Workshop: Advanced RIPE Atlas Usage

Daniel Quinn, Massimo Candela,
Chris Amin, Johan ter Beest, Philip Homburg
RIPE NCC

RIPE70 | Amsterdam | May 2015

Overview

- Finding public measurements (5 minutes)
  - **Exercise: Analyse results** **(10 minutes)**
- Creating a measurement (5 minutes)
  - **Exercises: Create a measurement** **(10 minutes)**
- Integration with network monitoring systems (5 minutes)
  - **Exercise: Create status checks** **(10 minutes)**
- Real-time performance monitoring (5 minutes)
  - **Exercise: Use streaming API** **(10 minutes)**

Participants

- How many of you have ever heard of RIPE Atlas?
- Who's used the graphical interface to schedule a measurement in RIPE Atlas?
- How many of you have used the RIPE Atlas API?
- Workshop description:
  - https://ripe70.ripe.net/programme/meeting-plan/workshops/

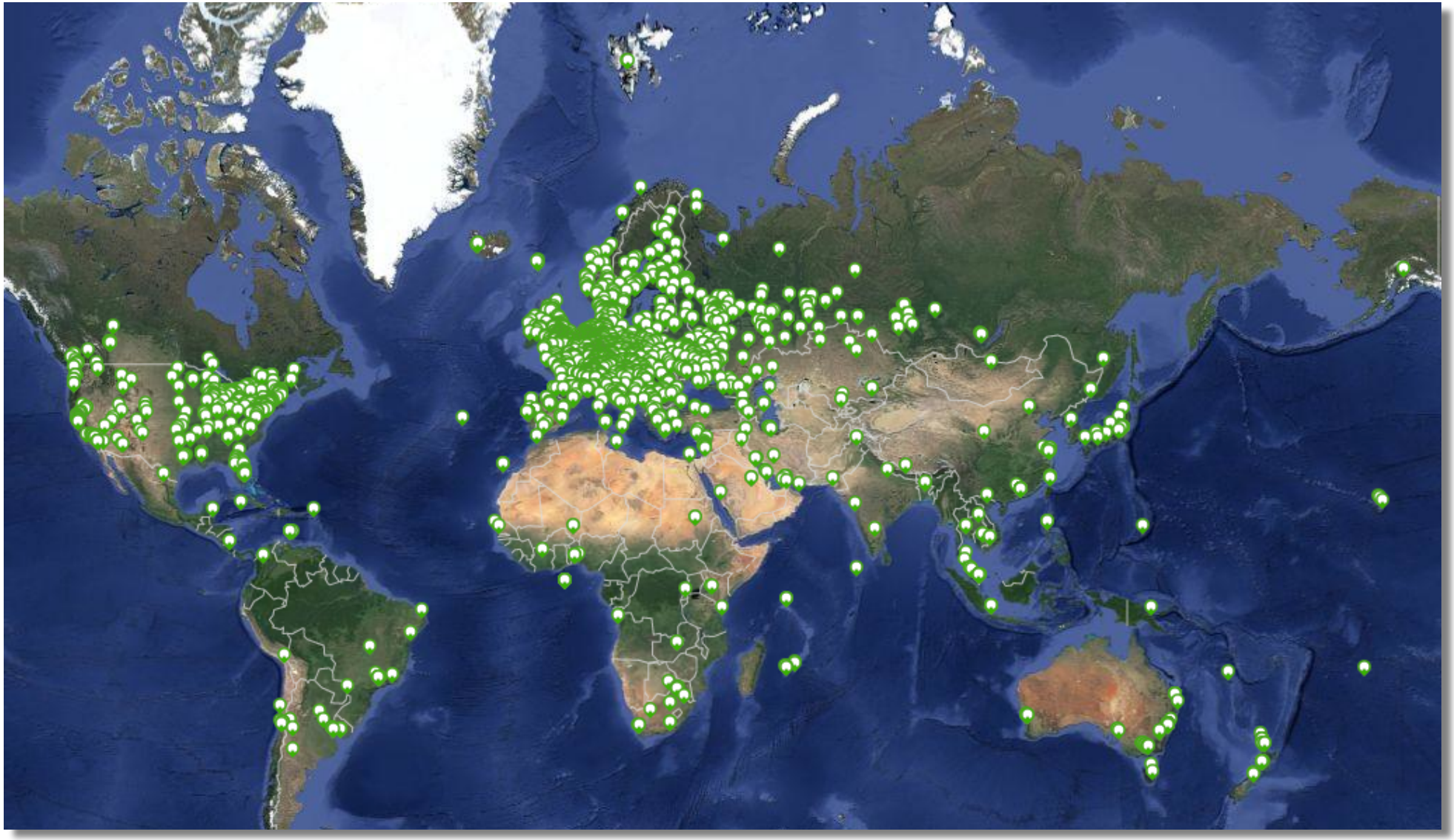
Introduction to RIPE Atlas

RIPE NCC

Definition

- RIPE Atlas = global active measurements platform
- Goal: View Internet reachability
- Probes hosted by volunteers
- Ongoing global measurements towards root name servers
  - Visualised as Internet traffic maps
- Ongoing regional measurements towards “anchors”
- Users can run customised measurements
  - Ping, traceroute, DNS, SSL/TLS and **NTP** (new)
- Data publicly available

RIPE Atlas coverage



RIPE NCC

Finding Results of Public Measurements

RIPE NCC

Use existing measurements!

