

---

# Simon Project

Determining Latin America and Caribbean  
Internet connectivity through latency  
measurements

Agustín Formoso | Lacnic | [agustin@lacnic.net](mailto:agustin@lacnic.net)  
[@aguformoso](https://twitter.com/aguformoso)



# Origins

---

- Main problem
  - Sub-optimal routing
  - Scarse knowledge of the LAC Internet conditions: are we improving?
- Main objective: provide recent and accesible information regarding conectivity measurements in LAC.
  - More and better peering agreements
  - More local content
  - More infrastructure investment
  - Have a better regional Internet
- Fisrt talks at NAPLA 2007
- Multidisciplinary team



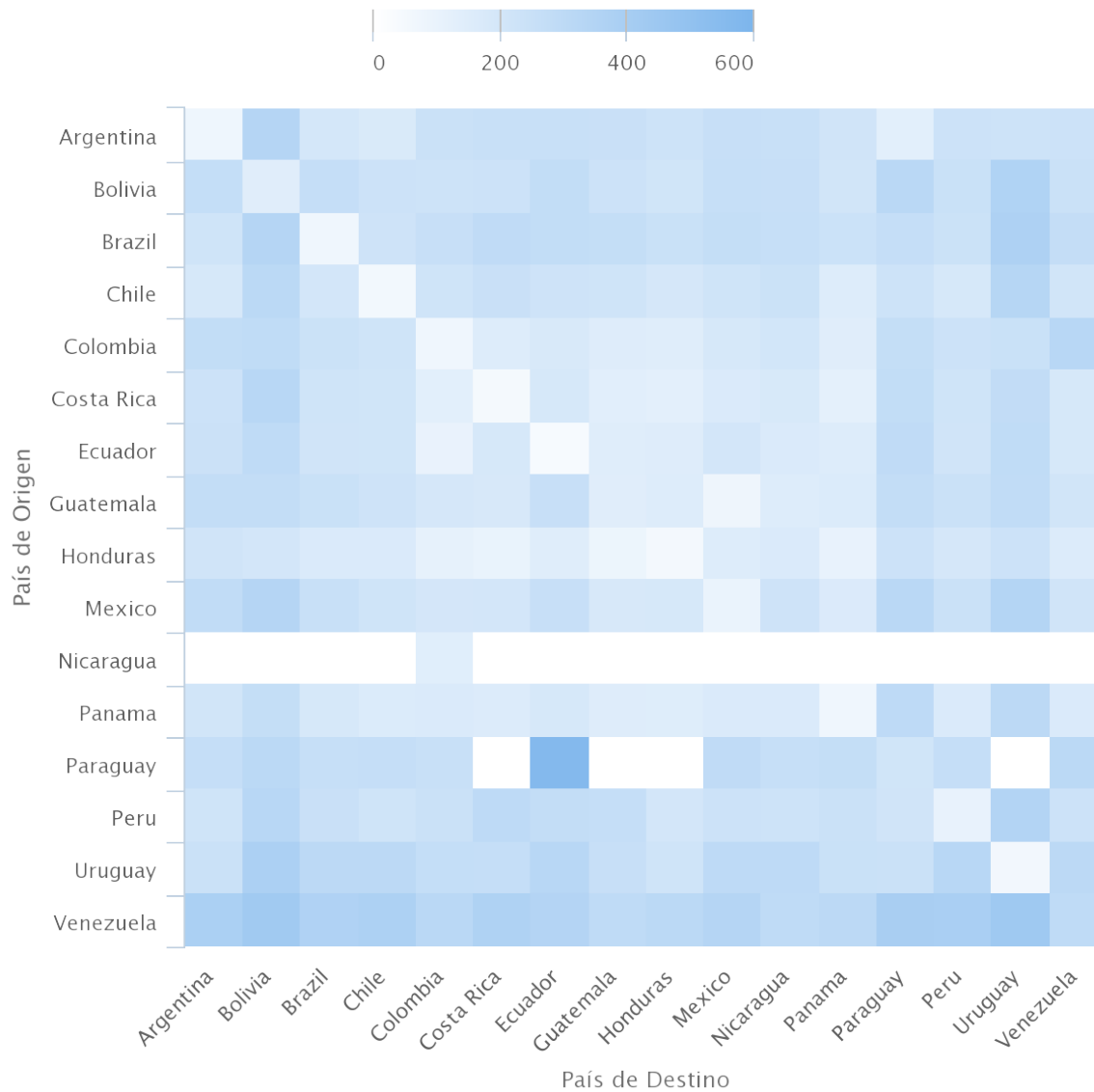
# Milestones

---

- 2009: Java Applet
  - End-user perception
- 2012
  - IPv6 test points
  - Public API
  - First web tool
- 2014
  - Better country reports
  - Portable web tool



