




Webinar: Advanced RIPE Atlas Usage

Vesna Manojlovic
Massimo Candela

Amsterdam | 5 November 2015

1




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
- Get your questions answered by developers

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

2




Prerequisites

- We 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 us your account in the chat window

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

3




Warm-up question

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

4




Overview

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

5



Introduction to RIPE Atlas

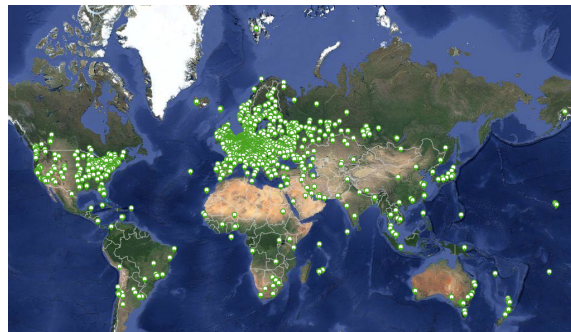
6

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



7

RIPE Atlas coverage



8

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

9

Numbers for October 2015



- 8,900+ probes connected
- 5,000+ active users in the last quarter
- 2,500+ results collected per second
- 35,000+ customised measurements weekly

10



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

12

11

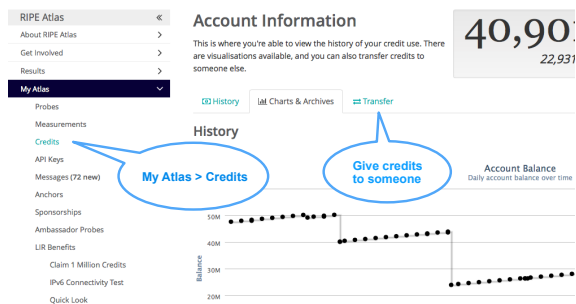
Credits system



- 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

13

Credits overview



14

Scheduling a measurement



- Log in to atlas.ripe.net
- “My Atlas” > “Measurements”
- Three methods:
 1. Quick & Easy
 - Choose measurement type
 - Specify target
 - Done!
 2. Advanced GUI usage
 3. CLI scripting using API

15

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 by the GUI

16

3: Using API to schedule a measurement



- Using command-line & scripting:
Application Programming Interface (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

17

- 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/>

18

Looking up measurements results

- Go to "My Atlas" > "Measurements"

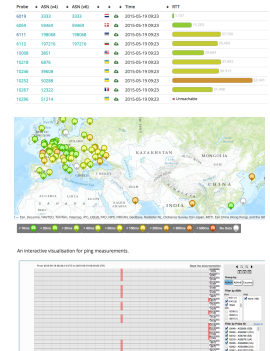
Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

18

19

Available visualisations: ping

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



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

19

20



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

- Warm-up: Create a measurement using the GUI
- Create API Key
- Schedule a measurement using the API

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

21

22

Sub-task 1: Use web interface

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

22

23

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)

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

23

24

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
```

25

```
air-becha:~ becha$ curl -H "Content-Type: application/json" -H "Accept: application/json" -X POST -d '{ "definitions": [ { "target": "ping.xs4all.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-fb1a007d060
{"measurements": [2421551]}air-becha:~ becha$
```

26



Integration with Network Monitoring Systems

27

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

28

Steps for integration



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



29

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/>

30

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/>

31



Exercise

Setting up "Status Checks"

32

Tasks



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

33

Solution



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

```
{
  "total_alerts": 32,
  "global_alert": true,
  "probes": {
    "19433": {
      "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": "23.72",
        "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"],
      }
    }
  }
}
```

34

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

35

36

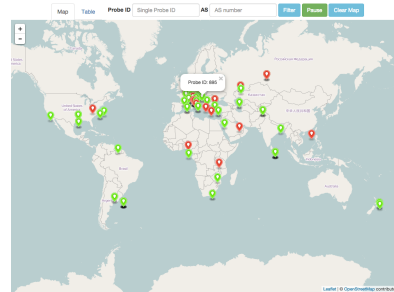
Real time streaming uses



- 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

37

Probe (dis)connection events



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

38



Exercise

Using streaming API

39

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)

40

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

41

Page Source



