



RIPE NCC
RIPE NETWORK COORDINATION CENTRE

RIPE Atlas Tutorial

Viktor Naumov
Michela Galante

ENOG 11 | 7 June 2016

Goals



- Learn how to:
 - Benefit from using RIPE Atlas measurements for network monitoring and troubleshooting
 - Use API calls to create measurements
 - Use Command Line Interface Tools
 - Real time network monitoring with RIPE Atlas streaming
- Opportunity for hands-on practice
- Get your questions answered

Overview



- Introduction to RIPE Atlas
- What can you get from RIPE Atlas as a visitor
- Creating a measurement and exercise
- Command Line Interface (CLI) Toolset
- Real-time performance monitoring and exercise
- Additional Topics
 - IXP Country Jedi
 - Other features

Prerequisites



- Laptop
- RIPE NCC Access account.
 - If you do not have one, create one:
<https://access.ripe.net/registration>
- Log in and visit atlas.ripe.net
- Credits!
 - <https://atlas.ripe.net/user/credits/#!redeem>
 - Voucher “**ENOG11TUTORIAL**”

Warm-up question



- What is your background?
 - Network operator?
 - Software engineer / programmer?
 - Data scientist?
 - Sysadmin?
 - Other? please specify :-)



Introduction to RIPE Atlas

Basics



- RIPE Atlas = global active measurements platform
- Goal: View Internet reachability
- Probes hosted by volunteers
- Measurements performed towards root name servers
 - Visualised as Internet traffic maps

(cont...) Basics



- Users can also run customised measurements
 - Ping, traceroute, DNS, TLS/SSL, NTP and HTTP (with some limitations)
- Data publicly available

Measurements devices



- v1 and v2: Lantronix XPort Pro
- v3: TP-Link TL-MR3020 powered from USB
 - Does not work as a wireless router
 - Same functionality as the old probe
- RIPE Atlas anchor: Soekris net6501-70



RIPE Atlas in numbers



Coverage

IPv4 ASNs covered
3379 (6.266%)

IPv6 ASNs covered
1222 (10.621%)

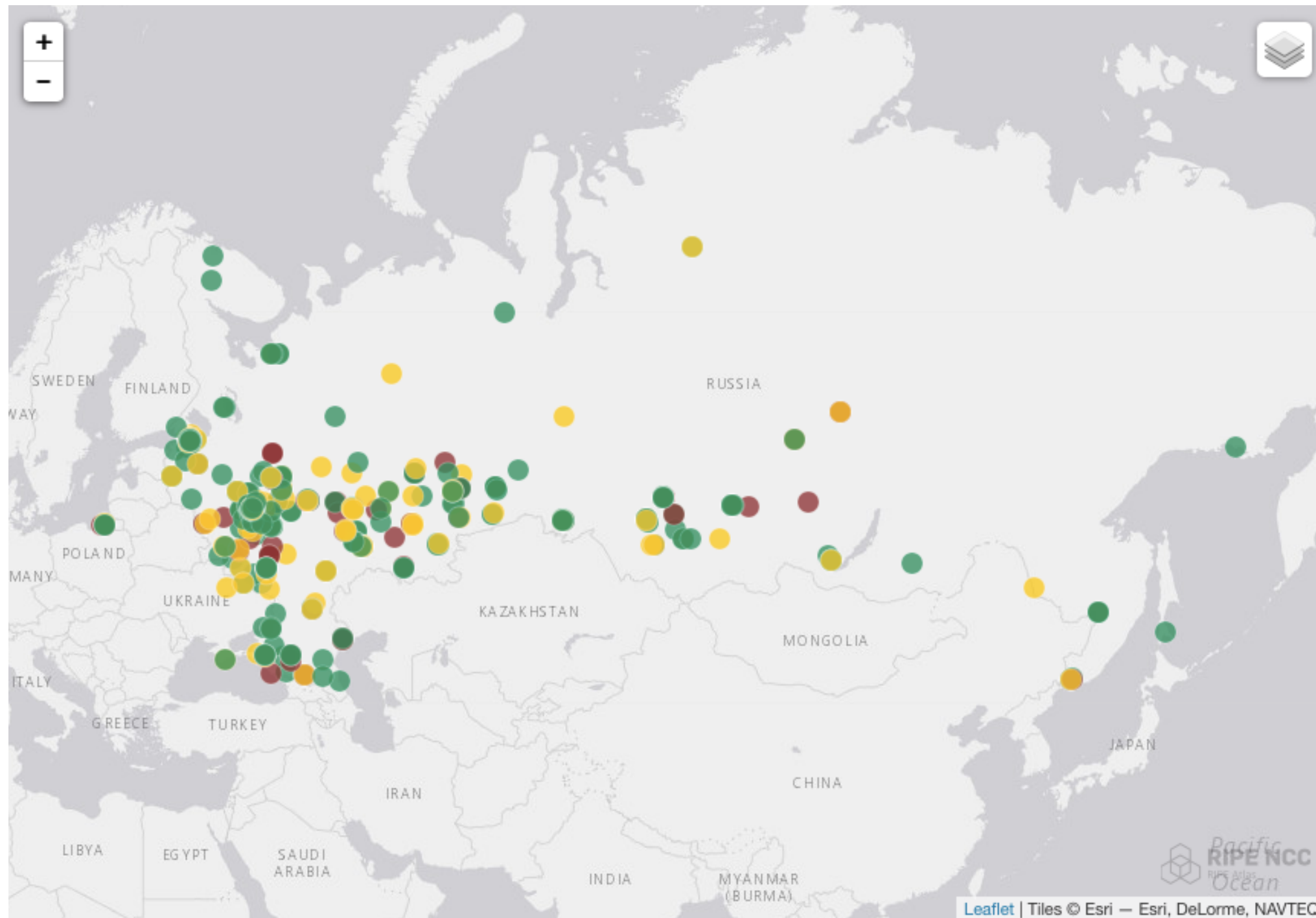
Number of countries covered
184 (93.878%)

Measurements currently running

	Built-in	User-defined
Ping	40	3788
Traceroute	43	2969
DNS	144	4325
SSL Certificate	4	163
NTP	0	38
HTTP	4	417

Country	Probes
United States of America	1053
Germany	997
France	752
United Kingdom	621
Netherlands	543
Russia	509
Czech Republic	269
Switzerland	257
Italy	251
Ukraine	214

RIPE Atlas in Russia



Connected: 505 Disconnected: 224 Abandoned: 162

RIPE Atlas Anchors



- Well-known targets and powerful probes
 - Regional baseline and “future history”
- Anchoring measurements
 - Measurements between anchors
 - 500 probes targeting each anchor with measurements
 - Each probe measures four-five anchors
 - Vantage points for DNSMON service
- 190+ RIPE Atlas anchors

Locations of Anchors



- <https://atlas.ripe.net/anchors/map/>
- Three RIPE Atlas anchors hosted by Fishnet Communications in St. Petersburg and by RIPN and Mail.ru Group in Moscow





What you can get from RIPE Atlas as a visitor

Internet traffic maps



RIPE Atlas <<

About RIPE Atlas >

Get Involved >

Probes and Anchors >

Measurements, Maps and Tools >

Measurements

Internet Maps

Tools

Resources >


RIPE NCC Members

My Atlas >

Staff Pages >

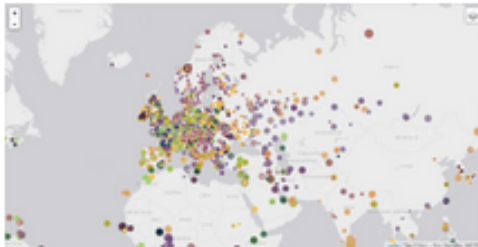
Internet Maps

DNS Root Instances




Shows, for each probe, which root DNS server instance the probe ends up querying, when they ask a particular root server. In other words, it shows the "gravitational radius" for root DNS server instances.

Comparative DNS Root RTT




Shows a comparison of response time for DNS SOA queries to all the root DNS servers. For each probe, a marker shows the "best" root server with colour identifying the related minimum response time.

Root Server Performance




This map shows the reply time to the SOA query of a particular root DNS server, over the selected transport protocol (UDP, TCP or comparison of the two) for each probe.

RTT to Fixed Destinations



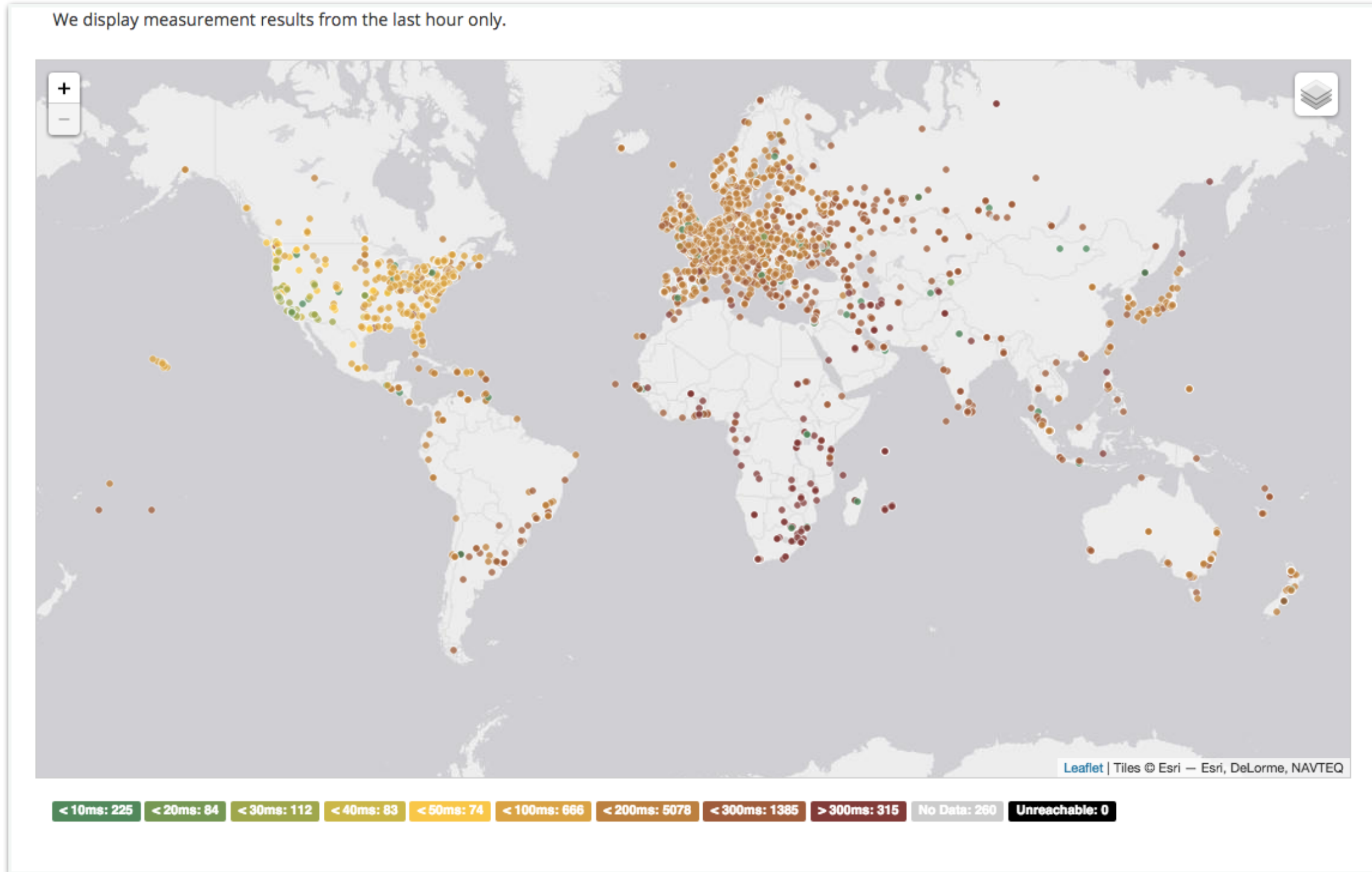
Shows the colour coding for the RTT value for the particular destination for each probe. The minimum / average / maximum values are based on standard "ping" measurements.

