



(10) **DE 10 2015 119 893 B4** 2021.11.18

(12)

Patentschrift

(21) Aktenzeichen: **10 2015 119 893.2**

(22) Anmeldetag: **17.11.2015**

(43) Offenlegungstag: **30.06.2016**

(45) Veröffentlichungstag
der Patenterteilung: **18.11.2021**

(51) Int Cl.: **H04L 5/06 (2006.01)**

H04L 29/06 (2006.01)

H04L 12/951 (2013.01)

H04L 12/70 (2013.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
14/583,337 26.12.2014 US

(73) Patentinhaber:
Intel Corporation, Santa Clara, Calif., US

(74) Vertreter:
**Maucher Jenkins Patentanwälte & Rechtsanwälte,
80538 München, DE**

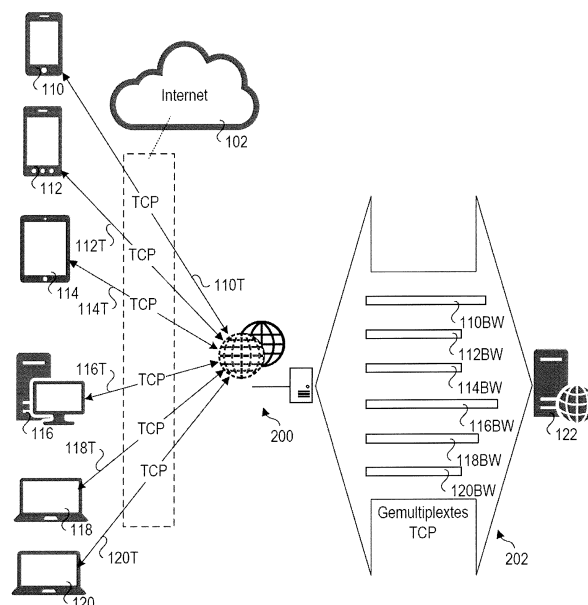
(72) Erfinder:
Perlman, Radia, Redmond, Wash., US

(56) Ermittelter Stand der Technik:

US	2006 / 0 085 559	A1
US	2012 / 0 054 316	A1
CN	103 023 987	A

(54) Bezeichnung: **Multiplexen von vielen Client-Datenströmen über eine einzige Verbindung**

(57) Hauptanspruch: Verfahren, umfassend:
das Aufbauen einer Vielzahl von TCP-Verbindungen mit einer Vielzahl von Clients an einem Proxyserver, wobei jede TCP-Verbindung unter Verwendung eines TCP-Ports einen Client mit dem Proxyserver verbindet;
das Koppeln des Proxyservers in Kommunikation mit einem Webserver über eine gemultiplexte TCP-Verbindung;
das Empfangen von eingehenden TCP-Kommunikationseinheiten an dem Proxyserver, die von der Vielzahl von Clients stammen, wobei jede eingehende TCP-Kommunikation zumindest ein TCP-Paket umfasst, das Paket-Nutzlastdaten umfasst;
das Puffern der Paket-Nutzlastdaten, die in TCP-Paketen enthalten sind, die über eine gegebene TCP-Verbindung in einem entsprechenden Client-Daten-Datenstrom empfangen werden;
das Einkapseln von Datensegmenten, die in den Client-Daten-Datenströmen sequentielle Abfolgen von Bits umfassen, in gemultiplexte (MUX-) TCP-Pakete an dem Proxyserver, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente von zumindest zwei getrennten Client-Daten-Datenströmen einkapselt;
das Übertragen der MUX-TCP-Pakete von dem Proxyserver über die gemultiplexte TCP-Verbindung an den Webserver; und
an dem Webserver
das Entkapseln von Datensegmenten in den MUX-TCP-Paketen und das Neuordnen der Client-Daten-Datenströme; und
das Extrahieren von Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen.



Beschreibung

HINTERGRUNDINFORMATION

[0001] Seit der Einführung des Mikroprozessors wurden Computersysteme immer schneller. In ungefähre Übereinstimmung mit dem Moore'schen Gesetz (das auf der Publikation des Mitgründers der Intel® Corporation Gordon Moore aus dem Jahr 1965 basiert, die vorhersagt, dass sich die Anzahl der Transistoren auf integrierten Schaltkreisen alle zwei Jahre verdoppeln wird), schoss die Geschwindigkeitserhöhung seit fast drei Jahrzehnten mit einer ziemlich gleichmäßigen Rate in die Höhe. Gleichzeitig erhöhte sich auch die Größe von sowohl der Speicherkapazität als auch der Permanentspeicherkapazität ständig, sodass viele der heutigen persönlichen Computer leistungsfähiger sind als Supercomputer von vor nur 10-15 Jahren. Zusätzlich dazu erfuhr auch die Geschwindigkeit der Netzwerkkommunikation eine astronomische Erhöhung.

[0002] Erhöhungen der Prozessorgeschwindigkeiten, Speicherkapazität, Permanentspeicherkapazität und Netzwerkbandbreitentechnologien führten zu dem Ausbau und dem Einsatz von Netzwerken mit ständig steigenden Kapazitäten. Kürzlich führte die Einführung von Cloud-basierten Diensten, wie jene, die von Amazon (z.B. Amazon Elastic Compute Cloud (EC2) und Simple Storage Service (S3)) und Microsoft (z.B. Azure und Office 365) bereitgestellt werden, zu einem zusätzlichen Netzerk Ausbau der öffentlichen Netzwerkinfrastruktur, und zusätzlich dazu zu dem Einsatz von massiven Datenzentren, um diese Dienste zu unterstützen, die die private Netzwerkinfrastruktur verwenden. Zusätzlich dazu wird erwartet, dass die neue Generation (d.h. 4G) der mobilen Netzwerkdatendienste in naher Zukunft einen signifikanten Einfluss auf die Verwendung von Festnetz-Netzwerken haben wird. Das Ergebnis dieser und anderer Überlegungen ist, dass erwartet wird, dass die Verwendung von Computernetzwerken in naher Zukunft weiterhin mit einer hohen Rate wachsen wird.

[0003] Ein großer Teil des Internetverkehrs verwendet das HyperText-Übertragungsprotokoll (HTTP) oder HTTPS (eine verschlüsselte und gesicherte Version von HTTP). Im Wesentlichen umfasst HTTP eine Kommunikation zwischen einem Client (z.B. ein Webbrowser auf einem Gerät eines Benutzers) und einen HTTP-Server (z.B. eine Webserver-Anwendung, die auf einem Server läuft und einen Zugriff auf eine Webseite hostet und/oder bereitstellt; wie hierin verwendet, können die Begriffe „Web“-Server und HTTP-Server austauschbar verwendet werden). HTTP arbeitet in dem Client-Server-Computing-Modell als ein Anfrage-Antwort-Protokoll. Unter einer typischen HTTP-Anfrage-Antwort-Sequenz übermittelt ein Client eine HTTP-Anfragenachricht an den Ser-

ver. Der Server, der Ressourcen wie HTML-Dateien und anderen Inhalt bereitstellt oder für den Client andere Funktionen ausführt, gibt dem Client eine Antwortnachricht zurück.

[0004] Fig. 1 zeigt eine Internetarchitektur **100**, die ausgewählte Internetkomponenten, einschließlich Netzwerkelemente, Client-Vorrichtungen, Server und Dienste, veranschaulicht. Das Internet, das üblicherweise als eine Internet-Wolke **102** dargestellt wird, ist eine massive Sammlung von zwischenverbundenen Netzwerkelementen, einschließlich Gateways **104** und Switches **106**, die durch Inter-Rechner-Verbindungen **108** zwischenverbunden sind. Client-Vorrichtungen („Clients“, anerkennend, dass sich ein Client auf die Vorrichtung selbst oder auf eine Anwendung, die auf der Vorrichtung läuft, beziehen kann), dargestellt durch Mobiltelefone (Clients **110** und **112**), einen Tablet-Computer (Client **114**), einen Desktop-Computer (Client **116**), und Laptop-Computer (Clients **118** und **120**), sind typischerweise über einen Gateway **104**, der durch einen Internetdienstanbieter (ISP) oder einen mobilen Dienstanbieter (wenn über ein mobiles Netzwerk auf das Internet zugegriffen wird) betrieben wird, mit dem Internet **102** verbunden. Zur Einfachheit, und um Unordnung zu vermeiden, sind die Verbindungen zwischen den Client-Vorrichtungen und einem Gateway als ein Pfeil mit zwei Spitzen gezeigt; in der Praxis bestehen zwischen den Client-Vorrichtungen und den Gateways verschiedene Netzwerkelemente und die einem Gateway zugeschriebene Funktionalität kann auf einem oder mehreren Netzwerkelementen implementiert sein.

[0005] Verschiedene Server sind auch mit dem Internet **102** verbunden, wie durch einen Webserver **122** und einen Drittparteien-E-Mail-Server **124** abgebildet. Der Webserver **122** stellt den „Front-End“-Zugriff auf eine Webseite **126** bereit. Größere Webseiten wie Suchseiten, Nachrichtenseiten und E-Commerce-Seiten können dutzende, hunderte oder sogar tausende Webserver aufweisen, um den Verkehr der Webseite abzuwickeln. Der Drittparteien-E-Mail-Server **124** ist eine konzeptuelle Einrichtung, die in der Praxis über verschiedene Webserver in einer Front-End-Schicht implementiert sein kann, die in einer oder mehreren „Mittel“-Schichten mit E-Mail-Anwendungsservern oder dergleichen gekoppelt sind, die wiederum in einer „Back-End“-Schicht mit Datenspeichern verbunden sind. Große Drittparteien-E-Mail-Dienste wie Microsoft Hotmail, Google Gmail und Yahoo Mail sind unter Verwendung großer Datenzentren (auch als Server-Farmen bezeichnet) implementiert, wie durch ein Datenzentrum **128** abgebildet.

[0006] Um eine HTTP/HTTPS-Kommunikation zwischen einem Client und einem Server zu ermöglichen, wird zunächst eine TCP-(Übertragungssteuerungsprotokoll-) Verbindung aufgebaut. TCP ist eine

der wichtigsten Protokolle der Internetprotokoll- (IP-) Familie, und eine TCP-Verbindung wird manchmal auch als TCP/IP-Verbindung bezeichnet. TCP verwendet die IP-Adressen eines Paares von TCP-Endpunkten (z.B. ein Client und ein Server), das jeweils die dem Client und dem Server zugeordneten IP-Adressen umfasst, die über die TCP-Verbindung zur Kommunikation verbunden sind.

[0007] Wenngleich TCP ein verlässliches Übertragungsprotokoll bereitstellt, definiert es nicht, wie TCP/IP-Pakete zwischen einem Client und einem Server geleitet werden sollen. Beispielsweise sind in **Fig. 1** zwei Multi-Hop-Pfade (einer als durchgehende Linie, einer als unterbrochene Linie) zwischen dem Gateway, der durch den Client **110** verbunden ist, und dem Gateway, der mit dem Webserver **122** verbunden ist, veranschaulicht. Wenngleich die Paket-„Flüsse“ mancher Verkehrsklassen über einen vorgegebenen Pfad weitergeleitet werden können, wird der Großteil des Internetverkehrs auf einer Best-Effort-Basis von Hop zu Hop weitergeleitet, wobei der Weiterleitungspfad für ein Paket nicht vorbestimmt ist.

[0008] Ein gegebener HTTP-Server kann 1000e gleichzeitige TCP-Verbindungen unterstützen, jede davon mit einem unterschiedlichen TCP-Endpunkt. Dies erzeugt eine große Menge von Verwaltungsdaten wie Timeout-Zeitgeber und getrennte Byte-Folgenummern-Räume für jede Client-Verbindung. Zusätzlich dazu hat jeder HTTP-Server eine endliche Bandbreitenbeschränkung, die dazu führen kann, dass an dem HTTP-Server Pakete fallengelassen werden. Wenngleich TCP eine Stausteuering unterstützt, sind seine Stausteueringmechanismen auf den individuellen TCP-Verbindungen implementiert und werden nicht kollektiv als eine Funktion der Auslastung des Servers verwaltet.

[0009] In US 2006 / 0 085 559 A1 ist ein System und Verfahren zur Weiterleitung von Client-Anfragen von einem Proxy-Server in einer TCP/IP-Computerumgebung beschrieben. Eine Vielzahl von vorübergehenden Anforderungen werden von einzelnen sendenden Clients in einer Anforderungswarteschlange empfangen. Jede Anforderung ist in der Regel an einen Ursprungsserver adressiert. Zeitschätzungen des TCP-Overheads, des Slow-Start-Overheads, der Time-to-Idle und der Anfrageübertragungszeit für das Senden der Anfragen über jede einer Vielzahl von verwalteten Verbindungen zum Ursprungsserver werden dynamisch berechnet, gleichzeitig mit dem Empfang und während der Verarbeitung jeder Anfrage. Die verwaltete Verbindung wird in der Reihenfolge der bevorzugten Auswahl ausgewählt aus einer warmen Leerlaufverbindung, einer aktiven Verbindung mit einer Zeit bis zur Leerlaufzeit, die kleiner ist als ein langsamer Start-Overhead, einer kalten Leerlaufverbindung, einer aktiven Verbindung mit einer Zeit bis zur Leerlaufzeit, die kleiner ist als ein

TCP-Overhead, einer neuen verwalteten Verbindung und einer bestehenden verwalteten Verbindung mit der kleinsten Zeit bis zur Leerlaufzeit. Jede Anfrage wird über die ausgewählte verwaltete Verbindung an den Ursprungsserver weitergeleitet.

[0010] In US 2012/0 054 316 A1 ist ein Proxy beschrieben, der mehrere TCP-Anfragen von Clients in einer TCP-Verbindung zwischen einem Proxy und einem Server zusammenfasst. Eine dauerhafte TCP-Verbindung wird zwischen dem Proxy und dem Server geöffnet, und eine Verbindungsanfrage wird von einem Client beim Proxy empfangen. Für den Client wird ein eindeutiger Bezeichner registriert, und dem Server wird über die dauerhafte Verbindung signalisiert, den eindeutigen Bezeichner zu registrieren. Als Reaktion auf die Verbindungsanforderung wird eine Client-Verbindung zwischen dem Client und dem Proxy aufgebaut. Eine Inhaltsanforderung wird vom Client über die Client-Verbindung empfangen. Der eindeutige Bezeichner wird der Inhaltsanforderung vorangestellt, die vorangestellte Inhaltsanforderung wird an den Server weitergeleitet, und der vorangestellte Inhalt wird vom Server empfangen. Die Kennung und der Inhalt werden aus dem vorangestellten Inhalt extrahiert, und ein Client und eine Client-Verbindung werden identifiziert, die der extrahierten Kennung entsprechen. Der extrahierte Inhalt wird an den entsprechenden Client zurückgegeben.

[0011] Aus CN 103023987 A ist ein Multiplexverfahren bekannt, das auf einer TCP-Verbindung (Transmission Control Protocol) basiert. Das Verfahren wird durch Multiplexing eines Kanals zwischen einem Proxy-Server und einem Server durch eine Vielzahl von Clients implementiert. Ein Client versucht, mit dem Server zu kommunizieren, eine SOCKET-Verbindungsanforderung muss zwischen dem Client und dem Proxy-Server hergestellt werden, ein TCP-Verbindungspool-Verwaltungsprogramm wird im Proxy-Server installiert, eine eindeutige ID wird an den Client verteilt, nachdem die Verbindungsanforderung legal ist, die ID und die SOCKET werden als Schlüssel-Wert-Paar verwendet und in einer Weiterleitungstabelle gespeichert, der Kommunikationsanforderung des Clients wird zugestimmt, die verteilte ID wird zurück übertragen, und entsprechend wird ein virtueller Kanal für bidirektionale Kommunikation zwischen dem Client und dem Server eingerichtet.

Figurenliste

[0012] Die vorangegangenen Aspekte und viele der begleitenden Vorteile dieser Erfindung werden klarer anerkannt werden, wenn diese mit Verweis auf die folgende ausführliche Beschreibung besser verstanden wird, wenn diese in Verbindung mit den begleitenden Zeichnungen herangezogen wird, worin, solange nichts Gegenteiliges angegeben ist, gleiche

Referenzzahlen in den verschiedenen Ansichten auf gleiche Teile verweisen:

Fig. 1 ist ein schematisches Diagramm, das eine Architekturansicht ausgewählter Komponenten des Internets veranschaulicht;

Fig. 2a ist ein schematisches Diagramm, das eine beispielhafte Netzwerkkonfiguration veranschaulicht, die eine gemultiplexte TCP-Verbindung zwischen einem Proxyserver und einem Webserver einsetzt;

Fig. 2b ist ein Inter-Rechner-Verbindungsdiagramm, das ferner Details individueller TCP-Verbindungen und der gemultiplexten TCP-Verbindung aus **Fig. 2a** veranschaulicht;

Fig. 2c ist ein Inter-Rechner-Verbindungsdiagramm, das eine optionale Konfiguration von **Fig. 2b** veranschaulicht, bei der ein Proxyserver über mehrere gemultiplexte TCP-Verbindungen mit einem Webserver verbunden ist;

Fig. 2d ist ein Inter-Rechner-Verbindungsdiagramm, das eine optionale Konfiguration von **Fig. 2b** veranschaulicht, bei der ein Proxyserver als ein Proxy für mehrere Webserver verwendet wird, mit einer getrennten gemultiplexten TCP-Verbindung zwischen dem Proxy und jedem Webserver;

Fig. 3a ist ein schematisches Diagramm, das TCP-Paket-Segmentdaten-Weiterleitungskomponenten gemäß einer Ausführungsform veranschaulicht, die durch einen Proxyserver in Verbindung mit dem Weiterleiten von von Clients empfangenen TCP-Paketen über eine gemultiplexte TCP-Verbindung, die Datensegmente überträgt, an einen Webserver, implementiert sind;

Fig. 3b ist ein schematisches Diagramm, das serverseitige Komponenten gemäß einer Ausführungsform veranschaulicht, die durch einen Webserver in Verbindung mit dem Verarbeiten von TCP -Paketdatensegmenten, die über eine gemultiplexte TCP-Verbindung von einem Proxyserver empfangen werden, implementiert sind;

Fig. 4a ist ein schematisches Diagramm, das Komponenten zum Verarbeiten von gemultiplexten TCP-Paketdatensegmenten gemäß einer Ausführungsform veranschaulicht, die von dem Webserver gesendet werden und über eine gemultiplexte TCP-Verbindung an einem Proxyserver empfangen werden;

Fig. 4a' ist ein schematisches Diagramm, das eine optionale Konfiguration zum Verarbeiten von gemultiplexten TCP-Paketdatensegmenten gemäß einer Ausführungsform veranschaulicht, die über eine gemultiplexte TCP-Verbindung an einem Proxyserver empfangen werden, unter

dem neu angeordnete TCP-Pakete in einen Paket-Ausgangspuffer geladen werden;

Fig. 4b ist ein schematisches Diagramm, das eine Paketverarbeitung in Verbindung mit dem Senden von TCP-Paketen von einem Webserver über eine gemultiplexte TCP-Verbindung als gemultiplexte Datensegmente an einen Proxyserver gemäß einer Ausführungsform veranschaulicht;

Fig. 5 ist ein Blockdiagramm, das eine Ausführungsform eines gemultiplexten TCP-Pakets veranschaulicht;

Fig. 6 ist ein Nachrichtenflussdiagramm, das einen beispielhaften Satz von Nachrichten veranschaulicht, die zwischen einem Client, einem Proxyserver und einem Webserver ausgetauscht werden;

Fig. 7 ist ein Nachrichtenflussdiagramm, das einen beispielhaften Satz von Befehlen und Steuerungsnachrichten veranschaulicht, die zwischen einem Client, einem Proxyserver und einem Webserver ausgetauscht werden;

Fig. 8a ist eine isometrische Frontansicht eines beispielhaften Bladeserver-Gehäuses, in dem eine Vielzahl von Serverblades eingebaut ist;

Fig. 8b ist eine isometrische Rückansicht des Bladeserver-Gehäuses aus **Fig. 8a**;

Fig. 8c ist eine isometrische Frontansicht eines beispielhaften Bladeserver-Gestells, in dem eine Vielzahl von Bladeserver-Einschubgehäusen, die **Fig. 8a** und **Fig. 8b** entsprechen, eingebaut ist; und

Fig. 9 zeigt gemäß einer Ausführungsform die Details der Komponenten eines typischen Serverblades.

AUSFÜHRLICHE BESCHREIBUNG

[0013] Hierin sind Ausführungsformen von Verfahren und Vorrichtungen zum Multiplexen von vielen Client-Datenströmen über eine einzige Verbindung beschrieben. In der folgenden Beschreibung werden zahlreiche spezifische Details dargelegt, um ein tiefgehendes Verständnis der Ausführungsformen der Erfindung bereitzustellen. Ein Fachmann auf dem betreffenden Gebiet wird jedoch erkennen, dass die Erfindung ohne eines oder mehrere der spezifischen Details, oder mit anderen Verfahren, Komponenten, Materialien etc. praktiziert werden kann. In anderen Fällen sind wohlbekannte Strukturen, Materialien oder Vorgänge nicht ausführlich gezeigt oder beschrieben, um ein Verbergen von Aspekten der Erfindung zu vermeiden.

[0014] In der gesamten vorliegenden Patentschrift bedeutet ein Verweis auf „eine Ausführungsform“,

dass ein/eine bestimmte(s), in Verbindung mit der Ausführungsform beschriebene(s) Merkmal, Struktur oder Eigenschaft in zumindest einer Ausführungsform der vorliegenden Erfindung umfasst ist. Daher bezieht sich das Auftreten der Phrase „in einer Ausführungsform“ an verschiedenen Stellen in der gesamten vorliegenden Patentschrift nicht notwendigerweise auf dieselbe Ausführungsform. Ferner können die bestimmten Merkmale, Strukturen oder Eigenschaften in einer oder mehreren Ausführungsformen in einer beliebigen geeigneten Weise kombiniert werden.

[0015] Zwecks Klarheit kann auf einzelne Komponenten in den Figuren hierin auch durch ihre Bezeichnungen in den Figuren verwiesen werden, anstatt durch eine bestimmte Referenzzahl. Zusätzlich dazu können Referenzzahlen, die sich auf eine bestimmte Art von Komponente beziehen (im Gegensatz zu einer bestimmten Komponente) durch eine Referenzzahl gefolgt von „(typ)“ gezeigt werden, was „typisch“ bedeutet. Es wird verständlich sein, dass die Konfiguration dieser Komponenten für ähnliche Komponenten typisch sein wird, die existieren können, aber in den Zeichnungsfiguren zwecks Einfachheit und Klarheit nicht gezeigt sind, oder anders ähnliche Komponenten, die nicht durch getrennte Referenzzahlen gekennzeichnet sind. Umgekehrt soll „(typ)“ nicht so ausgelegt werden, dass die Komponente, das Element etc. typischerweise für seine(n) offenbarte Funktion, Implementierung, Zweck etc. verwendet wird.

[0016] In Übereinstimmung mit Aspekten der hierin beschriebenen und veranschaulichten Ausführungsform werden Verfahren zum Implementieren von gemultiplexten TCP-Verbindungen, unter denen Kommunikationseinheiten von mehreren Clients über eine einzige TCP-Verbindung übertragen werden, offenbart. Das verringert die Verwaltungsdaten, die zum Erhalten von individuellen TCP-Verbindungen für jeden Client benötigt werden, deutlich. Zudem sind auch neue Flusssteuerungsverfahren bereitgestellt, die es gestatten, dass individuelle Flüsse, die dem Client-TCP-Verkehr innerhalb der gemultiplexten TCP-Verbindung entsprechen, abgewickelt werden, als ob eine individuelle TCP-Verbindung zwischen dem Client und dem Server verwendet würde.

[0017] Ein Aspekt der gemultiplexten TCP-Implementierung ist die Verwendung eines Proxyserver, der unter Verwendung individueller TCP-Verbindungen direkt mit den Clients kommuniziert. Zusätzlich dazu, dass es die Client-Server-Kommunikation unterstützt, ist HTTP darauf ausgelegt, Zwischennetzwerkelemente zu gestatten, um eine Kommunikation zwischen Clients und Servern zu verbessern oder zu ermöglichen. Vielbesuchte Webseiten profitieren oft von Webcache-Servern, die für Upstream-Server Inhalte liefern, um eine Reaktionszeit zu verbessern.

