



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 29 066 T2 2007.01.04**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 087 590 B1**

(21) Deutsches Aktenzeichen: **600 29 066.2**

(96) Europäisches Aktenzeichen: **00 308 051.2**

(96) Europäischer Anmeldetag: **15.09.2000**

(97) Erstveröffentlichung durch das EPA: **28.03.2001**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **28.06.2006**

(47) Veröffentlichungstag im Patentblatt: **04.01.2007**

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

G06F 13/12 (2006.01)

G06F 9/46 (2006.01)

G06F 3/12 (2006.01)

H04N 1/00 (2006.01)

(30) Unionspriorität:
407129 27.09.1999 US

(74) Vertreter:
**Schoppe, Zimmermann, Stöckeler & Zinkler, 82049
Pullach**

(73) Patentinhaber:
**Hewlett-Packard Development Company, L.P.,
Houston, Tex., US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(72) Erfinder:
Kumpf, David A., Rocklin, CA 95765, US

(54) Bezeichnung: **Server von Netzperipheriegeräten**

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

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

Beschreibung

[0001] Die vorliegende Erfindung bezieht sich allgemein auf einen Netzwerkperipheriegerätserver, z. B. zur Software- und Firmwaresteuerung eines Mehrfunktionsperipheriegeräts in einem Netzwerk durch einen Server, der mit dem Netzwerk verbunden ist, und Software und Firmware eines Peripheriegerätservers, der einen Zugriff auf eine Mehrzahl von Funktionen eines Mehrfunktionsperipheriegeräts in einem Netzwerk liefert.

[0002] Mehrfunktionsperipheriegeräte (MFP) werden immer beliebter, da dieselben eine Anzahl von unterschiedlichen Funktionen, z. B. Scannen, Drucken, Kopieren und Senden und Empfangen von Fäxen, in einer einzigen Vorrichtung liefern. Bis vor kurzem musste ein MFP direkt an einem Einzelcomputer angeschlossen sein, um auf alle Dienste zuzugreifen, die durch das MFP geboten werden. Diese Anordnung schloss jedoch andere Computer davon aus, auf das MFP zuzugreifen. Wenn das MFP in einem Netzwerk durch einen herkömmlichen Peripheriegerätserver implementiert ist, ist nur die Druckfunktion des MFP für die Clients zugänglich, bei denen es sich normalerweise um Personalcomputer (PC) handelt. Die U.S.-Patentveröffentlichung US 6412022 (25.06.2002) von Kumpf u. a. offenbart einen Netzwerkperipheriegerätserver, der in der Lage ist, gleichzeitig auf sowohl die Scanfunktion als auch die Druckfunktion eines MFP, das in ein Netzwerk geschaltet ist, zuzugreifen.

[0003] Der Peripheriegerätserver der vorgenannten Offenbarung von Kumpf u. a. erlaubt jedoch trotzdem keinen Zugriff auf irgendwelche anderen Funktionen eines MFP außer der Druck- und der Scanfunktion. Ein zusätzlicher Nachteil bekannter Server ist, dass dieselben mit dem angeschlossenen MFP nur durch bekannte I/O-Kanäle kommunizieren. Ein beliebiges MFP, das andere I/O-Kanäle verwendet, würde bei diesen Servern nicht funktionieren, ohne zuerst durch den Benutzer aktualisiert zu werden, was unpraktisch ist.

[0004] Die EP 0859321 offenbart eine Mehrfunktionsperipheriegerätsteuerung, die durch einen Kommunikationskanal mit einem einzelnen Mehrfunktionsperipheriegerät gekoppelt ist. Jede der Funktionen des Mehrfunktionsperipheriegeräts ist separat durch die Steuerung unter Verwendung einer eindeutigen logischen Einheitsnummer adressierbar.

[0005] Die EP 0843440 offenbart eine Netzwerkvorrichtung, die eine Liste von Vorrichtungsadressen unterhält. Es ist ein System zum Liefern einer Netzwerkvorrichtung in einem lokalen Netz (LAN) beschrieben, die als eine Listenverwaltungseinrichtung wirksam ist, die eine Liste von Vorrichtungsadressen für das LAN unterhält.

[0006] Die vorliegende Erfindung liefert einen verbesserten Netzwerkperipheriegerätserver, der z. B. einen Zugriff auf eine Mehrzahl von Funktionen liefern kann, die durch ein Mehrfunktionsperipheriegerät unterstützt werden, das mit einem Netzwerk verbunden ist.

[0007] Gemäß einem Aspekt der vorliegenden Erfindung wird ein Netzwerkperipheriegerätserver geliefert, wie derselbe in Anspruch 1 spezifiziert ist.

[0008] Gemäß einem weiteren Aspekt der vorliegenden Erfindung wird ein Verfahren zum Ermöglichen für eine Mehrzahl von Client-Vorrichtungen in einem Netzwerk geliefert, wie dasselbe in Anspruch 9 spezifiziert ist.

[0009] Weitere Ausführungsbeispiele der Erfindung sind so, wie sie in den abhängigen Ansprüchen dargelegt sind.

[0010] Der bevorzugte Server kann einen Zugriff auf eine Mehrzahl von Funktionen des Mehrfunktionsperipheriegeräts zusätzlich zu der Druck- und Scanfunktion liefern.

[0011] Bei einem Ausführungsbeispiel wird ein Netzwerkperipheriegerätserver geliefert, der einen Universalgateway (bzw. Netzübergang) umfasst, der einen Datendurchgang für eine Mehrzahl von Funktionen des Mehrfunktionsperipheriegeräts liefert, wodurch die Notwendigkeit beseitigt wird, einen separaten Gateway zu erzeugen, der jeder Funktion entspricht, auf die zugegriffen wird.

[0012] Vorteilhafterweise liefert der Server einen Netzwerkzugriff auf die Funktionen des Mehrfunktionsperipheriegeräts an jedem beliebigen Kanal, der durch das Mehrfunktionsperipheriegerät unterstützt wird.

[0013] Ein Ausführungsbeispiel der vorliegenden Erfindung ist im Folgenden nur als Beispiel unter Bezugnahme auf die beiliegenden Zeichnungen beschrieben. Es zeigen:

[0014] [Fig. 1](#) ein Blockdiagramm, das ein Ausführungsbeispiel eines Netzwerksystems zeigt;

[0015] [Fig. 2](#) ein logisches Blockdiagramm eines bevorzugten Servers;

[0016] [Fig. 3](#) ein Flussdiagramm, das die Operationen einer Client-Vorrichtung zum Zugreifen auf die Funktionen des Mehrfunktionsperipheriegerätserver veranschaulicht; und

[0017] [Fig. 4A](#) bis [Fig. 4E](#) Flussdiagramme, die die Operationen des Servers von [Fig. 2](#) veranschaulichen.

[0018] Grob gesagt betrifft das beschriebene Ausführungsbeispiel einen Netzwerkperipheriegerätservver, der angepasst ist, um es einer Mehrzahl von Client-Vorrichtungen, die in einem Netzwerk verbunden sind, zu ermöglichen, auf alle Funktionen oder Dienste zuzugreifen, die durch ein Mehrfunktionsperipheriegerät geboten werden, das ebenfalls in dem gleichen Netzwerk verbunden ist. Der Server erhält von dem Peripheriegerät die Adresse und den entsprechenden Peripheriegerätkommunikationskanal einer bestimmten Funktion, die durch einen Client angefordert wird, und verbindet den anfordernden Client durch diesen Peripheriegerätkanal mit dem Peripheriegerät, so dass die gewünschte Funktion dem Client verfügbar ist.

[0019] Der Server umfasst eine Netzwerkschnittstelle zum Kommunizieren mit den Clients in dem Netzwerk gemäß einem vorbestimmten Netzwerkprotokoll, eine Peripheriegerätschnittstelle zum Kommunizieren mit dem Mehrfunktionsperipheriegerät über eine Mehrzahl von Peripheriegerätkanälen, die der Mehrzahl von Funktionen entsprechen, die durch das Mehrfunktionsperipheriegerät unterstützt werden, und einen Gateway, der kommunikativ zwischen die Netzwerkschnittstelle und die Peripheriegerätschnittstelle geschaltet ist, zum Übertragen von Daten zwischen der Netzwerkschnittstelle und der Peripheriegerätschnittstelle. Ebenfalls in dem Server bereitgestellt ist Firmware und/oder Software, die auf vorbestimmte Anweisungen von einem oder mehr der Mehrzahl von Clients anspricht, zum wirksamen Verbinden des Clients mit ein oder mehr ausgewählten Funktionen der Mehrzahl von Funktionen, die durch das Mehrfunktionsperipheriegerät unterstützt werden, über einen entsprechenden Kanal der Mehrzahl von Peripheriegerätkanälen.

