Category:   
Vorstellung des Quic Protokolls von Google  
KWs:   
quic, google quic, quic protokoll, chrome quic, Quick UDP Internet Connections  
Internal links:  
Yes  
Title:  
QUIC – das Internet-Transportprotokoll auf UDP-Basis - 1&1  
Description:  
QUIC heißt die von Google initiierte Lösung für geschwindigkeitsoptimierte Datenverbindungen. Doch was genau steckt hinter dem Protokoll?  
Number of words: Target / Actual  
Min200- Max1000 / 2039  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# QUIC: Das steckt hinter dem experimentellen Google-Protokoll

Dank immer schnellerer DSL-Zugänge ist die **Ladezeit von Internetseiten** im Allgemeinen immer geringer geworden. In der Konsequenz werden schnelle Seitenaufrufe heute als selbstverständlich angesehen, sodass **langsam ladende Webprojekte** kaum noch Chancen haben, auf dem Markt zu bestehen. Erschwerend kommt hinzu, dass auch das Thema Verschlüsselung zunehmend an Bedeutung gewinnt: Der HTTPS-Standard [Internet Security – Sichere Websites mit SSL und HTTPS] (https://hosting.1und1.de/digitalguide/websites/webseiten-erstellen/wie-stelle-ich-meine-seite-auf-ssl-und-https-um) erweist sich zwar als verlässlicher Verbündeter beim **Schutz der Nutzer-Privatsphäre**, hat aber durch TLS-Handshake sowie Zertifikats- und Schlüsselaustausch eine zusätzliche Verzögerung des Ladeprozesses zur Folge – eine Situation, die das von Google initiierte **QUIC-Protokoll** zukünftig lösen soll.

# Was ist QUIC (Quick UDP Internet Connections)?

Bei QUIC handelt es sich um ein experimentelles Transportprotokoll des Suchmaschinenriesen Google, das der Öffentlichkeit erstmals im Jahr 2013 vorgestellt wurde. Der Name des Protokolls steht dabei für „**Q**uick **U**DP **I**nternet **C**onnections“, was darauf zurückzuführen ist, dass es das schnelle und unkomplizierte Versenden einfacher Datenpakete über das **verbindungslose** UDP-Protokoll (**U**ser **D**atagram **P**rotocol) ermöglicht. Hintergrund der Arbeiten an QUIC war der Wunsch, eine Alternative zu der etablierten Sicherheitslösung aus TCP, HTTP/2 [So optimiert HTTP/2 das World Wide Web] (https://hosting.1und1.de/digitalguide/hosting/hosting-technik/so-optimiert-http2-das-world-wide-web/) und TLS/SSL [Wikipedia-Eintrag zu TLS bzw. SSL] (https://de.wikipedia.org/wiki/Transport\_Layer\_Security) zu entwickeln, die den gleichen Schutz bietet, gleichzeitig aber eine **reduzierte Verbindungs- und Transportverzögerung** aufweist und Multiplex-Verbindungen ermöglicht.

Google hat QUIC zu diesem Zweck so entworfen, dass das Protokoll die **Verbindungskontrolle** selbst regelt. Beim ersten Handshake zwischen Sender und Empfänger tauschen diese die Zertifikate und Schlüssel aus, die für die Verschlüsselung der versendeten Datagramme benötigt werden. Bei späteren Kommunikationen entfällt dieser Austausch, was die Latenz minimiert. Als Verschlüsselungsprotokoll kommt dabei die aktuelle, **geschwindigkeitsoptimierte TLS-Version 1.3** (im März 2017 standardisiert) zum Einsatz, die den Vorzug vor der hauseigenen Crypto-Lösung erhielt. In Sachen **Multiplexing** orientiert sich QUIC an dem ebenfalls von Google erarbeiteten SPDY-Protokoll, das die Vorlage zu HTTP/2 gab: Über eine einzige Client-Server-Verbindung können mehrere verschiedene Datenströme übertragen werden, wodurch die Ladezeit zusätzlich reduziert wird.

