



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 602 08 786 T2** 2006.09.28

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 493 257 B1**

(21) Deutsches Aktenzeichen: **602 08 786.4**

(86) PCT-Aktenzeichen: **PCT/EP02/03802**

(96) Europäisches Aktenzeichen: **02 727 530.4**

(87) PCT-Veröffentlichungs-Nr.: **WO 2003/085924**

(86) PCT-Anmeldetag: **05.04.2002**

(87) Veröffentlichungstag
der PCT-Anmeldung: **16.10.2003**

(97) Erstveröffentlichung durch das EPA: **05.01.2005**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **18.01.2006**

(47) Veröffentlichungstag im Patentblatt: **28.09.2006**

(51) Int Cl.⁸: **H04L 29/06** (2006.01)
G06F 17/30 (2006.01)

(73) Patentinhaber:
**Telefonaktiebolaget LM Ericsson (publ),
Stockholm, SE**

(74) Vertreter:
Mohsler, G., Dipl.-Ing., Pat.-Anw., 50733 Köln

(84) Benannte Vertragsstaaten:
**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE, TR**

(72) Erfinder:
**QUINET, Raphael, B-4000 Liège, BE;
SCHAFFRATH, Daniel, 52064 Aachen, DE**

(54) Bezeichnung: **ÜBERTRAGUNGSKONTROLLE FÜR OBJEKTE IN EINEM KOMMUNIKATIONSNETZWERK**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

ALLGEMEINER STAND DER TECHNIK

1. Technisches Gebiet

[0001] Die Erfindung betrifft allgemein das Gebiet der Kommunikationsnetze und spezieller einen Objekttransfer von einer ersten Netzkomponente über eine Zwischenkomponente zu einer zweiten Netzkomponente, wobei die zweite Netzkomponente von der ersten Netzkomponente entfernt ist.

2. Beschreibung des Standes der Technik

[0002] Die Übermittlung von Informationen über moderne Kommunikationsnetze wie das öffentliche Internet oder interne Netze beruht auf speziellen Transferprotokollen. Das World Wide Web, welches einen wichtigen Aspekt des Internets darstellt, verwendet zum Beispiel das Hypertext-Transferprotokoll (HTTP) zum Austauschen von Dateien, welche Text, Bilder, Ton, Video und andere Inhalte enthalten.

[0003] Ein WWW-Server enthält zusätzlich zu den Dateien, die er übermitteln kann, eine HTTP-Komponente, welche dazu vorgesehen ist, auf HTTP-Anforderungen zu warten und sie abzuwickeln, wenn sie eintreffen. Ein WWW-Browser kann als ein HTTP-Client betrachtet werden, welcher so konfiguriert ist, dass er HTTP-Anforderungen an WWW-Server sendet. Jedes Mal, wenn ein Benutzer des Browsers eine Dateianforderung eingibt, indem er entweder eine WWW-Datei "öffnet" (durch Schreiben einer URL (Uniform Resource Locator, URL)) oder auf einen Hypertext-Link klickt, erzeugt der Browser eine entsprechende HTTP-Anforderung für die Datei und sendet sie an die Zieladresse. Die HTTP-Komponente im Zielsystem empfängt die HTTP-Anforderung und sendet die angeforderte Datei zurück.

[0004] Bei der angeforderten Datei kann es sich um eine Hyper Text Markup Language (HTML) Seite handeln, welche HTML-Code enthält. Wenn der Browser die HTML-Seite vom Server empfängt und erkennt, dass der HTML-Code, welcher selbst als ein Objekt betrachtet werden kann, weitere Objekte enthält, wie etwa (Hintergrund-)Bilder, Ton, Scripts oder HTML-Frames, gibt der Browser weitere HTTP-Anforderungen an den Server aus, um die weiteren Objekte abzuholen, welche in dem HTML-Code enthalten sind. Bei Empfang der weiteren HTTP-Anforderungen sendet der Server HTTP-Antworten, welche die angeforderten Objekte wie Bilder enthalten, an den Browser. Wie aus [Fig. 1](#) ersichtlich ist, werden die HTTP-Antworten vom Server an den auf dem Client ausgeführten Browser in derselben Reihenfolge gesendet, in der der Browser die HTTP-Anforderungen ausgegeben hat.

[0005] Die Reihenfolge, in der irgendwelche zusätzlichen Objekte, die im HTML-Code der HTML-Seite enthalten sind, vom Browser angefordert werden, hängt gewöhnlich davon ab, wie die HTML-Seite geschrieben wurde. Zum Beispiel wird ein Objekt, welches am Anfang des HTML-Codes enthalten ist, nicht unbedingt am Anfang der HTML-Seite angezeigt, da Merkmale wie etwa Tabellen, Layer und Frames dem HTML-Autor gestatten, komplexe Layouts zu verwenden. Außerdem hängt die Reihenfolge, in welcher ein Browser HTTP-Anforderungen ausgibt, auch von internen Algorithmen des Browsers ab. Zum Beispiel verwenden manche Browser komplexe Heuristiken zum Erzeugen der HTTP-Anforderungen, indem sie damit beginnen, dass sie die ersten vier Objekte anfordern, so wie sie im HTML-Code der Seite erscheinen. Nachdem die ersten vier Objekte angefordert worden sind, wird jedes zweite Objekt, vom oberen Ende des Bereiches aus gesehen, der gegenwärtig für den Benutzer sichtbar ist, angefordert, danach jedes vierte Objekt und so weiter. Andere Browser verwenden einen einfacheren Algorithmus, welcher die Objekte nacheinander anfordert, so wie sie im HTML-Code erscheinen. Aus dem oben Gesagten geht hervor, dass es gewöhnlich schwierig ist, die Reihenfolge vorherzusagen, in welcher HTTP-Anforderungen für Objekte, auf die in einem HTML-Code verwiesen wird, erzeugt werden.

[0006] Außerdem ist es schwierig, die Reihenfolge vorherzusagen, in welcher angeforderte Objekte vom Browser empfangen werden. Obwohl der aktuelle HTTP-Standard (HTTP/1.1) erfordert, dass auf jeder Transfer Control Protocol (TCP-)Verbindung die HTTP-Antworten vom Server zum Browser in derselben Reihenfolge gesendet werden, in der der Server die HTTP-Anforderungen empfängt, wird die Reihenfolge, in welcher HTTP-Antworten empfangen werden, unvorhersagbar, sobald mehr als eine Verbindung zum Server geöffnet ist. Der Grund hierfür ist die Tatsache, dass aufgrund sich ändernder Netzbedingungen und unterschiedlicher Dauern der Anforderungsverarbeitung manche Verbindungen HTTP-Antworten schneller übertragen können als andere.

[0007] In US 5987466 wird ein Verfahren zum Browsen des World Wide Web des Internet und Präsentieren von Elementen einer Webseite für einen Benutzer beschrieben. Der Webbrowser fordert eine gewünschte HTML an und fordert danach weitere Elemente der Seite auf der Basis von benutzerdefinierten Prioritätsstufen an. Elemente werden dem Benutzer in der Reihenfolge der Priorität präsentiert, sobald sie empfangen wurden.

[0008] Es besteht Bedarf an einem Verfahren und einer Vorrichtung, welche eine verbesserte Übertragung von Objekten von einer ersten Netzkomponente zu einer zweiten Netzkomponente, welche von der

ersten Netzkomponente entfernt ist, ermöglichen.

Komponente empfangen werden.

KURZDARSTELLUNG DER ERFINDUNG

[0009] Dieser Bedarf wird durch die Lehren der unabhängigen Ansprüche befriedigt. Weitere bevorzugte Ausführungsformen sind in den abhängigen Ansprüchen beschrieben.

[0010] Gemäß einem Aspekt der Erfindung wird dieser Bedarf befriedigt durch ein Verfahren zum Steuern eines Objekttransfers in einem Kommunikationsnetz von einer ersten Komponente über eine (Hardware- oder Software-)Zwischenkomponente zu einer zweiten Komponente, welche von der ersten Komponente entfernt ist, wobei der Objekttransfer auf mehreren Objekten beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird, die von der zweiten oder einer anderen Komponente des Kommunikationsnetzes zu verarbeiten sind, und die Zwischenkomponente die Schritte des Sendens einer Objektenanforderung an die erste Komponente und des Empfangens des angeforderten Objektes von der ersten Komponente ausführt, welches dadurch gekennzeichnet ist, dass die Zwischenkomponente ferner die folgenden Schritte ausführt: Bewerten und/oder Aktualisieren einer Priorität des angeforderten Objektes, wobei dem angeforderten Objekt auf der Basis einer Analyse der Objektenanforderung und/oder des Codes, welcher auf das angeforderte Objekt verweist, eine Anfangspriorität zugewiesen worden ist, und, in Abhängigkeit von der Priorität des angeforderten Objektes, Verzögern des angeforderten Objektes oder Weiterleiten des angeforderten Objektes zur zweiten Komponente. Ein Objekt kann selbst Codeabschnitte umfassen, die auf ein oder mehrere weitere Objekte verweisen. Ferner kann ein Code selbst ein (z.B. zuvor angefordertes) Objekt bilden.

[0011] Durch absichtliches Verzögern des Objekttransfers auf der Basis einer zugewiesenen Priorität wird der Gesamttransfer von Objekten vom Standpunkt eines Benutzers aus verbessert (selbst wenn die Gesamtmenge von zu übertragenden Daten nicht verringert wird), da wichtige Objekte bevorzugt übertragen werden. Ferner wird durch Implementierung geeigneter Schemata der Prioritätszuweisung der Objekttransfer von der ersten Netzkomponente zur zweiten Netzkomponente vorhersagbarer. Zum Beispiel kann Objekten, die für den Benutzer von größerer Bedeutung sind, eine höhere Priorität zugewiesen werden, und sie können somit bevorzugt zur zweiten Komponente übertragen werden. Andererseits können Objekte, die von geringerer Bedeutung sind, verzögert werden. Im Extremfall kann ein Objekt mit einer geringen Bedeutung dermaßen verzögert werden, dass es überhaupt nicht zur zweiten Komponente übertragen wird. Dies ermöglicht es, die Reihenfolge zu steuern, in welcher Objekte durch die zweite

[0012] Die Zuweisung von absoluten oder relativen Prioritäten zu Objekten (oder Klassen von Objekten) kann dynamisch durchgeführt werden. Sobald eine Priorität zu Anfang zugewiesen worden ist, kann diese Priorität in Bezug auf spezifische absolute Werte wie etwa Schwellwerte oder in Bezug auf Prioritäten anderer Objekte bewertet werden. Auf der Basis der Bewertung kann entschieden werden, ob ein Objekt zu verzögern ist oder nicht. Vor der Bewertung kann eine Priorität, die einem Objekt zunächst zugewiesen wurde, neu beurteilt werden, mit dem Zweck zu bestimmen, ob die Anfangspriorität aktualisiert werden muss.

[0013] Die Zwischenkomponente kann so betrieben werden, dass sie Objekte, die von der ersten Komponente empfangen werden, umordnet. Das Verzögern von Objekten wird somit vorzugsweise derart durchgeführt, dass sich die Reihenfolge, in welcher die Zwischenkomponente die Objekte von der ersten Komponente empfängt, von der Reihenfolge unterscheidet, in welcher die Objekte zur zweiten Komponente weitergeleitet werden. Das Umordnen kann auf den Prioritäten der zu übertragenden Objekte beruhen. Während des Umordnens könnte die Situation eintreten, dass aufgrund der Verzögerung mancher Objekte die Übertragung anderer Objekte in Wirklichkeit beschleunigt wird.

[0014] Die Objektenanforderung, welche an die erste Komponente gesendet wird, kann von der Zwischenkomponente erzeugt werden. Wenn die Zwischenkomponente das angeforderte Objekt empfängt, "pusht" sie es zur zweiten Komponente, ohne eine explizite Objektenanforderung von der zweiten Komponente empfangen zu haben. Stattdessen kann die Objektenanforderung auch von der zweiten Komponente erzeugt und zur Zwischenkomponente gesendet werden. Nach Empfang der Objektenanforderung von der zweiten Komponente kann sie von der Zwischenkomponente verarbeitet und zur ersten Komponente weitergeleitet werden.

[0015] Wenn die Zwischenkomponente das angeforderte Objekt von der ersten Komponente empfängt, kann das empfangene Objekt entweder verzögert oder direkt zur zweiten Komponente weitergeleitet werden. Es existieren verschiedene Möglichkeiten, wie das angeforderte Objekt, welches durch die Zwischenkomponente empfangen wird, verzögert werden kann. Das Verzögern des angeforderten Objektes kann zum Beispiel mindestens einen der Schritte enthalten: Anweisen der zweiten Komponente, die Objektenanforderung zu wiederholen, Sperren einer Verbindung zu der zweiten Netzkomponente, über welche das angeforderte Objekt weitergeleitet werden soll, und Informieren der zweiten Komponente, dass das angeforderte Objekt zu einem späteren

Zeitpunkt automatisch (z.B. ohne irgendeine weitere Objktanforderung von der zweiten Komponente) zur zweiten Komponente weitergeleitet wird.

[0016] Falls das Verzögern des angeforderten Objektes das Anweisen der zweiten Komponente beinhaltet, die Objktanforderung zu wiederholen, kann die Zwischenkomponente die folgenden Schritte ausführen: Zuweisen eines spezifischen Attributes zu dem zu verzögernden Objekt, Informieren der zweiten Komponente über das Attribut, Empfangen eines Verweises auf das Attribut (z.B. des Attributes selbst oder eines eindeutigen Verweises, der von dem Attribut abgeleitet ist) von der zweiten Komponente und, bei Empfang des Verweises auf das Attribut, Senden des verzögerten Objektes zu der zweiten Komponente oder weiteres Verzögern des verzögerten Objektes. Die Entscheidung, ob ein verzögertes Objekt zur zweiten Komponente zu senden oder weiter zu verzögern ist, kann auf der neu bewerteten relativen Priorität des wiederholt angeforderten Objektes begründet werden.

