(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 696 36 444 T2** 2007.03.15

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 751 656 B1**
(21) Deutsches Aktenzeichen: **696 36 444.1**
(96) Europäisches Aktenzeichen: **96 304 706.3**
(96) Europäischer Anmeldetag: **26.06.1996**
(97) Erstveröffentlichung durch das EPA: **02.01.1997**
(97) Veröffentlichungstag
der Patenterteilung beim EPA: **16.08.2006**
(47) Veröffentlichungstag im Patentblatt: **15.03.2007**

(51) Int Cl.[8]: ***H04L 29/06*** (2006.01)

(30) Unionspriorität:
495172 **27.06.1995** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT

(73) Patentinhaber:
Canon K.K., Tokio/Tokyo, JP

(72) Erfinder:
Man, Chiu Ming, Anaheim, California 92806, US

(74) Vertreter:
TBK-Patent, 80336 München

(54) Bezeichnung: **Protokollunabhängige, anpassungsfähige Netzwerkschnittstelle**

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] Die vorliegende Erfindung bezieht sich auf ein Verfahren zum Übertragen von Daten zwischen Anwendungsprogrammen, die auf verschiedenen Vorrichtungen laufen, die Schnittstellen mit einem lokalen Bereichsnetzwerk aufweisen, und insbesondere auf ein Verfahren zum Übertragen von Daten zwischen Anwendungsprogrammen unabhängig von jedwedem spezifischen Protokoll.

[0002] Computerisierte lokale Bereichsnetzwerke (LANs, „local area networks") werden weithin zum Untereinanderverbinden von vielen unterschiedlichen Computern und Peripheriegeräten verwendet, um Benutzern der Computer zu ermöglichen, miteinander zu kommunizieren, und ebenso, um jenen Benutzern einen gemeinsam verwendeten Zugriff auf die Peripheriegeräte zu ermöglichen. Jüngste Entwicklungen bei LANs zeigten die Einführung sogenannter „heterogener" LANs auf, d. h. LANs, auf denen viele unterschiedliche Kommunikationsprotokolle auf einem einzelnen Ethernet- oder Token-Ring-Medium getragen werden. Beispiele unterschiedlicher Protokolle sind IPX, das typischerweise von DOS-basierten PCs verwendet wird, UDP/IP, das typischerweise von UNIX-basierten Workstations verwendet wird, und DDP, das typischerweise von Macintosh-Computer verwendet wird. Jede Art von Computer oder Workstation kann durch Software unter Verwendung einer Mehrzahl von unterschiedlichen Protokollen zum Kommunizieren eingerichtet werden.

[0003] Ein Peripheriegerät kann ebenso Software enthalten, d. h. eine Mehrzahl von Protokollstapelmodulen, die dem Peripheriegerät ermöglicht, unter Verwendung einer Mehrzahl von Protokollen zu kommunizieren, um auf einem heterogenen LAN gemeinsam verwendet zu werden. Ein Protokollstapel ist ein Softwaremodul, das Datenpakete verarbeitet, die unter Verwendung des entsprechenden Protokolls von dem LAN empfangen oder zu diesem gesendet werden. Die Protokollstapel und die assoziierte Software niedrigerer Ebenen für Netzwerkkommunikationen werden typischerweise in einer Netzwerkschnittstellenvorrichtung, die in das Peripheriegerät eingebettet oder an dieses angebracht sein kann, gespeichert und ausgeführt. Die Netzwerkschnittstellenvorrichtung dient als eine Schnittstelle, die dem Peripheriegerät ermöglicht, mit anderen Netzwerkvorrichtungen über das LAN zu kommunizieren.

[0004] Netzwerkvorrichtungen, d. h. Netzwerkschnittstellenvorrichtungen und Computer, die eine Schnittstelle mit dem LAN aufweisen, führen ebenso Anwendungsprogramme aus. Diese Programme laufen auf einer Ebene überhalb der Protokollstapel ab und können beispielsweise Serverprogramme, Verwaltungsprogramme, die eine Kommunikation zwischen einem Computer und einem Peripheriegerät zum Konfigurieren oder Erhalten von Statusdaten von dem Peripheriegerät, und andere Programme enthalten, die Daten zwischen unterschiedlichen Vorrichtungen kommunizieren können, die eine Schnittstelle mit dem LAN aufweisen. Beispielbezogene Verwaltungsprogramme enthalten Programme, die ein SNMP (einfaches Netzwerkverwaltungsprotokoll, „Simple Network Management Protocol") und CMIP (allgemeines Verwaltungsinformationsprotokoll, „Common Management Information Protocol") implementieren. Es kann mehr als ein derartiges Verwaltungsprogramm auf einer einzelnen Netzwerkvorrichtung zur selben Zeit ablaufen.

[0005] Die Druckschrift US-A-5,301,303 offenbart eine Netzwerkkomponente in der Form eines Mehrkanalkonzentrators, der eine Anzahl von physikalischen Kommunikationskanälen zur Verbindung zu LANs bereitstellt, die unterschiedliche physikalische Medien, wie verdrillte Paare, koaxiale oder faseroptische, und/oder Kommunikationsprotokolle, wie Ethernet, Faserverteilungsdatenschnittstelle (FDDI, „fibre distriubtion data Interface"), Token-Bus und Token-Ring, aufweisen. Eine Anzahl von Medienverteilungsmodulen, die zum Implementieren eines einzelnen Protokolls wirksam sind und die Ports zur Verbindung mit einem spezifischen Medium aufweisen, stellen eine Verbindung zwischen dem einzelnen Medium und dem Verbinder bereit. Somit kann ein Netzwerk über ein Medienverteilungsmodul an den Konzentrator angebunden sein, um eine Verbindung mit einem anderen Netzwerk einzurichten. Wirkungs- und Wurzelmodule innerhalb des Konzentrators ermöglichen eine Kommunikation zwischen unterschiedlichen Kanälen und unterschiedlichen Netzwerken, die unterschiedliche Protokolle aufweisen, indem entweder ein Protokoll in ein anderes umgewandelt wird, oder indem Übertragungen gefiltert werden.

[0006] Die Druckschrift EP-A-0 288 713 offenbart eine Eingabe-/Ausgabesteuervorrichtung, die einen Hostcomputer an eine Vorrichtung in einem Informationsverarbeitungssystem ankoppelt. Jede Eingabe-/Ausgabevorrichtung weist eine assoziierte Steuertabelle zum Beziehen der Vorrichtung auf Programmanweisungen auf, die in der ein Kommunikationsprotokoll definierenden Steuereinrichtung gespeichert sind. Sind Daten zwischen der Eingabe-/Ausgabevorrichtung und dem Hostcomputer zu übertragen, dann werden Informationen in der Steuertabelle zum Aufrufen von Programmanweisungen verwendet, um das korrekte Kommunikationsprotokoll zum Leiten des Übermittelns der Daten zu der Eingabe-/Ausgabevorrichtung einzusetzen.

[0007] In einem herkömmlichen Ansatz kommuniziert ein Anwendungsprogramm über eine Anwendungsprogrammierschnittstelle (API, „application

programming Interface") mit einem Protokollstapel und verwendet den Protokollstapel zum Durchführen von Kommunikationsdiensten. Ein Anwendungsprogramm muss unterschiedliche APIs verwenden, um eine Schnittstelle mit jedem unterschiedlichen Protokollstapel zu bilden. Dies bedeutet, dass das Anwendungsprogramm sich der einzelnen Netzwerkumgebung, d. h. dem in Verwendung befindlichen Protokoll und der spezifischen, zu verwendenden Netzwerk-API, bewusst sein muss.

[0008] Der herkömmliche Ansatz führt zu vielen Schwierigkeiten. Muss ein Anwendungsprogramm eine Mehrzahl von Netzwerkprotokollen unterstützen, dann ist ein vermehrter Aufwand für die Anwendungssoftware erforderlich, um die unterschiedlichen APIs zu behandeln. Beispielsweise muss ein SNMP-Programm Softwarecode zum Kommunizieren mit einem IPX-Protokollstapel und einem DDP-Protokollstapel zusätzlich zu Code zum Kommunizieren mit einem UDP-Protokollstapel enthalten. Des Weiteren erfordert ein CMIP-Programm oder jedwede andere Anwendung ebenso den Sondercode zum Kommunizieren mit unterschiedlichen Protokollstapeln. Im Ergebnis erfordern Anwendungsprogramme, die zum Unterstützen einer Mehrzahl von Protokollen unter Verwendung des herkömmlichen Ansatzes entworfen sind, einen komplexeren Entwurf, weisen eine längere Entwicklungszeit auf, weisen eine geringere Ortsveränderbarkeit auf, und weisen höhere Wartungskosten als Anwendungsprogramme auf, die ein einzelnes Protokoll unterstützen.

[0009] Dementsprechend ist eine Art und Weise für Anwendungsprogramme erforderlich, um mit Anwendungsprogrammen auf andere Netzwerkvorrichtungen unabhängig von einem spezifischen Protokoll zu kommunizieren.

[0010] Dem vorstehend beschriebenen Erfordernis wendet sich die Erfindung zu, in der Daten zwischen Anwendungsprogrammen übertragen werden, die auf unterschiedlichen Vorrichtungen unabhängig von einem spezifischen Protokoll laufen. Gemäß der Erfindung ist ein Verfahren zum Zuführen eines Datenpakets bereitgestellt, das von einem ersten Anwendungsprogramm empfangen wird, das auf einer ersten Vorrichtung läuft, die eine Schnittstelle zu einem lokalen Bereichsnetzwerk aufweist, zu einem zweiten Anwendungsprogramm, das auf einer zweiten Vorrichtung läuft, die eine Schnittstelle zu dem lokalen Bereichsnetzwerk aufweist, wobei das Verfahren die Schritte umfasst:
Empfangen eines protokollunabhängigen Datenpakets von dem ersten Anwendungsprogramm zusammen mit Daten, die das erste Anwendungsprogramm identifizieren, und einer Ziel-ID, die das zweite Anwendungsprogramm identifiziert,
Bestimmen, welche Protokolle in dem lokalen Bereichsnetzwerk zur Verwendung zur Verfügung stehen, und
Zuweisen einer jeweiligen Zugriffsleitung zu jedem der Protokolle, die in dem lokalen Bereichsnetzwerk zur Verwendung zur Verfügung stehen,
wobei das Verfahren gekennzeichnet ist durch die Schritte:
Auswählen, aus den zur Verfügung stehenden Protokollen, des Protokolls, das der den geringsten Verkehr aufweisenden Zugriffsleitung zugewiesen ist,
Bestimmen von protokollspezifischen Informationen, die eine Art eines Protokoll-Headers und Adressinformationen in dem Header enthalten, die Daten entsprechen, die das erste Anwendungsprogramm und das in dem Auswahlschritt ausgewählte Protokoll identifizieren,
Bilden eines Übertragungspakets, das das Datenpaket, die Ziel-ID und die bestimmten protokollspezifischen Informationen enthält, und
Übertragen des Übertragungspakets über das lokale Bereichsnetzwerk zu dem zweiten Anwendungsprogramm.

[0011] Mittels dieser Anordnung kann ein Datenpaket, das von einem Anwendungsprogramm ohne jedwede ein Protokoll spezifizierende Daten empfangen wird, zu einem anderen Anwendungsprogramm lediglich auf der Grundlage von Informationen übertragen werden, die die Quell- und Zielprogramme identifizieren, die von dem einen Anwendungsprogramm empfangen sind.

Kurzbeschreibung der Zeichnungen

[0012] Es zeigen:

[0013] Fig. 1 eine Darstellung eines lokalen Bereichsnetzwerks,

[0014] Fig. 2 eine Darstellung der Softwarearchitektur, die für eine Kommunikation zwischen Anwendungsprogrammen gemäß einem beispielbezogenen Ausführungsbeispiel der Erfindung verwendet wird,

[0015] Fig. 3 eine Darstellung der Funktionsbeziehungen zwischen Softwaremodulen, die auf einem Computer und einer Netzwerkschnittstellenvorrichtung laufen,

[0016] Fig. 4 eine funktionelle Blockdarstellung einer Netzwerkerweiterungsplatine zum Bilden von Schnittstellen für einen Drucker zu einem lokalen Bereichsnetzwerk,

[0017] Fig. 5 Softwaremodule, die in einem Speicher auf der Netzwerkerweiterungsplatine gespeichert werden können,

[0018] Fig. 6 ein Ablaufdiagramm der Prozessschritte zum Übertragen eines Datenpakets zwischen einem ersten Anwendungsprogramm, das auf

einer ersten Netzwerkvorrichtung läuft, und einem zweiten Anwendungsprogramm, das auf einer zweiten Netzwerkvorrichtung läuft,

[0019] Fig. 7A und Fig. 7B Beispiele von Protokollabbildungstabellen, die durch protokollunabhängige Schnittstellen erstellt sind,

[0020] Fig. 8A und Fig. 8B Beispiele von Zugriffs-ID-Abbildungstabellen, die durch protokollunabhängige Schnittstellen erstellt sind,

[0021] Fig. 9A und Fig. 9B Beispiele von Protokolladressabbildungstabellen, die durch protokollunabhängige Schnittstellen erstellt sind,

[0022] Fig. 10A bis Fig. 10C Beispiele von Übertragungspaketformaten zur Verwendung mit unterschiedlichen Protokollen, und

[0023] Fig. 11 ein Ablaufdiagramm von Prozessschritten zum Empfangen eines Datenpakets, das von einem ersten Anwendungsprogramm, das auf einer Netzwerkvorrichtung läuft, zu einem anderen Anwendungsprogramm gesendet wird, das auf einer anderen Netzwerkvorrichtung läuft.

BESCHREIBUNG EINES BEISPIELBEZOGENEN AUSFÜHRUNGSBEISPIELS

[1. Systemübersicht]

[0024] Fig. 1 zeigt eine Darstellung eines heterogenen Netzwerksystems, das mehrere unterschiedliche Arten von Computern und mehrere unterschiedliche Peripheriegeräte enthält, auf die Computer gemeinsam zugreifen können. Die Erfindung kann ebenso bei Vorrichtungen verwendet werden, die mit einem homogenen Netzwerk verbunden sind, d. h. einem Netzwerk, in dem jede Vorrichtung das selbe Protokoll verwendet.