Zusätzlich dazu können HTTP-Proxyserver an privaten Netzwerkgrenzen eine Kommunikation für Clients ohne eine weltweit routbare Adresse ermöglichen, indem Nachrichten mit externen Servern übertragen werden. Proxyserver werden auch oft von großen Webseiten verwendet, wie jene, die durch Google und Amazon gehostet werden.

[0018] Fig. 2a bzw. Fig. 2b zeigt eine Netzwerkarchitektur und ein Inter-Rechner-Verbindungsdiagramm, die verwendet werden, um Aspekte einer beispielhaften gemultiplexten TCP-Verbindungsimplementierung zu veranschaulichen, unter der Clients über individuelle TCP-Verbindungen mit einem Proxyserver **200** kommunizieren, der konfiguriert ist, um einen entsprechenden HTTP-Verkehr über eine gemultiplexte Verbindung **202** zwischen dem Proxyserver **200** und einem Webserver **122** weiterzuleiten. Wie zuvor sind Clients **110**, **112**, **114**, **116**, **118** und **120** über ISP oder mobile Diensteanbieter Einrichtungen wie Internet-Gateways oder dergleichen mit einem Internet **102** verbunden. Jeder dieser Clients möchte eine HTTP-Verbindung mit dem Webserver **122** herstellen. Dies würde typischerweise durch Verwenden der Client-Vorrichtung erreicht werden, die den URL (Uniform Resource Locator) der Website eingeben, der der Webserver **122** zugeordnet ist, z.B. www Punkt Amazon Punkt com. Da Clients und Server im Internet IP-Adressen (z.B. IPv4 oder IPv6) verwenden, müssen die URLs in eine entsprechende IP-Adresse übersetzt werden. Die Zuordnung von URLs und IP-Adressen wird durch einen Domain Name Server (DNS) ermöglicht. Bei großen Internet-Auftritten wie Amazon und Google wäre es unmöglich, einen einzelnen Server mit einer einzelnen IP-Adresse zu verwenden, um den Client-Verkehr abzuwickeln. In manchen Fällen werden Proxyserver verwendet, um dieses Problem zu lösen. In einem typischen Proxyserver Schema gibt die Netzwerkadresse, die durch die DNS-Suche bereitgestellt ist, anstelle der IP-Adresse für den Webserver, der nominal mit dem URL verbunden ist, stattdessen die Adresse des Proxyserver aus.

[0019] Wie in Fig. 2b gezeigt, ist jeder der Clients **110**, **112**, **114**, **116**, **118** und **120** über eine entsprechende TCP-Verbindung **110T**, **112T**, **114T**, **116T**, **118T** und **120T** mit dem Proxyserver **200** verbunden. Bei einer konventionellen Proxyserververwendung würde der Proxyserver für jede Client-Verbindung eine entsprechende TCP-Verbindung mit dem Webserver aufrechterhalten. Bei einem üblichen Zugang verändert der Proxyserver lediglich die Ursprungs- und die Ziel-IP-Adresse in den TCP/IP-Headern und leitet die ursprünglichen Pakete (ohne die Veränderungen in der Ursprungs- und Zieladresse) weiter. Aus der Perspektive der Clients und des Webserver ist die Existenz des Proxyserver transparent, was bedeutet, dass sie überhaupt nicht merken, dass ein Proxyserver verwendet wird.

[0020] Der konventionelle Proxyserver-Zugang benötigt dennoch eine individuelle TCP-Verbindung, die für jeden Client, der mit dem Webserver kommuniziert, aufrechterhalten wird. Im Gegensatz dazu wird der TCP-Verkehr in der Ausführungsform, die in **Fig. 2a** und **Fig. 2b** veranschaulicht ist, der über die TCP-Verbindungen **110T**, **112T**, **114T**, **116T**, **118T** und **120T** übertragen wird, in Client-Datensegmente unterteilt, die unter Verwendung einer gemultiplexten TCP-Verbindung **202** zwischen dem Proxyserver **200** und dem Webserver **122** gemultiplext werden. Ein Aspekt des Implementierens einer gemultiplexten TCP-Verbindung ist eine Zuordnung und ein Betrieb verschiedener Pufferwarteschlangen, um den Verkehr für jeden TCP-Fluss zwischen einer Client-Vorrichtung und dem Server abzuwickeln. Zusätzlich dazu ist in Verbindung mit diesen Warteschlangen ein Flussverwaltungsschema implementiert, um wie durch die Bandbreitenzuordnungen **110BW**, **112BW**, **114BW**, **116BW**, **118BW** und **120BW** abgebildet, einen entsprechenden Teil der Bandbreite der gemultiplexten TCP-Verbindung zuzuordnen.

[0021] **Fig. 2c** ist ein Inter-Rechner-Verbindungsdiagramm, das eine optionale Konfiguration von **Fig. 2b** veranschaulicht, bei der der Proxyserver **200** über m gemultiplexte TCP-Verbindungen 202_1 , 202_2 , 202_m mit dem Webserver **122** verbunden ist. Anhängig von der Anzahl der eingehenden TCP-Verbindungen (d.h. von Clients zu dem Proxyserver), können verringerte Rückgaben bestehen, wenn die Anzahl der TCP-Verbindungen steigt. Es ist ferner anerkannt, dass jeder eines Proxyserver und eines Webserver typischerweise mehrere Netzwerkports aufweisen wird, dem jeweils eine getrennte IP-Adresse zugeordnet ist und der fähig ist, ein TCP-Endpunkt zu sein. Demgemäß gibt es Situationen, in denen es von Vorteil sein wird, zwischen einem Proxyserver und einem Webserver, die parallel arbeiten, mehrere gemultiplexte TCP-Verbindungen zu implementieren.

[0022] **Fig. 2d** ist ein Inter-Rechner-Verbindungsdiagramm, das eine andere optionale Konfiguration von **Fig. 2b** veranschaulicht, bei der der Proxyserver **200** mit einer getrennten gemultiplexten TCP-Verbindung zwischen dem Proxy und jedem Webserver als ein Proxy für k Webserver 122_1 , 122_2 , 122_k verwendet wird. Da die Auslastung eines Proxyserver typischerweise geringer sein wird als jene eines Webserver (auf Basis einer Verbindung pro Client), können Implementierungsumgebungen bestehen, in denen es vorteilhaft sein wird, einen Proxyserver einzusetzen, der als ein Proxy für mehrere Webserver arbeitet. Zusätzlich zu den in **Fig. 2b**, **Fig. 2c** und **Fig. 2d** veranschaulichten Konfigurationen können andere Konfigurationen implementiert werden, die einen Proxyserver unter Verwendung einer gemultiplexten TCP-Verbindung mit einem Webserver verbinden.

[0023] **Fig. 3a** veranschaulicht gemäß einer Ausführungsform schematisch Abwicklungs- und Weiterleitungsvorgänge, die durch den Proxyserver **200** in Verbindung mit eingehenden TCP-Paketen, die von den Clients **110**, **112**, **114**, **116**, **118** und **120** empfangen werden, ausgeführt werden. Der erste Satz von Verarbeitungsvorgängen ähnelt jenen, die durch einen konventionellen TCP-Endpunkt ausgeführt werden, der mehrere TCP-Verbindungen unterstützt. Wie veranschaulicht, wird eine Sequenz von TCP-Paketen **300** von einer gegebenen Client-Vorrichtung über eine entsprechende TCP-Verbindung, die zwischen der Client-Vorrichtung und dem Proxyserver **200** eingerichtet ist, transportiert. Paket-Datenströme sind in **Fig. 3a** zu veranschaulichenden Zwecken abgebildet - für eine gegebene HTTP-Anfrage oder -Antwort kann ein Client eines oder mehrere TCP-Pakete an den Webserver senden; mit der Zeit kann dies (aus der Perspektive einer empfangenden Vorrichtung) als ein Datenstrom von Paketen erscheinen, wenngleich angemerkt wird, dass ein HTTP-Verkehr, der sich nicht auf multimedialen Inhalt bezieht, anstelle einer Streaming-Verbindung eine zustandslose HTTP-Verbindung einsetzen wird. Zudem kann ein großer Teil des HTTP-Verkehrs von einem Client aus relativ kleinen HTTP-Anfragen bestehen, die oft in einem einzigen TCP-Paket übertragen werden können. Jedes TCP-Paket **300** umfasst Paketauslastungsdaten **301** wie eine HTTP-Anfrage.

[0024] Als Veranschaulichung ist jeder der Clients **110**, **112**, **114**, **116**, **118** und **120** so abgebildet, dass er über eine entsprechende TCP-Verbindung **110T**, **112T**, **114T**, **116T**, **118T** und **120T** eine Sequenz von Paketen sendet. Um eine TCP-Verbindung zu unterstützen, ist an jedem Ende der TCP-Verbindung ein TCP-Port (ein Software-Port, kein physikalischer Port) geöffnet, sodass jede TCP-Verbindung (aus der Perspektive des Proxyserver **200**) durch eine Kombination der IP-Adresse des Clients und der Nummer des TCP-Ports identifiziert werden kann. (Es wird angemerkt, dass ein gegebener Client mehrere gleichzeitige TCP-Verbindungen mit einem Server aufbauen kann, wenn mehrere TCP-Ports verwendet werden.) In einer Ausführungsform wird von dem Proxyserver **200** eine Client-ID- (Identifikation) Zuordnungstabelle **302** verwendet, um Client-IP-Adresse/TCP-Port-Nummer-Kombinationen den zugeordneten Client-ID-Werten zuzuordnen. Zum Zwecke der Veranschaulichung sind die Client-IDs in den Figuren hierin durch „Client“ + die Referenzzahl des Clients + „ID“ angezeigt, z.B. ist Client **110** ID die Client-ID für den Client **110** (wenn die TCP-Verbindung **110T** verwendet wird).

[0025] Wie bei einer konventionellen TCP-Endpunkt-Architektur wird jeder TCP-Verbindung (wie durch die Client-ID in **Fig. 3a** identifiziert) eine Eingangswarteschlange **304** zugeordnet. Eine TCP-Verbindungs-Eingangswarteschlange (die auch als ein

Puffer bezeichnet werden kann) wird verwendet, um Pakete vorübergehend zu puffern, wenn die Pakete empfangen werden, und um unter Verwendung der TCP-Bytefolgedaten in den TCP-Headern des Pakets beliebige Pakete, die nicht in der richtigen Reihenfolge sind, neu zu ordnen. Zusätzlich dazu setzt TCP einen Mechanismus zum Detektieren von fehlengelassenen Paketen ein und ein beliebiges fehlengelassenes Paket wird von der TCP-Quelle (in diesem Beispiel ein Client) neu übertragen werden müssen. Zwecks Einfachheit und Klarheit sind die Eingangswarteschlangen **304** als eine entsprechende einzelne Warteschlange für jede TCP-Verbindung abgebildet. In der Praxis können sie unter Verwendung von einem oder mehreren Puffern oder Warteschlangen, die im Speicher auf dem TCP-Endpunkt implementiert sind, implementiert sein. Im Allgemeinen kann eine Eingangswarteschlange abhängig von der Netzwerkadapterarchitektur über eine Software-schicht (z.B. eine TCP-Schicht in einem Software-Netzwerkstapel) implementiert sein, oder sie kann durch eine eingebettete Logik in dem Netzwerkadapter oder durch eine Kombination dieser beiden abgewickelt werden.

[0026] Wenn in einer Eingangswarteschlange **304** eine sequentielle Abfolge von TCP-Paketen detektiert wurde, werden alle oder ein Teil der sequentiellen Abfolge in eine Ausgangswarteschlange **306**, die dem TCP-Port zugeordnet ist, weitergeleitet (Weiterl). Alternativ dazu kann eine Ausgangswarteschlange **306** konfiguriert sein, um sequentielle Abfolgen von Paketen aus einer zugeordneten Eingangswarteschlange **304** zu „ziehen“. In einer Ausführungsform wird jede der Ausgangswarteschlangen **306** eine sequentielle Abfolge von Paket-Nutzlastdaten halten, die über eine bestimmte TCP-Client-Verbindung gesendet wurden. Zudem sind diese TCP/IP-Header in einer Ausführungsform von den Paket-Nutzlastdaten befreit und Nutzlast-Abgrenzungsmarkierungen sind dem Bitdatenstrom hinzugefügt, wie durch Client-Daten-Datenströme **307** abgebildet, worin das Pipe-Symbol „|“ zwischen Abfolgen von Bits eine Abgrenzung zwischen Paket-Nutzlastdaten darstellt. Bei einem alternativen Schema verbleiben die TCP-IP-Header in den Client-Daten-Datenströmen, sind jedoch befreit und werden während der nachfolgenden Verarbeitung durch den Webserver **122** ignoriert.

[0027] In einer Ausführungsform umfassen sequentielle Abfolgen der Bitdatenströme Segmente von Daten (Datensegmente), die in TCP-Pakete **308** gemultiplext (MUX) sind, die konfiguriert sind, um zwischen dem Proxyserver **200** und dem Webserver **122** übertragen zu werden. Der hier verwendete Begriff „MUX TCP“ soll diese Pakete von den TCP-Paketen **300** unterscheiden, die von den Clients empfangen werden - in ihrer Struktur umfassen die MUX-TCP-Pakete konventionelle TCP-Pakete, die eingebettete Header-Informationen umfassen, um das Ab-

wickeln von Datensegmenten in Verbindung mit dem Transport über die gemultiplexte TCP-Verbindung zu ermöglichen, wovon weitere Details in **Fig. 5** veranschaulicht und nachfolgend besprochen sind. Bevorzugt sind MUX-TCP-Pakete **308** so groß wie möglich, wodurch die relativen Bandbreiten-Verwaltungsdaten, die an die Kapselung verloren werden, verringert werden. Beispielsweise werden in einer Ausführungsform Jumbo-TCP-Pakete und Jumbo-Ethernet-Rahmen verwendet. Zusätzlich dazu kann die Größe der MUX-TCP-Pakete variieren, um die kollektive Größe ihrer eingekapselten Datensegmente aufzunehmen. In den MUX-TCP-Paketen kann nach Bedarf auch Padding implementiert sein.

[0028] In einer Ausführungsform sind Datensegmente, die sequentielle Abläufe von Bitdatenstrom-Daten umfassen, die den TCP-Paket-Nutzlastdaten entsprechen, die von den Clients **110**, **112**, **114**, **116**, **118** und **120** empfangen werden, eingekapselt und werden auf die folgende Weise über die gemultiplexte Verbindung **202** gesendet. Der Proxyserver **200** umfasst einen ausgehenden TCP-Datenstrom MUX und eine Anordnungslogik **310**, die einen Warteschlangenwähler **312**, einen ausgehenden Paketanordnungsblock **314**, einen Ethernet-Framing-Block **316** und eine Ethernet-Port-Ausgangswarteschlange **318**. Wie oben besprochen wird jede der Client-Ausgangswarteschlangen **306** eine sequentielle Abfolge von TCP-Paket-Nutzlast-Bitdatenstrom-Daten enthalten (außer eine bestimmte Warteschlange wurde geleert). Wie auch oben mit Verweis auf **Fig. 2b** besprochen, empfängt jeder der Clients **110**, **112**, **114**, **116**, **118** und **120** eine entsprechende Bandbreitenzuordnung **110BW**, **112BW**, **114BW**, **116BW**, **118BW** und **120BW**.

[0029] Die Bandbreitenzuordnung wird durch den Warteschlangenwähler **310** verwaltet (in Kombination mit Logik in sowohl dem Proxyserver **200** als auch dem Webserver **122**, wie nachfolgend besprochen), der auswählt, von welcher Client-Ausgangswarteschlange **306** Datensegmente gezogen werden, um in ein gegebenes TCP-Paket **308** eingekapselt zu werden. Verschiedene Arten von Auswahlalgorithmen können eingesetzt werden, wie Round-Robin, gewichtetes oder ungewichtetes Fair-Queueing, eine Auswahl basierend auf dem Füllstand der Warteschlangen etc. Der den Clients zugeordnete Bandbreitenabschnitt kann angesichts von Veränderungen der Verkehrsmuster und anderen Überlegungen dynamisch verändert werden. Es wird zusätzlich dazu angemerkt, dass wenn zwischen neu verbundenen Clients und dem Proxyserver **200** neue Client-TCP-Verbindungen aufgebaut werden, die Client-Eingangswarteschlangen **304** und die Client-Ausgangswarteschlangen **306** dynamisch zugeordnet werden. Ähnlich dazu wird, wenn detektiert wird, dass eine TCP-Verbindung abgebaut wurde, der entsprechende Pufferaum, der der Client-Eingangswarte-

schlange **304** und der Client-Ausgangswarteschlange **306**, die dieser TCP-Verbindung zugeordnet sind, freigesetzt, um einer neuen TCP-Verbindung neu zugeordnet zu werden.

[0030] Die sequentielle Abfolge von Bits, die Datenssegmente umfassen, die durch den Warteschlangenauswähler **312** von einem Client-Ausgangspuffer **306** gezogen wurden, sind durch den ausgehenden Paketanordnungsblock **314** in einem MUX-TCP-Paket **308** eingekapselt. In einer Ausführungsform wird die gemultiplexte TCP-Verbindung **202** als eine konventionelle TCP-Verbindung betrieben, wie sie auf die MUX-TCP-Pakete **308**, die zwischen dem Proxyserver **200** und dem Webserver **122** übertragen werden, angewendet wird. Jedoch verwenden sowohl die Erzeugung des Paket-Datenstroms an dem Proxyserver **200** als auch die Abwicklung des empfangenen Paket-Datenstroms an dem Webserver **122** nicht konventionelle Verfahren.

[0031] Fig. 5 veranschaulicht eine Ausführungsform eines MUX-TCP-Pakets **308**, das einen MUX-TCP/IP-Header **308H** umfasst, gefolgt von einer Paket-Nutzlast **308P**, die n eingekapselte Datenssegmente $309_1 \dots 309_n$, gefolgt von einem TCP-Footer **308F**. In einer Ausführungsform umfasst der MUX-TCP/IP-Header **308H** konventionelle TCP/IP-Header-Felder, die in Klammern gezeigt sind, einschließlich Ursprungs- und Ziel-IP-Adressen für den Proxyserver **200** und den Webserver **122**, TCP-Byte-Folgenummer etc., sowie einige zusätzliche Parameter, die optionalen Feldern in dem TCP/IP-Header hinzugefügt werden können. In einer Ausführungsform umfassen die Parameter ein Daten/Steuerungsstatuszeichen, eine Client-ID, eine Byte-Folgenummer und eine Länge (Len). Die Parameter können in individuelle Felder eingebettet sein, oder sie können kodiert sein, sodass sie in eine geringere Anzahl von Feldern eingebettet sein können, einschließlich in ein einziges Feld. Als eine andere Option können die Parameter in einem Parameter-String oder dergleichen kodiert sein, der an dem Beginn der Paket-Nutzlast **308P** lokalisiert ist. In der veranschaulichten Ausführungsform umfasst jedes der Datenssegmente $309_1 \dots 309_n$ einen Header **311**, der eine Client-ID, eine Segmentlänge und eine Byte-(Folge-)Nummer. Optionale Abgrenzungsdaten können an dem Ende jedes Datenssegments **309** hinzugefügt sein, ähnlich zu den Abgrenzungsdaten, die verwendet werden, um Ethernet-Rahmen (z.B. eine vordefinierte Bitsequenz wie einen String aus 1en oder 0en) zu trennen. Die Byte-Folgenummer wird verwendet, um die Bytesequenz in dem Bitdatenstrom neu zu ordnen, im Falle, dass ein MUX-TCP/IP-Paket **308** fallengelassen wird und neu gesendet werden muss.

[0032] Eines oder mehrere MUX-TCP-Pakete **308** sind in einen Ethernet-Rahmen im Ethernet-Framing-Block **316** auf die konventionelle Art eingekapselt.

Bevorzugt können Jumbo-Ethernet-Rahmen verwendet werden, um die Bandbreite, die aufgrund der Ethernet-Framing-Verwaltungsdaten verloren wurde, zu verringern. Die Ethernet-Rahmen warten dann in der Ethernet-Port-Ausgangswarteschlange **318** und werden nachfolgend an den Server **122** gestreamt.

[0033] Fig. 3b veranschaulicht, wie MUX-TCP-Pakete **308**, die über die gemultiplexte TCP-Verbindung **202** von dem Proxyserver **200** empfangen werden, auf dem Webserver **122** verarbeitet werden. Die Paketverarbeitung ist durch einen Eingangsdatenstrom DeMUX und eine Trennungslogik **320** implementiert, beginnt damit, dass empfangene MUX-TCP-Pakete **308** in einem Ethernet-Eingangsport-Puffer **322** gepuffert werden. In einem Block **324** wird ein(e) Ethernet-Deframing und -Depaketierung ausgeführt, um MUX-TCP-Pakete **308** aus ihren Ethernet-Rahmen zu extrahieren. Wie durch die hellgraue Schattierung abgebildet, sind die Vorgänge, die von dem Ethernet-Eingangsport-Puffer **322** ausgeführt werden, und die Ethernet-Deframing- und -Depaketierungs-Vorgänge konventionelle Vorgänge, die an TCP-Paketen ausgeführt werden, die an einem Ethernet-Eingangsport empfangen werden. Obwohl sie zwecks Einfachheit nicht gezeigt sind, würden auch zusätzliche Vorgänge, die sich auf das Implementieren des TCP-Protokolls beziehen, unter Verwendung wohlbekannter Verfahren ausgeführt werden.

[0034] Die nachfolgenden Verarbeitungsvorgänge entsprechen nicht dem Standard und sind neu. In einem eingekapselten Datenssegment-Trennungsblock **326** werden die depaketierten TCP-Pakete **300** getrennt, um pro Client-ID einen Bitdatenstrom von Datenssegmentdaten auszugeben. Empfangene Datenssegmente, die sequentielle Abfolgen von Bits für einen gegebenen Client umfassen, werden dann durch einen Warteschlangenauswähler **328** in eine geeignete TCP-Port-Eingangswarteschlange **330** gedrückt, die der Client-ID zugeordnet ist, die in dem Datenssegment-Header **311** enthalten ist. Nachfolgend werden die Datenssegmente in Sequenz verarbeitet, um neu angeordnete Client-Daten-Datenstroms zu erzeugen, die Paket-Nutzlastdaten **301R** in ihrem ursprünglichen Format umfassen. Wie durch neu angeordnete Client-Daten-Datenströme **110R**, **112R**, **114R**, **116R**, **118R** und **120R** abgebildet, wird der Client-Daten-Datenstrom für jede TCP-Verbindung an einem Software-Verarbeitungsstapel **332** empfangen. Die veranschaulichten Komponenten für den Software-Verarbeitungsstapel **332** umfassen einen Client-Datenstrom-Verarbeitungsblock **334** und eine HTTP/HTTPS-Schicht **338**. Der Client-Datenstrom-Verarbeitungsblock **334** wird verwendet, um die Paket-Nutzlastdaten, die HTTP-Daten (z.B. eine HTTP-Anfrage) umfassen, zu extrahieren und sie an die HTTP/HTTPS-Schicht **338** weiterzuleiten. In einer Ausführungsform ist der Client-Daten-Datenstrom-Verarbeitungsblock **334** derart implemen-

tiert, dass die HTTP-Daten aus der Perspektive der HTTP/HTTPS-Schicht **338** unter Verwendung eines konventionellen TCP-Pakets gesendet wurden. Die HTTP/HTTPS-Schicht **338** umfasst einen HTTP/HTTPS-Serverfunktionsblock **340**, der konfiguriert ist, um konventionelle HTTP- oder HTTPS-Serverfunktionen (wie zutreffend, davon abhängig, ob die Verbindung eine HTTP- oder HTTPS-Verbindung ist) auszuführen. Als Veranschaulichung ist dies ferner durch einen HTTP-Client-Sitzungszustand **342** für jede Client-ID abgebildet, wobei beobachtet wird, dass HTTP selbst zustandslos ist, während eine HTTP-Streaming-Verbindung eine ständige Verbindung unterstützt.

