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

(10) **DE 602 05 542 T2** 2006.06.22

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 271 885 B1**
(21) Deutsches Aktenzeichen: **602 05 542.3**
(96) Europäisches Aktenzeichen: **02 013 396.3**
(96) Europäischer Anmeldetag: **19.06.2002**
(97) Erstveröffentlichung durch das EPA: **02.01.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **17.08.2005**
(47) Veröffentlichungstag im Patentblatt: **22.06.2006**

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

(30) Unionspriorität:
896635 **29.06.2001** **US**

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

(73) Patentinhaber:
Nokia Corp., Nokia Group, FI

(72) Erfinder:
Forstadius, Antti, 33180 Tampere, FI

(74) Vertreter:
Eisenführ, Speiser & Partner, 80335 München

(54) Bezeichnung: **Verfahren und System für die Übertragung zwischen einem Client mit Kurzstrecken-Funk und einem entfernten Server**

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

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft allgemein eine Vorrichtung, ein Verfahren und ein System zum Weiterleiten von Information von einer drahtlosen Vorrichtung über ein Kommunikationsnetz. Insbesondere betrifft die vorliegende Erfindung eine Vorrichtung, ein Verfahren und ein System, welche ermöglichen, dass drahtlose Kurzstrecken-Hörfunkfrequenz-(„HF-") Punkt-zu-Punkt-Kommurtikationen quer durch ein Kommunikationsnetz überbrückt werden.

HINTERGRUND DER ERFINDUNG

NETZE

[0002] Netze sind allgemein so gedacht, dass sie aus dem Zusammenschalten und der Interoperation von Clients, Servern und Zwischenknoten in einer Graphentopologie bestehen. Es sollte erwähnt werden, dass der hier verwendete Begriff „Server" sich allgemein auf einen Computer, anderes Gerät, Software oder eine Kombination dieser bezieht, welche Anfragen von abgesetzten Nutzern über ein Kommunikationsnetz verarbeiten und beantworten. Server liefern ihre Information an anfragende „Clients". Ein Computer, anderes Gerät, Software oder eine Kombination dieser, welche Information und Anfragen erleichtern, verarbeiten und/oder den Durchlass von Information von einem Ausgangsnutzer zu einem Zielnutzer fördern, werden allgemein als „Knoten" bezeichnet. Netze sind allgemein dazu gedacht, die Informationsübertragung von Ausgangspunkten an Ziele zu ermöglichen. Es gibt viele Arten von Netzen, solche wie beispielsweise lokale Funknetze (Local Area Networks, nachfolgend als LANs bezeichnet), großräumige Netze (Wide Area Networks, nachfolgend als WANs bezeichnet), Piconetze usw.

INTERNET

[0003] Das Internet ist ein Netz aus Netzen. Es ist ein Zusammenschluss mehrerer und verschiedener Netze, welche zur Kommunikation mit einander eingerichtet sind. Diese durch das Internet bereitgestellte Anbindung und Zwischenkommunikationen werden in großen Teilen durch die Verwendung gemeinsamer Übertragungsprotokolle erleichtert.

[0004] Da die Verwendung des Internet steigt, steigt auch die Menge der über das Internet verfügbaren Information und/oder Dienste. Dieses macht das Internet zu einem wertvollen Informationstransportvehikel.

TRANSMISSION CONTROL PROTOCOL – INTERNETPROTOKOLL (TCP/IP)

[0005] Die Verbreitung und Expansion von Computersystemen, Datenbanken und Computernetzen wurde durch ein Zusammenschalten solcher Systeme und Netze in ein extraterritoriales Kommunikationsnetz, allgemein als das Internet bezeichnet, ermöglicht. Das Internet hat entwickelt und verwendet weitgehend das Transmission Control Protocol – Internetprotokoll (nachfolgend als TCP/IP bezeichnet). TCP/IP wurde in einem wissenschaftlichen Projekt des Departments of Defense (nachfolgend als DoD bezeichnet) entwickelt, um von mehreren und unterschiedlichen Netzanbietern erstellte Netze zusammenzuschließen als eine Grundlage für ein Netz aus Netzen, d.h. des Internet. Die Entwicklung von TCP/IP wurde zum Teil durch ein Erfordernis des DoD angetrieben, über ein Netz zu verfügen, welches selbst dann weiterarbeitet, wenn es während eines Kampfes beschädigt worden ist, somit der Information gestattend, um die beschädigten Teile des Kommunikationsnetzes herum an die Zieladresse weitergeleitet zu werden.

[0006] Das Internet ist ein Pakete vermittelndes Netz, und somit wird die Information in dem Internet in Stücke, als Pakete bezeichnet, aufgeteilt und in Paketform übertragen. Die Pakete enthalten IP-adressierende Informationen, Header genannt, welche von Routern verwendet werden, um das Leiten der Pakete von einem Ausgangspunkt zu einem Ziel über Zwischenknoten in dem Internet zu ermöglichen. Nach der Ankunft an dem Ziel werden die Pakete wieder zusammengesetzt, um die Originalnachricht zu bilden, und jegliche vermisste Pakete werden erneut angefragt.

[0007] Die IP-Komponente des Protokolls ist für das Weiterleiten der Informationspakete basierend auf einem vier-Byte-Adressierungsmechanismus verantwortlich; die Adresse wird in Form von vier durch Punkte getrennte Nummern geschrieben, wobei jede Nummer von 0 bis 255 reicht, bspw. „123.255.0.123". IP-Adressen werden von Internet-Direktionen und Registrierungsagenturen zugewiesen und sind eindeutig.

[0008] Der TCP-Anteil des Protokolls wird verwendet für das Verifizieren, dass Informationspakete von dem Ausgangspunkt durch den Zielcomputer richtig erhalten wurden, und falls nicht, zum erneuten Übertragen beschädigter Pakete. Andere die Zustellung nicht sicherstellende Übertragungssteuerungsprotokolle werden ebenfalls im Allgemeinen verwendet, solche wie User Datagram Protocol (nachfolgend als UDP bezeichnet). Das TCP-/IP-Protokoll ist in IEE/RFC1190 vom Januar 1991 spezifiziert.

OBJEKT-AUSTAUSCH- (OBEX-) PROTOKOLL

[0009] OBEX ist ein Sitzungsprotokoll und kann als ein binäres http-Protokoll beschrieben werden. Ein Beispiel einer OBEX-Serverimplementierung kann OpenOBEX sein, welches auf der Webseite sourceforge.net/projects/openobex gefunden werden kann.

[0010] Das OBEX-Protokoll und -Spezifikation können gefunden werden in: IOBEX, IrDA Objektaustauschprotokoll (Object Exchange Protocol), Counterpoint Systems Foundry, Inc., Microsoft Corporation, 18 März 1999 (Version 1.2).

BLUETOOTH-PROTOKOLL (BT)

[0011] Bluetooth ist eine drahtlose Technologie, welche in dem lizenzfreien Industrial-, Scientific-, Medical- (nachfolgend als ISM bezeichnet) Funkband von 2,4 GHz betrieben wird. Bluetooth-Technologie enthält eine Vielzahl von Protokollen, welche für Bluetooth freigegebenen Vorrichtungen gestattet, in einer Pikonetze bildenden Peer-To-Peer-Umgebung betrieben zu werden.

[0012] Albrecht M. u.a. „IP Services over Bluetooth: leading the way to new mobility", Local Computer Networks, 1999, IEEE Comput. Soc., US, 18 Oktober 1999, Seiten 2–11 offenbart ein System in einem BLUEtooth Public Access (nachfolgend als BLUEPAC bezeichnet) -Netz, in welchem für Bluetooth freigegebene Vorrichtungen die TCP-/IP-Protokolle verwenden können, um auf abgesetzte Server zuzugreifen. Durch die Bereitstellung der Unterstützung für lokale IP-Adressenzuweisung, Mobile IP und Handover kann das TCP-Protokoll ohne jegliche Modifikationen verwendet werden.

[0013] Ericsson u.a. "SyncML OBEX Binding", Netzseite von Syncml, 15 Juni 2001, Seiten 1–19 offenbart einen das OBEX-Protokoll enthaltenden Protokollstapel.

[0014] Das Bluetooth-Protokoll und -Spezifikation können gefunden werden in: Bluetooth System; Spezifikation, Bänder 1 und 2, „Core and Profiles": Version 1.1, 22-te, Februar 2001.

ZUSAMMENFASSUNG DER ERFINDUNG

[0015] Erfindungsgemäß werden ein Verfahren nach Anspruch 1, ein computerlesbares Medium nach Anspruch 6, ein System nach Anspruch 7 und eine OBEX-Bridge-Vorrichtung nach Anspruch 12 bereitgestellt.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0016] Die beigefügten Zeichnungen veranschaulichen bestimmte Ausführungsformen der Offenbarung.

[0017] Fig. 1 ist ein schematischer Überblick über ein für Kurzstrecken-HF freigegebenes und für einen Objektaustausch (OBEX) freigegebenes Client-Endgerät nach einer Ausführungsform der vorliegenden Erfindung;

[0018] Fig. 2 zeigt ausführlich den schematischen Überblick über eine für Kurzstrecken-HF freigegebene OBEX-Bridge nach einer Ausführungsform der vorliegenden Erfindung;

[0019] Fig. 3 ist ein schematischer Überblick über einen OBEX-Server nach einer Ausführungsform der vorliegenden Erfindung;

[0020] Fig. 4 zeigt eine OBEX-Client-Server-Bridge-Verbindung **400** nach einer Ausführungsform der vorliegenden Erfindung und stellt ein beispielhaftes Datenflussdiagramm bereit; und

[0021] Fig. 5 zeigt ausführlich ein Flussdiagramm eines beispielhaften Kommunikationsflusses zwischen einem beispielhaften OBEX-Client, einer beispielhaften OBEX-Bridge und einem OBEX-Server.

AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

FÜR KURZSTRECKEN-HF FREIGEGEBENES CLIENT-ENDGERÄT

[0022] Fig. 1 ist ein schematischer Überblick über ein für Kurzstrecken-HF und für einen Objektaustausch (OBEX) freigegebenes Clientendgerät (nachfolgend als „Client" bezeichnet) nach einer Ausführungsform der vorliegenden Erfindung.