[0017] Das Attribut kann als ein gemeinsamer Nenner betrachtet werden, welcher die Zwischenkomponente und die zweite Komponente in die Lage versetzt, über die Übertragung des Objektes zu verhandeln, welchem das Attribut zugewiesen wurde. Gewöhnlich wird die Form des Attributes von den wesentlichen Merkmalen des Transferprotokolls abhängen, welches für den Objekttransfer verwendet wird. Zum Beispiel kann im Falle von HTTP das Attribut von einer virtuellen URL gebildet werden, die von der Zwischenkomponente erzeugt wird. Es ist anzumerken, dass das attributbasierte Verzögerungsschema des Anweisens der zweiten Komponente, die Objktanforderung zu wiederholen, allgemein verwendet werden kann, um Objekte zu verzögern, und nicht unbedingt erfordert, dass den zu übertragenden Objekten Prioritäten zugewiesen worden sind.

[0018] In einem attributbasierten Verzögerungsschema kann das Objekt von der Zwischenkomponente zur zweiten Komponente in Reaktion auf eine Objktanforderung, die von der zweiten Komponente empfangen wird, oder entsprechend einem "Push-Schema", d.h. unabhängig von einer solchen Objktanforderung von der zweiten Komponente, gesendet werden. Falls das Verzögerungsschema auf einer von der zweiten Komponente empfangenen Objktanforderung beruht, kann die zweite Komponente über das Attribut im Zusammenhang mit einer Anweisung, die Objktanforderung zu wiederholen, informiert werden. In einem solchen Falle kann der Verweis auf das Attribut durch die Zwischenkomponente im Zusammenhang mit einer wiederholten Objktanforderung von der zweiten Komponente empfangen werden.

[0019] Die Objekte, die zur zweiten Komponente zu

übertragen sind, können zur zweiten Komponente über eine einzige Verbindung oder über mehrere Verbindungen zwischen den Zwischenkomponente und der zweiten Komponente weitergeleitet werden. In einem solchen Szenario mit mehreren Verbindungen können aus den Verbindungen zur zweiten Komponente ausgewählte Verbindungen in Abhängigkeit von den (anfänglichen oder aktualisierten) Prioritäten der angeforderten Objekte, welche über die ausgewählten Verbindungen zur zweiten Komponente weitergeleitet werden sollen, gesperrt werden. Die Zwischenkomponente kann somit eine oder mehrere Verbindungen sperren, so dass Objekte, die eine hohe Priorität besitzen, die zusätzliche Bandbreite nutzen können, die auf der Übertragungsstrecke zwischen der Zwischenkomponente und der zweiten Komponente freigegeben wird. Um verfügbare Bandbreite nicht zu verschwenden, ist die Zwischenkomponente vorzugsweise so konfiguriert, dass sie sicherstellt, dass eine von zwei oder mehr Verbindungen gebildete Übertragungsstrecke vollständig genutzt wird, bevor eine oder mehrere Verbindungen davon gesperrt werden.

[0020] Es existieren verschiedene Verfahren zum Sperren einer Verbindung. Zum Beispiel könnte die Übertragung über die Verbindung für eine bestimmte Zeitdauer blockiert werden, während die Verbindung als solche geöffnet gelassen wird (Zwischenzustand = geöffnet). Stattdessen könnte die Verbindung auch geschlossen werden (Zwischenzustand = geschlossen), während der Zustand der Verbindung gespeichert wird. In einem solchen Falle kann die Verbindung zu einem späteren Zeitpunkt wieder in demselben Zustand geöffnet werden, in dem sie geschlossen wurde. Gemäß einer dritten Möglichkeit kann die Verbindung vollständig geschlossen werden, ohne irgendwelche Informationen über den Zustand der Verbindung zu speichern. In jedem Falle kann die zweite Komponente informiert werden, dass das eine oder die mehreren Objekte, die über die geschlossene Verbindung zu übertragen sind, später gesendet werden, entweder in Reaktion auf eine oder unabhängig von einer (wiederholten) Objktanforderung.

[0021] Anstelle des Sperrens oder zusätzlich zum Sperren einzelner Verbindungen kann der Objekttransfer auch durch eine prioritätsbasierte Anpassung der Transfargeschwindigkeit verzögert werden. Zu diesem Zweck kann jedem Objekt oder jeder Verbindung ein spezifischer Anteil an Verarbeitungskapazitäten dynamisch zugeordnet werden. Im Falle von mehreren Verbindungen erhalten alle Verbindungen oder zumindest einige Verbindungen einen Anteil an der CPU-Zeit, d.h. einen Anteil an der Netzbandbreite. Der Anteil an Verarbeitungskapazitäten, der einer bestimmten Verbindung zugeordnet ist, kann verändert (z.B. ständig verringert) werden, während ein oder mehrere Objekte über die jeweilige Verbindung übertragen werden.

[0022] Weiter oben wurde erwähnt, dass der Objekttransfer auf mehreren Objktanforderungen beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird. Gemäß einer ersten Variante der Erfindung ist ein Code für die zweite Komponente und/oder die Zwischenkomponente jederzeit verfügbar. Gemäß einer zweiten Variante der Erfindung muss der Code noch durch eine der Komponenten, die zweite oder die Zwischenkomponente, geladen werden. Im letzteren Falle kann die Zwischenkomponente eine Codeanforderung, welche von der zweiten Komponente oder von der Zwischenkomponente erzeugt worden ist, an die erste Komponente oder an eine dritte Komponente, welche von der ersten Komponente verschieden ist, senden. Wenn der angeforderte Code von der ersten oder der dritten Komponente empfangen worden ist, kann er durch die Zwischenkomponente im Hinblick auf Verweise auf Objekte, die in dem Code enthalten sind, analysiert werden. Eventuelle Verweise auf Objekte, die in dem Code enthalten sind, können danach zum Zwecke des Zuweisens von (Anfangs-)Prioritäten zu den Objekten, auf die in dem empfangenen Code verwiesen wird, bewertet werden. Der von der ersten oder der dritten Komponente empfangene Code kann schließlich von der Zwischenkomponente zur zweiten Komponente weitergesendet werden.

[0023] Nach Empfang einer Antwort von der ersten Komponente kann das in der Antwort enthaltene angeforderte Objekt im Hinblick auf die Priorität des empfangenen Objektes beurteilt werden. Zum Beispiel kann zu diesem Zweck die Objektgröße und/oder der Objekthalt und/oder ein Header der Antwort analysiert werden. Danach kann bestimmt werden, ob eine Anfangspriorität eines empfangenen Objektes aktualisiert werden muss.

[0024] Vorzugsweise werden zumindest einige Informationen über jedes Objekt oder jede Klasse von Objekten zum Beispiel von der Zwischenkomponente gespeichert. Die gespeicherten Informationen können Prioritätsinformationen für einzelne Objekte oder Klassen von Objekten umfassen, vorzugsweise in Form einer Prioritätsliste. Die Prioritätsliste kann wiederholt bewertet werden. Eine solche Bewertung kann sich auf das Aktualisieren von Prioritätsinformationen und/oder das Löschen von Objekten oder Klassen von Objekten aus der Prioritätsliste beziehen.

[0025] Die Erfindung kann als eine Hardwarelösung oder als eine Softwarelösung implementiert werden. Im Falle einer Softwarelösung kann die Erfindung in Form eines Computerprogrammproduktes realisiert werden, das Programmcodeabschnitte zum Ausführen der einzelnen Schritte der Erfindung umfasst. Das Computerprogrammprodukt kann auf einem computerlesbaren Aufzeichnungsmedium gespeichert sein.

[0026] Gemäß einer bevorzugten Ausführungsform der Erfindung ist die Zwischenkomponente als eine Proxy-Komponente in Form eines Softwareprogramms implementiert, das auf der ersten oder der zweiten Komponente des Kommunikationsnetzes ausgeführt wird. Falls die Erfindung als eine Hardwarelösung implementiert ist, kann die Zwischenkomponente von einem separaten Hardwareelement wie etwa einem Proxy-Server gebildet werden, das zwischen der ersten und der zweiten Komponente im Kommunikationsnetz angeordnet ist. Die Zwischenkomponente kann eine oder mehrere geeignet konfigurierte Kommunikationsschnittstellen zum Kommunizieren wenigstens mit der ersten und der zweiten Komponente des Kommunikationsnetzes sowie einen Block zum Durchführen der Verarbeitung in Verbindung mit dem Verzögern von Objekten aufweisen.

[0027] In dem Kommunikationsnetz existieren eine erste Übertragungsstrecke zwischen der Zwischenkomponente und der ersten Komponente und eine zweite Übertragungsstrecke zwischen der Zwischenkomponente und der zweiten Komponente. Vorzugsweise weisen die erste Übertragungsstrecke und die zweite Übertragungsstrecke unterschiedliche Transfergeschwindigkeiten auf. Zum Beispiel kann eine schnelle Übertragungsstrecke zwischen der Zwischenkomponente und der ersten Komponente vorgesehen sein, und eine vergleichsweise langsamere Übertragungsstrecke kann zwischen der Zwischenkomponente und der zweiten Komponente vorgesehen sein. Diese Situation ist gewöhnlich gegeben, wenn die erste Komponente ein Netzserver ist, die zweite Komponente ein Netz-Client ist und die Zwischenkomponente sich in der Nähe (was Übertragungsstrecken des Netzes betrifft) des Netzservers oder am Netzserver befindet. Die Zwischenkomponente könnte sich jedoch auch in der Nähe (was Übertragungsstrecken des Netzes betrifft) des Netz-Clients oder am Netz-Client befinden. In einem solchen Falle hat die Übertragungsstrecke zwischen der Zwischenkomponente und der zweiten Komponente (dem Netz-Client) eine wesentlich höhere Kapazität und kürzere Latenzzeit als die Übertragungsstrecke zwischen der Zwischenkomponente und der ersten Komponente (dem Netzserver).

[0028] Es ist anzumerken, dass die Erfindung nicht auf den Fall beschränkt ist, dass die erste Komponente als Netzserver agiert und die zweite Komponente als Netz-Client agiert. Insbesondere könnte die Zwischenkomponente auch verwendet werden, um den Objekttransfer zwischen zwei Netz-Clients oder zwei Netzservern zu verbessern.

[0029] Gemäß einer besonders bevorzugten Ausführungsform der Erfindung ist die Zwischenkomponente Teil eines drahtlosen Kommunikationsnetzes wie etwa eines GSM-, GPRS- usw. Zellulernetzes. In einem solchen Netz kann die zweite Komponente

von einem mobilen Endgerät gebildet werden.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0030] Weitere Aspekte und Vorteile der Erfindung werden beim Studium der nachfolgenden ausführlichen Beschreibung bevorzugter Ausführungsformen der Erfindung und unter Bezugnahme auf die Zeichnungen ersichtlich, wobei:

[0031] [Fig. 1](#) eine Prinzipskizze ist, welche einen Transfer von Objekten zwischen einem Netzserver und einem Netz-Client gemäß HTTP veranschaulicht;

[0032] [Fig. 2](#) ein Blockschaltbild eines Netzsystems ist, das eine Zwischenkomponente in Form eines HTTP-Proxy-Servers gemäß der Erfindung umfasst;

[0033] [Fig. 3](#) ein Blockschaltbild des HTTP-Proxy-Servers von [Fig. 2](#) ist;

[0034] [Fig. 4](#) eine Prinzipskizze ist, welche eine auf Umleitung basierte Objektverzögerung in dem in [Fig. 2](#) dargestellten Netzsystem darstellt;

[0035] [Fig. 5](#) ein Flussdiagramm ist, welches die einer Objektverzögerung vorausgehenden Schritte zeigt;

[0036] [Fig. 6](#) ein Flussdiagramm ist, welches die Entscheidungen im Zusammenhang mit einem Objekttransfer gemäß der Erfindung darstellt; und

[0037] [Fig. 7](#) ein Blockschaltbild ist, welches eine Proxy-Komponente gemäß der Erfindung darstellt, die sich an einem Netz-Client befindet.

BESCHREIBUNG EINER BEVORZUGTEN AUSFÜHRUNGSFORM

[0038] Obwohl die vorliegende Erfindung in einem beliebigen Kommunikationsnetz realisiert werden kann, in welchem ein anforderungsbasierter Objekttransfer über eine Zwischenkomponente durchgeführt wird, wird die folgende Beschreibung bevorzugter Ausführungsformen beispielhaft in Bezug auf den Transfer von HTML-Code gemäß dem HTTP-Protokoll über das WWW dargelegt. Im Prinzip könnten ebenso gut von HTTP verschiedene Transferprotokolle wie das WAP Transportprotokoll oder gewisse Mechanismen des entfernten Prozeduraufrufs (Remote Procedure Call, RPC) und von HTML verschiedene Codes, zum Beispiel die WAP Markup-Language (WML) oder irgendwelche Ableitungen der eXtensible Markup-Language (XML), benutzt werden. Ferner könnte, obwohl die folgende Beschreibung hauptsächlich einen Objekttransfer von einem Server zu einem Client betrifft, der Objekttransfer auch zwischen beliebigen zwei oder mehr Netzkomponenten

durchgeführt werden.

