

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4487711号
(P4487711)

(45) 発行日 平成22年6月23日(2010.6.23)

(24) 登録日 平成22年4月9日(2010.4.9)

(51) Int.Cl.

F 1

H04L 29/06 (2006.01)

H04L 13/00 305C

H04L 12/56 (2006.01)

H04L 12/56 100A

H04N 7/173 (2006.01)

H04N 7/173 620D

請求項の数 17 (全 30 頁)

(21) 出願番号

特願2004-282086 (P2004-282086)

(22) 出願日

平成16年9月28日 (2004.9.28)

(65) 公開番号

特開2005-244929 (P2005-244929A)

(43) 公開日

平成17年9月8日 (2005.9.8)

審査請求日

平成19年9月6日 (2007.9.6)

(31) 優先権主張番号

特願2004-23126 (P2004-23126)

(32) 優先日

平成16年1月30日 (2004.1.30)

(33) 優先権主張国

日本国 (JP)

(73) 特許権者 000002185

ソニー株式会社

東京都港区港南1丁目7番1号

(74) 代理人 100082131

弁理士 稲本 義雄

(72) 発明者 河村 聖悟

東京都品川区北品川6丁目7番35号 ソ

ニー株式会社内

(72) 発明者 楠 忍

東京都品川区北品川6丁目7番35号 ソ

ニー株式会社内

審査官 阿部 弘

最終頁に続く

(54) 【発明の名称】送信装置および方法、受信装置、通信システム、記録媒体、並びにプログラム

(57) 【特許請求の範囲】

【請求項 1】

トランスポート層の第1のプロトコルによりデータを送信する複数の第1の送信手段と、

前記第1の送信手段のポート番号、および、利用ポート数を含む、前記データを再生するための制御情報を生成する生成手段と、

前記生成手段により生成された前記制御情報をトランスポート層の第2のプロトコルで送信する第2の送信手段と、

セッション層のプロトコルにより送信するデータを前記複数の第1の送信手段のそれぞれの動作状態に応じて割振って送信するように制御する制御手段と

を備えることを特徴とする送信装置。

【請求項 2】

前記制御手段は、セッション層のプロトコルにより送信するデータを前記複数の第1の送信手段のそれぞれの動作状態に応じて、ラウンドロビン方式により割振って送信するよう

に制御する

ことを特徴とする請求項1に記載の送信装置。

【請求項 3】

前記制御手段は、セッション層のプロトコルにより送信するデータを前記複数の第1の送信手段のそれぞれの動作状態に応じて、送信可能なものに割振って送信するように制御する

10

20

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 4】

前記セッション層のプロトコルにより送信するデータを複数のデータに分割する分割手段をさらに備え、

前記制御手段は、前記分割手段により分割された複数のデータを、それぞれ複数の第 1 の送信手段のそれぞれの動作状態に応じて割振ってトランスポート層の第 1 のプロトコルで送信するように制御する

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 5】

前記生成手段は、前記第 1 の送信手段のポート番号、および、利用ポート数を含む、前記分割手段により前記データが分割される際、分割されたデータを結合して再生させるための制御情報を生成する

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 6】

前記制御手段は、前記セッション層のプロトコルにより送信する同一のデータを、それぞれ複数の第 1 の送信手段によりトランスポート層の第 1 のプロトコルで送信するように制御する

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 7】

前記生成手段は、前記第 1 の送信手段のポート番号、および、利用ポート数を含む、前記セッション層のプロトコルにより送信するデータを識別して、前記データを再生するための制御情報を生成する

ことを特徴とする請求項 6 に記載の送信装置。

【請求項 8】

前記第 2 のプロトコルはTCPである

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 9】

前記セッション層のプロトコルはRTPである

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 10】

前記トランスポート層の第 1 のプロトコルは、送信したデータの着信確認をしないプロトコルである

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 11】

前記送信したデータの着信確認をしない、前記トランスポート層の第 1 のプロトコルは、UDPである

ことを特徴とする請求項 10 に記載の送信装置。

【請求項 12】

前記トランスポート層の第 1 のプロトコルは、TCPである

ことを特徴とする請求項 1 に記載の送信装置。

【請求項 13】

トランスポート層の第 1 のプロトコルによりデータを送信する複数の第 1 の送信ステップと、

前記第 1 の送信ステップの処理でのポート番号、および、利用ポート数を含む、前記データを再生するための制御情報を生成する生成ステップと、

前記生成ステップの処理で生成された前記制御情報をトランスポート層の第 2 のプロトコルで送信する第 2 の送信ステップと、

セッション層のプロトコルにより送信するデータを前記複数の第 1 の送信手段のそれぞれの動作状態に応じて送信するように制御する制御ステップと

を含むことを特徴とする送信方法。

10

20

30

40

50

【請求項 14】

トランスポート層の第1のプロトコルによりデータの送信を制御する複数の第1の送信制御ステップと、

前記第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、前記データを再生するための制御情報の生成を制御する生成制御ステップと、

前記生成制御ステップの処理で生成された前記制御情報をトランスポート層の第2のプロトコルでの送信を制御する第2の送信制御ステップと、

セッション層のプロトコルにより送信するデータを前記複数の第1の送信制御ステップの処理のそれぞれの動作状態に応じて送信するように制御する動作制御ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。 10

【請求項 15】

トランスポート層の第1のプロトコルによりデータの送信を制御する複数の第1の送信制御ステップと、

前記第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、前記データを再生するための制御情報の生成を制御する生成制御ステップと、

前記生成制御ステップの処理で生成された前記制御情報をトランスポート層の第2のプロトコルでの送信を制御する第2の送信制御ステップと、

セッション層のプロトコルにより送信するデータを前記複数の第1の送信制御ステップの処理のそれぞれの動作状態に応じて送信するように制御する動作制御ステップと 20

をコンピュータに実行させることを特徴とするプログラム。

【請求項 16】

トランスポート層の第1のプロトコルによりデータを受信する複数の第1の受信手段と、

前記第1の受信手段のポート番号、および、利用ポート数を含む、前記データを再生するための制御情報をトランスポート層の第2のプロトコルで受信する第2の受信手段と、

セッション層のプロトコルによる受信データとして、前記データを前記制御情報に基づいて再生する再生手段と

を備えることを特徴とする受信装置。 30

【請求項 17】

送信装置と受信装置からなる通信システムにおいて、

前記送信装置は、

トランスポート層のプロトコルによりデータを送信する複数の送信手段と、

前記第1の送信手段のポート番号、および、利用ポート数を含む、前記データを再生するための制御情報を生成する生成手段と、

前記生成手段により生成された前記制御情報をトランスポート層の第2のプロトコルで送信する第2の送信手段と、

セッション層のプロトコルにより送信するデータを前記複数の第1の送信手段のそれぞれの動作状態に応じて割振って送信するように制御する制御手段と 40

を備え、

前記受信装置は、

前記トランスポート層のプロトコルによりデータを受信する受信手段と、

前記第1の受信手段のポート番号、および、利用ポート数を含む、前記データを再生するための制御情報をトランスポート層の第2のプロトコルで受信する第2の受信手段と、

セッション層のプロトコルによる受信データとして、前記データを前記制御情報に基づいて再生する再生手段と

を備える

ことを特徴とする通信システム。

【発明の詳細な説明】

【技術分野】**【0001】**

本発明は、送信装置および方法、受信装置、通信システム、記録媒体、並びにプログラムに関し、特に、データの通信におけるデータの遅延を抑制し、安定した通信を実現できるようにした送信装置および方法、受信装置、通信システム、記録媒体、並びにプログラムに関する。

【背景技術】**【0002】**

大容量のストリーミングデータなどの再生時刻を保障してデータを配信する技術が一般に普及しつつある。

10

【0003】

インターネット上のルータでは、フォワード（転送）するデータのフローが監視され、大量データの送信が行われているフローのフォワード優先度を下げるといったフロー制御が行われることがある。一般的なルータにおいて、データのフローは、送信元および送信先のIPアドレス、トランスポートプロトコル、並びに、送信元および送信先のポート番号によって識別される。

【0004】

以上の通信の一例として、RTP (Real Time Protocol) 方式がある。RTP方式の場合、配信サーバとクライアントPC (Personal Computer)との間で、ストリーミングデータの通信が行われるとき、ストリーミング制御は、RTSP (Real Time Streaming Protocol) によって行われる。また、データの送信制御はRTCP (RTP Control Protocol) が利用され、動画の転送はRTPによって行われる。従来のRTPを利用した動画転送技術の場合、1つのRTPセッションについて、1つのポートを利用する事で、動画転送が実現されている。

20

【0005】

また、ビットストリームを分割し、異なるポートで送受信することが提案されている（例えば、特許文献1参照）。

【0006】

さらに、端末の能力に応じた最適なパケット通信処理を実現させるものが提案されている（例えば、特許文献2参照）。

30

【特許文献1】特開2002-017637号公報**【特許文献2】特開2003-152544号公報****【発明の開示】****【発明が解決しようとする課題】****【0007】**

ところで、従来のRTP方式による動画転送を行う場合、インターネット上のルータは、フォワードするデータのフローを監視しており、大量データの送信を行っているフローのフォワード優先度を下げるといったフロー制御によって、通信が過大に帯域を占有しないように制御している。

【0008】

また、例えば、ルータは、過負荷の状態になって、さらにバッファがいっぱいになると、受信したパケットを破棄するように制御している。

40

【0009】

これに対応するため、TCP (Transmission Control Protocol)においては、ルータ上でパケットが破棄される等の理由によって、輻輳が発生した場合、一旦速度を落とし、パケット長の倍数にあたるウィンドウサイズ（通信可能なデータ量）を徐々に引き上げながら最適な通信速度を模索する（スロースタート）。

【0010】

ここで、輻輳について説明する。例えば、TCPなどにおいて、通信経路上でデータが欠落した場合、再送要求が送信先から送信元に送信されることになるが、連続的にデータが欠落したような場合、再送要求も連続的に発生することになり、結果として、さらに通信

50

経路が込み合って、通信経路上でのデータの欠落をさらに悪化させてしまい、データの転送が安定的にできない状態となる。このように安定的にデータの転送ができない状態を輻輳と言う。例えば、図1で示されるように、輻輳は、通信が開始されてから、多少遅れて生じる。尚、図1においては、横軸が時間であり、縦軸が単位時間あたりに送信先で受信されるデータの量(byte)を示したものである。

【0011】

すなわち、図1で示されるように、通信が開始された当初は、データの転送量によるフロー制御がないため通信速度がある程度維持されるが、通信経路上の各所に存在するルータによりデータの破棄などが発生することにより、通信の開始からある程度の時間が経過すると、再送要求が頻繁に送信先から送られてくるため、再送するデータや、再送要求が通信経路をさらに混雑させて、最終的には、図1中のAで示されるように輻輳が生じる結果となる。10

【0012】

このため、2重3重に輻輳が発生すると、最適な通信速度を見つけ出すまでに、非常に時間がかかってしまい、伝送速度の遅い通信となってしまうことがあり、さらに、このような状況により、例えば、ストリーミングデータのバッファリングが不足すると、ストリーミングデータの再生が中断され、スムーズなストリーミングデータの再生ができない状態となることがあった。

【0013】

また、このような状況を回避するため、例えば、TCPの代わりにUDPを利用することも考えられるが、UDP(User Datagram Protocol)においては、TCPのようなフロー制御や、輻輳制御が行われないため、高速に通信することが可能とはなるが、送信データが確実に送られているかが確認されないので、データの到着には保障がなされないといった問題があった。20

【0014】

本発明はこのような状況に鑑みてなされたものであり、特に、セッション層における1セッションの通信データを複数のトランスポート層のポートを利用して送信することにより、高速で、かつ、安定的なストリーミングデータの通信を実現できるようにするものである。

【課題を解決するための手段】

【0015】

本発明の送信装置は、トランスポート層の第1のプロトコルによりデータを送信する複数の第1の送信手段と、第1の送信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報を生成する生成手段と、生成手段により生成された制御情報をトランスポート層の第2のプロトコルで送信する第2の送信手段と、セッション層のプロトコルにより送信するデータを複数の第1の送信手段のそれぞれの動作状態に応じて割振って送信するように制御する制御手段とを備えることを特徴とする。30

【0016】

前記制御手段には、セッション層のプロトコルにより送信するデータを複数の第1の送信手段のそれぞれの動作状態に応じて、ラウンドロビン方式により割振って送信するよう⁴⁰に制御せざるようにすることができる。

【0017】

前記制御手段には、セッション層のプロトコルにより送信するデータを複数の第1の送信手段のそれぞれの動作状態に応じて、送信可能なものに割振って送信するよう⁴⁰に制御せざるようにすることができる。

【0018】

前記セッション層のプロトコルにより送信するデータを複数のデータに分割する分割手段をさらに設けるようにさせることができ、制御手段には、分割手段により分割された複数のデータを、それぞれ複数の第1の送信手段によりトランスポート層の第1のプロトコルで送信するよう⁵⁰に制御せざるようにすることができる。

