



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 699 37 241 T2** 2008.01.10

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 102 442 B1**

(21) Deutsches Aktenzeichen: **699 37 241.0**

(86) PCT-Aktenzeichen: **PCT/JP99/03692**

(96) Europäisches Aktenzeichen: **99 929 754.2**

(87) PCT-Veröffentlichungs-Nr.: **WO 2000/014931**

(86) PCT-Anmeldetag: **08.07.1999**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **16.03.2000**

(97) Erstveröffentlichung durch das EPA: **23.05.2001**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **03.10.2007**

(47) Veröffentlichungstag im Patentblatt: **10.01.2008**

(51) Int Cl.<sup>8</sup>: **H04L 12/56** (2006.01)

**G06F 13/00** (2006.01)

**H04L 29/06** (2006.01)

(30) Unionspriorität:

**25431898 08.09.1998 JP**

(73) Patentinhaber:

**Fujitsu Ltd., Kawasaki, Kanagawa, JP**

(74) Vertreter:

**W. Seeger und Kollegen, 81369 München**

(84) Benannte Vertragsstaaten:

**DE, FR, GB**

(72) Erfinder:

**TAKAHASHI, Eiichi-Fujitsu Limited, Kawasaki-shi, Kanagawa 211-8588, JP; AOKI, Takeshi, Tama-shi, Tokyo 206-0035, JP; YOKOYAMA, Ken-Fujitsu Limited, Kawasaki-shi, Kanagawa 211-8588, JP; KIKUCHI, Shinji-Fujitsu Limited, Kawasaki-shi, Kanagawa 211-8588, JP**

(54) Bezeichnung: **VORRICHTUNG ZUR MESSUNG DER LAST EINES NETZWERKSERVERS UND BERÄT UND VERFAHREN ZUR ZUORDNUNG**

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**

Technisches Gebiet

**[0001]** Die vorliegende Erfindung betrifft im Allgemeinen das Aufteilen von Serverressourcen und insbesondere ein Verfahren zum Aufteilen von Diensten auf Server, um Netzwerkdienste durchzuführen.

Stand der Technik

**[0002]** Mit einer schnellen Ausbreitung des Internets und Intranets in den letzten Jahren wurden eine effiziente Verwendung und Dienststabilität eines Netzwerkdienstservers erforderlich. Ein optimales Aufteilen der Dienste auf die Server ist für die effiziente Verwendung und stabile Dienstversorgung der Server unbedingt notwendig, und, um diese optimale Aufteilung zu erreichen, ist es erforderlich, dass eine Belastung des Servers genau erkannt wird.

**[0003]** Die folgenden sind im Stand der Technik bekannte Verfahren zum Erkennen der Belastung des Servers.

(1) Agentenverfahren

**[0004]** Dies ist ein Verfahren, in welchem ein Programm zum Zählen einer Aktivitätskennzahl von Ressourcen, wie z.B. einer CPU, eines Speichers etc., auf dem Server installiert ist. Wenn ein Agent selbst die Serverbelastung erhöht und mit Außen kommuniziert, tritt eine Interferenz mit einer Belastungsmessgenauigkeit durch den Agenten auf, wie z.B. das Verbrauchen eines Bands dafür. Des Weiteren muss das Agentenprogramm auf dem Server installiert werden, und infolgedessen tritt ein Problem auf, dass keine Mehrzweckcharakteristik und hohe Instruktionkosten vorhanden sind.

(2) Belastungsmessungskommunikationsverfahren

**[0005]** Dies ist ein Verfahren zum Ausgeben eines Pingbefehls an den Server und Durchführen einer Pseudodienstkommunikation damit, und zum Erhalten einer Serverbelastung aus einer Antwortzeit. Die Kommunikation für die Messung verbraucht jedoch ein Band auf der Strecke, und der Server wird auch mit einer Last für die Antwort belastet, was zu einer Interferenz mit der Belastungsmessung führt. Des Weiteren muss der Server ein Protokoll etc. unterstützen, welches für die Messung verwendet wird, und es besteht weiterhin das Problem, dass keine Mehrzweckcharakteristik vorhanden ist.

(3) Zählverfahren von VC-Zählung, Verbindungszeit, Verbindungsfrequenz, Verbindungsfehlerrate und Antwortzeit

**[0006]** Dies ist ein Verfahren zum Erhalten der Ser-

verbelastung aus einer VC-Zählung, einer Verbindungszeit, einer Verbindungsfrequenz, einer Verbindungsfehlerrate und einer Antwortzeit in Bezug auf den Server, welche während eines Routingprozesses in einem Router zum Leiten eines Pakets von einem Client zum Server gezählt werden. Dieses Verfahren basiert jedoch auf einem Verhalten des Servers, wenn die Verbindung hergestellt wird, und daher ist ein Fehler groß. Eine große Anzahl von Verbindungen ist notwendig, um die Genauigkeit zu verbessern, so dass dieses Verfahren nicht für die Dienste geeignet ist, bei denen eine große Menge von Kommunikationen mit einer geringen Anzahl von Verbindungen durchgeführt wird. Des Weiteren ist das Routing unbedingt notwendig und infolgedessen besteht ein Problem darin, dass ein Durchsatz des Servers durch einen Durchsatz des Zählverfahrens beschränkt ist.

(4) Trefferzählungs-/Trefferratenberechnungsverfahren

**[0007]** Dies ist ein Verfahren, bei dem eine Zugriffszählung (Trefferzählung) und eine Zugriffsfrequenz (Zugriffsrate) pro Inhalt gezählt werden, wie z.B. eine Zugriffszieldatei, indem die Pakete an einen WWW-Server überprüft werden, und die Serverbelastung wird aus einem Ergebnis dieser Zählung erhalten. Dieses Verfahren benötigt einen Paketanalyseprozess pro Protokoll zum Spezifizieren der Zugriffszieldatei und ist nicht in der Lage, sich auf einen neuen Dienst zu beziehen. Außerdem muss eine Leistung des Servers bereits bekannt sein. Es gibt keine Alternative außer die Serverleistung aus Katalogwerten oder empirisch zu erhalten, um die Serverleistung im Voraus anzugeben. Die Serverleistung wird jedoch zum größten Teil durch eine Systemarchitektur und einen Betriebsmodus beeinflusst. Daher besteht ein Problem darin, dass der Katalogleistungswert basierend auf der Standardarchitektur und -Modus nicht präzise ist, und zumindest ein Problem unvermeidbar ist, wenn sie empirisch erhalten werden.

**[0008]** Wie oben erklärt sind alle Verfahren nicht in der Lage, die Serverbelastung effizient bei einer hohen Geschwindigkeit zu erkennen, ohne dem Server eine Last aufzuerlegen.

**[0009]** Des Weiteren kann die Serverbelastung nicht so genau erkannt werden, und es ist daher schwierig die Dienste zuzuteilen, welche durch den Server bereitgestellt werden.

**[0010]** Die folgenden Verfahren werden nur bezüglich des Aufteilens der Dienste vorgeschlagen.

(5) Round-Robin-DNS-Verfahren

**[0011]** Dies ist ein Verfahren, worin, in DNS- (Do-

main Name System) Diensten, ein Mapping eines Domainnamens auf eine Mehrzahl von IP-Adressen des Servers in einer Eintragungstabelle festgelegt wird, in Antwort auf eine Anforderung des Client für eine Abfrage der IP-Adresse des Servers, die jeweiligen Server entsprechend der Eintragungstabelle zyklisch zugeteilt werden (Round Robin), und die LP-Adresse des zugeteilten Servers ausgewählt wird, um dem Client zu antworten, wodurch die Dienste auf die Mehrzahl von Servern aufgeteilt werden.

**[0012]** Gemäß diesem Round-Robin-DNS-Verfahren können die Dienste jedoch nur bei einer gleichen oder einfachen Dienstaufteilungsrate aufgeteilt werden, und jeder Server muss den Dienst in Übereinstimmung mit der Aufteilungsrate durchführen, die ungeachtet seiner Leistungsfähigkeit und dynamischen Belastungszustand zugeteilt wird. Daher besteht ein Unterschied beim Belastungszustand zwischen den Servern, und das Verfahren ist im Ganzen ineffizient. Des Weiteren werden Teile einer DNS-Abfrageinformation normalerweise auf der Clientseite in den Cache kopiert, und infolgedessen besteht ein Problem darin, dass, selbst wenn sich die Rate ändert, diese Änderung nicht sofort reflektiert werden kann.

(6) Aufteilungsverfahren, das eine Hashtabelle verwendet

**[0013]** Dies ist ein Verfahren zum Zuteilen von Einträgen in einer Hashtabelle zum Verwalten der Verbindungen zu den Servern, und die Dienste werden mit einer Rate auf die Server aufgeteilt, welche der Anzahl von Einträgen entspricht, die zuzuteilen sind.

**[0014]** In diesem Verfahren, wenn der Client den Dienst anfordert, wird zunächst der Eintrag sowohl aus der Clientadresse als auch dem Dienst bestimmt. Diese Anforderung wird an den Server gesendet, dem dieser Eintrag zugeteilt ist. Dann werden die Dienste, deren Anzahl einer Kennzahl der Anzahl zugeleiteter Einträge entspricht, auf die Server aufgeteilt. Somit wird die effiziente Verwendung der Server aktualisiert durch Zuteilen der vielen Einträge an den Hochleistungsserver oder durch Neuzuteilen der Einträge an den Server mit der hohen Belastung an Server mit einer vergleichsweise geringen Belastung.

**[0015]** Gemäß diesem Aufteilungsverfahren, welches die Hashtabelle verwendet, ist jedoch eine Hashfunktion zum Erzeugen eines Hashwerts ohne Tendenz (engl. bias) notwendig, um eine Kennzahl der Anzahl von Hasheinträgen in der Dienstaufteilungsrate richtig zu reflektieren. Im Allgemeinen ist es jedoch nicht möglich, die Hashfunktion zum Erzeugen des Hashwerts ohne Tendenz herauszufinden, in Bezug auf alle Verteilungsarten von Hashschlüsseln (Clientadresse, Portnummern, etc.). Des Weiteren ist

die Genauigkeit der Aufteilungsrate proportional zu der Anzahl von Hasheinträgen, und infolgedessen wird eine Vielzahl von Einträgen benötigt, um die Genauigkeit zu verbessern, was zu einem erhöhten Verbrauch von Speicherressourcen (Puffern) führt, die einsetzbar sind, um die Verbindungen zu verwalten. Dort entsteht das Problem, dass eine große Anzahl von Zugriffen nicht bearbeitet werden kann.

(7) Aufteilungsverfahren basierend auf einem Zustand und einer Leistung des Servers

**[0016]** Dies ist ein Verfahren zum Aufteilen der Dienste, deren Anzahl der Belastung und einer Leistungskennzahl entspricht, durch Vorhersagen eines Ausmaßes der Serverbelastung oder Vorhersagen einer Leistungskennzahl zwischen den Servern, durch Zählen einer Antwortzeit, indem der Pingbefehl an den Server ausgegeben und eine Verbindungszeit und eine Verbindungsfehlerrate während einem Routingprozess durch Leiten des Pakets von dem Client gezählt wird.

**[0017]** Gemäß diesem Verfahren sind jedoch die Dienste für jeden Client gleichmäßig auf die Server aufgeteilt, ohne Rücksicht auf einen Durchsatz des Clients, einer Länge der Strecke zum Client und einer Bandbreite, so dass die Verwendungseffizienz des Servers nicht maximiert werden kann.

**[0018]** Ein Unterschied in der Leistung (insbesondere einer Geschwindigkeit) und der Belastung des Servers zeigen sich dem Client, dessen Strecke aufgrund einer kleinen oder großen Bandbreite der Strecke eine Engstelle ist, und dem Client mit einem geringen Durchsatz nicht in der QoS (Dienstgüte, engl. Quality of Service).

**[0019]** Umgekehrt nimmt der Unterschied in der Leistung und der Belastung des Servers einen großen Einfluss auf eine QoS für den Client, der an eine nahe und Hochgeschwindigkeitsleitung angeschlossen ist, oder für den Client, der einen hohen Durchsatz aufweist. Wenn dies der Fall ist, besteht ein Problem darin, dass, wenn versucht wird, alle Dienste für die Clients gleichmäßig aufzuteilen, daraus folgt, dass mehr Serverressourcen aufgeteilt werden als für die Clients benötigt werden, oder nicht anderes übrig bleibt als die ungenügenden Serverressourcen aufzuteilen.

**[0020]** US-A-5459837 (Caccavale; DEC) verwendet Sonden zwischen den Clients in einem "Broker Performance Mechanism" und führt Servergrundfunktionen aus, um eine Serverbelastung zu überwachen. US-A-5341477 von Pitkin et al. (DEC) wählt Server basierend auf einem Modellierungsverfahren aus. Der Artikel "Network Dispatcher: a connection router for scalable Internet services", Computer Networks and ISDN systems 30 (1998), Seiten 347–357, von

Hunt et al. (IBM), beschreibt ein System zum Bearbeiten großer Internetbelastungen.

**[0021]** Wie oben beschrieben sind die Probleme im Stand der Technik inhärent, sowohl in dem Erkennungsverfahren einer Serverbelastung als auch dem Serveraufteilungsverfahren.

**[0022]** Die vorliegende Erfindung, die entwickelt wurde, um die obigen Probleme zu vermeiden, zielt darauf ab, die Serverbelastung bei hoher Geschwindigkeit effizient zu erkennen, ohne dem Server eine Last aufzuerlegen, wobei die Dienste in Übereinstimmung mit dem dynamischen Belastungszustand im Server aufgeteilt werden, wobei die Dienstaufteilungsrate genau reflektiert wird, welche durch ein Festlegen und das Anpassen der Dienstaufteilung erhalten wird, und wobei die Verwendungseffizienz des Servers maximiert wird, indem die Dienste durch Vorschlagen der notwendigen Serverressourcen für jeden Client aufgeteilt werden.

**[0023]** Die vorliegende Erfindung ist in den Ansprüchen 1 und 8 als ein Verfahren, in Anspruch 10 als ein System und in Anspruch 11 als eine Vorrichtung definiert.

**[0024]** In einem Verfahren der vorliegenden Erfindung umfasst ein Erkennungsverfahren für eine Netzwerkserverbelastung einen Schritt zum Überwachen einer Kommunikation von einem Client zu einem Server, und Zählen einer Kommunikationsdatengröße pro Verbindung als eine Belastung des Servers, einen Schritt zum Erkennen einer Änderung der Kommunikationsdatengröße pro Verbindung, und Aufzeichnen eines Maximalwerts davon, und einen Schritt zum Beurteilen, wenn die Kommunikationsdatengröße pro Verbindung zu diesem Zeitpunkt in Bezug auf den Maximalwert abnimmt, dass der Server unter einer hohen Last ist.

**[0025]** Gemäß TCP (Transmission Control Protocol) etc. teilt der Server pro Verbindung die Speicherressourcen (Puffer) zum Speichern der Paketdaten gleichmäßig auf, die von den Clients übermittelt werden. Bei einem nächsten Empfang teilt der Server dem Client eine Datengröße mit, die in der Speicherressource (Puffer) gespeichert werden kann, und der Client sendet die Daten an den Server, welche die Größe besitzen, die der Server mitgeteilt hat. Dementsprechend ist der Server, wenn er eine hohe Belastung erreicht, nicht in der Lage, die Daten sofort zu verarbeiten, die von dem Client gesendet werden, und infolgedessen bleiben alle oder einige Teile der Daten in der Speicherressource (Puffer) des Servers gespeichert. Demzufolge hat der Server keine Alternative außer dem Client eine Datengröße kleiner mitzuteilen, durch die Daten, die in der Speicherressource (Puffer) gespeichert bleiben.

