



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 12 355 T2** 2007.11.08

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 550 274 B1**

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

(21) Deutsches Aktenzeichen: **603 12 355.4**

(86) PCT-Aktenzeichen: **PCT/IB03/04455**

(96) Europäisches Aktenzeichen: **03 751 098.9**

(87) PCT-Veröffentlichungs-Nr.: **WO 2004/034653**

(86) PCT-Anmeldetag: **09.10.2003**

(87) Veröffentlichungstag
der PCT-Anmeldung: **22.04.2004**

(97) Erstveröffentlichung durch das EPA: **06.07.2005**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **07.03.2007**

(47) Veröffentlichungstag im Patentblatt: **08.11.2007**

(30) Unionspriorität:
417651 P 11.10.2002 US

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

(73) Patentinhaber:
Nokia Corp., Espoo, FI

(72) Erfinder:
**CHEN, Maoke, 100084 Beijing, CN; HUANG, Hui,
100084 Beijing, CN; LI, Xing, 100084 Beijing, CN;
YAN, Cheng, 100084 Beijing, CN; LIU, Xin, 100084
Beijing, CN**

(74) Vertreter:
TBK-Patent, 80336 München

(54) Bezeichnung: **DYNAMISCHES PEER TUNNELING MIT LEISTUNGSOPTIMIERUNG**

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

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich auf ein Verfahren und ein Netzwerksystem zum Konfigurieren von Verbindungen zwischen einer Vielzahl von Netzwerkknoten.

HINTERGRUND DER ERFINDUNG

[0002] Es gibt eine Vielzahl von Netzwerken verschiedener Arten, d.h. Netzwerken mit verschiedenen Protokollen, wie etwa IPv4 (Internetprotokoll-Version 4) und IPv6 (Internetprotokoll-Version 6). Manche dieser Netzwerke sind weit verbreitet, so dass sie ein großes Gebiet abdecken (z.B. IPv4-Internet). Andere Netzwerke werden nur in vereinzelt Gegenden angewendet (z.B. IPv6-Internet, welches momentan nur an vereinzelt Standorten verwendet wird). Es ist erwünscht, diese vereinzelt Netzwerke der gleichen Art zu verbinden. Für diese Verbindung wurde ein „Tunnel“-Konzept vorgeschlagen.

[0003] Ein Tunnel ist eine virtuelle Verbindung zwischen zwei Netzwerkknoten. Das heißt, Tunneln funktioniert durch ein Einkapseln eines Protokolls des ersten Netzwerks in Pakete, die durch das zweite Netzwerk getragen werden. Im Fall von IPv6 und IPv4 bedeutet dies, dass das IPv6-Protokoll in die IPv4-Pakete eingebettet wird.

[0004] Ein anderes Beispiel ist das virtuelle private Netzwerk (VPN). In diesem Fall sind Organisationen dazu in der Lage, das Internet zu verwenden, um Daten über das VPN zu übertragen. Dies wird durch Einbetten des VPN-Netzwerkprotokolls in die TCP/IP-Pakete, die durch das Internet getragen werden, durchgeführt.

[0005] Folglich spielen solche Tunnel bei virtuellen Netzwerken bzw. vernetztem Arbeiten eine wichtige Rolle. Bisher wurde eine Konfiguration der Tunnel manuell durchgeführt, was mühevoll ist und viel Arbeit erfordert. Um die niedrige Leistungsfähigkeit solch einer manuellen Tunnelkonfiguration zu überwinden, wurden einige automatische Tunnel-Ansätze, wie etwa Tunnel-Broker (TB) (siehe z.B., A. Durand, P. Fasano, D. Lento, „IPv6 Tunnel-Broker“, RFC 3053, Januar 2001) und 6-zu-4 implizierter zustandsloser Tunnel (siehe z.B., B. Carpenter, K. Moore, „Connection of IPv6 Domains via IPv4 Clouds“, RFC 3056, Februar 2001), entwickelt und in IPv6-Netzwerken eingesetzt. Bei VPN (virtuelles privates Netzwerk)-Techniken kombinieren Tunnel all die Knoten, die unter geografisch unterschiedlichen Standorten verstreut sind, als ein einheitliches logisches Netzwerk.

[0006] 6-zu-4 (der Verbindungsmechanismus von IPv6-Domains über IPv4-Wolken, wie vorstehend erwähnt) ist eine zustandslose Lösung zum automatischen Tunneln von IPv6-„Inseln“, die durch IPv4-„Seen“ getrennt sind, angesichts eines spezifischen IPv6-Adressformats. Folgerichtig ist jedes Paar von Peer-Standorten in 6-zu-4 direkt im Sinne eines virtuellen Netzwerks verbunden, d.h. es gibt keine IPv6-Weiterleitung zwischen den Peers und das virtuelle Netzwerk (VN) bildet eine voll vernetzte Topologie. Da IPv6-Pakete von jedem Peer zu einem anderen nur über IPv4-Router gesendet werden, ist die Leistungsfähigkeit einer IPv6-Sitzung die gleiche wie die auf einem IPv4-Ende-zu-Ende-Pfad zwischen den entsprechenden Knoten.

[0007] Bei dem Tunnel-Broker-Ansatz macht der zustandsbehaftete Broker-Service die Adressierung flexibel. Bei dem Tunnel-Broker-System wird jedoch ein Tunnel-Server (TS) eines Weiterleitungszentrums für eine Gruppe von Tunnel-Klienten bereitgestellt. Jeder Tunnel-Klient (TC) hat einen vorgegebenen Leitweg zu dem anderen Teil der IPv6-Welt über den Tunnel-Server und jedes Paar von Tunnel-Klienten muss auf jeden Fall über die Weiterleitung des Tunnel-Servers kommunizieren, auch wenn eine direktes Tunneln der zwei Tunnel-Klienten weitaus besser wäre. Dann hängt die Leistungsfähigkeit einer IPv6-Sitzung zwischen zwei Tunnel-Klienten von einem Ende-zu-Ende-Verhalten zwischen dem Tunnel-Server zu beiden von diesen ab.

[0008] Beide vorstehend erwähnten Verfahren stellen nicht die Fähigkeit einer dynamischen Tunneländerung gemäß dem Leistungsfähigkeitsverhalten einer virtuellen Verbindung (Tunnel) bereit.

[0009] Die Druckschrift WO 0079730 offenbart ein Verfahren, bei dem in einem Überlappungsnetzwerk optimierte Leitweglenkungspfade durch Verwendung gemessener NetzwerkLeistungsfähigkeitsmetriken bestimmt werden.

[0010] Bis jetzt umfasst jedoch keine existierende Tunneltechnik eine Betrachtung des Leistungsfähigkeitsproblems, d.h. ein Anpassen des virtuellen Netzwerkprozesses zu der Leistungsfähigkeit und ihrer Variation über

der IPv4-Infrastruktur.

KURZFASSUNG DER ERFINDUNG

[0011] Die Aufgabe, die der vorliegenden Erfindung zugrunde liegt, liegt im Bereitstellen eines Verfahrens und eines Systems, durch welche virtuelle Verbindungen zwischen Netzwerkknoten verlässlich und effizient konfiguriert werden können.

[0012] Diese Aufgabe wird durch ein Verfahren gemäß Anspruch 1 gelöst.

[0013] Alternativ wird die vorstehende Aufgabe durch ein Netzwerk gemäß Anspruch 16 gelöst.

[0014] Somit ist es möglich, die Qualität von virtuellen Direktverbindungen (z.B. Tunneln) zwischen den Netzwerkknoten zu überwachen. Folglich kann ein Tunneln zwischen den Netzwerkknoten verlässlich durchgeführt werden.

[0015] Im Besonderen können die schlechtesten logischen Verbindungen gemäß einer momentanen Ende-zu-Ende-Leistungsfähigkeit des ersten Netzwerks (d.h. des Basisnetzwerks) ausgeschlossen werden.

[0016] Da eine Vielzahl von virtuellen Direktverbindungen (z.B. Tunneln) zwischen den Netzwerkknoten bereitgestellt wird, gibt es eine hohe Redundanz, so dass Verbindungen zwischen Netzwerkknoten auch über andere Netzwerke hergestellt werden können.

[0017] In dem Entscheidungsschritt, im Fall, dass entschieden wird, dass zwischen zwei Netzwerkknoten keine virtuelle Direktverbindung verwendbar ist, kann ein Leitweg zwischen diesen zwei Netzwerkknoten über zumindest einen anderen Netzwerkknoten basierend auf den Ergebnissen der Qualitätsmessungen bestimmt werden.

[0018] Auf diese Weise kann eine so genannte „nächste Etappe“ einfach bestimmt werden, wodurch eine sichere und schnelle Verbindung hergestellt werden kann.

[0019] Die Netzwerkknoten können Tunnelendpunkte sein und die virtuellen Direktverbindungen zwischen jedem Paar der Netzwerkknoten können Tunnel sein, wobei die Tunnel virtuelle Direktverbindungen zwischen den Knoten durch Einkapseln eines Netzwerkprotokolls erster Art in Daten, die von einem Netzwerk zweiter Art getragen werden, bereitstellen.

[0020] Die Qualitätsmessungen können durch jeden Netzwerkknoten mit Bezug auf virtuelle Direktverbindungen zu anderen Netzwerkknoten durchgeführt werden. Somit ist es möglich, Ergebnisse für all die Tunnel zu bekommen, die involviert sind, um eine genaue Entscheidung zu erhalten.

[0021] Die Qualitätsmessungen können eine Messung einer Verzögerungszeit auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen.

[0022] Ein Schwellenwert für eine maximale erlaubbare Verzögerungszeit kann derart gesetzt werden, dass wenn eine Verzögerungszeit auf einer virtuellen Direktverbindung den Schwellenwert überschreitet, die Verbindung als nicht verwendbar bestimmt wird.

[0023] Somit kann eine maximale erlaubbare Verzögerungszeit gesetzt werden. Somit kann eine virtuelle Direktverbindung als nicht verwendbar betrachtet werden, ganz gleich ob die virtuelle Direktverbindung in anderen Aspekten (z.B. bei einer Datenverlustrate) gute Qualität zeigt.

[0024] Außerdem können die Qualitätsmessungen eine Messung einer Datenverlustrate auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen. Die Datenverlustrate kann hier eine Paketverlustrate, z.B. im Falle eines paketvermittelten Netzwerks sein.

[0025] Ein Schwellenwert für eine maximale erlaubbare Datenverlustrate kann derart gesetzt werden, dass wenn eine Datenverlustrate auf einer virtuellen Direktverbindung den Schwellenwert überschreitet, die virtuelle Direktverbindung als nicht verwendbar bestimmt wird.

[0026] Somit kann eine maximale Datenverlustrate (z.B. Paketverlustrate) gesetzt werden. Folglich kann eine

virtuelle Direktverbindung als nicht verwendbar betrachtet werden, ganz gleich ob die virtuelle Direktverbindung in anderen Aspekten (z.B. bei einer Verzögerungszeit) gute Qualität zeigt.

[0027] Die Qualitätsmessung kann sowohl die vorstehende Verzögerungszeitmessung als auch Datenverlustenmessung umfassen. Somit können ein Ergebnis der Verzögerungszeitmessung und ein Ergebnis der Datenverlustenmessung zu einem einzelnen Qualitätsmessergebnis kombiniert werden.

