

# QUIC - Quick UDP Internet Connections

Denis Lugowski

Denis.Lugowski@haw-hamburg.de

Grundseminar

Department Informatik

HAW Hamburg

**Die Protokolle zum Transport von Daten im Internet sind trotz ihres Alters bis heute in ihrer Grundfunktion unverändert geblieben. So beherrschen TCP und UDP seit jeher das Geschehen in der Transportebene und bilden somit das Fundament für nahezu jede Netzwerkanwendung. Mit einem neuen Transportprotokoll namens QUIC hat Google einen Ansatz vorgestellt, mit dem der bestehende Transport der Daten verbessert werden soll, um auch in Zukunft den ständig steigenden Anforderungen an das Internet gerecht zu werden. Wie auch TCP bietet QUIC eine zuverlässige und in Reihenfolge gesicherte Übertragung an. Zusätzlich führt QUIC zahlreiche Neuerungen ein, die sich besonders den Schwächen von TCP widmen. In der Praxis findet QUIC bereits Anwendung in der Auslieferung der Services von Google. Jedoch mangelt es an einer Simulation des Protokolls, mit der die genaue Funktionsweise in Hinblick auf bestimmte Testszenarien untersucht werden kann. Die vorliegende Ausarbeitung wird sich daher genauer mit den Problemen der derzeitigen Transportprotokolle beschäftigen und einen Einblick in die grundlegenden Funktionen und die Simulation von QUIC geben.**

# 1 Einführung

Seit dem Jahr 1993 haben die Kernprotokolle des Internets keine wesentlichen Veränderungen mehr erfahren [1]. In diesem Jahr wurde das *Classless Inter-Domain Routing* [2] eingeführt, welches den Adressraum von IPv4 in Netzklassen unterteilt. Besonders die Transportprotokolle TCP und UDP sind lange vernachlässigt geblieben. So kann bei TCP nur auf die Einführung der *Congestion Control* [3] im Jahr 1984 zurückgeblickt werden, die Mechanismen definiert hat, um Überlast im Netzwerk zu vermeiden, während UDP seit seiner Einführung keine bedeutenden Änderungen oder Neuerungen vorzuweisen hat. Dagegen haben sich die Anforderungen an das Internet stark verändert. Die bloße Übermittlung von Information in Form einer E-Mail oder einer simplen Webseite wurde um zahlreiche weitere Möglichkeiten erweitert, wie z.B. Videos und Musik, wodurch sich das Internet zu einer Plattform zur Bereitstellung von multimedialen Inhalten entwickelt hat. Ein großer Faktor, der zu dieser Entwicklung beigetragen hat, ist das rasante Wachstum der Teilnehmer im Internet.

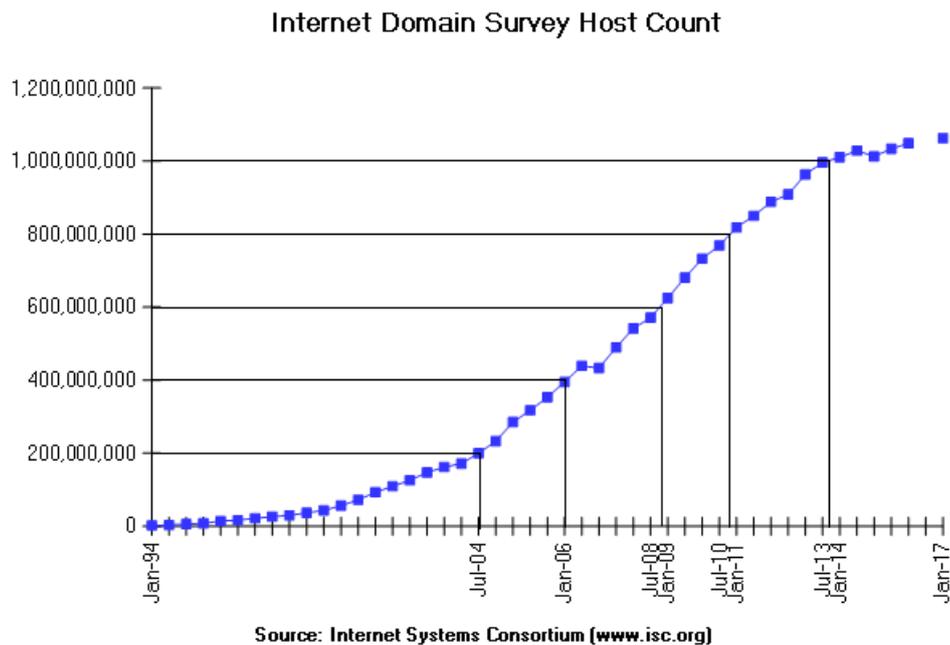


Abbildung 1: Anzahl der Hosts im Internet [4]

Die Abbildung 1 des *Internet Systems Consortiums* (ISC) gibt eine Übersicht über die Anzahl der mit dem Internet verbundenen Hosts. Die Bestimmung dieser Zahlen wird seit 1998 mithilfe eines *Reverse DNS Lookups* realisiert, welches die Anzahl der zu tätigen Anfragen drastisch

reduziert [5]. Insbesondere seit dem Jahr 2004 hat sich die Hostanzahl stark erhöht. So ist bereits nach knapp zwei Jahren eine Verdopplung zu verzeichnen. Die letzten Messungen deuten auf eine langsamere aber stetige Zunahme hin zu 1,2 Mrd. Hosts.