[Hinweis: Seit 2016 beschäftigt sich eine offizielle Arbeitsgruppe der IETF mit der Optimierung des QUIC-Protokolls. Beinahe 50 Entwickler von Google, Mozilla, Microsoft und anderen Firmen setzen sich unter der Leitung von Lars Eggert und Mark Nottingham für die Weiterentwicklung und Verbreitung der Spezifikation ein. Auf den **Google-Servern** ist das Protokoll bereits seit einigen Jahren (2013) im Einsatz. Zusätzlich wurde QUIC auch in den hauseigenen Browser **Chrome** implementiert, weshalb bereits aktuell ein Teil des Internetverkehrs (z. B. YouTube) über das fortschrittliche Transportprotokoll abgewickelt wird.]

## Welche Vorteile bietet QUIC?

Einige wichtige **Eigenschaften** und **Vorzüge** von QUIC sind bereits zur Sprache gekommen, sollen an dieser Stelle gemeinsam mit weiteren **Verbesserungen** aber noch einmal etwas genauer beleuchtet werden. Als Vergleichsprotokoll dient TCP, das zwar als Wegbereiter im Konzept des aufstrebenden Transportprotokolls eine wichtige Rolle spielt, dem Google-Protokoll aber in einigen Punkten deutlich unterlegen ist, wie folgende Vorteile von QUIC verdeutlichen.

### Schneller Verbindungsaufbau

Der Hauptgrund für das Performance-Plus von QUIC gegenüber TCP ist der wesentlich schnellere Verbindungsaufbau. Selbst ohne Verschlüsselung via SSL/TLS setzt eine Verbindung über das klassische Transportprotokoll mit dem sogenannten Drei-Wege-Handshake mehr Schritte voraus als die **UDP-basierte** Google-Lösung. QUIC startet eine Verbindung **mit einem einzigen Paket** (bzw. zwei Paketen, wenn es sich um die erste Verbindungsaufnahme handelt) und übermittelt dabei sogar alle notwendigen TLS- bzw. HTTPS-Parameter. In den meisten Fällen kann ein Client also **Daten direkt an einen Server senden**, ohne auf eine Antwort von diesem angewiesen zu sein, während TCP zunächst die Bestätigung des Servers einholen und verarbeiten muss.

### Möglichkeit von Multiplex-Verbindungen

TCP greift auf die TCP-Ports und IP-Adressen der verbundenen Systeme zurück, um eine Verbindung zu identifizieren. Aus diesem Grund ist es nicht möglich, dass ein Client in einer einzigen Verbindung über mehrere Ports mit dem Server kommuniziert. Das QUIC-Protokoll löst die Situation auf andere Weise: Es greift auf eine **64-Bit-Verbindungserkennung** und verschiedene „Streams“ zum Transport der Daten innerhalb einer Verbindung zurück. Eine QUIC-Verbindung ist daher nicht an einen bestimmten Port (in diesem Fall UDP-Port), eine IP-Adresse oder einen bestimmten Endpunkt gebunden. **Port- und IP-Wechsel** sind in der Konsequenz ebenso möglich wie die bereits erläuterten **Multiplex-Verbindungen**.

### Vergabe einzigartiger Sequenznummern

Jedes Datensegment einer QUIC-Verbindung erhält eine eigene Sequenznummer, unabhängig davon, ob es sich um ein **originales** oder um ein **weitergeleitetes Segment** handelt. TCP macht diesen Unterschied standardmäßig nicht, weshalb ein Host den Status einer Sequenz auch nicht ermitteln kann – erst durch den Einsatz der Timestamps-Erweiterung (dt. „Zeitstempel“) ermöglicht auch das klassische Transportprotokoll eine solche Unterscheidung. Die fortlaufende Markierung der Pakete ist deshalb von Vorteil, weil sie eine **akkuratere Schätzung der Paketumlaufzeit** (RTT: **R**ound **T**rip **T**ime) ermöglicht.

### Vorwärtsfehlerkorrektur

Verloren gegangene Pakete stellen beim Datentransport über QUIC kein großes Problem dar. Dank eines einfachen XOR-basierten Fehlerkorrektursystems ist keine erneute Übertragung der entsprechenden Daten notwendig. Diese können jederzeit mithilfe von **FEC-Paketen** (**F**orward **E**rror **C**orrection) – Back-ups derOriginalpakete für eine Datengruppe – rekonstruiert werden. Die Fehlerkorrektur funktioniert allerdings nicht, wenn mehrere Pakete einer Datengruppe fehlen.

