IPv6 Reliability Measurements

May 2019

Geoff Huston

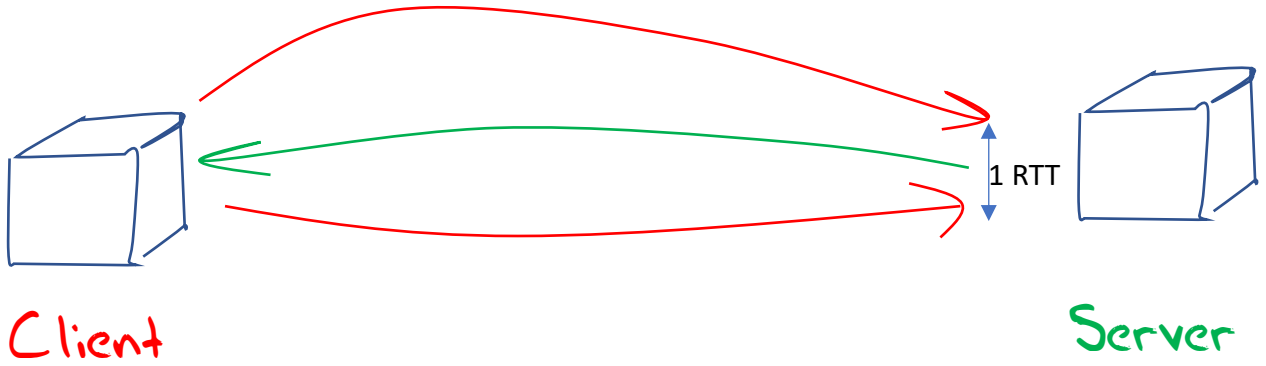
APNIC

The Measurement

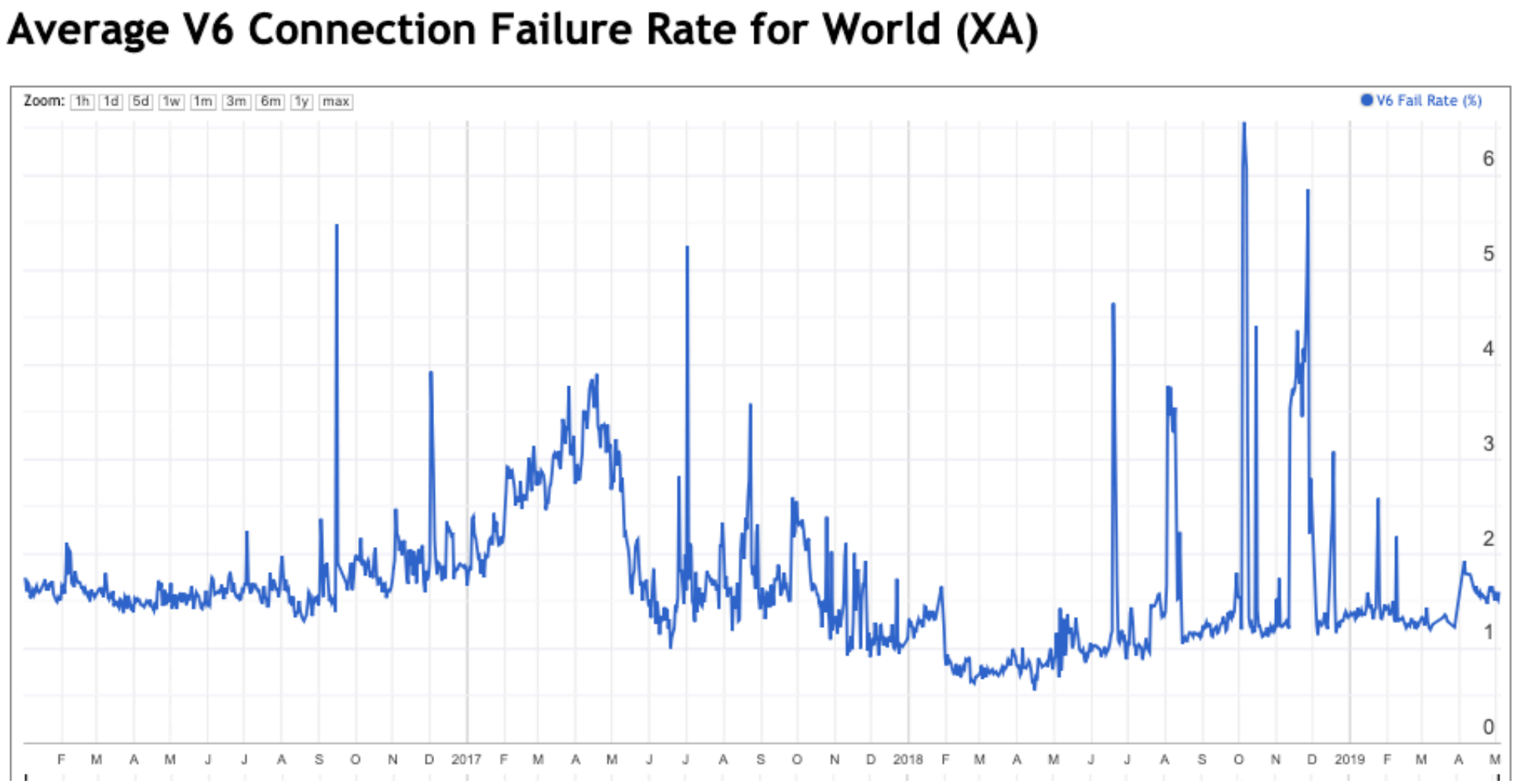
- The endpoint retrieves two URLs from the same remote server – one using IPv4 and the other using IPv6
  - Unique DNS names and TLS are used to ensure that caching does not play a role in the measurement
- We perform full packet capture at the server

Analysis

- We look at the SYN/ACK exchange at the start of the TLS session
- The time between receipt of the SYN and the subsequent ACK at the server is no less than one RTT between the server and the endpoint (and is a reasonable first order substitute for an RTT)
- A received SYN with no subsequent ACK is interpreted as a failed connection attempt

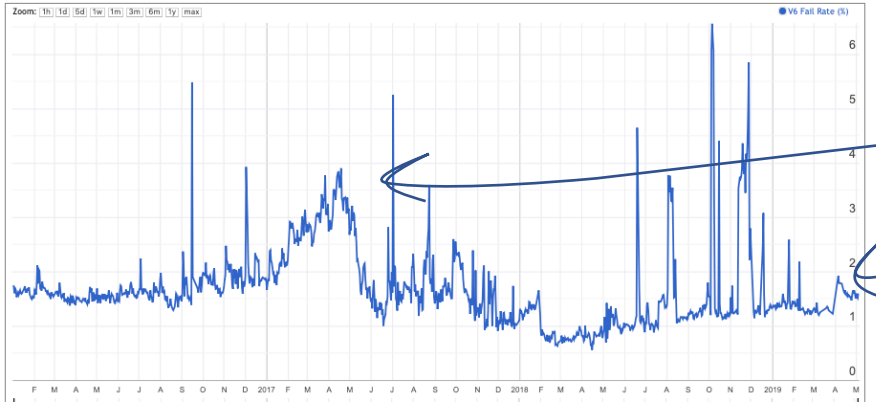


IPv6 TCP Connection Failure



IPv6 TCP Connection Failure

Average V6 Connection Failure Rate for World (XA)



