NZRS
an InternetNZ company

Using RIPE Atlas for network measurement and monitoring

Sebastian Castro, NZRS

(Material kindly provided by RIPE NCC)

APRICOT, Auckland, February 2016

Goals

- Learn how to:

  Benefit from using RIPE Atlas measurements for network monitoring and troubleshooting

  Use API calls to create measurements

  Integrate RIPE Atlas with existing monitoring systems
- Opportunity for hands-on practice

Prerequisites

- I assume you have already used RIPE Atlas
- Do you have a RIPE NCC Access account?
  - **If not - quickly create one: ripe.net/register**
- Do you have credits to spend?
  - **If not - tell me to transfer some credits**

Warm-up question

- What is your background?
  network operator?
  software engineer / programmer?
  data scientist?
  sysadmin?
  other? please specify :-)

Overview

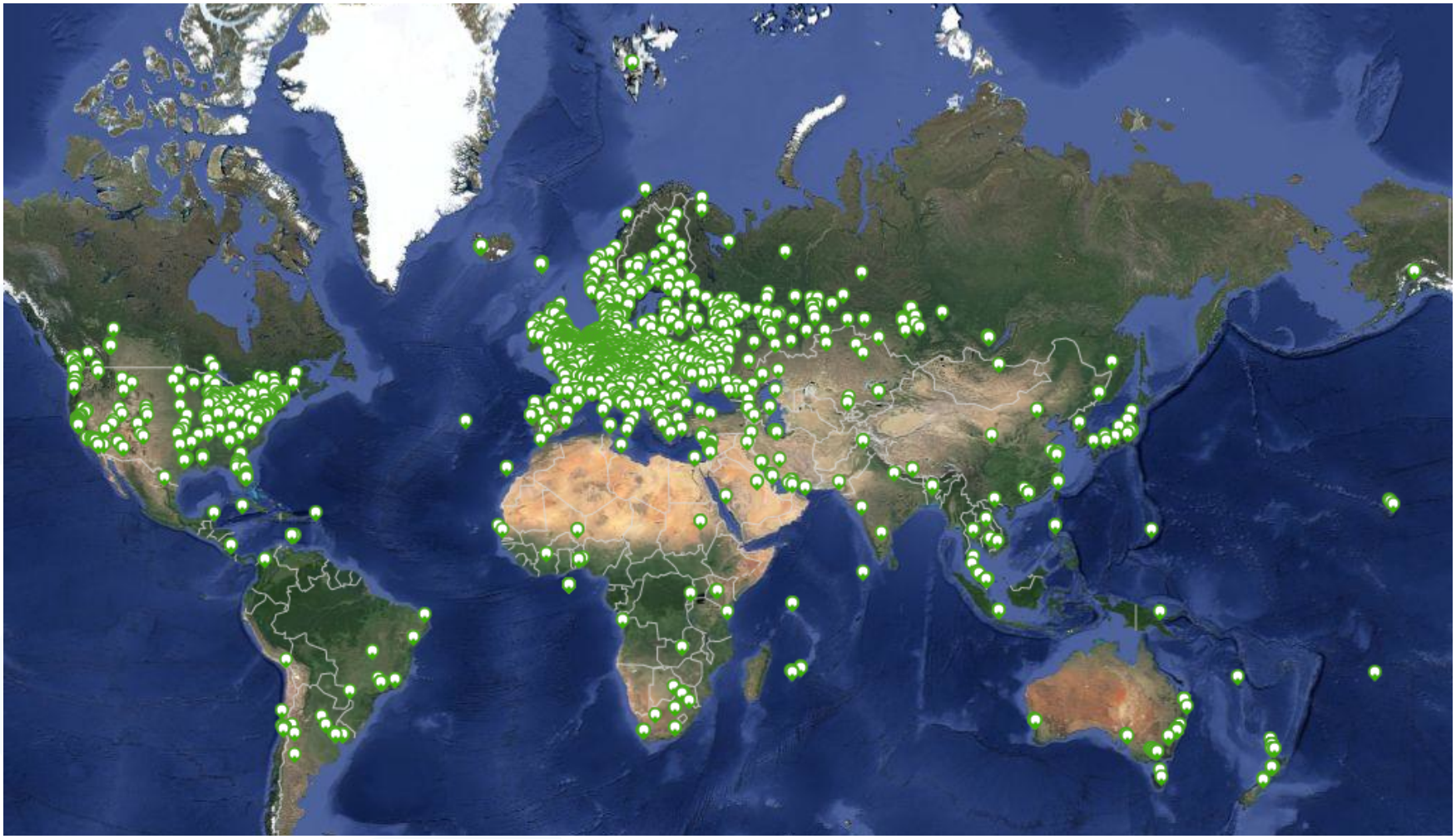
- Introduction to RIPE Atlas
- Creating measurements
- Integration with network monitoring systems
- Real-time performance monitoring
- Take part in the RIPE Atlas community
- News:
  - CLI Toolset
- Additional slides

Introduction to RIPE Atlas

Introduction

- RIPE Atlas is a global active measurements platform
- Goal: view Internet reachability
- Probes hosted by volunteers
  Small probes
  Anchors
- Data publicly available

RIPE Atlas coverage



RIPE Atlas results

- Ongoing global measurements towards root nameservers
  - Visualised as Internet traffic maps
- Ongoing regional measurements towards “anchors”
- Users can run customised measurements
  - ping, traceroute, DNS, SSL/TLS and NTP

Numbers for January 2016

- 9,200+ probes connected

- 5,000+ active users in the last quarter
- 2,500+ results collected per second
- 35,000+ customised measurements weekly

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
- Identify DNS cache poisoning/lies

Credits system

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

Credits overview

RIPE Atlas

About RIPE Atlas

Get Involved

Results

My Atlas

Probes

Measurements

Credits

API Keys

Messages (72 new)

Anchors

Sponsorships

Ambassador Probes

LIR Benefits

Claim 1 Million Credits

IPv6 Connectivity Test

Quick Look

My Atlas > Credits

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.

40,90

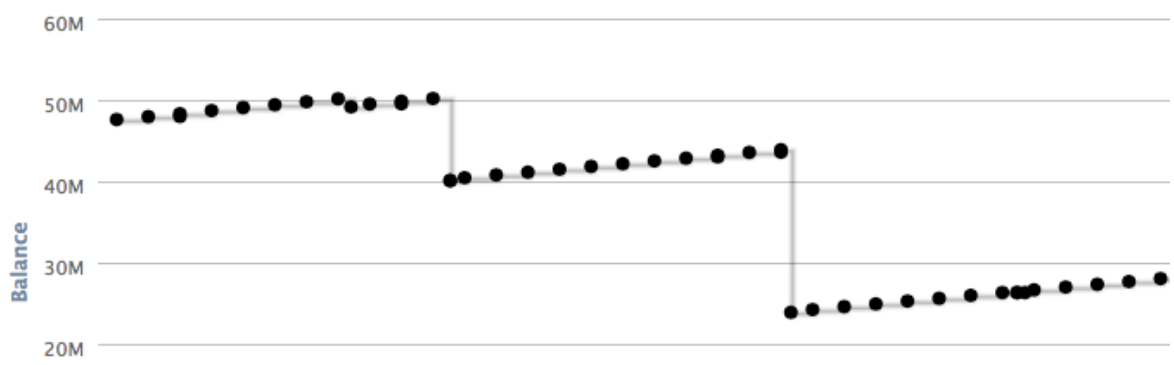
22,931

Charts & Archives

Give credits to someone

Account Balance

Daily account balance over time



Scheduling a measurement

- Log in to atlas.ripe.net
- “My Atlas” > “Measurements”
- Three methods:

1. Quick & Easy
    - Type
    - Target
    - Done!
2. Advanced GUI usage
3. CLI scripting using API

2: Using GUI to schedule a measurement

- Mostly used for a periodic, long time 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 too

3: Using API to schedule a measurement

- Using command-line & scripting:
  Application **P**rogramming **I**nterface (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/