## 2 Weiterentwicklung des Internets

Es ist von großer Bedeutung, dass die bestehenden und neu hinzukommenden Hosts weiterhin uneingeschränkt am Internet teilhaben können. Gleichzeitig steigt die Nachfrage an immer schnelleren Ladezeiten sowie sichereren und robusteren Verbindungen. Aus diesen Gründen ist eine stetige Weiterentwicklung des Internets unabdingbar. Eine Weiterentwicklung lässt sich durch zwei Möglichkeiten realisieren:

### 1. *Ausbau der Infrastruktur*

Dieser Ansatz sieht beispielsweise ein Aufrüsten der einzelnen Netzwerkknoten vor, indem ein Austausch durch leistungsfähigere Modelle geschieht oder den erweiterten Ausbau von Glasfaser.

### 2. *Effizientere Ausnutzung vorhandener Ressourcen*

Das Ziel dieses Ansatzes besteht darin, die von der Infrastruktur unabhängigen Protokolle derart weiterzuentwickeln, dass bei gleicher Leistung weniger Ressourcen in Anspruch genommen werden müssen oder eine bessere Leistung bei gleichem Ressourcenverbrauch erzielt werden kann.

Der Nachteil beim Ausbau der Infrastruktur ist darin zu sehen, dass dieser mit hohen Kosten verbunden ist und außerdem nur Zugang für die *Internet Service Provider* besteht. Auch die Verbreitung dieser Umsetzung ist ungenügend, da meist nur ein kleiner Teil von der Weiterentwicklung profitiert. Dagegen hat der zweite Ansatz den Vorteil, dass keine kostenaufwändige Überholung der Infrastruktur stattfinden muss. Außerdem ist es möglich, mithilfe eines Updates oder dem Austausch einer Software großflächig neue oder effizientere Protokolle zu verbreiten, wodurch innerhalb kürzester Zeit ein großer Teil der Hosts von der Weiterentwicklung ihren Nutzen ziehen könnte. Aus diesen genannten Gründen stellt die Möglichkeit der effizienteren Ausnutzung der vorhandenen Ressourcen einen willkommenen Anlass dar, um Forschungen am Internet hinsichtlich der gegenwärtigen und zukünftigen Anforderungen zu unternehmen

## 2.1 Gegenwärtige Anforderungen

Das Internet bietet eine Vielzahl an Qualitätskriterien, deren Optimierung sich positiv auf die Kommunikation im Internet auswirken kann. Jedoch muss genau abgewogen werden, welche Kriterien einer solchen bedürfen. Im Folgenden werden für wichtig erachtete Qualitätskriterien aufgelistet und ihre Notwendigkeit einer Optimierung diskutiert.

### A. *Robustheit*

Die Robustheit beschreibt, bis zu welchem Grad eine Komponente oder ein System bei fehlerhaften Eingaben oder unter widrigen Bedingungen funktioniert [6]. Im Falle der Netzwerkkommunikation stellt der Verlust von Datenpaketen eine solche Widrigkeit dar. In solch einer Situation ist es oftmals wünschenswert, dass eine Korrektur dieser Ausfälle stattfindet, um die Zuverlässigkeit der Kommunikation zu gewährleisten. Für diesen Zweck hält TCP Mechanismen zur erneuten Übertragung der verloren gegangenen Datenpakete bereit wie das *Fast Retransmit* [3] oder den *Retransmission Timeout* [7]. Den Anforderungen einer gewöhnlichen Netzwerkkommunikation genügen diese Mechanismen, weshalb hier kein Bedarf an einer Weiterentwicklung besteht.

### B. *Bandbreite*

Bei der Bandbreite handelt es sich um eine Größe die angibt, wieviele Daten innerhalb einer Zeitspanne übertragen werden. In den letzten Jahren wurde viel investiert, um ausreichend große Bandbreiten zur Verfügung stellen zu können. So lag im ersten Quartal 2010 die weltweit durchschnittliche Bandbreite noch bei unter einem Mbit/s [8]. Knapp sieben Jahre danach ist die Geschwindigkeit um mehr als das sieben-fache auf 7 Mbit/s angestiegen [9]. Anhand dieser Zahlen zeigt sich, dass keine Änderung der aktuellen Bandbreiten-Entwicklung vonnöten ist und der Ausbau höherer Bandbreiten weiter fortgesetzt wird.

