

理工学研究科;鹿

GD

K

0006506190

R

462

静岡大学附属図書館

静岡大学 博士論文

誤り回復を行う高速パケット通信の
順序制御に関する研究



2005年12月

大学院 理工学研究科

鹿間 敏弘

概要

「いつでも、どこでも、誰とでも、なんでも」の実現を目指すユビキタスネットワークでは、無線通信が重要となる。無線通信では伝送誤りが発生しやすく、誤り回復制御が不可欠である。このような伝送誤りに対して誤り訂正符号とともに IP のようなパケット通信では再送プロトコルである ARQ (Automatic Repeat reQuest) が用いられる。

ARQ には種々の方式が存在するが、無線通信が高速化すると帯域と遅延時間の積が大きくなるため、喪失したパケットのみを選択的に再送する SR (Selective-Repeat) ARQ が重要となる。SR ARQ は再送パケットによるオーバーヘッドを必要最小限とするので高いスループットが得られる。しかし、パケットの順序を保証するための制御が必要で、受信側ではこのために大量のバッファを用いる。このような順序制御に関しては従来から研究が行われているが、主に遅延時間やバッファ保留量の点から研究されてきた。

本研究では順序制御により SR ARQ の出力がバースト的となる問題を指摘し、解析によりバーストに含まれるパケット数とバースト発生間隔に関する確率分布を求める。このようなバースト性は従来注目されていなかったが、ネットワークの QoS に大きな影響を及ぼす可能性がある。また、SR ARQ の順序制御を行わなければバースト的なパケット出力を回避できるが、パケットの順序逆転が発生する。本研究では順序制御を全く行わないかまたは順序制御のために保留するパケット数を制限した場合について、順序逆転パケット数の確率分布を解析により求める。

従来の研究では再送回数を制限しない場合について順序制御による遅延時間が求められているが、本研究では再送回数が制限されている場合について順序制御による遅延時間分布を解析により求め、再送そのものによる遅延時間より順序制御による遅延時間が支配的であることを示す。また、SR ARQ 上に複数のフローを多重化する場合、HOL (Head Of Line) ブロッキングが生じ、これにより不必要に遅延時間が発生することを示す。

このような出力トラヒックのバースト性及び HOL ブロッキングを回避する方式として PFRS (Per-Flow ReSequencing) 方式を提案する。PFRS 方式では従来は一体として扱われていた再送制御と順序制御を分離し、パケットの喪失検出と再送は従来方式と同様に SR ARQ 上に多重化されたフロー全体で行うが、順序制御はフローごとに独立に行う方式

である。上位層として UDP 及び TCP が用いられる場合についてシミュレーションによる評価を行い、PFRS 方式により順序制御による遅延時間が減少すること、遅延時間の減少により TCP のスループットが改善されること、また SR ARQ が出力するバーストの大きさが TCP のウィンドウサイズ 1 本分に抑えられることを示す。また、先に述べた順序制御による遅延時間の解析から求めたスループットとシミュレーション結果を比較し、解析による評価が実際のリンク層プロトコルを用いた場合にも有効であることを示す。

PFRS 方式の実現においては SR ARQ における上位層フローの識別及び各上位層フローに属するパケットの順序識別が課題となる。このような課題を解決する方式として SR ARQ の通番をポインタとして用いる方式を提案する。ポインタはあるパケットにおいて同じフローに属するパケットで直前のものをその通番を用いて明示するものである。このようなポインタを用いると、SR ARQ 上に多重化されるフロー数の制限なしに PFRS 方式を実現できる特長がある。また通番の差分を用いる差分ポインタを提案し、値の大きな差分ポインタの頻度が小さくかつ順序制御による遅延時間が小さくなる性質を明らかにする。このような性質を利用して差分ポインタの値に上限を設け、上限を超えた場合は、順序制御を行わない方式及び全てのパケットに対し順序制御を行う方式を提案する。シミュレーションによりこれらの方式を用いれば差分ポインタによるオーバーヘッドを削減できることを示す。

このような PFRS 方式とは別に、SR ARQ における順序制御の問題を解決する方式として SR ARQ で順序制御を行わない方式を提案する。この方式は SR ARQ で順序制御を行わず、その代わりに TCP の順序制御を利用する。この場合、SR ARQ で順序制御を行わないことによりパケットの順序逆転が発生し、これが TCP において重複 ACK を発生させ、不必要な再送を招く問題がある。この問題を解決するため、SR ARQ の送信側で SR ARQ が再送を行ったパケットに関する TCP の重複 ACK を除去する方式を提案する。提案する SR ARQ で順序制御を行わない方式は、SR ARQ の受信側で順序制御によるバッファが不要である利点及び SR ARQ がバーストを全く発生しない利点がある。シミュレーションによりこの方式を評価し、PFRS 方式と同等のスループットが得られること、SR ARQ の出力トラヒックを中継するノードのバッファ数が限られている場合は、PFRS 方式に比べ優れた性能が得られることを示す。この方式は TCP を意識した制御が必要であるが SR ARQ の受信側でバッファが不要である。一方、PFRS 方式は SR ARQ の受信バッファが必要であるが上位層のプロトコルを意識した処理が不要となる利点がある。システムの要求条件によって、これらの方式を使い分けることが考えられる。

この研究において明らかにした順序制御の問題は無線通信が高速化して帯域と遅延時間の積が大きくなるに伴い顕著となる。提案方式は今後高速化が進む無線通信システムに有効である。

目次

第 1 章	序言	1
1.1	研究の背景及び目的	1
1.2	従来の研究の概観	12
1.3	研究の特長	15
1.4	論文の構成	16
第 2 章	SR ARQ が出力するパケットトラヒックのバースト性の解析	19
2.1	まえがき	19
2.2	解析モデルと仮定	21
2.3	バースト長とバースト発生間隔の確率分布	23
2.4	数値計算結果と考察	35
2.5	順序制御を行わない場合の順序逆転パケット数の分布	45
2.6	むすび	52
第 3 章	SR ARQ の順序制御による遅延時間の解析	53
3.1	まえがき	53
3.2	順序制御による遅延時間とその解析モデル	54
3.3	再送及び順序制御による遅延時間の確率分布	60
3.4	解析に基づく数値計算結果	66
3.5	むすび	75
第 4 章	PFRS 方式の提案と性能評価	77
4.1	まえがき	77
4.2	レイヤ 2 における ARQ と上位層フローとの関係	77
4.3	PFRS (Per Flow Re-Sequencing) 方式	80
4.4	シミュレーションモデル	81
4.5	シミュレーション結果	84

4.6	解析結果と実際の SR ARQ プロトコルを用いたシミュレーションとの比較	96
4.7	むすび	105
第 5 章	ポインタによる PFRS の実現方式	109
5.1	まえがき	109
5.2	PFRS 方式の実現課題とポインタ方式の提案	109
5.3	送受信処理アルゴリズム	115
5.4	SR ARQ 通番の差分によるポインタ方式	119
5.5	むすび	131
第 6 章	TCP の順序制御を用いる SR ARQ 方式	133
6.1	まえがき	133
6.2	関連する研究	135
6.3	順序制御を行わない SR ARQ に基づくリンク層プロトコル	137
6.4	シミュレーションによる提案方式の評価	144
6.5	むすび	157
第 7 章	結言	161
7.1	本研究のまとめ	161
7.2	今後の課題	163
付録 A	チャネルの状態確率の導出	165
付録 B	残余受信失敗回数による状態確率の導出	167
謝辞		169

第1章

序言

1.1 研究の背景及び目的

1.1.1 ユビキタスネットワークと無線通信ネットワーク

光ファイバが広く普及しコアネットワーク及びアクセスネットワークの光化が進展している。既存の電話回線を利用する ADSL の普及と FTTH (Fiber To The Home) の急速な展開によりブロードバンドネットワークが一般家庭でも手軽に利用できる時代となった。日本の有線通信によるブロードバンドサービスは政府が掲げた e-Japan 戦略 [1] の目標に達したものと認識されており、政府の目標はブロードバンドサービスを含む IT の利活用に重点を置いた u-Japan 構想 [2] に移っている。

u-Japan 構想ではユビキタスネットワーク社会の実現が目標に設定されており、ブロードバンドに続く今後の方向を示す言葉としてユビキタスが広く使われるようになった。ユビキタスという言葉は一般ユーザがコンピュータを身近に意識することなく使うユビキタスコンピューティングの概念から発生している [3]。日本ではコンピュータとともにネットワークの接続性に重点を置いたユビキタスネットワークの概念が提唱され [4]、総務省のユビキタスネットワークフォーラムなどを中心に国をあげた取組みが進められている。ユビキタスネットワークの定義は「どこにいても、ネットワーク、端末、コンテンツを自在に意識せずにストレスなく安心して利用できるネットワーク環境」とされているが、ユーザの視点からは「いつでも、どこでも、誰とでも、なんでも」を実現するネットワークである。

このようなユビキタスネットワークの実現においては、ADSL や光ファイバによる有線通信とともに衛星通信を含む無線通信ネットワークが重要で、これらを柔軟に接続しシームレスに利用できる環境が今後の目標となる。無線通信では端末の移動やローミングといった有線通信にはない固有の課題が存在するが、有線通信同様のブロードバンド化の

実現も大きな課題である。世界的には第三代移动通信 [11] やそれに続く次世代の技術、更に高速の無線 LAN の研究開発が進められている [5]。

1.1.2 無線通信と ARQ

一般に地上無線通信や衛星通信では、電波伝搬を利用するため伝送誤りが発生しやすく、特に移動無線通信では何らかの対策が不可欠である。最近ではターボ符号や LDPC (Low Density Parity Check Codes)[6] など誤り訂正符号技術の研究が進み、シャノンの理論限界に近い性能が符号技術により達成できるようになった。

しかし、データ通信では伝送誤り以外に受信側でのバッファオーバーフローなどによりデータが失われる場合があり、信頼性を必要とされる通信ではたとえ伝送路の品質が良い場合でも送達確認を行う必要がある。ここで、送達確認がとれない場合は一般にデータの再送が行われる。

このような送達確認と再送は、OSI (Open Systems Interconnection) のレイヤモデルにおいてデータリンク層とトランスポート層で行われる [48]。しかし、無線通信では物理層や MAC 層でも行われることが多い。再送を行うデータの単位は、この機能を実現するプロトコル層によってフレーム、パケット、セグメントなどの名称で呼ばれるが、本論文ではデータを運ぶ単位としてパケットの用語を基本的に用いる。

各層のプロトコルで行われる再送機能は、一般に ARQ (Automatic Repeat reQuest) と呼ばれ、比較的古くから用いられてきた。ARQ は大別すると、図 1.1 に示すように、Stop-and-Wait、Go-back-N (GBN)、Selective Repeat (SR) が存在する [7][9]。従来は単純さの点から Stop-and-Wait や GBN 型のプロトコルが主に使われ、Selective Repeat は衛星通信のように遅延時間の大きなシステムで用いられてきた。

Stop-and-Wait プロトコル

Stop-and-Wait では 1 パケットを送信後、そのパケットに対する肯定応答 (ACK) を受信して送達確認した後に、次のパケットを送信する。パケットを誤って受信した場合は、否定応答 (NACK) を返送し、送信側は NACK を受信するとパケットを再送する。ここで、NACK の代わりに送信側がタイマにより ACK を監視する方式もある。この場合、受信側は明示的に否定応答を送信するのではなく、パケットを誤って受信した場合はこれを無視する。送信側は ACK が一定時間待っても来なかったら NACK とみなして再送を行う。

Stop-and-Wait では通番を用いず、タイミング関係により送信パケットとそれに対する ACK 及び NACK を識別できる。動作が単純であるため、物理層や MAC 層における再

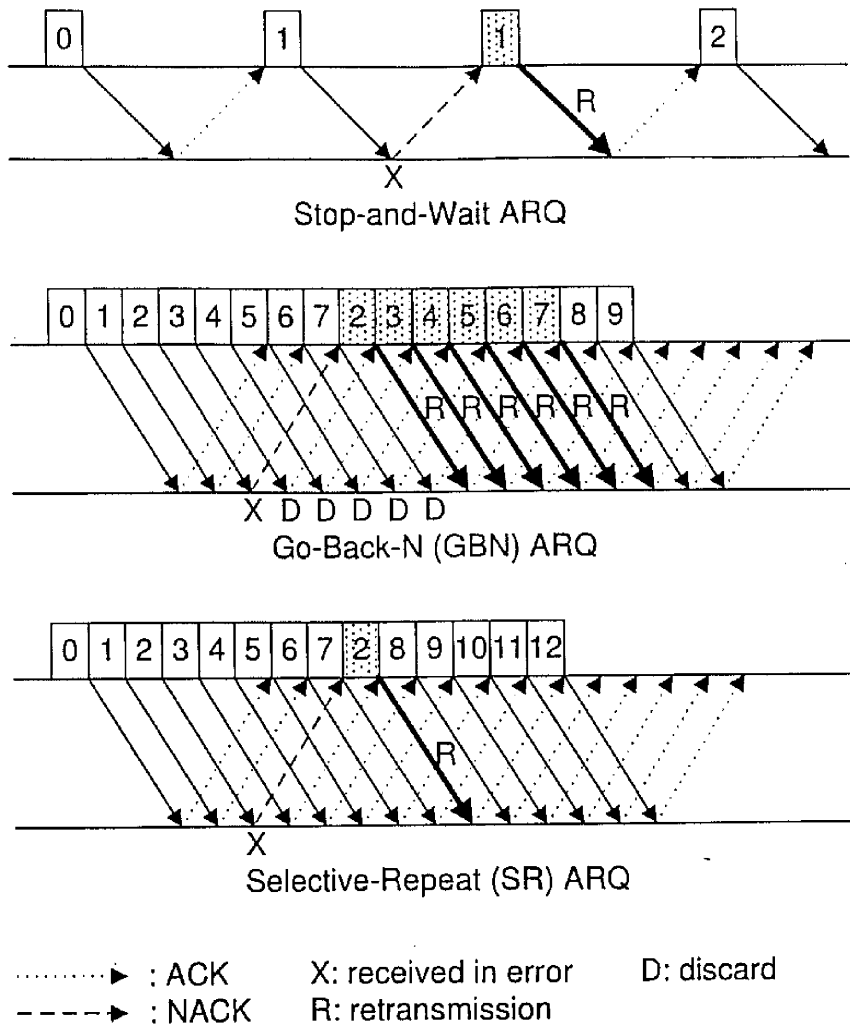


図 1.1 ARQ の種類

送は基本的に Stop-and-Wait が多い。無線 LAN などでは MAC 層で再送を行い、主に Stop-and-Wait を用いている。効率を上げるために、複数パケットを一つの LAN フレームに相乗りさせ、ACK は各パケットの送達確認を含む方式もある [12]。

GBN プロトコル

伝搬遅延時間や物理層での処理による遅延時間が長くなると、Stop-and-Wait プロトコルでは、パケットを送ってその送達確認を受け取るまで次のパケットを送ることができないので性能が劣化する。このため、送達確認を受けずに複数のパケットを送れるプロトコルが必要となる。このようなプロトコルではパケットに通番を振り、送達確認においてど

の packets が正しく受け取られたかを通番により識別することが必要となる。また送達確認なしに無制限に packets を送信できるようにすると受信側でバッファあふれを起こす可能性がある。このため、送達確認なしに送信できる packets 数（最大アウトスタンディング packets 数）を制限する場合が多い。このような packets 数の制御をウィンドウフロー制御と呼び、最大アウトスタンディング packets 数をウィンドウサイズと呼ぶこともある。ウィンドウサイズは ARQ のコネクションを設定する時点で固定的に決定する場合や通信の状況により動的に変更する場合もある。

このような packets の先送りを行うプロトコルで、伝送誤りにより packets が失われた場合、単純な再送方式としては GBN が用いられる。これは失われた packets 以降の全 packets を再送する方式である。再送された packets を受信するまで、受信側でも失われた packets 以降の正しく受信した packets をすべて廃棄する。このようにすると失われた packets 以降の情報をすべてクリアするので、送信側と受信側の状態を失われた packets を最初に送信する場合と等価な状況に戻すことができる。このため、状態情報を保存する必要がなくプロトコルがシンプルになる。ISO により標準化されその後の OSI が開発される契機となった HDLC (High Level Data Link Control) では基本的に REJ による再送が行われるが、これは GBN を基本としている。HDLC が標準化された時代のデータ通信は、アナログ回線でモデムを利用する場合が多く、ビット誤り率は 10^{-6} 程度、伝送速度も 9.6 kbit/s 程度が一般的であった。このような条件では GBN で十分な性能が得られた。

TCP はトランスポート層のプロトコルであるが、エンドツーエンドで再送を行っており、基本的には GBN を用いている。TCP では受信側が受信可能なウィンドウサイズを通知し、送信側ではこれと輻輳制御を考慮してウィンドウサイズを動的に変更している。ただし、TCP では受信側では正しく受信した packets を廃棄せずバッファに保留しており、純粋な GBN とは異なっている。このような TCP の受信側での動作は次に述べる SR ARQ に近い。

SR ARQ プロトコル

上で説明したように、GBN は誤りが発生すると受信側ではそれ以降に受信した packets をたとえ正しく受信しても廃棄し、送信側では誤りの発生した以降の packets をすべて再送する。図 1.1 で GBN の場合、通番 2 の packets が失われると、通番が 3 から 7 までの 5 packets は正しく受信されているにもかかわらず受信側で廃棄され、送信側は通番 2 から 7 までの packets を再送する。1 packets を送信するのに要する時間に対してラウンドトリップ時間が大きくなると先送りする packets の量が増加する。ラウンドトリップ時間は伝搬遅延時間や物理層での処理時間に依存し、1 packets を送信するのに必要な時間

は伝送速度が大きくなると小さくなる。これからラウンドトリップ時間と伝送速度の積が大きなシステムでGBNを用いると性能が低下することになる。従来、このような条件となる典型的な場合は、衛星通信であった。しかし、最近の第三代移動通信ではパケットの通信速度が増大しており、物理層の処理による遅延時間が比較的大きいため、ラウンドトリップ時間と伝送速度の積も大きくなっている。

このような条件では、失われたパケット以降に正しく受信したパケットを受信側でバッファに保留し、送信側では失われたパケットのみを選択的に送信するSR ARQプロトコルが必要となる。SR ARQでは失われたパケットのみを再送するため、GBNのように無駄にパケットを再送することがなく高い性能が得られる。ただし、SR ARQでは誤りの発生したパケットの後続パケットを正しく受信した場合は、これをバッファに保留し、再送されるパケットを待つ順序を揃えて出力する必要がある。図1.1でSR ARQの場合、通番2のパケットが失われ選択再送されているが、正しく受信された通番3から7までの5パケットは、順序を守るために受信側で保留される。本論文ではこのような制御を順序制御と呼ぶ。先に述べたようにGBNでは正しく受信した後続パケットを廃棄するので、このような順序制御は行う必要がなく、順序制御のためのバッファも不要である。

以上のようにSR ARQでは再送するパケットを最小限とするため、高い性能（スループット）が得られる。従来、SR ARQについては、高い性能を得るために受信側で大量のバッファが必要であること、また実現においてプロトコル処理が複雑であることが問題点として認識されていた。SR ARQがこれまで衛星通信など限定的にしか用いられて来なかった理由の一つは受信側の大容量バッファの問題と考えられる。しかし、最近はメモリが安価となり大容量のバッファを用いることが経済的にも許容できるようになった。また低コストなマイクロプロセッサの性能向上により処理負荷についても実用上問題ないレベルとなってきた。このため、今後の無線通信の高速化に伴ってSR ARQがより広く用いられるものと予想される。

1.1.3 ARQの最大スループットとSR ARQの適用領域

先に述べたようにSR ARQは主に衛星通信で用いられてきたが、今後、無線通信の高速化に伴い、伝搬遅延時間の短い地上無線通信であっても帯域遅延時間積が大きくなりSR ARQを用いる必要がある。図1.2と図1.3は伝搬遅延時間が同じ場合でも通信速度の高速化に伴い、SR ARQが必要であることを示している。図1.2は低速な通信の場合、図1.3は高速な通信の場合のGBNとSR ARQのシーケンス例を示している。これらの図で伝搬遅延時間を同じとしているが、通信が高速化するとパケットを送信して送達確認が返ってくるまでの間に送信されるパケット数が大きくなり、GBNでは正しく受信され

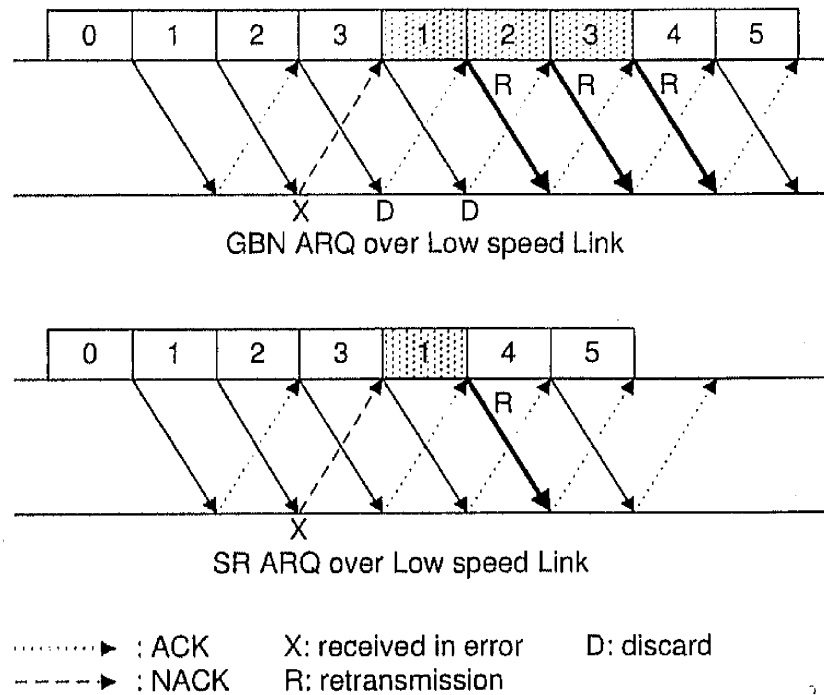


図 1.2 通信速度が小さい場合の GBN と SR ARQ のシーケンス例

たパケットで廃棄されるものの数が大きくなる。例えば、図 1.2 では GBN の場合、正常に受信された 2 パケットが廃棄されているが、図 1.3 では GBN の場合、5 パケットが廃棄されている。このように SR ARQ は伝搬遅延時間と帯域との積が大きい場合にその有効性が顕著となる。

上で述べた内容を ARQ のスループットの点から定量的に説明する。時間を 1 パケット送信に十分な大きさにスロット化し、パケットを送信してからその送達確認が返ってくるまでのラウンドトリップ時間を M スロットとする。パケット誤り率を ε とし、最大再送回数の制限がないものとする、各 ARQ の最大スループットは以下のように求められる [8]。ここで、1 パケットの送信に X スロット (X は確率変数) を要するものとする。

Stop-and-Wait ARQ の場合、 X に関する確率は次のようになる。

$$\text{Prob}[X = (k + 1)M] = (1 - \varepsilon)\varepsilon^k \quad (1.1)$$

ここで、 $k = 0, 2, \dots$ 。これから、1 パケットを転送するのに平均で $\frac{M}{1-\varepsilon}$ スロットを要することとなり、最大スループット S_{sw} は次のようになる。

$$S_{sw} = \frac{1 - \varepsilon}{M} \quad (1.2)$$

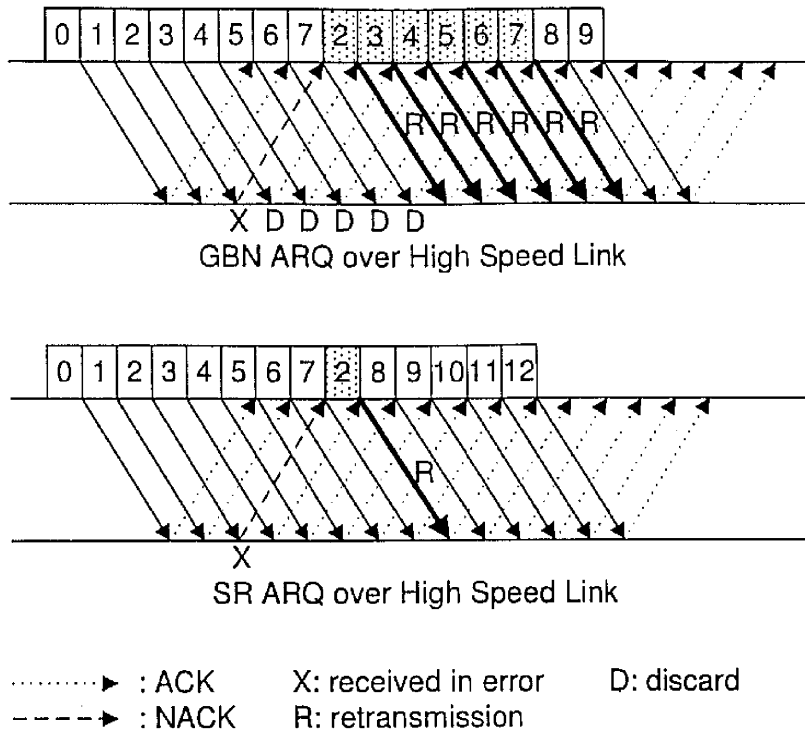


図 1.3 通信速度が大きい場合の GBN と SR ARQ のシーケンス例

GBN ARQ の場合、X に関する確率は次のようになる。

$$Prob[X = 1 + kM] = (1 - \epsilon)\epsilon^k \tag{1.3}$$

これから、1 パケットを転送するのに平均で $1 + \frac{M\epsilon}{1-\epsilon}$ スロットを要することが分かるので、その最大スループット S_{GBN} は次のようになる。

$$S_{GBN} = \frac{1 - \epsilon}{1 + (M - 1)\epsilon} \tag{1.4}$$

SR ARQ の場合、X に関する確率は次のようになる。

$$Prob[X = k + 1] = (1 - \epsilon)\epsilon^k \tag{1.5}$$

これから、1 パケットを転送するのに平均で $\frac{1}{1-\epsilon}$ スロットを要することが分かる。最大スループット S_{SR} は次のようになる。

$$S_{SR} = 1 - \epsilon \tag{1.6}$$

図 1.4 に M を 10 及び 100 とした場合のこれら最大スループットの計算例を示す。上記の式及び図 1.4 から、Stop-and-Wait ARQ はラウンドトリップ時間が大きくなると

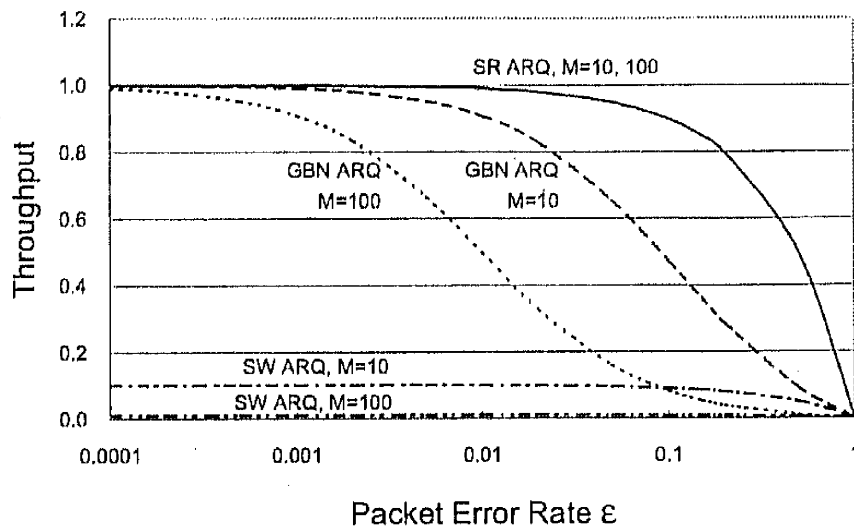


図 1.4 ARQ のスループット

急速に性能が低下すること、SR ARQ はラウンドトリップ時間の影響を受けないことが分かる。GBN ARQ はラウンドトリップ時間に 1 パケット程度の誤りが発生する条件で ($M\epsilon = 1$)、スループットが約半分に低下する。この条件は、図 1.4 で $M = 10$ の場合に $\epsilon = 0.1$ 、 $M = 100$ の場合に $\epsilon = 0.01$ のパケット誤り率に相当する。

パケットの伝送誤りがほとんど発生しない条件では GBN ARQ と SR ARQ はラウンドトリップ時間が比較的大きい場合でも良好な性能が得られる。しかし、ラウンドトリップ時間の間に 1 パケット以上の誤りが発生する状況では SR ARQ を用いた方がよいと言えることができる。

この条件をビット誤り率と伝送速度の関連から検討する。伝送路においてビット誤りが誤り率 e_b でランダムに発生するものとし、パケット長を L bit とする。伝送速度が T_x bit/s でラウンドトリップ時間を D_{RTT} とすると、パケット誤り率 ϵ は以下となる。

$$\epsilon = 1 - (1 - e_b)^L \approx L \cdot e_b \quad (1.7)$$

ラウンドトリップ時間に発生する平均誤りパケット数は次のようになる。

$$D_{RTT} \cdot (T_x/L) \cdot \epsilon \approx D_{RTT} \cdot T_x \cdot e_b \quad (1.8)$$

従って、SR ARQ の適用領域は $D_{RTT} \cdot T_x \cdot e_b > 1$ の条件となる。これはパケット長 L に依存せず、ラウンドトリップ時間、伝送速度、ビット誤り率の三パラメータの積が大きい場合に SR ARQ が適していることを示している。図 1.5 はラウンドトリップ時間を 0.5 sec, 0.05 sec, 0.005 sec, 0.0005 sec とした場合の、 $D_{RTT} \cdot T_x \cdot e_b = 1$ となるビット

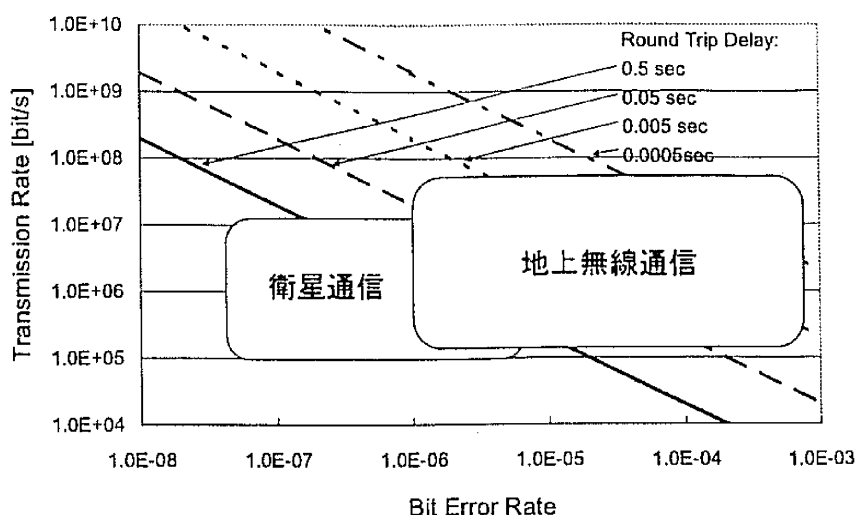


図 1.5 ARQ の適用領域

誤り率と伝送速度の関係を示している。これらの直線より上の領域が SR ARQ の適用領域となる。

衛星通信の場合、伝搬遅延時間が 0.25 sec となるため、比較的低い 1M bit/s 程度の速度で SR ARQ が必要となる。携帯電話などの地上無線通信では、ラウンドトリップ時間が短くなるものの（例えば 0.05 sec）、通信速度及びビット誤り率が大きいため、SR ARQ が必要となる。例えば、地上の無線通信で速度が 10 Mbit/s の場合、ラウンドトリップ時間が 0.01 sec でビット誤り率が 10^{-5} であれば、SR ARQ が必要といえる。一般に移动通信などでは物理層や MAC 層による遅延時間の大きいことが報告されている [94][95][96]。このため、第 2.5 世代の移动通信と呼ばれる GPRS では RLP (Radio Link Protocol) と呼ぶ HDLC ベースの SR ARQ プロトコルが用いられている [74]。また、第三世代の 3GPP では RLC (Radio Link Control) と呼ばれる SR ARQ によるリンク層プロトコルが用いられる [52]。このように、地上の無線通信でも高速化が進むと SR ARQ が必要となる [41]。また、固定の衛星通信では晴天の場合、良好な通信品質が得られるが、移動体衛星通信では障害物により電波が遮蔽され、伝送品質が大幅に低下する。Loreti らの研究では静止衛星 GEOS の場合の packets 誤り率として 10^{-3} から 10^{-1} 、低軌道周回衛星 LEOS の場合の packets 誤り率として 10% から 40% の条件を想定して、シミュレーションにより TCP の性能を評価している [13]。

1.1.4 リンクごとの再送とエンドツーエンドの再送

先に述べたように再送はデータリンク層以下とトランスポート層で行われる。データリンク層以下で行われる再送はリンク単位に行われる。ネットワーク層は複数ノードを経由する経路でパケットを転送するので、上位のトランスポート層による再送は、経路のエンドツーエンドにおける再送となる。初期のパケット交換網が整備された1980年代は光ファイバの導入が十分ではなくアナログの回線が主に用いられた。このような場合には、リンク単位に再送を行うことにより経路を構成するリンクの品質を向上させるリンクバイリンクの再送が主流であった。ITU-T 勧告 X.25 による公衆のパケット交換網では、網と端末のインタフェース及び網内部でリンクごとの再送がデータリンク層（レイヤ2）で行われている。

1990年代に普及したインターネットでは、特定のリンク層プロトコルを規定せず、パケットの再送はTCPによりエンドツーエンドに行っている。光ファイバによる品質の良い有線回線を利用する場合は、再送によりリンクの品質を向上させる必要はなく、エンドツーエンドの再送で十分であった。逆にデータリンク層におけるHDLCなどのプロトコルは、ノードの処理負荷を増大させネットワークの性能を低下させるとして排除されてきた。このような考え方は、1980年代後半に登場したフレームリレーや1990年代に標準化が進められたATMにおいても基本となっている。これらの方式ではノード装置を簡単にして処理能力を向上させるため、リンクごとの再送を省略している。

しかし、無線通信を利用するネットワークでは伝送誤りの発生が有線の伝送路より極めて多く、TCPによるエンドツーエンドの再送では十分な性能が得られない問題がある。端末が移動する携帯電話等ではビット誤り率が 10^{-3} 程度で設計されており[96]、衛星通信でも移動体に対する通信サービスを提供する場合には、シャドウイングなどにより電波が一時的に遮蔽され通信断が発生する。このように、安定した通信品質が確保できない場合、リンクごとに再送を行って、見かけ上のリンク品質を向上させるアーキテクチャが一般に採用されている。本研究では、このように比較的伝送品質の悪いリンクを対象とし、伝送品質の目安としては、先に説明したようにラウンドトリップ時間に1パケット程度以上の伝送誤りが発生する状況を対象と考えている。

1.1.5 本論文で指摘するSR ARQの問題点

以上述べたように、良好な伝送品質が常に得られるとは限らない無線通信区間ではARQが必要で、ここでSR ARQを用いて再送パケットを最小限とすることにより高い性能が得られると期待されるが、その代わりパケットの順序制御が必要となる。SR ARQ

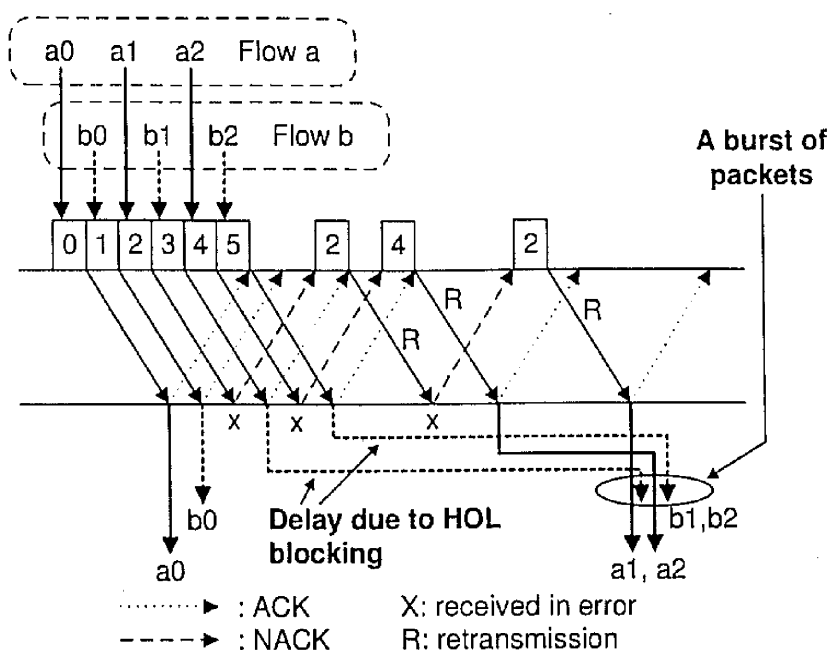


図 1.6 本論文で指摘する SR ARQ の問題点と研究の目的

の順序制御による影響については従来あまり注意が払われておらず研究もなされていない。再送は基本的に通信の QoS (Quality Of Service) を大きく損なう機能であり、特に SR ARQ ではこの傾向が著しい。ユビキタスネットワーク社会実現に向けて無線通信の高速化が進むのに伴い、従来は注目されていなかった問題が今後のネットワークにインパクトを与える可能性がある。本論文では、SR ARQ の順序制御により図 1.6 に示す次の問題が発生することを指摘する。

- SR ARQ の出力トラヒックがバースト的となる。
- SR ARQ 上に複数の上位層フローを多重化する場合、HOL (Head Of Line) ブロッキングにより、パケットに不必要な遅延時間が発生する。

これらの問題については 2 章と 3 章で詳しく説明し解析する。

本研究では、ユビキタスネットワークに向けた今後の無線通信のブロードバンド化の背景から従来あまり注目されていなかった SR ARQ の受信側で順序制御に着目し、そのネットワークへの影響を検討する。更に影響を緩和する方式を提案し評価を行う。

具体的なリンク層プロトコルとして、本研究では特定のプロトコルを仮定していない。想定される具体的な SR ARQ プロトコルの例としては、W-CDMA 移動通信 [11] における RLC[52] や GSM における RLP[74]、一般のデータ通信分野では SREJ を用いる HDLC[72][73] などが挙げられる。本研究では後で述べるシミュレーションにおい

て、プロトコルの動作が比較的シンプルで通番の空間が大きい SSCOP (Service Specific Connection Oriented Protocol) を用いている [65]。

このような広帯域無線通信環境で使用されるサービスはインターネットアクセスが最も多く利用される。WEB などの利用ではトランスポートプロトコルとして TCP が用いられる。インターネットトラフィックの 95 % 以上は TCP によるといわれており [35]、TCP を想定した評価が重要である。TCP は高信頼な通信を実現するため伝送誤りに対する再送機能を有するが [10]、再送能力が十分ではなく、無線通信を利用する場合、下位のデータリンク層で ARQ が実行され上位で TCP が用いられる場合が一般的と考えられる。しかし、下位レイヤによる再送が上位の TCP に悪影響を与える可能性がある。第 2.5 世代及び第三世代の移動通信における TCP の性能については報告されているが [93][98]、本研究では上位層で TCP が動作する状況での SR ARQ の影響をより詳細に検討する。

1.2 従来の研究の概観

これまで、ARQ については Stop-and-Wait と GBN における送信待ち行列の解析が最初に研究されている [15]。SR ARQ については、スループットを中心に評価が行われ [21]、次に送信側での待ち行列を中心に伝送遅延時間について研究されている [17][23][43][44][45]。また TDMA チャンネルにおける Stop-and-wait、GBN、SR ARQ について研究されている [20]。

SR ARQ には再送によるパケット喪失の回復を同時には 1 パケットしか行わない逐次再送型と同時に複数パケットの回復を行う多重再送型とに分類できる。従来の SR ARQ に関する研究はほとんどが後者を対象としているが、TCP における高速再送の動作などは前者に位置づけられる。筆者により、前者の場合の順序制御による待ち時間の解析が行われている [49][50]。また逐次再送型と多重再送型を比較した解析もなされている [51]。

SR ARQ をベースとしたプロトコルのバリエーションも多数研究が行われている。再送回数に制限を設けない場合、SR ARQ の受信側では順序制御のために無制限に受信バッファが必要で、受信バッファがオーバーフローする可能性がある。これらの研究はこれを防ぐことを狙っている。Bruneel の研究は一つのフレームに対して複数個のコピーを伝送することによりスループットの向上を図っている [25]。Miller らは SR ARQ で再送が成功しない場合、Stop-and-Wait または GBN に切り替えることにより受信バッファのオーバーフローを避ける方式を提案している [18]。また Weldon らは再送回数が増えるたびに複数個再送するコピーの数を増加させる方式を研究している [19]。また、Lin らは GBN をベースに正しく受信したパケットの保留と繰り返し再送を行う SETRAN 方式を提案している [16]。

この他のバリエーションとしては、藤井らはメッセージの順序を保存しない場合についてメッセージの遅延時間分布を求めている [46]。林田等はメッセージは GBN による再送、1 メッセージを構成する複数パケットについては SR ARQ による選択再送を行う方式を研究している [42][47]。SR ARQ をポイントツーマルチポイントに拡張した場合の性能について研究されている [24][26]。

SR ARQ の順序制御については、これによる遅延時間が回線がフルに稼動する条件のもとに解析による評価が行われている [28]。ここでは再送回数の制限を設けない条件となっている。Chang らは SR ARQ の順序制御を含むエンドツーエンドの遅延時間を近似的な解析により求めている [34]。Kim らは伝送路のパケット誤り率が 2 状態のマルコフモデルとした場合の順序制御及び送信側の待ち行列による遅延時間を解析している [37]。SR ARQ をマルチキャスト通信に用いる場合の順序制御による遅延時間も解析されている [27]。ノード間に複数のリンクが存在し、各リンクで SR ARQ を行い、パケットを複数のリンクに分散させる場合について受信側での順序制御による遅延時間とバッファ数について解析が行われている [31]。Rosberg らは SR ARQ の送信側の送達確認待ちによる保留バッファ数と順序制御による保留バッファ数の結合分布を解析している [30]。第三代移動通信の W-CDMA では無線区間のデータリンク層プロトコルである RLC において、SR ARQ を用いるが、その順序制御についてはいくつかのオプションが用意されており、評価が行われている [54][53]。また RLC の上位層で TCP を用いた場合の特性評価も報告されている [71]。

SR ARQ 関連以外に順序制御に関する一般化された研究が行われている。これらは、図 1.7 で示す一般的なモデルについて解析を行っている [38]。ここで、Mis-ordering Network におけるランダムな遅延時間の分布は、到着間隔分布や FIFO のサービス時間分布とは独立とした場合が解析されている。Baccelli らはランダムな遅延によりパケットの順序が乱れた場合の順序制御に関してエンドツーエンドの遅延時間を解析している [22]。図 1.7 で Mis-ordering Network におけるランダムな遅延時間分布として GI/GI/∞ による遅延時間分布を採用している。任意の独立な時間分布で解析しているため、分布のラプラス変換を用いて GI/GI/1 の場合と類似の Wiener-Hopf 型の積分方程式を導出している。スペクトル分解法により解を求める方法が示されているが、解はラプラス変換の形で求まるため、実際の分布関数が求められるのは指数分布のような単純なものに限られる。この研究では図 1.7 で FIFO Queue を含んだエンドツーエンドの遅延時間を求めているが、他の研究ではこの FIFO 部分を考慮していない。Xia らはパケットごとに独立なランダムな遅延が存在する場合の順序制御による遅延時間と必要なバッファサイズを解析している [38]。ここでランダムな遅延時間としては D/GI/∞ による遅延時間としているが、実際に解析しているのは指数分布とパレト分布の場合である。Jean-Marie らは複数の並

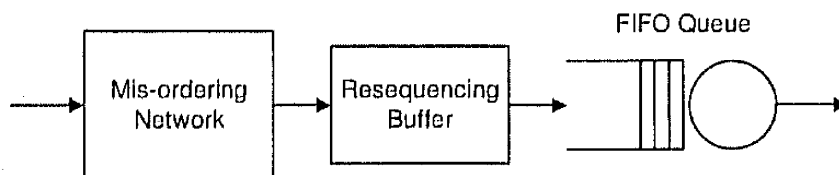


図 1.7 順序制御の一般的なモデル

列待ち行列にランダムに入力パケットを分配して、受信側で順序を揃える場合について解析を行っている [33]。この研究では図 1.7 に示すモデルで、ランダムな遅延時間分布として k 個の $M/GI/1$ による遅延時間を用いている。また同じ研究者により、ある確率でパケットが一定時間遅れる場合の順序制御による遅延時間も解析されている [36]。この解析では固定遅延時間を扱っており、ランダムに遅れる場合の順序制御には適用できない。このほか、図 1.7 の Mis-ordering Network におけるランダムな遅延時間分布としては、 $M/M/\infty$ による遅延時間分布 [110]、 $M/GI/\infty$ による遅延時間分布、 $M/H_2/K$ による遅延時間分布などがある [111]。

パケット交換ネットワークで同一フローのパケットを複数の経路に分散させる場合の受信側での順序制御についても解析がなされている。Plotkin らは ATM でセルを複数の VC に分散させる場合の順序制御に関する解析を行っている [32]。吉田はノード間で複数のリンクを用い、パケットを複数のリンクに分散させた場合の受信側での順序制御による遅延時間を解析している [29]。また Nebat らは同一ファイルの異なるセグメントを複数のサーバから並行してダウンロードする場合の順序制御による遅延時間を研究している [39][40]。

上記のように、SR ARQ により帯域遅延時間積の大きい環境で、高いスループットが得られることは知られていた。また、順序制御により受信側に多量のバッファが必要なこと、再送回数に制限を設けない場合は無限に受信バッファが必要となることは認識されていた。しかし、上位層のフローの多重化と順序制御との関係については研究されていない。また、順序制御による出力トラヒックのバーストについては注目されておらず、筆者等によるもの以外では研究されていない。また、順序制御による問題を解決する方式提案も筆者等によるもの以外では発表されていない。一つの理由は、SR ARQ の順序制御による問題は衛星通信において顕著であったものの、地上無線通信では遅延時間が比較的小さく、またこれまでの無線通信の速度が小さかったため、帯域と遅延時間の積が問題となるほど大きくなかったためと考えられる。しかし、無線 LAN や移動通信の通信速度は急速に大きくなっており 100 Mbit/s を超える速度が達成されつつある。帯域と遅延時間の積が大きくなるのに伴い、順序制御による問題が顕在化するものと予想される。

1.3 研究の特長

本研究では、SR ARQ の受信側で必要となる順序制御について考察し、順序制御により出力トラヒックに大きなバースト性が生じることを示す。また解析により、出力バーストに含まれるパケット数、バースト発生間隔、順序制御を省いた場合の順序逆転パケット数などについて確率分布を求める。SR ARQ の出力におけるバースト性及び順序制御を省いた場合のパケットの順序逆転特性は従来把握されていなかった問題で本研究で最初に明らかにされたものである。

また、本研究では順序制御により発生するパケットの遅延時間分布を解析する。解析において最大再送回数に上限を設けており、またフローごとに独立に順序制御を行う場合を考慮している。バースト性の解析とフローごとに順序制御を考慮した解析及びこれらの解析において最大再送回数に上限を設けた点は従来にはなく、本研究の解析における新しい展開である。また、本研究では SR ARQ 上に複数のフローを多重化する場合、HOL (Head Of Line) ブロッキングが発生して不必要に順序制御による遅延時間が発生する問題を明らかにする。

順序制御によるバースト性と HOL ブロッキングに関する問題を解決する方式として順序制御のみをフローごとに行う新しい PFRS (Per-Flow ReSequencing) 方式を提案しシミュレーション及び解析によりこの方式が有効であることを示す。PFRS 方式は本研究で提案された方式で、従来は一体として考えられていた SR ARQ における再送制御と順序制御を分離して考える新しい方式である。

HOL ブロッキングの問題は、データリンク層がレイヤモデルの考え方に忠実に従って上位層を一切意識しない結果生じる問題ともいえる。PFRS 方式はこのようなレイヤモデルの枠を緩和して性能を向上させる方式と考えることもできる。最近各レイヤごとの最適化に限界のあることが認識され、レイヤを跨って最適化を目指すクロスレイヤが研究の一つの流れとなっている [14]。PFRS 方式はこのクロスレイヤの一つの方向を示しているともいえる。

本研究では PFRS 方式の具体的な実現方式として SR ARQ の通番を利用して上位層フローの識別とフローごとのパケット順序識別を行うポインタ方式を提案し、シミュレーションによりその有効性を示す。この方式は SR ARQ の通番を同一フローのパケットを差すポインタとして利用するものである。またポインタを拡張した差分型ポインタを提案し、差分型ポインタによりポインタのビット数の削減が可能であることを示す。プロトコルにおいてこのようなポインタは従来にはない新しい概念で、本研究において最初に導入されたものである。

順序制御による遅延時間とバースト性能問題を解決する方式として PFRS 方式とは別に、SR ARQ で順序制御を行わず、エンドの TCP における順序制御を利用する方式を提案し有効性を評価している。SR ARQ で順序制御を行わないことにより TCP の重複 ACK が発生するが、SR ARQ で重複 ACK を除去することにより無効な再送の発生を防ぐ方式を提案している。SR ARQ で順序制御を行わない方式は他にも研究されているが重複 ACK 除去を組み合わせた方式は本研究で初めて提案されたものである。

以上のように本研究では新しい解析手法と解析結果、SR ARQ における機能の新しい分離方式やポインタの概念など、順序制御に関する新しい課題を提起し解決手法を提案している点に特長がある。

1.4 論文の構成

図 1.8 に本論文の構成を示す。2 章において SR ARQ の順序制御によるバースト性を解析し問題点について述べ、3 章で SR ARQ の順序制御による遅延時間を解析し、HOL ブロッキングによる遅延時間について説明する。これら 2 章及び 3 章は SR ARQ の問題点について分析している。4 章から 6 章は、これらの問題を解決する方式を提案している。4 章では解決方式として PFRS 方式を提案し、その性能評価について述べ、5 章で PFRS の実現方式としてポインタ方式を提案し性能を評価する。6 章では SR ARQ において順序制御を行わず重複 ACK 除去を行う方式を提案し性能評価について述べる。最後に 7 章でまとめと今後の課題を述べる。

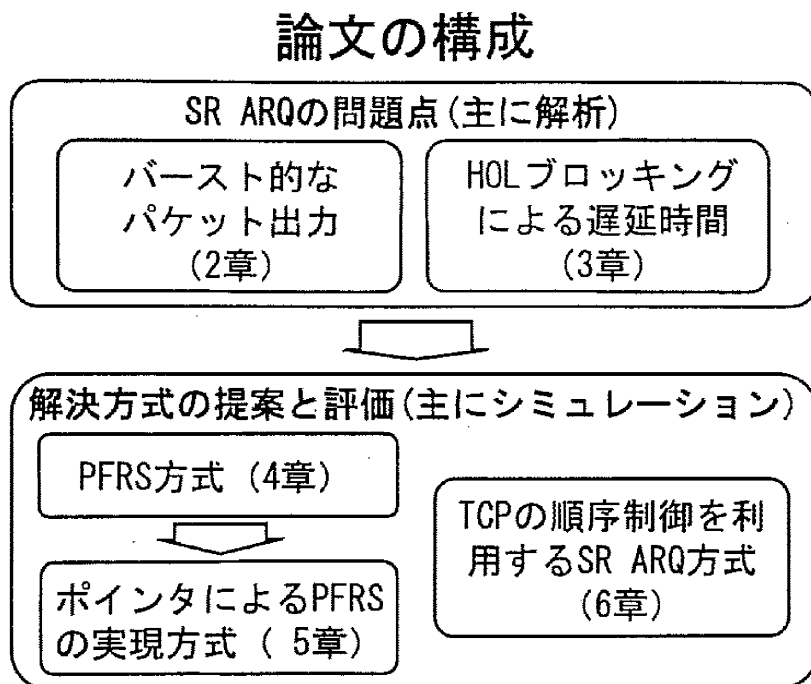


図 1.8 論文の構成

第2章

SR ARQ が出力するパケットトラヒックのバースト性の解析

2.1 まえがき

1章で述べたように、誤り回復プロトコルとしては単純さの点から GBN ARQ が従来用いられてきたが、通信速度が大きくなるにつれ、伝搬遅延や物理層及び MAC 層での処理による遅延時間が相対的に無視できなくなり、SR ARQ が用いられる傾向にある。この SR ARQ プロトコルは、伝送誤りにより喪失したパケットのみを再送し、このパケット以降に正常に受信したパケットは、順序を保証するため喪失したパケットが再送されるまで受信側で保留する。

図 2.1 に SR ARQ プロトコルのシーケンス例を示す。この図から分かるように再送されたパケットを受信した際、順序制御のために保留していたパケットを一斉に解放し、上位層に転送する。この同時に転送されるパケットの集合を本論文ではバーストと呼ぶ。このようなバーストは SR ARQ の出力パケットを中継するネットワークにおいてバッファオーバフローや遅延時間変動など QoS の劣化を引き起こす可能性があり、その分布を把握することは重要である。特に、高速の無線通信のように伝搬遅延時間と通信速度の積が大きな環境では、SR ARQ において送達確認を受けずに送信できるパケット数を大きくしなければならないが、これによりバースト的に出力されるパケット数も極めて大きくなる。

SR ARQ の順序制御によるバースト的なパケットトラヒックはこれまでの研究で全く注目されていなかった。SR ARQ は入力トラヒックの性質を著しく変化させるが、バーストに含まれるパケット数やバーストの発生間隔など出力トラヒックの性質は研究されていない。高速化が進む無線ネットワークでは SR ARQ による再送が重要であり、そのよ