The global failure rate of some 1.4% is better than earlier data (4% failure in early 2017), but its still bad

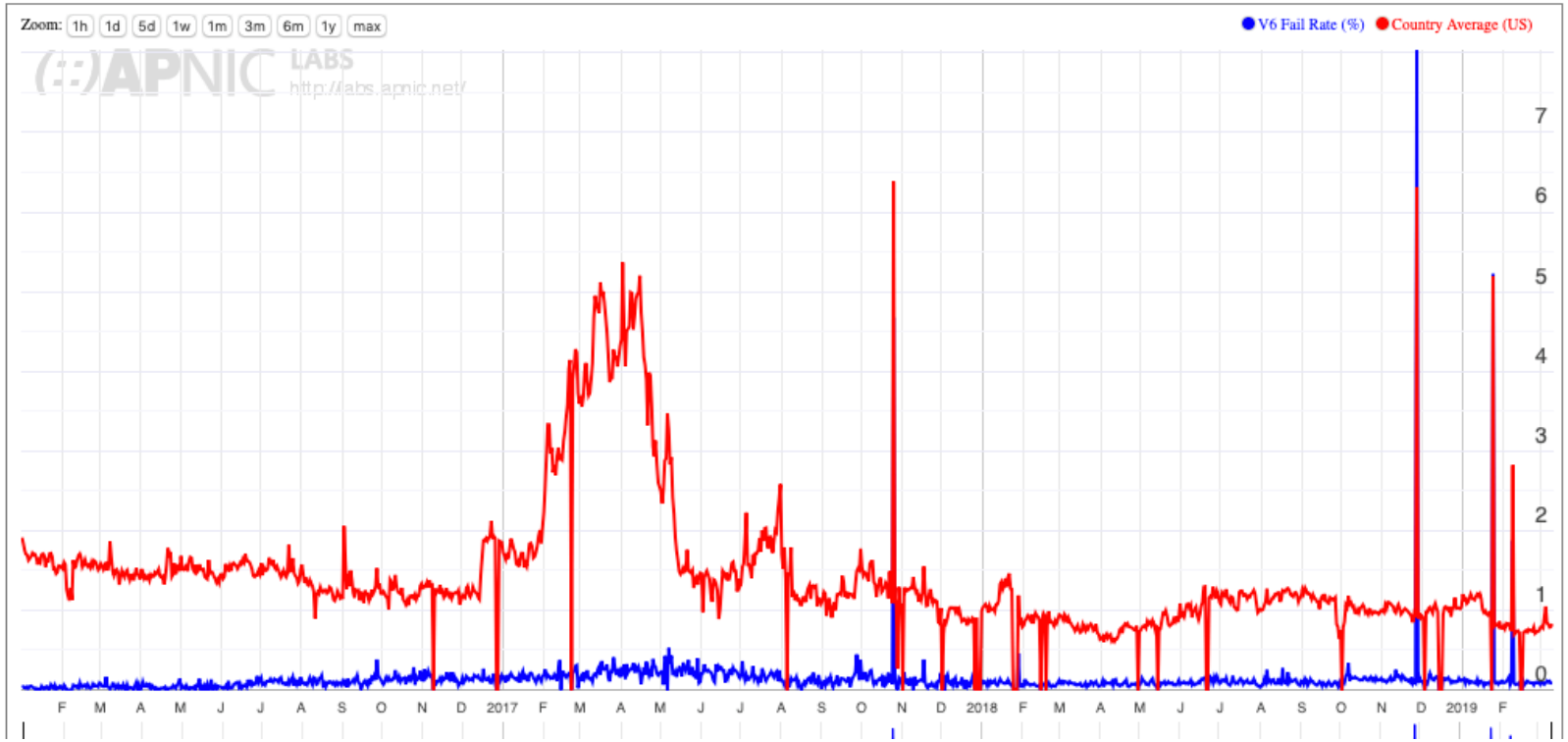
We cannot detect failure in attempting to deliver a packet from the client to the server – what we see as “failure” is a failure to deliver an IPv6 packet from the server to the client

Possible reasons:

- Endpoint using an unreachable IPv6 address
- End site firewalls and filters
- Transition mechanism failure

The Good

V6 Connection Failure Rate for AS21928: T-MOBILE-AS21928 - T-Mobile USA, States of America (US)



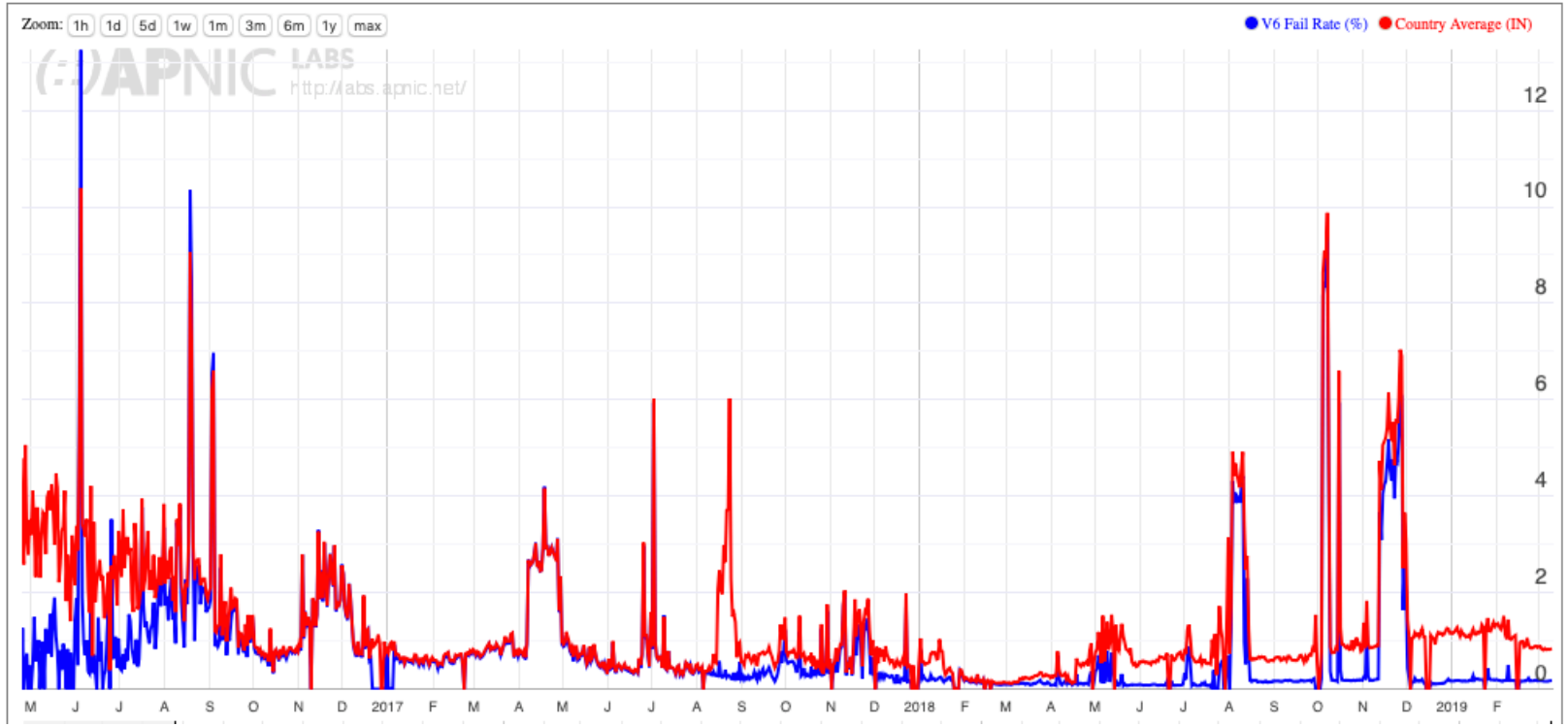
This 464XLAT mobile network (T-Mobile) has remarkably small failure rates – the endpoints are connected via native IPv6 and as this is a mobile network there is only a small amount of customer-operated filtering middleware