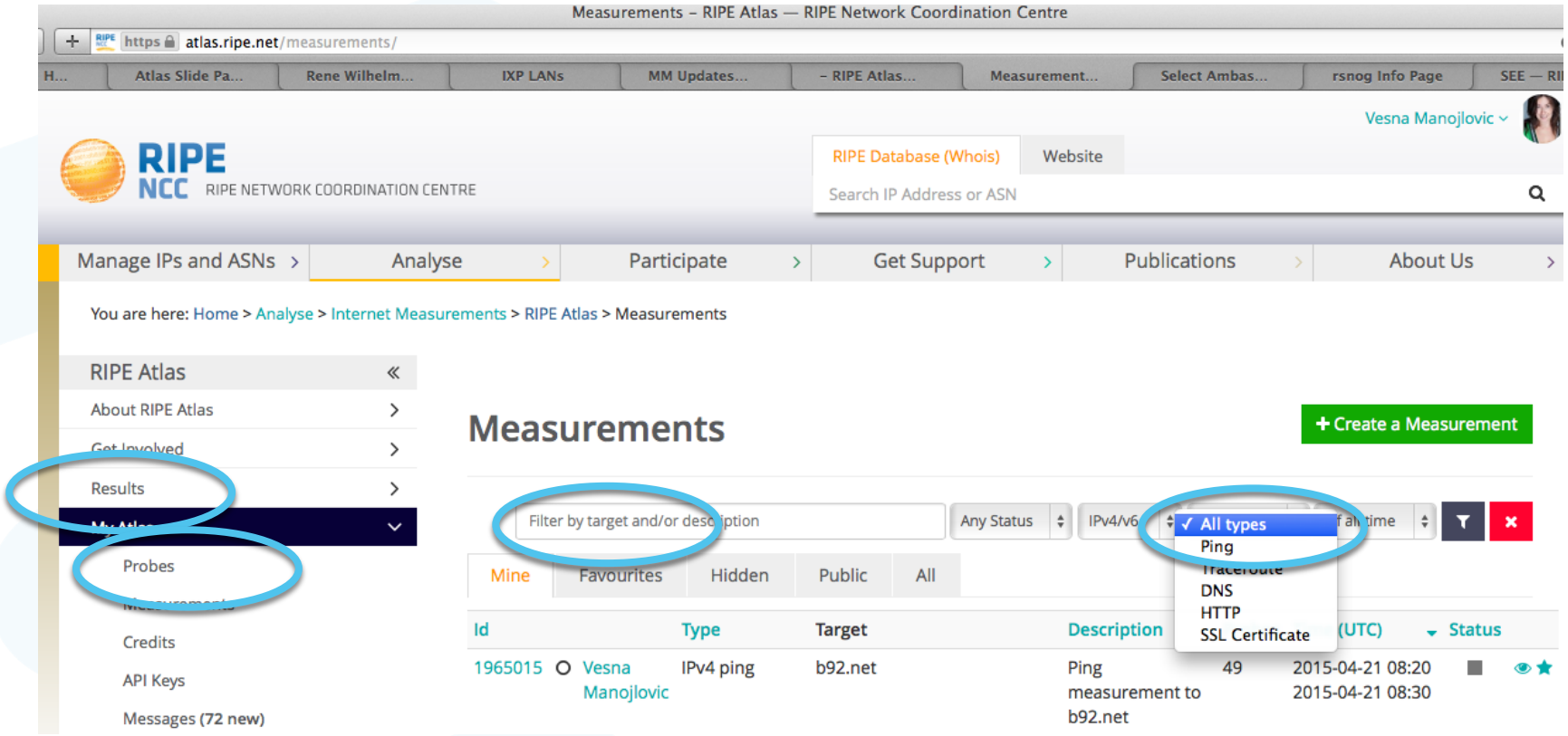
https://atlas.ripe.net/doc/credits

https://atlas.ripe.net/doc/udm

- https://atlas.ripe.net/keys/
- https://atlas.ripe.net/docs/keys2/

Looking up measurements results

- Go to “My Atlas” > “Measurements”

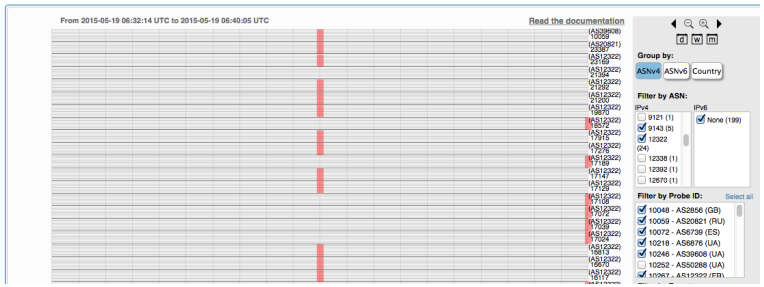


Available visualisations: ping

- 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.



Exercise

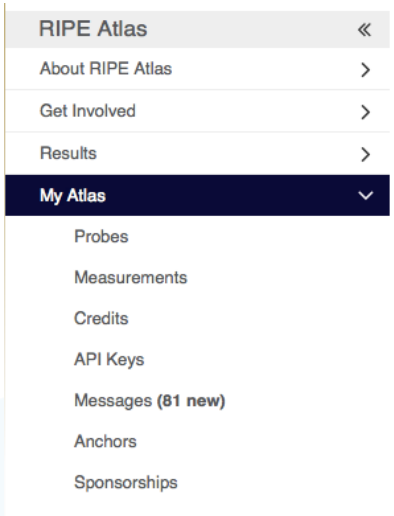
Create a Measurement

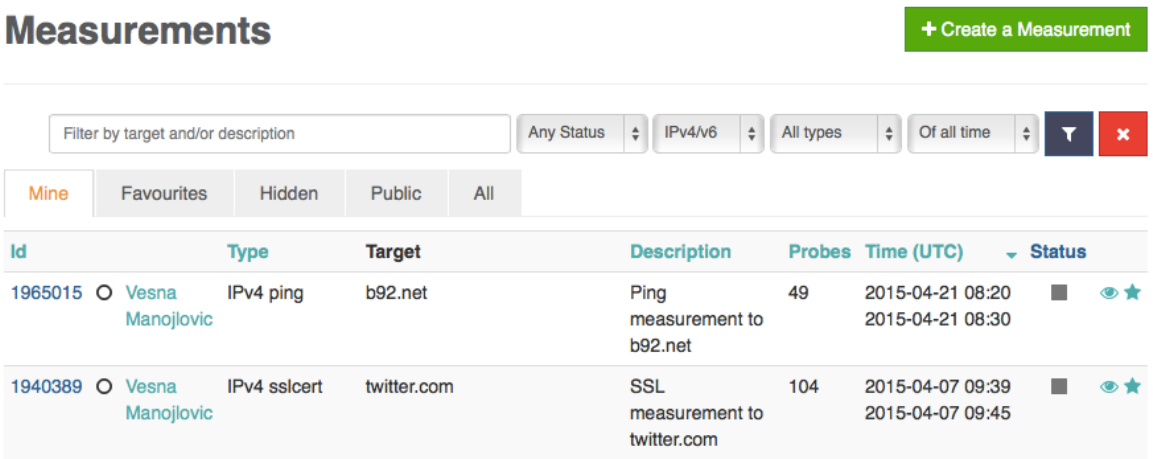
Tasks

- Create a **ping** measurement:
  - Involving ten probes
  - To a target of your choice
  - Source is your country
  - Duration of two days

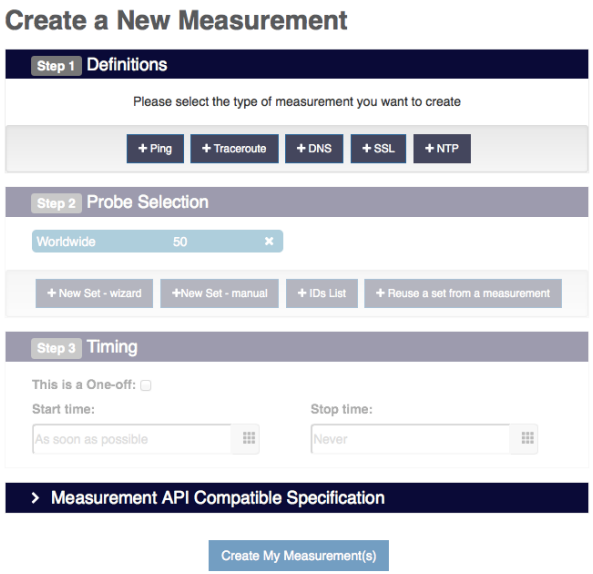
1. Warm-up: Create a measurement using the GUI
2. Create API Key
3. Schedule a measurement using the API

