(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
21. Mai 2004 (21.05.2004)

PCT

(10) Internationale Veröffentlichungsnummer
WO 2004/043014 A2

(51) Internationale Patentklassifikation[7]: H04L 12/56

(21) Internationales Aktenzeichen: PCT/EP2003/012015

(22) Internationales Anmeldedatum:
29. Oktober 2003 (29.10.2003)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
102 52 061.5 8. November 2002 (08.11.2002) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: **HAUBOLD, Sören** [DE/DE]; Schanzenbachstr. 9, 81371 München (DE). **KOCH, Rainer** [DE/DE]; Kolpingstr. 17A, 86504 Merching (DE). **SCHMID, Björn** [DE/DE]; Josef-Neumeier-Str. 18, 84503 Altötting (DE).

(74) Gemeinsamer Vertreter: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).

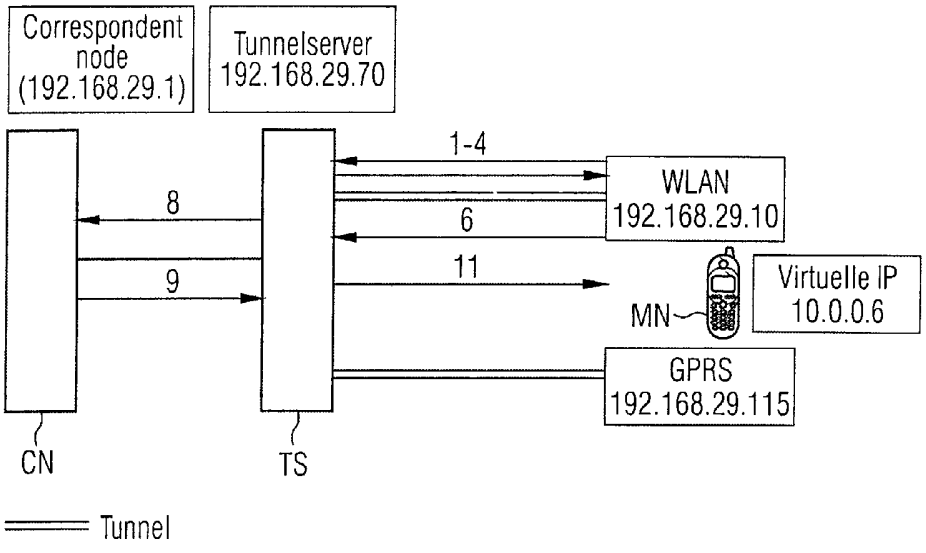
(81) Bestimmungsstaaten *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Bestimmungsstaaten *(regional)*: ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR TRANSMITTING DATA

(54) Bezeichnung: VERFAHREN ZUM ÜBERTRAGEN VON DATEN



(57) Abstract: The invention relates to a method for establishing a data connection and to a method for transmitting data or data packets without interruption from a mobile communications unit (MN) via a tunnel server (TS) with a public internet protocol (IP) address to different networks or communications components (CN) located therein. Data, which should be sent to an internet server serving as an example of a communications component, are sent in a tunnel to the tunnel server using a public IP address. Either UDP or TCP packets can be used as the tunnel. The tunnel server unpacks the packets and forwards them to the target computer after it has registered its own IP address as the sender. Packets that should be sent to the mobile communications unit are thus addressed to the tunnel server. When data arrives at the tunnel server, they are packed and forwarded to the current IP address of the mobile communications unit. The communications unit eliminates the tunnel and transfers the data to the application.

[Fortsetzung auf der nächsten Seite]

WO 2004/043014 A2

eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: Die vorliegende Erfindung betrifft ein Verfahren zum Herstellen einer Datenverbindung und ferner ein Verfahren zum Übertragen von Daten bzw. Datenpaketen ohne Unterbrechung von einem mobilen Kommunikationsgerät (MN) über einen Tunnelserver (TS) mit einer öffentlichen Internetprotokoll (IP)-Adresse zu verschiedenen Netzwerken bzw. Kommunikationskomponenten (CN) in diesen. Über öffentlichen IP-Adresse werden Daten, die an einen Internetserver als Beispiel für einen Kommunikationskomponente geschickt werden sollen, in einem Tunnel an den Tunnelserver geschickt. Als Tunnel können entweder UDP oder TCP Pakete verwendet werden. Der Tunnelserver entpackt die Pakete wieder und leitet sie an den Zielrechner weiter, nachdem er seine eigene IP-Adresse als Absender eingetragen hat. Somit werden auch Pakete, die an das mobile Kommunikationsgerät geschickt werden sollen, an den Tunnelserver adressiert. Wenn Daten beim Tunnelserver eintreffen, werden sie verpackt und an die aktuelle IP-Adresse des mobilen Kommunikationsgeräts weitergeleitet. Das Kommunikationsgerät entfernt das Tunnel und übergibt die Daten der Applikation.

Beschreibung

Verfahren zum Übertragen von Daten

Gebiet der Erfindung

Die vorliegende Erfindung betrifft ein Verfahren zum Herstellen einer Datenverbindung und ferner ein Verfahren zum Übertragen von Daten bzw. Datenpaketen ohne Unterbrechung von einem mobilen Kommunikationsgerät über einen Tunnelserver mit einer öffentlichen Internetprotokoll (IP)-Adresse zu verschiedenen Netzwerken bzw. Kommunikationskomponenten in diesen. Über öffentlichen IP-Adresse werden Daten, die an einen Internetserver als Beispiel für einen Kommunikationskomponente geschickt werden sollen, in einem Tunnel an den Tunnelserver geschickt. Als Tunnel können entweder UDP oder TCP Pakete verwendet werden. Der Tunnelserver entpackt die Pakete wieder und leitet sie an den Zielrechner weiter, nachdem er seine eigene IP Adresse als Absender eingetragen hat. Somit werden auch Pakete, die an das mobile Kommunikationsgerät geschickt werden sollen, an den Tunnelserver adressiert. Wenn Daten beim Tunnelserver eintreffen, werden sie verpackt und an die aktuelle IP-Adresse des mobilen Kommunikationsgeräts weitergeleitet. Das Kommunikationsgerät entfernt das Tunnel und übergibt die Daten der Applikation. Vorteilhafterweise wird vom mobilen Kommunikationsgerät automatisch auf die schnellste verfügbare Verbindung umgeschaltet, indem die Tunnelverbindung über ein Netz beendet wird und über das neue Netz aufgebaut wird.

Hintergrund der Erfindung

Es wird von einer Situation ausgegangen, bei der ein mobiles Kommunikationsgerät, wie ein Mobilfunkgerät bzw. Mobiltelefon, über ein ihm zugeordnetes Mobilfunknetz mit einer Kommunikationsserver, der sich beispielsweise im

Internet befindet, in Kommunikationsverbindung steht, um Daten von dem Kommunikationsserver herunterzuladen. Es sei zunächst angenommen, dass es sich bei dem Mobilfunknetz um ein WLAN (Wireless Local Area Network)-Netz handelt. Wechselt nun das mobile Kommunikationsgerät in ein anderes Mobilfunknetz, beispielsweise in ein GPRS (General Packet Radio Service)-Netz, so ändert sich dabei die Internetprotokoll-Adresse des mobilen Kommunikationsgeräts. Dies hat zur Folge, dass die von dem Kommunikationsserver versendeten Datenpakete somit nicht mehr das mobile Kommunikationsgerät erreichen und der Herunterladevorgang abbricht.

Darstellung der Erfindung

Es ist somit die Aufgabe der vorliegenden Erfindung, eine Möglichkeit zu schaffen, durch die eine ununterbrochene Datenverbindung zu einem mobilen Kommunikationsgerät bestehen bleibt, selbst wenn dieses zu einem anderen Kommunikationsnetz, insbesondere Mobilfunknetz wechselt.

Diese Aufgabe wird durch ein Verfahren gemäß Anspruch 1 gelöst. Vorteilhafte Ausgestaltungen sind Gegenstand der Unteransprüche.

Bevorzugte Ausführungsformen der vorliegenden Erfindung werden nachfolgend Bezug nehmend auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

Figur 1 den Aufbau eines IP Paketes;

Figur 2 die Adressierung der verschiedenen Subnetzklassen;

Figur 3 den Einsatz von Routern bei verschiedenen Subnetzen;

Figur 4 den Aufbau eines ICMP Paketes;

Figur 5 den Aufbau eines UDP Paketes;

Figur 6 den Aufbau eines TCP Paketes;

Figur 7 den Paketverlauf beim TCP Verbindungsaufbau;

Figur 8 den Paketverlauf beim TCP Datenverbindung;

Figur 9 den Paketverlauf beim TCP Verbindungsabbau;

Figur 10 der Aufbau eines "Sliding Window";

Figur 11 der Paketverlauf beim Einsatz eines "Sliding Window";

Figur 12 die Auswirkung eines TCP Paketes auf die Datenrate;

Figur 13 einen möglichen Aufbau eines realen Netzes;

Figur 14 einen vereinfachten Aufbau eines TCP- oder UDP-Tunnels;

Figur 15 den prinzipieller Aufbau der Netzstruktur gemäß einer Ausführungsform der Erfindung;

Figur 16 den möglichen Aufbau eines realen Netzes;

Figur 17 einen Aufbau eines beispielhaften Netzes für schnelle Netzwerkverbindungen;

Figur 18 einen Aufbau eines beispielhaften Netzes für langsame Netzwerkverbindungen;

Figur 19 Ausgetauschte Datenpakete beim Verbindungsaufbau eines UDP-Tunnels;

Figur 20 Ausgetauschte Datenpakete beim Verbindungsaufbau eines TCP Tunnels;

Figur 21 eine schematische Darstellung zur Darstellung des Aufbau einer Tunnelverbindung sowie einer Datenpaketübertragung über die Tunnelverbindung an einen Kommunikationspartner bzw. eine Correspondent Node;

Figur 22 einen weiteren möglichen Aufbau eines realen Netzes;

Figur 23 Aufbau einer Netzstruktur, in der der Tunnelserver im Corenetzwerk integriert ist.

Grundlagen - Überblick über Protokollebenen

Um bei der Datenkommunikation möglichst transparent arbeiten zu können, verwendet man das OSI Schichtmodell. Dieses besteht aus 7 verschiedenen Schichten, die aufeinander aufsetzen.

Im Bereich des Internets benutzt man üblicherweise nur 5 verschiedene Schichten. Jede Schicht nimmt die Daten der über ihr liegenden Schicht, fügt ihre eigenen Informationen und Funktionalität hinzu und reicht sie an die unter ihr liegende Schicht weiter. Aufgrund dieser Technik können in den einzelnen Schichten Veränderungen gemacht werden, ohne dass es für die anderen von Belang ist. Die 5 Schichten, die im Internetbereich verwendet werden, sind:

1. Physikal-Layer (Physikalische Schicht)
Der Physikal-Layer ist für die Bitübertragung zuständig. Er beinhaltet die verschiedenen Verfahren der Übertragung eines Bit über Luft oder Kabel. Hier sind z.B. Ethernet, ISDN oder die verschiedenen Funkerverfahren zu nennen.

2. Link-Layer (Verbindungsschicht)
In dieser Schicht werden die Daten zu Paketen verpackt und der Versand der Pakete organisiert. Außerdem beinhaltet diese Schicht Fehlerkorrekturmechanismen.

3. Network-Layer (Netzwerkschicht)
Die Aufgabe dieser Schicht ist das Weiterleiten (routen) der Daten. Dies wird normalerweise mit Hilfe von Adressen durchgeführt.

4. Transport-Layer (Transportschicht)
Diese Schicht regelt die Übertragung der Daten zwischen den Servern. Hierbei muss darauf geachtet werden, dass alle Pakete richtig ankommen.

5. Application-Layer (Anwendungsschicht)
Der Anwender kommuniziert mit dem Application Layer. Dies erfolgt durch das Programm, das er benutzt.

Unter Applikation versteht man das Programm, das auf dem mobilen Kommunikationsgerät ausgeführt wird. Je nach Anwendung werden verschiedene Protokolle verwendet. Ein Internet-Browser verwendet z.B. das HTTP (hypertext transport protocol) Protokoll. Ein Internet Browser ist ein Programm, mit dem man unter anderem HTML (hypertext markup language) Seiten darstellen kann. Diese Seiten können Text oder Grafiken enthalten. Die am weitesten verbreiteten Browser sind der Internet Explorer, der Netscape-Navigator und Opera. Zum Übertragen von HTML Seiten wird das HTTP Protokoll verwendet. Das HTTP Protokoll ist ein einfaches Protokoll. Es beschreibt einen definierten Satz von Nachrichten und Antworten, mit denen ein Client und ein Server während einer HTML Sitzung kommunizieren. Jede Anfrage eines Web-Browsers an einen Web-Server nach einem neuen Dokument stellt eine neue Verbindung dar. Das HTTP Protokoll dient der Adressierung der Objekte über URL (uniform resource locator), es wickelt die Interaktion zwischen Client und Server ab und

sorgt für die Anpassung der Formate zwischen Client und Server.

Ein FTP Programm ist eine Software, mit der Dateien von einem FTP Server heruntergeladen werden können. Für die Übertragung der Dateien wird das FTP Protokoll verwendet. FTP basiert auf dem Transportprotokoll TCP und kennt sowohl die Übertragung zeichencodierter Information als auch von Binärdaten. In beiden Fällen muss der Benutzer eine Möglichkeit besitzen zu spezifizieren, in welcher Form die Daten auf dem jeweiligen Zielsystem abzulegen sind. Die Dateiübertragung wird vom lokalen System aus gesteuert, die Zugangsberechtigung für das Zielsystem wird für den Verbindungsaufbau mittels User-Identifikation und Passwort überprüft.

Mit der Real-Player-Software von der Firma Real, können Audio- und Videostreams von speziellen Servern abgerufen werden. Hierbei kann zwischen verschiedenen Übertragungsprotokollen ausgewählt werden. Die Daten können sowohl mit TCP oder UDP übertragen werden. Beim Real Player kann die Datenrate eingestellt werden, mit der ein Stream heruntergeladen werden soll.

Mit dem Programm MGEN kann gezielter Datenverkehr in einem Netzwerk erzeugt werden. MGEN ist eine open source Software vom Naval Research Laboratory.

Im folgenden soll nun ausführlicher auf die Network-Layer und Transport-Layer eingegangen werden.

Network-Layer

IP-Protokoll

Das IP (Internet Protocol) ist für den Transport von Paketen über mehrere Netze hinweg verantwortlich. Im Header (Vorsatz)

eines IP Paketes stehen die Empfänger(Destination = Ziel)- und die Ursprungs(Source = Ursprung)-Adresse eines Paketes. IP ist ein verbindungsloses Protokoll, mit dem die Daten unabhängig versendet werden. Die korrekte Zusammensetzung der Pakete übernimmt der Layer 4.

Im folgenden sind die wichtigsten Eigenschaften eines IP Paketes erläutert: In den ersten 4 Bits eines IP Paketes steht die verwendete Version. Zur Zeit werden für die Version 4 die Bitfolge „0100" und für die Version 6 die Bitfolge „0110" eingetragen. Im Folgenden wird die Version 4 verwendet. Die Länge des IP Header in 32-Bit Worten wird in den folgenden 4 Bits angegeben. Die Angabe ist erforderlich, weil das Optionsfeld eine variable Länge hat. Die Type of Service Bits dienen der Quality of Service (QoS) Bestimmung. Sie geben an, wie die IP Pakete von Routern behandelt werden sollen.