**[0026]** Dementsprechend wird die Datengröße pro Verbindungszeit auf der Kommunikationsleitung erkannt, wodurch der Hochbelastungszustand des Servers erkannt wird.

**[0027]** Gemäß einem zweiten Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Zählen einer Verbindungszählung und der Kommunikationsdatengröße umfassen, bis eine Zählung überwachter Kommunikationen eine Mindestzählung überwachter Kommunikationen erreicht und bis eine Zählungszeit eine Überwachungsmindestzeit erreicht, durch Verwendung der Mindestzählung überwachter Kommunikationen und der Überwachungsmindestzeit.

**[0028]** Gemäß einem dritten Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Erkennen der Kommunikationen eines Starts und Endes der Verbindung, und Ausschließen von Kommunikationsdatengrößen des Starts und Endes der Verbindung von einem Belastungserkennungsziel umfassen.

**[0029]** Die Kommunikationsdaten eines Starts und eines Endes der Verbindung, welche klein sind und nicht von der Serverbelastung abhängen, werden daher vom Zählen einer Gesamtkommunikationsdatengröße ausgeschlossen, wodurch sich ein Effekt einer Verbesserung der Genauigkeiten des Messens der Last und des Beurteilens der hohen Belastung ergibt.

**[0030]** Gemäß einem vierten Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Beibehalten von Information der Kommunikation des Starts der Verbindung umfassen, bis die Verbindung beendet oder hergestellt ist, einen Schritt zum Erkennen der Kommunikation des Starts der Verbindung für eine erneute Verbindung umfassen, der ausgeführt wird, wenn beurteilt wird, dass sich der Client nicht basierend auf der beibehaltenen Information verbindet, und einen Schritt zum Festlegen einer Rate umfassen, bei welcher die Kommunikation der erneuten Verbindung die Anzahl der Kommunikationen des Starts einer Verbindung als eine Last des Servers belegt, und, wenn diese Rate hoch ist, Beurteilen, dass der Server unter einer hohen Last ist.

**[0031]** Wenn die Serverbelastung hoch ist, kommt der Server nicht dazu, eine Antwortmitteilung auf die Verbindungsanforderung, die vom Client gemacht wird, zurückzusenden. Dementsprechend kommt der Client dazu, die Verbindungsanforderung erneut zu übermitteln. Dementsprechend kann die hohe Belastung des Servers beurteilt werden, indem die erneute Übermittlung der Verbindungsanforderung durch den Client auf der Kommunikationsleitung erkannt wird.

**[0032]** Gemäß einem fünften Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Erhalten einer Verteilung der Kommunikationsdatengrößen von den Clients umfassen, einen Schritt zum Unterscheiden zwischen extrem kleinen Teilen von Kommunikationsdaten, die nicht mit der Belastung des Servers in Beziehung stehen, von der Kommunikationsdatengrößenverteilung umfassen, und einen Schritt zum Eliminieren der extrem kleinen Teile von Kommunikationsdaten von der Beurteilung über die Belastung umfassen.

**[0033]** Es ergibt sich ein Effekt einer Verbesserung der Genauigkeiten des Messens der Belastung und des Erkennens der hohen Belastung durch Ausschließen der extrem kleinen Teile von Kommunikationsdaten vom Zählen, die nicht mit der Serverbelastung in Beziehung stehen.

**[0034]** Gemäß einem sechsten Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Erhalten mindestens einer Laufnummer aus der Kommunikation von dem Client zu dem Server umfassen, einen Schritt zum Beibehalten eines Maximalwerts der Laufnummer umfassen, bis die Verbindung beendet ist, von dem Verbindungsstart an, einen Schritt zum Vergleichen der Laufnummer der empfangenen Kommunikation mit der beibehaltenen Laufnummer umfassen, und einen Schritt zum Ausschließen dieser Kommunikation vom Zählen umfassen, wenn die Laufnummer, die aus der Kommunikation erhalten wird, kleiner ist als die beibehaltene Laufnummer.

**[0035]** Die Laufnummern sind normalerweise in der aufsteigenden Reihenfolge angeordnet, jedoch, wenn eine Kommunikation sequentiell zerstört wird oder verloren geht, aufgrund einer Überlastung auf der Kommunikationsleitung, wird die aufsteigende Reihenfolge durcheinander gebracht. Der Server kann die Daten nach den Daten nicht verarbeiten, die noch nicht angekommen sind, und infolgedessen wird eine Datengröße, die durch den Server empfangen werden kann, ungeachtet der Serverbelastung klein. Dann steigt eine Kommunikationsdatengröße des Clients entsprechend an. Es wird ein Effekt einer Verbesserung der Genauigkeiten des Messens der Serverbelastung und des Erkennens der hohen Last erzeugt, indem der Einfluss der Strecke mit dem oben beschriebenen Verfahren vermieden wird.

**[0036]** Gemäß einem siebten Merkmal kann das Erkennungsverfahren für eine Netzwerkserverbelastung gemäß dem ersten Aspekt des Weiteren einen Schritt zum Zählen, wenn die Laufnummer, die aus der Kommunikation erhalten wird, kleiner ist als die beibehaltene Laufnummer, der Kommunikationsdaten umfassen, nachdem ein Gewichtungsprozess an

ihnen ausgeführt wurde, oder Vorhersagen einer Kommunikationsdatengröße, wenn es auf einer Strecke der zwei Laufnummern kein Problem gibt, und Zählen der vorhergesagten Datengröße zum Erkennen der Belastung.

**[0037]** Gemäß einem zweiten Verfahrenstyp der vorliegenden Erfindung umfasst ein Erkennungsverfahren für eine Netzwerkserverbelastung einen Schritt zum Überwachen einer Kommunikation von einem Server zu einem Client, und Zählen einer empfangbaren Datengröße und einer Verbindungszählung, welche der Server dem Client mitteilt, einen Schritt zum Erhalten der empfangbaren Datengröße pro Verbindung als eine Serverbelastung, einen Schritt zum Speichern eines Maximalwerts der empfangbaren Datengröße pro Verbindung, und Beurteilen, wenn die empfangbare Datengröße pro Verbindung klein wird in Bezug auf den Maximalwert, dass der Server unter hoher Belastung ist.

**[0038]** In einer Vorrichtung der vorliegenden Erfindung umfasst ein Erkennungssystem für eine Serverbelastung zum Überwachen einer Kommunikation von einem Client zu einem Server und Erkennen eines Belastungszustands des Servers eine Datengrößenberechnungsmodul zum Berechnen einer Größe von Kommunikationsdaten pro Verbindung, eine Speichereinheit zum Erkennen einer Änderung der Kommunikationsdatengröße pro Verbindung und Speichern eines Maximalwerts, und ein Belastungserkennungsmodul zum Erkennen einer hohen Belastung des Servers, wenn die Kommunikationsdatengröße pro Verbindung zu diesem Zeitpunkt in Bezug auf den Maximalwert gleich oder kleiner ist als ein festgelegter Wert.

**[0039]** In einem System, das nicht Gegenstand der vorliegenden Erfindung ist, umfasst ein Netzwerkserveraufteilungssystem zum Übertragen von Daten von einem Client an eine Mehrzahl von Servern über ein Netzwerk eine Routingeinheit zum Übertragen der Daten, die von dem Client an irgendeinen der Server übermittelt werden, durch Ändern eines Ziels der Daten, ein Verbindungsverwaltungsmodul, um ein Mapping zwischen den Daten und dem Server beizubehalten und um dem Routingmittel das Ziel anzuzeigen, und ein Serveraufteilungsmodul zum Erhalten von Durchsätzen des Servers, des Clients und einer Strecke, indem sie gezählt werden, zum Bestimmen einer Korrespondenz zwischen den Daten und dem Server durch Verwendung einer Funktion gemäß einer Dienstverteilungsrates basierend auf dem Durchsatz, und zum Übertragen dieser Korrespondenz an das Verbindungsverwaltungsmodul.

**[0040]** Die Dienste werden basierend auf Algorithmen aufgeteilt, die durch Messen der Leistung und Belastung des Servers und der Leistung und Belastung auf der Seite des Clients erhalten werden, und

es ist daher machbar, automatisch einer Änderung eines dynamischen Belastungszustands des Servers zu entsprechen. Des Weiteren gibt es den Effekt, bei dem die Server, die notwendig sind, um eine QoS aus der Sicht des Clients beizubehalten, aufgeteilt werden können, und eine Verwendungseffizienz des Servers maximiert werden kann. Ein weiterer Effekt ist, dass die Serveraufteilung durch die Verwendung einer Funktion bestimmt wird, und infolgedessen eine Dienstaufteilungsrate beim Aufteilen genau reflektiert werden kann. Außerdem kann ein ausreichender Effekt nur durch das einzelne Verbindungsverwaltungsmodul erhalten werden.

**[0041]** Gemäß einem weiteren Merkmal, in dem Netzwerkserveraufteilungssystem gemäß dem zehnten Aspekt, legt die Serveraufteilungseinheit eine modifizierte Wahrscheinlichkeitsverteilung als die Verteilungsrate fest, die durch Modifizieren einer Wahrscheinlichkeitsverteilung entsprechend dem Durchsatz des Servers erhalten wird, so dass die Wahrscheinlichkeitsverteilung besser an eine gleichmäßige Verteilung angenähert wird, wenn die Durchsätze des Clients und der Strecke niedriger werden.

**[0042]** Eine proportionale Kennzahl, in welchem Maße die Durchsätze des Clients und der Strecke und der Durchsatz des Servers einen Einfluss auf die QoS haben, wird in der Dienstaufteilungsrate reflektiert, und daher ist ein Effekt so, dass der Server, der den hohen Durchsatz aufweist, bevorzugt dem Client zugeweiht ist, in welchem der Serverdurchsatz einen großen Einfluss auf die QoS hat.

**[0043]** Gemäß einem weiteren Merkmal, in dem Netzwerkserveraufteilungssystem gemäß dem zehnten Aspekt, erhält die Serveraufteilungseinheit eine Verteilung der Durchsätze des Clients und der Strecke in Bezug auf den Client, der nun in Betrieb ist, erhält auch eine modifizierte Wahrscheinlichkeitsverteilung durch Ausführen solcher Modifikationen, um die Wahrscheinlichkeitsverteilung entsprechend dem Durchsatz des Servers besser an die gleichmäßige Verteilung anzunähern, wenn die Durchsätze eines neu angeschlossenen Clients und der Strecke für die Verteilung niedriger werden, und um umgekehrt den Durchsatz des Servers herausragender zu machen, wenn die Durchsätze eines neu angeschlossenen Clients und der Strecke höher werden, und legt diese modifizierte Wahrscheinlichkeitsverteilung als eine Verteilungsrate fest.

**[0044]** Die Dienstaufteilungsrate wird in Bezug auf eine Verteilung der Durchsätze des Clients im Betrieb und der Strecke angepasst, und infolgedessen ist ein Effekt, dass es möglich ist, automatisch auf einen Fall zu antworten, in dem sich eine Kennzahl der Clients von einem entfernten Platz und einem nahen Platz verändert.

**[0045]** Gemäß einem weiteren Merkmal, in dem Netzwerkserveraufteilungssystem gemäß dem zehnten Aspekt, wird eine Mehrzahl von Serveraufteilungseinheiten bereitgestellt und jede pro Client und Dienst ausgewählt.

**[0046]** Eine Skala des Verbindungsverwaltungsmoduls hängt nicht von der Dienstaufteilung ab, so dass es einen Effekt gibt, worin die Verwendungseffizienz der Speicherressource (Puffer) verbessert wird.

**[0047]** Es gibt dort auch solch einen Effekt, dass eine Servergruppe eines Aufteilungsziels separat pro Service und pro Client verwendet werden kann, und eine Dienstverteilungsmethode wird umgeschaltet, wodurch ein einzelnes System in der Lage ist, die Dienstaufteilung in einer Vielzahl von Formen auszuführen.

#### Kurze Beschreibung der Zeichnungen

**[0048]** [Fig. 1](#) ist ein Diagramm, welches eine Topologie eines Belastungserkennungssystems in einem Ausführungsbeispiel der vorliegenden Erfindung zeigt;

**[0049]** [Fig. 2](#) ist ein Graph, welcher ein Verhältnis zwischen einer Zeit und einer Datengröße zur Hilfe beim Beurteilen einer hohen Belastung eines Servers in dem Ausführungsbeispiel zeigt;

**[0050]** [Fig. 3](#) ist ein Flussdiagramm (1), welches ein Paketüberwachungsverfahren in einem Ausführungsbeispiel 1 zeigt;

**[0051]** [Fig. 4](#) ist ein Flussdiagramm (2), welches das Paketüberwachungsverfahren in dem Ausführungsbeispiel 1 zeigt;

**[0052]** [Fig. 5](#) ist ein erklärendes Diagramm, welches eine Verbindungsanforderung von einem Client an einen Server und einen Antwortprozess entsprechend einem Zustand eines Puffers zeigt;

**[0053]** [Fig. 6](#) ist ein erklärendes Diagramm, welches eine erneute Übertragung der Verbindungsanforderung von dem Client an den Server zeigt;

**[0054]** [Fig. 7](#) ist ein erklärendes Diagramm, welches ein Beispiel einer Unterscheidung zeigt, ob die Daten als Zieldaten abhängig von einer Datengröße bei einer Belastungserkennung festgelegt werden;

**[0055]** [Fig. 8](#) ist ein erklärendes Diagramm, welches einen Prozess des Servers basierend auf einer Laufnummer zeigt;

**[0056]** [Fig. 9](#) ist ein erklärendes Diagramm, welches zeigt, wie eine Kommunikation von dem Server an den Client überwacht wird;

[0057] **Fig. 10** ist ein Blockdiagramm, welches eine Architektur eines Serveraufteilungssystems in dem Ausführungsbeispiel zeigt;

[0058] **Fig. 11** ist ein Graph zum Erklären einer Wahrscheinlichkeitsverteilung PsD einer Serveraufteilung;

[0059] **Fig. 12** ist ein erklärendes Diagramm (1), welches eine Modifikationsfunktion zeigt;

[0060] **Fig. 13** ist ein Diagramm, welches ein Beispiel einer Tabelle zeigt, die durch ein Serverauswahlmodul in dem Ausführungsbeispiel erzeugt wird;

[0061] **Fig. 14** ist ein Graph, der ein Beispiel einer Verteilung von Durchsatzwerten auf der Clientseite in der Vergangenheit zeigt; und

[0062] **Fig. 15** ist ein erklärendes Diagramm (2), welches die Modifikationsfunktion zeigt.

Bester Modus zum Ausführung der Erfindung

[Ausführungsbeispiel 1]

[0063] **Fig. 1** zeigt eine Architektur von Funktionen eines Erkennungssystems **4** für eine Serverbelastung in einem Ausführungsbeispiel 1. Wie in der gleichen Figur gezeigt, ist das Erkennungssystem **4** für eine Serverbelastung an eine Kommunikationsleitung **3** angeschlossen, die mit einem Client **1** und einem Server **2** verbunden ist, und kann, um genau zu sein, in einen Router etc. implementiert werden.