[0039] In [Fig. 2](#) ist ein Blockschaltbild eines Netzsystems **10** gemäß der Erfindung dargestellt. Wie aus [Fig. 2](#) ersichtlich wird, umfasst das Netzsystem **10** eine erste Komponente in Form eines Servers **20**, eine Zwischenkomponente in Form eines HTTP-Proxy-Servers **30** und eine zweite Komponente in Form eines Clients **40**. Der Proxy-Server **30** ist im Netzsystem **10** derart angeordnet, dass er eine schnelle Übertragungsstrecke **12** zum Server **20** hin und eine vergleichsweise langsamere Übertragungsstrecke **14** zum Client **40** hin aufweist. Jede Übertragungsstrecke **12**, **14** wird von mehreren TCP-Verbindungen **50** gebildet. Jede TCP-Verbindung **50** ist so konfiguriert, dass sie den Transfer von HTTP-Anforderungen und HTTP-Antworten zwischen dem Server **20** und dem Client **40** ermöglicht.

[0040] Der Proxy-Server **30** führt einige herkömmliche Proxy-Funktionen aus, wie Caching und Filtern von Objekten. Außerdem ist der Proxy-Server **30** so konfiguriert, dass er ein Objekt, welches vom Server **20** empfangen wird und welches zum Client **40** weitergeleitet werden soll, künstlich verzögert. Dies geschieht durch die Verwendung einer Kombination von zeitweiliger Sperrung des Datentransfers auf einigen Verbindungen **50** und von HTTP-Umleitungsnachrichten, welche einen Browser, der auf dem Client **40** ausgeführt wird, zwingen, eine Objktanforderung nach einer gewissen Zeit zu wiederholen. Durch Verwendung dieser Mechanismen ordnet der Proxy-Server **30** die vom Server **20** empfangenen HTTP-Antworten auf eine solche Weise um, dass Objekte, die eine höhere Priorität besitzen, dem Client **40** zuerst übermittelt werden. Zu diesem Zweck weist der Proxy-Server **30** den Objekten, die zum Client **40** weitergeleitet werden sollen, dynamisch Prioritäten zu. Um sicherzustellen, dass das Verzögern der weniger wichtigen Objekte nicht dazu führt, dass die Übertragungsstrecke **14** zwischen dem Proxy-Server **30** und dem Client **40** frei wird, überwacht der Proxy-Server **30** fortlaufend oder zumindest wiederholt den Verkehr auf dieser Übertragungsstrecke **14**.

[0041] Der Aufbau des Proxy-Servers **30** ist in [Fig. 3](#) detaillierter dargestellt. Wie aus [Fig. 3](#) ersichtlich ist, umfasst der Proxy-Server **30** eine Kommunikationsschnittstelle **32**, die zwischen der ersten Übertragungsstrecke **12** zum Server **20** und der zweiten Übertragungsstrecke **14** zum Client **40** geschaltet ist. Die Kommunikationsschnittstelle **32** ist so konfiguriert, dass sie das Senden von Objktanforderungen zum und den Empfang der angeforderten Objekte vom Server **20** ermöglicht. Eine Verarbeitungseinheit **34** des Proxy-Servers **30** kommuniziert mit der Kommunikationsschnittstelle **32**. Die Verarbeitungseinheit **34** ermöglicht es, die Priorität eines beliebigen Objektes, das über die erste Übertragungsstrecke **12** vom Server **20** empfangen wird, zu bewerten und/oder an-

zupassen. Außerdem ermöglicht es die Verarbeitungseinheit **34**, einem angeforderten Objekt auf der Basis einer Analyse der Objektanforderung und/oder des Codes, der auf das angeforderte Objekt verweist, eine Anfangspriorität zuzuweisen. Mögliche Zuweisungsschemata werden später ausführlicher erörtert.

[0042] In Abhängigkeit von der anfänglichen oder aktualisierten Priorität eines angeforderten Objektes steuert die Verarbeitungseinheit **34** die Kommunikationsschnittstelle **32** so, dass ein angefordertes Objekt, das vom Server **20** empfangen wurde, entweder verzögert oder über die zweite Übertragungsstrecke **14** zum Client **40** weitergeleitet wird. Falls ein angefordertes Objekt, das vom Server **20** empfangen wurde, verzögert werden soll, wird es zeitweilig in einem Puffer **36** gespeichert, auf den sowohl durch die Kommunikationsschnittstelle **32** als auch durch die Verarbeitungseinheit **34** zugegriffen werden kann. Stattdessen oder zusätzlich kann der Puffer **36** auch als eine Software- oder Hardwarekomponente der Verarbeitungseinheit **34** implementiert sein.

[0043] Neben den oben umrissenen Aufgaben ist die Verarbeitungseinheit **34** außerdem so konfiguriert, dass sie es gestattet, einem Objekt, welches verzögert werden soll, ein spezifische Attribut zuzuweisen (ein beispielhaftes Format dieses Attributs wird später ausführlicher erörtert). Die Verarbeitungseinheit **34** ermöglicht es, die Kommunikationsschnittstelle **32** so anzusteuern, dass die Kommunikationsschnittstelle **32** den Client **40** über dieses Attribut informiert. Wenn die Kommunikationsschnittstelle **32** vom Client **40** einen Verweis auf das Attribut empfängt, wertet die Verarbeitungseinheit **34** diesen Verweis aus und steuert die Kommunikationsschnittstelle **32** so an, dass das verzögerte Objekt, welchem dieses Attribut zuvor zugewiesen wurde und welches gegenwärtig im Puffer **36** gespeichert ist, entweder zum Client **40** gesendet oder weiter verzögert wird. Eine weitere Verzögerung des im Puffer **36** gespeicherten Objektes kann notwendig werden, wenn Objekte mit einer höheren Priorität zuerst zum Client **40** weitergeleitet werden müssen.

[0044] Im Folgenden wird eine bevorzugte Funktionsweise des in [Fig. 2](#) dargestellten Netzsystems **10** beispielhaft beschrieben.

[0045] Wenn ein Benutzer eine URL eingibt, einen Link anklickt oder einem Favorit folgt, gibt der Browser, der auf dem Client **40** ausgeführt wird, eine HTTP-Anforderung für die entsprechende HTML-Seite aus, welche HTML-Code enthält. Die vom Browser ausgegebene HTTP-Anforderung wird durch den Proxy-Server **30** über die Übertragungsstrecke **14** empfangen und über die Übertragungsstrecke **12** zum Zielservers **20** weitergeleitet. Der Server **20** antwortet, indem er den HTML-Code für die angeforderten Seite zum Proxy-Server **30** sendet, welcher den

vom Server **20** empfangenen HTML-Code analysiert, um jeder Art von Objekten wie Links, Frames, Scripts, Bilder usw., auf die im HTML-Code verwiesen wird, eine Anfangspriorität zuzuweisen (erste Analysephase).

[0046] Unabhängig von dieser Analyse des HTML-Codes leitet der Proxy-Server **30** den HTML-Code über die Übertragungsstrecke **14** zum Client **40** weiter. Wenn der Browser, der auf dem Client **40** ausgeführt wird, die den HTML-Code enthaltende HTML-Seite empfängt, verarbeitet er den HTML-Code. Falls der Browser erkennt, dass der HTML-Code weitere Objekte enthält, gibt er weitere HTTP-Anforderungen für die Objekte aus, auf die im HTML-Code verwiesen wird. Diese HTTP-Anforderungen für weitere Objekte werden durch den Proxy-Server **30** empfangen und ausgewertet. Den meisten der angeforderten Objekte wird bereits während der vom Proxy-Server **30** durchgeführten vorherigen Analyse des HTML-Codes, den er zum Server **40** weitergeleitet hat, eine Anfangspriorität zugewiesen worden sein. In dieser zweiten Analysephase weist der Proxy-Server **30** jedoch entweder jenen Objekten, denen noch keine Priorität zugewiesen worden ist, Anfangsprioritäten zu, oder er aktualisiert die Priorität von solchen Objekten, denen bereits eine Anfangspriorität zugewiesen wurde. Die Priorität wird auf der Basis von zusätzlichen Informationen aktualisiert, die in der vom Client **40** empfangenen HTTP-Anforderung verfügbar sind.

[0047] Der Proxy-Server **30** leitet alle vom Client **40** empfangenen HTTP-Anforderungen für zusätzliche Objekte zum Server **20** weiter. Der Server **20** antwortet, indem er die angeforderten Objekte zum Proxy-Server **30** sendet. Der Proxy-Server **30** analysiert die HTTP-Antworten (welche die angeforderten Objekte enthalten) vom Server **20** in einer dritten Analysephase und aktualisiert, falls erforderlich, die Prioritäten der Objekte, die in der HTTP-Antwort enthalten sind. Außerdem bewertet der Proxy-Server **30** die relative Priorität eines beliebigen empfangenen Objektes in Bezug auf die Priorität von zuvor empfangenen Objekten, welche noch im Puffer **36** gespeichert sind (siehe [Fig. 3](#)), und die Prioritäten von Objekten, welche bald erwartet werden. Informationen hinsichtlich der Objekte, welche bald vom Server **20** empfangen werden, können aus dem HTML-Code abgeleitet werden, welcher zuvor zum Client **40** weitergeleitet wurde, oder aus den HTTP-Anforderungen vom Client **40**, für welche die entsprechenden Objekte noch nicht vom Server **20** empfangen worden sind. Stattdessen oder zusätzlich kann der Proxy-Server **30** auch so konfiguriert sein, dass er eine Priorität eines Objektes, welches gerade vom Server **40** empfangen worden ist, mit einem absoluten Wert wie etwa einer Prioritätsschwelle vergleicht.

[0048] Auf der Basis der Beurteilung der absoluten

und/oder relativen Priorität eines Objekts und des aktuellen und/oder vorhergesagten Verkehrs auf der Übertragungsstrecke **14** entscheidet der Proxy-Server **30**, ob dieses Objekt zu verzögern ist, z.B. ob dieses Objekt zeitweilig im Puffer **36** zu speichern ist oder ob die entsprechende HTTP-Anforderung noch nicht zum Server **20** weiterzuleiten ist, oder ob das Objekt umgehend zum Client **40** weiterzuleiten ist.

[0049] Die absichtliche Verzögerung einzelner Objekte verbessert vom Standpunkt eines Benutzers aus den Objekttransfer insgesamt. Es ist zum Beispiel wohlbekannt, dass die relative Wichtigkeit der verschiedenen Objekte, die in einer Webseite enthalten sind, sich stark unterscheidet: Ein Bild, welches verwendet wird, um ein grafisches Menü aufzubauen, kann für die Navigation einer Website wesentlich sein, während ein Hintergrundbild lediglich dazu dient, der Seite ein schöneres Aussehen zu verleihen. Die Erfindung kann somit angewendet werden, um dem Benutzer die wichtigeren Informationen zuerst zu liefern. Sobald genügend viele Informationen einer Webseite auf dem Bildschirm des Benutzers angezeigt sind, kann sich der Benutzer dafür entscheiden, einen Link anzuklicken und eine andere Webseite anzufordern, ohne auf die restlichen (weniger wichtigen) Teile der vorhergehenden Webseite, welche noch immer übertragen wird, warten zu müssen. Infolgedessen ist der Benutzer nicht mehr gezwungen zu warten, bis einige uninteressante Objekte übertragen worden sind, bevor er in der Lage ist, die wichtigen zu sehen. Dies ist insbesondere nützlich in Verbindung mit dem mobilen Internet, welches gewöhnlich langsamer und teurer ist als das feste Internet.

ZUWEISUNG UND ANPASSUNG VON PRIORITÄTEN

[0050] Wie aus dem Obigen ersichtlich geworden ist, kann der Proxy-Server **30** die Priorität eines Objektes, das an den Client **20** gesendet werden soll, während drei verschiedener Phasen zuweisen oder anpassen, nämlich wenn ein Verweis auf dieses Objekt in einem HTML-Code (d.h. in einem vorhergehenden Objekt) gefunden wird, welcher zum Client **40** weiterzuleiten ist, wenn eine HTTP-Anforderung betreffs dieses Objektes von dem auf dem Client **40** ausgeführten Browser ausgegeben wird, oder wenn die entsprechende HTTP-Antwort, die das angeforderte Objekt enthält, vom Server **20** empfangen wird.

[0051] Im Folgenden werden verschiedene Schemata des Zuweisens oder Aktualisierens von Prioritäten während jeder dieser drei Phasen beispielhaft beschrieben. Während jeder der drei Phasen wird eine Prioritätsliste, welche Prioritätsinformationen für einzelne Objekte oder Klassen von Objekten enthält, entweder erzeugt oder aktualisiert. In der Prioritätsliste sind die einzelnen Objekte oder Klassen von Ob-

jekten nach zunehmender oder abnehmender Priorität geordnet. Die Reihenfolge der Objekte in der Prioritätsliste kann somit als Prioritätsinformation betrachtet werden. Stattdessen oder zusätzlich können jedoch auch zusätzliche Prioritätsinformationen wie absolute oder relative Prioritätswerte (Zahlen) Teil der Prioritätsliste sein.

[0052] Die exakte Priorität, die spezifischen Objekten oder Klassen von Objekten zugewiesen wird, ist implementierungsabhängig und in der beispielhaften Ausführungsform durch den Betreiber des Proxy-Servers **30** oder einen Benutzer des auf dem Client **40** ausgeführten Browsers konfigurierbar. Um zum Beispiel dem Betreiber des Proxy-Servers **30** zu ermöglichen, mehr Kontrolle über die Priorität einiger Objekte zu haben, könnten eine oder mehrere Listen von URL-Adressen (die Mustervergleich (Pattern Matching) verwenden) vorhanden sein, welche dem Betreiber ermöglichen, die Priorität der in diesen Listen erscheinenden Objekte zu erhöhen oder zu verringern. Zum Beispiel könnte ein Betreiber beschließen, die Priorität aller Bilder zu erhöhen oder zu verringern, die von einer Werbefirma heruntergeladen werden. Dieselben Möglichkeiten können dem Benutzer zur Verfügung gestellt werden. Zum Beispiel könnte der Benutzer dem Proxy-Server seine Präferenzen senden. Dies kann durch eine spezielle Software erfolgen, die auf dem Client **40** ausgeführt wird, oder durch Verwendung spezieller Webseiten, die direkt durch den Proxy-Server **30** bereitgestellt werden und dem Benutzer ermöglichen, seine Präferenzen einzustellen.