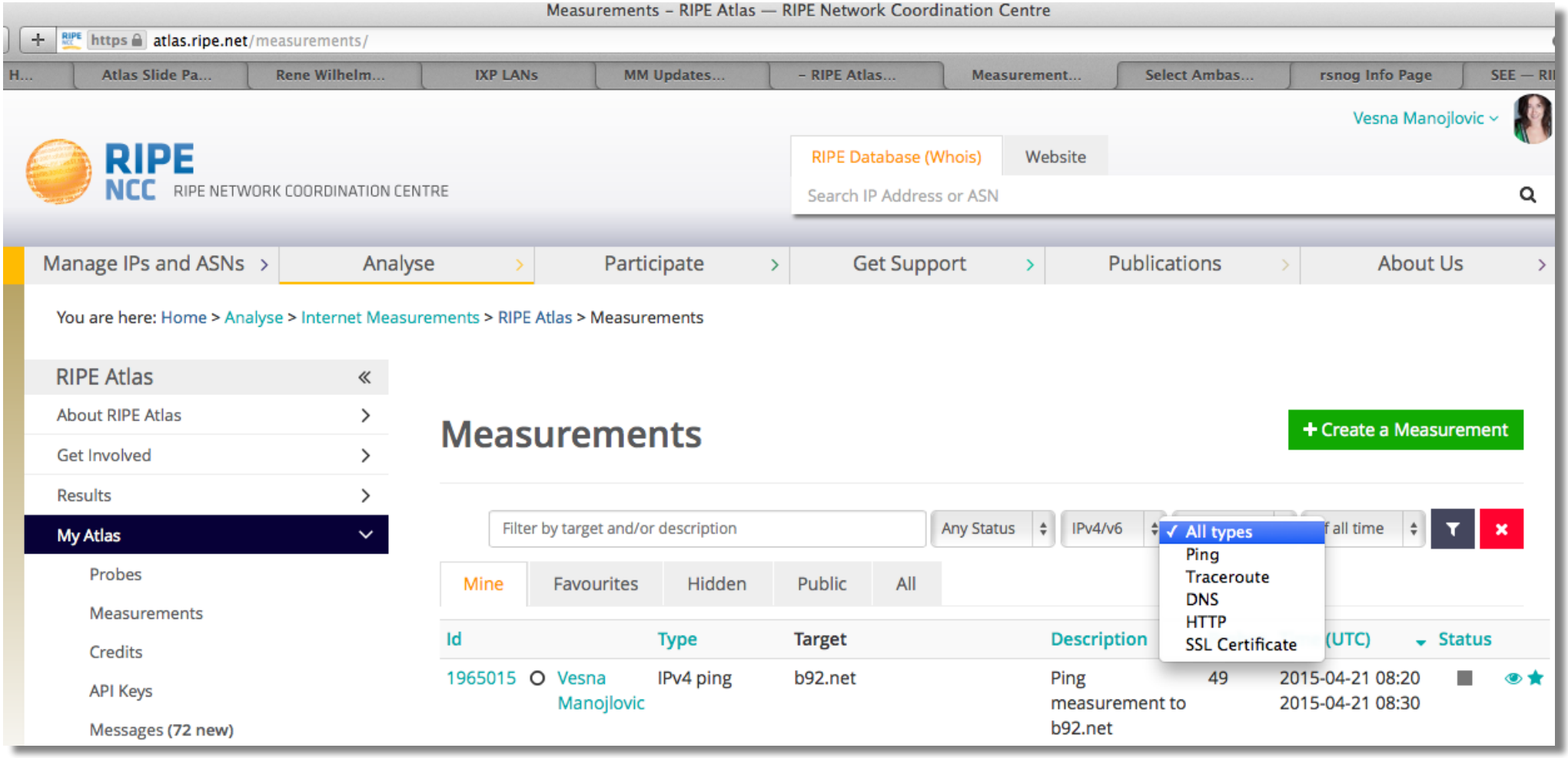
- There are many measurements already running!
- Search for existing public measurements first
- Schedule your own measurement if you don’t find what you’re looking for

Logging in

- Log in to atlas.ripe.net with your RIPE NCC Access account
  - This is the same account used to access the LIR Portal, RIPE Atlas, RIPEstat, RIPE Labs…)
  - Create an account if you don’t have one already

Looking up measurement results

- Go to “My Atlas” > “Measurements”

