Post-Quantum

Cryptography Conference

The impact of ML-KEM and ML-DSA on mTLS connection Time-to-Last-Byte

Multiple studies have evaluated the impact of PQC algorithms in TLS 1.3. These studies have been focusing on server authentication with PQC signatures. To our knowledge, there has been no study focusing on mTLS authentication where the client sends a PQ certificate chain as well. Such connections could be used in Zero Trust Architectures where the client opens multiple connections to various destinations each of which uses mTLS authentication. These sessions will be double impacted by the size of the "authentication data" travelling both directions. This presentation will share experimental results of the Time-to-Last-Byte (TTLB) of mTLS connections using ML-KEM and ML-DSA and transferring small and larger amounts of data. We will evaluate different round-trips, network bandwidth and TCP initial congestion windows. We will discuss the effect of PQC on mTLS sessions and compare it to previous experiments on typical TLS connections. We will cover potential mTLS use-cases that will suffer more than others and ways to improve them.

Mila Anastasova
Applied Scientist at Amazon Web Services (AWS)

Panos Kampanakis
Principal Security Engineer, Applied Scientist at Amazon Web Services (AWS)

SSL.com PQ SHIELD HID KEYFACTOR ENTRUST

January 15 and 16, 2025 - Austin, TX (US) | Online

PKI Consortium

The impact of ML-KEM and ML-DSA on mTLS connection Time-To-Last-Byte

Mila Anastasova & Panos Kampanakis
Amazon Web Services

Content

Introduction

Why PQ mTLSv1.3?

- Why PQ?
  - The challenges?
- Why mTLS?
- TLSv1.3 vs mTLSv1.3

PQ PKI Migration

Cost Estimation?

- TCP & mTLSv1.3
  - Communication Flow
- Round Trip Time (RTT)
- Congestion Window
  - TCP & TLS RTT impact
  - Application Data Impact

PQ PKI Performance

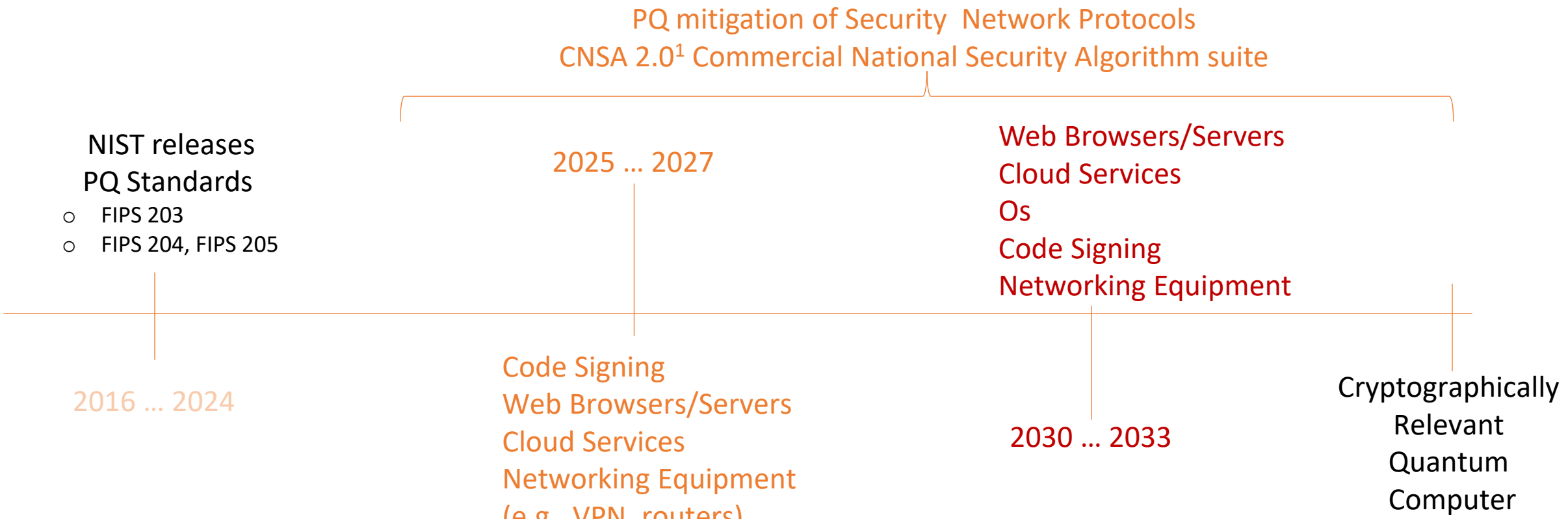
Reality?

- Performance Graphs
- Performance Analysis

PQ mTLSv1.3

→ Why PQ mTLSv1.3?

Why migrating to PQ ML-KEM and ML-DSA?



- Standards Preparation
- CNSA 2.0 as default and preferred
- CNSA 2.0 Exclusive Use

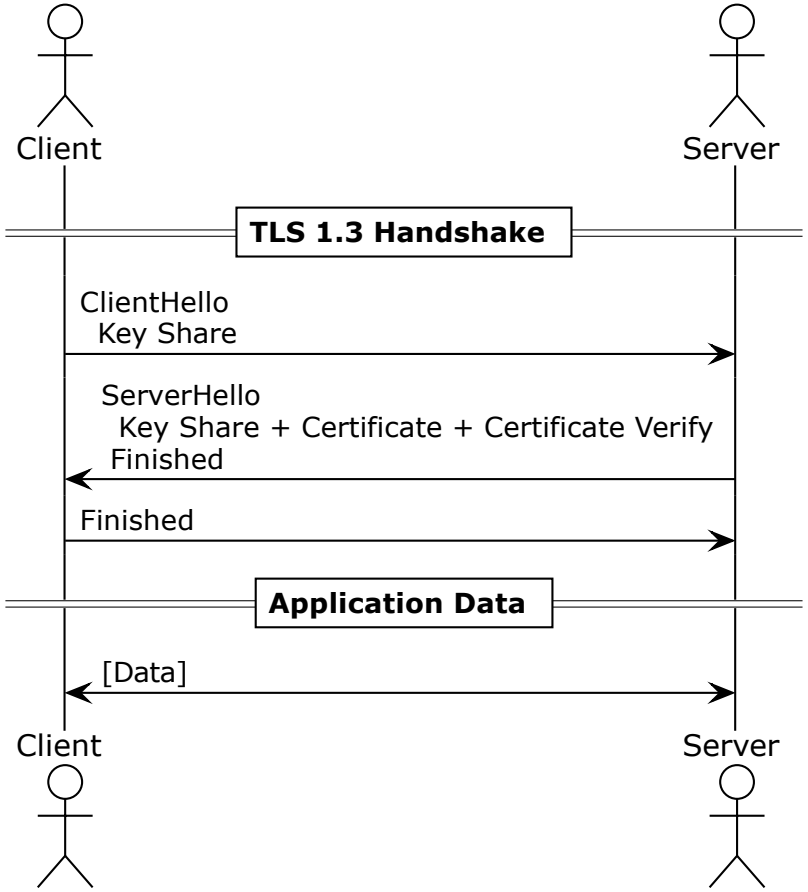
[1]https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF

Why mTLS?

