



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 699 33 852 T2 2007.07.12

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 046 260 B1

(21) Deutsches Aktenzeichen: 699 33 852.2

(86) PCT-Aktenzeichen: PCT/EP99/05767

(96) Europäisches Aktenzeichen: 99 941 549.0

(87) PCT-Veröffentlichungs-Nr.: WO 2000/010295

(86) PCT-Anmeldetag: 04.08.1999

(87) Veröffentlichungstag

der PCT-Anmeldung: 24.02.2000

(97) Erstveröffentlichung durch das EPA: 25.10.2000

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 02.11.2006

(47) Veröffentlichungstag im Patentblatt: 12.07.2007

(51) Int Cl.⁸: H04L 12/28 (2006.01)

H04L 29/06 (2006.01)

G06F 9/46 (2006.01)

(30) Unionspriorität:

133622 13.08.1998 US

(74) Vertreter:

Volmer, G., Dipl.-Ing., Pat.-Anw., 52066 Aachen

(73) Patentinhaber:

Koninklijke Philips Electronics N.V., Eindhoven,
NL

(84) Benannte Vertragsstaaten:

DE, ES, FR, GB, IT

(72) Erfinder:

FREEMAN, Lawrence, NL-5656 AA Eindhoven, NL

(54) Bezeichnung: HAUSNETZ- AUTOCONFIGURATION

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 auf ein Verfahren und ein System zum Vernetzen von Informationsverarbeitungssubsystemen. Die vorliegende Erfindung bezieht sich insbesondere auf die Konfiguration eines Netzwerkes von PCs in der Heimumgebung.

[0002] Eine bekannte Technologie der Vernetzung von Geräten ist Jini von Sun Microsystems. Jini ist eine Java-basierte Software-Technologie, die bei der Vernetzung von PCs und Randapparatur behilflich ist. Wenn in ein Netzwerk eingesteckt wird ein Jinifähiges Gerät seine Anwesenheit aussenden. Netzwerkunden, die dieses Gerät bereits verwenden, können die erforderliche Software von dem Gerät beantragen, wobei ein Server oder ein Netzwerkadministrator umgangen wird. Diese Architektur baut oben auf einem bestehenden Netzwerk. Es wird vorausgesetzt, dass das Netzwerk selber vorher bereits konfiguriert ist.

[0003] Zur Zeit hat eine wachsende Anzahl Haushalte mehr als nur einen PC. Dennoch sind die PCs nicht miteinander vernetzt, weil dies über die Kapazität des mittleren Benutzers geht. Sogar ein OS, wie Windows95, das ein sehr einfaches Geräteteilungsmodell hat, ist dennoch zu schwer zum Konfigurieren. Deswegen gibt es ein Bedürfnis nach einer Unterstützung des Benutzers beim Vernetzen von PCs.

[0004] Was erforderlich ist um Hausvernetzung allgegenwärtig zu machen ist, laut dem Erfinder, Autokonfiguration geteilter Mittel. Das heißt: wenn zwei oder mehr Maschinen miteinander verbunden sind, sollen die Mittel automatisch gemeinsam verwendet werden, und zwar ohne Zwischenkunft des Benutzers. Um eine Autokonfiguration gemeinsamer Mittel und gemeinsamer Dienst in einem PC-Netzwerk zu ermöglichen ist es erforderlich, dass die nachfolgenden Probleme gelöst werden: Detektion einer neuen Netzwerkverbindung; Zuordnung einer Netzwerkadresse ohne Zwischenkunft des Benutzers; und eine Implementierung eines Mittel/Dienst-Teilungsprotokolls.

[0005] Das erste Problem in Bezug auf die Detektion ist durch die "Plug-and-Play"-Technologie gelöst worden. So ermöglicht beispielsweise das Windows95 OS es, dass viele Ethernetkarten/"Token Ring"-Karten bei Einführung in das System automatisch funktionieren. Das zweite Problem in Bezug auf Adressenzuordnung wird zur Zeit in der Industrie erforscht. Eine bekannt Lösung ist ein Konfigurationsverwaltungsprotokoll, von dem ein Beispiel das DHCP-Protokoll ist ("Dynamic Host Configuration Protocol"), das Netzwerkadministratoren zentral verwalten und die Zuordnung der Internet-Protokoll-Adressen in einem Netzwerk der Organisation

automatisieren lässt.

[0006] Es ist nun u.a. eine Aufgabe der vorliegenden Erfindung eine Lösung für das dritte Problem zu schaffen, d.h. für die Implementierung eines gemeinsamen Protokolls.

[0007] Die vorliegende Erfindung wird durch die Hauptansprüche definiert, wobei einige mehr spezifische Ausführungsformen durch die Unteransprüche gedeckt werden.

[0008] Eine Ausführungsform der vorliegenden Erfindung bezieht sich auf ein Informationsverarbeitungssystem mit einem ersten Informationsverarbeitungssubsystem (beispielsweise einem PC), das mit einem zweiten Informationsverarbeitungssubsystem (beispielsweise einem anderen PC) gekoppelt ist. Das erste Subsystem hat ein erstes Register zum Registrieren wenigstens eines ersten Mittels oder eines ersten Dienstes örtlich für das erste Subsystem. Das zweite Subsystem hat ein zweites Register zum Registrieren wenigstens eines zweiten Mittels oder Dienstes, zu dem zweiten Subsystem gehörend. Das erste Subsystem hat einen ersten Proxy-Client, der in dem ersten Register registriert ist und das zweite Subsystem darstellt. Das zweite Subsystem hat einen zweiten Proxy-Client zur Kommunikation mit dem ersten Proxy-Client und um das erste Subsystem mit Zugriff auf das zweite Mittel bzw. den zweiten Dienst zu versehen.

[0009] Das Registrieren von Diensten und Mitteln des einen PCs bei dem anderen über die Proxy-Clients ermöglicht auf diese Weise eine automatische Konfiguration eines Netzwerkes um Mittel gemeinsam zu verwenden. Das Registrieren verbirgt die Idee, ob ein Mittel oder ein Dienst örtlich ist oder in einem anderen Gerät untergebracht ist. Mit anderen Worten die vorliegende Erfindung benutzt das Registrieren als Werkzeug zur Autokonfiguration eines Netzwerkes.

[0010] Jini Fokussiert auf den Prozess der Hinzufügung zu dem Netzwerk und auf das Aussenden von Information über das Gerät zu anderen Maschinen. Auf diese Weise schafft Jini einen "Nachschlag"-Dienst, der ermöglicht, dass Applikationen in anderen Maschinen das neu hinzugefügte Gerät benutzen. Die Annäherung von Jini setzt voraus, dass das Netzwerk und das Betriebssystem bereits derart konfiguriert sind, dass jeder Computer bereits über andere Computer informiert worden ist. Die Funktionalität von Jini tritt auf in einer Schicht über dem Netzwerk. Es löst beispielsweise nicht die Probleme der automatischen Konfiguration des Netzwerkes bei Verbindung, Entkopplung oder Neuverbindung.

[0011] Es setzt voraus, dass das Netzwerk aufwärts oder abwärts ist, unabhängig von Jini. Jini macht die

Dienste wirksam, die von dem Netzwerk geliefert werden um die Dienste zu implementieren. Mit anderen Worten, die vorliegende Erfindung benutzt die Registrierung als Werkzeug zur Autokonfiguration.