【0019】

前記生成手段には、第1の送信手段のポート番号、および、利用ポート数を含む、分割手段によりデータが分割される際、分割されたデータを結合して再生させるための制御情報を生成するようにすることができる。

【0020】

前記制御手段には、セッション層のプロトコルにより送信する同一のデータを、それぞれ複数の第1の送信手段によりトранSPORT層の第1のプロトコルで送信するように制御せしめるようにすることができる。

【0021】

前記生成手段には、第1の送信手段のポート番号、および、利用ポート数を含む、セッション層のプロトコルにより送信するデータを識別して、データを再生するための制御情報を生成するようにすることができる。 10

【0022】

前記第2のプロトコルはTCPとするようにすることができる。

【0023】

前記セッション層のプロトコルはRTPとするようにすることができる。

【0024】

前記トランSPORT層の第1のプロトコルは、送信したデータの着信確認をしないプロトコルとするようにすることができる。

【0025】

前記送信したデータの着信確認をしない、トランSPORT層の第1のプロトコルは、UDPとするようにすることができる。 20

【0026】

前記トランSPORT層の第1のプロトコルは、TCPとするようにすることができる。

【0027】

本発明の送信方法は、トランSPORT層の第1のプロトコルによりデータを送信する複数の第1の送信ステップと、第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、データを再生するための制御情報を生成する生成ステップと、生成ステップの処理で生成された制御情報をトランSPORT層の第2のプロトコルで送信する第2の送信ステップと、セッション層のプロトコルにより送信するデータを複数の第1の送信ステップの処理のそれぞれの動作状態に応じて割振って送信するように制御する制御ステップとを含むことを特徴とする。 30

【0028】

本発明の第1の記録媒体のプログラムは、トランSPORT層の第1のプロトコルによりデータの送信を制御する複数の第1の送信制御ステップと、第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、データを再生するための制御情報の生成を制御する生成制御ステップと、生成制御ステップの処理で生成された制御情報をトランSPORT層の第2のプロトコルでの送信を制御する第2の送信制御ステップと、セッション層のプロトコルにより送信するデータを複数の第1の送信制御ステップの処理のそれぞれの動作状態に応じて割振って送信するように制御する動作制御ステップとを含むことを特徴とする。 40

【0029】

本発明の第1のプログラムは、トランSPORT層の第1のプロトコルによりデータの送信を制御する複数の第1の送信制御ステップと、第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、データを再生するための制御情報の生成を制御する生成制御ステップと、生成制御ステップの処理で生成された制御情報をトランSPORT層の第2のプロトコルでの送信を制御する第2の送信制御ステップと、セッション層のプロトコルにより送信するデータを複数の第1の送信制御ステップの処理のそれぞれの動作状態に応じて割振って送信するように制御する動作制御ステップとをコンピュータに実行させることを特徴とする。 50

【0030】

本発明の受信装置は、トランスポート層の第1のプロトコルによりデータを受信する複数の第1の受信手段と、第1の受信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報をトランスポート層の第2のプロトコルで受信する第2の受信手段と、セッション層のプロトコルによる受信データとして、データを制御情報に基づいて再生する再生手段とを備えることを特徴とする。

【0041】

本発明の通信システムは、送信装置が、トランスポート層のプロトコルによりデータを送信する複数の送信手段と、第1の送信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報を生成する生成手段と、生成手段により生成された制御情報をトランスポート層の第2のプロトコルで送信する第2の送信手段と、セッション層のプロトコルにより送信するデータを複数の第1の送信手段のそれぞれの動作状態に応じて割振って送信するように制御する制御手段とを備え、受信装置が、トランスポート層のプロトコルによりデータを受信する受信手段と、第1の受信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報をトランスポート層の第2のプロトコルで受信する第2の受信手段と、セッション層のプロトコルによる受信データとして、データを制御情報に基づいて再生する再生手段とを備えることを特徴とする。10

【0042】

本発明の送信装置および方法、並びにプログラムにおいては、トランスポート層の第1のプロトコルによりデータが複数の通信経路で送信され、ポート番号、および、利用ポート数を含む、データを再生するための制御情報が生成され、生成された制御情報がトランスポート層の第2のプロトコルで送信され、セッション層のプロトコルにより送信するデータが複数の通信経路のそれぞれの動作状態に応じて割振られて送信されるように制御される。20

【0043】

本発明の受信装置においては、トランスポート層の第1のプロトコルによりデータが複数の通信経路で受信され、ポート番号、および、利用ポート数を含む、データを再生するための制御情報がトランスポート層の第2のプロトコルで受信され、セッション層のプロトコルによる受信データとして、データが制御情報に基づいて再生される。

【0044】

本発明の通信システムにおいては、送信装置により、トランスポート層のプロトコルによりデータが複数の通信経路で送信され、ポート番号、および、利用ポート数を含む、データを再生するための制御情報が生成され、生成された制御情報がトランスポート層の第2のプロトコルで送信され、セッション層のプロトコルにより送信するデータが複数の通信経路のそれぞれの動作状態に応じて割振られて送信されるように制御され、受信装置により、トランスポート層のプロトコルによりデータが受信され、ポート番号、および、利用ポート数を含む、データを再生するための制御情報がトランスポート層の第2のプロトコルで受信され、セッション層のプロトコルによる受信データとして、データが制御情報に基づいて再生される。30

【0045】

本発明の送信装置、受信装置、または、通信システムは、独立した装置であっても良いし、送信処理または受信処理を行うブロックであっても良い。

【発明の効果】**【0046】**

本発明によれば、高速で、かつ、安定的な通信を実現させるようにすることができる。

【発明を実施するための最良の形態】**【0047】**

以下に本発明の実施の形態を説明するが、本明細書に記載の発明と、発明の実施の形態との対応関係を例示すると、次のようになる。この記載は、本明細書に記載されている発明をサポートする実施の形態が本明細書に記載されていることを確認するためのものである50

る。従って、発明の実施の形態中には記載されているが、発明に対応するものとして、ここには記載されていない実施の形態があったとしても、そのことは、その実施の形態が、その発明に対応するものではないことを意味するものではない。逆に、実施の形態が発明に対応するものとしてここに記載されていたとしても、そのことは、その実施の形態が、その発明以外の発明には対応しないものであることを意味するものでもない。

【0048】

さらに、この記載は、本明細書に記載されている発明の全てを意味するものではない。換言すれば、この記載は、本明細書に記載されている発明であって、この出願では請求されていない発明の存在、すなわち、将来、分割出願されたり、補正により出現、追加される発明の存在を否定するものではない。

10

【0049】

即ち、本発明の送信装置は、トランスポート層の第1のプロトコルによりデータを送信する複数の第1の送信手段（例えば、図4のUDPポート87乃至89）と、第1の送信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報を生成する生成手段（例えば、図4のMulti Flow RTP 83）と、生成手段により生成された制御情報をトランスポート層の第2のプロトコルで送信する第2の送信手段（例えば、図4のTCPポート86）と、セッション層のプロトコルにより送信するデータを複数の第1の送信手段のそれぞれの動作状態に応じて割振って送信するように制御する制御手段（例えば、図4のRTCP 84）とを備えることを特徴とする。

【0050】

前記セッション層のプロトコルにより送信するデータを複数のデータに分割する分割手段（例えば、図4のMulti Flow RTP 83）をさらに設けるようにさせることができ、制御手段には、分割手段により分割された複数のデータを、それぞれ複数の第1の送信手段のそれぞれの動作状態に応じて割振ってトランスポート層の第1のプロトコルで送信するように制御させるようになることができる。

20

【0051】

本発明の送信方法は、トランスポート層の第1のプロトコルによりデータを送信する複数の第1の送信ステップ（例えば、図7のフローチャートのステップS16の処理）と、第1の送信ステップの処理でのポート番号、および、利用ポート数を含む、データを再生するための制御情報を生成する生成ステップ（例えば、図7のフローチャートのステップS14の処理）と、生成ステップの処理で生成された制御情報をトランスポート層の第2のプロトコルで送信する第2の送信ステップ（例えば、図7のフローチャートのステップS15の処理）と、セッション層のプロトコルにより送信するデータを複数の第1の送信ステップの処理のそれぞれの動作状態に応じて割振って送信するように制御する制御ステップ（例えば、図7のフローチャートのステップS16の処理）とを含むことを特徴とする。

30

【0052】

本発明の受信装置は、トランスポート層の第1のプロトコルによりデータを受信する複数の第1の受信手段（例えば、図4のUDPポート67乃至69）と、第1の受信手段のポート番号、および、利用ポート数を含む、データを再生するための制御情報をトランスポート層の第2のプロトコルで受信する第2の受信手段（例えば、図4のTCPポート66）と、セッション層のプロトコルによる受信データとして、データを制御情報に基づいて再生する再生手段（例えば、図4のMulti Flow RTP 63）とを備えることを特徴とする。

40

【0054】

尚、記録媒体、プログラム、および、通信システムの対応関係については、上述した送信装置および方法、並びに、受信装置と同様であるので、その説明は省略する。

【0055】

図2は、本発明を適用した配信システムの一実施の形態の構成を示す図である。

【0056】

図2で示されている本発明の配信システムは、クライアントPC1より、ストリームデー

50

タの配信要求があると、配信サーバ7から要求のあったストリームデータをクライアントPC1に配信するシステムである。

【0057】

クライアントPC1は、家庭用のルータ2を介して、最寄のインターネットサービスプロバイダサーバ3に対して、URL(Universal Resource Locator)などによりアドレスを指定して、配信サーバ7にストリームデータの配信を要求する。また、クライアントPC1は、ストリームデータの配信があった場合、インターネットサービスプロバイダサーバ3より送信されてくるストリームデータを家庭用ルータ2を介して取得し、再生する。

【0058】

家庭用のルータ2(以下、単にルータ2とも称する)は、異なるネットワーク同士を相互接続するものである。より詳細には、ルータ2は、通信経路が記述されたルーティングテーブルに従って、送信先のネットワークまでの通信経路を中継する。ルータ2は、OSI(Open Systems Interconnection)参照モデル(OSI階層モデルとも称される)のネットワーク層以上で動作するため、ネットワークプロトコルに動作が依存する。ルータ2には、様々な形式があり、TCP/IP(Transmission Control Protocol/Internet Protocol)、IPX/SPX(Internetwork Packet Exchange/Sequenced Packet Exchange)、AppleTalk(商標)、および、SNA(Systems Network Architecture)などに様々なプロトコル対応するタイプのものがあり、これらは、一般に「マルチプロトコルルータ」と称されている。また、ルータ2は、上述したように、フロー制御や輻輳制御を行う。

【0059】

インターネットサービスプロバイダサーバ3は、インターネット接続サービスを提供する通信事業者により管理運営されるサーバである。インターネットサービスプロバイダは、サーバを所有している教育機関や団体、企業などに所属していればLAN経由でインターネットを利用できるが、サーバを持たない企業や個人でもインターネットを利用できるよう、公共のサーバを提供する通信事業者である。

【0060】

インターネットサービスプロバイダサーバ3は、クライアントPC1からの要求に基づいて、インターネットに代表されるネットワーク5を介して、URLで指定される配信サーバ7にストリームデータを要求する。また、インターネットサーバルプロバイダサーバ3は、この要求に応じてネットワーク5を介して配信サーバ7より配信されてくるストリームデータを、ルータ4を介して取得し、クライアントPC1に供給する。

【0061】

ルータ4、6は、ルータ2と同様のものであり、ルータ4は、主に、インターネットサービスプロバイダサーバ3とネットワーク5とを、また、ルータ6は、配信サーバ7とネットワーク5とをそれぞれ接続し、通信経路を形成している。

【0062】

ネットワーク5は、複数のルータ11a乃至11fにより構成され、それぞれのルータ11a乃至11fが相互に通信経路を構成する。このルータ11a乃至11fについても、ルータ2と基本的には同様のものである。尚、図2においては、ルータ11aと11b、ルータ11aと11c、ルータ11aと11d、ルータ11bと11c、ルータ11bと11f、ルータ11cと11d、ルータ11cと11f、ルータ11dと11e、および、ルータ11eと11fが相互に接続されて、通信経路が形成されている場合について示されているが、ルータ11a乃至11fは、これ以外の組み合わせで接続され、通信経路が形成されるようにしてもよいし、また、これらの数以外の数のルータによりネットワーク5が形成されるようにしても良い。さらに、ルータ11a乃至11fを特に区別する必要がない場合、単に、ルータ11と称するものとし、その他の構成についても同様とする。

