



RIPE
NCC

RIPE Atlas Tutorial

emile.aben@ripe.net

Mumbai, India | August 2015

- Learn how to:
 - Use RIPE Atlas measurements for network monitoring and troubleshooting
 - Use API calls to create measurements
 - Integrate RIPE Atlas with existing monitoring systems

- Get your questions answered

- Who hosts a RIPE Atlas probe?
 - Who would like to host one?
- Who is a RIPE Atlas Ambassador?
- Who has created measurements with RIPE Atlas before?
- Who has programming experience?

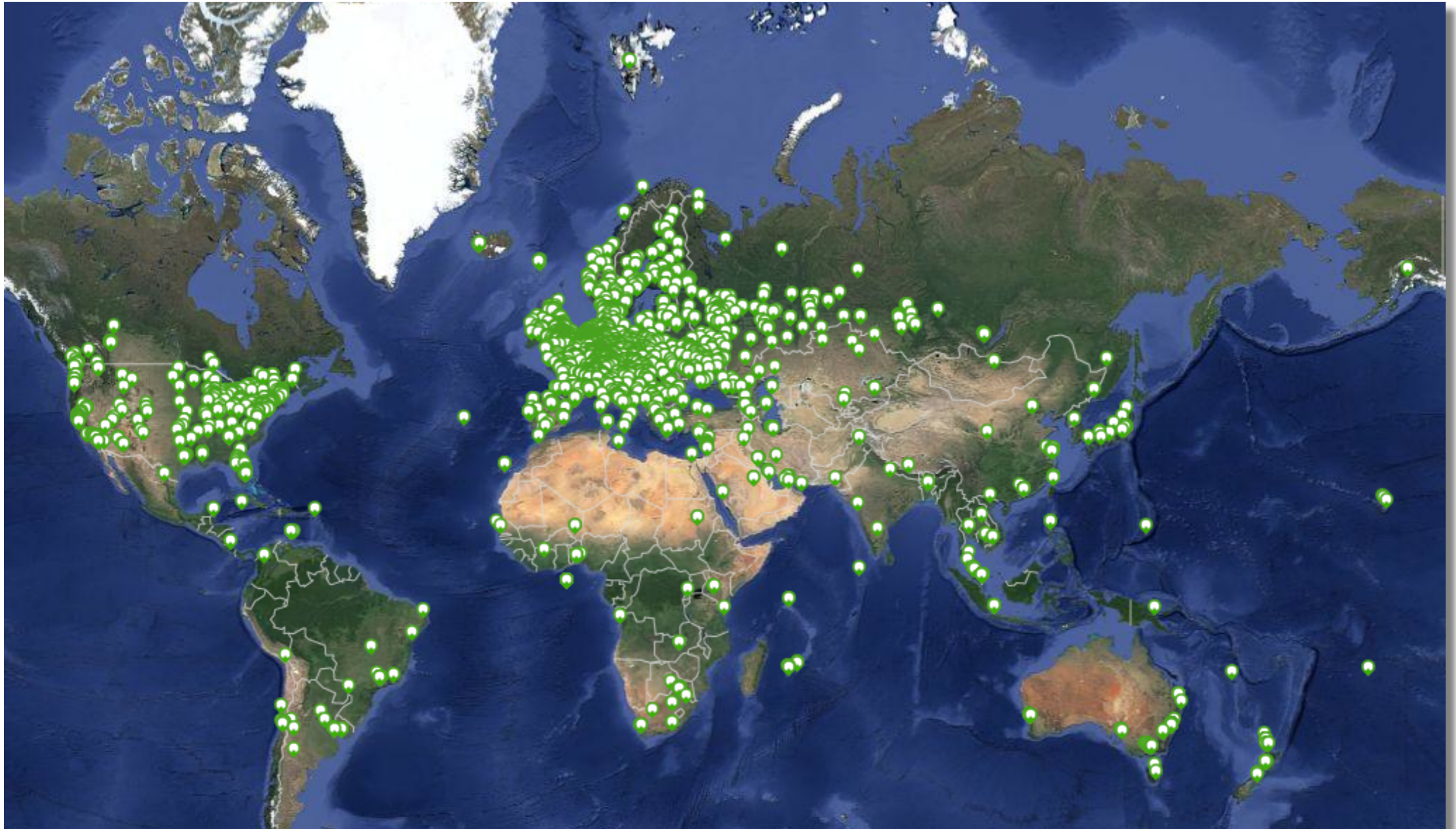
- Introduction to RIPE Atlas
- RIPE Atlas Probes
- RIPE Atlas Measurements
- Finding public measurements & analysing results
- Creating measurements
- Integration with network monitoring systems
- Real-time performance monitoring
- Take part in the RIPE Atlas community



Introduction to RIPE Atlas



- RIPE Atlas is a global active measurements platform
- Goal: Improve Internet through measurements
- Probes hosted by volunteers
- Data publicly available



<https://atlas.ripe.net/results/maps/network-coverage/>

- Regular probes (version 1,2,3)
 - Small form factor boxes
- RIPE Atlas anchors
 - 1U servers (Soekris)
- Future: virtual machine probes?
 - Interested?



- Ongoing global measurements
 - Towards DNS root name servers and RIPE Atlas infrastructure
 - Visualised as Internet maps
- Ongoing regional measurements
 - Towards RIPE Atlas anchors
- Users can run customised measurements
 - ping, traceroute, DNS, SSL/TLS and NTP

- Everything has a primary identification number
- RIPE Atlas probes: probe ID
 - Example:
 - Probe ID: 6040
 - URL: <https://atlas.ripe.net/probes/6040/>
- RIPE Atlas measurements: measurement ID
 - Example:
 - Measurement ID: 1004005
 - URL: <https://atlas.ripe.net/measurements/1004005/>



RIPE Atlas Probes



- Logged-in user:
 - Go to “My Atlas” > “Probes”
- Direct URL: <https://atlas.ripe.net/probes/>

Search box

Probe ID:
click for
details

Probes

Filter by id/asn/location/country/descript Connected IPv4/v6 Any Country

Public

Id	ASN v4	ASN v6	Country	Description	Connection Status
27472	28840				🟢 1 week
27465	3217			Ultramarine	🟢 2 days, 21 hours
27450	39811				🟢 1 day, 19 hours
27437	49893			gpon from bitrace telecom	🟢 2 hours, 39 minutes
27411	2609			http://probev3.ripe.net	🟢 11 minutes
24450	1267	200630		PRE DC Galitello PROBE #1	🟢 2 days, 13 hours
24428	44858			AS44858 - PROXSYS	🟢 3 days, 18 hours
24411	15895				🟢 12 hours, 13 minutes
23774	15595			Skyline Telecom Ltd.	🟢 1 day, 15 hours
23762	35382	35382		Oy Capnova Ltd	🟢 2 days, 3 hours
23755	50195	50195			🟢 1 week, 3 days
23752	5603	5603		vr17	🟢 9 hours, 3 minutes
23743	5603	5603		ARNES: AlexM	🟢 11 hours, 7 minutes

- Search box



Filter by id/asn/location/country/descript Connected IPv4/v6 Any Country

- Search by:

- Probe ID
- Autonomous System Number (ASN)
- Country
- Description
- Status (connected/disconnected)

Probes

Filter by id/asn/location/country/descript

Connected

IPv6

India (51)