[0028] Auf diese Weise ist eine einfache Handhabung der Messergebnisse möglich, da nur das kombinierte Messergebnis weiter verarbeitet werden muss, und nicht zwei getrennte Werte.

[0029] Das Ergebnis der Verzögerungszeitmessung und das Ergebnis der Datenverlustenmessung können beim Kombinieren von diesen entsprechend gewichtet werden. Somit kann ein Netzwerkbetreiber entscheiden und frei festlegen, was von Verzögerungszeit oder Datenverluste für ihn wichtiger ist.

[0030] Beim Kombinieren können das Verzögerungszeitmessergebnis und das Paketdatenverlustenmessergebnis entsprechend normiert werden. Somit werden beide Ergebnisse in einen dimensionslosen Wert umgewandelt, welcher leicht mit anderen Ergebnissen von anderen virtuellen Direktverbindungen verglichen werden kann.

[0031] Das einzelne Qualitätsmessergebnis kann in einen Ganzzahlenwert umgewandelt werden. Auf diese Weise kann der Verkehr in dem Netzwerk reduziert werden, da Fließkommazahlen mehr Daten erfordern.

[0032] Das Messen und das Entscheiden über die virtuellen Direktverbindungen kann in vorbestimmten Intervallen durchgeführt werden. Das Messen und das Entscheiden über die virtuellen Direktverbindungen kann ebenso durchgeführt werden, wenn es einige Änderungen in den Netzwerkkonfigurationen gibt. Dies reduziert ebenso die Verkehrsmenge und die Berechnungslast, da auf diese Weise eine Messung und Entscheidung nicht kontinuierlich stattfinden. Das Intervall kann frei gesetzt werden, so dass es von der allgemeinen Bedingung des Netzwerks, der Verkehrsmenge und Ähnlichem unabhängig gemacht werden kann.

[0033] Die Ergebnisse der Entscheidung können zu den Netzwerkknoten gesendet werden und die Netzwerkknoten können entsprechend Leitweglenktabellen aktualisieren.

[0034] Somit können die Informationen bezüglich der Entscheidung über die virtuellen Direktverbindungen in die Leitweglenktabellen der Netzwerkknoten übertragen werden.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0035] Die vorliegende Erfindung wird mit Bezug auf die begleitenden Zeichnungen leichter verstanden, wobei:

[0036] **Fig. 1(a)** und **(b)** Unterschiede zwischen dem Tunnel-Broker-Konzept und dem Tunnel-Peers des Konzeptes gemäß einem Ausführungsbeispiel der Erfindung zeigen,

[0037] **Fig. 2** eine dynamische Tunnel-Peering-Architektur gemäß dem Ausführungsbeispiel zeigt,

[0038] **Fig. 3** ein Flussdiagramm des grundlegenden Arbeitsprozedurablaufs gemäß dem Ausführungsbeispiel zeigt,

[0039] **Fig. 4(a)** bis **(d)** eine Normierung, Kombination und Quantisierung eines Leistungsfähigkeitsparameters gemäß dem Ausführungsbeispiel zeigen,

[0040] **Fig. 5(a)** und **(b)** ein Beispiel für ein Ende-zu-Ende-Leistungsfähigkeitmessergebnis einer RTT-(Durchlaufzeit)-Verzögerung und einer Paketverlustrate gemäß dem Ausführungsbeispiel zeigen,

[0041] **Fig. 6** ein Beispiel für den vollständigen gewichteten Graphen zeigt, und

[0042] **Fig. 7** einen optimierten Teilgraphen gemäß dem vorliegenden Ausführungsbeispiel zeigt.

DETAILLIERTE BESCHREIBUNG VON BEVORZUGTEN AUSFÜHRUNGSBEISPIELEN

[0043] Im Folgenden wird ein bevorzugtes Ausführungsbeispiel der Erfindung detaillierter mit Bezug auf die begleitenden Zeichnungen beschrieben.

[0044] Das Verfahren gemäß dem vorliegenden Ausführungsbeispiel wird als eine Lösung für ein automatisches Tunneln von IPv6-über-IPv4 mit einer dynamischen Leistungsfähigkeitsoptimierung vorgeschlagen. In dieser Beschreibung wird diese Prozedur dynamisches Tunnel-Peering mit Leistungsfähigkeitsoptimierung basierend auf einer Ende-zu-Ende-Messung genannt, was als DTP-POEM abgekürzt wird. Außerdem sind die Ausdrücke „virtuelles Netzwerk/Basisnetzwerk“ in der gesamten Beschreibung des ersten Ausführungsbeispiels äquivalent zu IPv6/IPv4. Trotzdem sei angemerkt, dass die Erfindung auch auf andere Arten von Netzwerken (z.B. VPN) und nicht nur auf IPv6-/IPv4-Internet anwendbar ist.

[0045] Im Detail stellt die vorliegende Erfindung ein Schema zum Verbinden von IPv6-Standorten über IPv4-Netzwerke über Tunnel mit einer dynamischen Leistungsfähigkeitsoptimierung bereit. So genannte Peers (oder Tunnel-Peers) agieren als die Endpunkte von IPv6-zu-IPv4-Tunneln und als die Router in dem virtuellen IPv6-Netzwerk. Ob ein Tunnel-Peer die Rolle eines Routers in dem IPv4-Netzwerk spielt, ist für die Erfindung nicht von Bedeutung. Die Peers werden ebenso als Netzwerkknoten bezeichnet.

[0046] Diese Prozedur gemäß der vorliegenden Erfindung ist typischer Weise eine Lösung innerhalb einer Domain für automatisches und dynamisches Tunneln mit Leistungsfähigkeitsoptimierung.

[0047] Die Prozedur gemäß dem Ausführungsbeispiel wird auf die folgende Umgebung angewendet:

- 1) Die vorstehend beschriebenen Peers sind in einer verschiedenartigen Infrastruktur geografisch verstreut, und die Ende-zu-Ende-Pfade zwischen diesen variieren in Leistungsfähigkeit. Des Weiteren ist die Ende-zu-Ende-Leistungsfähigkeit wesentlich durch die Verkehrslastschwankungen in dem Basisnetzwerk beeinflusst.
- 2) Peers, die in dem virtuellen Netzwerk die Rolle von Weiterleitungsknoten spielen, sind meistens Endsyste-me in dem Basisnetzwerk, wobei dann deren Berechnungsbetriebsmittel weder dediziert für ein Tunneln noch für ein Leitweglenken bezeichnet sind.

[0048] Ein dedizierter Server (welcher nachstehend auch als Netzwerkkonfigurationssteuerungselement bezeichnet wird), genannt Tunnel-Arbiter (TA), wird als die Kernkomponente der Architektur definiert, welche eine Entscheidung für ein Tunneln und Leitweglenken trifft, so dass verkehrsreiche Pfade (oder welche mit schlechtem Verhalten) zwischen den Peers von dem Satz von logischen Verbindungen ausgeschlossen werden. All die Tunnelendpunkte stehen in einer Peer-Beziehung, d.h. es gibt keine Server-Klient-Unterscheidung.

[0049] Der allgemeine Aufbau der Tunnel ist in **Fig. 1** gezeigt, in welcher ebenso die Unterschiede zu dem Tunnel-Broker-Konzept dargestellt sind. **Fig. 1(a)** zeigt das Tunnel-Broker-Konzept. Es gibt eine Vielzahl von Tunnel-Klienten (TC), die alle mit einem zentralen Tunnel-Server (TS) verbunden sind. Es gibt keine virtuellen Verbindungen zwischen den TCs, sondern nur Tunnel zwischen jedem TC und dem Tunnel-Server.

[0050] Andererseits, bezugnehmend auf **Fig. 1(b)**, gibt es in dem dynamischen Tunnel-Peering-Modell gemäß dem vorliegenden Ausführungsbeispiel keinen zentralen Punkt für die Verbindungen und die Tunnel-Peers (TP) erzeugen automatisch Tunnel nach Bedarf, wie im Folgenden detailliert beschrieben ist.

[0051] Gemäß dem vorliegenden Ausführungsbeispiel kann ein Tunnel-Peer (TP) die Liste von anderen Peers über die Peer-Registrierungsdatenbank (PRD) des Tunnel-Arbiters bekommen. Die Peers messen dann Leistungsfähigkeitsparameter für die Ende-zu-Ende-Pfade von jedem zu einem anderen und senden die quantisierten und normierten Werte an den Tunnel-Arbiter, der die optimierte Topologie berechnet. Ein Ändern der virtuellen Tunnelverbindungen entsprechend der Berechnung wird periodisch durchgeführt.

[0052] Durch die dynamische Tunnel-Peering-Architektur basierend auf Ende-zu-Ende-Leistungsfähigkeitsmessungen gemäß den vorliegenden Ausführungsbeispielen werden die folgenden Effekte erreicht:

- 1) Ein dynamischer Tunnel-Erzeugungs- und Lösch-Mechanismus wird eingeführt, um die Leistungsfähigkeitsverteilung über dem IPv4-Basisnetzwerk anzupassen.
- 2) Eine Ende-zu-Ende-Leistungsfähigkeit wird in dimensionslose Ganzzahlenwerte quantisiert, so dass der Zustand endlich ist und die Topologie nicht für kleine Störungen anfällig ist.
- 3) Die Tunnel-Entscheidungen, die durch das Kriterium des kürzesten Pfades von allen Pfaden (APSP) im Hinblick auf eine Leistungsfähigkeit einer virtuellen Verbindung bestimmt werden, enthalten ebenso die

Leitweglenkungsinformationen.

[0053] Konzeptionell gibt es zwei Netzwerkebenen in der DTP-POEM-Architektur. Dies ist in [Fig. 2](#) dargestellt. Unten ist das Basisnetzwerk, was gemäß diesem Ausführungsbeispiel das IPv4-Internet ist. Das Basisnetzwerk stellt eine globale Kollektivität mit einer eingesetzten physikalischen Infrastruktur bereit. Jeder Tunnel-Peer (in der Figur durch Bezugszeichen TP angegeben) wird auf einen Knoten in dem Basisnetzwerk abgebildet. Diese Knoten besitzen nur eine IPv4-Kommunikation mit dem Tunnel-Arbitr (TA), wie vorstehend beschrieben.

[0054] Auf der oberen Ebene ist eine virtuelle Netzwerkebene, in welcher all die Verbindungen logisch sind. Die logischen Verbindungen werden durch den Tunnel-Arbitr gesteuert und aktualisiert, wenn sich die Leistungsfähigkeitsbedingung auf der Basisnetzwerkebene ändert. Wie in [Fig. 2](#) gezeigt, besitzen alle Tunnel-Peers eine Verbindung zum globalen IPv6-Internet. Jeder Tunnel-Peer kann mit einem individuellen einzelnen IPv6-Standort verbunden werden oder zwei oder mehr Tunnel-Peers können mit solch einem individuellen IPv6-Standort verbunden werden. Innerhalb eines solchen einzelnen IPv6-Standorts können mehrere IPv6-eigene Router zum Weiterleiten von Datenpaketen bereitgestellt sein. Zusätzlich können die vorstehenden Tunnel-Peers die Rolle von Weiterleitungs-Standorten für die anderen Peers spielen.