【0063】

配信サーバ7は、インターネットサービスプロバイダサーバ3、および、ネットワーク5を介してクライアントPC1より要求のあったストリームデータを、要求のあったクライ

10

20

30

40

50

アントPC1に配信する。

【0064】

次に、図3を参照して、クライアントPC1の構成について説明する。

【0065】

CPU(Central Processing Unit)21は、ROM(Read Only Memory)22、または記憶部28に記憶されているプログラムに従って各種の処理を実行する。RAM(Random Access Memory)23には、CPU21が実行するプログラムやデータなどが適宜記憶される。これらのCPU21、ROM22、およびRAM23は、バス24により相互に接続されている。

【0066】

CPU21には、バス24を介して入出力インターフェース25が接続されている。入出力インターフェース25には、キーボード、マウス、マイクロフォンなどよりなる入力部26、ディスプレイ、スピーカなどよりなる出力部67が接続されている。CPU21は、入力部26から入力される指令に対応して各種の処理を実行する。

【0067】

入出力インターフェース25に接続されている記憶部28は、例えばハードディスクなどで構成され、CPU21が実行するプログラムや各種のデータを記憶する。通信部29は、例えば、モデムのようなものであり、図示せぬネットワークを介して通信機能を有する装置と各種のデータの授受を行う。

【0068】

入出力インターフェース25に接続されているドライブ30は、磁気ディスク41、光ディスク42、光磁気ディスク43、或いは半導体メモリ44などが装着されたとき、それらを駆動し、そこに記録されているプログラムやデータなどを取得する。取得されたプログラムやデータは、必要に応じて記憶部28に転送され、記憶される。

【0069】

尚、インターネットサービスプロバイダサーバ3、および、配信サーバ7の構成は、基本的に、クライアントPC1と同様の構成であるので、その説明は省略するが、ROM22、および、記憶部28、並びに、ドライブ30に装着された磁気ディスク41、光ディスク42、光磁気ディスク43、或いは半導体メモリ44に記憶されるプログラムは、後述する機能を実現させるために、インターネットサービスプロバイダサーバ3、および、配信サーバ7においては、それぞれクライアントPC1とは異なるものが記憶されるようにしてもよいし、また、処理に応じてCPU21の処理速度や、RAM23、および、記憶部28の容量を変更させることによっても良い。

【0070】

次に、図3を参照して説明したクライアントPC1、および、同様の構成からなる配信サーバ7により実現される機能について説明する。尚、図4においては、ルータ2,4,6,11、インターネットサービスプロバイダサーバ3、および、ネットワーク5の図示は省略するが、いずれも通信経路上に存在するものとする。

【0071】

プレイヤ61は、ストリームデータの再生を制御するプログラムであり、GUI(Graphical User Interface)などにより所定のストリームデータの再生を指示するとき、または、ストリームデータの再生を停止させるときなどに操作され、操作内容に応じた、再生、または、停止といったコマンドを、RTSP(Real Time Streaming Protocol)62に供給し、TCP(Transmission Control Protocol)ポート65により配信サーバ7に対して送信させると共に、上述したコマンドに応じてMulti Flow RTP63より受信される配信されてきたストリームデータを必要に応じて一時的に記憶し、再生する(キャッシュ機能により必要に応じて受信したデータを一時的に記憶したのち、再生に適したビットレートで読み出して再生する)。

【0072】

RTSPは、リアルタイムストリームデータの配信を制御し、プレイヤ61より供給された再生、または、停止といったストリームデータの制御コマンドを、TCPポート65を介し

10

20

30

40

50

て配信サーバ7に供給する。

【0073】

また、RTSP 6 2は、SETUPメソッド（配信サーバ7とクライアントPC1の間でストリームデータ送信のために必要なネゴシエーションを行う手順）中のTransportヘッダフィールド（SETUPメソッドにおいて、データ送信プロトコルのネゴシエーションに利用されるフィールド）によってストリームデータ送信プロトコルのネゴシエーションを実行する。このとき、RTSP 6 2は、送信プロトコルにRTP/UDPを指定するが、その際client_portパラメータおよびserver_portパラメータにはMulti Flow RTP 6 3およびRTCP 6 4で使用するUDPポート（図4においては、UDPポート67乃至69）の範囲を指定する。

【0074】

Multi flow RTP 6 3は、複数のフローによってストリームデータ送信を行うプロトコルである。ここで、フローとは、送信元、送信先のIPアドレス、トランスポートプロトコル、および、ポートの組み合わせによって定義されるデータ送信単位である。また、フローは、一般的なルータがフォワード優先度の制御を行う際の単位である。

【0075】

Multi flow RTP 6 3は、RTSP 6 2により指定されたUDPポート67乃至69を制御して、各UDPポート67乃至69に分割して送信されてきたストリームデータを取得する。また、Multi flow RTP 6 3は、RTCP 6 4よりTCPポート66を介して配信サーバ7より供給される分割されたストリームデータの送信順序や、配置などの、分割されているデータを組み立てて（結合させて）再構築するための制御情報を取得する。さらに、Multi flow RTP 6 3は、分割された状態で受信したストリームデータを、RTCP 6 4より取得した制御情報に基づいて結合させ、元のストリームデータを再構築してプレイヤ61に供給する。

【0076】

配信サーバ7の配信プログラム81は、TCPポート85を介してクライアントPC1より指定されたストリームデータの再生、または、停止といった指令を受け取ると、これに対応する指示をRTSP 8 2に供給する。

【0077】

RTSP 8 2は、基本的には、クライアントPC1のRTSP 6 2と対応するものであるが、配信サーバ7においては、配信プログラム81より指示のあったストリームデータの再生、または、停止の指示に基づいて、再生、または、停止が指示されたストリームデータのアドレスを指定し、Multi Flow RTP 8 3に供給する。また、RTSP 8 2は、送信プロトコルにRTP/UDPを指定するが、その際client_portパラメータおよびserver_portパラメータにはMulti Flow RTP 8 3およびRTCP 8 4で使用するUDPポート（図4においては、UDPポート67乃至69）の範囲を指定する。

【0078】

Multi Flow RTP 8 3は、基本的には、クライアントPC1のMulti Flow RTP 6 3と対応するものであるが、RTSP 8 2によりアドレスが指定されたストリームデータを読み出し、これを指定されたUDPポート87乃至89で送信できるように分割し、各UDPポート87乃至89に供給すると共に、その際、分割されたストリームデータを再構築するための情報と、UDPポート87乃至89のうち通信に利用するポートを識別するポート番号および利用ポート数の情報を含む制御情報を生成して、RTCP 8 4に供給する。

【0079】

より詳細には、1つのRTPセッションで送信されるべきストリームデータがUDPポート87乃至89に対応するように分割される場合、制御情報は、分割された個々のデータに付されるシーケンス番号（Sequence Number）に基づいて、各シーケンス番号のデータをどのような順序で分割したか、すなわち、どのような順序で再構築するかを示した情報と、通信に利用されるUDPポート87乃至89のうちの通信に利用されるUDPポートを識別するためのポート番号と利用ポート数を含む情報となる。

【0080】

また、Multi Flow RTP 8 3は、複数のUDPポート87乃至89を使ってデータを実際に

10

20

30

40

50

送信する際、各UDPポートの動作状態（ソケットへの書き込み状態）を監視し、動作状態に応じて、分割されたデータを割振って、それぞれに送信させる。割振り方法には、ラウンドロビン方式や即時方式などがある。ラウンドロビン方式は、分割されたデータを複数のUDPポートに対して所定の順序で割振って送信させる方法である。また、即時方式は、複数のUDPポートに対して、順序とは無関係に書き込み可能であれば、同時に送信すべきデータを割振って送信させる方法である。割振り方法については、ラウンドロビン方式または即時方式のいずれであってもよいが、例えば、モードとして切替えて使用できるようにしてもよいし、それ以外の割振り方法であってもよい。

【0081】

RTCP 8 4 は、基本的には、クライアントPC 1 のRTCP 6 4 と同様のものであるが、Multi Flow RTP 6 3 より供給された制御情報に基づいて、UDPポート 8 7 乃至 8 9 を制御して、クライアントPC 1 にストリームデータを供給させると共に、供給された制御情報を、TCPポート 8 6 を制御してクライアントPC 1 に送信する。10

【0082】

図4に示したプレイヤ6 1、RTSP 6 2、Multi Flow RTP 6 3、RTCP 6 4、TCP 6 5，6 6、および、UDP 6 7 乃至 6 9、並びに、配信プログラム 8 1、RTSP 8 2、Multi Flow RT P 8 3、RTCP 8 4、TCP 8 5，8 6、および、UDP 8 7 乃至 8 9 は、いずれもプロトコルであるが、図5で示されるように、それぞれは、OSI参照モデルにより、以下のように分類される。20

【0083】

すなわち、プレイヤ6 1、および、配信プログラム 8 1 は、アプリケーション層であり、クライアントPC 1 や配信サーバ7 を操作するユーザより見えるアプリケーションを規定する。このとき、ストリームデータが、例えば、動画データである場合、その動画データに大生する動画フォーマットが、プレゼンテーション層となり、通信におけるフォーマットやコードを規定する。20

【0084】

RTSP 6 2，8 2、Multi Flow RTP 6 3，8 3、RTCP 6 4，8 4 は、セッション層であり、通信の手順を規定するプロトコルである。TCPポート 6 5，6 6，8 5，8 6、並びに、UDPポート 6 7 乃至 6 9、8 7 乃至 8 9 は、トранスポート層であり、論理的な通信路を規定する。30

【0085】

以上の元で、TCP/IPにおけるプロトコルであるIP (Internet Protocol) は、ネットワークを介した通信経路を規定するネットワーク層となり、各機器の物理的なアドレスであるMACアドレス (Media Access Control Address) が、物理的に隣り合った機器間の論理信号手順（例えば、パケット化の手順など）を規定するデータ層となり、さらに、LANなどの物理的な電気的接続の規定である物理層となる。

【0086】

図6は、図4の機能を、図5を参照して説明したOSI参照モデルに基づいて、各層毎に配置したものである。図6において、各ブロックの重なり合っている部分、または、接触している部分は、図中の上部のブロックが下部のブロックを制御していることを示し、各矢印は、データの伝送方向を示している。40

【0087】

すなわち、RTSP 6 2，8 2 は、同じセッション層のMulti Flow RTP 6 3，8 3 をそれぞれ制御すると共に、トランスポート層のTCPポート 6 5，8 5 をそれぞれ制御する。また、Multi Flow RTP 6 3，8 3 が、同じセッション層におけるRTCP 6 4，8 4 をそれぞれ制御する。また、セッション層のRTCP 6 4，8 4 は、トランスポート層のUDP 6 7 乃至 6 9、8 7 乃至 8 9、および、トランスポート層のTCP 6 6，8 6 をそれぞれ制御する。このとき、配信サーバ7 のUDPポートと、クライアントPC 1 のUDPポートは、それぞれ1対1でペアを組んでデータの授受を行う。すなわち、図6においては、UDP 6 7 と 8 7、UDP 6 8 と 8 8、および、UDP 6 9 と 8 9 が、それぞれペアを形成し、データがそれぞれのペア毎50

に授受される。

【0088】

以上の機能により、1つのセッションにより送信されるストリームデータの送信が複数のポートで行われる。ここで、Multi-Flow RTP 63, 83によるストリーミング送信の最大単位を、「RTPセッション」と定義し、あるRTPセッションに属する一対の送信先・送信元UDPポートのペア（例えば、図6においては、UDPポート67と87、UDPポート68と88、または、UDPポート69と89）により授受されるデータの単位を、「RTPデータフロー」と定義するものとする。

【0089】

この場合、Multi Flow RTP 63, 83における各RTPデータフローは、RFC1889のRTPデータ送信規定に従うが、RTCP 64, 84は、各UDPポートのペアで授受されるRTPデータフロー単位で制御を行わず、UDPポート67乃至69、および、UDPポート87乃至89を1つとして捉え、RTPセッション全体を制御する。10

【0090】

このとき、Multi Flow RTP 63, 83は、RTPデータパケットのシーケンス番号（sequence number）をRTPデータフロー単位ではなくRTPセッション単位で付与し、異なるRTPデータフローによって送信されたデータパケットであっても、シーケンス番号によって一意に管理する。

【0091】

また、通常のRTPでは、SSRC（Synchronization Source：RTPにおいてストリームデータパケットのソース識別を行うための32bitのフィールド）によってストリームデータソースが識別されるが、Multi-Flow RTP 83は、SSRCをRTPセッションと1対1に対応させる。これにより、Multi-Flow RTP 63は、複数のRTPデータフローを1つのRTPセッションとしてまとめることができ、また複数のRTPセッションに属するデータを同一UDPポートで受信した場合であっても適正に仕分けることができる。20

【0092】

尚、RTPデータフローの各UDPポートは、原則として連続したポート範囲で割り当てるものとするが、RFC1889の規定に従い連続したポート範囲のうち偶数ポートのみを割り当てるようにしてもよい。