Ein IP Paket darf eine Größe von 576 bis 65 536 Bytes haben. Da der kleinste Header schon 20 Byte beansprucht, können somit maximal 65 516 Byte in einem Paket übertragen werden. Für die Längenangabe sind im Protokoll 16 Bit reserviert. Das DF (Don't fragment) Bit gibt an, ob ein Paket weiter aufgeteilt werden darf oder nicht. Der Fragment Offset beschreibt in Verbindung mit dem MF (more fragment) Bit die Position des Segments relativ zum Dateianfang.

Der Quellrechner eines Paketes gibt eine Zeit vor, die sich das Paket maximal im Netz aufhalten darf. Diese Zeit nennt man TTL (Time to live), und für sie sind 8 Bits reserviert. Jeder Router (Netzknoten), der von dem Paket durchlaufen wird, verringert diesen Zeitstempel um 1. Wenn der TTL Wert 0 beträgt, wird das Paket verworfen.
Danach folgen die Source und Destination IP Adresse.

Die IP Adressen geben den Zielrechner bzw. den Ursprungsrechner des Paketes an. Damit die IP Adressen besser zu lesen sind, werden sie meistens als vier Dezimalzahlen, die durch einen Punkt getrennt sind, geschrieben z.B. 192.168.2.20. Aus den 32 Bit ergeben sich 2^{32} = 4 294 967 296 mögliche Adressen.

Der IP Adressraum ist hierarchisch aufgebaut, somit ergibt sich ein Zusammenhang zwischen der verwendeten IP Adresse und dem wirklichen Standpunkt des Subnetzes. Ein Teil der IP Adresse gibt das Netz wieder, in dem sich der Rechner befindet, der andere Teil die Rechnernummer in diesem Netz. Der vordere Teil der 32 Bit enthält die Netz ID, der hintere die Host ID. Bei der IP Version 4 gibt es 3 wichtige IP Adressklassen, die die Art des Subnetzes bestimmen.
In einem Klasse A Netz können sich 16,7 Mio Rechner befinden, es können maximal 128 Klasse A Netze aufgebaut werden.
Von Klasse B Netzen können 16 000 mit 65 000 Teilnehmern eingerichtet werden. Wie aus Figur 2 ersichtlich ist, kann ein Klasse C Netz verwendet werden, wenn nur 256 Rechner adressiert werden sollen. Ein Rechner in einem Subnetz kann nur angesprochen werden, wenn seine IP Adresse die Netzwerk ID dieses Netzes hat.

Damit Rechner in verschiedenen Subnetzen untereinander kommunizieren können, müssen sie über einen Router verbunden werden (vgl. dazu auch Figur 3, in der Subnetze A,B,C durch Router mit dem Internet verbunden werden). Wenn ein Rechner mit einem Partner in einem anderen Subnetz eine Kommunikation starten möchte, schickt er die Pakete an den Router des eigenen Netzes. Dieser leitet alle Pakete, deren Ziel nicht im eigenen Subnetz liegt, in sein Nachbarnetz weiter. In der Routing Tabelle, die jeder Router besitzt, sind Regeln eingetragen, nach denen die eintreffenden Pakete weitergeleitet werden.

ICMP

Das ICMP (Internet control message protocol) ist ein Protokoll zur Übertragung von Statusinformationen und Fehlermeldungen der Protokolle zwischen IP Netzknoten. Besonders Gateways und Hosts benutzen ICMP, um Berichte über Probleme mit Datagrammen zur Originalquelle zurückzuschicken.

Das ICMP wird von IP wie ein Protokoll höherer Schichten behandelt und ist integraler Bestandteil des IP Protokolls. Daher setzt sich der ICMP Header auch aus einem IP Header (in Figur 4 durch die oberen sechs Zeilen dargestellt) mit nachfolgenden ICMP Daten zusammen.

Das Datenformat von ICMP kennt den ICMP Typ (8 Bit), den ICMP Code (8 Bit) und die ICMP Checksum (16 Bit), gefolgt von der ICMP message (224 Bit). Je nach Meldung werden weitere Datenfelder hinzugefügt. Im Falle des Timestamp-Request werden weitere drei 4 Byte Felder für den Timestamp hinzugefügt: Originate Timestamp (4 Byte), Receive Timestamp (4 Byte) und Transmit Timestamp (4 Byte).

UDP

Das User Datagram Protocol (UDP) wird benutzt, um Daten verbindungslos zu übertragen. UDP hat einen minimalen Protokollmechanismus. Dabei wird weder die Ablieferung eines Datagrammes beim Zielpartner garantiert, noch sind Vorkehrungen gegen eine Duplizierung oder eine Reihenfolgevertauschung getroffen. Somit ist nur ein sehr kleiner Header notwendig. Im Header stehen der Ursprungs (Source) Port, der Empfänger (Destination) Port, die Länge und eine Prüfsumme.

Die Vergabe der Port-Nummern dient der Identifikation der verschiedenen Datenströme. Über diese Port-Nummern erfolgt

der gesamte Datenaustausch zwischen UDP und den Anwendungsprozessen. Jeder Rechner hat verschiedene Ports, über die er kommunizieren kann, somit können verschiedene Prozesse parallel arbeiten. Die Vergabe der Port-Nummern an Anwendungsprozesse geschieht dynamisch und wahlfrei, es können 65535 Ports pro Rechner vergeben werden. 1024 Ports sind für bestimmte, häufig benutzte Anwendungsprozesse fest vergeben. Diese werden als Assigned Numbers bezeichnet. Das Source Portnummern Feld in einem UDP Paket ist optional.

Daten, die mit dem UDP Protokoll verschickt werden, werden nicht von dem Empfänger bestätigt. Falls auf dem Weg Daten verloren gehen, werden diese nicht noch einmal übertragen. UDP-Pakete (vgl. für den Aufbau eines UDP-Pakets Figur 5) werden vor allem für Broadcast- oder Multicast-Pakete benötigt. Außerdem wird es oft bei der Übertragung von Video- oder Audiodaten verwendet, da es hier nicht so viel ausmacht, wenn ein Paket verloren geht.

TCP

Etwas anderes ist das beim TCP-Protokoll. Dieses Protokoll wird verwendet, um eine zuverlässige Datenverbindung aufzubauen. Um dies zu gewährleisten, wird im Header eine Paketnummer mit übertragen. Der Datenstrom wird in kleine Segmente unterteilt, und jedes Segment erhält eine solche Paketnummer.

Der Empfänger setzt den Datenstrom wieder in der richtigen Reihenfolge zusammen und bestätigt den Empfang mit einer ACK Meldung. Falls ein Paket verloren geht, fordert der Empfänger dies erneut vom Sender an.

Der Aufbau eines TCP Paketes ist in Figur 6 dargestellt.

Die in Figur 6 verwendeten Abkürzungen bedeuten dabei:

URG: Dringend-Zeiger ist gültig
ACK: Bestätigungsnummer ist gültig
PSH: Push-Daten: sofort an Anwendung weitergegebn
RST: Reset einer Verbindung
SYN: Signalisiert Verbindungsaufbau und Verbindungsbereitschaft
FIN: Abbau der Verbindung

Wenn ein TCP Protokoll benutzt wird, muß zuerst eine TCP - Verbindung aufgebaut werden, wie es beispielhaft in Figur 7 gezeigt ist. Hierfür schickt die Node (Knoten bzw. Netzwerkknoten) A eine SYN Meldung an die Node B. Diese Meldung wird genauso mit dem Wunsch nach einem Verbindungsaufbau beantwortet. Außerdem wird das erste Paket mit einem SYN-ACK bestätigt. Der Verbindungsaufbau ist abgeschlossen, wenn die Node B ein Bestätigungs-SYN - ACK von der Node A erhalten hat.

Nachdem die Verbindung aufgebaut ist, kann mit dem Übertragen von Daten begonnen werden, wie es beispielhaft in Figur 8 gezeigt ist. Beim Übertragen eines Datenpaketes wird in den Header die aktuelle Sequenz - Nummer, hier zum Beispiel 3777, eingetragen. In diesem Beispiel werden mit dem ersten Paket 120 Bytes übertragen.

Die Node B bestätigt das Paket mit einem ACK, indem sie die ACK-Nummer auf das erste noch nicht richtig übertragene Paket setzt. In unserem Beispiel 3897 (3777 + 120). Hiermit werden alle vorherigen Sequenznummern bestätigt. Beim TCP Protokoll kann eine Bestätigungsmeldung mit einem Datenpaket kombiniert werden. Wenn die Node A ein ACK erhalten hat, schickt sie das nächste Paket.

Der Abbau einer TCP-Verbindung, wie es beispielhaft in Figur 9 gezeigt ist, wird von einer FIN-Meldung eingeleitet. Diese

wird von dem Verbindungspartner mit einem ACK und einer Aufforderung zum Beenden der Verbindung bestätigt. Daraufhin bestätigt auch die Node A den Erhalt des Paketes.

Sliding Window / Flusscontrolle (Sendefenster / Lastfenster)

Um die Übertragungsgeschwindigkeit einer bestehenden TCP Verbindung anzupassen, wird die Technik des sliding Window (gleitendes Fenster) benutzt. Ein Host muss mit dem Versenden neuer Daten nicht immer warten, bis das letzte Paket bestätigt worden ist, sondern kann gleich das nächste Paket schicken. Der Sender darf nur eine bestimmte Anzahl von Daten verschicken, die noch nicht bestätigt worden sind, damit der Puffer auf der Empfängerseite nicht überläuft. Diese Anzahl wird vom Empfänger angegeben und mit der ACK Meldung übertragen. In Figur 10 ist das mögliche Window als die zweite und dritte Daten-Markierung umfassend eingezeichnet. Die Daten, die schon bestätigt wurden, sind links vom Window, die Daten, die noch nicht bestätigt wurden, sind im Window (zweite Datenmarkierung) dargestellt. In diesem Fall dürfte der Sender noch Daten versenden (dritte Datenmarkierung). Wenn die Window-Größe vom Empfänger auf 0 gesetzt wird, stoppt der Sender die Datenübertragung.

Im Beispiel von Figur 11 sendet Node A ein Paket mit 500 Bytes an Node B. Node B bestätigt dieses Paket mit dem ACK = m + 500 und schickt selbst 200 Bytes an Node A. A hat eine Window size von 1500 Bytes. Sie sendet ein zweites Datenpaket mit 400 Bytes. Die unbestätigte Datenmenge ist nun 500 + 400 Bytes und somit kleiner als die Window size von 1500 Bytes. Nun trifft die Bestätigungsmeldung von Node B ein. Node A könnte 1500 - 400 Bytes = 1100 Bytes versenden und schickt ein Paket mit 450 Bytes. Mit dem nächsten Paket quittiert die Node B Datenpaket Nummer 2 und gleich darauf Paket Nummer 3. Da keine Daten mehr gesendet werden sollen, quittiert Node A Paket Nummer 2.

Flusskontrolle / Staukontrolle

Es gibt noch eine zweite Fenstergröße, das „congestion window", das durch TCP selbst dynamisch verändert wird, indem die Netzlast beobachtet wird. Dafür werden die ACK Meldungen analysiert.
Wenn ein Paket nicht nach einer gewissen Zeit rechtzeitig bestätigt wird und ein Timer ausläuft geht der Empfänger von einem Stau aus und verkleinert dieses Fenster um die Hälfte.

Ein ACK kann zu spät oder gar nicht gesendet werden, wenn ein Router aus Überlastungsgründen Pakete verwirft oder wenn auf der Funkstrecke ein Paket verloren geht. Da das TCP Protokoll aber immer von einem Stau ausgeht, da früher die Möglichkeit eines Paketverlustes auf einer Funkstrecke noch nicht bedacht wurde, verringert der Sender die Datenrate.
Damit einer Überlastung vorgebeugt wird, wird die Datenrate langsam wieder erhöht (Slow Start), indem mit jedem bestätigtem Datenpaket der Wert des window verdoppelt (Linie b in Figur 12) wird. Nach einiger Zeit wird dann nicht mehr exponentiell erhöht, sondern nur noch linear (Linie g in Figur 12). Diese Technik wurde 1988 von Jacobson erstmals beschrieben.

Falls aber auch das Wiederholungspaket verloren geht, wird nach einem Ablauf eines Timers erneut versucht, das Paket zu senden. Außerdem wird die Timeout Zeit verdoppelt, sie kann maximal eine Minute betragen. Nach 12 Wiederholungsversuchen wird die Verbindung abgebrochen. Durch den Slow Start Mechanismus zeigt der Verlust eines TCP Paketes sehr große Auswirkungen auf die übertragene Datenrate.

NAT / PAT Router

NAT (Network Address Translation) und PAT (Port and Address Translation) Router ermöglichen es, öffentliche IP Adressen einzusparen. Dies wird immer wichtiger, da es in Zukunft nicht mehr genug IP Adressen geben wird. Ein NAT (Network Address Translation) Router ermöglicht es, innerhalb des WLAN Systems private IP Adressen zu benutzen. Das NAT-Verfahren registriert die IP Adressen eines privaten Netzes und setzt diese Adressen auf öffentliche IP Adressen um.

Beim PAT Router hingegen, wie es beispielsweise in der folgenden Tabelle gezeigt ist, werden die privaten IP Adressen auf eine einzelne öffentliche IP Adresse umgesetzt. Die Unterscheidung der Rechner, die hinter dem PAT Router eingesetzt sind, wird über verschiedene Ports durchgeführt. Dafür unterhält der Router eine Tabelle mit der Zuordnung von IP Adressen und Port-Nummern. Jeder Rechner im privaten Netz bekommt einen eigenen Port am PAT Router zugeordnet.

Private Rechner	PAT Router	
IP Adresse	Port	IP Adresse
192.168.2.1	50001	129.187.26.184
192.168.2.2	50002	
192.168.2.3	50003	
192.168.2.8	50004	

Im folgenden sollen allgemeine Problem beim Handover bzw. Netzwechsel eines mobilen Kommunikationsgeräts sowie Lösungen dazu erläutert werden.

Lösungsansätze für einen Handover

Bei einem Netzwechsel z.B. von einem WLAN(Wireless Local Area Network)- zu einem GPRS(General Packet Radio Service)-Netz, wird normalerweise auch ein Subnetzwechsel durchgeführt. Dies bedeutet, dass sich die IP-Adresse der des mobilen Kommunikationsgeräs bzw. der Mobile Node MN ändert, da Adressen hierarchisch aufgebaut sind. Wenn sich aber die IP-Adresse ändert, können bestehende Netzverbindungen nicht aufrecht erhalten werden, z.B. wird eine TCP(Transmission Control Protocol)-Verbindung unterbrochen. In der Literatur werden verschiedene Verfahren zur Lösung dieses Problems beschrieben und diskutiert.

Das zur Zeit am meisten behandelte Verfahren ist die Erweiterung des IP Protokolls mobile IP Version 4.