# Latency matrix (Applet)

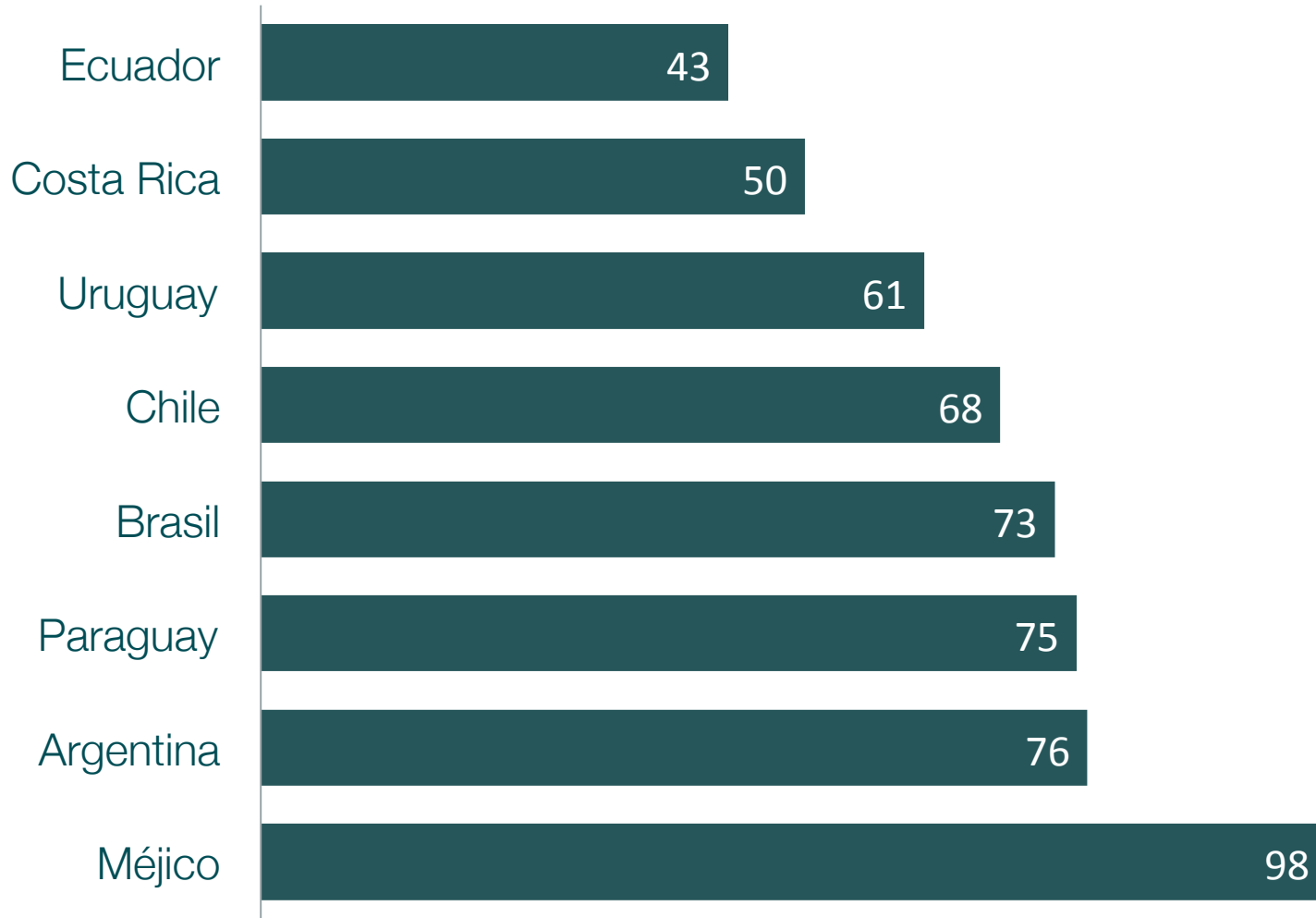


Highcharts.com

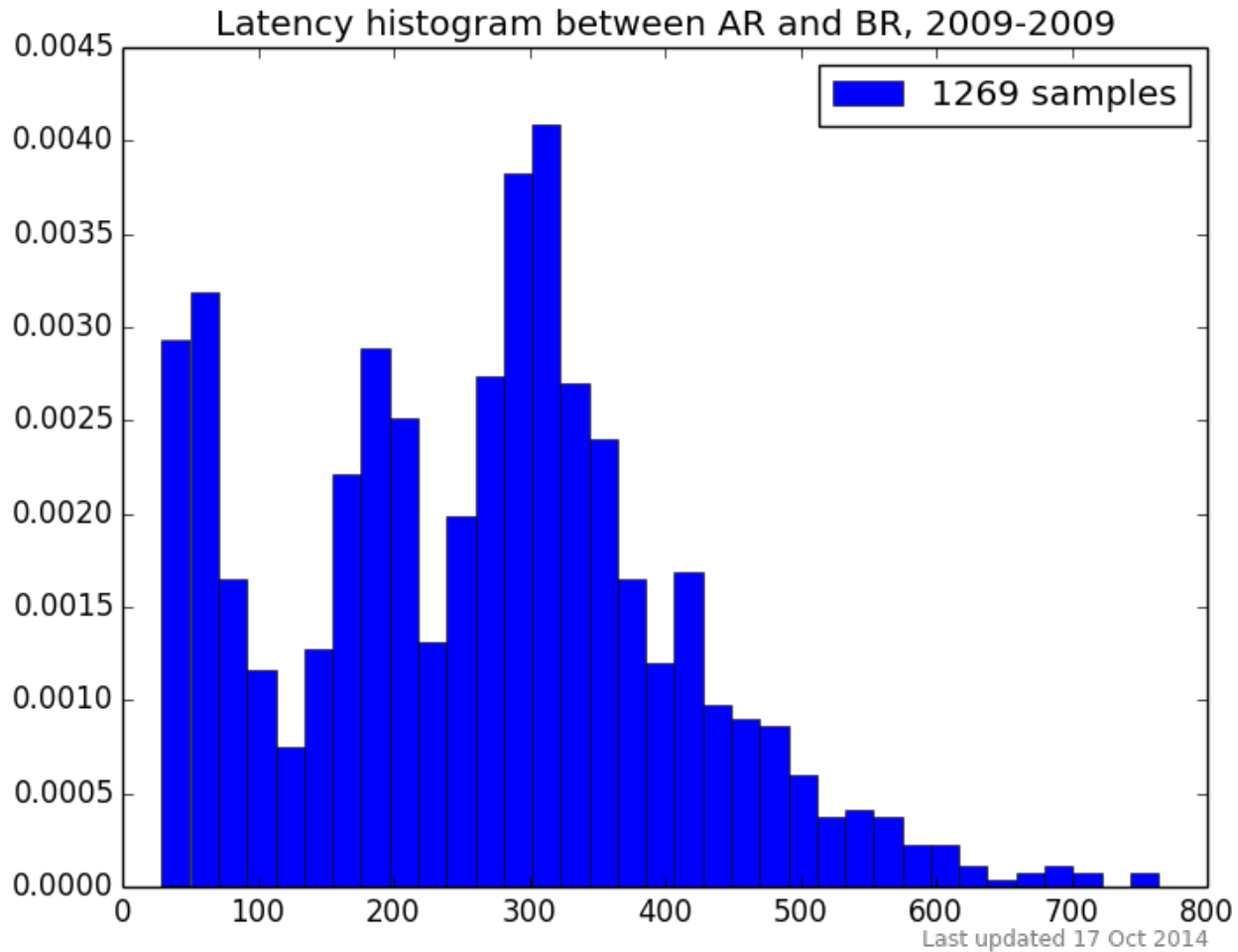


# Latency matrix (Applet)

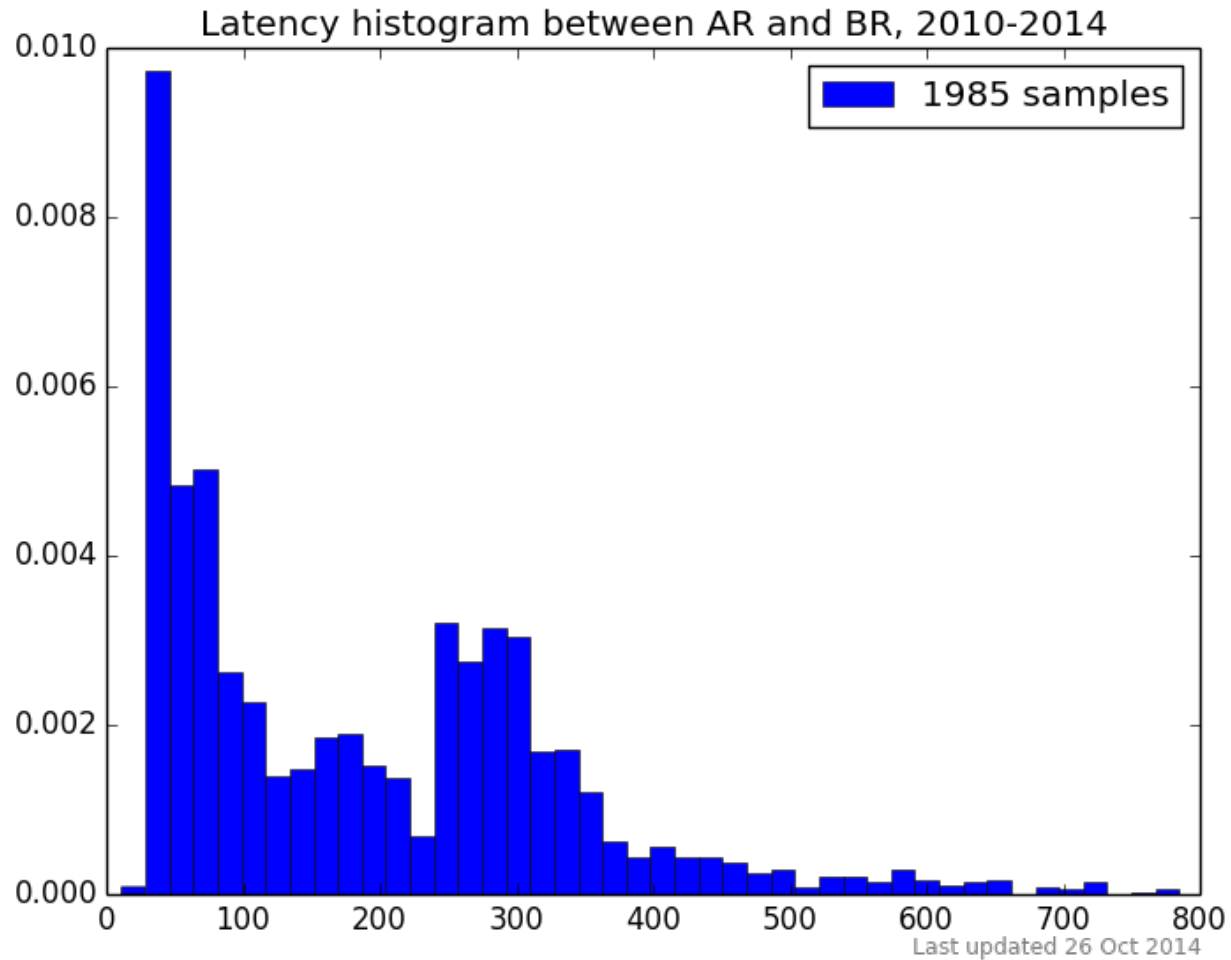
---



# Conclusions (Applet)



# Conclusions (Applet)



# Looking for better tools (1)

---

- Server – server?
- JavaScript as a TCP RTT tester
- Resources in the wild
  - Speedtest sites
  - Web sites
  - Internet user traffic
- More users, more samples
- More sites, more test points
- No user intervention
- Configurable





# Looking for better tools (2)

---

