(19) Deutsches Patent- und Markenamt

(10) **DE 11 2012 003 840 T5** 2014.08.14

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2013/040438**
in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2012 003 840.1**
(86) PCT-Aktenzeichen: **PCT/US2012/055545**
(86) PCT-Anmeldetag: **14.09.2012**
(87) PCT-Veröffentlichungstag: **21.03.2013**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **14.08.2014**

(51) Int Cl.: ***H04W 80/06*** (2009.01)
H04W 84/22 (2009.01)

(30) Unionspriorität:
61/535,316 **15.09.2011** **US**

(71) Anmelder:
Fisher-Rosemount Systems, Inc., Round Rock, Tex., US

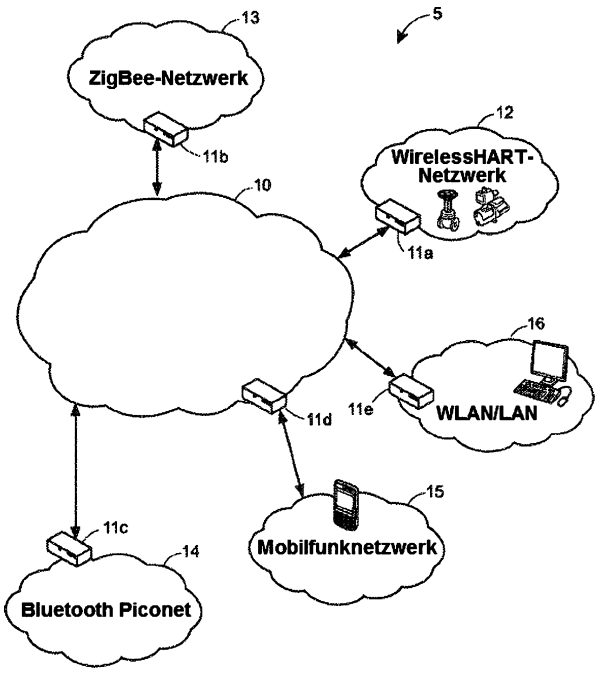
(74) Vertreter:
Meissner, Bolte & Partner GbR, 80538, München, DE

(72) Erfinder:
Nixon, Mark J., Round Rock, Tex., US; Han, Song, Austin, Tex., US; Rotvold, Eric D., West St. Paul, Minn., US; Chen, Deji, Austin, Tex., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Kommunikation von Datenframes über Kommunikationsnetzwerke hinweg, die inkompatible Netzwerk-Routingprotokolle verwenden**

(57) Zusammenfassung: Ein Kommunikationsverfahren, das dazu dient, Internet-Protokoll(IP)-Datenframes, z. B. IPv6-Datenframes, über ein Kommunikationsnetz zu übertragen, das ein Nicht-IP-Netzwerkrouting-Protokoll verwendet, z. B. ein Kommunikationsnetzwerk, das ein Netzwerkrouting-Protokoll implementiert, das nicht übereinstimmt oder inkompatibel ist mit einem IP-Netzwerkrouting-Protokoll, z. B. mit einem WirelessHART-Protokoll. Dieses Kommunikationsverfahren ermöglicht es beispielsweise Feldgeräten oder anderen intelligenten Geräten innerhalb eines Prozessanlagennetzwerks, das ein Nicht-IP-Netzwerkrouting benutzt (ein Netzwerk, das kein IP-basiertes Netzwerkrouting benutzt), Message-Routing mittels IP-Datenframes durchzuführen, die von Internetprotokoll-fähigen Geräten innerhalb oder außerhalb des Prozessanlagennetzwerks erzeugt und empfangen werden. Das Kommunikationsverfahren beeinflusst oder ändert nicht die normale Kommunikation innerhalb des Nicht-IP-Kommunikationsnetzwerks, da das Kommunikationsverfahren das Netzwerkrouting oder die Struktur des Netzwerkroutings des Nicht-IP-Kommunikationsnetzwerks benutzt, um die IP-Datenframes innerhalb des Nicht-IP-Kommunikationsnetzwerks zu übertragen und dabei die IP-Netzwerkrouting-Informationen des IP-Datenframes zu bewahren, die nötig ist, um anschließend die Nachrichten in einem IP-basierten Kommunikationsnetzwerk zu routen oder zu entschlüsseln und um die IP-Nachrichtendatenframes in einem IP-fähigen Gerät zu benutzen.



Beschreibung

GEBIET DER OFFENBARUNG

[0001] Die vorliegende Offenbarung bezieht sich auf das Empfangen, Verarbeiten und Übermitteln von Datenframes zwischen Geräten über verschiedene Kommunikationsnetzwerke, um das Routing von Kommunikationsdatenframes zu ermöglichen, indem ein Netzwerk-Routingprotokoll mithilfe eines zweiten Netzwerk-Routingprotokolls über ein Netzwerk gesendet wird.

HINTERGRUND

[0002] Ursprünglich wurden zur Prozesssteuerung in Prozessanlagen eingesetzte Prozessleitsysteme und Feldgeräte dezentral überwacht und gesteuert. Fehlende standardisierte Kommunikationsprotokolle und Rechenleistung führten dazu, dass das Überwachen und Steuern von Regelkreisen etwas war, das dem Anlagenbediener überlassen wurde, der sich in der Regel am Ort der Anlage selbst befand. Die Notwendigkeit für eine zentralisierte Überwachung und Kontrolle der Prozesssteuerung aus der Ferne führte zur Entwicklung und Verbreitung von Feldgeräten, die beispielsweise die bekannten zweiadrigen 4–20 mA Stromschleifen-Geräte und andere festverdrahtete Kommunikationssysteme von Punkt zu Punkt verwendeten.

[0003] Die Verfügbarkeit von kostengünstigen Computern mit geringem Energiebedarf förderte den Einsatz von intelligenten Feldgeräten mit verbesserter Rechenleistung (Mikroprozessoren) und elektrischen Kommunikationsfähigkeiten. Intelligente Feldgeräte konnten nun verstärkt miteinander (als Netzwerk) und mit zentralisierten Prozessleitsystemen kommunizieren, um die automatische Steuerung durchzuführen. Diese verbesserte Netzwerkfähigkeit wurde in eine verbesserte Prozessanlagensteuerung und infolgedessen verbesserte Prozessanlagenleistung umgesetzt.

[0004] Mit dem Augenmerk auf einer Verbesserung der Akzeptanz von intelligenten Feldgeräten in der Prozesssteuerungsindustrie und teilweise zur Verbesserung der Interoperabilität zwischen diesen Feldgeräten, haben Hersteller von Feldgeräten mehrere digitale Netzwerkprotokolle entwickelt und standardisiert, um die Kommunikation von Feldgeräten zwischen verschiedenen Prozessanlagen zu ermöglichen. Einige der Standards, beispielsweise das Highway Addressable Remote Transducer (HART) Protokoll, waren besonders attraktiv, da sie die bestehende analoge zweiadrige 4–20 mA-Infrastruktur nutzten, die inzwischen in allen Prozessanlagen vorhanden war, um digitale Informationen zwischen Feldgeräten zu senden und zu empfangen. Andere Standards umfassten bus-basierte Systeme, wie den FOUNDATION® Fieldbus-Standard, den Profibus-Standard usw.

[0005] Schließlich machten eine höhere Rechenleistung, erhöhte Komponentenintegration und Entwicklungen im allgemeinen Bereich der Signalverarbeitung es wirtschaftlich rentabel, eine Schmalband-Funkfrequenz (RF) im niedrigen Bereich oder drahtlose Kommunikationsstandards für die Prozesssteuerungsindustrie zu entwickeln und einzusetzen. Einige dieser Protokolle wurden entwickelt, um in den unlizenzierten oder locker lizenzierten industriellen, wissenschaftlichen und medizinischen (ISM) elektromagnetischen Frequenzbändern, zum Beispiel dem 2,4 GHz-Band, eingesetzt zu werden. Feldgeräte, die solche Schmalband-Funkfrequenzprotokolle im niedrigen Bereich im ISM-Band unterstützen, wurden und werden in Prozessanlagenumgebungen eingesetzt. Obwohl unlizenziert, können Regierungsbehörden Vorschriften erlassen, welche die maximale Übertragungsleistung beschränken. Daher werden solche Protokolle oft als Low Power-Kommunikationsprotokolle bezeichnet. Darüber hinaus wurden mehrere drahtgebundene Netzwerkprotokolle, unter anderem das HART-Protokoll, so angepasst, dass sie die Drahtlos-Fähigkeiten solcher Feldgeräte steigern. Insbesondere entstand das WirelessHART-Protokoll aus dem drahtgebundenen HART-Protokoll, um die Interoperabilität von WirelessHART-fähigen Feldgeräten innerhalb eines drahtgebundenen HART-Prozessanlagennetzwerks zu unterstützen und zu ermöglichen. Trotz dieser Verbesserungen ist die Prozessanlagenkommunikation im Allgemeinen auf Prozessanlagenumgebungen beschränkt, die ganz spezielle Herausforderungen hinsichtlich Robustheit und Sicherheit bieten.

[0006] Allerdings wurden, in einer eigenen Branche, Personalcomputer seit Beginn der 1960er Jahre weltweit immer selbstverständlicher. Gleichzeitig führten Fortschritte auf dem Gebiet der allgemeinen Computernetzwerke zu der Entstehung des Internets, über das Personalcomputer nahtlos miteinander kommunizieren. Diese nahtlose Kommunikation wurde zum großen Teil durch die Akzeptanz und Einhaltung des Internetprotokolls Version 4 (IPv4) ermöglicht. Das IPv4-Protokoll weist, zum Teil, Computern, die über ein Netzwerk oder allgemein über das Internet kommunizieren, eindeutige IP-Adressen zu. Das die breite Akzeptanz von IPv4 nicht vorhergesehen wurde, haben die Entwickler der IPv4-Spezifikationen die Maximallänge der IP-Adresse eines

Computers auf 32 Bit beschränkt. Allerdings hat die starke Zunahme der vernetzten Computer, einschließlich Server, Handheld-Computer und Personalcomputer eine Situation geschaffen, in der IPv4-Adressen in absehbarer Zukunft erschöpft sein könnten. Um diese Situation präventiv abzuwenden, erhöht das neue geplante IPv6-Protokoll zum Teil den Platz oder die Länge der IP-Adresse auf 128 Bit. Mathematisch bedeutet das 3,4 mal 10^{38} eindeutige IPv6-Adressen. In jedem Fall ist die Nutzung des Internetprotokolls (im Folgenden als IP oder IP-Netzwerkprotokoll bezeichnet) zur Ausführung von Routing und anderen Kommunikationsaktivitäten in drahtgebundenen und drahtlosen Umgebungen weit verbreitet. Tatsächlich werden die meisten heutigen Anwendungen dazu entwickelt, außerhalb eines Gerätes unter Nutzung des Internetprotokolls als Backbone des Kommunikationsnetzwerks zu kommunizieren, oder um vernetzt zu kommunizieren.

[0007] Außerdem hängen die Anforderungen in der Prozessanlagenindustrie immer mehr mit den Entwicklungen in den allgemeinen Computertechnologien zusammen. So wurden beispielsweise Prozessanlagen wie Raffinerien, Ölfelder und Bergbauaktivitäten verstärkt in den entlegensten Ecken der Welt geplant, damit sie sich in der Nähe der Fundorte der Rohmaterialien befinden. Zunehmende Globalisierung hat dazu geführt, dass diese Anlagen im Besitz von Konzernen sind und von ihnen überwacht und kontrolliert wurden, die sich in anderen Ländern, wenn nicht gar auf anderen Kontinenten befinden. Geopolitische Instabilität, klimatische Bedingungen und/oder fehlende erträgliche Bedingungen am Standort dieser Prozessanlagen haben Konzerne gezwungen, Mittel zu suchen, diese Prozessanlagen aus der Ferne zu überwachen und zu steuern. Obwohl in anderen Arten von vernetzter Kommunikation zahlreiche allgemeine Entwicklungen der Computernetzwerke Anwendung finden, wie zum Beispiel Mobiltelefone, Sicherheitsüberwachungsanwendungen usw., sind bestehende Netzwerkprotokolle zur Prozesssteuerung schlecht ausgestattet und in der Regel nicht kompatibel mit Universal-Computernetzwerkprotokollen wie IPv4 oder IPv6, und können die Integration dieser Protokolle innerhalb einer Prozessanlagenumgebung nicht ohne eine wesentliche unterstützende Kommunikationsinfrastruktur daher nicht ermöglichen. Tatsächlich wurden die meisten der bestehenden Prozesssteuerungsnetzwerkprotokolle unabhängig von oder ohne Bezug zu nun üblicheren oder umfassenderen Universal-Computerprotokollen wie den Kommunikationsprotokollen IPv4 oder IPv6 entwickelt. Außerdem haben die robusten Sicherheitsmechanismen der allgemeinen Computerprotokolle, wie die 128 Bit Verschlüsselung, nicht den Weg in die Prozessanlagennetzwerke gefunden, was auch die Fähigkeit einschränkt, diese Prozessanlagennetzwerke außerhalb der kontrollierten Anlagenumgebungen zu erweitern. Tatsächlich sind die meisten, wenn nicht gar alle Prozessanlagenkommunikationsprotokolle, die Sicherheitsmerkmale einsetzen, nicht mit den weit verbreiteten IPv4- und IPv6-Sicherheitsprotokollen kompatibel.

[0008] Allerdings ist es nun eventuell möglich, dank des neuen erweiterten IPv6-Adressbereichs jedem intelligenten Feldgerät innerhalb einer kommunikationsfähigen Prozessanlage eine IPv6-Adresse zuzuweisen. Es ist daher äußerst vorteilhaft, einfach ein allgemeines Computernetzwerkprotokoll wie IPv4 oder IPv6 zu nutzen, um innerhalb einer Prozessanlagenumgebung zu kommunizieren und diese Kommunikation einfach auf Geräte außerhalb der Anlage zu erweitern. Allerdings kann nicht jedes intelligente Feldgerät angepasst oder einfach daran angepasst werden, das IPv4- oder das IPv6-Protokoll einzuhalten, und so ist gegenwärtig die Nutzung des IPv4- oder IPv6-Kommunikationsprotokolls und anderer Routingfunktionen innerhalb einer Prozessanlagenumgebung nicht umsetzbar. Insbesondere machen es Hardwarebeschränkungen wie begrenzter Feldgerätespeicher und Rechenleistung es technisch unmöglich, bestehende Feldgeräte nachträglich mit Software oder Hardware auszustatten, die für die Unterstützung des IPv4- oder IPv6-Protokolls nötig oder mit ihnen kompatibel ist. Ähnlich ist es auch wirtschaftlich nicht machbar, jedes intelligente Feldgerät gegen ein IPv6-fähiges Feldgerät auszutauschen, indem in aktuellen Netzwerken für besondere Zwecke bestehende Low Power-Schmalband-Kommunikationsprotokolle eingesetzt werden, z. B. für diejenigen in einer Prozessanlage. Als Folge davon ist es gegenwärtig nicht machbar oder nicht sehr praktisch, die bekannten und nun allgegenwärtigen IPv4- oder IPv6-Protokolle zu verwenden, um innerhalb von oder zwischen Geräten in Prozessanlagen zu kommunizieren, obwohl viele Computer innerhalb einer Prozessanlage möglicherweise davon profitieren, Anwendungen ausführen zu können, die zur Nutzung in der Umgebung eines IPv4- oder IPv6 Kommunikationsprotokolls entwickelt wurden oder dazu, einfach mit Geräten außerhalb des Prozessanlagennetzwerks zu kommunizieren, die ein solches Kommunikationsnetzwerkprotokoll verwenden. Dadurch besteht ein Bedarf für Feldgeräte, die darauf abgestimmt sind, innerhalb der bisherigen drahtlosen Prozessanlagennetzwerke zu funktionieren, und die gleichzeitig daran angepasst werden, mit Geräten zu kommunizieren, die sich irgendwo im Internet befinden.

ZUSAMMENFASSUNG

[0009] Ein Kommunikationsverfahren dient dazu, Internet-Protokoll(IP)-Datenframes, wie IPv6-Datenframes oder andere Arten von allgemeinen Computer-Datenframes nahtlos über ein Kommunikationsnetz zu übertragen, das ein Nicht-IP-Netzwerkroutingprotokoll verwendet, wie zum Beispiel ein Mesh-Netzwerkroutingproto-

koll oder ein impliziertes Ziel- oder Adressierungs-Routingprotokoll, das mit einem IP-Netzwerkroutingprotokoll, das ein explizites Zieladressprotokoll ist, nicht übereinstimmt oder nicht damit kompatibel ist. Das WirelessHART-Protokoll ist ein Beispiel für ein Nicht-IP-Kommunikationsnetzwerk (das auch ein implizites Ziel- oder Adressierungs-Netzwerkroutingprotokoll ist und Graphen-Routing im Netzwerklayer verwendet), das daher nicht grundsätzlich das Routing von IP-Datenframes unterstützt, die ein IP-Netzwerkroutingprotokoll verwenden, d. h. eines, das IP-Adressrouting oder explizite Zieladressierung nutzt. Dieses Kommunikationsverfahren ermöglicht es beispielsweise Feldgeräten oder anderen intelligenten Geräten innerhalb eines Prozessanlagennetzwerks, das ein Nicht-IP-Netzwerkrouting nutzt (ein Netzwerk, das kein IP-basiertes Netzwerkrouting wie das WirelessHART-Kommunikationsprotokoll nutzt), Message-Routing mittels IP-Datenframes durchzuführen, die von Internetprotokoll-fähigen Geräten innerhalb oder außerhalb des Prozessanlagennetzwerks erzeugt und empfangen werden. Wichtig ist es, dass das Kommunikationsverfahren nicht die normale Kommunikation innerhalb des Nicht-IP-Kommunikationsnetzwerks beeinflusst oder verändert, da das Kommunikationsverfahren das Netzwerkrouting oder die Struktur des Netzwerkroutingprotokolls des Nicht-IP-Kommunikationsnetzwerks nutzt (wie z. B. ein Mesh-Netzwerkroutingprotokoll oder ein implizites Ziel- oder Adressierung-Routingprotokoll), um die IP-Datenframes innerhalb des Nicht-IP-Kommunikationsnetzwerks zu übertragen und dabei die IP-Netzwerkroutinginformationen des IP-Datenframes zu bewahren, die nötig sind, um anschließend die Nachrichten in ein IP-basiertes Kommunikationsnetzwerk zu routen oder zu entschlüsseln und die IP-Nachrichtendatenframes in einem IP-fähigen Gerät zu verwenden. Generell bewahrt dieses Kommunikationsverfahren die IP-Netzwerkroutinginformationen auf dem gesamten Netzwerkroutingpfad und an den Übertragungspunkten zwischen einem Nicht-IP-Kommunikationsnetzwerk und einem IP-Kommunikationsnetzwerk, so dass der IP-Datenframe von IP-fähigen Anwendungen in Geräten genutzt werden kann (entweder innerhalb oder außerhalb des Nicht-IP-Kommunikationsnetzwerkes), um die Standard-IP-Netzwerkkommunikation auszuführen.

[0010] Im Wesentlichen verwendet das hier beschriebene Kommunikationsverfahren die Netzwerkroutingstruktur oder -mechanismen des Nicht-IP-Netzwerks zum Übermitteln und Empfangen von IP-Datenframes über Kommunikationskanäle an das Nicht-IP-Netzwerkroutingprotokoll, auch wenn das Nicht-IP-Kommunikationsnetzwerk nicht grundsätzlich die Übertragung von IP-Datenframes oder von einem IP-Netzwerkroutingprotokoll unterstützt. Das Kommunikationsverfahren basiert generell auf dem Routen der IP-Datenframes über das Nicht-IP-Kommunikationsnetzwerk unter Nutzung des Netzwerk-Routingmechanismus (Netzwerk-Routingprotokollinformationen) des Nicht-IP-Kommunikationsnetzwerks unter Bewahrung der IP-Netzwerkroutinginformationen (IP-Netzwerk-Routingprotokollinformationen) innerhalb des Datenframes, damit diese Informationen in einem IP-Kommunikationsnetzwerk von Anwendungen genutzt werden können, die mithilfe von IP-Datenframes und IP-Stacks in Geräten kommunizieren, in denen sich die Anwendung befindet. Durch dieses Merkmal können Standard-IP-Datenframes zusammen mit oder gleichzeitig mit Datenframes gesendet werden, die entsprechend einem Nicht-IP-Kommunikationsnetzwerk mittels Protokoll-Stacks (z. B. Datenlinklayer-Mechanismen und die physische Layerstruktur) des Nicht-IP-Kommunikationsnetzwerks konfiguriert wurden. In einem Fall kann das Verfahren möglicherweise die IP-Datenframes in ein oder mehr Fragmente fragmentieren und die fragmentierten Datenframes über die Kommunikationskanäle des Nicht-IP-Kommunikationsnetzwerk routen, indem beispielsweise der Fragmentierungstechnik genutzt wird, die als Teil des 6LoWPAN-Protokolls entwickelt oder bereitgestellt wurde.

[0011] Von Vorteil ist es, dass das hier beschriebene Kommunikationsverfahren es einem IP-Kommunikationsnetzwerk erlaubt, nahtlos mit einem Nicht-IP-Kommunikationsnetzwerk verbunden zu sein, wie z. B. spezialisierte Kommunikationsnetzwerke im Zusammenhang mit dem Senden von Informationen zur Prozesssteuerung, damit IP-Datenframes nahtlos über diese Netzwerke kommuniziert werden können, ohne die Rechenleistung oder Fähigkeiten der Geräte an dem Nicht-IP-Kommunikationsnetzwerk wesentlich zu ändern oder ihnen hinzugefügt zu werden. Außerdem können aufgrund dieses Kommunikationsverfahrens IP-Datenframes von einem Gerät im Nicht-IP-Kommunikationsnetzwerk direkt an ein Gerät in einem IP-fähigen Kommunikationsnetzwerk (z. B. über Internet) gesendet werden, indem der Netzwerkroutingmechanismus des Nicht-IP-Kommunikationsnetzwerk für einen Teil des Kommunikationspfades genutzt wird und der Netzwerkroutingmechanismus des IP-Kommunikationsnetzwerks für einen anderen Teil des Kommunikationspfades verwendet wird, ohne das Nicht-IP-Kommunikationsnetzwerk zu zwingen, IP-Netzwerkrouting einzusetzen. Auch können aufgrund dieses Verfahren IP-Datenframes, die innerhalb eines IP-Kommunikationsnetzwerks generiert wurden, von Geräten innerhalb eines Nicht-IP-Kommunikationsnetzwerkes als IP-Datenframes gesendet und empfangen werden, ohne alle Geräte in dem Nicht-IP-Kommunikationsnetzwerk mit IP-fähigen Stacks versorgen zu müssen, damit im Nicht-IP-Kommunikationsnetzwerk kommuniziert werden kann. Dadurch ermöglicht es dieses Verfahren socketbasierten Anwendungen, die entwickelt wurden, um IP-Datenframes zu generieren, zu empfangen und zu nutzen, oder um eine IP-basierte Kommunikation innerhalb des Nicht-IP-Kommunikationsnetzwerks durchzuführen und dennoch unter Nutzung von IP-Datenframes zu kommunizieren, entweder mit

Hilfe von Anwendungen in Geräten in dem Nicht-IP-Kommunikationsnetzwerk oder mit Geräten in anderen Netzwerken, z. B. in IP-Kommunikationsnetzwerken. Dadurch können Anwendungen, die IP-Kommunikationen verwenden und darauf basieren (z. B. socketbasierte Anwendungen), auf Geräten innerhalb des Nicht-IP-Kommunikationsnetzwerks ausgeführt werden und nahtlos mit anderen Anwendungen auf Geräten kommunizieren, die sich innerhalb des Nicht-IP-Kommunikationsnetzwerks befinden oder extern mit dem Nicht-IP-Kommunikationsnetzwerk verbunden sind, indem IP-Datenframes genutzt werden. Dieser Vorteil sorgt für ein große Interoperabilität der über verschiedene Arten von Kommunikationsnetzwerken hinweg verwendeten Anwendungen.

[0012] Außerdem kann eine Schnittstelle zur Anwendungsprogrammierung (API) verwendet werden, um das Kommunikationsverfahren zu implementieren, indem der Programmierer die Funktionsaufrufe erhält, die in den Anwendungen genutzt werden können, um IP-Datenframes zu generieren und diese IP-Datenframes über die Netzwerkroutingmechanismen des Nicht-IP-Kommunikationsnetzwerk zu senden. Diese API erlaubt es Programmierern, socketbasierte Anwendungen innerhalb eines Nicht-IP-Kommunikationsnetzwerks auf effiziente Weise einzurichten und zu nutzen, um ein IP-basierte Kommunikation sowohl innerhalb als auch außerhalb eines Nicht-IP-Kommunikationsnetzwerks bereitzustellen. Außerdem kann eine Programmierumgebung, wie zum Beispiel ein Software Development Kit (SDK), einem Nutzer bei der Programmierung von Anwendungen für Geräte innerhalb des Nicht-IP-Kommunikationsnetzwerks helfen, indem sie dem Nutzer erlaubt, ein Gerät auszuwählen, das in einem Nicht-IP-Kommunikationsnetzwerk funktioniert, in dem die Anwendung laufen soll, um eine Anwendung für ein Gerät zu entwickeln, das IP-Datenframes generiert, diese IP-Datenframes über die Netzwerkroutingmechanismen des Nicht-IP-Kommunikationsnetzwerks zu senden und die Anwendung zu testen. Die Programmierumgebung kann Routinen enthalten, um die Anwendung nach ihrer Entwicklung in Form von IP-Datenframes an das Gerät im Nicht-IP-Kommunikationsnetzwerk zu übertragen.

BESCHREIBUNG DER ZEICHNUNGEN

[0013] Fig. 1 zeigt ein Beispiel für ein Kommunikationssystem, in dem Geräte in einem Nicht-IP-Kommunikationsnetzwerk IP-Datenframes in dem Netzwerk oder über verschiedene Kommunikationsnetzwerke hinweg empfangen und senden können, vom denen einige ein IP-Netzwerkroutingprotokoll unterstützen und einige dies nicht tun.

[0014] Fig. 2A zeigt die Erzeugung eines Standard-IP-Datenframes innerhalb eines Kommunikations-Stacks eines Netzwerkgerätes.

[0015] Fig. 2B zeigt die Erzeugung eines IP-Datenframes entsprechend des hier beschriebenen Verfahrens, um es dem IP-Datenframe zu ermöglichen, in einem Nicht-IP-Kommunikationsnetzwerk geroutet zu werden.

[0016] Fig. 3 zeigt ein Verfahren, IP-Datenframes zu erzeugen und über ein Nicht-IP-Kommunikationsnetzwerk zu senden, das kein IP-Netzwerkroutingprotokoll verwendet oder unterstützt.

[0017] Fig. 4A ist ein Ablaufschema eines Kommunikationsverfahren, das verwendet werden kann, um einen IP-Datenframe in einem ersten Netzwerk unter Nutzung eines ersten IP-Netzwerkroutingprotokolls zu erzeugen und diesen Datenframe an ein Gerät in einem zweiten Netzwerk zu senden, das ein Nicht-IP-Netzwerkroutingprotokoll verwendet, z. B. ein Mesh-Netzwerkroutingprotokoll oder ein implizites Ziel- oder Adressierungs-Routingprotokoll.

[0018] Fig. 4B ist ein Ablaufschema eines Kommunikationsverfahrens, das verwendet werden kann, um einen IP-Datenframe unter Nutzung eines Nicht-IP-Netzwerkroutingprotokolls zu erzeugen, z. B. ein Mesh-Netzwerkroutingprotokoll oder ein implizites Ziel- oder Adressierungs-Routingprotokoll, und diesen Datenframe an ein Gerät in einem Netzwerk zu senden, das ein IP-Netzwerkroutingprotokoll oder ein explizites Ziel- oder Adressierungs-Routingprotokoll verwendet.

[0019] Fig. 5 zeigt ein Beispiel eines WirelessHART-Netzwerks, in dem WirelessHART-Feldgeräte betrieben werden können, um IPv6-Datenframes innerhalb eines WirelessHART-Netzwerks und zwischen verschiedenen WirelessHART-Netzwerken zu empfangen und zu senden, wozu ein Kommunikationsnetzwerk genutzt wird, welches das IPv6-Netzwerkroutingprotokoll unterstützt oder implementiert.

[0020] Fig. 6 zeigt ein anderes Beispiel eines WirelessHART-Netzwerks, über das remote angeschlossene Computergeräte mit WirelessHART-Feldgeräten mittels IPv6-Datenframes kommunizieren können.

[0021] Fig. 7 ist ein Blockdiagramm eines Beispiels eines Netzwerkprotokoll-Stacks, der in einem Gateway-Gerät verwendet werden kann, um IPv6-Datenframes zu empfangen und zu senden, beispielsweise aus dem Internet, während gleichzeitig das Routing von IPv6-Datenframes in einem Nicht-IP-Kommunikationsnetzwerk unterstützt wird.

[0022] Fig. 8 ist ein Blockdiagramm eines Beispiels eines Netzwerkprotokoll-Stacks, der verwendet werden kann, um IPv6-Datenframes sowie WirelessHART-Datenframes von Geräten in einem WirelessHART-Kommunikationsnetzwerk zu empfangen und an sie zu senden.

[0023] Fig. 9 ist ein Blockdiagramm eines Beispiels eines 6LoWPAN-Blocks, der in einem Feldgerät oder in Gateways in einem WirelessHART-Kommunikationsnetzwerk implementiert werden kann, um das Routen von IPv6-Datenframes über ein Nicht-IP-Kommunikationsnetzwerk zu ermöglichen, z. B. dem WirelessHART-Netzwerk.

[0024] Fig. 10A ist ein Blockdiagramm eines Beispiels eines Fragmentierungsmoduls, das in das Beispiel des 6LoWPAN-Blocks aus **Fig.** 8 implementiert werden kann.

[0025] Fig. 10B ist ein Blockdiagramm eines Beispiels eines Zusammenbaumoduls, das in das Beispiel des 6LoWPAN-Blocks aus **Fig.** 8 implementiert werden kann.

[0026] Fig. 11A zeigt die Erzeugung von IPv6-Datenframe-Fragmenten von einem empfangenen IPv6-Datenframe, eingekapselt in den Protokollheader für das WirelessHART-Netzwerkrouting.

[0027] Fig. 11B zeigt ein Beispiel eines Datenfeldes im Protokollheader für das WirelessHART-Netzwerkrouting, das verwendet werden kann, um einen IPv6-Datenframe anzuzeigen.

[0028] Fig. 11C zeigt ein Beispiel eines Datenfeldes im Protokollheader für das WirelessHART-Netzwerkrouting, das verwendet werden kann, um mehrere mögliche Sicherheitsmechanismen anzuzeigen, die verwendet werden können, um einen IPv6-Datenframe zu empfangen und zu senden.

[0029] Fig. 12 zeigt ein Set von WirelessHART-Geräten in einem WirelessHART-Kommunikationsnetzwerk, das ein erstes Beispiel einer Kommunikationstechnik implementiert, um IPv6-Datenframes und WirelessHART-Datenframes innerhalb des WirelessHART-Kommunikationsnetzwerks zu routen.

[0030] Fig. 13 zeigt ein Set von WirelessHART-Geräten in einem WirelessHART-Kommunikationsnetzwerk, das ein zweites Beispiel einer Kommunikationstechnik implementiert, um IPv6-Datenframes und WirelessHART-Datenframes innerhalb eines WirelessHART-Kommunikationsnetzwerks zu routen.

[0031] Fig. 14 zeigt ein Set von WirelessHART-Geräten in einem WirelessHART-Kommunikationsnetzwerk, das ein drittes Beispiel einer Kommunikationstechnik implementiert, um IPv6-Datenframes und WirelessHART-Datenframes innerhalb eines WirelessHART-Kommunikationsnetzwerks zu routen, die keine IPv6-Datenframes oder IPv6-Netzwerkroutingmechanismen unterstützen.

[0032] Fig. 15 ist ein Blockdiagramm eines Beispiels eines SDK, das verwendet werden kann, um maßgeschneiderte Anwendungen zu erzeugen, die für die Nutzung der Schnittstelle zur Anwendungsprogrammierung (API) eines Feldgerätes konfiguriert sind, um so IPv6-Datenframes über ein WirelessHART-Kommunikationsnetzwerk zu empfangen und zu senden.

AUSFÜHRLICHE BESCHREIBUNG

[0033] Generell umfasst ein System und ein Verfahren zur nahtlosen Übertragung von Datenframes, die basierend auf einem ersten Netzwerkroutingprotokoll, z. B. IP-Datenframes, über mehrere unterschiedliche Kommunikationsnetzwerke erstellt wurden, das Erstellen oder Empfangen von Datenframes, die mit Netzwerkroutingprotokoll-Informationen konfiguriert wurden (auch als Netzwerk-Routinginformationen bezeichnet), die mit einem ersten Netzwerkroutingprotokoll verbunden sind, z. B. dem Internetprotokoll Version 6(IPv6)-Protokoll oder einem anderen expliziten oder Ziel- oder Adressierungs-Routingprotokoll oder Netzwerktopologie-Adressierungsprotokoll, wodurch Netzwerkroutingprotokoll-Informationen hinzugefügt werden, wie sie durch ein zweites Netzwerkroutingprotokoll festgelegt wurden, z. B. durch das WirelessHART-Netzwerkroutingprotokoll oder ein anderes Mesh-Netzwerk, ein implizites Ziel- oder Adressierungsprotokoll oder ein nicht auf der Netzwerktopologieadresse basierendes Routingprotokoll, während die erste Netzwerkroutinginformation be-

wahrt wird, und unter Verwendung der Netzwerkroutingmechanismen und Verfahren des zweiten Kommunikationsnetzwerks zur Kommunikation des Datenframes über das zweite Kommunikationsnetzwerk, während die Netzwerkroutinginformationen des ersten Netzwerkroutingprotokolls innerhalb des Datenframes bewahrt werden. Darüber hinaus kann das System und Verfahren den Datenframe decodieren, z. B. an einem Endgerät in dem zweiten Kommunikationsnetzwerk, wie beispielsweise an einem Gerät mit einer Anwendung, an die der Datenframe gerichtet ist, oder an einem Gateway-Gerät in dem zweiten Kommunikationsnetzwerk. Weiterhin kann es die Netzwerkroutinginformationen von dem ersten Netzwerkroutingprotokoll nutzen, um die Kommunikation des ursprünglichen Datenframes in dem Kommunikationsnetzwerk entsprechend an das erste Netzwerkroutingprotokoll zu leiten, z. B. dem Internet, oder um auf die Nachricht mit einem Datenframe zu antworten, der dem ersten Netzwerkroutingprotokoll entspricht.

[0034] Zum Hintergrund sei erwähnt, dass es generell zwei Arten von Netzwerkroutingprotokollen gibt, die hier als explizite Ziel-(auch als explizite Adressierungs-)Routingprotokolle bezeichnet werden, sowie implizite Ziel-(auch als implizite Adressierungs-)Routingprotokolle. Explizite Ziel- oder explizite Adressierungs-Netzwerkroutingprotokolle legen im Allgemeinen eines oder mehrere Ziele oder Zieladressen fest, die Teil der Netzwerkroutinginformationen in einem über das Netzwerk gerouteten Datenpaket sind, dessen Adressen für zwischengeschaltete Geräte innerhalb des Netzwerkes ausreichen, um so das Routing im Netzwerk auszuführen. Die Zielinformationen oder Zieladressen in den Netzwerkroutinginformationen in dem explizit gesendeten Paket (d. h. in und für sich selbst) legen das Gerät oder den logischen Ort (das Ziel) im Netzwerk fest, an das bzw. den das Paket geliefert werden soll. Diese Adressinformation wird verwendet (zusammen mit den Routingtabellen, die von Geräten im Netzwerk entwickelt wurden), um das Datenpaket an das mit der Zieladresse verbundene Zielgerät zu leiten. IP-Adressierungs- oder IP-Netzwerkroutingprotokolle sind besondere Beispiele für explizite Ziel- oder Adressierungsroutingprotokolle, da die IP-Zieladresse des Datenpakets im Allgemeinen die einzige Information ist, die vom Paket benötigt wird, damit Geräten im Netzwerk das Datenpaket im Netzwerk routen können. Tatsächlich verwendet die IP-Adressierung Adressen (z. B. Zieladressen), die eine Funktion der Route oder Netzwerktopologie sind, die notwendig ist, um das Gerät mit der Zieladresse zu erreichen. Das heißt, IP-Adressen werden als eine Funktion einer Position eines Gerätes (der die IP-Adresse zugewiesen ist) innerhalb der Topologie des Netzwerks erzeugt oder zugewiesen, um es Geräten zu ermöglichen oder sie zu unterstützen, ein Paket innerhalb eines Netzwerkes zu routen, um so das Paket an zwischengeschaltete Geräte zu senden, durch die das Zielgerät erreicht werden können. Daher basiert die IP-Adressierung auf der Netzwerktopologie oder hängt von ihr ab und wird hier auch als Netzwerktopologie-basiertes Adressierungsprotokoll bezeichnet. Viele Punkt-zu-Punkt-, drahtgebundenen oder Hochleistungs-Netzwerkroutingprotokolle (einschließlich IP-Routingprotokolle) nutzen explizites Zielrouting oder Zieladressrouting (einschließlich Netzwerktopologie-basierter Adressierung), da es generell wenig oder keine Bedenken gibt, dass ein Paket innerhalb des Netzwerks aufgrund von unzureichender Leistung in einem zwischengeschalteten Routinggerät oder durch den Verlust eines Kommunikationskanals zwischen Geräten innerhalb des Netzwerks aufgrund der Entfernungen zwischen den Geräten verloren geht (nicht geroutet wird).

[0035] Auf der anderen Seite nutzt implizites Zielrouting oder implizites Adressierungsrouting andere Informationen neben den Zieladressen in einem Datenpaket, um das Verfahren festzulegen, mit dem das Paket im Netzwerk geroutet wird. Einige implizite Zielroutingtechniken nutzen beispielsweise ein Konzept namens Graphen-Routing, in dem Graphen, die verschiedene Routen durch das Netzwerk definieren, festgelegt und dann verwendet werden, um das Verfahren anzuzeigen, mit dem ein bestimmtes Datenpaket durch das Netzwerk geleitet wird. Bei Einsatz des Graphen-Routing wird jedes Datenpaket mit Netzwerkroutinginformationen in Form einer Graphen-Identifizierung (Graph-ID) versehen, die den Graphen identifiziert, der verwendet werden soll, um innerhalb des Netzwerks das Routing für ein bestimmtes Paket durchzuführen. In einigen Fällen umfassen Netzwerk-Routing-Protokolle, die Graphen-Routing verwenden, auch eine oder mehrere Zieladressen innerhalb der Netzwerkroutinginformationen in einem Paket, aber diese Adressen werden in Verbindung mit der Graph-ID oder anderen Informationen in den Netzwerkroutinginformationen des Pakets verwendet, und sind daher selbst nicht ausreichend für die Durchführung des Routings innerhalb des Kommunikationsnetzwerks. Natürlich ist das Graphen-Routing nur ein Beispiel für eine implizite Zielroutingtechnik.

[0036] In der Tat verwenden viele drahtlose Mesh-Netzwerke implizite Netzwerkroutingtechniken, wie beispielsweise Graphen-Routing-Techniken, da diese Netzwerke sicherstellen müssen, dass die Datenpakete in einer bestimmten Art und Weise durch das Netzwerk geroutet werden, um die Verbindung sicherzustellen, ohne dass eine Überlastung der Kommunikations-Stacks oder der Leistung (Batterie) der Geräte im Netzwerk stattfindet, und weil diese Netzwerke entweder keine Zieladressen für das Routing verwenden, oder wenn sie es tun, keine Zieladressen in einer Weise vergeben, die von der Position des Geräts innerhalb des Netzwerks abhängig ist (die Adressen sind nicht von der Netzwerktopologie abhängig), wodurch es schwierig ist, Netzwerk-Routing nur auf der Grundlage der Zieladresse durchzuführen. Aus diesen Gründen verwenden die

meisten Mesh-Netzwerkroutingprotokolle kein explizites Zielrouting, wie zum Beispiel IP-Adressrouting, und sind mit solchen Routingprotokollen tatsächlich inkompatibel.

[0037] Allgemein gesagt verwendet ein Mesh-Netzwerk ein(e) Mesh-Netzwerk(topologie), die eine Art Netzwerk ist, wobei jeder Knoten nicht nur seine eigenen Daten erfassen und verbreiten muss, sondern auch als Relais für andere Knoten dient; daher müssen alle Knoten zusammenarbeiten, um die Daten im Netzwerk zu verbreiten. Ein Wireless Mesh-Netzwerk (WMN) ist ein Kommunikationsnetzwerk, das aus Funkknoten besteht, die in einer Mesh-Topologie organisiert sind. Wireless Mesh-Netzwerke umfassen oft Mesh-Clients, Mesh-Routers und Gateways. Bei den Mesh-Clients handelt es sich oft um Laptops, Mobiltelefone und andere drahtlose Geräte, während Mesh-Router Traffic von und zu Gateways weiterleiten, die mit dem Internet verbunden sein können, aber nicht müssen. Der Versorgungsbereich der Funkknoten, die als ein einzelnes Netzwerk operieren, wird manchmal als Mesh Cloud bezeichnet. Der Zugriff auf diese Mesh Cloud ist davon abhängig, dass die Funkknoten harmonisch miteinander daran arbeiten, ein Funknetzwerk herzustellen. Ein Mesh-Netzwerk ist verlässlich und bietet Redundanz. Wenn ein Knoten nicht länger arbeitet, können die restlichen Knoten immer noch miteinander kommunizieren, entweder direkt oder durch einen oder über mehrere zwischengeschaltete Knoten. Beispiele von Techniken für das Mesh-Netzwerkrouting beinhalten das Adhoc-On-Demand Distance Vector Routing, das von ZigBee-Netzwerken verwendet wird, und Graphen-Routing, das in WirelessHART-Netzwerken genutzt wird. In AODV (Adhoc-On-Demand Distance Vector) sendet ein Gerät eine Route-Anfrage an alle seine Nachbarn, um das Zielgerät zu finden. Die Nachbarn senden dann die Anfrage an ihre Nachbarn usw., bis das Ziel erreicht ist. Sobald das Ziel erreicht ist, sendet es seine Route-Antwort mittels einer Unicast-Übertragung über den Pfad, der mit den geringsten Kosten verbunden ist, zurück an die Quelle. Sobald die Quelle die Antwort erhält, aktualisiert sie die Routing-Tabelle für die Zieladresse mit dem nächsten Hop im Pfad und den Pfadkosten. Pfadkosten können die Anzahl der Hops beinhalten, was der Entfernung entspricht.

[0038] Bezugnehmend auf **Fig.** 1 wird ein Beispiel für ein Kommunikationssystem 5 gezeigt, das verwendet werden kann, um ein Kommunikationsverfahren zu implementieren, das die Netzwerkroutinginformationen von mehreren unterschiedlichen Netzwerkroutingprotokollen verwendet, um Datenframes oder Datennachrichten innerhalb von oder über mehrere unterschiedliche physische Netzwerke hinweg mit unterschiedlichen Kommunikationsnetzwerken zu routen, denen verschiedene Kommunikationsnetzwerke zugeordnet sind. In diesem Beispiel wird das Kommunikationsverfahren zum Routen verwendet oder um Datenframes aus einem ersten Netzwerkroutingprotokoll, wie z. B. einem explizitem Zielroutingprotokoll oder einem auf der Adresse der Netzwerktopologie basierenden Protokoll, z. B. einem IP-Protokoll und insbesondere einem IPv6-Netzwerkprotokoll, über ein oder mehrere Kommunikationsnetzwerke zu senden, die kein Routing mit dem ersten Netzwerkroutingprotokoll unterstützen, wie beispielsweise einem impliziten Zielroutingprotokoll wie einem Mesh-Netzwerkroutingprotokoll, oder einem auf der Adresse der Netzwerktopologie basierenden Adressierungsprotokoll, wie z. B. einem WirelessHART-Kommunikationsnetzwerk. Somit ist generell das erste Netzwerkroutingprotokoll der hier beschriebenen Beispiele ein IP-Netzwerkroutingprotokoll, beispielsweise ein IPv4- oder ein IPv6-Netzwerkroutingprotokoll, das IP-Adressierung verwendet (was ein explizites Zielroutingprotokoll und ein auf der Adresse der Netzwerktopologie basierenden Routingprotokoll ist), und diese IP-Adressierung kann unter anderem ein TCP/IP-Protokoll, ein UDP/IP-Protokoll oder ein ICMP/IP-Protokoll sein (alle hier allgemein als IP-Netzwerkroutingprotokolle bezeichnet). Darüber hinaus wird das zweite Netzwerkroutingprotokoll in den unten aufgeführten Beispielen als implizites Zielroutingprotokoll oder ein Nicht-IP-Netzwerkroutingprotokoll beschrieben, z. B. ein Netzwerkroutingprotokoll, das keine IP-Adressen, kein IP-Routing oder keine auf der Adresse der Netzwerktopologie basierenden Routingprotokolle verwendet oder unterstützt. Beispiele solcher Protokolle umfassen die WirelessHART-Protokolle, das ZigBee-Protokoll usw. Jedoch versteht es sich, dass andere Arten und besondere Beispiele von Netzwerkroutingprotokollen als die ersten und zweiten hier beschriebenen Netzwerkroutingprotokolle verwendet werden können, und dass die ersten und zweiten hier beschriebenen Netzwerkroutingprotokolle nicht auf IP- und Nicht-IP-Protokolle beschränkt sind. Zum Beispiel könnten die hierin beschriebenen Kommunikations- und Routing-Techniken verwendet werden, damit Datenpakete eines ersten impliziten Ziel-Routingprotokolls oder Nicht-IP-Netzwerkroutingprotokolls oder nicht auf der Adresse der Netzwerktopologie basierenden Protokolls über ein Kommunikationsnetzwerk gesendet werden können, das ein zweites implizites Ziel-Routing-Protokoll oder Nicht-IP-Netzwerkroutingprotokoll oder ein nicht auf der Adresse der Netzwerktopologie basierendes Protokoll verwendet. Als weiteres Beispiel könnten die hierin beschriebenen Kommunikations- und Routing-Techniken verwendet werden, damit die Datenpakete eines ersten expliziten Zielroutingprotokolls, das kein IP-Netzwerkroutingprotokoll ist, über ein Kommunikationsnetzwerk gesendet werden können, das ein zweites explizites Ziel-Routingprotokoll (IP- der nicht-IP-basiert) verwendet oder verwenden könnte, damit Datenpakete eines ersten impliziten Ziel-, Nicht-IP- oder nicht auf der Adresse der Netzwerktopologie basierenden Netzwerkroutingprotokolls über ein Kommunikationsnetzwerk gesendet werden können, das ein zweites explizites Ziel-Routingprotokoll verwendet, das beispielsweise kein P-Netzwerk-Routingprotokoll ist. Natürlich könnten auch andere Kombinationen von ersten und zweiten Typen von

Netzwerken verwendet werden, um die hierin beschriebenen Kommunikations- und Routing-Techniken zu implementieren.

