



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 601 13 549 T2** 2006.06.22

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 303 970 B1**

(21) Deutsches Aktenzeichen: **601 13 549.0**

(86) PCT-Aktenzeichen: **PCT/IB01/01664**

(96) Europäisches Aktenzeichen: **01 965 510.9**

(87) PCT-Veröffentlichungs-Nr.: **WO 02/009389**

(86) PCT-Anmeldetag: **24.07.2001**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **31.01.2002**

(97) Erstveröffentlichung durch das EPA: **23.04.2003**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **21.09.2005**

(47) Veröffentlichungstag im Patentblatt: **22.06.2006**

(51) Int Cl.<sup>8</sup>: **H04L 29/06** (2006.01)  
**H04L 12/56** (2006.01)

(30) Unionspriorität:  
**0018119 24.07.2000 GB**

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

(74) Vertreter:  
**TBK-Patent, 80336 München**

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

(72) Erfinder:  
**RUUTU, Jussi, FIN-02210 Espoo, FI; MA, Jian,  
Beijing, CN**

(54) Bezeichnung: **TCP-FLUSSSTEUERUNG**

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

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

## Beschreibung

**[0001]** Diese Erfindung bezieht sich auf ein Steuern des Datenflusses, insbesondere in Netzwerken, deren Betreiben bei hoher Geschwindigkeit erforderlich ist.

**[0002]** Viele Netzwerke verwenden das Übertragungssteuerprotokoll (TCP, „Transmission control protocol“) als ihr Transportschichtprotokoll. Das Übertragungssteuerprotokoll ist das primäre Transportprotokoll in der üblicherweise verwendeten TCP/IP-Protokollgruppe. Das Internetprotokoll (IP) stellt keinen zuverlässigen Bitstrom zur Verfügung: Fehler im Laufe einer Übertragung werden durch das IP nicht korrigiert. Ein Überlagern von TCP auf eine IP-Anbindung implementiert einen zuverlässigen Bytestrom über den durch das IP bereitgestellten, unzuverlässigen Datagrammdienst. Als ein Teil der Implementierung des zuverlässigen Dienstes ist das TCP auch für Fluss- und Blockierungssteuerung verantwortlich: Sicherstellen, dass die Daten bei einer mit den Kapazitäten sowohl des Empfängers als auch der zwischenliegenden Anbindungen in dem Netzwerkweg bzw. -pfad konsistenten Rate übertragen werden. Im Ergebnis sind die meisten Durchsatzprobleme in einem TCP/IP-System in dem TCP begründet.

**[0003]** Das TCP ist eines der wenigen Transportprotokolle, das Blockierungssteuermechanismen aufweist (siehe W. Richard Stevens, „TCP/IP Illustrated Volume 1, The Protocols“, und IETF RFC 793, „Transport Control Protocol“, September 1981). Ein Schlüsselement des TCP-Blockierungsmechanismus ist sein Langsam-Startsondieralgorithmus („Slow Start Probe Algorithm“). Fährt eine TCP-Verbindung hoch, dann erfordert die TCP-Spezifikation, dass die Verbindung konservativ ist, und nimmt an, dass die für den Empfänger zur Verfügung stehende Bandbreite klein ist. Das TCP soll einen Langsam-Start („Slow Start“) genannten Algorithmus zur Sondierung des Pfads verwenden, um in Erfahrung zu bringen, wie viel Bandbreite zur Verfügung steht, bevor die Rate des Datenflusses erhöht wird. Der Langsam-Start wird verwendet, so dass sich der TCP-Fluss, wenn eine neue TCP-Verbindung gestartet wird, nach einer unbestimmten Untätigkeitszeitspanne, oder wenn eine Bestätigung (ACK, „Acknowledgement“) nach einer (ein verlorenes Paket anzeigenden) eingestellten Zeitspanne nicht zurückgekehrt ist, fortsetzt. Außerdem wird die Größe des Übertragungsfensters verringert. Jede TCP-Bestätigung spezifiziert ein entsprechendes Paket, so dass das System identifizieren kann, welches Paket verloren gegangen ist.

**[0004]** Eine TCP-Anbindung wird zwischen zwei Einrichtungen eingerichtet, von denen eine die Quelle bzw. Quelleneinrichtung einer zu übertragenden Datennachricht ist, und die andere die Datennachricht empfangen soll. Die beiden Einrichtungen sind durch einen Datenpfad verbunden, über den sich die Daten bewegen sollen. Die Quelleneinrichtung erzeugt Datagramme (z. B. Pakete), die zusammen die zu sendende Nachricht darstellen. Die Datagramme werden über den Datenpfad übertragen. Schließlich werden die Datagramme durch die Empfängereinrichtung wieder zusammengesetzt, die ebenso ein Wiederübertragen aller Datagramme anfordert, die sie nicht korrekt dekodieren kann.

**[0005]** Während der Langsam-Startalgorithmus läuft, startet die TCP-Quelleneinrichtung mit einem auf ein Segment eingestellten, nominellen Übertragungsfenster. In diesem Zustand überträgt sie nicht mehr als ein Datagramm (z.B. Paket), wenn sie nicht eine Bestätigung von der Empfängereinrichtung empfangen hat. Wird eine Bestätigung (ACK) empfangen, dann wird die Fenstergröße um eins erhöht. In der Praxis tendiert die Fenstergröße zu einer exponentiellen Erhöhung, da eine ACK die Größe auf zwei erhöht, ACKs von diesen beiden Segmenten die Fenstergröße auf vier erhöhen, deren ACKs wiederum die Größe auf acht erhöhen usw. Dies ist gemäß [Fig. 1](#) gezeigt. Linie 1 gemäß [Fig. 1](#) zeigt das exponentielle Wachstum der Fenstergröße FG bis zu einer maximalen  $FG_{\text{VERF}}$  einem Zeitpunkt  $t_1$ .

**[0006]** Es gibt zwei Probleme mit dem Langsam-Startalgorithmus auf Hochgeschwindigkeitsnetzwerken. Zuerst ist der Sondieralgorithmus derart gestaltet, dass es eine beträchtliche Zeit dauern kann, bis die Anbindung ihre Maximalgeschwindigkeit erreicht. Die zum Erreichen der Maximalgeschwindigkeit erforderliche Zeit ( $t_1$  gemäß [Fig. 1](#)) ist zur Umlaufzeit (RTT, „Round Trip Time“) und dem Verzögerungs-Bandbreitenprodukt (DBW, „Delay Bandwidth Product“) direkt proportional und zur mittleren Segmentlänge (L) indirekt proportional. Erhöht sich die verfügbare Bandbreite, oder erhöht sich die Umlaufzeit, oder beide, dann kann diese Hochfahrzeit sehr lang sein. Ist die Anbindung ansonsten untätig, dann wird während jener Zeitspanne ein Großteil der Anbindungsbandbreite nicht verwendet, und deshalb potentiell verschwendet. Dies ist gemäß [Fig. 1](#) gezeigt. Es kann angenommen werden, dass von dem Start der TCP-Verbindung an eine der verfügbaren Fenstergröße  $FG_{\text{VERF}}$  entsprechende Bandbreite zur Verfügung steht. Es wird jedoch bis zum Zeitpunkt  $t_1$  die Bandbreite unterhalb jener nicht zur Gänze verwendet. Deshalb stellt der schraffierte Bereich 2 gemäß [Fig. 1](#) die verschwendete Bandbreite dar.

**[0007]** Ein potentiell bedeutsameres Problem besteht darin, dass in vielen Fällen (insbesondere wenn vergleichsweise kurze Nachrichten zu senden sind) die gesamte Datenübertragung vollendet ist, bevor der Langsam-Startalgorithmus beendet und die bandbreitenbegrenzte maximale Fenstergröße  $FG_{VERF}$  erreicht ist. In dieser Situation widerfährt dem Benutzer nie die volle Anbindungsbandbreite. Die gesamte Übertragungszeit wird in dem Langsam-Start verbracht. Es gibt einige praktische Situationen, in denen dies auftreten kann. Dieses Problem ist beispielsweise insbesondere für Hyper-Texttransferprotokoll- (HTTP, „Hyper-Text Transfer Protocol“)-Verbindungen (wie zur Übertragung von Daten des world wide web verwendet) schwerwiegend, da das HTTP bekanntermaßen für jedes Element auf einer Netzseite eine neue TCP-Verbindung startet.

**[0008]** Es liegt in dem Langsam-Startalgorithmus selbst zusätzlich zu der Verschwendung von Bandbreite, die durch den Langsam-Startalgorithmus verursacht wird, eine Schwäche darin, dass dieser einen Paketverlust als ein Anzeichen einer Blockierung interpretiert. Tatsächlich kann ein Paketverlust aufgrund anderer Ursachen, wie Übertragungsfehler, auftreten. Es steht unter dem TCP kein einfacher Weg zur Unterscheidung von Verlusten aufgrund von Übertragungsfehlern von Verlusten aufgrund einer Blockierung offen, weswegen der Langsam-Startalgorithmus auf der konservativen Annahme ausgelegt ist, dass alle Verluste aufgrund einer Blockierung auftreten. Somit senkt die TCP-Quelleneinrichtung ihre Übertragungsrate, selbst wenn ein Paketverlust aufgrund eines Übertragungsfehlers anstelle einer Blockierung auftritt. Nach dieser Ratensenkung kann es eine unnötig lange Zeit dauern, bis die volle Übertragungsrate erreicht ist. Im Ergebnis wird wieder zur Verfügung stehende Bandbreite verschwendet.

**[0009]** Diese Probleme werden erschwert, wenn das TCP über Anbindungen läuft, die vergleichsweise hohe Fehlerraten oder eine variable Datenkapazität aufweisen. Es weisen beispielsweise in einigen Fällen drahtlose Anbindungen (wie jene zu Mobiltelefonen oder Satelliten) höhere Fehlerraten als herkömmliche kabelgebundene terrestrische Anbindungen auf. Derartige höhere Fehlerraten spielen aus zwei Gründen eine Rolle. Sie verursachen erstens Fehler in Datagrammen, die dann wieder übertragen werden müssen. Zweitens interpretiert das TCP einen Verlust typischerweise als ein Anzeichen einer Blockierung, und wenn ein Verlust angegriffen wird, greift es auf eine modifizierte Version des Langsam-Startalgorithmus zurück.

