QUIC vs PEP
over Satellite Links

IETF 105 PANRG
July 25, 2019

John Border (John.Border@Hughes.com)
Bhavit Shah, CJ Su, Rob Torres
Hughes Network Systems

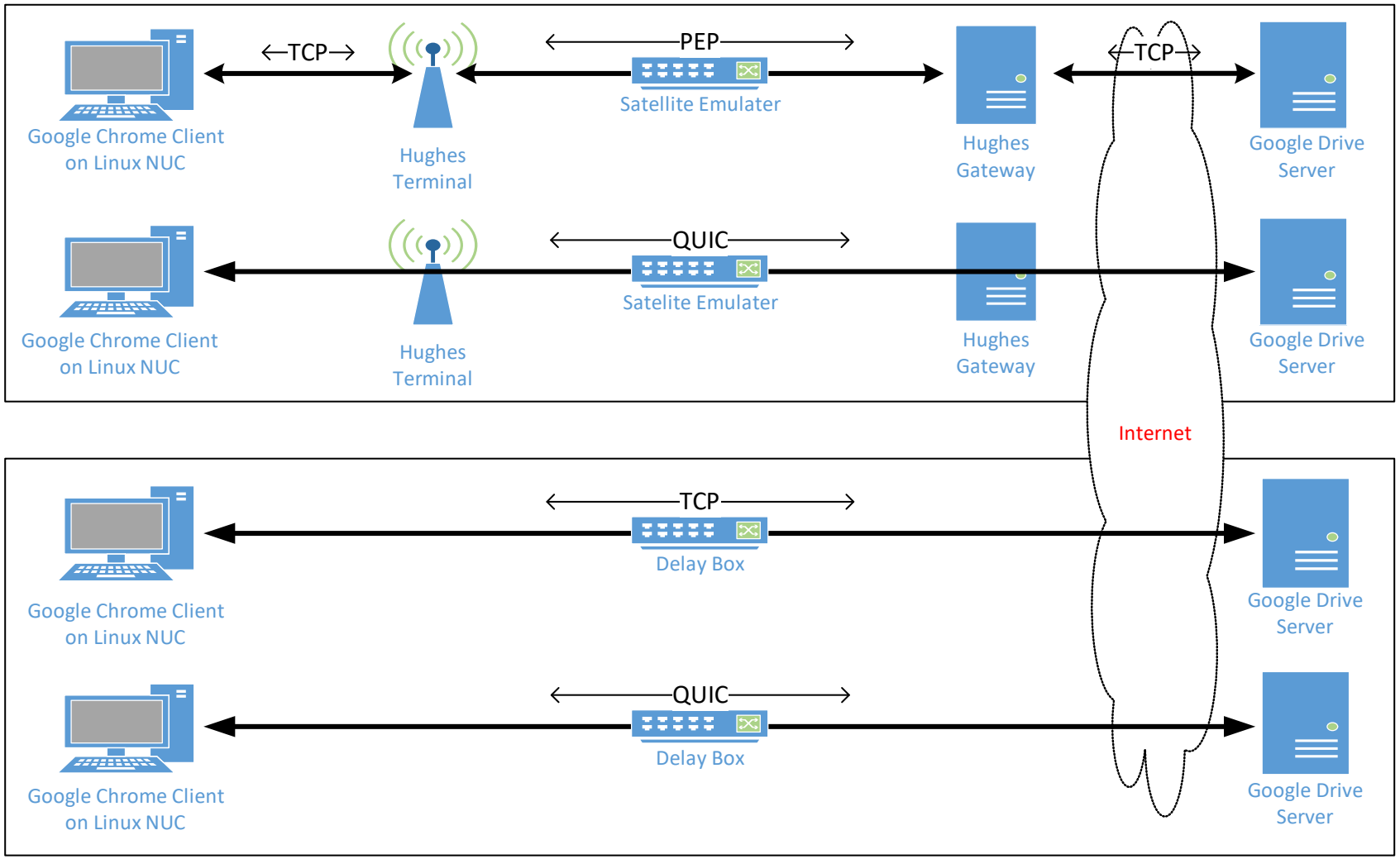
Problem Statement

- Transport layer protocols, in particular, TCP, do not perform well enough over high delay bandwidth product links like GEO satellites links without any modifications
  - Long latency impacts error recovery not just window sizes
- A split-TCP PEP is typically implemented to improve network performance over such links
- QUIC is a new transport layer protocol, originally developed by Google and now being standardized by the IETF
- QUIC can’t be split in the same way as TCP and hence suffers from relatively poor performance (when compared to PEP-ed TCP) on high latency links
- Testing being done to see how big the performance disparity is
  - Illustrates the need for path awareness

Test Setup

- Google Drive Server
- Google Chrome Client (v75.0.3770)
  - This is Google QUIC not IETF QUIC!
- Puppeteer library used to automate testing
  - Node.js script browses to Google Drive and downloads the specified file
  - Watcher setup for changes to the download directory by using fs.watch API
    - Listens for eventType= change for the file being downloaded and keeps checking that file's size
    - Watcher initiated when the file is clicked for download and closed when downloaded file's size is equal to the expected file size
  - Start and end time captured by the script when the watcher starts and closes
- Multiple (typically 100) runs for each test

Setup


TCP
PEP
TCP
Google Chrome Client on Linux NUC
Hughes Terminal