### C. *Sicherheit*

Unter dem Begriff Sicherheit sind sowohl die Sicherheit gegenüber Angriffen von außen (*security*) als auch die innere Sicherheit (*safety*) zusammenzufassen. Hier ist festzustellen, dass bereits zahlreiche Verfahren und Maßnahmen zur Verfügung stehen, die eine angemessene Sicherheit gewährleisten können [10]. Es liegt in der Verantwortung der Entwickler, diesen Richtlinien zu folgen und sie richtig anzuwenden.

### D. *Verzögerung*

Im Vergleich zu den anderen Qualitätskriterien blieb die Verzögerung lange unbeachtet.

Erst seit dem Vorhandensein größerer Bandbreiten tritt diese Thematik immer mehr in den Vordergrund, da sie einen größeren Einfluss auf die Kommunikation besitzt als noch in der Vergangenheit, als die Bandbreite oft der limitierende Faktor war. Im folgenden Abschnitt wird der Effekt einer verbesserten Verzögerung erläutert.

### 2.1.1 Potenzial einer verbesserten Verzögerung

Die Bandbreite ist, wie in Abschnitt B. erläutert, nur eine Größe über die Anzahl der übertragenen Daten. Allerdings ist auch die Zeit zur Auslieferung von Daten von großer Bedeutung, wie die folgende Abbildung veranschaulicht:

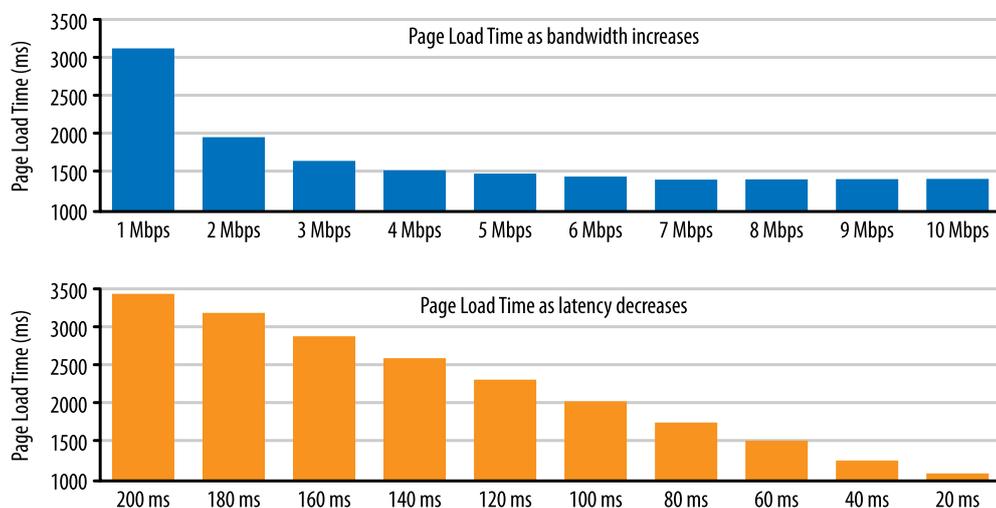


Abbildung 2: Ladezeiten bei Erhöhung/Verringerung der Bandbreite und Verzögerung [11]

Zu sehen ist in beiden Diagrammen die Ladezeit einer Webseite in Millisekunden. Die obere Abbildung zeigt, wie sich die Ladezeit bei einer stetigen Erhöhung der Bandbreite von 1 Mbit/s bis hin zu 10 Mbit/s verhält. Die Verzögerung stellt hier die *Round Trip Time* (RTT) zwischen dem Server und dem Client dar und wurde in diesem Szenario fest auf 60 ms eingestellt. Beträgt die Seitenladezeit bei einer Bandbreite von 1 Mbit/s noch über 3 s, so sind es bei 4 Mbit/s bereits weniger als die Hälfte. Weitere Erhöhungen der Bandbreite bewirken kaum noch Verbesserungen in der Ladezeit, sodass ab ca. 6 Mbit/s eine Stagnation festzustellen ist.

Unten ist hingegen eine stetige Verringerung der Verzögerung um 20 ms beginnend bei 200 ms abgebildet. Die Bandbreite wurde für die Messungen auf 5 Mbit/s fixiert. Es wird hier deutlich, dass mit jeder Abnahme der Verzögerung eine kontinuierliche Abnahme der Seitenladezeit ein-

hergeht, während die Erhöhung der Bandbreite zuvor früher limitiert war.