Public

Id	ASN v4	ASN v6	Country	Description	Connection Status
22793	45528			MUM2	2 days, 2 hours
21518	24560			adav-Germany	7 hours, 20 minutes
20814	24560			Airtel DSL Gurgaon	3 hours, 1 minute
20254	55577			Sandeep's Probe -Hyd	13 hours, 19 minutes
17590	24560			DD	1 day
17011	9829			Ripe Probe	20 minutes
13602	24560			Prashant	1 day, 5 hours
11832	24560			Lucknow	10 hours, 33 minutes
4913	2697	2697		EIB-BLR	1 day, 3 hours
1083	45942	45942		Gomti Broadband Services	2 hours, 15 minutes

- In probe search result:
 - Click on probe ID
- Direct URL: https://atlas.ripe.net/probes/<probe_id>

xs4all adsl 52/5 IPv6

General | **Network** | Built-in Measurements | User-defined Measurements

General Information

Id	4
Architecture	probev1
Firmware Version	4700
Router Type	FritzBox AVM 7360
Shared Publicly	Yes

User Tags

DSL Home NAT
Native IPv6
iwantbcp38compliancetesting

System Tags

V1 Resolves A Correctly
Resolves AAAA Correctly
IPv4 Works IPv6 Works
IPv4 Capable IPv6 Capable
IPv4 RFC1918

Connection & Traffic

Bits/s Packets/s

Connected Time

3 weeks

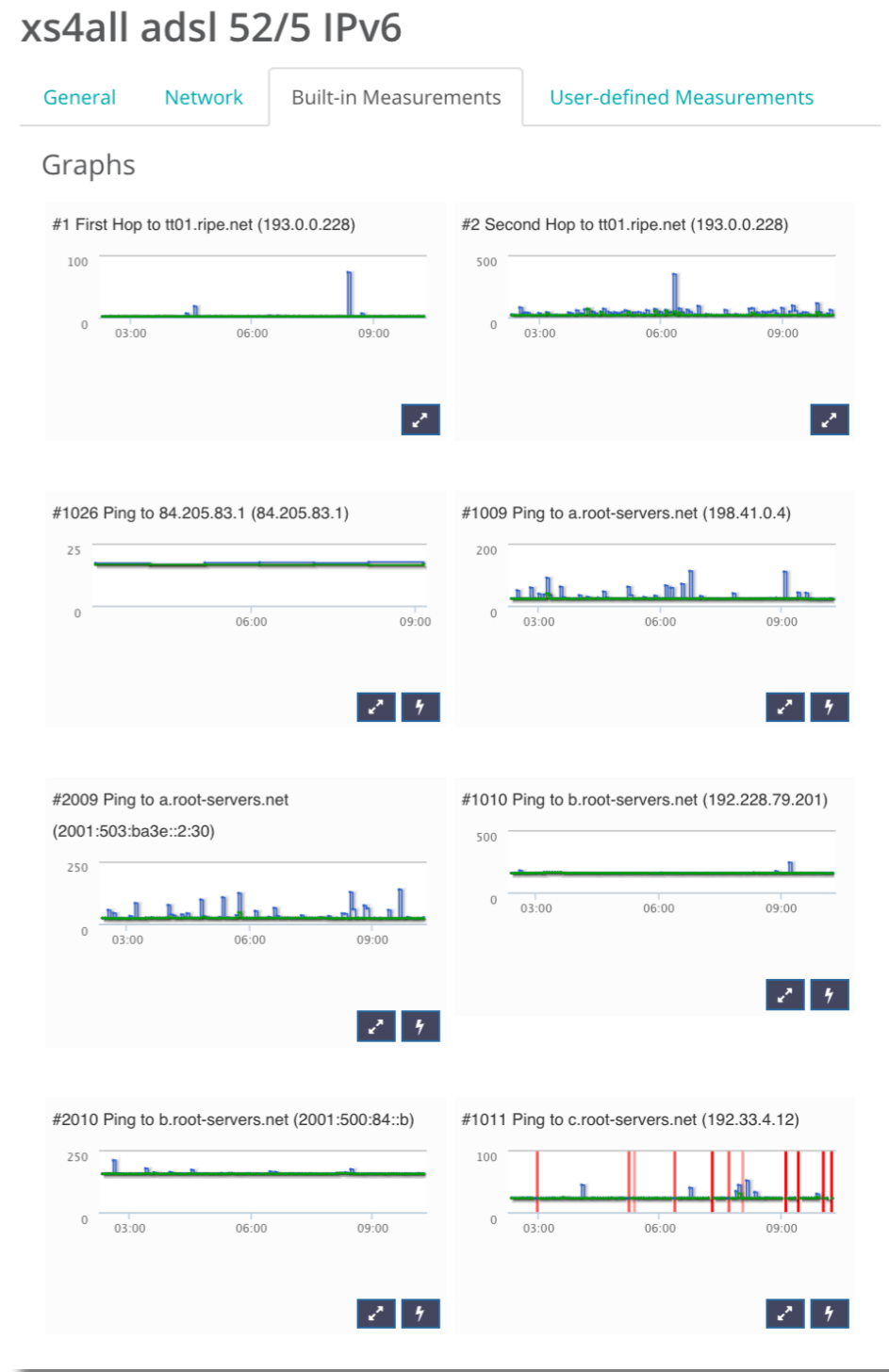
Metadata

3 weeks
Firmware 4700
Architecture probev1
MAC Address 00:20:4A:C8:24:4A

The displayed location is approximated to protect the probe host's privacy.



- Built-in measurements (more in next section)
- Shows RTT to fixed set of destinations (mostly DNS root-servers)
- Red = packet loss



- “Super probes” and targets - 1U servers
- <https://atlas.ripe.net/anchors/list/>
- <https://atlas.ripe.net/anchors/map/>



DEMO



RIPE Atlas Measurements



- A set of RIPE Atlas probes performing a specific type of measurement towards a single destination
- Destination can be IP or hostname
 - Hostname resolution by RIPE Atlas infrastructure (default) or on RIPE Atlas probe
- Single IP protocol: either IPv4 or IPv6
- Measurement types:
 - ping, traceroute, DNS, SSL, NTP, (HTTP)

- Built-in measurements:
 - Root name servers, RIPE Atlas infrastructure
 - <https://atlas.ripe.net/docs/built-in/>
- Anchoring measurements:
 - Towards RIPE Atlas anchors
- User-defined measurements
 - You define the measurement
 - Lots of parameters:
 - Packet size, number of packets, resolve on probe
 - <https://atlas.ripe.net/docs/udm/>