【0093】

次に、図7のフローチャートを参照して、図4のクライアントPC1と配信サーバ7によるストリームデータの配信処理について説明する。30

【0094】

ステップS1において、プレイヤ61は、入力部26が操作されて、所定のストリームデータの配信を要求する操作がなされたか否かを判定し、所定のストリームデータの配信を要求する操作がなされるまで、その処理を繰り返す。そして、ユーザが、入力部26を操作して、データの配信を要求したと判定した場合、その処理は、ステップS2に進む。

【0095】

ステップS2において、プレイヤ61は、ストリームデータの配信を要求する操作指示をRTSP 62に供給して、TCPポート65を介して配信サーバ7に送信させる。このとき、当然のことながら、図1で示したように、ルータ2、インターネットサービスプロバイダサーバ3、ルータ4、ネットワーク5、および、ルータ6を介した経路を経て、ストリームデータの配信を要求する情報が配信サーバ7に送信される。尚、以後においては、同様の経路を経てクライアントPC1と配信サーバ7のデータが授受されるものとする事を前提とするので、ルータ2、インターネットサービスプロバイダサーバ3、ルータ4、ネットワーク5、および、ルータ6におけるデータの授受に関する説明は適宜省略する。40

【0096】

ステップS11において、配信プログラム81は、RTSP 82よりTCPポート85を介して、ストリームデータの配信がクライアントPC1より要求されたか否かを判定し、ストリームデータの配信がクライアントPC1より要求されたと判定されるまで、その処理を繰り50

返す。ステップS11において、例えば、ステップS2の処理により、ストリームデータの配信がクライアントPC1より要求されたと判定された場合、配信プログラム81は、RTSP82を制御して、送信すべきストリームデータが格納されているアドレスを指定する情報をMulti Flow RTP83に供給させる。

【0097】

ステップS13において、Multi Flow RTP83は、RTSP82より供給されたアドレスの情報に基づいて、送信すべきストリームデータを読み出して、UDPポート87乃至89に対応する数に分割し、UDPポート87乃至89に供給すると共に、このときの分割したストリームデータを再構築するために必要な制御情報を生成してRTCP84に供給する。

【0098】

ステップS14において、Multi Flow RTP83は、分割したストリームデータを再構築するために必要な情報と、通信に使用するUDPポートのポート番号と利用ポート数の情報とを含む制御情報を生成する。

【0099】

ステップS15において、RTCP84は、TCPポート86を制御して、制御情報をクライアントPC1に送信させる。

【0100】

ステップS16において、複数UDPポート送信処理が実行され、RTCP84により、UDPポート87乃至89が制御されて、Multi Flow RTP83により分割された状態で個々に供給されたストリームデータがクライアントPC1に送信される。尚、複数UDPポート送信処理については、図18乃至図22を参照して後述する。

【0101】

ステップS3において、Multi Flow RTP63は、RTCP64を制御して、TCPポート66に問い合わせて、分割されて各UDPポート67乃至69に受信されるストリームデータを再構築するために必要な情報（例えば、データを再構築するための順序の情報）と、通信に利用されるUDPポート67乃至69を特定するポート番号、および、利用ポート数の情報とを含む制御情報が送信されているかを判定し、送信されてくるまで、その処理を繰り返す。そして、例えば、ステップS3において、ステップS15の処理によりデータが送信されてきていると判定された場合、その処理は、ステップS4に進む。

【0102】

ステップS4において、RTCP64は、制御情報をTCPポート66を制御して受信させて、さらに、Multi Flow RTP63に供給する。

【0103】

ステップS5において、RTCP64は、制御情報の通信に使用されるポート番号と利用するポート数の情報に基づいて、UDPポート67乃至69に問い合わせて、それぞれ対応するUDPポート87乃至89よりデータが送信されてきているか否かを確認し、データが送信されてくるまで、その処理を繰り返す。ステップS5において、例えば、ステップS16の処理によりデータが送信されてきていると判定された場合、その処理は、ステップS6に進む。

【0104】

ステップS6において、複数UDPポート受信処理が実行され、RTCP64により、制御情報に含まれている通信に利用されるUDPポート67乃至69を特定するポート番号、および、利用ポート数の情報に基づいて、UDPポート67乃至69が制御されて、送信されてきた分割されているストリームデータが受信され、Multi Flow RTP63に供給される。尚、複数UDPポート受信処理については、図18乃至図22を参照して後述する。

【0105】

ステップS7において、Multi Flow RTP63は、各UDPポートより受信された分割されているストリームデータを、制御情報に含まれているストリームデータを再構築するために必要な情報に基づいて組み立てることにより、再構築し、プレイヤ61に供給する。

【0106】

10

20

30

40

50

ステップS8において、プレイヤ61は、Multi Flow RTP63より供給された組み立てられて、再構築されている元のストリームデータを所定時間だけバッファリングした後、再生する。

【0107】

以上の処理により、ストリームデータが分割された状態で複数のUDPポートによって送信されることになるので、例えば、図8で示されるように、複数の通信経路により分割されてストリームデータが配信されるので、図8で示されるように、これまで輻輳が発生する以前のタイミングで全てのデータを送信することが可能となる。

【0108】

尚、図8においては、例えば、図6のUDPポートのペアであるUDPポート67と87、UDPポート68と88、または、UDPポート69と89によるRTPデータフローをそれぞれ、実線の太線、点線の太線、および、1点鎖線の太線で示している。すなわち、例えば、UDPポート67と87のRTPデータフローは、ルータ6, 11f, 11e, 11d, 4、インターネットサービスプロバイダサーバ3、ルータ2を介したものであり、UDPポート68と88のRTPデータフローは、ルータ6, 11f, 11b, 11c, 11d, 4、インターネットサービスプロバイダサーバ3、ルータ2を介したものであり、さらに、UDPポート69と89のRTPデータフローは、ルータ6, 11f, 11c, 11d, 4、インターネットサービスプロバイダサーバ3、ルータ2を介したものであり、それぞれが異なる通信路を使うことにより、帯域を広く使うことができ、高速でデータを送信することが可能となるので、図9で示されるように、これまで、輻輳が発生していた時刻t1付近までの間に、送信を完了させることができ、遅延などを抑制して高速の通信を実現させることができる。尚、図9においては、点線が、これまでの通信であり、実線が、図4の配信システムによるデータの転送状況を示したものであり、これまでの通信においては、図9における時刻t1を越えた付近において、輻輳が発生している。

【0109】

以上においては、ストリームデータのうち1のRTPセッションにより配信するデータを分割し、複数のRTPデータフローにより送信する例について説明してきたが、以上の通信においては、UDPポートによる通信であるため、通信経路上のルータのいずれかがフロー制御などによりパケットを破棄した場合、送信したデータが着信しているか否かの確認がなされず、Multi Flow RTP101が、ストリームデータを再構築することができない恐れがある。そこで、通信経路上でパケットが破棄された場合、欠落したパケットを再送要求するようにさせてても良い。

【0110】

図10は、欠落したパケットを再送要求できるようにしたクライアントPC1と配信サーバ7により実現される機能を説明する図である。尚、図4で示されるクライアントPC1と配信サーバ7により実現される機能と同様の機能については、同一の符号を付してあり、その説明は適宜省略する。

【0111】

図10のクライアントPC1と配信サーバ7において、図4のクライアントPC1と配信サーバ7と異なる機能は、Multi Flow RTP63, 83に代えて、再送処理機能付Multi Flow RTP101, 111を設けた点である。再送処理機能付Multi Flow RTP101は、基本的にMulti Flow RTP63と同様の機能を備えているが、さらに、制御情報に含まれている、ストリームデータを再構築するために必要とされる情報に基づいて、UDPポート67乃至69により供給された、分割されているストリームデータのうち、データの欠落の有無を確認し、欠落を検出した場合、RTCP64を制御してTCPポート66より、欠落したデータの再送を要求する。また、再送処理機能付Multi Flow RTP111は、RTCP84を制御して、TCPポート86より再送処理機能付Multi Flow RTP101からの再送要求を受信すると、再送要求のあったデータ(パケット)を複数のUDPポート87乃至89を制御して送信する。

【0112】

10

20

30

40

50

次に、図11のフローチャートを参照して、図10のクライアントPC1と配信サーバ7による配信システムによる配信処理について説明する。尚、図11のフローチャートのステップS31乃至S36,S38,S39の処理、および、ステップS51乃至S56の処理は、図7のフローチャートのステップS1乃至S8およびS11乃至S16の処理と同様であるので、その説明は省略する。

【0113】

ステップS37において、再送処理機能付Multi Flow RTP101は、RTCP64より供給された制御情報に基づいて、UDPポート67乃至69より供給されたデータに欠落がなく全て受信されているか否かを判定する。例えば、図12で示されるように、UDPポート68と88のペアで送信されてくるべきデータがルータ11cにより破棄されてしまったような場合、データが欠落し、全てのデータが受信されていないと判定され、その処理は、ステップS40に進む。10

【0114】

ステップS40において、再送処理機能付Multi Flow RTP101は、遅延許容時間内であるか否かを判定し、遅延許容時間内であると判定した場合、その処理は、ステップS37に戻る。すなわち、全てのデータが受信されていないと判定された状態でも、遅延許容時間内では、ステップS37,S40の処理が繰り返される。そして、遅延許容時間内ではないと判定された場合、ステップS41において、再送処理機能付Multi Flow RTP101は、RTCP64を制御して、TCPポート66より欠落しているデータの再送要求を配信サーバ7に送信させ、その処理は、ステップS33に戻り、それ以降の処理が繰り替えされる。より詳細には、欠落しているRTPデータパケットのシーケンス番号が指定されて、再送が要求される。20

【0115】

ステップS57において、再送処理機能付Multi Flow RTP111は、RTCP84を制御して、TCPポート86に再送要求が送信されてきているか否かを判定し、例えば、ステップS41の処理により再送要求があったと判定された場合、その処理は、ステップS58に進む。

【0116】

ステップS58において、再送処理機能付Multi Flow RTP111は、再送要求のあったデータのアドレス（再送の要求があったデータのシーケンス番号に対応するデータのアドレス）を指定し、その処理は、ステップS53に戻り、それ以降の処理が繰り返される。30

【0117】

ステップS37において、全てのデータが受信されていると判定された場合、その処理は、ステップS38に進む。また、ステップS57において、再送要求がなかったと判定された場合、その処理は、終了する。

【0118】

すなわち、図13で示されるように、RTPデータフロー121が、複数のUDPポートのペア（例えば、図6のUDPポートのペアであるUDPポート67と87、UDPポート68と88、または、UDPポート69と89）を介して、配信サーバ7から送信されると図11のフローチャートのステップS38の処理により、プレイヤ61が、データ131を再生する（所定時間だけバッファリングしてから再生する）。同様にして、クライアントPC1においては、RTPデータフロー122が、同様に複数のUDPポートペアにより送信されることにより、データ132が再生される。40

【0119】

ところが、図13で示されるように、RTPデータフロー123が、例えば、図12で示される点線の通信経路上のルータ11cにより、破棄されてしまったような場合、その分のデータが欠落することになる。従って、この場合、対応するデータはプレイヤ61に供給されないことになる。図13においては、その後に送信されてくるRTPデータフロー124が、受信され、対応するデータ135がバッファリングされる。この間、データ123が欠落してから、遅延許容時間が経過すると、上述したステップS41の処理により、50

RTCP 6 4 がTCPポート 6 6 を制御して再送を要求する。

【 0 1 2 0 】

すると、この再送要求に基づいて、RTPデータフロー 1 2 6 が配信サーバ 7 より送信されてきて、対応する欠落したデータ 1 3 4 がバッファリングされた後、ストリームデータが再生される。図 1 3 においては、その後、RTPデータフロー 1 2 7 , 1 2 8 が、配信サーバ 7 より引き続き送信されてきて、対応するデータ 1 3 6 , 1 3 7 がバッファリングされて、再生される様子が示されている。尚、この遅延許容時間は、バッファリング時間から予測される再送レスポンス時間（再送要求に基づいてデータ送信されてくるまでの予測時間）を減じた時間よりも短くする必要がある。

【 0 1 2 1 】

以上の処理により、複数のUDPポートにより分割された状態でストリームデータ送信されることになるので、これまで輻輳が発生する以前のタイミングで全てのデータを送信することが可能となると共に、通信経路上でデータの欠落が発生したような場合にでも、再送要求によってデータの欠落を抑制し、安定したストリームデータの再生を実現させることができ可能となる。

【 0 1 2 2 】

尚、以上においては、1 の RTPセッションについて、複数の RTPデータフロー（複数の UDPポート）を使用して送受信する例について説明してきたが、例えば、図 1 4 で示されるように、RTSP 6 2 , 8 2 - よりも上位の層に RTPセッションそのものを分割するセッション分割プログラム 1 4 1 、および、分割されたセッションを結合するセッション結合プログラム 1 4 2 をそれぞれ設けるようにしても良い。