[0055] Es sei angemerkt, dass die Tunnel-Peers (TPs) Router in IPv6, aber nicht notwendigerweise auch in IPv4 sind. In [Fig. 2](#) gibt es zwei Ebenen, die entsprechend eine virtuelle IPv6-Netzwerkschicht und die IPv4-Netzwerkschicht, wie vorstehend beschrieben, darstellen. Ein Tunnel-Peer spielt die Rolle eines Routers in der virtuellen Netzwerkschicht (als Ellipse gezeigt), während er in dem IPv4-Netzwerk ein einfacher Host (als Box gezeigt) sein kann. IPv4-Router, die Pakete zwischen diesen „Boxen“ zustellen (d.h. die Knoten in IPv4, die dem TP in IPv6 entsprechen) sind in [Fig. 2](#) nicht gezeichnet, da sie für die Erfindung nicht von Bedeutung sind.

[0056] Im Folgenden wird die Prozedur gemäß dem vorliegenden Ausführungsbeispiel detaillierter mit Bezug auf einen Prozedurablauf des Systems, eine Ende-zu-Ende-Messmethode, eine Tunnelanweisungsführung usw. beschrieben.

1. Grundlegender Prozedurablauf

[0057] Der grundlegende Arbeitsprozedurablauf des dynamischen Tunnel-Peerings basierend auf Ende-zu-Ende-Leistungsfähigkeitsmessungen (DTP-POEM) gemäß dem vorliegenden Ausführungsbeispiel wird im Folgenden durch Bezugnahme auf das in [Fig. 3](#) gezeigte Ablaufdiagramm beschrieben.

a) In Schritt S1 überträgt ein Knoten mit einem Doppelstapel seine Registrierungsinformationen an den TA um einen TP-Bezeichner zu erhalten. Ein Knoten mit einem Doppelstapel ist ein Knoten mit zwei Protokollstapeln und ist dazu in der Lage, ein Basisnetzwerk und ein virtuelles Netzwerk zu bedienen. Das heißt gemäß dem vorliegenden Ausführungsbeispiel besitzt solch ein Knoten Protokollstapel für das IPv4-Internet und das IPv6-Internet.

Als ein Ergebnis der Registrierung wird der Knoten als ein Tunnel-Peer (TP) bezeichnet und bekommt den TP-Bezeichner, welcher ihn eindeutig als ein Tunnel-Peer bezeichnet.

Einen Registrierungseintrag für jedem TP umfasst zumindest: a) einen eindeutigen Bezeichner (d.h. den TP-Bezeichner); b) die IPv4-Adresse des TP; c) eine IPv6-Adresse des TP; d) das IPv6-Adresspräfix, das der TP hält; usw. Das IPv6-Adresspräfix (oder Adressblock) ist ein Satz von zusammenhängenden IPv6-Adressen. Ein Beispiel für solch ein Adresspräfix ist 3ffe:3211::/32.

b) In Schritt S2 behält der TA all die Registrierungsinformationen für die TPs in einer dedizierten Datenbank bei. Die Registrierung enthält zumindest die Adressen der TPs auf dem Basisnetzwerk (BN), d.h. die IPv4-Adressen der TPs.

c) In Schritt S3 bekommt jeder TP die IPv4-Adressen der anderen TPs von dem TA. Dann führt jeder TP einer nach dem anderen Ende-zu-Ende-Messungen in Schritt S4 durch. Das Ergebnis wird in Schritt S5 normiert und quantisiert. Danach wird das quantisierte Ergebnis in Schritt S6 an den TA gesendet.

d) In Schritt S7 erzeugt der TA einen gewichteten vollständigen Graphen mit allen TP als seine Eckpunkte gemäß dem Messergebnis, das durch all die TP gesendet wird. Dann wird in Schritt S8 ein APSP (kürzester Pfad von allen Pfaden)-Algorithmus vorgenommen, um die optimierte virtuelle Topologie zu erhalten.

e) In Schritt S9 sendet der TA Informationen entsprechend der optimierten virtuellen Topologie an die TPs. Das heißt, der TA sendet ein Tunnel-Arbitr-Befehl an die TPs, so dass sie die Tunnelverbindungen untereinander automatisch einstellen, wobei die Leitweglenktabelle in dem TPs in Schritt S10 entsprechend ebenso aktualisiert wird.

Es sei angemerkt, dass die IPv6-Leitweglenktabellen der TPs aktualisiert werden. Der Tunnel-Arbitr gibt

keine Informationen an die IPv4-Router. Und zwar werden nur die virtuellen Verbindungen optimiert, wobei eine Leistungsfähigkeit der IPv6-Router innerhalb des IPv4-Netzwerks für die vorliegende Erfindung nicht von Bedeutung ist.

f) Das System wiederholt die Schritte S2 bis S10 (die Prozesse b)–e)) periodisch oder durch irgendeinen Auslöser, wie ein Hinzufügen eines neuen Knotens zu dem Netzwerk. Ein Beispiel einer angemessenen Aktualisierungsperiode ist 30 Minuten. Das heißt in Schritt S11 wartet der Prozess solch eine vorbestimmte Periode lang und kehrt dann zu Schritt S2 zurück.

Die Systemdetails bezüglich der Messungen und Verarbeitung der Messungen usw. werden in den folgenden vier Unterabschnitten beschrieben.

2. Ende-zu-Ende-Leistungsfähigkeitsmessung

[0058] Ein TP kann eine Anforderung für eine Liste von all den Peers an den TA senden und dann die Ende-zu-Ende-Leistungsfähigkeitsmessung durchführen.

[0059] Eine Ende-zu-Ende-Leistungsfähigkeitsmessmethode ist nicht in dieser Anmeldung enthalten. Der Implementierer kann zum Beispiel dem Dokument „Framework for IP Performance Metrics“, RFC 2330, von V. Paxson, u.a., Mai 1998 folgen. Alternativ können ebenso andere Messverfahren verwendet werden. Die Parameterauswahl hängt von den Netzwerkentwurfskriterien ab. Für allgemeine Zwecke, betrachtet man die Einfachheit der Messoperation, wird eine Durchlaufverzögerung akzeptiert. Solch eine Durchlaufverzögerung wird in „A Round-Trip Delay Metrics for IPPM“, RFC 2681, von G. Almes, S. Kalindini, M. Zekauskas, September 1999 (IPPM bedeutet IP-(Internetprotokoll)-Leistungsfähigkeitsmetrik) beschrieben. Eine Durchlaufverzögerungsmetrik einer P-Art kann durch die ICMP-Echoanforderung/Antwort mit einer dedizierten Paketlänge gemessen werden. Dies kann mit einer bekannten „Ping“-Prozedur vorgenommen werden. Vorzugsweise sollte ein maßgeschneiderter „Ping“-Prozess innerhalb der TP-Programmfolge kodiert werden, anstatt das „Ping“-Werkzeug zu verwenden, das durch das Betriebssystem bereitgestellt ist, um eine optimale Kompatibilität zu der tatsächlich durchgeführten Leistungsfähigkeitsmessung zu haben.

[0060] In diesem Beispiel wird das Ergebnis der Ende-zu-Ende-Messung mit einem UDP-Protokoll („User Datagram Protocol“) an den TA gesendet.

3. Parameternormierung, -kombination und -quantisierung

[0061] Das System muss den Zielkonflikt zwischen der Einfachheit und den Effekten berücksichtigen. Das heißt, es sollte vermieden werden, einen großen Aufwand für das Erreichen der gewünschten Effekte zu haben. Somit, gemäß dem vorliegenden Ausführungsbeispiel, bekommen die TPs die Parameterwerte vorzugsweise auf eine einfache Weise, zum Beispiel durch mehrmaliges Senden eines „Ping“ und durch Messen der mittleren RTT und der Paketverlustrate auf diese Weise. Dann formen Normierungsfunktionen den Verzögerungs- und Paketverlustratenwert in dimensionslose Werte, so dass deren Additionsoperation (d.h. eine geeignete Kombination von Verzögerung und Paketverlustrate) als eine wohl definierte Gewichtungsfunktion physikalischen Fakten entspricht.

[0062] Offensichtlich sollte die Normierungsfunktion für die RTT-Verzögerung linear sein. Diese Erfindung schlägt eine Definition eines Grenzwertes, z.B. 3000 ms (oder andere geeignete Werte) für die Normierungsfunktion vor, der besagt, dass das System feststellen wird, dass eine virtuelle Verbindung unerreichbar ist, wenn die RTT-Verzögerung auf ihr den Schwellenwert überschreitet. Dies ist in **Fig. 4(a)** dargestellt, in welcher die RTT-Verzögerung RTT auf der Abszisse gezeigt ist und der RTT-Verzögerungsleistungsfähigkeitswert d auf der Ordinate gezeigt ist. Dann ergibt sich

$$d(RTT) = \begin{cases} RTT/M & \forall 0 < RTT \leq M \\ 1 & \forall RTT > M \end{cases}$$

wobei M der Schwellenwert für „unerreichbar“ ist.

[0063] Für die Paketverlustrate (definiert als die Anzahl von verlorenen Paketen/Anzahl von allen übertragenen Paketen, üblicherweise in % angegeben) ist der Fall unterschiedlich. Zum Beispiel wird angenommen, dass es drei Tunnel-Peers TP1, TP2, und TP3 gibt. Wenn die Paketverlustrate PLR von TP1 zu TP2 x ist, während die von TP2 zu TP3 y ist, dann sollte die Paketverlustrate von TP1 zu TP3 über TP2 $1 - (1 - x)(1 - y)$ sein. Dann ist gemäß dem vorliegenden Ausführungsbeispiel eine Funktion r(PLR) für eine Paketverlustratennormierung wie folgt:

$$r(PLR) = \begin{cases} \log_{1-p}(1-PLR) & \forall 0 \leq PLR < p \\ 1 & \forall p \leq PLR < 1 \end{cases}$$

wobei p ($0 < p < 1$) der Paketverlustratenschwellenwert für „unerreichbar“ ist. Dies ist in **Fig. 4(b)** dargestellt, in welcher die Paketverlustrate PLR auf der Abszisse gezeigt ist und der Paketverlustratenleistungsfähigkeitswert r auf der Ordinate gezeigt ist.

[0064] Die Kombinationsfunktion stellt dann adaptive Gewichtungen bereit, die die Verzögerungs- und Paketverlustratenwerte zu einem einzelnen machen. Es ist erforderlich, dass der kombinierte Wert zu einem des Verzögerungs- oder Paketverlustratenwerts linear ist, wenn der andere Null ist. Des Weiteren sollte die Kombinationsfunktion noch die „Unerreichbarkeit“ behalten. Somit nimmt die Erfindung die folgende Funktion an, um die Rolle einer Kombination zu spielen, d.h.

$$u(d, r) = \begin{cases} qd + (1-q)r, & \forall 0 \leq d, r < 1; \\ 1, & \text{if } \max\{d, r\} = 1. \end{cases}$$

wobei die Konstante $0 < q < 1$ die relative Wichtigkeit der Verzögerung gegenüber der Paketverlustrate ist. Allgemein ist $q = 1/2$ (**Fig. 4(c)**).

