社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－AVM－59 (12)
2007／12／14

複数TCPコネクションを利用したリアルタイム伝送方法

馬場 昌之　西川 博文　加藤 嘉明

三菱電機株式会社情報技術総合研究所 〒247-8501 神奈川県鎌倉市大船 5-1-1
E-mail: Baba.Masayuki@dy.MitsubishiElectric.co.jp

あらまし インターネットの普及と高速化に伴い，ビデオやオーディオなどのメディアデータをリアルタイムに IP 伝送するようになってきた．しかしながら利用するネットワーク環境により TCP しか使用できないこともある．本稿では，TCP を使用したリアルタイム伝送方法を提案する．既存の TCP を用いて，無線環境での使用にも耐えうるように，バーストパケットロスに対応できるようにパケット伝送方法を制御する．本提案方法では，複数の TCP コネクションを用意し，1パケットずつ異なるコネクションで送信する．これによりパケットロスが発生しても他のパケットに遅延を与えず，バーストロス時にもフロー制御によるレートの低下が発生しない．
キーワード TCP，複数 TCP，リアルタイム伝送，パケットロス，スループット

Real-time Transmission Method using Multiple TCP Connections

Masayuki BABA, Hirofumi NISHIKAWA, and Yoshiaki KATO

Information Technology R&D Center, Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan
E-mail: Baba.Masayuki@dy.MitsubishiElectric.co.jp

Abstract Real-time multimedia communication over UDP/IP has been put into practical use because of the popularization of the high-speed Internet. However there are some communication environments that can be used only TCP/IP. In this report, we propose a new real-time transmission method using general TCP/IP and dealing with some consecutive packet losses occurred in wireless communication environment. Our proposed method uses multiple TCP connections, and sends packets over different TCP connections. This method can recover some consecutive packet losses quickly and keep the throughput performance at a high level.
Key words TCP, Multiple TCP, Real-time transmission, Packet loss, Throughput

1. はじめに

インターネットの普及と高速化に伴い，ビデオやオーディオなどのマルチメディアデータをリアルタイムに IP 伝送するようになってきた．一般にリアルタイム伝送には UDP が用いられるが，利用するネットワーク環境により TCP しか使用できないこともある．

本稿では，TCP を使用したリアルタイム伝送方法を検討する．TCP は，順序制御，再送制御，フロー制御，ふくそう制御などにより信頼性のある通信を実現する．しかしこれらの制御ために，データ伝送におけるリアルタイム性は低下してしまうこともある．

一方，ウィンドウサイズを大きくしたり [1] [2]，複数の TCP コネクションを束ねてたり [3] することで，TCP のスループットを向上させ，同時にリアルタイム性を向上させることも可能である．しかしながら，スループットは長い時間における伝送量であり，必ずしも1つ1つのパケットの伝送遅延時間を短縮するとはかぎらない．特に無線を用いた TCP 通信時に，IP パケットのバーストロスが発生した場合，多くの IP パケットの再送のために遅延が増大する可能性がある．

本稿では，高スループットを保ちつつ，低遅延を確保する TCP リアルタイム伝送方法を提案する．そしてシミュレーションによりその有効性を検証する．

2. TCP によるリアルタイム通信

まず，リアルタイム通信に TCP を使用した場合の課題を明確にする．リアルタイム性が問題となるのは特にパケットロス時であるため，その時の TCP の動作を説明しながら，リアルタイム性の向上のための検討を行う．

2.1 TCP 動作テストモデル

図 1 に示すように，2台の PC(OS:WindowsXP) 間で TCP によるデータ伝送を行い，ネットワークエミュレータでパケッ

表 1　テスト時の設定

パケット送信間隔	10 msec
TCP パケットデータ長	1460 byte
平均データレート	1.170 Mbps
RTT	2 msec
ウィンドウサイズ	65535 byte

