

Evaluation des Transportprotokolls QUIC in OMNeT++

Denis Lugowski

Denis.Lugowski@haw-hamburg.de

Hauptseminar

Master Informatik

HAW Hamburg

Die bisherige Entwicklung des Internets war bisher stark geprägt vom Bandbreitenausbau. Eine ebenso wichtige Rolle hat die Latenz bei der Übertragung von Daten, die in den aktuell verwendeten verbindungsorientierten Transportprotokollen im Internet wenig Beachtung findet. Das Protokoll QUIC adressiert diesen Aspekt der Datenübertragung und führt mit Funktionen wie Stream Multiplexing, einer verschlüsselten Datenübertragung oder der Implementierung im Userspace neue Verbesserungen ein, die den stetig steigenden Anforderungen des Internets gerecht werden sollen. Um die Verbesserungen quantitativ unter kontrollierten Bedingungen erfassen zu können, bedarf es eines Simulationsmodells von QUIC, das zu diesem Zeitpunkt noch nicht existiert. Das Ziel im Masterprojekt ist die Entwicklung und Evaluation eines Simulationsmodells von QUIC in OMNeT++. Die Vorgehensweise und dabei verwendete Methodik zur Realisierung des Vorhabens werden in dieser Ausarbeitung dargestellt.

Inhaltsverzeichnis

1	Einführung	3
2	Hintergrund	4
2.1	QUIC Transportprotokoll	4
2.2	Simulationsframework OMNeT++	5
3	Relevante Arbeiten	5
4	Motivation	8
5	Vorgehen und Methodik	9
5.1	Bildung eines Teams	9
5.2	Machbarkeits-, Notwendigkeitsanalyse	9
5.3	Evaluation	10
6	Risiken	11
7	Ausblick	11

1 Einführung

Die Anzahl der Teilnehmer im Internet steigt stetig und hat im Juli 2017 bei der ISC Internet Domain Survey einen Höchststand von ungefähr 1,1 Mrd. erreicht [1]. Mit der Anzahl an Hosts ist auch ein wachsendes Bedürfnis an schnelleren und zuverlässigeren Internetverbindungen entstanden, um den Anforderungen neuer und bestehender Applikationen nachzukommen. In der Vergangenheit wurde zur Beschleunigung des Internetverkehrs viel in den Ausbau der Bandbreite investiert. Im ersten Quartal 2010 lag die weltweit durchschnittliche Bandbreite noch bei unter einem Mbit/s [2]. Sieben Jahre danach beträgt die Bandbreite mit 7,2 Mbit/s bereits mehr als das sieben-fache [3].

Neben der Bandbreite trägt auch die Latenz zu einer schnellen Datenübertragung bei. Eine verbesserte Latenz hat ausgehend von den bereits zur Verfügung stehenden Bandbreiten eine bessere Nutzererfahrung zur Folge, als die weitere Erhöhung der Bandbreite. In [4] zeigt eine stetige Verringerung der Latenz bei einer Bandbreite von 5 Mbit/s eine stetige Verbesserung der Seitenladezeit, während die alleinige Erhöhung der Bandbreite bis auf 10 Mbit/s ab 4 Mbit/s dem Nutzer kaum einen Mehrwert bietet. Das Potential einer verringerten Latenz wurde auf der Applikationsebene durch Google frühzeitig erkannt und mit SPDY hat Google ein Netzwerkprotokoll entwickelt, das die Latenz bei HTTP verringern soll [5]. Da HTTP/2 viele Funktionen von SPDY übernommen hat, wurde im Jahr 2015 die Weiterentwicklung von SPDY zugunsten einer größeren Verbreitung des offenen HTTP/2-Standards aufgegeben [6].

Auf der Transportprotokollebene für verbindungsorientierte Kommunikation gab es in der Vergangenheit wenige Bemühungen, Mechanismen für geringere Latenzen zu etablieren, obwohl das weit verbreitete TCP [7] die Basis für viele Applikationen bildet. Von einer Verbesserung würden somit eine Vielzahl von Anwendungen profitieren. Auch derzeit bestehende Probleme wie das Head-of-line blocking [8] verringern bei Auftreten den Durchsatz und beeinträchtigen alle darauf basierenden Services.