Es soll hier beispielsweise anhand von Figur 13 nur ein kurzer Überblick über dieses Verfahren gegeben werden, um die Probleme, die dabei entstehen, zu diskutieren.

Eine mobile Node (Mobiler Knoten) bzw. ein mobiles Kommunikationsgerät MN ist ein Gerät, wie ein Notebook (insbesondere mit Funkmodul) oder auch Mobiltelefon, das den Ort des Netzanschlusses wechseln kann und den Datendienst benötigt.

Ein Home Agent bzw. eine Heimatnetzverwaltungseinheit HA ist eine Einheit im Heimatnetz der mobile Node. Er verwaltet den aktuellen Aufenthaltsort der MN und tunnelt die Daten zu einem Foreign Agent. Das Tunnelverfahren wird später erklärt.

Der Foreign Agent bzw. Fremdnetzverwaltungseinheit ist eine Einheit im Fremdnetz. Er enttunnelt die Daten und leitet sie an die MN weiter. Außerdem stellt er eine Care-of-Adresse (CoA) zur Verfügung.

Die Care-of-Adresse ist eine Adresse, die die MN in einem fremden Netz bekommt.

Eine Correspondent node (CN) ist ein Kommunikationspartner bzw. eine Kommunikationskomponente (z.B. in Form eines Servers) der MN.

Wenn eine CN mit der MN kommunizieren will, sendet sie die Daten an die IP Adresse, die die MN normalerweise im Heimatnetz hat. Wenn sich die MN nicht im Heimatnetz befindet, fängt der Home Agent das Paket ab und tunnelt es an den Foreign Agent des Netzes, in dem sich die MN zur Zeit aufhält. Der Foreign Agent (FA) empfängt die Daten, entpackt sie und leitet sie an die MN weiter.

Um zu verdeutlichen, wie ein Paket in einem Tunnel verschickt wird, ist in Figur 14 der Aufbau eines solchen Tunnels dargestellt. Das ursprüngliche Paket U wird von einem neuen Paket R ummantelt. So werden die Daten nicht an die ursprüngliche Adresse geschickt, sondern an den Foreign Agent, dessen IP Adresse in der Destination Adresse des neuen TCP Pakets steht.

Wenn eine MN ein Zelle verlässt, erkennt sie das, da der FA jede Sekunde eine Agent-Adervtise-Message via IP Multicast verschickt. Mit dieser Meldung werden aktuelle Informationen über die Überlastung, den Default-Router und mögliche Optionen verschickt. Wenn diese Meldung nicht mehr empfangen wird, wird ein Timer gestartet. Falls nach Ablauf dieses Timers immer noch keine Meldung eingetroffen ist, geht die MN davon aus, dass sie die Verbindung zum Netz verloren hat. Daraufhin kann das Umschalten auf einen neuen FA z.B. in einem GPRS Netz eingeleitet werden. Ein Problem bei den Advertise Messages ist, dass ein großes Datenvolumen erzeugt wird, das ein GPRS-Nutzer bezahlen muß.

Um den neuen FA zu finden, wartet die MN, bis der FA des neuen Netzes eine Agent-Advertising-Message verschickt hat, und fordert vom FA eine CoA an. Dem Home Agent wird nun über den FA der aktuelle Aufenthaltsort der MN mitgeteilt. Da diese Registrierung nur eine begrenzte Lebensdauer besitzt, muss sie periodisch aufgefrischt werden.

Für mobile IP Version 4 sind in der Literatur mehrere Probleme beschrieben. Zum einen ist die Authentifizierung beim Foreign Agent problematisch, da sich dieser zum Teil bei einem fremden Provider befindet.
Ein weiteres Problem besteht darin, dass die MN die CN, mit der sie kommuniziert, direkt anspricht. Sie verschickt somit Pakete, die eine andere Source Adress haben als das aktuelle Netzwerk. Wenn das Netz durch eine Firewall geschützt ist, werden solche Pakete normalerweise herausgefiltert.
Auch dauert es sehr lange, bis ein Verlust der Verbindung erkannt wird und danach der Home Agent die neue Adresse vom neuen FA mitgeteilt bekommt. Ein Handover mit mobile IP kann mehrere Sekunden dauern.

Das größte Problem bei mobile IP ist aber, dass die derzeitige Fassung von mobile IP nicht über NAT / PAT Router arbeiten. Diese Router werden immer öfter eingesetzt, da die Anzahl der Internet- Adressen knapp wird.

Tunnelserver Lösung für schnellen Handover

Überblick

Ziel ist es, ein Verfahren zu entwickeln, das wenig neue Hardware erfordert und mit dem ein Handover sehr schnell durchgeführt werden kann, damit ein Einsatz in "no-coupling-Netzen" (bei dieser Art von Netzen haben die Netzwerkbetreiber (Operators) eines WLAN- und eines GPRS-Netzes keine gemeinsam benutzten Ressourcen. Die

Handoverfunktionaliät wird von einem drittem Service-Provider angeboten. Hier finden Techniken, wie mobile IP Anwendung, wobei ein Handover mit diesen Protokollen zum Teil sehr langsam ist) möglich ist. Dabei ist es wichtig, mehrere Verbindungen über verschiedene Ports aufbauen zu können, damit die, in den Ausblicken gezeigten Implementierungsmöglichkeiten durchgeführt werden können. Außerdem ist es wichtig, ein Protokoll zu entwickeln, das bei jeder Internetverbindung, auch mit privaten IP-Adressen, funktioniert. Es soll somit sichergestellt werden, dass die Handover-Technologie in jedem öffentlichen WLAN-Hotspot funktioniert. Der Nutzer soll keinerlei Einschränkungen im Betrieb haben und den Handover - Provider jederzeit erreichen können.

Wie bereits erwähnt, ändert sich beim Umschalten zwischen zwei unterschiedlichen Datenverbindungen in der Regel das Subnet und somit die IP-Adresse. Zum einen bricht dabei die TCP Verbindung ab, da sich die IP-Adresse ändert, zum anderen müssen die Daten über andere Router geführt werden. Zur Lösung dieses Problems wird, wie es anhand des prinzipiellen Aufbaus einer möglichen Netzstruktur in Figur 15 gezeigt ist, erstmals ein Tunnelserver (TS) installiert. Dieser Tunnelserver hat eine öffentliche, feste IP Adresse. Das mobile Endgerät, die mobile Node (MN), baut über eine Datenschnittstelle eine Verbindung mit dem Tunnelserver auf und authentifiziert sich mit Hilfe eines Handshake Passwortverfahrens. Bei einer Datenanfrage der MN an eine CN werden die Pakete in ein UDP-Pakete oder TCP-Pakete eingepackt und an den Tunnelserver geschickt. Der Tunnelserver entpackt das Paket und leitet das ursprüngliche Paket an die Correspondent Node (CN) weiter. Beim Entpacken der Daten wird außerdem in das Source Adressfeld die IP Adresse des Tunnelservers eingetragen. Nachdem die CN die Daten empfangen hat, schickt sie die Antwortdaten an die MN, indem sie diese zuerst an den Tunnelserver adressiert. Der Tunnelserver kennt die aktuelle IP Adresse der MN und

verpackt die empfangenen Daten in ein UDP Tunnel oder TCP Tunnel mit der aktuellen IP Adresse der MN. Das Tunnelpaket wird von der MN entpackt und die Daten werden an die Applikation-Layer weitergereicht.

Damit ein Tunnel-Server mehrere MN verwalten kann, wird eine virtuelle IP (VIP) eingeführt. Jedes MN bekommt eine VIP, über die der Applikation-Layer kommuniziert. Über diese VIP kann auch die Abrechnung der übertragenen Daten durchgeführt werden. Dazu muss das Datenvolumen vom Tunnelserver für jede VIP mitgeschrieben werden. Ein weiterer Vorteil der Technik ist, dass die Daten, die auf der unsicheren Luftschnittstelle verschickt werden, verschlüsselt werden können.

Ausführlichere Darstellung eines Aufbaus einer Tunnelverbindung

Das Mobile Device hat z.B. die IP Adresse 192.168.29.10 von dem WLAN-Provider bekommen.

Bei der Authentifizierung werden, wie es in Figur 21 gezeigt ist, TCP-Datenpakete zwischen dem Tunnelserver und dem Mobile Node ausgetauscht. Das Mobile Node authentifiziert sich beim Tunnelserver und der Tunnelserver bei dem Mobile Node. Für die Authentifizierung kann z.B. ein dreistufiges Authentifizierungsprotokoll verwendet werden.

1.

D-Adr: 192.168.29.70	S-Adr: 192.168.29.10	Request

2.

D-Adr: 192.168.29.10	S-Adr: 192.168.29.70	Daten

3.

D-Adr:	S-Adr:	Daten
192.168.29.70	192.168.29.10	

4.

D-Adr:	S-Adr:	OK
192.168.29.70	192.168.29.10	

Nachdem die Authentifizierung erfolgreich durchgeführt worden ist, bekommt die Mobile Node eine virtuelle IP-Adresse (VIP), hier die 10.0.0.6, zugewiesen. Diese VIP wird von dem Anwendungsprogramm, das auf dem Mobilen Device ausgeführt wird, benutzt. Sie bleibt auch bei einem Netzwechsel gleich. Wenn von dem Anwendungsprogramm eine Anfrage an einen Internetserver, hier als Correspondent Node mit der IP-Adresse 192.168.29.1, gestartet wird, wird ein Paket mit der Source-Adresse 10.0.0.6 (virtuelle IP Adresse) an die Correspondent Node (192.168.29.1) verschickt.

5.

D-Adr:	S-Adr:	Daten
192.168.29.1	10.0.0.6	

Von dem Mobile Device wird dieses Datenpakete in ein Tunnel, mit der Tunnelserver IP als Zieladresse (192.168.29.70) und der aktuellen Netzwerkschnittstelle des Mobile Device als Ursprungsadresse (192.168.29.10), verpackt.

6.

D-Adr:	S-Adr:	D-Adr:	S-Adr:	Daten
192.168.29.70	192.168.29.10	192.168.29.1	10.0.0.6	

Beim Eintreffen dieses Pakets, wird vom Tunnelserver der hinzugefügte Header wieder entfernt ...

7.

D-Adr: 192.168.29.1	S-Adr: 10.0.0.6	Daten

... und die Source-Adresse gegen die IP-Adresse des Tunnelservers ausgetauscht.

8.

D-Adr: 192.168.29.1	S-Adr: 192.168.29.70	Daten

Das Datenpaket wird an die Correspondent Node weitergeleitet. Diese Antwortet mit dem folgenden Paket:

9.

D-Adr: 192.168.29.70	S-Adr: 192.168.29.1	Daten

Der Tunnelserver ersetzt die Destination Adresse des eingehenden Paketes durch die virtuelle IP Adresse des Mobilen Devices ...

10.

D-Adr: 10.0.0.6	S-Adr: 192.168.29.1	Daten

... und verpackt das Paket nun wiederum mit einem neuen zusätzlichen Header.

11.

D-Adr: 192.168.29.10	S-Adr: 192.168.29.70	D-Adr: 10.0.0.6	S-Adr: 192.168.29.1	Daten

Nachdem das Paket bei dem Mobile Device eingegangen ist, wird der zusätzliche Header wieder entfernt.

12.

D-Adr:	S-Adr:	Daten
10.0.0.6	192.168.29.1	

Es ist zu erkennen, dass die Applikation auf der Mobile Node, nur über die VIP mit der Correspondent Node kommuniziert. Die Correspondent Node sieht nur den Tunnelserver als Kommunikationspartner, die Applikation nur die VIP.

Es sei bemerkt, dass im Fall von mehreren Mobile Nodes, zur Adressierung des Tunnelservers seitens der Mobile Node neben dessen Internetprotokoll-Adresse auch eine bestimmte (einer Mobile Node zugeordnete) Port-Adresse anzugeben ist, damit eine eindeutige Zuordnung möglich ist. Entsprechend sind von dem Tunnelserver bei der Weiterleitung von Datenpaketen an die Correspondent Node entsprechende Port-Adressen in der Source-Adresse anzugeben.

Im Falle eines Handover bzw. eines Netzwechsels wird eine andere Netzwerkschnittstelle der Mobile Node, z.B. die GPRS-Schnittstelle, mit der IP-Adresse 192.168.29.115 verwendet. Es wird nun ein Tunnel, wie oben beschrieben, aufgebaut, nur dass hier statt der 192.168.29.10 die 192.168.29.115 verwendet wird. Die VIP bleibt auch in diesem Fall gleich und somit bricht die Verbindung nicht ab.

Da die Kommunikation auf Transportebene, d.h. auf der Ebene der Transportschicht, durchgeführt wird, können über verschiedene Ports mehrere parallele Tunnel-Verbindungen aufgebaut werden.

Weitere Möglichkeiten zum Aufbau einer Tunnelverbindung

Das beschriebene Verfahren basiert, wie bereits erwähnt, auf einem Tunnel zwischen Layer 3 (Network-Layer) und 4 (Transport Layer), womit es möglich ist, mehrere Tunnel

parallel aufzubauen. Solange sich die IP Adresse, z.B. bei GPRS, nicht ändert, kann diese virtuelle Verbindung aufrecht erhalten werden.

In dem in Figur 22 gezeigten Beispiel bewegt sich die Mobile Node MN von einem GPRS-Netz (1) in einen WLAN-Hotspot (2) und wieder zurück in das GPRS Netz (3). Zuerst befindet sich die MN im GPRS-Netz und baut über GPRS eine Tunnel zum Tunnelserver auf. Dies geschieht wie oben beschrieben. Wenn sich die MN nun in den WLAN Hotspot bewegt (2), wird eine zweite Tunnelverbindung aufgebaut. Die Daten können während der Authentifizierung über WLAN, weiterhin über GPRS gesendet werden.
Wenn die neue Verbindung fertig aufgebaut ist, schickt die MN über die neue Verbindungen die Meldung zum Handover. Der Tunnelserver routet alle folgenden Daten in den neuen Tunnel über die WLAN Verbindung. Das GPRS Tunnel bleibt aber weiterhin bestehen. Wenn sich die MN nun an den Rand der WLAN Zelle bewegt und die Signalstärke abnimmt, wird der Handover auf GPRS eingeleitet, indem die MN an den Tunnelserver eine Meldung zum Handover sendet. Dieser ändert wieder die Routing Tabelle und alle weiteren Daten werden über die GPRS Verbindung empfangen.

Es sei bemerkt, dass die Verbindung Wa eine aktive WLAN-Verbindung, Ga eine aktive GPRS-Verbindung und Vd nicht aktive Verbindungen darstellt.

Netzaufbau

Reales Netz

Im realen Netz verbindet der Tunnelserver als zentrale Einheit die verschiedenen Netze. Es ist möglich, Ethernet-, WLAN-, GPRS- oder UMTS(universal mobile telecommunications system)-Netze zu verwenden. Eine Erweiterung auf weitere

Netze, wie Bluetooth, USB oder IrDA ist möglich. In Figur 16 ist ein beispielhafter realer Netzaufbau dargestellt, die getunnelten Verbindungen sind durch Doppelverbindungen gekennzeichnet.