- Client – server
  - Yahoo! Boomerang, performance web scripts
  - Client – server link only!
- Client – servers: this would be useful
  - Many links



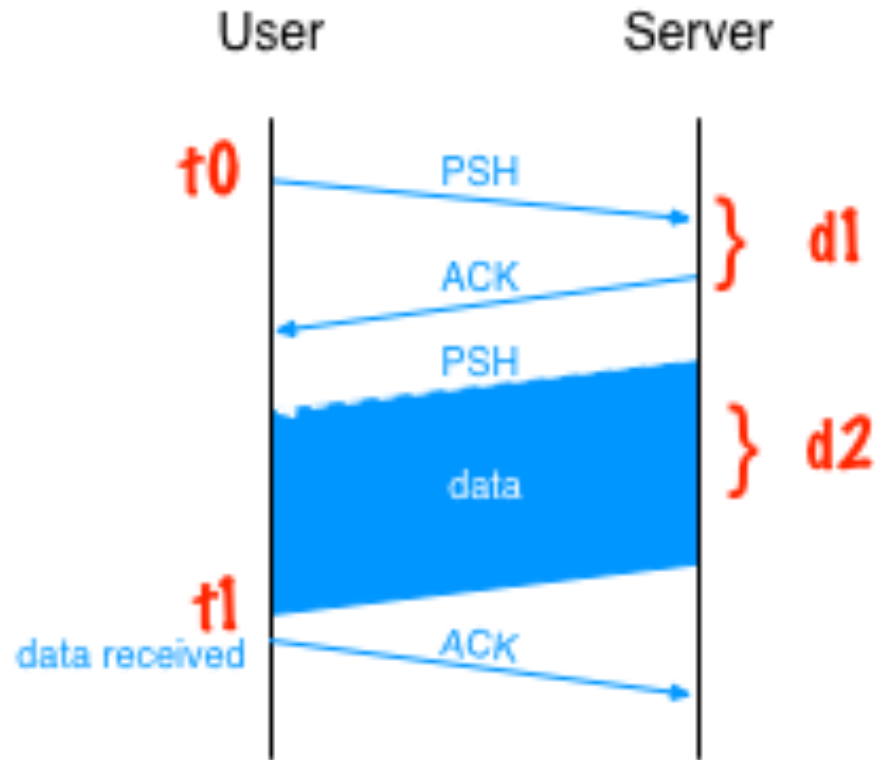
# Looking for better tools (3)

---

1. Get the test point
2. Check it's online
  1. HTTP GET at "/" (TCP handshake)
3. Latency measurement
  1. HTTP GET at "\$RANDOM" x times (TCP PUSH + ACK)
  2. IQR filtering
4. POST
5. Some handles
  1. ¿More points? ¿your own points? ¿Chart? ¿POST to your own database?
  2. before\_start, after\_end, before\_each, after\_each