[0012] Der Vollständigkeit halber sei auf GB-A-2 305 271 verwiesen. Diese Veröffentlichung bezieht sich auf einen Mechanismus in objekt-orientierter Technologie zum Schutz gegen ungültigen Gebrauch von Proxy-Objekten nach einem Defekt eines Servers und auch zur transparenten Neuschaffung von Proxy-Objekten in einem Client eines Client-Server verteilten Verarbeitungssystems. Eine Proxy-Klasse wird verwendet, die zusätzliche Attribute hat, die den Namen des Zielobjektes in dem Server angeben, eine Anzeige wie, ob der Name zur Zeit gültig ist und einen wechselnden Zeiger auf das Zielobjekt. Ein Proxy-Registerobjekt in dem Client behält Zeiger auf alle Proxy-Objekte, die auf Objekte in dem Server zeigen. Bei einem Defekt des Servers und folglich bei einer Ungültigkeit der Proxy-Objekte sorgt das Proxy-Registerobjekt dafür, dass alle Proxy-Objekte erneuert werden.

[0013] In der vorliegenden Erfindung stellt der erste Proxy-Client das zweite Subsystem bei dem ersten Subsystem dar. Der erste Proxy-Client wird mit dem Register des ersten Subsystems registriert. Das Register des ersten Subsystems registriert auch Mittel, die für das erste Subsystem örtlich sind. Auf entsprechende Weise lehrt weder suggeriert GB-A 2 305 271 dass das erste Register Mittel oder Dienste örtlich für das erste Subsystem registriert und auch nicht den Proxy-Client registriert, der für das zweite Subsystem repräsentativ ist. GB-A 2 305 271 weder lehrt noch suggeriert, dass das zweite Subsystem ein zweites Register hat zum Registrieren von Mitteln, die für das zweite Subsystem örtlich sind. Das Problem, das die vorliegende Erfindung zu lösen versucht, ist anders als das Problem in GB-A 2 305 271.

[0014] Ausführungsbeispiele der vorliegenden Erfindung sind in der Zeichnung dargestellt und werden im Folgenden näher beschrieben. Es zeigen:

[0015] [Fig. 1](#) eine Darstellung der Konfiguration eines Systems nach der vorliegenden Erfindung; und

[0016] [Fig. 2](#) bis [Fig. 7](#) je eine Darstellung, die mehrere Schritte in dem Autokonfigurationsprozess und im Betrieb darstellt.

[0017] In den Figuren bezeichnen dieselben Bezeichnungen ähnliche oder entsprechende Elemente.

[0018] [Fig. 1](#) ist ein Blockschaltbild mit den Hauptlementen eines Betriebssystems **100** nach der vorliegenden Erfindung. Das System **100** umfasst einen ersten PC **102** und einen zweiten PC **104**, die über einen Bus **106** miteinander gekoppelt sind. Der Bus

106 kann ein verdrahteter Bus oder ein drahtloser Bus sein oder eine Kombination derselben sein. Der PC **102** hat Mittel und schafft Dienste. So hat beispielsweise der PC **102** eine Festplatte **108**, eine Email-Fähigkeit **110**, einen Web-Browser **112**, einen Drucker **114** usw. Auf gleiche Weise hat der PC **104** Mittel und Dienste, wie eine Festplatte **116**, einen Textverarbeitungsdienst **118**, ein Graphikprogramm **120**, einen Drucker **122**, usw. Der Ausdruck "Mittel" und "Dienst" werden nachstehend der Kürze wegen austauschbar verwendet.

[0019] Der PC **102** hat ein Register **124** zum Registrieren von Schnittstellen zu Mitteln und Diensten **108–114** örtlich für den PC **102**. Applikationen, die in dem PC **102** laufen, können auf diese Schnittstellen zugreifen. Die Schnittstellen leiten die Nachrichten oder Anträge zu örtlichen Mitteln oder Diensten **108–114** weiter. Auf gleiche Weise hat der PC **104** ein Register **126** zum Registrieren von Schnittstellen zu Mitteln und Diensten **116–122**, örtlich für den PC **104**.

[0020] Der PC **102** umfasst weiterhin die folgenden Entitäten, die übereinstimmend sind und Prozesse sein können: einen Sender **128**, einen Porthörer **130** und einen Sendehörer **132**. Auf gleiche Weise hat der PC **104** einen Sender **134**, einen Porthörer **136** und einen Sendehörer **138**. Der PC **102** umfasst weiterhin einen Proxy-Client **142** und einen Proxy-Server **144**. Der PC **104** hat einen Proxy-Client **140** und einen Proxy-Server **144**. Der Proxy-Client **142** kommuniziert mit dem Proxy-Server **146**, und der Proxy-Client **140** kommuniziert mit dem Proxy-Server **146**. Die von den Komponenten **124–138** in der Autokonfiguration des Systems **100** gespielten Rollen und die von den Proxy-Servern **146** und **144** und den Proxy-Clients **142** und **144** gespielten Rollen werden anhand der [Fig. 2–Fig. 7](#) erläutert.

[0021] [Fig. 2](#) ist eine Darstellung der Anfangskonfiguration des Systems **100**, wenn die PCs **102** und **104** mit dem Bus **106** funktionell verbunden sind. In dieser Konfiguration senden die Sender **128** zu dem Bus **106** eine Netzwerkadresse für den PC **102**, sagen wir "X" und eine Portnummer, sagen wir "x", über einen bestehenden Kanal. Auf gleiche Weise sendet der Sender **134** ein Signal zu dem Bus **106** mit einer Netzwerkadresse für PC **104**, "Y" und eine Portnummer "y" über einen anderen bestehenden Kanal. Der Sendehörer **138** des PCs **104** empfängt die von dem PC **102** ausgesendete Nachricht. Der Sendehörer **132** des PCs **102** empfängt die von dem PC **104** ausgesendete Nachricht.

[0022] [Fig. 3](#) zeigt einen nächsten Schritt des Autokonfigurationsprozesses. Nachdem die von dem PC **104** ausgesendete Nachricht empfangen worden ist, erzeugt der Sendehörer **132** des PCs **102** einen Proxy-Client **142**. Der Proxy-Client **142** stellt daraufhin eine Verbindung mit dem Porthörer **136** des PCs **104**

beim Port "y" har. Auf gleiche Weise erzeugt der Senderhörer **138** des PC **104** den Proxy-Client **140**, der eine Verbindung mit dem Porthörer **130** beim Port "x" herstellt.

[0023] [Fig. 4](#) zeigt einen weiteren Schritt in dem Autokonfigurationsprozess. Der Porthörer **130** startet einen Proxy-Server **144** um Anträge von dem Fern-Client **140** zu erledigen. Der Proxy-Server **144** sendet Information über Mittel **108–114**, wie diese beispielsweise in dem Register **124** enthalten sind, zu dem Proxy-Client **140**. Der Proxy-Client **140** registriert diese Information mit dem Register **126**. Auf gleiche Weise startet der Porthörer **136** einen Proxy-Server **146** um Anträge von dem Fern-Client **142** zu erledigen. Der Proxy-Server **146** sendet Information über Mittel **116–122** dem Proxy-Client **142** zu, der daraufhin diese Information mit dem Register **124** registriert.