Beispielnetze

Im folgenden werden ausgehend von dem allgemeinen möglichen Aufbau eines realen Netzes gemäß Figur 16 zwei spezifischere Ausführungen für reale Netze dargestellt. Ein Netz dient dabei für die schnelle Datenkommunikation im Mbit - Bereich und ein Netz für die langsame Datenkommunikation im kbit - Bereich mit GPRS Unterstützung.

Das Netz für die schnelle Datenkommunikation ist in Figur 17 abgebildet. Der Tunnelserver ist dabei mit einem Switch (Schalteinrichtung) verbunden und hat z.B. die IP Adresse 192.168.1.20. Der Accesspoint (Zugangspunkt) ist über einen Router mit diesem Switch verbunden. Die WLAN Karte in der mobile Node hat z.B. die IP Adresse 192.168.2.32. Diese Adressen aus dem "2er" Subnetz (192.168.2.x) werden vom Router in das 192.168.1.x Netz ("1er" Subnetz) geroutet. Wenn bei der MN eine Ethernet Verbindung benutzt wird, wird sie direkt mit dem Switch verbunden und erhält die IP Adresse 192.168.1.31.
Der Tunnelserver kann über den PAT Router und den Proxy eine Verbindung in das Internet aufbauen.

Netzwerkkomponente	IP Adresse
Tunnelserver	192.168.1.20
PAT Router	192.168.1.1
Router	192.168.2.1
MN - WLAN	192.168.2.32
MN - Ethernet	192.168.1.31

Das Netzwerkszenario für die langsame Datenkommunikation über ein GPRS-Netz ist in Figur 18 dargestellt. Hier werden langsame Versuche mit GPRS durchgeführt. Der Tunnelserver hat über eine ISDN Leitung eine Verbindung mit dem Leibniz Rechenzentrum (LRZ) Zugangsserver, wobei hier das LRZ als Internet-Provider dient. Er erhält vom LRZ eine dynamische öffentliche IP-Adresse. Außerdem ist der Tunnelserver an den Switch angeschlossen. Die Ethernetkarte hat die feste IP-Adresse 192.168.2.20, der Tunnelserver dient gleichzeitig als Router ins Internet. Die Wireless-LAN-Karte, die über den Accesspoint und das Switch mit dem Tunnelserver verbunden ist, erhält z.B. die IP Adresse 192.168.2.32. In diesem Aufbau wird kein Subnetz für den WLAN-Accesspoint verwendet.

Die IP-Adresse für das GPRS-Netz bekommt die MN z.B. von T-Mobile (Tochter der Deutschen Telekom) zugewiesen. Daten werden von der MN über das GPRS-Modem an T-Mobile geschickt und über das Corenetz und den LRZ-Server zum Tunnelserver geroutet.

Die folgende Tabelle zeigt eine Zusammenfassung der den einzelnen Netzwerkkomponenten zugeordneten IP-Adressen.

Netzwerkkomponente	IP Adresse
Tunnelserver	Dynamische IP von LRZ z.B. 129.187.26.184
Router ins Internet	192.168.2.20
MN - WLAN	192.168.2.32
MN - Ethernet	192.168.2.31
MN - GPRS	Dynamische IP von T-Mobile

Handover-Szenario

Anmeldung

UDP Tunnel

Im folgenden wird vorausgesetzt, dass sich die MN zunächst bei seinem Kommunikationsnetz angemeldet bzw. eingebucht hat. Damit die MN eine Datenverbindung mit der CN aufbauen kann, muss sich die MN zuerst am Tunnelserver anmelden und authentifizieren. Die Authentifizierung wird über eine TCP Verbindung durchgeführt, da hier eine gesicherte Verbindung notwendig ist (vgl. hierzu beispielsweise Figur 19).

Mit den Paketen 1 bis 3 wird eine TCP-Verbindung (vgl. oben) aufgebaut. Nachdem die Verbindung aufgebaut ist, schickt der Tunnelserver die Versionsnummer der Serversoftware an die MN(Schritt: 4), damit zukünftig der Standard auch erweitert werden kann. Die MN bestätigt den Erhalt des TCP Paketes mit einem ACK (Schritt: 5) und schickt den Hostname an den Tunnelserver (Schritt: 6). Dieser bestätigt den Eingang mit einem ACK (Schritt: 7), berechnet eine Zufallszahl und schickt sie an die MN (Schritt: 8). Die MN verschlüsselt das Passwort mit der Zufallszahl und schickt dieses an den Tunnelserver (Schritt: 9). Daraufhin entschlüsselt der Tunnelserver das Passwort und schickt ein OK Flag an die MN (Schritt: 10). Falls das Passwort nicht stimmt wird die Verbindung abgewiesen. Bei erfolgreicher Authentifizierung beginnt der Aufbau des UDP-Sockets. Der Tunnelserver schickt seine eigene Portnummer (Schritt: 11), liest dann die Portnummer der MN ein und öffnet den Socket. (Schritt: 12) Sobald der Socket erstellt ist, können Daten im Tunnel übertragen werden. Mit den Paketen gemäß den Schritten 13 bis 16 wird die TCP-Verbindung von beiden Seiten abgebaut. Beim Aufbau eines UDP-Tunnels müssen folglich 16 Pakete verschickt werden.

TCP Tunnel

Auch beim TCP-Tunnel beginnt der Prozess mit dem Aufbau einer TCP Verbindung (vgl. dazu Figur 20, Pakete gemäß Schritte 1 bis 3). Danach erfolgt die Authentifizierung wie beim UDP-Tunnel (gemäß Schritte 1 bis 10). Allerdings muss nach dem OK-Flag kein neuer Socket aufgebaut werden, da der TCP-Socket, der für die Authentifizierung benutzt wird, auch für die Tunneldaten verwendet werden kann. Das letzte Paket wird nur mit einem ACK bestätigt(Schritt 11). Danach können sofort Daten über das Tunnel versendet werden.

Beim Aufbau eines TCP-Tunnels müssen folglich nur 11 Pakete verschickt werden. Diese Zahlen haben eine sehr große Auswirkung auf die Handoverzeit.

Datenverbindung über WLAN

Um eine TCP-Verbindung über die verschiedenen Netze aufbauen zu können, müssen zuerst die Informationen in der Routing-Tabelle richtig eingetragen werden. Dieser Vorgang soll mit dem zweiten Netzwerkszenario für langsame Datenverbindungen beschrieben werden. Beim Aufbau der Verbindung dürfen keinerlei Routing-Informationen in der Routing-Tabelle der MN stehen. Die Software trägt zuerst eine Route auf den Gateway (Netzübergangseinheit bzw. Vermittlungseinheit zwischen verschiedenen Datennetzen) des Netzes ein und gibt als Device bzw. Komponente die gewünschte Schnittstellenkarte an. Somit wird als Device eth1 eingetragen. Im zweiten Netzwerkszenario ist die IP-Adresse der Ethernetkarte die des Gateways, da der Tunnelserver gleichzeitig als Router dient. Der Eintrag lautet folglich:

Destination	**Gateway**	**Device**
192.168.2.20	0	eth1

Außerdem muss eine Route für die öffentliche IP-Adresse des Tunnelservers über die Netzwerkschnittstelle des Accesspoints unter Nennung des Gateways eingetragen werden.

Die Destination(Ziel)-IP-Adresse des Tunnelservers ist hier die vom Provider erhaltene öffentliche IP-Adresse, z.B. 129.187.26.184.

Die Routing-Tabelle sieht nun folgendermaßen aus:

Destination	**Gateway**	**Device**
192.168.2.20	0	eth1
129.187.26.184	192.168.2.20	eth1

Die Applikationssoftware führt nun die Authentifizierung durch und erstellt ein neues Netzwerkdevice mit dem Namen "TUN" für Tunnel und der virtuellen IP-Adresse. In diesem Beispiel wird die 10.0.0.5 verwendet. Zu diesem Netzwerkdevice wird eine Point-to-Point-Verbindung erstellt, und es wird eine Standardroute auf TUN eingerichtet. Nach dem Aufbau des Tunnels stehen folgende Einträge in der Routing-Tabelle:

Destination	**Gateway**	**Device**
10.0.0.5	0	tun
192.168.2.20	0	eth1
129.187.26.184	192.168.2.20	eth1
0.0.0.0	0.0.0.0	tun

Der Eintrag 0.0.0.0 bedeutet, dass alle restlichen Daten in den Tunnel geroutet werden. Die restlichen Daten sind die Pakete, die nicht an den Gateway oder an den Tunnelserver gehen.

Die Software liest die Daten der Schnittstelle TUN ein und verpackt sie in ein neues UDP oder TCP Paket wie vorher

beschrieben. Danach wird es wieder verschickt und nimmt nun die Route über die angegebene Schnittstelle.

Beim ersten Netzwerkszenario wird nicht der Tunnelserver, sondern der zusätzliche Router mit der IP-Adresse 192.168.2.1 als Gateway eingetragen

Destination	**Gateway**	**Device**
10.0.0.5 0	tun	
192.168.2.1	0	eth1
129.187.26.184	192.168.2.1	eth1
0.0.0.0	0.0.0.0	tun

Handover

Ein Handover wird vorteilhafterweise durchgeführt, wenn die Signalstärke der WLAN Verbindung abnimmt oder das Ethernetkabel gezogen wird. Außerdem kann ein Handover eingeleitet werden, wenn eine schnellere Datenverbindung erkannt wird. Diese Daten werden von der Software ständig analysiert, und bei einem Abbrechen oder Aufbau einer Datenverbindung wird ein Handover eingeleitet.

Dafür versucht die MN eine Close-Meldung (Abbruchsmeldung) an den Tunnelserver zu verschicken und baut ihre Socket-Schnittstelle ab. Die Close-Meldung kann aber nur versendet werden, wenn bei einem Handover eine Kommunikation über die alte Verbindung weiterhin möglich ist.
Der Tunnelserver baut seinen Socket beim Eingehen einer Close-Meldung ab oder terminiert den Prozess, wenn von der MN versucht wird, eine neue Datenverbindung aufzubauen und die Close-Meldung nicht eingetroffen ist.

Nun wird die neue Datenverbindung über GPRS oder Ethernet aufgebaut.

Datenverbindung über GPRS

Der Aufbau einer Datenverbindung über GPRS bzw. Ethernet funktioniert genauso wie der bei einer WLAN-Verbindung, nur dass andere Gateway-Adress und Devices benutzt werden. Beispielsweise heißt bei GPRS das Device ppp0 und die Gateway-Adress 192.168.254.254, bei Ethernet eth0 und 192.168.2.20. Die GPRS Verbindung wird nur im zweiten Netzwerkscenario verwendet.

Daraus ergeben sich somit folgende Routing-Tabellen für GPRS:

Destination	**Gateway**	**Device**
10.0.0.5	0	tun
192.168.254.254	0	ppp0
129.187.26.184	192.168.254.254	ppp0
0.0.0.0	0.0.0.0	tun

Datenverbindung über Ethernet

Für Ethernet im zweiten Netzwerkszenario sind folgende Werte in die Routing-Tabelle eingetragen:

Destination	**Gateway**	**Device**
10.0.0.5	0	tun
192.168.2.20	0	eth0
129.187.26.184	192.168.2.20	eth0
0.0.0.0	0.0.0.0	tun

Im ersten Netzwerkscenario sieht die Routing-Tabelle für Ethernet wie folgt aus:

Destination	**Gateway**	**Device**
10.0.0.5	0	tun
192.168.1.20	0	eth0
129.187.26.184	192.168.1.20	eth0
0.0.0.0	0.0.0.0	tun

Offenbart ist ein Verfahren zum Aufbau einer Datenverbindung bzw. zum Übertragen von Daten, insbesondere als Datenpakete. Dabei wurde berücksichtigt, dass herkömmlicherweise wenn zwischen zwei verschiedenen Netzwerken geschalten wird, die Benutzerverbindung verlorengeht, weil, wenn einmal ein neues Netzwerk gefunden ist, dieses dem mobilen Kommunikationsgerät des Benutzers eine neue IP(Internet Protokoll)-Adresse zuweist und somit die ursprüngliche Verbindung verloren geht. Gemäß der hier vorgestellten technischen Lösung wird vorgeschlagen, eine Vermittlungskomponente in der Ausführung eines Tunnel-Servers (TS) zu installieren, der ununterbrochen mit dem Internet verbunden ist und eine öffentliche IP-Adresse aufweist. Ein mobiles Kommunikationsgerät tritt mit dem Tunnel-Server in Verbindung und verifiziert die Authentifizierung z.B. mit einem Hand-Shake-Passwort (Passwort basierend auf einem gegenseitigen Austausch).

Wenn Daten von dem mobilen Kommunikationsgerät zu einer bestimmten Kommunikationskomponente (CN), wie einem Server zum Austauschen von Daten, übermittelt werden, dann werden die Datenpakete zur Tunnelung z.B. innerhalb eines UDP(User Datagram Protocol)-Headers vorgesehen und werden zu dem Tunnel-Server gesendet. Der Tunnel-Server entpackt die getunnelten Datenpakete dann (welche nun die Quelladresse des Tunnel-Servers haben) und leitet sie um zu der Kommunikationskomponente (CN). Wenn die Kommunikationskomponente hingegen Daten zurücksenden will, verwendet sie die IP-Adresse des Tunnel-Servers als Ziel für die Datenpakete. Der Tunnel-Server kennt die aktuelle IP-Adresse des mobilen Kommunikationsgeräts und tunnelt die Daten zu dieser Adresse z.B. wieder in einem UDP-Tunnel bzw. einer UDP-Tunnelverbindung. Die Daten werden von dem mobilen Kommunikationsgerät entpackt und zu der Anwendungsschicht geleitet.

Es wird eine virtuelle IP-Adresse (VIP) implementiert, um zu ermöglichen, dass ein Tunnelserver verschiedene mobile Kommunikationsgeräte handhaben bzw. verwalten kann. Jedes mobile Kommunikationsgerät bekommt eine VIP. Die Anwendungsschicht kommuniziert über dieser diese dediizierte IP-Adresse. Die Kommunikationskomponente verwendet nur die IP-Adresse des Tunnel-Servers und das mobile Kommunikationsgerät sieht nur seine eigene VIP-Adresse. Dies schafft für den Endbenutzer eine nahtlose Verbindung.

Es wird angenommen, dass sich das mobile Kommunikationsgerät in einem WLAN(Wireless Lan)-Netz befindet, in dem die kontinuierliche Erfassung der WLAN-Signalstärke auf Seiten des mobilen Kommunikationsgeräts ein schnelles und nahtloses Handover (Weiterreichen) zwischen dem WLAN-Netzwerk und beispielsweise einem GPRS(General Package Radio System)-Netzwerk ermöglicht. Wenn der WLAN-Träger eine schwache Signalstärke hat, dann wird ein Handover initiiert, die WLAN-Verbindung wird entfernt bzw. abgebrochen, eine neue Verbindung über GPRS wird aufgebaut und die TCP(Transmission Control Protocol)-Sitzung wird ununterbrochen fortgesetzt.

Ein weiterer Vorteil dieser Lösung besteht darin, dass die Daten verschlüsselt werden können, was insbesondere wichtig bzw. kritisch mit Bezug darauf ist, dass WLAN-Netzwerke bzw. WLAN-Verbindung nicht vollständig sicher sind.