[0065] Schließlich wäre es besser, die Parameterwerte in kleine Ganzzahlen zu quantisieren, so dass der Übertragungs-overhead zwischen der TA und den TPs so gering wie möglich ist. Die APSP-Berechnung auf einem Ganzzahl-gewichteten vollständigen Graphen wäre ebenso viel schneller, als die auf einem Fließkommazahl-gewichteten. Wichtiger ist, dass eine Quantisierung ein häufiges Aktualisieren von dynamischen Leistungsfähigkeitsstatusinformationen vermeidet. Besonders der „unerreichbar“ wird zu einem gesättigten Wert quantisiert, was ein „infinites“ Gewicht bedeutet. Das ist in **Fig. 4(d)** und in der folgenden Formel dargestellt, in welcher der Ganzzahlenwert v aus dem kombinierten Leistungsfähigkeitswert u erzeugt wird.

$$v = \begin{cases} 1 & \forall 0 \leq u < 0.1 \\ 2 & \forall 0.1 \leq u < 0.2 \\ 3 & \forall 0.2 \leq u < 0.3 \\ \dots & \\ 10 & \forall 0.9 \leq u < 1 \\ 255 & \forall u = 1 \end{cases}$$

[0066] Es sei angemerkt, dass hier Werte 1 bis 10 mit den Fließkommawerten von u schrittweise verknüpft sind, wohingegen für „unerreichbar“ oder unverwendbar 255 verknüpft wird.

4. Lösen des APSP-Problems

[0067] Durch Verwendung der quantisierten und normierten Leistungsfähigkeitswerte v , die auf die vorstehend beschriebene Weise bestimmt werden, kann durch den TA ein gewichteter vollständiger Graph erzeugt werden. Die Werte können ebenso verwendet werden, um eine entsprechende gewichtete Adjazenzmatrix zu erzeugen. In solch einer gewichteten Adjazenzmatrix definiert ein Element $a(i, j)$ den Leistungsfähigkeitswert v zwischen einem Tunnelpunkt TP_i und einem Tunnelpunkt TP_j , wobei der Leistungsfähigkeitswert ebenso auf diese Weise gerichtet ist, d.h. von TP_i zu TP_j . In der Matrix bezeichnet der Wert i die Spalte der Matrix und j bezeichnet die Reihe, wobei i und j Ganzzahlen sind.

[0068] Solange der gewichtete vollständige Graph durch den TA erzeugt wurde, kann jeder APSP-Algorithmus angewendet werden, um den optimierten Teilgraph zu berechnen. Der ursprünglich erzeugte vollständige Graph ist gerichtet. Die Messungen werden in einem Durchlaufweg vorgenommen und dementsprechend sollte die gewichtete Adjazenzmatrix für den Graphen mit Bezug auf seine Diagonale symmetrisch sein. Manchmal jedoch kann die Matrix in der Praxis aufgrund der Messfehler und der asymmetrischen dynamischen Bedingungen zwischen den zwei Enden eines Paares tatsächlich asymmetrisch sein. Das heißt, $a(i, j)$ kann ungleich zu $a(j, i)$ sein. Deshalb, angenommen A ist die ursprünglich gemessene Adjazenzmatrix, definieren wir eine

neue Adjazenzmatrix W , so dass

$$w(i, j) = w(j, i) = a(i, j) + a(j, i).$$

[0069] Die Matrix W ist definitiv symmetrisch und die Berechnung würde auf ihr anstatt auf A basieren.

[0070] Man kann einen Floyd-Warshall-APSP-Algorithmus verwenden, um das vorstehend mit der Matrix beschriebene APSP-Problem zu lösen. Der Floyd-Warshall-Algorithmus ist durch E. Minieka in „Optimization Algorithms for Networks and Graphs“, Marcel Dekke, Inc. 1978, ISBN 0-8247-6642-3 beschrieben. Der Algorithmus kann wie folgt in C-Sprache kodiert sein:

Alg.1: Floyd-Warshall-APSP-Algorithmus:

N: die Anzahl von Eckpunkten
 W: die Adjazenzmatrix des gewichteten vollständigen Graphen, mit den Gewichtungen initialisiert
 P: die Vorgängermatrix für den optimierten Teilgraphen, für alle Elemente mit -1 initialisiert.

```
=====
void FloydWarshall(int N, int *W, int *P)
{
    int i, j, k;
    for (k=0; k < N; k++)
        for (i=0; i<N; i++)
            for (j=0; j<N; j++) {
                if (W[i*N+j] > W[i*N+k] + W[k*N+j]) {
                    W[i*N+j] = W[i*N+k] + W[k*N+j];
                    P[i*N+j] = k;
                }
            }
    /* da die Matrix W symmetrisch ist, kann der
    Anfangswert von j auf i gesetzt werden.*/
} /*FloydWarshall*/
```

=====

[0071] Es sei angemerkt, dass W und P als eindimensionale Array-Felder behandelt werden, so dass alle N (Anzahl von Eckpunkten) Reihen der entsprechenden Matrizen in eine Reihe umgeschrieben werden.

[0072] Nachdem der Algorithmus ausgeführt wurde, ist die Vorgängermatrix P zum Bestimmen des nächsten Etappen-Leitwegs für jeden Eckpunkt ausreichend und deshalb kann der TA Leitweglenkinformationen zu den TPs zusammen mit den Tunnelbefehlen bereitstellen. Ein einfacher Algorithmus, der die nächste Etappe von einer Quelle zu einem Ziel berechnet, ist durch die Erfindung wie nachstehend bezeichnet. Wenn der TA die Informationen über virtuelle Netzwerkblöcke, die mit jedem TP verknüpft sind, beibehält, dann kann ein Nächste-Etappe-Verfahren („NextHop method“), das im Folgenden beschrieben wird, angewendet werden, um die Leitweglenktabelle dynamisch zu erzeugen.

Alg.2: Erzeugen der Leitweglenktabelle mit der P-Matrix:

N: die Anzahl von Eckpunkten
 P: die Vorgängermatrix für den optimierten Teilgraph
 u, v: die Bezeichner der Quelle und des Zieles

```

=====
int NextHop(int N, int *P, int u, int v)
{
    int k,r;
    k = P[u*N+v]; r = u;
    while (k != -1) {
        r = k;
        k = P[u*N+k];
    }
    return r;
} /* NextHop */
=====

```

[0073] Es ist klar, dass das Nächste-Etappe-Verfahren für einen bestimmten Quellknoten keine Reihen für andere Peers mit einbezieht. Deshalb kann der Algorithmus an jedem TP separat ausgeführt werden.

5. Vornehmen der Tunnelentscheidung

[0074] Ein Lösen des APSP-Problems ergibt die optimierte Vorgängermatrix P. Gemäß dieser Matrix kann der TA einfach die virtuelle Netzwerktopologie durch Entfernen dieser virtuellen Verbindungen erhalten, deren entsprechende Werte in der Matrix P positiv sind. Das heißt, ein unveränderter Wert -1 repräsentiert eine Tunnelverbindung des virtuellen Netzwerks, während jeder positive Wert ein Vorwärtsweiterleiten angibt. Diese Matrix kann als die globale Tunnelentscheidung angesehen werden, und die k-te Reihe von P ist die Entscheidung für das k-te TP. Dann sendet der TA entsprechend Entscheidungen an die TPs. Eine Entscheidung enthält sowohl Tunnel- als auch Leitweglenkinformationen. Die letztgenannten können mit einem Nächste-Etappe-Prozess in eine Leitweglenktabelle, wie vorstehend erwähnt, dekodiert werden.

[0075] Im Folgenden wird ein Beispiel durch Bezugnahme auf die [Fig. 3](#) und 5 bis [Fig. 7](#) gegeben.

[0076] Es wird angenommen, dass es sechs IPv6-Standorte gibt, welche über das DTP-POEM-System gemäß dem Ausführungsbeispiel verbunden werden.

a) Registrierung

[0077] Ein Endpunkt eines jeden Standorts sendet seine Registrierung an den Tunnel-Arbitr, wobei er seine eigene Tunnel-Peer-ID und eine Liste von allen Peers erhält (Schritte S1 bis S3 in [Fig. 3](#)).

b) Messung

[0078] Jeder TP nimmt eine Messung einer Ende-zu-Ende-Leistungsfähigkeit vor (Schritt S4 in [Fig. 3](#)), was einen vollständigen Graph mit Leistungsfähigkeitsparametern auf den Kanten ergibt, wenn die Messungen von allen TPs berücksichtigt werden. Das Ergebnis ist in [Fig. 5\(a\)](#) bezüglich der RTT-Verzögerung gezeigt und in [Fig. 5\(b\)](#) bezüglich der Paketverlustrate (PLR).

c) Normierung und Quantisierung

[0079] Jeder TP normiert Leistungsfähigkeitswerte, kombiniert Verzögerungs- und Paketverlustrate und quantisiert dann die dimensionslosen Werte zu einer kleinen Ganzzahl, bevor diese an den TA gesendet werden (Schritte S5 und S6 in [Fig. 3](#)). Als ein Ergebnis konstruiert der TA einen gewichteten vollständigen Graphen (Schritt S7 in [Fig. 3](#)).

[0080] Das Ergebnis ist in [Fig. 6](#) gezeigt, wobei die Adjazenzmatrix des gewichteten vollständigen Graphen wie folgt lautet:

W	0	1	2	3	4	5
0	0	1	3	6	255	1
1	1	0	3	1	5	1
2	3	3	0	255	1	1
3	6	1	255	0	1	3
4	255	5	1	1	0	7
5	1	1	1	3	7	0

[0081] Zum Beispiel zeigt der Tunnel zwischen TP0 und TP4 eine RTT-Verzögerungszeit von 3950 ms. Sie ist somit höher als der Schwellenwert M von 3000 ms. Folglich wird dieser Tunnel als nicht verwendbar bestimmt, d.h. TP ist für TP0 über einen direkten Tunnel unerreichbar. Deshalb ist der entsprechende Eintrag in der vorstehenden Adjazenzmatrix 255 ($w(0, 4)$).