【 0 1 2 3 】

さらに、以上においては、1 の RTPセッションからなるデータを複数の RTPデータフロー（複数の UDPポート）に分割して同時に送受信する例について説明してきたが、例えば、図 1 5 で示されるように、UDPポート 6 7 乃至 6 9 , 8 7 乃至 8 9 に代えて、TCPポート 1 5 1 乃至 1 5 3 , 1 6 1 乃至 1 6 3 を設けるようにしても良い。この場合、UDPポートの場合よりも通信速度の低下が見られる可能性があるが、TCPはデータの着信を確認する機能を備えているので、より安定したストリームデータの配信を実現することができる。

【 0 1 2 4 】

尚、図 1 4 , 図 1 5 において、Multi Flow RTP 6 3 , 8 3 は、図 1 0 で示した Multi Flow RTP 1 0 1 , 1 1 1 に置き換えるようにして、再送処理機能を備えるようにさせてても良い。また、UDPポート 6 7 乃至 6 9 、および、8 7 乃至 8 9 は、いずれも 3 個で構成される例について説明してきたが、これに限るものではなく、それ以外の数の UDPポート、または、TCPポートにより構成されるようにしても良い。

【 0 1 2 5 】

以上においては、1 の RTPセッションからなるデータを分割して送信する例について説明してきたが、1 の RTPセッションからなるデータを、同時に複数の UDPポートから送信するようにしてもよい。

【 0 1 2 6 】

図 1 6 は、1 の RTPセッションからなるデータを、同時に複数の UDPポートから送信するようにしたクライアント PC 1 と配信サーバ 7 により実現される機能を説明する図である。尚、図 4 で示されるクライアント PC 1 と配信サーバ 7 により実現される機能と同様の機能については、同一の符号を付してあり、その説明は適宜省略する。

【 0 1 2 7 】

図 1 6 のクライアント PC 1 と配信サーバ 7 において、図 4 のクライアント PC 1 と配信サーバ 7 と異なる機能は、Multi Flow RTP 6 3 , 8 3 、および、RTCP 6 4 , 8 4 に代えて、並列 Multi Flow RTP 1 7 1 , 1 8 1 、および、RTCP 1 7 2 , 1 8 2 を設けた点である。並列 Multi Flow RTP 1 7 1 は、基本的に Multi Flow RTP 6 3 と同様の機能を備えているが、送信するデータを分割せずに、UDPポート 8 7 乃至 8 9 に供給すると共に、送信しようとするデータを識別する情報である制御信号を RTCP 1 7 2 に供給する。RTCP 1 7 2 は、基本

10

20

30

40

50

的に、RTCP 8 4 と同様のものであるが、ストリームデータが分割されないので、分割されたデータを再構築するための制御情報ではなく、送信しようとするデータを識別する情報である制御信号を、TCPポート 8 6 を制御して、クライアントPC 1 に送信させると共に、UDPポート 8 7 乃至 8 9 を制御して、それぞれからデータをクライアントPC 1 に送信させる。

【 0 1 2 8 】

また、並列Multi Flow RTP 1 8 1 は、RTCP 1 8 2 を制御して、TCPポート 6 6 により配信サーバ 7 からの送信されてくるデータを識別する情報である制御情報を受信し、UDPポート 6 7 乃至 6 9 のいずれかで、その制御情報に基づいて識別されるデータ（パケット）が受信されると、その他のUDPポート 6 7 乃至 6 9 で受信されるデータを破棄すると共に、最初受信されたデータをプレイヤ 6 1 に供給して再生させる。

10

【 0 1 2 9 】

次に、図 1 7 のフローチャートを参照して、図 1 6 のクライアントPC 1 と配信サーバ 7 からなる配信システムによる配信処理について説明する。尚、図 1 7 のフローチャートのステップ S 7 1 , S 7 2 の処理、および、ステップ S 9 1 , S 9 2 の処理は、図 7 のフローチャートのステップ S 1 , S 2 および S 9 1 , S 9 2 の処理と同様であるので、その説明は省略する。

【 0 1 3 0 】

ステップ S 9 3 において、並列Multi Flow RTP 1 7 1 は、RTSP 8 2 より供給されたアドレスの情報に基づいて、送信すべきストリームデータを読み出し、そのストリームデータを識別する情報と、UDPポート 8 7 乃至 8 9 のうちの通信に使用するポート番号と利用ポート数の情報を含む制御情報を生成する。

20

【 0 1 3 1 】

ステップ S 9 4 において、並列Multi Flow RTP 1 7 1 は、RTCP 8 4 を制御して、TCPポート 8 6 より、生成した送信しようとするストリームデータを識別する情報と、UDPポート 8 7 乃至 8 9 のうちの通信に使用するポート番号と利用ポート数の情報を含む制御情報をクライアントPC 1 に送信させる。

【 0 1 3 2 】

ステップ S 9 5 において、複数UDPポート送信処理が実行され、並列Multi Flow RTP 1 7 1 により、送信しようとするストリームデータがUDPポート 8 7 乃至 8 9 に、それぞれ供給され、クライアントPC 1 に送信される。すなわち、この処理により、送信すべき同一のストリームデータが同時に複数のUDPポート 8 7 乃至 8 9 よりクライアントPC 1 に送信される。尚、ステップ S 9 5 の処理においては、ステップ S 1 6 , S 5 6 の処理と異なり、送信されるデータが分割されていない状態であるが、複数のUDPポート 8 7 乃至 8 9 を使用した複数UDPポート送信処理そのものは同様である。

30

【 0 1 3 3 】

ステップ S 7 3 において、RTCP 1 8 2 は、TCPポート 6 6 を制御して、配信サーバ 7 より制御情報が送信されてきたか否かを判定し、送信されてくるまでその処理を繰り返す。例えば、ステップ S 9 3 の処理により制御情報が送信されてきた場合、その処理は、ステップ S 7 4 に進む。

40

【 0 1 3 4 】

ステップ S 7 4 において、複数UDPポート受信処理が実行され、RTCP 1 8 2 により、TCPポート 6 6 が制御されて制御情報が受信され、さらに、受信された制御情報が並列Multi Flow RTP 1 8 1 に供給される。尚、ステップ S 7 4 の処理においては、ステップ S 6 , S 3 6 の処理と異なり、受信されるデータが分割されていない状態であるが、複数のUDPポート 6 7 乃至 6 9 を使用した複数UDPポート受信処理そのものは同様である。

【 0 1 3 5 】

ステップ S 7 5 において、並列Multi Flow RTP 1 7 1 は、制御情報に基づいて、UDPポート 6 7 乃至 6 9 のいずれかにおいて、送信されてきたストリームデータが受信されたか否かを判定し、受信されるまで、その処理を繰り返す。例えば、ステップ S 9 4 の処理に

50

より、ステップS76において、複数UDPポート受信処理が実行され、UDPポート87乃至89のいずれかから送信されてきたストリームデータが受信された場合、その処理は、ステップS77に進む。

【0136】

ステップS77において、並列Multi Flow RTP171は、UDPポート67乃至69のいずれかにより受信されたストリームデータと、受信されている制御情報を比較し、既に受信済みのストリームデータであるか否かを判定する。より詳細には、並列Multi Flow RTP171は、受信されたストリームデータを識別する情報と、制御情報に含まれている送信されてくるべきストリームデータを識別する情報とを比較し、受信したストリームデータが、送信されてくるべきストリームデータであるか否かを識別し、送信されてくるべきストリームデータであった場合で、かつ、既に受信されているものであるか否かを判定する。例えば、ステップS76において、受信されたストリームデータが、まだ、受信されていない最初のストリームデータであると判定した場合、その処理は、ステップS77に進む。

10

【0137】

ステップS78において、並列Multi Flow RTP171は、受信したストリームデータをプレイヤ61に供給し、再生させる。

【0138】

ステップS77において、既に、受信済みのストリームデータであると判定された場合、すなわち、ステップS94の処理により、ストリームデータは、複数のUDPポートにより同時に並列に送信されてくるので、そのうちのいずれか複数の経路上でパケットが破棄されることなく、複数のストリームデータが、複数のUDPポートで受信された場合、ステップS79において、並列Multi Flow RTP171は、2番目以降に受信されたストリームデータのパケットを破棄する。

20

【0139】

このため、従来UDPパケットによるデータ転送では、パケットが通信路上のいずれかのルータで破棄された場合、例えば、ストリームデータが動画であるとき、データが到達しないまま再生が進む事で、画質の劣化が生じてしまうので、画質を下げることによって対応することしかできなかったが、以上の処理により、同一のストリームデータを複数ポートで並列的に転送する事が可能となり、結果として、並列に送信されたストリームデータのうちの1つでも届けば、ストリームデータが届かないといった事体を回避することができるので、通信の信頼性を高めることが可能となる。また、正しいパケットが1つでも到達した時点で、他のパケットの破棄処理を行う事で、確認応答がないためにTCP方式の通信に比べて高速なUDP方式の通信の特徴を活かしたままで、UDP方式による通信における弱点であった信頼性の高いデータ転送を実現することが可能となる。

30

【0140】

また、単一のTCPコネクションやUDPパケットでは、単一のコネクション内や単一のパケットにおける送信できるデータ容量の限界があり、さらに、TCP方式の場合、フロー制御等で、単一のコネクションでは、利用可能帯域一杯まで利用する事が難しいといった問題があったが、以上のような処理により、複数のポートを利用した転送を、TCP方式、または、UDP方式で行う事で、帯域を限界近くまで利用することが可能となり、転送容量を高めることが可能となる。

40

【0141】

すなわち、送信しようとするストリームデータを分割し、連続データとして複数のポートで同時送受信する事で、より高速でデータを転送させることができる。また、同一データを複数のポートを用いて送信する場合、パケットが破棄されてしまうといった状況を意識した同一データの複数ポートでの受信を行うことにより、通信の信頼性を向上させつつ、従来の単一ポートでの送受信と変わらない速度での転送が可能になる。

【0142】

さらに、以上のように複数ポートを利用して転送することで、クライアントPC側にキヤ

50

ツシユ機能を設ける事によって、必要なデータを高速で転送することができ、また、ネットワークの利用率を複数転送で高める事で、一般的な通信が单一ポート利用したときに生じる、フロー制御による通信速度が低下する現象や、長時間の利用や通信頻度の低下によって引き起こされる、ルータの優先制御によって通信速度が下がる等の現象を回避する事が可能になる。

【0143】

次に、図18のフローチャートを参照して、ラウンドロビン方式を用いた複数UDPポート送信処理、および、複数UDPポート受信処理について説明する。図18のフローチャートにおいては、複数UDPポート送信処理、および、複数UDPポート受信処理を図7のフローチャートにおけるステップS6, S16の詳細な処理として説明するが、図11のフローチャートにおけるステップS36, S56、および、図17のフローチャートにおけるステップS76, S95の処理も同様である。10

【0144】

ステップS111において、Multi Flow RTP83は、UDPポート87乃至89を識別するためのカウンタpを1に初期化する。尚、以下の説明においては、カウンタp=1は、UDPポート87に、カウンタp=2は、UDPポート88に、カウンタp=3は、UDPポート89にそれぞれ対応するものとし、UDPポートが3個であるため、カウンタpの上限Pは3であるものとするが、もちろん、それ以外の数であってもよい。

【0145】

ステップS112において、Multi Flow RTP83は、カウンタpに対応するUDPポートの検査処理を実行する。すなわち、カウンタpに対応するUDPポートとは、例えば、最初の処理においては、カウンタp=1に対応するUDPポートの検査処理が実行される。20

【0146】

ここで、図19のフローチャートを参照して、UDPポート検査処理について説明する。

【0147】

ステップS151において、Multi Flow RTP83は、検査対象となるUDPポートを指定する。より詳細には、例えば、UNIX(登録商標)系のSystemV(商標)をOSとして使用した場合、Multi Flow RTP83は、FD-SET(OS(=SystemV)の標準マクロ)を実行することにより、検査対象となるカウンタpに対応するUDPポートを特定するディスクリプタを指定する。30

【0148】

ステップS152において、Multi Flow RTP83は、検査対象となるUDPポートが通信可能であるか否かを検査する。すなわち、より詳細には、Multi Flow RTP83は、例えば、select()関数(OS(=SystemVなど)の標準関数)を実行することによりFD-SETで指定されたディスクリプタの動作状態を検査する(より具体的には、検査対象となるUDPポートのソケットが書き込み可能であるか否かが検査される)。

【0149】

ステップS153において、Multi Flow RTP83は、検査対象となるUDPポートが通信可能であるか否かの検査結果を確認する。すなわち、より詳細には、Multi Flow RTP83は、例えば、select()関数の結果を、FD-ISSET(OS(=SystemV)の標準関数)を実行することにより確認する。40

【0150】

以上のように、ステップS151乃至S153の処理により、カウンタpで指定される1個の検査対象となるUDPポートが送信可能であるか否かが検査される。

【0151】

ここで、図18のフローチャートに戻る。

【0152】

ステップS113において、Multi Flow RTP83は、検査対象となったカウンタpに対応するUDPポートが送信可能であるか否かを判定する。例えば、送信可能ではない、すなわち、例えば、別のデータを送信中などの状態で、送信できない状態である場合、その処50