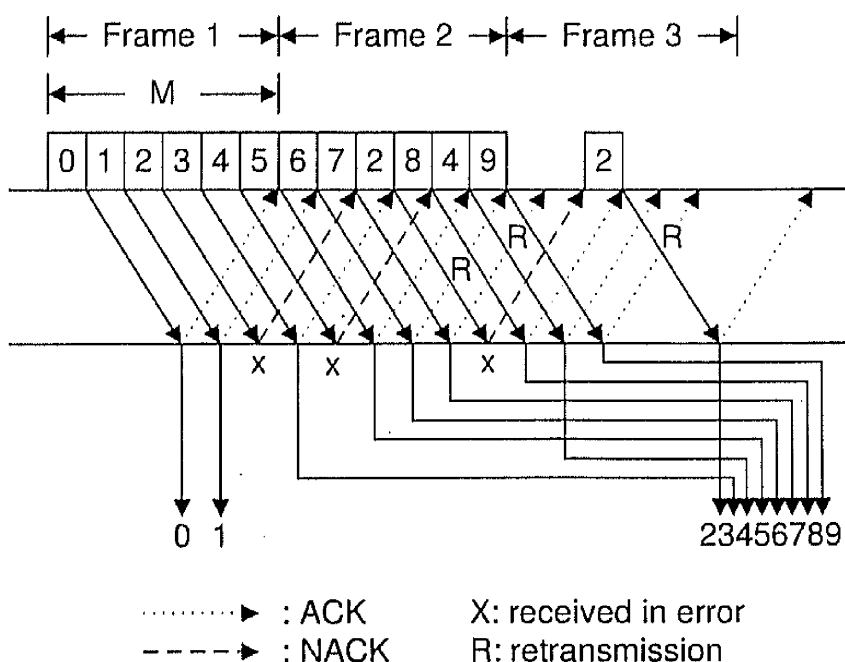


図 2.1 AR ARQ プロトコルのシーケンス例

うな無線ネットワークを設計する上でバーストに含まれるパケット数やバースト発生間隔の確率分布などの正確な把握は重要である。このような背景から、この章ではバーストに含まれるパケット数及びバーストの発生間隔の確率分布について解析を示す。更に特定の値より長さの大きなバーストの平均発生間隔についても解析を行う。

また、順序制御を全く行わないか部分的に行えば、バースト的なパケットトラヒックは回避または緩和することが出来る。しかし、その代わりにパケットの順序逆転が発生する。順序逆転パケットの発生と順序の正しいパケットトラヒックのバースト性との間にはトレードオフが存在する。この章では、順序制御により保留されるパケット数を制限した場合について、順序逆転パケット数の確率分布についても解析を行う。また解析と同じ条件で行ったシミュレーション結果との比較により解析の妥当性を確認する。

この章は以下のように構成される。2.2 節で解析モデルについて述べ、2.3 節でバースト長とバーストの発生間隔に関する確率分布を導出する。2.4 節で数値計算例と同じ条件で行ったシミュレーションとの比較を示す。2.5 節で順序逆転パケット数の確率分布を求め数値計算結果をシミュレーション結果と比較する。最後に 2.6 でまとめを述べる。

2.2 解析モデルと仮定

2.2.1 解析における仮定と条件

順序制御によるバースト的なパケット出力を評価するために、以下を仮定して解析を行った。

1. 伝送リンクにおいてパケットの伝送誤りはパケット誤り率 ε によりランダムに発生する。
2. パケットが伝送誤りなく受信された場合、肯定応答 (ACK) が送信側に返送される。伝送誤りが発生した場合は、否定応答 (NACK) が返送される。ACK 及び NACK はパケットが受信されると直ちに送信されるものとする。
3. ACK 及び NACK には伝送誤りは発生しないものとする。
4. パケットの長さを固定とし、時間を 1 パケットを送信するのに十分な大きさにスロット化する。
5. NACK を受信すると送信側は該当するパケットを直ちに再送する。再送は新規パケットより優先して送信する。
6. 各スロットにおいて、送信側は常に送信すべきパケットを有するものとする。これは伝送リンクの帯域をフルに利用することを意味する。
7. パケットの送信開始からそのパケットに対する ACK または NACK を受信するまでの連続したスロットをフレームと呼ぶ。図 2.1 に示すように 1 フレームは M スロットから構成され、フレーム内のスロットに 0 から $M - 1$ の番号を割り当てる。
8. フレームを構成するスロット数 M は、ラウンドトリップ時間に相当するスロット数にパケットの送信時間に相当する 1 スロットを加えたものとなる。
9. 連続するフレームにおいて同じ番号 (位置) のスロットの集合をチャンネルと呼ぶ。フレームには 0 から $M - 1$ までのチャンネルが存在する。
10. 最大再送回数を N_r とする。もし、 N_r 回再送されたパケットが伝送誤りにより受信されなかった場合、順序制御のために受信側で保留されていたパケットは解放され、上位層に転送されるものとする。解析を単純にするため、この解放タイミングは最後の再送パケットが受信側に到着した時点とする。

最大回数の再送の後に伝送誤りによりパケットの受信に失敗した場合、もし受信側で順序制御により保留されているパケットが存在しなければ上位層に転送されるパケットもない。ここでの解析ではこのような場合でも長さが 0 のバーストが出力されたものと見なす。伝送誤りが発生しない状況が続くと、各パケットは正しく受信され、1 パケットが上

位層に転送される。このような場合、1パケットから成るバーストが出力されているものと見なす。以下の解析でパケットの受信完了の表現を用いるが、これはパケットを正しく受信した場合及び最大回数の再送を行ってもパケットを正しく受信しなかった場合の両方を含んでいる。

上記の条件では、1パケットを送信後、 $M - 1$ スロット後に ACK または NACK が帰ってくることを仮定している。時間をスロット化しているので、時刻 t においてパケットの送信が完了したとすると、ACK または NACK は次の条件を満足する時刻 t_{ACK} に返ればよい。ここで、 Δ を1スロット時間とする。

$$t + (M - 2)\Delta < t_{ACK} \leq t + (M - 1)\Delta \quad (2.1)$$

これから、伝搬遅延時間 D_{prop} が次の条件を満足する場合に1フレームが M スロットにより構成されることになる。

$$(M - 2)\Delta/2 < D_{prop} \leq (M - 1)\Delta/2 \quad (2.2)$$

このように、条件を満足する伝搬遅延時間の値は1スロット幅の半分の誤差が許容されることが分かる。

2.2.2 解析モデル

パケットが一つのチャンネルのスロットで送信されると、そのパケットの再送は受信完了まで常に同じチャンネルの後続スロットで行われる。この一つのチャンネルでのパケットの再送動作は Stop-and-Wait プロトコルと等価である。これから、上記の解析モデルでは SR ARQ を独立した M 個のチャンネルにおける Stop-and-Wait プロトコルの集合としてモデル化できる。図 2.2 にスロット、チャンネル、フレームの例を示す。

図 2.3 は図 2.1 に示す SR ARQ のシーケンス例において、この SR ARQ を独立した M 個の Stop-and-Wait プロトコルの集合に分解した場合の例を示している。

このような独立したチャンネルの集合からバースト長やバースト発生間隔の確率分布を求めることが可能である。以下の解析では、一つのチャンネルに注目して解析を行うが、フレームの位相は任意であるため、一般性を損なうものではない。

一つのチャンネルの状態は、そのチャンネルにおいて一つのパケットの連続受信失敗回数として定義できる。例えば、図 2.4 はこのような定義に基づいたチャンネル #2 の状態変化を示している。伝送誤りによりパケットの受信に失敗した場合は状態の値が1増加し、パケットの受信に成功した場合は状態の値が0に戻る。最大回数の再送を行ってパケットの受信に失敗した場合も、状態の値は0に戻る。複数のパケットを伝送誤りなしに連続して受信した場合、チャンネルの状態は連続したフレームで0に留まる。このような状態の定義

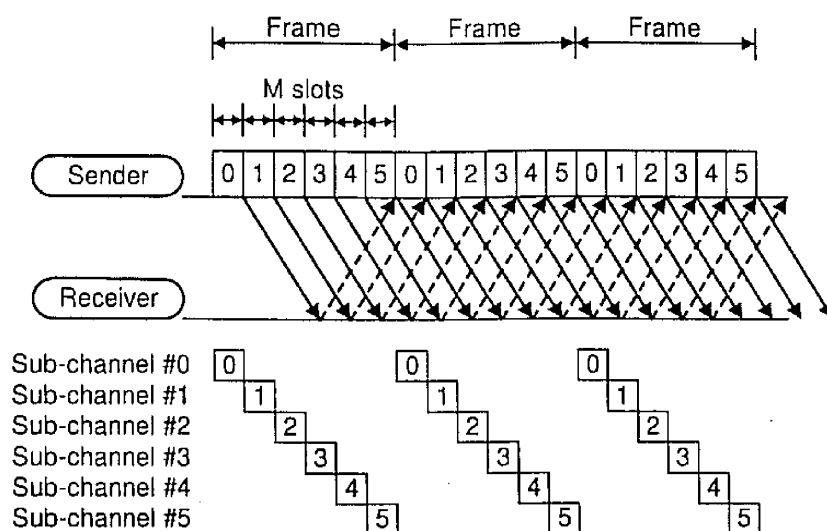


図 2.2 スロット、チャネル、フレームの例

に基づく状態遷移は図 2.5 に示す状態遷移図により表すことができる。これから状態確率 $p_s(k)$ 及び最大再送回数を制限したことにより生じるパケット損失確率 p_L は以下の式で与えられる。

$$p_s(k) = \frac{(1 - \varepsilon)\varepsilon^k}{1 - \varepsilon^{N_r+1}} \quad (2.3)$$

$$p_L = \varepsilon^{N_r+1}. \quad (2.4)$$

2.3 バースト長とバースト発生間隔の確率分布

この節では、チャンネル #0 においてパケットの受信を完了した時点のバースト長とバースト発生間隔の確率分布を求める。解析の大まかな流れは以下のようなになる。

- 最初にパケットの受信を完了した時点でバーストが発生する確率を求める。
- 次にバースト生成スロットを定義する。バーストは基本的にはこのバースト生成スロットで送信されたパケットにより構成される。また、バースト生成スロットを二つのグループに分割するブロッキングパケットを定義する。ここで、同じグループに属するバースト生成スロットで送信されたパケットが同じバーストを構成する。

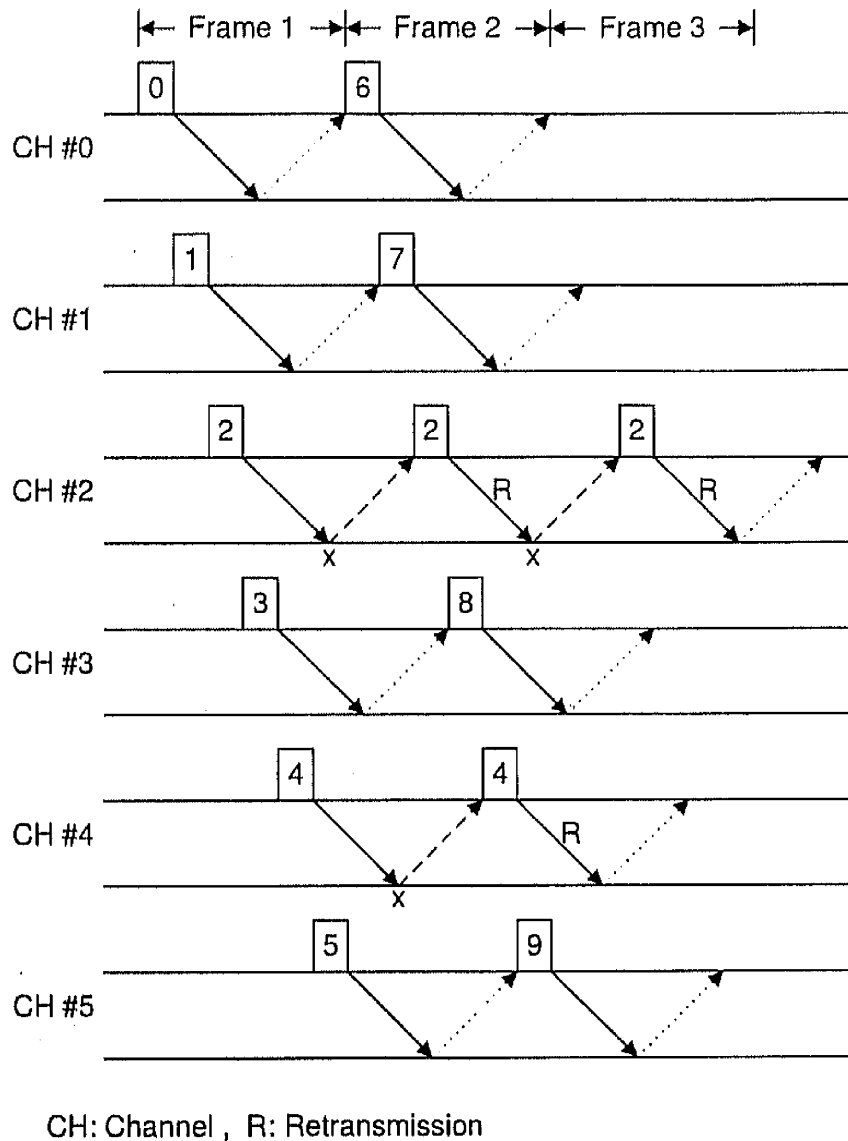


図 2.3 独立した複数チャネルにおける Stop-and-Wait プロトコルによる SR ARQ のシーケンス例

- バースト生成スロットのグループが与えられた場合について、バースト長の確率分布を計算する。
- バースト生成スロットの確率を考慮したバースト長の確率分布を計算する。
- バースト発生間隔の確率分布を計算する。これはあるバーストの生成から次のブロッキングパケットの受信完了までに期間である。
- 最後に、上記で求めた確率分布より特定の長さ以上のバーストが発生する平均間隔

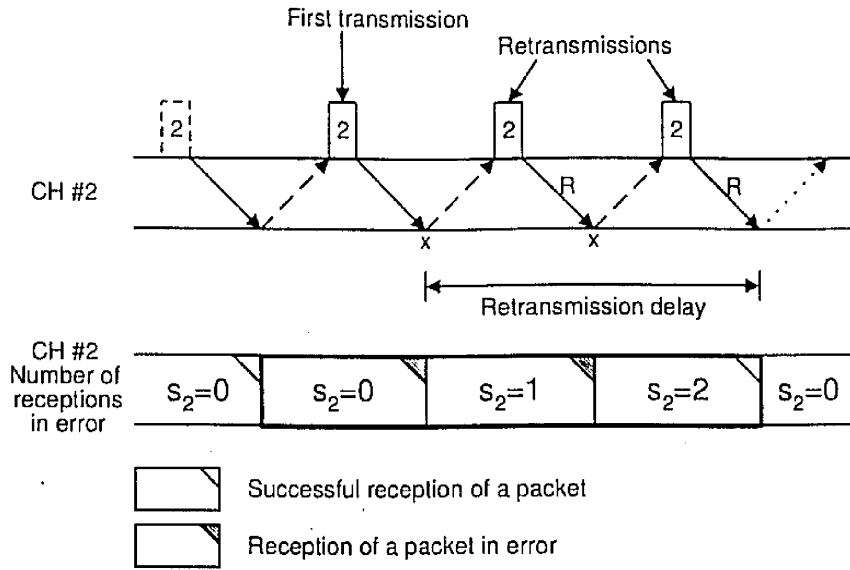


図 2.4 チャンネル #2 の状態遷移例

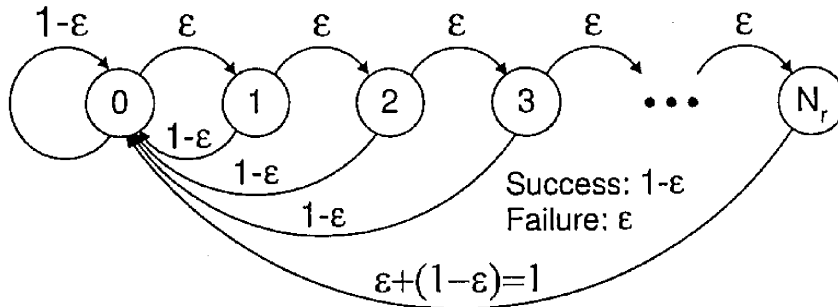


図 2.5 受信失敗回数に基づく状態遷移図

を求める。

2.3.1 パケットの受信を完了した場合にバーストが発生する確率

パケットの受信を完了した場合、バーストが常に発生するとは限らない。受信したパケットより前に送信されたパケットでまだ受信が完了していないものがあれば、受信したパケットは順序制御のために保留される。パケットの受信を完了した場合、そのパケットより前に送信されたパケットすべての受信が完了していればそのパケットの受信を待って保留されていたパケットが解放され、同時に上位層に転送される。

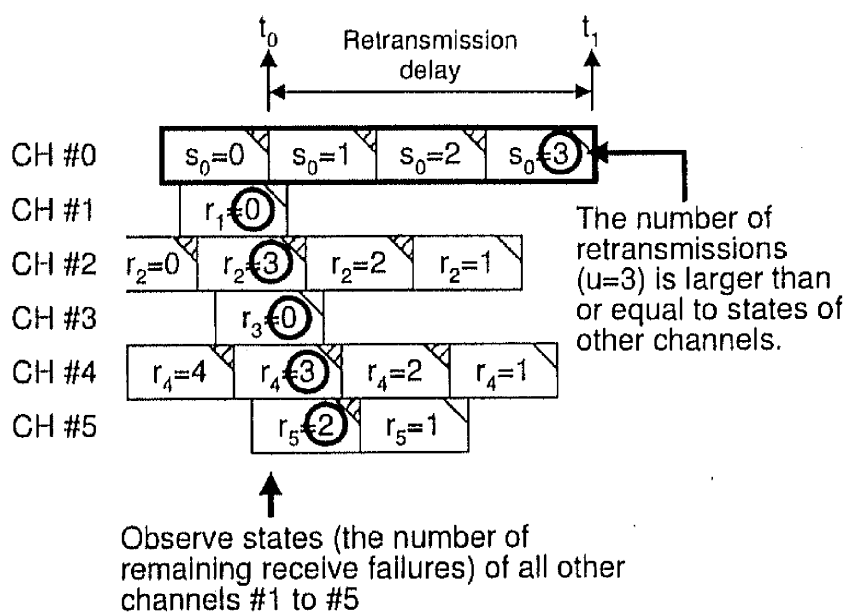


図 2.6 バーストが生成される場合の例

以下で特定のチャンネル #0 に注目して解析を進める。あるパケットが u 回の受信失敗の後に受信完了したとする。受信完了時点で他のすべてのチャンネル #1 から # $(M-1)$ を観測する。もし、他のすべてのチャンネルの状態が u 以下であれば、チャンネル #0 のパケットがすべてのチャンネルのパケットの中で最も古いので、バーストが生成される。図 2.6 はバーストが生成される場合の例を示している。バーストが生成される確率を P_{bg} とすると、以下のように求められる。

$$P_{bg} = \sum_{u=0}^{N_r} p_t(u) Q(u)^{M-1} \quad (2.5)$$

ここで、 $p_t(u)$ は u 回の受信失敗（再送）の後にパケットの受信が完了する確率である。この確率は、 u の受信失敗の後に受信に成功する場合 ($u \leq N_r$) と最大回数の再送の後に受信に失敗する場合 ($u = N_r$) とから計算される。

$$p_t(u) = \begin{cases} (1-\varepsilon)\varepsilon^u & u = 0, 1, \dots, N_r - 1 \\ \varepsilon^{N_r} & u = N_r \end{cases} \quad (2.6)$$

また、 $Q(n)$ はあるチャンネルにおける受信失敗回数が n 以下となる確率で、以下で与えられる。

$$Q(n) = \sum_{k=0}^n p_s(k) = \frac{1 - \varepsilon^{n+1}}{1 - \varepsilon^{N_r+1}} \quad (2.7)$$

2.3.2 ブロッキングパケットとバースト生成スロット

チャンネル #0 でパケットの受信が完了した時点で、他のチャンネルの状態を観測した場合に、チャンネル # i の受信失敗回数がチャンネル #1 から # $(M - 1)$ の中で最大であるとする。図 2.7 の例では $i = 3$ となっている。より正確には、チャンネル # i が以下の条件を満足する場合である。

- チャンネル # i の受信失敗回数がチャンネル #1 から # $(i - 1)$ の受信失敗回数より大きい。
- チャンネル # i の受信失敗回数がチャンネル # $(i + 1)$ to # $(M - 1)$ の受信失敗回数より大きいか等しい。

チャンネル #0 におけるパケットの受信が時刻 t_0 で完了した時に、この時点で保留されていたパケットすべてが解放されるとは限らない。解放されるパケットはチャンネル # i のパケットが最初に送信された時点より以前に送信されたパケットに限られる。以下の説明で、チャンネル # i のパケットのようなパケットをブロッキングパケットと呼ぶ。図 2.7 において、チャンネル #3 のパケットがブロッキングパケットに相当する。また、この図で網掛けを施した複数スロットで受信したパケットは時刻 t_0 におけるバーストを構成する可能性が高い。このようなスロットをバースト生成スロットと呼ぶ。

チャンネル # i の状態（受信失敗回数）を v とし、このチャンネルのパケットがブロッキングパケットとなる確率を $B(v, i)$ とする。各チャンネルは独立なので、チャンネル #1 から # $(M - 1)$ の状態が、それぞれ s_1, \dots, s_{M-1} となる確率は、各チャンネルの状態確率の積で与えられる。

$$P_{rob}(s_1, \dots, s_i, \dots, s_{M-1}) = \prod_{k=1}^{M-1} p_s(s_k) \quad (2.8)$$

$B(v, i)$ は以下のように計算できる。

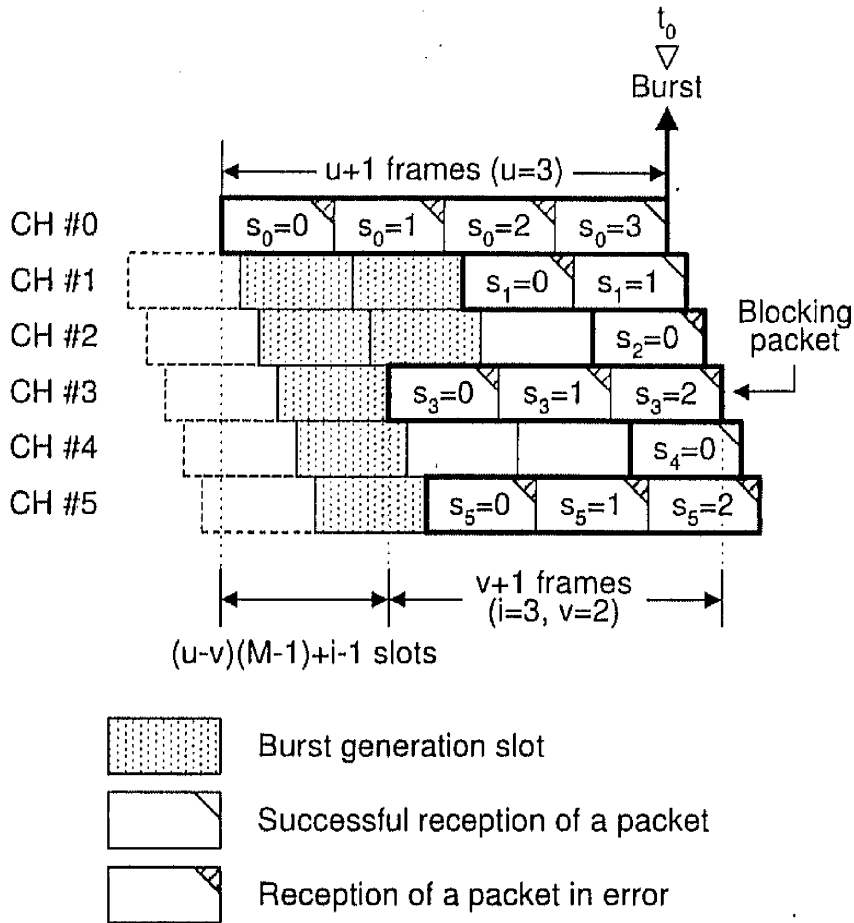


図 2.7 バースト生成スロットとブロッキングパケットの例

$$\begin{aligned}
 B(v, i) &\triangleq P_{\text{rob}}(s_1 < v, \dots, s_{i-1} < v, s_i = v, s_{i+1} \leq v, \dots, s_{M-1} \leq v) \\
 &= p_s(v) \sum_{s_1=0}^{v-1} \dots \sum_{s_{i-1}=0}^{v-1} \sum_{s_{i+1}=0}^v \dots \sum_{s_{M-1}=0}^v \prod_{k=1, k \neq i}^{M-1} p_s(s_k) \\
 &= p_s(v) Q(v-1)^{i-1} Q(v)^{M-1-i}
 \end{aligned} \tag{2.9}$$

2.3.3 バースト生成スロットが与えられた条件でのバースト長の確率分布

チャンネル #0 でパケットの受信を完了した時点で、このチャンネルの状態（パケットの受信失敗回数）を u 、チャンネル # i の状態を v とする。また $g(u, v, i)$ をバースト生成スロット数とする。例えば、図 2.7 の場合、 $u = 3$ 、 $v = 2$ 、 $i = 3$ 、 $M = 6$ 、 $g(u, v, i) = 7$ であ

る。この図から $g(u, v, i)$ は以下で与えられることが分かる。

$$g(u, v, i) = (u - v)(M - 1) + i - 1 \quad (2.10)$$

バースト生成スロット数が与えられた場合のバーストに含まれるパケット数が b となる条件付確率 $P_b(b|g(u, v, i))$ を求める必要がある。この確率を厳密に求めることは難しい。しかし、最大再送回数の制限がない場合 (N_r が大きいまたは $\epsilon^{N_r} \approx 0$)、この確率は二項分布により求められる。この二項分布を最大再送回数が制限された場合にも近似として用いる。

$$P_b(b|g(u, v, i)) \approx \binom{g(u, v, i)}{b} (1 - \epsilon)^b \epsilon^{g(u, v, i) - b} \quad (2.11)$$

以下、最大再送回数の制限がない場合について、この式が成り立つことを証明する。次の二つの例外的な場合に留意する必要がある。

バーストを構成するパケットがバースト生成スロットにおいては受信に失敗するが、バースト生成スロットに続く後のスロットで受信に成功する場合がある。図 2.8 において、チャンネル #2 はこの場合の例を示している。フレーム F0 でパケット pkt_a が正しく受信されているが、このパケットはチャンネル #0 のパケットの最初の送信時刻より前に送信されているので、時刻 t_0 のバーストには含まれない。フレーム F1 でパケット pkt_b が正しく受信されており、このパケットは時刻 t_0 のバーストに含まれる。パケット pkt_c は最後のバースト生成スロットで受信されるが、受信失敗となっている。しかし、同じチャンネルの後続のスロットで正しく受信される。このパケットはブロッキングパケットより前に最初の送信が行われており、その受信成功は時刻 t_0 より前であるため、時刻 t_0 のバーストに含まれる。

パケットがバースト生成スロットで正しく受信されたにもかかわらず、バースト生成スロットに対応するバーストに含まれない場合もある。図 2.8 のチャンネル #4 はこの場合の例を示している。フレーム F0 でパケット pkt_d は伝送誤りのため受信されず、同じチャンネルの後続バースト生成スロットで正しく受信されている。この場合、パケット pkt_d の最初の送信はチャンネル #0 のパケットより先に行われているので、このパケットは時刻 t_0 のバーストに含まれない。

これらを考慮すると、フレーム F0 でチャンネル # j ($j \neq 0$ and $j \neq i$) のパケットが正しく受信されたか否かの二つの場合を考える必要がある。

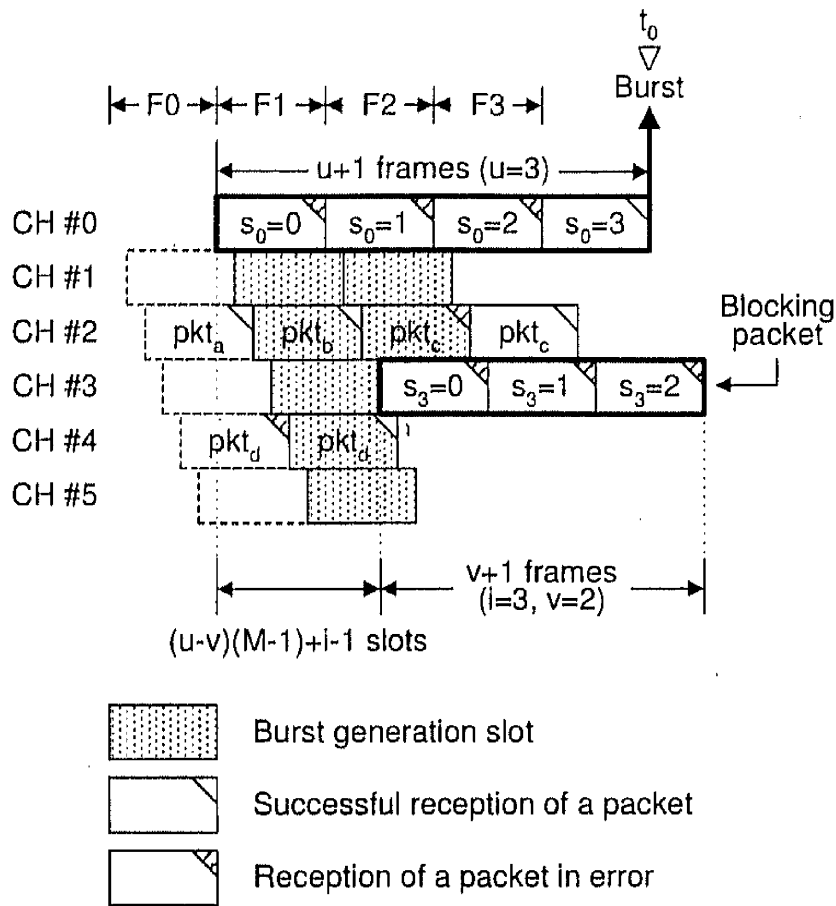


図 2.8 バースト生成スロットとバーストとの対応が例外的となる場合

2.3.3.1 フレーム F0 においてチャンネル # j のパケットが正しく受信された場合

この場合、チャンネル # j のバースト生成スロットで正しく受信したパケットはすべてバーストに含まれる。 $P_{sbl}(b_j)$ をチャンネル # j のバースト生成スロットにおいて正しく受信したパケット数が b_j である確率とする。チャンネル # j のバースト生成スロットの最後のスロットでパケットの受信に失敗した場合、同じチャンネルの後続のスロットで必ずこのパケットが時刻 t_0 までに受信され、バーストに含まれる。これからバースト生成スロットの最後のスロットでは、等価的にパケットが必ず正しく受信されることが出来る。 g_j をチャンネル # j のバースト生成スロット数とする。確率 $P_{sbl}(b_j)$ はチャンネル # j のバースト生成スロット数から 1 を差し引いた数（最後のスロット分を除外）からの二項分布により求めることができる。

$$P_{sb1}(b_j) = \binom{g_j - 1}{b_j - 1} (1 - \varepsilon)^{b_j - 1} \varepsilon^{g_j - b_j} \quad (2.12)$$

2.3.3.2 フレーム F0 においてチャンネル #j のパケットが受信されなかった場合

この場合、チャンネル #j のバースト生成スロット及びその後続スロットで最初に正しく受信されたパケットは注目するチャンネル #0 のパケットより、先に新規送信されている。従って、このパケットは時刻 t_0 のバーストに含まれない。この場合も、チャンネル #j のバースト生成スロットの最後のスロットでは、等価的にパケットが必ず正しく受信されると考えることが出来るので、正しく受信されるパケットは少なくとも 1 個存在する。時刻 t_0 のバーストへの寄与に関して、この 1 個は常に除外して考えることが出来る。 $P_{sb2}(b_j)$ をチャンネル #j のバースト生成スロットにおいて正しく受信したパケット数が b_j である確率とする。確率 $P_{sb2}(b_j)$ の以下のように二項分布を用いて計算することができる。

$$P_{sb2}(b_j) = \binom{g_j - 1}{b_j} (1 - \varepsilon)^{b_j} \varepsilon^{g_j - 1 - b_j} \quad (2.13)$$

2.3.3.3 バースト生成スロットが与えられた条件でのバースト長の確率分布

$P_{sb}(b_j)$ をチャンネル #j において、バーストに寄与するパケット数が b_j である確率とする。以下の計算から、確率 $P_{sb}(b_j)$ は g_j からの二項分布となることが分かる。

$$\begin{aligned} P_{sb}(b_j) &= (1 - \varepsilon)P_{sb1}(b_j) + \varepsilon P_{sb2}(b_j) \\ &= \binom{g_j}{b_j} (1 - \varepsilon)^{b_j} \varepsilon^{g_j - b_j} \end{aligned} \quad (2.14)$$

バースト長はチャンネル #0 を除く他のすべてのチャンネルからのパケットの総和である。各チャンネルからのパケット数の分布が同じパケットの誤り率による二項分布となるので、パケットの総和チャンネル #0 を除く他のすべてのチャンネルのバースト生成スロットの総和 $g(u, v, i) = \sum_{j=1}^{M-1} g_j$ から二項分布で求めることができる。以上により (2.11) が成り立つことが分かる。

この式によるシンプルな結果は、バースト生成スロットで正しく受信したパケットで差し引かれる分と、バースト生成スロットの後続スロットで正しく受信したパケットが加わる分とがバランスするものと直感的に考えられる。同様のバランスは最大再送回数が制限された場合でも期待できると直感的には考えられるので、(2.11) 式を近似として最大再送回数が制限された場合にも用いる。

2.3.4 バースト生成スロット数の確率分布を考慮したバースト長の確率分布

上で述べたように、 u, v, i が与えられるとバーストに含まれるパケット数に関する確率分布はバースト生成スロット数 $g(u, v, i)$ から近似的に二項分布で与えられる。最終的にバースト長の確率分布は、以下の条件に対する u, v, i の確率を考慮して計算できる。

- チャンネル #0 において u 回の受信失敗の後にパケットの受信が完了する。
- チャンネル #0 において受信が完了した時に、ブロッキングパケットがチャンネル # i に存在し、その受信失敗回数がその時点で v である。

2.3.4.1 チャンネル #0 のパケットが u 回の受信失敗の後に正しく受信される場合 ($0 \leq u \leq N_r$)

■他のチャンネルで再送が行われる場合 ($v > 0$) この場合、チャンネル #0 において u 回の受信失敗の後にパケットを受信完了した時点で、他のチャンネルで再送が行われている。ブロッキングパケットがチャンネル # i において存在し、その時点での受信失敗回数が v であるとする。 $P_{s1}(u, v, i)$ をこの事象の確率であるとする。ここで、 $v > 0$ かつ $i > 0$ である。 $P_{s1}(u, v, i)$ は以下で与えられる。

$$P_{s1}(u, v, i) = (1 - \epsilon)\epsilon^u B(v, i) \quad (2.15)$$

■他のすべてのチャンネルで再送が行われていない場合 ($v = 0$) チャンネル #0 において u 回の受信失敗の後にパケットを受信完了した時点で、他のすべてのチャンネルにおいて再送が行われていない場合、次のチャンネル #1 のパケットがブロッキングパケットとなり、その受信失敗回数は 0 である。 $P_{s1}(u, 0, 1)$ をこの事象の確率とすると、

$$P_{s1}(u, 0, 1) = (1 - \epsilon)\epsilon^u p_s(0)^{M-1} \quad (2.16)$$

チャンネル #0 で正しく受信したパケットはバーストに含まれる。バーストに含まれるパケット数が b となる確率 $P_{b2}(b)$ は、 $g(u, v, i)$ 個のスロットの中で $b - 1$ 個のパケットが正しく受信される確率である。2.3.3 で説明したようにこの確率は二項分布により求められる。

$$\begin{aligned}
P_{b1}(b) \approx & \sum_{u=1}^{N_r} \sum_{v=1}^u \sum_{i=1}^{M-1} \binom{g(u,v,i)}{b-1} (1-\varepsilon)^{b-1} \varepsilon^{g(u,v,i)-b+1} P_{s1}(u,v,i) \\
& + \sum_{u=1}^{N_r} \binom{g(u,0,1)}{b-1} (1-\varepsilon)^{b-1} \varepsilon^{g(u,0,1)-b+1} P_{s1}(u,0,1)
\end{aligned} \quad (2.17)$$

ここで、 $g(u,v,i) \geq b-1$ 、 $g(u,0,1) \geq b-1$ である。

2.3.4.2 チャンネル 0 で最大回数の再送の後にパケットの受信に失敗した場合 ($u = N_r$)

$P_{s2}(v,i)$ を他のチャンネルで再送が行われている場合で、ブロッキングパケットがチャンネル # i において存在し、その時点での受信失敗回数が v である確率とする。 $P_{s2}(0,1)$ を他のすべてのチャンネルで再送が行われていない場合の確率とする。この場合、次のチャンネル #1 のパケットがブロッキングパケットとなり、その受信失敗回数は 0 である。これらの確率は 2.3.4 と同様の方法で求めることができる。

$$P_{s2}(v,i) = \varepsilon^{N_r+1} B(v,i) \quad (2.18)$$

$$P_{s2}(0,1) = \varepsilon^{N_r+1} p_s(0)^{M-1} \quad (2.19)$$

チャンネル #0 でパケットの受信に失敗しているので、バーストに含まれるパケット数が b となる確率は $g(N_r, v, i)$ 個のスロットから b 個のパケットが正しく受信される確率として求められる。2.3.3 で説明したように、この確率も二項分布により以下のように求められる。

$$\begin{aligned}
P_{b2}(b) \approx & \sum_{v=1}^{N_r} \sum_{i=1}^{M-1} \binom{g(N_r,v,i)}{b} (1-\varepsilon)^b \varepsilon^{g(N_r,v,i)-b} P_{s2}(v,i) \\
& + \binom{g(N_r,0,1)}{b} (1-\varepsilon)^b \varepsilon^{g(N_r,0,1)-b} P_{s2}(0,1)
\end{aligned} \quad (2.20)$$

ここで、 $g(N_r, v, i) \geq b$ 、 $g(N_r, 0, 1) \geq b$ である。

2.3.4.3 バーストが生成される条件でのバースト長の分布

$P_{bp}(b)$ をバーストが生成される条件でのバースト長の分布確率とすると、以下のようになる。

$$P_{bp}(b) \approx \frac{P_{b1}(b) + P_{b2}(b)}{P_{bg}} \quad (2.21)$$

2.3.5 バースト発生間隔分布

スロットを時間の単位としてバースト発生間隔分布を求める。チャンネル #0 においてパケットの受信が u 回の受信失敗の後に完了した場合に、ブロッキングパケットがチャンネル # i に存在しその受信失敗回数が v であるとする。次のバーストはこのブロッキングパケットの受信が完了した時点で生じる。もし、ブロッキングパケットが更に w 回の受信失敗の後に受信完了したとすると、次のバースト発生までのバースト発生間隔は図 2.9 を参照することにより $wM + i$ スロットとなることが分かる。

$p_{intvl}(w, i)$ をこの場合の確率とする。ここで、 $0 \leq w \leq (N_r - v)$ 。 $w = 0$ となる場合は、ブロッキングパケットの受信がチャンネル #0 においてパケットの受信が完了したフレームと同じフレームでブロッキングパケットの受信が完了する場合を意味する。 $p_w(w|v)$ を現在の受信失敗回数が v の条件で、更に w の再送により受信が完了する確率とする。

$$p_w(w|v) = \begin{cases} (1 - \varepsilon)\varepsilon^w & 0 \leq w < (N_r - v) \\ \varepsilon^{(N_r - v)} & w = N_r - v \end{cases} \quad (2.22)$$

バースト発生間隔が $wM + i$ スロットとなる確率 $P_{intvl}(w, i)$ は、 u と v に関する可能なすべての組み合わせに対して $p_w(w|v)$ の和を求めることにより計算できる。

$$\begin{aligned} P_{intvl}(w, i) &= \sum_{v=0}^{N_r - w} \sum_{u=v}^{N_r} p_i(u) p_w(w|v) B(v, i) \\ &= \sum_{v=0}^{N_r - w} \varepsilon^v p_w(w|v) B(v, i) \end{aligned} \quad (2.23)$$

2.3.6 特定の値より長さが大きいバーストの平均発生間隔

$P_B(n)$ をバーストの長さが n 以上となる確率とする。この確率は式 (2.21) から次のように計算される。

$$P_B(n) = 1 - \sum_{b=0}^{n-1} P_{bp}(b) \quad (2.24)$$

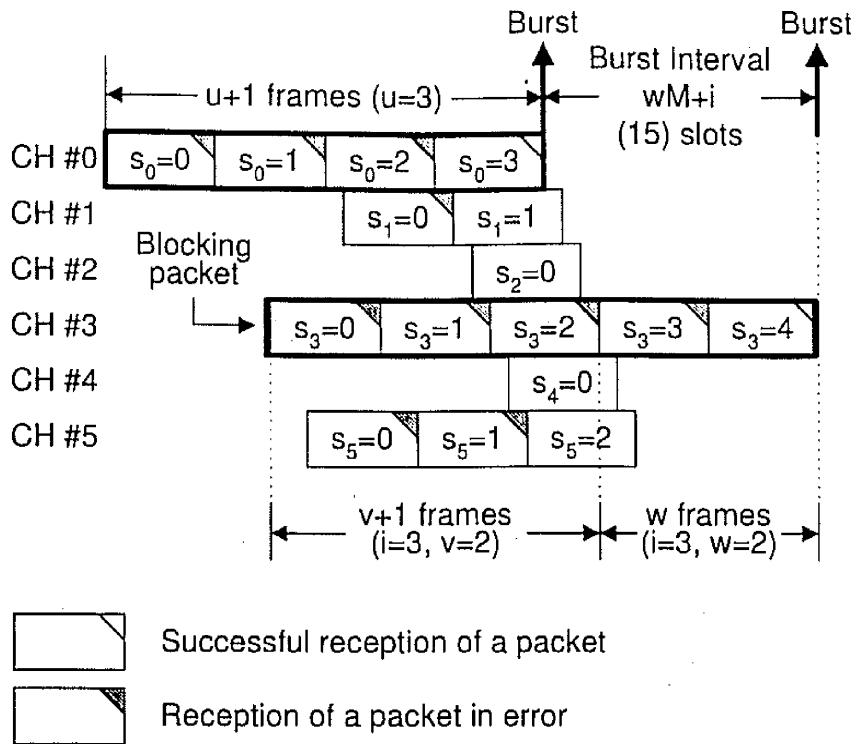


図 2.9 バースト発生間隔の例

長さが n 以上となるバーストがランダムに発生すると仮定すると、このようなバーストは平均で $1/P_B(n)$ スロットごとに発生する。 m_b を (2.23) 式で与えられる平均バースト発生間隔とする (スロット単位)。また $m_B(n)$ を長さが n 以上となるバーストの平均発生間隔 (スロット単位) とすると、以下のように計算される。

$$m_B(n) = m_b / P_B(n) \tag{2.25}$$

2.4 数値計算結果と考察

2.4.1 バースト長とバースト発生間隔の確率分布

SR ARQ を 1.5 Mbit/s の衛星回線に適用し、パケット長を 1500 Byte、衛星回線の伝搬遅延時間を 0.255 sec とした場合について数値計算結果例を示す。これらの条件からフレームを構成するスロット数 M は 64 となる。図 2.10 及び図 2.11 は、最大再送回数 N_r を 3、パケット誤り率 ε を 0.1 と 0.01 とした場合のバースト長に関する確率分布を示し

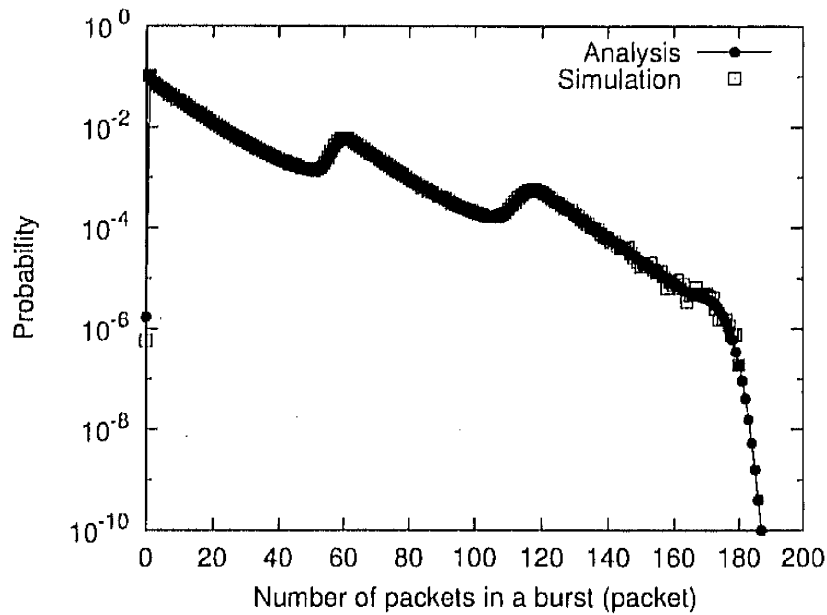


図 2.10 バースト長に関する確率分布 ($M = 64, N_r = 3, \varepsilon = 0.1$)

ている。図 2.12 及び図 2.13 は同じ条件におけるバースト発生間隔の確率分布を示している。これらの図で、解析を確認するためのシミュレーション結果も併せて示している。解析より求めた数値計算結果はシミュレーション結果と良く一致しており、本章で述べた解析の妥当性を示している。

一般的な傾向としてバースト長が大きくなるとその確率が小さくなるのが分かる。同様の傾向はバースト発生間隔の確率分布においても見られる。しかし、パケット誤り率 ε が大きな値をとる場合 ($\varepsilon = 0.1$)、 ε が小さな値の場合に比べ、長さの大きなバーストが発生する確率が比較的高い値のままで残る傾向が見られる。例えば、 ε が 0.1 の場合、長さが 130 パケットのバーストが 10^{-4} より大きな確率で発生する。一方、 ε が 0.01 の場合、長さが 64 未満のバーストの発生確率はこの値より小さい。しかし、 ε が 0.01 の場合、全体的に長さの大きなバーストの発生確率は小さくなるが、 $(M-1)$ スロットの倍数で確率のピークが見られる。これは次に様に説明できる。パケット誤り率が小さい場合、喪失したパケットが再送されている間、この間に送信された後続パケットは正しく受信される可能性が高く、これらの後続パケットがバーストを形成する (バーストが断片化されない)。一方、パケット誤り率が大きくなると、複数パケットの再送が並行して行われる場合が多くなり、バーストが断片化されるため確率分布のピークが減少する。

バースト長の確率分布の解析において $\varepsilon^{N_r} \approx 0$ の条件による近似を用いたが、この近似は ε が小さく N_r が大きい場合に有効である。近似の影響を評価するため、パケット

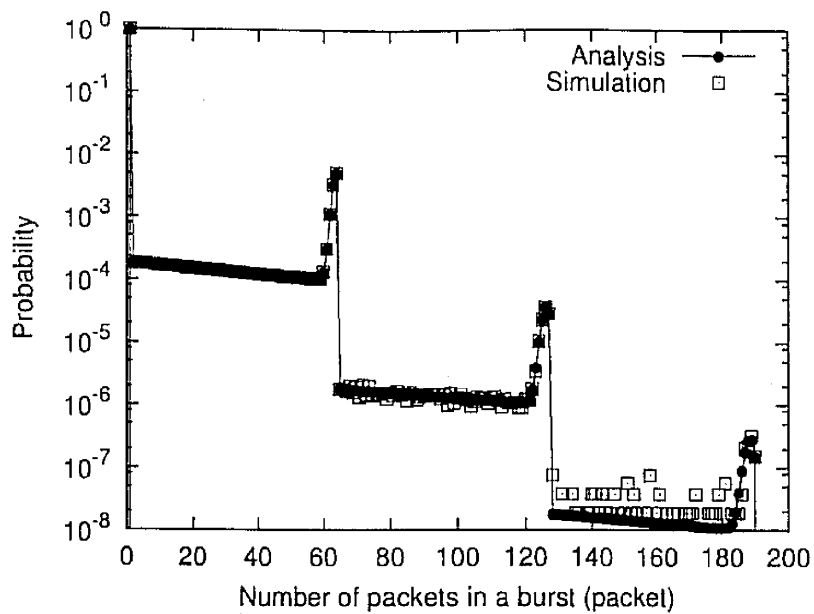


図 2.11 バースト長に関する確率分布 ($M = 64, N_r = 3, \varepsilon = 0.01$)

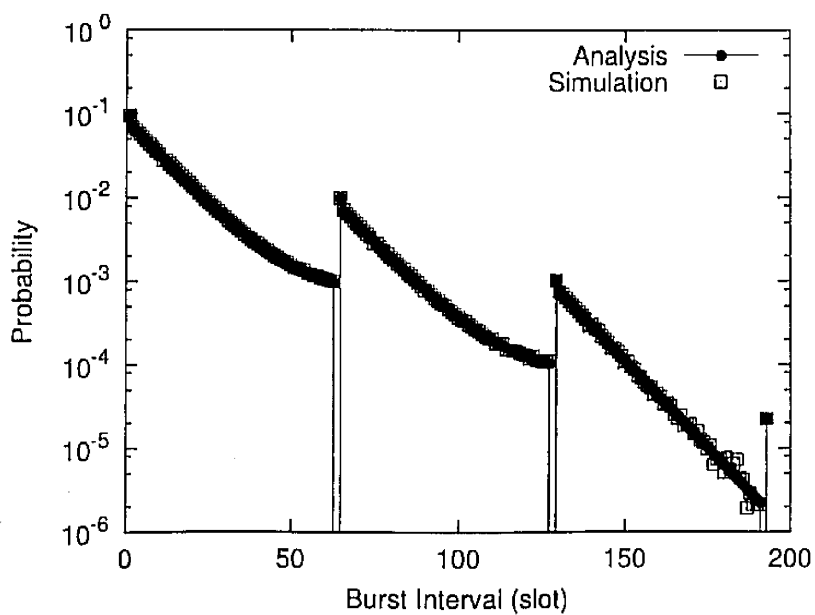


図 2.12 バースト発生間隔に関する確率分布 ($M = 64, N_r = 3, \varepsilon = 0.1$)

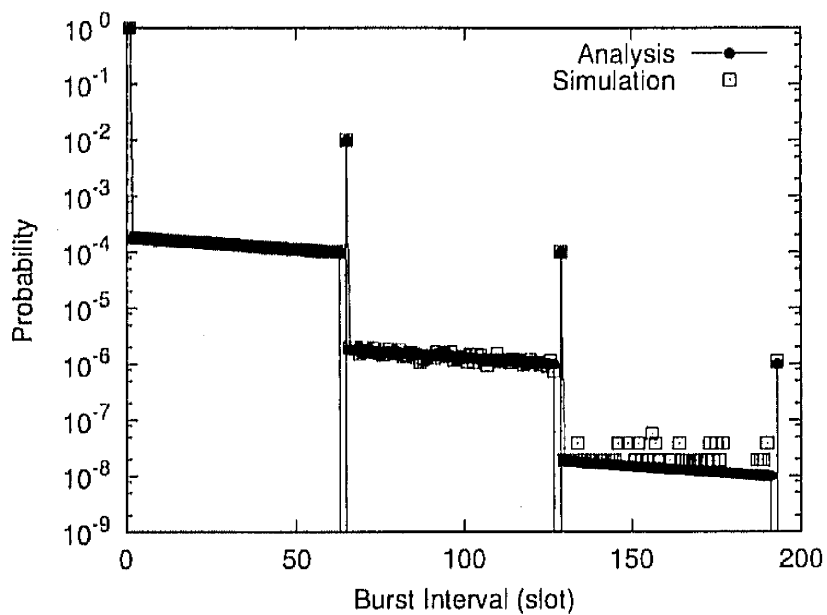


図 2.13 バースト発生間隔に関する確率分布 ($M = 64, N_r = 3, \varepsilon = 0.01$)

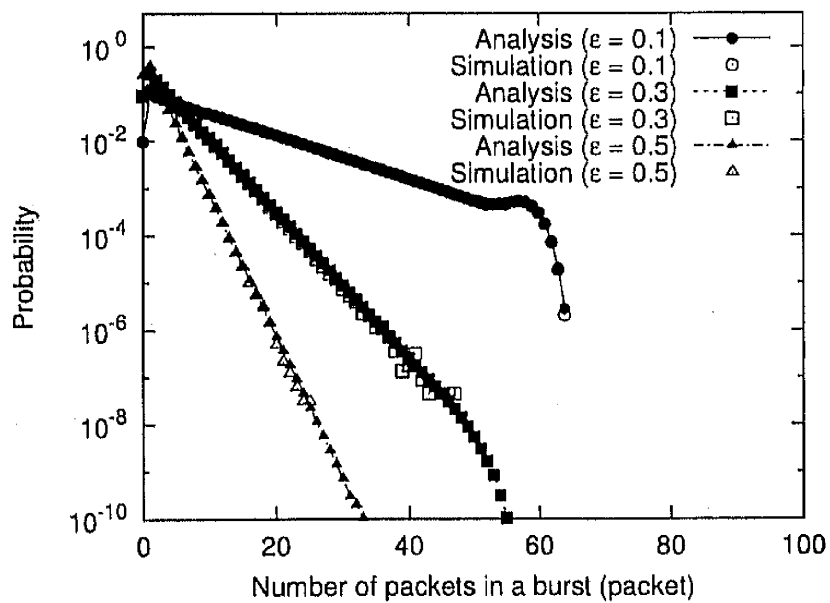


図 2.14 バースト発生間隔に関する確率分布 ($M = 64, N_r = 1, \varepsilon = 0.1, 0.3, 0.5$)