Sub-task 1: Use web interface

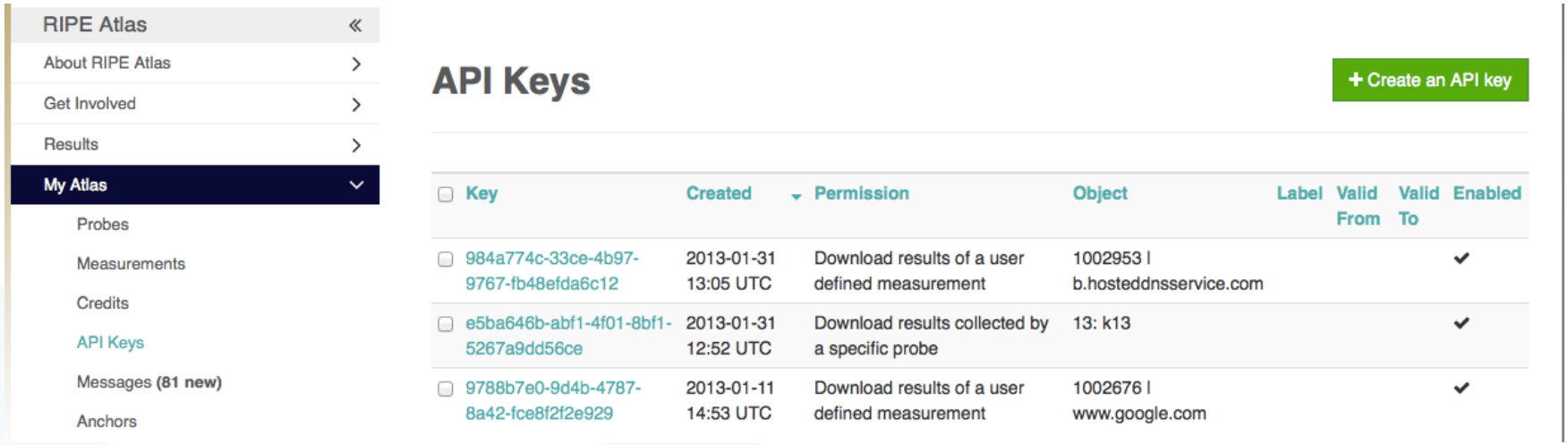




- Useful hint: once you generate a measurement, copy “API Compatible Specification” to text file
- Note Measurement-ID



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

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

- Example:

```
$ 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
```

Example with command line

```
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$
```

Integration with Network Monitoring Systems

Network monitoring

- 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

Steps for integration

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

Creating status checks

- 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
  The maximum packet loss acceptable
- Documentation: https://atlas.ripe.net/docs/status-checks/

Icinga examples

- 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”

Tasks

- Set up and configure a “status check”
  For an existing ping measurement
  https://atlas.ripe.net/measurements/3369314/ Hint: https://atlas.ripe.net/api/v1/status-checks/3369314/
- 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/3369314/?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"]},
```

Real-time performance monitoring

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 two types of data:

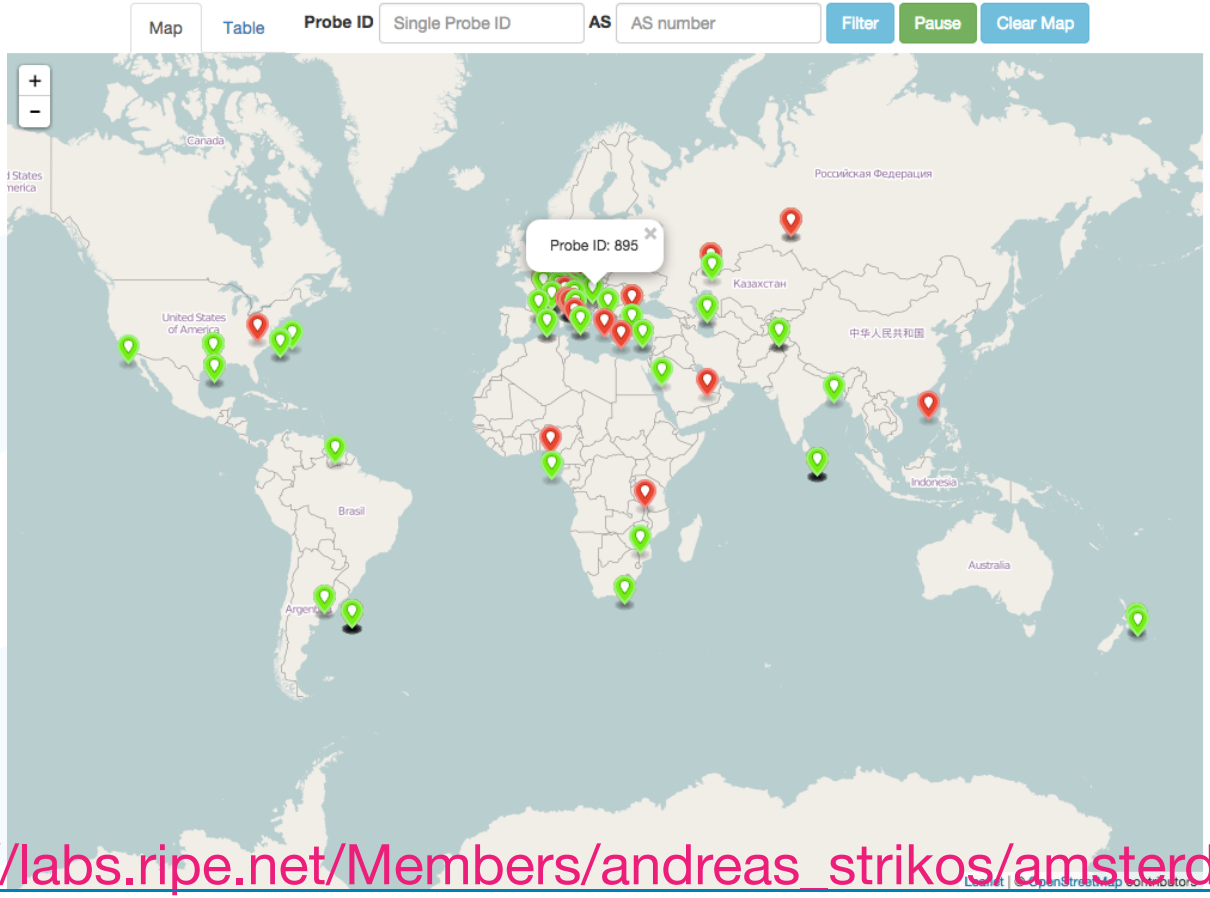
  Measurement results

  Probe connection status events

Real time streaming uses

- 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

Probe (dis)connection events



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

Exercise

Using streaming API

Monitoring server reachability

- 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 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. Optional: Save the HTML file locally and edit the code to your liking

Page Source

view-source:sg-pub.ripe.n

view-source:sg-pub.ripe.net/webinar/streaming01.html

```
<!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 coming from all the probes involved in the measurement 2340408
        socket.emit("atlas_subscribe", { stream_type: "result", msm: 2340408 });

    </script>
</html>
```

Example of results

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/?EIO=2&transport=polling&t=1431095373684-0".
XHR finished loading: GET "http://atlas-stream.ripe.net/stream/socket.io/?EIO=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…}
```

Task 2: View (dis)connect events

See in the console of your browser the connection and disconnection events of all the RIPE Atlas probes

Steps:

- Create your empty HTML page
- Connect to the streaming
- Subscribe to stream_type: “probestatus”