### Überlastungssteuerung (Packet Pacing)

TCP versucht Daten immer so schnell wie möglich zu versenden, was für schnelle Datenverbindungen zwar von Vorteil, aber auch mit einer gewissen Verlustrate verbunden ist. Geht ein Paket verloren, wird ebenso schnell die **Neuübertragung** (TCP Fast Retransmit) initiiert. Hierfür verkleinert TCP jedoch vorübergehend das Staufenster, was häufig zur Folge hat, dass die Daten stoßweise übertragen werden. Das QUIC-Protokoll wirkt derartigen Lastspitzenmithilfe des sogenannten Packet Pacing entgegen. Das Verfahren sorgt dafür, dass die **Übertragungsrate automatisch begrenzt** wird. So kommt es auch bei Verbindungen mit geringer Bandbreite nicht zu Überlastungen. Neu ist diese Technik jedoch nicht: Einige Linux-Kernel nutzen das Verfahren auch für das TCP-Protokoll.

### Authentifizierung und Verschlüsselung

Der Sicherheitsaspekt stand bei der Planung und Konzipierung von QUIC von Beginn an im Fokus. Im Zuge dessen haben sich die Entwickler auch einem der größten Probleme von TCP gewidmet: Der **Header der verschickten Pakete** liegt im Klartext vor und kann ohne vorherige Authentifizierung gelesen werden. Man-in-the-Middle-Attacken [Man-in-the-Middle-Attack: Angriffsmuster im Überblick] (https://hosting.1und1.de/digitalguide/server/sicherheit/man-in-the-middle-attack-angriffsmuster-im-ueberblick/) und Paketmanipulationen (z. B. der Sequenznummern) sind daher keine Seltenheit. QUIC-Pakete jedoch sind immer authentifiziert und **größtenteils verschlüsselt** (inklusive Payload). Die Teile des Headers, die nicht in verschlüsselter Form vorliegen, sind aufgrund der **Authentifizierung auf Empfängerseite** vor Injektion und Manipulation geschützt.

### Hardware-Unabhängigkeit

Ein weiterer großer Vorteil von QUIC gegenüber TCP: Das Google-Protokoll ist **vom System losgelöst**. Während das TCP-Protokoll von den jeweiligen Plattformen bzw. Geräten unterstützt werden muss, damit eine Kommunikation möglich ist, ist die QUIC-Unterstützung lediglich **auf Anwendungsebene** notwendig. Software-Firmen haben es also prinzipiell selbst in der Hand, das Protokoll einzubinden – sie sind dabei nicht auf die Hardware-Hersteller angewiesen. Bis dato sind es zwar vor allem Google-Anwendungen wie die Google-Server, Chromium oder Chrome, die über **QUIC-Implementierungen** verfügen. Mit dem Browser Opera, der Server-Software Caddy und den Load-Balancing- und Webserver-Produkten von LiteSpeed Technologies gibt es aber bereits Drittanbieter-Applikationen, die Verbindungen über das neue Transportprotokoll ermöglichen.

## Die Nachteile des QUIC-Protokolls

Dass QUIC künftig wohl immer häufiger zum Einsatz kommen dürfte, ist vor allem auf das Engagement der IETF zurückzuführen. Dank der **Anpassungen an allgemeine Standards** seit Gründung der Arbeitsgruppe im Jahr 2016 hat sich das Protokoll von einem stark auf Google ausgerichteten zu einem allgemeinen Netzwerkprotokoll entwickelt, das zunehmend an Bedeutung gewinnt. Der **Optimierungsprozess** ist allerdings **noch lange nicht abgeschlossen**: Das QUIC-Team befasst sich auch weiterhin mit bestehenden Problemen, für die es die passenden Lösungen zu finden gilt.