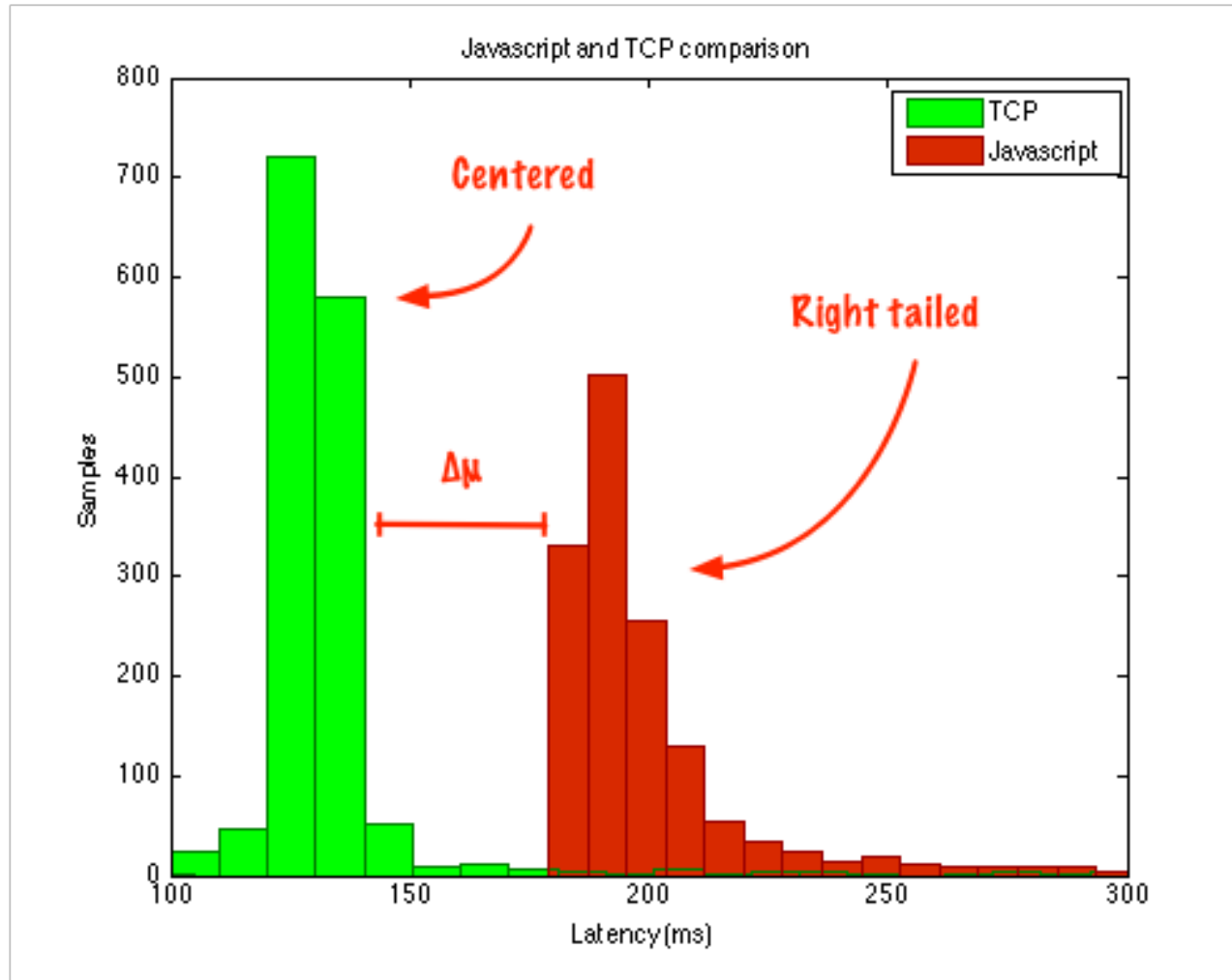
# JavaScript and TCP latency (1)



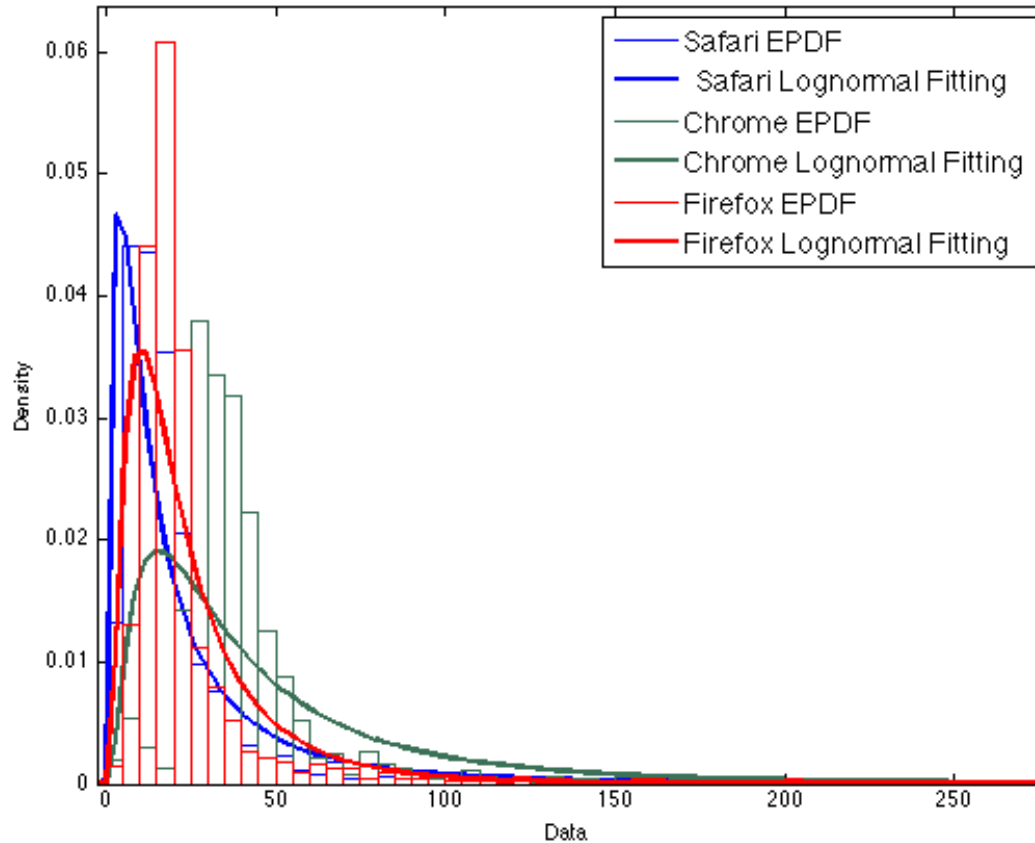
If  $d_1 \rightarrow 0$  and  $d_2 \rightarrow 0$ , then  
 $RTT = t_1 - t_0$



# JavaScript and TCP latency (2)



# Browsers and Javascript performance



	$\Delta\mu$ (ms)	$\Delta\sigma$ (ms)
Blue	0	0
Red	15	5
Green	18	13

Browsers normalization table  
(example values)