理は、ステップS112に戻る。すなわち、送信可能であると判定されるまで、ステップS112, S113の処理が繰り返される。

【0153】

そして、ステップS113において、送信可能であると判定された場合、その処理は、ステップS114に進む。

【0154】

ステップS114において、RTCP84は、カウンタpに対応するUDPポートを制御して、Multi Flow RTP83により分割されたストリームデータをクライアントPC1に送信させる。

【0155】

ステップS131において、RTCP64は、制御情報に含まれている通信に利用されるUDPポート67乃至69を特定するポート番号、および、利用ポート数の情報に基づいて、UDPポート67乃至69を制御して、送信されてきた分割されているストリームデータを受信させ、Multi Flow RTP63に供給させる。すなわち、UDPポート67乃至69のうち、上述したカウンタpに対応するUDPポート87乃至89のいずれかで分割されているストリームデータが受信される。

【0156】

一方、ステップS115において、Multi Flow RTP83は、送信すべきストリームデータがあるか、すなわち、分割したストリームデータのうち、送信されずに残っているものがあるか否かを判定し、送信すべきストリームデータが存在しない場合、その処理は終了する。

【0157】

ステップS115において、送信すべき、分割されているストリームデータがあると判定された場合、ステップS116において、Multi Flow RTP83は、カウンタpを1インクリメントする。

【0158】

ステップS117において、Multi Flow RTP83は、カウンタpが上限であるPよりも大きいか否かを判定し、例えば、大きくないと判定された場合、その処理は、ステップS112に戻る。

【0159】

また、ステップS117において、カウンタpが上限であるPよりも大きいと判定された場合、その処理は、ステップS111に戻る。

【0160】

すなわち、以上の処理により、各UDPポートについて、UDPポート87, 88, 89の順番に送信可能であるか否かが判定され、UDPポート89の検査が終了すると、再びUDPポート87が送信可能であるかが判定され、その処理が繰り返される。さらに、送信不能である場合、送信可能となるまで待ち、送信可能となった時点で、そのUDPポートからストリームデータを送信した後に、次のUDPポートの検査を実行しているため、分割されたストリームデータは、必ずUDPポート87, 88, 89, 87, 88, 89…の順番に割振られる（ラウンドロビン方式で割振られる）。従って、クライアントPC1は、UDPポート67, 68, 69, 67, 68, 69…の順番に受信されることになる。

【0161】

結果として、複数のUDPポートにより分割されたストリームデータが効率よく送信されることになる。尚、図17においては、複数のUDPポート87, 88, 89から並列に同じストリームデータが送信されることが想定されているが、上述したように同時タイミングではない。

【0162】

次に、図19のフローチャートを参照して、即時方式による複数UDPポート送信処理および複数UDPポート受信処理について説明する。尚、図19のフローチャートにおけるステップS171乃至S177, S191の処理は、図17のフローチャートにおけるステ

10

20

30

40

50

ツップS111乃至S117, S131の処理と同様であるので、その説明は省略する。

【0163】

すなわち、図19のフローチャートの各処理と図17のフローチャートの各処理は、全く同一である。しかしながら、図17においては、ステップS174において、送信不能であると判定された場合、ステップS174の処理がスキップされて、その処理は、ステップS175に進む。

【0164】

従って、送信不能と判定されたUDPポートが送信可能と判定されるまでの待ち状態がなくなる。このため、複数のUDPポート87乃至89の送信順序は、一定とはならないが、UDPポートが送信不能であると判定された場合、即時、別のUDPポートを検査し、送信可能なUDPポートに対して優先的に分割されているストリームデータの送信が割振られる。10

【0165】

結果として、即時方式による複数UDPポート送信処理および複数UDPポート受信処理は、ラウンドロビン方式による複数UDPポート送信処理および複数UDPポート受信処理よりも高速でストリームデータを送信することが可能となる。

【0166】

尚、以上の説明においては、複数のポートの使用例として複数のUDPポートを使用した場合の例について説明してきたが、上述した処理は、プロトコルに依存しない処理であるため、複数のポートは、必ずしもUDPポートである必要が無く、例えば、複数のUDPポートの代わりに複数のTCPポートを使用しても同様の効果が得られる。20

【0167】

図21, 図22は、それぞれラウンドロビン方式と即時方式による複数のTCPポートによる送信処理および複数のTCPポートによる受信処理におけるTCPポート数とスループットの関係を示している。尚、図21, 図22においては、実線が、50Mbpsまで、点線が30Mbpsまで、1点鎖線が10Mbpsまでのそれぞれ使用可能な帯域の通信経路を示している。また、縦軸は、スループット(Mbps)を示し、横軸はTCPポート数を示している。

【0168】

図21で示されるように、ラウンドロビン方式を採用した場合、帯域が最大50Mbpsのとき、ポート数が2個で、帯域が最大30Mbpsのとき、ポート数が3個で、帯域が最大10Mbpsのとき、ポート数が6個で、それぞれ急激にスループットが低下することが示されている。30

【0169】

一方、図22で示されるように、即時方式を採用した場合、帯域が最大50Mbpsのとき、帯域が最大30Mbpsのとき、帯域が最大10Mbpsのときのいずれにおいても、TCPポート数に関わらず帯域幅を十分に使うことができる事が示されている。

【0170】

結果として、図21, 図22からもわかるように、ラウンドロビン方式よりも即時方式の方が、複数のTCPポートを使用して通信した場合、帯域を十分に使用することができ、高速で、データを送受信できることが示されている。尚、当然のことながら、複数のUDPポートを使用した場合についても、図21, 図22を参照して説明した効果と同様な効果を得ることが可能であることは言うまでもない。40

【0171】

以上によれば、セッション層における1セッションの通信データを複数のトランスポート層のポートを利用して送信することにより、高速通信と安定的な通信を実現することが可能となる。

【0172】

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行させることが可能な、例えば汎用のパーソナルコンピュータなどに記録媒体からインストール50

される。

【0173】

プログラムが記録されている記録媒体は、図3に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク21(フレキシブルディスクを含む)、光ディスク22(CD-ROM(Compact Disc-Read Only Memory), DVD(Digital Versatile Disk)を含む)、光磁気ディスク23(MD(Mini-Disc)を含む)、もしくは半導体メモリ24などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記録されているROM22や、記憶部28に含まれるハードディスクなどで構成される。

10

【0174】

尚、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理は、もちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理を含むものである。

【0175】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【図面の簡単な説明】

【0176】

【図1】輻輳を説明する図である。

20

【図2】本発明を適用した配信システムの一実施の形態の構成を示す図である。

【図3】図2のクライアントPCの構成を示す図である。

【図4】図2のクライアントPCと配信サーバ7により実現される機能を示す図である。

【図5】OSI参考モデルを説明する図である。

【図6】図4で示されるクライアントPCと配信サーバ7により実現される機能をOSI参考モデルに対応させた図である。

【図7】図4の機能による配信処理を説明するフローチャートである。

【図8】配信処理を説明する図である。

【図9】データの転送速度を説明する図である。

【図10】図2のクライアントPCと配信サーバ7により実現されるその他の機能を示す図である。

30

【図11】図10の機能による配信処理を説明するフローチャートである。

【図12】図10の機能による配信処理を説明する図である。

【図13】図10の機能による配信処理を説明する図である。

【図14】図2のクライアントPCと配信サーバ7により実現されるさらにその他の機能を示す図である。

【図15】図2のクライアントPCと配信サーバ7により実現されるさらにその他の機能を示す図である。

【図16】図2のクライアントPCと配信サーバ7により実現されるさらにその他の機能を示す図である。

40

【図17】図16の機能による配信処理を説明するフローチャートである。

【図18】ラウンドロビン方式による複数UDPポート送信処理および複数UDPポート受信処理を説明するフローチャートである。

【図19】UDPポート検査処理を説明するフローチャートである。

【図20】即時方式による複数UDPポート送信処理および複数UDPポート受信処理を説明するフローチャートである。

【図21】ラウンドロビン方式による複数UDPポート送信処理および複数UDPポート受信処理におけるスループットとUDPポート数の関係を説明する図である。

【図22】即時方式による複数UDPポート送信処理および複数UDPポート受信処理におけるスループットとUDPポート数の関係を説明する図である。

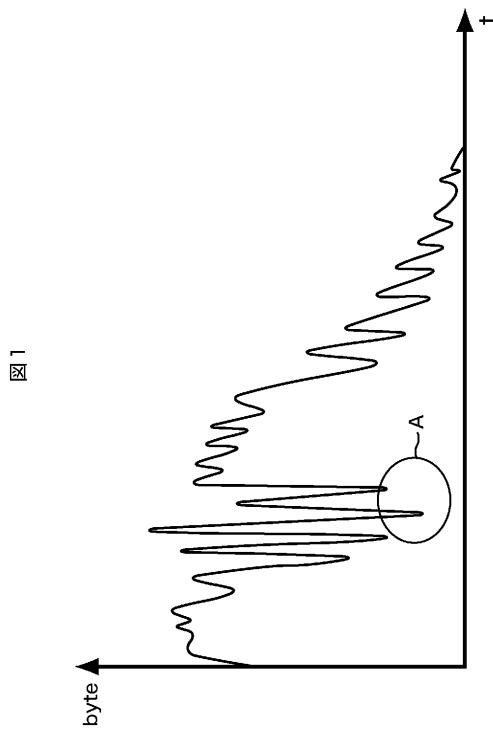
50

【符号の説明】

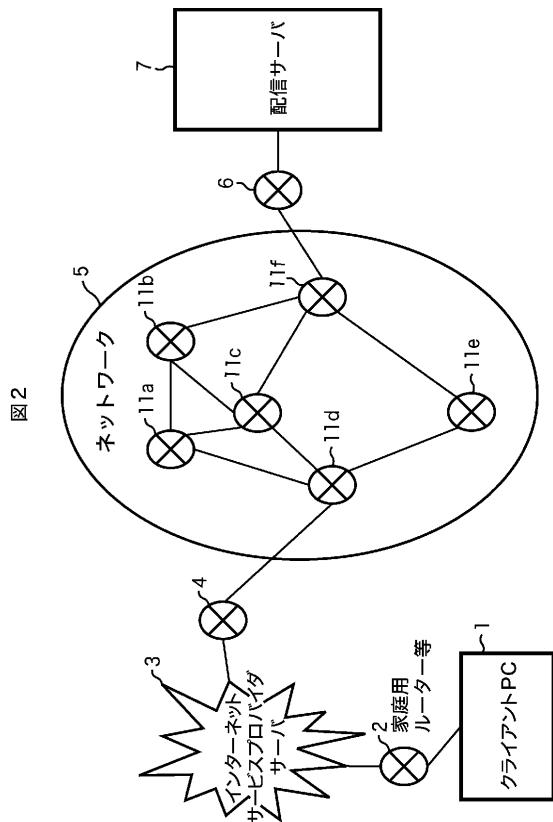
【0177】

1 クライアントPC, 2 ルータ, 3 インターネットサービスプロバイダサーバ,
 4 ルータ, 5 ネットワーク, 6 ルータ, 7 配信サーバ, 11, 11a
 乃至 11f ルータ, 61 プレイヤ, 62 RTSP, 63 Multi Flow RTP,
 64 RTCP, 65, 66 TCP, 67 乃至 69 UDP, 81 プレイヤ, 82 RT
 SP, 83 Multi Flow RTP, 84 RTCP, 85, 86 TCP, 87 乃至 89 U
 DP, 101, 111 再送処理機能付Multi Flow RTP, 171 再送処理機能付Mu
 lti Flow RTP, 172 RTSP, 181 並列Multi Flow RTP, 182 RTSP

