



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) DE 600 14 852 T2 2005.02.10

(12)

## Übersetzung der europäischen Patentschrift

(97) EP 1 153 490 B1

(21) Deutsches Aktenzeichen: 600 14 852.1

(86) PCT-Aktenzeichen: PCT/FI00/00107

(96) Europäisches Aktenzeichen: 00 905 087.3

(87) PCT-Veröffentlichungs-Nr.: WO 00/49748

(86) PCT-Anmeldetag: 14.02.2000

(87) Veröffentlichungstag

der PCT-Anmeldung: 24.08.2000

(97) Erstveröffentlichung durch das EPA: 14.11.2001

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 13.10.2004

(47) Veröffentlichungstag im Patentblatt: 10.02.2005

(51) Int Cl.<sup>7</sup>: H04J 3/26

H04L 12/66, H04L 12/56

(30) Unionspriorität:

990335 17.02.1999 FI

(84) Benannte Vertragsstaaten:

AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,  
LI, LU, MC, NL, PT, SE

(73) Patentinhaber:

Nokia Corp., Espoo, FI

(72) Erfinder:

PARANTAINEN, Janne, FIN-00500 Helsinki, FI;  
SHKUMBIN, Hamiti, FIN-02230 Espoo, FI

(74) Vertreter:

COHAUSZ & FLORACK, 40211 Düsseldorf

(54) Bezeichnung: HEADERKOMPRESSION IN ECHTZEITDIENSTEN

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 eingelebt, 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

**[0001]** Die vorliegende Erfindung betrifft die Datenübertragung und insbesondere ein Verfahren zum Versenden eines Datenpaketes von einem Komprimierer zu einem Dekomprimierer, wobei dieses Datenpaket einen Nachrichtenkopf mit Nachrichtenkopfdatenfeldern enthält, wobei eine Anzahl der Nachrichtenkopfdatenfelder, die während der Datenübertragung unverändert bleiben, in dem Dekomprimierer gespeichert werden. Das Verfahren umfasst das Senden eines Datenpaketes vom Komprimierer und Empfangen dieses Datenpaketes beim Dekomprimierer, wobei dieses Datenpaket Informationen über ein oder mehrere Nachrichtenkopfdatenfelder umfasst, die sich während der Datenübertragung ändern, sowie das Dekomprimieren des Nachrichtenkopfes mittels der gespeicherten Nachrichtenkopfdatenfelder und der empfangenen Informationen über das eine oder die mehreren Nachrichtenkopfdatenfelder, die sich während der Datenübertragung ändern. Die Erfindung betrifft außerdem Zugangsnetzwerkelemente, welche das erfindungsgemäße Verfahren implementieren.

**[0002]** Echtzeitdienste stellen eine aufkommende Gruppe von Technologien dar, die das Zusammenwirken von Sprache, Daten und Video über Paketvermittlungsnetze ermöglichen.

**[0003]** Neben der Standardisierung von Paketvermittlungsfunksystemen hat das Interesse an der Bereitstellung von Echtzeitdiensten, die ebenfalls über drahtlose Netze angeboten werden, zugenommen. Bei Echtzeitdiensten erfolgt die Paketübermittlung mittels einer Anzahl von Protokollen. Dies bringt eine große Protokollverwaltungsdatenlast mit sich und verursacht eine ineffiziente Bandbreitenausnutzung. Da die Übertragungsraten in drahtlosen Systemen begrenzt sind, bedeutet das Übermitteln großer Nachrichtenköpfe eine Vergeudung von Kapazität.

**[0004]** Um das Problem großer Nachrichtenköpfe in den Griff zu bekommen, wurde bereits eine Vielzahl von Nachrichtenkopfkomprimierungsschemata realisiert. Die Publikation "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links" von S. Casner und V. Jacobson, Internet Engineering Task Force, INTERNET-DRAFT, draft-ietf-avt-crtp-05.txt, Juli 1998, stellt effektive Nachrichtenkopfkomprimierungsalgorithmen vor, die eine Verringerung der Nachrichtenkopfgröße um eine ganze Größenordnung ermöglichen. Die vorgestellte Nachrichtenkopfkompression basiert auf der Tatsache, dass einige der Bytes in den IP-, UDP- und RTP-Nachrichtenköpfen über die Dauer der Verbindung hinweg unverändert bleiben. Nach dem Senden eines unkomprimierten Nachrichtenkopfes können diese Felder aus den komprimierten Nachrichtenköpfen, die folgen, ausgelassen werden. Des Weiteren wird an den sich verändernden Feldern Differenzkodierung verwendet, um ihre Größe zu verringern. Bei RTP-Nachrichtenköpfen ist die Differenz von Paket zu Paket oft konstant, und darum ist auch die Differenz zweiter Ordnung gleich Null.

**[0005]** Dadurch, dass man den Komprimierer und den Dekomprimierer sowohl den unkomprimierten Nachrichtenkopf als auch die Differenzen erster Ordnung im Sitzungszustand gemeinsam nutzen lässt, muss vor allem angezeigt werden, dass die Differenz zweiter Ordnung zwischen aufeinanderfolgenden Paketen Null ist. Es wird außerdem vorgeschlagen, dass eine Komprimiererimplementierung den Zustand für mehrere Sitzungskontexte aufrechterhalten könnte. Das Anwenden einer Hash-Funktion auf einige zuvor festgelegte Felder zum Indexieren einer Tabelle gespeicherter Sitzungskontexte und das Integrieren eines Sitzungskontextidentifikators in komprimierte Pakete würde es dem Dekomprimierer ermöglichen, eine Tabelle gespeicherter Sitzungskontexte direkt zu indexieren.

**[0006]** Das Dokument US 5,293,379 offenbart ein paketgestütztes Datenkomprimierungsverfahren, bei dem jedes Datenpaket umformatiert wird, indem sein Datenfeld einer ersten Paketregion zugeordnet wird und seine dynamischen Felder einer zweiten Paketregion zugeordnet werden. Es wird eine statische Tabelle gebildet, die statische Informationen aus der ersten Paketregion eines Datenpakets enthält, und es werden gemeinsame statische Feldinformationen aus der ersten Paketregion eines nachfolgenden Datenpakets ermittelt. Die gemeinsamen statischen Informationen werden so kodiert, dass ihre Datenlänge verkürzt wird.

**[0007]** Echtzeitdienste erlegen strikte Beschränkungen für Übertragungsverzögerungen auf, so dass normale Neuübertragungsverfahren (beispielsweise beim Transport Control Protocol, TCP, das im Allgemeinen für die Übertragung von IP-Paketen verwendet wird) im Allgemeinen nicht anwendbar sind. Ein Link bei den Echtzeitdiensten wird darum in der Praxis als ein Simplexlink angesehen.

**[0008]** In dem Beispiel nach dem Stand der Technik wird für Simplexlinks ein periodisches Refresh-Schema vorgeschlagen. Immer, wenn der Dekomprimierer einen Fehler in einem Paketstrom erkennt, so verwirft er alle Pakete in diesem Strom und nimmt die Dekomprimierung erst wieder auf, wenn ein richtig übermittelter unkom-

primierter Nachrichtenkopf (Refresh-Header) empfangen wird. Das bedeutet, dass nach einem Übertragungsfehler alle Pakete bis zum nächsten Refresh-Paket verloren sind. In Übertragungsverbindungen, wo Fehler nur relativ selten vorkommen, wirkt sich dies nicht sonderlich auf den Übertragungsdurchsatz aus. In einem Link, der mit einem hohen Risiko einer Vielzahl von Übertragungsfehlern behaftet ist, kommt es hingegen zu massiven Behinderungen. Das gilt besonders für drahtlose Übertragungen.

**[0009]** Das Dokument Handley, Mark: "GERM Generic RTP Multiplexing", Internet Engineering Task Force, 11. November 1998 (1998-11-11), Seiten 1–7, XP002139359, draft-ietf-avt-germ-00.ps, offenbart das Verringern der Nachrichtenkopfverwaltungsdatenlast mittels eines Multiplexing-Verfahrens, wobei mehrere voneinander unabhängige unterschiedliche RTP-Ströme zu einem einzelnen RTP-Paket kombiniert werden, obgleich das Dokument auch lehrt, dass ein solches Multiplexing im Allgemeinen vermieden werden sollte. Das Dokument offenbart eine RTP-Nachrichtenkopfkompression anhand geringstwertiger Bits für das SSRC-Feld von Paket-Nachrichtenköpfen aus den unterschiedlichen Quellen, d. h. aus den unterschiedlichen Strömen. Jedes gemultiplexte RTP-Paket hat einen kompletten RTP-Nachrichtenkopf, der die SSRC, die Folgenummer, den Zeitstempel usw. enthält, so dass die RTP-Pakete den gesamten Kontext enthalten, mit dem die Dekompression ausgeführt wird.

**[0010]** Die Publikation "Low-loss TCP/IP header compression for wireless networks" von Mikael Degermark, Mathias Engan, Björn Nordgren und Stephen Pink, Wireless Networks 3 (1997), 375–387, J. C. Balzer AG, Science Publishers, stellt Nachrichtenkopfkomprimierungsschemata für das UDP/IP- und TCP/IP-Protokoll vor, wo das Problem von Simplexlinks und verlustbehafteten Links behandelt wurde.

**[0011]** In dem vorgestellten System sendet der Komprimierer einen kompletten Nachrichtenkopf und einen Komprimierungssidentifikator, bei dem es sich um eine kleine einmalige Nummer handelt, die auch von komprimierten Nachrichtenköpfen getragen wird. Der komplette Nachrichtenkopf wird als ein Komprimierungszustand vom Dekomprimierer gespeichert. Die Komprimierungssidentifikatoren in komprimierten Nachrichtenköpfen dienen dem Nachschlagen des entsprechenden Komprimierungszustandes, der für die Dekomprimierung zu verwenden ist. Um eine falsche Dekomprimierung infolge eines widersprüchlichen Komprimierungszustandes zu vermeiden, werden einige weitere Mechanismen eingebaut. Jede Version des Komprimierungszustandes ist einer Generation zugeordnet, die durch eine Nummer dargestellt wird, welche von kompletten Nachrichtenköpfen, die diesen Komprimierungszustand installieren oder auffrischen, und in Nachrichtenköpfen, die mit ihm komprimiert wurden, getragen wird. Auf diese Weise ist der Dekomprimierer in der Lage zu erkennen, wenn sein Komprimierungszustand veraltet ist, indem er seine Generation mit der Generation in komprimierten Nachrichtenköpfen vergleicht.

**[0012]** Um lange Perioden mit verworfenen Paketen zu vermeiden, wo komplette Nachrichtenköpfe verloren gehen, und um andererseits möglichst hohe Kompressionsraten zu erreichen, beginnt der Komprimierer des Weiteren mit einem kurzen Intervall zwischen kompletten Nachrichtenköpfen, und der Refresh-Intervall wird mit jedem Refresh exponentiell erhöht, bis ein stationärer Refresh-Zeitraum erreicht ist (langsam Kompresionsbeginn). Im Gegenzug wird eine geringfügige Anpassung der Nachrichtenkopfkompression vorgeschlagen.

**[0013]** Obgleich das Verwenden einer Komprimierungszustandsgeneration das Erkennen widersprüchlicher Komprimierungszustände erleichtert, gehen die Pakete so oder so verloren, bis ein kompletter Nachrichtenkopf einen ordnungsgemäßen Komprimierungszustand installiert. Ein langsamer Komprimierungsbeginn hilft beim Auffinden einer brauchbaren Balance zwischen der Komprimierungsrate und einer akzeptablen Erholzeit, aber unter problematischen Übertragungsbedingungen verschlechtert sich die Komprimierungsrate in jedem Fall, und der Vorteil der Nachrichtenkopfkompression geht zum Teil verloren.

**[0014]** Es wurden nun ein Verfahren und Netzwerkelemente, mit denen das Verfahren implementiert wird, erfunden. Mit ihrer Hilfe lassen sich diese Probleme vermeiden, bzw. ihre Auswirkungen können verringert werden.

