



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 602 13 297 T2** 2007.08.23

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 233 591 B1**

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

(21) Deutsches Aktenzeichen: **602 13 297.5**

(96) Europäisches Aktenzeichen: **02 002 194.5**

(96) Europäischer Anmeldetag: **29.01.2002**

(97) Erstveröffentlichung durch das EPA: **21.08.2002**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **26.07.2006**

(47) Veröffentlichungstag im Patentblatt: **23.08.2007**

(30) Unionspriorität:

681195 16.02.2001 US

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE, TR**

(73) Patentinhaber:

Microsoft Corp., Redmond, Wash., US

(72) Erfinder:

Gemmell, David J., Danville, California 94526, US

(74) Vertreter:

**Grünecker, Kinkeldey, Stockmair &
Schwanhäusser, 80538 München**

(54) Bezeichnung: **Multimediatatenübertragung durch vorwärtsgerichtetes Streamen**

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

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

Beschreibung

Technisches Gebiet

[0001] Die Erfindung bezieht sich auf das Rendern von Streaming-Medien, und spezieller auf ein System und einen Vorgang zum Erlangen von Versionen eines Audio- und/oder Video-Programms mit stufenweise höherer Qualität über ein Client-Server basiertes Netzwerk.

Hintergrundtechnik:

[0002] Audio- und Videoinformation wird im Allgemeinen von einem Server zu einem Client über eine Netzwerkverbindung, insbesondere über das Internet, gesendet. Zum Beispiel enthalten viele Nachrichten- und Sport-Websites im Internet kurze Videoclips, auf welche durch einen Benutzer zugegriffen werden kann. Eine typische Art und Weise, wie dies passiert, ist, dass ein Benutzer die Daten zugehörig zu dem gewünschten Clip herunterlädt. Dies wird dadurch erreicht, dass ein Clientcomputer zugehörig zu dem Benutzer eine Anfrage nach den Daten von dem Server, auf dem die Daten liegen, macht. Der Server übermittelt dann die angefragten Daten zu dem Client über das Netzwerk. Sobald alle Daten durch den Client empfangen worden sind, rendert der Clientcomputer sie und präsentiert sie dem Benutzer in gewohnter Weise. Wenn Daten über das Netzwerk, und im Besonderen über das Internet, übertragen werden, können die Kanäle zwischen dem Server und dem Client jedoch bezüglich der Kapazität drastisch schwanken, oft in zwei oder drei Größenordnungen. Diese Kapazitätsunterschiede existieren, weil die Datenübertragungsraten zugehörig zu den Verbindungen zu einem bestimmten Client variieren können (z. B. Telefonleitungskapazität, LAN- und/der Modem-Geschwindigkeiten). Diese Heterogenität der Kapazität kann Probleme verursachen, besonders wenn Audio und Video mit hoher Qualität gewünscht ist. Zum Beispiel könnte das Herunterladen einer Version eines Videoclips mit hoher Qualität, und deshalb großer Bandbreite, von einer Website über das Internet bedeuten, für eine viel längere Dauer zu warten, als die Dauer des Clips selbst. Deshalb muss der Benutzer warten, um den Videoclip zu sehen, oft mit dem Ergebnis, dass der Benutzer frustriert ist. Des Weiteren kann der Benutzer nicht wissen, ob der Videoclip von Interesse ist, ohne ihn zu sehen, somit ist das Warten, um etwas herunterzuladen, das noch nicht einmal interessant sein kann, doppelt unattraktiv.

[0003] Das Problem mit dem Herunterladen kann durch das Verwenden einer Audio- und Videodatenübertragungsform, die als Echtzeit-Unicast-Multimedia-Präsentation bezeichnet wird, verhindert werden. Im Wesentlichen involviert dieses Schema das Streamen von Daten zugehörig zu einem angefragten Vi-

deoprogramm von dem Server zu dem Client über das Netzwerk. Während die Daten durch den Client empfangen werden, werden sie gerendert und dem Benutzer auf einer nahezu Echtzeitbasis präsentiert. Die zuvor erwähnten Bandbreitenbeschränkungen, die typisch für ein Netzwerk und speziell für das Internet sind, erzeugen jedoch auch Probleme für diesen Übertragungstyp. Zum Beispiel ist die typische Bandbreite, die auf einem Netzwerk, wie dem Internet, verfügbar ist, unzureichend, um das Streamen eines Farbvideos mit hoher Qualität zu erlauben. Deshalb kann ein bestimmter Client nicht die Bandbreite verfügbar haben, um die Übertragung mit der höchsten Qualität zu empfangen, die ein Server in der Lage ist, bereitzustellen.

[0004] Um dieses Bandbreitenproblem zu bewältigen, kann Audio- und Videoinformation über ein Schichtmodell (layered scheme) übermittelt werden. In einem Schichtmodell wird Audio- und Videoinformation in Layern bezüglich der Bedeutung codiert. Jeder dieser Layer wird in einem separaten Datenstrom übermittelt, welcher im Wesentlichen eine Sequenz von Paketen ist. Der Basislayer ist ein Informationsstrom, der die minimale Menge an Information für die geringste akzeptable Qualität enthält. Nachfolgende Layer verbessern die vorhergehenden Layer, aber wiederholen nicht die Daten, die in einem niedrigeren Layer enthalten sind. Um die höhere Qualität zu erlangen, muss ein Client die niedrigeren Layer zusätzlich zu den höheren Layern, die die gewünschte Qualität bereitstellen, empfangen. Deshalb sind die Layer dadurch hierarchisch, dass es mindestens einen Basislayer und einen oder mehrere zusätzliche Erweiterungslayer mit höherem Level gibt. Es kann tatsächlich etliche hierarchische Layer geben, die auf einem Basislayer aufbauen, und wobei jeder nachfolgende Layer abhängig von den Daten eines oder mehrerer Layer geringeren Levels ist, und diese Layer mit geringerem Level erweitert. Ein illustratives (aber vielleicht nicht realistisches) Beispiel eines geschichteten Videoprogramms würde einen Basislayer einschließen, der aus einem Schwarz-Weiß-Video von jedem ungerade nummerierten Videoframe besteht, einem zweiten Layer, der aus einem Schwarz-Weiß-Video von jedem gerade nummerierten Videoframe besteht und einem dritten Layer, der aus Farbinformationen für alle Frames besteht. Das Abspielen von nur dem ersten Layer würde ein Schwarz-Weiß-Video mit halber Frame-Rate geben (d.h. irgendwie ruckartig). Das Abspielen des ersten und zweiten Layer zusammen würde zu einem Schwarz-Weiß-Video mit voller Frame-Rate führen (glatter Bewegung). Das Abspielen aller drei Layer würde zu einem Farbvideo voller Framerate führen.

[0005] In einem Schichtmodell kann ein Client so viele Layer wie gewünscht abfragen, vorausgesetzt, die Gesamtbandbreite der Layer ist nicht größer als die auf dem Netzwerk verfügbare Bandbreite. Zum

Beispiel, wenn der Client mit dem Internet durch ein 28,8 Kbps-Modem verbunden ist, kann er praktikabel einen, zwei oder drei 8 Kbps-Videolayer abonnieren. Wenn er mehr als drei solcher Layer abonniert, wird es bestimmt zu einer Stauung führen und viele Pakete würden zufälligerweise fallengelassen, was zu einer geringen Videoqualität führt. Durch das Beobachten des Fallenlassens der Pakete kann die maximale Anzahl an Layern, die unterstützt werden kann, ermittelt werden.

[0006] Chow, R.K.Y et al, "Scalable Video Delivery to Unicast Handheld-based Clients", Conference article, IEEE, US, 5. September 2000, Seiten 93 bis 98 beziehen sich auf das Design und die Architektur eines Videolieferungs-Proxy-Gateway. Das Proxy-Gateway ist Teil eines skalierbaren Multicast-Videolieferungssystems, und seine Rolle ist es, den Videostream, der aufgeteilt und multicast ist, in einen einzelnen Unicaststream zu übersetzen, der in einer benutzerspezifischen Form für den Unicast-Client geliefert wird. Das skalierbare Lieferungssystem setzt einen geschichteten Wavelet-basierten Video-Codec ein, um einen skalierbaren Datenstrom zu erhalten, welcher unter Verwendung von Multicast geteilt und verteilt wird. Der Codec erlaubt es dem codierten Videostream in separate Layer geteilt zu werden. Diese Layer werden in Basis- und Erweiterungslayer klassifiziert. Der minimal notwendige Layer, um ein Video zu reproduzieren, ist der Basislayer, während das Hinzufügen von Erweiterungslayern die Qualität des Videos verbessert. Sowohl Basis- als auch Erweiterungslayer sind multicast als verschiedene Multicastgruppen. In dem beschriebenen Modell werden der Server und seine Clients entkoppelt, d. h. der Videoserver behält die Clients, die er bedient, nicht im Auge, und es ist der Client, der sich für die Kombination und Anzahl von Multicastgruppen entscheidet, die er abonniert. Des Weiteren schließt das Proxy-Gateway vier Hauptobjekte ein, den Multicastempfänger, Datenvorprozessor, Unicastsender und Clientmanagerobjekte. Der Datenvorprozessor wird mit den Aufgaben geladen, alle eingehenden Pakete in einem Hashpuffer zu speichern, die Pakete, wie durch den Client gefordert, in ein einzelnes Paket zu vereinen und die Speicherbereinigung periodisch auszuführen.

Kurzfassung der Erfindung

[0007] Die Aufgabe der Erfindung ist es, den Prozess des Herunterladens eines Audio-Video-Programms zu verbessern.

[0008] Diese Aufgabe wird durch die Erfindung, wie in den unabhängigen Ansprüchen beansprucht, gelöst. Bevorzugte Ausführungsformen werden durch die abhängigen Ansprüche definiert.

[0009] Die vorliegende Erfindung ist auf ein System

und einen Prozess zum Einsetzen eines Schicht-Unicast-Schemas (layered unicast scheme) gerichtet, das sowohl einem Benutzer erlaubt, die Verzögerung und ungenutzte Zeit in Zusammenhang mit dem Download eines Audio-Video-Programms (AV-Programms), von dem er oder sie anschließend herausfindet, dass es nicht von Interesse ist, zu vermeiden, als auch zur selben Zeit eine Programmversion mit höherer Qualität, in dem Fall, dass es für den Benutzer von Interesse ist, zu erlangen. Es wird bemerkt, dass es viele verschiedene Layertechniken gibt, die derzeit in Verwendung sind, und dass irgendeine davon in Verbindung mit der vorliegenden Erfindung verwendet werden kann.

[0010] Das Schicht-Unicast-Schema gemäß der vorliegenden Erfindung wird im Allgemeinen wie folgt erreicht. Als Antwort auf einen Benutzerbefehl, ein bestimmtes AV-Programm zu sehen, fragt der Benutzer bei dem Server, auf dem die zugehörigen geschichteten AV-Daten liegen, an, die Daten zu dem Client über das Netzwerk zu übermitteln. Dadurch spezifiziert der Client auch, wie viele aufeinanderfolgende Layer, beginnend mit dem Basislayer, übermittelt werden sollen. Diese Entscheidung basiert auf der aktuell verfügbaren Bandbreite der Netzwerkverbindung zwischen dem Client und dem Server. Auf das Empfangen der Clientanfrage hin übermittelt der Server die angefragten Daten über das Netzwerk an den Client. Der Client empfängt die Daten von dem Server, rendert sie, während er sie empfängt, und präsentiert sie dem Betrachter über konventionelle Verfahren. Der Client speichert die empfangenen Daten auch für eine zukünftige Verwendung zwischen.