[0064] Dieses Erkennungssystem **4** für eine Serverbelastung umfasst, wie in der gleichen Figur gezeigt, ein Erfassungsmodul **5** für Kommunikationsdaten zum Erfassen von Paketdaten (TCP-Paket: Transmission Control Protocol Paket), welche über die Kommunikationsleitung **3** übermittelt werden. Ein Erkennungsmodul **6** für eine Verbindungszählung, ein Paketzählermodul **8** und ein Berechnungsmodul **7** für eine Paketgröße werden an dieses Erfassungsmodul **5** für Kommunikationsdaten angeschlossen.

[0065] Das Erkennungsmodul **6** für eine Verbindungszählung besitzt eine Funktion zum Erkennen einer Verbindungszählung C pro Einheit Zeit von den TCP-Paketen, welche durch das Erfassungsmodul **5** für Kommunikationsdaten erfasst werden. Dieses Erkennungsmodul **6** für eine Verbindungszählung addiert 1, wenn sie ein SYN-Paket erkennt, welches ein Kopfpaket repräsentiert, und subtrahiert 1, wenn sie ein FIN-Paket erkennt, welches das letzte Paket repräsentiert. Die Anzahl von Clients, die gegenwärtig an den relevanten Server angeschlossen ist, kann dadurch erkannt werden.

[0066] Das Paketzählermodul **8** besitzt eine Funkti-

on zum Zählen einer Paketzählung N der TCP-Pakete, die pro Einheit Zeit durch das Erfassungsmodul **5** für Kommunikationsdaten erfasst werden. Das Berechnungsmodul **7** für eine Paketgröße besitzt eine Funktion zum Berechnen einer Gesamtgröße S der TCP-Pakete, die pro Einheit Zeit durch das Erfassungsmodul **5** für Kommunikationsdaten erfasst werden.

[0067] Berechnungs-/Zählungsdaten dieser Module werden an ein Belastungserkennungsmodul **10** übertragen, worin eine Belastung basierend auf einem vorbestimmten arithmetischen Prozess beurteilt wird, der später erklärt werden wird.

[0068] Die Paketgesamtgröße S, welche durch das Berechnungsmodul **7** für eine Paketgröße berechnet wird, wird sequenziell erhöht durch eine Paketgröße des Pakets, das jede Zeit erreicht wird, wobei die Größe S auf 0 festgelegt wird, wenn die Zählung begonnen wird. Es sollte beachtet werden, dass eine Größe jedes der SYN- und FIN-Pakete kleiner ist als das Datenpaket, und ihr Einfluss auf die Serverbelastung klein genug ist, um ignoriert zu werden.

[0069] Die Paketzählung N der Pakete, welche durch das Paketzählermodul **8** gezählt wird, addiert 1, jedes mal wenn das Paket erreicht wird, wobei die Paketzählung auf 0 gesetzt wird, wenn die Zählung begonnen wird. Es sollte beachtet werden, dass die Zählung der SYN- und FIN-Pakete aus dem oben erklärten Grund ignoriert werden kann.

[0070] Das Paketzählermodul **8** fährt fort zu zählen, bis N einen bestimmten Wert Nmin überschreitet, jedoch, wenn eine Zählungszeit vom Start an kürzer ist als eine vorgegebene Zeit Tmin, selbst wenn über Nmin, fährt die Zählung weiter fort, bis die Zeit Tmin abläuft.

[0071] Hier werden Nmin und Tmin vorher in dem Paketzählermodul **8** festgelegt. Somit werden Nmin und Tmin in Kombination verwendet, und dies macht es möglich, einen Berechnungsfehler zu reduzieren, der aufgrund einer geringen Probezählung auftrat, welche die Belastung erkennt, um einen Überlauf zu vermeiden, der aufgrund der Probezählung auftrat, die zu viel war, und um eine Belastungserkennungsgenauigkeit zu verbessern.

[0072] Das Belastungserkennungsmodul **10** erkennt die Belastung durch Ausführen der folgenden arithmetischen Prozesse.

[0073] Zunächst erhält das Belastungserkennungsmodul **10**, wenn es die Verbindungszählung C von dem Erkennungsmodul **6** für eine Verbindungszählung und die Paketgröße S von dem Berechnungsmodul **7** für eine Paketgröße empfängt, einen Indexwert L einer Serverbelastung basierend auf der For-

mel, die folgt.

**[0074]** Es sollte beachtet werden, dass  $T$  hier eine Messzeit ist, welche durch einen Timer **11** gemessen wird. Wenn die Paketzählung  $N$ , welche die Proben anzeigt,  $N_{min}$  überschreitet wenn  $T_{min}$  abläuft, wird die Messzeit  $T$  beispielsweise auf  $T = T_{min}$  festgelegt.

$$L = (S/C)/T$$

wo  $L$  eine Datentransferanzahl pro Verbindung für die Einheit Zeit ist. Eine Belastung des Servers **2** kann durch die Verwendung von  $L$  erkannt werden.

**[0075]** Des Weiteren aktualisiert das Lasterkennungsmodul **10** einen Voraussagewert  $L_{max}$  für ein Durchsatzlimit. Hier ist ein Anfangswert von  $L_{max}$  0, und, wenn  $L$   $L_{max}$  überschreitet, wird ein Wert von  $L_{max}$  als  $L$  festgelegt. Hier kann, wenn die folgende Beziehung zwischen  $L$  und  $L_{max}$  hergestellt wird, beurteilt werden, dass der Server unter einer hohen Last ist.

$$L < \alpha L_{max} \quad (1)$$

mit  $0 < \alpha \leq 1$ , und  $\alpha$  ist eine vorgegebene Konstante.

**[0076]** [Fig. 3](#) ist ein Flussdiagramm, welches zeigt, wie das Belastungserkennungsmodul **10**, das oben beschrieben wird, die Belastung erkennt.

**[0077]** Zunächst, bei einem Start einer Zählung, werden die Paketzählung  $N$  und der Indexwert  $L$  einer Serverbelastung zurückgesetzt, und der Timer **11** wird gestartet (Schritt **301**). Dann, wenn ein Empfang des Pakets über das Erfassungsmodul **5** für Kommunikationsdaten gestartet wird (**302**), wird beurteilt, ob das Paket ein Verbindungsstartpaket SYN ist oder nicht (**303**) beziehungsweise ob das Paket ein Verbindungsendepaket FIN ist oder nicht (**305**). Hier wird, wenn es das Verbindungsstartpaket SYN ist, eine Variable  $V$  um 1 erhöht (**304**). Des Weiteren wird, wenn es das Verbindungsendepaket FIN ist, die Variable  $V$  um 1 erniedrigt (**306**).

**[0078]** Als nächstes wird die Paketzählung  $N$  jedes Mal um 1 erhöht, wenn ein neues Paket empfangen wird, und das Belastungserkennungsmodul **10** berechnet den Indexwert  $L$  einer Serverbelastung (**307**). Diese Berechnung wird basierend auf der Berechnungsformel durchgeführt, die vorher erklärt wurde. Dann, wenn der Indexwert  $L$  einer Serverbelastung  $\alpha L_{max}$  in der Formel (1), die oben angegeben wird, überschreitet, wird beurteilt, dass der Server in einem Zustand hoher Belastung ist.

**[0079]** Ein Ende dieser Beurteilung hoher Belastung wird ausgelöst, wenn der Timerwert gleich oder grö-

ßer wird als die vorgegebene Zeit  $T_{min}$  und wenn die Paketzählung  $N$  gleich oder größer wird als der vorgegebene Wert  $N_{min}$  (**308**).

**[0080]** Hier teilt der Server **2** gemäß TCP Speicherressourcen (Puffer) zum Speichern der Paketdaten auf, die von dem Client **1** gleichmäßig pro Verbindung übermittelt werden. Der Server **2** teilt dem Client **1** eine Datengröße mit, die in der Speicherressource (Puffer) gespeichert werden kann, wenn das nächste Mal empfangen wird. Der Client **1** sendet die Datengröße, welche ihm der Server **2** mitgeteilt hat, über die Kommunikationsleitung **3** an den Server **2**.

**[0081]** Dementsprechend ist der Server **2**, wenn er die hohe Belastung erlangt, nicht mehr in der Lage, die Daten sofort zu verarbeiten, die von dem Client **1** gesendet werden, und infolgedessen bleiben alle oder einige Teile der Daten in der Speicherressource (Puffer) des Servers **2** gespeichert. Demzufolge hat der Server **2** keine Alternative, außer dem Client eine Datengröße mitzuteilen, die kleiner ist, durch die Daten, die in der Speicherressource (Puffer) gespeichert bleiben.

**[0082]** Hier ist TCP ein Protokoll, das zum Senden und Empfangen der Daten konzipiert ist, deren Größe so groß wie möglich ist, und daher ist die Datengröße der Daten, die von dem Client **1** an den Server übermittelt werden, maximal in einem Zustand, bevor der Server **2** die hohe Belastung erlangt. Danach, wenn die Last am Server **2** ansteigt, nimmt die Datengröße der Daten auch ab, die auf der Kommunikationsleitung **3** übermittelt werden. In diesem Ausführungsbeispiel, wie in [Fig. 2](#) gezeigt, wird der Zustand hoher Belastung des Servers mit einer Gewichtung erkannt, dass die Datengröße abnimmt.

**[0083]** In diesem Ausführungsbeispiel werden die Daten in der Datenbank **12** gespeichert, wobei die Datengröße der Daten, die auf der Kommunikationsleitung **3** in dem Zustand übermittelt werden, bevor der Server **2** die hohe Belastung erlangt, den Maximalwert  $L_{max}$  besitzt. Dann, wie in der Formel (1) gezeigt, wird der Wert (Schwellenwert), der durch Multiplizieren von  $L_{max}$  mit der Konstante  $\alpha$  erhalten wird, mit  $L$  verglichen. Wenn  $L$  gleich oder kleiner ist als der Schwellenwert, wird beurteilt, dass der Server **2** im Zustand hoher Belastung ist.

**[0084]** Somit ist es in diesem Ausführungsbeispiel möglich, eine falsche Beurteilung mit einer Reduzierung der Gesamtdatengröße zu verhindern, aufgrund der Abnahme der Verbindungszählung selbst, indem die Datengröße pro Verbindung überprüft wird, und die falsche Erkennung der hohen Belastung kann aufgrund von  $L$ -Schwankungen, die sich durch eine Störung ergeben, durch Verwendung der Konstante  $\alpha$  verhindert werden.

[0085] Es sollte beachtet werden, dass [Fig. 4](#) ein Flussdiagramm ist, welches im Wesentlichen das gleiche wie das Flussdiagramm in [Fig. 3](#) ist, und Schritte zum Beurteilen der hohen Belastung zeigt, ohne das Kommunikationsstartpaket SYN und das Kommunikationsendepaket FIN zu betrachten.

[Ausführungsbeispiel 2]

[0086] Ein Ausführungsbeispiel 2 veranschaulicht ein Erkennungsverfahren für eine hohe Belastung, wobei ein Neuübermittlungsprozess von dem Client 1 an den Server 2 eingesetzt wird.

[0087] Eine Systemarchitektur, die im Ausführungsbeispiel 2 verwendet wird, ist im Wesentlichen die gleiche wie jene, die in [Fig. 1](#) in dem Ausführungsbeispiel 1 gezeigt ist, und infolgedessen wird ihre Erläuterung ausgelassen.

[0088] Im Ausführungsbeispiel 2 wird die Information des einzelnen Kommunikationsstartpakets SYN in der Datenbank 12 aufgezeichnet (siehe [Fig. 6\(a\)~6\(c\)](#)). Dann wird die Information des einzelnen Kommunikationsstartpakets SYN durch ein Tupel einer Clientadresse (IP), einer Clientportnummer (sp) und einer Serverportnummer (dp) identifiziert.

[0089] Gemäß TCP, wenn der Server 2 das Kommunikationsstartpaket SYN vom Client 1 empfängt, wird ein SYN-Empfangsbestätigungspaket zurück an den Client 1 gesendet. Hier wird, wenn der Client 1 nicht in der Lage ist, das SYN-Empfangsbestätigungspaket von dem Server 2 zu empfangen, selbst wenn eine festgelegte Zeitdauer abläuft, das Kommunikationsstartpaket SYN erneut an den Server 2 übermittelt.

[0090] [Fig. 5](#) zeigt dieses Konzept. Bezug nehmend auf [Fig. 5\(a\)](#), sendet der Client 1a zuerst eine Verbindungsanfrage (das Kommunikationsstartpaket SYN) an den Server 2. Andererseits sendet auch der andere Client 1b die Verbindungsanfrage (das Kommunikationsstartpaket SYN) an den Server 2. Hier, wenn ein Puffer 51 des Servers 2 eine Erlaubnis besitzt, das heißt, wenn er in einem Zustand niedriger Belastung ist, sendet der Server 2 eine Antwortmitteilung (Empfangsbestätigungspaket) an die Clients 1a und 1b, wohingegen wenn der Puffer 51 des Servers 2 keine Erlaubnis besitzt, wie in [Fig. 5\(b\)](#) gezeigt, nicht auf die Verbindungsanforderung (das Kommunikationsstartpaket SYN) von dem Client 1 geantwortet werden kann. Dann sendet der Client 1, wie in [Fig. 5\(c\)](#) gezeigt, wenn er nicht in der Lage ist, die Antwortmitteilung (Empfangsbestätigungspaket) von dem Server 2 innerhalb der festgelegten Zeitdauer zu empfangen, die Verbindungsanforderung erneut an den Server 2.

[0091] In dem Ausführungsbeispiel 2 zählt das Er-

kennungsmodul 6 für eine Verbindungszählung eine Anzahl Cs der Startpakete SYN, und die Anzahl von Neuübermittlungsprozessen des Startpakets SYN von dem Client 1 wird erkannt, wodurch eine Rate Rs der Anzahl von Neuübermittlungsprozessen des Startpakets SYN berechnet wird. Diese Rate Rs wird als ein Indexwert Crs einer Serverbelastung festgelegt.

[0092] Hier kann das Startpaket SYN beurteilt werden, um erneut übermittelt zu werden, wenn die SYN-Information, die aus dem Startpaket SYN extrahiert wird, bereits in der Datenbank 12 aufgezeichnet wurde. [Fig. 6](#) zeigt wie dies ist. Bezug nehmend auf [Fig. 6\(a\)](#), werden SYN1 (IP1, sp1, dp1), SYN2 (IP2, sp2, dp2) und SYN3 (IP3, sp3, dp3) als die SYN-Information in der Datenbank des Belastungserkennungssystems 4 aufgezeichnet. Zu diesem Zeitpunkt sendet der Client 1 die Verbindungsanforderung (Startpaket SYN4) über die Kommunikationsleitung 3 an den Server. Das Belastungserkennungssystem 4, wenn diese Verbindungsanforderung eine Verbindungsanforderung ist, die nicht in der Selbstdatenbank 12 gespeichert ist, das heißt, wenn dies die erste Verbindungsanforderung ist, speichert diese Verbindungsanforderung (SYN4: IP4, sp4, dp4) in der gleichen Datenbank 12 ([Fig. 6\(b\)](#)).

[0093] Dann, wenn der Server 2 dem Client 1 nicht irgendeine Antwort auf diese Verbindungsanforderung (SYN4) mitteilt, sendet der Client 1 die gleiche Verbindungsanforderung (SYN4) erneut an den Server 2. Das Erfassungsmodul 5 für Kommunikationsdaten erfasst diese Verbindungsanforderung (SYN4) und das Belastungserkennungsmodul 10 durchsucht die Datenbank 12, wobei das Belastungserkennungssystem 4 weiß, dass die Verbindungsanforderung bereits in dem System 4 selbst gespeichert wurde, und beurteilt demzufolge, dass die gleiche Verbindungsanforderung (SYN4) eine Neuverbindungsanforderung ist.