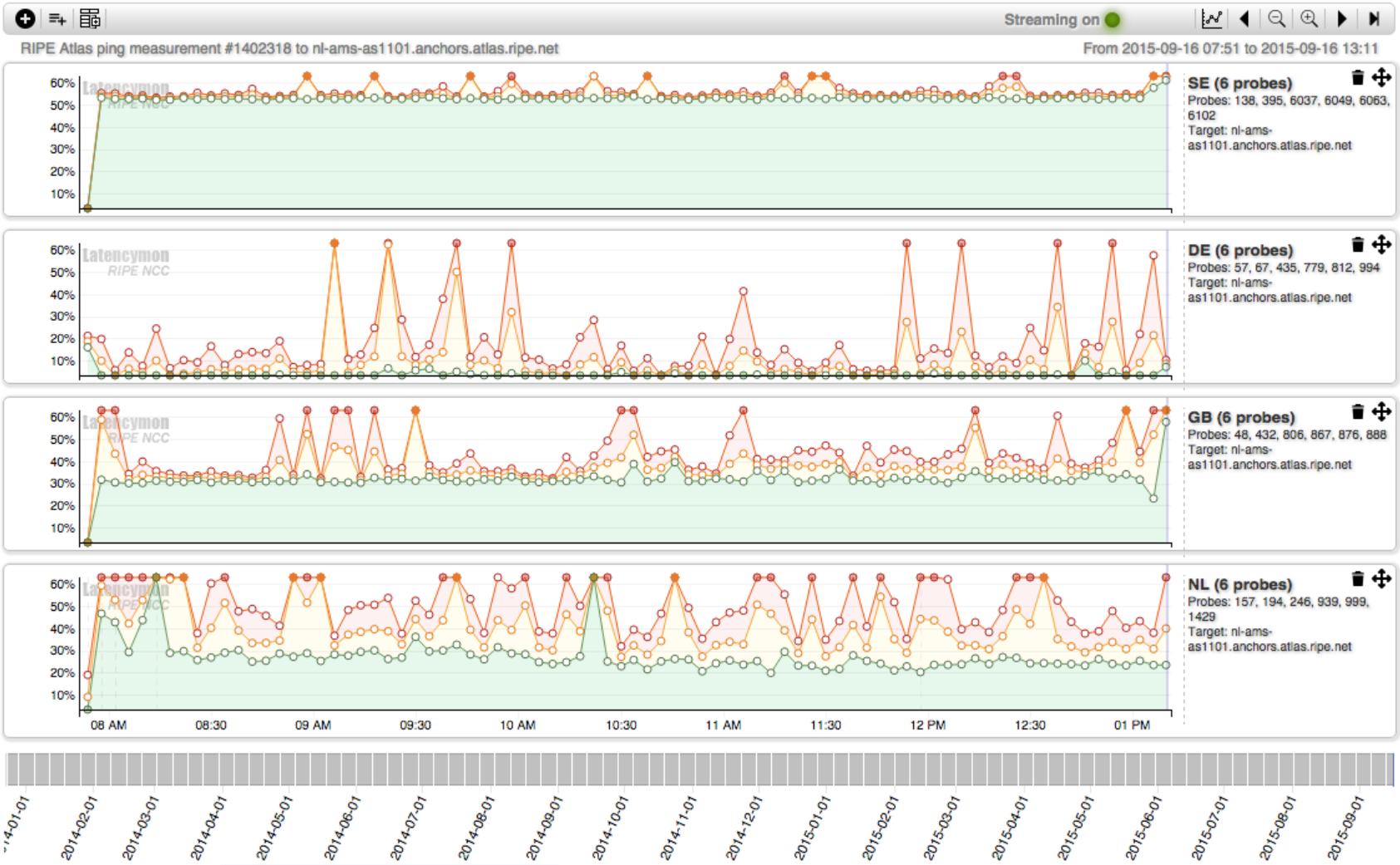
Solution

```
<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>
```

The Newest Features

LatencyMON

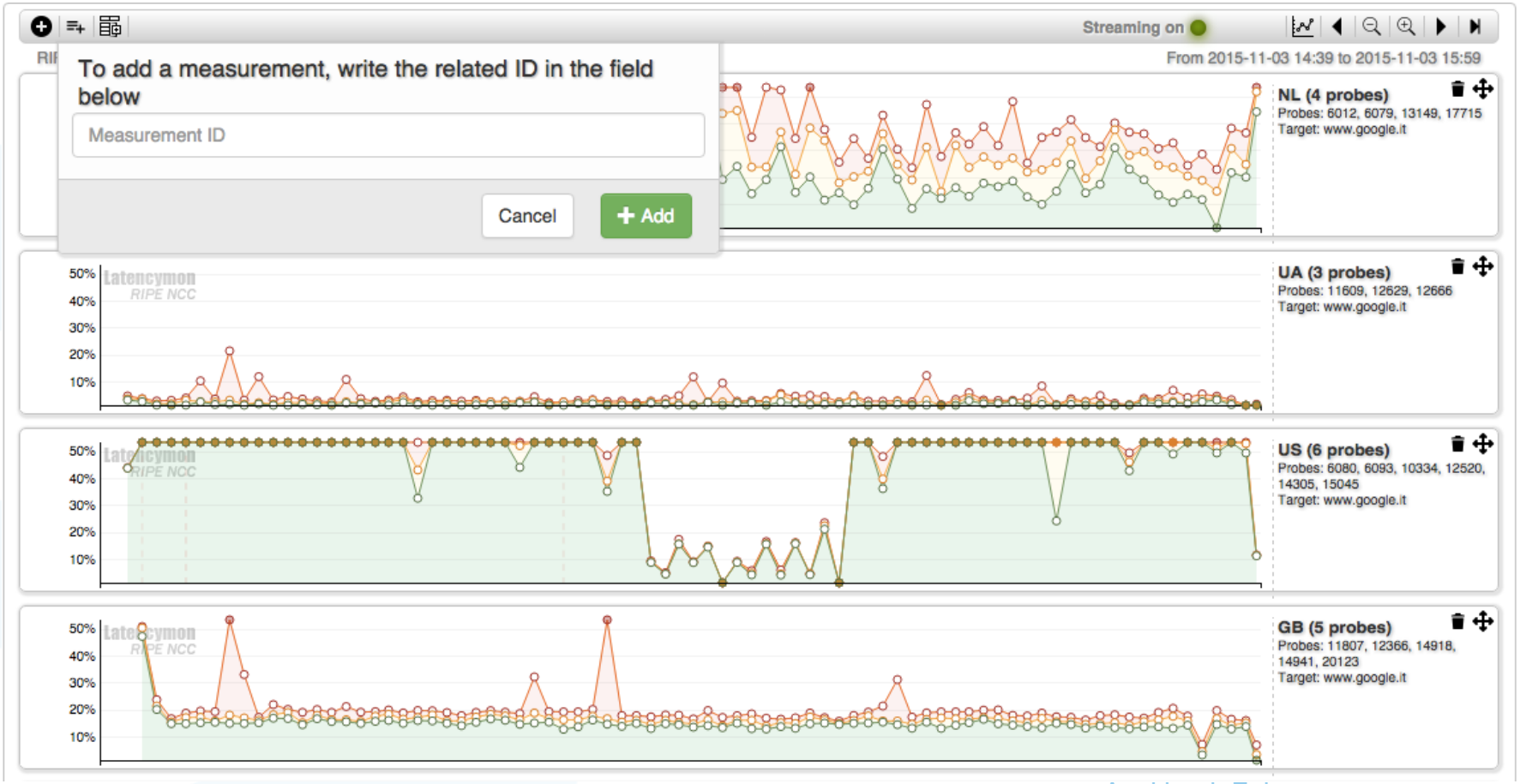
LatencyMON



Source: RIPE NCC

Adding multiple measurements

- If multiple targets are involved, the auto-grouping will be by target.



Source: RIPE NCC

Adding a group of probes

- You can search for any probe attribute
- You can specify a group name

Select the probes you want to add to the group. NOTE: you can use only probes not participating in other charts

Search

	Probe ID	Country	ASv4	ASv6	IPv4	IPv6	Measurement ID
	57	DE	20621	20621	217.69.64.206	2001:aa8:fffe:3:220:4aff:fec8:2098	1402318
	67	DE	31334	31334	95.90.204.77	2a02:8108:9e40:a0:220:4aff:fec8:249e	1402318
	157	NL	3265	3265	82.95.106.192	2001:981:5e40:1:220:4aff:fec8:20b9	1402318
	194	NL	39309	39309	88.159.164.218	2a01:670:6aa4:da00:220:4aff:fec8:2099	1402318
	226	AU	4739	4739	203.16.208.142	2001:44b8:1121:1a00:220:4aff:fec8:245d	1402318
	239	SN	8346	8346	196.1.95.16	2001:4278:1000:1::16	1402318
	246	NL	6830	-	77.251.180.141	-	1402318
	333	JP	17676	17676	126.72.61.194	2400:2410:20c0:4400:220:4aff:fec8:242e	1402318

Showing 1 to 8 of 426 rows

<< < 1 2 3 4 5 > >>

Group name: Group name

Cancel

+ Group

Command Line Interface (CLI) Toolset

Why CLI RIPE Atlas tools

- Access RIPE Atlas from the terminal / shell console
- Quick & dirty shortcuts for network troubleshooting
- FLOSS (open source) tools
  written & maintained by RIPE NCC
  open for contributions by the community
- Before you can use the toolset
  download the tools
  install
  configure

Use cases for CLI tools

1. create a measurement
2. generate a simple report abut ongoing measurement
3. look at the results
4. collect results from the ongoing measurement (streaming)

