

(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 I		
HO 4 L	29/06	(2006. 01)	HO 4 L	13/00 3 O 5 C
HO 4 L	12/56	(2006. 01)	HO 4 L	12/56 1 O O A
HO 4 N	7/173	(2006. 01)	HO 4 N	7/173 6 2 O D

請求項の数 17 (全 30 頁)

(21) 出願番号	特願2004-282086 (P2004-282086)	(73) 特許権者	000002185
(22) 出願日	平成16年9月28日 (2004. 9. 28)		ソニー株式会社
(65) 公開番号	特開2005-244929 (P2005-244929A)		東京都港区港南1丁目7番1号
(43) 公開日	平成17年9月8日 (2005. 9. 8)	(74) 代理人	100082131
審査請求日	平成19年9月6日 (2007. 9. 6)		弁理士 稲本 義雄
(31) 優先権主張番号	特願2004-23126 (P2004-23126)	(72) 発明者	河村 聖悟
(32) 優先日	平成16年1月30日 (2004. 1. 30)		東京都品川区北品川6丁目7番35号 ソ
(33) 優先権主張国	日本国 (JP)	(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
			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

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 Protokol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 Protokol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

10

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 Protokol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 Protokol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 Protokol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

20

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 Protokol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 Protokol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】

30

前記トランスポート層の第 1 の Protokolは、送信したデータの着信確認をしない Protokol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 Protokol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 Protokol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。

40

【請求項 13】

トランスポート層の第 1 の Protokol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 Protokolで送信する第 2 の送信ステップと、

セッション層の Protokol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。

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

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。

20

【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

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】

50

【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）。

【特許文献1】特開2002-017637号公報

30

【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、輻輳制御が行われられないため、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ため、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】

30

【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。

【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に制御させるようにすることができる。

40

【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に制御させ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に制御させるようにすることができる。

50

【 0 0 1 9 】

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させるようにすることができる。

【 0 0 2 0 】

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することができる。

【 0 0 2 1 】

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させるようにすることができる。

10

【 0 0 2 2 】

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。

【 0 0 2 3 】

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。

【 0 0 2 4 】

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とするようにすることができる。

【 0 0 2 5 】

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

20

【 0 0 2 6 】

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

【 0 0 2 7 】

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

【 0 0 2 8 】

本発明の第 1 の記録媒体のプログラムは、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、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

【 0 0 2 9 】

本発明の第 1 のプログラムは、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、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

【 0 0 3 0 】

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。

【 0 0 4 1 】

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

【 0 0 4 2 】

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

【 0 0 4 3 】

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。

【 0 0 4 4 】

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

【 0 0 4 5 】

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。

40

【 発明の効果 】

【 0 0 4 6 】

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 】

【 0 0 4 7 】

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

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。

【 0 0 4 8 】

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

【 0 0 4 9 】

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。

【 0 0 5 0 】

20

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することができる。

【 0 0 5 1 】

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

【 0 0 5 2 】

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

【 0 0 5 4 】

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。

【 0 0 5 5 】

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。

【 0 0 5 6 】

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ビスプロバイダサーバ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PC 1 に配信する。

【 0 0 6 4 】

次に、図 3 を参照して、クライアントPC 1 の構成について説明する。

【 0 0 6 5 】

CPU (Central Processing Unit) 2 1 は、ROM (Read Only Memory) 2 2 、または記憶部 2 8 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) 2 3 には、CPU 2 1 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CPU 2 1 、ROM 2 2 、およびRAM 2 3 は、バス 2 4 により相互に接続されている。

【 0 0 6 6 】

CPU 2 1 には、バス 2 4 を介して入出力インタフェース 2 5 が接続されている。入出力インタフェース 2 5 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 2 6 、ディスプレイ、スピーカなどよりなる出力部 6 7 が接続されている。CPU 2 1 は、入力部 2 6 から入力される指令に対応して各種の処理を実行する。

【 0 0 6 7 】

入出力インタフェース 2 5 に接続されている記憶部 2 8 は、例えばハードディスクなどで構成され、CPU 2 1 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 2 9 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。

【 0 0 6 8 】

入出力インタフェース 2 5 に接続されているドライブ 3 0 は、磁気ディスク 4 1 、光ディスク 4 2 、光磁気ディスク 4 3 、或いは半導体メモリ 4 4 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 2 8 に転送され、記憶される。

【 0 0 6 9 】

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PC 1 と同様の構成であるので、その説明は省略するが、ROM 2 2 、および、記憶部 2 8 、並びに、ドライブ 3 0 に装着された磁気ディスク 4 1 、光ディスク 4 2 、光磁気ディスク 4 3 、或いは半導体メモリ 4 4 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PC 1 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CPU 2 1 の処理速度や、RAM 2 3 、および、記憶部 2 8 の容量を変更させるようにしても良い。

【 0 0 7 0 】

次に、図 3 を参照して説明したクライアントPC 1 、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 , 1 1 、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。

【 0 0 7 1 】

プレイヤー 6 1 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 (Graphic al 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) 6 2 に供給し、TCP (Transmission Control Protocol) ポート 6 5 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 RTP 6 3 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) 。

【 0 0 7 2 】

RTSPは、リアルタイムストリームデータの配信を制御し、プレイヤー 6 1 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 6 5 を介し

10

20

30

40

50

て配信サーバ7に供給する。

【0073】

また、RTSP62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、RTSP62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 RTP63およびRTCP64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 RTP63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 RTP63は、RTSP62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 RTP63は、RTCP64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 RTP63は、分割された状態で受信したストリームデータを、RTCP64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RTSP82に供給する。

【0077】

RTSP82は、基本的には、クライアントPC1のRTSP62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 RTP83に供給する。また、RTSP82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 RTP83およびRTCP84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 RTP83は、基本的には、クライアントPC1のMulti Flow RTP63と対応するものであるが、RTSP82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とを含む制御情報を生成して、RTCP84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 RTP83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式である。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、即時に送信すべきデータを割振って送信させる方式である。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 84は、基本的には、クライアントPC1のRTCP 64と同様のものであるが、Multi Flow RTP 63より供給された制御情報に基づいて、UDPポート87乃至89を制御して、クライアントPC1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86を制御してクライアントPC1に送信する。

