PeeringDB Accuracy

Is blind faith reasonable?

Job Snijders

job.snijders@atrato.com

NANOG58

Who am I?

Job Snijders

Network Architect @
AS 5580 - Atrato IP Networks

Founder of NLNOG RING

Proud member of PeeringDB promo team

Twitter: @JobSnijders

Hobbies: IP Routing, LISP, MPLS, IPv6,

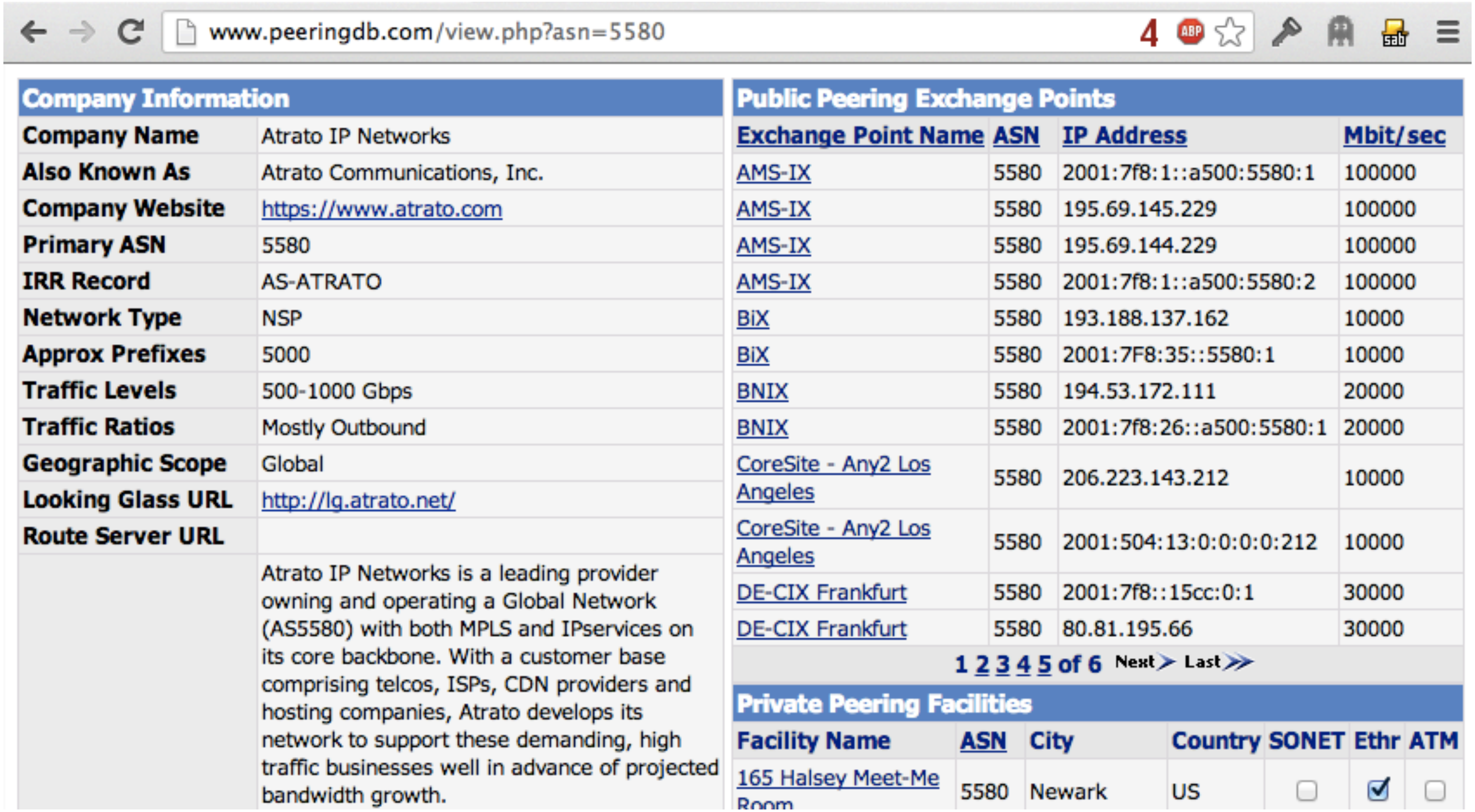
Shoe size: 45/EU

What is PeeringDB? (first iteration of PDB in early sixties)

What is PeeringDB?