Client-Systemisierung

[0023] In einer nicht beschränkenden, beispielhaften Ausführungsform kann der Client **101** einen Taktgeber **103**, einen Hauptprozessor (nachfolgend als CPU bezeichnet) **102**, einen Speicher **105**, eine Stromquelle **104**, Eingangs- **108**, **109** und Ausgangs- **110**, **111** (I/O) Komponenten **112**, **105**, **107** aufweisen. Die Stromquelle **104** stellt den Client-Strom bereit. Eine der I/O-Komponenten ist vorzugsweise ein Bluetooth-Chip **106**, wie etwa Cambridge Silicon Radio Inc.'s BlueCore IC, und ein Bluetooth-Sendeempfangsgerät **107**, welches imstande ist, Bluetooth-Protokollkommunikationen zu übertragen und zu empfangen. Es versteht sich, dass die Verwendung von Bluetooth-Komponenten/-Protokollen in der beispielhaften Ausführungsform eher veranschaulichend als beschränkend beabsichtigt ist und dass daher alternativ andere Kurzstrecken-Funkfrequenztechnologien verwendet werden können. Optional kann der Client auch andere drahtlose Protokollsendeempfangs-

geräte **112** verwenden, solche wie die für die zelluläre Telekommunikation verwendeten. Herkömmlich, aber nicht notwendigerweise, werden die Clientkomponenten alle zusammengeschaltet und/oder kommunizieren über einen Systembus **177**. Der Systemtaktgeber **103** hat typischer Weise einen Quarzoszillator und stellt ein Basissignal bereit. Der Taktgeber ist typischer Weise gekoppelt an den Systembus und verschiedene Mittel, welche die Basisbetriebsfrequenz für andere in dem Client zusammengeschaltete Komponenten erhöhen oder senken. Der Taktgeber und verschiedene Komponenten in dem Client steuern überall im Client Information enthaltende Signale. Solche Übertragung und solcher Empfang von Information enthaltenden Signalen überall im Client kann allgemein als Kommunikationen bezeichnet werden. Diese kommunikativen Signale können ferner übertragen, empfangen werden und die Ursache für die Rück- und/oder Antwortsignalkommunikation über den augenblicklichen Client hinaus an: Kommunikationsnetze, Eingangsvorrichtungen, Computersysteme (z.B. Server), Bridges, andere Clients, periphere Vorrichtungen und/oder ähnliche. Gewiss, jede von den oben aufgeführten Komponenten kann direkt miteinander verbunden, mit der CPU verbunden und/oder in zahlreichen Variationen organisiert sein, welche so verwendet werden, wie beispielhaft mittels verschiedener drahtloser und für Kurzstrecken-HF freigegebener Vorrichtungen erläutert, solcher wie, aber nicht auf solche beschränkt: zellulare Telefone, Handcomputer (Portable Digital Assitants, nachfolgend als PDA bezeichnet), Laptops und ähnliche. Optional kann der Client verschiedene Eingangs-/Ausgangsvorrichtungen enthalten, welche durch den Systembus und/oder direkt auch in Kommunikation mit der CPU angewendet werden. Solche Eingangsvorrichtungen können ein Mikrofon **108**, eine Eingabetastatur **109**, ein Sensorbildschirm (nicht abgebildet) und/oder ähnliche enthalten. Ausgangsvorrichtungen können einen LCD **110** und einen Lautsprecher **111** enthalten.

CPU

[0024] Die CPU **102** enthält zumindest einen Datenprozessor, geeigneten zum Ausführen von Programm-Modulen für das Ausführen von nutzer- und/oder systemgenerierten Anfragen. Die CPU kann ein Mikroprozessor sein, wie bspw. der Intel-Pentium-Prozessor und/oder ähnliche. Die CPU interagiert mit dem Speicher mittels Signalübergabe über leitfähige Kanäle, um gespeicherten Programmcode gemäß den herkömmlichen Datenverarbeitungstechniken auszuführen. Solche Signalübergabe erleichtert Kommunikation innerhalb der Kommunikationsnetze und darüber hinaus über verschiedene Schnittstellen.

Speicher

[0025] Es versteht sich, dass der Client verschiedene Arten von Speicher **105** verwenden kann. In einer typischen Konfiguration enthält der Speicher **105** ROM, RAM und möglicherweise eine befestigte Speichervorrichtung, z.B. eine Festplatte. Ebenso kann der Bluetooth-Chip **106** verschiedene Bluetooth-Protokolle innerhalb seines eigenen Speichers enthalten, welche entweder der CPU **102** und/oder dem Speicher **105** bereitgestellt werden können. Allgemein wird jede Mechanisierung und/oder jede Ausführungsform, welche einem Prozessor das Steuern des Speicherns und/oder das Beschaffen von Informationen gestatten, als Speicher **105** betrachtet. Daher benötigt und verwendet ein Client im Allgemeinen einen Speicher. Jedoch ist Speicher eine ersetzbare Technologie und Ressource, somit kann jegliche Anzahl von Speicherausführungsformen anstelle von oder miteinander gemeinsam verwendet werden.

Modulkollektion

[0026] Der Speicher **105** kann eine Kollektion von Programm-Modulen und/oder Daten enthalten, solche wie, aber nicht beschränkt auf: ein Betriebssystemmodul **130** (Betriebssystem); eine OBEX-Client-Anwendung **121**; Generic Object Exchange Profile(n)(nachfolgend als GOEP bezeichnet), welches auch als ein Bluetooth-Profil dienen kann; Speicherpuffer **123**; zellulare Kommunikationsprotokolle; Bluetooth-Protokollstapel **124**, weitere Kurzstrecken-Funkfrequenzprotokolle und/oder andere. Der Bluetooth-Protokollstapel kann einen Verbindungsmanager (Link Manager, nachfolgend als LM bezeichnet) **174**, ein Logical Link Control and Application Protocol (nachfolgend als L2CAP bezeichnet) **175**, ein Service Discovery Protocol (nachfolgend als SDP bezeichnet) **176**, RFCOMM **177** (d.h. ein Serial Line Emulation Protocol) und/oder ähnliche enthalten. Obwohl nichtherkömmliche Softwaremodule, solche wie in der Modulkollektion, typisch und vorzugsweise im Speicher **105** gespeichert werden, können diese auch in solchen Speicher geladen und/oder gespeichert werden, wie: periphere Vorrichtungen, ROM, abgesetzte Speichereinrichtungen über ein Kommunikationsnetz, verschiedene Speicherarten und/oder ähnliche.

Betriebssystem

[0027] Das Betriebssystemmodul **130** ist ein ausführbarer, das Betreiben des Clients erleichternder Programmcode. Typischer Weise erleichtert das Betriebssystem Zugriffe auf I/O, Netzwerkschnittstellen, periphere Vorrichtungen, Speichervorrichtungen und/oder ähnliche.

[0028] Das Betriebssystem kann auch das Interagieren des Nutzers mit dem Client gestattende Nut-

zerschnittstellenfunktionalität bereitstellen. Ein beispielhaftes Client-Betriebssystem ist Linux. Ein Betriebssystem kann auf und/oder mit anderen Modulen in einer Modulkollektion, sich selbst einschließend, und/oder ähnlichen Einrichtungen kommunizieren. Herkömmlich kommuniziert das Betriebssystem mit anderen Programmmodulen, Nutzerschnittstellen und/oder ähnlichen. Zum Beispiel kann das Betriebssystem Programmmodule, System-, Nutzer- und/oder Datenkommunikationen, Anfragen und/oder Antworten enthalten, kommunizieren, generieren, erhalten und/oder bereitstellen. Das Betriebssystem, einmal von der CPU durchgeführt, kann die Interaktion mit Kommunikationsnetzen, Daten, I/O, peripheren Vorrichtungen, Programmmodulen, Speicher, Nutzereingangsvorrichtungen und/oder ähnlichen freigeben. Vorzugsweise stellt das Betriebssystem Kommunikationsprotokolle bereit, welche dem Client das Kommunizieren mit anderen Einheiten über ein Kommunikationsnetz **144** gestatten. Verschiedene Kommunikationsprotokolle können von dem Client als Zwischenträgertransportmechanismus verwendet werden für das Interagieren mit anderen für Kurzstrecken-HF freigegebenen Vorrichtungen, solchen wie, aber nicht beschränkt auf: TCP/IP **122**, Bluetooth (d.h. mittels RFCOMM), OBEX **121** und/oder ähnliche.

Bluetooth-Protokolle

[0029] Im Speicher **105** werden verschiedene Bluetooth-Protokolle **124** und/oder andere Kurzstrecken-HF-Protokolle gespeichert. Die Bluetooth-Protokolle können ein Verbindungsmanagerprotokoll **174** (link manager protocol, nachfolgend als LM bezeichnet) enthalten. Die Verbindungsmanagersoftware läuft auf der CPU in dem Client, um Kommunikationen zwischen sich selbst und anderen Bluetooth-Vorrichtunen zu verwalten. Ein weiteres Protokoll ist das Service Discovery Protocol **176** (nachfolgend als SDP bezeichnet). Nach dem Verbinden eines Bluetooth-Clients mit einer weiteren Vorrichtung gibt das Service Discovery Protocol das Abfragen und die Identifizierung der Leistungsfähigkeiten weiterer Bluetooth-Vorrichtungen frei. Ein weiteres Protokoll ist das Logical Link Control and Adaptation Protocol **175** (nachfolgend als L2CAP bezeichnet). Das L2CAP stellt den Multiplexbetrieb, die Paketsegmentierung und das Wiederzusammensetzen von Daten bereit, wie es zwischen dem Client und weiteren für Bluetooth freigegebenen Vorrichtungen kommuniziert wird. Ein weiteres, im Speicher **105** enthaltenes Protokoll ist das RFCOMM, welches ein Bluetooth-Vorrichtungen zur Zusammenkommunikation über das Emulieren einer seriellen Leitung freigebendes serielles Leitungsemulationsprotokoll ist. Diese verschiedenen Protokolle interagieren, um von der CPU über einen Basisband **107** gegebene Daten zu codieren und zu decodieren. LM und L2CAP laufen direkt im obersten Teil vom Basisband **107**. RFCOMM und SDP laufen im obersten Teil von L2CAP.

[0030] Ferner befindet sich innerhalb des RAM **105** eine Objektaustausch- (nachfolgend als OBEX bezeichnet) Client-Anwendung **121**. Die Client-Anwendung erstellt OBEX-Anfragen basierend auf einer Nutzereingabe in die Client-Vorrichtung und erhält und interpretiert OBEX-Antworten von einem OBEX-Server **199**. OBEX ist ein Sitzungsschichtprotokoll, welches entwickelt wurde, um den Austausch von Objekten auf eine einfache und spontane Weise zu erlauben. OBEX gibt Objektaustauschdienste ähnlich zum in dem World Wide Web (dem Web) verwendeten HTTP frei, jedoch benötigt OBEX zum Laufen wesentlich weniger Ressourcen. Die OBEX-Client-Anwendung gibt den Client frei, Objekte auszutauschen und über das RFCOMM zu kommunizieren. Ferner kann der Speicher ein Generic Object Exchange Profile (nachfolgend als GOEP oder Profil bezeichnet) **125** enthalten. GOEP spezifiziert eine Weise, auf welche Bluetooth-Vorrichtungen Objektaustauschfähigkeiten einschließlich File Transfer Profils, Object Push Profils und Synchronisierungs-Profils unterstützen können. Die GOEP-Spezifikation erlaubt durch das Definieren der Zusammendurchführbarkeitsanforderungen für die Anwendungs- und die höheren Bluetooth-Protokollschichten (d.h. L2CAP, RFCOMM usw.) den Bluetooth-Vorrichtungen, zusammenbetrieben zu werden. Ferner kann der Speicher **105** einen Bereich bereitstellen, welches als ein Puffer **123** arbeitet.

[0031] Der Client **101** kann drahtlos **133** über das Kommunikationsnetz **144** kommunizieren, um letztendlich einen OBEX-Server **199** zu erreichen. Dieses wird durch eine OBEX-Bridge **150** erleichtert.

OBEX-BRIDGE FÜR KURZSTRECKEN-HF-KOMMUNIKATIONEN

[0032] Fig. 2 zeigt ausführlich den schematischen Überblick über eine für Kurzstrecken-HF freigegebene OBEX-Bridge (nachfolgend als Bridge bezeichnet) nach einer Ausführungsform der Erfindung.

