Challenges and Benefits of Precisely Specifying Congestion Control Algorithms

Ken McMillan (UT Austin)

Lenore Zuck (UIC)

What we are trying to do

Obtain formal specifications of a CCA (New Reno) that allow:

- Formally prove (some) properties of model
- *Automatically* test existing implementations for *conformance* with model

Why are we doing that

Why are we doing that

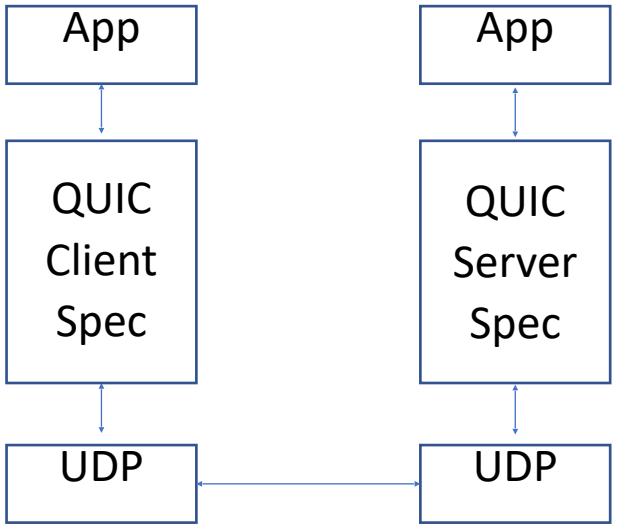
- Formal specification
  - Provides unambiguous reference for protocol definition
  - Clarifies intent and exposes hidden assumptions

Why are we doing that

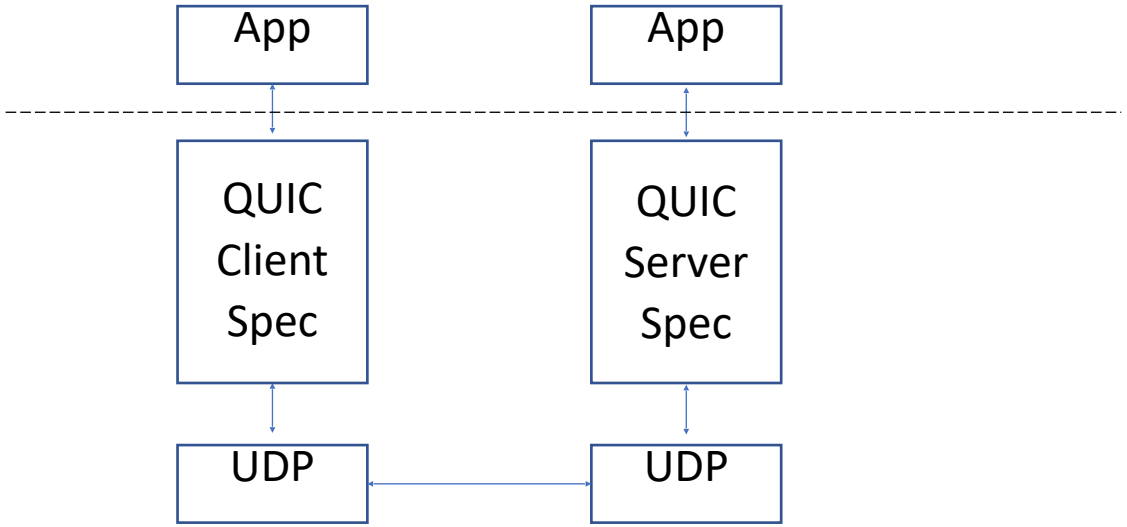
- Formal specification
  - Provides unambiguous reference for protocol definition
  - Clarifies intent and exposes hidden assumptions
- Specification-based testing
  - Connects formal models with reality
  - Exposes conformance errors that interop testing misses
    - Example: downgrade attacks due to non-conformance SSL implementations in the wild
  - Exposes errors and ambiguities in RFC's
  - Exposes weaknesses in formal specifications

