(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6284549号
(P6284549)
(45)発行日　平成30年2月28日(2018.2.28)　(24)登録日　平成30年2月9日(2018.2.9)

(51)Int.Cl.　F I
H04L 1/00 (2006.01)　H04L 1/00 B

請求項の数 21　(全 48 頁)

(21)出願番号	特願2015-553843 (P2015-553843)
(86)(22)出願日	平成26年1月17日 (2014.1.17)
(65)公表番号	特表2016-504005 (P2016-504005A)
(43)公表日	平成28年2月8日 (2016.2.8)
(86)国際出願番号	PCT/US2014/011988
(87)国際公開番号	WO2014/113636
(87)国際公開日	平成26年7月24日 (2014.7.24)
審査請求日	平成28年10月11日 (2016.10.11)
(31)優先権主張番号	61/753,884
(32)優先日	平成25年1月17日 (2013.1.17)
(33)優先権主張国	米国 (US)
(31)優先権主張番号	61/818,106
(32)優先日	平成25年5月1日 (2013.5.1)
(33)優先権主張国	米国 (US)

(73)特許権者　507364838
クアルコム，インコーポレイテッド
アメリカ合衆国　カリフォルニア　９２１２１　サン　ディエゴ　モアハウス　ドライブ　５７７５
(74)代理人　100108453
弁理士　村山　靖彦
(74)代理人　100163522
弁理士　黒田　晋平
(72)発明者　マイケル・ジョージ・ルビー
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５

最終頁に続く

(54)【発明の名称】マルチパスストリーミングのためのＦＥＣベースの信頼性のある転送制御プロトコル

(57)【特許請求の範囲】
【請求項１】
コンピュータ可読記憶媒体であって、
クライアントデバイスのプロセッサによって実行されると、前記プロセッサに、
サーバデバイスから複数の並列ネットワークパスを介して前方誤り訂正されたデータを受信させる
命令を記憶し、
前記受信されたデータが、データのブロックに対する1つまたは複数の符号化単位を含み、
前記プロセッサに、前記前方誤り訂正されたデータを受信させる前記命令が、前記プロセッサに、
転送プロトコルのレート制御プロトコルから独立したインターリーブされた信頼性制御プロトコルを含む前記転送プロトコルを実行させる
命令を含み、
前記転送プロトコルが、転送制御プロトコル(TCP)とは異なり、かつ、TCPを含まず、
前記プロセッサに、
前記並列ネットワークパスの各々での前記データのロスを決定させ、
前記並列ネットワークパスの各々での前記データのフィードバックデータを前記サーバデバイスへ送信させる
命令を記憶し、

前記プロセッサに、前記フィードバックデータを送信させる前記命令が、前記プロセッサに、

前記並列ネットワークパスの各々に対して、それぞれの並列ネットワークパスを介して受信されたパケットフローに対して受信された最大のシーケンス番号を特定するデータを送信させる

命令を含み、

前記プロセッサに、

前記並列ネットワークパスの各々での前記データの前記フィードバックデータに基づいて、データの前記ブロックに対する1つまたは複数の追加の符号化単位を受信させる

命令を記憶し、

前記追加の符号化単の数は、前記受信された最大のシーケンス番号と前記サーバデバイスによって送信されている現在のシーケンス番号とに基づいて計算された、送信された確認応答されていない符号化単位の数に基づいている、

前記プロセッサに、

前方誤り訂正を使用して、前記受信された符号化単位から前記ブロックを復元させる

命令を記憶した、コンピュータ可読記憶媒体。

【請求項２】

前記プロセッサに前記フィードバックデータを送信させる前記命令が、前記プロセッサに、符号化単位が受信された複数のブロックの各々を特定するデータ、前記ブロックの各々に対して必要とされる符号化単位の数、および、前記ブロックの各々に関するネットワークパケットに対して受信された最大のシーケンス番号を定義するデータを送信させる命令を含む、請求項1に記載のコンピュータ可読記憶媒体。

【請求項３】

さらに、前記プロセッサに、

前記サーバデバイスへの前記ブロックに対して必要とされる追加の符号化単位の数を計算させ、

追加の符号化単位の前記数を表すデータを前記サーバデバイスへ送信させる

命令を含む、請求項1に記載のコンピュータ可読記憶媒体。

【請求項４】

追加の符号化単位の前記数を表す前記データが、前記並列ネットワークパスの各々での前記データの前記フィードバックデータを含む、請求項3に記載のコンピュータ可読記憶媒体。

【請求項５】

前記プロセッサに、追加の符号化単位の前記数を計算させる前記命令が、前記プロセッサに、

前記ブロックを復元するために必要とされる符号化単位の数を、前記ブロックのサイズおよび前記ブロックに対する前記受信された符号化単位のサイズに基づいて計算させ、

追加の符号化単位の前記数を、前記ブロックを復元するために必要とされる符号化単位の前記数と受信された符号化単位の前記数との差として計算させる

命令を含む、請求項3に記載のコンピュータ可読記憶媒体。

【請求項６】

さらに、前記プロセッサに、

前記ブロックに対する符号化単位を受信させ、

前記ブロックがアクティブかどうかを判定させ、

前記ブロックがアクティブではないとき、前記符号化単位を廃棄させ、

前記ブロックがアクティブであるとき、使用されるべき前記ブロックに対する符号化単位のセットへ前記符号化単位を追加させて、前記ブロックを復元させる

命令を含む、請求項1に記載のコンピュータ可読記憶媒体。

【請求項７】

前記プロセッサに受信させる前記命令が、前記プロセッサに、複数のブロックに対する

符号化単位を受信させる命令を含み、
前記複数のブロックに対する前記符号化単位は、前記サーバデバイスで、前記複数のブロックに対する前記符号化単位が互いにインターリーブされる、請求項1に記載のコンピュータ可読記憶媒体。
【請求項８】
前記プロセッサに、前記符号化単位を受信させる前記命令が、前記プロセッサに、
前記並列ネットワークパスの第1のパスを通じて、第1のブロックに対する第1の符号化単位を受信させ、
前記第1の符号化単位を受信した後で、前記第1のパスを通じて、第2のブロックに対する第2の符号化単位を受信させ、
前記第2の符号化単位を受信した後で、前記第1のパスを通じて、前記第1のブロックに対する第3の符号化単位を受信させる
命令を含む、請求項7に記載のコンピュータ可読記憶媒体。
【請求項９】
コンピュータ可読記憶媒体であって、
サーバデバイスのプロセッサによって実行されると、前記プロセッサに、
クライアントデバイスへ複数の並列ネットワークパスを介して前方誤り訂正されたデータを送信させる
命令を記憶し、
前記送信されたデータが、データのブロックに対する1つまたは複数の符号化単位を含み、
前記プロセッサに、前記前方誤り訂正されたデータを送信させる前記命令が、前記プロセッサに、
転送プロトコルのレート制御プロトコルから独立したインターリーブされた信頼性制御プロトコルを含む前記転送プロトコルを実行させる
命令を含み、
前記転送プロトコルが、転送制御プロトコル(TCP)とは異なり、かつ、TCPを含まず、
前記プロセッサに、
前記クライアントデバイスから前記並列ネットワークパスの各々を通じて送信される前記データのフィードバックデータを受信させる
命令を記憶し、
前記プロセッサに、前記フィードバックデータを受信させる前記命令が、前記プロセッサに、
前記並列ネットワークパスの各々に対して、それぞれの並列ネットワークパスを介して送信されたパケットフローに対して受信された最大のシーケンス番号を特定するデータを受信させる
命令を含み、
前記プロセッサに、
前記フィードバックデータに基づいて、前記並列ネットワークパスを通じた後続のデータ送信のために送信される前方誤り訂正データの量を修正させ、
前記並列ネットワークパスの各々での前記データの前記フィードバックデータに基づいて、データの前記ブロックに対する1つまたは複数の追加の符号化単位を送信させる
命令を記憶し、前記追加の符号化単の数は、前記受信された最大のシーケンス番号と前記サーバデバイスによって送信されている現在のシーケンス番号とに基づいて計算された、送信された確認応答されていない符号化単位の数に基づいている、コンピュータ可読記憶媒体。
【請求項１０】
前記プロセッサに前記フィードバックデータを受信させる前記命令が、前記プロセッサに、符号化単位が前記クライアントデバイスによって受信された複数のブロックの各々を特定するデータ、前記ブロックの各々に対して必要とされる符号化単位の数、および、前

記ブロックの各々に関するネットワークパケットに対して前記クライアントデバイスによって受信された最大のシーケンス番号を定義するデータを受信させる命令を含む、請求項9に記載のコンピュータ可読記憶媒体。

【請求項１１】

さらに、前記プロセッサに、

前記クライアントデバイスから、前記ブロックに対して必要とされる追加の符号化単位の数を表すデータを受信させ、

前記ブロックに対する追加の符号化単位の前記数を前記クライアントデバイスへ送信させる

命令を含む、請求項9に記載のコンピュータ可読記憶媒体。

【請求項１２】

追加の符号化単位の前記数を表す前記データが、前記並列ネットワークパスの各々を通じて送信される前記データの前記フィードバックデータを含む、請求項11に記載のコンピュータ可読記憶媒体。

【請求項１３】

さらに、前記プロセッサに、

前記クライアントデバイスから、受信された符号化単位の数を表すデータを受信させ、

前記ブロックを復元するために必要とされる符号化単位の数を、前記ブロックのサイズおよび前記ブロックに対する前記受信された符号化単位のサイズに基づいて計算させ、

追加の符号化単位の数を、前記ブロックを復元するために必要とされる符号化単位の前記数と受信された符号化単位の前記数との差として計算させ、

追加の符号化単位の前記数を前記クライアントデバイスへ送信させる

命令を含む、請求項9に記載のコンピュータ可読記憶媒体。

【請求項１４】

前記プロセッサに送信させる前記命令が、前記プロセッサに、複数のブロックに対する符号化単位を送信させる命令を含み、

前記複数のブロックに対する前記符号化単位は、前記サーバデバイスで、前記複数のブロックに対する前記符号化単位が互いにインターリーブされる、請求項9に記載のコンピュータ可読記憶媒体。

【請求項１５】

前記プロセッサに、前記符号化単位を送信させる前記命令が、前記プロセッサに、

前記並列ネットワークパスの第1のパスを通じて、第1のブロックに対する第1の符号化単位を送信させ、

前記第1の符号化単位を送信した後で、前記第1のパスを通じて、第2のブロックに対する第2の符号化単位を送信させ、

前記第2の符号化単位を送信した後で、前記第1のパスを通じて、前記第1のブロックに対する第3の符号化単位を送信させる

命令を含む、請求項14に記載のコンピュータ可読記憶媒体。

【請求項１６】

前記プロセッサに、データの前記ブロックに対する前記符号化単位を、データのブロック全体が前記プロセッサに対して利用可能になる前に送信させる命令をさらに含む、請求項9に記載のコンピュータ可読記憶媒体。

【請求項１７】

前記プロセッサに、

第1のブロックに対する符号化単位の第1のセットを送信することであって、符号化単位の前記第1のセットが、前記第1のブロックを復元するために必要とされる符号化単位の最小の数よりも少ない符号化単位を含む、送信することと、

符号化単位の前記第1のセットを送信した後で、第2のブロックに対する符号化単位の第2のセットを送信することと、

符号化単位の前記第2のセットを送信した後で、前記第1のブロックに対する1つまたは

複数の符号化単位を含む符号化単位の第3のセットを送信することと
を行わせる命令をさらに含む、請求項9に記載のコンピュータ可読記憶媒体。
【請求項１８】
前記プロセッサに、符号化単位の前記第1のセットを送信させる前記命令が、前記プロセッサに、前記第1のブロックが完全に形成される前に、符号化単位の前記第1のセットを送信させる命令を含む、請求項17に記載のコンピュータ可読記憶媒体。
【請求項１９】
前記第1のブロックおよび前記第2のブロックが、メディアコンテンツに対するデータの複数のブロックのうちのブロックを含み、
前記プロセッサに、符号化単位の前記第1のセット、符号化単位の前記第2のセット、および符号化単位の前記第3のセットを送信させる前記命令が、前記プロセッサに、符号化単位の前記第1のセット、符号化単位の前記第2のセット、および符号化単位の前記第3のセットを、データの前記複数のブロックが送信される際に介する複数の並列ネットワークパスのうちの1つのパスを介して送信させる命令を含む、請求項17に記載のコンピュータ可読記憶媒体。
【請求項２０】
前記プロセッサに、
前記並列ネットワークパスの各々での前記データの前記フィードバックデータを前記クライアントデバイスから受信させる命令をさらに含む、請求項19に記載のコンピュータ可読記憶媒体。
【請求項２１】
前記プロセッサに、
符号化単位の前記第1のセットを送信した後で、フィードバックデータの第1のセットを受信することであって、フィードバックデータの前記第1のセットが、前記第1のブロックを特定するデータと、前記第1のブロックを復元するために必要とされる符号化単位の数を示すデータと、前記第1のブロックに関するネットワークパケットに対して受信された最大のシーケンス番号を定義するデータとを含む、受信することと、
符号化単位の前記第2のセットを受信した後で、フィードバックデータの第2のセットを受信することであって、フィードバックデータの前記第2のセットが、前記第2のブロックを特定するデータと、前記第2のブロックを復元するために必要とされる符号化単位の数を示すデータと、前記第2のブロックに関するネットワークパケットに対して受信された最大のシーケンス番号を定義するデータとを含む、受信することと
を行わせる命令をさらに含む、請求項17に記載のコンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、2013年1月17日に出願された米国仮出願第61/753,884号、および2013年5月1日に出願された米国仮出願第61/818,106号の利益を主張し、これらの米国仮出願の各々の内容全体が参照によって本明細書に組み込まれる。
【０００２】
本開示は、メディアデータの転送に関する。
【背景技術】
【０００３】
デバイスは、複数のパスと低レイテンシの処理とを使用したデータ通信ネットワークを通じて、エンドシステム間のデータの高速な送信を実行することができる。多くのデータ通信システムおよび高水準データ通信プロトコルが、信頼性のあるデータ転送の簡便な通信抽象化を提供し、レート制御を実現する。すなわち、それらは、ネットワークの条件に基づいてパケット送信レートを自動的に調整する。汎用的な転送制御プロトコル(TCP)のような、低水準のパケット化されたデータ転送に関する従来の基礎となる実装形態は、次の条件、すなわち、(a)送信機と受信機との間の接続が大きな往復遅延時間(RTT)を有する

こと、(b)データの量が多くネットワークがバースト的で過渡的なロスを被ることの少なくとも1つが発生すると、悪くなる。

【０００４】

1つの広く使用されている信頼性のある転送プロトコルは、転送制御プロトコル(TCP)である。TCPは、確認応答の機構を有する、一般に使用されている点対点のパケット制御方式である。TCPは、送信者と受信者の間にロスがほとんどなく送信者と受信者との間のRTTが小さいとき、1対1の信頼性のある通信について良好に動作する。しかしながら、TCPのスループットは、ロスが非常に少ないときであっても、または、送信者と受信者との間に大きなレイテンシがあるとき、大きく低下する。

【０００５】

TCPを使用して、送信者は命令されたパケットを送信し、受信者は各パケットの受信に確認応答する。パケットが失われると、確認応答は送信者に送信されず、後で受信されるパケットの受信とタイムアウトのいずれかに基づいて、送信者はパケットを再送信する。TCPのようなプロトコルによって、確認応答のパラダイムは完全な失敗を伴わずにパケットが失われることを可能にする。それは、失われたパケットは、確認応答の欠如または受信者からの明示的な要求のいずれかに応答して再送信されるだけであり得るからである。

【０００６】

TCPは、信頼性制御とレート制御の両方を提供する。すなわち、TCPを実装するデバイスは、元のデータのすべてが受信機に配信されることを確実にして、混雑またはパケットロスのようなネットワーク条件に基づいてパケット送信レートを自動的に調整する。TCPによって、信頼性制御プロトコルおよびレート制御プロトコルは絡み合い、分離可能ではなくなる。その上、増大するRTTおよびパケットロスに従うTCPのスループット性能は、最適からは程遠い。

【先行技術文献】

【特許文献】

【０００７】

【特許文献１】米国特許第6,307,487号
【特許文献２】米国特許第6,320,520号
【特許文献３】米国特許第6,373,406号
【特許文献４】米国特許第6,614,366号
【特許文献５】米国特許第6,411,223号
【特許文献６】米国特許第6,486,803号
【特許文献７】米国特許出願公開第2003/0058958号
【特許文献８】米国特許第7,447,235号

【非特許文献】

【０００８】

【非特許文献１】「A Modular Analysis of Network Transmission Protocols」、Micah Adler、Yair Bartal、John Byers、Michael Luby、Danny Raz、Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems、1997年6月

【発明の概要】

【課題を解決するための手段】

【０００９】

全般に、本開示は、複数の並列ネットワークパスを通じて送信されるデータに適用される前方誤り訂正(FEC)に関する技法を説明する。一例では、データは、転送されるべきデータをデータブロックへと編成することによって、送信機から受信機へと信頼性をもって転送されることが可能であり、各データブロックは複数の符号化単位を含む。第1のデータブロックの符号化単位は、複数のパスを通じて送信機から受信機に送信されてよく、すなわち、第1のデータブロックのいくつかの符号化単位はあるパスを通じて送信されるが、第1のデータブロックの他の符号化単位は第2のパスを通じて送信される、などである。送信機は、受信機による符号化単位の受信の確認応答を検出することができる。受信機に

おいて第1のデータブロックを復元するのに十分な、第1のデータブロックに対するすでに送信された符号化単位からの第1のデータブロックの符号化単位を、受信機が受信するであろう確率が、送信機において検出されることが可能であり、この確率は、所定の試験が満たされるどうかを判定するために、閾値の確率に対して試験され得る。試験のステップの後、かつ受信機における第1のデータブロックの復元の確認を送信機が受信する前に、所定の試験が満たされると、送信機は、複数のパスを通じて送信機から第2のデータブロックの符号化単位を送信することができる。いくつかの例では、送信機は、受信機が第1のブロックを復元するのに十分な数の符号化単位が第1のブロックについて送信される前に、第2のブロックの符号化単位を送信し得るので、送信機は、第2のブロックに対する少なくともいくつかの符号化単位を送信した後に、第1のブロックに対する追加の符号化単位を送信することができる。さらに、送信機は、ブロックが完全に形成される前に、ブロックの符号化単位を受信機に送信することができる。いくつかの例では、所定の試験は、閾値の確率に対する確率の比較であり、確率が閾値の確率より大きいとき、所定の試験は満たされると判定され得る。

【００１０】

いくつかの例では、送信機はまた、各ブロックのサイズまたは期間を、それが生成されるときに動的に決定し、ブロック中のデータが生成されるレートを決定し、複数のパスの各々を通じて符号化単位が受信機に送信されるレートを決定することができる。

【００１１】

一例では、コンピュータ可読記憶媒体は命令を記憶しており、この命令は、実行されると、クライアントデバイスのプロセッサに、複数の並列ネットワークパスを介して前方誤り訂正されたデータをサーバデバイスから受信させ、ネットワークパスの各々でのデータのロスを決定させ、ネットワークパスの各々でのデータのロスを表すデータをサーバデバイスへ送信させる。

【００１２】

別の例では、コンピュータ可読記憶媒体は命令を記憶しており、この命令は、実行されると、サーバデバイスのプロセッサに、複数の並列ネットワークパスを介して前方誤り訂正されたデータをクライアントデバイスへ送信させ、ネットワークパスの各々を通じて送信されるデータのロスを表すデータをクライアントデバイスから受信させ、ロスを表すデータに基づいて、並列ネットワークパスを通じた後続のデータ送信のために送信される前方誤り訂正データの量を修正させる。

【００１３】

1つまたは複数の例の詳細が、以下の添付の図面および説明において述べられる。他の特徴、目的、および利点は、説明、図面、および特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】