[0053] In der ersten Analysephase kann der Proxy-Server **30** einem Objekt eine Anfangspriorität zuweisen, indem er einen HTML-Code analysiert, welcher durch den Client **40** beim Server **20** angefordert worden ist. Insbesondere können Verweise auf Objekte, auf die im HTML-Code verwiesen wird, zu diesem Zweck bewertet werden. Auf diese Weise kann eine Prioritätsliste erzeugt werden, in der einzelne Objekte, auf welche im HTML-Code verwiesen wird, in der folgenden Reihenfolge aufgelistet sind (Objekt mit höchster Priorität sind zuerst genannt):

- Links zu anderen Seiten
- Frames
- Inline-Grafiken (falls der IMG-Tag Breiten- und Höheninformationen enthält, können diese Informationen verwendet werden, um die Priorität von Bildern in Abhängigkeit von den erwarteten Abmessungen zu verfeinern, so dass kleinere Bilder eine höhere Priorität erhalten als größere)
- Stylesheets
- Scripts (JavaScript, VBScript usw.), eingebettete Objekte und Applets
- Hintergrundbild (Seitenhintergrund, Tabellenhintergrund, Stylesheet usw.)
- Ein Objekt, welches bereits zum Client **40** gesendet wurde, erhält die niedrigste Priorität.

[0054] Außerdem kann die Priorität eines bestimmten Objekts verringert werden, wenn sich das Objekt nicht auf demselben Server **20** oder in derselben Domain befindet wie die aktuelle HTML-Seite.

[0055] Eine weitere Möglichkeit, die Priorität eines Objektes zuzuweisen oder anzupassen, ergibt sich, wenn eine Anforderung für dieses Objekt vom Browser ausgegeben und durch den Proxy-Server **30** empfangen wird. In einem solchen Falle kann die HTTP-Anforderung durch den Proxy-Server **30** in Verbindung mit einer URL analysiert werden (zweite Analysephase). Da Anfangsprioritäten bereits während der Analyse des HTML-Codes, der zu dieser HTTP-Anforderung geführt hat, zugewiesen worden sind, hat die Analyse der HTTP-Anforderung gewöhnlich eine Anpassung der Anfangspriorität zum Ergebnis. In manchen Fällen können jedoch auch Anfangsprioritäten zugewiesen werden (siehe oben).

[0056] Die Anpassung oder Zuweisung von Anfangsprioritäten in der zweiten Analysephase kann in Abhängigkeit von zusätzlichen Informationen durchgeführt werden, die zum Beispiel im Header der HTTP-Anforderung verfügbar sind. Eine oder mehrere der folgenden Regeln zum Aktualisieren der Prioritäten können implementiert werden:

- Die Analyse führt zu dem Ergebnis, dass der Browser dasselbe Objekt schon einmal angefordert hat. Einem solchen Objekt wird vorzugsweise eine sehr hohe Priorität zugewiesen, um Endlosschleifen zu vermeiden, die von Browsern verursacht werden, welche nicht vollständig HTTP/1.1-kompatibel sind und den Retry-After Header ignorieren (dieser Header wird weiter unten ausführlicher erörtert).
- Das Objekt hat noch keine Priorität, doch die Dateinamenerweiterung sieht wie HTML (".HTML", ".HTM") oder XML (".XML") aus oder sieht wie ein Verzeichnisindex aus (endet mit "/"). Eine solche Prioritätszuweisung stellt sicher, dass eine von den Favoriten aus angeforderte oder manuell eingegebene HTML-Seite mit einer hohen Priorität angefordert wird.
- Der Browser führt eine bedingte HTTP-Anforderung für ein Objekt durch, unter Verwendung von Bedingungen der Art "falls geändert seit" oder ähnlichen Bedingungen. Dies deutet darauf hin, dass der Browser wahrscheinlich eine Kopie des Objekts "gecacht" hat. Es ist zu erwarten, dass die Antwort von geringem Umfang ist, wenn die "gecachte" Kopie noch gültig ist (HTTP-Antwortcode "304 not modified").
- Die URL des angeforderten Objektes wurde beim Parsing einer vorhergehenden HTML-Seite gefunden.
- Ein Objekt, welches beim Parsing der HTML-Tags einer vorhergehenden Seite nicht in die Liste aufgenommen wurde, wurde wahrscheinlich indirekt von einem Script angefordert

und sollte eine niedrigere Priorität erhalten als die meisten anderen angeforderten Objekte.

[0057] In der dritten Analysephase beruht die Anpassung der Prioritäten von Objekten auf einer Analyse der HTTP-Antwort vom Server **20**. Die Anpassung der Prioritäten in der dritten Analysephase (ebenso wie in der oben erwähnten zweiten Analysephase) kann als eine gewichtete Summe mehrerer Kriterien berechnet werden. Die Gewichte können durch den Betreiber des Proxy-Servers **30** oder den Benutzer des auf dem Client **40** ausgeführten Browsers konfigurierbar sein.

[0058] In der dritten Analysephase werden gewöhnlich alle Objekte eine ihnen zugewiesene Priorität haben. Diese Priorität kann jedoch aktualisiert werden, bevor die angeforderten Objekte zum Client **40** gesendet werden. Um die Prioritäten anzupassen, können die Header und die Inhalte der HTTP-Antworten, die vom Server **20** empfangen wurden, bewertet werden. Die Prioritäten können danach entsprechend einer oder mehreren der folgenden Regeln angepasst werden:

- Bewertung des Antwortcodes: Der relative Priorität der HTTP-Antworten hängt von der ersten Ziffer des Antwortcodes ab. Fehlercodes (4xx, 5xx) sollten eine höhere Priorität haben als normale Antworten (2xx).
- Bewertung des Typs des Inhalts: Ein HTML-Code sollte eine höhere Priorität haben als irgendein Bild.
- Bewertung der Größe der Objekte (abgeleitet von der Inhaltslänge, falls in den Headern angegeben, oder von der Gesamtgröße, falls das Objekt bereits "gecacht" ist): Kleinere Objekte sollten eine etwas höhere Priorität haben als große.
- Analyse des Inhalts: Zum Beispiel kann animierten Bildern eine niedrigere Priorität zugewiesen werden als statischen Bildern. Die Analyse des Inhalts kann auch die Grundlage für die Schätzung der Größe des Objektes bilden, falls diese Information nicht im HTTP-Header verfügbar ist und das Objekt noch nicht "gecacht" worden ist.
- Falls der Antwortcode eine permanente oder zeitweilige Umleitung (3xx) ist, welche einen neuen Speicherort für das angeforderte Objekt festlegt, so erhält dieser neue Speicherort dieselbe Priorität wie das ursprüngliche Objekt.

[0059] In den oben beschriebenen drei Analysephasen legt der Proxy-Server **30** die Prioritäten der Objekte fest und aktualisiert sie, welche zum Client **40** zu übertragen sind. Der Proxy-Server **30** bewahrt somit gewisse Informationen über jedes Objekt auf, wie die URL des Objekts, die Priorität, den Zeitpunkt der letzten Anforderung und, falls erforderlich, einige weitere Parameter. Diese Informationen können jedoch nicht für immer aufbewahrt werden, da andernfalls der Speicher des Proxy-Servers **30** überlaufen wür-

de. Außerdem werden, falls ein Objekt, welchem eine hohe Priorität zugewiesen worden ist, niemals angefordert wird, vorzugsweise Maßnahmen ergriffen, damit es nicht verhindert, dass andere Objekte übertragen werden.

[0060] Aus diesen Gründen ist eine Routine implementiert, welche sicherstellt, dass Informationen, welche nicht mehr aktuell sind oder welche nicht mehr benötigt werden, gelöscht werden. Zu diesem Zweck können einer oder mehrere der folgenden Mechanismen verwendet werden:

- Ein Objekt, welches erfolgreich zum Client übertragen worden ist, wird als "gesendet" markiert oder wird auf eine separate Liste von Objekten, welche zum Client **40** gesendet worden sind, bewegt.
- Es wird eine maximale Größe für eine oder mehrere Listen, welche relevante Informationen enthalten, festgelegt und so konfiguriert, dass ältere Objekte oder Objekte mit der niedrigsten Priorität zuerst aus der Liste gelöscht werden.
- Jedes Mal, wenn der Client eine TCP-Verbindung zurücksetzt, bevor das Objekt vollständig übertragen ist (was bedeutet, dass der Benutzer den Download abgebrochen hat und eventuell eine andere Seite gewählt hat), die Informationen von zu sendenden Objekten löschen.
- Als Alternative zur vorhergehenden Lösung Querverweise für jedes Objekt aufbewahren, um jede HTML-Seite mit den Objekten zu verknüpfen, welche sie enthält, und umgekehrt. Wenn der Client **40** eine TCP-Verbindung zurücksetzt, nur diejenigen Objekte entfernen, welche in derselben HTML-Seite enthalten sind.
- Die Priorität aller Objekte kann nach Ablauf einer bestimmten Zeit, oder nachdem eine bestimmte Anzahl von HTTP-Anforderungen oder HTTP-Antworten verarbeitet worden ist, verringert werden.

UMORDNEN VON OBJEKTEN

[0061] Im vorhergehenden Kapitel wurden die Erzeugung einer Prioritätsliste für die angeforderten Objekte, die zum Client **40** zu übertragen sind, sowie mögliche Aktualisierungsmechanismen für diese Prioritätsliste beschrieben. Gewöhnlich werden die angeforderten Objekte durch den Proxy-Server **30** vom Server **20** in einer Reihenfolge empfangen, welche von der Reihenfolge verschieden ist, die in der Prioritätsliste angegeben ist. Folglich muss der Proxy-Server **30** die vom Server **20** empfangenen Objekte auf eine solche Weise umordnen, dass sie vom Proxy-Server **30** zum Client **40** in einer Reihenfolge weitergeleitet werden, welche die Reihenfolge in der Prioritätsliste so gut wie möglich widerspiegelt. Der Proxy-Server **30** ordnet die vom Server **20** empfangenen Objekte um, indem er Objekte mit einer niedrigeren Priorität absichtlich verzögert und indem er Objekte

mit einer höheren Priorität ohne irgendeine wesentliche Verzögerung zum Client **40** weiterleitet.

[0062] Der Proxy-Server **30** verwendet eine Kombination von verschiedenen Verzögerungsmechanismen, um die vom Server **20** empfangenen Objekte umzuordnen. Im Folgenden werden zwei dieser Verzögerungsmechanismen, nämlich Sperren von TCP-Verbindungen einerseits und HTTP-Umleitungen andererseits, beispielhaft genauer beschrieben.

Sperren von TCP-Verbindungen

[0063] Wie aus [Fig. 2](#) ersichtlich ist, wird die Übertragungsstrecke **14** zwischen dem Proxy-Server **30** und dem Client **40** durch mehrere TCP-Verbindungen **50** gebildet. In Abhängigkeit von den Prioritäten der angeforderten Objekte, welche über einzelne Verbindungen von den Verbindungen **50** weiterzuleiten sind, werden eine oder mehrere der Verbindungen **50**, welche für den Transfer von Objekten mit einer niedrigen Priorität bestimmt sind, gesperrt. Durch diese Sperrung einzelner Verbindungen wird eine gewisse Bandbreite auf der Übertragungsstrecke **14** freigegeben, welche nun verfügbar ist, um vorzugsweise Objekte mit einer höheren Priorität zu übertragen. Demzufolge werden Objekte mit einer höheren Priorität vor Objekten mit einer niedrigeren Priorität übermittelt.

[0064] Die Sperrung einer einzelnen Verbindung **50** wird so durchgeführt, dass die Verbindung **50** für eine gewisse Zeitdauer geöffnet gelassen wird, ohne dass Objekte übertragen werden. Stattdessen könnte die Sperrung einer Verbindung **50** auch durchgeführt werden, indem die Verbindung **50** geschlossen wird, entweder mit oder ohne Speichern des Zustands der Verbindung **50** vor ihrer Sperrung. Wenn der Zustand der gesperrten Verbindung **50** gespeichert wird, kann sie zu einem späteren Zeitpunkt in demselben Zustand geöffnet werden, in dem sie gesperrt (d.h. geschlossen) wurde.

[0065] Die Sperrung einer Verbindung **50** hat den Vorteil, dass keine zusätzlichen Daten über die Übertragungsstrecke **14** gesendet werden müssen. Vorzugsweise wird eine Verbindung **50** nur dann gesperrt, wenn die freigegebene Bandbreite vollständig durch andere Verbindungen **50** für den Transfer von Objekten mit einer hohen Priorität verwendet werden kann.

[0066] Der Proxy-Server **30** ist folglich derart konfiguriert, dass er prüft, ob die Übertragungsstrecke **14** vollständig genutzt wird, bevor er eine Verbindung **50** sperrt, um verfügbare Bandbreite nicht zu verschwenden. Eine mögliche Routine zum Vorhersagen, ob die Übertragungsstrecke **14** teilweise ungenutzt ist oder sein wird, beinhaltet das Vergleichen des Durchsatzes (während der letzten N Sekunden)

aller Verbindungen **50**, die zum Client **40** führen, mit der Datenmenge, welche gegenwärtig im Puffer **36** des Proxy-Servers **30** (siehe [Fig. 3](#)) "gecacht" oder zwischengespeichert ist und bereit ist, gesendet zu werden. Ebenso gut könnten andere und insbesondere einfachere Verfahren zum Schätzen der Ausnutzung der Übertragungsstrecke **14** implementiert werden, insbesondere wenn die verfügbare Bandbreite auf der Übertragungsstrecke **14** durch andere Mittel bekannt ist.