- Almost a third of the criticial DFZ ASNs register (partly) their interconnection options in PeeringDB
- True game changer in Peering/IXP market
- Run by volunteers
- Data is 100% user submitted (think wikipedia)

Example peering record



www.peeringdb.com/view.php?asn=5580

Company Information	
Company Name	Atrato IP Networks
Also Known As	Atrato Communications, Inc.
Company Website	https://www.atrato.com
Primary ASN	5580
IRR Record	AS-ATRATO
Network Type	NSP
Approx Prefixes	5000
Traffic Levels	500-1000 Gbps
Traffic Ratios	Mostly Outbound
Geographic Scope	Global
Looking Glass URL	http://lg.atrato.net/
Route Server URL	
	Atrato IP Networks is a leading provider owning and operating a Global Network (AS5580) with both MPLS and IPservices on its core backbone. With a customer base comprising telcos, ISPs, CDN providers and hosting companies, Atrato develops its network to support these demanding, high traffic businesses well in advance of projected bandwidth growth.

Public Peering Exchange Points

Exchange Point Name	ASN	IP Address	Mbit/sec
AMS-IX	5580	2001:7f8:1::a500:5580:1	100000
AMS-IX	5580	195.69.145.229	100000
AMS-IX	5580	195.69.144.229	100000
AMS-IX	5580	2001:7f8:1::a500:5580:2	100000
BiX	5580	193.188.137.162	10000
BiX	5580	2001:7F8:35::5580:1	10000
BNIX	5580	194.53.172.111	20000
BNIX	5580	2001:7f8:26::a500:5580:1	20000
CoreSite - Any2 Los Angeles	5580	206.223.143.212	10000
CoreSite - Any2 Los Angeles	5580	2001:504:13:0:0:0:0:212	10000
DE-CIX Frankfurt	5580	2001:7f8::15cc:0:1	30000
DE-CIX Frankfurt	5580	80.81.195.66	30000

1 2 3 4 5 of 6 Next > Last >>

Private Peering Facilities

Facility Name	ASN	City	Country	SONET	Ethr	ATM
165 Halsey Meet-Me Room	5580	Newark	US	□	☑	□

Current interfaces to PDB

Today

- MySQL (peeringdb.com:3306)
- Nightly SQL Dump (use this for automation!)
- Web interface

Future

- RESTful APIs
- Modern stuff

Problem statement

Is PeeringDB worthy of my trust?