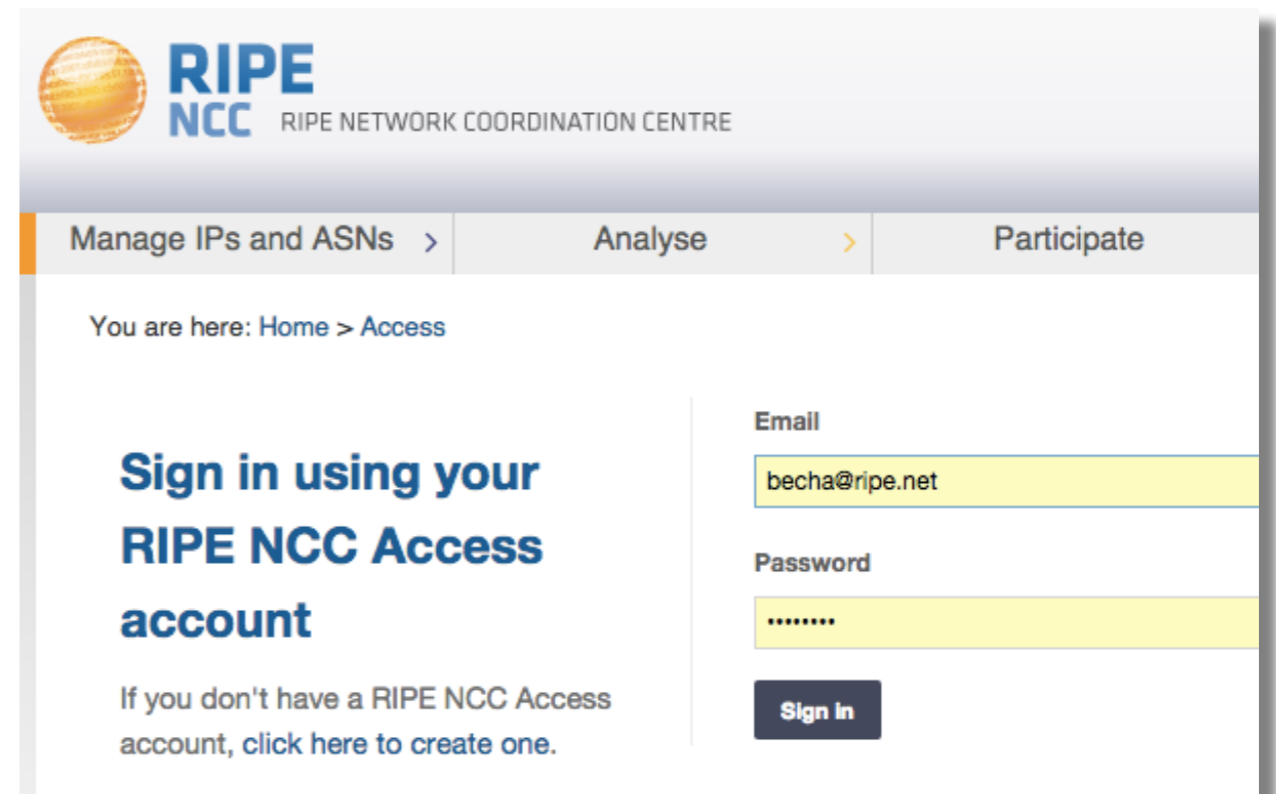
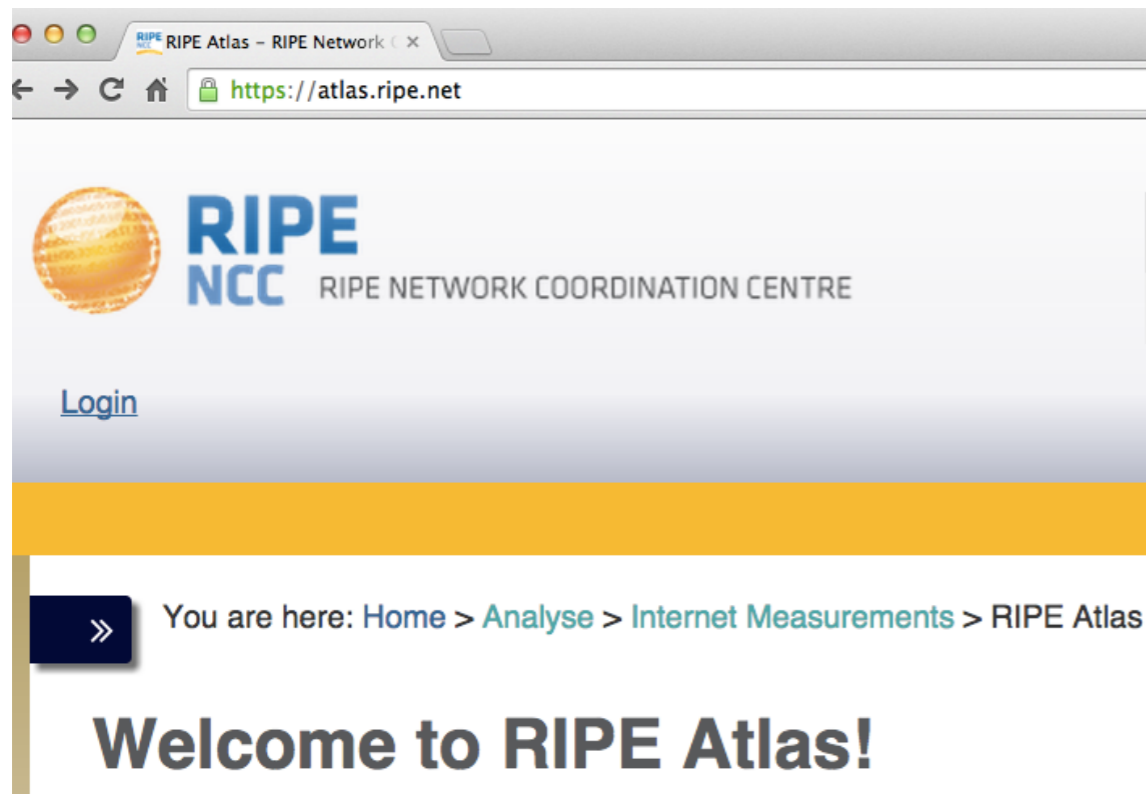


Finding Results of Public Measurements



- Apart from built-in measurements, users have created a lot of measurements already!
- Search for existing public measurements first
 - Even if you do not host a probe!
- Schedule your own measurement if you don't find what you're looking for
 - You'll need credits for this

- Log in to atlas.ripe.net
 - Use your RIPE NCC Access account
 - Create an account if you don't have one already



- Test login credits (only works today):
 - site: <https://atlas.ripe.net/>
 - user: ***testripeatlas@yahoo.com***
 - password: ***sanog2626***
- 1M credits
 - Who can create the most interesting measurement?

- Direct URL: <https://atlas.ripe.net/measurements/>
- Go to “My Atlas” > “Measurements”

Measurements - RIPE Atlas — RIPE Network Coordination Centre

https://atlas.ripe.net/measurements/

RIPE NCC RIPE NETWORK COORDINATION CENTRE

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 > Measurements

RIPE Atlas <<
About RIPE Atlas >
Get Involved >
Results >
My Atlas >
Probes >
Measurements >
Credits >
API Keys >
Messages (72 new)

Measurements [+ Create a Measurement](#)

Filter by target and/or description Any Status IPv4/v6 **All types** of all time

Mine Favourites Hidden Public All

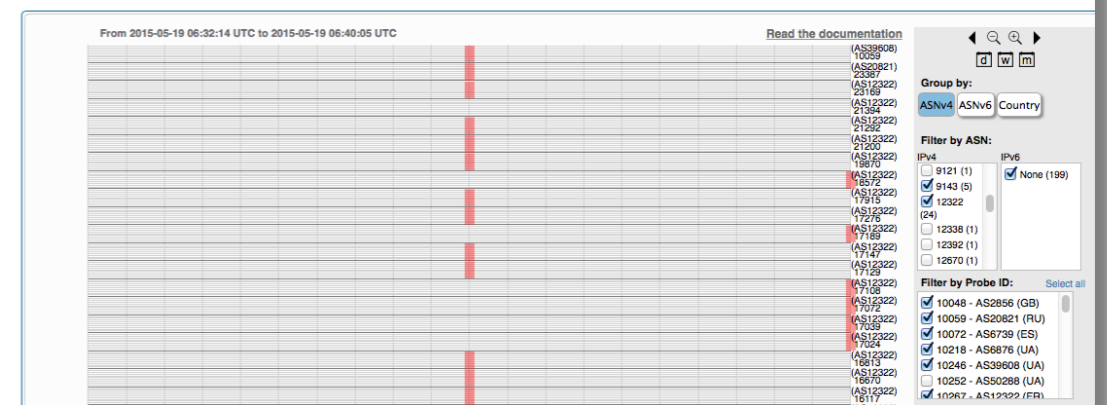
Id	Type	Target	Description	(UTC)	Status
1965015	IPv4 ping	b92.net	Ping measurement to b92.net	49 2015-04-21 08:20 2015-04-21 08:30	■ 👁 ★

- List of probes: sortable by RTT
- Map: colour-coded by RTT
- Seismograph: stacked multiple pings with packet loss

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



An interactive visualisation for ping measurements.