[0024] [Fig. 5](#) zeigt den Schritt, in dem der Proxy-Client **142** mit dem Register **124** als örtlicher Server für jedes Mittel oder für jeden Dienst, der in dem Register **126** verfügbar ist, registriert und wobei der Proxy-Client **140** mit dem Register **126** für jedes Mittel oder für jeden Dienst, der in dem Register **124** verfügbar ist, registriert. Das Ergebnis ist, dass der PC **102** sein Register **124**, das die Adressen der örtlichen Mittel und Dienste spezifiziert, kopiert hat, wobei die Kopie dem Register **126** hinzugefügt wird. Auf gleiche Weise hat der PC **104** sein Register **126** zu dem PC **102** registriert, wo es zu dem Register **124** hinzugefügt wird. Die beiden PCs **102** und **104** sind nun miteinander im Register. Wenn ein dritter PC **148** mit dem Bus **106** verbunden wird, tritt automatisch ein Prozess entsprechend dem oben beschriebenen Prozess auf. Die Register **124** und **126** verstecken auf diese Art und Weise die Idee, ob ein Mittel oder ein Dienst örtlich ist oder in einem anderen Gerät zu Hause ist. Jede der Adressen in dem Register **124** ist in dem ganzen Register **124** einzigartig. Auf gleiche Weise ist jede der Adressen in dem Register **126** in dem ganzen Register **126** einzigartig. Ein Benutzer, der mit dem PC **102** arbeitet und ein örtliches Mittel oder einen örtlichen Dienst beantragt, d.h. eines der Mittel oder einen der Dienste **108–114**, sendet den Antrag unmittelbar zu dem beantragten Mittel oder Dienst, angegeben durch die entsprechende Adresse in dem Register **124**. Wenn der Benutzer einen Ferndienst oder ein Fernmittel beantragt, d.h. eines der Mittel oder einen der Dienste **116–122**, das bzw. der für den Fern-PC örtlich ist, wird der Antrag dem Proxy-Client **142** zugeführt und von dem Proxy-Server **146** verarbeitet, wie nachstehend anhand der [Fig. 7](#) beschrieben wird.

[0025] [Fig. 6](#) zeigt eine Darstellung **600** und **602** als ein Beispiel der Registrierung des Proxy-Clients **142** mit dem Register **124**. Die Darstellung **600** stellt das Anfangsregister **126** mit einer Liste örtlicher Mittel

und Dienste dar, die in dem PC **104** verfügbar sind, sowie deren betreffende örtliche Adressen. Die Darstellung **602** stellt das Register **124** dar, nachdem der Client **142** damit registriert worden ist. Das Register **124** umfasst zunächst die Liste mit Mitteln und Diensten **108–114** mit örtlichen Adressen #1 bis #K. Nachdem der Client **142** registriert hat, hat das Register **124** einen Eingang für den PC **104** als Proxy-Anordnung auf der Adresse #Q. Die Fernmittel und Ferndienste **116–122** haben nun Adressen, die von der Adresse #Q abhängig sind.

[0026] [Fig. 7](#) zeigt diesen auf eine Adresse basierten Vorgang betreffend den Proxy-Client **142** und Proxy-Server **146**. Eine Software-Applikation **702**, die im PC **102** läuft, erzeugt einen Antrag für das Mittel **118** in dem Fern-PC **104**. Dem Register **124** ist ein Bezugswert auf das Mittel **118** hinzugefügt worden, wie oben beschrieben. Der Bezugswert hat einen Zeiger (Pfeil **704**) zu dem Proxy-Client **142**. Der Proxy-Client **142** im PC **102** kontaktiert (Pfeil **706**) den Proxy-Server **146** in dem Fern-PC **104**. Der Proxy-Server **146** hat eine Adresse, oder eine Hantel (Pfeil **708**) zu dem Mittel **118** über das örtliche Register **126**. Ergebnisse der Verarbeitung durch das Mittel **118** werden über den Proxy-Server **146** und den Proxy-Client **143** (Pfeil **706**) zu der Applikation **702** (Pfeil **712**) zurückgeführt. Die Proxy-Schaltung **142** und der Proxy-Server **146** sind als Pipeline wirksam.

[0027] Wenn ein Antrag oder eine Nachricht, erzeugt in dem PC **102**, eine Adresse für eines der örtlichen Mittel oder Dienste **108–114** hat, wird die Nachricht unmittelbar dem Dienst- oder Anordnungs-Driver (nicht dargestellt) des betreffenden Mittels zugeführt. Fall die Adresse nicht örtlich ist, beispielsweise sich auf das Mittel **118** in dem PC **104** bezieht, wird die Nachricht dem Proxy-Client **142** zugeführt (**702**), der an sich wieder den Antrag dem Proxy-Server **146** bei dem zugeordneten Port zusendet (**704**). Der Proxy-Server **146** verarbeitet den Antrag und sendet diesen (**706**) dem betreffenden Mittel der Mittel **116–122** über das Register **126** zu. Das Ergebnis des verarbeiteten Antrags wird danach von dem Proxy-Server **146** dem Proxy-Server **142** zugeführt, von woraus das Ergebnis beispielsweise einer örtlichen Applikation im PC **102** oder einem Wiedergabe-Driver (nicht dargestellt) örtlich im PC **102** zugeführt wird.

[0028] Wenn die Verbindung zwischen den zwei PCs unterbrochen wird, kann die Adresse der Fern-Clients **142** und **140** aus den örtlichen Registern **124** bzw. **126** entfernt werden.

[0029] Die Mittel zum Bilden der Autokonfiguration, wie oben beschrieben: Sender, Porthörer, Senderhörer, Register, Proxy-Client und Proxy-Server sind, anfangs, in eigenständigen PC **102** und **104** installiert, beispielsweise als Teil der Betriebssysteme oder als

Applikationen, die ganz oben in dem Betriebssystem laufen. Die Autokonfigurationsmittel werden verwendet, wenn der eigenständige PC **102** oder **104** mit einem Netzwerk verbunden wird oder wenn die eigenständigen PCs **102** und **104** miteinander verbunden werden. Die Software wird den Benutzern beispielsweise als ein Programm auf einer Diskette zur Verfügung gestellt oder kann von dem Web herunter geladen werden.

[0030] Eine Implementierung dieses Autokonfigurationssystems kann unter Anwendung eines Java-basierten Systems demonstriert werden. In diesem Szenario haben zwei eigenständige PCs Java, eine Netzwerkkarte und einen TCP/IP-Stack in den Maschinen installiert. Zur Anwendung von TCP/IP wird eine betreffende IP-Nummer für jede betreffende Maschine verwendet. Die Idee dabei ist, dass die IP-Adresse und die Standard-TCP/IP-Einstellungen nicht von dem Benutzer vorkonfiguriert worden sind. Zusätzlich zu dieser IP-Adresse eine einzigartige ID (UID) erzeugt – dies kann eine beliebige Zahl einer ausreichenden Komplexität sein, so dass ein Zusammenstoß statistisch unwahrscheinlich ist (ein gutes Beispiel davon ist "Global Unique IDentifier" von Microsoft [GUID]). Die IP-Adresse wird verwendet um es zu ermöglichen, dass TCP/IP jeden PC identifiziert und die UID wird verwendet um zu gewährleisten, dass die beliebig erzeugten IPs nicht gleich sind. Dies wird geschieht, da dieselbe ID-Adresse notwendigerweise mit derselben UID verbunden werden muss.