**[0015]** Gemäß einem ersten Aspekt der vorliegenden Erfindung wird ein Verfahren zum Übertragen eines Datenpaketes von einem Komprimierer zu einem Dekomprimierer bereitgestellt, wobei dieses Datenpaket einen Nachrichtenkopf mit Nachrichtenkopfdatenfeldern enthält, wobei eine Anzahl der Nachrichtenkopfdatenfelder, die während der Datenübertragung unverändert bleiben, in dem Dekomprimierer gespeichert werden, wobei das Verfahren folgendes umfasst:

Senden eines Datenpaketes vom Komprimierer und Empfangen dieses Datenpaketes beim Dekomprimierer, wobei dieses Datenpaket Informationen über ein oder mehrere Nachrichtenkopfdatenfelder umfasst, die sich

während der Datenübertragung ändern; und Dekomprimieren des Nachrichtenkopfes mittels der gespeicherten Nachrichtenkopfdatenfelder und der empfangenen Informationen über das eine oder die mehreren Nachrichtenkopfdatenfelder, die sich während der Datenübertragung ändern. Das Verfahren ist dadurch gekennzeichnet, dass nach der Sitzungseinrichtung im Zusammenhang mit einem sich verändernden Nachrichtenkopfdatenfeld lediglich ein komprimierter Wert vom Komprimierer versendet und vom Dekomprimierer empfangen wird, wobei dieser komprimierte Wert das Datenpaket in einer Komprimierungssequenz identifiziert und wobei der komprimierte Wert aus einer ersten Anzahl geringst-wertiger Bits des Nachrichtenkopfdatenfeldes besteht; dass in dem Dekomprimierer Kontextdaten gespeichert werden, die Informationen umfassen, mit denen der empfangene komprimierte Wert einer entsprechenden Komprimierungssequenz zugeordnet wird, wobei die Informationen entsprechend den empfangenen komprimierten Werten aktualisiert werden; und dass der komprimierte Wert und die Informationen der entsprechenden Komprimierungssequenz dazu verwendet werden, den komprimierten Wert in ein dekomprimiertes Nachrichtenkopfdatenfeld hinein abzubilden.

**[0016]** Der Vorteil der Erfindung basiert auf der Tatsache, dass die meisten der Felder in dem Netzwerkschichtpaket während einer ganzen Sitzung hindurch unverändert bleiben. Der Begriff "Netzwerkschicht" meint in diesem Zusammenhang Protokollsichten auf Paketdatennetzwerkebene, beispielsweise IP-, UDP- und RTP-Protokolle. Des Weiteren ist festzustellen, dass Änderungen in den Feldern, die sich von einem Paket zum anderen unterscheiden, vorhersagbar sind. Solche Felder werden dem Dekomprimierer in einer komprimierten Form zugesandt. Aufgrund der vorherigen Kenntnis der erwarteten Änderungen erzeugt der Dekomprimierer Kontextdaten (und behält diese bei), die anhand der Informationen von den Paketen, die beim Dekomprimierer empfangen werden, aktualisiert werden. Komprimierte Daten ordnen einer Änderung in einem dekomprimierten Wert in einer Gruppe aufeinanderfolgender Datenpakete eindeutig eine Komprimierungssequenz zu. In den Kontextdaten werden Informationen über eine oder mehrere Komprimierungssequenzen gespeichert. Diese Informationen stellen Mittel bereit, um die empfangenen komprimierten Daten einer richtigen Komprimierungssequenz zuzuordnen.

**[0017]** Durch Verwenden der empfangenen komprimierten Daten mit den entsprechenden Kontextdaten, die im Dekomprimierer gespeichert sind, können die komprimierten Daten während der gesamten Sitzung eindeutig einem kompletten Paketdatennachrichtenkopffeld auf der Dekomprimiererseite zugeordnet werden. Datenpakete, die Informationen tragen, welche nicht richtig einer der Komprimierungssequenzen der Kontextdaten, die im Dekomprimierer gespeichert sind, zugeordnet werden können, werden vorzugsweise schon auf der Komprimiererseite herausgefiltert.

**[0018]** Im Vergleich zu den Lösungen der Vergangenheit wird die Komprimierung bei dem erfindungsgemäß Verfahren erhöht, da auch die veränderlichen Felder, die mit der Identifizierung von Paketen zu tun haben, komprimiert werden können. Übertragungsfehler wirken sich nach wie vor nur auf die Übertragung einzelner Pakete aus, und Probleme bei der Übertragung eines Paketes pflanzen sich nicht in nachfolgende Pakete hinfort. Das Schema der Auffrischung kompletter Nachrichtenkopfinformationen kann so gestaltet werden, dass es in längeren Zeitabständen erfolgt, oder es kann ganz und gar weggelassen werden.

**[0019]** Weitere Aspekte der Erfindung sind in den unabhängigen Ansprüchen 9 und 11 dargestellt. Bevorzugte Ausführungsformen sind in den abhängigen Ansprüchen dargestellt.

**[0020]** Zum besseren Verständnis der vorliegenden Erfindung, und um zu zeigen, wie die vorliegende Erfindung in der Praxis realisiert werden kann, wenden wir uns nun den begleitenden Zeichnungen zu, die lediglich veranschaulichende Beispiele darstellen:

**[0021]** **Fig.** 1 stellt die Anhäufung der Nachrichtenkopfverwaltungsdatenlast durch die verschiedenen Schichten bei der Übertragung eines einzelnen Sprachdatenblocks **10** über eine drahtlose IP-Verbindung dar.

**[0022]** **Fig.** 2 veranschaulicht einige der Funktionselemente und die zugehörigen Protokollstrukturen eines GPRS-Netzwerks.

**[0023]** **Fig.** 3 veranschaulicht Felder der RTP-, UDP- und IP-Nachrichtenköpfe der Netzwerkschicht.

**[0024]** **Fig.** 4 veranschaulicht Funktionen in der Empfangseinheit gemäß der Erfindung.

**[0025]** **Fig.** 5 stellt ein Format für einen komprimierten Nachrichtenkopf dar, das in einer Ausführungsform der Erfindung verwendet wird.

**[0026]** **Fig.** 6 veranschaulicht Schritte der Ausführungsform des erfindungsgemäßen Verfahrens, wobei der abgekürzte Zeitstempel verwendet wird.

**[0027]** **Fig.** 7 stellt ein Beispiel eines Filteralgorithmus' gemäß der Erfindung dar.

**[0028]** **Fig.** 8 veranschaulicht das Format eines SNDCLP SN-UNITDATA PDU.

**[0029]** **Fig.** 9 veranschaulicht eine alternative Ausführungsform.

**[0030]** **Fig.** 10 veranschaulicht Blöcke, die für verschiedene Funktionen in einer erfindungsgemäßen Mobilstation zuständig sind.

**[0031]** Die Erfindung wird anhand einer Ausführungsform veranschaulicht, wo ein ITU-T-Sprachkodierer G.723.1, Internet-Protokoll Version 4 und das General Packet Radio System (GPRS) von ETSI verwendet werden, die dem Fachmann jeweils allgemein bekannt sind. Es ist festzustellen, dass für alle diese Systeme parallele oder entsprechende Technologien existieren und sich weiterentwickeln.

**[0032]** Folglich ist der Geltungsbereich des Schutzes nicht durch die Einzelheiten der Protokolle, die in der folgenden Beschreibung verwendet werden, beschränkt. Das im vorliegenden Text dargestellte Verfahren ist zwar auch in Festnetzen verwendbar, doch weil das Problem in der drahtlosen Kommunikation noch hinderlicher ist, wird in diesem Beispiel eine solche Struktur verwendet.

**[0033]** **Fig.** 1 stellt die Anhäufung der Nachrichtenkopfverwaltungsdatenlast durch die verschiedenen Schichten bei der Übertragung eines einzelnen Sprachdatenblocks **10** über eine drahtlose IP-Verbindung dar. Die schraffierten Blöcke stellen Nachrichtenköpfe dar, und die weißen Blöcke stellen die Nutzinformationen in einem Datenblock dar. Zuerst wird der Sprachdatenblock **10** in einem Real Time Protocol (RTP)-Paket **11** eingekapselt, das in ein User Data Protocol (UDP)-Paket **12** und weiter in ein Internet Protocol (IP)-Paket **13** gesteckt wird. Das IP-Paket **13** wird weiter unter Verwendung des Sub-Network Dependent Convergence Protocol (SNDCLP) **14** und des Logical Link Control Protocol in einen LLC-Block **15** eingekapselt, der in eine passende Anzahl RLC-Blöcke aufgeteilt wird, die jeweils einen separaten Nachrichtenkopf enthalten. Wie zu sehen ist, ist die kumulative Verwaltungsdatenlast sehr groß. Schon in der IP-Schicht beträgt die Protokollverwaltungsdatenlast **40** Bytes, und die Bandbreitenauslastung beträgt etwa 33%, wenn ein G723.1-Kodierer verwendet wird. Da durch die Protokollnachrichtenköpfe des drahtlosen Links noch mehr Verwaltungsdatenlast erzeugt wird, verschlimmert sich die Situation noch.

**[0034]** Bei dieser Ausführungsform erfolgen die Nachrichtenkopfkomprimierung und -dekomprimierung in der Protokollsicht, die für das Zugangsnetzwerk spezifisch ist – in diesem Fall ist das die SNDCLP-Schicht. In **Fig.** 2 sind einige der Funktionselemente und die zugehörigen Protokollstrukturen eines GPRS-Netzes veranschaulicht.

**[0035]** GPRS wird logisch zur allgemeinen GSM-Struktur mittels zweier Netzwerkelemente implementiert, nämlich einem Gateway GPRS Support Node (GGSN) und einem Serving GPRS Support Node (SGSN). GGSN ist der erste Paketdatennetz-Zugangspunkt in einem GSM-Netzwerk, das GPRS unterstützt.

**[0036]** Datenpakete, deren PDP-Adresse (Packet Data Protocol, beispielsweise IP oder X.25) einen GPRS-Teilnehmer anzeigen, werden zum GGSN geroutet. GGSN stellt Informationen bereit, die benötigt werden, um die Datenpakete zum momentanen Zugangsknoten-SGSN des Teilnehmers zu routen. GGSN kann die Standortdaten des Teilnehmers von einem GSM Home Location Register (HLR) abfragen. SGSN ist ein Zugangsknoten, der die Mobilstation (MS) bedient. Für eine GPRS-Verbindung stellt SGSN eine Mobilitätsverwaltungsfunktion für die MS und eine PDP-Funktion für das Paketdatennetz her, um die Datenpakete zum GGSN zu routen. SGSN und GGSN können in denselben physischen Knoten integriert werden, oder sie können sich in separaten Knoten befinden.

**[0037]** Die SNDCLP-Funktion des Zugangsnetzes stellt für die Netzwerksicht einen Service zum Übertragen einer Mindestmenge an Daten zwischen dem SGSN und der MS durch unterschiedliche Komprimierungstechniken bereit. GPRS stellt einen Verfahrensablauf, der in Verbindung mit der Service-Verhandlung implementiert wird, bereit, womit die MS und das SGSN sich auf einen Komprimierungsalgorithmus einigen, der während der Sitzung verwendet wird. Bei dem erfindungsgemäßen Verfahren werden Teile des Nachrichtenkopfes, die unverändert bleiben sollen, in den SNDCLP-Einheiten gespeichert. Als nächstes werden die Struktur und der Inhalt eines Netzwerksichtprotokoll-Nachrichtenkopfes eingehender untersucht.

**[0038]** In Fig. 3 sind Felder der RTP-, UDP- und IP-Nachrichtenköpfe der Netzwerkschicht gezeigt. Beim RTP zeigt Feld 310 die verwendete RTP-Version an und ändert sich während der Sitzung nicht. Feld 311 enthält das Stopbit und ändert sich nur, wenn zusätzliches Ausfüllen am Ende des Nachrichtenkopfes enthalten ist, beispielsweise für verschlüsselungsalgorithemische Zwecke in der Anwendungsschicht. Feld 312 zeigt an, ob auf einen festen Nachrichtenkopf eine Nachrichtenkopferweiterung folgt, und ändert sich während der Sitzung nicht. Feld 313 entspricht der CSRC-Zählung, die für Multiplexing-Zwecke benötigt wird, beispielsweise um anzuzeigen, wie viele Nutzer an den Nutzinformationen beteiligt sind. In vielen Fällen bleibt dieser Wert während der gesamten Sitzung unverändert. Feld 315 zeigt die Nutzinformationsart an und ist für einen Datentyp konstant. Im Allgemeinen sind die Zuführungsquelle und die Synchronisierungsquelle während der gesamten Übertragung über die Luftschnittstelle unverändert, so dass Feld 318 konstant bleibt.

**[0039]** Feld 314 enthält ein Markierungsbit, das optional zum Markieren wichtiger Ereignisse im Paketstrom verwendet wird, beispielsweise den Beginn eines Sprachsignalbündels oder ein letztes Paket in einem Videodatenblock. Wenn das Markierungsbit 314 verwendet wird, so muss es im komprimierten Nachrichtenkopf gesendet werden. Feld 316, das die Folgenummer anzeigt, und Feld 317, das den Zeitstempel anzeigt, ändern sich bei allen RTP-Paketen.

**[0040]** Beim UDP werden Feld 321, das die Ursprungsportnummer anzeigt, und Feld 322, das eine Zielportnummer anzeigt, dazu verwendet, um unterschiedliche Datenströme zu trennen, die derselben Anwendung zugeordnet sind.

**[0041]** Beispielsweise können Daten und Steuerungsinformationen der RTP-Schicht zu verschiedenen Ports geleitet werden, d. h. unterschiedliche Nutzinformationstypen können unterschiedliche UDP-Portpaare verwenden. Diese Felder bleiben unverändert, solange der gleiche Datentyp übertragen wird. Feld 323, das die UDP-Paketlänge anzeigt, bleibt unverändert, solange die Länge des darin befindlichen RTP-Paketes unverändert bleibt. In Fällen, wo die UDP-Länge während der gesamten Sitzung variabel ist (beispielsweise Übertragung von Video), muss die Länge des Paketes im komprimierten Nachrichtenkopf übertragen werden. Feld 324 entspricht der UDP-Prüfsumme und dient dem Erkennen der Fehler in der Übertragung.