HTTP-Umleitungen

[0067] Falls der Proxy-Server **30** erwartet, dass bald Objekte mit einer hohen Priorität zum Client **14** übertragen werden müssen (z.B. weil zuvor vom Proxy-Server **30** während des Weiterleitens des HTML-Codes einer HTML-Seite zum Client **40** Übertragungsstrecken zu diesen Objekten erkannt worden sind), und eine Sperrung einer Verbindung **50** eine gewisse Bandbreite verschwenden würde, da gegenwärtig auf den anderen Verbindungen **50** nichts anderes zu übertragen ist, kann der Proxy-Server **30** ein anderes Verzögerungsschema benutzen. Ein mögliches Verzögerungsschema, welches in einem solchen Falle anwendbar ist, beinhaltet das Anweisen des Clients **40**, eine Objektorforderung zu einem späteren Zeitpunkt zu wiederholen.

[0068] Obwohl HTTP festlegt, dass alle HTTP-Antworten zum Client **40** in derselben Reihenfolge gesendet werden müssen, in der der Browser, der auf dem Client **40** ausgeführt wird, die HTTP-Anforderungen ausgegeben hat, ist es möglich, den Browser anzuweisen, eine HTTP-Anforderung später erneut zu versuchen. Dies kann geschehen, indem eine HTTP-Antwort mit dem Statuscode "302" an den Client **40** gesendet wird, zusammen mit dem Header-Feld "Retry-After". Dieses Header-Feld fordert den Client **40** auf, seine HTTP-Anforderung nach Ablauf einer festgelegten Zeit erneut zu versuchen. In Reaktion auf den Empfang des Statuscodes "302" (der möglicherweise das Header-Feld "Retry-After" enthält) verschieben gegenwärtige Browser die HTTP-Anforderung auf einen Zeitpunkt, nachdem alle noch anstehenden HTTP-Anforderungen verarbeitet worden sind.

[0069] Der Statuscode "302", so wie er im HTTP-Standard 1.1 spezifiziert ist, teilt dem Browser mit, dass ein gegebenes Objekt (zeitweilig) an einem Speicherort zu finden ist, welcher von dem, der angefordert wurde, verschieden ist. Die zum Client **40** gesendete HTTP-Antwort enthält den neuen Speicherort des Objektes. Dieser Mechanismus wird verwendet, um das Verzögerungsschema gemäß der Erfindung zu implementieren.

[0070] Gemäß der Erfindung erzeugt der Proxy-Server **30** ein Attribut in Form einer neuen (virtuel-

len) URL für das zu verzögernde Objekt. Danach weist der Proxy-Server **30** den auf dem Client **40** ausgeführten Browser an, die HTTP-Anforderung erneut zu versuchen und dabei dieses temporäre Attribut (d.h. die URL) als neuen Speicherort des Objektes zu verwenden. Der Browser wird somit angewiesen, die HTTP-Anforderung auf einen späteren Zeitpunkt zu verschieben. Wenn der Browser seine HTTP-Anforderung (welche das temporäre Attribut, d.h. die virtuelle URL enthält) wiederholt, konvertiert der Proxy-Server **30** das Attribut in die ursprüngliche URL und leitet die HTTP-Anforderung zum Server **20** weiter. Stattdessen könnte auch die ursprüngliche HTTP-Anforderung zum Server **20** weitergeleitet worden sein, nachdem der Proxy-Server **30** sie empfangen hat. In einem solchen Falle wird die vom Server **20** empfangene HTTP-Antwort zeitweilig vom Proxy-Server **30** gespeichert, bis dieser die wiederholte HTTP-Anforderung vom Client **40** empfängt.

[0071] Im Folgenden wird der oben erörterte erfindungsgemäße Verzögerungsmechanismus unter Bezugnahme auf die [Fig. 4](#) und [Fig. 5](#) beispielhaft ausführlicher beschrieben. Es wird angenommen, dass die HTTP-Anforderung, die der Proxy-Server **30** vom Client **40** empfängt, sich auf ein Objekt bezieht, welches vom Proxy-Server **30** als eine niedrige Priorität aufweisend betrachtet wird.

[0072] In einem ersten Schritt **402** von [Fig. 4](#) empfängt der Proxy-Server **30** eine HTTP-Anforderung vom Client **40** über die Übertragungsstrecke **14**. In dem beispielhaften Fall von HTTP könnte diese erste Anforderung vom Client das folgende Format haben:
GET http://example.com/some/image.png HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (X11; Linux i686; en-US; rv:0.9.9) Gecko/20020311
Connection: close

[0073] In Reaktion auf die im Schritt **402** vom Client **40** empfangene HTTP-Anforderung leitet der Proxy-Server **30** den Client **40** im Schritt **404** zu einem neuen (virtuellen) Speicherort um und weist den Client **40** an, eine gewisse Zeit (Verzögerung) zu warten, bevor er die HTTP-Anforderung erneut versucht. Die Umleitungsnachricht vom Proxy-Server **30** auf der Basis des Statuscodes "302", so wie er im HTTP-Standard 1.1 spezifiziert ist, kann das folgende Format haben:

```
HTTP/1.1 302 Found
Date: Thu, 21 Mar 2002 15:12:47
Server: PrioTest/0.9
Location: http/example.com/some/()image(0001).png
Retry-After: 5
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset = iso-8859-1
(Es folgt etwas visuell lesbarer Text)
```

[0074] Der Proxy-Server **30** kann nun jederzeit nach dem Empfang der anfänglichen HTTP-Anforderung vom Client **40** (Schritt **402**) und vor dem Zeitpunkt, zu dem er beschließt, das Objekt zum Client **40** zu übermitteln, das Objekt vom Server **20** anfordern (Schritte **406** und **408**).

[0075] Nach Empfang der obigen Umleitungsnachricht wartet der Client **40** die in der Umleitungsnachricht festgelegte Zeit ab, d.h. fünf Sekunden (oder bis alle anderen Objekte übertragen worden sind). Im Schritt **412** wiederholt der Client **40** dann seine HTTP-Anforderung, wobei er den neuen (virtuellen) Speicherort wie folgt angibt:

```
GET http://example.com/some/()image(00001).png
HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (X11; Linux i686; en-US; rv:
0.9.9) Gecko/20020311
Connection: close
```

[0076] Nach dem Empfang der wiederholten HTTP-Anforderung vom Client **40** entscheidet der Proxy-Server **30**, ob das angeforderte Objekt noch weiter verzögert werden soll (z.B. durch Sperren einer Verbindung **50** oder durch eine weitere Umleitungsnachricht), oder ob das verzögerte Objekt zum Client **40** weitergeleitet werden soll. Falls der Proxy-Server **30** entscheidet, das verzögerte Objekt ohne eine weitere Verzögerung weiterzuleiten, so kann dies im folgenden Format geschehen:

```
HTTP/1.1 200 KO
Date: Thu, 21 Mar 2002 15:12:50
Server: Apache/1.3.23 (Unix)
Content-Type: image/png
Content-Length: 1520
(Der Inhalt des Bildes folgt)
```

[0077] Nachfolgend werden die Entscheidungen, die vom Proxy-Server **30** im Verlaufe der oben unter Bezugnahme auf [Fig. 4](#) erörterten Umleitungsroutine getroffen werden, unter Bezugnahme auf [Fig. 5](#) erläutert.

[0078] Im Schritt **502** empfängt der Proxy-Server **30** die HTTP-Anforderung vom Client **40**. Im folgenden Schritt **504** bestimmt der Proxy-Server **30**, ob die in der HTTP-Anforderung enthaltene URL eine modifizierte URL ist, d.h. das Ergebnis einer vorhergehenden Umleitung. Falls dies der Fall ist, decodiert der Proxy-Server **30** im Schritt **506** die URL und stellt den ursprünglichen Speicherort wieder her und geht dann über den Knoten **508** zu Schritt **510**. Andernfalls führt der Ablauf des Verfahrens direkt vom Schritt **504** über den Knoten **508** zum Schritt **510**.

[0079] Im Schritt **510** bestimmt der Proxy-Server **30**, ob das mittels der HTTP-Anforderung angeforderte Objekt bereits verfügbar, d.h. im Puffer **36** des Proxy-Servers **30** (siehe [Fig. 3](#)) gespeichert ist. Falls das

Objekt noch nicht "gecacht" ist, fordert der Proxy-Server **30** im Schritt **514** das Objekt vom Server **20** an oder markiert es für einen späteren Abruf. Vom Schritt **514** führt der Ablauf des Verfahrens über den Knoten **512** zum Schritt **516**. Der Schritt **516** kann auch direkt vom Schritt **510** über den Knoten **512** erreicht werden, falls das durch den Client **40** angeforderte Objekt bereits verfügbar ist.

[0080] Im Schritt **516** ist die das angeforderte Objekt enthaltende HTTP-Antwort bereit, zum Client **40** gesendet zu werden. Dies entspricht dem ersten Schritt im Flussdiagramm von [Fig. 6](#), welches weiter unten erörtert wird.

[0081] Im Prinzip kann die Umleitungsnachricht (Schritt **404** in [Fig. 4](#)) zu einem beliebigen Zeitpunkt, während die Schritte **502** bis **516** ausgeführt werden, oder nach dem Schritt **516** zum Client **40** gesendet werden. Es ist anzumerken, dass die Verfügbarkeit eines Objekts (d.h. ob ein angefordertes Objekt bereits "gecacht" ist, gegenwärtig übertragen wird oder ob das Objekt noch nicht angefordert worden ist) ein weiterer Faktor ist, welcher die Anfangspriorität eines angeforderten Objekts beeinflusst.

Umordnungsentscheidungen

[0082] Sobald die HTTP-Antwort bereit ist, zum Client **40** gesendet zu werden (Schritt **516** in [Fig. 5](#)), muss entschieden werden, ob ein Objekt, das in der HTTP-Antwort enthalten ist, tatsächlich zum Client **40** übermittelt werden sollte, ob der Client **40** umgeleitet werden sollte oder ob die Verbindung **50**, über welche dieses Objekt zum Client **40** zu senden ist, gesperrt werden sollte. Ein Flussdiagramm, welches ein diesbezügliches beispielhaftes Entscheidungsschema darstellt, ist in [Fig. 6](#) abgebildet.

[0083] Im Schritt **602** ist die HTTP-Antwort bereit, zum Client gesendet zu werden. Im folgenden Schritt **604** wird die Priorität dieses Objektes in Bezug auf die Tatsache beurteilt, ob die Priorität aktualisiert werden muss oder nicht. Falls die Priorität des Objektes aktualisiert werden muss, wird die Priorität im Schritt **604** angepasst.

[0084] Im nächsten Schritt **606** wird die aktuelle Priorität des Objektes bewertet, um zu ermitteln, ob es die höchste Priorität unter allen Objekten aufweist, welche gerade übertragen werden oder welche demnächst erwartet werden. Falls im Schritt **606** ermittelt wird, dass das angeforderte Objekt tatsächlich die höchste Priorität aufweist, führt der Ablauf des Verfahrens über den Knoten **610** zum Schritt **612**. Im Schritt **612** wird das angeforderte Objekt zum Client **40** gesendet.

[0085] Falls der Vergleich der Priorität des gegenwärtig angeforderten Objektes mit der Priorität ande-

rer Objekte im Schritt **606** zu dem Ergebnis führt, dass das gegenwärtig angeforderte Objekt nicht die höchste Priorität aufweist, wird das Verfahren mit Schritt **614** fortgesetzt. In Bezug auf Schritt **606** ist anzumerken, dass, anstatt die Priorität des gegenwärtig angeforderten Objektes mit den Prioritäten anderer Objekte zu vergleichen, es auch möglich ist, die Priorität des gegenwärtig angeforderten Objektes mit einem festen Schwellwert zu vergleichen oder beim Vergleich einen Abstand zu addieren, so dass zwei Prioritäten sich um mehr als einen vorgegebenen Schwellwert unterscheiden müssen, damit die Differenz als ausreichend signifikant betrachtet wird.

[0086] Im Schritt **614** schätzt der Proxy-Server **30** ein, ob die Übertragungsstrecke **14** zum Client **40** (siehe [Fig. 2](#)) ungenutzt ist oder sein wird. Wie oben erläutert wurde, gibt es verschiedene Wege, um dies zu tun. Einer von ihnen umfasst das Berechnen eines gleitenden Mittelwertes der maximalen Datenmenge, welche während der letzten N Sekunden über alle Verbindungen **50**, die zu demselben Client **40** führen, gesendet und quittiert wurde. Dies liefert eine Schätzung des verfügbaren maximalen Durchsatzes.

[0087] Das auf diese Weise erhaltene Ergebnis wird danach mit der Datenmenge verglichen, welche bereit steht, um auf den Verbindungen **50**, welche nicht gesperrt sind, gesendet zu werden. Falls der Proxy-Server **30** erkennt, dass er nicht genügend Daten zu senden hat, um die Übertragungsstrecke während wenigstens einer kompletten Umlaufzeit zu füllen, so nimmt er an, dass eine gewisse Reserve an Bandbreite auf der Übertragungsstrecke **14** zum Client **40** vorhanden ist, und fährt mit Schritt **616** fort.

[0088] Im Schritt **616** wird ein Vergleich durchgeführt, der dem von Schritt **606** ähnlich ist. Falls im Schritt **616** ermittelt wird, dass das gegenwärtig angeforderte Objekt eine höhere Priorität besitzt als alle Objekte, die demnächst erwartet werden, fährt der Proxy-Server **30** über den Knoten **610** mit Schritt **612** fort und sendet das gegenwärtig angeforderte Objekt sofort zum Client **40**. Andernfalls wird das Verfahren mit Schritt **618** fortgesetzt, und es wird eine Umleitungsnachricht (Statuscode "302") an den Client **40** gesendet, wie oben in Verbindung mit [Fig. 4](#) erörtert wurde.

