Quantifying Interference between Measurements in the RIPE Atlas Platforms

T. Holterbach et al., IMC 2015

Presented by: Fabián E. Bustamante
RIPE Atlas

- What?: A measurement platform made of many small devices
  - There are others out there

- What for?
  - Measure network performance, e.g. latency in broadband access, BGP routing changes impact on delay
  - Map the Internet
  - Detect routing attacks, routing anomalies, censorship
Measurement interference?

- Atlas schedules measurements in parallel w/o providing feedback to the user
- This scheduling model + platform limitations wrt resources and software …
  - *Would other measurement interfere with yours?*

- Empirical study of measurement interference
  - How? Against themselves
RIPE Atlas – The platform

- As of April 2015: >6,700 probes over 197 countries

- Hardware – 3 versions
  - V1 and V2 – Lantronix XPort Pro with 167MHz CPU, 8 or 16MB of RAM and 16MB flash
    - Before 2013; currently ~28%
  - V3 – TP-Link TL-MR3020 router with 400MHz CPU, 32MB of RAM, 4MB NAND

- Software
  - Based on BusyBox
    - https://www.busybox.net
  - ATLAS probe – source code available
RIPE Atlas by 2015
Use and credit

- Measurement types supported: traceroute, ping, DNS, SSL and HTTP by request

- Measurement specs
  - Type, frequency, set of probes, start time or “as soon as possible”, end time or “never”, repeat or only-once?

- Use regulated by credit system
  - Collect credit by hosting
  - Use credit w/ measurements; different amount for different type of measurements
Interference between measurements

Impact of measurement from and toward a probe
Interference measurement scenarios

- **When increasing load ...**
  - On probe
    - Impact on ping delays initiated by (D1)
    - ... And targeting the probe (D2)
  - Towards probe
    - Impact on ping delays initiated by (D3)
    - ... And targeting the probe (D4)

- **When increasing load**
  - On probe
    - Impact on one-off traceroute measurement completion time (C1)
  - Towards probe
    - Impact on ... measurement completion time (C2)

- **While preventing effect of external factors**
  - Use Ring* node within the same LAN (found through traceroutes)

*https://ring.nlnog.net/
Delays from / traceroute from

Source: 1.4 ping/sec
Increase load by launching successively 10, 25, ...
... 500 one-off traceroutes
Delays from / ping towards

Source: ping colocated RING at 9 ping/min
Increase set of RING nodes ping source, each at 16 ping/s

- 0.22ms Increase in median delay at 400 ping/s
- Overloaded at 1000 ping/s
- 10% pings lost
Delay to / ping to

- Median RTT increases by 0.45ms at 1000 ping/s
- Reported delays skyrocket; lost packets
Load in source and destination
One-off traceroute completion time

Results can be delayed by more than half an hour
Does it matter?

- Two interesting examples
  - Rimondini et al. uses changepoint detection to detect RTT changes and correlate with routing changes
    - They run the same tests and the algorithm label the load increase as a changepoint
  - Cicalese et al. use minimum RTT to detect anycast replicas
    - A 1ms ~ 100km radius …

- An a general advise – beware of pre-2013 datasets from RIPE Atlas
What do we do then?!

- Can we weight the results with some indication of confidence based on load?
  - How do you determine the “label”?

- Can you schedule experiments to control load?
  - Mario’s work?

- So, how can we design measurement platforms that provide more isolation between users or between experiments while still being efficient?

- Other issue with large-scale measurements in similar platforms?