Red = packet loss

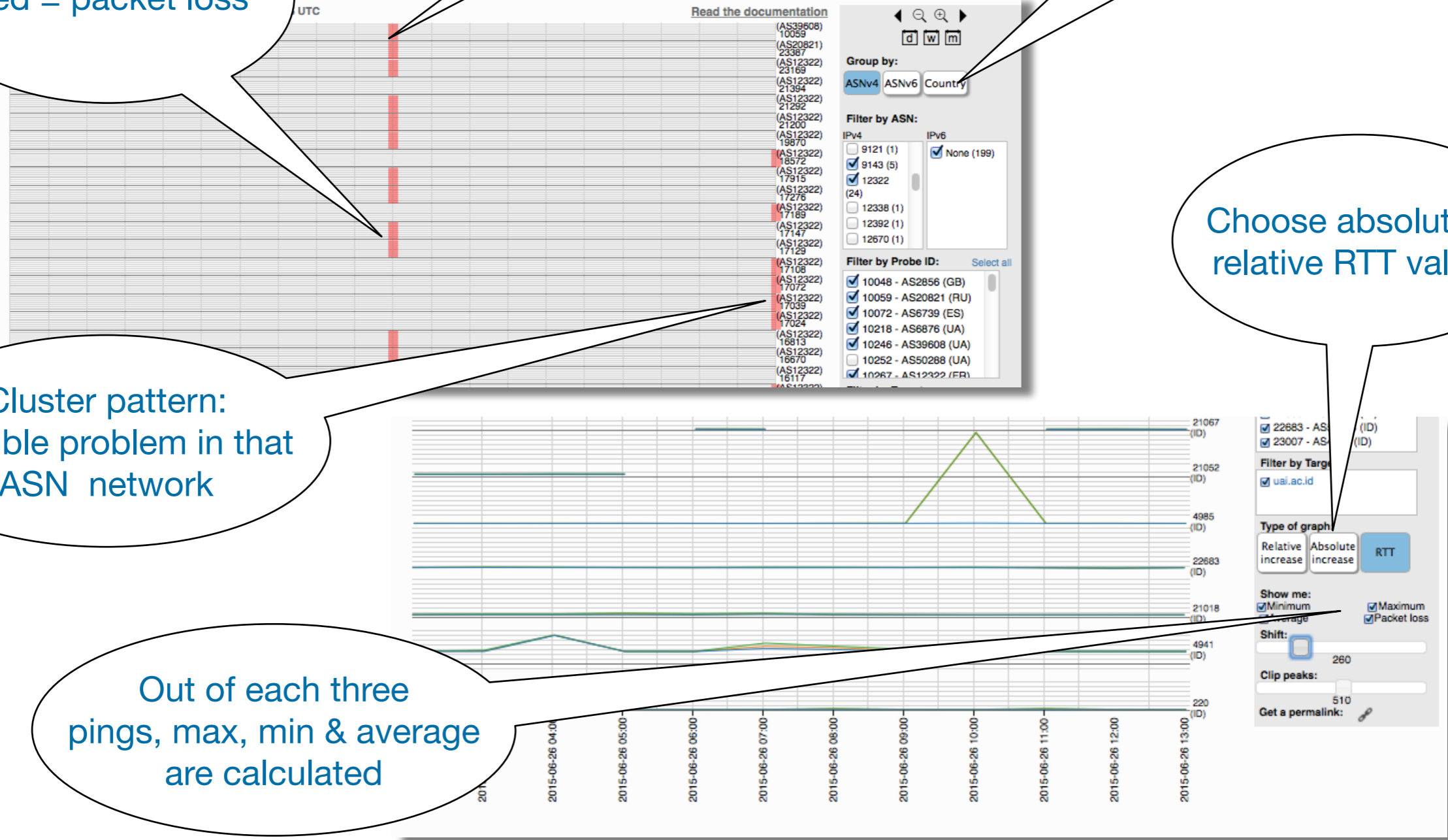
Vertical pattern:
possible problem in
your network

Filtering &
grouping by:
country, IPv4 ASN,
IPv6 ASN















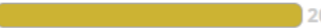



Choose absolute or
relative RTT values

Cluster pattern:
possible problem in that
ASN network

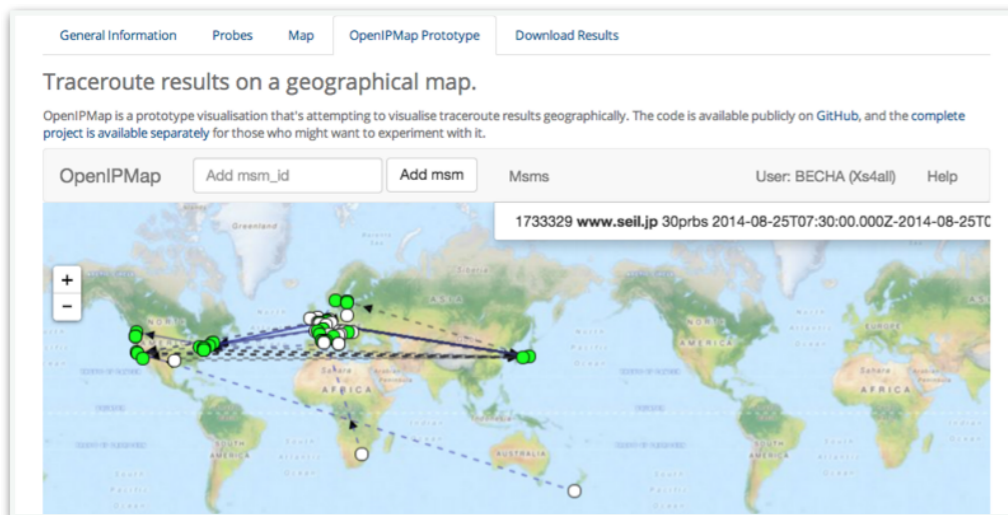
Out of each three
pings, max, min & average
are calculated



- Map
- List of probes, colour-coded number of hops

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

- Traceroute paths map, geolocation using OpenIPMap



- <https://github.com/RIPE-Atlas-Community/openipmap>

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



DNS measurement to ns1.opteamax.de

General Information		Probes	Map	Download Results	Modification Log	
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

- Documentation for analysing measurements results:
 - <https://atlas.ripe.net/docs/rest/>
 - <https://github.com/RIPE-NCC/ripe.atlas.sagan>
- More tools:
 - <https://github.com/RIPE-Atlas-Community>
 - <https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/README.md>

DEMO



Exercise: Analyse Measurement Results



- Download results of a specific public measurement
- Read the text of the result, to understand structure

- Find the measurement
 - ping, IPv6 to google.com
 - msm-ID 1004005
- Click on measurement, then “Download”
 - Specify the time period (e.g. yesterday)
- Results in JSON

- Solution URL:

- <https://atlas.ripe.net/api/v1/measurement/1004005/result/?start=1435104000&stop=1435276799&format=json>