[0020] Die beschriebenen Ausführungsbeispiele liefern auch ein Verfahren zum Ermöglichen, dass eine Mehrzahl von Clients, die über einen Peripheriegerätservver mit einem Netzwerk verbunden sind, auf eine Mehrzahl von Funktionen zugreift, die durch ein Mehrfunktionsperipheriegerät unterstützt werden, das mit dem Netzwerk verbunden ist. Das Verfahren umfasst ein Senden einer Dienstnamennachschlaganweisung von einem beliebigen der Mehrzahl von Clients zum Anfordern eines Peripheriegerätkanals des Servers, der der ausgewählten Funktion des Mehrfunktionsperipheriegeräts entspricht, ein Öffnen des entsprechenden Peripheriegerätkanals, um den Client kommunikativ mit der ausgewählten Funktion zu verbinden, und ein Übertragen von Daten zwischen dem Client und dem Mehrfunktionsperipheriegerät durch den Peripheriegerätservver.

[0021] Unter jetziger Bezugnahme auf [Fig. 1](#) ist ein Server **10** angepasst, um mit einem Netzwerk **12** zusammen mit einer Mehrzahl von Client-Vorrichtungen **14** (zwei sind gezeigt) verbunden zu sein. Der Server

10 ist auch mit einem Mehrfunktionsperipheriegerät (MFP) **16** verbunden. Wirksam ist das MFP **16** über den Server **10** mit dem Netzwerk **12** verbunden.

[0022] Unter jetziger Zuwendung zu [Fig. 2](#) umfasst der Server **10** eine Netzwerkschnittstelle **18**, die bevorzugt zumindest einen TCP-Port und/oder zumindest ein SPX-Socket zur Verbindung mit dem Netzwerk **12** unter Verwendung eines herkömmlichen TCP-Protokolls bzw. eines SPX-Protokolls umfasst. Ebenfalls enthalten ist eine Peripheriegerätschnittstelle **20**, bei der es sich bevorzugt um eine bekannte BIO-Schnittstelle handelt, obwohl andere bekannte Schnittstellen, wie z. B. EIO und MIO, ebenfalls in Betracht gezogen werden. Die Schnittstelle **20** ist angepasst, um den Server **10** mit dem MFP **16** zu verbinden und zu ermöglichen, dass der Client **14** auf jede der Funktionen, die durch das MFP unterstützt werden, über eine Mehrzahl von entsprechenden Peripheriegerätkanälen **22** zugreift, bei denen es sich bevorzugt um IEEE-1284.4-Kanäle handelt. Kommunikationsverbindungen, wie z. B. PCI, eine Seriellverbindung, wie z. B. USB, oder andere bekannte Verbindungen werden ebenfalls in Betracht gezogen.

[0023] Ein Universalgateway (GGW) **24** ist zwischen die Netzwerkschnittstelle **18** und die Peripheriegerätschnittstelle **20** geschaltet und liefert einen bidirektionalen Datendurchgang zwischen der Netzwerkschnittstelle und der Peripheriegerätschnittstelle für die gewünschten Funktionen. Der GGW **24** ermöglicht auch, dass der Client **14** einen spezifischen Peripheriegerätkanal **22** anfordert, der einer bestimmten Funktion des MFP **16** entspricht, wodurch die Notwendigkeit beseitigt wird, einen neuen Gateway für jede der gewünschten Funktionen einzurichten.

[0024] Unter jetziger Zuwendung zu [Fig. 3](#) ist die Operation eines Softwareprogramms in der Client-Vorrichtung **14** veranschaulicht, die es ermöglicht, dass der Client auf eine bestimmte MFP-Funktion zugreift. Anfangs, wenn ein Benutzer an einem vernetzten Client einen Zugriff auf eine bestimmte Funktion des MFP **16** anfordert (Block **26**), wird bestimmt, ob das vorgesehene Peripheriegerät mit dem Netzwerk **12** oder direkt mit dem Client **14** verbunden ist (Block **28**). Falls das MFP **16** nicht vernetzt ist, endet der Prozess (Block **30**), und das MFP wird auf eine bekannte Weise durch den Computer betrieben, mit dem das MFP verbunden ist.

[0025] Falls das MFP **16** vernetzt ist, wird eine Netzwerkverbindung zwischen dem Client **14** und dem Server **10** eingerichtet (Block **31**). Der Server **10** befindet sich anfangs in einem Befehlsmodus und horcht auf Befehle von dem Client **14** an den ein oder mehr TCP-Ports und den ein oder mehr SPX-Sockets. Wenn ein Befehl von dem Client **14** empfangen wird, gibt der Server **10** eine Antwort aus, bevor-

zugt mit einer dreistelligen Ergebniscodezahl, die der FTP- und der SMTP-Übereinkunft folgt. Zum Beispiel ist eine beliebige Zahl, die mit einer „2“ beginnt, ein erfolgreiches Ergebnis, und diejenige mit einer „4“ oder einer „5“ ist ein nicht erfolgreiches Ergebnis. Die restlichen Stellen geben weitere Details über das Ergebnis an.

[0026] Wenn die Netzwerkverbindung erfolgreich eingerichtet worden ist (Block 32), sendet der Client 14 einen ZEIT-Befehl (Block 34), der den Leerlaufzeitablauf der eingerichteten Verbindung in Sekunden festlegt. Falls während der festgelegten Sekunden keine Daten zwischen dem Client 14 und dem MFP 16 ausgetauscht werden, schließt der Server 10 die Verbindung zwischen denselben. Nachdem der Leerlaufzeitablauf festgelegt worden ist (Block 36), sendet der Client 14 einen SERV-Befehl an den Server 10, um ein Peripheriegerätdienstnamennachschatzen anzufordern (Block 38), d. h. um Informationen über das MFP 16 zu sammeln und eine Verbindung zwischen dem Client und dem MFP für eine angeforderte Funktion oder einen Dienst zu konfigurieren, die bzw. der durch das MFP unterstützt wird. Wenn der SERV-Befehl ausgeführt worden ist (Block 40), gibt der Client 14 einen ÖFFNEN-Befehl an den Server 10 aus, um den Peripheriegerätkanal zu öffnen, der der angeforderten Funktion des MFP entspricht (Block 42). Wenn dieser Kanal geöffnet worden ist (Block 44), gibt der Client 14 einen DATEN-Befehl aus (Block 46), der anfordert, dass dieser Server 10 in einen Datendurchgangsmodus übergeht. Wenn derselbe in diesen Modus versetzt worden ist (Block 48), kommuniziert der Client 14 direkt mit dem MFP 16, um auf die gewünschten Funktionen zuzugreifen, und der GGW 24 des Servers 10 leitet lediglich Daten zwischen dem Client und dem MFP (Block 50), bis die Verbindung geschlossen wird (Block 30). Es sei darauf hingewiesen, dass, wenn sich derselbe in dem Durchgangsmodus befindet, der Client 14 die Verbindung schließen muss, um zu beenden, dass Daten durch den GGW 24 übertragen werden. In dem Fall jedoch, dass das MFP 16 den Kanal schließt, schließt der Server 10 die Verbindung. Falls bei irgendeinem der Blöcke 28, 32, 36, 40, 44 und 48 der Server 10 eine Antwort ausgibt, die ein nicht erfolgreiches Ergebnis anzeigt, d. h. eine Zahl sendet, die mit einer „4“ oder einer „5“ beginnt, wird dem Benutzer eine Fehlernachricht angezeigt (Block 52), und der Prozess wird beendet (Block 30).

[0027] Unter jetziger Zuwendung zu den [Fig. 4A](#) – [Fig. 4E](#) ist die Operation des Servers 10, der auf Befehle anspricht, die von dem Client 14 empfangen werden, veranschaulicht. Der Server 10 ist ereignisgetrieben und wartet somit auf ein Ereignis, d. h. einen Befehl oder eine Antwort, von dem Client 14 oder dem MFP 16 nach einem Initialisieren (Block 54). Wenn ein Ereignis auftritt, wird der Leerlaufzeitablauf durch den Server 10 rückgesetzt, der anfangs durch

den Client 14 festgelegt wurde (Block 56), und der Server bestimmt, welches Ereignis aufgetreten ist (Block 58). Der Server 10 schließt dann jede Ereignisprozedur ab, bevor das nächste Ereignis gehandhabt wird.