【0082】

図4に示したプレイヤ61、RTSP 62、Multi Flow RTP 63、RTCP 64、TCP 65、66、および、UDP 67乃至69、並びに、配信プログラム81、RTSP 82、Multi Flow RTP 83、RTCP 84、TCP 85、86、および、UDP 87乃至89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。

【0083】

すなわち、プレイヤ61、および、配信プログラム81は、アプリケーション層であり、クライアントPC1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。

【0084】

RTSP 62、82、Multi Flow RTP 63、83、RTCP 64、84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65、66、85、86、並びに、UDPポート67乃至69、87乃至89は、トランスポート層であり、論理的な通信路を規定する。

【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。

【0087】

すなわち、RTSP 62、82は、同じセッション層のMulti Flow RTP 63、83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65、85をそれぞれ制御する。また、Multi Flow RTP 63、83が、同じセッション層におけるRTCP 64、84をそれぞれ制御する。また、セッション層のRTCP 64、84は、トランスポート層のUDP 67乃至69、87乃至89、および、トランスポート層のTCP 66、86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PC1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 67と87、UDP 68と88、および、UDP 69と89が、それぞれペアを形成し、データがそれぞれのペア毎

10

20

30

40

50

に授受される。

【 0 0 8 8 】

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 6 3 , 8 3 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 6 7 と 8 7、UDPポート 6 8 と 8 8、または、UDPポート 6 9 と 8 9）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。

【 0 0 8 9 】

この場合、Multi Flow RTP 6 3 , 8 3 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 6 4 , 8 4 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 6 7 乃至 6 9、および、UDPポート 8 7 乃至 8 9 を 1 つとして捉え、RTPセッション全体を制御する。

【 0 0 9 0 】

このとき、Multi Flow RTP 6 3 , 8 3 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。

【 0 0 9 1 】

また、通常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 8 3 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 6 3 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容易になり、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することができる。

【 0 0 9 2 】

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。

【 0 0 9 3 】

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PC 1 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。

【 0 0 9 4 】

ステップ S 1 において、プレイヤ 6 1 は、入力部 2 6 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 2 6 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 S 2 に進む。

【 0 0 9 5 】

ステップ S 2 において、プレイヤ 6 1 は、ストリームデータの配信を要求する操作指示をRTSP 6 2 に供給して、TCPポート 6 5 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、以降においては、同様の経路を経てクライアントPC 1 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。

【 0 0 9 6 】

ステップ S 1 1 において、配信プログラム 8 1 は、RTSP 8 2 よりTCPポート 8 5 を介して、ストリームデータの配信がクライアントPC 1 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PC 1 より要求されたと判定されるまで、その処理を繰り返す。

返す。ステップS 1 1において、例えば、ステップS 2の処理により、ストリームデータの配信がクライアントPC 1より要求されたと判定された場合、配信プログラム 8 1は、RTSP 8 2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 RTP 8 3に供給させる。

【 0 0 9 7 】

ステップS 1 3において、Multi Flow RTP 8 3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て、UDPポート 8 7乃至 8 9に対応する数に分割し、UDPポート 8 7乃至 8 9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RTCP 8 4に供給する。

【 0 0 9 8 】

ステップS 1 4において、Multi Flow RTP 8 3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。

【 0 0 9 9 】

ステップS 1 5において、RTCP 8 4は、TCPポート 8 6を制御して、制御情報をクライアントPC 1に送信させる。

【 0 1 0 0 】

ステップS 1 6において、複数UDPポート送信処理が実行され、RTCP 8 4により、UDPポート 8 7乃至 8 9が制御されて、Multi Flow RTP 8 3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PC 1に送信される。尚、複数UDPポート送信処理については、図 1 8乃至図 2 2を参照して後述する。

【 0 1 0 1 】

ステップS 3において、Multi Flow RTP 6 3は、RTCP 6 4を制御して、TCPポート 6 6に問い合わせ、分割されて各UDPポート 6 7乃至 6 9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 6 7乃至 6 9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S 3において、ステップS 1 5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S 4に進む。

【 0 1 0 2 】

ステップS 4において、RTCP 6 4は、制御情報をTCPポート 6 6を制御して受信させて、さらに、Multi Flow RTP 6 3に供給する。

【 0 1 0 3 】

ステップS 5において、RTCP 6 4は、制御情報の通信に利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 6 7乃至 6 9に問い合わせ、それぞれ対応するUDPポート 8 7乃至 8 9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S 5において、例えば、ステップS 1 6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S 6に進む。

【 0 1 0 4 】

ステップS 6において、複数UDPポート受信処理が実行され、RTCP 6 4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 6 7乃至 6 9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 6 7乃至 6 9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 RTP 6 3に供給される。尚、複数UDPポート受信処理については、図 1 8乃至図 2 2を参照して後述する。

【 0 1 0 5 】

ステップS 7において、Multi Flow RTP 6 3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、プレイヤー 6 1に供給する。

【 0 1 0 6 】

ステップS 8において、プレイヤ6 1は、Multi Flow RTP 6 3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6 7と8 7、UDPポート6 8と8 8、または、UDPポート6 9と8 9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6 7と8 7のRTPデータフローは、ルータ6, 1 1 f, 1 1 e, 1 1 d, 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6 8と8 8のRTPデータフローは、ルータ6, 1 1 f, 1 1 b, 1 1 c, 1 1 d, 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6 9と8 9のRTPデータフローは、ルータ6, 1 1 f, 1 1 c, 1 1 d, 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t 1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t 1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 RTP 1 0 1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。

【0111】

図10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に代えて、再送処理機能付Multi Flow RTP 1 0 1, 1 1 1を設けた点である。再送処理機能付Multi Flow RTP 1 0 1は、基本的にMulti Flow RTP 6 3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6 7乃至6 9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、RTCP 6 4を制御してTCPポート6 6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 RTP 1 1 1は、RTCP 8 4を制御して、TCPポート8 6より再送処理機能付Multi Flow RTP 1 0 1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8 7乃至8 9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 PC 1 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 S 31 乃至 S 36, S 38, S 39 の処理、および、ステップ S 51 乃至 S 56 の処理は、図 7 のフローチャートのステップ S 1 乃至 S 8 および S 11 乃至 S 16 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 S 37 において、再送処理機能付 Multi Flow RTP 101 は、RTCP 64 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 S 40 に進む。