[0089] Die Entscheidung, die im Schritt **616** getroffen wird, kann von der Größe des gegenwärtig angeforderten Objektes (falls die bereits bekannt ist) beeinflusst werden. Falls bekannt ist, dass das Objekt klein ist (z.B. kleiner als das Doppelte der Größe einer Umleitungsnachricht), sende der Proxy-Server immer das Objekt anstelle des Sendens einer Umleitungsnachricht, selbst wenn dem Objekt bis dahin eine sehr niedrige Priorität zugewiesen war (dies kann als ein Weg des Aktualisierens der Priorität angesehen werden).

[0090] Falls im Schritt **614** ermittelt wird, dass genügend zu sendende Daten vorhanden sind, um die Übertragungsstrecke **14** zu füllen, fährt das Verfahren mit Schritt **620** fort und sperrt die Verbindung **50** zum Client **40**, über welche das gegenwärtig angeforderte Objekt übertragen werden soll.

[0091] Nach jedem der Schritte **612**, **618** und **620** setzt sich das Verfahren mit Schritt **622** fort. Im Schritt **622** bewertet der Proxy-Server **30** alle gesperrten Verbindungen **50** neu, um zu ermitteln, ob irgendwelche von den gesperrten Verbindungen **50** wieder zu öffnen sind.

[0092] Das Protokoll HTTP/1.1 unterstützt die Pipelining-Option, welche dem Client **40** ermöglicht, mehrere Anforderungen an den Server **20** oder den Proxy-Server **30** auf derselben TCP-Verbindung zu senden, ohne die vorhergehenden Antworten abzuwarten. In dem Falle, wenn Pipelining verwendet wird, könnten mehrere Objekte, welche bereit sind, gesendet zu werden, auf derselben Verbindung **50** zum Client **40** gesendet werden. Es ist offensichtlich, dass in einem solchen Falle ein Objekt mit einer niedrigen Priorität ein zweites Objekt mit einer höheren Priorität, welches auf derselben Verbindung **50** gesendet werden soll, blockieren könnte. Falls mehrere wartende Objekte auf derselben Verbindung **50** vorhanden sind, kann in den Schritten **606** und **616** die maximale oder mittlere Priorität dieser Objekte bestimmt und betrachtet werden. Dies stellt sicher, dass Objekte mit niedrigerer Priorität nicht Objekte mit höherer Priorität blockieren.

MÖGLICHE ERWEITERUNGEN

[0093] Es könnten verschiedene Erweiterungen der unter Bezugnahme auf [Fig. 1](#) bis [Fig. 6](#) beschriebenen beispielhaften Ausführungsform implementiert werden, von denen einige bereits kurz erörtert wurden.

[0094] Eine dieser Erweiterungen betrifft das Beschränken der Übertragungsgeschwindigkeit auf der Übertragungsstrecke **14** in Abhängigkeit von der Priorität der zu übertragenden Objekte. Dies könnte zum Beispiel dadurch geschehen, dass jeder Verbindung **50** in Abhängigkeit von der Priorität der zu übertragenden Objekte ein bestimmter Anteil der Verarbeitungskapazitäten zugewiesen wird. Die Priorität der einzelnen Objekte wird verringert, während der Prozess abläuft. Falls der Proxy-Server **30** Verbindungen **50** nach dem Round-Robin-Prinzip verarbeitet, kann dieses Merkmal implementiert werden, indem die Datenmenge, die pro Runde auf jeder Verbindung **50** übertragen wird, durch eine Zahl begrenzt wird, die von der Priorität des betreffenden Objektes abgeleitet wird. Die dynamische Zuweisung von Verarbeitungskapazitäten kann vorteilhaft mit den oben erörterten Mechanismen der Sperrung und

der Umleitung kombiniert werden. Die Verarbeitungskapazitäten können auch irgendeine Transformation der Objekte oder Codes, welche übertragen werden, einschließen.

[0095] Gemäß einer weiteren Erweiterung der Erfindung werden die Prioritäten direkt durch den Browser zugewiesen, der auf dem Client **40** ausgeführt wird. Der Browser kann dann die HTTP-Anforderungen für einzelne Objekte entsprechend ihrer Priorität zeitlich planen. Falls die Prioritäten direkt durch den Browser zugewiesen werden, gibt es einige zusätzliche Faktoren, die benutzt werden können, um die Priorität der einzelnen Objekte zu verfeinern:

- Position des Objektes in der Seite (Koordinaten);
- relative Position des sichtbaren Bereiches (Objekte, welche sich außerhalb des sichtbaren Bereiches befinden, haben eine niedrigere Priorität);
- falls das Laden von Bildern oder die Ausführung von Scripts deaktiviert ist, weiß der Browser sofort, dass es nicht notwendig ist, diesen Objekten eine Priorität zuzuweisen.

[0096] Außerdem weiß es der Browser, wenn ein Benutzer eine neue HTML-Seite aus den Favoriten oder durch manuelle Eingabe einer neuen URL wählt. Der Browser kann somit Objekte, welche nicht mehr benötigt werden, aus der Liste löschen.

[0097] Eine weitere Erweiterung der Erfindung betrifft eine Proxy-Komponente, welche sich auf demselben Computer (Client) wie der Browser oder in der Nähe dieses Computers, was Netzverbindungen betrifft, befindet. Diese Situation ist in [Fig. 7](#) dargestellt.

[0098] Wie aus [Fig. 7](#) ersichtlich ist, umfasst der Client **40** nicht nur eine Browser-Komponente **42**, sondern auch eine zusätzliche Proxy-Komponente **30** (welche die in [Fig. 3](#) dargestellte Struktur aufweist), die mit der Browser-Komponente **42** kommuniziert. In einem solchen Falle hat die Übertragungsstrecke **14** zwischen der Proxy-Komponente **30** und der Browser-Komponente **42** eine wesentlich höhere Kapazität und kürzere Latenzzeit als die Übertragungsstrecke **12** zwischen der Proxy-Komponente und dem Server **20** (in [Fig. 7](#) nicht dargestellt).

[0099] Demzufolge beeinflusst die Proxy-Komponente **30** auf der Client-Seite den Anforderungsstrom, da sie, was den Strom der Antworten angeht, wenig unternehmen kann. Auf der Basis der Prioritätssinformationen, die von der aktuellen HTTP-Anforderung und vorhergehenden HTTP-Antworten (z.B. vorhergehenden HTML-Codes, die auf weitere Objekte verweisen) abgeleitet wurden, sowie einer Schätzung der Verfügbarkeit der Übertragungsstrecke entscheidet die Proxy-Komponente **30** auf der Client-Seite, ob eine HTTP-Anforderung zum Server weitergeleitet werden sollte und/oder ob sie sofort mit

einer Umleitungsnachricht antworten sollte.

[0100] Die zu treffenden Entscheidungen sind einfacher, als es im Flussdiagramm von [Fig. 6](#) dargestellt ist. Genauer, die Entscheidungen können darauf reduziert werden einzuschätzen, ob das gegenwärtig angeforderte Objekt eine niedrigere Priorität aufweist als einige der Objekte, die gerade übertragen werden oder deren Übertragung demnächst erwartet wird, und ob die Übertragungsstrecke bereits vollständig genutzt wird. Falls beide Bedingungen erfüllt sind, wird eine Umleitungsnachricht an die Browser-Komponente **42** gesendet. In allen anderen Fällen wird die HTTP-Anforderung sofort zum Server weitergeleitet.

[0101] Falls auf einer bestimmten Verbindung HTTP-Pipelining angewendet wird und falls gerade einige vorhergehende HTTP-Anforderungen verarbeitet werden, kann die Proxy-Komponente **30** auf der Client-Seite die HTTP-Anforderung sperren, bis sie die Entscheidung treffen kann. Sie muss eine Entscheidung spätestens dann treffen, wenn die vorhergehenden Objekte vollständig empfangen worden sind.

[0102] Eine weitere Erweiterung der Erfindung betrifft das Blockieren von Downloads für gewisse Objekte auf der Basis ihrer Priorität. Dies bedeutet, dass die Priorität, welche den Objekten zugewiesen wird, auch verwendet werden kann, um die Objekte vollständig zu blockieren. Falls ein Schwellwert für die Priorität festgelegt worden ist, wird ein beliebiges Objekt, welches eine Priorität besitzt, die niedriger als dieser Schwellwert ist, nicht zum Client **40** gesendet. In diesem Falle würde der Proxy-Server (oder die Proxy-Komponente) **30** einfach einen der Statuscodes "4xx" oder "5xx", die im Standard HTTP/1.1 definiert sind, an den Client zurücksenden, wenn er ein solches Objekt erkennt. Mögliche Statuscodes sind "403 Forbidden", "409 Conflict" oder "503 Service Unavailable".

[0103] Als eine zusätzliche Erweiterung könnten die den Objekten zugewiesenen Prioritäten für ein Prefetching (Vorabladen) verwendet werden. Ein Prefetching-Mechanismus ermöglicht dem Proxy-Server **30**, seinen Puffer **36** (siehe [Fig. 3](#)) mit einigen Objekten zu füllen, bevor der Client **40** sie tatsächlich anfordert. Ein solcher Prefetching-Mechanismus kann zum Beispiel auf der Analyse eines HTML-Codes beruhen, welcher vom Proxy-Server **30** zum Client **40** gesendet wird und welcher Verweise auf weitere Objekte enthält. Auf der Grundlage der Analyse der Verweise auf die Objekte weist die Proxy-Komponente **30** den einzelnen Objekten Prioritäten zu, und auf der Basis dieser Zuweisung wird definiert, welche Objekte vorabgeladen werden sollen und in welcher Reihenfolge. Vorzugsweise werden die Objekte beginnend mit den Objekten mit der höchsten Priorität vor-

abgeladen.

[0104] Gemäß einer weiteren Erweiterung, welche mit dem oben erörterten Prefetching-Mechanismus kombiniert werden kann, werden spezielle Push-Schemata implementiert, welche es ermöglichen, Objekte vom Proxy-Server **30** zum Client **40** zu übertragen, ohne dass eine explizite Objektanforderung vom Client **40** empfangen worden ist.

[0105] Es sind verschiedene Modifikationen der bevorzugten Ausführungsform möglich, ohne den Schutzbereich der vorliegenden Erfindung zu verlassen. Obwohl die Erfindung in Verbindung mit einer bevorzugten Ausführungsform beschrieben wurde, soll die Erfindung durch diese Beschreibung selbstverständlich nicht auf diese beschränkt werden. Vielmehr soll die Erfindung alle Modifikationen und/oder Ergänzungen der obigen Beschreibung umfassen, ohne den Schutzbereich der Erfindung zu verlassen.

Patentansprüche

1. Verfahren zum Steuern eines Objekttransfers in einem Kommunikationsnetz (**10**) von einer ersten Komponente (**20**) über eine Zwischenkomponente (**30**) zu einer zweiten Komponente (**40**), welche von der ersten Komponente (**20**) entfernt (remote) ist, wobei der Objekttransfer auf mehreren Objektanforderungen beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird, die von der zweiten (**40**) oder einer anderen Komponente des Kommunikationsnetzes (**10**) zu verarbeiten sind, wobei die Zwischenkomponente die folgenden Schritte ausführt:

- Senden einer Objektanforderung (**406**) an die erste Komponente (**20**);
- Empfangen des angeforderten Objektes (**408**) von der ersten Komponente (**20**);
- dadurch gekennzeichnet**, dass die Zwischenkomponente ferner die folgenden Schritte ausführt:
 - Bewerten und/oder Aktualisieren einer Priorität des angeforderten Objektes (**604**), wobei dem angeforderten Objekt auf der Basis einer Analyse der Objektanforderung und/oder des Codes, welcher auf das angeforderte Objekt verweist, eine Anfangspriorität zugewiesen worden ist; und
 - in Abhängigkeit von der Priorität des angeforderten Objektes Verzögern (**618; 404, 410, 412**) des angeforderten Objektes oder Weiterleiten (**612**) des angeforderten Objektes zur zweiten Komponente (**40**).

2. Verfahren nach Anspruch 1, wobei das Verzögern (**618; 404, 410, 412**) derart durchgeführt wird, dass sich die Reihenfolge, in welcher die Objekte von der ersten Komponente (**20**) empfangen werden, von der Reihenfolge unterscheidet, in welcher die Objekte zur zweiten Komponente (**40**) weitergeleitet werden.

3. Verfahren nach Anspruch 1 oder 2, wobei die Objektanforderung von der zweiten Komponente (**40**) empfangen wird (**402**) oder von der Zwischenkomponente (**30**) erzeugt wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, wobei das Verzögern (**618; 404, 410, 412**) des angeforderten Objektes mindestens einen der Schritte enthält: Anweisen der zweiten Komponente (**40**), die Objektanforderung zu wiederholen, Sperren einer Verbindung (**620**) zu der zweiten Komponente (**40**), über welche das angeforderte Objekt weitergeleitet werden soll, und Informieren der zweiten Komponente (**40**), dass das angeforderte Objekt zu einem späteren Zeitpunkt automatisch weitergeleitet wird.

5. Verfahren nach Anspruch 3, wobei das Anweisen der zweiten Komponente (**40**), die Objektanforderung zu wiederholen, beinhaltet:

- Zuweisen eines spezifischen Attributes zu dem zu verzögernden Objekt;
- Informieren der zweiten Komponente (**40**) über das Attribut;
- Empfangen eines Verweises auf das Attribut von der zweiten Komponente (**40**); und
- bei Empfang des Verweises auf das Attribut Senden des verzögerten Objektes zu der zweiten Komponente (**40**) oder weiteres Verzögern des verzögerten Objektes.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei angeforderte Objekte über mehrere Verbindungen (**50**) zu der zweiten Komponente (**40**) zur zweiten Komponente (**40**) weitergeleitet werden.