- Save the measurement(s) locally

```
$ curl https://atlas.ripe.net/api/v1/measurement/1004005/result/?start=1435104000&stop=1435276799&format=json > measurement-test.json
```

Task 2: Look at the Result

```
[{"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"}]
```

Reference
(msm ID)

Destination (IP
& name)

Source (probe
public IP address)

Packet loss:
difference between
sent & received!

https://atlas.ripe.net/docs/data_struct/#v4610_ping

- Find out how many times RTT was above 60ms
 - Use Python or JavaScript or something else

- For the JavaScript solution, you can use this as a starting point:
 - https://stat.ripe.net/widgets/demo/script_me.html

Python:

Parse json and find total avg:

```
import json
f = open("measurement.json","r")
measurements = json.load(f)
for m in measurements:
    for r in m["result"]:
        rtt = r["rtt"]
    if rtt > 60: i += 1
```

JavaScript:

```
<script>
var dataAPIUrl = "https://atlas.ripe.net/api/v1/
measurement/1004005/result/?start=1410220800";
jQuery.ajax({

url: dataAPIUrl, error: function() {

alert("error"); },

success: function( response ) { var i = 0;

for ( var i = 0, n = response.length; i < n; i++) { var
measurement = response[i];

for ( var j = 0, m = measurement.result.length; j < m; j++) {
var rtt = measurement.result[j].rtt;
console.log(rtt);
if (rtt > 60)

i++; }

}
jQuery("p").html("The RTT has been above 60ms for " + i
+ " times");

},

dataType: "jsonp" });

</script>
```

```
{
  "af":6,
  "dst_addr":"2001:7fd::1",
  "dst_name":"2001:7fd::1",
  "endtime":1386596919,
  "from":"2001:67c:2e8:13:220:4aff:fec6:cc9d",
  "fw":4570,
  "msm_id":6001,
  "paris_id":4,
  "prb_id":9,
  "proto":"UDP",
  "result": [
    {
      "hop":1,
      "result": [
        {
          "from":"2001:67c:2e8:13::2",
          "rtt":4.7960000000000003,
          "size":88,
          "ttl":64
        },
        {
          "size":40,
          "src_addr":"2001:67c:2e8:13:220:4aff:fec6:cc9d",
          "timestamp":1386596906,
          "type":"traceroute"
        }
      ]
    },
    {
      "hop":2,
```

https://atlas.ripe.net/docs/data_struct/#v4610_traceroute

- Use [ripe.atlas.sagan](#) library to parse abuf

```
{
  "af": 6,
  "dst_addr": "2001:7fd::1",
  "from": "2001:67c:2e8:13:220:4aff:fec6:cd06",
  "fw": 4460,
  "msm_id": 11001,
  "prb_id": 114,
  "proto": "UDP",
  "result": {
    "ANCOUNT": 1,
    "ARCOUNT": 13,
    "ID": 14016,
    "NSCOUNT": 13,
    "QDCOUNT": 1,
    "abuf": "NsCEAAABAAEADQANAAAG <LARGE BUFFER YOU HAVE TO DECODE HERE> ==",
    "rt": 3.4279999999999999,
    "size": 493
  },
  "timestamp": 1339664565,
  "type": "dns"
}
```

https://atlas.ripe.net/docs/data_struct/#v4610_dns



Creating a Measurement



- 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

- Running your own measurements costs credits
 - ping = 3 credits, traceroute = 30, etc.
- Why? Fairness and to avoid overload
- Daily spending limit & max measurements user can create
- Hosting a RIPE Atlas probe earns credits
- Get extra credits by:
 - Being a RIPE NCC member
 - Hosting an anchor
 - Sponsoring probes
 - Credit transfers (ask me!)

- Test login credits (only works today):
 - user: *testripeatlas@yahoo.com*
 - password: *sanog2626*
- 1M credits
 - Who can create the most interesting measurement?

<https://atlas.ripe.net/user/credits/>

- RIPE Atlas <<
- About RIPE Atlas >
- Get Involved >
- Results >
- My Atlas** v
- Probes
- Measurements
- Credits
- API Keys
- Messages (72 new)
- Anchors
- Sponsorships
- Ambassador Probes
- LIR Benefits
 - Claim 1 Million Credits
 - IPv6 Connectivity Test
 - Quick Look

Account Information

This is where you're able to view the history of your credit use. There are visualisations available, and you can also transfer credits to someone else.

40,900
22,931

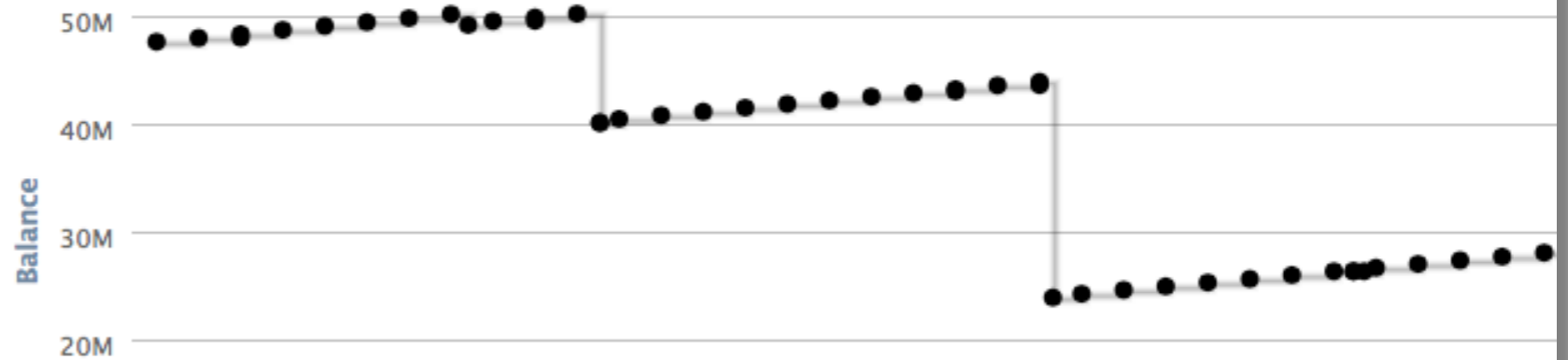
- History
- Charts & Archives
- Transfer

History

My Atlas > Credits

Give credits to someone

Account Balance
Daily account balance over time



- Log in to atlas.ripe.net
- “My Atlas” > “Measurements”
- Three methods:
 1. Quick & Easy
 - Measurement type
 - Target IP/hostname
 - Done! (Default values are used...)
 2. Advanced GUI usage
 3. CLI scripting using API

- Default: periodic, long-term measurement
 - If just once, right away, choose “One-off”
- Choose type, target, frequency, # of probes, region...
 - Improved interactive interface helps you at each step
- You will spend credits

- Using command-line & scripting:
Application Programming Interface (API)
 - <https://atlas.ripe.net/docs/measurement-creation-api/>
- You will need “API keys”
 - <https://atlas.ripe.net/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/docs/credits/>
 - <https://atlas.ripe.net/docs/udm/>
 - <https://atlas.ripe.net/keys/>
 - <https://atlas.ripe.net/docs/keys/>



Exercise: Create a Measurement



- Create a **ping** measurement:
 - Involving ten probes
 - To a target of your choice
 - From SANOG region (or not, you decide)
 - Duration of two days
1. Warm-up: Create a measurement using the GUI
 2. Create API Key
 3. Schedule a measurement using the API

Measurements

Filter by target and/or description Any Status IPv4/v6 All types Of all time

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	■ 👁 ★

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

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

+ 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)

