Updates on QUIC Over In-sequence Paths with Different Characteristics

Nicolas KUHN

nicolas.kuhn@cnes.fr

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

VIVERIS
Innover. Simplifier. Partager.

In-sequence paths with different characteristics

Satellite systems:

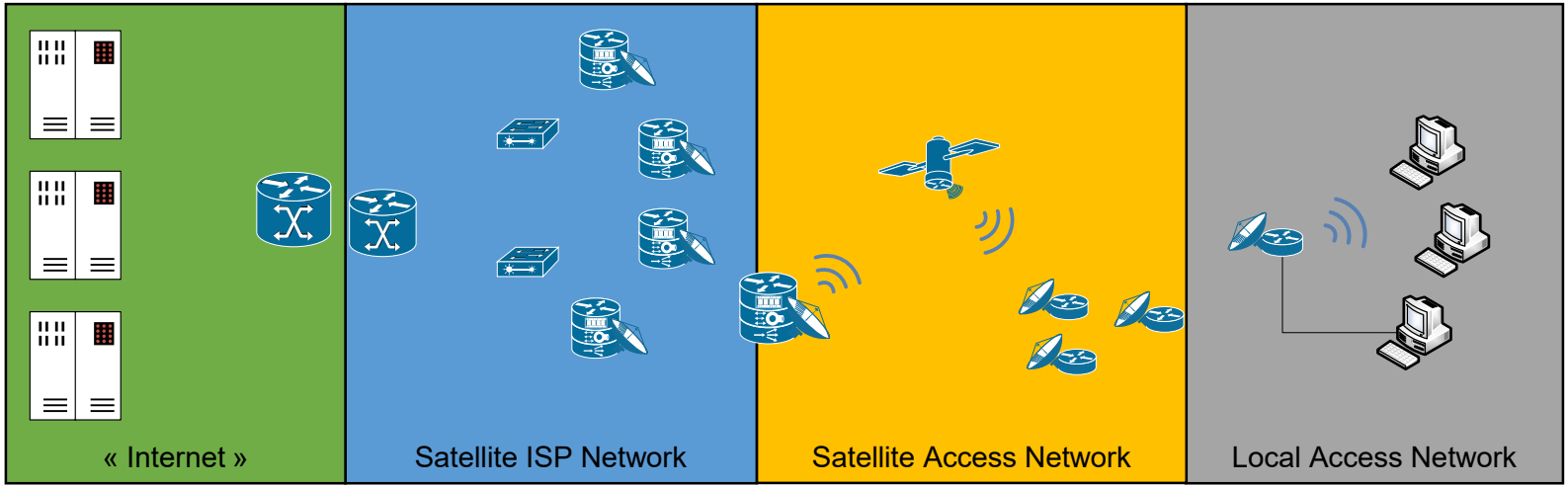
- Point-to-point links or TV broadcast
- Use as an access technology for remote locations
- Backup and rapid deployment of new services
- Transit networks
- Backhaul of various types of IP networks

➢ **Satellite: IP network segment one part of the end-to-end path**

User traffic can experience a path that includes:

- Satellites capacity (long delay link)
- With a wide variety of other network technologies (Ethernet, cable modems, WiFi, cellular, radio links, etc)

Typical GEO satellite-based Internet access



Data rate	High	High	Variable	Average
Latency	Low	Low	High	Low
Loss	No loss	Congestion losses	No loss	Loss if Wi-Fi

Typical GEO satellite-based Internet access

- **Paths with different characteristics**

	Path #1	Path #2	Path #3	Path #4	**END-TO-END**
Data rate	High	High	Variable	Average	**Variable**
Latency	Low	Low	High	Low	**High**
Loss	No loss	Congestion losses	No loss	Loss if Wi-Fi	**Congestion losses and Wi-Fi losses**

- **Complex for end-to-end protocols when local break-out is not possible**
  - Solution #1 : adapt the end-to-end protocols
  - Solution #2 : inform end point of the path characteristics

Solution #1 : adapt the end-to-end protocols

Definition of scenarios in draft-kuhn-quic-4-sat-05

Case	Download path (Mbps)	Upload path (Mbps)	Traffic	Loss
Medium public satellite broadband access	50	10	Download and upload 100 MB	None
Medium public satellite broadband access	50	0.5	Download 100 MB	None
Medium public satellite broadband access	50 -> 10 (after 5s)	10	Download 100 MB	None
Loss-free large public satellite broadband access	250	3	Download 100 MB – wait 10s – repeat Download 100 MB	None
Lossy large public satellite broadband access	250	3	Download 100 MB	Uniform (1%)