OBEX-Bridge-Systemisierung

[0033] In einer nicht beschränkenden, beispielhaften Ausführungsform kann die Bridge **250** einen Taktgeber **203**, einen Hauptprozessor (nachfolgend als CPU bezeichnet) **202**, einen Speicher **205**, eine Stromquelle **204**, I/O-Komponenten **220**, **207** aufweisen. Die Stromquelle **204** stellt der Bridge Strom bereit. Eine der I/O-Komponenten ist ein Bluetooth-Chip **206**, solcher wie Cambridge Silicon Radio Inc.'s BlueCore IC, und ein Bluetooth-Sendeempfangsgerät **207**, welches imstande ist, Bluetooth-Protokollkommunikationen zu übertragen und zu empfangen. Wie oben in Verbindung mit „Client-Systemisierung" (Fig. 1) diskutiert, wird die Verwendung von Blue-

tooth-Komponenten/-Protokollen in der beispielhaften Ausführungsform eher veranschaulichend als beschränkend beabsichtigt und dass daher alternativ andere Kurzstrecken-Hörfunk-Frequenztechnologien verwendet werden können. Herkömmlich, obgleich nicht notwendiger Weise, sind die Client-Komponenten alle zusammengeschaltet und/oder kommunizieren über einen Systembus **277**. Der Systemtaktgeber **203** hat typischer Weise einen Quarzoszillator und stellt ein Basissignal bereit. Der Taktgeber ist typischer Weise an den Systembus und verschiedene Mittel gekoppelt, welche die Basisbetriebsfrequenz für andere in der Bridge zusammengeschaltete Komponenten erhöhen oder senken. Der Taktgeber und verschiedene Komponenten in dem Client steuern überall im Client Information enthaltende Signale. Solche Übertragung und solcher Empfang von Information enthaltenden Signalen überall im Client kann allgemein als Kommunikation bezeichnet werden. Diese kommunikativen Signale können ferner übertragen, empfangen werden und die Ursache sein für den Rücktransport- und/oder Antwortsignalkommunikation über augenblickliche Brücke hinaus an: Kommunikationsnetze, Eingangsvorrichtungen, Computersysteme (z.B. Server), periphere Vorrichtungen, Clients und/oder ähnliche. Gewiss, jede der oben aufgeführten Komponenten kann direkt miteinander verbunden, mit der CPU verbunden und/oder in zahlreichen Variationen organisiert sein, welche so verwendet werden, wie beispielhaft mittels verschiedener für Kurzstrecken-HF freigegebener Vorrichtungen erläutert. Optional kann der Client verschiedene Eingangs-/Ausgangsvorrichtungen enthalten, welche über den Systembus und/oder direkt auch in Kommunikation mit der CPU verwendet werden. Das Bluetooth-Modul **206** ermöglicht drahtlose Kommunikation über ein drahtloses Sendeempfangsgerät **207** während eine Netzschnittstelle **220**, wie etwa eine Ethernet-Karte, eine ISDN-Karte, eine DSL-Karte und/oder ähnliche, Kommunikationen **266** mit dem Kommunikationsnetz **244** freigibt. Somit überbrückt die Bridge **250** drahtlose Kurzstrecken-HF-Kommunikationen **233**, **207** mit Netzkommunikationen **220**, **266**, **244**.

CPU

[0034] Die CPU **102** enthält zumindest einen Datenprozessor, geeigneten zum Ausführen von Programm-Modulen für das Ausführen von nutzer- und/oder systemgenerierten Anfragen. Die CPU kann ein Mikroprozessor sein, wie etwa der Intel-Pentium-Prozessor und/oder ähnliche. Die CPU interagiert mit dem Speicher mittels Signalübergabe durch leitfähige Kanäle, um gespeicherten Programmcode gemäß der herkömmlichen Datenprozessierungstechniken auszuführen. Solche Signalübergabe erleichtert Kommunikation innerhalb der Kommunikationsnetze und darüber hinaus mittels verschiedener Schnittstellen.

Speicher

[0035] Es versteht sich, dass der Client verschiedene Arten von Speicher **205** verwenden kann. In einer typischen Konfiguration enthält der Speicher **205** ROM, RAM und möglicherweise eine befestigte Speichervorrichtung, z.B. eine Festplatte. Ebenso kann der Bluetooth-Chip **206** verschiedene Bluetooth-Protokolle innerhalb seines eigenen Speichers enthalten, welche entweder der CPU **102** und/oder dem Speicher **105** bereitgestellt werden können. Allgemein wird jede Mechanisierung und/oder jede Ausführungsform, welche einem Prozessor das Steuern des Speicherns und/oder das Beschaffen von Informationen gestatten, als Speicher **105** betrachtet. Daher benötigt und verwendet eine Bridge im Allgemeinen einen Speicher. Jedoch ist Speicher eine ersetzbare Technologie und Ressource, somit kann jegliche Anzahl von Speicherausführungsformen anstelle von oder miteinander gemeinsam verwendet werden.

Modulkollektion

[0036] Der Speicher **205** kann eine Kollektion von Programm-Modulen und/oder Daten enthalten, solche wie, aber nicht beschränkt auf: ein Betriebssystemmodul **230** (Betriebssystem); eine OB-EX-Bridge-Anwendung **221**; einen TCP-/IP-Stapel **222**; Generic Object Exchange Profile(n) (nachfolgend als GOEP bezeichnet), welches auch als ein Bluetooth-Profil dienen kann; Speicherpuffer **223**; zellulare Kommunikationsprotokolle; Bluetooth-Protokollstapel **224**, weitere Kurzstrecken-HF-Protokolle und/oder andere. Der Bluetooth-Protokollstapel kann einen Verbindungsmanager (Link Manager, nachfolgend als LM bezeichnet) **274**, ein Logical Link Control and Application Protocol (nachfolgend als L2CAP bezeichnet) **275**, ein Service Discovery Protocol (nachfolgend als SDP bezeichnet) **276**, RFCOMM **177** (d.h. ein Serial Line Emulation Protocol) und/oder ähnliche enthalten. Obwohl nichtherkömmliche Softwaremodule, solche wie in der Modulkollektion, typisch und vorzugsweise im Speicher **205** gespeichert werden, können diese auch in solchen Speicher geladen und/oder gespeichert werden, wie: periphere Vorrichtungen, ROM, abgesetzte Speichereinrichtungen über ein Kommunikationsnetz, verschiedene Speicherarten und/oder ähnliche.

Betriebssystem

[0037] Das Betriebssystemsmodul **230** ist ein ausführbarer, das Betreiben der Bridge erleichternder Programmcode. Typischer Weise erleichtert das Betriebssystem Zugriffe auf I/O, Netzwerkschnittstellen, periphere Vorrichtungen, Speichervorrichtungen und/oder ähnliche. Das Betriebssystem kann auch das Interagieren des Administratoren mit der Bridge gestattende Nutzerschnittstellenfunktionalität bereitstellen. Ein beispielhaftes Bridge-Betriebssystem ist

Linux. Ein Betriebssystem kann auf und/oder mit anderen Modulen in einer Modulkollektion, sich selbst einschließend, und/oder ähnlichen Einrichtungen kommunizieren. Herkömmlich kommuniziert das Betriebssystem mit anderen Programmmodulen, Nutzerschnittstellen und/oder ähnlichen. Zum Beispiel kann das Betriebssystem Programmmodul-, System-, Nutzer- und/oder Daten-Kommunikationen, Anfragen und/oder Antworten enthalten, kommunizieren, generieren, erhalten und/oder bereitstellen. Das Betriebssystem, einmal von der CPU durchgeführt, kann die Interaktion mit Kommunikationsnetzen, Daten, I/O, peripheren Vorrichtungen, Programmmodulen, Speicher, Nutzereingangsvorrichtungen und/oder ähnlichen freigeben. Vorzugsweise stellt das Betriebssystem Kommunikationsprotokolle bereit, welche dem Client das Kommunizieren mit anderen Einheiten über ein Kommunikationsnetz **244** gestatten. Verschiedene Kommunikationsprotokolle können von der Bridge als Zwischenträgertransportmechanismus verwendet werden für das Interagieren mit anderen für Kurzstrecken-HF freigegebenen Vorrichtungen, solchen wie, aber nicht beschränkt auf: Bluetooth (d.h. mittels RFCOMM), Mehrpunktkommunikation (Multicast), OBEX **121**, TCP/IP **122**, UDP, Punkt-Punkt-Kommunikation (Unicast) und/oder ähnliche. Der TCP-/IP-Stapel gibt TCP-/IP-Kommunikationen über die Netzschnittstelle **220** frei.

Bluetooth-Protokolle

[0038] Im Speicher **205** werden verschiedene Bluetooth-Protokolle **224** und/oder andere Kurzstrecken-HF-Protokolle gespeichert. Die Bluetooth-Protokolle können ein LM-Protokoll **274**, SDP **276**, L2CAP **275**, RFCOMM **277** und ein Basisband enthalten, wie es bereits oben diskutiert worden ist. Die Verbindungsmanagersoftware läuft auf der CPU in der Bridge, um Kommunikationen zwischen sich selbst und anderen Bluetooth-Vorrichtunen, solchen wie Clients **101**, zu verwalten. Nach dem Verbinden einer Bluetooth-Bridge mit einer weiteren Vorrichtung gibt das Service Discovery Protocol das Abfragen und die Identifizierung der Leistungsfähigkeiten weiterer Bluetooth-Vorrichtungen frei. Diese verschiedenen Protokolle interagieren, um von der CPU über einen Basisband **207** gegebene Daten zu codieren und zu decodieren. LM und L2CAP laufen direkt im obersten Teil vom Basisband **207**. RFCOMM und SDP laufen im obersten Teil von L2CAP.

[0039] Ferner befindet sich innerhalb des Speichers **205** eine OBEX-Bridge-Anwendung **221**. Die Bridge-Anwendung erhält Informationen von OBEX-Clients **101** und leitet diese ferner weiter an OBEX-Server **299**.

[0040] Die OBEX-Bridge **250** kommuniziert drahtlos **233** über das Bluetooth-Sendeempfangsgerät **207** mit einem Client **101**. Ferner wird die OBEX-Bridge über die Netzschnittstelle **220** zur Kommunikation **266** mit einem Kommunikationsnetz **244** eingerichtet, welches Kommunikationen **298** mit einem OBEX-Server **299** freigibt.

OBEX-SERVER

[0041] Fig. 3 ist ein schematischer Überblick über einen OBEX-Server nach einer Ausführungsform der Erfindung.

OBEX-Server-Systemisierung

[0042] In einer nicht beschränkenden, beispielhaften Ausführungsform kann der OBEX-Server **399** einen Taktgeber **303**, einen Hauptprozessor (nachfolgend als CPU bezeichnet) **302**, einen Speicher **305**, eine Stromquelle **304**, I/O-Komponenten **380**, **381**, **382**, **320** aufweisen. Die Stromquelle **304** stellt dem OBEX-Server Strom bereit. Die I/O-Schnittstelle **380** kann jede Anzahl von Bussen sein, welche das Zusammenshalten von peripheren und Eingangs-, Ausgangs-Vorrichtungen gewährleisten, solchen wie, aber nicht beschränkt auf, eine Tastatur **380**, ein Monitor **381**, eine Netzschnittstelle **320** und/oder ähnlichen. Herkömmlich, aber nicht notwendigerweise, werden die Clientkomponenten alle zusammengeschaltet und/oder kommunizieren über einen Systembus **377**. Der Systemtaktgeber **303** hat typischer Weise einen Quarzoszillator und stellt ein Basissignal bereit. Der Taktgeber ist typischer Weise gekoppelt an den Systembus und verschiedene Mittel, welche die Basisbetriebsfrequenz für andere in der Bridge zusammengeschatete Komponenten erhöhen oder senken. Der Taktgeber und verschiedene Komponenten in dem Client steuern überall im Client Information enthaltende Signale. Solche Übertragung und solcher Empfang von Information enthaltenden Signalen überall im Client kann allgemein als Kommunikation bezeichnet werden. Diese kommunikativen Signale können ferner übertragen, empfangen werden und die Ursache sein für den Rücktransport- und/oder Antwortsignalkommunikation über augenblicklichen Client hinaus an: Kommunikationsnetze, Eingangsvorrichtungen, weitere Computersysteme (z.B. Server), periphere Vorrichtungen, Bridges und/oder ähnliche. Gewiss, jede von den oben aufgeführten Komponenten kann direkt mit einander verbunden, mit der CPU verbunden und/oder in zahlreichen Variationen organisiert sein, welche so verwendet werden, wie beispielhaft mittels verschiedener Computersysteme und Server erläutert. Optional kann der Client verschiedene Eingangs-/Ausgangsvorrichtungen enthalten, welche durch den Systembus und/oder direkt auch in Kommunikation mit der CPU angewendet werden. Die Netzschnittstelle **320** kann eine Ethernet-Karte, eine ISDN-Karte, eine DSL-Karte und/oder ähnliches sein, welche Kommunikationen **398** mit dem Kommunikationsnetz **344** freigeben.