**[0010]** Es existieren Systeme, die sich eine Restbandbreite auf einer Anbindung zu Nutzen machen können, die momentan von anderen über die Anbindung laufenden Verbindungen übrig geblieben ist. Ein Beispiel ist der Verfügbare-Bitrate- (ABR, „Available Bit Rate“)-Dienst des asynchronen Übertragungsbetriebsart- (ATM, „Asynchronous Transfer Mode“)-Systems. Aufgrund des ABR-Dienstes kann ein Benutzer Bandbreite verwenden, die von einer konstanten Bitrate (CBR, „Constant Bit Rate“) und anderem Verkehr höherer Priorität übrig gelassen wird. Ein weiteres Beispiel ist das generische Paketfunksystem (GPRS, „Generic Packet Radio System“), das die Bandbreite eines globalen Systems für ein Mobilkommunikations- (GSM, „Global System for Mobile Communications“)-Zellulartelefonnetzwerk verwendet, das nicht für GSM-Sprachverbindungen verwendet wird. In dem GSM-System gibt es eine Anzahl von GSM-Frequenzschlitzen, von denen jeder in acht Zeitschlitze unterteilt ist. Ein jeder Zeitschlitz und eine jede Frequenzkombination kann eine GSM-Sprachverbindung unterstützen. Die grundlegende Idee des GPRS ist eine Verwendung von Zeitschlitzen, die sich momentan nicht für Sprachverbindungen in Verwendung befinden, zum Führen von Datenpaketen. Es ändert sich natürlich die Anzahl verfügbarer Zeitschlitze für GPRS abhängig von der momentanen Nachfrage für GSM-Sprachverbindungen über die Zeit.

**[0011]** Ein Problem dynamischer Systeme dieser Art besteht darin, dass die für den Dienst zugewiesene Bandbreite über die Zeit variiert. Wird beispielsweise TCP-Verkehr über GPRS übertragen, dann kann sich die für die TCP-Verbindungen verfügbare Bandbreite plötzlich erhöhen oder verringern (wenn mehrere GSM-Rufe gleichzeitig beendet oder begonnen werden). Dies bedeutet, dass eine TCP-Verbindung über ein derartiges dynamisches System womöglich häufig anhalten und starten muss, und dass in jener Umgebung der Langsam-Startalgorithmus des TCP die TCP-Verbindung daran hindert, vollen Gebrauch von der verfügbaren Bandbreite zu machen. Somit werden Netzwerkressourcen verschwendet, und ein Benutzer kann nicht die volle Leistung der dynamischen verfügbaren Bandbreite erhalten. Dieses Problem ist bei drahtlosen Kommunikationssystemen besonders wichtig, da die Funkschnittstellenkapazität begrenzt und teuer ist.

**[0012]** Hätte das TCP-System mehr Informationen bezüglich der Eigenschaften des Pfads, über den die Daten gesendet werden, dann könnte es zu einer Beschleunigung des Langsam-Startvorgangs in der Lage sein, beispielsweise durch Überspringen zumindest eines Teils des Langsam-Startvorgangs und durch Beginnen einer Übertragung bei einer beträchtlich höheren Rate, als durch ein Datagramm dargestellt wird, falls eine Unterstützung durch die Anbindung als wahrscheinlich bekannt wäre. Zur Beschleunigung des Langsam-Startvorgangs wurden mehrere Modelle vorgeschlagen.

**[0013]** Eine derartige Technik ist als TCP-Aufspüren („TCP Snooping“) bekannt. (Siehe Z. Liu u.a., „Evaluation of TCP Vegas: Emulation and Experiment“) proc. ACM SIGCOMM' 95, August 1995, Seiten 185 bis 196; und H. Balakrishnan u.a., „Improving TCP/IP Performance over Wireless Networks“) proc. ACM SIGCOMM' 95, August 1995). Diese Idee ist gemäß [Fig. 2](#) gezeigt. Die Idee erfordert, dass ein Router oder anderer Knoten **10** zwischen der TCP-Quelleneinrichtung **11** und der TCP-Empfängereinrichtung **12** vorhanden ist. Der Knoten **10** sendet ACKs für TCP-Daten, die die Quelleneinrichtung gesendet hat, zurück zu der Quelleneinrichtung **11**, um der Quelleneinrichtung **11** die Illusion eines kurzen Verzögerungspfads zu vermitteln. Der Knoten **10** unterdrückt die von der Empfängereinrichtung **12** zurückkehrenden ACKs, und trägt für die Wiederübertragung jedweder Segmente zu der Empfängereinrichtung **12** Sorge, die hinter dem Router verloren gingen (d.h., in dem Abschnitt des Pfads zwischen dem Knoten **10** und der Empfängereinrichtung **12**). Dadurch vermittelt ein derartiger TCP-Aufspürknoten **10** („Snooping Node“) der TCP-Quelleneinrichtung **11** den Eindruck, dass der Knoten **10** der tatsächliche TCP-Empfänger ist.

**[0014]** [Fig. 2](#) zeigt den Fluss von Paketen (PKn) und von entsprechend nummerierten Bestätigungen (ACKn). In der gemäß [Fig. 2](#) gezeigten Anordnung erzeugt der Knoten, wenn ein Paket PK64 den Knoten **10** von der Quelleneinrichtung **11** erreicht, eine entsprechende Bestätigung ACK64 für das Paket und gibt jene zu der Quelleneinrichtung zurück. Der Knoten **10** speichert im Speicher **13** Pakete, die durch ihn empfangen, aber noch nicht durch die Empfängereinrichtung **12** bestätigt wurden. Der Knoten **10** sendet wiederum Pakete zu der Empfängereinrichtung (siehe Pakete PK59, PK60). Die Empfängereinrichtung gibt Bestätigungen für die Pakete zurück, die sie korrekt empfangen hat (ACK58, ACK59). Empfängt der Knoten **10** eine derartige Bestätigung (z.B. ACK57 wie gemäß [Fig. 2](#) gezeigt), dann verwirft er das entsprechende Paket aus seinem Speicher **13**. Der Aufspürknoten **10** muss daher eine beträchtliche Menge von Arbeit ableisten, nachdem er eine ACK zu der Quelleneinrichtung **11** sendet. Er muss das soeben empfangene Datensegment puffern, da es der Quelleneinrichtung **11** nun freisteht, ihre Kopie aufgrund der Bestätigung des Segments zu verwerfen, und daher muss der Router in der Lage sein, die volle Verantwortung für eine Wiederübertragung des Segments zu übernehmen, wenn dieses zwischen dem Router und der Empfängereinrichtung verloren geht. Deshalb können die Daten in dem Puffer nicht gelöscht werden, bis der Aufspürknoten die relevanten ACKs von der TCP-Empfängereinrichtung **12** erhält. Unglücklicherweise weist dieses Verfahren das große Problem auf, dass sich der Puffer in einem Aufspürknoten schnell füllen kann, wenn die verfügbare Bandbreite (wie zellulare Netzwerkressourcen) zwischen dem Knoten und der TCP-Empfängereinrichtung (Anbindung **14** gemäß [Fig. 2](#)) verringert. Das Problem ist gemäß [Fig. 3](#) gezeigt, die ein Paket PK69 zeigt, das den Puffer **13** in dem Knoten **10** zum Überlauf bringt.

**[0015]** Es sei darauf hingewiesen, dass das Erfordernis zur Speicherung von Segmenten für den Knoten **10**, und somit das Problem eines Pufferüberlaufs, aus der Tatsache erwächst, dass der Aufspürknoten eine Bestätigung erzeugt, die dem selben TCP-Segment entspricht, wie er es soeben empfangen hat.

**[0016]** Ein anderer Ansatz zur Bewältigung einiger der Probleme des TCP-Langsam-Startalgorithmus ist als aufgespaltetes TCP („Split TCP“) bekannt. In diesem Ansatz wird eine TCP-Verbindung mit einer speziellen TCP-Verbindung in viele TCP-Verbindungen unterteilt. Da jede TCP-Verbindung terminiert wird, ist das aufgespaltete TCP nicht für asymmetrische Pfade anfällig. Und in Fällen, in denen Anwendungen aktiv bei der TCP-Verbindungsverwaltung beteiligt sind (wie Netzcash-Bildung, „web caching“), funktioniert es gut. Ansonsten weist das aufgespaltete TCP die selben Probleme wie das TCP-Aufspüren auf, insbesondere das Einrichten von Puffern.

**[0017]** Es bespricht „TCP ACK pacing in ATM networks“ von N. Ghani eine Bestätigungsabschreitvorrichtung in einem Übertragungspfad. Die Vorrichtung moduliert die Zeitgabe zwischen TCP-ACK-Paketen, um Verwürfe während Blockierungszeitspannen zu begrenzen. Ein ACK-Paket wird durch die Abschreitvorrichtung abgefangen und verzögert, falls die Weiterleitungsanbindung blockiert ist.

**[0018]** Die Druckschrift WO 00/21231 offenbart eine puffernde Netzwerkfunktionseinheit, die sich in einem Übertragungspfad befindet. Die puffernde Funktionseinheit fängt Pakete ab und sendet eine Zustandsnachricht zu der sendenden Funktionseinheit. Die puffernde Funktionseinheit überträgt das empfangene Paket zu der empfangenden Funktionseinheit. Bei Empfang einer Bestätigung von der empfangenden Funktionseinheit, welche ein korrektes Empfangen des Pakets bestätigt, überträgt die puffernde Funktionseinheit eine weitere Zustandsnachricht zu der übertragenden Funktionseinheit, die die erfolgreiche Übertragung des Pakets anzeigt.

**[0019]** Es besteht deshalb ein Bedarf an einem verbesserten Verfahren zum Betrieb in Protokollen, wie dem TCP, um zumindest teilweise eines oder mehrere der vorstehend beschriebenen Probleme zu bewältigen.

