

# Updates on TCP Fast Open deployment

Praveen Balasubramanian

[pravb-at-microsoft-dot-com](mailto:pravb-at-microsoft-dot-com)



# Recap from IETF 100

- Enabled by default in Edge browser on Fall Creators Update (builds 16299+)
  - Now the significant majority of all Windows 10 devices
- TFNO – the middlebox with all possible TFO failure modes
- Passive probing algorithm to minimize disruption to user experience
- Around 26% devices successfully used TFO and did not fallback
- A/B test result - No measurable increase in navigation failures
- Failures are correlated with geography
- Failures are correlated with specific networks

# TCP Fast Open – Fallback algorithm

- Limited passive probing
- Probing is limited to “Internet connected” networks and no presence of incompatible 3<sup>rd</sup> party antivirus callout drivers
- TFO probing needs multiple connections to same server
- Allow only one probe TFO connection to proceed at a time (semaphore)
- When the probe connection is closed, mark success if all of the following hold
  - no RST in response to SYN
  - no SYN timeout
  - no full connection timeout (data packet retransmitted N times to failure)
  - data is exchanged in both directions
  - connection wasn't cancelled
  - no sudden RTT increase
- If a successful probe connection succeeded *exercising* cookie, success
- If a network hits fallback, persist & never attempt again
- If a network hits success, stop probing & remember for boot session

# Shortcomings of the Fallback algorithm

- SYN timeout heuristic makes fallback very aggressive
- Too conservative – doesn't persist success
- Assumes problems are close to the client and not the server
- Possibly long delays to determine success or fallback due to requiring multiple probe connections to close (reach CLOSED state)
  - long lived HTTP/2 probe connection
  - long timeouts
- Worst case middlebox behavior can still cause connectivity issues
  - i.e. dropping all packets from client IP address upon SYN with cookie request

# Improvements in Spring Creators Update

- Persist both success and fallback per network
  - Start off every device to probe again on upgrade
- Fallback only if SYN with TFO option fails and subsequent SYN succeeds
  - Loss of connectivity or server side firewall / downtime will not trigger fallback
  - Probing is no longer restricted to “Internet connected” networks
- Unknown TCP options reflected back!
  - Added this as a fallback condition

# Some more data

- In overall population
  - 45% devices complete probing
  - Of these devices
    - 46% succeed, 54% fallback
    - We expect improvements here in future - the aggressive SYN timeout logic is going away
- Particularly poor success rates in some geographies
  - In China, only 3% of probe completed devices succeed, rest fallback
  - In India, only 18% of probe completed devices succeed, rest fallback
- Repeated A/B experiment of TFO on/off on larger retail population - no statistically significant impact on page navigation failures

# Looking ahead

- Explore active probing to solve
  - Long delays
  - Minimize impact to user experience
  - Simplify the algorithm
- Augment with some form of happy eyeballs like approach to solve problems closer to the server
  - Unclear if this is really needed based on data so far
- Request other browsers to enable TFO on Windows - no need for any complex fallback logic in the browser
- Should RFC 7413 be Standards Track? Should it document possible fallback algorithms?

# Q&A