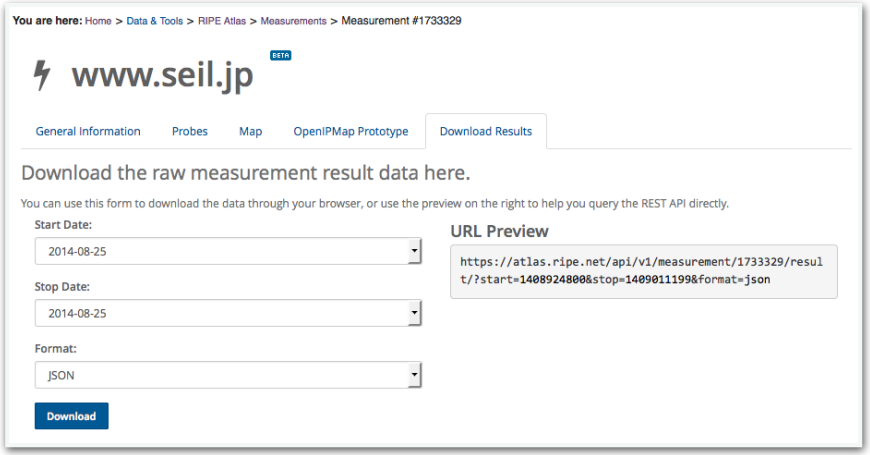


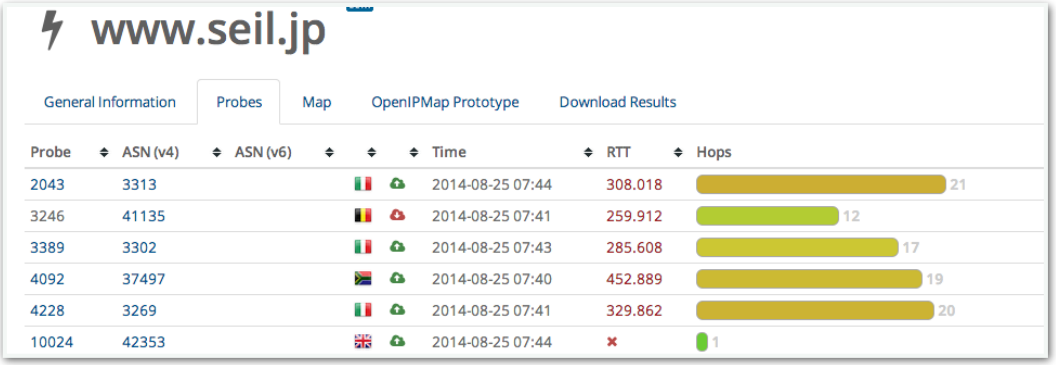
Downloading measurement results

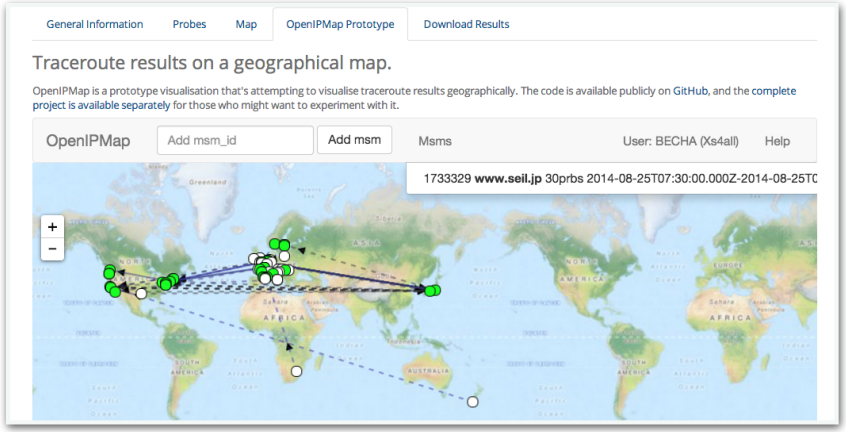
- Click on measurement, then “Download”

- Example:
  - Target: www.seil.jp
  - ID #1733329
  - https://atlas.ripe.net/measurements/1733329/

- Results in JSON







Docs

- 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

RIPE NCC

Tasks

- Get the msm-ID 1004005 (ping IPv6)
  - Measurement data from 2014-09-09 to 2014-09-11
  - https://atlas.ripe.net/api/v1/measurement/1004005/result/?start=1410220800&stop=1410479999
- Find out how many times RTT was above 60ms
  - Use Python, Javascript or something else
  - For Javascript, you can use this as a starting point: https://stat.ripe.net/widgets/demo/script_me.html

Tips and solutions

Save the measurement(s) locally:

curl https://atlas.ripe.net/api/v1/measurement/1004005/result/? start=1410220800 > measurement.json