[0039] Im Allgemeinen umfasst das Kommunikationssystem **5** aus **Fig.** 1 ein IP-Kommunikationsnetzwerk in Form des Internet **10**, das mit einem oder mehreren anderen physischen Netzwerken verbunden ist, die entweder IP-Netzwerkroutingprotokolle oder Nicht-IP-Netzwerkroutingprotokolle implementieren können. In diesem Fall wird davon ausgegangen, dass die Kommunikation über das Internet-Netzwerk **10** ein IPv6-Netzwerkroutingprotokoll verwendet, aber die Kommunikation über das Internet **10** kann zusätzlich oder stattdessen ein IPv4-Netzwerkroutingprotokoll oder ein anderes IP-Routing-Protokoll verwenden. Die Kommunikationsnetzwerke, die in **Fig.** 1 als mit dem Internet **10** verbunden dargestellt sind, umfassen ein WirelessHART-Netzwerk **12**, ein ZigBee-Netzwerk **13**, ein Bluetooth Piconet **14**, ein Mobilfunknetzwerk **15** und ein drahtloses lokales Netzwerk(WLAN)/lokales Netzwerk (LAN) **16**. Es versteht sich, dass die Netzwerke **12**–**16** verschiedene Netzwerkroutingprotokolle (hier auch als Routingprotokolle bezeichnet) implementieren oder nutzen können, um die Netzwerkkommunikation darin durchzuführen, und sie können unterschiedliche Datenlink- und physische Layerprotokolle verwenden, um diese Kommunikation durchzuführen. In vielen Fällen, beispielsweise im Bluetooth-Netzwerk, dem WLAN/LAN-Netzwerk und dem Mobilfunknetzwerk, kann ein Kommunikationsnetzwerk IP-Netzwerkrouting oder Netzwerktopologie-Adressierungsrouting oder ein anderes explizites Zielroutingprotokoll unterstützen, während in anderen Fällen, wie beim WirelessHART-Netzwerk, ein Kommunikationsnetzwerk kein IP-Netzwerkroutingprotokoll oder Netzwerktopologie-Adressierungsrouting oder andere explizite Zielroutingprotokolle nutzen oder unterstützen kann, da dieses Netzwerk keine IP-Adressierung für Netzwerk-Routingaktivitäten nutzen kann. Zum Beispiel verwendet das WirelessHART-Netzwerk wie erwähnt typischerweise ein Konzept namens Graphen-Routing, um das Routing von Nachrichten in dem WirelessHART-Netzwerk durchzuführen.

[0040] Das nachstehend beschriebene Kommunikationsverfahren kann verwendet werden, damit ein Gerät innerhalb des WirelessHART-Netzwerks **12** IPv6- oder IPv4-Datenframes erzeugen und diese Datenframes über das WirelessHART-Netzwerk **12** kommunizieren kann, wozu das Netzwerkroutingprotokoll des WirelessHART-Netzwerks **12** genutzt wird, das ein Nicht-IP-Netzwerkroutingprotokoll ist, da dieses Netzwerkroutingprotokoll keine IP-Adressierung für Routing-Zwecke verwendet oder unterstützt. Außerdem versteht es sich, dass das WirelessHART-Netzwerk ein Mesh-Netzwerkroutingprotokoll implementiert, während das Internet **10** und ein oder mehrere andere Kommunikationsnetze **13**–**16** eventuell ein Nicht-Mesh-Netzwerkroutingprotokoll, wie beispielsweise ein Punkt-zu-Punkt-Routingprotokoll, verwenden. In jedem Fall können diese IPv6-Datenframes mit diesem Kommunikationsverfahren an andere Geräte innerhalb des WirelessHART-Netzwerks **12** gesendet werden oder von diesen oder durch Geräte in einem oder mehreren anderen Kommunikationsnetzwerken **13**–**16** decodiert werden, wie **Fig.** 1 zeigt. In der gleichen Weise kann das nachfolgend beschriebene Kommunikationsverfahren verwendet werden, damit ein Gerät in dem WirelessHART-Netzwerk **12** IPv6-Datenframes als IPv6-Datenframes empfangen und decodieren kann, und somit diese Datenframes verwenden und beantworten kann, und das im Wesentlichen in der gleichen Weise, als ob das Gerät mit einem entsprechenden IP-Netzwerkrouting-Kommunikationsnetzwerk verbunden ist (d. h. mit einem Kommunikationsnetzwerk, das IP-Netzwerkroutingmechanismen verwendet). In diesem Fall kann das Empfangsgerät im WirelessHART-Netzwerk **12** die IPv6-Datenframes von einem anderen Gerät innerhalb des WirelessHART-Netzwerks **12** oder von einem Gerät in einem der anderen Netzwerke **13**–**16** von **Fig.** 1 empfangen.

[0041] Im Allgemeinen verbindet ein IPv6-fähiges WirelessHART-Gateway **11a** das WirelessHART-Netzwerk **12** mit dem Internet **10**, um das hier beschriebene Kommunikationsverfahren zu implementieren. Ebenso, wie in **Fig.** 1 dargestellt, verbinden andere Gateway-Geräte **11b**, **11c**, **11d** und **11e** jeweils das ZigBee-Netzwerk **13**, das Bluetooth Piconet **14**, das Mobilfunknetzwerk **15** und das WLAN/LAN **16** mit dem Internet **10**. Die Gateway-Geräte **11** sowie ein oder mehrere Geräte innerhalb der Kommunikationsnetzwerke **12**–**16** können Schritte oder Aspekte des im Folgenden näher beschriebenen Kommunikationsverfahrens implementieren, damit beispielsweise eine IP-basierte Kommunikation zwischen zwei Geräten (oder Anwendungen) innerhalb des WirelessHART-Netzwerks **12** oder zwischen einem ersten Gerät (oder einer Anwendung) im WirelessHART-Netzwerk **12** und einem zweiten Gerät (oder einer Anwendung) in einem anderen der Netzwerke **13**–**16** möglich ist.

[0042] Insbesondere können die Geräte innerhalb des WirelessHART-Netzwerks **12** (einschließlich des Gateway-Geräts **11a**) Datenframes über die physischen Layer des WirelessHART-Netzwerks **12** empfangen und senden, indem Datenframes verwendet werden, die so konfiguriert sind, dass sie sowohl Nicht-IP-Netzwerkroutinginformationen als auch IP-Netzwerkroutinginformationen umfassen, und so in der Lage sind, IP-Datenframes nahtlos aus einem und an ein Kommunikationsnetzwerk zu senden, das ein Netzwerkroutingprotokoll und/oder ein Datenlinklayerprotokoll verwendet, das keine Kommunikation per IP-Netzwerkroutingpro-

tokoll verwendet. Beispielsweise kann das IPv6-fähige WirelessHART-Gateway **11a** in **Fig.** 1 Datenframes mit Informationen, die gemäß IPv6-Netzwerkroutingprotokoll eingekapselt sind, über das Internet **10** empfangen, wobei solche Datenframes über die Kommunikationskanäle im Internet **10** übertragen werden, wozu ein 802.3 Datenlinklayer-Protokoll und/oder ein Standard- oder unterstütztes physisches Layer innerhalb des Internets **10** verwendet wird. Die vom Internet **10** über das IPv6-fähige WirelessHART-Gateway **11a** empfangenen Datenframes können ursprünglich von Anwendungen in Geräten in einem ZigBee-Netzwerk **13**, einem Bluetooth Piconet **14**, einem Mobilfunknetzwerk **15** oder dem WLAN/LAN **16** übertragen oder erzeugt worden sein, beispielsweise als Standard-IPv6-Datenframes. Das IPv6-fähige WirelessHART-Gateway **11a** kann dann die empfangenen IPv6-Datenframes an Netzwerkpositionen übertragen (z. B. an Anwendungen) innerhalb von Geräten in dem WirelessHART-Netzwerk **12**, die ein WirelessHART-konformes Netzwerkroutingprotokoll nutzen, um dadurch IPv6-Nachrichten oder Datenframes zwischen Geräten innerhalb eines der Netzwerke **13**–**16** und einem Gerät im WirelessHART-Netzwerk **12** zu übertragen. In ähnlicher Weise kann das WirelessHART-Gateway **11a** Datenframes oder Nachrichten von IP-fähigen Geräten innerhalb des WirelessHART-Netzwerks **12** empfangen, die ursprünglich nach dem IPv6-Netzwerkroutingprotokoll eingekapselt waren und dann modifiziert und über das WirelessHART-Netzwerk **12** unter Verwendung eines hier detaillierter beschriebenen WirelessHART-Netzwerkroutingprotokolls geroutet wurden. Das WirelessHART-Gateway **11a** kann diese Datenframes oder Nachrichten über einen 802.15.4 Datenlinklayer und ein physisches Protokoll empfangen, das von dem WirelessHART-Netzwerk **12** benutzt wird. Das IPv6-fähige WirelessHART-Gateway **11a** kann dann die IP-Netzwerkroutinginformationen innerhalb der empfangenen Nachricht oder des Datenframes verwenden, um die Datenframes unter Verwendung von IP-Netzwerkrouting oder -adressierung an ein Gerät in einem der anderen Netzwerke **13** bis **16** zu senden. Natürlich sendet in diesem Fall das IPv6-fähige WirelessHART-Gateway **11a** die empfangenen IPv6-Datenframes an Anwendungen oder Geräte über das Internet **10**, indem das standardmäßige 802.3 physische Layer-Kommunikationsprotokoll und die IPv6-Netzwerkprotokolltechniken verwendet werden.

[0043] Es versteht sich, dass die Geräte im WirelessHART-Netzwerk **12** Datenframes über Kommunikationskanäle entsprechend dem Standard 802.15.4 empfangen und senden. Wie im Folgenden detaillierter beschrieben, können jedoch das IPv6-fähige WirelessHART-Gateway **11a** oder andere IP-fähige Geräte im WirelessHART-Netzwerk **12** beim Senden eines IPv6-Datenframes über das WirelessHART-Netzwerk **12** die IPv6-Datenframes in mehrere Fragmente aufteilen, beispielsweise entsprechend dem 6LoWPAN-Standard. In diesem Fall können die Geräte die IPv6-Datenframe-Fragmente über die Kommunikationskanäle entsprechend dem WirelessHART-Netzwerkroutingprotokoll und dem Datenlinklayerprotokoll 802.15.4 an WirelessHART-Geräte übertragen. In diesem Fall kann das IPv6-fähige WirelessHART-Gateway **11a** die IPv6-Datenframe-Fragmente mit Header-Informationen entsprechend dem WirelessHART-Netzwerkroutingprotokoll einkapseln, um diese IPv6-Datenframe-Fragmente über das WirelessHART-Netzwerk **12** zu senden. Danach können im WirelessHART-Protokoll-Header enthaltene Informationen von einem oder mehreren WirelessHART-Geräten im WirelessHART-Netzwerk **12** genutzt werden, um die IPv6-Datenframes zum vorgesehenen WirelessHART-Gerät oder zur gewünschten Anwendung zu routen. Bezeichnenderweise können WirelessHART-Geräte im WirelessHART-Netzwerk **12** weiterhin kommunizieren oder Nicht-IP-Anwendungsdaten oder Payloads austauschen. Dies fördert die Abwärtskompatibilität und ermöglicht gleichzeitig Nicht-IP-Kommunikation zwischen Netzwerken und IP-Kommunikation sowohl in und zwischen Netzwerken.

[0044] Damit WirelessHART-Geräte im Netzwerk **12** erkennen können, ob eine bestimmte über das Netzwerk geroutete WirelessHART-Nachricht oder ein entsprechender Datenframe eine Standard WirelessHART-Nachricht oder eingekapselte IPv6-Datennachricht ist, kann das IPv6-fähige WirelessHART-Gateway **11a** oder ein anderes Gerät, das die Nachricht erzeugt, ein Flag in den WirelessHART-Netzwerkroutinginformationen des Datenframes enthalten, der angibt, ob dieser Datenframe ein IPv6-Datenframe oder ein Standard-WirelessHART-Datenframe ist. Das Flag kann ein oder mehrere Datenfelder in den Informationen für das WirelesssHART-Netzwerkrouting umfassen. Vorzugsweise, um die Abwärtskompatibilität und Interoperabilität zwischen nicht-IP-aktivierten und IP-fähigen WirelessHART-Geräten zu fördern, kann das Flag ein zuvor ungenutztes Datenfeld in den Informationen für das WirelessHART-Netzwerkrouting nutzen.

[0045] Es versteht sich, dass, wenn die IPv6-Datenframes fragmentiert und über das WirelessHART-Netzwerk **12** gesendet werden, das IPv6-fähige WirelessHART-Gateway **11a** oder andere Geräte innerhalb des WirelessHART-Netzwerks **12** in **Fig.** 1 die IPv6-Datenframe-Fragmente empfangen, die in Informationen für das WirelessHART-Netzwerkrouting-Protokoll eingekapselt sind, über Kommunikationskanäle entsprechend dem Kommunikationsprotokoll 802.15.4. Die IPv6-Datenframe-Fragmente können, wie oben erwähnt, dem 6LoWPAN-Standard entsprechen. Das IPv6-fähige WirelessHART-Gateway **1la** oder andere WirelessHART-Geräte können dann ein Flag in den WirelessHART-Netzwerkroutinginformationen eines Datenframefragments erkennen, das angibt, dass der Datenframe einem IPv6-Datenframe entspricht. In diesem Fall fügt das IPv6-

fähige WirelessHART-Gateway **11a** oder ein anderes WirelessHART-Gerät dann die 6LoWPAN-Datenframes zusammen, um einen IPv6-Datenframe zu erzeugen. Das WirelessHART-Gateway **11a** kann dann die IP-Zieladressinformationen der IPv6-Datenframes und der Software und Hardware für das Standard-IP-Routing verwenden, um die IPv6-Datenframes an ein Gerät im ZIGBEE-Netzwerk **13**, dem Bluetooth Piconet **14**, dem Mobilfunknetzwerk **15** und/oder dem WLAN/LAN **16** über die Standard-IPv6-Datenkommunikation zu senden. Wenn die IP-Adresse des zusammengesetzten Datenframes eine Anwendung innerhalb des Empfangsgerätes anzeigt, wird der Datenframe natürlich an die Anwendung geliefert und gemäß IPv6-konformen Standardtechniken als ein IPv6-Datenframe verarbeitet.

[0046] Während die IPv6-fähige Kommunikation generell so beschrieben wurden, wie sie in **Fig.** 1 in Bezug auf das Nicht-IP-Netz als WirelessHART-Netzwerk **12** auftreten, versteht es sich, dass die gleichen Techniken verwendet werden könnten, um IPv6-basierte Kommunikation in anderen Nicht-IP-Netzwerken, wie z. B. im ZigBee-Netzwerk **13**, dem Bluetooth Piconet **14**, dem Mobilfunknetzwerk **15** und/oder dem WLAN/LAN **16**, durchzuführen (wenn diese Netzwerke beispielsweise Nicht-IP-Netzwerkroutingprotokolle verwenden), indem die hier beschriebenen Techniken genutzt werden, mit der Ausnahme, dass in diesen Fällen die Geräte in den Netzwerken **13**–**16** das Protokoll für physische Layer, Datenlinklayer und Netzwerkroutinglayer nutzen, die durch die anderen Netzwerke anstelle der WirelessHART-Protokolle für physische Layer, Datenlinklayer und Netzwerkroutinglayer unterstützt oder festgelegt werden.

[0047] Um die besondere Weise vollständiger zu beschreiben, in der IP-Datenframes codiert und über Nicht-IP-Netzwerke geroutet werden, d. h. Netzwerke, die ein IP-Netzwerkroutingprotokoll verwenden, um Netzwerkrouting durchzuführen, hilft es, die allgemeine Weise zu beschreiben, in der Datenframes im Allgemeinen und IP-Datenframes im Besonderen erzeugt und über Kommunikationsnetzwerke geroutet werden. Zunächst ist es wichtig zu beachten, dass es eine Reihe von verschiedenen Layern oder Levels gibt, in denen Datenframes erzeugt und geroutet werden, und es ist notwendig, die Art und Weise zu verstehen, in der diese verschiedenen Level miteinander verbunden sind und in den unterschiedlichen Netzwerk- und Kommunikationsprotokollen verwendet werden, die gegenwärtig zur vernetzten Kommunikation genutzt werden. Generell beinhaltet ein Datenframe, der über ein Kommunikationsnetzwerk gesendet wird, Informationen, die mit einer Reihe von Kommunikations-Stacklayern verbunden sind oder durch sie bereitgestellt werden, typischerweise definiert durch das Open System Interconnection(OSI)-Modell.

[0048] Generell definiert das OSI-Modell einen Vernetzungsrahmen für die Umsetzung von Netzwerkprotokollen mit sieben grundlegenden Layern. Die Kontrolle wird von einem Layer zum nächsten weitergegeben, beginnend bei dem obersten Layer in einer Station und bis zum untersten Layer in dieser Station, um eine Nachricht über den Kommunikationskanal an die nächste Station zu senden. An der Empfangsstation erfolgt die Kontrolle in dem untersten Layer und durchläuft die Hierarchie zurück zum obersten Layer, um die Decodierung der empfangenen Nachricht durchzuführen. Generell beinhalten die sieben Layer des OSI-Modells den Anwendungslayer (oberster Layer, als Layer 7 bezeichnet), den Darstellungslayer (Layer 6), den Sessionlayer (Layer 5), den Transportlayer (Layer 4), den Netzwerklayer (Layer 3), den Datenlinklayer (Layer 2) und den physischen Layer (Layer 1). Im Allgemeinen unterstützt der Anwendungslayer Anwendungen und Endanwenderprozesse. Alles auf dem Anwendungslayer ist anwendungsspezifisch, und der Anwendungslayer (oder die Anwendung auf dem Anwendungslayer) kann Kommunikationspartner definieren, Servicequalität, Benutzerauthentifizierung, Datenschutz, Einschränkungen für Datensyntax usw. Dieser Layer bietet auch Anwendungsdienste für Datei-Übertragungen, E-Mail und andere Netzwerkdienste. Der Darstellungslayer, manchmal auch Syntaxlayer genannt, bietet Unabhängigkeit von Unterschieden in der Datenrepräsentation (z. B. Verschlüsselung) durch die Übersetzung vom Anwendungsformat ins Netzwerkformat und umgekehrt. Somit arbeitet der Darstellungslayer daran, Daten in die Form umzuwandeln, die der Anwendungslayer akzeptieren kann, zum Beispiel wenn diese Daten codiert oder verschlüsselt wurden. Dieser Layer formatiert und verschlüsselt somit Daten, um sie über ein Netzwerk hinweg zu senden, das die Kompatibilitätsprobleme reduziert oder eliminiert.

[0049] Der Sessionlayer errichtet, verwaltet und beendet Verbindungen zwischen Anwendungen, und er errichtet, koordiniert und beendet generell Gespräche, Austausche und Dialoge zwischen den Anwendungen an jedem Ende der Kommunikation. Der Sessionlayer befasst sich auch mit der Koordinierung von Sitzung und Verbindung. Der Transportlayer sorgt für einen transparenten Transfer von Daten zwischen Endsystemen oder Hosts und ist für die End-zu-End-Fehlerkorrektur und Flusssteuerung verantwortlich. Dieser Layer stellt den vollständigen Datentransfer sicher. Der Netzwerklayer beschreibt, wie eine Reihe von Austauschvorgängen über mehrere Datenverbindungen Daten zwischen zwei beliebigen Knoten in einem Netzwerk liefern können. Der Netzwerklayer definiert die Adressierung, das Umschalten und die Routingstruktur des Netzwerks, und er erzeugt oder definiert logischen Pfade, bekannt als virtuelle Schaltungen, zur Übertragung von Daten von Knoten zu Knoten. Es ist wichtig, dass der Netzwerklayer Funktionen für das Netzwerkrouting und für die Wei-

terleitung definiert sowie Netzwerk-Adressierungsschemata, Internet, Fehlerbehandlung, Überlastungssteuerung und Paket-Sequenzierung.

[0050] Der Datenlinklayer codiert und decodiert Datenpakete aus und in Bits, und er beschreibt die logische Organisation der auf ein bestimmtes Medium übertragenen Datenbits. Dieser Layer definiert das Framing, die Adressierung und Prüfsummenverfahren (Fehlerbehandlung) von Paketen auf dem physischen Layer. Der Datenlinklayer ist in der Regel in zwei Sublayer unterteilt, einschließlich des Media Access Control (MAC) Layers und des Logical Link Control (LLC) Layers. Der MAC-Sublayer steuert, wie ein Computer in dem Netzwerk Zugriff auf die Daten und die Erlaubnis zur Übertragung dieser Daten erhält. Der LLC-Layer steuert Frame-Synchronisierung, Flusssteuerung und Fehlerprüfung. Der physische Layer übermittelt den Bitstrom, zum Beispiel über elektrische Impulse, Licht- oder Funksignale, auf elektrischer und mechanischer Ebene über das Netzwerk. Der physische Layer stellt auf einem Träger die Hardware zur Verfügung, die zum Senden und Empfangen von Daten benötigt wird, einschließlich der Definition der Kabel, Karten und physischen Aspekte der Netzwerk-Hardware.

[0051] Es ist wichtig, dass eine Reihe von Protokollen für einige dieser Layer definiert wird, einschließlich insbesondere für den Transportlayer (Layer 4), den Netzwerklayer (Layer 3), den Datenlinklayer (Layer 2) und den physischen Layer (Layer 1). Beispielsweise gibt es verschiedene Transportlayerprotokolle, darunter Universal Datagram Protocol (UDP), Transmission Control Protocol (TCP) und Internet Control Message Protocol (ICMP). Diese Protokolle stellen besondere Transportlayer-Regeln und Verfahren bereit, die beispielsweise mit einem IP-Netzwerklayerprotokoll verwendet werden. Außerdem gibt es verschiedene Netzwerklayerprotokolle, einschließlich der vorherrschenden IP-Protokolle, wie IPv4- und IPv6-Netzwerklayerprotokolle, sowie viele Nicht-IP-Netzwerklayerprotokolle, einschließlich der WirelessHART-Netzwerklayerprotokolle. Netzwerklayerprotokolle definieren allgemein die Regeln und Verfahren, um Nachrichten oder Datenframes von einem logischen Punkt im Netzwerk zu einem anderen logischen Punkt im Netzwerk zu routen, und um außerdem die Netzwerklayerinformationen zu definieren, die in dem Datenpaket enthalten sein müssen, um das Netzwerkrouting zu ermöglichen. Netzwerklayerprotokolle werden hier auch als Netzwerkroutingprotokolle bezeichnet. Weiterhin wurden verschiedene physische Layer- oder Datenlinklayerprotokolle definiert, einschließlich beispielsweise des Datenlinklayerprotokolls 802.3 (Ethernet), des Protokoll des physischen Layers 802.15.4 (z. B. verwendet vom physischen WirelessHART-Layerprotokoll), usw. Es versteht sich, dass verschiedene Netzwerke das gleiche Netzwerklayerprotokoll und unterschiedliche Datenlinklayerprotokolle oder umgekehrt verwenden können. Darüber hinaus sind bestimmte Datenlinklayerprotokolle mit bestimmten Netzwerklayerprotokollen möglicherweise nicht kompatibel. Eine häufige Ursache für dieses Problem ist, dass das Netzwerklayerprotokoll eine maximale Nachrichtengröße oder -länge festlegen kann, die länger ist als die von einem Datenlinklayerprotokoll erlaubte Nachricht.

[0052] Obwohl die hier beschriebenen Kommunikationstechniken die oben beschriebenen mehreren Layer verwenden können, können in einigen Implementierungen die Kommunikationsverfahren den ”Layer”-Ansatz vermeiden und eine Funktionalität implementieren, die nachstehend als einzelner monolithischer Anweisungsblock beschrieben wird. Dieser Ansatz ist bei der Implementierung von Kommunikationsverfahren in Geräten mit begrenztem Speicher besonders attraktiv. Doch der einzige monolithische Anweisungsblock enthält immer noch Elemente oder Stack-Elemente, die in dem hier beschriebenen Layer-Ansatz verwendet werden.

[0053] Gewöhnlich können ein Computer und/oder Anwendungen im Computer Netzwerk- oder Kommunikations-Stacks nutzen, um die verschiedenen Layer des OSI-Modells zu implementieren oder nach unten zu durchlaufen, um eine Nachricht zu erzeugen (codieren), damit sie über einen physischen Layer des Kommunikationsnetzwerk gesendet wird, mit dem das Gerät verbunden ist, oder sie laufen nach oben durch die verschiedenen Layer des OSI-Modells, um eine im physischen Layer des Kommunikationsnetzwerks empfangene Nachricht zu decodieren. Der Kommunikations-Stack des Geräts kann somit in einem Prozessor ausgeführte Anwendungen umfassen, die Verfahren implementieren, die von den verschiedenen Protokollen auf den verschiedenen Layern des Stacks definiert wurden, um Datenframes gemäß den jeweiligen Protokollen zu packen, bevor die Datenframes gesendet werden, oder um die im Gerät empfangenen Datenframes zu entpacken oder zu decodieren, um eine decodierte Nachricht für eine Anwendung bereitzustellen, an welche die Nachricht gerichtet ist. Das Packen eines Datenframes entsprechend einem Protokoll kann das Anhängen von Header- und/oder Trailer-Informationen umfassen, wie durch das Protokoll des Layers festgelegt, der aktuell auf dem Datenframe läuft, und diese Datenframes werden an den nächstniedrigeren Layer des Stacks übergeben. Ebenso umfasst das Entpacken des Datenframes das Entfernen und Decodieren von Header- und/oder Trailer-Informationen aus dem Datenframe für den betreffenden Layer des Stacks und die Übergabe des entpackten Datenframes an den nächsthöheren Layer des Stacks. Die Informationen im Header- und/oder Trailer-Abschnitt eines Datenframes für einen bestimmten Layer im Stack entsprechen typischerweise den

Regeln, die vom jeweiligen Protokollstandard im entsprechenden Layer festgelegt wurden. Zum Beispiel umfasst das Packen des Datenframes gemäß dem TCP/IP-Protokoll das Anhängen eines TCP-Headers an den Datenframe (am Transportlayer des Stacks) und folglich das Anhängen eines IP-Headers an den entstehenden Datenframe (am Netzwerklayer des Stacks). Der TCP-Header entspricht zum Beispiel dem TCP-Protokoll und der IP-Header dem IPv4- oder IPv6-Standard. Einzelheiten zum IPv6-Protokoll werden in einer Spezifikation beschrieben, die zu finden ist unter http://www.ietf.org/rfc/rfc2460.txt. Beispiele für andere Netzwerklayerprotokolle umfassen das WirelessHART-Protokoll, das ZigBee-Netzwerklayerprotokoll usw. Natürlich kann ein gemäß einem bestimmten Netzwerkprotokoll verpackter Datenframe über Kommunikationskanäle versendet werden, die einem beliebigen Protokoll aus einem Satz von Datenlinklayerprotokollen oder physischen Layerprotokollen entsprechen, wie dem 802.3-Standard (Ethernet), dem 802.11 a/b/g/n-Standard (Wi-Fi), dem 802.15.4-Standard, dem 802.16-Standard (Wimax), Mobilfunk-Standards wie CDMA, GSM, LTE usw.

[0054] **Fig.** 2A zeigt ein Diagramm, das ein Verfahren zur Erstellung eines Datenframes oder einer Datennachricht anzeigt, wenn der Datenframe oder die Datennachricht durch die verschiedenen Layer des OSI-Kommunikations-Stacks eines Geräts nach unten verarbeitet werden. Insbesondere erzeugt eine Anwendung (im Anwendungslayer) eine ursprüngliche Datennachricht **20** als eine Reihe von Bits und Bytes, die in jeder gewünschten Weise durch die Anwendung definiert sind, und liefert diese Nachricht an den Transportlayer. Im Allgemeinen kann das ”Liefern” einer Nachricht an einen Layer erreicht werden, indem eine Schnittstelle zur Anwendungsprogrammierung (API) oder ein ”Wrapper” für den Layer aufgerufen wird. Ein Durchschnittsfachmann kann auch eine Nachricht bereitstellen, indem für eine Schicht ”eine Wrapper-Funktion aufgerufen” wird. (Hier werden die Details der Session- und Darstellungslayer nicht im Detail besprochen, und es wird angenommen, dass sie im Transportlayer des Stacks ausgeführt werden.) Der Transportlayer des Stacks behandelt die Datennachricht von der Anwendung als Payload (wie in **Fig.** 2 dargestellt) und fügt Header-(und möglicherweise Trailer-)Informationen dazu, so wie sie durch das verwendete Transportlayerprotokoll definiert werden. Die Header-(und möglicherweise Trailer-)Informationen, die hier als Transportlayerinformation **22** bezeichnet werden, können eine Definition der Sessioninformationen (z. B. eine Session-ID), eine Definition oder die Angabe der Art des verwendeten Transportprotokolls usw. umfassen. Die entstehende Nachricht **23** wird dem Netzwerklayer des Stacks **23** zur Verfügung gestellt, der die Nachricht vom Transportlayer als Payload behandelt, wie in **Fig.** 2 dargestellt. Im Netzwerklayer werden die Netzwerkrouting-Informationen oder Informationen für das Netzwerkrouting-Protokoll **24** in Form von zusätzlichen Header- und Trailer-Informationen an Payload **23** angehängt (in **Fig.** 2 als Header-Abschnitt dargestellt). Die Netzwerkroutinginformationen **24** können beispielsweise Identifikationen des Quellen- und Ziel-Netzwerks enthalten (wie beispielsweise eine Quellen-IP-Adresse und eine Ziel-IP-Adresse für die IP-Netzwerkroutingprotokolle, oder Graphen-IDs für das WirelessHART-Netzwerkroutingprotokoll), Timeout-Informationen (Time-to-Live-Informationen), Prüfsummen-Informationen, die Länge der Nachricht oder Payload-Informationen, verwendete Flags für verschiedene Zwecke usw.

[0055] Danach wird die Nachricht, so wie durch den Netzwerklayer angelegt, dem Datenlinklayer bereitgestellt, wo die durch den Netzwerklayer erzeugte Nachricht als Payload **25** behandelt wird und zusätzliche Datenlinkinformationen **26** in Form von Header- und/oder Trailerabschnitten hinzugefügt werden, so wie es durch das Datenlinkprotokoll definiert wird, das im Kommunikationsnetzwerk verwendet wird. Insbesondere der Datenlinklayer fügt in der Regel Datenverbindungsinformationen in Form von Quell- und Zielgerätidentifikationen (wie Quell- und Ziel-MAC-Adressen), Medien-Steuerinformationen, wie Start- und Stopbits, Flusssteuerungsbits, Längeninformationen, Prüfsummeninformationen, Fragmentierungsinformationen einschließlich Auftragsinformationen usw. hinzu. In einigen Fällen fügen Anwendungen, die mit dem physikalischen Layer des Stacks verbunden sind, dem Datenframe assoziierte physikalische Layerprotokollinformationen hinzu.

[0056] Die Nachricht wird, so wie sie vom Datenlinklayer erzeugt wurde, an den physischen Layer ausgegeben und über den physischen Layer an das Gerät gesendet, so wie es durch die Zieladresse in den Informationen von Datenlink-Header **26** definiert wird. Es versteht sich, dass, wenn sie in einem Gerät decodiert ist, die Nachricht den Stack des Empfangsgerätes hinaufläuft, wobei jeder Layer des Stacks die entsprechenden Header- und Trailer-Informationen entfernt, die der Layer benutzt hat, um die Nachricht zu decodieren und sie entsprechend zu verarbeiten. Somit entfernt und decodiert im Empfangsgerät der Datenlinklayer des Stacks die Datenlink-Header- und Trailerinformationen **26**, um zu bestimmen, ob die Nachricht korrekt empfangen und an dieses Gerät gerichtet wurde (beispielsweise wie durch die MAC-Adresse in den Datenlinkinformationen **26** festgelegt). Wenn dies der Fall ist, wird der Payload-Abschnitt **25** der Nachricht am Datenlinklayer dem die Netzwerklayer bereitgestellt, der die Header- und Trailerinformationen **24** des Netzwerklayer-Headers entfernt, um das Senden und Empfangen von Netzwerkadressen oder anderen Routinginformationen festzulegen. Der Netzwerklayer des Stacks kann diese Netzwerkroutinginformationen verwenden, um den logischen Punkt in dem Netzwerk zu bestimmen, der diese Nachricht empfangen soll, sowie die entsprechende Art des Routings

dieser Nachricht zum richtigen Punkt im Netzwerk. Unter der Annahme, dass die Netzwerkadresse oder der logische Punkt im Netzwerk dem Empfangsgerät oder einer Anwendung im Empfangsgerät zugeordnet ist, wird die verbleibende Payload **23** dann für den Transportlayer bereitgestellt, der die Transportlayerinformationen **22** entfernt, um das in der Nachricht zu verwendende Transportlayerprotokoll, die vom Transportlayer des Stacks verwendeten Datentypen usw. zu bestimmen, damit die Nachricht vor der Ausgabe der verbleibenden Nachricht **20** an die entsprechende Anwendung im Anwendungslayer verarbeitet werden kann.

[0057] Es versteht sich, dass Netzwerkroutinginformationen (oder Netzwerkprotokollinformationen) generell Informationen umfassen, die der logischen Adresse eines Computers oder einer Anwendung entsprechen, die geplanter Empfänger des Datenframes sind. Die Netzwerkroutinginformationen des Datenframes können auch Informationen enthalten, die der logischen Adresse des Computers entsprechen, die den Datenframe übertragen hat. Beispielsweise enthält der IP-Header des in das IP-Netzwerkprotokoll eingekapselten Datenframes eine Quellen-IP-Adresse und eine Ziel-IP-Adresse. Die Quellen-IP-Adresse entspricht der logischen Adresse des Computers oder der Anwendung, der bzw. die den Datenframe übertragen hat, und die Ziel-IP-Adresse entspricht der logischen Adresse des Computers oder der Anwendung, der bzw. die der vorgesehene Empfänger des Datenframes ist. (Computer können mehr als eine logische IP-Adresse haben.) Zwischengeschaltete Geräte wie beispielsweise Router verwenden die IP-Adressen, um den Datenframe an den beabsichtigten Empfänger des Datenframes zu ”routen” oder weiterzuleiten.

[0058] Das hierin beschriebene Kommunikationsverfahren ändert jedoch das oben aufgeführte Verfahren zur Erzeugung und Decodierung von Datenframes, um die Fähigkeit zu erhalten, Datennachrichten zu erstellen, die mit einem ersten Netzwerkroutingprotokoll oder einer entsprechenden Technik konform und durch sie decodierbar sind (beispielsweise eine Technik zum expliziten Ziel- oder Adressierungsrouting, eine Technik zum Nicht-Mesh-Netzwerkrouting oder eine Technik zum Routing auf Basis einer Netzwerktopologie-Adressierung), die aber über ein Kommunikationsnetzwerk gesendet wird, das ein zweites und unterschiedliches Netzwerkroutingprotokoll oder eine entsprechende Technik verwendet (wie eine Technik zum impliziten Ziel- oder Adressierungsrouting, eine Technik zum Mesh-Netzwerkrouting oder eine Technik für ein nicht auf der Adresse der Netzwerktopologie basierendes Routing). Im Allgemeinen erzeugt das neue Kommunikationsverfahren eine Nachricht mit Netzwerkroutinginformationen oder Netzwerkprotokollinformationen, verbunden mit mehreren unterschiedlichen Netzwerkroutingprotokollen, die in der Nachricht codiert werden, wenn die Nachricht über das Kommunikationsnetzwerk gesendet wird. Dieses Merkmal erlaubt es, eine Nachricht so zu erstellen, dass sie unter Verwendung eines ersten Netzwerkroutingprotokolls gesendet wird, erlaubt es dann aber, dass die Nachricht tatsächlich über ein Kommunikationsnetzwerk gesendet wird, welches das zweite Netzwerkroutingprotokoll verwendet oder ihm entspricht. Außerdem kann die Nachricht einfach an Gateway-Geräten konvertiert werden, die an der Verbindung zwischen zwei Kommunikationsnetzwerken angebracht sind, indem verschiedene Netzwerkroutingprotokolle verwendet werden, ohne die Nachricht im Anwendungslayer decodieren zu müssen. Stattdessen muss das Gateway-Gerät nur die zweiten Informationen des Netzwerkrouting-Protokolls entfernen, wenn eine Nachricht vom zweiten Netzwerk an das erste Netzwerk geliefert wird, oder um die zweiten Informationen des Netzwerkrouting-Protokolls hinzuzufügen, wenn die Nachricht in das zweite Netzwerk gesendet wird, um von einem Gerät im zweiten Netzwerk empfangen zu werden.

[0059] **Fig.** 2B und **Fig.** 3 werden verwendet, um ein Verfahren zum Erzeugen eines IP-Datenframes zu beschreiben, beispielsweise ein IP-fähiges WirelessHART-Gerät, das über ein WirelessHART-Kommunikationsnetzwerk geroutet und von einem anderen IP-fähigen WirelessHART-Gerät, z. B. einem IP-fähigen WirelessHART-Gateway im WirelessHART-Kommunikationsnetzwerk, als IP-Datenframe decodiert werden kann. Insbesondere zeigt **Fig.** 2B einen Datenframe, der gemäß dem hierin beschrieben Kommunikationsverfahren erzeugt wurde, während **Fig.** 3 ein Flussdiagramm **40** zeigt, das die Schritte der Kommunikationstechnik veranschaulicht, die verwendet wird, um den Datenframe aus **Fig.** 2B zu erzeugen.

[0060] Insbesondere erzeugt an einem Block **41** des Flussdiagramms **40** aus **Fig.** 3 eine Anwendung im Anwendungslayer des WirelessHART-Geräts eine Datennachricht **30** (**Fig.** 2B) unter der Annahme, dass die Nachricht **30** unter Verwendung von Standard-IP-Routing, so wie es von dem IPv4- oder IPv6-Protokoll bereitgestellt wird, codiert und gesendet wird. Falls gewünscht, kann diese oder eine andere Anwendung in dem Anwendungslayer des WirelessHART-Gerätes zuerst eine Nachricht an ein Gateway-Gerät senden (über das WirelessHART-Netzwerk unter Verwendung eines WirelessHART-Netzwerkroutingprotokolls) und fordern, dass das Gateway-Gerät daraufhin DNS(Domain Name System)-Dienste ausführt oder implementiert, und um für das WirelessHART-Gerät eine IP-Adresse (z. B. eine IPv6- oder IPv4-Adresse) für eine bestimmte Netzwerkeinheit bereitzustellen, beispielsweise für ein Gerät, eine Webseite, einen Server, einen Router, einen E-Mail-Empfänger usw. In diesem Fall kann das WirelessHART-Gerät zunächst einen Domain-Namen für die Netzwerkeinheit bereitstellen (z. B. eine Einheit im zweiten oder IP-Netzwerk), z. B. einen Website-Hostnamen

oder einen E-Mail-Server-Hostnamen oder einen anderen von DNS-Diensten verwendeten Namen. Nach Erhalt der IP-Adresse für die Netzwerkeinheit über die DNS-Dienste (die im Gateway oder in einem anderen Gerät im IP-Netzwerk implementiert werden können), gibt das Gateway-Gerät dann die IP-Adresse für die Netzwerkeinheit über das WirelessHART-Netzwerk zurück an die Anwendung innerhalb des WirelessHART-Geräts, wozu ein WirelessHART-Netzwerkroutingprotokolls verwendet wird; daraufhin kann die Anwendung dann diese IP-Adresse für die Kommunikation unter Nutzung der IPv6- oder IPv4-Datenpakete verwenden. Die Anwendung kann eine socketbasierte Anwendung sein, die eine IP-basierte Socketadressierung verwendet, um mit anderen IP-fähigen Geräten zu kommunizieren.

[0061] In jedem Fall wird an einem Block **42** die Anwendungsnachricht in dem Transportlayer des Stacks des IP-fähigen WirelessHART-Geräts verarbeitet, das der Datennachricht **30** typische IP-basierte Transportlayerinformationen hinzufügt, wie z. B. TCP-Header- und Trailer-Informationen (diese Nachricht wird als Payload-Abschnitt des Datenframes behandelt, der im Transportlayer des Stacks erzeugt wird).

[0062] Als nächstes fügt ein Block **43**, der in einem Netzwerklayer des Stacks des IP-fähigen Geräts durchgeführt werden kann, dem Datenframe IP-konforme Netzwerkroutinginformationen hinzu, beispielsweise IP-Quell- und Ziel-Adressinformationen, Informationen zu Datenpaketlängen, usw. Diese IP-Netzwerkroutinginformationen sind in **Fig.** 2B als Headerinformationen **34a** dargestellt und werden so dargestellt, dass sie mit einem ersten Netzwerkroutingprotokoll verbunden mit oder dadurch festgelegt werden (in diesem Fall ein IP-Netzwerkroutingprotokoll). Wie oben angemerkt, kann die IP-Zieladresse über die Kommunikation mit einem Gateway-Gerät empfangen werden, das DNS-Dienste durchführt oder aufruft, um eine IP-Zieladresse zu bestimmen. Außerdem kann, falls gewünscht, ein Block **44** aus **Fig.** 3 den durch den Block **43** erzeugten Datenframe fragmentieren, damit der durch Block **43** erzeugte IP-Datenframe über ein Kommunikationsnetzwerk zu senden, indem ein Netzwerkroutingprotokoll, ein Datenlinklayerprotokoll oder ein Protokoll des physischen Layers, das keine Datenframes in der Größe oder Länge unterstützt, die vom IP-Netzwerkrouting unterstützt werden. Eine solche Fragmentierung ist in Low-Power-IP-Netzwerken üblich, die in der Regel die Größe oder Länge der gesendeten Datenpakete auf eine Länge beschränken, die kleiner ist als die Größe, die vom traditionellen IP-Netzwerkrouting unterstützt wird. Eine gängige, in Low-Power-IP-Netzwerken verwendete Fragmentierungstechnik ist das 6LoWPAN-Protokoll, das jedes IPv6-Datenpaket oberhalb einer bestimmten Größe in mehrere Pakete oder fragmentierte Datenframes aufteilt, die eine Länge oder Größe haben, die von den Datenlink- und physischen Layerprotokollen des Netzwerks unterstützt wird. Auch wenn die 6LoWPAN-Fragmentierungstechnik hier so beschrieben wird, dass sie zur Fragmentierung und Defragmentierung verwendet wird, könnten auch andere Arten von Fragmentierungstechniken verwendet werden. Außerdem ist in einigen Situationen, z. B. wenn ein zweites Netzwerkroutingprotokoll oder physisches Netzwerk Datenframes der Länge oder Größe des ersten Netzwerkprotokolls unterstützt, möglicherweise keine Fragmentierung erforderlich. In diesen Fällen kann Block **44** übersprungen werden.

[0063] In jedem Fall, unter der Annahme, dass eine Fragmentierung durchgeführt wird, behandelt ein Block **46** (der in einem Netzwerklayer des Stacks durchgeführt werden kann, der mit der Durchführung des Netzwerkroutings unter Verwendung des zweiten Netzwerkroutingprotokolls verbunden ist) das Datenpaket vom Block **44** als Payload und fügt jedem Datenframe oder fragmentiertem Datenframe Netzwerkroutinginformationen hinzu, so wie sie durch das zweite Netzwerkroutingprotokoll festgelegt sind. **Fig.** 2B zeigt ein Datenpaket in diesem Zustand gezeigt, das zweite Netzwerkroutinginformationen **34b** umfasst, die als Header-Informationen dem durch Block **43** (oder Block **44**) erzeugten Paket **35** hinzugefügt wurden. Diese zweiten Netzwerkroutinginformationen **34b** können zum Beispiel eine Graph-ID für das Graphen-Routing innerhalb eines WirelessHART-Netzwerks umfassen, sowie andere Netzwerkroutinginformationen für das Datenpaket, so wie sie beispielsweise durch den WirelessHART-Netzwerkroutinglayer des WirelessHART-Protokolls definiert werden. Natürlich können andere Netzwerkroutingprotokolle, wie andere Mesh-Netzwerke, implizite Ziel- oder Adressierungsprotokolle, nicht auf der Adresse der Netzwerktopologie basierende oder Nicht-IP-Netzwerkroutingprotokolle als das zweite Netzwerkroutingprotokoll verwendet werden. Außerdem, als Teil der Netzwerkroutinginformationen, die dem Datenpaket auf dieser Ebene hinzugefügt wurden, fügt Block **46** ein Bit-Flag hinzu oder setzt es (z. B. als ein Ein-Bit-Flag wie in **Fig.** 2B als Bit 39 gezeigt) in einem ansonsten unbenutzten Abschnitt oder in ein Bit der Netzwerkroutinginformationen **34b**, um anzuzeigen, dass das zugrunde liegende Datenpaket oder Datenframe **35** eigentlich ein Datenframe ist, das bzw. der mit dem ersten Netzwerkroutingprotokoll und nicht mit dem zweiten Netzwerkroutingprotokoll verbunden ist. Insbesondere gibt es normalerweise in den Netzwerkroutinginformationen nicht-verwendete Bits oder Bytes, die von einem bestimmten Netzwerkroutingprotokoll jedem Datenpaket hinzugefügt werden. Das Gerät, welches das Netzwerkroutingprotokoll decodiert, ignoriert diese Bits in der Regel während des Decodierungsprozesses. In diesem Fall können jedoch eines oder mehrere dieser sonst unbenutzten Bits oder Bytes als ein Flag verwendet werden, um anzuzeigen, ob das zugrundeliegende Datenpaket eines ist, das ursprünglich mit dem ersten Netzwerkroutingprotokoll oder

dem zweiten Netzwerkroutingprotokoll verbunden ist. Zum Beispiel kann das Setzen eines bestimmten Bits in den Netzwerkroutinginformationen **34b** für das zweite Netzwerkroutingprotokoll auf eine ”1” anzeigen, dass das zugrundeliegende Datenpaket **35** ein IPv6-Datenpaket ist, während das Setzen dieses bestimmten Bits in den Netzwerkroutinginformationen für das zweite Netzwerkroutingprotokoll auf eine ”0” anzeigen kann, dass die zugrundeliegenden Daten ursprünglich unter Verwendung des zweiten Netzwerkroutingprotokolls erstellte oder eingekapselte Daten sind (z. B. ein Datenframe, der ursprünglich als WirelessHART-Datenframe erzeugt wurde). Natürlich können solche Flags oder Angaben von jeder gewünschten Länge sein, beispielsweise ein einzelnes Bit, mehrere Bits, ein Byte, mehrere Bytes, usw.