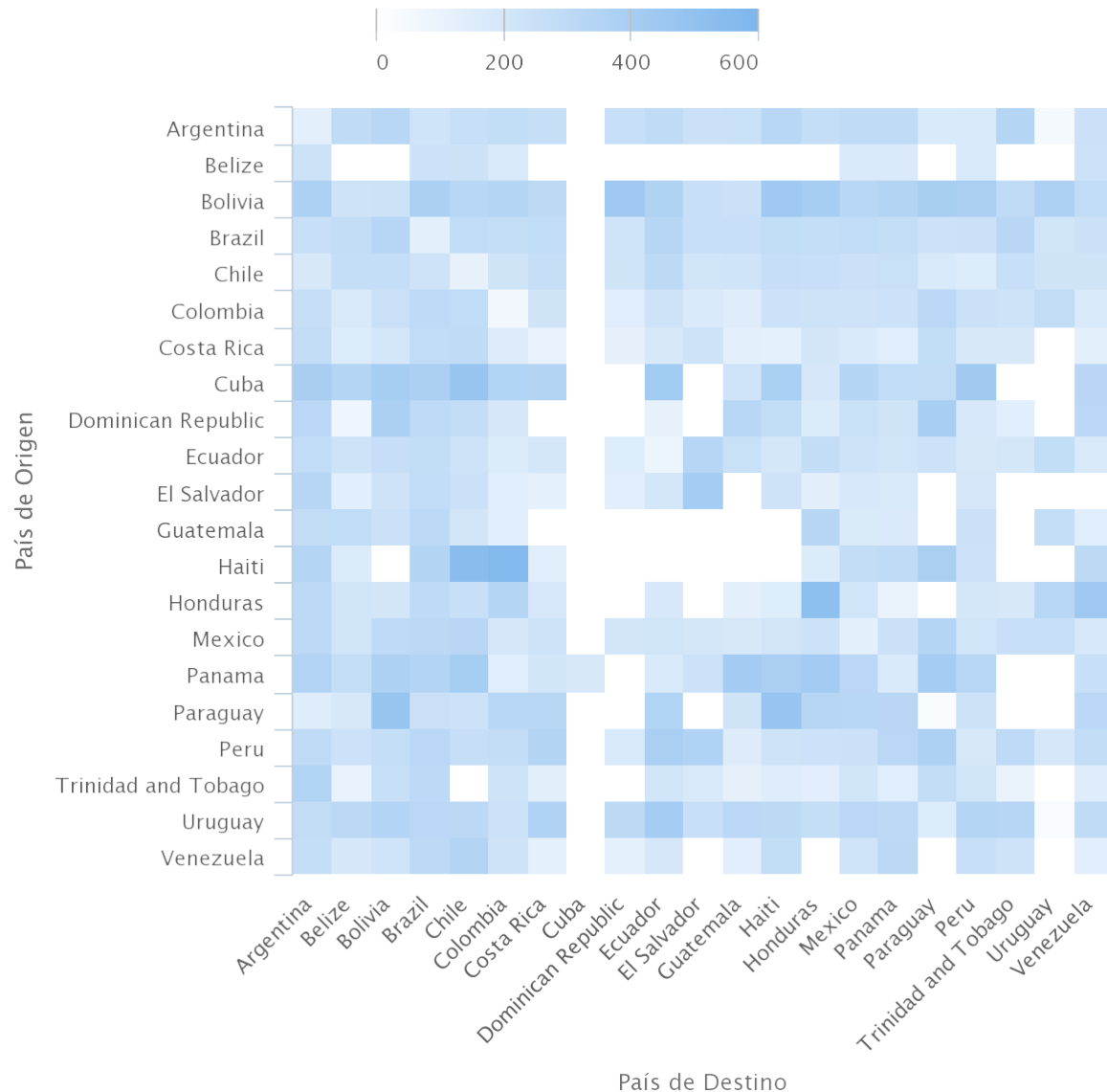
# Some problems

---

- Solved
  - Tabs, windows, and even desktops in “energy saving mode”
  - Unwanted network errors
  - Solution: IQR and sensible timeouts
- To be solved
  - Congested networks
  - Wireless links
  - Anybody experimenting with this stuff in JavaScript?



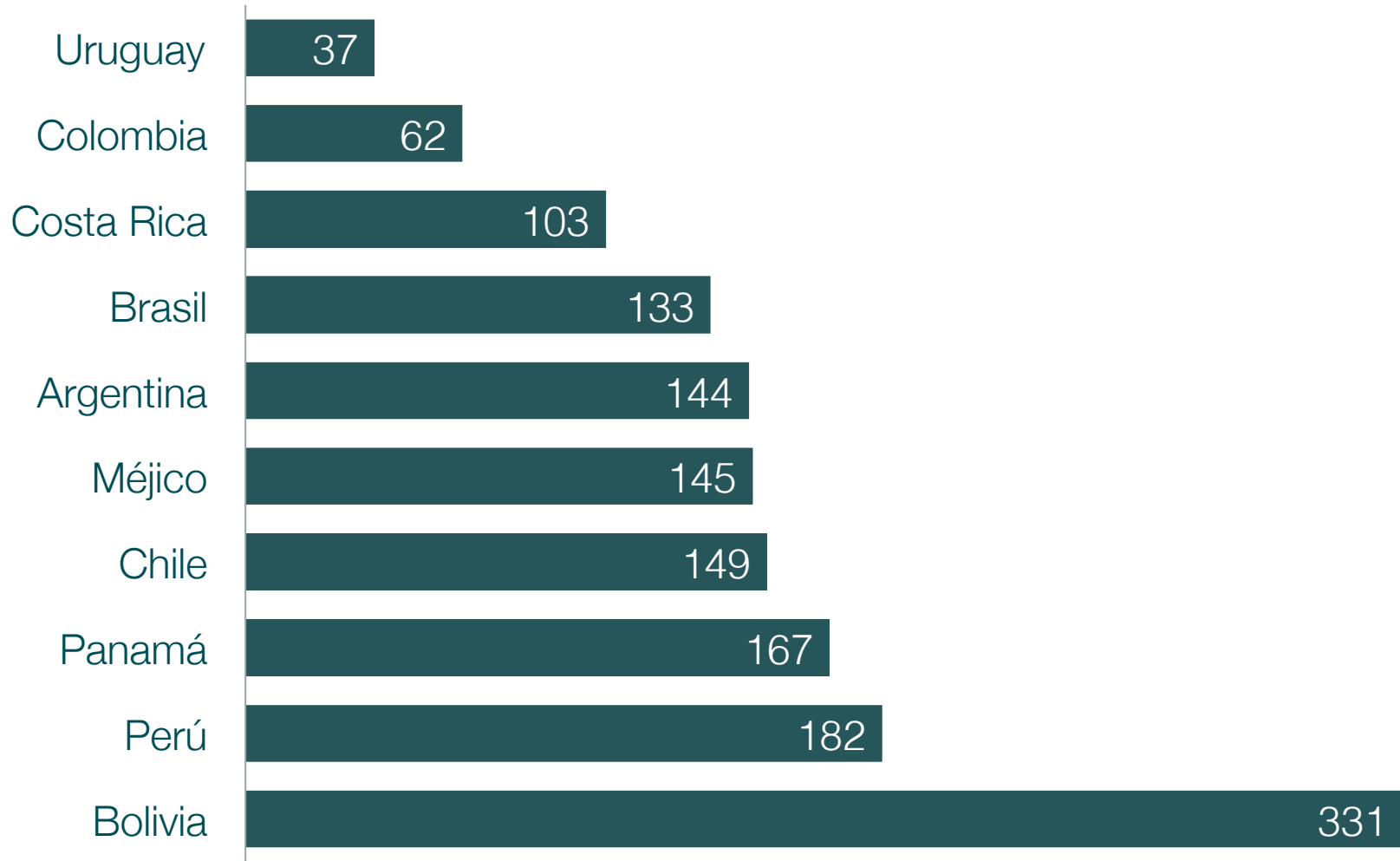
# Latency Matrix (JavaScript)