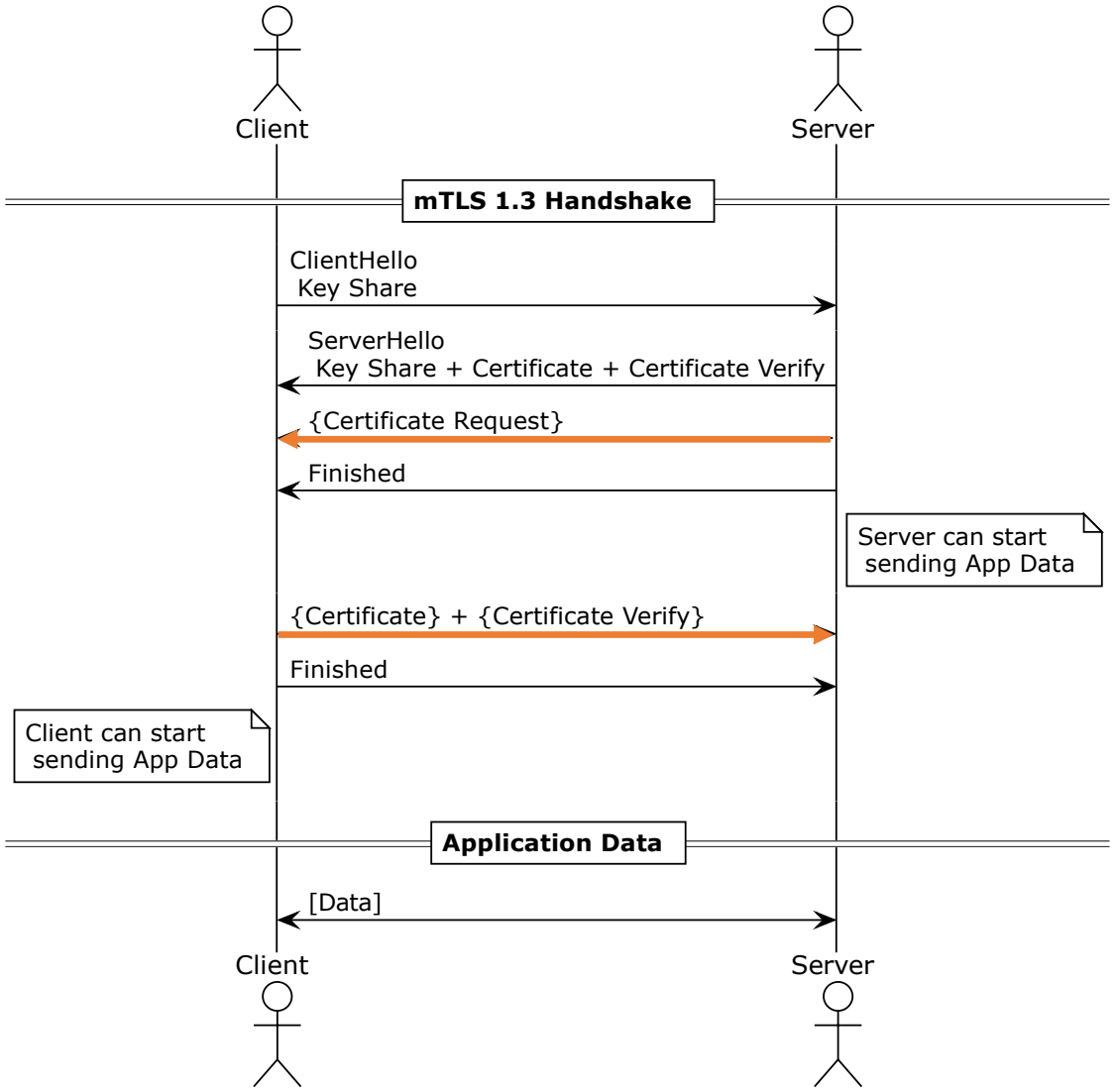
TLS	mTLS
• Secure Websites (HTTPS) • Public APIs • … • Application where only the server authentication is required	• Microservices Communication • Internal APIs • e-banking • Edge-to-Cloud Communication • Zero Trust Architecture • … • Any application with **mutual** trust of authentication needed
Many studies on the impact of ML-KEM and ML-DSA on TLS	**No studies on the impact of ML-KEM and ML-DSA on mTLS**

TLS 1.3 vs mTLS1.3?

TLSv1.3



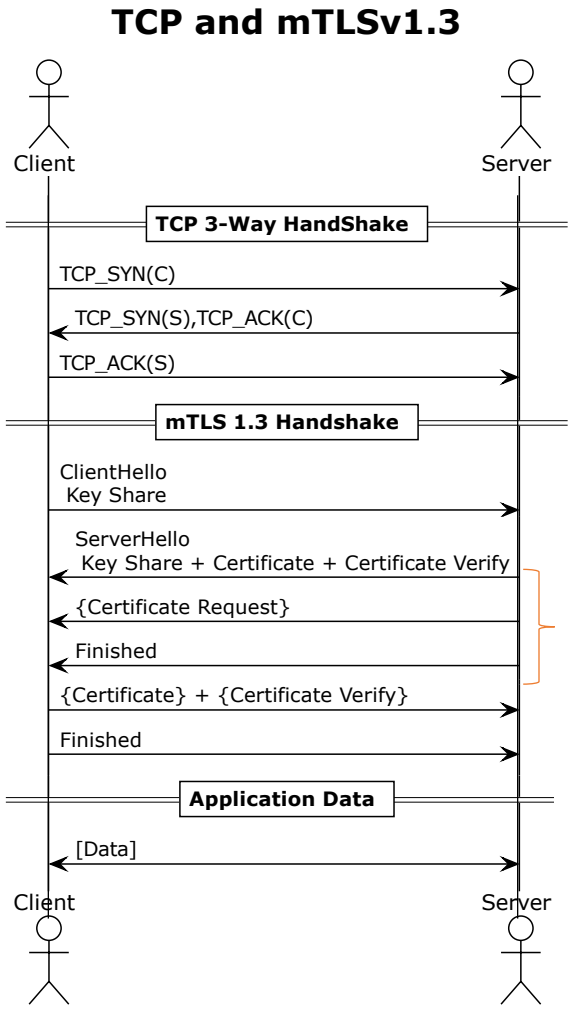
mTLSv1.3



PQ mTLSv1.3 Performance Estimation

→ How to estimate the PQ mTLSv1.3 latency overhead?

How is a secure network connection established?