【００１４】

【図１】本開示の教示を使用し得る例示的なネットワーク、送信機エンドシステム、および受信機エンドシステムのブロック図である。

【図２】モジュール式の信頼性のある転送プロトコルアーキテクチャと、そのようなプロトコルを使用して動作するための関連するシステムとを示す図である。

【図３】送信機のFECベースの信頼性制御プロトコルアーキテクチャと、そのようなプロトコルを使用して動作するための関連するシステムとの例を示す図である。

【図４】受信機のFECベースの信頼性制御プロトコルアーキテクチャと、そのようなプロトコルを使用して動作するための関連するシステムとの例を示す図である。

【図５】TF信頼性制御プロトコルを実装するシステムによって使用され得るフォーマットの1つの可能なセットを示す図である。

【図６】送信機のTF信頼性制御プロトコルを実装するシステムの論理を示すフローチャートである。

【図７】受信機のTF信頼性制御プロトコルを実装するシステムの論理を示すフローチャー

トである。
【図８】アクティブブロックを示す図である。
【図９】インターリーブされた信頼性制御プロトコルによって使用され得るフォーマットの可能なセットを示す図である。
【図１０】基本送信機のインターリーブされた信頼性制御プロトコルを実装するシステムの論理の説明のための例の図である。
【図１１】基本受信機のインターリーブされた信頼性制御プロトコルを実装するシステムの論理の説明のための例の図である。
【図１２】マルチパスストリーミングシステムのブロック図である。
【図１３】マルチパス信頼性制御プロトコルを実装するマルチパスストリーミングシステムによって使用され得るデータフォーマットの1つの可能なセットを示す図である。
【図１４】マルチパスストリーミング送信機のブロック図である。
【図１５】マルチパスストリーミング受信機のブロック図である。
【図１６】マルチパスのFECベースの信頼性転送制御の方法の動作の間のソースブロックの様々な分類のスナップショットを示す図である。
【発明を実施するための形態】
【００１５】
TCPを使用するとき、データ転送のスループット(パケット毎秒単位の)は、RTT(秒単位の)と、エンドツーエンド接続上での損失率の平方根との積に反比例することを、多くの研究者による研究は示している。たとえば、米国と欧州との間の典型的なエンドツーエンドの地上接続は、200ミリ秒のRTTと2%の平均パケットロスとを有し、IPパケットのサイズは通常10キロビット前後である。これらの条件のもとでは、どれだけ多くの帯域幅がエンドツーエンドで利用可能であったとしても、TCP接続のスループットは最大でも300～400キロビット毎秒(kbps)前後である。高いRTTに加えて様々な大気による影響で情報が失われる衛星リンクでは、状況はより厳しい。
【００１６】
別の例として、モバイルネットワーク、3G、またはLTEネットワーク中のモバイルデバイスは、大きなRTT、RTTの大きな変動、および利用可能な帯域幅の大きな変動を体験することが知られている。モバイルデバイスは、セル内でのモバイルデバイスの変化する位置、またはカバレッジの変動を引き起こし得るセル境界をまたがる移動、モバイルデバイスにカバレッジを提供しているセルに他のモバイルデバイスが近付くことまたはそれから離れることによるネットワークの変動する負荷、および種々の他の理由を含む、種々の理由で、上記の体験を有し得る。これらのタイプの条件におけるTCPの低い性能の主な理由は、TCPによって使用されるレート制御プロトコルがこれらの条件ではうまく動作しないことである。たとえば、利用可能な帯域幅が短期間は高かったとしても、TCPプロトコルは、利用可能な帯域幅が再び減る前により高い利用可能な帯域幅を利用するために送信レートを上げるのに、十分高速に反応することができない。
【００１７】
TCPによって使用される信頼性制御プロトコルおよびレート制御プロトコルは分離不可能であるため、これは、TCPプロトコルが全体としてこれらの条件ではうまく動作しないことを示唆する。TCPのこれらの制約を克服しようとする1つの方法は、別々のパスを通じた複数のTCP接続を使用して、総合的なエンドツーエンドスループットをさらに上げることである。さらに、転送のための異なる適用例の要件はまちまちであるが、TCPはすべてのネットワーク条件における種々の適用例においてかなり普遍的に使用されているので、多くの状況において低い性能につながる。
【００１８】
たとえば、リアルタイムのビデオストリーミング適用例では、ビデオは、モバイルデバイス上のフィールドで生成され、場合によっては異なる3Gまたは4G/LTE接続を通じて、場合によっては生成側のモバイルデバイスにWiFiによってつながれた、または接続された複数のモバイルデバイスを使用して、元のビデオストリームを再構築することになる受信側

デバイスへ複数のTCP接続を通じてストリーミングされ得る。しかしながら、利用可能な帯域幅の変動およびRTTの変化が原因で、これらの複数のTCP接続はそれでも、利用可能な帯域幅を完全に利用できないことがある。そのようなストリーミングの適用例において、エンドツーエンドの遅延の要件はリアルタイムのストリーミングの適用例では極めて厳しいことがあるので、ストリームがデータのブロックのシーケンスからなるというさらなる複雑さおよび要件があり、受信側デバイスにおけるストリームのデータのブロックの再構築を可能にするのに、データの各ブロックに対する異なるTCP接続を通じて送信される十分な符号化単位が受信される必要があり、一般に、データのブロックは、データの各ブロックが送信機において(一部または全体が)利用可能にされるときと、データのブロックが再構築され受信側デバイスにおける再生または消費のために利用可能にされるときとの間の最小の可能な遅延を伴って、受信側デバイスにおいて順番に消費または再生されることになる。これらの要件により、TCPベースの解決法の能力は、最も遅いTCP接続により制約されるようになり得るので、全体としてのTCPベースの解決法は極めて劣ったものになり得る。

【００１９】

本開示は、独立に使用され得る改善された信頼性制御プロトコルとレート制御プロトコルの説明を含み、選択された実際のレート制御プロトコルがアプリケーションの要件およびアプリケーションが実行されるネットワーク条件に基づき得るように、同じ信頼性制御プロトコルが種々の異なるレート制御プロトコルとともに使用され得る。Micah Adler、Yair Bartal、John Byers、Michael Luby、Danny Razによる論文「A Modular Analysis of Network Transmission Protocols」、Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems、1997年6月(以後「Adler」と呼ばれ、参照のために本明細書に組み込まれる)は、信頼性のある転送プロトコルを独立した信頼性制御プロトコルおよびレート制御プロトコルに区分することを主張する、転送プロトコルを構築することに対するモジュール式の手法を紹介する。

【００２０】

任意の信頼性制御プロトコルに対して、その性能の2つの主要な尺度は、どれだけのバッファリングが必要とされるかということと、その「グッドプット」である。送信機と受信機の両方において、バッファリングが信頼性制御プロトコルに導入される。送信機におけるバッファリングは、たとえば、データが最初に送信された後で、データが受信機において受信されたという確認応答を送信機が有するまでに、データがバッファリングされるときに、発生する。受信機におけるバッファリングは同様の理由で発生する。バッファリングは、(1)送信機および受信機の信頼性制御プロトコルがどれだけのメモリを使用するかに直接影響を与える、(2)送信機および受信機の信頼性制御プロトコルがどれだけのレイテンシをもたらすかに直接影響を与えるという2つの理由で、関心の対象である。グッドプットは、転送されるべきデータのサイズを、転送の間に受信機エンドシステムにおいて受信された送信データの量で割ったものとして定義される。たとえば、元のデータを転送するためにパケットで送信されるデータの量が元のデータのサイズである場合、グッドプット=1.0であり、冗長なデータがまったく送信されなければグッドプット=1.0が達成され得る。

【００２１】

Adlerは、使用されるレート制御プロトコルから大部分が独立した信頼性制御プロトコルについて要約しており、この信頼性制御プロトコルは以下では「非符号信頼性制御プロトコル」と呼ばれる。非符号信頼性制御プロトコルは、元のデータがブロックに区分され各ブロックがパケットのペイロードにおいて送信され、次いで各ブロックの正確なコピーが信頼性のある転送を確実にするために受信される必要があるという点で、TCPで実装される信頼性制御プロトコルとある面では同様である。非符号信頼性制御プロトコルの問題は、グッドプットが最適(基本的には1に等しい)であっても、パケットロスがあるときには非符号信頼性制御プロトコルのもたらすバッファリングがかなり多くなり得るということである。Adlerは、プロトコルが最適なグッドプットを有し、送信機と受信機において

必要とされるバッファリングの量を最小にすることに関して、最適解から一定の係数の範囲内にあることが証明可能であるという点で、非符号信頼性制御プロトコルが、データを転送するためにコーディングを使用しない信頼性制御プロトコルの中で、最適解から一定の係数の範囲内にあることを証明している。

【００２２】

信頼性制御プロトコルで使用されてきた1つの技法は、リードソロモン符号、トルネード符号、または連鎖反応符号(これらは情報を追加する符号である)のような、前方誤り訂正(FEC)符号である。FEC符号を使用すると、元のデータは、パケットのペイロードよりも大きなブロックに区分され、次いで符号化単位が、これらのブロックから生成され、符号化単位をパケットで送信する。コーディングを使用しない信頼性制御プロトコルに対するこの手法の1つの基本的な利点は、フィードバックがはるかに簡単で頻度が低くなり得るということであり、すなわち、各ブロックに対して、受信機は、どの符号化単位が受信されたかの厳密なリストの代わりに、受信された符号化単位の量を送信機に示すだけでよい。さらに、元のデータブロックの長さよりも多くの符号化単位をまとめて生成し送信する能力は、信頼性制御プロトコルの設計において強力な道具である。

【００２３】

リードソロモン符号またはトルネード符号のような消失訂正符号は、一定の長さのブロックに対して一定の数の符号化単位を生成する。たとえば、B個の入力単位を含むブロックに対して、N個の符号化単位が生成され得る。これらのN個の符号化単位は、B個の元の入力単位およびN-B個の冗長な単位を含み得る。記憶容量が許せば、送信機は、各ブロックに対する符号化単位のセットを一度だけ計算し、カルーセルプロトコルを使用して符号化単位を送信することができる。

【００２４】

いくつかのFEC符号の1つの問題は、動作のために過剰な計算能力またはメモリを必要とすることである。別の問題は、必要な符号化単位の数がコーディング処理の前に決定されていなければならないということである。これは、パケットの損失率が過剰に見積もられていた場合、効率低下につながることがあり、パケットの損失率が過小に見積もられていた場合、障害につながることがある。

【００２５】

従来のFEC符号では、生成され得る可能な符号化単位の数は、ブロックが区分される入力単位の数と同じオーダーの大きさである。必ずではないが通常は、これらの符号化単位の大半またはすべてが、送信ステップの前の前処理ステップで生成される。これらの符号化単位は、元のブロックと長さが同じである、または元のブロックよりも長さがわずかに長い、符号化単位の任意のサブセットからすべての入力単位が再生成され得るという特性を有する。

【００２６】

米国特許第6,307,487号(以後「Luby I」と呼ばれ本明細書に参照によって組み込まれる)で説明される連鎖反応復号は、上記の問題に対処するある形式の前方誤り訂正を提供することができる。連鎖反応符号では、生成され得る可能な符号化単位の蓄積は、入力単位の数よりも大きな大きさのオーダーであり、候補の蓄積からランダムにまたは擬似ランダムに選択された符号化単位は、非常に迅速に生成され得る。連鎖反応符号では、符号化単位は、オンザフライで、送信ステップと同時の「必要に応じた」方式で生成され得る。コンテンツのすべての入力単位が、元のコンテンツよりもわずかに長さが長いランダムにまたは擬似ランダムに生成された符号化単位のセットのサブセットから再生成され得ることを、連鎖反応符号は可能にする。

【００２７】

米国特許第6,320,520号、6,373,406号、6,614,366号、6,411,223号、6,486,803号、および米国特許出願公開第2003/0058958号(以後「Shokorollahi I」と呼ばれる)のような他の文書は、様々な連鎖反応符号方式を説明しており、参照によって本明細書に組み込まれる。

【００２８】
連鎖反応符号を使用する送信機は、送信される各ブロックに対する符号化単位を継続的に生成することができる。符号化単位は、ユーザデータグラムプロトコル(UDP)ユニキャストを介して、または可能であればUDPマルチキャストを介して、受信者に送信され得る。各受信者は、パケットで受信された適切な数の符号化単位を復号して元のブロックを取得する、復号ユニットを備えると仮定される。
【００２９】
以後「TF信頼性制御プロトコル」と呼ばれる、既存の単純なFECベースの信頼性制御プロトコルは、データの所与のブロックを復元するのに十分な符号化単位が受信されたという確認応答を受信機から受信するまで、そのブロックに対する符号化単位を送信し、次いで、送信機は次のブロックに移る。
【００３０】
RTTを、送信機がパケットを送信したときから、パケットが到着したという確認応答を送信機が受信機から受信するまでにかかる秒数とし、Rを、パケット/秒単位の送信機の現在の送信レートとし、Bを、パケットの単位でのブロックのサイズとする。TF信頼性制御プロトコルを使用すると、ブロックを復元するために必要とされる最後のパケットの後に送信された、ブロックに対する符号化単位を含む有用ではないパケットの数は、N=R*RTTである。したがって、パケットの一部分f=N/(B+N)は無駄になるので、グッドプットは最大でも1-fである。たとえば、R=1000パケット/秒であり、RTT=1秒であり、B=3000パケットである場合、f=0.25であり、すなわち受信されたパケットの25%が無駄になる。したがって、この例のグッドプットは(1.0という最大の可能なグッドプットと比較して)不十分な0.75である。
【００３１】
この例では、単純なFECベースの信頼性制御プロトコルによってもたらされるレイテンシが少なくとも4秒であり(各ブロックが全体で4秒間送信される)、少なくとも1つのブロック、すなわち3000パケットのデータのバッファリングを必要とすることを、ブロックBのサイズとともにレートRが示唆することにも留意されたい。さらに、グッドプットを上げることはバッファリングを増やすことを必要とし、または逆に、バッファリングを減らすことはグッドプットを下げることを必要とする。
【００３２】
Luby他による米国特許第7,447,235号(以後「Luby II」と呼ばれる)は、改善されたFECベースの信頼性プロトコルを説明する。しかしながら、複数のパスを通じたストリーミングのための改善された信頼性制御プロトコルが望ましい。さらに、信頼性およびグッドプットを最大にしエンドツーエンドのレイテンシを最小にする、複数のパスを通じたストリーミングに適した転送プロトコルをもたらすために、改善された信頼性制御プロトコルと組み合わされ得る、対応するレート制御プロトコルを提供することが望ましい。
【００３３】
本開示による例では、マルチパス送信のためのインターリーブされた信頼性制御プロトコルが、TCP、TF信頼性制御プロトコルおよび非符号信頼性制御プロトコル、ならびにLuby IIで説明されたFECベースの信頼性制御プロトコルに対する改善を提供するために使用され得る。さらに、信頼性およびグッドプットを最大にしエンドツーエンドのレイテンシを最小にする、複数のパスを通じたストリーミングに適した転送プロトコルを提供するために、改善された信頼性制御プロトコルと組み合わされ得る、改善されたレート制御プロトコルおよび生成レートプロトコルが導入される。
【００３４】
信頼性制御プロトコルによって、データのブロックが、一連の符号化単位として送信機から受信機に送信され、受信機は符号化単位またはブロックの復元を確認応答し、これによって、送信機が、受信機がデータを受信したかどうかを判定し、受信されていない場合、データを再送信し、または受信されたデータを復元するために使用可能な他のデータを送信することを可能にする。いくつかのインターリーブされた信頼性制御プロトコルの1

つの特性は、異なるブロックに対する符号化単位がインターリーブされた方式で送信されることである。インターリーブされた信頼性制御プロトコルは、実質的に任意のレート制御プロトコルと組み合わされると、エンドシステムにおけるバッファリング(および結果としてのレイテンシ)を最小にして転送のグッドプットを最大にする、効率的な信頼性のあるデータ転送を提供するという、特性を有する。

【００３５】

本開示の技法によれば、インターリーブされた信頼性制御プロトコルは、大きなロスおよび/または大きなRTTがあるときであっても、適切なレート制御プロトコルとともに使用されて、高いスループットを維持しながらデータの信頼性のある転送を確実にすることができる。たとえば、レート制御プロトコルは、一定のレートで送信することほど単純であってよく、インターリーブされた信頼性制御プロトコルは、その一定のレートと、到着に成功したパケットの割合とを乗じたものに等しいレートでデータが転送されることを保証しながら、転送の間のバッファリングおよびレイテンシを最小にする。

【００３６】

ここで導入されるインターリーブされた信頼性制御プロトコルによって提供される定量的な改善の例として、レート制御プロトコルがRパケット毎秒という一定のレートでパケットを送信するものであり、送信機と受信機との間の往復遅延時間がRTT秒であり、したがってN=R*RTTが、インフライトの確認応答されていないパケットの数であると仮定する。非符号信頼性制御プロトコルでは、送信機における全体のバッファサイズBは少なくともN*ln(N)であり、グッドプットは約1.0であり、必要とされるバッファリングの量とグッドプットとの間には可能な他のトレードオフ点がない。ここで、ln(x)はxの自然対数として定義される。TF信頼性制御プロトコルでは、送信機における全体のバッファサイズは少なくともBであり、グッドプットは約B/(B+N)であり、ここでBはパケット単位での選択されたブロックサイズであり、必要とされるバッファリングをグッドプットに対してトレードオフするように選択され得る。対照的に、Luby IIで説明されるようなインターリーブされた信頼性制御プロトコルでは、送信機における全体のバッファサイズは最大でもBであり、グッドプットは約N/(N+X)であり、ここでXは、必要とされるバッファリングをグッドプットに対してトレードオフするように選択された正の整数のパラメータであり、B=N*(1+ln((N/X)+1))が、パケット単位でのバッファサイズである。

【００３７】

ある例として、レートRが1000パケット/秒でありRTTが1秒である場合、N=1000パケットである。非符号信頼性制御プロトコルでは、送信機におけるバッファサイズは少なくとも7000パケットである。TF信頼性制御プロトコルでは、Bが4000パケットとなるように選択される場合、グッドプットは約0.80である。Xが50となるように選択される、Luby IIで説明されるインターリーブされた信頼性制御プロトコルでは、B=4000パケット(TF信頼性制御プロトコルの場合と同じ値)であり、グッドプットは0.95を超え、すなわち、最大でも受信されたパケットの5%しか無駄にならない。したがって、この例では、インターリーブされた信頼性制御プロトコルは、ほとんど同じ最適なグッドプットを伴いながら、非符号信頼性制御プロトコルよりもはるかに少ないバッファリングしか必要とせず、同じ量のバッファリングに対するTF信頼性制御プロトコルのグッドプットをはるかに超え、すなわち、インターリーブされた信頼性制御プロトコルでは無駄にされる送信は最大でも5%であるのに対して、TF信頼性制御プロトコルでは25%である。

【００３８】

実質的に任意のレート制御プロトコルが、インターリーブされた信頼性制御プロトコルとともに使用されて、信頼性のある転送プロトコルを提供し、たとえば、一定のレートで送信し、TCPと同様のウィンドウベースの混雑制御を使用し、TCPフレンドリーレート制御(TFRC)のような等式ベースの混雑制御プロトコルを使用し、または、本明細書で紹介されるレート制御プロトコルを含む実質的に任意の他のレート制御プロトコルを使用することができる。

【００３９】

以下の後続する説明は、送信機から受信機へ複数の独立のパスを通じてデータを送信するのに適切であり得るレート制御プロトコルを説明し、さらには、これらのレート制御プロトコルと組み合わされ得る改善されたインターリーブされた信頼性制御プロトコルを説明する。データが送信のためにどれだけ高速に生成されるべきかを決定するために使用され得る、生成レートプロトコルも説明される。最後に、この説明は、これらのプロトコルを組み合わせて、信頼性およびグッドプットを最大にしてエンドツーエンドのレイテンシを最小にする、複数の独立したパスを通じて送信機から受信機へデータをストリーミングするのに適した転送プロトコル全体を提供する方法の例を説明する。

【００４０】

この説明では、信頼性のある転送プロトコルは、送信されるパケットの一部が受信されない可能性がある場合であっても、すべてのデータが転送されるような方法で、パケットベースのネットワークを通じて送信機エンドシステムから受信機エンドシステムへデータを信頼性をもって転送するプロトコルである。

【００４１】

図1は、信頼性のある転送プロトコルがその上で動作し得る、ネットワーク130、送信機エンドシステムのセット100(1)～100(J)、および受信機エンドシステム160(1)～160(K)のセットの例を示す概念図である。送信機エンドシステム100はまた、サーバデバイスと表現されることがあり、受信機エンドシステム160は、クライアントデバイスと表現されることがある。通常、そのようなプロトコルはまた、パケット送信レートを調整するためのいくつかの機構を含み、ここでこの送信レートは、プロトコルが内蔵されるアプリケーション、ユーザ入力パラメータ、および送信機エンドシステムと受信機エンドシステムとの間のネットワーク条件を含む、種々の要因に依存し得る。

【００４２】

TCPのような信頼性のある転送プロトコルは、通常はいくつかのステップを伴う。これらのステップは、エンドシステムが、データの利用可能性を告知し、他のエンドシステムに対するデータの転送を開始し、どのデータが転送されるべきかを伝え、データの信頼性のある転送を実行するための方法を含む。エンドシステムが、利用可能性を告知し、転送を開始し、何が転送されるべきかを伝えるための種々の標準的な方法、たとえば、セッション告知プロトコル、セッション開始プロトコルなどが存在する。これらのステップはよく知られており、ここで詳細に説明される必要はない。