[0025] Gemäß Fig. 1 ist LAN **10** als ein Ethernet-Medium dargestellt, das eine busartige Architektur aufweist, es kann aber ebenso ein Token-Ring-Medium verwendet werden, das eine ringartige Architektur aufweist. Es sind mit dem LAN **10** ein PC **20**, der als ein Systemadministratorcomputer dient, ein PC **30**, der als ein Druckserver für Drucker **85** und **95** dient, ein Macintosh-Computer **40**, eine UNIX-Workstation **50** und eine allgemeine Workstation **60**, die eine Steuereinheit **61** und eine Anzeige **62** aufweist, verbunden. Ein Dateien-Server **70** ermöglicht gemeinsam verwendeten Zugriff auf eine Netzwerkplatte **75**. Eine Netzwerkerweiterungsplatine (NEB, „Network Expansion Board") **100** ermöglicht einen gemeinsam verwendeten Zugriff auf einen Drucker **105**, und eine Netzwerkerweiterungseinrichtung (NED, „Network Expansion Device") **110** ermöglicht gemeinsam verwendeten Zugriff auf einen Drucker **115**. Außerdem ermöglicht eine Netzwerkschnittstellenplatine (NIB, „Network Interface Board") **120** einen gemeinsam verwendeten Zugriff über eine Mehrfachvorrichtungssteuereinrichtung (MDC, „Multiple Device Controller") **130** auf einem Kopierer **135**.

[0026] Die Erfindung bezieht sich auf eine Kommunikation zwischen Anwendungsprogrammen, die auf unterschiedlichen Netzwerkvorrichtungen laufen. Eine bevorzugte Form der Erfindung ist nachstehend im Zusammenhang einer Kommunikation zwischen PC **20** und NEB **100** beschrieben. Die Erfindung ist jedoch allgemein bei Computern und eingebetteten Netzwerkvorrichtungen anwendbar. Dementsprechend kann die Erfindung bei einer Kommunikation zwischen anderen Computern, wie einem Macintosh-Computer **40**, einer UNIX-Workstation **50**, einer allgemeinen Workstation **60** und anderen Netzwerkschnittstellengeräten, wie eine NED **110** (ein Beispiel derer ist in der ebenfalls anhängigen US-Patentanmeldung S.N. 08/489,116 beschrieben, die am 9. Juni 1995 eingereicht wurde und den Titel „Outputting a Network Device Log File" trägt) und eine NIB **120** (ein Beispiel derer ist in der US-Patentanmeldung S.N. 08/409,034 (die der europäischen Anmeldung Nr. 96 301 994.8 entspricht) beschrieben, die am 23. März 1995 eingereicht wurde und den Titel „Network Interface Board for Digital Copierer" trägt, die dem Rechtsnachfolger der Erfindung überschrieben ist), angewendet werden.

[0027] Die Erfindung kann ebenso bei einer Kommunikation zwischen Anwendungsprogrammen angewendet werden, die auf unterschiedlichen Computern laufen, die die Fähigkeit zum Verwenden einer Mehrzahl von Protokollen aufweisen. Des Weiteren ist die Erfindung nicht auf Netzwerkanwendungen beschränkt, sondern kann statt dessen für Vorrichtungen verwendet werden, die eine direkte Verbindung durch jedwede bidirektionale Schnittstelle, z.B. einen gemeinsam verwendeten Speicher, eine SCSI-Schnittstelle, eine RS-1284-Parallelschnittstelle oder dergleichen aufweisen.

[2. Softwarearchitektur]

[0028] Fig. 2 zeigt die Softwarearchitektur von Programmmodulen, die auf einer Netzwerkvorrichtung, wie der NEB **100**, laufen. Ähnliche Software läuft auf einem Computer, wie dem PC **20**. Eine Netzwerkschnittstellenansteuereinrichtung **210** ist die tiefste Softwareebene, die eine Schnittstelle mit dem LAN **10** bildet und das Senden und Empfangen von Paketen im LAN **10** durch Hinzufügen oder Abtrennen von Paketrahmenheadern verwaltet. Die Netzwerkschnittstellenansteuereinrichtung **210** enthält eine medienartspezifische Komponente, die entweder für ein Ethernet- oder ein Token-Ring-Netzwerkmedium entworfen ist und die eine Vielzahl von logischen Platinen zum jeweiligen Verarbeiten von

Paketen, die unterschiedliche Rahmenarten aufweisen, enthält. Ein Multiplexersoftwaremodul **220** dient als ein Multiplexer, der Pakete zwischen der Netzwerkschnittstellenansteuereinrichtung **210** und einem oder mehreren Protokollstapeln verkehrslenkt. Jeder Protokollstapel empfängt Pakete, die das entsprechende Protokoll verwenden, bestimmt, was mit den Paketen zu tun ist, und verkehrslenkt die Pakete zu den geeigneten Anwendungsprogrammen zur Weiterverarbeitung. Das bevorzugte Ausfürungsbeispiel unterstützt drei Protokollstapel: IPX-Stapel **230**, UDP-Stapel **231** und DDP-Stapel **232**. Es können womöglich nicht alle Protokollstapel in eine einzelne Vorrichtung geladen werden. Es können ferner zusätzliche Stapel für andere Protokolle enthalten sein. In der bevorzugten Ausführungsbeispiel gehen die Netzwerkschnittstellenansteuereinrichtung **210**, das Multiplexersoftwaremodul **220** und Protokollstapel **230** bis **232** konform mit der offenen Datenanbindungsschnittstellen- (ODI, „Open Data-Link Interface") -Spezifikation, die in „Open-Data-Link Interface Developer's Guide for DOS Workstation Protocol Stacks", Version 1.10, herausgegeben durch Novell Inc. am 18. März 1992 beschrieben ist.

[0029] Eine protokollunabhängige Schnittstelle (PII, „Protocol Independent Interface") **250** dient als eine Schnittstelle zwischen Protokollstapeln **230** bis **232** und Verwaltungsanwendungsprogrammen, wie einem SNMP-Anwendungsprogramm **260** und einem CMIP-Anwendungsprogramm **270**. Die PII **250** „horcht" nach Datenpaketen, die an einzelne Sockets, d. h. Adressen, adressiert sind und akzeptiert jene Pakete von den Protokollstapeln **230** bis **232** zur Verarbeitung und Weiterleitung zu den Anwendungsprogrammen. Da das SNMP-Programm **260** und das CMIP-Programm **270** auf einer eingebetteten Vorrichtung ablaufen, d. h. der NEB **100**, sind jene Programme „Agenten"-Programme. Ein Agentenprogramm sammelt und speichert Daten hinsichtlich der Netzwerkschnittstellenvorrichtung, d. h. der NEB **100**, und des Peripheriegeräts, d. h. des Druckers **105**, und antwortet auf gesendete Befehle unter Verwendung des assoziierten Netzwerkverwaltungsprotokolls, z. B. SNMP oder CMIP, von einem verwandten „Verwaltungs"-Programm, das auf einem Computer läuft.

[0030] Es werden beispielsweise die nachstehenden vorbestimmten Adressen in dem bevorzugten Ausführungsbeispiel zum Empfangen von Datenpakten unter Verwendung des SNMP-Netzwerkverwaltungsprotokolls verwendet: (1) für IPX „Socket" 900 F_H und 9010_H (Agenten-Socket bzw. Unterbrechungs-Socket), (2) für UDP „Port" 160_H und 161_H und (3) für DDP ein eindeutiger Name „SNMP-Agent" und „SNMP-Unterbrechungsverwalter" und „Socket" 8_H und 9_H (Agenten-Socket bzw. Unterbrechungs-Socket). Datenpakete, die nicht an ein durch ein Verwaltungsprogramm verwendetes Socket adressiert sind, werden zu anderen Anwendungsprogrammen verkehrsgelenkt. Beispielsweise empfängt ein PSERVER-Programm **245** Datenpakete über eine API **240**. Es können ebenso andere Anwendungsprogramme einschließlich weiterer Verwaltungsprogramme enthalten sein.

[0031] Fig. 3 zeigt eine Darstellung der Funktionsbeziehung zwischen Verwaltungsprogrammen, die auf einem PC **20** laufen, und Agentenprogrammen, die auf einer NEB **100** laufen. Wie gemäß Fig. 3 gezeigt, enthält PC **20** eine protokollunabhängige Schnittstelle (PII) **255**, einen SNMP-Verwalter **265** und einem CMIP-Verwalter **275**. Während eines nachstehend beschriebenen Initialisierungsprozesses weist die PII **255** jedem Verwaltungsprogramm eine eindeutige Kennung zu, die Zugriffs-ID genannt wird. Die Zugriffs-ID kann beispielsweise zusammen mit einer zusätzlichen Zahl die MAC-Adresse der Vorrichtung sein, um jeden SNMP-Verwalter **265** und jeden CMIP-Verwalter **265** eindeutig zu identifizieren. Bezugszeichen **281**, **282** und **283** bezeichnen logische Kanäle, die durch unterschiedliche jeweilige Protokolle verwendet werden, wie IPX, UDP und/oder DDP. Die PII **255** weist jedem der Protokolle während einer Initialisierung eine Kennung zu, die „logische Zugriffsleitung" genannt wird, d. h. logische Kanäle **281** bis **283**. Durch dynamisches Zuweisen von Zugriffs-IDs und logische Zugriffsleitungen kann die PII leicht zum Unterstützen von später entwickelten Protokollen ohne ein Beeinträchtigen der existierenden Funktionalität angepasst werden.

[0032] Die PII **250** in NEB **100** weist ebenso jedem SNMP-Agenten **260** und jedem CMIP-Agenten **270** eine Zugriffs-ID zu und weist jedem der Protokolle eine logische Zugriffsleitung zu. Da jedoch die Zugriffs-IDs und logischen Zugriffsleitungen für die PIIs spezifisch sind, die jene zuweist, können die durch die PII **250** in NEB **100** zugewiesenen Zugriffs-IDs und logischen Zugriffsleitungen von jenen durch die PII **250** in PC **20** zugewiesenen abweichen. Wie beispielsweise gemäß Fig. 3 gezeigt, weist SNMP-Verwalter **265** Zugriffs-ID #2 in PC **20** auf, aber der entsprechende Agent in NEB **100**, d. h. SNMP-Agent **260**, weist Zugriffs-ID #1 auf. Die PII **250** weist ähnlich wie gemäß in Fig. 3 gezeigt einem logischen Kanal **281** eine logische Zugriffsleitung #1 zu, was beispielsweise einem IPX-Protokoll entsprechen kann, während die PII **255** dem selben logischen Kanal/Protokoll eine logische Zugriffsleitung #3 zuweist.

[0033] Die PII **250** und die PII **255** führen identische Schnittstellenfunktionen durch. Es gibt jedoch aufgrund der unterschiedlichen Plattformen, auf denen diese Softwaremodule laufen, einige Unterschiede in der Implementierung. Da beispielsweise die PII **250** auf einer eingebetteten Vorrichtung läuft, wird sie als eine CSR- (Terminieren und Resident bleiben, „terminate and stay resident") – Routine implementiert.

Demgegenüber ist der SNMP-Verwalter **265** eine Windows-basierte Anwendung, und PII **255** ist als eine Windows-basierte DLL (Dynamische Anbindungssammlung, „Dynamic Link Library") implementiert. Andere Unterschiede sind, wo geeignet, in der nachstehenden Beschreibung gezeigt.

[3. NEB-Architektur]

[0034] Fig. 4 zeigt eine funktionelle Blockdarstellung von NEB **100**. Grob gesprochen ist NEB **100** eine interaktive Netzwerkplatine, die Drucker **105** an LAN **10** ankoppelt, was Drucker **105** zu einem antwortenden und interaktiven Netzwerkmitglied macht. Die NEB **100** enthält einen gemeinsam verwendeten Speicher SRAM **200**, der für eine bidirektionale Kommunikation zwischen der NEB **100** und dem Drucker **105** verwendet wird. Drucker **105** enthält eine Druckerschnittstellenkarte **220** (nicht gezeigt), die einen Mikroprozessor **225** (nicht gezeigt) aufweist, der Daten aus dem SRAM **200** liest und Daten in SRAM **200** schreibt. Der Drucker **105** enthält ebenso einen Druckantrieb **250** (nicht gezeigt), der mit der Druckerschnittstellenkarte **220** verbunden ist.

[0035] Die NEB **100** empfängt Druckdaten, Statusanforderungen und Steuerbefehle vom LAN **10**, überträgt Druckdaten, Statusanforderungen und Steuerbefehle zu Drucker **105** zur Ausführung und überträgt Statusinformationen zurück zu LAN **10**. Somit kann die NEB **100** nicht lediglich Ferndruckdienste RPRINTER und Druckserverfunktionalitäten PSERVER durchführen, sondern kann ebenso Netzwerkmitgliedern jedes von der Peripherieschnittstelle zur Verfügung stehende Status- und Steuermerkmal bereitstellen.

[0036] Es wird der NEB **100** von einer +5V-Energiequelle **398** Energie für alle Schaltungen zugeführt. Es wird Energie von der Energiequelle **398** bereitgestellt, um Umwandler **396** zu energetisieren, der einem Sendeempfänger **390** -9V-Energie bereitstellt, und um Umwandler **397** zu energetisieren, der einem Flash-EPROM **350** zum „Flashing" (d. h. zum Reprogrammieren des EPROMs) +12V-Energie bereitstellt. Eine Netzwerk- und Netzwerkschnittstellensteuerlogik **340** ist vorzugsweise eine einzelne anwendungsspezifische integrierte Schaltung (ASIC, „Application specific Integrated circuit") mit 144 Pins, die eine Netzwerksteuereinrichtung **330** und eine Schnittstellensteuerlogik **320** enthält. Die Netzwerksteuereinrichtung **330** ist eine NCR-Makrozelle, die mit einer Ethernet-Steuereinrichtung National DP83902A „ST-NIC" kompatibel ist, deren Einzelheiten in National Semiconductor's Local Area Networks Databook, National Semiconductor p/n 400055, National Semiconductor, 1993 nachzuschlagen sind. Die Netzwerksteuereinrichtung **330** ist zum Bilden von Schnittstellen mit lokalen Bereichsnetzwerken vom Typ CSMA/CA (Trägeraufspürmehrfachzugriff mit Kollisionserfassung, „carrier sense multiple access with collision detection") entworfen.