[0035] Um einen Verkehrsfluss in die umgekehrte Richtung von dem Webserver **122** über den Proxyserver **200** zu den Clients **110**, **112**, **114**, **116**, **118** und **120** zu unterstützen, sind ähnliche Komponenten und Logiken, die in **Fig. 3a** und **Fig. 3b** gezeigt sind, vertauscht, wie in **Fig. 4a**, **Fig. 4a'** und **Fig. 4b** gezeigt, wobei gleich nummerierte Komponenten ähnliche Funktionen ausführen. Zusätzlich dazu sind die folgenden Vorkehrungen hinzugefügt.

[0036] Genauer gesagt empfangen Software-Anwendungen, die auf dem Webserver **122** oder auf einem anderen Server (z.B. einem Anwendungsserver) laufen, der mit dem Webserver **122** verbunden ist, HTTP-Anfragen, die von Clients gesendet werden und erzeugen HTTP-Antworten, die durch Clients zurückgegeben werden sollen. Ein Teil des HTTP-Verkehrs kann gänzlich durch den HTTP/HTTPS-Serverfunktionsblock **340** abgewickelt werden, wie das Aufbauen und Abbauen von HTTP- und TCP-Verbindungen. In anderen Fällen wird die Daten-Nutzlast, die einer eingehenden HTTP-Kommunikation zugeordnet ist, durch die HTTP-Schicht herausgetrennt und an die zutreffende Anwendung weitergeleitet. In der umgekehrten Richtung erzeugen Anwendungen HTTP-Antwortdaten, die an HTTP-Clients zurückgegeben werden.

[0037] Die HTTP-Antwortdaten werden an der HTTP-Schicht **338** empfangen und ein zutreffender HTTP-Header wird hinzugefügt (abhängig von der Art der HTTP-Antwort). Anschließend wird die HTTP-Antwort, die Paket-Nutzlastdaten **303** umfasst, einem ausgehenden Client-Daten-Datenstrom hinzugefügt, wie durch Client-Daten-Datenströme **110T**, **112T**, **114T**, **116T**, **118T** und **120T** abgebildet. Der ausgehende Client-Daten-Datenstrom für jede TCP-Verbindung wird in einer geeigneten Client-Ausgangswarteschlange **346** gepuffert. In Verbindung mit dem Öffnen einer Client-ID-TCP-Verbindung wird durch den Webserver **122** eine Client-Ausgangswarteschlange **346** zugeordnet. In einer Ausführungsform wird die zutreffende Client-Ausgangswarteschlange **346** durch ein Durchsuchen der Client-ID-Zuordnungstabelle **302** identifiziert.

[0038] Der Bitdatenstrom-Inhalt, der die Client-Daten-Datenströme umfasst, wird auf eine Weise, die ähnlich dem ist, wie der Client-Daten-Datenstrom-Inhalt in den Client-Ausgangswarteschlangen **306** von dem Proxyserver **200** an den Webserver **122** weitergeleitet wird, über die gemultiplexte TCP-Verbindung als Datensegmente an den Proxyserver **200** weitergeleitet. Wie zuvor umfasst dies das Einkapseln der Datensegmente **309** in MUX-TCP-Pakete **308** und das Transportieren der MUX-TCP-Pakete über die gemultiplexte TCP-Verbindung.

[0039] Bei **Fig. 4a** fortsetzend wird der Datenstrom des MUX-TCP-Pakets **308** auf eine ähnliche Weise wie zuvor mit Verweis auf **Fig. 3b** besprochen durch den MUX-TCP-Eingangsdatenstrom DeMUX und die Trennungslogik **320** empfangen und verarbeitet. In einer Ausführungsform wird durch den Proxyserver **200** in Verbindung mit dem Aufsetzen einer Client-ID-TCP-Verbindung eine Client-Ausgangswarteschlange **348** und eine Client-Eingangswarteschlange **350** zugeordnet. Wie ferner in **Fig. 4a** veranschaulicht, wird der Bitdatenstrom-Inhalt für jedes durch den Datensegment-Trennungsblock **326** eingekapselte Datensegment durch den Warteschlangenwähler **328** in eine zutreffende Client-Eingangswarteschlange **350** gedrückt und nachfolgend werden die Client-Daten-Datenströme entweder in die zutreffende Client-Ausgangswarteschlange **348** gedrückt oder sie werden von der Client-Eingangswarteschlange in die Client-Ausgangswarteschlange gezogen. Sobald sie in den Client-Ausgangswarteschlangen **348** sind, werden die Paket-Nutzlastdaten **303** extrahiert und in TCP-Pakete **305** eingekapselt, die dann unter Verwendung einer konventionellen ausgehenden TCP-Paket-Abwicklung an Clients weitergeleitet werden können, worin TCP-Pakete von einem Netzwerk-Port auf dem Proxyserver **200**, der durch die Zieladresse in dem TCP/IP-Header identifiziert ist (z.B. über eine weiterleitende tabellengestützte Suche), gesendet werden.

[0040] Eine optionale Konfiguration für einen Proxyserver **200a** ist in **Fig. 4a'** gezeigt. In dieser Konfiguration werden alle oder ein Teil der Client-Daten-Datenströme an einen Ausgangspaketpuffer **352** weitergeleitet, worin Paket-Nutzlastdaten **303** extrahiert werden und in TCP-Pakete **305** eingekapselt werden. Die TCP-Pakete **305** werden dann auf die konventionelle Weise durch den Proxyserver **200a** an Clients **110**, **112**, **114**, **116**, **118** und **120** weitergeleitet.

[0041] Zusätzlich zu den vorangegangenen gemultiplexten TCP-Paketverarbeitungsvorgängen umfassen der Proxyserver **200** und der Webserver **122** eine zusätzliche Logik zum Ausführen eines TCP-Verbindungsaufbaus und von Clientbasierten TCP-Verbindungsflusssteuerungsvorgängen. Auf dem Proxyserver **200** werden diese Vorgänge durch eine Proxyserver-Steuerungslogik **356** (zum Anschließen an

TCP-Clients) und eine Proxyserver-zu-Webserver-Steuerungslogik **358** ermöglicht. Auf dem Webserver **122** ist dies als eine Proxyserver-zu-Webserver-Steuerungslogik **360** abgebildet. Die Verwaltung von Verbindungen und Client-Warteschlangen wird durch Steuerungsnachrichten, die zwischen dem Proxyserver **200** und dem Webserver **122** gesendet werden, ermöglicht. Nachrichtenflussdiagramme, die einem beispielhaften Aufbau und/oder Steuerungsfunktionen entsprechen, sind in **Fig. 6** und **Fig. 7** abgebildet.

[0042] **Fig. 6** zeigt ein Nachrichtenflussdiagramm, das Nachrichten veranschaulicht, die zwischen einem Client **120**, einem Proxyserver **200** und einem Webserver **122** gesendet werden. Die Nachrichtensequenz beginnt mit einer HTTP-Anfrage **600**, die von dem Client **120** an den Webserver **122** gesendet wird. Wie oben besprochen, kann dies durch einen Benutzer des Clients **120** erreicht werden, der den URL für eine Webseite eingibt, die durch den Webserver **122** gehostet oder auf die anders durch diesen zugegriffen wird. Als Antwort wird der DNS-Server (nicht gezeigt) die IP-Adressen für den Proxyserver **200** anstelle des Webserver **122** zurückgeben.

[0043] Eine HTTP-Sitzung ist eine Sequenz von Anfrage-Antwort-Netzwerktransaktionen. Ein HTTP-Client initiiert eine Anfrage, indem auf einem Server eine TCP-Verbindung mit einem bestimmten Port (typischerweise Port **80**, obwohl auch viele andere Ports verwendet werden können) aufgebaut wird. Ein HTTP-Server an der IP-Adresse, die in der anfänglichen HTTP-Anfrage umfasst ist, die diesen Port „abhört“, ist konfiguriert, um die TCP-Verbindung aufzubauen. Bei dem konventionellen Ansatz sendet der HTTP-Server, wenn er die Anfrage empfängt, eine HTTP-Antwortnachricht zurück, die eine Zustandszeile wie „HTTP/1.1 200 OK“ umfasst. Der Körper dieser Nachricht ist typischerweise die angefragte Ressource, obwohl auch eine Fehlermeldung oder andere Informationen zurückgegeben werden können.

[0044] Bei dem hierin bereitgestellten Proxyserver-Zugang tritt eine andere Sequenz auf. Beim Empfang der HTTP-Anfrage **600** erkennt der Proxyserver **200** diese als eine Anfrage für eine neue TCP-Verbindung und leitet die Anfrage als eine Nachricht **602** an den Webserver **122** weiter. Der Webserver **122** überprüft, ob er weitere Verbindungen unterstützen kann, und wenn ja, gibt er an den Proxyserver **200** eine neue Proxyverbindung-Zuordnungsanfrage **604** zurück, die eine Client-ID umfasst. Als Antwort ordnet der Proxyserver **200** dieser Client-ID einen geeigneten Satz von Warteschlangen zu (wenn verfügbar) und gibt eine neue Proxyverbindung-Anerkennungsnachricht (ACK) **606** zurück. Wenn der Proxyserver **200** die Client-ID-Warteschlangen nicht zuordnen kann, wird er eine andere Nachricht zurückgeben, die anzeigt, dass zu diesem Zeitpunkt keine neue Proxyverbindung unterstützt werden kann.

[0045] Wie oben besprochen werden zwischen den Clients und dem Proxyserver individuelle TCP-Verbindungen aufgebaut. Demgemäß gibt der Proxyserver **200**, wenn die neue Proxyverbindung erfolgreich zugeordnet ist, dem Client **120** gemeinsam mit der TCP-Port-Nummer eine HTTP-Antwort **608** zurück, die anzeigt, dass die TCP-Verbindung aufgebaut wurde. Da es möglich ist, dass ein gegebener Client mehrere TCP-Verbindungen mit demselben Proxyserver aufbaut (wobei jede unterschiedliche TCP-Ports verwendet), wird jede einzigartige TCP-Verbindung durch die Kombination der IP-Adresse des Clients und der TCP-Port-Nummer identifiziert, wie auch oben identifiziert.

[0046] An diesem Punkt besteht zwischen dem Client **120** und dem Proxyserver **200** eine HTTP-Verbindung, jedoch erscheint es für alle Absichten und Zwecke aus der Perspektive von sowohl dem Client **120** als auch den HTTP/HTTPS-Serverfunktionen **340**, die auf dem Webserver **122** arbeiten, dass die HTTP-Verbindung zwischen dem Client **120** und dem Webserver **122** besteht. Daher ist der Client **120** ein physikalischer HTTP-Endpunkt und der Webserver **122** ist ein virtueller HTTP-Endpunkt. Ein nachfolgender Austausch von HTTP-Anfragen und HTTP-Antworten (durch HTTP-Anfragen **610** und **614** sowie HTTP-Antworten **612** und **616** abgebildet) wird dann zwischen diesen HTTP-Endpunkten ausgetauscht, wobei jede HTTP-Anfrage zuerst eine TCP-Verbindung zwischen dem Client und dem Proxyserver durchquert und dann über die gemultiplexte TCP-Verbindung an den Webserver weitergeleitet wird, und jede HTTP-Antwort den umgekehrten Pfad durchquert.

[0047] Wie durch eine HTTP-Freigabenachricht **618** abgebildet, kann jeder der HTTP-Endpunkte ausdrücklich anfragen, dass die HTTP-Verbindung freigegeben wird. Beispielsweise sendet der Webbrowser (der den Client umfasst) bei einem sich ordnungsgemäß verhaltenden Webbrowser wie Google Chrome, Microsoft Internet Explorer, Apple Safari oder Mozilla Firefox, wenn ein Benutzer eine Registerkarte schließt, eine HTTP-Freigabenachricht an den Webserver, der dem URL für die Registerkarte, die geschlossen wurde, zugeordnet ist. Gegebenenfalls kann ein Webserver über ein HTTP-Timeout **620** detektieren, dass eine HTTP-Verbindung nicht mehr besteht (was keine getrennte Nachricht ist, sondern eher etwas, das intern durch den Webserver bestimmt wurde).

[0048] An diesem Punkt wird die Proxyverbindung zusammen mit den Client-ID-Warteschlangenressourcen freigegeben. Der Webserver **122** sendet eine Proxyverbindung-Freigabenachricht **622**, die die Client-ID umfasst, an den Proxyserver **200**. Als Antwort gibt der Proxyserver **200** eine Proxyverbindung-Freigabe-ACK-Nachricht zurück, die die Client-ID umfasst. An diesem Punkt gibt sowohl der Proxy-

server **200** als auch der Webserver **122** die Client-ID-Warteschlangen frei, die der Proxyverbindung, die freigegeben wurde, zugeordnet waren.

[0049] Ein neuer Aspekt der gemultiplexten TCP-Architektur ist die Fähigkeit, dass ein Webserver eine Flusssteuerung für individuelle TCP-Verbindungen ausführt. Wenngleich die hierin offenbarten gemultiplexten TCP-Verbindungen beispielsweise manche Ähnlichkeiten mit konventionellen getunnelten Verbindungen teilen, kapseln solche getunnelten Verbindungen lediglich Paketdatenströme ein, ohne Vorkehrungen für getrennte Bandbreitenzuordnungen für individuelle TCP-Verbindungen. Bei den hierin offenbarten Verfahren kann ein Webserver den einer gegebenen TCP-Verbindung zugeordneten Bandbreitenabschnitt dynamisch neu zuordnen, einschließlich eines vorübergehenden Unterbrechens der Verbindung. In Verbindung mit diesen Flusssteuerungsvorgängen, die durch den Webserver verwaltet werden, kann der Proxyserver (wie zutreffend) an seinen TCP-Verbindungen mit dem Client entsprechende TCP-Flusssteuerungsmaßnahmen implementieren, um Pufferüberläufe und daraus resultierende Paketabfälle zu verhindern.

[0050] In einer Ausführungsform ist ein Webserver aktiviert, um einen Proxyserver dazu zu informieren oder anzuweisen, wie viele Clients der Webserver unterstützen kann. Wie in **Fig. 7** gezeigt kann dies mit einer Nachricht zur Anzahl der unterstützten Clients **700** erreicht werden, die eine Anzahl (n) von Clients identifiziert, die der Webserver **122** unterstützen kann. Als Antwort gibt der Proxyserver **200** eine ACK-Nachricht zur Anzahl der unterstützten Clients **702** zurück.

[0051] Ein Webserver kann auch einen Proxyserver anweisen, einen gegebenen Client zu drosseln oder das Weiterleiten von Paketen für einen gegebenen Client vorübergehend abzubrechen. Beispielsweise kann der Webserver **122** einen Drosselungsbefehl **704** senden, der einen Proxyserver **200** anweist, das Weiterleiten von Paketen, die einer bestimmten Client-Verbindung entsprechen, wie durch die Client-ID identifiziert, zu drosseln. Als Antwort wird der Proxyserver **200** mit einer Drosselungsbefehl-ACK-Nachricht **706** antworten. Wenn noch immer zu viele eingehende Daten bestehen, kann ein Webserver selektiv entscheiden, den Empfang von TCP-Paketen für eine oder mehrere Proxy-Verbindungen vorübergehend zu unterbrechen. Dies ist in **Fig. 7** als eine Verbindungsabbruchnachricht **708** abgebildet, die die Client-ID identifiziert, die vorübergehend unterbrochen werden soll, sowie die Byte-Folgennummer, die das letzte zu sendende Byte vor dem Unterbrechen darstellt. Der Proxyserver **200** antwortet mit einer Unterbrechungs-ACK-Nachricht, die die Client-ID identifiziert.

[0052] Während laufender Vorgänge kann ein Webserver detektieren, dass er zu viele Clients bedient (und als Ergebnis gezwungen ist, Pakete fallenzulassen). Als Antwort auf diese Art von Bedingungen sendet der Webserver **122** eine Client-Abweisungsnachricht **712**, die eine oder mehrere Client-IDs identifiziert, die abzuweisenden Client-Verbindungen entsprechen. Als Antwort gibt der Proxyserver **200** eine Client-Abweisungs-ACK-Nachricht **714** zurück. Der Proxyserver **200** sendet auch eine HTTP-Verbindungsabbaunachricht **716** an den/die zutreffenden Client(s).

[0053] Ein Proxyserver und/oder Webserver kann Informationen, die den Warteschlangen-Füllstand von einer oder mehreren Warteschlangen und zugeordneten Client-IDs betreffen, senden, wie durch einen Warteschlangen-Füllstand **718** abgebildet. Diese Art von Nachricht kann auch in Verbindung mit einem Drosselungsbefehl gesendet werden. Als eine andere Option kann ein sendender Server einem empfangenden Server eine(n) Hochdrosselungsbefehl oder -anfrage senden, die den empfangenden Server anfragt oder diesen anweist, einen erhöhten Paketfluss für eine gegebene Client-ID-Verbindung zu erwarten oder zu konfigurieren.

[0054] Im Allgemeinen können Steuerungsnachrichten zwischen einem Proxyserver und einem Webserver über eine gemultiplexte TCP-Verbindung gesendet werden (z.B. indem das Daten/Steuerungsstatuszeichen auf den Steuerungsindikator gesetzt wird), oder sie können über eine getrennte Verbindung (nicht gezeigt) gesendet werden, die manchmal als Rückkanal, Steuerungskanal oder Seitenbandverbindung bezeichnet wird. Verschiedene Paketformate können verwendet werden, um die Steuerungsnachrichtenübermittlung zu ermöglichen, und das bestimmte verwendete Format liegt außerhalb des Schutzzumfangs dieser Erfindung, da Verfahren zum Implementieren einer Steuerungsnachrichtenübermittlung zwischen Servern auf dem Gebiet der Erfindung wohlbekannt sind. Zusätzlich dazu könnten Nachrichten, die in **Fig. 7** abgebildet sind, wie sie von dem Webserver **122** stammen unter zutreffenden Umständen auch von dem Proxyserver **200** stammen. Beispielsweise kann ein Proxyserver auch einen Drosselungsbefehl oder einen Befehl zum vorübergehenden Abbruch an einen Webserver ausgeben.

[0055] Weitere Beispiele von Steuerungsinformationen umfassen das Folgende. Die Steuerungsinformationen von einem Proxy könnten sagen: „Neuer Client, ich möchte den Nickname N verwenden.“ Als Antwort könnte der Server „Zu viele Clients“ oder „OK. Ich nehme den neuen Clienten N an.“ sagen. In einer Ausführungsform beginnt die Folgenummer für Daten für Client N mit 1 oder einer anderen vordefinierten Zahl (im Gegensatz zu TCP, das mit anfäng-

lichen Folgenummern seltsame Dinge macht, aber hier ist sie innerhalb einer TCP-Verbindung zwischen dem Proxy und dem Server, daher ist es in Ordnung, die Daten jedes Clients mit 1 zu beginnen).

[0056] Um das Abbauen einer Verbindung zu initiieren, könnte entweder der Proxy oder der Server eine Steuerungsnachricht senden, die „Client N ist fertig und ich möchte die Client-ID #N wieder verwenden können“ sagt. Als Antwort könnte der Empfänger (entweder der Proxy oder der Server, wie zutreffend) sagen: „Ich nehme die Daten des Client N bis zur Folgenummer **S1** und die Daten von Client J bis zur Folgenummer **S2** an.“

[0057] Als ein weiteres Beispiel könnte der Proxy vielleicht bei Folgenummer **191** für die P-S-TCP-Verbindung (interne TCP-Informationen an den Server) senden:

Client X, Sequenz #15, 92 Byte Daten

Client Y, Sequenz #51, 12 Byte Daten

[0058] Als Antwort könnte der Server „Ich habe alle Daten, die du bis zu unserer TCP-Folgenummer **305** gesendet hast, angenommen“ sagen oder er könnte getrennt davon „ACK Client X bis 106, ACK Client Y bis 63“ sagen. Oder der Server könnte, wenn er keine Daten mehr von Y annehmen kann, „ACK Client Y bis 53“ sagen (wenn er nur 2 Byte Daten von Y angenommen hat, und dann wird der Proxy den Rest von Y später senden müssen).

Beispielhafte Implementierungsumgebung und Serverarchitekturen

[0059] Es ist vergegenwärtigt, dass Aspekte der Ausführungsformen hierin in verschiedenen Arten von Rechner- und Netzwerkausstattungen wie verschiedenen Arten von Servern, die in einer Datenzentrum- und/oder Serverfarmumgebung eingesetzt werden, implementiert werden können. Typischerweise umfassen die Server, die in Datenzentren und Serverfarmen verwendet werden, angeordnete Serverkonfigurationen wie Gestell-basierte Server oder Bladeserver. Diese Server sind zur Kommunikation über verschiedene Netzwerkvorkehrungen zwischenverbunden, wie dass Serversätze in LANs unterteilt werden, mit geeigneten Schalt- und Führeinrichtungen zwischen den LANs, um ein privates Internet zu bilden. Beispielsweise können Cloud-Hosting-Einrichtungen typischerweise große Datenzentren mit einer Vielzahl von Servern einsetzen.

[0060] Als Überblick sind in **Fig. 8a-c** und **Fig. 9** typische Bladeserver-Komponenten und -Systeme gezeigt. Bei einer typischen Konfiguration wird ein Einschubgehäuse **800** eingesetzt, um für eine Vielzahl von Serverblades (auch bekannt als Blades) **802** Leistungs- und Kommunikationsfunktionen bereitzustellen, wovon jedes einen entsprechenden Steck-

platz besetzt. (Es wird angemerkt, dass nicht alle Steckplätze in einem Gehäuse besetzt sein müssen.) Im Gegenzug kann eines oder mehrere Gehäuse **800** in ein in **Fig. 8c** gezeigtes Bladeserver-Gestell **803** eingebaut sein. Jedes Blade ist beim Einbau über eine oder mehrere passende Steckverbindungen an eine Schnittstellenebene **804** (z.B. eine Rückebene oder eine Zwischenebene) gekoppelt. Typischerweise wird die Schnittstellenebene eine Vielzahl entsprechender passender Steckverbindungen umfassen, die den Blades Leistungs- und Kommunikationssignale bereitstellen, und geführte Signalfade zum Koppeln von Ethernet-Signalen zwischen Blades beinhalten. Bei den derzeitigen Praktiken stellen Schnittstellenebenen eine „Hot-Swapping“-Funktionalität bereit - das bedeutet, Blades können während des Betriebs hinzugefügt oder entfernt („hot-geswappt“) werden, ohne dass das gesamte Gehäuse durch ein geeignetes Leistungs- und Datensignalpuffern abgenommen wird.

[0061] In **Fig. 8a** und **Fig. 8b** ist eine typische Zwischenebene-Schnittstellenebene-Konfiguration gezeigt. Die Rückseite der Schnittstellenebene **804** ist an eine oder mehrere Leistungsversorgungen **806** gekoppelt. Oftmals sind die Leistungsversorgungen redundant und Hot-Swap-fähig, indem sie an geeignete Leistungsebenen und Aufbereitungsschaltungen gekoppelt sind, um im Fall eines Leistungsversorgungsversagens einen weiteren Betrieb zu ermöglichen. Bei einer optimalen Konfiguration kann eine Anordnung von Leistungsversorgungen verwendet werden, um Leistung an ein gesamtes Blade-Gestell zu liefern, worin es keine eins-zueins Entsprechung zwischen der Leistungsversorgung und dem Gehäuse gibt. Eine Vielzahl von Kühlungsventilatoren **808** wird eingesetzt, um Luft durch das Gehäuse zu ziehen, um die Serverblades zu kühlen.

[0062] Ein wichtiges Merkmal, das von allen Bladeservern erfordert wird, ist die Fähigkeit, extern mit anderen IT-Infrastrukturen zu kommunizieren. Dies wird typischerweise über eine oder mehrere Netzwerkverbindungskarten **810** ermöglicht, von denen jede an die Schnittstellenebene **804** gekoppelt ist. Im Allgemeinen kann eine Netzwerkverbindungskarte eine physikalische Schnittstelle umfassen, die eine Vielzahl von Netzwerkport-Verbindungen (z.B. RJ-45-Ports) umfasst, oder sie kann eine Steckverbindung hoher Dichte Anschluss umfassen, der konzipiert ist, um eine direkte Verbindung mit einer Netzwerkvorrichtung wie einem Netzwerkschalt, -hub oder -router herzustellen. Es wird angemerkt, dass auch andere Netzwerkarchitekturen verwendet werden können, wie eine Top-of-Rack-Schaltungsarchitektur und geteilte Schaltungsarchitekturen, und die Netzwerkverknüpfungen können verdrahtete und/oder optische Kabel umfassen.