【００４３】

パケットデータの信頼性のある転送は、転送中の各点において、どのデータをパケットで送信すべきか、およびどのようなレートでパケットを送信すべきかを決めることを含む。各時点において行われるこの決定は、受信機エンドシステムから送信されるフィードバックおよび他の要因に依存し得る。通常は、データはデータのストリームとして送信機エンドシステムにおいて提示され、信頼性のある転送プロトコルは、このストリームを、それが送信されたのと同じ順序で、受信機エンドシステムへ信頼性をもって配信することが意図される。しばしば、転送が開始される前にストリームの全体の長さが知られていないということがある。

【００４４】

本開示は、信頼性のある転送プロトコルのためのモジュール式のアーキテクチャの例を説明する。Adlerは、任意の信頼性のある転送プロトコルがどのように信頼性制御プロトコルとレート制御プロトコルの組合せとして考えられ得るかを説明している。信頼性制御プロトコルは、転送の間にどのデータを各パケットに配置すべきかを決める、転送プロトコル全体のうちの部分である。レート制御プロトコルは、各データパケットをいつ送信するかを決める。多くの転送プロトコルでは、信頼性制御プロトコルおよびレート制御プロトコルは、動作において分離不可能に絡まり合っており、すなわちTCPについてこれが当てはまる。しかしながら、このことは、そのような絡まり合ったプロトコルが信頼性制御プロトコルおよびレート制御プロトコルへと概念的に区分され得るとしても、当てはまる。

【００４５】
Adlerは、信頼性制御プロトコルおよびレート制御プロトコルを独立に設計することによる、信頼性のある転送プロトコルの設計を主張する。そのような手法の利点は、同じ信頼性制御プロトコルが種々のレート制御プロトコルとともに使用され得るので、同じ信頼性制御プロトコルが、信頼性のある転送プロトコル全体が使用される適用例およびネットワーク条件に対して適切であるレート制御プロトコルとともに使用され得ることである。設計に対するこのモジュール式の手法は非常に有利であることがあり、それは、同じ信頼性制御プロトコルが、異なる適用例およびネットワーク環境においてレート制御プロトコルの多様なセットとともに使用されることが可能であり、したがって、各適用例およびネットワーク環境に対する信頼性のある転送プロトコル全体の完全な再設計を回避するからである。たとえば、TCPは、異なるネットワーク環境における種々の適用例のために使用され、TCPは、レート制御プロトコルによって決定されるような、TCPの実現する低いスループットが原因で、これらの適用例およびネットワーク環境の一部に対して、不十分に動作する。残念ながら、信頼性制御プロトコルおよびレート制御プロトコルはTCPアーキテクチャにおいて非常に絡み合っているので、TCPが不十分に動作する状況においてスループット性能を改善するために、TCP内で異なるレート制御プロトコルを単に使用することは可能ではない。

【００４６】
図1の例では、送信機エンドシステム100の1つが、本開示の技法を使用して、ネットワーク130を介してデータを受信機エンドシステム160の1つに転送することができる。たとえば、送信機エンドシステム100(1)は、本開示の技法の1つまたは複数を使用して、メディアデータを受信機エンドシステム160(1)に送信することができる。一例では、送信機エンドシステム100(1)は、ネットワーク130を通る複数の並列パスを通じて、前方誤り訂正されたメディアデータを受信機エンドシステム160(1)に送信することができる。そのような前方誤り訂正されたデータは複数のブロックを含んでよく、複数のブロックの各々は複数の符号化単位を使用してFEC符号化され得る。したがって、各ブロックに対して、送信機エンドシステム100(1)は、複数の符号化単位を受信機エンドシステム160(1)に送信することができる。

【００４７】
受信機エンドシステム160(1)は、並列ネットワークパスの各々でのロスを表すデータを送信することができる。そのようなデータの一例は、図9および図13を参照して以下でより詳細に説明される。一般に、データは、各ブロックに対して、受信された符号化単位の数および受信された最高のシーケンス番号を示し得る。加えて、または代替的に、データは、様々な並列ネットワークパスを通る各パケットフローに対して、受信された最大のシーケンス番号を示し得る。同様に、送信機エンドシステム100(1)は、ロスに関する、受信機エンドシステム160(1)によって送信されるデータを受信することができる。送信機エンドシステム100(1)は、ロスに関するこのデータを使用して、後続の符号化単位がどのように形成されるかを修正することができ、たとえば、ロスが予想よりも多かった場合に追加のFECデータを含めること、または、ロスが予想より少なかった場合にFECデータの量を減らすことができる。符号化単位に含まれるFECデータのこの修正は、並列ネットワークパスでのロスの総計に基づき得る。

【００４８】
加えて、または代替的に、送信機エンドシステム100(1)は、ブロックが完全に形成される前に、ブロックの符号化単位を送信することができる。たとえば、現在のブロックのサイズがK個の符号化単位であると仮定すると、送信機エンドシステム100(1)は、現在のブロックに対するすべてのK個の符号化単位が送信機100(1)に対して利用可能にされる前に、現在のブロックに対する1つまたは複数の符号化単位を送信することができる。また、受信機エンドシステム160(1)は、現在のブロックに対するK個の符号化単位を受信する前に、後続のブロックの1つまたは複数の符号化単位を受信することができる。

【００４９】

図2は、Adlerにおいて主張される、例示的なモジュール式の信頼性のある転送プロトコルアーキテクチャを示すブロック図である。送信機の転送プロトコル210は、送信機の信頼性制御プロトコル220および送信機のレート制御プロトコル230に区分される。送信機の信頼性制御プロトコル220は、各データパケットで何が送信されるかを決定し、送信機のレート制御プロトコル230は、各データパケットがいつ送信されるかを決定する。送信機の信頼性制御プロトコル220は、受信機の転送プロトコル290内の受信機の信頼性制御プロトコル280によって使用され得る各データパケットに、追加の信頼性制御情報を置くことができる。

【００５０】

送信機の信頼性制御プロトコル220はまた、各データパケットで何が送信されるかを決定するのを助けるために使用される信頼性制御情報250を、受信機の転送プロトコル290内の対応する受信機の信頼性制御プロトコル280から受信することができる。同様に、送信機のレート制御プロトコル230は、受信機の転送プロトコル290内の受信機のレート制御プロトコル270によって使用され得る各データパケットに、追加のレート制御情報を置くことができる。送信機のレート制御プロトコル230はまた、各データパケットがいつ送信されるかを決定するのを助けるために使用されるレート制御情報250を、受信機の転送プロトコル290内の対応する受信機のレート制御プロトコル270から受信することができる。

【００５１】

送信機の信頼性制御プロトコル220と受信機の信頼性制御プロトコル280との間で通信される信頼性制御情報は、パケットロスのような種々の要因に依存することがあり、いくつかの詳細において後で説明されるように種々の情報を含むことがある。同様に、送信機のレート制御プロトコル230と受信機のレート制御プロトコル270との間で通信されるレート制御情報は、パケットロスおよび測定された往復遅延時間(RTT)のような種々の要因に依存し得る。さらに、データパケット240またはフィードバックパケット250で送信される情報が、信頼性制御とレート制御の両方のために使用され得るという点で、信頼性制御情報およびレート制御情報は重複することがある。一般に、送信機の転送プロトコル210から受信機の転送プロトコル290に送信される信頼性制御情報およびレート制御情報は、データパケット240中でデータとともに送信されてよく、または別個の制御パケット240で送信されてよく、またはその両方であってよい。これらのプロトコルは、送信機から受信機に、または受信機から送信機に送信される必要のある制御情報の量を最小にするように設計されるべきである。

【００５２】

多くの適用例では、データはストリームとして転送されるべきであり、すなわち、データが送信機エンドシステムに到着するに従って、データは、それが送信機エンドシステムに到着したのと同じ順序で、受信機エンドシステムへと可能な限り迅速に信頼性をもって転送されるべきである。いくつかの適用例では、たとえばストリーミングの適用例では、または、2つのエンドシステムの間でデータの小さなバーストが可能な限り迅速に双方向に送信されるべきである対話型の適用例では、転送プロトコル全体によりもたらされるレイテンシが最小にされるべきである。したがって、転送プロトコルによってもたらされる全体のレイテンシは最小にされるべきである。

【００５３】

送信機の信頼性制御プロトコル220および受信機の信頼性制御プロトコル280は通常、両方とも、データを一時的に記憶するためのバッファを必要とする。一般に、送信機の信頼性制御プロトコル220においてバッファリングされるデータは、送信機の信頼性制御プロトコル220が受信機の信頼性制御プロトコル280から復元の確認応答をまだ受信していないストリーム中の少なくとも最早のデータから、送信機の信頼性制御プロトコル220がデータパケットで送信を開始したストリーム中の最遅のデータまでを含む。受信機の信頼性制御プロトコル280におけるバッファのサイズは一般に、少なくとも、まだ復元されていない最早のデータからデータパケットが受信された最遅のデータまでの、ストリーム中のデータの量である。

【００５４】
送信機の信頼性制御プロトコル220のバッファリング要件は、どれだけの一時的な記憶空間が送信機の信頼性制御プロトコル220によって必要とされるか、および、どれだけのレイテンシを送信機の信頼性制御プロトコル220が信頼性のあるデータ転送全体にもたらすかに、直接の影響を有する。受信機の信頼性制御プロトコル280のバッファリング要件は、同様の影響を有する。したがって、送信機の信頼性制御プロトコル220と受信機の信頼性制御プロトコル280の両方のバッファリング要件を最小にすることが重要である。
【００５５】
信頼性制御プロトコルは、各データパケットで何が送信されるかを決定する。エンドシステム間の接続を効率的に利用するために、受信機の信頼性制御プロトコル280において受信されるどのようなデータパケットも、元のデータストリームの部分を復元するのに有用であることを確実にするために、送信機の信頼性制御プロトコル220が可能な限り少量の冗長なデータをパケット中で送信することが重要である。信頼性制御プロトコルのグッドプットは、データの元のストリームの長さを、データの元のストリームの復元の間に受信機の信頼性制御プロトコル280によって受信されるデータパケットの全体の長さで割ったものとして定義される。グッドプットの目標は、信頼性制御プロトコルが1.0またはそれに近いグッドプットをもたらすことであり、この場合、データの元のストリームを復元するために最小限の量のデータが受信される。いくつかの信頼性制御プロトコルでは、グッドプットは1.0より小さいことがあり、この場合、送信されたデータパケットの一部が無駄になる。したがって、送信機エンドシステムから受信機エンドシステムへと移動するデータパケットによって消費される帯域幅を効率的に使用するために、グッドプットが可能な限り1.0に近くなるように信頼性制御プロトコルを設計することが重要である。
【００５６】
信頼性制御プロトコルにおいて使用されている1つの解決法は、リードソロモン符号もしくはトルネード符号のような前方誤り訂正(FEC)符号、またはLuby IもしくはShokrollahi Iで説明されるような連鎖反応符号(これは情報を追加する符号である)の解決法である。元のデータは、パケットのペイロードよりも大きなブロックへと区分され、次いで符号化単位が、これらのブロックから生成され、パケットで符号化単位を送信する。リードソロモン符号またはトルネード符号のような消失訂正符号は、一定の長さのブロックに対して一定の数の符号化単位を生成する。たとえば、入力単位を含むブロックに対して、N個の符号化単位が生成され得る。これらのN個の符号化単位は、B個の元の入力単位およびN-B個の冗長な単位を含み得る。
【００５７】
FECベースの信頼性制御プロトコルは、FEC符号を使用する信頼性制御プロトコルである。図3は、送信機のFECベースの信頼性制御プロトコル220の例を示すブロック図であり、図4は、受信機のFECベースの信頼性制御プロトコル280の例を示すブロック図である。送信機の信頼性制御論理310は、データの元のストリームをデータブロック330に区分し、次いで、各ブロックに対する符号化単位を生成するようにFECエンコーダ320に指示する。送信機の信頼性制御論理310は、送信機のレート制御プロトコル230を扱うデバイスへ符号化単位および信頼性制御情報340がどのように渡されるかを決定し、それはまた、図4に示される受信機のFECベースの信頼性制御論理410によって送信される信頼性制御情報350を扱う。
【００５８】
送信機の信頼性制御論理310は、各ブロックが復元されることを確実にするのに十分な符号化単位が、図4に示される受信機のFECベースの信頼性制御プロトコル280によって受信されることを確実にしなければならない。すべてのブロックは基本的に同じ長さであってよく、またはブロックの長さは、データのストリームが送信機に利用可能にされるレート、データパケットの送信レート、ネットワーク条件、適用例の要件、およびユーザの要件を含む、種々のパラメータに従って、転送の間に大きく変化し得る。
【００５９】

データの所与のブロックの長さがB個の符号化単位であると仮定する。いくつかのFEC符号については、データの元のブロックを復元するために必要とされる符号化単位の数はちょうどBであるが、他のFEC符号については、データの元のブロックを復元するために必要とされる符号化単位の数はBよりわずかに多い。FECベースの信頼性制御プロトコルの説明を簡単にするために、データブロックの復元にはB個の符号化単位で十分であると仮定され、ここで、ブロックを復号するためにB個よりも多くの符号化単位を必要とするFECコードは、わずかに小さなグッドプットおよびわずかに厳しいバッファリングの要件とともに使用され得ることを理解されたい。

【００６０】

図4の受信機の信頼性制御論理410は、データブロックを復号するためにB個の符号化単位が受信されることを確実にすることを担い、次いでFECデコーダ420がデータブロック430を復元するために使用される。受信機の信頼性制御論理410は、符号化単位および送信機のFECベースの信頼性制御プロトコル220から送信された信頼性制御プロトコル340を受信することと、送信機の信頼性制御論理310へ最終的に送信されそれによって処理される信頼性制御情報350を生成し送信することとを担う。

【００６１】

TF信頼性制御プロトコルは、データのストリームを全般に等しいサイズのブロックに区分する。全体のアーキテクチャは、任意の時点において1つのアクティブなデータブロックがあり、ブロックを再構築するのに十分な符号化単位が到着したことを示すメッセージを受信機から送信機が受信するまで、送信機がデータブロックに対する符号化単位を生成し送信し、上記のメッセージを受信した時点で送信機が次のブロックに移るというものである。したがって、後続のブロックに対する任意の符号化単位が生成され送信される前に、所与のブロックに対するすべての符号化単位が生成され送信され、ブロックが復元される。

【００６２】

図5は、TF信頼性制御プロトコルによって使用され得るフォーマットの1つの可能なセットを示すブロック図である。この例での送信機データフォーマットは、送信機のTF信頼性制御プロトコルが符号化単位および対応する信頼性制御情報を受信機のTF信頼性制御プロトコルに送信するフォーマットを表す。これは、どのブロックから符号化単位が生成されるかを示すブロック番号510、符号化単位がブロックからどのように生成されるかを示す符号化単位ID 520、および、ブロックを復元するために受信機のTF信頼性制御プロトコル内でFECデコーダによって使用され得る符号化単位530を含む。受信機フィードバックフォーマットは、受信機のTF信頼性制御プロトコルが信頼性制御情報を送信機のTF信頼性制御プロトコルに送信するフォーマットを表す。これは、受信機のTF信頼性制御プロトコルがブロックを復元するために符号化単位を受信している現在のブロックのブロック番号であるブロック番号540と、受信機のTF信頼性制御プロトコルがブロックを復元するために必要とする追加の符号化単位の数である必要とされる符号化単位550とを含む。

【００６３】

図6は、送信機のTF信頼性制御プロトコルを実装するための処理の例を示すフローチャートである。この例示的な処理によれば、送信機デバイス(図1の送信機エンドシステム100の1つのような)は、対応する送信機のレート制御プロトコルによって決定される、送信機データを送信する時間であるかどうかを確かめるために、継続的に確認する(ステップ610)。送信機データを送信する時間である場合、アクティブブロックから符号化単位が生成され、送信機データが送信される(620)。送信機データの形式の例が図5に示されるフォーマットである。この処理はまた、受信機フィードバックが受信されたかどうかを確かめるために、継続的に確認する(630)。受信機フィードバックデータの形式の例が図5に示されるフォーマットである。受信機フィードバックがある場合、それは、アクティブブロックを復元するためにどれだけの追加の符号化単位を受信機が必要とするかについての情報を更新するために処理される。送信機デバイスは次いで、必要とされる符号化単位の数が0であるかどうかを確かめるために確認し(640)、0である場合、次いで、データのストリ

ーム中の次のブロックが利用可能であるかどうかを確かめる(650)。利用可能ではない場合、送信機デバイスは、準備が整うまで次のブロックに備え(660)、次いで、現在のアクティブブロックを非アクティブ化し、次のブロックをアクティブ化することに続く(670)。一般に、次のブロックは、現在のアクティブブロックが送信されている間に準備されていてよい。

【００６４】

本明細書で説明されるプロトコルの各々は、デバイスまたは適切なプロセッサによって実行されるソフトウェアもしくはファームウェアによって実装され得ることを理解されたい。たとえば、ルータおよびホストコンピュータのようなネットワークデバイスを使用して実装が行われてよく、さらには、ワイヤレス送信機、再送信機、および他のワイヤレスデバイス上で実装されてよい。本明細書で説明されるプロトコルは、ソフトウェアで実装されてよく、そのようなプロトコルを実装するように構成される方法および/または装置を有する。

【００６５】

図7は、受信機のTF信頼性制御プロトコルを実装するための例示的な処理を示すフローチャートである。受信機のTF信頼性制御プロトコルによれば、受信機エンドシステム160(図1)の1つのような受信機デバイスは、図5に示される送信機データフォーマット中にある、送信機データが受信されたかどうかを確かめるために継続的に確認することができる(710)。受信されている場合、送信機データ内の符号化単位がアクティブブロックからのものかどうかが確認される(720)。符号化単位がアクティブブロックからのものではない場合、それは廃棄され(760)、したがって、符号化単位は、いずれのブロックを復元するにも有用ではないので、無駄になる送信機データである。

【００６６】

符号化単位がアクティブブロックからのものである場合、それは、アクティブブロックに対してすでに受信されている符号化単位のセットに追加され、ブロックに対する符号化単位の必要な数は1だけデクリメントされる(730)。受信機デバイスは次いで、符号化単位の必要な数が0かどうかを確かめるために確認し(740)、0である場合、次いで受信機デバイスは、FECデコーダを使用してアクティブブロックを復元し、次のアクティブブロックに対する符号化単位の受信に備える(750)。受信機のTF信頼性制御プロトコルはまた、対応する受信機のレート制御プロトコルによって決定される、受信機フィードバックを送信する時間かどうかを確かめるために、継続的に確認する(770)。その時間である場合、次いで受信機フィードバックが準備されて送信され(780)、そのフィードバックは、図5に示される受信機フィードバックフォーマットのフォーマットである。

【００６７】

これは、TF信頼性制御プロトコル全体の部分的な説明であることに留意されたい。たとえば、これは、受信機フィードバックが受信機のTF信頼性制御プロトコルによって送信される条件を規定しない。これは、受信された送信機データの受信によって、時々鳴動するタイマーによって、またはこれらのイベントの任意の組合せ、もしくは受信機のレート制御プロトコルによって決定されるような任意の他のイベントによって引き起こされ得る。一般に、受信機フィードバックは、受信機のTF信頼性制御プロトコルにおける符号化単位の受信の進行について送信機のTF信頼性制御プロトコルが定期的に知らされ続けるのに十分頻繁に、しかし、送信機のTF信頼性制御プロトコルから受信機のTF信頼性制御プロトコルに送信される符号化単位を含む送信機データとほぼ同じ帯域幅を消費するほど頻繁にではなく、送信される。

【００６８】

TF信頼性制御プロトコルは、次の意味で「無駄が多い」と考えられ得ることに留意されたい。Bを符号化単位の単位での各データブロックのサイズとし、Rをパケットがレート制御プロトコルによって送信される際のレートとし、RTTを送信機エンドシステムと受信機エンドシステムとの間の往復遅延時間とし、N=R*RTTとする。送信機と受信機の間でのパケットロスはないと仮定する。そうすると、送信機のTF信頼性制御プロトコルがアクティ

ブブロックに対してB個の符号化単位(これはブロックを復元するのに十分である)を送信した後、送信機のTF信頼性制御プロトコルは、ブロックを復元するのに十分な符号化単位が到着したことを示す受信機フィードバックを受信機のTF信頼性制御プロトコルから受信するまで、N個の追加の符号化単位を送信し続け、これらのN個の符号化単位のすべてが無駄になる。長さBのブロックを復元することは、B+N個の符号化単位を送信することを必要とするので、グッドプットはB/(B+N)である。