**[0042]** Dieses Feld braucht nicht über den Übertragungslink übertragen zu werden, der einen leistungsstarken Fehlerschutzmechanismus oder Mittel zum Erkennen von Übertragungsfehlern besitzt (beispielsweise kleinere Protokollschnittschitprüfsummen). In dieser Ausführungsform kann die SNDNP-Dekomprimierungsfunktion beispielsweise die Prüfsumme aus den dekomprimierten Feldern errechnen und die errechnete Prüfsumme in dem dekomprimierten Paket verwenden.

**[0043]** Beim IP wird davon ausgegangen, dass Feld 331, das die IP-Version anzeigt, Feld 332, das die Nachrichtenkopflänge anzeigt, Feld 333, das den Servicetyp anzeigt, und Feld 334, das die Gesamtlänge des Paketes anzeigt, wenigstens so lange unverändert bleiben, wie Sprachdatenblöcke, die mit konstanter Bitrate kodiert werden, übertragen werden. Beim Feld 336, das die Flags anzeigt, kann man davon ausgehen, dass es unverändert bleibt, solange die Fragmentierung nicht verwendet wird.

**[0044]** Beim Feld 337, das den 13-Bit-Fragmentversatz enthält, wird davon ausgegangen, dass es unverändert bleibt, wie auch Feld 339, welches das Protokoll anzeigt. Feld 338, das die Lebensdauer anzeigt, und Feld 340, das die Prüfsumme anzeigt, können durch die SNDNP-Funktion definiert werden. Während der Datenübertragung bleiben IP-Ursprung und -Ziel unverändert, so dass davon ausgegangen wird, dass die Felder 341 und 342, welche die Ursprungs-IP-Adresse bzw. die Ziel-IP-Adresse anzeigen, unverändert bleiben. Das Identifizierungsfeld 335 wird hauptsächlich zur IP-Paketfragmentierung verwendet. Wenn keine Fragmentierung verwendet wird, so braucht dieses Feld nicht gesendet zu werden. Wenn Fragmentierung verwendet wird, so muss SNDNP zuerst das Paket neu aufbauen, bevor es komprimiert wird.

**[0045]** Die Felder, von denen davon ausgegangen wird, dass sie die meiste Zeit unverändert bleiben, werden zu einer Menge von Konstantfeldern gruppiert. In dieser Ausführungsform, und in Verbindung mit Sprachdatenblöcken von einem Konstantbitratenkodierer, umfasst die Menge folgende Felder: 310, 311, 312, 313, 315, 318, 321, 322, 331, 332, 333, 334, 335, 336, 337, 338, 339, 341, 342, 343. Diese Felder bilden einen Komprimierungszustand, der wenigstens auf der Empfängerseite (der Dekomprimierungsseite) des Links aufrechterhalten wird.

**[0046]** Wie bereits zuvor erwähnt, und im weiteren Zusammenhang mit den Konstantfeldern, kann eine zweite Menge an Feldern, deren Inhalt aus den empfangenen Informationen hergeleitet werden kann, in dem komprimierten Nachrichtenkopf weggelassen werden. Solche Felder in den vorgestellten Ausführungsformen sind

die Felder **324** und **340**, welche die Prüfsummen umfassen, die zum Überprüfen der Validität der Pakete verwendet werden. Diese Summen können im Dekomprimierer anhand der dekomprimierten Datenfelder berechnet werden. Die Validität der Pakete kann anhand der Prüfsummen der niedrigeren Ebene bestimmt werden, beispielsweise Dateneinheiten auf Zugangsnetzebene.

**[0047]** Die erfindungsgemäße Lösung zum Verwalten der Felder, die sich bei jedem Paket ändern, ist in allgemeiner Form in **Fig. 4** veranschaulicht, wo Funktionen in der Empfängereinheit – in dieser Ausführungsform die SGSN (auch: Dekomprimierer) – gezeigt sind. In Verbindung mit der Sitzungseinrichtung werden Informationen, die für den Komprimierungszustand benötigt werden, im Dekomprimierer empfangen (beispielsweise ein kompletter Nachrichtenkopf). Um sicherzustellen, dass ein richtiger Komprimierungszustand verwendet wird, kann in die Sitzungseinrichtungssignalisierung ein Bestätigungsverfahren eingebaut werden. Bei Schritt **40** wird der Komprimierungszustand SoC (State of Compression) im Dekomprimierer gespeichert. Bei Schritt **41** wird ein Sitzungskontext, der einen oder mehrere Kontextwerte  $C_j$  umfasst, die sich jeweils auf eine bestimmte Komprimierungssequenz beziehen, im Dekomprimierer initialisiert. Es wird ein Paket von der Sendeeinheit, in diesem Fall die MS (auch: Komprimierer), empfangen (Schritt **42**). Das Paket umfasst ein komprimiertes Datenfeld Dcom. Falls mehr als ein Kontextwert gespeichert wird (Maximalwert für  $j > 1$ ), wird eine Menge an Entscheidungsregeln  $D_m$  für das Zuordnen eines empfangenen Dcom zu einem entsprechenden Kontextwert  $C_j$  definiert. Es wird eine Entscheidungsregel  $D_m$ , die zu dem empfangenen Dcom passt, ermittelt (Schritt **43**), und es wird der dekomprimierte Wert  $D_{full}$  aus dem Wert des empfangenen IDcom und dem Kontextwert  $C_m$  des ermittelten  $D_m$  abgeleitet (Schritt **44**). Entsprechend der erwarteten Entwicklung der Felder werden kein Kontextwert  $C_m$ , ein Kontextwert  $C_m$  oder mehrere Kontextwerte  $C_m$  aktualisiert (Schritt **45**), um die Kontextdaten zu verwalten. Dieses Verfahren wird für neue Pakete während der gesamten Datenübertragungssitzung angewendet.

**[0048]** Die sich verändernden Felder, die in dieser Ausführungsform zu übertragen sind, sind Feld **316**, das die RTP-Folgenummer anzeigt, Feld **317**, das den RTP-Zeitstempel anzeigt, und Feld **335**, das die IP-Identifikation anzeigt. Unter Berücksichtigung der Tatsache, dass die Inkrementa in diesen Feldern während der gesamten Sitzung im Allgemeinen unverändert bleiben, könnte man ein Deltakodierungsschema (unter Bezugnahme auf ein zuvor übertragenes Paket) vorschlagen. In jedem Fall wird, um die oben beschriebenen Probleme zu vermeiden, eine unabhängige Identifikation für jedes Netzwerkschichtpaket, das komprimiert wird, erzeugt.

**[0049]** In der ersten Ausführungsform werden Nachrichtenkopffelder zu abgekürzten Feldern komprimiert und über den Link übertragen. Die Länge eines abgekürzten Feldes wird so gewählt, dass Informationen übertragen werden, die eine korrekte Identifizierung des Paketes während einer Komprimierungssequenz ermöglichen – ein Intervall, der im Allgemeinen kürzer ist als die Sitzung.

**[0050]** Eine kurzzeitige Identifizierung, die durch die abgekürzten Felder bereitgestellt wird, welche zu einem längeren Kontext kombiniert werden, der im Dekomprimierer aufrecht erhalten wird, stellt eine einheitliche Identifikation von Paketen während der gesamten Datenübertragungssitzung bereit und ermöglicht somit eine unzweideutige Zuordnung von komprimierten Nachrichtenkopffeldern zu kompletten Nachrichtenkopffeldern während einer gesamten Datenübertragungssitzung.

**[0051]** Als ein Beispiel für eine solche Anordnung wird ein Fall eines abgekürzten Zeitstempels vorgestellt. **Fig. 5** stellt ein Format für einen komprimierten Nachrichtenkopf dar, das in dieser Ausführungsform verwendet wird. Feld **51** zeigt den Typ  $T$  des komprimierten Pakets an. Wenn  $T = 0$ , so ist das letzte 8-Bit-Byte **56** nicht enthalten, und die letzten sechs Bits **53** des ersten 8-Bit-Bytes werden auf Null gesetzt und für einen anderen Zweck verwendet, beispielsweise als CRC-Check, oder sie werden für einen abgekürzten Zeitstempel verwendet. Wenn  $T = 1$ , so muss der komprimierte Nachrichtenkopf das Längen-Oktett enthalten, und die Bits **53** und das letzte 8-Bit-Byte **56** dienen dazu, die Länge der RTP-Nutzinformationen anzuzeigen. Diese Längeninformation wird bei den Bit-Strömen benötigt, wo sich die Paketlänge ändern kann, beispielsweise bei Video-Bitströmen. Das Feld **52** zeigt das Markierungsbit des RTP-Nachrichtenkopfes an, wie oben erläutert. Der abgekürzte Zeitstempel ist in dieser Ausführungsform ein 16-Bit-Feld, das die 16 geringstwertigen Bits des RTP-Zeitstempels anzeigt. Die Kontextdaten umfassen die 16 höchstwertigen Bits des RTP-Zeitstempels und werden mindestens auf der Dekomprimiererseite des Links aufrechterhalten.

**[0052]** Das Ablaufdiagramm von **Fig. 6** veranschaulicht die Schritte der Ausführungsform des erfindungsgemäßen Verfahrens, wobei der abgekürzte Zeitstempel verwendet wird. Bei Schritt **61** wird der Komprimierungszustand empfangen, wie in diesem Fall in einem kompletten Zeitstempel  $TS_{full}$ , der am Anfang einer Sitzung empfangen wird. Am Anfang der Sitzung werden die Kontextdaten initialisiert, in diesem Fall  $TS_{mem1}$ , das die

16 geringstwertigen Bits des ursprünglichen Zeitstempels umfasst, und Tsmem2, das die 16 höchstwertigen Bits des ursprünglichen Zeitstempels umfasst (Schritt 62). Bei Schritt 63 wird ein neuer abgekürzter Zeitstempel TSabb empfangen, der die 16 geringstwertigen Bits des ursprünglichen Zeitstempels eines neuen komprimierten Netzwerkschichtdatenpaketes trägt. Der neue abgekürzte Zeitstempel wird mit dem gespeicherten Wert Tsmem1 in dem Dekomprimierer verglichen. Wie später zu sehen sein wird, umfasst der Wert von Tsmem1 die 16 geringstwertigen Bits des größten Zeitstempels, der bis dahin empfangenen wurde.

**[0053]** Wenn der neue abgekürzte Zeitstempel TSabb größer ist als der gespeicherte Wert Tsmem1, so wird des Weiteren geprüft, ob der neue abgekürzte Zeitstempel TSabb größer ist als die Summe des gespeicherten Wertes Tsmem1 und eines zuvor festgelegten Wertes  $\Delta$ . Der Wert  $\Delta$  repräsentiert eine maximale Änderung bei den geringstwertigen Bits, die so interpretiert werden kann, dass sie durch ein erwartetes Phänomen hervorgerufen wurde, wie beispielsweise Stille, verlorene Pakete oder Pakete, die in einer geringfügig falschen Reihenfolge ankommen. Wenn die Zahl, die durch die 16 geringstwertigen Bits des Zeitstempels repräsentiert wird, das Maximum erreicht hat, so wird sie zurückgesetzt und beginnt wieder beim kleinsten Wert (Komprimierungssequenz). Wenn ein Paket verspätet beim Komprimierer ankommt, so ist es möglich, dass der gespeicherte Wert Tsmem1 sich bereits herumgelegt hat, so dass der Wert des empfangenen abgekürzten Zeitstempels TSabb erheblich größer ist als Tsmem1. Durch Definieren eines zweckmäßigen Wertes für  $\Delta$  können solche Fälle in dem Strom empfangener Pakete entdeckt werden. Wenn der empfangene abgekürzte Zeitstempel TSabb größer ist als Tsmem1, aber die Differenz nicht zu groß ist (kleiner als  $\Delta$ ), so kann der Zeitstempel rekonstruiert werden, indem man die 16 höchstwertigen Bits, die in dem Wert Tsmem2 gespeichert sind, verwendet und sie mit den 16 geringstwertigen Bits kombiniert, die vom Komprimierer erhalten werden (Schritt 64). Der empfangene abgekürzte Zeitstempel TSabb ist der größte abgekürzte Zeitstempel, der bis dahin empfangenen wurde, und er wird somit als Tsmem1 gespeichert. Wenn der empfangene abgekürzte Zeitstempel TSabb größer ist als Tsmem1 und die Differenz größer ist als  $\Delta$ , so wird davon ausgegangen, dass TSabb zu spät eintraf und Tsmem1 sich bereits herumgelegt hat.