[0063] Bladeserver stellen üblicherweise eine Art Verwaltungsschnittstelle zum Verwalten von Vorgängen auf den individuellen Blades bereit. Dies kann im Allgemeinen durch ein eingebautes Netzwerk oder einen Kommunikationskanal oder -Kanäle ermöglicht werden. Beispielsweise können einer oder mehrere Busse zum Ermöglichen eines „privaten“ oder „Verwaltungs-“ Netzwerks und geeignete Schaltungen in die Schnittstellenebene eingebaut sein, oder ein privates Netzwerk kann durch eine eng gekoppelte Netzwerkverkabelung und ein Netzwerk implementiert sein. Gegebenenfalls können das Schalten und eine andere Verwaltungsfunktionalität durch eine Verwaltungsschaltkarte **812** bereitgestellt sein, die an die Rückseite oder Vorderseite der Schnittstellenebene gekoppelt ist. Als noch eine andere Option kann ein Verwaltungs- oder Konfigurationsserver eingesetzt werden, um Blade-Aktivitäten zu verwalten, worin die Kommunikation über eine standardmäßige Computernetzwerk-Infrastruktur, beispielsweise Ethernet, abgewickelt wird.

[0064] Mit Verweis auf **Fig. 9** sind weitere Details eines beispielhaften Blades **900** gezeigt. Wie oben besprochen umfasst jedes Blade eine getrennte Rechenplattform, die konfiguriert ist, um serverähnliche Funktionen auszuführen, d.h. sie ist ein „Server auf einer Karte“. Demgemäß umfasst jedes Blade Komponenten, die bei konventionellen Servern üblich sind, einschließlich einer gedruckten Leiterplatte (Hauptplatine) **901**, die eine interne Verdrahtung (d.h. Busse) zum Koppeln von geeigneten integrierten Schaltkreisen (ICs) und andere Komponenten, die auf der Platte befestigt sind, umfasst. Diese Komponenten umfassen einen oder mehrere Prozessoren (auch bekannt als CPUs) **902**, die mit einem Systemspeicher **904** (z.B. eine Form eines Direktzugriffsspeichers (RAM)), einen Cache-Speicher **906** (z.B. SDRAM) und eine Firmware-Speichervorrichtung **908** (z.B. einen Flash-Speicher) gekoppelt sind. Ein NIC-(Netzwerkschnittstellensteuerungs-) Chip **910** ist bereitgestellt, um konventionelle Netzwerkkommunikationsfunktionen bereitzustellen, wie um eine Kommunikation zwischen einem Blade und einer externen Netzwerkinfrastruktur zu unterstützen. Andere veranschaulichte Komponenten umfassen Zustands-LEDs (lichtemittierende Dioden) **912**, einen oder mehrere RJ-45-Ethernet-Ports **914** (wovon zwecks Einfachheit nur einer gezeigt ist), und einen Schnittstellenebenenanschluss **916**. Zusätzliche Komponenten umfassen verschiedene passive Komponenten (z.B. Widerstände, Kondensatoren), Leistungsaufbereitungskomponenten und Peripherievorrichtungsanschlüsse.

[0065] Im Allgemeinen kann jedes Blade **900** auch einen eingebauten Speicher umfassen. Dies wird typischerweise über eine oder mehrere eingebaute Plattensteuerungen und entsprechende Anschlüsse, an die eines oder mehrere Plattenlaufwerke

918 gekoppelt sind, ermöglicht. Beispielsweise umfassen typische Plattensteuerungen SATA-Steuerungen, SCSI-Steuerungen und dergleichen. Im Allgemeinen kann ein Plattenlaufwerk **918** ein magnetisches Laufwerk oder ein Festkörperlaufwerk (SSD) umfassen. Als eine Option können die Plattenlaufwerke getrennt von den Blades in demselben oder einem getrennten Gestell aufgenommen sein, wie es der Fall sein kann, wenn eine netzgebundene Speichervorrichtung (NAS) oder ein Backend-Speicher-Subsystem zum Speichern von großen Datenmengen eingesetzt wird.

[0066] Der NIC **910** umfasst eine Schaltung und Logik zum Ermöglichen entsprechender Netzwerkvorgänge, wie einer Unterstützung einer physikalischen Schicht (**L1**) und Daten-Inter-Rechner-Verbindungsschichtvorgänge (**L2**) (z.B. Ethernet-MAC-Vorgänge). Typischerweise werden Vorgänge der oberen Schicht durch einen Betriebssystem-Netzwerkstapel ermöglicht, der durch ein Betriebssystem gehostet sein würde, das auf dem Prozessor **902** läuft. In manchen Ausführungsformen kann ein NIC jedoch über eine eingebettete Logik oder dergleichen seinen eigenen Netzwerkstapel einsetzen.

[0067] Im Allgemeinen setzen heutige Serverprozessoren System-on-a-Chip-(SoC-) Architekturen ein, bei denen mehrere funktionale Komponenten, Blöcke und Schnittstellen auf einem Chip integriert sind. Bei manchen SoC-Architekturen umfasst ein Prozessor einen CPU, der mehrere Kerne umfasst.

[0068] Bei einem typischen Datenzentrumseinsatz umfassen Netzwerkschalt Elemente eine Rackmount-Ausstattung, wie sie einen 1U-, 2U- oder 4U-Steckplatz besetzen würden, oder sie können über eines oder mehrere Serverblades implementiert sein. Gegebenenfalls kann ein Netzwerkschalt Element unter Verwendung von einem oder mehreren Serverblades implementiert sein.

[0069] Abhängig von der Auslastung können die hierin beschriebenen Proxyserver- und Webservervorgänge über ein einzelnes Blade, einen Satz von Blades, die einen Teil der Steckplätze in einem Gehäuse besetzen, ein Gehäuse mit mehreren Blades oder andere Arten von Servern implementiert sein, die in heutigen und zukünftigen Datenzentrumsumgebungen wie Servern mit mehreren Sockets eingesetzt werden. Die Proxyserver und Webserver können in manchen Ausführungsformen als physikalische Einheiten implementiert sein, während andere Ausführungsformen eine Kombination von physikalischen und virtuellen Einheiten einsetzen können.

[0070] Weitere Aspekte des hierin beschriebenen Gegenstands sind in den folgenden nummerierten Punkten dargelegt:

[0071] 1. Verfahren umfassend:

das Aufbauen einer Vielzahl von TCP-Verbindungen mit einer Vielzahl von Clients an einem Proxyserver, wobei jede TCP-Verbindung unter Verwendung eines TCP-Ports einen Client mit dem Proxyserver verbindet;

das Koppeln des Proxyservers in Kommunikation mit einem Webserver über eine gemultiplexte TCP-Verbindung;

das Empfangen von eingehenden TCP-Kommunikationseinheiten an dem Proxyserver, die von der Vielzahl von Clients stammen, wobei jede eingehende TCP-Kommunikation zumindest ein TCP-Paket umfasst, das Paket-Nutzlastdaten umfasst;

das Puffern der Paket-Nutzlastdaten, die in TCP-Paketen enthalten sind, die über eine gegebene TCP-Verbindung in einem entsprechenden Client-Daten-Datenstrom empfangen werden;

das Einkapseln von Datensegmenten, die in den Client-Daten-Datenströmen sequentielle Abfolgen von Bits umfassen, in gemultiplexte (MUX-) TCP-Pakete an dem Proxyserver, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente von zumindest zwei getrennten Client-Daten-Datenströmen einkapselt;

das Übertragen der MUX-TCP-Pakete von dem Proxyserver über die gemultiplexte TCP-Verbindung an den Webserver; und

an dem Webserver

das Entkapseln von Datensegmenten in den MUX-TCP-Paketen und das Neuaneordnen der Client-Daten-Datenströme; und

das Extrahieren von Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen.

[0072] 2. Verfahren nach Punkt 1, worin das eine oder die mehreren TCP-Pakete, die über eine erste TCP-Verbindung empfangen werden, eine HyperText-Übertragungsprotokoll- (HTTP-) Anfrage umfassen, die von einem ersten Client gesendet wird, wobei das Verfahren ferner das Zurückgeben einer HTTP-Antwort an den ersten Client umfasst, indem:

die HTTP-Antwort in einen Client-Daten-Datenstrom eingebettet wird, der für den ersten Client bestimmt ist,

Datensegmente, die sequentielle Abfolgen von Bits umfassen, die dem Client-Daten-Datenstrom entsprechen, in eines oder mehrere MUX-TCP-Pakete eingekapselt werden;

das eine oder die mehreren MUX-TCP-Pakete über die gemultiplexte TCP-Verbindung von

dem Webserver an den Proxyserver übertragen werden;

die Datensegmente von dem MUX-TCP-Paket entkapselt werden;

der Client-Daten-Datenstrom neu angeordnet wird; und

die HTTP-Antwort extrahiert wird und die HTTP-Antwort in eines oder mehrere TCP-Pakete eingekapselt wird, die über die erste TCP-Verbindung von dem Proxyserver an den betreffenden Client weitergeleitet werden.

[0073] 3. Verfahren nach Punkt 1 oder 2, ferner umfassend:

für jede TCP-Verbindung

das Zuordnen von zumindest einer Warteschlange auf dem Proxyserver für die TCP-Verbindung; und

das Puffern eines Client-Daten-Datenstroms, der Paket-Nutzlastdaten umfasst, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in der zumindest einen Warteschlange empfangen werden, die der TCP-Verbindung zugeordnet ist.

[0074] 4. Verfahren nach Punkt 3, worin die zumindest eine Warteschlange auf dem Proxyserver, die jeder TCP-Verbindung zugeordnet ist, eine Ausgangswarteschlange umfasst, die ferner umfasst:

das Puffern eines Client-Daten-Datenstroms, der sequentielle Abfolgen von Paket-Nutzlastdaten umfasst, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in den Ausgangswarteschlangen auf dem Proxyserver übertragen werden; worin jede Ausgangswarteschlange einer entsprechenden TCP-Verbindung zugeordnet ist;

das Erzeugen eines Datenstroms von MUX-TCP-Paketen, die über die gemultiplexte TCP-Verbindung an den Webserver übertragen werden sollen, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, von denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

[0075] 5. Verfahren nach einem der vorangegangenen Punkte, ferner umfassend:

das Empfangen einer HyperText-Übertragungsprotokoll- (HTTP-) Anfrage, die von einem neuen Client gesendet wird, um auf einen Uniform Resource Locator (URL) für eine Webseite zuzugreifen, auf die über den Webserver zugegriffen wird;

das Aufbauen einer neuen TCP-Verbindung zwischen dem neuen Client und dem Proxyserver; und

das Zuordnen von Warteschlangen auf jedem des Proxyserver und des Webserver, um das Weiterleiten von HTTP-Anfragen, die an dem Proxyserver empfangen werden, von dem neuen Client an den Webserver über die gemultiplexte TCP-Verbindung zu unterstützen, und um HTTP-Antworten, die an dem Webserver erzeugt werden, über die gemultiplexte TCP-Verbindung zu dem Proxyserver und die neue TCP-Verbindung an den neuen Client weiterzuleiten.

[0076] 6. Verfahren nach einem der vorangegangenen Punkte, worin die gemultiplexte TCP-Verbindung eine Gesamtbandbreite aufweist, wobei das Verfahren ferner umfasst:

das Zuordnen eines entsprechenden Bandbreitenabschnitts der Gesamtbandbreite der gemultiplexten TCP-Verbindung zu jeder TCP-Verbindung, die mit dem Proxyserver aufgebaut ist, und für die TCP-Pakete über die gemultiplexte TCP-Verbindung weitergeleitet werden sollen;

das Ermöglichen, dass der einer individuellen TCP-Verbindung zugeordnete Bandbreitenabschnitt dynamisch angepasst wird.

[0077] 7. Verfahren nach einem der vorangegangenen Punkte, ferner umfassend:

das Ermöglichen, dass zumindest einer aus dem Webserver und dem Proxyserver eine Datenübertragungsgeschwindigkeit für individuelle TCP-Verbindungen verwaltet, die über die gemultiplexte TCP-Verbindung weitergeleitet werden.

[0078] 8. Verfahren nach Punkt 7, ferner umfassend

das Ermöglichen, dass zumindest einer aus dem Webserver und dem Proxyserver unterbricht, dass der Verkehr, der einer ausgewählten TCP-Verbindung entspricht, über die gemultiplexte TCP-Verbindung weitergeleitet wird.

[0079] 9. Verfahren nach Punkt 8, worin der Webserver eine Anfrage an den Proxyserver sendet, um den Verkehr für eine gegebene TCP-Verbindung zu unterbrechen, wobei das Verfahren ferner umfasst:

das Fortsetzen des Pufferens von eingehendem TCP-Verkehr, der über die TCP-Verbindung von einem Client empfangen wird;

das Empfangen einer Anzeige, dass es dem Verkehr für die TCP-Verbindung gestattet ist, wieder aufgenommen zu werden; und

das Wiederaufnehmen des Sendens von Daten-segmenten, die dem TCP-Verkehr entsprechen, der sich in dem Puffer angesammelt hat, von dem Proxyserver zu dem Webserver.

[0080] 10. Verfahren nach einem der vorangegangenen Punkte, ferner umfassend:

das Senden von Informationen von dem Webserver an den Proxyserver, die eine Maximalanzahl von gleichzeitigen Daten-Datenströmen identifizieren, die gemultiplext und über die gemultiplexte TCP-Verbindung gesendet werden können;

das Aufbauen einer Anzahl von Client-Verbindungen, die der Maximalanzahl von gleichzeitigen Daten-Datenströmen entspricht;

das Empfangen von Daten von einem neuen Client, der mit dem Webserver kommunizieren möchte, an dem Proxyserver;

das Puffern der Daten von dem neuen Client; und

als Antwort auf das Bestimmen, dass ein bestehender Client das Kommunizieren mit dem Server abgeschlossen hat,

das Vereinbaren einer neuen Client-Verbindung mit dem Webserver für den neuen Client.

[0081] 11. Eine Webserver-Vorrichtung, umfassend:

einen Prozessor;

einen Speicher, der mit dem Prozessor wirkgekoppelt ist;

einen Netzwerkadapter, der zumindest einen Netzwerk-Port umfasst, der mit dem Prozessor wirkgekoppelt ist;

eine Speichervorrichtung, die darin gespeicherte Anweisungen umfasst, die eine Vielzahl von Softwaremodulen umfassen, die konfiguriert sind, um durch den Prozessor ausgeführt zu werden, um es der Webserver-Vorrichtung zu ermöglichen,

eine gemultiplexte TCP-Verbindung zwischen einem Proxyserver und dem Webserver aufzubauen, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungspfad, der eine Inter-Rechner-Verbindung umfasst, die mit einem Netzwerkport auf dem Netzwerkadapter gekoppelt ist, zu übertragen, worin eine Vielzahl von Clients über entsprechende TCP-Verbindungen mit dem Proxyserver verbunden ist, und

Client-Daten-Datenstromdaten, die für die Vielzahl von Clients bestimmt sind, als einen Datenstrom von gemultiplexten (MUX-) TCP-Paketen über die gemultiplexte TCP-Verbindung an den

Proxyserver zu senden, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die sequentielle Abfolgen von Bits von einem oder mehreren Client-Daten-Datenströmen umfassen, die Paket-Nutzlastdaten umfassen, die in TCP-Pakete eingekapselt werden sollen, um über den Proxyserver über die TCP-Verbindung zwischen dem Proxyserver und dem Client zu einem Client weitergeleitet zu werden, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente für zumindest zwei getrennte Client-Daten-Datenströme einkapselt.

[0082] 12. Webserver-Vorrichtung nach Punkt 11, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

über die gemultiplexte TCP-Verbindung einen Datenstrom von MUX-TCP-Paketen von dem Proxyserver zu empfangen, worin jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, und jedes Datensegment einen Teil eines Client-Daten-Datenstroms umfasst, der von einem TCP-Client empfangen wird;

Datensegmente in den MUX-TCP-Paketen zu entkapseln und die Client-Daten-Datenströme neu anzuordnen; und

Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen zu extrahieren.

[0083] 13. Webserver-Vorrichtung nach Punkt 12, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verbinden, in einem Speicherraum des Speichers einen entsprechenden Satz von Warteschlangen zuzuordnen, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange und zumindest eine Eingangswarteschlange umfasst.

[0084] 14. Webserver-Vorrichtung nach Punkt 12, worin jede TCP-Verbindung und die Warteschlangen in dem Satz von Warteschlangen, der der Verbindung zugeordnet ist, durch eine einzigartige Client-Identifikation identifiziert sind.

[0085] 15. Webserver-Vorrichtung nach einem der Punkte 11-15, worin die Vielzahl von Modulen einen Software-Verarbeitungsstapel umfasst, und worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

Client-Daten-Datenströme, die durch den Software-Verarbeitungsstapel generiert werden, in eine Vielzahl von Ausgangswarteschlangen einzureihen, worin Daten, die in den Client-Daten-Datenströmen enthalten sind, die in jede Ausgangswarteschlange eingereiht werden, für einen einzigen Client bestimmt sind;

einen Datenstrom von MUX-TCP-Paketen zu erzeugen, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, aus denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

[0086] 16. Webserver-Vorrichtung nach Punkt 15, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen einen entsprechenden Bandbreitenabschnitt der gemultiplexten TCP-Verbindung zuzuordnen; und

einen Warteschlangen-Auswahlalgorithmus zu implementieren, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den den TCP-Verbindungen zugeordneten Bandbreitenabschnitten Datensegmente gezogen werden sollen.

[0087] 17. Webserver-Vorrichtung nach Punkt 16, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um der Webserver-Vorrichtung zu ermöglichen:

zumindest eines aus Drosseln oder Unterbrechen von Client-Daten-Datenströmen, die über die gemultiplexte TCP-Verbindung übertragen werden und auf individueller Basis ausgewählten TCP-Verbindungen zugeordnet sind.

[0088] 18. Proxyserver-Vorrichtung, die konfiguriert ist, um als ein Proxyserver für einen Webserver zu arbeiten, umfassend:

einen Prozessor;

einen Speicher, der mit dem Prozessor wirkgekoppelt ist;

eine Vielzahl von Netzwerkports, die mit dem Prozessor wirkgekoppelt sind; und

eine Speichervorrichtung, die darin gespeicherte Anweisungen aufweist, die eine Vielzahl von Softwaremodulen umfassen, die konfiguriert sind, um durch den Prozessor ausgeführt zu werden, um es der Proxyserver-Vorrichtung zu ermöglichen,

an einem ersten Netzwerkport mit entsprechenden Clients eine Vielzahl von TCP-Verbindungen aufzubauen;

zwischen dem Proxyserver und dem Webserver eine gemultiplexte TCP-Verbindung aufzubauen, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungspfad, der eine Inter-Rechner-Verbindung umfasst, die an einen zweiten Netzwerkport auf der Proxyserver-Vorrichtung gekoppelt ist, zu übertragen;

über die Vielzahl von TCP-Verbindungen TCP-Pakete von der Vielzahl von Clients zu empfangen;

TCP-Pakete zu identifizieren, die über die Proxyserver-Vorrichtung an den Webserver geproxymt werden sollen;

Daten, die in TCP-Paketen enthalten sind, die über die Proxyserver-Vorrichtung geproxymt werden sollen und die über eine gegebene TCP-Verbindung empfangen werden, in einem entsprechenden Client-Daten-Datenstrom zu puffern;

Datensegmente einzukapseln, die in gemultiplexten (MUX-) TCP-Paketen sequentielle Abfolgen der Client-Daten-Datenströme umfassen, wobei zumindest ein Teil der MUX-TCP-Pakete Datensegmente einkapselt, die sequentielle Abfolgen von zumindest zwei getrennten Client-Daten-Datenströmen umfassen; und

die von dem zweiten Netzwerkport ausgehenden MUX-TCP-Pakete, die an den Webserver weitergeleitet werden sollen, über die gemultiplexte TCP-Verbindung zu senden.

[0089] 19. Proxyserver-Vorrichtung nach Punkt 18, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:

einen Datenstrom von MUX-TCP-Paketen über die gemultiplexte TCP-Verbindung von dem Webserver zu empfangen, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die in einem oder mehreren Client-Daten-Datenströmen sequentielle Abfolgen von Bits umfassen;

das eine oder die mehreren Datensegmente, die in den MUX-TCP-Paketen eingekapselt sind, zu entkapseln;

die Client-Daten-Datenströme neu anzuordnen;

eine TCP-Verbindung zu identifizieren, die einem gegebenen Client-Daten-Datenstrom zugeordnet ist;

Paket-Nutzlastdaten, die aus dem gegebenen Client-Daten-Datenstrom extrahiert werden, in TCP-Pakete einzukapseln, die für einen Client

bestimmt sind, der dem gegebenen Client-Daten-Datenstrom zugeordnet ist; und

die TCP-Pakete über die TCP-Verbindung für den Client an den bestimmten Client weiterzuleiten.

[0090] 20. Proxyserver-Vorrichtung nach Punkt 18, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verbinden, in einem Speicherraum des Speichers einen entsprechenden Satz von Warteschlangen zuzuordnen, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange umfasst, in der TCP-Pakete, die über die gemultiplexte TCP-Verbindung an den Webserver weitergeleitet werden sollen, gepuffert werden.

[0091] 21. Proxyserver-Vorrichtung nach Punkt 20, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen einen entsprechenden Bandbreitenabschnitt der gemultiplexten TCP-Verbindung zuzuordnen; und

einen Warteschlangen-Auswahlalgorithmus zu implementieren, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den den TCP-Verbindungen zugeordneten Bandbreitenabschnitten Datensegmente gezogen werden sollen.

[0092] 22. Proxyserver-Vorrichtung nach Punkt 18, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um der Proxyserver-Vorrichtung zu ermöglichen:

einen aus einem Drosselungsbefehl und einem Unterbrechungsbefehl von dem Webserver, der eine TCP-Verbindung identifiziert, zu empfangen; und als Antwort

eines aus Drosseln und Unterbrechen eines der TCP-Verbindungen zugeordneten Client-Daten-Datenstroms, der über die gemultiplexte TCP-Verbindung übertragen wird.

[0093] 23. Zumindest ein nichtflüchtiges maschinenlesbares Medium, das darauf gespeicherte Anweisungen aufweist, die eine Vielzahl von Softwaremodulen umfasst, die konfiguriert sind, um auf einem Proxyserver und einem Webserver ausgeführt zu werden, um es dem Proxyserver und dem Webserver zu ermöglichen:

an einem Proxyserver mit einer Vielzahl von Clients eine Vielzahl von TCP-Verbindungen aufzubauen, wobei jede TCP-Verbindung unter Verwendung eines TCP-Ports einen Client mit dem Proxyserver verbindet;

den Proxyserver über eine gemultiplexte TCP-Verbindung in Kommunikation mit dem Webserver zu koppeln;

eingehende TCP-Kommunikationseinheiten an dem Proxyserver zu empfangen, die von der Vielzahl von Clients stammen, wobei jede eingehende TCP-Kommunikation zumindest ein TCP-Paket umfasst, das Daten enthält, die eine HyperText-Übertragungsprotokoll- (HTTP-) Anfrage umfassen;

Paket-Nutzlastdaten zu extrahieren, die für eine gegebene TCP-Verbindung HTTP-Anfragen von den TCP-Paketen umfassen, und die extrahierten Paket-Nutzlastdaten einem Client-Daten-Datenstrom hinzuzufügen;

an dem Proxyserver Datensegmente, die sequentielle Abfolgen von Bits von Client-Daten-Datenströmen umfassen, in gemultiplexte (MUX-) TCP-Pakete einzukapseln, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente von mehreren Client-Daten-Datenströmen umfasst;

Übertragen der MUX-TCP-Pakete über die gemultiplexte TCP-Verbindung von dem Proxyserver an den Webserver; und

an dem Webserver

Datensegmente in den MUX-TCP-Paketen zu entkapseln und die Client-Daten-Datenströme neu anzuordnen; und

HTTP-Anfragen aus den neu angeordneten Client-Daten-Datenströmen zu extrahieren.