[0028] Wenn eine Neuverbindungsanforderung von dem Client 14 empfangen wird (Block 60), bestimmt der Server 10, ob derselbe die Neuverbindungsanforderung akzeptieren kann (Block 62). Bei dem bevorzugten Ausführungsbeispiel ist der Server 10 in der Lage, zwei gleichzeitige Verbindungen zu unterstützen, obwohl mehr gleichzeitige Verbindungen in Betracht gezogen werden. Falls die maximale Anzahl von Verbindungen bereits eingerichtet ist, sendet der Server eine „421“-Fehlernachricht an den Client (Block 64), schließt die Netzwerkverbindung (Block 66) und wartet darauf, dass ein weiteres Ereignis eintritt (Block 54). Falls der Server 10 die Verbindung akzeptieren kann, setzt derselbe den Verbindungszustand auf BEFEHL (Block 68), sendet eine „200“-Erfolgsnachricht an den Client 14 (Block 70) und wartet auf ein Ereignis (Block 54).

[0029] Wenn Daten von dem Client 14 empfangen werden (Block 72), bestimmt der Server 10, ob der Client Daten an das MFP 16 sendet, d. h. der Client befindet sich in einem DATEN-Modus, oder einen Befehl ausgibt, d. h. der Client befindet sich in einem BEFEHL-Modus (Block 74). Falls sich der Client 14 in dem DATEN-Modus befindet, prüft der Server 10, ob der Kanal 22, der in den Daten angefordert wird, zu dem MFP 16 offen ist (Block 76). Ist dies der Fall, werden die Daten über diesen Kanal 22 an das MFP 16 gesendet (Block 78). Ist dies nicht der Fall, werden die Daten in eine Warteschlange gestellt, bis dieser Kanal 22 offen ist (Block 80). Nach dem Abschluss jeder der beiden Prozeduren wartet der Server 10 auf ein weiteres Ereignis (Block 54). Andererseits ruft der Server 10, falls sich der Client 14 in dem BEFEHL-Modus befindet, die geeigneten Befehlsverarbeitungsprozeduren auf, die in [Fig. 4B](#) gezeigt sind.

[0030] Unter jetziger Zuwendung zu [Fig. 4B](#) muss der Server 10, um einen Befehl zu verarbeiten, zuerst warten, bis der Client 14 eine vollständige Textzeile gesendet hat (Block 82), da eine Zeile in Stücken über das Netzwerk 12 übertragen werden kann. Wie es in der Technik bekannt ist, ist eine Textzeile eine beliebige Sequenz von Schriftzeichen, die bevorzugt mit einem optionalen Wagenrücklauf- (CR-) Schriftzeichen (ASCII-Code 13) gefolgt von einem vorgeschriebenen Zeilenvorschubs- (LF-) Schriftzeichen (ASCII-Code 10) endet. Der Server 10 sammelt die Schriftzeichen der Zeile an, bis derselbe die Ende-der-Zeile-Sequenz CR-LF oder nur LF empfängt (Block 84).

[0031] Wenn derselbe über eine volle Zeile verfügt, prüft der Server 10 die ersten vier Schriftzeichen auf

einen erkannten Befehl (Block 86). Falls ein ZEIT-Befehl empfangen wird (Block 88), analysiert der Server 10 syntaktisch ein Zahlenargument, das dem Befehl folgt (Block 90), und legt den neuen Leerlaufzeitablauf auf diesen Wert fest (Block 92). Bei dem bevorzugten Ausführungsbeispiel zeigt der Wert Null einen unendlichen Zeitablauf an, der niemals abläuft, und der Voreinstellungszeitablauf beträgt 90 Sekunden. Dann sendet der Server 10 eine „200“-Erfolgsnachricht an den Client 14 (Block 94) und wartet auf ein weiteres Ereignis (Block 54).

[0032] Bei einem MFP 16, das spezielle Anforderungen bezüglich der Anzahl von Datenpaket-„Krediten“ aufweist, die dasselbe von dem Server 10 benötigt, um es demselben zu ermöglichen, Daten zurück an den Server zu senden, wie es in der Technik bekannt ist, sendet der Client 14 einen MPCT-Befehl in einem Zahlenargument an den Server 10, um die minimale Anzahl von Paketkrediten festzulegen, die durch das MFP benötigt werden (Block 96). Falls der Zustand des angeforderten Kanals nicht LEERLAUF ist (Block 98), sendet der Server 10 eine „452“-Fehlernachricht an den Client 14 (Block 100). Ansonsten analysiert der Server 10 syntaktisch das Zahlenargument (Block 102), speichert dasselbe zur späteren Verwendung, wenn der Client einen ÖFFNEN-Befehl ausgibt (Block 104), und sendet eine „200“-Erfolgsnachricht (Block 106).

[0033] Falls ein ÖFFNEN-Befehl für einen bestimmten Kanal 22 empfangen wird (Block 108), prüft der Server 10, ob sich dieser Kanal in einem LEERLAUF-Zustand befindet (Block 110). Falls der Kanalzustand nicht LEERLAUF ist, sendet der Server 10 eine „452“-Fehlernachricht (Block 112). Ansonsten analysiert der Server 10 syntaktisch das Zahlenargument, das dem ÖFFNEN-Befehl folgt (Block 114), setzt den Kanalzustand auf ÖFFNEN-WARTEN (Block 116) und sendet eine Kanal-Öffnen-Anforderung an das MFP 16 (Block 118). Das MFP 16 ist bezüglich des Servers 10 asynchron wirksam, so dass der Server das Ergebnis der Kanal-Öffnen-Anforderung nicht sofort weiß. Eine Antwort von dem MFP 16 kommt im Allgemeinen bald an und wird als ein separates Ereignis verarbeitet, wie es im Folgenden bei Block 176 in [Fig. 4D](#) beschrieben ist. Bis zu diesem Zeitpunkt wird keine Ergebnisnachricht an den Client 14 gesendet.

[0034] Wenn ein DATEN-Befehl zum Anfordern, dass der Server 10 in einen Datendurchgangsmodus übergeht, von dem Client 14 empfangen wird (Block 120), prüft der Server, ob der Kanal 22, der durch den ÖFFNEN-Befehl angefordert wird, sich entweder in einem OFFEN-Zustand oder in einem ÖFFNEN-WARTEN-Zustand befindet (Block 122). Falls der Kanalzustand OFFEN oder ÖFFNEN-WARTEN ist, setzt der Server 10 den Client-Zustand auf DATEN (Block 124), sendet eine „200“-Erfolgsnachricht

(Block 126) und sendet dann jegliche MFP-Daten, die sich in dem Server in einer Warteschlange befinden (Block 128) (siehe Blöcke 160–166 in [Fig. 4A](#), die im Folgenden beschrieben ist, für ein Antworten auf Daten, die von dem MFP 16 zu dem Client 14 empfangen werden). Ansonsten sendet der Server 10 eine „453“-Fehlernachricht und wartet auf ein weiteres Ereignis (Block 54).

[0035] Unter jetziger Zuwendung zu [Fig. 4C](#) analysiert, wenn ein SERV-Befehl von dem Client 14 empfangen wird (Block 132), der ein Dienstnamennachschatzen für einen bestimmten Dienst anfordert, der durch das MFP 16 unterstützt wird, der Server 10 syntaktisch das Wort, das dem SERV-Befehl folgt (Block 134), und sendet eine Dienstnamennachschatzgeanforderung an das MFP 14 mit diesem Wort als dem Namen, der nachgeschlagen werden soll (Block 136). Der Server 10 wartet dann auf ein Ereignis (Block 54), da eine Antwort später als ein separates Ereignis ankommt, wie es im Folgenden beschrieben und bei Blöcken 168–174 von [Fig. 4D](#) veranschaulicht ist. Fachleute werden erkennen, dass ein Dienstnamennachschatzen ein Merkmal vieler Kommunikationsprotokolle ist, die zwischen einem MFP-Server, wie z. B. dem HP JetDirect, und einem MFP verwendet werden. Bei dem bevorzugten Ausführungsbeispiel wird das IEEE-1284.4-Protokoll für Parallelports durch das MFP 14 verwendet. Bei der Kommunikationsverbindung kann es sich jedoch auch um eine Backplane, wie z. B. PCI, eine Seriellverbindung, wie z. B. USB, oder viele andere handeln. Der Befehl verwendet das Dienstnamennachschatzgeprotokoll, das durch die Kommunikationsverbindung mit dem MFP 16 unterstützt wird. Falls das Kommunikationsprotokoll kein Dienstnamennachschatzgeprotokoll unterstützt (Block 138), sendet der Server 10 sofort eine „451“-Fehlernachricht an den Client 14 zurück und sendet keine Anforderung an das MFP 14 (Block 140).