Reachability of Fixed Destinations



Shows if the particular fixed destination is reachable or not from each probe. Red markers indicate that the specific destination for these probes are unreachable and green reachable.

Where is B-root?



Searching for probes



- <https://atlas.ripe.net/probes/>

Filter based on
ASN, country,
location...

RIPE NCC
RIPE NETWORK COORDINATION CENTRE

RIPE Database (Whois) Website
Search IP Address or ASN

Manage IPs and ASNs > Analyse > Participate > Get Support > Publications > About Us >

You are here: Home > Analyse > Internet Measurements > RIPE Atlas > Probes

Probes

This is a list of all current RIPE Atlas probes, including information specific to each probe. More probes are continually coming online.

- [Learn more about probes](#)
- [See the probes map](#)
- [Apply for your own probe](#)

Filter by id/asn/country/description Any Status IPv4/v6 Any Country

Public Login to see more

Id	ASN v4	ASN v6	Country	Description	Connection Status
6175	1103	1103		SURFnet bv	🟢 4 weeks
6146	60781	60781		Leaseweb Network B.V.	🟢 4 weeks
6152	28753	28753		Leaseweb Network B.V.	🟢 4 weeks
6137	3333	3333		nl-ams-as3333-preprod	🟢 4 weeks
6147	33280	33280		Afilias	🟢 4 weeks
6112	197216	197216		Delta Softmedia Ltd	🟢 4 weeks
6161	27843	27843		Optical Technologies	🟢 4 weeks
6142	63403	63403		Afilias	🟢 4 weeks
6008	2607	2607		AA sk-bts-as2607	🟢 4 weeks
6001	3333	3333		AA nl-ams-as3333	🟢 4 weeks

Probe details



» You are here: [Home](#) > [Analyse](#) > [Internet Measurements](#) > [RIPE Atlas](#) > [Probes](#) > [Probe #10010](#)

Probe #10010 [\(Register\)](#)

[General](#) [Network](#) [Built-in Measurements](#) [User-defined Measurements](#)

General Information [Edit](#)

Id	10010
MAC Address	F8:D1:11:A9:F3:2C
Architecture	tl-mr3020
Firmware Version	4680 (1070)
Router Type	
Bandwidth Limit	Not set
DNS Entry	Off
Shared Publicly	Yes
User Tags	NAT Chello 200MB
System Tags	V3 Resolves A Correctly Resolves AAAA Correctly IPv4 Works Auto GEOIP city IPv4 Capable IPv4 RFC1918

Connection & Traffic [Edit](#)

5 k
2.5 k
0

08:00 12:00

Bits/s Packets/s

Connected Time [3 days, 9 hours](#)

April May

[3 days, 9 hours](#)

Firmware #10010
4680

Architecture
tl-mr3020

MAC Address
F8:D1:11:A9:F3:2C

The displayed location is an automatic best guess of the **city** based on IP address.
By manually setting a more accurate location you can help to improve the usefulness and correctness of RIPE Atlas.

[Update Location](#)

[Edit](#)

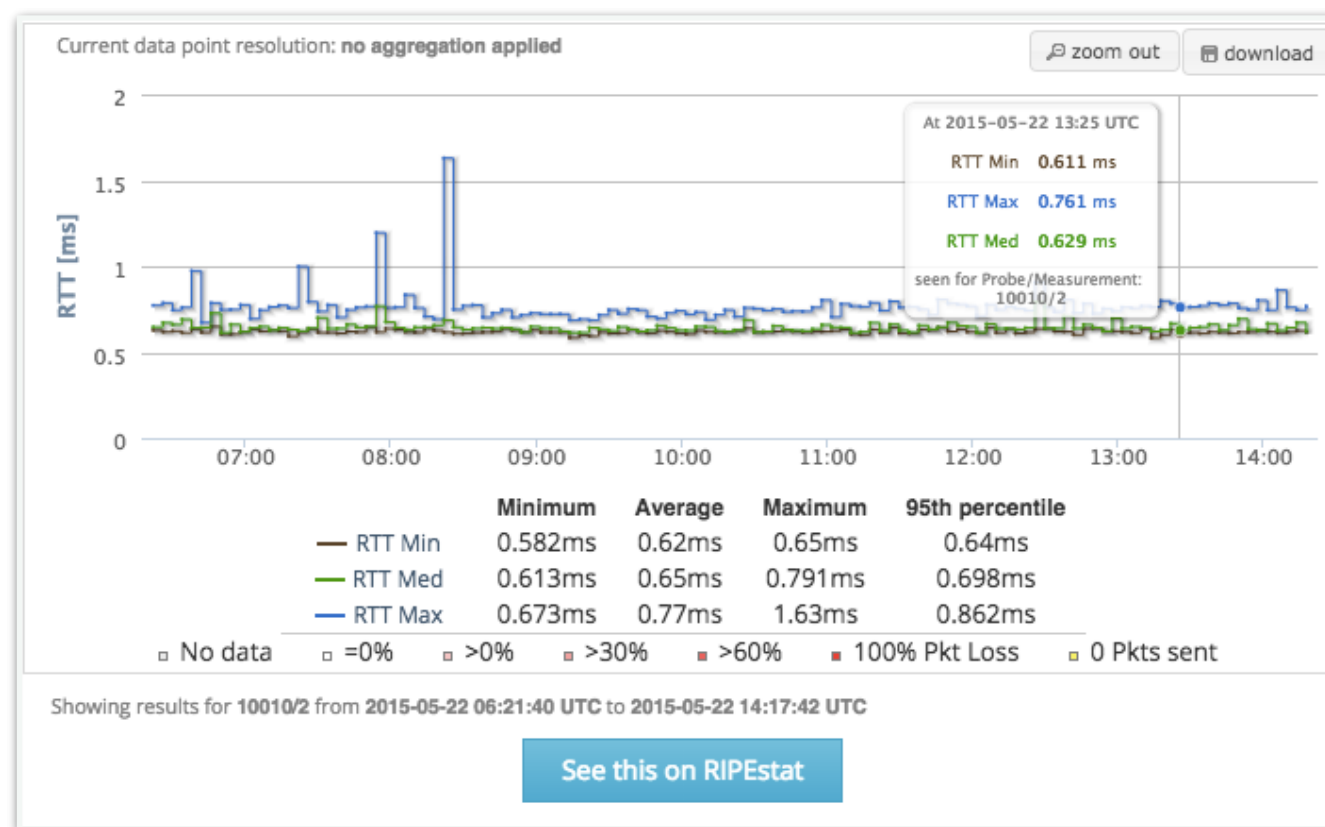
Management Sharing

Only the probe host is permitted to administer this probe.

Zoomable ping graph



- Zoom in / out in time, in the same graph
- Easier visualisation of an event's details
- Selection of RTT class (max, min, average)



Looking up measurements results



- <https://atlas.ripe.net/measurements/>

Manage IPs and ASNs > **Analyse** > Participate > Get Support > Publications > About Us >

RIPE Atlas >> Measurements > RIPE Atlas > Measurements

About RIPE Atlas >

Get Involved >

Probes and Anchors >

Measurements, Maps and Tools v

Measurements

Internet Maps

Tools

Resources >

RIPE NCC Members

Filter by target and/or description

Any Statu: v IPv4/v6 v All types v Of all time v

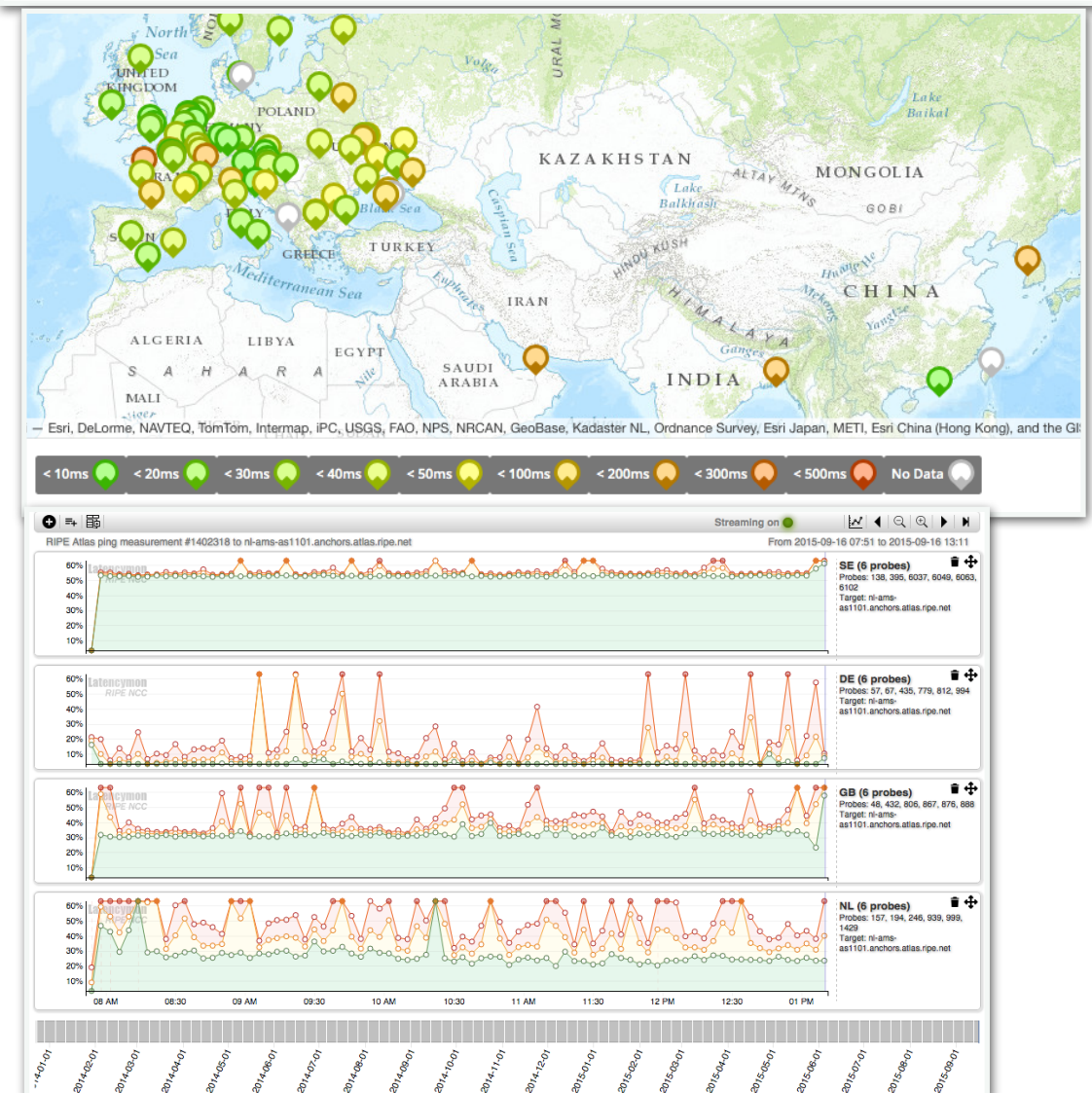
			Description	Probes	Time (UTC)	Status
		nog.net	de-fra-as5580.anchors.atlas.ripe.net	0	2019-11-14 00:30 No Stop Defined	⚙
1411440	○ ⚡	de-muc-as5539.anchors.atlas...	de-muc-as5539.anchors.atlas.ripe.net	0	2019-08-01 00:15 No Stop Defined	⚙
3625872	⊞ ⚡	uk-lon-as5459.anchors.atlas...	Traceroute measurement to uk-lon-as5459.anchors...	Calculating...	2016-03-17 12:00 2016-03-21 12:00	⚙
3625873	⊞ ⚡	ca-mtr-as852.anchors.atlas....	Traceroute measurement to ca-mtr-as852.anchors....	Calculating...	2016-03-17 12:00 2016-03-21 12:00	⚙
3625874	⊞ ⚡	it-mil-as16004.anchors.atla...	Traceroute measurement to it-mil-as16004.anchor...	Calculating...	2016-03-17 12:00 2016-03-21 12:00	⚙
3625875	○ ⚡	nl-haa-as201682.anchors.atl...	Traceroute measurement to nl-haa-as201682.ancho...	Calculating...	2016-03-17 10:42 No Stop Defined	⚙
3625876	○ ⚡	nl-haa-as201682.anchors.atl...	Traceroute measurement to nl-haa-as201682.ancho...	Calculating...	2016-03-17 10:42 No Stop Defined	⚙