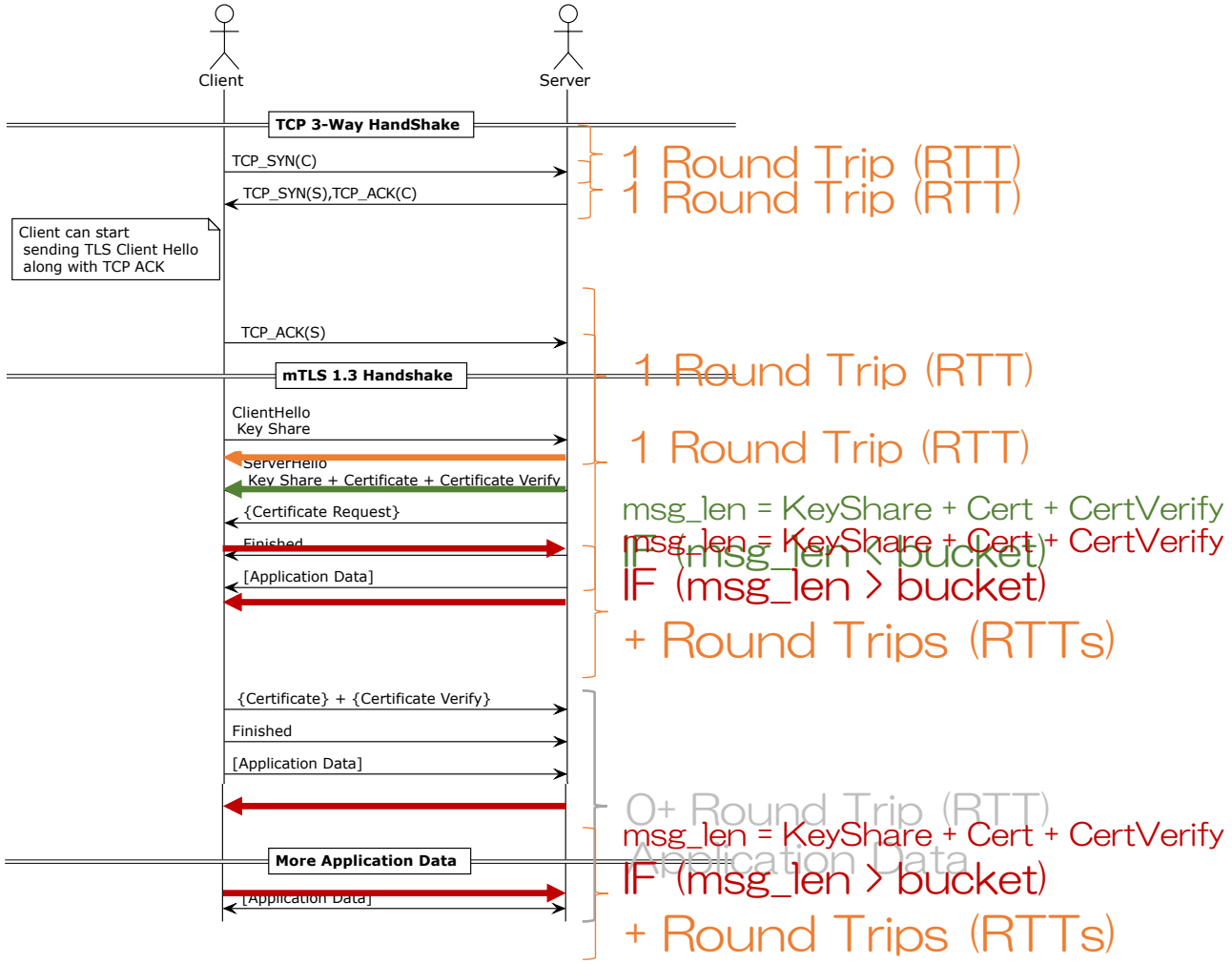
- Communication flow = mainly round trips
- Each round trip – 2-way communication flow
- Round trips are expensive (high latency)
  - Round Trip Time (RTT)

Multiple messages are merged up to a given size (‘bucket’), forming part of a single round trip

- How PQ PKI affects the communication latency?
  - Additional RTTs?

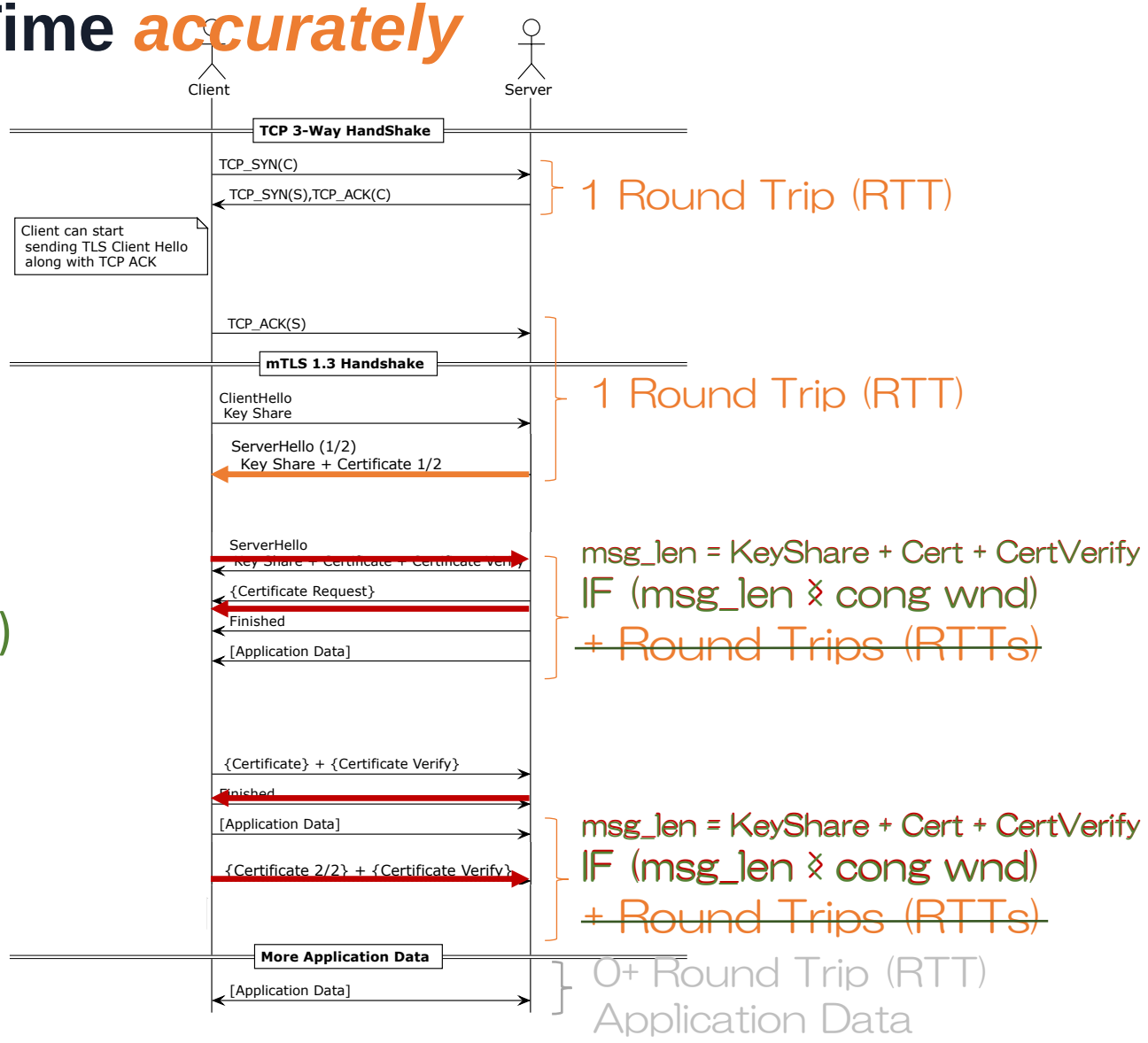
How is the *cost* of a secure network connection estimated?
Round Trip Time Estimate

- Classical Crypto primitives
  - (small keys and signatures)
  - Easily fit into a single ‘bucket’ size
    - Single Round Trip
- PQ crypto primitives
  - (large keys and signatures)
  - They don’t fit into a single ‘bucket’
    - Introduce Additional Round Trip



Congestion Window
Estimate the Round Trip Time *accurately*

- **Congestion WiNDow** (cwnd) (‘the bucket’)
  - The amount of data that fits into a Single Round Trip (~15KB)
- PQ Certificates (~22KB) > cwnd (~15KB)
  - Additional Round Trips
- Increase Congestion Window
  - PQ Certificates (~22KB) < cwnd (~34KB)
    - No Additional Round Trips
- Application Data Transfer would also benefit from large congestion window
  - Improved Time-To-Last-Byte