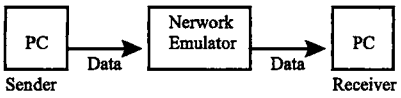

図 1　テストモデル

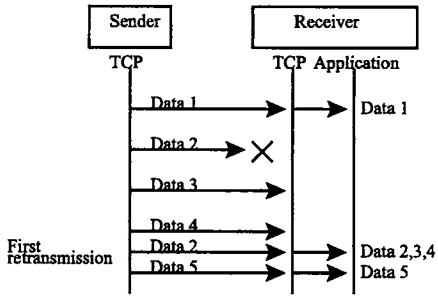

図 3　受信側アプリケーションのパケット受信タイミング

トのロスと遅延の設定を行う．テストは Windows のソケットを使用し，一定間隔でパケットを伝送した．テスト時の各種設定ない内容は表 1 のとおり．

2.2　1 パケットロス時の動作

図 2 は，1 パケットのロスが発生した時の受信側でのパケット番号とその受信時刻の関係を示したものである．図ではパケット番号 994 のパケットがロスしたため，パケット番号 996 のパケットを受信した直後，再送されたパケット番号 994 のパケットを受信している．このように TCP の fast retransmission の機能により，後続のパケットの伝送を継続しながら，ロスしたパケットはすばやく再送される．このためスループットの低下はほとんどない．

しかしながら，図 3 に示すように，パケットロスが発生後にそのロスパケットの再送が行われるまで，受信側のアプリケーションでは TCP の順序制御のためにデータを受け取ることができない．ロスパケットの再送がさらにロスすることになれば，データを受け取れない時間はさらに延長される．極端に言えば，1 つのパケットがロスし続ければ，後続のデータを一切受信できないことになる．

リアルタイム性を求めるアプリケーションでは，一定時間以上遅延したデータは不要となる．不要なデータのために，受信側に届いている後続のデータをアプリケーションに提供できず，さらに再送を待つ間に届いていた後続のデータまでもが一定時間以上遅延してしまうかもしれない．リアルタイム性を重視するのであれば，信頼性を多少犠牲にしても，データの滞りを避ける方法が必要である．

2.3　バーストパケットロス時の動作

図 4 は，バーストパケットロスが発生した時の受信側でのパケット番号とその受信時刻の関係を示したものである．図ではパケット番号 994 から 996 までの 3 連続パケットのロスが発生したため，ロス後の受信タイミングが大きく崩れているのがわかる．パケット番号 994 と 995 のパケットはすぐに再送されるが，パケット番号 996 のパケットが再送されるまでに，後続のパケットの伝送を 400msec 以上中断してしまっている．その後，パケット番号 996 のパケットの再送を行い，後続のパケットを一気に伝送する．

1 パケットロスの場合では，まるでロスが発生しなかったかのように後続のパケットを伝送しながら，ピンポイントでロスパケットの再送を行っていたが，バーストパケットロスの場合，ピンポイントでロスパケットを再送する方法ではスムーズに対処できていない．また，図 4 では，ロスパケット再送完了後に後続のパケットを高レートで伝送しているが，伝送帯域が十分でない場合にはパケットロスを誘発してしまい，再びデータ伝送の中断を引き起こす可能性がある．

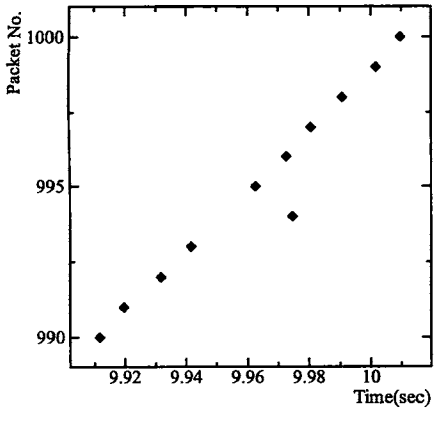

図 2　1 パケットロス時のパケット受信タイミング（従来方法）

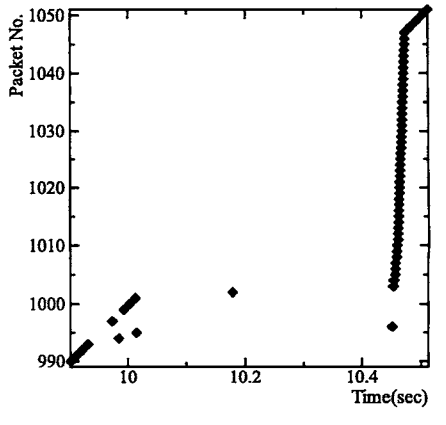

図 4　バーストパケットロス時の受信タイミング（従来方法）

このようにバーストパケットロス時にデータ伝送が滞るため，リアルタイム通信を行う際にはバーストパケットロスを引き起こさないようにするか，ロスパケット再送後の高レートの伝送を引き起こさないようにする対策が必要である．