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”]
ifrtt >60: i += 1
```

i must be > than 14563.

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>
```

Creating a Measurement

RIPE NCC

Scheduling a measurement

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

- Three methods:
  1. Quick & Easy
     - Type
     - Target
     - Done! (default values are used)
  2. Advanced GIU
  3. CLI scripting using API

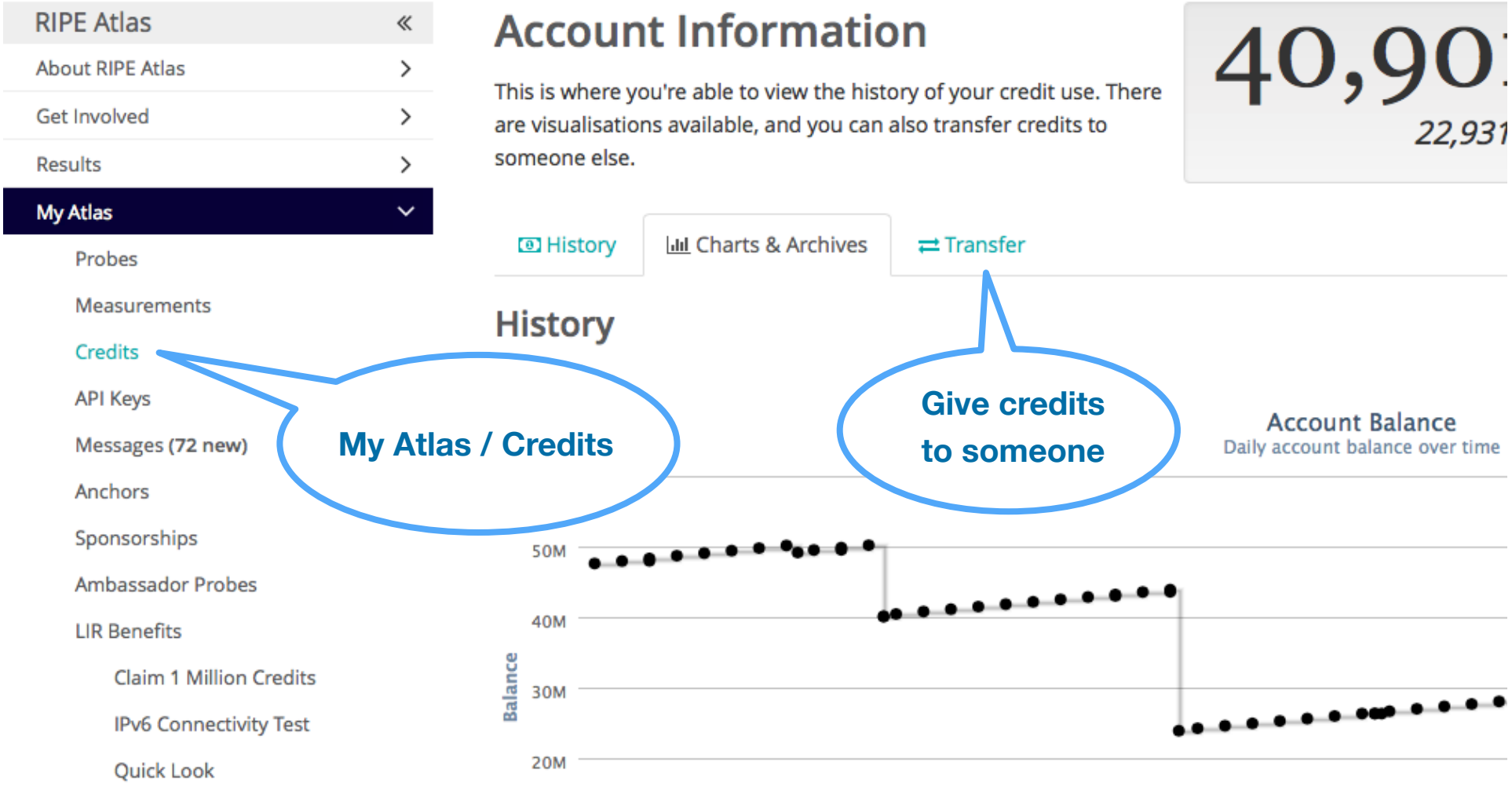
2: Using GUI to schedule a measurement

- Mostly: a periodic, long-term measurement
  - If just want it once, choose “one off”
- Choose type, target, frequency, # of probes, region…
  - Improved, interactive interface helps you at each step
- https://atlas.ripe.net/doc/udm
- You will spend credits

Credit system

- Measurements cost credits
  - Ping = 10 credits, traceroute = 20, etc.
- Why? Fairness and to avoid overload
- Daily spending limit and max number measurements
  - Talk to us if you need more!
- Hosting a RIPE Atlas probe earns credits
  - https://atlas.ripe.net/doc/credits
- Earn extra credits by:
  - Being a RIPE NCC member
  - Hosting an anchor
  - Sponsoring probes