Available visualisations: ping



- List of probes: sortable by RTT
- Map: colour-coded by RTT
- LatencyMON: compare multiple latency trends

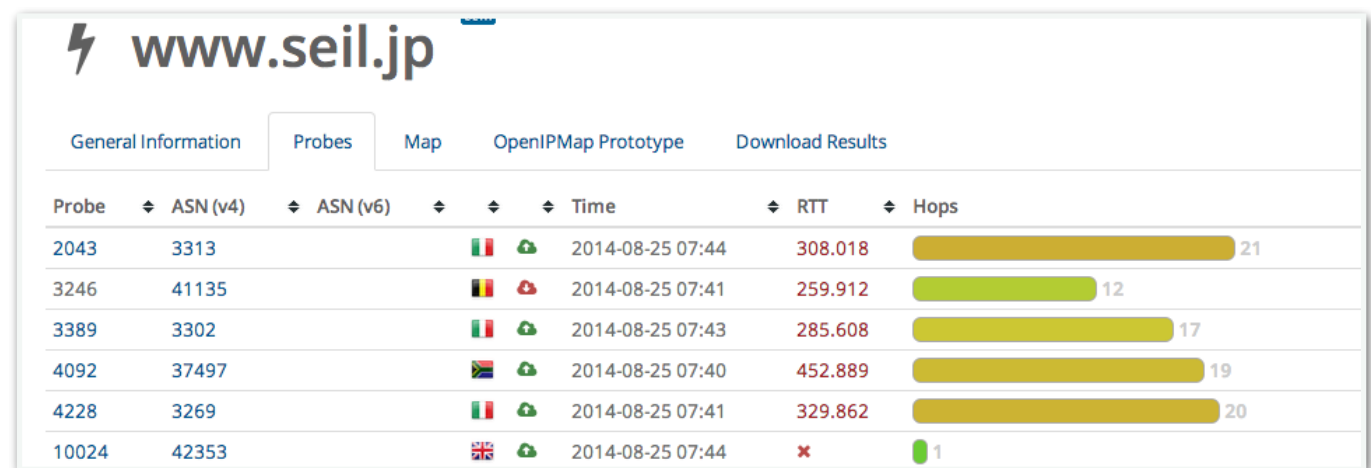
Probe	ASN (v4)	ASN (v6)		Time	RTT
6019	3333	3333		2015-05-19 09:23	1.157
6069	59469	59469		2015-05-19 09:23	15.253
6111	198068	198068		2015-05-19 09:23	37.760
6112	197216	197216		2015-05-19 09:23	35.494
10008	3851			2015-05-19 09:23	24.664
10218	6876			2015-05-19 09:23	37.952
10246	39608			2015-05-19 09:23	36.313
10252	50288			2015-05-19 09:23	62.441
10267	12322			2015-05-19 09:23	31.498
10296	51214			2015-05-19 09:23	Unreachable



Available visualisations: traceroute



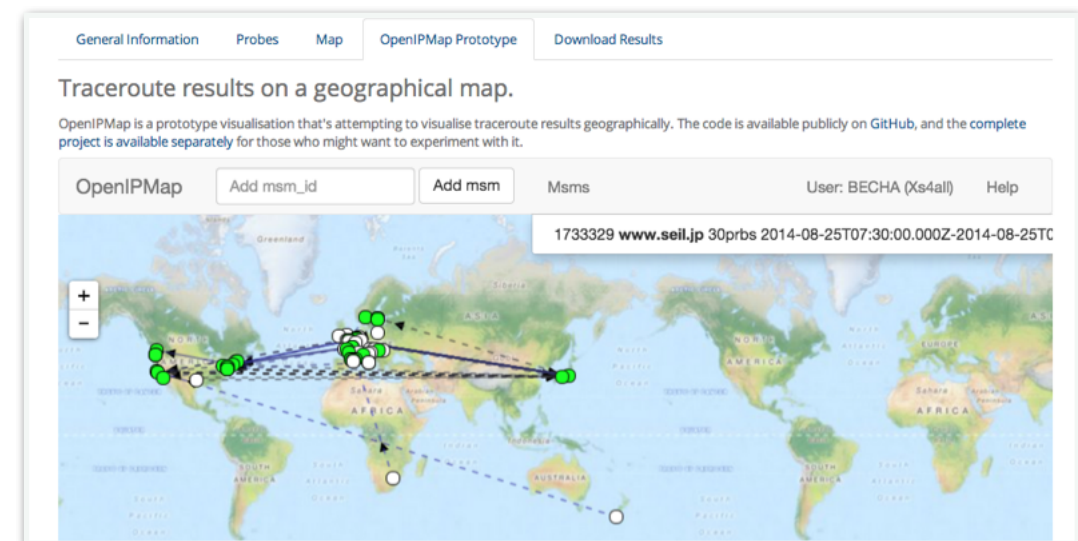
- List of probes, colour-coded number of hops
- Map
- Traceroute paths map, geolocation using OpenIPMap: github.com/RIPE-Atlas-Community/openipmap



www.seil.jp

General Information Probes Map OpenIPMap Prototype Download Results

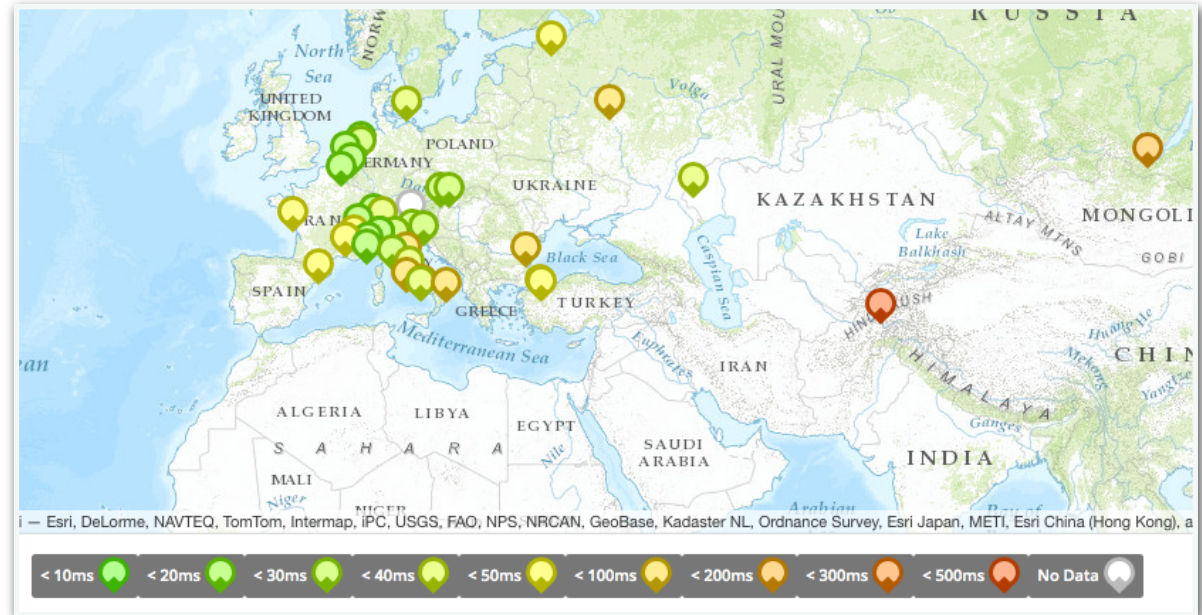
Probe	ASN (v4)	ASN (v6)		Time	RTT	Hops
2043	3313			2014-08-25 07:44	308.018	21
3246	41135			2014-08-25 07:41	259.912	12
3389	3302			2014-08-25 07:43	285.608	17
4092	37497			2014-08-25 07:40	452.889	19
4228	3269			2014-08-25 07:41	329.862	20
10024	42353			2014-08-25 07:44	×	1



Available visualisations: DNS



- Map, colour-coded response time or diversity
- List of probes, sortable by response time



DNS measurement to ns1.opteamax.de

General Information							
Probes							
Probe	ASN (v4)	ASN (v6)			Time	Name	Response Time
17840	6327		🇨🇦	🌐	2015-05-19 09:38	null	362.009
18035	43030		🇮🇳	🌐	2015-05-19 09:50	null	347.39
18129	327805		🇳🇮	🌐	2015-05-19 09:49	null	207.743
15844	32098		🇮🇹	🌐	2015-05-19 09:48	null	184.237
17857	852		🇨🇦	🌐	2015-05-19 09:37	null	177.694
19894	6327		🇨🇦	🌐	2015-05-19 09:36	null	168.689
19204	21513		🇨🇦	🌐	2015-05-19 09:50	null	141.199
15922	30036		🇺🇸	🌐	2015-05-19 09:47	null	133.309

Downloading measurements results



- Click on msm, then “Download”

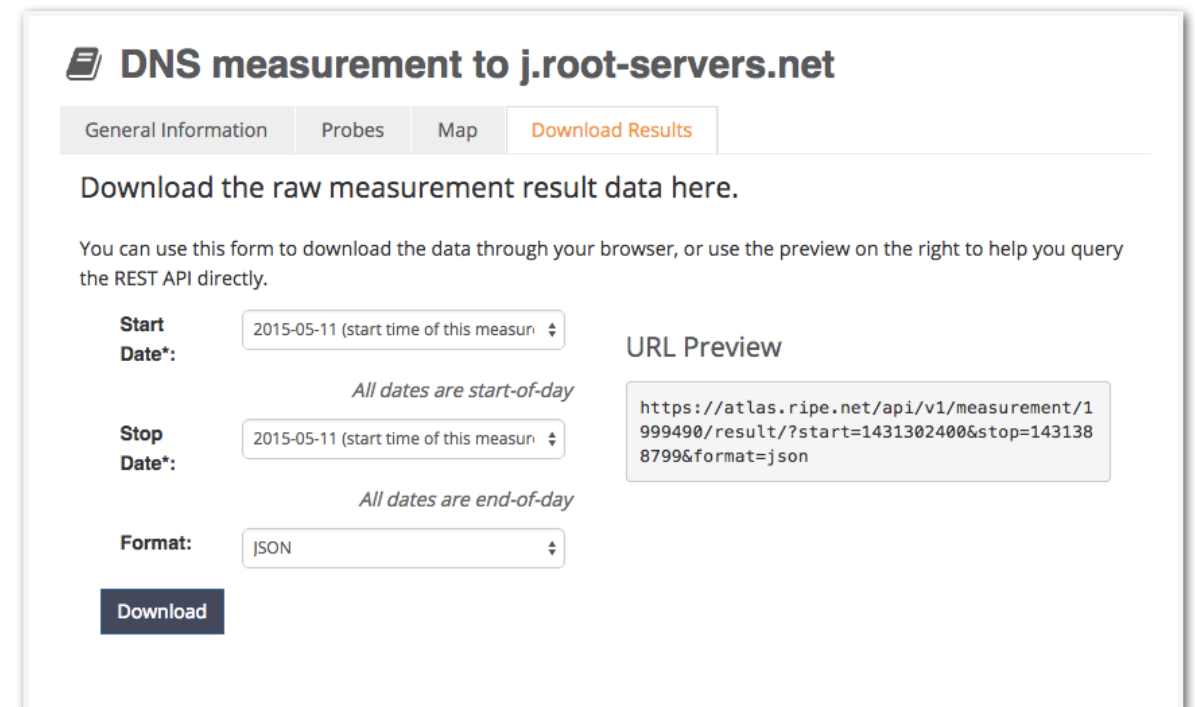
- Or go to URL

- Or use the API

- Results in JSON

- Libraries for parsing available on GitHub

- github.com/RIPE-NCC/ripe.atlas.sagan
- github.com/RIPE-Atlas-Community/



DNS measurement to j.root-servers.net

General Information Probes Map **Download Results**

Download the raw measurement result data here.

You can use this form to download the data through your browser, or use the preview on the right to help you query the REST API directly.