Das in der Standardisierung befindliche Protokoll QUIC widmet sich den Problemen von TCP und bringt zahlreiche Funktionen mit, mit denen die Anforderungen eines schnellen und sicheren Internets besser erfüllt werden sollen.

2 Hintergrund

2.1 QUIC Transportprotokoll

QUIC [9] (ehemals ein Akronym für *Quick UDP Internet Connections*) ist ein von Google initiiertes Projekt [10], das derzeit unter der Leitung der QUIC Working Group an der IETF für die Standardisierung vorbereitet wird. Wie in Abbildung 1 dargestellt, basiert

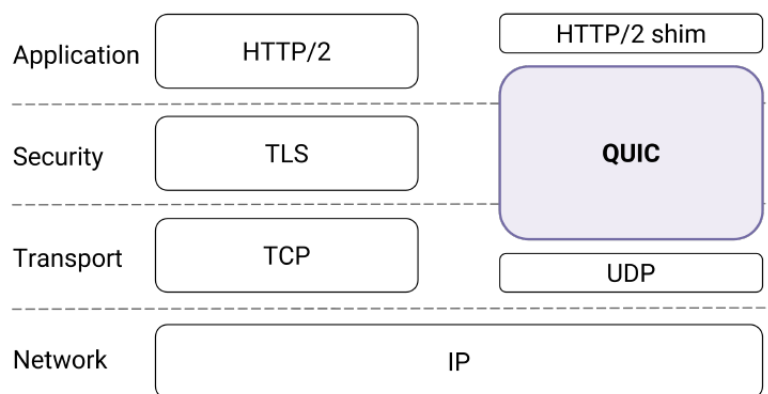


Abbildung 1: Vergleich eines gewöhnlichen Protokollstapels zu QUIC [11]

QUIC auf dem verbindungslosen Transportprotokoll UDP und bietet darüberliegenden Applikationen eine verbindungsorientierte und zuverlässige Ende-zu-Ende-Datenübertragung an. Mit TLS 1.3 bringt QUIC eine Verschlüsselung mit, die sowohl den Inhalt der Applikationen als auch den Header der Transportprotokollebene verschlüsselt. Das eingeführte Multiplexing stellt ein Äquivalent des in HTTP/2 bereits bekannten Multiplexings dar. Wurde noch bei TCP beim Aufkommen eines Paketverlusts die Auslieferung aller nachfolgenden Pakete bis zur erneuten Übertragung des verloren gegangenen Pakets aufgehalten (*head-of-line blocking*), so können bei QUIC durch das Multiplexing vom Paketverlust nicht betroffene Streams uneingeschränkt ihre Daten senden. Wie zuvor beschrieben, ist die Latenz ein bedeutender Faktor bei der Übertragung von Daten. Aus diesem Grund stellt die Verringerung der Latenz bei QUIC ein zentrales Ziel dar, das insbesondere beim Verbindungsaufbau zum Tragen kommt. Wenn der Server und der Client zuvor noch nicht miteinander kommuniziert haben, dann erfolgt der Verbindungsaufbau mit dem *1-RTT-Handshake*, der den kryptographischen Handshake und den des Protokolls innerhalb einer *Round Trip Time* (RTT) in sich vereint. Hierbei werden zusätzlich Verbindungsinformationen in Form eines Cookies beim Client gespeichert. Bei einem erneuten Verbindungsauf-

bau mit demselben Server sendet der Client zur Authentifizierung den zuvor gespeicherten Cookie an den Server und kann zusätzlich daran seine verschlüsselten Daten anhängen, womit für den Versand der ersten Daten keine zusätzliche RTT für den Verbindungsaufbau entfällt (*0-RTT-Handshake*).

Bereits heute kann QUIC mit dem Google Chrome Browser in Verbindung mit den Google Services verwendet werden. Auch viele andere große Organisationen wie Cloudflare, Facebook oder Mozilla arbeiten aktiv an eigenen Implementierungen von QUIC [12], um diese in ihre eigenen Services integrieren zu können. Das große Interesse an QUIC zeigt sich besonders in dessen Verbreitung, obwohl noch kein endgültiger Standard feststeht. Im August 2018 betrug die Anzahl an QUIC-fähigen IPv4-Hosts noch ~186.000. 14 Monate danach, im Oktober 2017, ist die Anzahl auf ~617.000 Hosts angestiegen [13]. Besonders hervorzuheben ist, dass im Jahr 2016 bereits 7% der Internetverkehrs durch QUIC ausgeliefert worden ist, was sich auch auf die intensive Nutzung der Google Services zurückführen lässt [11].