[0094] 24. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach Punkt 23, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

eine HTTP-Antwort auf eine HTTP-Anfrage in einen Client-Daten-Datenstrom einzubetten, der für den Client bestimmt ist, der die HTTP-Anfrage sendete;

ein oder mehrere Datensegmente, die dem Client-Daten-Datenstrom entsprechen, in eines oder mehrere MUX-TCP-Pakete einzukapseln;

das eine oder die mehreren MUX-TCP-Pakete über die gemultiplexte TCP-Verbindung von dem Webserver an den Proxyserver zu übertragen;

das eine oder die mehreren Datensegmente, die in dem einen oder mehreren MUX-TCP-Paketen eingekapselt sind, zu entkapseln;

den Client-Daten-Datenstrom neu anzuordnen;

eine TCP-Verbindung zu identifizieren, die dem Client-Daten-Datenstrom zugeordnet ist;

die HTTP-Antwort aus dem neu angeordneten Client-Daten-Datenstrom zu extrahieren und die HTTP-Antwort in eines oder mehrere TCP-Pakete, die für einen Client, der dem Client-Daten-Datenstrom zugeordnet ist, einzukapseln; und

das eine oder die mehreren TCP-Pakete über die TCP-Verbindung für den Client an den betreffenden Client weiterzuleiten.

[0095] 25. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach Punkt 24, worin jedes der einen oder der mehreren TCP-Pakete, in die die HTTP-Antwort eingebettet ist, eine Internetprotokoll- (IP-) Ursprungsadresse, die einer IP-Adresse für den Proxyserver entspricht, und eine IP-Zieladresse, die einer IP-Adresse für den betreffenden Client entspricht, aufweist.

[0096] 26. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-25, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

für jede TCP-Verbindung

der TCP-Verbindung auf dem Proxyserver zumindest eine Warteschlange zuzuordnen; und

Client-Daten-Datenströme zu puffern, die Paket-Nutzlastdaten umfassen, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in der zumindest einen Warteschlange, die der TCP-Verbindung zugeordnet ist, empfangen werden.

[0097] 27. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach Punkt 26, worin die zumindest eine Warteschlange auf dem Proxyserver, die jeder TCP-Verbindung zugeordnet ist, eine Ausgangswarteschlange umfasst, und worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

einen Client-Daten-Datenstrom zu puffern, der sequentielle Abfolgen von Paket-Nutzlastdaten umfasst, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in den Ausgangswarteschlangen auf dem Proxyserver übertragen werden;

einen Datenstrom von MUX-TCP-Paketen zu erzeugen, die über die gemultiplexte TCP-Verbindung an den Webserver übertragen werden sollen.

len, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, aus denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

[0098] 28. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-27, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

eine HTTP-Anfrage zu empfangen, die von einem neuen Client gesendet wird, um auf einen Uniform Resource Locator (URL) für eine Webseite zuzugreifen, auf die über den Webserver zugegriffen wird;

eine neue TCP-Verbindung zwischen dem neuen Client und dem Proxyserver aufzubauen; und

Warteschlangen auf jedem des Proxyserver und des Webserver zuzuordnen, um das Weiterleiten von HTTP-Anfragen, die an dem Proxyserver empfangen werden, von dem neuen Client an den Webserver über die gemultiplexte TCP-Verbindung zu unterstützen, und um HTTP-Antworten, die an dem Webserver erzeugt werden, über die gemultiplexte TCP-Verbindung zu dem Proxyserver und die neue TCP-Verbindung an den neuen Client weiterzuleiten.

[0099] 29. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-28, worin die gemultiplexte TCP-Verbindung eine Gesamtbandbreite aufweist, und worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

jeder TCP-Verbindung, die mit dem Proxyserver aufgebaut ist und für die TCP-Pakete über die gemultiplexte TCP-Verbindung weitergeleitet werden, einen entsprechenden Bandbreitenabschnitt der Gesamtbandbreite der gemultiplexten TCP-Verbindung zuzuordnen; und

es dem Bandbreitenabschnitt, der einer individuellen TCP-Verbindung zugeordnet wurde, zu ermöglichen, dynamisch angepasst zu werden.

[0100] 30. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-29, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es bei der Ausführung zumindest einem aus dem Webserver und dem Proxyserver zu ermöglichen, eine Datenübertragungsgeschwindigkeit für individuelle TCP-Verbindungen zu verwalten, die über die gemultiplexte TCP-Verbindung weitergeleitet werden.

[0101] 31. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-30, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es bei der Ausführung zumindest einem aus dem Webserver und dem Proxyserver zu ermöglichen, zu unterbrechen, dass der Verkehr, der einer ausgewählten TCP-Verbindung entspricht, über die gemultiplexte Verbindung weitergeleitet wird.

[0102] 32. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach Punkt 31, worin der Webserver eine Anfrage an den Proxyserver sendet, um den Verkehr für eine gegebene TCP-Verbindung zu unterbrechen, und worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

fortzusetzen, dass der eingehende TCP-Verkehr, der über die TCP-Verbindung von einem Client empfangen wird, gepuffert wird;

eine Anzeige zu erhalten, dass es dem Verkehr für die TCP-Verbindung gestattet ist, wieder aufgenommen zu werden; und

das Senden von Datensegmenten, die dem TCP-Verkehr entsprechen, der sich in dem Puffer angesammelt hat, von dem Proxyserver zu dem Webserver wiederaufzunehmen.

[0103] 33. Zumindest ein nichtflüchtiges maschinenlesbares Medium nach einem der Punkte 23-32, worin die eine Vielzahl von Softwaremodulen ferner konfiguriert ist, um es dem Proxyserver und dem Webserver bei der Ausführung zu ermöglichen:

Informationen von dem Webserver an den Proxyserver zu senden, die eine Maximalanzahl von gleichzeitigen Daten-Datenströmen, die gemultiplext und über die gemultiplexte TCP-Verbindung gesendet werden können, identifizieren;

eine Anzahl von Client-Verbindungen, die der Maximalanzahl der gleichzeitigen Daten-Datenströme entspricht, aufzubauen;

an dem Proxyserver Daten von einem neuen Client zu empfangen, der mit dem Webserver kommunizieren möchte;

die Daten von dem neuen Client zu puffern; und

als Antwort auf das Bestimmen, dass ein bestehender Client das Kommunizieren mit dem Server abgeschlossen hat,

das Vereinbaren einer neuen Client-Verbindung mit dem Webserver für den neuen Client.

[0104] 34. Webserver-Vorrichtung, umfassend:

einen Prozessor;

einen Speicher, der mit dem Prozessor wirkgekoppelt ist;

einen Netzwerkadapter, der zumindest einen Netzwerkport umfasst, der mit dem Prozessor wirkgekoppelt ist; und

Mittel zum

Aufbauen einer gemultiplexten TCP-Verbindung zwischen einem Proxyserver und dem Webserver, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungspfad, der eine Inter-Rechner-Verbindung umfasst, die an einen Netzwerkport auf dem Netzwerkadapter gekoppelt ist, zu übertragen, worin eine Vielzahl von Clients über entsprechende TCP-Verbindungen mit dem Proxyserver verbunden ist; und

Senden von Client-Daten-Datenstrom-Daten, die für die Vielzahl von Clients bestimmt sind, über die gemultiplexte TCP-Verbindung als ein Datenstrom von gemultiplexten (MUX-) TCP-Paketen an den Proxyserver, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die sequentielle Abfolgen von Bits von einem oder mehreren Client-Daten-Datenströmen umfassen, die Paket-Nutzlastdaten umfassen, die in TCP-Pakete eingekapselt werden sollen, um über den Proxyserver über die TCP-Verbindung zwischen dem Proxyserver und dem Client an einen Client weitergeleitet zu werden, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente für zumindest zwei getrennte Client-Daten-Datenströme einkapselt.

[0105] 35. Webserver-Vorrichtung nach Punkt 34, ferner umfassend Mittel zum

Empfangen eines Datenstroms von MUX-TCP-Paketen über die gemultiplexte TCP-Verbindung von dem Proxyserver, worin jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt und jedes Datensegment einen Teil eines von einem TCP-Client empfangenen Client-Daten-Datenstroms umfasst; Entkapseln von Datensegmenten in den MUX-TCP-Paketen und Neu-Anordnen der Client-Daten-Datenströme; und

Extrahieren von Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen.

[0106] 36. Webserver-Vorrichtung nach Punkt 35, ferner umfassend Mittel zum Zuordnen eines entsprechenden Satzes von Warteschlangen in einem Speicherraum des Speichers für jede der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verbinden, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange und zumindest eine Eingangswarteschlange umfasst.

[0107] 37. Webserver-Vorrichtung nach Punkt 35, worin jede TCP-Verbindung und die Warteschlangen in dem Satz von Warteschlangen, der der Verbindung

zugeordnet ist, durch eine einzigartige Client-Identifikation identifiziert sind.

[0108] 38. Webserver-Vorrichtung nach einem der Punkte 34-38, ferner umfassend Mittel zum:

Einreihen von Client-Daten-Datenströmen, die durch den Software-Verarbeitungsstapel erzeugt werden in eine Vielzahl von Ausgangswarteschlangen, worin Daten, die in den Client-Daten-Datenströmen enthalten sind, die in jede Ausgangswarteschlange eingereiht sind, für einen einzigen Client bestimmt sind;

Erzeugen eines Datenstroms von MUX-TCP-Paketen, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, aus denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

[0109] 39. Webserver-Vorrichtung nach Punkt 38, ferner umfassend Mittel zum:

Zuordnen eines entsprechenden Bandbreitenabschnitts der gemultiplexten TCP-Verbindung zu jeder der TCP-Verbindungen; und

Implementieren eines Warteschlangen-Auswahlalgorithmus, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den den TCP-Verbindungen zugeordneten Bandbreitenabschnitten Datensegmente gezogen werden sollen.

[0110] 40. Webserver-Vorrichtung nach Punkt 39, ferner umfassend Mittel zum:

zumindest eines aus Drosseln oder Unterbrechen von Client-Daten-Datenströmen, die über die gemultiplexte TCP-Verbindung übertragen werden und auf individueller Basis ausgewählten TCP-Verbindungen zugeordnet sind.

[0111] 41. Proxyserver-Vorrichtung, die konfiguriert ist, um als ein Proxyserver für einen Webserver zu arbeiten, umfassend:

einen Prozessor;

einen Speicher, der mit dem Prozessor wirkgekoppelt ist;

eine Vielzahl von Netzwerkports, die mit dem Prozessor wirkgekoppelt sind; und Mittel zum

Aufbauen einer Vielzahl von TCP-Verbindungen mit entsprechenden Clients an einem ersten Netzwerkport;

Aufbauen einer gemultiplexten TCP-Verbindung zwischen dem Proxyserver und dem Webserver, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungs-

pfad, der eine Inter-Rechner-Verbindung umfasst, die an einen zweiten Netzwerkport auf der Proxyserver-Vorrichtung gekoppelt ist, zu übertragen;

Empfangen von TCP-Paketen von der Vielzahl von Clients über die Vielzahl von TCP-Verbindungen;

Identifizieren von TCP-Paketen, die über die Proxyserver-Vorrichtung an den Webserver geproxymt werden sollen;

Puffern von Daten, die in TCP-Paketen, die über die Proxyserver-Vorrichtung geproxymt werden sollen und die über eine gegebene TCP-Verbindung empfangen werden, in einem entsprechenden Client-Daten-Datenstrom;

Einkapseln von Datensegmenten, die sequentielle Abfolgen der Client-Daten-Datenströme umfassen, in gemultiplexte (MUX-) TCP-Pakete, wobei zumindest ein Teil der MUX-TCP-Pakete Datensegmente einkapselt, die sequentielle Abfolgen von zumindest zwei getrennten Client-Daten-Datenströmen umfassen; und

Senden der von dem zweiten Netzwerkport ausgehenden MUX-TCP-Pakete, um über die gemultiplexte TCP-Verbindung an den Webserver weitergeleitet zu werden.

[0112] 42. Proxyserver-Vorrichtung nach Punkt 41, ferner umfassend Mittel zum:

Empfangen eines Datenstroms von MUX-TCP-Paketen über die gemultiplexte TCP-Verbindung von dem Webserver, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die sequentielle Abfolgen von Bits in einem oder mehreren Client-Daten-Datenströmen umfassen;

Entkapseln des einen oder der mehreren Datensegmente, die in den MUX-TCP-Paketen einkapselt sind;

Neuanordnen der Client-Daten-Datenströme;

Identifizieren einer TCP-Verbindung, die einem gegebenen Client-Daten-Datenstrom zugeordnet ist;

Einkapseln von Paket-Nutzlastdaten, die aus dem gegebenen Client-Daten-Datenstrom extrahiert werden, in TCP-Pakete, die für einen Client bestimmt sind, der dem gegebenen Client-Daten-Datenstrom zugeordnet ist; und

Weiterleiten der TCP-Pakete an den betreffenden Client über die TCP-Verbindung für den Client.

[0113] 43. Proxyserver-Vorrichtung nach Punkt 41 oder 42, ferner umfassend Mittel zum Zuordnen eines entsprechenden Satzes von Warteschlangen in ei-

nem Speicherraum der Speicherkapazität für jede der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verwenden, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange umfasst, in der TCP-Pakete, die über die gemultiplexte TCP-Verbindung an den Webserver weitergeleitet werden sollen, gepuffert werden sollen.

[0114] 44. Proxyserver-Vorrichtung nach Punkt 43, ferner umfassend Mittel zum:

Zuordnen eines entsprechenden Bandbreitenabschnitts der gemultiplexten TCP-Verbindung für jede der TCP-Verbindungen; und

Implementieren eines Warteschlangen-Auswahlalgorithmus, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den Bandbreitenabschnitten, die den TCP-Verbindungen zugeordnet sind, Datensegmente gezogen werden sollen.

[0115] 45. Proxyserver-Vorrichtung nach einem der Punkte 41-44, ferner umfassend Mittel zum:

Empfangen von einem aus einem Drosselungsbefehl und einem Unterbrechungsbefehl von dem Webserver, der eine TCP-Verbindung identifiziert; und als Antwort

eines aus Drosseln und Unterbrechen eines der TCP-Verbindung zugeordneten Client-Daten-Datenstroms, der über die gemultiplexte TCP-Verbindung übertragen wird.

[0116] Wie oben besprochen können verschiedene Aspekte der Ausführungsformen hierin durch entsprechende Software- und/oder Firmware-Komponenten und -Anwendungen ermöglicht werden, wie Software und/oder Firmware, die durch einen Serverprozessor oder dergleichen ausgeführt wird. Daher können Ausführungsformen der vorliegenden Erfindung als ein Softwareprogramm, Softwaremodule, Firmware und/oder verteilte Software, das/die auf einer Art von Prozessor, Verarbeitungskern oder eingebetteter Logik oder auf einer virtuellen Maschine, die auf einem Prozessor oder Kern läuft, ausgeführt wird oder anders auf oder in einem computerlesbaren oder maschinenlesbaren nichtflüchtigen Speichermedium implementiert oder realisiert ist, verwendet werden, oder verwendet werden, um diese(s) zu unterstützen. Ein computerlesbares oder maschinenlesbares nichtflüchtiges Speichermedium umfasst einen beliebigen Mechanismus zum Speichern oder Übertragen von Informationen in einer Form, die durch eine Maschine (z.B. einen Computer) gelesen werden kann. Beispielsweise umfasst ein computerlesbares oder maschinenlesbares nichtflüchtiges Speichermedium einen beliebigen Mechanismus, der Informationen in einer Form bereitstellt

(d.h. speichert und/oder überträgt), auf die durch einen Computer oder eine Rechnermaschine (z.B. Rechnervorrichtung, elektronisches System etc.) zugegriffen werden kann, wie ein aufzeichnungsfähiges/nicht aufzeichnungsfähiges Medium (z.B. Nur-Lese-Speicher (ROM), Direktzugriffsspeicher (RAM), Magnetscheiben-Speichermedium, optisches Speichermedium, Flashspeichervorrichtungen etc.). Der Inhalt kann direkt ausführbar („Objekt-“ oder „ausführbare“ Form), ein Source-Code oder ein Differenz-Code („Delta-“ oder „Patch-“ Code) sein. Ein computerlesbares oder maschinenlesbares nichtflüchtiges Speichermedium kann auch einen Speicher oder eine Datenbank umfassen, von dem/der ein Inhalt heruntergeladen werden kann. Das computerlesbare oder maschinenlesbare nichtflüchtige Speichermedium kann auch eine Vorrichtung oder ein Produkt umfassen, das einen Inhalt aufweist, der zu einem Verkaufs- oder Lieferungszeitpunkt darauf gespeichert ist. Daher kann das Liefern einer Vorrichtung mit gespeichertem Inhalt oder das Anbieten eines Inhalts zum Herunterladen über ein Kommunikationsmedium als Bereitstellen eines Herstellungartikels verstanden werden, der ein computerlesbares oder maschinenlesbares nichtflüchtiges Speichermedium mit einem solchen hierin beschriebenen Inhalt umfasst.

[0117] Verschiedene Komponenten, die hierin oben als Prozesse, Server oder Werkzeuge beschrieben werden, können ein Mittel zum Ausführen der beschriebenen Funktionen sein. Die Vorgänge und Funktionen, die durch verschiedene hierin beschriebene Komponenten ausgeführt werden können durch Software, die auf einem Verarbeitungselement läuft, über eingebettete Hardware oder dergleichen oder eine beliebige Kombination von Hardware und Software implementiert sein. Solche Komponenten können als Softwaremodule, Hardwaremodule, Spezial-Hardware (z.B. anwendungsspezifische Hardware, ASICs, DSPs etc.), eingebettete Steuerungen, festverdrahtete Schaltungen, Hardware-Logik etc. implementiert sein. Ein Software-Inhalt (z.B. Daten, Anweisungen, Konfigurationsinformationen etc.) kann durch einen Herstellungartikel bereitgestellt sein, der ein computerlesbares oder maschinenlesbares nichtflüchtiges Speichermedium umfasst, das einen Inhalt bereitstellt, der Anweisungen darstellt, die ausgeführt werden können. Der Inhalt kann darin resultieren, dass ein Computer verschiedene hierin beschriebene Funktionen/Vorgänge ausführt.

[0118] Obwohl manche Ausführungsformen mit Verweis auf bestimmte Implementierungen beschrieben wurden, sind gemäß manchen Ausführungsformen andere Implementierungen möglich. Zusätzlich dazu muss die Anordnung und/oder Reihenfolge von Elementen oder anderen Merkmalen, die in den Zeichnungen veranschaulicht und/oder hierin beschrieben sind, nicht auf die bestimmte veranschaulichte und beschriebene Weise angeordnet sein. Viele andere

Anordnungen sind gemäß manchen Ausführungsformen möglich.

[0119] In jedem System, das in einer Figur gezeigt ist, können die Elemente in manchen Fällen jeweils eine gleiche Referenzzahl oder eine andere Referenzzahl aufweisen, um anzudeuten, dass die dargestellten Elemente unterschiedlich und/oder ähnlich sein könnten. Jedoch kann ein Element flexibel genug sein, um unterschiedliche Implementierungen aufzuweisen und mit manchen oder allen der hierin gezeigten oder beschriebenen Systeme zu arbeiten. Die verschiedenen, in den Figuren gezeigten Elemente können die gleichen oder unterschiedliche sein. Auf welches als ein erstes Element verwiesen wird und welches ein zweites Element genannt wird, ist willkürlich.

[0120] In der Beschreibung und den Ansprüchen können die Begriffe „gekoppelt“ und „verbunden“ gemeinsam mit ihren Ableitungen verwendet werden. Es sollte verständlich sein, dass diese Begriffe nicht als Synonyme füreinander beabsichtigt sind. Stattdessen kann „verbunden“ in bestimmten Ausführungsformen verwendet werden, um anzuzeigen, dass zwei oder mehr Elemente in direktem physikalischen oder elektrischen Kontakt miteinander stehen. „Gekoppelt“ kann bedeuten, dass zwei oder mehr Elemente in direktem physikalischen oder elektrischen Kontakt miteinander stehen. Jedoch kann „gekoppelt“ auch bedeuten, dass zwei oder mehr Elemente nicht in direktem Kontakt miteinander stehen, aber dennoch miteinander zusammenwirken oder wechselwirken.

[0121] Eine Ausführungsform ist eine Implementierung oder ein Beispiel der Erfindungen. Ein Verweis auf „eine Ausführungsform“, „manche Ausführungsformen“ oder „andere Ausführungsformen“ in der Patentschrift bedeutet, dass ein(e) bestimmtes Merkmal, Struktur oder Eigenschaft, das/die in Verbindung mit den Ausführungsformen beschrieben ist, in zumindest manchen, nicht aber notwendigerweise in allen, Ausführungsformen der Erfindungen umfasst ist. Die verschiedenen Vorkommen von „einer Ausführungsform“ oder „manchen Ausführungsformen“ verweisen nicht notwendigerweise alle auf dieselben Ausführungsformen.

[0122] Nicht alle Komponenten, Merkmale, Strukturen, Eigenschaften etc., die hierin beschrieben und veranschaulicht sind, müssen in einer bestimmten Ausführungsform oder Ausführungsformen umfasst sein. Wenn die Patentschrift besagt, dass eine Komponente, ein Merkmal, eine Struktur oder eine Eigenschaft beispielsweise umfasst sein „kann“ oder „könnte“, ist es nicht erforderlich, dass diese(s) bestimmte Komponente, Merkmal, Struktur oder Eigenschaft umfasst ist. Wenn die Patentschrift oder der Anspruch auf „ein“ Element verweist, bedeutet dies

nicht, dass es nur eines des Elements gibt. Wenn die Patentschrift oder der Anspruch auf „ein zusätzliches“ Element verweist, schließt dies nicht aus, dass es mehr als eines des zusätzlichen Elements gibt.

[0123] Kursiv gesetzte Buchstaben wie „n“ in der vorangegangenen ausführlichen Beschreibung werden verwendet, um eine ganze Zahl darzustellen, und die Verwendung eines bestimmten Buchstabens beschränkt sich nicht auf bestimmte Ausführungsformen. Ferner kann derselbe Buchstabe in getrennten Ansprüchen verwendet werden, um getrennte ganze Zahlen darzustellen, oder es können unterschiedliche Buchstaben verwendet werden. Zusätzlich dazu kann die Verwendung eines bestimmten Buchstabens in der ausführlichen Beschreibung dem Buchstaben entsprechen, der in einem Anspruch verwendet wird, der in der ausführlichen Beschreibung denselben Gegenstand betrifft.

[0124] Wie hierin verwendet kann eine Liste von Artikeln, die durch den Ausdruck „zumindest einer/eine/eines aus“ verbunden ist, eine beliebige Kombination der angeführten Begriffe bedeuten. Beispielsweise kann die Phrase „zumindest eines aus A, B oder C“ A; B; C; A und B; A und C; B und C; oder A, B und C bedeuten.

[0125] Die obige Beschreibung von veranschaulichten Ausführungsformen der Erfindung, einschließlich dessen, das in der Zusammenfassung beschrieben ist, ist nicht darauf ausgelegt, erschöpfend zu sein oder die Erfindung auf die präzisen offenbarten Formen zu beschränken. Wenngleich spezifische Ausführungsformen und Beispiele der Erfindung hierin zu veranschaulichenden Zwecken beschrieben sind, sind verschiedene äquivalente Modifikationen innerhalb des Schutzzumfangs der Erfindung möglich, wie Fachleute auf dem relevanten Gebiet erkennen werden.

[0126] Diese Modifikationen können im Lichte der obigen ausführlichen Beschreibung an der Erfindung durchgeführt werden. Die Begriffe, die in den folgenden Ansprüchen verwendet werden, sollen nicht so ausgelegt werden, dass sie die Erfindung auf die spezifischen Ausführungsformen, die in der Patentschrift und den Zeichnungen offenbart sind, beschränken. Stattdessen soll der Schutzzumfang der Erfindung gänzlich durch die folgenden Ansprüche bestimmt werden, die in Übereinstimmung mit den etablierten Doktrinen der Auslegung von Ansprüchen ausgelegt werden sollen.