CPU

[0043] Die CPU **102** enthält zumindest einen Datenprozessor, geeigneten zum Ausführen von Programm-Modulen für das Ausführen von nutzer- und/oder systemgenerierten Anfragen. Die CPU kann ein Mikroprozessor sein, wie bspw. der Intel-Pentium-Prozessor und/oder ähnliche. Die CPU interagiert mit dem Speicher mittels Signalübergabe über leitfähige Kanäle, um gespeicherten Programmcode gemäß den herkömmlichen Datenverarbeitungstechniken auszuführen. Solche Signalübergabe erleichtert Kommunikation innerhalb der Kommunikationsnetze und darüber hinaus über verschiedene Schnittstellen.

Speicher

[0044] Es versteht sich, dass der OBEX-Server verschiedene Arten von Speicher **305** verwenden kann. In einer typischen Konfiguration enthält der Speicher **305** ROM, RAM und eine befestigte Speichervorrichtung **340**, z.B. eine Festplatte. Allgemein wird jede Mechanisierung und/oder jede Ausführungsform, welche einem Prozessor das Steuern des Speicherns und/oder das Beschaffen von Informationen gestatten, als Speicher **305** betrachtet. Daher benötigt und verwendet ein OBEX-Server im Allgemeinen einen Speicher. Jedoch ist Speicher eine ersetzbare Technologie und Ressource, somit kann jegliche Anzahl von Speicherausführungsformen anstelle von oder miteinander gemeinsam verwendet werden.

Modulkollektion

[0045] Der Speicher **305** kann eine Kollektion von Programm-Modulen und/oder Daten enthalten, solche wie, aber nicht beschränkt auf: ein Betriebssystemmodul **330** (Betriebssystem); eine OBEX-Client-Anwendung **321**; einen TCP-/IP-Stapel **322**; und einen OBEX-Datei-Inhalt **341**, welche nach Bedarf in der Speichervorrichtung **340** gespeichert und im Speicher **305** abgefragt werden können.

OBEX-Server-Anwendung

[0046] Ein OBEX-Server-Anwendungsmodul **321** ist ein gespeicherter Programmcode, welcher von der CPU ausgeführt wird. Der OBEX-Server kann ein herkömmlicher OBEX-Informationsserver sein, wie etwa, aber nicht beschränkt auf, OpenOBEX. Optional gestattet der Informationsserver die Ausführung von Programmmodulen durch solche Einrichtungen wie C++, Java, JavaScript, ActiveX, Common Gateway Interface (nachfolgend als CGI bezeichnet) Skripte, Active Server Page (ASP) und ähnliche. Die Ausführung dieser Programmmodule wird allgemein benötigt, um dynamischen Inhalt zu produzieren (äquivalent zur Verwendung in Web-Servern). Optional unterstützt der Informationsserver sichere Kommunikationsprotokolle, solche wie, aber nicht beschränkt auf, File Transfer Protocol (nachfolgend als FTP bezeichnet); Hyper Text Transfer Protocol (nachfolgend als HTTP bezeichnet), Object Exchange (OBEX)-Protokoll, Secure Hypertext Transfer Protocol (nachfolgend als HTTPS bezeichnet), Secure Socket Layer (nachfolgend als SSL bezeichnet) und ähnliche. Herkömmlich stellt ein OBEX-Server Ergebnisse in Form von Austauschobjekten bereit und gestattet manipulierte Generierung von Austauschobjekten über Interaktion mit anderen Programmmodulen. Ein OBEX-Server kann an und/oder mit anderen Modulen in einer Modulkollektion, einschließlich sich selbst, und/oder ähnlichen Einrichtungen kommunizieren. Am häufigsten kommuniziert der OBEX-Server mit Betriebssystemen, anderen Programmmodulen, Nutzerschnittstellen und/oder ähnlichen. Ein Informationsserver kann Programmmodul-, System-, Nutzer- und/oder Daten-Kommunikationen, Anfragen und/oder Antworten enthalten, kommunizieren, generieren, erhalten und/oder bereitstellen.

Betriebssystem

[0047] Das Betriebssystemmodul **330** ist ein ausführbarer, das Betreiben des OBEX-Servers erleichternder Programmcode. Typischer Weise erleichtert das Betriebssystem Zugriffe auf I/O, Netzwerkschnittstellen, periphere Vorrichtungen, Speichervorrichtungen und/oder ähnlichen. Das Betriebssystem kann auch das Interagieren eines Nutzers mit der Bridge gestattende Nutzerschnittstellenfunktionalität bereitstellen. Vorzugsweise ist das Betriebssystem ein herkömmliches Produkt, solches wie Apple Macintosh OS X Server, AT&T Plan 9, Microsoft Windows NT Server, Unix und/oder ähnliche Betriebssysteme. Vorzugsweise ist das Betriebssystem sehr fehlertolerant, skalierbar und sicher. Ein Betriebssystem kann auf und/oder mit anderen Modulen in einer Modulkollektion, sich selbst einschließend, und/oder ähnlichen Einrichtungen kommunizieren. Herkömmlich kommuniziert das Betriebssystem mit anderen Programmmodulen, Nutzerschnittstellen und/oder ähnlichen. Zum Beispiel kann das Betriebssystem Programmmodul-, System-, Nutzer- und/oder Daten-Kommunikationen, Anfragen und/oder Antworten enthalten, kommunizieren, generieren, erhalten und/oder bereitstellen. Das Betriebssystem, einmal von der CPU ausgeführt, kann die Interaktion mit Kommunikationsnetzen, Daten, I/O, peripheren Vorrichtungen, Programmmodulen, Speicher, Nutzereingangsvorrichtungen und/oder ähnlichen freigeben. Vorzugsweise stellt das Betriebssystem Kommunikationsprotokolle bereit, welche dem Client das Kommunizieren mit anderen Einheiten über ein Kommunikationsnetz **344** gestatten. Verschiedene Kommunikationsprotokolle können von der Bridge als Zwischenträgertransportmechanismus verwendet werden für das Interagieren mit anderen Vorrichtungen, solchen wie, aber nicht beschränkt auf: Mehrpunkt-

kommunikation (Multicast), OBEX **121**, TCP/IP **122**, UDP, Punkt-Punkt-Kommunikation (Unicast) und/oder ähnlichen. Der TCP-/IP-Stapel gibt TCP-/IP-Kommunikationen über die Netzschnittstelle **320** frei.

[0048] Der OBEX-Server wird in Kommunikation **398** mit einem Kommunikationsnetz **344** über die Netzschnittstelle **320** angewendet. Dieses gibt Kommunikationen **366** mit der OBEX-Bridge **250** frei, welche weiterhin in Kommunikation **333** mit einem OBEX-Client **101** angewendet wird.

OBEX-CLIENT-BRIDGE-SERVER-VERBINDUNG

[0049] Fig. 4 zeigt eine OBEX-Client-Server-Bridge-Verbindung **400** nach einer Ausführungsform der Erfindung und stellt ein beispielhaftes Datenflussdiagramm bereit. Die Datenflüsse treten zwischen einem OBEX-Client **401**, einer OBEX-Bridge **450** und einem OBEX-Server **499** auf. Der OBEX-Client ist im Inneren der Umgebung der OBEX-Bridge angekommen, um eine Kurzstrecken-HR-Kommunikationsverbindung **433** solche wie Bluetooth, über das Basisband **406** einzurichten. Der OBEX-Client führt eine OBEX-Client-Anwendung **121** aus. Die OBEX-Client-Anwendung erstellt bestimmte Datenanfragen im OBEX-Format. Diese Anfragen werden von den Kurzstrecken-HF-Protokollen, solchen wie Bluetooth, anwendenden CPU des Clients in RFCOMM-Datenblöcke **423a** verarbeitet. Der Verbindungsmanager (Link Manager) **428a** und das L2CAP **475a** helfen, die RFCOMM-codierten OBEX-Nachrichten anzunehmen, und stellen diese einem Basisband-Sendeempfangsgerät **406a** als Daten bereit, welches **450** durch das Aufstellen einer Kommunikation zu dem Basisband-Sendeempfangsgerät **406a** der OBEX-Bridge über Bluetooth-Protokolle mit einer OBEX-Bridge kommuniziert. Die erhaltenen Datensignale folgen einer umgekehrten Route in die OBEX-Bridge; die von dem OBEX-Client **401** durch die OBEX-Bridge **450** in der Basisband-Einrichtung **406b** der OBEX-Bridge erhaltenen HF-Kommunikationen werden dann von dem LM **478b** und dem L2CAP **475b** über das RFCOMM **423b** in ein serielles Datenformat decodiert, was zu der ursprünglichen OBEX-Nachricht von dem Client **401** führt.

[0050] Für jeden eine Verbindung **433** aufstellenden OBEX-Client erstellt die OBEX-Bridge **450** eine Handhabe auf die über einen RFCOMM **423b**-Kanal bereitgestellte Kommunikationen (typischer Weise ist die Handhabe ein Zeiger, welcher auf einen Speicherbereich eines Speicherbuffers **223** verweist) und weist und ordnet dieser Handhabe eine interne, eindeutige IP-Adresse zu. Auf diese Weise erstellt und wartet die OBEX-Bridge eine Tabellendatenstruktur mit zumindest zwei Spalten (Bridge-Tabelle) **489**, welche eine Handgabe auf einen von einer Client-Kommunikationsverbindung aufgestellten RFCOMM-Kanal (Client-RFCOMM-Handhabe) **487** und auf eine zugeordnete, generierte interne IP-Adresse (Client-IP-Adresse) **488** aufweist. Die Bridge-Tabelle kann durch Verwendung verschiedener Standarddatenstrukturen implementiert werden, wie die eines Arrays, einer Hash-Struktur, einer (verketteten) Liste, einer Struktur und/oder ähnlicher. Solche Datenstrukturen können im Speicher und/oder in (strukturierten) Dateien gespeichert werden.