[0082] Als ein anderes Beispiel ist die Paketverlustrate PLR zwischen TP2 und TP3 29%. Sie ist somit höher als der Schwellenwert p, welcher auf 20% gesetzt werden kann. Folglich wird auch dieser Tunnel als nicht verwendbar bestimmt, und deshalb ist der entsprechende Eintrag n in der vorstehenden Adjazenzmatrix 255 ($w(2, 3)$).

d) Berechnung

[0083] Der TA berechnet das APSP-Problem, um die Optimierung zu erhalten. Das Ergebnis wird durch eine Vorgängermatrix (wie vorstehend beschrieben) dargestellt und jede Reihe in der Matrix ist die Entscheidung für das entsprechende TP. Die Vorgängermatrix lautet wie folgt:

P	0	1	2	3	4	5
0	-1	-1	5	1	3	-1
1	-1	-1	5	-1	3	-1
2	5	5	-1	4	-1	-1
3	1	-1	4	-1	-1	1
4	3	3	-1	-1	-1	2
5	-1	-1	-1	1	2	-1

e) Ausführung

[0084] Nach einem Erhalten der Tunnel- und Leitweglenkentscheidung von dem TA, aktualisiert ein TP seine Tunnelschnittstellenkonfiguration und modifiziert dann die Leitweglenktabelle mit dem Alg.2 (d.h. der im vorstehenden beschriebenen Nächste-Etappe-Routine). Die hervorgehobenen Einträge in der Vorgängermatrix (d.h. die erste Reihe der Vorgängermatrix) bezieht sich auf die Tunnel- und Leitweglenkinformationen für TP0. Diese Entscheidung wird in die Leitweglenktabelle von TP0 übertragen, wie im Folgenden gezeigt:

	Tunnelentscheidung					
	0	1	2	3	4	5
Peer-Tunneln?	-	Y	N	N	N	Y

	Leitweglenktabelle für TP0					
Ziel	0	1	2	3	4	5
Nächste Etappe	0	1	5	1	1	5

[0085] Der entsprechende optimierte Teilgraph ist in [Fig. 7](#) gezeigt. In diesem Beispiel werden direkte Verbindungen (d.h. Tunnel) nur zwischen TP0 und TP1, TP0 und TP5, TP5 und TP1, TP1 und TP3, TP2 und TP5, TP4 und TP2 und zwischen TP3 und TP4 bereitgestellt.

[0086] Somit, wenn TP0 betrachtet wird (erster Eintrag, d.h. erste Reihe oder erste Spalte in der vorstehenden Vorgängermatrix), wird ein direktes Tunneln nur zu TP1 und TP5 durchgeführt. Ein Tunneln zu TP2 wird über TP5 als die nächste Etappe durchgeführt (Eintrag in der Vorgängermatrix in Spalte 2 entsprechend TP2). Ein Tunneln zu TP3 wird über TP1 als die nächste Etappe durchgeführt. Ein Tunneln zu TP4 wird über TP3 durchgeführt, d.h. die nächste Etappe ist TP1 und dann TP3.

[0087] Wenn eine neue Periode kommt, überprüfen die TPs die Peerliste und starten eine neue Runde von Messaktivitäten.

[0088] Um die Erfindung zu implementieren, sollten viele Parameter und Methoden wie vorstehend erwähnt im Voraus ausgehandelt werden. Vorzugsweise sollte ein Protokolldokument bearbeitet werden, das Datenformate und gemeinsame Regeln definiert, die der TA und alle TPs befolgen sollten.

[0089] Mit Bezug auf die Komplexität der Berechnung ist es vorteilhaft, eine Fließkommazahlberechnung so weit wie möglich zu vermeiden. Gemäß der vorliegenden Erfindung werden die Leistungsfähigkeitswerte als Ganzzahlen übertragen. Zusätzlich kann ein Tabellenabstastverfahren zum Berechnen der logarithmischen Werte angewendet werden, um die Berechnungslast weiter zu reduzieren.

[0090] Außerdem, um den Overhead-Verkehr zu verringern, wird gemäß dem vorliegenden Ausführungsbeispiel ein „keep-alive“-Ansatz bzw. ein „Am-Leben-Erhalten“-Ansatz verwendet. Das heißt, wenn die Entscheidung für einen bestimmten TP nicht geändert werden muss, wird eine einfache Am-Leben-Erhalten-Nachricht anstelle einer ganzen Entscheidung an den TP gesendet. Eine weitere Maßnahme, um den Verkehr von Leistungsfähigkeitswerten zu minimieren, ist ein periodischer Aktualisierungs-/Am-Leben-Erhalten-Mechanismus. Das heißt, gemäß dem vorliegenden Ausführungsbeispiel wird zwischen zwei Entscheidungen eine vorbestimmte Zeitdauer lang gewartet. Wie vorstehend beschrieben, kann die vorbestimmte Dauer zum Beispiel 30 Minuten sein.

[0091] TPs sind verschiedenartig und arbeiten in einer Peer-Betriebsart. Jedoch kann als eine Alternative, um die Verlässlichkeit zu ermöglichen, ein Einführen eines zuverlässigen Tunnel-Servers die Robustheit des gesamten Systems verbessern. Vorzugsweise wird der Tunnel-Server unter den Tunnel-Peers ausgewählt, statt durch den Tunnel-Arbiträr permanent dediziert zu werden. Auf jeden Fall sind eine stabile hohe Leistungsfähigkeit beim Berechnen, eine hohe Geschwindigkeit bei einer globalen Verbindung und immer eingeschaltete TPs wünschenswert.

[0092] Der TA ist ein dedizierter Server. Vorzugsweise wird ein relationales Datenbanksystem verwendet, um die TP-Registrierung und Tunnel-Zustände beizubehalten. Auf dem TA kann ebenso ein HTTP-(Hyper-Text-Transfer Protocol)-Dämon laufen, so dass sich jeder Benutzer einfach in dem DTP-POEM-System registrieren kann und die laufende Topologie als auch die gesamte Leistungsfähigkeitsstufe visualisiert werden kann.

[0093] Somit präsentiert gemäß der Erfindung das Tunnel-Konzept nicht nur einen Weg für Konnektivität, sondern auch einen Weg um sowohl eine dynamische virtuelle Topologie aufzubauen als auch die entspre-

chenden Leitweglenkungstabellen, die bessere Dienste bereitstellen als die Hinterlassenschaft „best-effort“, wobei durch das virtuelle Netzwerk frei von den verkehrsreichsten Pfaden im IPv4-Internet gelenkt wird.

[0094] Wie vorstehend beschrieben stellt die Erfindung eine Lösung innerhalb einer Domain für automatisches Tunneln mit Leistungsfähigkeitsoptimierung bereit. IPv6-Netzwerkstandorte können über virtuelle Tunnelverbindungen verbunden werden, wenn die globale IPv6-Infrastruktur nicht eingesetzt wurde.

[0095] Frühere Lösungen, wie 6-zu-4 und Tunnel-Broker stellen nur eine virtuelle Netzwerkkonnektivität ohne Leistungsfähigkeitsbetrachtungen bereit. Diese Erfindung stellt einen Weg bereit, um eine virtuelle Topologie auf eine bessere Weise zu erstellen, so dass die verkehrsreichsten oder am meisten verzögernsten Ende-zu-Ende-Pfade nicht in den Satz der virtuellen Verbindungen ausgewählt werden. Außerdem ist die Lösung dynamisch anpassungsfähig, d.h. passt sich der Leistungsfähigkeitsvariation auf dem Basisnetzwerk an und stellt von Augenblick zu Augenblick eine beste virtuelle Topologie bereit. Eine Topologieoptimierung basierend auf einer quantisierten Ende-zu-Ende-Verhaltensmessung ist besonders vorteilhaft.

[0096] Somit, gemäß der Erfindung, nimmt der Tunnel-Arbitrer (TA, ein Beispiel für ein Netzwerkkonfigurationssteuerungselement) die Daten über die Ende-zu-Ende-Leistungsfähigkeit und trifft Entscheidungen, ob ein Tunnel zwischen einem bestimmten Paar von Tunnel-Peers erzeugt werden sollte, und wie jeder Tunnel-Peer seine Leitweglenktafel setzt (d.h. die IPv6-Leitweglenktafel).

[0097] Die durch die Erfindung erreichten Vorteile sind wie folgt:

[0098] Die schlechtesten logischen Verbindungen für virtuelle Netzwerke werden gemäß der momentanen Ende-zu-Ende-Leistungsfähigkeit aus dem Basisnetzwerk ausgeschlossen.

[0099] Bei der gegenseitigen Konnektivität der Peers wird eine Redundanz bereitgestellt, ohne einen verwundbaren zentralen Punkt in dem Tunnel-Server, wie bei der Tunnel-Broker-Architektur. Des Weiteren wird die gleiche Redundanz zu der Konnektivität der Peers zum globalen IPv6-Internet bereitgestellt, solange mehr als ein Peer universell angeschlossen wurde (**Fig. 1**).

[0100] Obwohl die Tunnel-Peers zu jedem anderen tunneln können, tun sie dies nur, wenn es notwendig ist, d.h. „Tunneln nach Bedarf“. Wenn die existierenden virtuellen Verbindungen einen Pfad für zwei Knoten bereitgestellt haben und die allgemeine Leistungsfähigkeit des Pfades besser ist, als die der direkten virtuellen Verbindung zwischen diesen, wenn es eine solche gibt, dann wird die virtuelle Direktverbindung nicht in die virtuelle Topologie aufgenommen.

[0101] Eine zentralisierte Berechnung, die durch die Tunnel-Arbitrer-Komponente durchgeführt wird, nimmt eine globale Optimierung in Echtzeit vor, wobei die Topologie eingestellt wird, um den zeitlichen Basisnetzwerkleistungsfähigkeitsvariationen zu entsprechen.

[0102] Es ist nicht notwendig, für diesen Ansatz einen speziellen Adressraum zu definieren, wie bei der 6-zu-4-Architektur.

[0103] Im schlimmsten Fall kann die Architektur, die durch diese Erfindung präsentiert wird, eine Topologie mit angemessener Konnektivität erzeugen.

[0104] Die vorstehende Beschreibung und begleitenden Zeichnungen stellen die vorliegende Erfindung nur anhand eines Beispiels dar. Somit kann das Ausführungsbeispiel innerhalb dem Umfang der beigefügten Ansprüche variieren.

[0105] Zum Beispiel wird gemäß dem vorstehend erwähnten Ausführungsbeispiel das Verfahren auf ein IPv6-Internet über ein IPv4-Internet angewendet. Die Erfindung kann jedoch ebenso in anderen virtuellen Verbindungsnetzwerken wie etwa VPN (virtuelles privates Netzwerk), IP RAN (Internet-Protokollfunkzugriffsnetzwerk), All-IP, usw. verwendet werden.

[0106] Außerdem wird für das APSP-Problem gemäß dem vorliegenden Ausführungsbeispiel der Floyd-Warshall-Algorithmus verwendet, weil er einfach und am verständlichsten ist. Trotzdem können als Alternativen andere Algorithmen verwendet werden, wie zum Beispiel in U. Zwick, „All Pairs Shortest Paths Using Bridging Sets and Rectangular Matrix Multiplication“, August 2000 beschrieben.

[0107] Des Weiteren ist gemäß dem Ausführungsbeispiel der Tunnel-Arbiter (TA, das Netzwerkkonfigurationssteuerungselement) ein separates Netzwerkelement. Die Funktion des Tunnel-Arbiter kann jedoch in anderen Netzwerkelementen eingebettet sein.

[0108] Außerdem, um einen Verlust von Paketen während eines Änderns der Tunnel zu vermeiden oder zu minimieren, kann eine drahtlose Umschalttechnik angewendet werden, und die Quantisierung hält die Topologie stabil und robust.

[0109] Außerdem sei angemerkt, dass das vorstehende Ausführungsbeispiel in einem Fall beschrieben wurde, bei dem zwei unterschiedliche Netzwerkarten betroffen sind, und zwar IPv6 und IPv4. Es ist jedoch ebenso möglich, virtuelle Direktverbindungen (Tunnel) in der gleichen Netzwerkart, z.B. IPv4-Tunnel über ein IPv4-Netzwerk bereitzustellen.