【００６９】

BがNと比較して相対的に小さい場合、グッドプットは最適からは程遠く、送信機と受信機との間で使用される帯域幅の多くが無駄になる。一方、BがNと比較して大きい場合、送信機のTF信頼性制御プロトコルと受信機のTF信頼性制御プロトコルにおけるバッファのサイズは大きいことがあり、このことは、受信機におけるデータストリームの配信のレイテンシが大きいことも示唆する。ある例として、符号化単位のサイズが1キロバイトであり、レートRが1000符号化単位毎秒=1メガバイト毎秒=8メガビット毎秒であり、RTTが1秒であると仮定する。すると、N=R*RTT=1メガバイトである。ブロックのサイズがB=3メガバイトに設定される場合、グッドプットは約(B/(B+N))=0.75にすぎず、すなわち、送信される符号化単位の約25%が無駄になる。送信される符号化単位の約2%しか無駄にならないように、グッドプットをたとえば0.98に上げるためには、B=49メガバイトという非常に大きなバッファサイズが必要である。すると、このサイズのバッファは、少なくとも50秒という、信頼性制御プロトコルにより追加されるレイテンシをもたらす。

【００７０】

上で説明されたTF信頼性制御プロトコルには多くの可能な変形形態がある。たとえば、送信機のTF信頼性制御プロトコルは、B個の符号化単位がブロックから送信された後で符号化単位の送信を停止し、ブロックを復元するのに十分な符号化単位が受信されたかどうかを示すための受信機フィードバックを受信するのを待機することができる。ロスがなければ、この変形形態は無駄になる符号化単位をまったく送信しないが、この場合であっても、各ブロックの間にはRTT時間という間隙があり、帯域幅が他の目的で使用されていない場合、このプロトコルは依然としてR*RTTという量の無駄になる帯域幅をもたらす。さらに、全体の配信時間は、理想よりもB/(B+N)倍遅くなる。ロスがある場合、この変形形態はさらなるレイテンシおよび配信の速度低下をもたらし、それは、失われた符号化単位の代わりに、ブロックを復元するために追加の符号化単位が最終的には送信されなければならないからである。

【００７１】

TF信頼性制御プロトコルは、非符号信頼性制御プロトコルに対する利点を有し、それは、任意の失われた符号化単位が、受信機フィードバックを必要とせずに、同じブロックから生成される任意の後で受信される符号化単位により埋め合わされ得るからである。TF信頼性制御プロトコルに無駄が多い主な理由は、次のブロックの転送が開始する前に各ブロックの転送が完了するという意味で、プロトコルの順次的な性質によるものである。本明細書で説明される改善された信頼性制御プロトコルは、インテリジェントな方式でブロックの処理をインターリーブするために使用され得る。

【００７２】

インターリーブの説明のための例が図8に示される。この例では、第1のアクティブブロックAB 1(810)および第2のアクティブブロックAB 2(820)という、2つのアクティブブロックがある。図8の下側部分は、経時的なデータパケットの送信のパターンの例を示し、各パケットは、対応するパケットがAB 1に対する符号化単位を含むかAB 2に対する符号化単位を含むかに応じて、AB 1とAB 2のいずれかとして標識される。この例では、AB 1(830(1)、830(2)、830(3)、および830(4))に対する符号化単位を含む4個のパケットがまず送信され、次いで、AB 2(830(5)および830(6))に対する符号化単位を含む2個のパケットが送信され、AB 1(830(7))に対する符号化単位を含む1個のパケット、AB 2(830(8))に対する符号化単位を含む1個のパケット、およびAB 1(830(9))に対する符号化単位を含む1個のパケットが続く。一般に、異なるブロックに対する符号化単位の間のインターリーブは、グ

ッドプットを最大にして、全体のバッファリング要件(および結果としてもたらされるレイテンシ)を最小にするように設計されるべきである。

【００７３】

送信機エンドシステム100(図1)の1つのような送信機デバイスは、図8に示されるように、受信機エンドシステム160(図1)の1つのような受信機デバイスに、インターリーブされたデータを送信することができる。このようにして、図8は、第1のブロックに対する符号化単位の第1のセットを送信するステップであって、符号化単位の第1のセットが第1のブロックを復元するために必要とされる最小の数よりも少ない符号化単位を含む、ステップと、符号化単位の第1のセットを送信するステップの後で、第2のブロックに対する符号化単位の第2のセットを送信するステップであって、第2のブロックがデータのストリーム中で第1のブロックの後にある、ステップと、符号化単位の第2のセットを送信するステップの後で、第1のブロックに対する1つまたは複数の符号化単位を含む符号化単位の第3のセットを送信するステップとを含む、方法の例を示す。同様に、図8はまた、第1のブロックに対する符号化単位の第1のセットを受信するステップであって、符号化単位の第1のセットが第1のブロックを復元するために必要とされる最小の数よりも少ない符号化単位を含む、ステップと、符号化単位の第1のセットを受信するステップの後で、第2のブロックに対する符号化単位の第2のセットを受信するステップと、符号化単位の第2のセットを受信するステップの後で、第1のブロックに対する1つまたは複数の符号化単位を含む符号化単位の第3のセットを受信するステップとを含む、方法の例を示す。

【００７４】

図8は、あるブロックが完全に形成される前にそのブロックに対する符号化単位が送信される例を描く。たとえば、AB 1が完全に形成される前にアクティブブロックAB 1に対する符号化単位が送信されることがあり、すなわち、AB 1に対する第1の符号化単位830(1)、830(2)が送信機によって送信されるときに、送信機においてはまたAB 1のすべてが利用可能にはなっておらず、それにもかかわらず、AB 1が完全に形成される前にAB 1の符号化単位が送信されることがある。加えて、または代替的に、AB 2は完全には形成されていないと考えられることがあり、それにもかかわらず、AB 2が完全に形成される前にAB 2の符号化単位が送信されることがある。たとえば、AB 2のすべてが送信機において完全に利用可能になる前に、AB 2に対する符号化単位830(4)、830(5)が送信されることがある。同様に、図8は、インターリーブされた方式の、AB 1およびAB 2に対する符号化単位の送信の例を描く。

【００７５】

図9は、インターリーブされた信頼性制御プロトコルによって使用され得るフォーマットの例示的なセットを示すブロック図である。送信機データフォーマットは、送信機のインターリーブされた信頼性制御プロトコルが符号化単位および対応する信頼性制御情報を受信機のインターリーブされた信頼性制御プロトコルに送信できるフォーマットを表す。この例は、どのブロックから符号化単位が生成されるかを示すブロック番号910、どれだけの符号化単位がこのブロックから送信されたかを示すシーケンス番号920、符号化単位がブロックからどのように生成されるかを示す符号化単位ID 930、および、ブロックを復元するために受信機のインターリーブされた信頼性制御プロトコル内でFECデコーダによって使用され得る符号化単位940を含む。受信機フィードバックフォーマットは、受信機のインターリーブされた信頼性制御プロトコルが信頼性制御情報を送信機のインターリーブされた信頼性制御プロトコルに送信できるフォーマットを表す。アクティブブロックの各々に対して、これは、ブロック番号(950(1)、950(2))、ブロック(960(1)、960(2))を復元するためにどれだけの追加の符号化単位が必要とされるか、および、ブロック(970(1)、970(2))以降これまでに受信された最大のシーケンス番号を含む。

【００７６】

このようにして、図9は、複数の並列ネットワークパスの各々でのデータのロスを表すデータ、符号化単位が受信された複数のブロックの各々を特定するデータを含むロスを表すデータ、ブロックの各々に対して必要とされる符号化単位の数、および、ブロックの各

々に関するネットワークパケットに対して受信された最大のシーケンス番号を定義するデータを、受信機エンドシステム160の1つのような受信機デバイスが送信し、送信機エンドシステム100の1つのような送信機デバイスが受信する例を表す。

【００７７】

図10は、基本送信機のインターリーブされた信頼性制御プロトコルの論理の例を示すフローチャートである。図10の例示的なプロトコルは、図1の送信機エンドシステム100の1つのような送信機デバイスによって実行され得る。プロトコルのこの例では、基本送信機のインターリーブされた信頼性制御プロトコルは、対応する送信機のレート制御プロトコルによって決定される、送信機データを送信する時間であるかどうかを確かめるために、継続的に確認する(1005)。送信機データを送信する時間である場合、基本送信機のインターリーブされた信頼性制御プロトコルは、規則の次のセットを使用して、どのアクティブブロックから符号化単位を生成し送信するかを決定する。

【００７８】

基本送信機のインターリーブされた信頼性制御プロトコルは、各アクティブブロックiに対して次の変数を追跡する(1010)。B_iは、そのブロックを復元するために必要とされる符号化単位の数である。R_iは、基本受信機のインターリーブされた信頼性制御プロトコルがそのブロックから受信したことを受信された受信機フィードバックに基づいて基本送信機のインターリーブされた信頼性制御プロトコルが知っている符号化単位の数である。L_i=B_i-R_iは、基本受信機のインターリーブされた信頼性制御プロトコルがブロックを復元するために受信する必要があることを基本送信機のインターリーブされた信頼性制御プロトコルが知っている確認されていない符号化単位の残りの数である。U_iは、ブロックに対して送信されたが基本送信機のインターリーブされた信頼性制御プロトコルによって確認応答がまだ受信されていない符号化単位の数である。X_iは、基本送信機のインターリーブされた信頼性制御プロトコルがブロックに対する符号化単位をどの程度積極的に送信するかを決定するパラメータである。

【００７９】

これらの変数は、次のように決定され得る。B_iの値は、ブロックのサイズおよび各符号化単位のサイズによって決定される。処理は、送信機がまだ完全には利用可能ではないブロックに対して開始し得ること、すなわち、まだ完全には形成されていないブロックがアクティブとなることができ、符号化単位がそれらから送信され得ることに留意されたい。この場合、B_iは、ブロックが送信機に対して現在利用可能であるサイズであった場合に、ブロックiを復元するために必要とされる符号化単位の数となるように決定されてよく、この場合、B_iは、ブロックiのより多くが送信機に対して利用可能にされるにつれて、ブロックiが完全に形成されるまで、送信処理の間に大きく増えることがあり、ブロックiが完全に形成された時点でB_iの値は変化しないままになる。したがって、B_iは、ブロックiの初期部分がまず利用可能となる1で開始し、ブロックiのより多くが送信機に対して利用可能となるにつれて、B_iが最終的な値に到達するまで、すなわち、B_iが完全なブロックiのサイズに達するまで、増える。

【００８０】

一般に、各符号化単位は、所与のブロックに対して、かつ場合によってはすべてのブロックに対して同じサイズであるが、そのサイズは、データパケットのペイロードに対して適したものとなるように選択され、たとえば、符号化単位の長さは1024バイトであり得る。各ブロックのサイズは一般に同じであってよく、または変化してよく、または送信機におけるデータストリームの到着レートに依存してよく、またはデータパケットの送信レートに依存してよく、またはこれらおよび他の要因の組合せに依存してよい。R_iの値は、ステップ1030で受信された受信機フィードバックに基づいて決定される。U_iの値は、ブロックに対する符号化単位を含む送信された最後の送信機データ中のシーケンス番号と、ブロックに対する受信機フィードバックが受信された最大のシーケンス番号との差である。

【００８１】

このようにして、ブロックを復元するために必要とされる符号化単位の数(たとえば、B_i)は、ブロックのサイズおよびブロックに対する符号化単位のサイズに基づいて計算され得る。同様に、ブロックを復元するために必要とされる追加の符号化単位の数(たとえば、L_i)は、ブロックを復元するために必要とされる符号化単位の数(たとえば、B_i)と受信された符号化単位の数(たとえば、R_i)との差として計算され得る。

【００８２】

X_iの値は、信頼性制御プロトコル全体の関数であり、後で説明されるように、X_iの選択にはトレードオフがある。X_iの値は、ブロックに対するすべての符号化単位の送信の間は不変のままであってよく、または、種々の異なる方法で値を変化させてよく、それらの方法のいくつかが後で説明される。基本的には、各時点におけるX_iは、基本送信機のインターリーブされた信頼性制御プロトコルが、どれだけの追加の符号化単位を、基本受信機のインターリーブされた信頼性制御プロトコルからの任意の追加の受信機フィードバックを何ら伴わずに、ブロックを復元するのに必要とされる最小限のものを超えて送信することを望むかの、尺度である。L_iは、すでに確認応答されている受信された符号化単位を超える、ブロックiを復元するために必要とされる符号化単位の数であるので、かつ、U_iは、インフライトでありまだ確認応答されていないブロックiに対する符号化単位の数であるので、L_i+X_i-U_iは、基本送信機のインターリーブされた信頼性制御プロトコルがこの時点で送信することを望む、ブロックiに対する追加の符号化単位の数である。

【００８３】

X_iの値に対するトレードオフは次の通りである。X_iが増えるにつれてグッドプットは下がる。それは、アクティブブロックiを復元するために必要とされる最小限のものを場合によっては最大でX_i個超える符号化単位が、基本受信機のインターリーブされた信頼性制御プロトコルによって受信され得るからである。一方で、アクティブブロックの全体のサイズは、X_iが増えるにつれて小さくなる。それは、アクティブブロックiの信頼性のある受信を完了するために割り振られるパケットタイムスロットの数が、X_iが増えるにつれて増えるからである。ブロックiに対するX_i個の符号化単位は失われ得るが、それでも基本受信機は、追加の符号化単位の送信を引き起こすための受信機フィードバックを待つことなくブロックを復元することができ、これは、ブロックiがアクティブになったときのブロックiのより高速な復元を可能にする。X_iの関数としての全体のバッファサイズとグッドプットとのトレードオフは、TF信頼性制御プロトコルまたは非符号信頼性制御プロトコルのような他の信頼性制御プロトコルの対応するトレードオフよりもはるかに好ましいことが、判明している。

【００８４】

ステップ1015において、不等式L_i+X_i-U_i>0を満たすアクティブブロックiがあるかどうかを判定するための試験が行われる。L_iの値は、受信機フィードバックによってすでに確認応答されている符号化単位に基づく、受信機がブロックを復元するために必要とするであろう符号化単位の数である。U_iは、このブロックに対してインフライトである確認応答されていない符号化単位の数であるので、L_i-U_iは、インフライトである符号化単位のいずれもが失われない場合に送信される必要があるであろう追加の符号化単位の数であり、したがって、L_i-U_iが0以下である場合、受信機は、ブロックに対するインフライトであるすべての符号化単位が到着すれば、ブロックを復元することが可能である。一方、符号化単位のいくつかが失われることがあり、X_iは、後続の受信機フィードバックによって引き起こされるブロックに対する追加の符号化単位の送信を行わなくても済むように、ロスから守るために送信機が前もって送信することを望む、追加の符号化単位の数である。

【００８５】

したがって、L_i+X_i-U_i>0である場合、送信機は、ブロックiに対してより多くの符号化単位を送信することを望み、L_i+X_i-U_iが0または負である場合、送信機は、ブロックiに対してより多くの符号化単位を送信することを望まない。したがって、ステップ1015において、L_i+X_i-U_i>0を満たすアクティブブロックiがある場合、ステップ1020におい

て、そのような最早のアクティブブロックに対して、符号化単位が生成され、対応する送信機データが送信される。そのようなアクティブブロックがない場合、ステップ1025において、符号化単位が生成され、対応する送信機データがすべてのアクティブブロックの中で最早のアクティブブロックから送信される。好ましくは、パラメータは、ステップ1025の実行を強いる、ステップ1015の条件を満たすブロックの不在を可能な限り避けるような方法で設定される。それは、基本的には、ステップ1025は、基本送信機のインターリーブされた信頼性制御プロトコル内のバッファを空にするための最後の手段として行われるべきであるからである。ステップ1020および1025における送信されるデータは、複数のパスを介して同じ受信機に送信され得る。

【００８６】

まだ完全には形成されていないアクティブブロックがある場合、ステップ1015は、これらのまだ完全には形成されていないブロックも考慮するように修正されるべきである。1つの変形形態では、アクティブであるがまだ完全には形成されていないブロックに対するステップ1015で示される対応する条件は代わりに、まだ完全には形成されていないアクティブブロックに対するソース符号化単位のすべてが送信されたかどうかであるべきであり、送信されていなければ、残りの送信されていないソース符号化単位が送信され得る。したがって、この変形形態では、ブロックiに対して送信され得る符号化単位の数は最大でB_iであり、B_iは、部分的に形成されているブロックiの中のソース符号化単位の現在の数である。第2の変形形態では、アクティブであるがまだ完全には形成されていないブロックに対するステップ1015で示される対応する条件は代わりに、X_i=0に設定すること、および、失われた、または受信されたものとして確認応答された符号化単位の中でどの符号化単位が失われたかも示す、改良された受信機フィードバックに基づいて、受信されていないものとして確認応答された符号化単位を再送信することであるべきである。したがって、この変形形態では、ステップ1015の条件L_i+X_i-U_i>0は、ブロックiがアクティブでありまだ完全には形成されていなければ、条件L_i-U_i>0により置き換えられる。この場合、すべてのソース符号化単位がすでに送信されているが、一部が失われたものとして確認応答されている場合、これらの符号化単位は、条件L_i-U_i>0が満たされれば再送信され得る。

【００８７】

プロトコルの1つの変形形態は次の通りである。アクティブブロックの番号は1で開始し、すなわち、データストリームの第1のブロックがアクティブ化される。ステップ1015の条件を満たすアクティブブロックがないときだけ、データのストリーム中の新たなブロックがアクティブ化される。この簡単な戦略を使用すると、ブロックは、必要なときだけアクティブブロックになるので、アクティブブロックの数、そして結果としてバッファサイズは、ブロックiに対してグッドプットB_i/(B_i+X_i)を保証するのに必要とされる数に自己調整される。

【００８８】

プロトコルの別の変形形態は次の通りである。この変形形態では、全体のバッファサイズは常に同じサイズのままであるが(すべてのブロックが同じサイズであれば、これは常に一定の数のアクティブブロックがあることを意味する)、グッドプットは変化し得る。ステップ1015の条件を満たすアクティブブロックがないときは常に、アクティブブロックに対するX_iの値は、ステップ1015の条件を満たすアクティブブロックが現れるまで増大させられる。適切であるときは常に、ステップ1015の条件を満たすアクティブブロックが常に存在するという制約を伴いながら、アクティブブロックiに対するX_iの値が下げられる。X_iの値を増やし減らすための多くの可能な方法があり、たとえば、すべての値を等しく増やす、すべての値を比率的に等しく増やす、最初のアクティブブロックの値を最後のアクティブブロックの値よりも大きく増やす、最後のアクティブブロックの値を最初のアクティブブロックの値よりも大きく増やすという方法がある。同様の戦略が、X_iの値を減らすために使用され得る。当業者は、多くの他の変形形態も想起することができる。

【００８９】

説明するには多すぎるプロトコルのこれらの変形の多くの他の組合せおよび拡張があることが、当業者には明らかであろう。

【００９０】

ステップ1030において、任意の受信機フィードバックが受信されたかどうかが確認され、受信されている場合、パラメータのすべてが、すなわち、すべてのアクティブブロックiに対するパラメータR_i、U_iおよびX_iが、ステップ1035においてこれに基づいて更新される。ステップ1040において、最早のアクティブブロックが完全に復元されたものとして確認応答されたかどうかが確認され、確認応答されている場合、最早のアクティブブロックがステップ1045で非アクティブ化されて処理はステップ1040に戻り、確認応答されていない場合、処理はステップ1050へ続く。ステップ1050において、別のブロックがアクティブになる準備ができているかどうかが確認され、準備ができている場合、ステップ1060において、次のブロックがアクティブにされて処理はステップ1050に戻り、準備ができていない場合、処理はステップ1005へ続く。一般に、次のブロックまたはいくつかの次のブロックは、現在のアクティブブロックが送信されている間は準備中であることがあり、最早のアクティブブロックが非アクティブ化されるべきとき、またはその前に、アクティブ化される準備ができることがある。

【００９１】

このようにして、図10は、複数の並列ネットワークパスを介して前方誤り訂正されたデータをクライアントデバイスへ送信するステップと、ネットワークパスの各々を通じて送信されるデータのロスを表すデータをクライアントデバイスから受信するステップと、ロスを表すデータに基づいて、並列ネットワークパスを通じた後続のデータ送信のために送信される前方誤り訂正データの量を修正するステップとを含む、方法の例を描く。ロスを表す受信されたデータは、並列ネットワークパスの各々に対して、それぞれの並列ネットワークパスを介して送信されるパケットフローに対して受信された最大のシーケンス番号を特定するデータ(たとえば、図13に関して以下で論じられるような)、ならびに/または、符号化単位がクライアントデバイスによって受信された複数のブロックの各々を特定するデータ、ブロックの各々に対して必要とされる符号化単位の数、および、ブロックの各々に関するネットワークパケットに対してクライアントデバイスによって受信された最大のシーケンス番号を定義するデータ(図9に関して上で論じられたような)を含み得る。