誤り率 ε が大きく、最大再送回数 N_r が 1 の場合についてシミュレーションを行った。図 2.14 は ε が 0.1、0.3、0.5 の場合の結果を示している。解析による数値計算結果はシミュレーションによる結果とこのような条件でも良く一致している。これらの結果からこの章で述べた解析はこのような条件においても適用できることが明らかである。

パケット誤り率が大きな値をとると長さの大きなバーストの発生する確率が減少することが分かる。パケット誤り率の大きい条件で、バーストの長さが b となる確率は概ね $(1-\varepsilon)^{(b-1)}$ に比例する（幾何分布）。このため、確率分布の傾きは $\log_{10}(1-\varepsilon)$ に近くなる。 ε の値が小さい場合、この傾きは -0.43ε で近似される。この式及び図 2.14 から分かるように、パケット誤り率 ε が大きい場合、確率分布の傾きは大きくなる事が分かる。

図 2.15 はパケット誤り率 ε を変化させた場合の平均バースト長を示している。ここで、最大再送回数は 1、2、3、4 とした場合を示している。図 2.16 は同じ条件で、パケット誤り率 ε を変化させた場合の平均バースト発生間隔を示している。パケット誤り率が増加すると平均バースト長と平均バースト発生間隔の両方が増加する。しかし、ピークが存在する。パケット誤り率 ε が 0.3 より大きくなると、バーストが断片化するため両方の平均値は減少する。最大再送回数 N_r が大きな値をとるとピーク値が大きくなる事が分かる。これは、最大再送回数 N_r に比例して最大バースト長及び最大バースト間隔が大きくなることによる。図 2.17 はパケット誤り率 ε を変化させた場合のバースト長の変動係数を示している。ここで、最大再送回数 N_r を 1 から 4 に変化させ、またフレームを構成するスロット数 M が 64 と 256 の場合を示している。図 2.18 は同じ条件でのバースト発生間隔の変動係数を示している。どちらも同じような傾向が見られる。

M が 64 の場合、平均値はパケット誤り率 ε が 0.1 より大きい場合に最大となるが、変動係数はパケット誤り率 ε が 0.01 付近で最大となる。これは、パケット誤り率が小さい場合にバースト長やバースト発生間隔の変動が大きいことを意味する。このような変動係数のピークが存在する理由は次のように考えられる。パケット誤り率が大きくなるとともに、パケットの再送頻度が大きくなる。しかし、複数パケットの再送が複数チャンネルで並行して行われる可能性も大きくなり、バーストの断片化が生じバースト発生間隔も短くなる。このため、長さの大きいバーストは誤り率が小さい方が発生しやすくなるが、あまり小さいと 1 回の再送で終了する場合も多くなる。

M が 256 の場合、変動係数のピークはパケット誤り率 ε が 0.025 付近で発生し、その値は M が 64 の場合に比べ 2 倍程度大きい。ピークをとるパケット誤り率はフレームを構成するスロット数 M に逆比例する傾向のあることが分かる。 M は伝送リンクの帯域と遅延時間に依存して決まるので、SR ARQ によるバースト的なパケット出力の影響は帯域と遅延時間の積が大きくなるのに伴い、より小さなパケット誤り率で顕著になるといえる。

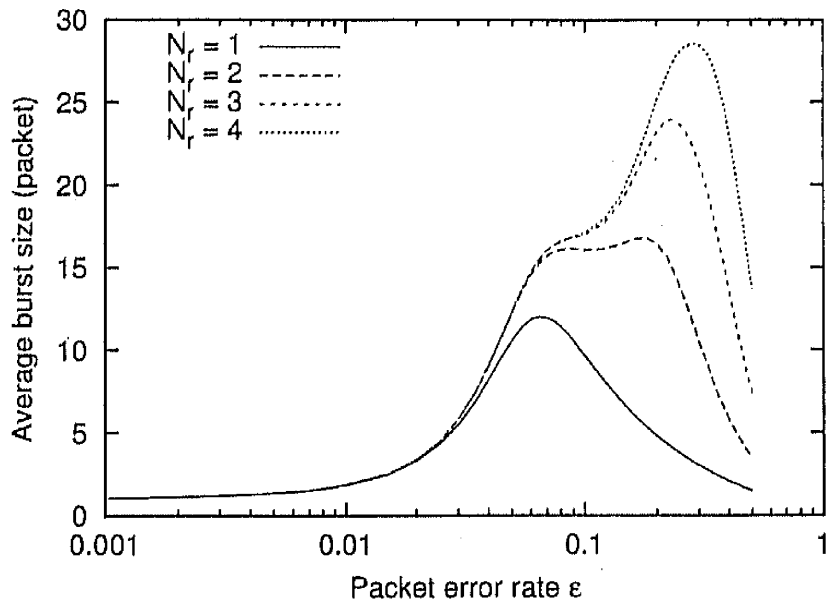


図 2.15 パケット誤り率 ϵ と平均バースト長の関係 ($M = 64$)

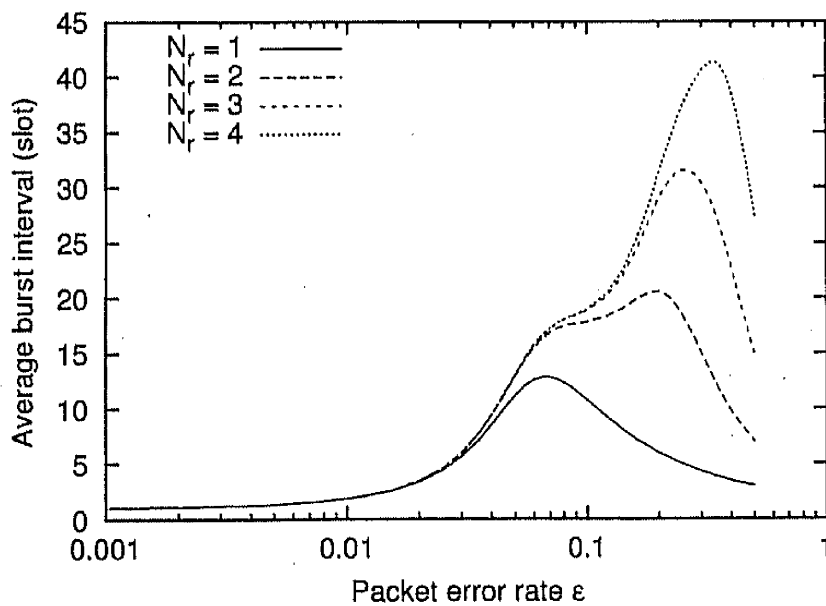


図 2.16 パケット誤り率 ϵ と平均バースト発生間隔の関係 ($M = 64$)

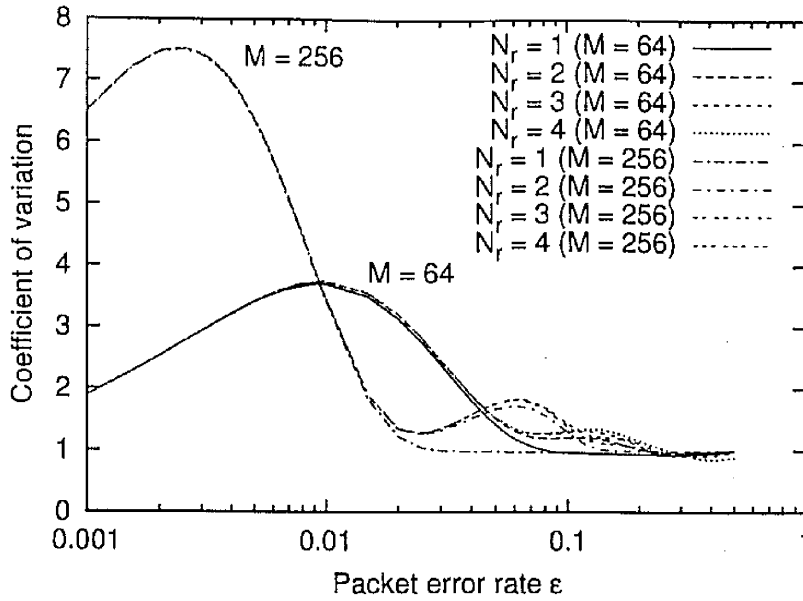


図 2.17 パケット誤り率 ϵ とバースト長の変動係数の関係 ($M = 64, 256$)

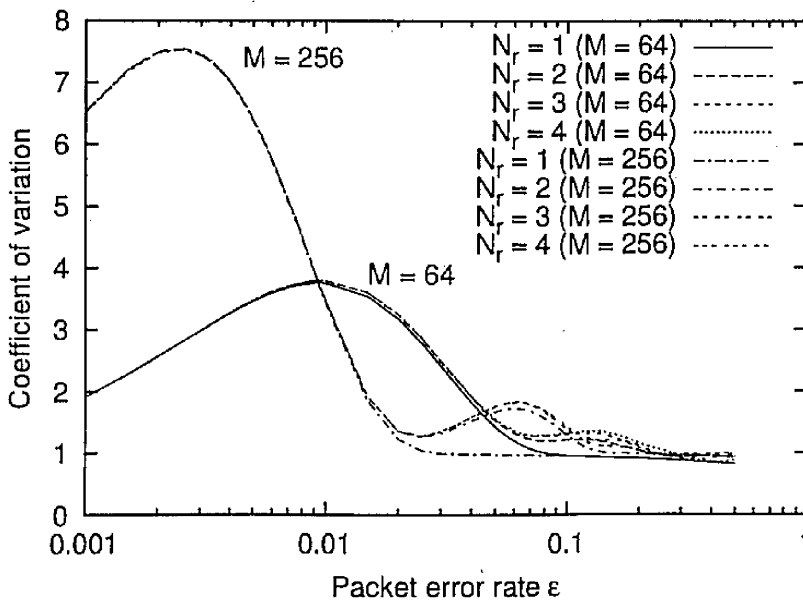


図 2.18 パケット誤り率 ϵ とバースト発生間隔の変動係数の関係 ($M = 64, 256$)

2.4.2 特定の値以上の長さをもつバーストの平均発生間隔

表 2.1 は、パケット誤り率 ϵ が 0.01 の場合について長さが 1、20、50、100、150、190 以上のバーストの平均発生間隔の数値計算例を示している。表 2.2 はパケット誤り率 ϵ が 0.1 の場合の同様の数値計算結果である。ここで、最大再送回数 N_r を 3 としている。これらの表では、同じ条件で行ったシミュレーションの結果も併せて示しており、シミュレーション時間は 10^8 スロット、シミュレーションの結果は 10 回の独立した試行により平均などを求めている。この条件で生じ得る最大バースト長は $N_r(M-1)+1=190$ パケットとなる。これらの表では平均値とともに 95% 信頼度区間も併せて示している。解析より求めた数値計算結果はシミュレーション結果と良く一致しており、解析が妥当であることを示している。パケット誤り率 ϵ が 0.1 の場合、シミュレーションでは長さが 190 となるバーストは一度も発生しなかったため、この長さについて結果は得られていない。これは、解析による数値計算結果で平均バースト発生間隔が極めて大きい値となることと符合する。

以上述べたように、バースト長とバースト発生間隔の確率分布により、特定の値以上の長のバーストについて平均発生間隔を求めることが可能であることを示した。これらの結果から SR ARQ のバースト的なパケット出力をバッファ数の制限されたルータで中継する場合、バッファオーバフローの頻度をある程度予測することが可能である。

これらの表において、長さが 50 パケット以上のバーストの平均発生間隔は二つのパケット誤り率でほぼ同じとなっている。より正確には、パケット誤り率 ϵ が 0.01 の場合の平均発生間隔は 165 スロットで、これは、パケット誤り率 ϵ が 0.1 の場合の平均発生間隔は 173 スロットより若干小さくなっている。この条件では、0.5 sec が 128 スロットに相当する。両方の誤り率で、長さが 50 パケット以上のバーストは平均で約 1.3 sec ごとに生じることになる。

バースト長が 100 または 150 以上の場合、パケット誤り率 ϵ が 0.1 の場合の平均発生間隔はパケット誤り率 ϵ が 0.01 の場合より小さくなっている。しかし、この条件での最大バースト長である 190 パケットの場合は、パケット誤り率 ϵ が 0.01 の方が平均発生間隔が小さくなっている。最大のバースト長はあるパケットが最大再送回数である N_r 回再送されている間に後続のパケットにおいて誤りが全く発生しない条件で発生する。パケット誤り率が大きくなるとこの条件を満足する事象が発生しにくくなる。これは、長いバーストが最も発生しやすいパケット誤り率が存在することを示している。

表 2.1 特定の値以上の長さをもつバーストの平均発生間隔 ($\varepsilon = 0.01$)

$\varepsilon = 0.01, M = 64, N_r = 3, m_b = 1.8862$			
n	$P_B(n)$	$m_B(n)$	sim. average \pm 95% conf.
1	1.0000e+00	1.8862e+00	1.8859e+00 \pm 1.05e-03
20	1.5363e-02	1.2277e+02	1.2278e+02 \pm 8.30e-02
50	1.1412e-02	1.6528e+02	1.6534e+02 \pm 1.20e-01
100	1.2899e-04	1.4623e+04	1.4555e+04 \pm 9.30e+01
150	1.4540e-06	1.2972e+06	1.2628e+06 \pm 9.64e+04
190	1.4686e-07	1.2843e+07	1.2832e+07 \pm 2.59e+06

表 2.2 特定の値以上の長さをもつバーストの平均発生間隔 ($\varepsilon = 0.1$)

$\varepsilon = 0.1, M = 64, N_r = 3, m_b = 18.9030$			
n	$P_B(n)$	$m_B(n)$	sim. average \pm 95% conf.
1	1.0000e+00	1.8903e+01	1.8898e+01 \pm 6.66e-03
20	2.4674e-01	7.6611e+01	7.6646e+01 \pm 2.67e-02
50	1.0929e-01	1.7296e+02	1.7299e+02 \pm 1.03e-01
100	1.1481e-02	1.6467e+03	1.6483e+03 \pm 4.00e+00
150	2.2480e-04	8.4084e+04	8.4027e+04 \pm 1.67e+03
190	4.5385e-14	4.1651e+14	not available

2.4.3 最大長のバーストが発生する確率を最大にするパケット誤り率

上記の考察から、大きなバースト長が発生する確率を最大とするパケット誤り率が存在することが予想される。以下では、一例として最大長のバーストが発生する確率を計算する。 $P_{max,b}(\varepsilon)$ を最大長のバーストが発生する確率とする。この確率は以下の二つの事象の積事象として求めることができる。

- パケットが N_r 回の再送により正しく受信される。
- 上記のパケットが再送されている間、後続のパケットがすべて誤りなく受信される。これは $(M - 1)N_r$ 個の後続パケットがすべて誤りなく受信されることを意味する。

これから、 $P_{max,b}(\varepsilon)$ は以下のように求められる。

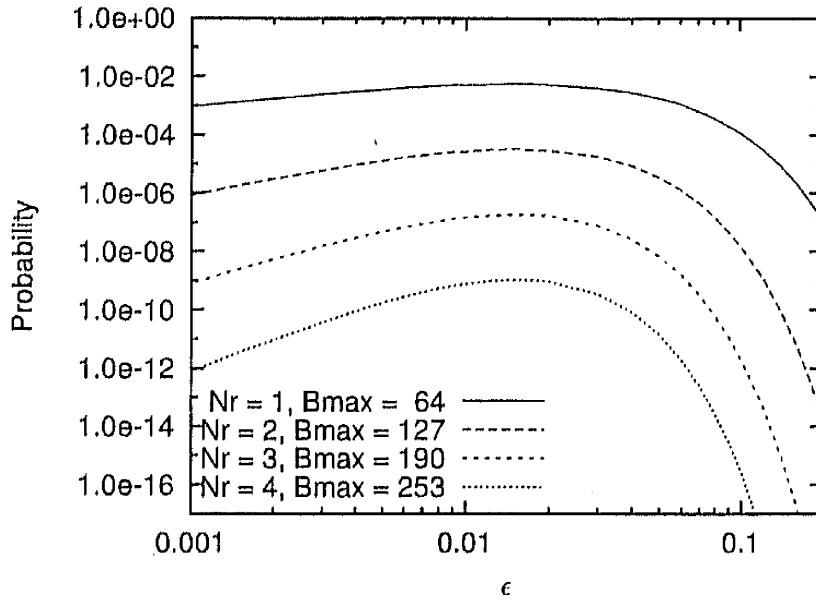


図 2.19 パケット誤り率 ε を変化させた場合の最大長のバーストの発生確率 ($M = 64$)

$$\begin{aligned} P_{max.b}(\varepsilon) &= (1 - \varepsilon)\varepsilon^{N_r}(1 - \varepsilon)^{(M-1)N_r} \\ &= \varepsilon^{N_r}(1 - \varepsilon)^{N_r(M-1)+1}. \end{aligned} \quad (2.26)$$

図 2.19 はパケット誤り率を変化させた場合の最大の長さをもつバーストの発生確率を示している。ここで、最大再送回数 N_r は 1 から 4 までの値としている。最大再送回数 N_r に応じてバーストの最大長は、64、127、190、253 の値をとる。すべての最大再送回数において、パケット誤り率 ε が 0.02 の場合にピーク値が存在することが分かる。

$P_{max.b}(\varepsilon)$ を最大とするパケット誤り率 ε_{max} は $\frac{dP_{max.b}(\varepsilon_{max})}{d\varepsilon} = 0$ から求めることが出来る。

$$\frac{dP_{max.b}(\varepsilon)}{d\varepsilon} = \varepsilon^{N_r-1}(1 - \varepsilon)^{N_r(M-1)+1} \cdot \{N_r - \varepsilon(N_r M + 1)\} \quad (2.27)$$

ε_{max} は次のように計算される。

$$\varepsilon_{max} = \frac{N_r}{N_r M + 1} \quad (2.28)$$

フレームを構成するスロット数 M は 1 より十分大きいので、 $M \gg 1/N_r$ である。

$$\varepsilon_{max} \approx \frac{1}{M} \quad (2.29)$$

数値計算例で、 $M = 64$ 及び $N_r = 3$ としているので $\epsilon_{max} = 0.01554$ となる。この結果から、パケット誤り率 ϵ が 0.01 の方が、パケット誤り率 ϵ が 0.1 の場合に比べ最大長のバーストが発生しやすいことが分かる。パケット誤り率が $\epsilon_{max} \approx 1/M$ の場合に、最大長のバーストが発生しやすい結果は $M = 64$ の場合に $\epsilon = 0.01$ で変動係数が最大となること、及び $M = 256$ の場合に $\epsilon = 0.01$ で変動係数が最大になることと符合している。フレームを構成するスロット数 M が大きくなると、 M に反比例して変動係数が最大となるパケット誤り率 ϵ が小さくなるといえる。

2.5 順序制御を行わない場合の順序逆転パケット数の分布

2.5.1 順序逆転パケット数の確率分布

これまで説明したように、SR ARQ の順序制御は出力においてバースト的なパケットトラヒックを発生させる。もし SR ARQ で順序制御を行わなければ、このようなバースト的なパケット出力は回避できる。しかし、その代わりに順序が逆転したパケットが上位層に転送される。このような順序逆転パケットは、上位層で TCP が用いられる場合には重複 ACK を引き起こす。重複 ACK により不必要なパケットの再送が行われ、輻輳ウィンドウサイズの減少が行われるので、TCP の性能が低下する結果を招く。

順序逆転パケット数の定義にはいくつか考えられる。ここでは、SR ARQ の受信側が次に受信を期待している通番の値より大きい値の通番をもつパケットの数と定義している。ある時点で SR ARQ が次に受信を期待している通番のパケット (in-sequence なパケット) を受信してから次に in-sequence なパケットを受信するまでの間に通番が期待値よりも大きなパケットの数が順序逆転パケット数となる。従って、in-sequence なパケットを受信するたびに順序逆転パケット数が定まり観測される。

2.3 節で説明したブロッキングパケットが存在する場合、上位層も次にこのパケットの受信を期待している。順序制御を行わない場合、前のブロッキングパケットを受信してから次のブロッキングパケットを受信するまでの間に正しく受信したパケット数が、順序逆転パケット数となる。順序逆転パケット数は連続する二つのブロッキングパケットの間のスロットの中で、ブロッキングパケットが再送に使用するスロットを除いた残りのスロットで正しく受信したパケットの数となる。

図 2.20 は順序逆転パケットを受信する可能性のあるスロットを示している。二つの連続するブロッキングパケットの間隔は、2.3 節で述べたバースト間隔と密接な関係がある。図 2.20 から、もしバースト間隔が $wM + i$ スロットであるとすると、順序逆転パケットはブロッキングパケットが再送に使用するスロットを除いた $w(M - 1) + i - 1$ 個のスロットで発生する可能性がある。 $P_{misorder}(n)$ を順序逆転パケット数が n となる確率とする

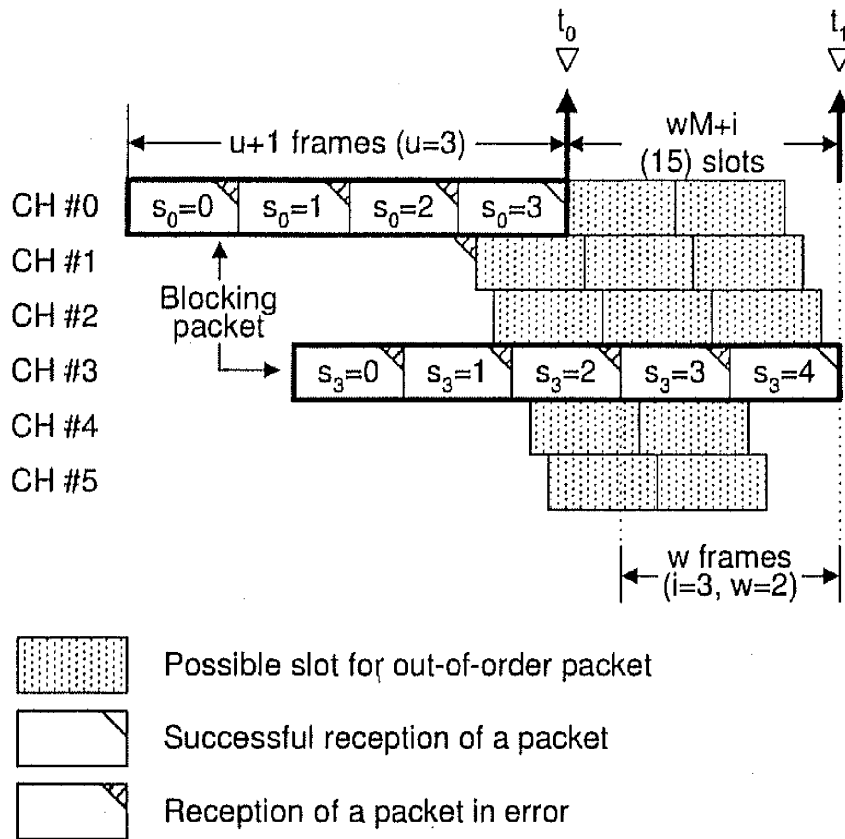


図 2.20 順序逆転パケットの発生可能なスロット

と、 $w(M-1)+i-1$ 個のスロットで正しく受信したパケットが、順序逆転パケットとなるので、 $P_{misorder}(n)$ は以下のように二項分布により求めることができる。

$$P_{misorder}(n) = \sum_{w=0}^{N_r} \sum_{i=1}^{M-1} \binom{w(M-1)+i-1}{n} \cdot (1-\varepsilon)^n \varepsilon^{w(M-1)+i-1-n} P_{init}(w, i), \quad (2.30)$$

ここで、 $0 \leq n \leq N_r(M-1) + M - 2$ である。

2.5.2 完全に順序制御を行わない場合

ここで、TCP NewReno のような TCP が SR ARQ の上位層に用いられる場合を考える。また、単一の TCP コネクションが SR ARQ の上位に存在するものと仮定する。複

数の TCP コネクションが存在する場合は、今後の課題である。順序が逆転したパケットを受信するたびに TCP の受信側は重複 ACK を TCP の送信側に返送する。もし複数の順序逆転パケットを受信した場合は、同じ数の重複 ACK が TCP の送信側に返送される。TCP の送信側が特定の数 (通常 3) の重複 ACK を受信すると、重複 ACK の通番のパケットが不必要に再送され (高速再送)、また輻輳ウィンドウサイズがそれ以前の半分の大きさに減少する (高速リカバリ) [55]。

$P_{retrans}$ を不必要な高速再送が起動される確率とすると、 $P_{retrans}$ は順序逆転パケット数が特定の値 k ($k = 3$) を超える確率として計算される。

$$P_{retrans} = 1 - \sum_{n=0}^{k-1} P_{misorder}(n) \quad (2.31)$$

表 2.3 は順序逆転パケット数の平均値、標準偏差、そして $P_{retrans}$ に関する数値計算結果を示している。パケット誤り率が増加するとともに順序逆転パケット数の平均値が大きくなるのが分かる。パケット誤り率が大きい場合、 $P_{retrans}$ はパケット誤り率とほぼ同じか大きい値となっている。

これは in-sequence でパケットを受信した場合に、高速再送がパケット誤り率と同じかそれ以上の確率で発生し、不必要な再送と輻輳ウィンドウサイズの減少が発生することを示している。この結果から、SR ARQ 上で TCP を用いる場合、順序制御は必須であるといえる。

2.5.3 部分的に順序制御を行う場合

順序逆転パケットの問題を緩和する一つの方法として、順序制御により保留されるパケット数の最大値を特定の値に制限することが考えられる。他の方法としてパケットを保留する時間を制限することも考えられるが、ここでは上記の結果を応用して簡単に評価できるパケット数の制限を考える。この場合、一つのバーストに含まれるパケット数の最大値も制限される (特定の値 + 1)。ある程度、順序制御を行うことにより順序逆転パケットの発生確率が無視できるほど小さくなることを期待する。

最大 h 個のパケットを順序制御で保留する場合にパケットの順序逆転が発生する確率を $P_o(h)$ とすると、これは、 $P_{misorder}(n)$ より以下のように計算できる。

$$P_o(h) = 1 - \sum_{n=0}^h P_{misorder}(n) \quad (2.32)$$

表 2.3 順序逆転パケット数の平均と標準偏差及び $P_{retrans}$

$M = 64$		packet loss rate	no. of out-of-order packets		
N_r	ϵ		mean	std	$P_{retrans}$
4	0.0001	1.0e-020	0.006	0.631	0.000101
	0.0010	1.0e-015	0.065	2.013	0.001062
	0.0100	1.0e-010	0.867	6.993	0.018011
	0.1000	1.0e-005	16.111	22.513	0.753763
3	0.0001	1.0e-016	0.006	0.631	0.000101
	0.0010	1.0e-012	0.065	2.013	0.001062
	0.0100	1.0e-008	0.867	6.993	0.018011
	0.1000	1.0e-004	16.014	22.024	0.753756
2	0.0001	1.0e-012	0.006	0.631	0.000101
	0.0010	1.0e-009	0.065	2.013	0.001062
	0.0100	1.0e-006	0.867	6.991	0.018011
	0.1000	1.0e-003	15.091	19.337	0.753098
1	0.0001	1.0e-008	0.006	0.631	0.000101
	0.0010	1.0e-006	0.065	2.010	0.001062
	0.0100	1.0e-004	0.856	6.863	0.018005
	0.1000	1.0e-002	8.783	9.342	0.713162

図 2.21 はパケット誤り率 ϵ が 0.1 の場合の最大保留数 h と確率 $P_o(h)$ の関係を示している。図 2.22 はパケット誤り率 ϵ が 0.01 の場合の最大保留数 h と同じ確率の関係を示している。

これらの図において同じ条件で実行されたシミュレーション結果も合わせて示している。解析より求めた数値計算結果がシミュレーションと良く一致することが分かる。これらの図から、順序逆転パケット数が k より大きくなる確率を抑えるためには、かなりの数のパケットを保留しなければならないことが分かる。例えば、パケット誤り率が $\epsilon = 0.1$ の場合、順序逆転パケットの発生確率を 10^{-5} 以下とするためには、172 パケットを順序制御のために保留する必要がある。パケット誤り率が $\epsilon = 0.01$ の場合は、126 パケットとなる。高速再送は $k(k = 3)$ 回の重複 ACK 受信により引き起こされるので、 $k - 1$ 個のパケットは上記の計算から差し引いて考えることができるので、パケット誤り率が $\epsilon = 0.1$ の場合は、170 パケット、パケット誤り率が $\epsilon = 0.01$ の場合は、124 パケットとなる。これらの値は、この条件で完全な順序制御を行うのに必要な最大保留数 $(M - 1)N_r = 189$

に近い値である。

$P_{ooo}(l, h)$ を最大 h 個のパケットを保留した場合に、順序逆転パケット数が 1 個となる確率とする。 $P_{ooo}(l, h)$ は以下のように計算される。

$$P_{ooo}(l, h) = \begin{cases} \sum_{n=0}^h P_{misorder}(n) & l = 0 \\ P_{misorder}(l + h) & l \geq 0 \end{cases} \quad (2.33)$$

最大 h 個のパケットを順序制御で保留する場合に、 $P_{retrans2}(h)$ を順序逆転パケット数が特定の値 k ($k = 3$) より大きくなる確率とする。

$$P_{retrans2}(h) = 1 - \sum_{l=0}^{k-1} P_{ooo}(l, h) \quad (2.34)$$

バーストに含まれるパケット数は $h + 1$ 以下に制限されるので、バースト長の確率分布 $P_{bp2}(b, h)$ は以下のように計算される。

$$P_{bp2}(b, h) = \begin{cases} P_b(bn) & b \leq h + 1 \\ 1 - \sum_{d=0}^h P_{bp}(d) & b = h + 1 \end{cases} \quad (2.35)$$

図 2.23 は、順序制御のために保留される最大パケット数を変化させた場合の TCP において高速再送が起動される確率 $P_{retrans2}(h)$ を示している。ここで、パケット誤り率 ε は 0.1、0.01、0.001 としている。この図も TCP による不必要な高速再送を抑えるためには、パケット誤り率が大きい場合は大量のパケットを保留する必要があることを示している。図 2.24 は、順序制御のために保留できる最大バッファ数を変化させた場合のバースト長の平均及び標準偏差を示している。ここで、パケット誤り率 ε は 0.1 及び 0.01 としている。この図から、保留されるパケット数が 60 を超えると平均バースト長及びその標準偏差はほぼ一定となり変わらないことが分かる。もし順序制御により保留されるパケット数が 60 より少ない場合、バースト的なパケット出力は緩和されるが、それより大きい場合はほとんど効果がないことが分かる。前に述べたように、順序逆転パケットによる不必要な高速再送を 10^{-5} 以下とするためにはパケット誤り率 ε が 0.01 の場合、124 パケット以上を保留する必要があったが、これではバースト性は改善されないことが分かる。

これらの結果は、順序逆転パケットをその影響が無視できるレベルとなるよう保留すると、順序制御によるバースト的な性質が顕著になるというトレードオフを示している。

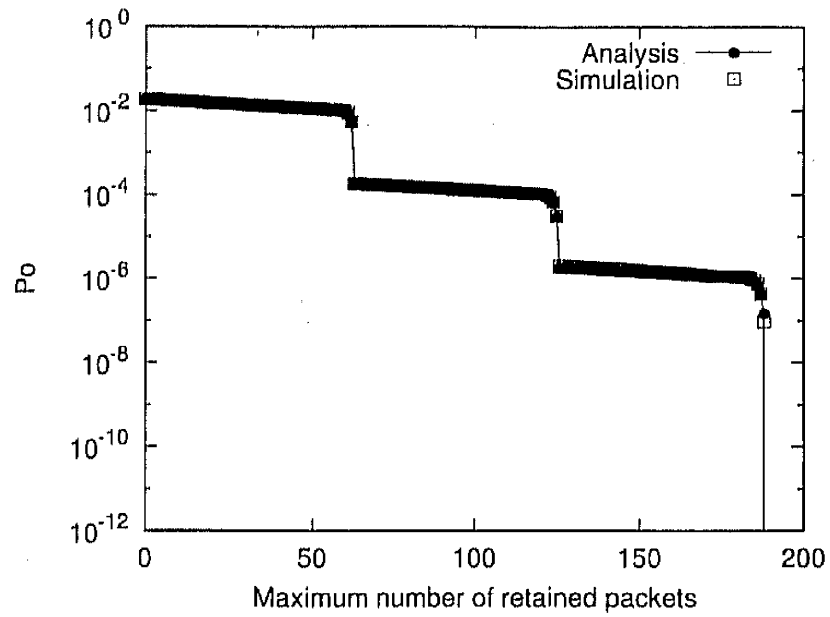


図 2.21 順序逆転パケット数が最大保留パケット数より大きくなる確率 ($\epsilon = 0.01, M = 64, N_r = 3$)

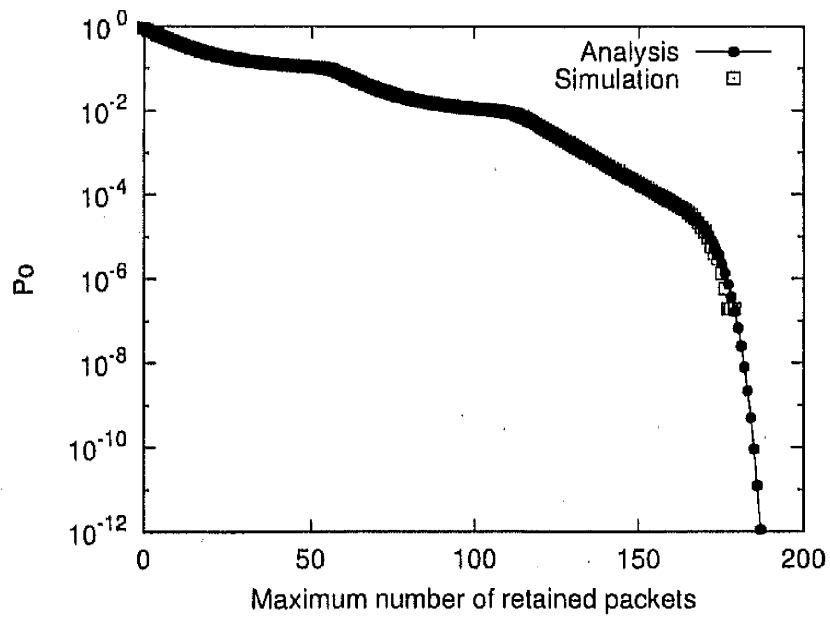


図 2.22 順序逆転パケット数が水平軸の値より大きくなる確率 ($\epsilon = 0.1, M = 64, N_r = 3$)

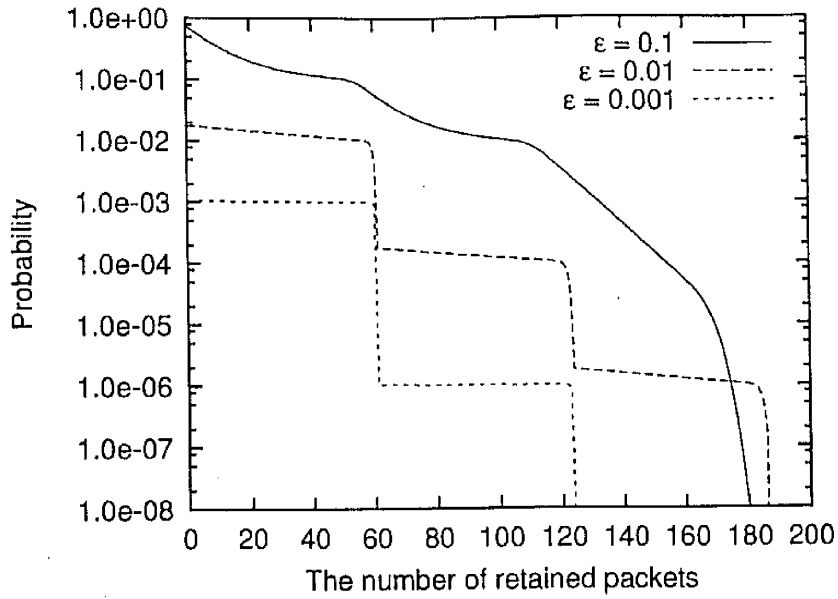


図 2.23 高速再送が起動される確率 ($\epsilon = 0.1, 0.01, 0.001, M = 64, N_r = 3$)

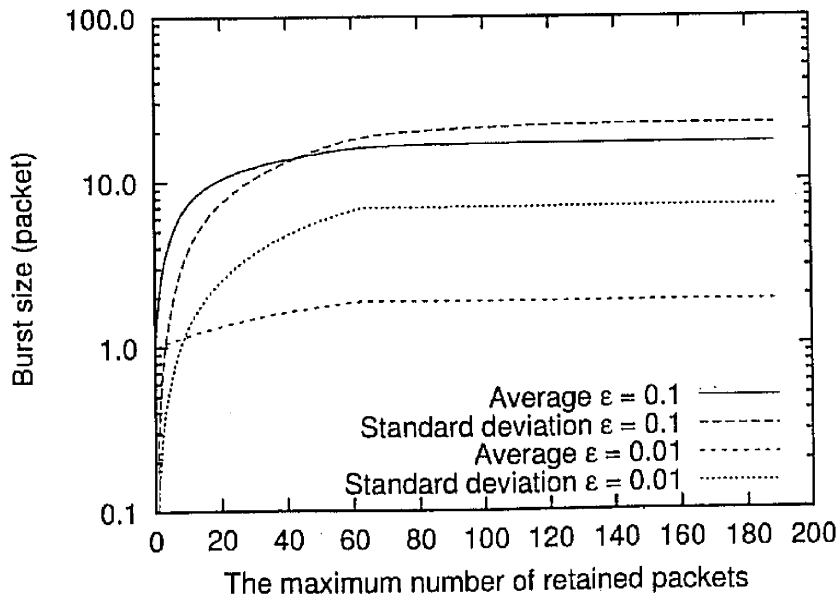


図 2.24 最大バッファ数を変化させた場合のバースト長の平均と標準偏差 ($\epsilon = 0.1, 0.01, M = 64, N_r = 3$)

2.6 むすび

この章では SR ARQ によるバースト的なパケット出力について検討し、バーストに含まれるパケット数の分布とバーストの発生間隔分布を求めた。また、ある値以上の大きさのバーストが発生する平均間隔を求めた。これらの解析結果をシミュレーションと比較し、解析の妥当性を確認した。これらから、最大再送回数が制限された場合でも、バースト長の分布とバースト間隔分布の評価が可能であることを示した。この章の解析による興味深い結果の一つは、最大長のバーストが発生する確率はパケット誤り率が $\varepsilon \approx 1/M$ の時に最大となることである。また別の興味深い結果は、バースト長とバースト間隔の変動係数が同様の値でピークをとることである。これらの結果は、パケット誤り率の値がラウンドトリップ時間に送信されるパケット数の逆数の場合に長さの大きなバーストが発生しやすいことを示している。バースト的なパケット出力はパケット誤り率が大きい場合にその影響が顕著となる訳ではなく、帯域と遅延時間の積が大きい場合は、小さなパケット誤り率でもその影響に注意する必要がある。

もしパケットの順序逆転が許容されるならば、SR ARQ によるバースト的なパケット出力は、SR ARQ において順序制御を行わなければ発生しない。この章では、SR ARQ が順序制御を部分的に行う場合や全く行わない場合について順序逆転パケット数の分布についても解析した。SR ARQ の上で TCP が用いられる場合、TCP への順序逆転パケットによる影響を無視できるレベルとするためには完全な順序制御に近いかなりの数のパケット保留が必要であることが分かった。

この章では、リンクはパケットの送信にフルに利用されることを仮定したが、実際のネットワークにおいてこの条件は必ずしも満足されているとは限らない。この章の解析は最悪条件で行っており安全側の評価といえる。この章では衛星回線の場合について数値計算結果とシミュレーションを示した。今後、SR ARQ は帯域と遅延時間の積が大きくなる高速の地上無線通信にも適用されるものと考えられる。この章の結果はこのような地上無線通信の場合にも適用できる。既に述べたように第三代移動通信の W-CDMA ではレイヤ 2 のプロトコルとして RLC (Radio Link Control) を用いているが、このプロトコルは SR ARQ に基づいている [52]。より高速の HSDPA はハイブリッド ARQ を採用しているが、このプロトコルは解析で用いた複数の Stop-and-Wait プロトコルが並列動作するモデルと類似の動作となっている [56] [57]。この章で述べた解析はこのようなプロトコルにも適用できる。

第3章

SR ARQ の順序制御による遅延時間の解析

3.1 まえがき

この章では SR ARQ の順序制御による遅延時間の確率分布を解析し、HOL ブロッキングによる影響を明らかにする。この章で述べる解析に関連した研究が発表されており、以下に関係を述べる。Rosberg らの解析は回線をフルに利用する条件で順序制御による遅延時間について平均値を求めている [28]。Shacham らは送信側と受信側で複数のチャンネルが存在し、パケットを複数のチャンネルに分配して転送する場合の順序制御による遅延時間を解析している [31]。これらの研究では再送回数に上限を設けておらず複数のチャンネルがフルに利用される場合を想定している。また Shacham らの研究では各チャンネルの時間をスロット化しているが、すべてのチャンネルでスロットの境界タイミングが同期している。これらの条件はこの章で述べる解析の対象とは異なっている。他の研究として、UMTS の RLC プロトコルにおける順序制御による遅延時間のヒューリスティックな解析に関するものがある [58]。ここでは RLC におけるセグメンティングとリアセンブリングを考慮しており、上位層の SDU はリンク層の PDU に分割され、受信側は受信した PDU を集めて SDU を再構築する。この研究のねらいとこの章における解析との間には類似点が認められるが、以下の点が異なっている。

- Rossi らの研究では SR ARQ の膨大な状態空間を扱うことを避けるためヒューリスティックな近似を導入している。この章で述べる解析は SR ARQ の可能なすべての状態を考慮しており、厳密な解析となっている。ヒューリスティックな近似による解析の結果はシミュレーション結果とよい一致を示しているが、近似の限界が明らかではない問題がある。

- Rossi らの研究においても伝送リンクがフルに利用されている条件（重負荷条件）で行われている。一方、この章の解析では伝送リンクの利用率を指定するパラメータを導入している。このため、この章で述べる解析ではパラメータを変化させることにより重負荷を含む任意の負荷条件における解析が可能である。

この章は以下のように構成されている。3.2 節で、HOL ブロッキング、解析モデル及び解析の条件、状態の定義と状態確率について説明する。3.3 節で、再送と順序制御による遅延時間の確率分布を求める。3.4 節は解析に基づく数値計算結果を示し、シミュレーション結果と比較する。最後に、3.5 節でまとめを述べる。

3.2 順序制御による遅延時間とその解析モデル

3.2.1 従来の SR ARQ と HOL ブロッキング

従来用いられている SR ARQ による再送シーケンスの例を図 3.1 に示す。SR ARQ は伝送リンクで送信されるパケットすべての順序を保存する。このような従来の SR ARQ では、SR ARQ 上に複数のフローを多重化する場合、不必要なパケットの保留とそれに伴う遅延時間が発生する問題がある。この図で、a と b の 2 本のフローが SR ARQ 上に多重化されている。パケット a1 と a2 が伝送リンクで伝送誤りにより失われ再送されている。従来の SR ARQ ではフロー b に属するパケット b1 及び b2 が、失われたパケット a1 が再送され正しく受信されるまで不必要に保留されている。このような状況はフロー a のパケットによりフロー b のパケットに対し HOL (Head Of Line) ブロッキングが発生しているといえる。

このような SR ARQ に伴う HOL ブロッキングに関して明確に問題点を指摘した研究は、これまで行われていない。リンク層における ARQ に関して、RFC3366 の中では、個々のフロー内ではパケットの順序を保存するが、異なるフローに属するパケットについてパケットの順序を守らない方がよいとの記述が見られる [70]。しかし、具体的な評価は報告されていない。

TCP における誤り回復では、受信側で順序逆転したパケットを保留し順序制御を行う。TCP で複数のストリームを転送する場合、この順序制御により HOL ブロッキングが発生することが述べられている [91]。このような複数ストリームを扱うプロトコルとして最近では SCTP (Stream Control Transmission Protocol) が提案され、IETF で標準化されている [88][89]。SCTP を衛星通信に適用した場合の性能評価も報告されている [90][92]。

また無線 LAN では送信パケットが伝送誤りより失われると、再送が行われるが、後続のパケットが遅れないため HOL ブロッキングの生じることが問題となっており、解決方

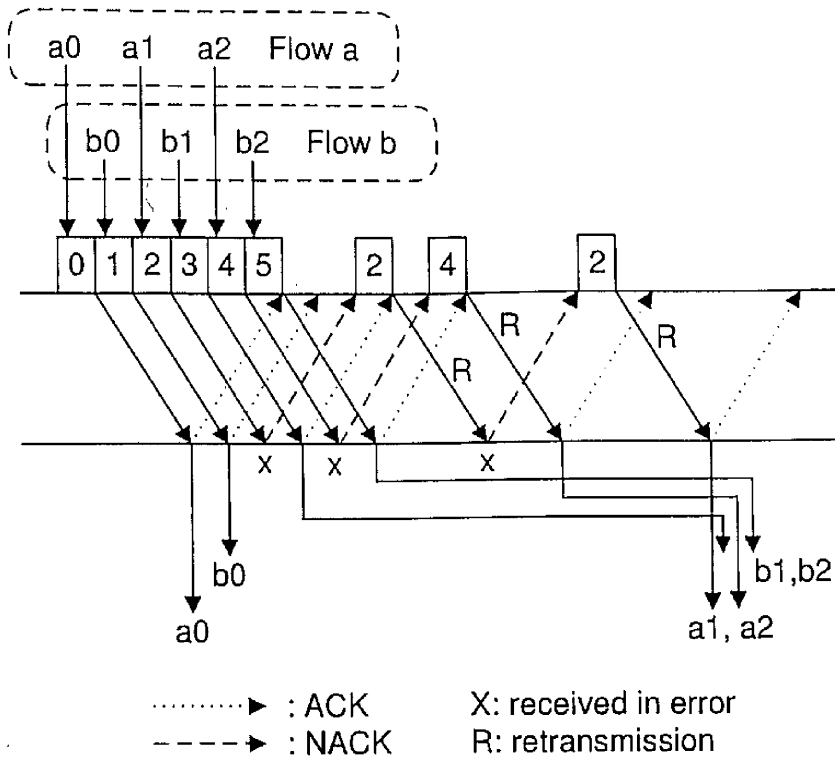


図 3.1 HOL ブロッキングが発生する場合の例

式が研究されている [99][100]。

図 3.2 に示すように、順序制御が理想的に行われればパケット b1 及び b2 は保留されることなく上位層に転送され、HOL ブロッキングは回避される。この場合、順序制御による遅延時間は少なくなると予想される。

SR ARQ の HOL ブロッキングによる影響は SR ARQ に多重化されるフロー数が多くなるに従って顕著になると予想される。この章では、SR ARQ 上に多重化されるフロー数と順序制御による遅延時間の関係を解析により評価し、HOL ブロッキングの影響を明らかにする。従来の研究による解析では SR ARQ 上に複数のフローが存在する場合は研究されておらず、以下で説明する解析は従来にない新しいアプローチである。

3.2.2 解析モデルと関連する仮定

順序制御による遅延時間を評価するため、以下を仮定して解析を行った。

1. 伝送リンクにおいてパケットの誤りはランダムに発生するものとし、パケット誤り率を ϵ とする。

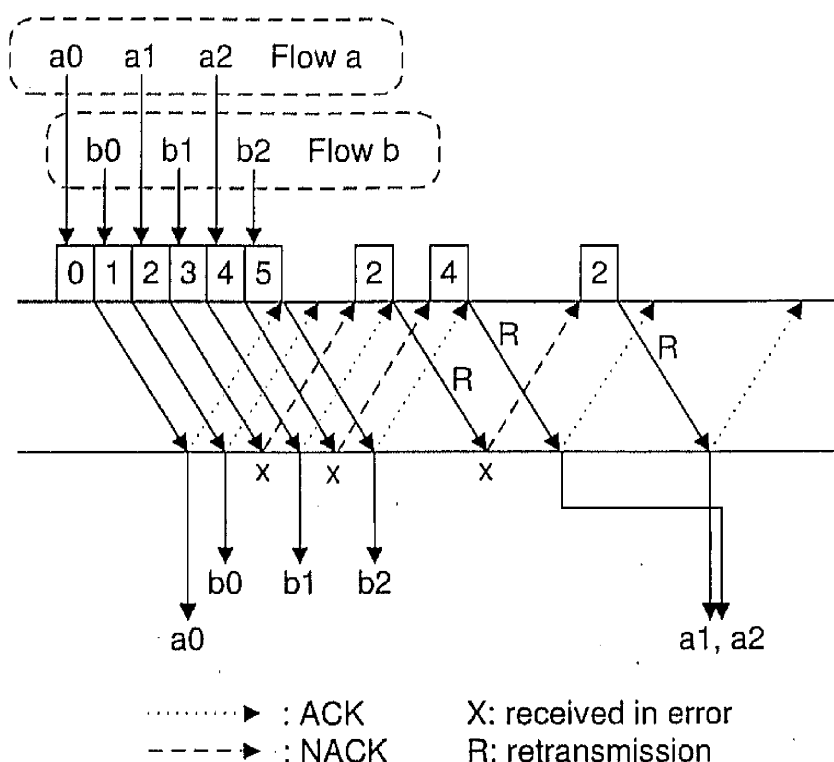


図 3.2 順序制御が理想的に行われ HOL ブロッキングが回避される場合の例

2. パケットが誤りなく受信された場合、送信側に肯定応答 (ACK) が直ちに返送され、誤りが発生した場合は否定応答 (NACK) が直ちに返送される。
3. ACK 及び NACK とも伝送誤りが発生しないものとする。
4. NACK が送信側に到着すると直ちにパケットが再送される。再送は新規パケットより高い優先順位で行われる。
5. 再送回数の最大値を N_r に制限する。 N_r 回再送されたパケットが正しく受信されなかった場合、このパケットの受信を待って受信側で保留されていたパケットは解放され、上位層に転送されるものとする。
6. パケット長は固定とし、時間をパケットが送信できる長さにスロット化する。
7. 各スロットにおいてチャンネルが利用できる場合 (再送が行われていない場合)、新規パケットの送信が確率 α で行われる。
8. 伝送リンク上に m 本のフローが存在する。フロー j に属するパケットが確率 β_j ($j = 1, \dots, m$) でランダムに発生する。ここで、 $\sum_{j=1}^m \beta_j = 1$ である。
9. あるパケットの送信開始時刻から、このパケットが正しく受信され、ACK が送信側に返るまでの連続するスロットの集合をフレームと呼ぶ。

10. フレームを構成するスロット数を M とする。フレーム内の各スロットには先頭より 0 から $M - 1$ までの番号が割当てられる。
11. M はラウンドトリップ時間に相当するスロット数に 1 パケットを送信するための 1 スロットを加えた値となる。
12. 連続するフレームにおいて同じ位置のスロットの集合がチャンネルを構成する。フレーム内には 0 から $M - 1$ までのチャンネルが存在する。

2章で述べたように、上記のような条件では、パケットが一旦あるチャンネルで送信されると、そのパケットが正しく受信されるかまたは最大再送回数に達して再送が放棄されるまで、パケットの再送は常に同じチャンネルのスロットで行われる。これから SR ARQ を独立な M 個のチャンネルの集合と考えることができる。ここで、各チャンネルにおけるパケットの送信は単純な Stop-and-Wait プロトコルでモデル化できる。

この章のモデルは 2章で述べたモデルを拡張しており、伝送リンクの負荷を変化させるためのパラメータとして確率 α を導入している。順序制御による遅延時間は伝送リンクの負荷により変わると予想されるので、 α を変化させることによりこれを評価することができる。また、確率 β_j ($j = 1, \dots, m$) は伝送リンク上に複数のフローが存在する場合を評価するために導入している。図 3.3 はこれらのパラメータにより複数フローのパケットが生成される例を示している。スロットが利用可能な場合（再送が行われていない場合）、新規パケットの送信が確率 α で行われる。送信された新規パケットはランダムに属するフローが決まるものとし、例えばフロー j に属する確率は β_j となる。このようなチャンネルの集合を解析することにより順序制御による遅延時間の確率分布を求めることが可能となる。以下で特定チャンネルのパケットに注目して解析を行うが、フレームの位相は任意であるため一般性を失うものではない。

3.2.3 チャンネルの状態確率

2章と同様にチャンネルの状態をそのチャンネルで送信されたパケットの連続受信失敗回数として定義する。例えば、図 3.4 はチャンネル #2 の状態の変化を示している。伝送誤りによりパケットの受信に失敗した場合、チャンネルの状態は 1 増加し、成功した場合は 0 に戻る。最大回数の再送の後にパケットの受信に失敗した場合も、状態は 0 に戻る。新しいパケットが伝送誤りなしに連続して受信された場合、連続するスロットで状態の値は 0 をとる。パケットの状態を連続受信失敗回数とした場合、状態遷移図は図 3.5 のように表される。