Patentansprüche

1. Verfahren zum Konfigurieren von Verbindungen zwischen einer Vielzahl von Netzwerkknoten (TP0 bis TP5), wobei jedes Paar des Netzwerkknotens über virtuelle Direktverbindungen verbunden ist, wobei die Netzwerkknoten Tunnelendpunkte sind und die virtuelle Direktverbindung zwischen jedem Paar der Netzwerkknoten (TP0 bis TP5) Tunnel sind, wobei die Tunnel virtuelle Direktverbindungen zwischen den Knoten durch Einkapseln eines Netzwerkprotokolls erster Art in Daten, die von einem Netzwerk zweiter Art getragen werden, bereitstellen, wobei das Verfahren die Schritte aufweist:

Durchführen von Qualitätsmessungen der virtuellen Direktverbindungen zwischen jedem Paar der Netzwerkknoten (S4), und

Entscheiden, basierend auf den Ergebnissen der Qualitätsmessungen, ob eine virtuelle Direktverbindung zum Transportieren von Daten zu verwenden ist oder nicht (S7 bis S10), wobei die Ergebnisse der Qualitätsmessungen an ein Netzwerkkonfigurationssteuerungselement (TA) übertragen werden, welches den Entscheidungsschritt durchführt.

2. Verfahren gemäß Anspruch 1, wobei in dem Entscheidungsschritt, im Fall, dass entschieden wird, dass zwischen zwei Netzwerkknoten (TP0, TP3) keine virtuelle Direktverbindung verwendbar ist, ein Leitweg zwischen diesen zwei Netzwerkknoten über zumindest einen anderen Netzwerkknoten (TP1) basierend auf den Ergebnissen der Qualitätsmessungen bestimmt wird.

3. Verfahren gemäß Anspruch 1, wobei die Qualitätsmessungen durch jeden Netzwerkknoten (TP) mit Bezug auf die virtuellen Direktverbindungen zu anderen Netzwerkknoten (TP0 bis TP5) durchgeführt werden.

4. Verfahren gemäß Anspruch 1, wobei die Qualitätsmessungen eine Messung einer Verzögerungszeit auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen.

5. Verfahren gemäß Anspruch 4, wobei ein Schwellenwert (M) für eine maximal erlaubbare Verzögerungszeit gesetzt wird, und wenn eine Verzögerungszeit auf einer virtuellen Direktverbindung den Schwellenwert übersteigt, die Verbindung als nicht verwendbar bestimmt wird.

6. Verfahren gemäß Anspruch 1, wobei die Qualitätsmessungen eine Messung einer Datenverlustrate auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen.

7. Verfahren gemäß Anspruch 6, wobei ein Schwellenwert für eine maximal erlaubbare Datenverlustrate (p) gesetzt wird, und wenn eine Datenverlustrate auf einer virtuellen Direktverbindung den Schwellenwert übersteigt, die Verbindung als nicht verwendbar bestimmt wird.

8. Verfahren gemäß Anspruch 6, wobei die Qualitätsmessungen weiter eine Messung einer Verzögerungszeit auf einer virtuellen Direktverbindung zwischen den zwei Netzwerkknoten umfassen und ein Ergebnis der Verzögerungszeitmessung und ein Ergebnis der Datenverlustratenmessung zu einem einzelnen Qualitätsmessergebnis kombiniert werden.

9. Verfahren gemäß Anspruch 8, wobei das Ergebnis der Verzögerungszeitmessung und das Ergebnis der Datenverlustratenmessung beim Kombinieren von diesen entsprechend gewichtet werden.

10. Verfahren gemäß Anspruch 8, wobei beim Kombinieren das Verzögerungszeitmessergebnis und das Paketdatenverlustratenmessergebnis entsprechend normiert werden.

11. Verfahren gemäß Anspruch 8, wobei ein Schwellenwert (M) für eine maximal erlaubbare Verzögerungszeit gesetzt wird, und wenn eine Verzögerungszeit auf einer Verbindung den Schwellenwert übersteigt, die Verbindung als nicht verwendbar bestimmt wird.

12. Verfahren gemäß Anspruch 8, wobei das einzelne Qualitätsmessergebnis in einen Ganzzahlenwert umgewandelt wird.

13. Verfahren gemäß Anspruch 1, wobei der Mess- und der Entscheidungsschritt bei jedem vorbestimmten Intervall durchgeführt werden.

14. Verfahren gemäß Anspruch 1, wobei der Mess- und der Entscheidungsschritt nach einem Veranlassen durch einen Betreiber des Netzwerks oder nach einem Ändern der Netzwerkkonfiguration durchgeführt werden.

15. Verfahren gemäß Anspruch 1, wobei die Ergebnisse des Entscheidungsschritts an die Netzwerkknoten gesendet werden und die Netzwerkknoten entsprechend Leitweglenktabellen aktualisieren.

16. Netzwerksystem mit einer Vielzahl von Netzwerkknoten (TP0 bis TP5) und einem Netzwerkkonfigurationssteuerungselement (TA), wobei jedes Paar des Netzwerkknotens über virtuelle Direktverbindungen verbunden ist, wobei die virtuelle Direktverbindung zwischen jedem Paar der Netzwerkknoten (TP0 bis TP5) Tunnel sind, wobei die Tunnel virtuelle Direktverbindungen zwischen den Knoten durch Einkapseln eines Netzwerkprotokolls erster Art in Daten, die von einem Netzwerk zweiter Art getragen werden, bereitstellen, wobei die Netzwerkknoten angepasst sind, um Qualitätsmessungen der virtuellen Direktverbindungen durchzuführen und um Ergebnisse der Qualitätsmessungen an das Netzwerkkonfigurationssteuerungselement zu senden, und das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um basierend auf den Ergebnissen der Qualitätsmessungen zu entscheiden, ob eine virtuelle Direktverbindung zum Transportieren von Daten zu verwenden ist oder nicht.

17. Netzwerksystem gemäß Anspruch 16, wobei das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um im Fall, dass es entschieden hat, dass zwischen zwei Netzwerkknoten (TP0, TP3) keine virtuelle Direktverbindung verwendbar ist, einen Leitweg zwischen diesen zwei Netzwerkknoten über zumindest einen anderen Netzwerkknoten (TP1) basierend auf den Ergebnissen der Qualitätsmessungen zu bestimmen.

18. System gemäß Anspruch 16, wobei die Qualitätsmessungen eine Messung einer Verzögerungszeit auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen.

19. System gemäß Anspruch 18, wobei ein Schwellenwert (M) für eine maximal erlaubbare Verzögerungszeit gesetzt ist und das Netzwerkkonfigurationssteuerungselement (TA) und/oder jeder Netzwerkknoten (TP) angepasst ist/sind, um die virtuelle Direktverbindung als nicht verwendbar zu bestimmen, wenn eine Verzögerungszeit auf einer virtuellen Direktverbindung den Schwellenwert übersteigt.

20. System gemäß Anspruch 16, wobei die Qualitätsmessungen eine Messung einer Datenverlustrate auf einer virtuellen Direktverbindung zwischen zwei Netzwerkknoten umfassen.

21. System gemäß Anspruch 20, wobei ein Schwellenwert für eine maximal erlaubbare Datenverlustrate (p) gesetzt ist und das Netzwerkkonfigurationssteuerungselement (TA) und/oder jeder Netzwerkknoten (TP) angepasst ist/sind, um die virtuelle Direktverbindung als nicht verwendbar zu bestimmen, wenn eine Datenverlustrate auf einer virtuellen Direktverbindung den Schwellenwert übersteigt.

22. System gemäß Anspruch 20, wobei die Qualitätsmessungen weiter eine Messung einer Verzögerungszeit auf einem Tunnel zwischen den zwei Netzwerkknoten umfassen und jeder Netzwerkknoten angepasst ist, um ein Ergebnis der Verzögerungszeitmessung und ein Ergebnis der Datenverlustratenmessung zu einem einzelnen Qualitätsmessergebnis zu kombinieren.

23. System gemäß Anspruch 22, wobei jeder Netzwerkknoten (TP) angepasst ist, das Ergebnis der Verzögerungszeitmessung und das Ergebnis der Datenverlustratenmessung beim Kombinieren von diesen entsprechend zu gewichten.

24. System gemäß Anspruch 22, wobei jeder Netzwerkknoten (TP) angepasst ist, um das Verzögerungszeitmessergebnis und das Paketdatenverlustratenmessergebnis entsprechend zu normieren.

25. System gemäß Anspruch 22, wobei ein Schwellenwert (M) für eine maximal erlaubbare Verzögerungszeit gesetzt ist und das Netzwerkkonfigurationssteuerungselement (TA) und/oder jeder Netzwerkknoten (TP) angepasst ist/sind, um die virtuellen Direktverbindung als nicht verwendbar zu bestimmen, wenn eine Verzögerungszeit auf einer Verbindung den Schwellenwert übersteigt.

26. System gemäß Anspruch 22, wobei jeder Netzwerkknoten angepasst ist, um das einzelne Qualitätsmessergebnis in einen Ganzzahlenwert umzuwandeln.

27. System gemäß Anspruch 16, wobei das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um bei jedem vorbestimmten Intervall die Qualitätsmessung anzufordern und die Messung anzufordern und die Entscheidung durchzuführen.

28. System gemäß Anspruch 16, wobei das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um nach einem Veranlassen durch einen Betreiber der Netzwerke oder nach einem Ändern der Netzwerkkonfiguration die Qualitätsmessung anzufordern und die Messung anzufordern und die Entscheidung durchzuführen.

29. System gemäß Anspruch 16, wobei das Netzwerkkonfigurationssteuerungselement angepasst ist, um die Ergebnisse der Entscheidung an die Netzwerkknoten zu senden und die Netzwerkknoten angepasst sind, um Leitweglenktabellen entsprechend zu aktualisieren.

30. Netzwerkknoten, wobei der Netzwerkknoten (TP0) über virtuelle Direktverbindungen mit anderen Netzwerkknoten (TP1 bis TP5) verbunden ist, und die virtuellen Direktverbindungen Tunnel sind, wobei die Tunnel virtuelle Direktverbindungen durch Einkapseln eines Netzwerkprotokolls erster Art in Daten, die von einem Netzwerk zweiter Art getragen werden, bereitstellen, wobei der Netzwerkknoten (TP0 bis TP5) angepasst ist, um Qualitätsmessungen der virtuellen Direktverbindungen durchzuführen und um Ergebnisse der Qualitätsmessungen an das Netzwerkkonfigurationssteuerungselement zu senden.

31. Netzwerkknoten gemäß Anspruch 30, wobei die Qualitätsmessungen eine Messung einer Verzögerungszeit auf einer virtuellen Direktverbindung zu einem anderen der anderen Netzwerkknoten umfassen.

32. Netzwerkknoten gemäß Anspruch 30, wobei die Qualitätsmessungen eine Messung einer Datenverlustrate auf einer virtuellen Direktverbindung zu einem anderen der anderen Netzwerkknoten umfassen.