[0031] In dem Java-System wird eine funktionsfähige Laufzeit gestartet, die drei Objekte hervorbringt: einen Sender, einen Sendehörer und einen Porthörer – je als Java-Thread. In dem Fall könnte der Sendehörer einen Java-Mehrfachsendungsanschluss um eine Mehrfachsendung zu abonnieren. Auf gleiche Weise könnte der Sender auch einen Java-Mehrfachsendungsanschluss verwenden um einer Mehrfachsendungsgruppe Information zuzusenden. Der Porthörer könnte als Java-Serveranschluss implementiert werden.

[0032] Wenn der Sendehörer eine Nachricht empfängt und die IP-Adresse ist dieselbe wie die eigene IP-Adresse und die UID ist nicht dieselbe, schaltet dies jeden der Proxy-Server aus, erzeugt beliebig eine neue IP-Adresse und konfiguriert sich selber automatisch neu. Auf diese Weise werden die IP-Adressen jeder Maschine ggf. derart gewährleistet, dass sie einzigartig sind.

[0033] Wenn der Sendehörer eine Nachricht empfängt und die IP-Adresse ist dieselbe wie die eigene IP-Adresse und die UID ist dieselbe, dann wird die Nachricht ignoriert, da die Nachricht von dem eigenen Sender erzeugt wird.

[0034] Wenn der Sendehörer eine Nachricht empfängt und die IP-Adresse ist anders und diese IP-Adresse wurde vorher gefunden, dann wird ein neuer Java-Faden erzeugt; ein Proxy-Client, der eine Verbindung mit dem Porthörer in der anderen Maschine hat. Die gesendete Nachricht enthält eine Port- und eine IP-Adresse für den Proxy-Client um diesen als Java-Abschluss zu verbinden. Bei Empfang dieses Antrags bringt der Porthörer über das Java-Akzeptationsverfahren einen neuen Java-Faden hervor um diesen Antrag zu erledigen: einen Proxy-Server. Mittelinformation strömt von dem Proxy-Server zu dem Proxy-Client zurück zu dem Register.

[0035] Zwecks dieses Java-Beispiels kann das Register als Hash-Tabelle organisiert sein, wobei jedes Mittel durch eine einzigartige Adresse, die erzeugt wird, wenn das Mittel oder der Dienst zu dem Register hinzugefügt wird, identifiziert werden kann. Wenn der Proxy-Client zunächst eine Verbindung mit dem Proxy-Server bildet, sendet der Proxy-Server Information über das aktuelle Register. Der Proxy-Client fügt danach diese Information dem Register zu, der für den Proxy-Client örtlich ist. Zwecks dieses Beispiels ist die wichtige Idee, dass eine Hantel mit dem Proxy-Client in dem Register assoziiert ist, so dass der Prozess oder der Faden, der für die neue Adresse örtlich ist, der Proxy-Client statt eines örtlichen Prozesses oder Fadens. Wenn der Dienst verwendet wird, wird Steuerinformation dem Proxy-Client zugeführt, der diese Information dem Proxy-Server zuführt, der danach diese Information dem Mittel oder dem Dienst in der anderen Maschine zuführt.

[0036] In dem Fall der Java-Implementierung besteht jeder der Dienste aus einem Java-Objekt, das ByteDaten empfängt, entweder von einem anderen Objekt unter Verwendung des Dienstes oder des Mittels, oder von dem Proxy-Server. Auf diese Weise erscheinen die Client-Applikation unter Verwendung des Dienstes und der Dienst selber örtlich in derselben Maschine.

[0037] Die wichtige Idee hinter dem verwendeten Register ist, dass sie Dienste/Mittel mit Adressen assoziiert und Datenpakete nimmt, die sie zu dem betreffenden Dienst oder Mittel überträgt, und zwar auf Basis dieser Adresse. Im Falle einer Java-Implementierung werden diese Daten unter Verwendung eines Java-Dateneingangsstromes und eines Datenausgangsstromes weitergeleitet.

[0038] Wenn die Verbindung zwischen den beiden PCs unterbrochen wird, kann die Adresse der Fern-Clients **142** und **140** aus den örtlichen Registern **124** bzw. **126** entfernt werden. So gewährleistet ein Herzschlagprotokoll zwischen den PCs, dass jeder den anderen bewusst ist bis der Herzschlag eines der PCs verschwindet. Ein Zeitgeber erlischt und triggert

automatisch die Prozedur um die Eingabe des Registers zu dem betreffenden Client zu sperren.

Patentansprüche

1. Informationsverarbeitungssystem (100) mit einem ersten Informationsverarbeitungssubsystem (102), das mit einem zweiten Informationsverarbeitungssubsystem (104) gekoppelt ist, wobei:

- das erste Subsystem ein erstes Register (124) aufweist zum Registrieren wenigstens eines ersten Hilfsmittels oder Dienstes (108 114), die zu dem ersten Subsystem gehören;
- das zweite Subsystem einen zweiten Proxy-Server (146) aufweist zur Kommunikation mit dem ersten Proxy-Client und um dem ersten Subsystem mit Zugriff auf das zweite Hilfsmittel oder den zweiten Dienst zu versehen, und wobei das Informationsverarbeitungssystem

dadurch gekennzeichnet ist, dass

- das zweite Subsystem ein zweites Register (126) hat zum Registrieren wenigstens eines zweiten Hilfsmittels oder Dienstes (116 122), die zu dem zweiten Subsystem gehören, und
- das erste Subsystem einen Proxy-Client (142) hat, der mit dem ersten Register registriert ist, und das zweite Subsystem darstellt.

2. System nach Anspruch 1, wobei:

- das zweite Subsystem einen zweiten Proxy-Client (140) aufweist, der durch das zweite Register registriert ist, und
- das erste Subsystem einen ersten Proxy-Client (144) hat zur Kommunikation mit dem zweiten Proxy-Client und zum Zugreifen auf das erste Hilfsmittel oder den ersten Dienst.

3. System nach Anspruch 1, wobei das erste Subsystem einen ersten PC aufweist, und wobei das zweite Subsystem einen zweiten PC aufweist.

4. Informationsverarbeitungssubsystem (102), das Folgendes umfasst:

- ein Register (124) zur Registrierung eines Hilfsmittels oder eines Dienstes (108 114), dem Subsystem gehörend, und das Kennzeichen aufweist, dass es Folgendes umfasst:
- ein Sendemodul (128) zum Aussenden einer Nachricht,
- einen Sendungszuhörer (132) zum Empfangen einer Antwort von einem anderen Informationsverarbeitungssystem in Reaktion darauf, dass das andere Subsystem die Nachricht empfangen hat, und zum daraufhin Schaffen eines Proxy-Clients (142), und dass der Proxy-Client mit dem Register als ein Vertreter des anderen Subsystems für das Subsystem registriert wird um auf ein anderes Hilfsmittel oder einen anderen Dienst zuzugreifen, das bzw. der einem anderen Subsystem zugehört.

5. Subsystem nach Anspruch 4, das weiterhin einen Portzuhörer (130) aufweist um in Reaktion auf den Empfang einer weiteren Nachricht von dem anderen Subsystem einen Proxy-Server (144) zu schaffen zum Liefern von Mitteln für das andere Subsystem zum Zugreifen auf das Mittel oder den Dienst, das bzw. der dem Subsystem zugehört.