**[0054]** Für diese Fälle wird ein weiterer Kontextdatenwert Tsmem3, der sich auf die vorherige Komprimierungssequenz bezieht, in den Kontextdaten verwaltet. Tsmem3 umfasst den zuvor gespeicherten Wert von Tsmem2. Die Rekonstruktion des Zeitstempels erfolgt nun dadurch, dass man die 16 höchstwertigen Bits, die in dem Wert Tsmem3 gespeichert sind, verwendet und sie mit den 16 geringstwertigen Bits kombiniert, die vom Komprimierer erhalten werden. Es erfolgen keine Aktualisierungen der Werte von Tsmem1, Tsmem2 oder Tsmem3.

**[0055]** Wenn der Wert des empfangenen abgekürzten Zeitstempels TSabb kleiner ist als Tsmem1, so wird geprüft, ob die Differenz größer ist als  $\Delta$ . Wenn dies der Fall ist, so hat der abgekürzte Zeitstempel, der die 16 geringstwertigen Bits des Zeitstempels umfasst, das Maximum erreicht, hat sich herumgelegt, und der gespeicherte Wert Tsmem2 muss auf den nächstmöglichen Wert inkrementiert werden (Schritt 67). Anschließend kann der Zeitstempel rekonstruiert werden, und der gespeicherte Wert Tsmem1 kann aktualisiert werden, wie bei den Schritten 64 und 65 erläutert. Nehmen wir zum Beispiel einen Fall, wo die gespeicherten Zeitstempelwerte Tsmem1 = FF FF (hex), Tsmem2 = 02 FF (hex) und  $\Delta$  = 0FFF (hex) sind und der empfangene abgekürzte Zeitstempelwert TSabb = 00 0A (hex) ist. Nun ist der abgekürzte Zeitstempelwert 00 0A kleiner als der gespeicherte Zeitstempelwert FF FF, die Differenz ist größer als  $\Delta$ , und somit müssen die 16 höchstwertigen Bits 02 FF von Tsmem2 um Eins auf einen Wert von 03 00 aktualisiert werden. Der resultierende Zeitstempelwert ist somit 03 00 00 0A. Wenn die Differenz kleiner ist als  $\Delta$ , so bedeutet dies, dass das Paket zur aktuellen Sequenz gehört, aber verspätet eintrifft. In einem solchen Fall kann der Zeitstempel dadurch rekonstruiert werden, dass man die 16 höchstwertigen Bits, die in dem Wert Tsmem2 gespeichert sind, verwendet und sie mit dem abgekürzten Zeitstempel TSabb kombiniert, der vom Komprimierer erhalten wurde. Da dies nicht der größte abgekürzte Zeitstempel ist, der bis dahin empfangen wurde, wird der Wert Tsmem1 nicht aktualisiert. Dieses Verfahren wird angewendet, solange weitere neue Pakete eintreffen.

**[0056]** Diese Idee kann auch auf andere Felder angewendet werden.

**[0057]** Wir wollen beispielsweise einmal eine komplette Folgenummer des RTP-Nachrichtenkopfes untersuchen. Wenn die ursprünglichen Folgenummern (in binärer Form) 00010000, 000100001, 00010010 lauten, so lauten die abgekürzten Folgenummern, die zum Dekomprimierer übertragen werden, 0000, 0001, 0010. Im Dekomprimierer werden Kontextdaten verwaltet, die wenigstens die aktuellen höchstwertigen Bits umfassen. Mit einer ähnlichen Art von Entscheidungsregeln können die komprimierten Daten der Komprimierungssequenz bzw. den Komprimierungssequenzen im Dekomprimierer zugeordnet und in ein komplettes Nachrichtenkopffeld hinein abgebildet werden.

**[0058]** Es ist auch eine andere Art der Beziehung zwischen den komprimierten Daten und den Inkrementen, die in dem Dekomprimierer verwendet werden, möglich. Wenn beispielsweise bekannt ist, dass der Zeitstempel sich für jedes Paket um 240 ändern kann, so kann ein Inkrement von Eins im Komprimierer einem Inkrement von 240 im Dekomprimierer zugeordnet werden. Dabei gilt (Kompressionswert → dekomprimierter Zeitstempel): 0001 → 240, 0010 → 480, 0011 → 720, 0100 → 960, usw.

**[0059]** Wie gezeigt, werden die Kontextdaten gemäß den Informationen in dem empfangenen abgekürzten Feldern aktualisiert. Der Grad an Abkürzung, d. h. die Menge an Bits, die für die Darstellung des abgekürzten Feldes verwendet wird, wirkt sich auf die Rate aus, mit der die Kontextinformationen im Dekomprimierer aktualisiert werden. Je kürzer beispielsweise die Komprimierungssequenz ist, desto öfter muss der Zeitstempelwert TSmem2, der die höchswertigen Bits des Zeitstempels speichert, aktualisiert werden. Selbst wenn einige Pakete verloren gehen oder ihre Reihenfolge sich geringfügig verändert, sorgen die bereits angesprochenen Validitätsvergleiche dafür, dass die Daten richtig regeneriert werden können. Bei drahtlosen Verbindungen führt das Verlieren einer langen Paketsequenz im Allgemeinen zu einem Abbruch des laufenden Telefonats.

**[0060]** Solange die Verbindung aufrechterhalten werden kann, ist es somit auch möglich, im Dekomprimierer übereinstimmende Kontextinformationen mit einem brauchbaren Komprimierungsgrad beizubehalten. Beispielsweise ist es mit 6 Bits möglich, 64 Pakete voneinander zu unterscheiden. Es ist unwahrscheinlich, dass 64 aufeinanderfolgende Pakete verloren gehen und die Verbindung dennoch aufrecht erhalten bleibt, so dass das erfindungsgemäße Verfahren solange einwandfrei funktioniert, solange die Verbindung aufrecht erhalten werden kann.

**[0061]** Für Pakete, welche die Komprimierung stören könnten, beispielsweise solche, die sehr spät beim Komprimierer eintreffen und dadurch möglicherweise die Reihenfolge der Aktualisierung der Komprimierungsinformationen durcheinanderbringen, wird im Komprimierer vorzugsweise eine weitere Funktion bereitgestellt. Der Empfang eines verspäteten Pakets, bei dem festgestellt wird, dass das abzukürzende Feld das Risiko in sich birgt, einen Fehler in den Komprimierungsinformationen zu verursachen, führt bereits auf der Komprimiererseite zu einer Korrekturmaßnahme. Beispielsweise können solche Pakete allesamt verworfen werden. Beispielsweise können Pakete, die verspätet eintreffen, aber zur vorherigen Komprimierungssequenz gehören, mittels des Kontextwertes TSmem3 wiederhergestellt werden, wie in Verbindung mit **Fig. 6** erläutert wurde. Ein Paket, das zu einer Komprimierungssequenz gehört, die noch vor der vorangegangenen Komprimierungssequenz liegt, würde nicht mehr richtig wiederhergestellt werden, so dass solche Pakete vorzugsweise bereits im Komprimierer verwaltet werden. Das Ablaufdiagramm von **Fig. 7** stellt ein Beispiel eines solchen Filteralgorithmus' dar, der dem erfindungsgemäßen Verfahren auf der Komprimiererseite hinzugefügt werden kann, um Situationen mit stark verspäteten Paketen zu behandeln.

**[0062]** Bei Schritt **71** wird der gesamte Zeitstempel des ersten erhaltenen Pakets gespeichert. Wenn ein neues Paket empfangen wird (Schritt **72**), so wird sein Zeitstempel TSnew gelesen (Schritt **73**), und es wird die Differenz D zwischen dem gespeicherten Zeitstempel TS und dem neuen Zeitstempel TSnew berechnet (Schritt **74**). Wenn die Differenz D größer ist als ein bestimmter zuvor festgelegter Wert Dmax, so erachtet der Komprimierer das Paket als zu sehr verspätet und leitet eine Korrekturmaßnahme ein (Schritt **75**). Eine solche Maßnahme wäre beispielsweise das Senden ganzer Felder anstelle von abgekürzten Feldern und das Integrieren eines Signals an den Dekomprimierer, die Kontextdaten nicht zu aktualisieren. Eine solche Maßnahme wäre auch das einfache Verwerfen solcher verspäteten Pakete. Dies wäre eine selbstverständliche Maßnahme im Zusammenhang mit Echtzeitdatenpaketen, da Pakete, die sehr spät eintreffen, in jedem Fall für die Anwendung nutzlos sind, weshalb sie bereits auf der Komprimiererseite verworfen werden können. Wenn die Differenz D nicht größer ist als Dmax, so wird der Zeitstempel TS gemäß dem erfindungsgemäßen Verfahren komprimiert. Wenn die Differenz größer als Null ist, so heißt das, dass das Paket geringfügig verspätet eintraf. Der gespeicherte Wert TS repräsentiert vorzugsweise den größten Wert des Zeitstempels, der bis dahin übertragen wurde, weshalb der Wert des gespeicherten Zeitstempels nicht aktualisiert wird. Wenn die Differenz D kleiner als Null ist, so wird der Wert des gespeicherten Zeitstempels aktualisiert (Schritt **77**). Dieses Verfahren wird für jedes Paket während der gesamten Sitzung angewendet.

**[0063]** In einigen Fällen können die Identifizierungsdaten in dem komprimierten Nachrichtenkopf auf ein Minimum komprimiert werden, wobei die Komprimierungssequenz trotzdem die gesamte Sitzung überspannt. Eine solche Ausführungsform umfasst eine Zuordnung zwischen Feldern von Netzwerkschicht- und Zugangsnetzwerkschichtprotokollen.

**[0064]** Netzwerkschicht meint in diesem Zusammenhang Protokollsichten auf Paketdatennetzwerkebene, womit IP-UDP- und RTP-Protokolle in der hier gezeigten Ausführungsform gemeint sind. Zugangsnetzwerk-

schicht meint in diesem Zusammenhang die Protokollsicht, die für das Zugangsnetzwerk spezifisch ist und für Komprimierungs- und Dekomprimierungsfunktionen verantwortlich ist, in diesem Fall das SNDCP. Eine Packet Data Unit (PDU) des SNDCP enthält eine integrale Anzahl von 8-Bit-Bytes, einen Nachrichtenkopfteil und einen Datenteil. Es werden zwei verschiedene SN-PDU-Formate definiert: SN-DATA PDU für bestätigte Datenübertragung und SN-UNITDATA PDU für nicht-quittierte Datenübertragung. **Fig. 8** veranschaulicht das Format einer SNDCP SN-UNITDATA PDU, die bei der nicht-quittierten Übertragung von GPRS verwendet wird. SN-UNITDATA PDU umfasst ein Feld-N-PDU-Nummer **81**, die eine laufende Nummer ist, welche die gesamte Sitzung hindurch fortschreitet.

**[0065]** Es kann eine Zuordnung zwischen Netzwerkschichtdatenpaketen und Datenpaketen der Protokollsicht, die für die Komprimierung zuständig sind, erzeugt werden. In der hier beschriebenen Ausführungsform wird eine Zuordnung zwischen der Feld-N-PDU-Nummer der SNDCP SN-UNITDATA PDU und der RTP-Folgenummer, der IP-Identifikation und dem RTP-Zeitstempel gezeigt. Wenn die N-PDU-Nummer um Eins zunimmt, so erhöhen sich die Werte im RTP-Folgenummernfeld **316** und im IP-Identifikationsfeld **335** im Allgemeinen um einen Wert, der während der gesamten Sitzung unverändert ist. Des Weiteren gibt es Codecs, für die die Differenz zwischen aufeinanderfolgenden RTP-Zeitstempeln konstant ist. Unter Verwendung einer Hexadezimaldarstellung ist für einen Fall, wo diese Differenz = F0 ist und die Inkremente für die RTP-Folgenummer Eins und für die IP-Identifikation 0100 betragen, die folgende Zuordnung effektiv:

N-PDU-Nummer = 5

RTP-Folgenummer = 16C5

RTP-Zeitstempel = 02FFBF EF

IP-Identifikation = E7E6

N-PDU-Nummer = 6

RTP-Folgenummer = 16C6

RTP-Zeitstempel = 02FFC0 DF

IP-Identifikation = E8E6

N-PDU-Nummer = 7

RTP-Folgenummer = 16C7

RTP-Zeitstempel = 02FFC1 DF

IP-Identifikation = E9E6

**[0066]** Obgleich hier konstante Inkremente verwendet werden, kann die Zuordnung auch auf verschiedene andere Arten erfolgen. Beispielsweise kann eine Zuordnungsfunktion zwischen Zugangsnetzwerkschichtprotokoll und den Netzwerkschichtprotokoll-Nachrichtenkopffeldern eingerichtet werden. Des Weiteren kann auch – je nach der Anwendung – eine Zuordnung zwischen allen anderen Feldern des Zugangsnetzwerkschichtprotokolls und des Netzwerkschichtprotokolls genutzt werden. Die Kontextdaten im Dekomprimierer umfassen die Informationen, die für eine Zuordnung des Zugangsnetzwerkschichtprotokollfeldes zu den Netzwerkschichtprotokollfeldern benötigt werden. Der Vergleichsschritt des in **Fig. 4** dargestellten Verfahrens umfasst eine einfache Validitätsprüfung des Inhalts des Zugangsnetzwerkschichtfeldes. Die Kontextdaten bleiben vorzugsweise unverändert und erfordern somit keine Aktualisierung (siehe Schritt **45**).