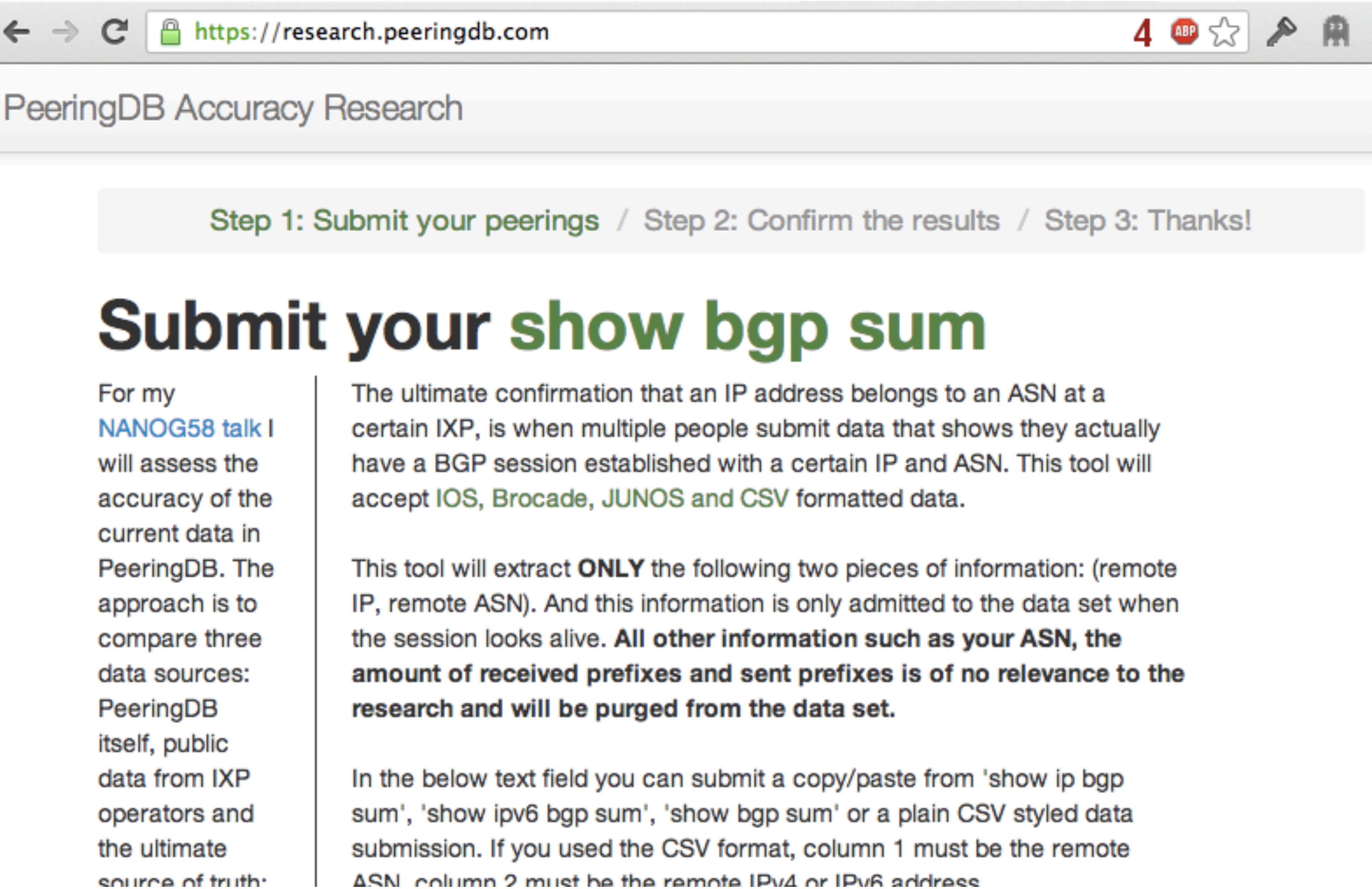
Can I use the data in automated tooling for peering?

(one button peering)

Methodology

- 2 data sources are used as input for validating the dataset:
  - – PeeringDB SQL dump
  - – Actual 'show bgp sum' from many networks

- Throw all data into database, compare



PeeringDB Accuracy Research

Step 1: Submit your peerings / Step 2: Confirm the results / Step 3: Thanks!