Start Date*: 2015-05-11 (start time of this measurement) All dates are start-of-day

Stop Date*: 2015-05-11 (start time of this measurement) All dates are end-of-day

Format: JSON

Download

URL Preview

`https://atlas.ripe.net/api/v1/measurement/1999490/result/?start=1431302400&stop=1431388799&format=json`

Looking at the result



Reference
(msm ID)

Destination (IP
& name)

Source (probe
public IP address)

Packet loss:
difference
between sent &
received!

```
[{"af":6,"avg":61.32,
"dst_addr":"2a00:1450:4004:802::1014","dst_name":"www.google.com",
"dup":0,
"from":"2001:8a0:7f00:b201:220:4aff:fec5:5b5b",
"fw":4660,"lts":411,
"max":62.148,"min":60.372,
"msm_id":1004005,"msm_name":"Ping",
"prb_id":722,"proto":"ICMP","rcvd":10,
"result":[{"rtt":62.148}, {"rtt":61.437}, {"rtt":61.444}, {"rtt":61.448},
{"rtt":61.794}, {"rtt":61.533}, {"rtt":60.372}, {"rtt":60.373}, {"rtt":61.384},
{"rtt":61.267}],
"sent":10,"size":64,
"src_addr":"2001:8a0:7f00:b201:220:4aff:fec5:5b5b",
"step":240,"timestamp":1410220847,"ttl":54,"type":"ping"},
```

Use existing measurements



- There are many measurements already running!
- Search for existing public measurements first...
- Only then schedule your own measurement if you don't find what you're looking for



Creating a Measurement

Benefits of your own measurements



- A customer reports a problem: they cannot reach one of your servers
 - You can schedule pings or traceroutes from up to 500 RIPE Atlas probes from a particular region to check where the problem might be
- Measuring packet loss on a suspected “bad” link
- Testing anycast deployment

Credits system



- Running your own measurements costs credits: ping = 10 credits, traceroute = 20, etc.
- Why? Fairness and to avoid overload
- Daily spending limit and max measurements user can create
- Earn credits by:
 - Hosting a RIPE Atlas probe + extra credits for:
 - Being a RIPE NCC member
 - Hosting an anchor
 - Sponsoring probes

Logging in



- Log in to atlas.ripe.net
 - Use your RIPE NCC Access account
 - Same account for LIR Portal, RIPE Atlas, RIPEstat, RIPE Labs...
 - Create an account if you don't have one already:
<https://access.ripe.net/registration>

The screenshot shows the RIPE NCC website with the 'Atlas' section highlighted in the navigation menu. The main content area features a 'Welcome to RIPE Atlas!' message, a map of the global probe network, and statistics such as 'Probes connected to RIPE Atlas: 9397' and 'Measurements currently running: 11986'. There is also a 'Log In' button and a 'Get Involved' link.

The screenshot shows the 'Access' page of the RIPE NCC website. It features a 'Sign in using your RIPE NCC Access account' section with a 'Sign in' button. There are input fields for 'Email' and 'Password'. A link for 'Forgot your password?' is visible. A yellow box at the bottom mentions 'New: Two-step verification. Learn more...'. The navigation menu is also visible at the top.

Get credits



- Do you have credits to spend?
 - Redeem this voucher “ENOG11TUTORIAL”
 - <https://atlas.ripe.net/user/credits/#!/redeem>

Credits

Here you can see the history of your credit use and current consumption, transfer credits to someone else, and redeem a voucher for credits if you have one.

40,648,876
-587.71 credits / hour

[History](#)[Charts & Archives](#)[Transfer](#)[Standing Order](#)[Share Access](#)[Redeem voucher](#)

Redeem voucher

Your new credit balance will appear at the top right of this page once you have redeemed your voucher.

Voucher code:

Redeem

Credits overview



You are here: [Home](#) > [Analyse](#) > [Internet Measurements](#) > [RIPE Atlas](#) > [My Atlas](#) > My Credits

RIPE Atlas <<

About RIPE Atlas >

Get Involved >

Probes and Anchors >

Measurements, Maps and Tools >

Resources >

RIPE NCC Members

My Atlas v

Credits

API Keys

Messages

Anchors

Settings

Staff Pages >

Credits

Here you can see the history of your credit use and current consumption, transfer credits to someone else, and redeem a voucher for credits if you have one.

76,408,270

29,518.33 credits / hour

History

Charts & Archives

Transfer

Standing Order

Share Access

Redeem voucher

<< < Page 2 of 52 > >>

Time	Comment	Change	Balance
2016-04-18 01:02 UTC	Probe ID:6019 Anchor uptime 5x extra credit	+ 108,000	74,784,849
2016-04-18 01:02 UTC	Probe ID:6019 Anchor host 5x extra credit	+ 108,000	74,676,849
2016-04-18 00:16 UTC	Measurement: 3460108 Samples: 627	- 1,881	74,568,849
2016-04-17 18:16 UTC	Measurement: 3460108 Samples: 631	- 1,893	74,570,730
2016-04-17 12:15 UTC	Measurement: 3460108 Samples: 630	- 1,890	74,572,623
2016-04-17 06:16 UTC	Measurement: 3460108 Samples: 630	- 1,890	74,574,513
2016-04-17 01:02 UTC	Probe ID:6019 Anchor uptime 5x extra credit	+ 108,000	74,576,403
2016-04-17 01:02 UTC	Probe ID:6019 Anchor host 5x extra credit	+ 108,000	74,468,403
2016-04-17 00:16 UTC	Measurement: 3460108 Samples: 628	- 1,884	74,360,403
2016-04-16 18:16 UTC	Measurement: 3460108 Samples: 630	- 1,890	74,362,287
2016-04-16 12:16 UTC	Measurement: 3460108 Samples: 631	- 1,893	74,364,177
2016-04-16 06:16 UTC	Measurement: 3460108 Samples: 633	- 1,899	74,366,070

My Atlas >
Credits

Give credits
to someone

2: Using GUI to schedule a measurement



- Mostly used for a periodic, long term measurement
 - If just once, ASAP, choose “One-off”
- Choose type, target, frequency, # of probes, region...
 - Interactive interface helps you at each step
- Each measurement will have unique ID
- “API Compatible Specification” is generated by the GUI

Scheduling a measurement



- Log in to atlas.ripe.net
- Measurements, Maps and Tools>Measurements
- Three methods:
 1. Quick & Easy
 - Type
 - Target
 - Done!
 2. Advanced GUI usage
 3. CLI scripting using API

3: Using API to schedule a measurement



- Scripting with the API
 - <https://atlas.ripe.net/docs/measurement-creation-api/>
 - <https://atlas.ripe.net/keys/>
- You will need API keys
 - To create measurements without logging in
 - To securely share your measurement data

API documentation



- <https://atlas.ripe.net/docs/measurement-creation-api/>
 - <https://atlas.ripe.net/doc/credits>
 - <https://atlas.ripe.net/doc/udm>
- <https://atlas.ripe.net/keys/>
- New: API v2
 - <https://atlas.ripe.net/docs/api/v2/manual/>
 - https://labs.ripe.net/Members/suzanne_taylor_muzzin/new-ripe-atlas-apis



Exercise

Create a measurement

Tasks



- Create a ping measurement:
 - Involving ten probes
 - To a target of your choice
 - Source is your country
 - Duration of two days
- 1. Warm-up: Create a measurement using the GUI
- 2. Create API Key
- 3. Schedule a measurement using the API

Sub-task 1: Use web interface



- Useful hint: once you generate a measurement, copy “API Compatible Specification” to text file
- Note Measurement-ID

RIPE Atlas	«
About RIPE Atlas	>
Get Involved	>
Results	>
My Atlas	▼
Probes	
Measurements	
Credits	
API Keys	
Messages (81 new)	
Anchors	
Sponsorships	

Measurements

Filter by target and/or description		Any Status	IPv4/v6	All types	Of all time	T	X
Mine	Favourites	Hidden	Public	All			
Id	Type	Target	Description	Probes	Time (UTC)	Status	
1965015	Vesna Manojlovic	IPv4 ping	b92.net	Ping measurement to b92.net	49	2015-04-21 08:20 2015-04-21 08:30	■
1940389	Vesna Manojlovic	IPv4 sslcert	twitter.com	SSL measurement to twitter.com	104	2015-04-07 09:39 2015-04-07 09:45	■

Create a New Measurement

Step 1 Definitions

Please select the type of measurement you want to create

+ Ping + Traceroute + DNS + SSL + NTP

Step 2 Probe Selection

Worldwide 50 x

+ New Set - wizard + New Set - manual + IDs List + Reuse a set from a measurement

Step 3 Timing

This is a One-off: ☐

Start time: As soon as possible Stop time: Never

> Measurement API Compatible Specification

Create My Measurement(s)

Sub-task 2: Create API key



- Click on “Create an API Key”
- Choose type: “create a new user-defined measurement”
- “Object” is not applicable (N/A) for this type
- Give it a label

(...cont) Sub-task 2: Create API key



- Give it a duration of validity (leave empty for defaults)
- “Key” value to be passed on to the API call (next step)

RIPE Atlas

About RIPE Atlas

Get Involved

Results

My Atlas

Probes

Measurements

Credits

API Keys

Messages (81 new)

Anchors

API Keys

+ Create an API key

<input type="checkbox"/> Key	Created	Permission	Object	Label	Valid From	Valid To	Enabled
<input type="checkbox"/> 984a774c-33ce-4b97-9767-fb48efda6c12	2013-01-31 13:05 UTC	Download results of a user defined measurement	1002953 I b.hosteddnsservice.com				✓
<input type="checkbox"/> e5ba646b-abf1-4f01-8bf1-5267a9dd56ce	2013-01-31 12:52 UTC	Download results collected by a specific probe	13: k13				✓
<input type="checkbox"/> 9788b7e0-9d4b-4787-8a42-fce8f2f2e929	2013-01-11 14:53 UTC	Download results of a user defined measurement	1002676 I www.google.com				✓

Sub-task 3: Use API



- Schedule a measurement using API
 - Use the “key” you just generated
 - Hint: copy and past API call syntax from the measurement generated by the GUI

```
$ curl -H "Content-Type: application/json" -H "Accept: application/json" -X POST  
-d '{ "definitions": [ { "target": "ping.xs4all.nl", "description": "My First API  
Measurement", "type": "ping", "af": 4 } ], "probes": [ { "requested": 10, "type":  
"country", "value": "RS" } ] }' https://atlas.ripe.net/api/v1/measurement/?  
key=YOUR\_API\_KEY
```

```
air-becha:~ becha$ curl -H "Content-Type: application/json" -H "Accept:  
application/json" -X POST -d '{ "definitions": [ { "target": "ping.xs4al  
l.nl", "description": "My First Measurement", "type": "ping", "af": 4 }  
], "probes": [ { "requested": 10, "type": "country", "value": "RS" } ] }  
' https://atlas.ripe.net/api/v1/measurement/?key=7b4c3441-4504-4d83-9ed7  
-fbf1a007d060  
{"measurements": [2421551]}air-becha:~ becha$
```




Command-line (CLI) Toolset

RIPE Atlas CLI



- Network troubleshooting for command line pros
- Familiar output (ping, dig, traceroute)
- Linux/OSX
 - <http://ripe-atlas-tools.readthedocs.org/en/latest/installation.html#requirements-and-installation>
- Windows [experimental]
 - <https://github.com/chrisamin/ripe-atlas-tools-win32>

Install RIPE Atlas tools on Ubuntu



- <https://ripe-atlas-tools.readthedocs.org/en/latest/>
- `$ sudo apt-get install python-dev libffi-dev libssl-dev`
- `$ sudo apt-get install python-virtualenv python-pip`
- Setup virtualenv
- `pip install ripe.atlas.tools`

Install RIPE Atlas tools on *nix



- Install virtualenv
\$ sudo easy_install pip
\$ sudo pip install virtualenv
- Create virtualenv for atlas-tools
\$ virtualenv venv-atlas
- Activate virtualenv (note the '.')
\$.venv-atlas/bin/activate
- Install atlas-tools
\$ pip install ripe.atlas.tools
- Add to PATH
export PATH=\$PATH:~/venv-atlas/bin