Die Optimierung der Handover-Zeit zwischen verschiedenen Netzwerken unter Verwendung des Mobile-IP-Standard hingegen ist sehr kompliziert, weil die Mobile-IP-Version 4 ursprünglicher Weise nicht für einen schnellen Handover entwickelt worden ist. Das Ziel des oben erwähnten Ansatzes besteht darin, eine bezüglich der Geschwindigkeit optimierte Lösung zu schaffen. Ein Handover basierend auf dem Mobile-IP-Standard benötigt bis zu mehreren Sekunden. Dies ist für die Übertragung bzw. den Empfang von Echtzeitdiensten unzureichend, die eine Handover-Zeit von <100ms erfordern, um

zu verhindern, dass der Datenstrom unterbrochen wird. Die Handover-Zeit der oben erwähnten Lösung liegt nur zwischen 4 bis 70ms. Diese Zeit variiert abhängig von der Zeit, die für die Identifizierung zwischen dem mobilen Kommunikationsgerät und dem Tunnel-Server benötigt wird und hängt von der Geschwindigkeit der Datenverbindung ab. Um die gesamte Handover-Zeit zu berechnen, muss ferner die Umlaufzeit des verwendeten Netzwerks auch dazu addiert werden. Für ein WLAN-Netzwerk beträgt diese ungefähr nur 3ms.

Gemäß der oben erwähnten neuen Lösung könnte der Tunnel-Server eine integrierte Komponente des Core-Network (Kernnetzwerks) werden. Ausgehend von der in Figur 23 dargestellten Abbildung ist der Serving-GPRS-Support-Node (SGSN) für die Authentifizierung, das Routing (die Wegleitung), die Übertragung und das Mobilitätsmanagement der GPRS-Daten verantwortlich.

Der Serving-WLAN-Support-Node (SWSN) ist für das Routing, die Authentifizierung, die Übertragung und das Filtern der WLAN-Daten verantwortlich. Dieser Knoten unterstütz das DHCP (Dynamic Host Configuration Protocol) zum Allokieren bzw. Zuordnen von IP-Adressen und zum Übertragen von Vergebührungsdaten zu dem Authentifizierungs- und Vergebührungszentrum. Die Authentifizierung kann über eine SIM(Subscriber Identity Module)-Authentifizierung oder ein Passwort-Login (Passworteinbuchung) implementiert werden, während der Access Point (Zugangspunkt) für den Endbenutzer ein beliebiges WLAN-Netzwerk sein kann, dass einem Operator gehört.

Der Tunnel-Server-Support-Node (TSSN) ist für das Handover zwischen einer WLAN- und GPRS-Verbindung verantwortlich und dieser Knoten entpackt die Daten. Wenn eine WLAN-Datenverbindung verfügbar ist, werden die Daten verschlüsselt und von dem mobilen Kommunikationsgerät über den Access Point

zu dem SWSN getunnelt. Dieser erfasst den Umfang der Daten für die Vergebührung und sendet dann Pakete zu den TSSN.

Die Daten werden von dem TSSN entpackt und dann zu dem Internet gesendet. In dem Fall einer GPRS-Verbindung, werden die Daten über den GGSN geleitet und zu dem TSSN getunnelt, bei dem sie entpackt und zum Internet bzw. einer Kommunikationskomponente, beispielsweise in Form eines Datenaustausch-Servers, gesendet werden.

Patentansprüche

1. Verfahren zum Aufbau einer Datenverbindung, mit folgenden Schritten:

Verbinden eines mobilen Kommunikationsgeräts (MN) mit einem auf einem Internetprotoll basierenden ersten Kommunikationsnetz (WLAN), in dem das mobile Kommunikationsgerät (MN) eine erste reale Internetprotokoll-Adresse hat;

Anmelden des mobilen Kommunikationsgeräts (MN) über das erste Kommunikationsnetz bei einer mit diesem verbundenen Vermittlungskomponente (TS) mit einer vorbestimmten Internetprotokoll-Adresse, um dadurch eine erste Tunnelverbindung zwischen dem mobilen Kommunikationsgerät und der Vermittlungskomponente herzustellen;

wobei durch die erste Tunnelverbindung Tunnelpakete austauschbar sind, welche Datenpakete enthalten, die für eine Kommunikation des mobilen Kommunikationsgeräts mit einer mit der Vermittlungskomponente verbundenen Kommunikationskomponente (CN) bestimmt sind.

2. Verfahren nach Anspruch 1, bei dem Tunnelpakete mit von dem mobilen Kommunikationsgerät (MN) an die Kommunikationskomponente gerichteten Datenpaketen bei der Vermittlungskomponente entpackt werden, mit der vorbestimmte Internetprotokoll-Adresse der Vermittlungskomponente als Herkunftsadresse werden und zu der Kommunikationskomponente (CN) gesendet werden.

3. Verfahren nach Anspruch 2, bei dem die Kommunikationskomponente (CN) als Antwort auf von der Vermittlungskomponente (TS) übermittelte Datenpakete, an das mobile Kommunikationsgerät (MN) gerichtete Datenpakete entsprechend der angegebenen Herkunftsadresse an die

Vermittlungskomponente sendet, welche die von der Kommunikationskomponente empfangenen Datenpakete in Tunnelpakete einpackt und diese zum mobilen Kommunikationsgerät (MN) weiterleitet.

4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem die Vermittlungskomponente (TS) dem mobilen Kommunikationsgerät (MN) eine virtuelle Internetprotokoll-Adresse (IP) vergibt, die für die Kommunikation mit der Vermittlungskomponente feststehend bleibt.

5. Verfahren nach einem der Ansprüche 1 bis 4, bei dem sich das mobile Kommunikationsgerät (MN) mit einem auf einem Internetprotoll basierenden zweiten Kommunikationsnetz verbindet, in dem das mobile Kommunikationsgerät eine zweite reale Internetprotokoll-Adresse hat.

6. Verfahren nach Anspruch 5, bei dem sich das mobile Kommunikationsgeräts (MN) über das zweite Kommunikationsnetz bei der mit diesem verbundenen Vermittlungskomponente (TS) anmeldet, um dadurch eine zweite Tunnelverbindung zwischen dem mobilen Kommunikationsgerät und der Vermittlungskomponente herzustellen.

7. Verfahren nach einem der Ansprüche 1 bis 6, bei dem Tunnelpakete von der Vermittlungskomponente an das mobile Kommunikationsgerät über die erste und/oder zweite Tunnelverbindung übertragen werden.

8. Verfahren nach einem der Ansprüche 6 oder 7, bei dem Tunnelpakete über diejenige Tunnelverbindung der ersten oder zweiten Tunnelverbindung mit der größeren Übertragungsrate übertragen werden.

9. Verfahren nach einem der Ansprüche 6 oder 7, bei dem Tunnelpakete über diejenige Tunnelverbindung der ersten oder zweiten Tunnelverbindung mit der größeren

Übertragungssicherheit und/oder Übertragungsqualität übertragen werden.

10. Verfahren nach einem der Ansprüche 6 bis 9, bei dem eine der ersten oder zweiten Tunnelverbindung getrennt wird.

11. Verfahren nach einem der vorhergehenden Ansprüche, bei dem das Verbinden mit dem ersten und/oder zweiten Kommunikationsnetz, insbesondere in der Ausführung eines Mobilfunknetzes, ein Einbuchen umfasst, bei dem ein Autorisierungsvorgang stattfindet.

12. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die Tunnelpakete als TPC-Pakete übermittelt werden.

13. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die Vermittlungskomponente in einem mit dem ersten und dem zweiten Kommunikationsnetz verbundenen auf einem Internetprotoll basierenden dritten Kommunikationsnetz, wie dem Internet, angeordnet ist.

14. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die Kommunikationskomponente, insbesondere in der Ausführung eines Datenaustauchservers, in einem mit dem dritten Kommunikationsnetz verbundenen auf einem Internetprotoll basierenden vierten Kommunikationsnetz oder ebenso in dem dritten Kommunikationsnetz angeordnet ist.

15. Verfahren nach einem der vorhergehenden Ansprüche, bei dem das erste und/oder das zweite Kommunikationsnetz als ein Kommunikationsnetz auf der Basis von WLAN, UMTS, GPRS, EDGE, Bluetooth, IrDa oder Ethernet ausgebildet sind.

16. Verfahren nach einem der vorhergehenden Ansprüche, bei dem das Kommunikationsgerät ein Funkmodul umfasst, und insbesondere als ein Mobiltelefon oder ein tragbarer Computer ausgebildet ist.

FIG 1

<table>
<tr><td>4-bit Version</td><td>4-bit header Länge</td><td>8-bit type of service</td><td colspan="4">16-bit Gesamtlänge in bytes</td></tr>
<tr><td colspan="3">16-bit Identification</td><td>0</td><td>DF</td><td>MF</td><td>13-bit Fragment Offset</td></tr>
<tr><td colspan="2">8-bit Time to Live</td><td>8-bit Protokoll</td><td colspan="4">16-bit Header Prüfsumme</td></tr>
<tr><td colspan="7">32-bit Source IP Address</td></tr>
<tr><td colspan="7">32-bit Destination IP Address</td></tr>
<tr><td colspan="7">Optionen</td></tr>
<tr><td colspan="7">Nutzdaten</td></tr>
</table>