**[0067]** Für Netzwerkschichtpakete, bei denen die Möglichkeit besteht, dass sich die Konstantfelder während einer Sitzung ändern, wird eine alternative Ausführungsform, die in **Fig. 9** gezeigt ist, vorgeschlagen. Die Ausführungsform wird auch hier anhand des Beispiels dargestellt, wo die Sendeeinheit die MS ist und die Empfangseinheit der SGSN ist. In Verbindung mit der Sitzungseinrichtung wird der Komprimierungszustand sowohl in der MS ( $SoC_c$ ) als auch im SGSN gespeichert (Schritt **91**) ( $SoC_d$ ). Wenn ein Paket vom Sprach-Codec empfangen wird, um es zum SGSN zu senden (Schritt **9**), so wird geprüft (Schritt **93**), ob es Änderungen in den Konstantfeldern des zu komprimierenden Nachrichtenkopfes und den Feldern des Komprimierungszustandes gibt. Wenn keine Änderungen festgestellt werden, so wird der Nachrichtenkopf wie zuvor beschrieben komprimiert (Schritt **94**), und das komprimierte Paket wird zum Dekomprimierer gesandt (Schritt **95**). In jedem Fall extrahiert, wenn Änderungen festgestellt werden, eine neue SNDCP-Funktion aus dem neuen Nachrichtenkopf lediglich die geänderten Konstantfelder (Schritt **96**), aktualisiert sie auf den gespeicherten Komprimierungszustand (Schritt **97**), sendet diese Werte zum SGSN (Schritt **98**) und aktualisiert die Werte auch auf den im SGSN gespeicherten Komprimierungszustand (Schritt **98**). Die Übertragung solcher Informationen kann unter Verwendung eines quittierten Modus' oder eines leistungsstarken Fehlerschutzes erfolgen.

**[0068]** In einem Komprimierungs-/Dekomprimierungsalgorithmus kann jede beliebige Kombination dieser

Ausführungsformen verwendet werden. Als nächstes wird ein Beispiel der Verwendung des erfindungsgemäßen Verfahrens gegeben.

**[0069]** Tabelle 1 zeigt Felder in einem kompletten Nachrichtenkopf, der den Netzwerkschichtprotokollen IP, UDP, RTP entspricht. Die abgedunkelten Felder entsprechen den Feldern, die während der gesamten Sitzung unverändert bleiben, und die weißen Felder repräsentieren Felder, die sich während der Sitzung verändern können.

Tabelle 1. Beispiel eines IP-, UDP- und RTP-Nachrichtenkopfes

45	00	00	40
E7	E6	00	00
80	11	63	DC
82	E9	F4	89
82	E9	F4	8D
C1	C0	C1	C4
00	2C	AF	5E
80	04	16	C5
02	FF	BF	EF
A7	F0	03	96

**[0070]** Nehmen wir an, dies wäre der erste RTP/IP/UDP-Nachrichtenkopf, der beim SND-CP-Komprimierer empfangen wird. Die hier gezeigten Werte haben das Hexadezimalformat, und es gibt vier 8-Bit-Bytes in einer Zeile. Die ersten fünf Zeilen (zwanzig 8-Bit-Bytes) stellen einen IP-Nachrichtenkopf dar. Die beiden nächsten Zeilen sind 8-Bit-Bytes eines UDP-Nachrichtenkopfes, und die drei letzten Zeilen stellen den RTP-Nachrichtenkopf dar, die alle zusammen einen typischen RTP/UDP/IP-Nachrichtenkopf bilden (40 Bytes). Der SND-CP-Komprimierer muss diese Werte kopieren, und der komplette Nachrichtenkopf wird an das entsprechende SND-CP-Element gesandt. Tabelle 2 zeigt einen nachfolgenden Nachrichtenkopf, der beim Komprimierer ein geht.

Tabelle 2. Nächster IP/UDP/RTP-Nachrichtenkopf

45	00	00	40
E8	E6	00	00
80	11	62	DC
82	E9	F4	89
82	E9	F4	8D
C1	C0	C1	C4
00	2C	04	40
80	04	16	C6
02	FF	C0	DF
A7	F0	03	96

**[0071]** Die Felder, die sich von den gespeicherten Werten unterscheiden, sind in Tabelle 1 und 2 abgedunkelt und umfassen:

- 16-Bit-Identifikationsfeld des IP-Nachrichtenkopfes:

von E7E6 bis E8E6

- 16-Bit-Nachrichtenkopf-Prüfsumme des IP-Nachrichtenkopfes:

von 63DC bis 62DC

- 16-Bit-UDP-Prüfsumme:

von AF5E bis D440

- Folgenummer des RTP-Nachrichtenkopfes:

von 16C5 bis 16C6

- Zeitstempel des RTP-Nachrichtenkopfes:

von 02FFBF0F bis 02FFC0DF

**[0072]** Die anderen Felder bleiben unverändert. Nun wird gemäß der Erfindung folgender komprimierter Nachrichtenkopf erzeugt:

Tabelle 3. Komprimierter Nachrichtenkopf

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T=0	M=0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1

**[0073]** Das Format des komprimierten Nachrichtenkopfes folgt dem, das in Verbindung mit **Fig. 5** gezeigt wurde, und ist in binärer Form dargestellt. Da die Länge des Paketes sich nicht geändert hat, ist das 7. Bit des ersten Oktetts = 0, das Markierungsbit ist 0, und in diesem Fall sind die übrigen Bits im ersten Oktett Nullen. Die beiden nächsten Oktette enthalten den abgekürzten Zeitstempel (C0 DF in Hexadezimalformat). Ein SND-CP-Paket, das den komprimierten Nachrichtenkopf und die RTP-Nutzinformationen umfasst, wird an den Dekomprimierer gesandt.

**[0074]** Der komplette Zeitstempel wird so aus dem abgekürzten Zeitstempelwert rekonstruiert, wie es oben in Verbindung mit **Fig. 6** beschrieben wurde. Die Folgenummer des RTP-Nachrichtenkopfes und die 16-Bit-Identifikationsnummer des IP-Nachrichtenkopfes sind unter Verwendung der N-PDU-Nummer des SND-CP-Pakets als ein Versatz vom letzten kompletten Nachrichtenkopf abzuleiten. Als der Dekomprimierer das erste Paket, das den kompletten Nachrichtenkopf umfasste, empfing, stellte der Algorithmus eine Verbindung zwischen der RTP-Folgenummer und der N-PDU-Nummer des SND-CP-Pakets her. In diesem Fall wäre die Folgenummer 16 C5, und die N-PDU-Nummer wäre beispielsweise x. Wenn der komprimierte Nachrichtenkopf versendet wird, so ist die SN-UNITDATA N-PDU-Nummer = y, was anzeigt, dass die Differenz zwischen den N-PDUs = y - x, und diese Nummer ist der gespeicherten Folgenummer hinzuzuschlagen.

**[0075]** Als ein Beispiel eines Netzwerkelements, welches das hier beschriebene Verfahren implementiert, wird ein mobiles Endgerät eines Mobilkommunikationssystems dargestellt.

**[0076]** Die Struktur des mobilen Endgerätes gemäß der Erfindung ist weitestgehend die eines herkömmlichen mobilen Endgerätes nach dem Stand der Technik, wie sie dem Fachmann bekannt ist. Wie in **Fig. 10** gezeigt, steuert eine Zentrale Recheneinheit (Central Processing Unit – CPU) **101** die Blöcke, die für die verschiedenen Funktionen der Mobilstation zuständig sind: ein Speicher (MEM) **102**, ein Hochfrequenzblock (Radio Frequency – RF) **103**, eine Benutzerschnittstelle (User Interface – UI) **104** und eine Schnittstelleneinheit (Interface Unit – IU) **105**. Die CPU ist in der Regel mit einem oder mehreren funktional zusammenwirkenden Mikroprozessoren implementiert. Der Speicher umfasst vorzugsweise ein ROM (Read Only Memory) und ein RAM (Random Access Memory) und ist im Allgemeinen um einen Speicher ergänzt, der mit dem SIM-Benutzeridentifikationsmodul versehen ist. Gemäß seinem Programm verwendet der Mikroprozessor den RF-Block **103** zum Senden und Empfangen von Nachrichten auf dem Funkweg.

**[0077]** Die Kommunikation mit dem Benutzer wird über die UI **104** verwaltet, die in der Regel einen Lautsprecher, eine Anzeige und eine Tastatur umfasst. Die Schnittstelleneinheit **105** ist die Verbindung zu einer Datenverarbeitungseinheit und wird durch die CPU **101** gesteuert. Bei der Datenverarbeitungseinheit **101** kann es sich um einen integrierten Datenprozessor oder ein externes Datenverarbeitungsgerät handeln.

**[0078]** **Fig. 10** veranschaulicht außerdem die Funktionsmodule einer Datenverarbeitungseinheit TE gemäß

der Erfindung.

**[0079]** Bei dem Endgerät TE kann es sich beispielsweise um einen Personalcomputer, wie er aus Büroumgebungen bekannt ist und dem Stand der Technik entspricht, oder um eine Workstation handeln. Die TE kann auch ein integraler Bestandteil der MS (beispielsweise Smartphone) sein und kann Elemente wie beispielsweise die UI und die CPU mit der MS gemeinsam nutzen. Umgekehrt kann auch die MS in die TE integriert sein (beispielsweise Kartentelefon). Es versteht sich, dass – auch wenn in **Fig. 10** zwei separate Blöcke gezeigt sind – damit keinerlei Einschränkungen bezüglich der Konfiguration impliziert werden.

**[0080]** Die TE umfasst im Wesentlichen eine Schnittstelleneinheit IU **107**, die derjenigen in der MS entspricht, um die Schnittstelle zur MS zu steuern. Sie umfasst des Weiteren eine Benutzerschnittstelle UI **108** zum Empfangen von Benutzerbefehlen und zum Ausgeben von Informationen an den Benutzer, einen Speicher MEM **109** zum Speichern von Anwendungen SW APP **110** und anwendungsbezogenen Daten und einen Prozessor CPU **111** zum Steuern der Arbeitsabläufe der TE und zum Ausführen der Verfahrensabläufe der Anwendung.

**[0081]** Es gibt eine Mehrzahl von Verfahren, um die Mobilstation MS und die Datenverarbeitungseinheit miteinander zu verbinden, die dem Fachmann allesamt bekannt sind. Eines der Verfahren besteht darin, die Geräte über Schnittstelleneinheiten IU miteinander zu verbinden, die eine verdrahtete Verbindung und eine geeignete Verbindungsschnittstelle, beispielsweise einen seriellen Port, umfassen, ergänzt um geeignete Schnittstellensoftware in den CPUs, welche die Funktion der Schnittstelleneinheiten IU steuert. Ein weiteres Verfahren ist die Verwendung einer drahtlosen Verbindung im infraroten Wellenlängenbereich oder von Hochfrequenz-Transceivereinheiten mit geringer Leistung. Die neuen Lösungen, wobei die MS in die TE integriert ist, bieten auch eine sehr zweckmäßige Plattform für das erfindungsgemäße System.

**[0082]** Wenn ein Benutzer – auf der Basis von Befehlen von Benutzereingabegeräten – mit der TE Zugang zu einem Paketdatennetz haben will, so ruft der Prozessor CPU **111** aus dem Speicher MEM **109** eine Paketdatenzugriffsanwendung SW APP **110** auf. Die Anwendung wird in der CPU **111** in der Paketform verarbeitet, und wann immer die Notwendigkeit entsteht, anwendungsbezogene Informationen zu senden, wird ein Paket über die IU **107** an die MS weitergeleitet.

**[0083]** In der ersten Ausführungsform sind die Komprimierungs- und Dekomprimierungsfunktionen in der SNDCP-Schicht des Protokollstapels des mobilen Endgerätes implementiert. In der hier gezeigten Ausführungsform sind die mit SNDCP-Funktionen befassten Elemente diejenigen, die hier im MS-Teil beschrieben wurden. In Uplink-Richtung fungiert die MS als Komprimierer, und in Downlink-Richtung fungiert die MS als Dekomprimierer. Die Werte, die für die Funktionsabläufe der Komprimierung und Dekomprimierung verwendet werden, werden im Speicher **102** der MS gespeichert. Die Komprimierung erfolgt in der Zentraleinheit **111**, und die komprimierten Einheiten werden an die RF-Einheit **102** weitergeleitet, um sie durch das BS zum SGSN zu übertragen. Die komprimierten Pakete vom SGSN werden von der RF-Einheit empfangen und zum Zweck des Dekomprimierens zur CPU **111** weitergeleitet. Die dekomprimierten Pakete werden über die Schnittstelleneinheiten **105** und **107** an die TE weitergeleitet.

**[0084]** Obgleich die Erfindung anhand einer bevorzugten Ausführungsform gezeigt und beschrieben wurde, erkennt der Durchschnittsfachmann, dass Modifizierungen an der bevorzugten Ausführungsform vorgenommen werden können, ohne dass von dem weiter unten beanspruchten Geltungsbereich der Erfindung abgewichen wird.