Install RIPE Atlas tools on Windows



- github.com/chrisamin/ripe-atlas-tools-win32
- [github.com/chrisamin/ripe-atlas-tools-win32/
releases/download/v0.1.1/
RipeAtlasToolsSetup.exe](https://github.com/chrisamin/ripe-atlas-tools-win32/releases/download/v0.1.1/RipeAtlasToolsSetup.exe)

RIPE Atlas CLI



- Open source
 - RIPE NCC-led community contribution
- Documentation
 - <https://ripe-atlas-tools.readthedocs.org/>
- Source:
 - <https://github.com/RIPE-NCC/ripe-atlas-tools/>
- How to contribute:
 - <https://github.com/RIPE-NCC/ripe-atlas-tools/blob/master/CONTRIBUTING.rst>

Configure RIPE Atlas CLI



- Reuse the API key of the previous exercise
 - Or create a new one at <https://atlas.ripe.net/keys/>
- Configure your CLI
 - `ripe-atlas configure --set authorisation.create=MY_API_KEY`

Fetch an existing measurement



- Fetch the ping measurement 2340408
 - ripe-atlas report 2340408

Search probes



- Search all probes in AS3333
 - ripe-atlas probes --asn 3333
- Show specific fields
 - ripe-atlas probes --asn 3333 --field asn_v6 --field country --field is_public --field description --field status
- Search for probes in and around Moscow
 - ripe-atlas probes --location "Moscow, Russia" --radius 150

Create a measurement



- Create a ping measurement to wikipedia.org
 - One-off, default parameters
 - `ripe-atlas measure ping --target wikipedia.org`

Looking good! Your measurement was created and details about it can be found here:

<https://atlas.ripe.net/measurements/3499718/>

Connecting to stream...

```
48 bytes from probe #18433 94.112.176.45 to 91.198.174.192 (91.198.174.192): ttl=50 times:41.979, 41.492, 40.769,
48 bytes from probe #20111 37.151.230.180 to 91.198.174.192 (91.198.174.192): ttl=57 times:100.511, 100.136, 100.325,
48 bytes from probe #25003 176.193.48.211 to 91.198.174.192 (91.198.174.192): ttl=59 times:47.967, 47.476, 47.403,
48 bytes from probe #20313 5.199.160.9 to 91.198.174.192 (91.198.174.192): ttl=58 times:36.501, 36.245, 36.285,
48 bytes from probe #22573 89.176.43.44 to 91.198.174.192 (91.198.174.192): ttl=52 times:28.747, 27.712, 28.446,
48 bytes from probe #19413 89.71.47.56 to 91.198.174.192 (91.198.174.192): ttl=51 times:49.89, 49.779, 50.277,
48 bytes from probe #18635 78.52.132.137 to 91.198.174.192 (91.198.174.192): ttl=57 times:37.462, 38.095, 37.73,
48 bytes from probe #23223 62.65.126.46 to 91.198.174.192 (91.198.174.192): ttl=53 times:23.169, 23.412, 33.067,
48 bytes from probe #17511 87.81.148.2 to 91.198.174.192 (91.198.174.192): ttl=56 times:13.281, 12.885, 13.039,
48 bytes from probe #12584 46.175.22.202 to 91.198.174.192 (91.198.174.192): ttl=59 times:36.073, 35.788, 35.883,
```

Other examples of ping



- Geo-specific from 20 probes from Canada:
 - `ripe-atlas measure ping --target example.com --probes 20 --from-country ca`
- 20 Canadian probes that definitely support IPv6:
 - `ripe-atlas measure ping --target example.com --probes 20 --from-country ca --include-tag system-ipv6-works`
- Create a recurring measurement:
 - `ripe-atlas measure ping --target example.com --interval 3600`



Exercise

Using RIPE Atlas CLI

Search probes



- Use the traceroute command to test the reachability of wikipedia.org on TCP port 443 from 20 probes in France
- Render the results collected in the previous exercise in JSON format

Search probes



- Use the traceroute command to test the reachability of wikipedia.org on TCP port 443 from 20 probes in France
 - `ripe-atlas measure traceroute --protocol TCP --target wikipedia.org --port 443 --probes 20 --from-country fr`
- Render the results collected in the previous exercise in JSON format
 - `ripe-atlas report {MSM_ID} --renderer raw`



Monitoring

See your network from the outside



- Integrate “status checks” with existing monitoring tools (such as Icinga)
- Developed by community: RIPE Atlas Monitor
- Using real-time data streaming
 - Server monitoring
 - Detecting and visualising outages

Steps for integration



1. Create a RIPE Atlas ping measurement
2. Go to “status checks” URL (RESTful API call)
 - https://atlas.ripe.net/api/v1/status-checks/2340408/?median_rtt_threshold=10
 - <https://atlas.ripe.net/docs/api/v2/manual/measurements/status-checks.html>
3. Add your alerts in Nagios or Icinga
 - Make use of the built-in “check_http” plugin
 - https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/scripts_for_nagios_icinga_alerts



Versatile “RIPE Atlas Monitor”



- Pier Carlo Chiodi's work (@pierky)
 - <https://github.com/pierky/ripe-atlas-monitor>
 - <https://ripe-atlas-monitor.readthedocs.org/>
- Additional use cases: traceroute analysis, hostname resolution, AS path detection, verifying TLS connections...
- Recently published on RIPE Labs
 - https://labs.ripe.net/Members/pier_carlo_chiodi/ripe-atlas-monitor

RIPE Atlas streaming



- RIPE Atlas streaming is an architecture that allows users to receive the measurement results as soon as they are sent by the probes
 - in real time
 - Publish/subscribe through web sockets
- There are three types of data:
 - Measurement results
 - Probe connection status events
 - Measurements metadata

(con...) RIPE Atlas streaming



- Visualising network outages
 - <http://sg-pub.ripe.net/demo-area/atlas-stream/conn/>
- Real-time server and performance monitoring
- Filtering and reusing measurement results
 - By Target
 - By Source
 - and More...
- Documentation:
 - <https://atlas.ripe.net/docs/result-streaming/>

RIPE Atlas Result Streams



Stream type: result

When stream_type is set to "result", the client will receive measurement results.

```
<script src="https://atlas-stream.ripe.net/socket.io.js"></script>
<script>

  // Create a connection (it can be also http on port 80)
  var socket = io("https://atlas-stream.ripe.net:443", { path : "/stream/socket.io" });

  // Subscribe to results coming from all the probes involved in the measurement 1791207
  socket.emit("atlas_subscribe", { stream_type: "result", msm: 1791207 });

  // Declare a callback to be executed when a measurement result is received
  socket.on("atlas_result", function(result){
    console.log("I received ", result);
  });

</script>
```

Stream type: probestatus

When stream_type is set to "probestatus", the client will receive connection and disconnection events of probes.

```
<script src="https://atlas-stream.ripe.net/socket.io.js"></script>
<script>

  // Create a connection
  var socket = io("https://atlas-stream.ripe.net:443", { path : "/stream/socket.io" });

  // Subscribe to the connection events of the probe 22527
  socket.emit("atlas_subscribe", { stream_type: "probestatus", prb: 22527 });

  // Declare a callback to be executed when a probe connection event is received
  socket.on("atlas_probestatus", function(status){
    console.log("I received ", status);
  });

</script>
```



Exercise

Using streaming API

EX1: Monitoring server reachability



- Scenario: customers sometimes complain that it occasionally takes a long time to reach your service or server
- Action: ping your server from 500 probes
 - Decide what is acceptable latency threshold to apply
 - Notice and react when you start receiving samples
- Task: Use the ping measurement ID 2340408
 - Choose which threshold (e.g. greater than 30ms)
 - Imposes the threshold on “min” (the minimum result of the three ping attempts)

Steps



1. Go to
<http://atlas.ripe.net/webinar/streaming01.html>
2. Open the development console
3. Wait for results to arrive
4. Save the HTML file locally and edit the code

Page source



```
view-source:sg-pub.ripe.net/webinar/streaming01.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Streaming exercise 01</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   </head>
8   <body>
9     <div>Current maximum RTT: <b><span id="output">nothing yet</span></b></div>
10    <div>Open the source code to see how it works. Create your tool/visualisation with the
RIPE Atlas streaming!</div>
11  </body>
12
13  <script src="https://stat.ripe.net/widgets/lib/js/jquery/jquery-1.11.2.min.js"></script>
14
15
16  <!-- The following file is needed for the streaming -->
17  <script src="https://atlas-stream.ripe.net/socket.io.js"></script>
18  <script>
19    var $outputDiv = $("#output");
20
21    // Create a connection
22    var socket = io("https://atlas-stream.ripe.net", { path : "/stream/socket.io" });
23
24    // Declare a callback to be executed when a measurement result is received
25    socket.on("atlas_result", function(result){
26
27      console.log("I received ", result); // Print the result in the console
28
29      if (result.hasOwnProperty("max")) {
30        $outputDiv.html(result["max"]); // Print the result in the html page
31      }
32
33    });
34
35    // Subscribe to results coming from all the probes involved in the measurement 2340408
36    socket.emit("atlas_subscribe", { stream_type: "result", msm: 2340408 });
37
38  </script>
39 </html>
40
41
```


Example of results



```

Q Elements Network Sources Timeline Profiles Resources Audits Console AngularJS
<top frame> Preserve log
Filter  ☐ Regex ☒ All ☐ Errors ☐ Warnings ☐ Info ☐ Logs ☐ Debug ☐ Hide network messages
XHR finished loading: GET "http://atlas-stream.ripe.net/stream/socket.io/?EI0=2&transport=polling&t=1431095373684-0".
XHR finished loading: GET "http://atlas-stream.ripe.net/stream/socket.io/?EI0=2&transport=polling&t=1431095373739-1&sid=eB0kM7zfWFT2c-ScAAaH".
I received ▶ Object {af: 4, prb_id: 16669, result: Array[3], ttl: 42, avg: 326.841...}
I received ▶ Object {af: 4, prb_id: 16669, result: Array[3], ttl: 42, avg: 325.7933333333...}
I received ▶ Object {af: 4, prb_id: 16669, result: Array[3], ttl: 42, avg: 326.048...}
I received ▶ Object {af: 4, prb_id: 16669, result: Array[3], ttl: 42, avg: 327.3253333333...}
I received ▶ Object {af: 4, prb_id: 15965, result: Array[3], ttl: 45, avg: 47.6313333333...}
I received ▶ Object {af: 4, prb_id: 15965, result: Array[3], ttl: 45, avg: 47.6996666667...}
I received ▶ Object {af: 4, prb_id: 15965, result: Array[3], ttl: 45, avg: 47.4816666667...}
I received ▶ Object {af: 4, prb_id: 19566, result: Array[3], ttl: 40, avg: 47.054...}
I received ▶ Object {af: 4, prb_id: 19566, result: Array[3], ttl: 40, avg: 47.8626666667...}
I received ▶ Object {af: 4, prb_id: 19566, result: Array[3], ttl: 40, avg: 47.5946666667...}
I received ▶ Object {af: 4, prb_id: 19566, result: Array[3], ttl: 40, avg: 47.5003333333...}
I received ▶ Object {af: 4, prb_id: 18311, result: Array[3], ttl: 49, avg: 32.577...}
I received ▶ Object {af: 4, prb_id: 18311, result: Array[3], ttl: 49, avg: 34.0843333333...}
I received ▶ Object {af: 4, prb_id: 18311, result: Array[3], ttl: 49, avg: 32.7513333333...}
I received ▶ Object {af: 4, prb_id: 16010, result: Array[3], ttl: 46, avg: 182.4463333333...}
I received ▶ Object {af: 4, prb_id: 16010, result: Array[3], ttl: 46, avg: 193.9953333333...}
I received ▶ Object {af: 4, prb_id: 16010, result: Array[3], ttl: 46, avg: 182.2913333333...}
I received ▶ Object {af: 4, prb_id: 16010, result: Array[3], ttl: 46, avg: 191.6103333333...}
I received ▶ Object {af: 4, prb_id: 14918, result: Array[3], ttl: 49, avg: 34.817...}
I received ▶ Object {af: 4, prb_id: 14918, result: Array[3], ttl: 49, avg: 35.0093333333...}
I received ▶ Object {af: 4, prb_id: 14918, result: Array[3], ttl: 49, avg: 35.0843333333...}
I received ▶ Object {af: 4, prb_id: 20668, result: Array[3], ttl: 45, avg: 38.8846666667...}
I received ▶ Object {af: 4, prb_id: 20668, result: Array[3], ttl: 45, avg: 38.8626666667...}
I received ▶ Object {af: 4, prb_id: 20668, result: Array[3], ttl: 45, avg: 38.8806666667...}
I received ▶ Object {af: 4, prb_id: 6093, result: Array[3], ttl: 49, avg: 128.7273333333...}
I received ▶ Object {af: 4, prb_id: 6093, result: Array[3], ttl: 49, avg: 128.7373333333...}
I received ▶ Object {af: 4, prb_id: 6093, result: Array[3], ttl: 49, avg: 128.8883333333...}