- **Question on the performance comparison between**
  - Multiple local paths
  - In-sequence paths

Solution #1 : adapt the end-to-end protocols

Exploited plate-forme for tests

QUIC SERVER

picoquic/picoquic (h3-25/24/23)
or
h2o/quicly(h3-25)

Delay / Bandwidth limitation
Losses

QUIC CLIENT

picoquic/picoquic (h3-25/24/23)
or
curl/ngtcp2 (h3-25)

End-to-end path without split

Multiple paths with split

IPERF3 SERVER

Default
Kernel 4.15
Ubuntu 16.04
iperf3 v3.6

PEPSal

Delay / Bandwidth limitation

PEPSal

Losses

IPERF3 CLIENT

Default
Kernel 4.15
Ubuntu 16.04
iperf3 v3.6

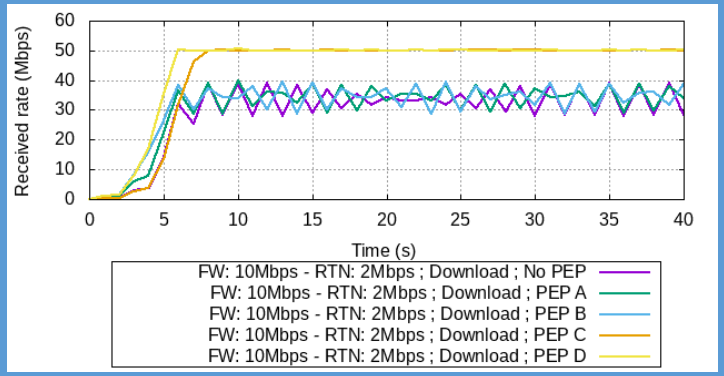
Default ; Kernel 4.4
Same configuration on both PEP client and PEP server

	TCP_WMEM_MAX (MB)	TCP_RMEM_MAX (MB)	CORE_WMEM_MAX CORE_RMEM_MAX (MB)	ICWND IRWND (packets)
NO PEP	4	6	0,2	10
PEP A	4	6	0,2	10
PEP B	4	6	0,2	100
PEP C	33	33	33	10
PEP D	33	33	33	100

Solution #1 : adapt the end-to-end protocols

Focus on the 50 Mbps / 10 Mbps use-case

Multiple paths with split



With TCP-Proxy:

- Capacity to reach channel capacity
- Reduced transmission time

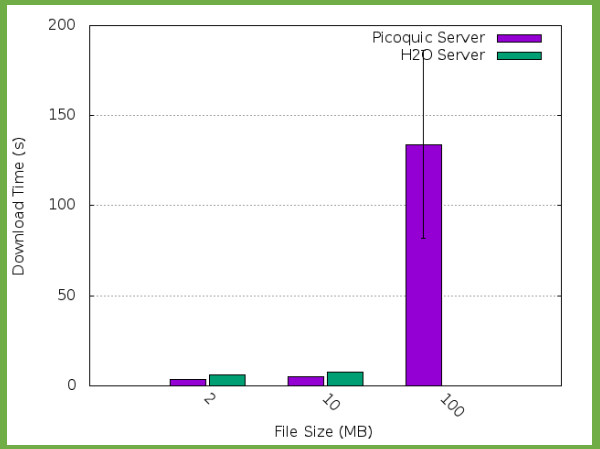
Proposed objectives :

- 2MB: 3 sec
- 10 MB: 5 sec
- 100MB: 20 sec

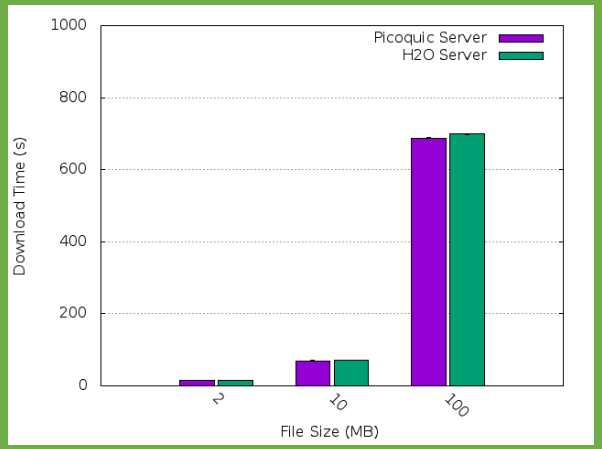
Solution #1 : adapt the end-to-end protocols