**[0085]** Beispielsweise werden in der Zukunft GPRS oder entsprechende Dienste (GPRS-Ableitungen) auch in anderen Mobiltelekommunikationssystemen implementiert. Der WCDMA-Standard (Wideband Code Division Multiple Access) der dritten Generation besitzt eine Struktur, die dem GPRS sehr ähnelt, und umfasst eine L3CE-Schicht, die dem SNDCP des GPRS entspricht. Eine solche Schicht zum Integrieren der Funktionen des erfindungsgemäßen Verfahrens ist die SNDCE-Schicht des CDMA2000. Es ist möglich, die Erfindung auch in Festnetzen anzuwenden. Die Möglichkeiten zum Realisieren und Verwenden der Erfindung sind darum nur durch die angehängten Ansprüche beschränkt.

### Patentansprüche

1. Verfahren zum Übertragen eines Datenpaketes von einem Komprimierer (MS) zu einem Dekomprimierer (SGSN), wobei dieses Datenpaket einen Nachrichtenkopf mit Nachrichtenkopfdatenfeldern enthält, wobei eine Anzahl der Nachrichtenkopfdatenfelder, die während der Datenübertragung unverändert bleiben, in dem Dekomprimierer gespeichert werden, wobei das Verfahren folgendes enthält:

Senden eines Datenpaketes vom Komprimierer und Empfangen dieses Datenpaketes beim Dekomprimierer, wobei dieses Datenpaket Informationen über ein oder mehrere Nachrichtenkopfdatenfelder umfasst, die sich während der Datenübertragung ändern;

Dekomprimieren des Nachrichtenkopfes mittels der gespeicherten Nachrichtenkopfdatenfelder und der empfangenen Informationen über das eine oder die mehreren Nachrichtenkopfdatenfelder, die sich während der Datenübertragung ändern;

**dadurch gekennzeichnet,**

dass nach der Sitzungseinrichtung im Zusammenhang mit einem sich verändernden Nachrichtenkopfdatenfeld lediglich ein komprimierter Wert vom Komprimierer versendet und vom Dekomprimierer empfangen wird, wobei dieser komprimierte Wert das Datenpaket in einer Komprimierungssequenz identifiziert und wobei der komprimierte Wert aus einer ersten Anzahl geringst-wertiger Bits des Nachrichtenkopfdatenfeldes besteht;

dass in dem Dekomprimierer Kontextdaten gespeichert werden, die Informationen umfassen, mit denen der empfangene komprimierte Wert einer entsprechenden Komprimierungssequenz zugeordnet wird, wobei die Informationen entsprechend den empfangenen komprimierten Werten aktualisiert werden;

dass der komprimierte Wert und die Informationen der entsprechenden Komprimierungssequenz dazu verwendet werden, den komprimierten Wert in ein dekomprimiertes Nachrichtenkopfdatenfeld hinein abzubilden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Komprimierungssequenz eine Gruppe aufeinanderfolgender Datenpakete umfasst, die anhand der Auflösung, die von dem Komprimierungswert bereitgestellt wird, voneinander unterschieden werden können.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass es sich bei dem Nachrichtenkopf um einen RTP/UDP/IP-Nachrichtenkopf handelt, wobei es sich bei dem Datenfeld um eine IP-Identifikation oder um eine RTP-Folgenummer oder um einen RTP-Zeitstempel handelt.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass die Kontextdaten mindestens eine zweite Anzahl der höchst-wertigen Bits des Datenfeldes umfassen.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass es sich bei dem Komprimierer und dem Dekomprimierer um Netzwerkelemente eines Zugangsnetzwerkes zu einem IP-Paketdatennetzwerk handelt.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass es sich bei dem Komprimierer und dem Dekomprimierer um Netzwerkelemente eines drahtlosen Zugangsnetzwerkes zu einem IP-Paketdatennetzwerk handelt.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass es sich bei dem Komprimierer und dem Dekomprimierer um Netzwerkelemente eines Mobilkommunikationsnetzwerkes handelt, das GPRS unterstützt.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass die Komprimierungs- und Dekomprimierungsfunktionen in der SNDNP-Schicht des GPRS implementiert werden.

9. Zugangsnetzwerkelement (MS), umfassend:

Mittel (**101, 103**) zum Übertragen eines Datenpaketes zu einem Dekomprimierer (SGSN), wobei dieses Datenpaket einen Nachrichtenkopf mit Nachrichtenkopfdatenfeldern enthält;

Mittel (**101, 103**) zum Komprimieren des Nachrichtenkopfes durch Ausschließen von Nachrichtenkopfdatenfeldern, die während der Datenübertragung unverändert bleiben, vom Senden;

Mittel (**101, 103**) zum Senden eines Datenpakets an den Dekomprimierer, wobei dieses Datenpaket Informationen über ein oder mehrere Nachrichtenkopfdatenfelder umfasst, die sich während der Datenübertragung ändern;

gekennzeichnet durch:

Mittel (**101, 103**), die so konfiguriert sind, dass sie nach der Sitzungseinrichtung im Zusammenhang mit einem sich verändernden Nachrichtenkopfdatenfeld lediglich einen komprimierten Wert senden, der mit dem Identifizieren des Datenpaketes in einer Komprimierungssequenz verbunden ist, wobei der komprimierte Wert aus einer ersten Anzahl geringstwertiger Bits des Nachrichtenkopfdatenfeldes besteht.

10. Zugangsnetzwerkelement nach Anspruch 9, gekennzeichnet durch:

Mittel (**101, 103**) zum Empfangen eines Datenpaketes, das eine Ordnungszahl umfasst, wobei diese Ordnungszahl die Reihenfolge des Paketes in einer Reihe gesendeter Pakete angibt;

Mittel (**101, 103**) zum Vergleichen der Ordnungszahl des empfangenen Paketes mit einer zuvor gespeicherten Ordnungszahl;

Mittel (**101, 103**) zum Initiiieren einer Fehlerfunktion in Reaktion darauf, dass die Differenz zwischen der Ordnungszahl des empfangenen Paketes und der Vergleichsordnungszahl ein zuvor festgelegtes Limit überschreitet;

Mittel (**101, 103**) zum Initiiieren eines Algorithmus' zum Komprimieren eines Nachrichtenkopfes in Reaktion darauf, dass die Differenz zwischen der Ordnungszahl des empfangenen Paketes und der Vergleichsordnungszahl ein zuvor festgelegtes Limit unterschreitet;

Mittel (**101, 103**) zum Speichern der Ordnungszahl des empfangenen Paketes als die Vergleichsordnungszahl in Reaktion darauf, dass die Ordnungszahl des empfangenen Paketes größer ist als die Vergleichsordnungszahl.

11. Zugangsnetzwerkelement (MS), umfassend:

Mittel (**101, 103**) zum Empfangen von Datenpaketen, wobei diese Datenpakete einen Nachrichtenkopf mit Nachrichtenkopfdatenfeldern enthalten;

Mittel zum Speichern (**101, 102**) von Nachrichtenkopfdatenfeldern, die während der Datenübertragung unverändert bleiben;

Mittel zum Empfangen (**101, 103**) komprimierter Datenpakete, die Informationen über ein oder mehrere Nachrichtenkopfdatenfelder umfassen, die sich während der Datenübertragung ändern;

Mittel zum Dekomprimieren (**101**) komprimierter Datenpakete mittels der gespeicherten Nachrichtenkopfdatenfelder und der empfangenen Informationen über das eine oder die mehreren Nachrichtenkopfdatenfelder, die sich während der Datenübertragung ändern;

gekennzeichnet durch:

Mittel (**101, 103**), die so konfiguriert sind, dass sie nach der Sitzungseinrichtung im Zusammenhang mit einem sich verändernden Nachrichtenkopfdatenfeld in einem Datenpaket lediglich einen komprimierten Wert empfangen, der das Datenpaket in einer Komprimierungssequenz identifiziert, wobei der komprimierte Wert aus einer ersten Anzahl geringstwertiger Bits des Nachrichtenkopfdatenfeldes besteht;

Mittel zum Speichern (**101, 102**) von Kontextdaten, die Informationen umfassen, mit denen der empfangene komprimierte Wert einer entsprechenden Komprimierungssequenz zugeordnet wird, wobei die Informationen entsprechend den empfangenen komprimierten Werten aktualisiert werden;

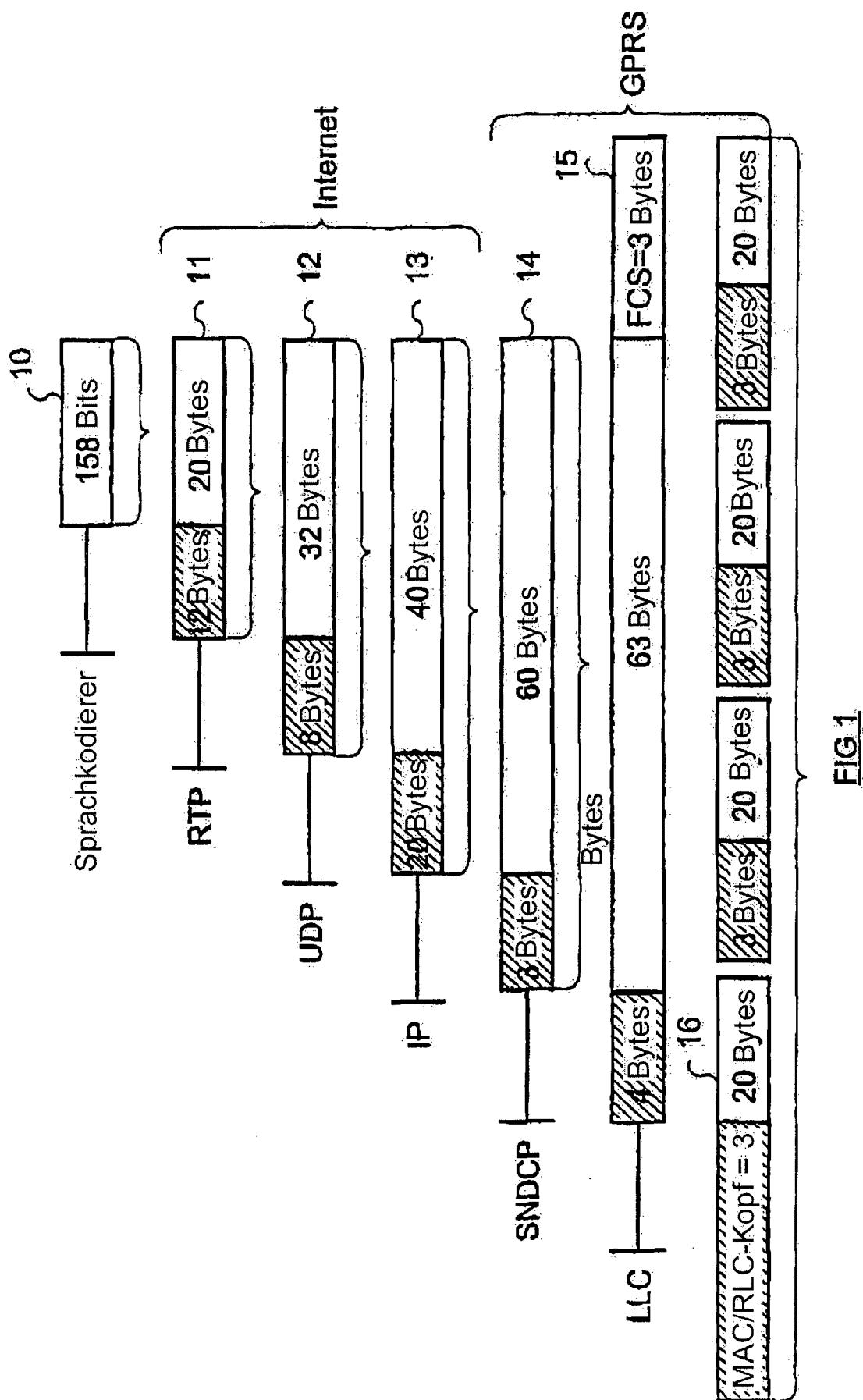
Mittel (**101, 102**) zum Verwenden des komprimierten Wertes und der Informationen der entsprechenden Komprimierungssequenz zu dem Zweck, den komprimierten Wert in ein Nachrichtenkopfdatenfeld in einem dekomprimierten Datenpaket hinein abzubilden.