【００９２】

図11は、基本受信機のインターリーブされた信頼性制御プロトコルの論理の例を示すフローチャートである。プロトコルのこのバージョンでは、基本受信機のインターリーブされた信頼性制御プロトコルは、たとえば図9に示される送信機データフォーマットであり得る、送信機データが受信されたかどうかを確かめるために継続的に確認する(1105)。受信されている場合、基本受信機のインターリーブされた信頼性制御プロトコルは、ステップ1110ですべてのアクティブブロックについての情報を更新し、送信機データ内の受信された符号化単位がアクティブブロック1115からのものかどうかを確かめるために確認する。符号化単位が、すでに復元されているブロックからのものである場合、または、現在のアクティブブロックとするにはデータストリーム中で前方にありすぎるブロックからのものである場合、符号化単位はステップ1135で廃棄され、したがって、ブロックを復元するには有用ではないためその符号化単位は無駄になる送信機データである。それ以外の場合、符号化単位が、その生成元のアクティブブロックに対する符号化単位の蓄積に追加され、どれだけの符号化単位がアクティブブロックを復元するために必要かが、ステップ1120で更新される。

【００９３】

ブロックiに対する必要とされる符号化単位の数は、B_iから、受信された符号化単位の数を引いたものとして計算される。B_iの値を基本受信機のインターリーブされた信頼性制御プロトコルに伝える種々の方法があり、たとえば、B_iの値は各送信機データに含まれてよく、B_iの値は別個の制御メッセージで送信されてよく、B_iの値はすべてのブロックに対して同じでありセッション開始の間に伝えられてよい、などである。

【００９４】
次いで、ステップ1125において、最早のアクティブブロックに対する符号化単位の必要な数が0かどうかが確認され、0である場合、次いで基本受信機のインターリーブされた信頼性制御プロトコルは、ステップ1130において、FECデコーダを使用してアクティブブロックを復元し、次のアクティブブロックに対する符号化単位の受信に備える。基本受信機のインターリーブされた信頼性制御プロトコルはまた、対応する受信機のレート制御プロトコルによって決定される、受信機フィードバックを送信する時間かどうかを確かめるために、継続的に確認する(1140)。その時間である場合、ステップ1145において次いで受信機フィードバックが準備され送信され、そのフィードバックはたとえば、図9に示される受信機データフォーマットであり得る。

【００９５】
上記は、基本的なインターリーブされた信頼性制御プロトコル全体の部分的な説明であることに留意されたい。たとえば、これは、受信機フィードバックが基本受信機のインターリーブされた信頼性制御プロトコルによって送信される条件を規定しない。これは、受信された送信機データの受信によって、時々鳴動するタイマーによって、またはこれらのイベントの任意の組合せ、もしくは受信機のレート制御プロトコルによって決定されるような任意の他のイベントによって引き起こされ得る。一般に、受信機フィードバックは、基本受信機のインターリーブされた信頼性制御プロトコルにおける符号化単位の受信の進行について基本送信機のインターリーブされた信頼性制御プロトコルが定期的に知らされ続けるのに十分頻繁に、しかし、基本送信機のインターリーブされた信頼性制御プロトコルから基本受信機のインターリーブされた信頼性制御プロトコルに送信される符号化単位を含む送信機データとほぼ同じ帯域幅を消費するほど頻繁にではなく、送信される。

【００９６】
基本的なインターリーブされた信頼性制御プロトコルは、TF信頼性制御プロトコルまたは非符号信頼性制御プロトコルよりもはるかに良好な、グッドプットとバッファのサイズとの間のトレードオフを有し得る。たとえば、基本的なインターリーブされた信頼性制御プロトコルに対して最大でも2つのアクティブブロックしかないと仮定する。Bを符号化単位の単位での各データブロックのサイズとし、Rをパケットがレート制御プロトコルによって送信される際のレートとし、RTTを送信機エンドシステムと受信機エンドシステムとの間の往復遅延時間とし、N=R*RTTとし、Xをすべてのアクティブブロックに対する不変の定数とする。この例では、これらのパラメータのすべてが不変の値を有すると仮定するが、一般には、それらのパラメータはデータ転送の間に大きく変化することがあり、B≧Nと仮定する。

【００９７】
送信機と受信機の間でのパケットロスはないと仮定する。次いで、基本送信機のインターリーブされた信頼性制御プロトコルは、最早のアクティブブロックに対するB+X個の符号化単位を送信し、次いで、基本受信機のインターリーブされた信頼性制御プロトコルによって最早のアクティブブロックが成功裏に復元されたことを示す受信機フィードバックを受信するまで、次のアクティブブロックからの符号化単位を送信する。この時点で、基本送信機のインターリーブされた信頼性制御プロトコルは最早のアクティブブロックを非アクティブ化し、いくつかの符号化単位がすでに送信されている次のアクティブブロックが最早のアクティブブロックになり、次のブロックがアクティブブロックとなるようにアクティブ化される。したがって、B+X個の符号化単位は、長さBのブロックを復元するために使用されるので、送信された符号化単位のうちのXが無駄になる。

【００９８】
一方、B≧Nである場合、図9のステップ1015で示される不等式を満たすアクティブブロックが常に存在する。したがって、グッドプットはB/(B+X)であるが、バッファの全体のサイズは、2つのアクティブブロックがある場合は2*Bである。ある例として、符号化単位のサイズが1キロバイトであり、レートRが1000符号化単位毎秒=1メガバイト毎秒=8メガビット毎秒であり、RTTが1秒であると仮定する。すると、N=R*RTT=1メガバイトである。ブ

ロックのサイズが1メガバイトである場合、これはB=1000個の符号化単位でありXが10個の符号化単位に設定されることを意味し、グッドプットは約(B/(B+X))=0.99であり、すなわち、送信される符号化単位の最大でも1%しか無駄にされず、一方で、全体のバッファサイズはわずか2MBであり、これは、基本送信機のインターリーブされた信頼性制御プロトコルがこの例では約2秒のレイテンシを追加することを意味する。このバッファサイズは、同じ状況での送信機のTF信頼性制御プロトコルのバッファサイズよりも、25倍小さいことに留意されたい。

【００９９】

パケットロスがない、上で説明された例では、Xの値は0に設定されてよく、グッドプットを最大で1.0に上げる。しかしながら、パケットロスがあるとき、X>0という設定は大きな利点を有し得ることが判明している。たとえば、上の例で送信された各々の1000個の符号化単位のうち、最大で10個の符号化単位が失われる場合、同じグッドプットおよびバッファサイズがX=10によって達成されるが、これはX=0では必ずしも当てはまらないことを、分析は示している。パケットロスがより可変であり知られていないとき、具体的には、Bパケットごとの失われるパケットの数がXより多くなり得るときでも、基本的なインターリーブされた信頼性制御プロトコルによって達成され得るグッドプットおよびバッファサイズは非常に良好であり、TF信頼性制御プロトコルまたは非符号信頼性制御プロトコルを使用して達成され得るものよりも定量的に良好であることが判明している。

【０１００】

別の例として、送信レートがパケット毎秒単位でRであり、往復遅延時間RTTが一定のままであり、N=R*RTTであると仮定する。各パケットがpの確率で失われるように、パケットロスがランダムであると仮定する。サイズがB_iである各ブロックiはパケットの単位で同じサイズCであり、各X_iは同じ値Yであるとさらに仮定する。必要なときにだけ新たなブロックをアクティブ化する、上で説明されたプロトコルの変形形態が使用されるとさらに仮定する。ブロックが最初にアクティブ化された時間から、ブロックが復元されたという確認応答が受信機から受信されたのでブロックが非アクティブ化された時間までの、ブロックを考える。ブロックのC-N個のパケットが確認応答されたある時間tにおいて、確認応答されていないインフライトであるF=N+Y個のパケットがあり、送信機は、ブロックを復元するために受信機がこれらのパケットのうちのN=F-Y個を必要としていることを知っている。時間t+RTTにおいて、時間tにおけるブロックに対してインフライトであったF個のパケットのうち、(1-p)*F個のパケットが受信機によって受信されており、送信機は確認応答を受信している。

【０１０１】

したがって、時間t+RTTにおいて、送信機は、受信機が必要とする残りのパケットの数が現在はN-(1-p)*F=p*F-Yであり、したがってインフライトであるパケットの数が現在はp*Fであることを知っている。論理を続けると、時間t+i*RTTにおいて、送信機は、受信機が必要とする残りのパケットの数がp^i*F-Yであり、したがってインフライトであるパケットの数がp^i*Fであることを知っている。受信機が必要としていることを送信機が知っているパケットの数が0を下回るとき、ブロックは完了し、これは、iがp^i*F-Y≦0を満たすときの時間t+i*RTTにおいて当てはまる。この不等式が真であるときのiの最小の値は、iが約ln((N/Y)+1)/ln(1/p)であるときである。

【０１０２】

各RTTにおいて、約(1-p)*N個のパケットが受信機によって受信されるので、上記のことは、ブロックが受信されたものとして確認応答される時間までに、送信機プロトコルがそのブロックを超えてデータストリーム中で考慮のために先取りし得る最遠のものが、最大で(ln((N/Y)+1)/ln(1/p))*(1-p)*N個のパケットであることを意味する。pのすべての値に対して(1-p)/ln(1/p)≦1であることに留意すると、これは、バッファのサイズが最大でC+ln((N/Y)+1)*N個のパケットの長さであることを意味する。当然、これはすべて、ランダム処理が予想される挙動の通りに厳密に振る舞うことを仮定したものであるが、少なくともYは小さすぎることはないので、これは、プロトコルがどのように振る舞うかというこ

とについて概略的な考え方を与える。この場合、グッドプットはC/(C+Y)である。したがって、たとえば、RTT=1、R=1000、C=1000、Y=50である場合、N=1000であり、ln(1000/50)は約3であり、バッファサイズは約1000+(3+1)*1000=5000であり、グッドプットは1000/(1000+50)でありこれは約0.95である。

【０１０３】

この説明を読んだ後で明らかになるであろう上で説明された基本的なインターリーブされた信頼性制御プロトコルに対する、多くの変形形態がある。たとえば、上で説明されたように、送信機の信頼性制御プロトコルは、一度に3つ以上のアクティブブロックを使用することができ、これは、より多くのアクティブブロックを管理する際のさらなる複雑さという犠牲を伴って、送信機および受信機の信頼性制御プロトコルにおいて使用されるバッファの全体のサイズを減らすことが可能であるという、潜在的な利点を有する。

【０１０４】

変形形態の別の例として、どのアクティブブロックからの符号化単位が送信されるべきかを決定するために、ランダム処理を使用するのが有益であり得る。これは、パケットロスのパターンが対称的であることがあり、必ずしもランダムではないからであり、したがって、次にどの符号化単位を送信するかを選択するために使用される任意の決定論的な手順に対して、一部のブロックが決して復元されないがそれでもパケットが受信機に配信されるようなパケットロスのパターンがある。たとえば、決定論的な手順が特定のアクティブブロックからの符号化単位を送信すると常に、その符号化単位が失われるが、任意の他のアクティブブロックに対する符号化単位を送信すると常に、その符号化単位が受信機に到着するという、ロスのパターンを考える。

【０１０５】

すると、この例では、受信機が依然として符号化単位を受信していても、受信機はそのアクティブブロックを決して復元しない。このタイプの系統的なロスを克服するには、送信機の信頼性制御プロトコルが、どのアクティブブロックから次の符号化単位を送信するかをランダム化することが有利である。これを達成する1つの簡単な方法は、送信機の信頼性制御プロトコルが、送信されるべきQ個の符号化単位の束を一緒にバッファリングして、次いで、Q個の符号化単位の各束をランダムな順序で送信することである。より洗練された方法も使用されてよく、たとえば、送信されるべき各符号化単位に対して、符号化単位が送信されるべき次の時間に符号化単位が送信される動的に変化する確率を割り当て、ここでこの確率は、それが選択されないより多くの時間を増やす。別の変形形態は、ステップ1015の条件を満たすアクティブブロックの中から、(より早いアクティブブロックを好むことがあり経時的に大きく変化することがある適切に選ばれた確率分布を使用して)送信される符号化単位がランダムに生成されるように、基本送信機のインターリーブされた信頼性制御プロトコルの図10に示されるようなステップ1020を修正することである。

【０１０６】

パラメータX_iが、アクティブブロックiに対する符号化単位をいつ送信するかを決定するために使用される場合、送信の間にX_iをどのように調整するかについての多くの変形形態がある。1つの例は、X_iをある値に固定し、その値を送信の間は維持することである。たとえば、X_iは0に、または10のよう何らかの他の固定された値に設定されてよい。別の例は、X_iをアクティブブロックiからの符号化単位の送信の始めにおいてある値に固定することであり、X_iは次いで、符号化単位が送信され、アクティブブロックiからの符号化単位を送信するための条件が満たされなくなるたびに、インクリメントされる。X_iがどのようにインクリメントされ得るかについて、多くの変形形態がある。ある例として、X_iは、最初のN回のそのような機会ではインクリメントされず、その後の各々の機会ではN/Bだけインクリメントされ得る。いくつかのステップにおいて、X_iのインクリメントが負であり得る可能性もある。

【０１０７】

他の変形形態として、基本的なインターリーブされた信頼性制御プロトコルにおいて説明されたように各アクティブブロックiに対するパラメータX_iのみを使用する代わりに、

ある特定のアクティブブロックから符号化単位が送信されるべきかどうかを判定する他の方法を使用することができる。たとえば、パケットロスの確率の平均が維持されてよく、次いで、アクティブブロックから送信されることが許可された符号化単位の数が、最近のパケットロスの確率が現在のパケットロスの確率を良好に予測するものであるという仮定に基づいて決定され得る。たとえば、平均のロスの確率が現在pである場合、1つの戦略は、条件がL_i+X_i/(1-p)-U_i*(1-p)>0となるように、基本送信機のインターリーブされた信頼性制御プロトコルの図10に示されるようなステップ1015を修正することである。

【０１０８】

この特定の選択の背後にある理論は、U_i個の符号化単位がアクティブブロックiに対してインフライトである場合、それらのうちの一部分1-pのみが基本受信機のインターリーブされた信頼性制御プロトコルに到着し、X_i/(1-p)個の追加のパケットが送信される場合、X_iが基本受信機のインターリーブされた信頼性制御プロトコルに到着するというものである。したがって、全体として平均的に、基本受信機のインターリーブされた信頼性制御プロトコルは、アクティブブロックiに対してB_i+X_i個の符号化単位を受信し、X_i個の追加の符号化単位の値は、ブロックを復元するのに十分な数の符号化単位の送信に対する受信機フィードバックに依存するのを避けるために、パケットロス確率の変動を十分考慮するように設定され得る。

【０１０９】

インターリーブされた信頼性制御プロトコルの他の変形形態は、パケットが送信順序と同じ順序で受信機に到着しないことがあるという可能性を考慮する。したがって、受信機からの後続の受信機フィードバックは、たとえば、以前の受信機フィードバックよりも多数の、所与のアクティブブロックに対する受信された符号化単位を、そのブロックから受信された最大のシーケンス番号が同じであったとしても報告することができる。したがって、基本的なインターリーブされた信頼性制御プロトコルにおける論理は、並べ替えられたパケットを考慮してそれに適合するように、送信機と受信機の両方において修正され得る。

【０１１０】

前に説明されたように、図10に示されるような基本送信機のインターリーブされた信頼性制御プロトコルのステップ1025は一般に、少なくとも1つのアクティブブロックが各時点で条件1015を満たすように、パラメータを適切に設定することによって回避されることになる。ステップ1025の変形形態は、どのアクティブブロックからの符号化単位を生成し送信するかの選択を変化させることである。たとえば、アクティブブロックは、ステップ1025でランダムに選択されてよく、またはこの選択は、アクティブブロックのセットの中で周期的に変化してよい。

【０１１１】

図10のステップ1040から1060は、復元されたブロックを非アクティブ化し、送信すべき追加のブロックをアクティブ化するための方法を説明する。1つの簡単な方法は、復元によって最早のブロックが非アクティブ化されるときに次のブロックを常にアクティブ化することであり、これによって、同じ数のアクティブブロックを常に維持する。全体のバッファサイズおよび結果としてのレイテンシを節約できる変形形態は、最新の現在のアクティブブロックを超えて、符号化単位をブロックから送信する時間になったときにだけ、次のブロックをアクティブ化することである。

【０１１２】

基本的なインターリーブされた信頼性制御プロトコルのいくつかの変形形態では、任意の時点におけるアクティブブロックの数は固定される。1つの変形形態は、アクティブブロックの数が、送信のためにどのようなレートデータが利用可能にされるか、どれだけのパケットロスが発生しているか、パケットの送信レートの変動などを含む、種々の要因に応じて変化するのを可能にすることである。たとえば、低パケットロスの条件および低送信レートの条件のもとでは、アクティブブロックの数は小さいままにされ得るが、ロスの条件が悪くなるにつれて、または送信レートが上がるにつれて、アクティブブロックの数

は一時的に増えることが許容され得る。したがって、バッファリングおよびレイテンシは、プロトコルが動作している条件に応じて大きく変化する。

【０１１３】

アクティブブロックの全体サイズも、アクティブブロックの数が固定されたままであったとしても、変化することが可能にされ得る。この場合、各々の後続のアクティブブロックのサイズは前のブロックとは異なり得る。たとえば、データ利用可能レートが上がるにつれて、後続のアクティブブロックのサイズも大きくなってよく、送信レートが上がるにつれて、後続のアクティブブロックのサイズが大きくなってよい。各アクティブブロックの長さは時間の関数であってよく、たとえば、新たなブロックが形成される前に最大でそれだけの時間が経過してよく、各アクティブブロックの長さは長さの関数であってよく、すなわち、各アクティブブロックは最大でそれだけの長さであってよく、または、各アクティブブロックの長さは、これらおよび他の要因の組合せであってよい。

【０１１４】

あるブロックの終了と次のブロックの開始は、インターリーブされた信頼性制御プロトコルによって自動的に判断されてよく、アプリケーションによって決定されてよく、またはこれらおよび他の要因の何らかの組合せであってよい。たとえば、データストリームのブロックは、アプリケーションに対する論理的な意味を有することがあり、たとえば、MPEGストリームのGroup of PicturesブロックまたはI-frameであることがあるので、インターリーブされた信頼性制御プロトコルがデータのストリームをブロックに区分する方法は、論理的なアプリケーションブロックの境界を尊重し得る。あるいは、アプリケーションは、ブロック間の好ましい境界をインターリーブされた信頼性制御プロトコルに示すことができ、インターリーブされた信頼性制御プロトコルは、可能な限りこれらの境界を尊重しようとするが、それでも、アプリケーションにより提供されるもの以外の点にブロック間の境界を設けることが許容され得る。

【０１１５】

インターリーブされた信頼性制御プロトコルの別の変形形態は、プロトコルが、すべてのブロックを受信機へと順番に信頼性をもって配信することではなく、代わりに、他の制約のもとでこの目標を達成するために可能な限り努力することを、可能にすることである。たとえば、ストリーミングの適用例では、データのストリームを可能な限り信頼性をもって配信することが重要であり得るが、データストリームに対するタイミングの制約のような他の制約もある。たとえば、ある時間の後で、データのある部分がもはや関連がないこと、または、たとえば双方向のビデオ会議の適用例においてインターリーブされた信頼性制御プロトコルがどれだけのレイテンシを加えられるかについての強い制限があることがある。これらの場合、送信機のインターリーブされた信頼性制御プロトコルおよび受信機のインターリーブされた信頼性制御プロトコルは、ブロックの一部が、それらが完全に復元される前は省略されることを可能にするように修正され得る。

【０１１６】

たとえば、送信機のインターリーブされた信頼性制御プロトコルは、所与の長さの時間だけアクティブブロックがアクティブになるのを許容するように制約されてよく、または、ブロックに対する符号化単位を送信することがその後はもはや許可されない、アプリケーションによって与えられる各ブロックに対する厳しい時間的制約を有してよく、または、各ブロックに対して与えられた最大の数の符号化単位だけを送信することが許可されてよく、またはこれらの制約の任意の組合せがある。同様の制約は、受信機のインターリーブされた信頼性制御プロトコルに適用可能であり得る。これらの適用例では、インターリーブされた信頼性制御プロトコルは、これらの制約を尊重するように修正され得る。