2.2 Simulationsframework OMNeT++

Bei OMNeT++ [14] handelt es sich um ein ereignisorientiertes Simulationsframework für Netzwerke, das die Entwicklung eigener Simulationsmodelle in der Programmiersprache C++ ermöglicht. Es unterstützt die Modellierung von kabelgebundenen und ungebundenen Netzwerken, Queues, Protokollen und allen anderen Systemen, die sich mit einer ereignisorientierten Simulation darstellen lassen. Das INET-Framework [15] stellt eine Erweiterung zu OMNeT++ dar und stellt viele Netzwerkprotokolle wie IPv4, TCP, UDP usw. zur Verfügung. Zum aktuellen Zeitpunkt ist jedoch noch keine Implementierung von QUIC vorhanden.

3 Relevante Arbeiten

Taking a Long Look at QUIC (2017) [16]

In dieser umfassenden Performanceanalyse wurde QUIC mit einem Protokollstack bestehend aus HTTP/2, TLS und TCP verglichen. Dabei wurde untersucht, wie die Seitenladezeit mit einer variierenden Anzahl (1 - 200) und Größe (5 kB - 1 MB) an Ressourcen, unterschiedlichen Latenzen (0 ms - 100 ms), Bandbreiten (10 Mbit/s - 100 Mbit/s) und

Jitter im Internet ausfällt. Es hat sich gezeigt, dass QUIC in fast allen Szenarien dem

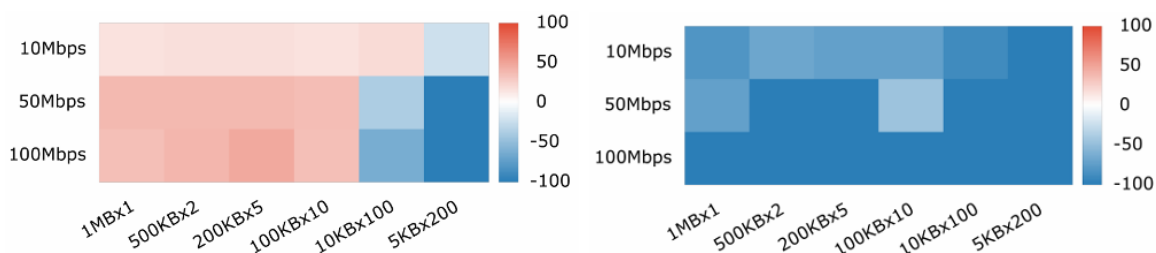


Abbildung 2: QUIC im Vergleich zu TCP [16]

TCP-Protokollstack überlegen ist und eine kleinere Seitenladezeit aufweist. Ausgenommen sind die Szenarien, in denen viele kleine Ressourcen mit einer höheren Bandbreite übertragen wurden. In der linken Abbildung 2 ist dies durch die blaue Farbe ersichtlich, während eine rötliche Farbe ein besseres Abschneiden von QUIC aufzeigt. Kommt zusätzlich ein Jitter von 10 ms hinzu (siehe Abbildung 2 rechts), dann ist eine signifikant schlechtere Performance bei QUIC zu beobachten. Die Ursache liegt nach Ansicht der Autoren an fälschlich angenommenen Paketverlusten bei nicht in Reihenfolge ankommenden Paketen.

How quick is QUIC? (2016) [17]

Diese Arbeit hat einen Vergleich zwischen QUIC, SPDY und HTTP/1.1 unternommen. Auch hier wurde mit einer unterschiedlichen Anzahl und Größe an Ressourcen, unterschiedlichen Latenzen und Bandbreiten getestet. Die Ergebnisse des Vergleichs decken sich mit denen der vorangegangenen Arbeit und zeigen, dass QUIC bei vielen Ressourcen mit einer Größe von 128 kB und einer relativ hohen Bandbreite von 50 Mbit/s eine deutlich höhere Seitenladezeit aufweist als die anderen Protokolle (siehe Abbildung 3).