[0064] Als nächstes kann in einem Block **47** der **Fig.** 3 ein Datenlinklayer eines Stacks Datenverbindungsinformationen **36** (in Form von Header- und/oder Trailerinformationen) zu dem empfangenen Paket **37** hinzufügen, das von Block **46** erzeugt wurde, wobei die Datenlinklayer-Informationen **36** durch das jeweilige Datenlinkprotokoll festgelegt wird, das im zweiten Kommunikationsnetzwerk verwendet wird. Ein Block **48** kann dann physische Layerinformationen **38** zu dem von Block **47** empfangenen Paket hinzufügen, so wie es durch den physischen Layer des zweiten Kommunikationsnetzwerks definiert wird. Ein Block **49** kann dann das Datenpaket senden oder übermitteln (mit Netzwerkroutinginformationen für ein erstes und ein zweites Netzwerkroutingprotokoll darin).

[0065] Generell werden an einem Empfangsgerät im Kommunikationsnetzwerk die verschiedenen physischen und Datenlinklayerinformationen der Pakete unter Verwendung des zweiten Netzwerkroutingprotokolls (d. h. das zweite Kommunikationsnetzwerk) entfernt und decodiert und in typischer Weise verwendet. Darüber hinaus verfügt der Stack des Empfangsgeräts über einen Netzwerklayer, der die Netzwerkroutinginformationen entfernt und für das zweite Netzwerkroutingprotokoll decodiert. Während dieses Prozesses decodiert der Netzwerklayer des Stacks des Empfangsgeräts das besondere Bit oder Byte dieser zweiten Netzwerkrouting-Informationen, die als ein Flag oder ein anderes Verfahren verwendet wird, um anzuzeigen, ob das zugrunde liegende Datenpaket ein erstes oder zweites Netzwerkroutingprotokoll-Paket ist, und verwendet diese Informationen zur weiteren Verarbeitung. Insbesondere, wenn das Flag oder die Anzeige nach der Decodierung angibt, dass das zugrundeliegende Paket ein typisches oder normales Paket ist, welches das zweite Netzwerkroutingprotokoll nutzt, wird das Paket an einen Transportlayer des Stacks übergeben, der gemäß dem zweiten Kommunikationsnetzwerkprotokoll arbeitet. Wenn andererseits das Flag oder die andere Anzeigeart nach der Decodierung angibt, dass das zugrundeliegende Paket beispielsweise ein IPv6-Paket ist (das heißt, dem ersten Netzwerkroutingprotokoll zugeordnet ist), wird das Paket dann einem Netzwerklayer des Stacks übergeben, der Decodierung und Routing von Paketen gemäß dem ersten Netzwerkroutingprotokoll vornimmt. Wenn das Paket ein IPv6-Paket ist, kann beispielsweise der Netzwerk-Stack die Pakete defragmentieren, indem er eine Reihe von empfangenen Paketen zusammenfügt, um das ursprüngliche IPv6-Paket zu bilden, und dann das defragmentierte IPv6-Paket für einen Transportlayer des Stacks bereitstellen, wobei der Stack gemäß dem ersten Netzwerkkommunikationsprotokoll konfiguriert ist, um die Decodierung und Verarbeitung des Transportlayers in der üblichen Art durchzuführen. Auf diese Weise wird das IPv6-Paket für die richtige Anwendung in dem Gerät zur Verarbeitung bereitgestellt, oder sie kann unter Verwendung von IP-Netzwerk-Routing über ein anderes Netzwerk weitergeleitet werden.

[0066] Es versteht sich jedoch, dass die unter Verwendung eines ersten Netzwerkkommunikationsprotokolls erstellten Datenpakete (d. h. ein Protokoll, das auf einem ersten Netzwerkroutingprotokoll beruht) über das zweite Kommunikationsnetzwerk gesendet werden (das ein zweites Netzwerkroutingprotokoll für Routing-Kommunikationen verwendet), und sie enthalten Netzwerkroutinginformationen für das erste Netzwerkroutingprotokoll und das zweite Netzwerkroutingprotokoll oder wie durch diese definiert. Diese Kommunikationstechnik ermöglicht die Erzeugung von Datenpaketen in Übereinstimmung mit oder unter der Annahme von, dass sie über ein Netzwerk unter Verwendung eines ersten Netzwerkkommunikationsprotokolls (das auf einem ersten Netzwerkroutingprotokoll wie beispielsweise einem IP-Netzwerkroutingprotokoll basiert) in einem beliebigen Gerät übertragen werden, das gilt auch für Geräte, die mit einem Kommunikationsnetzwerk verbunden sind, welches das erste Netzwerkroutingprotokoll für Netzwerkkommunikationen nicht verwendet. Durch diese Kommunikationstechnik können auch Geräte in einem zweiten Kommunikationsnetzwerk, das nicht das erste Netzwerkroutingprotokoll verwendet, gemäß dem ersten Netzwerkroutingprotokoll über das zweite Kommunikationsnetzwerk erzeugte Datenpakete senden und empfangen, selbst wenn das zweite Kommunikationsnetz das Routing unter Verwendung des ersten Netzwerkroutingprotokolls nicht nutzt oder unterstützt. Durch dieses Merkmal können Geräte im zweiten Kommunikationsnetzwerk unter Verwendung von Datenpaketen miteinander kommunizieren, die dem ersten Netzwerkroutingprotokoll entsprechen, und es können zusätzlich Geräte im zweiten Kommunikationsnetzwerk direkt mit Geräten in einem anderen Kommunikationsnetz kommunizieren (z. B. in einem Netzwerk, das Routing mit einem ersten Netzwerkroutingprotokoll nutzt), indem Datenpaketen verwendet werden, die gemäß dem ersten Netzwerkroutingprotokoll konfiguriert sind.

[0067] Eine Reihe von Beispielen von Kommunikationsarten, die unter Verwendung dieser Kommunikationstechnik auftreten können, werden nun detaillierter besprochen. In einigen der unten angegebenen Beispiele erzeugt ein erster Computer in einem ersten Kommunikationsnetzwerk Datenframes nach einem Netzkommunikationsprotokoll, indem ein erstes Netzwerkroutingprotokoll verwendet wird, z. B. TCP/IP, und diese Datenframes werden an einen zweiten Computer in einem zweiten Kommunikationsnetzwerk übertragen, das ein zweites Netzwerkroutingprotokoll nutzt, beispielsweise das WirelessHART-Protokoll. (Wie oben erwähnt, können jedoch auch andere Typen oder spezielle Beispiele von Netzwerkroutingprotokollen in den ersten und zweiten Kommunikationsnetzwerken verwendet werden.) In den folgenden Beispielen kann der erste Computer die Datenframes über die Kommunikationskanäle senden, die einem Datenlinklayer-Kommunikationsprotokoll entsprechen, z. B. dem Standard 802.3, und der zweite Computer kann die Datenframes über die Kommunikationskanäle empfangen, die demselben oder einem anderen Datenlinkprotokoll entsprechen, z. B. dem Standard 802.15.4. So wie es hier verwendet wird, wird ein Datenlinkprotokoll auch allgemein als physisches Layerprotokoll bezeichnet.

[0068] Wie oben erwähnt, legen Datenlinkprotokolle oder Standards unter anderem die maximale Größe eines Datenframes fest, der während einer Kommunikationstransaktion über Kommunikationskanäle, die dem bestimmten Kommunikationsprotokoll entsprechen, übertragen oder empfangen werden kann. Die maximale Größe eines Datenframes wird generell als Maximum Transmission Unit (MTU) Größe bezeichnet. Beispielsweise ist die MTU-Größe für einen über einen Kommunikationskanal gemäß dem Standard 802.15.4 übertragenen Datenframe typischerweise 102 Byte. Die MTU-Größe für einen über einen Kommunikationskanal gemäß dem Standard 802.11 übertragenen Datenframe ist typischerweise 2272 Byte. Die MTU-Größe für einen über einen Kommunikationskanal gemäß dem Standard 802.3 übertragenen Datenframe ist typischerweise 1500 Byte.

[0069] Getrennt davon kann ein Standard für ein Netzwerkroutingprotokoll auch eine MTU-Größe für gepackte Datenframes festlegen, die dem Netzwerkroutingprotokoll entsprechen. Beispielsweise legt der IPv6-Standard den MTU-Wert eines Datenframes typischerweise bei 1280 Byte fest. Somit ist es in einigen Szenarien eventuell nicht möglich, in einer einzelnen Transaktion einen IPv6-Datenframe über Kommunikationskanäle zu übertragen, die dem Standard 802.15.4 entsprechen. In diesen Szenarien kann ein IPv6-Datenframe in mehrere Datenframes fragmentiert werden, bevor der IPv6-Datenframe über Kommunikationskanäle gesendet wird, die Standard 802.15.4 entsprechen. Die Datenframe-Fragmente können mit Informationen versehen sein, die für den Zusammenbau der Datenframe-Fragmente in einem Empfangsgerät vorgesehen sind, und sie können in dem Computer, der als Empfänger der IPv6-Datenframes vorgesehen ist, zusammengesetzt werden. Fragmentierung und Zusammensetzung können durch Nutzung eines Protokolls wie beispielsweise 6LoWPAN erfolgen. Als Folge der Fragmentierung enthalten möglicherweise einige oder alle IPv6-Datenfragmente keinen IP-Header.

[0070] In nachstehend beschriebenen Beispielen können die Fragmente eines IPv6-Datenframes mit Headerinformationen versehen sein, die einem zweiten Netzwerkroutingprotokoll entsprechen, beispielsweise dem WirelessHART-Netzwerkroutingprotokoll, um das Routing von IPv6-Datenframes zwischen Computern zu erleichtern, die so konfiguriert sind, dass sie Datenframes entsprechend dem WirelessHART-Protokoll verarbeiten. In einigen Beispielen kann, wie oben erwähnt, der Header des WirelessHART-Netzwerkprotokolls ein Flag umfassen, der anzeigt, dass der Datenframe ein IPv6-Datenframe ist. Hier kann der Computer so konfiguriert sein, dass IPv6-Datenframefragmente, die in WirelessHART-Netzwerkprotokollinformationen eingekapselt sind, und Nicht-IPv6-Datenframes, die in WirelessHART-Netzwerkprotokollinformationen eingekapselt sind, empfangen und gesendet werden.

[0071] Im Allgemeinen bedeutet die hier beschriebene Kommunikationstechnik in einem Fall das Erstellen eines Datenframes eines ersten Typs, d. h. unter Verwendung eines ersten Netzwerkprotokolls wie beispielsweise eines IP-Protokolls, und das Senden dieses Datenframes über ein Netzwerk unter Verwendung des Netzwerk-Routings nach dem ersten Netzwerkprotokoll an ein zweites Gerät, das mit einem zweiten Kommunikationsnetz verbunden ist, das ein zweites und unterschiedliches Netzwerkrouting-Protokoll verwendet, beispielsweise ein Nicht-IP-Protokoll. Das zweite Gerät kapselt den Datenframe mit der ersten Netzwerkrouting-Informationen darin in einen Datenframe ein, der vom zweiten Netzwerkrouting-Protokoll festgelegte Netzwerkrouting-Informationen enthält, und sendet oder routet den Datenframe über das zweite Kommunikationsnetzwerk unter Verwendung des zweiten Netzwerkrouting-Protokolls an ein drittes Gerät, das dann die empfangenen Datenframes decodiert, um den Datenframe des ersten Typs (z. B. ein IP-Datenframe) zu erzeugen. Ein Flussdiagramm **200** aus **Fig.** 4A veranschaulicht dieses Verfahren im Detail und umfasst eine Reihe von Schritten in unterschiedlichen Geräten in den unterschiedlichen Kommunikationsnetzwerken, über die Kommunikation erfolgt.

[0072] Insbesondere werden die Blöcke **202**, **204** und **206** des Flussdiagramms **200** aus **Fig.** 4A in einem ersten, mit einem ersten Kommunikationsnetzwerk verbundenen Gerät ausgeführt, das ein erstes Netzwerkrouting-Protokoll nutzt, z. B. ein IP-Netzwerkrouting-Protokoll (z. B. IPv6). Die Blöcke **208**, **210** und **212** werden in einem zweiten Gerät ausgeführt, beispielsweise einem Gateway-Gerät, das sich an einem Übergang zwischen dem ersten Kommunikationsnetzwerk und einem zweiten Kommunikationsnetzwerk befindet, das ein zweites Netzwerkrouting-Protokoll verwendet, z. B. ein WirelessHART-Kommunikationsnetzwerk, das sich vom ersten Netzwerkrouting-Protokoll unterscheidet. (Hier ist zu bemerken, dass das WirelessHART-Kommunikationsnetzwerk ein Mesh-Netzwerk ist, das typischerweise in der Prozesssteuerungsindustrie verwendet wird, um die Kommunikation in verfahrenstechnischen Anlagen und andere Arten von Fertigungsanlagen zu realisieren. Generell erfordert das WirelessHART-Netzwerkprotokoll, dass verschiedene andere Arten von Informationen, wie die Prozesssteuerung oder Prozesssystembefehle, in die Netzwerkrouting-Informationen eines Datenpakets platziert werden, um ein besseres Routing innerhalb des Prozesssteuerungsnetzwerks zu ermöglichen. Solche Befehle werden nicht von IP-Netzwerkrouting-Protokollen unterstützt.) Die Blöcke **214**, **216**, **218** und **220** werden in einem Empfangsgerät ausgeführt, das mit dem zweiten Kommunikationsnetzwerk verbunden ist und welches das Gerät ist, mit dem das ursprüngliche Datenpaket vom ersten Gerät im ersten Kommunikationsnetzwerk empfangen wurde.

[0073] Insbesondere erzeugt im Block **202** eine Anwendung innerhalb des ersten Gerätes im ersten Kommunikationsnetzwerk (z. B. ein IP-basiertes Kommunikationsnetzwerk) ein Anwendungsdatenpaket oder eine Nachricht in einer Weise, die mit dem Routing dieser Nachricht über ein Kommunikationsnetz unter Verwendung eines ersten Netzwerkrouting-Protokolls übereinstimmt (für dieses Beispiel wird angenommen, dass es ein IP-Protokoll ist, z. B. ein IPv6-Protokoll). Der Block **204** funktioniert so, dass er die Anwendungsnachricht (so wie sie durch einen Darstellungslayer, einen Sessionlayer und einen Transportlayer eines Stacks verarbeitet wird, wenn dies gewünscht wird) in die ersten Netzwerkrouting-Informationen einkapselt, die dem ersten Netzwerkrouting-Protokoll entsprechen. In diesem Beispiel kann die Ausgabe des Blocks **204** ein Standard-IPv6-Datenpaket sein, bereit zur Übertragung über ein IP-Netzwerk, z. B. das Internet. Als nächstes bewirkt ein Block **206**, dass das eingekapselte Anwendungsdatenpaket über das erste Kommunikationsnetzwerk oder den ersten Kommunikationskanal übertragen wird, das bzw. der das erste Netzwerkrouting-Protokoll verwendet, beispielsweise unter Verwendung von IP-Adress-Routing. Natürlich kann der Block **206** das Datenpaket am Datenlinklayer und dem physischen Layer des Kommunikations-Stacks verarbeiten, bevor das IPv6-Datenpaket über das erste Kommunikationsnetzwerk gesendet wird.

[0074] Als nächstes empfängt und decodiert ein Block **208** beispielsweise in einem Gateway-Gerät, welches das erste Kommunikationsnetzwerk mit dem zweiten Kommunikationsnetzwerk verbindet, den Datenframe und erkennt, dass die IP-Adresse innerhalb der ersten Netzwerkrouting-Informationen des Datenframes für ein Gerät im zweiten Kommunikationsnetzwerk bestimmt oder diesem zugeordnet ist. Der Block **208** in dem Gateway-Gerät erzeugt dann einen Datenframe gemäß dem zweiten Kommunikationsnetzwerkprotokoll, das vom zweiten Kommunikationsnetzwerk genutzt wird, indem das empfangene Datenpaket (einschließlich der Netzwerkrouting-Informationen für das erste Netzwerkrouting-Protokoll) in zweite Netzwerkrouting-Informationen eingeschlossen wird, die dem zweiten Netzwerkrouting-Protokoll entsprechen und so konfiguriert sind, dass das Datenpaket an das Gerät geleitet wird, das mit der IP-Adresse innerhalb des ursprünglichen IPv6-Datenpakets im zweiten Kommunikationsnetzwerk verbunden ist. Wenn erforderlich oder gewünscht, kann der Block **208** so funktionieren, dass er das empfangene IPv6-Datenpaket fragmentiert, beispielsweise unter Verwendung von 6LoWPAN-Fragmentierungstechniken, um eine Anzahl von Datenpaketen aus dem empfangenen IPv6-Datenpaket zu erzeugen, bevor die zweiten Netzwerkrouting-Informationen auf jedem dieser erstellen Datenpakete (oder Datenpaket-Fragmente) platziert werden.

[0075] Ein Block **210** erzeugt daneben ein Indiz, beispielsweise durch Setzen eines Flags in den zweiten Netzwerkrouting-Informationen der Datenpakete, das anzeigt, dass die zugrundeliegenden Datenframes innerhalb der erzeugten Pakete Anwendungsdaten besitzen, die mit dem ersten Netzwerkrouting-Protokoll verbunden sind. Ein Block **212** in dem Gateway-Gerät bewirkt dann, dass der Datenframe über das zweite Kommunikationsnetzwerk oder einen Kanal auf dem zweiten Kommunikationsnetzwerk übertragen wird, das gemäß dem zweiten Netzwerkrouting-Protokoll unter Verwendung der zweiten Informationen des Netzwerkrouting-Protokolls in den Datenpaketen arbeitet.

[0076] Ein Block **214** im Empfangsgerät, d. h. das Gerät, das mit der IP-Adresse des ursprünglichen Datenframes verbunden, aber im zweiten Kommunikationsnetzwerk angeordnet ist, empfängt und decodiert die Datenframes in den physischen, Datenlink- und Netzwerklayern des Stacks auf den zweiten Netzwerkrouting-Informationen des Datenframes. Ein Block **216** in diesem Gerät funktioniert so, dass er das Vorhandensein oder die Abwesenheit von dem Indiz bzw. Flag in den zweiten Netzwerkrouting-Informationen des empfangenen

Datenpakets entdeckt. Wenn ein derartiges Flag nicht existiert oder wenn das Flag anzeigt, dass das zugrunde liegende Datenpaket ursprünglich als ein Datenpaket erstellt wurde, das mit dem zweiten Netzwerkrouting-Protokoll verbunden ist, stellt der Block **218** das Datenpaket einem Transportlayer des Stacks des Empfangsgeräts bereit, das für die Verarbeitung mit dem zweiten Netzwerkkommunikationsprotokoll verbunden ist. Wenn das Indiz jedoch existiert oder wenn das Flag anzeigt, dass das zugrunde liegende Datenpaket ursprünglich als ein Datenpaket erstellt wurde, das mit dem ersten Netzwerkrouting-Protokoll verbunden ist (z. B. ein IP-Protokoll), stellt der Block **220** die Daten einem Netzwerklayer des Stacks des Empfangsgeräts bereit, das die ersten Netzwerkrouting-Informationen verwendet, um das Paket zu decodieren und zu verarbeiten. Dieser Netzwerklayer des Stacks kann zunächst eine Reihe von solchen empfangenen Paketen defragmentieren, um das ursprüngliche IPv6-Datenpaket wieder zusammenzusetzen, und er kann dann das IPv6-Datenpaket unter Verwendung der Standardverarbeitung von IP-Netzwerklayer und Transportlayer verarbeiten, so dass das IPv6-Datenpaket so an eine Anwendung oder einen Anwendungslayer im Stack des Empfangsgeräts geliefert werden kann, als ob es mithilfe eines IP-Netzwerkrouting-Protokolls an das Empfangsgerät gesendet wurde.

[0077] Ähnlich kann die hier beschriebene Kommunikationstechnik das Erstellen eines Datenframes eines ersten Typs bedeuten, d. h. unter Verwendung eines ersten Netzwerkprotokolls, beispielsweise eines IP-Protokolls, und das Senden dieses Datenframes zunächst über ein zweites Netzwerk unter Verwendung eines zweiten Netzwerkprotokolls (das sich vom ersten Netzwerkrouting-Protokoll unterscheidet) an ein zweites Gerät, das ebenfalls mit einem ersten Kommunikationsnetzwerk verbunden ist. In diesem Fall kapselt das erste Gerät den Datenframe mit den ersten Informationen des Netzwerkrouting-Protokolls darin in zweite Netzwerkrouting-Informationen ein und sendet oder routet den Datenframe über das zweite Kommunikationsnetzwerk unter Verwendung des zweiten Netzwerkrouting-Protokolls an das zweite Gerät. Das zweite Gerät decodiert dann die empfangenen Datenframes unter Verwendung des zweiten Netzwerkrouting-Protokolls, um einen Datenframe des ersten Typs (z. B. einen IP-Datenframe) zu erzeugen, und sendet die decodierten Datenframes unter Verwendung des ersten Netzwerkrouting-Protokolls über das erste Kommunikationsnetzwerk. Ein Flussdiagramm **250** aus **Fig.** 4B veranschaulicht dieses Verfahren im Detail und umfasst eine Reihe von Schritten in unterschiedlichen Geräten in den unterschiedlichen Kommunikationsnetzwerken, über die Kommunikation erfolgt.

[0078] Insbesondere werden die Blöcke **252**, **254**, **256**, **258** und **260** des Flussdiagramms **250** aus **Fig.** 4B in einem ersten, mit einem zweiten Kommunikationsnetzwerk verbundenen Gerät ausgeführt, das ein zweites Netzwerkrouting-Protokoll nutzt, wie z. B. ein WirelessHART-Netzwerkrouting-Protokoll. Die Blöcke **262**, **264** und **268** werden in einem zweiten Gerät ausgeführt, beispielsweise einem Gateway-Gerät, das sich an einem Übergang zwischen dem ersten Kommunikationsnetzwerk (z. B. einem IP-basierten Kommunikationsnetzwerk) und dem zweiten Kommunikationsnetzwerk befindet, das ein zweites Netzwerkrouting-Protokoll verwendet. Der Block **270** kann in einem Empfangsgerät ausgeführt werden, das mit dem ersten Kommunikationsnetzwerk verbunden ist und das Gerät ist, mit dem das ursprüngliche Datenpaket empfangen werden soll, das vom ersten Gerät im zweiten Kommunikationsnetzwerk entwickelt wurde.

[0079] Insbesondere wird Block **252** innerhalb eines IP-fähigen Gerätes im zweiten Kommunikationsnetzwerk ausgeführt (z. B. ein IP-fähiges WirelessHART-Gerät) und erzeugt ein Anwendungsdatenpaket unter der Annahme, dass das Datenpaket über ein Kommunikationsnetzwerk gesendet wird, das dem ersten Kommunikationsprotokoll entspricht (z. B. ein Datenpaket gemäß einem IP-Netzwerkprotokoll). Die Anwendung, die dieses Datenpaket erzeugt, kann daher eine socketbasierte Anwendung sein. Wie oben in Bezug auf **Fig.** 4A aufgeführt, kann die in diesem Datenpaket verwendete IP-Adresse die erste sein, die durch die Kommunikation mit einem Gateway-Gerät erhalten wird, beispielsweise in einem WirelessHART-Kommunikationsnetzwerk, das mit einem IP-Netz verbunden ist, wobei das Gateway-Gerät DNS-Dienste ausführt (z. B. als DNS-Server agiert), um die richtige IP-Adresse für einen Domain-Namen zu bestimmen, der durch das WirelessHART-Gerät bereitgestellt wird. Der Block **254** des IP-fähigen Gerätes kapselt die Anwendungsnachricht (nach der Verarbeitung durch die entsprechenden Darstellungs-, Session- und Transportlayer eines Stacks innerhalb des IP-fähigen Gerätes) in die ersten Netzwerkrouting-Informationen ein, die dem ersten Netzwerkrouting-Protokoll entsprechen, beispielsweise durch Hinzufügen von IP-Adressinformationen zu der Datennachricht. Als nächstes erzeugt ein Block **256** einen Datenframe entsprechend einem zweiten Netzwerkrouting-Protokoll, indem ein Framing des Anwendungsdatenpakets von dem Block **254** mit den zweiten Netzwerkrouting-Informationen erfolgt, die dem zweiten Netzwerkrouting-Protokoll entsprechen. In diesem Fall kann Block **256** das durch den Block **254** erzeugte Datenpaket fragmentieren, um ein Set von Datenpaketen zu erzeugen, die der von dem zweiten Kommunikationsnetzwerk erlaubten Größe entsprechen, bevor die zweiten Netzwerkrouting-Informationen jedem der fragmentierten Datenpaketen hinzugefügt werden. Ein Block **258** erzeugt oder platziert in den zweiten Netzwerkrouting-Informationen eines solchen Datenpakets zusätzlich ein Indiz, das anzeigt, dass der zugrunde liegende Datenframe Anwendungsdaten besitzt, die mit dem ersten Netzwerkrouting-Protokoll verbunden sind (z. B. IPv6-Datenpakete). Ein Block **260** bewirkt dann, dass die von Block **256** und **258** erzeugten

Datenframes über einen Kanal des zweiten Kommunikationsnetzwerks übertragen werden, das gemäß dem zweiten Netzwerkrouting-Protokoll unter Verwendung der zweiten Informationen des Netzwerkrouting-Protokolls in diesen Paketen arbeitet.

[0080] Ein Block **262**, ausgeführt beispielsweise in einem Gateway-Gerät, das an verschiedenen Eingängen mit dem zweiten sowie dem ersten Kommunikationsnetzwerk verbunden ist, empfängt und decodiert die über das zweite Kommunikationsnetzwerk empfangenen Datenframes basierend auf den zweiten Netzwerkrouting-Informationen in diesen Datenframes. Ein Block **264** in diesem Gerät funktioniert so, dass er das Vorhandensein oder die Abwesenheit des Indizes bzw. Flags in den zweiten Netzwerkrouting-Informationen des empfangenen Datenpakets entdeckt. Wenn ein solches Flag in einem Datenpaket nicht existiert, kann ein Block **266** das Datenpaket dem Transportlayer des Empfangsgeräts bereitstellen, das gemäß dem zweiten Kommunikationsprotokoll arbeitet, um die Daten zu decodieren und zu verarbeiten, oder er kann weitere Routings dieser Daten über ein zweites Kommunikationsnetzwerk ausführen. Falls das Flag jedoch existiert oder gesetzt ist, ruft ein Block **268** den zugrunde liegenden IP-Datenframe ab, indem er zum Beispiel die IP-Datenframe-Informationen von verschiedenen, über das zweite Kommunikationsnetzwerk empfangenen Datenpaketen decodiert und zusammenfügt, um das ursprüngliche IPv6-Datenpaket zu erzeugen, einschließlich der IP-Netzwerkrouting-Informationen in dem ursprünglichen IPv6-Datenpaket. Der Block **268** kann dann die IP-Netzwerkrouting-Informationen (d. h. die IP-Adressinformation) innerhalb dieses zusammengefügten IPv6-Datenpaketes benutzen, um das Datenpaket über einen Kommunikationskanal des ersten Kommunikationsnetzwerks unter Verwendung des ersten Netzwerkrouting-Protokolls an einen Empfänger zu übermitteln. An einem Block **270** kann das Empfangsgerät (d. h. das Gerät mit der IP-Zieladresse des IPv6-Datenpakets wie von dem ersten Gerät erzeugt) das IPv6-Datenpaket auf eine beliebige bekannte oder standardmäßige Weise empfangen und decodieren.

[0081] Wie man versteht, können diese grundlegenden Kommunikationstechniken in verschiedenen Szenarien benutzt werden, um das Messaging oder die Kommunikation IP-Daten (z. B. Nachrichten von IP-Datenframes wie IPv6-Datenframes) in oder über Kommunikationsnetzwerke, z. B. Kommunikationsnetzwerke von verfahrenstechnischen Anlagen, bereitzustellen, die kein IP-basiertes Routing für die Kommunikation darin benutzen oder unterstützen. Einige dieser Szenarien werden im Folgenden exemplarisch beschreiben.

[0082] **Fig.** 5 zeigt ein Beispiel eines Kommunikationsnetzwerks **50**, in dem Feldgeräte in einer verfahrenstechnischen Anlage die hier beschriebenen Kommunikationsverfahren implementieren, um IPv6-Datenframes über das Internet **51** und über eins oder mehrere prozesssteuerungsbasierte Kommunikationsnetzwerke **53** und **71** zu empfangen und zu senden, die WirelessHART-Netzwerke sein können, die kein IP-Netzwerkrouting unterstützen. Im Allgemeinen umfassen Prozesssteuerungssysteme, wie sie in Chemie-, Erdöl- oder anderen verfahrenstechnischen Anlagen verwendet werden, Feldgeräte, wie in **Fig.** 5 als Feldgeräte **59**–**65** und **76**–**81** dargestellt, die Sensor- und physische Steuerungsfunktionen innerhalb eines Prozesses durchführen. Jedes der Feldgeräte **59**–**65** und **76**–**81** kann zum Beispiel ein Ventil, ein Ventilstellglied, ein Schalter, ein Sensor (z. B. Temperatur-, Druck- oder Durchflusssensor), eine Pumpe, ein Ventilator, ein Controller, ein E/A-Gerät usw. sein. Feldgeräte führen Steuer- und/oder physikalische Funktionen innerhalb eines Prozesses oder einer Prozesssteuerungsschleife durch, z. B. das Öffnen oder Schließen von Ventilen und die Messung von Prozessparametern. In den drahtlosen Kommunikationsnetzwerken **53** und **71** sind die Feldgeräte **59**–**65** und **76**–**81** Erzeuger und Verbraucher von Datenframes, z. B. von WirelessHART-Datenframes und IPv6-Datenframes. Geräte innerhalb der Netzwerke **53** und **71**, die in der Lage sind, IPv6-Datenframes zu erzeugen, zu empfangen und zu benutzen, werden hier als IP-fähige Geräte bezeichnet.

[0083] Im Allgemeinen nutzen Computer innerhalb der WirelessHART-Netzwerke **53** und **71** die Techniken des WirelessHART-Netzwerkroutings, um Datenframes zu Computern zu routen, die vorgesehene Empfänger der Datenframes sind. Die Feldgeräte können, wie es typisch ist, typische WirelessHART-Datenframes über die WirelessHART-Netzwerke **53** und **71** erzeugen und senden. Wie jedoch aus der oben stehenden Erklärung ersichtlich ist, können die Datenframes auch IPv6-Datenframe-Fragmente enthalten, die in Informationen von WirelessHART-Netzwerkprotokolle eingekapselt sind, oder sie können Fragmente enthalten, die nicht zu IPv6-Datenframes gehören und in Informationen von WirelessHART-Netzwerkprotokollen eingekapselt sind. In diesem Zusammenhang kann ein IPv6-fähiges WirelessHART-Gateway **54** zum Beispiel die Schritte der Fragmentierung von IPv6-Datenframes durchführen, die unter Verwendung geeigneter Protokolle vom Internet **51** empfangen wurden, zum Beispiel 6LoWPAN, und die Datenframe-Fragmente in Informationen von WirelessHART-Netzwerkprotokollen einkapseln, die es ermöglichen, die Datenframe-Fragmente innerhalb des WirelessHART-Netzwerks **53** zu routen.

[0084] Wie in **Fig.** 5 dargestellt, sind das IPv6-fähige WirelessHART-Gateway **54**, das mit dem Netzwerk **53** verknüpft ist, und ein IPv6-fähiges WirelessHART-Gateway **70**, das mit dem Netzwerk **71** verknüpft ist, mit dem Internet **51** verbunden. Die IPv6-fähigen WirelessHART-Gateways **54** und **70** können jeweils als eigenständiges Gerät implementiert werden, z. B. als Karte, die in einen Erweiterungsslot von Workstations eingeführt wird (nicht gezeigt), als Teil des E/A-Subsystems eines auf einer Speicherprogrammierbaren Steuerung (SPS) basierenden Systems, oder in einer beliebigen anderen Weise. Zusätzlich zur Umwandlung von Protokollen und Befehlen können die IPv6-fähigen WirelessHART-Gateways **54** und **70** eine synchronisierte Taktung von Zeitfenstern und Superframes (Sätze von Kommunikations-Zeitfenstern in gleichem Abstand) der Zeitplanung der WirelessHART-Kommunikationsnetzwerke **53** und **71** bereitstellen.

[0085] Daneben kann das WirelessHART-Kommunikationsnetzwerk **53** Router-Geräte **57** und **58** beinhalten. Das Router-Gerät **57** ist beispielsweise ein Netzwerkgerät, das Datenframes von einem drahtlosen Netzwerkgerät an ein anderes weiterleitet. Ein Netzwerkgerät, das als Router-Gerät agiert, verwendet interne Routing-Tabellen, um zu entscheiden, welches drahtlose Netzwerkgerät der nächste Empfänger eines bestimmten Datenframes ist. Eigenständige Router wie der Router **57** sind in diesen Szenarien möglicherweise nicht erforderlich, wenn alle Geräte im WirelessHART-Kommunikationsnetzwerk **53** das Routing unterstützen. Jedoch kann es vorteilhaft sein (z. B. um das Netzwerk zu erweitern oder um die Leistung eines Feldgeräts in dem Netzwerk zu speichern), im WirelessHART-Kommunikationsnetzwerk **53** eigene Router zu haben, z. B. das Router-Gerät **57**.

[0086] Das WirelessHART-Kommunikationsnetzwerk **53** wird auch mit IPv6-fähigen WirelessHART-Feldgeräten **62**, **63**, **64** und **65** dargestellt. Das IPv6-fähige WirelessHART-Feldgerät **62** kann zum Beispiel in der Lage sein, in Informationen des WirelessHART-Netzwerkrouting-Protokolls eingekapselte 6LoWPAN IPv6-Datenframe-Fragmente zu empfangen und zu senden, und es kann in Informationen des WirelessHART-Netzwerkrouting-Protokolls eingekapselte 6LoWPAN IPv6-Datenframe-Fragmente erzeugen und routen. Ähnlich ist das IPv6-fähige WirelessHART-Feldgerät **62** in der Lage, zum Beispiel Nicht-IPv6-fähige Datenframes mit einem eingekapselten WirelessHART-Protokoll-Header an ein anderes WirelessHART-Feldgerät zu routen. Um das Routing durchzuführen, verwendet das WirelessHART-Feldgerät **62** zum Beispiel die Routing-Informationen im Header für das WirelessHART-Netzwerkrouting, der einen bestimmten Datenframe einkapselt.

[0087] Das WirelessHART-Kommunikationsnetzwerk **53** kann auch die klassischen WirelessHART-Feldgeräte **59**, **60** und **61** umfassen. Die WirelessHART-Feldgeräte **59**, **60** und **61** sind in der Lage, Datenframes mit eingekapselten Informationen des WirelessHART-Netzwerkprotokolls zu empfangen und zu senden. Die WirelessHART-Feldgeräte **59**, **60** und **61** können ebenso in der Lage sein, in einem Header des WirelessHART-Netzwerkrouting-Protokolls eingekapselte 6LoWPAN IPv6-Datenframe-Fragmente an ein IPv6-fähiges WirelessHART-Feldgerät zu leiten, das als Empfänger der Datenframe-Fragmente vorgesehen ist. Um das Routing durchzuführen, verwendet das WirelessHART-Feldgerät **59** zum Beispiel die Routing-Informationen im Header für das WirelessHART-Netzwerkrouting-Protokoll, die ein 6LoWPAN IPv6-Datenframe-Fragment einkapseln, um das Routing durchzuführen. Während die WirelessHART-fähigen Feldgeräte **59**, **60** und **61** in der Lage sind, WirelessHART-Nachrichten oder -Datenframes zu routen, einschließlich solcher Datenframes, die IPv6-Datenframes enthalten, können diese Geräte IPv6-Datenframes an sich nicht decodieren oder benutzen.

[0088] Ein Router **58** im WirelessHART-Netzwerk **53** kann in der Lage sein, Datenframes mit einem eingekapselten Header eines WirelessHART-Protokolls beispielsweise von einem Feldgerät **59** zu empfangen. Der Router **58** kann den Datenframe an ein benachbartes Feldgerät **60** routen, beispielsweise auf Grundlage der Routing-Informationen im Header des WirelessHART-Netzwerkprotokolls.

[0089] In dem Beispielsystem aus **Fig.** 5 beinhaltet das WirelessHART-Kommunikationsnetzwerk **71** das IPv6-fähige WirelessHART-Gateway-Gerät **70**, ein oder mehrere Router-Geräte **74** und **75**, mehrere IPv6-fähige Feldgeräte **79**, **80**, **81** und **82** sowie mehrere klassische WirelessHART-Feldgeräte **76**, **77** und **78**. Die oben aufgeführten Geräte können über Kommunikationskanäle, die dem Standard 802.15.4 entsprechen, Datenframes empfangen und senden. In diesem Fall kann der Computer **82** im WirelessHART-Kommunikationsnetzwerk **71** ein Handheld-Computer sein. Der Computer **82** kann mit einem Adapter zur Verfügung gestellt werden, der fähig ist, Datenframes in dem WirelessHART-Kommunikationsnetzwerk **71** über die dem Standard 802.15.4 entsprechenden Kommunikationskanäle zu empfangen und zu senden. Daneben kann der Computer **82** IPv6-Datenframe-Fragmente, die in Informationen des WirelessHART-Netzwerkprotokolls eingekapselt sind, und Nicht-IPv6-Datenframes, die in Informationen des WirelessHART-Netzwerkprotokolls eingekapselt sind, von einem IPv6-fähigen Feldgerät **82** empfangen und senden. Falls gewünscht, kann der Computer **82** Nicht-IPv6-Datenframes, die in Informationen des WirelessHART-Netzwerkprotokolls eingekapselt sind, von einem nicht-IPv6-fähigen Feldgerät **81** empfangen und senden. In jedem dieser Beispiele werden die Routing-

Informationen in dem Header des WirelessHART-Netzwerkprotokolls, der die Datenframes einkapselt, durch die zwischengeschalteten IPv6-fähigen WirelessHART-Geräte und die nicht-IPv6-fähigen WirelessHART-Geräte verwendet, um die Datenframes innerhalb des WirelessHART-Netzwerks **71** an das Gerät zu routen, das der vorgesehene Empfänger des Datenframes ist, basierend auf den Informationen für das WirelessHART-Netzwerkrouting des Datenframes.

[0090] In einem hier beschriebenen Szenario können zwei IPv6-fähige WirelessHART-Geräte in den unterschiedlichen WirelessHART-Netzwerken **53** und **71** ebenfalls miteinander über IPv6-Datenframes kommunizieren, die über jedes der Netzwerke **53** und **71** und zwischen den Netzwerken **53** und **71** sowie über das Internet **51** gesendet werden. Als ein Beispiel kann der Computer **82** im Netzwerk **71** mit einem IPv6-fähigen Feldgerät **63** kommunizieren, das sich beispielsweise in dem anderen WirelessHART-Netzwerk **53** befindet, indem IPv6-Datenframes gesendet und empfangen werden, die über diese Netzwerke und über das Internet **51** gesendet werden. In diesem Szenario kann eine Anwendung, wie beispielsweise ein Webbrowser, der auf einem Computer **82** ausgeführt wird, einen IPv6-Datenframe erzeugen. Als Beispiel können die Daten in dem Datenframe einer Anforderung von Webseitendaten von dem IPv6-fähigen Feldgerät **63** entsprechen. Der Computer **82** kann einen TCP/IP-Netzwerkprotokoll-Stack verwenden, um entsprechend der Webseitenanforderung einen oder mehrere IPv6-Datenframes zu erzeugen.

[0091] Auch kann, wie oben erwähnt, der Computer **82** die zu benutzende IP-Adresse oder IP-Adressen bestimmen, um die IPv6-Datenframes zu erzeugen (z. B. mit dem Gerät oder dem Host für das Gerät **63** verknüpft), indem zuerst unter Verwendung des standardmäßigen WirelessHART-Messaging über das WirelessHART-Netzwerk **71** mit dem Gateway-Gerät **71** kommuniziert wird. In diesem Fall kann das Gerät **82** eine Nachricht mit dem Namen senden, z. B. einem Domain-Namen einer Website oder eines anderen Hosts, die in dem Gerät **63** erreicht werden sollen, und es kann das Gateway-Gerät **71** auffordern, DNS-Dienste zu implementieren (oder als DNS-Server im Internet **51** zu agieren oder diesen zu kontaktieren), um den Domain-Namen in eine IP-Adresse aufzulösen. Das Gateway-Gerät **70** kann dann diese IP-Adresse an das Gerät **82** zurücksenden, um es dem Gerät **82** zu ermöglichen, das IPv6-Datenpaket (zum Beispiel) mit der richtigen IPv6-Zieladresse darin zu erzeugen. Während das Gerät **82** standardmäßiges WirelessHART-Messaging benutzen könnte, um DNS-Dienste der IP-Adresse vom Gateway-Gerät **72** anzufordern, könnte das Gerät **82** ebenfalls an das Gateway-Gerät **70** adressierte IPv6 Datenpakete benutzen, um die IP-Adresse für eine andere Netzwerkeinheit unter Verwendung der hier beschriebenen Kommunikationstechniken anzufordern (z. B. durch Einkapseln eines solchen IP-Datenframes in Informationen für das WirelessHART-Netzwerkrouting und Senden des eingekapselten Datenframes an das Gateway-Gerät **70** zur Verarbeitung und Decodierung durch eine IP- oder socket-basierte Anwendung im Gateway-Gerät **70**). Ebenso könnte das Gateway-Gerät **70** über das WirelessHART-Netzwerk **71** die ermittelte IP-Adresse mittels einer IP-Datennachricht, eingekapselt in Informationen für das WirelessHART-Netzwerkrouting, oder über standardmäßiges WirelessHART-Messaging zurücksenden.

[0092] Nach ihrer Erzeugung können die IPv6-Datenframes dann durch einen 6LoWPAN-Protokoll-Stack fragmentiert werden und das oder die Fragmente kann bzw. können in die Informationen für das WirelessHART-Netzwerkrouting eingekapselt werden, die dazu ausgelegt sind, den Datenframe in der beschriebenen Weise mit Bezug auf das Flussdiagramm **40** aus **Fig.** 3 an das IPv6-fähige WirelessHART-Gateway **70** zu senden. Der Computer **82** kann dann die Datenfragmente unter Verwendung von WirelessHART-Netzwerkrouting über 802.15.4-konforme Kommunikationskanäle innerhalb des WirelessHART-Netzwerks **71** an das IPv6-fähige WirelessHART-Gateway **70** übertragen. Die Datenframe-Fragmente können vom IPv6-fähige WirelessHART-Gateway **70** empfangen werden, und das IPv6-fähige WirelessHART-Gateway **70** kann die verschiedenen Datenframe-Fragmente wieder zusammenfügen, um die IPv6-Datenframes, die der Webseitenanforderung entsprechen, zu erzeugen.

[0093] Das IPv6-fähige WirelessHART-Gateway **70** kann dann die Ziel-IP-Adresseninformation im IPv6-Header des wieder zusammengefügten IPv6-Datenframes verwenden, um die IPv6-Datenframes über das Internet **51** an ein IPv6-fähiges WirelessHART-Gateway **54** zu routen. In diesem Fall entspricht die IP-Zieladresse des IPv6-Datenpakets der IPv6-Adresse des IPv6-fähigen Feldgerätes **63**, und das Gateway **54** speichert eine Liste der IP-Adressen der Geräte im Netzwerk **53**. Natürlich kann das IPv6-fähige WirelessHART-Gateway **70** typische oder verfügbare TCP/IP-Routingprotokolle verwenden, um eine geeignete Route zum IPv6-fähigen WirelessHART-Gateway **54** über das Internet zu bestimmen.

[0094] Das IPv6-fähige WirelessHART-Gateway **54** kann dann das Verfahren des späteren Teils des Flussdiagramms **40** aus **Fig.** 3 benutzen, um die IPv6-Datenframes über das Netzwerk **53** an das IP-fähige Feldgerät **63** zu senden. Insbesondere kann das Gateway **54** die empfangenen IPv6-Datenframes in ein oder

mehrere 6LoWPAN-Datenframe-Fragmente fragmentieren und dann die Datenframe-Fragmente in WirelessHART-Informationen des Netzwerkrouting-Protokolls einkapseln, bevor die Datenframe-Fragmente über das WirelessHART-Netzwerk **53** an das Gerät **63** in diesem Netzwerk adressiert oder geroutet werden. Die WirelessHART-Informationen des Netzwerkrouting-Protokolls enthalten WirelessHART-Routing-Informationen, die es zwischengeschalteten WirelessHART-Geräten **59**, **58** und **65** ermöglichen, beispielsweise die IPv6-Datenframe-Fragmente (eingekapselt in Informationen für das WirelessHART-Netzwerkrouting) an das IPv6-fähige WirelessHART-Feldgerät **63** zu routen.