```

EX2: Monitoring server reachability



- Imagine you are in the situation described in the exercise before, but you didn't schedule a measurement in advance
 - You don't have a measurement ID
- You want to get all the measurements reaching 193.0.10.197
- Now restrict the results to just include ping measurements

EX2: Results



```
<!DOCTYPE html>
<html>
  <head>
    <title>Streaming exercise 01</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div>Current maximum RTT: <b><span id="output">nothing yet</span></b></div>
    <div>Open the source code to see how it works. Create your tool/visualisation with the RIPE Atlas streaming!</div>
  </body>

  <script src="https://stat.ripe.net/widgets/lib/js/jquery/jquery-1.11.2.min.js"></script>

  <!-- The following file is needed for the streaming -->
  <script src="https://atlas-stream.ripe.net/socket.io.js"></script>
  <script>
    var $outputDiv = $("#output");

    // Create a connection
    var socket = io("https://atlas-stream.ripe.net", { path : "/stream/socket.io" });

    // Declare a callback to be executed when a measurement result is received
    socket.on("atlas_result", function(result){

      console.log("I received ", result); // Print the result in the console

      if (result.hasOwnProperty("max")) {
        $outputDiv.html(result["max"]); // Print the result in the html page
      }

    });

    // Subscribe to results reaching 193.0.10.197
    socket.emit("atlas_subscribe", { stream_type: "result", destinationAddress: "193.0.10.197" });

  </script>
</html>
```



**Take part in the
RIPE Atlas community**

RIPE Atlas community (part 1)



- Individual volunteers host probes in homes or offices
- Organisations host RIPE Atlas anchors
- Sponsor organisations give financial support or host multiple probes in their own networks

RIPE Atlas community (part 2)



- Ambassadors help distribute probes at conferences, give presentations, etc.
- Developers contribute free and open software
- Network operators create measurements to monitor and troubleshoot
- Researchers and students write papers



Hosting a probe



- Create a RIPE NCC Access account
- Go to <https://atlas.ripe.net/apply>
- You will receive a probe by post
- Register your probe
- Plug in your probe
- If you receive a probe from an ambassador (trainer, sponsor, someone at a conference), just register it and plug it in!

More Hackathons!



- Join the hackathon in 2016
 - Before each RIPE Meeting - save the dates!
 - 22-23 October, Madrid



Contact us



- <https://atlas.ripe.net> and <http://roadmap.ripe.net/ripe-atlas/>
- Users' mailing list: ripe-atlas@ripe.net
- Articles and updates: <https://labs.ripe.net/atlas>
- Questions and bugs: atlas@ripe.net
- Twitter: [@RIPE_Atlas](https://twitter.com/RIPE_Atlas) and [#RIPEAtlas](https://twitter.com/hashtag/RIPEAtlas)



Additional slides



“IXP country Jedi”

Measuring Impact of IXPs on
Keeping Traffic Local

Benefits (part 1)



- Operators
 - Routing and traffic optimisation
- IXP operators
 - Shows how IXPs help keep traffic local and regional
- IPv6 advocates
 - Comparing IPv4 and IPv6 paths

Benefits (part 2)



- Country level: regulators, politicians, cyber-security...
 - How much traffic stays within the country? Where do the paths go?
 - Comparing countries with each other
- RIPE Atlas community
 - More probes in more networks = higher quality of measurements data

Benefits (part 3)

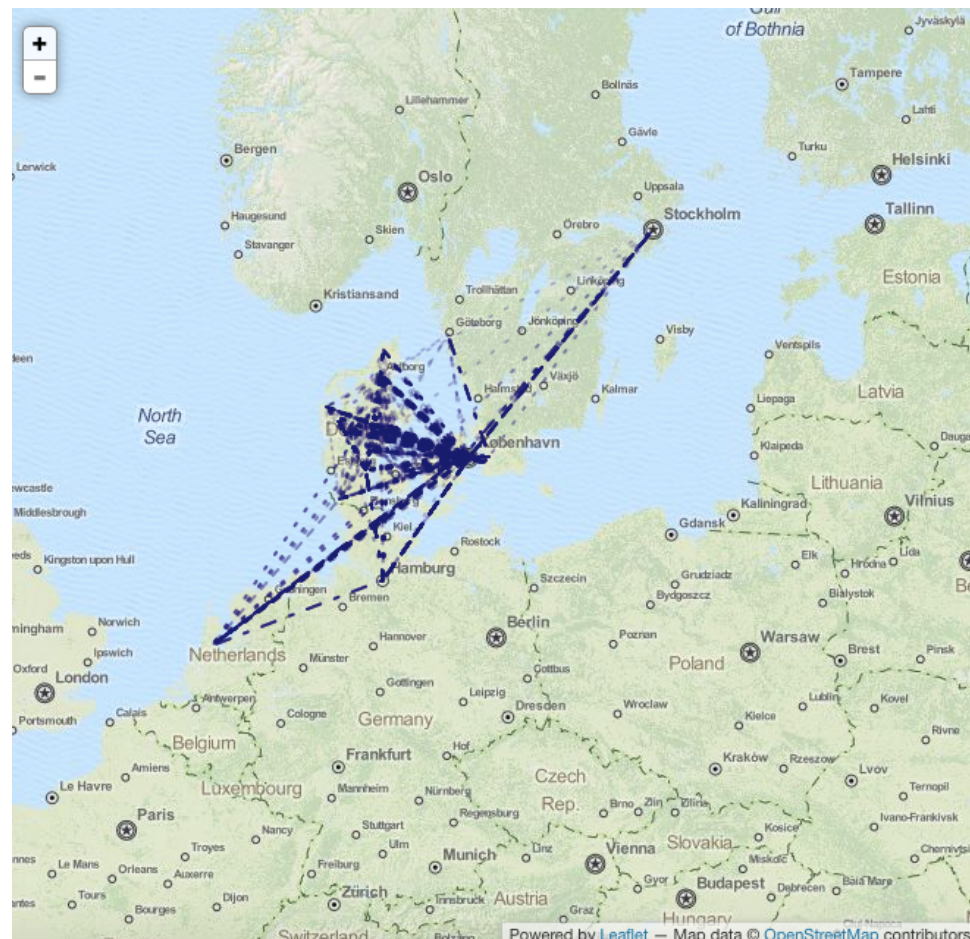


- Geolocation data community
 - Use case for improving data quality
- Examples:
 - <https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas>
 - <https://labs.ripe.net/Members/emileaben/measuring-countries-and-ixps-in-the-see-region>
 - <http://sg-pub.ripe.net/emile/ixp-country-jedi/CL+AR-2015-04/geopath/>

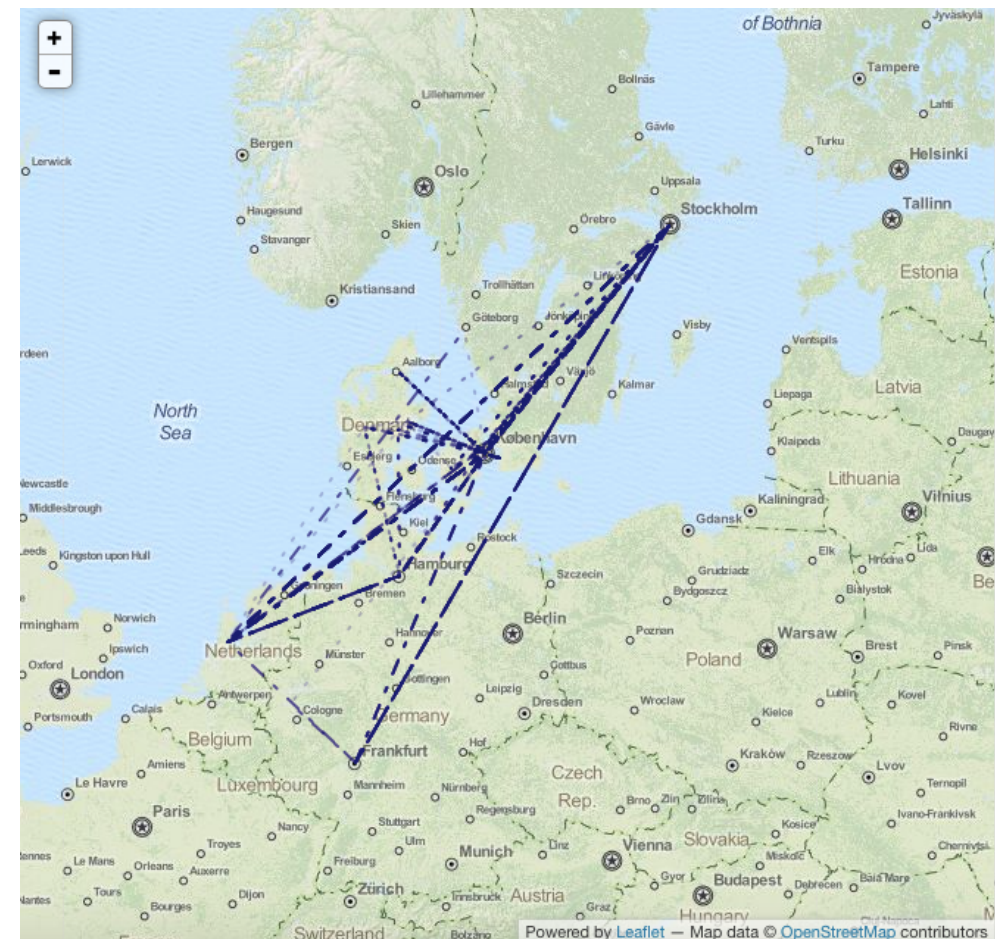
Paths staying in the country?



- Difference between IPv4 and IPv6 paths
 - <http://sg-pub.ripe.net/emile/ixp-country-jedi/history/2016-05-01/DK/geopath/>