7. Verfahren nach Anspruch 6, wobei aus den Verbindungen (**50**) zur zweiten Komponente (**40**) ausgewählte Verbindungen in Abhängigkeit von den Prioritäten der angeforderten Objekte, welche von der ersten Komponente (**20**) empfangen wurden und welche über die aus den Verbindungen (**50**) ausgewählten Verbindungen weitergeleitet werden sollen, gesperrt werden.

8. Verfahren nach Anspruch 6 oder 7, wobei jeder Verbindung ein spezifischer Anteil an Verarbeitungskapazitäten dynamisch zugeordnet wird.

9. Verfahren nach einem der Ansprüche 1 bis 8, ferner umfassend:

- Senden einer Codeanforderung an die erste (**20**) oder eine dritte Komponente;
- Empfangen des angeforderten Codes von der ersten (**20**) oder der dritten Komponente;
- Analysieren des empfangenen Codes im Hinblick auf Verweise auf Objekte;
- Bewerten der Verweise auf Objekte zu dem Zweck, den Objekten, auf die in dem empfangenen Code verwiesen wird, Anfangsprioritäten zuzuweisen.

10. Verfahren nach einem der Ansprüche 1 bis 9, wobei bei Empfang einer das angeforderte Objekt enthaltenden Antwort von der ersten Komponente (20) die Antwort im Hinblick auf die Priorität des empfangenen Objektes ausgewertet wird, um zu bestimmen, ob die Anfangspriorität des empfangenen Objektes aktualisiert werden muss oder nicht.

11. Verfahren nach einem der Ansprüche 1 bis 10, ferner umfassend das Erzeugen einer Prioritätsliste, welche Prioritätsinformationen für einzelne Objekte oder Klassen von Objekten enthält.

12. Verfahren nach Anspruch 11, ferner umfassend ein wiederholtes Bewerten der Prioritätsliste im Hinblick auf das Aktualisieren von Prioritätsinformationen und/oder das Löschen von Objekten oder Klassen von Objekten oder entsprechenden Informationen aus der Prioritätsliste.

13. Verfahren nach einem der Ansprüche 1 bis 12, wobei die Schritte von einer Proxy-Komponente ausgeführt werden, die sich an der ersten Komponente oder an der zweiten Komponente befindet oder als eine separate Hardwarekomponente (30) des Kommunikationsnetzes konfiguriert ist.

14. Verfahren zum Verzögern eines Objekttransfers in einem Kommunikationsnetz (10) von einer ersten Komponente (20) über eine Zwischenkomponente (30) zu einer zweiten Komponente (40), welche von der ersten Komponente (20) entfernt (remote) ist, wobei der Objekttransfer auf mehreren Objekten beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird, die von der zweiten (40) oder einer anderen Komponente des Kommunikationsnetzes (10) zu verarbeiten sind, dadurch gekennzeichnet, dass die Zwischenkomponente (30) die folgenden Schritte ausführt:

- Zuweisen eines spezifischen Attributes zu einem Objekt, welches verzögert werden soll;
- Informieren der zweiten Komponente (40) über das Attribut;
- Empfangen eines Verweises auf das Attribut von der zweiten Komponente (40); und
- bei Empfang des Verweises auf das Attribut Senden des verzögerten Objektes, welchem das Attribut zugewiesen worden ist, zu der zweiten Komponente (40) oder weiteres Verzögern des verzögerten Objektes.

15. Verfahren nach Anspruch 14, wobei das Objekt zur zweiten Komponente (40) entsprechend einem Einspeicherungsschema (pushing scheme) oder in Reaktion auf eine von der zweiten Komponente (40) empfangene Objektenforderung gesendet wird.

16. Verfahren nach Anspruch 15, wobei die zweite Komponente (40) über das Attribut im Zusammen-

hang mit einer Anweisung informiert wird, die Objektenforderung zu wiederholen, und wobei der Verweis auf das Attribut von der zweiten Komponente (40) im Zusammenhang mit einer wiederholten Objektenforderung empfangen wird.

17. Computerprogrammprodukt, welches Programmcodeabschnitte zum Ausführen der Schritte der Ansprüche 1 bis 16, wenn das Computerprogrammprodukt auf einem Computersystem (30) ausgeführt wird, umfasst.

18. Computerprogrammprodukt nach Anspruch 17, das auf einem computerlesbaren Aufzeichnungsmedium gespeichert ist.

19. Zwischenkomponente (30) zum Steuern eines Objekttransfers in einem Kommunikationsnetz (10) von einer ersten Komponente (20) über die Zwischenkomponente (30) zu einer zweiten Komponente (40), welche von der ersten Komponente (20) entfernt (remote) ist, wobei der Objekttransfer auf mehreren Objekten beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird, die von der zweiten oder einer anderen Komponente des Kommunikationsnetzes zu verarbeiten sind, wobei die Zwischenkomponente (30) eine Kommunikationsschnittstelle (32) zum Senden einer Objektenforderung an die erste Komponente (20) und zum Empfangen des angeforderten Objektes von der ersten Komponente (20) umfasst, dadurch gekennzeichnet, dass die Zwischenkomponente (30) eine Verarbeitungseinheit (34) zum Bewerten und/oder Aktualisieren einer Priorität des angeforderten Objektes umfasst, wobei dem angeforderten Objekt auf der Basis einer Analyse der Objektenforderung und/oder des Codes, welcher auf das angeforderte Objekt verweist, eine Anfangspriorität zugewiesen worden ist, und wobei die Verarbeitungseinheit (34) in Abhängigkeit von der Priorität des angeforderten Objektes das angeforderte Objekt verzögert oder die Kommunikationsschnittstelle (32) zum Weiterleiten des angeforderten Objektes zu der zweiten Komponente (40) ansteuert.

20. Zwischenkomponente (30) zum Verzögern eines Objekttransfers in einem Kommunikationsnetz (10) von einer ersten Komponente (20) über die Zwischenkomponente (30) zu einer zweiten Komponente (40), welche von der ersten Komponente (20) entfernt ist, wobei der Objekttransfer auf mehreren Objekten beruht, die Objekte betreffen, auf die in einem oder mehreren Codes verwiesen wird, die von der zweiten oder einer anderen Komponente des Kommunikationsnetzes zu verarbeiten sind, dadurch gekennzeichnet, dass die Zwischenkomponente (30) eine Verarbeitungseinheit (34) zum Zuweisen eines spezifischen Attributes zu einem Objekt, welches zu verzögern ist, und eine Kommunikationsschnittstelle (32) zum Informieren der zweiten Kom-

ponente **(40)** über das Attribut, zum Empfangen eines Verweises auf das Attribut von der zweiten Komponente **(40)** und zum Senden des verzögerten Objektes, welchem das Attribut zugewiesen worden ist, bei Empfang des Verweises auf das Attribut zu der zweiten Komponente **(40)** oder zum weiteren Verzögern des verzögerten Objektes umfasst.

21. Zwischenkomponente nach Anspruch 19 oder 20, welche als Proxy-Server **(30)** konfiguriert ist.

22. Netzsystem **(10)**, welches wenigstens eine der Komponenten **(30)** nach Anspruch 19 bis 21 umfasst.

23. Netzsystem nach Anspruch 22, ferner umfassend eine erste Übertragungsstrecke **(12)** zwischen der Zwischenkomponente **(30)** und der ersten Komponente **(20)** und eine zweite Übertragungsstrecke **(14)** zwischen der Zwischenkomponente **(30)** und der zweiten Komponente **(40)**, wobei die erste Übertragungsstrecke **(12)** und die zweite Übertragungsstrecke **(14)** unterschiedliche Transfargeschwindigkeiten aufweisen.

24. Netzsystem nach Anspruch 22 oder 23, welches eine zweite Komponente **(40)** in Form eines mobilen Endgerätes umfasst.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