Creating ping measurement

- Simplest: one-off, using default values: (50 probes to “target”)

```
ripe-atlas measure ping
--auth API_KEY
--target wikipedia.org
```

```
Terminal
(2015-11-02 14:55:44) $ 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/2904352/


Connecting to stream...

48 bytes from probe #10218 46.151.195.130  to 91.198.174.192 (91.198.174.192): ttl=58 times:46.213,  45.994,  45.972,
48 bytes from probe #22628 109.68.46.106   to 91.198.174.192 (91.198.174.192): ttl=55 times:35.662,  35.331,  35.359,
48 bytes from probe #18566 81.201.61.39    to 91.198.174.192 (91.198.174.192): ttl=59 times:17.79,   18.442,  17.882,
48 bytes from probe #20350 82.113.54.198   to 91.198.174.192 (91.198.174.192): ttl=56 times:22.789,  22.465,  22.666,
48 bytes from probe #17129 88.184.60.220   to 91.198.174.192 (91.198.174.192): ttl=55 times:22.135,  21.617,  21.978,
48 bytes from probe #16893 79.194.220.192  to 91.198.174.192 (91.198.174.192): ttl=57 times:28.603,  27.745,  27.795,
48 bytes from probe #19084 27.122.15.216   to 91.198.174.192 (91.198.174.192): ttl=57 times:227.216, 227.051, 225.372,
48 bytes from probe #16651 81.157.15.155   to 91.198.174.192 (91.198.174.192): ttl=50 times:22.73,   16.033,  16.583,
48 bytes from probe #20511 89.101.106.27   to 91.198.174.192 (91.198.174.192): ttl=52 times:31.829,  31.628,  32.272,
48 bytes from probe #17930 212.158.157.8   to 91.198.174.192 (91.198.174.192): ttl=53 times:24.271,  24.009,  33.372,
48 bytes from probe #20372 82.208.31.38    to 91.198.174.192 (91.198.174.192): ttl=55 times:19.358,  19.555,  20.386,
48 bytes from probe #14808 212.59.56.113   to 91.198.174.192 (91.198.174.192): ttl=57 times:15.564,  16.186,  15.162,
48 bytes from probe #18607 185.75.138.141  to 91.198.174.192 (91.198.174.192): ttl=47 times:25.74,   24.893,  24.948,
48 bytes from probe #18068 93.88.224.8     to 91.198.174.192 (91.198.174.192): ttl=60 times:17.606,  17.66,   17.785,
48 bytes from probe #24599 87.104.31.93    to 91.198.174.192 (91.198.174.192): ttl=56 times:11.439,  11.212,  11.306,
48 bytes from probe #17515 83.105.126.208  to 91.198.174.192 (91.198.174.192): ttl=55 times:39.78,   39.492,  39.796,
48 bytes from probe #22511 65.190.176.14   to 91.198.174.192 (91.198.174.192): ttl=47 times:131.451, 125.54,  122.173,
48 bytes from probe #19139 162.226.230.169 to 91.198.174.192 (91.198.174.192): ttl=47 times:122.889, 121.986, 122.019,
48 bytes from probe #16713 90.32.217.67    to 91.198.174.192 (91.198.174.192): ttl=49 times:47.695,  45.995,  47.142,
48 bytes from probe #13414 86.184.50.10    to 91.198.174.192 (91.198.174.192): ttl=53 times:16.523,  16.87,   16.877,
48 bytes from probe #18444 188.175.7.132   to 91.198.174.192 (91.198.174.192): ttl=55 times:24.727,  24.534,  24.551,
48 bytes from probe #20623 94.199.192.180  to 91.198.174.192 (91.198.174.192): ttl=57 times:26.421,  26.102,  26.383,
48 bytes from probe #11334 212.185.87.38   to 91.198.174.192 (91.198.174.192): ttl=57 times:16.389,  16.317,  16.263,
48 bytes from probe #16771 89.24.189.186   to 91.198.174.192 (91.198.174.192): ttl=51 times:33.142,  33.336,  33.39,
48 bytes from probe #16168 84.242.85.91    to 91.198.174.192 (91.198.174.192): ttl=52 times:31.078,  30.997,  31.459,
48 bytes from probe #22608 207.38.156.228  to 91.198.174.192 (91.198.174.192): ttl=54 times:91.974,  91.746,  90.971,
48 bytes from probe #17915 82.230.179.223  to 91.198.174.192 (91.198.174.192): ttl=58 times:67.069,  65.529,  67.286,
48 bytes from probe #23879 84.28.4.6       to 91.198.174.192 (91.198.174.192): ttl=59 times:201.169, 16.039,  9.765,
48 bytes from probe #11643 213.46.66.217   to 91.198.174.192 (91.198.174.192): ttl=58 times:9.102,   8.897,   7.653,
48 bytes from probe #18433 94.112.176.45   to 91.198.174.192 (91.198.174.192): ttl=50 times:40.459,  39.253,  38.474,
48 bytes from probe #18635 92.225.9.253    to 91.198.174.192 (91.198.174.192): ttl=52 times:36.214,  35.388,  35.011,
48 bytes from probe #20236 94.112.19.136   to 91.198.174.192 (91.198.174.192): ttl=52 times:32.036,  31.315,  31.616,
48 bytes from probe #13627 5.83.189.162    to 91.198.174.192 (91.198.174.192): ttl=57 times:9.078,   8.865,   9.046,
48 bytes from probe #25047 112.118.195.171 to 91.198.174.192 (91.198.174.192): ttl=57 times:316.169, 316.591, 316.473,
48 bytes from probe #23411 124.198.160.1   to 91.198.174.192 (91.198.174.192): ttl=51 times:264.655, 265.057, 264.989,
48 bytes from probe #14974 193.104.37.249  to 91.198.174.192 (91.198.174.192): ttl=58 times:22.11,   10.32,   10.408,
48 bytes from probe #22573 89.176.43.44    to 91.198.174.192 (91.198.174.192): ttl=52 times:32.782,  32.476,  32.372,
48 bytes from probe #17189 88.166.162.43   to 91.198.174.192 (91.198.174.192): ttl=57 times:48.633,  47.839,  48.979,
48 bytes from probe #16265 37.24.137.141   to 91.198.174.192 (91.198.174.192): ttl=54 times:21.328,  21.949,  21.781,
48 bytes from probe #11213 212.67.89.66    to 91.198.174.192 (91.198.174.192): ttl=56 times:18.134,  18.063,  18.072,
48 bytes from probe #13580 86.181.90.51    to 91.198.174.192 (91.198.174.192): ttl=53 times:11.823,  11.668,  11.498,
48 bytes from probe #20560 193.86.23.19    to 91.198.174.192 (91.198.174.192): ttl=55 times:42.168,  67.495,  54.532,
48 bytes from probe #16270 188.66.77.211   to 91.198.174.192 (91.198.174.192): ttl=54 times:16.95,   17.16,   16.886,
48 bytes from probe #24314 173.35.72.65    to 91.198.174.192 (91.198.174.192): ttl=52 times:108.039, 121.712, 106.547,
48 bytes from probe #18898 71.179.141.140  to 91.198.174.192 (91.198.174.192): ttl=250 times:97.083,  98.468,  97.011,
48 bytes from probe #19152 70.42.233.238   to 91.198.174.192 (91.198.174.192): ttl=55 times:165.898, 165.874, 165.801,
48 bytes from probe #11851 185.13.208.200  to 91.198.174.192 (91.198.174.192): ttl=57 times:11.761,  11.593,  11.665,
48 bytes from probe #16731 188.134.205.252 to 91.198.174.192 (91.198.174.192): ttl=53 times:30.634,  30.464,  30.316,
48 bytes from probe #11421 92.77.26.34     to 91.198.174.192 (91.198.174.192): ttl=56 times:23.531,  23.106,  22.65,

Disconnecting from stream

You can find details about this measurement here:

  https://atlas.ripe.net/measurements/2904352/
```

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
```

- Report for the traceroute
  - `ripe-atlas measure traceroute --probes 2 --target google.ca`

```
Terminal
(2015-11-02 19:34:25) $ ripe-atlas measure traceroute --probes 2 --target google.ca

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

  https://atlas.ripe.net/measurements/2904858/


Connecting to stream...