33. Netzwerkkonfigurationssteuerungselement zum Konfigurieren eines Netzwerks mit einer Vielzahl von Netzwerkknoten (TP0 bis TP5), wobei jedes Paar des Netzwerkknotens über virtuelle Direktverbindungen verbunden ist, wobei die virtuelle Direktverbindung zwischen jedem Paar der Netzwerkknoten (TP0 bis TP5) Tunnel sind, wobei die Tunnel virtuelle Direktverbindungen zwischen den Knoten durch Einkapseln eines Netzwerkprotokolls erster Art in Daten, die von einem Netzwerk zweiter Art getragen werden, bereitstellen, wobei das Netzwerkkonfigurationssteuerungselement angepasst ist, um basierend auf Ergebnissen von durch die Netzwerkknoten durchgeführten Qualitätsmessungen zu entscheiden, ob eine virtuelle Direktverbindung zum Transportieren von Daten zu verwenden ist oder nicht.

34. Netzwerkkonfigurationssteuerungselement gemäß Anspruch 33, wobei das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um bei jedem vorbestimmten Intervall die Qualitätsmessung anzufordern und die Messung anzufordern und die Entscheidung durchzuführen.

35. Netzwerkkonfigurationssteuerungselement gemäß Anspruch 33, wobei das Netzwerkkonfigurationssteuerungselement (TA) angepasst ist, um nach einem Veranlassen durch einen Betreiber der Netzwerke oder nach einem Ändern der Netzwerkkonfiguration die Qualitätsmessung anzufordern und die Messung anzufordern und die Entscheidung durchzuführen.

Es folgen 6 Blatt Zeichnungen

Anhängende Zeichnungen

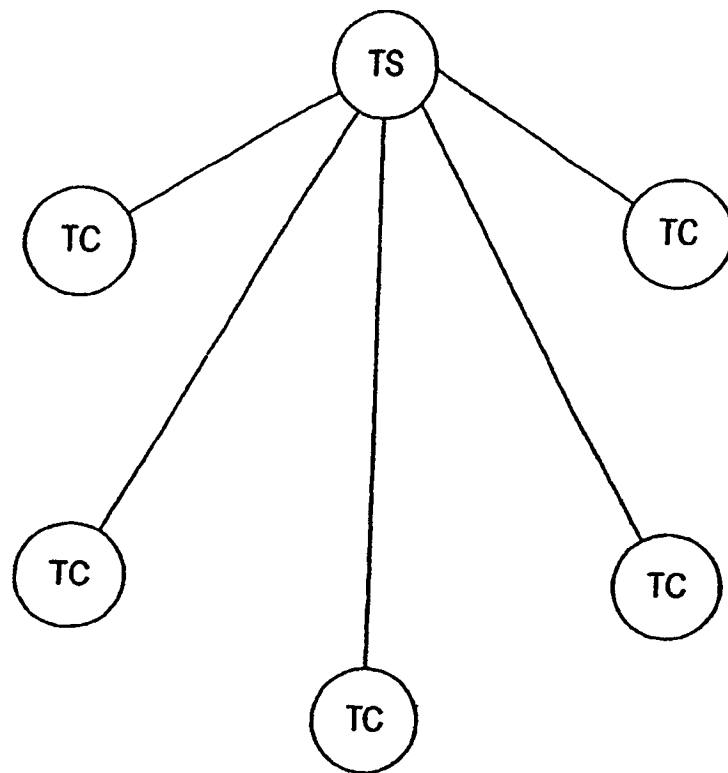


Fig. 1(a)

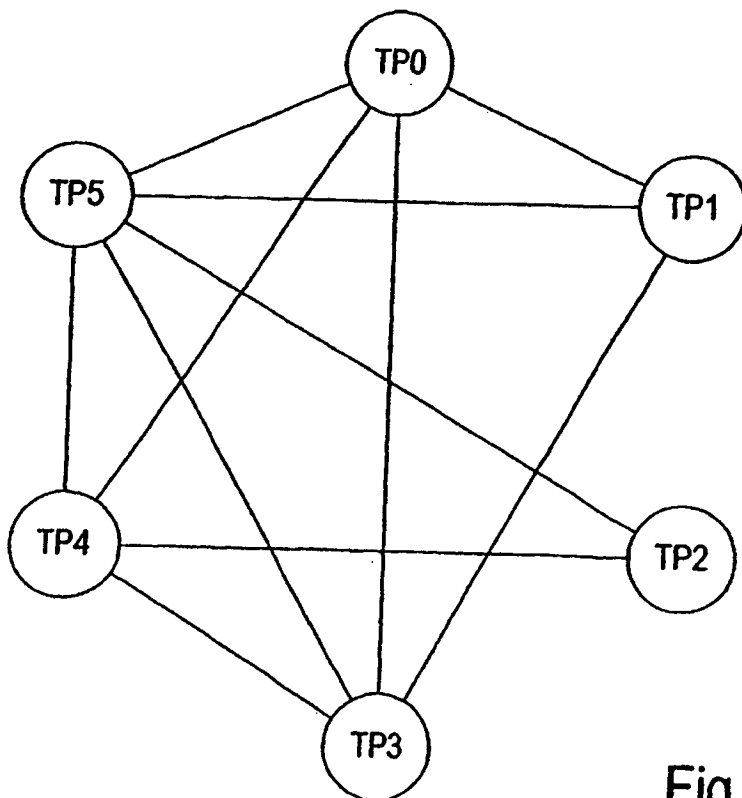


Fig. 1(b)