6. Subsystem nach Anspruch 4, mit einem PC.

7. Verfahren zum Schaffen von Mitteln für ein erstes Informationsverarbeitungssubsystem (102) zum Adressieren eines zweiten Hilfsmittels oder Dienstes (116 122), das bzw. der mit einem zweiten Register (126) eines zweiten Informationsverarbeitungssubsystems (104) im Register ist, wobei das Verfahren Folgendes umfasst:

- das Schaffen von Mitteln zum Schaffen eines Proxy-Clients (142) bei dem ersten Subsystem zur Kommunikation mit dem zweiten Subsystem, wobei der Proxy-Client ein Vertreter des zweiten Hilfsmittels oder Dienstes ist,
- das Schaffen von Mitteln zum Registrieren des Proxy-Clients als ein örtliches Hilfsmittel oder Dienstes mit einem ersten Register (124) des ersten Subsystems, wobei das Register wenigstens ein erstes Hilfsmittel oder einen ersten Dienst, das bzw. der dem ersten Subsystem zugehört, registriert, und
- das Schaffen von Mitteln zum Schaffen eines Proxy-Servers (146) bei dem zweiten Informationsverarbeitungssubsystem zum Behandeln eines Antrags von dem Proxy-Client.

8. Verfahren nach Anspruch 7, das den nachfolgenden Verfahrensschritt umfasst:

- das Schaffen von Mitteln für das zweite Informationsverarbeitungssubsystem zum Adressieren des ersten Hilfsmittels oder Dienstes (108 114), wobei das Verfahren weiterhin die nachfolgenden Verfahrensschritte umfasst:
- das Schaffen von Mitteln zum Schaffen eines weiteren Proxy-Clients (140) bei dem zweiten Subsystem zur Kommunikation mit dem ersten Subsystem, wobei der weitere Proxy-Client Vertreter des ersten Hilfsmittels oder Dienstes ist,
- das Schaffen von Mitteln um den weiteren Proxy-Client als ein weiteres örtliches Hilfsmittel oder einen weiteren örtlichen Dienst mit dem zweiten Register (126) in Register zu bringen,
- das Schaffen von Mitteln zum Schaffen eines weiteren Proxy-Servers (144) bei dem ersten Informationsverarbeitungssubsystem zum Behandeln eines weiteren Antrags von dem weiteren Proxy-Client.

9. Verfahren nach Anspruch 7 oder 8, wobei das erste und das zweite Subsystem je einen betreffenden PC aufweisen.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

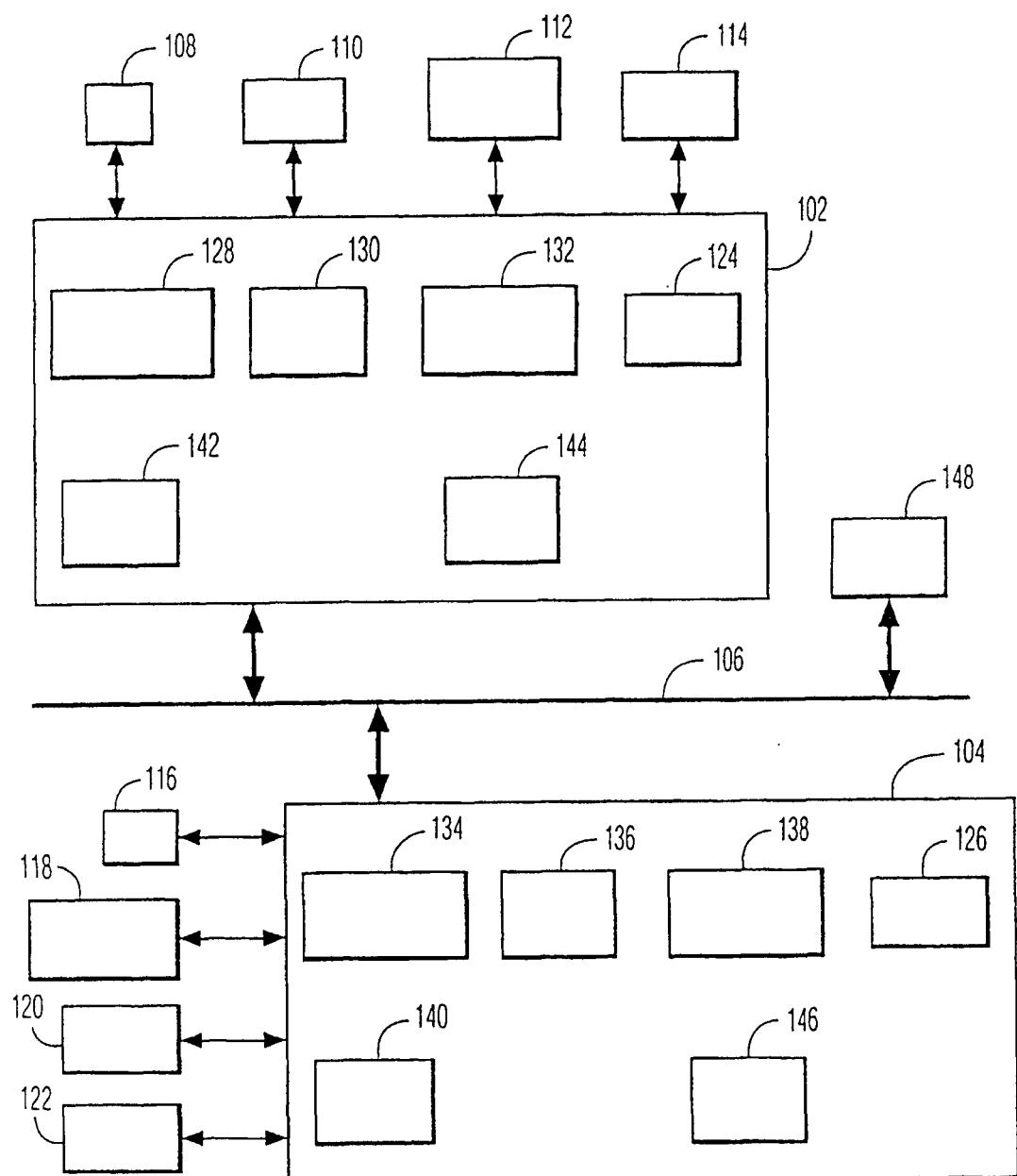
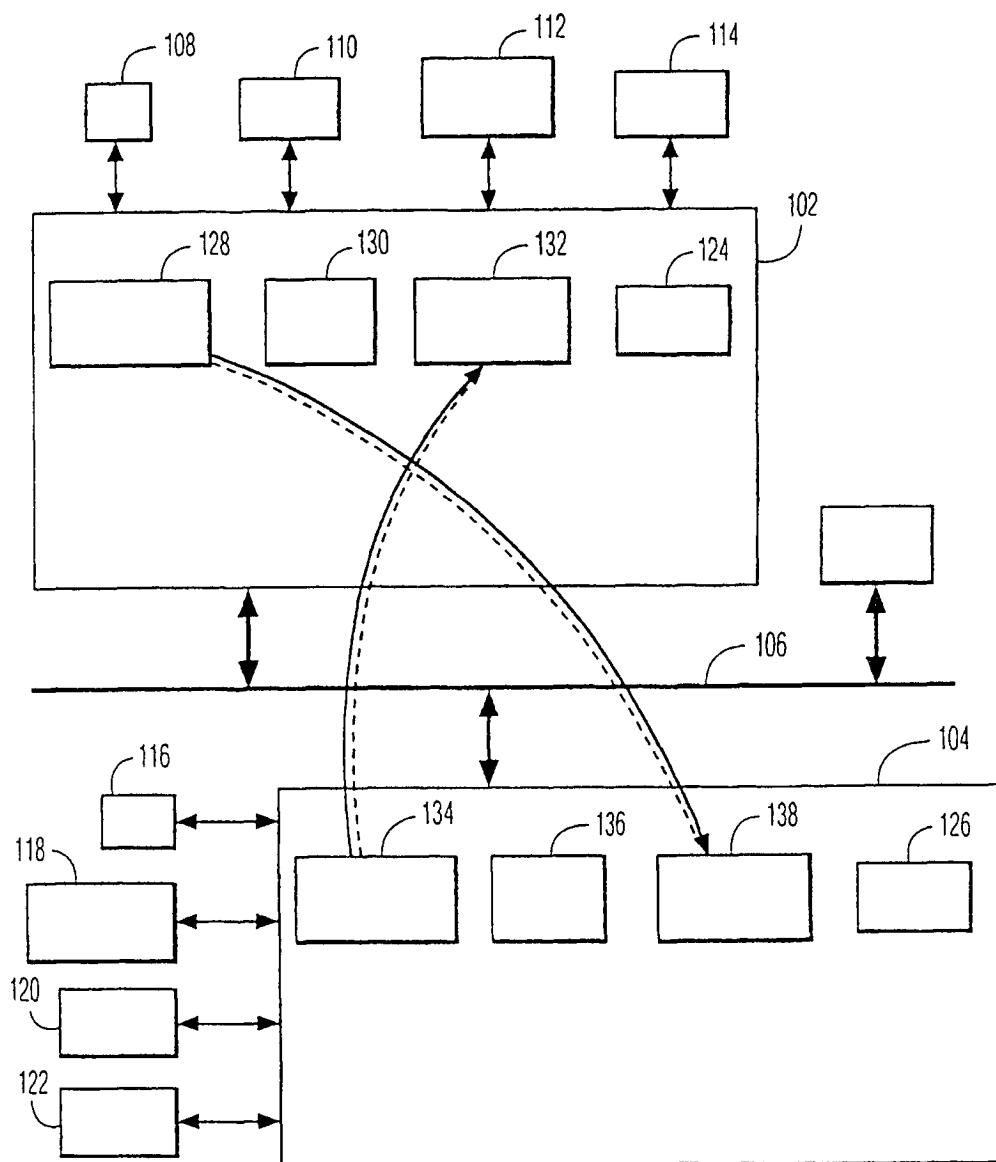
100