【０１１７】

インターリーブされた信頼性制御プロトコルのいくつかの変形形態では、1つの送信機と1つの受信機がある。他の変形形態は、限定はされないが、1つの送信機と複数の受信機、1つの受信機と複数の送信機、複数の送信機と複数の受信機を含む。たとえば、1つの送信機/複数の受信機の変形形態では、送信チャンネルがブロードキャストチャンネルまた

はマルチキャストチャンネルであるとき、送信機の信頼性制御プロトコルは、送信機が各アクティブブロックiに対して、図10のステップ1010における任意の受信機からの受信される確認応答された符号化単位の最小の数として、R_iの値を計算するように、修正され得る。

【０１１８】

1つの送信機/複数の受信機の変形形態の別の例として、送信機が各受信機にパケットの別個のストリームを送信するとき、送信機の信頼性制御プロトコルは、送信機が、各アクティブブロックiに対して、かつ、各受信機jに対して、アクティブブロックiに対する受信機jからの受信される確認応答された符号化単位の数として、R_ijの値を計算し、図10のステップ1010でL_ij=B_i-R_ijを計算するように修正されてよく、U_ijは、受信機jに送信されたアクティブブロックiに対する、送信されたがまだ確認応答されていない符号化単位の数として計算されてよく、そして、ステップ1015の条件は、いくつかの受信機jに対して、L_ij+X_i-U_ij>0となるようなアクティブブロックiがあるかどうかを判定するように変更され得る。

【０１１９】

別の例として、多数の送信機/1つの受信機の変形形態では、受信機の信頼性制御プロトコルは、受信機が複数の送信機から同時に符号化単位を受信し、同じまたは異なるアクティブブロックに対して、すべての送信機へのブロードキャストチャンネルとマルチキャストチャンネルのいずれかによって、または、各送信機への場合によっては別個の受信機フィードバックを伴う別個のパケットストリームを使用して受信機フィードバックを送信するように、修正され得る。別の例として、複数の送信機/複数の受信機の変形形態では、1つの送信機/複数の受信機の場合および複数の送信機/1つの受信機の場合に上で説明された修正されたステップは組み合わされ得る。

【０１２０】

別の変形形態は、送信機が、送信機のインターリーブされた信頼性制御プロトコルの別個のインスタンスを、または、異なるデータストリームを考慮する送信機のインターリーブされた信頼性制御プロトコルのバージョンを各々使用して、複数のデータストリームを同時に送信していてよいことであり、たとえば、すべてのストリームに対するすべてのパケットの全体の送信レートは制限されていることがあるので、送信機は、いくつかのデータストリームに対するパケットの送信を、他よりも優先すると判断することができる。同様に、受信機は、受信機のインターリーブされた信頼性制御プロトコルの別個のインスタンスを、または、異なるデータストリームを考慮する受信機のインターリーブされた信頼性制御プロトコルのバージョンを各々使用して、複数のデータストリームを同時に受信していてよく、たとえば、すべてのストリームに対するすべてのパケットの全体の受信レートは制限されていることがあるので、送信機は、パケットを受信することと、いくつかのデータストリームに対する受信機フィードバックを処理して送信することとを、他よりも優先すると判断することができる。

【０１２１】

上記の変形形態のいずれもが、互いに組み合わされ得る。たとえば、タイミングおよび/または帯域幅の制限がたとえば原因でいくつかのブロックが受信機へと信頼性をもって配信され得ないプロトコルが、複数の送信機/複数の受信機の変形形態と組み合わされ得る。

【０１２２】

このようにして、図11は、複数の並列ネットワークパスを介して前方誤り訂正されたデータをサーバデバイスから受信するステップと、ネットワークパスの各々でのデータのロスを決定するステップと、ネットワークパスの各々でのデータのロスを表すデータをサーバデバイスに送信するステップとを含む、方法の例を描く。

【０１２３】

図12は、本開示の技法による、マルチパスのFECベースの信頼性転送プロトコルの方法を利用し得る、マルチパスストリーミングシステムのブロック図である。この例では、ビ

デオ生成器(1205)がビデオストリーム(1210)を生成し、ビデオストリームは送信機の転送プロトコル(1215)によって受信される。送信機の転送プロトコル(1215)は、ビデオストリーム(1210)のためにどのデータを送信すべきかを決定し、送信されるべきデータの各々に対して、どのパスフローに沿ってそれを送信すべきかを決定する。

【０１２４】

各パスフローに対して、そのパスフローに対する送信機(1220(1)、1220(2))は、受信機の転送プロトコル(1225)によって少なくとも一部受信されるべき、そのパスフローに対するデータを、ネットワーク(1222)を通じて送信する。受信機の転送プロトコル(1225)は、元のビデオストリームを可能な限り最大の忠実度で復元し、このビデオストリーム(1230)を他のデバイスに対して、場合によっては他のネットワークを通じて、かつサーバの他のセットを通じて、エンドユーザのデバイスの再生に向けて生成する。ビデオは例示的なデータストリームであり、非ビデオストリーム(たとえば、オーディオストリームまたは他のデータストリーム)が同様に扱われ得ることに留意されたい。

【０１２５】

複数のパスを使用することによって、かつどのパスがいつ使用されるかを調整することなどによって、本明細書で説明されたように、全体的なレイテンシの低下またはより高い全体的なデータスループットのような改善を得ることができ、これらは一般に、より高品質のストリーム体験を示唆する。これは、複数のパスが個々のレイテンシのような異なるレイテンシを有する場合であっても当てはまることがあり、帯域幅および損失率、ならびにそれらの特性は、パスごとに変化するのに加えて、時間ごとにも変化し得る。したがって、ある順序でビデオ生成器によって最初に放出されたパケットは、一部のパスが他のパスよりも高速にパケットを配信し得るので、異なる順序で受信され得る。

【０１２６】

このアーキテクチャでは、送信機の転送プロトコル(1215)および受信機の転送プロトコル(1225)は、それらが互いに通信するために使用するデータフォーマットのよく確立されたセットを有してよく、送信機の転送プロトコル(1215)から受信機の転送プロトコル(1225)へ流れるデータとともに、受信機の転送プロトコル(1225)から送信機の転送プロトコル(1215)へ流れる制御情報およびフィードバック情報が存在し得る。

【０１２７】

パスフローの数および送信機の数は任意の数であってよく、この数は異なっていてよく、たとえば、10個のパスフローおよび3個の送信機があってよく、パスフローのうちの4個が第1の送信機(1220(1))を通ってよく、3個のパスフローが第2の送信機と第3の送信機の各々を通ってよい。送信機(1220)は、送信機の転送プロトコル(1215)と同じハードウェアデバイス内に共設されてよく、または送信機(1220)は、送信機の転送プロトコル(1215)をホストするハードウェアデバイスとは別個のハードウェアデバイスとともにあってよく、これらのデバイスは、たとえばBluetooth(登録商標)またはWiFiを使用して、TCPまたはUDPのような標準的なトランスポートプロトコルに基づいて互いに通信することができる。送信機(1220)がデータを受信機の転送プロトコル(1225)に送信するネットワークは、異なる送信機に対しては異なっていてよく、たとえば、送信機の一部は3Gネットワークを使用して送信することがあり、他の送信機はLTEを使用することがあり、他の送信機はWiFiを使用することがある。送信機は、同じタイプまたは異なるタイプの様々な事業者ネットワークを使用することができ、たとえば、第1の送信機はAT&Tのネットワークを使用してよく、一方第2の送信機はVerizonのネットワークを使用してよい。受信機の転送プロトコル1225の前に中間の受信デバイスがあってよく、たとえば、第1の送信機は、第1のCDNによって操作される第1のサーバに送信することができ、一方第2の送信機は、第2のCDNによって操作される第2のサーバに送信することができ、第1のサーバおよび第2のサーバは、それらが受信したものを受信機の転送プロトコル1225に送信することができる。

【０１２８】

マルチパスのFECベースのインターリーブされた信頼性制御プロトコルの方法が、図13を参照してここで説明される。図13は、例示的なマルチパスの信頼性制御プロトコルのデ

ータパケットフォーマットと対応するフィードバック情報フォーマットとを示す概念図である。

【０１２９】

図13の上部に示されるように、各データパケットは、フロー識別子、FID(1305)と、FIDのシーケンス番号、FIDに対するSEQN(1310)と、ソースブロック番号、SBN(1315)と、ソースシンボル単位でのソースブロックの長さ、SBL(1318)と、符号化シンボル識別子(符号化ユニット識別子とも呼ばれる)、ESI(1320)と、1つまたは複数の符号化シンボル(1325)とを含む。(符号化シンボルは符号化単位とも呼ばれることがある。)FID(1305)は、このパケットが送信されるべきであるパスフローを特定し、FIDに対するSEQN(1310)は、このフローに送信される各パケットに対して1だけインクリメントする番号であり、したがって、FIDに対するSEQN(1310)はFID(1305)によって調査される。

【０１３０】

SBN(1315)は、このパケットで搬送されている符号化シンボル(1325)がどのソースブロックから生成されたかを特定し、SBNは、送信されるべきデータの各々の後続のソースブロックに対して1だけ全般にインクリメントする番号である。SBL(1318)は、ソースブロック中のソースシンボルの数を特定する。SBL(1318)は、パケットがソースシンボルを搬送するときは省略されてよく、これはいくつかの適用例では好ましい。それは、ソースブロックのソースシンボルの少なくともいくつかが送信されるときにソースブロック中のソースシンボルの数が知られていない、すなわちオープンソースブロックであることがあるからであり、たとえば、以下のオープンソースブロックの説明を参照されたい。

【０１３１】

あるいは、SBL(1318)は、すべてのデータパケット中で搬送され得るが、その値は、ソースブロックのサイズが決定される前に送信されるソースシンボルを搬送するパケットに対しては0に設定されてよく、または、SBL(1318)は、ソースシンボルを搬送するすべてのパケットに対して0に設定されてよく、または、SBL(1318)は、ソースシンボルを搬送するパケットが送信される時点でのソースブロック中のソースシンボルの現在の数に設定されてよい。SBL(1318)は、好ましくは、修復シンボルを搬送するすべてのパケットに対して、ソースブロック中のソースシンボルの数に設定される。SBL(1318)はまた、2つのサブフィールドに区分されてよく、1ビットのフラグはソースブロックがオープンかクローズドかを示し、すなわち、このフラグは、パケットが送信されるときにソースブロックのサイズが決定されていない(オープンソースブロック)場合は0に設定され、パケットが送信されるときにソースブロックのサイズが決定されている(クローズドソースブロック)場合は1に設定され、SBL(1318)の残りの部分は、ソースブロック中のソースシンボルの数を提供する。

【０１３２】

送信機のフィードバック論理ユニット(1420)は、ソースブロックの最後のソースシンボルを搬送するパケットにソースブロックが閉じ込められていることを示すようにこのSBLフラグを設定することができ、ソースブロックに対する修復シンボルを搬送するすべてのパケットにそのソースブロックが閉じ込められていることをこのSBLフラグによって示すことに加えて、かつ、ソースブロックがクローズドであることをフラグが示す各パケットにおいて、SBLサイズは、クローズドソースブロック中のソースシンボルの実際の数に設定される。ESI(1320)は、SBN(1315)によって特定されたソースブロックに対してどの符号化シンボル(1325)がこのパケット中で搬送されるかを特定し、したがって、ESI(1320)はSBN(1315)によって調査される。新たなデータパケットが特定のパスフローに沿って送信されることになるたびに、そのパスフローのFIDがパケット中に配置され、そのFIDのSEQNが1だけインクリメントされてそのパケット中に配置され、符号化シンボルが送信されるべきアクティブソースブロックのSBNがパケット中に配置され、符号化シンボルの対応するESIがパケット中に配置され、符号化シンボルはすべて、パケットが送信される前にパケット中に配置される。

【０１３３】

受信機の転送プロトコル(1225)は、送信機の転送プロトコル(1215)に送信されるべきフィードバックを生成する。可能な受信機フィードバック情報フォーマットが、図13の下部に示されている。示されるように、受信機の転送プロトコル(1225)は、各FID(1350(1)、1350(2))に対して、そのFIDに対して受信された対応する最大のSEQN(1355(1)、1355(2))を報告する。加えて、受信機の転送プロトコル(1225)は、各アクティブソースブロックに対して、そのソースブロックに対してこれまでに受信された符号化シンボル(1365(1)、1365(2))の数とともに、そのソースブロックのSBN(1360(1)、1360(2))を報告する。

【０１３４】

また、受信機の転送プロトコル(1225)は、最小のアクティブソースブロックのソースブロック番号(1370)を報告する。受信機の転送プロトコル(1225)によって報告される最小のアクティブソースブロック番号(1370)は一般に、最小のソースブロック番号を伴う現在のアクティブソースブロックが受信機の転送プロトコル(1225)によって十分な確度で復元可能であると考えられ、かつさらなる符号化シンボルが現在のソースブロックに対して必要とされないので、このソースブロックが受信機の転送プロトコル(1225)によって非アクティブであると指定されるときに、増やされる。十分な符号化シンボルが十分にタイムリーに受信されない状況では、受信機の転送プロトコルによってまだ復元されていない最小のソースブロック番号を伴うソースブロックがあるときであっても、受信機の転送プロトコル(1225)が最小のアクティブソースブロック番号(1370)を増やすことも可能である。

【０１３５】

本開示を読めば、他の変形形態が可能であることは明らかであろう。システムは、送信機の転送プロトコル(1215)が、図13の上部に示されるような送信機のマルチパスデータパケットフォーマットで、最小のアクティブソースブロック番号をシグナリングすることを可能にするように、増強され得る。たとえば、「最小のアクティブソースブロック番号」という追加のパラメータが、送信機の転送プロトコル(1215)から受信機の転送プロトコル(1225)へのこのシグナリングを可能にするために、図13の上部に示されるようなマルチパスデータパケットフォーマットに追加され得る。この機能は次いで、送信機の転送プロトコル(1215)が、システムのエンドツーエンドのレイテンシの要件を満たす方法でソースブロックの送信および復元を完了することが可能ではない可能性のあるソースブロックを飛ばしてシグナリングすることを、可能にする。

【０１３６】

上記の多くの変形形態がある。たとえば、どのソースデータがパケットペイロードで搬送されるかを示すために、ESIの代わりにバイト範囲が使用され得る。別の例として、パケットペイロードで送信される任意のFECデータを生成するためにどのソースデータが使用されるかを示すために、SBNおよびSBLの代わりに、どの特定のデータがパケットペイロードで送信されるかを示すためのESIとともに、データのストリーム全体の中でのブロックのバイト範囲が使用され得る。別の変形形態として、異なるブロックからの符号化シンボルは、同じデータパケットの中に含まれてよく、SBN、SBL、およびESI(またはそれらの等価物)の、複数の3つ1組が、データパケットで搬送される符号化シンボルを特定するために、パケットヘッダに含まれ得る。たとえば、各アクティブソースブロックに対するパケットで搬送されるシンボルの比率は、各々のそのようなソースブロックに対して現在送信され得るシンボルの数に比例するように選ばれてよく、すなわち、アクティブブロックiに対して送信されるべきパケット中のシンボルの比率は、L_i+X_i-U_iの現在の値に比例するように選ばれる。

【０１３７】

図14は、マルチパスストリーミング送信機のブロック図の一部をより詳細に示す。図1の送信機エンドシステム100のいずれかまたはすべてが、図14のマルチパスストリーミング送信機と同様のコンポーネントを含み得る。図14に示されるように、ビデオ生成器(1205)によって生成されるビデオストリーム(1210)は、送信機の転送プロトコル(1215)内のソースデータバッファ(1405)内に一時的に記憶される。FECエンコーダ(1410)が、すでに形成されているソースブロックに対するFEC修復シンボルを生成し、得られたFEC修復シンボ

ルを、それらが送信のために必要とされるまで修復シンボルバッファ(1415)に配置する。FECエンコーダ(1410)は、必要に応じて動作することができ、アクティブソースブロックに対する修復シンボルを、それらが送信のために必要とされるときに生成する。あるいは、FECエンコーダ(1410)は、アクティブソースブロックブロックに対するいくつかの修復シンボルが必要とされるとすぐに送信される準備ができているように、かつ、FECエンコーダ(1410)に対する呼出しのオーバーヘッドを減らすために、それらの修復シンボルを事前に生成することができる。

【０１３８】

ソースブロックに対する追加の修復シンボルをFECエンコーダ(1410)に生成させることは、追加の符号化シンボルを送信することが可能になるたびにどの追加の符号化シンボルを送信するかを決定する、送信機のフィードバック論理ユニット(1420)からの信号によって引き起こされ得る。したがって、これらのステップは、サーバデバイスへのブロックに対して必要とされる追加の符号化単位の数を計算するステップと、サーバデバイスのような送信デバイスに、その数の追加の符号化単位を表すデータを送信するステップとを含む、方法のステップの例を表す。

【０１３９】

ソースブロックは、その開始側の境界と終了側の境界の両方が決定されているときにクローズドであると見なされ、すなわち、ソースブロック内のデータの範囲は、ソースブロックがクローズドであり、データのビデオストリームのコンテキスト内に開始バイトインデックスおよび終了バイトインデックスを有するときに、決定されている。たとえば、第1のソースブロックはビデオデータストリーム内のバイトインデックス0で開始し、バイトインデックス4432で終了することがあり、この場合、インデックス0から4431のバイトインデックスを含むソースブロック内の4432バイトのデータがある。この例を続けると、第2のソースブロックはバイトインデックス4432で開始するが、その終了バイトインデックスは、何らかの後の時点までソースブロック生成器ユニット(1425)によって決定されることが不可能であり、第2のソースブロックに対する終了バイトインデックスが決定されるまで、第2のソースブロックはオープンであると見なされる。

【０１４０】

したがって、一般に、ビデオデータストリームは、決定されるプロセスの中にある多くとも1つのオープンソースブロックが後に続く、クローズドソースブロックのシーケンスであると考えられ得る。さらに、ソースブロックのシーケンスは、1つまたは複数のアクティブソースブロック(ビデオストリーム全体の配信が成功したときの配信の最後は除く)が後に続く0個以上の非アクティブソースブロックを含み、非アクティブソースブロックは、受信機の転送プロトコル(1225)への配信に成功し、受信機の転送プロトコル(1225)から送信されたフィードバックに基づいて送信機の転送プロトコル(1215)への配信が成功したと確認応答されたものであり、または、配信するには遅すぎると見なされ、したがってもはや受信機において復元されるために必要とされないソースブロックである。ソースブロック生成器ユニット(1425)は、最新のアクティブソースブロックをいつ閉じ、それによって新たなアクティブオープンソースブロックを開始するかを決定する。系統的なFEC符号が使用されるとき、すなわち、ソースブロックのソースシンボルが、FECデコーダによってソースブロックを復元するために使用され得る符号化シンボルの中にあるとき、アクティブオープンソースブロックに対する符号化シンボル、特にソースシンボルの送信を可能にすることが、可能であり好ましい。

【０１４１】

多くのよく知られているFEC符号、たとえば、IETF RFC 5510で規定されるようなリードソロモン符号、またはIETF RFC 6330で規定されるようなRaptorQ符号、またはIETF RFC 5053で規定されるRaptor符号は、系統的である。アクティブオープンソースブロックに対するソースシンボルを送信することは、ビデオストリームの配信のエンドツーエンドのレイテンシを低減でき、さらに、同じエンドツーエンドのレイテンシの許容範囲内でより高品質で信頼性のある配信を実現できるので、好ましい。利点がある1つの理由は、ソース

ブロック全体が利用可能になる前、またはそのサイズが知られる前に、ソースブロックの配信が開始できるからである。アクティブオープンソースブロックがソースブロック生成器ユニット(1425)によって閉じられると、このソースブロックに対する修復シンボルは、FECエンコーダ(1410)によって生成され修復シンボルバッファ(1415)に記憶されてよく、このソースブロックに対する追加の符号化シンボルが送信機のフィードバック論理ユニット(1420)の方法に従って送信されるべきであるとき、送信され得る。

【０１４２】

あるいは、修復シンボルバッファ(1415)はなくてよく、FECエンコーダ(1410)が、このソースブロックに対する追加の符号化シンボルが送信機のフィードバック論理ユニット(1420)の方法に従って送信されるべきであるとき、即刻の送信のために修復シンボルをオンザフライで生成することができる。図16は、非アクティブソースブロックとアクティブソースブロック、さらに、クローズドソースブロック(非アクティブソースブロックおよびアクティブソースブロックの混合)と多くとも1つのオープンソースブロック(これはアクティブである)を示す。