[0011] Die gerenderten Daten, die dem Benutzer präsentiert werden, werden üblicherweise ziemlich rudimentär sein, weil typischerweise nur der Basislayer und vielleicht ein paar Erweiterungslayer aus den unteren Leveln aufgrund der typischen Bandbreitenbeschränkungen, wie z. B. in Verbindung mit dem Internet, angefragt werden können. Der Benutzer hat an diesem Punkt des Verfahrens drei Auswahlmöglichkeiten. Wenn das AV-Programm für den Benutzer nicht interessant ist, könnte der Benutzer nichts machen, und zusätzliche Daten von Erweiterungslayern würden nicht übermittelt. Alternativ könnte der Benutzer anweisen, dass der Empfang des eingehenden Datenstroms unterbrochen wird. Oder letztendlich könnte der Benutzer das AV-Programm nochmals anschauen und könnte interessiert genug sein, eine Programmversion mit höherer Qualität empfangen zu wollen. Dazu würde der Benutzer den Client anweisen, eine erweiterte Version des Programms bereitzustellen.

[0012] In dem Fall, wo der Benutzer eine erweiterte Version des AV-Programms anfragt, würde der Client wieder anfragen, dass der Server AV-Daten zugehörig zu dem gewünschten Programm bereitstellt, um

zu ihm über das Netzwerk übermittelt zu werden. Dieses Mal fragt der Client jedoch so viele der aufeinanderfolgenden Erweiterungslayer an, wie es angesichts der verfügbaren Bandbreite möglich ist, beginnend mit dem Layer des geringsten Levels, das in dem letzten Schritt nicht angefragt worden ist. Als Antwort darauf übermittelt der Server die angefragten Daten in der gleichen Weise, wie vorher beschrieben. Auf den Empfang der angefragten Daten hin, verbindet der Client sie mit den vorher gespeicherten Layerdaten mit geringerem Level unter Verwendung der Standardverfahren, die mit dem Typ des eingesetzten Layerschemas einsetzbar sind. Die kombinierten Daten werden dann verwendet, um das Programm zu rendern und es dem Benutzer zu präsentieren. Das gerenderte Programm wird eine Version mit höherer Qualität des vorher betrachteten Programms darstellen. Der Client speichert wieder die neuen Daten für eine zukünftige Verwendung zusammen mit den vorher gespeicherten Daten, zugehörig zu demselben Programm.

[0013] An diesem Punkt hat der Benutzer die gleichen drei Optionen, wie oben diskutiert, und der Vorgang wird, soweit erforderlich, wiederholt, abhängig von den Optionen, die der Benutzer gewählt hat. Zum Beispiel, wenn der Benutzer wünscht, dass ihm eine Version des AV-Programms mit noch höherer Qualität bereitgestellt wird, werden mehr Erweiterungslayer (angenommen, es sind welche vom Server verfügbar) durch den Client angefragt und auf den Empfang hin, wie oben beschrieben, verarbeitet.

[0014] Es wird darauf hingewiesen, dass das/der vorangegangene Schicht-Unicast-System und -Vorgang auf die gesamten Layerdatenströme angewandt werden können, so dass eine einzelne Clientanfrage zu den Datenströmen der angefragten Layer führt, die in ihrer Gesamtheit von dem Server übermittelt werden. Dies muss jedoch nicht der Fall sein.

[0015] Die verfügbare Bandbreite kann sich über die Zeitspanne, die es braucht, die angefragten Layer zu übermitteln, signifikant verändern. Angesichts dessen können das vorangegangene System und Verfahren nicht auf die gesamten Layerdatenströme angewandt werden, aber auf manche Teile von den Strömen, die ein Segment des AV-Programms darstellen. Mit anderen Worten, anstatt dass der Client eine Reihe von Layern in ihrer Gesamtheit anfragt, fragt er stattdessen nur einen gleich großen entsprechenden Teil von jedem der involvierten Layer, basierend auf der aktuellen verfügbaren Bandbreite, an. Die Layerteile sind dadurch entsprechend, dass sie zu dem gleichen zeitlichen Segment des AV-Programms gehören. Die restlichen, oben beschrieben, Schritte sind die gleichen, ausgenommen, dass sie stattdessen auf die Daten der Layerteile angewandt werden. Das Ergebnis ist, dass die Daten zugehörig zu einem Teil von jedem Layerstrom empfangen, ge-

rendert und zwischengespeichert werden. Der Client fragt dann den nächsten sequenziellen Teil der AV-Programmlayer an, und der Vorgang des Empfangens, Renderns und Zwischenspeicherns wird wiederholt, bis das gesamte Programm dem Benutzer vorgespielt wurde. Es wird darauf hingewiesen, dass mit der Anfrage jeder Reihe von Layerteilen die Anzahl der involvierten Layer, basierend auf der dann verfügbaren Netzwerkbandbreite, neu eingeschätzt werden. Deshalb kann jede Anfrage zu Teilen mit derselben Anzahl von Layern, oder weniger, oder mehr führen, die angefragt werden. Wenn der Benutzer anweist, dass eine Version des AV-Programms mit höherer Qualität gespielt wird, wiederholt der Client die vorangegangenen Vorgänge, ausgenommen, dass der Client für jedes Segment des Programms jene Layerteile zugehörig zu der maximalen Anzahl an möglichen Erweiterungslayern angesichts den aktuellen Bandbreitenkonditionen auf dem Netzwerk anfragt, beginnend mit dem Layer, der das nächsthöhere Level in Bezug auf die vorher empfangenen Layerteile zugehörig zu dem Segment des zur Diskussion stehenden Programms darstellt.

[0016] Das vorhergehende Schicht-Unicast-Präsentationssystem und -Verfahren kann des Weiteren eine vorwegnehmende Option einschließen. Diese Option erlaubt es einem Benutzer, auszuwählen, zusätzliche Erweiterungslayer, die notwendig sind, um Versionen des Programms mit höherer Qualität bereitzustellen, automatisch durch den Client angefragt, empfangen und zwischengespeichert werden sollen, nachdem die Layer zugehörig zu einer früheren Programmversion mit geringerer Qualität alle angefragt worden sind. Es ist zu beachten, dass dies getan wird, bevor der Benutzer eine Version mit höherer Qualität anfragt. Die vorwegnehmende Option führt dazu, dass die Layerdaten, die notwendig sind, um eine Version des AV-Programms mit höherer Qualität zu rendern, in dem Clientspeicher verfügbar sind, und es dabei dem Client erlauben, das Programm dem Benutzer schneller bereitzustellen.

[0017] Es wird darauf hingewiesen, dass während das vorliegende Schicht-Unicast-Schema oberhalb in Verbindung mit einem AV-Programm beschrieben worden ist, die Erfindung gleichermaßen auf nur Audio- oder nur Videoprogramme anwendbar ist.

[0018] Es wird des Weiteren darauf hingewiesen, dass das Schicht-Unicast-System und -Verfahren gemäß der vorliegenden Erfindung vorteilhaft durch Internetwebsites eingesetzt werden kann, um Versionen von AV-Programmen mit hoher Qualität an Abonnenten zu verkaufen, die die Website besuchen. Eine Version des Programms mit "Basisqualität" könnte einem Sitebesucher umsonst oder für eine Schutzgebühr angeboten werden. Wenn die Basisversion des AV-Programms von ausreichendem Interesse ist, könnte der Benutzer gegen eine Gebühr unterschrei-

ben, um eine Version des Programms mit verbesserter Qualität zu sehen. Dies wird, wie oben beschrieben, mit den Erweiterungslayern erreicht, die zu dem Clientcomputer zugehörig zu dem Besucher gesendet werden. Der Clientcomputer verbindet dann die Erweiterungslayerdaten mit den vorher zwischengespeicherten Layerdaten, um eine Version des Programms mit höherer Qualität zur Betrachtung durch den Besucher zu erzeugen. Der Besucher könnte dann die Anfrage wiederholen, um Versionen des Programms mit noch höherer Qualität anzusehen, solange der Server zugehörig zu der Website Erweiterungslayer höheren Levels verfügbar hat.

[0019] Zusätzlich zu dem gerade beschriebenen Nutzen werden andere Vorteile der vorliegenden Erfindung durch die detaillierte Beschreibung, welche hier anschliessend folgt, ersichtlich, wenn sie in Verbindung mit den Figuren, welche ihr angehängt sind, betrachtet werden.

Kurze Beschreibung der Figuren

[0020] Die spezifischen Merkmale, Aspekte und Vorteile der vorliegenden Erfindung werden in Bezug auf die folgende Beschreibung, angefügten Ansprüche und begleitenden Zeichnungen besser verstanden, wobei:

[0021] [Fig. 1](#) ein Diagramm ist, das ein Allzweckcomputergerät beschreibt, das ein exemplarisches System zum Implementieren der vorliegenden Erfindung bezeichnet.

[0022] [Fig. 2A–B](#) zeigen ein Flussdiagramm, das den gesamten Vorgang zum Erlangen von Versionen eines Audio- und/oder Videoprogramms mit stufenweise höherer Qualität über ein Client-Server basiertes Netzwerk gemäß der vorliegenden Erfindung schematisiert.

[0023] [Fig. 3A–B](#) zeigen ein Flussdiagramm, das einen Teil des gesamten Vorgangs von [Fig. 2](#), der modifiziert worden ist, schematisiert, um eine vorwegholende Option gemäß der vorliegenden Erfindung zu implementieren.

Detaillierte Beschreibung

[0024] In der nachfolgenden Beschreibung von bevorzugten Ausführungsformen der vorliegenden Erfindung wird Bezug auf die beigefügten Zeichnungen gemacht, welche einen Teil hiervon bilden, und in welchen als Illustration spezifische Ausführungsformen gezeigt werden, in denen die Erfindung praktiziert werden kann. Es gilt als verstanden, dass andere Ausführungsformen eingesetzt werden können, und strukturelle Veränderungen gemacht werden können, ohne von dem Umfang der vorliegenden Erfindung abzuweichen.

[0025] Bevor eine Beschreibung der bevorzugten Ausführungsformen der vorliegenden Erfindung bereitgestellt wird, wird eine kurze, allgemeine Beschreibung einer passenden Computerumgebung, in der die Erfindung implementiert werden kann, beschrieben. [Fig. 1](#) stellt ein Beispiel für eine passende Computersystemumgebung **100** dar. Die Computersystemumgebung **100** ist nur ein Beispiel einer geeigneten Computerumgebung und ist nicht gedacht, irgendwelche Einschränkungen bezüglich des Umfangs der Verwendung oder Funktionalität der Erfindung vorzuschlagen. Noch sollte die Computerumgebung **100** interpretiert werden, als hätte sie irgendwelche Abhängigkeiten oder Erfordernisse in Bezug auf irgendeine oder eine Kombination von Komponenten, die in der exemplarischen Arbeitsumgebung **100** dargestellt sind.

[0026] Die Erfindung ist mit zahlreichen anderen Allzweck- oder Spezialcomputersystemumgebungen oder -konfigurationen funktionsbereit. Beispiele von gut bekannten Computersystemen, -umgebungen und/oder -konfigurationen, die zur Verwendung mit der Erfindung geeignet sein können, schließen ein, sind aber nicht darauf begrenzt, Personalcomputer, Servercomputer, Handheld- oder Laptopgeräte, Multiprozessorsysteme, Mikroprozessor-basierte Systeme, Settopboxen, programmierbare Unterhaltungselektronik, Netzwerk-PCs, Minicomputer, Mainframecomputer, verteilte Computerumgebungen, die irgendeines der oben genannten Systeme oder Geräte einschließen, und Ähnliches.