Specification-based testing of QUIC

Specification-based testing of QUIC

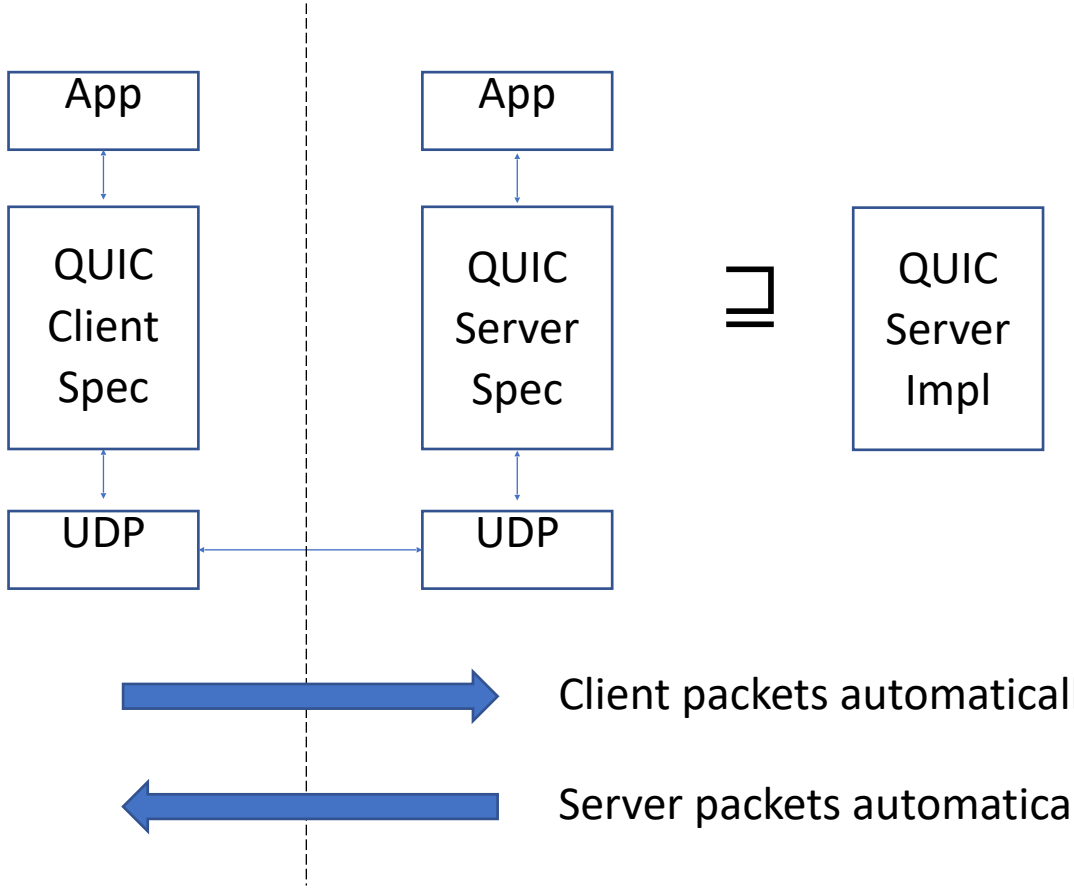


Specification-based testing of QUIC



Formally prove user-level guarantees

Specification-based testing of QUIC



⊒

QUIC
Server
Impl

Test refinement of spec by impl

Client packets automatically generated from spec

Server packets automatically validated against spec

Results of spec-based testing of QUIC

Results of spec-based testing of QUIC

- Numerous corrections to the formal specifications
  - Both strengthening and weakening

Results of spec-based testing of QUIC

- Numerous corrections to the formal specifications
  - Both strengthening and weakening
- Numerous errors uncovered in four implementations
  - Conformance errors
  - Crashes due to low-level coding errors
  - Takeaway: interop testing is not enough to ensure conformance!
    - Specification-based testing produces more general stimuli

Results of spec-based testing of QUIC

- Numerous corrections to the formal specifications
  - Both strengthening and weakening
- Numerous errors uncovered in four implementations
  - Conformance errors
  - Crashes due to low-level coding errors
  - Takeaway: interop testing is not enough to ensure conformance!
    - Specification-based testing produces more general stimuli
- Various of the errors were *exploitable*
  - Off-path denial of service scenario due to RFC's client migration handling
  - Heartbleed-style information leak

Why hard to do same with New Reno

- We need to understand the protocol
- We need a quantitative model of the environment (network)
- We need to understand its (quantitative) guarantees
  - And those of CCA in general

Why hard to do same with New Reno

- We need to understand the protocol
- We need a quantitative model of the environment (network)
- We need to understand its (quantitative) guarantees
  - And those of CCA in general

Network model and quantitative properties need to be agreed on by the community

Understanding New Reno

- The models of the literature (including RFCs) are not precise
  - the behavior after time-out (Exponential Backoff)
- Possible to reverse-engineer from (e.g.) Linux implementation
  - Ideally, the specs should not require that

Quantitative Model of Network

- To define CCA's guarantees, there must be a network model
  - For QUIC: a functional model for UDP is clear
- We can make something up
  - How to know it is "good enough"
- Impossible to define properties without a good quantitative model of the network

Properties of New Reno (& CCAs)

Properties of New Reno (& CCAs)

- Note: in QUIC, there is a consensus on the simple functional guarantees of the Transport Layer

Properties of New Reno (& CCAs)

- Note: in QUIC, there is a consensus on the simple functional guarantees of the Transport Layer
- CCA’s guarantees are harder to distill esp. in view of ill-defined network

Properties of New Reno (& CCAs)

- Note: in QUIC, there is a consensus on the simple functional guarantees of the Transport Layer
- CCA's guarantees are harder to distill esp. in view of ill-defined network
- There are nice studies of formal properties of CCAs
  - Assuming an ideal AIMD and depending on its A/M constants which are not constant in New Reno
- But hard to apply to New Reno
  - E.g., α-efficiency is that, in steady state, utilization of channel is $\geq \alpha$, where α depends on the constants that aren't
- Studies exclude timeouts
  - we don't understand some behaviors after timeouts (return from ExpBck to SlSt)

Why are we here?

- To (try to) convince you that formal specification is valuable for CCAs
  - Helpful beyond (often impractical) formal verification
  - E.g., parametric analysis of real-time protocols
- To get help in designing good models for networks
  - Essential to derive the right high-level properties
- To get help in understanding CCAs that are in use
  - Understanding CCAs is harder than most other protocols
  - There is no consensus as to the quantitative guarantees in terms of env & CCA
- There may be many definitions of CC in different network environment

Conclusion

- Benefits of formal specifications:
  - formally prove key properties of CCA
  - Rigorously test implementations conform to the specs
  - Both (ioho) have high benefits in engineering CCs
- To create formal specification we need:
  - Definition(s) of network model
  - Definition(s) of CC (possibly dependent on network model)