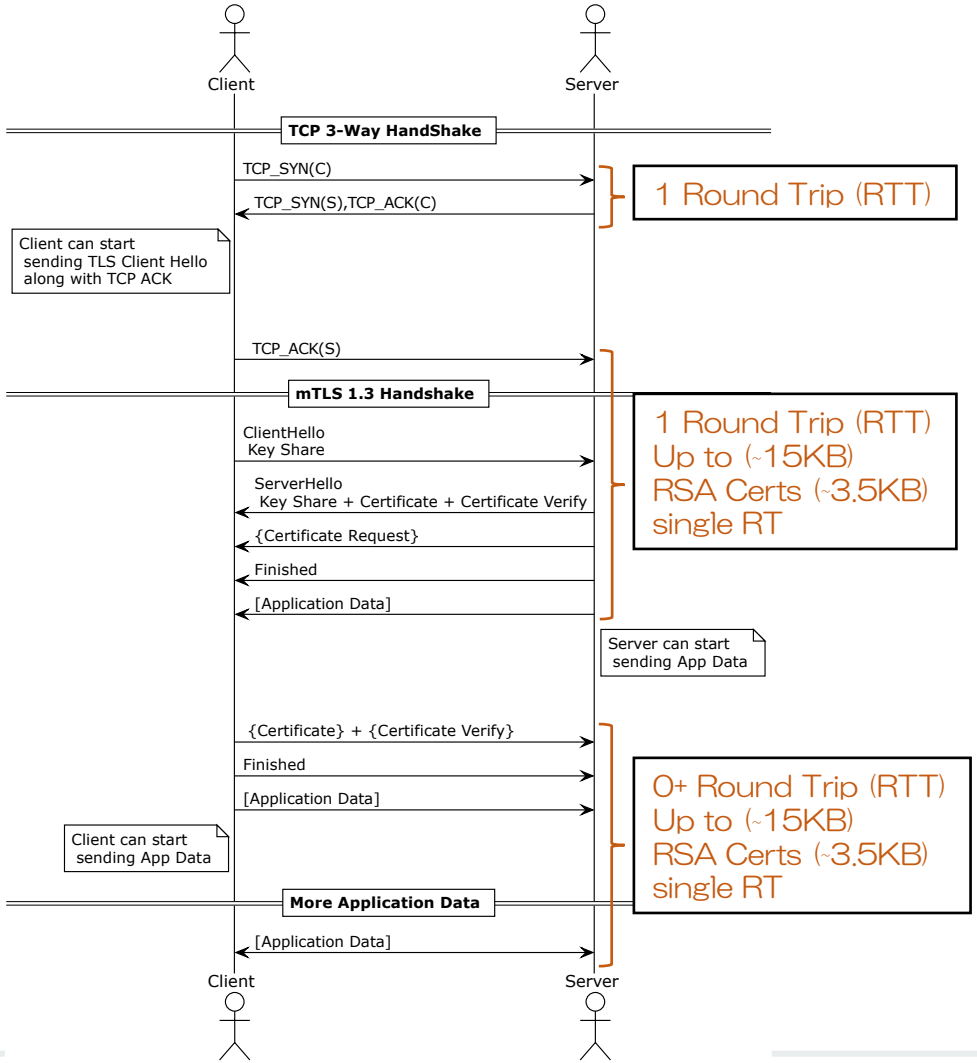


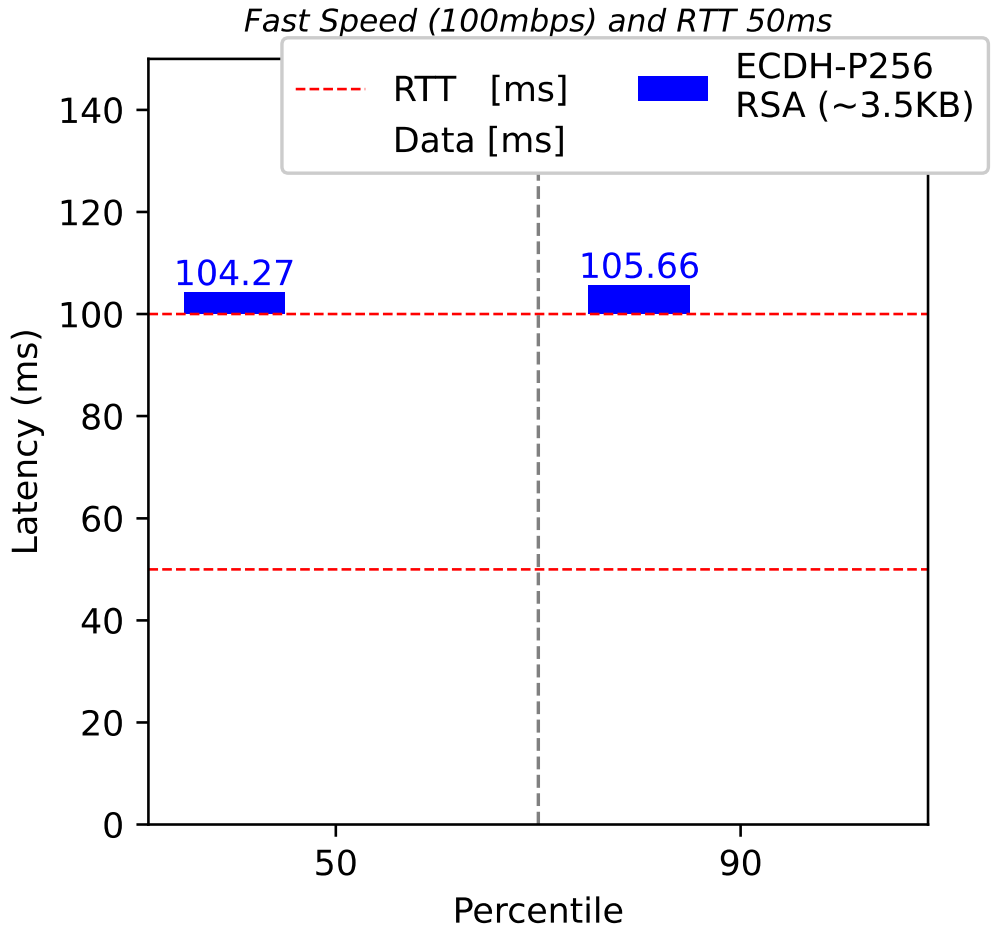
PQ PKI Performance

→ The actual results

mTLSv1.3 handshake - Classical Cryptography

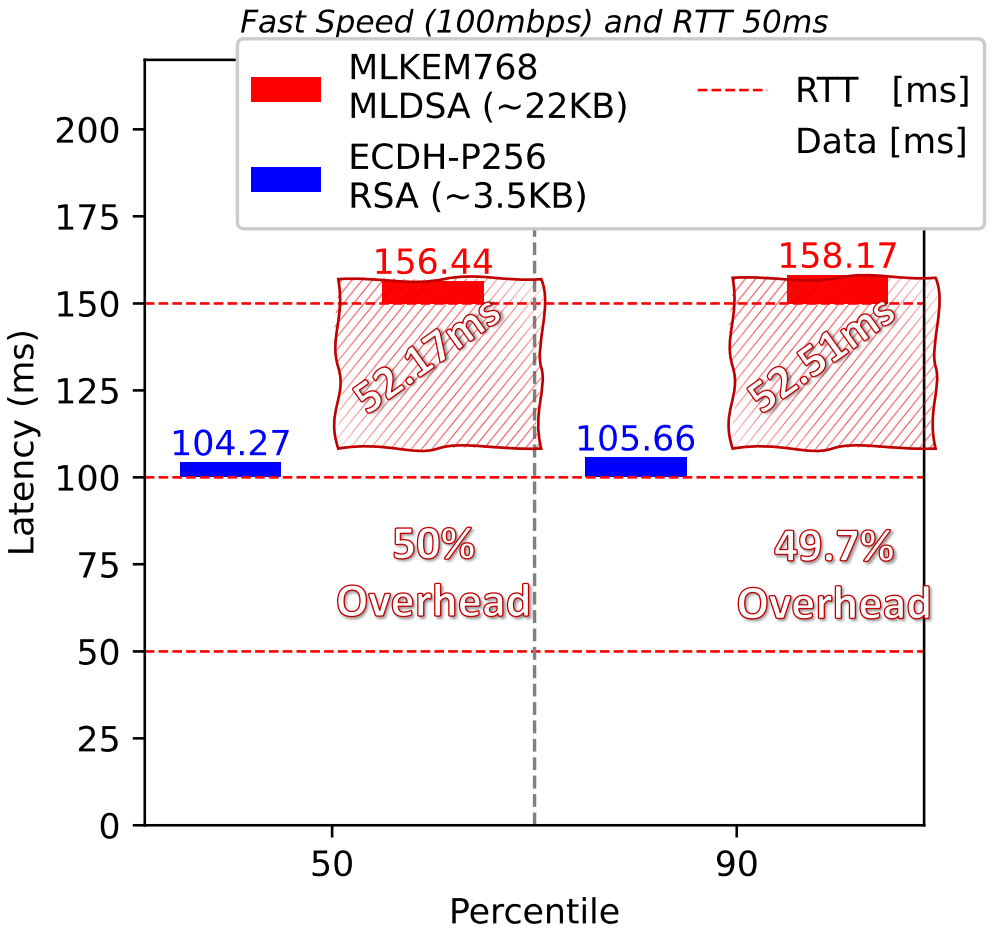
Classical (small) Certs with Small Congestion Window