[0036] Ähnlich wie bei dem SERV-Befehl analysiert, wenn ein SKID-Befehl empfangen wird (Block 142), der Server 10 das Wort syntaktisch, das dem Befehl zum Spezifizieren der „Socket-ID“ folgt (Block 144), und sendet eine Socket-ID-Nachschatzgeanforderung an das MFP 16 mit dieser Zahl als dem Socket, das nachgeschlagen werden soll (Block 146), jedoch nur falls das Kommunikationsprotokoll die Socket-ID-Nachschatzgeanforderung unterstützt (Block 148). Ansonsten sendet der Server 10 sofort eine „451“-Fehlernachricht an den Client 14 zurück und sendet keine Anforderung an das MFP 14 (Block 150). Die Antwort kommt später als ein separates Ereignis an, wie es im Folgenden beschrieben und bei den Blöcken 168 bis 174 von [Fig. 4D](#) veranschaulicht ist.

[0037] Wenn ein BEENDEN-Befehl empfangen wird (Block 152), sendet der Server eine „200“-Erfolgs-

nachricht an den Client **14** (Block **154**) und ruft dann die Schließprozedur auf, die im Folgenden beschrieben und in [Fig. 4E](#) veranschaulicht ist. Falls der empfangene Befehl nicht erkannt wird (Block **156**), sendet der Server **10** eine „500“-Fehlernachricht an den Client (Block **158**) und ignoriert den Befehl. Dann wartet der Server **10** auf ein weiteres Ereignis (Block **54**).

[0038] Unter jetziger erneuter Zuwendung zu [Fig. 4A](#) leitet, wenn der Server **10** Daten von dem MFP **16** empfängt, während derselben auf ein Ereignis wartet (Block **160**), derselbe einfach die MFP-Daten an den Client **14** weiter (Block **164**), falls sich der Client in dem DATEN-Modus befindet (Block **162**). Falls sich der Client **14** in dem BEFEHL-Modus befindet (Block **162**), stellt der Server **10** die Daten in eine Warteschlange, bis der Client einen DATEN-Befehl ausgibt (Block **166**).

[0039] Unter jetziger Zuwendung zu [Fig. 4D](#) extrahiert, wenn der Server **10** eine Antwort von dem MFP **14** ansprechend auf entweder den Dienstnamen-nachschlage- oder den Socket-ID-Nachschlagebefehl empfängt (Block **168**), die im Vorhergehenden bei den Blöcken **138–150** von [Fig. 4C](#) beschrieben sind, der Server **10** die zurückgesendete Socket-ID bzw. den Dienstnamen und sendet die- bzw. denselben in einer „250“-Nachricht an den Client **14** (Block **172**), falls das MFP ein erfolgreiches Nachschlagen zurücksendet (Block **170**). Ansonsten sendet der Server **10** eine „451“-Fehlernachricht an den Client **14** (Block **174**). Auf diese Weise wird die Socket-ID des gewünschten Dienstes von dem Client **14** erhalten, der dann einen Kommunikationskanal zu dieser Socket-ID zum Verwenden des gewünschten Dienstes öffnen kann. Umgekehrt erhält der Client **14** den Namen des Dienstes, der einer bestimmten Socket-ID zugeordnet ist, von dem Dienstnamen, der durch das MFP **14** zurückgesendet wird.

[0040] Wenn der Server **10** eine Antwort von dem MFP **16** ansprechend auf einen ÖFFNEN-Befehl von dem Client **14** empfängt, wie es im Vorhergehenden bei den Blöcken **108–118** in [Fig. 4B](#) beschrieben ist (Block **176**), prüft der Server **10** das Ergebnis der Anforderung (Block **178**). Falls dasselbe erfolgreich war, prüft der Server **10** dann den Kanalzustand (Block **180**). Falls der Zustand ÖFFNEN-WARTEN ist, setzt der Server **10** den Zustand auf OFFEN (Block **182**), sendet jegliche Daten, die in dem Server in eine Warteschlange gestellt wurden, der auf den OFFEN-Zustand wartet, an das MFP **16** (Block **184**) (im Vorhergehenden bei den Blöcken **72–76** und **80** in [Fig. 4A](#) beschrieben) und sendet eine „200“-Erfolgsnachricht an den Client **14**, um auf den ÖFFNEN-Befehl zu antworten. Falls der Kanalzustand ÖFFNEN-WARTEN-SCHLIEßen ist, wie es im Folgenden bei Block **218** von [Fig. 4E](#) beschrieben ist (Block **180**), setzt der Server **10** den Zustand auf OFFEN und ruft dann sofort die Verbindung-Schließen-Prozedur auf, die im

Folgenden beschrieben und in [Fig. 4E](#) gezeigt ist (Block **188**). Falls die Kanal-Öffnen-Anforderung nicht erfolgreich war (Block **178**), prüft der Server **10** den Kanalzustand (Block **190**). Falls der Zustand ÖFFNEN-WARTEN ist, setzt der Server **10** den Zustand auf LEERLAUF (Block **192**) und sendet eine „450“-Fehlernachricht an den Client **14** (Block **194**). Falls der Zustand ÖFFNEN-WARTEN-SCHLIEßen ist, setzt der Server **10** den Zustand einfach auf LEERLAUF (Block **196**).

[0041] Wenn der Server **10** eine Kanalschließantwort von dem MFP **16** empfängt (Block **198**), setzt der Server den Kanalzustand sofort auf LEERLAUF (Block **200**). Falls die Kommunikationsverbindung zu dem MFP **16** aus irgendeinem Grund versagt (Block **202**), wobei z. B. das Kommunikationskabel getrennt ist oder das MFP heruntergefahren ist, schließt der Server **10** sofort alle Netzwerkverbindungen (Block **204**) und spült jegliche sich in einer Warteschlange befindliche Daten aus (Block **206**).

[0042] Unter jetziger Zuwendung zu [Fig. 4E](#) leitet der Server **10** eine Prozedur zum Schließen der Verbindung mit dem MFP **16** immer dann ein, wenn ein BEENDEN-Befehl durch den Client ausgegeben wird, wie bei Block **152** im Vorhergehenden bei [Fig. 4C](#), immer wenn der Server ein Schließen aufgrund eines Fehlers oder irgendeiner anderen nicht-behebbaren Bedingung einleiten möchte, oder wenn der Leerlaufzeitablauf abläuft (Block **208**). Um eine Verbindung zu schließen, prüft der Server **10** zuerst den Zustand des Kommunikationskanals **22** zu dem MFP **16** (Block **210**). Falls derselbe OFFEN ist, setzt der Server **10** denselben auf SCHLIEßen-WARTEN (Block **212**) und sendet eine Kanal-Schließen-Anforderung an das MFP **16** (Block **214**). Falls der Zustand ÖFFNEN-WARTEN ist (Block **216**), setzt der Server **10** denselben auf einen ÖFFNEN-WARTEN-SCHLIEßen-Zustand (Block **218**), was dazu führt, dass eine Schließen-Anforderung gesendet wird, sobald eine Öffnen-Antwort empfangen wird (siehe die obige Beschreibung der Blöcke **176–180** und **188** von [Fig. 4D](#)). Keine Kanalaktionen werden bei irgendeinem anderen Zustand benötigt (Block **220**). Nachdem der Kanalzustand erledigt ist, schließt der Server **10** die Netzwerkverbindung oder beendet das Schließen derselben (Block **220**). Schließlich löscht der Server **10** jegliche Daten, die für entweder das MFP **16** oder den Netzwerk-Client **14** in einer Warteschlange sind (Blöcke **222**, **224**), und wartet auf ein weiteres Ereignis (Block **54**).

[0043] Aus der vorhergehenden Beschreibung ist es ersichtlich, dass ein verbesserter Peripheriegeräts-server gezeigt und beschrieben wurde, der viele erwünschte Attribute und Vorteile aufweist. Derselbe ist konzipiert, um es einer Mehrzahl von Clients in einem Netzwerk zu ermöglichen, auf eine Mehrzahl von Funktionen zuzugreifen, die durch ein Mehrfunktions-

peripheriegerät unterstützt werden, das mit dem Netzwerk verbunden ist. Der Server kann es ermöglichen, dass der Client zumindest eine Funktion auswählt, wobei dann veranlasst wird, dass dieselbe für den Client durch einen Peripheriegerätkommunikationskanal zugänglich ist, der der Adresse der ausgewählten Funktion entspricht. Ein einziger Gateway wird bevorzugt verwendet, um den Client mit einem beliebigen der ausgewählten Kommunikationskanäle zu verbinden.