[0094] Ein spezielles Zählverfahren in dem Belastungserkennungsmodul 10 basiert auf dem Zählverfahren der Verbindungszählung C und dem Erkennungsverfahren der Paketgröße S, die in Ausführungsbeispiel 1 erklärt wurden.

[0095] Hier, wenn die folgende Formel (2) in Bezug auf die erhaltene Rate Rs der Anzahl von Neuübermittlungsprozessen des Startpakets SYN eingeführt wird, das heißt Crs, wird der Server 1 beurteilt, eine hohe Belastung zu besitzen.

$$Crs > \beta \quad (2)$$

mit  $0 < \beta < 1$ ,  $\beta$  ist eine vorgegebene Konstante.

[0096] Der Server 2 teilt die Puffer 51 zum Speichern der Daten von dem Client 1 pro Verbindung

auf. Wenn die Puffer **51**, die aufzuteilen sind, aufgebraucht sind, wird die Antwortmitteilung (SYN-Empfangsbestätigungspaket) nicht zurück an den Client **1** gesendet, ohne die Verbindung herzustellen. Daher folgt, dass der Client **1** eine erhöhte Rate der Neuübermittlungsprozesse des Startpakets SYN besitzt. Dementsprechend kann die hohe Belastung des Servers aus der Formel (2) erkannt werden. **Fig. 6(d)** ist ein Graph, der eine Beziehung zwischen der Neuübermittlungskennzahl (die Anzahl von Neuübermittlungsprozessen/die Anzahl von Kommunikationen), die oben beschrieben wird, und der Serverbelastung zeigt.

**[0097]** Es sollte beachtet werden, dass die Konstante  $\beta$  in der Formel (2), die oben angegeben wird, festgelegt wird, um die falsche Erkennung aufgrund der Störung und dem vorübergehenden Zustand hoher Belastung zu verhindern. Der vorübergehende Zustand hoher Belastung besitzt eine geringe Wahrscheinlichkeit aufzutreten, dauert jedoch nicht lange an, und kann daher ignoriert werden.

[Ausführungsbeispiel 3]

**[0098]** Ein Ausführungsbeispiel 3 veranschaulicht eine Technologie, um eine Unterscheidung eines Zählungsziels zu machen, abhängig von der Kommunikationsdatengröße, wenn die Belastung erkannt wird. Es sollte beachtet werden, dass die Systemarchitektur in dem Ausführungsbeispiel 3 auch im Wesentlichen die gleiche ist wie jene, die in **Fig. 1** gezeigt ist, und wird daher mit Bezugnahme auf **Fig. 1** erklärt.

**[0099]** In dem Ausführungsbeispiel 3 erkennt das Belastungserkennungsmodul **10**, wenn die folgende Beziehung zwischen einer Paketgröße  $S_i$  des Pakets von dem Client **1** und  $D_s$  hergestellt wird, die Belastung, ohne  $S_i$  zu der Gesamtpaketgröße  $L$  hinzuzufügen.

$$S_i < \gamma D_s \quad (3)$$

mit  $0 < \gamma < 1$ ,  $D_s = f(S_1, S_2, \dots, S_{i-1})$ ,  $\gamma$  ist eine vorgegebene Konstante,  $D_s$  ist eine Funktion zum Erhalten eines Verteilungsindex der gezählten Paketgröße und kann als ein Mittelwert festgelegt werden. Des Weiteren, wenn eine Mehrzahl von Werten als ein Ergebniswert von  $D_s$  angegeben ist, können diese Werte als ein einzelner Wert festgelegt werden, durch eine gewichtete Addition und Auswahl.

**[0100]** Gemäß TCP legt der Client **1** nach der Verbindung die Übertragungsdatengrößen kleiner fest als die Datengröße, die der Server **2** mitgeteilt hat, startet dann die Übertragung und erhöht nach und nach die Datengröße bis zu der mitgeteilten Datengröße. Daher ist die Paketgröße, die von dem Client **1** angegeben wird, kurz nachdem die Verbindung ge-

startet wird, klein, ungeachtet der Last am Server **2**.

**[0101]** Dementsprechend, wenn eine große Anzahl der Clients **1** kurz nach dem Start der Verbindung vorhanden ist, wird die Datentransferanzahl  $L$  in der Formel (1) klein vorhergesagt, aufgrund einer Vielzahl kleiner Teile von Übertragungsdaten, und es kann eine Möglichkeit geben, bei der sich Genauigkeiten der Messung der Belastung und eine Erkennung der hohen Belastung verschlechtern.

**[0102]** **Fig. 7** zeigt konzeptionell wie dies ist. Bezugnehmend auf **Fig. 7(a)**, übermittelt der Client **1a** Paketdaten  $A$  mit einer vergleichsweise großen Größe an den Server **2**. Der Client **1b** übermittelt jedoch Paketdaten  $B$  mit einer vergleichsweise kleinen Größe, wie z.B. einen Befehl und ein Antwortsignal, da dies kurz nach dem Start der Kommunikation ist. Wenn solch eine kleine Größe von Paketdaten ignoriert werden kann, wenn die Last am Server erkannt wird, kann kein Problem auftreten.

**[0103]** In solch einem Fall, in Übereinstimmung mit dem Ausführungsbeispiel 3, durch Verwendung der Formel (3), wird das Paket vom Client **1** kurz nach dem Start der Verbindung erkannt und vom Zählungsziel ausgeschlossen, wodurch die Genauigkeiten des Messens der Belastung und des Erkennens der hohen Belastung verbessert werden.

**[0104]** Wenn der Server eine hohe Belastung erreicht, nimmt die Datengröße der Daten aller verbundenen Clients ab, jedoch ist die Reduzierung in den Puffern **51** zum Speichern der Daten vergleichsweise moderat, und infolgedessen ist auch die Abnahme von  $L$  moderat. Des Weiteren ist sie gering bezüglich der Wahrscheinlichkeit, dass alle Clients die neuen Verbindungen gleichzeitig starten, so dass die Formel (3) ausreichend ist.

**[0105]** Ein unterer Grenzwert  $D_{smin}$  von  $D_s$  wird als eine angewendete Bedingung in der Formel (3) festgelegt, um die Genauigkeit zu erhöhen, und, wenn  $D_s$  gleich oder kleiner als  $D_{smin}$  ist, wird die Formel (3) nicht angewendet.  $S_i$  kann nämlich zu  $L$  hinzu addiert werden.

[Ausführungsbeispiel 4]

**[0106]** Ein Ausführungsbeispiel 4 veranschaulicht eine Technologie zum Verhindern einer falschen Erkennung der hohen Belastung des Servers, aufgrund eines Paketwiderrufs, welcher durch eine Überlastung etc. auf der Kommunikationsleitung auftritt, wenn die Belastung in der Diskussion des Ausführungsbeispiels 1 erkannt wird.

**[0107]** Eine Systemarchitektur in dem Ausführungsbeispiel 4 ist die gleiche wie jene in **Fig. 1**. Hier werden von dem Paket von dem Client **1** zu dem Server

**2** ein Tupel (Paketkennung) einer Clientadresse (IP), einer Clientportnummer (sp) und einer Serverportnummer (dp) und eine Laufnummer von dem Start der Verbindung bis zu dem Ende davon in der Datenbank **12** gespeichert. Zu diesem Zeitpunkt soll die gespeicherte Laufnummer ein Maximalwert sein (der letzte Wert zu diesem Zeitpunkt).

**[0108]** Das Belastungserkennungssystem **4** erhält, wenn das Paket empfangen wird, das von dem Client **1** an den Server **2** übermittelt wird, die Paketkennung und die Laufnummer  $P_i$  von diesem Paket, und vergleicht sie mit einer Laufnummer  $P_j$  der gleichen Paketkennung, die in der Datenbank **12** gespeichert ist.

**[0109]** Hier kann es, wenn eine Beziehung, wie z.B.  $P_i < P_j$  hergestellt wird, basierend auf einer Beurteilung des Belastungserkennungsmoduls **10**, bekannt sein, dass ein Vorbeiziehen des Pakets auf der Kommunikationsleitung **3** auftritt, oder ein Paket erneut übermittelt wird, aufgrund des Verschwindens des Pakets auf der Strecke.

**[0110]** Auf jeden Fall folgt daraus, dass die Daten, welche durch den Server **2** in diesem Zustand empfangen werden, einen Verlust auf halber Strecke besitzen, und der Server **2** nicht in der Lage ist, die Daten nach dem verlorenen Teil zu verarbeiten. Es folgt, dass die Daten nach dem verlorenen Teil in dem Puffer **51** gespeichert bleiben. Die Datengröße, die durch den Server **2** empfangen werden kann, wird dadurch verringert, jedoch ist der Grund dafür nicht die Serverbelastung sondern eine Überlastung etc. auf der Strecke zwischen dem Client und dem Server. [Fig. 8](#) zeigt konzeptionell wie dies ist. In [Fig. 8](#) werden Teile von Paketdaten [1~3] von dem Client **1** an den Server **2** übertragen, und nur die Paketdaten [2] gehen durch einen Faktor wie z.B. die Streckenüberlastung etc. verloren. Der Client **2** speichert die Paketdaten [1, 3], die im Puffer **51** empfangen wurden. Hier wird dem Client **1** eine Antwort mitgeteilt (eine Anforderung zum erneuten Übermitteln der Paketdaten [2]), jedoch werden die Paketdaten [2] nicht im Puffer des Servers **2** selbst empfangen, und infolgedessen tritt ein Zustand auf, in dem er nicht in der Lage ist, die Daten nach den Paketdaten [3] zu verarbeiten, die ihn bereits erreicht haben.

**[0111]** Der Client **1**, wenn er die Antwortmitteilung über die Paketdaten [2] doppelt empfängt, übermittelt erneut die Paketdaten [2]. Somit sind die Paketdaten [2~5] alle angekommen, wodurch der Server **2** einen Zustand erreicht, in dem er in der Lage ist, diese Teile von Paketdaten zu verarbeiten, die er empfangen hat, jedoch nicht in der Lage ist, sofort auf die Verarbeitung umzuschalten. Infolgedessen ist eine freie Größe des Puffers, die dem Client **1** mitgeteilt wird,  $n$ , was bei weitem kleiner ist als eine ursprüngliche Puffergröße  $N$ .

**[0112]** Als nächstes überträgt der Client **1** die Paketdaten [6], deren Größe in einer Größe  $n$  gespeichert werden kann, die der Server **2** mitgeteilt hat. Tatsächlich werden jedoch, in dem Schritt zum Empfangen dieses Teils von Paketdaten [6], die Paketdaten [1~5] verarbeitet, und daher existiert ein großer freier Platz im Puffer, so dass der Zustand hoher Belastung nicht auftritt.

**[0113]** Das heißt, in dem Ausführungsbeispiel **4** wird der Zustand, der in [Fig. 8](#) gezeigt ist, in der Regel nicht als die hohe Belastung beurteilt.

**[0114]** Aus den oben erläuterten Gründen wird das Paket  $P_i$  mit einem Einführen von  $P_i < P_j$  aus der Zählung ausgeschlossen. Alternativ kann der Wert durch Angeben eines bestimmten Gewichts oder durch weiteres Hinzufügen von  $P_j - P_i$  zu der Paketgröße in dem Belastungserkennungsmodul **10** berechnet werden.

**[0115]** Hier impliziert die Berechnung von  $P_j - P_i$ , dass, wenn kein Datenverlust auftritt durch Hinzufügen einer Vorhersagegröße der Daten, die in dem Puffer **51** innerhalb des Servers **2** gespeichert bleiben, zu der Paketgröße, eine Datengröße vorhergesagt werden kann, das heißt, eine vorliegende Paketgröße, die der Server **2** dem Client **1** mitteilt.

[Ausführungsbeispiel 5]

**[0116]** Ein Ausführungsbeispiel **5** ist, dass der Server **2** die Paketdaten überwacht, die an den Client **1** übertragen werden, wodurch die Last am Server **2** beurteilt wird.

**[0117]** In dem Belastungserkennungssystem **4** in dem Ausführungsbeispiel **5** überwacht der Server **2** einen Gesamtwert  $S_w$  einer Fenstergröße und eine Verbindungszählung  $C$  in dem Paket, welches durch den Server **2** an den Client **1** übermittelt wird. Die Fenstergröße ist eine Datengröße, die empfangen werden kann und die der Server **2** dem Client **1** mitteilt.

**[0118]** Ein Wert der Verbindungszählung  $C$  wird erhalten, indem um 1 erhöht wird, wenn das Startpaket SYN erkannt wird, welches von dem Server **2** an den Client **1** übermittelt wird, und indem um 1 erniedrigt wird, wenn das Endpaket FIN erkannt wird. Hier ist die Zählung von  $S_w$  und  $C$  die gleiche wie im Ausführungsbeispiel **1**.

**[0119]** Ein Indexwert  $L_3$  einer Belastung des Servers **2** wird durch die folgende Formel erhalten.  $T$  ist das gleiche wie  $T$  im Ausführungsbeispiel **1**, ist jedoch nicht notwendigerweise unbedingt notwendig.

$$L_3 = (S_w/c)/T \quad (4)$$

Der Indexwert L3 einer Belastung impliziert eine Fenstergröße pro Verbindung. Das Folgende ist ein Verfahren zum Erkennen der hohen Last am Server **2** durch die Verwendung von L3.

**[0120]** Zunächst wird ein Vorhersagewert L3max einer Durchsatzbeschränkung des Servers **2** aktualisiert. Ein anfänglicher Wert von L3max ist 0, und, wenn L3 L3max überschreitet, wird ein Wert von L3max auf L3 festgelegt.

**[0121]** Hier, wenn die folgende Beziehung zwischen L3 und L3max hergestellt wird, wird beurteilt, dass der Server **2** unter hoher Last ist.

$$L3 = \alpha_3 \cdot L_{max} \quad (5)$$

Mit  $0 < \alpha_3 \leq 1$ , und  $\alpha_3$  ist eine vorgegebene Konstante.

**[0122]** Der Server **2** teilt dem Client **1** eine freie Größe des Puffers **51** mit, bei der er selbst in der Lage ist zu Arbeiten, das heißt die Fenstergröße (**Fig. 9(a)**). Hier, wenn jedoch der Server **2** eine erhöhte Belastung besitzt, ausreichend um nicht in der Lage zu sein, die Daten vollständig zu verarbeiten, die vom Client **1** übertragen werden, wie in **Fig. 9(b)** gezeigt, teilt der Server **2** dem Client **1** die Fenstergröße  $n$  mit, die kleiner ist als davor (was insbesondere die Datenmenge ist, die das nächste Mal empfangen werden kann). Daher ist **Fig. 9(b)** ein Graph, welcher eine Beziehung zwischen der Zeit und der Fenstergröße zeigt, welche der Server **2** dem Client mitteilt.

**[0123]** Die Last am Server **2** hat einen Einfluss auf alle verbundenen Clients, und infolgedessen nimmt L3 mit einem Anstieg der Serverbelastung ab. Dementsprechend kann die Serverbelastung durch die Formel (4) berechnet werden, und die hohe Last kann durch die Formel (5) erkannt werden.