<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 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 www.google.com				✓

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

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

- Example:

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

DEMO

- probe API
 - <https://atlas.ripe.net/docs/rest/#probe>
- measurement API
 - <https://atlas.ripe.net/docs/rest/#measurement>
- probe archive API (historic probe status)
 - <https://atlas.ripe.net/docs/rest/#probe-archive>
- participation API (add probes to existing measurement)
 - <https://atlas.ripe.net/docs/rest/#participation-request>



Integration of RIPE Atlas with Network Monitoring Systems



- Network operators use tools for monitoring network health (e.g. Nagios and Icinga)
- These tools can receive input from RIPE Atlas via the API
- Benefits:
 - pings from 500 out of 8,000+ probes around the world
 - See your network from the outside
 - Plug into your existing practices

1. Create a RIPE Atlas **ping** measurement
2. Go to “Status Checks” URL
3. Add your alerts in Nagios or Icinga



- Status checks work via RIPE Atlas' RESTful API
 - https://atlas.ripe.net/api/v1/status-checks/MEASUREMENT_ID/
- You define the alert parameters, for example:
 - Threshold for percentage of probes that successfully received a reply
 - How many of the most recent measurements to base it on
 - Maximum packet loss acceptable
- Documentation:
 - <https://atlas.ripe.net/docs/status-checks/>

- Community of operators contributed configuration code!
 - Making use of the built-in “check_http” plugin
- GitHub examples:
 - https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/scripts_for_nagios_icinga_alerts
- Post on Icinga blog:
 - <https://www.icinga.org/2014/03/05/monitoring-ripe-atlas-status-with-icinga-2/>



Exercise: Setting up “Status Checks”



- Set up and configure a “status check”
 - For an existing IPv6 ping msm to in-bom-as33480.anchors.atlas.ripe.net
 - <https://atlas.ripe.net/api/v1/status-checks/1849608/>
- Configure the status check in such a way that
 - it will trigger an alert for this measurement
 - it will not trigger an alert for this measurement
- Optional: set-up status check for your own ping measurement!

- Useful parameters:

Argument	Default	Description
max_packet_loss	75	The acceptable percentage packet loss per probe
show_all	false	Show all RTT responses. The default is to only show all responses for alerting probes
permitted_total_alerts	0	The total number of probes you would permit to respond with an alert before a global alert is issued
lookback	1	The total number of measurement results to compare to generate a median RTT value.
median_rtt_threshold	N/A	The threshold at which an alert should be issued when you compare the latest RTT value to the median values (based on the lookback)

- Possible solution:

- <https://atlas.ripe.net/api/v1/status-checks/1849608/>
- https://atlas.ripe.net/api/v1/status-checks/1849608/?permitted_total_alerts=12



Example Application “eyeballtrace”



- Idea: what if I could do a **traceroute** from all of the major “**eyeball**” **networks in a country**?
- Workflow:
 - Extract list of ASNs with $> 1\%$ market share (external API)
 - Find usable RIPE Atlas probes in these ASNs (probe API)
 - Create measurements from these probes (measurement API)
 - Fetch results (streaming API)
 - Provide text-based, annotated (host, ASN for hops) output (RIPEstat / OpenIPMap)

- <https://github.com/emileaben/eyeballtrace>
- command-line: `eyeballtrace -c IN flipkart.com`

```
## AS9829/BSNL-NIB - National Internet Backbone (23.6% of market in IN)
#prb:17011 dst:flipkart.com
1 () 10.10.10.1 [0.377, 0.4, 0.574]
2 () 192.168.1.1 [1.055, 1.142, 1.212]
3 (AS9829) 117.206.176.1 [22.017, 22.512, 22.771]
4 (AS9829) 218.248.171.158 [39.914, 142.812, 188.124]
5 (AS9829) 218.248.235.130 [29.958, 30.113, 31.221]
6 (AS9498) aes-static-233.134.23.125.airtel.in [101.072, 101.42, 102.495]
7 (AS9498) aes-static-021.37.144.59.airtel.in [97.964, 99.393, 117.358]
8 err:{u'x': u''}
8 (AS9498) aes-static-138.126.17.125.airtel.in [73.969]
9 (AS17439) 180.179.165.186 [74.373, 74.472, 76.232]
10 (AS17439) m200.flipkart.com [74.383, 74.604, 74.65]
11 (AS17439) m200.flipkart.com [75.563, 75.934, 141.835]
12 (AS9752) 163.53.76.21 [75.184, 75.495, 76.207]
```

```
## AS17813/MTNL-AP - Mahanagar Telephone Nigam Ltd. (3.1% of market in IN)
NO RIPE Atlas coverage!!
If you are in a position to put a probe in this network: https://atlas.ripe.net/get-involved/become-a-host/
```

