

2.17 report

Zhewen Yang

Testbed setup

Sender and **Receiver** are running on the same machine.

QUIC establishes **three transmission paths** through **three different ports**:

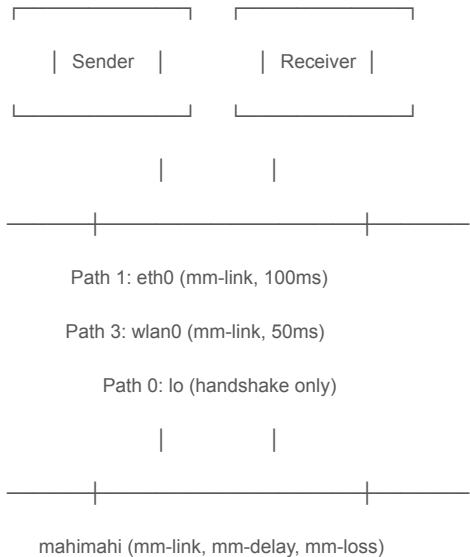
- **Path 1:** `eth0` (simulating a wired network)
- **Path 2:** `wlan0` (simulating WiFi)
- **Path 3:** `lo` (only used for QUIC handshake)

Network Emulation with `mahimahi` (<http://mahimahi.mit.edu/>)

- `mahimahi` is used to introduce network constraints:
 - **Bandwidth Limitation:** `mm-link` restricts different paths' bandwidth.
 - **Latency Emulation:** `mm-delay` adds artificial delay.
 - **Packet Loss Control:** `mm-loss` simulates packet loss.

(But for now I do not have complex network trace, just set a fixed delay)

Testbed setup



EXP

Video: Big_bunny

Total Duration: 20.066s

Resolution: 1920x1080 (Full HD)

Frame Rate: 30 fps

Bitrate: 3724023 bps

Audio Codec: AC-3 (48kHz)

Segment Duration: 3.246s

Total Segments: 6

Initial EXP

- Total video length: **20.066s**, segment duration: **2.87s**.
- Max buffer size: **1s**, initial buffer: **1s**.
- Buffer management: **Alpha = 1s, Beta = 2s**.
- **Processing first segment:** `segment_3724023_.mp4` (init segment)
 - a. Bitrate: **3724023 bps**
- **Download Metrics:**
 - a. Segment size: **1742 bytes**
 - b. Download time: **9.2ms**
- **Final segment downloaded:** `segment_3724023_6.m4s`
 - a. Size: **2071529 bytes**
 - b. Download time: **24.95ms**

Total data downloaded: 8006383 bytes

Playback completed at: `20.066s`

Final buffer state: `0s`, indicating smooth playback with no interruptions.

Initial EXP

Multi-Path QUIC Transmission

- **Path 0:** Loopback (1o), RTT = $560\mu\text{s}+t$
- **Path 1:** IP 10.188.59.107:36563, RTT = $297\mu\text{s}+t_1$
- **Path 3:** IP 10.110.152.185:45226, RTT = $153\mu\text{s}+t_3$ (shorter) (t is the delay set by mahimahi)

Path	Packets Sent	Retransmissions	Packets Lost	Packets Received	RTT
0	6	0	0	7	$560\mu\text{s}+t$
1	1414	2	0	2547	$297\mu\text{s}+t$
3	2603	3	0	4874	$153\mu\text{s}+t$

Still have retransmission but with no pkt lost. RTT might be too low, or CWND is too small. I can change the trace to other network environment.

Compare w sp-quic (same video)

	SP	MP
path	1	2
Handshake Time	Faster (no path management overhead)	Slower (multiple paths need synchronization)
RTT	$87\mu s + t$	Path 0: $297\mu s + t$, Path 3: $153\mu s + t$
Retransmission	2	5
Segment Download Time		-30%-10%

SP-QUIC has **lower RTT and fewer retransmissions**, making it more efficient for stable networks.

MP-QUIC leverages **multiple paths** but suffers from **ACK delays and occasional spurious retransmissions**.

The improvement is not 50%(of course), but need to identify the reason. (Many paper have conclusions, but I need to find out in the exp logs.)

For stable, low-latency networks → SP-QUIC is preferable due to lower overhead and better efficiency.

For lossy or fluctuating networks → MP-QUIC can provide better robustness by utilizing multiple paths.

Conclusion

- **No network bottleneck observed.**
- **QUIC multi-path scheduling is effective.**
- **Playback starts with minimal delay and runs smoothly.**
- **Further optimizations can improve reliability and video quality.**

Next to do :

I am modifying mahimahi to cooperate with our quic testbed to make it work with more complex network trace. Try to run more experiments to see results.

And try to stream 3d/spatial video.

Now still use minRTT scheduler. I am working on implementing other scheduler (like schedule the packet according to frame)