[0027] Die Erfindung kann im allgemeinen Kontext von Computer-ausführbaren Instruktionen beschrieben werden, wie z. B. Programmmodulen, die durch einen Computer ausgeführt werden. Im Allgemeinen schließen Programmmodule Routinen, Programme, Objekte, Komponenten, Datenstrukturen, etc. ein, die bestimmte Aufgaben ausführen oder bestimmte abstrakte Datentypen implementieren. Die Erfindung kann auch in einer verteilten Computerumgebung praktiziert werden, wo Aufgaben durch Remote verarbeitende Geräte ausgeführt werden, die durch ein Kommunikationsnetzwerk verbunden sind. In einer verteilten Computerumgebung können Programmmodule sowohl lokal als auch auf Remote-Computerspeichermedien, einschließlich Datenspeichergeräten, liegen.

[0028] Mit Bezug auf [Fig. 1](#) schließt ein exemplarisches System zum Implementieren der Erfindung ein Allzweckcomputergerät in der Form von einem Computer **110** ein. Komponenten des Computers **110** können einschließen, sind dabei aber nicht darauf begrenzt, eine Prozessoreinheit **120**, einen Systemspeicher **130** und einen Systembus **121**, der verschiedene Systemkomponenten koppelt, einschließlich dem Systemspeicher mit der Prozessoreinheit **120**. Der Systembus **121** kann irgendeiner von ver-

schiedenen Busstrukturtypen sein, einschließlich einem Speicherbus oder Speichercontroller, einem Peripheriebus und einem lokalen Bus, der irgendeine Sorte von Busarchitekturen verwendet. Als Beispiel, und nicht Einschränkung, schließen solche Architekturen einen Industry-Standard-Architecture-Bus (ISA-Bus), Micro-Channel-Architecture-Bus (MCA-Bus), Enhanced-ISA-Bus (EISA-Bus), Video-Electronics-Standards-Association-Local-Bus (VESA-Local-Bus) und Peripheral-Component-Interconnect-Bus (PCI-Bus), auch bekannt als Mezzanine-Bus, ein.

[0029] Der Computer **110** schließt üblicherweise eine Vielfalt von computerlesbaren Medien ein. Computerlesbare Medien können irgendwelche verfügbaren Medien sein, auf die durch den Computer **110** zugegriffen werden kann, und die sowohl flüchtige als auch nichtflüchtige Medien, entfernbare und nicht-entfern timer Medien einschließen. Als Beispiel, und nicht Einschränkung, können computerlesbare Medien Computerspeichermedien und Kommunikationsmedien umfassen. Computerspeichermedien schließen sowohl flüchtige und nicht-flüchtige, entfernbare und nicht-entfern timer Medien ein, die in irgendeinem Verfahren oder einer Technologie zum Speichern von Informationen implementiert sind, wie z. B. computerlesbaren Instruktionen, Datenstrukturen, Programmmodulen oder anderen Daten. Computerspeichermedien schließen ein, sind aber nicht darauf begrenzt, RAM, ROM, EEPROM, Flashmemory oder eine andere Speichertechnologie, CD-ROM, Digital Versatile Disks (DVD) oder andere optische Diskspeicher, magnetische Kassetten, magnetische Bänder, magnetische Diskspeicher oder andere magnetische Speichergeräte, oder irgendein anderes Medium, welches verwendet werden kann, um die gewünschten Informationen zu speichern, und auf welches durch den Computer **110** zugegriffen werden kann. Kommunikationsmedien enthalten üblicherweise computerlesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten in einem modulierten Datensignal, wie z. B. einer Trägerwelle oder anderem Transportmechanismus, und schließen irgendwelche Informationsliefermedien ein. Der Begriff "moduliertes Datensignal" meint ein Signal, das eine oder mehrere seiner Charakteristiken in solch einer Weise gesetzt oder verändert hat, um Informationen in dem Signal zu codieren. Als Beispiel, und nicht Einschränkung, schließen Kommunikationsmedien verkabelte Medien, wie z. B. ein verkabeltes Netzwerk oder Direktkabelverbindungen, und kabellose Medien, wie z. B. akustische, RF-, infrarote und andere kabellose Medien ein. Kombinationen irgendwelcher der oben genannten sollten ebenso im Umfang der computerlesbaren Medien eingeschlossen sein.

[0030] Der Systemspeicher **130** schließt Computerspeichermedien in Form von flüchtigen und/oder

nicht-flüchtigen Speicher ein, wie z. B. Read Only Memory (ROM) **131** und Random Access Memory (RAM) **132**. Ein Basic Input/Output-System **133** (BIOS), das die Basisroutinen enthält, die helfen, Informationen zwischen Elementen innerhalb des Computers **110** zu übertragen, wie z. B. während des Hochfahrens, sind üblicherweise in dem ROM **131** gespeichert. RAM **132** enthält üblicherweise Daten und/oder Programmmodule, die sofort zugreifbar sind und/oder auf denen in Kürze durch die Prozessoreinheit **120** gearbeitet wird. Als Beispiel, und nicht Einschränkung, stellt [Fig. 1](#) das Betriebssystem **134**, Anwendungsprogramme **135**, andere Programmmodule **136** und Programmdateien **137** dar.

[0031] Der Computer **110** kann auch andere entfernbare/nicht-entfern timer, flüchtige/nicht-flüchtige Computerspeichermedien einschließen. Nur als Beispiel stellt [Fig. 1](#) ein Festplattenlaufwerk **141** dar, das von oder zu nicht-entfern timer, nicht-flüchtigen magnetischen Medien liest oder schreibt, ein magnetisches Disklaufwerk **151**, das von oder zu einer entfernbaren, nicht-flüchtigen magnetischen Disk **152** liest oder schreibt, und einen optischen Disklaufwerk **155**, das von oder zu einer entfernbaren, nicht-flüchtigen optischen Disk **156**, wie z. B. einer CD-ROM oder anderen optischen Medien, liest oder schreibt. Andere entfernbare/nicht-entfern timer, flüchtige/nicht-flüchtige Computerspeichermedien, die in der exemplarischen Arbeitsumgebung verwendet werden können, enthalten, sind aber nicht darauf begrenzt, magnetische Bandkassetten, Flashmemorykarten, Digital Versatile Disks, digitale Videobänder, Solid State RAM, Solid State ROM, und Ähnliches. Das Festplattenlaufwerk **141** ist üblicherweise mit dem Systembus **121** durch eine Schnittstelle für nicht-entfern timer Speicher, wie z. B. Schnittstelle **140**, verbunden, und das magnetische Disklaufwerk **151** und optische Disklaufwerk **155** sind üblicherweise mit dem Systembus **121** durch eine Schnittstelle für entfernbare Speicher, wie z. B. Schnittstelle **155**, verbunden.

[0032] Die Laufwerke und ihre zugehörigen Computerspeichermedien, wie oberhalb diskutiert und in [Fig. 1](#) dargestellt, stellen Speicher für computerlesbare Instruktionen, Datenstrukturen, Programmmodule und andere Daten für den Computer **110** bereit. In [Fig. 1](#), z. B., ist das Festplattenlaufwerk **141** so dargestellt, dass es das Betriebssystem **144**, Anwendungsprogramme **145**, andere Programmmodule **146** und Programmdateien **147** speichert. Es ist zu beachten, dass diese Komponenten entweder dieselben oder verschieden von dem Betriebssystem **134**, den Anwendungsprogrammen **135**, den anderen Programmmodulen **136** und den Programmdateien **137** sein können. Dem Betriebssystem **144**, den Anwendungsprogrammen **145**, anderen Programmmodulen **146** und Programmdateien **147** sind hier verschiedene Nummern gegeben worden, um darzustellen

len, dass sie wenigstens verschiedene Kopien sind. Ein Benutzer kann Befehle und Informationen in den Computer **110** durch Eingabegeräte, wie z. B. ein Keyboard **162** und ein Zeigergerät **161**, allgemein bezeichnet als eine Maus, einen Trackball oder ein Touchpad, eingeben. Andere Eingabegeräte (nicht gezeigt) können ein Mikrofon, einen Joystick, ein Gamepad, eine Satellitenschüssel, einen Scanner oder Ähnliches einschließen. Diese und andere Eingabegeräte sind oft mit der Prozessoreinheit **120** durch eine Benutzereingabeschnittstelle **160** verbunden, die mit dem Systembus **121** gekoppelt ist, kann aber durch andere Schnittstellen- und Busstrukturen, wie z. B. einem Parallelbus, Gameport oder einem Universal Serial Bus (USB), verbunden sein. Ein Monitor **191** oder anderer Anzeigegerätetyp ist auch mit dem Systembus **121** über eine Schnittstelle, wie z. B. einer Videoschnittstelle **190**, verbunden. Zusätzlich zu dem Monitor können Computer auch andere periphere Ausgabegeräte, wie z. B. Lautsprecher **197** und Drucker **196**, einschließen, welche über eine Ausgangsperipherieschnittstelle **195** verbunden sein können. Von besonderer Bedeutung für die vorliegende Erfindung ist, dass eine Kamera **163** (wie z. B. eine digitale/elektronische Bild- oder Videokamera, oder ein Filmfotografischer Scanner), die zum Aufnehmen einer Sequenz von Bildern **164** geeignet ist, auch als Eingabegerät für den Personalcomputer **110** eingeschlossen sein kann. Des Weiteren könnten, während nur eine Kamera gezeigt ist, viele Kameras als Eingabegeräte für den Personalcomputer **110** eingeschlossen sein. Die Bilder **164** von der einen oder mehreren Kameras werden in den Computer **110** über eine passende Kameraschnittstelle **164** eingegeben. Diese Schnittstelle **165** ist mit dem Systembus **121** verbunden, und erlaubt dadurch, dass die Bilder zu dem RAM **132**, oder einem der anderen Datenspeichergeräte zugehörig zu dem Computer **110**, geroutet und darin gespeichert werden. Es wird jedoch darauf hingewiesen, dass Bilddaten auch von irgendeinem der vorher erwähnten computerlesbaren Medien in den Computer **110** eingegeben werden können, ohne die Verwendung der Kamera **163** zu erfordern.

[0033] Der Computer **110** kann in einer Netzwerkumgebung unter Verwendung logischer Verbindungen zu einem oder mehreren Remotecomputern, wie z. B. Remotecomputer **180**, arbeiten. Der Remote-Computer **180** kann ein Personalcomputer, ein Server, ein Router, ein Netzwerk-PC, ein Peergerät, oder anderer allgemeiner Netzwerkknoten sein und schließt üblicherweise viele oder alle der Elemente, die oben mit Bezug auf Computer **110** beschrieben wurden, ein, obwohl nur ein Speichergerät **181** in [Fig. 1](#) dargestellt worden ist. Die logischen Verbindungen, die in [Fig. 1](#) dargestellt sind, schließen ein Local Area Network (LAN) **171** und ein Wide Area Network (WAN) **173** ein, können aber auch andere Netzwerke einschließen. Solche Netzwerkumgebun-

gen sind alltäglich in Büros, unternehmensweiten Computernetzwerken, Intranets und dem Internet.