464XLAT Performance

- These networks operate in a “native” IPv6 mode
- IPv6 connections to a server require no network processing and no client handling

The Good

V6 Connection Failure Rate for AS55836: RELIANCEJIO-IN Reliance Jio India (IN)



Similar story in India with Reliance JIO – the endpoints are connected via native IPv6 and as this is a mobile network there is only a small amount of customer-operated filtering middleware

The Bad

Average V6 Connection Failure Rate for Vietnam (VN)

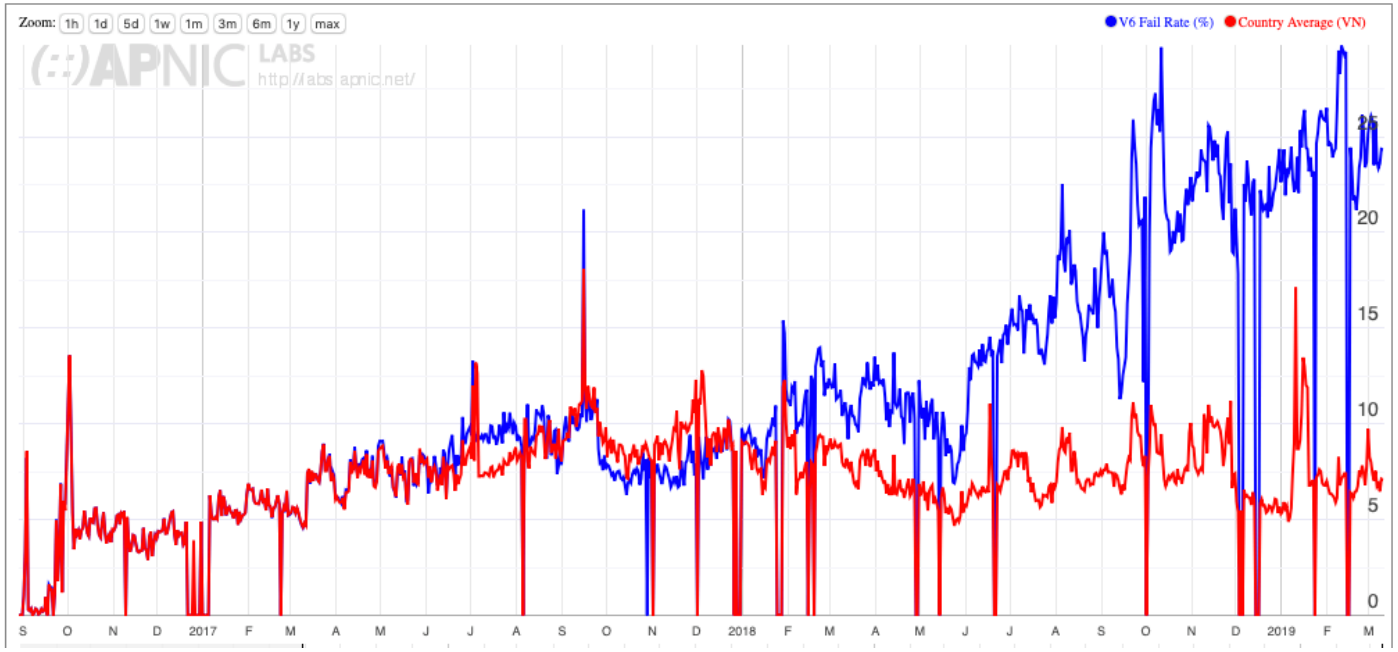
Seriously?

A 6%-10% IPv6 connection failure rate is bad enough

A sustained failure rate for over 2 years seems worse!

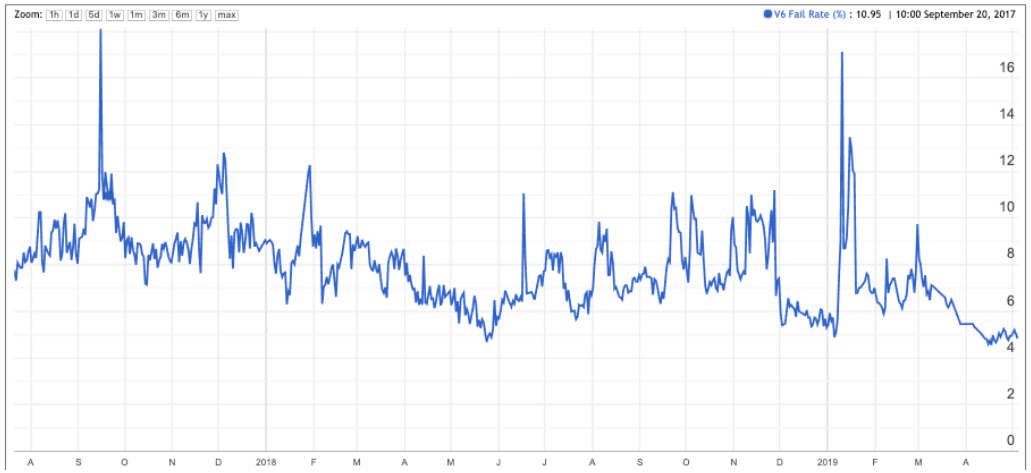
The Appalling!