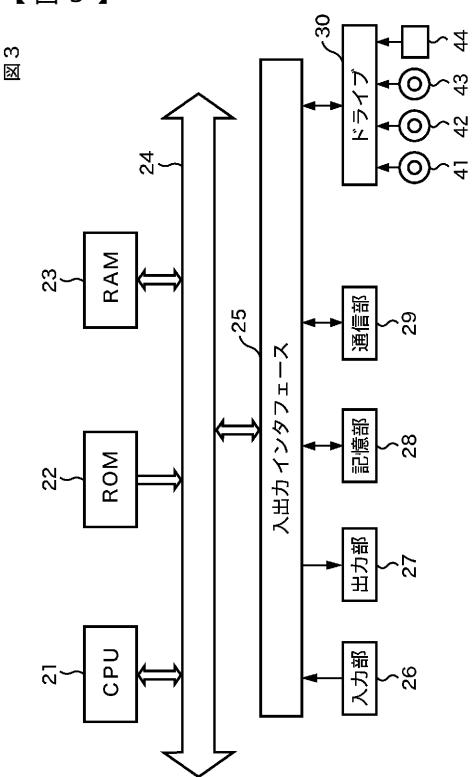
【図1】



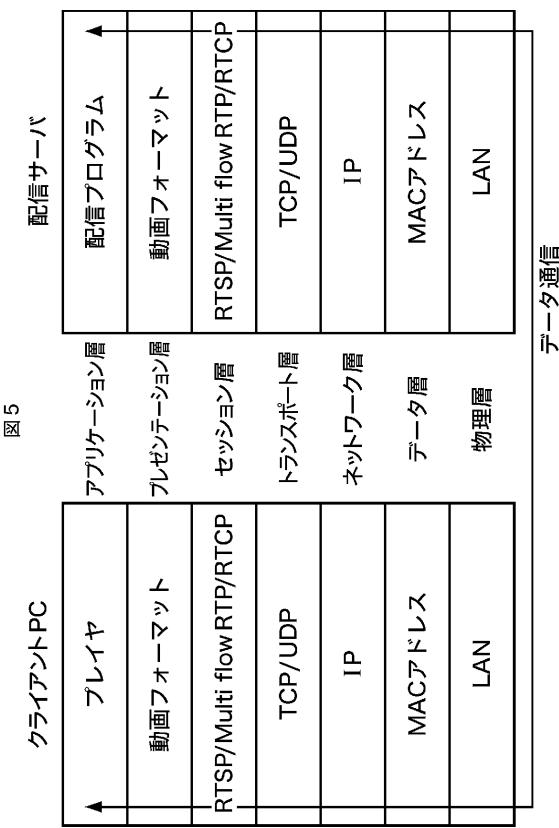
【図2】



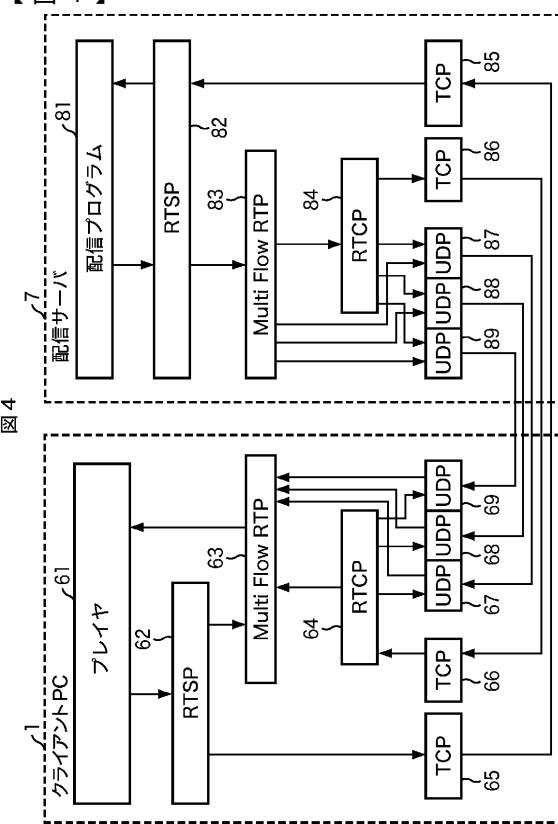
【図3】



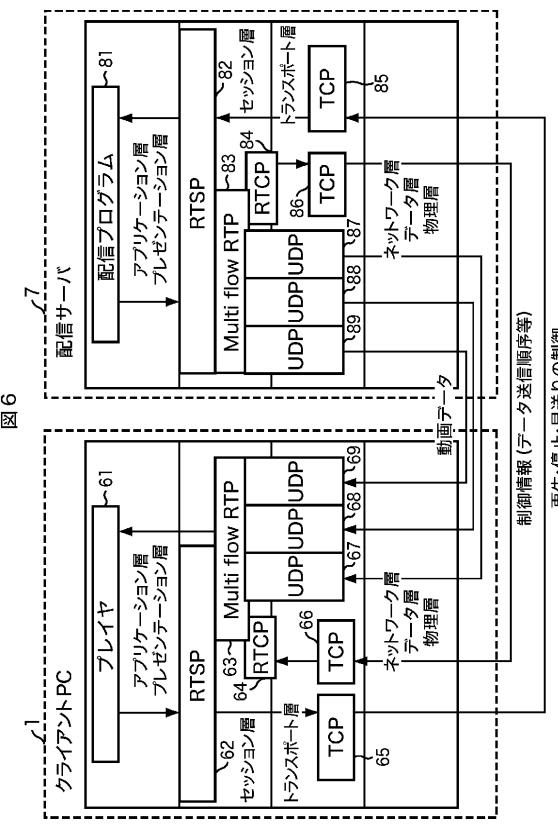
【図5】



【図4】

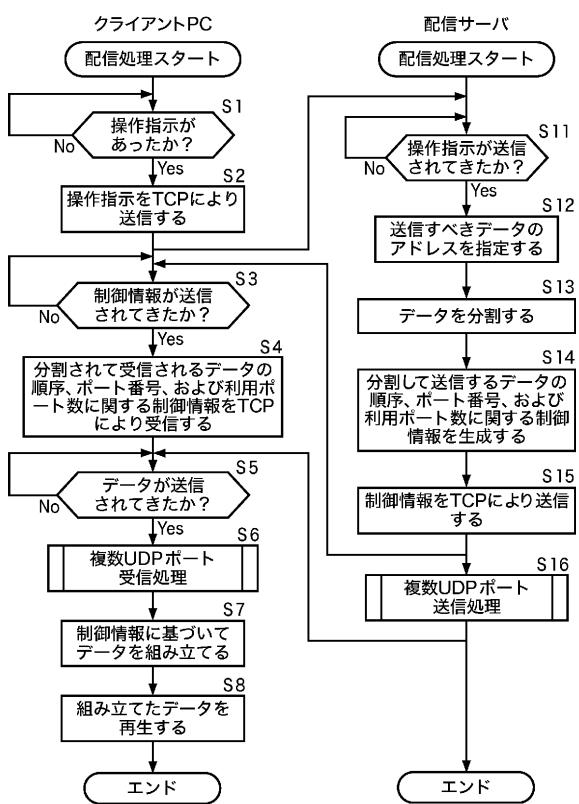


【図6】



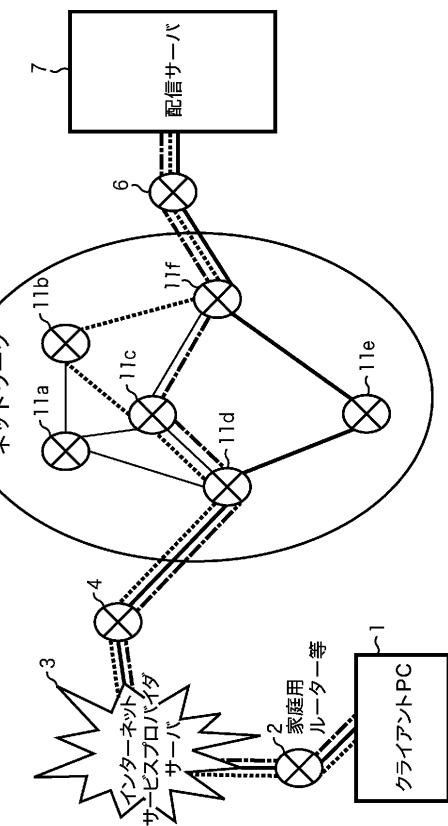
【図7】

図7



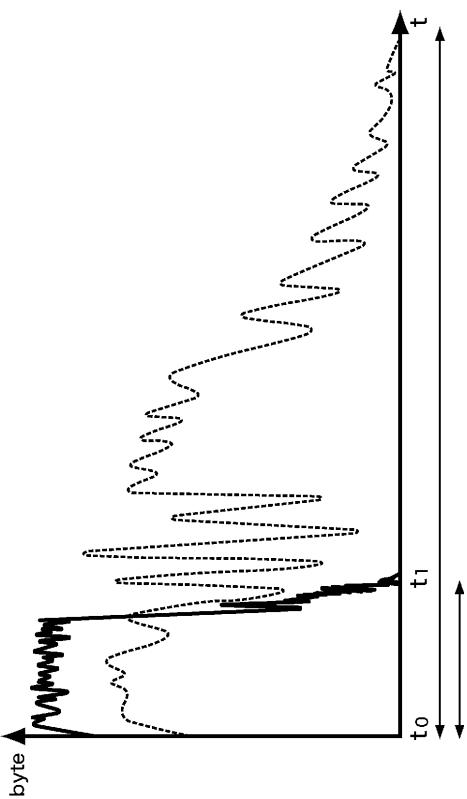
【図8】

図8

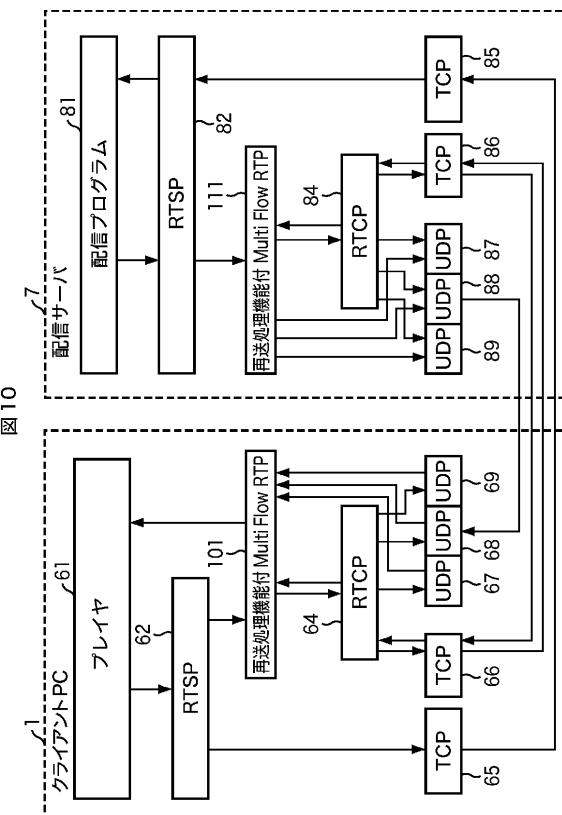


【図9】

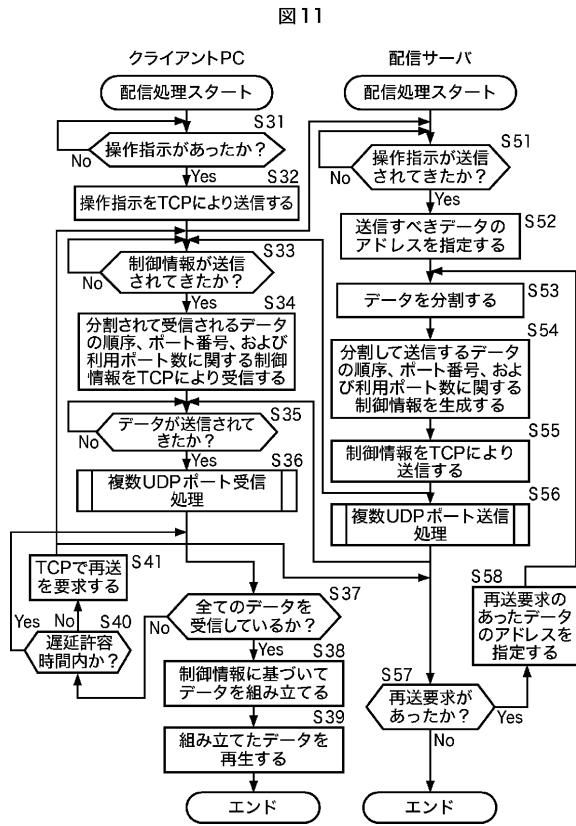
図9



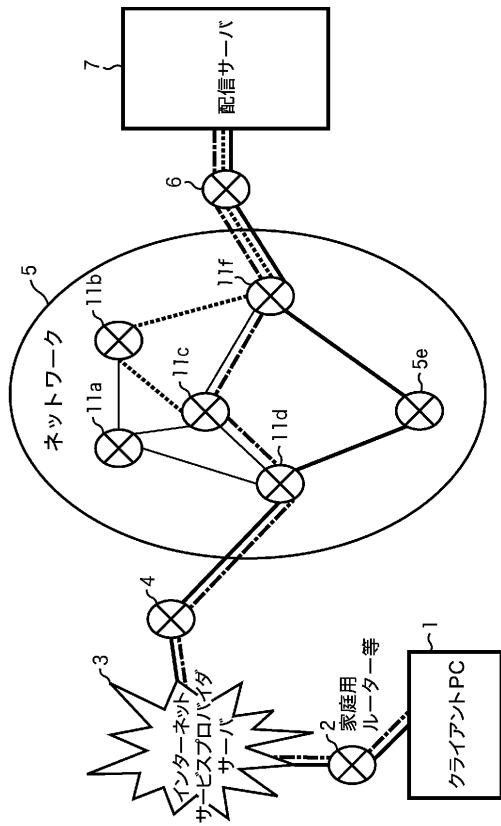
【図10】



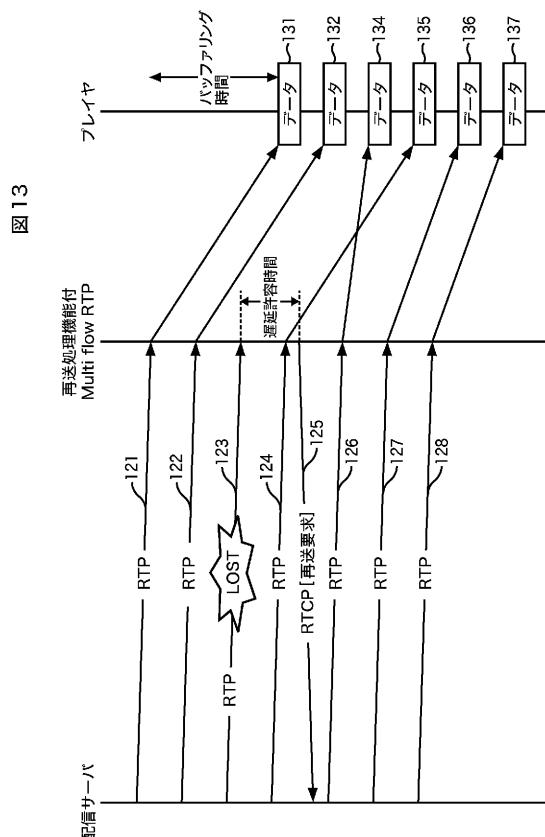
【図11】



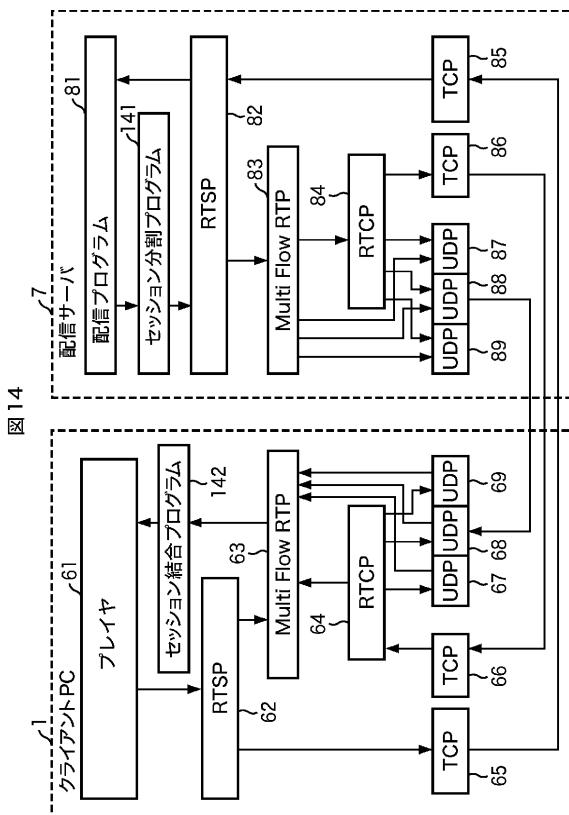
【図12】



【図13】