状態確率 $p_s(k)$ とパケット損失率 p_L は、それぞれ以下のようになる。

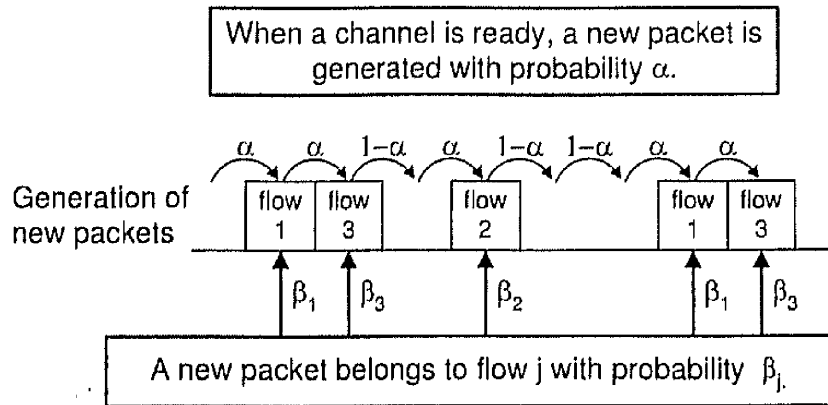


図 3.3 パラメータ α と β による複数フローの packets 生成

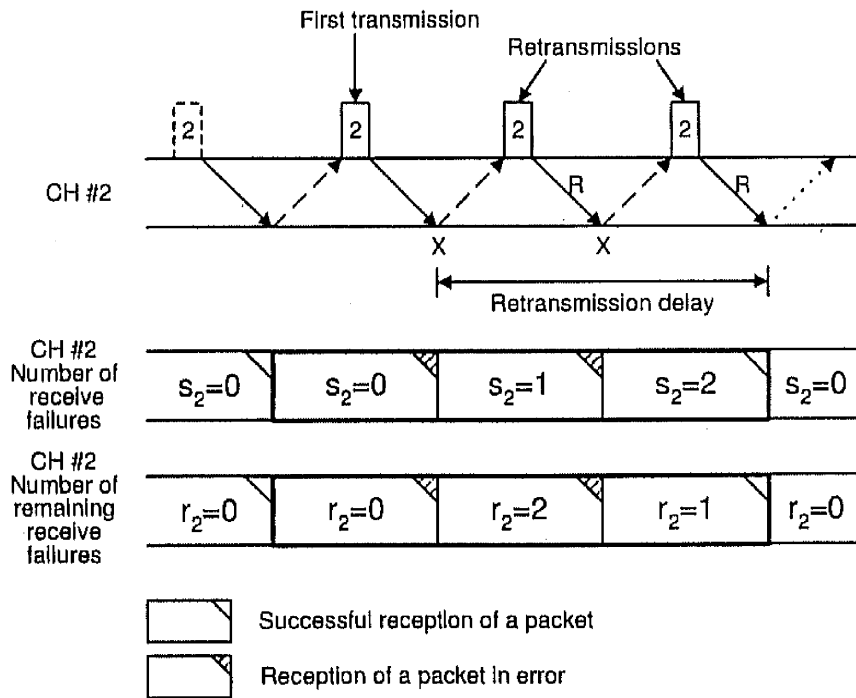


図 3.4 チャンネル #2 における状態遷移の例

$$p_s(0) = \frac{1 - \epsilon}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}} \tag{3.1}$$

$$p_s(k) = \frac{\alpha(1 - \epsilon)\epsilon^k}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}}, 1 \leq k \leq N_r \tag{3.2}$$

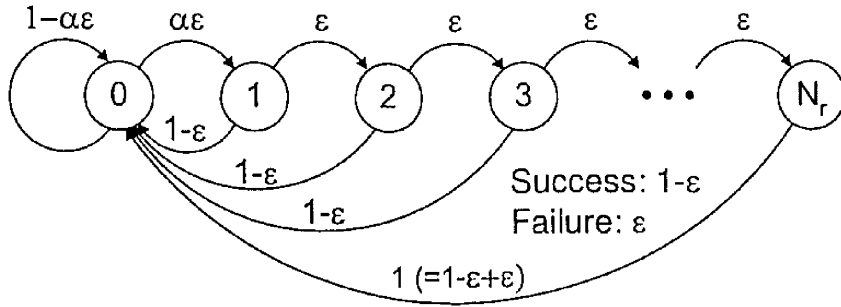


図 3.5 連続受信失敗回数に基づく状態遷移図

$$p_L = \varepsilon^{N_r+1} \quad (3.3)$$

以降に述べる順序制御による遅延時間の解析では、現在のパケットの受信が終了するまでに更に発生する受信失敗回数が必要となる。これを残余受信失敗回数と呼ぶ。図 3.4 は残余受信失敗回数を状態とした場合の状態の変化例も示している。この場合、パケットの第 1 回目の受信時点で受信失敗の場合、状態はこのパケットの受信が終了するまでのトータルな受信失敗回数に値が変化する。その後、状態値はこのパケットの受信に失敗するたびに値が 1 だけ減少する。ここで、パケットの受信終了とはパケットが正しく受信されて終了したか、または最大回数の再送後に受信失敗して終了した場合を意味する。残余受信失敗回数を状態とした場合、その状態確率 $p_r(r)$ は最初に定義した状態確率 $p_s(k)$ から求めることができる。

$r = 0$ の場合、

$$\begin{aligned} p_r(0) &= \{1 - \alpha + \alpha(1 - \varepsilon)\} p_s(0) \\ &\quad + \sum_{k=1}^{N_r-1} (1 - \varepsilon) p_s(k) + p_s(N_r) \\ &= \frac{1 - \varepsilon}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \end{aligned} \quad (3.4)$$

$1 \leq r \leq N_r$ の場合、

$$\begin{aligned} p_r(r) &= \alpha\varepsilon^r (1 - \varepsilon) p_s(0) + \sum_{k=1}^{N_r-r-1} (1 - \varepsilon) \varepsilon^r p_s(k) \\ &\quad + \varepsilon^r p_s(N_r - r) \\ &= \frac{\alpha(1 - \varepsilon) \varepsilon^k}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \end{aligned} \quad (3.5)$$

上で示したように残余受信失敗回数に基づく状態確率は、式(3.1)及び(3.2)による最初の定義による状態確率と同じ形となる。この結果は直感的に次のように説明できる。説明の便宜のために、前のパケットの受信終了時点から次のパケットの受信終了時点までをサイクルと呼ぶことにする。各サイクルにおいて、図3.4に示すように、最初の定義による状態値(連続受信失敗回数)と後で定義した状態値(残余受信失敗回数)との間には1対1の対応関係が存在する。従って、各状態値がとる確率は同じとなる。

先に述べたように、以下の解析では伝送リンクの負荷をパラメータ α により変化させる。チャンネルの状態値が0の場合、伝送リンクのスロットは確率 $(1-\alpha)$ で空となる。これから、伝送リンクの利用率 ρ は以下ようになる。

$$\begin{aligned}\rho &= 1 - (1 - \alpha)p_s(0) \\ &= \frac{\alpha(1 - \epsilon^{N_r+1})}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}}\end{aligned}\quad (3.6)$$

また、1スロットにおけるパケットの送信成功確率として求めたスループット S は以下となる。

$$\begin{aligned}S &= \alpha(1 - \epsilon)p_s(0) + \sum_{k=1}^{N_r} (1 - \epsilon)p_s(k) \\ &= \frac{\alpha(1 - \epsilon)(1 - \epsilon^{N_r+1})}{1 - \epsilon + \alpha\epsilon - \alpha\epsilon^{N_r+1}}\end{aligned}\quad (3.7)$$

3.3 再送及び順序制御による遅延時間の確率分布

この節では、2種類の遅延時間について確率分布を求める。一つは順序制御のみによる遅延時間の確率分布、他は再送と順序制御の両方による遅延時間の確率分布である。

3.3.1 順序制御のみによる遅延時間の確率分布

3.3.1.1 順序制御が行われる場合 ($i \neq 0$)

順序制御のみによる遅延時間はパケットが正しく受信されてから、それが上位層に転送されるまでの時間である。チャンネル#0においてフロー j のパケットが u 回の受信失敗の後に正しく受信されたとする($u = 0, 1, 2, \dots, N_r$)。

図3.6はチャンネル#0のパケットがフロー j に属するチャンネル#2のパケットの受信を待って遅延を被る場合を示している。この図で、チャンネル#0のパケットは $u(u=2)$ 回の受信失敗の後に正しく受信され、最初の受信失敗は時刻 t_0 で起こっている。この時点で他のチャンネルの状態を観測すると、少なくとも一つのチャンネルの状態(残余受信失敗

回数) が $u + 1$ 以上の場合に順序制御による遅延時間が発生することが分かる。この図で、時刻 t_0 におけるチャンネル #2 と #4 の状態はそれぞれ 4 と 3 である。チャンネル # i の状態 (残余受信失敗回数) がチャンネル #1 から # $(M - 1)$ までの中で最大で、その値が $u + w + 1$ ($w = 0, 1, 2, \dots, N_r - 1$) である場合、チャンネル #0 のパケットが被る順序制御による遅延時間は $wM + i$ スロットとなる。図 3.6 で、チャンネル #2 の状態が最も大きく、これから $i = 2$ 、 $u + w + 1 = 4$ 、 $w = 1$ となる。この図で $M = 6$ としているので、順序制御による遅延時間は $wM + i = 8$ スロットとなる。この事象の確率を $P_{reseq}(w, i|\beta_j)$ とすると、以下のように計算される。

$$P_{reseq}(w, i|\beta_j) = \sum_{u=0}^{N_r-w-1} p_t(u)R(w+u+1, i|\beta_j) \quad (3.8)$$

ここで、 $p_t(u)$ はチャンネル #0 のパケットが u 回の受信失敗の後に正しく受信される確率である。

$$p_t(u) = \frac{(1-\varepsilon)\varepsilon^u}{1-\varepsilon^{N_r+1}} \quad (3.9)$$

$R(w+u+1, i|\beta_j)$ は、チャンネル # i のパケットがフロー j に属しそしてチャンネル # i の状態がチャンネル #1 から # $(M - 1)$ までの中で最大でその値が $w + u + 1$ をとる確率である。

より正確には、以下のすべての条件を満足する確率として定義されるものである。

- チャンネル # i のパケットがチャンネル #0 のパケットと同じフロー j に属する。
- チャンネル # i のパケットの残余受信失敗回数が n 。
- チャンネル #1 から # $(i-1)$ までのパケットがフロー j に属さないか、またはフロー j に属す場合はその残余受信失敗回数がチャンネル # i のパケットのそれと等しいか小さい。
- チャンネル # $(i+1)$ から # $(M-1)$ までのパケットがフロー j に属さないか、またはフロー j に属す場合はその残余受信失敗回数がチャンネル # i のパケットのそれより小さい。

$U(n|\beta_j)$ をあるチャンネルのパケットがフロー j に属さないか、またはフロー j に属す場合はその残余受信失敗回数が n 以下の確率と定義する。

$$\begin{aligned}
U(n|\beta_j) &= 1 - \beta_j + \beta_j \sum_{k=0}^n p_r(k) \\
&= 1 - \beta_j + \beta_j \frac{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{n+1}}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}
\end{aligned} \tag{3.10}$$

各チャネルは独立なので、チャネル #1 から #(M-1) までの残余受信失敗回数が、それぞれ r_1, \dots, r_{M-1} となる確率は各々の状態確率の積により与えられる。

$$P_{rob}(r_1, \dots, r_i, \dots, r_{M-1}) = \prod_{k=1}^{M-1} p_r(r_k) \tag{3.11}$$

この式に基づいて、 $R(n, i|\beta_j)$ は上記で列挙した条件を満足するよう計算される。

$$\begin{aligned}
R(n, i|\beta_j) &= \prod_{k=1}^{i-1} U(n|\beta_j) \cdot \beta_j p_r(n) \cdot \prod_{k=i+1}^{M-1} U(n-1|\beta_j) \\
&= \beta_j p_r(n) U(n|\beta_j)^{i-1} U(n-1|\beta_j)^{M-1-i}
\end{aligned} \tag{3.12}$$

式 (3.8)、(3.10)、(3.12) を用いることにより、 $i \neq 0$ の場合の順序制御による遅延時間の確率分布を求めることが出来る。

3.3.1.2 順序制御が行われない場合 ($i = 0$)

チャネル #0 において正しく受信されたパケットが他のチャネルのパケットを待つ場合に順序制御による遅延時間が発生する。 $i = 0$ は、チャネル #0 において正しく受信されたパケットが同じチャネルの別のパケットを待つこととなり、このような事象は存在しないので、 $w \neq 0$ となる確率は 0 である。しかし、順序制御による遅延時間が 0 の場合は存在するので、 $P_{resq}(0, 0|\beta_j)$ の確率は 0 でない値をとる。図 3.7 は他のチャネルのパケットがすべてフロー j に属しているが、順序制御による遅延時間が発生しない場合の例を示している。

この図でチャネル #0 のパケットは $u(u = 3)$ 回の受信失敗の後に正しく受信されている。時刻 t_0 で他のチャネルの状態 (残余受信失敗回数) を観測する。時刻 t_0 はチャネル #0 のパケットの最初の送信が受信側に到着した時刻である。各チャネルのパケットがフロー j に属さないか、フロー j に属す場合はその残余受信失敗回数が u 以下であれば、順序制御による遅延時間は生じない。これから、 $P_{resq}(w, 0|\beta_j)$ は以下ようになる。

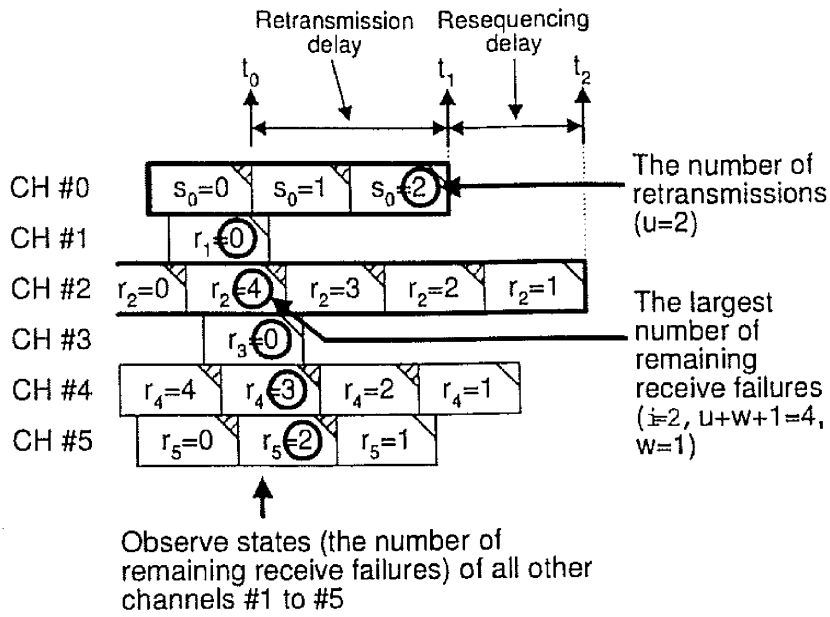


図 3.6 · 順序制御による遅延時間の例

$$P_{reseq}(w, 0 | \beta_j) = \begin{cases} \sum_{u=0}^{N_r} p_t(u) U(u | \beta_j)^{M-1} & w = 0 \\ 0 & w \neq 0 \end{cases} \quad (3.13)$$

3.3.1.3 複数のフローが存在する場合の順序制御による遅延時間の確率分布

m 本のフローが存在する場合、パケット全体での順序制御による遅延時間の確率分布は次のように求めることができる。

$$P_{reseq}(w, i) = \sum_{j=1}^m \beta_j P_{reseq}(w, i | \beta_j) \quad (3.14)$$

m 本のフローにおけるパケット生成確率が同じである場合、確率分布は以下のようになる。ここで、 $\beta = 1/m$ 。

$$P_{reseq}(w, i) = P_{reseq}(w, i | \beta) \quad (3.15)$$

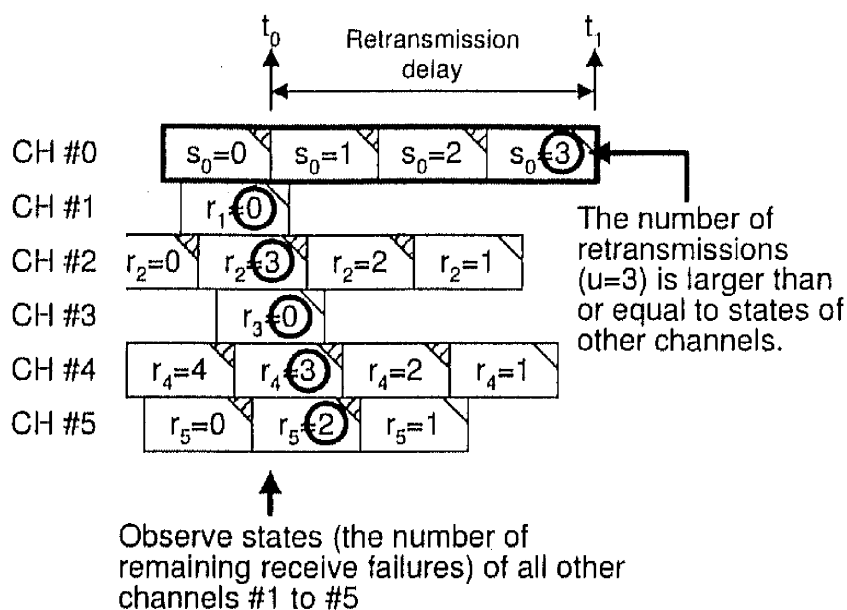


図 3.7 順序制御による遅延時間が発生しない場合の例

3.3.2 再送と順序制御の両方による遅延時間の確率分布

チャンネル #0 のパケットが u 回の受信失敗の後に正しく受信された場合に再送と順序制御の両方による遅延時間が $uM + i$ スロットとなる確率を $P_{delay}(u, i | \beta_j)$ とする。

3.3.2.1 順序制御が行われる場合 ($i \neq 0$)

再送と順序制御の両方による遅延時間が $uM + i$ スロットとなるのは、次の場合である。チャンネル #0 でパケットが u 回以下の回数受信失敗した後に正しく受信されたとする。他のすべてのチャンネルの状態（残余受信失敗回数）を時刻 t_0 で観測する。時刻 t_0 はこのパケットの最初の送信が受信側に到着した時刻である。フロー j に属するパケットを運ぶチャンネルの中で、チャンネル # i の状態が最大で（チャンネル #0 を除き）、その値が $u + 1$ 。この場合の確率は以下のように求められる。

$$\begin{aligned}
P_{delay}(u, i|\beta_j) &= \left(\sum_{k=0}^u p_t(k) \right) R(u+1, i|\beta_j) \\
&= \beta_j q_t(u) p_r(u+1) U(u+1|\beta_j)^{i-1} U(u|\beta_j)^{M-1-i} \\
&= \beta_j q_t(u) p_r(u+1) \frac{U(u|\beta_j)^{M-1}}{U(u+1|\beta_j)} \left(\frac{U(u+1|\beta_j)}{U(u|\beta_j)} \right)^i
\end{aligned} \tag{3.16}$$

ここで、 $q_t(u)$ はパケットが u 回以下の受信失敗の後に正しく受信される確率である。

$$q_t(u) \triangleq \sum_{k=0}^u p_t(k) = \frac{1 - \varepsilon^{u+1}}{1 - \varepsilon^{N_r+1}} \tag{3.17}$$

3.3.2.2 順序制御が行われない場合 ($i = 0$)

チャンネル #0 のパケットが u 回の受信失敗後に正しく受信されたとする。他のチャンネルの状態（残余受信失敗回数）を時刻 t_0 で観測する。時刻 t_0 は前と同様にこのパケットの最初の送信が受信側に到着した時刻である。各チャンネルにおいて、フロー j に属するパケットが存在しないか、存在する場合はその残余受信失敗回数が u 以下であれば、順序制御は行われず順序制御による遅延時間は 0 となる。この場合の確率は、以下のように求められる。

$$P_{delay}(u, 0|\beta_j) = p_t(u) U(u|\beta_j)^{M-1} \tag{3.18}$$

3.3.2.3 複数のフローが存在する場合

順序制御による遅延時間の場合と同様に、再送と順序制御の両方による遅延時間の確率分布は以下のように計算できる。

$$P_{delay}(u, i) = \sum_{j=1}^m \beta_j P_{delay}(u, i|\beta_j) \tag{3.19}$$

各フローにおけるパケット生成確率が同じであるならば、再送と順序制御の両方による遅延時間の確率分布は次のようになる。

$$P_{delay}(u, i) = P_{delay}(u, i|\beta) \tag{3.20}$$

3.4 解析に基づく数値計算結果

解析に基づいてSR-ARQを衛星回線に用いた場合について数値計算を行った。ここで、衛星回線の帯域を1.5 Mbit/s、パケット長を1500 Byte、衛星回線の伝搬遅延時間を0.255 secとした。これらの条件から M の値は64となる。解析の妥当性を検証するため、同じ条件でシミュレーションも行った。

3.4.1 α を変化させた場合

表3.1は、 α を変化させた場合について、利用率、再送による平均遅延時間、順序制御による平均遅延時間、再送と順序制御の両方による平均遅延時間を示している。ここで、 $\beta = 1$ としている。これは、従来の全順序制御が用いられている場合または上位層のフロー数が1の場合に相当する。パラメータ α は伝送リンクの利用率を変化させることを意図して導入されており、 α の値を小さくすると利用率が減少し、ほぼ α に近い値をとる。しかし、パケット誤り率が大きくなると、再送によるトラヒックが増加するため利用率は α より大きくなる。

図3.9は α を変化させた場合について、再送による平均遅延時間、順序制御による平均遅延時間を示している。ここで、平均遅延時間はフレームを構成するスロット数 M により規格化している。この図で、左端の列は再送による平均遅延時間、他の列は順序制御による平均遅延時間を示している。表3.1及び図3.9から、順序制御による平均遅延時間が再送による平均遅延時間に比べ支配的であることが分かる。この理由は図3.8を参照して、以下のように説明できる。再送による遅延時間の場合、伝送誤りで失われ再送されたパケットのみが再送による遅延を被る。図3.8では再送パケットは6スロットの遅延となっている。一方、順序制御による遅延時間は正しく受信された多数の後続パケットが遭遇する。3.8では後続の5パケットがそれぞれ5, 4, 3, 2, 1スロットの遅延を被っている。このように順序制御は後続の複数パケットに遅延を発生させるため、平均で評価するとはるかに影響が大きくなる。

表に示すように、パケット誤り率が一定であれば順序制御による遅延時間はチャンネルの利用率が減少すると小さくなる。これは利用率が小さくなると、順序制御に巻き込まれるパケット数が減少するためである。

図3.10は順序制御のみによる遅延時間の確率分布を示している。ここで、パケット誤り率 ε は0.1、パラメータ α は0.9としている。また、図3.11は同じ条件における再送と順序制御の両方による遅延時間の確率分布を示している。これらの図において、シミュレーション結果も併せて示している。解析による数値計算結果とシミュレーション結果が

表 3.1 利用率、再送による平均遅延時間 (slot)、順序制御による平均遅延時間 (slot)、再送及び順序制御による平均遅延時間 (slot)

$N_r = 3, \beta = 1.0$			retrans.	reseq.	total
α	ϵ	ρ	delay	delay	delay
1.00	0.0001	1.0000	0.0064	0.2012	0.2076
	0.0010	1.0000	0.0641	1.9770	2.0410
	0.0100	1.0000	0.6465	16.7627	17.4092
	0.1000	1.0000	7.0855	71.8210	78.9065
0.90	0.0001	0.9000	0.0064	0.1811	0.1875
	0.0010	0.9001	0.0641	1.7831	1.8472
	0.0100	0.9009	0.6465	15.3821	16.0286
	0.1000	0.9091	7.0855	69.4644	76.5499
0.80	0.0001	0.8000	0.0064	0.1610	0.1674
	0.0010	0.8002	0.0641	1.5884	1.6524
	0.0100	0.8016	0.6465	13.9441	14.5906
	0.1000	0.8163	7.0855	66.7985	73.8840
0.70	0.0001	0.7000	0.0064	0.1409	0.1473
	0.0010	0.7002	0.0641	1.3928	1.4569
	0.0100	0.7021	0.6465	12.4459	13.0923
	0.1000	0.7216	7.0855	63.7298	70.8153
0.60	0.0001	0.6000	0.0064	0.1208	0.1272
	0.0010	0.6002	0.0641	1.1964	1.2605
	0.0100	0.6024	0.6465	10.8845	11.5309
	0.1000	0.6250	7.0855	60.1194	67.2049

よく一致しており、解析が妥当であることを示している。遅延時間分布は階段状に値が減少するが、これは再送回数が異なるためである。0 から 63 スロットの間のように再送回数が同じ場合、遅延時間が大きくなると確率が大きくなる傾向が見られる。これは、再送されたパケットの直後に送信されたパケットは順序制御に巻き込まれやすくかつ順序制御による遅延時間も再送による遅延時間の値に近くなることによるものである。

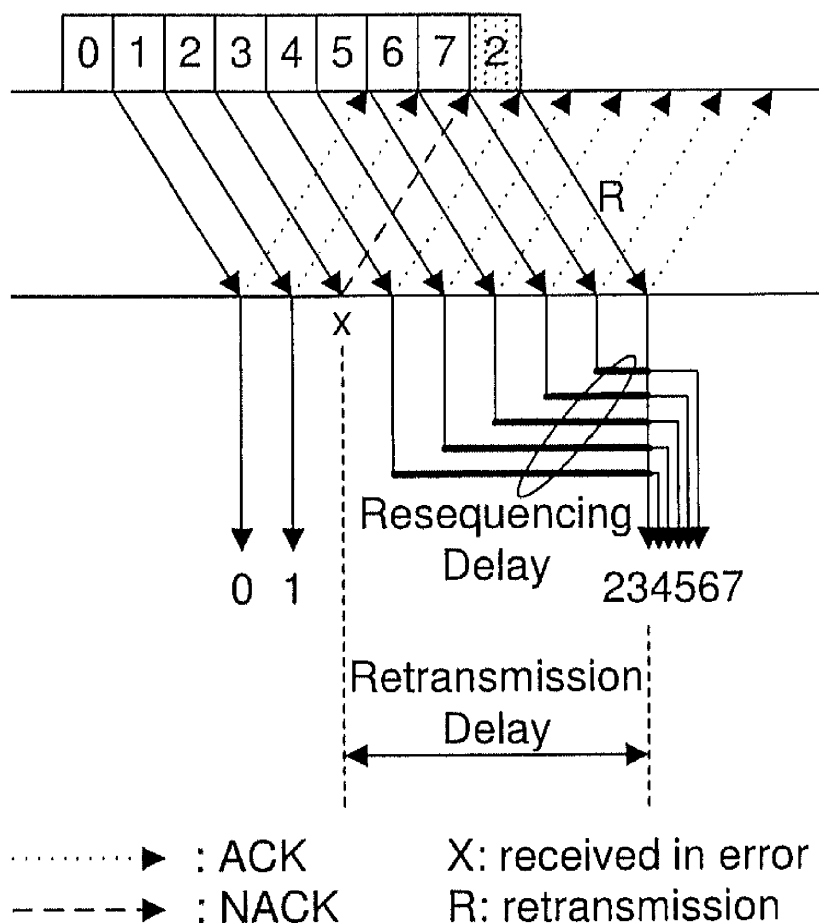


図 3.8 再送よりも順序制御による遅延時間の影響が大きい理由

3.4.2 最大再送回数 N_r を変化させた場合

表 3.2 は最大再送回数 N_r を 1 から 5 まで変化させた場合の遅延時間とパケット損失率 p_L を示している。ここで、 α は 0.9 の値としている。他の条件は表 3.1 と同じである。 N_r の値が小さい場合、遅延時間は小さいもののパケット誤り率が大きい場合にパケット損失率が無視できない値となる。 N_r の値が 3 より大きい場合、パケット損失率は小さくなり遅延時間は N_r が 4 と 5 でほぼ同じ値となっている。

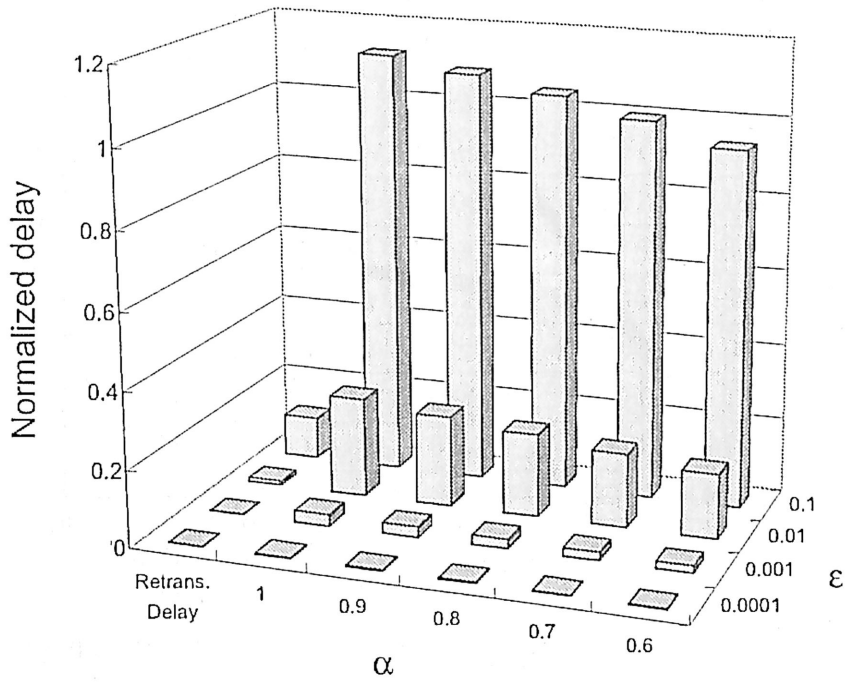


図 3.9 再送による平均遅延時間及び α を変化させた場合の順序制御による平均遅延時間 ($M = 64, N_r = 3, \beta = 1.0$)

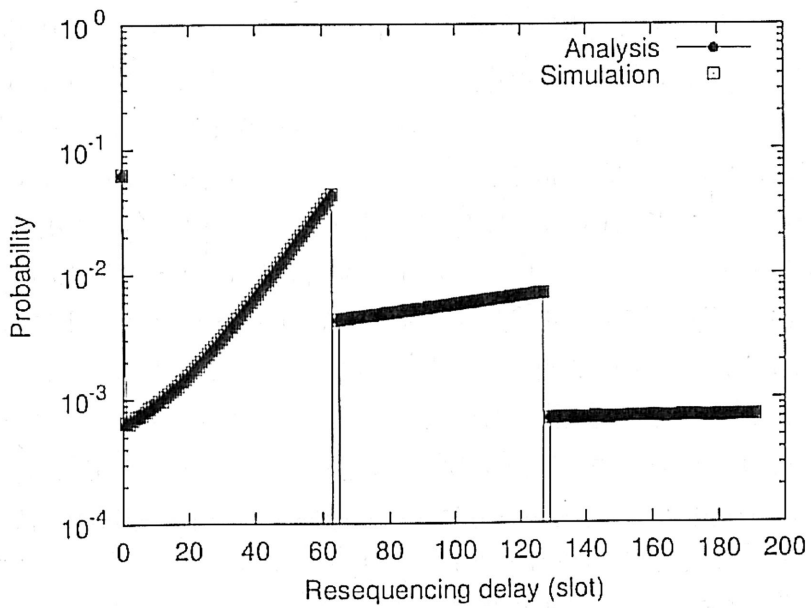


図 3.10 順序制御による遅延時間分布 ($\epsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 1.0$)

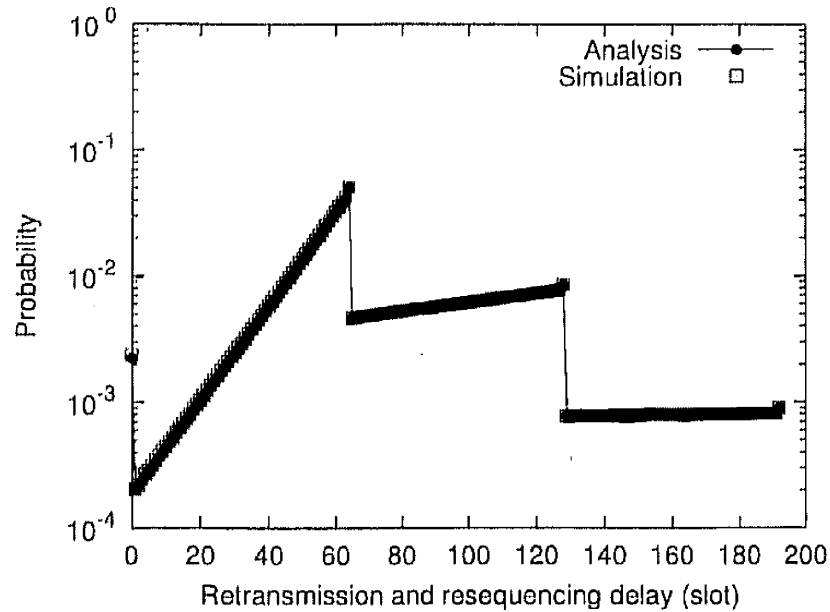


図 3.11 再送と順序制御による遅延時間分布 ($\epsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 1.0$)

3.4.3 β を変化させた場合

表 3.3 は β を変化させた場合の再送による平均遅延時間、順序制御による平均遅延時間、及び再送と順序制御による平均遅延時間を示している。ここで、 α は 0.9 としている。

図 3.12 は β を変化させた場合の再送による平均遅延時間と順序制御による平均遅延時間を示している。図 3.9 の場合に述べたように、左端の列は再送による平均遅延時間、他の列は β を変化させた場合の順序制御による平均遅延時間を示している。これらの遅延時間はフレームを構成するスロット数 M で規格化している。 β の値が 1.0、0.5、0.2、0.1 は速度が同じフロー数がそれぞれ 1、2、5、10 の場合に相当する。従来の全順序制御は β が 1.0 の場合で、上位層フローの数によらずすべてのフローの packets について順序が保存される。上位層フローの数が増加するとともに順序制御による遅延時間が顕著に減少することが分かる。 β の値が小さい場合の遅延時間と β の値が 1.0 の場合の遅延時間を比較することにより HOL ブロッキングの影響が大きいことは明らかである。例えば、パケット誤り率 ϵ が 0.01 の場合、 $\beta = 1.0$ における順序制御の平均遅延時間は 15.4 スロットであるが、 $\beta = 0.1$ における順序制御の平均遅延時間は 1.8 スロットに減少する。HOL ブロッキングの影響がなければ、順序制御の平均遅延時間は 11.7% に減少することになる。これは 1 本のフローにおける単位時間当りのパケット数が減少すると、順序制御に巻き込まれるパケット数が減ることによる。伝送リンク上で同時にアクティブなフローの数

表 3.2 利用率、再送による平均遅延時間 (slot)、順序制御による平均遅延時間 (slot)、再送及び順序制御による平均遅延時間 (slot)

$\alpha = 0.9, \beta = 1.0$			retrans.	reseq.	total
N_r	ε	p_L	delay	delay	delay
1	0.0001	1.00e-008	0.0064	0.1811	0.1875
	0.0010	1.00e-006	0.0639	1.7778	1.8417
	0.0100	1.00e-004	0.6337	14.9246	15.5583
	0.1000	1.00e-002	5.8182	47.2160	53.0342
2	0.0001	1.00e-012	0.0064	0.1811	0.1875
	0.0010	1.00e-009	0.0641	1.7831	1.8472
	0.0100	1.00e-006	0.6463	15.3740	16.0203
	0.1000	1.00e-003	6.9189	65.0906	72.0095
3	0.0001	1.00e-016	0.0064	0.1811	0.1875
	0.0010	1.00e-012	0.0641	1.7831	1.8472
	0.0100	1.00e-008	0.6465	15.3821	16.0286
	0.1000	1.00e-004	7.0855	69.4644	76.5499
4	0.0001	1.00e-020	0.0064	0.1811	0.1875
	0.0010	1.00e-015	0.0641	1.7831	1.8472
	0.0100	1.00e-010	0.6465	15.3822	16.0287
	0.1000	1.00e-005	7.1079	70.2163	77.3242
5	0.0001	1.00e-024	0.0064	0.1811	0.1875
	0.0010	1.00e-018	0.0641	1.7831	1.8472
	0.0100	1.00e-012	0.6465	15.3822	16.0287
	0.1000	1.00e-006	7.1107	70.3243	77.4350

はユーザの数やアプリケーションに依存する。HTTP/1.0 を用いる WEB ブラウザでは同時に最大 8 本の TCP コネクション、HTTP/1.1 では最大 2 本の TCP コネクションをオープンすることが報告されている [59]。これは HOL ブロッキングの影響が 1 ユーザの場合でも存在することを示している。

図 3.13 は再送と順序制御による遅延時間の確率分布を示している。ここで、上位層フローの数は 5 ($\beta = 0.2$) としている。他の条件は図 3.11 と同じである。図 3.14 は図 3.13 と同じ条件で、パケット誤り率 ε を 0.01 とした場合の再送と順序制御による遅延時間の確率分布を示している。解析による数値計算結果とシミュレーション結果が完全に重なっ

表 3.3 β を変化させた場合の再送による平均遅延時間、順序制御による平均遅延時間及び再送と順序制御による平均遅延時間 ($M = 64, N_r = 3, \alpha = 0.9$)

$N_r = 3, \alpha = 0.9$		retrans.	reseq.	total
β	ϵ	delay	delay	delay
1.00	0.0001	0.0064	0.1811	0.1875
	0.0010	0.0641	1.7831	1.8472
	0.0100	0.6465	15.3821	16.0286
	0.1000	7.0855	69.4644	76.5499
0.50	0.0001	0.0064	0.0906	0.0970
	0.0010	0.0641	0.8998	0.9639
	0.0100	0.6465	8.3823	9.0288
	0.1000	7.0855	52.0038	59.0893
0.20	0.0001	0.0064	0.0363	0.0427
	0.0010	0.0641	0.3619	0.4260
	0.0100	0.6465	3.5375	4.1839
	0.1000	7.0855	29.7156	36.8011
0.10	0.0001	0.0064	0.0181	0.0245
	0.0010	0.0641	0.1813	0.2454
	0.0100	0.6465	1.8012	2.4477
	0.1000	7.0855	17.2001	24.2856

ており、解析が妥当であることを示している。これらの図で、再送による遅延時間（スロット数が M の倍数）で確率のピークが目立っている。これはフローごとに順序制御が行われるため、順序制御による遅延時間の確率が減少し、相対的に再送による遅延時間が顕著に現れることによる。

図 3.15 は複数あるフローの速度が均一ではない場合について、遅延時間の確率分布を示している。この図で上位層フローの数は 6 としている。1 本のフローの確率 (β_1) を 0.5、他の 5 本のフローの確率 ($\beta_i, i = 2, 3, 4, 5, 6$) を 0.1 としている。他の条件は図 3.14 と同じである。解析から求めた数値計算結果とシミュレーション結果はよく一致しており、この章で述べた解析は異なる速度のフローが混在する場合にも適用できることが分かる。

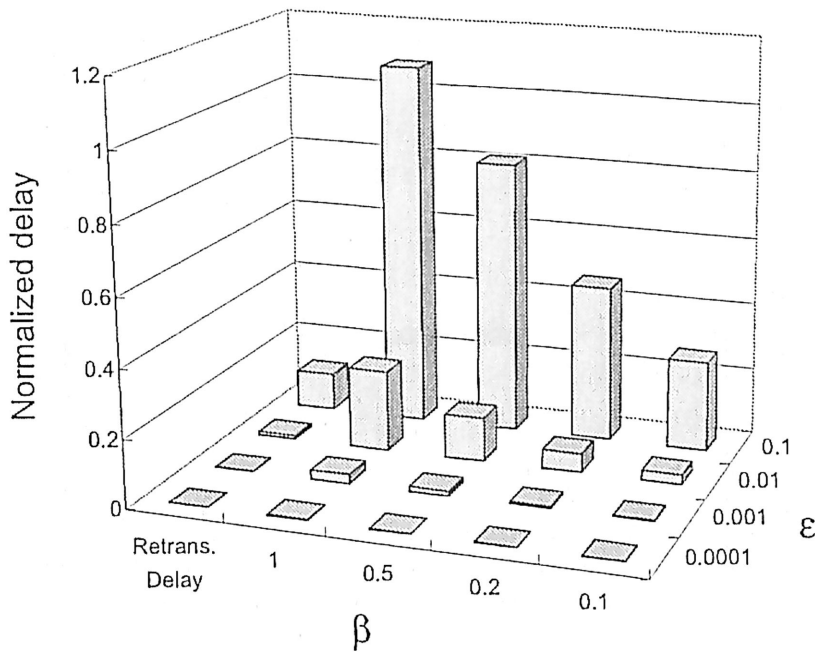


図 3.12 再送による平均遅延時間及び β を変化させた場合の順序制御による平均遅延時間 ($M = 64, N_r = 3, \alpha = 0.9$).

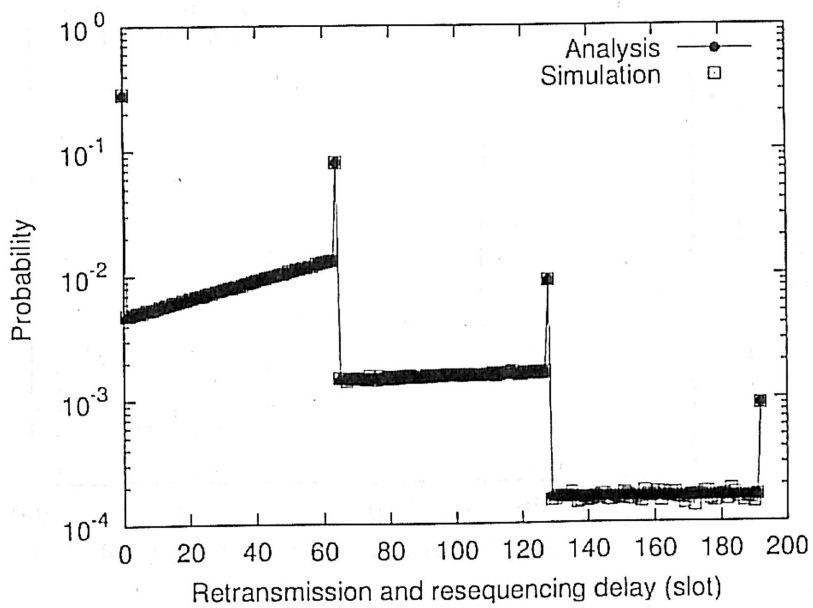


図 3.13 順序制御のみによる遅延時間の確率分布 ($\epsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta = 0.2$)

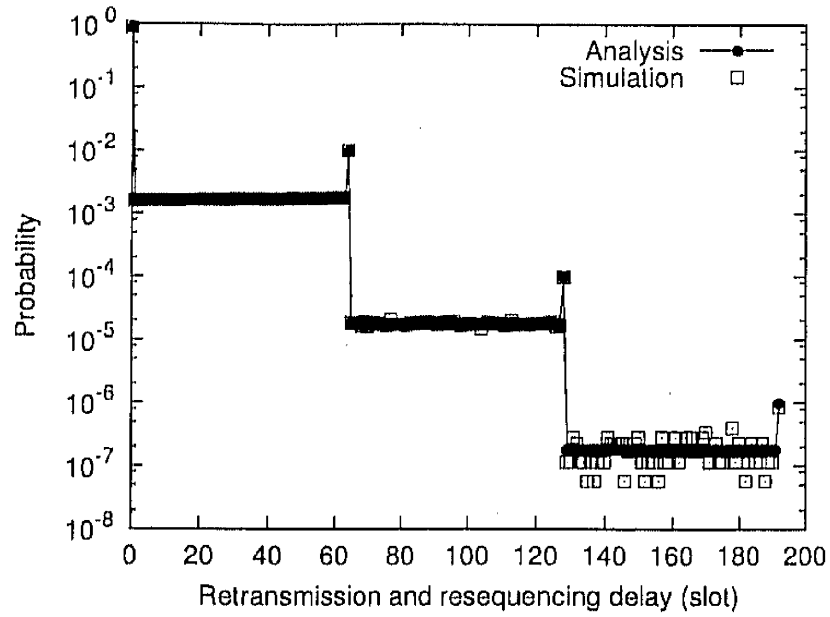


図 3.14 再送と順序制御による遅延時間の確率分布 ($\epsilon = 0.01, M = 64, N_r = 3, \alpha = 0.9, \beta = 0.2$).

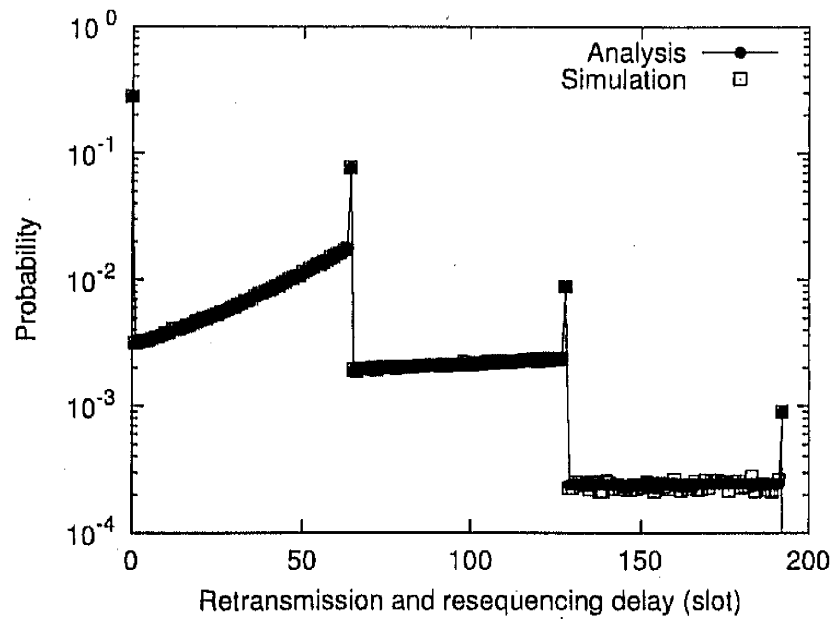


図 3.15 再送と順序制御による遅延時間分布 ($\epsilon = 0.1, M = 64, N_r = 3, \alpha = 0.9, \beta_1 = 0.5, \beta_i = 0.1, i = 2, 3, 4, 5, 6$)

3.5 むすび

この章では SR ARQ について順序制御のみによる遅延時間の確率分布及び再送と順序制御による遅延時間の確率分布に関する解析を示した。最大再送回数を制限した場合の数値計算が可能で、同じ条件によるシミュレーション結果と計算とが一致することにより解析が正しいことを確認した。本論文で述べた解析により、すべてのパケットの順序を保存する従来の全順序制御方式と HOL ブロッキングを回避した場合の両方の遅延時間分布を求めることが可能である。また、HOL ブロッキングを回避した場合においては、速度が異なるフローの遅延時間分布も求めることができる。最大再送回数をある程度大きくすればパケット損失は無視できるレベルとなり、本論文で述べた解析は最大再送回数を制限しない場合の評価にも適用可能と考えられる。計算結果から全順序制御方式において、再送そのものよりも再送に伴う順序制御による遅延時間が支配的であること、順序制御に伴う HOL ブロッキングにより大きな遅延時間が発生することを示した。

衛星通信の場合について計算結果を示したが、今後、地上無線通信の速度が向上するにつれてラウンドトリップ時間に複数のパケットが送信されるようになりレイヤ 2 の誤り回復プロトコルとして SR ARQ が用いられるものと予想される。本論文の計算手法はこのような高速の地上無線通信においても適用できる。

第4章

PFRS 方式の提案と性能評価

4.1 まえがき

これまで述べたように、帯域と遅延時間の積の大きい環境での伝送誤りに対し再送により高い性能を得るためには、伝送誤りにより失われたパケットのみを選択的に再送する SR ARQ 方式が必要となる。この SR ARQ ではパケットを正しく受信して、その前のパケットが伝送誤りにより受信されていないことを検出した場合、再送されてくるパケットを受信するまで正しく受信したパケットを保留する順序制御が必要となる。この順序制御により、2章においてバースト的なトラヒックが出力される問題、3章において HOL ブロッキングにより不必要な遅延時間が発生する問題のあることを述べた。この章ではこのような問題を解決する方式を提案し評価を行う。

以下、この章では最初にレイヤ 2 (リンク層) における再送機能と上位層のフローとの関係について検討し、次にレイヤ 2 で ARQ を行う場合の順序制御に関して従来の問題を解決する PFRS 方式を提案する。提案した PFRS 方式を衛星通信に適用した場合についてシミュレーションによる評価を行い、性能が大幅に改善されることを示す。

4.2 レイヤ 2 における ARQ と上位層フローとの関係

ARQ の機能を OSI 参照モデルのどのレイヤで実現するかについては種々の選択肢がある [48]。

4.2.1 レイヤ 2 における ARQ の必要性

一般に IP ネットワークでは信頼性の高い通信が必要とされる場合には TCP が用いられ、伝送誤りが発生した場合はパケットの再送がエンドツーエンドで行われる。地上の光

ファイバー回線を用いる IP ネットワークでは、回線品質が良いため伝送誤りの発生は少なく、またエンドツーエンドの遅延時間も比較的小さい。

一方、地上無線や衛星通信では、常に安定した回線品質が得られるとは限らないため再送が必須であるが、TCP によるエンドツーエンドの誤り回復では十分な性能が得られない問題がある。TCP では、パケットの喪失は中継するネットワークでの輻輳により生じるものとし、タイムアウトや重複 ACK によりパケットの喪失を検出して再送を行う場合、輻輳を回避するためにウィンドウサイズを減少させる制御を行う。このため、頻りにパケット誤りが発生する状況では、ウィンドウサイズが小さい状態が続き十分な性能が得られない問題がある。特に衛星通信のようにラウンドトリップ時間の大きい場合は、送達確認が返るまでの時間が長いことウィンドウサイズが回復するまでに時間を要し、この間にまたパケットの伝送誤りが発生すると、現行の TCP メジャーバージョンではウィンドウサイズが十分な値に回復しない状況が続く可能性がある。この問題を解決するため、一つの方向として Spoofing[60] のように中継ノードで TCP の ACK を返送する方式や、Snoop[61] のように TCP ヘッダの情報に基づいてリンクバイリンクに再送制御を行う方式が研究されている。しかし、これらのアプローチでは TCP がエンドツーエンドの通信とならないため、IPSec が使えない問題や TCP のヘッダに基づいて処理を行う中継ノードが故障した場合に、TCP による回復が行われない問題などが指摘されている [62]。

本論文ではエンドツーエンドの TCP を中継ノードで終端することなく使用することを前提とし、地上無線回線や衛星回線での誤りをリンクバイリンクで回復する方式を検討する。また TCP のウィンドウサイズの制限により高速な衛星回線では、帯域をフルに利用できないことが指摘されているが、多数の TCP コネクションが衛星回線に多重化され、帯域をフルに使用する環境を想定する。一般に地上無線システムでは無線回線部分でリンクバイリンクに再送が行われており、同様のアプローチとなっている。

また、TCP とともに UDP によるストリームデータ転送においても、信頼性の高い通信を実現するためにレイヤ 2 において SR ARQ を用いる場合を想定し、SR ARQ による UDP の遅延時間及びその変動を小さくすることを狙っている。

4.2.2 SR ARQ と上位層のフローとの関係

無線回線区間の SR ARQ と上位層のフローの関係について、図 4.1 に示す 2 方式が考えられる。

- 上位層のフローと 1 対 1 に対応して SR ARQ のコネクションを設定する。
- 1 本の SR ARQ コネクションに上位層の複数フローを多重化する。

以下、この章で前者を個別フロー ARQ 方式、後者を全フロー ARQ 方式と呼ぶ。

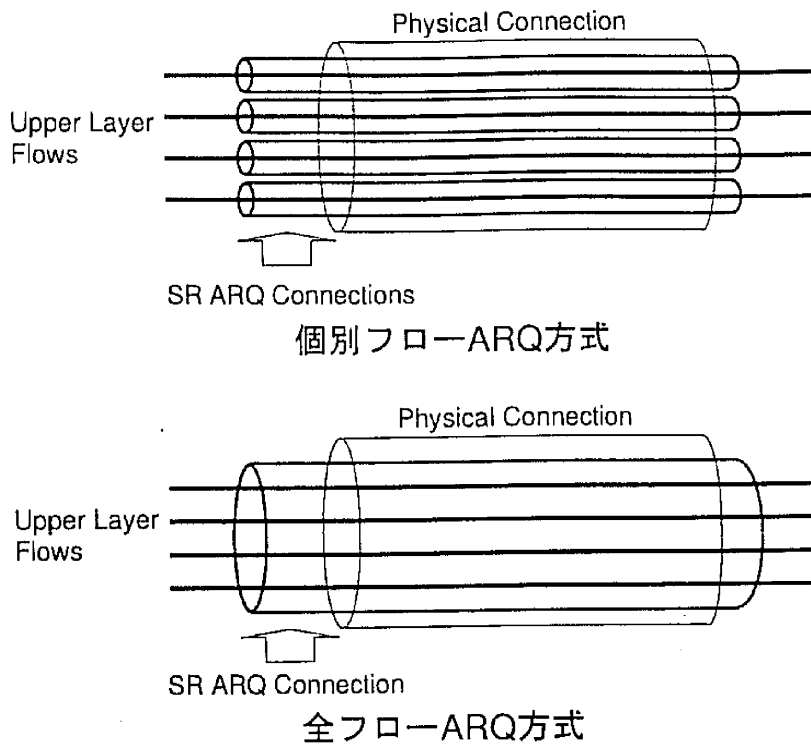


図 4.1 個別フロー及び全フロー ARQ 方式における ARQ のコネクションと上位層フローとの関係

個別フロー ARQ 方式では、ARQ の送達確認やポーリングが各 ARQ コネクションで独立に行われオーバーヘッドが大きくなる問題がある。また、ARQ コネクションの設定及び解放を上位層フローの開始及び終了に伴って動的に行う必要があり、これらによるオーバーヘッド及びコネクション管理が複雑化する問題もある。更に ARQ のコネクション設定に伴って、ARQ の情報を管理するテーブルや受信バッファのためのリソース確保が必要となるが、無線回線を通ずるフロー数の上限を規定することができないため、どの程度のリソースが必要か決定するのが困難な問題もある。実際の ARQ では、パケットの喪失発生後、最初に正しくパケットを受信した時点で、通番の飛びによりパケットの喪失を検出することが多い。ARQ をフロー単位に行うと、パケットの頻度が小さいフローの ARQ コネクションではパケット喪失後に次のパケットを受信するまでに時間を要し、パケットの喪失検出が遅れる問題もある。