【０１４３】

ソースブロック生成器ユニット(1425)は、様々な方法を使用して、いつ現在のアクティブソースブロックを閉じて次のアクティブオープンソースブロックを開始するかを決定することができる。たとえば、ソースブロック生成器ユニット(1425)は、現在のアクティブオープンソースブロックに対する符号化シンボルを含む最初のパケットが送信された時点かその後に送信された、パケットが受信されたことを示す受信機のフィードバック論理ユニット(1525)からのフィードバックを受信したという、送信機のフィードバック論理ユニット(1420)からの情報を受信すると、現在のアクティブオープンソースブロックを閉じると決めることができる。上記の時点は、現在のアクティブオープンソースブロックからの符号化シンボルを含む最初のパケットが送信された時点でのフローの各々に対する現在のシーケンス番号を記録し、次いで、あるフローに対する、上記の記録の時点でのそのフローの現在のシーケンス番号と少なくとも同等である最大のシーケンス番号を伴う、受信機のフィードバック論理ユニット(1525)からのフィードバックを送信機のフィードバック論理ユニット(1420)が受信するとすぐに、ソースブロック生成器ユニット(1425)への指示が提供されるべきであると決定することによって、送信機のフィードバック論理ユニット(1420)によって決定され得る。

【０１４４】

この方法を使用すると、現在のアクティブオープンソースブロックが閉じられてソースブロックのサイズが決定されるときの、現在のアクティブオープンソースブロックのサイズは、概ねデータのRTT量である。あるいは、ソースブロック生成器ユニット(1425)は、ある一定の長さの時間の後、たとえば、以前のソースブロックが閉じられてから1秒後に、現在のアクティブオープンソースブロックを閉じると決定することができる。この場合、送信レートが可変であれば、各ソースブロックは異なるサイズである可能性が高いが、送信レートが一定であれば、ソースブロックは概ね等しいサイズである可能性が高い。別の代替形態として、現在のアクティブオープンソースブロックは、オープンソースブロックのサイズが所定のサイズ、たとえば100000バイトに達するとすぐに、ソースブロック生成器ユニット(1425)によって閉じられ得る。

【０１４５】

他の代替形態として、ソースブロック生成器ユニット(1425)は、上の方法の組合せを使用して、現在のアクティブオープンソースブロックを閉じることができ、たとえば、ソースブロックのサイズが所定のサイズに達するとすぐに、または、以前のソースブロックが閉じられてから所定の長さの時間が経過するまでに、これらのうちのいずれが先に発生しても、ソースブロックを閉じることができる。別の例として、ソースブロック生成器ユニット(1425)は、そのソースブロックに対するフィードバックの指示が最初に送信機のフィードバック論理ユニット(1420)からソースブロック生成器ユニット(1425)へ示されるとき、または、以前のソースブロックを閉じてから一定の長さの時間が経過した後で、これら

のうちのいずれが先に発生しても、現在のアクティブオープンソースブロックを閉じることができる。

【０１４６】

図14の送信機のフィードバック論理ユニット(1420)は、図13の下部に示されるフォーマットで与えられる、図15の受信機のフィードバック論理ユニット(1525)から受信されたフィードバックを処理する。送信機のフィードバック論理ユニット(1420)は、受信された最小のアクティブSBN(1370)に基づいて、アクティブソースブロックのセットを更新する。送信機のフィードバック論理ユニット(1420)は、ビデオストリームに対する次のデータパケットをいつ送信するか、次のデータパケットをどのパスフローに送信するか、および、どのアクティブソースブロック(または複数のソースブロック)からの符号化シンボルをデータパケット内で送信するかを決定する。送信機のフィードバック論理ユニット(1420)は、次のように、どのアクティブソースブロックから次の符号化シンボルを送信するかを決定することができる。

【０１４７】

以前に使用されたものと同様の表記を使用して、SBN=Iを伴う各アクティブソースブロックに対して、BIを、ある所望のレベルの確度でソースブロックIを復元するために受信される必要のある符号化シンボルの数とする。たとえば、リードソロモンFEC符号、たとえばIETF RFC 5510に記載されるようなものを使用すると、BIの値は、ソースブロックのソースシンボルの数と等しくてよく、ソースブロック全体の復元は完全な確度を伴い、一方、他の符号、たとえばIETF RFC 6330に記載されるRaptorQ符号では、BIの値は、かなりの確度でソースブロックのソースシンボルの数と等しくてよく、より大きなBIの値は確度の向上を可能にする。

【０１４８】

送信機のフィードバック論理ユニット(1420)は、使用されているFEC符号の特性、およびソースブロックI中のソースシンボルの数に基づいて、BIの値を計算することができる。送信機のフィードバック論理ユニット(1420)は、ソースブロックIに対して受信機のフィードバック論理ユニット(1525)から送信機のフィードバック論理ユニット(1420)が受信した、受信された符号化シンボル(1365)の数の最大値としてRIを計算することができる。送信機のフィードバック論理ユニット(1420)は、LI=BI-RIを計算することができ、これは、ある規定されたレベルの確度でソースブロックIを復元するために、受信機によって受信されていると送信機が知っている数に加えて受信機が受信しなければならない、追加の符号化シンボルの数である。

【０１４９】

UIを、ソースブロックIに対して送信されたが、確認応答が受信機から送信機においてまだ受信されていない、符号化シンボルの数とする。送信機のフィードバック論理ユニット(1420)は、受信機のフィードバック論理ユニット(1525)から受信されたフィードバックに基づいて、次のようにUIを計算することができる。送信機のフィードバック論理ユニット(1420)は、各フローIDの値Jに対して、送信機によってFID=Jに対して送信されている現在のシーケンス番号Cから、図13の下部に示されるフィードバック情報フォーマットで受信機のフィードバック論理ユニット(1525)から送信機のフィードバック論理ユニット(1420)が受信したFID=Jに対する最大のシーケンス番号Sまでの間である、FID=Jに対するシーケンス番号の範囲の中にあるソースブロックIに対して送信された符号化シンボルの数を決定することができる。送信機のフィードバック論理ユニット(1420)は、FID=Jに対するSからCまでの範囲にある各シーケンス番号Kに対して、FID=JおよびシーケンスKを伴うパケットでソースブロックIに対する符号化シンボルがどれだけ搬送されたかを記録することによって、この計算を行うことができる。

【０１５０】

これに基づいて、送信機のフィードバック論理ユニット(1420)は、シーケンス番号S+1からC-1の範囲内にある、パスフローJに送信されたソースブロックIに対するそのような符号化シンボルの数を加算することができる。次いで、送信機のフィードバック論理ユニ

ット(1420)は、異なるパスフローにわたってこれらの量を加算して、全体でどれだけの符号化シンボルUIがソースブロックIに対して送信されたがまだ確認応答されていないかを決定することができる。各フローまたはパスに対するフロー識別子およびフローシーケンス番号を使用して、データパケットの中およびフィードバック情報の中で提供される情報、および本明細書で説明される方法は、送信機が、各パスに対する送信されたがまだ確認応答されていない(失われたか受信されたかのいずれかの)符号化シンボルの数を正確に計算することを可能にし、したがって、送信機が、すべてのパスにわたって、送信されたがまだ確認応答されていない符号化シンボルの総数を正確に推定することを可能にすることに、留意されたい。パケットがパスに送信される順序と、そのパケットが(失われなければ)パスから受信される順序との間にほとんど差がなければ、送信機の推定の精度は高く、これは一般に当てはまる。前に言及されたように、パケットが送信されるパスを考慮しない、複数のパスを通じて送信されるパケットの全体的な送信順序および受信順序は、大きく異なり得る。したがって、パスごとの情報およびフィードバックを提供して使用することの利点の1つは、送信機が、冗長なデータの送信を最小にして、ストリームのブロックの復元のエンドツーエンドのレイテンシを最小にするために、どれだけのデータを総計で送信するかをより正確に推定することが可能になることである。したがって、送信機のフィードバック論理によって受信されるデータは、パケットロスと変化するデータスループットとパスのレイテンシとを経験し得る複数の並列ネットワークパスを通じてデータが信頼性をもってストリーミングされるべきであるとき、パス固有の情報を追跡し、報告し、使用することの例を表す。

【０１５１】

前のように、XIを、ソースブロックIに対して前もって送信され得ると、送信機のフィードバック論理ユニット(1420)が決定した符号化シンボルの数とする。送信機は、規定された規則に基づいてXIの値を計算することができ、たとえば、XIはBIの何らかの固定された部分であり、たとえばXI=0.05*BIであり、または前に説明されたものと同様の他の規則に基づいてXIの値を計算することができる。次いで、送信機のフィードバック論理ユニット(1420)は、LI+XI-UI>0である場合、別の符号化シンボルがアクティブクローズドソースブロックに対して送信され得ると決定する。

【０１５２】

データレート調整器ユニット(1430)は、各フローに対して、次のデータパケットがそのフローにいつ送信され得るかを決定する。データレート調整器ユニット(1430)は、各送信機(1220(1)、1220(2)など)と通信してこの決定を行う。たとえば、UDPデータパケットが送信される場合、送信機(1220(1))は、Linux(登録商標)またはいくつかの他のUnix(登録商標)のようなオペレーティングシステムが使用される場合、TIOCOUTQ ioctl()を伴うUDP送信キューのサイズを決定することができる。送信キューのサイズを監視することによって、送信機(1220(1))は、送信機(1220(1))における過剰なバッファリングを回避し、送信機(1220(1))の内部送信機キューが少ないときだけ新たな出力パケットをキューに入れることができ、したがって、送信機(1220(1))が送信されないあまりにも多くのデータを蓄積するのを回避する。送信機(1220(1))の送信キューが空ではないが少ない状態に保たれる場合、完全な送信スループットが達成され得る。したがって、各フローに対する各送信機(1220(1))は、送信のために別のデータパケットをいつ受け入れられるかを、データレート調整器ユニット(1430)に示し、受け入れられる時点で、データレート調整器ユニット(1430)は、データパケットが送信機(1220(1))と関連付けられるフローに送信され得ると決定する。データレート調整器ユニット(1430)は、可能な送信機(1220(1)、1220(2)など)の各々から、上記の方法を使用して指示を受信し、各々のあり得るフローに対して、そのフローに対する次のデータパケットがいつ送信され得るかを決定することができる。

【０１５３】

あるいは、または加えて、送信機(1220(1))は、SO_SNDBUF値を使用して、送信機のウィンドウサイズを十分に小さな値に設定することができる。送信機(1220(1))は次いで、たとえばselect()システムコールまたはpoll()システムコールを使用することによって、UD

Pソケットが書き込み可能になるのを待ち、送信機(1220(1))と関連付けられるフローに対する別のデータパケットを送信することがいつ可能になるかを決定することができる。こうすることで、送信機の送信キューのサイズは、定期的にポーリングされなくてよい。

【０１５４】

すべての送信機(1220(1)、1220(2)など)が、送信機と関連付けられるフローに別のデータパケットを送信するための容量を有することを送信機が示したとき、各送信機(1220(1)、1220(2)など)にどのデータパケットを送信するかを決定するデータレート調整器ユニット(1430)と通信していてよい。送信機(1220(1)、1220(2)など)とデータレート調整器ユニット(1430)との間の通信は、高帯域幅、低レイテンシ、および低パケットロスを伴うローカルネットワークで発生していることがあるので、通信がTCPを使用する場合、その通信は完全に満足のいくものであり得る。そのような設定では、送信機は次いで、次のループを実行する。

以下を無期限に繰り返す。

1. (TIOCOUTQを監視することと、低送信バッファおよび上で説明されたようなselect()を使用することのいずれか、または両方によって)送信機の送信キューが少なくなるまで待つ。

2. 新たなパケットに対する要求をデータレート調整器ユニット(1430)に送信する。(たとえば、データレート調整器ユニット(1430)へのsend()またはwrite())。

3. 送信すべきデータパケットを含む、データレート調整器ユニット(1430)からの応答を待つ。(たとえば、select()を使用する)

4. (たとえば、send()システムコールまたはwrite()システムコールを使用することによって)UDPを通じてデータパケットをネットワーク(1222)に送信する。

データレート調整器ユニット(1430)は次のことを行う。

以下を無期限に繰り返す。

1. (たとえば、select()を使用して)任意の送信機からの要求を待つ。

2. 送信すべき新たなデータパケットを要求した各送信機に対して、送信すべき新たなデータパケットを構築し、それを送信機に提供する。

【０１５５】

データレート調整器ユニット(1430)はまた、生成されているビデオストリームデータレートを上げ、生成されているビデオストリームデータレートを下げ、または、ビデオデータレートを同じに保つために、情報をビデオ生成器(1205)に提供することができ、この情報はたとえば、ソースデータバッファ(1405)中のデータの量と、データレート調整器ユニット(1430)がデータパケットを様々なフローに沿って送信している全体としてのレートとに基づいてよい。

【０１５６】

送信機のフィードバック論理ユニット(1420)は、送信されるのに利用可能でありまだ送信されていないソースブロックIの少なくとも1つのソースシンボルがある場合、別の符号化シンボルがアクティブオープンソースブロックIに対して送信され得ると決定する。送信機のフィードバック論理ユニット(1420)は、符号化シンボルを搬送する次のデータパケットが送信されるべきであることをデータレート調整器ユニット(1430)が示す時点で論理的に利用可能なすべてのフィードバックおよび送信情報を使用して、上の計算のすべてを実行する(しかし、送信機のフィードバック論理ユニット(1420)は、次のデータパケットが送信されるべきときより前の任意の時点で、計算のいくつかまたはすべてを行うことができる)。次のデータパケットが特定のフローに送信されるべきであることをデータレート調整器ユニット(1430)が示すとき、送信機のフィードバック論理ユニット(1420)は、Iが、上で説明されたような決定されたその時点で送信機のフィードバック論理ユニット(1420)によって符号化シンボルが送信され得る、すべてのアクティブソースブロックの中の最小のソースブロック番号となるように、アクティブソースブロックを決定し、ソースブロックIに対する1つまたは複数の符号化シンボルは、次のデータパケットの中に配置され、次のデータパケットはそのフローに送信される。

【０１５７】
図14の様々な論理ブロックのいずれかまたはすべては、ハードウェア、ソフトウェア、ファームウェア、またはこれらの組合せで実装され得る。ソフトウェアまたはファームウェアで実装されるとき、必要なハードウェアも提供されてよく、たとえば、1つまたは複数のコンピュータ可読媒体が説明される機能を実行するための命令を含み、命令を実行するための1つまたは複数の処理ユニットを含むことを、理解されたい。
【０１５８】
この方式で、図14は、並列ネットワークパスの第1のパスを通じて第1のブロックに対する第1の符号化単位を送信し、第1の符号化単位を送信した後で、第1のパスを通じて第2のブロックに対する第2の符号化単位を送信し、第2の符号化単位を送信した後で、第1のパスを通じて第1のブロックに対する第3の符号化単位を送信するように構成される、1つまたは複数のプロセッサを含むデバイスの例を表す。
【０１５９】
図15は、例示的なマルチパスストリーミング受信機をより詳細に示すブロック図である。図1の受信機エンドシステム160のいずれかまたはすべてが、図15のマルチパスストリーミング受信機と同様のコンポーネントを含み得る。フロー1に対する送信機(1220(1))によって送信されるパケットは、フロー1に対する対応する受信機(1505(1))によって受信され(それらのパケットが送信と受信の間に失われなければ)、他のフローに対しても同様である。すべての受信されるパケットは、受信データバッファ(1505)へと受信機の転送プロトコル(1225)によって集約される。FECデコーダ(1510)は、ソースブロックを復元するのに十分な符号化シンボルが受信されている、アクティブソースブロックを復元するために実行され、復元されたソースブロックは、復元されたソースデータバッファ(1520)へと、好ましくは、復元されたビデオストリーム(1230)が元のビデオストリーム(1210)と同じ順序となるような昇順のソースブロック番号の順序で、配置される。
【０１６０】
受信機のフィードバック論理ユニット(1525)は、受信データバッファ(1505)中の受信されたパケットを監視し、対応する送信機のフィードバック論理ユニット(1420)に送信されるフィードバック情報を、たとえば図13の下部に示されるフォーマットで生成する。受信機のフィードバック論理ユニット(1525)はまた、ソースブロックがいつ復元され得るかを、そのソースブロックに対して受信されるすべてのデータパケットの中で受信された最大のSBL(1318)の値に基づいて、かつ、ソースブロックがオープンかクローズドかを示すSBLフラグにも場合によっては基づいて決定することができるので、受信機のフィードバック論理ユニット(1525)は、ソースブロックが復元され得ると判定されるときにFECデコーダ(1510)を呼び出すことができる。一般に、受信機のフィードバック論理ユニット(1525)は、たとえば、ソースブロックがクローズドであることを示すSBLフラグがもしあればそれによって決定されるような、または、ソースブロックがクローズドであるという暗黙的な指示を提供するソースブロックに対する修復シンボルを搬送するパケットを受信することによって決定されるような、ソースブロックがクローズドであることの指示を受信機のフィードバック論理ユニットが受信していない場合、ソースブロックが復元可能であると宣言しないことが好ましい。
【０１６１】
ソースブロックが復元され得るとき、または、たとえばソースブロックがエンドツーエンドの時間的な制約が原因で省略されるべきであるとき、受信機のフィードバック論理ユニット(1525)は、送信機のフィードバック論理ユニット(1420)に提供されるフィードバック情報中の最小のアクティブSBN(1370)を、適度により高いSBN値にリセットすることができる。受信機のフィードバック論理ユニット(1525)は、各アクティブソースブロックに対して、そのソースブロックに対する受信された符号化シンボルの数を決定する。受信機のフィードバック論理ユニット(1525)は、各フローに対して、そのフローに対して受信された最大のシーケンス番号を決定する。この情報のすべては、たとえば、図13の下部で与えられる受信機のマルチパスフィードバック情報フォーマットを使用して、受信機のフィー

ドバック論理ユニット(1525)から送信機のフィードバック論理ユニット(1420)へと継続的に送信される。

【０１６２】

図15の様々な論理ブロックのいずれかまたはすべては、ハードウェア、ソフトウェア、ファームウェア、またはこれらの組合せで実装され得る。ソフトウェアまたはファームウェアで実装されるとき、必要なハードウェアも提供されてよく、たとえば、1つまたは複数のコンピュータ可読媒体が説明される機能を実行するための命令を含み、命令を実行するための1つまたは複数の処理ユニットを含むことを、理解されたい。

【０１６３】

この方式で、図15は、並列ネットワークパスの第1のパスを通じて第1のブロックに対する第1の符号化単位を受信し、第1の符号化単位を受信した後で、第1のパスを通じて第2のブロックに対する第2の符号化単位を受信し、第2の符号化単位を受信した後で、第1のパスを通じて第1のブロックに対する第3の符号化単位を受信するように構成される、1つまたは複数のプロセッサを含むデバイスの例を表す。

【０１６４】

このようにして、図15は、複数の並列ネットワークパスを介して前方誤り訂正されたデータをサーバデバイスから受信し、ネットワークパスの各々でのデータのロスを決定し、ネットワークパスの各々でのデータのロスを表すデータをサーバデバイスに送信するように構成される、1つまたは複数のプロセッサを含むデバイスの例を表す。

【０１６５】

例に応じて、本明細書で説明される技法のいずれかのいくつかの行為またはイベントは異なる順序で実行されてもよく、一緒に追加され、統合され、または省略されてもよい(たとえば、説明される行為またはイベントのすべてが技法の実施のために必要とは限らない)ことを認識されたい。さらに、いくつかの例では、行為またはイベントは、順次的にではなく、たとえばマルチスレッド処理、割り込み処理またはマルチプロセッサを通じて同時に実行され得る。

【０１６６】

1つまたは複数の例では、説明される機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、1つまたは複数の命令またはコードとしてコンピュータ可読媒体上に記憶され、あるいはコンピュータ可読媒体を介して送信されてよく、かつハードウェアに基づく処理ユニットによって実行されてよい。コンピュータ可読媒体は、データ記憶媒体のような有形媒体、または、たとえば通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を支援する任意の媒体を含む通信媒体に相当する、コンピュータ可読記憶媒体を含み得る。このようにして、コンピュータ可読媒体は一般に、(1)非一時的な有形コンピュータ可読記憶媒体または(2)信号波もしくは搬送波のような通信媒体に相当し得る。データ記憶媒体は、本開示で説明される技法を実装するための、命令、コード、および/またはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であってよい。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【０１６７】

限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、フラッシュメモリ、または、命令もしくはデータ構造の形態の所望のプログラムコードを記憶するために使用され、コンピュータによってアクセスされ得る任意の他の媒体を含み得る。また、当然、あらゆる接続がコンピュータ可読媒体と呼ばれる。たとえば、命令が、同軸ケーブル、光ファイバケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波な

どのワイヤレス技術は、媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的な媒体を含まず、代わりに非一時的な有形記憶媒体を指すことを理解されたい。本明細書で使用される場合、ディスク(disk)およびディスク(disc)は、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびブルーレイディスクを含み、ディスク(disk)は通常、磁気的にデータを再生し、ディスク(disc)はレーザーで光学的にデータを再生する。前述の組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。