V6 Connection Failure Rate for AS18403: FPT-AS-AP The Corporation for Financing Promoting Technology, Vietnam (VN)



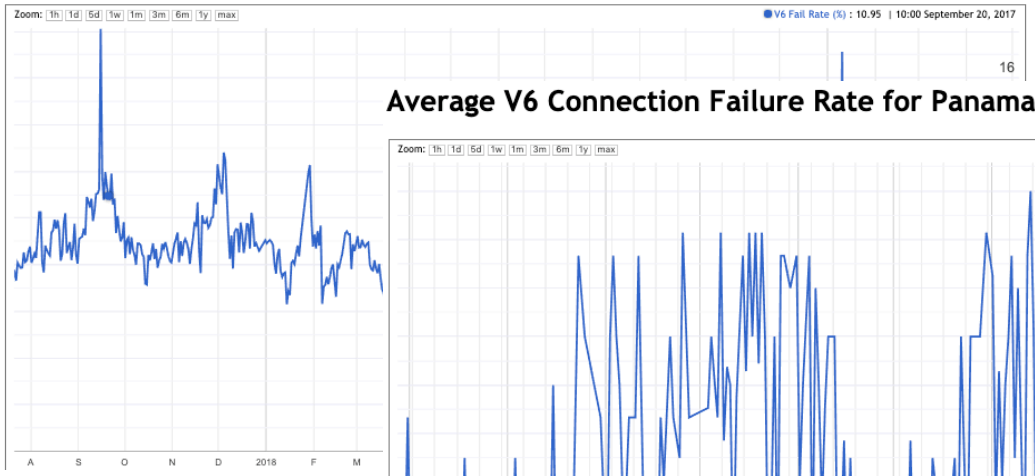
High IPv6 Failure Levels

Average V6 Connection Failure Rate for Vietnam (VN)

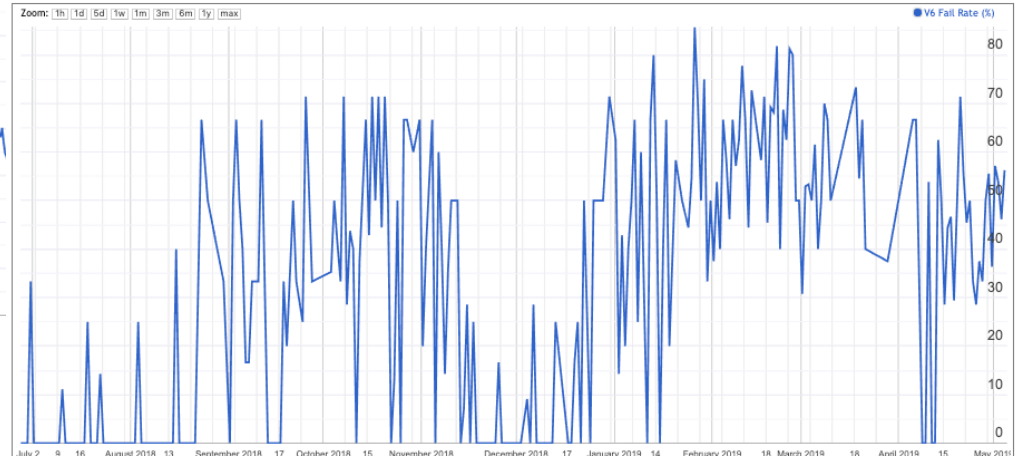


High IPv6 Failure Levels

Average V6 Connection Failure Rate for Vietnam (VN)

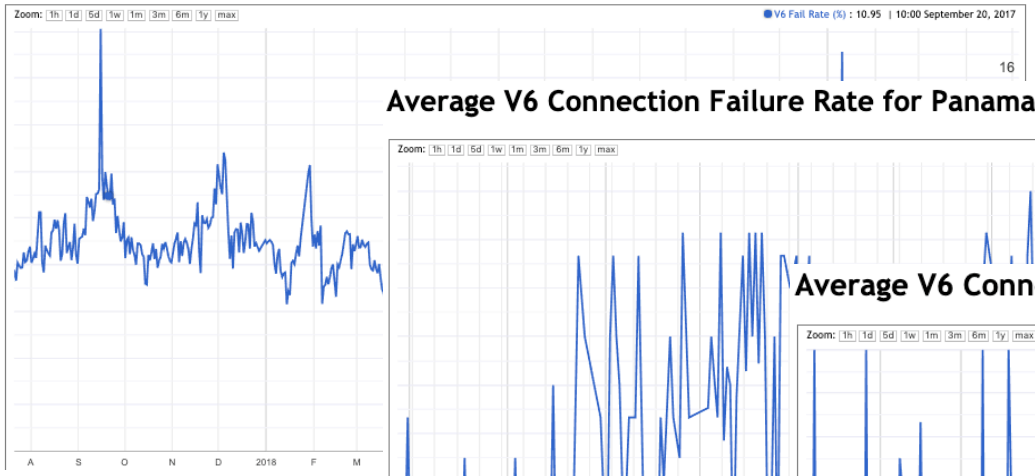


Average V6 Connection Failure Rate for Panama (PA)

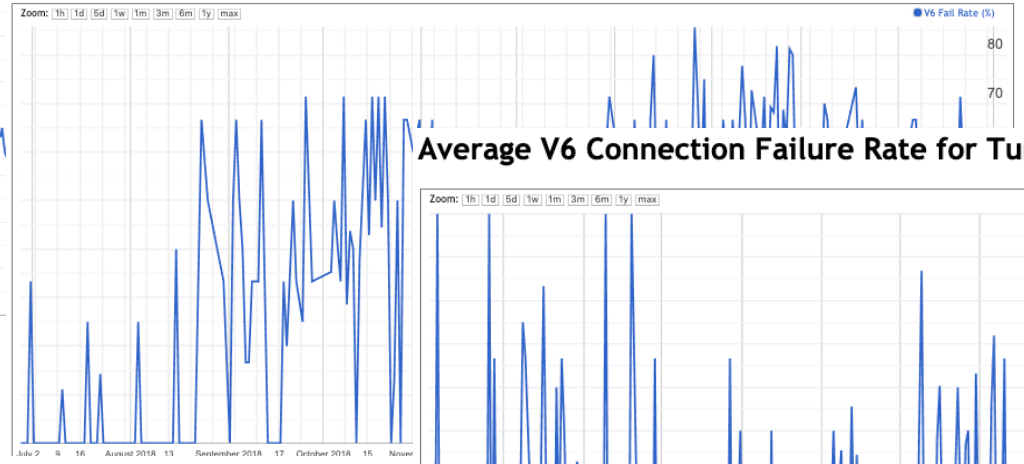


High IPv6 Failure Levels

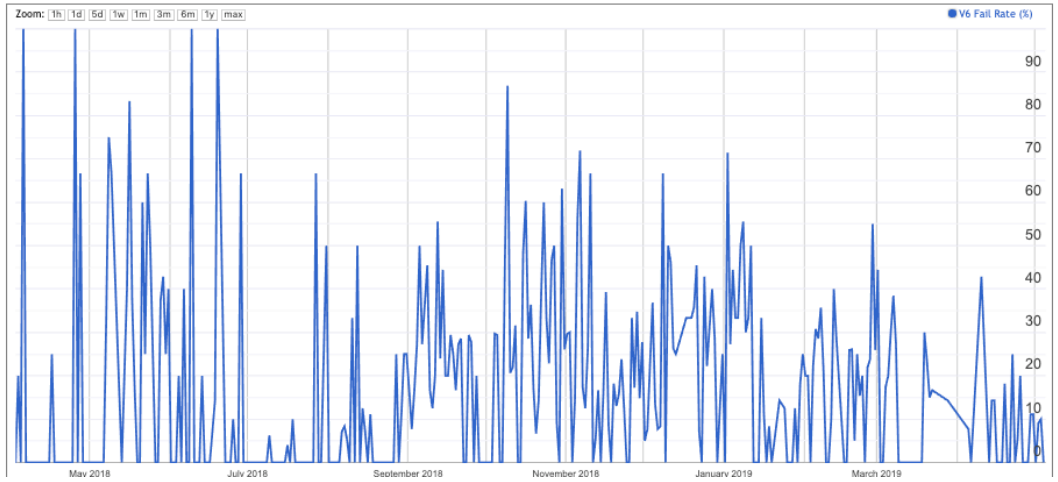
Average V6 Connection Failure Rate for Vietnam (VN)