DEMO

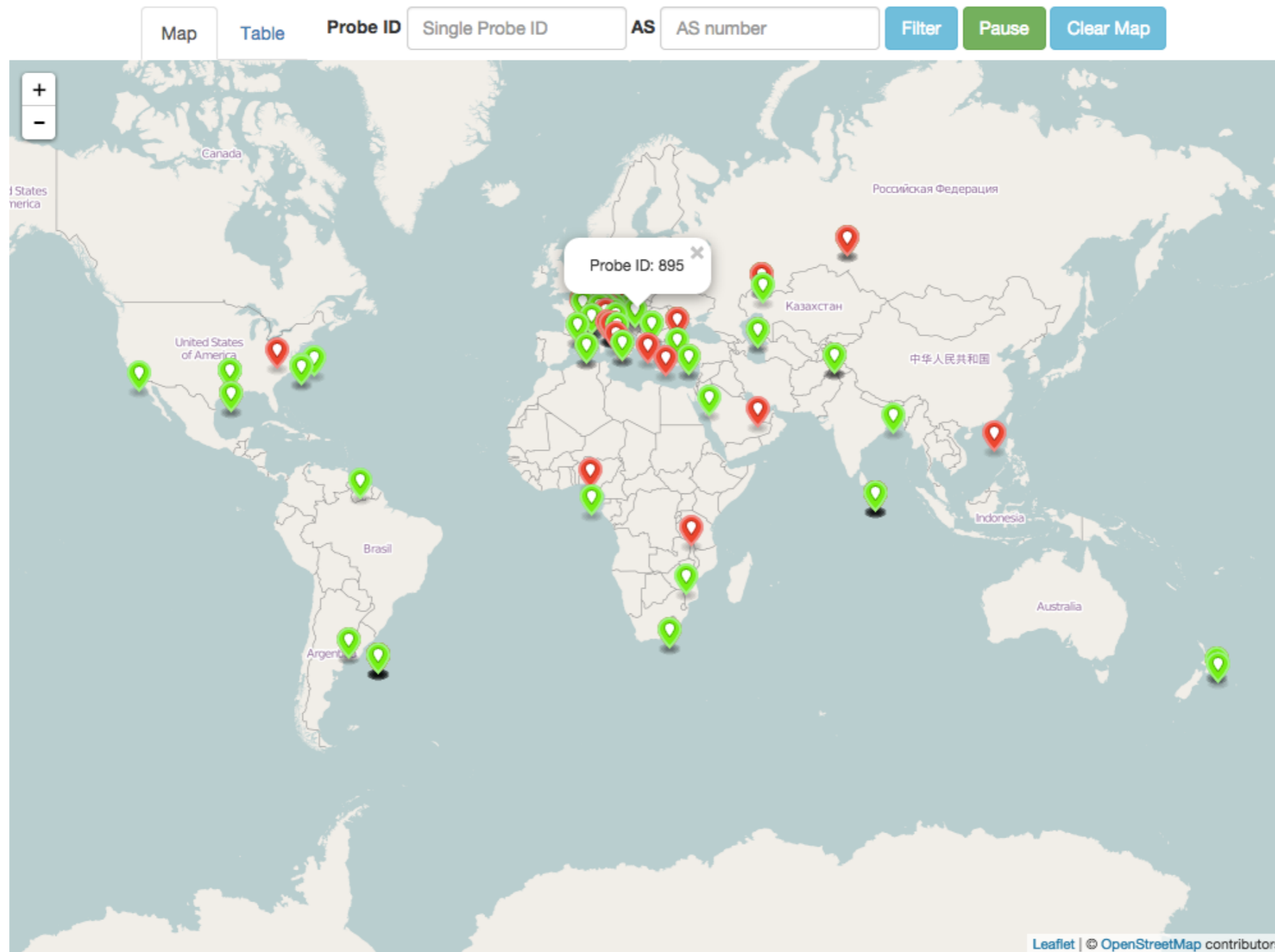


Real-time Performance Monitoring



- **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 (RFC 6455)
- There are two types of data:
 - Measurement results
 - Probe connection status events

- Visualising network outages
- Server and performance monitoring
- In March 2015: used by almost all hackathon teams:
<https://labs.ripe.net/Members/becha/ripe-atlas-hackathon-results>
- Documentation:
 - <https://atlas.ripe.net/docs/result-streaming/>
 - https://labs.ripe.net/Members/suzanne_taylor_muzzin/data-streaming-in-ripe-atlas



https://labs.ripe.net/Members/andreas_strikos/amsterdam-power-outage-as-seen-by-ripe-atlas



Exercise: Using Streaming API



- **Scenario:** Customers are complaining that it occasionally takes a long time to reach your service or server
- **Action:** Ping your server from 500 probes
 - Decide what the acceptable latency threshold is
 - Notice and react when you start receiving samples
- **Task:** Use the ping measurement ID 1791207
 - Choose threshold (e.g. greater than 30ms)
 - Impose threshold on “min” (minimum result of three ping attempts)

1. Go to <https://stat.ripe.net/widgets/workshops/webinar/ripe-atlas/streaming-01.html>
2. Open the development console
3. Wait for results to arrive
4. Optional: Save the HTML file locally and edit the code to your liking

```
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...}
```

- See the connection and disconnection events of all the RIPE Atlas probes in your browser console
- Steps:
 - Create your empty HTML page
 - Connect to the streaming
 - Subscribe to stream_type: “probestatus”

```
<script src="http://atlas-stream.ripe.net/socket.io.js"></script>
<script>
  var socket = io("http://atlas-stream.ripe.net:80", { path : "/stream/socket.io" });

  socket.on("atlas_probestatus", function(status){
    console.log("I received ", status);
  });

  socket.emit("atlas_subscribe", { stream_type: "probestatus" });
</script>
```



Take Part in the RIPE Atlas Community



RIPE
NCC

- 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



- **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



- <https://atlas.ripe.net>
- Roadmap: <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)

- <https://atlas.ripe.net/docs/rest/>
- <https://github.com/RIPE-NCC/ripe.atlas.sagan>
- <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/>
- <https://atlas.ripe.net/docs/keys/>

- Basics:
 - <http://www.ripe.net/lir-services/training/courses/tailor-made-workshops/#tools>
- Webinar material:
 - <https://www.ripe.net/support/training/learn-online/webinars/advanced-ripe-atlas-usage-webinar>
- More tools:
 - <https://github.com/RIPE-Atlas-Community>



Additional Slides



- 8,300+ probes connected
- 5,000+ active users in the last quarter
- 2,500+ results collected per second
- 35,000+ customised measurements weekly
- Five types of customised measurements available:
ping, traceroute, DNS, SSL, NTP

- Individuals can host a probe:
 - Go to <https://atlas.ripe.net/apply>
 - You will receive a probe by post
 - Register your probe
 - Plug in your probe
 - One per ASN!
- For organisations:
 - Host an anchor
 - Sponsor RIPE Atlas
- Help us distribute probes: become an ambassador!

- Well-known targets and powerful probes
 - Regional baseline and “future history”
- Anchoring measurements
 - Measurements between anchors
 - 200 probes targeting each anchor with measurements
 - Each probe measures 4-5 anchors
 - Vantage points for DNSMON service
- 130+ RIPE Atlas anchors



- New measurement type: NTP
 - https://labs.ripe.net/Members/philip_homburg/ntp-measurements-with-ripe-atlas

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)



Measuring Impact of IXPs on Keeping Traffic Local

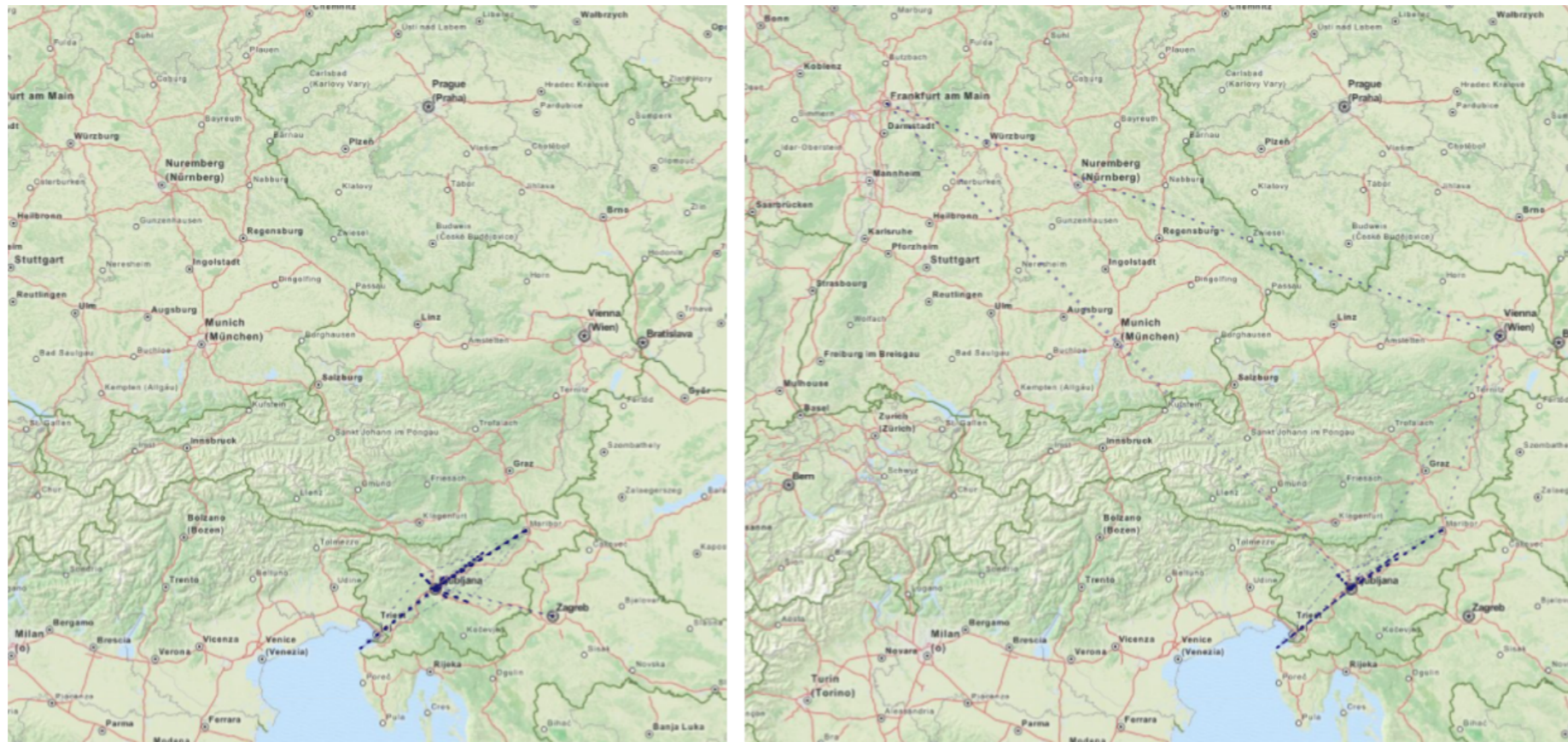
“IXP Country Jedi”