Insbesondere das Thema **Sicherheit**, eines der wichtigsten bei der Entwicklung des Protokolls, sorgt dabei für Diskussionsstoff. Denn während Authentifizierung und Verschlüsselung zweifelsohne für einen sichereren Datentransport sorgen, sind sie gleichzeitig auch für einen entscheidenden Nachteil von QUIC verantwortlich: Da die Paket-Header weniger Klartextinformationen enthalten als es bei TCP-Verbindungen der Fall ist, sind Aufgaben wie **Fehlerbehebung**, Traffic-Regulierung oder Netzwerk-Management bei QUIC-Verbindungen **deutlich erschwert**. Netzbetreiber sowie Hersteller von Firewalls und anderen Mittelboxen sehen deshalb Schwierigkeiten darin, die Qualität ihrer Dienste zu gewährleisten.

Ein weiteres Problem des QUIC-Protokolls ist, dass die automatische Überlastungssteuerung bei Datenverbindungen mit hoher Bandbreite in einigen Fällen zu einer **schlechteren Übertragungsrate** führen kann.

## QUIC aktivieren bzw. deaktivieren – so funktioniert’s

Auch wenn die Entwicklung des QUIC-Protokolls insbesondere in den letzten Jahren erheblich vorangetrieben wurde, kommt es bis dato ausschließlich **experimentell** in den Browsern Google Chrome [Google Chrome: Downloadseite] (https://www.google.de/chrome/browser/desktop/index.html) und Opera [Homepage des Webbrowsers Opera] (http://www.opera.com/de/computer) zum Einsatz. In Ersterem ist es standardmäßig aktiviert, in Letzterem ist es standardmäßig deaktiviert. Opera-Nutzer müssen das Protokoll also zunächst manuell freischalten, um schon jetzt von dem potenziellen **Performance-Boost** profitieren zu können. Wie genau Aktivierung und Deaktivierung von QUIC in den beiden Web-Clients funktionieren, erfahren Sie in den folgenden Abschnitten.

### So konfigurieren Sie QUIC in Chrome

Um die Einstellungen des QUIC-Protokolls in Googles Browser Chrome zu ändern, ist es notwendig, das Konfigurationsmenü der experimentellen Features aufzurufen. Hierfür geben Sie einfach folgenden Befehl in die **Adresszeile** ein:

|  |
| --- |
| chrome://flags |

Suchen Sie dort nach dem Menüpunkt „**Experimental QUIC protocol**“, beispielsweise mithilfe der Suchfunktion, die Sie über die Tastenkombination [STRG] + [F] starten können. Haben Sie bisher an den **Grundeinstellungen** keinerlei Änderungen vorgenommen, sollte für das Protokoll die Option „**Default**“ (dt. Standard) ausgewählt sein. Im Fall von QUIC steht diese Chrome-Standardkonfiguration dafür, dass das Protokoll aktiviert ist.

C:\Users\jkrueger\Desktop\QUIC 1.JPG

[Alt-Text: QUIC-Protokoll-Konfiguration in Chrome] [Bildunterschrift: Da Chrome plattformübergreifend nutzbar ist, ist auch die Unterstützung von QUIC auf den verschiedenen Systemen wie macOS, Windows, Linux oder Android gewährleistet.]

Wollen Sie das Protokoll deaktivieren, wählen Sie einfach den Eintrag „**Disabled**“ aus und klicken anschließend auf die Schaltfläche „JETZT NEU STARTEN“. Chrome wird in der Folge beendet – beim nächsten Start des Browsers sind die neuen Einstellungen aktiv. Wollen Sie das Protokoll wieder aktivieren, gehen Sie auf die gleiche Weise vor, wählen jedoch entweder die eigentliche Einstellung „**Default**“ oder alternativ den Punkt „**Enabled**“ aus.

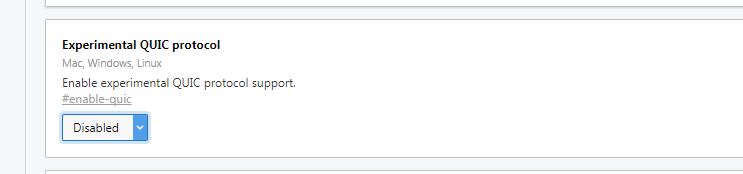
[Tipp: Chrome bietet die Möglichkeit, **aktive QUIC-Sitzungen einzusehen**. Hierfür muss lediglich der Befehl *chrome://net-internals/#quic* in die Adressleiste eingefügt werden.]