[0034] Wenn er in einer LAN-Netzwerkumgebung verwendet wird, ist der Computer **110** mit dem LAN **171** durch eine Netzwerkschnittstelle oder Adapter **170** verbunden. Wenn er in einer WAN-Netzwerkumgebung verwendet wird, schließt der Computer **110** üblicherweise ein Modem **172** oder andere Mittel zum Herstellen von Kommunikationen über das WAN **173**, wie z. B. dem Internet, ein. Das Modem **172**, welches intern oder extern sein kann, kann mit dem Systembus **121** über die Benutzereingabeschnittstelle **160** oder andere geeignete Mechanismen verbunden sein. In einer Netzwerkumgebung können Programmmodule, die mit Bezug auf Computer **110** gezeigt sind, oder Teile davon, in dem Remote-Datenspeichergerät gespeichert sein. Als Beispiel, und nicht Einschränkung, stellt [Fig. 1](#) Remote-Anwendungsprogramme **185** so dar, dass sie in dem Speichergerät **181** liegen. Es wird begrüßt, dass die gezeigten Netzwerkverbindungen exemplarisch sind und andere Mittel zum Herstellen eines Kommunikationslinks zwischen den Computern verwendet werden kann.

[0035] Die exemplarische Arbeitsumgebung ist nun diskutiert worden, der restliche Teil dieses Beschreibungsabschnitts ist der Beschreibung von den Programmmodulen, die die Erfindung verkörpern, gewidmet. Im Allgemeinen beziehen das System und der Prozess gemäß der vorliegenden Erfindung einen Client ein, der Versionen eines Audio- und/oder Video-Programms mit stufenweise höherer Qualität über ein Client-Server-basiertes Netzwerk erlangt. Dies wird unter Verwendung dessen erreicht, was als ein Schicht-Unicast-Schema bezeichnet wird. Im Speziellen, mit Bezug auf das Flussdiagramm von [Fig. 2A–B](#), weist ein Benutzer zuerst den Client an, ein bestimmtes AV-Programm zu erlangen (Prozessvorgang **200**). Dies wird mittels irgendeiner konventionellen Client-Benutzerschnittstelle durchgeführt. In dem Fall, wo das Netzwerk das Internet ist, wird der Benutzer üblicherweise ein bestimmtes Programm von einer Website auswählen, die solche Programme bereitstellt. Der Client fragt dann an, dass der Server, auf dem die geschichteten AV-Daten zugehörig zu dem gewünschten Programm liegen, die Daten zu dem Client über das Netzwerk übermittelt (Prozessvorgang **202**). Dabei spezifiziert der Client auch, wie viele aufeinanderfolgende Layer, beginnend mit dem Basislayer, übermittelt werden sollen. Diese Entscheidung basiert auf der Bandbreite, die aktuell in der Netzwerkverbindung zwischen dem Client und dem Server verfügbar ist. Um die passende Anzahl an aufeinanderfolgenden Layern zu spezifizieren, muss der Client natürlich die Besonderheiten des Schicht-Schemas, das eingesetzt wurde, die AV-Daten zu segmentieren, wissen, und wie die gewünschten Layer identifiziert werden, wenn sie von dem Ser-

ver angefragt werden. Der Client kann diese Information, in Fall eines Netzwerks, wo ein standardisiertes Schicht- und Identifikationsschema eingesetzt wird, kennen. Für Netzwerke, wie z. B. dem heutigen Internet, wo viele verschiedenen Schicht- und Identifikationsschemata eingesetzt werden können, kann man sich jedoch ausmalen, dass der Prozessvorgang, bei dem der Client die AV-Daten zugehörig zu dem ausgewählten Programm anfragt, einschließt, dass von dem Server zuerst Besonderheiten bezüglich des Schicht- und Identifikationsschemas, das in Verbindung mit dem Programm eingesetzt ist, erlangt werden.

[0036] Beim Empfangen der Anfrage des Clients übermittelt der Server die angefragten Daten zu dem Client über das Netzwerk (Prozessvorgang **204**). Die Übertragungsrate sollte mindestens gleich der Abspielrate sein, so dass es keine Verzögerung beim Rendern der Daten für eine Echtzeitpräsentation für den Benutzer durch den Client gibt. Der Client empfängt die Daten von dem Server, gibt diese vorzugsweise durch einen "Abspielpuffer" ein und rendert sie für den Betrachter über konventionelle Verfahren (Prozessvorgang **206**). Der Abspielpuffer zögert den Start des Abspielens durch irgendeine kleine Zeiteinheit, um Extradaten anzusammeln. Dieses Puffern erlaubt dem Client, sicherzustellen, dass die Abspielrate konstant ist. Ohne solch eine Pufferung kann das gerenderte Programm aufgrund von kleinen Veränderungen in der Rate, mit der Daten über das Netzwerk empfangen werden, "flackernd" sein. Der Client speichert die empfangenen Daten auch für eine zukünftige Verwendung zwischen, wie es kurz erklärt wird (Prozessvorgang **208**).

[0037] An diesem Punkt in dem Prozess hat der Benutzer drei Auswahlmöglichkeiten. Wenn das AV-Programm für den Benutzer nicht von Interesse ist, könnte der Benutzer nichts machen, und zusätzliche erweiternde Layerdaten würden nicht übermittelt werden. Dazu wird im Prozessvorgang **210** ermittelt, ob der Benutzer irgendwelche weiteren Befehle hinsichtlich des vorher ausgewählten AV-Programms eingibt. Wenn eine Eingabe von dem Benutzer empfangen wird, wird festgestellt, ob der Benutzer angewiesen hat, dass der Empfang des eingehenden Datenstroms zugehörig zu dem ausgewählten AV-Programm abgebrochen werden soll (Prozessvorgang **212**). Mit anderen Worten, hat der Benutzer das AV-Programm abgebrochen. Wenn dem so ist, beendet der Client den eingehenden Datenstrom (Prozessvorgang **214**). Es wird darauf hingewiesen, dass der Benutzervorgang des Abbrechens des Programms den Prozess nicht beendet, weil dies nicht notwendigerweise bedeutet, dass der Benutzer nicht an einem Betrachten von erweiterten Versionen des Programms interessiert ist. Zum Beispiel kann der Benutzer gerade genug von der Basisversion des Programms mit geringer Qualität anschauen, um zu

entscheiden, dass er oder sie eher sofort eine erweiterte Version sehen möchte, als mit dem Betrachten der Version mit geringer Qualität fortzufahren. Deshalb wird es als nächstes ermittelt, ob der Benutzer die erweiterte Version des ausgewählten AV-Programms anfragt (Prozessvorgang **216**).

[0038] Das dritte der vorher erwähnten Szenarien bezieht ein, dass der Benutzer das gesamte AV-Programm nochmals betrachtet und interessiert genug wird, um eine Version des Programms mit höherer Qualität empfangen zu wollen. In diesem Fall wurde es im Prozessvorgang **212** ermittelt, dass der Benutzer nicht angewiesen hat, den Empfang des eingehenden Datenstroms abubrechen, sondern eher, dass er eine erweiterte Version des Programms angefragt hat (Prozessvorgang **218**).

[0039] In diesem Fall, wo der Benutzer nicht eine erweiterte Version des AV-Programms anfragt, werden keine Maßnahmen ergriffen, und der Client würde die zwischengespeicherten Daten für die Zeitspanne aufbewahren, die durch die Zwischenspeicherregeln der von dem Client eingesetzten Netzwerksoftware vorgeschrieben ist. Wenn jedoch der Benutzer anweist, dass eine erweiterte Version des AV-Programms bereitgestellt werden soll (egal, ob vorher das volle Programm betrachtet wurde oder ob es mit-tendrin abgeschaltet wurde), würde der Client anfragen, dass erweiternde AV-Daten zugehörig zu dem gewünschten Programm zu ihm über das Netzwerk bereitgestellt werden. Dies setzt jedoch voraus, dass der Server zusätzliche Erweiterungslayer zur Verfügung hat. Deshalb wird vor dem Anfragen der Erweiterungsdaten zuerst im Prozessvorgang **220** ermittelt, ob irgendwelche vorher nicht angefragten Erweiterungslayer zugehörig zu dem ausgewählten AV-Programm von dem Server verfügbar sind. Vorzugsweise wird der Client dies basierend auf den vorher empfangenen Informationen aufgrund des Schicht- und Identifikationsschemas, das in Verbindung mit dem ausgewählten AV-Programm eingesetzt wird, bereits wissen. Wenn zusätzliche Layer nicht verfügbar sind, wird der Benutzer durch den Client darüber informiert (Prozessvorgang **222**) und der Prozess endet. Wenn zusätzliche Layer verfügbar sind, dann fragt der Client an, dass der Server die Erweiterungs-AV-Daten zugehörig zu dem gewünschten Programm bereitstellt (Prozessvorgang **224**). Im Speziellen fragt der Client so viele der aufeinanderfolgenden Erweiterungslayer an, wie es hinsichtlich der verfügbaren Bandbreite möglich ist, beginnend mit dem Layer mit dem geringsten Level, der in dem letzten Schritt nicht angefragt wurde. Als Antwort übermittelt der Server die angefragten Daten in derselben Weise wie vorher beschrieben (Prozessvorgang **226**).

[0040] Während die angefragten Daten empfangen werden, speichert der Client die neuen Daten (Pro-

zessvorgang **228**) und kombiniert sie mit den vorher gespeicherten Layerdaten mit geringerem Level unter Verwendung der Standardverfahren, die für den Typ der involvierten Layerdaten geeignet ist (Prozessvorgang **230**). Die kombinierten Daten werden dann verwendet, um das Programm zu rendern und es dem Benutzer zu präsentieren (Prozessvorgang **232**). Das gerenderte Programm wird eine Version des vorher angesehenen Programms mit höherer Qualität darstellen.

[0041] An diesem Punkt, wenn der Benutzer entscheidet, dass das AV-Programm nicht länger von Interesse ist, kann er oder sie wieder nichts machen und zusätzliche Erweiterungslayerdaten würden nicht übermittelt werden, oder der Benutzer könnte eine noch weiter erweiterte Version des ausgewählten Programms anfragen oder der Benutzer könnte den eingehenden Datenstrom unterbrechen, bevor das Programm vollständig angesehen wurde. Deshalb hat der Benutzer die gleichen drei Optionen wie zuvor. Dazu wird in Prozessvorgang **234** ermittelt, ob der Benutzer irgendwelche weiteren Befehle hinsichtlich des vorher ausgewählten AV-Programms eingibt. Wenn der Benutzer Befehle eingibt, dann wird der vorhergehende Prozess (der Vorgänge **212** bis **234**) wiederholt. Auf diese Weise kann der Prozess des Erlangens von Versionen des ausgewählten AV-Programms mit immer höherer Qualität fortgesetzt werden, bis entweder der Benutzer aufhört, erweiterte Versionen anzufragen oder es keine von dem Server verfügbaren übrigen Erweiterungslayer gibt.