[0095] Das IPv6-fähige WirelessHART-Feldgerät **63** kann die verschiedenen Datenfragmente (eingekapselt in Informationen für das WirelessHART-Netzwerkrouting) in mehreren WirelessHART-Datenpaketen über das WirelessHART-Kommunikationsnetzwerk **53** empfangen und die Datenfragmente mit den Informationen für das WirelessHART-Netzwerkrouting decodieren, um zu überprüfen, ob die Datenframes für das Gerät **63** vorgesehen sind. In diesem Fall fügt das Gerät die verschiedenen Datenframe-Fragmente aus verschiedenen Datenpaketen zusammen, um die IPv6-Datenframes entsprechend der vom Computer **82** übermittelten Webseitenanforderung zu erzeugen, und gibt diese Anforderung an die Anwendung weiter, an die der IPv6-Datenframe ursprünglich gesendet wurde. Das Gerät kann die IP-Adressinformationen im IPv6-Datenpaket benutzen, um das richtige Ziel im Gerät **63** zu bestimmen. Das IPv6-fähige WirelessHART-Feldgerät **63** kann dann zum Beispiel nach der gleichen Methodik Hypertext Markup Language (HTML) IPv6-Datenframes entsprechend den Webseitendaten zurück an das anfragende Gerät **82** senden, d. h. durch das Erzeugen und das Senden dieser IPv6-Datenpakete über das WirelessHART-Netzwerk **53**, das Internet **51** und das WirelessHART-Netzwerk **71**. Insbesondere durch die Implementierung ähnlicher Schritte wie oben beschrieben, können die HTML-IPv6-Datenframes an den Computer **82** über ein oder mehrere WirelessHART-Feldgeräte in dem Netzwerk **53**, das IPv6-fähige WirelessHART-Gateway **54**, das Internet **51**, das IPv6-fähige WirelessHART-Gateway **70** und ein oder mehrere WirelessHART-Feldgeräte im Netzwerk **71** gesendet und empfangen werden.

[0096] Der Computer **82** kann durch Nutzung geeigneter Kommunikationsprotokolle auch mit einem oder mehreren anderen Kommunikationsnetzwerken kommunizieren. In Bezug auf **Fig.** 1 kann zum Beispiel der Computer **82** über das Gateway-Gerät **70** mit einem Mobilfunknetzwerk **15** kommunizieren. In einem anderen Szenario kann der Computer **82** über das Gateway-Gerät **70** mit einem WLAN-Netzwerk **16** kommunizieren. Weiter können IP-Nachrichten oder Datenpakete von einem ersten IP-fähigen Feldgerät in einem der WirelessHART-Netzwerke **53** oder **71** erzeugt und an ein anderes IP-fähiges Feldgerät innerhalb des gleichen WirelessHART-Netzwerks **53** oder **71** mittels der hier beschriebenen Kommunikationstechniken gesendet werden, um dadurch IP-Messaging zwischen zwei Geräten in einem Nicht-IP-Kommunikationsnetzwerk zu ermöglichen. In diesem Fall werden, wie man versteht, von dem ersten IP-fähigen Feldgerät erzeugte Datennachrichten mit Informationen für das WirelessHART-Netzwerkrouting codiert, um die Daten in der oben beschriebenen Weise über das WirelessHART-Kommunikationsnetzwerk **53** oder **71** zu senden. In diesem Fall müssen jedoch die Nachrichten möglicherweise immer noch durch eines der Gateway-Geräte **54** oder **70** gehen, weil diese Geräte Links oder Routing-Tabellen speichern, die Verbindungen zwischen IP-Adressen für Geräte im entsprechenden WirelessHART-Netzwerk **53** oder **71** und in Informationen für das WirelessHART-Netzwerkrouting für diese Geräte definieren. In einem anderen Fall können jedoch die IP-fähigen Geräte selbst IP-Adressen und Routing-Tabellen für die Intra-Netzwerkkommunikation speichern und diese IP-Routing-Tabellen benutzen, um die Informationen für das WirelessHART-Netzwerkrouting zu bestimmen, die für das Routing innerhalb des WirelessHART-Netzwerks in einen Header eines Datenpakets platziert werden müssen.

[0097] **Fig.** 6 zeigt ein weiteres Beispiel eines WirelessHART-Kommunikationsnetzwerks **100**, in dem WirelessHART-Feldgeräte nach dem WirelessHART-Kommunikationsprotokoll arbeiten, um IPv6-Datenpakete über das Internet **101** an und von Computer **115**, **116** und **117** zu empfangen und zu senden. Die Computer **115**, **116** und **117** können über Nicht-WirelessHART-Netzwerke kommunikativ an das Internet **101** gekoppelt sein, beispielsweise über Netzwerke, die WLAN-, GSM-, CDMA-, Bluetooth-, ZIGBEE-Protokolle usw. implementieren.

[0098] Das WirelessHART-Kommunikationsnetzwerk **100** beinhaltet ein IPv6-fähiges WirelessHART-Gateway **102**, mehrere WirelessHART-Router **105** und **106**, mehrere IPv6-fähige Feldgeräte **107**, **108**, **109** und **110** sowie und mehrere nicht-IPv6-fähige WirelessHART-Feldgeräte **111**, **112** und **113**. Wie in **Fig.** 6 dargestellt, ist das IPv6-fähige WirelessHART-Gateway **102** kommunikativ mit dem Internet **101** und mit mehreren Feldgeräten **103**, **104** und **111** innerhalb des Netzwerks **100** gekoppelt.

[0099] In einem Beispiel empfängt und sendet der Computer **115** unter Verwendung sowohl des Internets **101** als auch der Struktur des WirelessHART-Netzwerks **100** IPv6-Datenframes von einem und an ein IPv6-fähiges WirelessHART-Gerät **107**. Der Computer **115** kann kommunikativ über ein WLAN-Netzwerk **16** an das

Internet **101** gekoppelt sein, wie beispielsweise in **Fig.** 1 gezeigt. In diesem Beispiel wird jedoch angenommen, dass der Computer **115** nur IPv6-fähige Kommunikationsfähigkeiten hat und nicht in der Lage ist, direkt im WirelessHART-Netzwerk **100** zu kommunizieren. Jedoch kann der Computer **115** einen geeigneten Near Field Communication (NFC) Scanner beinhalten, beispielsweise einen RFID-Transceiver (Radio Frequency Identification, Funkerkennung) Scanner **115-1**. Das IPv6-fähige WirelessHART-Feldgerät **107** kann mit einem RFID-Tag **118** versehen werden, und dieses Tag kann einen Identifikationscode oder eine ID (z. B. eine IP-Adresse) speichern, der bzw. die der Identität des IPv6-fähigen WirelessHART-Feldgeräts **107** entspricht. Ein Benutzer in der Anlage kann zum Beispiel das RFID-Tag auf dem Gerät **107** über den RFID-Scanner **115-1** in den Computer **115** einscannen. Informationen auf dem RFID-Tag können einer IPv6-Adresse, einer Web-URL oder andere Daten entsprechen, die mit dem IPv6-fähigen WirelessHART-Feldgerät **107** verbunden sein können. In einem anderen Szenario kann der Computer **115** mit den von dem RFID-Tag **118** empfangenen Informationen eine Remote-Datenbank abfragen, um eine IPv6-Adresse oder eine Web-URL für das IPv6-fähige WirelessHART-Feldgerät **107** zu erhalten, damit der Computer **115** mit dem Feldgerät **107** kommunizieren kann.

[0100] Eine Anwendung, die auf dem Computer **115** ausgeführt wird, z. B. ein Web-Browser, kann entsprechend einer Anforderung von Webseitendaten einen IPv6-Datenframe erzeugen, der beispielsweise über die IP-Adresse des Feldgeräts **107** an das IPv6-fähige WirelessHART-Feldgerät **107** übertragen wird. Als solches kann die Ziel-IP-Adresse im IP-Header des IPv6-Datenframes der IPv6-Adresse des IPv6-fähigen WirelessHART-Feldgerätes **107** entsprechen. Der Computer **115** kann den IPv6-Datenframe zum Beispiel über ein WLAN-Netzwerk **16** an das Internet **101** übertragen. Der IPv6-Datenframe kann letztendlich mit Standard- oder IPv6-IP-Techniken für Netzwerkrouting-Protokolle über das Internet **101** an das IPv6-fähige WirelessHART-Gateway **102** geroutet und von diesem empfangen werden. Wie bereits beschrieben, kann das IPv6-fähige WirelessHART-Gateway **102** IP-Routing-Tabellen benutzen, um festzustellen, dass das mit der Ziel-IP-Adresse des IPv6-Datenframes verbundene Gerät sich innerhalb des WirelessHART-Netzwerks **100** befindet, und es kann WirelessHART-Netzwerkrouting-Informationen bestimmen, die benötigt werden, um über das Netzwerk **100** Informationen an dieses Gerät zu senden. Das IPv6-fähige WirelessHART-Gateway **102** kann dann den empfangenen IPv6-Datenframe fragmentieren und jedes der Datenframe-Fragmente mit den Informationen für das WirelessHART-Netzwerkrouting codieren, die nötig sind, um diese Datenframes an das Gerät **107** im Netzwerk **100** zu routen. Das Gateway **102** kann diese Informationen für das WirelessHART-Routing auch mit einem Flag codieren, das anzeigt, dass das zugrunde liegende Paket oder die Informationen von einem IPv6-Datenpaket stammen oder mit diesem verknüpft sind, und es kann dann die codierten Datenframes mithilfe des WirelessHART-Netzwerkrouting über das WirelessHART-Netzwerk **100** an das Feldgerät **107** (oder an eine IP-Adresse, die mit der Anwendung oder dem Feldgerät **107** verbunden ist) senden. In einem Beispiel kann das WirelessHART-Gateway **102** den IPv6-Datenframe gemäß dem 6LoWPAN-Protokoll fragmentieren und dann die resultierenden Datenframe-Fragmente in einen Protokoll-Header für das WirelessHART-Netzwerkrouting einkapseln, der WirelessHART-Adressierungsinformationen für das IPv6-fähige WirelessHART-Feldgerät **107** enthält, z. B. eine Graphen-ID. Natürlich überträgt das WirelessHART-Gateway **102** IPv6-Datenframe-Fragmente, die in das WirelessHART eingekapselt sind, über Low-Power-Kommunikationskanäle in das WirelessHART-Netzwerk **100**, zum Beispiel über Kommunikationskanäle, die nach dem Standard 802.15.4 arbeiten.

[0101] In diesem Beispiel empfängt das IPv6-fähige Feldgerät **107** die in WirelessHART eingekapselten IPv6-Datenframes über standardmäßige WirelessHART-Kommunikation über das Netzwerk **100**. Das IPv6-fähige Feldgerät **107** kann bei Empfang der in WirelessHART eingekapselten Pakete die verschiedenen Datenframe-Fragmente zusammenfügen, um einen IPv6-Datenframe zu erzeugen, der dem IPv6-Datenframe entspricht, der vom Computer **115** übertragen wurde, und diesen Datenframe einer Anwendung im Feldgerät **107** bereitstellen, an die das Paket adressiert war. Das IPv6-fähige Feldgerät **107** kann dann IPv6-Datenframes erzeugen, die Informationen enthalten, die vom Computer **115** angefordert wurden, und diese IPv6-Datenframes mithilfe der hier beschriebenen Techniken an den Computer **115** senden. Wie zuvor beschrieben, können die am Feldgerät **107** erzeugten IPv6-Datenframes fragmentiert werden, und die Fragmente können mit Protokoll-Headern für WirelessHART-Netzwerklayer in Feldgerät **107** eingekapselt werden. Das IPv6-fähige Feldgerät **107** kann dann die Datenframe-Fragmente über das WirelessHART-Netzwerk **100** an das WirelessHART-Gateway-Gerät **102** übertragen, um sie über das Internet **101** an das Gerät **115** zu senden. Das IPv6-fähige WirelessHART-Gateway **102** kann die empfangenen Datenframe-Fragmente wieder zusammenfügen, um die IPv6-Datenframes entsprechend den vom IPv6-fähigen Feldgerät **107** erzeugten IPv6-fähigen Datenframes zu erzeugen. Das IPv6-fähige WirelessHART-Gateway **102** überträgt dann die IPv6-Datenframes unter Verwendung der IP-Adresse der IPv6-Datenframes über das Internet **101** an die IP-Adresse des Geräts **115**. Der Computer **115** kann die IPv6-Datenframes zum Beispiel über das WLAN-Netzwerk **16** empfangen, und er kann die Informationen in den IPv6-Datenframes in einem Webbrowser decodieren und anzeigen, beispielsweise für den Benutzer. So kann in diesem Beispiel das Gerät **115**, das über keine WirelessHART-Kommunikationsfähigkeiten verfügt, die hier beschriebenen Kommunikationstechniken benutzen, um IPv6-Datenpakete direkt an

ein Feldgerät **107** in einem WirelessHART-Netzwerk zu kommunizieren, in dem die Kommunikation zwischen Geräten über die WirelessHART-Kommunikationsprotokolle (das kein IPv6-Datenframe-Routing unterstützt) erfolgen muss. Diese Technik ermöglicht es somit Anwendungen in Geräten außerhalb des WirelessHART-Netzwerks **100**, mithilfe von IP-basierter Kommunikation mit Geräten innerhalb des WirelessHART-Netzwerks zu kommunizieren, auch wenn das WirelessHART-Kommunikationsnetzwerk keine IP-basierte Kommunikation unterstützt. Diese Technik ermöglicht auch die Benutzung von Anwendungen in Geräten innerhalb des WirelessHART-Netzwerks **100**, um mittels IP-basierter Kommunikation oder socketbasierten Anwendungen mit Geräten außerhalb des WirelessHART-Netzwerks zu kommunizieren, auch wenn das WirelessHART-Kommunikationsnetzwerk von Natur aus keine IP-basierte Kommunikation unterstützt. Diese Vorteile ermöglichen es somit standardmäßigen IP-fähigen Anwendungen (die sehr weit verbreitet und bekannt sind), innerhalb von Geräten in spezialisierten Kommunikationsnetzwerken benutzt zu werden oder mit ihnen zu kommunizieren, z. B. im WirelessHART-Netzwerk, das keine IP-basierte Kommunikation benutzt. Diese Techniken ermöglichen wiederum, dass eine standardmäßige IP-basierte Kommunikation über das WirelessHART-Kommunikationsnetzwerk oder über andere Kommunikationsnetzwerke erfolgt, die kein IP-basiertes Netzwerkrouting oder IP-Adressierung unterstützen oder benutzen.

[0102] In einem anderen Beispiel kann ein QR-Reader **116** Informationen aus einem QR-Code **119** auslesen, der sich an einem IPv6-fähigen WirelessHART-Feldgerät **109** befindet. Die Informationen können die IPv6-Adresse für die Kommunikation mit dem IPv6-fähigen WirelessHART-Feldgerät **109** mithilfe von IP-basierter Kommunikation beinhalten. Der QR-Reader **116** kann kommunikativ über ein drahtgebundenes oder drahtloses Netzwerk mit dem Internet **101** gekoppelt sein und die aus dem QR-Code **119** ausgelesenen Informationen nutzen, um über das Internet **101** IPv6-Datenframes an das und von dem IPv6-fähigen WirelessHART-Feldgerät **109** auf die gleiche Weise wie in Bezug auf das Gerät **115** beschrieben zu senden und zu empfangen. In einem anderen Beispiel kann das Barcode-Lesegerät **117** den in dem IPv6-fähigen WirelessHART-Feldgerät **120** enthaltenen Barcode **120** auslesen. Das Barcode-Lesegerät **117** kann die in den Barcode **120** codierten Informationen verwenden, um IPv6-Datenframes über das Internet **101** an das IPv6-fähige WirelessHART-Feldgerät **109** zu senden und von ihm zu empfangen. Natürlich können viele andere Arten von Anwendungen benutzt werden, um mit IPv6-fähigen Feldgeräten innerhalb des Netzwerks **100** zu kommunizieren, um Informationen von diesen Geräten zu erhalten und um Informationen an diese Geräte zu senden (wie beispielsweise Befehle, Anweisungen, Datenanforderung, etc.) Daher kann in einem Fall ein Bediener oder Techniker IP-basierte Anwendungen in mit dem Internet verbundenen Geräten benutzen, um mit Feldgeräten innerhalb eines WirelessHART-Netzwerks oder einem andern Netzwerk zu kommunizieren, das kein IP-basiertes Netzwerkrouting nutzt, um Informationen jeglicher Art von diesen Geräten zu erhalten oder Informationen jeglicher Art an diese Geräte zu senden.

[0103] **Fig.** 7 ist ein Blockdiagramm eines Beispiels für einen Netzwerkprotokoll-Stack **300**, der genutzt werden kann, um Teile der hier beschriebenen Kommunikationstechniken zu implementieren, um einerseits IPv6-Datenframes aus dem Internet zu empfangen und dorthin zu senden, und um andererseits IPv6-Datenpakete über ein WirelessHART-Netzwerk zu routen. In diesem Fall kann der Kommunikationsprotokoll-Stack **300** zum Beispiel in das IPv6-fähige WirelessHART-Gateway **11a** aus **Fig.** 1 oder in eines der Gateway-Geräte **54** oder **70** der **Fig.** 5 oder in das Gateway-Gerät **102** aus **Fig.** 6 implementiert werden. Das Gerät, das den Stack **300** implementiert, kann ein Allzweckcomputer sein oder alternativ ein spezieller eingebetteter Computer, verbunden beispielsweise mit einem Feldgerät (das eine Prozesssteuerung sein kann) oder mit einem Prozesssteuerungsnetzwerk. Die verschiedenen Funktionsblöcke, die den Kommunikationsprotokoll-Stack **300** umfassen, können mittels Softwareanweisungen, Hardware, Firmware, ASICs, usw. implementiert werden. In dem Fall, dass Softwareanweisungen benutzt werden, können die Softwareanweisungen mehrere Blöcke enthalten, die durch einen Hardware-Mikroprozessor ausgeführt werden können, und diese Blöcke können kommunikativ und programmtechnisch über Software-Messaging-Warteschlangen, Software-Rückrufe, Speicherpuffer, Busse, usw. miteinander gekoppelt werden. Die Softwareblöcke können mehrere Hardware-Ressourcen benutzen, wie Timer, Hardware-Interrupts, Fast-Ethernet-Controller, serielle Kommunikations-Controller usw., um IPv6-Datenframes zu empfangen und zu senden. Obwohl nicht in **Fig.** 7 gezeigt, kann der Kommunikationsprotokoll-Stack **300** sich programmtechnisch mit einem Echtzeit-Betriebssystem (Real-Time Operating System, RTOS) verbinden. Beispiele von RTOS beinhalten PSOS, UCOS, RTLinux usw. Wie oben erwähnt, können einige oder alle der verschiedenen Funktionsblöcke, die den Kommunikationsprotokoll-Stack **300** ausmachen, in einem anwendungsspezifischen integrierten Schaltkreis (Application Specific Integrated Circuit, ASIC) implementiert werden, oder in einem feldprogrammierbaren Gate-Array(FPGA)-Gerät oder anderen Hardware-/Firmware-Geräten, wenn dies gewünscht wird.

[0104] In dem System aus **Fig.** 7 beinhaltet der Stack **300** allgemein eine erste Abzweigung **303** und eine zweite Abzweigung **305**. Die erste Abzweigung ist allgemein mit empfangenden und decodierenden oder

mit codierenden und übertragenden IP-Datenframes verbunden, z. B. IPv6-Datenframes über eine Ethernet-Schnittstelle **301** unter Verwendung eines mit dem IP-Netzwerkrouting konformen Kommunikationsnetzwerks. Die zweite Abzweigung **305** ist allgemein mit empfangenden und decodierenden oder mit codierenden und übertragenden IPv6-Datenpaketen oder anderen Arten von Paketen über ein nicht mit dem IP-Netzwerkrouting konformes Kommunikationssystem verbunden, z. B. über ein WirelessHART-Kommunikationsnetzwerk über einen WirelessHART-Zugangspunkt **313**. Die Abzweigungen **303** und **305** sind miteinander und mit den oberen Layern des Stacks verbunden (z. B. mit dem Transportlayer und den Layern des Stacks **300**), die allgemein auf IPv6-Datenframe-Kommunikation basieren oder sie benutzen.

[0105] Insbesondere die erste Abzweigung **303** des Kommunikationsprotokoll-Stacks **300** umfasst einen Ethernet-Treiber **302**, der mit der Ethernet-Schnittstelle **301** verbunden ist, die in diesem Fall eine einzige physische Schnittstelle umfassen kann. In anderen Fällen kann die Ethernet-Schnittstelle **301** zwei oder mehr physische Schnittstellen umfassen. In IPv4, IPv6 und IPv4–IPv6 eingekapselte Datenframes können über eine Ethernet-Schnittstelle **301** vom Internet empfangen (Eingang) und dorthin übertragen (Ausgang) werden und somit ein Stack-Element eines physischen Layers und/oder ein Stack-Element eines Datenlinklayers implementieren. In diesem Beispiel wird die Ethernet-Schnittstelle **301** so angepasst, dass sie nach dem IEEE-Standard 802.3 arbeitet. Jedoch kann die Ethernet-Schnittstelle **301** gemäß einer anderen der synchronen, asynchronen und/oder isochronen seriellen Kommunikationsstandards arbeiten, und der Ethernet-Treiber **302** kann nach Bedarf auch die Ethernet-Schnittstelle **301** konfigurieren. Natürlich empfängt und überprüft der Ethernet-Treiber **302** die Integrität der über die Ethernet-Schnittstelle **301** empfangenen Datenframes, und der Ethernet-Treiber **302** kann die empfangenen Datenframes in einem oder mehreren Speicherpuffern in dem Gerät speichern, in dem der Stack **300** läuft. Der Ethernet-Treiber **302** kann einen Block zur IPv4-Verkapselung/Entkapselung **304** alarmieren, wenn ein Datenframe über die Ethernet-Schnittstelle **301** empfangen wird und/oder wenn ein Datenframe in den Speicherpuffern gespeichert wird, und der Ethernet-Treiber **302** kann den Block zur IPv4-Verkapselung/Entkapselung **304** mit einer Angabe entsprechend der Position des empfangenen Datenframes in den Speicherpuffern bereitstellen. Der Ethernet-Treiber **302** kann auch über die Ethernet-Schnittstelle **301** vom Block zur IPv4-Verkapselung/Entkapselung **304** empfangene Datenframes übertragen.

[0106] Der Block zur IPv4-Verkapselung/Entkapselung **304** funktioniert so, dass er die über die Ethernet-Schnittstelle **301** empfangenen Datenframes verarbeitet. Der Block zur IPv4-Verkapselung/Entkapselung **304** kann bestimmen, ob der empfangene Datenframe ein IPv4-Datenframe, ein IPv6-Datenframe oder ein in IPv4-eingekapselter IPv6-Datenframe ist. Falls gewünscht, kann der Block zur IPv4-Verkapselung/Entkapselung **304** so in einem IPv4-eingekapselten IPv6-Datenframe agieren, dass er einen IPv6-Datenframe erzeugt. Der Block zur IPv4-Verkapselung/Entkapselung **304** kann während oder nach abgeschlossener Entkapselung eines empfangenen IPv4-eingekapselten IPv6-Datenframes auf einen Block zur IPSec-Verschlüsselung/Entschlüsselung **306** aufmerksam machen.

[0107] Natürlich kann der Block zur IPv4-Verkapselung/Entkapselung **304** auch Datenframes von dem Block zur IPSec-Verschlüsselung/Entschlüsselung **306** erhalten, und der Block zur IPv4-Verkapselung/Entkapselung **304** kann bestimmen, ob ein von dem Block zur IPSec-Verschlüsselung/Entschlüsselung **306** empfangener Datenframe ein IPv4-Datenframe oder ein IPv6-Datenframe ist. Der Block zur IPv4-Verkapselung/Entkapselung **304** kann so konfiguriert sein, dass er einen IPv4-eingekapselten IPv6-Datenframe aus dem vom Block zur IPSec-Verschlüsselung/Entschlüsselung **306** empfangenen IPv6-Datenframe erzeugt. Der Block zur IPv4-Verkapselung/Entkapselung **304** kann so agieren, dass er keinen IPv4-Datenframe einkapselt, der vom Block zur IPSec-Verschlüsselung/Entschlüsselung **306** empfangenen wurde. In jedem Fall stellt der Block zur IPv4-Verkapselung/Entkapselung **304** dem Ethernet-Treiber **302** den IPv4-eingekapselten IPv6-Datenframe oder den IPv4-Datenframe zur Verfügung. Wie bereits beschrieben, kann der Ethernet-Treiber **302** über die Ethernet-Schnittstelle **301** vom Block zur IPv4-Verkapselung/Entkapselung **304** empfangene Datenframes übertragen. In einem reinen IPv6-Netzwerk wird der Block zur IPv4-Verkapselung/Entkapselung **304** möglicherweise nicht benötigt.

[0108] Der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** empfängt entkapselte IPv6-Datenframes und/oder IPv4-Datenframes vom Block zur IPv4-Verkapselung/Entkapselung **304**. Auf diese Weise stellt der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** fest, ob der empfangene Datenframe verschlüsselt ist. Falls erforderlich, kann der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** den empfangenen Datenframe mit einem oder mehreren Verschlüsselungs-/Entschlüsselungs-Algorithmen entschlüsseln. Eine unvollständige Liste der häufigsten Beispiele für Verschlüsselungs-/Entschlüsselungs-Algorithmen bieten der Data Encryption Standard (DES), Advanced Encryption Standard (AES), etc. Um die Verschlüsselung/Entschlüsselung durchzuführen, kann der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** Protokolle implementieren (nicht gezeigt), welche die Programmierung und den Austausch von Verschlüsselungs-/Entschlüsselungs-

Schlüsseln ermöglichen. Nach Fertigstellung stellt der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** einen Datenframe-Routingblock **308** mit dem entschlüsselten Datenframe bereit.

[0109] Andererseits empfängt der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** auch Datenframes von dem Datenframe-Routingblock **308** und ist so konfiguriert, dass er die Datenframes mit einem geeigneten Verschlüsselungsalgorithmus verschlüsselt. In diesem Fall stellt der Block zur IPSec-Verschlüsselung/Entschlüsselung **306** dem Block zur IPv4-Verkapselung/Entkapselung **304** entweder verschlüsselte oder unverschlüsselte Datenframes bereit, die über die Ethernet-Schnittstelle **301** übertragen werden.

[0110] Im Allgemeinen empfängt der Datenframe-Routingblock **308**, der ein Stack-Element des Netzwerkrouting-Layers implementieren oder enthalten kann, von einem oder mehreren Datenframe-Sendern einen Datenframe. Basierend auf den Informationen in den empfangenen Datenframes überträgt der Datenframe-Routingblock **308** die empfangenen Datenframes an einen oder mehrere vorgesehene Empfänger der Datenframes, wobei die Empfänger entweder Anwendungen innerhalb des Anwendungslayers des Stacks **300** sind, oder Anwendungen in Geräten im Ethernet-Kommunikationsnetzwerk und erreichbar über die Stack-Abzweigung **303**, oder Geräte im WirelessHART-Netzwerk und erreichbar über die Stack-Abzweigung **305**. Natürlich kann ein vorgesehener Empfänger des Datenframes in einem Fall auch als Datenframe-Sender in anderen Fällen arbeiten. Generell umfassen die unmittelbaren Datenframe-Sender und Datenframe-Empfänger im Stack aus **Fig.** 7 den Block zur IPSec-Verschlüsselung/Entschlüsselung **306**, einen 6LoWPAN-Anpassungslayer **310** und einen Transportlayer **311**, die in diesem Fall das TCP-, UDP- und ICMP-Übertragungsprotokoll benutzen.

[0111] Während des Betriebs, beim Empfang von Datenframes vom Block zur IPSec-Verschlüsselung/Entschlüsselung **306**, untersucht der Datenframe-Routingblock **308** die empfangenen Datenframes, um die IP-Adresse des vorgesehenen Empfangsgerätes der empfangenen Datenframes zu bestimmen. Der Datenframe-Routingblock **308** kann feststellen, ob der vorgesehene Empfänger des Datenframes das IP-erweiterte Gateway selbst ist, wobei der Datenframe-Routingblock **308** den Transportlayer-Stack **311** mit einer Anzeige darauf aufmerksam machen kann, dass ein Datenframe zur Verfügung steht. Details des Transportlayer-Stacks **311** (der, falls gewünscht, auch ein Teil des Stack-Elements des Netzwerklayers im Stack **300** sein oder es implementieren kann) werden später genauer ausgeführt. Alternativ kann der Datenframe-Routingblock **308** feststellen, ob das vorgesehene Empfängergerät der Datenframes ein oder mehrere IPv6-fähige WirelessHART-Geräte innerhalb eines WirelessHART-Netzwerks sind, das an ein IP-fähiges Gateway-Gerät IP angeschlossen ist, in dem der Datenframe-Routingblock **308** über den WLAN-Zugangspunkt **313** lokalisiert wird. In diesem Fall stellt der Datenframe-Routingblock **308** dem 6LoWPAN-Anpassungslayer **310** den Datenframe bereit. Weiterhin kann der Datenframe-Routingblock **308** feststellen, ob das vorgesehene Empfängergerät des Datenframes ein oder mehrere Geräte sind, die mit dem IP-fähigen Gateway-Gerät verbunden sind, in dem der Datenframe-Routingblock **308** über die Ethernet-Schnittstelle **301** lokalisiert wird. In diesem Fall stellt der Datenframe-Routingblock **308** den Datenframe dem Block zur IPSec-Verschlüsselung/Entschlüsselung **306** bereit.

[0112] Der Datenframe-Routingblock **308** kann eine oder mehrere ”Routing-Tabellen” (in **Fig.** 7 als Routing-Tabellen **309** dargestellt) führen, um den Datenframe-Routingblock **308** bei der Bestimmung zu unterstützen, wohin ein bestimmter Datenframe zu routen ist. Im Allgemeinen enthalten die Routing-Tabellen **309** einen oder mehrere Routeneinträge, und jeder Routeneintrag kann eine Teilnetzmaske, die IP-Adresse von einem anderen Gateway oder Router, usw. beinhalten. Die Einträge in der Routing-Tabelle **309** können vom Datenframe-Routingblock **308** verwendet werden, um die Adresse oder die Position des nächsten Gateways/Routers zu bestimmen, an die der Datenframe gesendet wird, damit der Datenframe die vorgesehenen, mit dem Internet verbundenen Empfängergeräte erreichen kann. Die von dem Datenframe-Routingblock **308** in der IP-Routing-Tabellen **309** geführten Routing-Tabelleneinträge können konfigurierbar sein, und die Routing-Tabellen **309** können mit Routen-Einträgen aktualisiert werden, die aus dem Internet oder in einigen Fällen über das WirelessHART-Netzwerk empfangen wurden.

[0113] Der Datenframe-Routingblock **308** kann auch IPv6-Datenframes vom 6LoWPAN-Anpassungslayer **310** erhalten, und/oder IPv6-Datenframes vom Netzwerklayer-Stack **311** des IP-fähigen Gateways. Der Datenframe-Routingblock **308** kann die IP-Adresse des vorgesehenen Empfängergerätes der empfangenen IPv6-Datenframes feststellten, und auf der Grundlage der Informationen in der Routing-Tabelle **309** kann der Datenframe-Routingblock **308** den IPv6-Datenframe einem entsprechenden Routeneintrag zuordnen. Der Datenframe-Routingblock **308** kann dann die IPv6-Datenframes an den Block zur IPSec-Verschlüsselung/Entschlüsselung **306** mit dem zugehörigen Routeneintrag übertragen, oder er kann die IPv6-Datenframes zum Verarbeiten und Senden über das WirelessHART-Netzwerk an die zweite Abzweigung **305** senden, falls die IP-Adresse des IPv6-Pakets einem Gerät im WirelessHART-Netzwerk zugeordnet ist.

[0114] Generell gesagt, um ein IPv6-Datenpaket über das WirelessHART-Netzwerk zu senden, empfängt der 6LoWPAN-Anpassungslayer **310** IPv6-Datenframes vom Datenframe-Routingblock **308**, wenn der Datenframe-Routingblock **308** feststellt, dass ein Datenframe an ein Gerät innerhalb des WirelessHART-Netzwerks, mit dem das IP-fähige Gateway-Gerät verbunden ist, oder an ein mit diesem Netzwerk verknüpftes Gerät gesendet werden muss. In diesem Fall führt der 6LoWPAN-Anpassungslayer **310** eine Datenframe-Fragmentierung durch, um einen oder mehrere 6LoWPAN-Datenframes aus dem vom Datenframe-Routingblock **308** empfangenen IPv6-Datenframe zu erzeugen, und dadurch kann der 6LoWPAN-Anpassungslayer **310** jeden der erzeugten 6LoWPAN-Datenframes mit der IPv6-Adresse des vorgesehenen Empfängers des IPv6-Datenframes verknüpfen. Natürlich kann die IPv6-Adresse des vorgesehenen Empfängers des IPv6 Datenframes auch vom IPv6-Datenframe selbst abgerufen werden.

[0115] Wie jedoch in **Fig.** 7 veranschaulicht, kann der 6LoWPAN-Anpassungslayer **310** IPv6-Datenframes auch direkt vom Transportlayer-Stack-Element **311** (das auch, zumindest teilweise, ein Stack-Element des Netzwerklayers für das IP-Netzwerk implementieren kann) in dem Gateway-Gerät zur Implementierung des Stacks **300** empfangen, um an ein Gerät gesendet zu werden, das sich im WirelessHART-Netzwerk befindet oder damit verbunden ist. In diesem Fall funktioniert der 6LoWPAN-Anpassungslayer **310** so, dass er aus dem vom Transportlayer-Stack **311** empfangenen IPv6-Datenframe einen oder mehrere 6LoWPAN-Datenframes erzeugt. Generell gesagt, wenn der 6LoWPAN-Anpassungslayer **310** IPv6-Datenframes vom Datenframe-Routingblock **308** und/oder vom Stack **311** empfängt, erzeugt der 6LoWPAN-Anpassungslayer **310** einen oder mehrere 6LoWPAN Datenframes für jeden IPv6-Datenframe, der empfangen wird. Der 6LoWPAN-Anpassungslayer **310** überträgt dann den oder die 6LoWPAN-Datenframes an einen WirelessHART-Netzwerklayer **315** (der ein Stack-Element des Netzwerklayers für das WirelessHART-Kommunikationsnetzwerk ist) in der Abzweigung **305**. Als Teil dieser Kommunikation kann der WirelessHART-Netzwerklayer **315** die IPv6-Adresse des vorgesehenen Empfängers des oder der 6LoWPAN-Datenframes, die vom 6LoWPAN-Anpassungsblock **310** empfangen wurden, empfangen, oder der WirelessHART-Netzwerklayer **315** kann solche IP-Adressen selbst anhand der Datenframes feststellen.

[0116] Natürlich kann, wie man versteht, der 6LoWPAN-Anpassungslayer **310** auch 6LoWPAN-Datenframes vom WirelessHART-Netzwerklayer **315** empfangen. In diesem Fall funktioniert der 6LoWPAN-Block **310** so, dass er einen oder mehrere IPv6-Datenframes aus den empfangenen 6LoWPAN-Datenframes durch Defragmentieren dieser Datenframes erzeugt, d. h. durch die Kombination von Daten-Payloads aus verschiedenen 6LoWPAN-Datenframes oder fragmentierten Datenframes, um einen IPv6-Datenframe zu erzeugen. In diesem Fall kann der 6LoWPAN-Block **310** die erzeugten IPv6-Datenframes an den Datenframe-Routingblock **308** oder an den Transportlayer-Stack **311** übertragen.

[0117] Nun, wenn der WirelessHART-Netzwerklayer **315** 6LoWPAN-Datenframes vom 6LoWPAN-Block **310** erhält, kann der WirelessHART-Netzwerklayer **315** auf eine WirelessHART-Routing- oder Verknüpfungstabelle **316** zugreifen, die als Teil des Betriebs des Stacks **300** gespeichert oder bewahrt wird, um die Positions- oder Routing-Informationen für das Gerät innerhalb des WirelessHART-Netzwerks zu bestimmen, das die IP-Adresse in oder den 6LoWPAN-Datenframes hat oder damit verbunden ist (die IP-Adresse des ursprünglichen IPv6-Datenframes). Natürlich kann die WirelessHART-Routing- oder Verknüpfungstabelle **316** mehrere Routingeinträge umfassen, und jeder Eintrag in der Routing- oder Verknüpfungstabelle kann die IPv6-Adresse eines IPv6-fähigen WirelessHART-Geräts im WirelessHART-Netzwerk und die WirelessHART-Routing-Informationen für das IPv6-fähige WirelessHART-Gerät beinhalten. Somit speichert die Routing- oder Verknüpfungstabelle **316** Informationen, die bestimmte IP-Adressen (z. B. Routing-Informationen im ersten Netzwerkrouting-Protokoll) für Geräte im WirelessHART-Kommunikationsnetzwerk mit WirelessHART-Routing-Informationen, wie Graphen-IDs usw., (d. h. Routing-Informationen für das zweite Netzwerkrouting-Protokoll) für diese Geräte verknüpfen, damit diese Geräte mithilfe des WirelessHART-Routing-Protokolls erreichbar sind. In diesem Fall kann der WirelessHART-Netzwerklayer **315** die IPv6-Adresse mit jedem der zugeordneten 6LoWPAN-Datenframes nutzen, um WirelessHART-Routing-Informationen für das WirelessHART-Gerät ”nachzuschlagen”, das die IPv6-Adresse aus der WirelessHART-Routing-Tabelle **316** hat. Der WirelessHART-Netzwerklayer **315** kann dann jeden der 6LoWPAN-Datenframes mit Netzwerk-Routing-Informationen einkapseln, die dem WirelessHART-Netzwerkrouting-Protokoll entsprechen, die nötig sind, um ein Datenpaket an das Gerät innerhalb des WirelessHART-Netzwerk zu routen, und dabei die Netzwerk-Routing-Informationen für das IPv6-Protokoll in diesen Datenpaketen bewahren. Darüber hinaus kann der WirelessHART-Netzwerklayer **315** Informationen für das WirelessHART-Netzwerkrouting-mit einem Flag codieren, das anzeigt, dass das zugrunde liegende Datenpaket ein IPv6-Datenpaket ist oder auf einem IPv6-Datenpaket basiert (d. h. ein 6LoWPAN-Fragment eines IPv6-Datenpakets ist).

[0118] Natürlich kann der WirelessHART-Netzwerklayer **315** auch in WirelessHART eingekapselte 6LoWPAN-Datenframes von einem seriellen Port-Treiber **312** empfangen. In diesem Fall kann der WirelessHART-Netzwerklayer **315** aus den empfangenen, in WirelessHART eingekapselten 6LoWPAN-Datenframes einen oder mehrere 6LoWPAN-Datenframes erzeugen, indem die Informationen für das WirelessHART-Netzwerkrouting aus diesen Datenpaketen entfernt werden und die erzeugten 6LoWPAN-Datenframes für den 6LoWPAN-Anpassungslayer **310** bereitgestellt werden.

[0119] Der serielle Treiber **312** kann mit einem seriellen Empfänger-/Sender-Gerät (nicht gezeigt) eine Verbindung schaffen (die ein Stack-Element des Datenlinklayers und ein Stack-Element des physikalischen Layers implementiert), um innerhalb des WirelessHART-Netzwerks Kommunikation zu ermöglichen. In einem Beispiel kann das serielle Empfänger-/Sendergerät gemäß dem RS-232-Protokoll arbeiten, während in einem anderen Beispiel das serielle Empfänger-/Sendergerät gemäß dem RS-485-Protokoll arbeiten kann. Die hier aufgeführten seriellen Protokolle dienen nur als Beispiele, denn im Allgemeinen können alle geeigneten drahtgebundenen oder drahtlosen, seriellen oder parallelen Protokolle genutzt werden. In jedem Fall überträgt der serielle Treiber **312** über das serielle Empfangs-/Sendegerät in WirelessHART eingekapselte 6LoWPAN-Datenframes an WirelessHART-Zugangspunkte (Access Points, AP), wie zum Beispiel die Zugangspunkte **103** und **104** aus **Fig.** 6. Natürlich kann der serielle Treiber **312** über das serielle Empfangs-/Sendegerät auch in WirelessHART eingekapselte 6LoWPAN-Datenframes von einem WirelessHART-Zugangspunkt (AP) empfangen.

[0120] Wie zuvor erörtert, kann, wenn der Datenframe-Routingblock **308** feststellt, dass der Transportlayer-Stack **311** der vorgesehene Empfänger eines IPv6-Datenframes ist (zum Beispiel unter Verwendung der IP-Routing-Tabelle **309**), der Datenframe-Routingblock **308** den empfangenen Datenframe prüfen (und insbesondere die Transportinformationen im Header des Datenframes), um zu bestimmen, ob der Datenframe einem TCP-, UDP- oder ICMP-Datenframe entspricht. Beim Feststellen, ob ein empfangener Datenframe einem TCP-, UDP- oder ICMP-Datenframe entspricht, kann jeweils der Datenframe-Routingblock **308** einen TCP-Block **311-1**, einen UDP-Block **311-2** oder einen ICMP-Block **311-3** des Pakets alarmieren und das Paket dem entsprechenden Abschnitt des Transportlayers des Stacks **311** bereitstellen. In diesem Fall können eine oder mehrere standardmäßige und/oder kundenspezifische Anwendungen in einem Anwendungslayer **314** des Geräts liegen, in dem sich der Stack **300** befindet, und diese Standardanwendungen können zum Beispiel diejenigen umfassen, die das File Transfer Protokoll (FTP), TELNET, Hypertext Transfer Protokoll (HTTP), Simple Network Management Protokoll (SNMP), Dynamic Host Control Protokoll (DHCP), usw. implementieren. Kundenspezifische Anwendungen können diejenigen umfassen, die bestimmte Aufgaben ausführen, einschließlich beispielsweise der Überwachung und Konfiguration des IP-fähigen Gateways. Die Anwendungen im Anwendungslayer **314** können innerhalb eines Software-Frameworks laufen, der vom Framework und der Schnittstelle für die Anwendungsprogrammierung **315** bereitgestellt wird. Residente Anwendungen können Softwaremodule im Framework und der Schnittstelle für die Anwendungsprogrammierung **315** verwenden, um Datenframes vom TCP-Block **311-1**, vom UDP-Block **311-2** oder vom ICMP-Block **311-3** zu empfangen. In einigen Fällen kann das Framework und die Schnittstelle für die Anwendungsprogrammierung **315** eine ”Socket-”Schnittstelle bereitstellen, um es ausführenden Anwendungen zu ermöglichen, Datenframes zu empfangen, zu verarbeiten und zu senden. Daneben können residente Anwendungen Datenframes als Antwort auf empfangene Datenframes erzeugen. Die residenten Anwendungen können Datenframes erzeugen, autonom oder als Antwort auf empfangene Datenframes, und die residenten Anwendungen können die erzeugten Datenframes unter Verwendung von Softwaremodulen übertragen, die sich im Framework und der Schnittstelle für die Anwendungsprogrammierung **315** befinden. Diese Module können wiederum die erzeugten Datenframes an den TCP-Block **311-1**, den UDP-Block **311-2** oder den ICMP-Block **311-3** übertragen. Der TCP-Block **311-1**, der UDP-Block **311-2** oder der ICMP-Block **311-3** kann dann zusätzliche Informationen an die Datenframes anhängen und die Datenframes an den Datenframe-Routingblock **308** für die Bereitstellungen an ein anderes Gerät übertragen, entweder über die Ethernet-Schnittstelle **301** oder den WirelessHART-Zugangspunkt **313**.

[0121] **Fig.** 8 ist ein Blockdiagramm eines anderen Beispiels eines Netzwerkprotokoll-Stacks **320**, der in ein IP-fähiges WirelessHART-Feldgerät implementiert werden kann, beispielsweise das Feldgerät **109** aus **Fig.** 5, um sowohl WirelessHART-Kommunikationen als auch Kommunikationen, die IPv6-Datenframes nutzen (oder andere IP-Datenframes) zu implementieren. Generell hat der Protokoll-Stack **320** ein gemeinsames Set von unteren Layern **321**, die den physischen Layer und den Datenlinklayer des WirelessHART-Kommunikationsprotokolls sowie einen erweiterten WirelessHART-Netzwerklayer des Stacks **320** implementieren. Der Protokoll-Stack **320** umfasst auch zwei Sätze der oberen Layer **323** und **325**, von denen jedes Set der oberen Layern für die Implementierung der Transport- und Anwendungslayer entweder des WirelessHART-Kommunikationsnetzwerkprotokolls (**323**) oder eines IP-Kommunikationsnetzwerkprotokolls (**325**) verantwortlich ist. Hier wird das gemeinsame Set der unteren Layern **321** so dargestellt, dass es einen 802.15.4 PHY-Block **322** umfasst, der über Kommunikationskanäle, die dem Protokoll-Standard 802.15.4 entsprechen und mit dem Wireless-

HART-Kommunikationsnetzwerk verbunden sind, Datenframes empfängt und sendet. Die Kommunikationskanäle können zum Beispiel Kommunikationskanälen in einem WirelessHART-Kommunikationsnetzwerk **100** wie in **Fig.** 6 entsprechen. Der 802.15.4 PHY-Block **322** kommuniziert mit Hardware-Geräten wie digitalen Abschwächern, Synthesizern, analogen Filter, Analog-Digital-Wandler (ADC) und Digital-Analog-Wandlern sowie anderen Geräten innerhalb des WirelessHART-Kommunikationsnetzwerks. Dazu implementiert der 802.15.4 PHY-Block **322** Algorithmen, um die Integrität der empfangenen Datenframes zu filtern, zu demodulieren und zu überprüfen; Algorithmen, um Datenframes in ein Format zu übertragen, das dem Protokollstandard 802.15.4 vor der Übertragung entspricht; und Algorithmen, um die Übertragung von Datenframes zu planen, die dem Protokollstandard 802.15.4 entsprechen. Wie es allgemein der Fall ist, ist der 802.15.4 PHY-Block **322** mit einem WirelessHART Media Access Control (MAC) Block **324** verbunden und überträgt einen Hinweis an den Block **324**, wenn ein Datenframe gemäß Protokollstandard 802.15.4 empfangen wird. Der 802.15.4 PHY-Block **322** (der ein Stack-Element eines physischen Layers oder ein Element eines physischen Layers ist) empfängt ebenfalls einen Hinweis von dem WirelessHART MAC-Block **324**, wenn ein Datenframe bereitsteht, um über Kommunikationskanäle übertragen zu werden, die dem Protokollstandard 802.15.4 entsprechen und in Standardmanier agieren, um den Datenframe in einem solchen Fall über das WirelessHART-Netzwerk an ein anderes Gerät innerhalb des WirelessHART-Netzwerks zu übertragen.