```
11 <script src="https://atlas.ripe.net/widgets/lib/jquery/jquery-1.11.2.min.js"></script>
12
13 <!-- The following file is needed for the streaming -->
14 <script src="https://atlas-stream.ripe.net/socket.io.js"></script>
15
16 var OutputDiv = $('#output');
17
18 // Create a connection
19 var socket = io('https://atlas-stream.ripe.net', { path: '/stream/socket.io' });
20
21 // Declare a callback to be executed when a measurement result is received
22 socket.on('atlas_result', function(result) {
23   console.log('I received ', result); // Print the result in the console
24   if (result.hasOwnProperty('max')) {
25     $('#output').html(result['max']); // Print the result in the html page
26   }
27 });
28
29 // Subscribe to results coming from all the probes involved in the measurement 2340408
30 socket.emit('atlas_subscribe', { stream_type: 'result', mem: 2340408 });
31
32 </script>
33 </html>
```

42

Contact us



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

49

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/>

50

More material



- 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>
 - <https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/blob/master/README.md>
 - <https://github.com/RIPE-NCC/ripe-atlas-tools/>

51



Additional slides

52



The Newest Features

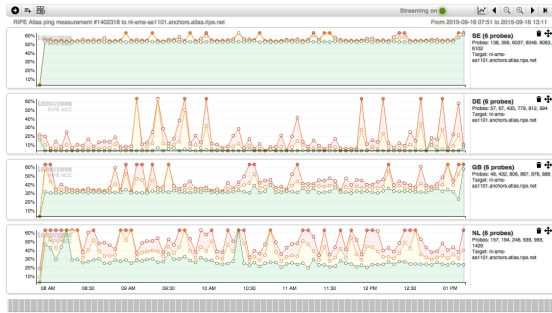
53



LatencyMON

54

LatencyMON



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

50

55

Adding multiple measurements

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



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

51

56

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

<input type="checkbox"/>	Probe ID	Country	ASv4	ASv6	IPv4	IPv6	Measurement ID
<input type="checkbox"/>	57	DE	20621	20621	217.69.64.206	2001:aa8:f1e:c3:220:4aff:fec8:2098	1402318
<input type="checkbox"/>	67	DE	31334	31334	95.90.204.77	2a02:8108:9e40:a0:220:4aff:fec8:249e	1402318
<input type="checkbox"/>	157	NL	3265	3265	82.95.108.192	2001:981:5e40:1:220:4aff:fec8:20b9	1402318
<input type="checkbox"/>	194	NL	39309	39309	88.159.184.218	2a01:810:8aa4:1d80:220:4aff:fec8:2099	1402318
<input type="checkbox"/>	226	AU	4739	4739	203.16.208.142	2001:44b8:1121:1:a00:220:4aff:fec8:245d	1402318
<input type="checkbox"/>	239	SN	8346	8346	196.1.85.18	2001:4278:1000:1:16	1402318
<input type="checkbox"/>	246	NL	6830	-	77.251.180.141	-	1402318
<input type="checkbox"/>	333	JP	17676	17676	126.72.61.194	2400:2410:20c0:4400:220:4aff:fec8:242e	1402318

Showing 1 to 8 of 428 rows

Group name:

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

52

57

Command Line Interface (CLI) Toolset



58

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

54

59

Use cases for CLI tools

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

55

60



Finding results of public measurements

67

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

Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

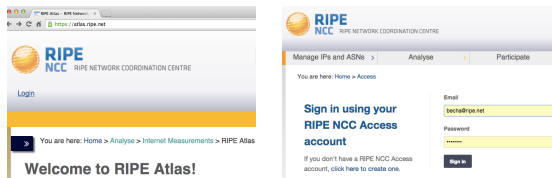
63

68

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



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

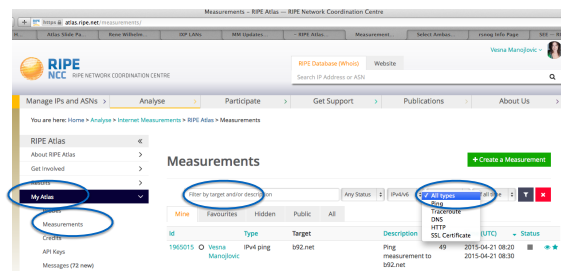
64

69

Looking up measurements results



- Go to "My Atlas" > "Measurements"



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

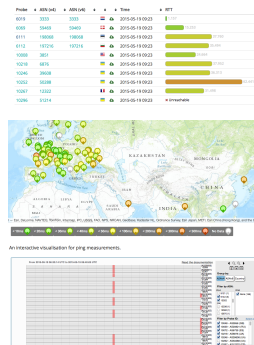
65

70

Available visualisations: ping



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

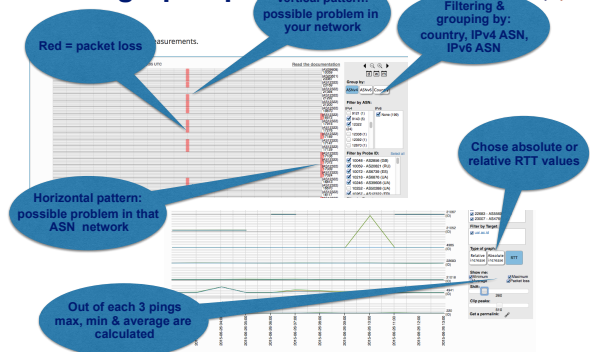


Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

66

71

Seismograph tips



Vesna Manojlovic | RIPE Atlas webinar | 5 November 2015

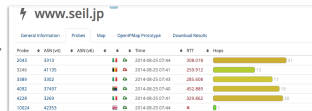
67

72

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>



73

Available visualisations: DNS



- Map, colour-coded response time or diversity



- List of probes, sortable by response time

DNS measurement to nst.opteamax.de

Probe	ASN (v4)	ASN (v6)	Time	Name	Response Time
17480	6207	AS1956	2015-05-19 09:50:18	nst	201.700
18035	43038	AS1956	2015-05-19 09:50:18	nst	211.000
18129	32785	AS1956	2015-05-19 09:49:59	nst	207.700
15244	32188	AS1956	2015-05-19 09:48	nst	207.000
17827	852	AS1956	2015-05-19 09:37	nst	206.000
10804	6327	AS1956	2015-05-19 09:36	nst	205.000
15024	21513	AS1956	2015-05-19 09:30	nst	203.000
15023	30504	AS1956	2015-05-19 09:27	nst	193.000

74

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>

75



Exercise

Analyse Measurement Results

76

Tasks



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

77

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

78

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
```