Dieses Experiment führt zur Erkenntnis, dass trotz einer relativ geringen Bandbreite von 5 Mbit/s, welche unter dem weltweiten Durchschnitt liegt (siehe Abschnitt B.), ein besseres Resultat hinsichtlich der Ladezeit einer Webseite erzielt werden konnte als durch die bloße Erhöhung der Bandbreite.

## 2.2 Verzögerungen am Beispiel von HTTP

Diese starken Auswirkungen der Verzögerung auf die Seitenladezeit sind auf einige Probleme in HTTP zurückzuführen. HTTP/1.1 [12] bringt die Möglichkeit mit, für den Download einer Webseite mehrere TCP-Verbindungen gleichzeitig zu erstellen, um mehrere Ressourcen parallel herunterladen zu können. Dieses Vorgehen stellt eine wesentliche Verbesserung gegenüber dem Download über eine einzelne Verbindung dar. Jedoch muss jede dieser Verbindungen sowohl die Phase des Verbindungsaufbaus als auch die Slow-Start-Phase zur Herstellung der optimalen Übertragungsgeschwindigkeit durchlaufen, was sich negativ auf die Aufrufgeschwindigkeit einer Webseite auswirkt. Desweiteren leidet HTTP/1.1 unter dem Problem des sogenannten *head-of-line blocking*. Dieses Problem beschreibt die Situation, in der eine Ressource vor einer anderen zum Versand zum Client bereit steht, jedoch die Erstgenannte auf die Auslieferung der anderen Ressource warten muss und so eine Blockade verursacht. Bei HTTP/1.1 kommt dieses Problem aufgrund der Beschränkung zustande, dass der Server Anfragen nur in der Reihenfolge beantworten kann, in der sie bei diesem eintreffen [13]. Somit kann pro Verbindung nur die Bearbeitung einer Anfrage zur Zeit stattfinden.

Der Nachfolger HTTP/2 [14] hat sich den Nachteilen von HTTP/1.1 angenommen und zahlreiche Verbesserungen eingeführt. Die wesentlichste Neuerung ist das *Multiplexing* von Anfragen und Antworten. Dabei wird jedem Anfrage-Antworten-Paar mit den Ressourcen der Webseite eine *Stream ID* zugeordnet. Innerhalb eines *Streams* werden die Daten in *Frames* unterteilt und stehen in dieser Form in keiner Abhängigkeit zueinander. Dies hat den Vorteil, dass die *Frames* in beliebiger Reihenfolge unabhängig voneinander versendet werden können und eine Auslieferung unterschiedlicher Ressourcen gleichzeitig vonstatten gehen kann. Dadurch entfällt das Problem des *head-of-line blocking*. Ebenso ist es durch die fehlende Abhängigkeit nicht notwendig mehr als eine TCP-Verbindung auf einmal zu erstellen. Dies hat einen deutlich beschleunigten Verbindungsaufbau zur Folge, da nur noch eine TCP-Verbindung benötigt wird und die Slow-Start-Phase nur ein einziges Mal durchlaufen werden muss.

### 3 QUIC-Protokoll

Die vorangegangenen Verbesserungen von HTTP/2 hinsichtlich der Verzögerung betrafen bisher nur die Applikationsebene. Mit dem verbindungsorientierten Protokoll QUIC (*Quick UDP Internet Connections*) [15] hat Google einen Ansatz entwickelt, mit dem die Verzögerungen in der Transportebene zum Teil drastisch reduziert werden können. Dabei übernimmt QUIC die Eigenschaften von TCP und gewährleistet darüber liegenden Applikationen so eine zuverlässige und in Reihenfolge gesicherte Übertragung. Zusätzlich bringt das Protokoll Funktionen mit, die

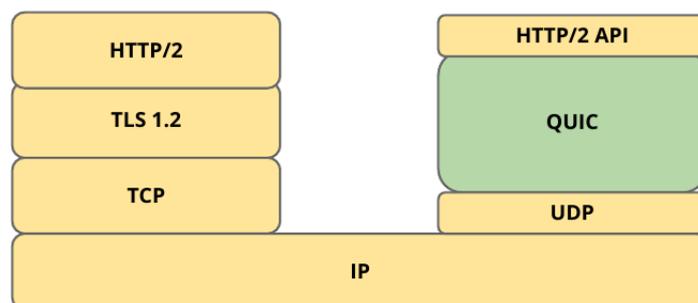


Abbildung 3: QUIC vereint Eigenschaften von TCP, TLS und HTTP/2 [16]

eine verschlüsselte und *multiplexed* Kommunikation erlauben. Insgesamt vereint QUIC damit die Eigenschaften der Protokolle TCP, TLS und HTTP/2 (Abbildung 3).