[0037] Die Netzwerksteuereinrichtung **330** verbindet sich direkt mit einem RJ-45-Verbinder **385** und einem koaxialen Verbinder **395** durch Sendempfänger **390**, der vorzugsweise eine koaxiale Sendeempfängerschnittstelle National Semiconductor DP8392 ist, deren Einzelheiten ebenso in National's Local Area Networks Databook nachzuschlagen sind. Die Netzwerksteuereinrichtung **330** ist ebenso an einen 8KB-SRAM **380** gekoppelt, der als ein Eingabe-/Ausgabepaketpuffer für Ethernet-Daten verwendet wird. Dieser Speicher sollte vorzugsweise eine Zugriffszeit von etwa 70 ns oder weniger aufweisen.

[0038] Die Schnittstellensteuerlogik **320** stellt eine Schnittstelle zwischen der Netzwerksteuereinrichtung **330**, einem Mikroprozessor **300** und Speichergeräten EPROM **350** und DRAM **360** bereit. Die Schnittstellensteuerlogik **320** stellt ebenso eine Schnittstelle mit einem nicht-flüchtigen Speicher **370** mit Wahlfreiem Zugriff (NVRRM, „non-volatile random access memory") bereit, der ein serieller, elektrisch löschbarer/programmierbarer Speicher mit 256 Byte ist, der zu Initialisierungsdatenspeicherung während einem periodischen Energiedurchlaufen des Druckers **105** verwendet wird. Netzwerk- und Druckerkonfigurationsparameter werden in den NVRAM **370** geschrieben, wenn der Drucker **105** erstmals in dem Netzwerk installiert wird, um einer NEB-Software ein Ermitteln der Installationsparameter zu ermöglichen, nachdem die Druckerenergie einen Aus- und Einzyklus durchlaufen hat.

[0039] Die Schnittstellensteuerlogik **320** ist ebenso an einen seriellen Portverbinder **325** gekoppelt, der einen Empfangsdatenpin **326** und einen Sendedatenpin **327** umfasst, die jeweils serielle Datenströme für Fehlerbehebungszwecke empfangen und senden können. Die Schnittstellensteuerlogik **320** spürt Daten auf, die auf der Empfangsdatenleitung vorhanden sind, und tastet die seriellen Bits in regelmäßigen Intervallen ab.

[0040] Die zentrale Steuereinrichtung von NEB **100** ist Mikroprozessor **300**, der vorzugsweise ein Intel 80C188EA-20 8-Bit-Prozessor ist, dessen Einzelheiten in dem 80C186EA/80188EA Benutzerhandbuch, Intel p/n 270950-001, Intel Corp., nachzuschlagen sind. Dieser Prozessor ist ein 8-Bit-Prozessor mit einem Direktspeicherzugriff (DMA, „direct memory access"), Unterbrechungen, Zeitgebern und einer DRAM-Auffrischsteuerung. Andere Mikroprozessoren, wie ein AMD 80C188-20 8-Bit-Mikroprozessor können womöglich alternativ verwendet werden. Es sind ein 256-KB-Flash-EPROM **350** und ein 512-KB-DRAM **360** an den Mikroprozessor **300** über die Schnittstellensteuerlogik **320** gekoppelt, während ein 32-KB-SRAM **200** (der mit einer Druckerschnitt-

stellenkarte **220** gemeinsam verwendet wird) über eine Entscheidersteuerlogik **400** mit dem Mikroprozessor **300** gekoppelt. Ein Kristalloszillator **310** mit 40 MHz und 50 PPM versorgt Mikroprozessor **300** mit einem Taktsignal, das vollkommen separat von und asynchron mit dem Taktsignal ist, das dem Mikroprozessor **225** auf Druckerschnittstellenkarte **220** bereitgestellt ist.

[0041] Der Mikroprozessor **300** führt Anweisungen in dem Flash-EPROM **350** aus, der Steuer-Firmware und Druckanwendungssoftware speichert. Nach einer Selbstprüfung bei Energie-Ein (POST, „power-on self-test") wird Code vom EPROM **350** selektiv zu dem 512-KB-DRAM **360** höherer Leistungsfähigkeit zur tatsächlichen Ausführung bewegt, der vorzugsweise eine Zugriffszeit von etwa 80 ns aufweisen sollte.

[0042] Die gesamte Kommunikation zwischen NEB **100** und Druckerschnittstellenkarte **220** wird über den gemeinsam verwendeten SRAM **200** mit 32 KB ausgeführt. Die Entscheidersteuerlogik **400**, vorzugsweise eine einzelne ASIC mit 100 Pins, entscheidet zwischen den zwei-byte-breiten Speicherzugriffen des Druckerschnittstellenmikroprozessor **225** und den einzel-byte-breiten Speicherzugriffen von NEB-Mikroprozessor **300**, von denen jeder von dem anderen vollkommen unabhängig ist.

[0043] Allgemein gesprochen, kommuniziert der 8-Bit-Datenbus des Mikroprozessors **300** auf Platine NEB **100** mit einer Bussteuerlogik **410**, während der 32-bit-Datenbus von Mikroprozessor **225** auf Platinendruckerschnittstellenkarte **220** mit einer Bussteuerlogik **420** kommuniziert. Speicherzugriffe von jedem Bus werden zu dem gemeinsam verwendeten Speicherentscheider **430** verkehrsgelenkt, der bestimmt, welcher Bus Priorität aufweist, und erlaubt dem Bus mit Priorität, über die SRAM-Schnittstelle **440** auf den SRAM **200** zuzugreifen. Auf ein Unterbrechungssteuerregister **450** wird ebenso durch den gemeinsam verwendeten Speicherentscheider **430** zu gegriffen, um einem Mikroprozessor zu ermöglichen, den anderen zu unterbrechen.

[0044] Alle durch Mikroprozessor **300** ausgeführten Softwaremodule sind in dem Flash-EPROM **350** gespeichert. Jene Module, die erforderlich sind, werden selektiv von dem EPROM **350** in den DRAM **360** geladen, und aus dem DRAM ausgeführt. Dies ermöglicht eine flexible Konfiguration der NEB **100** durch Auswahl, welche Module zu laden sind.

[0045] Fig. 5 zeigt ein Beispiel von Codeblöcken oder Softwaremodulen, die in den Flash-EPROM **350** gespeichert sind. Das PII-Modul enthält Prozessschritte zum Bereitstellen der erforderlichen Funktionen einer protokollunabhängigen Schnittstelle, wie nachstehend ausführlich beschrieben. Das XPL-Modul stellt eine standardisierte Schnittstelle zwischen Drucker **105** und NEB **100** bereit. Eine MLID (Multianbindungsschnittstellenansteuereinrichtung „Multi Link Interface Driver") dient als die Netzwerkschnittstellenansteuereinrichtung **210** und ist ein Teil des Novell-Codes (Medienunterstützungsmodul, oder MSM, „Media Support Module"), der mit einem Teil von maßgeschneidertem Code (Hardwareunterstützungsmodul, oder HSM, „Hardware Support Modul") verlinkt ist, der die niedrigste Ebene einer Netzwerkverbindung ist, während eine LSL-Anbindungsunterstützungsschicht, „Zink Support Layer") als das Multiplexersoftwaremodul **220** dient und ein Teil des Novell-Codes ist, der als ein Multiplexer zwischen der Niederebenen-MLID und den mehreren, darüber liegenden Protokollstapeln agiert. CNETX ist maßgeschneiderter Code, der lokale DOS-artige Funktionsaufrufe in Netzwerkfunktionsaufrufe umwandelt, was Dateifunktionen wie ÖFFNEN, LESEN, SCHREIBEN und SCHLIEßEN bereitstellt.

[0046] Das PRETASK-Modul ist verantwortlich, zu bestimmen, welche Rahmenarten mit den verschiedenen möglichen Protokollstapeln assoziiert sind. Da die NEB **100** eine Mehrzahl von Protokollstapeln unterstützt, existiert dieses Modul solange wie die NEB **100** läuft.

[0047] Der IPX/SPX-Protokollstapel von Novell ist in dem Flash-EPROM **350** enthalten und wird durch SAP (Dienstwerbungsprotokoll, „Service Advertising Protocol") unterstützt. SAP ist eine Novell-Konzept, das Vorrichtungen ermöglicht, sich selbst in dem Bindewerk des Dateien-Servers zu registrieren, das aktive und inaktive Netzwerkfunktionseinheiten auflistet. Da Druckserver eine besondere Art von Bindewerksgegenstand sind, registriert das SAP die NEB **100** über CPSOCKET, und falls die NEB **100** als ein Druckserver konfiguriert ist, dann registriert SAP den Druckserver ebenso bei dem NetWare-Bindewerk.

[0048] CPSERVER ist eine maßgeschneiderte Implementierung einer Novell-Druckserveranwendung. Dieses Modul stellt selbsterzeugte Druckbanner, Benutzerbenachrichtigung bezüglich Vollendung und Ausnahmestatus, und Übertragung von Druckdaten und Statusbefehlen zu dem Drucker bereit. Dies weicht von dem Novell-Druckserver dahingehend ab, dass der CPSERVER einem Ansteuern des lokalen Druckers (d. h. Drucker **105**, in dem die NEB **100** installiert ist) gewidmet ist, und keine Fern-RPRINTER ansteuern kann. Dieses Programm nimmt die Druckdatenleitungen für die Dauer eines Druckauftrags in Besitz. Ein CRPRINTER ist eine maßgeschneiderte Implementierung einer Novell-RPRINTER-Druckanwendung. Dieses Modul ist eine Slave-Anwendung, der Daten durch eine Novell-Druckserveranwendung zugesendet werden, die sich an einer anderen Stelle im LAN **10** befindet.

[0049] Der TCP/IP-Protokollstapel weist in sich das Benutzerdatagrammprotokoll (UDP, „User Datagramm Protokoll"), das Umgekehrte-Adressenauflösungsprotokoll (RARP, „Reverse Address Resolution Protocol") und die BootP-Untersützung auf. INTERRUPT ist die Unterbrechungsverwaltung für die TCP/IP-Task, während TIMERTICK der Zeitgeber für UNIX-TCP/IP-Netzwerktasks ist. LPRINTSERVER ist die TCP/IP-Druckserveranwendung, und nimmt ebenso die Druckdatenleitungen für die Dauer eines Druckauftrags in Besitz. DDP ist das Modul zum Implementieren eines Datagrammvermittlungsprotokolls (DDP, „Datagramm Delivery Protocol"), das beispielsweise zur Kommunikation mit einem Macintosh-Computer verwendet wird.

[0050] Das CPSOCKET-Programm läuft für alle Protokollstapel. Das Programm antwortet auf Anforderungen nach Verbindung, Anforderungen nach Datendownload oder Anforderungen nach Diensten von entfernten Einheiten, und stellt über eine Zwischenprozesskommunikation anderen Tasks Status und Steuerung bereit. Da der CPSOCKET typischerweise die Status- und Steuerleitungen zwischen NEB **100** und Drucker **105** in Besitz nimmt, ist er die einzige Task, die die Fähigkeit aufweist, einen Druckerstatus über die Statusleitungen zu erhalten. Der CPSOCKET ist für die Netzwerkverbindung und Paketinhalte zwischen dem Novell-orientierten Status- und Steuereinheiten (CPNET oder die entsprechende Windowsversion von client-basierten Softwareeinheiten), oder zwischen den UNIX-orientierten Status- und Steuereinheiten (CPUTIL) verantwortlich.

[0051] MONITOR ist ein maßgeschneiderter Multitasking-Überwacher, der eine Taskerstellung, ein Taskzerstörung und ein Mikroprozessorabfertigen durchführt. MONITOR weist ebenso Speicherverwaltungssubmodule MEMGET und MEMFREE auf.

[0052] RESIDENT ist ein Block von Routinen, der generische Dienste bereitstellt, wie Lesen aus und Schreiben in den Flash-EPROM **350**, FLASH-Code, ROM-basierte Fehlerbehebung, Hardwarezeitgabe und andere grundlegende Merkmale. POST ist ein Selbstprüfmodul bei Energie-ein, das die Integrität von NEB-Hardware und -Software bei einem Energie-ein prüft.

[0053] Ebenso sind in dem EPROM **350** ein Netzwerkidentifikationsdatei- (NIF „ Network Identification file") -Datenblock, der Platinen variante Informationen speichert, die für jede Netzwerkplatine eindeutig sind, Hardwarekonfigurationsdaten, Platinenrevisionsnummer und dergleichen, sowie änderbare Informationen, wie eine Softwareversionsnummer, gespeichert. Die Informationen in dem NIF-Datenblock werden verwendet, um sicher zu stellen, dass der Flash-EPROM nicht mit einem inkompatiblen Firmware-Abbild reprogrammiert wird.

[0054] Der EPROM **350** speichert insbesondere „Platinen"-Informationen, wie Modellnummer, Firmware-Ebene und Platinenrevionsnummer, sowie „Netzwerk"-Informationen, wie eine Medienzugriffssteuer- (MAC, „Media Access Control") -Adresse, die für jede Netzwerkplatine eindeutig ist, einen Platinennamen, eine Netzwerkrahmenart, eine Primär-Dateienserver-Identifikation, bediente Warteschlangen, Netzwerkprotokoll, Abtastfrequenz, PSERVER-Name, Zonenname und dergleichen.