IPv4 paths

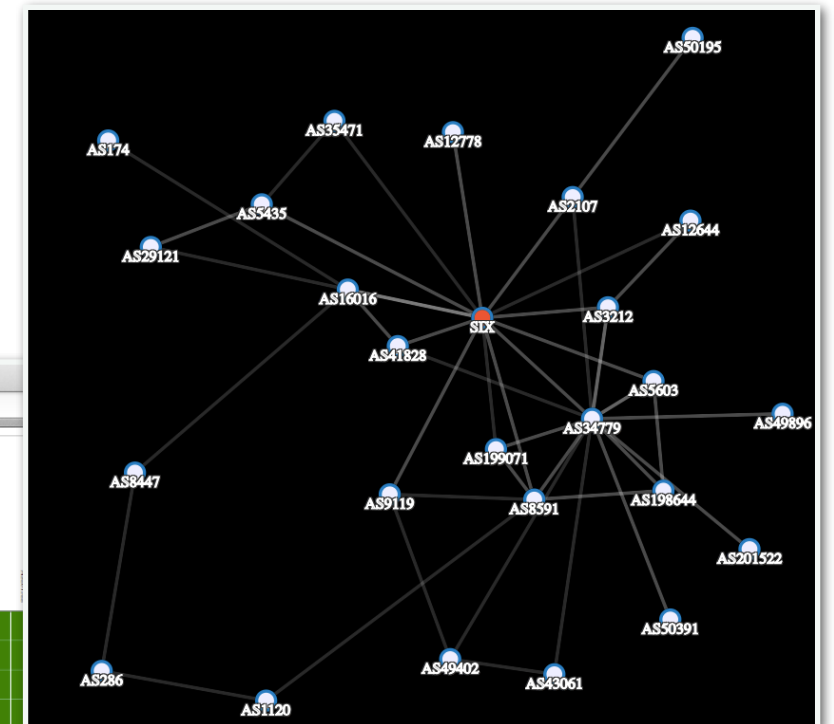
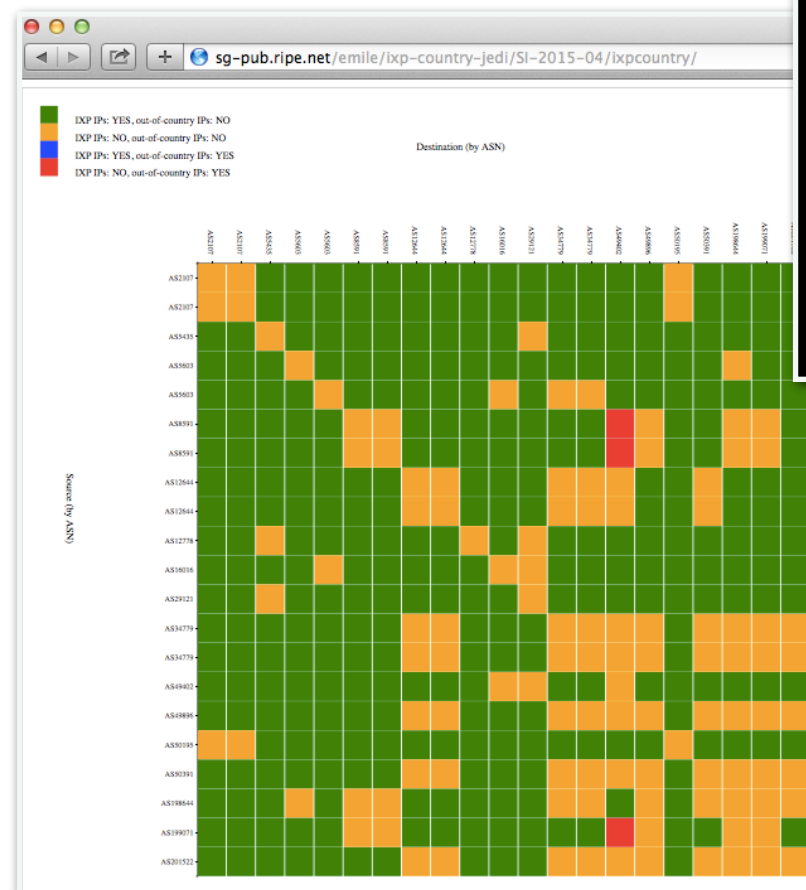
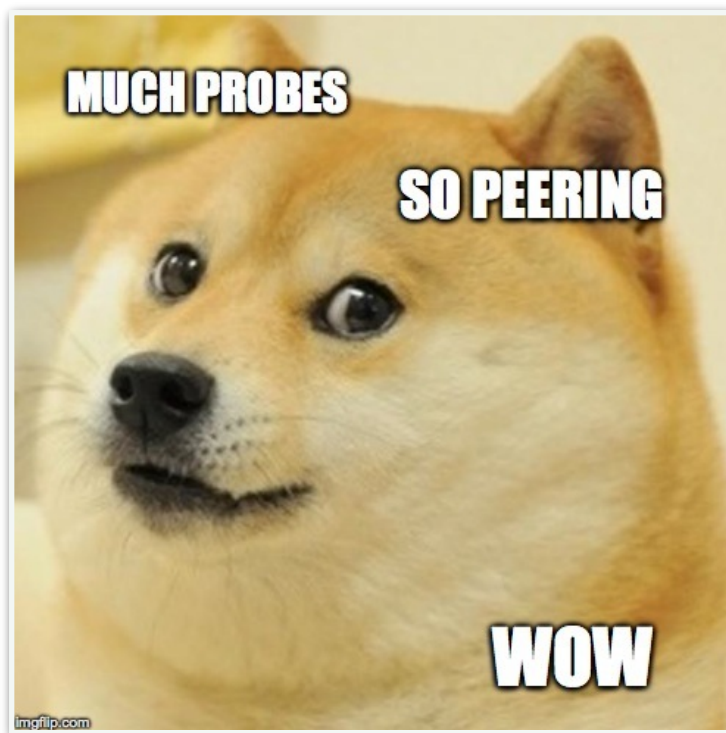


IPv6 paths

Paths going via an IXP?



- <http://sg-pub.ripe.net/emile/ixp-country-jedi/SI-2015-04/ixpcountry/>



Method



- Traceroute measurements using RIPE Atlas probes
- Steps:
 - Identify ASNs in the country using RIPEstat
 - Identify IXPs and IXP LANs using PeeringDB
 - Construct mesh: from all (*) country's probes to each other
 - *Maximum of two probes per ASN and only “public” probes with “good” geolocation
- Hops geolocated using “OpenIPMap” database

Actions (part 1)



- Use this tool to find possible suboptimal routing and fix it
 - Find your ASN in the mesh
 - Find the person from another ASN
 - Take them out for tea :)

Actions (part 1)



- To improve accuracy of this diagnostic tool
 - If your ASN is not on the graph, apply for a RIPE Atlas probe
 - Add more probes to your country to increase “resolution”
 - If you move, remember to update your probe’s geolocation

Actions (part 2)

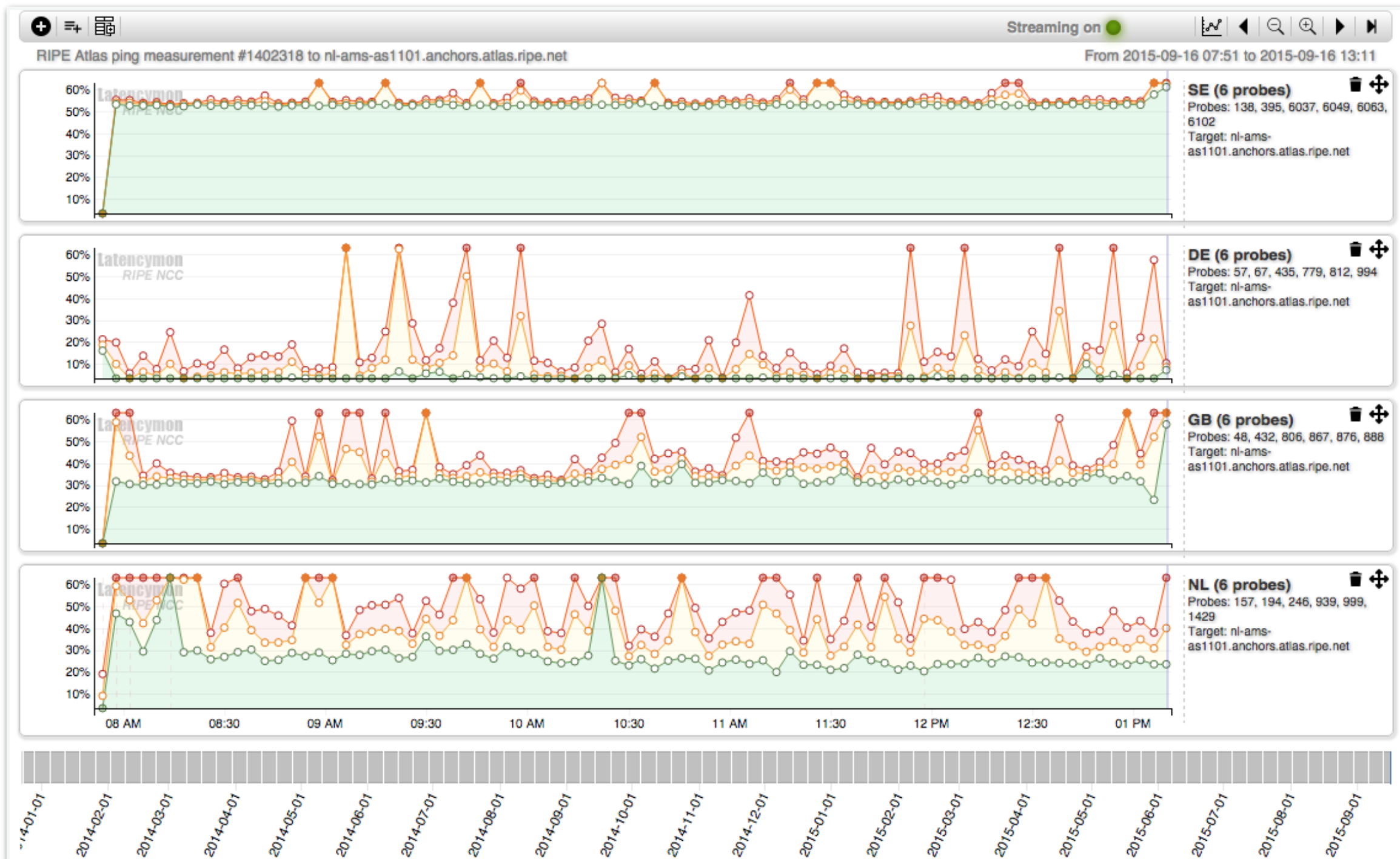


- Reuse and rewrite the code: it is free and open source software
 - <https://github.com/emileaben/ixp-country-jedi>
- Improve infrastructure geolocation: contribute data to OpenIPMap!
 - <https://marmot.ripe.net/openipmap/>
 - <https://github.com/RIPE-Atlas-Community/openipmap>