### So schalten Sie QUIC in Opera ein bzw. aus

Opera, der auf Chromium basiert, hat bereits seit Version 16, die im August 2013 veröffentlicht wurde, eine experimentelle Fassung des QUIC-Protokolls integriert. Der Unterschied zu Google Chrome besteht darin, dass das Protokollin Opera **standardmäßig deaktiviert** ist. Um die neue Datentransport-Technologie in Opera zu nutzen, müssen Sie also selbst aktiv werden. Die passende Option finden Sie – ähnlich wie im Google-Browser – im Konfigurationsmenü für die **experimentellen Features**. Dieses heißt in Opera „Experimente“ und lässt sich mit folgendem Befehl über die Adressleiste aufrufen:

|  |
| --- |
| opera://flags |

In der Auflistung der verschiedenen Test-Features finden Sie das Optionsmenü für das Protokoll unter dem Eintrag „**Experimental QUIC protocol**“. Um QUIC zu aktivieren, ändern Sie den Status einfach auf „**Enabled**“ und wählen im Anschluss die Schaltfläche „Jetzt neu starten“. Wollen Sie die Einstellung zu einem späteren Zeitpunkt rückgängig machen, können Sie dies an gleicher Stelle durch die Auswahl „Disabled“ tun.



[Alt-Text: Opera-„Experimente“-Menü: Experimental QUIC protocol] [Bildunterschrift: Die Entwickler von Opera weisen im „Experimente“-Bereich explizit darauf hin, dass die Aktivierung bzw. Deaktivierung von Features zu Komplikationen führen kann.]

[Tipp: Auch in Opera können Sie sich **aktive Datenverbindungen**, die über QUIC laufen, anzeigen lassen. Hierfür fügen Sie nach der Aktivierung des Protokolls den Befehl *opera://net-internals/#quic* in die Browserzeile ein.]

## Welche Websites nutzen das QUIC-Protokoll bereits?

Als Initiator von QUIC hat Google das Protokoll bereits 2013 in seine Server integriert, weshalb die verschiedenen **Google-Dienste** zu den bekanntesten Webanwendungen zählen, die den Datentransport über das womöglich zukunftsweisende Protokoll erlauben. Allen voran ist hierbei natürlich die im Zentrum des Unternehmens stehende **Suchmaschine** zu nennen. Doch auch die **anderen Webservices** wie der Kartendienst Google Maps, das Netzwerk Google+, der E-Mail-Dienst Google Mail, die Office-Lösung Google Docs oder das Video-Portal YouTube können über das QUIC-Protokoll ausgeliefert werden, sofern der Benutzer den passenden Client verwendet.

Welche anderen Webangebote QUIC unterstützen, können Chrome-User mithilfe der Extension HTTP/2 and SPDY indicator [Chrome Web Store: http/2 and SPDY indicator] (https://chrome.google.com/webstore/detail/http2-and-spdy-indicator/mpbpobfflnpcgagjijhmgnchggcjblin) überprüfen. Die Erweiterung fügt neben der Adressleiste des Browsers ein kleines **Blitz-Symbol** hinzu, das sich grün färbt, wenn die aufgerufene Seite das Transportprotokoll unterstützt. Navigiert man mit der Maus über das Symbol, verrät ein Tooltip zudem die **Versionsnummer**.

C:\Users\jkrueger\Desktop\QUIC 3.JPG

[Alt-Text: Google Chrome: Browserleiste] [Bildunterschrift: HTTP/2 and SPDY indicator ist in erster Linie entwickelt worden, um Websites auf ihre Unterstützung von HTTP/2 (blaue Färbung) und dessen Vorgänger SPDY (grün wie bei QUIC) zu überprüfen.]

[Teaser]

Quick UDP Internet Connections, kurz QUIC, ist ein experimentelles Netzwerkprotokoll aus dem Hause Google. Mittlerweile wird die Weiterentwicklung von der IETF vorangetrieben, die zu diesem Zweck eine spezielle Arbeitsgruppe gegründet hat. Das QUIC-Protokoll soll eine geschwindigkeitsoptimierte Alternative zum derzeitigen Transport-Standard TCP bieten und baut daher auf dem verbindunglosen Protokoll UDP auf – nicht die einzige Neuerung des auf der Anwendungsschicht agierenden Übertragungsstandards.