Credits overview



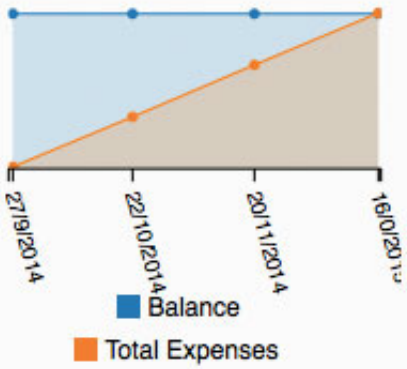
Create a New Measurement BETA

Summary

Daily costs

By scheduling this measurement, your total daily consumption will be 100.0% of your daily income

You will run out of credits in about 81 days



Step 1 Definitions

Ping measurement

Target

An IP address or hostname

Description

Ping measurement

A free-form description of this measurement

Address Family

IPv6

Interval

240

How often this should be done (seconds between samples). Note that this value is ignored for one-off measurements.

Packets

3

Resolve on Probe

Force the probe to do DNS resolution

Size

48

+ Ping + Traceroute + DNS + SSL

RIPE NCC

3: Using API to schedule a measurement

  - https://atlas.ripe.net/docs/measurement-creation-api/
  - Or copy and paste from GUI
- You will need API keys
  - To create measurements without logging in
  - To securely share your measurement data
- To create, manage and delete API keys:
  - https://atlas.ripe.net/keys/
  - https://atlas.ripe.net/docs/keys/
- API documentation
  - https://atlas.ripe.net/docs/measurement-creation-api/

Exercise:
Create a Measurement

RIPE NCC

Tasks

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

1. Create a measurement using the GUI

   OR

2.1 Create API key

2.2 Schedule a measurement using API

Integration with Network Monitoring Systems

RIPE NCC

Network monitoring

- Network operators use tools for monitoring network health
  - For example, Nagios & 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 Icinga or Nagios

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 the percentage of probes that successfully received a reply
  - How many most recent measurements to base it on
  - 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:
Set Up “Status Checks”

RIPE NCC

Tasks

- Set up and configure a status check:
  - For an existing IPv6 ping msm to www.google.com
  - https://atlas.ripe.net/api/v1/status-checks/1004005/

- Configure the status check in such a way that you will trigger an alert for this measurement

Solution

- One possible solution:
  - Set the median RTT to a lower level:
  - https://atlas.ripe.net/api/v1/status-checks/1004005/?median_rtt_threshold=10

Real-time Performance Monitoring

RIPE NCC

New: RIPE Atlas streaming

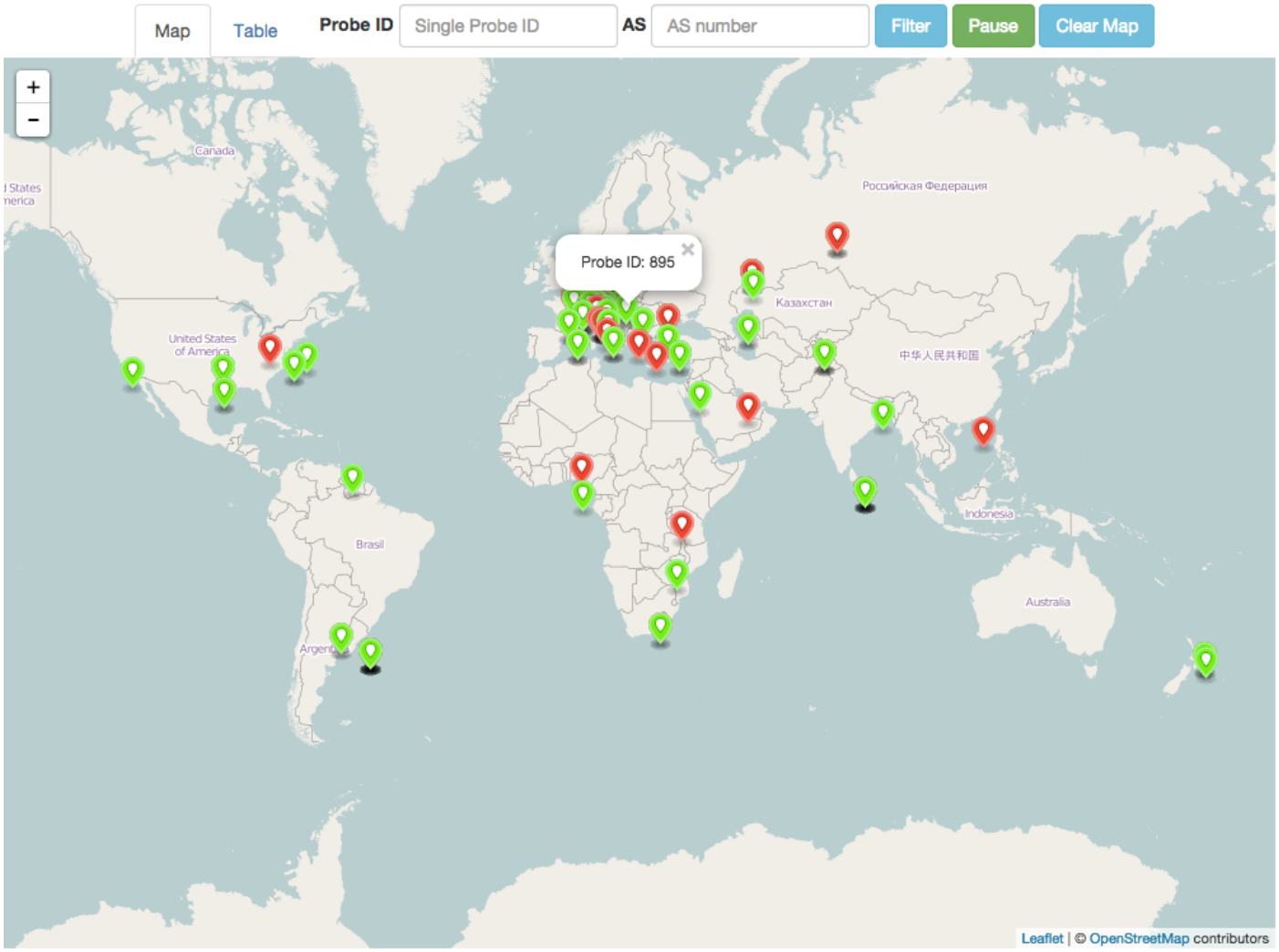
- **RIPE Atlas streaming** is a new architecture that allows users to receive measurement results as soon as they are sent by the probes, **in real time**
  - Publish/subscribe through sockets
- There are two types of data:
  - Measurements results
  - Probe connection status events
- Next: Possibility to replay history