[4. Funktion der PII]

[0055] Fig. 6 zeigt ein Ablaufdiagramm von Prozessschritten zum Implementieren eines protokollunabhängigen Verfahrens zum Übertragen eines Datenpakets von einem ersten Anwendungsprogramm, das auf einer ersten Vorrichtung läuft, die eine Schnittstelle mit einem LAN aufweist, zu einem zweiten Anwendungsprogramm, das auf einer zweiten Vorrichtung läuft, die eine Schnittstelle zu dem LAN aufweist. Kurz gesprochen, wird gemäß Fig. 6 ein protokollunabhängiges Schnittstellen- (PII) -Programm initialisiert, das bestimmt, welche Protokolle zur Verwendung zur Verfügung stehen, das jedem Protokoll, das zur Verwendung zur Verfügung steht, eine Zugriffsleitung zuweist, das dem ersten Anwendungsprogramm eine Zugriffs-ID zuweist und das Abbildungsinformationen erstellt, die eine Eins-zu-Eins-Entsprechung zwischen einem Zugriffs-ID/Zugriffsleitungs-Paar und einem Block von protokollspezifischen Informationen, die einen vorbestimmte Adressdaten aufweisenden Protokollheader enthalten, angeben. Diese Eins-zu-Eins-Abbildung kann unter Verwendung einer Abbildungstabelle, wie in der bevorzugten, nachstehend beschriebenen Form, oder durch Implementieren einer Datenstruktur, die die Abbildungsinformationen führt, durchgeführt werden. Ein Datenpaket wird zu dem PII-Programm zusammen mit der Zugriffs-ID des ersten Anwendungsprogramms und einer Ziel-ID für das zweite Anwendungsprogramm gesendet, und es wird eines der zur Verfügung stehenden Protokolle zum Senden des Datenpakets ausgewählt. Und es wird eines der zur Verfügung stehenden Protokolle zum Senden des Datenpakets ausgewählt. Ein Block von protokollspezifischen Informationen wird aus der ersten Abbildungstabelle auf der Grundlage der Zugriffs-ID des ersten Anwendungsprogramms und der Zugriffsleitung, die dem ausgewählten Protokoll entspricht, geholt, und es wird ein Übertragungspaket gebildet, das das Datenpaket, die Ziel-ID und den geholten Block von protokollspezifischen Informationen enthält. Das Übertragungspaket wird dann über das LAN übertragen.

[0056] Es zeigen genauer gesagt die Prozessschritte gemäß Fig. 6 eine Übertragung eines Datenpakets von dem SNMP-Verwalter **265** auf PC **20** zu dem SNMP-Agenten **260** in der NEB **100**. Die Funktion

der PII **250** zum Übertragen eines Datenpakets von dem SNMP-Agenten **260** zu dem SNMP-Verwalter **265** ist dieselbe, wobei die einzigen Unterschiede die nachstehend beschriebenen Unterschiede in der Implementierung sind. In Schritt S601 führt der SNMP-Verwalter **265** einen Initialisierungsbefehl zum Initialisieren der PII **255** aus. Dieser Befehl muss vor einer Ausführung jedweder anderer PII-Befehle ausgeführt werden, und muss lediglich einmal durchgeführt werden. Wie vorstehend beschrieben, ist die PII **255** in Windows als eine DLL implementiert. Dementsprechend werden, als Antwort auf den Initialisierungsbefehl, die erforderlichen Vorgänge durchgeführt, um dem SNMP-Verwalter **265** zu ermöglichen, andere PII-Befehle unter Verwendung der DLL durchzuführen. Im Gegensatz dazu gibt eine Initialisierung der PII **250** durch den SNMP-Agenten **260** in der NEB **100** eine Tabelle von Eintrittspunkten zurück, die der SNMP-Agent **260** zum Aufrufen anderer PII-Routinen verwendet.

[0057] Die PII **255** bestimmt dann, welche Protokolle für eine Verwendung durch PC **20** zur Verfügung stehen. In dem bevorzugten Ausführungsbeispiel, in dem der PC **20** in einer Windows- oder DOS-Novell-ODI-Umgebung operiert, werden Funktionsaufrufe, die eine Angabe des Vorhandenseins oder Fehlens eines jeden Protokollstapels erhalten, auf eine gemeinsame Antragsweise erteilt, um zu bestimmen, welche Protokollstapel zur Verfügung stehen. Es können alternativ Befehle zum Ausführen jeweiliger Initialisierungsroutinen für jeden Protokollstapel auf eine gemeinsame Antragsweise erteilt werden. Schlägt eine Initialisierungsroutine fehl, dann wird der Fehler dahingehend interpretiert, anzugeben, dass der Protokollstapel in dieser Instanz nicht zur Verfügung steht. In der eingebetteten Plattform, d. h. für die PII **250** in der NEB **100**, weist das PRETASK-Modul Informationen hinsichtlich der zur Verfügung stehenden Protokollstapel auf, und jene Informationen werden durch die PII **250** erhalten und verwendet.

[0058] Nach einem Bestimmen, welche Protokolle zur Verfügung stehen, öffnet die PII **255** einen Socket für jedes zur Verfügung stehende Protokoll abhängig von der Art des Protokolls und richtet eine Protokollabbildungstabelle ein. Diese Tabelle weist eine Eins-zu-Eins-Entsprechung zwischen einer Zugriffsleitung und einer Art von Protokollstapel auf. Wie vorstehend beschrieben, ist die Abbildungstabelle eine von vielen möglichen Implementierungen. Die Protokollabbildung kann beispielsweise ebenso als eine Bitabbildung oder eine andere Datenstruktur implementiert werden. Der Schlüssel liegt darin, dass eine Art und Weise des Angebens einer Eins-zu-Eins-Entsprechung zwischen der Zugriffsleitung und dem Protokollstapel vorliegt. Die in dem bevorzugten Ausführungsbeispiel verwendete Abbildungstabelle wird durch Zuweisen einer Zugriffsleitung zu jedem zur Verfügung stehenden Protokoll und durch Speicherung von Daten, die die einem jeden Protokoll zugewiesenen Zugriffsleitung in einem Speicherabschnitt angibt, der zur Verwendung durch die PII **255** reserviert ist, gebildet. Fig. 7A zeigt ein Beispiel einer Protokollabbildungstabelle, die durch die PII **255** unter Verwendung der beispielbezogenen Zugriffleitungszuweisungen, die gemäß Fig. 3 gezeigt sind, erstellt werden kann, wenn UDP-, IPX- und DDP-Protokolle zur Verfügung stehen.

[0059] Nach einer Initialisierung der PII **255** in Schritt **601** geht der Ablauf zu Schritt S602 über, in dem ein Öffnen-Befehl durch den SNMP-Verwalter **265** zum Öffnen einer Sitzung ausgeführt wird. Dieser Befehl gibt eine Zugriffs-ID für ein zu verwendendes Verwaltungsprogramm zurück, um sich selbst zu identifizieren, z. B. Zugriffs-ID #2 für den SNMP-Verwalter **265**. Die PII **255** speichert Daten, die die Beziehung zwischen Zugriffs-IDs und Verwaltungsprogrammen angeben, in dem Speicherabschnitt, der zur Verwendung durch die PII **255** reserviert ist. Fig. 8A zeigt ein Beispiel einer Zugriffs-ID-Abbildungstabelle, die durch die PII **255** eingerichtet ist, um die Beziehung zwischen Zugriffs-IDs und Verwaltungsprogrammen anzugeben. Anhand der vorstehend beschriebenen Protokollabbildungstabelle ist die Abbildung zwischen Zugriffs-IDs und Anwendungsprogrammen zu vielen Implementierungen in der Lage, die sich von einer Abbildungstabelle unterscheiden, z. b. eine Bitabbildung oder eine andere Datenstruktur. Die Daten gemäß Fig. 8A entsprechen der beispielbezogenen Zuweisung von Zugriffs-IDs, die gemäß Fig. 3 gezeigt sind. Die Zugriffs-IDs sind als XXXX1 oder XXXX2 bezeichnet, wobei XXXX die MAC-Adresse von PC **20** darstellt und die 1 und 2 den CMIP-Verwalter **275** bzw. den SNMP-Verwalter **265** angeben. Der Zweck einer Zugriffs-ID liegt im eindeutigen Identifizieren von unterschiedlichen Funktionseinheiten, die die PII verwenden, beispielsweise der CMIP-Verwalter **275** und der SNMP-Verwalter **265**.

[0060] Als nächstes geht der Ablauf zu Schritt S603 über, in dem ein Datenpaket von dem SNMP-Verwalter **265** zu der PII **255** gesendet wird. Das Paket wird durch Bereitstellen eines Zeigers auf den Paketort im Speicher zusammen mit der Zugriffs-ID für den SNMP-Verwalter **265** und einer Ziel-ID für das Paketziel bereitgestellt. Da keine ein Übertragungsprotokoll angebenden Informationen für das Paket erforderlich sind, ist das Paket protokollunabhängig, und es ist eine einzelne Schnittstelle zwischen einem Anwendungsprogramm und allen Protokollstapeln bereitgestellt. Die Ziel-ID des entsprechenden Agenten, z. B. SNMP-Agent **260**, der dem SNMP-Verwalter **265** entspricht, wird in dem bevorzugten Ausführungsbeispiel durch Durchführen einer Lokalisiere Agent-Funktion durchgeführt, bevor Daten zu einem Agenten gesendet werden. In dem Fall beispielswei-

se, in dem Novell-Netware verwendet wird, wird diese Funktion durch Nachsehen in einem Bindewerk in dem Dateien-Server **70** durchgeführt, um Vorrichtungen zu bestimmen, die in dem Bindewerk registriert sind, z. B. durch Verwenden der SAP-Funktion von IPX, und die kompatible Agenten aufweisen können. Eine Kommunikation mit jenen Vorrichtungen findet dann zum Erhalten einer Ziel-ID für den Agenten statt. In einer UNIX-Umgebung wird die Lokalisiere Rgent-Funktion durch Verwenden von vorbestimmten Host-Tabellen durchgeführt. In einer AppelTalk-Umgebung wird der Agent durch Verwenden eines eindeutigen Namens lokalisiert. Es kann ebenso jedwede andere Technik verwendet werden, die dem Verwaltungsprogramm ermöglicht, eine Ziel-ID für ein Agentenprogramm zu erhalten.

[0061] Nachdem das Paket zu PII **255** gesendet ist, geht der Ablauf zu Schritt S604 über, in dem ein Protokoll zum Senden des Datenpakets ausgewählt wird. Steht lediglich ein Protokoll zur Verfügung, dann wird jenes Protokoll verwendet. In dem bevorzugten Ausführungsbeispiel wird ein Protokoll durch Verwenden eines voreingestellten oder bevorzugten Protokolls, falls verfügbar, ausgewählt. Beispielsweise ist UDP das bevorzugte Protokoll zum Übertragen von Daten zwischen dem SNMP-Verwalter **265** und dem SNMP-Agenten **260**. Steht das bevorzugte Protokoll nicht zur Verfügung, dann wird das erste zur Verfügung stehende Protokoll verwendet. Es gibt jedoch viele alternative Varianten zum Auswählen eines zu verwendenden Protokolls. Ein Protokoll kann beispielsweise zufällig ausgewählt werden, oder es kann das erste zur Verfügung stehende Protokoll verwendet werden. Es kann ebenso das den geringsten Verkehr aufweisende Protokoll verwendet werden. In einer Novell-Umgebung beispielsweise können Sammlungsfunktionen, wie HohleLokalesZiel („GetLocalTarget") einen Schätzwert der Zeit zurückgeben, die zum Zuführen eines 567-Byte-Pakets zu einem ausgewiesenen Ziel erforderlich ist. Diese Funktionen stellen unter Verwendung des entsprechenden Protokolls Informationen bereit, die auf Netzwerkverkehr bezogen sind. Ein Aufruf jener Funktionen kann zum Erhalten von Informationen verwendet werden, die angeben, welches Protokoll den geringsten Verkehr aufweist. Ferner kann die PII **255** einen Zähler für jedes Protokoll (oder jede Zugriffsleitung) speichern und kann diejenige auswählen, die die PII **255** am wenigsten häufig verwendet hat, oder die PII **255** kann einen Zeitpunkt speichern, zu dem jedes Protokoll (oder jede Zugriffsleitung) als letztes verwendet wurde, und kann diejenige auswählen, deren Verwendung am weitesten zurück liegt.

[0062] Nach Auswählen eines Protokolls zum Übertragen des Datenpakets geht der Ablauf zu Schritt S605 über, in dem die PII **255** einen Block von protokollspezifischen Informationen, d. h. eine protokollspezifische Adresse, aus einer in einem Speicher gespeicherten Protokolladressabbildungstabelle holt. Die Informationen werden auf der Grundlage der Zugriffs-ID des Übertragungsprogramms und der Zugriffsleitung geholt, die dem ausgewählten Protokoll entspricht. Weist die Abbildungstabelle keinen Eintrag für das Zugriffs-ID/Zugriffs-Leitungspaar auf, beispielsweise wenn das Paket das erste Paket ist, das durch die PII **250** für einen bestimmten Verwalter unter Verwendung eines bestimmten Protokolls übertragen wird, dann wird der Tabelle ein Eintrag hinzugefügt.

[0063] Fig. 9A zeigt ein Beispiel einer Protokolladressabbildungstabelle, die durch die PII **255** erstellt ist. Es liegt eine Eins-zu-Eins-Entsprechung zwischen einem Zugriffs-ID/Zugriffsleitungspaar und einer protokollspezifischen Adresse vor. Bezugszeichen **901** gibt eine Spalte von Zugriffs-ID/Zugriffsleitungspaaren an. Bezugszeichen **902** stellt eine Spalte von protokollspezifischen Adressen dar, von denen jede einen Protokollheader einer in Spalte **903** angegebenen Art enthält. Jeder Protokollheader weist ein unterschiedliches Format auf. Die verschiedenen Protokollheaderformate sind nachstehend ausführlich unter Bezug auf Fig. 10A bis Fig. 10C beschrieben. Ohne Rücksichtnahme auf die Art enthält jedoch jeder Protokollheader eine Art von Adressdaten, z. b. einen Socket für einen IPX-Header, einen Port für einen UDP-Header und einen Socket und Namen für einen DDP-Header. Bezugszeichen **904** gibt eine Spalte an, die die spezifischen Adressen zeigt, die in jedem Protokollheader enthalten sind.