Patentansprüche

1. Ein Netzwerkperipheriegerätservier, der konzipiert ist, um es einer Mehrzahl von Client-Vorrichtungen (14) in einem Netzwerk (12) zu ermöglichen, auf eine Mehrzahl von Funktionen zuzugreifen, die durch zumindest ein Mehrfunktionsperipheriegerät (16) unterstützt werden, das mit dem Netzwerk verbunden ist, wobei der Server folgende Merkmale aufweist: eine Netzwerkschnittstelle (18) zum Kommunizieren mit Client-Vorrichtungen (14) in dem Netzwerk (12) gemäß einem vorbestimmten Netzwerkprotokoll; eine Peripheriegerätschnittstelle (20) zum Kommunizieren mit dem Mehrfunktionsperipheriegerät (16) über eine Mehrzahl von Peripheriegerätkanälen (22), die der Mehrzahl von Funktionen entsprechen, die durch das Mehrfunktionsperipheriegerät unterstützt werden; einen Gateway (24), der kommunikativ zwischen die Netzwerkschnittstelle (18) und die Peripheriegerätschnittstelle (20) geschaltet ist, zum Übertragen von Daten zwischen der Netzwerkschnittstelle und der Peripheriegerätschnittstelle; und eine Steuereinrichtung (10), die auf Anweisungen von zumindest einer der Mehrzahl von Client-Vorrichtungen (14) anspricht, um Informationen über das Mehrfunktionsperipheriegerät (16) zu sammeln und um eine Verbindung zwischen der zumindest einen Client-Vorrichtung (14) und dem Mehrfunktionsperipheriegerät (16) für eine ausgewählte Peripheriegerätfunktion, die durch das Mehrfunktionsperipheriegerät (16) unterstützt wird, über einen der Mehrzahl von Peripheriegerätkanälen (22) zu konfigurieren, der der ausgewählten Peripheriegerätfunktion entspricht.

2. Ein Server gemäß Anspruch 1, bei dem die Anweisungen eine Anweisung zum Anfordern einer Adresse der zumindest einen ausgewählten Funktion umfassen, und die Steuereinrichtung (10) wirksam ist, um die Adresse von dem Mehrfunktionsperipheriegerät (16) zu erhalten und um die Adresse an eine Client-Vorrichtung (14) zu liefern, die die Adresse angefordert hat.

3. Ein Server gemäß Anspruch 2, bei dem die Anweisungen eine Anweisung zum Öffnen eines Peripheriegerätkanals (22), der der Adresse entspricht, umfassen, und die Steuereinrichtung (10) auf die An-

weisung zum Öffnen des Peripheriegerätkanals anspricht, um die Client-Vorrichtung (14) kommunikativ mit dem Peripheriegerätkanal zu verbinden.

4. Ein Server gemäß Anspruch 3, bei dem die Steuereinrichtung (10) wirksam ist, um Daten, die von der Client-Vorrichtung (14) empfangen werden, durch den Peripheriegerätkanal (22) an das Mehrfunktionsperipheriegerät (16) zu senden, wenn der Peripheriegerätkanal offen ist, und um die Daten in dem Server in eine Warteschlange zu stellen, wenn der Peripheriegerätkanal nicht offen ist.

5. Ein Server gemäß einem der vorhergehenden Ansprüche, bei dem die Anweisungen eine Anweisung zum Öffnen eines Peripheriegerätkanals (22), der einer Adresse der ausgewählten Funktion entspricht, umfassen, und die Steuereinrichtung (10) auf die Anweisung zum Öffnen des Peripheriegerätkanals anspricht, um eine Client-Vorrichtung (12), die die Anweisung ausgegeben hat, kommunikativ über den Gateway mit dem Peripheriegerätkanal zu verbinden.

6. Ein Server gemäß einem der vorhergehenden Ansprüche, bei dem die Anweisungen eine Anweisung zum Anfordern eines Typs einer Funktion umfassen, der einem ausgewählten der Mehrzahl von Peripheriegerätkanälen (22) entspricht, und die Steuereinrichtung (10) auf die Anweisung anspricht, um den Typ der Funktion von dem Mehrfunktionsperipheriegerät zu erhalten, um den Typ der Funktion an eine Client-Vorrichtung (14) zu liefern, die den Typ der Funktion angefordert hat.

7. Ein Server gemäß einem der vorhergehenden Ansprüche, bei dem die Steuereinrichtung (10) konzipiert ist, um gleichzeitig zumindest zwei Funktionen des Mehrfunktionsperipheriegeräts (16) mit der zumindest einen Client-Vorrichtung (14) zu verbinden.

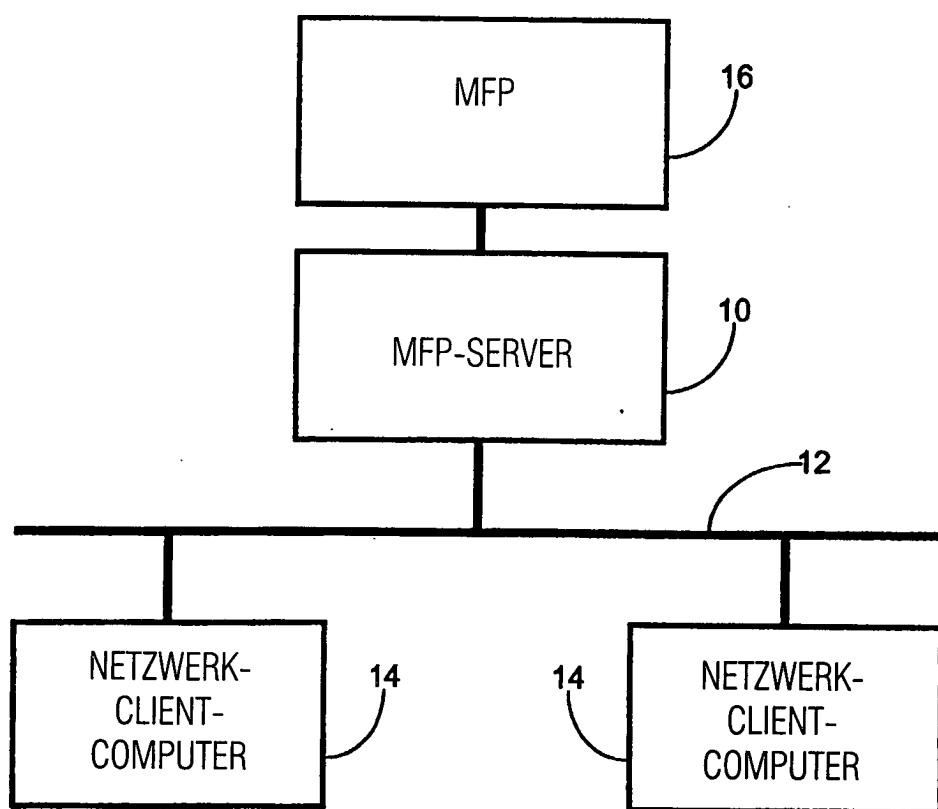
8. Ein Server gemäß einem der vorhergehenden Ansprüche, bei dem der Gateway (24) kommunikativ zwischen dem Netzwerk (12) und der Peripheriegerätschnittstelle (20) für jedes Mehrfunktionsperipheriegerät (16) eingerichtet wird, wenn eine Mehrzahl der Mehrfunktionsperipheriegeräte mit dem Netzwerk verbunden ist.

9. Ein Verfahren zum Ermöglichen, dass eine Mehrzahl von Client-Vorrichtungen (14), die über einen Peripheriegerätservier (10) mit einem Netzwerk (12) verbunden sind, auf eine Mehrzahl von Funktionen zugreift, die durch ein Mehrfunktionsperipheriegerät (16) unterstützt werden, das mit dem Netzwerk verbunden ist, wobei das Verfahren folgende Schritte aufweist Senden einer Anweisung von zumindest einer der Mehrzahl von Client-Vorrichtungen zum Anfordern eines Peripheriegerätkanals des Servers, der zumindest einer ausgewählten Funktion des Mehr-

funktionsperipheriegeräts (16) entspricht;
ansprechend auf die Anweisung, Sammeln von Informationen über das Mehrfunktionsperipheriegerät (16) und Konfigurieren einer Verbindung zwischen der zumindest einen Client-Vorrichtung (14) und dem Mehrfunktionsperipheriegerät (16), falls die zumindest eine ausgewählte Funktion durch das Mehrfunktionsperipheriegerät (16) unterstützt wird;
Öffnen des entsprechenden Peripheriegerätkanals, um die zumindest eine Client-Vorrichtung (14) kommunikativ mit der zumindest einen ausgewählten Funktion (42) zu verbinden, falls die ausgewählte Funktion durch das Mehrfunktionsperipheriegerät (16) unterstützt wird; und
Übertragen von Daten zwischen der zumindest einen Client-Vorrichtung und dem Mehrfunktionsperipheriegerät durch den Peripheriegerätserver (50).