[Ausführungsbeispiel 6]

**[0124]** Ein Ausführungsbeispiel 6 zeigt einen Fall, bei dem ein Serveraufteilungssystem der vorliegenden Erfindung als ein Router zum Leiten des TCP-Pakets zwischen dem Client und dem Server aktualisiert wird.

**[0125]** Bezug nehmend auf **Fig. 10**, eine Adressübersetzungs-/Paketroutingeinheit **1002** gibt, wenn ein Paket **1010**, das vom Client **1** empfangen wird, das Startpaket SYN ist, welches die Verbindungsanforderung impliziert, einen Serveraufteilungshinweis **1020** an ein Serverauswahlmodul **1007** einer Serveraufteilungseinheit **1001** aus, um einen Server zu Aufteilen der Dienste zu bestimmen. Dann übergibt die Adressübersetzungs-/Paketroutingeinheit **1002** einen Zählungshinweis **1021** an ein Durchsatzzählungsmodul **1008** auf der Clientseite.

**[0126]** Eine Zählereinheit **1004** für einen Serverdurchsatz berechnet einen Durchsatz jedes Servers und sendet Ergebnisdaten **1013** an ein Berechnungsmodul **1006** für eine Serveraufteilungswahrscheinlichkeit. Der Durchsatz jedes Servers kann aus einer Antwortzeit nach der Übertragung eines Ping, etc. an den Server **2** berechnet werden, oder der Benutzer kann den Durchsatz vorgeben. Des Weiteren kann auch das Erkennungssystem für eine Serverbelastung, das in den Ausführungsbeispielen 1~5 beschrieben wird, verwendet werden.

**[0127]** Das Durchsatzzählungsmodul **1008** auf der Clientseite berechnet bei einem Hinweis, der von der Routingeinheit **1002** gegeben wird, Durchsätze **1018** des Clients **1** und der Kommunikationsleitung **3**, und informiert ein Datenerzeugungsmodul **1009** über eine Wahrscheinlichkeitsänderung der Serveraufteilung. Hier kann der Durchsatz auf der Clientseite aus einer Antwortzeit nach dem Übertragen erhalten werden, beispielsweise eines Ping, etc. an den Client. Des Weiteren kann der Durchsatz auf der Clientseite auch durch die Verwendung eines Bandmessungsverfahrens, wie z.B. Bprob etc., und aus vergangenen Aufzeichnungen der Kommunikationen über den Client **1** wie auch aus der Fenstergröße und TTL (Time-To-Live), was aus dem Paket extrahiert wird, erhalten werden.

**[0128]** Das Datenerzeugungsmodul **1009** für eine Änderung der Serveraufteilung erzeugt eine Änderungsfunktion **1022** in Bezug auf eine Wahrscheinlichkeitsverteilung der Serveraufteilung aus dem Durchsatz **1018** auf der Clientseite.

**[0129]** **Fig. 11** zeigt ein Beispiel einer Wahrscheinlichkeitsverteilung PsD einer Serveraufteilung. **Fig. 12** zeigt ein Beispiel einer Änderungsfunktion M (**1022**) in dem unteren Teil.

**[0130]** Das Berechnungsmodul **1006** für eine Serveraufteilungswahrscheinlichkeit wendet die Änderungsfunktion M (**1022**) auf die Wahrscheinlichkeitsverteilung PsD der Serveraufteilung an, wodurch eine Wahrscheinlichkeitsverteilung MPsD der Serveraufteilung erhalten wird.

**[0131]** Die Wahrscheinlichkeitsverteilung PsD wird mit solch einer Verteilung festgelegt, dass die Aufteilungswahrscheinlichkeit größer wird, wenn der Server einen höheren Durchsatz zu der vorgegebenen Zeit besitzt. Wenn beispielsweise  $p_1, p_2, \dots, p_n$  ( $n$  ist die Anzahl von Server) Durchsatzwerte (was später erklärt werden wird) der jeweiligen Server zu der vorgegebenen Zeit sind, kann eine Aufteilungswahrscheinlichkeit  $P_i$  auf den Server  $S_i$  durch die folgende Formel erhalten werden:

$$P_i = p_i / (p_1 + p_2 + \dots + p_n) \quad (6)$$

**[0132]** Die Änderungsfunktion  $M$  der Wahrscheinlichkeit wird, wie in [Fig. 12](#) gezeigt, eine Funktion zum Ändern von  $P_{sD}$ , so dass  $P_{sD}$  besser an eine gleichmäßige Verteilung angenähert wird, wenn der Durchsatz auf der Clientseite niedriger wird. Wenn beispielsweise eine Antwortzeit  $T_{ping}$ , basierend auf einem  $g_{ing}$ , als ein Durchsatz auf der Clientseite behandelt wird, kann ein geändertes  $P_i'$  jedes Serverdurchsatzes  $P_i$  auch aus der folgenden Formel erhalten werden:

$$P_i' = P_i + (P_{av} - P_i) \cdot 2/\pi \cdot \arctan(\alpha \cdot T_{ping}) \quad (7)$$

wo  $P_{av}$  ein Mittelwert von  $P_i$  ist,  $\alpha$  eine vorgegebene Zahl größer als 0 ist und  $\arctan(x) \tan^{-1}(x)$  repräsentiert.

**[0133]** Die modifizierte Wahrscheinlichkeitsverteilung  $MP_{sD}$  wird aus diesem  $P_i'$  erhalten.

**[0134]** Das Berechnungsmodul **1006** für eine Serveraufteilungswahrscheinlichkeit sendet die erhaltene Verteilung  $MP_{sD}$  an das Serveraufteilungsmodul (**1007**). Das Serveraufteilungsmodul (**1007**) wird durch Erzeugen einer Tabelle, die in [Fig. 13](#) gezeigt ist, aus  $MP_{sD}$  aktualisiert, wobei einheitliche Zufallswerte verwendet werden, die zufällige Werte von 0~1 annehmen. Die Tabelle in [Fig. 13](#) wird beispielsweise durch ein Array von Feldern aktualisiert, von denen jedes eine Servernummer enthält, wobei jedes Feld ein Tupel von Maximal- und Minimalwerten in einem Bereich  $P_i$  von 0~1 und eine Serveradresse enthält. Die Serveradresse in dem Feld, welches einen Bereich besitzt, der die gleichmäßige Zufallszahl enthält, kann als eine Serveradresse für eine Dienstaufteilung festgelegt werden. Der Bereich jedes Felds ist jedoch entworfen, um nicht mit den Bereichen anderer Felder zu überlappen.

**[0135]** In Bezug auf die Wahrscheinlichkeitsverteilung von  $P_{sD}$  und  $MP_{sD}$  können die Serverdurchsatzwerte  $P_i$  und  $P_i'$  als eine Verteilungsmaß aktualisiert werden. In diesem Fall nehmen die gleichmäßigen Zufallswerte einen Bereich von 0 bis zu einem Gesamtwert aller  $P_i$  an.

**[0136]** Das Serverauswahlmodul **1007** sendet nach dem Bestimmen des Aufteilungsservers eine Serveradresse **1012** davon an die Verbindungsverwaltungseinheit **1003**.

**[0137]** Die Verbindungsverwaltungseinheit **1003** extrahiert ein Tupel von Daten, wie z.B. die Clientadresse (IP), die Clientportnummer (sp), die Zielportnummer (dp), aus dem Startpaket SYN oder einem Teil dieses Pakets, das von der Adressübersetzungs-/Paketroutingeinheit empfangen wird. Dann zeichnet die Verbindungsverwaltungseinheit **1003** ein Mapping des Tupels von Daten und die Serveradresse auf, die von der Serveraufteilungseinheit **1001** erhalten wird.

Hier kann die Aufzeichnung die Verwendung einer Hash-Tabelle bedingen, wobei das Tupel von Daten als ein Schlüssel dient. Die Verbindungsverwaltungseinheit **1003** überträgt die Serveradresse **1012** an die Adressübersetzungs-/Paketroutingeinheit **1002**.

**[0138]** Die Adressübersetzungs-/Paketroutingeinheit **1002** übersetzt ein Ziel des Pakets, das vom Client **1** empfangen wird, in die Serveradresse **1012**, die von der Verbindungsverwaltungseinheit **1003** empfangen wird, und überträgt die Serveradresse **1012** an den Server **2**.

**[0139]** Während der Servierung übermittelt die Adressübersetzungs-/Paketroutingeinheit **1002** ein Paket **1014** an die Verbindungsverwaltungseinheit **1003**. Diese Verbindungsverwaltungseinheit **1003** erhält die Aufteilungsserveradresse **1012** von dem Tupel von Daten, die aus dem Paket **1014** extrahiert werden, und sendet die Adresse **1012** an die Adressübersetzungs-/Paketroutingeinheit **1002**. Wie in dem Fall des Startpakets SYN übersetzt die Adressübersetzungs-/Paketroutingeinheit **1002** das Ziel des Pakets, das von dem Client **1** empfangen wird, in die Serveradresse **1012**, die von der Verbindungsverwaltungseinheit **1003** erhalten wird, und überträgt die Serveradresse **1012** an den Server **2**.

**[0140]** Wenn die Servierung beendet ist, das heißt, wenn das Endpaket FIN empfangen wird, obwohl ähnlich zu dem Betriebsprozess, verwirft die Verbindungsverwaltungseinheit **1003**, welche dieses Paket empfängt, das Tupel von Daten entsprechend dem Paket.

**[0141]** In diesem Ausführungsbeispiel wird die Dienstaufteilung durch Verwendung der Wahrscheinlichkeitsverteilung bestimmt, wodurch ein Aufteilen des Servers vereinfacht wird, welcher einen höheren Durchsatz für den Client aufweist, der einen höheren Durchsatz aufweist. Daher können die Dienste entsprechend danach aufgeteilt werden, wie stark der Serverdurchsatz die QoS beeinflusst, wie z.B. die Antwortzeit.

**[0142]** Das Datenerzeugungsmodul **1009** für eine Wahrscheinlichkeitsänderung der Serveraufteilung der Serveraufteilungseinheit **1001** erhält eine Verteilung ([Fig. 14](#)) der Durchsätze auf der Clientseite in der Vergangenheit, erhält auch einen Zwischenraum **6** aus der Verteilung der Durchsatzwerte auf der Clientseite eines neu angeschlossenen Clients, und erhält einen Zwischenraum **6c** aus einer Verteilung der Durchsatzwerte auf der Clientseite eines neu angeschlossenen Clients. Dann fügt das Datenerzeugungsmodul **1009** für eine Wahrscheinlichkeitsänderung der Serveraufteilung **6c** zu der Änderungsfunktion  $M$  hinzu, wodurch die Wahrscheinlichkeitsverteilung verändert wird ([Fig. 15](#)). Hier wird  $\delta c$  beispielsweise durch die folgende Formel erhalten:

$$\delta c = P_{ca} - p_{ci}$$

wo  $P_{ca}$  ein Durchschnittswert auf der Clientseite in der Vergangenheit ist und  $p_{ci}$  ein Durchschnittswert auf der Clientseite des neu angeschlossenen Clients ist. Die Änderungsfunktion  $M$  wird so festgelegt, dass der Serverdurchsatzwert  $p_i$  besser an den Durchschnittswert aller  $p_i$  angenähert wird, wenn  $\delta c$  kleiner wird, und wird von dem Durchschnittswert weiter entfernt, wenn  $\delta c$  größer wird. Wenn der Zwischenraum von dem Durchschnittswert groß festgelegt wird, muss der modifizierte Wert  $p_i'$  von  $p_i$  keine negative Zahl sein. Beispielsweise kann die Formel (7) wie folgt verändert werden:

$$p_i' = p_i + (P_{av} - p_i) \cdot \beta \cdot 2/\pi \cdot \arctan(\alpha \cdot \delta c + \gamma) \quad (7')$$

wo  $P_{av}$  ein Mittelwert von  $p_i$  ist,  $\alpha$ ,  $\gamma$  vorgegebene Zahlen größer als 0 sind und  $\arctan(x)$   $\tan^{-1}(x)$  repräsentiert. Wenn  $\beta - 1$  ist,  $\delta c < 0$  und wenn  $-dp_i/p_i$ ,  $\delta > 0$ .  $P_j$  ist der Mindestwert von  $p_i$ , und  $dp_i = P_{av} - p_j$ .

**[0143]** Somit wird die Dienstaufteilung basierend auf dem Zwischenraum aus der Verteilung der vergangenen Durchsatzwerte auf der Clientseite der Durchsatzwerte auf der Clientseite des neu verbundenen Clients ausgeführt, wodurch die Serveraufteilung entsprechend dem Client zu jedem Zeitpunkt erreicht werden kann. Beispielsweise kann die Serveraufteilung automatisch einem Fall entsprechen, indem die Clientkennzahlen von einem entfernten Ort und einem benachbarten Ort abhängig von einer Zeitzone schwanken.

**[0144]** Des Weiteren kann in diesem Ausführungsbeispiel eine Mehrzahl von Serveraufteilungseinheiten (**1001**) angeordnet werden und jede ausgewählte entspricht der Clientadresse, der Clientportnummer und der Dienstportnummer.

**[0145]** Eine Aufteilungszielservergruppe kann separat pro Dienst und pro Client verwendet werden, und eine Dienstverteilungsmethode kann umgeschaltet werden, wodurch ein einzelnes System in der Lage ist, die Dienstaufteilung in einer Vielzahl von Arten auszuführen.

**[0146]** Wie oben diskutiert wird gemäß der vorliegenden Erfindung die Serverbelastung gemessen, und die hohe Belastung wird durch Überwachen der Kommunikationen zwischen dem Client und dem Server erkannt. Infolgedessen wird weder eine technische Ergänzung am Server benötigt noch werden andere Pakete als für die Dienste ausgegeben. Dementsprechend gibt es sich ergebende Effekte, wobei das System in der Lage ist, jedem Server zu entsprechen, und es tritt keine Interferenz mit der Last auf, mit niedrigen Kosten für die Einführung. Des Weiteren werden die Belastungsmessung und die Erkennung

der hohen Belastung mit dem Index durchgeführt, der nicht vom Protokoll abhängt, so dass ein Effekt ist, dass das System in der Lage ist, irgendeinem Dienst zu entsprechen. Es gibt auch einen Effekt, bei dem der Einfluss der Störung gering ist und die Genauigkeit hoch ist, aufgrund der Überwachung des Kommunikationszustands während dem Dienst.

**[0147]** Außerdem, wenn die Dienste, die durch den Server bereitgestellt werden, durch die Vielzahl von Servern aufgeteilt werden, werden die Belastungen der jeweiligen Server automatisch effizient aufgeteilt, entsprechend dazu, wie stark sich der Serverdurchsatz, aus Sicht des Clients auf die QoS auswirkt, entsprechend den Änderungen der Serverarchitektur und des Serverzustands. Daher ist ein Effekt, dass der Client eine prompte Lieferung der Dienste empfangen kann.

**[0148]** Gewerbliche Anwendbarkeit

**[0149]** Die vorliegende Erfindung kann auf das Belastungsmesssystem in dem Netzwerksystem angewendet werden, welches durch die Server und die Clients konfiguriert wird.