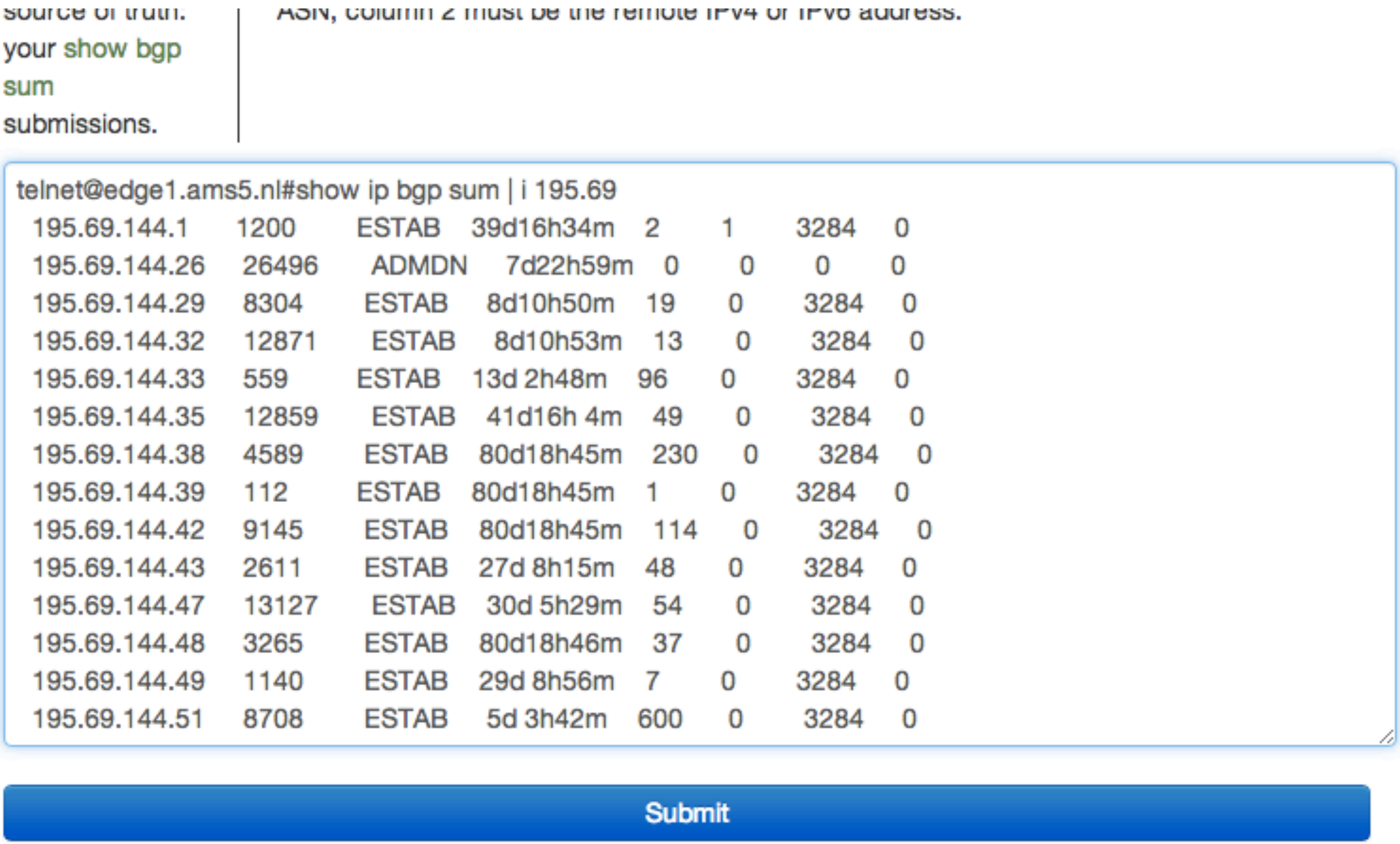
Submit your show bgp sum

For my NANOG58 talk I will assess the accuracy of the current data in PeeringDB. The approach is to compare three data sources: PeeringDB itself, public data from IXP operators and the ultimate source of truth:

The ultimate confirmation that an IP address belongs to an ASN at a certain IXP, is when multiple people submit data that shows they actually have a BGP session established with a certain IP and ASN. This tool will accept IOS, Brocade, JUNOS and CSV formatted data.

This tool will extract **ONLY** the following two pieces of information: (remote IP, remote ASN). And this information is only admitted to the data set when the session looks alive. **All other information such as your ASN, the amount of received prefixes and sent prefixes is of no relevance to the research and will be purged from the data set.**