Probe #10663

  1 10.254.251.13      15.411ms     0.457ms     0.419ms
  2 162.217.72.1        2.495ms     3.031ms     2.479ms
  3 172.17.18.82        3.914ms      7.36ms     2.478ms
  4 172.17.17.130       3.618ms      3.84ms     2.294ms
  5 172.17.18.90        3.621ms     4.222ms    25.349ms
  6 172.17.18.2         3.319ms     4.438ms     4.119ms
  7 208.69.43.140       4.092ms     4.234ms      4.22ms
  8 208.69.43.194       4.224ms     5.496ms     3.511ms
  9 208.76.187.13       3.639ms     3.255ms     3.462ms
 10 208.76.185.18       4.829ms     4.297ms     4.091ms
 11 72.14.204.109        6.55ms     6.035ms     6.999ms
 12 209.85.249.5        5.601ms     5.701ms     7.148ms
 13 209.85.246.20       5.724ms     5.858ms     5.886ms
 14 216.239.46.212     72.449ms    55.478ms    54.985ms
 15 216.239.51.168     78.764ms     89.09ms    72.288ms
 16 209.85.244.82       56.77ms    54.386ms    54.447ms
 17 209.85.143.214     71.201ms    71.888ms    71.303ms
 18 209.85.254.250    142.758ms   141.019ms   141.557ms
 19 209.85.253.68     150.371ms    149.74ms   149.984ms
 20 216.239.56.74       153.8ms   153.093ms   153.521ms
 21 216.239.56.83     154.186ms   153.589ms     153.1ms
 22 None             None    ms  None    ms  None    ms
 23 74.125.136.94      153.18ms   152.445ms   152.818ms

Probe #16940

  1 192.168.1.234      17.391ms     0.537ms     0.525ms
  2 95.140.15.46      195.899ms   136.709ms   107.218ms
  3 95.140.15.41       51.503ms   119.094ms   132.108ms
  4 213.200.76.185    131.306ms   326.224ms    38.032ms
  5 89.149.181.141     85.066ms    51.023ms     66.33ms
  6 72.14.215.238      95.743ms   107.734ms   139.804ms
  7 72.14.239.205      79.432ms     75.24ms   141.722ms
  8 209.85.245.72      63.215ms   125.553ms    93.301ms
  9 209.85.242.229    150.569ms   111.643ms   107.425ms
 10 209.85.240.220    178.939ms   164.764ms   126.063ms
 11 209.85.255.75     116.381ms    88.085ms    72.565ms
 12 216.239.49.36     111.364ms    54.734ms    54.516ms
 13 None             None    ms  None    ms  None    ms
 14 74.125.136.94      54.982ms    54.685ms    54.285ms

Disconnecting from stream

You can find details about this measurement here:

  https://atlas.ripe.net/measurements/2904858/
```

Searching for existing measurements

```
Terminal
(2015-11-02 19:12:56) $ ripe-atlas measurements --af 6 --status ongoing --limit 15 --search google

Filters:
  Search: google
  Af: 6
  Status in: (2,)

Id      Type       Description                                          Status
==============================================================================
1004005 ping       google - v6                                         Ongoing
1004732 traceroute google v6 traceroute                                Ongoing
1007128 dns        Google.fi AAAA reply                                Ongoing
1012449 sslcert    www.google.com                                      Ongoing
1024911 ping       IPv6 Google DNS                                     Ongoing
1404300 ping       IPv6 Ping to Google                                 Ongoing
1665737 ping       google.com - 2404:6800:4003:c00::71                 Ongoing
1796260 ping       Ping measurement to www.google.com                  Ongoing
1889086 traceroute Traceroute measurement to ipv6.google.com           Ongoing
2062542 traceroute Traceroute measurement to ipv6.google.com           Ongoing
2062543 ping       Ping measurement to ipv6.google.com                 Ongoing
2143865 ping       Ping measurement to ipv6.google.com                 Ongoing
2486602 traceroute IPv6 Traceroute measurement to snapchat.com         Ongoing
2486820 ping       Google IPv6                                         Ongoing
==============================================================================
                                         Showing 14 of 14 total measurements
```

```
ripe-atlas measurements --af 6 --status ongoing --limit 15
--search google
```

Searching for very specific probes

```
Terminal
(2015-11-02 19:27:57) $ ripe-atlas probes --asn 3333 --field id --field asn_v6 --field country --field is_public --field description --field status

Filters:
  ASN: 3333

ID    Asn_v6 Country Public Description                    Status
==========================================================================
9     3333     nl      ✔    SG office 1                   Connected
14    3333     nl      ✔    vty probe                     Connected
15    3333     nl      ✔    SG office 2                   Connected
111   3333     br      ✘    NIC.br                        Disconnected
237   3333     nl      ✔    The Traveling Probe           Connected
1108  3333     us      ✔    probe 1                       Disconnected
2009  3333     nl      ✔    NCC Office 2009               Connected
3497  3333     nl      ✘                                  Disconnected
6001  3333     nl      ✔    AA nl-ams-as3333              Connected
6012  3333     nl      ✘    AA pre-production             Connected
6018  3333     nl      ✘                                  Disconnected
6019  3333     nl      ✔    RIPE NCC Anchor v2            Disconnected
6137  3333     nl      ✔    nl-ams-as3333-preprod         Connected
10004 3333     nl      ✔    RIPE NCC R&D Office           Disconnected
10105 3333     nl      ✘                                  Disconnected
10888 3333     nl      ✔    Ridderkerk - UPC 120/10Mbits  Disconnected
12989          de      ✔    TeraStream Test Lab           Disconnected
13343 3333     ch      ✔    FSIT AG - CH-DIE001 - out of 0 Disconnected
14004 3333     nl      ✘                                  Disconnected
14013 3333     nl      ✘                                  Disconnected
14020 3333     nl      ✘                                  Disconnected
14024 3333     nl      ✘                                  Disconnected
14026 3333     nl      ✘                                  Disconnected
14754 3333     us      ✔    RETURNED TO RIPE FOR REPAIR   Disconnected
15027 3333     nl      ✘                                  Disconnected
==========================================================================
                                          Showing 25 of 33 total probes
```