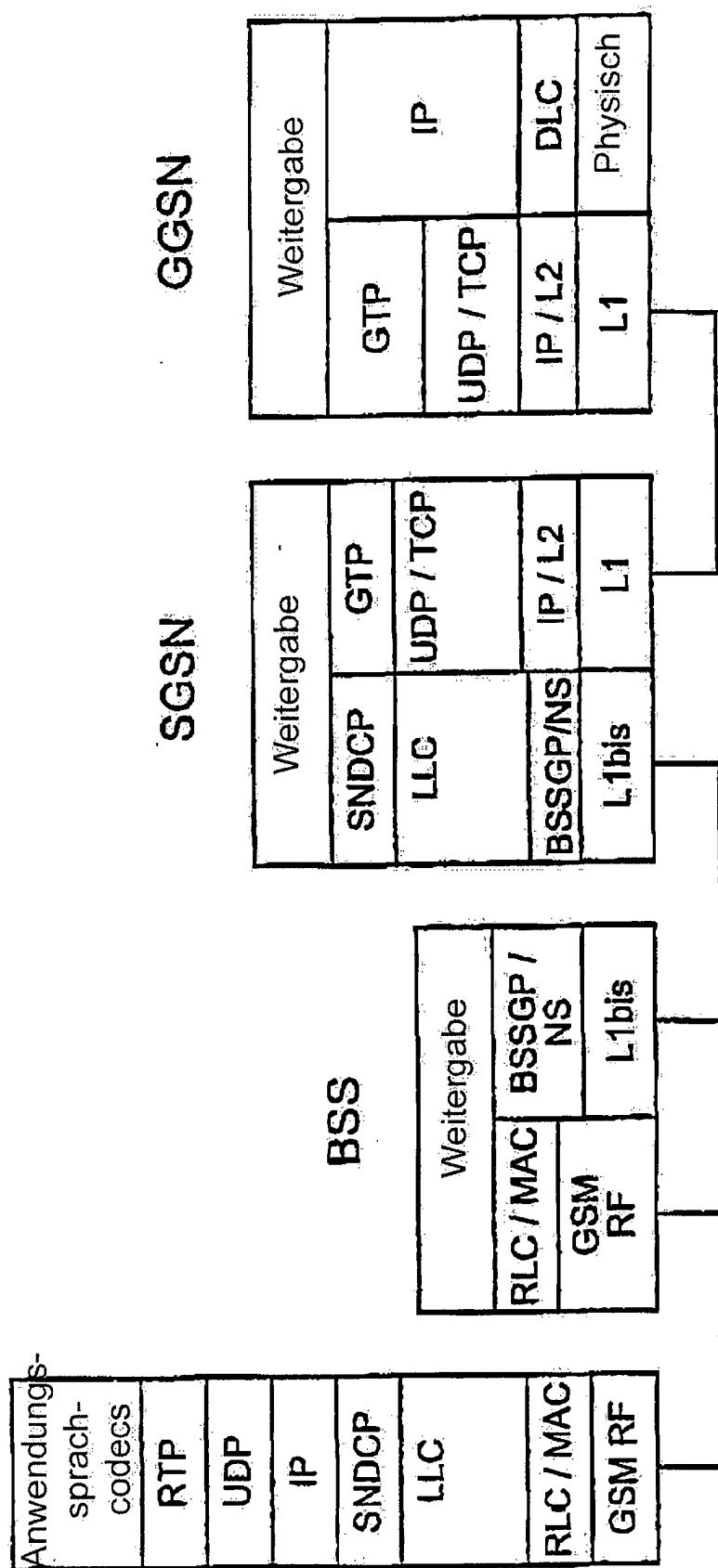
12. Zugangsnetzwerkelement nach Anspruch 9 oder 11, dadurch gekennzeichnet, dass es sich bei dem Element um ein Mobil-Endgerät eines Mobilkommunikationsnetzwerkes handelt.

13. Zugangsnetzwerkelement nach Anspruch 9 oder 11, dadurch gekennzeichnet, dass es sich bei dem Element um ein SGSN eines Mobilkommunikationsnetzwerkes handelt, das GPRS unterstützt.

Es folgen 8 Blatt Zeichnungen

## Anhängende Zeichnungen

**FIG. 1**

FIG 2

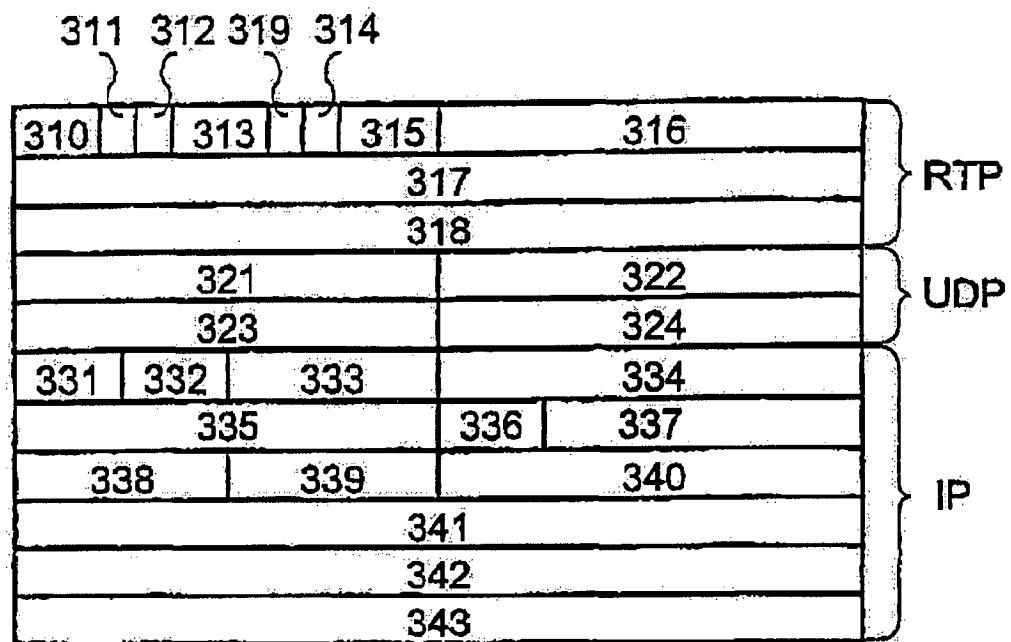


FIG 3

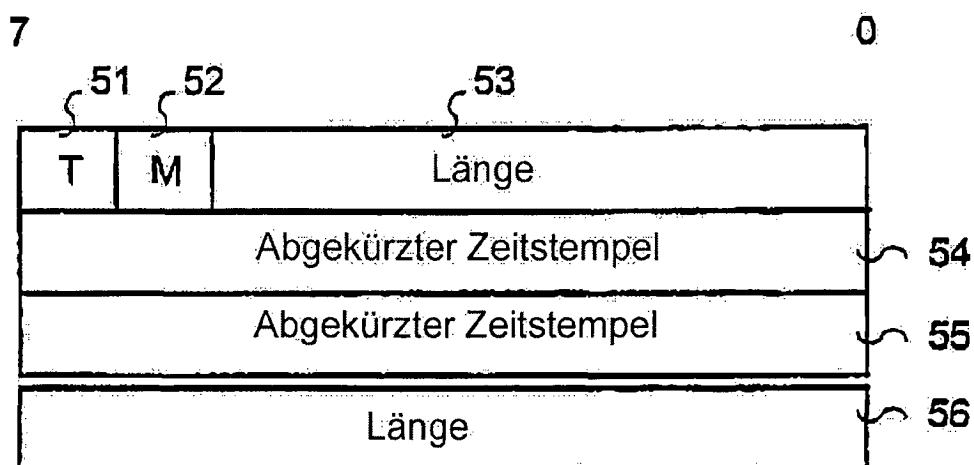
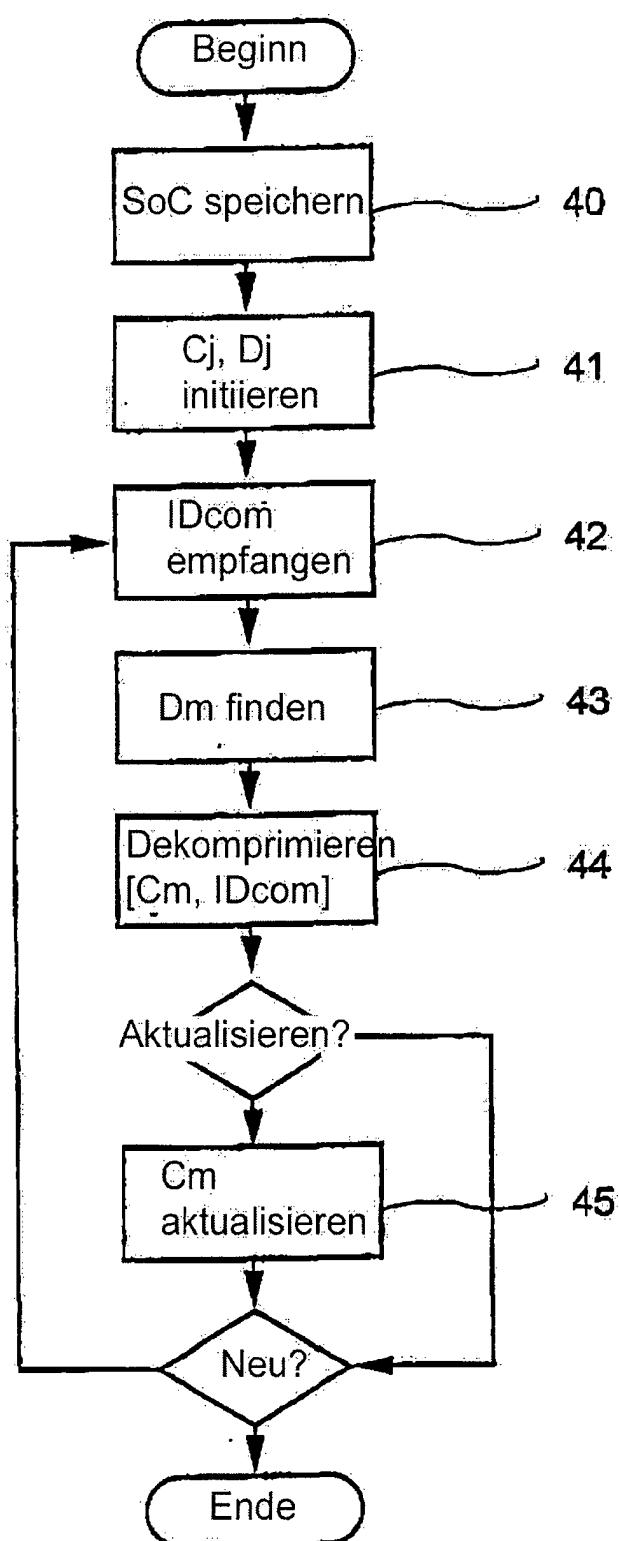