10

【0114】

ステップ S 40 において、再送処理機能付 Multi Flow RTP 101 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 S 37 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 S 37, S 40 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 S 41 において、再送処理機能付 Multi Flow RTP 101 は、RTCP 64 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 S 33 に戻り、それ以降の処理が繰り返される。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 S 57 において、再送処理機能付 Multi Flow RTP 111 は、RTCP 84 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 S 41 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 S 58 に進む。

【0116】

ステップ S 58 において、再送処理機能付 Multi Flow RTP 111 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 S 53 に戻り、それ以降の処理が繰り返される。

30

【0117】

ステップ S 37 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 S 38 に進む。また、ステップ S 57 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 から送信されてくると図 11 のフローチャートのステップ S 38 の処理により、プレイヤー 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、クライアント PC 1 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てくることにより、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ー 61 に供給されないことになる。図 13 においては、その後送信されてくる 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 S 41 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ー 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S 7 6において、複数UDPポート受信処理が実行され、UDPポート8 7乃至8 9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S 7 7に進む。

【0 1 3 6】

ステップS 7 7において、並列Multi Flow RTP 1 7 1は、UDPポート6 7乃至6 9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 RTP 1 7 1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、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S 7 6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S 7 7に進む。

10

【0 1 3 7】

ステップS 7 8において、並列Multi Flow RTP 1 7 1は、受信したストリームデータをプレイヤー6 1に供給し、再生させる。

【0 1 3 8】

ステップS 7 7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S 9 4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S 7 9において、並列Multi Flow RTP 1 7 1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

【0 1 3 9】

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

【0 1 4 0】

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、利用可能帯域を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

【0 1 4 1】

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。

【0 1 4 2】

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側にキャ

50

ッシュ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、一般的な通信が単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。

【 0 1 4 3 】

次に、図 1 8 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 1 8 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 S 6 , S 1 6 の詳細な処理として説明するが、図 1 1 のフローチャートにおけるステップ S 3 6 , S 5 6、および、図 1 7 のフローチャートにおけるステップ S 7 6 , S 9 5 の処理も同様である。

10

【 0 1 4 4 】

ステップ S 1 1 1 において、Multi Flow RTP 8 3 は、UDPポート 8 7 乃至 8 9 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 8 7 に、カウンタ p = 2 は、UDPポート 8 8 に、カウンタ p = 3 は、UDPポート 8 9 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。

【 0 1 4 5 】

ステップ S 1 1 2 において、Multi Flow RTP 8 3 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

【 0 1 4 6 】

ここで、図 1 9 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。

【 0 1 4 7 】

ステップ S 1 5 1 において、Multi Flow RTP 8 3 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 RTP 8 3 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

【 0 1 4 8 】

ステップ S 1 5 2 において、Multi Flow RTP 8 3 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 RTP 8 3 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)。

【 0 1 4 9 】

ステップ S 1 5 3 において、Multi Flow RTP 8 3 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 RTP 8 3 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

【 0 1 5 0 】

以上のように、ステップ S 1 5 1 乃至 S 1 5 3 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。

【 0 1 5 1 】

ここで、図 1 8 のフローチャートに戻る。

【 0 1 5 2 】

ステップ S 1 1 3 において、Multi Flow RTP 8 3 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 S 1 1 2 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 S 1 1 2 , S 1 1 3 の処理が繰り返される。

【 0 1 5 3 】

そして、ステップ S 1 1 3 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 S 1 1 4 に進む。

【 0 1 5 4 】

ステップ S 1 1 4 において、RTCP 8 4 は、カウンタ p に対応するUDPポートを制御して、Multi Flow RTP 8 3 により分割されたストリームデータをクライアントPC 1 に送信させる。

【 0 1 5 5 】

ステップ S 1 3 1 において、RTCP 6 4 は、制御情報に含まれている通信に利用されるUDPポート 6 7 乃至 6 9 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 6 7 乃至 6 9 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 RTP 6 3 に供給させる。すなわち、UDPポート 6 7 乃至 6 9 のうち、上述したカウンタ p に対応するUDPポート 8 7 乃至 8 9 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。

【 0 1 5 6 】

一方、ステップ S 1 1 5 において、Multi Flow RTP 8 3 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。

【 0 1 5 7 】

ステップ S 1 1 5 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 S 1 1 6 において、Multi Flow RTP 8 3 は、カウンタ p を 1 インクリメントする。

【 0 1 5 8 】

ステップ S 1 1 7 において、Multi Flow RTP 8 3 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 S 1 1 2 に戻る。

【 0 1 5 9 】

また、ステップ S 1 1 7 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 S 1 1 1 に戻る。

【 0 1 6 0 】

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 8 7 , 8 8 , 8 9 の順番に送信可能であるかが判定され、UDPポート 8 9 の検査が終了すると、再びUDPポート 8 7 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 8 7 , 8 8 , 8 9 , 8 7 , 8 8 , 8 9 . . 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PC 1 は、UDPポート 6 7 , 6 8 , 6 9 , 6 7 , 6 8 , 6 9 . . の順番に受信されることになる。

【 0 1 6 1 】

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 1 7 においては、複数のUDPポート 8 7 , 8 8 , 8 9 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。

【 0 1 6 2 】

次に、図 1 9 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 1 9 のフローチャートにおけるステップ S 1 7 1 乃至 S 1 7 7 , S 1 9 1 の処理は、図 1 7 のフローチャートにおけるステ

10

20

30

40

50

ップ S 1 1 1 乃至 S 1 1 7 , S 1 3 1 の処理と同様であるので、その説明は省略する。

【 0 1 6 3 】

すなわち、図 1 9 のフローチャートの各処理と図 1 7 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 1 7 においては、ステップ S 1 7 4 において、送信不能であると判定された場合、ステップ S 1 7 4 の処理がスキップされて、その処理は、ステップ S 1 7 5 に進む。