全フロー ARQ 方式では、単一の静的な ARQ コネクションを設定すればよく、また多重化された大きなトラヒックでパケットの喪失を検出するので、検出までの時間が短くなる。しかし、多重化された大きなトラヒック全体に対し SR ARQ を行うので、順序制御

により保留されるパケット数が多くなり平均遅延時間が大きくなる問題、また順序制御により必要となるバッファ量及び出力されるパケットのバースト性も大きくなる問題がある。一方、個別フロー ARQ 方式ではフローごとに順序制御を行うのでこれらの問題が少ない。

SR ARQ において、従来は再送制御と順序制御が明確に分けられていなかったが、本論文ではこれらを分離し、パケット喪失の検出と再送は全フロー ARQ 方式、順序制御は個別フロー ARQ 方式の長所を取り入れた PFRS (Per Flow Re-Sequencing) 方式を提案する。

4.3 PFRS (Per Flow Re-Sequencing) 方式

PFRS 方式は全フロー ARQ 方式を基本とし、パケット喪失の検出と再送は多数のフローを多重化した1本のSR ARQにより行う。ここで、同一のフローではパケットの順序保証が必要であるが、異なるフローに属するパケット間ではパケットの順序を保証する必要はない。これから、PFRS 方式では受信側での順序制御はフローごとに個別に行う。3章で説明したように、SR ARQ に多重化したすべてのパケットについて順序を保証する従来の全フロー ARQ 方式では、SR ARQ で不必要にパケットの保留と遅延を生じる場合がある。図 4.2 で、a と b の2本のフローが1本のSR ARQ に多重化されているとしている。フロー a のパケット a1 と a2 が伝送誤りにより再送されているが、全フロー ARQ 方式ではフロー b のパケット b1 と b2 が、異なるフロー a のパケット a1 を受信するまで不必要に保留されており、一種の HOL ブロッキングが発生する。提案する PFRS 方式は図 4.3 に示すように、パケットの送達確認や再送は従来のように全フローまとめて SR ARQ で行うが、順序制御についてはフロー単位に行うため、フロー b のパケット b1 と b2 は受信後保留されずに直ちに上位層に転送される。このような PFRS 方式により、HOL ブロッキングによる不必要なパケット保留を解消でき、以下の効果を期待できる。

- パケットの平均遅延時間が小さくなる。
- バースト的なパケット出力は全体のフローではなく1本のフローごとに生じ、バースト性が小さくなる。
- 順序制御が個々のフローごとに独立に行われるので、順序制御のために保留される受信パケット数が少なくなり、平均所要バッファ量も小さくなる。
- パケットの平均遅延時間が小さくなるため、上位層として TCP を用いた場合にスループットが全フロー ARQ 方式に比べ大きくなる。また UDP を用いた場合も遅延時間及び遅延時間揺らぎを削減する効果を期待できる。

このような PFRS 方式と類似の研究として、Bharath-Kumar らはパケットを複数回線に分散させた場合の順序制御に関して、順序制御をバーチャルサーキット (VC) ごとに行う場合とリンク全体で行う場合とを比較している [107]。また Yum らは、この研究を拡張してリンクごとに順序制御を行う場合とエンドツーエンドで順序制御を行う場合を評価している [108]。しかし、これらの研究では指数分布のサービス時間 (遅延時間) を前提としており、本論文が対象とする SR ARQ の再送におよぶ遅延時間を扱うことができない。

PFRS 方式実現のためには以下の機能が必要となる。

- 上位層のフロー識別
- フローごとのパケット順序識別

PFRS 方式では可能な限り細かいフローを識別してその単位で順序制御を行うのがよい。フロー識別は IP ヘッダに含まれる宛先及び送信元の IP アドレスとポート番号及び上位プロトコル ID により可能となる。

フローごとの順序情報については、レイヤ 2 の送信側で上位層のフローを識別し、レイヤ 2 のヘッダにフロー識別情報とフローごとの順序情報を設定し、受信側では上位層を意識することなくレイヤ 2 のヘッダ情報により順序制御を行うことが望ましい。これからレイヤ 2 のヘッダにフロー識別子とフローごとの通番を設ける方式などが考えられる。しかし、同時に流れる上位層フロー数の最大値を把握できないのでフロー数の制限のないフロー識別方法、フローの発生と終了の検出及びそれに伴うフロー識別子と通番の管理、これらの実現においてオーバーヘッドを小さくすることが課題となる。このような課題を解決する方式については 5 章で詳細を説明する。

以上のように、PFRS 方式では SR ARQ において上位の IP 層や TCP 層のヘッダ情報の一部を参照する必要がある。しかし、上位層ヘッダの変更は行わないので、上位層として IP 及び TCP が用いられた場合、エンドツーエンドの通信を保存する。

上記の PFRS 方式による効果は、1 本の SR ARQ に多数のフローが多重化されている場合に顕著になるものと予想される。このような PFRS 方式の効果を確認するため、次節に述べるシミュレーションを行った。

4.4 シミュレーションモデル

シミュレーションは VINT プロジェクトのネットワークシミュレータ ns-2[63] を用い、具体的なレイヤ 2 の SR ARQ プロトコルとして SSCOP (Service Specific Connection Oriented Protocol) を実装した。このプロトコルは Q.2110 として ATM のシグナリング用に ITU で標準化されたものである [65][66]。24 ビット長の通番を用いて SR ARQ を

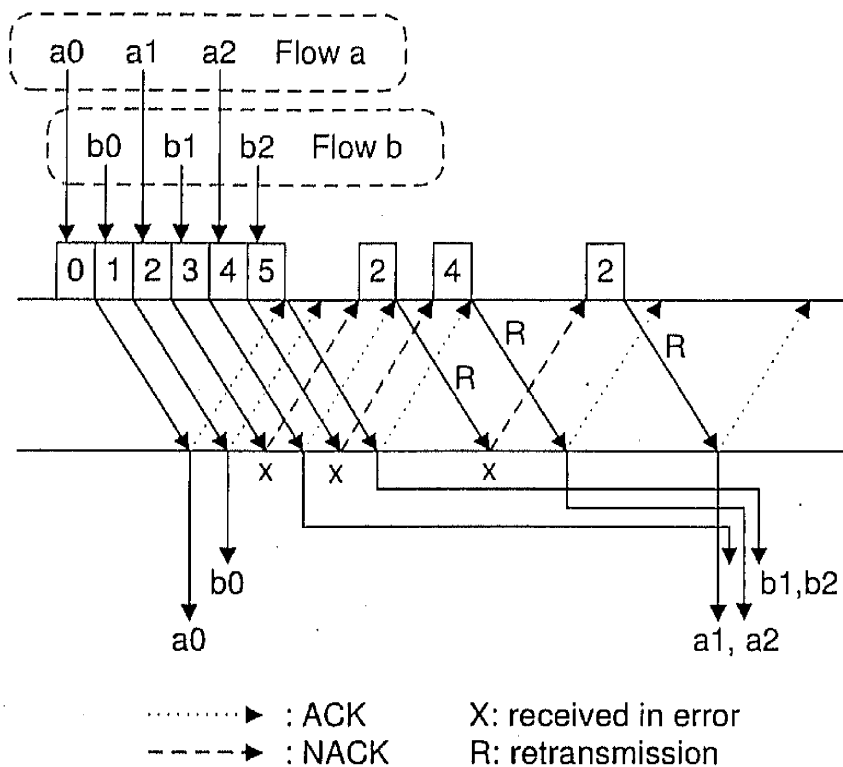


図 4.2 全フロー ARQ 方式のシーケンス例

実現しており、衛星通信のような遅延時間の大きい環境で高速通信する場合に適している。地上無線におけるレイヤ 2 による ARQ では伝送路の特性に合わせて上位層のデータをより小さなブロックに分割し、受信側ではこれを組み立てた後に上位層に渡すセグメンティングとリアセンブリング機能が行われるが、伝搬遅延時間の大きい衛星通信で IP パケットの分割を行うと ARQ におけるアウトスタンディングパケット数が大きくなることを考慮し、本シミュレーションでは IP パケットを分割せずそのまま SR ARQ で送受信するものとしている。

SSCOP では図 4.4(a) に示すように、受信側が通番の不連続を検出すると抜けたパケットの通番情報を選ぶ USTAT によりパケットの喪失が送信側に通知され、選択再送が行われる。また USTAT が伝送誤りにより失われた場合や、パケットが失われた後で正しく受信したパケットがない場合などは、図 4.4(b) のように送信側から周期的に送信されるポーリング POLL に対する応答 STAT により失われたパケットの通番が通知され、選択再送が行われる。SSCOP はシンプルなメカニズムで効率良く SR ARQ を実現するため [69]、TCP に代わる衛星通信用のエンドツーエンドプロトコルとしても提案されている

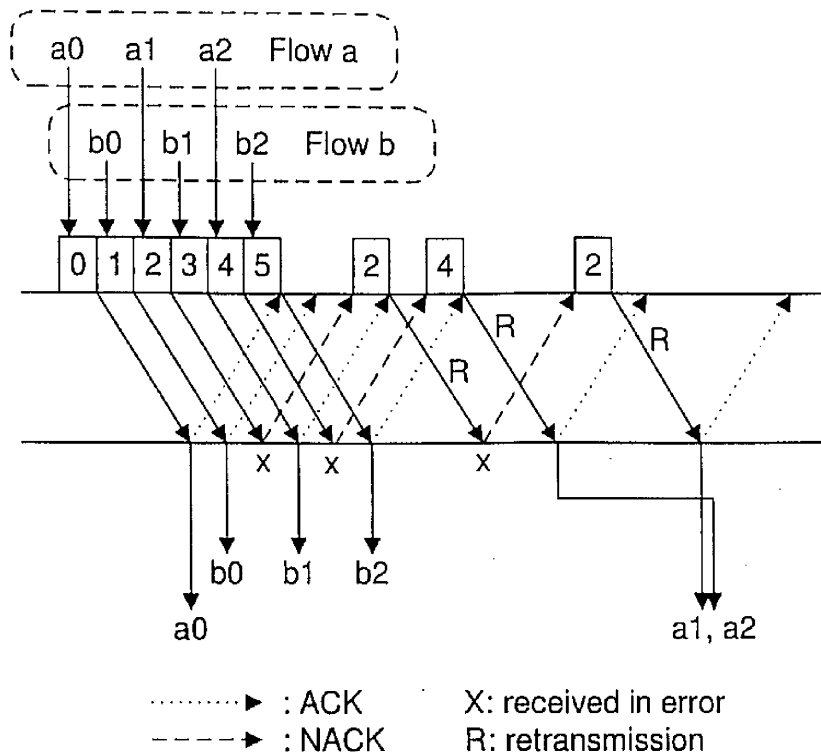


図 4.3 PFRS 方式のシーケンス例

[64][68][67]。

表 4.1 にシミュレーション条件を示す。遅延時間やスループットなどは、乱数の初期値を変えた試行を 10 回行い平均値を求めた。パケット遅延時間と受信バッファ保留に関しては UDP で評価を行った。また実際のアプリケーションでは TCP が用いられる場合が多いので、エンドツーエンドで TCP により通信が行われ、複数の TCP コネクションが衛星回線上に設定される場合を想定した評価も併せて行った。比較のため以下に示す 4 方式をシミュレーションしている。

- 方式 A PFRS 方式
- 方式 B 全フロー ARQ 方式
- 方式 C 個別フロー ARQ 方式
- 方式 D 無レイヤ 2ARQ 方式

方式 D はレイヤ 2 で SR ARQ を用いず、TCP によりエンドツーエンドでのみ再送を行う。想定するネットワークの構成を図 4.5 に示す。衛星回線においてレイヤ 2 として SR ARQ による再送が行われ、この上に UDP または TCP による複数フローの通信が行わ

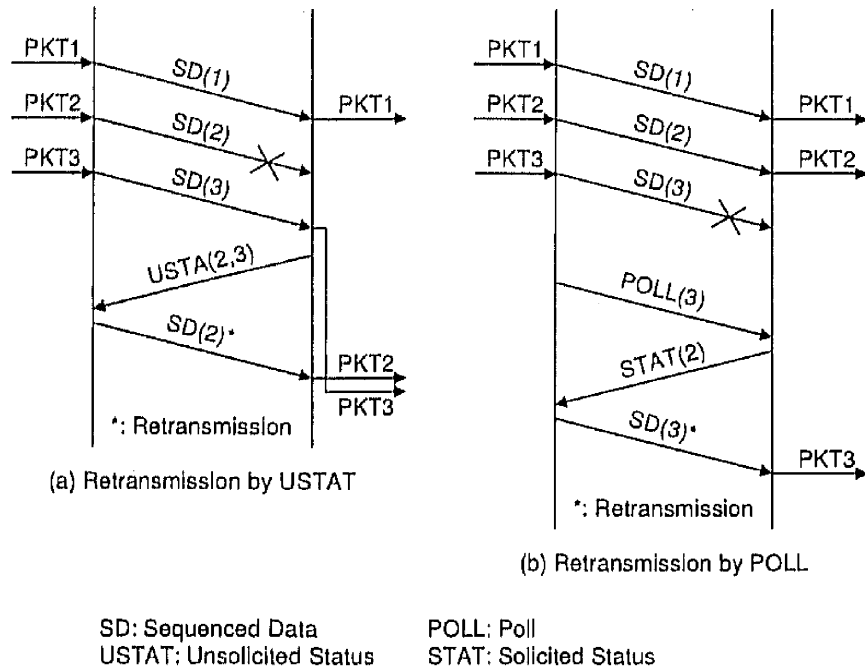


図 4.4 SSCOP のシーケンス例

れる。

PFRS 方式の実現には方式により多少のオーバーヘッドが必要となる場合があるが、シミュレーションを行った UDP 及び TCP のパケット長に比べて小さいと考えられるので、オーバーヘッドは考慮していない。

4.5 シミュレーション結果

4.5.1 UDP の場合

UDP の場合のパケット平均遅延時間の結果を表 4.2 に示す。方式 D は TCP を前提としているので比較から除外している。また、平均遅延時間は衛星回線による伝搬遅延時間を差し引いた値を示している。ビット誤り率が大きくなるとともに再送と順序制御により平均遅延時間が大きくなるのが分かる。

フロー数が 1 の場合は方式 A、B、C で結果が同じとなるが、フロー数が 2 以上では全フロー ARQ 方式 (方式 B) の平均遅延時間が最も大きい。PFRS 方式 (方式 A) と個別フロー ARQ 方式 (方式 C) の比較ではフロー数が増えると、PFRS 方式 (方式 A) の遅延時間が小さくなる。PFRS 方式により全フロー ARQ 方式に比べ順序制御による平均遅延時間が改善されていること、また PFRS 方式ではフロー数が増えても平均遅延時間が

表 4.1 シミュレーション条件

衛星回線速度	6.3 Mbit/s
衛星回線の伝搬遅延時間	0.25 sec
衛星回線ビット誤り率	10^{-8} , 10^{-7} , 10^{-6} , 10^{-5}
誤りの発生	ランダム
SR ARQ プロトコル	SSCOP
SSCOP のポーリング時間間隔	0.2 sec
UDP パケット長	1000 Byte
1 本の UDP フロー速度	500 kbit/s
複数 UDP フロー間の初期位相	ランダム
UDP フロー数	1, 2, 5, 10
TCP タイプ	NewReno
TCP パケット長	1000 Byte
TCP ウィンドウサイズ	50
TCP コネクション数	1, 2, 5, 10
シミュレーション時間	60 sec
シミュレーション試行回数	10 回

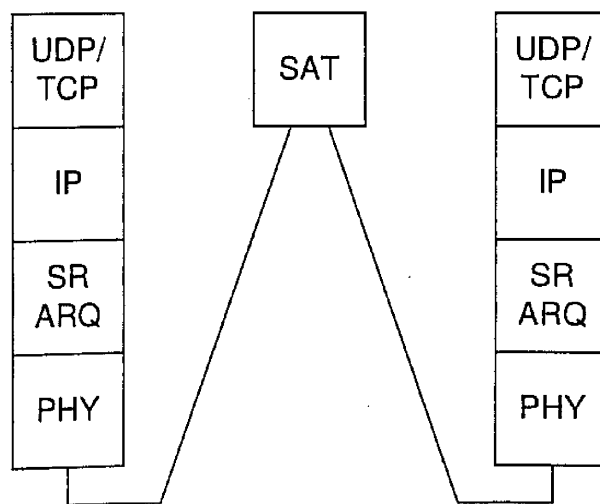


図 4.5 シミュレーションで評価する衛星通信システムの構成

増加せず、フロー数による影響をほとんど受けないことが分かる。逆に表 4.2 から PFRS 方式 (方式 A) において、わずかではあるがフロー数が増えると平均遅延時間が減少する傾向が読みとれる。UDP では同じフローの packets が等間隔で送信されるが、PFRS 方式では UDP のフロー数が多いほど、伝送誤りによる packets 喪失を次の packets の正常受信により検出するまでの時間が平均的に短くなり、再送が早く行われるので平均遅延時間が短縮されるものと考えられる。個別フロー ARQ 方式では、このような効果を期待できないので、フロー数が多くなると平均遅延時間が大きくなる傾向がある。このように PFRS 方式ではできるだけ多くのフローを多重化した方が、packets 喪失の検出が早く行われ性能がよくなる。

ビット誤り率が 10^{-7} と 10^{-6} の場合について方式 A、B、C による UDP packets 遅延時間の相対頻度分布を図 4.6 と図 4.7 に示す。ここで、packets の遅延時間は伝搬遅延時間 (0.25 sec) を含んでいる。頻度は時間幅 0.01 sec を単位として計測した。これらの図から PFRS 方式 (方式 A) 及び個別フロー ARQ 方式 (方式 C) に比べ、全フロー ARQ 方式 (方式 B) が大きな遅延時間をとる相対頻度の大きいことが分かる。図 4.6 では 1 回、図 4.7 では 2 回の再送が発生しており、図 4.7 において遅延時間分布が段階的に変化するのは再送回数の違いによるものである。また同じ再送回数では遅延時間の大きな値で頻度が増加する傾向が見られる。これは再送された packets の後続 packets で再送 packets に近いものほど順序制御による遅延を受けやすく、かつその遅延時間が再送による遅延時間に近くなることによる。

表 4.3 は順序制御のために受信側で保留される平均バッファ数を示している。全フロー ARQ 方式 (方式 B) の平均保留バッファ数が最も大きく、PFRS 方式 (方式 A) と個別フロー ARQ 方式 (方式 C) では差は少ないものの PFRS 方式 (方式 A) の平均保留バッファ数が最も小さい。ビット誤り率が小さい場合、全フロー ARQ 方式 (方式 B) の平均保留バッファ数は PFRS 方式 (方式 A) のほぼフロー数倍となっている。この差はビット誤り率が大きくなると減少するが、ビット誤り率が 10^{-5} でもフロー数が 5 以上では約 2 倍の差となっている。ビット誤り率が大きくなると差が小さくなる理由は、PFRS 方式において複数フローで同時に順序制御が行われる確率が高くなるためと考えられる。

以上から PFRS 方式は個別フロー ARQ 方式及び従来の全フロー ARQ 方式に比べ、packets の平均遅延時間が減少し受信側でのバッファ数も少なくすむことが分かる。

4.5.2 TCP コネクション数を変化させた場合のスループット

図 4.8 から図 4.11 は TCP コネクションの数を 1、2、5、10 と変化させた場合のビット誤り率と回線速度で規格化した全体の平均スループットを示している。図 4.8 の TCP

表 4.2 衛星回線での伝搬遅延時間を含まない平均パケット遅延時間

ビット誤り率	UDP フロー数	平均	Scheme A	Scheme B	Scheme C
		標準偏差			
10^{-8}	1	average	0.0037	0.0037	0.0037
		std	0.0106	0.0106	0.0106
	2	average	0.0036	0.0045	0.0033
		std	0.0141	0.0196	0.0096
	5	average	0.0034	0.0063	0.0038
		std	0.0144	0.0312	0.0155
	10	average	0.0033	0.0092	0.0046
		std	0.0146	0.0446	0.0157
10^{-7}	1	average	0.0103	0.0103	0.0103
		std	0.0518	0.0518	0.0518
	2	average	0.0106	0.0182	0.0115
		std	0.0522	0.0718	0.0559
	5	average	0.0097	0.0357	0.0110
		std	0.0488	0.1012	0.0537
	10	average	0.0094	0.0628	0.0115
		std	0.0480	0.1319	0.0527
10^{-6}	1	average	0.0731	0.0731	0.0731
		std	0.1482	0.1482	0.1482
	2	average	0.0722	0.1258	0.0746
		std	0.1454	0.1778	0.1513
	5	average	0.0669	0.2340	0.0741
		std	0.1389	0.1906	0.1519
	10	average	0.0661	0.3396	0.0763
		std	0.1398	0.1791	0.1540
10^{-5}	1	average	0.4986	0.4986	0.4986
		std	0.3046	0.3046	0.3046
	2	average	0.4734	0.6302	0.4889
		std	0.3026	0.3080	0.2972
	5	average	0.4694	0.8516	0.5048
		std	0.3129	0.3529	0.3105
	10	average	0.4950	1.0787	0.5014
		std	0.3257	0.3703	0.3090

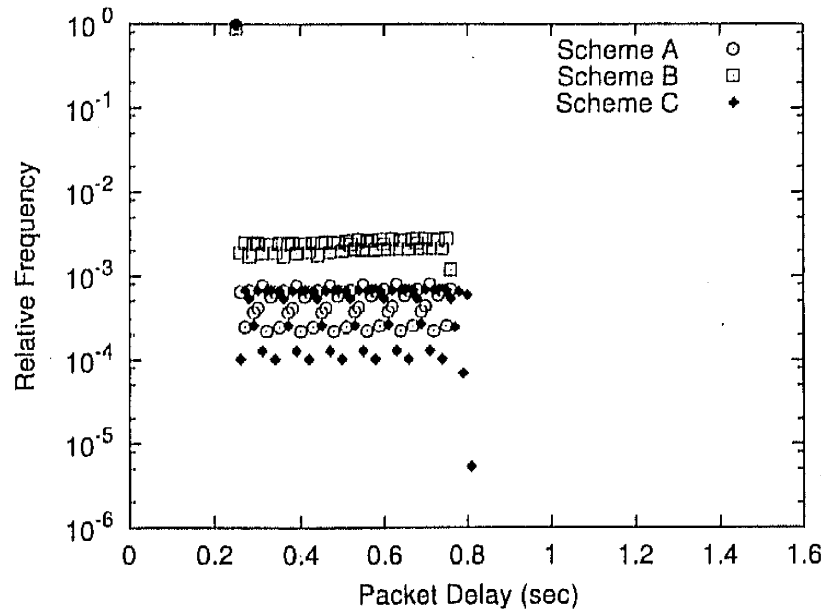


図 4.6 UDP パケット遅延時間の相対頻度分布、ビット誤り率 = 10^{-7} 、UDP フロー数 = 5

の接続数が 1 の場合は、PFRS 方式 (方式 A) と全フロー ARQ 方式 (方式 B) 及び個別フロー ARQ 方式 (方式 C) の動作は完全に同じとなるので、シミュレーション結果も完全に一致している。ビット誤り率が大きくなるとリンクバイリンクに再送を行う方式 A、B、C の性能が、TCP のみによりエンドツーエンドに再送を行う方式 D よりもよくなる。図 4.9 で TCP の接続数が 2 になると、PFRS 方式 (方式 A) 及び個別フロー ARQ 方式 (方式 C) の性能が全フロー ARQ 方式 (方式 B) よりもよくなり、TCP の接続数が 5、10 と増えるに従って PFRS 方式 (方式 A) と個別フロー ARQ 方式 (方式 C) の優位性が際立っている。これら方式の優位性は順序制御による遅延時間が少なくなる効果によるものと考えられる。UDP では差があったが、TCP では PFRS 方式 (方式 A) と個別フロー ARQ 方式 (方式 C) の性能はほぼ同じ結果となっている。これは、TCP の場合と同じフローに属するパケットが連続して送信されることが多く、個別フロー ARQ 方式 (方式 C) においても PFRS 方式 (方式 A) と同様に正しく受信したパケットの通番の飛びによるパケット喪失の検出が早く行われることによる。

方式 D の性能が劣る原因は、伝送誤りにより失われたパケットを TCP により高速再送するが [103]、この際にウィンドウサイズを減少させるためである [102]。

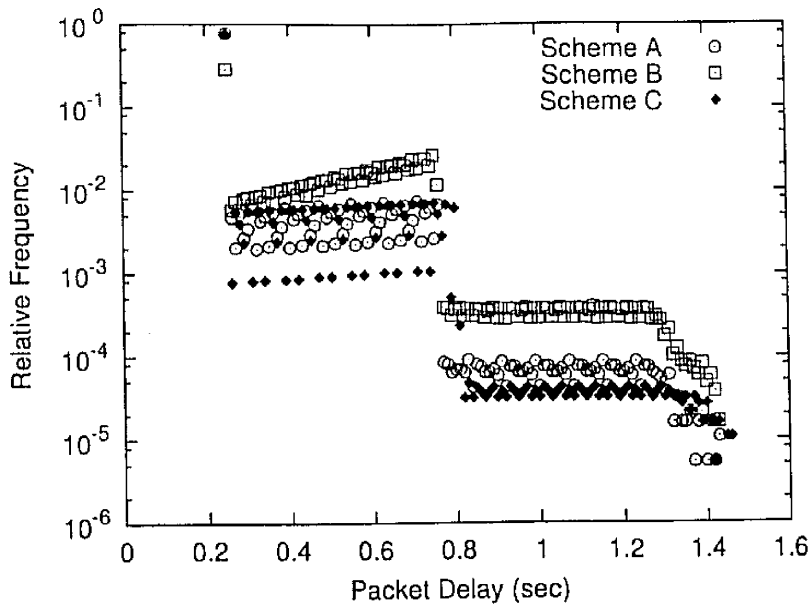


図 4.7 UDP パケット遅延時間の相対頻度分布、ビット誤り率 = 10^{-6} 、UDP フロー数 = 5

4.5.3 TCP の場合のバースト長分布

図 4.12 から図 4.15 は TCP コネクションの数を 5 に固定して、ビット誤り率をそれぞれ 10^{-8} 、 10^{-7} 、 10^{-6} 、 10^{-5} とした場合の SR ARQ の出力におけるバースト長の相対頻度分布を示している。ここでバースト長の定義は SR ARQ が同時に上位層に転送するパケット数である。順序制御により保留されるパケット数は、TCP のウィンドウ制御により制限される。全フロー ARQ 方式 (方式 B) の場合、全 TCP コネクションのウィンドウサイズの総和分が保留され、最大 250 パケットが一気に解放される可能性がある。このため、これらの図で全フロー ARQ 方式でのバースト長の最大値は 250 となっている。上位層へのパケット転送は再送されたパケットを受信した時点で発生するが、PFRS 方式 (方式 A) では、同時に転送されるパケットは再送パケットと同じ 1 つの TCP コネクションに属するパケットに限られる。1 本のコネクションにおいて、最大で TCP のウィンドウサイズ分のパケットが保留されるので、図 4.13 と図 4.14 で PFRS 方式のバースト長の最大値が 50 となっている。個別フロー ARQ 方式 (方式 C) においても順序制御がフローごとに行われるので、バースト長の最大値は同様に 50 となる。ビット誤り率が小さい場合、バーストの発生頻度は小さいものの、上記の最大サイズ程度のバーストが発生しやすいことが分かる。ビット誤り率が大きくなるとバーストの頻度は大きくなるものの断片化される傾向が見られる。2.4.1 で述べたように、ビット誤り率が大きい場合、確率分布は

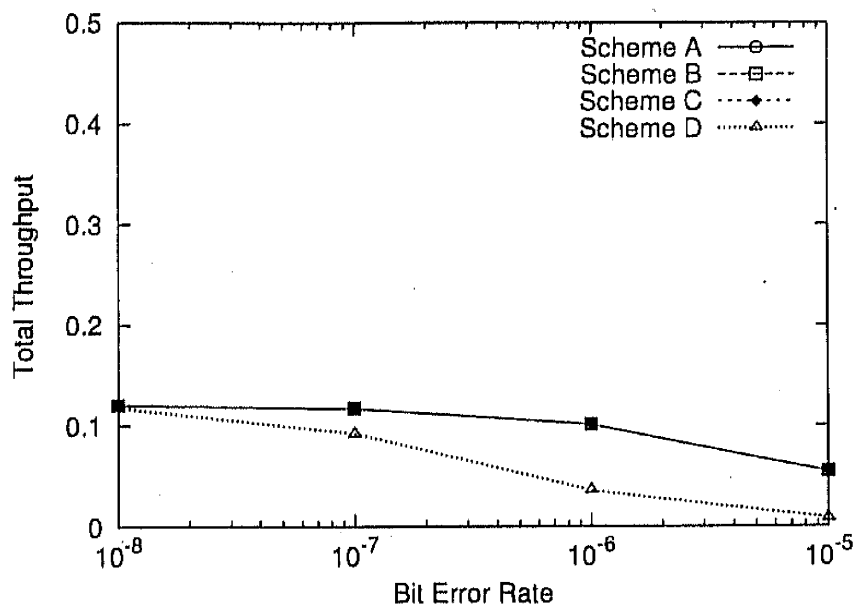


図 4.8 TCP スループット、TCP コネクション数 = 1

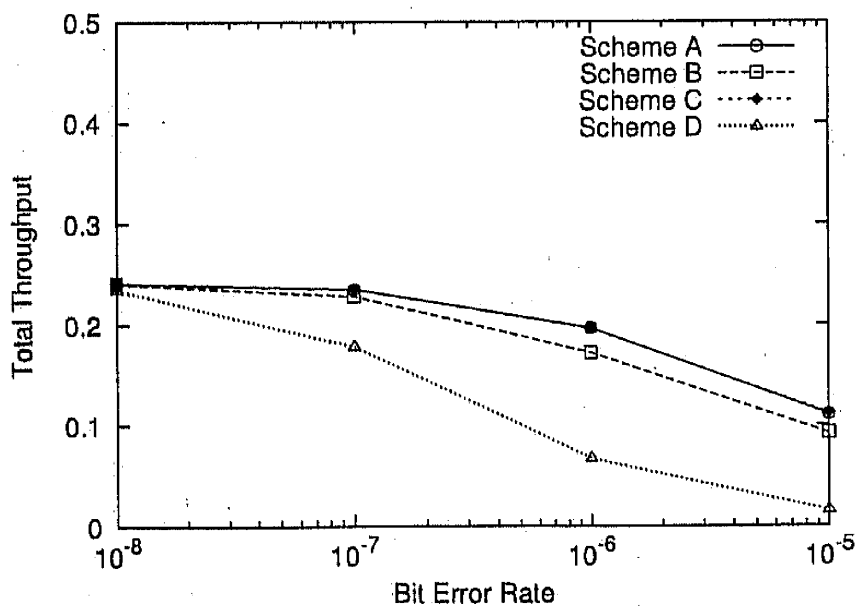


図 4.9 TCP スループット、TCP コネクション数 = 2

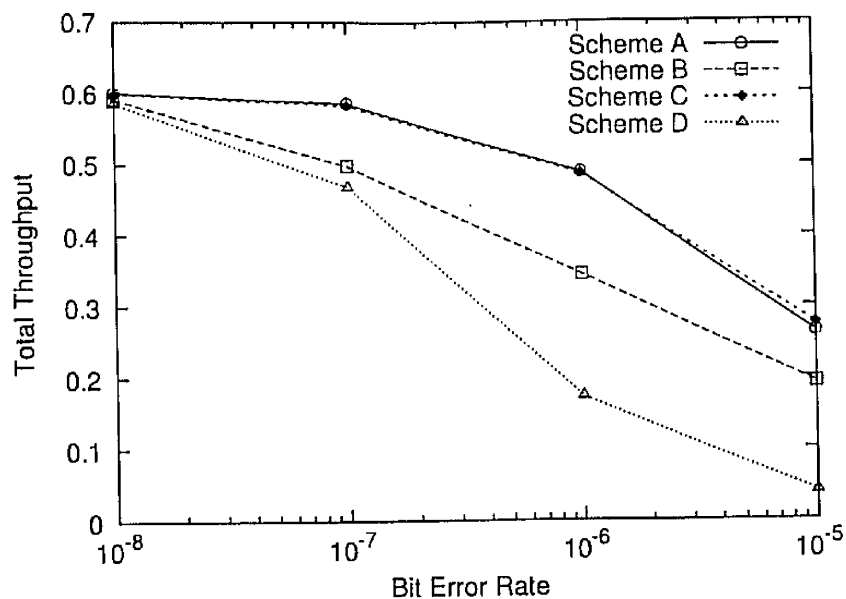


図 4.10 TCP スループット、TCP コネクション数 = 5

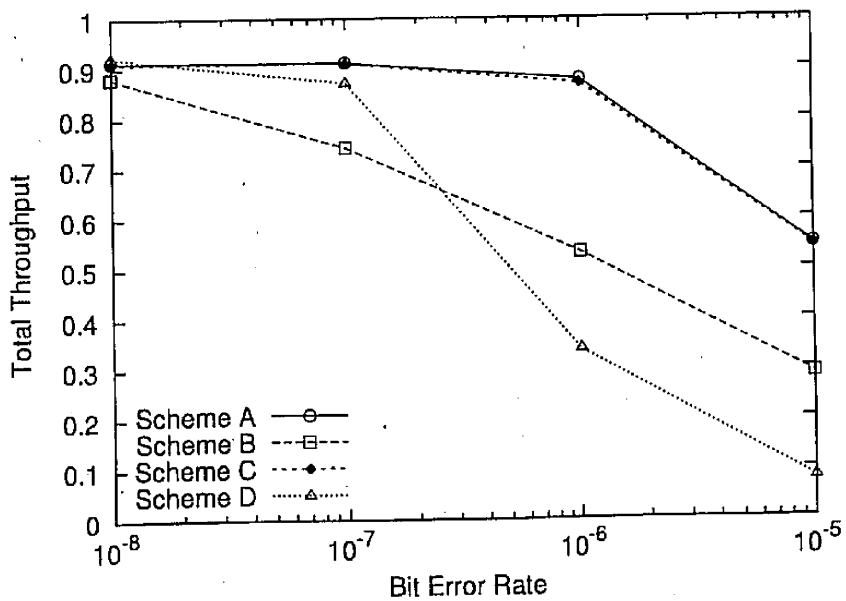


図 4.11 TCP スループット、TCP コネクション数 = 10

表 4.3 順序制御による平均保留バッファ数

ビット誤り率	UDP フロー数	Scheme A	Scheme B	Scheme C
10^{-8}	1	0.0592	0.0592	0.0592
	2	0.1089	0.2210	0.0727
	5	0.2195	1.1074	0.2455
	10	0.3925	3.9722	0.4500
10^{-7}	1	0.4431	0.4431	0.4431
	2	0.9238	1.8525	1.0054
	5	2.0135	9.9022	2.3097
	10	3.8583	36.2338	4.3829
10^{-6}	1	4.0066	4.0066	4.0066
	2	7.8927	14.3831	8.1728
	5	18.1843	68.6793	20.2109
	10	35.9222	200.9997	41.1490
10^{-5}	1	27.1023	27.1023	27.1023
	2	51.3499	70.1287	52.9048
	5	127.3409	241.6816	136.7611
	10	264.4769	613.7893	270.9591

片対数表示で傾きが $\log_{10}(1 - \epsilon)$ の直線になることを説明したが、図 4.15 はこれに近い結果となっている。

以上から全フロー ARQ 方式 (方式 B) ではバースト長の最大値がコネクション数に依存して大きくなるが、PFRS 方式 (方式 A) 及び個別フロー ARQ 方式 (方式 C) では 1 本の TCP コネクションのウィンドウサイズにバースト長が抑えられ、バースト的なパケット出力が改善されることが分かる。なお、図 4.13 で全フロー ARQ 方式 (方式 B) においてバースト長が全 TCP ウィンドウサイズの和である 250 を超える場合、及び図 4.14 で PFRS 方式 (方式 A) 及び個別フロー ARQ 方式 (方式 C) においてバースト長が TCP コネクション 1 本のウィンドウサイズの 50 を超える場合がわずかに見られる。これは再送及び順序制御による遅延の変動が大きいことにより、TCP でタイムアウトが発生して無効な再送が行われ、パケットの重複が発生しているためと考えられる。

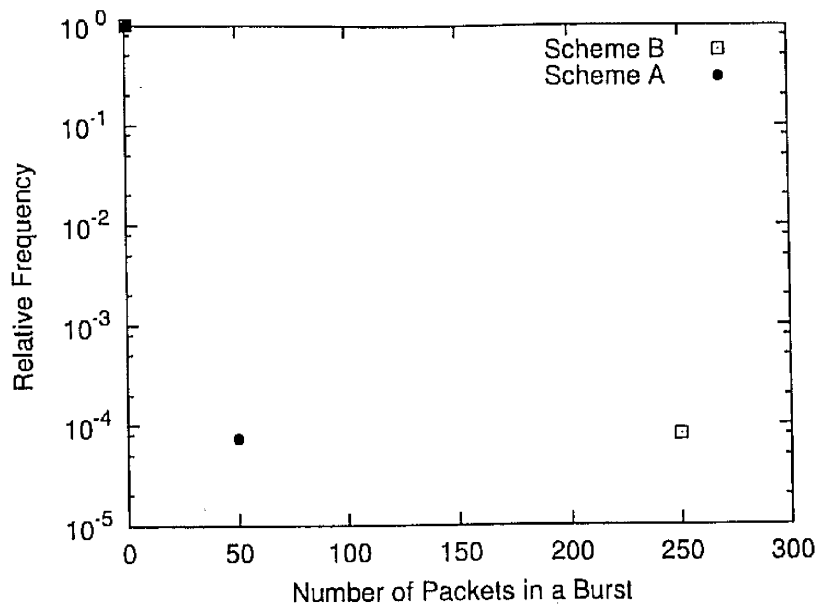


図 4.12 方式 A、方式 B、方式 C のバースト長分布、ビット誤り率 = 10⁻⁸

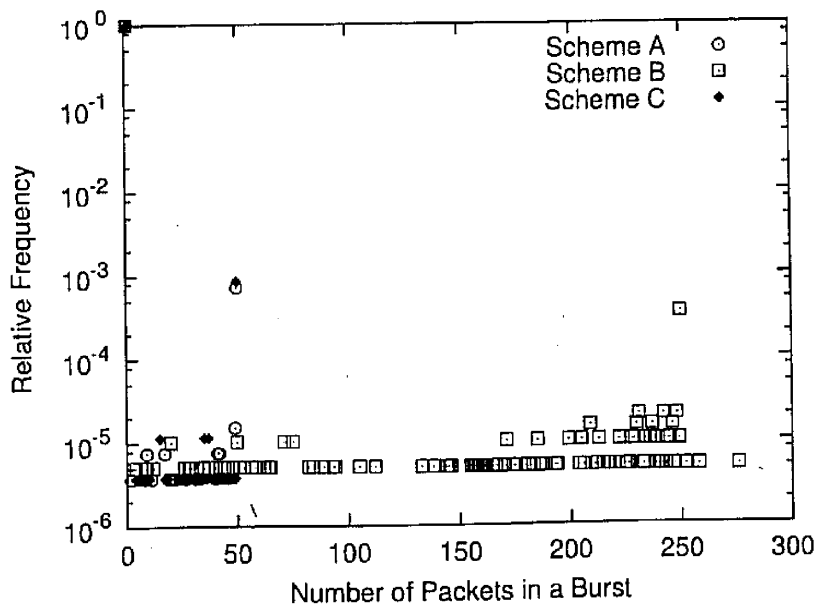


図 4.13 方式 A、方式 B、方式 C のバースト長分布、ビット誤り率 = 10⁻⁷

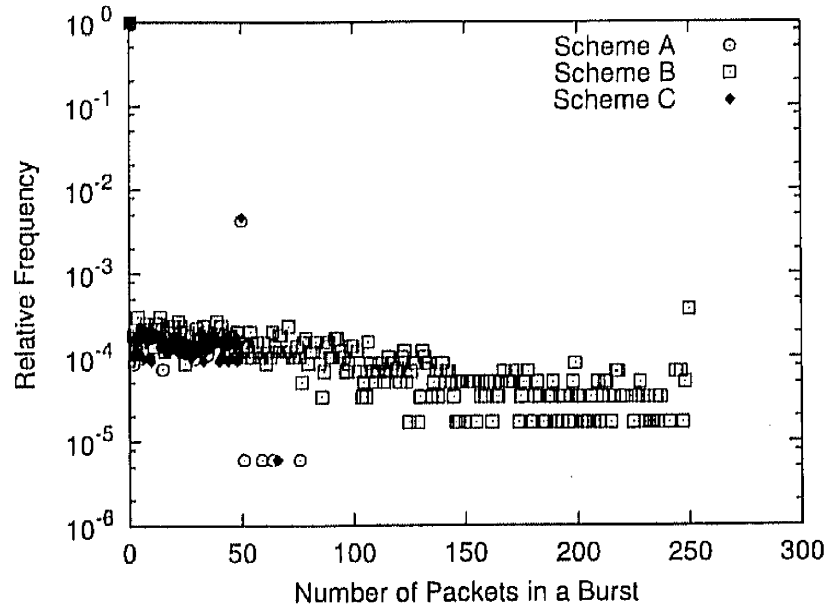


図 4.14 方式 A、方式 B、方式 C のバースト長分布、ビット誤り率 = 10^{-6}

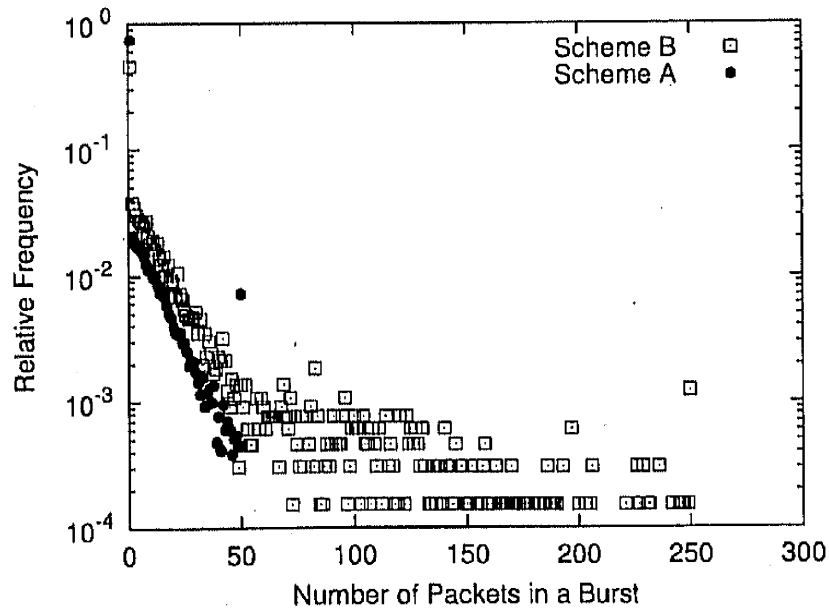


図 4.15 方式 A、方式 B、方式 C のバースト長分布、ビット誤り率 = 10^{-5}

4.5.4 バッファ数に制限のあるルータで中継する場合の性能

SR ARQ が出力するバースト的なパケットの出力は、これを IP ネットワークで中継する場合にルータでのバッファあふれや QoS の劣化などの悪影響を及ぼす可能性がある。以下では、シミュレーションによりルータのバッファ数が制限されている状況での性能を評価する。想定するネットワークの構成を図 4.16 に示す。SR ARQ の出力が LAN を介してルータに入力され、ルータから地上回線を経由して宛先端末に転送される構成である。地上回線の速度を衛星回線と同じ 6.3 Mbit/s、TCP のコネクション数を 5 としている。SR ARQ で発生したバースト的なパケットは LAN により 100 Mbit/s の速度で高速転送され、ルータで地上回線との速度の整合をとるためバッファリングが行われる。ここでルータのバッファを無制限とすると、図 4.10 と同程度の性能となる。図 4.17 はルータのバッファ数を 250 パケット、図 4.18 はバッファ数を 50 パケットに制限した場合のスループットを示している。バッファ数を超えた大きさのバーストが到着するとパケットの廃棄が発生し、性能の低下を引き起こすことになる。全フロー ARQ 方式 (方式 B) ではバースト長の最大値が 250 となるため、バッファ数が 250 の場合に比べバッファ数を 50 とした場合に性能が著しく低下する。方式 A 及び方式 C ではバースト長の最大が 50 程度に抑えられるため、バッファ数を 50 とした場合でも性能の低下は少ない。衛星回線で SR ARQ を用いない方式 D ではバーストがまったく発生しないので、パケットの廃棄は発生せず、バッファ数が 250 と 50 で同じ性能となる。図からルータのバッファ数が 50 で誤り率が小さい場合には、全フロー ARQ 方式 (方式 B) の性能が一番悪くなることが分かる。このシミュレーションでは TCP のコネクション数を 5 としたので全フロー ARQ 方式のバーストの最大値は 250 程度であったが、TCP のコネクション数を増やすとこれに比例してバースト長の最大値が増加する。衛星回線上で通信する TCP のコネクション数は予め最大数を規定することが難しく、ルータでの所要バッファ数を予測することは困難である。PFRS 方式及び個別フロー ARQ 方式では、TCP のコネクション数に関係なくバースト長の最大値が 50 程度に抑えられる。このような 1 本の TCP のウィンドウサイズ分のバーストは、LAN に直接接続された端末による通信でも TCP により発生する可能性があり、実用上許容可能な範囲と考えられる。

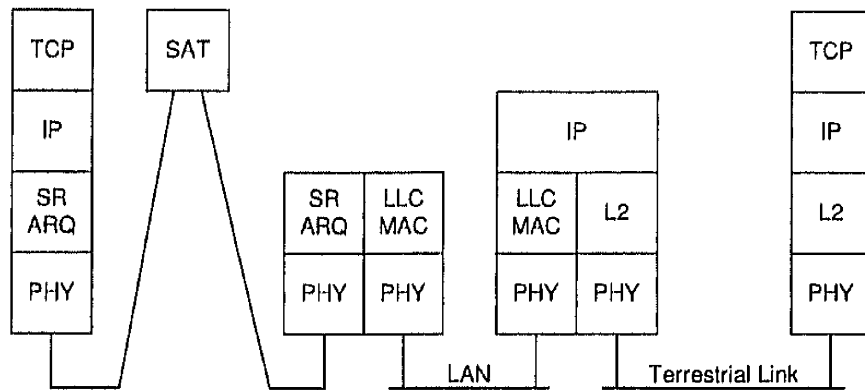


図 4.16 シミュレーションで評価する衛星通信システムの構成

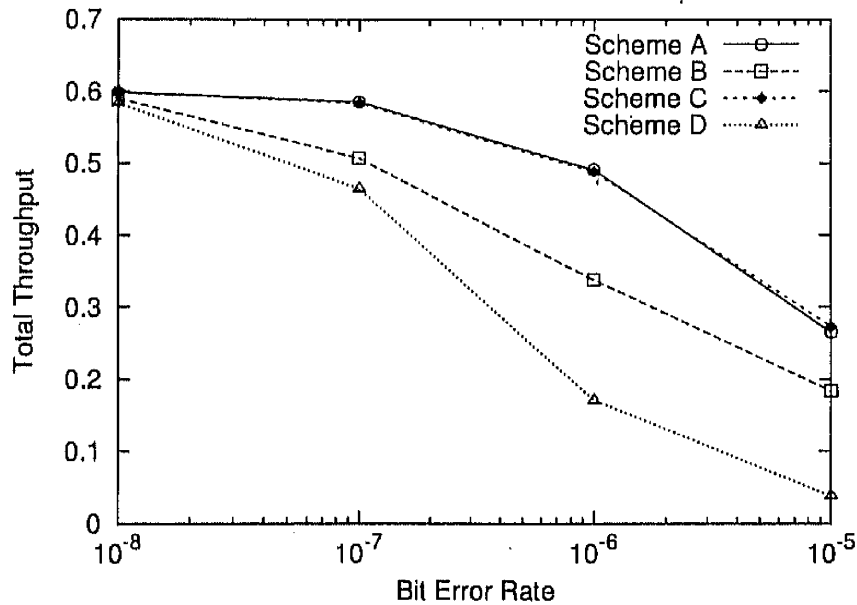


図 4.17 TCP スループット、ルータのバッファ数 = 250

4.6 解析結果と実際の SR ARQ プロトコルを用いたシミュレーションとの比較

3章において順序制御による遅延時間を解析し、同じ条件で行ったシミュレーション結果と比較して解析が正しいことを確認した。解析モデルは実際に使用される SR ARQ プロトコルの動作と完全に一致するものではない。この節では、SR ARQ を実現する実

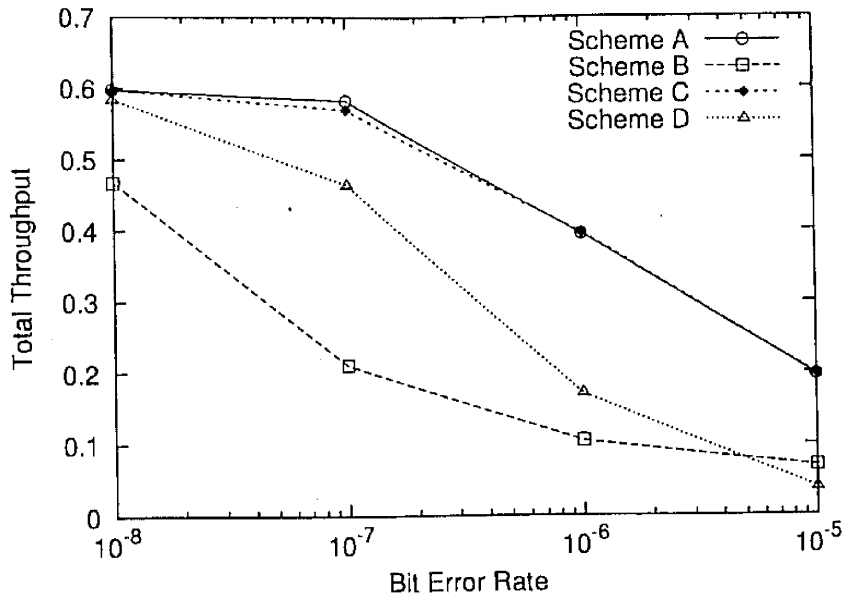


図 4.18 TCP スループット、ルータのバッファ数 = 50

際のプロトコル及び上位層に UDP や TCP を用いた場合のシミュレーション結果と解析結果との比較を行い、3章で述べた解析手法の妥当性について検討する。また解析より TCP における無効なタイムアウトの発生確率を計算し、PFRS 方式と全フロー ARQ 方式を無効なタイムアウトの発生点から比較する。想定するネットワークの構成は図 4.5 と同じである。シミュレーション条件も表 4.1 とデータ長以外は同じである。UDP 及び TCP のデータ長は 1000 Byte としているが、これ以外に UDP の場合は UDP ヘッダや IP ヘッダ更に SSCOP のオーバーヘッドとして全部で 38 Byte を想定し、衛星回線上でのパケット長を 1038 Byte としている。TCP の場合も同様のオーバーヘッドとして 50 Byte を想定し、衛星回線上でのパケット長を 1050 Byte としている。TCP のスループットはオーバーヘッドを除いたデータ分のみで計測している。TCP のバージョンは一般的に用いられている NewReno を用い、特に衛星回線を意識した拡張オプション [75] を採用せずウィンドウサイズは 50 としている。UDP の場合のトラフィック発生は 500 kbit/s の一定速度で、5 本のフローが存在するものとしている。TCP の場合のトラフィックの発生は、送信すべきデータが常に存在する条件とし、実際のパケット送信はウィンドウサイズにより制御される。また、SSCOP においては最大再送回数の制限は設けていない。スループットなどは乱数の初期値を変えた試行を 10 回行い平均値及び信頼度区間を求めた。

4.6.1 UDP を用いた場合の遅延時間分布

表 4.4 は、全フロー ARQ 方式の場合の UDP パケットの遅延時間についてシミュレーションによる平均と標準偏差を解析から求めた値と比較している。回線上での UDP のパケット長を 1038 Byte としたので、解析ではフレームのスロット数 M を 380 としている。シミュレーション結果については 95% 信頼度区間も併せて示している。

表 4.5 は同じ比較を PFRS 方式について行った結果である。解析結果は再送と順序制御による遅延時間、シミュレーション結果はエンドツーエンドの遅延時間から伝搬遅延時間と 1 パケットの送信に要する時間を差し引いた値を示している。ビット誤り率が 10^{-5} と大きい場合に、シミュレーションによる平均値と標準偏差が解析より大きくなるが、それ以外のビット誤り率ではシミュレーションと解析の両方の結果がよく一致している。ビット誤り率が大きい場合、1 パケットにおいて 2 回以上再送が行われる場合が増えてくる。後で説明するように 2 回以上の再送では再送に要する時間が実際のプロトコルでは解析モデルより大きくなる。ビット誤り率が大きい場合に結果が一致しない原因はこれによるものと考えられる。