[0051] Die OBEX-Bridge-Anwendung **421** erhält dann die ursprünglich von dem Client **121** mit Client-RFCOMM-Handhaben generierte OBEX-Datenanfrage (für von einem bestimmten Client **410** erhaltenen Daten). Nach dem Erhalt der OBEX-Datenanfrage von dem Speicherbuffer kapselt die OBEX-Bridge-Anwendung **421** dann die OBEX-Datenanfrage in TCP-/IP-Pakete ein. Diese Pakete werden dann an eine TCP-/IP-Schicht **422a** gesendet, welche ferner die TCP-/IP-Nachrichten als eine Datenverbindung **407a** bereitstellt, welche schließlich über ein physikalisches Trägermedium, wie etwa Ethernet **420a**, gesendet wird. Beim Generieren der TCP-/IP-Paketköpfe kennzeichnet die OBEX-Bridge die ursprüngliche IP-Adresse als die Client-IP-Adresse **488**, welche generiert und mit einer bestimmten Client-RFCOMM-Handhabe **487** verknüpft wurde; die OBEX-Bridge-Anwendung kann das so durch Sichten der Bridge-Tabelle **489** erreichen. Beim Bestimmen der Ziel-IP-Adresse für die OBEX-Datenanfrage kann die OBEX-Bridge-Anwendung **421** eine interne Look-Up-Liste des OBEX-Servers konsultieren. Diese Look-Up-Liste beinhaltet IP-Adressen. Die Liste kann einen einzelnen Eintrag oder einer Vielzahl von IP-Adressen enthalten. Die Wahl, welcher OBEX-Server aus der Look-Up-Liste ausgewählt wird, kann durch beliebige Anzahl von lastausgleichenden und/oder kontextsensitiven Heuristiken geregelt werden. Zum Beispiel, basierend auf dem GOEP-Profil des Clients und dem in dem File Transfer Profile spezialisierten OBEX-Server kann ein Object-Push-Profil oder ein Synchronisierungs-Profil spezifiziert werden. In einer weiteren nicht beschränkenden beispielhaften Ausführungsform kann ein OBEX-Server mit der niedrigsten Nutzlast verwendet werden. Die Look-Up-Liste kann implementiert werden durch Verwendung verschiedener Standardstrukturen, wie etwa ein Array, eine Hash-Struktur, eine (verkettete) Liste, eine Struktur und/oder ähnliches. Solche Datenstrukturen können im Speicher und/oder in (strukturierten) Dateien gespeichert werden. Nach dem Kennzeichnen der Herkunfts- und der Ziel-IP-Adresse in den entstandenen TCP-/IP-Paketen werden die TCP-/IP-Nachrichten von der OBEX-Bridge über die Netzschnittstelle **420a** der OBEX-Bridge kommuniziert **466**. Die OBEX-Nachrichten werden dann von der OBEX-Bridge **421** über ein Kommunikationsnetz **344** an eine Netzschnittstelle **420b** auf dem OBEX-Server **499** kommuniziert **466**. Diese Nachrichten werden der Reihe nach der Da-

tenverbindung **407b** des OBEX-Servers **499** bereitgestellt und in der Form von TCP-/IP-Paketen **422b** bereitgestellt, welche dann von der Anwendungssoftware **221** des OBEX-Servers interpretiert werden. Die OBEX-Server-Anwendung **221** kann dann die OBEX-codierten Daten und Anfragen betreiben und Ergebnisse und Antworten über die Bridge **450** zurück an den Client **401** bereitstellen. Auf diese Weise nimmt das OBEX-Antwortssignal von der OBEX-Server-Anwendung **221** den umgekehrten Weg, von welchem es herrührte.

[0052] Auf diese Weise wird die Verbindung zwischen einem OBEX-Client und einer OBEX-Bridge **433**, welche über HF-Signale des Basisbands **406** erfolgt, durch einen konzeptuellen RFCOMM-Kanal **423**, **433b** bereitgestellt. Ferner wird die TCP-Verbindung zwischen der OBEX-Bridge und dem OBEX-Server **466**, welche über Netzschnittstellen **420** und ein Kommunikationsnetz **344** erfolgt, durch eine konzeptuelle TCP-/IP-Verbindung **466b** bereitgestellt. Somit bildet er RFCOMM-Kanal zwischen dem Client und der Bridge plus die TCP-/IP-Verbindung zwischen der Bridge und dem Server eine konzeptuelle OBEX-Verbindung **400**. Es ist wichtig anzumerken, dass diese OBEX-Verbindung über ein Kommunikationsnetz stattfinden kann und dass der OBEX-Server nicht in der Umgebung des OBEX-Clients platziert werden muss, vielmehr können Kommunikationen über die OBEX-Bridge über ein Kommunikationsnetz, wie etwa das Internet, stattfinden.

OBEX-SERVER

[0053] Fig. 5 zeigt ausführlich ein Flussdiagramm eines beispielhaften Kommunikationsflusses zwischen einem OBEX-Client, einer OBEX-Bridge und einem OBEX-Server. Die Box **501** zeigt, dass eine OBEX-Client-Anwendung instanziert ist. Die Anwendung kann durch aufrufende Befehle über eine Eingangstastatur oder -Sprache oder andere Eingabemittel instanziert werden, welche die Ausführung der Client-Anwendung durch die CPU des Clients bewirken. In einem nicht beschränkenden Beispiel wird die OBEX-Client-Anwendung instanziert, sobald der Client eingeschaltet ist. Nach dieser Instanzierung der OBEX-Client-Anwendung zeigt die Box **502** den Client eine Kurzstrecken-HF-Verbindung, wie etwa Bluetooth, mit der OBEX-Bridge so aufbauend, wie es in der Fig. 4 diskutiert wurde.

[0054] Die Box **503** zeigt, dass der Client nach dem Aufbauen der Kommunikationen zwischen dem Client und der OBEX-Bridge-Anwendung das Dienstermittlungsprotokoll verwendet, um weitere Fähigkeiten von Kurzstrecken-HF-Vorrichtungen und die zugehörigen GEOP-Parameter abzufragen (in diesem Fall den Access Point (nachfolgend als AP bezeichnet), d.h. den eine OBEX-Bridge enthaltenden AP, welche die Bridge-Anwendung enthält). Die Box **504** zeigt, dass die OBEX-Bridge-Anwendung **421** mittels der Verwendung eines Dienstermittlungsprotokolls antwortet, wobei sie die für das GOEP-Profil benötigten Parameter ausgibt. Die Box **505** zeigt den Client Protokolle aufbauend, basierend auf dem von dem Server über die OBEX-Bridge erhaltenen GOEP-Profil. Die Box **506** zeigt, dass ein RFCOMM-Kanal zwischen dem Client und der OBEX-Bridge-Applikation aufgebaut wird.

[0055] Die Box **507** zeigt, dass die Bridge nach dem Aufbau eines RFCOMM-Kanals zwischen dem Client und der Bridge intern eine IP-Adresse zuweist und diese an eine Client-RFCOMM-Handhabung bindet. Die Box **508** zeigt, dass nach dem Generieren und Zuordnen einer internen IP-Adresse der Client-RFCOMM-Handhabung, dass die OBEX-Bridge danach eine TCP-/IP-Verbindung mit dem abgesetzten OBEX-Server aufstellt. Die Box **509** zeigt, dass nach der Aufstellung einer TCP-/IP-Verbindung mit dem OBEX-Server der OBEX-Client dann das Versenden von Daten an die OBEX-Bridge über die RFCOMM-Verbindung beginnen kann. Die Box **510** zeigt, dass die Bridge Daten von dem Client über die RFCOMM-Verbindung erhält und diese in TCP-/IP-Pakete einkapselt. Daher leitet die OBEX-Bridge-Anwendung **421** Daten von den RFCOMM-Kommunikationen **433b** des Clients an das TCP-/IP-Socket in der OBEX-Bridge weiter. Diese TCP-/IP-eingekapselten Pakete werden an die Netzschnittstelle **420a** der OBEX-Bridge gesendet. Nachdem die OBEX-Bridge-Anwendung **421** die Daten in TCP-/IP-Pakete codiert hat, werden die Daten an den OBEX-Server gesendet, wobei eine OBEX-Server-Anwendung **221** danach die Daten erhält und die OBEX-Anfragen bearbeitet. Jegliche Antworten von dem OBEX-Server werden zurück an die Bridge über TCP-/IP-Nachrichten über die OBEX-Server-Netzschnittstelle **420b** gesendet.

[0056] Die Box **511** zeigt, dass die Antworten von dem OBEX-Server **499**, die für den OBEX-Client **401** bestimmt sind, von der OBEX-Bridge **450** in der entsprechenden TCP-/IP-Adresse erhalten werden, welche von der OBEX-Bridge generiert wurde. Die OBEX-Bridge verwendet die Bridge-Tabelle **489**, um die mit der die Kommunikationen von dem OBEX-Server enthaltenden Client-IP-Adresse **488** verknüpfte Client-RFCOMM-Handhabe **487** nachzusehen. Als nächstes kapselt die Bridge die in der Form von TCP-/IP-Paketen von dem OBEX-Server erhaltenen Daten ein und leitet diese Daten von dem TCP-/IP-Socket an die RFCOMM-Handhabe weiter. Diese Weiterleitung wird mittels einer Look-Up-Tabelle durchgerührt, welche nach der Erstellung einer RFCOMM-Handhabe eingerichtet wurde, wie in der Box **507** und in Fig. 4 dargestellt ist. Somit agiert die Bridge-Tabelle **489** als eine Weiterleitungstabelle, um sicherzustellen, dass Information richtig zwischen jeder Client-Datenanfrage und allen entsprechenden

Ergebnissen von einem OBEX-Server fließt. Dies ist wichtig in einer Umgebung, in der für mehrfach OBEX- und Kurzstrecken-HF freigegebene Clients vorhanden sind, welche für Kommunikationen mit der OBEX-Bridge vorgesehen sind. Somit wird die von dem OBEX-Server erhaltene TCP-/IP-Information nach dem Bereitstellen eines Look-Up für die korrekte RFCOMM-Handhabe in RFCOMM-Pakete eingekapselt. Die OBEX-Bridge sendet die RFCOMM-codierten Daten über einen drahtlosen Basisbandkanal zurück an den Client.

[0057] Die Box **512** zeigt, dass der Client die Daten über den RFCOMM-Kanal erhält. Die Box **513** zeigt, dass Boxen **509**, **510**, **511** und **512** wiederholt werden können, bis sich der Client von der RFCOMM-Verbindung oder der Kurzstrecken-HF-Verbindung (z.B. Bluetooth) trennt.

[0058] Die Box **514** zeigt, dass die Bridge die TCP-/IP-Verbindung zu dem abgesetzten Server beendet und die zugewiesene IP-Adresse freigibt.

[0059] Gemäß einer Ausführungsform eines Aspektes der Erfindung wird ein Verfahren der drahtlosen Kommunikation zwischen einem OBEX-Client und einem OBEX-Server über ein Kommunikationsnetz bereitgestellt, welches aufweist:
Empfangen OBEX-codierter Nachrichten;
Aufstellen eines Basisbandkommunikationskanals durch Verwendung eines Bluetooth-GOEP-Profils zwischen dem OBEX-Client und einer für Bluetooth freigegebenen OBEX-Bridge;
Einkapseln der OBEX-codierten Nachrichten in Basisbandnachrichten;
Bereitstellen der im Basisband eingekapselten Nachrichten der OBEX-Bridge;
Zuweisen einer internen IP-Adresse innerhalb der OBEX-Bridge;
Binden der IP-Adresse an den Basisbandkommunikationskanal;
Aufbauen eines TCP-/IP-Basisbandkommunikationskanals zwischen die OBEX-Bridge und einen abgesetzten Server; und
Weiterleiten der OBEX-codierten Nachrichten von dem Basisbandkanal an den TCP-/IP-Kanal.

[0060] Vorteilhafter Weise umfasst das Verfahren ferner das Empfangen OBEX-codierter Datenantworten von dem abgesetzten Server über den TCP-/IP-Kanal; und das Weiterleiten der OBEX-codierten Datenantworten an den Basisbandkanal mit Bezug auf die IP-Adresse.

[0061] Gemäß einer Ausführungsform eines weiteren Aspektes der Erfindung wird ein System für drahtlose Kommunikation zwischen einem OBEX-Client und einem OBEX-Server über ein Kommunikationsnetz bereitgestellt, welches aufweist:
Mittel zum Erhalten OBEX-codierter Nachrichten;
Mittel zum Aufstellen eines Basisbandkommunikationskanals durch Verwendung eines Bluetooth-GOEP-Profils zwischen dem OBEX-Client und einer für Bluetooth freigegebenen OBEX-Bridge;
Mittel zum Einkapseln der OBEX-codierten Nachrichten in Basisbandnachrichten;
Mittel zum Bereitstellen der im Basisband eingekapselten Nachrichten der OBEX-Bridge;
Mittel zum Zuweisen einer internen IP-Adresse innerhalb der OBEX-Bridge;
Mittel zum Binden der IP-Adresse an den Basisbandkommunikationskanal;
Mittel zum Aufbauen eines TCP-/IP-Basisbandkommunikationskanals zwischen die OBEX-Bridge und einen abgesetzten Server; und
Mittel zum Weiterleiten der OBEX-codierten Nachrichten von dem Basisbandkanal an den TCP-/IP-Kanal.