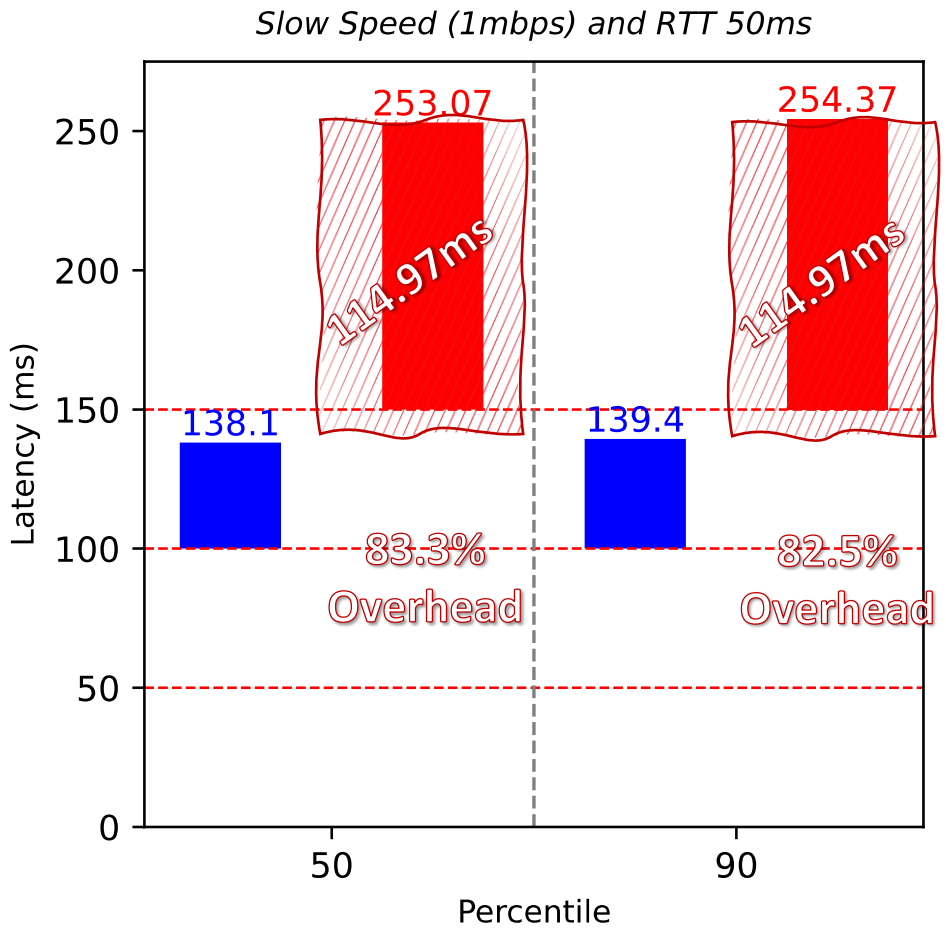




mTLSv1.3 handshake - Post Quantum Cryptography

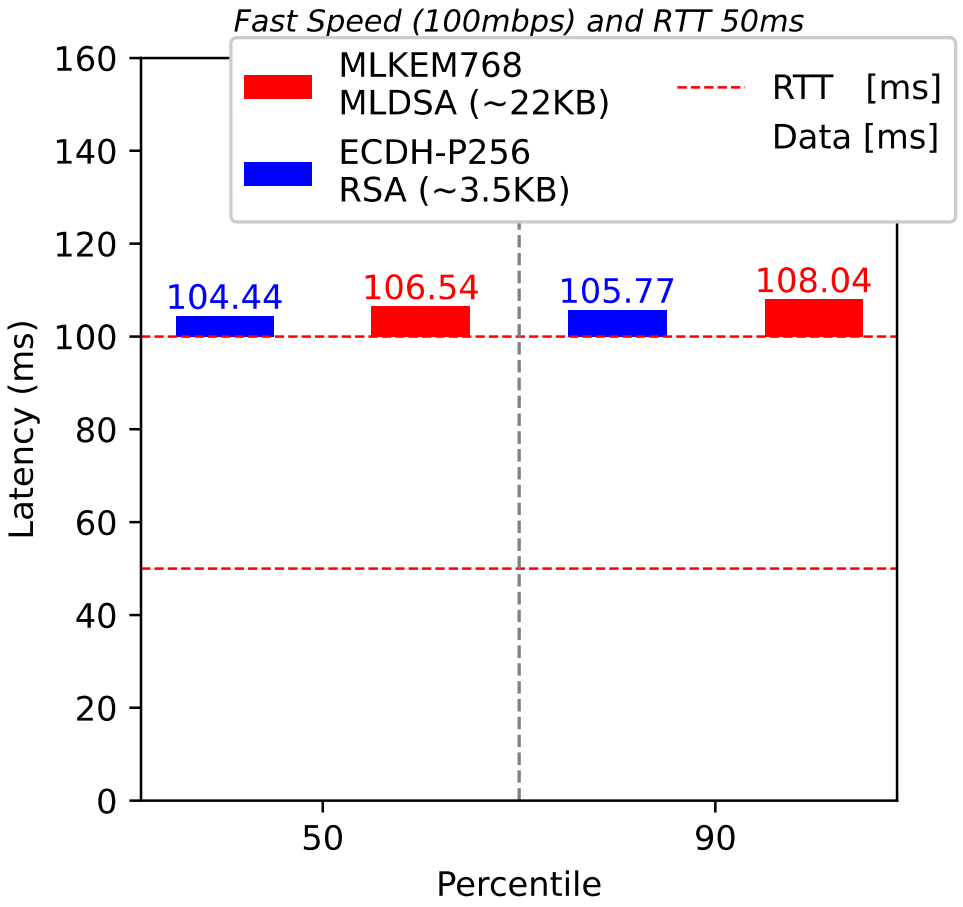
Classical vs. PQ (large) Certs with Small Congestion Window