図 4.19 と図 4.20 はビット誤り率が 10^{-7} ($\epsilon = 0.0008256$) と 10^{-6} ($\epsilon = 0.008256$) の場合のパケットの遅延時間分布を示している。この遅延時間は、伝搬遅延時間と再送及び順序制御による遅延時間を含んでいる。これらの図で全フロー ARQ 方式と PFRS 方式について、シミュレーションと計算の両方の結果を示している。ns-2 によるシミュレーション結果は、0.01 sec の時間幅を単位とする相対頻度分布で、10 回の試行で得られたデータすべてを用いて頻度分布を求めている。これと比較するため計算結果も同じ時間幅を単位とする確率分布に変換している。ビット誤り率が 10^{-7} の場合、シミュレーションでは最大 1 回の再送でパケットが送信されており、計算結果とシミュレーションによる遅延時間分布はよく一致している。シミュレーションでは 2 回の再送を行う場合のデータがない結果となっているが、これは 2 回の再送が発生する確率が小さいためと考えられる。ビット誤り率が 10^{-6} の場合、シミュレーションは概ね最大 2 回の再送でパケットが送信されているが、1 回の再送による遅延時間の範囲では分布はよく一致している。2 回の再送による遅延時間分布は、シミュレーション結果の方が確率が若干小さく、分布が遅延時間の大きい方向に広がっている。これは解析したモデルと実際のプロトコルの動作との差異によるものである。SR ARQ の解析モデルでは、パケットで受信誤りが発生すると直ちに NACK が返送され、NACK には伝送誤りが発生せず、NACK が送信側に返ると直ちに再送が行われるとしている。このため複数回の再送であってもラウンドトリップ時間の整数倍の時間で再送が行われる。SSCOP ではパケットに受信誤りが発生すると、次に

表 4.4 全フロー ARQ 方式の遅延時間の平均と標準偏差

Bit error rate	Average [sec]			Standard deviation [sec]		
	Simulation	95% conf.	Analysis	Simulation	95% conf.	Analysis
10^{-5}	0.849081	0.016718	0.734385	0.352933	0.031989	0.266671
10^{-6}	0.231380	0.008465	0.227552	0.190579	0.005751	0.188118
10^{-7}	0.033118	0.005064	0.031513	0.101189	0.006899	0.098559
10^{-8}	0.003736	0.001758	0.003269	0.031188	0.012595	0.032954

表 4.5 PFRS 方式の遅延時間の平均と標準偏差

Bit error rate	Average [sec]			Standard deviation [sec]		
	Simulation	95% conf.	Analysis	Simulation	95% conf.	Analysis
10^{-5}	0.466763	0.007874	0.431763	0.312852	0.011798	0.260143
10^{-6}	0.064335	0.002991	0.064572	0.138888	0.003984	0.138601
10^{-7}	0.007105	0.001240	0.006850	0.048778	0.004119	0.048115
10^{-8}	0.000802	0.000370	0.000689	0.014366	0.005816	0.015363

正しくパケットを受信した時点で NACK が送信され、また NACK 自体が誤る場合もある。また最初の再送に失敗すると 2 回目以降は、送信側からのポーリングにより再送失敗が検出され再再送が行われるため、再送に時間を要するとともにその時間のバラツキが大きくなる。誤り率が 1 回の再送程度の条件であれば、本論文のモデルは実際のプロトコルに近い結果が得られるといえる。また、解析では複数フローのパケットがランダムに発生するとしたが、シミュレーションの条件では同じフローのパケットが 5 パケットごとに周期的に発生する。図 4.19 と図 4.20 から、全フロー順序制御方式と比較した PFRS 方式による遅延時間削減の効果について、計算結果とシミュレーション結果は傾向がよく一致しており、複数フローのパケットがランダムに発生する仮定が概ね妥当であることを示している。以上から、UDP パケットの遅延時間ゆらぎを評価する上で本論文の解析は有用と考えられる。

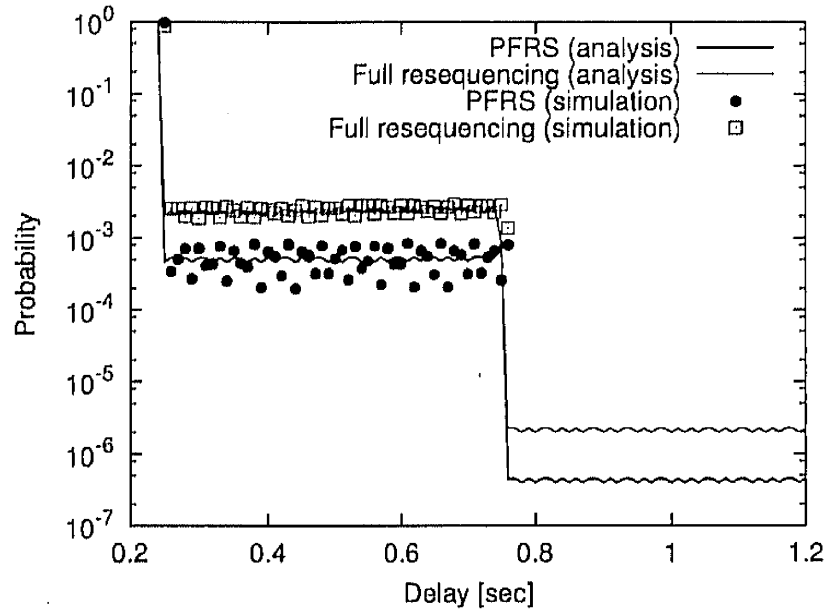


図 4.19 UDP のフロー数を 5 とした場合の遅延時間分布 ($M = 376, N_r = 8, \varepsilon = 0.0008256$)

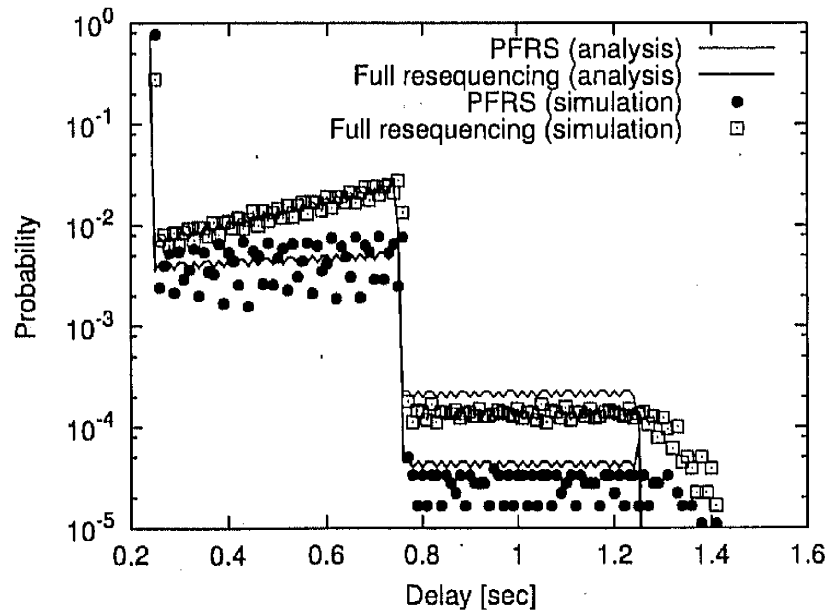


図 4.20 UDP のフロー数を 5 とした場合の遅延時間分布 ($M = 376, N_r = 8, \varepsilon = 0.008256$)

4.6.2 解析により求めた TCP のスループット

レイヤ 2 の SR ARQ における再送と順序制御による遅延時間及びその変動は上位層に影響を与え、特に上位層で TCP が用いられる場合は遅延時間によりスループットが低下する。ここでは解析で得られた遅延時間分布からパケット誤り率と TCP のスループットとの関係を計算し、ns-2 によるシミュレーション結果と比較する。

衛星回線上での TCP のパケット長を 1050 Byte としたので、この条件から解析におけるフレームのスロット数 M を 376 とした。複数の TCP コネクションの条件が等しいので、各コネクションのパケットが出現する確率は等しいものとし、TCP のコネクション数を N_{TCP} 、各 TCP のコネクションにおける TCP のウィンドウサイズを W とする。再送と順序制御による遅延時間の確率分布より計算される平均遅延時間を $D(\alpha, \beta)$ スロットとする。

$$D(\alpha, \beta) \triangleq \sum_{u=0}^{N_r} \sum_{i=0}^{M-1} (uM + i) P_{delay}(u, i) \quad (4.1)$$

回線の帯域がボトルネックとはならない条件を想定し、送信側での送信待ち行列遅延が無視できるものと仮定する。TCP のセグメント（以下パケットの用語を用いる）を送信してから応答が返るまでのラウンドトリップ時間は、1 フレーム時間と再送及び順序制御による遅延時間の和から成り、 $M + D(\alpha, \beta)$ スロットとなる。1 本の TCP コネクションごとに 1 ラウンドトリップ時間に W パケットが送信されるので、回線速度で規格化されたスループットを S とすると、以下のようになる。

$$S = \frac{N_{TCP} W}{M + D(\alpha, \beta)} \quad (4.2)$$

一方、(3.7) 式から α と S の間には次の関係が成り立つ。

$$\alpha = \frac{S(1 - \epsilon)}{(1 - \epsilon)(1 - \epsilon^{N_r}) - S(\epsilon - \epsilon^{N_r+1})} \quad (4.3)$$

β を与えると、式 (4.2) と (4.3) から反復計算により α とスループット S を求めることができる。 $\beta = 1.0$ とすると全フロー ARQ 方式の場合のスループットを計算できる。またコネクション数 N_{TCP} が 5 の場合は、各 TCP のコネクションは同じ条件なので $\beta = 0.2$ とすると PFRS 方式のスループットを求めることができる。図 4.21 は TCP のコネクション数 N_{TCP} を 1、2、5 とした場合の回線速度で規格化されたスループットを示している。ここで全フロー ARQ 方式と PFRS 方式とを比較を示しており、この計算で最大再送回数は 8 とし、最大再送回数によるパケット損失は無視できるものとしている。ま

た、50 Byte 分のオーバーヘッドを除くため計算で得られた S に $0.9523 (= 1000/1050)$ を乗じて補正している。ns-2 によるシミュレーション結果では、95% 信頼度区間も表示しているが、プロットした点程度の区間幅となっている。

図 4.21 から、誤り率が大きくなると PFRS 方式により全フロー ARQ 方式に比べスループットが向上すること、また、コネクションの数が大きい場合 ($N_{TCP} = 5$) に、スループット向上の効果が顕著となることが分かる。ns-2 によるシミュレーション結果は、計算により求めたスループットより同じパケット誤り率では小さくなっているが、全フロー ARQ 方式と PFRS 方式のスループットの差については概ね同じ傾向となっており、順序制御による遅延時間がスループットに与える影響の大きいことを裏付けている。

図 4.22 は、シミュレーション結果における再送と順序制御による平均遅延時間及び平均送信待ち時間を示している。図で左側 3 列は全フロー ARQ 方式の場合で、左から計算により得られた再送と順序制御による平均遅延時間、シミュレーションにより得られた再送と順序制御による平均遅延時間、シミュレーションにより得られた平均送信待ち時間を示している。図で FR は Full Resequencing の略で全フロー ARQ 方式を指す。右側 3 列は PFRS 方式の場合で同じ各時間を示している。再送と順序制御による時間は、シミュレーション結果の方が若干大きい値ではあるが、ほぼ同程度となっており、PFRS 方式による全フロー ARQ 方式に対する遅延時間の改善については傾向がよく一致している。パケット誤り率が大きい場合にスループットに関する計算結果とシミュレーション結果が一致しない原因は以下が考えられる。

- UDP の場合に説明したように、解析における SR ARQ では再送が理想的に行われると仮定しているのに対し、シミュレーションで用いた SSCOP では 2 回目以降の再送に時間を要しスループットが低下する。
- 解析では送信待ち時間を 0 としたが、図 4.22 において平均送信待ち時間は無視できない値となっており、このためにシミュレーションのスループットが計算結果より低くなる。この平均送信待ち時間は、PFRS 方式より全フロー ARQ 方式の方が大きい。この送信待ち時間は SR ARQ の順序制御によるバースト的なパケット出力に起因するものと考えられる。TCP の受信側で SR ARQ が出力したパケットがバースト的に到着すると、多数の ACK がほぼ同時に送信側に返送される。TCP の送信側では常に送信データが存在するとしているので、これらの ACK が到着すると TCP の送信側はバースト的にパケットを出力する。PFRS 方式では、複数の TCP コネクションごとに非同期にこのようなバースト的なパケット出力が発生するが、全フロー ARQ 方式では複数の TCP コネクションで同期してバースト的なパケット出力が生じる可能性があり、送信待ち時間が大きくなるものと考えら

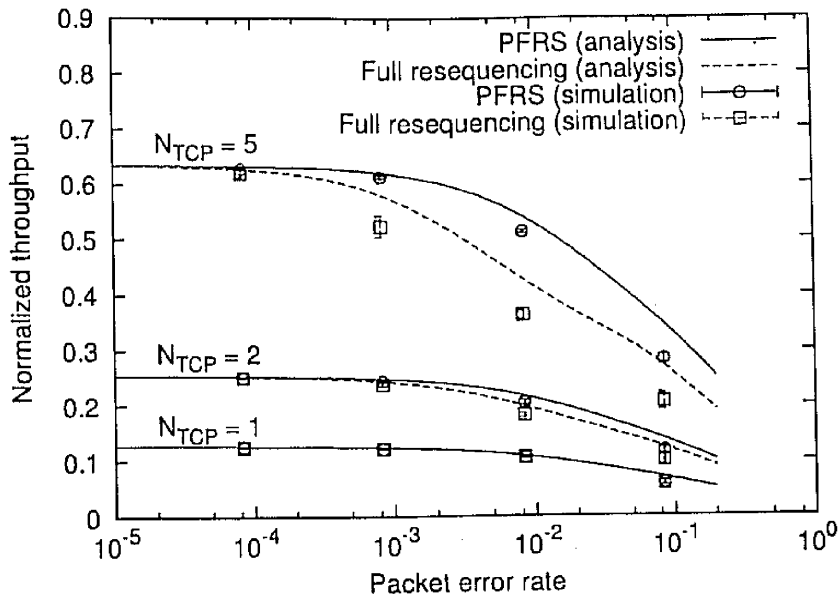


図 4.21 TCP のコネクション数を 1,2,5 とした場合の packet 誤り率とスループットの関係 ($M = 376, N_r = 8$)

れる。

解析では複数フローの packet がランダムに出現すると仮定したが、実際の TCP のトラヒックでは必ずしもランダムとは限らない。しかし、計算とシミュレーション結果でスループットの傾向はよく似ており、この仮定は概ね妥当であると考えられる。

4.6.3 解析により求めた TCP のタイム値とスプリアスタイムアウトの発生確率

上で述べたようにレイヤ 2 の SR ARQ における再送により、TCP から見た遅延時間は大きく変動し、TCP ではタイムアウトが発生する可能性がある。このタイムアウトは packet が失われていないにもかかわらず発生するため、スプリアスタイムアウト (spurious timeout) と呼ばれる。TCP ではラウンドトリップ遅延時間の平均とその平均偏差 (mean deviation) を測定し、平均遅延時間に平均偏差の 4 倍を加えた値をタイム値としている [10]。

実際の推定値では、特定の重みによる移動平均を計測しており、遅延時間の変動により平均遅延時間、平均偏差及びタイム値も変動するが、以下では遅延時間分布の解析より求

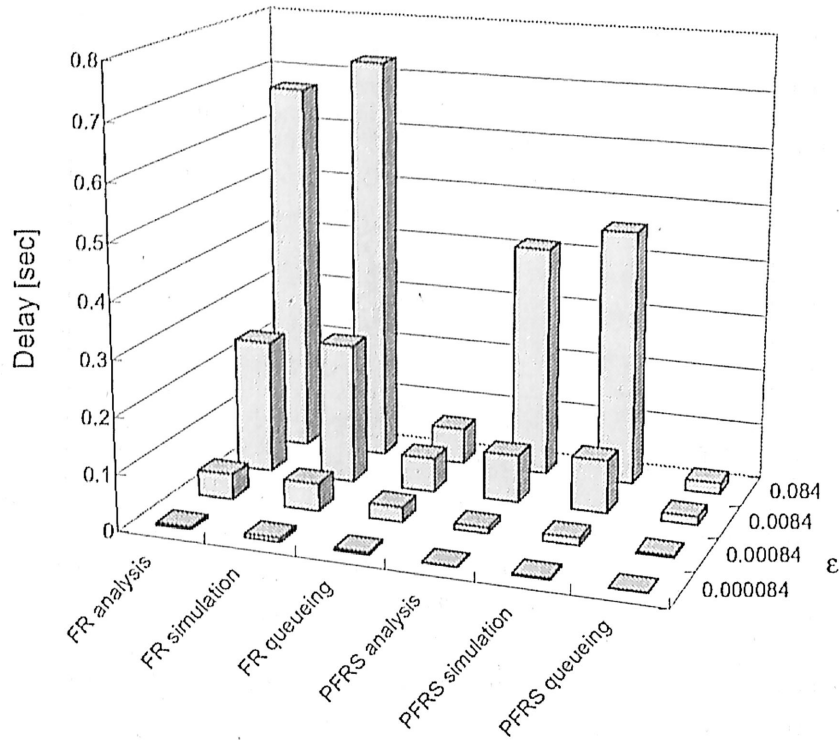


図 4.22 再送と順序制御による平均遅延時間及び平均送信待ち時間 ($M = 376, N_r = 8$)

められる平均値及び平均偏差でタイム値を評価する。

図 4.23 は、図 4.21 で $N_{TCP} = 5$ の場合におけるタイム値の計算結果を示している。誤りがなく再送が全く発生しない場合、セグメントの送信開始から ACK が帰るまでの時間はフレーム長の時間 (M スロット) となる。図 4.23 において、平均遅延時間と TCP のタイム値については、このフレーム長分の固定時間を差し引いており、またすべての値はフレーム長の時間で規格化している。PFRS 方式では順序制御による遅延時間が少なくなるので、同じパケット誤り率では全フロー ARQ 方式よりタイム値は小さく設定されることが分かる。パケット誤り率が大きくなると基本的にはタイム値が大きくなるが、全フロー ARQ 方式の場合、パケット誤り率が 0.01 から 0.03 で若干の減少が見られる。これは、遅延時間分布の形がパケット誤り率により変化し平均偏差がこの部分で減少することによるものである。

図 4.24 は、図 4.21 の場合で TCP のコネクション数 N_{TCP} を 5 とした場合のタイムアウトの発生確率を示している。この確率は計算したタイム値よりも順序制御による遅延時間が大きくなる確率を計算により求めている。図 4.23 及び図 4.24 において、 α の値はパケット誤り率とともに変化するため、式 (4.2) と (4.3) から反復計算により求めた α の値

を用い、これからタイム値とタイムアウト発生確率を計算している。

実際にはタイムアウトが生じるとタイム値を2倍とする指数バックオフが行われるため、図で示す確率そのままの値でタイムアウトが発生するとは限らないが、全フロー ARQ 方式と PFRS 方式でタイムアウトの発生について傾向を把握することができる。パケット誤り率が小さい場合、全フロー ARQ 方式の方が PFRS 方式に比べタイムアウト発生確率が大きい。これは順序制御による遅延時間とその変動が大きいことによる。全フロー ARQ 方式のタイムアウト確率は 10^{-3} 付近で急に小さくなっているが、これはタイム値がフレーム長の時間を超えたことによる。ここで、図においてタイム値はフレーム長で規格化しているので、タイム値が1を超えるとタイムアウト確率が小さくなる。SR ARQ で再送が発生すると最小でも1フレーム時間分の遅延が発生する。タイム値がフレーム長より短い場合、SR ARQ で再送が発生すると、TCP でも必ずタイムアウトが発生することになるが、タイム値が1を超えると1回の再送ではTCPのタイムアウトは発生しなくなる。これが急にタイムアウト確率が小さくなる理由である。PFRS 方式においても 10^{-2} 弱のパケット誤り率で急に確率が小さくなるが、同じ理由による。このようなタイムアウト確率が急に小さくなる現象は、図 4.23 において規格化されたタイム値が1, 2, 3等の整数値を超える度に生じる。

以上から全フロー ARQ 方式では PFRS 方式より小さなパケット誤り率でタイムアウトが発生しやすい傾向のあることが分かる。実際の TCP におけるタイムの設定は同時には1タイムのみであり、すべてのパケットにタイムが設定されるわけではなく、またタイムは新規 ACK 受信時にリセットされる [81]。このため、タイムアウトの発生確率は必ずしも計算値と定量的に一致するわけではない。しかし、同じ条件で PFRS 方式と全フロー ARQ 方式と比較した場合の結果は有効と考えられる。

4.7 むすび

この章では衛星回線において SR ARQ により誤り回復を行うアーキテクチャについて考察し、多数のフローを SR ARQ に多重化する場合、SR ARQ の受信側での順序制御により遅延時間が大きくなる問題、またこの順序制御によりパケットがバースト的に出力される問題のあることを述べた。

これらを解決する方式として、パケット喪失の検出と再送は上位層のフロー全体で行い、順序制御のみを上位層のフロー個別に行う PFRS 方式を提案し、エンドツーエンドで UDP 及び TCP が用いられる場合についてシミュレーションによる評価を行った。この結果、従来の全フロー ARQ 方式と比較すると、UDP トラフィックの場合は PFRS 方式により平均遅延時間や順序制御のために保留されるバッファ量が減少することが分かった。

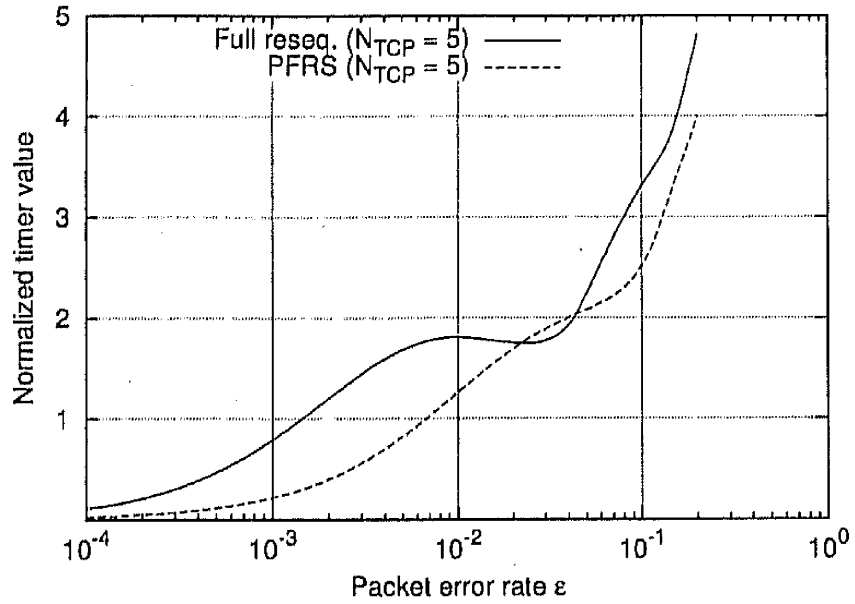


図 4.23 TCP のタイマ値 ($M = 380, N_r = 8, N_{TCP} = 5$)

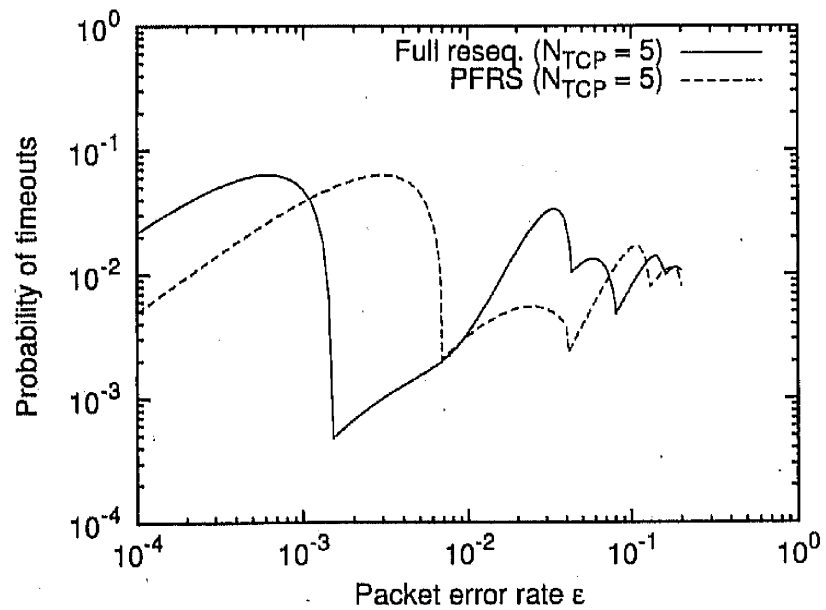


図 4.24 タイムアウト発生確率 ($M = 380, N_r = 8, N_{TCP} = 5$)

またエンドツーエンドで TCP が用いられる場合、出力でのバースト長が小さくなること、スループットが向上することが分かった。フローごとに ARQ を用いる個別フロー ARQ 方式と比較すると、UDP トラフィックの場合は PFRS 方式の遅延時間や保留バッファ数が少なく、TCP トラフィックの場合はスループットやバースト性がほぼ同等であることが分かった。

3 章において、理想化されたモデルを用いて順序制御による遅延時間を解析したが、これから求めた遅延時間分布及び TCP のスループットを本章のシミュレーションと比較した。その結果、解析が実際の SR ARQ プロトコルの場合にも有効であることが明らかとなった。

提案した PFRS 方式は 1 本の静的な SR ARQ コネクションを用いればよく、個別フロー ARQ 方式に比べ SR ARQ コネクションの設定や SR ARQ が必要とする順序制御のためのバッファなどのリソース管理が単純になる特長がある。このような特長は、上位層のフローを把握することが難しい IP トラフィックにおいて特に有効である。この章では PFRS 方式の具体的な実現方式については述べなかったが、これについては次章で述べる。

この章では、衛星通信を対象として SR ARQ の順序制御による問題について検討したが、この検討の結果は衛星通信に限らず伝搬遅延時間と通信速度の積が大きなシステムに対し適用できる。地上の無線 LAN や無線アクセスシステムで、通信速度が 100 Mbit/s 以上のものが検討されているがこのようなシステムで、特に移動通信を考慮する場合には強力な ARQ が必要で SR ARQ を用いることになる。このような場合についてもこの章で提案した PFRS 方式が有効と考えられる。

第5章

ポインタによる PFRS の実現方式

5.1 まえがき

4章では SR ARQ の順序制御による遅延時間やバースト的なパケット出力の問題を解決する方式として PFRS (Per Flow ReSequencing) 方式を提案した。この PFRS 方式の実現については、上位層フローごとのパケットの順序識別の課題が存在する。この章ではこの PFRS 方式の実現方式について検討し、同一フローに属するパケットの識別方式として SR ARQ の通番を利用するポインタ方式を提案する。このようなポインタはパケットを伝送する上での制御情報のオーバーヘッドを増加させるが、通番の差分によるポインタを用いることにより、ポインタによるオーバーヘッドの増加を抑えることができることを示す。ここで、SR ARQ としては SREJ を用いる HDLC[72] や SSCOP[65] など送信データに通番を付与するプロトコルを想定している。また、SR ARQ がレイヤ 2 で実現されるものとし、SR ARQ が送受信するデータの単位をこの章では L2 フレーム (レイヤ 2 フレーム) と呼び、上位層のパケットがこの L2 フレームにより運ばれるものとする。

以下、5.2 では PFRS 方式とその実現課題及びこれを解決するポインタ方式について述べ、5.3 でその送受信処理アルゴリズムを説明する。5.4 では差分ポインタについて説明し、シミュレーションによる評価を行う。その結果、差分ポインタでは値に上限を設けることによりポインタによるオーバーヘッドの削減が可能であることを示す。

5.2 PFRS 方式の実現課題とポインタ方式の提案

5.2.1 PFRS 方式の概要

同一のフローではパケットの順序保証が必要であるが、異なるフローに属するパケット間ではパケットの順序を保証する必要はない。4章で提案した PFRS 方式はこれに着目

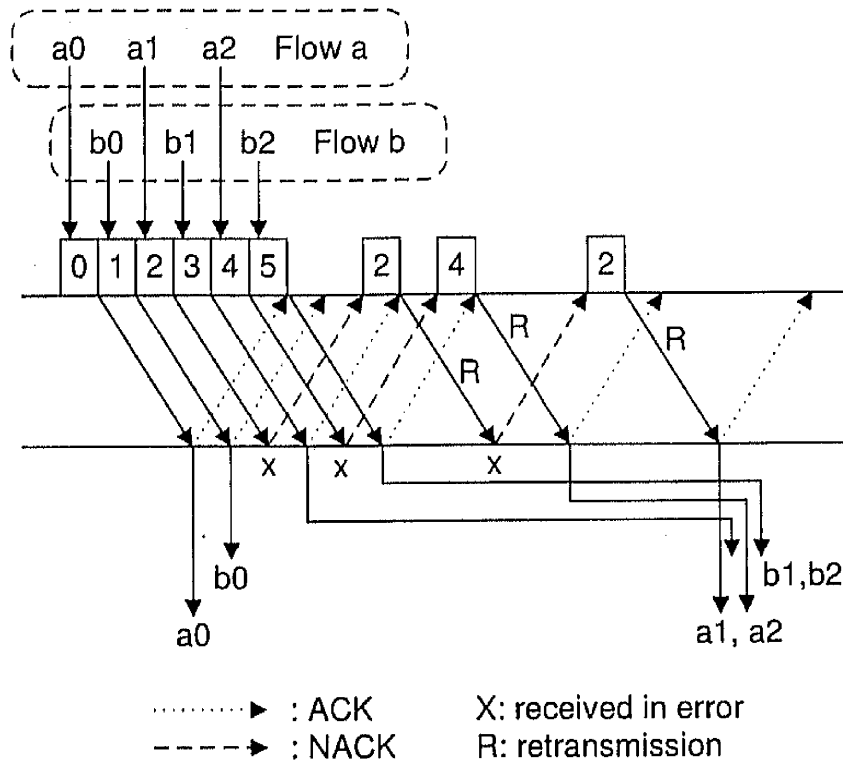


図 5.1 全順序制御方式のシーケンス例

し、喪失したパケットの検出と再送は複数のフローが多重化された 1 本の SR ARQ により行うが、受信側での順序制御はフローごとに独立に行う。図 5.1 に示すように SR ARQ に多重化したすべてのパケットについて順序を保証する従来の全順序制御方式では、不必要にパケットの保留と遅延を引き起こす場合がある。図で、a と b の 2 本のフローが SR ARQ に多重化されているとしている。フロー a のパケット a1 と a2 が伝送誤りにより再送されているが、全順序制御では異なるフロー b のパケット b1 と b2 が、フロー a のパケット a1 を受信するまで不必要に保留されており、一種の HOL ブロッキングが発生している。PFRS 方式は図 5.2 に示すように、パケットの送達確認や再送は従来のように SR ARQ で行うが順序制御についてはフロー単位に行うため、フロー b のパケット b1 と b2 は受信後保留されずに直ちに上位層に転送される。このような PFRS 方式により、HOL ブロッキングによる不必要なパケット保留を解消でき、順序制御による遅延時間の削減やバーストに含まれるパケット数の削減が可能である。

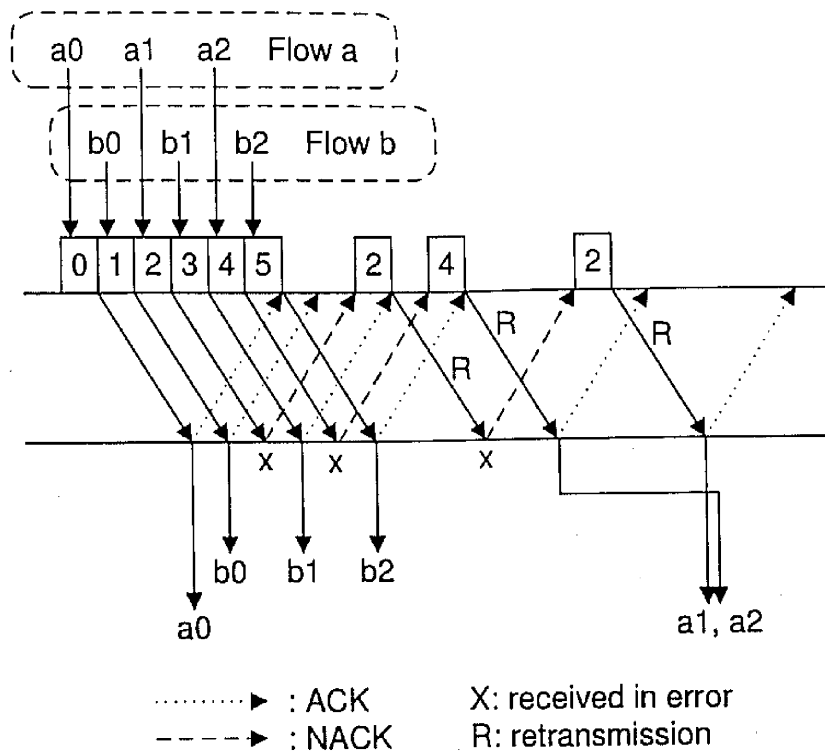


図 5.2 PFRS 方式のシーケンス例

5.2.2 PFRS 方式の課題

PFRS 方式実現のためには以下の機能が必要となる。

- 上位層のフロー識別
- フローごとのパケット順序識別

フロー識別は IP ヘッダに含まれる宛先及び送信元の IP アドレスとポート番号及び上位プロトコル ID により行う。このように、PFRS 方式では SR ARQ において上位の IP 層や TCP 層のヘッダ情報の一部を参照する必要がある。しかし、上位層ヘッダの変更などは行わないので、上位層として IP 及び TCP が用いられた場合、エンドツーエンドの通信を保存する。

PFRS 方式では可能な限り細かいフローを識別してその単位で順序制御を行うのがよい。しかし IPsec などが用いられると上位層のポート番号などが識別できなくなる。このような場合は宛先及び送信元アドレスのみで識別することになる。同一の宛先及び送信

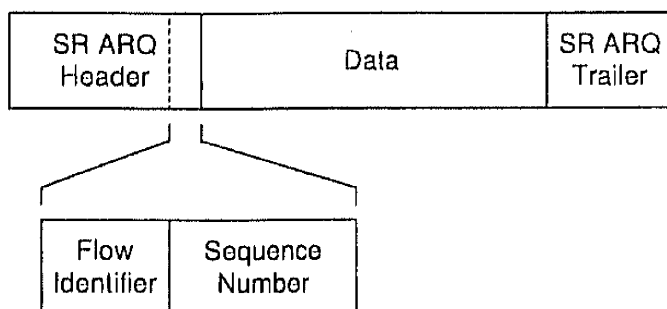


図 5.3 フロー識別子と通番

元アドレスのフロー間では HOL ブロッキングによる不必要な受信パケット保留が発生する可能性はあるが、異なる宛先及び送信元アドレスをもつフロー間では順序制御が独立に行われ HOL ブロッキングの問題は解消される。

5.2.3 同一フローに属するパケットを運ぶレイヤ 2 フレームの順序識別情報

フローごとのパケットの順序識別については、TCP などの上位層プロトコルの通番を参照する方式が考えられる。しかし上位層のプロトコルに強く依存する問題、また TCP の場合、SR ARQ の区間以外でパケットの喪失が発生すると TCP の通番の不連続が発生し、SR ARQ の順序制御への利用は難しい問題もある。フローの識別は上位層の情報を利用するとしても、各フローに属するパケットを運ぶ L2 フレームの順序識別は上位層に頼らず SR ARQ で管理することが望ましい。これを実現するためには、図 5.3 に示すようなフロー識別子 (Flow Identifier) と各フロー識別子ごとに独立した通番 (Sequence Number) を設け、L2 フレームのヘッダに付加する方式が一般的に考えられる。しかしこの方式では以下の問題がある。

- フロー識別子として用いるビットの数を決めなければならないが、SR ARQ に多重化されるフロー数の上限を規定することができず、どれだけのビット数が必要か予測できない。
- フロー識別子は新しいフローの検出時に空いている値を割当て、フローが終了したら解放して別のフローへの割当てを行う管理が必要となる。新しいフローの開始を検出することは容易と考えられるが、既に識別子を割り当てたフローの終了を検出するには、タイマで一定時間以上当該フローのパケットを運ぶ L2 フレームが流れていないことを検出するか、上位層のプロトコルのヘッダを見てフローの終了を検

出すなどの処理が必要となる。継続時間の短いフローが多数 SR ARQ に多重化される場合、この処理負荷が大きくなる。

- フロー識別子を再利用する場合、通番のリセットを送信側と受信側で同期をとって行う必要がある。
- 通番としては基本的に SR ARQ と同じビット数の通番が必要となるが、フロー識別子と通番によるオーバーヘッドが存在する。

本論文では、これらの問題を解決する方式として SR ARQ の通番を利用して同一フローの packets を運ぶ L2 フレームを識別する方式を提案する。この方式をポインタ方式と呼ぶ。

5.2.4 ポインタ方式

この方式は、あるフローに属する packets を運ぶ L2 フレームを送信する際、同じフローに属する直前の packets を運ぶ L2 フレームを SR ARQ の通番により指定する方式である。SR ARQ のヘッダに通常の通番とともに L2 フレームを指定するための通番を付加する。以降の説明でこの付加された通番をポインタと呼ぶ。L2 フレームの形式としては図 5.3 において、フロー識別子を省いた通番のみの形となる。

新規フローの最初の L2 フレームは順序制御を行わず受信側で無条件に出力できる。このような場合、ポインタの値を自分自身の SR ARQ 通番と同じ値とし、自分自身の L2 フレームを指す場合は順序制御の必要なしの意味とする。また、同じフローに属する packets を運ぶ L2 フレームで以前に送られたものがすべて送達確認されている場合も、順序制御を行わず無条件に出力することができる。このような場合にもポインタの値を順序制御の必要なしとする。ACK の通番がその通番より前の L2 フレームすべての送達確認を意味する累積確認応答型のプロトコルでは、すべてが送達確認されたか否かの判断の実現が容易である。

図 5.4 はポインタによる順序制御のシーケンス例、図 5.5 はポインタによる同一フローに属する L2 フレームの識別例を示している。図で L2 フレームを In_p と表記しているが、 n は SR ARQ の通番、 p はポインタを示している。SR ARQ で以前に送信が行われた通番 5 までの L2 フレームはすべて送達確認されているものとし、通番 6 の L2 フレームから送信している。また図 5.4 において NACK は伝送誤りのある L2 フレームの受信時点ではなく、次に誤りのない L2 フレームを受信した時点で通番の不連続により L2 フレームの喪失を検出して送信されており、また抜けた通番を含んでいる。実際に用いられる SR ARQ プロトコルではこのように伝送誤りのある L2 フレームを無視し、正しく受信した L2 フレームのみを利用するものが一般的である。

図で I6,6 及び I7,7 の L2 フレームはそれぞれフロー b とフロー a のパケットを運ぶ L2 フレームで、I7,7 の L2 フレームに伝送誤りが発生したとしている。以前に送信された L2 フレームはすべて送達確認されているので、どちらも順序制御の必要はなく、ポインタの値はそれぞれの通番の値と等しくなっている。フロー b のパケット b4 と b5 はそれぞれ L2 フレーム I8,6 と I9,8 により送信されている。フロー b のパケット b3 と b4 は、それぞれ通番 6 と 8 の L2 フレームで運ばれているので、通番 8 と 9 の L2 フレームのポインタ値はそれぞれ 6 と 8 となる。このようなポインタにより、フロー b のパケットを運ぶ L2 フレーム間でポインタによる順序関係が図 5.5 に示すように指定される。L2 フレーム I8,6 が受信されると、そのポインタの指す L2 フレーム I6,6 は既に上位層に転送されているので、L2 フレーム I8,6 が運ぶパケット b4 を直ちに上位層に転送できる。このようにフロー a のパケット a3 を運ぶ L2 フレーム I7,7 が失われているが、これに影響されずフロー b のパケットを転送できる。フロー a のパケット a4 を運ぶ通番 10 の L2 フレームはポインタの値が 7 となる。この L2 フレームは受信された時点でポインタの指す通番 7 の L2 フレームが受信されていないので、通番 7 の L2 フレームが再送されてくるまで保留される。同様に、フロー a のパケット a5 を運ぶ通番 11 の L2 フレームはポインタの値が 10 となり、L2 フレーム 10 のパケットが上位層に転送されるまで保留される。以上のように受信側では受信した L2 フレームのポインタにより順序制御による保留の要否、また再送された L2 フレームを受信して、保留されている L2 フレームの中でどの L2 フレームのパケットを上位層に転送するかを判断することができる。

このような SR ARQ の通番によるポインタを用いる方式は以下の特長がある。

- ポインタは送信する L2 フレームと同じフローに属するパケットを運ぶ L2 フレームで送達確認待ちのものがある場合に限り送達確認待ちの最後の L2 フレームを指すだけでよい。ポインタとして SR ARQ の通番と同じビット数があれば任意の数のフローを識別でき、多重化するフロー数に制限がない。
- 送信側でフローの開始や終了を特に意識する必要がない。トランザクショナルなトラフィックが多い場合やコネクションレス通信に適した方式である。
- 受信側はパケットに付与されたポインタに従って処理を行えばよく、送信側と受信側での通番の同期処理などは不要である。
- SR ARQ の通番相当のオーバーヘッドは必要であるが、フロー識別子は不要でありこの分オーバーヘッドが少なくて済む。

以上の特長からポインタ方式は PFRS 方式の実現に適しているといえる。ポインタ方式において、多重化するフロー数に制限がない利点は、順序関係の指定が送達確認待ちの L2 フレームに対してのみ行えばよいこと、そして送達確認待ちの L2 フレームは SR

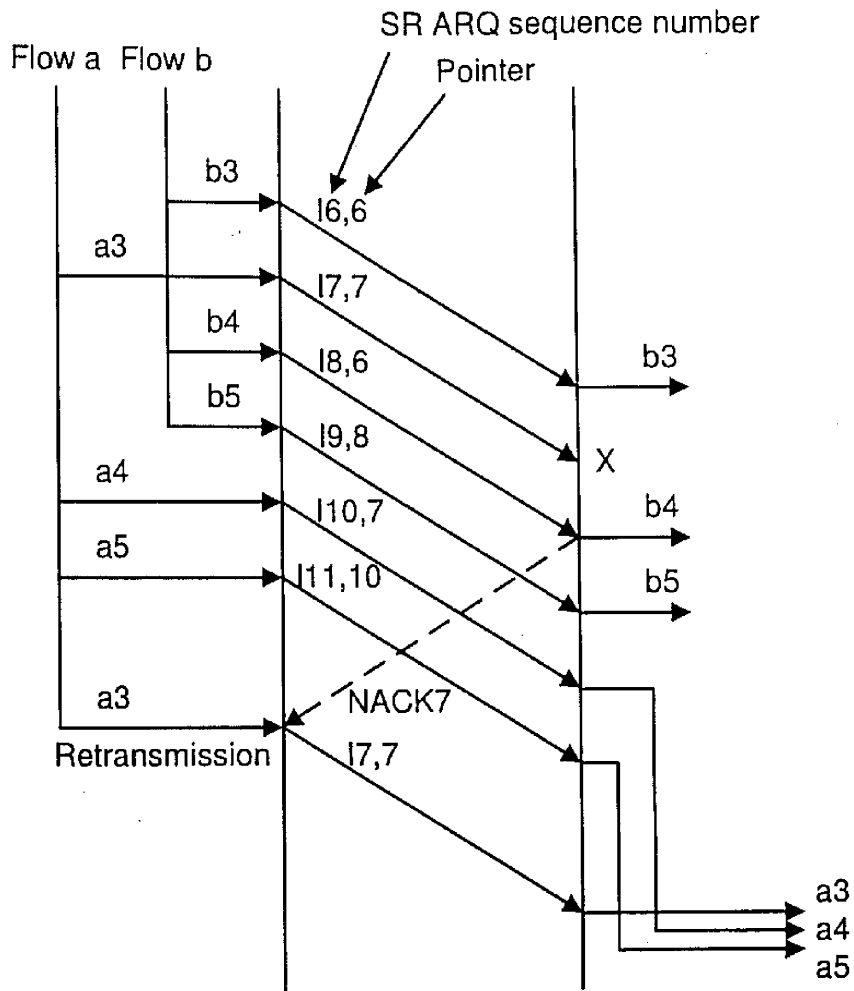


図 5.4 ポインタによる順序制御のシーケンス例

ARQ の通番により常に一意に識別できることから生じている。

5.3 送受信処理アルゴリズム

提案したポインタ方式では L2 フレームの送受信において処理の追加が必要となる。累積確認応答型のプロトコルの場合について具体的なアルゴリズムを以下に説明する。

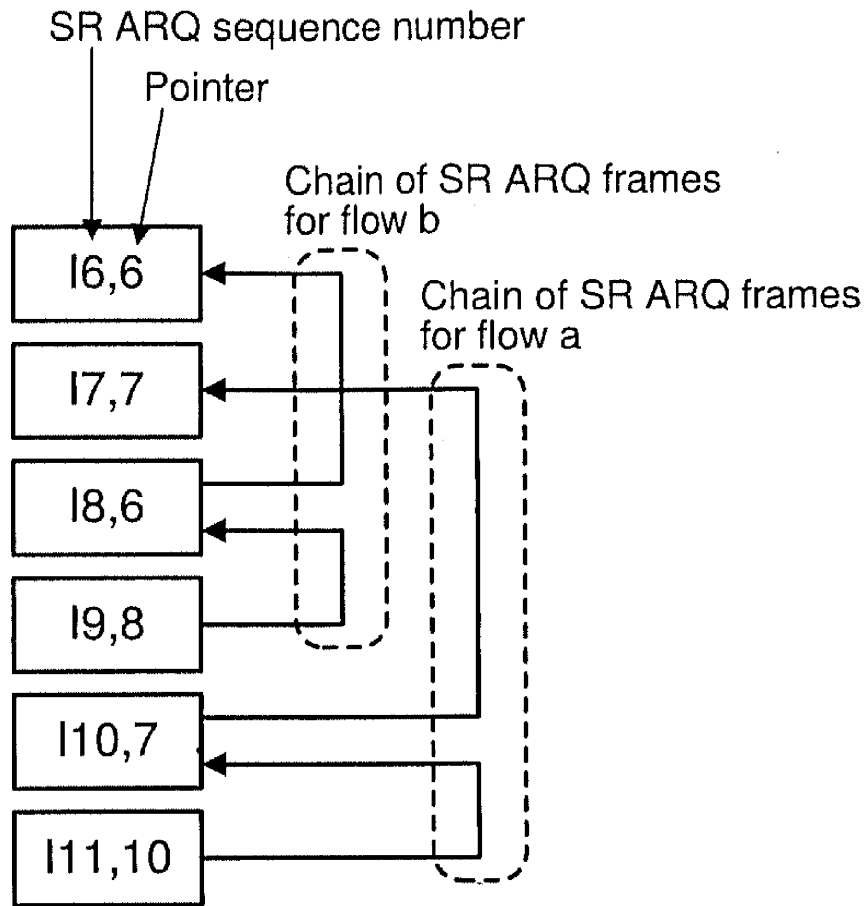


図 5.5 ポインタによる同一フローに属する L2 フレームの順序関係

5.3.1 送信処理

基本的には新規送信の場合にポインタを決定する。以下に述べる処理の中で再送時のポインタ更新処理は必須ではないが、受信側での処理を簡単にする効果がある。

■新規送信の場合

- 新規に送信するパケットのフローを識別する。
- 送達未確認の L2 フレームの中で同じフローのパケットを運ぶ L2 フレームを探す。
- 送達確認されていない L2 フレームの中で、同じフローに属するパケットを運ぶ L2 フレームが存在する場合は、最後に送信された L2 フレームの通番をポインタとする。図 5.6 は左から右の方向に送信した通番を表しており一番右の通番がこれから送信する L2 フレームの通番、四角い箱の中は送達確認待ちの通番を示している。

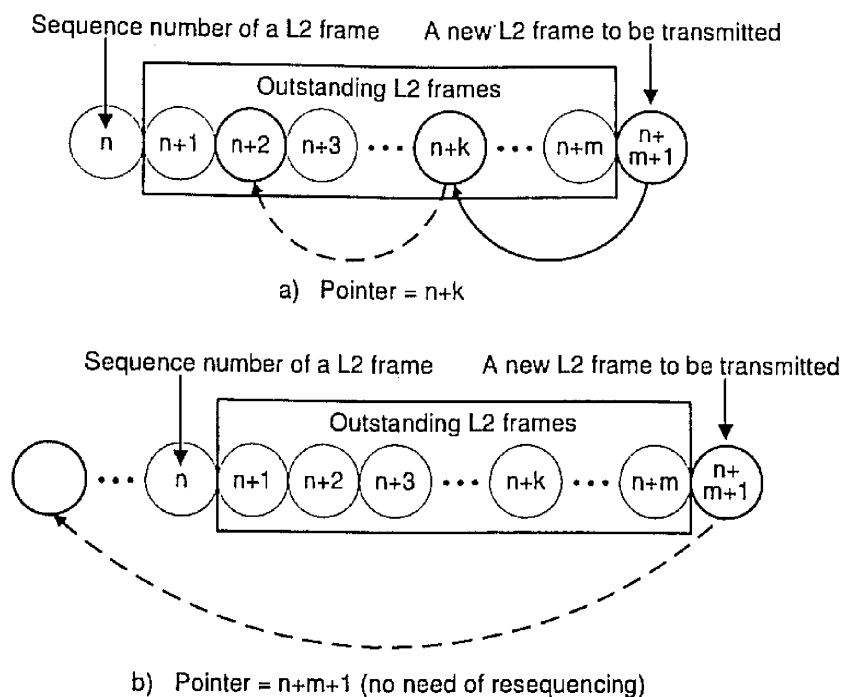


図 5.6 送信側でのポインタ値の決定

新規に L2 フレームを送信する場合、右から左の方向にこの L2 フレームが運ぶパケットと同じフローのパケットを運ぶ L2 フレームが存在するか探索し、図の a) の場合のように箱の中に存在すればその L2 フレームの通番をポインタとする。

- 図 5.6 の b) の場合のように、送達確認待ちの L2 フレームの中で、同じフローに属するパケットを運ぶ L2 フレームが存在しない場合は、順序制御の必要性はないのでポインタの値を自身の通番の値とし、順序制御が必要ないことを示す。

■L2 フレームを再送する場合

- 再送する L2 フレームのポインタが指す L2 フレームが既に送達確認されていれば順序制御の必要はないのでポインタの値を自身の通番の値とする。
- 送達確認されていない場合は、前回送信と同じポインタの値を変更せずに用いる。

5.3.2 受信処理

受信処理の概要を図 5.7 に示す。既に説明したように L2 フレームに付与されるポインタは、同じフローに属する直前のパケットを運ぶ L2 フレームを指しており、L2 フレー

ムの送信順序とは逆方向となっている。受信側での処理では、再送された L2 フレームや保留されていた L2 フレームを上位層に転送した場合、同じフローに属する後続の L2 フレームで保留されているものが存在するか探索する必要がある。この探索はポインタの指す方向とは逆方向である。受信側では後続の L2 フレームを指す順方向ポインタを用いると処理がシンプルとなるため、図 5.7 では通番ごとの状態を示す領域と、通番ごとに同一フローの次 L2 フレームへの順方向ポインタ (next frame pointer) を格納する領域をもつテーブルを使用することを前提としている。順方向ポインタは受信した L2 フレームのポインタから割り出してテーブルに設定する。図 5.7 で幅広の矢印内の番号は、以下の説明において該当する段落番号または小段落番号を示している。既存のプロトコルでは、次に受信を期待する通番が重要な役割を果たしており、既存プロトコルに PFRS 方式を追加する場合にはこれを意識した処理が必要である。

■受信 L2 フレームの通番が次に受信を期待する値の場合 受信した L2 フレームの packets を上位層に転送し、通番の状態を配達済みとする。上位層に転送した L2 フレームの順方向ポインタが保留している L2 フレームを指していれば、この保留 L2 フレームも上位層に転送し、同様な処理を順方向ポインタの指す保留 L2 フレームがなくなるまで続ける。最後に、次に受信を期待する通番を更新する。

■受信 L2 フレームの通番が次に期待する値ではない場合

(1) 受信 L2 フレームのポインタ値が「順序制御の必要なし」の値か、またはポインタの指す L2 フレームが上位層に配達済みの場合 L2 フレームの packets を上位層に転送し、その通番の状態を配達済みとする。上位層に転送した L2 フレームの順方向ポインタが保留している L2 フレームを指していれば、この保留 L2 フレームも上位層に転送し、同様な処理を順方向ポインタの指す保留 L2 フレームがなくなるまで続ける。

(2) 受信 L2 フレームのポインタ値が「順序制御の必要なし」の値でもなく、かつこのポインタの指す L2 フレームが上位層への配達済みでもない場合 受信した L2 フレームを順序制御用のバッファに保存するとともに、ポインタの指すテーブルの通番エントリに受信 L2 フレームの通番を順方向ポインタとして書き込む。

図 5.8 は図 5.4 のシーケンス例における具体的な受信処理を示している。この図では上で述べた通番ごとの状態を示す領域と、同一フローの次 L2 フレームへの順方向ポインタ (next frame pointer) を格納する領域をもつテーブルを使用している。また説明の便宜上、状態を示す領域には受信または保留中の L2 フレームも表示している。図 5.8 では通番 5 までの L2 フレームはすべて配達済み (delivered) とし、通番 5 以降のみを示している。最初に L2 フレーム I6,6 を受信すると、この L2 フレームのポインタは自分自身を指しているため順序制御を行う必要はなく、L2 フレームの packets を上位層に転送すると