FIG. 1



100

FIG. 2

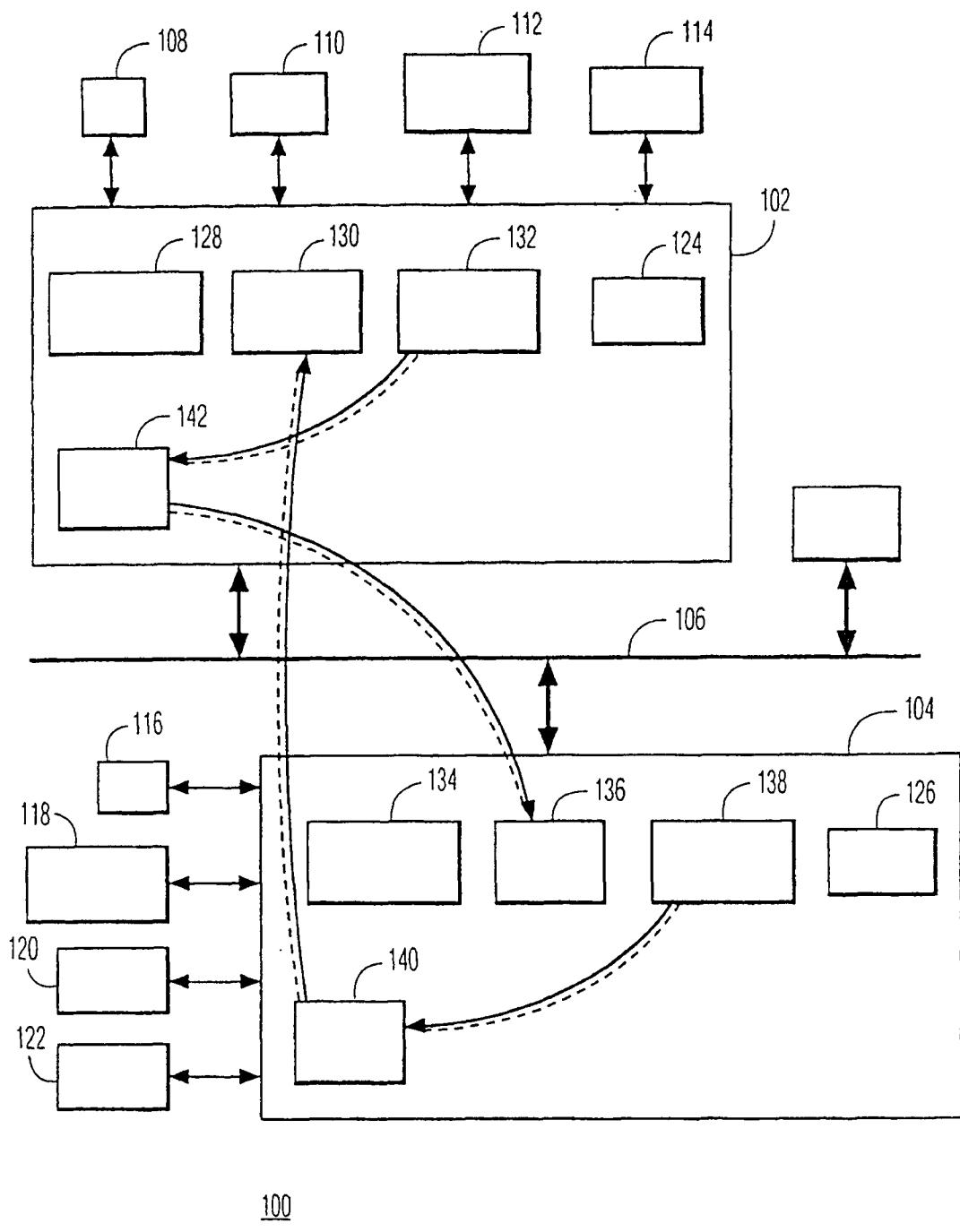
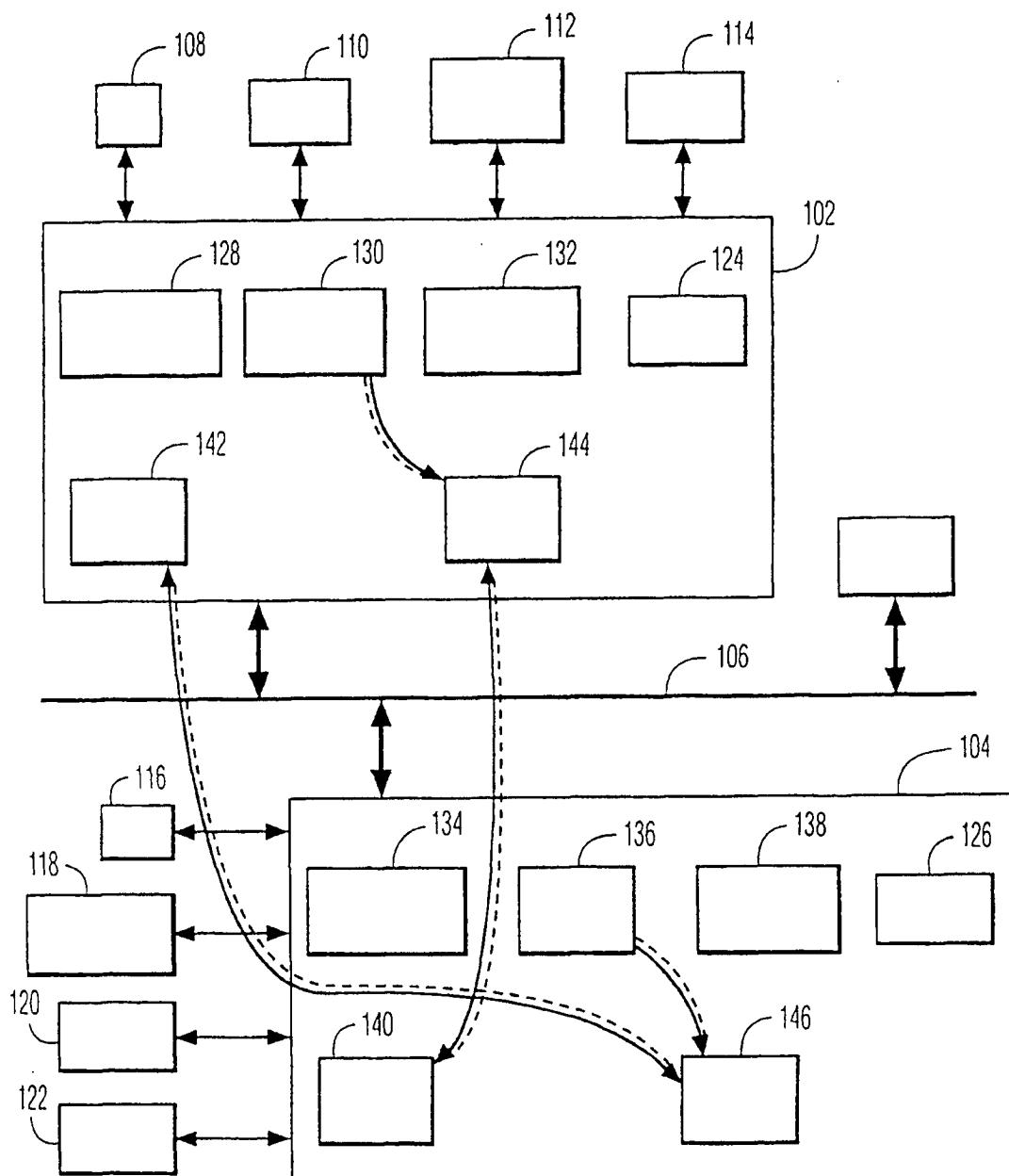