Focus on the 50 Mbps / 10 Mbps use-case

End-to-end path without split



PICO-QUIC CLIENT



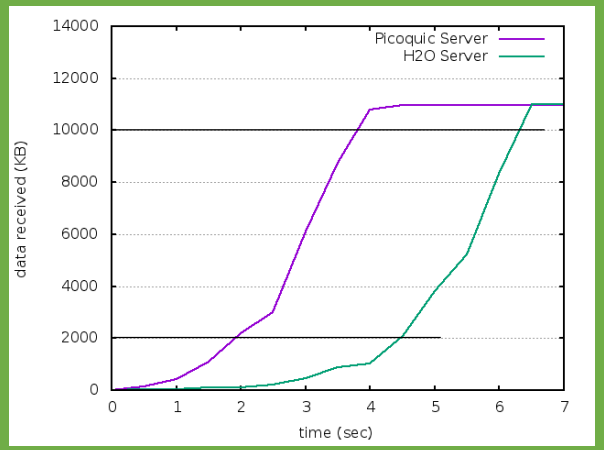
CURL CLIENT

- Issue in the case H20 server and PICO-QUIC client and 100 MB
  - At PICO-QUIC CLIENT: “[picoquic_retransmit_needed]: Too many retransmits of packet number 6350, disconnect”
  - Managed by the MAX_ACK_DELAY and ACK_DELAY_EXPONENT parameters by PICO-QUIC SRV
- PICO-QUIC SRV show better performance
  - congestion control is BBR but trend confirmed with RENO
  - Impact of other parameters (e.g. INITIAL_CWND of INITIAL_RTT) ?

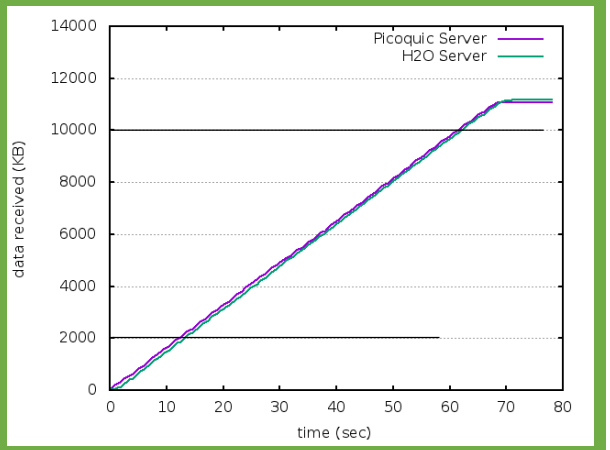
Solution #1 : adapt the end-to-end protocols

Focus on the 50 Mbps / 10 Mbps use-case

End-to-end path without split



PICO-QUIC CLIENT



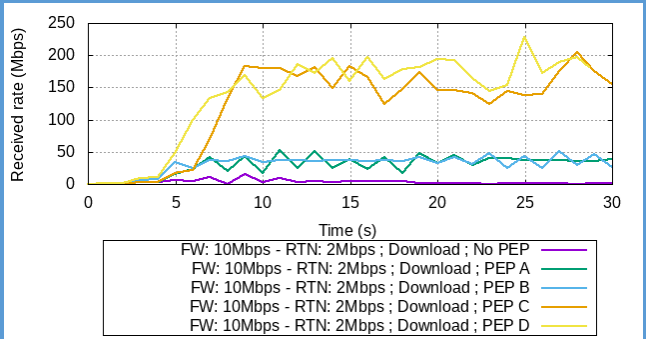
CURL CLIENT

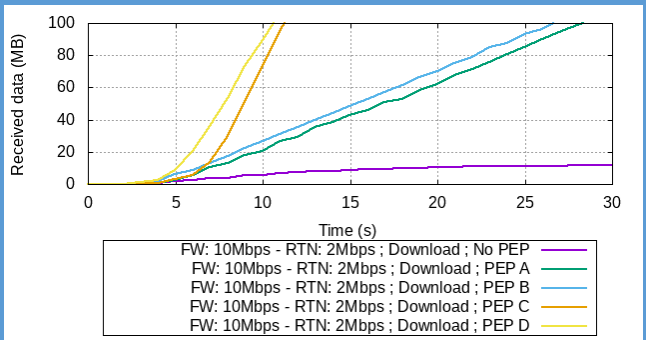
- PICO-QUIC client
  - PICO-QUIC server : the objectives are met
  - H20 server : the objectives are not met
- CURL client (any server)
  - The objectives are not met

