash parsers**

Work in progress

- We're working on a **paper submission** – look out for a preprint soon
- We're looking for collaborators to improve MRT (adding **peer capabilities** and **RPKI features**)

Try our parser (MIT licensed)

- https://github.com/leitwert-net/ftlbgp
- python3 –m pip install ftlbgp



Dr. Johann **SCHLAMP**

schlamp@leitwert.net EMAIL

F958 5A39 FCDC 383E E007
A911 E6CC 7F59 8B24 15A9 PGP

+49 841 93768493 PHONE
+49 174 4944947 MOBILE

Leitwert GmbH ADDRESS
Donaustrasse 17
85049 Ingolstadt

GERMANY

WE'RE HIRING

Actually, I do like MRT.

THANK YOU | Q&A