[0064] Die in Spalte **904** angegebenen Adressdaten stellen Standardadressdaten dar, die zur Verwendung durch einen einzelnen Verwalter und Protokoll definiert sind. Beispielsweise verwendet das SNMP-Protokoll (Zugriffs-ID XXXX2 gemäß Fig. 8A und Fig. 9A) die nachstehenden Adressdaten: (1) für IPX „Socket" $900F_H$ und 9010_H (Agenten-Socket bzw. Unterbrechungs-Socket), (2) für UDP „Port" 160_H und 161_H und (3) für DDP einen eindeutigen Namen „SNMP-Agent" und „SNMP-Unterbrechungsverwalter" und „Socket" 8_H und 9_H (Agenten-Socket bzw. Unterbrechungs-Socket). Dementsprechend verwendet das CMIP protokollspezifische Adressdaten, wie einen CMIP-Port für UDP, einen CMIP-Socket für IPX und ein CMIP-Namen und -Socket für DDP. Ein SNMP-Protokollpaket und ein CMIP-Protokollpaket können die gleiche Header-Art aufweisen, aber jeder Header enthält unterschiedliche spezifische Adressdaten. Beispielsweise haben die protokollspezifischen Adressen, die in Zeilen 1 und 4 gemäß Fig. 9A gezeigt sind, jede einen Header vom Typ UDP aber jeder Header enthält unterschiedliche Adressdaten, wie in Spalte **904** gezeigt. Weist SNMP-Verwalter **265** Zugriffs-ID XXXX2 auf, wie gemäß Fig. 8A gezeigt, und wird das bevorzugte Protokoll UDP verwendet und weist dieses Zugriffsleitung #1 auf, wie

gemäß Fig. 7A gezeigt, dann wird die protokollspezifische Adresse in der vierten Zeile gemäß Fig. 9A durch die PII **255** geholt, die dem Zugriffs-ID/Zugriffsleitungspaar XXXX2/1 entspricht.

[0065] Wird die Erfindung bei Anwendungsprogrammen verwendet, die keine vordefinierten Sockets aufweisen, dann müssen Standardidentifikationswerte für jene Anwendungsprogramme definiert werden und müssen die sowohl mit dem Verwalter als auch Agentenprogrammen verwendeten PIIs programmiert werden, jene Standardidentifikationswerte zu verwenden.

[0066] Wie vorstehend beschrieben, weist jede Art von Protokollheader ein unterschiedliches Format auf, wie hinsichtlich Fig. 10A bis Fig. 10C beschrieben. Fig. 10A zeigt ein Format für einen Lokalnetzwerkrahmen (oder „Übertragungspaket") **1000**, wenn der Rahmen unter Verwendung eines UDP-Protokolls übertragen wird. In diesem Fall enthält der Netzwerkrahmen **1000** einen Lokalnetzwerk-Header **1010** und eine Lokalnetzwerk-Endkennung **1015**. Er enthält ebenso einen IP-Header **1021**, einen UDP-Header **1020** und Daten **1030**. Wie gemäß Fig. 10A gezeigt, enthält der UDP-Header **1020** Felder für einen Quellport, einen Zielport, eine Länge und eine Prüfsumme. Fig. 10B zeigt ein Format für einen Netzwerkrahmen **1000**, wenn der Rahmen unter Verwendung eines IPX-Protokolls übertragen wird. In jenem Falle enthält der Netzwerkrahmen **1000** keinen IP-Header **1021**, und enthält einen IPX-Header **1025** anstelle des UDP-Headers **1020**. Der IPX-Header **1025** enthält Felder für eine Prüfsumme, eine Paketlänge, einen Transportsteuerwert, eine Paketart, ein Zielnetzwerk, einen Zielknoten, ein Zielsocket, ein Quellnetzwerk, einen Quellknoten und ein Quellsocket. Als letztes zeigt Fig. 10C ein Format für Netzwerkrahmen **1000**, wenn der Rahmen unter Verwendung eines DDP-Protokolls übertragen wird. In diesem Fall fehlt dem Netzwerkrahmen ebenso der IP-Header **1021** und er enthält einen DDP-Header **1028** anstelle sowohl des UDP-Headers **1020** als auch des IPX-Headers **1025**. Der DDP-Header **1028** enthält Felder, die 00, Hop, eine Datagrammlänge, eine Datagrammprüfsumme, ein Zielnetzwerk, ein Quellnetzwerk, eine Zielknoten-ID, eine Quellknoten-ID, eine Zielsocketnummer, eine Quellsocketnummer, und eine DDP-Art enthalten.

[0067] Unter Bezugnahme auf Fig. 6 geht der Ablauf nach Holen der protokollspezifischen Informationen zu Schritt S606 über, in dem die PII **255** ein Übertragungspaket bildet, das für das ausgewählte Protokoll spezifisch ist, d. h. ein Netzwerkrahmen **1000**, der eines der gemäß den Fig. 10A bis Fig. 10C gezeigten Formate aufweist. Der Übertragungsrahmen wird unter Verwendung der Zielinformationen, die durch den SNMP-Verwalter **265** bereitgestellt sind, und der Informationen gebildet, die aus der Protokolladressabbildungstabelle in Schritt **605** geholt wurden. Ferner sind Daten **1030** die Daten, die durch den SNMP-Verwalter **265** gesendet werden.

[0068] Nach Bilden eines Übertragungspakets **1000** geht der Ablauf zu Schritt S607 über, in dem ein Übertragungspaket **1000** über das LAN **10** zu dem Zielanwendungsprogramm übertragen wird.

[0069] Fig. 11 zeigt ein Ablaufdiagramm von Prozessschritten zum Empfangen eines Übertragungspakets **1000** bei der NEB **100**. In Schritt S1101 erteilt der SNMP-Agent **260** einen Befehl zum Initialisieren der PII **250**. Die PII **250** muss initialisiert werden, bevor sie Daten aus dem LAN **10** oder dem SNMP-Agenten **260** holen kann. Wie vorstehend beschrieben, erhält der SNMP-Agent **260** eine Tabelle von Eintrittspunkten in Routinen von PII **250** bei Initialisierung, und PII **250** bestimmt durch Erhalten von Informationen von dem PRETASK-Softwaremodul, welche Protokolle zur Verfügung stehen. Fig. 7B zeigt eine beispielbezogene Protokolltabelle, die durch die PII **250** eingerichtet und gespeichert wird, wenn das UDP-, IPX-, und DDP-Protokoll zur Verfügung stehen und Zugriffsleitungen wie gemäß Fig. 3 gezeigt zugewiesen sind.

[0070] Der Ablauf geht dann zu Schritt S1102 über, in dem der SNMP-Agent **260** einen PII-Öffnen-Befehl erteilt, um eine Zugriffs-ID zu erhalten. Fig. 8B zeigt eine beispielbezogene Abbildung von Zugriffs-IDs auf Verwaltungsagenten, die durch die PII **255** auf der Grundlage der beispielbezogenen, gemäß Fig. 3 gezeigten Zuweisung von Zugriffs-IDs eingerichtet und gespeichert wird. Gemäß Fig. 8B stellt YYYY die MAC-Adresse von NEB **100** dar. Der SNMP-Agent **260** ist eine passive Funktionseinheit, die antwortet, wenn der SNMP-Verwalter **265** Informationen anfordert, aber keine Kommunikation initiiert. Dementsprechend horcht der SNMP-Agent **260** nach Paketen auf allen zur Verfügung stehenden Protokollen, d. h. Pakete, die an jedweden zur SNMP-Verwendung definierten Socket adressiert sind. Ist deshalb erst einmal eine Zugriffs-ID für einen Agenten bereitgestellt, z. B. SNMP-Agent **260**, dann wird eine Abbildungstabelle mit Zugriffs-ID/Zugriffsleitungspaareinträgen für jede Zugriffsleitung erstellt. Fig. 9A zeigt eine beispielbezogene Protokolladressabbildungstabelle für die PII **250** auf der Grundlage der beispielbezogenen Daten gemäß Fig. 7A und Fig. 8A.

[0071] Der Ablauf geht dann zu Schritt S1103 über, in dem ein Übertragungspaket aus dem LAN **10** empfangen wird. Unter Bezugnahme auf Fig. 2 wird Empfangspaket **1000** aus dem LAN **10** durch die Netzwerkschnittstellenansteuereinrichtung **210** empfangen und wird zu dem Multiplexersoftwaremodul **222** und dann zu einem der Protokollstapel **230** bis **232** verkehrsgelenkt. Stellt beispielsweise Übertragungspaket **1000** Daten dar, die von dem SNMP-Ver-

walter **265** unter Verwendung des bevorzugten UDP-Protokolls zu SNMP-Agent **260** gesendet wurden, dann wird das Übertragungspaket **1000** zu dem UDP-Protokollstapel **231** verkehrsgelenkt. Die PII **250** horcht nach Paketen, die an spezifische Sockets adressiert sind, z. B. Sockets, die zur Verwendung durch Verwaltungsprogramme definiert sind, und ignoriert alle anderen. Da das Übertragungspaket **1000** an einen der Sockets adressiert sind, nach dem PII **250** horcht, empfängt PII **250** das Paket.

[0072] Der Ablauf geht dann zu Schritt S1104 über, in dem PII **250** ein Zugriffs-ID/Zugriffsleitungspaar auf der Grundlage von protokollspezifischen Informationen **1020** in Übertragungspaket **1000** durch Bezugnahme auf die gemäß

[0073] Fig. 9A gezeigte Protokolladressabbildungstabelle für PII **250** erhält. Für das beispielbezogene Übertragungspaket **1000** wird Zugriffs-ID/Zugriffsleitungspaar YYYY1/2 aus Zeile **2** gemäß Fig. 9A geholt.

[0074] Der Ablauf geht dann zu Schritt S1105 über, in dem Daten **1030** zu dem geeigneten Verwaltungsprogramm durchgereicht werden. Dies wird durch Bezugnahme auf die Zugriffs-ID-Abbildungsinformation für PII **250** durchgeführt, was vorstehend unter Bezugnahme auf Fig. 8B beschrieben ist. Zugriffs-ID YYYY1 entspricht beispielsweise dem SNMP-Agenten **260**, und so werden Daten **1030** des Übertragungspakets **1000** zu dem SNMP-Agenten **260** durchgereicht.

[0075] Obwohl vorstehend ein Beispiel zum Übertragen von Daten von einem Verwalter in PC **20** zu einem Agenten in NEB **100** beschrieben wurde, ist derselbe Prozess für ein Übertragen von Daten von NEB **100** zu PC **20** anwendbar. Des Weiteren ist die Erfindung wie vorstehend beschrieben nicht auf eine Übertragung von Daten zwischen Verwaltungsanwendungsprogrammen beschränkt, und ist ebenso wenig nicht auf Netzwerkübertragungen beschränkt. Es ist dementsprechend zu verstehen, dass während das bevorzugte Ausführungsbeispiel der Erfindung vorstehend beschrieben ist, die Erfindung nicht auf die vorstehend beschriebenen Ausführungsbeispiele beschränkt ist, und das vielfältige Änderungen und Modifikationen durch den Durchschnittsfachmann durchgeführt werden können, ohne den Schutzbereich der Erfindung zu verlassen.

Patentansprüche

1. Verfahren zum Zuführen eines Datenpakets, das von einem ersten Anwendungsprogramm empfangen wird, das auf einer ersten Vorrichtung läuft, die eine Schnittstelle zu einem lokalen Bereichsnetzwerk (**10**) aufweist, zu einem zweiten Anwendungsprogramm, das auf einer zweiten Vorrichtung läuft, die eine Schnittstelle zu dem lokalen Bereichsnetzwerk (**10**) aufweist, wobei das Verfahren die Schritte umfasst:
Empfangen eines protokollunabhängigen Datenpakets von dem ersten Anwendungsprogramm zusammen mit Daten, die das erste Anwendungsprogramm identifizieren, und einer Ziel-ID, die das zweite Anwendungsprogramm identifiziert,
Bestimmen, welche Protokolle in dem lokalen Bereichsnetzwerk (**10**) zur Verwendung zur Verfügung stehen, und
Zuweisen einer jeweiligen Zugriffsleitung (**281**, **282**, **283**) zu jedem der Protokolle, die in dem lokalen Bereichsnetzwerk (**10**) zur Verwendung zur Verfügung stehen,
wobei das Verfahren gekennzeichnet ist durch die Schritte:
Auswählen, aus den zur Verfügung stehenden Protokollen, des Protokolls, das der den geringsten Verkehr aufweisenden Zugriffsleitung zugewiesen ist,
Bestimmen von protokollspezifischen Informationen, die eine Art eines Protokoll-Headers und Adressinformationen in dem Header enthalten, die Daten entsprechen, die das erste Anwendungsprogramm und das in dem Auswahlschritt ausgewählte Protokoll identifizieren,
Bilden eines Übertragungspakets, das das Datenpaket, die Ziel-ID und die bestimmten protokollspezifischen Informationen enthält, und
Übertragen des Übertragungspakets über das lokale Bereichsnetzwerk (**10**) zu dem zweiten Anwendungsprogramm.

2. Verfahren gemäß Anspruch 1, wobei das protokollunabhängige Schnittstellenprogramm:
dem ersten Anwendungsprogramm eine eindeutige Zugriffs-ID zuweist, und
Abbildungsinformationen erstellt, die eine Eins-zu-Eins-Entsprechung zwischen jeder/jedem Zugriffs-ID/Zugriffsleitungspaar (**90**) und einem Block von protokollspezifischen Informationen aufweisen, und wobei der Bestimmungsschritt ein Ermitteln eines Blocks von protokollspezifischen Informationen aus den Abbildungsinformationen auf der Grundlage der Zugriffs-ID des ersten Anwendungsprogramms und der Zugriffsleitung (**281**, **282**, **283**), die dem in dem Auswahlschritt ausgewählten Protokoll entspricht, umfasst.