3. 提案 TCP リアルタイム伝送方法

3.1 提案方法の検討

前章の検討により，TCP をリアルタイム通信で使用する場合には以下の項目に対応する必要がある．

・パケットロス時の後続データの滞り
・バーストパケットロス時のデータ伝送停止
・バーストパケットロス後の高レート伝送

各項目に関してそれぞれ検討を行い，提案方法としての対応方法を示す．

（1）パケットロス時の後続データの滞り対策

パケットロス時の後続データの滞りは，TCP の順序制御により避けることができない．そのため，データの滞りの影響を最小限に抑える方法を取る．つまり，パケットロスが発生した場合，後続のデータを増やさない．別の TCP コネクションを使用して伝送を継続することで，パケットロス時の後続データの滞りを最小限に抑え，伝送を継続して行うことが可能となる．

（2）バーストパケットロス時のデータ伝送停止対策

バーストパケットロス時のデータ伝送停止もソケットの動作として避けることができないのであれば，バーストパケットロスが発生しにくいような方法をとる．一般的なバーストエラー対策として使用されているインターリーブの考え方を適用し，複数の TCP コネクションを時分割に使用してパケット伝送を行う．

図 5 に，複数 TCP コネクションを使用したパケット伝送例を示す．まず，仮に 5 つの TCP コネクションを用意する．これら 5 つの TCP コネクションを順番に使用してパケットを連続的に送信する．この例では，最大 5 連続パケットロスが発生しても，各 TCP コネクションでは 1 パケットロスでしかないため，バーストロスの状況にはならない．そのため各 TCP コネクションでは fast retransmission の機能により，後続のパケットは継続して伝送されるとともに，ロスパケットの再送が並行して行われる．

（3）バーストパケットロス後の高レート伝送対策

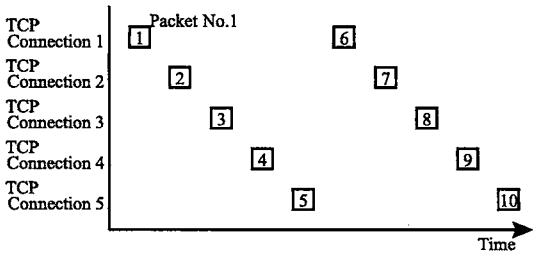

図 5 複数 TCP コネクション使用時のパケット送信タイミング

バーストパケットロス後の高レート伝送もソケットの動作として避けることができないのであれば，高レート伝送を極力抑える方法をとらざるを得ない．バーストパケットロス後の高レート伝送の時間を短くすることで対応する．方法としては 1 番目の項目と同じように，バーストパケットロス発生後はその TCP コネクションでのデータ送信を中止し，別の TCP コネクションを使用してデータの送信を行う．送信側では，ウィンドウサイズを小さくし，ウィンドウサイズ以上のデータを送信しないように制御することでパケットロス後の送信データ量を少なく抑えることができる．

（4）まとめ

以上の検討により，TCP を用いてリアルタイム通信を行う提案方法は以下の特徴を持つ．

・複数 TCP を順番に使用してパケットを伝送
・パケット伝送が滞ったらコネクションを使用中止

なお，複数 TCP コネクション間での順序制御のために，パケット番号に順ずる情報を各パケットに埋め込む必要がある．

3.2 提案リアルタイム伝送方法

複数 TCP コネクションを利用したリアルタイム伝送方法を提案する．提案するモデルを図 6 に示す．複数の TCP コネクション（図では 5 つ）を確立し，そのコネクションを順番に使用してパケット伝送を行う．

この提案モデルを利用したリアルタイム伝送方法として，完全リアルタイム方法と準リアルタイム方法の 2 つの方法を提案する．

（1）完全リアルタイム方法

完全リアルタイム方法は非再送パケットのみを用いて通信を実現する．概念的には，パケットを全て異なる TCP コネクションで伝送することにより，各パケットは他のパケットの伝送状況に依存せず伝送が行える．TCP の順序制御，フロー制御，ふくそう制御などは，複数のパケットが存在するために発生する制御であるため，1 パケット送信時には制約は受けない．そこで各 TCP コネクションで 1 パケットのみを送信することで，リアルタイムにパケット伝送が可能となる．