**[0020]** Gemäß einer Ausgestaltung der Erfindung ist ein Datenkommunikationsknoten zum Betrieb in einem Kommunikationsweg zwischen einem Datagrammübertrager und einem Datagrammempfänger gemäß einem Protokoll bereitgestellt, wobei die Datagrammübertragung durch den Übertrager davon abhängig ist, dass er eine Bestätigungsnachricht für ein zuvor übertragenes Datagramm empfängt, wobei der Knoten eine Erfassungsvorrichtung zur Erfassung von Kommunikationen in dem Weg, einen Bestätigungserzeuger, der zur Erzeugung von Bestätigungsnachrichten für Datagramme und zur Übertragung dieser Nachrichten zu dem Übertrager angepasst ist, und eine Flussunterbrechungsvorrichtung, die zur Unterbrechung von Kommunikationen in dem Weg angepasst ist, umfasst, dadurch gekennzeichnet, dass der Knoten zum Betrieb in dem zweiten und zumindest in einem der ersten und dritten der nachstehenden Modi angepasst ist: ein erster Modus, bei dem er die Kommunikationen in dem Weg nicht unterbricht, ein zweiter Modus, bei dem er eine Bestätigungsnachricht für ein Datagramm auf das letzte an dem Knoten erfasste Datagramm hin folgend überträgt, und ein dritter Modus, bei dem er die Mitteilung von Bestätigungsnachrichten zu dem Übertrager verzögert.

**[0021]** In der ersten Betriebsart beeinflusst der Knoten vorzugsweise nicht die Kommunikation in dem Pfad, geeigneterweise so, dass Datagramme von dem Übertrager bzw. der Sendereinrichtung frei durch den Knoten zu dem Empfänger hin fließen können, und dass Bestätigungsnachrichten von dem Empfänger bzw. der Empfängereinrichtung frei zu der Sendereinrichtung hin fließen können.

**[0022]** In der zweiten und/oder dritten Betriebsart unterbricht der Knoten vorzugsweise die Kommunikation von Bestätigungsnachrichten von der Empfängereinrichtung zu der Sendereinrichtung. In der zweiten und/oder dritten Betriebsart unterbricht der Knoten vorzugsweise die Kommunikation von Datagrammen von der Sendereinrichtung zu der Empfängereinrichtung.

**[0023]** Der Knoten kann einen Speicher umfassen, der zur Speicherung von Datagrammen und/oder Bestätigungsnachrichten in der Lage ist. In der zweiten und/oder dritten Betriebsart speichert der Knoten vorzugsweise bei einer Übertragung von der Sendereinrichtung zu der Empfängereinrichtung erfasste Datagramme. Der Knoten kann dann in der zweiten und/oder dritten Betriebsart ein Datagramm zu der Empfängereinrichtung wieder übertragen, wenn er innerhalb einer eingestellten Zeitspanne keine Bestätigungsnachricht für das Datagramm von der Empfängereinrichtung erfasst. Bei Empfangen einer Bestätigungsnachricht für ein Datagramm kann der Knoten jenes Datagramm aus dem Speicher löschen.

**[0024]** Der Knoten umfasst vorzugsweise eine Betriebsartbestimmungseinrichtung zur Bestimmung der Betriebsart des Knotens. Die Betriebsartbestimmungseinrichtung kann vorzugsweise bestimmen, ob Bandbreite in dem Kommunikationspfad zur Verfügung steht, beispielsweise unbenutzte Bandbreite in dem Kommunikationspfad (vorzugsweise in beiden Richtungen über den Pfad) und/oder Kapazität in dem Speicher des Knotens (falls vorhanden) zur Verfügung steht. Steht eine derartige Kapazität zur Verfügung, dann kann die Betriebsartbestimmungseinrichtung vorzugsweise den Knoten zum Betrieb in der zweiten Betriebsart veranlassen. Die Betriebsartbestimmungseinrichtung kann vorzugsweise bestimmen, ob eine Blockierung in dem Kommunikationspfad vorliegt. Eine Blockierung in dem Kommunikationspfad kann durch Verzögerung bei einer Kommunikation über den Pfad (entweder von Datagrammen oder von Bestätigungen), überhöhten Datenverlust, überhöhte Anforderungen auf dem Pfad usw. angezeigt sein. Die Blockierung kann aufgrund von einer oder beider aus einer Erhöhung der Menge von über dem Pfad gerichteten Daten und einer Verringerung der Datenkapazität (Bandbreite) des Pfads auftreten. Bestimmt die Einrichtung, dass eine Blockierung vorliegt, dann kann sie vorzugsweise den Knoten zum Betrieb in der dritten Betriebsart veranlassen. Ansonsten (d.h., falls keine Kapazität zur Verfügung steht oder eine Blockierung in dem Kommunikationspfad vorliegt) kann die Bestimmungseinrichtung geeigneterweise den Knoten zum Eintreten in die erste Betriebsart veranlassen.

**[0025]** Der Knoten ist vorzugsweise zum Betrieb in jeder der ersten, zweiten und dritten Betriebsart der Lage.

**[0026]** Gemäß einer zweiten Ausgestaltung der Erfindung ist ein Verfahren für Datenkommunikation über einen Kommunikationsweg zwischen einem Datagrammübertrager (**20**) und einem Datagrammempfänger gemäß einem Protokoll bereitgestellt, wobei die Datagrammübertragung durch den Übertrager davon abhängig ist, dass er eine Bestätigungsnachricht für ein zuvor übertragenes Datagramm empfängt, wobei ein Knoten in dem Kommunikationsweg zwischen dem Übertrager und dem Empfänger lokalisiert ist, dadurch gekennzeichnet, dass das Verfahren den die Datagramme zu dem Empfänger übertragenden Übertrager, den die Datagramme erfassenden Knoten, und den Knoten umfasst, der eine Bestätigungsnachricht für ein dem neuesten an dem Knoten erfassten Datagramm nochfolgendes Datagramm zu dem Übertrager wiederholt überträgt.

**[0027]** Gemäß beiden Ausgestaltungen der Erfindung ist das Protokoll vorzugsweise ein Übertragungssteuerprotokoll (TCP). Das Protokoll stellt vorzugsweise eine Betriebsart bereit, in der der Übertrager bzw. die Sen-

dereinrichtung ein Datagramm nicht überträgt, bis sie eine Bestätigungsnachricht für ein ein gespeichertes Intervall zuvor übertragenes Datagramm empfangen hat, und in der eine Erhöhung des gespeicherten Intervalls besonders bevorzugt wird, falls eine Bestätigungsnachricht empfangen wird. Dies kann eine Langsam-Startbetriebsart sein.

**[0028]** Das gespeicherte Intervall ist vorzugsweise als eine Anzahl von Dateneinheiten ausgedrückt, beispielsweise eine Anzahl von Datagrammen oder eine Anzahl von Bits. Vorzugsweise ist das Intervall als eine Anzahl von Datagrammen ausgedrückt. Dann stellt das Protokoll vorzugsweise eine Betriebsart bereit, in der ein Datagramm nicht übertragen wird, bis eine Bestätigung für das Datagramm empfangen wird, das n Datagramme zuvor übertragen wurde, wobei n die Anzahl durch das gespeicherte Intervall ausgedrückter Datagramme ist.

**[0029]** Gemäß beiden Ausgestaltungen der Erfindung ist zumindest ein Teil des Kommunikationspfads vorzugsweise durch eine Funkanbindung bereitgestellt, insbesondere ist eine zellulare Funkanbindung bevorzugt.

**[0030]** Die Erfindung ist nachstehend beispielhaft unter Bezugnahme auf die beiliegenden Zeichnungen beschrieben. Es zeigen:

**[0031]** [Fig. 1](#) den TCP-Langsam-Startalgorithmus,

**[0032]** die [Fig. 2](#) und [Fig. 3](#) Ausgestaltungen des TCP-Aufspürvorgangs,

**[0033]** [Fig. 4](#) die Architektur eines Kommunikationssystems schematisch,

**[0034]** [Fig. 5](#) bevorzugte Betriebsartanordnungen,

**[0035]** die [Fig. 6](#) bis [Fig. 9](#) den Betrieb des Systems gemäß [Fig. 4](#) in den Betriebsarten gemäß [Fig. 5](#), und

**[0036]** [Fig. 10](#) eine Strategie zur Auswahl einer Betriebsart.

**[0037]** [Fig. 4](#) zeigt eine schematische Ansicht eines Kommunikationssystems. Das Kommunikationssystem umfasst eine Sendereinrichtung **20**, die durch einen bidirektionalen Kommunikationspfad **22** mit einer Empfängereinrichtung **21** verbunden ist. Die Sendereinrichtung kann eine Nachricht in Pakete zerlegen und sie unter Verwendung des TCP-Protokolls zur Empfängereinrichtung senden. Zur leichteren Erklärung ist der Kommunikationspfad **22** als einen ausgehenden Kanal **23** und einen Rückführkanal **24** umfassend gezeigt, der zum Senden der Signale in entgegengesetzten Richtungen in der Lage ist, der Pfad kann aber in der Praxis bei jedem Punkt entlang seiner Länge ein einzelner physikalischer Kanal sein. Der Pfad kann durch eine Kette einer oder mehrerer in Reihe gestellter Anbindungen bereitgestellt sein, die von der selben oder verschiedenen physikalischen Arten sein könnten (z.B. Festdrähte, Funkkanäle oder optische Kanäle).

**[0038]** In den Kommunikationspfad **22** ist ein Knoten **25** gesetzt. Der Knoten **25** umfasst Schalter **26**, **27**, die in den ausgehenden bzw. Rückführkanal **23**, **24** gesetzt sind. Die Schalter funktionieren unter der Steuerung einer Steuereinrichtung **28**. Eine Paketerfassungseinrichtung **29** ist mit einem Ausgang des Schalters **26** verbunden. Es ist eine Bestätigungserfassungseinrichtung **30** mit einem Ausgang des Schalters **27** verbunden. Die anderen Ausgänge der Schalter **26**, **27** reichen zu der Empfängereinrichtung **21** bzw. der Sendereinrichtung **20** durch. Die Ausgänge der Erfassungseinrichtung **29**, **30** reichen zu der Steuereinrichtung **28**. Der Knoten enthält ebenso einen Speicher **31**, in dem Daten durch die Steuereinrichtung **28** gespeichert werden können, und eine Bestätigungserfassungseinrichtung **36**, die Bestätigungsnachrichten für Pakete unter der Weisung durch die Steuereinrichtung **28** erzeugen kann. In einer Einstellung („A“) des Schalters **26** werden an dem Knoten auf Kanal **23** ankommende Signale (von Eingang **32**) direkt zu dem Ausgang **33** des Knotens **25** auf Kanal **23** verwiesen. In der anderen Einstellung („B“) des Schalters **26** werden Signale von Eingang **32** vom Ausgang **33** isoliert; stattdessen wird der Ausgang des Speichers **31** zu dem Ausgang **33** verwiesen. In einer Einstellung („X“) des Schalters **27** werden an dem Knoten auf Kanal **24** ankommende Signale (von Eingang **34**) direkt zu dem Ausgang **35** des Knotens **25** auf Kanal **24** verwiesen. In der anderen Einstellung („Y“) des Schalters **27** werden Signale vom Eingang **34** von Ausgang **35** isoliert; stattdessen wird der Ausgang der Bestätigungserzeugungseinrichtung **36** zu dem Ausgang **35** verwiesen.