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

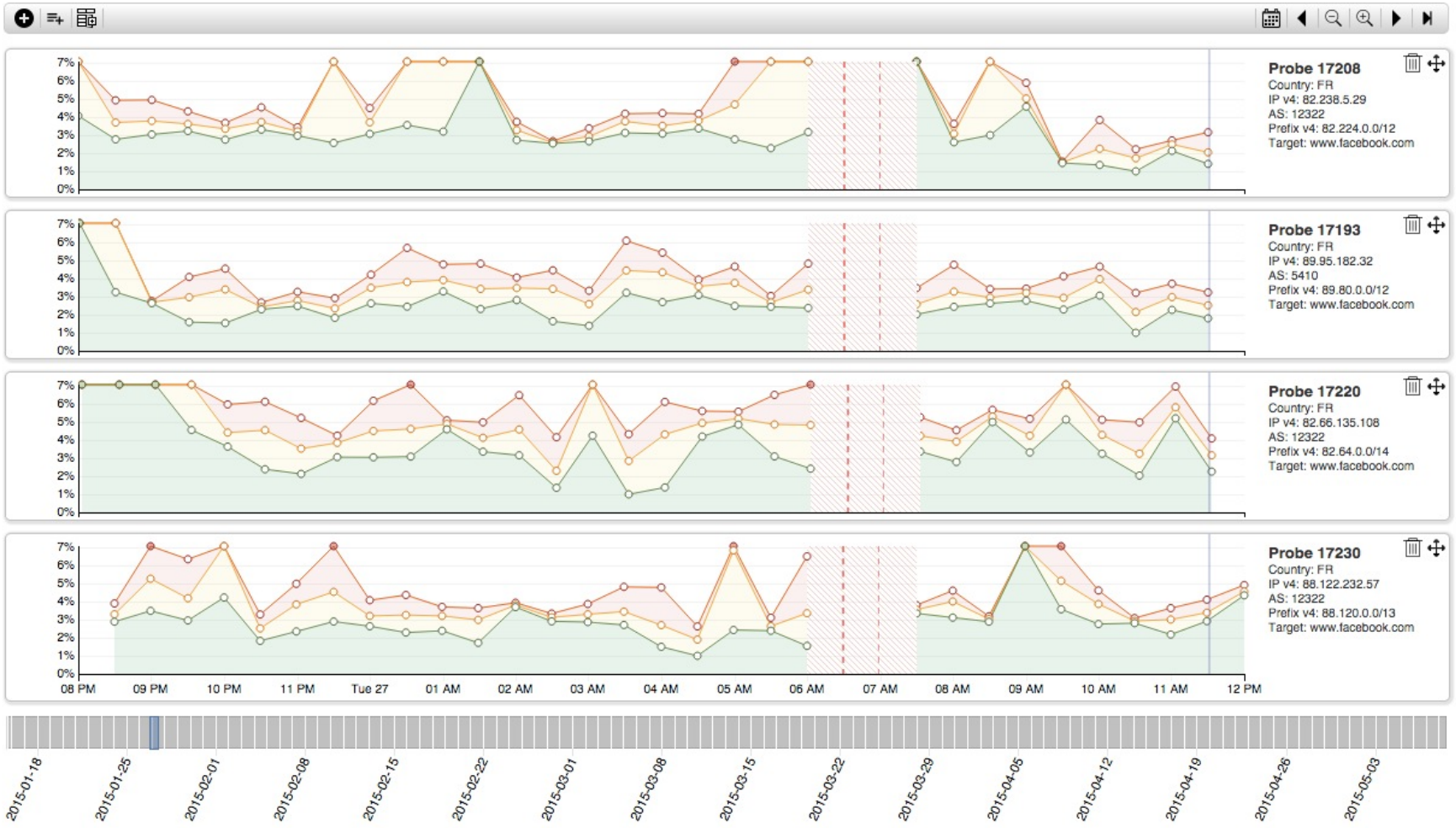
Probe (dis)connection events



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

Early beta: ping visualisation

http://sg-pub.ripe.net/demo-area/massimo-pingviz/



Exercise:
Use Streaming API

RIPE NCC

Exercise: Monitoring server reachability

- Scenario: Customers are occasionally complaining that it takes a long time to reach your service/server

- Action: Ping your server from 500 probes
  - Decide on acceptable latency threshold to apply
  - Notice and react when you start receiving samples

- Task: Use the ping measurement ID1791207
  - Choose the threshold (e.g. greater than 30ms)
  - Impose the threshold on “min” (the minimum result of the three ping attempts)