## Patentansprüche

1. Ein Erkennungsverfahren für eine Netzwerkserversbelastung zur Verwendung in einem Erkennungssystem (**4**), das an eine Kommunikationsleitung (**3**) zwischen einem Server (**2**) und einem Client (**1**) angeschlossen ist, welche durch Datenpakete miteinander kommunizieren, umfassend die folgenden Schritte:

- Überwachen einer Kommunikation von dem Client (**1**) zu dem Server (**2**), und Zählen der Kommunikationsdatengröße pro Verbindung als die Belastung des Servers;
- Erkennen einer Änderung der Kommunikationsdatengröße für jede Verbindung, und Aufzeichnen des Maximalwerts der Größe; und
- Beurteilen, wenn die Kommunikationsdatengröße pro Verbindung zu diesem Zeitpunkt in Bezug auf den Maximalgrößenwert abnimmt, dass der Server unter einer hohen Belastung ist.

2. Ein Erkennungsverfahren für eine Netzwerkserversbelastung nach Anspruch 1, des Weiteren umfassend einen Schritt zum Zählen einer Verbindungszählung und der Kommunikationsdatengröße bis die Zählung überwachter Kommunikationen eine Mindestzählung überwachter Kommunikationen erreicht und die Zählungszeit eine Überwachungsmindestzeit erreicht, durch Verwendung der Mindestzählung überwachter Kommunikationen und der Überwachungsmindestzeit.

3. Ein Erkennungsverfahren für eine Netzwerkserversbelastung nach Anspruch 1 oder 2, des Weiteren

ren umfassend einen Schritt zum Erkennen der Kommunikationen eines Starts und Endes der Verbindung, und Ausschließen von Kommunikationsdatengrößen des Starts und Endes der Verbindung von einem Belastungserkennungsziel.

4. Ein Erkennungsverfahren für eine Netzwerksverbelastung nach einem der vorhergehenden Ansprüche, des Weiteren umfassend:  
einen Schritt zum Beibehalten von Information der Kommunikation des Starts der Verbindung bis die Verbindung beendet oder hergestellt ist;  
einen Schritt zum Erkennen der Kommunikation des Starts der Verbindung für eine erneuten Verbindung, der ausgeführt wird, wenn beurteilt wird, dass sich der Client nicht basierend auf der beibehaltenen Information verbindet; und  
einen Schritt zum Festlegen der Rate, bei welcher die Kommunikation der erneuten Verbindung die Anzahl der Kommunikationen des Starts der Verbindung als die Belastung des Servers belegt, und, wenn diese Rate hoch ist, Beurteilen, dass der Server unter einer hohen Belastung ist.

5. Ein Erkennungsverfahren für eine Netzwerksverbelastung nach einem der vorhergehenden Ansprüche, des Weiteren umfassend:  
einen Schritt zum Erhalten der Verteilung der Kommunikationsdatengrößen von den Clients;  
einen Schritt zum Unterscheiden zwischen extrem kleinen Teilen von Kommunikationsdaten, die nicht mit der Belastung des Servers in Beziehung stehen, von der Verteilung der Kommunikationsdatengröße; und  
einen Schritt zum Eliminieren der extrem kleinen Teile von Kommunikationsdaten von der Beurteilung über die Belastung.

6. Ein Erkennungsverfahren für eine Netzwerksverbelastung nach einem der vorhergehenden Ansprüche, des Weiteren umfassend:  
einen Schritt zum Erhalten einer Laufnummer aus der Kommunikation von dem Client zu dem Server;  
einen Schritt zum Beibehalten des Maximalwerts der Laufnummer bis die Verbindung beendet ist, von dem Start der Verbindung an;  
einen Schritt zum Vergleichen der Laufnummer der empfangenen Kommunikation mit der beibehaltenen Laufnummer; und  
einen Schritt zum Ausschließen dieser Kommunikation vom Zählen, wenn die Laufnummer, die aus der Kommunikation erhalten wird, kleiner ist als die beibehaltene Laufnummer.

7. Ein Erkennungsverfahren für eine Netzwerksverbelastung nach einem der vorhergehenden Ansprüche, des Weiteren umfassend:  
einen Schritt zum Zählen, wenn die Laufnummer, die aus der Kommunikation erhalten wird, kleiner ist als die beibehaltene Laufnummer, der Kommunikations-

daten nach einem Ausführen eines Gewichtungsprozesses daran, oder Vorhersagen einer Kommunikationsdatengröße, wenn es auf einer Strecke der zwei Laufnummern kein Problem gibt, und Zählen der vorhergesagten Datengröße zum Erkennen der Belastung.

8. Ein Erkennungsverfahren für eine Netzwerksverbelastung zur Verwendung in einem Erkennungssystem (4), das an eine Kommunikationsleitung (3) zwischen einem Server (2) und einem Client (1) angeschlossen ist, welche durch Datenpakete miteinander kommunizieren, umfassend die folgenden Schritte:  
– Überwachen einer Kommunikation von dem Server (2) zu dem Client (1), und Zahlen der empfangbaren Datengröße und der Verbindungszählung, welche der Server dem Client mitteilt;  
– Erhalten der empfangbaren Datengröße für jede Verbindung als die Serverbelastung; und  
– Speichern des Maximalwerts der empfangbaren Datengröße pro Verbindung, und Beurteilen, wenn die empfangbare Datengröße pro Verbindung klein wird in Bezug auf den Maximalwert, dass der Server unter hoher Belastung ist.

9. Ein Verfahren zum Betreiben eines Netzwerks, welches eine Mehrzahl von Servern (2) umfasst, wobei die Belastungen der Server durch ein Verfahren gemäß einem der vorhergehenden Ansprüche überwacht werden, und eine Serverauswahl basierend auf den so bestimmten Serverbelastungen durchgeführt wird.

10. Ein Erkennungssystem (4) für eine Netzwerksverbelastung zum Überwachen einer Kommunikation von einem Client (1) zu einem Server (2) und Erkennen des Belastungszustands des Servers, umfassend:  
– Berechnungsmittel (7) für eine Datengröße zum Berechnen der Größe von Kommunikationsdaten pro Verbindung;  
– Speichermittel (12) zum Erkennen einer Änderung der Kommunikationsdatengröße pro Verbindung und Speichern eines Maximalwerts; und  
– Belastungserkennungsmittel (10) zum Erkennen einer hohen Belastung des Servers, wenn die Kommunikationsdatengröße pro Verbindung zu diesem Zeitpunkt in Bezug auf den Maximalwert gleich oder kleiner ist als ein festgelegter Wert.