Average V6 Connection Failure Rate for Panama (PA)

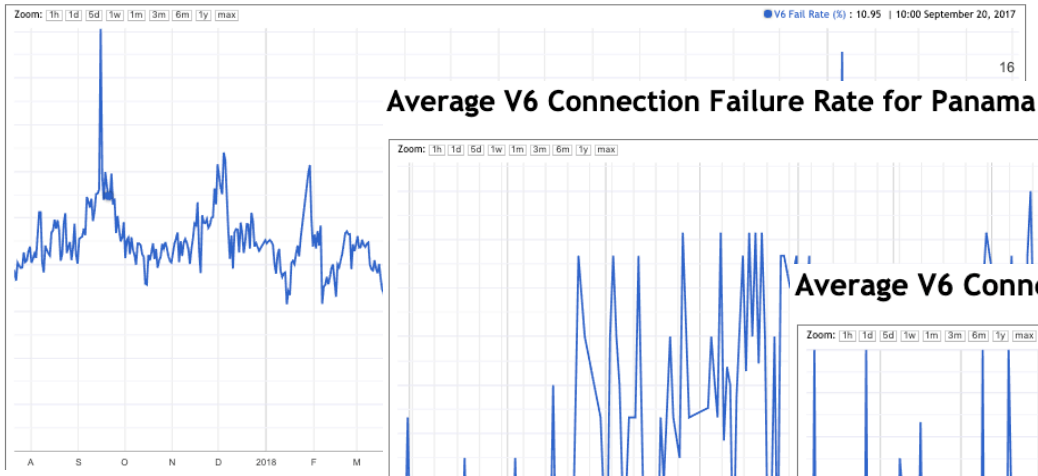


Average V6 Connection Failure Rate for Turkey (TR)

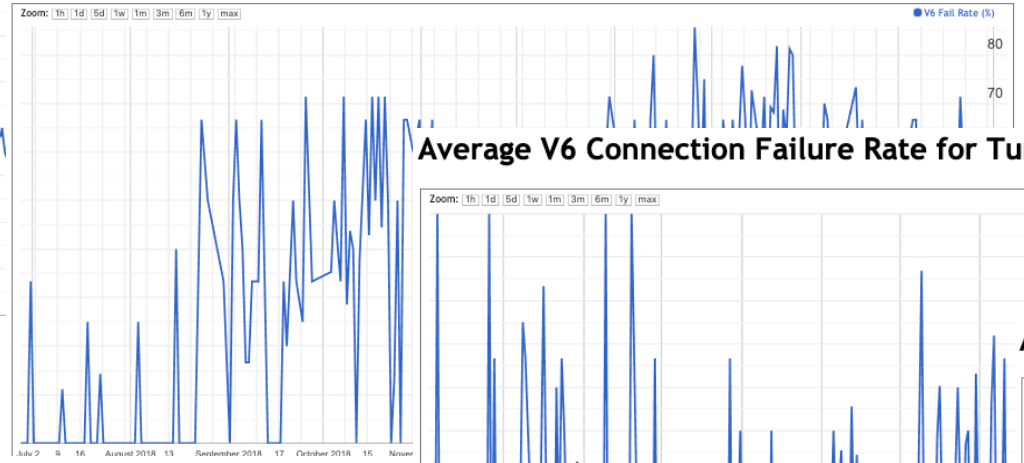


High IPv6 Failure Levels

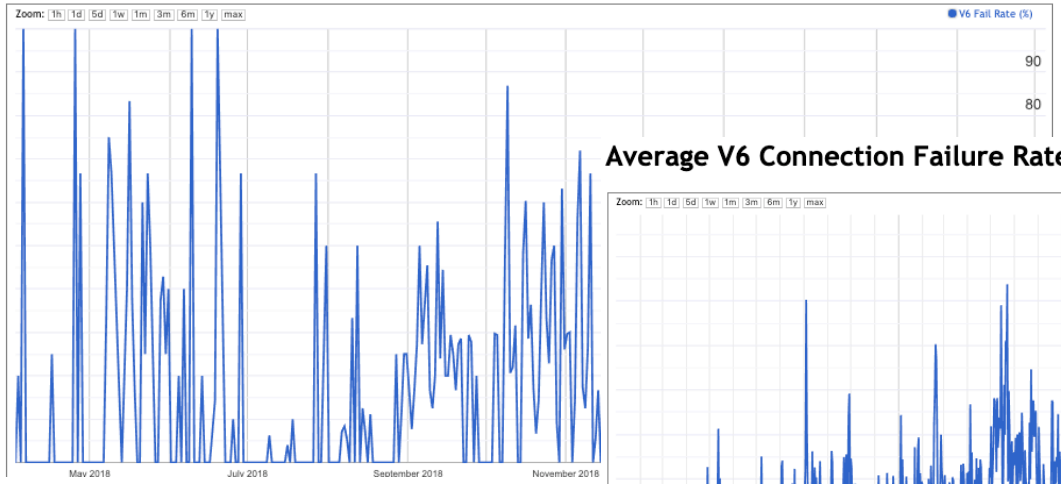
Average V6 Connection Failure Rate for Vietnam (VN)



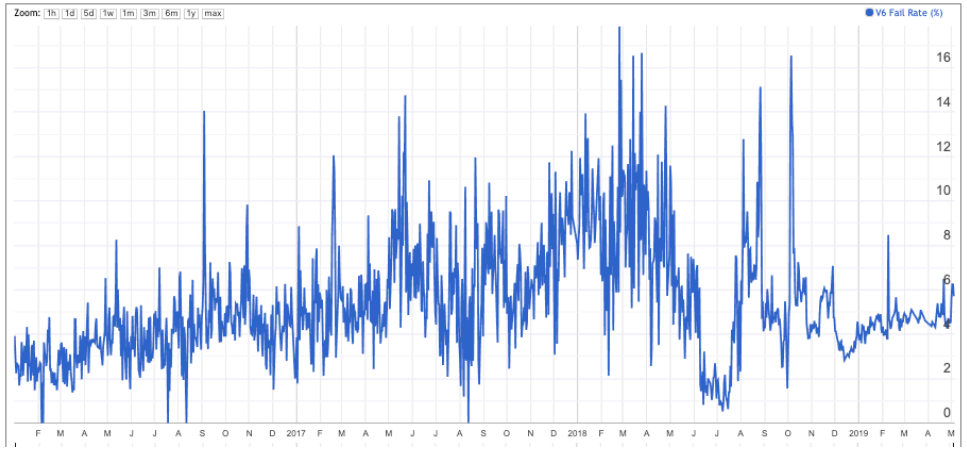
Average V6 Connection Failure Rate for Panama (PA)



Average V6 Connection Failure Rate for Turkey (TR)



Average V6 Connection Failure Rate for China (CN)



Comment

- For many end users in Vietnam, Panama, Morocco, Turkey, Venezuela, China and Bangladesh their IPv6 service looks pretty broken
  - The combination of Dual Stack and Happy Eyeballs masks the problem so that the user does not experience a degraded service
  - But this only will work while Dual Stack is around
- Other ISPs have managed to do a much better job, such as in the United States, Sweden, Thailand and Korea and the IPv6 connection failure rates are close to experimental noise levels
- What's happening in the second set of countries and ISPs that is NOT happening in the first set?