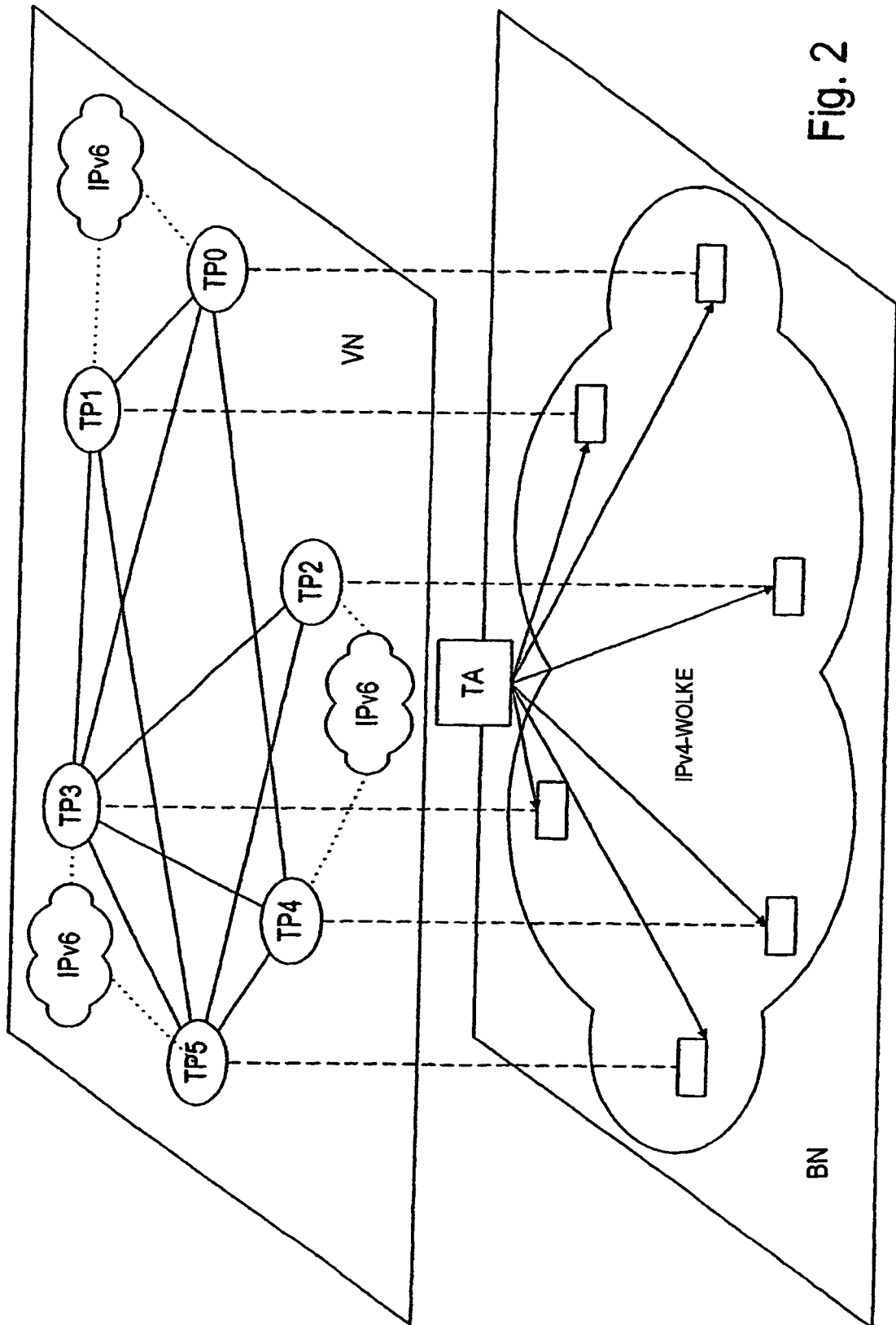


Fig. 2

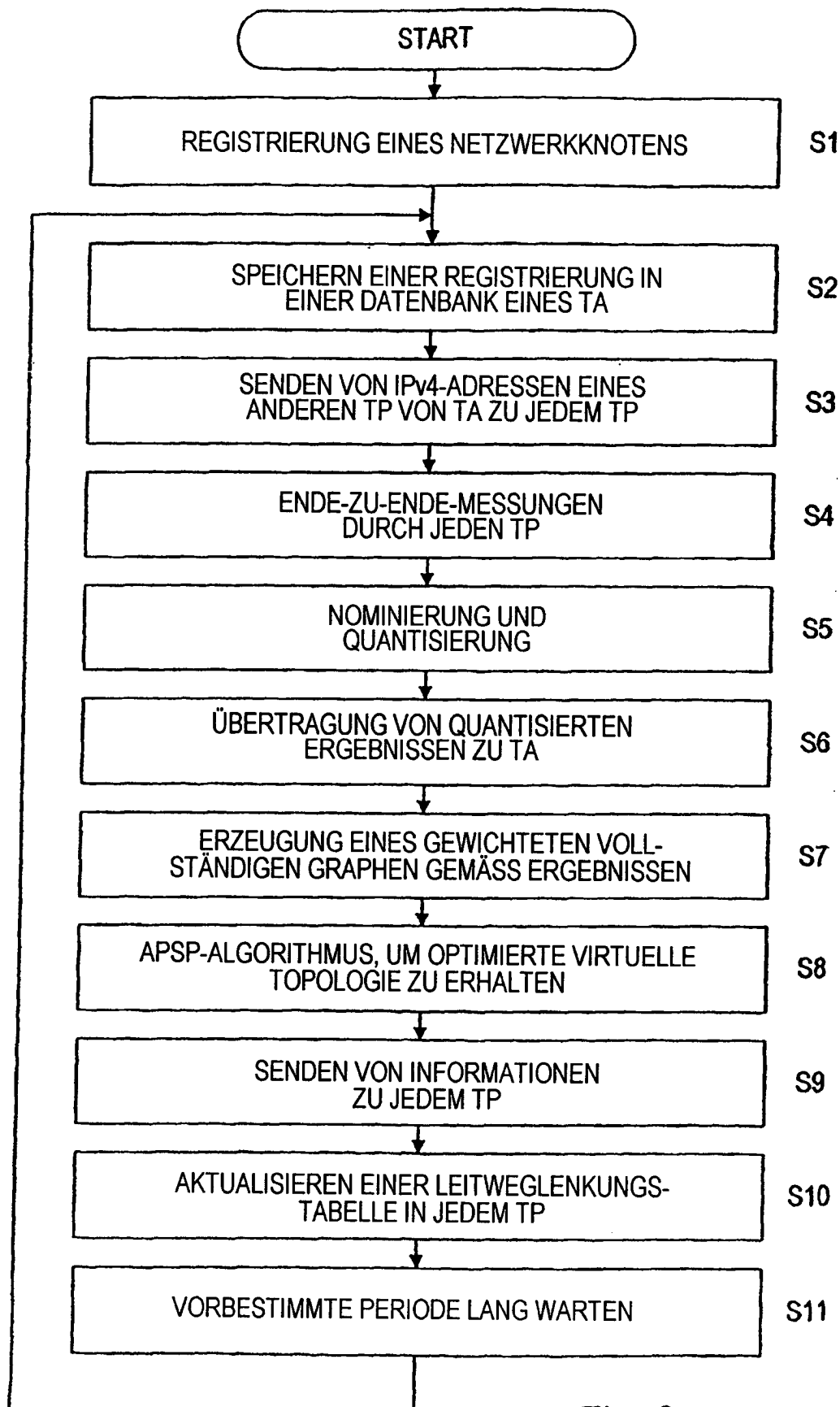


Fig. 3

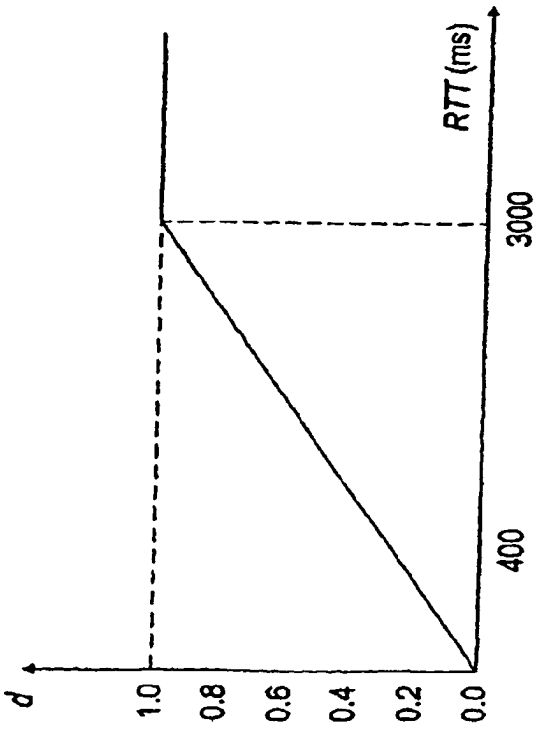


Fig. 4(a)

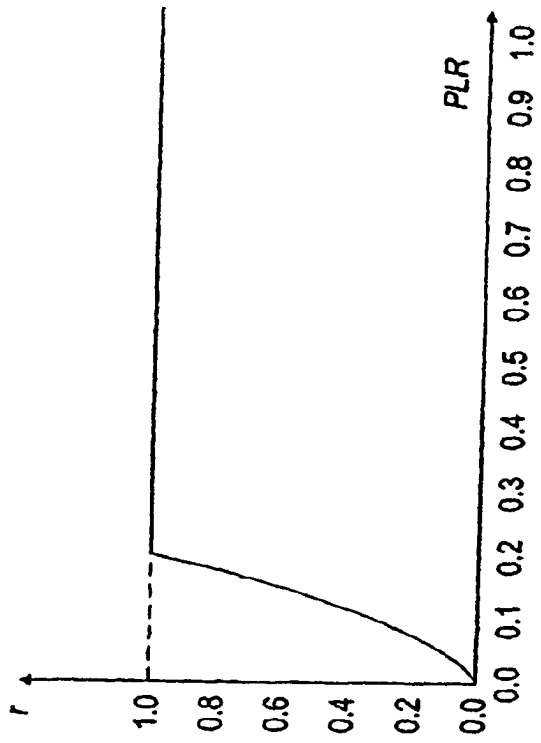


Fig. 4(b)

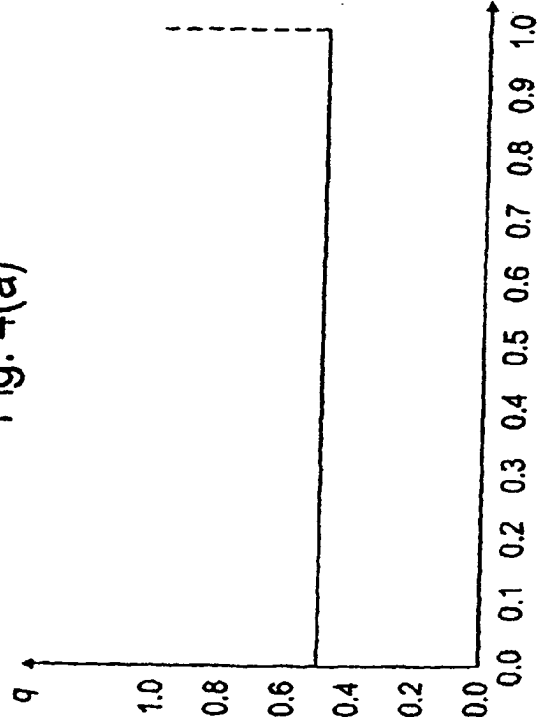


Fig. 4(c)

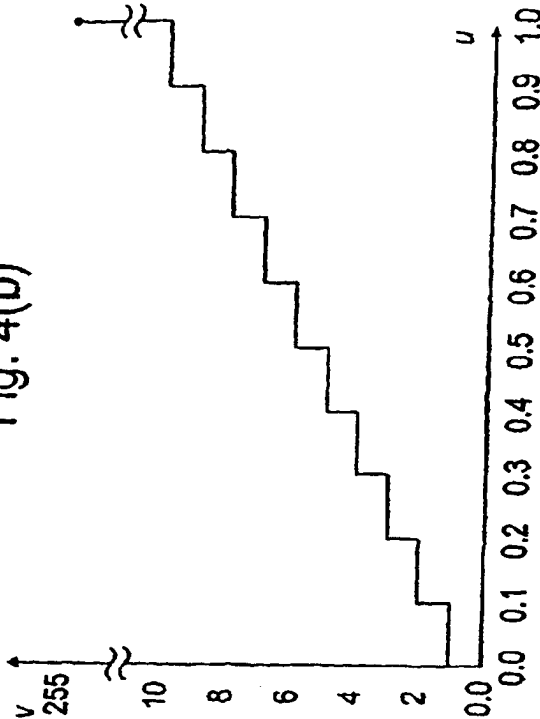


Fig. 4(d)

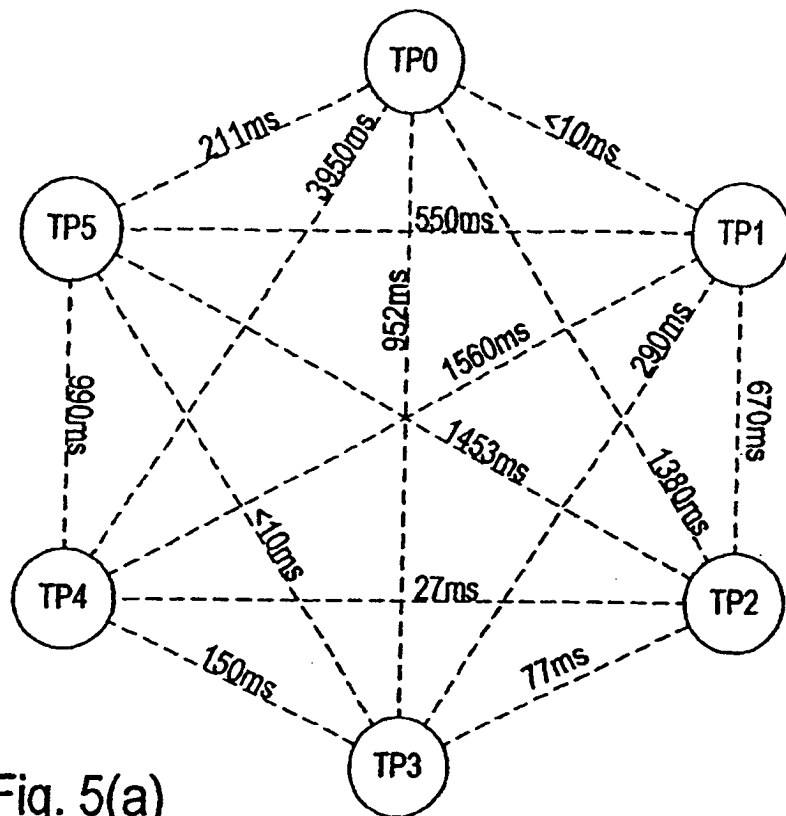


Fig. 5(a)

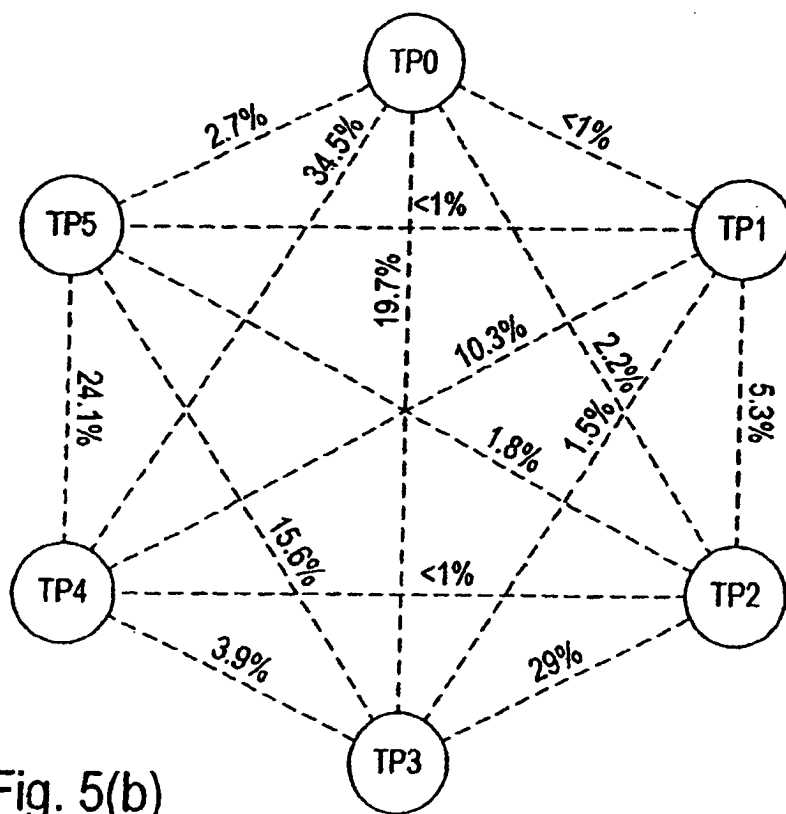


Fig. 5(b)