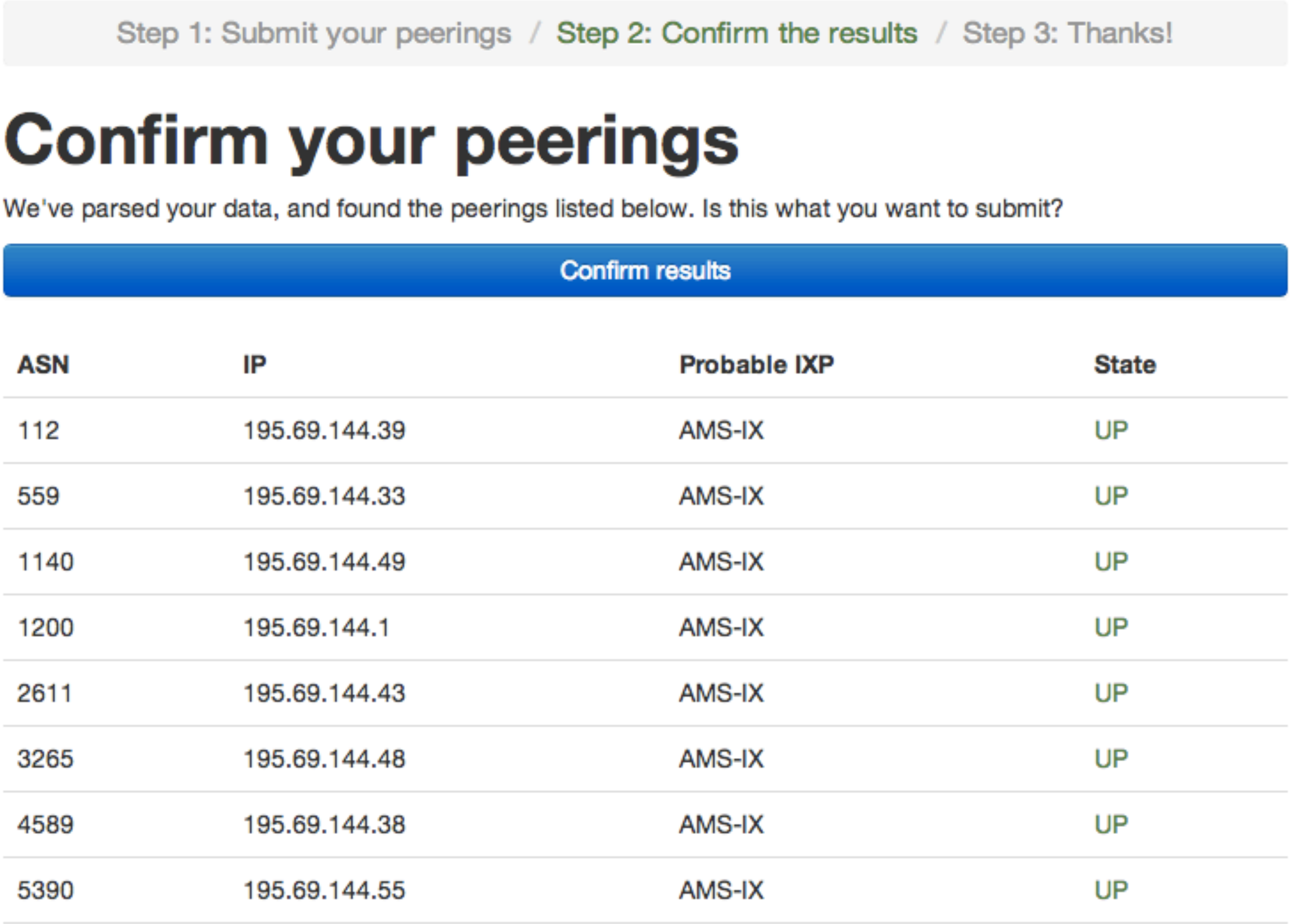
In the below text field you can submit a copy/paste from 'show ip bgp sum', 'show ipv6 bgp sum', 'show bgp sum' or a plain CSV styled data submission. If you used the CSV format, column 1 must be the remote ASN, column 2 must be the remote IPv4 or IPv6 address.



source of truth.
your show bgp sum submissions.

ASN, column 2 must be the remote IPv4 or IPv6 address.

```
telnet@edge1.ams5.nl#show ip bgp sum | i 195.69
  195.69.144.1     1200      ESTAB   39d16h34m   2      1      3284    0
  195.69.144.26    26496      ADMDN    7d22h59m   0      0      0      0
  195.69.144.29    8304       ESTAB    8d10h50m   19     0      3284    0
  195.69.144.32    12871      ESTAB    8d10h53m   13     0      3284    0
  195.69.144.33    559       ESTAB   13d 2h48m   96     0      3284    0
  195.69.144.35    12859      ESTAB   41d16h 4m   49     0      3284    0
  195.69.144.38    4589       ESTAB   80d18h45m   230    0      3284    0
  195.69.144.39    112       ESTAB   80d18h45m   1      0      3284    0
  195.69.144.42    9145       ESTAB   80d18h45m   114    0      3284    0
  195.69.144.43    2611       ESTAB   27d 8h15m   48     0      3284    0
  195.69.144.47    13127      ESTAB   30d 5h29m   54     0      3284    0
  195.69.144.48    3265       ESTAB   80d18h46m   37     0      3284    0
  195.69.144.49    1140       ESTAB   29d 8h56m   7      0      3284    0
  195.69.144.51    8708       ESTAB    5d 3h42m   600    0      3284    0
```


Step 1: Submit your peerings / Step 2: Confirm the results / Step 3: Thanks!
Confirm results

Confirm your peerings

We've parsed your data, and found the peerings listed below. Is this what you want to submit?

ASN	IP	Probable IXP	State
112	195.69.144.39	AMS-IX	UP
559	195.69.144.33	AMS-IX	UP
1140	195.69.144.49	AMS-IX	UP
1200	195.69.144.1	AMS-IX	UP
2611	195.69.144.43	AMS-IX	UP
3265	195.69.144.48	AMS-IX	UP
4589	195.69.144.38	AMS-IX	UP
5390	195.69.144.55	AMS-IX	UP

Networks which contributed data

Atrato, PCCW, Leaseweb (and many others!)