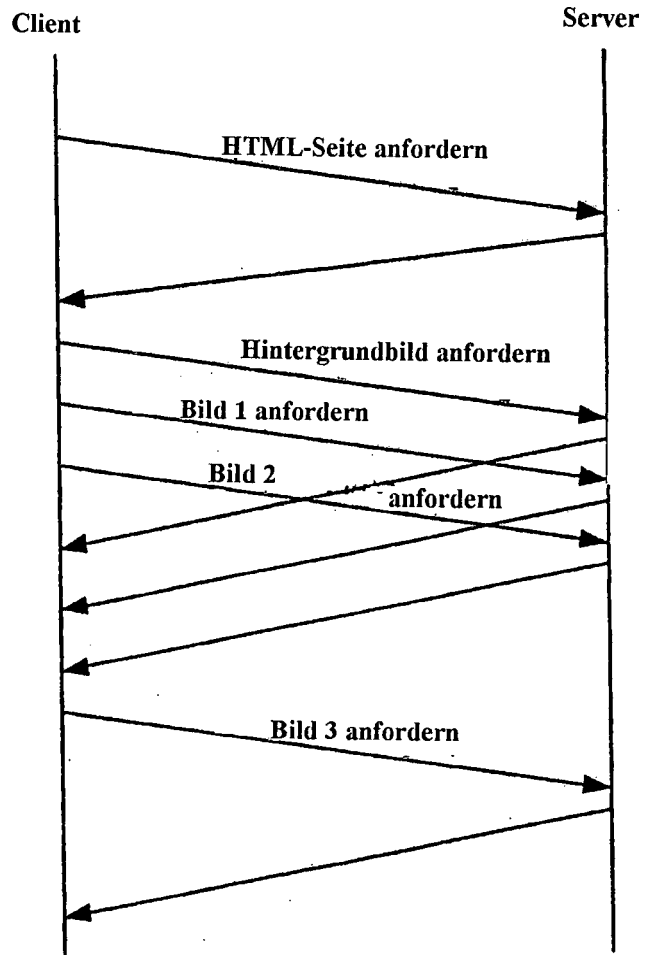


Fig. 1

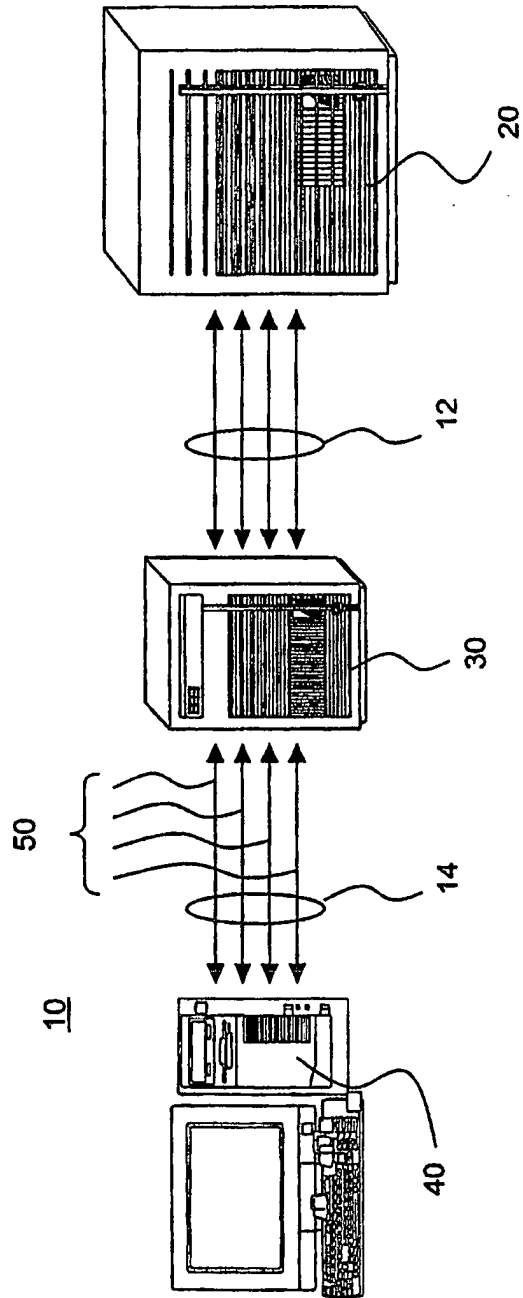


Fig. 2

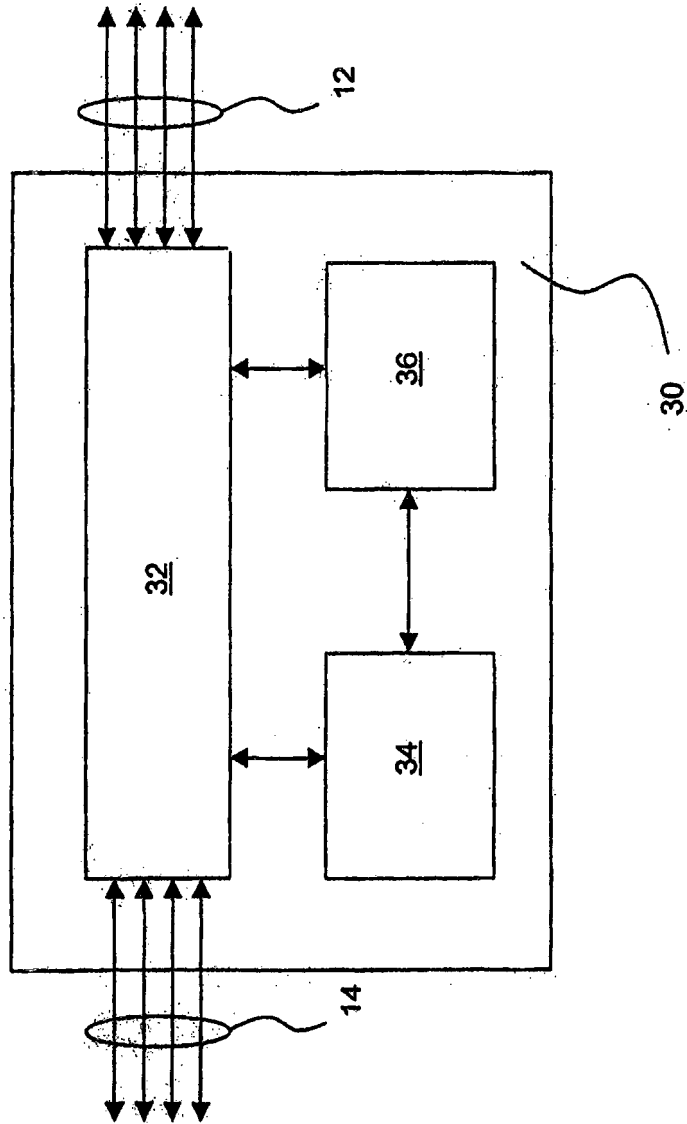


Fig. 3

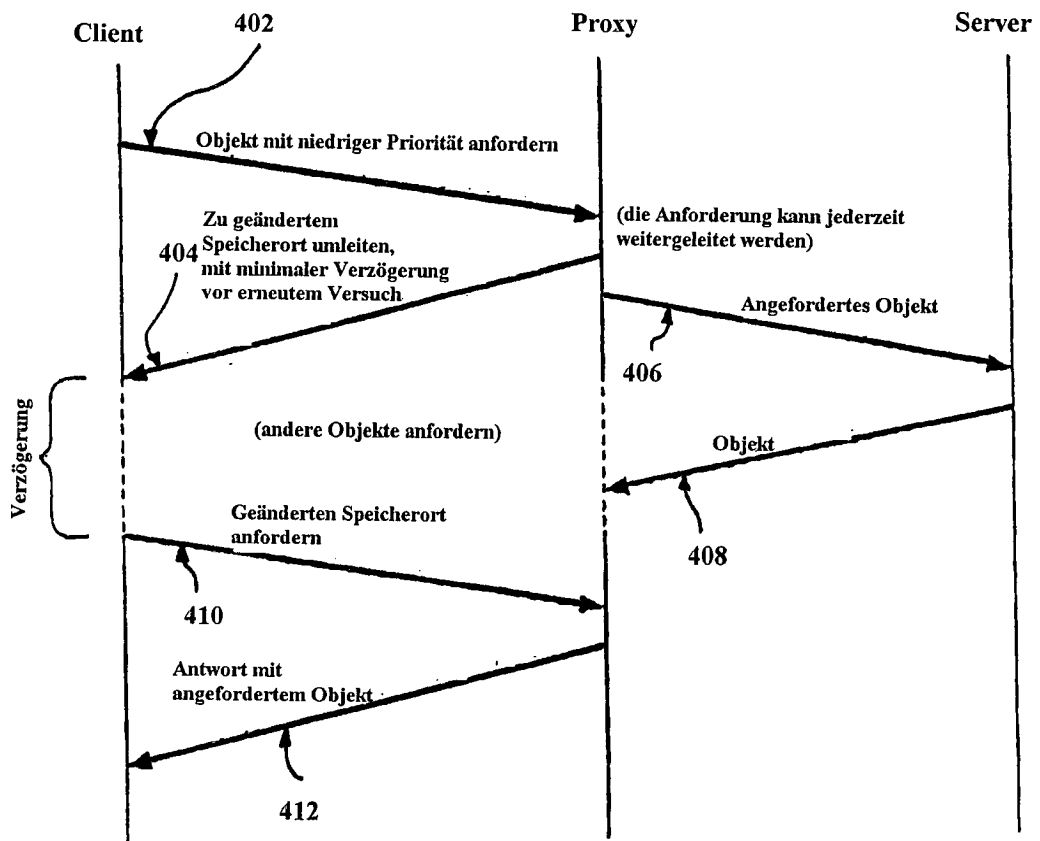


Fig. 4

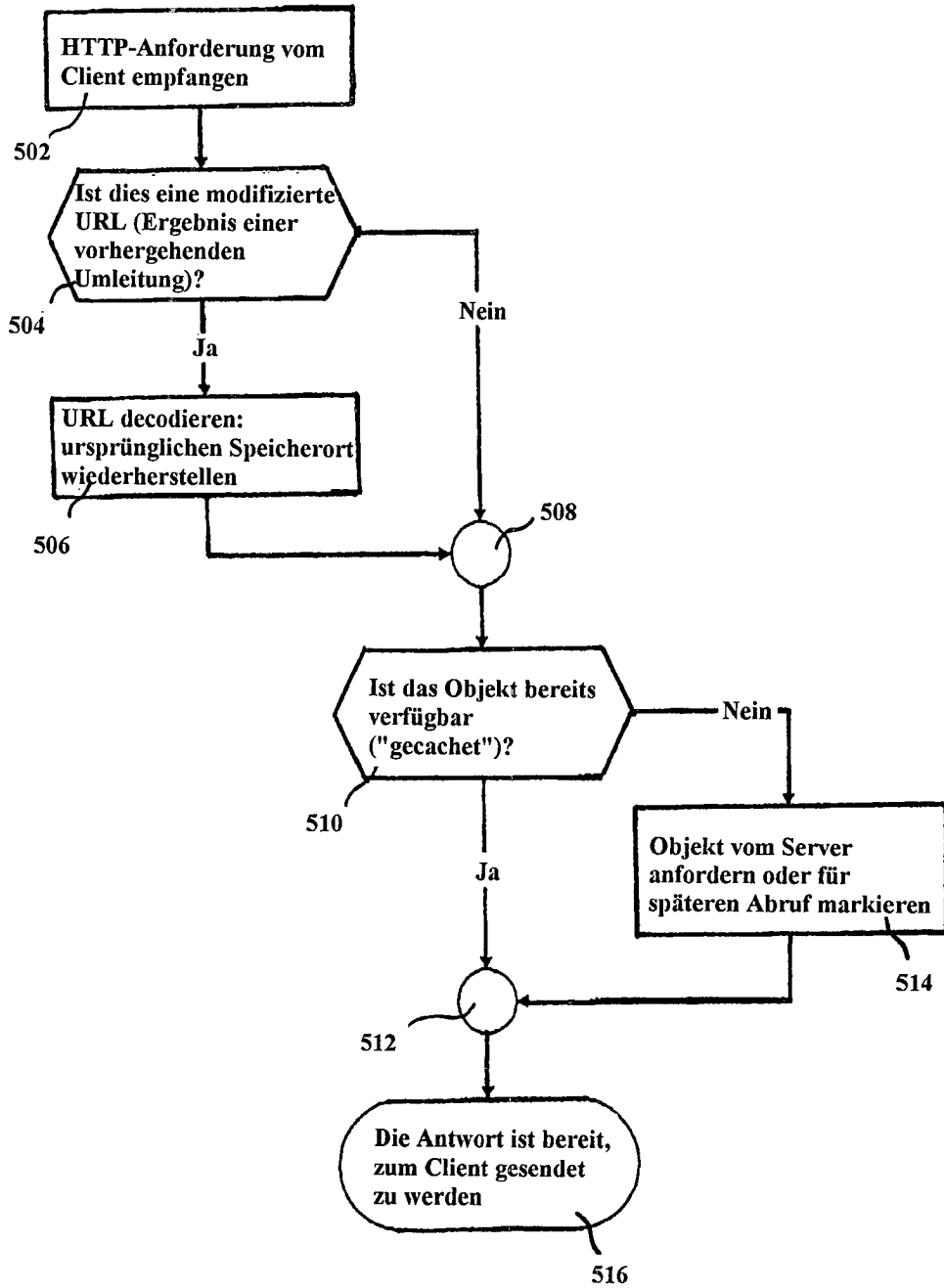


Fig. 5

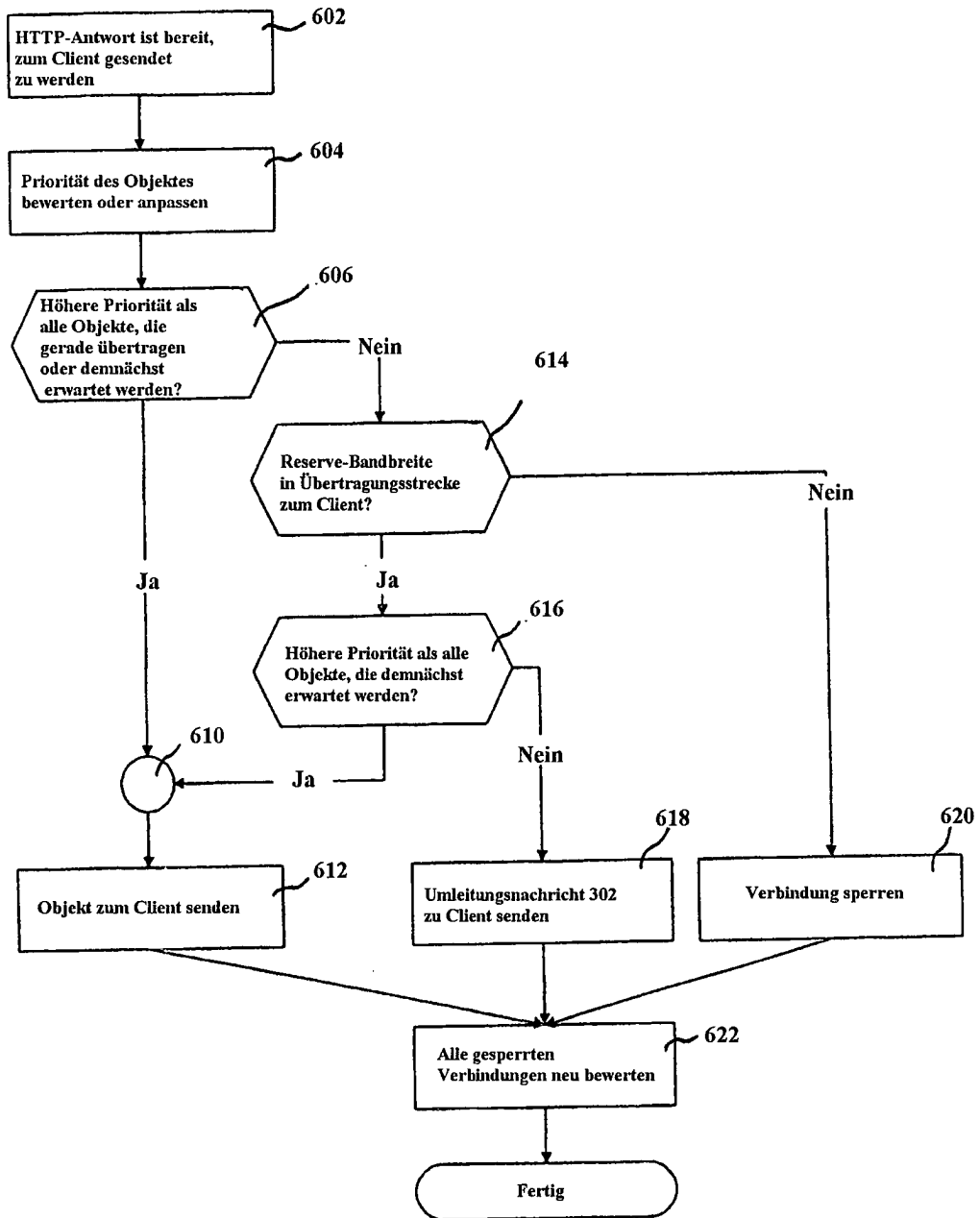


Fig. 6

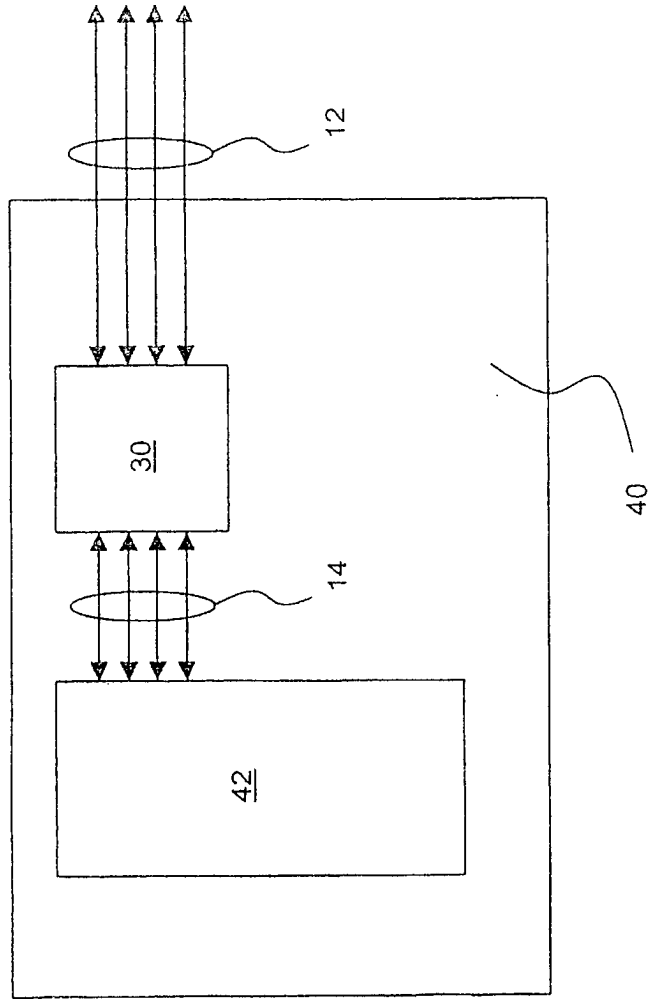


Fig. 7