Es ist besonders hervorzuheben, dass QUIC auf dem verbindungslosen und unzuverlässigen UDP aufbaut. Dadurch wird die Problematik vermieden QUIC auf eine hohe Akzeptanz gegenüber *Middleboxes* (z.B. NAT, Firewall) zu optimieren, da UDP bereits von diesen breit unterstützt wird. Außerdem erlaubt dieser Schritt eine Implementierung von QUIC im *User Space* des Betriebssystems anstelle des für Transportprotokolle sonst üblichen *Kernel Space*. Dies ermöglicht es QUIC wesentlich schneller zu verbreiten und auch Updates einzuspielen [16]. So verfügt zum Beispiel der weit verbreitete Browser Google Chrome bereits über eine Unterstützung für QUIC [17].

#### 3.1 Multiplexing

Das am Beispiel von HTTP/2 erläuterte *Multiplexing* (Abschnitt 2.2) hat in Form von QUIC nun auch in der Transportebene Einzug gefunden. Der Grund dafür ist, dass auch TCP selbst

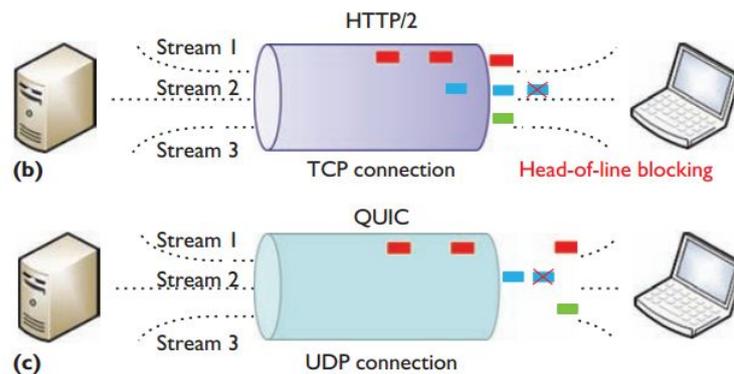


Abbildung 4: Vergleich TCP und QUIC Datenübertragung bei Paketverlust [16]

am Problem des *head-of-line blocking* leidet. In der Abbildung 4 ist die Übertragung von Daten über TCP und QUIC mit HTTP/2 zu sehen. Während dieser Übertragung kommt es auf dem *HTTP/2-Stream 2* zu einem Paketverlust. Obwohl die anderen *Streams 1* und *3* unabhängig vom Transport des zweiten *Streams* sind, wird bei TCP die Auslieferung dieser aufgrund des verloren gegangenen Pakets solange blockiert, bis das verlorene Paket bei einer erneuten Übertragung ankommt. Erst nach Erhalt des besagten Pakets werden alle empfangenen Daten in Reihenfolge zum Client übergeben. QUIC hingegen unterstützt das *Multiplexing* von HTTP/2, indem es die Auslieferung der *Streams 1* und *3* nicht beeinträchtigt. Innerhalb der *Streams* werden die Daten weiterhin in Reihenfolge zum Client ausgeliefert. Die Übertragung kommt somit nicht wie in TCP vollkommen zum Erliegen und erfährt insgesamt eine deutliche Steigerung der Geschwindigkeit [16].

### 3.2 Verbindungsaufbau

Der Verbindungsaufbau stellt eine zentrale Komponente dar, die soweit wie möglich minimiert werden muss. Jede Verbindung muss diese Phase durchlaufen und kann erst danach mit der Übertragung der eigentlichen Daten beginnen. Besonders bei kurzlebigen Verbindungen hat der Verbindungsaufbau einen großen Einfluss auf die Verzögerung. Deshalb wurde bei der Entwicklung von QUIC großer Wert auf die Optimierung dieses Vorgangs gelegt. Die Abbildung 5 stellt den Verbindungsaufbau von TCP mit TLS 1.2 und TLS 1.3 gegenüber QUIC dar, welches bereits von sich aus über eine verschlüsselte Übertragung verfügt. Der obere Teil der Abbildung 5 (a) stellt den ersten Verbindungsaufbau eines Clients zum Server dar. Mit TCP und TLS 1.2 sind drei RTTs nötig, bis die Übertragung der Daten erfolgen kann. Davon entfällt eine RTT