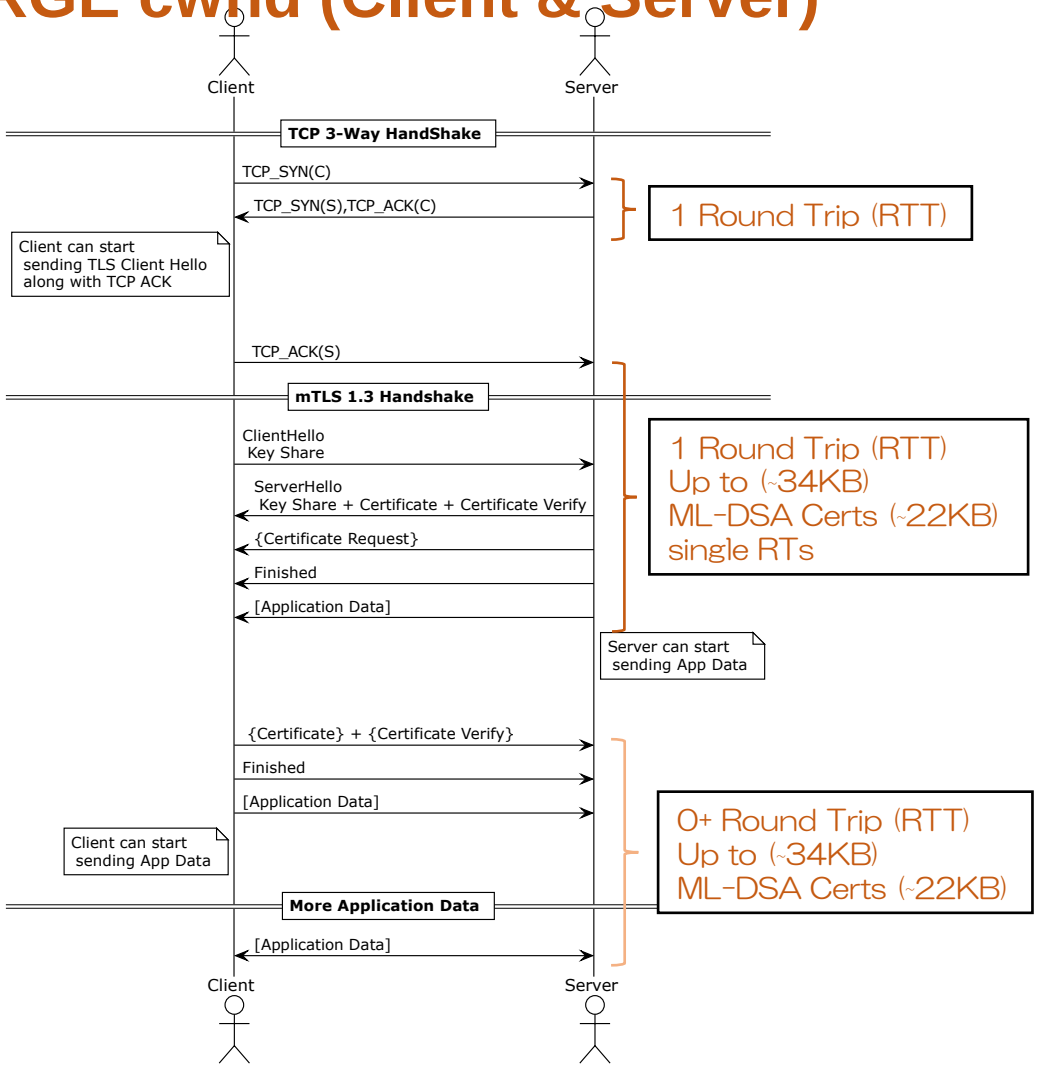




The *actual* mTLSv1.3 handshake performance

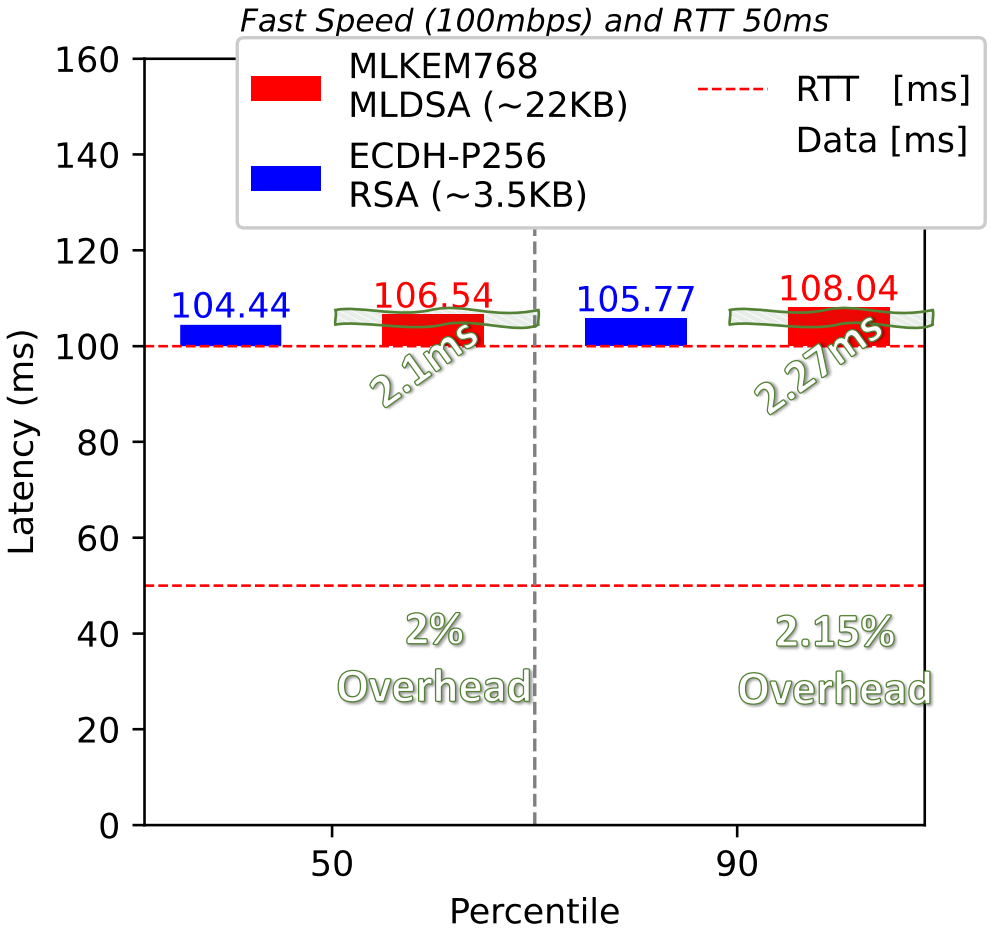
Classical vs. PQ (large) Certs with LARGE cwnd (Client & Server)