Patentansprüche

1. Verfahren, umfassend:

das Aufbauen einer Vielzahl von TCP-Verbindungen mit einer Vielzahl von Clients an einem Proxyserver, wobei jede TCP-Verbindung unter Verwendung eines

TCP-Ports einen Client mit dem Proxyserver verbindet;

das Koppeln des Proxyservers in Kommunikation mit einem Webserver über eine gemultiplexte TCP-Verbindung;

das Empfangen von eingehenden TCP-Kommunikationseinheiten an dem Proxyserver, die von der Vielzahl von Clients stammen, wobei jede eingehende TCP-Kommunikation zumindest ein TCP-Paket umfasst, das Paket-Nutzlastdaten umfasst;

das Puffern der Paket-Nutzlastdaten, die in TCP-Paketen enthalten sind, die über eine gegebene TCP-Verbindung in einem entsprechenden Client-Daten-Datenstrom empfangen werden;

das Einkapseln von Datensegmenten, die in den Client-Daten-Datenströmen sequentielle Abfolgen von Bits umfassen, in gemultiplexte (MUX-) TCP-Pakete an dem Proxyserver, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente von zumindest zwei getrennten Client-Daten-Datenströmen einkapselt;

das Übertragen der MUX-TCP-Pakete von dem Proxyserver über die gemultiplexte TCP-Verbindung an den Webserver; und

an dem Webserver

das Entkapseln von Datensegmenten in den MUX-TCP-Paketen und das Neuordnen der Client-Daten-Datenströme; und

das Extrahieren von Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen.

2. Verfahren nach Anspruch 1, worin das eine oder die mehreren TCP-Pakete, die über eine erste TCP-Verbindung empfangen werden, eine Hypertext-Übertragungsprotokoll-(HTTP-) Anfrage umfassen, die von einem ersten Client gesendet wird, wobei das Verfahren ferner das Zurückgeben einer HTTP-Antwort an den ersten Client umfasst, indem: die HTTP-Antwort in einen Client-Daten-Datenstrom eingebettet wird, der für den ersten Client bestimmt ist,

Datensegmente, die sequentielle Abfolgen von Bits umfassen, die dem Client-Daten-Datenstrom entsprechen, in eines oder mehrere MUX-TCP-Pakete eingekapselt werden;

das eine oder die mehreren MUX-TCP-Pakete über die gemultiplexte TCP-Verbindung von dem Webserver an den Proxyserver übertragen werden;

die Datensegmente von dem MUX-TCP-Paket entkapselt werden;

der Client-Daten-Datenstrom neu angeordnet wird; und

die HTTP-Antwort extrahiert wird und die HTTP-Antwort in eines oder mehrere TCP-Pakete eingekapselt wird, die über die erste TCP-Verbindung von dem Proxyserver an den betreffenden Client weitergeleitet werden.

3. Verfahren nach Anspruch 1 oder 2, ferner umfassend:

für jede TCP-Verbindung

das Zuordnen von zumindest einer Warteschlange auf dem Proxyserver für die TCP-Verbindung; und
das Puffern eines Client-Daten-Datenstroms, der Paket-Nutzlastdaten umfasst, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in der zumindest einen Warteschlange empfangen werden, die der TCP-Verbindung zugeordnet ist.

4. Verfahren nach Anspruch 3, worin die zumindest eine Warteschlange auf dem Proxyserver, die jeder TCP-Verbindung zugeordnet ist, eine Ausgangswarteschlange umfasst, die ferner umfasst:
das Puffern eines Client-Daten-Datenstroms, der sequentielle Abfolgen von Paket-Nutzlastdaten umfasst, die in TCP-Paketen enthalten sind, die über die TCP-Verbindung in den Ausgangswarteschlangen auf dem Proxyserver übertragen werden; worin jede Ausgangswarteschlange einer entsprechenden TCP-Verbindung zugeordnet ist;
das Erzeugen eines Datenstroms von MUX-TCP-Paketen, die über die gemultiplexte TCP-Verbindung an den Webserver übertragen werden sollen, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, von denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

5. Verfahren nach einem der vorangegangenen Ansprüche, ferner umfassend:
das Empfangen einer HyperText-Übertragungsprotokoll- (HTTP-) Anfrage, die von einem neuen Client gesendet wird, um auf einen Uniform Resource Locator (URL) für eine Webseite zuzugreifen, auf die über den Webserver zugegriffen wird;
das Aufbauen einer neuen TCP-Verbindung zwischen dem neuen Client und dem Proxyserver; und
das Zuordnen von Warteschlangen auf jedem des Proxyserver und des Webserver, um das Weiterleiten von HTTP-Anfragen, die an dem Proxyserver empfangen werden, von dem neuen Client an den Webserver über die gemultiplexte TCP-Verbindung zu unterstützen, und um HTTP-Antworten, die an dem Webserver erzeugt werden, über die gemultiplexte TCP-Verbindung zu dem Proxyserver und die neue TCP-Verbindung an den neuen Client weiterzuleiten.

6. Verfahren nach einem der vorangegangenen Ansprüche, worin die gemultiplexte TCP-Verbindung eine Gesamtbandbreite aufweist, wobei das Verfahren ferner umfasst:
das Zuordnen eines entsprechenden Bandbreitenabschnitts der Gesamtbandbreite der gemultiplexten TCP-Verbindung zu jeder TCP-Verbindung, die mit dem Proxyserver aufgebaut ist, und für die TCP-Pakete über die gemultiplexte TCP-Verbindung weitergeleitet werden sollen;

das Ermöglichen, dass der einer individuellen TCP-Verbindung zugeordnete Bandbreitenabschnitt dynamisch angepasst wird.

7. Verfahren nach einem der vorangegangenen Ansprüche, ferner umfassend:
das Ermöglichen, dass zumindest einer aus dem Webserver und dem Proxyserver eine Datenübertragungsgeschwindigkeit für individuelle TCP-Verbindungen verwaltet, die über die gemultiplexte TCP-Verbindung weitergeleitet werden.

8. Verfahren nach Anspruch 7, ferner umfassend das Ermöglichen, dass zumindest einer aus dem Webserver und dem Proxyserver unterbricht, dass der Verkehr, der einer ausgewählten TCP-Verbindung entspricht, über die gemultiplexte TCP-Verbindung weitergeleitet wird.

9. Verfahren nach Anspruch 8, worin der Webserver eine Anfrage an den Proxyserver sendet, um den Verkehr für eine gegebene TCP-Verbindung zu unterbrechen, wobei das Verfahren ferner umfasst:
das Fortsetzen des Pufferns von eingehendem TCP-Verkehr, der über die TCP-Verbindung von einem Client empfangen wird;
das Empfangen einer Anzeige, dass es dem Verkehr für die TCP-Verbindung gestattet ist, wieder aufgenommen zu werden; und
das Wiederaufnehmen des Sendens von Datensegmenten, die dem TCP-Verkehr entsprechen, der sich in dem Puffer angesammelt hat, von dem Proxyserver zu dem Webserver.

10. Verfahren nach einem der vorangegangenen Ansprüche, ferner umfassend:
das Senden von Informationen von dem Webserver an den Proxyserver, die eine Maximalanzahl von gleichzeitigen Daten-Datenströmen identifizieren, die gemultiplext und über die gemultiplexte TCP-Verbindung gesendet werden können;
das Aufbauen einer Anzahl von Client-Verbindungen, die der Maximalanzahl von gleichzeitigen Daten-Datenströmen entspricht;
das Empfangen von Daten von einem neuen Client, der mit dem Webserver kommunizieren möchte, an dem Proxyserver;
das Puffern der Daten von dem neuen Client; und
als Antwort auf das Bestimmen, dass ein bestehender Client das Kommunizieren mit dem Server abgeschlossen hat,
das Vereinbaren einer neuen Client-Verbindung mit dem Webserver für den neuen Client.

11. Zumindest ein nichtflüchtiges maschinenlesbares Medium, das darauf gespeicherte Anweisungen aufweist, die eine Vielzahl von Softwaremodulen umfassen, die konfiguriert sind, um auf einem Proxyserver und einem Webserver ausgeführt zu werden, um es dem Proxyserver und dem Webserver zu ermöglichen,

chen, das Verfahren nach einem der vorangegangenen Ansprüche durchzuführen.

12. Webserver-Vorrichtung, umfassend:

einen Prozessor;
einen Speicher, der mit dem Prozessor wirkgekoppelt ist;
einen Netzwerkadapter, der zumindest einen Netzwerk-Port umfasst, der mit dem Prozessor wirkgekoppelt ist;
eine Speichervorrichtung, die darin gespeicherte Anweisungen umfasst, die eine Vielzahl von Softwaremodulen umfassen, die konfiguriert sind, um durch den Prozessor ausgeführt zu werden, um es der Webserver-Vorrichtung zu ermöglichen, eine gemultiplexte TCP-Verbindung zwischen einem Proxyserver und dem Webserver aufzubauen, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungspfad, der eine Inter-Rechner-Verbindung umfasst, die an einen Netzwerkport auf dem Netzwerkadapter gekoppelt ist, zu übertragen, worin eine Vielzahl von Clients über entsprechende TCP-Verbindungen mit dem Proxyserver verbunden ist; und

Client-Daten-Datenstrom-Daten, die für die Vielzahl von Clients bestimmt sind, als einen Datenstrom von gemultiplexten (MUX-) TCP-Paketen über die gemultiplexte TCP-Verbindung an den Proxyserver zu senden, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die sequentielle Abfolgen von Bits von einem oder mehreren Client-Daten-Datenströmen umfassen, die Paket-Nutzlastdaten umfassen, die in TCP-Pakete eingekapselt werden sollen, um über den Proxyserver über die TCP-Verbindung zwischen dem Proxyserver und dem Client zu einem Client weitergeleitet zu werden, worin zumindest ein Teil der MUX-TCP-Pakete Datensegmente für zumindest zwei getrennte Client-Daten-Datenströme einkapselt.

13. Webserver-Vorrichtung nach Anspruch 12, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

über die gemultiplexte TCP-Verbindung einen Datenstrom von MUX-TCP-Paketen von dem Proxyserver zu empfangen, worin jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, und jedes Datensegment einen Teil eines Client-Daten-Datenstroms umfasst, der von einem TCP-Client empfangen wird;
Datensegmente in den MUX-TCP-Paketen zu entkapseln und die Client-Daten-Datenströme neu anzuordnen; und
Paket-Nutzlastdaten aus den neu angeordneten Client-Daten-Datenströmen zu extrahieren.

14. Webserver-Vorrichtung nach Anspruch 13, worin die Vielzahl von Softwaremodulen ferner kon-

figuriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verbinden, in einem Speicherraum des Speichers einen entsprechenden Satz von Warteschlangen zuzuordnen, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange und zumindest eine Eingangswarteschlange umfasst.

15. Webserver-Vorrichtung nach Anspruch 13, worin jede TCP-Verbindung und die Warteschlangen in dem Satz von Warteschlangen, der der Verbindung zugeordnet ist, durch eine einzigartige Client-Identifikation identifiziert sind.

16. Webserver-Vorrichtung nach einem der Ansprüche 12-15, worin die Vielzahl von Modulen einen Software-Verarbeitungsstapel umfasst, und worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

Client-Daten-Datenströme, die durch den Software-Verarbeitungsstapel generiert werden, in eine Vielzahl von Ausgangswarteschlangen einzureihen, worin Daten, die in den Client-Daten-Datenströmen enthalten sind, die in jede Ausgangswarteschlange eingereiht werden, für einen einzigen Client bestimmt sind;

einen Datenstrom von MUX-TCP-Paketen zu erzeugen, indem ein Warteschlangenwähler verwendet wird, um Ausgangswarteschlangen auszuwählen, aus denen Datensegmente gezogen werden sollen, um in entsprechende MUX-TCP-Pakete in dem Datenstrom von MUX-TCP-Paketen eingekapselt zu werden.

17. Webserver-Vorrichtung nach Anspruch 16, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen:

jeder der TCP-Verbindungen einen entsprechenden Bandbreitenabschnitt der gemultiplexten TCP-Verbindung zuzuordnen; und

einen Warteschlangen-Auswahlalgorithmus zu implementieren, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den den TCP-Verbindungen zugeordneten Bandbreitenabschnitten Datensegmente gezogen werden sollen.

18. Webserver-Vorrichtung nach Anspruch 17, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um der Webserver-Vorrichtung zu ermöglichen:

zumindest eines aus Drosseln oder Unterbrechen von Client-Daten-Datenströmen, die über die gemultiplexte TCP-Verbindung übertragen werden und auf individueller Basis ausgewählten TCP-Verbindungen zugeordnet sind.

19. Webserver-Vorrichtung nach einem der Ansprüche 12-18, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Webserver-Vorrichtung zu ermöglichen, eine Vielzahl von gemultiplexten TCP-Verbindungssteuerungsbefehlen an den Proxyserver zu senden und eine Vielzahl von Befehlsantworten von dem Proxyserver zu erhalten, worin die Steuerungsbefehle verwendet werden, um die gemultiplexte TCP-Verbindung zu verwalten.

20. Proxyserver-Vorrichtung, die konfiguriert ist, um als ein Proxyserver für einen Webserver zu arbeiten, umfassend:
 einen Prozessor;
 einen Speicher, der mit dem Prozessor wirkgekoppelt ist;
 eine Vielzahl von Netzwerkports, die mit dem Prozessor wirkgekoppelt sind; und
 eine Speichervorrichtung, die darin gespeicherte Anweisungen aufweist, die eine Vielzahl von Softwaremodulen umfassen, die konfiguriert sind, um durch den Prozessor ausgeführt zu werden, um es der Proxyserver-Vorrichtung zu ermöglichen,
 an einem ersten Netzwerkport mit entsprechenden Clients eine Vielzahl von TCP-Verbindungen aufzubauen;
 zwischen dem Proxyserver und dem Webserver eine gemultiplexte TCP-Verbindung aufzubauen, wobei die gemultiplexte TCP-Verbindung konfiguriert ist, um Daten über einen Weiterleitungspfad, der eine Inter-Rechner-Verbindung umfasst, die an einen zweiten Netzwerkport auf der Proxyserver-Vorrichtung gekoppelt ist, zu übertragen;
 über die Vielzahl von TCP-Verbindungen TCP-Pakete von der Vielzahl von Clients zu empfangen;
 TCP-Pakete zu identifizieren, die über die Proxyserver-Vorrichtung an den Webserver geproxyt werden sollen;
 Daten, die in TCP-Paketen enthalten sind, die über die Proxyserver-Vorrichtung geproxyt werden sollen und die über eine gegebene TCP-Verbindung empfangen werden, in einem entsprechenden Client-Daten-Datenstrom zu puffern;
 Datensegmente einzukapseln, die in gemultiplexten (MUX-) TCP-Paketen sequentielle Abfolgen der Client-Daten-Datenströme umfassen, wobei zumindest ein Teil der MUX-TCP-Pakete Datensegmente einkapselt, die sequentielle Abfolgen von zumindest zwei getrennten Client-Daten-Datenströmen umfassen; und
 die von dem zweiten Netzwerkport ausgehenden MUX-TCP-Pakete, die an den Webserver weitergeleitet werden sollen, über die gemultiplexte TCP-Verbindung zu senden.

21. Proxyserver-Vorrichtung nach Anspruch 20, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:

einen Datenstrom von MUX-TCP-Paketen über die gemultiplexte TCP-Verbindung von dem Webserver zu empfangen, wobei jedes MUX-TCP-Paket ein oder mehrere Datensegmente einkapselt, die in einem oder mehreren Client-Daten-Datenströmen sequentielle Abfolgen von Bits umfassen;
 das eine oder die mehreren Datensegmente, die in den MUX-TCP-Paketen eingekapselt sind, zu entkapseln;
 die Client-Daten-Datenströme neu anzuordnen;
 eine TCP-Verbindung zu identifizieren, die einem gegebenen Client-Daten-Datenstrom zugeordnet ist;
 Paket-Nutzlastdaten, die aus dem gegebenen Client-Daten-Datenstrom extrahiert werden, in TCP-Pakete einzukapseln, die für einen Client bestimmt sind, der dem gegebenen Client-Daten-Datenstrom zugeordnet ist; und
 die TCP-Pakete über die TCP-Verbindung für den Client an den bestimmten Client weiterzuleiten.

22. Proxyserver-Vorrichtung nach Anspruch 21, worin die Paket-Nutzlastdaten HyperText-Übertragungsprotokoll- (HTTP-) Antworten umfassen.

23. Proxyserver-Vorrichtung nach Anspruch 21, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:
 jeder der TCP-Verbindungen, die verwendet werden, um die Vielzahl von Clients mit dem Proxyserver zu verbinden, in einem Speicherraum des Speichers einen entsprechenden Satz von Warteschlangen zuzuordnen, worin der entsprechende Satz von Warteschlangen für eine gegebene TCP-Verbindung zumindest eine Ausgangswarteschlange umfasst, in der TCP-Pakete, die über die gemultiplexte TCP-Verbindung an den Webserver weitergeleitet werden sollen, gepuffert werden.

24. Proxyserver-Vorrichtung nach Anspruch 23, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um es der Proxyserver-Vorrichtung zu ermöglichen:
 jeder der TCP-Verbindungen einen entsprechenden Bandbreitenabschnitt der gemultiplexten TCP-Verbindung zuzuordnen; und
 einen Warteschlangen-Auswahlalgorithmus zu implementieren, der Ausgangswarteschlangen auswählt, aus denen in Übereinstimmung mit den den TCP-Verbindungen zugeordneten Bandbreitenabschnitten Datensegmente gezogen werden sollen.

25. Proxyserver-Vorrichtung nach einem der Ansprüche 20-24, worin die Vielzahl von Softwaremodulen ferner konfiguriert ist, um der Proxyserver-Vorrichtung zu ermöglichen:
 einen aus einem Drosselungsbefehl und einem Unterbrechungsbefehl von dem Webserver, der eine TCP-Verbindung identifiziert, zu empfangen; und als Antwort

eines aus Drosseln und Unterbrechen eines der TCP-Verbindung zugeordneten Client-Daten-Datenstroms, der über die gemultiplexte TCP-Verbindung übertragen wird.

Es folgen 15 Seiten Zeichnungen

Anhängende Zeichnungen

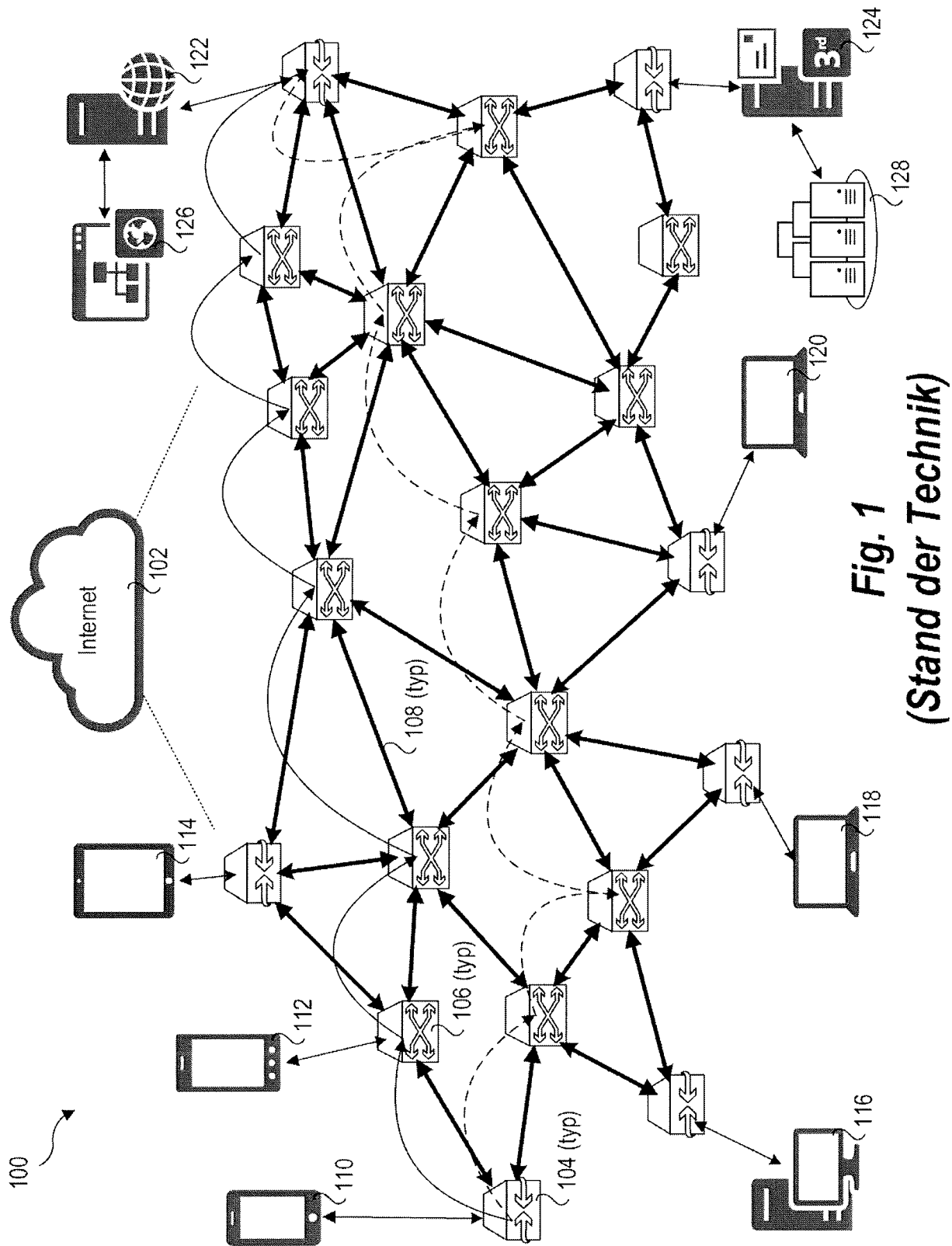


Fig. 1
(Stand der Technik)