[0062] Vorteilhafter Weise umfasst das System ferner Mittel zum Empfangen OBEX-codierter Datenantworten von dem abgesetzten Server über den TCP-/IP-Kanal; und Mittel zum Weiterleiten der OBEX-codierten Datenantworten an den Basisbandkanal mit Bezug auf die IP-Adresse.

[0063] Gemäß einer Ausführungsform eines weiteren Aspektes der Erfindung wird ein Computerprogramm bereitgestellt, welches aufweist:
ein Modul zum Erhalten OBEX-codierter Nachrichten;
ein Modul zum Aufstellen eines Basisbandkommunikationskanals durch Verwendung eines Bluetooth-GOEP-Profils zwischen dem OBEX-Client und einer für Bluetooth freigegebenen OBEX-Bridge;
ein Modul zum Einkapseln der OBEX-codierten Nachrichten in Basisbandnachrichten;
ein Modul zum Bereitstellen der im Basisband eingekapselten Nachrichten der OBEX-Bridge;
ein Modul zum Zuweisen einer internen IP-Adresse innerhalb der OBEX-Bridge;
ein Modul zum Binden der IP-Adresse an den Basisbandkommunikationskanal;
ein Modul zum Aufbauen eines TCP-/IP-Basisbandkommunikationskanals zwischen die OBEX-Bridge und einen abgesetzten Server; und
ein Modul zum Weiterleiten der OBEX-codierten Nachrichten von dem Basisbandkanal an den TCP-/IP-Kanal.

[0064] Vorteilhafter Weise umfasst das Medium ferner ein Modul zum Empfangen OBEX-codierter Datenantworten von dem abgesetzten Server über den TCP-/IP-Kanal; und ein Modul zum Weiterleiten der OBEX-codierten Datenantworten an den Basisbandkanal mit Bezug auf die IP-Adresse.

[0065] Gemäß einer Ausführungsform eines weiteren Aspektes der Erfindung wird eine OBEX-Bridge-Vorrichtung bereitgestellt, welche auf-

weist:
einen Prozessor;
einen Speicher, welcher mit dem Prozessor zur Kommunikation verbunden ist;
ein Programm, welches in dem Speicher gespeichert ist und enthält:
ein Modul zum Erhalten OBEX-codierter Nachrichten;
ein Modul zum Aufbauen eines Basisbandkommunikationskanals durch Verwendung eines Bluetooth-GOEP-Profils zwischen dem OBEX-Client und einer für Bluetooth freigegebenen OBEX-Bridge;
ein Modul zum Einkapseln der OBEX-codierten Nachrichten in Basisbandnachrichten;
ein Modul zum Bereitstellen der im Basisband eingekapselten Nachrichten der OBEX-Bridge;
ein Modul zum Zuweisen einer internen IP-Adresse innerhalb der OBEX-Bridge;
ein Modul zum Binden der IP-Adresse an den Basisbandkommunikationskanal;
ein Modul zum Aufbauen eines TCP-/IP-Basisbandkommunikations-kanals zwischen die OBEX-Bridge und einen abgesetzten Server; und
ein Modul zum Weiterleiten der OBEX-codierten Nachrichten von dem Basisbandkanal an den TCP-/IP-Kanal.

[0066] Vorteilhafter Weise umfasst die Vorrichtung ferner ein Modul zum Empfangen OBEX-codierter Datenantworten von dem abgesetzten Server über den TCP-/IP-Kanal; und ein Modul zum Weiterleiten der OBEX-codierten Datenantworten an den Basisbandkanal mit Bezug auf die IP-Adresse.

[0067] Gemäß einer weiteren Ausführungsform eines der Aspekte der Erfindung wird ein Verfahren der Erstellung einer eine IP-Adresse an ein RF-COMM-Kommunikationskanal bindenden Datenstruktur bereitgestellt, welches aufweist:
Erstellen einer Client-Basisband-Handhabe mit Bezug auf einen Speicherplatz in einem OBEX-Bridge-Speicher für eingehende Basisbandnachrichten;
Erstellen einer internen IP-Adresse innerhalb einer OBEX-Bridge, welche Nachrichten gestattet, zum und von einem Kommunikationsnetz über die IP-Adresse zu fließen;
Erstellen eines Eintrages in dem Speicher, um den Client-Basisband-Handhabungsdatentyp zu speichern;
Erstellen eines Eintrages in dem Speicher, um den IP-Adressendatentyp zu speichern;
Binden des Client-Basisband-Handhabungsdatentyps mit dem zugehörigen IP-Adressendatentyp.

[0068] Gemäß einer weiteren Ausführungsform eines Aspektes der Erfindung wird ein System für das Erstellen einer eine IP-Adresse an ein RF-COMM-Kommunikationskanal bindenden Datenstruktur bereitgestellt, welches aufweist:
Mittel zum Erstellen einer Client-Basisband-Handhabe mit Bezug auf einen Speicherplatz in einem OBEX-Bridge-Speicher für eingehende Basisbandnachrichten;
Mittel zum Erstellen einer internen IP-Adresse innerhalb einer OBEX-Bridge, welche Nachrichten gestattet, zum und von einem Kommunikationsnetz über die IP-Adresse zu fließen;
Mittel zum Erstellen eines Eintrages in dem Speicher, um den Client-Basisband-Handhabungsdatentyp zu speichern;
Mittel zum Erstellen eines Eintrages in dem Speicher, um den IP-Adressendatentyp zu speichern;
Mittel zum Binden des Client-Basisband-Handhabungsdatentyps mit dem zugehörigen IP-Adressendatentyp.

[0069] Gemäß einer weiteren Ausführungsform eines der Aspekte der Erfindung wird ein auf einem vom Computer lesbaren Medium gespeicherten Computerprogramm bereitgestellt, wobei das Programm aufweist:
ein Modul zum Erstellen einer Client-Basisband-Handhabe mit Bezug auf einen Speicherplatz in einem OBEX-Bridge-Speicher für eingehende Basisbandnachrichten;
ein Modul zum Erstellen einer internen IP-Adresse innerhalb einer OBEX-Bridge, welche Nachrichten gestattet, zum und von einem Kommunikationsnetz über die IP-Adresse zu fließen;
ein Modul zum Erstellen eines Eintrages in dem Speicher, um den Client-Basisband-Handhabungsdatentyp zu speichern;
ein Modul zum Erstellen eines Eintrages in dem Speicher, um den IP-Adressendatentyp zu speichern;
ein Modul zum Binden des Client-Basisband-Handhabungsdatentyps mit dem zugehörigen IP-Adressendatentyp.

[0070] Gemäß einer weiteren Ausführungsform eines der Aspekte der Erfindung wird eine OBEX-Bridge-Vorrichtung bereitgestellt, welche aufweist:
einen Prozessor;
einen Speicher, welcher mit dem Prozessor zur Kommunikation verbunden ist;
ein Programm, welches in dem Speicher gespeichert ist und enthält:
ein Modul zum Erstellen einer Client-Basisband-Handhabe mit Bezug auf einen Speicherplatz in einem OBEX-Bridge-Speicher für eingehende Basisbandnachrichten;
ein Modul zum Erstellen einer internen IP-Adresse innerhalb einer OBEX-Bridge, welche Nachrichten gestattet, zum und von einem Kommunikationsnetz über die IP-Adresse zu fließen;
ein Modul zum Erstellen eines Eintrages in dem Speicher, um den Client-Basisband-Handhabungsdatentyp zu speichern;
ein Modul zum Erstellen eines Eintrages in dem Spei-

cher, um den IP-Adressendatentyp zu speichern;
ein Modul zum Binden des Client-Basisband-Handhabungsdatentyps mit dem zugehörigen IP-Adressendatentyp.

[0071] Gemäß einer weiteren Ausführungsform eines der Aspekte der Erfindung wird ein Speicher für das Zugreifen durch ein auf einem Prozessor auszuführendes Programmmodul bereitgestellt, welcher aufweist:
eine Datenstruktur, welche in dem Speicher gespeichert ist und enthält:
einen Client-Basisband-Handhabungsdatentyp
einen IP-Adressendatentyp;
eine bindungsassoziative Referenz. zwischen dem Client-Basisband-Handhabungsdatentyp und dem IP-Adressendatentyp.

[0072] Gemäß einer weiteren Ausführungsform eines der Aspekte der Erfindung wird ein Speicher für das Zugreifen durch ein auf einem Prozessor auszuführendes Programmmodul bereitgestellt, welcher aufweist:
eine Datenstruktur, welche in dem Speicher gespeichert ist und enthält:
ein IP-Paket, welches enthält:
einen Ausgangs-IP-Adressendatentyp, wobei die Ausgangsadresse eine intern erstellte, von einer OBEX-Bridge generierte IP-Adresse ist;
einen Ziel-IP-Adressendatentyp, wobei die Ziel-IP-Adresse von der OBEX-Bridge bestimmt wird.

[0073] Es versteht sich, dass die obige Beschreibung nur für veranschaulichende Ausführungsformen repräsentativ ist. Für die Bequemlichkeit des Lesers, haben sich die oberen Beschreibungen auf eine repräsentative Auswahl aller möglichen Ausführungsformen gerichtet, eine Auswahl, welche die Grundsätze der Erfindung lehrt. Die Beschreibung hat nicht versucht, erschöpfend alle mögliche Variationen aufzuzählen. Dass alternative Ausführungsformen für einen spezifischen Teil der Erfindung nicht vorgestellt worden sind oder dass weitere unbeschriebene alternative Ausführungsformen für einen Teil verfügbar sein können, wird nicht als Disclaimer für solche alternative Ausführungsformen betrachtet. Es wird anerkannt werden, dass viele solcher unbeschriebenen Ausführungsformen die gleichen Grundsätze der Erfindung enthalten und sonst äquivalent sind. Somit soll verstanden werden, dass die hier gezeigten und beschriebenen Ausführungsformen und Variationen lediglich illustrativ für die Grundsätze der Erfindung sind und dass verschiedene Modifikationen ohne Abweichen vom Umfang der Erfindung implementiert werden können.

Patentansprüche

1. Verfahren zur Verwendung in einer OBEX-Bridge (**150**, **250**, **450**) und zum Ermöglichen einer Kommunikation zwischen einem Kurzstrecken-HF-Client (**101**, **401**) und einem abgesetzten OBEX-Server (**199**, **299**, **399**, **499**) über eine TCP/IP-Verbindung (**466**), wobei das Verfahren die Schritte umfasst:
Empfangen von RFCOMM-codierten OBEX-Nachrichten über eine Kurzstreckenfunkverbindung von dem HF-Client (**101**, **401**);
Zuweisen einer eindeutigen IP-Adresse, bei der es sich um eine interne Adresse der OBEX-Bridge handelt, zu den empfangenen RFCOMM-codierten OBEX-Nachrichten;
Bestimmen einer Ziel-IP-Adresse für die empfangenen RFCOMM-codierten OBEX-Nachrichten;
Einkapseln von OBEX-Datenanfragen der empfangenen RFCOMM-codierten OBEX-Nachrichten in TCP/IP-Pakete; und
Übertragen der TCP/IP-gekapselten Pakete an den OBEX-Server über eine TCP/IP-Verbindung.

2. Verfahren nach Anspruch 1, des weiteren enthaltend:
Auswählen eines OBEX-Servers, an den die empfangenen OBEX-Nachrichten zu senden sind; und
wobei der Bestimmungsschritt eine Bestimmung der IP-Adresse des ausgewählten OBEX-Servers umfasst.

3. Verfahren nach Anspruch 2, wobei der Auswahlschritt hinsichtlich der empfangenen RFCOMM-codierten OBEX-Nachrichten inhaltssensitiv ist.

4. Verfahren nach Anspruch 2 oder 3, wobei der Auswahlschritt basierend auf dem GOEP-Profil des HF-Clients und der Spezialisierung des OBEX-Servers durchgeführt wird.