Highcharts.com

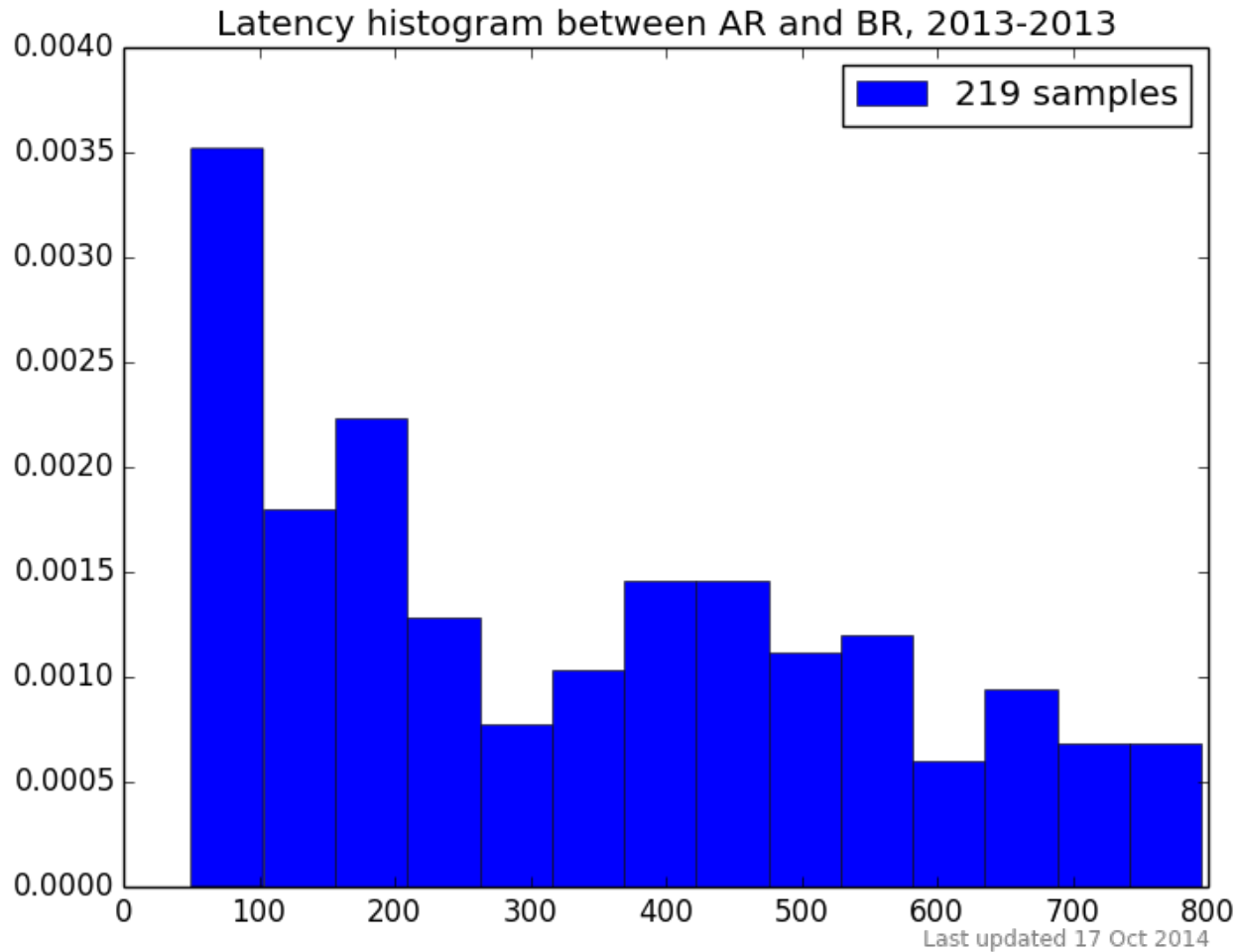


# Latency Matrix (JavaScript)