Transition Technologies

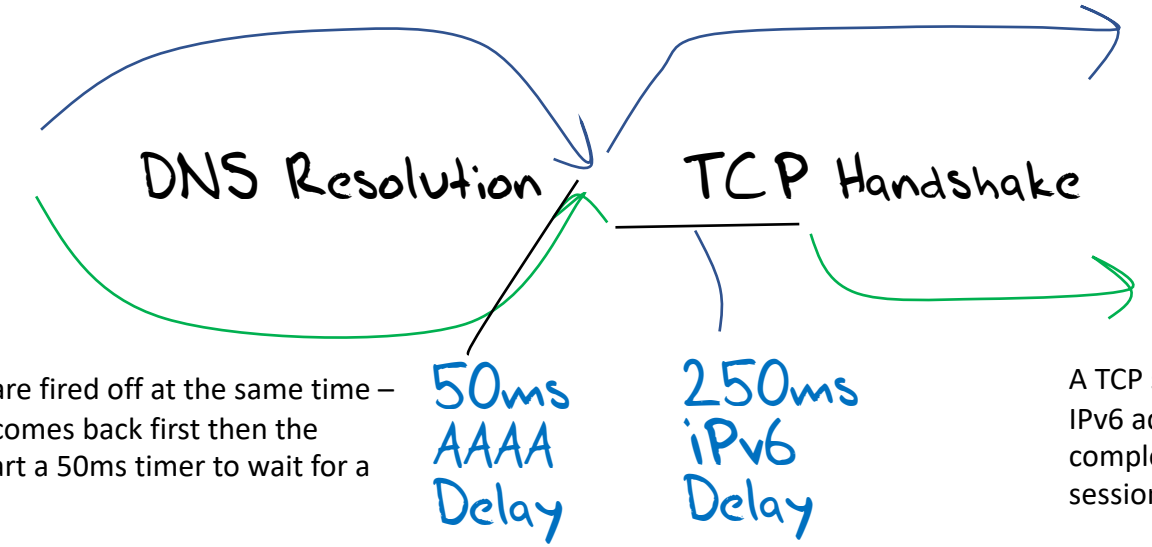
- Stateful transition technologies that involve protocol translation show higher levels of instability
- Translation technologies that require orchestration of DNS and network state are also more unstable

Dual Stack is NOT the Goal

- Despite all the grim predictions that IPv4 will be around for a long time to come, the aim of this transition is NOT to make Dual Stack work optimally
- The goal is to automatically transition the network to operate over IPv6
- The way to achieve this is for client systems to prefer to use IPv6 whenever it can

Happy Eyeballs

- An unconditional preference for IPv6 can lead to some very poor user experience instances
  - Linux uses a 108 second connection timer, for example
- Applications (particularly browsers) have used a “Happy Eyeballs” approach



DNS A and AAAA are fired off at the same time – if the A response comes back first then the application will start a 50ms timer to wait for a AAAA response

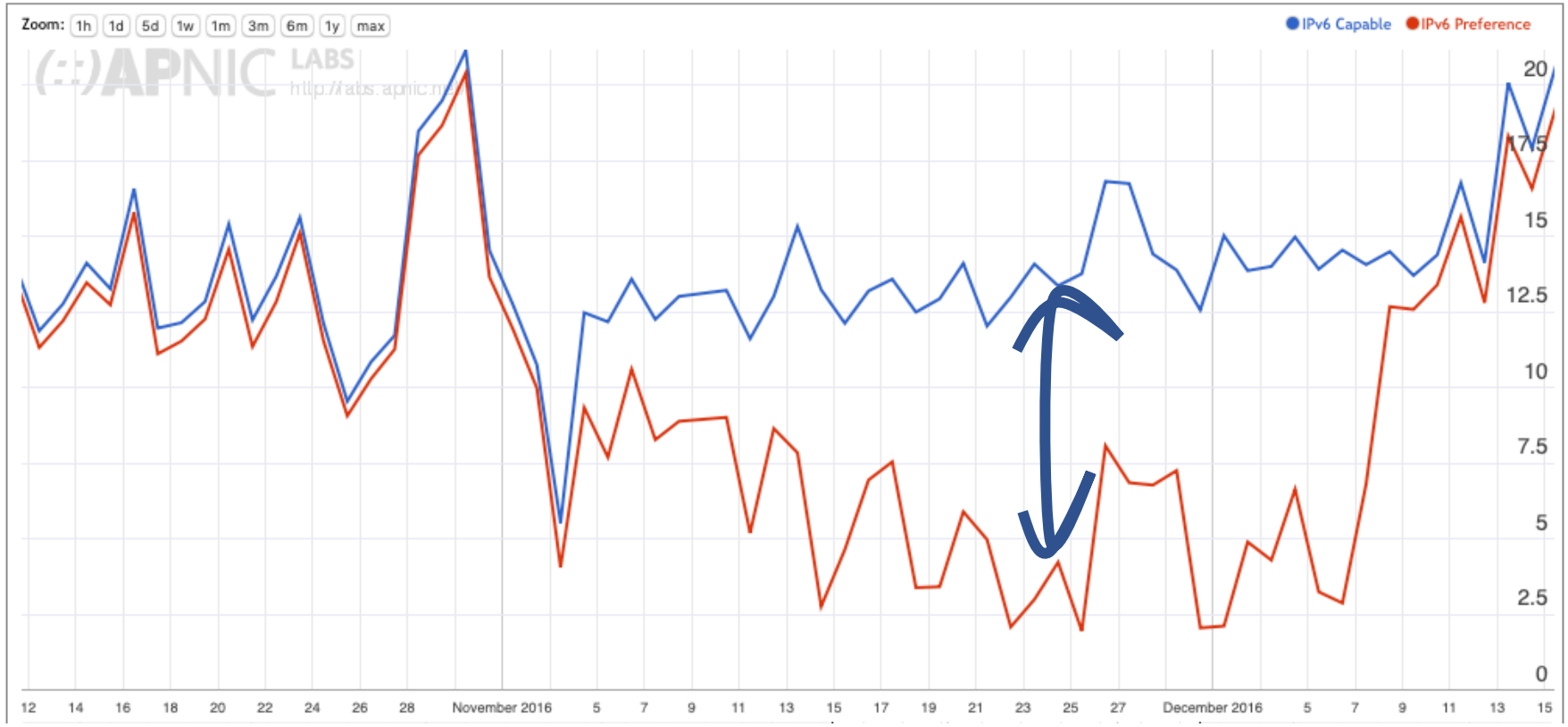
A TCP session will be started in IPv6 if there is a IPv6 address record. If the handshake is not completed within 250 ms then an IPv4 TCP session is also fired off

