

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Radek Zikmund	
Název práce	QUIC Protocol Implementation for .NET	
Rok odevzdání	2021	
Studijní program	Informatika	Studijní obor Softwarové a datové inženýrství
Autor posudku	Adam Dingle	Role Oponent
Pracoviště	KSVI	

Text posudku:

In the Introduction to this master's thesis, Radek gives a nice overview of the motivation for developing HTTP/3 and QUIC. He clearly explains the reasons for building an experimental implementation of QUIC in managed C# code and sets clear goals for the thesis work.

In the QUIC Protocol section, Radek gives a clear and detailed overview of the protocol itself. I found his explanation much easier to follow than the official QUIC specification. His diagrams illustrating packet flow are especially helpful.

The Analysis section explains the high-level design of the QUIC implementation. Memory allocation in a managed environment may potentially be expensive, so it's great that the design tries to minimize allocations. The design also pays careful attention to threading issues, integration with an external TLS library, testability, and debugging.

The developer documentation clearly explains the major classes in the system and indicates where to find them in the fork of the large .NET repository. The diagrams illustrating the relationships between classes are especially helpful.

The user documentation is quite clear. It's great that it explains the complex build process in a lot of detail, and I appreciate that Radek included binaries so that a reader of this thesis does not necessarily have to go through the whole build process.

The evaluation section is the goal of the entire thesis work, so I'm pleased that it goes into detail and attempts to explain the experimental observations. I'm surprised that the managed QUIC implementation has a much larger throughput (by a factor of almost 4) than msquic with 256-byte messages and a small number of connections (e.g. less than 8), as visible in figures 6.1 and 6.5. It would be nice to have some explanation of that fact. Perhaps msquic is simply not optimized for such small messages. It's interesting that this same throughput difference does not occur in the simulated cellular network in Figure 6.8, where the two QUIC implementations have similar throughput. Presumably that is because the 25-ms latency in this environment causes a throughput bottleneck, as explained in the text. In the end it's a bit disappointing that both QUIC implementations have less throughput than TCP in the cellular simulation, since a goal of QUIC is to improve performance in situations where packets may be lost, e.g. by preventing head-of-line blocking. However, current TCP implementations are the result of decades of implementation and optimization work, so it is not surprising that it will take time to match their performance.

In summary, my opinion is this thesis work is outstanding. Implementing even a subset of the complex QUIC protocol is a significant undertaking and I believe that the author succeeded in that implementation. The entire thesis text is written exceptionally clearly,

with detailed analysis of important points and very helpful diagrams throughout. As a native English speaker, I found the text very easy to read and the only grammatical errors I noticed were occasional missing articles (i.e. "a" or "the"). I definitely recommend that this thesis work be accepted.

Práci doporučuji k obhajobě.

Práci nenavrhují na zvláštní ocenění.

Pokud práci navrhuje na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).

Datum 27.1.2021

Podpis