FIG 2

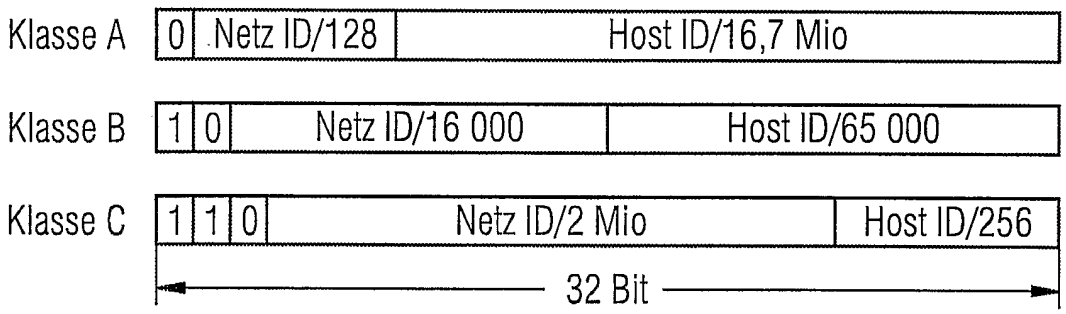


FIG 3

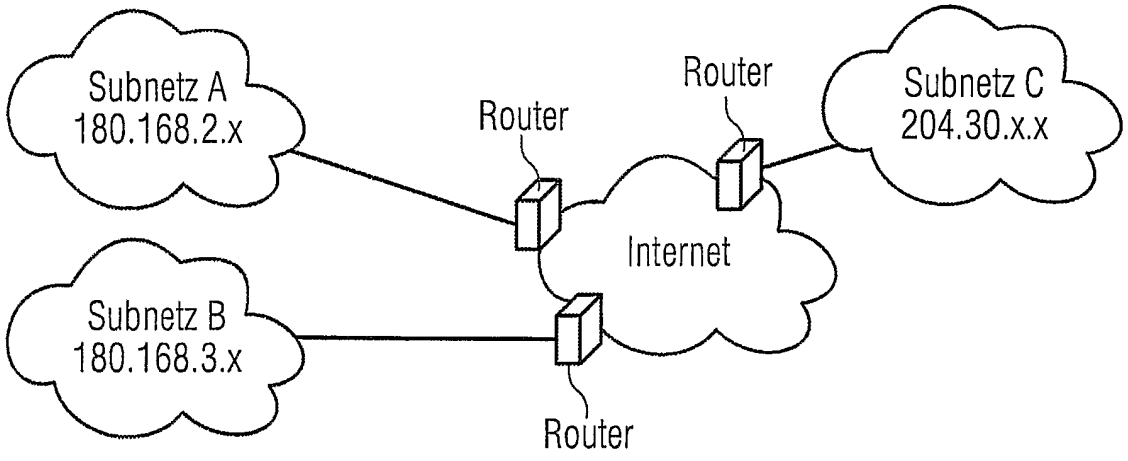


FIG 4

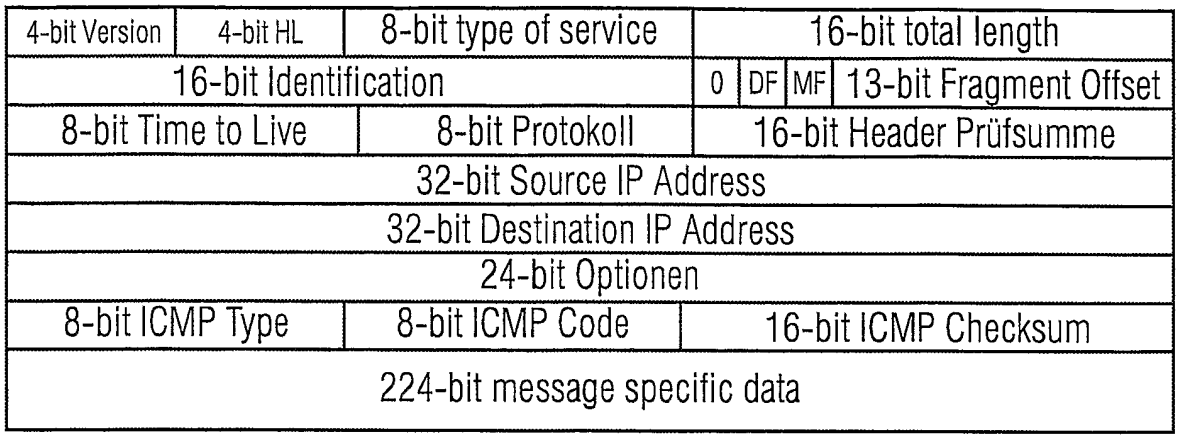


FIG 5

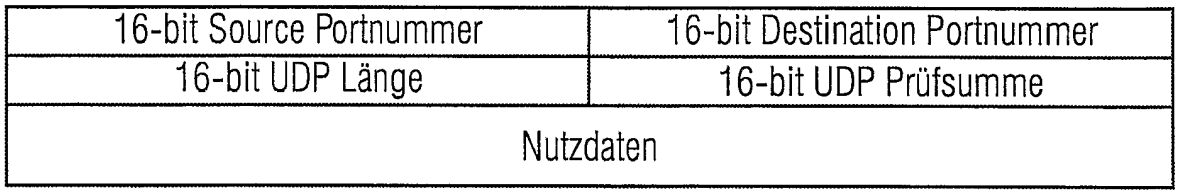


FIG 6

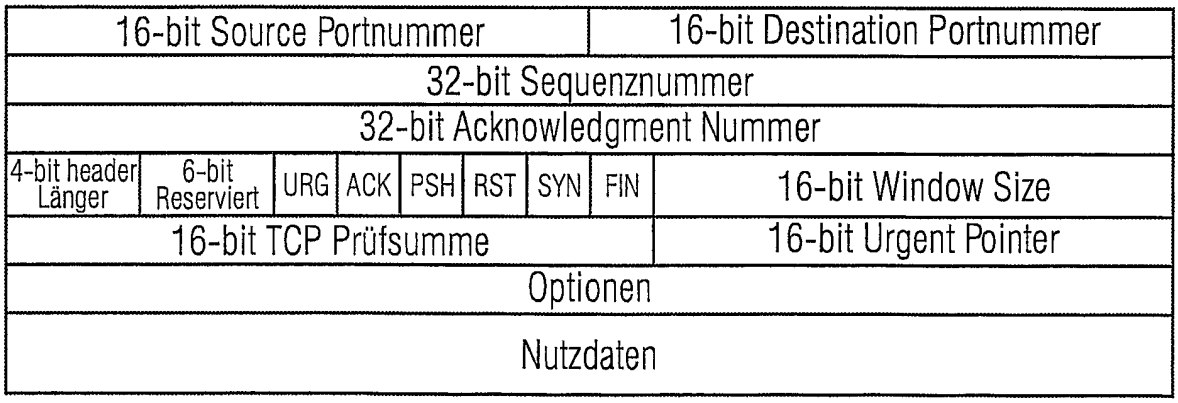


FIG 7

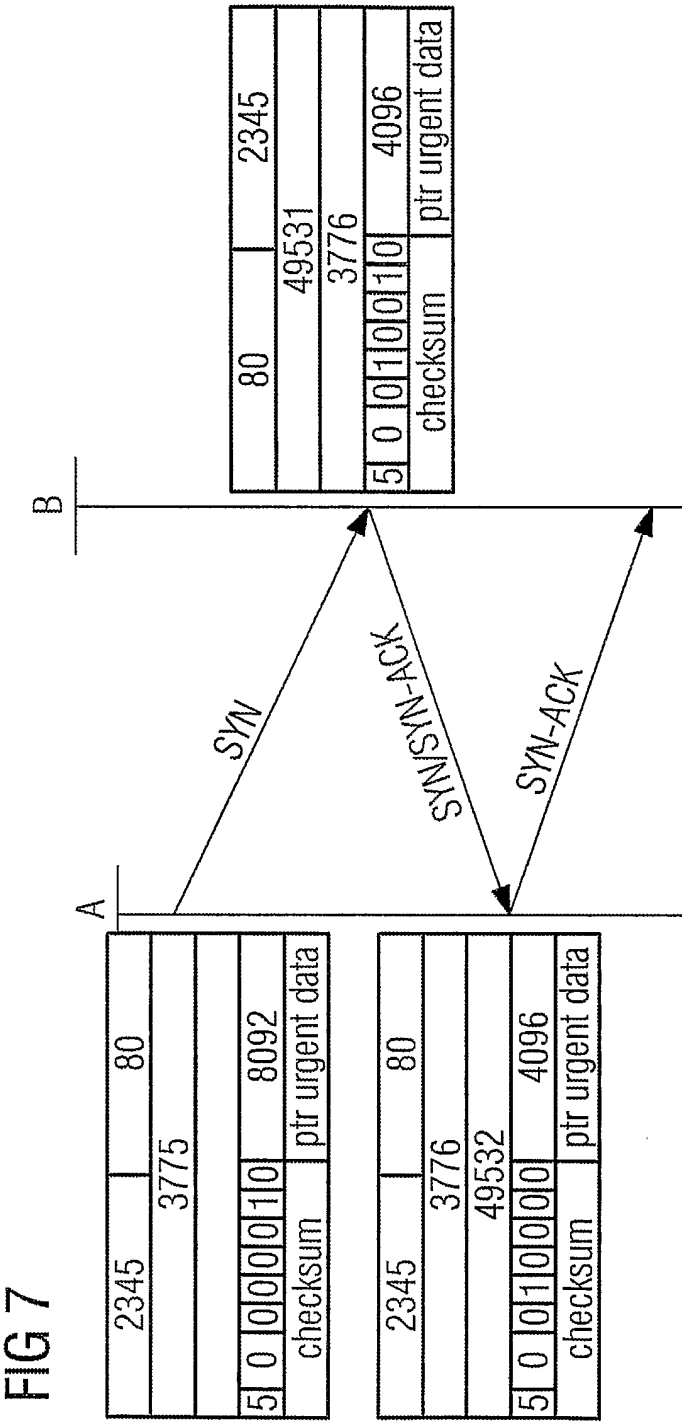

A
B
SYN
SYN/SYN-ACK
SYN-ACK
2345
80
3775
5 0 0 0 0 0 1 0
8092
checksum
ptr urgent data
2345
80
3776
49532
5 0 0 1 0 0 0 0
4096
checksum
ptr urgent data
80
2345
49531
3776
5 0 0 0 1 0 0 1 0
4096
checksum
ptr urgent data

FIG 8

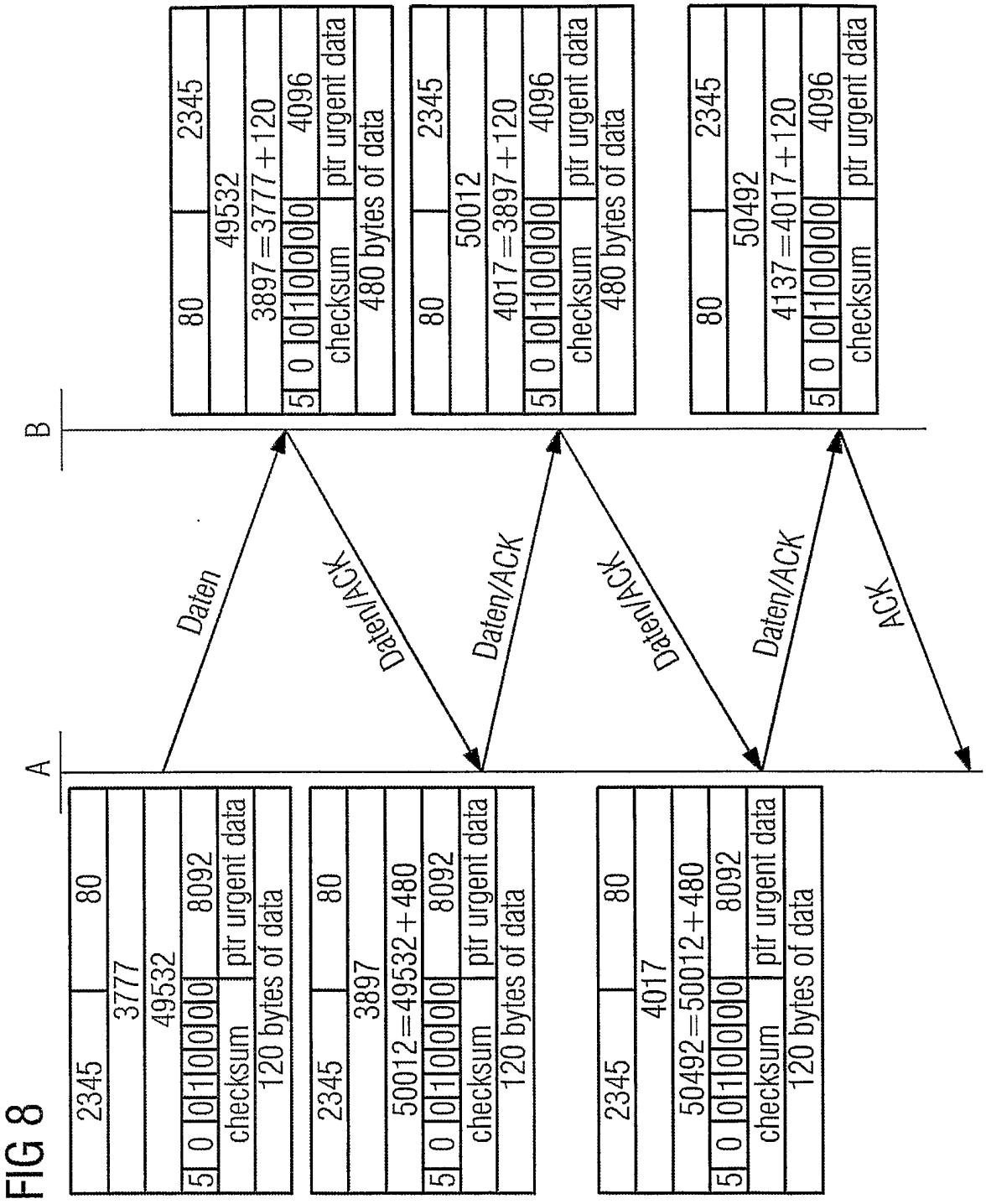

A
B
Daten
Daten/ACK
Daten/ACK
Daten/ACK
Daten/ACK
ACK
2345 | 80
3777
49532
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 8092
checksum | ptr urgent data
120 bytes of data
80 | 2345
49532
3897=3777+120
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4096
checksum | ptr urgent data
480 bytes of data
2345 | 80
3897
50012=49532+480
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 8092
checksum | ptr urgent data
120 bytes of data
80 | 2345
50012
4017=3897+120
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4096
checksum | ptr urgent data
480 bytes of data
2345 | 80
4017
50492=50012+480
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 8092
checksum | ptr urgent data
120 bytes of data
80 | 2345
50492
4137=4017+120
5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4096
checksum | ptr urgent data

FIG 9

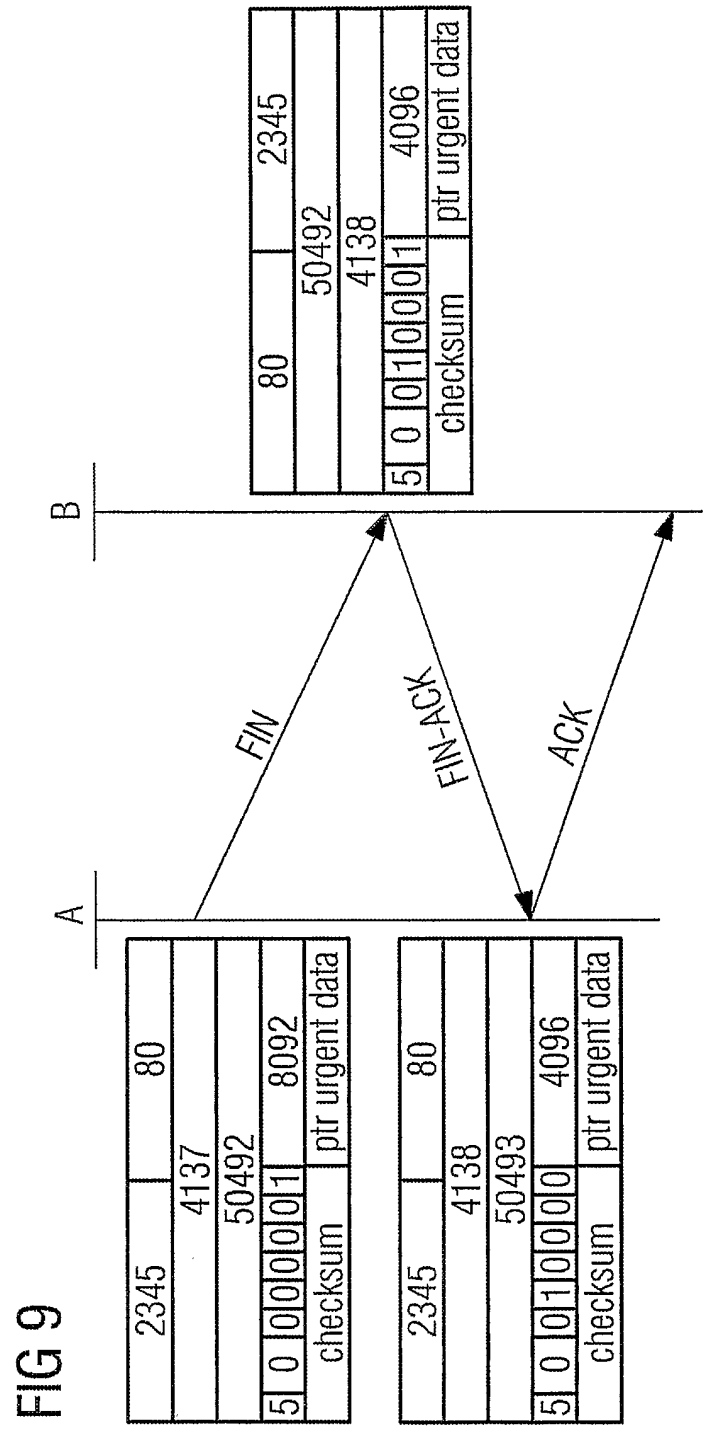

A
B
FIN
FIN-ACK
ACK
2345
80
4137
50492
5 0 0 0 0 0 1
8092
checksum
ptr urgent data
2345
80
4138
50493
5 0 0 1 0 0 0 0
4096
checksum
ptr urgent data
80
2345
50492
4138
5 0 0 1 0 0 0 1
4096
checksum
ptr urgent data

FIG 10

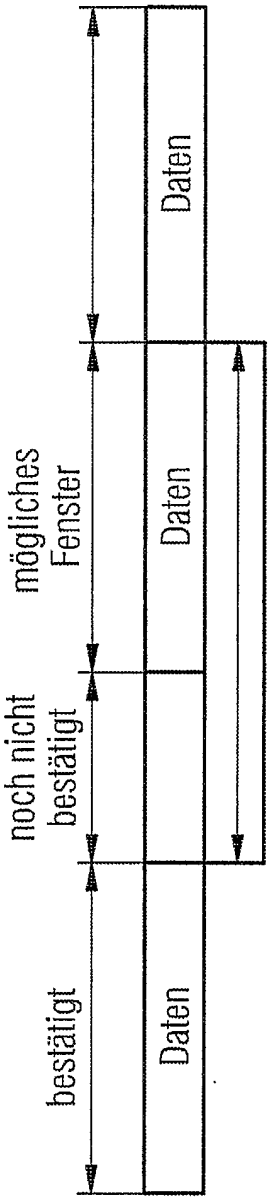

bestätigt
noch nicht bestätigt
mögliches Fenster
Daten
Daten
Daten

FIG 11

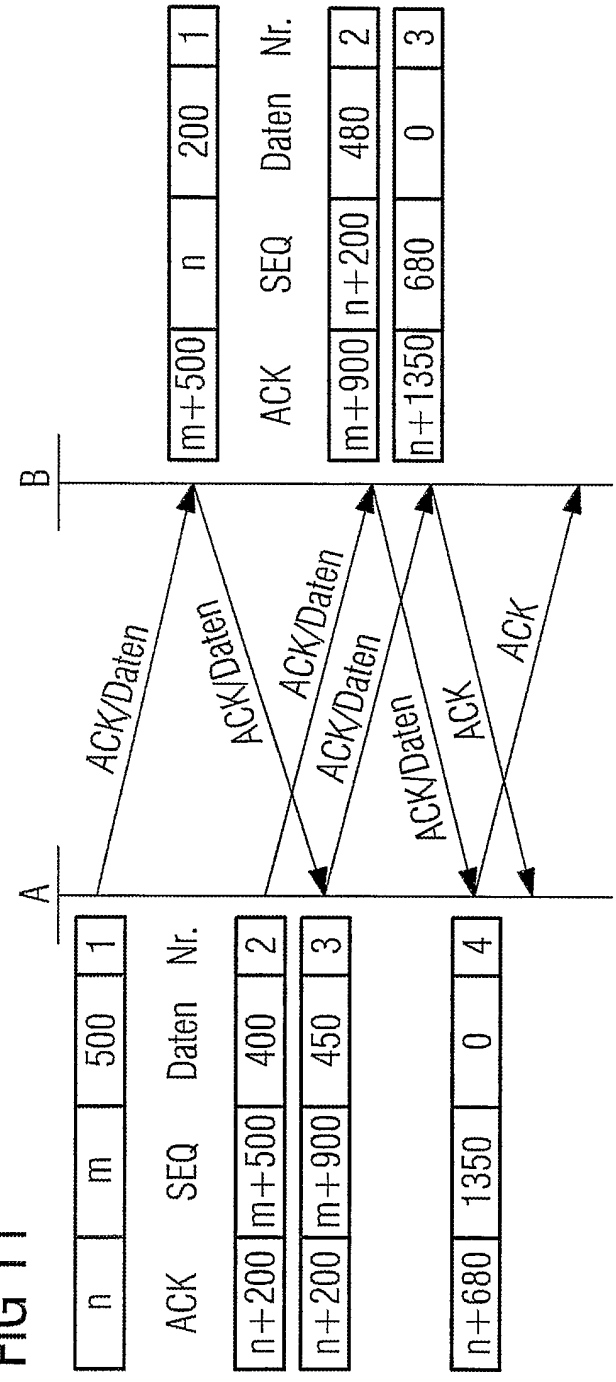

A
B
ACK/Daten
ACK/Daten
ACK/Daten
ACK/Daten
ACK/Daten
ACK
ACK
n | m | 500 | 1
n+200 | m+500 | 400 | 2
n+200 | m+900 | 450 | 3
n+680 | 1350 | 0 | 4
m+500 | n | 200 | 1
m+900 | n+200 | 480 | 2
n+1350 | 680 | 0 | 3
ACK SEQ Daten Nr.
ACK SEQ Daten Nr.