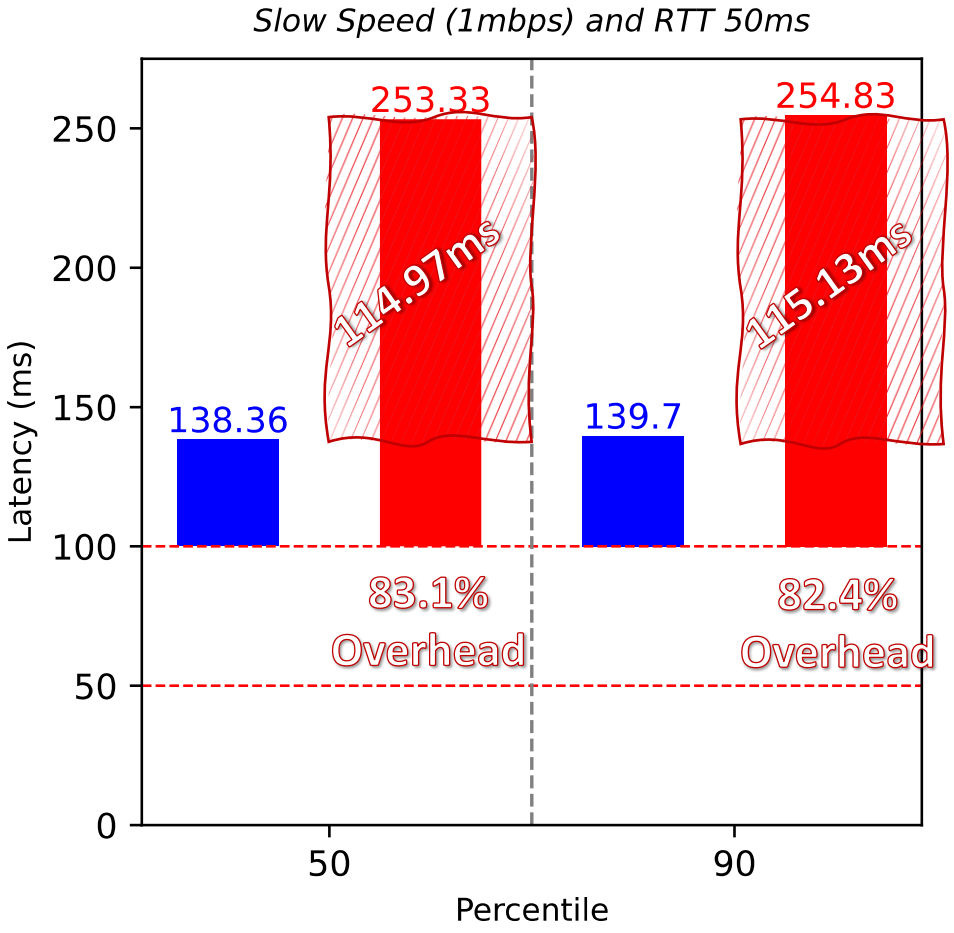




The *actual* mTLSv1.3 handshake performance

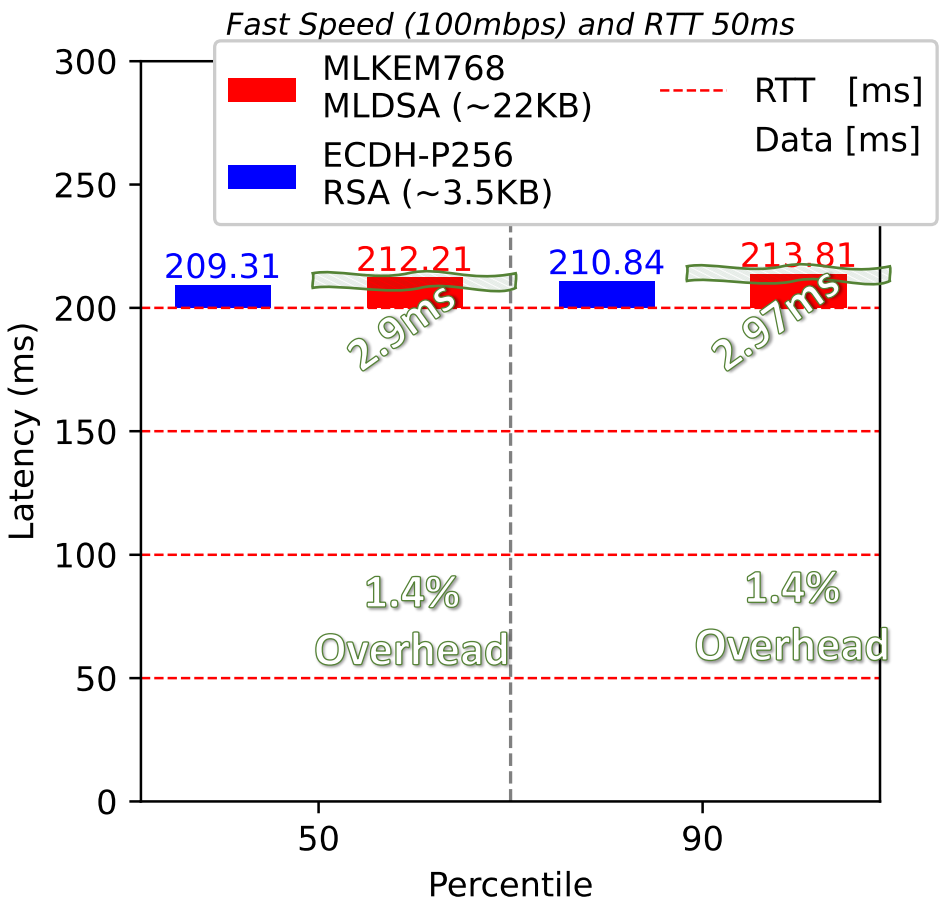
Classical vs. PQ (large) Certs with LARGE cwnd (Client & Server)