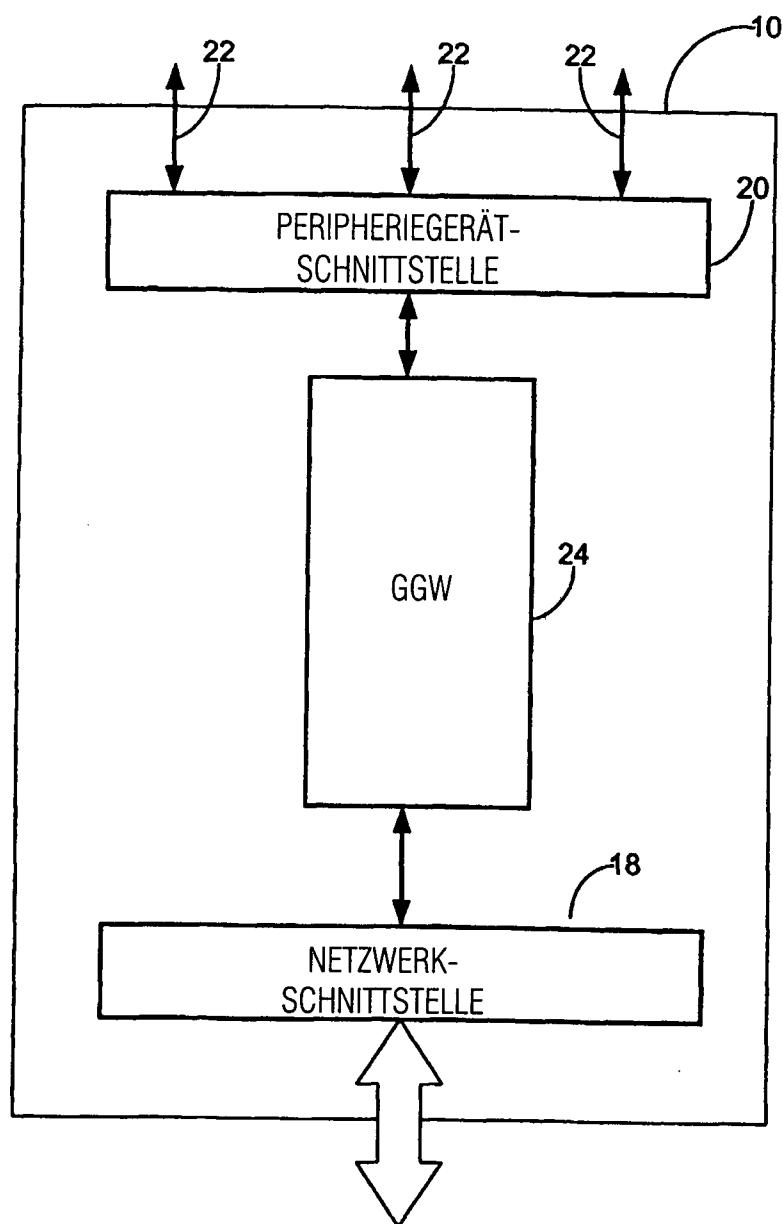
10. Ein Verfahren gemäß Anspruch 9, bei dem die Daten durch einen einzigen Gateway (24) in dem Peripheriegerätserver (10) übertragen werden.