**[0039]** Die Steuereinrichtung **28** funktioniert unter der Überwachung durch die Betriebsartauswahleinrichtung **37**, die die Kanäle **23** und **24** überwacht, um den Zustand des Kommunikationssystems zu bestimmen, und ist durch ein externes Signal **38** steuerbar. Die Betriebsartauswahleinrichtung **37** kann ebenso die Kanäle **23** und



**24** überwachen, insbesondere wenn diese einen drahtlosen Pfad oder einen anderen Pfad enthalten, der einen Flaschenhals darstellen könnte.

**[0040]** Für die Klarheit der Beschreibung ist der Knoten **25** nachstehend als bestimmte Einzelvorrichtungen umfassend beschrieben. Der Knoten **25** könnte jedoch als ein Block oder mehrere Blöcke einer Funktionalität bereitgestellt sein, die von jenen gemäß [Fig. 4](#) gezeigten verschieden ist. Die Funktionen des Knotens **25** könnten zur Gänze oder teilweise durch auf einem oder mehreren Mikroprozessoren laufende Software bereitgestellt sein.

**[0041]** Bei Betrieb weist der Knoten **25** drei Betriebsarten auf, die hiernach „normale“ Betriebsart, „schnelle“ Betriebsart und „verfrühte Bestätigungs-“ Betriebsart genannt sind.

- Bei der normalen Betriebsart wird der Schalter **26** auf Einstellung A und Schalter **17** wird auf Einstellung X gesetzt. Dies erlaubt Signalen von der Sendereinrichtung **20** ein direktes Passieren durch den Knoten zu der Empfängereinrichtung **21**, und Signalen von der Empfängereinrichtung ein direktes Passieren zu der Sendereinrichtung.
- Bei der schnellen Betriebsart ist Schalter **26** auf Einstellung A, Schalter **27** auf Einstellung Y gesetzt. Empfängt der Knoten ein Paket von der Sendereinrichtung **20**, dann leitet es die Steuereinrichtung **28** zu der Empfängereinrichtung weiter. Die Empfängereinrichtung erzeugt Bestätigungen für die Pakete, die sie empfängt. Wird ein Paket bestätigt, leitet der Knoten **25** die Bestätigung nach einer Verzögerung zu der Sendereinrichtung weiter. Der Knoten **25** kann außerdem das von der Sendereinrichtung **21** empfangene Paket speichern, und es wieder zu der Empfängereinrichtung **21** senden, falls nach einer eingestellten Zeitspanne keine Bestätigung empfangen wird.
- Bei der verfrühten Bestätigung sind die Schalter **26** bzw. **27** auf die Einstellung B bzw. Y gesetzt. Dieser Vorgang ist nachstehend ausführlicher beschrieben.

**[0042]** Bei der normalen Betriebsart funktioniert die Anbindung **22** zwischen der Sendereinrichtung **20** und der Empfängereinrichtung **21** auf eine Ende-zu-Ende-Weise, als ob der Knoten **25** nicht vorhanden wäre. Bei der verfrühten Bestätigungsbetriebsart stellt der Knoten die Fähigkeit zur Beschleunigung der Erhöhung bei der Übertragungsrate durch die Sendereinrichtung unter dem TCP-Langsam-Start durch Senden sofortiger Bestätigungen bereit. Bei der schnellen Betriebsart kann der Knoten die Übertragungsrate durch den Sender verringern, um der verringerten Kapazität über die Anbindung, oder um dem Füllen der Pufferspeicher der Knoten gerecht zu werden. Die Verfügbarkeit von mehr als einer dieser Betriebsarten versieht den Knoten mit der Fähigkeit zur Funktionssteigerung der TCP-Anbindung in einer Fülle von Umständen, wie nachstehend ausführlicher beschrieben. Insbesondere ermöglicht die Reihe von Betriebsarten ein Betriebsmodell, das die Mängel des TCP-Langsam-Startalgorithmus umgehen und den Pufferaufbau in dem Weiterleistungsknoten entlasten kann.

**[0043]** Die Betriebsartauswahleinrichtung **37** überwacht die Anbindung **22** und veranlasst die Steuereinrichtung **28** abhängig von den Anbindungsbedingungen zur Annahme einer der verfügbaren Betriebsarten. [Fig. 5](#) zeigt eine bevorzugte Strategie. Bei der normalen Betriebsart wählt die Betriebsartauswahleinrichtung **37** die normale Betriebsart aus. Weist der Pufferspeicher **31** verfügbaren Platz auf und steht verfügbare Bandbreite auf der Anbindung in beiden Richtungen zur Verfügung, dann wählt die Betriebsartauswahleinrichtung die verfrühte Bestätigungsbetriebsart zum Ansteuern der Paketübertragungsrate zu deren Erhöhung aus. Gibt es eine Blockierung auf der Anbindung, oder ist der Pufferspeicher voll, dann wählt die Betriebsartauswahleinrichtung die schnelle Betriebsart zum Ansteuern der Pakettransmissionsrate zu deren Verringerung aus.

**[0044]** Die Wirkung der verfügbaren Betriebsarten ist nachstehend ausführlicher beschrieben.

**[0045]** Steht Bandbreite oberhalb einer eingestellten Schwelle zur Verfügung, und erfasst die Betriebsartauswahleinrichtung **37** keinen Paketverlust aufgrund einer Blockierung, dann wird die verfrühte Bestätigungsbetriebsart ausgewählt. Wird ein Paket durch Knoten **25** empfangen und durch die Erfassungseinrichtung **29** erfasst, dann steuert die Steuereinrichtung **28** die Bestätigungserzeugungseinrichtung **36** zur Erzeugung einer „verfrühten“ Bestätigung, und gibt diese zu der Sendereinrichtung **20** ab. Die in der verfrühten Bestätigungsnachricht spezifizierte Paketnummer ist jene eines Pakets, das noch nicht durch den Knoten **25** empfangen wurde (aber zur Kompatibilität mit den meisten Protokollen eines ist, dass wahrscheinlich durch die Sendereinrichtung gesendet wurde, bevor die Bestätigung diese erreicht). Während sich jenes tatsächliche Datenpaket auf seinem Weg von der Bestätigungssteuereinrichtung zu der TCP-Empfängereinrichtung befindet, befindet sich dadurch die durch die TCP-Empfängereinrichtung erzeugte Bestätigung auf ihrem Weg von der Empfängereinrichtung zu der Bestätigungssteuereinrichtung, oder nicht. Die Anzahl, um die die Bestätigungsanzahl dem letzten durch den Knoten empfangenen Paket vorausseilt, hängt von der zur Verfügung stehenden

Bandbreite, der Natur des Paketverlusts und dem zur Verfügung stehenden Pufferplatz in dem Netzwerkknoten ab. Dadurch sagt die Bestätigungssteuereinrichtung den Bedarf an einem Senden einer wahren Bestätigungsnachricht für ein bestimmtes Paket voraus und sendet eine durch sie selbst erzeugte, vorausgesehene Bestätigung zu der TCP-Quelleneinrichtung, so dass die TCP-Quelleneinrichtung eine Bestätigung früher erhält, als sie es sogar bei Verwendung eines Aufspürknotens würde. Der Knoten **25** leitet dann das Paket bei Empfang zu der Empfängereinrichtung weiter, oder erlaubt dem Paket einfach das Passieren durch den Knoten **25**. Erreicht die durch die TCP-Empfängereinrichtung erzeugte, entsprechende Bestätigungsnachricht den Knoten, dann verwirft die Steuereinrichtung **28** es aus dem Speicher **31**.

**[0046]** Um die Effizienz dieser Betriebsart zu zeigen, kann mit Definieren der Sequenznummer des ankommenden Pakets bei dem Knoten als  $PK(i)$ , der ankommenden Bestätigung als  $ACK^1(i)$  bzw. der erzeugten Bestätigung als  $ACK^0(i)$  die  $ACK^0(i)$  als eine Funktion der zur Verfügung stehenden Bandbreite  $BW_a$  und des zur Verfügung stehenden Pufferplatzes  $BS_a$  und von  $ACK^0(i-1)$  eingestellt werden. Sie kann formuliert werden als

$$ACK^0(i) = ACK^0(i-1) + f(BW_a, BS_a)$$

$$ACK^1(i) \leq ACK^0(i) \leq PK(i) \quad (1)$$

**[0047]** Die Funktion  $f$  kann beispielsweise sein

$$f = (PK - ACK^0(i-1)) \left( a_1 \frac{BS_a}{BS_{\max}} + a_2 BW_a \right) \quad (2)$$

wobei  $BS_{\max}$  der maximale Pufferplatz ist, und  $a_1$  und  $a_2$  Konstanten sind. Es sei darauf hingewiesen, dass sich das verfrühte Bestätigungsverfahren zu dem herkömmlichen TCP-Aufspüren verringert, falls  $f = PK - ACK^0(i-1)$ .

**[0048]** Bei der verfrühten Bestätigungsbetriebsart wird gegenüber der TCP-Quelleneinrichtung schnell bestätigt, dass ihre Pakete empfangen wurden, und die Quelleneinrichtung beginnt mit einer Erhöhung ihrer Übertragungsrates. Dies verringert wiederum die für das TCP-Protokoll zum Erreichen seiner vollen Geschwindigkeit erforderliche Zeit und verbessert den allgemeinen Durchsatz.