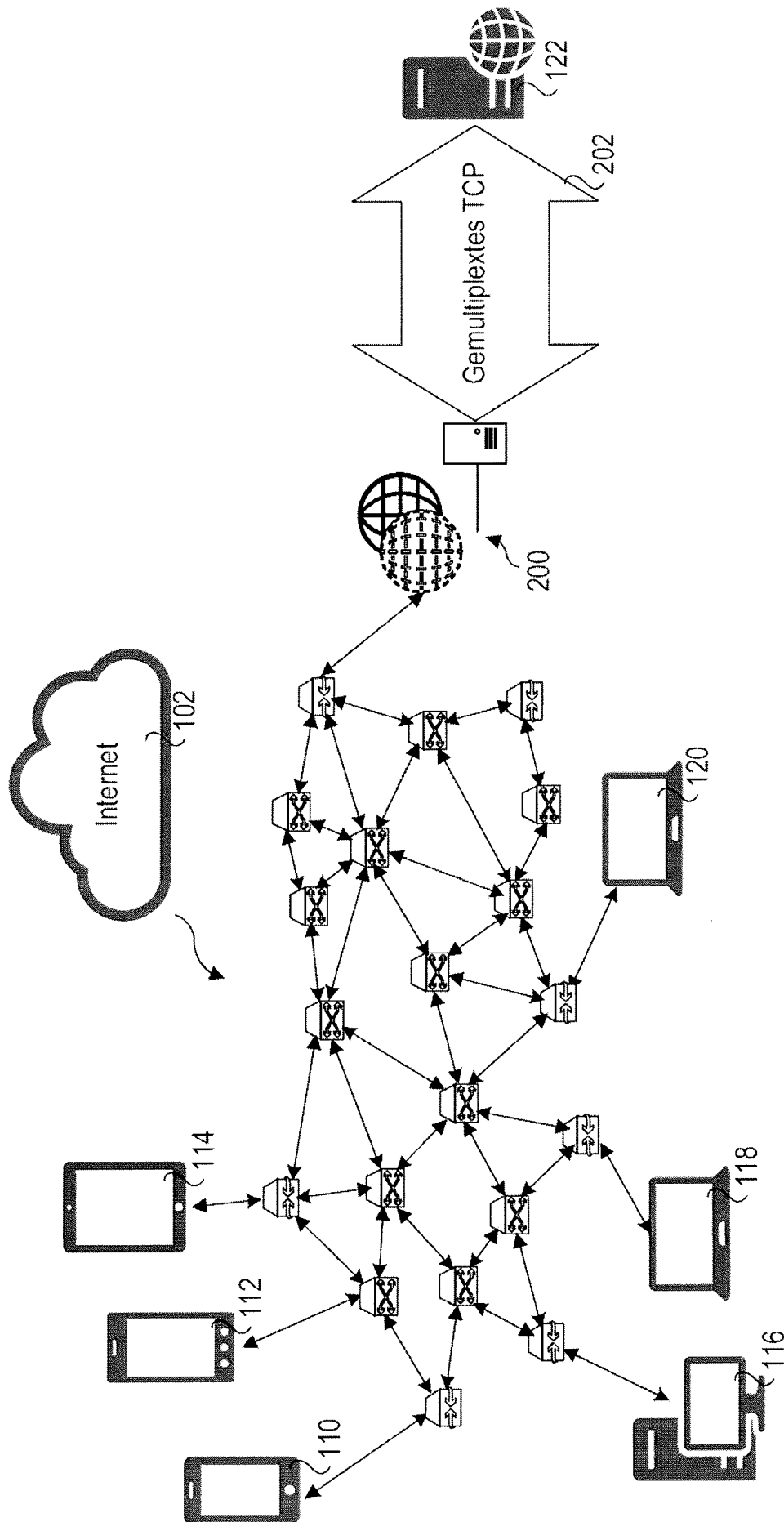


Fig. 2a

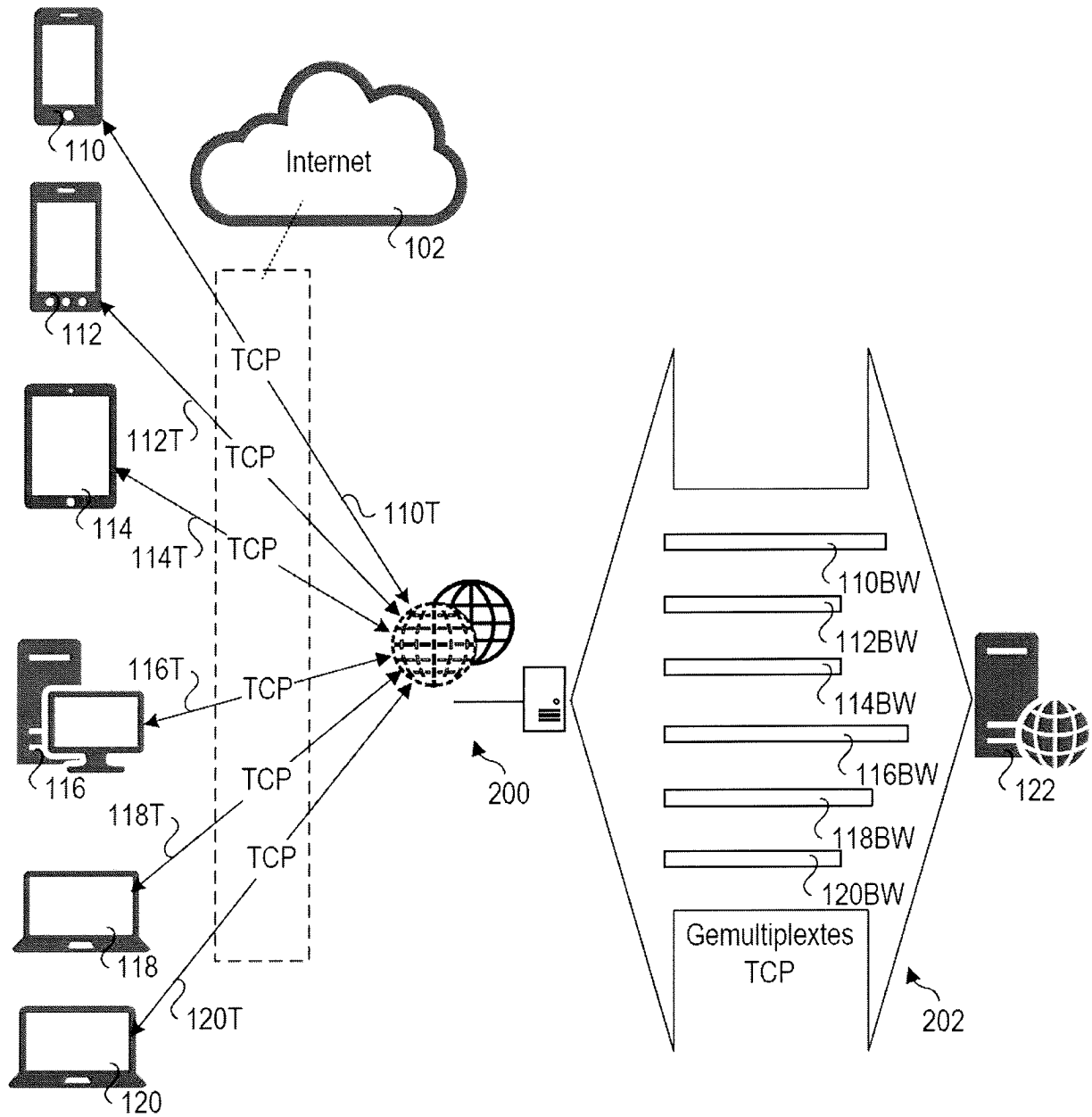


Fig. 2b

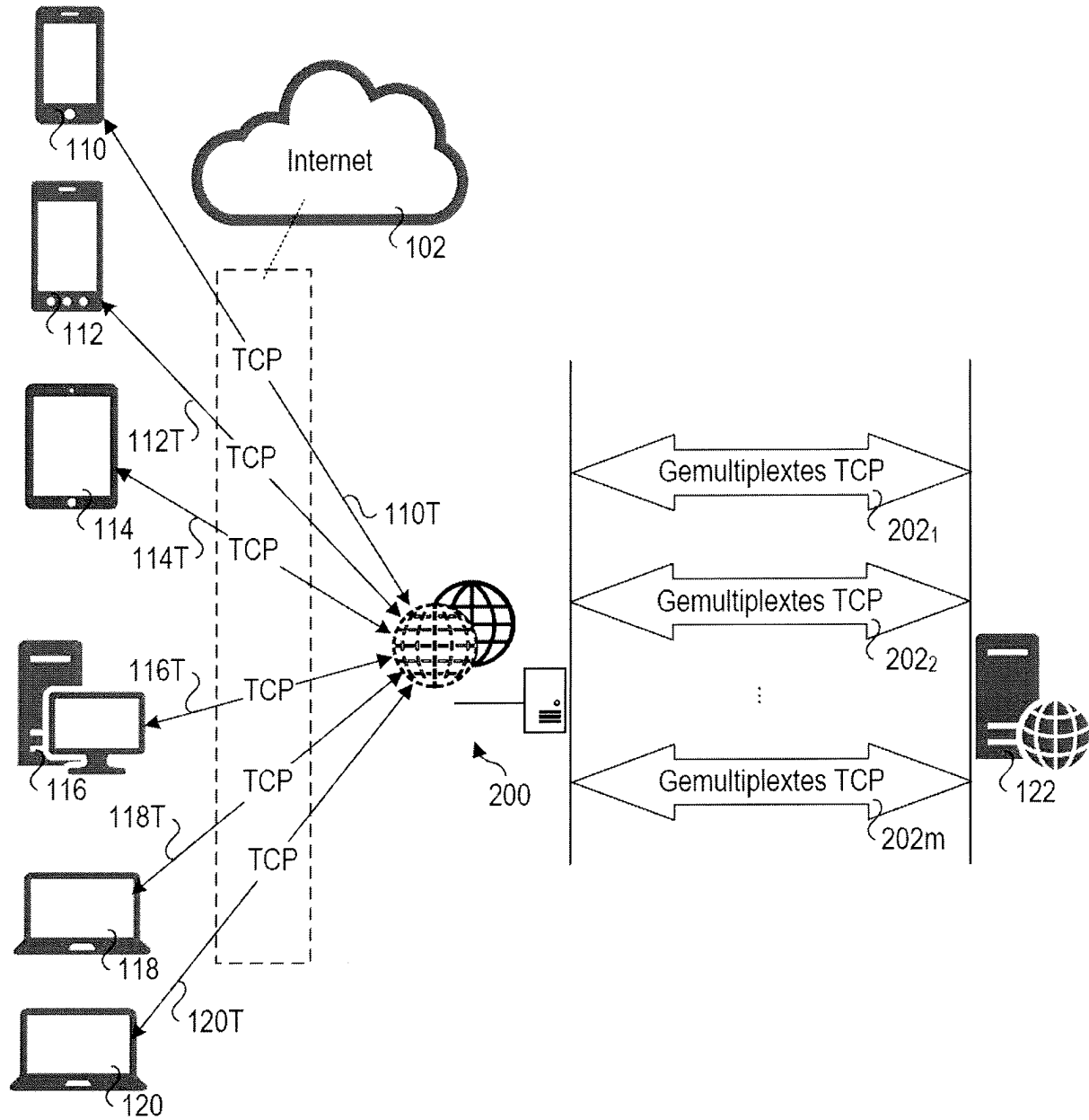


Fig. 2c

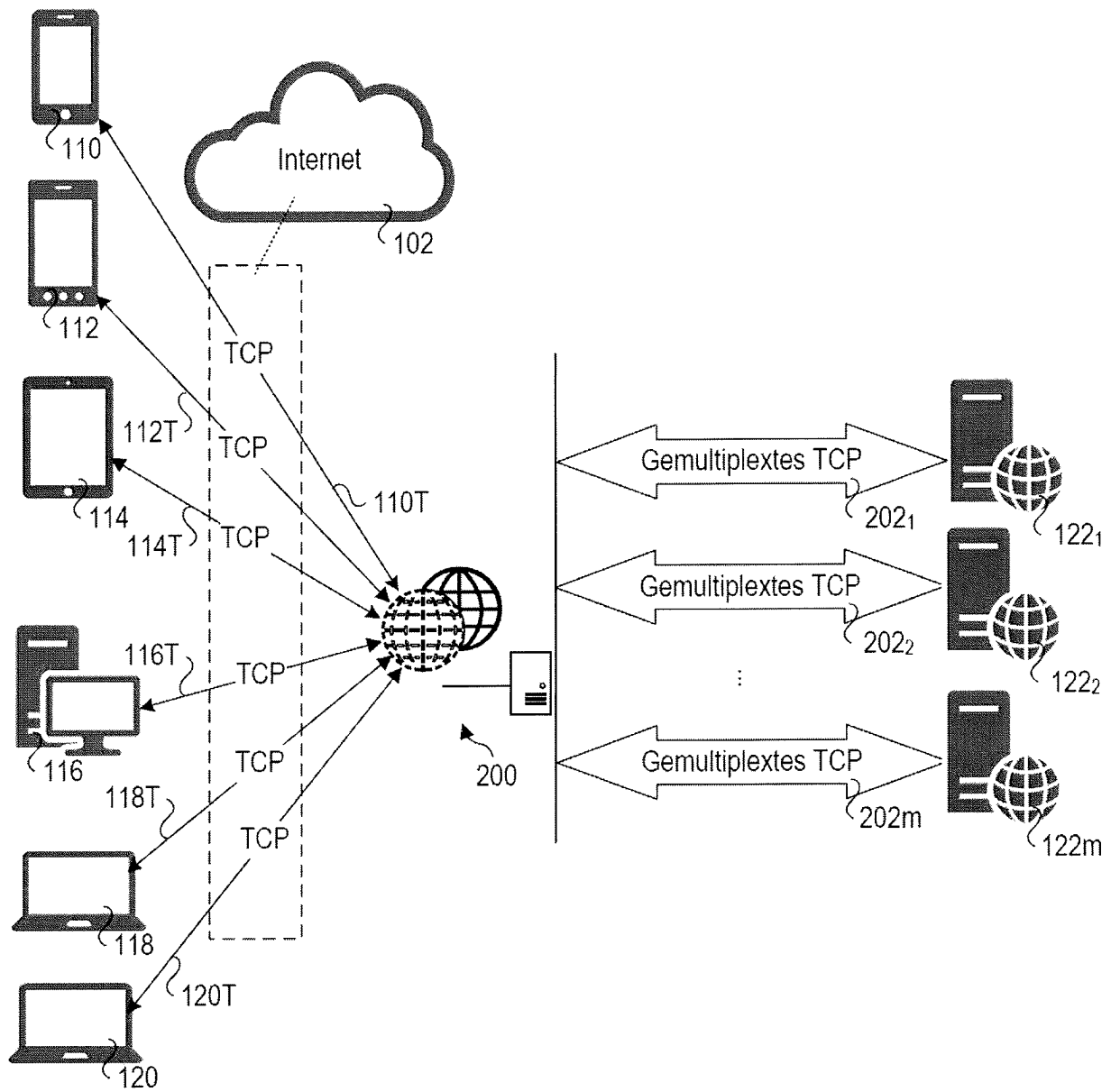


Fig. 2d

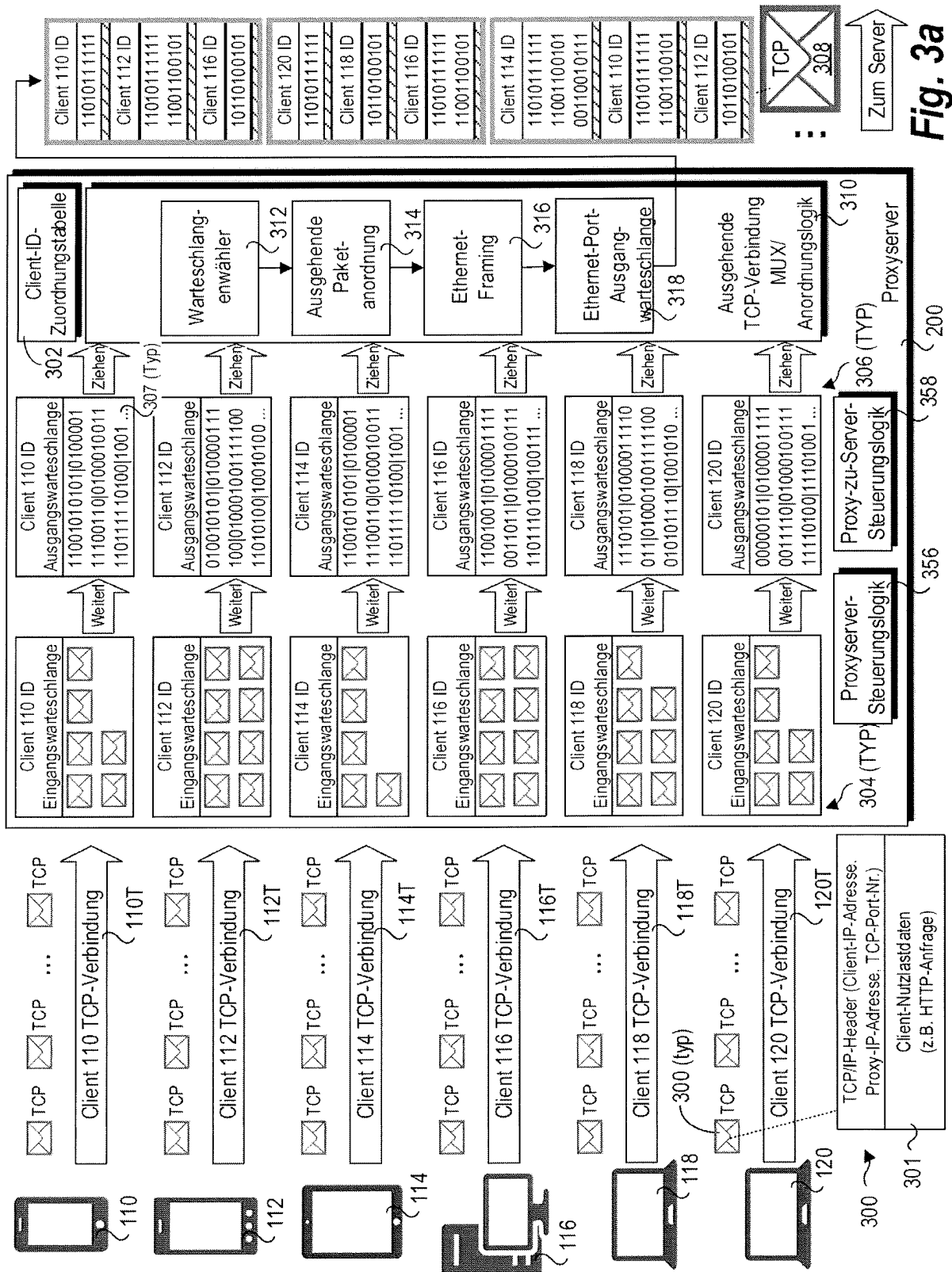
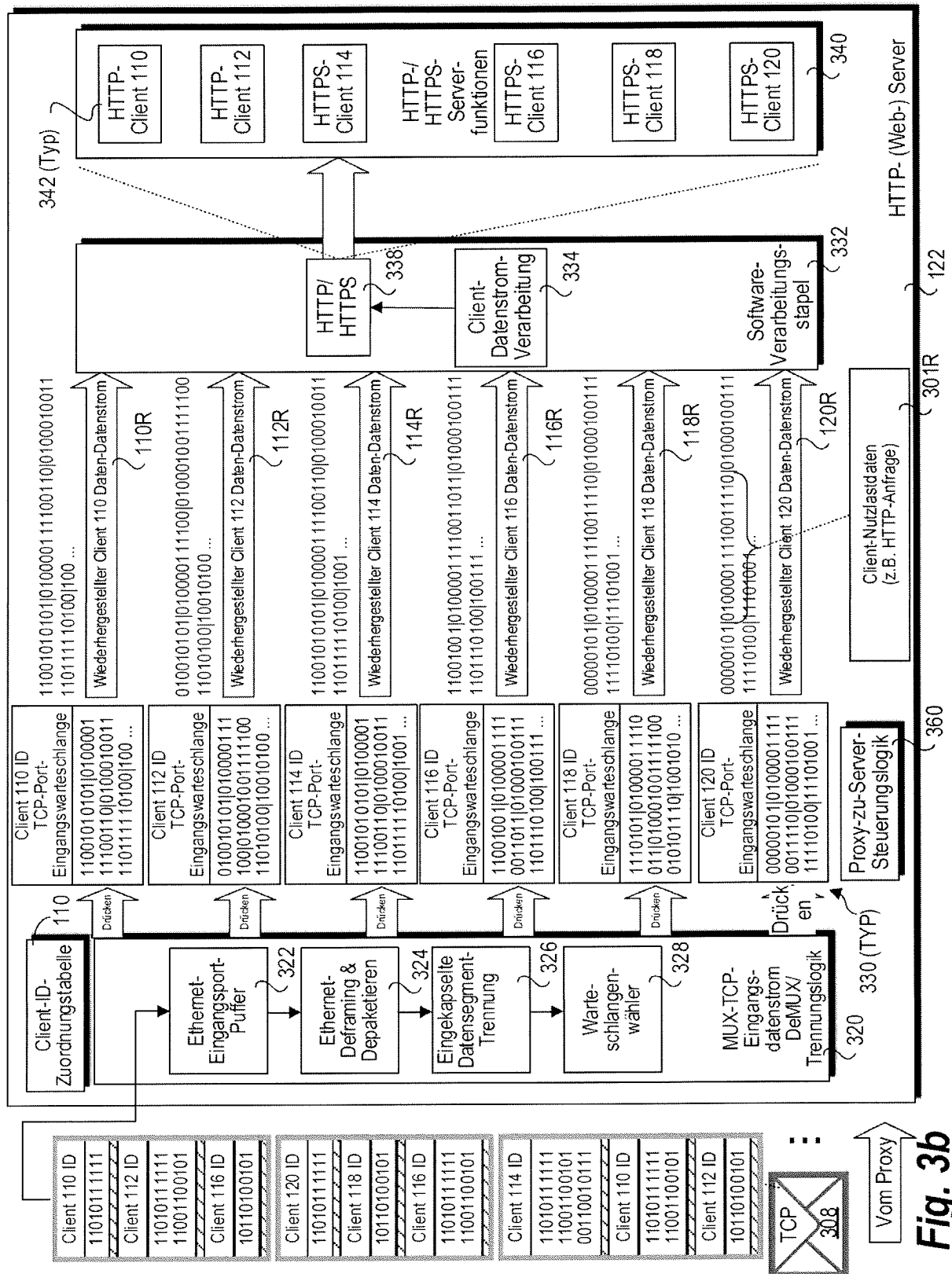
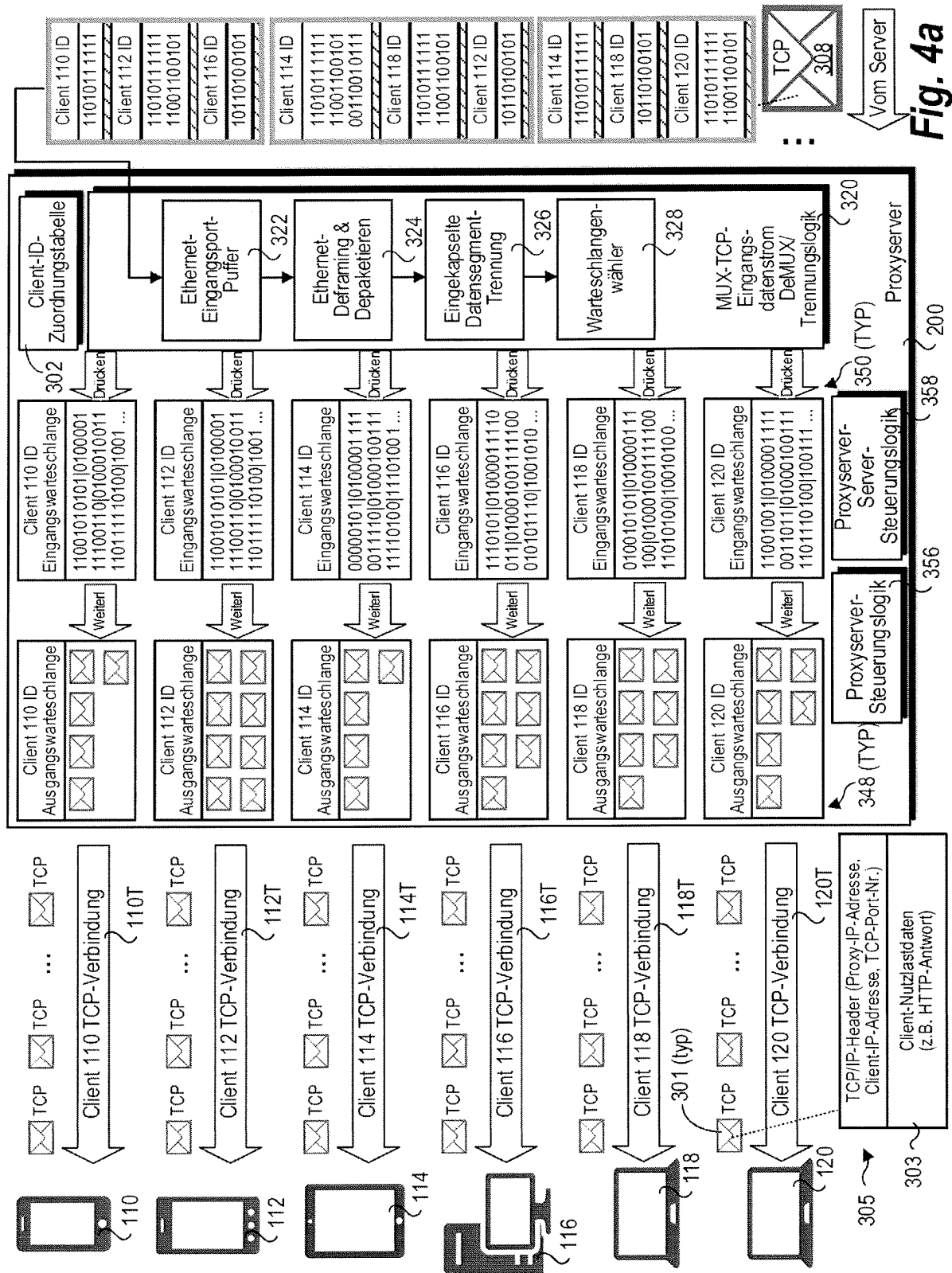
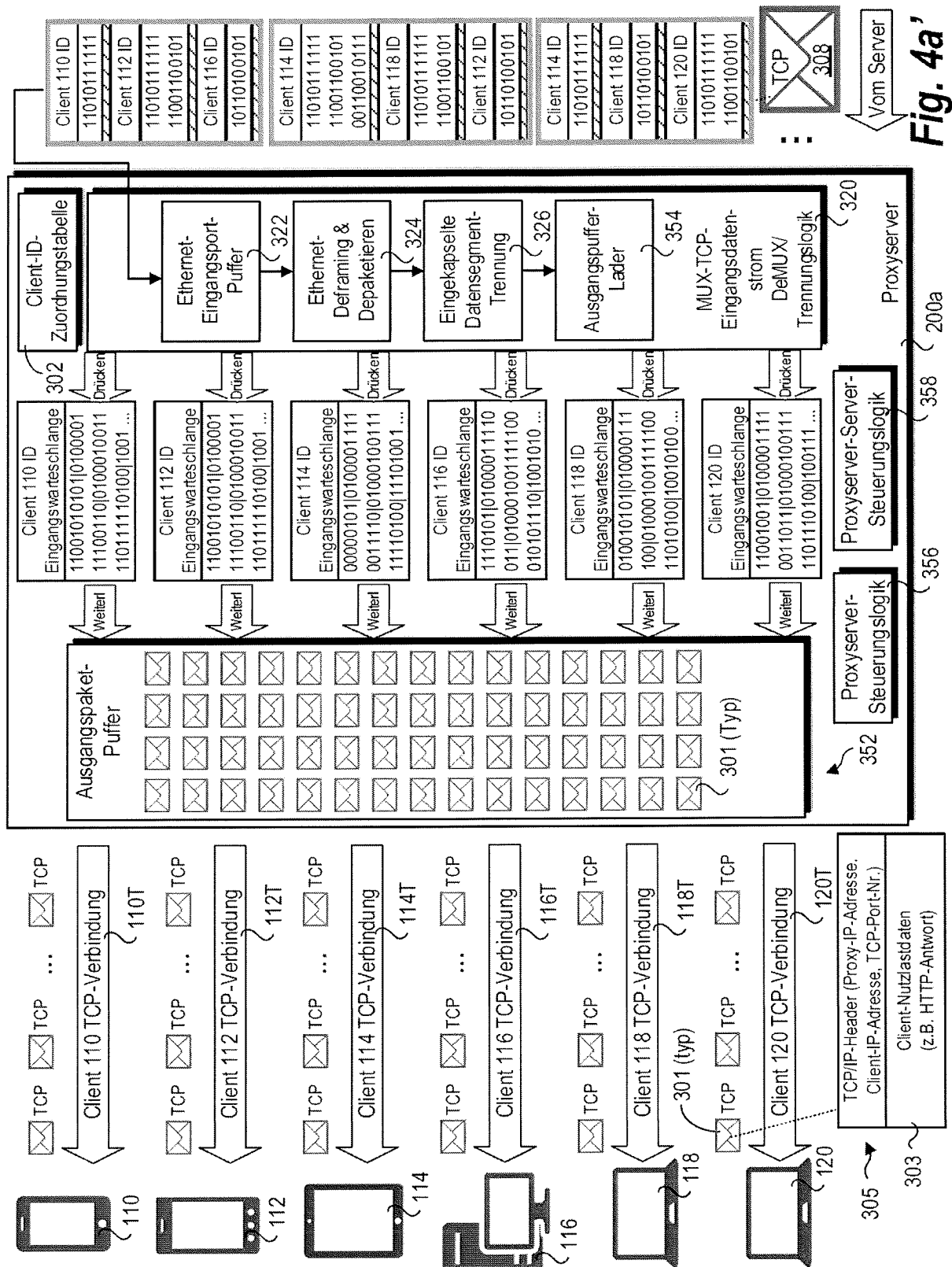