しかしながら，送信するパケット数分の TCP コネクションを確立することは現実的でないため，パケット送信後に送達確

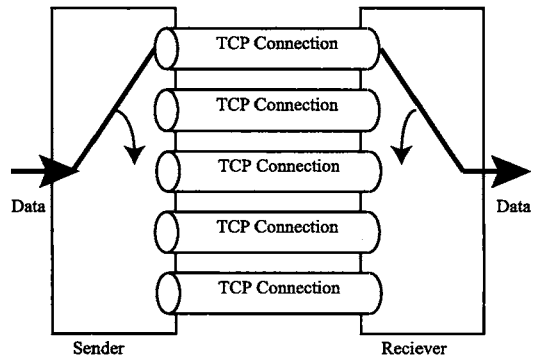

図 6 提案モデル

認を受け取ったTCPコネクションを再利用する．送達確認を受け取ることで，それ以前に送信したパケットの影響を受けることなく次のパケットを送信できる．なお，受信側ではパケットロス等により再送されたパケットは使用せず，非再送パケットのみを使用することで，信頼性を犠牲にしてリアルタイム性を保つ．

（2）準リアルタイム方法

準リアルタイム方法は，受信側で再送遅延が小さい再送パケットも利用することで，ある程度の信頼性を保ちつつリアルタイム性を実現するものである．

図6のように，複数のTCPコネクションを順番に使用してパケット伝送を行う．コネクション数と同数のバーストパケットロスまでは，fast retransmission機能により即座にロスパケットが再送されるため，再送による遅延が小さい．受信側でこの再送パケットを利用することで，完全リアルタイム方法に比べて，リアルタイム性は劣るが信頼性を向上することができる．

もし，各コネクション内でバーストパケットロスが発生したり，ロスパケットの再送が再びロスした場合，再送による遅延が増大するため，このような場合の再送パケットを受信側で使用しないようにすることでリアルタイム性を保つ．なお，受信側で処理対象のパケットの伝送遅延の最大値を制限することで，最大遅延時間を制限したリアルタイム性を実現できる．

4. 提案方法の動作検証

提案方法の動作検証を行う．データ伝送に関して，完全リアルタイム方法はTCPの特性にほとんど依存しないため，ここでは検証を行わず，準リアルタイム方法のみ検証を行う．

4.1 提案方法動作検証モデル

TCPの動作検証と同じように，図1の構成，および表1の設定でテストを行った．なお，送受信のPC間で5つのTCPコネクションを使用し，パケットの伝送を行った．

4.2 1パケットロス時の動作

図7は，1パケットのロスが発生した時の受信側でのパケット番号とその受信時刻の関係を示したものである．図ではTCPコネクションNo.5で伝送されたパケット番号965のパケットがロスしたため，パケット番号980のパケットを受信した直後に，再送されたパケット番号965のパケットを受信している．複数のTCPコネクションを使用してパケット伝送を行い，パケットロスが発生した場合でも，ロスパケット以外のパケット伝送動作には影響を与えず，安定したスループットを達成している．

TCPコネクションNo.5において，送信側はパケット番号965のパケットがロスした後で，続けてパケット番号970，975，980のパケットを送信し，受信側から重複応答を受信することでパケット番号965のパケットを再送している．1つのTCPコネクションを使用してパケットを送信する場合に比べて，複数のTCPを使用する場合には重複応答を受信する間隔が広がるため，ロスパケットの再送タイミングも遅れてしまっている．

しかしながら，図の例では約150msecの遅延でパケットが再送されているため，極端にリアルタイム性が損なわれることはない．再送時の遅延は，1つのコネクション内でのパケット送信間隔に依存するため，この値を調整することで所望の遅延時間を設定できる．

4.3 バーストパケットロス時の動作

図8は，バーストパケットロスが発生した際の受信側でのパケット番号とその受信時刻の関係を示したものである．図ではパケット番号981から983までの3連続パケットのロスが発生しているが，その約150msec後に3つのパケットは再送されている．TCPコネクションを1つだけ使用した時の図4で発生していたパケット伝送の中断もなく，ロスパケット以外のパケット伝送動作には影響を与えず，安定したスループットを達成している．