[0042] Ein einfaches Beispiel des vorangegangenen Prozesses, wo jeweils nur ein Layer aufgrund von Bandbreitenbeschränkungen gestreamt werden kann, ist das Folgende. Während des ersten Abspielens wird der Basislayer gestreamt, eingespielt und gespeichert. Während eines vom Benutzer angewiesenen zweiten Abspielens wird der Layer **2** gestreamt, gespeichert und in Verbindung mit dem Basislayer abgespielt. Während eines dritten vom Benutzer angewiesenen Abspielens wird der Layer **3** gestreamt und gespeichert und in Verbindung mit dem Basislayer und Layer **2** abgespielt. Dies kann solange fortfahren, solange es vom Server verfügbare übrige Erweiterungslayer gibt, und der Benutzer in dem Betrachten von Versionen des AV-Programms mit höherer Qualität interessiert ist. Der Prozess ist ähnlich, wenn mehr als ein Layer auf einmal gestreamt werden kann. Zum Beispiel, in dem Fall, wo zwei Layer gleichzeitig innerhalb der Bandbreitenbeschränkungen gestreamt werden können, würde der Prozess wie folgt sein. Während des erstens Abspielens werden Layer **1** (d. h. der Basislayer) und Layer **2** gestreamt, abgespielt und gespeichert. Während eines vom Benutzer angewiesenen zweiten Abspielens werden Layer **3** und **4** gestreamt, gespeichert und in Verbindung mit den Layern **1** und **2** abgespielt. Und wäh-

rend eines dritten vom Benutzer angewiesenen Abspielens werden Layer **5** und **6** gestreamt und gespeichert und in Verbindung mit den Layern **1**, **2**, **3** und **4** abgespielt. Dieser Prozess kann wieder solange fortfahren, solange es vom Server verfügbare übrige Erweiterungslayer gibt, und der Benutzer im Betrachten erweiterter Versionen interessiert ist.

[0043] Es wird darauf hingewiesen, dass während das vorliegende Schicht-Unicast-System und -Prozess oberhalb in Verbindung mit einem AV-Programm beschrieben wurde, die Erfindung gleichermaßen auf nur Audio- oder nur Videoprogramme anwendbar ist.

[0044] Das vorangegangene Schicht-Unicast-System und -Prozess kann auf die gesamte Layerdatenströme angewandt werden, so dass eine einzelne Clientanfrage dazu führt, dass die Datenströme der angefragten Layer in ihrer Gesamtheit von dem Server übermittelt werden. Dies muss jedoch nicht der Fall sein. Die verfügbare Bandbreite kann sich über die Zeitspanne, die es benötigt, die angefragten Layer zu übertragen, signifikant verändern. Angesichts dessen kann das vorliegende Schicht-Unicast-System und -Vorgang nicht auf die gesamten Layerdatenströme angewandt werden, aber auf Teile der Ströme, die ein zeitliches Segment des AV-Programms darstellen. Mit anderen Worten, anstatt dass der Client eine Reihe von Layern in ihrer Gesamtheit anfragt, kann er stattdessen nur einen gleich großen, entsprechenden Teil jedes der involvierten Layer, basierend auf der aktuell verfügbaren Bandbreite anfragen. Die Layerteile entsprechen sich, in dem sie zu dem gleichen zeitlichen Segment des AV-Programms gehören. Der Prozess, der oberhalb in Bezug auf **Fig. 2** beschrieben wurde, würde für diese stückweise Ausführungsform auch gelten, wobei der Prozessvorgang **202** einbeziehen würde, dass der Client Schicht-AV-Daten, zugehörig zu dem ausgewählten Programm, in einer stückweisen Art und Weise anfragt. Im Speziellen wird eine erste Reihe von entsprechenden Teilen einer Anzahl aufeinander folgender Layer des Programms angefragt, von denen durch den Client ermittelt wurde, dass sie angesichts der aktuell verfügbaren Bandbreite des Netzwerkes empfangen werden können. Der Client fragt dann den nächsten sequenziellen Teil der AV-Programmlayer an, usw., bis ermittelt wird, dass das gesamte Programm für den Benutzer abgespielt worden ist, oder der Benutzer das Programm abbricht. Es wird darauf hingewiesen, dass mit der Anfrage für jede Reihe von Layerteilen die Anzahl der involvierten Layer, basierend auf der dann verfügbaren Netzwerkbandbreite erneut festgestellt wird. Deshalb kann jede Anfrage zu Teilen mit derselben Anzahl an Layern oder weniger oder mehr führen, die angefragt werden. Es gibt einige Vorteile der stückweisen Ausführungsform des vorliegenden Systems und Verfahrens. Zum Beispiel kann die verfügbare Bandbreite

auf dem Netzwerk sich während des Übertragens des AV-Programms genügend ändern, dass entweder zusätzliche Daten zugehörig zu einem Erweiterungslayer eines höheren Levels hinzugefügt werden könnten, wobei die Qualität der Präsentation für den Benutzer verbessert wird, oder umgekehrt die Bandbreite soweit abnehmen könnte, dass die Daten zugehörig zu einem Layer fallengelassen werden müssen, um einen übermäßigen Verlust der übermittelten Daten zu verhindern. Das Nettoergebnis ist, dass das gesendete AV-Programm, das durch den Client dem Benutzer bereitgestellt wird, über jede Zeitspanne zugehörig zu dem Teil der gesendeten Daten das bestverfügbare ist. Natürlich ist diese stückweise Ausführungsform bezüglich des Clientprozessors intensiver, indem der Client die Anzahl der Layer im Auge behalten muss, die in jeder der Anfragen für einen Teil der Layerdaten involviert ist, so dass, wenn eine Version des Programms mit höherer Qualität durch den Benutzer angefragt wird, der Client weiß, mit welchem der Layer des höheren Levels er beginnen muss, wenn Erweiterungsdaten für einen bestimmten Teil des AV-Programms angefragt werden. Zusätzlich muss der Client wiederholt Layerdaten abfragen, statt es nur einmal pro Programm machen zu müssen.

[0045] In der stückweisen Ausführungsform, wenn der Benutzer anweist, dass eine Version des AV-Programms mit höherer Qualität gespielt wird, involviert der Prozessvorgang **220** aus **Fig. 2** für jedes Segment des Programms den Client dadurch, dass er jene Layerteile zugehörig zu der maximalen Anzahl an möglichen Erweiterungslayern hinsichtlich der aktuellen Bandbreitegegebenheiten auf dem Netzwerk abfragt, beginnend mit dem Layer, der das nächst höhere Level in Bezug auf die vorher empfangenen Layerteile zugehörig zu demselben zeitlichen Segment des Programms darstellt.

[0046] Es wird darauf hingewiesen, dass die Größe der Teile der Daten in jedem Layer, die in den wiederholten Anfragen in Bezug auf die stückweise Ausführungsform einbezogen sind, irgendeine gewünschte sein kann. Es wird jedoch bevorzugt, dass die Größe so ausgewählt ist, dass die Dauer des Teils ungefähr die Menge an Zeit ist, von der erwartet wird, dass die Netzwerkbandbreite konstant bleiben wird. Diese Größenbestimmung kann, wenn notwendig, zum Äußersten getrieben werden, so dass jeder Teil nur ein einzelnes Paket des Datenstroms von jedem der Layer ist, die in einer Anfrage involviert sind.

[0047] Das vorangegangene Schicht-Unicast-Präsentationssystem und -Prozess kann des Weiteren eine vorwegnehmende (pre-fetching) Option einschließen. Diese Option erlaubt es einem Benutzer, auszuwählen, ob zusätzliche Erweiterungslayer, die gebraucht werden, um Versionen des Programms mit höherer Qualität bereitzustellen, automatisch durch

den Client angefragt, empfangen und zwischengespeichert werden sollen, nachdem die Layer, zugehörig zu einer vorhergehenden Version des Programms mit geringer Qualität vollständig empfangen worden sind. Es ist zu beachten, dass dies getan wird, bevor der Benutzer eine Version mit höherer Qualität anfragt. Wenn der Benutzer anfragt, dass der Client automatisch die AV-Daten verlangt, die notwendig sind, um eine Version des ausgewählten Programms mit höherer Qualität zu erzeugen, ist der Prozess im Wesentlichen der gleiche wie in **Fig. 2** skizziert bis zu dem Punkt, wo der oder die erste(n) Layer des ausgewählten Programms empfangen, gerendert und zwischengespeichert worden sind (d. h., bis zu dem Prozessvorgang **208**). Danach ändert sich der Prozess, wie in **Fig. 3** skizziert. Im Speziellen, während der Zeit, in der der Client die angefragten AV-Daten empfängt, rendert und zwischenspeichert, wird es ermittelt, ob der Benutzer angewiesen hat, dass der Empfang des eingehenden Datenstroms abgebrochen wird (Prozessvorgang **300**). Wenn nicht, fragt der Client die AV-Daten an, die benötigt werden, um eine erweiterte Version des ausgewählten Programms zu erzeugen, sobald alle Daten empfangen worden sind (Prozessvorgang **302**), um die Version mit geringer Qualität, die von dem Benutzer betrachtet wird, zu erzeugen. Es wird keine Anweisung, dies zu tun, von dem Benutzer benötigt. Die erweiternden Daten würden, wie vorher beschrieben, angefragt, wobei so viele Layer wie möglich, angesichts der verfügbaren Netzwerkbandbreite angefragt werden, beginnend mit dem Layer, der ein Level höher ist als der Layer mit dem höchsten Level der vorher angefragten Daten. Der Client könnte stoppen, sobald alle erweiternden Layer, zugehörig zu dem gesamten Programm für den letzten Schritt abgefragt wurden. Vorzugsweise fragt der Client jedoch eine Reihe von Erweiterungslayern mit noch höheren Leveln an, usw., bis alle verfügbaren Layerdaten angefragt worden sind. Wenn der Benutzer jedoch das Programm abbricht, beendet der Client den eingehenden Datenstrom (Prozessvorgang **304**). Wie vorher angegeben, ist es an diesem Punkt nicht bekannt, ob die Handlung des Benutzers bezüglich des Abbrechens des Programms bedeutet, dass er oder sie nicht in der Betrachtung erweiterter Versionen des Programms interessiert ist, oder ob er oder sie eine erweiterte Version lieber gleich sehen möchte als mit dem Betrachten der Version mit geringer Qualität fortzufahren. Deshalb beginnt der Client in dem Fall, wo der Benutzer das eingehende Programm abbricht, die benötigten AV-Daten, wie oben beschrieben, abzufragen, um eine erweiterte Version des ausgewählten Programms zu erzeugen (Prozessvorgang **306**). Weil die vorhergehende Version des Programms abgebrochen wurde, können natürlich nur die erweiternden Daten, zugehörig zu dem vorher empfangenen Teil des Programms in dem ersten Schritt der Anfrageprozedur erlangt werden. Die rudimentäreren Layer für die restlichen Teile des Programms werden

angefragt. Wie zuvor würde der Server die angefragten Daten übermitteln (Prozessvorgang **308**) und der Client würde die eingehenden Daten empfangen und zwischenspeichern (Prozessvorgang **310**). Es wird danach ermittelt, ob der Benutzer anfragt, dass eine Version des ausgewählten Programms mit höherer Qualität bereitgestellt wird (Prozessvorgang **312**). Wenn solch eine Anfrage gemacht wird, kombiniert der Client alle Layer der AV-Daten, die er zwischengespeichert hat, um die erweiterte Version des ausgewählten Programms zu erzeugen (Prozessvorgang **314**). In dem Fall, wo die Layer in einer stückweisen Art und Weise angefragt werden, bezieht dieser letzte Vorgang das Kombinieren aller entsprechenden Layerteile von jedem aufeinander folgenden Segmente des AV-Programms ein. Die kombinierten Daten werden verwendet, um das Programm zu rendern, und es wird dem Benutzer präsentiert (Prozessvorgang **316**). Es sollte beachtet werden, dass erweiterte Programme in dem Fall der vorwegnehmenden Option von einer höheren Qualität sein können, als sie in dem Fall des Prozesses wie in **Fig. 2** skizziert erzeugt werden, weil mehr als eine Iteration von Erweiterungs-layeranfragen gemacht werden könnten, und die Daten empfangen und zwischengespeichert werden könnten, bevor der Benutzer die erweiterte Version des Programms anfragt.