11. Eine Netzwerksverteilungsvorrichtung, umfassend ein Belastungserkennungssystem (4) gemäß Anspruch 10.

Es folgen 13 Blatt Zeichnungen

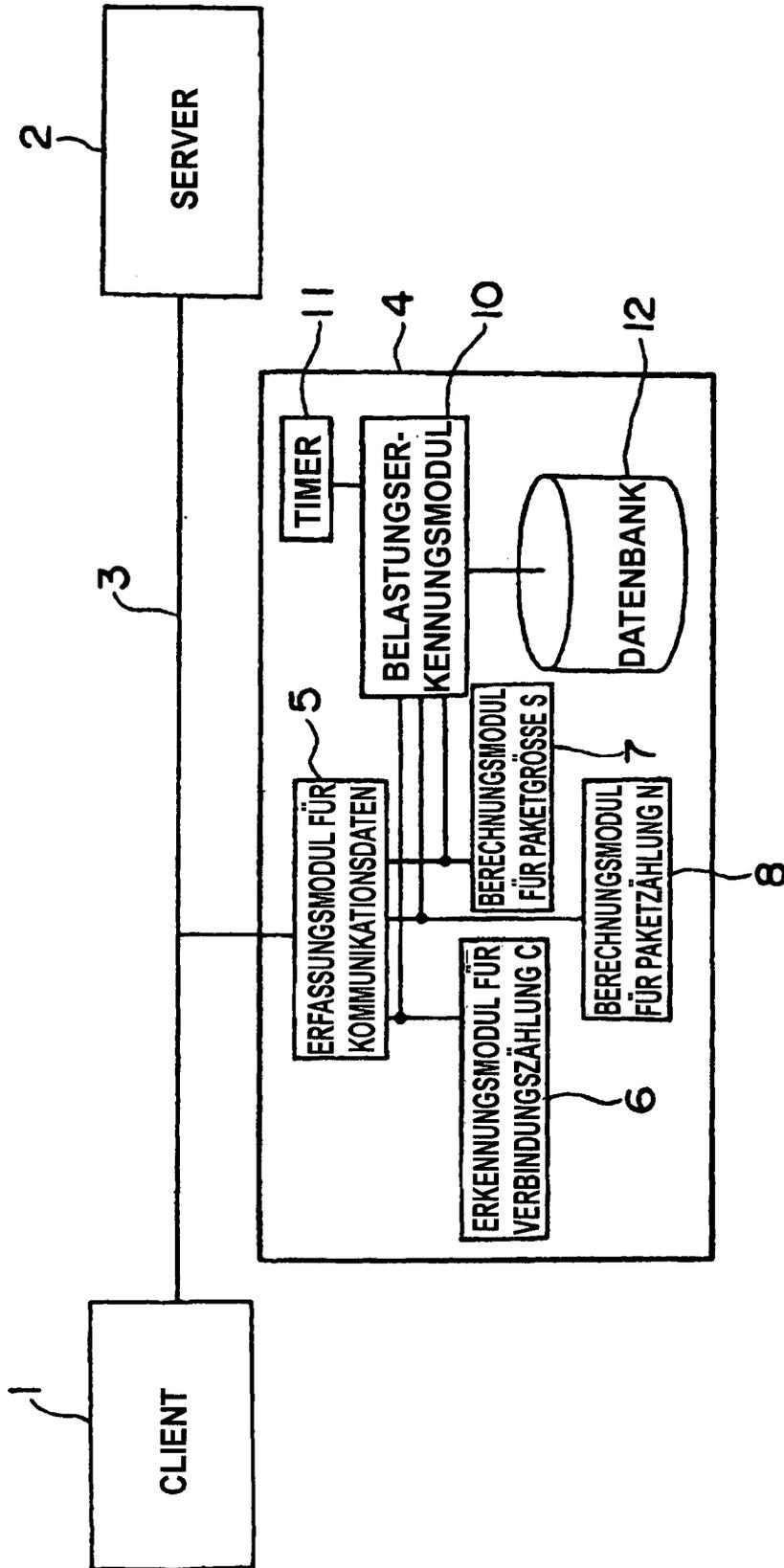


FIG. 1

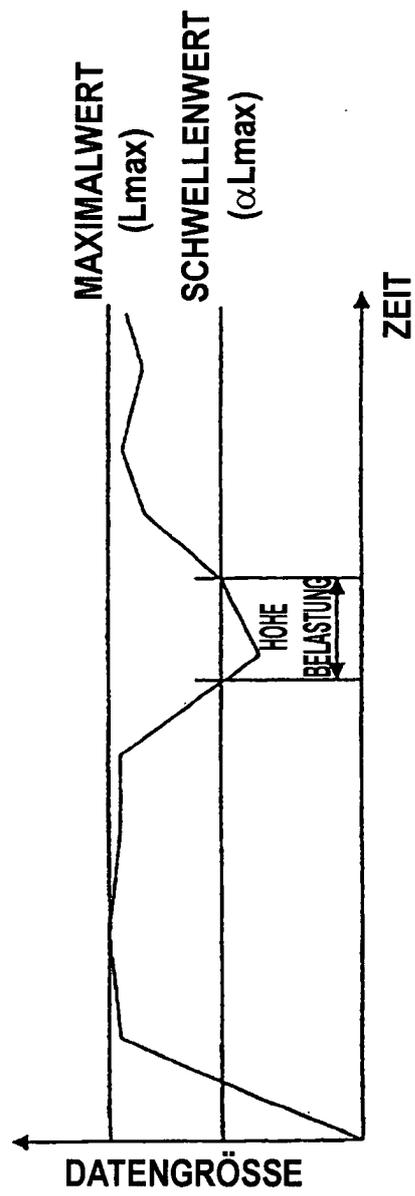


FIG. 2

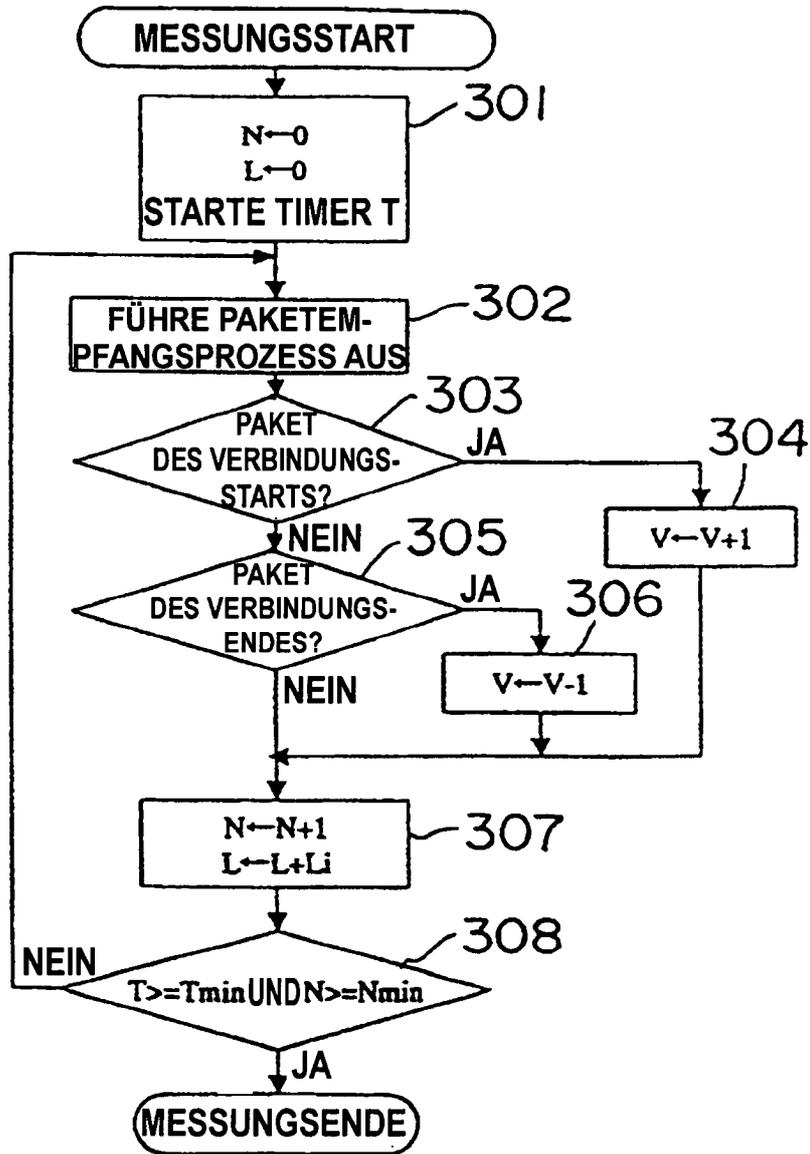


FIG. 3

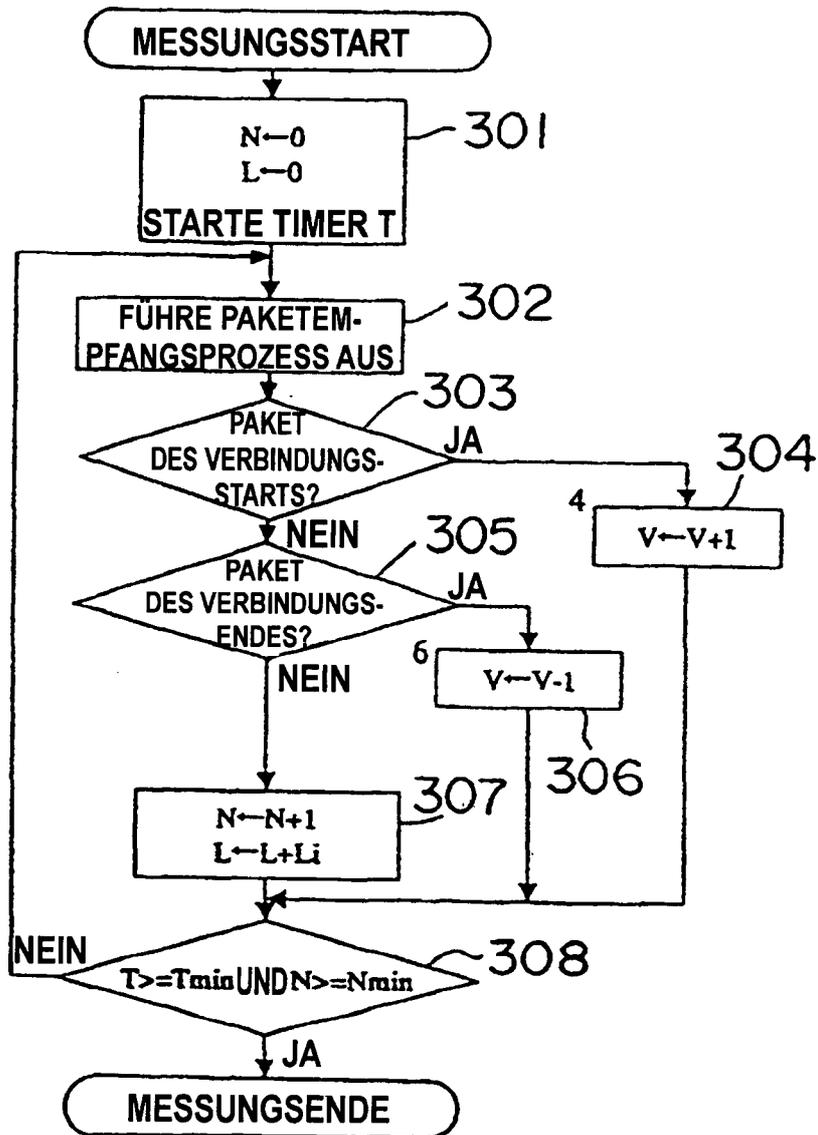


FIG. 4

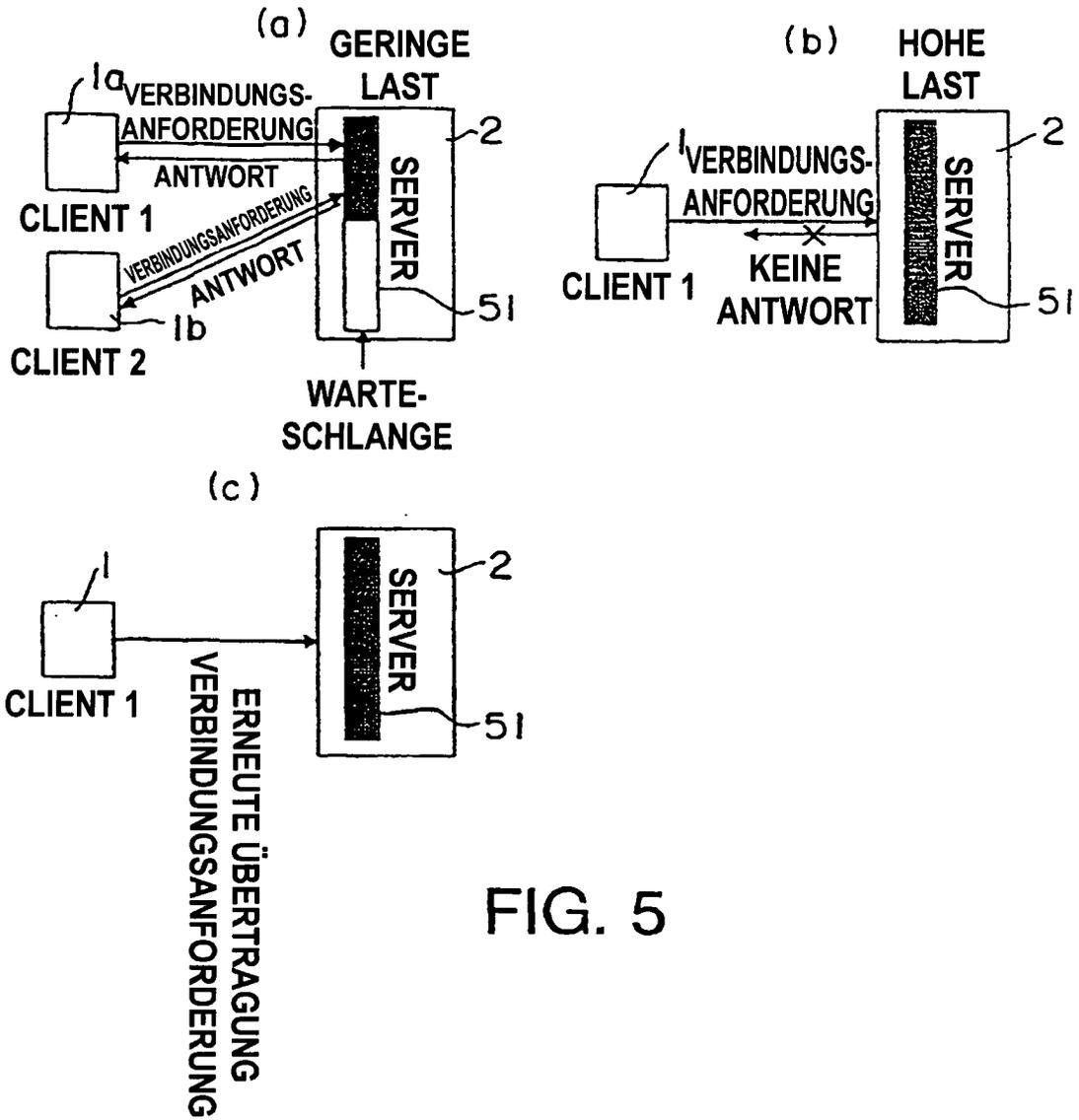


FIG. 5

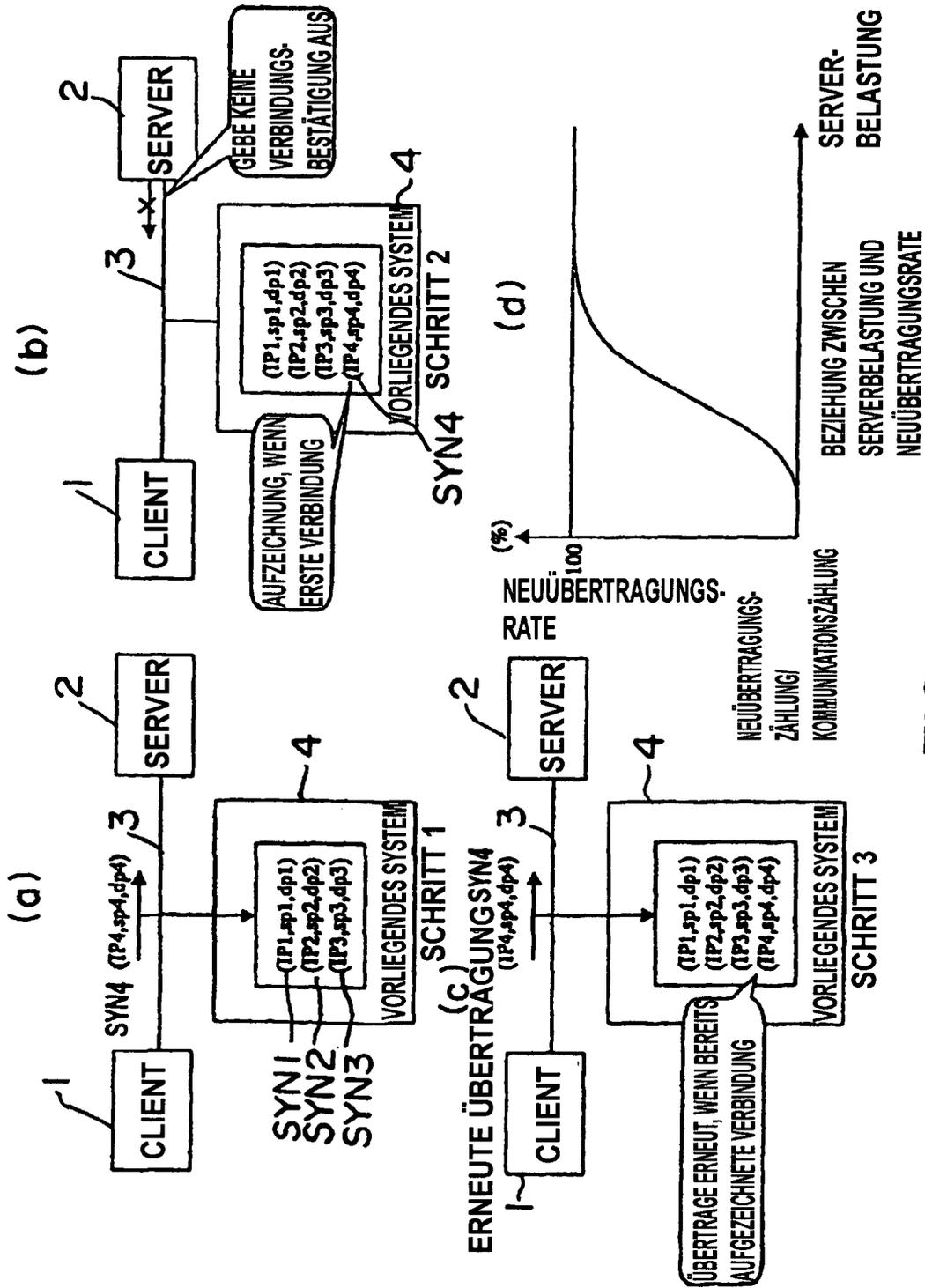


FIG. 6

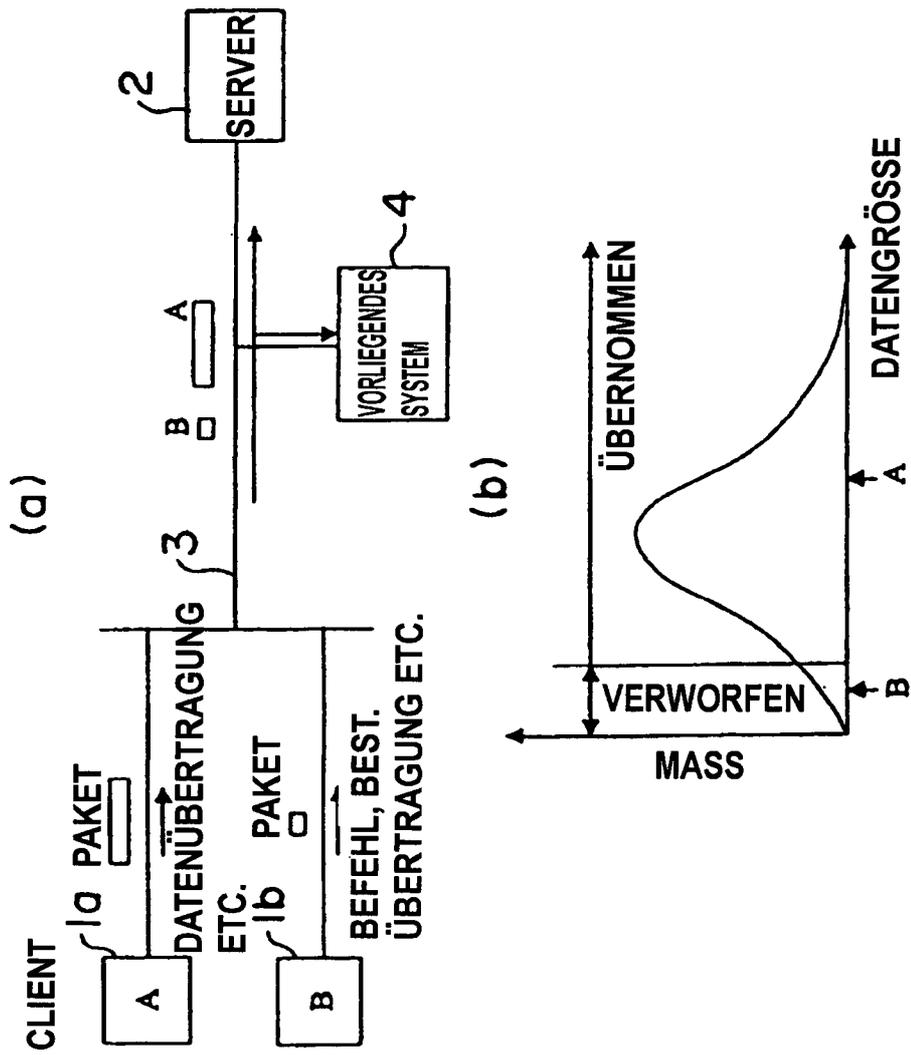