Tuning IPv6 for Happy Eyeballs

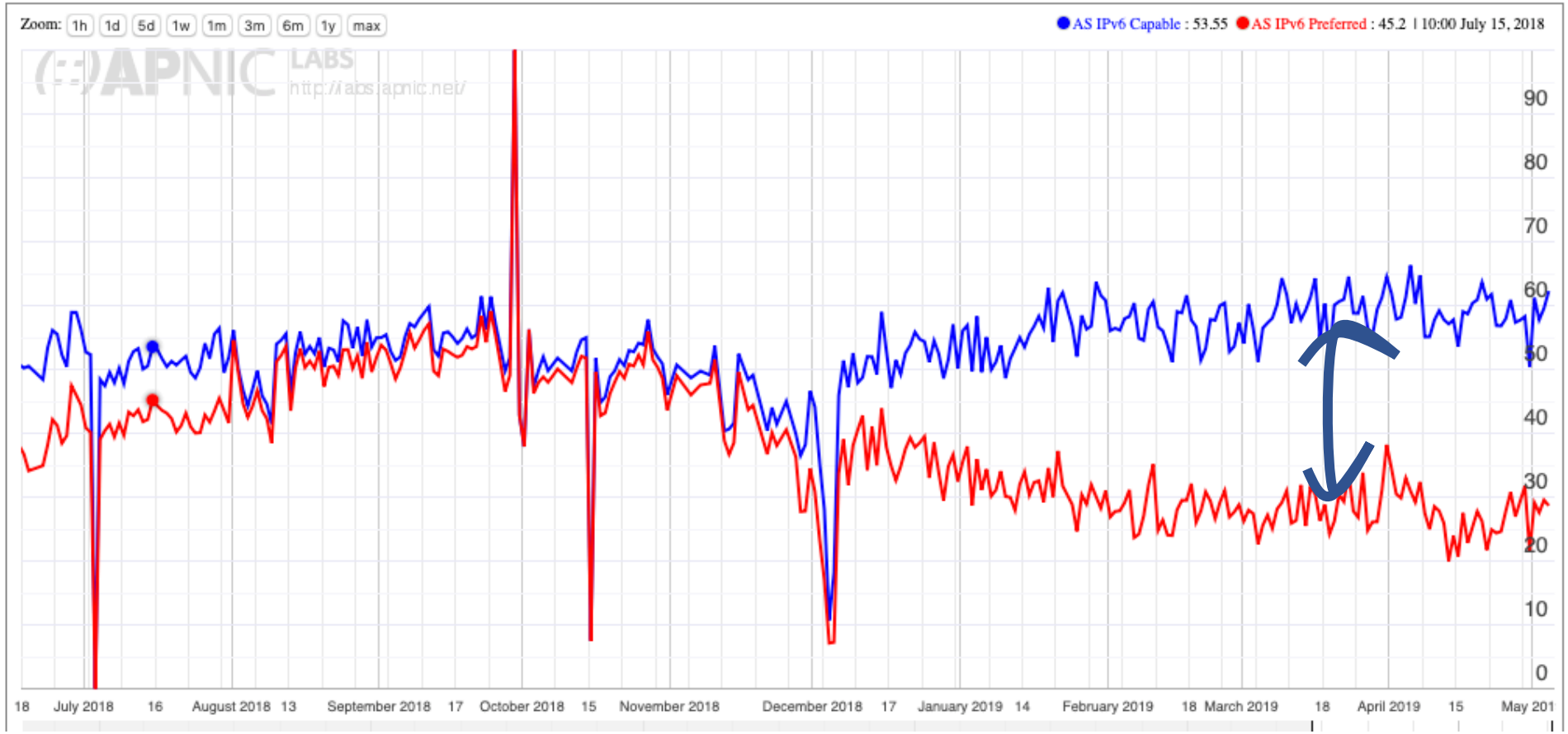
- When connecting to a remote dual stack service, the Routing Path selection for IPv6 should be similar to IPv4
- Where there are path deviations, the path discrepancy should be contained