Satellite Emulater
Hughes Gateway
Google Drive Server
QUIC
Satelite Emulater
Google Chrome Client on Linux NUC
Hughes Terminal
Hughes Gateway
Google Drive Server
Internet
TCP
Delay Box
Google Chrome Client on Linux NUC
Google Drive Server
QUIC
Delay Box
Google Chrome Client on Linux NUC
Google Drive Server

Testing Variants

- Protocols
  - HTTP/1.1 over TCP (--disable-http2 flag)
  - HTTP/2 over TCP
  - HTTP/2 over QUIC (--enable-quic flag)
- Testbed
  - 1 Gbps connection to the Internet
  - Delay box simulating satellite delay
  - Two variants
    - Going through Hughes' terminal and gateway
      - Spoofed TCP and QUIC
    - Bypassing Hughes equipment
      - Unspoofed TCP and QUIC
- Files Sizes – 0.5 GB, 1.0 GB and 1.5 GB
- Packet Loss Rates – 0%, 0.1%, 1% and 10%
  - At the Delay Box

Testing Status

- We originally had hoped to complete all of the testing and have produced a white paper summarizing the results prior to IETF 105 but we did not make it
  - Error free testing is essentially done but we are just starting testing with controlled packet loss
- We did get far enough along to show some preliminary results which already highlight the need for QUIC path awareness for satellite

Results Sample – 1 GB File

- Running through the Hughes terminal and gateway – No Packet Loss
  - TCP HTTP 1.1 with PEP ~217 Mbps
  - TCP HTTP 2.0 with PEP ~43 Mbps
  - QUIC HTTP 2.0 ~36 Mbps

- Running direct – No Packet Loss
  - TCP HTTP 1.1 ~33.2 Mbps
  - TCP HTTP 2.0 ~30.4 Mbps
  - **QUIC HTTP 2.0** **~33.8 Mbps**

- Running direct – 1% Packet Loss
  - TCP HTTP 1.1 ~20.6 Mbps
  - TCP HTTP 2.0 ~17.8 Mbps
  - **QUIC HTTP 2.0** **~17.7 Mbps**