**[0049]** Da die Funktion  $f$  (Gleichung 2) die zur Verfügung stehende Bandbreite und die Pufferbelegung berücksichtigt, bewältigt sie die Probleme des TCP-Aufspürverfahrens. Die übertragene  $ACK^0$  nähert sich effektiv der  $ACK^1$  an, wenn sich der zur Verfügung stehende Pufferplatz und/oder die Bandbreite verringern/verringert. Steht keine unbenutzte Bandbreite zur Verfügung, dann ist  $ACK^1$  gleich  $ACK^0$ , und der Normalbetrieb des TCP ist erreicht. Erhöhen/Erhöht sich demgegenüber die zur Verfügung stehende Bandbreite und/oder der Pufferplatz, dann nähert sich die  $ACK^0$   $PK$  an. Falls viel unbenutzte Bandbreite zur Verfügung steht, funktioniert dadurch der Knoten **25** in der verfrühten Bestätigungsbetriebsart auf eine dem vorstehend beschriebenen TCP-Aufspürverfahren ähnliche Weise.

**[0050]** Die normale Betriebsart wird verwendet, wenn das TCP-System bei der Sendereinrichtung erst einmal die volle Geschwindigkeit erreicht hat, und es im Wesentlichen keine unbenutzte, zur Verfügung stehende Bandbreite in der Richtung von der Sendereinrichtung zu der Empfängereinrichtung gibt. Die Ankunft eines TCP-Pakets bei dem Knoten löst nicht länger eine Erzeugung einer verfrühten Bestätigung aus. Stattdessen lässt die Bestätigungssteuereinrichtung einfach die TCP-Pakete und die entsprechenden Bestätigungen auf eine natürliche, ungestörte Weise durch sich hindurch fließen.

**[0051]** Bei der schnellen TCP-Betriebsart, die verwendet wird, falls die Betriebsartauswahleinrichtung einen Paketverlust aufgrund einer Blockierung erfasst, verzögert die Bestätigungssteuereinrichtung die Zurückgabe von Bestätigungen zu der Sendereinrichtung. Dies könnte durch den Knoten durch Empfangen von Bestätigungen, durch deren Speicherung für eine Zeitspanne, und dann durch Weiterleiten zu der Sendereinrichtung erreicht werden; oder indem der Knoten eine Verzögerung einführt, bevor er selbst eine Bestätigung erzeugt. Der Vorgang des Verzögerns von Bestätigungen könnte auf eine wie in der Druckschrift WO 99/04536 beschriebene Weise durchgeführt werden, deren Inhalte hier durch Bezugnahme mit aufgenommen sind. Die Wirkung des Verzögerns von Bestätigungen liegt in der Verringerung einer Blockierung und damit des Paketverlusts.

**[0052]** [Fig. 6](#) zeigt weiterhin die verfrühte Bestätigungsbetriebsart. Auf der Grundlage der durch sie erfassten Informationen bezüglich der Natur des Paketverlusts und der zur Verfügung stehenden Bandbreite wählte die



Betriebsartauswahleinrichtung des Knotens **25** die verfrühte Bestätigungsbetriebsart aus, und veranlasste die Steuereinrichtung zum Betrieb in jener Betriebsart. Ein TCP-Paket mit der Sequenznummer PK kommt von der Quelleneinrichtung zu dem Knoten. Die Bestätigungssteuereinrichtung empfängt das TCP-Paket. Eine Bestätigung  $ACK^0$  wird durch den Knoten zu der TCP-Quelleneinrichtung gesendet. Die Bestätigung könnte zur Gänze durch den Knoten erzeugt werden, oder ein geeignetes Bestätigungspaket, das den Knoten zu jenem Zeitpunkt durchläuft (das durch die TCP-Empfängereinrichtung erzeugt wurde), könnte abgefangen und zum Führen der  $ACK^0$  modifiziert werden. Ein Verfahren für den Knoten zur Verwendung bei einer Berechnung des Werts von  $ACK^0$  ist vorstehend in Gleichung 1 gezeigt.

**[0053]** Das Senden von  $ACK^0$  und dergleichen Veranlasst die TCP-Quelleneinrichtung zu einer schnellen Beschleunigung, da sie jetzt Bestätigungen schneller als mit anderen Verfahren empfängt. Es gibt eine Ersparnis von deutlich mehr als einer Umlaufzeit zwischen dem Knoten und der TCP-Empfängereinrichtung. Die TCP-Quelleneinrichtung sieht, dass dort ein kürzerer Abstand zu der Empfängereinrichtung als selbst der Abstand zu dem Knoten zu sein scheint.

**[0054]** Das TCP-Paket PK wird zu der TCP-Empfängereinrichtung weitergeleitet. Es sei darauf hingewiesen, dass die Erzeugung und die Übertragung (oder das Abfangen und die Änderung) von  $ACK^0$  und das Weiterleiten (oder das Zulassen des Passierens) von PK ebenso in umgekehrter Reihenfolge auftreten kann. Eine Kopie des TCP-Pakets PK wird in Speicher **31** gespeichert, so dass es durch den Knoten wieder übertragen werden kann, falls das Paket zwischen der  $ACK$ -Steuereinrichtung und der TCP-Empfängereinrichtung verloren geht.

**[0055]** Die TCP-Empfängereinrichtung empfängt das TCP-Paket PK und bestätigt es wie üblich durch Senden einer entsprechenden  $ACK^1$  zu der TCP-Quelleneinrichtung hin. Die  $ACK$ -Steuereinrichtung fängt schließlich  $ACK^1$  ab und verwirft diese, oder verwendet die  $ACK$  für eine andere  $ACK^0$ .

**[0056]** Die  $ACK$ -Steuereinrichtung muss ebenso für Wiederübertragungen Sorge tragen. Daher überwacht sie den Fluss von Bestätigungen  $ACK^1$  von der TCP-Empfängereinrichtungen. Ist während einer spezifizierten Zeit keine  $ACK$  hervorgekommen, dann überträgt sie das TCP-Paket mit der Sequenznummer PK wieder. Es sei darauf hingewiesen, dass es hier möglich ist, die Standard TCP-Wiederübertragungsmechanismen wie in der IETF RFT 793, „Transport Control Protocol“, September 1981, spezifiziert, zu verwenden.

**[0057]** [Fig. 7](#) zeigt ein ausführlicheres Beispiel einer Übertragung gemäß der Erfindung. Es ist hier angenommen, dass genügend Bandbreite und Pufferplatz für eine weitere Erhöhung der Übertragungsraten zur Verfügung stehen. Es sei angenommen, dass ein Paket mit der Sequenznummer 64 bei dem Knoten von der Quelleneinrichtung ankommt, während eine Bestätigungsnachricht mit der Nummer 57 von der Empfängereinrichtung ankommt. Auf der Grundlage der zur Verfügung stehenden Bandbreite bestimmte der Knoten, die verfrühte Bestätigungsbetriebsart zu verwenden. Er erzeugt eine Bestätigungsnachricht mit der Nummer 60 und überträgt sie zu der TCP-Quelleneinrichtung. Es sei darauf hingewiesen, dass die Nummer dieser Nachricht unterhalb von 64 (die Nummer des angekommenen Pakets) liegt. Die von der TCP-Empfängereinrichtung empfangene Bestätigungsnachricht mit 57 wird verworfen (oder sie hätte verändert werden können, um den Wert 60 zu tragen). Das Paket 64 wird in dem Pufferspeicher des Knotens für den Fall gespeichert, dass es wieder zu übertragen ist. Das Paket 57 wird aus dem Puffer entfernt, da es nun empfangen wurde.

**[0058]** In einem gemäß [Fig. 8](#) gezeigten, späteren Stadium steht keine unbenutzte Bandbreite mehr zur Verfügung. Deshalb werden von der TCP-Empfängereinrichtung zu dem Knoten kommende Bestätigungen einfach durchgelassen, so dass  $ACK^0 = ACK^1$ . Dies entspricht der normalen TCP-Betriebsart.

**[0059]** In einem gemäß [Fig. 9](#) gezeigten, anderen Stadium kann eine Blockierung auftreten. Der Puffer in dem Netzwerkknoten wurde voll und/oder die unbenutzte Bandbreite ist klein und/oder es werden Fehler aufgrund einer Blockierung erfasst. Dann beginnt der Knoten, von der TCP-Empfängereinrichtung zu der TCP-Quelleneinrichtung fließende Bestätigungen ( $ACK$ 's) zur Verringerung der Blockierung zu verzögern.

**[0060]** Die Erfindung ist besonders zur Verwendung in Verbindung mit Funknetzwerken geeignet, wie GPRS oder zellularen Netzwerken der dritten Generation (3GN), oder anderen Systemen, in denen es eine Anbindung geben kann, die eine beträchtlich variierende Kapazität für die TCP-Verbindung aufweist (beispielsweise aufgrund von sich verändernden Interferenzpegeln). In einem Funksystem könnte der Knoten **25** in einer Kommunikationsanbindung nahe der Funkschnittstelle angeordnet sein. Der Knoten könnte ein Signal empfangen, das die momentan über die Funkschnittstelle zur Verfügung stehenden Kapazität angibt. Der Knoten könnte beispielsweise bei der Funknetzwerksteuereinrichtung (RNC, „Radio Network Controller“) eines 3GN-Systems

angeordnet sein. Der Knoten überwacht die zur Verfügung stehende Kapazität. Gibt es viel freie Kapazität für von außerhalb des 3GN-Systems ankommenden TCP-Verkehr, dann beginnt der Knoten mit einer Erzeugung von verfrühten Bestätigungen, selbst wenn sich die eigentlichen Oktette immer noch auf ihrem Weg über die Funkschnittstelle befinden. Dadurch erhält die TCP-Sendereinrichtung Bestätigungen eher, beschleunigt ihre Übertragungen, und nutzt die zur Verfügung stehende Kapazität besser. Demgegenüber kann der Knoten während einer Blockierung (beispielsweise aufgrund schlechter Funkbedingungen) in die schnelle TCP-Betriebsart eintreten.

**[0061]** Das verfrühte Bestätigungsverfahren kann den TCP-Langsam-Startvorgang beschleunigen, und auf diese Weise die zur Verfügung stehende Bandbreite effizienter nutzen, sowie die Pufferaufbauprobleme lindern, die in früheren TCP-Aufspürsystemen angetroffen werden können. Das Verfahren wird durch einen zwischen einer TCP-Sendereinrichtung und einer TCP-Empfängereinrichtung lokalisierten Knoten ausgeführt. Der Knoten erzeugt Bestätigungen und gibt sie zu der TCP-Sendereinrichtung für Datenpakete zurück, die den Knoten noch nicht erreicht, und die deshalb durch die TCP-Empfängereinrichtung noch nicht empfangen und bestätigt wurden. Dies kann die Zeit verringern, die die TCP-Sendereinrichtung auf eine Bestätigung warten muss, und ermöglicht der TCP-Sendereinrichtung somit, ihre Übertragungen schneller als ohne diesen Mechanismus zu beschleunigen. Die Sequenz von Bestätigungen wird gemäß der zur Verfügung stehenden Bandbreite und dem Pufferplatzes, einer Messung einer Sequenznummer angekommener Pakete und Ankunftsbestätigungen zu dem Knoten eingestellt.