FIG. 7

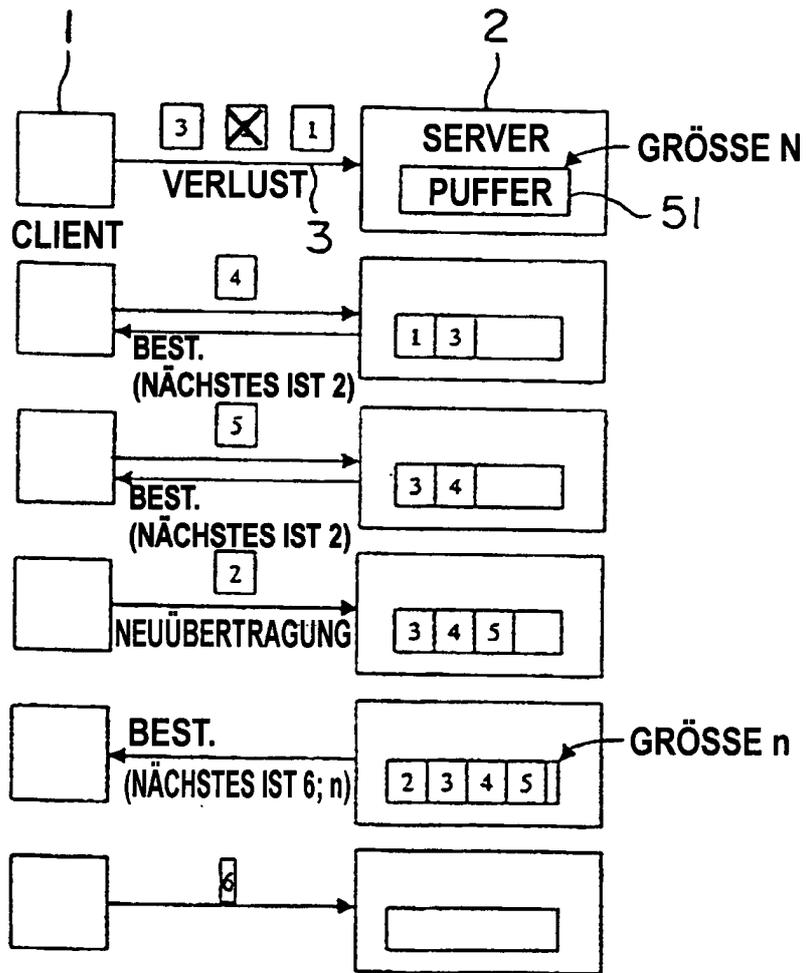


FIG. 8

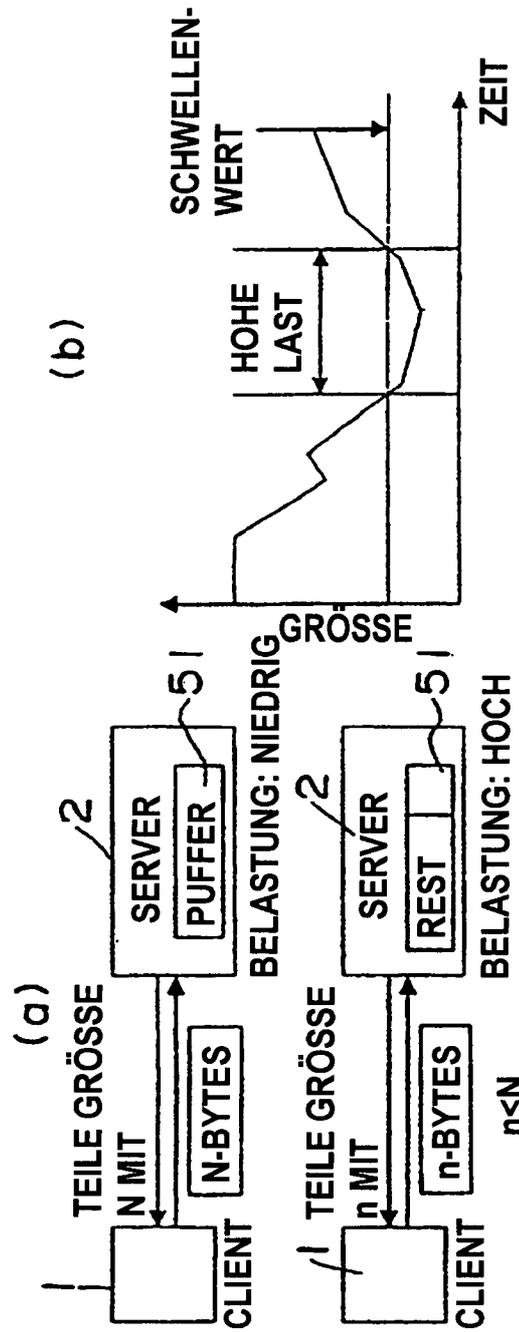


FIG. 9

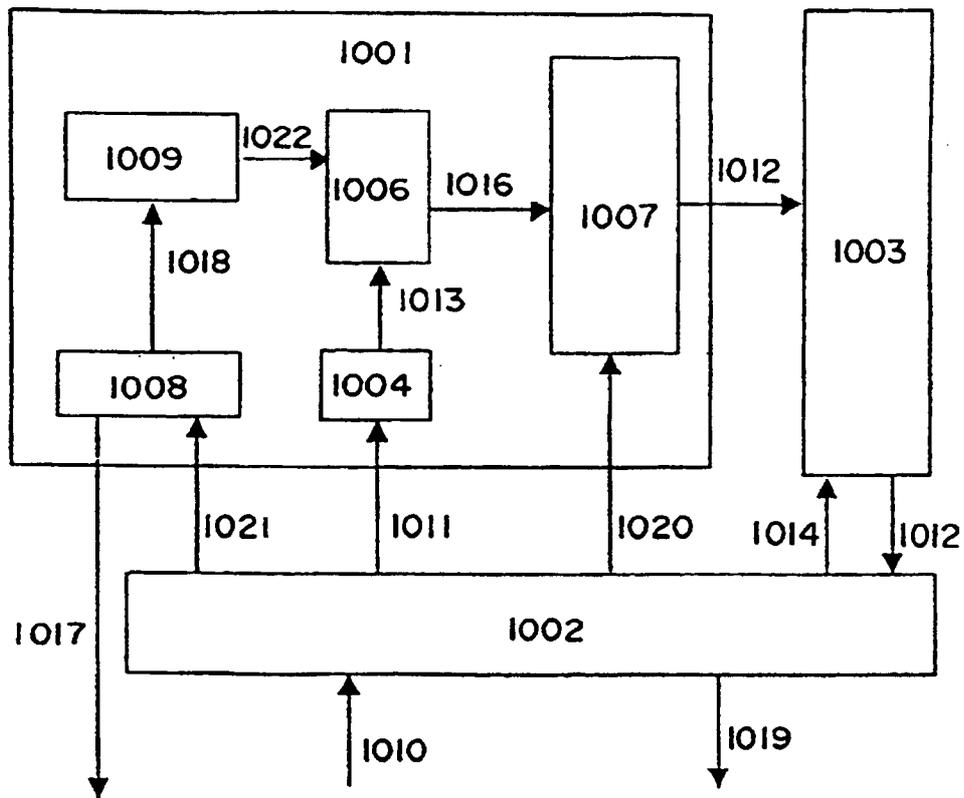


FIG. 10

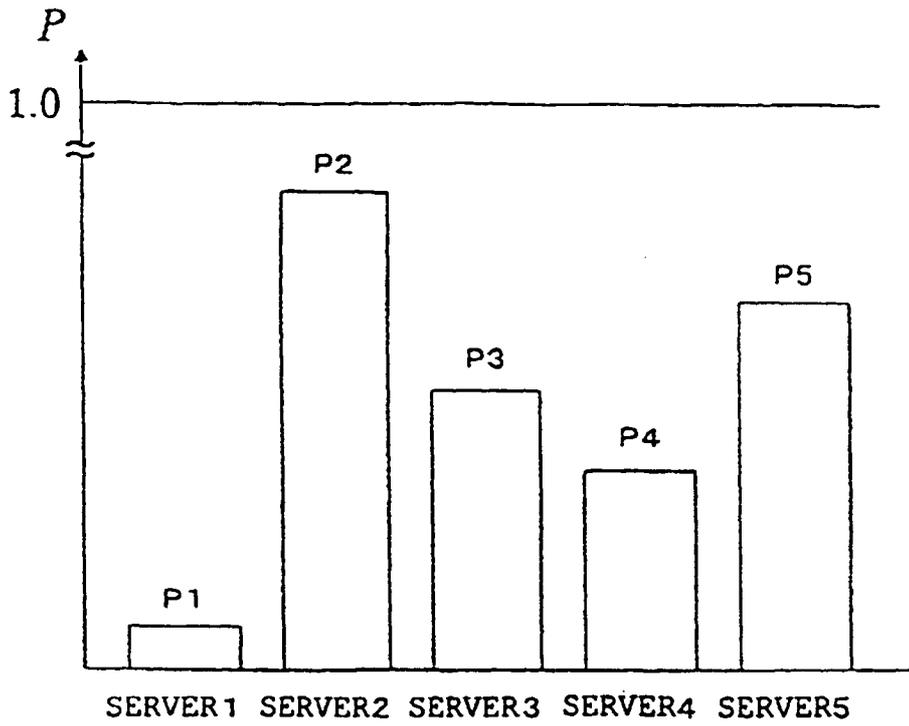


FIG. 11

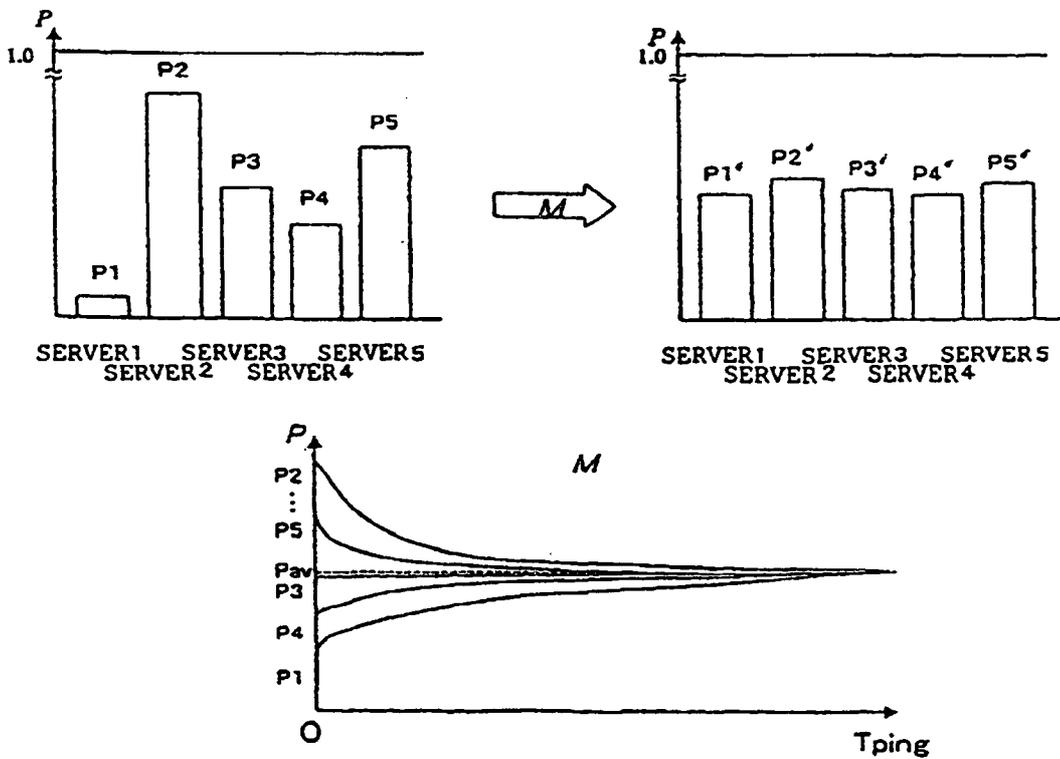


FIG. 12

#0	#1	#2	#3	#4
0~0.04 SERVER 1	0.05~0.34 SERVER 2	0.35~0.55 SERVER 3	0.56~0.80 SERVER 4	0.81~1.00 SERVER 5

FIG. 13

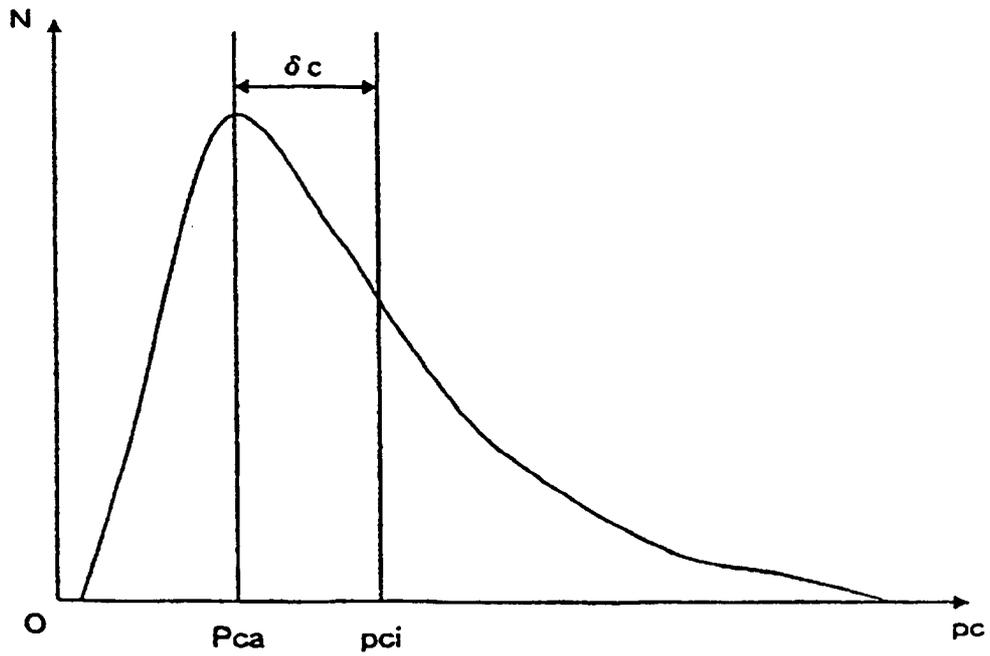


FIG. 14

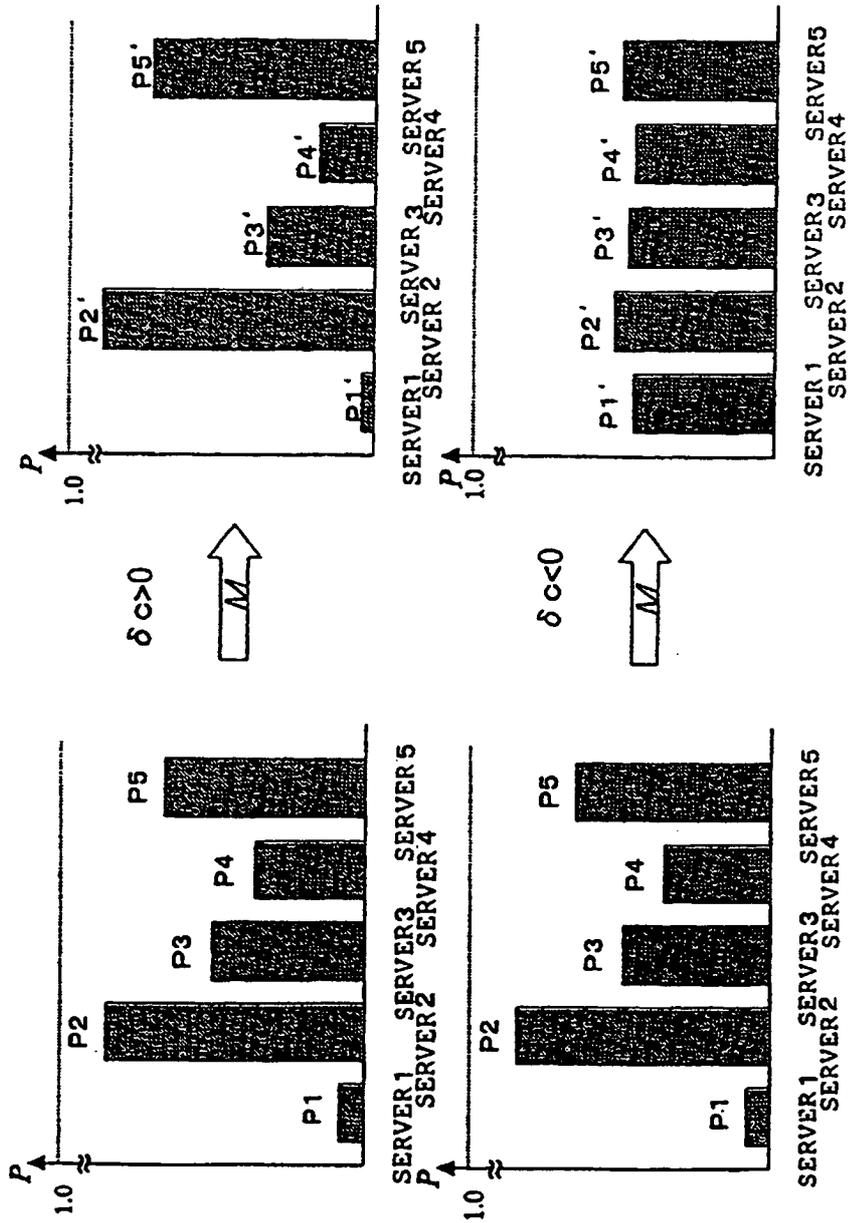


FIG. 15