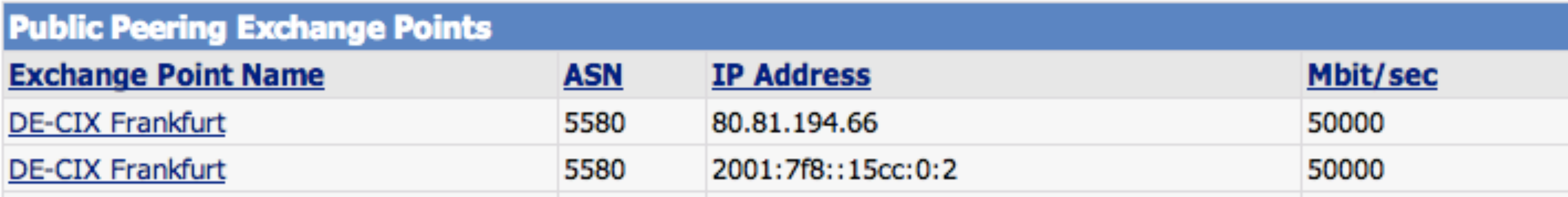
Statistics

- 33500 unique BGP sightings
- 256 individual submissions
- Over period of 7 days

Source #1: PDB

- 4738 networks in PeeringDB (10% of DFZ)
- 15142 connections are described in PDB
  - IP Address field currently is VARCHAR(128)

Example:



Public Peering Exchange Points

Exchange Point Name	ASN	IP Address	Mbit/sec
DE-CIX Frankfurt	5580	80.81.194.66	50000
DE-CIX Frankfurt	5580	2001:7f8::15cc:0:2	50000

Worst IP address entries

91.232.229.x

2a01.6e00.0010::1

2001:7F8:20:101::(245/247):61/64

2001:7f8:20:101(2)::244(6):28

“Multiple”

“Soon”

“Coming Aug 1st, 2012”

Parsing / Scoring

1559 entries don’t parse out of the box, let’s try to save a few:

```
def scrub(ip_address):
    ip = ip_address.strip()
    ip, s, t = ip.partition('/')
    try:
        ip = IPAddress(ip)
        return ip
    except:
        return None
```

Voila! 877 entries are recovered ☺

Usability of PDB

96.5%

Of user submitted data is parsable

(only 682 entries kinda suck)

Source #2 – Ground truth

When networks report seeing an established sessions with a certain (IP, ASN) tuple, we'll take it as ground truth.

Statistics!

- 33500 sightings (28% IPv6 – 72% IPv4)
- 11300 unique (IP, ASN) tuples
- 8937 unique tuples recognizable as IXP IP

Source #2 – Ground truth - Results

- 6721 out of 8937 sightings match with PDB
  - 75% of sightings have corresponding PDB entries
  - 44% of entire PDB data-set has been verified now

- 64 IP addresses conflict between truth and parsable peering records

What are those 64 conflicts?

- 57 IPv4 versus 7 IPv6 entries.
  - (are we making less mistakes with IPv6? ☺)

- 60% - old stale data - IP recycling @ IXP
- 26% - migrations (mergers / acquisitions)
- 14% - False information in PDB (usually typos)
- 8% - No idea, couldn't figure it out

The end score!

- 75% overlap between peeringdb and sightings
- 44% of peeringdb has now been verified
- Only 64 conflicts…

99% of parsable entries are correct.

ddp
ddp
ddp

Conclusion

- Yes, you can trust peeringDB if you accept fault margin ~ 1%
- Scrubbing the entries is worth the effort
  - 2 lines of python saved 60% of unparsable records
- Keep it up!

How can you improve your part?

- These common guidelines are used for entering data in PeeringDB (although not enforced)
  - 1 IPv4 or IPv6 address per entry
    - (yes, this messes up the total capacity, but who cares)
  - IP addresses without subnet length (not add /22)
  - Etc ….

Automation efforts

- When you start automating your peering, take these things into consideration:
  - Scrubbing the data is worth the effort!
  - PeeringDB 2.0 should include a 'freshness' factor for records to work around stale data
  - Should we create a feedback loop with data-sets like these and sanitize the peeringdb dataset?

Questions?

Ps. I will delete the peeringdb census data