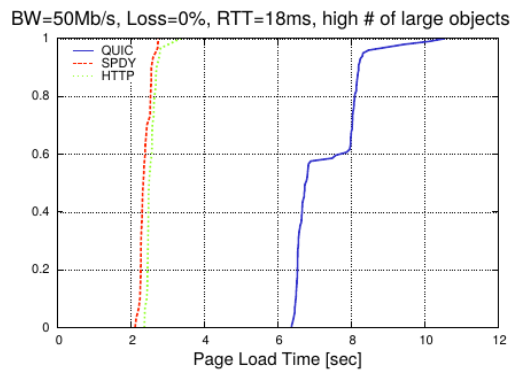


Abbildung 3: QUIC im Vergleich zu SPDY und HTTP/1.1 [17]

Performance Analysis of QUIC Protocol under Network Congestion (2017) [18]

In dieser Masterarbeit wurden QUIC und TCP mit dem Tool *iPerf*, das zur künstlichen Erzeugung von Testdaten und der Messung der Bandbreiten verwendet wird, gegenübergestellt. Die Ergebnisse der Messung ergeben, dass beide Protokolle fast gleich abschneiden. Jedoch ist hinzuzufügen, dass die maximale Linkkapazität in den Tests nur 16 Mbit/s betrug.

Zusammenfassend lässt sich festhalten, dass eine große Anzahl an Ressourcen in Verbindung mit einer hohen Bandbreite negative Auswirkungen auf die Performance von QUIC haben. Der Umfang an Experimenten war bei der zweiten und insbesondere bei der letzten vorgestellten Arbeit relativ klein. Die verwendete Implementierung von QUIC war in allen Arbeiten eine von Google entwickelte Version. Hierbei stellt sich die Frage, ob die aufgetretenen Performanceprobleme durch die Implementierung oder das Protokolldesign bedingt sind. Zuletzt muss erwähnt werden, dass die Experimente der ersten beiden Arbeiten im Internet erfolgten. Der im Internet herrschende *Quality of Service* (QoS) ist *Best Effort*, was bedeutet, dass keine Garantien über die Übertragungsdauer, die gesicherte Paketreihenfolge oder überhaupt über die Ankunft von Paketen ohne Verluste gemacht werden können. Eine Reproduzierbarkeit derselben Ergebnisse ist damit nicht vollständig gegeben. Aus diesem Grund haben die Experimente eine verminderte Aussagekraft über die Performance von QUIC.

4 Motivation

Die im vorhergehenden Kapitel beschriebenen Probleme der Experimente zeigen auf, dass reale Implementierungen des QUIC-Protokolls nur bedingt für wissenschaftliche Studien geeignet sind. Dagegen kann die Durchführung einer Simulation helfen, aussagekräftige Ergebnisse zu produzieren. Die Vorteile einer Simulation im Vergleich zu Experimenten in realen Umgebungen sind:

- **Skalierbarkeit:** Die Anzahl an kommunizierenden Hosts oder die Komplexität und der Umfang der darunter liegenden Infrastruktur können in einer Simulation beliebig skaliert werden. Die einzige Beschränkung ist die Leistungsfähigkeit des ausführenden Systems.
- **Reproduzierbarkeit:** Es macht keinen Unterschied, wie oft eine Simulation durchgeführt wird. Die hervorgebrachten Ergebnisse lassen sich unter denselben eingestellten Bedingungen jederzeit reproduzieren und sind somit nicht vom *Best Effort* QoS abhängig. Damit geht auch eine hohe Testbarkeit einher.
- **Wartbarkeit:** Die Simulation von Hosts oder einer ganzen Infrastruktur machen eine Wartung eines äquivalenten Aufbaus in Form von Updates der Software oder der Hardware in der Realität überflüssig.
- **Kosten-, Zeitersparnis:** Die Verwaltung eines einzigen Systems zur Ausführung der Simulation führt zu einer großen Kosten- und Zeitersparnis, die sonst für die Beschaffung, Einrichtung und Unterhaltung einer äquivalenten realen Umgebung aufgewendet werden müsste.