FIG. 3



100

FIG. 4

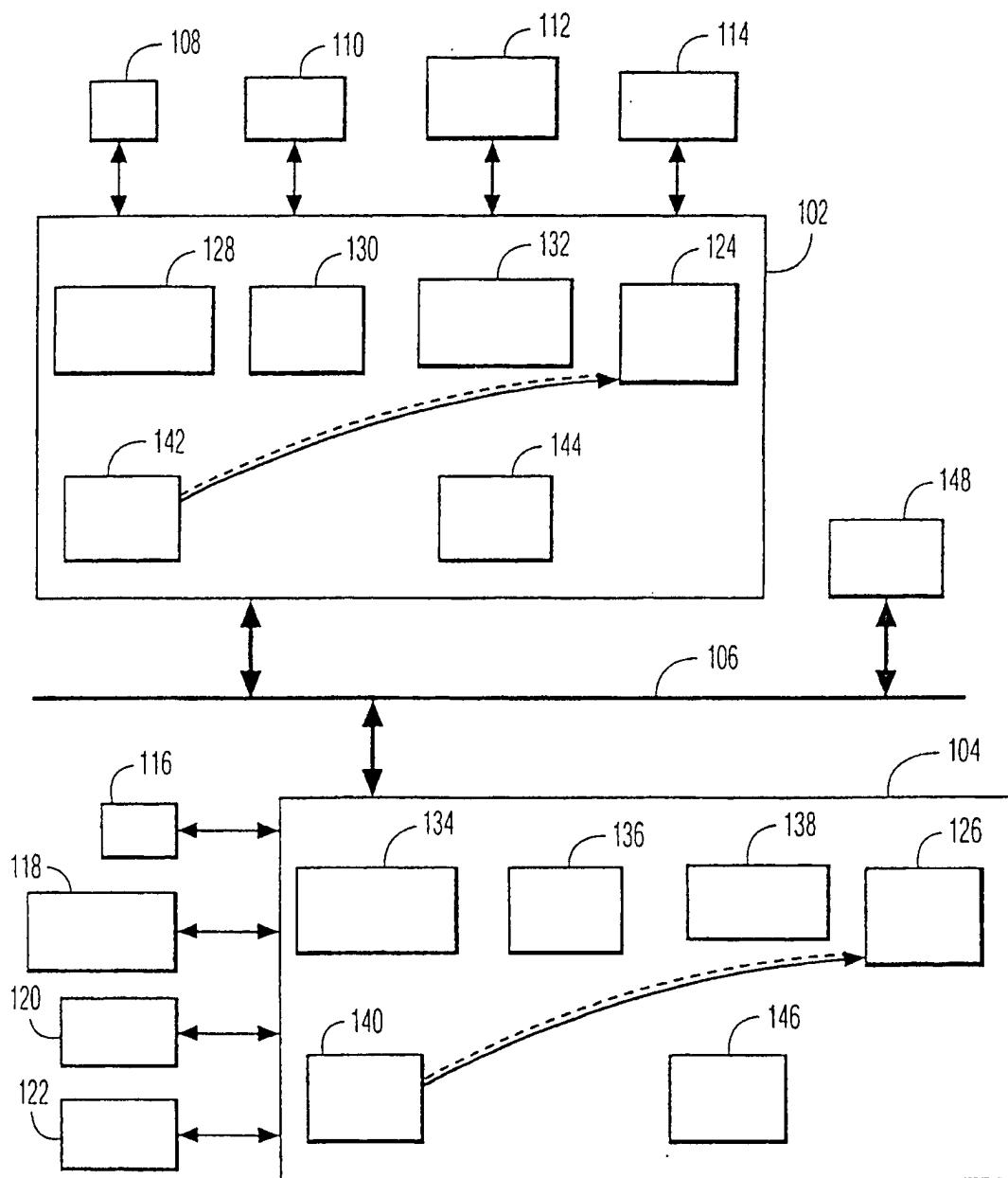
100

FIG. 5