[0122] In Allgemeinen führt der WirelessHART MAC-Block **324** die Schritte der Implementierung des Routings von empfangenen Datenframes gemäß dem WirelessHART-Datenlinklayer-Protokoll aus und ist somit ein Stack-Element des Datenlinklayers. Der WirelessHART MAC-Block **324** kann den Header des WirelessHART-Datenlinklayer-Protokolls eines empfangenen Datenframes untersuchen, um den vorgesehenen Empfänger des empfangenen Datenframes im WirelessHART-Netzwerk zu bestimmen. Zum Beispiel kann der WirelessHART MAC-Block **324** das Zieladressfeld im Header des WirelessHART-Datenlinklayer-Protokolls verwenden, um den vorgesehenen Empfänger des empfangenen Datenframes zu bestimmen. In einem Szenario kann die Zieladresse im Header des WirelessHART-Protokolls eines empfangenen Datenframes der WirelessHART MAC-Adresse des Feldgerätes entsprechen, das den WirelessHART MAC-Block **324** implementiert. In diesem Fall kann der WirelessHART MAC-Block **324** eine Anzeige an den erweiterten WirelessHART-Netzwerklayer **326** übertragen, der das Vorhandensein dieses Datenpakets anzeigt, und er kann das Paket zur Verarbeitung in dem Netzwerklayer **326** weiterleiten. In einem anderen Szenario kann der WirelessHART MAC-Block **324** feststellen, ob ein anderes Feldgerät der vorgesehene Empfänger eines empfangenen Datenframes ist. In diesem Szenario kann der WirelessHART MAC-Block **324** die empfangenen Daten über den 802.15.4 PHY-Block **322** an das vorgesehene Empfangsgerät des Datenframes übertragen.

[0123] Der WirelessHART MAC-Block **324** kann auch Datenframes vom erweiterten WirelessHART-Netzwerkblock **326** empfangen (der ein Stack-Element des Netzwerklayers für das WirelessHART-Netzwerk ist). Wie man versteht, kann der WirelessHART MAC-Block **324** den Datenframe einkapseln, um ihn über das WirelessHART-Netzwerk mit einem Header des WirelessHART-Datenlink-Protokolls zu senden. Der WirelessHART MAC-Block **324** kann den Protokoll-Header des WirelessHART-Datenlinklayers mit Informationen aktualisieren, die beispielsweise der Zieladresse des oder der vorgesehenen Empfangsgeräte entsprechen. Wie bereits beschrieben, kann der WirelessHART MAC-Block **324** den 802.15.4 PHY-Block **322** darauf hinweisen, dass ein Datenframe zur Übertragung bereitsteht.

[0124] Der erweiterte WirelessHART-Netzwerkblock **326** funktioniert so, dass er einen Hinweis vom WirelessHART MAC-Block **324** empfängt, wenn die Zieladresse z. B. in dem WirelessHART-Protokoll-Header eines empfangenen Datenframes anzeigt, dass das Feldgerät der vorgesehene Empfänger des Datenframes ist. Wie zuvor erläutert, kann eine Sendegerät, zum Beispiel ein IPv6-fähiges WirelessHART-Gateway **11** aus **Fig.** 1 oder ein anderes Feldgerät im WirelessHART-Netzwerk, ein Flag im WirelessHART-Header des Netzwerkrouting-Protokolls umfassen, der ein IPv6-Datenframe-Fragment einkapselt, das darauf hinweist, dass der Datenframe mit einem zugrunde liegenden IPv6-Datenpaket oder Datenframe verbunden ist. Der erweiterte WirelessHART-Netzwerkblock **326** kann feststellen, ob der empfangene Datenframe ein IPv6-Datenframe-Fragment ist, basierend auf der Erkennung des Flags oder eines Hinweises im Protokoll-Header des WirelessHART-Netzwerklayers, oder ob stattdessen der empfangene Datenframe ein standardmäßiges WirelessHART-Datenpaket ist. Beim Erkennen des Flags, das anzeigt, dass das Paket ein IPv6-Paket oder -Fragment enthält, kann der erweiterte WirelessHART-Netzwerkblock **326** das IPv6-Datenframe-Fragment in eine Warteschlange kopieren und dann einen Hinweis auf einen 6LoWPAN-Block **328** in dem zweiten Set der oberen Layer **325** bereitstellen, dass ein oder mehrere IPv6-Datenframe-Fragmente in die Warteschlange kopiert werden.

[0125] Daneben kann während des Betriebs der erweiterte WirelessHART-Netzwerkblock **326** auch IPv6-Datenframe-Fragmente vom 6LoWPAN-Block **328** empfangen, um sie unter Verwendung des WirelessHART-Netzwerkrouting-Protokolls über das WirelessHART-Netzwerk zu senden. Dazu kapselt der WirelessHART-

Netzwerkblock **326** die empfangenen IPv6-Datenframes in einen WirelessHART-Header des Netzwerkrouting-Protokolls ein, wobei alle IP-Informationen des Netzwerkrouting-Protokolls in diesen Paketen erhalten bleiben. Jedoch setzt der erweiterte WirelessHART-Netzwerkblock **326** auch das Flag oder einen anderen Hinweis im Header des WirelessHART-Netzwerkrouting-Protokoll, das bzw. der angibt, dass das WirelessHART-Datenpaket (oder der Header des Netzwerkrouting-Protokolls) ein IPv6-Datenframe-Fragment einkapselt.

[0126] Wenn ein Set von Paketen an das zweite Set der oberen Layer des Stacks **325** geliefert wird, der mit IP-Datenpaketen verbunden ist, funktioniert der 6LoWPAN-Block **328** so, dass er den oder die von dem erweiterten WirelessHART-Netzwerkblock **326** empfangenen IPv6-Datenframes zusammenfügt, um einen IPv6-Datenframe zu erzeugen. Der 6LoWPAN-Block **328** kann einen Hinweis an einen Block des IPv6-Netzwerkrouting-Layers und an den Block des Transportlayer-Protokolls **330** übertragen, wenn der 6LoWPAN-Block **328** einen IPv6-Datenframe erzeugt. Der 6LoWPAN-Block **328** kann auch IPv6-Datenframes von dem Block des IPv6-Netzwerkrouting-Layers und dem Block des Transportprotokolls **330** empfangen, und in diesem Fall funktioniert der 6LoWPAN-Block **328** so, dass er die IPv6-Datenframes unter Verwendung der 6LoWPAN-Fragmentierungstechnik in einen oder mehrere Datenframes fragmentiert. Auf diese Weise überträgt der 6LoWPAN-Block **328** einen Hinweis an den erweiterten WirelessHART-Netzwerkblock **326** und/oder er liefert dem Netzwerkblock **326** die erzeugten Datenframe-Fragmente für das Senden über das WirelessHART-Kommunikationsnetzwerk unter Verwendung des WirelessHART-Netzwerkrouting-Protokolls, sowie einen Hinweis des WirelessHART-Geräts, an das die Pakete gesendet werden sollen. In vielen Fällen gilt dieser Hinweis einem Gateway-Gerät im WirelessHART-Netzwerk. In jedem Fall werden Struktur- und Funktionselemente des 6LoWPAN-Blocks **328** ausführlicher unter Bezugnahme auf **Fig.** 9 erörtert.

[0127] Außerdem implementiert der IPv6-Block des Netzwerkrouting-Layers und der Block des Transportprotokolls **330** innerhalb des zweiten Sets der oberen Layer des Stacks **325** ein Stack-Element des Netzwerkrouting-Layers und ein Stack-Element des Transportlayers, das IP-basierte Aktivitäten in Bezug auf Netzwerkrouting-Protokolle und die Verarbeitung von Transportlayer-Protokollen durchführt und UDP, TCP- bzw. einen ICMP-Transportblock umfasst, die entweder vom 6LoWPAN-Block **328** bereitgestellte IPv6-Datenframes decodiert oder IPv6-Datenframes codiert, die von einem Anwendungsblock **334** über eine Schnittstelle zur Anwendungsprogrammierung (API) **332** bereitgestellt werden. Beispiele für eine Schnittstelle zur Anwendungsprogrammierung umfassen eine Sockets-Schnittstelle. Natürlich führt der Block **330** auch Aktivitäten in Bezug auf Netzwerkrouting-Layer aus, z. B. die Codierung und Decodierung von Netzwerkrouting-Informationen in den IPv6-Datenpaketen. Unabhängig davon können Anwendungen, die im Anwendungsblock **334** ausgeführt werden, über die Schnittstelle zur Anwendungsprogrammierung **332** auf feldgerätspezifische Informationen **336** zugreifen, diese steuern und konfigurieren. Anwendungen, die im Anwendungsblock **334** ausgeführt werden (der ein IP- oder socketbasiertes Stack-Element des Anwendungslayers ist oder ein solches implementiert), können dem Constrained Application Protocol (CoAP) entsprechen. Einzelheiten zum CoAP-Protokoll gibt es unter der Web-URL tools.ietf.org/html/draft-ietf-core-coap-03.

[0128] Beispielsweise kann eine Anwendung, die im Anwendungsblock **334** ausgeführt wird, über ein WirelessHART-Kommunikationsnetzwerk **100** eine Anforderung von Feldgerät-Statusinformationen von einem Remote-Computer empfangen, beispielsweise dem Gerät **116** aus **Fig.** 6. Die Anwendung kann die Feldgerät-Statusinformationen über die Schnittstelle zur Anwendungsprogrammierung **332** abrufen und die Statusinformationen als einen oder mehrere IPv6-Datenframes über den IPv6-Transportprotokollblock **330**, den 6LoWPAN-Block **328** und das gemeinsame Set der unteren Layern **321** der Stacks an den Computer **116** übertragen. Wie bereits in Bezug auf **Fig.** 6 beschrieben, können IPv6-Datenframes über das WirelessHART-Kommunikationsnetzwerk **100** an den Computer **116** übertragen werden.

[0129] Wie zuvor erörtert, kann der erweiterte WirelessHART-Netzwerkblock **326** ein von einem WirelessHART MAC-Block **324** empfangenes Datenpaket verarbeiten und feststellen, ob der von einem WirelessHART-Header des Netzwerkrouting-Protokolls eingekapselte Datenframe einem IPv6-Datenframe-Fragment entspricht, indem das Vorhandensein eines Flags in den Netzwerkrouting-Informationen dieses Datenpakets erkannt wird. In Fällen jedoch, in denen der WirelessHART-Protokoll-Header, der einen empfangenen Datenframe einkapselt, nicht das Flag umfasst (oder wenn das Flag nicht gesetzt ist, um auf ein zugrunde liegendes IPv6-Datenpaket hinzuweisen), überträgt der erweiterte WirelessHART-Netzwerkblock **326** einen Hinweis über das Vorhandensein eines Standard-WirelessHART-Datenpakets an einen WirelessHART-Transportblock **338** im ersten Set der oberen Layer des Stacks **323**. Der WirelessHART-Transportblock **338**, der ein Stack-Element des WirelessHART-Transportlayers implementiert, kann dann auf dieses Datenpaket zugreifen, das Datenpaket mittels der Standardverarbeitung für WirelessHART-Transportlayer verarbeiten und nach der Verarbeitung der empfangenen Datenframes einen Hinweis an einen HART-Anwendungsblock **340** übertragen (der ein Stack-Element eines WirelessHART-Anwendungslayers ist oder ein solches implementiert). Der An-

wendungsblock **340** implementiert dann WirelessHART-konforme Anwendungen, um das Datenpaket auf eine beliebige bekannte oder gewünschte Weise zu verarbeiten.

[0130] Wie man versteht, kann der erweiterte WirelessHART-Netzwerkblock **326** auch WirelessHART-Datenframes empfangen, um sie über das WirelessHART-Netzwerk vom WirelessHART-Transportblock **338** zu übertragen. Wie bereits ausgeführt, kann der WirelessHART-Netzwerkblock **326** so agieren, dass er empfangene WirelessHART-Datenframes in den Header einen WirelessHART-Netzwerkprotokolls einkapselt. Allerdings markiert der erweiterte WirelessHART-Netzwerkblock **326** die vom WirelessHART-Transportblock **338** empfangenen Datenframes in diesem Fall nicht als IPv6-Datenframe-Fragmente, da die Datenframes vom WirelessHART-Transportblock **338** empfangen wurden und damit standardmäßige WirelessHART-Datenpakete sind. Als Ergebnis dieses Vorgangs verarbeitet der WirelessHART-Netzwerkblock **326** die WirelessHART-Pakete nach bekannten WirelessHART-Netzwerkroutingprotokollverfahren, um die WirelessHART-Pakete über das WirelessHART-Kommunikationsnetzwerk zu senden.

[0131] Wie zuvor erwähnt, können Anwendungen im Anwendungslayer **314** aus **Fig.** 7 und Anwendungsblock **334** aus **Fig.** 8. angepasst werden, um unter Verwendung einer API zu funktionieren. Programmierinformationen, einschließlich des Aufrufs von Funktionen in der API, können durch Feldgerätehersteller in Form eines Software Development Kits (SDK) zur Verfügung gestellt werden. Im Betrieb können unabhängige Software-Entwickler das SDK über ein greifbares, nicht-vergängliches Medium erhalten, z. B. als CD-ROM. Entwickler können unter Verwendung des SDK benutzerdefinierte Anwendungen auf Allzweck-Computern erstellen. Solche benutzerdefinierten Anwendungen können über das Internet an das WirelessHART-Netzwerk ”gepusht” werden, wenn dies gewünscht ist. Alternativ können Anbieter solche kundenspezifischen Anwendungen in einem Online-Application Store ”hosten”. Anlagenbetreiber können kundenspezifische Anwendungen selektiv aus dem Online-Application Store über das Internet und das WirelessHART-Netzwerk an ein Feldgerät ”pullen”.

[0132] **Fig.** 9 ist ein Blockdiagramm eines Beispiels eines 6LoWPAN-Blocks **350**, der IPv6-Datenframes von empfangenen 6LoWPAN-Datenframes erzeugen kann sowie 6LoWPAN-Datenframes von empfangenen IPv6-Datenframes. Der 6LoWPAN Block-**350** kann dem 6LoWPAN-Anpassungslayer **310** aus **Fig.** 7 entsprechen, wobei der 6LoWPAN-Block **350** über eine Schnittstelle **360** mit einem Datenframe-Routingblock **308** (**Fig.** 7) kommunizieren kann. Der 6LoWPAN-Block **350** kann auch über ein Interface **361** mit dem WirelessHART-Netzwerklayer **315** (**Fig.** 7) kommunizieren. In einem anderen Beispiel kann ein 6LoWPAN-Block **350** dem 6LoWPAN-Block **328** aus **Fig.** 8 entsprechen. In diesem Beispiel kann der 6LoWPAN-Block **350** IPv6-Datenframes vom IPv6-Transportprotokollblock **330** aus **Fig.** 8 empfangen und an ihn übertragen, und der 6LoWPAN-Block **350** kann IPv6-Datenframe-Fragmente vom erweiterten WirelessHART-Netzwerkblock **326** aus **Fig.** 8 empfangen und an ihn senden.

[0133] Wie in **Fig.** 9 dargestellt, kann der 6LoWPAN-Block **350** ein Header-Komprimierungsmodul **351** enthalten, das kommunikativ mit einem Fragmentierungsmodul **353** gekoppelt ist, und ein Zusammenbaumodul **354**, das kommunikativ mit einem Header-Dekomprimierungsmodul **352** gekoppelt ist. Hier kann das Header-Komprimierungsmodul **351** IPv6-Datenframes über die Schnittstelle **360-1** empfangen, und es kann bei einem empfangenen IPv6-Datenframe so agieren, dass es einen 6LoWPAN-konformen komprimierten IPv6-Header erzeugt. Das an das Header-Komprimierungsmodul **351** gekoppelte Fragmentierungsmodul **353** empfängt den IPv6-Datenframe einschließlich des komprimierten IPv6-Headers vom Header-Komprimierungsmodul **351**, und agiert so, dass es den IPv6-Datenframe in mehrere 6LoWPAN-konforme Datenframes ”fragmentiert”. Das Fragmentierungsmodul **353** kann einen ”Fragment-Header” an jeden der verschiedenen 6LoWPAN-Datenframes anhängen. Ein Beispiel eines Fragmentierungsmoduls **353** wird nachstehend mit Bezug auf **Fig.** 10A beschrieben. Der Fragment-Header kann Informationen enthalten, die das Zusammenbauen der verschiedenen 6LoWPAN-Datenframes ermöglicht, um den ursprünglichen IPv6-Datenframe zu erhalten. Selbstverständlich kann das Zusammenbauen in einem Gerät erfolgen, das der vorgesehene Empfänger der IPv6-Datenframes ist, oder in einem zwischengeschalteten Gerät. Informationen in Bezug auf die Erzeugung eines komprimierten IPv6-Headers und die Fragmentierung eines IPv6-Datenframes nach dem 6LoWPAN-Protokoll wird insbesondere in RFC 4944, ”Transmission of IPv6 Packets over IEEE 802.15.4 Networks” (Übertragung von IPv6-Paketen über IEEE 802.15.4-Netzwerke) beschrieben.

[0134] Das Fragmentierungsmodul **353** aus **Fig.** 9. kann die von einem IPv6-Datenframe erzeugten verschiedenen 6LoWPAN-konformen Datenframes an einen WirelessHART-Netzwerklayer **315** übertragen, zum Beispiel über eine Schnittstelle **361-1**. Als Teil dieses Prozesses kann das Fragmentierungsmodul **353** dem WirelessHART-Netzwerklayer **315** die IPv6-Adresse des vorgesehenen Empfängers der verschiedenen 6LoWPAN-konformen Datenframes bereitstellen. Wie bereits beschrieben, kann der WirelessHART-Netzwerklayer **315** die IPv6-Adresse verwenden, um in einer WirelessHART-Routing-Tabelle **316** (**Fig.** 7) WirelessHART-Routing-

Informationen für ein IPv6-fähiges WirelessHART-Gerät ”nachzuschlagen”, das der vorgesehene Empfänger der verschiedenen 6LoWPAN-Datenframes ist. Natürlich können solche Nachschlagtabellen in jedem der anderen hier beschriebenen IP-fähigen WirelessHART-Geräte vorhanden sein.

[0135] Separat kann der 6LoWPAN-Block **350** aus **Fig.** 9 6LoWPAN-Datenframe-Fragmente vom WirelessHART-Netzwerklayer **315** empfangen, zum Beispiel über die Schnittstelle **361-2**. Die 6LoWPAN-Datenframes selbst können durch ein IPv6-fähiges WirelessHART-Gerät aus einem IPv6-Frame erzeugt werden. Ein Zusammenbaumodul **354** kann dann aus den empfangenen 6LoWPAN-Datenframes einen IPv6-Datenframe erzeugen. Jedes der empfangenen 6LoWPAN-Datenframe-Fragmente enthält typischerweise einen Fragment-Header, und das Zusammenbaumodul **354** nutzt die Fragment-Header, um einen IPv6-Datenframe in jeder bekannten Weise wieder zusammenzubauen. Ein Beispiel eines Zusammenbaumoduls **354** wird nachstehend mit Bezug auf **Fig.** l OB beschrieben. Das Zusammenbaumodul **354** kann einen zusammengesetzten IPv6-Datenframe an das Header-Dekomprimierungsmodul **352** übertragen, das den komprimierten IPv6-Header ”dekomprimiert”, um den ursprünglichen IPv6-Header zu erzeugen (mit den IP-Informationen des Netzwerkrouting-Protokolls, beispielsweise die IP-Adressen des Datenpakets). Das Header-Dekomprimierungsmodul **352** überträgt dann den IPv6-Datenframe zum Beispiel über eine Schnittstelle **360-2** an einen Datenframe-Routingblock **308** aus **Fig.** 7.

[0136] **Fig.** 10A zeigt ein Beispiel eines Fragmentierungsmoduls **500**, das in den 6LoWPAN-Block **350** implementiert werden kann. In diesem Fall kann eine Warteschlange **501** einen oder mehrere IPv6-Datenframes vom Header-Komprimierungsmodul **351** empfangen, und ein Fragmentierer **502** kann einen Hinweis darauf empfangen, wann ein oder mehrere IPv6-Datenframes in der Warteschlange **501** empfangen werden. Der Fragmentierer **502** kann einen oder mehrere IPv6-Datenframes aus einer Warteschlange **501** mithilfe von standardmäßigen 6LoWPAN-Fragmentierungstechniken fragmentieren, und er kann die Fragmente in einem Sendepuffer **503** speichern. Hier entspricht jedes Set aus Fragmenten **503-1**, **503-2** und **503-3** den verschiedenen Fragmenten von einem 6LoWPAN-Datenframe, und die Größe (Länge) dieser Fragmente kann festgelegt werden, um sicherzustellen, dass jedes Fragment innerhalb der maximal zulässigen Größe eines Datenpakets innerhalb des WirelessHART-Protokolls liegt, nachdem ihm weitere erwartete Informationen in Bezug auf WirelessHART-Netzwerkrouting, Datenlink-Layer und physische Layer hinzugefügt wurden. Wie zuvor erörtert, kann der Fragmentierer **502** an jeden der mehreren 6LoWPAN-Datenframes **503-1**, **503-2**, **503-3**, usw. einen Fragment-Header anhängen. Der Fragment-Header kann an jeden der 6LoWPAN-Datenframes **503-1** bis **503-3** angehängt werden, bevor die 6LoWPAN-Datenframes **503-1** bis **503-3** im Sendepuffer **503** gespeichert werden, falls gewünscht. Obwohl der Sendepuffer **503** als ein einzelner zusammenhängender Puffer dargestellt wird, kann in anderen Ausführungsformen ein Sendepuffer **503** mehrere nicht zusammenhängende Puffer umfassen.

[0137] Der Fragmentierer **502** kann auch eine Sendepuffer-Karte **504** mit mehreren Zeigern **504-1** bis **504-3** umfassen, von denen jeder auf eine Position im Sendepuffer **503** verweist. Jeder dieser Zeiger kann den Beginn eines 6LoWPAN-Datenframes anzeigen. Falls gewünscht, kann jeder Zeiger auch mit einem Statusblock verbunden werden, der einen Status des entsprechenden 6LoWPAN-Datenframes angibt. Der Fragmentierer **502** kann, basierend auf der Sendepuffer-Karte **504**, auf den Sendepuffer **503** zugreifen, und er kann jeden der 6LoWPAN-Datenframes **503-1** bis **503-2** an die ausgehende Warteschlange **505** übertragen. In diesem Fall kann der Fragmentierer **502** einen Hinweis auf einen WirelessHART-Netzwerklayer **315** übertragen, nachdem jeder der 6LoWPAN-Datenframes **503-1** bis **503-2** an die ausgehende Warteschlange **505** weitergeleitet wurde, und, wie zuvor erörtert, kann der Fragmentierer **502** dem WirelessHART-Netzwerklayer **315** die IP-Adresse des vorgesehenen Empfängers der 6LoWPAN-Datenframes **503-1** bis **503-3** bereitstellen usw.

[0138] **Fig.** 10B zeigt ein Beispiel eines Zusammenbaumoduls **520**, das beispielsweise in das 6LoWPAN-Anpassungsmodul **350** implementiert werden kann. Hier enthält ein Zusammenbaumodul **520** einen Reassembler **522**, der einen Hinweis von einem WirelessHART-Netzwerklayer (z. B. Netzwerklayer **315**) empfängt, wenn einer oder mehrere 6LoWPAN-Datenframes in einer eingehenden Warteschlange **521** empfangen werden. Der Reassembler **522** kann jeden der 6LoWPAN-Datenframes von der eingehenden Warteschlange **521** an einen Empfangspuffer **523** übertragen. Der Reassembler **522** kann dann die Fragment-Header verwenden, die bei der Erzeugung an jeden der 6LoWPAN-Datenframes angehängt wurden, um einen IPv6-Datenframe unter Verwendung von standardmäßigen oder bekannten 6LoWPAN-Zusammenbautechniken zu generieren. Der Reassembler **522** kann eine Empfängerpuffer-Karte **524** verwenden, um den Empfang von jedem der 6LoWPAN-Datenframes zu verfolgen, aus denen sich ein bestimmter IPv6-Datenframe zusammensetzt. In einem Szenario kann der WirelessHART-Netzwerklayer **315** auch außer Betrieb 6LoWPAN-Datenframes empfangen, und der Reassembler **522** kann dennoch den IPv6-Datenframe zusammenbauen, wenn alle Datenpakete, ein-

schließlich der Datenframe-Fragmente, empfangen und in der eingehenden Warteschlange **521** gespeichert wurden.

[0139] Fig. 11A zeigt eine Darstellung eines Beispiels eines Datenframes **550**, der über das Internet als ein IP-Datenframe empfangen und über ein zweites Netzwerk-Routingprotokoll, beispielsweise ein WirelessHART-Netzwerkrouting-Protokoll, unter Verwendung von Low-Power-Fragmentierungstechniken übertragen werden kann. Zum Zweck der Darstellung kann der Datenframe **550** ein IPv6-Datenframe sein, und es wird angenommen, dass er von einem IPv6-fähigen WirelessHART-Gateway über das Internet empfangen wurde. Wie in **Fig.** 11 A dargestellt, umfasst der Datenframe **550** Anwendungs-Payload-Daten **550-1**, die von einem Remote-Computer erzeugt sein können, sowie die daran angehängten verschiedenen anderen Header und Trailer gemäß der normalen Erzeugung und dem Routing von IP-Datenpaketen.

[0140] Es kann ein geeignetes Protokoll verwendet werden, um die IPv6-Datenframes **550** in ein oder mehrere Datenframe-Fragmente **552-1** bis **552-n** zu fragmentieren. In diesem Beispiel kann der IPv6-Datenframe **550** unter Verwendung von Verfahren fragmentiert werden, die dem 6LoWPAN-Protokoll entsprechen, aber alternativ können auch andere Fragmentierungstechniken verwendet werden. Jedes Datenframe-Fragment kann in Informationen des 6LoWPAN-Fragment-Headers **553** eingekapselt werden, so dass das Datenframe-Fragment **552-1** so dargestellt wird, dass Informationen des Fragment-Headers **553-1** eingekapselt sind. Die Header-Informationen des Datenframe-Fragments werden so dargestellt, dass sie einen Fragment-Header, einen komprimierten IP-Protokoll-Header und einen komprimierten Transportprotokoll-Header (für ein IP-Transportprotokoll) umfassen, während die Payload eines jeden Datenframe-Fragments ein Abschnitt der Anwendungs-Payload **550-1** des ursprünglichen IPv6-Datenpakets ist.

[0141] Wie man versteht, ist jedes Datenframe-Fragment, beispielsweise das Datenframe-Fragment **552-1**, in einen geeigneten zweiten Header des Netzwerkrouting-Protokolls eingekapselt, um das Routing des Datenframe-Fragments **552-1** innerhalb eines Low-Power-Netzwerks zu ermöglichen, z. B. dem WirelessHART-Netzwerk. In diesem Beispiel kann der Datenframe **552-1** in einen WirelessHART-Protokoll-Header eingekapselt sein, um ein in WirelessHART eingekapseltes IPv6-Datenframe-Fragment **554** zu erzeugen. Der WirelessHART-Protokoll-Header besteht aus einem WirelessHART-Netzwerkrouting-Header **554-1** und einem Header des WirelessHART-Datenlinklayers **554-2**. Das in WirelessHART eingekapselte IPv6-Datenframe-Fragment **554** kann vor der Übertragung über das WirelessHART-Kommunikationsnetzwerk weiterhin in einen physischen Header **554-3** und einen physischen Trailer **554-4** eingekapselt sein, die dem Low-Power-Kommunikationsstandard, beispielsweise dem Standard 802.15.4, entsprechen.

[0142] Ein oder mehrere Bits in den Datenfeldern, die dem WirelessHART-Protokoll entsprechen, können aufgeteilt werden, um einem Empfangsgerät anzuzeigen, dass das Datenframe-Fragment einem IPv6-Datenframe entspricht. **Fig.** 11B zeigt ein DLPDU-Feld **560** des HART-Netzwerkrouting-Headers **554-1**. In diesem Beispiel wird Bit 7 (durch Bezugszeichen **561** angegeben) als ein Flag verwendet, um einem Empfangsgerät anzuzeigen, dass der WirelessHART-Header ein IPv6-Datenframe-Fragment einkapselt. Bit 7 (**561**) kann auf eine '1' gesetzt werden, bevor die WirelessHART-Datenframes übertragen werden, um die Existenz eines IPv6-Datenframe-Fragments im WirelessHART-Datenframe anzuzeigen. In Bezug auf **Fig.** 7 und **Fig.** 8 können der WirelessHART-Netzwerklayer **315** und der erweiterte WirelessHART-Netzwerkblock **326** Bit 7 (**561**) des DLPDU-Felds **560** eines WirelessHART-Netzwerkroutingprotokoll-Headers auf eine logische '1' setzen, bevor ein Datenframe übertragen wird.

[0143] Fig. 11C entspricht dem Steuerbyte **565** eines WirelessHART-Netzwerkrouting-Headers **554-1**, und die Datenfelder im Steuerbyte **565** können auch oder stattdessen aufgeteilt werden, um anzuzeigen, wenn der WirelessHART-Protokoll-Header ein Datenframe-Fragment über einen verschlüsselten Kommunikationskanal empfängt. Zum Beispiel kann Bit 4 (durch Bezugszeichen **566** angegeben) verwendet werden, um anzuzeigen, wenn das Datenframe-Fragment über verschlüsselte WirelessHART-Kommunikationskanäle empfangen wird. In diesem Beispiel wird das Bit 5 (**567**) verwendet, um die Position eines Sicherheitsheaders innerhalb eines Datenframe-Fragments anzuzeigen.

[0144] Die nachstehende Tabelle 1 zeigt eine Reihe von möglichen Kombinationen von Bit 4 (**566**) und Bit 5 (**567**) des Steuerbytes, die verwendet werden können, um innerhalb der WirelessHART-Netzwerkrouting-Informationen Verschlüsselung und Sicherheit bereitzustellen.

Bit 4 (**566**)	Bit 5 (**567**)	Interpretation
0	Nicht zutreffend	Keine WirelessHART-Sicherheit
1	1	WirelessHART-Sicherheitslayer vor Fragment-Header
1	0	WirelessHART-Sicherheitslayer nach Fragment-Header

Tabelle 1

[0145] Die Einstellungen von Bit 4 und Bit 5 können in einem IPv6-fähigen WirelessHART-Gateway sowie in Feldgeräten innerhalb eines WirelessHART-Netzwerks konfiguriert werden. Bezugnehmend auf Tabelle 1, können Feldgeräte, die innerhalb WirelessHART-Kommunikationsnetzwerken arbeiten, in Fällen, in denen das Bit 4 als eine logische '0' konfiguriert ist, sichere IP-Kommunikation verwenden, um Datenframes zu empfangen und zu senden. Alternativ können die Feldgeräte über unsichere Kommunikationskanäle kommunizieren. In Fällen, in denen sichere IP-Kommunikationskanäle genutzt werden, können die Anwendungen für die Durchsetzung sicherer Kommunikation verantwortlich sein.

[0146] In Fällen, in denen Bit 4 (**566**) als eine logische '1' konfiguriert ist und Bit 5 (**567**) als eine logische '1', kann ein IPv6-fähiges WirelessHART-Gateway im WirelessHART-Netzwerklayer Verfahren implementieren, um nach dem WirelessHART-Netzwerk-Header **554-1** einen Sicherheitsheader einzufügen. In diesen Fällen kann ein IPv6-fähiges WirelessHART IPv6-Datenframes aus dem Internet über sichere IP-Kommunikationskanäle empfangen, aber die IPv6-Datenframe-Fragmente können entschlüsselt werden, bevor sie in einem IPv6-fähigen WirelessHART-Gateway fragmentiert werden.

[0147] In Fällen, in denen Bit 4 (**566**) als eine logische '1' konfiguriert ist und Bit 5 (**567**) als eine logische '0', kann ein IPv6-fähiges WirelessHART-Gateway im WirelessHART-Netzwerklayer Verfahren implementieren, um nach dem Fragment-Header DES 6LoWPAN-Datenframes **554-5** einen Sicherheitsheader einzufügen. Ein Empfangsfeldgerät würde Bit 4 und Bit 5 so interpretieren, dass die entsprechenden Sicherheitsmechanismen beim Empfangen und Senden von IPv6-Datenframe-Fragmenten durchgesetzt werden.

[0148] Wie bereits in Bezug auf **Fig.** 8 erörtert, kann der erweiterte WirelessHART-Netzwerkblock **326** bestimmen, dass Bit 7 (**561**) eines empfangenen Datenframes nicht gesetzt ist (0). In diesem Fall überträgt der erweiterte WirelessHART-Netzwerkblock **326** eine Anzeige an den WirelessHART-Transportblock **338** des eingehenden Datenframes, um die Verarbeitung unter Verwendung von standardmäßiger WirelessHART-Kommunikation durchzuführen. Der Datenframe **554** kann ein Felde für den WirelessHART-Protokollindikator enthalten, das Informationen beinhalten kann, die vom WirelessHART-Transportblock **338** und/oder dem HART-Anwendungsblock **340** verwendet werden können, um zum Beispiel zu bestimmen, welchem WirelessHART-Protokoll oder welcher WirelessHART-Anwendung der Anwendungs-Payload des Datenframes **554** entspricht.

[0149] Erneut bezugnehmend auf das Beispielnetzwerk aus **Fig.** 6 kann ein IPv6-fähiges Feldgerät **108** in das WirelessHART-Protokoll eingekapselte IPv6-Datenframes über das WirelessHART-Kommunikationsnetzwerk **100** empfangen, wobei die vorgesehenen Empfänger der empfangenen IPv6-Datenframe-Fragmente andere IPv6-fähige Feldgeräte sind. Das IPv6-fähige Feldgerät **108** kann im WirelessHART-Protokoll eingekapselte Nicht-IPv6-Datenframes über das WirelessHART-Kommunikationsnetzwerk **100** empfangen, wobei die vorgesehenen Empfänger der IPv6-Datenframe-Fragmente andere IPv6-fähige Feldgeräte sind. Mehrere Verfahren, die in WirelessHART-Feldgeräte implementiert werden können, um IPv6-Datenframe-Fragmente innerhalb eines WirelessHART-Kommunikationsnetzwerks zu routen, werden nun im Detail erörtert. Wie man versteht, können IPv6-fähige WirelessHART-Feldgeräte und in einigen Fällen nicht-IPv6-fähige WirelessHART-Feldgeräte so konfiguriert werden, dass sie eins der verschiedenen nachstehend erörterten Verfahren nutzen, um IPv6-Datenframe-Fragmente und standardmäßige WirelessHART-Datenframes innerhalb eines WirelessHART-Kommunikationsnetzwerks zu routen. Obwohl in einigen Fällen die nachstehend erörterten Verfahren ein einzelnes zwischengeschaltetes IPv6-fähiges WirelessHART-Gerät oder ein einzelnes zwischengeschaltetes nicht-IPv6-fähiges WirelessHART-Gerät benutzen, um IPv6-Datenframe-Fragmente zwischen zwei anderen IPv6-fähigen WirelessHART-Feldgeräten zu routen, können mehrere zwischengeschaltete IPv6-fähige WirelessHART-Geräte oder nicht-IPv6-fähige WirelessHART-Geräte als zwischengeschaltete Geräte nach den unten erörterten Verfahren benutzt werden, um ”Multi-Hop”-Routing von IPv6-Datenframes in einem WirelessHART-Kommunikationsnetzwerk zu ermöglichen.

[0150] **Fig.** 12 zeigt eine Darstellung eines Beispiels eines Routing-Verfahrens, das in IPv6-fähige WirelessHART-Feldgeräte innerhalb eines exemplarischen WirelessHART-Kommunikationsnetzwerks **600** implementiert werden kann. Stacks für die Netzwerkkommunikation **601**, **602** und **603** der verschiedenen IPv6-fähigen WirelessHART-Feldgeräte im Netzwerk **600** werden jeweils in separate IPv6-fähige WirelessHART-Feldgeräte implementiert. Wie man versteht, können die Stacks **601**, **602** und **603** gleich oder ähnlich sein, und sie können im Allgemeinen in Übereinstimmung mit der Beschreibung des Stacks **320** aus **Fig.** 8 arbeiten. In diesem Fall wird der Stack für die Netzwerkkommunikation **602** in einem zwischengeschalteten WirelessHART-Feldgerät implementiert und funktioniert so, dass sowohl WirelessHART-Datenframes als auch IPv6-Datenframe-Fragmente, die in WirelessHART-Datenframes eingekapselt sind, zwischen den Feldgeräten geroutet werden, welche die Stacks für die Netzwerkkommunikation **601** und **603** implementieren. Wie man versteht, werden die IPv6-Datenframe-Fragmente über das Kommunikationsnetz **600** auf Basis von Informationen in einem WirelessHART-Protokoll-Header geroutet (einschließlich Netzwerkrouting-Informationen, Datenlinklayer-Informationen und Protokollinformationen des physischen Layers), die jeweils eins der IPv6-Datenframe-Fragmente einkapseln.

[0151] In einem Beispiel-Szenario erzeugt eine Anwendung, die in einem HART-Anwendungsblock **601-1** innerhalb des Stacks **601** ausgeführt wird, einen WirelessHART-Datenframe für eine Anwendung, die im HART-Anwendungsblock **603-1** des Gerätes ausgeführt wird, das den Stack **603** ausführt. Der Anwendungsblock **601-1** stellt diesen Datenframe einem WirelessHART-Transportblock **601-3** für die Lieferung an den erweiterten WirelessHART-Netzwerkblock **601-2** bereit, der das Datenpaket durch Hinzufügen von Informationen für das WirelessHART-Netzwerkrouting zu dem Datenpaket verarbeitet und dieses Datenpaket an den WirelessHART MAC(Datenlink)-Layer **601-6** und an den physischen Layer **601-7** für die Verarbeitung und das Routing über das Netzwerk **600** sendet. In einem anderen Beispiel-Szenario erzeugt eine Anwendung, die in einem Anwendungsblock **601-4** ausgeführt wird, einen IPv6-Datenframe für eine IPv6-Anwendung, die in einem Anwendungsblock **603-4** des Geräts ausgeführt wird, das den Stack **603** implementiert. Wie zuvor mit Bezug auf **Fig.** 8 beschrieben, fragmentiert ein 6LoWPAN-Block **601-5** im IP-Transportlayer des Stacks **601** die IPv6-Datenframes, die von der Anwendung empfangen wurden, die im Anwendungsblock **601-4** ausgeführt wird. In diesem Fall kapselt der erweiterte WirelessHART-Netzwerkblock **601-2** jedes der IPv6-Datenframe-Fragmente (vom Anwendungsblock **601-4** und vom 6LoWPAN-Block **601-5**) wie oben beschrieben mithilfe der unteren Layer der Stacks **601-2**, **601-6** und **601-7** in einen WirelessHART-Protokoll-Header ein. Natürlich aktualisiert der erweiterte WirelessHART-Netzwerkblock **601-2** den WirelessHART-Protokoll-Header entsprechend jedem Datenframe mit Routing-Informationen, die der logischen Adresse oder der Route zu dem WirelessHART-Feldgerät entsprechend, das den Stack für die Netzwerkkommunikation **603** implementiert, bevor die Übertragung des Datenframes durch das WirelessHART-Kommunikationsnetzwerk erfolgt. Diese Informationen können in Routing-Tabellen (nicht in **Fig.** 12 gezeigt) gespeichert werden, die IP-Adressen von Geräten innerhalb des WirelessHART-Netzwerks mit WirelessHART-Routing-Informationen verknüpfen, um diese Geräte zu erreichen. Der erweiterte WirelessHART-Netzwerkblock **601-2** codiert auch den WirelessHART-Protokoll-Header von Datenframes, die IPv6-Datenframe-Fragmenten mit einem Flag entsprechen, das angibt, dass der WirelessHART-Protokoll-Header ein IPv6-Datenframe-Fragment vor der Übertragung einkapselt.

[0152] Nachdem sie über das Netzwerk gesendet wurden, empfängt ein IPv6-fähiges Feldgerät, das den Kommunikationsstack **602** implementiert, die verschiedenen WirelessHART-codierten IPv6-Datenframe-Fragmente oder die WirelessHART-Pakete und verarbeitet diese Pakete im physischen Layer **602-7**, dem Datenlinklayer **602-6** und den Abschnitten des Netzwerklayers **602-2** von Stack **602**. In beiden Fällen geben die Informationen für das WirelessHART-Netzwerkrouting von jedem Datenpaket (entweder ein WirelessHART-Datenpakets oder das im Header des WirelessHART-Protokolls eingekapselte IPv6-Datenframe-Fragment) an, dass die Pakete an ein anderes Gerät im WirelessHART-Netzwerk **600** gesendet werden sollen. Die unteren Layer **602-7**, **602-6** und **602-2** des Kommunikationsstacks **602** verarbeiten das Paket mithilfe von standardmäßigem WirelessHART-Netzwerkrouting sowie Datenlink- und physischer Link-Adressierung, um die entsprechenden Informationen für WirelessHART-Netzwerkrouting, Datenlink- und physische Layer in dem Paket zu decodieren und neu zu codieren, bevor dieses Pakets an das nächste WirelessHART-Gerät gesendet wird, das in diesem Fall das Gerät ist, das den Stack **603** implementiert. Wie man versteht, da das Datenpaket (entweder ein WirelessHART-Datenpaket oder ein IPv6-Datenpaket eingekapselt in Informationen für das WirelessHART-Netzwerkrouting) nicht an eine Anwendung in dem Gerät adressiert ist, das den Stack **602** implementiert, muss jedes Datenpaket nur bis zum WirelessHART-Netzwerkblock **602-2** des Stacks **602** verarbeitet werden, um über dieses Gerät geroutet zu werden. Somit verwendet in diesem Fall der erweiterte WirelessHART-Netzwerkblock **602-2** die Informationen im WirelessHART-Protokoll-Header, der jedes empfangene Datenframe-Fragment im WirelessHART-Paket einkapselt, um das Netzwerk- und Datenlinklayer-Routing (mithilfe der Protokolle für das WirelessHART-Netzwerk und das Datenlink-Routing) durchzuführen, damit die empfangenen Datenframe-Fragmente oder das WirelessHART-Paket an ein IPv6-fähiges Feldgerät gesendet werden kön-

nen, das den Kommunikationsstack **603** implementiert. Der Pfad **610** entspricht einer Beispiel-Route, die sowohl IPv6- als auch Nicht-IPv6-Datenframes nehmen können, wobei die gestrichelten Linien die verschiedenen Pfade eines standardmäßigen WirelessHART-Pakets oder eines IPv6-codierten Pakets innerhalb der Stacks **601**, **602**, **603** anzeigen.

[0153] Natürlich werden die Nachrichten, wenn die Nachrichten oder Datenpakete in dem Gerät empfangen werden, das den Stack **603** implementiert, wie oben beschrieben mit Bezug auf **Fig.** 8 bis zum erweiterten WirelessHART-Netzwerkblock **603-2** verarbeitet. In diesem Fall, da das Paket letztendlich über die Netzwerkrouting-Informationen des Datenpakets an eine Anwendung in dem Gerät adressiert ist, das den Stack **603** implementiert, stellt der Netzwerkblock **603-2** das decodierte Paket entweder dem WirelessHART-Transportblock **603-3** (wenn das Paket ein WirelessHART-Paket oder -Datenframe ist) oder dem IP-Transportlayer **603-5** (wenn das Paket fragmentierte IPv6-Datenframes enthält) zur Decodierung und Verarbeitung bereit. Natürlich werden im letzteren Fall die IP-Netzwerkrouting-Informationen im Block **603-5** bewahrt, so dass die IP-Adresse der Empfänger-Anwendung verwendet und bestimmt werden kann, um dadurch die IPv6-Datenframes an das entsprechende Socket für die Lieferung an die richtige Anwendung im Anwendungslayer **603-4** des Stacks **603** bereitzustellen.

[0154] **Fig.** 13 zeigt ein anderes exemplarisches Routing-Verfahren, das in IPv6-fähige WirelessHART-Feldgeräte innerhalb des exemplarischen WirelessHART-Kommunikationsnetzwerks **600** implementiert werden kann. Dieses Kommunikationsverfahren ähnelt stark demjenigen, das in Bezug auf **Fig.** 12 beschrieben wurde, jedoch mit einem Unterschied. In diesem Verfahren routet das zwischengeschaltete IPv6-fähige WirelessHART-Feldgerät (Implementierung des Stacks **602**) Nicht-IPv6-Datenframe-Fragmente mithilfe der im WirelessHART-Protokoll-Header enthaltenen Informationen, setzt aber die IPv6-Datenframe-Fragmente neu zusammen und fragmentiert sie erneut, bevor es die neu fragmentierten IPv6-Datenframe-Fragmente über das WirelessHART-Netzwerk **600** zurückleitet. Diese Technik kann in IPv6-fähigen WirelessHART-Geräten wie Router oder Gateways erforderlich sein, die möglicherweise IP-Adressen benutzen müssen, um festzustellen, ob und wie der Datenframe oder eine Nachricht über das WirelessHART-Netzwerk geroutet wird, oder ob die IP-basierten Sicherheitstechniken unter Verwendung von Daten in den Informationen für das IP-Netzwerkrouting in einem IPv6-Datenpaket implementiert werden.