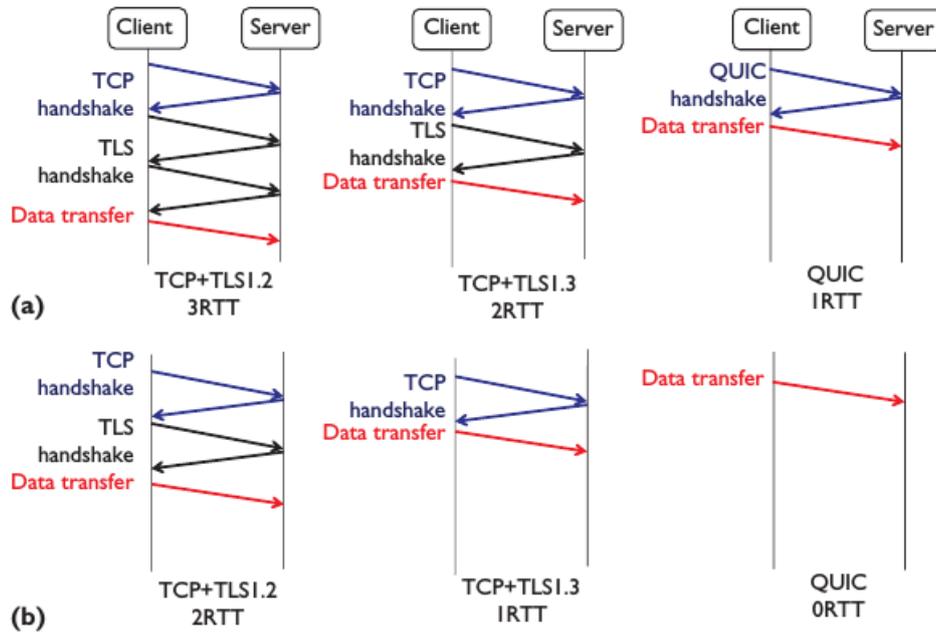


Abbildung 5: Vergleich Verbindungsaufbau zwischen TCP mit TLS und QUIC [16]

für den TCP-Handshake und zwei weitere für die TLS-Verschlüsselung. Die Weiterentwicklung von TLS 1.2 TLS 1.3 kann den initialen Verbindungsaufbau um eine RTT reduzieren, sodass nur noch zwei RTTs erforderlich sind. QUIC führt die Besonderheit mit sich, dass sowohl der Handshake des Transportprotokolls selbst als auch der Verschlüsselungs-Handshake in einer Anfrage kombiniert werden können. Daraus resultiert nur eine RTT bis zum Versand der Daten. Im unteren Teil (b) der Abbildung 5 ist der Verbindungsaufbau zu einem bereits bekannten Server, mit dem schon vorher kommuniziert wurde, zu erkennen. Bei TLS 1.2 und 1.3 kann in diesem Fall jeweils eine RTT eingespart werden. Auch QUIC kann hier eine Verbesserung erzielen und benötigt keine RTT bis zum Versand der Daten. Dies ist dadurch möglich, dass der Client mit den verschlüsselten Daten auch einen kryptographischen Cookie mitschickt, der beim initialen Verbindungsaufbau mit dem Server ausgehandelt wurde. Dieser Cookie authentifiziert den Client gegenüber dem Server und ermöglicht so den sofortigen Transport der Daten [16].

### 3.3 Standardisierungsprozess

Im Jahr 2015 hat sich Google dazu entschieden, mit der Einreichung eines Internet-Drafts [18] ihr Protokoll QUIC bei der IETF standardisieren zu lassen. Seit 2016 bemüht sich eine *Working Group* (QUIC WG) innerhalb der IETF um eine Standardisierung von QUIC [19]. Bis heute wurden zahlreiche Dokumente in Übereinstimmung mit der QUIC WG veröffentlicht, wovon keines bisher über den Status eines Internet-Drafts hinausgekommen ist [20]. Bei Internet-Drafts handelt es sich um Dokumente, die zur informellen Überprüfung von der IETF veröffentlicht wurden. Dies hat den Zweck, dass auf diesem Weg der Öffentlichkeit die Möglichkeit geboten wird, Anteil an der Weiterentwicklung des Internet-Drafts zu nehmen [21]. Eine Änderung oder gar der Austausch durch eine neuere Version des Dokuments kann jederzeit erfolgen. Wenn innerhalb von sechs Monaten keine Änderungen am Dokument durchgeführt worden ist und seitens der IESG (*Internet Engineering Steering Group*), welche für den Prozess der Standardisierung zuständig ist [22], keine Empfehlung für die Veröffentlichung als RFC (*Request for Comments*) abgegeben wurde, so wird das Dokument aus dem Verzeichnis der aktiven Internet-Drafts entfernt [23]. Danach stehen die Internet-Drafts weiterhin unter der Kategorie der zugehörigen Internet-Drafts zur Verfügung.

Aufgrund des großen Interesses vielerorts an QUIC [24] ist stark davon auszugehen, dass das Protokoll in den nächsten Jahren den Status des Standards erreichen wird.

### Relevante Veranstaltungen

In Hinblick auf anstehende Veranstaltungen, die im Zusammenhang mit QUIC eine Relevanz aufweisen, sind folgende zu nennen:

- **ACM SIGCOMM**

Jedes Jahr findet diese Konferenz zu den Themen Kommunikation und Computer-Netzwerke statt [25]. In diesem Jahr kann die SIGCOMM vom 21. - 24. August in Los Angeles besucht werden [26]. Aus dem Konferenz-Programm ist zu entnehmen, dass es einen Vortrag zu QUIC zum Design und zu der Internet-Skalierung geben wird, an dem viele Mitglieder der QUIC WG beteiligt sind. Ebenso ist auch der Initiator von QUIC, Jim Roskind [27], bei diesem Vortrag anwesend.