Es folgen 8 Blatt Zeichnungen

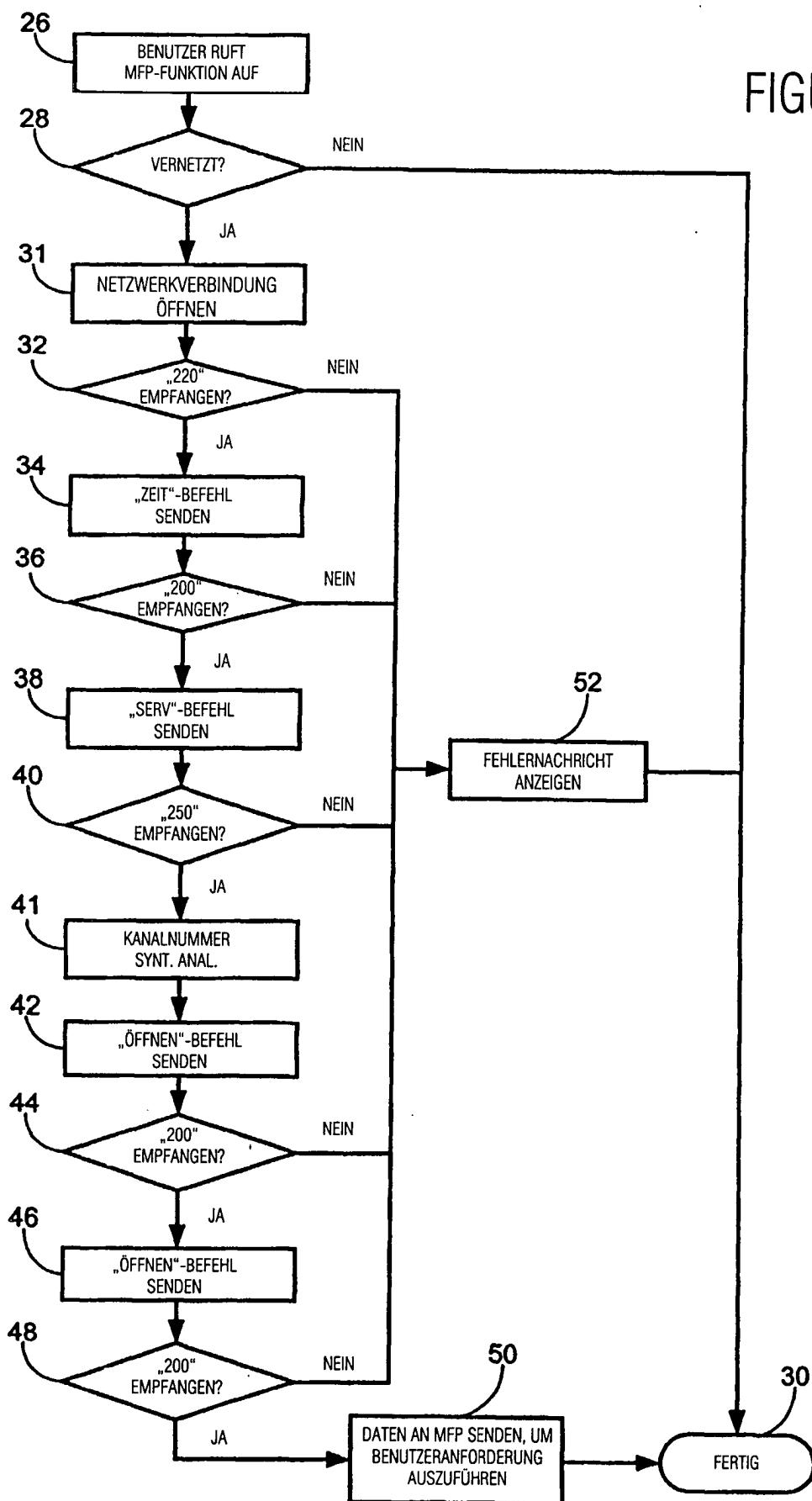


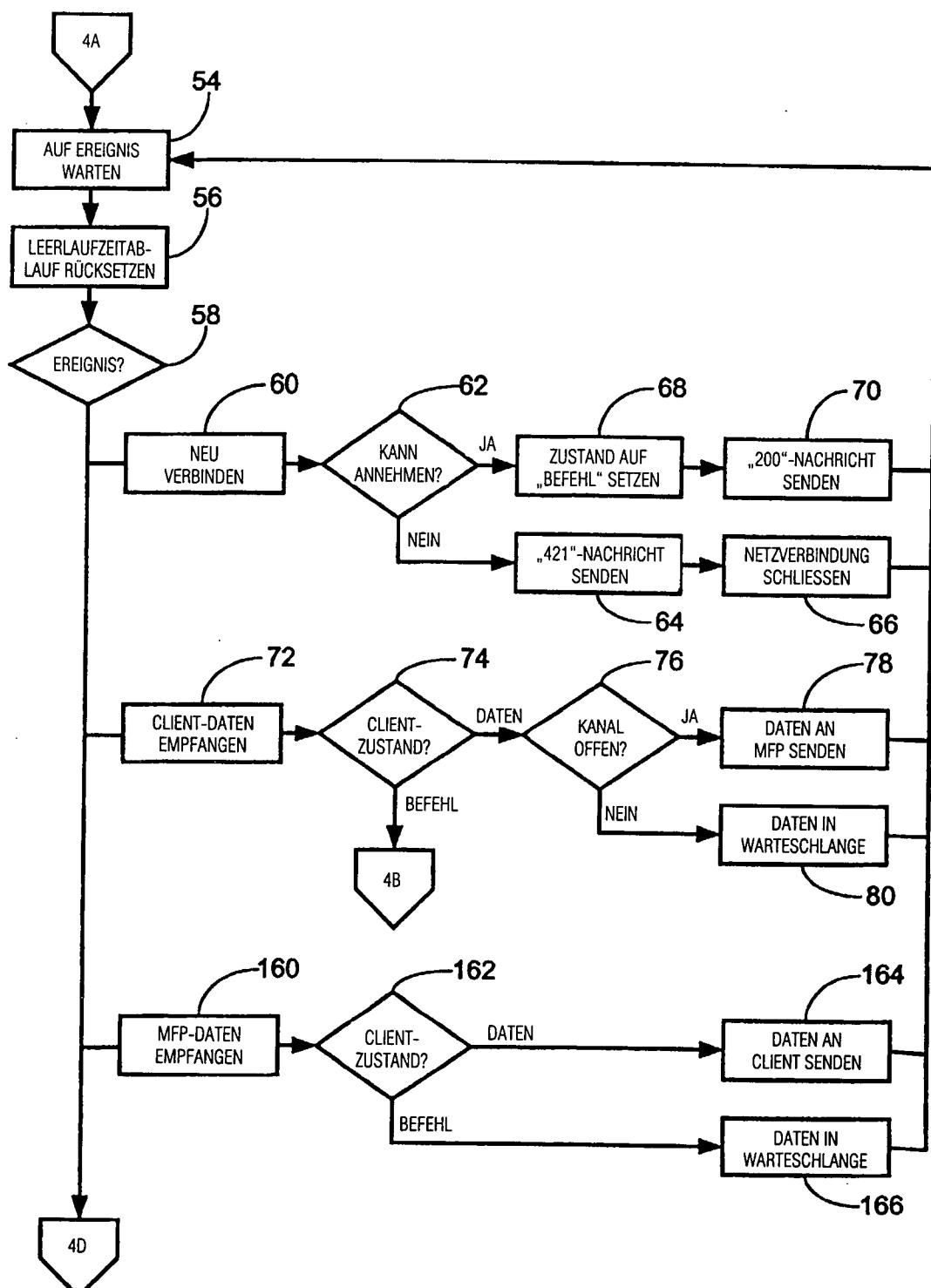
FIGUR 1

FIGUR 2

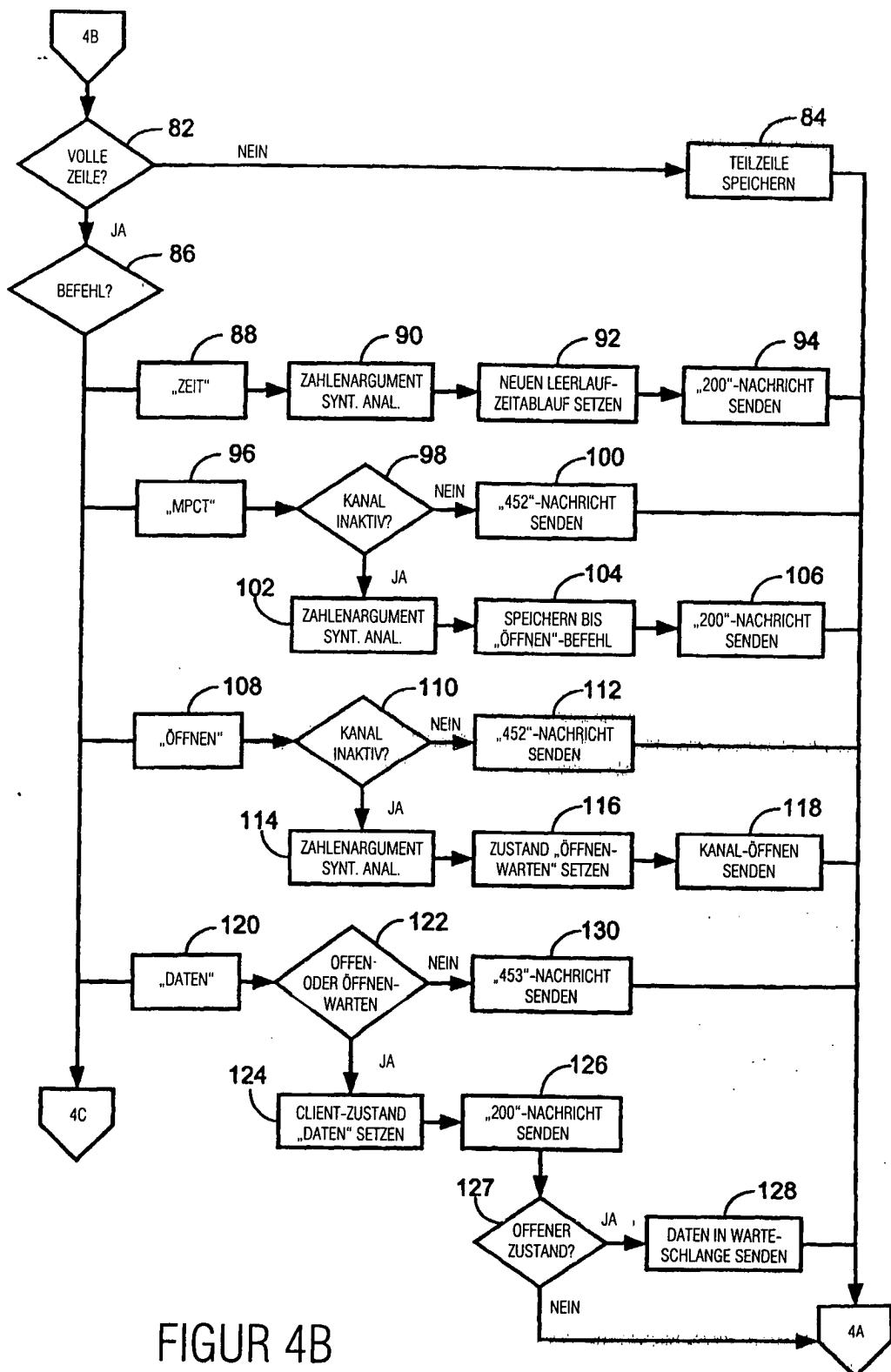


FIGUR 3