FIG 12

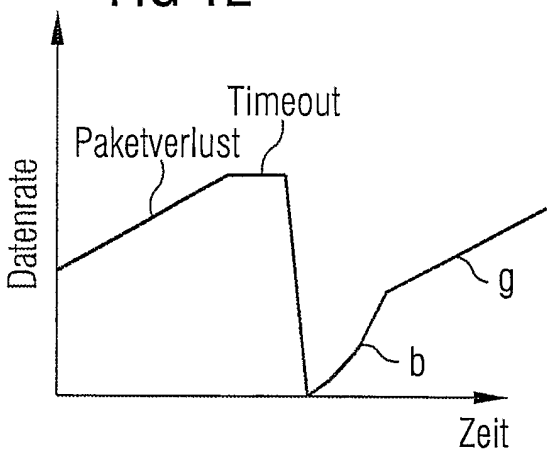


FIG 13

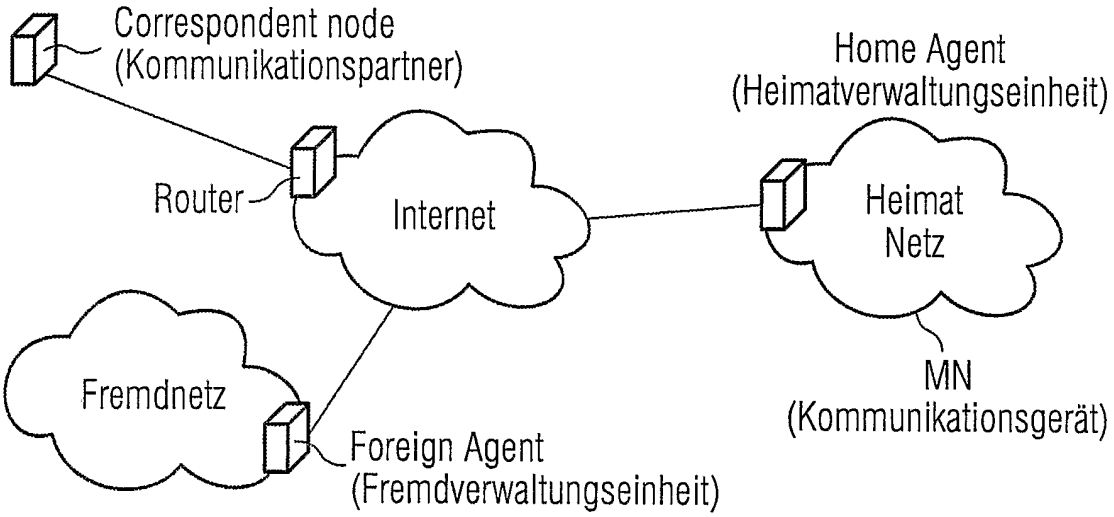


FIG 14

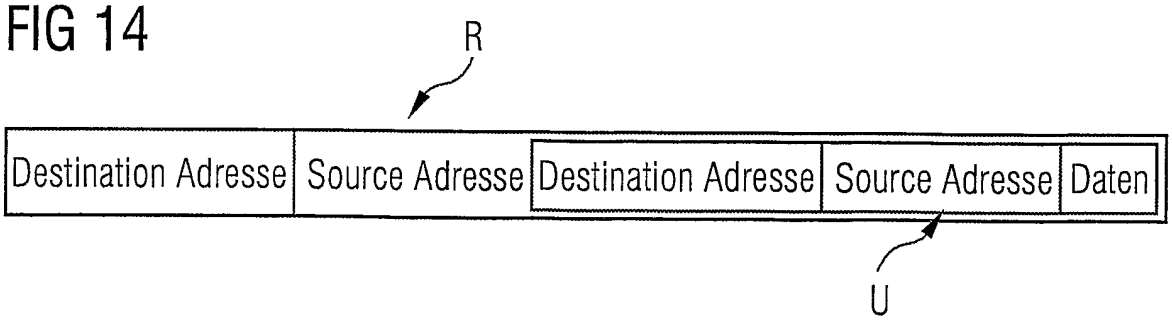


FIG 15

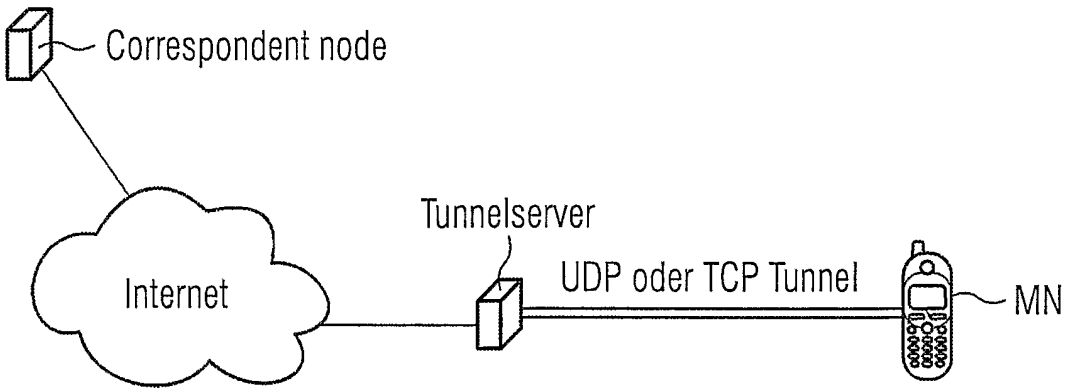

Correspondent node
Internet
Tunnelserver
UDP oder TCP Tunnel
MN

FIG 16

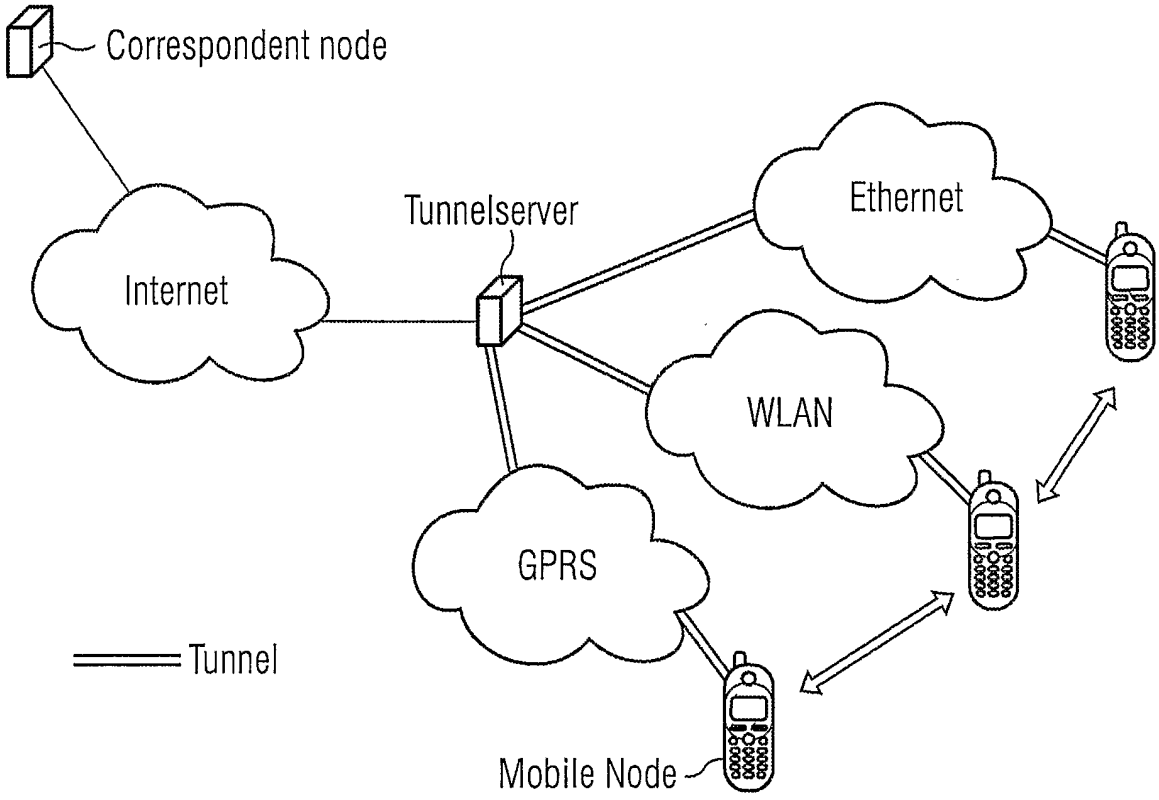

Correspondent node
Internet
Tunnelserver
Ethernet
WLAN
GPRS
Tunnel
Mobile Node

FIG 17

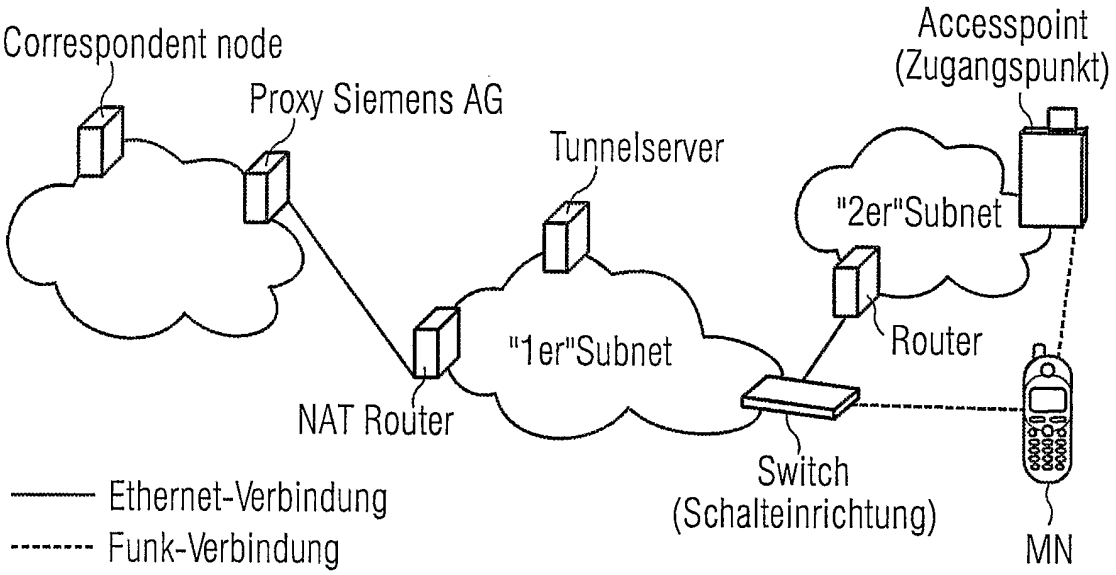

Correspondent node
Proxy Siemens AG
Tunnelserver
Accesspoint
(Zugangspunkt)
"2er"Subnet
"1er"Subnet
Router
NAT Router
Switch
(Schalteinrichtung)
MN
Ethernet-Verbindung
Funk-Verbindung

FIG 18

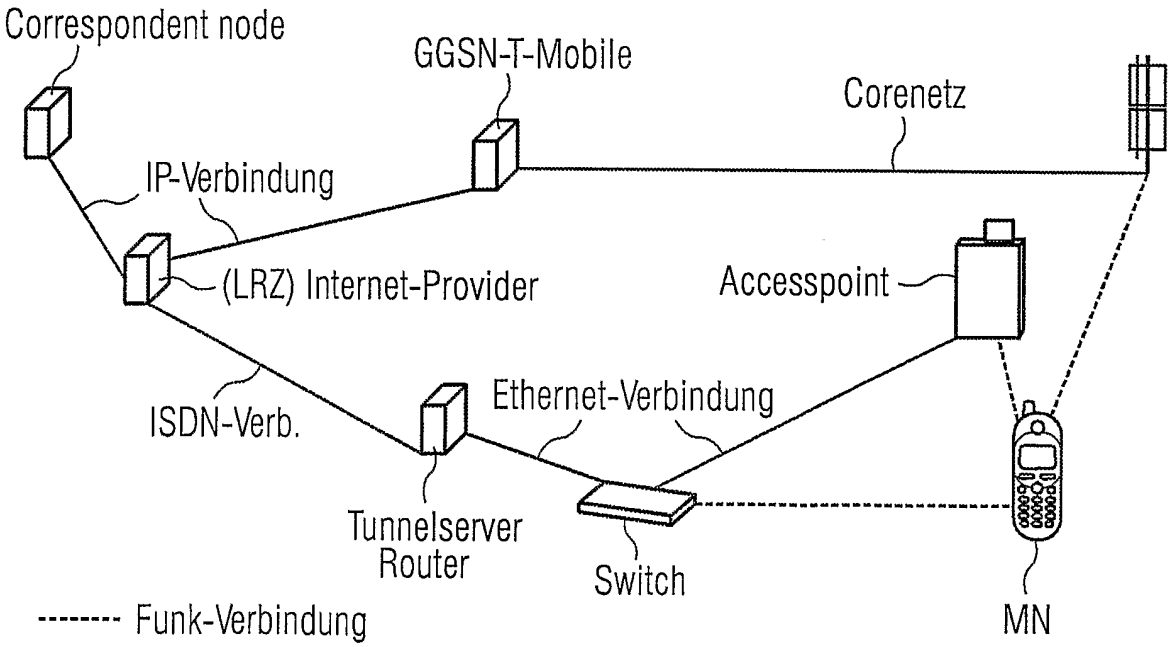

Correspondent node
GGSN-T-Mobile
Corenetz
IP-Verbindung
(LRZ) Internet-Provider
Accesspoint
ISDN-Verb.
Ethernet-Verbindung
Tunnelserver
Router
Switch
MN
Funk-Verbindung