さらに，図4のTCPコネクションを1つだけ使用した場合に発生していたロスパケット再送後の高レートでの伝送は，図8では発生していない．10ms間隔でロスした3連続パケットは，それぞれ150msec後に10ms間隔を保って再送されているため，後続のパケット伝送と重なって再送時のみ最大2倍の伝送レートとなるが，極端な高レートとはならず，またその時

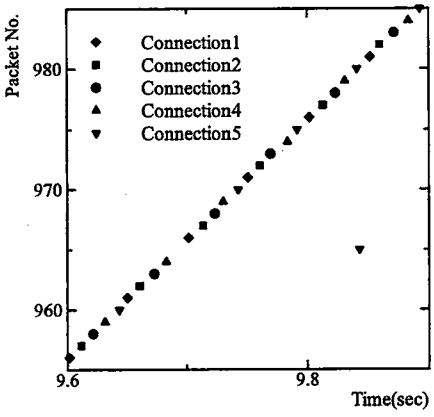

図7 1パケットロス時のパケット受信タイミング（提案方法）

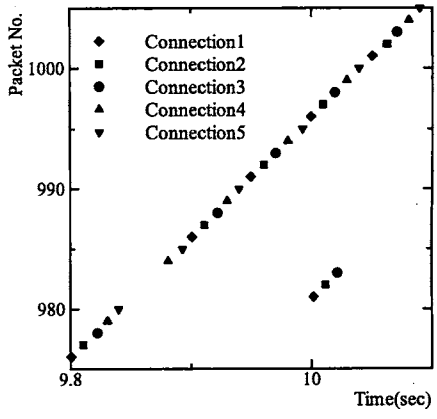

図8 バーストパケットロス時のパケット受信タイミング（提案方法）

間も短い．

バーストパケットロスの状況においても，1パケットロスと同様の伝送特性をなるため，ロスパケットの再送遅延は，1つのコネクション内でのパケット送信間隔に依存する．

このようにバーストパケットロス時にも，安定したデータ伝送を継続しながら，ロスパケットの再送が行われることが確認できた．

5. 考　　察

5.1 使用TCPコネクション数

パケット伝送時に使用するTCPコネクション数は，対応可能なバーストパケットロス数となる．TCPコネクション数を増やせば，バーストロス耐性は強くなるが，TCPコネクション数に比例してロスパケットの再送遅延は大きくなる．

そのため，実際にシステムを構築するときは，利用するネットワークの特性を考慮してバーストロス耐性を決定し，それに対応したTCPコネクション数を算出する．そして各TCPコネクション内でのパケット送信間隔からロスパケットの再送遅延を導き出し，受信側ではその再送遅延時間分だけ受信パケットを蓄積しながらメディアデータを再生することになる．もしくは受信側でのメディア再生時の遅延可能時間よりバーストロス耐性を求めることもできる．

5.2 情報発生量制御

本提案方法では，一定時間以上遅延したパケットを受信側で使用しない方針である．しかしながら，受信側で必要ないにもかかわらず，TCPの機能としてロスしたパケットを再送し続ける．さらに遅延時間を考慮せずにデータ伝送中に滞ったパケットを，受信側で確実に届けられるまで伝送動作を続ける．

そのため送信元で発生する情報量とTCPコネクションで再送している情報量を合わせたものがネットワークを流れる実際の伝送量となる．送信側では，合計の伝送量を計算しつつ，再送対象の情報量が多い場合は，ネットワークのふくそう回避のために送信元の情報発生量を制御する必要がある．発生した情報のリアルタイム伝送性を重視するのであれば，ふくそう時に発生した情報は全て廃棄し，ネットワークが安定してから発生した情報の伝送を再開すべきである．なお，再送対象の情報量は，各TCPコネクションのウィンドウサイズの使用量により概算できる．

5.3 データの蓄積

本提案方式はリアルタイム性を重視しているので，一定時間以上遅延したパケットは受信側で使用しない．しかしながら，TCPの信頼性のある伝送機能のため，受信側で不必要にも関わらず，全てのパケットは確実に受信側に届けられる．TCPのこの特性を利用して，受信側ではリアルタイムで受信処理を行うのと並行して，全ての受信パケットを蓄積することで，通信終了時には完全なデータを受信できている．