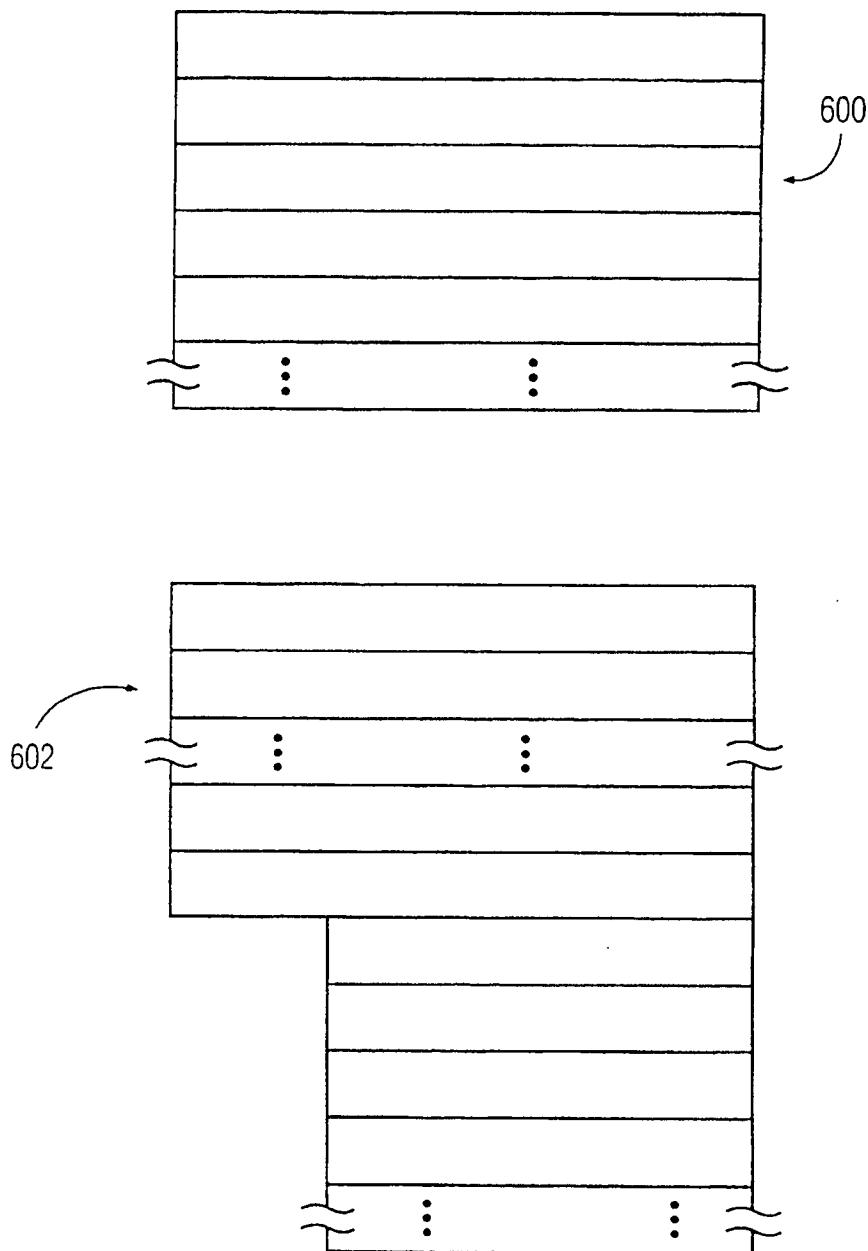


FIG. 6

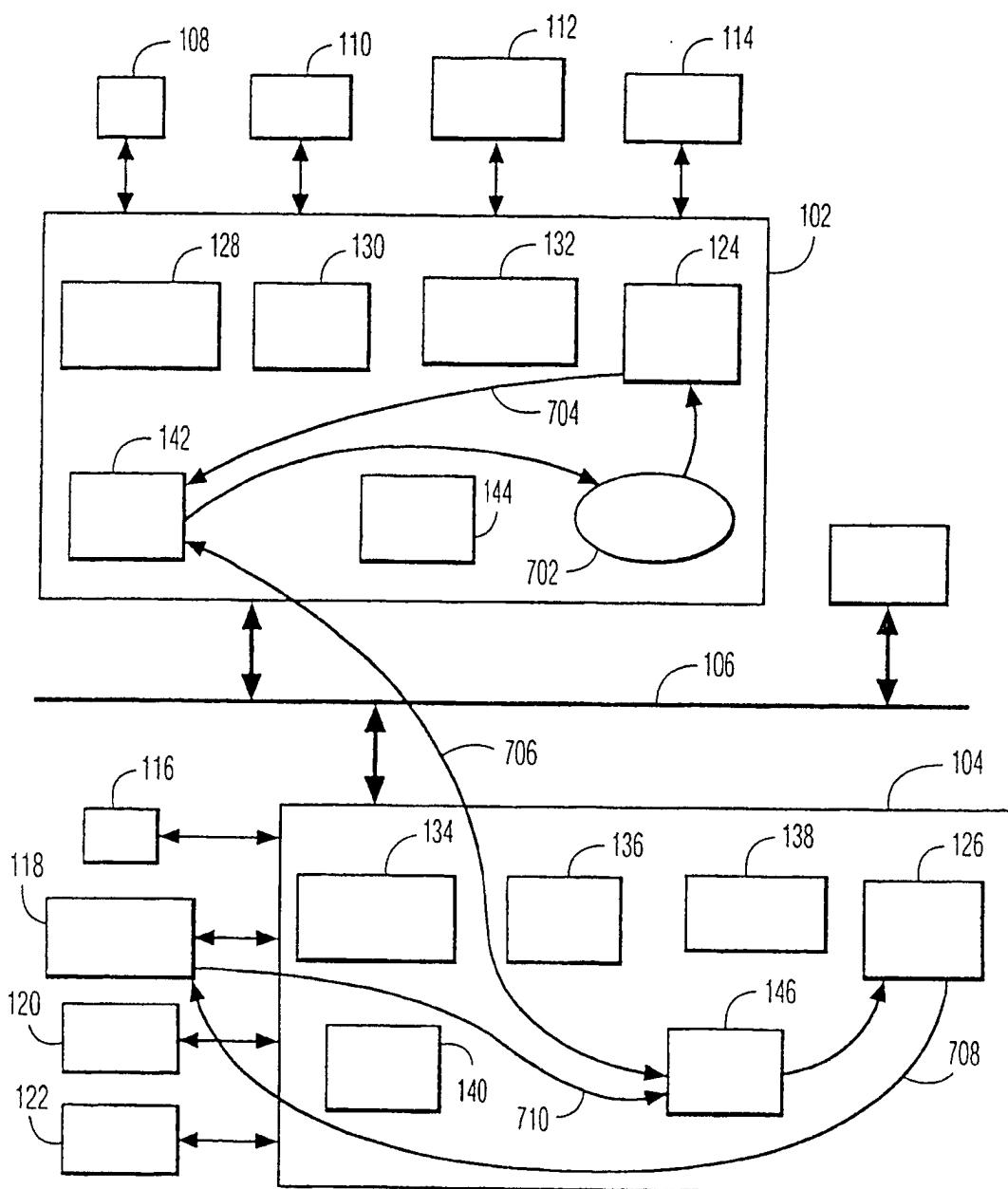


FIG. 7