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

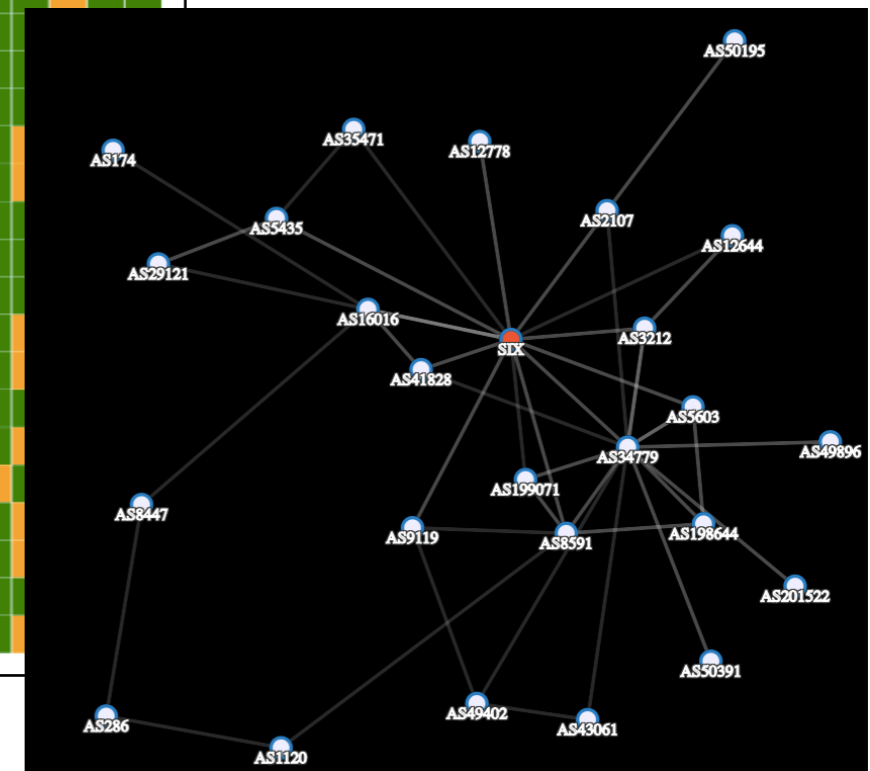
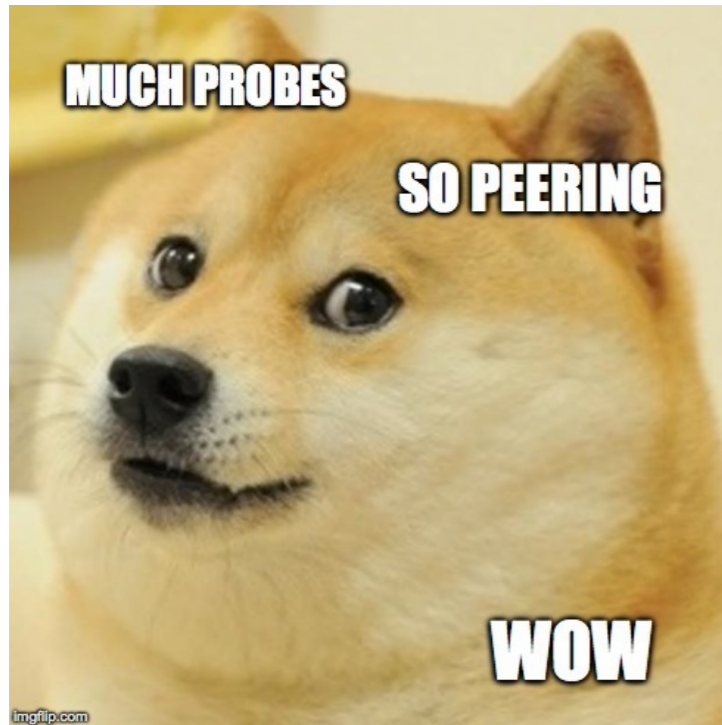
- Difference between IPv4 and IPv6 paths



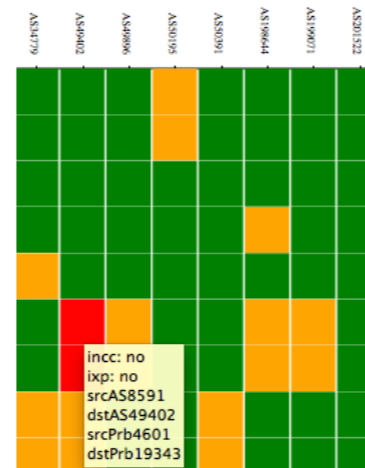
[http://sg-pub.ripe.net/emile/ixp-country-jedi/SI-2015-04/
geopath/s/SI/{RO, BG, HR, BA, ME, AL, GR}/](http://sg-pub.ripe.net/emile/ixp-country-jedi/SI-2015-04/geopath/s/SI/{RO, BG, HR, BA, ME, AL, GR}/)

Paths Via an IXP?

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



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



```
## msm_id:1962254 prb_id:4601 dst:193.169.48.40 ts:2015-04-16 09:01:06 -00:00
1 (AS8591) maribor10-ge-2-20-v987.amis.net [1.593, 1.602, 2.292] |Maribor,Maribor,SI
2 (AS8591) mx-mbl-te-1-2-0.amis.net [1.619, 1.697, 1.944] ||
3 (AS8591) mx-ljl-te-2-3-1.amis.net [3.599, 3.865, 5.148] ||
4 (AS8591) mx-zgl-xe-2-0-1.amis.net [5.568, 5.576, 5.69] ||
5 () 75.64-127.15.192.193.in-addr.arpa [5.955, 5.98, 5.985] |Zagreb,Grad Zagreb,HR|
6 (AS9119) 212.13.240.249 [5.778, 5.83, 5.935] ||
7 (AS9119) 212.103.133.4 [7.099, 7.84, 7.926] ||
8 (AS9119) 212.13.240.62 [6.597, 7.674, 7.696] ||
9 (AS9119) hsl.gw0.hsl.eu [5.833, 6.079, 6.368] ||
10 (AS49402) ntp.hsl.eu [6.657, 7.273, 8.155] ||
11 (AS49402) 193.169.48.40 [6.661, 6.691, 6.872] ||
```

- Green: “good”, as far as we can see it
 - Not a judgment, only one way of visualising data
- Red or blue: path is going out of country
 - If this is a surprise: talk to your upstream(s)
- Yellow: path is not going via a local IXP
 - If this is undesired: make a new peering agreement

- 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 ANS and only “public” probes with “good” geolocation
- Hops geolocated using “OpenIPMap” database

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

- Re-use 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>