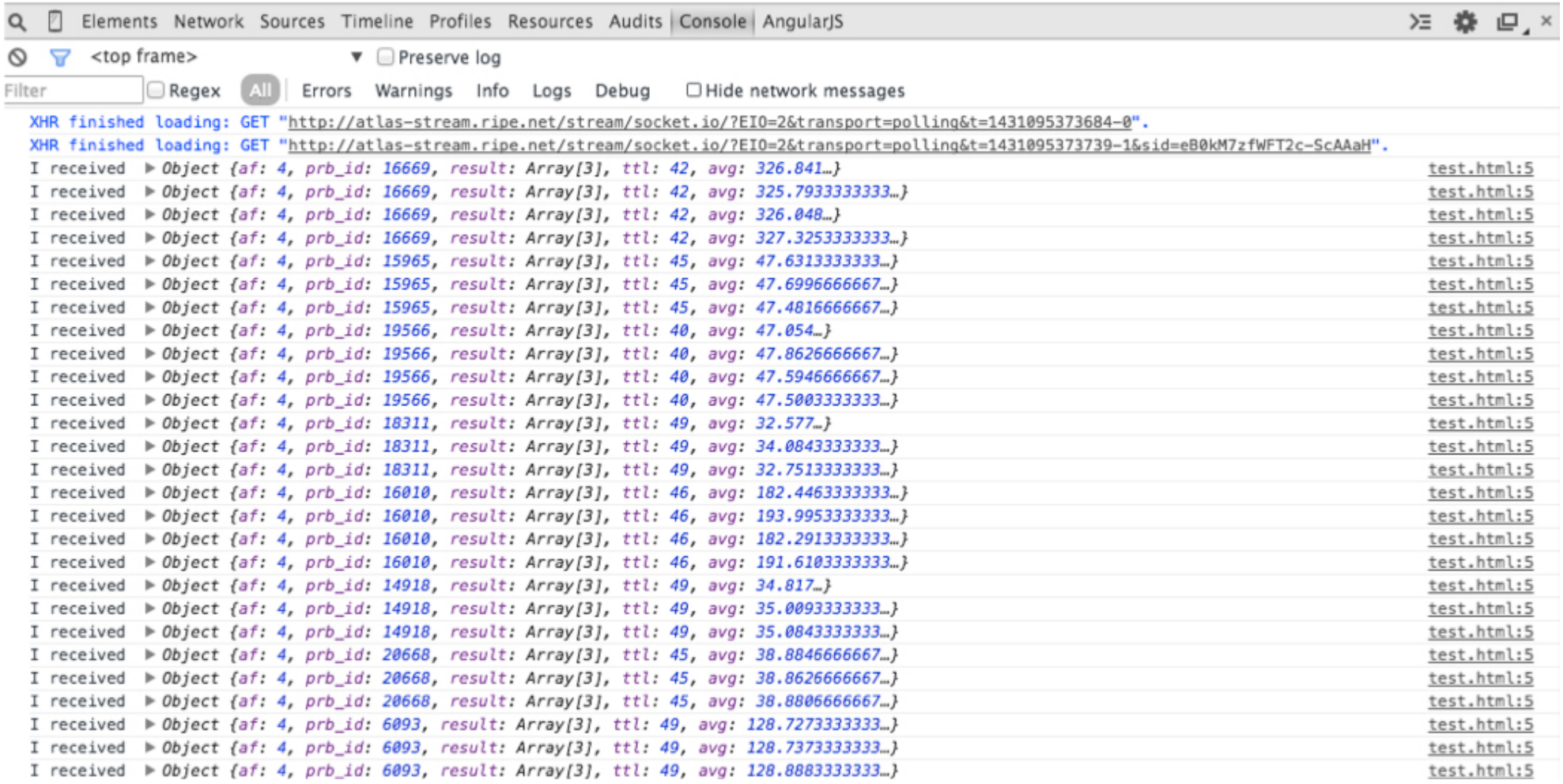
Steps

- Create your empty HTML page
- Copy the example from the documentation: https://atlas.ripe.net/docs/result-streaming/

- Edit the example to fit the task

- Open HTML page
- Open the console
- Wait for results to arrive

Example of results



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_result", function(result){
        console.log("I received ", result);
    });

   socket.emit("atlas_subscribe", {
                        stream_type: "result",
                        msm: 1791207,
                        greaterThan: {min: 30}
});
</script>
```

Task 2: View (dis)connect events (optional)

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

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

Task 2: 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>
```

Questions?

Contact us

- https://atlas.ripe.net
- Users' mailing list: ripe-atlas@ripe.net
- Articles and updates on RIPE Labs:
  - https://labs.ripe.net/atlas
- Questions and bugs: atlas@ripe.net
- Twitter: @RIPE_Atlas and #RIPEAtlas

Additional Slides

RIPE NCC