Da zu diesem Zeitpunkt kein Simulationsmodell von QUIC im INET-Framework von OM-NeT++ für die Durchführung einer umfassenden Simulation existiert, besteht die Aufgabe darin, ein solches Modell für QUIC zu implementieren. Die Implementierung orientiert sich an den Drafts der QUIC Working Group an der IETF und würde im Anschluss an die Fertigstellung der INET-Community zur freien Verfügung gestellt werden. Damit würde ein Beitrag zur offenen Open-Source-Kultur eingehen und somit andere Personen die Durchführung eigener Simulationen anhand des entwickelten Modells ermöglichen.

5 Vorgehen und Methodik

Das Hauptdraft von QUIC [19], an dem die QUIC Working Group aktuell bei der IETF an einer Standardisierung des Protokolls arbeitet, zeigt einen Umfang auf, der für die Entwicklung eines Simulationsmodells von QUIC eine sorgfältige Planung erfordert. Im Folgenden wird das Vorgehen und die dabei verwendete Methodik zur Realisierung des Projekts vorgestellt.

5.1 Bildung eines Teams

Das Vorhaben der Entwicklung eines Simulationsmodells von QUIC in OMNeT++ erfordert aufgrund des Umfangs und der Komplexität die Beteiligung mehrerer Personen am Entwicklungsprozess. Das Team, welches sich zu diesem Zweck zusammengeschlossen hat, besteht aus 8 Personen (3 Universität Duisburg-Essen, 1 FH Münster und 4 HAW Hamburg).

5.2 Machbarkeits-, Notwendigkeitsanalyse

Die Umsetzung eines bisher nur in realen Implementierungen vorhandenen Transportprotokolls wie QUIC in ein äquivalentes Simulationsmodell erfordert eine Analyse der Laufzeitumgebung, in diesem Fall OMNeT++ in Verbindung mit dem INET-Framework und eine Begutachtung aller Funktionen von QUIC hinsichtlich ihrer Notwendigkeit und ihrer Machbarkeit. Dazu wurde ein Plan der Features von QUIC aus dem Hauptdraft in Form einer Feature-Liste erstellt. Im Anschluss wurde eine erste Analyse über die Notwendigkeit und Machbarkeit dieser Funktionen durchgeführt. Dabei stellte sich heraus, dass einige Funktionen wie das Error-Handling, die *Proof of Source Address Ownership* oder die *Stateless Retries* auf einen späteren Zeitpunkt der Entwicklung verschoben werden können, da sie auf die Kernfunktion des Protokolls keinen Einfluss haben. Die Implementierung einer einzigen Version von QUIC in OMNeT++ macht die Verhandlung zwischen zwei Hosts über eine gemeinsame QUIC-Version (*Version Negotiation*) überflüssig und wird deshalb nicht berücksichtigt.

Für die verschlüsselte Datenübertragung ist der Austausch von kryptografischen Informationen während des *Handshakes* notwendig. Die dafür benötigte TLS-Bibliothek steht jedoch im INET-Framework nicht zur Verfügung und müsste selbst umgesetzt werden.

Eine damit einhergehende Verspätung beim Start der QUIC-Implementierung macht es erforderlich, eine Vereinfachung vorzunehmen und die Kommunikation ohne Verschlüsselung ablaufen zu lassen. Anstelle der kryptographischen Informationen werden beim *Handshake* Dummy-Daten versendet. Um weiterhin vergleichbare Ergebnisse mit den realen Implementierungen von QUIC sicherstellen zu können müssen, die Aufwände, die bei den Hosts durch die TLS-Verschlüsselung zustande kommen, im Simulationsmodell Berücksichtigung finden. Das Vorgehen dabei, um dieses Vorhaben zu bewerkstelligen, besteht aus einer empirischen Analyse der Aufwände, die bei der Verschlüsselung entstehen. Die ermittelten Aufwände werden bei der Durchführung der Tests am Modell mit einfließen.

Die Analyse über die Simulationsumgebung und der Funktionen von QUIC hat gezeigt, dass eine vollständige Implementierung für erste Tests nicht erforderlich ist. Insbesondere das Problem des fehlenden Verschlüsselungsaufwands kann durch das nachträgliche Hinzunehmen selbst ermittelter Aufwände aufgefangen werden.