【０１６８】

命令は、1つまたは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、または他の等価の集積論理回路もしくはディスクリート論理回路のような、1つまたは複数のプロセッサによって実行され得る。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造、または、本明細書で説明される技法の実装に適した任意の他の構造の、いずれをも指し得る。加えて、いくつかの態様では、本明細書で説明される機能は、符号化および復号のために構成された、専用のハードウェアモジュールおよび/またはソフトウェアモジュール内で提供されてよく、または、組み合わされたコーデックに組み込まれてよい。また、技法は、1つまたは複数の回路素子または論理素子において完全に実装されてもよい。

【０１６９】

本開示の技法は、ワイヤレスハンドセット、集積回路(IC)、またはICのセット(たとえば、チップセット)を含む、多様なデバイスまたは装置において実装され得る。様々なコンポーネント、モジュール、またはユニットが、開示される技法を実行するように構成されるデバイスの機能的な側面を強調するために、本開示において説明されるが、必ずしも異なるハードウェアユニットによる実現を必要としない。むしろ上で説明されたように、様々なユニットは、コーデックハードウェアユニットへと組み合わされてよく、または、適切なソフトウェアおよび/またはファームウェアとともに、上で説明されたような1つまたは複数のプロセッサを含む相互動作可能なハードウェアユニットの集合によって与えられてよい。

【０１７０】

様々な例が説明されてきた。これらの例および他の例は、以下の特許請求の範囲内に入る。

【符号の説明】

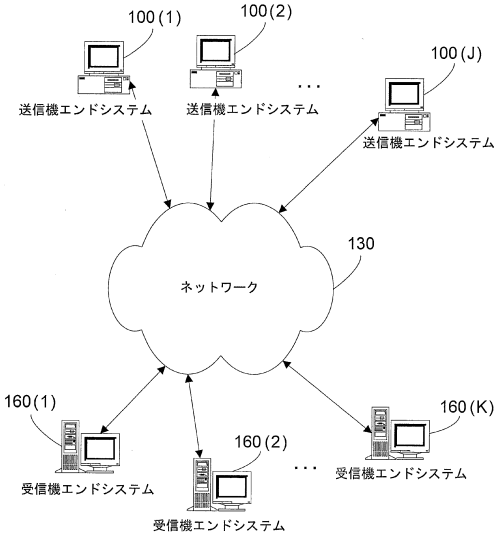
【０１７１】

100(1)　送信機エンドシステム
100(2)　送信機エンドシステム
100(J)　送信機エンドシステム
130　ネットワーク
160(1)　受信機エンドシステム
160(2)　受信機エンドシステム
160(K)　受信機エンドシステム
200　データ
210　送信機の転送プロトコル
220　送信機の信頼性制御プロトコル
230　送信機のレート制御プロトコル
240　データパケット
250　フィードバックパケット
260　ネットワーク
270　受信機のレート制御プロトコル
280　受信機の信頼性制御プロトコル

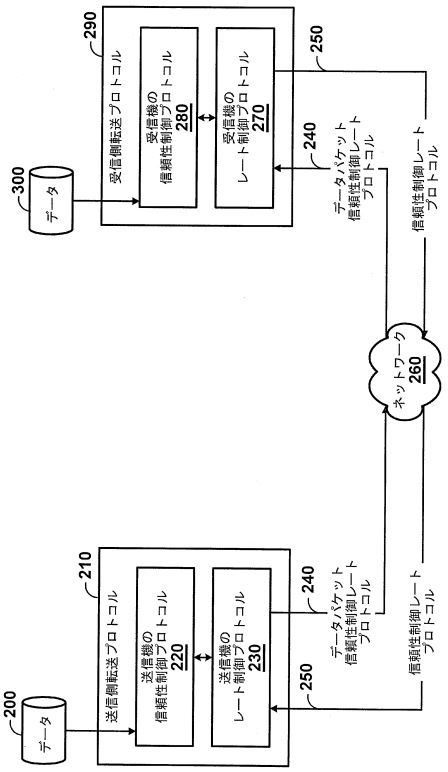
290 受信機の転送プロトコル
300 データ
310 送信機の信頼性制御論理
320 FECエンコーダ
330 データブロック
340 符号化単位および信頼性制御情報
350 信頼性制御情報
410 受信機の信頼性制御論理
420 FECデコーダ
430 データブロック
510 ブロック番号
520 符号化単位ID
530 符号化単位
540 ブロック番号
550 必要とされる符号化単位
810 第1のアクティブブロックAB 1
820 第2のアクティブブロックAB 2
830(1) AB1
830(2) AB1
830(3) AB1
830(4) AB1
830(5) AB2
830(6) AB2
830(7) AB1
830(8) AB2
830(9) AB1
910 ブロック番号
920 シーケンス番号
930 符号化単位ID
940 符号化単位
950(1) ブロック番号
950(2) ブロック番号
960(1) 必要とされる符号化単位
960(2) 必要とされる符号化単位
970(1) 受信された最大のシーケンス番号
970(2) 受信された最大のシーケンス番号
1205 ビデオ生成器
1210 ビデオストリーム
1215 送信機の転送プロトコル
1220(1) フロー1に対する送信機
1220(2) フロー2に対する送信機
1225 受信機の転送プロトコル
1230 ビデオストリーム
1305 FID
1310 FIDに対するSEQN
1315 SBN
1318 SBL
1320 ESI
1325 符号化シンボル
1350(1) FID 1

1350(2) FID 2
1355(1) FID1に対する最大のSEQN
1355(2) FID2に対する最大のSEQN
1360(1) SBN 1
1360(2) SBN 2
1365(1) SBN 1に対する受信された符号化シンボル
1365(1) SBN 2に対する受信された符号化シンボル
1370 最小のアクティブSBN
1405 ソースデータバッファ
1410 FECエンコーダ
1415 修復シンボルバッファ
1420 送信機のフィードバック論理ユニット
1425 ソースブロック生成器ユニット
1430 データレート調整器ユニット
1505 受信データバッファ
1505(1) フロー1に対する受信機
1505(2) フロー2に対する受信機
1510 FECデコーダ
1520 復元されたソースデータ

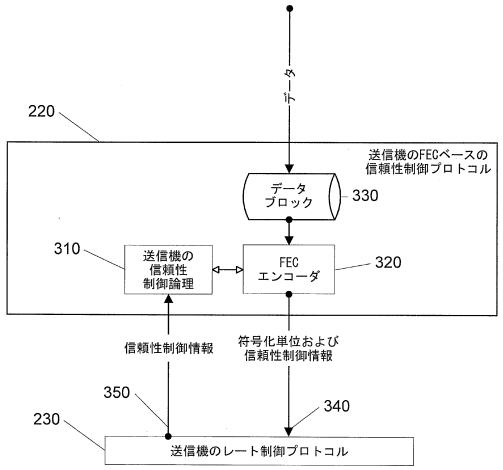
【図１】



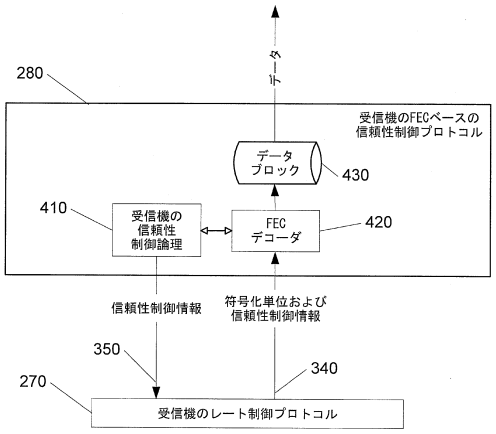
【図２】



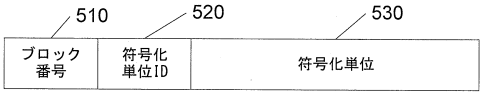
【図３】



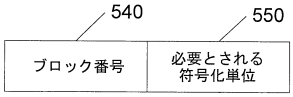
【図４】



【図５】

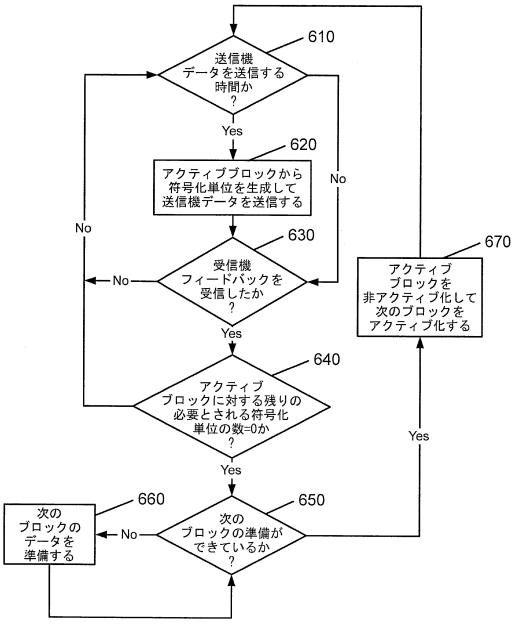


送信機データフォーマット

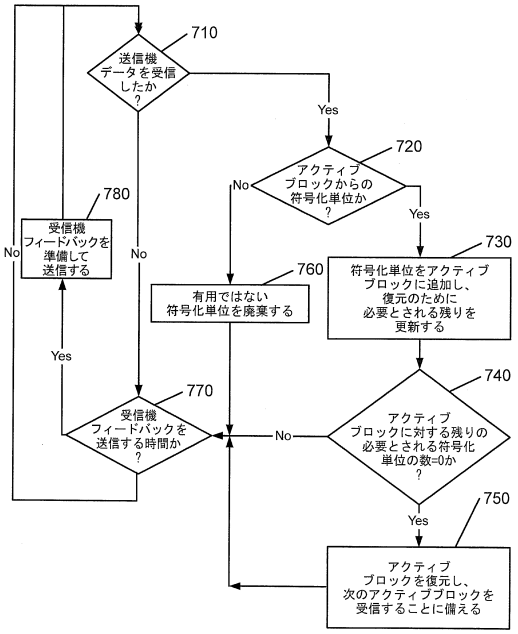


受信機フィードバックフォーマット

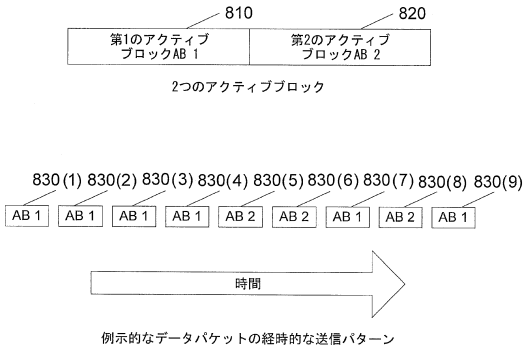
【図６】



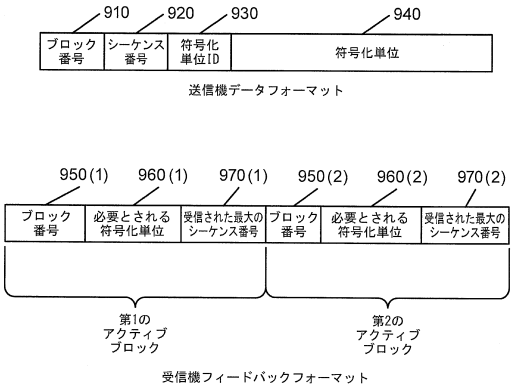
【図７】



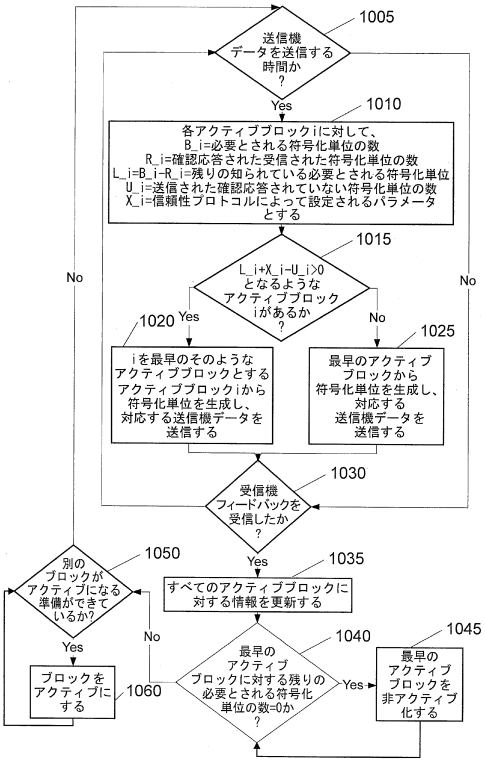
【図８】



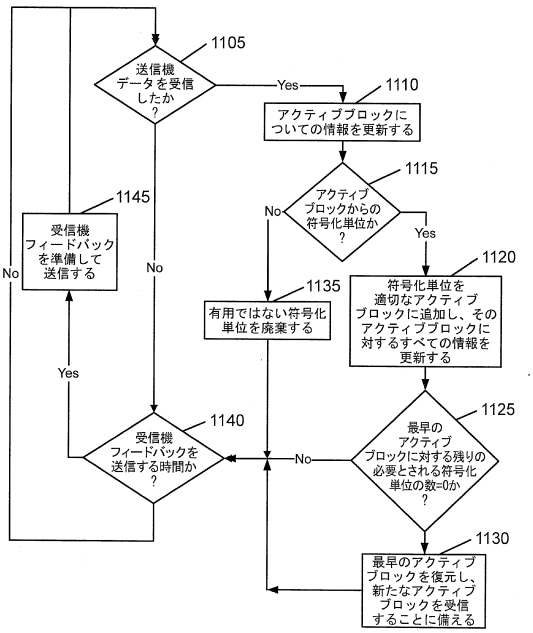
【図９】



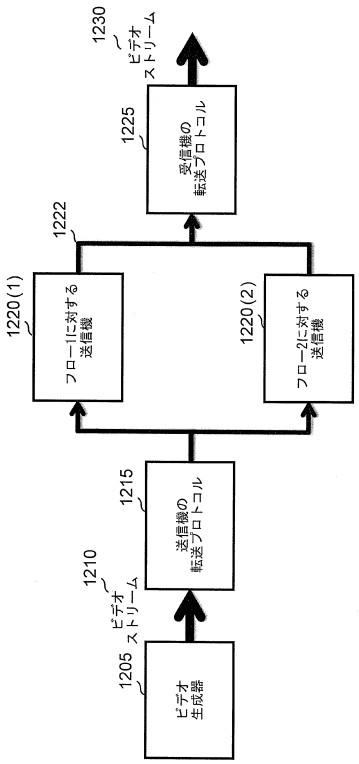
【図１０】



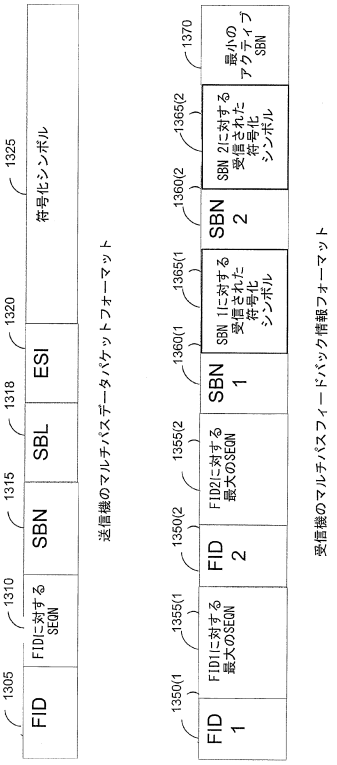
【図１１】



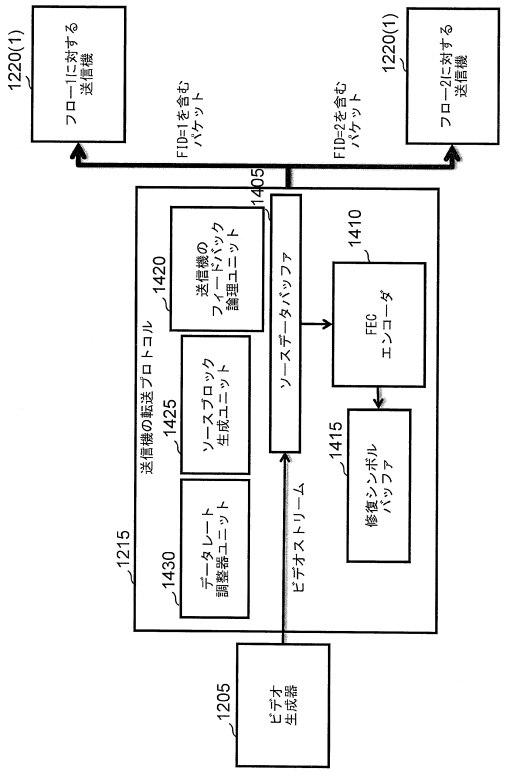
【図１２】



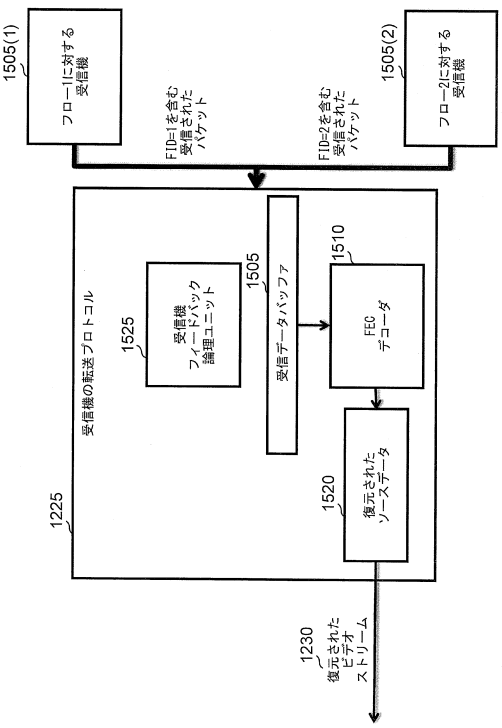
【図１３】



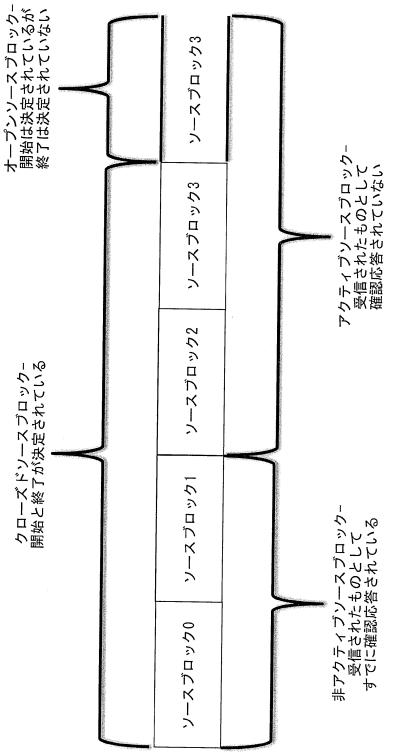
【図１４】



【図１５】



【図１６】


オープンソースブロック-
開始は決定されているが
終了は決定されていない
クローズドソースブロック-
開始と終了が決定されている
ソースブロック0
ソースブロック1
ソースブロック2
ソースブロック3
ソースブロック3
非アクティブソースブロック-
受信されたものとして
すでに確認応答されている
アクティブソースブロック-
受信されたものとして
確認応答されていない

フロントページの続き

(31)優先権主張番号　14/157,290
(32)優先日　平成26年1月16日(2014.1.16)
(33)優先権主張国　米国(US)

早期審査対象出願

(72)発明者　ロレンツ・クリストフ・ミンダー
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５

審査官　谷岡　佳彦

(56)参考文献　特表２００７－５０８７５０（ＪＰ，Ａ）
国際公開第２０１１／０７１４７２（WO，Ａ１）
Yong Cui,et al.，FMTCP:A Fountain Code-Based Multipath Transmission Control Protocol，Distributed Computing Systems(ICDCS),2012 IEEE 32nd International Conference on，２０１２年　6月，p.366-375
浜　崇之　他，動的ＦＥＣ－ＴＣＰの実装評価，電子情報通信学会２００８年総合大会講演論文集　通信2，２００８年，p.413，B-11-21
渡部　郁恵　他，複数経路を活用したＴＣＰ－Ｆｒｉｅｎｄｌｙなストリーミングシステムの設計と実装，情報処理学会研究報告，日本，社団法人情報処理学会，２００７年，Ｖｏｌ．２００７　Ｎｏ．３，p.37-42，２００７－ＱＡＩ－２２（８）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　１／００