[0048] Wenn der Benutzer bestimmt, eine erweiterte Version des ausgewählten AV-Programms zu betrachten, kann der Client die Funktion des Anfragens aller Layer, die für das Programm verfügbar sind, abgeschlossen haben. Deshalb wird es als nächstes im Prozessvorgang **318** ermittelt, ob irgendwelche restlichen Layer vom Server verfügbar sind. Wenn dem so ist, werden die Prozessvorgänge **300** bis **318** wiederholt. Wenn jedoch alle verfügbaren Daten angefragt worden sind, endet der Prozess.

[0049] Die vorwegnehmende Option hat manche reizvollen Vorteile. Als Erstes führt das Vorwegnehmen dazu, dass Layerdaten, die benötigt werden, um eine Version des AV-Programms mit höherer Qualität zu rendern, in dem Clientspeicher verfügbar sind, und erlaubt es dabei dem Client, das Programm dem Benutzer schneller bereitzustellen. Zusätzlich, wie oben angedeutet, kann der Client im Wesentlichen kontinuierlich Layerdaten zugehörig zu dem ausgewählten Programm anfragen und zwischenspeichern. Deshalb ist es möglich, mehr als einen Iterationsschritt mit Erweiterungs-layerdaten zu haben, die fertig zum Rendern sind, wenn der Benutzer die erweiterte Version des ausgewählten Programms anfragt. Als Folge kann eine Version des Programms mit höherer Qualität bereitgestellt werden, als es durch Anfragen und Rendern des Programms mit jeweils einer Reihe von Erweiterungs-layern sein kann.

[0050] Eine Variation der vorwegnehmenden Option, die auch vorteilhaft ist, involviert eher das Fortfah-

ren des Anfragens (falls nötig), Empfangens und Zwischenspeicherns von AV-Layerdaten, selbst wenn der Benutzer angewiesen hat, dass die Präsentation des zugehörigen Programms beendet wird, als das Beenden des eingehenden AV-Datenstroms. Auf diese Weise wird, wenn der Benutzer anschließend anweist, dass eine Version des AV-Programms mit höherer Qualität bereitgestellt werden soll, das vorher beschriebene Problem, dass man nur einen Teil der Layerdaten, zugehörig zu der beendeten Version des Programms hat, vermieden. Dies ist vorteilhaft, weil die Version des Programms mit höherer Qualität, die dem Benutzer präsentiert wird, eine im Allgemeinen gleichmäßigere Qualität durchwegs haben würde. Ohne die vorhergehende Prozedur würde der letzte Teil des Programms mit höherer Qualität eine merkliche Abnahme des Qualitätslevels haben, weil der Client eher Layer mit geringem Level zugehörig zu der beendeten Version des Programms anfragen müsste, als zusätzliche Erweiterungs-layer.

[0051] Viele Websites des Internets bieten AV-Programme an, die zu einem Clientcomputer gestreamt werden können, wie z. B. Nachrichten- und Sport-Websites, wie vorher erwähnt. Das Schicht-Unicast-System und -Prozess gemäß der vorliegenden Erfindung kann vorteilhaft durch solche Sites eingesetzt werden, um Versionen mit höherer Qualität des AV-Programms an Abonnenten, die die Website besuchen, zu verkaufen. Im Besonderen könnte eine Version des Programms mit "Basisqualität" einem Site-Besucher umsonst oder für eine Schutzgebühr angeboten werden. Diese Version des Programms mit Basisqualität kann die Version sein, die ursprünglich durch einen Clientcomputer zugehörig zu dem Site-Besucher angefragt und gerendert wurde, welche den Basislayer zugehörig zu dem Programm und so viele Erweiterungs-layer, wie es die verfügbare Bandbreite erlaubt, einschließt. Jedoch kann die Version des AV-Programms mit Basisqualität begrenzt sein auf den Basislayer und vielleicht eine kleine Anzahl von Erweiterungs-layern, wie durch den Server zugehörig zu der Website angewiesen, selbst wenn der Client ursprünglich mehr Layer anfragt. Der Zweck des Einschränkens an Layern, die ursprünglich einem Sitebesucher bereitgestellt werden, ist es, den Besucher eine Version des AV-Programms ansehen zu lassen, die gerade gut genug ist, die Neugierde zu wecken und den Benutzer zu verlocken, die verbesserte Version zu kaufen. Wenn die Basisversion des AV-Programms von genügendem Interesse ist, könnte der Besucher gegen eine Gebühr unterschreiben, eine Version des Programms mit verbesserter Qualität zu betrachten. Dies wird, wie oben beschrieben, dadurch erreicht, dass die Erweiterungs-layer zu dem Clientcomputer zugehörig zu dem Besucher gesendet werden. Der Clientcomputer kombiniert anschließend die Erweiterungs-layerdaten mit den vorher zwischengespeicherten Layerdaten, um eine Version des Programms mit

höherer Qualität zu erzeugen, um von dem Besucher betrachtet zu werden. Der Besucher könnte anschließend die Anfrage wiederholen, um eine Version des Programms mit noch höherer Qualität zu betrachten, solange der Server zugehörig zu der Website Erweiterungslayer mit höherem Level verfügbar hat.

[0052] Als Ein Beispiel des vorangegangenen Prozesses nehme man einen Benutzer an, der eine Sportwebsite im Internet besucht, die Hockey-Video-Highlights bereitstellt. Der Besucher kann auf eines der Highlights "klicken", die auf der Site aufgelistet sind, und kann es mit dem "Basis"-Qualitätslevel umsonst anschauen. Bei diesem Qualitätslevel ist es dem Benutzer möglich, allgemein zu sehen, was passiert, aber er kann nicht genügend Details sehen, um überhaupt den Puck auszumachen. Wenn das Highlight für den Besucher nicht von Interesse ist, werden keine weiteren Anfragen gemacht, oder der Besucher kann sogar das Playback anhalten, bevor es beendet ist. Wenn das Highlight jedoch von Interesse ist, wird der Benutzer eingeladen, für eine Version mit höherer Qualität gegen eine Gebühr zu unterschreiben. Die Bezahlungseinrichtung kann durch irgendein konventionelles Schema behandelt werden, wie z. B. jene, die derzeit auf kommerziellen Internet-Websites verwendet werden. Diesmal sieht der Besucher aufgebesserte Details, einschließlich des Pucks. Es ein drittes Mal abzuspielen, könnte mehr Videoframes einführen, was das Abspielen glatter macht.

[0053] Als ein anderes Beispiel enthalten viele Websites Audiobeispiele oder Lieder. Diese Sites können den Zugriff auf eine Version des Audioprogramms mit geringer Qualität umsonst anbieten. Wenn der Benutzer anschließend in einer Version des Programms mit höherer Qualität interessiert ist, würde die Übertragung von zusätzlichen qualitätsverbessernden Audiodatenlayern das Bezahlen einer Gebühr erfordern.

[0054] In Bezug auf das Bezahlen für Versionen eines Programms mit höherer Qualität kann eine einzelne Gebühr vom Besucher für so viele "Verbesserungs"-Anfragen verlangt werden, wie der Besucher machen möchte, oder alternativ kann von dem Besucher jede Anfrage in Rechnung gestellt werden.

Patentansprüche

1. Computer-Implementiertes Verfahren zum Erlangen von Versionen eines Audio- und/oder Video-Programms mit stufenweise höherer Qualität über ein Client-Server-basiertes Netzwerk, das einen Client-Computer einschließt, der die folgenden Verfahrensschritte ausführt:

Abfragen (**202**) einer Version des Programms mit Grundqualität von einem Server über das Netzwerk, wobei die Version des Programms mit Grundqualität mindestens einen Basis-Layer eines Schicht-Unicast

umfasst;

Empfangen (**206**) und Zwischenspeichern (**208**) von Daten des abgefragten Layers, die zugehörig zu der Version des Programms mit Grundqualität sind; Abfragen (**224**) mindestens eines Erweiterungs-Layers des Schicht-Unicast von dem Server über das Netzwerk;

Empfangen (**228**) und Zwischenspeichern (**228**) von Daten des abgefragten Erweiterungs-Layers; Verknüpfen (**230**) der Daten des abgefragten Erweiterungs-Layers, während sie Empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu der Version des Programms mit Grundqualität sind, um eine Version des Programms mit höherer Qualität zu erzeugen; und Rendern (**232**, **316**) der Version des Programms mit höherer Qualität von den verknüpften Layer-Daten, und sie dem Benutzer präsentieren.

2. Verfahren nach Anspruch 1, des Weiteren einen Verfahrensschritt des Rendern (**206**) der Version des Programms mit Grundqualität umfassend, während die Abgefragten Daten empfangen werden, und sie dem Benutzer präsentieren.

3. Verfahren nach Anspruch 2, das des Weiteren die folgenden Verfahrensschritte umfasst: Ermitteln (**210**, **234**), ob der Benutzer anweist (**212**, **300**), dass die Präsentation der Version des Programms mit Grundqualität abgebrochen wird; Abbrechen der Präsentation der Version des Programms mit Grundqualität für den Benutzer.

4. Verfahren nach Anspruch 3, wobei der Verfahrensschritt des Abbrechens der Präsentation den Schritt des Abbrechens (**214**, **304**) des eingehenden Datenstroms, der zugehörig zu der Version des Programms mit Grundqualität ist, umfasst.

5. Verfahren nach Anspruch 3, wobei der Verfahrensschritt des Abbrechens der Präsentation die Schritte des Stoppens des Rendern der Version des Programms mit Grundqualität umfasst, während das Empfangen und Zwischenspeichern des eingehenden Datenstroms, der zugehörig zu der Version des Programms mit Grundqualität ist, fortgesetzt wird.

6. Verfahren nach Anspruch 1, das des Weiteren die folgenden Verfahrensschritte umfasst: Ermitteln (**212**), ob der Benutzer anweist (**218**), dass die Präsentation der Version des Programms mit höherer Qualität abgebrochen wird; Abbrechen der Präsentation der Version des Programms mit höherer Qualität für den Benutzer.

7. Verfahren nach Anspruch 6, wobei der Verfahrensschritt des Abbrechens der Präsentation den Schritt des Abbrechens (**214**) des eingehenden Datenstroms, der zugehörig zu der Version des Programms mit höherer Qualität ist, umfasst.

8. Verfahren nach Anspruch 6, wobei der Verfahrensschritt des Abbrechens der Präsentation die Schritte des Stoppens des Rendern der Version des Programms mit höherer Qualität umfasst, während das Empfangen und Zwischenspeichern des eingehenden Datenstroms, der zugehörig zu der Version des Programms mit höherer Qualität ist, fortgesetzt wird.

9. Verfahren nach Anspruch 1, wobei die Verfahrensschritte des Abfragens mindestens eines Erweiterungs-Layers, des Empfangens und Zwischenspeicherns von Daten des abgefragten Erweiterungs-Layers und des Verknüpfens der Daten des abgefragten Erweiterungs-Layers, während sie empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu der Version des Programms mit Grundqualität sind, um die Version des Programms mit höherer Qualität zu erzeugen, nur ausgeführt werden, wenn ein Benutzer den Client anweist (218), eine Version des Programms mit höherer Qualität im Vergleich zu der Version mit Grundqualität bereitzustellen.