**[0062]** Das verfrühte Bestätigungsverfahren kann in einem Knoten als Ausführungsbeispiel vorliegen, der zum Betrieb in zwei oder mehreren (vorzugsweise zumindest drei) Betriebsarten in der Lage ist, beispielsweise: der verfrühten Bestätigungsbetriebsart, einer normalen TCP-Betriebsart und einer schnellen TCP-Betriebsart. Die Entscheidung bezüglich der zu verwendenden Betriebsart kann beispielsweise auf der zur Verfügung stehenden, unbenutzten Bandbreite, auf Pufferbelegungsniveaus und auf einer Blockierung basieren. [Fig. 10](#) zeigt ein Ablaufdiagramm eines Beispiels eines Entscheidungsvorgangs darüber, welche Betriebsart zu verwenden ist.

**[0063]** Die Erfindung ist nicht auf die vorstehend beschriebenen Beispiele beschränkt.

**[0064]** Die Anmelderin macht auf die Tatsache aufmerksam, dass die Erfindung jedwedes Merkmal oder jedwede Kombination von Merkmalen, die hier entweder implizit oder explizit offenbart sind, oder jedwede Verallgemeinerung derer enthalten kann.

### Patentansprüche

1. Datenkommunikationsknoten (**25**) zum Betrieb in einem Kommunikationsweg (**22**) zwischen einem Datagrammübertrager (**20**) und einem Datagrammempfänger (**21**) gemäß einem Protokoll, wobei die Datagrammübertragung durch den Übertrager davon abhängig ist, dass er eine Bestätigungsnachricht für ein zuvor übertragenes Datagramm empfängt, wobei der Knoten umfasst:

eine Erfassungsvorrichtung (**29**) zur Erfassung von Kommunikationen in dem Weg,  
einen Bestätigungserzeuger (**36**), der zur Erzeugung von Bestätigungsnachrichten für Datagramme und zur Übertragung dieser Nachrichten zu dem Übertrager angepasst ist, und  
eine Flussunterbrechungsvorrichtung, die zur Unterbrechung von Kommunikationen in dem Weg angepasst ist,

**dadurch gekennzeichnet**, dass der Knoten zum Betrieb in dem zweiten und zumindest in einem der ersten und dritten der nachstehenden Modi angepasst ist:

- ein erster Modus, bei dem er die Kommunikationen in dem Weg nicht unterbricht,
- ein zweiter Modus, bei dem er eine Bestätigungsnachricht für ein Datagramm auf das letzte an dem Knoten erfasste Datagramm hin folgend überträgt, und
- ein dritter Modus, bei dem er die Mitteilung von Bestätigungsnachrichten zu dem Übertrager verzögert.

2. Datenkommunikationsknoten gemäß Anspruch 1, wobei in dem zweiten und/oder dritten Modus der Knoten die Mitteilung von Bestätigungsnachrichten von dem Empfänger zu dem Übertrager unterbricht.

3. Datenkommunikationsknoten gemäß Anspruch 1 oder 2, der einen Speicher umfasst, und wobei in dem zweiten und/oder dritten Modus der Knoten in dem Speicher Datagramme speichert, die in einer Übertragung von dem Übertrager zu dem Empfänger erfasst sind.

4. Datenkommunikationsknoten gemäß Anspruch 3, wobei in dem zweiten und/oder dritten Modus der

Knoten die Mitteilung von Datagrammen von dem Übertrager zu dem Empfänger unterbricht.

5. Datenkommunikationsknoten gemäß Anspruch 3 oder 4, wobei der Knoten in dem zweiten und/oder dritten Modus, bei nicht erfasster Bestätigungsnachricht für ein Datagramm von dem Empfänger innerhalb einer eingestellten Zeitspanne, dieses Datagramm wieder zu dem Empfänger überträgt.

6. Datenkommunikationsknoten gemäß einem der Ansprüche 3 bis 5, wobei der Knoten in dem zweiten und/oder dritten Modus, bei Empfang einer Bestätigungsnachricht für ein Datagramm, dieses Datagramm aus dem Speicher löscht.

7. Datenkommunikationsknoten gemäß einem der vorstehenden Ansprüche, der eine Modusbestimmungseinrichtung (**37**) zur Bestimmung des Betriebsmodus des Knotens umfasst.

8. Datenkommunikationsknoten gemäß Anspruch 7, wobei die Modusbestimmungseinrichtung zur Bestimmung in der Lage ist, ob Kapazität in dem Kommunikationsweg verfügbar ist, und den Knoten bei Verfügbarkeit von Kapazität zum Betrieb in dem zweiten Modus veranlassen kann.

9. Datenkommunikationsknoten gemäß Anspruch 7 oder 8, wobei die Modusbestimmungseinrichtung zur Bestimmung in der Lage ist, ob es eine Blockierung in dem Kommunikationsweg gibt und den Knoten bei Vorhandensein einer Blockierung zum Betrieb in dem dritten Modus veranlassen kann.

10. Datenkommunikationsknoten gemäß Anspruch 9 in Abhängigkeit von Anspruch 8, wobei die Bestimmungseinrichtung bei Nichtverfügbarkeit von Kapazität oder Nichtvorhandensein einer Blockierung in dem Kommunikationsweg den Knoten zum Eintritt in den ersten Modus veranlassen kann.

11. Datenkommunikationsknoten gemäß einem der vorstehenden Ansprüche, wobei der Knoten zum Betrieb in jedem des ersten, zweiten oder dritten Modus in der Lage ist.

12. Datenkommunikationsknoten gemäß einem der vorstehenden Ansprüche, wobei das Protokoll ein TCP-Protokoll (TCP, „transmission control protocol“) ist.

13. Datenkommunikationsknoten gemäß einem der vorstehenden Ansprüche, wobei das Protokoll eine Betriebsart bereitstellt, in der der Übertrager kein Datagramm überträgt, bis er eine Bestätigungsnachricht für ein um ein gespeichertes Intervall zuvor übertragenes Datagramm empfangen hat.

14. Datenkommunikationsknoten gemäß Anspruch 13, wobei in der Betriebsart das Protokoll eine Erhöhung des gespeicherten Intervalls bereitstellt, wenn eine Bestätigungsnachricht empfangen wird.

15. Datenkommunikationsknoten gemäß Anspruch 13 oder 14, wobei die Betriebsart eine slow-start-Betriebsart (Langsam-Start, „slow start“) ist.

16. Datenkommunikationsknoten gemäß einem der vorstehenden Ansprüche, wobei zumindest ein Teil des Kommunikationswegs eine Funkverbindung darstellt.

17. Datenkommunikationsknoten gemäß Anspruch 16, wobei die Funkverbindung eine zellulare Funkverbindung ist.

18. Verfahren für Datenkommunikation über einen Kommunikationsweg (**22**) zwischen einem Datagrammübertrager (**20**) und einem Datagrammempfänger (**21**) gemäß einem Protokoll, wobei die Datagrammübertragung durch den Übertrager davon abhängig ist, dass er eine Bestätigungsnachricht für ein zuvor übertragenes Datagramm empfängt, wobei ein Knoten (**25**) in dem Kommunikationsweg zwischen dem Übertrager und dem Empfänger lokalisiert ist, dadurch gekennzeichnet, dass das Verfahren umfasst:  
den die Datagramme zu dem Empfänger übertragenden Übertrager,  
den die Datagramme erfassenden Knoten, und  
den Knoten, der eine Bestätigungsnachricht für ein dem neuesten an dem Knoten erfassten Datagramm nachfolgendes Datagramm zu dem Übertrager wiederholt überträgt.

19. Verfahren gemäß Anspruch 18, wobei das Protokoll ein TCP-Protokoll ist.

20. Verfahren gemäß Anspruch 18 oder 19, wobei das Protokoll eine Betriebsart bereitstellt, in der der

Übertrager kein Datagramm überträgt, bis er eine Bestätigungsnachricht für ein um ein gespeichertes Intervall zuvor übertragenes Datagramm empfangen hat.

21. Verfahren gemäß Anspruch 20, wobei in der Betriebsart das Protokoll eine Erhöhung des gespeicherten Intervalls bereitstellt, wenn eine Bestätigungsnachricht empfangen wird.

22. Verfahren gemäß Anspruch 20 oder 21, wobei die Betriebsart eine slow-start-Betriebsart ist.

23. Verfahren gemäß einem der vorstehenden Ansprüche, wobei zumindest ein Teil des Kommunikationswegs eine Funkverbindung darstellt.