79

## 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!

80

## Task 3: Analyse results (optional)



- 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](https://stat.ripe.net/widgets/demo/script_me.html)

81

## 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:

```
<script>
var dataAPIURL = "https://atlas.ripe.net/api/v1/measurement/1004005/result?start=1410220800";
$.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>
```

82

## 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

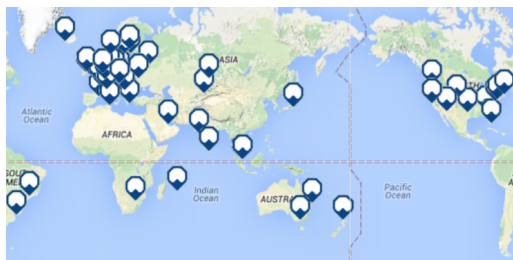
- Vantage points for DNSMON service

- 130+ RIPE Atlas anchors

84

83

## Locations of anchors



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

85

## “IXP country Jedi”

Measuring Impact of IXPs on Keeping Traffic Local



## “IXP country Jedi”

Measuring Impact of IXPs on Keeping Traffic Local

86

## 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

87

## Benefits (part 2)



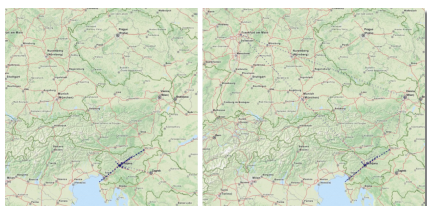
- RIPE Atlas community
  - More probes in more networks = higher quality of measurements data
- Geolocation data community
  - Use case for improving data quality
- Examples:
  - <https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas>
  - <https://labs.ripe.net/Members/emileaben/measuring-countries-and-ixps-in-the-see-region>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

88

## Paths staying in the country?



- Difference between IPv4 and IPv6 paths



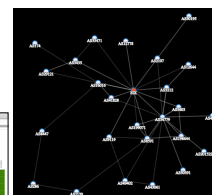
<http://sg-pub.ripe.net/emile/ixp-country-jedi/SI-2015-04/geopath/s/{SI}{RO, BG, HR, BA, ME, AL, GR}/>

89

## Paths going via an IXP?



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

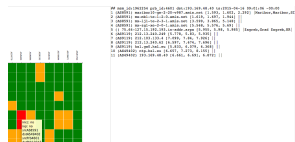


90

## 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/>

91

## 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 ANS and only "public" probes with "good" geolocation
- Hops geolocated using "OpenIPMap" database

92

## 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"

93

## 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>

94