- **MobiCom**

Die MobiCom ist eine Konferenz, die sich mit *Mobile Computing* und der Vernetzung

beschäftigt [28]. Das diesjährige Programm der MobiCom enthält keinen unmittelbaren Vortrag zum QUIC-Protokoll, jedoch können andere Vorträge aufgrund des verwandten Themengebiets bezogen auf QUIC von Bedeutung sein. Zum jetzigen Zeitpunkt liegt das vollständige Programm noch nicht vor.

- **IETF 99**

Vom 16. - 21. Juli findet in Prag, Tschechien das 99. Meeting der IETF statt [29]. Dieses *Meeting* ist aufgrund des Zusammenkommens der QUIC WG von besonderer Relevanz. Auf der Agenda der *Working Group* [30] steht die Überprüfung und Diskussion von *Implementation Drafts* auf dem Plan. Danach folgen eine Problemdiskussion und Vorträge zu Experimenten von QUIC in Verbindung mit *Multipath* und der Nutzung von *Explicit Congestion Notification* (ECN).

## Ausblick

Bei QUIC handelt es sich um ein junges und sich noch in der Entwicklung befindendes Protokoll. Dies stellt sich als Vorteil heraus, da eine genaue Beobachtung der Entwicklung von Anfang an ermöglicht wird. Durch den Standardisierungsprozess werden einem Einblicke in die Funktionsweise der Standardisierung von Protokollen durch die IETF gewährt und vor Augen geführt, wie hoch der damit verbundene Aufwand ist.

Aufgrund des jungen Bestehens von QUIC existieren noch nicht viele wissenschaftliche Ausarbeitungen oder Analysen, welche eine Untersuchung und Beurteilung der Funktionsweise erlauben. Mithilfe einer Simulation ließe sich eine wissenschaftliche Auseinandersetzung mit dem Protokoll bewerkstelligen, weshalb die Entwicklung einer solchen mit einer anschließenden Analyse des Protokolls im Zentrum der Projektarbeit im Master-Studium steht. Diese Vorgehensweise der Simulation hat den Vorteil, dass eine beliebige Skalierung von Ressourcen erzielt werden kann. Zusätzlich geht mit der Simulation ein deutliches Kosten- und Zeitersparnis einher, welche die Entscheidung bekräftigen. Für die Entwicklung der Simulation eignet sich das Simulations-Framework OMNeT++<sup>1</sup> in Verbindung mit dem INET-Framework<sup>2</sup>, das über Simulationsmodelle verschiedener Netzwerkprotokolle verfügt. Damit wird letztendlich eine Plattform geschaffen, die ein ausgiebiges Testen des Protokolls mit einer anschließenden Auswertung erlaubt und dabei frei für jeden Interessierten zur Verfügung stehen soll.

---

<sup>1</sup><https://omnetpp.org/> (Abgerufen am 05.07.2017)

<sup>2</sup><https://inet.omnetpp.org/> (Abgerufen am 05.07.2017)

## Literatur

- [1] M. Handley, “Why the Internet Only Just Works”, *BT Technology Journal*, Jg. 24, Nr. 3, S. 119–129, Juli 2006, ISSN: 1358-3948. DOI: 10.1007/s10550-006-0084-z. Adresse: <http://dx.doi.org/10.1007/s10550-006-0084-z>.
- [2] V. Fuller und T. Li, “Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan”, RFC Editor, RFC 4632, Aug. 2006. Adresse: <http://www.rfc-editor.org/rfc/rfc4632.txt>.
- [3] M. Allman, V. Paxson und E. Blanton, “TCP Congestion Control”, RFC Editor, RFC 5681, Sep. 2009. Adresse: <http://www.rfc-editor.org/rfc/rfc5681.txt>.
- [4] Internet Systems Consortium. (2017). ISC Internet Domain Survey Host Count, Adresse: <https://www.isc.org/network/survey/> (besucht am 18.04.2017).
- [5] Internet Systems Consortium. (2005). The Internet Domain Survey, Adresse: <https://ftp.isc.org/www/survey/reports/current/survey.html> (besucht am 18.04.2017).
- [6] “IEEE Standard Glossary of Software Engineering Terminology”, *IEEE Std 610.12-1990*, S. 1–84, Dez. 1990. DOI: 10.1109/IEEESTD.1990.101064.
- [7] “Transmission Control Protocol”, RFC Editor, RFC 793, Sep. 1981. DOI: 10.17487/rfc793. Adresse: <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [8] 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 26.05.2017).
- [9] Akamai Technologies. (2017). Akamai’s State Of The Internet Report Q4 2016, Adresse: <https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q4-2016-state-of-the-internet-connectivity-report.pdf> (besucht am 26.05.2017).
- [10] Bundesamt für Sicherheit in der Informationstechnik (BSI). (2016). IT-Grundschutz-Kataloge 15. Ergänzungslieferung, Adresse: [https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge\\_2016\\_EL15\\_DE.pdf](https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf) (besucht am 28.05.2017).
- [11] 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>.
- [12] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach und T. Berners-Lee, “Hypertext Transfer Protocol – HTTP/1.1”, RFC Editor, RFC 2616, Juni 1999. Adresse: <http://www.rfc-editor.org/rfc/rfc2616.txt>.