3. Verfahren gemäß Anspruch 1 oder 2, wobei die zweite Vorrichtung eine Netzwerkschnittstellenvorrichtung (**120**) ist, die eine Schnittstelle zwischen einer Peripherie (**135**) und dem lokalen Bereichsnetzwerk (**10**) bildet, und die eine Dienstroutine ausführt, wobei das Verfahren ferner den Schritt des Übertragens von Datenpaketen, die durch die Dienstroutine zu bedienende Daten enthalten, umfasst.

Es folgen 12 Blatt Zeichnungen

Anhängende Zeichnungen

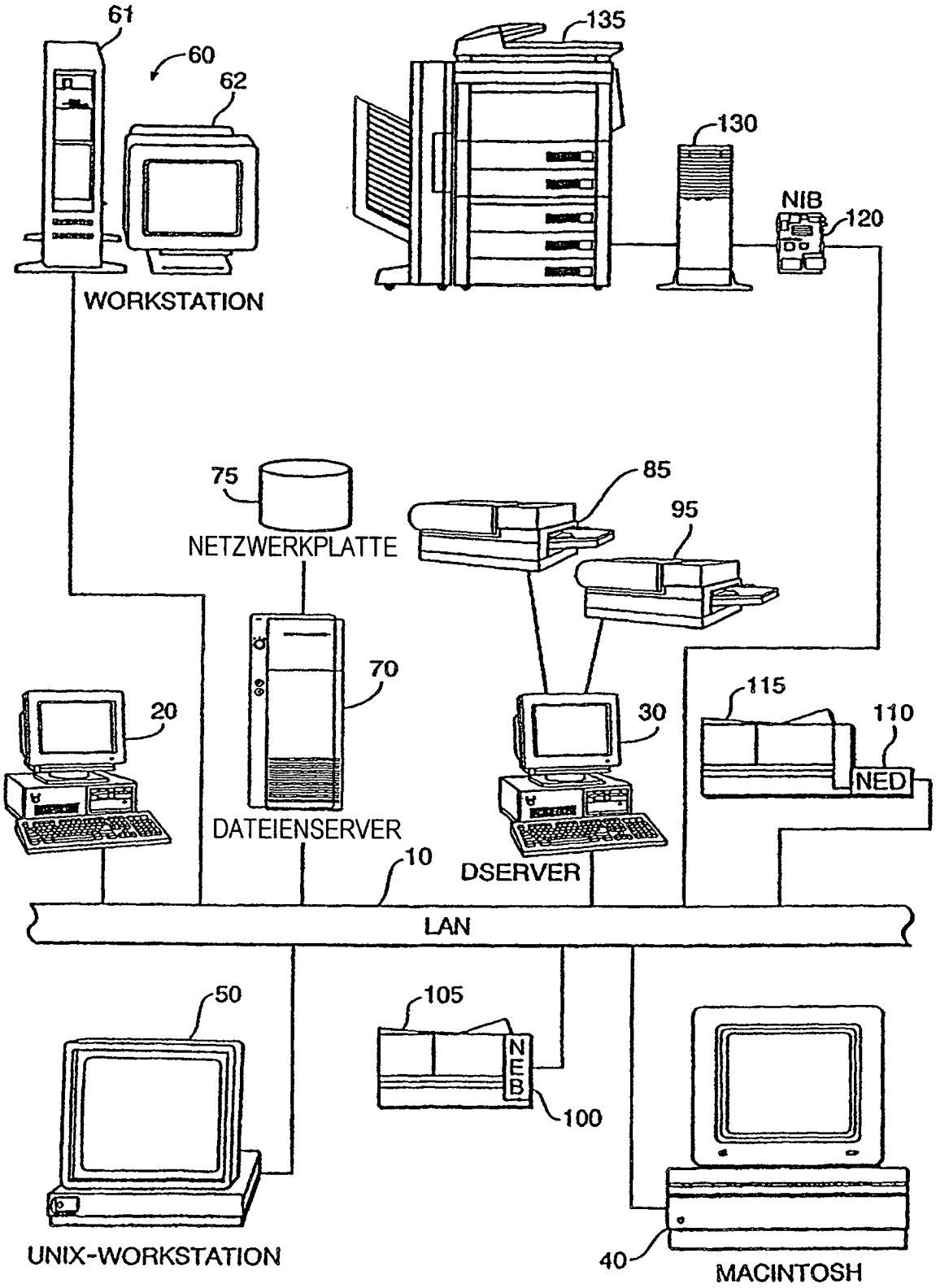


FIG. 1

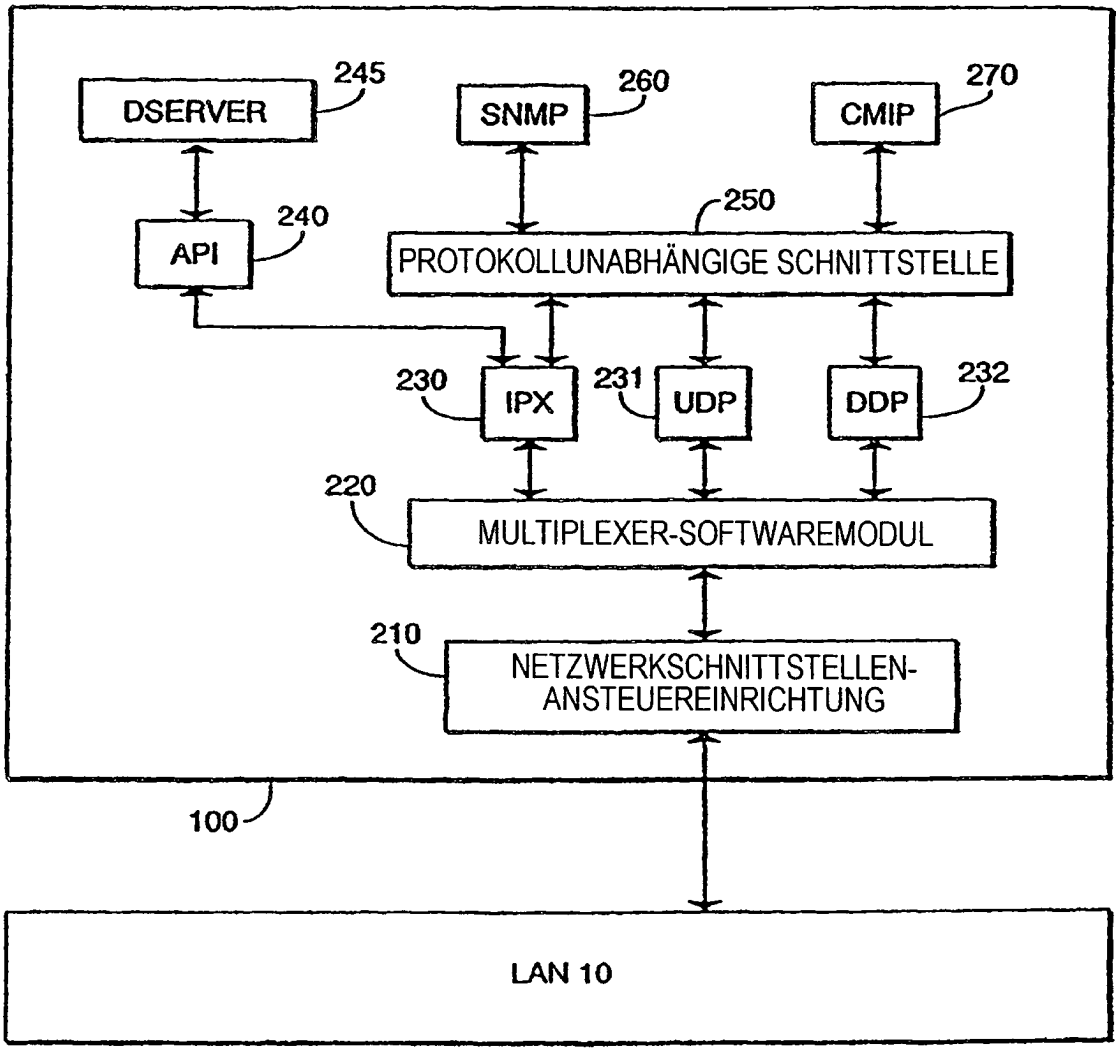

DSERVER
245
SNMP
260
CMIP
270
API
240
250
PROTOKOLLUNABHÄNGIGE SCHNITTSTELLE
230
IPX
231
UDP
DDP
232
220
MULTIPLEXER-SOFTWAREMODUL
210
NETZWERKSCHNITTSTELLEN-
ANSTEUEREINRICHTUNG
100
LAN 10

FIG. 2

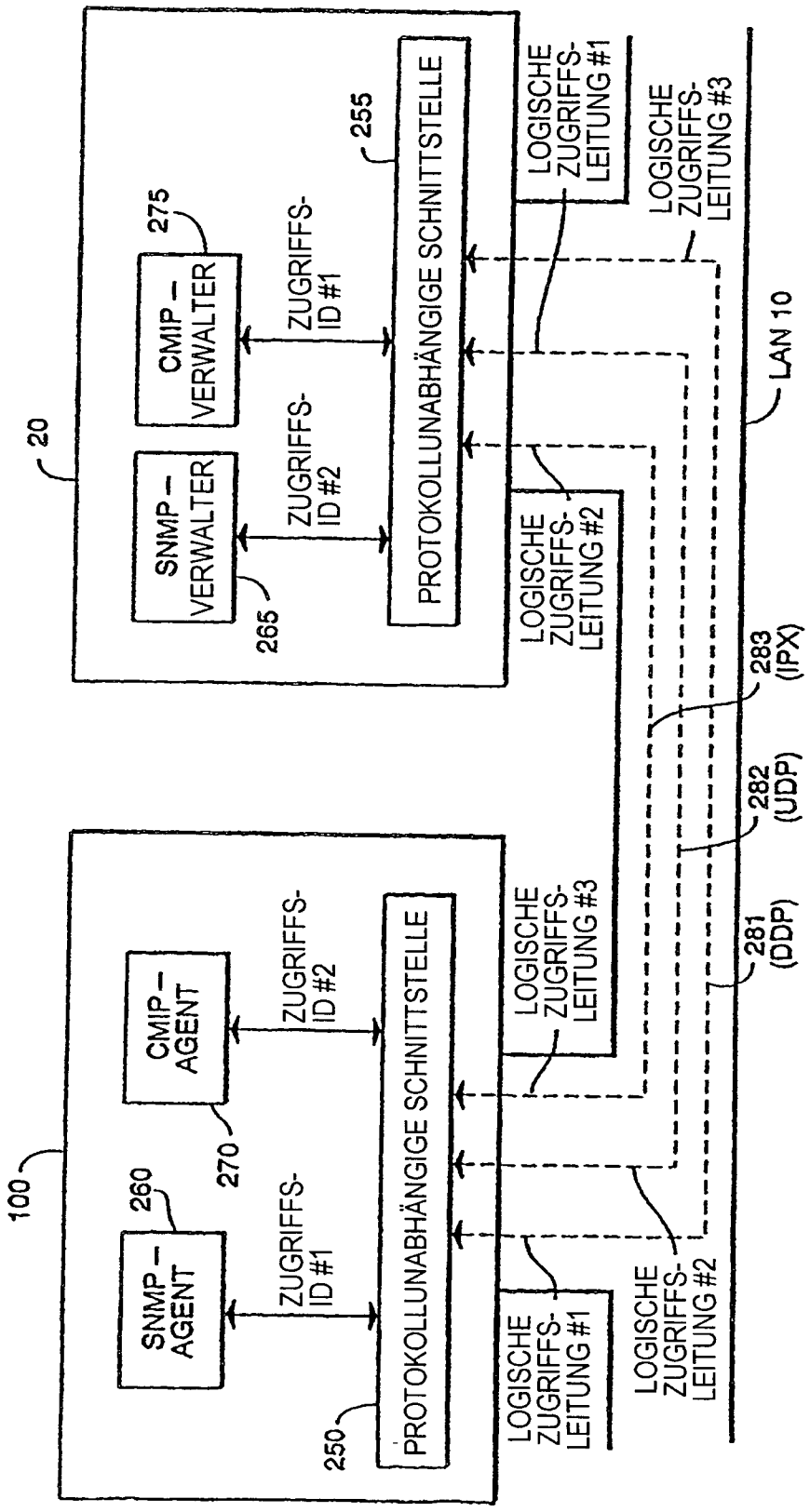

20
SNMP – VERWALTER
265
CMIP – VERWALTER
275
ZUGRIFFS-ID #2
ZUGRIFFS-ID #1
255
PROTOKOLLUNABHÄNGIGE SCHNITTSTELLE
LOGISCHE ZUGRIFFS-LEITUNG #1
LOGISCHE ZUGRIFFS-LEITUNG #2
LOGISCHE ZUGRIFFS-LEITUNG #3
LAN 10
283 (IPX)
282 (UDP)
281 (DDP)
100
SNMP – AGENT
260
CMIP – AGENT
270
ZUGRIFFS-ID #1
ZUGRIFFS-ID #2
250
PROTOKOLLUNABHÄNGIGE SCHNITTSTELLE
LOGISCHE ZUGRIFFS-LEITUNG #1
LOGISCHE ZUGRIFFS-LEITUNG #2
LOGISCHE ZUGRIFFS-LEITUNG #3