24. Verfahren gemäß Anspruch 23, wobei die Funkverbindung eine zellulare Funkverbindung ist.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

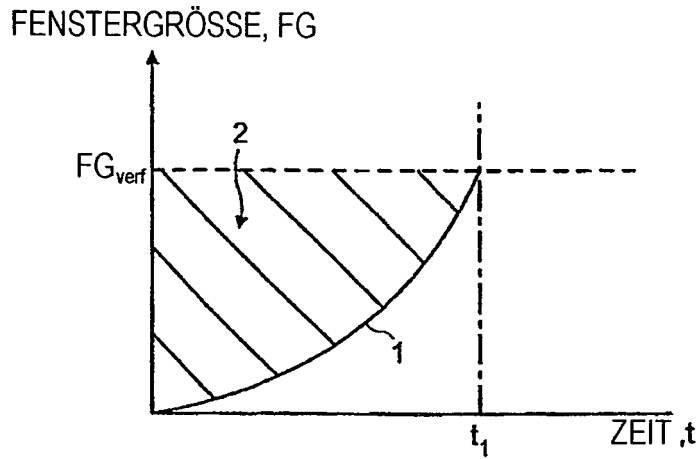


FIG. 1

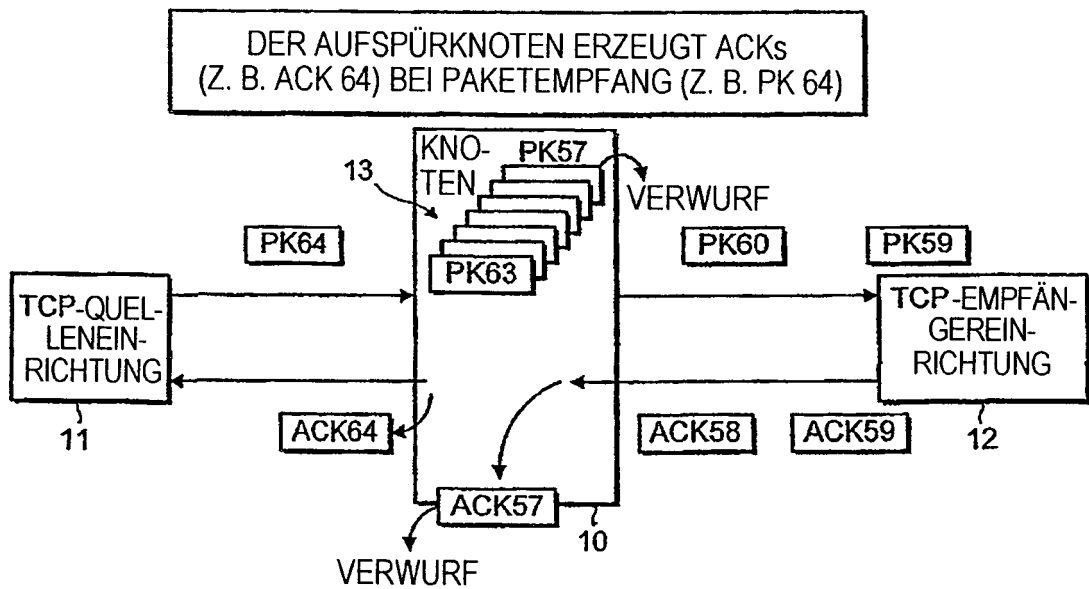
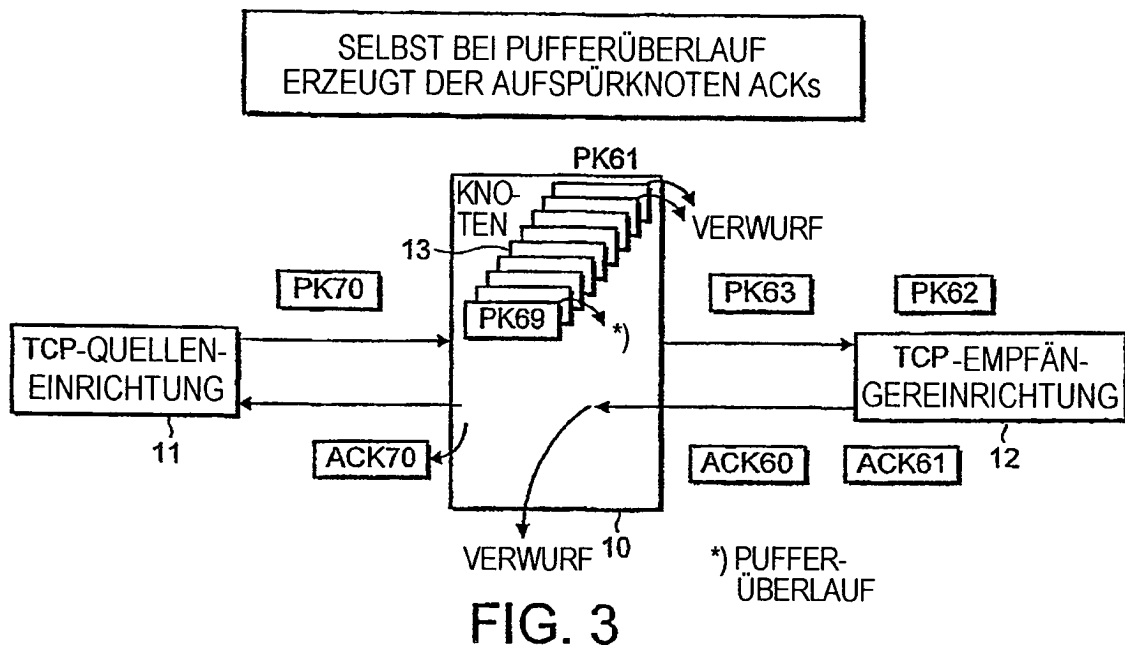


FIG. 2





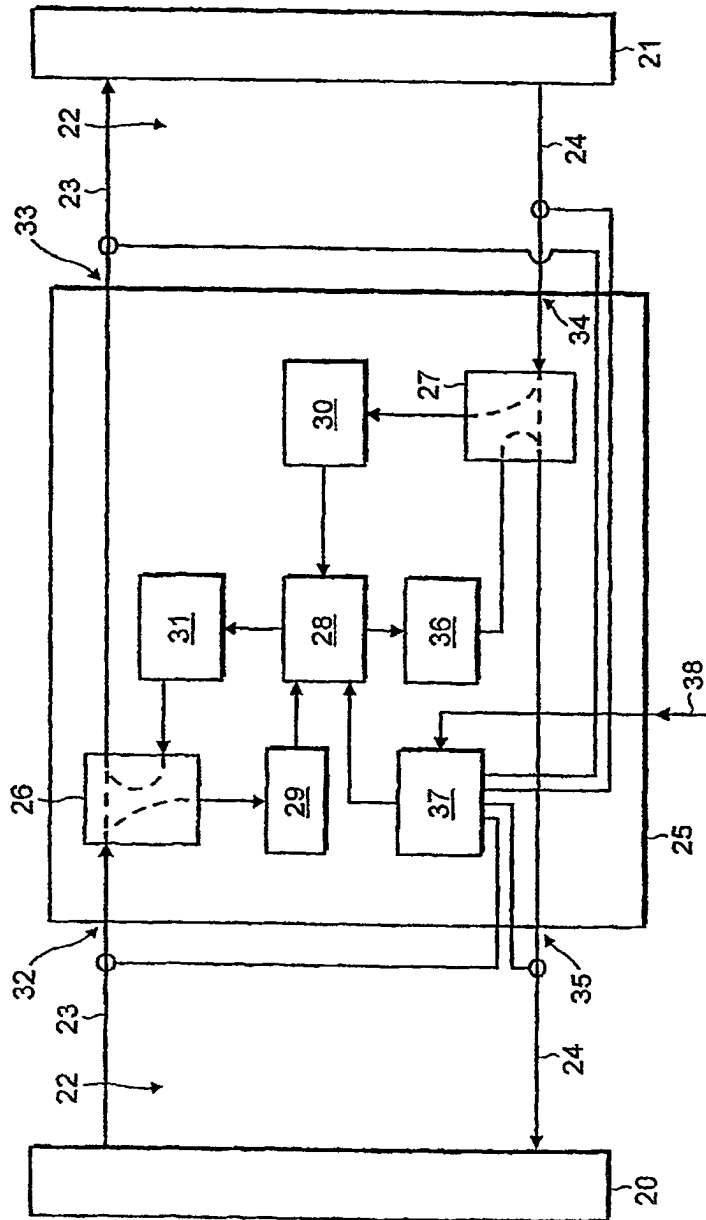
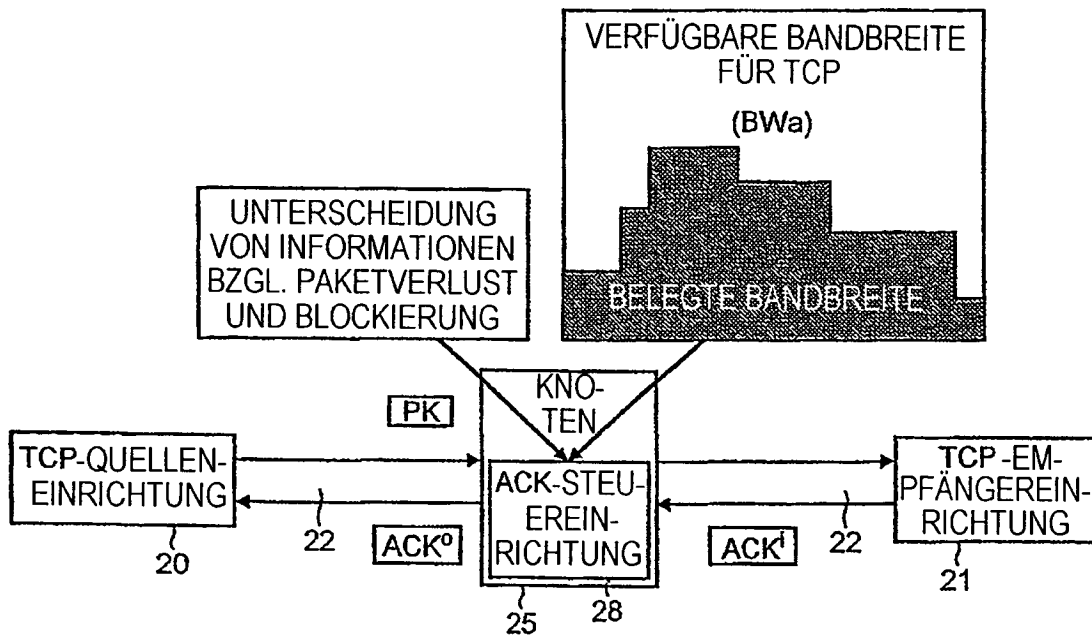
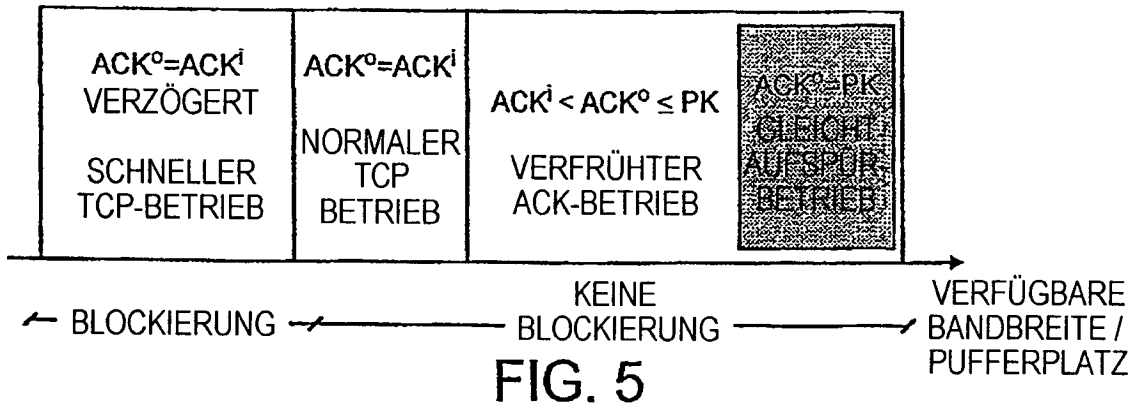