5.3 Evaluation

Das Ziel am Ende des Masterprojekts besteht in einer Evaluation des selbst entwickelten Simulationsmodells von QUIC mithilfe einer Parameterstudie. In solch einer Studie wird eine Auswahl an zu untersuchenden Parametern getroffen und im Anschluss an die Tests die Zusammenhänge dieser analysiert. Die Ergebnisse der vorgestellten Arbeiten dienen als Annahmen und es wird simulativ überprüft, ob die dort dokumentierten Probleme implementierungs- oder protokollbedingte Ursachen haben. Die Umgebung, in der die Tests durchgeführt werden, soll an den realen Bedingungen des Internets mit unterschiedlichen Linkkapazitäten, Routern, die unterschiedlich befüllte Buffer besitzen und einer sich ändernden Topologie orientiert sein. Damit wäre ein Szenario geschaffen, das vergleichbar mit der Internet-Umgebung ist und das jederzeit reproduzierbare und unverfälschte Ergebnisse generieren kann, die besonders bei der Ursachenanalyse hilfreich sind.

6 Risiken

Die Entwicklung eines ganzen Transportprotokolls in OMNeT++ birgt Risiken, die sich in die folgenden drei Kategorien einteilen lassen:

- **QUIC:** Das Protokoll ist noch nicht fertig standardisiert worden und befindet sich deshalb in einem stetigen Entwicklungsprozess. Das birgt das Risiko, dass die für das Simulationsmodell festgelegte QUIC-Version bereits bei der Fertigstellung veraltet sein kann.
- **Simulation:** Die Simulation der Verschlüsselungsaufwände kann dazu führen, dass diese sich von den realen Aufwänden unterscheiden und damit eine Ergebnisabweichung von der realen QUIC-Implementierung zustande kommt. Ebenso kann die selbst erstellte Topologie für die Simulation nicht die realen Bedingungen widerspiegeln und damit auch zu einer Abweichung der Ergebnisse führen.
- **Open-Source-Projekt:** Wie in anderen größeren Projekten birgt der hohe Entwicklungsumfang das Risiko, dass die Analysen nicht im vollem Umfang durchgeführt werden können. Die Nichteinhaltung von Verbindlichkeiten innerhalb des Teams bei der Entwicklung können dieses Risiko noch weiter erhöhen.

7 Ausblick

In dieser Ausarbeitung wurde dargestellt, wie das Vorhaben der Entwicklung eines Simulationsmodells von QUIC und dessen Evaluation bewerkstelligt wird. Einen großen Anteil des Projekts hat der eigentliche Entwicklungsprozess, weshalb eine möglichst schnelle Fertigstellung der Implementierung angestrebt wird. Im Anschluss daran müssen geeignete Testszenarien geschaffen werden, die eine umfassende Analyse von QUIC im Rahmen der Parameterstudie ermöglichen. Des Weiteren erfolgt eine Analyse der Verschlüsselungsaufwände bei der realen QUIC-Implementierung. Während der Durchführung der Tests werden die ermittelten Aufwände künstlich hinzugefügt. Im selben Zug ist eine Analyse der Topologie in einem Teilbereich des Internets durchzuführen, um eine geeignete Abbildung im Simulationsumfeld zu schaffen.