FIG. 3

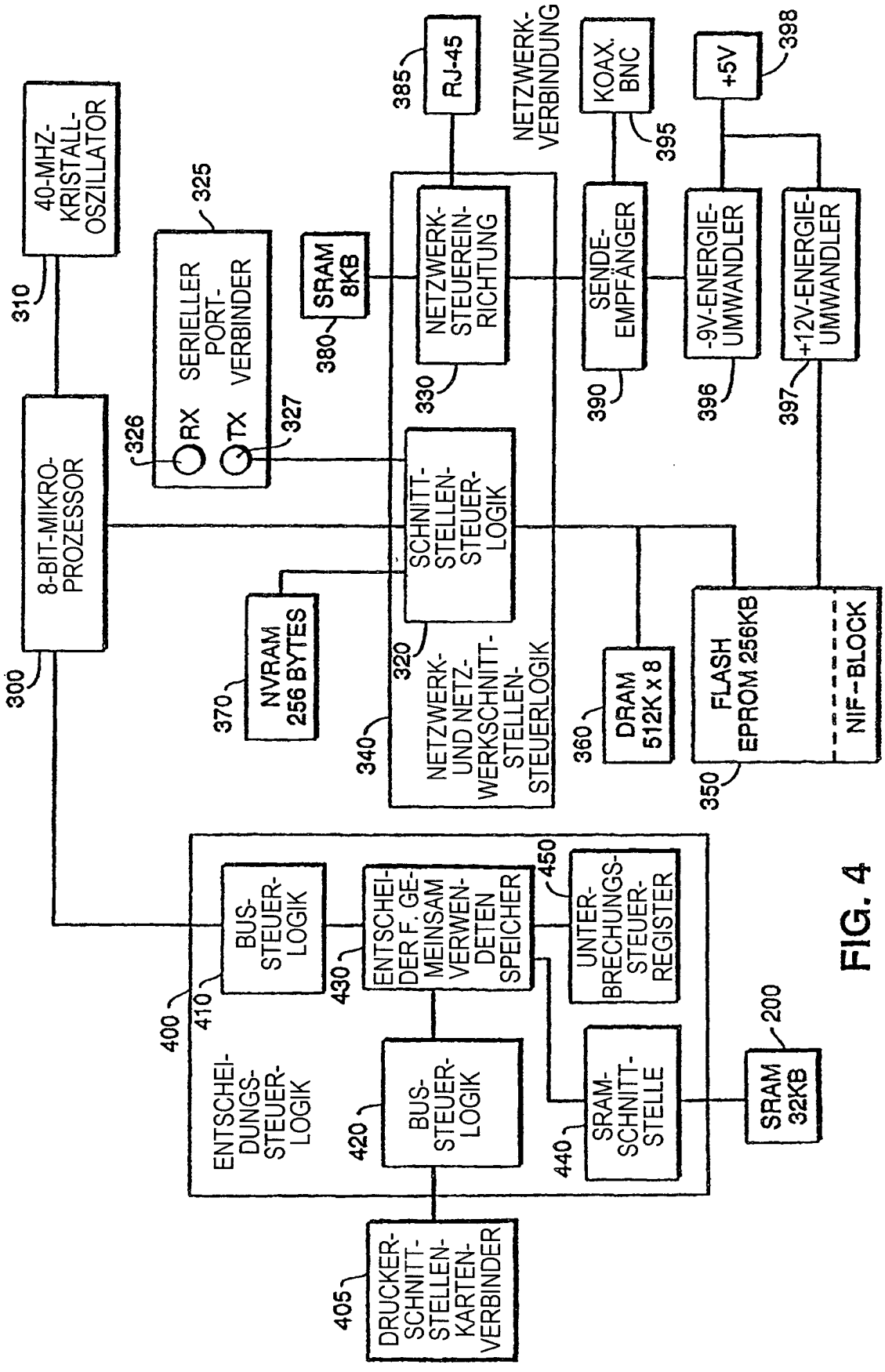

40-MHZ-KRISTALL-OSZILLATOR
310
8-BIT-MIKRO-PROZESSOR
300
SERIELLER PORT-VERBINDER
325
326
RX
TX
327
SRAM 8KB
380
385
RJ-45
NETZWERK-VERBINDUNG
NETZWERK-STEUEREIN-RICHTUNG
330
KOAX. BNC
395
SENDE-EMPFÄNGER
390
+5V
398
-9V-ENERGIE-UMWANDLER
396
+12V-ENERGIE-UMWANDLER
397
SCHNITT-STELLEN-STEUER-LOGIK
320
NVRAM 256 BYTES
370
NETZWERK- UND NETZ-WERKSCHNITT-STELLEN-STEUERLOGIK
340
DRAM 512K x 8
360
FLASH EPROM 256KB
NIF-BLOCK
350
ENTSCHEI-DUNGS-STEUER-LOGIK
400
BUS-STEUER-LOGIK
410
ENTSCHEI-DER F. GE-MEINSAM VERWEN-DETEN SPEICHER
430
450
UNTER-BRECHUNGS-STEUER-REGISTER
BUS-STEUER-LOGIK
420
SRAM-SCHNITT-STELLE
440
SRAM 32KB
200
DRUCKER-SCHNITT-STELLEN-KARTEN-VERBINDER
405

FIG. 4

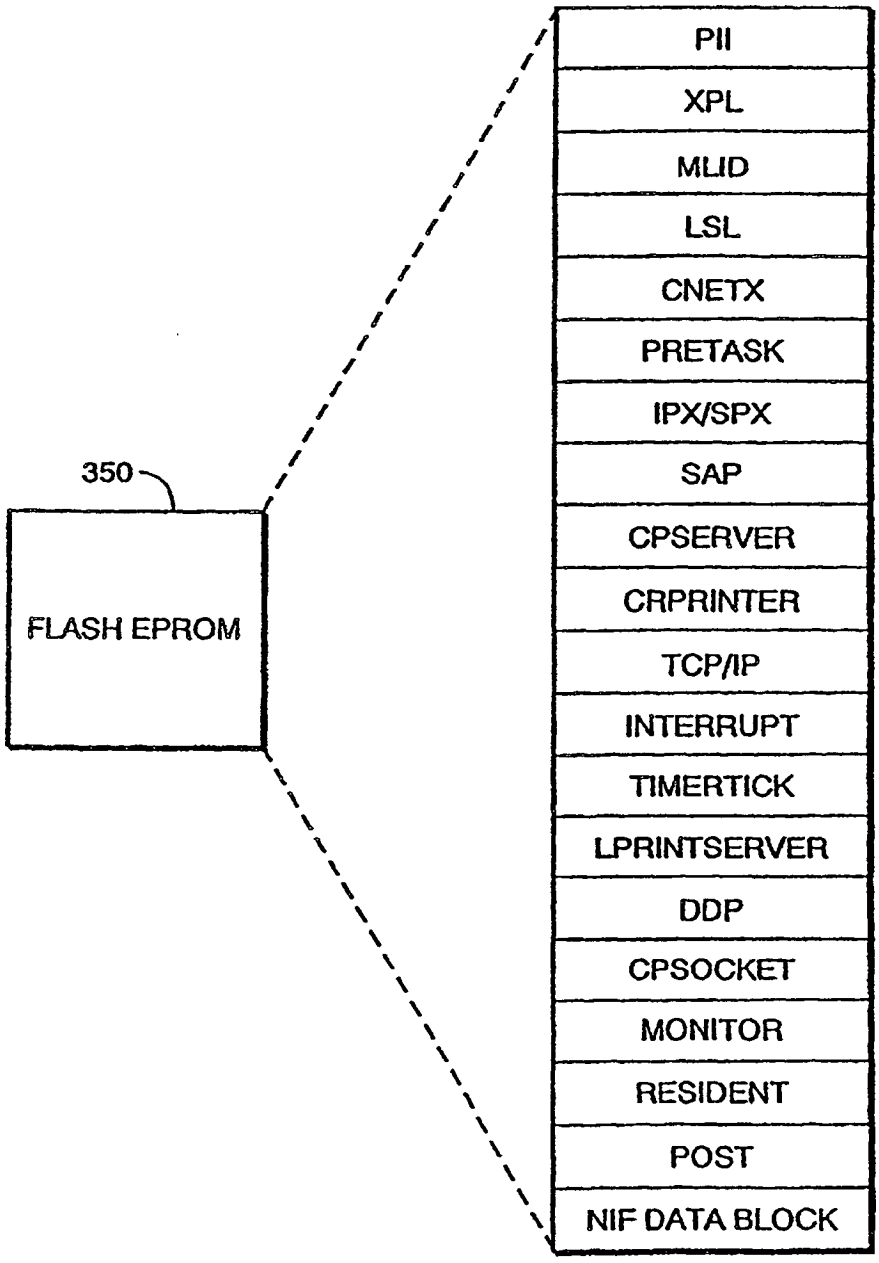

350
FLASH EPROM
PII
XPL
MLID
LSL
CNETX
PRETASK
IPX/SPX
SAP
CPSERVER
CRPRINTER
TCP/IP
INTERRUPT
TIMERTICK
LPRINTSERVER
DDP
CPSOCKET
MONITOR
RESIDENT
POST
NIF DATA BLOCK

FIG. 5

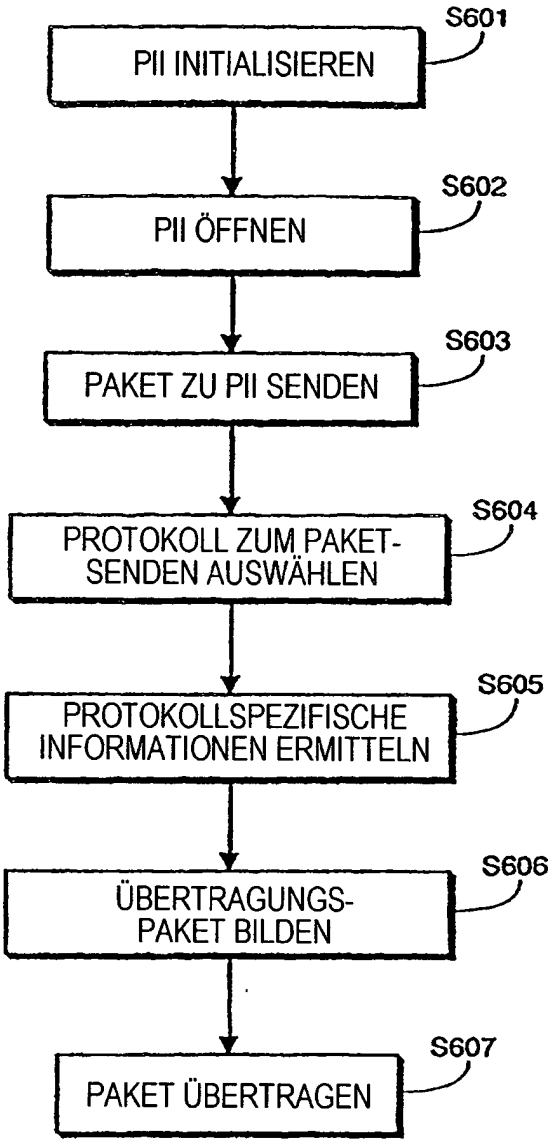

S601
PII INITIALISIEREN
S602
PII ÖFFNEN
S603
PAKET ZU PII SENDEN
S604
PROTOKOLL ZUM PAKET-SENDEN AUSWÄHLEN
S605
PROTOKOLLSPEZIFISCHE INFORMATIONEN ERMITTELN
S606
ÜBERTRAGUNGS-PAKET BILDEN
S607
PAKET ÜBERTRAGEN

FIG. 6

FIG. 7A

PROTOKOLL	ZUGRIFFSLEITUNG #
UDP	1
IPX	2
DDP	3

FIG. 7B

PROTOKOLL	ZUGRIFFSLEITUNG #
UDP	2
IPX	3
DDP	1

FIG. 8A

PROGRAMM	ZUGRIFFS-ID #
CMIP -VERWALTER	XXXX1
SNMP -VERWALTER	XXXX2

FIG. 8B

PROGRAMM	ZUGRIFFS-ID #
CMIP-AGENT	YYYY2
SNMP-AGENT	YYYY1

FIG. 9A

902

ZUGRIFFS-ID/ ZUGRIFFSLEITUNG	PROTOKOLLSPEZIFISCHE ADRESSE	
	HEADER-ART	ADRESSE IM HEADER
XXXX1/1	UDP-HEADER	CMIP-PORT
XXXX1/2	IPX-HEADER	CMIP-SOCKET
XXXX1/3	DDP-HEADER	CMIP-SOCKET/NAME
XXXX2/1	UDP-HEADER	PORT 160/161
XXXX2/2	IPX-HEADER	SOCKET 900F/9010
XXXX2/3	DDP-HEADER	SOCKET 8/9 - SNMP AGENT/SNMP - UNTERBRECHUNGS-BEHANDLUNG

901 903 904

FIG. 9B

ZUGRIFFS-ID/ ZUGRIFFSLEITUNG	PROTOKOLLSPEZIFISCHE ADRESSE	
	HEADER-ART	ADRESSE IM HEADER
YYYY1/1	DDP-HEADER	SOCKET 8/9 - SNMP AGENT/SNMP - UNTERBRECHUNGS-BEHANDLUNG
YYYY1/2	UDP-HEADER	PORT 160/161
YYYY1/3	IPX-HEADER	SOCKET 900F/9010
YYYY2/1	DDP-HEADER	CMIP-SOCKET/NAME
YYYY2/2	UDP-HEADER	CMIP-PORT
YYYY2/3	IPX-HEADER	CMIP-SOCKET