【 0 1 6 4 】

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 8 7 乃至 8 9 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

【 0 1 6 5 】

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。

【 0 1 6 6 】

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

【 0 1 6 7 】

図 2 1 , 図 2 2 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 2 1 , 図 2 2 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。

【 0 1 6 8 】

図 2 1 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

【 0 1 6 9 】

一方、図 2 2 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。

【 0 1 7 0 】

結果として、図 2 1 , 図 2 2 からわかるように、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、データを送受信す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 2 1 , 図 2 2 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

【 0 1 7 1 】

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。

【 0 1 7 2 】

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

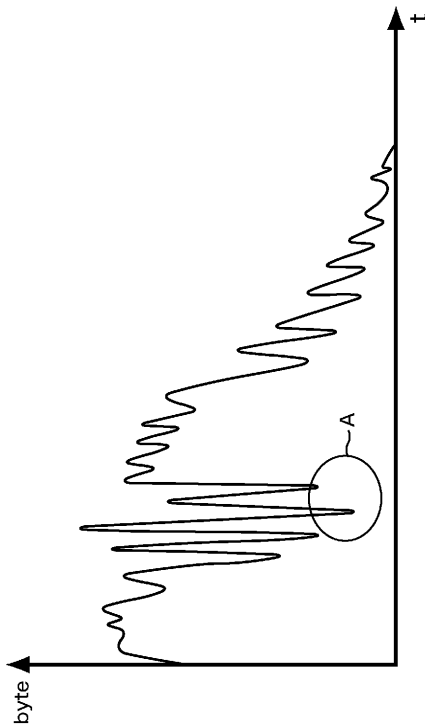
【符号の説明】

【 0 1 7 7 】

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

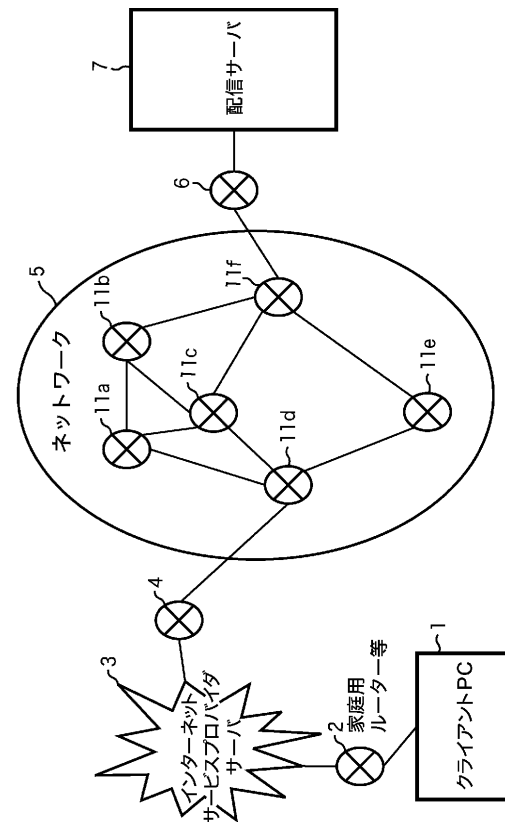
【図1】

図1



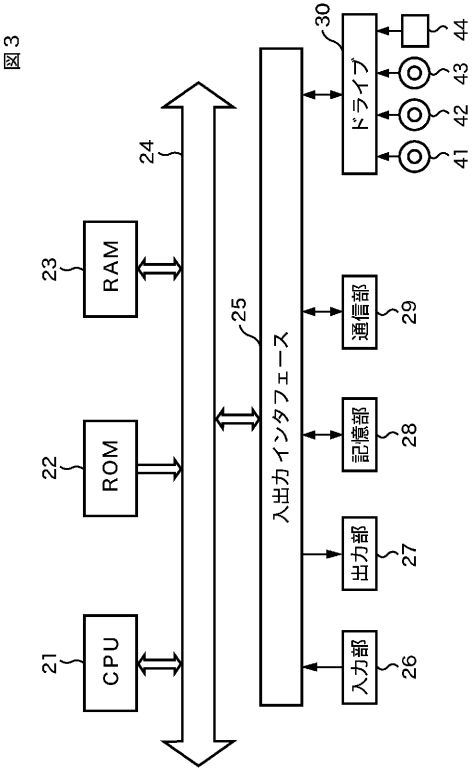
【図2】

図2



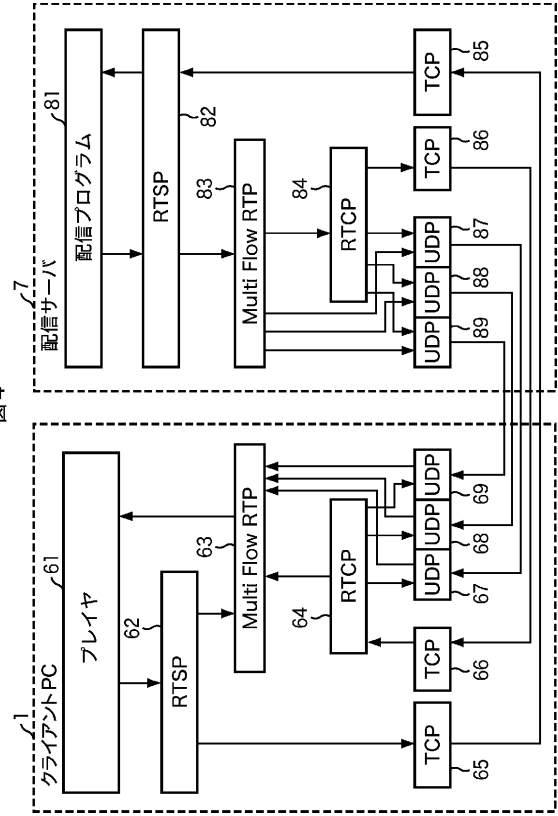
【図 3】

図 3



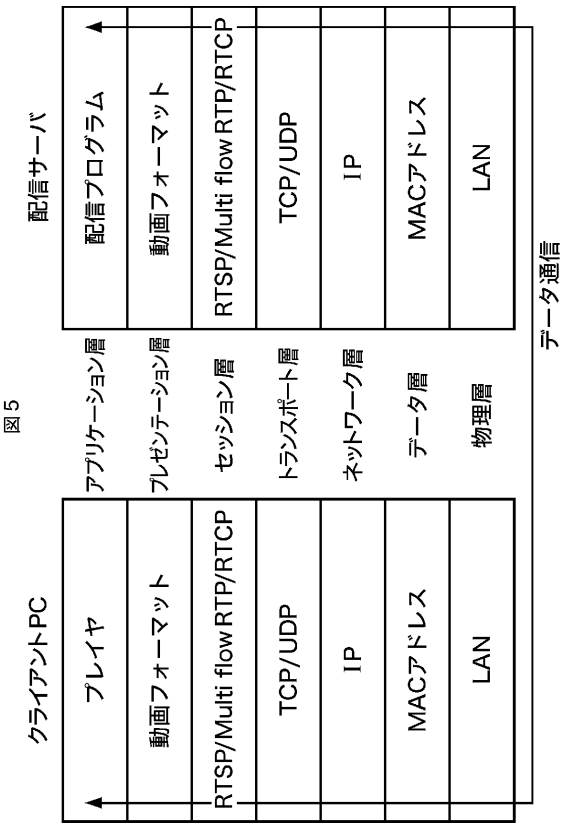
【図 4】

図 4



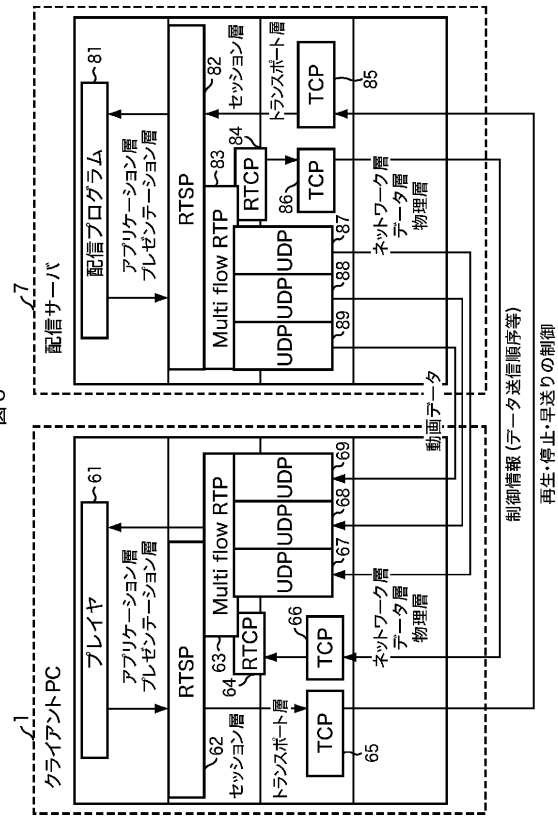
【図 5】

図 5



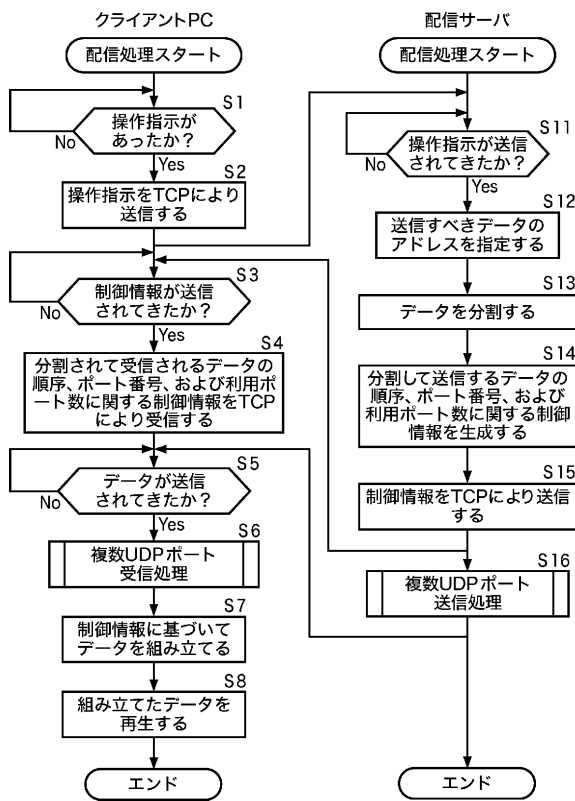