[0155] Wie in **Fig.** 13 durch die Routenlinie **660** dargestellt, erzeugt das IPv6-fähige WirelessHART-Gerät, das den Stack **601** implementiert, WirelessHART-Pakete und/oder IPv6-Datenpakete, die über das Netzwerk **600** gesendet werden, und diese Pakete werden wie oben beschrieben im Stack **601** verarbeitet, um im Kommunikationsnetzwerk **600** platziert zu werden. Das zwischengeschaltete IPv6-fähige WirelessHART-Feldgerät, das den Kommunikationsstack **602** implementiert, empfängt IPv6-Datenframe-Fragmente und Nicht-IPv6-Datenframes (z. B. WirelessHART-Datenframes) über Standardmäßige WirelessHART-Routing-Techniken. Die empfangenen Pakete werden wie oben beschrieben in den unteren Stacklayern **602-7**, **602-6** und **602-2** verarbeitet. Jedoch funktioniert nun der WirelessHART-Netzwerkblock **602-2** so, dass er die Anwesenheit eines Flags in dem WirelessHART-Protokoll-Header erkennt, das anzeigt, dass der Datenframe ein IPv6-Datenframe-Fragment ist und, wenn ein solches Flag entdeckt wird, routet der erweiterte WirelessHART-Netzwerkblock **602-2** die IPv6-Datenframe-Fragmente an den 6LoWPAN-Block **602-5**. Der erweiterte WirelessHART-Netzwerkblock **602-5** decodiert dann die IPv6-Netzwerklayer-Informationen von den Datenframe-Fragmenten zum Beispiel durch das Defragmentieren dieser Pakete und das Auslesen der IPv6-Netzwerklayer-Informationen aus diesen Paketen, um die IP-Adresse und Sicherheitsinformationen zu erhalten, die für das Standard-IP-Routing erforderlich sein können. Diese Aktion wird durch die gestrichelte Linie als Abschnitt des Pfades **660** durch den Block **602-5** angezeigt. In diesem Fall kann der 6LoWPAN-Block **602-5** die verschiedenen IPv6-Datenframe-Fragmente unter Verwendung der Information in den 6LoWPAN-Fragment-Headern zusammenfügen, um den IPv6-Datenframe zu regenerieren. Der 6LoWPAN-Block **602-5** decodiert und nutzt die Ziel-IP-Adressinformationen und/oder Sicherheitsinformationen im IP-Header des erzeugten IPv6-Datenframes und kann eine Sicherheitsverarbeitung ausführen. Darüber hinaus kann der Block **602-5** WirelessHART-Adressinformationen nachschlagen, die dem IPv6-fähigen WirelessHART-Feldgerät entsprechen, das den Kommunikationsstack **603** implementiert, d. h. dem Gerät, an das der IPv6-Datenframe durch die IPv6-Adressierung adressiert ist. Der 6LoWPAN-Block **652-2** kann dann den erzeugten IPv6-Datenframe in mehrere IPv6-Datenframe-Fragmente fragmentieren, bevor die IPv6-Datenfragmente zurück über den erweiterten WirelessHART-Netzwerkblock **602-2** hinaus übertragen werden, wobei das WirelessHART-Gerät über die Datenframes informiert wird, die über das WirelessHART-Kommunikationsnetzwerk **600** gesendet werden sollen. Der erweiterte WirelessHART-Netzwerkblock **602-2** fügt dann Informationen für das WirelessHART-Netzwerkrouting zu diesen Frames hinzu und sendet diese Frames zur Verarbeitung an die unteren Layer **602-6** und **602-7** des Stacks **602**, um sie über das WirelessHART-Kommunikationsnetzwerk **600** zu routen.

[0156] Im Empfängergerät, das den Stack **603** implementiert, werden die Datenpakete wie oben beschrieben verarbeitet und den entsprechenden Anwendungslayern oder Block **603-1** oder **603-4** basierend auf der Art des Datenframes bereitgestellt. Natürlich kann es, während nur ein zwischengeschaltetes Gerät, das den Stack **602** implementiert, in **Fig.** 13 dargestellt wird, mehr als ein zwischengeschaltetes Gerät im Pfad **660** geben, und jedes zwischengeschaltete Gerät kann die oben beschriebene Verarbeitung in Bezug auf Stack **602** ausführen.

[0157] **Fig.** 14 zeigt ein weiteres Routing- oder Kommunikationsverfahren, das genutzt werden könnte, um sowohl WirelessHART-Datenpakete als auch IPv6-Datenpakete über ein WirelessHART-Kommunikationsnetzwerk zu routen oder bereitzustellen. In dem Beispiel aus **Fig.** 14 erfolgt das Routing in der gleichen Weise wie oben in Bezug auf **Fig.** 12 beschrieben. Jedoch sind in diesem Fall möglicherweise eins oder mehrere der zwischengeschalteten Geräte ein nicht-IP-fähiges Feldgerät und schließen so die Netzwerk-, Transport- und Anwendungslayer eines IP-basierten Netzwerk-Stacks nicht ein. Wie in **Fig.** 14 dargestellt, können jedoch IP-basierte Datenframes und WirelessHART-Datenframes immer noch durch diese zwischengeschalteten Geräten geroutet werden, da jedes Datenpaket nur bis zu dem WirelessHART-Netzwerklayerblock **602-2** verarbeitet wird, bevor es neu codiert und an das nächste Gerät im Routing gesendet wird. Das heißt, in diesem Fall müssen zwischengeschaltete Geräte im WirelessHART-Netzwerk IPv6 nicht aktiviert haben, da das Routing durch diese Geräte nur unter Verwendung von Informationen für das WirelessHART-Netzwerkrouting-Protokoll erfolgen kann. Dadurch muss der Netzwerklayerblock **602-2** im Stack **602** kein ”erweiterter” WirelessHART-Netzwerklayer sein, weil dieser Block nicht das Vorhandensein von zugrunde liegenden IPv6-Datenpaketen erkennen oder Flags in die Informationen für den WirelessHART-Netzwerkrouting-Layer codieren muss, die das Vorhandensein dieser Art von Datenpaketen anzeigen.

[0158] Wie bereits erörtert, beispielsweise in Bezug auf **Fig.** 7 und **Fig.** 8, können der Anwendungslayer **314** und der Anwendungsblock **334** kundenspezifische Anwendungen beinhalten, die das Framework und die Schnittstelle für die Anwendungsprogrammierung **315** und eine Schnittstelle zur Anwendungsprogrammierung **332** verwenden, um erzeugte IPv6-Datenframes über ein WirelessHART-Kommunikationsnetzwerk zu empfangen und zu senden. Ein Anbieter von IPv6-fähigen WirelessHART-Feldgeräten **107**, **108**, **109** und **110** aus **Fig.** 6 kann zum Beispiel in einigen Szenarien ein Software Development Kit (SDK) erzeugen, um es Anwendungsentwicklern von Drittanbietern zu ermöglichen, eigene Anwendungen zu erstellen und zu verteilen. Solch ein SDK kann beispielsweise über eine CD-ROM bereitgestellt werden oder von einem Server (nicht gezeigt) über das Internet heruntergeladen werden.

[0159] **Fig.** 15 zeigt ein Blockdiagramm für ein beispielhaftes SDK **700**, das auf einem Allzweck-Computer **701** installiert werden kann, um es einem Entwickler zu ermöglichen, feldgerätspezifische, maßgeschneiderte Anwendungen zu entwickeln, die zum Beispiel die Schnittstelle zur Anwendungsprogrammierung **315** aus **Fig.** 7 verwenden können, um IPv6-Datenframes zu empfangen und zu senden. Wenn sie in einem IPv6-fähigen WirelessHART-Feldgerät **715** ausgeführt wird, das in einem WirelessHART-Kommunikationsnetzwerk **716** arbeitet, kann eine benutzerdefinierte Anwendung konfiguriert werden, um feldgerätspezifische Konfigurations- und Statusinformationen abzurufen, beispielsweise von dem IPv6-fähigen WirelessHART-Feldgerät **715**, und diese Informationen in einem IPv6-Datenframe über ein Gateway **720** an das Internet **717** an einen anderen Computer (z. B. das Gerät **701**) übermitteln.

[0160] Ein beispielhaftes SDK **700** kann eine Feldgerät-Datenbank **702** umfassen, die Informationen für einige oder alle Modelle der IPv6-fähigen, in **Fig.** 6 genannten WirelessHART-Feldgeräte **107**, **108**, **109** und **110** und beispielsweise das von einem Anbieter hergestellte IPv6-fähige WirelessHART-Feldgerät **715** speichert. Natürlich kann die Feldgerät-Datenbank **702** regelmäßig mit neueren Modellen der IPv6-fähigen WirelessHART-Feldgeräte aktualisiert werden. Informationen für ein bestimmtes Modell von IPv6-fähigem WirelessHART-Feldgerät kann die unterstützte Schnittstelle zur Anwendungsprogrammierung und eine Prozessor-Kennung enthalten, welche die Familie des Mikroprozessors in dem jeweiligen Modell des IPv6-fähigen WirelessHART-Feldgeräts usw. anzeigt.

[0161] Eine Datenbank für Feldgerät-Prozessor-Toolketten **704** kann Compiler und Linker umfassen, die verwendet werden können, um ausführbare benutzerdefinierte Anwendungen zu erzeugen. Ein Fachmann wird erkennen, dass jede Familie von Mikroprozessoren ihre eigene einzigartige Toolkette benötigt, um Software-Code zur Erzeugung von ausführbaren benutzerdefinierten Anwendungen zu kompilieren.

[0162] Die verschiedenen hier beschriebenen Datenbanken können sich lokal auf dem Allzweck-Computer **701** befinden, oder sie können sich alternativ auf einem anderen mit dem Internet **717** verbundenen Computer befinden oder auf ihn ”abgebildet” sein. Ein grafisches Benutzeroberflächen(GUI)-Modul **706** kann es einem Entwickler von Drittanbieter-Software (Programmierer) erlauben, mit dem SDK **700** zu interagieren. Selbstver-

ständlich kann die GUI **706** Programmiereingaben empfangen, zum Beispiel über eine Tastatur und eine Maus **718**. Auch kann die GUI **706** zum Beispiel über einen Anzeigemonitor **719** eine Windows-basierte Software-Schnittstelle bereitstellen. Programmierer können mit dem SDK **700** über einen Feldgeräte-Wahlschalter **707** interagieren, um das besondere Modell des IPv6-fähigen WirelessHART-Feldgeräts **715** auszuwählen, beispielsweise aus der Feldgeräte-Datenbank **702**. Der Feldgeräte-Wahlschalter **707** kann beispielsweise auf der GUI **706** als Drop-Down-Menü dargestellt werden.

[0163] Eine Build-Engine **708** kann verwendet werden, um automatisch eine geeignete Toolkette aus der Datenbank für Feldgerät-Prozessor-Toolketten **704** auszuwählen, wenn ein Programmierer ein IPv6-fähiges WirelessHART-Feldgerät über den Feldgeräte-Wahlschalter **707** auswählt. Natürlich würde die gewählte Toolkette zum Beispiel dem Mikroprozessor in dem ausgewählten Modell des IPv6-fähigen WirelessHART-Feldgeräts **715** entsprechen.

[0164] Die GUI-**706** ermöglicht es einem Programmierer, ein Software-Anwendung (Quellcode) in einem Code-Editor-Modul **710** zu schreiben. Der Programmierer kann zum Beispiel die Build-Engine **708** anweisen, die Software-Anwendung zu kompilieren und eine benutzerdefinierte Anwendung zu erzeugen, die in dem Anwendungsblock **334** des Modells des IPv6-fähigen WirelessHART-Feldgeräts **715** ausgeführt werden kann. Die Build-Engine **708** wird die zuvor ausgewählte Toolkette verwenden.

[0165] Ein Programmierer kann dann die ausführbare benutzerdefinierte Anwendung über einen Feldgerät-Emulator **712** validieren und testen. Der Feldgerät-Emulator **712** kann zum Beispiel in dem ausgewählten Modell des IPv6-fähigen WirelessHART-Feldgeräts **715** eine Software-Umgebung einschließlich der Schnittstelle zur Anwendungsprogrammierung **334** bereitstellen.

[0166] Die ausführbare benutzerdefinierte Anwendung und der entsprechende Source-Code können in einer Anwendungsdatenbank **714** gespeichert werden, wenn dies gewünscht wird. Ein Anlagenbetreiber kann die ausführbare benutzerdefinierte Anwendung über das Internet **717** auf das IPv6-fähige WirelessHART-Feldgerät herunterladen, z. B. auf das Gerät **715**, wenn dies gewünscht wird. Die benutzerdefinierte Anwendung kann zum Beispiel in dem Anwendungsblock **334** des IPv6-fähigen WirelessHART-Feldgerätes **715** ausgeführt werden, und die API **332** nutzen, um mit gerätespezifischen Funktionen im Block **336** zu interagieren. Auch die kundenspezifische Anwendung kann über das WirelessHART-Kommunikationsnetz **717** IPv6-Datenframes aus dem Internet **717** empfangen und dorthin senden, indem die Funktionalität in der API **332** und die hier beschriebenen Techniken genutzt werden.

[0167] Wie oben erwähnt, können zumindest einige der beschriebenen beispielhaften Kommunikationsverfahren und/oder Vorrichtungen zur Implementierung dieser Verfahren durch ein oder mehrere Software- und/oder Firmware-Programme auf einem Computerprozessor laufen. Jedoch können spezifische Hardware-Implementierungen, einschließlich, aber nicht beschränkt auf, anwendungsspezifische integrierte Schaltungen, programmierbare Logik-Arrays und andere Hardware-Geräte, ebenfalls konstruiert werden, um einige oder alle der beispielhaften Verfahren und/oder hier beschriebenen Vorrichtungen ganz oder teilweise zu implementieren. Außerdem können alternative Software-Implementierungen, einschließlich, aber nicht beschränkt auf, die verteilte Verarbeitung oder Komponenten-/Objekt-verteilte Verarbeitung, parallele Verarbeitung oder eine virtuelle Maschinenverarbeitung auch konstruiert werden, um die beispielhaften hier beschriebenen Verfahren und/oder Systeme zu implementieren.

[0168] Es sei auch bemerkt, dass die hier beschriebene Beispiel-Software und/oder Firmware-Implementierungen auf einem greifbaren computerlesbaren Speichermedium gespeichert werden, beispielsweise auf einem magnetischen Medium (z. B. eine Magnetplatte oder ein Band), auf einem magneto-optischen oder optischen Medium, z. B. eine optische Disk, oder auf einem Solid-State-Medium, z. B. eine Speicherkarte, oder ein anderes Paket, das einen oder mehrere (nicht-flüchtige) Speicher, Direktzugriffsspeicher oder andere wiederbeschreibbare (volatile) Speicher beherbergt. Dementsprechend kann die hier beschriebene beispielhafte Software und/oder Firmware auf einem greifbaren Speichermedium, z. B. die oben beschriebenen oder Nachfolgespeichermedien, gespeichert werden. Soweit die obenstehende Ausführung beispielhafte Komponenten und Funktionen unter Bezugnahme auf bestimmte Standards und Protokolle beschreibt, ist es verständlich, dass der Schutzumfang dieses Patents nicht auf solche Standards und Protokolle beschränkt ist. Zum Beispiel ist jeder der Standards für das Internet und andere paketvermittelte Netzwerkübertragung (z. B. Transmission Control Protocol(TCP)/Internet Protocol (IP), User Datagram Protocol(UDP)/IP, Hypertext Markup Language (HTML), Hypertext Transfer Protokoll (HTTP)), IPv4, IPv6, WirelessHART usw.) ein Beispiel für den aktuellen Stand der Technik. Solche Normen werden regelmäßig durch schnellere oder effizientere Entsprechungen mit derselben allgemeinen Funktionalität ersetzt. Dementsprechend sind Ersatz-Standards und Protokolle mit den

gleichen Funktionen Entsprechungen, die durch dieses Patent in Betracht gezogen werden und innerhalb des Umfangs der beigefügten Ansprüche enthalten sein sollen.

[0169] Daneben sei darauf hingewiesen, dass, obwohl dieses Patent beispielhafte Verfahren und Vorrichtungen offenbart, einschließlich der auf Hardware ausgeführten Software oder Firmware, solche Systeme lediglich veranschaulichend und nicht als einschränkend zu betrachten sind. Beispielsweise ist es denkbar, dass einige oder alle dieser Hardware- und Software-Komponenten ausschließlich in Hardware, ausschließlich in Software, ausschließlich in Firmware oder in irgendeiner Kombination aus Hardware, Firmware und/oder Software enthalten sind. Dementsprechend gilt, dass, während die obige Ausführung beispielhafte Verfahren, Systeme und/oder maschinenzugängliche Medien beschreibt, die Beispiele nicht die einzige Möglichkeit sind, solche Systeme, Verfahren und maschinenzugänglichen Medien zu implementieren. Daher ist der Umfang des Anwendungsbereichs dieses Patents nicht auf diese beschränkt, auch wenn bestimmte beispielhafte Verfahren, Systeme und maschinenzugängliche Medien hier beschrieben worden sind.

Patentansprüche

1. Verfahren zum Routen von Datenpaketen in einem Kommunikationsnetzwerk, Folgendes umfassend:
Erhalt eines primären Datenpakets in einem ersten Gerät im Kommunikationsnetzwerk als ein Datenpaket, das erste Netzwerkrouting-Informationen entsprechend einem ersten Kommunikationsprotokoll enthält, welches das erste Netzwerkrouting-Protokoll verwendet.
Hinzufügen von zweiten Netzwerkrouting-Informationen zum primären Datenpaket, wobei die zweiten Netzwerkrouting-Informationen dem zweiten Kommunikationsprotokoll entsprechen, das ein zweites Netzwerkrouting-Protokoll verwendet, wobei sich das zweite Netzwerkrouting-Protokoll von dem ersten Netzwerkrouting-Protokoll unterscheidet und ein implizites Ziel-Routingprotokoll umfasst, um dadurch ein eingekapseltes primäres Datenpaket zu erzeugen; und
Senden des eingekapselten primären Datenpakets vom ersten Gerät an ein zweites Gerät über das Kommunikationsnetzwerk unter Verwendung des zweiten Netzwerkrouting-Protokolls.

2. Das Verfahren zum Routen eines Datenpakets nach Anspruch 1, einschließlich des Decodierens der zweiten Netzwerkrouting-Informationen von dem gesendeten eingekapselten primären Datenpaket im zweiten Gerät, um ein primäres Datenpaket mit den ersten Netzwerkrouting-Informationen darin herzustellen.

3. Das Verfahren zum Routen eines Datenpakets nach Anspruch 2, einschließlich der Nutzung der ersten Netzwerkrouting-Informationen im zweiten Gerät, um ein Payload des primären Datenpakets an eine Anwendung im zweiten Gerät zu senden.

4. Das Verfahren zum Routen eines Datenpakets nach Anspruch 2, einschließlich der Nutzung der ersten Netzwerkrouting-Informationen des primären Datenpakets im zweiten Gerät, um das primäre Datenpaket über ein zweites Kommunikationsnetzwerk, welches das erste Netzwerkrouting-Protokoll verwendet, an ein weiteres Gerät zu übertragen.

5. Das Verfahren zum Routen eines Datenpakets nach Anspruch 4, wobei das erste Netzwerkrouting-Protokoll ein Internetprotokoll (IP) ist, das IP-Adressierung benutzt, und wobei die ersten Netzwerkrouting-Informationen des primären Datenpakets eine oder mehrere IP-Adressen umfassen, und wobei das zweite Netzwerkrouting-Protokoll ein Nicht-IP-Routingprotokoll ist, das keine IP-Adressierung benutzt, um Datenpakete zu routen.

6. Das Verfahren zum Routen eines Datenpakets nach Anspruch 5, wobei das zweite Netzwerkrouting-Protokoll ein WirelessHART-Netzwerkrouting-Protokoll ist.

7. Das Verfahren zum Routen eines Datenpakets nach Anspruch 5, wobei das zweite Netzwerkrouting-Protokoll ein Netzwerkrouting-Protokoll ist, das Graph-Routing nutzt, um das Netzwerkrouting durchzuführen.

8. Das Verfahren zum Routen eines Datenpakets nach Anspruch 1, einschließlich des Fragmentierens des primären Datenpakets im ersten Gerät vor dem Hinzufügen der zweiten Netzwerkrouting-Informationen, um ein Set von fragmentierten primären Datenpaketen zu erzeugen, und wobei das Hinzufügen der zweiten Netzwerkrouting-Informationen zum primären Datenpaket das Hinzufügen von zweiten Netzwerkrouting-Informationen zu jedem Set von fragmentierten primären Datenpaketen umfasst.

9. Das Verfahren zum Routen eines Datenpakets nach Anspruch 8, wobei das Fragmentieren des primären Datenpakets im ersten Gerät das Hinzufügen eines Fragment-Headers an jedes Set der fragmentierten primären Datenpakete umfasst.

10. Das Verfahren zum Routen eines Datenpakets nach Anspruch 8, einschließlich des Zusammenfügens des Sets von fragmentierten primären Datenpaketen im zweiten Gerät nach dem Decodieren der zweiten Netzwerkrouting-Informationen von jedem Set der fragmentierten primären Datenpakete.

11. Das Verfahren zum Routen eines Datenpakets nach Anspruch 8, wobei das Fragmentieren des primären Datenpakets im ersten Gerät das Verwenden eines 6LoWPAN-Fragmentierungsprotokolls umfasst, um das erste primäre Datenpaket zu fragmentieren.

12. Das Verfahren zum Routen eines Datenpakets nach Anspruch 1, wobei das Hinzufügen der zweiten Netzwerkrouting-Informationen zum primären Datenpaket das Codieren der zweiten Netzwerkrouting-Informationen beinhaltet, mit einer Anzeige, dass das eingekapselte primäre Datenpaket das primäre Datenpaket mit der ersten Netzwerkrouting-Informationen umfasst, die dem ersten Netzwerkrouting-Protokoll entspricht.

13. Das Verfahren zum Routen eines Datenpakets nach Anspruch 12, wobei das Codieren der zweiten Netzwerkrouting-Informationen einen Hinweis beinhaltet, dass ein bestimmtes Bit der zweiten Netzwerkrouting-Informationen auf einen bestimmten Wert gesetzt ist.

14. Das Verfahren zum Routen eines Datenpakets nach Anspruch 12, wobei das Decodieren der zweiten Netzwerkrouting-Informationen vom gesendeten eingekapselten primären Datenpaket im zweiten Gerät, um das primäre Datenpaket mit der ersten Netzwerkrouting-Informationen zu erstellen, das Erkennen der Anzeige umfasst, dass das gesendete eingekapselte primäre Datenpaket das primäre Datenpaket mit der ersten Netzwerkrouting-Informationen umfasst, die dem ersten Netzwerkrouting-Protokoll entspricht.

15. Das Verfahren zum Routen eines Datenpakets nach Anspruch 1, wobei das Senden des eingekapselten primären Datenpakets über das Kommunikationsnetzwerk unter Verwendung des zweiten Netzwerkprotokolls vom ersten Gerät zum zweiten Gerät das Senden des eingekapselten primären Datenpakets über ein oder mehrere zwischengeschaltete Geräte im Kommunikationsnetzwerk mithilfe des zweiten Netzwerkrouting-Protokolls beinhaltet.

16. Das Verfahren zum Routen eines Datenpakets nach Anspruch 1, wobei der Erhalt des primären Datenpakets in einem ersten Gerät im Kommunikationsnetzwerk als ein Datenpaket, das erste Netzwerkrouting-Informationen entsprechend dem ersten Kommunikationsprotokoll enthält, welches das erste Netzwerkrouting-Protokoll verwendet, den Empfang des primären Datenpakets über ein weiteres Kommunikationsprotokoll umfasst, welches das erste Netzwerkrouting-Protokoll nutzt.

17. Das Verfahren zum Routen eines Datenpakets nach Anspruch 16, einschließlich der Nutzung der ersten Netzwerkrouting-Informationen des primären Datenpakets und einer gespeicherten Routing-Tabelle im ersten Gerät, um festzustellen, ob das primäre Datenpaket an ein zweites Gerät im Kommunikationsnetzwerk gesendet werden soll.

18. Verfahren zum Senden eines Datenpakets über mehrere Kommunikationsnetzwerke, Folgendes umfassend:
Erzeugen eines Datenpakets in einem ersten Gerät in einem ersten Kommunikationsnetzwerk, das ein erstes Netzwerkrouting-Protokoll verwendet, das ein implizites Ziel-Routingprotokoll ist, um Netzwerkkommunikation durchzuführen, so dass die Datenpakete über ein zweites Kommunikationsnetzwerk an ein zweites Gerät gesendet werden, das ein zweites Netzwerkrouting-Protokoll verwendet, wobei das Datenpaket zweite Netzwerkrouting-Informationen umfasst, so wie sie vom zweiten Netzwerkrouting-Protokoll definiert sind;
Einkapseln des Datenpakets in erste Netzwerkrouting-Informationen, wobei die ersten Netzwerkrouting-Informationen dem ersten Netzwerkrouting-Protokoll entsprechen, um dadurch ein eingekapseltes Datenpaket zu erzeugen;
Senden des eingekapselten Datenpakets über das erste Kommunikationsnetzwerk unter Verwendung des ersten Netzwerkrouting-Protokolls vom ersten Gerät an ein Gateway-Gerät, das mit dem ersten Kommunikationsnetzwerk und mit dem zweiten Kommunikationsnetzwerk verbunden ist;
Decodieren der ersten Netzwerkrouting-Informationen von dem gesendeten eingekapselten Datenpaket im Gateway-Gerät, um das Datenpaket mit den zweiten Netzwerkrouting-Informationen darin herzustellen, und

Verwenden der zweiten Netzwerkrouting-Informationen des Datenpakets im Gateway-Gerät, um das Datenpaket unter Nutzung des zweiten Netzwerkrouting-Protokolls über das zweite Kommunikationsnetzwerk an das zweite Gerät zu übertragen.

19. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, wobei das zweite Netzwerkrouting-Protokoll ein Internetprotokoll(IP)-Routingprotokoll ist, das IP-Adressierung verwendet, und wobei das erste Netzwerkrouting-Protokoll ein Nicht-IP-Routingprotokoll ist, das keine IP-Adressierung benutzt.

20. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, wobei das erste Netzwerkrouting-Protokoll ein WirelessHART-Kommunikationsnetzwerk und das zweite Kommunikationsnetzwerk ein auf einem Internetprotokoll basierendes Kommunikationsnetzwerk ist.

21. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, einschließlich des Fragmentierens des Datenpaketes im ersten Gerät vor dem Einkapseln des Datenpakets in die ersten Netzwerkrouting-Informationen, um ein Set von fragmentierten Datenpaketen zu erzeugen, und wobei das Einkapseln des Datenpakets in die ersten Netzwerkrouting-Informationen das Hinzufügen von ersten Netzwerkrouting-Informationen zu jedem Set von fragmentierten Datenpaketen umfasst.

22. Das Verfahren zum Senden eines Datenpakets nach Anspruch 21, wobei das Fragmentieren des primären Datenpakets im ersten Gerät das Hinzufügen eines Fragment-Headers an jedes Set der fragmentierten Datenpakete umfasst.

23. Das Verfahren zum Senden eines Datenpakets nach Anspruch 21, einschließlich des Zusammenfügens des Sets von fragmentierten Datenpaketen, um das Datenpaket im Gateway-Gerät mit den zweiten Netzwerkrouting-Informationen zu erstellen, nachdem das Decodieren der ersten Netzwerkrouting-Informationen von jedem Set der fragmentierten Datenpakete erfolgt ist.

24. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, wobei das Einkapseln des Datenpakets mit den ersten Netzwerkrouting-Informationen das Einkapseln der ersten Netzwerkrouting-Informationen mit einem Hinweis darauf beinhaltet, dass das eingekapselte Datenpaket ein Datenpaket umfasst, das mit den zweiten Netzwerkrouting-Informationen verbunden ist.

25. Das Verfahren zum Senden eines Datenpakets nach Anspruch 24, wobei das Decodieren der ersten Netzwerkrouting-Informationen vom gesendeten eingekapselten Datenpaket im Gateway-Gerät, um das Datenpaket mit den zweiten Netzwerkrouting-Informationen zu erstellen, das Erkennen des Hinweises in den ersten Netzwerkrouting-Informationen umfasst, dass das gesendete eingekapselte Datenpaket ein primäres Datenpaket umfasst, das mit zweiten Netzwerkrouting-Informationen verbunden ist.

26. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, wobei das Senden des eingekapselten Datenpakets über das erste Kommunikationsnetzwerk unter Verwendung des ersten Netzwerkrouting-Protokolls vom ersten Gerät zum Gateway-Gerät das Senden des eingekapselten Datenpakets über ein oder mehrere zwischengeschaltete Geräte im ersten Kommunikationsnetzwerk mithilfe der ersten Informationen des Netzwerkrouting-Protokolls beinhaltet.

27. Das Verfahren zum Senden eines Datenpakets nach Anspruch 18, einschließlich des Empfangs des Datenpakets über das zweite Kommunikationsnetzwerk in einem weiteren Gateway-Gerät, das mit einem zweiten Kommunikationsnetzwerk und einem dritten Kommunikationsnetzwerk verbunden ist, und unter Verwendung der zweiten Netzwerkrouting-Informationen des Datenpakets, um festzustellen, ob das Datenpaket an ein zweites Gerät im dritten Kommunikationsnetzwerk gesendet werden soll.

28. Das Verfahren zum Senden eines Datenpaketes nach Anspruch 27, wobei das dritte Kommunikationsnetzwerk das erste Netzwerkrouting-Protokoll verwendet, einschließlich des Bestimmens weiterer erster Netzwerkrouting-Informationen aus den zweiten Netzwerkrouting-Informationen des Datenpakets, die benötigt werden, um das Datenpaket über das dritte Kommunikationsnetzwerk an das zweite Gerät zu senden und das Datenpaket mit den zweiten Netzwerkrouting-Informationen darin in die weiteren ersten Netzwerkrouting-Informationen einzukapseln und das eingekapselte Datenpaket in die weiteren ersten Netzwerkrouting-Informationen unter Verwendung des ersten Netzwerkrouting-Protokolls über das dritte Kommunikationsnetzwerk an das zweite Gerät zu senden.

29. Das Verfahren zum Senden eines Datenpakets nach Anspruch 28, wobei das Einkapseln des Datenpakets mit den zweiten Netzwerkrouting-Informationen darin in die weiteren ersten Netzwerkinformation das Fragmentieren des Datenpakets umfasst und gleichzeitig die zweiten Netzwerkrouting-Informationen bewahrt, um ein Set von weiteren fragmentierten Datenpaketen zu erzeugen und weitere erste Netzwerkrouting-Informationen zu jedem Set von fragmentierten Datenpaketen hinzuzufügen.

30. Verfahren zum Routen eines Datenpakets über mehrere Kommunikationsnetzwerke, Folgendes umfassend:
Erzeugen eines Datenpakets in einem ersten Gerät;
Bereitstellen des Datenpakets im ersten Gerät mit ersten Netzwerkrouting-Informationen, die einem ersten Netzwerkrouting-Protokoll entsprechen, wobei die ersten Netzwerkrouting-Informationen festlegen, dass das Datenpaket an ein zweites Gerät geroutet wird;
Senden des Datenpakets über ein erstes Kommunikationsnetzwerk hinweg, welches das erste Netzwerkrouting-Protokoll verwendet;
Empfang des gesendeten Datenpakets in einem an das erste Kommunikationsnetzwerk und an ein zweites Kommunikationsnetzwerk gekoppelten Gateway-Gerät, das ein zweites Netzwerkrouting-Protokoll verwendet, das ein implizites Ziel-Routingprotokoll ist;
im Gateway-Gerät anhand der ersten Netzwerkrouting-Informationen feststellen, ob das empfangene Datenpaket über das zweite Kommunikationsnetzwerk an das zweite Gerät gesendet werden soll;
Einkapseln des Datenpakets mit den ersten Netzwerkrouting-Informationen darin in zweite Netzwerkrouting-Informationen entsprechend dem zweiten Netzwerkrouting-Protokoll, um dadurch ein eingekapseltes Datenpaket zu erzeugen;
Senden des eingekapselten Datenpaketes über das zweite Kommunikationsnetzwerk an das zweite Gerät, welches das zweite Netzwerkrouting-Protokoll verwendet;
Empfangen des gesendeten eingekapselten Datenpakets am zweiten Gerät und anhand der zweiten Netzwerkrouting-Informationen feststellen, ob das Datenpaket vom zweiten Gerät empfangen werden soll; und
Decodieren der ersten Netzwerkrouting-Informationen am zweiten Gerät und anhand des ersten Netzwerkroutings im zweiten Gerät ein Ziel bestimmen, welches das Datenpaket empfangen soll.

31. Das Verfahren zum Routen eines Datenpakets nach Anspruch 30, wobei das erste Netzwerkrouting-Protokoll ein Internetprotokoll(IP)-Routingprotokoll ist, das IP-Adressierung verwendet, und wobei das zweite Netzwerkrouting-Protokoll ein Nicht-IP-Routingprotokoll ist, das keine IP-Adressierung benutzt.

32. Das Verfahren zum Routen eines Datenpakets nach Anspruch 30, einschließlich des Fragmentierens des Datenpaketes im Gateway-Gerät vor dem Einkapseln des Datenpakets in die zweiten Netzwerkrouting-Informationen, um ein Set von fragmentierten Datenpaketen zu erzeugen, und wobei das Einkapseln des Datenpakets in die zweiten Netzwerkrouting-Informationen das Hinzufügen von zweiten Netzwerkrouting-Informationen zu jedem Set von fragmentierten Datenpaketen umfasst.

33. Das Verfahren zum Routen eines Datenpakets nach Anspruch 32, einschließlich des Zusammenfügens des Sets von fragmentierten Datenpaketen im zweiten Gerät, um das Datenpakt mit den ersten Netzwerkrouting-Informationen darin zu erstellen, nachdem das Decodieren der zweiten Netzwerkrouting-Informationen von jedem Set der fragmentierten Datenpakete erfolgt ist.

34. Das Verfahren zum Routen eines Datenpakets nach Anspruch 30, wobei das Einkapseln des Datenpakets in die zweiten Netzwerkrouting-Informationen das Einkapseln der zweiten Netzwerkrouting-Informationen mit einem Hinweis darauf beinhaltet, dass das eingekapselte Datenpaket ein Datenpaket mit ersten Netzwerkrouting-Informationen umfasst.

35. Das Verfahren zum Routen eines Datenpakets nach Anspruch 34, wobei das Decodieren der zweiten Netzwerkrouting-Informationen vom gesendeten eingekapselten Datenpaket im zweiten Gerät, um das Datenpaket mit den ersten Netzwerkrouting-Informationen zu erstellen, das Erkennen des Hinweises in den zweiten Netzwerkrouting-Informationen umfasst, dass das gesendete eingekapselte Datenpaket ein Datenpaket mit ersten Netzwerkrouting-Informationen umfasst.

36. Ein Kommunikationsgerät zur Kommunikation in einem ersten Kommunikationsnetzwerk, das ein erstes Kommunikationsprotokoll verwendet, dass ein erstes Netzwerkrouting-Protokoll, ein Datenlink-Protokoll und ein physisches Protokoll enthält, Folgendes umfassend:
ein physisches Layer-Element, gekoppelt an ein erstes Kommunikationsnetzwerk, das mithilfe des physischen Protokolls Datenpakete empfängt und sendet;

ein Datenlinklayer-Stack-Element, gekoppelt an das erste physische Layer-Element, das Datenpakete mithilfe des Datenlink-Protokolls codiert und decodiert;
ein erstes Stack-Element des Netzwerk-Routinglayers, gekoppelt an das Stack-Element des Datenlinklayers, das Datenpakete für das Senden über das Kommunikationsnetzwerk codiert, indem erste Netzwerkrouting-Informationen zu Datenpaketen hinzugefügt werden, um sie über das erste Kommunikationsnetzwerk mit den ersten Netzwerkrouting-Informationen zu senden, die dem ersten Netzwerkrouting-Protokoll entsprechen, und das Datenpakete decodiert, die über das erste Kommunikationsnetzwerk empfangen wurden, indem die ersten Netzwerkrouting-Informationen ausgelesen werden, die über das erste Kommunikationsnetzwerk als Datenpakete empfangen wurden, wobei die ersten Netzwerkrouting-Informationen dem ersten Netzwerkrouting-Protokoll entsprechen;
ein erstes Stack-Element des Anwendungslayers, gekoppelt an das erste Stack-Element des Netzwerkrouting-Layers, das Anwendungen ausführt, die im ersten Kommunikationsnetzwerk unter Verwendung des ersten Stack-Elements des Netzwerkrouting-Layers kommunizieren;
ein zweites Stack-Element des Netzwerkrouting-Layers, gekoppelt an das erste Stack-Element des Netzwerkrouting-Layers, das so funktioniert, dass es die zweiten Netzwerkrouting-Informationen innerhalb von Datenpaketen codiert und decodiert, die einem zweiten Netzwerkrouting-Protokoll entsprechen, wobei das zweite Stack-Element des Netzwerkrouting-Layers so funktioniert, dass es Datenpakete codiert, die über das erste Kommunikationsnetzwerk gesendet werden, indem die zweiten Netzwerkrouting-Informationen zu Datenpaketen hinzugefügt werden, die über das erste Kommunikationsnetzwerk entsprechend einem zweiten Netzwerkrouting-Protokoll gesendet werden, das sich vom ersten Netzwerkrouting-Protokoll unterscheidet und das so funktioniert, dass es Datenpakete decodiert, die vom ersten Stack-Element des Netzwerkrouting-Layers empfangen werden, indem die zweiten Netzwerkrouting-Informationen von Datenpaketen ausgelesen werden, die vom ersten Stack-Element des Netzwerklayers empfangen wurden, wobei die zweiten Netzwerkrouting-Informationen dem zweiten Netzwerkrouting-Protokoll entsprechen; und
ein zweites Stack-Element des Anwendungslayers, gekoppelt an das zweite Stack-Element des Netzwerkrouting-Layers, das die Anwendungen ausführt, die mithilfe von Datenpaketen kommunizieren, die das zweite Stack-Element des Netzwerkrouting-Layers benutzen;
wobei das erste Stack-Element des Netzwerkrouting-Layers die ersten Netzwerkrouting-Informationen in Datenpaketen vom Stack-Element des Datenlinklayers entsprechend dem ersten Netzwerkrouting-Protokoll decodiert, um festzustellen, ob die Datenpakete Daten enthalten, die mit dem zweiten Netzwerkrouting-Protokoll verbunden sind, und die decodierten Datenpakete an das erste Stack-Element des Anwendungslayers liefert, wenn die decodierten Datenpakete keine Daten enthalten, die mit dem zweiten Netzwerkrouting-Protokoll verbunden sind, und um die decodierten Datenpakete an das zweite Stack-Element des Netzwerklayers zu liefern, wenn die decodierten Datenpakete Daten enthalten, die mit dem zweiten Netzwerkrouting-Protokoll verbunden sind, und wobei das erste Stack-Element des Netzwerkrouting-Layers Datenpakete codiert, die sowohl über das erste Stack-Element des Anwendungslayers als auch über das zweite Stack-Element des Netzwerkrouting-Layers empfangen wurden.

37. Das Kommunikationsgerät aus Anspruch 36, einschließlich eines ersten Stack-Elements des Transportlayers, gekoppelt zwischen dem ersten Stack-Element des Netzwerkrouting-Layers und dem ersten Stack-Element des Anwendungslayers, um das Codieren und Decodieren von Transportinformationen von Datenpaketen gemäß einem Transportprotokoll des ersten Kommunikationsnetzwerks auszuführen;

38. Das Kommunikationsgerät aus Anspruch 37, einschließlich eines zweiten Stack-Elements des Transportlayer, gekoppelt zwischen dem zweiten Stack-Element des Netzwerkrouting-Layers und dem zweiten Stack-Element des Anwendungslayers, um das Codieren und Decodieren von Transportinformationen von Datenpaketen gemäß einem Transportprotokoll eines zweiten Kommunikationsnetzwerks auszuführen;

39. Das Kommunikationsgerät aus Anspruch 36, wobei das erste Stack-Element des Netzwerklayers die ersten Netzwerkrouting-Informationen in den Datenpaketen, die vom Stack-Element des Datenlinklayers empfangen wurden, entsprechend dem ersten Netzwerkrouting-Protokoll decodiert, indem eine Kennung innerhalb der ersten Netzwerkrouting-Informationen ausgelesen wird, die in den Datenpaketen enthalten war, die vom Stack-Element des Datenlinklayers empfangen wurden.

40. Das Kommunikationsgerät nach Anspruch 36, wobei das erste Stack-Element des Netzwerklayers die ersten Netzwerkrouting-Informationen decodiert, die in dem vom zweiten Stack-Element des Netzwerklayers empfangenen Datenpaket empfangen wurden, indem in den ersten Netzwerkrouting-Informationen, die in den Datenpaketen enthalten sind, die vom zweiten Stack-Element des Netzwerklayers empfangen wurden, eine Kennung untergebracht wird, dass das zugrunde liegende Datenpaket Daten beinhaltet, die mit dem zweiten Netzwerkrouting-Protokoll verbunden sind.

41. Das Kommunikationsgerät aus Anspruch 36, wobei das zweite Stack-Element des Netzwerkrouting-Layers Datenpakete codiert, um darin zweite Netzwerkinformationen einzufügen, und einen Paketfragmentierer einfügt, der jedes der codierten Datenpakete in zwei oder mehr Datenpakete fragmentiert und die fragmentierten Datenpakete dem ersten Stack-Element des Netzwerkrouting-Layers bereitstellt.

42. Das Kommunikationsgerät aus Anspruch 41, wobei das zweite Stack-Element des Netzwerkrouting-Layers einen Defragmentierer beinhaltet, der zwei oder mehr Datenpakete zusammenfügt, die vom ersten Stack-Element des Netzwerkrouting-Layers empfangen wurden, um ein Datenpaket einschließlich der zweiten Netzwerkrouting-Informationen zu erstellen.

43. Das Kommunikationsgerät aus Anspruch 36, einschließlich einer Routing-Tabelle, welche die Verknüpfungsinformationen speichert, die Routing-Informationen für ein bestimmtes Gerät im ersten Netzwerkrouting-Protokoll mit Routing-Informationen für das bestimmte Gerät im zweiten Netzwerkrouting-Protokoll verknüpft.

44. Das Kommunikationsgerät aus Anspruch 36, wobei das zweite Netzwerkrouting-Protokoll ein Internetprotokoll(IP)-Routingprotokoll ist, das IP-Protokoll-Routing verwendet, und das erste Netzwerkrouting-Protokoll ein Nicht-IP-Routingprotokoll ist.

45. Das Kommunikationsgerät aus Anspruch 44, wobei das zweite Netzwerkrouting-Protokoll ein IP Version 6 Protokoll ist.

46. Das Kommunikationsgerät aus Anspruch 45, wobei das erste Netzwerkrouting-Protokoll ein WirelessHART-Netzwerkrouting-Protokoll ist.

47. Ein Gateway-Gerät zur Kommunikation in einem ersten Kommunikationsnetzwerk, das ein erstes Netzwerkrouting-Protokoll verwendet, und zur Kommunikation in einem zweiten Kommunikationsnetzwerk, das ein zweites Netzwerkrouting-Protokoll verwendet, das sich vom ersten Netzwerkrouting-Protokoll unterscheidet, Folgendes umfassend:
Einen ersten Kommunikationsstack mit:
einem physischen Layer-Element für Verbindungen mit dem ersten Kommunikationsnetzwerk, das Datenpakete mithilfe eines ersten physischen Layer-Protokolls des ersten Kommunikationsnetzwerks empfängt und sendet;
einem erstem Stack-Element des Datenlinklayers, gekoppelt an das erste physische Layer-Element, das Datenpakete mithilfe des ersten Datenlinkprotokolls des ersten Kommunikationsprotokolls codiert und decodiert;
ein erstes Stack-Element des Netzwerk-Routinglayers, gekoppelt an das erste Stack-Element des Datenlinklayers, um Datenpakete, die über das Kommunikationsnetzwerk mithilfe von ersten Netzwerkrouting-Informationen in Datenpaketen gesendet werden sollen, um sie über das erste Kommunikationsnetzwerk mit den ersten Netzwerkrouting-Informationen zu senden, die dem ersten Netzwerkrouting-Protokoll entsprechen, das Datenpakete decodiert, die über das erste Kommunikationsnetzwerk empfangen wurden, indem die ersten Netzwerkrouting-Informationen ausgelesen werden, die in Datenpaketen enthalten sind, die über das erste Kommunikationsnetzwerk empfangen wurden, wobei die ersten Netzwerkrouting-Informationen dem ersten Netzwerkrouting-Protokoll entsprechen;
einen zweiten Kommunikationsstack mit:
einem zweiten physischen Layer-Element für Verbindungen mit dem zweiten Kommunikationsnetzwerk, das Datenpakete mithilfe eines zweiten physischen Layer-Protokolls des zweiten Kommunikationsnetzwerks empfingt und sendet;
einem zweiten Stack-Element des Datenlinklayers, gekoppelt an das zweite physische Layer-Element, das Datenpakete mithilfe des zweiten Datenlinkprotokolls des zweiten Kommunikationsprotokolls codiert und decodiert;
einem zweiten Stack-Element des Netzwerk-Routinglayers, gekoppelt an das zweite Stack-Element des Datenlinklayers, das Datenpakete für das Senden über das zweite Kommunikationsnetzwerk codiert, indem zweite Netzwerkrouting-Informationen zu Datenpaketen hinzugefügt werden, um sie über das zweite Kommunikationsnetzwerk mit den zweiten Netzwerkrouting-Informationen zu senden, die dem zweiten Netzwerkrouting-Protokoll entsprechen, das ein implizites Ziel-Adressierungs-Netzwerk ist, das und das Datenpakete decodiert, die über das zweite Kommunikationsnetzwerk empfangen wurden, indem die zweiten Netzwerkrouting-Informationen ausgelesen werden, die über das zweite Kommunikationsnetzwerk als Datenpakete empfangen wurden, wobei die zweiten Netzwerkrouting-Informationen dem zweiten Netzwerkrouting-Protokoll entsprechen; und
eine Routing-Tabelle, welche die Verknüpfungsinformationen für ein oder mehrere Geräte in dem zweiten Kommunikationsnetzwerk speichert, wobei die Verknüpfungsinformationen, welche die ersten Netzwerkrou-

ting-Informationen für das oder die Geräte im zweiten Kommunikationsnetzwerk mit den zweiten Netzwerkrouting-Informationen für das oder die Geräte im zweiten Kommunikationsnetzwerk verbunden werden;
wobei das zweite Stack-Element des Netzwerkrouting-Layers Datenpakete decodiert, die erste Netzwerkrouting-Informationen enthalten, die vom zweiten Stack-Element des Datenlinklayers empfangen wurden, und das die Datenpakete mit darin enthaltenen ersten Netzwerkrouting-Informationen an den ersten Kommunikationsstack sendet, der sie an ein Gerät liefert, das die ersten Netzwerkrouting-Informationen nutzt, und wobei das zweite Stack-Element des Netzwerkrouting-Layers die Datenpakete mit darin enthaltenen ersten Netzwerkrouting-Informationen empfängt und die Datenpakete mit darin enthaltenen ersten Netzwerkrouting-Informationen an Geräte im zweiten Kommunikationsnetzwerk sendet, indem die zweiten Netzwerkrouting-Informationen hinzugefügt werden, die auf den in der Routing-Tabelle gespeicherten Verknüpfungsinformationen basieren.