Numbers for May 2015

- 8,200+ 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** (new)

How to take part

- For individuals - 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

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

- 120+ RIPE Atlas anchors

Locations of anchors

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

New Features

- Impact of IXPs on keeping traffic local

  https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas

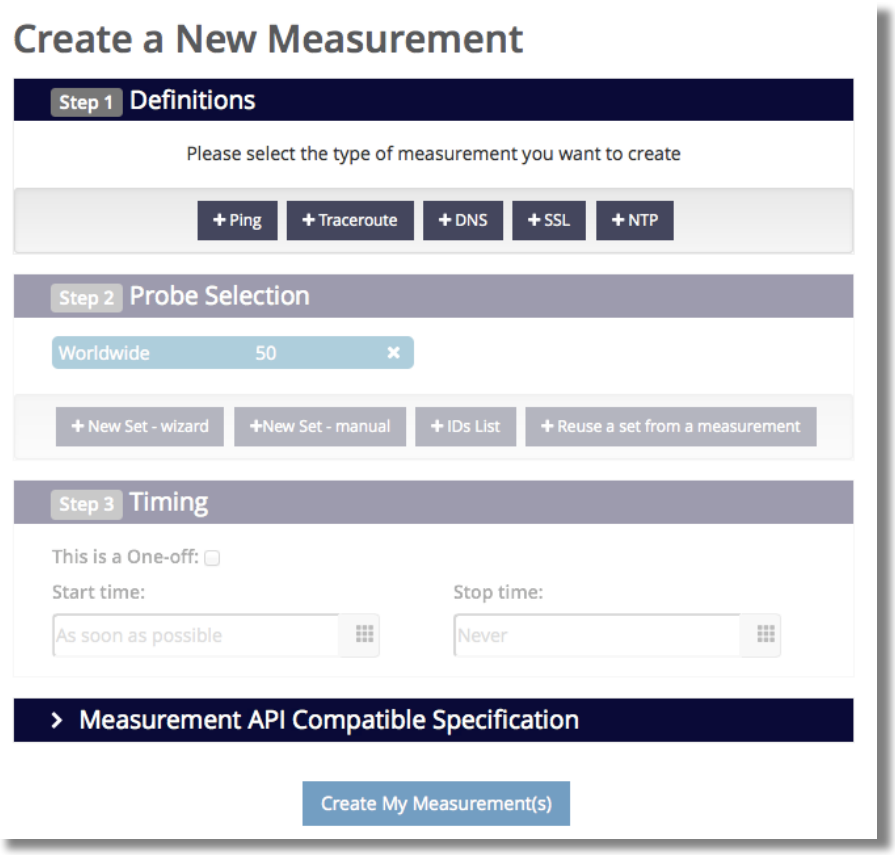
  https://labs.ripe.net/Members/emileaben/measuring-countries-and-ixps-in-the-see-region

  http://sg-pub.ripe.net/emile/ixp-country-jedi/CL+AR-2015-04/geopath/

New Features

- New measurement type: NTP

  https://labs.ripe.net/Members/philip_homburg/ntp-measurements-with-ripe-atlas



- RIPE NCC membership additional benefits:
  - Can perform measurements without hosting a probe
  - Get 1,000,000 credits each month
  - Share probe management with LIR colleagues

Other related projects on GitHub

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

- https://github.com/emileaben/ixp-country-jedi