Fig. 3a







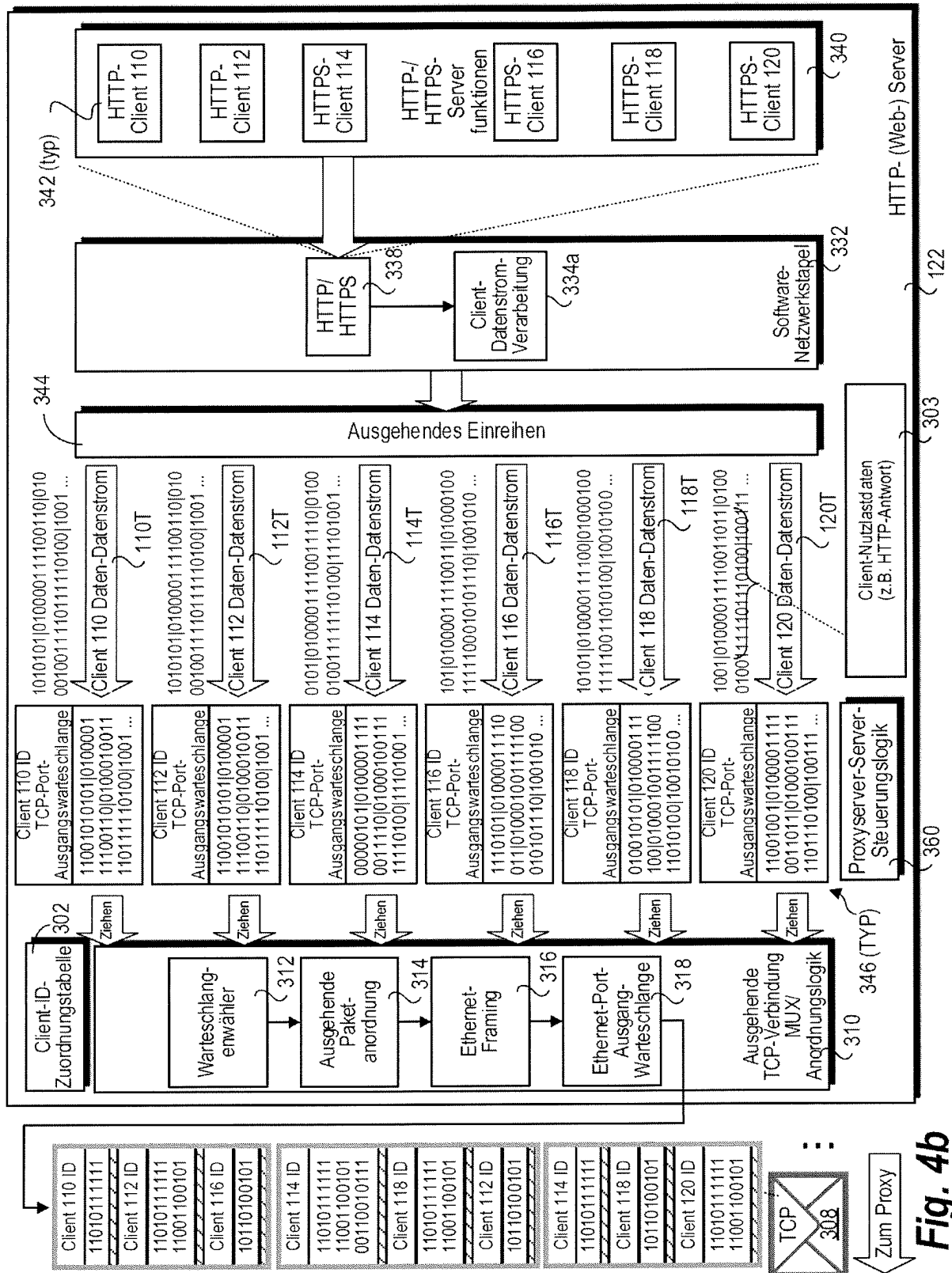
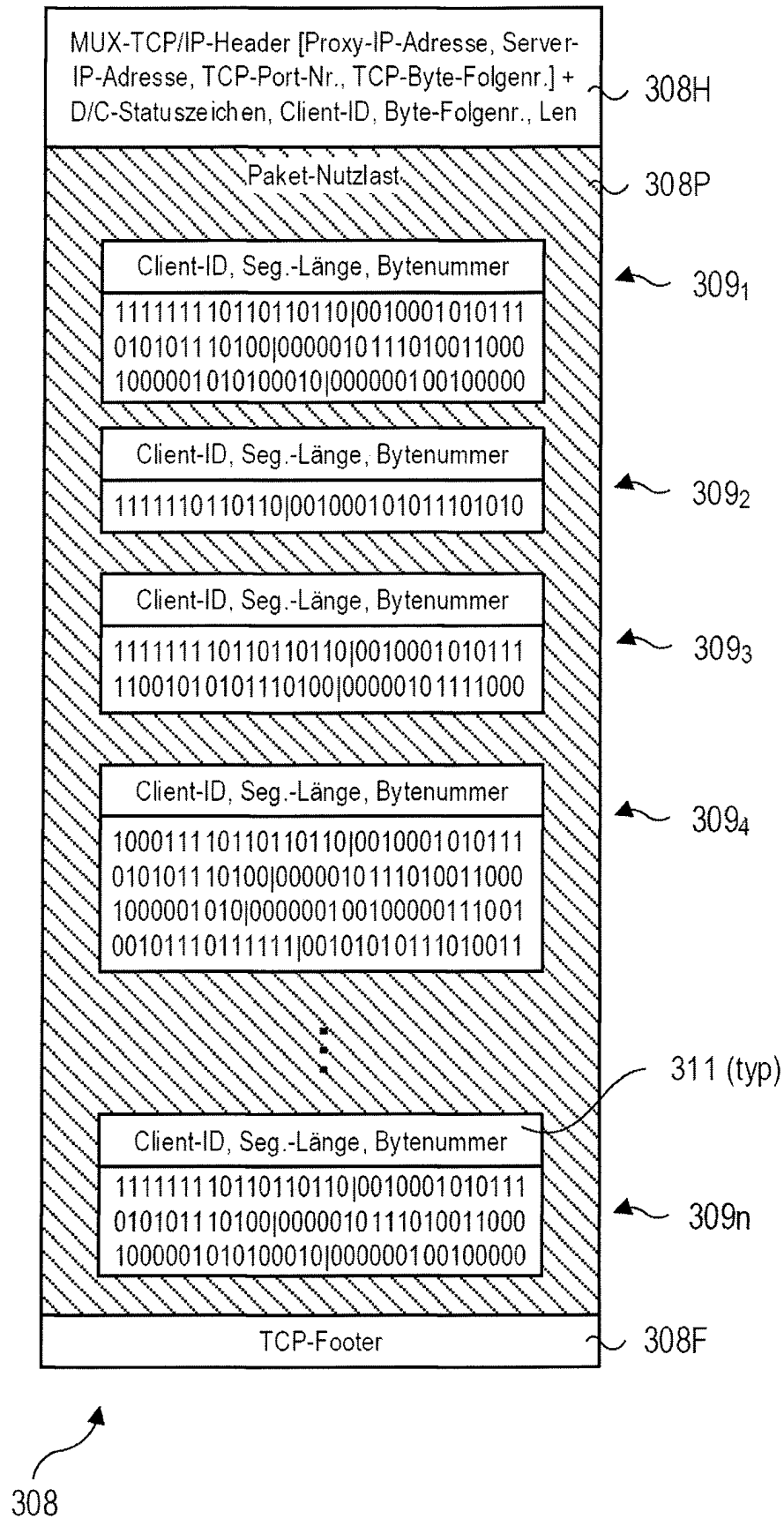
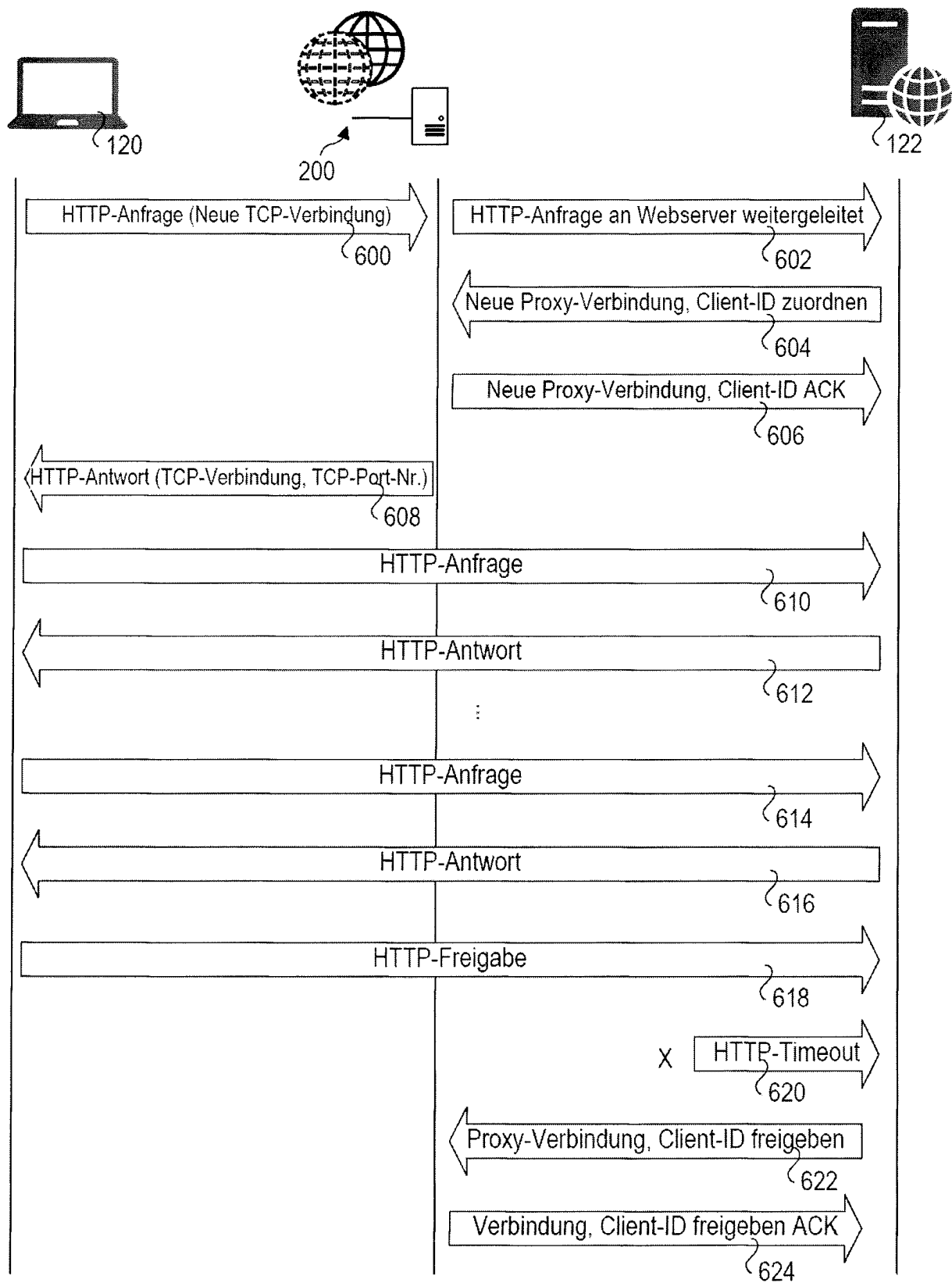


Fig. 4b

**Fig. 5**

**Fig. 6**

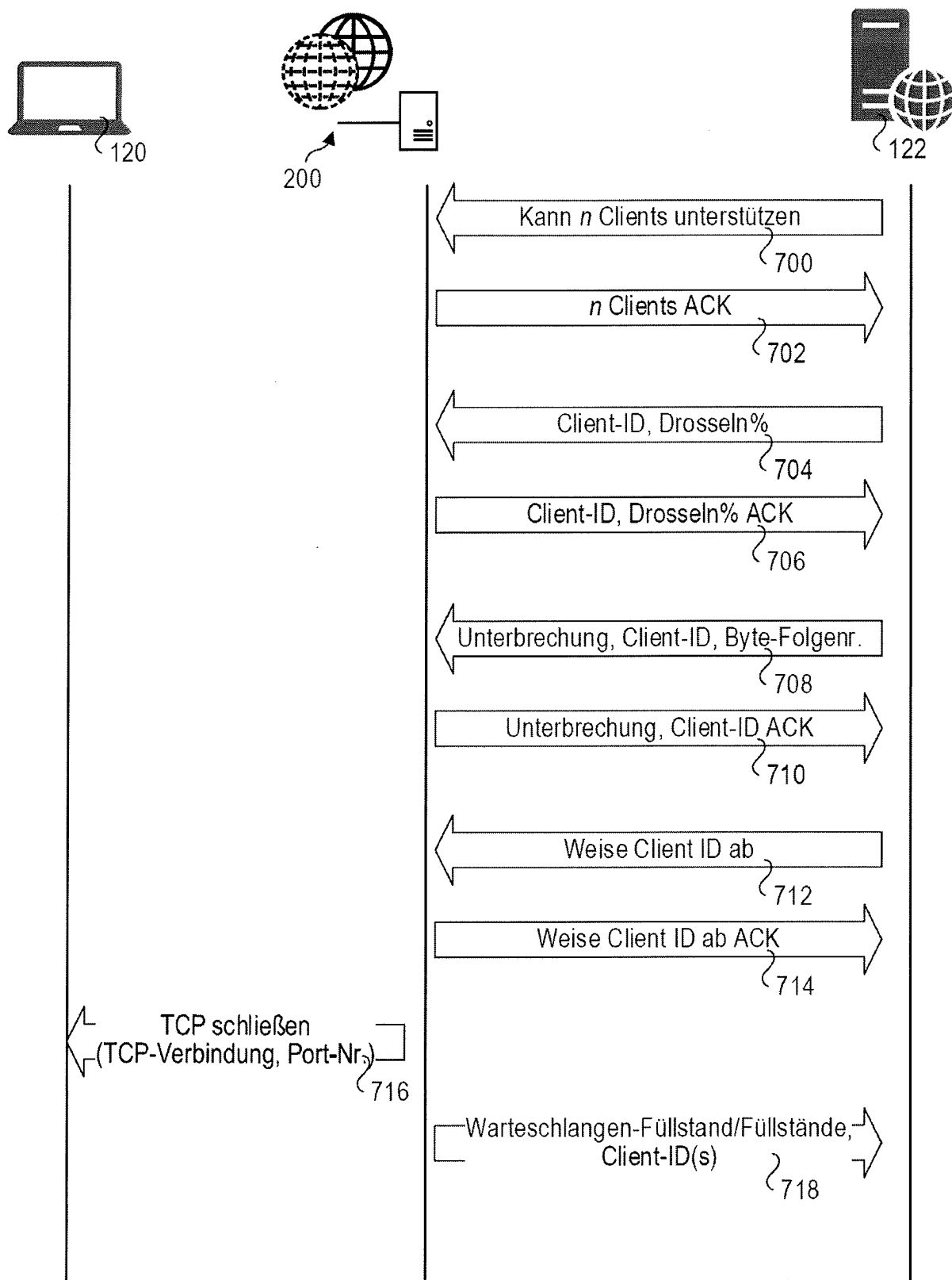


Fig. 7

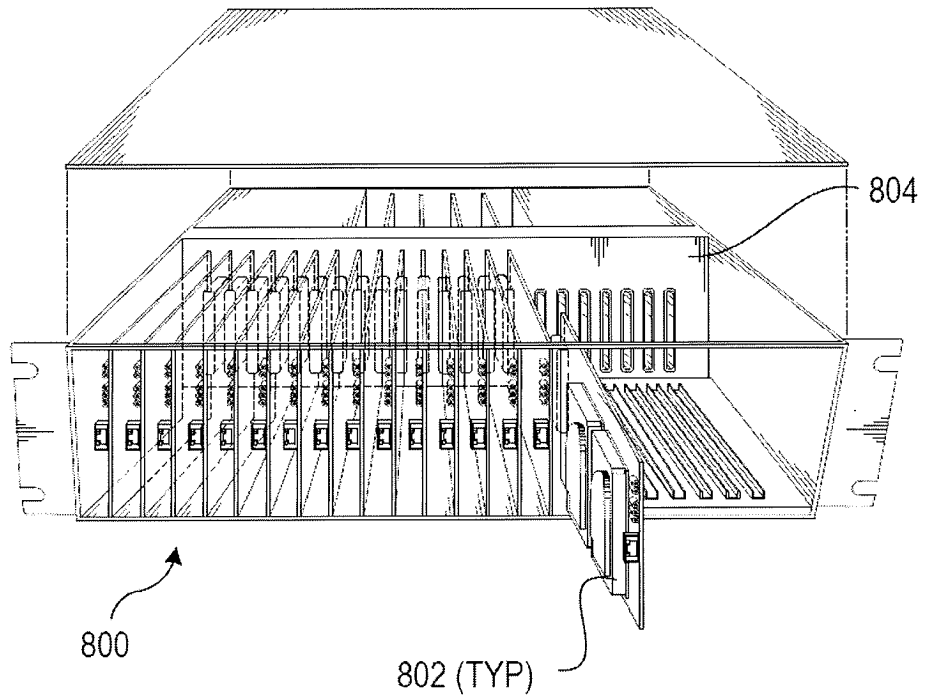


Fig. 8a

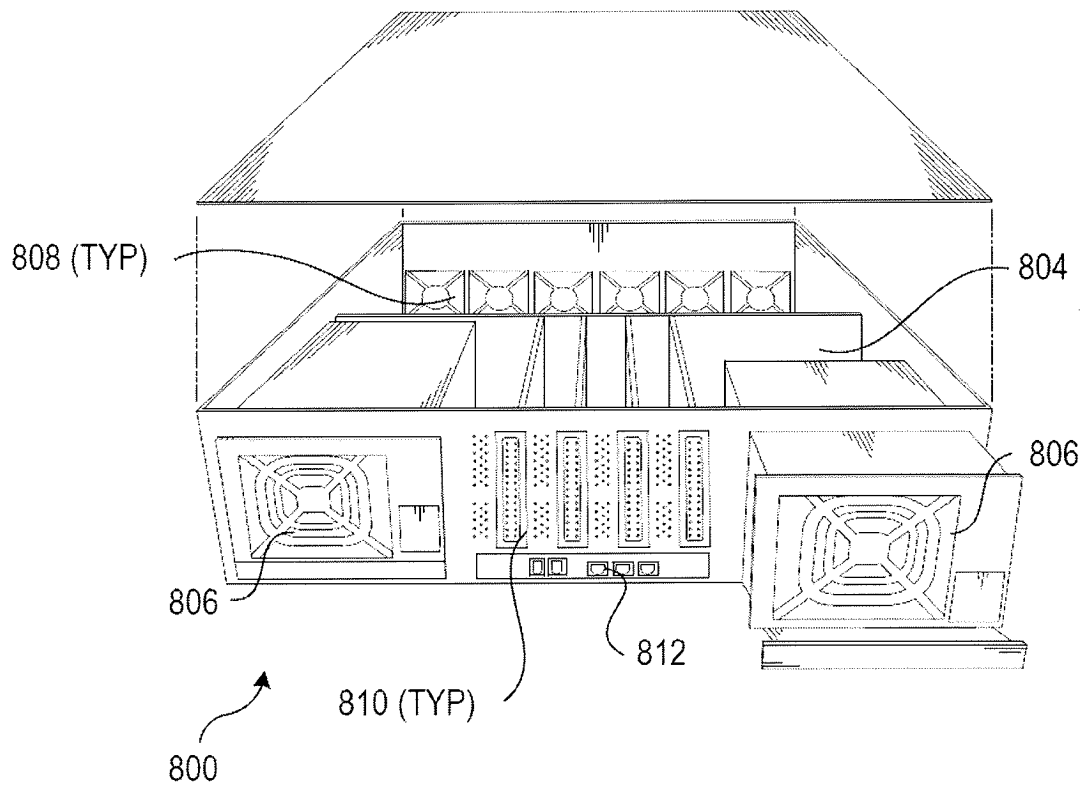


Fig. 8b

Fig. 8c