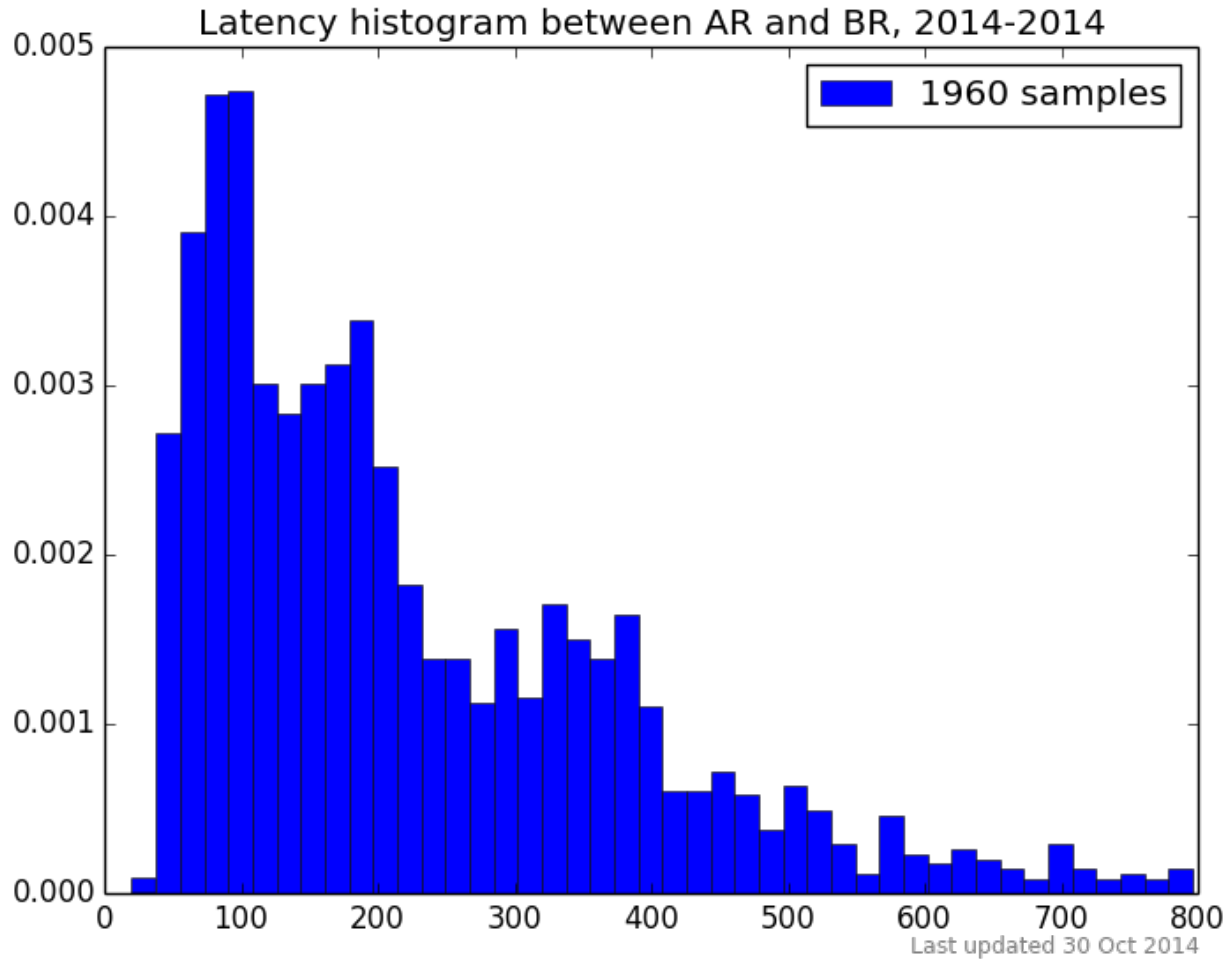




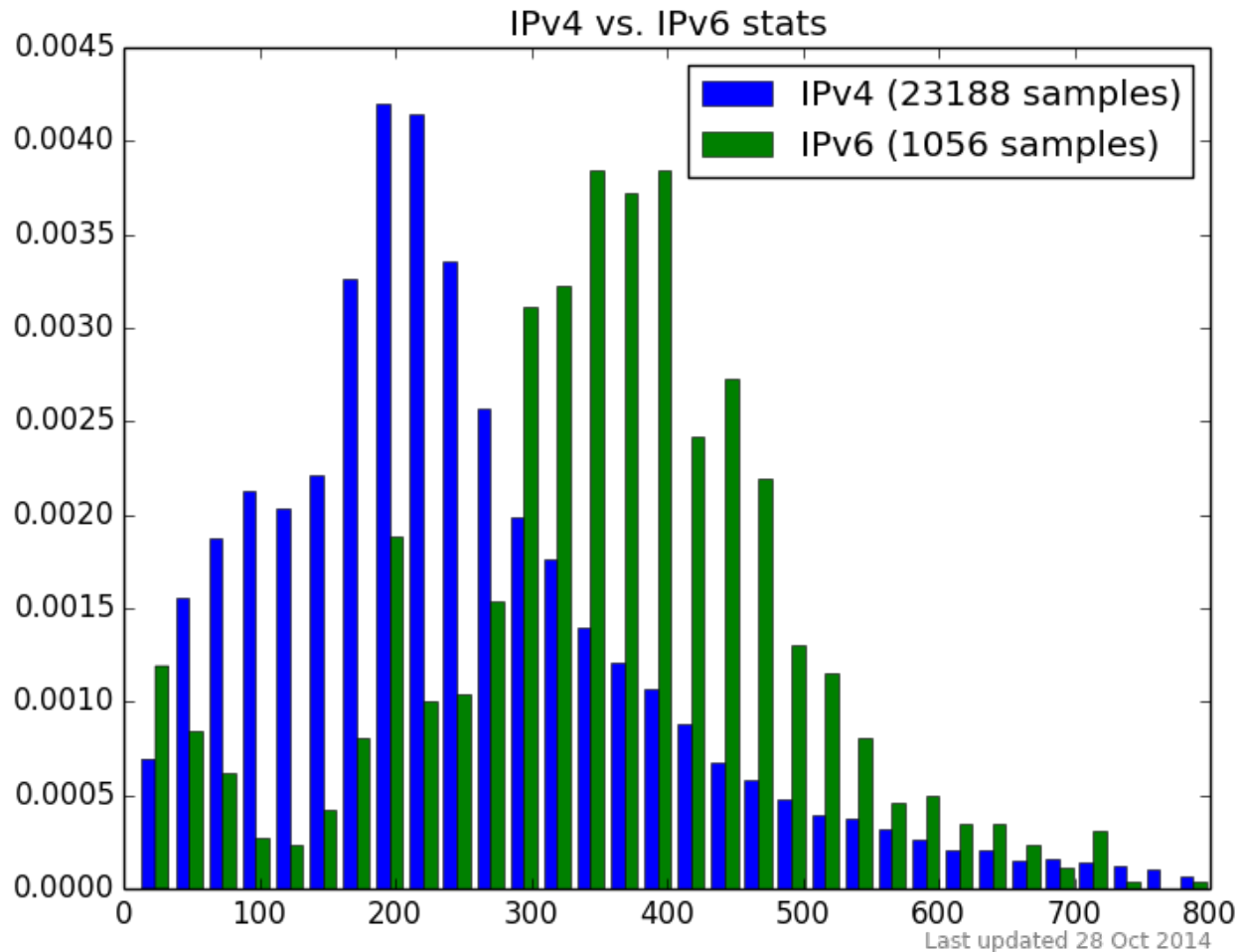
# Conclusions (JavaScript)