【図 6】

図 6



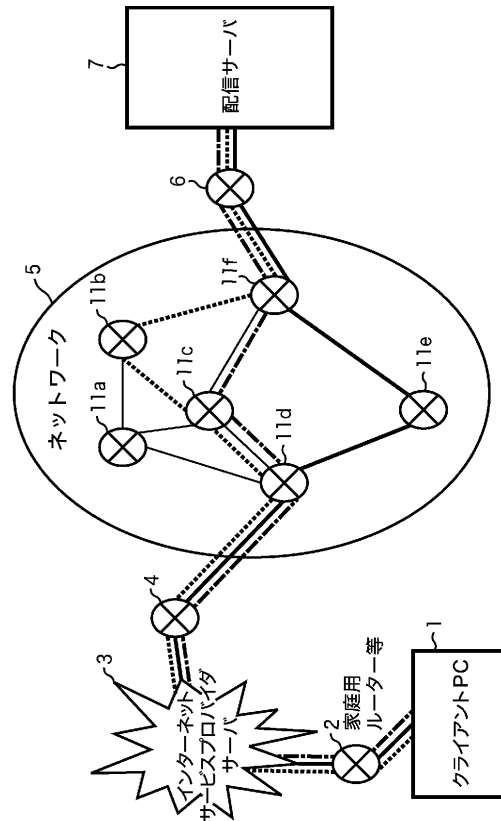
【図 7】

図 7



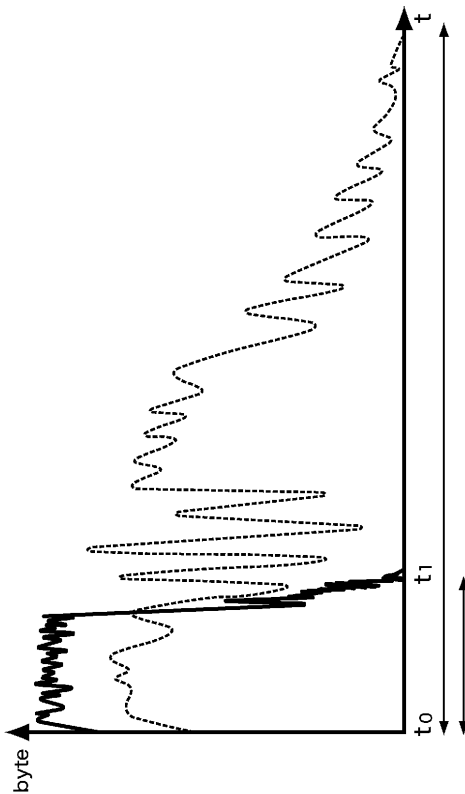
【図 8】

図 8



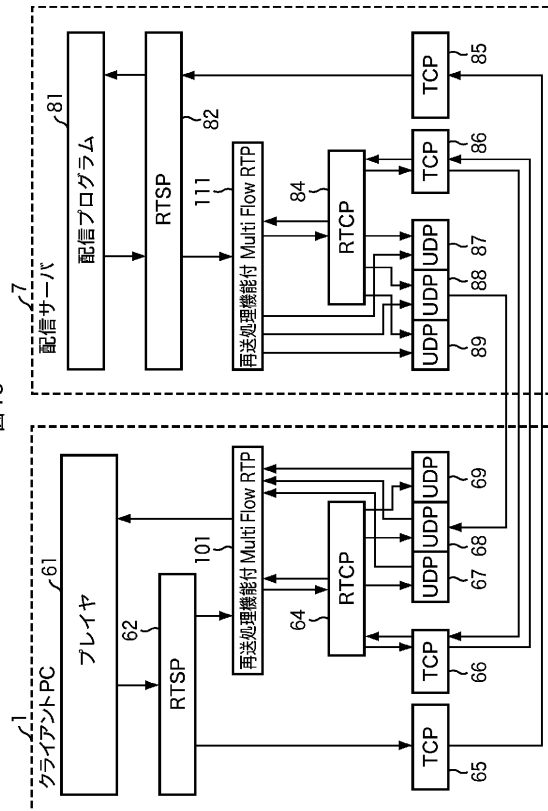
【図 9】

図 9

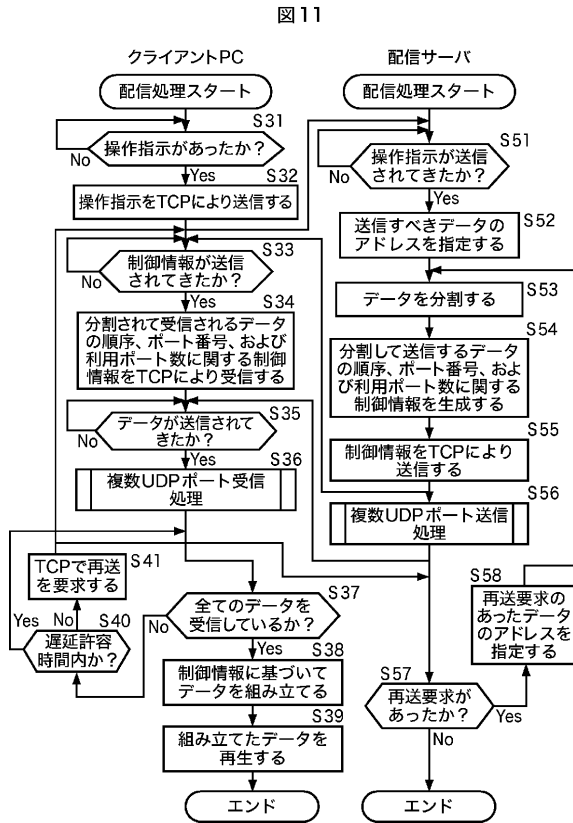


【図 10】

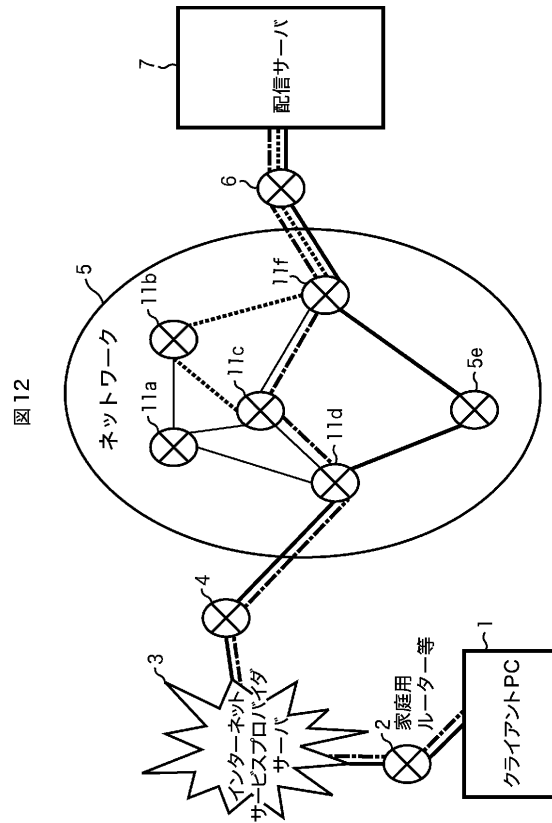
図 10



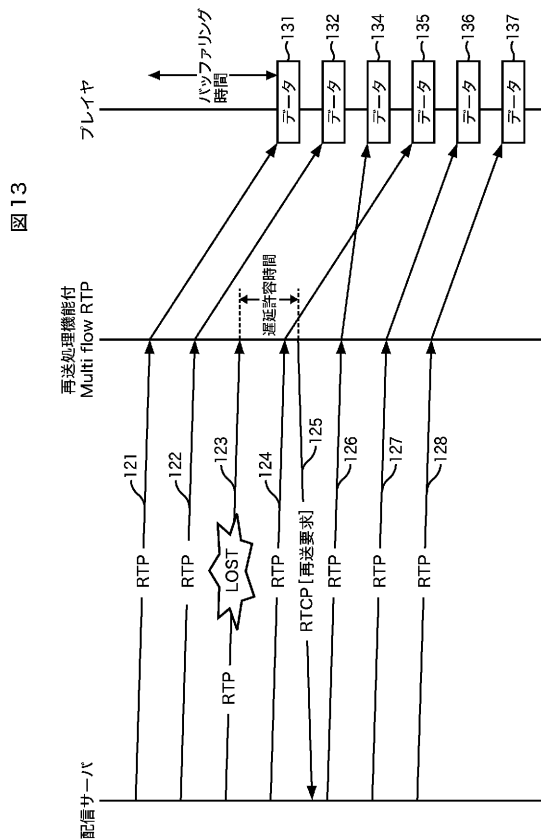
【図 1 1】



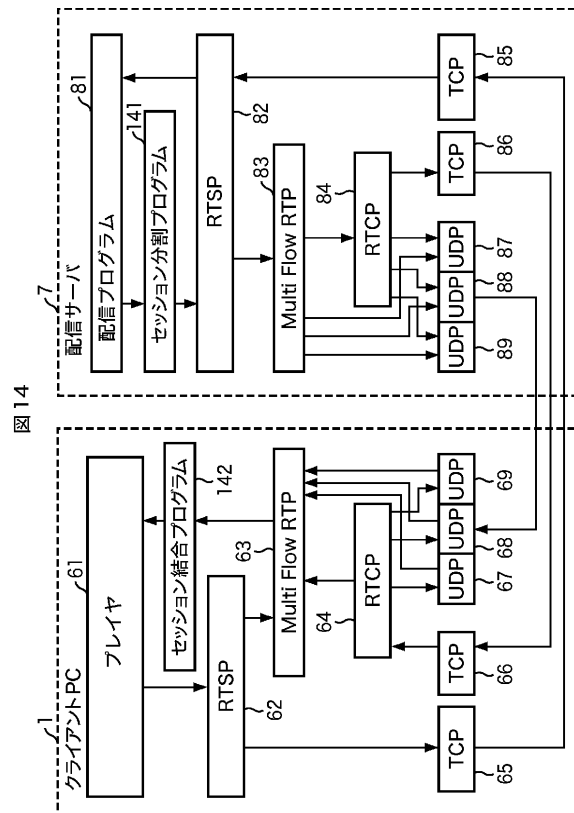
【図 1 2】



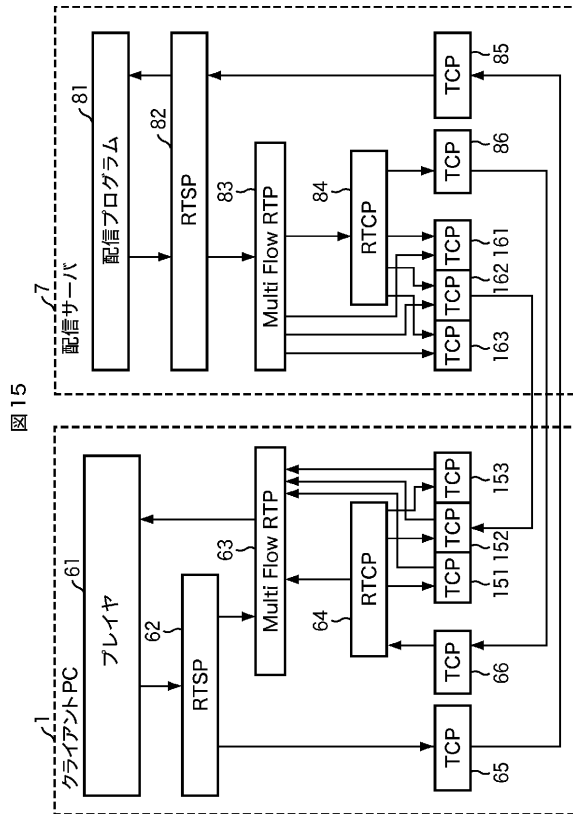
【図 1 3】