10. Verfahren nach Anspruch 1, wobei die Verfahrensschritte des Abfragens (306) mindestens eines Erweiterungs-Layers, des Empfangens (310) und Zwischenspeicherns (310) von Daten des abgefragten Erweiterungs-Layers und des Verknüpfens (314) der Daten des abgefragten Erweiterungs-Layers, während sie empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu der Version des Programms mit Grundqualität sind, um die Version des Programms mit höherer Qualität zu erzeugen (316), automatisch ausgeführt (302) werden, sobald alle abgefragten Layer-Daten, die zugehörig zu der Version des Programms mit Grundqualität sind, empfangen und zwischengespeichert worden sind.

11. Verfahren nach Anspruch 1, das des Weiteren die folgenden Verfahrensschritte umfasst: Abfragen mindestens eines zusätzlichen Erweiterungs-Layers des Schicht-Unicast von dem Server über das Netzwerk; Empfangen und Zwischenspeichern von Daten des abgefragten zusätzlichen Erweiterungs-Layers; und Verknüpfen der Daten des abgefragten zusätzlichen Erweiterungs-Layers, während sie empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu den Versionen des Programms mit Grundqualität und mit höherer Qualität sind, um eine erweiterte Version des Programms mit höherer Qualität zu erzeugen.

12. Verfahren nach Anspruch 1, das des Weiteren die folgenden Verfahrensschritte umfasst: Feststellen (220), ob der Server irgendwelche übrigen dem Programm zugehörigen Erweiterungs-Layer verfügbar hat; und

sooft festgestellt wird, dass der Server mindestens einen übrigen dem Programm zugehörigen Erweiterungs-Layer verfügbar hat,

Abfragen (224) mindestens eines zusätzlichen Erweiterungs-Layers des Schicht-Unicast von dem Server über das Netzwerk,

Empfangen und Zwischenspeichern (228) von Daten des abgefragten zusätzlichen Erweiterungs-Layers, und

Verknüpfen (230) der Daten des abgefragten zusätzlichen Erweiterungs-Layers, während sie empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu den Versionen des Programms mit Grundqualität und mit höherer Qualität sind, um eine erweiterte Version des Programms mit höherer Qualität zu erzeugen.

13. Verfahren nach Anspruch 12, wobei die Verfahrensschritte des Abfragens mindestens eines zusätzlichen Erweiterungs-Layers, des Empfangens und Zwischenspeicherns von Daten des abgefragten zusätzlichen Erweiterungs-Layers und des Verknüpfens der Daten des abgefragten zusätzlichen Erweiterungs-Layers mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu den Versionen des Programms mit Grundqualität und mit höherer Qualität sind, um die erweiterte Version des Programms mit höherer Qualität zu erzeugen, nur ausgeführt werden, wenn ein Benutzer den Client anweist (218), eine erweiterte Version des Programms mit höherer Qualität bereitzustellen.

14. Verfahren nach Anspruch 12, wobei die Verfahrensschritte des Abfragens mindestens eines zusätzlichen Erweiterungs-Layers, des Empfangens und Zwischenspeicherns von Daten des abgefragten zusätzlichen Erweiterungs-Layers und des Verknüpfens der Daten des abgefragten zusätzlichen Erweiterungs-Layers mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu den Versionen des Programms mit Grundqualität und mit höherer Qualität sind, um die erweiterte Version des Programms mit höherer Qualität zu erzeugen, automatisch ausgeführt (302) werden, sobald alle abgefragten Layer-Daten, die zugehörig zu der Version des Programms mit höherer Qualität sind, empfangen und zwischengespeichert worden sind.

15. Verfahren nach Anspruch 12, des Weiteren einen Verfahrensschritt des Informierens (222) des Benutzers umfassend, dass eine erweiterte Version des Programms mit höherer Qualität nicht bereitgestellt werden kann, sooft festgestellt wird, dass der Server keine übrigen dem Programm zugehörigen Erweiterungs-Layer verfügbar hat.

16. Verfahren nach Anspruch 1, wobei die Layer des Schicht-Unicast in einer hierarchischen Beziehung stehen, in der der Layer der untersten Ebene ein Grund-Layer ist, und jeder nachfolgende höhere

Layer erweiternde Informationen zum Erweitern der Qualität des Programms hinzufügt, der von den Layern ausgehend, die ihm in der Hierarchie vorausgehen, gerendert werden kann, und wobei der Verfahrensschritt des Abfragens einer Version des Programms mit Grundqualität von einem Server über das Netzwerk den Schritt des Abfragens von so vielen Layern in der Reihenfolge ihrer Position in der Hierarchie, beginnend mit dem Grund-Layer, umfasst, wie von dem Server zu dem Client überfragt werden können, ohne die verfügbare Bandbreite des Netzwerks zu überschreiten.

17. Verfahren nach Anspruch 16, wobei der Verfahrensschritt des Abfragens mindestens eines Erweiterungs-Layers den Verfahrensschritt des Abfragens von so vielen Erweiterungs-Layern in der Reihenfolge ihrer Position in der Hierarchie, beginnend mit dem nächst höheren Layer in der Hierarchie ausgehend von dem Layer der höchsten Ebene, der in Verbindung mit der Version des Programms mit Grundqualität abgefragt wurde, umfasst, wie von dem Server zu dem Client übertragen werden können, ohne die verfügbare Bandbreite des Netzwerks zu überschreiten.

18. Verfahren nach Anspruch 1, wobei der Verfahrensschritt des Abfragens einer Version des Programms mit Grundqualität und des Abfragens mindestens eines Erweiterungs-Layers das Abfragen umfasst, dass die Daten, die jeden Layer bilden, in vollem Umfang bereitgestellt werden.

19. Verfahren nach Anspruch 1, wobei der Verfahrensschritt des Abfragens einer Version des Programms mit Grundqualität den Schritt des Abfragens der Daten, die jeden Layer der Version mit Grundqualität bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen umfasst, so dass die Layer-Teile, die zugehörig zu einem Zeitsegment am Anfang des Programms sind, zuerst abgefragt werden, und anschliessend die Layer-Teile, die zugehörig zu dem sequenziell nächstem Zeitsegment des Programms sind, abgefragt werden, und so weiter.

20. Verfahren nach Anspruch 19, wobei die Layer des Schicht-Unicast in einer hierarchischen Beziehung stehen, in der der Layer der untersten Ebene ein Grund-Layer ist, und jeder Layer einer höheren Ebene erweiternde Informationen zum Erweitern der Qualität des Programms hinzufügt, der von den Layern ausgehend, die ihm in der Hierarchie vorausgehen, gerendert werden kann, und wobei der Verfahrensschritt des Abfragens der Daten, die jeden Layer der Version mit Grundqualität bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen den Schritt des Abfragens der Layer-Teile von so vielen Layern in der Reihenfolge ihrer Position in der Hierarchie, beginnend mit dem Grund-Layer, umfasst, wie von dem Server zu dem Client übertragen

werden können, ohne die verfügbare Bandbreite des Netzwerks zu überschreiten.

21. Verfahren nach Anspruch 19, wobei der Verfahrensschritt des Abfragens mindestens eines Erweiterungs-Layers des Programms den Schritt des Abfragens von Daten, die jeden Erweiterungs-Layer bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen umfasst, so dass die Layer-Teile, die zugehörig zu einem Zeitsegment am Anfang des Programms sind, zuerst abgefragt werden, und anschliessend die Layer-Teile, die zugehörig zu dem sequenziell nächstem Zeitsegment des Programms sind, abgefragt werden, und so weiter.

22. Verfahren nach Anspruch 21, wobei der Verfahrensschritt des Abfragens von Daten, die jeden Erweiterungs-Layer bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen den Schritt des Abfragens der Erweiterungs-Layer-Teile von so vielen Erweiterungs-Layern in der Reihenfolge ihrer Position in der Hierarchie, beginnend mit dem nächst höheren Layer in der Hierarchie ausgehend von dem Layer der höchsten Ebene, der in Verbindung mit der Version des Programms mit Grundqualität abgefragt wurde, umfasst, wie von dem Server zu dem Client übertragen werden können, ohne die verfügbare Bandbreite des Netzwerks zu überschreiten.

23. Verfahren nach Anspruch 21, wobei die Länge jedes Zeitsegments des Programms der Rate, mit der die verfügbare Bandbreite auf dem Netzwerk schwankt, angeglichen wird, so dass jedes Zeitsegment kurz genug ist, dass die Netzwerkbandbreite nicht massgeblich über die Periode hinaus schwankt.

24. Computer-Implementiertes Verfahren zum Bereitstellen von Versionen mit stufenweise höherer Qualität eines Audio- und/oder Video-Programms über ein Client-Server-basiertes Netzwerk, das einen Client-Computer, der angepasst ist, um die Verfahrensschritte von einem der Ansprüche 1 bis 23 auszuführen, und einen Server-Computer einschliesst, wobei das Computer-Implementierte Verfahren den Server-Computer umfasst, und wobei der Server-Computer folgenden Verfahrensschritte ausführt: Bereitstellen (204) der Version des Programms mit Grundqualität an den Client-Computer über das Netzwerk; und Bereitstellen (226) des mindestens einen Erweiterungs-Layers des Schicht-Unicast an den Client über das Netzwerk, ohne zusätzlich irgendeinen Layer der Version des Programms mit Grundqualität bereitzustellen.

25. Verfahren nach Anspruch 24, wobei der Verfahrensschritt des Bereitstellens der Version des Programms mit Grundqualität den Schritt des Bereitstellens der Daten, die jeden Layer der Version mit

Grundqualität bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen umfasst, so dass die Layer-Teile, die zugehörig zu einem Zeitsegment am Anfang des Programms sind, zuerst bereitgestellt werden, und anschliessend die Layer-Teile, die zugehörig zu dem sequenziell nächstem Zeitsegment des Programms sind, bereitgestellt werden, und so weiter.

26. Verfahren nach Anspruch 24, wobei der Verfahrensschritt des Bereitstellens mindestens eines Erweiterungs-Layers des Programms den Schritt des Bereitstellens von Daten, die jeden Erweiterungs-Layer bilden, in sequenziellen, gleichgrossen, zeitlich in Beziehung stehenden Teilen umfasst, so dass die Layer-Teile, die zugehörig zu einem Zeitsegment am Anfang des Programms sind, zuerst bereitgestellt werden, und anschliessend die Layer-Teile, die zugehörig zu dem sequenziell nächstem Zeitsegment des Programms sind, bereitgestellt werden, und so weiter.

27. Verfahren nach Anspruch 24, den Server-Computer umfassend, der folgenden Verfahrensschritte ausführt:

Bereitstellen der Version des Programms mit Grundqualität an den Client-Computer über das Netzwerk auf eine erste Anfrage durch den Client-Computer über das Netzwerk, das Programm bereitzustellen; auf eine nachfolgende Anfrage durch den Client-Computer über das Netzwerk, eine Version des Programms mit höherer Qualität bereitzustellen,
 – Auffordern zum Bezahlen einer vorgeschriebenen Gebühr,
 – Bereitstellen mindestens eines Erweiterungs-Layers des Schicht-Unicast an den Client über das Netzwerk auf das Bezahlen der vorgeschriebenen Gebühr hin.

28. Verfahren nach Anspruch 27, wobei der Verfahrensschritt des Aufforderns zum Bezahlen der vorgeschriebenen Gebühr nur auf eine erste Anfrage von dem Client-Computer über das Netzwerk, eine Version des Programms mit höherer Qualität bereitzustellen, hin ausgeführt wird, und danach nicht mehr wiederholt wird.