Path Awareness

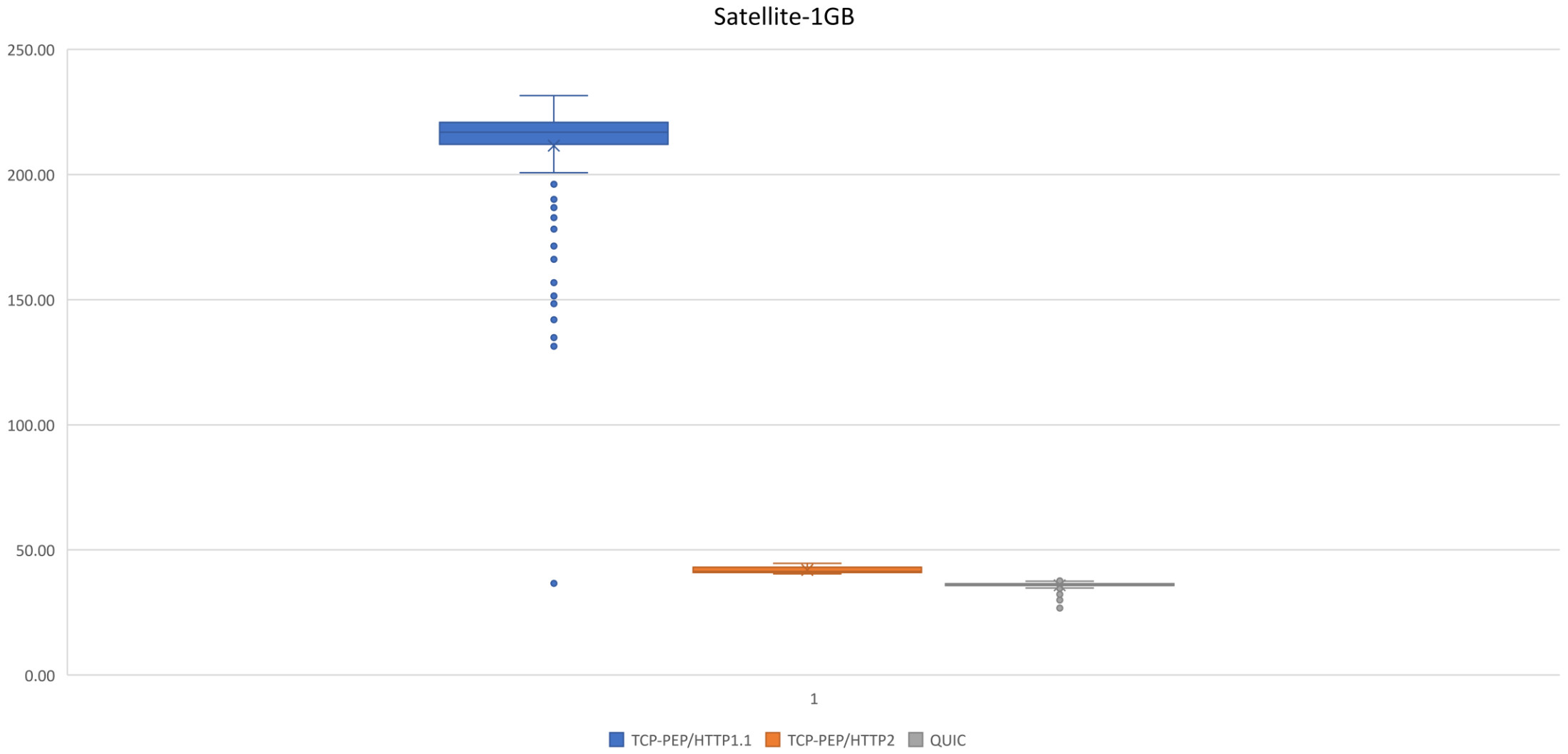
- In order for QUIC over satellite to match PEP performance, QUIC needs:
  - A very large window
  - Some sort of optimized packet loss recovery
    - For example FEC (as described in draft-swett-nwcrg-coding-for-quic)
- The above will not make good default values for general (non-satellite) QUIC use cases so some awareness of when these things are needed is required
- LOOPS-like solutions may help but it is difficult to cover the entire end to end path with a solution embedded in the transport