ともに通番の状態を配達済みとする。次に、L2 フレーム I8,6 を受信すると、そのポインタが指す通番 6 の L2 フレームは配達済みであるので、この L2 フレームのパケットを上位層に転送するとともに通番 8 の状態を配達済みとする。L2 フレーム I9,8 も同様の処理となる。L2 フレーム I10,7 を受信すると、そのポインタの指す通番 7 の L2 フレームは未受信であるので、通番 10 の L2 フレームを保留する。この際、ポインタの指す通番 7 の順方向ポインタに自身の L2 フレームの通番である 10 を登録する。同様に、L2 フレーム I11,10 を受信すると、ポインタの指す通番 10 の L2 フレームは保留中であるので、通番 11 の L2 フレームを保留し、ポインタの指す通番 10 の順方向ポインタに自身の L2 フレーム通番である 11 を登録する。再送された L2 フレーム I7,7 を受信すると、そのポインタは自分自身を指しているので順序制御を行う必要はなく、この L2 フレームのパケットを上位層に転送するとともに通番 7 の状態を配達済みとする。次にテーブルにおいて通番 7 の順方向ポインタは通番 10 の L2 フレームを指しているので、通番 10 の L2 フレームのパケットを上位層に転送し、通番 10 の状態を配達済みとする。更に通番 10 の順方向ポインタは L2 フレーム 11 を指しているので L2 フレーム 11 についても同様の処理を行う。通番 11 の順方向ポインタは L2 フレームを指していないので、上位層への転送処理は終了する。次に L2 フレーム I7,7 は次に受信を期待されていた L2 フレームであったので、次に受信を期待する通番を更新し、この例ではその値が 12 となる。

以上説明したように、テーブルにおいて喪失した L2 フレームの通番の順方向ポインタの領域に後続 L2 フレームの通番を登録することにより、再送 L2 フレームを受信した際に同じフローのパケットを運ぶ保留 L2 フレームを効率よく探索できる。

5.4 SR ARQ 通番の差分によるポインタ方式

5.4.1 差分ポインタ

前節ではフローごとに順序制御を行う PFRS 方式の実現手法として SR ARQ の通番そのものをポインタとして用いる方式を説明した。このバリエーションとして、通番の差分をポインタとして用いる方式も考えられる。これは新規に送信する L2 フレームの通番 SN_{new} から同じフローに属する直前のパケットを運ぶ L2 フレームの通番 SN_{prev} を差し引いた値を用いるものである。

$$Pointer = (SN_{new} - SN_{prev}) \bmod 2^N \quad (5.1)$$

ここで N はポインタのビット数である。以下、このようなポインタを差分ポインタと呼ぶ。順序制御の必要がない場合、差分ポインタの値は 0 となる。先の説明でポインタは送達確認待ちの L2 フレームの中で同じフローに属するパケットを運ぶ最後の L2 フレーム

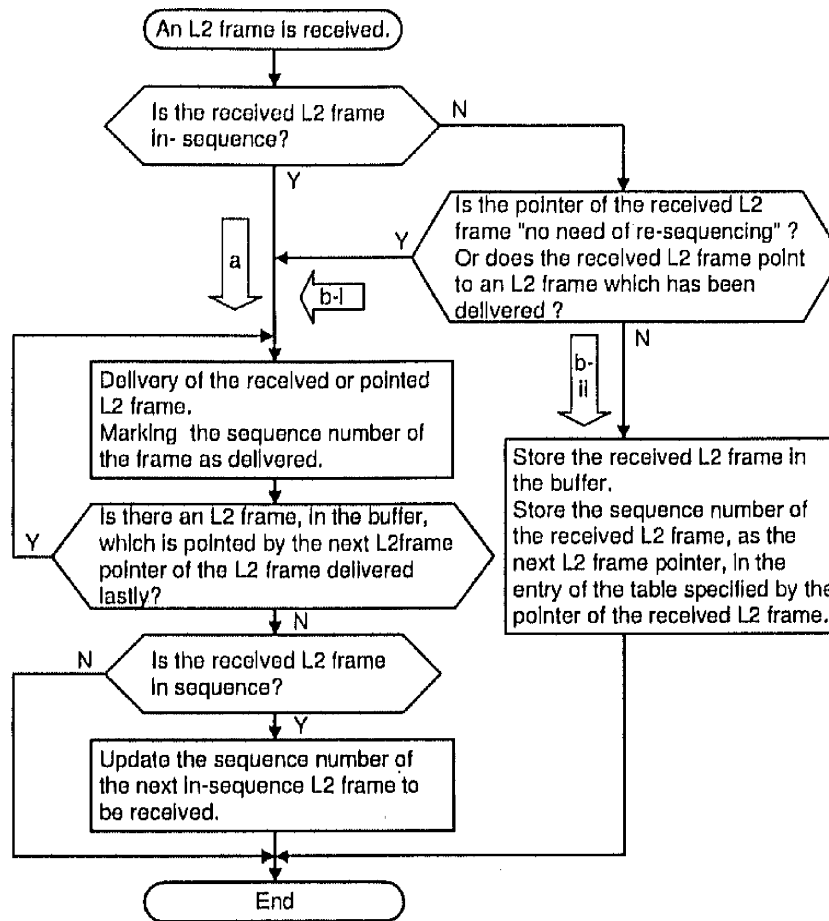


図 5.7 受信側での処理

を指せばよいことを説明した。一般に送達確認待ちの L2 フレーム数はその最大値が送信バッファ量や受信側のバッファ量から決まり、SR ARQ の通番のスペースよりも小さい場合が多い。このような場合は、通番の差分によるポインタを用いれば、ポインタのビット数を最大送達確認待ちの L2 フレーム数を指定可能なビット数に減少させることが可能である。

また以下の定性的な考察から、この差分ポインタは大きな値をとる場合は少なく、かつその値が大きいほど順序制御による遅延時間も小さくなると考えられる。

- a) 一般にアクティブなフローでは一定速度または連続してパケットが送信され、同一フローのパケット間に他のフローのパケットが割り込むことにより、差分ポインタの値が変動する。その値は同時に送信するアクティブなフロー数により抑えられ、差分ポインタがある程度以上の大きな値をとる場合は比較的少ないと考えられる。

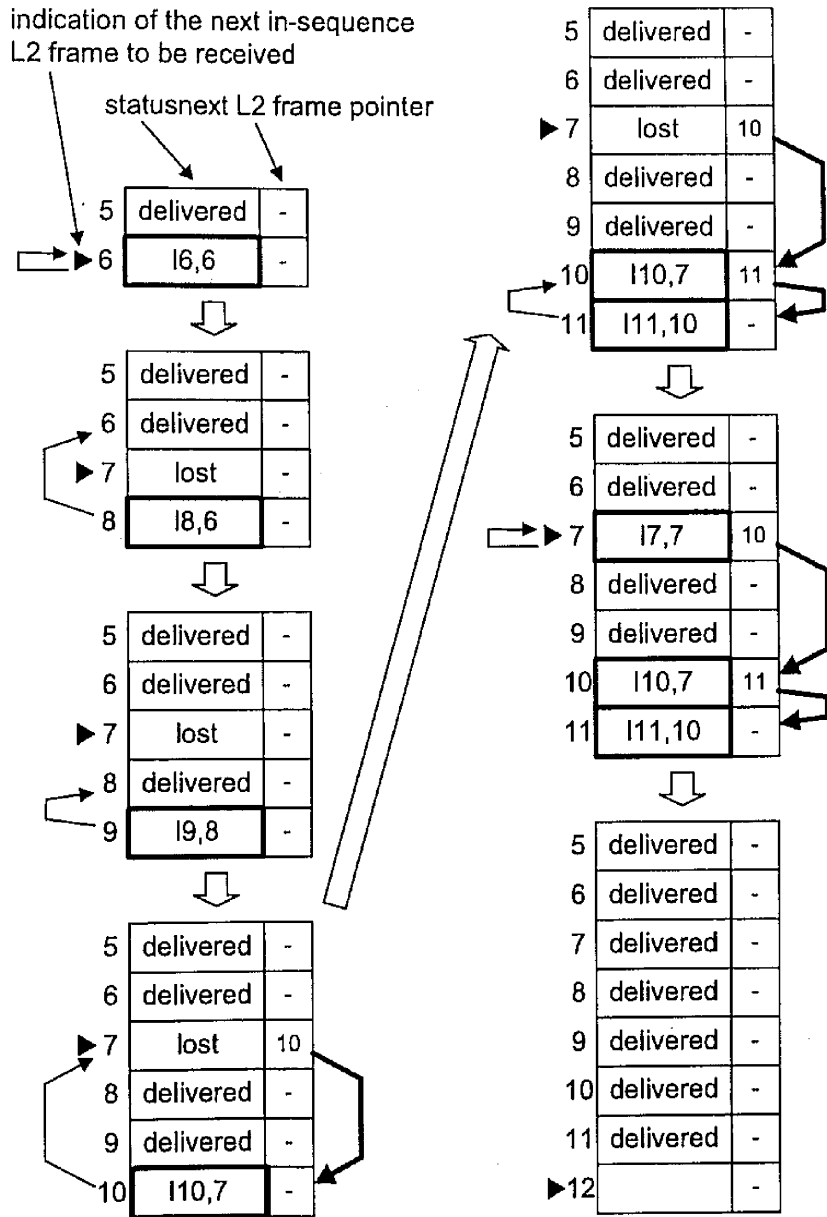


図 5.8 受信処理の例

- b) あるフローのパケットを運ぶ L2 フレームが再送され、同じフローの次のパケットを運ぶ L2 フレームが誤りなく受信され順序制御により保留される場合を考えると、順序制御による遅延時間は前の L2 フレームの再送に要する時間から二つの L2 フレームの送信時間間隔 (L2 フレームの初回送信時刻の差) を引いた時間となり、送信時間間隔が大きいほど順序制御による遅延時間は少なくなる。差分ポインタの値が大きい場合、同じフローのパケットを運ぶ L2 フレーム間に (差分ポインタ値-1) 個の L2 フレームが送信されるので、送信時間間隔が大きくなる。これから、差分ポインタの値が大きな L2 フレームは順序制御による遅延時間が小さくなる傾向があるものと考えられる。図 5.9 は、正しく受信したパケットについて再送パケットからの通番の差と順序制御による遅延時間の関係を示している。この図で通番 2 のパケットが再送され、通番 3, 4, 5, 6, 7 のパケットでは順序制御によりそれぞれ 5, 4, 3, 2, 1 スロットの遅延が発生する。このように再送パケットから通番が離れるほど順序制御による遅延時間が小さくなることが分かる。また、例えば通番 7 のパケットの順序制御による遅延時間が 1 スロットとなるためには、パケット 3, 4, 5, 6, 7 が誤りなく受信されることが条件となる。一方、通番 3 のパケットの順序制御による遅延時間が 5 スロットとなるための条件はパケット 3 が正しく受信されることである。これから順序制御による遅延時間が小さなパケット程、再送パケットから離れるのでその発生確率も小さくなる傾向が予想される。

以上のような傾向を利用すれば、差分ポインタ値の範囲を最大送達確認待ち L2 フレーム数よりも更に減少させることが可能と予想される。以上を確認するためシミュレーションを行った。シミュレーションは VINT プロジェクトのネットワークシミュレータ ns-2[63] を用い、具体的なレイヤ 2 の SR ARQ プロトコルとして SSCOP を用いた。想定するネットワークの構成は図 4.5 と同じとしている。衛星通信により 6.3 Mbit/s の帯域が提供され、この回線上で SSCOP による SR ARQ が行われ、更に SR ARQ 上を複数の TCP コネクションが設定されているものとしている。シミュレーション条件を表 5.1 に示す。TCP パケットのデータ長を 64、1000、1500 Byte、TCP のバージョンは一般的に用いられている NewReno とした。パケットにはこれら長さのデータ以外に、TCP ヘッダや IP ヘッダ更に SSCOP のヘッダによるオーバーヘッドが 50 Byte 存在するものとし、ウィンドウサイズは 50 としている。データ長が 1000 Byte の場合に、TCP のコネクション数が 5 本で回線の使用率が 60% 程度、10 本で回線がボトルネックとなるような条件として設定した。TCP によるパケットの発生は、常に送信すべきデータが存在する条件で行い、実際の発生はウィンドウサイズにより制御される。スループットなどは乱数の初期値を変えた試行を 10 回行い平均値及び 95% 信頼度区間を求めた。

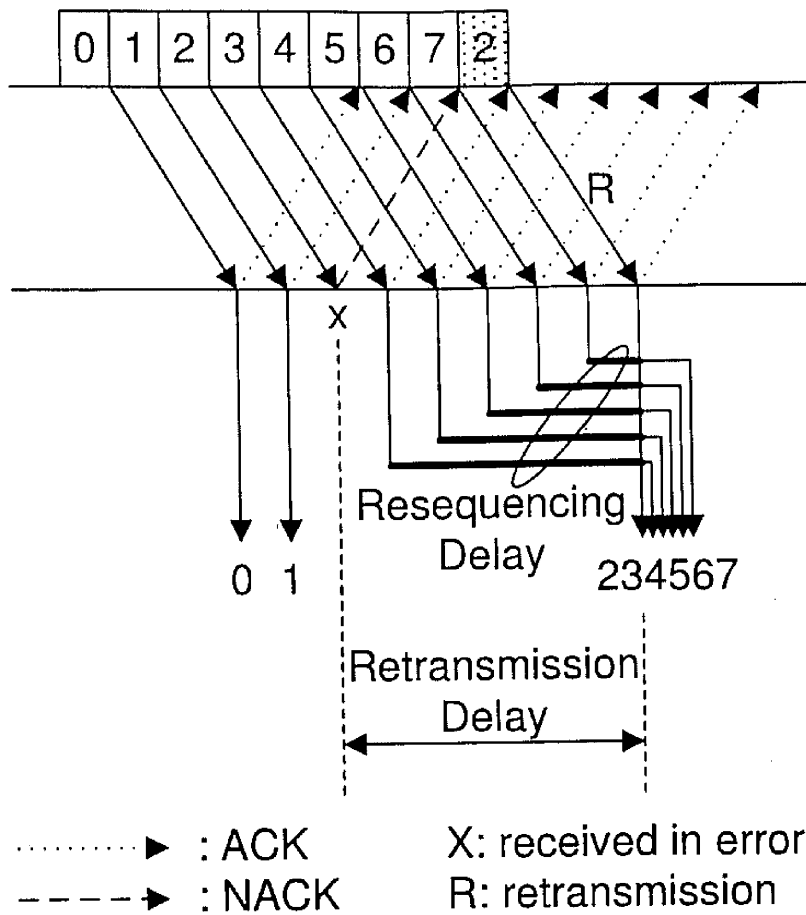


図 5.9 再送パケットからの通番の差と順序制御による遅延時間

5.4.2 差分ポインタと順序制御による遅延時間の関係

図 5.10 は回線の帯域により規格化した平均スループット特性で、差分ポインタに上限を設けない場合の PFRS 方式の性能を示している。ここで、データ長 D_L を 1000 Byte、TCP のコネクション数 N_{TCP} を 10 及び 5 としている。この図で 95% 信頼度区間も示しているが、区間はプロットした点より小さいか同程度となっている。差分ポインタに上限を設ける場合、この図に示す性能が比較対象となる。

図 5.11 と図 5.12 はビット誤り率がそれぞれ 10^{-6} と 10^{-5} の場合において、再送なく受信した L2 フレームについてその差分ポインタの値と順序制御による遅延時間の相関図 (散布図) を示している。この場合もデータ長 D_L を 1000 Byte、TCP のコネクション数 N_{TCP} を 10 とし、1 回のシミュレーションで再送なく受信したすべての L2 フレームに

表 5.1 シミュレーション条件

衛星回線速度	6.3 Mbit/s
衛星回線の伝搬遅延時間	0.25 sec
衛星回線ビット誤り率	10^{-8} , 10^{-7} , 10^{-6} , 10^{-5}
誤りの発生	ランダム
SR ARQ プロトコル	SSCOP
SSCOP のポーリング時間間隔	0.2 sec
TCP タイプ	NewReno
TCP パケットのデータ長 D_L	64, 1000, 1500 Byte
ヘッダによるオーバーヘッド	50 Byte
TCP ウィンドウサイズ	50
TCP コネクション数 N_{TCP}	1, 2, 5, 10, 20, 50, 100
TCP によるパケットの発生	送信すべきデータは常に存在
シミュレーション時間	60 sec
シミュレーション試行回数	10 回

ついて、各 L2 フレームの差分ポインタ値を横軸、各 L2 フレームの順序制御による遅延時間を縦軸にプロットしている。同一点での重なる頻度に関する情報は得られないが、点の分布からポインタの値と順序制御による遅延時間について概略の相関を把握することができる。これらの図から、5.4.1 で考察したように差分ポインタが大きな値をとる頻度が小さい傾向、また差分ポインタが大きな値をとる場合は順序制御による遅延時間が小さくなる傾向が見られる。特に差分ポインタの値が 500 を超えると順序制御による遅延時間はほとんど発生せず、ビット誤り率が 10^{-6} から 10^{-5} に増大しても、遅延時間自体は大きくなるもののこれらの傾向は大きく変化しない。図 5.11 では最大 2 回、図 5.12 では最大 4 回程度の再送が発生している。シミュレーションの条件では、L2 フレームを送信してから応答が返るまでのラウンドトリップ時間は 0.5sec で、この間に約 400 個の L2 フレームの送信が可能である。4 回の再送が発生する場合、最大で 2000 個程度の L2 フレームが送達確認待ちとなる可能性がある。ポインタに上限を設けない場合、最低でもこの程度の値を指定するのに十分なビット数が必要となる。これに対し、図 5.11 と図 5.12 は、ポインタの値の範囲として 500 程度までであれば十分である可能性を示している。

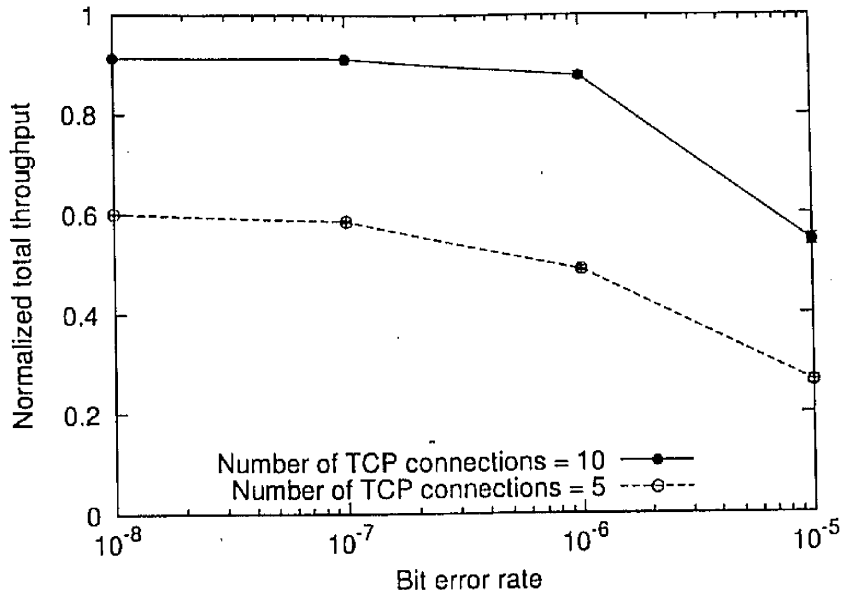


図 5.10 ビット誤り率とスループットの関係 ($D_L = 1000$ Byte, $N_{TCP} = 5, 10$)

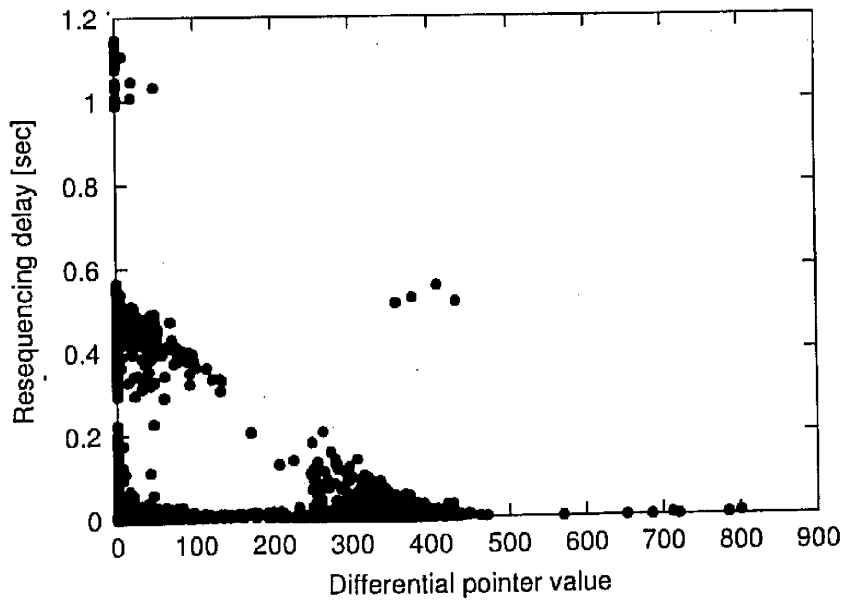


図 5.11 差分ポインタと順序制御による遅延時間の相関図 (ビット誤り率 = 10^{-6} , $D_L = 1000$ Byte, $N_{TCP} = 10$)

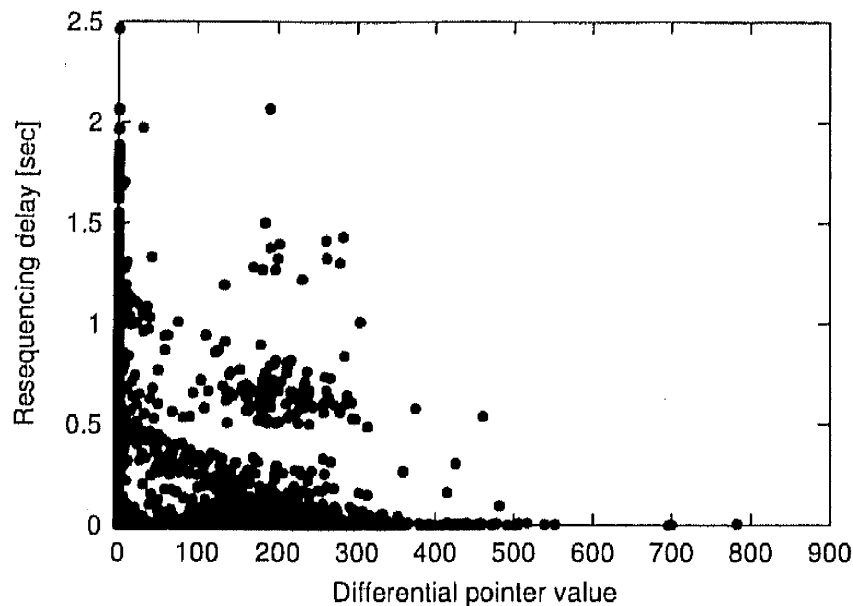


図 5.12 差分ポインタと順序制御による遅延時間の相関図 (ビット誤り率 = 10^{-5} 、 $D_L = 1000$ Byte、 $N_{TCP} = 10$)

5.4.3 差分ポインタに上限を設けた場合の性能

差分ポインタの値に上限を設ける場合、上限を超えた場合の扱いが課題で以下が考えられる。

方式 A1: 上限を超える L2 フレームの場合は、差分ポインタの上限値が指す L2 フレーム及びそれ以前の L2 フレームがすべて受信された条件で、その L2 フレームが運ぶパケットを上位層に転送する。

方式 A2: 上限を超える場合は順序制御を行わず受信 L2 フレームが運ぶパケットを上位層に転送する。

方式 A は、L2 フレームの順序逆転をできる限り起こさない方針で、差分ポインタによる距離が離れていれば既に受信されている確率が高く、全てのパケットの順序を保存する全順序制御を行ったとしても遅延時間はそれほど大きくならないとの考え方によるものである。方式 B は、L2 フレームの順序逆転が発生する場合はあるが、差分ポインタによる距離が離れていれば既に受信されている可能性が高く、順序逆転発生の確率は小さいとするものである。

図 5.13 は、ビット誤り率が 10^{-6} の場合について差分ポインタの上限値を変化させた場合のスループット性能を示している。ここで、TCP のコネクション数 N_{TCP} を 10、データ長 D_L を 1000 Byte とし、図では 95% 信頼度区間も併せて示している。方式 A も B も差分ポインタの上限値が 500 以上では性能がほとんど変わらず、ほぼ上限を設けない場合の PFRS 方式の性能が得られている。

差分ポインタの上限値が小さい場合 (1~100)、方式 B は方式 A よりもかなり性能が悪い。この理由はかなりの頻度でパケットの順序逆転が発生するためと考えられる。TCP では順序の逆転したパケットを受信すると次に正常に受信を期待するセグメント通番の ACK を返送する。複数の順序逆転パケットを連続して受信すると同じセグメント通番の重複 ACK が連続して返送されることになる。TCP の送信側ではこのような重複 ACK を 3 回連続して受信するとパケットが失われたものとして再送を行い、TCP Reno や NewReno では同時に輻輳ウィンドウの大きさを半分にする。TCP による無効な再送とともに、輻輳ウィンドウの半減は伝搬遅延時間の大きい衛星通信では大幅なスループットの低下を招き、これが方式 B の性能が低下する理由である。

差分ポインタの上限値が小さい場合 (1~100)、方式 A のスループットも低いが、これは差分ポインタの上限が小さいと全順序制御方式の動作に近くなり、5.2.1 で述べた HOL ブロッキングが発生して、順序制御による遅延時間が大きくなることによる。TCP のスループットはセグメントを送信してから ACK を受信するまでのラウンドトリップ時間の間にウィンドウサイズ分のセグメントを送信することにより決まる。順序制御による遅延時間が大きくなるとラウンドトリップ時間が大きくなり、ウィンドウサイズは一定としているのでスループットが低下する。方式 A の性能は差分ポインタの上限値が 50 を超えると徐々に改善されるが、これはフロー単位に順序制御を行う PFRS 方式の効果によるものと考えられる。

方式 B において差分ポインタの上限値が 100 を超えると急にスループットが改善され、上限値が 300 では方式 A より若干よくなる。これは方式 B で順序逆転パケットが発生するものの、その発生頻度が小さくなり、3 回の重複 ACK による再送発生の確率が小さくなるためと考えられる。図 5.13 からは、データ長 D_L が 1000 Byte の条件でポインタの上限値を 300~500 とするのであれば、方式 B をとる方が性能がよいといえる。実現する上で、方式 B は同じフローに属する直前パケットを運ぶ L2 フレームの通番との差が上限に達した場合に、差分ポインタの値を 0 (順序制御の必要なし) とするだけでよく送信側のみの処理で容易に実現できる。一方、方式 A の実現は上限値が指す L2 フレーム及びそれ以前のすべての L2 フレームが受信されるまで、L2 フレームを保留する処理の追加が受信側に必要となる。

表 5.2 差分ポインタに上限を設けない場合と比較した TCP の相対スループット
(ビット誤り率 = 10^{-6} 、 $D_L = 1000$ Byte)

N_{TCP}	Scheme A1			Scheme A2		
	Maximum pointer value			Maximum pointer value		
	300	400	500	300	400	500
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.9922	0.9991	1.0000	1.0000	1.0000	1.0000
10	0.8135	0.9429	0.9958	0.8429	0.9999	1.0000
20	0.9833	0.9940	0.9947	0.9999	1.0005	1.0002
50	0.9915	0.9930	0.9930	0.9999	1.0000	0.9998
100	0.9937	0.9962	0.9968	1.0007	1.0000	1.0003

表 5.2 はビット誤り率を 10^{-6} 、データ長 D_L を 1000 Byte、差分ポインタの上限値を 300、400、500 とした場合について、TCP のコネクション数 N_{TCP} を変化させた場合のスループットを示している。ここで、差分ポインタの上限を設けない場合のスループットとの相対値を示している。方式 A も B も TCP のコネクション数が少ない 1、2、5 では、上限値 300 で十分な性能が得られるが、TCP のコネクション数 N_{TCP} が 10 になると性能が低下し、特に方式 A で著しい。TCP のコネクション数 N_{TCP} が 10 で性能が一番劣化するが、10 より大きなコネクション数で劣化が少なくなる原因は次のように考えられる。TCP のコネクション数 N_{TCP} が 10 までは TCP のコネクション 1 本当りのスループットは変わらずに全体のトラフィックが増加するが、10 を超えると回線の帯域が飽和し TCP コネクション 1 本当りのスループットが低下する。このため順序制御による遅延時間やパケットの順序逆転の影響が少なくなる。方式 B では上限値を 400 とすれば TCP のコネクション数 N_{TCP} が変化してもほぼ上限を設けない場合と同等の性能が得られる。方式 A はポインタの上限値を 400 とすると TCP の性能が約 5% 低下する場合があり、上限値を 500 にするとほぼ上限を設けない場合と同等の性能となる。

シミュレーションで用いた SSCOP は 24 ビットの通番を用いており、5.2 で説明したポインタ方式では 24 ビットのポインタが必要であるが、差分ポインタを用いるとデータ長が 1000 Byte の場合は 9 ビットで十分な性能が得られ、ポインタのオーバーヘッドを削減できる。

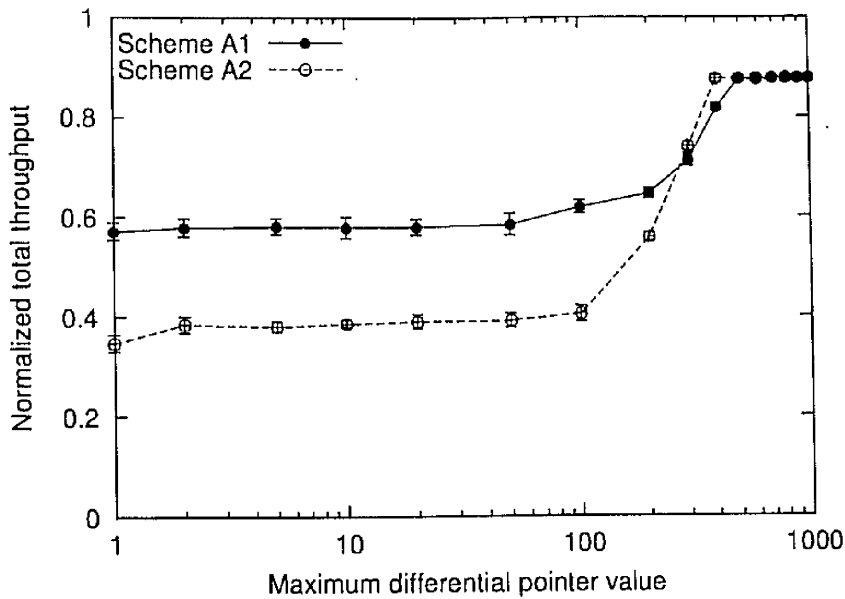


図 5.13 差分ポインタの値とスループットの関係 (ビット誤り率 = 10^{-6} 、 $N_{TCP} = 10$ 、 $D_L = 1000$ Byte)

5.4.4 データ長を変えた場合の差分ポインタの上限値

以上は、パケットのデータ長が 1000 Byte の場合について説明した。図 5.14 はパケットのデータ長 D_L を 64 Byte 及び 1500 Byte とした場合について差分ポインタの値とスループットの関係を示している。ここで、ビット誤り率は 10^{-6} 、TCP の接続数 D_L は 1500 Byte の場合に 10、64 Byte の場合に 36 とし、図では 95% 信頼度区間も併せて示している。また 64 Byte の場合に TCP のウィンドウサイズを 100 としている。64 Byte の場合に TCP の接続数と TCP のウィンドウサイズを増加させているのは、回線の帯域をフルに利用する条件とするためである。またスループットは正味のデータ分で測定しているため、データ長が短くなるとパケットや L2 フレームのヘッダによる相対的なオーバーヘッドが増加し、スループットが低下する。図から、データ長 D_L を大きくするとポインタの上限値はより小さくできること、逆にデータ長 D_L を小さくするとポインタの上限値は大きくする必要のあることが分かる。データ長 D_L が 1500 Byte の場合も 1000 Byte の場合と同じく、差分ポインタ値が小さい場合では方式 A のスループットが方式 B より大きく、差分ポインタの上限値がある値を超えると急速に方式 B のスループットが改善され、方式 A より小さな差分ポインタの上限値でスループットが最大に達する。データ長 D_L が 64 Byte の場合、ポインタの上限値が小さい場合に方式 A

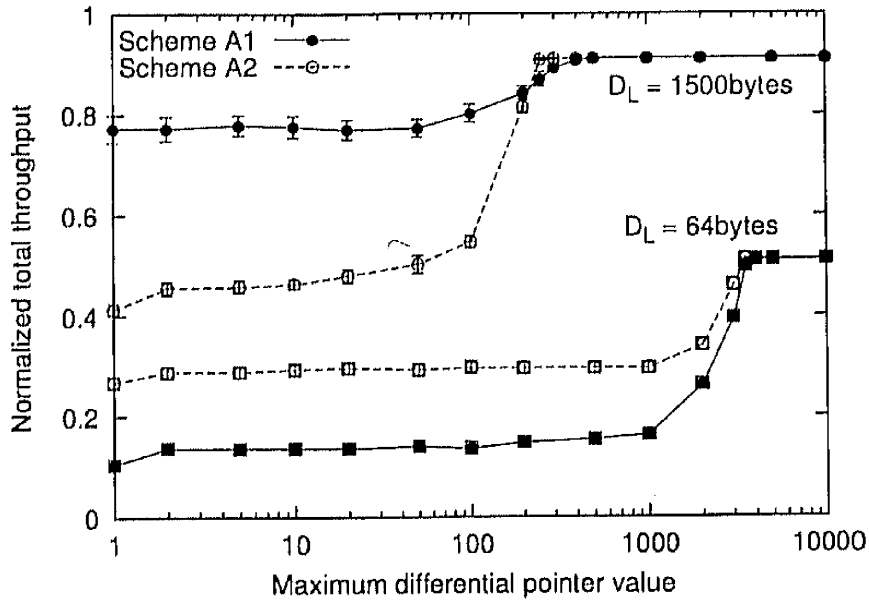


図 5.14 パケットのデータ長 D_L を 64 Byte と 1500 Byte とした場合の差分ポインタの値とスループットの関係 (ビット誤り率 = 10^{-6} 、 $N_{TCP} = 10$ ($D_L = 1500$ Byte)、 $N_{TCP} = 36$ ($D_L = 64$ Byte))

の性能が方式 B より悪い結果となっており、データ長 D_L が 1000 Byte や 1500 Byte の場合と異なっている。5.4.3 で説明したように差分ポインタの上限値が小さい場合に方式 A では順序制御による遅延時間が大きくなるが、データ長 D_L が 64 Byte の場合はこの遅延時間によりタイムアウトが頻繁に発生している。タイムアウトが発生すると、TCP のスロースタートにより TCP の輻輳ウィンドウの大きさが 1 にリセットされる。ポインタの上限値が小さい場合に、方式 B においても頻繁に順序逆転が生じ無効な高速再送が行われるが、この場合は輻輳ウィンドウサイズがそれ以前の値の半分となる。データ長 D_L が 64 Byte でポインタの上限値が小さい場合は、方式 B の輻輳ウィンドウが平均的に大きくなり方式 A より高いスループットが得られるものと考えられる。このように、上限値が小さい場合の方式 A と方式 B の優劣は一概にはいえないが、どのデータ長においても方式 B の方が小さな差分ポインタの上限値でスループットが最大に達する傾向が見られる。

データ長 D_L が 64 Byte の場合は、差分ポインタに上限を設ける点では最悪の条件と考えられるが、この場合でも本論文で評価した条件では差分ポインタの上限を 4000 程度とすれば、上限を設けない場合と同等のスループットが得られるといえる。これは差分ポインタとして 12 ビット程度を用いればよいことを示しており、SSCOP の通番の半分のビット数で済むことになる。

5.5 むすび

この章では、SR ARQ において順序制御を上位層のフローごとに行う PFRS 方式について実現方法を検討し、SR ARQ の通番を利用するポインタ方式を提案した。このポインタ方式では SR ARQ 上で多重化されるフロー数の上限がなくフローごとに順序制御を実現できる特長がある。また SR ARQ の通番の差分を用いるポインタ方式について検討し、値の大きい差分ポインタの頻度が小さくかつ順序制御による遅延時間が短くなる傾向のあることを示した。これらを利用することにより、差分ポインタ値に上限を設けて差分ポインタによるオーバーヘッドを減少させることが可能であることを示した。また上限を超える場合の処置として順序制御を行わない方式と上限値より前の L2 フレームをすべて受信するまで保留する方式を比較し、順序制御を行わない方式の方が小さな上限値で済むこと、また実現が容易であることを示した。

衛星通信の場合について評価を行ったが、SR ARQ のポインタを用いた PFRS 方式はレイヤ 2 において選択再送型のプロトコルを用いる場合に有効であり、この種のプロトコルが必要となる高速の地上無線通信にも適用できると考えられる。

第 6 章

TCP の順序制御を用いる SR ARQ 方式

6.1 まえがき

一般にリンク単位でローカルな再送を行うリンク層（レイヤ 2）プロトコルは、入力されたパケットの順序を保存するという原則に基づいて設計されてきた。図 6.1 は SR ARQ によるリンク層プロトコル上に TCP のパケットが送信される場合の従来のシーケンス例を示している。パケットが伝送誤りにより失われた場合、SR ARQ がこのパケットを再送し、更に後続パケットで正しく受信したものを順序を保存するために保留する。これまで説明したように、SR ARQ は伝送効率の点で優れているが、このような順序制御は保留された各パケットに遅延時間を発生させる。このような問題の他に、再送されたパケットが正しく受信されると、このパケットの受信を待って保留されていたすべてのパケットが解放され同時に出力される。従って、SR ARQ はバースト的にパケットを出力する。このようなパケットのバーストは、中継ノードでのバッファあふれや帯域を共有するフローの QoS の劣化などの望ましくない影響を与える。

4 章で説明した PFRS 方式は、フロー単位でパケットの順序を保証することを基本としている。逆にフロー間ではパケットの順序を保証しないことにより HOL ブロッキングを回避することが可能となっている。

この章ではこのようなアプローチとは全く別の方向で検討する。一般に、IP ネットワークでは信頼性を必要とする通信では TCP を用いる。初期の TCP はパケットの順序を保存しないデータグラムサービスを前提に設計が行われており、順序の乱れたパケットを受信して順序制御を行う能力を有している。もし、TCP の順序制御機能をリンク層でのパケット喪失と再送に対する順序制御に利用できれば、リンク層での順序制御を省くことが

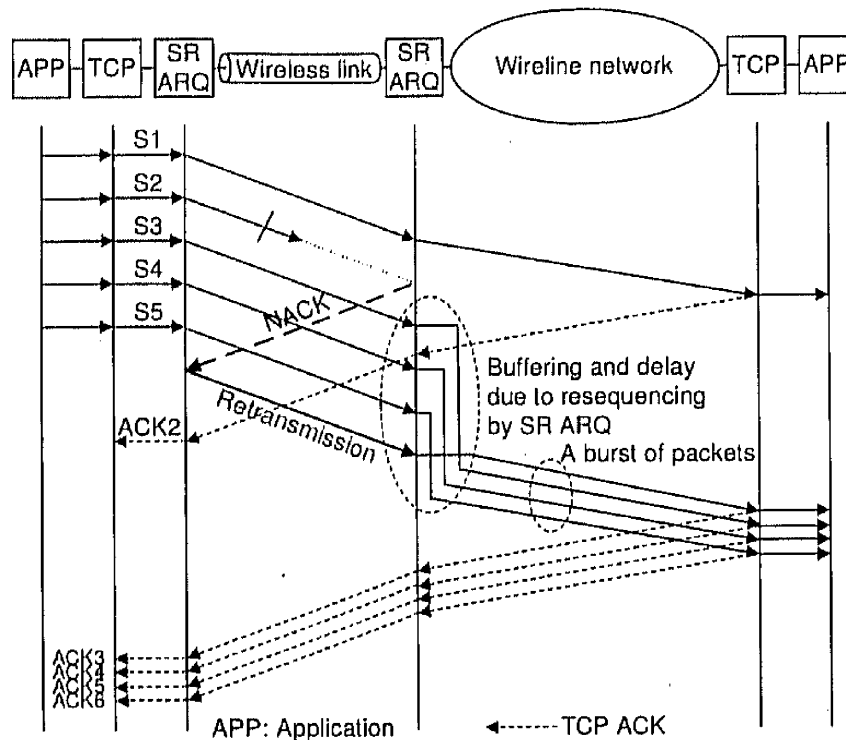


図 6.1 SR ARQ によるリンク層プロトコルが用いられた場合の TCP データパケットの転送例

可能となる。これからリンク層での順序制御により引き起こされる種々の問題を回避できる可能性がある。このような背景から、この章では SR ARQ を行うリンク層プロトコルで順序制御を省略し、TCP により順序制御を行う方式を提案する。

具体的なリンク層プロトコルとして、本研究では特定のプロトコルを仮定していない。想定される具体的な SR ARQ プロトコルの例としては、W-CDMA 移动通信 [11] における RLC[52] や一般のデータ通信分野では SREJ を用いる HDLC[72] などが挙げられる。本研究では後で述べるシミュレーションにおいて、プロトコルの動作が比較的シンプルで通番の空間が大きい SSCOP を用いている [65]。

本章の構成は以下のようになっている。6.2 節ではリンク層プロトコルによるローカルな再送に関する関連研究について述べ、6.3 節ではリンク層での順序制御に伴う問題を解決する方式を提案する。6.4 節でシミュレーション条件とシミュレーション結果及び関連する考察を述べ、最後に結論を 6.5 節でまとめる。

データの単位はそれを扱うプロトコル層によりセグメント、パケット、フレームなど一般に用語が使い分けられる。本章では用語の混乱を防ぐため、プロトコルの層によらずパケットの用語に統一する。

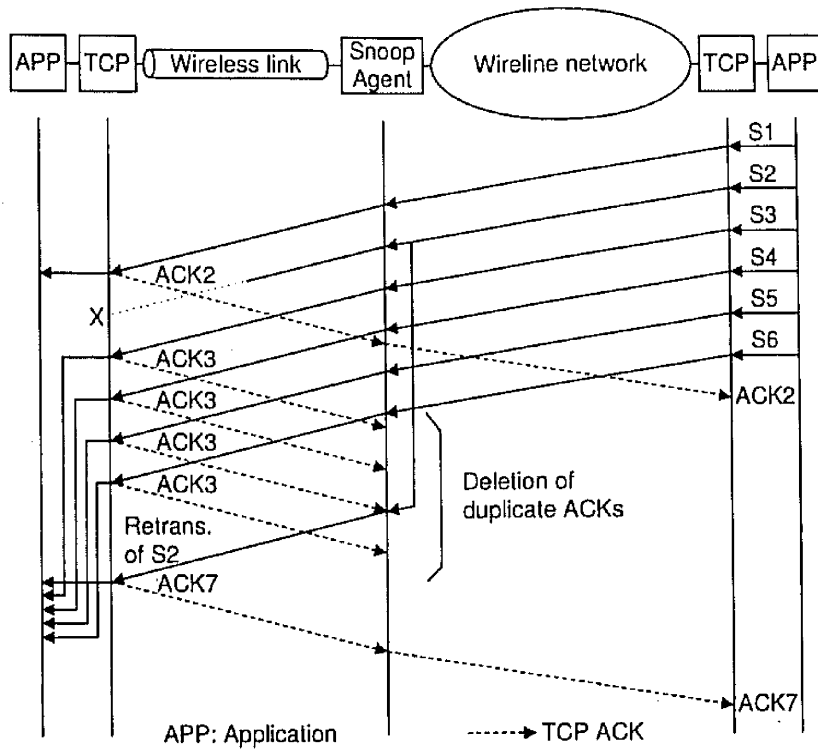


図 6.2 Snoop 方式のシーケンス例

6.2 関連する研究

リンク層プロトコルによるローカルな再送と TCP の性能に関しては多くの研究が行われている [62] [85]。Snoop はローカルな再送により無線リンクにおけるパケットの喪失を上位層の TCP から隠蔽する方式である [61]。Snoop は無線リンクにおけるパケットの喪失を TCP の重複 ACK により検出する。また TCP パケットのコピーを保留し、パケットの喪失を検出した場合は、喪失したパケットのコピーを再送する。図 6.2 に Snoop 方式のシーケンス例を示す。

Snoop では喪失パケットの再送を行うだけなので、宛先端末では順序が逆転したパケットを受信する。これにより宛先端末は不必要な重複 ACK を返送するが、これを防ぐため Snoop では重複 ACK の除去を行う。先に述べたように Snoop はパケット喪失を検出するのに重複 ACK を利用するが、ローカルな再送のために最適化されている一般的なリンク層プロトコルに比べ検出に時間を要する問題がある。また、再送されたパケットが再度失われた場合、Snoop は重複 ACK を用いて喪失を検出することができず、TCP のタイ

ムアウトによる再送が必要となる。一般にリンク層プロトコルはこのような場合でも効率的に再送を行うよう設計されている。

Snoopの再送は基地局から端末方向に向かうパケット転送に限られる。逆方向の転送の場合、TCPの送信端末に対しローカルな否定応答を送ることが必要となる。このためには端末側でSACK[76]を用いたTCPが必要となる制約がある。

ローカルな再送に関する別の研究としてDelayed Duplicate Acknowledgments (DDA) [77]がある。この方式では再送を行うリンク層プロトコルを用いるが、順序制御は行わない。この研究ではリンク層プロトコル自体について詳細は明らかにしていないが、Snoop相当のプロトコルでTCPを意識しない方式を目的としている。この方式ではTCPの受信側でACKの返送を遅らせ、不必要な再送を発生させる重複ACKの送信を抑える。この方式はリンク層プロトコルにおいてTCPを意識した処理を行う必要はないが、重複ACK（正確には三番目及びそれ以降の重複ACK）を送信するか否かを判断するタイミングの決定が困難な問題がある。また、この機能をTCPに追加するためTCPの受信側に関して修正が必要となる問題もある。更に、この方式ではネットワークの有線部分で輻輳によりパケットの喪失が発生した場合、輻輳制御の起動が不必要に遅れる問題もある。

半二重無線通信リンクに適したリンク層プロトコルとして、TULIPが提案されている[78][79]。このプロトコルは通番とビットベクタにより選択再送を行うSR ARQを実現している。通番は累積応答を実現し、ビットベクタのビットは送信済の各パケットに対応し、パケットが正しく受信されたか否かを識別する。この方式はTCPのデータパケットの順序を保存するので、SnoopやDDAのように重複ACKが発生する問題はなく、またTCPを意識した処理も必要ない。この方式はTCPのデータパケットの順序を完全に保存するので、全順序制御を行う従来のSR ARQと等価な方式と考えられる。TULIPは伝搬遅延時間の短い無線ネットワークを想定しており、伝搬遅延時間が長く送達確認待ちのパケット数が多い場合には6.3.4で述べたようにHOLブロッキングなど従来方式と同様の問題があるものと考えられる。

上で述べたすべての方式は伝送品質の悪い無線リンクでローカルな再送を用いている。しかし、PFRS方式を除きどの方式も遅延時間の小さいリンクを主に想定している。従って、帯域と遅延時間の積が大きな環境では、性能が悪いかプロトコルのメカニズムが適していないものと考えられる。以下では、伝送誤りが多くかつ遅延時間と帯域が大きな環境におけるリンク層プロトコルアーキテクチャを提案する。

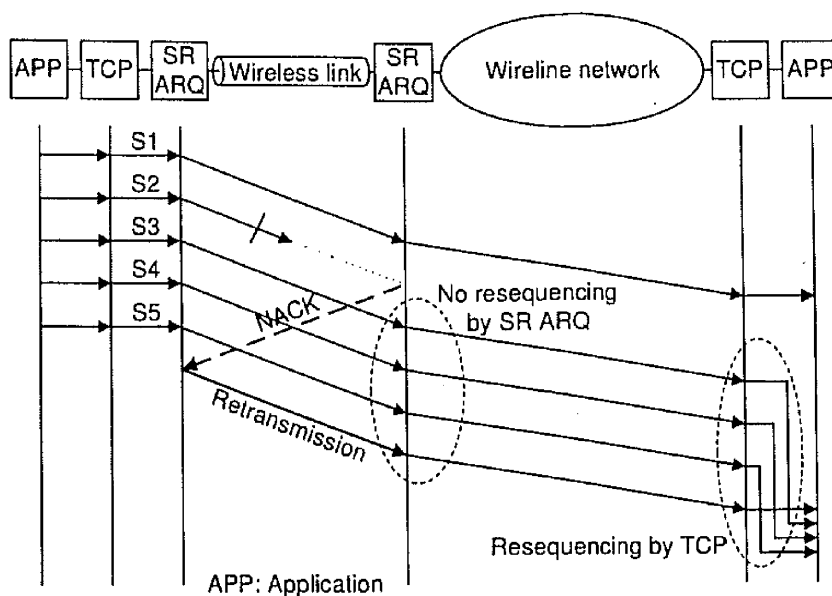


図 6.3 提案方式の基本アーキテクチャ

6.3 順序制御を行わない SR ARQ に基づくリンク層プロトコル

6.3.1 基本方式

提案プロトコルの基本的な方式を図 6.3 に示す。図を見やすくするため受信側からの ACK を省略している。提案プロトコルでは、パケット喪失の検出と再送は従来の SR ARQ と同様に行うが、SR ARQ の受信側で順序制御は行わない。伝送誤りによるパケット喪失により必要となるパケットの順序制御は TCP によりエンドツーエンドで行われる。

一般に TCP の受信側では順序制御に必要な受信バッファを確保し、このバッファ量を告知ウィンドウサイズ (advertising window size) として TCP の送信側に通知する機能をもつ [80]。TCP の送信側では輻輳ウィンドウサイズとこの告知ウィンドウサイズの両方を超えないようにパケットを送信する。順序制御に必要なバッファ量は TCP の受信側で予め確保されているので、提案方式はこれを利用しようとするものである。

6.3.2 SR ARQで順序制御を行わないことにより派生する問題

SR ARQで順序制御を行わない場合、TCPの受信側でパケットの順序逆転が発生する。図6.4に示すように順序逆転パケットは重複ACKを発生させる。図において受信側TCPはパケットS2の受信を期待するが、順序逆転パケットS3、S4、S5を受信している。これらの順序逆転パケットを受信するたびに、受信側TCPは重複ACK2を返送する。ここでACKの通番である値2は次に正常に受信を期待するパケットの通番である。送信側TCPは一定数(通常は3)の重複ACKを受信すると、高速再送(fast retransmit)を行い重複ACKの通番のパケット(図の場合はS2)を再送する。この図の場合、リンク層プロトコルによりパケットS2は既に再送されているので、TCPは必要のない重複した再送を行っている。この重複して再送されたパケットは受信側TCPにより除去される。高速再送の後、送信側TCPは高速リカバリ(fast recovery)を行い、輻輳ウィンドウサイズ(cwnd)をそれ以前の半分の値に減少させる。高速リカバリの後、パケットの正常な送受信により輻輳ウィンドウサイズは増加するが、その増加は一般に緩やかでTCPのスループットは輻輳ウィンドウサイズの制約により制限される。

6.3.3 重複ACKによる再送問題の解決方式

上記の重複ACKによる不必要な再送問題を解決する一つの方法としては、TCPが順序逆転パケットに耐えられるようTCPを修正する方向が考えられる[104][105]。先に説明したように、DDAはこのアプローチをとっているが[77]、既存TCPの修正と置き換えが必要であることそして6.2節で説明したようにDDA自体にも解決すべき問題が存在している。ここでは、TCPを意識した(TCP-aware)リンク層プロトコルとなるが、既存のTCPを変更しない方式を提案する。Snoopと同様に、提案方式ではリンク層プロトコルがリンク層で喪失したパケットにより生じる重複ACKを削除する。この方式のシーケンス例を図6.5に示す。

有線ネットワークでの輻輳により発生するパケットの喪失に対応するため、エンドツーエンドのTCPで行われる高速再送は保存する必要がある。このため、重複ACKの除去は無線リンクで発生したパケット喪失により発生する重複ACKの場合にのみ行う必要がある。無線リンクで再送を行うSR ARQはどのパケットが無線リンクで喪失し再送されたかを把握しているので、SR ARQは重複ACKが無線リンクでの喪失で発生したものか有線ネットワークでのパケット喪失により発生したものを判別できる。

このような機能を実現するために、SR ARQの送信側は再送されたパケットについて、そのフロー識別子とTCPの通番を記録するテーブルを用いる。ここでフロー識別子は

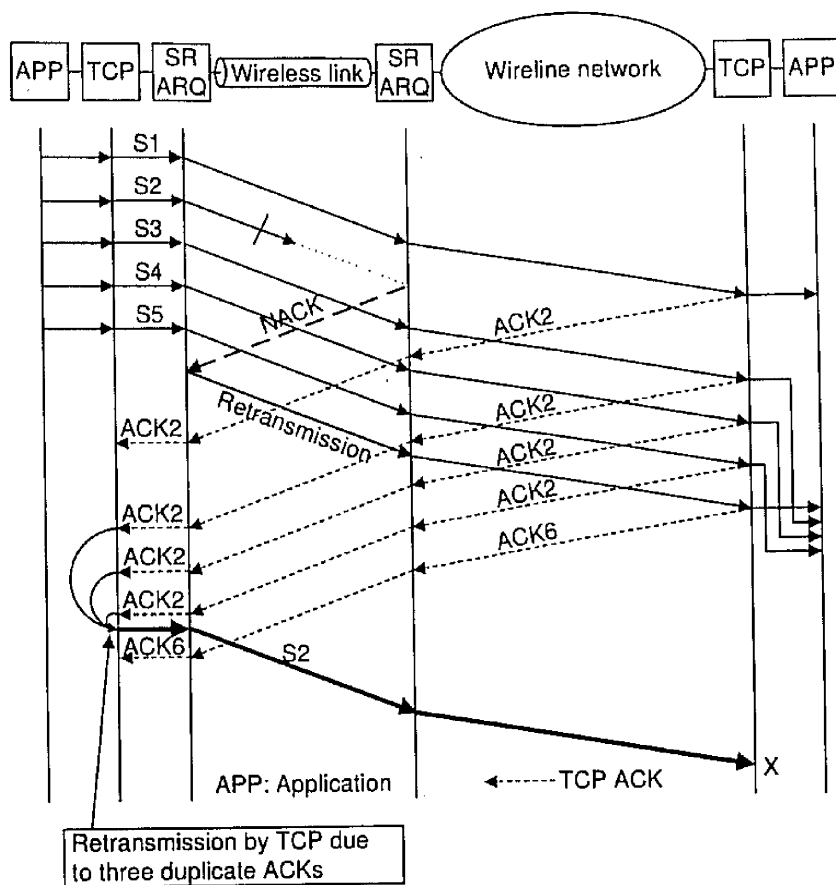


図 6.4 SR ARQ で順序制御を行わないことによる問題点

送信元及び宛先アドレス、送信元及び宛先ポート番号、そしてプロトコル ID から構成される。

図 6.6 は処理フローの概要を示している。SR ARQ の送信側がパケットを再送するたびに、再送パケットのフロー識別子と TCP の通番をテーブルに記録する。SR ARQ の送信側が重複 ACK を受信した場合、もし重複 ACK のフロー識別子と ACK の通番がテーブルに登録されていれば重複 ACK を除去し、登録されていなければ重複 ACK を転送する。