29. Computer-lesbares Medium, das Computer-ausführbare Instruktionen zum Bereitstellen von Versionen eines Audio- und/oder Video-Programms mit stufenweise höherer Qualität über ein Client-Server-basiertes Netzwerk enthält, wobei die Computer-ausführbaren Instruktionen die Verfahrensschritte nach einem der Ansprüche 1 bis 26 umfassen.

30. Client-Server-basiertes Computernetzwerk zum Erlangen von Versionen eines Audio- und/oder Video-Programms mit stufenweise höherer Qualität, wobei das Netzwerk umfasst:
 einen Client, der mindestens ein Allzweck-Computer-

gerät umfasst; und

ein Computerprogramm, dass Programmmodule umfasst, die durch den Client ausführbar sind, wobei der Client durch die Programmmodule angewiesen wird zum:

- Empfangen (**200**) von Instruktionen von einem Benutzer, um das Programm zum Anschauen bereitzustellen,
- Abfragen (**202**) einer Version des Programms mit Grundqualität von einem Server über das Netzwerk, wobei die Version des Programms mit Grundqualität mindestens einen Basis-Layer eines Schicht-Unicast umfasst,
- Empfangen (**206**) und Zwischenspeichern (**206**) von Daten des abgefragten Layers, die zugehörig zu der Version des Programms mit Grundqualität sind,
- Rendern (**206**) der Version des Programms mit Grundqualität, während die abgefragten Daten empfangen werden, und Präsentieren dieser dem Benutzer,
- Ermitteln (**210**), ob der Benutzer anweist (**218**), eine Version des Programms mit höherer Qualität zum Anschauen bereitzustellen,
- Sooft ermittelt wird, dass der Benutzer angewiesen hat, eine Version des Programms mit höherer Qualität bereitzustellen,
- Abfragen (**224**) mindestens eines Erweiterungs-Layers des Schicht-Unicast von dem Server über das Netzwerk,
- Empfangen (**228**) und Zwischenspeichern (**228**) von Daten des abgefragten Erweiterungs-Layers,
- Verknüpfen (**230**) der Daten des abgefragten Erweiterungs-Layers, während sie empfangen werden, mit den zuvor zwischengespeicherten Layer-Daten, die zugehörig zu der Version des Programms mit Grundqualität sind, um eine Version des Programms mit höherer Qualität zu erzeugen, und
- Rendern (**232**) der Version des Programms mit höherer Qualität von den verknüpften Layer-Daten, und sie dem Benutzer präsentieren.

31. Netzwerk nach Anspruch 30, das des Weiteren ein Programmmodul umfasst zum:

Ermitteln (**210**, **234**), ob der Benutzer anweist (**212**), dass die Präsentation der Version des Programms mit Grundqualität abgebrochen wird;
 Sooft ermittelt wird, dass der Benutzer angewiesen hat, die Präsentation der Version des Programms mit Grundqualität abzubrechen, Abbrechen (**214**, **304**) der Präsentation.

32. Netzwerk nach Anspruch 30, des Weiteren umfassend:

einen Server, der mindestens ein Allzweck-Computergerät umfasst; und

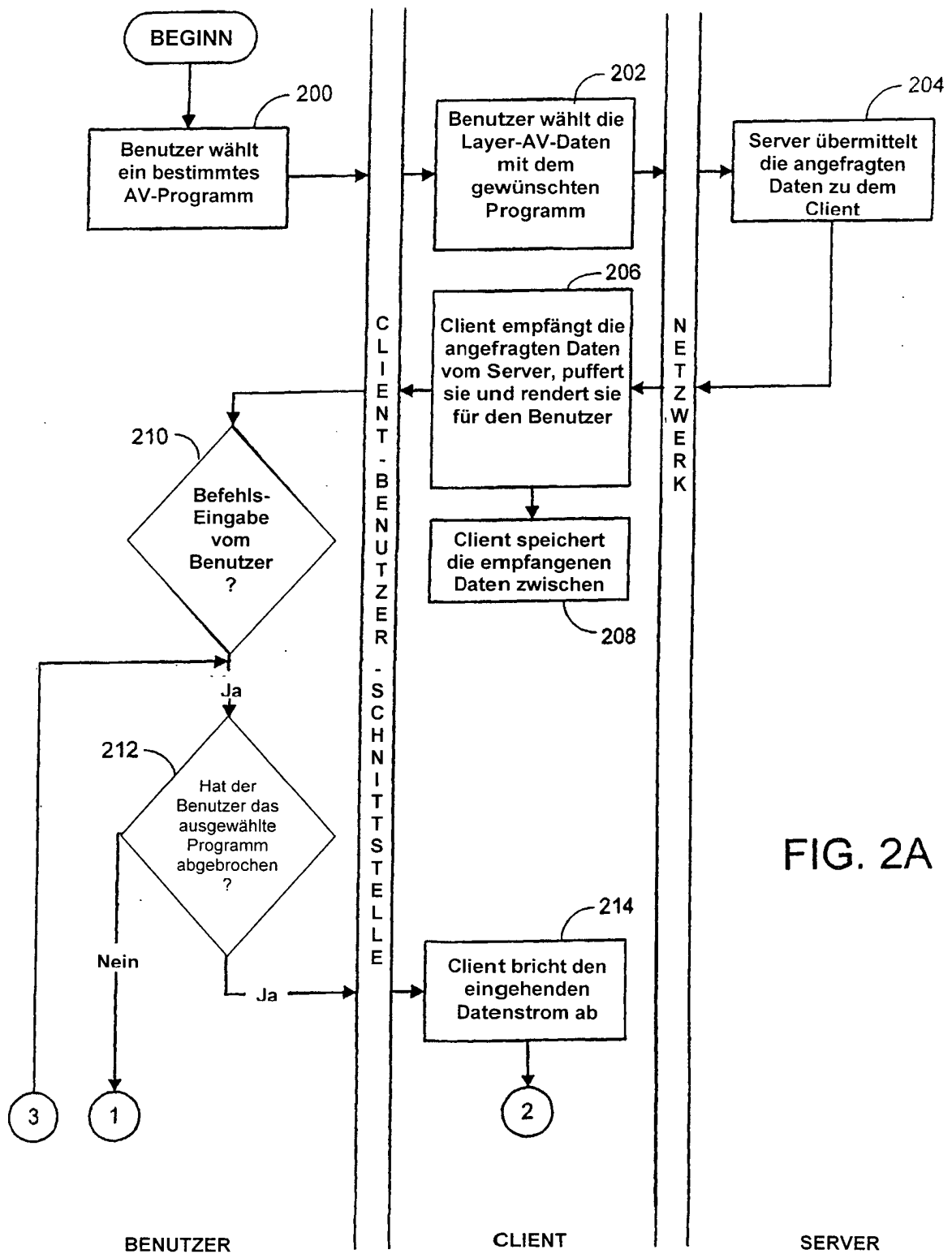
ein Computerprogramm, dass Programmmodule umfasst, die durch den Server ausführbar sind, wobei der Server durch die Programmmodule angewiesen wird zum:

- Bereitstellen (**204**) der Version des Programms mit

Grundqualität an den Client-Computer über das Netzwerk auf eine erste Anfrage von dem Client-Computer über das Netzwerk, das Programm bereitzustellen; und

– Bereitstellen **(226)** mindestens eines Erweiterungs-Layers des Schicht-Unicast an den Client über das Netzwerk auf eine nachfolgende Anfrage von dem Client-Computer über das Netzwerk, eine Version des Programms mit höherer Qualität bereitzustellen, ohne zusätzlich irgendeinen Layer der Version des Programms mit Grundqualität bereitzustellen.

Es folgen 5 Blatt Zeichnungen



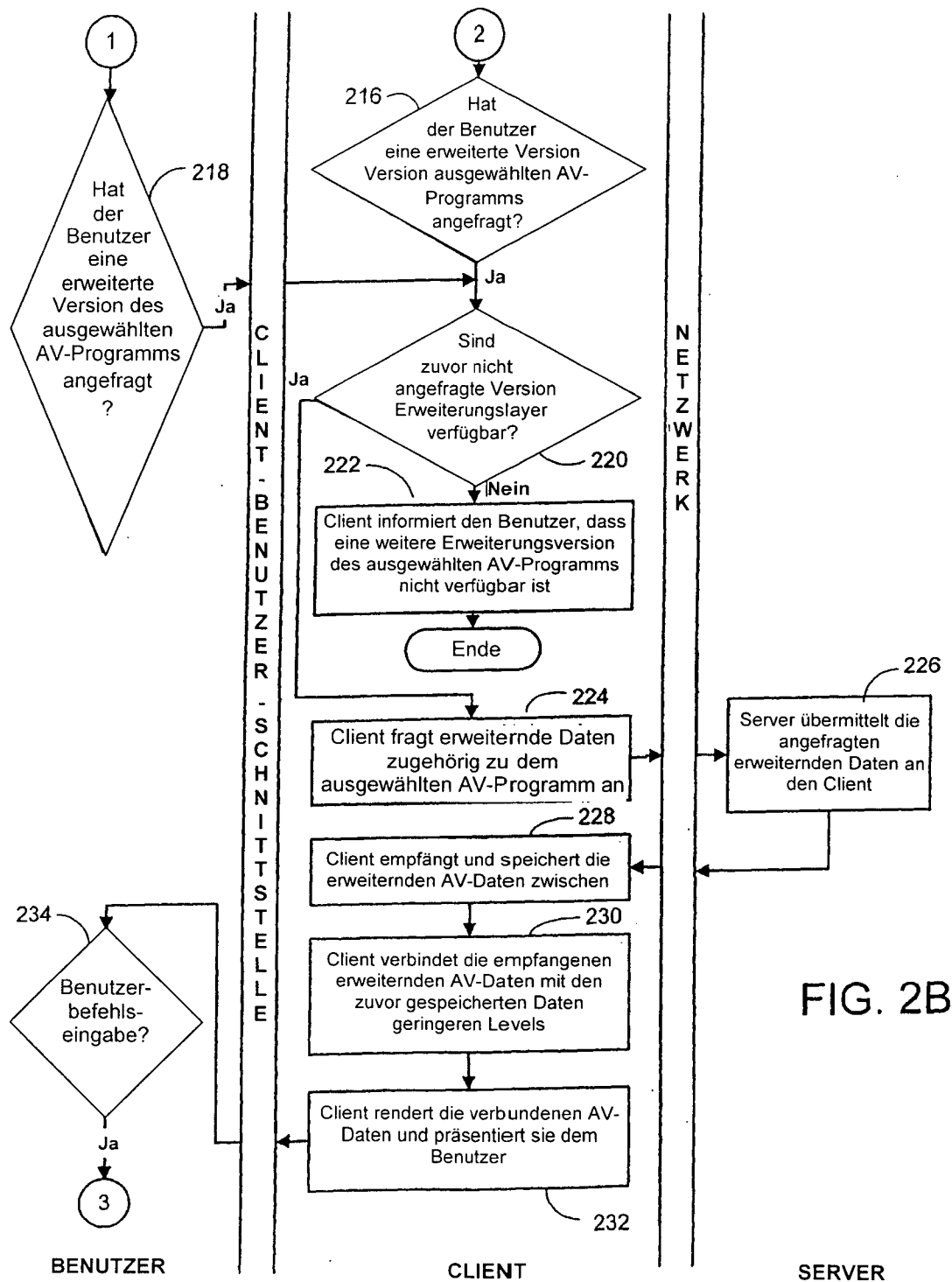


FIG. 2B