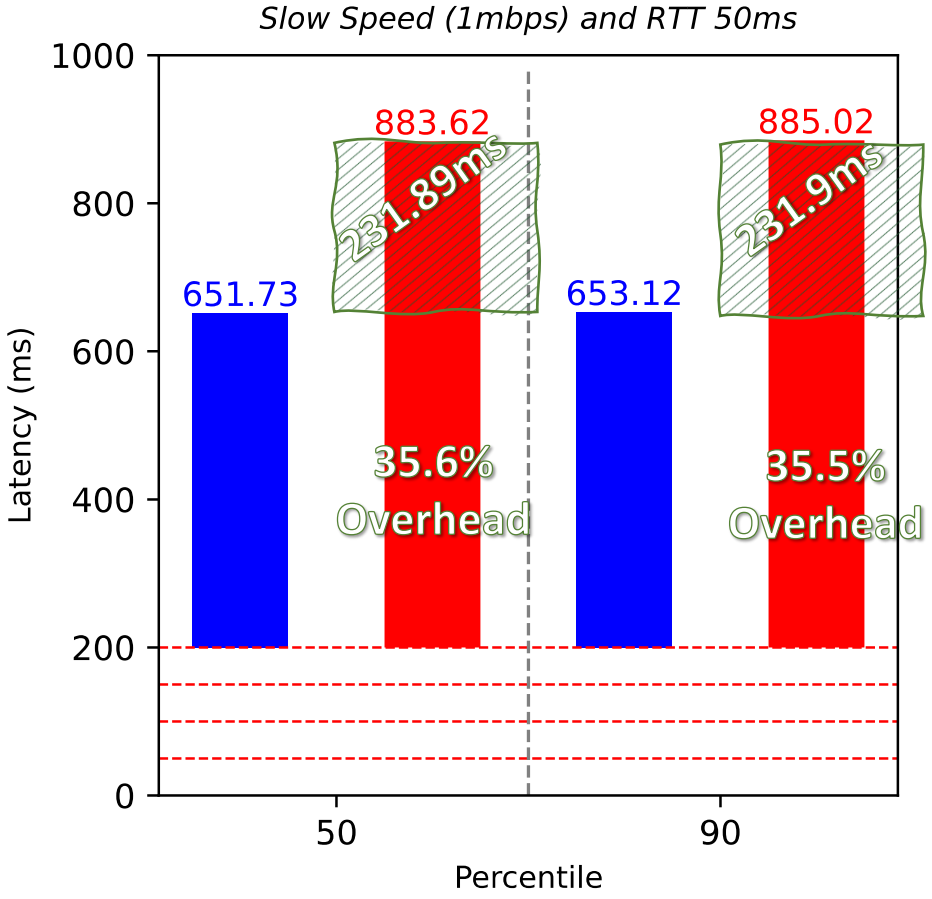




The *actual* mTLSv1.3 performance

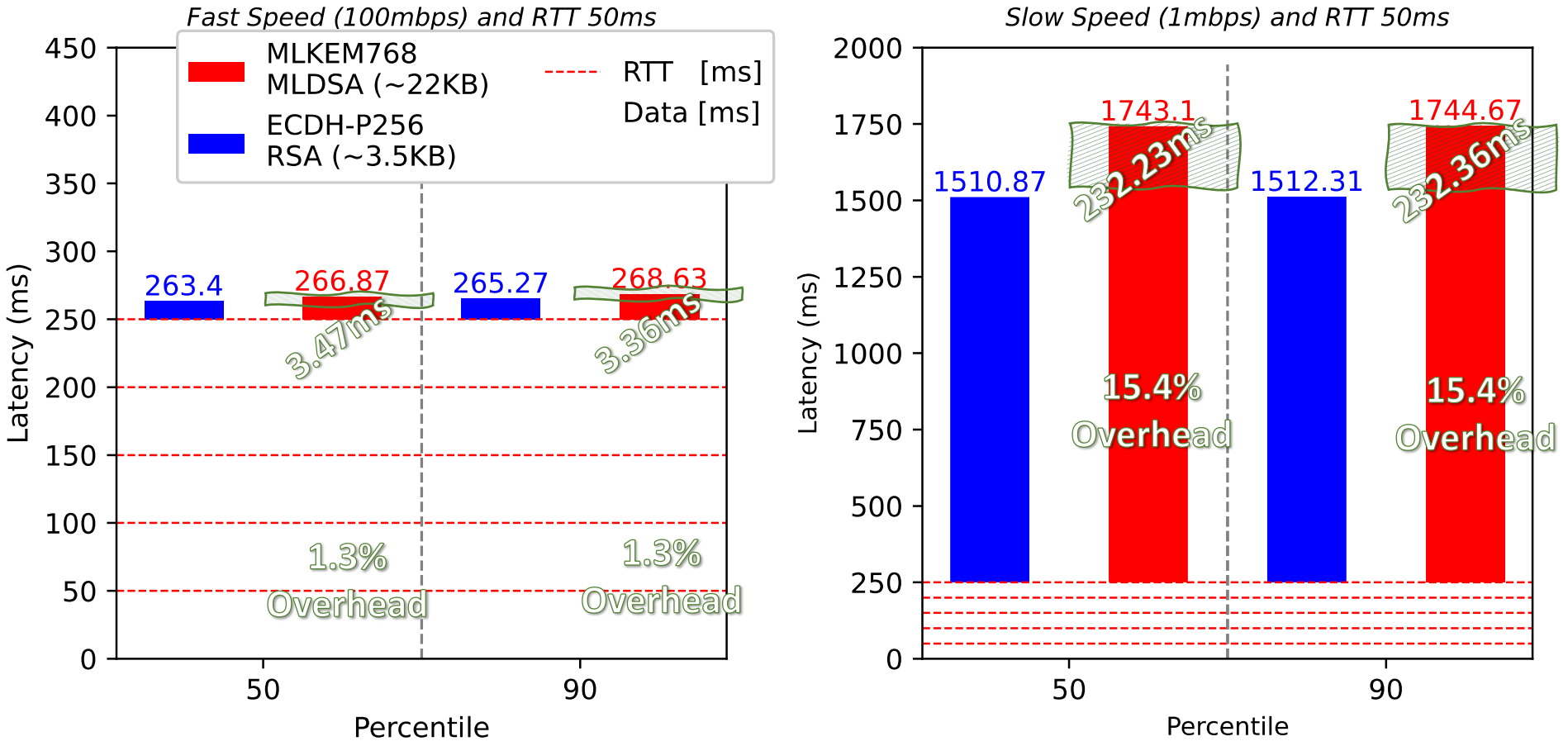
Transfer Data (50KB) - Large cwnd (Client & Server)





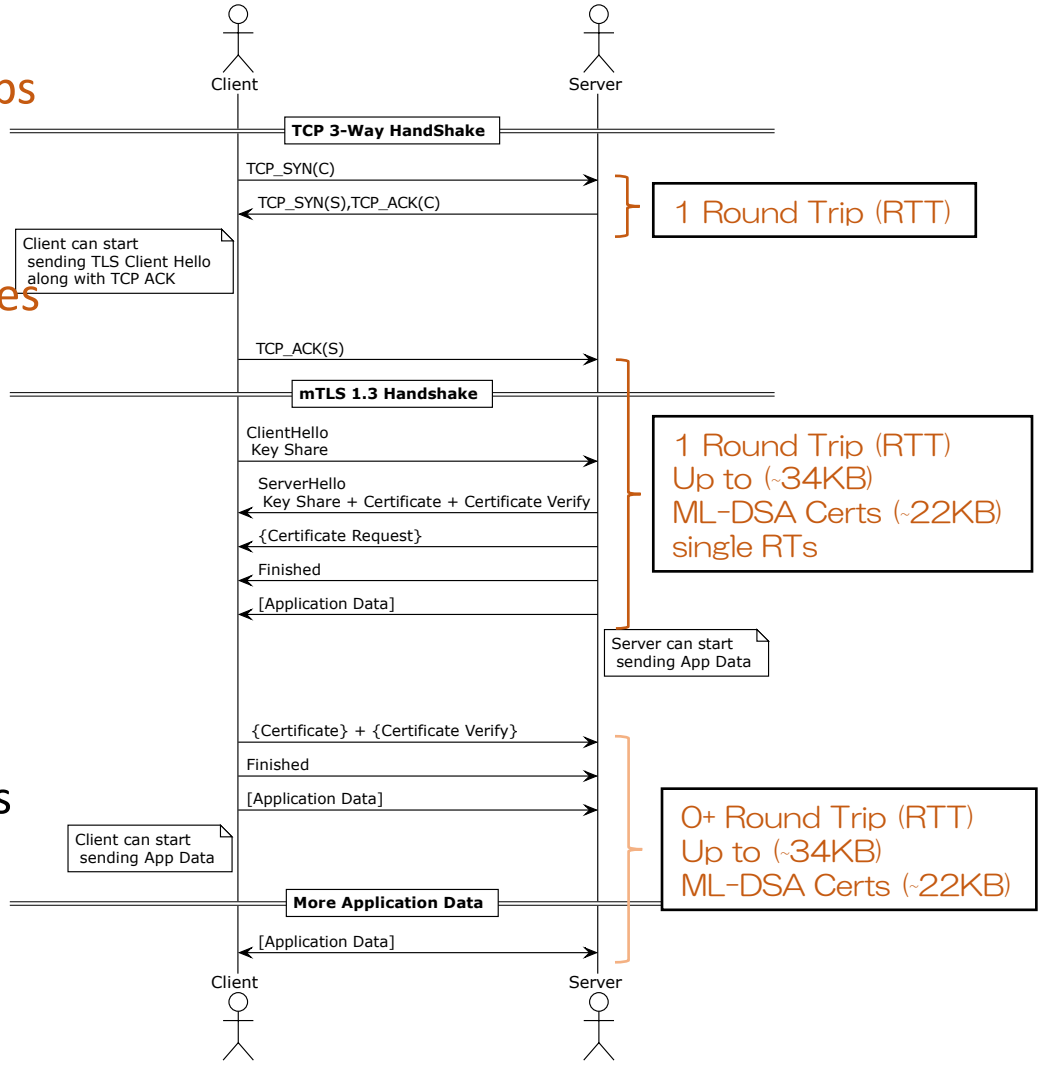
The *actual* mTLSv1.3 performance

Transfer Data (150KB) - Large cwnd (Client & Server)



Conclusions

- PQ mTLS (ML-KEM and ML-DSA) increases the number of round trips with small congestion window.
- Increasing the congestion window allows to exchange PQ certificates in the same number of round trips as classical certificates.
- The main overhead of PQ mTLSv1.3 depends on the network speed
  - 2× ~18.5KB *larger* certificates @100mbps
    - ~ 3.2 ms overhead (mTLS)
  - Application Data transfer is *usually* much larger
- While Time-To-*First*-Byte would be impacted, the user experience is *usually* related to Time-To-*Last*-Byte (Application Data Transfer).
  - The overall impact of PQ mTLS migration (Time-To-*Last*-Byte) decreases with larger data exchange.



Thank you!

Q&A