5. Verfahren nach einem der Ansprüche 1 bis 4, des weiteren umfassend:
Empfangen von TCP/IP-Packeten mit OBEX-codierten Datenantworten über die TCP/IP-Verbindung von dem OBEX-Server;
Weiterleiten der OBEX-codierten Datenantworten basierend auf der eindeutigen IP-Adresse des Zuweisungsschrittes; und
Übertragen der OBEX-codierten Datenantworten zu dem HF-Client über die Kurzstreckenfunkverbindung.

6. Computerlesbares Medium, in dem ein Computerprogramm zum Ermöglichen einer Kommunikation zwischen einem Kurzstrecken-HF-Client (**101**, **401**) und einem abgesetzten OBEX-Server (**199**, **299**, **399**, **499**) über eine TCP/IP-Verbindung (**466**) gespeichert ist, wobei das Computerprogramm zum Veranlassen einer Durchführung der in einem der Ansprüche 1 bis 5 genannten Schritte durch eine OBEX-Bridge dient, wenn das Computerprogramm auf einem in der OBEX-Brigde enthaltenen Allzweck-

computer ausgeführt wird.

7. System zum Ermöglichen einer Kommunikation zwischen einem Kurzstrecken-HF-Client (**101**, **401**) und einem abgesetzten Server (**199**, **299**, **399**, **499**) über eine TCP/IP-Verbindung (**466**),
dadurch gekennzeichnet, dass
das System einen abgesetzten OBEX-Server (**199**, **299**, **399**, **499**) enthält und dass das System eine OBEX-Bridge enthält, wobei die OBEX-Bridge enthält:
Mittel zum Empfangen RFCOMM-codierter OBEX-Nachrichten über eine Kurzstreckenfunkverbindung von dem HF-Client (**101**, **401**);
Mittel zum Zuweisen einer eindeutigen IP-Adresse, bei der es sich um eine interne Adresse der OBEX-Bridge handelt, zu den empfangenen RFCOMM-codierten OBEX-Nachrichten;
Mittel zum Bestimmen einer Ziel-IP-Adresse für die empfangenen RFCOMM-codierten OBEX-Nachrichten;
Mittel zum Einkapseln von OBEX-Datenanfragen der empfangenen RFCOMM-codierten OBEX-Nachrichten in TCP/IP-Pakete; und
Mittel zum Übertragen der TCP/IP-gekapselten Pakete zu dem OBEX-Server über eine TCP/IP-Verbindung.

8. System nach Anspruch 7, wobei die OBEX-Bridge des weiteren enthält:
Mittel zum Auswählen eines bestimmten OBEX-Servers, an den die empfangenen OBEX-Nachrichten zu senden sind; und
wobei die Bestimmungsmittel ausgestaltet sind zum Bestimmen der IP-Adresse des ausgewählten OBEX-Servers.

9. System nach Anspruch 8, wobei das Mittel zum Auswählen hinsichtlich der empfangenen RFCOMM-codierten OBEX-Nachrichten inhaltssensitiv ist.

10. System nach Anspruch 8 oder 9, wobei das Mittel zum Auswählen auf das GOEP-Profil des HF-Clients und die Spezialisierung des OBEX-Servers anspricht.

11. System nach einem der Ansprüche 7 bis 10, wobei die OBEX-Bridge des weiteren enthält:
Mittel zum Empfangen von TCP/IP-Paketen mit OBEX-codierten Datenantworten über die TCP-IP-Verbindung von dem OBEX-Server;
Mittel zum Weiterleiten der OBEX-codierten Datenantworten basierend auf der den empfangenen RFCOMM-codierten OBEX-Nachrichten zugewiesenen eindeutigen IP-Adresse; und
Mittel zum Übertragen der OBEX-codierten Datenantworten zu dem HF-Client über die Kurzstreckenfunkverbindung.

12. OBEX-Bridge-Vorrichtung (**150**, **250**, **450**) zum Ermöglichen einer Kommunikation zwischen einem Kurzstrecken-HF-Client (**101**, **401**) und einem abgesetzten OBEX-Server (**199**, **299**, **399**, **499**) über eine TCP/IP-Verbindung (**466**), wobei die OBEX-Bridge-Vorrichtung enthält:
einen Prozessor (**202**);
einen Speicher (**205**), der kommunikativ mit dem Prozessor verbunden ist;
ein in dem Speicher gespeichertes Programm mit:
einem Modul zum Empfangen RFCOMM-codierter OBEX-Nachrichten über eine Kurzstreckenfunkverbindung von dem HF-Client (**101**, **401**);
einem Modul zum Zuweisen einer eindeutigen IP-Adresse, bei der es sich um eine interne Adresse der OBEX-Bridge handelt, zu den empfangenen RFCOMM-codierten OBEX-Nachrichten;
einem Modul zum Bestimmen einer Ziel-IP-Adresse für die empfangenen RFCOMM-codierten OBEX-Nachrichten;
einem Modul zum Einkapseln von OBEX-Datenanfragen der empfangenen RFCOMM-codierten OBEX-Nachrichten in TCP/IP-Paketen; und
einem Modul zum Übertragen der TCP/IP-gekapselten Packete zu dem OBEX-Server über eine TCP/IP-Verbindung.

13. OBEX-Bridge-Vorrichtung nach Anspruch 12, des weiteren enthaltend:
ein Modul zum Auswählen eines bestimmten OBEX-Servers, an den die empfangenen OBEX-Nachrichten zu senden sind; und
wobei das Bestimmungsmodul ausgestaltet ist zum Bestimmen der IP-Adresse des ausgewälten OBEX-Servers.

14. OBEX-Bridge-Vorrichtung nach Anspruch 13, wobei das Modul zum Auswählen hinsichtlich der empfangenen RFCOMM-codierten OBEX-Nachrichten inhaltssensitiv ist.

15. OBEX-Bridge-Vorrichtung nach Anspruch 13 oder 14, wobei das Modul zum Auswählen auf das GOEP-Profil des HF-Clients und die Spezialisierung des OBEX-Servers anspricht.

16. OBEX-Bridge-Vorrichtung nach einem der Ansprüche 12 bis 15, wobei die OBEX-Bridge-Vorrichtung des weiteren enthält:
ein Modul zum Empfangen von TCP/IP-Paketen mit OBEX-codierten Datenantworten über die TCP/IP-Verbindung von dem OBEX-Server;
ein Modul zum Weiterleiten der OBEX-codierten Datenantworten basierend auf der den empfangenen RFCOMM-codierten OBEX-Nachrichten zugewiesenen eindeutigen IP-Adresse; und
ein Modul zum Übertragen der OBEX-codierten Datenantworten zu dem HF-Client über die Kurzstreckenfunkverbindung.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

