**2.9.1.10: Python Challenge – Analyse RIPE ATLAS data to modify Static Routes**

RIPE Atlas is a global active measurements platform

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

RIPE Atlas RIPE Atlas measures Internet connectivity and reachability

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

RIPE Atlas measurements are taken between probes and or anchors <https://atlas.ripe.net/landing/probes-and-anchors/>

On the measurements page, you can see a list of all scheduled, ongoing and completed measurements, which can be filtered by measurement type, target, protocol type and time period. You do not need to take your own measurements use measurements taken by other organisations, measurements are referenced by their measurement ID

RIPE ATLAS Cousteau is a python wrapper for the RIPE API

<https://ripe-atlas-cousteau.readthedocs.io/en/latest/>

RIPE ATLAS Sagan is a parsing library for the RIPE API measurements

<https://ripe-atlas-sagan.readthedocs.io/en/latest/>

## Objectives

Analyse Ripe atlas data for three RTT delay measurements

Use python scripts to modify the router configuration using its available APIs

The complete network topology is taken from COM513 Packet tracer assessment

Diagram

Description automatically generated

**Note: You only need to modify the configuration for the Edge\_router using the router API**

**Simulate the Edge\_Router using a Cisco CSR1000v virtual router**

**The ISP connections can be simulated by loopbacks**

**The LAN networks are not required in the simulation**

**Primary Skills:**

**Grades F3-F1**

* Send a configuration to the router using python

**Grades D3- D1**

* Accept the input from a user of two ping measurement IDs for a RIPE ATLAS Measurement ( see section 1.2.4.1: The Input Function in COM513 Model Driven Programmability netacad course)
* Get the Average RTT of the two ping measurements using the RIPE ATLAS API (hint use the ripe.atlas.sagan library – PingResult function)
* Create a configuration for a static route to customer network a based on comparing two RIPE ATLAS ping measurements using Average Round Trip Time (RTT)
* Output static route to console or file

Pseudo code:

**if**  RTT 1 > RTT2

**do**  Create a static route to Customer 1 via ISP1

**else If** RTT1 < RTT2

**do** Create a static route to Customer 1 via ISP2

**print**  static route

**Grade C3-C1**

* Accept the input of three ping measurement IDs for a RIPE ATLAS Measurement
* Get the Average RTT of the two ping measurements using the RIPE ATLAS API
* Create a configuration for a static route to customer network a based on comparing two RIPE ATLAS ping measurements using Average Round Trip Time (RTT)
* Automatically create the loopback interfaces using the Router API

Pseudo code:

if RTT1 > RTT2 and RTT3

do Create static route to Customer 1 via ISP1 (Loopback1)

if RTT2 > RTT1 and RTT3

do Create static route to Customer 1 via ISP2 (Loopback2)

if RTT3 > RTT1 and RTT2

do Create static route to Customer 1 via ISP3 (Loopback3)

**print**  static route

**Challenge Skills: Grades B3 -B1**

* Create a configuration for a static route to customer network a based on comparing three RIPE ATLAS ping measurements for **each** customer (3 customers) using the Average Round Trip Time (RTT) data
* Apply three static routes one for each customer to the EDGE\_Router using the router API

**Master Skills: Grade A4- A1**

* When the user inputs a measurement id detect the measurement type as either Ping , traceroute, or DNS, SSL or HTTP
* If the measurement type is Ping compare the average round trip time of **any** three measurements and create a static a route to the customer via the ISP based on the lowest RTT
* If the measurement type is Traceroute compare and select the lowest Median Round trip time and create a static a route to the customer via the ISP based on the lowest value
* Apply the static route to EDGE\_Router using a router API
* Repeat for all three customers

Last modified: Monday, 2 March 2021, 4:40 PM