図9は，本提案方法を用いた場合における，受信側での非再送パケットと再送パケットの処理を示したものである．非再送パケットはそのままリアルタイムデータとして出力する．さらに非再送パケットと再送パケットを組み合わせて信頼性のあるデータとして蓄積する．

例えば，本提案方式を監視システムに適用し，リアルタイムで監視映像を伝送する場合を考える．一定時間以上遅延したパケットは受信側で再生に使用しないために，一時的に受信映像がフリーズするかもしれない．一方，リアルタイム性を要求されない蓄積動作は，一定時間以上遅延したパケットも蓄積用データとして利用することができる．つまり，蓄積用データはTCPの信頼性を利用してロスのないデータを蓄積することができ，完全な監視映像を後で見ることができる．

このように，本伝送方法を用いることで，リアルタイム性を重視した再生と信頼性を確保した蓄積を両立することができる．

5.4 ロス対策

UDPによるパケット伝送のロス対策として使用されているFECパケット[4]を本提案方法に利用することで，リアルタイム性を損なわず，より信頼性の高いパケット伝送が可能となる．

準リアルタイム方法に適用する場合，例えば1回までの再送パケットが許容されていれば，再送遅延時間以内にFECパケットを送信すればよく，FECパケットをFEC対象パケットとずらして送信することで，FECパケット自体のバーストロスの影響を受けづらくなる．これにより高いロス耐性が実現できる．

6. ま と め

複数のTCPコネクションを利用したリアルタイム伝送方法を提案した．複数のTCPコネクションを順番に使用してパケット伝送することで，バーストパケットロスが発生した場合でも低遅延かつ安定したスループットを確保できることを示した．

特に，1つのTCPを使用した場合には，バーストパケットロス時にパケット伝送が一時中断し，その後高レートでパケット伝送を行っていたが，本提案方式ではこのような異常な状況は発生せずに安定した伝送を行える．

また，使用するTCPコネクション数やパケット送信間隔により，バーストパケットロス耐性や受信処理最大遅延時間を自由に設定することが可能である．

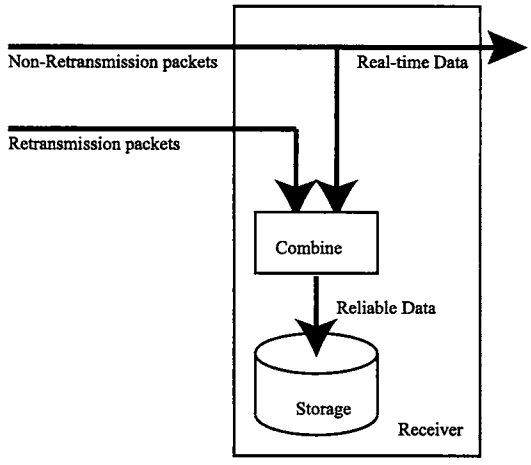

図9　提案方式の拡張例

非再送パケットのみを使用しリアルタイム性を最優先にした完全リアルタイム方法と，一部の再送パケットを使用しリアルタイム性を多少犠牲にしながら信頼性を向上させた準リアルタイム方法を示した．さらにリアルタイム性を重視した受信データの再生を行いつつ，全ての再送パケットを蓄積することで，完全な信頼性を実現する方法も合わせて示した．

以下に本提案方法の特徴を示す．

- リアルタイム性と信頼性のトレードオフが可能
- バーストパケットロス耐性を設定可能
- 最大受信遅延を指定可能
- 蓄積時には完全な受信データを再現可能

今後，バーストパケットロスが発生する無線環境等で本方法の試験を行い，パケットロス耐性を検証するとともに，さらなる方法の改良を行っていく予定である．また，本提案方法で対応しきれないほどのバーストパケットロスが発生した場合の対応方法に関しても検討を行う予定である．

文　献

[1] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[2] M. Allman, S. Floyd, C. Partridge, "Increasing TCP's Initial Window", RFC 3390, October 2002.

[3] 長谷川 洋平，村瀬 勉，"TCP の特性を考慮した複数経路通信方法の提案と評価，"信学技報，NS2003-328，p.p.175-178，March 2004.

[4] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

[5] S. Floyd, T. Henderson, A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 3782, April 2004.