Collaboration and Future Work

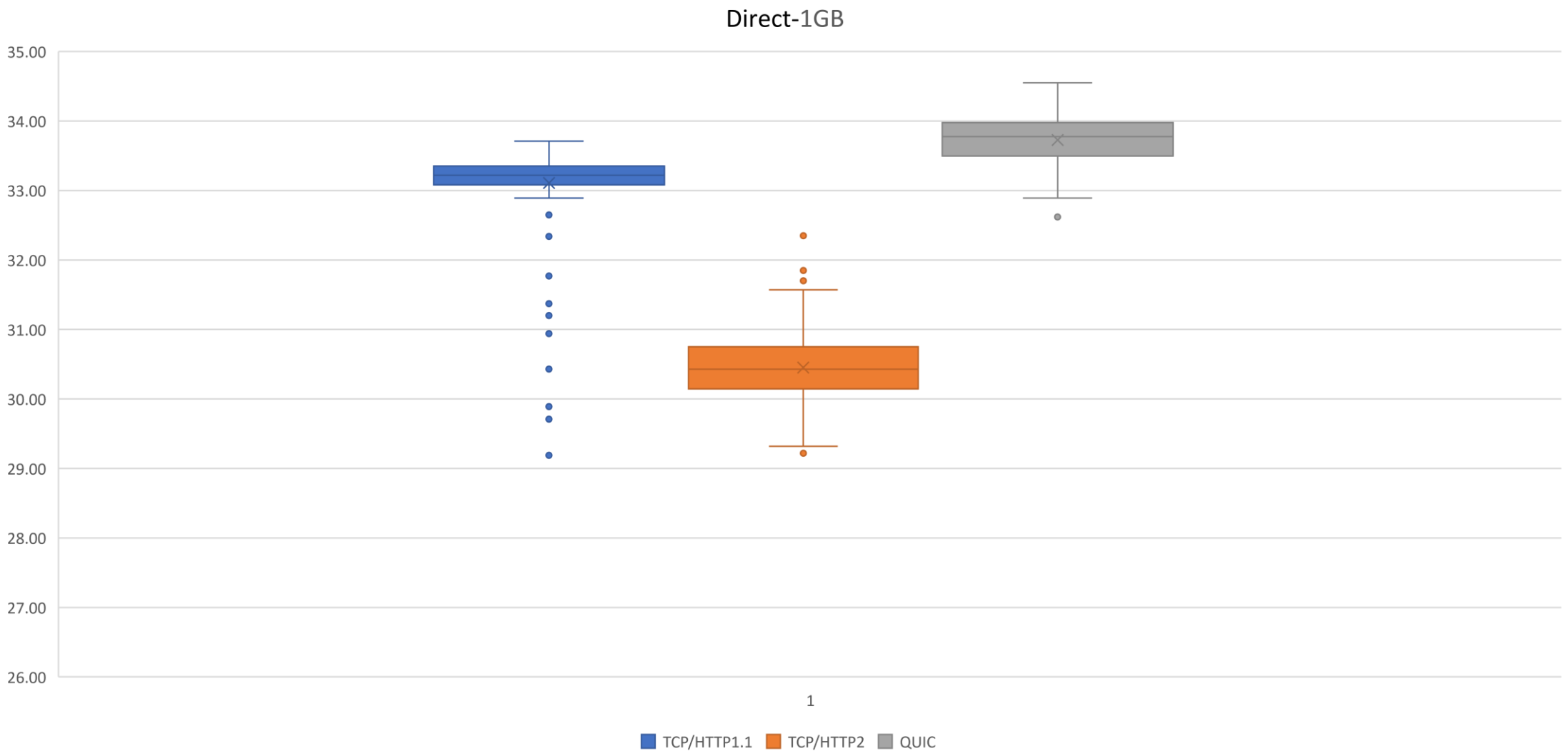
- We are already trying to coordinate with other people doing similar testing
- We in particular are interested in working with someone who has control of a QUIC-capable server reachable via the Internet which supports very high speed access
  - We would like to have some control over when BBR is and is not used

Backup Slides

1.0 GB File Results Sample with Hughes PEP



1.0 GB File Results Sample Direct



1.0 GB File Results Sample Direct – 1% Packet Loss