FIG. 4



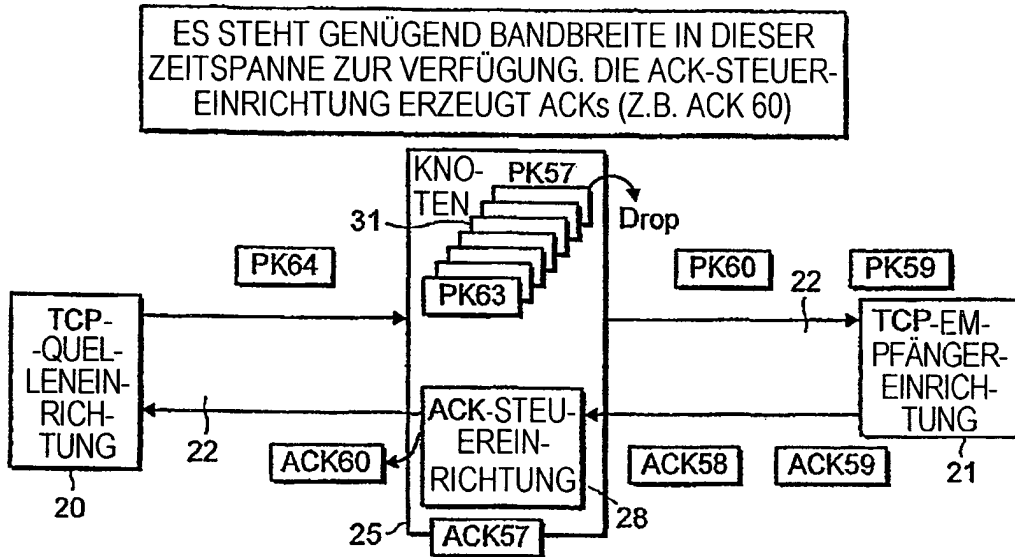


FIG. 7

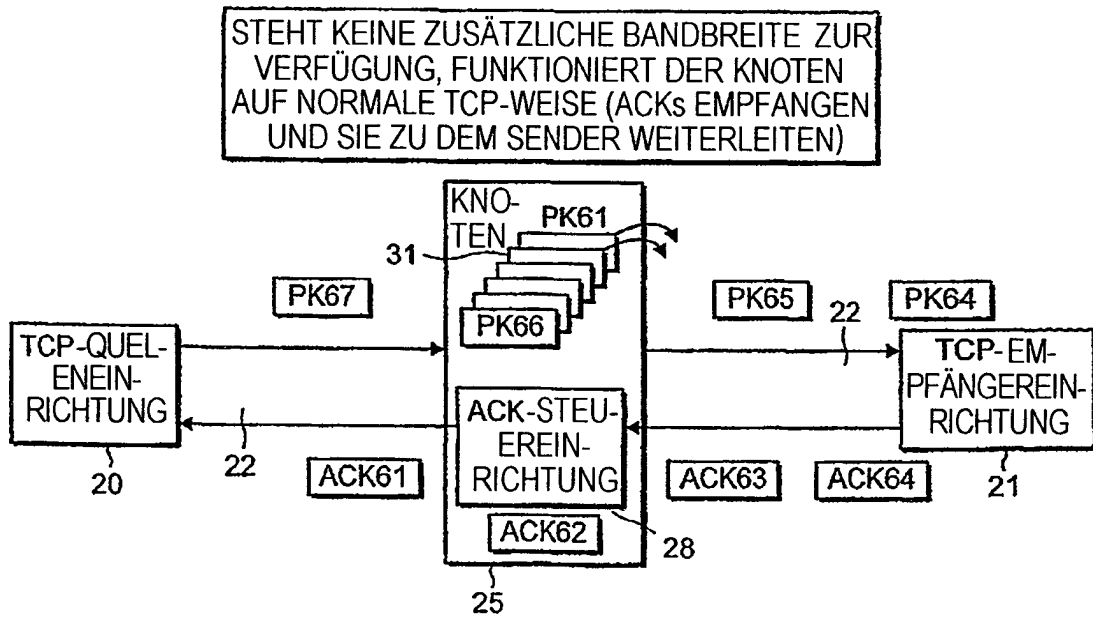


FIG. 8

ES STEHT NICHT GENÜGENDE BANDBREITE ZUR VERFÜGUNG,  
UND DER PUFFER WIRD VOLL. DIE ACK-STEUEREINRICHTUNG  
VERZÖGERT ACKs. (=SCHNELLES TCP-VERFAHREN)

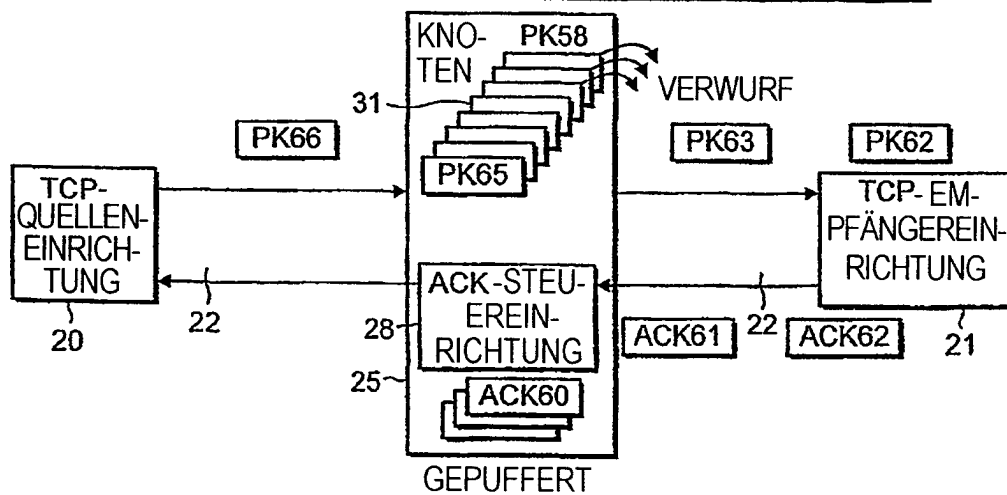


FIG. 9

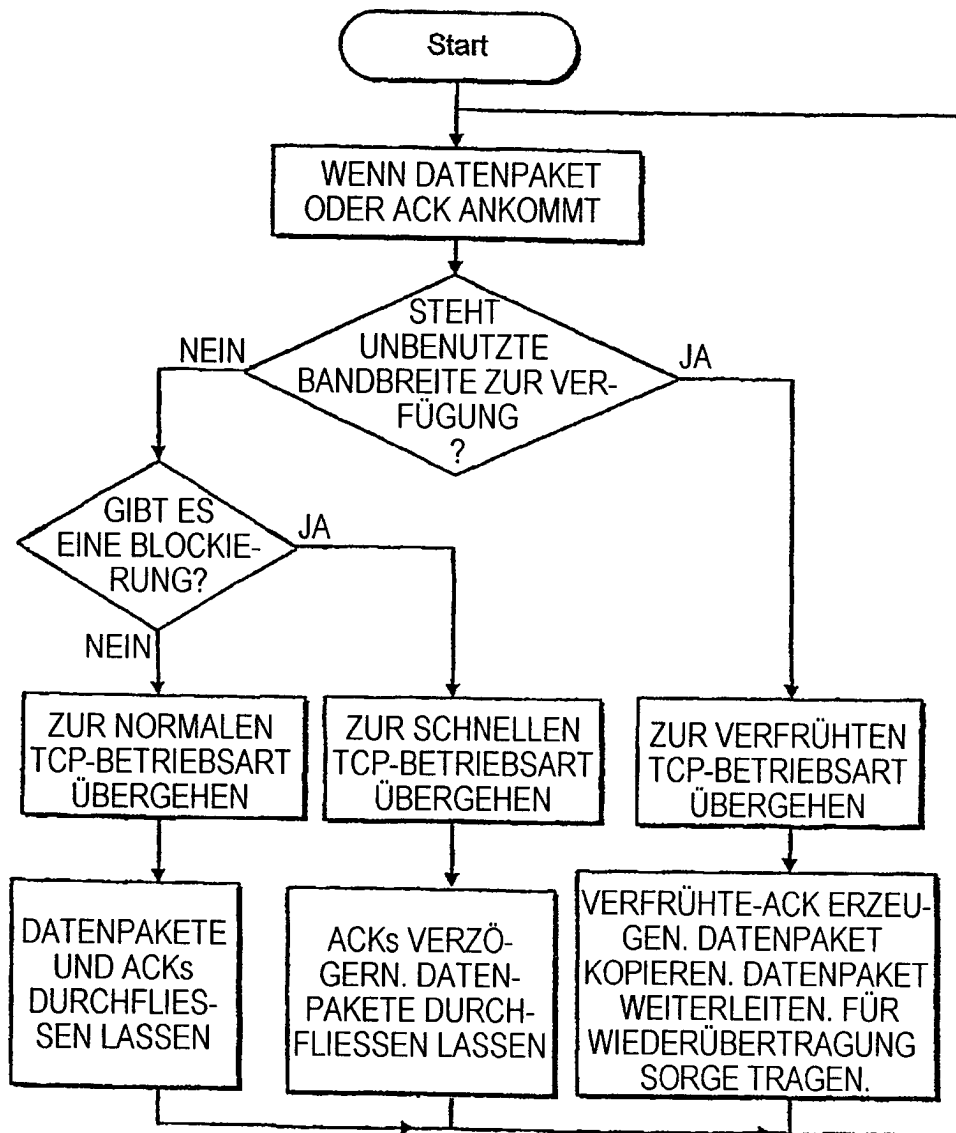


FIG. 10