- This is not always the case…

India, late 2016



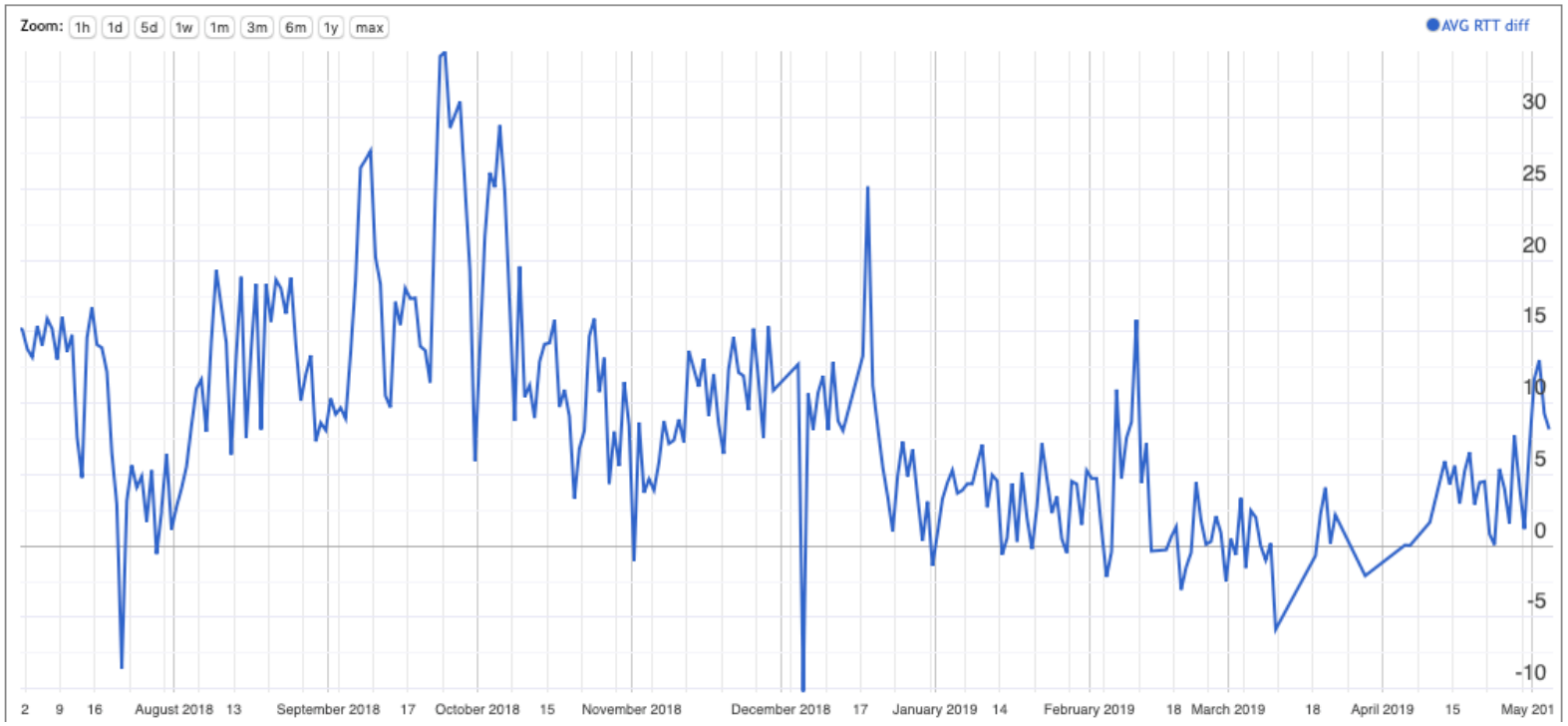
Vodaphone New Zealand - 2019



AS9500

Worldwide Performance

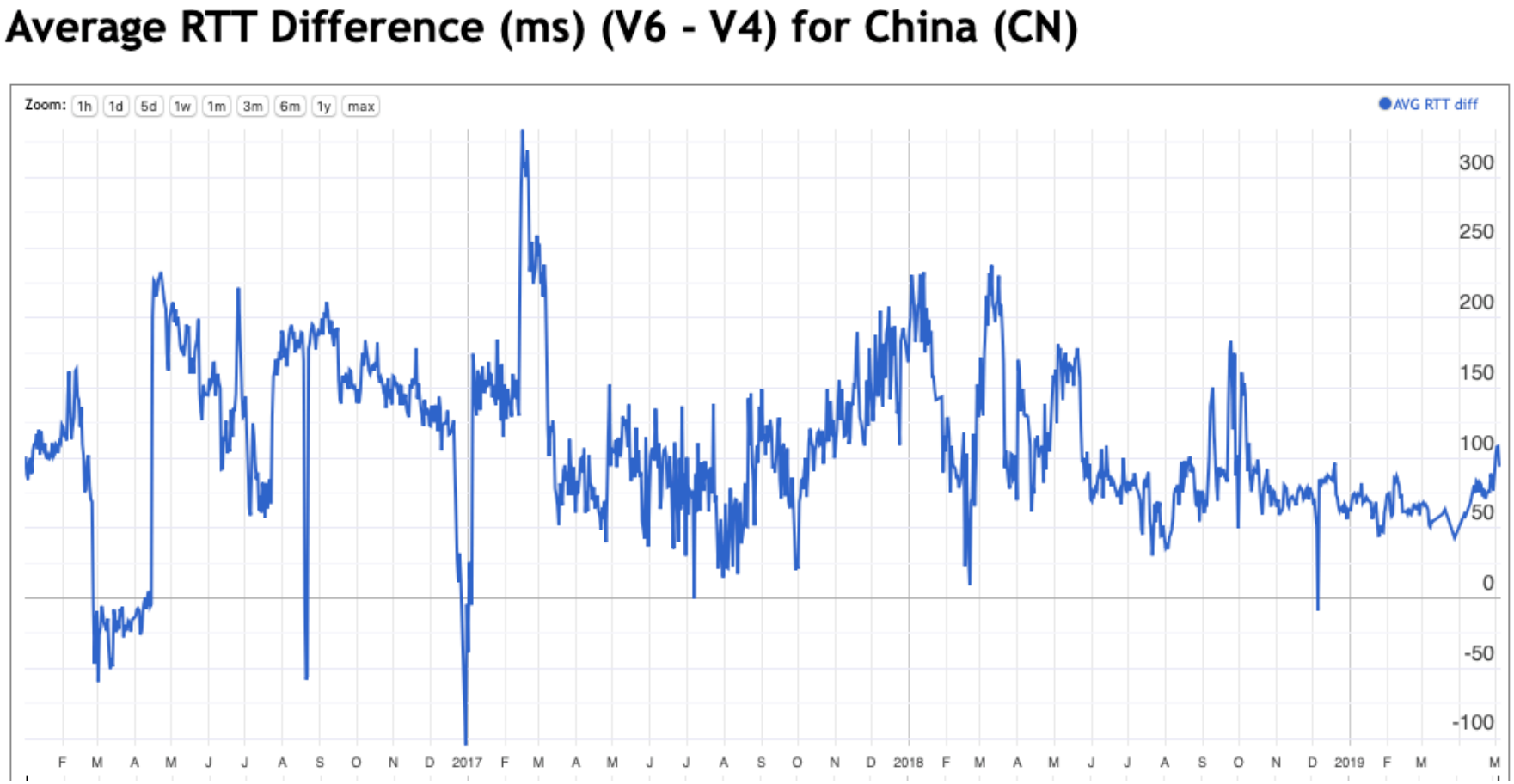
Average RTT Difference (ms) (V6 - V4) for World (XA)



Across the sample set the RTT for IPv4 is on average ~7ms faster than IPv6

This is not a major cause for concern

China's IPv6 Network

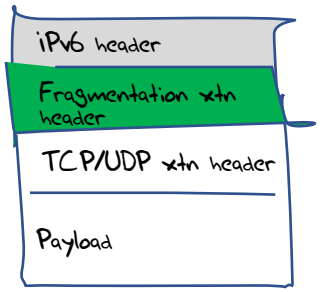


3 Suggestions to Assist IPv6 Robustness

- Avoid stateful IPv6 -> IPv4 transition mechanisms if possible – if you can operate IPv6 in native mode all the better!
- Avoid using IPv6-in-IPv4 encapsulations
  - Not only are tunnels unstable, but the reduced IPv6 MTU may cause problems with extension header based packet discard
- Keep IPv4 and IPv6 paths congruent if possible
  - Yes, this can be really challenging for multi-homed networks, but try to use transit and peer arrangements that are dual stack

But that's not all…

- IPv6 used a new approach to extension headers, including packet fragmentation by inserting them between the IPv6 header and the transport header
- Which means that hardware will have to spend cycles to hunt for a transport header
- Or it can just drop the packet…



2017 Measurement

V6, the DNS and Fragmented UDP

Total number of tests: 10,851,323

Failure Rate in receiving a large response: 4,064,356

IPv6 Fragmentation Failure Rate: **38%**

This measurement test involved sending a fragmented UDP packet to recursive resolvers

2017 Measurement

What about TCP and Fragmentation?

1,961,561 distinct IPv6 end point addresses

434,971 failed to receive Fragmented IPv6 packets

22% failure rate

This measurement test involved sending a fragmented TCP packet to browser endpoints

What can we say?

- There are ongoing issues with IPv6 reliability in many parts of the world
  - This appears to relate to local security policies at the client edge of the network
  - We can expect most of this to improve over time by itself
- But there are also very serious issues with Path MTU management and handling of IPv6 extension headers
  - This is a more challenging issue
  - Should we just avoid IPv6 extension headers?
  - Or try to clean up the IPv6 switching infrastructure?

Thanks!