48. Das Gateway-Gerät aus Anspruch 47, das eine Defragmentierungseinheit beinhaltet, die an das zweite Stack-Element des Netzwerkrouting-Layers gekoppelt ist, wobei die Defragmentierungseinheit Datenpakete zusammenfügt, die erste Netzwerkrouting-Informationen enthalten, die zwei oder mehr fragmentierte Datenpakete umfassen, und die fragmentierten Datenpakete dem zweiten Stack-Element des Netzwerkrouting-Layers bereitstellt, um sie über das zweite Kommunikationsnetzwerk zu senden.

49. Das Gateway-Gerät aus Anspruch 47, das eine Defragmentierungseinheit beinhaltet, die an das zweite Stack-Element des Netzwerkrouting-Layers gekoppelt ist, wobei die Defragmentierungseinheit Datenpakete zusammenfügt, die vorher aus einem Datenpaket erstellt wurden, das die ersten Netzwerkrouting-Informationen enthält und über das zweite Kommunikationsnetzwerk empfangen wurde, um das Datenpaket neu zu erstellen, dass die ersten Netzwerkrouting-Informationen enthält.

50. Das Gateway-Gerät nach Anspruch 47, einschließlich einer Fragmentierungs-/Defragmentierungseinheit, gekoppelt an das zweite Stack-Element des Netzwerkrouting-Layers und an das erste Stack-Element des Netzwerkrouting-Layers, wobei die Fragmentierungs-/Defragmentierungseinheit Datenpakete, welche die ersten Informationen des Netzwerkrouting-Layers enthalten, die vom ersten Stack-Element des Netzwerkrouting-Layers empfangen wurden, in zwei oder mehr Datenfragmentpakete fragmentiert und die fragmentierten Datenpakete dem zweiten Stack-Element des Netzwerklayers zum Senden über das zweite Kommunikationsnetzwerk bereitstellt, und wobei die Fragmentierungs-/Defragmentierungseinheit die fragmentierten Datenpakete zusammenfügt, die vorher von einem Datenpaket mit ersten Netzwerkrouting-Informationen erstellt und über das zweite Kommunikationsnetzwerk empfangen wurden, um das Datenpaket mit ersten Netzwerkrouting-Informationen wieder zu erzeugen und das neu erstellte Datenpaket an das erste Stack-Element des Netzwerkrouting-Layers zu liefern.

51. Das Gateway-Gerät aus Anspruch 50, einschließlich eines ersten Stack-Elements des Anwendungslayers und eines Stack-Elements des Transportlayers, gekoppelt zwischen dem ersten Stack-Element des Netzwerkrouting-Layers und dem Stack-Element des Anwendungslayers, um das Codieren und Decodieren von Transportinformationen von Datenpaketen gemäß einem Transportprotokoll des ersten Kommunikationsnetzwerks auszuführen.

52. Das Gateway-Gerät aus Anspruch 50, wobei das erste Stack-Element des Netzwerkrouting-Layers eine Routing-Tabelle umfasst, die Routing-Informationen für Routing-Datenpakete über das erste Kommunikationsnetzwerk beinhaltet.

53. Das Gateway-Gerät aus Anspruch 47, wobei das zweite Stack-Element des Netzwerklayer die zweiten Netzwerkrouting-Informationen in Datenpaketen vom zweiten Stack-Element des Datenlinklayers entsprechend dem zweiten Netzwerkrouting-Protokolls decodiert, indem einer Kennung innerhalb der zweiten Netzwerkrouting-Informationen ausgelesen wird, die auf das Vorhandensein von Datenpaketen mit ersten Netzwerkrouting-Informationen darin hinweist.

54. Das Gateway-Gerät aus Anspruch 47, wobei das zweite Stack-Element des Netzwerkrouting-Layers zweite Netzwerkrouting-Informationen in Datenpaketen durch Platzieren einer Kennung in den zweiten Netzwerkrouting-Informationen codiert, die darüber informiert, dass die zugrunde liegenden Daten mit den ersten Netzwerkrouting-Informationen verbunden sind.

55. Das Gateway-Gerät aus Anspruch 54, einschließlich einer Routing-Tabelle, welche die Verknüpfungsinformationen speichert, die Routing-Informationen für ein bestimmtes Gerät im ersten Netzwerkrouting-Protokoll mit Routing-Informationen für das bestimmte Gerät im zweiten Netzwerkrouting-Protokoll verknüpft.

56. Das Gateway-Gerät aus Anspruch 47, wobei das erste Netzwerkrouting-Protokoll ein Internetprotokoll (IP)-Routingprotokoll ist, das IP-Protokoll-Routing verwendet, und das zweite Netzwerkrouting-Protokoll ein Nicht-IP-Routingprotokoll ist.

57. Das Gateway-Gerät aus Anspruch 56, wobei das erste Netzwerkrouting-Protokoll ein IP Version 6 Protokoll ist.

58. Das Gateway-Gerät aus Anspruch 57, wobei das zweite Netzwerkrouting-Protokoll ein WirelessHART-Netzwerkrouting-Protokoll ist.

59. Verfahren zum Weiterleiten eines Datenpakets, wobei das Verfahren Folgendes umfasst:
das Empfangen des Datenpaketes über einen ersten Kommunikationskanal entsprechend einem ersten Netzwerkrouting-Protokoll;
das Identifizieren einer ersten Zieladresse für das Datenpaket, basierend auf der Analyse von ersten Routing-Informationen entsprechend einem ersten Routingprotokoll, wobei sich die erste Routing-Informationen in dem Datenpaket befinden;
basierend auf der Identifizierung der ersten Zieladresse, das Bestimmen einer zweiten Zieladresse, wobei die zweite Zieladresse einem zweiten Routingprotokoll entspricht, das ein Mesh-Netzwerkrouting-Protokoll ist;
das Framing des Datenpakets mit zweiten Routing-Informationen entsprechend dem zweiten Routingprotokoll, wobei das Framing des Datenpakets die Bewahrung der ersten Routing-Informationen umfasst, und
das Weiterleiten des Datenpakets an die erste Zieladresse über einen zweiten Kommunikationskanal entsprechend einem zweiten Kommunikationsprotokoll, wobei das Datenpaket unter Verwendung der zweiten Routing-Informationen weitergeleitet wird.

60. Verfahren zum Übertragen eines Datenpakets, wobei das Verfahren Folgendes umfasst:
Empfangen des Datenpakets entsprechend einem ersten Routingprotokoll, wobei das Datenpaket eine Angabe der ersten Zieladresse enthält und wobei die erste Zieladresse dem ersten Routingprotokoll entspricht;
basierend auf der ersten Zieladresse das Framing des Datenpakets mit Routing-Informationen, die einem zweiten Routingprotokoll entsprechen, das ein nicht auf einem Internetprotokoll basierendes Protokoll zum Adressierungsrouting ist, wobei die Routing-Informationen eine zweite Zielinformation beinhalten, die dem zweiten Routingprotokoll entspricht, und wobei das Einschließen des Datenpakets das Erzeugen eines Hinweises umfasst, dass das Datenpaket dem ersten Routing-Protokoll entspricht, und
Senden des Datenpakets über einen Kommunikationskanal, erstellt gemäß einem Kommunikationsprotokoll, wobei das Kommunikationsprotokoll die Routing-Informationen verwendet, die dem zweiten Routingprotokoll entspricht, um das Datenpaket an die erste Zieladresse zu routen.

61. Verfahren zum Kommunizieren in einem Netzwerk, wobei das Verfahren Folgendes umfasst:
Empfangen eines Datenpakets über einen Kommunikationskanal, das in Routing-Informationen eingekapselt ist, die einem ersten Netzwerkrouting-Protokoll entsprechen, wobei das erste Netzwerkrouting-Protokoll eine Technik zum Adressierungsrouting verwendet, die nicht auf einem Internet-Protokoll basiert;
Bestimmen, ob das eingekapselte Datenpaket Informationen für eine Anwendung beinhaltet, die dem zweiten Netzwerkrouting-Protokoll entspricht, das die Informationen entsprechend dem ersten Netzwerkrouting-Protokoll nutzt, und
Weiterleiten des eingekapselten Datenpakets an die Anwendung über den Kommunikationsstack entsprechend dem zweiten Netzwerkrouting-Protokoll, basierend auf der Feststellung, ob das eingekapselte Datenpaket Informationen für eine Anwendung enthält, die dem zweiten Netzwerkrouting-Protokoll entsprechen.

62. Das Verfahren nach Anspruch 61, wobei das erste Netzwerkprotokoll das WirelessHART-Protokoll ist.

63. Das Verfahren nach Anspruch 61, wobei das Datenpaket eines von mehreren Datenpaketen ist, wobei mehrere Datenpaketen einem Datenframe entsprechen.

64. Das Verfahren nach Anspruch 63, wobei der Datenframe dem zweiten Netzwerkprotokoll entspricht.

65. Das Verfahren nach Anspruch 64, wobei das zweite Netzwerkprotokoll das Internetprotokoll Version 6 ist (IPv6).

66. Das Verfahren nach Anspruch 63, wobei die Vielzahl der Datenpakete dadurch generiert wird, dass der Datenframe entsprechend dem 6LoWPAN-Protokoll fragmentiert wird.

67. Das Verfahren nach Anspruch 61, wobei der Kommunikationskanal so konfiguriert ist, dass er gemäß einem 802.15.4 Protokoll agiert.

68. Das Verfahren nach Anspruch 61, wobei die Anwendung dem Constrained Application Protocol (CoAP) entspricht.

69. Das Verfahren nach Anspruch 61, wobei das Datenpaket verschlüsselt ist.

70. Das Verfahren nach Anspruch 61, wobei der Datenframe verschlüsselt ist.

71. Verfahren zum Übertragen von mindestens einem von mehreren Datenpaketen in einem drahtlosen Netzwerk über den ersten Kommunikationskanal, das so konfiguriert ist, dass es gemäß einem ersten Kommunikationsprotokoll agiert, wobei das Verfahren Folgendes umfasst:
das Empfangen eines Datenframes über einen zweiten Kommunikationskanal, der so konfiguriert ist, dass er entsprechend einem zweiten Kommunikationsprotokoll agiert, wobei der Datenframe einem zweiten Netzwerkprotokoll entspricht;
das Fragmentieren des Datenframes, um mehrere Datenpakete zu erzeugen;
das Einkapseln von mindestens einem der verschiedenen Datenpakete in Informationen, die einem ersten Netzwerkprotokoll entsprechen, wobei die Informationen Routing-Informationen umfassen, die dem logischen Aufbau des drahtlosen Netzwerks entsprechen;
das Erstellen eines Hinweises in den Informationen, die dem ersten Netzwerkprotokoll entsprechen, der darauf hinweist, dass mindestens eines der verschiedenen Datenpakete Daten umfasst, die dem zweiten Netzwerkprotokoll entsprechen, und
das Übertragen von mindestens einem der verschiedenen Datenpakete über den ersten Kommunikationskanal.

72. Das Verfahren nach Anspruch 71, wobei das erste Kommunikationsprotokoll das Protokoll 802.15.4 ist.

73. Das Verfahren nach Anspruch 71, wobei das zweite Kommunikationsprotokoll das Protokoll 802.3 ist.

74. Das Verfahren nach Anspruch 71, wobei das erste Netzwerkprotokoll das WirelessHART-Netzwerkprotokoll ist.

75. Das Verfahren nach Anspruch 71, wobei das zweite Netzwerkprotokoll das Internetprotokoll Version 6 ist (IPv6).

76. Das Verfahren nach Anspruch 71, wobei das Fragmentieren des Datenframes das Fragmentieren des Datenframes entsprechend dem 6LoWPAN-Protokoll umfasst.

77. Das Verfahren nach Anspruch 71, wobei der empfangene Datenframe verschlüsselt ist und das Verfahren zudem das Entschlüsseln des Datenframes umfasst.

78. Verfahren zum Übermitteln eines Datenframes über einen Kommunikationskanal, der so konfiguriert ist, dass er entsprechend einem ersten Kommunikationsprotokoll agiert, wobei der Datenframe einem ersten Netzwerkprotokoll entspricht. Das Verfahren umfasst:
Empfangen von mehreren Datenpaketen über ein Wireless-Netzwerk, wobei jedes der verschiedenen Datenpakete in Routing-Informationen eingekapselt ist, die einem zweiten Netzwerkrouting-Protokoll entsprechen, und wobei die Routing-Informationen solche Routing-Informationen enthalten, die dem zweiten Netzwerkrouting-Protokoll entsprechen, und wobei die Routing-Informationen dem logischen Aufbau des drahtlosen Netzwerks entsprechen;
das Erfassen eines Hinweises in den Routing-Informationen, die dem zweiten Netzwerkrouting-Protokoll entsprechen, für ein Subset von Datenpaketen aus den verschiedenen Datenpaketen, wobei der Hinweis anzeigt, dass eins oder mehrere Subsets von Datenpaketen dem ersten Netzwerkrouting-Protokoll entsprechende Daten umfassen;
das Verarbeiten des Subsets von Datenpaketen zur Erzeugung des Datenframes entsprechend dem ersten Netzwerkprotokoll, und
das Übertragen des Datenframes über den Kommunikationskanal, der so konfiguriert ist, dass er gemäß dem ersten Kommunikationsprotokoll agiert.

79. Das Verfahren nach Anspruch 78, wobei das Wireless-Netzwerk Kommunikationskanäle umfasst, die gemäß einem zweiten Kommunikationsprotokoll agieren.

80. Das Verfahren nach Anspruch 79, wobei das erste Netzwerkrouting-Protokoll das Internetprotokoll Version 6 (IPv6) ist.

81. Das Verfahren nach Anspruch 79, wobei das erste Kommunikationsprotokoll das Protokoll 802.3 ist.

82. Das Verfahren nach Anspruch 79, wobei das zweite Kommunikationsprotokoll das Protokoll 802.15.4 ist.

83. Das Verfahren nach Anspruch 78, wobei das zweite Netzwerkrouting-Protokoll das WirelessHART-Protokoll ist.

84. Das Verfahren nach Anspruch 78, wobei jedes Subset von Datenpaketen einen 6LoWPAN-Header enthält.

85. Das Verfahren nach Anspruch 84, wobei die Verarbeitung des Subsets von Datenpaketen das Zusammenfügen des Subsets von Datenpaketen umfasst, um den Datenframe zu erzeugen, basierend auf jedem der 6LoWPAN-Header.

86. Das Verfahren nach Anspruch 78, wobei der Datenframe gemäß dem Advanced Encryption Standard (AES) verschlüsselt ist.

87. Das Verfahren nach Anspruch 78, wobei der übertragene Datenframe gemäß dem Data Encryption Standard (DES) verschlüsselt ist.

88. Verfahren zum Kommunizieren in einem ersten Kommunikationsnetzwerk, das ein erstes Netzwerkrouting-Protokoll verwendet, das ein Netzwerkrouting-Protokoll verwendet, das nicht auf einer Internet Protokoll-Adresse basiert (Non-IP), und das ein erstes Gerät enthält, das ein mit dem ersten Kommunikationsnetzwerk verbundenes Gerät beinhaltet, und ein Gateway-Gerät, das sowohl mit dem ersten Kommunikationsnetzwerk als auch mit einem zweiten Kommunikationsnetzwerk verbunden ist, das ein Netzwerkrouting-Protokoll verwendet, das auf einer IP-Adresse basiert, und Folgendes umfasst:
Senden einer Nachricht vom ersten Gerät an das Gateway-Gerät mithilfe des ersten Netzwerkrouting-Protokolls; die Nachricht fordert eine mit dem zweiten Netzwerkrouting-Protokoll verbundene IP-Adresse für eine bestimmte Netzwerkeinheit an;
Empfangen der Nachricht an dem Gateway-Gerät über das erste Kommunikationsnetzwerk;
Durchführen einer Suche nach einer IP-Adresse für die bestimmte Netzwerkeinheit über das Gateway-Gerät, um die IP-Adresse für die bestimmte Netzwerkeinheit festzulegen;
Senden der bestimmten IP-Adresse in einer Nachricht vom Gateway-Gerät zum ersten Gerät über das erste Kommunikationsnetzwerk;
Erzeugen eines primären Datenpakets in einem ersten Gerät als ein Datenpaket, das erste Netzwerkrouting-Informationen enthält, die dem auf der IP-Adresse basierenden Kommunikationsprotokoll entsprechen, und das die IP-Adresse für die bestimmte Netzwerkeinheit als Teil der ersten Netzwerkrouting-Informationen beinhaltet;
Hinzufügen von zweiten Netzwerkrouting-Informationen zum primären Datenpaket, wobei die zweiten Netzwerkrouting-Informationen dem ersten Kommunikationsprotokoll entsprechen, welches das nicht auf der IP-Adresse basierende Netzwerkrouting-Protokoll verwendet, und
Senden des eingekapselten primären Datenpakets vom ersten Gerät an ein zweites Gerät über das erste Kommunikationsnetzwerk unter Verwendung des ersten Netzwerkrouting-Protokolls.

89. Das Kommunikationsverfahren nach Anspruch 88, wobei das zweite Gerät das Gateway-Gerät ist.

90. Das Kommunikationsverfahren nach Anspruch 88, einschließlich des Routens des primären Datenpakets vom zweiten Gerät an die IP-Adresse der Netzwerkeinheit über das zweite Kommunikationsnetzwerk unter Verwendung der ersten Netzwerkrouting-Informationen des primären Datenpakets.

91. Das Kommunikationsverfahren aus Anspruch 88, wobei das Senden der Nachricht vom ersten Gerät an das Gateway-Gerät mithilfe des ersten Netzwerkrouting-Protokolls die Identifizierung der Netzwerkeinheit unter Verwendung eines Domain-Namens beinhaltet.

92. Das Kommunikationsverfahren aus Anspruch 88, wobei das Senden der Nachricht vom ersten Gerät an das Gateway-Gerät das Senden der Nachricht mit dem Einfügen eines Datenframes umfasst, der Netzwerkrouting-Informationen entsprechend dem zweiten Netzwerkrouting-Protokoll enthält, eingekapselt in Netzwerkrouting-Informationen entsprechend dem ersten Netzwerkrouting-Protokoll.

93. Ein greifbarer computerlesbarer Speicher, der elektrisch mit einem Prozessor eines Prozesssteuerungsgerätes verbunden ist; das Speichern von Befehlen darauf, deren Ausführung durch den Prozessor:
die Übertragung von Prozesssteuerungsdaten an eine Anwendung bewirkt, die auf dem Prozesssteuerungsgerät ausgeführt wird, wobei die Übertragung von Prozesssteuerungsdaten als Antwort auf den Empfang einer Anforderung von Prozesssteuerungsdaten durch die Anwendung erfolgt;
als Antwort auf den Empfang von Payload-Daten von der Anwendung einen Datenframe erzeugt, wobei das Erzeugen des Datenframes die Einkapselung der Payload-Daten in die ersten Routing-Informationen umfasst, die einem ersten Netzwerkprotokoll entsprechen, und weiteres Framing der eingekapselten Payload-Daten in zweite Routing-Informationen, die einem zweiten Netzwerkprotokoll entsprechen, wobei die ersten Routing-Informationen eine Zieladresse einer Empfänger-Anwendung beinhalten und die zweiten Routing-Informationen eine Routendefinition beinhalten, und
die Übertragung der Datenframes über den Kommunikationskanal bewirkt, der gemäß dem ersten Kommunikationsprotokoll funktioniert.

94. Der greifbare computerlesbare Speicher nach Anspruch 93, der Anweisungen umfasst, die, wenn sie durch den Prozessor als Antwort auf den Empfang eines Datenpakets über den Kommunikationskanal ausgeführt werden, feststellen, ob das Datenpaket in Routing-Informationen eingekapselt ist, die dem ersten Netzwerkprotokoll entsprechen, wobei das Feststellen, ob das Datenpaket in Routing-Informationen eingekapselt ist, die dem ersten Netzwerkprotokoll entsprechen, beinhaltet, dass ein Hinweis in den Routing-Informationen erkannt wird, die dem zweiten Netzwerkprotokoll entsprechen, wobei die Routing-Informationen, die dem zweiten Netzwerkprotokoll entsprechen, die Routing-Informationen einschließen, die dem ersten Netzwerkprotokoll entsprechen.

Es folgen 16 Seiten Zeichnungen

Anhängende Zeichnungen

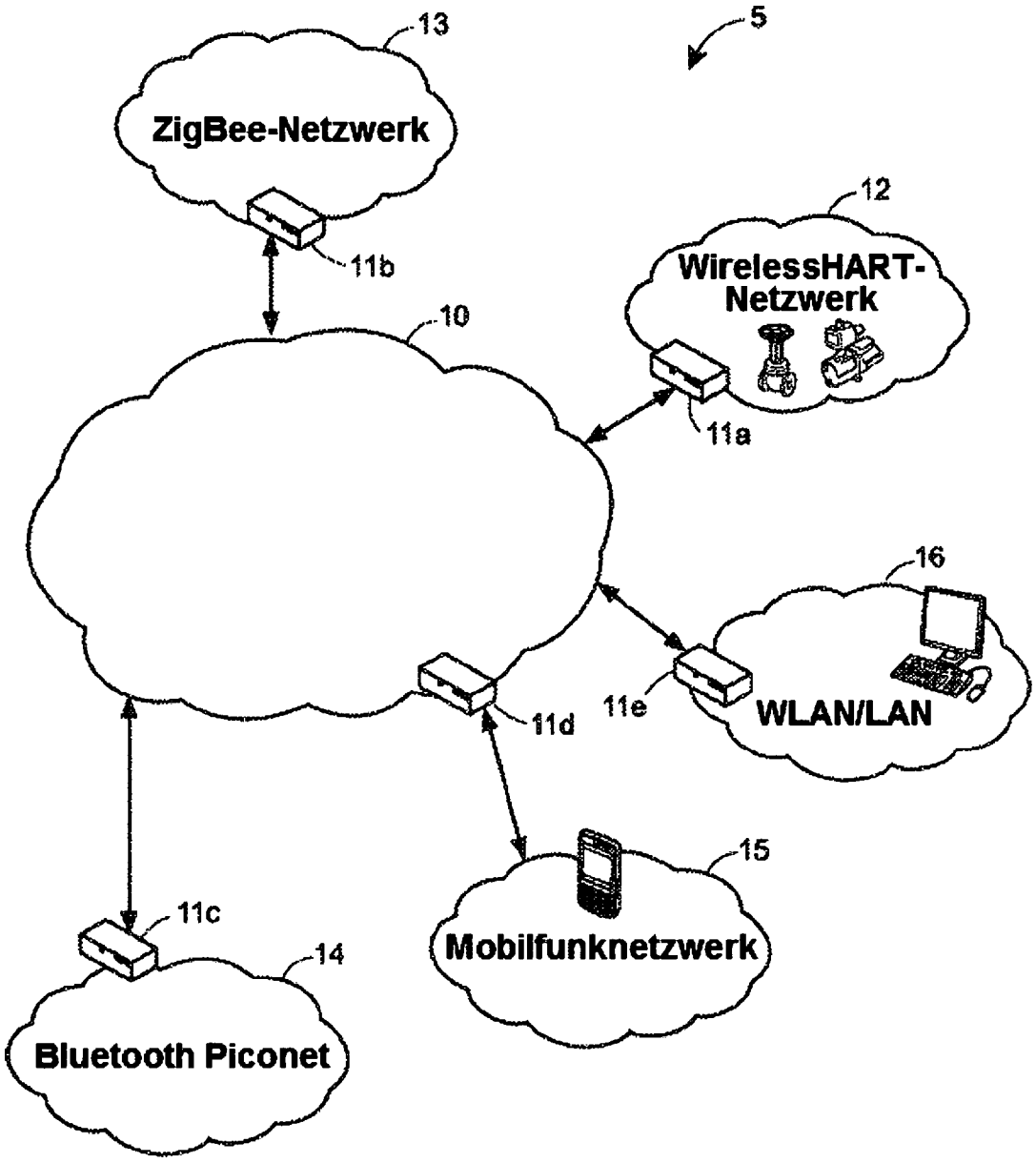


FIG. 1

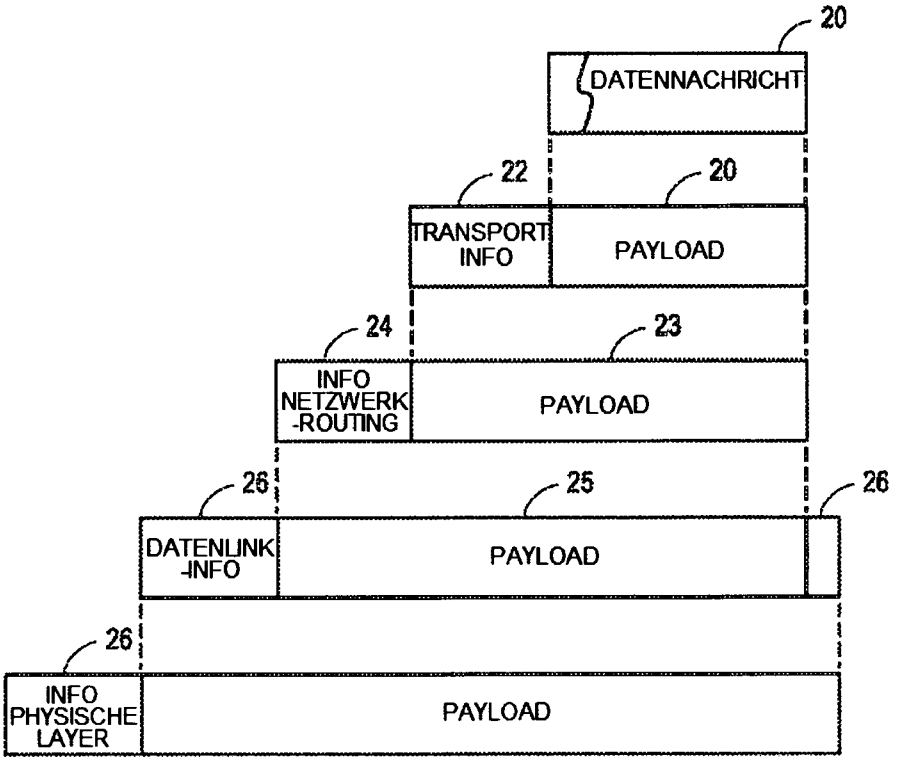

20
DATENNACHRICHT
22
20
TRANSPORT INFO
PAYLOAD
24
23
INFO NETZWERK -ROUTING
PAYLOAD
26
25
26
DATENLINK -INFO
PAYLOAD
26
INFO PHYSISCHE LAYER
PAYLOAD

FIG. 2A

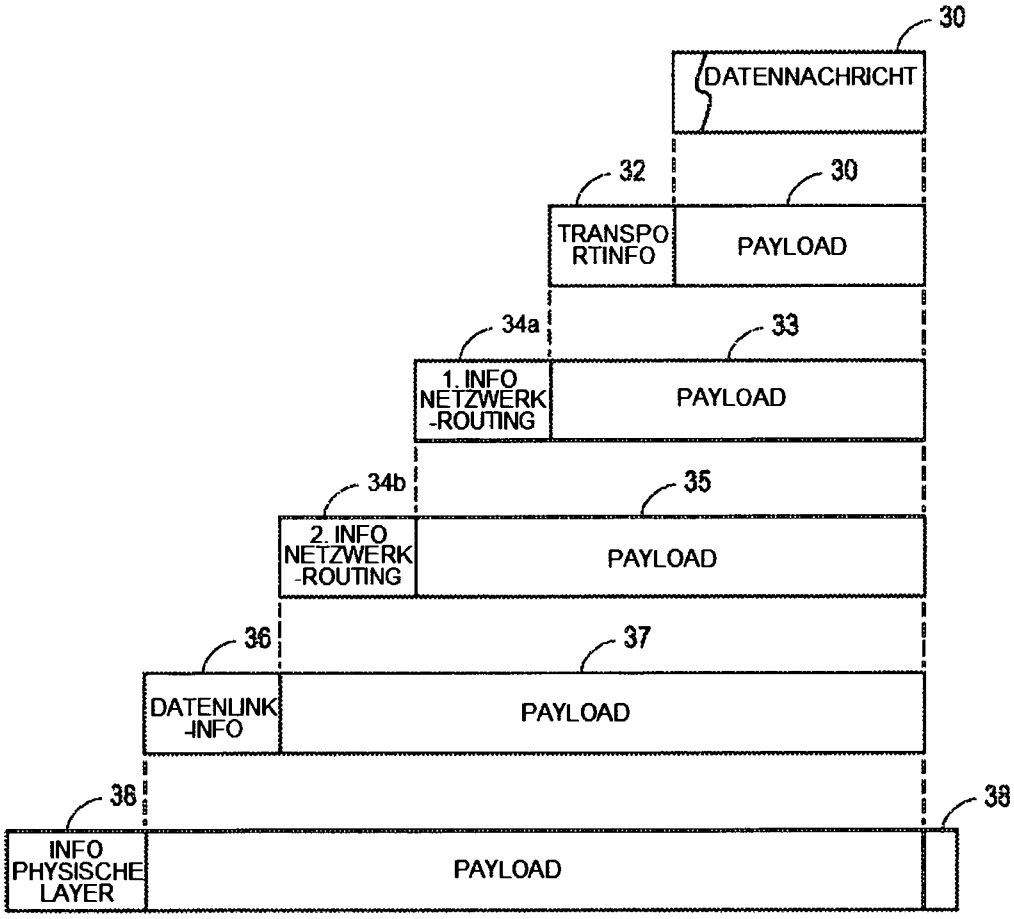

30
DATENNACHRICHT
32
30
TRANSPO RTINFO
PAYLOAD
34a
33
1. INFO NETZWERK -ROUTING
PAYLOAD
34b
35
2. INFO NETZWERK -ROUTING
PAYLOAD
36
37
DATENLINK -INFO
PAYLOAD
38
38
INFO PHYSISCHE LAYER
PAYLOAD

FIG. 2B

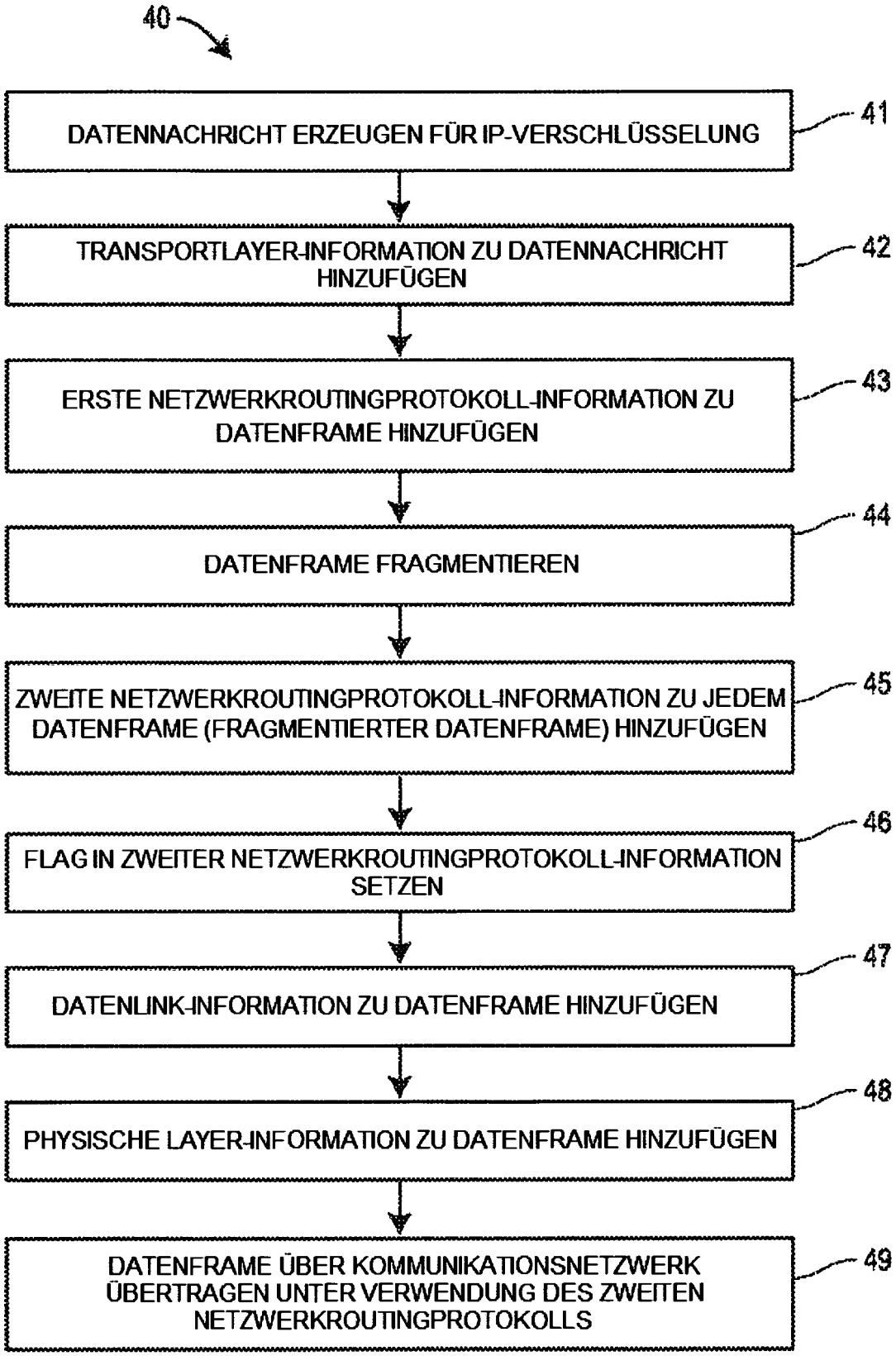

40
DATENNACHRICHT ERZEUGEN FÜR IP-VERSCHLÜSSELUNG
41
TRANSPORTLAYER-INFORMATION ZU DATENNACHRICHT HINZUFÜGEN
42
ERSTE NETZWERKROUTINGPROTOKOLL-INFORMATION ZU DATENFRAME HINZUFÜGEN
43
DATENFRAME FRAGMENTIEREN
44
ZWEITE NETZWERKROUTINGPROTOKOLL-INFORMATION ZU JEDEM DATENFRAME (FRAGMENTIERTER DATENFRAME) HINZUFÜGEN
45
FLAG IN ZWEITER NETZWERKROUTINGPROTOKOLL-INFORMATION SETZEN
46
DATENLINK-INFORMATION ZU DATENFRAME HINZUFÜGEN
47
PHYSISCHE LAYER-INFORMATION ZU DATENFRAME HINZUFÜGEN
48
DATENFRAME ÜBER KOMMUNIKATIONSNETZWERK ÜBERTRAGEN UNTER VERWENDUNG DES ZWEITEN NETZWERKROUTINGPROTOKOLLS
49

FIG. 3

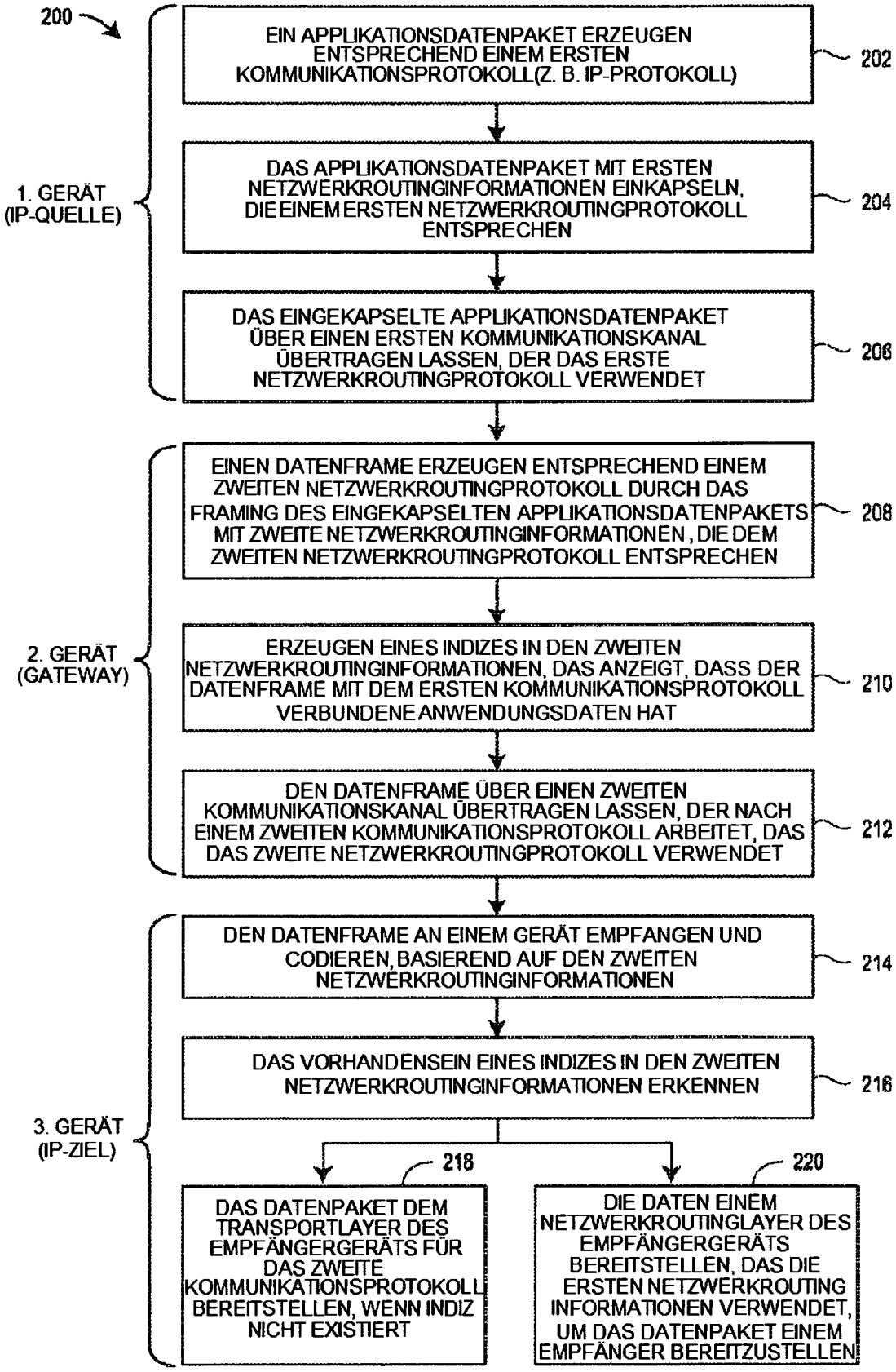

200
EIN APPLIKATIONSDATENPAKET ERZEUGEN ENTSPRECHEND EINEM ERSTEN KOMMUNIKATIONSPROTOKOLL(Z. B. IP-PROTOKOLL)
202
DAS APPLIKATIONSDATENPAKET MIT ERSTEN NETZWERKROUTINGINFORMATIONEN EINKAPSELN, DIE EINEM ERSTEN NETZWERKROUTINGPROTOKOLL ENTSPRECHEN
204
DAS EINGEKAPSELTE APPLIKATIONSDATENPAKET ÜBER EINEN ERSTEN KOMMUNIKATIONSKANAL ÜBERTRAGEN LASSEN, DER DAS ERSTE NETZWERKROUTINGPROTOKOLL VERWENDET
206
1. GERÄT (IP-QUELLE)
EINEN DATENFRAME ERZEUGEN ENTSPRECHEND EINEM ZWEITEN NETZWERKROUTINGPROTOKOLL DURCH DAS FRAMING DES EINGEKAPSELTEN APPLIKATIONSDATENPAKETS MIT ZWEITE NETZWERKROUTINGINFORMATIONEN, DIE DEM ZWEITEN NETZWERKROUTINGPROTOKOLL ENTSPRECHEN
208
ERZEUGEN EINES INDIZES IN DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN, DAS ANZEIGT, DASS DER DATENFRAME MIT DEM ERSTEN KOMMUNIKATIONSPROTOKOLL VERBUNDENE ANWENDUNGSDATEN HAT
210
DEN DATENFRAME ÜBER EINEN ZWEITEN KOMMUNIKATIONSKANAL ÜBERTRAGEN LASSEN, DER NACH EINEM ZWEITEN KOMMUNIKATIONSPROTOKOLL ARBEITET, DAS DAS ZWEITE NETZWERKROUTINGPROTOKOLL VERWENDET
212
2. GERÄT (GATEWAY)
DEN DATENFRAME AN EINEM GERÄT EMPFANGEN UND CODIEREN, BASIEREND AUF DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN
214
DAS VORHANDENSEIN EINES INDIZES IN DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN ERKENNEN
216
3. GERÄT (IP-ZIEL)
218
DAS DATENPAKET DEM TRANSPORTLAYER DES EMPFÄNGERGERÄTS FÜR DAS ZWEITE KOMMUNIKATIONSPROTOKOLL BEREITSTELLEN, WENN INDIZ NICHT EXISTIERT
220
DIE DATEN EINEM NETZWERKROUTINGLAYER DES EMPFÄNGERGERÄTS BEREITSTELLEN, DAS DIE ERSTEN NETZWERKROUTING INFORMATIONEN VERWENDET, UM DAS DATENPAKET EINEM EMPFÄNGER BEREITZUSTELLEN