FIG 19

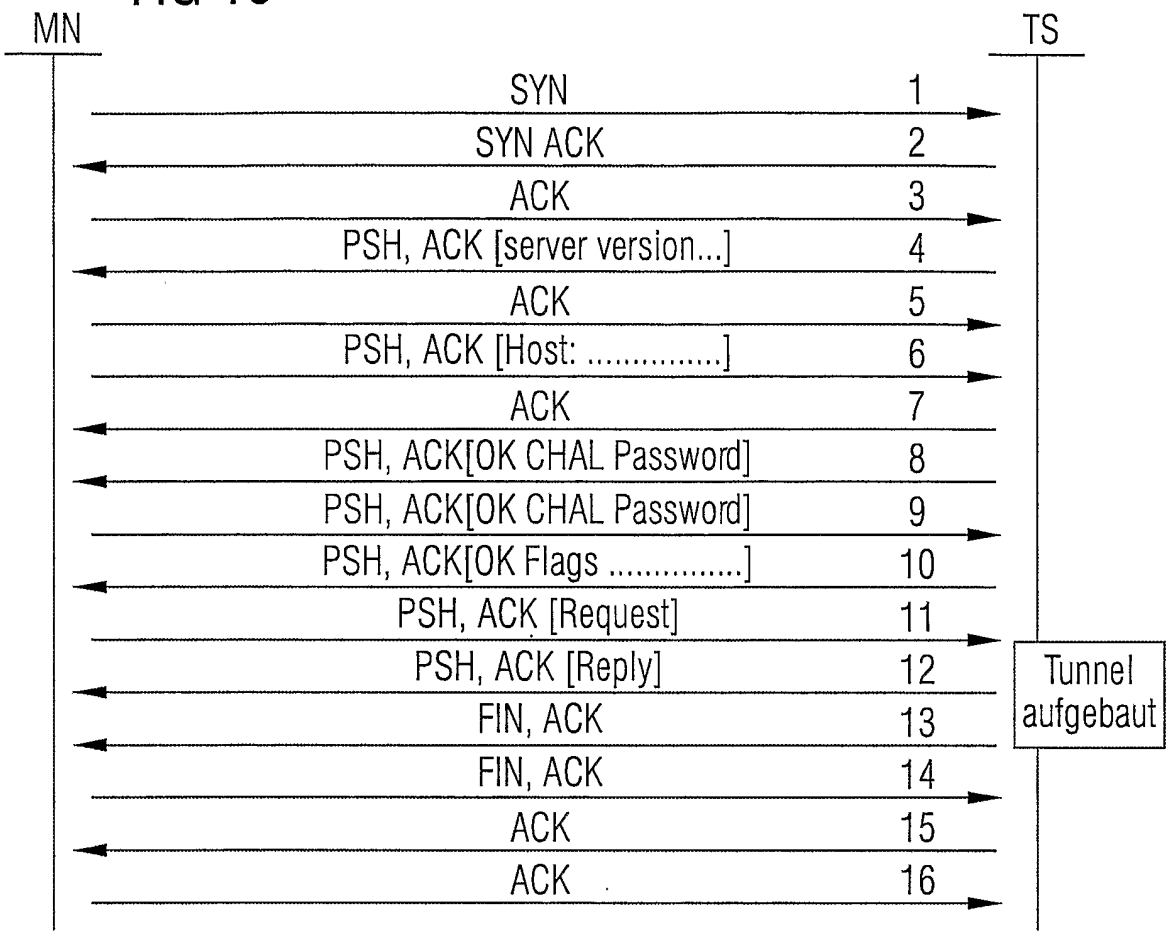

MN
TS
SYN 1
SYN ACK 2
ACK 3
PSH, ACK [server version...] 4
ACK 5
PSH, ACK [Host: .....] 6
ACK 7
PSH, ACK[OK CHAL Password] 8
PSH, ACK[OK CHAL Password] 9
PSH, ACK[OK Flags .....] 10
PSH, ACK [Request] 11
PSH, ACK [Reply] 12
FIN, ACK 13
FIN, ACK 14
ACK 15
ACK 16
Tunnel aufgebaut

FIG 20

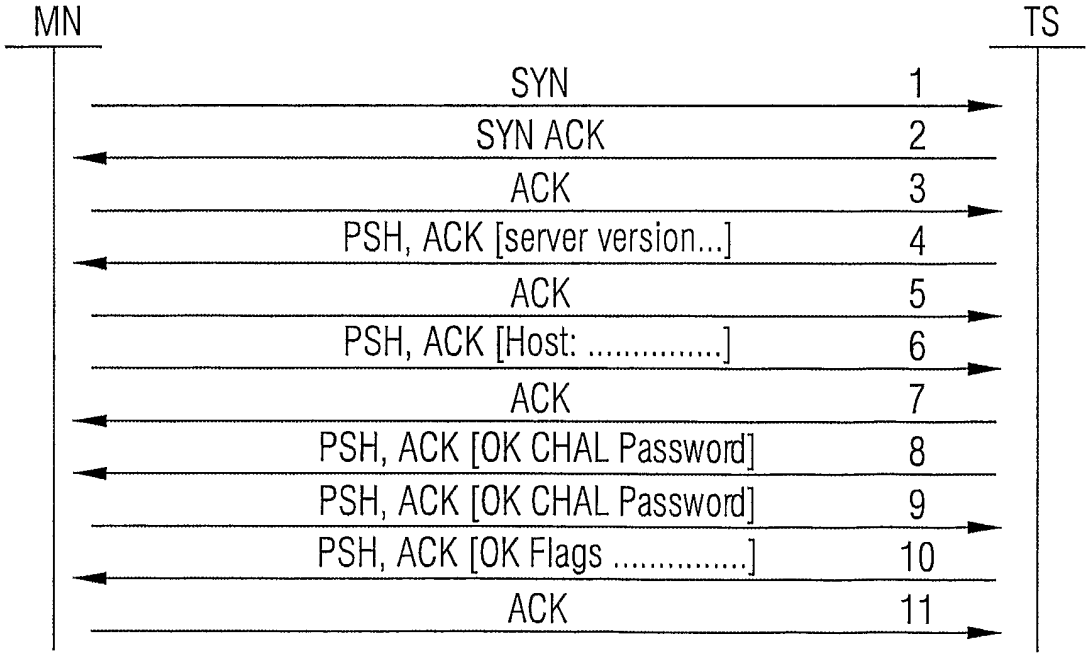

MN
TS
SYN 1
SYN ACK 2
ACK 3
PSH, ACK [server version...] 4
ACK 5
PSH, ACK [Host: .....] 6
ACK 7
PSH, ACK [OK CHAL Password] 8
PSH, ACK [OK CHAL Password] 9
PSH, ACK [OK Flags .....] 10
ACK 11

FIG 21

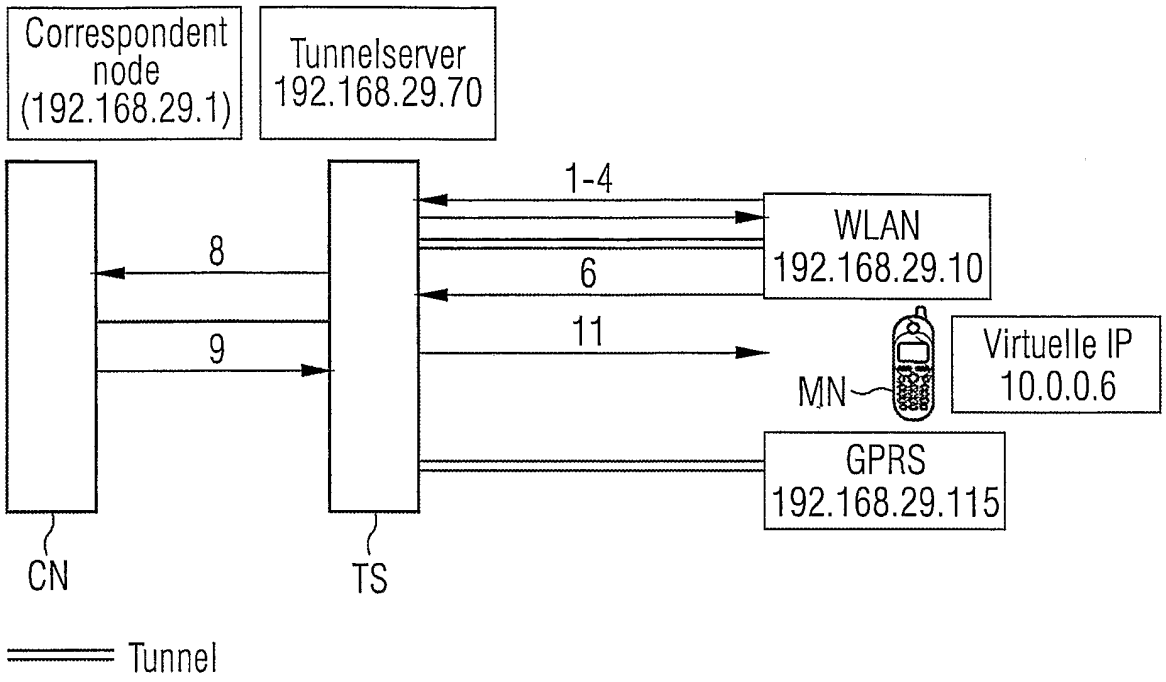

Correspondent node (192.168.29.1)
Tunnelserver 192.168.29.70
1-4
WLAN 192.168.29.10
8
6
9
11
MN
Virtuelle IP 10.0.0.6
GPRS 192.168.29.115
CN
TS
Tunnel

FIG 22

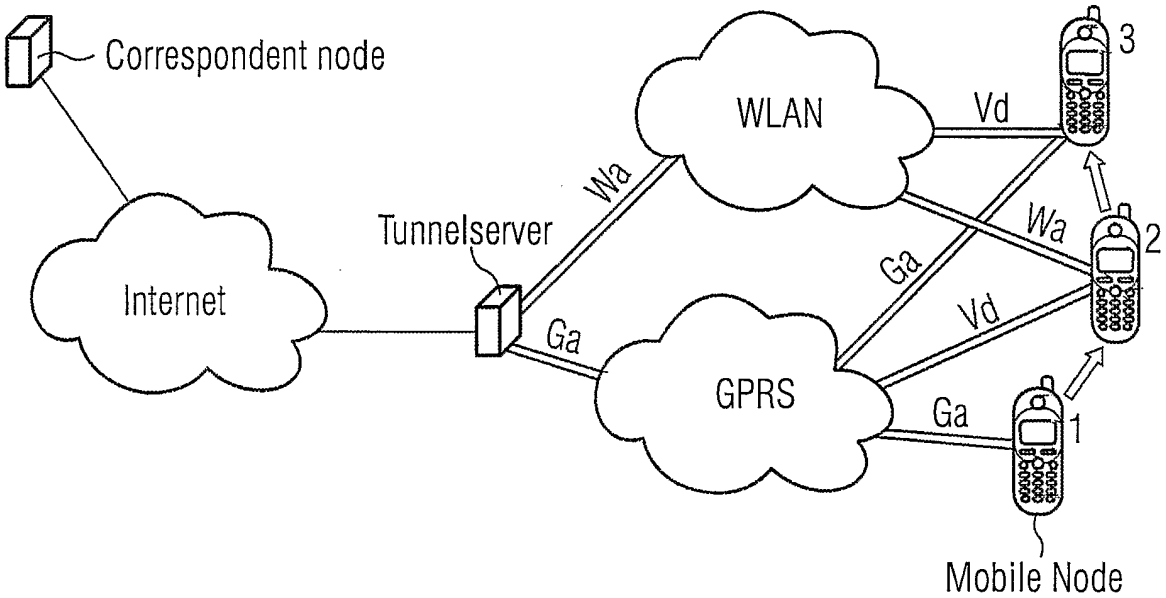
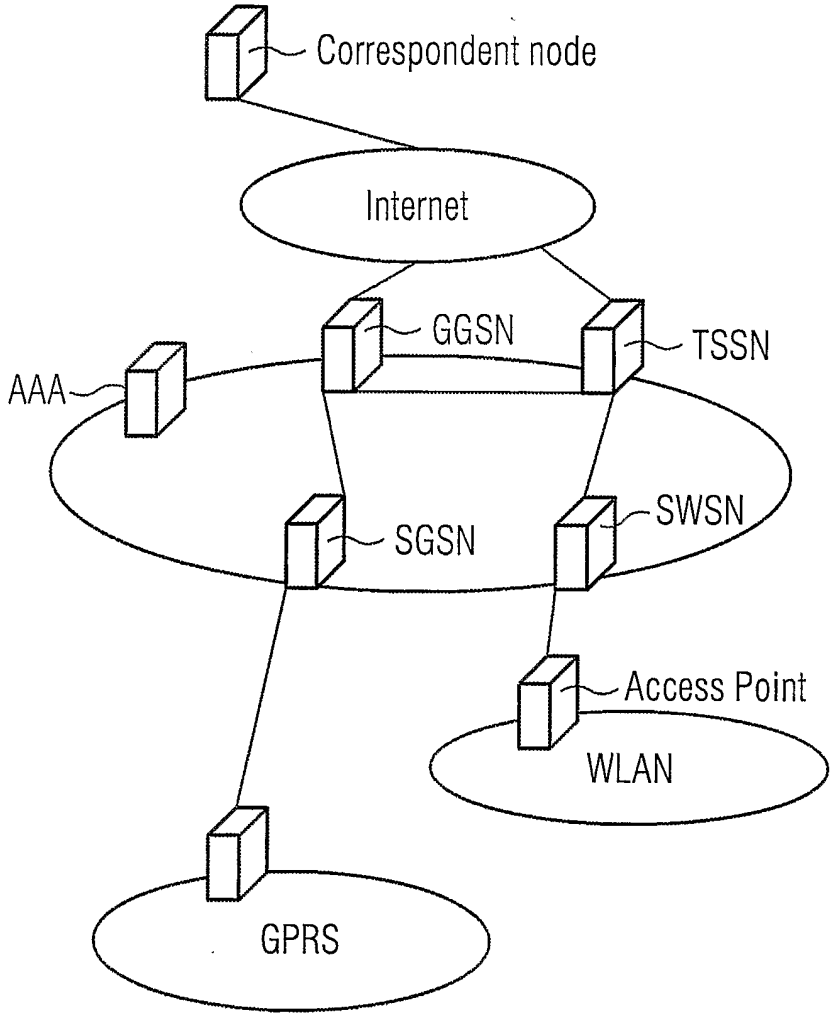

Correspondent node
Internet
Tunnelserver
WLAN
GPRS
Wa
Ga
Vd
3
2
1
Mobile Node

FIG 23


Correspondent node
Internet
GGSN
TSSN
AAA
SGSN
SWSN
Access Point
WLAN
GPRS