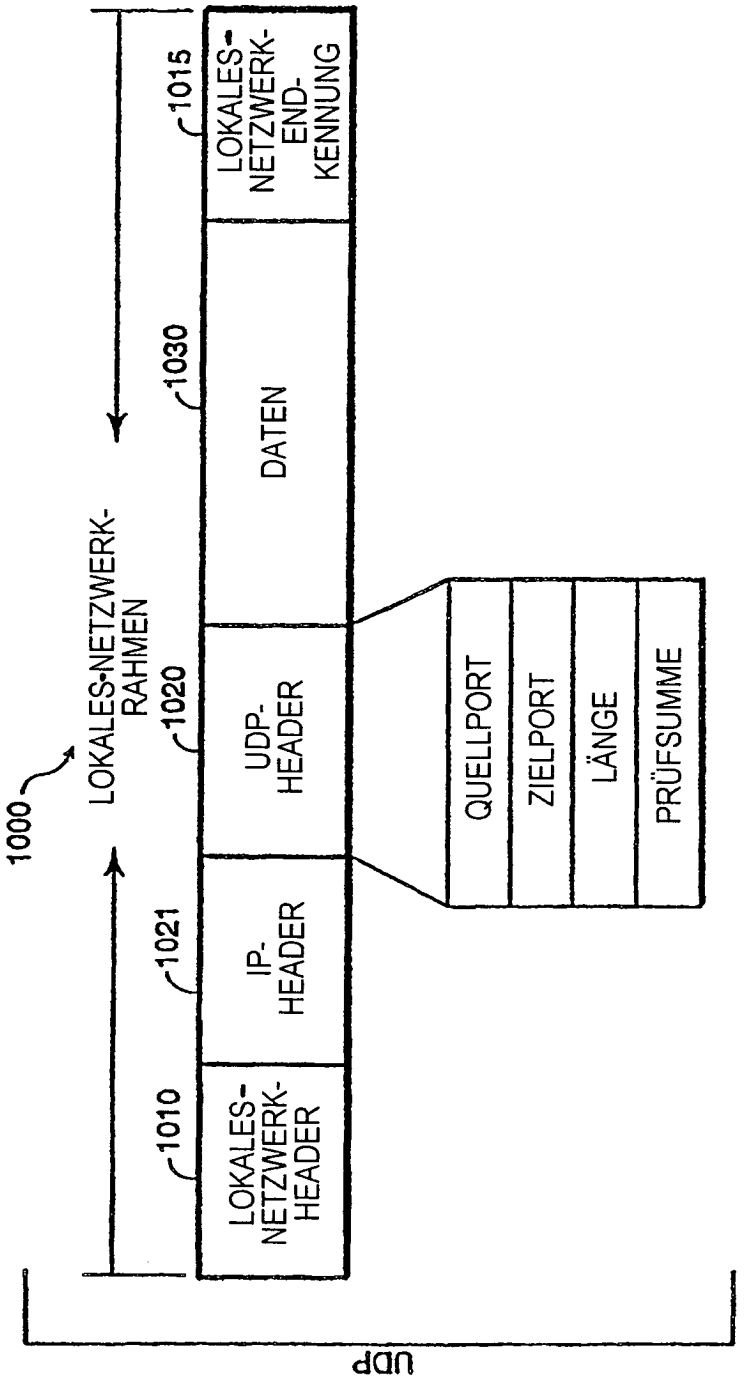

1000
LOKALES-NETZWERK-RAHMEN
1010
LOKALES-NETZWERK-HEADER
1021
IP-HEADER
1020
UDP-HEADER
1030
DATEN
1015
LOKALES-NETZWERK-END-KENNUNG
QUELLPORT
ZIELPORT
LÄNGE
PRÜFSUMME
UDP

FIG. 10A

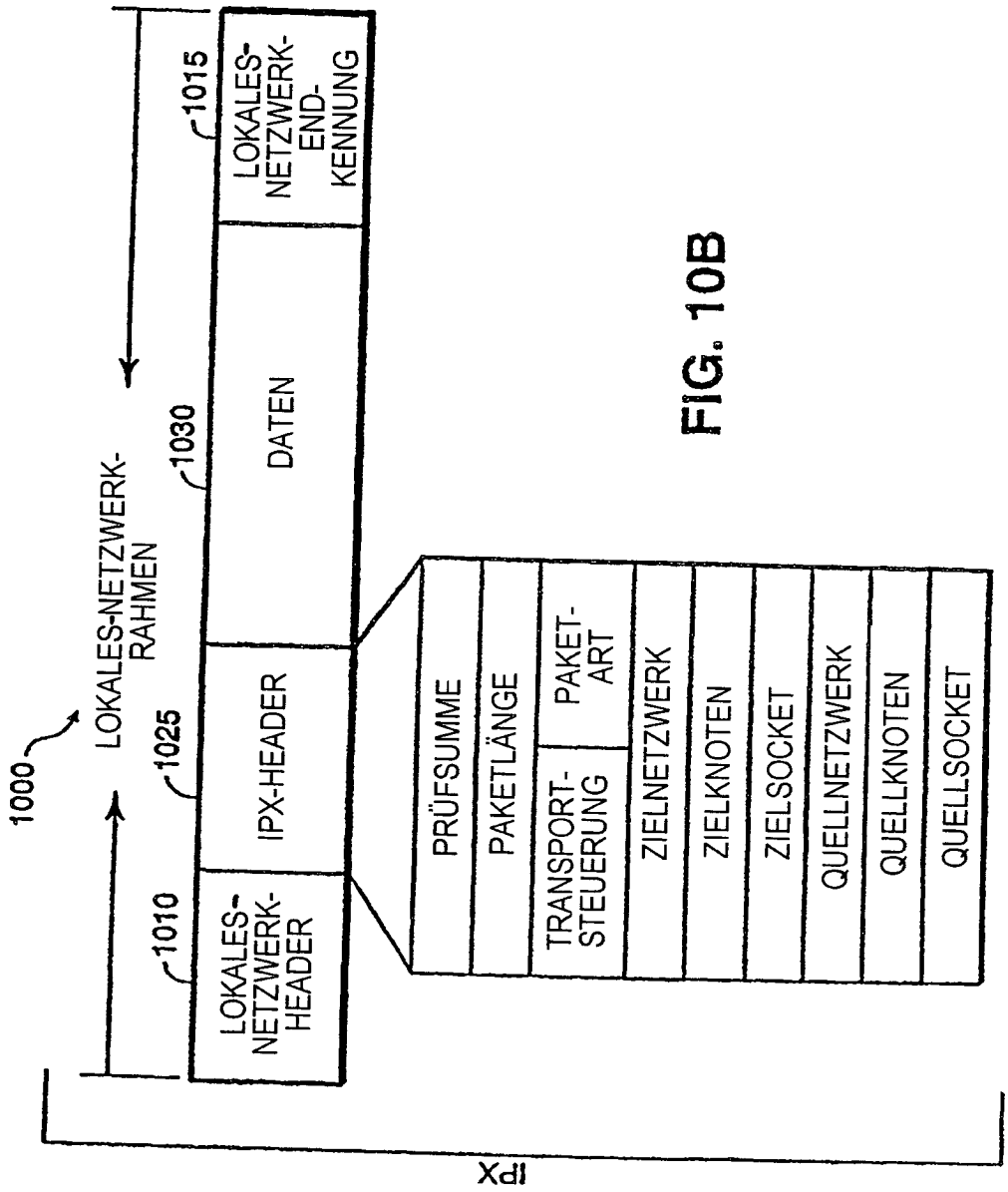

1000
LOKALES-NETZWERK-RAHMEN
1010
LOKALES-NETZWERK-HEADER
1025
IPX-HEADER
1030
DATEN
1015
LOKALES-NETZWERK-END-KENNUNG
PRÜFSUMME
PAKETLÄNGE
TRANSPORT-STEUERUNG
PAKET-ART
ZIELNETZWERK
ZIELKNOTEN
ZIELSOCKET
QUELLNETZWERK
QUELLKNOTEN
QUELLSOCKET
IPX

FIG. 10B

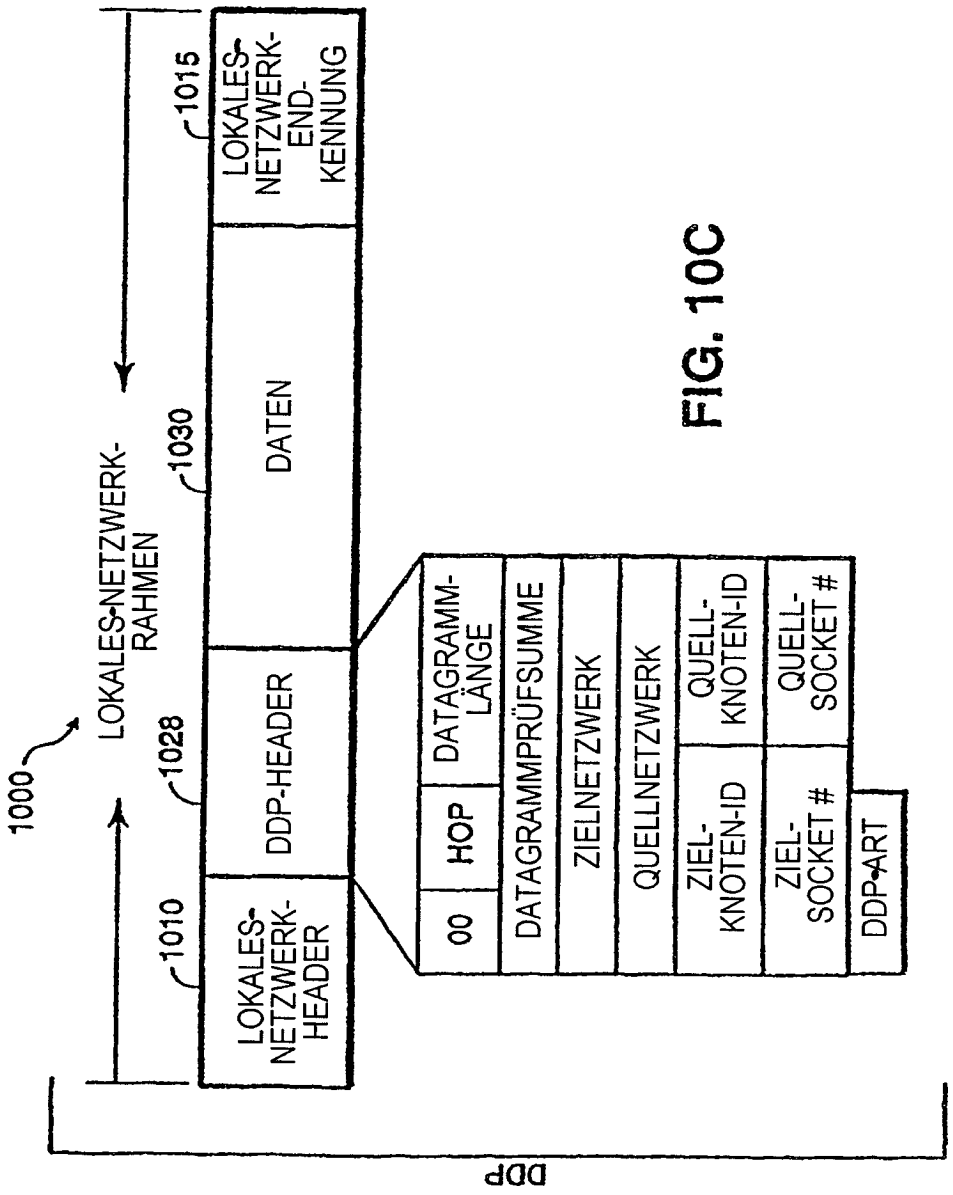

1000
LOKALES-NETZWERK-RAHMEN
1010
LOKALES-NETZWERK-HEADER
1028
DDP-HEADER
1030
DATEN
1015
LOKALES-NETZWERK-END-KENNUNG
00
HOP
DATAGRAMM-LÄNGE
DATAGRAMMPRÜFSUMME
ZIELNETZWERK
QUELLNETZWERK
ZIEL-KNOTEN-ID
QUELL-KNOTEN-ID
ZIEL-SOCKET #
QUELL-SOCKET #
DDP-ART
DDP

FIG. 10C

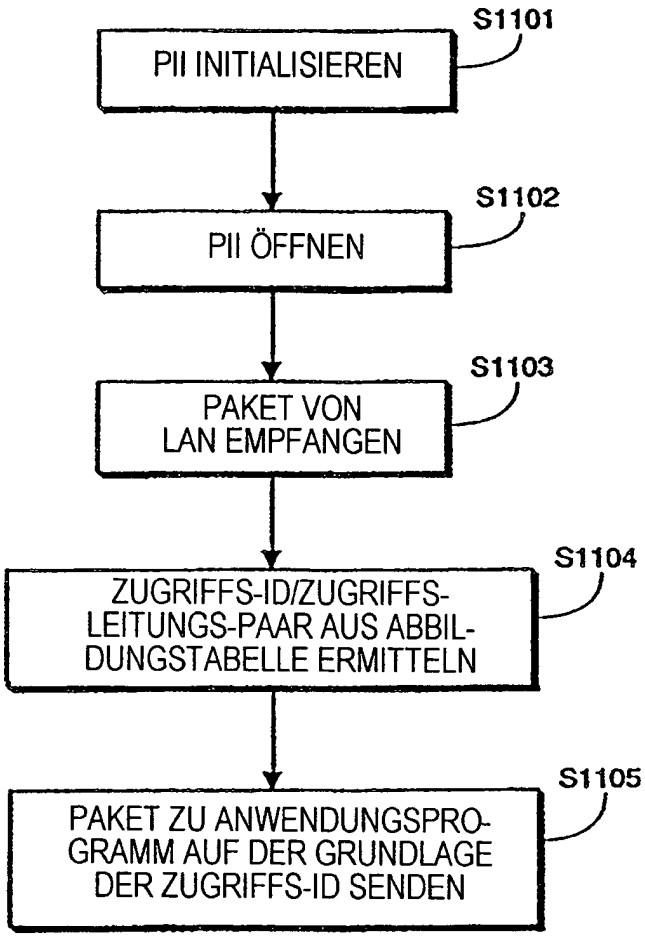

S1101
PII INITIALISIEREN
S1102
PII ÖFFNEN
S1103
PAKET VON LAN EMPFANGEN
S1104
ZUGRIFFS-ID/ZUGRIFFS-LEITUNGS-PAAR AUS ABBILDUNGSTABELLE ERMITTELN
S1105
PAKET ZU ANWENDUNGSPROGRAMM AUF DER GRUNDLAGE DER ZUGRIFFS-ID SENDEN

FIG. 11