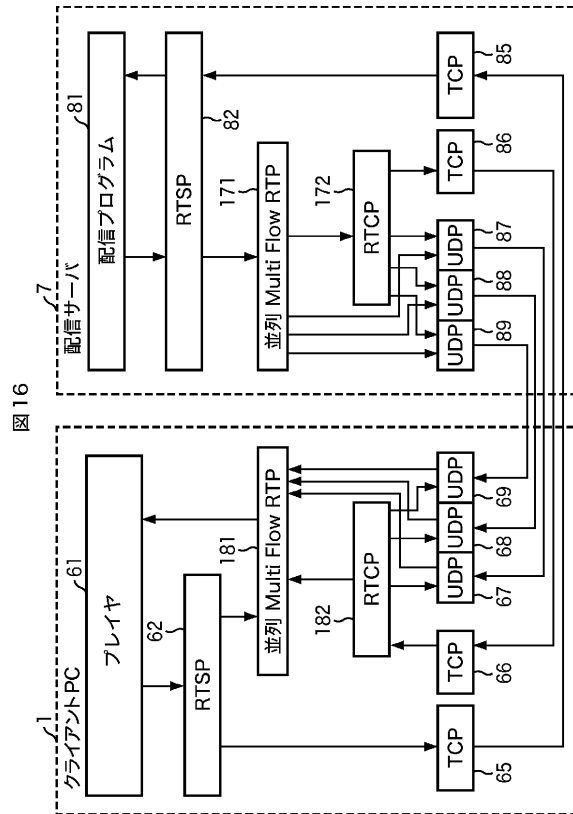
【図 1 4】



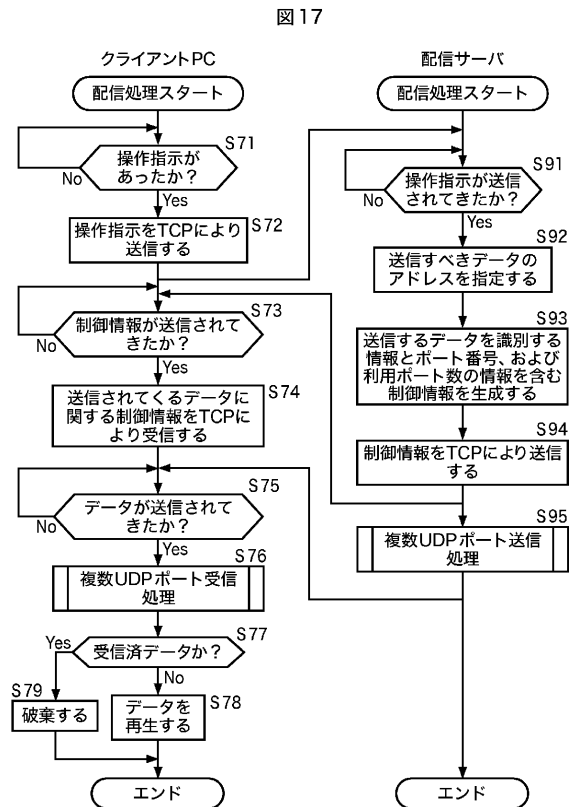
【 図 1 5 】



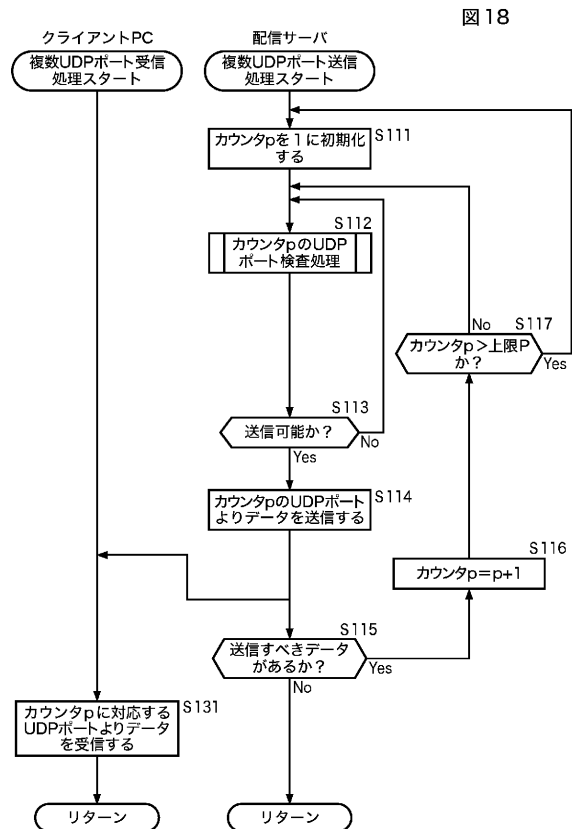
【 図 1 6 】



【 図 1 7 】

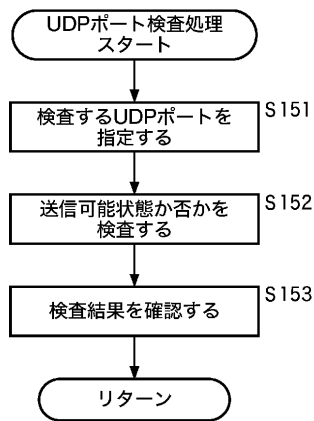


【 図 1 8 】



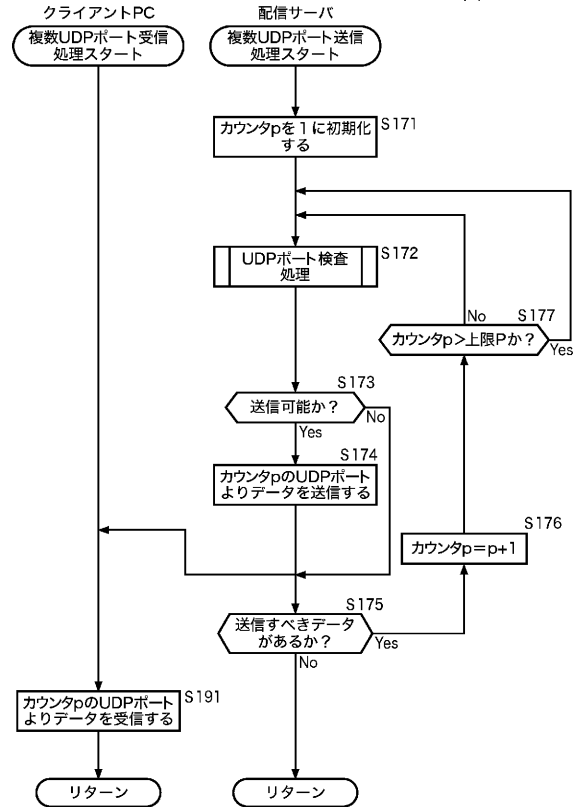
【図 19】

図 19



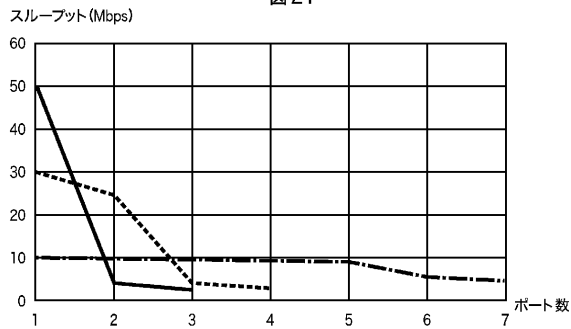
【図 20】

図 20



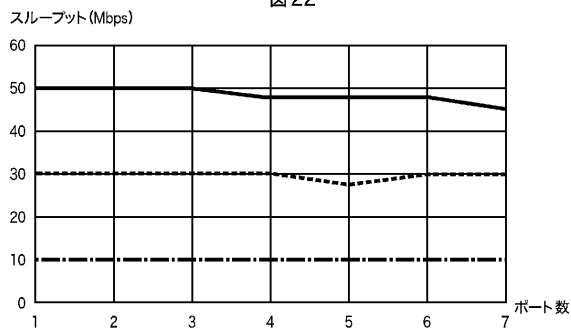
【図 21】

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