- [13] I. Grigorik, *High Performance Browser Networking*. O'Reilly Media, Incorporated, 2013, Kap. 11, ISBN: 9781449344764. Adresse: <https://hpbn.co/http1x/>.
- [14] M. Belshe, R. Peon und M. Thomson, "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC Editor, RFC 7540, Mai 2015. Adresse: <http://www.rfc-editor.org/rfc/rfc7540.txt>.
- [15] M. Thomson und J. Iyengar, "QUIC: A UDP-Based Multiplexed and Secure Transport", Internet Engineering Task Force, Internet-Draft draft-ietf-quic-transport-03, Mai 2017, Work in Progress. Adresse: <https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-03>.
- [16] Y. Cui, T. Li, C. Liu, X. Wang und M. Kühlewind, "Innovating Transport with QUIC: Design Approaches and Research Challenges", *IEEE Internet Computing*, Jg. 21, Nr. 2, S. 72–76, März 2017, ISSN: 1089-7801. DOI: 10.1109/MIC.2017.44.
- [17] Chromium Project. (2017). QUIC Implementierung in Chromium, Adresse: <https://chromium.googlesource.com/chromium/src/+master/net/quic/> (besucht am 13.06.2017).
- [18] J. Iyengar und I. Swett, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2", Internet Engineering Task Force, Internet-Draft draft-tsvwg-quic-protocol-00, Work in Progress. Adresse: <https://datatracker.ietf.org/doc/html/draft-tsvwg-quic-protocol-00>.
- [19] Internet Engineering Task Force. (2016). QUIC Working Group, Adresse: <https://datatracker.ietf.org/doc/charter-ietf-quic/> (besucht am 13.06.2017).
- [20] Internet Engineering Task Force. (). QUIC Internet Drafts, Adresse: <https://datatracker.ietf.org/wg/quic/documents/> (besucht am 13.06.2017).
- [21] Internet Engineering Task Force. (). Internet-Drafts (I-Ds), Adresse: <https://www.ietf.org/id-info/> (besucht am 05.07.2017).
- [22] R. Hovey und S. Bradner, "The Organizations Involved in the IETF Standards Process", RFC Editor, RFC 2028, Oktober 1996, Section 3.5 Internet Engineering Steering Group. Adresse: <http://www.rfc-editor.org/rfc/rfc2028.txt>.
- [23] S. Bradner, "The Internet Standards Process – Revision 3", RFC Editor, RFC 2026, Oktober 1996, Section 2.2 Internet-Drafts. Adresse: <http://www.rfc-editor.org/rfc/rfc2026.txt>.
- [24] (2016). QUIC Working Group Blue Sheet - IETF 96, Adresse: <https://www.ietf.org/proceedings/96/bluesheets/bluesheets-96-quic-01.pdf> (besucht am 05.07.2017).
- [25] SIGCOMM. (). About SIGCOMM, Adresse: <http://www.sigcomm.org/about> (besucht am 05.07.2017).

- [26] SIGCOMM. (). SIGCOMM Webseite, Adresse: <http://conferences.sigcomm.org/sigcomm/2017/> (besucht am 05.07.2017).
- [27] J. Roskind. (2012). Quick UDP Internet Connections – Multiplexed Stream Transport over UDP, Adresse: [https://docs.google.com/document/d/1RNHkx\\_VvKWYWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/edit](https://docs.google.com/document/d/1RNHkx_VvKWYWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/edit) (besucht am 05.07.2017).
- [28] MobiCom. (). About MobiCom, Adresse: <https://www.sigmobile.org/mobicom/2017/about.php> (besucht am 05.07.2017).
- [29] Internet Engineering Task Force. (). IETF 99 Prague, Czech Republic, Adresse: <https://www.ietf.org/meeting/99/> (besucht am 05.07.2017).
- [30] QUIC WG. (2017). QUIC Working Group Agenda - IETF 99, Adresse: <https://www.ietf.org/proceedings/99/agenda/agenda-99-quic-00> (besucht am 05.07.2017).