FIG. 4A

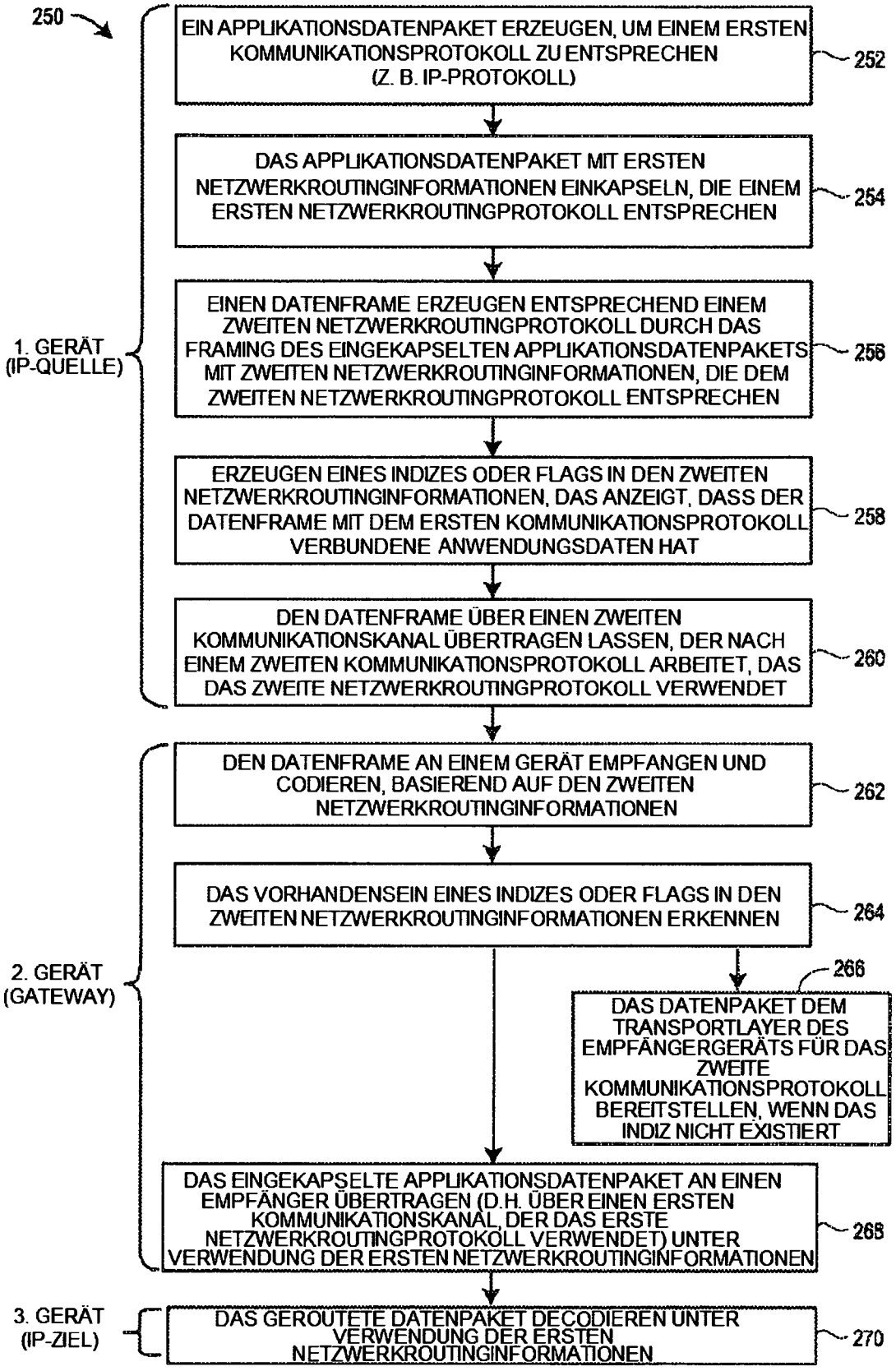

250
EIN APPLIKATIONSDATENPAKET ERZEUGEN, UM EINEM ERSTEN KOMMUNIKATIONSPROTOKOLL ZU ENTSPRECHEN (Z. B. IP-PROTOKOLL)
252
DAS APPLIKATIONSDATENPAKET MIT ERSTEN NETZWERKROUTINGINFORMATIONEN EINKAPSELN, DIE EINEM ERSTEN NETZWERKROUTINGPROTOKOLL ENTSPRECHEN
254
EINEN DATENFRAME ERZEUGEN ENTSPRECHEND EINEM ZWEITEN NETZWERKROUTINGPROTOKOLL DURCH DAS FRAMING DES EINGEKAPSELTEN APPLIKATIONSDATENPAKETS MIT ZWEITEN NETZWERKROUTINGINFORMATIONEN, DIE DEM ZWEITEN NETZWERKROUTINGPROTOKOLL ENTSPRECHEN
256
1. GERÄT (IP-QUELLE)
ERZEUGEN EINES INDIZES ODER FLAGS IN DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN, DAS ANZEIGT, DASS DER DATENFRAME MIT DEM ERSTEN KOMMUNIKATIONSPROTOKOLL VERBUNDENE ANWENDUNGSDATEN HAT
258
DEN DATENFRAME ÜBER EINEN ZWEITEN KOMMUNIKATIONSKANAL ÜBERTRAGEN LASSEN, DER NACH EINEM ZWEITEN KOMMUNIKATIONSPROTOKOLL ARBEITET, DAS DAS ZWEITE NETZWERKROUTINGPROTOKOLL VERWENDET
260
DEN DATENFRAME AN EINEM GERÄT EMPFANGEN UND CODIEREN, BASIEREND AUF DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN
262
DAS VORHANDENSEIN EINES INDIZES ODER FLAGS IN DEN ZWEITEN NETZWERKROUTINGINFORMATIONEN ERKENNEN
264
2. GERÄT (GATEWAY)
266
DAS DATENPAKET DEM TRANSPORTLAYER DES EMPFÄNGERGERÄTS FÜR DAS ZWEITE KOMMUNIKATIONSPROTOKOLL BEREITSTELLEN, WENN DAS INDIZ NICHT EXISTIERT
DAS EINGEKAPSELTE APPLIKATIONSDATENPAKET AN EINEN EMPFÄNGER ÜBERTRAGEN (D.H. ÜBER EINEN ERSTEN KOMMUNIKATIONSKANAL, DER DAS ERSTE NETZWERKROUTINGPROTOKOLL VERWENDET) UNTER VERWENDUNG DER ERSTEN NETZWERKROUTINGINFORMATIONEN
268
3. GERÄT (IP-ZIEL)
DAS GEROUTETE DATENPAKET DECODIEREN UNTER VERWENDUNG DER ERSTEN NETZWERKROUTINGINFORMATIONEN
270

FIG. 4B

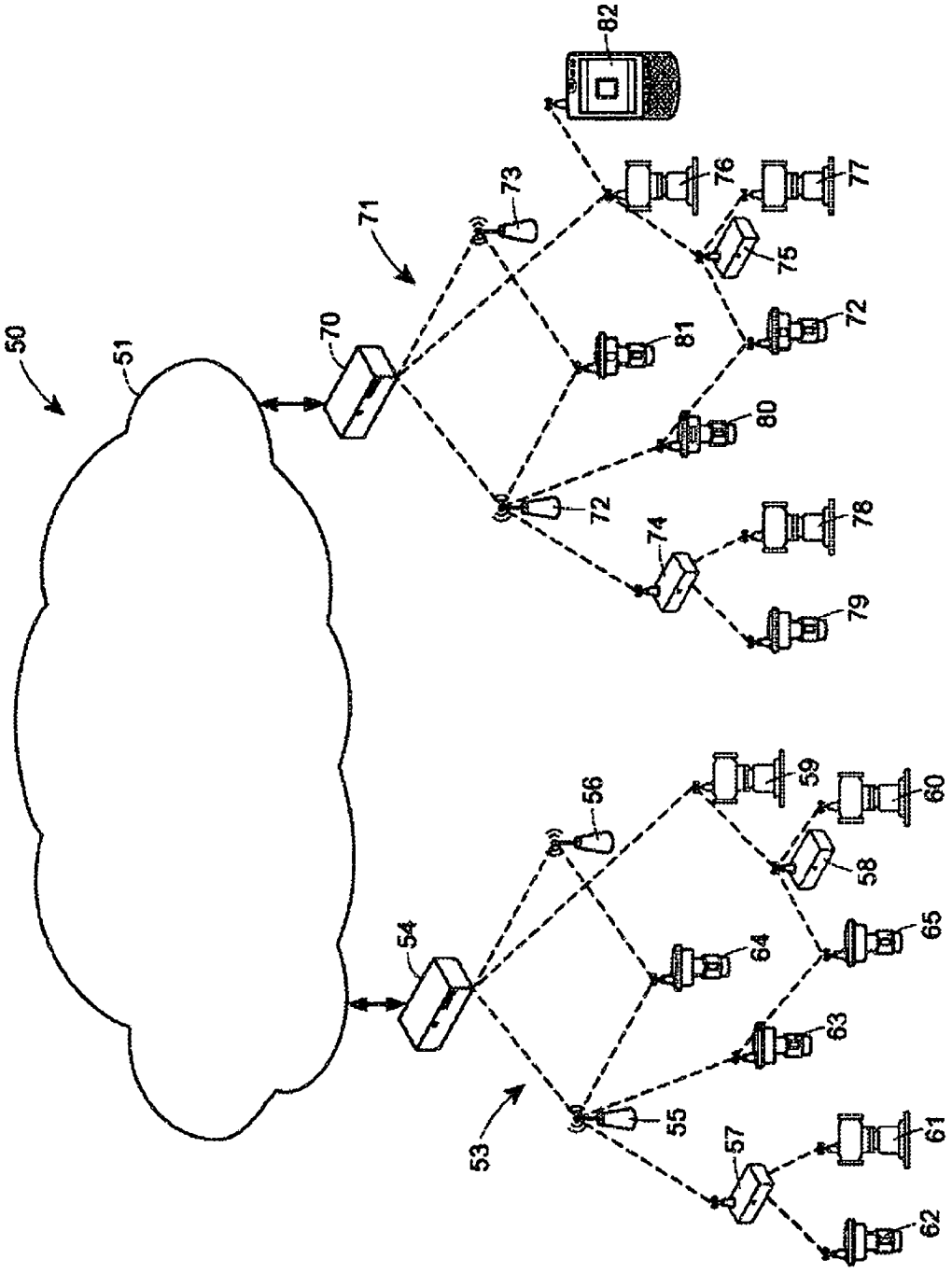

50
51
70
71
72
73
74
75
76
77
78
79
80
81
82
53
54
55
56
57
58
59
60
61
62
63
64
65

FIG. 5

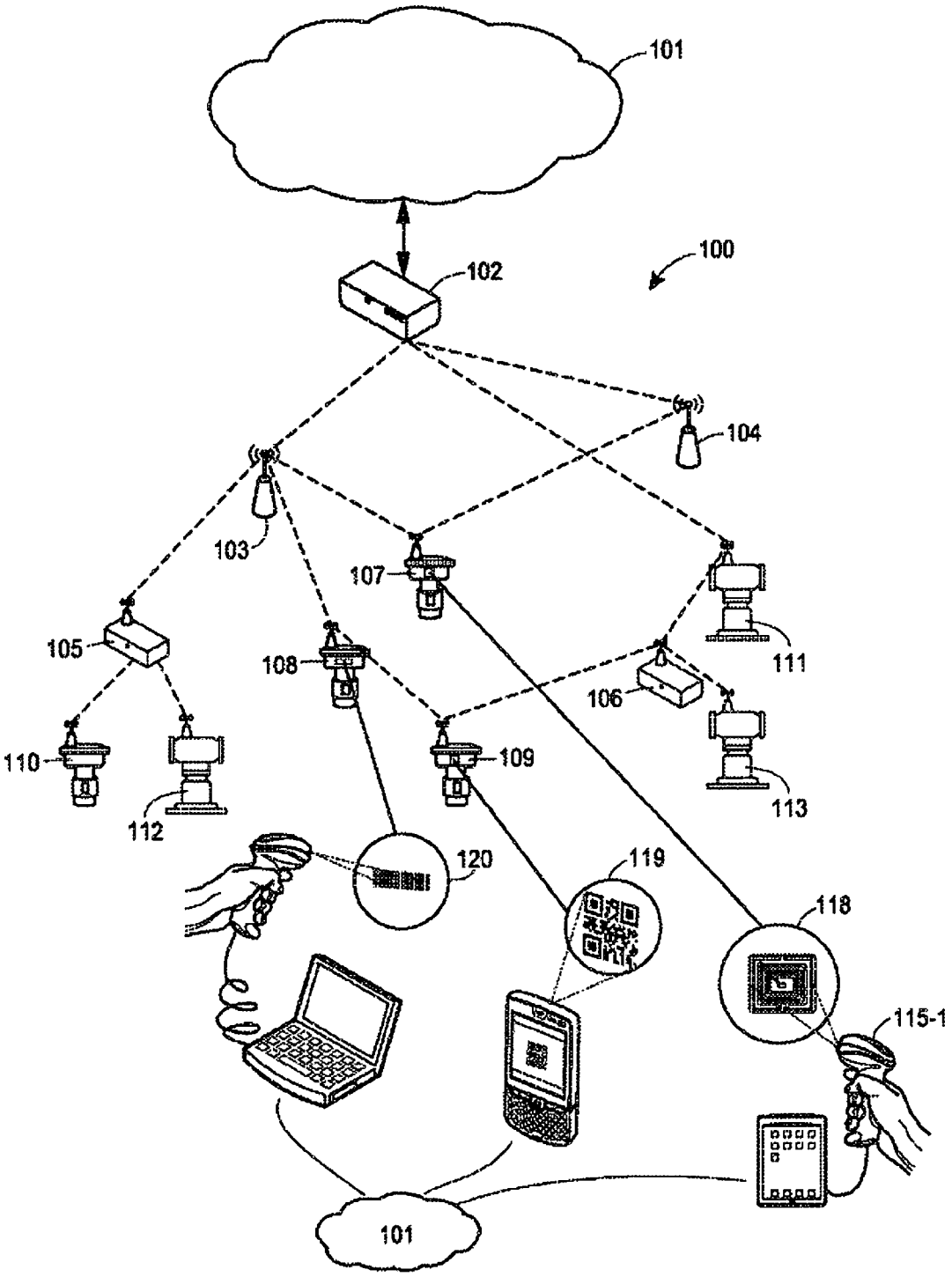

101
100
102
104
103
107
105
108
111
106
110
109
113
112
120
119
118
115-1
101

FIG. 6

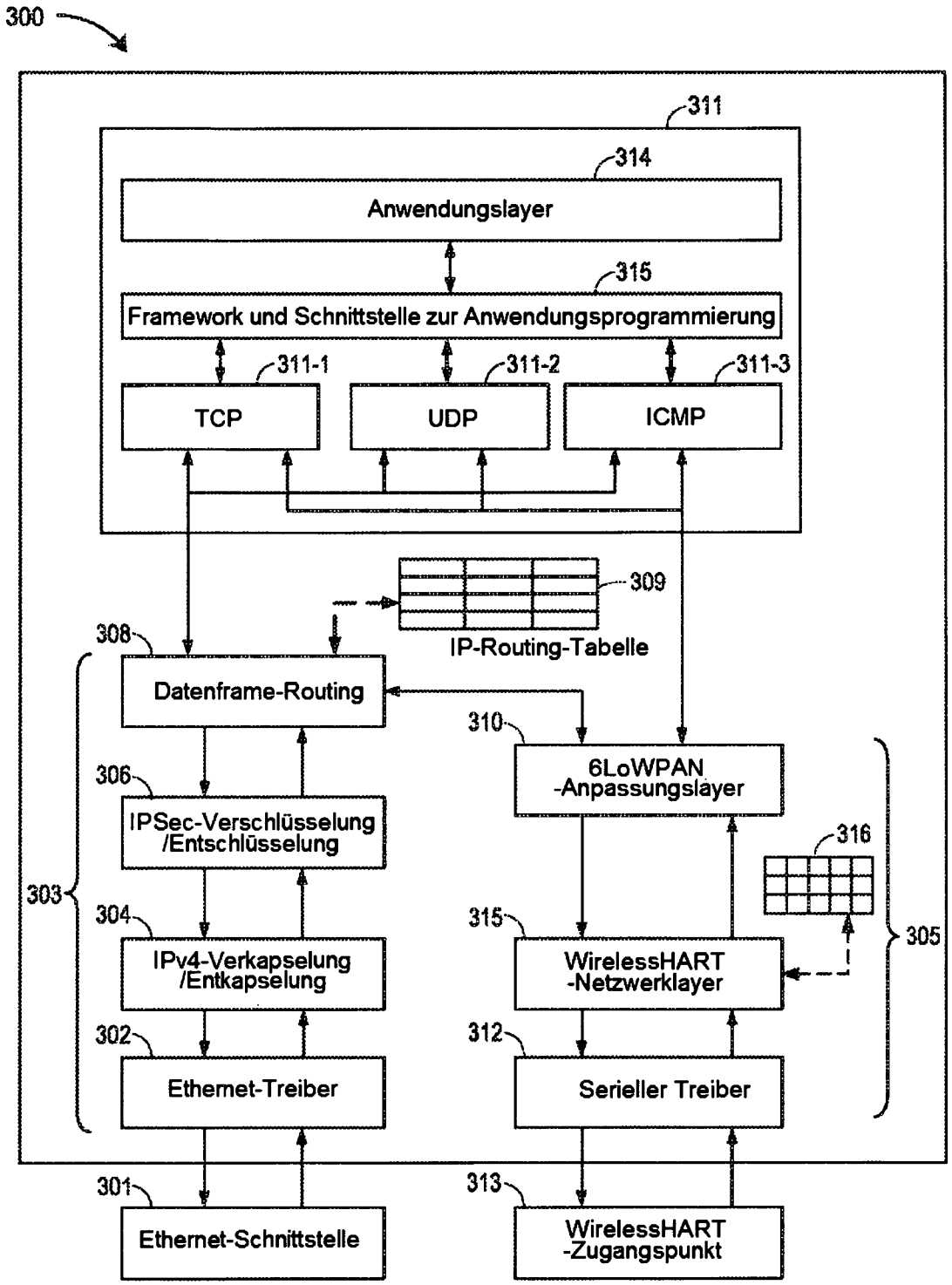

300
311
314
Anwendungslayer
315
Framework und Schnittstelle zur Anwendungsprogrammierung
311-1
311-2
311-3
TCP
UDP
ICMP
309
IP-Routing-Tabelle
308
Datenframe-Routing
310
6LoWPAN
-Anpassungslayer
306
IPSec-Verschlüsselung
/Entschlüsselung
316
303
304
315
IPv4-Verkapselung
/Entkapselung
WirelessHART
-Netzwerklayer
305
302
312
Ethernet-Treiber
Serieller Treiber
301
313
Ethernet-Schnittstelle
WirelessHART
-Zugangspunkt

FIG. 7

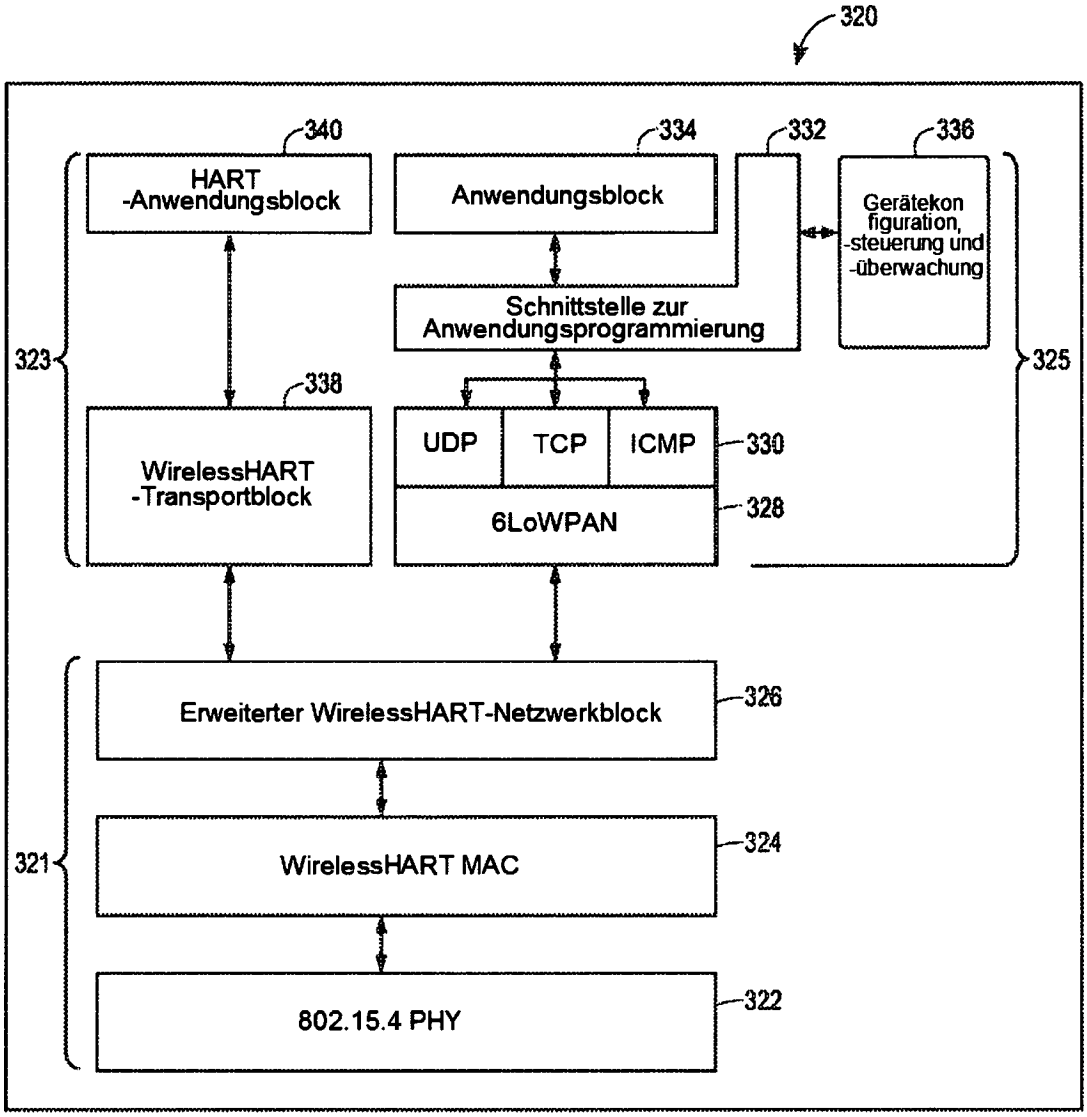

320
340
HART
-Anwendungsblock
334
Anwendungsblock
332
336
Gerätekon
figuration,
-steuerung und
-überwachung
Schnittstelle zur
Anwendungsprogrammierung
323
325
338
WirelessHART
-Transportblock
UDP
TCP
ICMP
330
6LoWPAN
328
Erweiterter WirelessHART-Netzwerkblock
326
321
WirelessHART MAC
324
802.15.4 PHY
322

FIG. 8

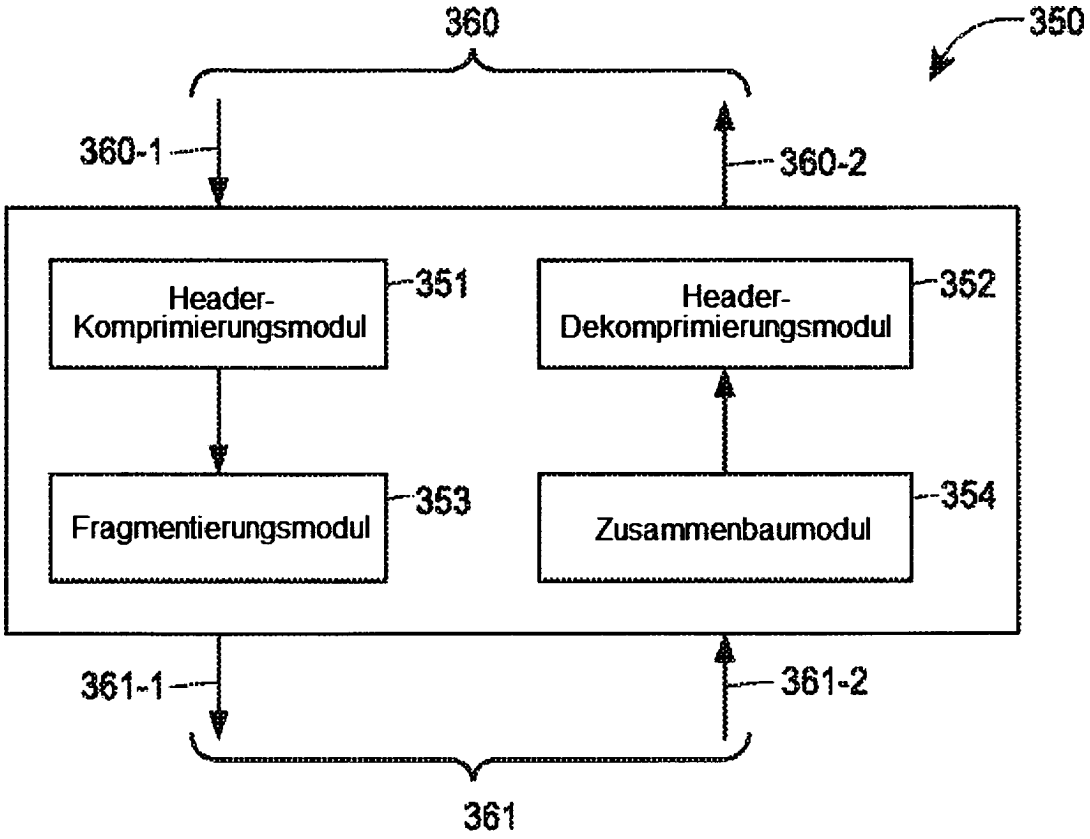

360
350
360-1
360-2
Header-
Komprimierungsmodul
351
Header-
Dekomprimierungsmodul
352
Fragmentierungsmodul
353
Zusammenbaumodul
354
361-1
361-2
361

FIG. 9

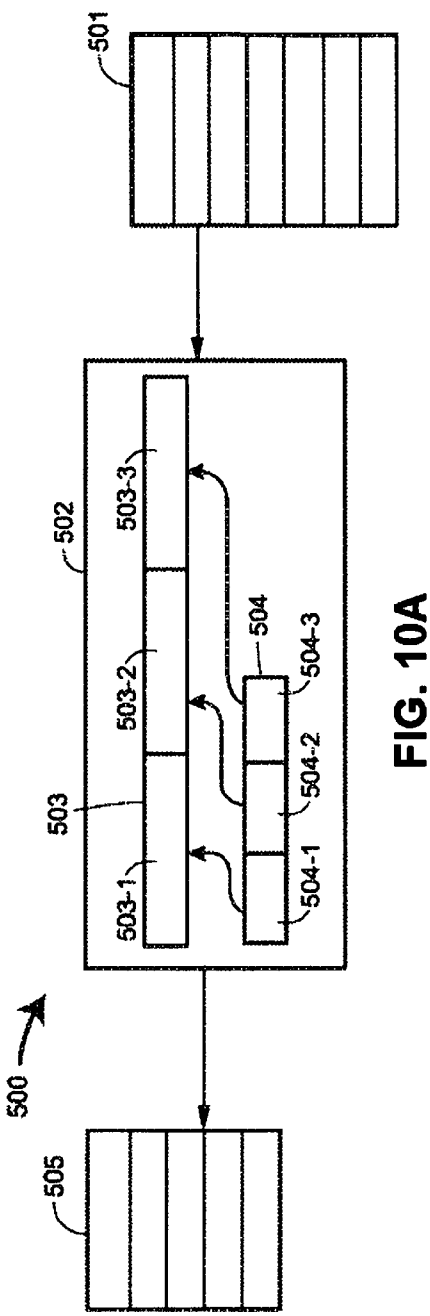


FIG. 10A

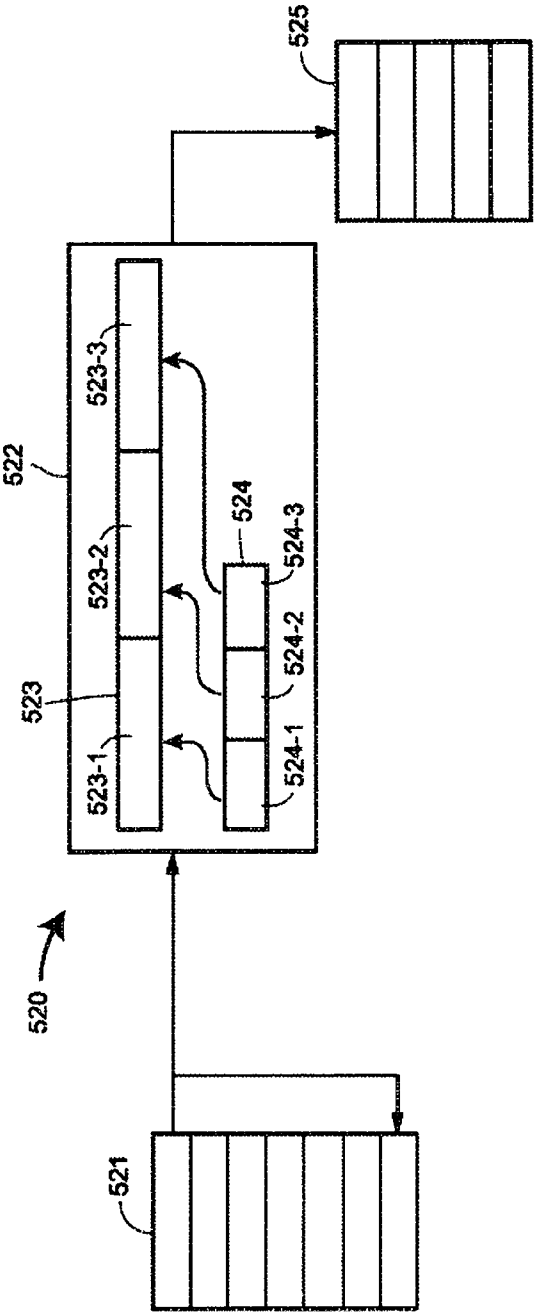


FIG. 10B

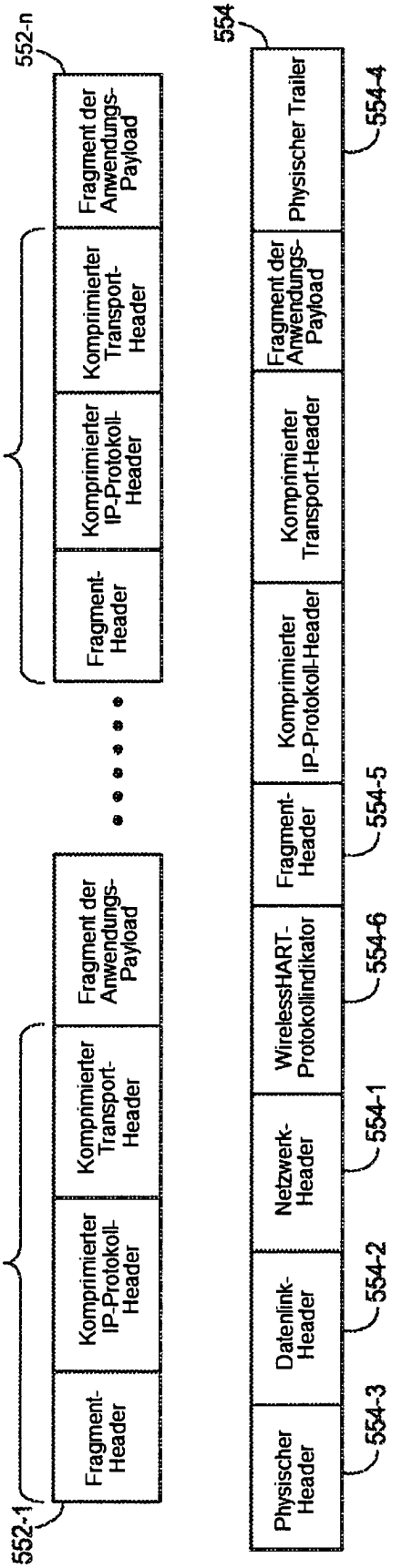


FIG. 11A

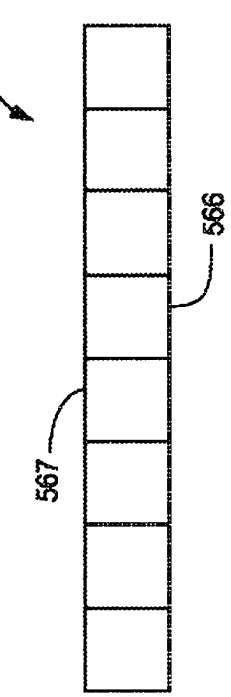


FIG. 11C

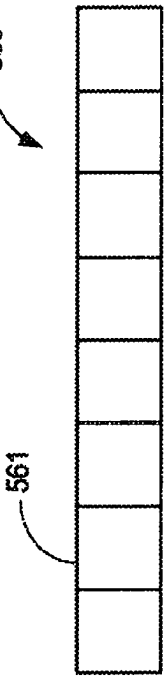


FIG. 11B

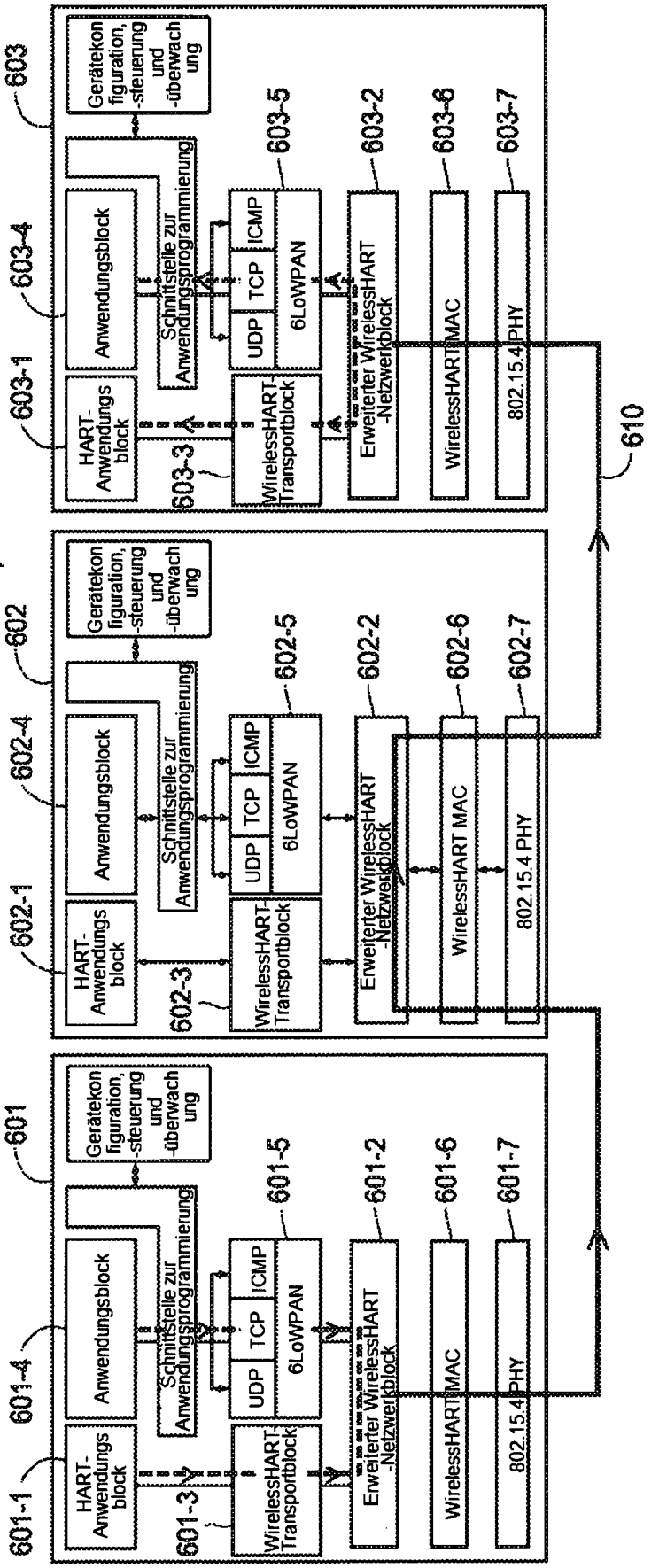

600
601
601-1
601-4
601-3
601-5
601-2
601-6
601-7
602
602-1
602-4
602-3
602-5
602-2
602-6
602-7
603
603-1
603-4
603-3
603-5
603-2
603-6
603-7
610
HART-Anwendungsblock
Anwendungsblock
Schnittstelle zur Anwendungsprogrammierung
Gerätekonfiguration, -steuerung und -überwachung
WirelessHART-Transportblock
UDP
TCP
ICMP
6LoWPAN
Erweiterter WirelessHART-Netzwerkblock
WirelessHART MAC
802.15.4 PHY

FIG. 12

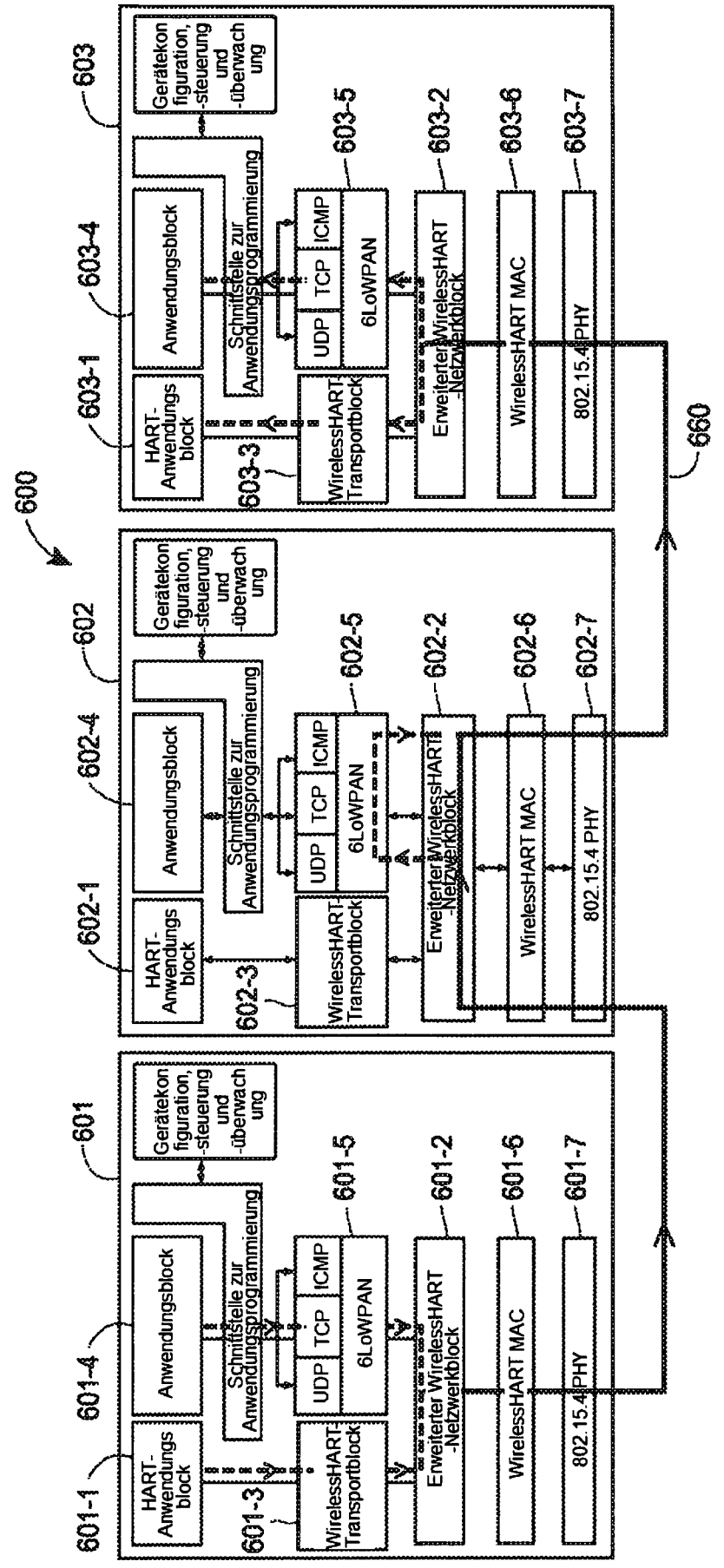

600
601
602
603
601-1
602-1
603-1
601-4
602-4
603-4
HART-Anwendungsblock
Anwendungsblock
Schnittstelle zur Anwendungsprogrammierung
Gerätekonfiguration, -steuerung und -überwachung
601-3
602-3
603-3
WirelessHART Transportblock
UDP
TCP
ICMP
6LoWPAN
601-5
602-5
603-5
Erweiterter WirelessHART -Netzwerkblock
601-2
602-2
603-2
WirelessHART MAC
601-6
602-6
603-6
802.15.4 PHY
601-7
602-7
603-7
660

FIG. 13

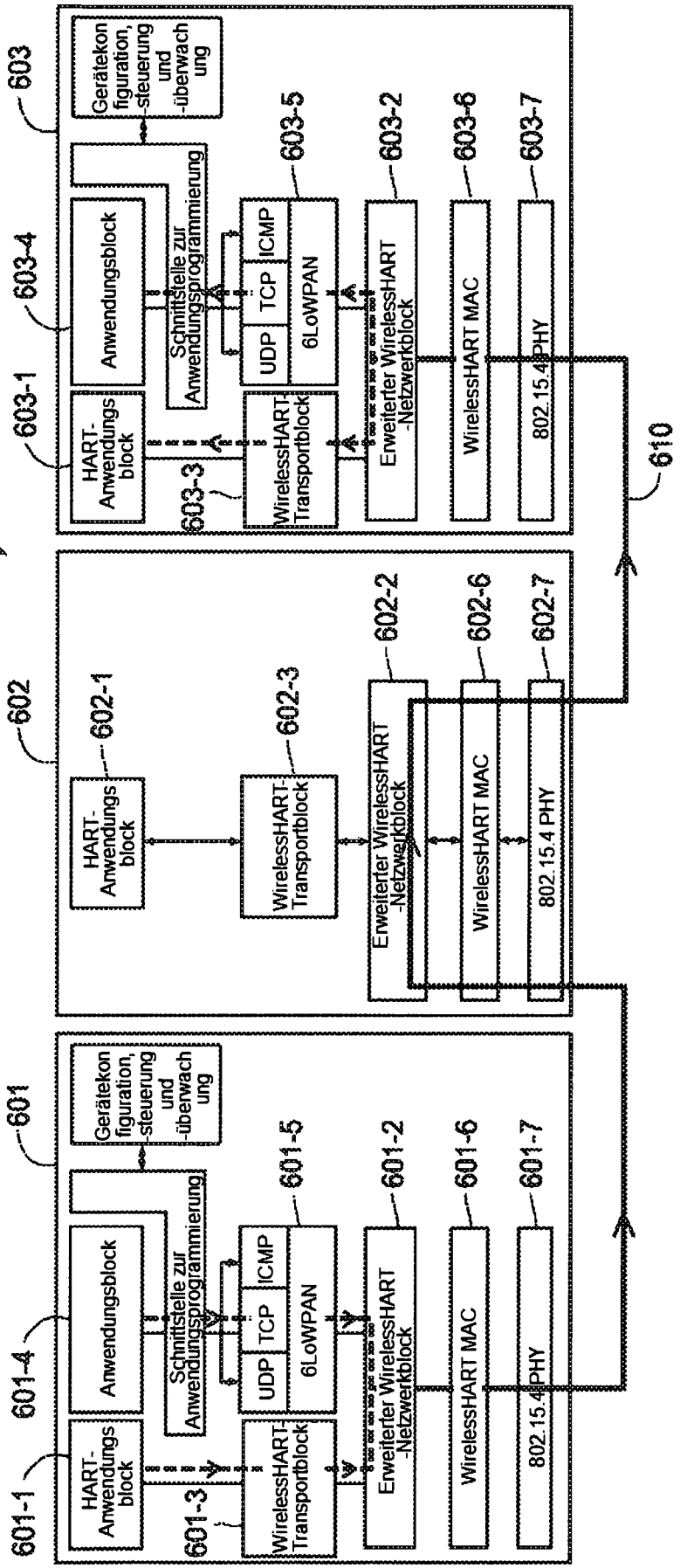

600
601
601-1
601-4
HART-Anwendungsblock
Anwendungsblock
601-3
Schnittstelle zur Anwendungsprogrammierung
Gerätekonfiguration, -steuerung und -überwachung
WirelessHART-Transportblock
UDP
TCP
ICMP
6LoWPAN
601-5
Erweiterter WirelessHART-Netzwerkblock
601-2
WirelessHART MAC
601-6
802.15.4 PHY
601-7
602
602-1
602-3
602-2
602-6
602-7
603
603-1
603-4
603-5
603-2
603-6
603-7
610

FIG. 14

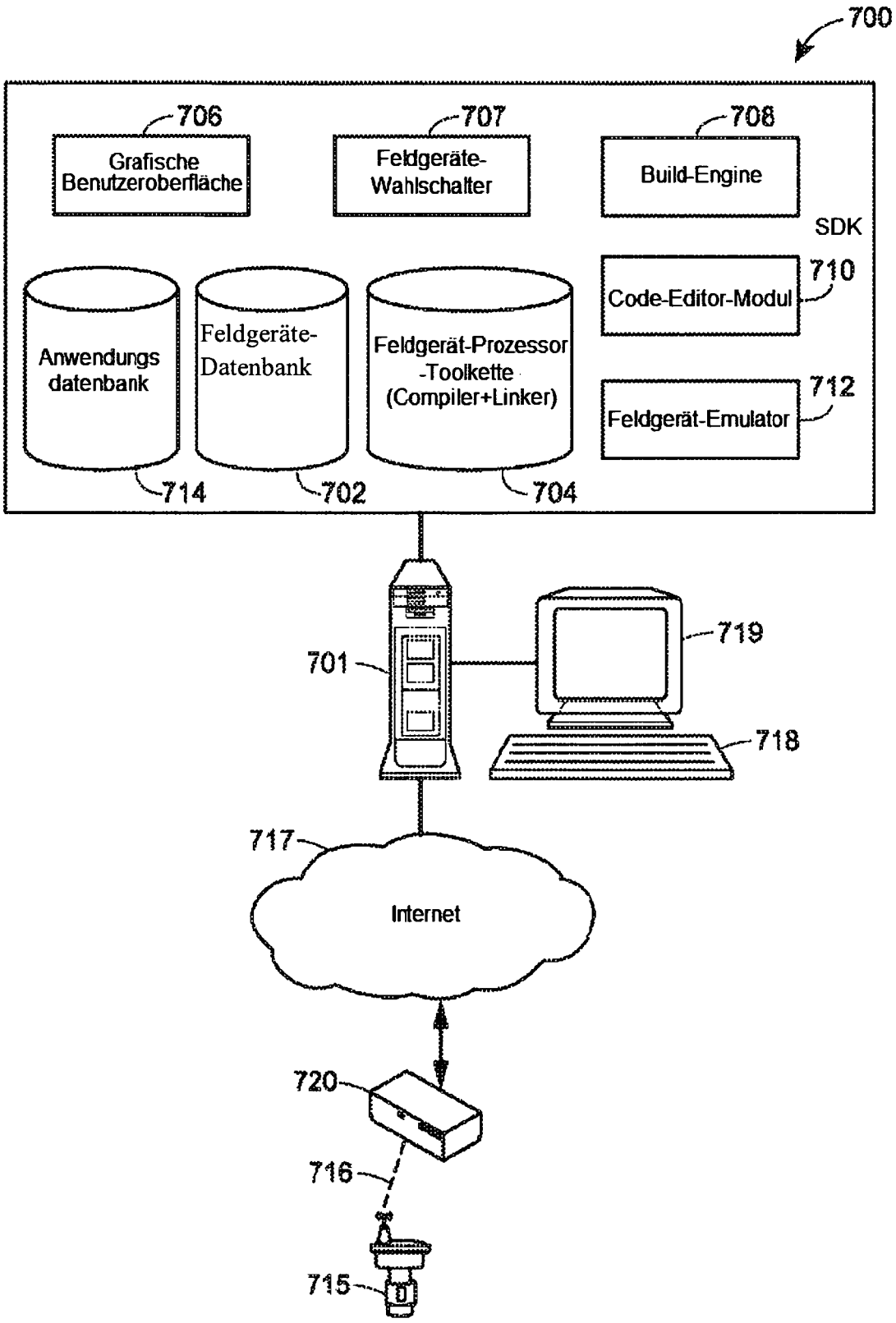

700
706
Grafische Benutzeroberfläche
707
Feldgeräte-Wahlschalter
708
Build-Engine
SDK
710
Code-Editor-Modul
712
Feldgerät-Emulator
Anwendungs datenbank
Feldgeräte-Datenbank
Feldgerät-Prozessor -Toolkette (Compiler+Linker)
714
702
704
701
719
718
717
Internet
720
716
715

FIG. 15