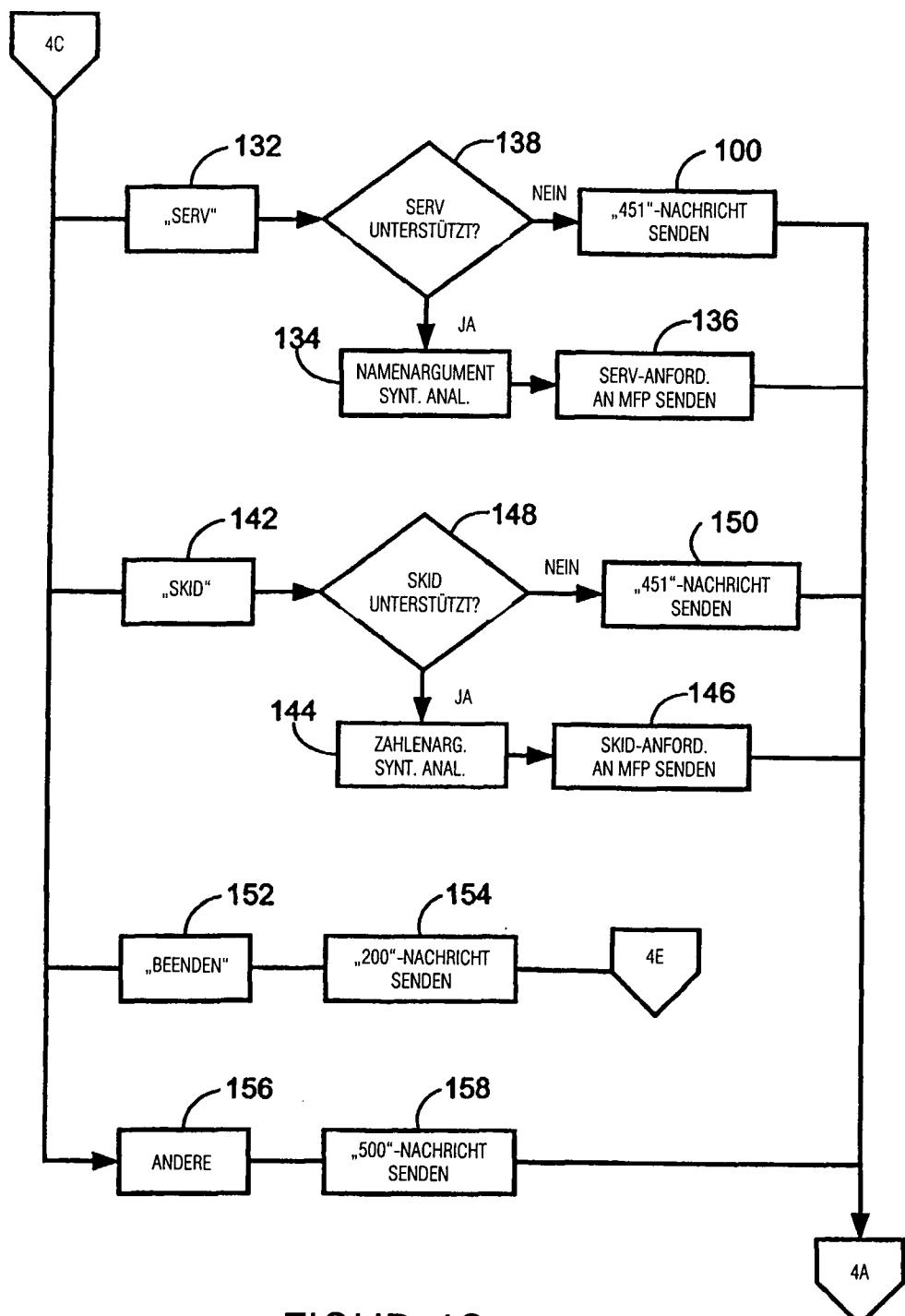




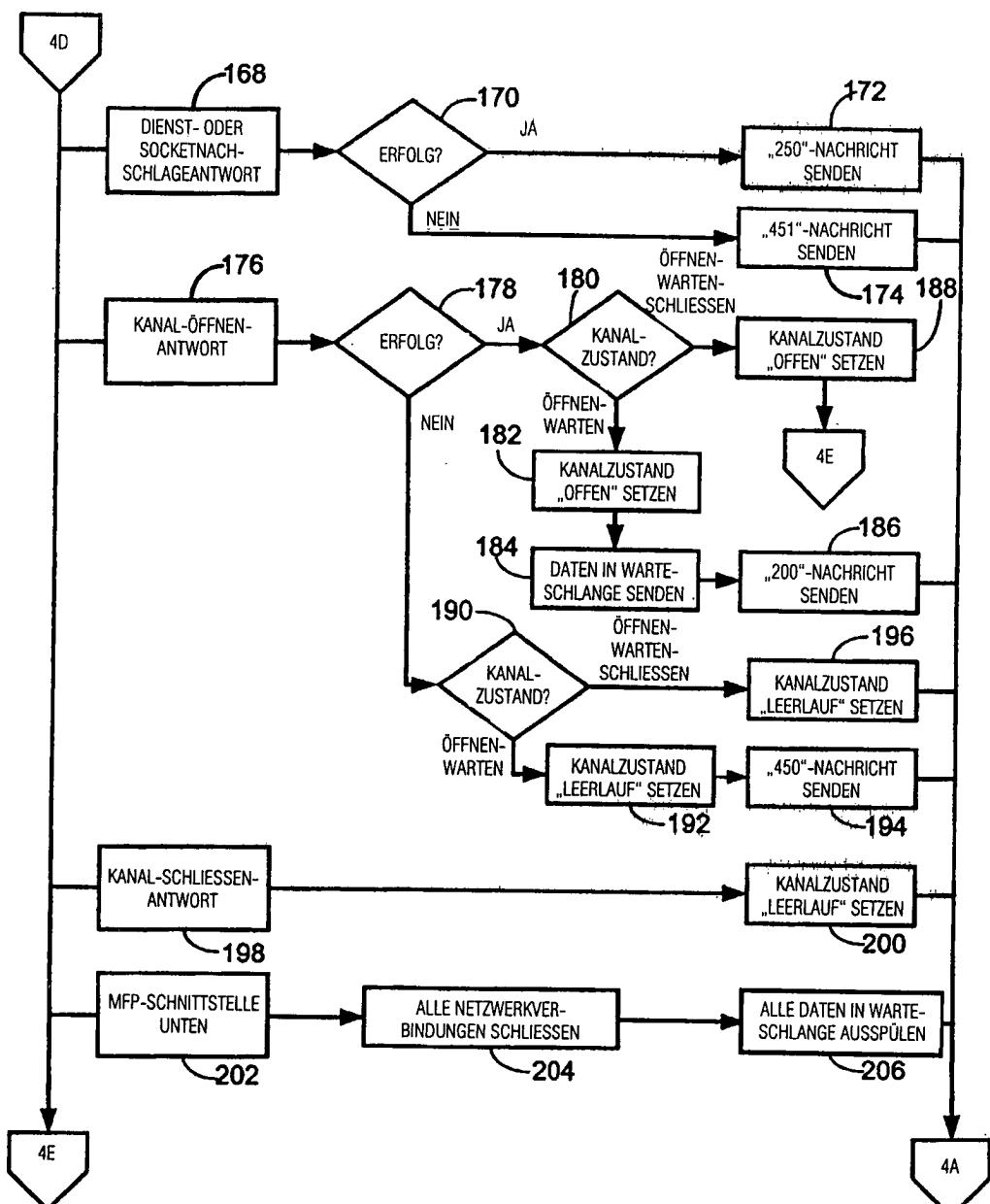
FIGUR 4A



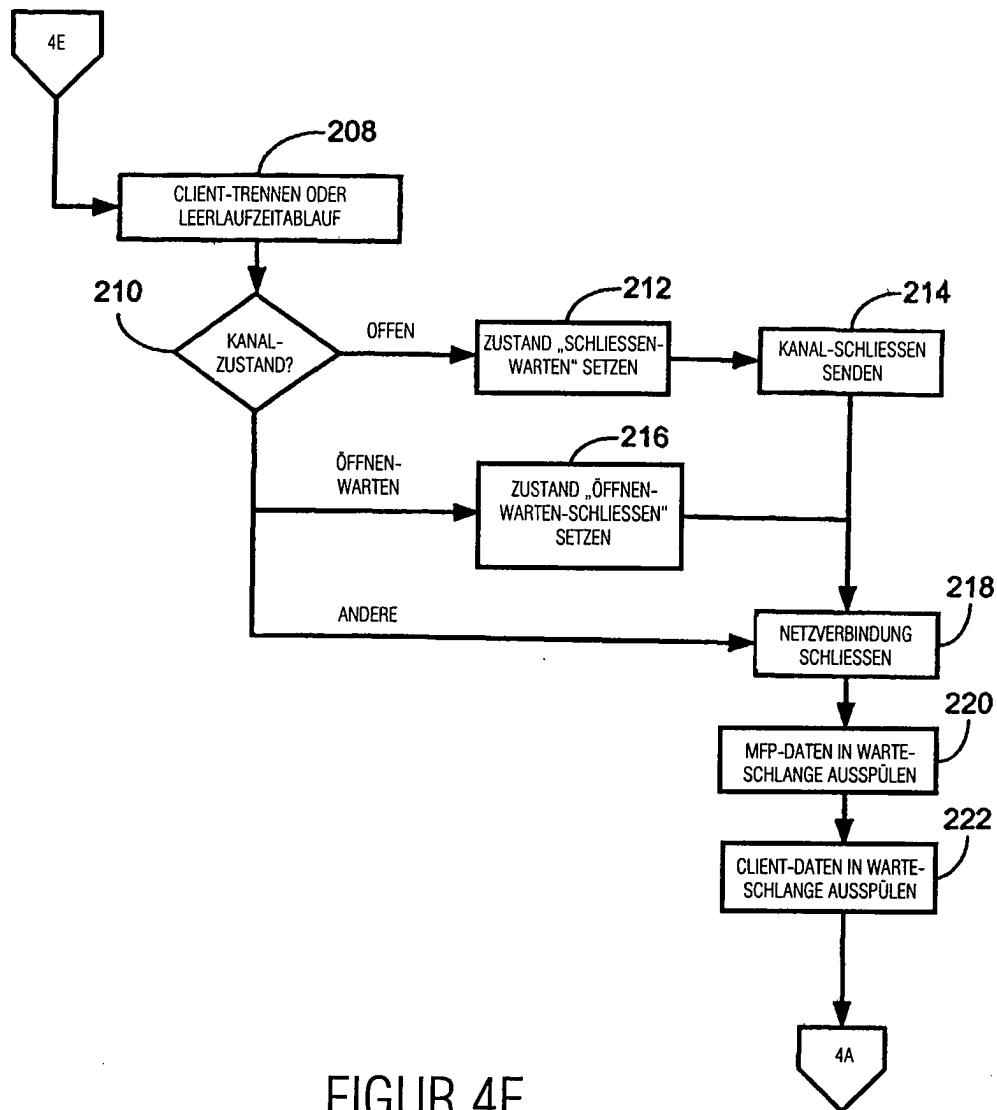
FIGUR 4B



FIGUR 4C



FIGUR 4D



FIGUR 4E