Literatur

- [1] Internet Systems Consortium. (2017). ISC Internet Domain Survey Host Count, Adresse: <https://www.isc.org/network/survey/> (besucht am 23.07.2018).
- [2] Akamai Technologies. (2010). Akamai Announces First Quarter 2010 State Of The Internet Report, Adresse: <https://www.akamai.com/us/en/about/news/press/2010-press/akamai-announces-first-quarter-2010-state-of-the-internet-report.jsp> (besucht am 23.07.2018).
- [3] D. Belson, Hrsg. (2017). Akamai's State of the Internet Report Q1 2017, Adresse: <https://www.akamai.com/fr/fr/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf>.
- [4] I. Grigorik, *High Performance Browser Networking*. O'Reilly Media, Incorporated, 2013, Kap. 10, ISBN: 9781449344764. Adresse: <https://hpbn.co/primer-on-web-performance/#latency-as-a-performance-bottleneck>.
- [5] M. Belshe und R. Peon, "SPDY Protocol", IETF Secretariat, Internet-Draft draft-mbelshe-httpbis-spdy-00, Feb. 2012, <http://www.ietf.org/internet-drafts/draft-mbelshe-httpbis-spdy-00.txt>. Adresse: <http://www.ietf.org/internet-drafts/draft-mbelshe-httpbis-spdy-00.txt>.
- [6] C. Bentzel. (9. Feb. 2015). Hello HTTP/2, Goodbye SPDY, Adresse: <https://blog.chromium.org/2015/02/hello-http2-goodbye-spdy.html> (besucht am 23.07.2018).
- [7] J. Postel, "Transmission Control Protocol", RFC Editor, STD 7, Sep. 1981, <http://www.rfc-editor.org/rfc/rfc793.txt>. Adresse: <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [8] I. Grigorik, *High Performance Browser Networking*. O'Reilly Media, Incorporated, 2013, Kap. 10, ISBN: 9781449344764. Adresse: <https://hpbn.co/building-blocks-of-tcp/#head-of-line-blocking>.
- [9] J. Iyengar und M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", IETF Secretariat, Internet-Draft draft-ietf-quic-transport-13, Juni 2018, <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-13.txt>. Adresse: <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-13.txt>.
- [10] Jana und I. Swett, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2", IETF Secretariat, Internet-Draft draft-tsvwg-quic-protocol-00, Juni 2015, <http://www.ietf.org/internet-drafts/draft-tsvwg-quic-protocol-00.txt>. Adresse: <http://www.ietf.org/internet-drafts/draft-tsvwg-quic-protocol-00.txt>.

- [11] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kou-ranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tennesi, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang und Z. Shi, “The QUIC Transport Protocol: Design and Internet-Scale Deployment”, in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, Ser. SIGCOMM ’17, Los Angeles, CA, USA: ACM, 2017, S. 183–196, ISBN: 978-1-4503-4653-5. DOI: [10.1145/3098822.3098842](https://doi.acm.org/10.1145/3098822.3098842). Adresse: <http://doi.acm.org/10.1145/3098822.3098842>.
- [12] QUIC WG. (2018). QUIC Implementierungen, Adresse: <https://github.com/quicwg/base-drafts/wiki/Implementations> (besucht am 24. 07. 2018).
- [13] J. R uth, I. Poesse, C. Dietzel und O. Hohlfeld, “A First Look at QUIC in the Wild”, in *Passive and Active Measurement*, Springer International Publishing, 2018, S. 255–268, ISBN: 978-3-319-76481-8.
- [14] A. Varga. (). Simulation Manual OMNeT++ version 5.3, Adresse: <https://www.omnetpp.org/doc/omnetpp/manual/> (besucht am 21. 06. 2018).
- [15] Opensim Ltd. (). What Is INET Framework?, Adresse: <https://inet.omnetpp.org/Introduction.html> (besucht am 24. 07. 2018).
- [16] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru und A. Mislove, “Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols”, in *Proceedings of the 2017 Internet Measurement Conference*, Ser. IMC ’17, London, United Kingdom: ACM, 2017, S. 290–303, ISBN: 978-1-4503-5118-8. DOI: [10.1145/3131365.3131368](https://doi.acm.org/10.1145/3131365.3131368). Adresse: <http://doi.acm.org/10.1145/3131365.3131368>.
- [17] P. Megyesi, Z. Kr amer und S. Moln ar, “How quick is QUIC?”, in *2016 IEEE International Conference on Communications (ICC)*, Mai 2016, S. 1–6. DOI: [10.1109/ICC.2016.7510788](https://doi.org/10.1109/ICC.2016.7510788).
- [18] A. Srivastava, “Performance Analysis of QUIC Protocol under Network Congestion”, Master Thesis, Worcester Polytechnic Institute, 2017.
- [19] J. Iyengar und M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport”, IETF Secretariat, Internet-Draft draft-ietf-quic-transport-10, M arz 2018, <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-10.txt>. Adresse: <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-10.txt>.