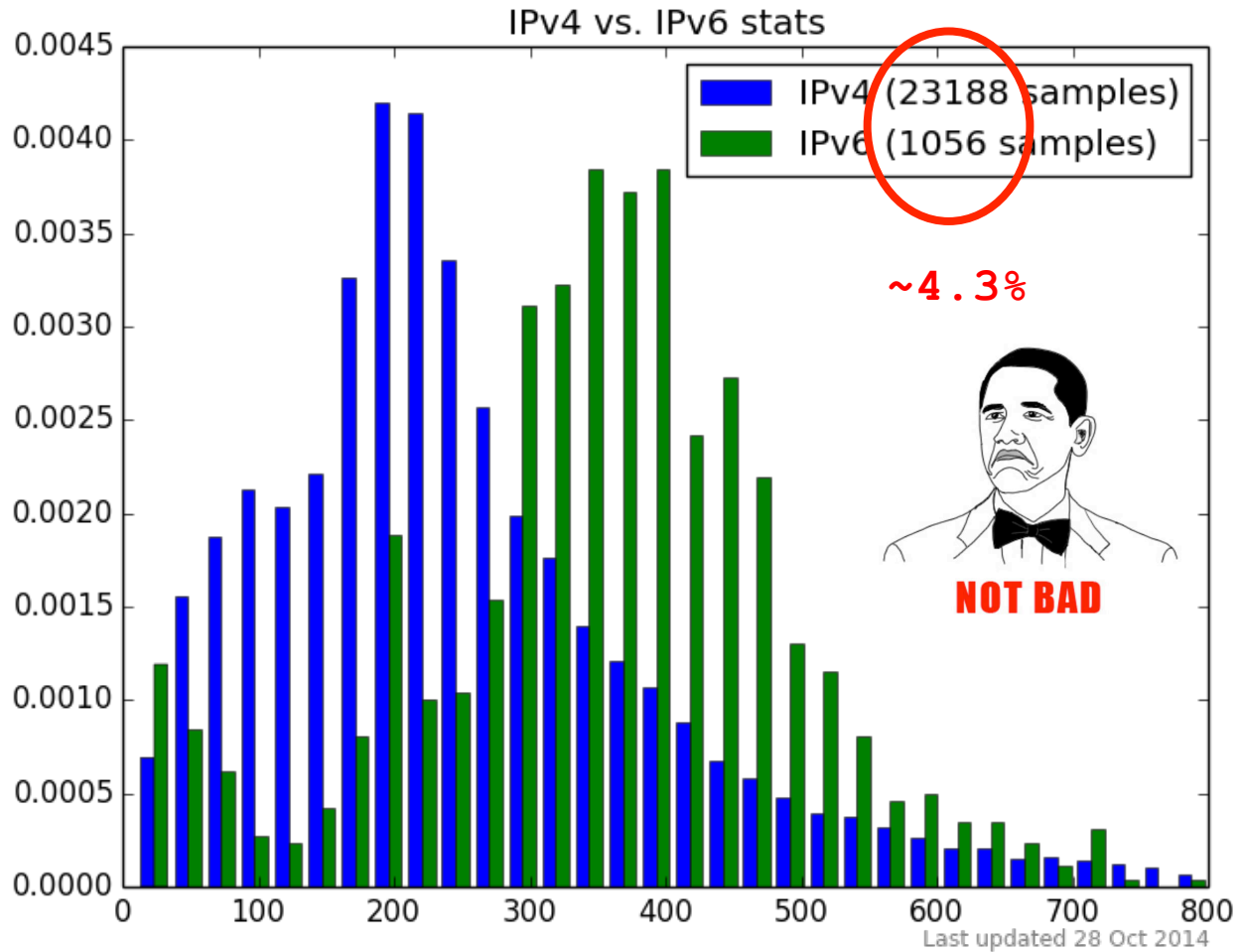
# Conclusions (JavaScript)



# Conclusions (JavaScript)



# Conclusions (JavaScript)



# Conclusions (JavaScript)

---

- Volume and test quality
  - ~500 daily tests....
  - ....to ~600 test points in LAC
  - Highly scalable
  - No manual effort
- “HTTP GET” latency
  - In AS: 51 ms
  - In country: 126 ms
  - Regional: 257 ms



# Near future...

---

- Script deployment in LAC
- Rigorous statistical approximations
- AS level stats
- Build OS + browser → TCP table
- JavaScript API Navigation Timing
- Integrate major measurement projects
  - RIPE ATLAS
  - CAIDA Archipiélago
  - Google M-Lab
  - Others welcome!



# Special thanks

---

- Lacnic Software Development Team
- CAIDA Archipiélago Project
- You!

[simon.labs.lacnic.net](https://simon.labs.lacnic.net)



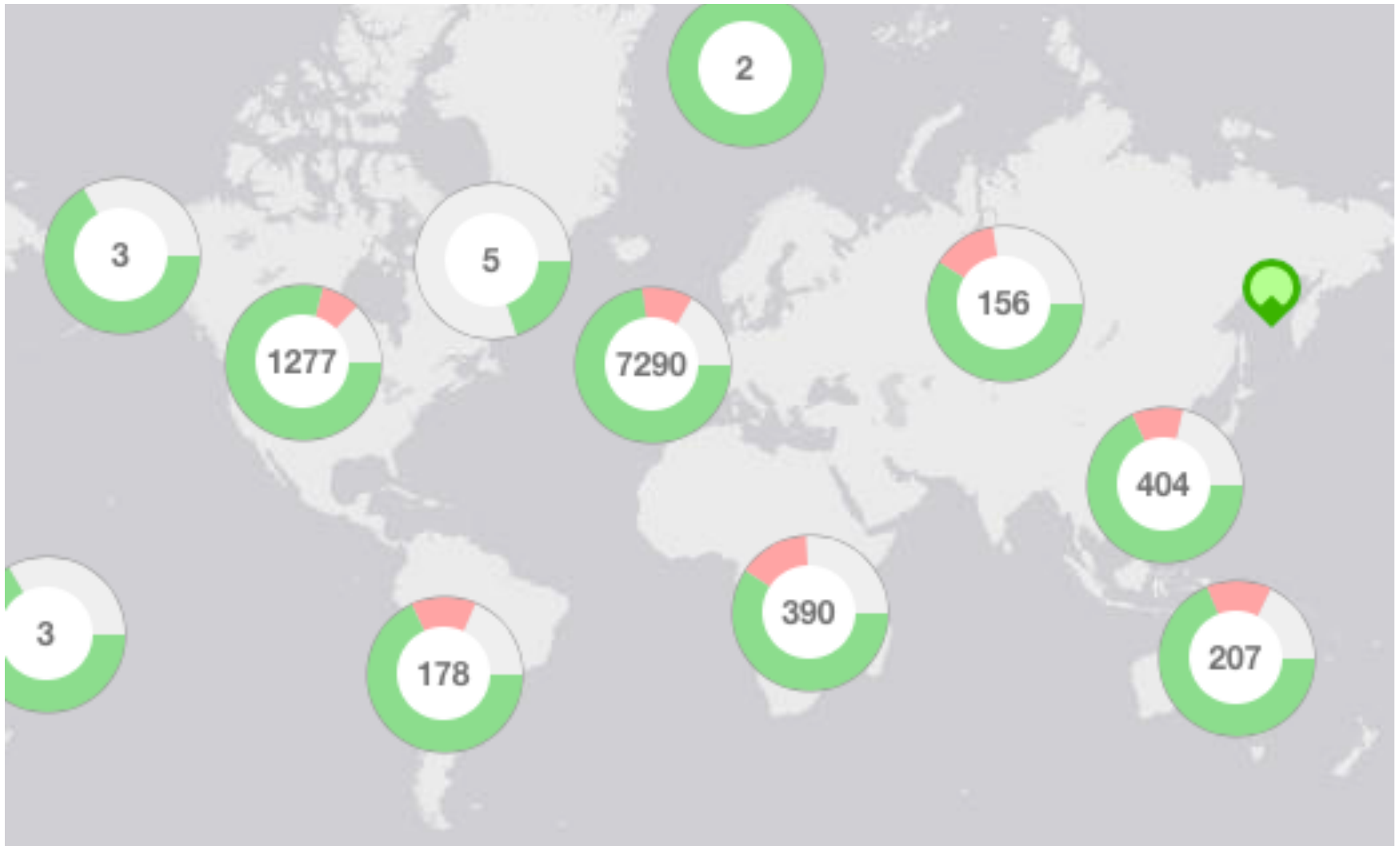
[@proyectosimon](https://twitter.com/proyectosimon)



[LACNIC/simon](https://github.com/LACNIC/simon)



# RIPE ATLAS at LAC?





# RIPE ATLAS at LAC?