【図14】



【図15】

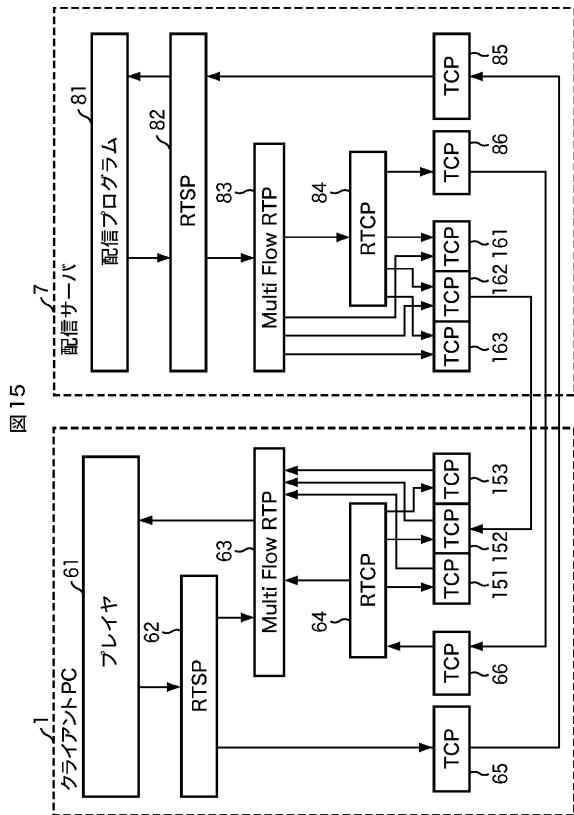


図15

【図16】

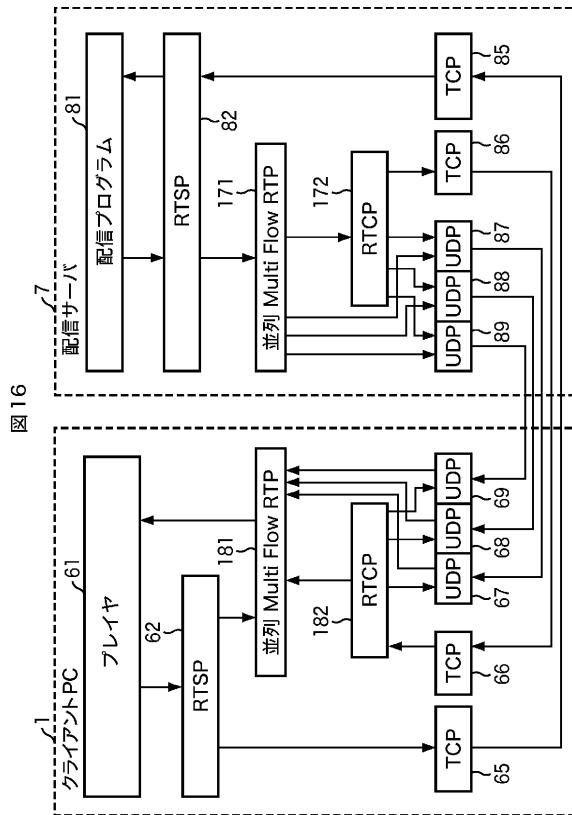
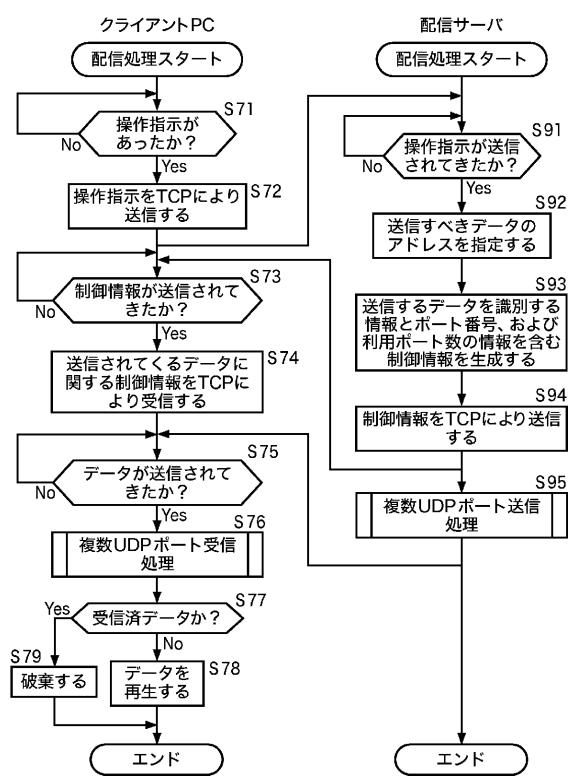


図16

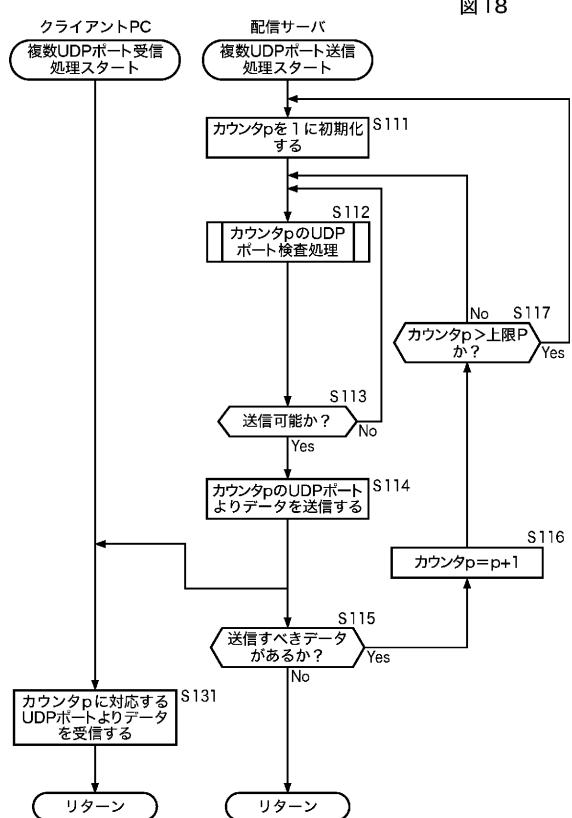
【図17】

図17

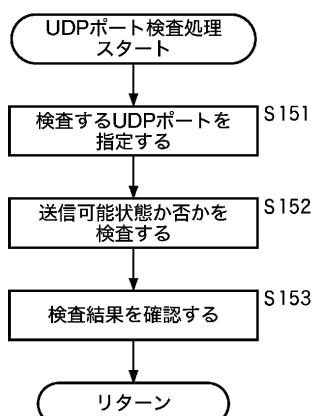


【図18】

図18

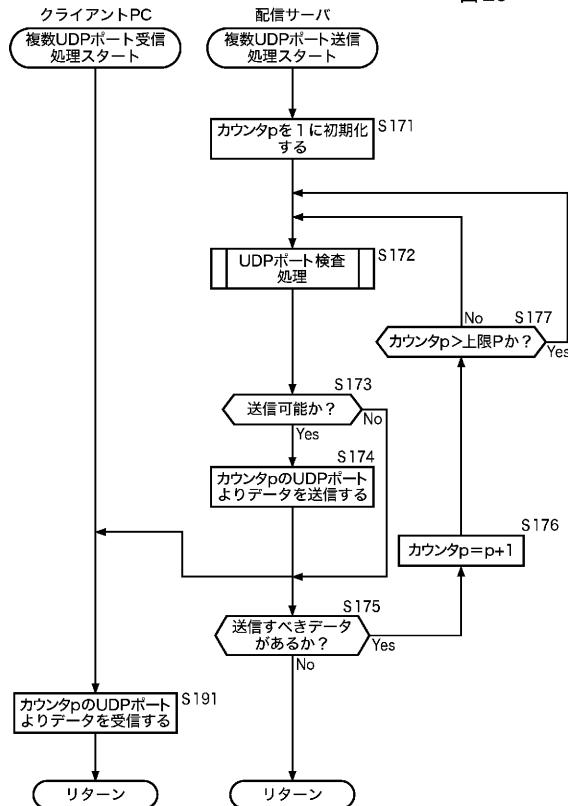


【図19】
図19



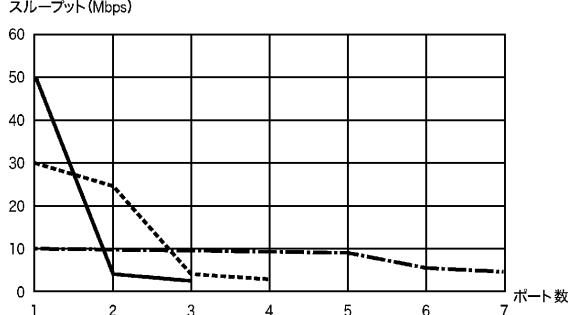
【図20】

図20



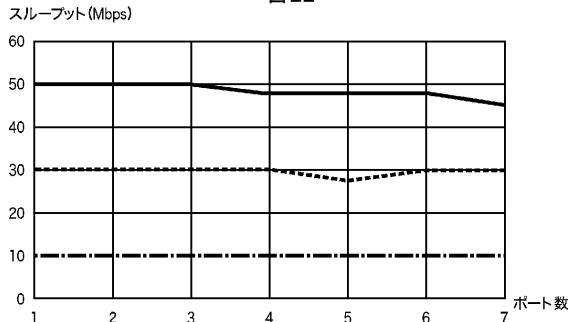
【図21】

図21



【図22】

図22



フロントページの続き

(56)参考文献 特開2003-110604(JP,A)
特開2002-185488(JP,A)
特開平06-350671(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04L 29/06
H04L 12/56
H04N 7/173