```
ripe-atlas probes --asn 3333 --field id --field asn_v6 --field
country --field is_public --field description --field status
```

Links

- Source: https://github.com/RIPE-NCC/ripe-atlas-tools/
- Documentation: https://ripe-atlas-tools.readthedocs.org/
- How to contribute: https://github.com/RIPE-NCC/ripe-atlas-tools/blob/master/CONTRIBUTING.rst

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

Contact us

- https://atlas.ripe.net & 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 and #RIPEAtlas

Documentation

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/keys2/

Additional slides

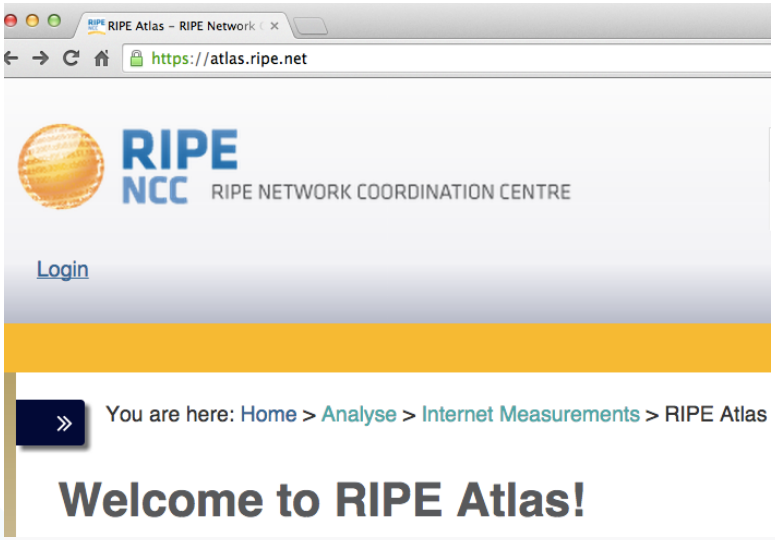
Finding results of public measurements

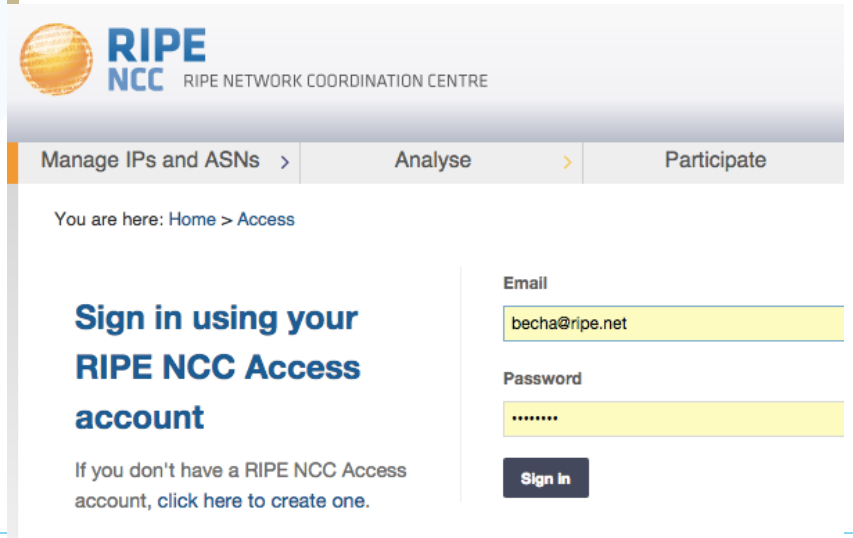
Use existing measurements

- There are many measurements already running!
- Search for existing public measurements first
- Schedule your own measurement if you don’t find what you’re looking for

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





Looking up measurements results

Measurements – RIPE Atlas — RIPE Network Coordination Centre

https atlas.ripe.net/measurements/

H... | Atlas Slide Pa... | Rene Wilhelm... | IXP LANs | MM Updates... | – RIPE Atlas... | Measurement... | Select Ambas... | rsnog Info Page | SEE — RI

Vesna Manojlovic

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 > Measurements

RIPE Atlas

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

Measurements

+ Create a Measurement

Filter by target and/or description | Any Status | IPv4/v6 | ✓ All types | Ping | Traceroute | DNS | HTTP | SSL Certificate | of all time

Mine | Favourites | Hidden | Public | All

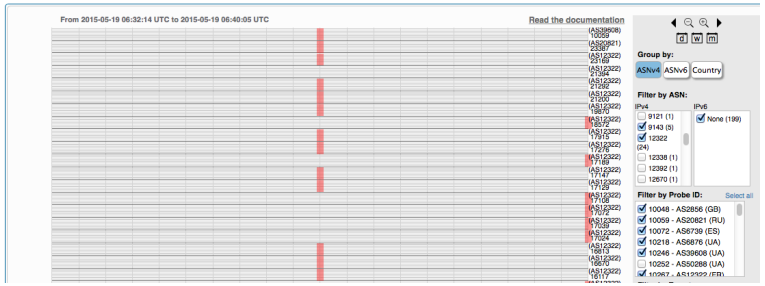
Id		Type	Target	Description		(UTC)	Status
1965015	Vesna Manojlovic	IPv4 ping	b92.net	Ping measurement to b92.net	49	2015-04-21 08:20 2015-04-21 08:30	

Available visualisations: ping

- 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.



Graph tips

Red = packet loss

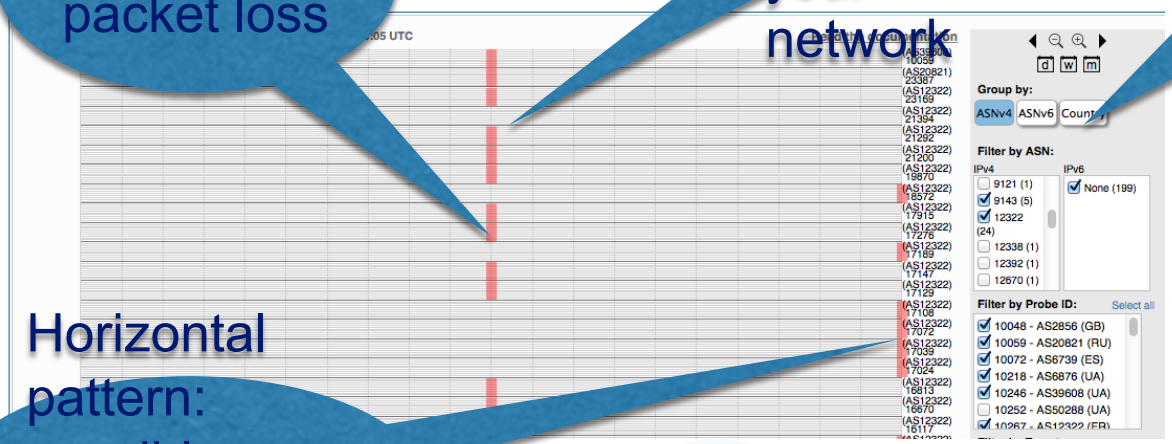
Vertical pattern: possible problem in your network

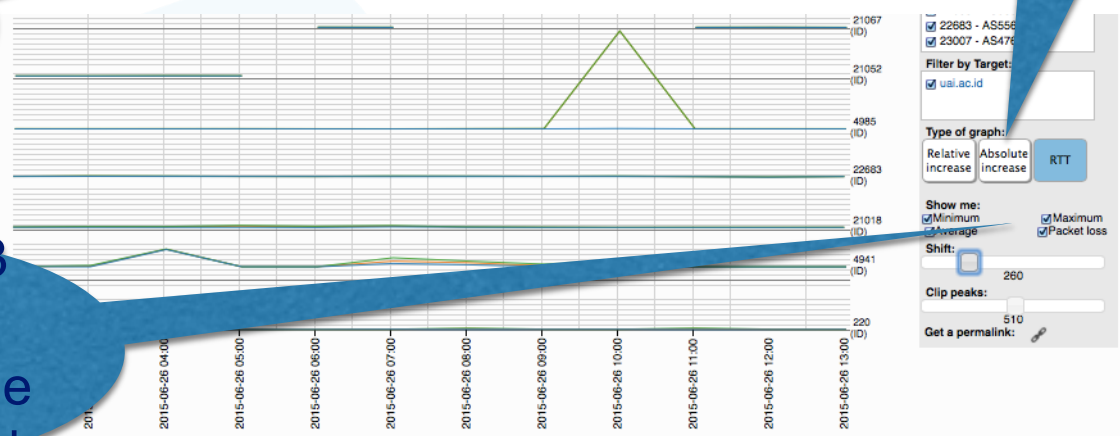
Filtering & grouping by: country, IPv4 ASN, IPv6 ASN

Chose absolute or relative RTT values

Horizontal pattern: possible problem in that ASN network

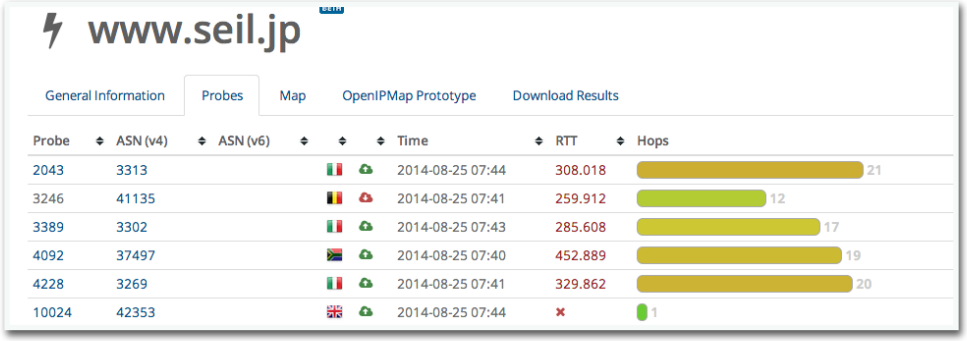
Out of each 3 pings max, min & average are calculated

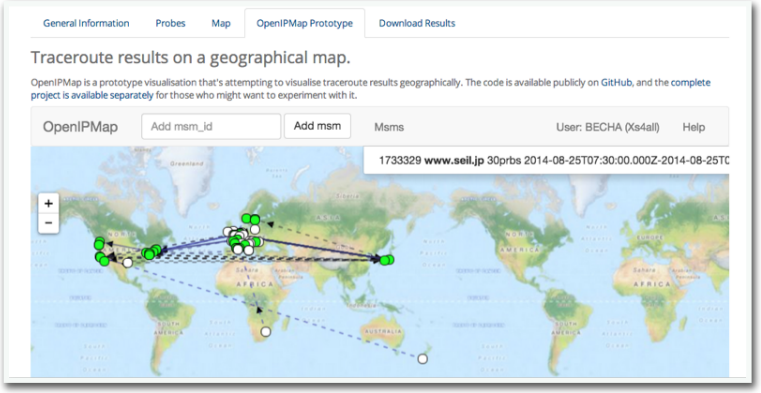




Available visualisations: traceroute

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





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 | 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

- 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

Exercise

Analyse Measurement Results

Tasks

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

Task 1: Download measurement results

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

Tips for downloading results

- 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

Destination (IP & name)

Reference (msm ID)

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",
- ":0,
- om":"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"},

Task 3: Analyse results (optional)

- Find out how many times RTT was above 60ms
  Use Python o 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

Task 4: Examples of code

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

i must be > than 14563.
```

Javascript:

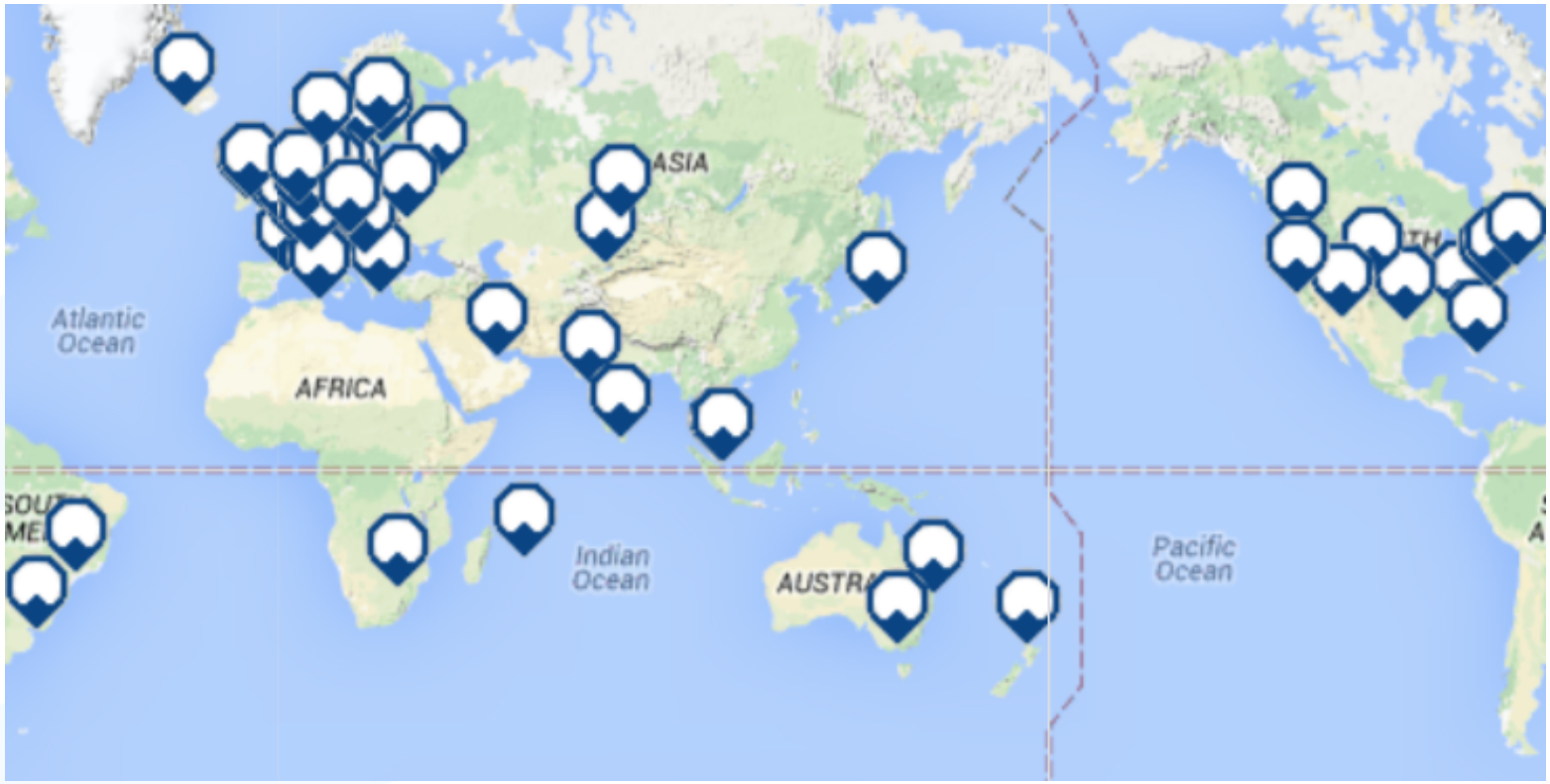
```
var dataAPIUrl = "https://atlas.ripe.net/
api/v1/measurement/1004005/result/?
start=1410220800"; jQuery.ajax({

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" });
```

RIPE Atlas anchors

RIPE Atlas anchors

- 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

Locations of anchors



https://atlas.ripe.net/results/maps/network-coverage/#anchors

IXP country Jedi

Measuring Impact of IXPs on Keeping Traffic Local

Benefits (part 1)

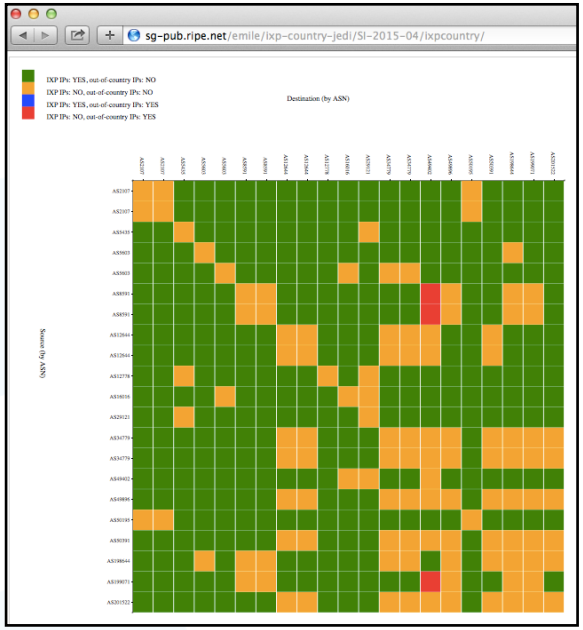
- Operators
  Routing and traffic optimisation
- IXP operators
  Shows how IXPs help keep traffic local and regional
- IPv6 advocates
  Comparing IPv4 and IPv6 paths
- Country level: regulators, politicians, cyber-security…
  How much traffic stays within the country?
  Where do the paths go?
  Comparing countries with each other

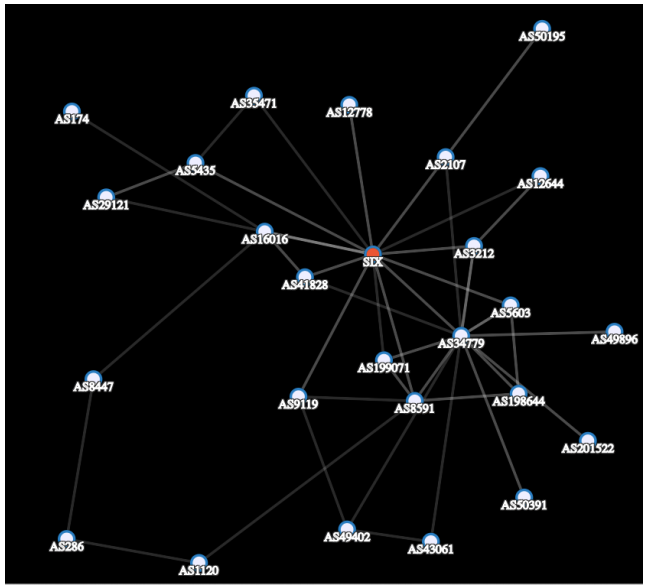
Benefits (part 2)

- 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/

Paths going via an IXP?

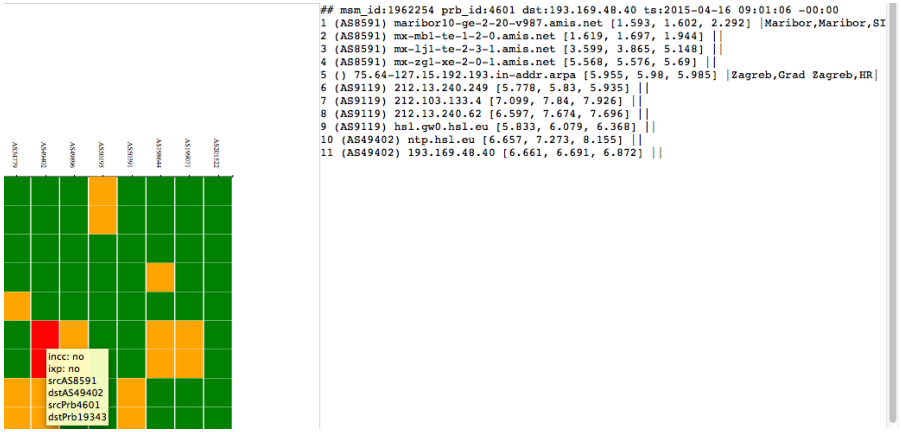
http://sg-pub.ripe.net/emile/ixp-country-jedi/SI-2015-04/ixpcountry/





Interactive diagnostic tool

- 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



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 :)

- 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)

- 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

Want to host a probe?

Contact: sebastian@nzrs.net.nz
www.nzrs.net.nz

NZRS
an InternetNZ company