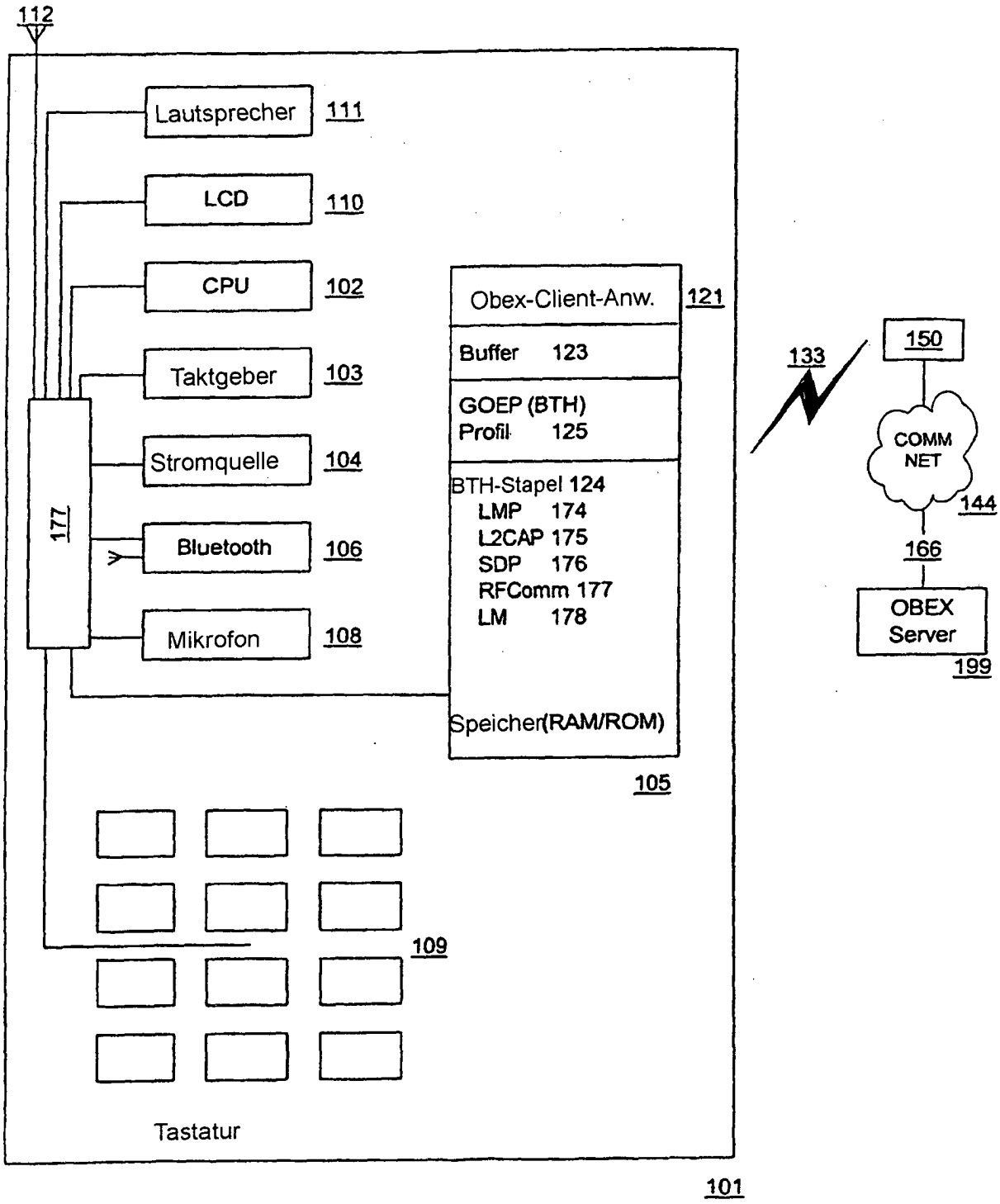

FIG. 1

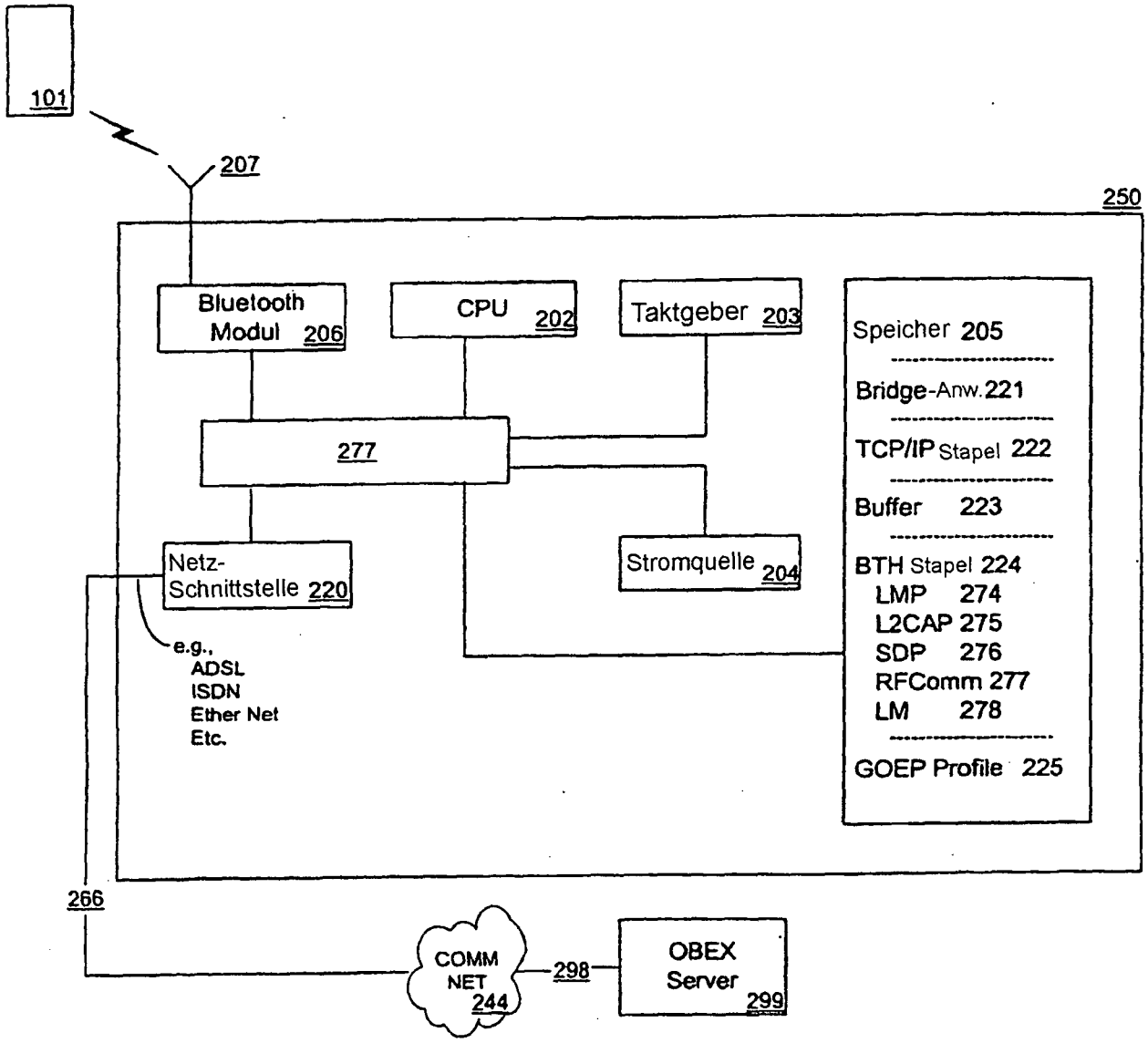
101
207
250
Bluetooth Modul 206
CPU 202
Taktgeber 203
277
Netz-Schnittstelle 220
e.g.,
ADSL
ISDN
Ether Net
Etc.
Stromquelle 204
Speicher 205
Bridge-Anw. 221
TCP/IP Stapel 222
Buffer 223
BTH Stapel 224
LMP 274
L2CAP 275
SDP 276
RFComm 277
LM 278
GOEP Profile 225
266
COMM NET 244
298
OBEX Server 299

FIG. 2

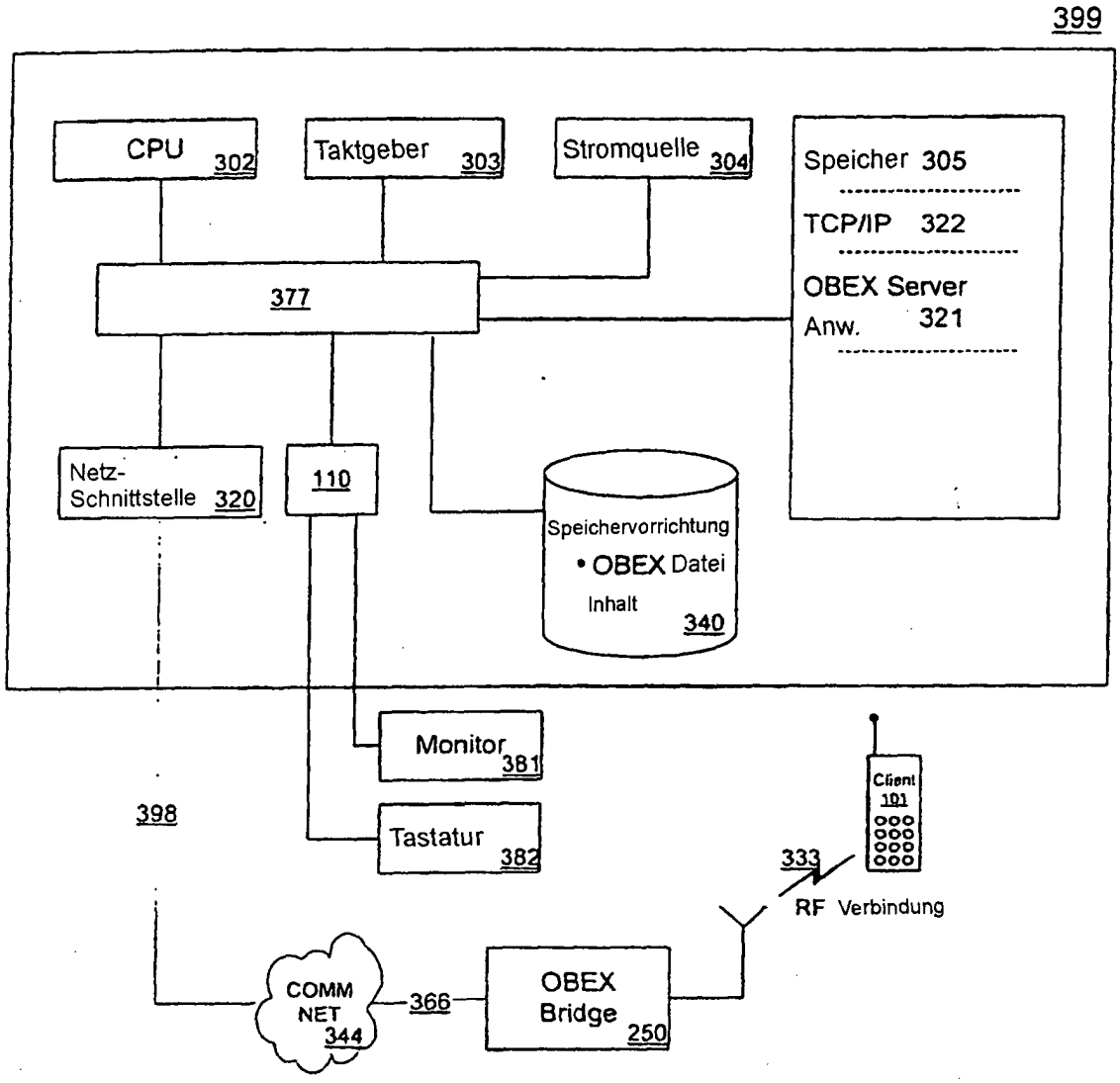
399
CPU 302
Taktgeber 303
Stromquelle 304
Speicher 305
TCP/IP 322
OBEX Server Anw. 321
377
Netz-Schnittstelle 320
110
Speichervorrichtung
• OBEX Datei Inhalt
340
Monitor 381
Tastatur 382
398
Client 101
333
RF Verbindung
COMM NET 344
366
OBEX Bridge 250

FIG. 3

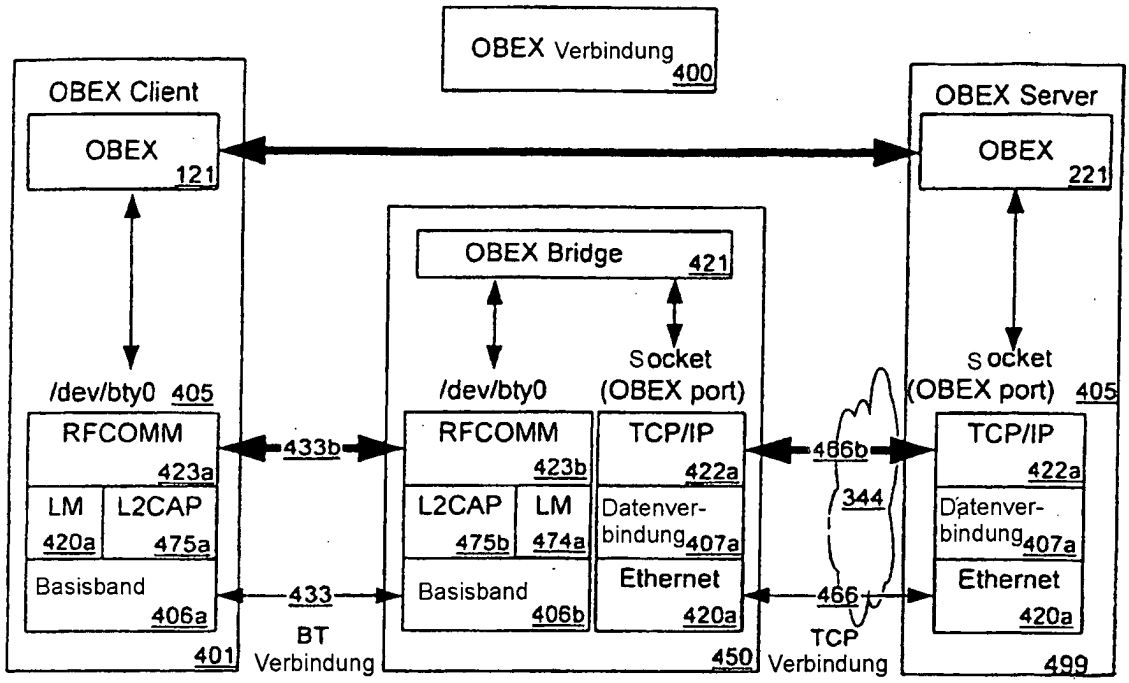

FIG. 4

Client RFComm Handhabe 487	Client IP Adresse 488
pointer@men_Add_RF_Client_1	192.168.1.101
pointer@men_Add_RF_Client_2	192.168.1.102
pointer@men_Add_RF_Client_3	192.168.1.103

489

501) Das Endgerät startet die OBEX-Client-Anwendung
502) Das Endgerät baut eine Bluetooth-Verbindung zur AP auf
503) Das Endgerät verwendet .5DP, um OBEX-Servers Fähigkeit für AP zu prüfen
504) AP antwortet über SDP, dabei für das GEDP-Profil benötigten Parameter ausgebend
505) Das Endgerät iniziiert Protokolle nach dem GEDP-Profil
506) Der RFComm-Kanal (serielle Kabelemmulation) wird zwischen dem Endgerät und der AP aufgebaut
507) AP weist intern eine IP-Adresse zu und bindet diese an die aufgestellte RFCOMM-Handhabung
508) AP stellt TCP-/IP-Verbindung zum abgesetzten OBEX-Server auf
509) OBEX-Client im Endgerät sendet OBEX-Daten über die RFCOMM-Verbindung
510) AP-Bridge empfängt Daten vom RFCOMM und kapselt diese in TCP-/IP-Pakete ein (leitet Daten vom RFCOMM an das TCP-/IP-Socket weiter)
511) Antworten vom abgesetzten OBEX-Endgerät werden von dem der AP-Bridge entsprechenden TCP-/IP-Socket empfangen, und die Bridge kapselt diese in RFCOMM-Pakete ein(leitet Daten vom TCP-/IP-Socket an die RFCOMM-Handhabe weiter)
512) OBEX-Client im Endgerät empfängt Daten über das TFCOMM-Kanal..
513) Schritte 0...12 werden wiederholt, bis das Endgerät RFCOMM-Verbindung oder Bluetooth-Verbindung trennt
514) AP-Bridge beendet TCP-/IP-Verbindung zum abgesetzten Server und gibt zugewiesene IP-Adresse frei

FIG. 5