Solution #1 : adapt the end-to-end protocols

Focus on the 250 Mbps / 3 Mbps use-case / 1% random loss

Multiple paths with split





With TCP-Proxy:

- Capacity to reach channel capacity
- Reduced transmission time
- Local recovery

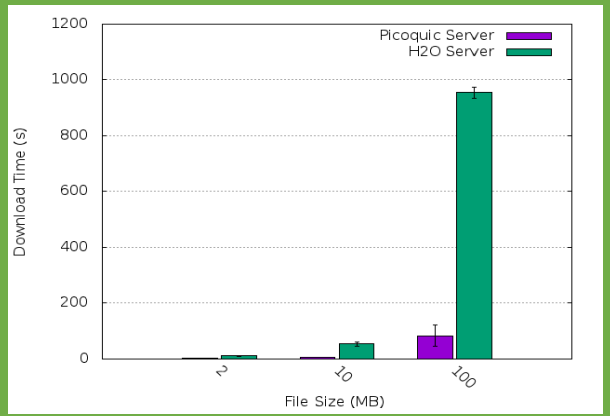
Proposed objectives :

- 2MB: 3 sec
- 10 MB: 6 sec
- 100MB: 10 sec

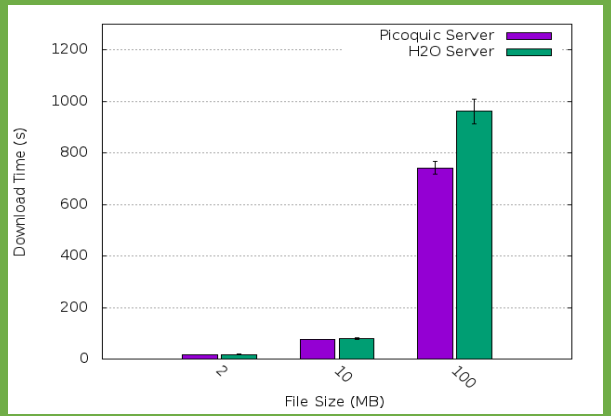
Solution #1 : adapt the end-to-end protocols

Focus on the 250 Mbps / 3 Mbps use-case / 1% random loss

End-to-end path without split



PICO-QUIC CLIENT



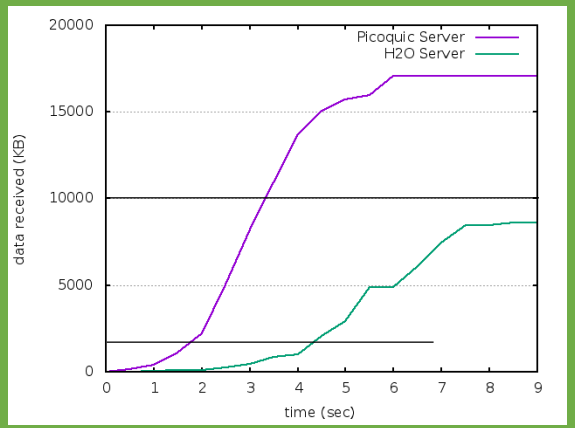
CURL CLIENT

- Trends of the 50 Mbps / 10 Mbps use case are exaggerated
- PICO-QUIC at both client and servers exhibit better performance
  - But still does not reach the available bottleneck – limited to 50 Mbps (flow control limits are reached)
- The difference between PICO-QUIC client and CURL client are less important than in other use case