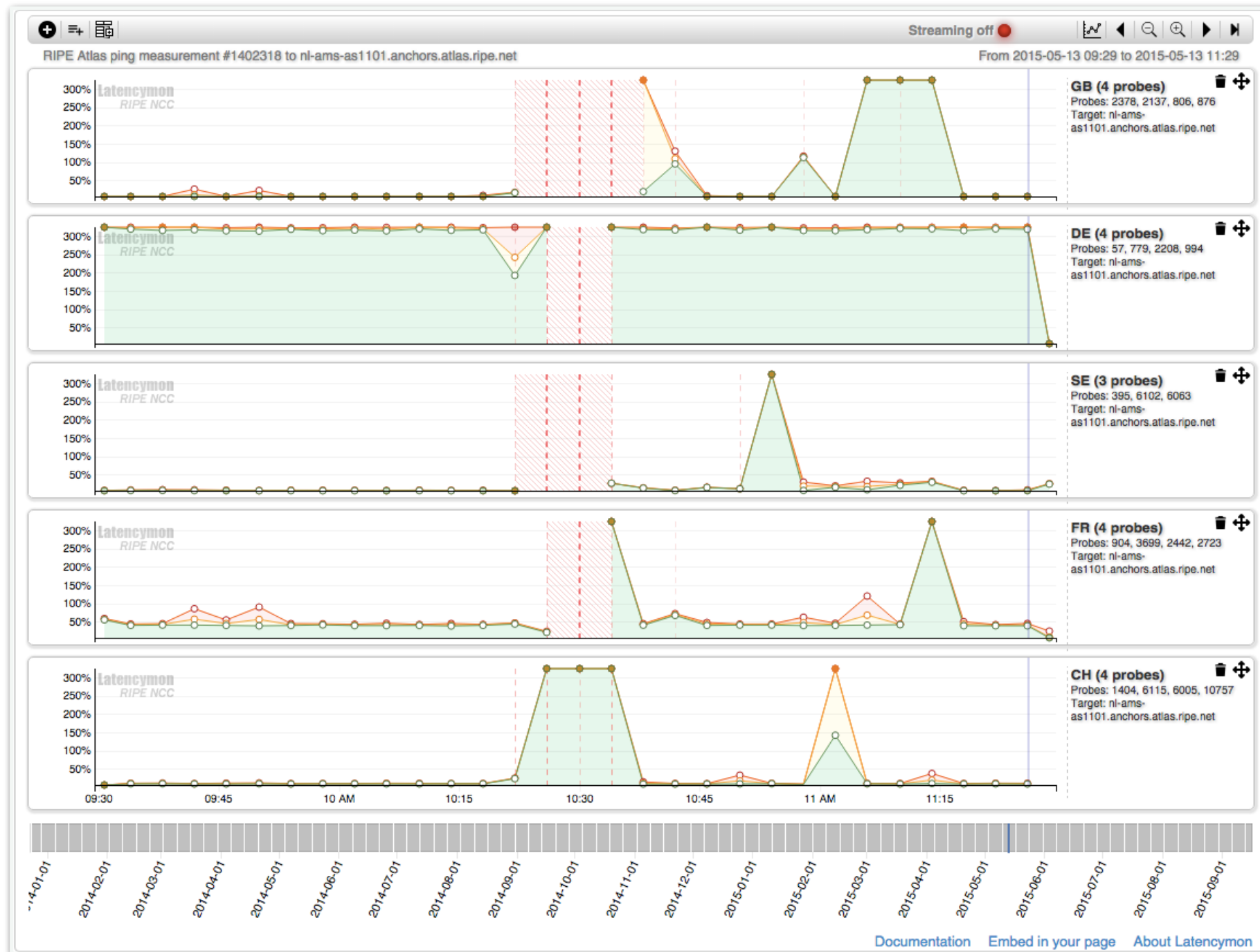


LatencyMON

LatencyMON



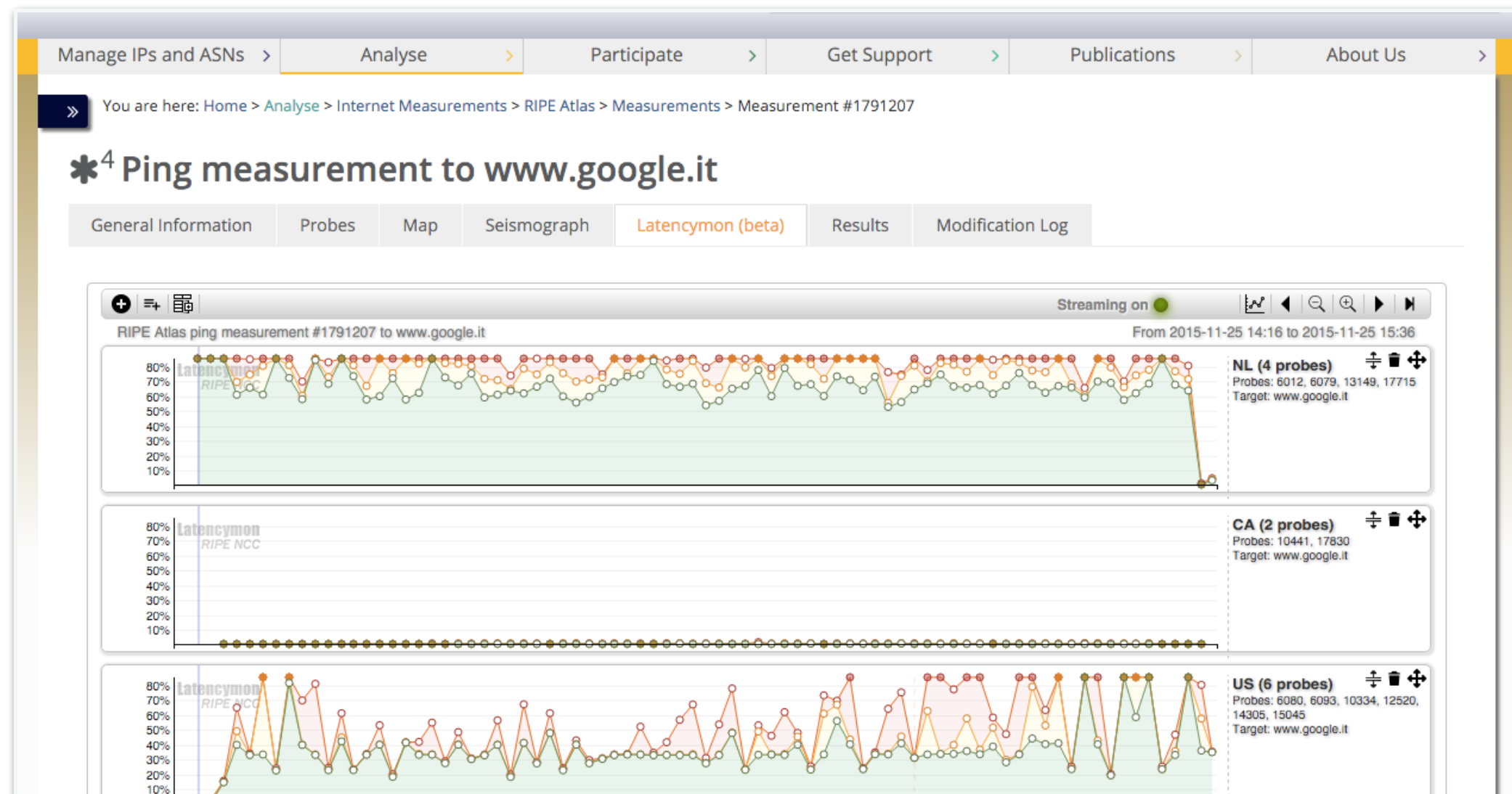
LatencyMON packet loss



LatencyMON tab



- Demo: <https://atlas.ripe.net/measurements/1791207/>



LatencyMON goals



- Performance comparisons to reach a service or website from different countries or providers
- Measuring the spread of a network outage
- Measuring and comparing CDN or DNS resolution in multiple geographic areas

(...cont) LatencyMON goals



- Reusing measurements - even for measurement types other than ping - to get information about latencies
- Comparing multiple ISPs or hosting providers at the same time from vantage points with characteristics similar to those at the user end
- Creating views that are easily shareable and can be embedded in reports

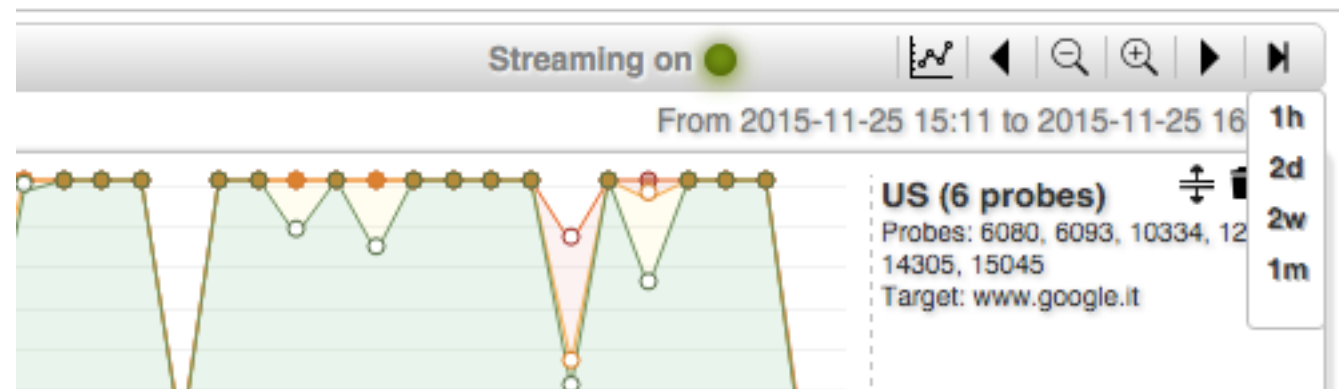
Monitoring with LatencyMON



- Embed latencyMON

```
<script src="https://atlas.ripe.net/resource/latencymon/latencymon-widget-main.js"></script>
<div id="place-here"></div>
<script>
  initLatencymon(
    '#place-here',
    {}, // Tool options, see table below for more info
    { measurements:[1791207, 2083078] } // Query options, see table below for more info
  );
</script>
```

- Enable real-time streaming



- Documentation: <https://atlas.ripe.net/docs/tools-latencymon/#embed>



Exercise

Check your geographical distribution

Tasks 1/2



- You want to improve the geographical distribution of your contents by checking where the delay can be reduced
 - You created a measurement in the previous exercise and you have the measurement ID (otherwise use: 1791207)
 - Click on the LatencyMON tab of your measurement
 - Delete all the default-created charts

Tasks 2/2



- Create one group of two probes from DE (Germany)
- Create one group of two probes from NL (Netherlands)
- Create one group of two probes from US (United States)
- What is the country with the greatest latency in your case? Type it in the chat!



Exercise

Compare HTTP over IPv4 and IPv6

Tasks



- You want to check the performances of accessing an infrastructure over IPv4 and IPv6 (two measurements) as a real end user
 - You need an anchor for HTTP measurements!
e.g. use <https://atlas.ripe.net/probes/6001/>
 - Open the LatencyMON tab of the HTTP IPv6 (2841527) measurement
 - Add the HTTP IPv4 measurement ID in LatencyMON (2841526)
 - Create **two groups of eight probes** each: one per measurement
- Share the link of the view of the last two days



Exercise

Setting up “Status Checks”

Tasks



- Set up and configure a “status check”
 - For an existing ping measurement <https://atlas.ripe.net/measurements/2340408/>
 - Hint: <https://atlas.ripe.net/api/v1/status-checks/2340408/>
- Configure the status check in such a way that you will trigger an alert for this measurement
- Optional: set up status check for your own ping measurement!

Solution



- One possible solution:
 - Set the median RTT to a lower level:
 - https://atlas.ripe.net/api/v1/status-checks/2340408/?median_rtt_threshold=10
- Example of the alerts

```
{"total_alerts":32,"global_alert":true,
"probes":{
  "18433":{"all":[null,null,null],"last":null,"last_packet_loss":100.0,"alert":true,"source":"Area: WW","alert_reasons":["loss"]},
  "15041":{"source":"Area: WW","last_packet_loss":0.0,"last":19.928,"alert":false},
  "18696":{"all":[null,null,null],"last":null,"last_packet_loss":100.0,"alert":true,"source":"Area: WW","alert_reasons":["loss"]},
  "16265":{"source":"Area: WW","last_packet_loss":0.0,"last":22.72,"alert":false},
  "20236":{"all":[null,null,null],"last":null,"last_packet_loss":100.0,"alert":true,"source":"Area: WW","alert_reasons":["loss"]},
  "12944":{"all":[null,null,null],"last":null,"last_packet_loss":100.0,"alert":true,"source":"Area: WW","alert_reasons":["loss"]},
  "2195":{"all":[null,null,null],"last":null,"last_packet_loss":100.0,"alert":true,"source":"Area: WW","alert_reasons":["loss"]},
```



More RIPE Atlas features

Most popular features



- Six types of measurements: ping, traceroute, DNS, SSL/TLS, NTP and HTTP (to anchors)
- APIs to start measurements and get results
- Powerful and informative visualisations
- CLI tools
- Streaming data: real-time results
- Plus: “Time Travel”, LatencyMON, DomainMON
- Roadmap

Latest results API



- <https://atlas.ripe.net/docs/measurement-latest-api/>
 - Widget monitoring value in real time (100 probes pinging websites worldwide)
 - Alert based on average measurements per hour
 - Big network event, e.g. Internet outage in a region
 - DNS domain monitoring; configurable measurements using ten RIPE Atlas anchors
- https://labs.ripe.net/Members/suzanne_taylor_muzzin/ripe-atlas-latest-results-api-and-parsing-library

Secure measurement creation and sharing



- Use API keys to:
 - Create measurements without logging in
 - Securely share your measurement data with others
- To create, manage and delete API keys:
 - <https://atlas.ripe.net/keys/>
 - <https://atlas.ripe.net/docs/keys/>
- Examples:
 - <https://atlas.ripe.net/docs/rest/>

Security aspects



- Probes:
 - Hardware trust material (regular server address, keys)
 - No open ports; initiate connection; NAT is okay
 - Don't listen to local traffic
 - No passive measurements
- Measurements triggered by “command servers”

(...cont) Security aspects



- Inverse SSH tunnels
- Source code published
- Reported vulnerabilities:
 - <https://atlas.ripe.net/docs/security/>

Crowdsourced Infrastructure

Geolocation: OpenIPMap



- Visualising traceroutes on the map is difficult!
 - Routers' geolocation data is often very inaccurate
 - RIPE Atlas performs many traceroutes through Internet core
- Community of operators contributes data to Open IP Map (think: OpenStreetMap for IPs)
 - <https://marmot.ripe.net/openipmap/>
- You can modify, reuse and improve the code
 - <https://github.com/RIPE-Atlas-Community/openipmap>



Questions



atlas@ripe.net
[@RIPE_Atlas](#)