FIG 5

FIG 4

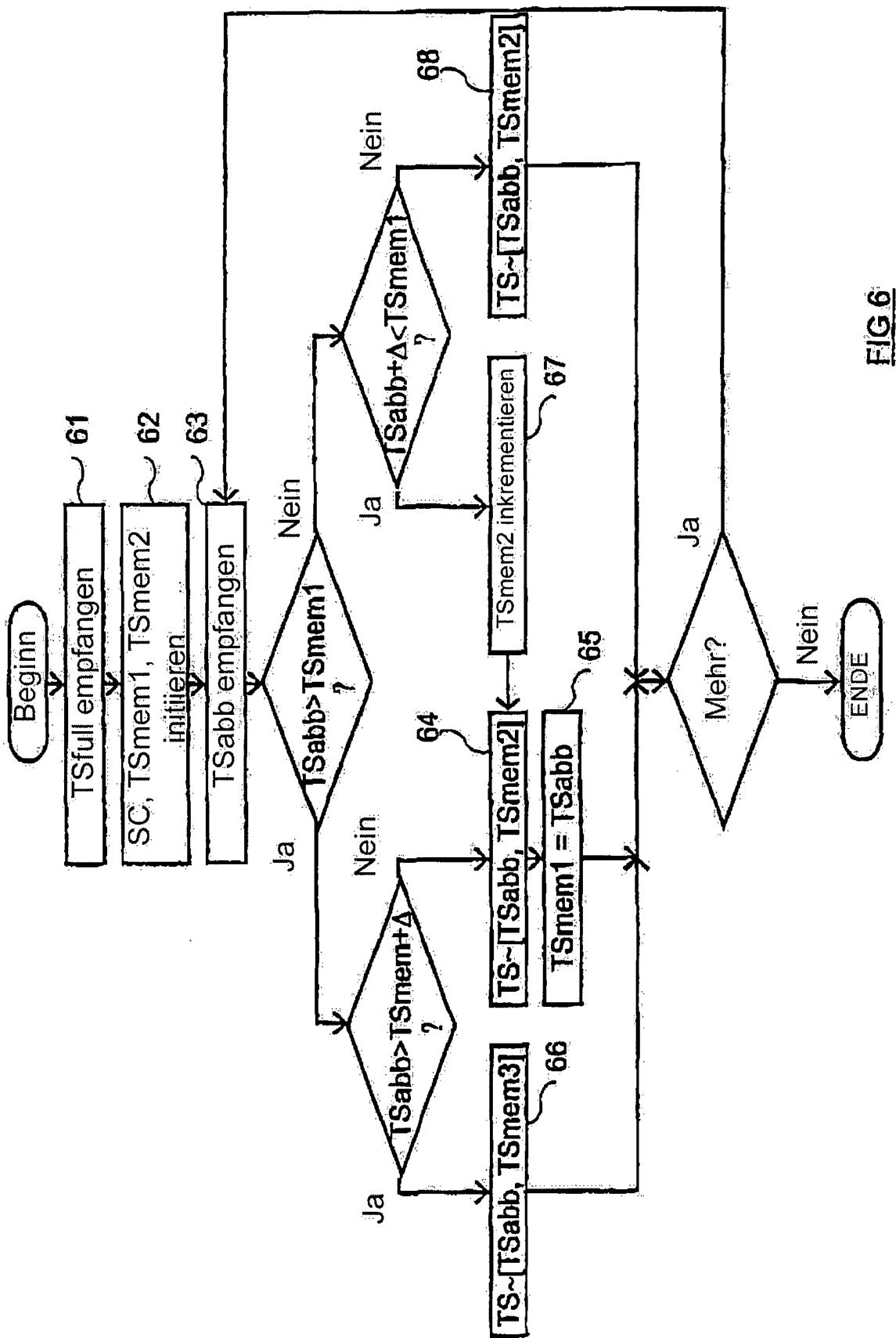


FIG 6

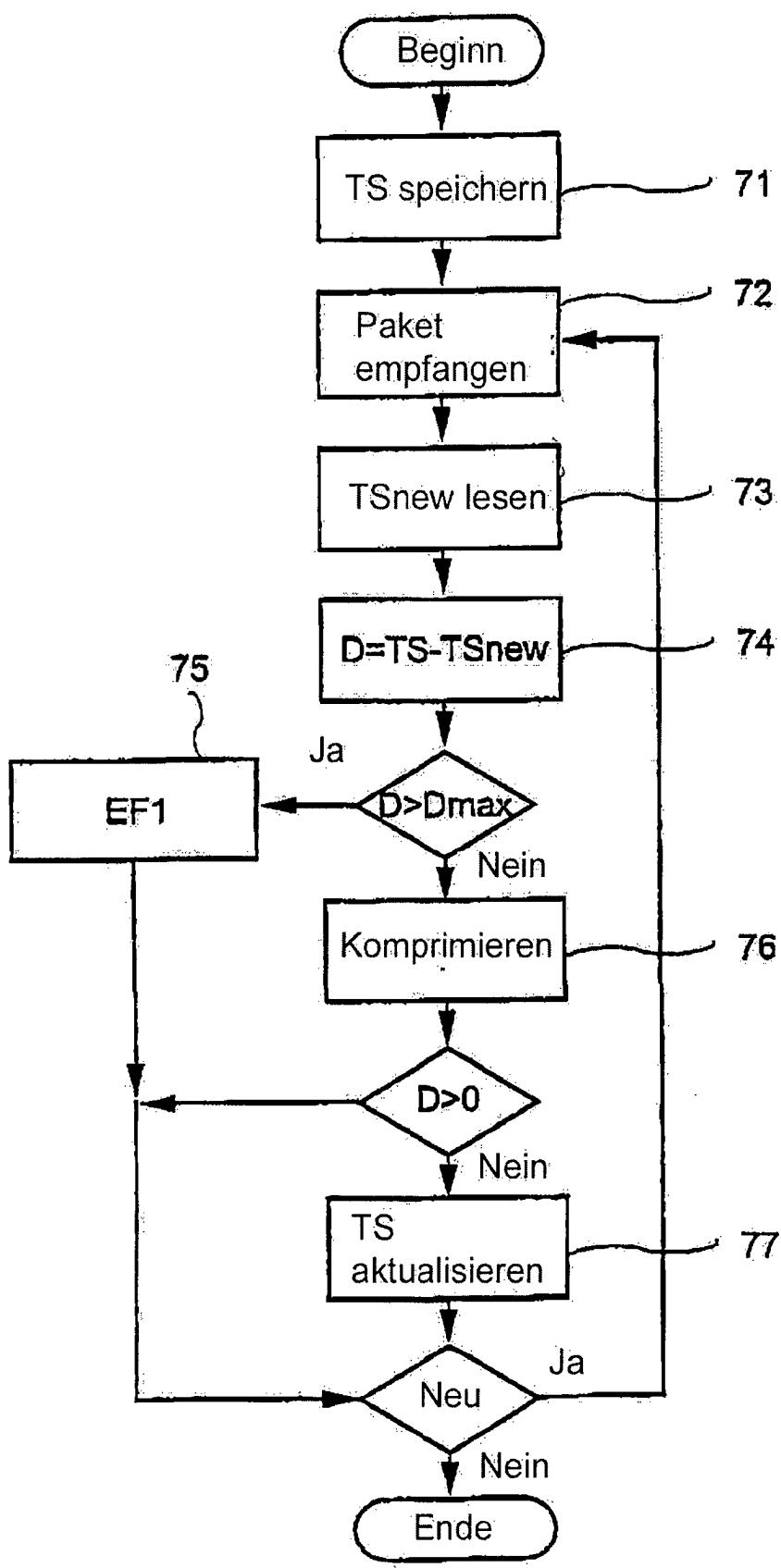
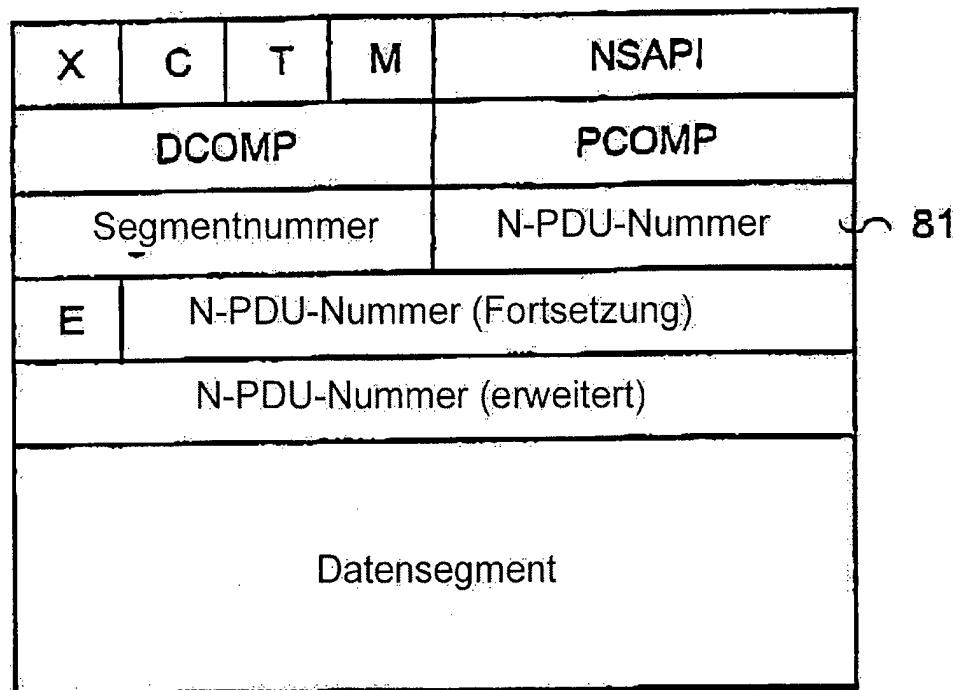
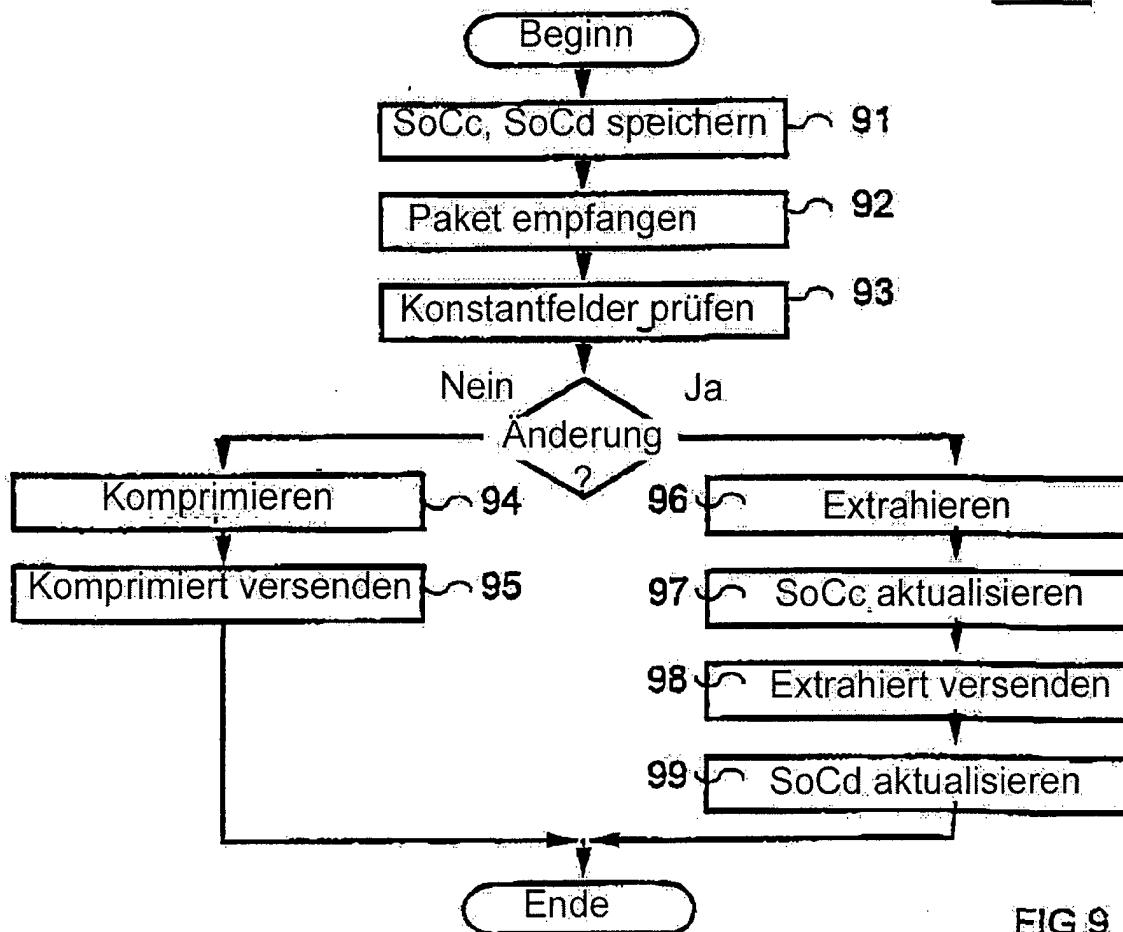


FIG 7

7

0

FIG 8FIG 9

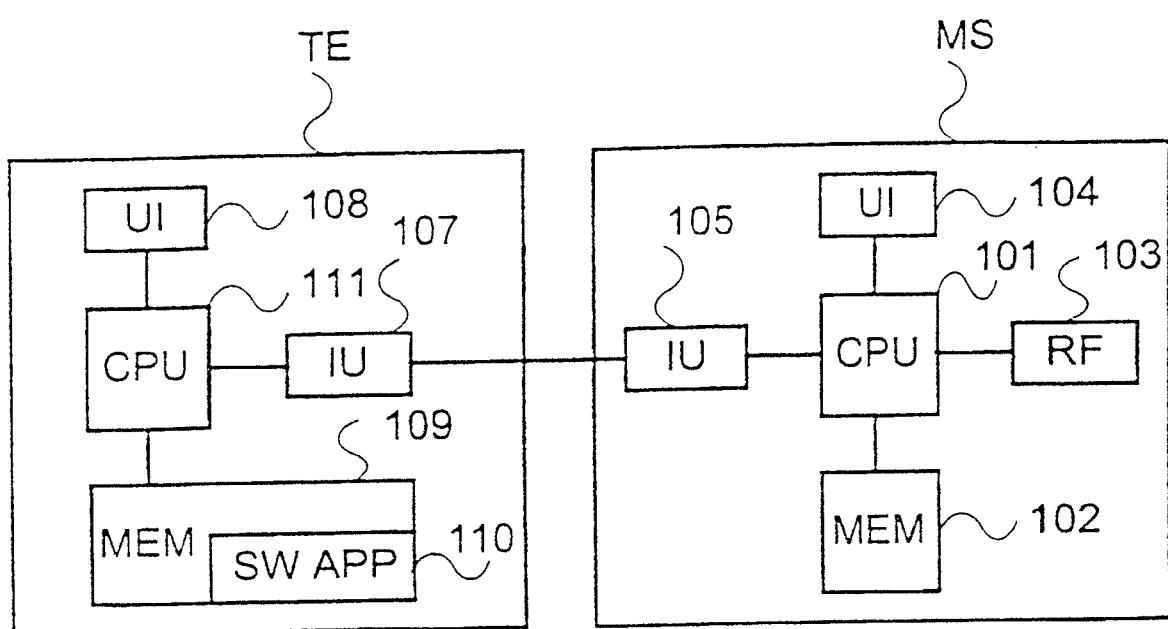


FIG 10