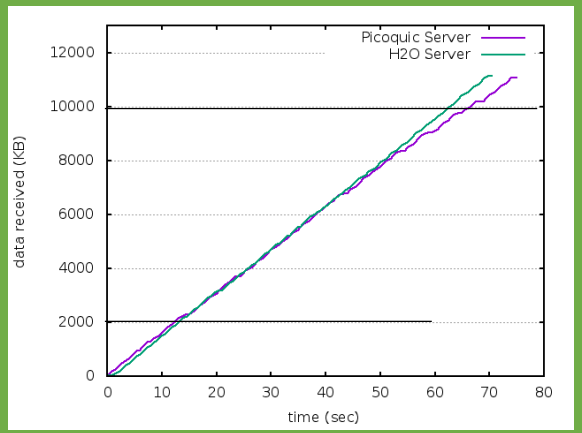
Solution #1 : adapt the end-to-end protocols

Focus on the 250 Mbps / 3 Mbps use-case / 1% random loss

End-to-end path without split



PICO-QUIC CLIENT



CURL CLIENT

- PICO-QUIC client and server is the only combination that meets the objectives

Solution #2 : inform end point of the path characteristics

- **Designing a CC that is relevant for all deployment cases may not be relevant**
- **Knowing about the path characteristics can help in adapting the CC in specific deployment scenarios**
  - Tuning RTT_INIT
  - Tuning flow control parameters (MAX_STREAM_DATA)
- **See draft-kuhn-quic-0rtt-bdp-06 for how to do it in QUIC**
  - There is also a strawman algorithm in the draft on how to safely jump to the available capacity

Why PICOQUIC meets objectives ?

- Different default transport parameters

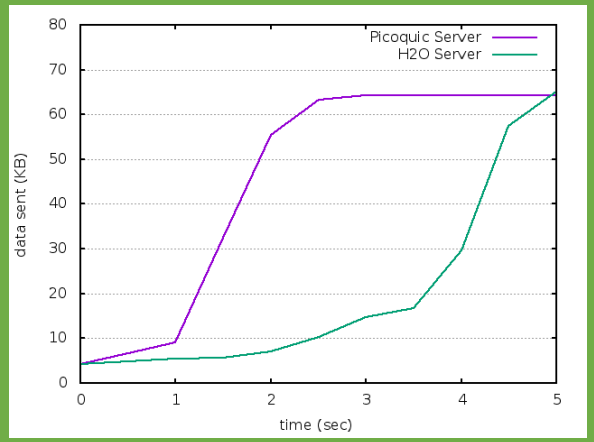
	picoquic client/server	h2o server	Curl client
MAX_PACKET_SIZE	1440B	1280B	1280B
INITIAL_CWND	10 * 1440	10 * 1280	10 * 1280
INITIAL_RTT	250ms	100ms	
ACK_RATIO	2 :1	2 :1	2:1
ACK_MAX_DELAY	10ms	25ms	25ms
ACK_DELAY_EXPONENT	3	10	3
Congestion control	BBR	Reno	N.A.

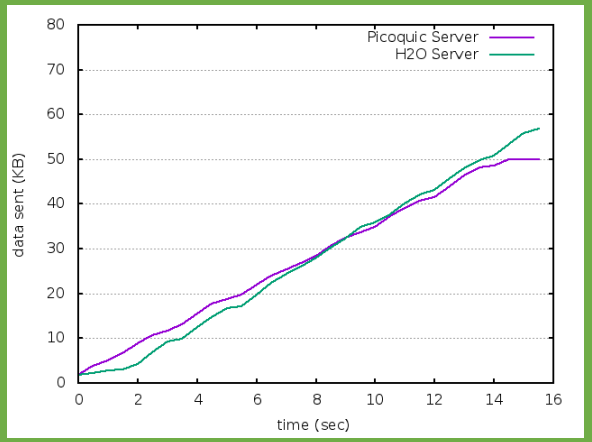
- On going investigations to assess what parameters are game changers

Why PICOQUIC meets objectives ?

- **ACK strategy (50 Mbps / 10 Mbps use case)**

End-to-end path without split





- **PICOQUIC implements ACK coalescing**
  - Starts with ACK ratio 2:1 but quickly increases it to 10:1

Next steps

- **Further work on game-changer parameters for the satellite use-case and implement 0-RTT draft**
- **PICOQUIC implements non standard parameters or algorithms that are very relevant for SATCOM use-case**
  - Are they relevant for other cases ? (e.g. ACK management)
  - PICOQUIC can still do better by increasing flow control limits for high BDP use-cases
- **Integrate other QUIC implementations**
- **Release the code that has been used**