有線ネットワークで輻輳が発生しパケットが喪失した場合、SR ARQ の送信側は重複 ACK を受信するがそのフロー識別子と TCP の通番はテーブルに登録されていないので、重複 ACK は通常のように転送されエンドの TCP で通常の高速再送と高速リカバリが実行される。このように、この方式では SR ARQ の送信側が再送される TCP のデータパケットを把握することが前提となっている。上記の方式で重複 ACK の除去は、通番が進

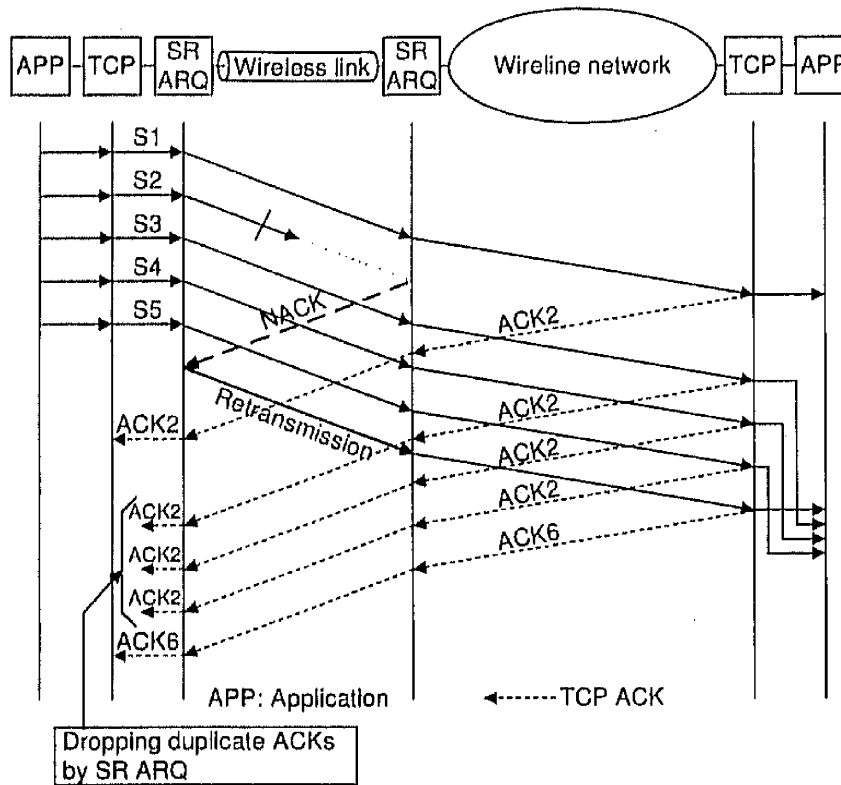


図 6.5 重複 ACK の削除を行う提案方式

んだ ACK を受信することにより自動的に終了する。

SR ARQ によっては最大再送回数が制限される場合がある。もし、パケットが最大回数の再送を行った後に正しく受信されない場合、このパケットは SR ARQ で回復されないことになる。また、SR ARQ による再送により正しく受信されたパケットが、有線部分のネットワークで輻輳により喪失される場合もある。これらの場合、TCP の送信側でタイムアウトが発生しエンドツーエンドで再送が行われる。この TCP により再送されたパケットが受信側 TCP で正しく受信されると、通番が更新された新しい ACK が TCP の送信側に返送される。これにより、SR ARQ の送信側で行っていた重複 ACK パケットの除去は終了する。

SR ARQ が TCP のデータパケットを再送している間に TCP のタイムアウトが発生すると、TCP の送信側は重複する TCP のデータパケットを再送することになる。SR ARQ はこれを受信して通常のパケットと同様に送信する。SR ARQ が最初のパケットを送信中に、TCP により重複して再送されたパケットが先に正しく受信されると、通番の更新された新しい ACK が返送されるので、SR ARQ の送信側における重複 ACK の除去は終了する。もし、TCP により重複して再送されたパケットが無線リンクでの伝送誤

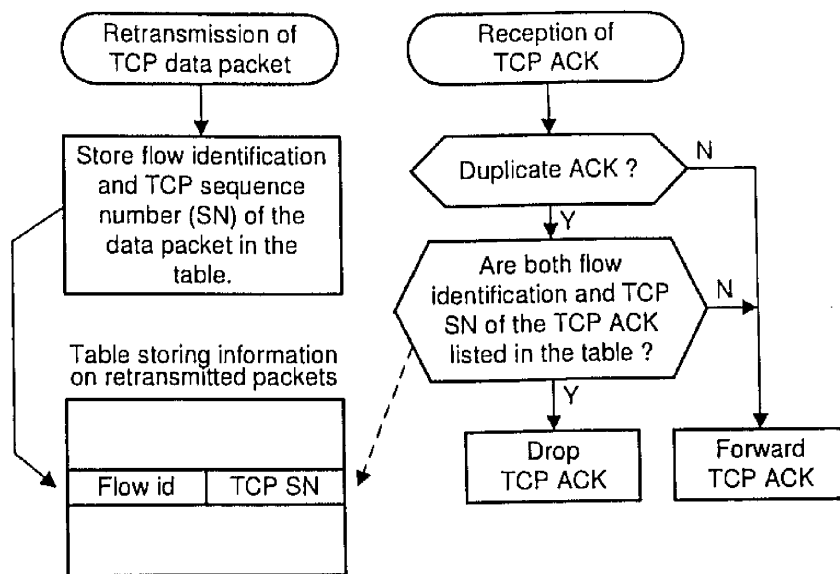


図 6.6 重複 ACK 除去処理の概要

りにより喪失した場合、SR ARQ の送信側は喪失した重複再送パケットのフロー識別情報と通番をテーブルに登録する。このパケットの情報は既に登録されているので、テーブルに同じ情報を上書きすることになる。最初のパケットか重複して再送されたパケットのどちらかが正しく受信されると、TCP の受信側は通番が更新された新しい ACK を返送し、SR ARQ の送信側における重複 ACK の除去は終了する。

図 6.7 は有線ネットワークの両端に無線リンクが存在し、両方の無線リンクで提案方式が用いられる場合を示している。この場合、一方の無線リンクで再送されたパケットが他方の無線リンクでも失われ再送される場合がある。図 6.7 で、パケット S2 が左側の無線リンクで失われ再送されている。この間、後続のパケット S3、S4、S5 は有線ネットワークに転送され、エンドの TCP は重複 ACK2 を返送する。パケット S2 の喪失によるこれらの重複 ACK は左側の無線リンクの出口で廃棄される。この図で、左側の無線リンクで再送されたパケット S2 は右側の無線リンクに転送され、ここでも伝送誤りにより失われるため再送が行われる。右側の無線リンクで再送が行われる前に、後続のパケット S6、S7、S8 が転送されるが、エンドの TCP は重複 ACK を返送し続け、これらの重複 ACK は右側の無線リンクの出口で廃棄される。右側の無線リンクでパケット S2 の再送が完了すると、TCP の送達確認情報が更新され新しい ACK9 が返送される。この新しい ACK9 により両方の無線リンクにおける重複 ACK 除去が終了する。このように、有線ネットワークの両端で順序制御を行わない SR ARQ が用いられる場合でも提案方式による重複 ACK 除去は問題なく動作する。

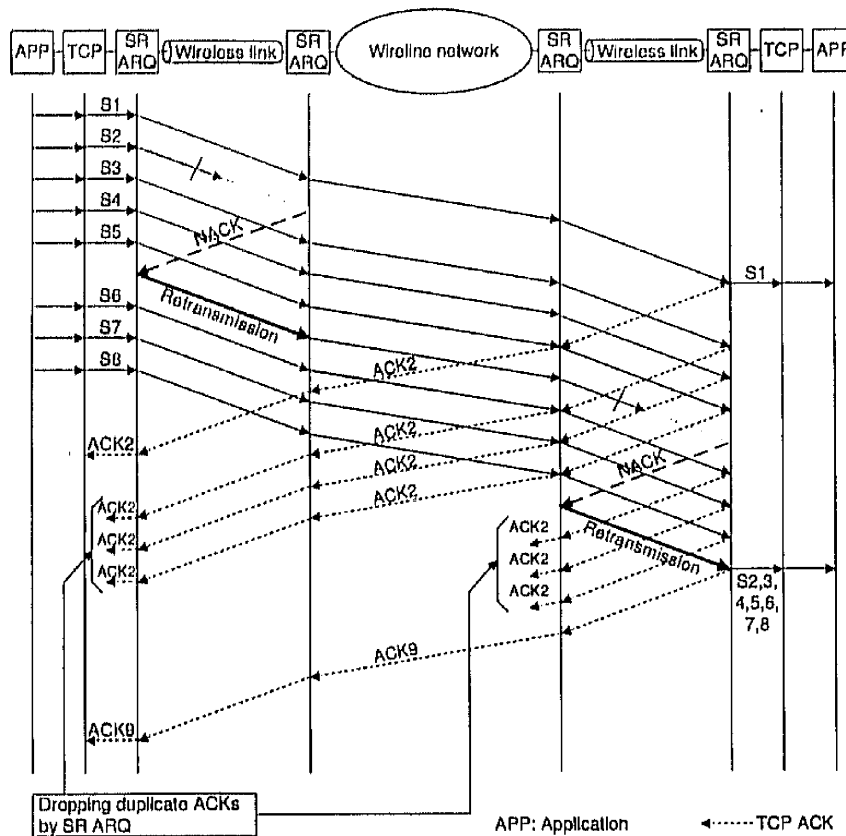


図 6.7 有線ネットワークの両端に無線リンクが存在し、両方の無線リンクで提案方式が用いられる場合

6.3.4 提案方式の特長

提案方式は SR ARQ の受信側において順序制御用のバッファを必要としない大きな利点を有する。従来の SR ARQ 方式では、SR ARQ の受信側における順序制御用バッファの所要量は以下のパラメータが大きくなると顕著に増大する。

- リンクの通信帯域と遅延時間の積
- SR ARQ に多重化される TCP のコネクション数
- TCP のウィンドウサイズ

TCP のコネクション数と TCP のウィンドウサイズはエンドの端末に依存するため予測することは困難で、従来の SR ARQ において順序制御用の所要バッファ量を規定することは一般には不可能である。一方、提案方式では順序制御用のバッファはエンドの端末

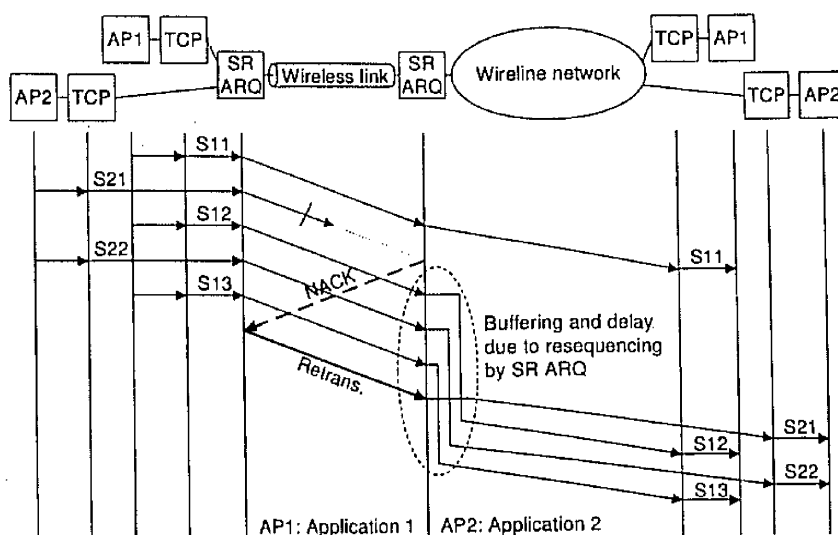


図 6.8 従来方式における HOL ブロッキング

が確保しており、SR ARQ ではバッファは不要なので TCP のコネクション数や TCP のウィンドウサイズに対しスケラビリティを有する大きな利点がある。また、SR ARQ がバースト的なパケット出力を行わない利点もある。

更に、多数のフローを SR ARQ 上に多重化する際に生じる HOL ブロッキングの問題も発生しない利点も存在する。図 6.8 は HOL ブロッキングによる問題の例を示している。ここで無線リンク上に設定された SR ARQ 上に 2 本の TCP コネクションが多重化されている場合を想定している。従来の全順序制御方式では SR ARQ を流れるすべてのパケットの順序が保存されるので、パケット S12 と S13 はパケット S21 が受信されるまで SR ARQ の受信側で保留される。この場合、パケット S21 と S22 はパケット S12 及び S13 とは異なる TCP のコネクションに属するので、順序制御のためにパケット S12 と S13 を保留する必要はない。もし、SR ARQ による順序制御が行わなければ、この HOL ブロッキングは発生せずパケット S12 と S13 は遅延を被ることなく転送される。図 6.9 に示すように、提案方式では SR ARQ で順序制御を行わず TCP の順序制御を利用するのでこのような HOL ブロッキングが発生しない。

上記のような利点に加え、提案方式は単に重複 ACK の除去を行うだけなので、ネットワークの輻輳制御など TCP におけるエンドツーエンドの通信を保存する特長も有する。

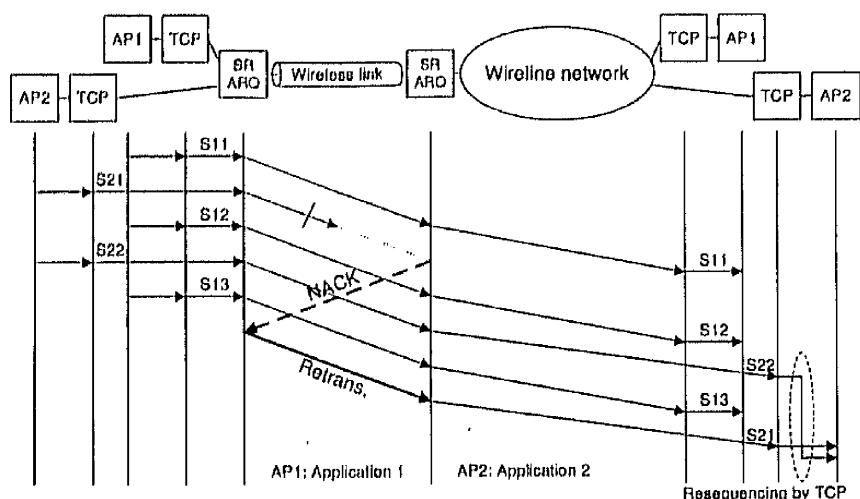


図 6.9 提案方式により HOL ブロッキングが解消される例

6.4 シミュレーションによる提案方式の評価

6.4.1 シミュレーション構成

提案方式の性能を評価するためシミュレーションを行った。ネットワークシミュレータは VINT プロジェクトにより開発された ns-2 (ns-2.27) を用いた [63]。図 6.10 はシミュレーションしたネットワークの構成を示している。3 ノードが有線リンクと無線リンクにより接続されている。これらのリンク上に複数の TCP コネクションが設定されている。シミュレーション条件を表 6.1 に示す。伝搬遅延時間と無線リンクのビット誤り率は表に示す通りである。無線リンク上で用いる具体的なリンク層プロトコルとしては SSCOP [65] を用いた。このプロトコルは比較的シンプルな動作で SR ARQ を実現している。TCP のウィンドウサイズは一般に用いられる値を参考に 32 KByte とした [71]。この値では、1 本の TCP コネクションがシミュレーション条件のリンクの帯域すべてを利用することはできないが、SR ARQ コネクション上に複数の TCP コネクションが存在し、これら複数の TCP コネクションによりリンクの帯域をすべて使用できる場合を想定している。また、パケットにおける伝送誤りはビット誤り率とパケット長に従ってランダムに発生するものとしている。

以下の 5 方式についてシミュレーションを行い比較した。

- Full resequencing (全てのパケットの順序を保障する全順序制御方式)
- PFRS

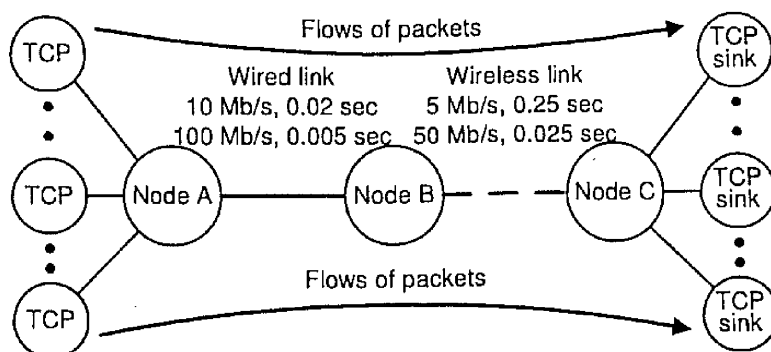


図 6.10 シミュレーションにおけるネットワーク構成

- SR ARQ with no resequencing (提案方式)
- Snoop
- No link layer ARQ (TCP によるエンドツーエンドの再送)

先に述べたように Snoop では基地局から端末方向へのパケット伝送においてパケットの喪失が発生した場合、パケットの再送を行う。このため、シミュレーションにおけるデータパケットの流れは図 6.10 に示すようにネットワークの有線部分から無線リンクに向かう方向とした。シミュレーション結果は 10 回のシミュレーション試行を行い平均を求めた。

6.4.2 シミュレーション結果

図 6.11 から図 6.12 は方式の比較を示しており、TCP のコネクション数は 5 及び 10 としている。無線リンクの帯域は 5 Mbit/s で、伝搬遅延時間を 0.25 sec とし衛星回線に相当する場合でシミュレーションを行った。

図は全体のスループットを示しており、すべての TCP コネクションのスループットの合計を無線リンクの帯域で規格化している。シミュレーション条件では、TCP のコネクション数が 5 の場合は無線リンクの全帯域を使い切ることができないが、TCP のコネクション数が 10 の場合は帯域を使い切る条件となっている。

ビット誤り率が 10^{-5} と大きい場合、比較した 5 方式の中で、PFRS 方式と提案方式の性能が最も良いことが分かる。提案方式のスループットは PFRS 方式より若干劣るが、その差は小さい。図 6.11 及び図 6.12 においても、TCP のコネクション数に係わらず同様の傾向が見られる。

ビット誤り率が 10^{-8} と小さい場合、Snoop 方式の性能は他の方式より若干優れている

表 6.1 シミュレーション条件

回線速度	5 Mbit/s, 50 Mbit/s
回線の伝搬遅延時間	0.025 s (50 Mbit/s) 0.25 s (5 Mbit/s)
回線のビット誤り率	10^{-8} , 10^{-7} , 10^{-6} , 10^{-5}
誤りの発生	ランダム
SR ARQ プロトコル	SSCOP
SSCOP のポーリング時間間隔	0.2 sec
TCP のタイプ	NewReno
TCP のパケット長	1460 Byte
TCP の ACK 遅延	遅延なし
SSCOP のオーバーヘッド	10 Byte
TCP ウィンドウサイズ	32 kByte
TCP のコネクション数	5, 10
TCP のトラフィック発生	送信データは常に存在
シミュレーション時間	60 sec
シミュレーション試行回数	10 回

る。これは Snoop の場合、リンク層プロトコルのオーバーヘッドが存在しないためである。しかしビット誤り率が大きくなると Snoop 方式の性能は PFRS 方式及び提案方式に比べ低下する。この理由は SR ARQ をベースとするリンク層プロトコルの再送能力が、重複 ACK やタイムアウトに頼る Snoop の再送能力より大きいためである。

すべてのパケットの順序を保存する従来の全順序制御方式は低い性能を示している。これは HOL ブロッキングが発生しているためである。リンク層で再送を行わず TCP によるエンドツーエンドの再送に頼る方式の性能も低い。この理由は、パケットの喪失検出と再送に時間を要すること、再送を行う場合に輻輳ウィンドウサイズを減少させることによる。

図 6.13 は無線リンクの帯域を 50 Mbit/s、その伝搬遅延時間を 0.025 sec とした場合の性能の比較を示している。ここで、有線リンクの帯域がボトルネックとなることを避けるためにその帯域を 100 Mbit/s に変更している。また帯域と遅延時間の積が図 6.12 の場合とほぼ同じとなるようにするため伝搬遅延時間を 0.0005 sec に変更している。遅延時間は短いですが、無線リンクのビット誤り率が増加するとネットワークの性能は劣化すること

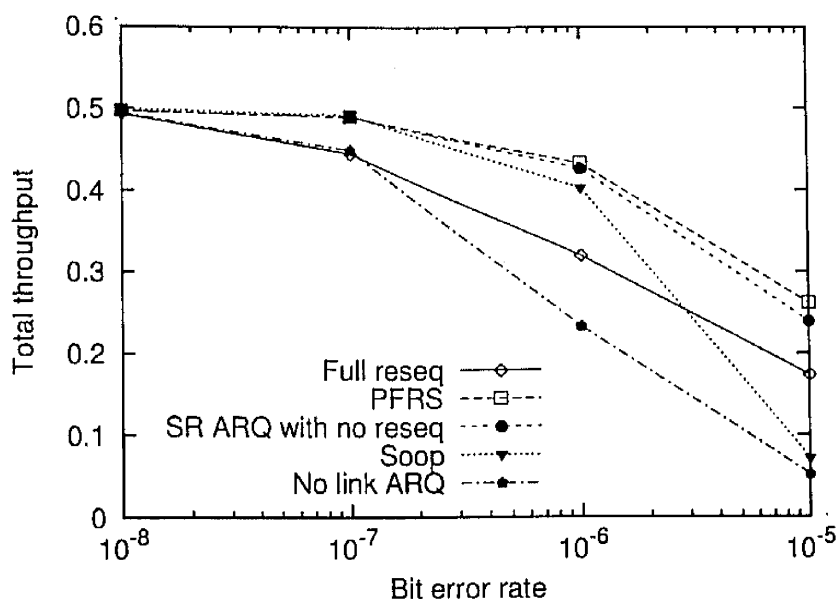


図 6.11 スループットの比較、TCP のコネクション数が 5、伝送リンクの帯域と遅延時間が 5 Mbit/s と 0.25 sec の場合

が分かる。この場合、提案方式と PFRS 方式の性能は完全に同じとなっている。これら 2 方式は他の 3 方式に比べ優れた性能を示している。

上記のシミュレーションの結果から提案方式は PFRS 方式に近い優れた性能を達成することが分った。提案方式の性能は伝搬遅延時間が大きい場合のみならず、無線リンクの伝搬遅延時間が小さいが通信速度が大きい場合にも顕著であるといえる。

6.4.3 DDA 方式との比較

前述したように DDA 方式は提案方式と類似のアプローチを採用しており、リンク層プロトコルは順序制御を行わない。DDA では TCP の受信側で重複 ACK が不必要に TCP の送信側に返ることを防ぐため重複 ACK の送信を遅らせる。もしこの遅らせる期間に、次に受信を期待するパケットが TCP の受信側に到着すると重複 ACK の送信をキャンセルする。ここで問題は重複 ACK の送信を遅らせる遅延時間の選択である。もし時間が短いと、不必要な高速再送を引き起こす重複 ACK が送信される可能性が大きくなる。逆に時間が長くと、ネットワークの有線部分で輻輳によりパケットを喪失した場合、パケットの回復が遅れる問題が発生する。

図 6.14 は DDA 方式と提案方式の比較を示している。ここで、DDA 方式において重複 ACK の送信遅延時間 (d) を 1.0 sec、2.0 sec とし、無線リンクの伝搬遅延時間 d と通信

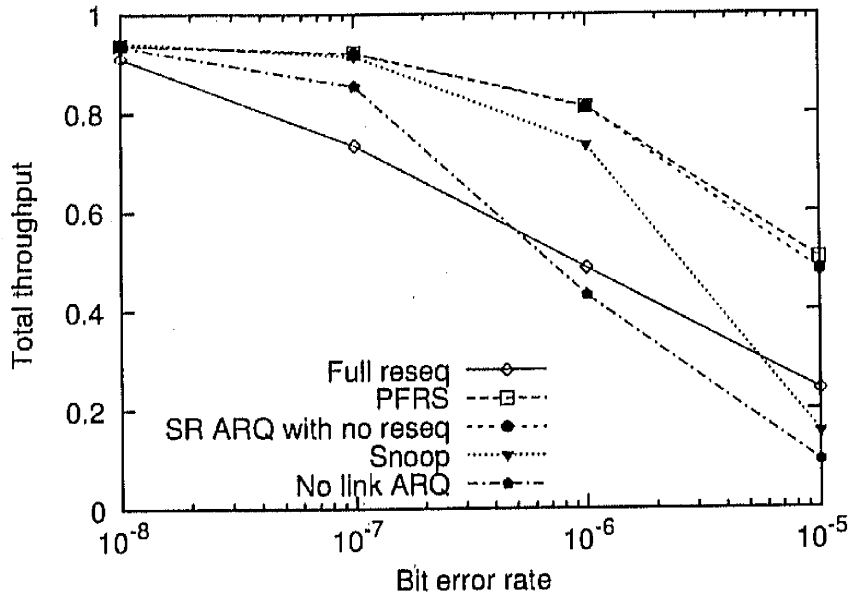


図 6.12 スループットの比較、TCP のコネクション数が 10, 伝送リンクの帯域と遅延時間が 5 Mbit/s と 0.25 sec の場合

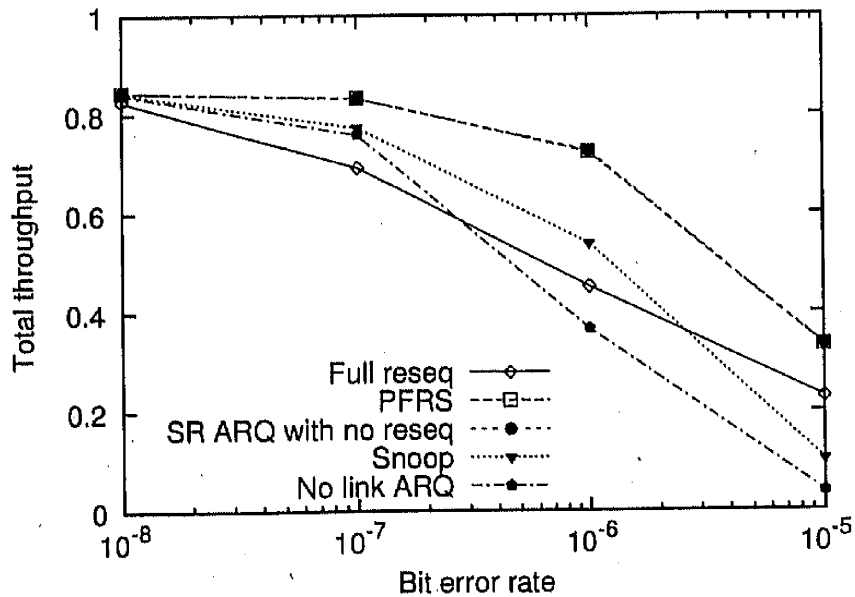


図 6.13 スループットの比較、TCP のコネクション数が 10, 伝送リンクの帯域と遅延時間が 50 Mbit/s と 0.025 sec の場合

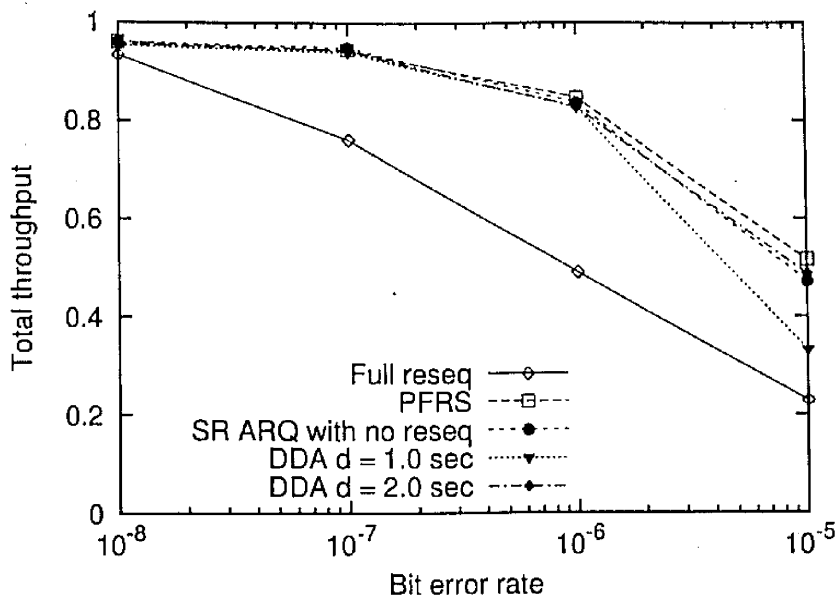


図 6.14 DDA 方式との比較、TCP のコネクション数が 10, 無線リンクの帯域と遅延時間が 5 Mbit/s と 0.25 sec の場合

速度をそれぞれ 0.25 sec と 5 Mbit/s としている。

遅延時間 d が 1.0 sec の場合、DDA の性能は他の方式より劣っている。遅延時間 d が 2.0 sec になると、DDA の性能は改善され PFRS 方式及び提案方式とほぼ同等の性能となっている。

図 6.15 はネットワークの有線部分でランダムにパケットの喪失が発生した場合の TCP によるパケット喪失への応答の遅れを示している。ここで、ネットワークの有線部分でのパケット喪失の確率を 0.001 としている。また無線リンクにおいてもパケットの伝送誤りが発生している。

ここでパケット喪失への応答とは高速再送またはスロースタートによる輻輳ウィンドウサイズの減少を意味している。有線部分でのパケット喪失に対する応答は無線リンクの伝送誤りにより遅れるが、DDA 方式は PFRS 方式及び提案方式と比較してパケット喪失の回復に時間を要している。

ネットワークの有線部分におけるパケット喪失は輻輳により発生するので、輻輳ウィンドウのサイズを減少させるリアクティブな制御を高速に行うことが望まれる。提案方式及び PFRS 方式はこの点で優れているが、DDA はネットワークの有線部分で輻輳が発生した場合の対応に時間を要する問題のあることが分かる。

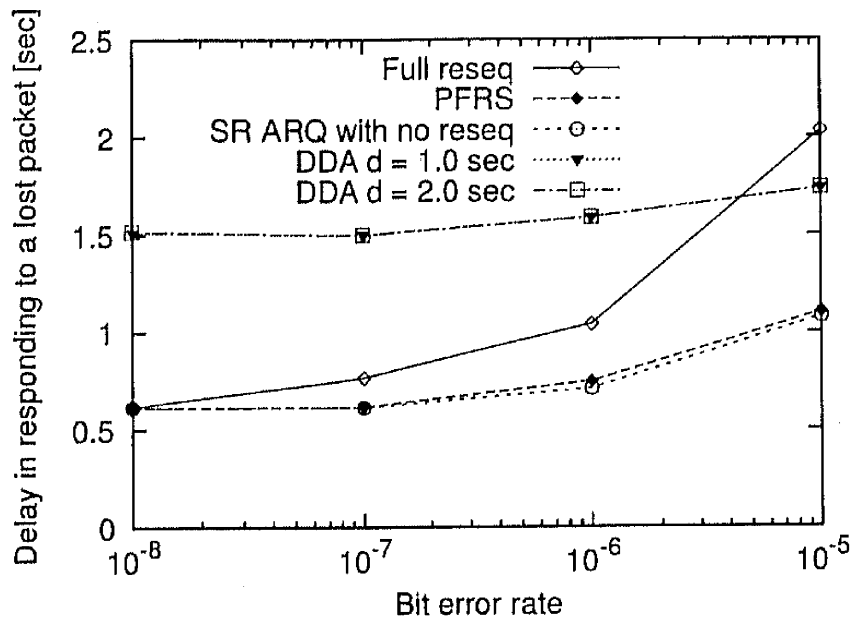


図 6.15 パケットの喪失に対する応答遅延時間、無線リンクの帯域と遅延時間が 5 Mbit/s と 0.25 sec、ネットワークの有線部分におけるパケット喪失の確率が 0.001 の場合

6.4.4 パケットの中継ノードにおけるバッファ量が限られている場合の性能

提案方式がバースト的なパケットトラフィックを発生しない点について、PFRS 方式と比較評価を行う。図 6.16 に構成を示すように、SR ARQ が出力するパケットを中継するノードのバッファ量が限られている場合についてシミュレーションを行った。この図で、パケットのフローはノード C からノード A の方向としている。PFRS 方式の場合、バースト的なパケットがノード B における順序制御により出力され、ノード D に 100 Mbit/s の速度で転送される。ノード D は受信したパケットを 10 Mbit/s の速度のリンクを用いてノード A に送信する。リンク速度の違いによりノード D においてパケットの待ち合わせが発生する。シミュレーションはノード D からノード A に転送する際のバッファ数が 50、20、18、15、10 の各場合について行った。

提案方式はパケットをバースト的に出力することはないので、バッファ数に係わらず性能は同じとなっている。PFRS 方式はパケットをバースト的に出力する。ここで、バースト的に出力されるパケット数は TCP のウィンドウサイズにより制限される。シミュレーションではウィンドウサイズを 32 KB としたので、最大で 22 パケットがノード D に転

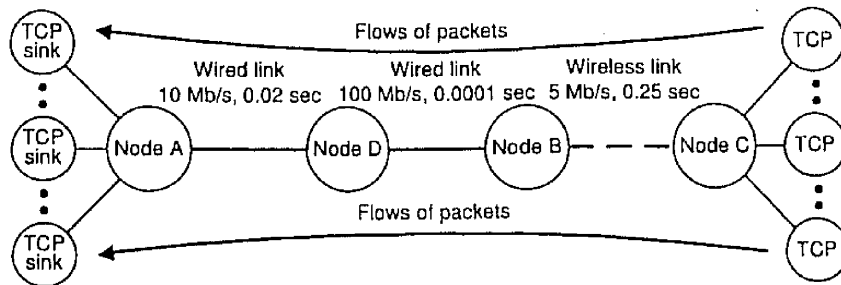


図 6.16 パースト的なパケット出力の影響を評価する場合のネットワーク構成

送され待ち行列に入る。バッファ数がウィンドウサイズより決まるパケット数よりも小さい場合はパケットの喪失が発生することになる。図 6.17 はシミュレーション結果を示している。バッファ数が少なくなると PFRS 方式のスループットが劣化することが分かる。

図 6.18 は SR ARQ の受信側で順序制御のために保留される平均バッファ数を示している。ここで、ノード D のバッファ数には制限がないものとし、TCP のコネクション数 N_{TCP} は 5 及び 10 とした場合を示している。ビット誤り率と TCP のコネクション数 N_{TCP} が増加すると PFRS 方式において順序制御のために保留されるバッファ数が大きくなることが分かる。提案方式は順序制御を行わないので、すべての場合においてバッファが保留されることはない。

図 6.17 と図 6.18 は SR ARQ の出力を中継するノードのバッファ数が限られている場合や SR ARQ の受信バッファ数に制限がある場合、提案方式が PFRS 方式より優れていることを示している。

6.4.5 スプリアスなタイムアウトの発生と輻輳ウィンドウサイズの回復

先に説明したように、ビット誤り率が大きい場合に提案方式のスループットは PFRS 方式より若干劣る結果となっている。この差は、以下の理由によるものと考えられる。

- 提案方式では、ビット誤り率が高い場合に TCP における無効なタイムアウト（スプリアスなタイムアウト）が PFRS 方式に比べ発生しやすい傾向がある。
- 提案方式では、TCP のタイムアウトが生じた場合に輻輳ウィンドウサイズの回復が PFRS 方式より遅い。

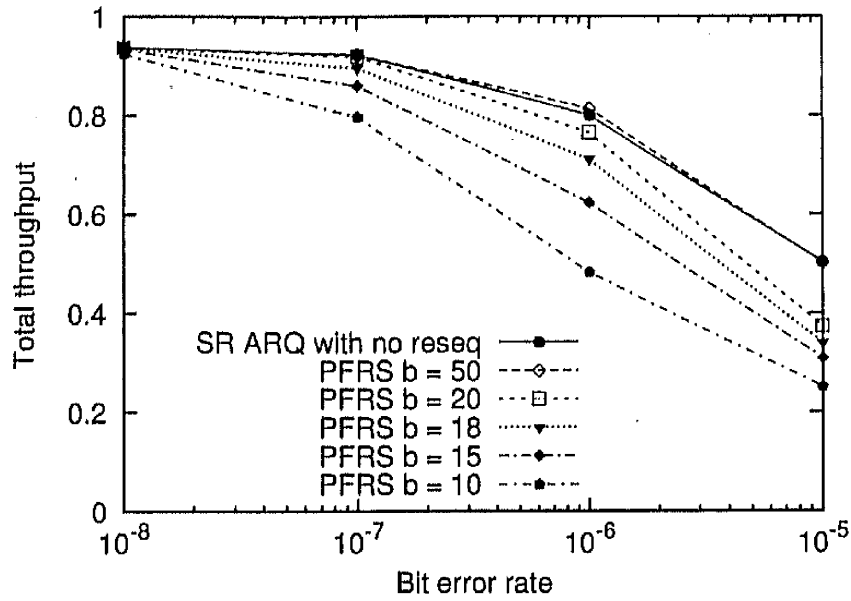


図 6.17 中継のノードのバッファ数が限られている場合の提案方式と PFRS 方式の比較

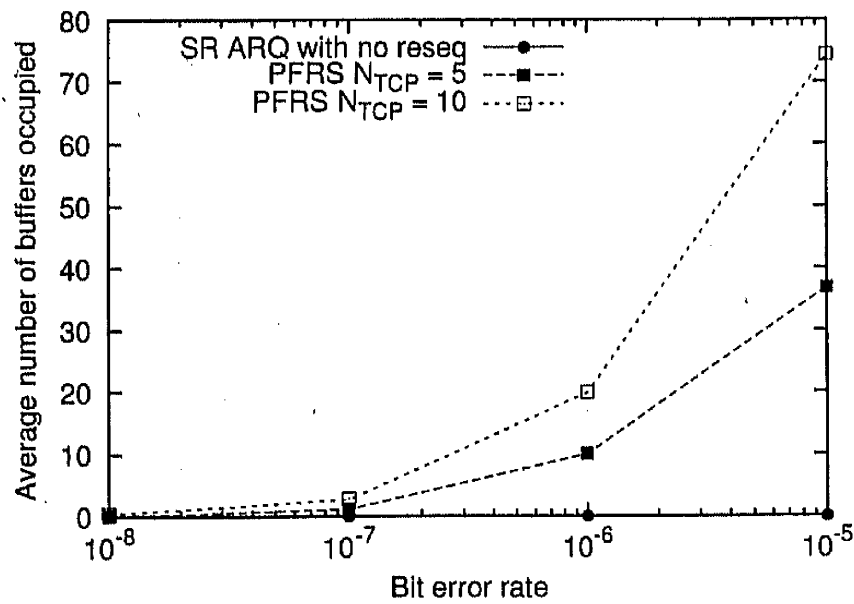


図 6.18 順序制御のために使用される平均バッファ数

(1) スプリアスなタイムアウト

表 6.2 はシミュレーションにおいて TCP のコネクション 1 本における TCP の平均タイムアウト発生回数を示している。この表において「No reseq.」は提案方式を示している。提案方式においてスプリアスなタイムアウトが PFRS 方式よりより頻繁に発生していることがわかる。提案方式においてスプリアスなタイムアウトが発生しやすい理由は以下のように考えられる。表 6.2 はシミュレーションにおいて TCP が計測しているラウンドトリップ時間 (RTT) の平均値を示している。この表から、ビット誤り率が小さい場合に提案方式において計測された RTT 値は PFRS 方式で計測された場合より小さいことが分かる。シミュレーションにおいて TCP は常時 RTT を計測しているが、同時には 1 回のみである [81]。TCP の受信側で SR ARQ の再送に伴う順序制御を行うと、複数の TCP のデータパケットに対する単一の ACK が TCP の送信側に返送される。1 回の RTT の計測はこの時点で終了する可能性が高く、次の RTT の計測は次の新しいデータパケットが送信されるときに開始される。この新しいパケットの前に送信されたパケットはすべて送達確認されているか、または送達未確認のパケットは少ない。このため新しく送信されたパケットが順序制御により遅れる可能性は通常のパケットより小さい。従って、このパケットにより計測される RTT 値は通常の場合よりも小さくなる傾向がある。これが、再送タイマの値を小さめに設定することとなり、このため提案方式においてタイムアウトが発生しやすくなるものと考えられる。

このような TCP のスプリアスタimeアウトによる問題を解決する方法としては、Eifel アルゴリズムが提案されている [82][83][97]。Eifel アルゴリズムは TCP のタイムスタンプオプションを用いて、スプリアスな再送か否かを判断し、スプリアスな再送の場合は輻輳制御の状態を元に戻そうとするものである。このアルゴリズムについては輻輳制御の状態を元に戻す場合に、バースト的にパケットの送信が行われる問題が指摘されており [84]、今後の研究が必要である。

(2) タイムアウト後の輻輳ウィンドウサイズの回復

図 6.19 は提案方式における輻輳ウィンドウサイズの変化例を示している。ここで、ビット誤り率は 10^{-6} 、TCP のコネクション数は 5 としている。図 6.20 は同じ条件における PFRS 方式の輻輳ウィンドウサイズの変化例を示している。両方の図において、シミュレーションにおいて用いたネットワーク構成は図 6.16 と同じで、ノード D のバッファ数は無制限としている。TCP におけるスプリアスなタイムアウトはパケットがリンク層で喪失し再送される場合に発生しやすい。タイムアウトが発生すると輻輳ウィンドウ (cwnd) のサイズは 1 MSS (Maximum Segment Size) となり、スロースタートが実行さ

表 6.2 TCP のコネクション 1 本における平均タイムアウト発生回数、TCP のコネクション数 (N_{TCP}) が 10 の場合

Bit error rate	Scheme	Timeouts	RTT [sec]
10^{-6}	PFRS	0.38	0.617
	No reseq.	0.56	0.593
10^{-5}	PFRS	0.41	0.955
	No reseq.	0.59	0.804

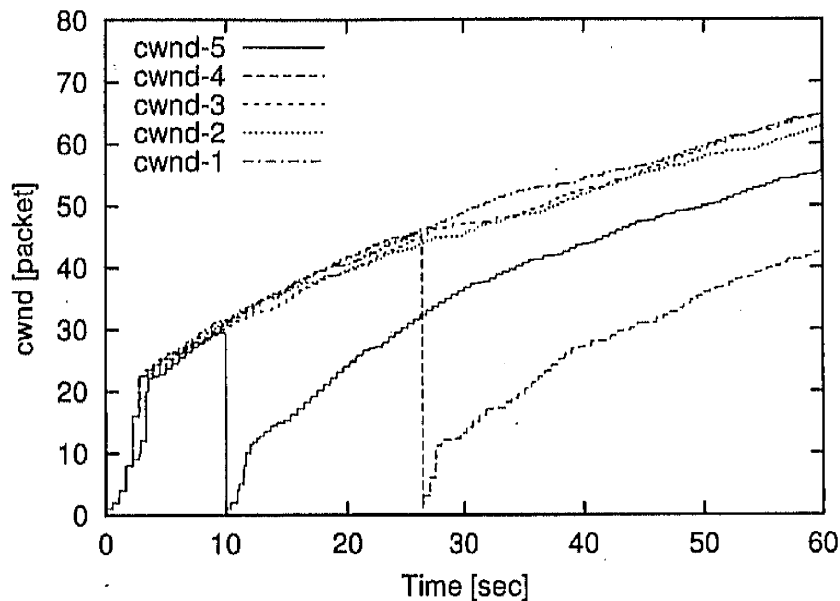


図 6.19 提案方式における輻輳ウィンドウサイズ (cwnd) の変化例、ビット誤り率は 10^{-6} 、TCP のコネクション数は 5、無線リンクの帯域と遅延時間はそれぞれ 5 Mbit/s と 0.25 sec

れる。これらの図から分かるように、提案方式における輻輳ウィンドウサイズの回復は PFRS 方式に比べ緩やかである。提案方式では輻輳ウィンドウサイズが小さい時間が長いためスループットが低下することになる。

PFRS 方式では、SR ARQ による順序制御の後に順序の正しい複数の TCP データパケットが連続して TCP の受信側に到着する。このため、通番が更新された新しい ACK 複数が TCP の送信側に返送される。TCP におけるスプリアスなタイムアウトが発生した場合、タイムアウト後にこれらの新しい ACK は TCP の送信側に連続して到着する可能性が高い。輻輳ウィンドウの更新は新しい ACK を受信するたびに行われるので、

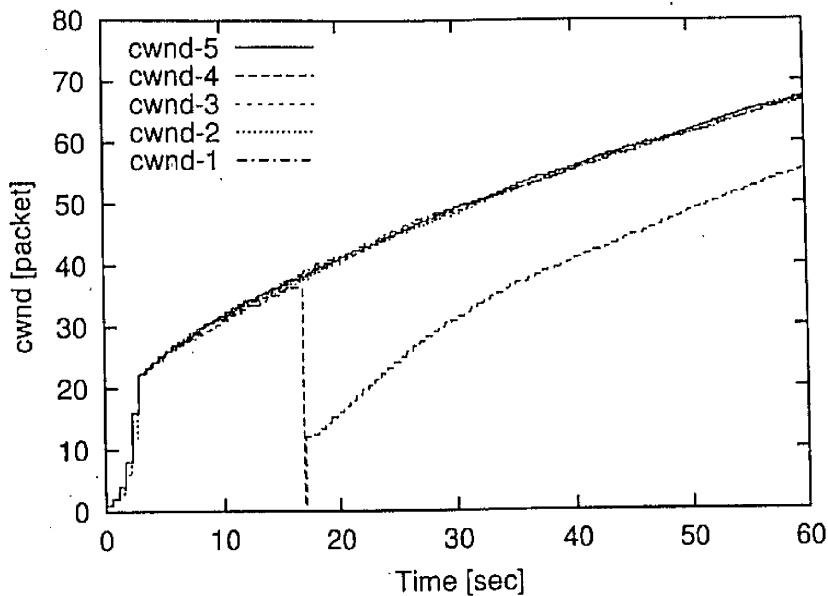


図 6.20 PFRS 方式における輻輳ウィンドウサイズ (cwnd) の変化例、ビット誤り率は 10^{-6} 、TCP のコネクション数は 5、無線リンクの帯域と遅延時間はそれぞれ 5 Mbit/s と 0.25 sec

PFRS 方式では輻輳ウィンドウサイズが急速に増加する。

図 6.24 は PFRS 方式において TCP の送信側に到着した ACK の通番の時間変化を示している。図中には時刻が 17.055 sec から 17.65 sec の間の詳細な変化を拡大して示している。この図では TCP の送信側が送信したデータパケットの通番も併せて示している。シミュレーションでは各 TCP のデータパケットに通番が付与され、新しい TCP のデータパケットが送信するたびに通番の値は 1 増加する。図ではタイムアウトが時刻 16.9 sec 付近で発生している。その後、時刻 17.057 sec から 17.083 sec の間に通番の更新去れた 22 の新しい ACK が到着している。新しい ACK が到着するたびに 1 または 2 個の新しいデータパケットが送信されている。到着した ACK の通番と送信されたデータパケットの通番の差は輻輳ウィンドウ (cwnd) の大きさに相当する。この期間に輻輳ウィンドウの回復が急速に行われることが分かる。

一方、提案方式では、SR ARQ による再送の後に TCP の受信側で順序制御が行われると、複数の TCP データパケットの送達を確認する新しい ACK が 1 パケット返送されるのみである。喪失したパケットがリンク層による再送を回復された場合、輻輳ウィンドウ (cwnd) の更新は 1 度しか行われない。このために提案方式では輻輳ウィンドウの回復が PFRS 方式より遅くなる。

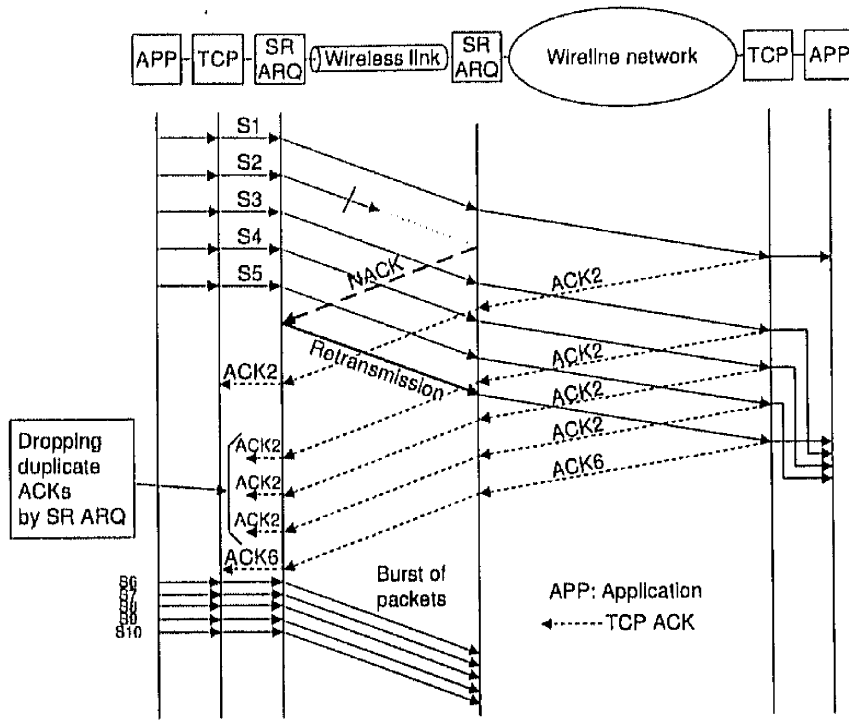


図 6.21 提案方式において複数パケットがバースト的に送信される場合のシーケンス例

6.4.6 TCPによるバースト的なパケット出力

上で述べたように、提案方式では喪失したパケットがリンク層により再送された後、TCPで順序制御が行われると、複数のTCPのデータパケットをまとめて送達確認するACKが1パケットのみTCPの送信側に返送される。TCPの送信側がこのACKを受信するとウィンドウが更新され、複数の新しいデータパケットをまとめて送信することが可能である。図6.21は提案方式において複数パケットがバースト的に出力される場合のシーケンス例を示している。

TCPが同時に送信する複数パケットをバーストと呼ぶ。TCPは1RTTの間にウィンドウサイズ分のパケットが送達確認なしに送ることができるのでバーストには最大でウィンドウサイズ相当数の新しいパケットが含まれる。このようなバーストは非対称な経路を用いる場合においてACKの頻度を減少させる場合にも発生することが知られている[101]。

図6.22及び図6.23はTCPの送信側におけるPFRS方式と提案方式それぞれにおけるバーストに含まれるパケット数の分布(相対度数)を示している。ここで、ビット誤り

率は 10^{-5} 、TCP のコネクション数は 10 としている。図中には頻度が小さい部分の詳細を拡大して示している。ウィンドウサイズが 32 KB でパケット長を 1460 B としているので、最大で 22 パケットが送達未確認となり得る。図 6.23 から提案方式は最大で 22 パケットからなるバーストを発生させる場合のあることが分かる。一方、PFRS 方式は最大で 2 パケットのバーストしか発生しない。この 2 パケットのバーストは主にスロースタート期間に発生している。シミュレーションではこのような TCP が発生するバーストを受信出来る十分なバッファを有していた。しかし、ノード内のバッファが限られている場合にはパケットがバッファ不足により失われる可能性もある。

この問題を解決する一つの方法は新しい ACK を受信したときに同時に送信できるデータパケット数を制限することが考えられる [86]。シミュレーションに使用した TCP は単一の新しい ACK 受信時に送信できるデータパケット数を制限するパラメータ “maxburst” を有するが [106][86]、シミュレーションはこのパラメータによる送信パケット数の制限は行っていない。他のアプローチとしてはバーストの発生を抑制する TCP pacing を用いることが考えられる。TCP pacing はウィンドウサイズ分のパケットの送信を 1RTT 時間の間に均等に分散させるものである [87]。また、提案方式では SR ARQ において重複 ACK を削除したが、SR ARQ が複数データパケットまとめて送達確認する単一 ACK を受信した場合は、複数の新しい ACK (例えば 1 パケットのみを送達確認する) を生成する可能性も考えられる。これらのアプローチの評価に関して更なる今後の研究が必要と考えられる。

6.5 むすび

この章では SR ARQ における順序制御の問題を解決する方式として、SR ARQ では順序制御を行わず、順序制御をエンドの TCP に任せる方式を提案した。提案方式は SR ARQ の受信側においてバッファが不要であること、バースト的なパケット出力が発生しないことの利点がある。さらに多数の TCP コネクションを SR ARQ に多重化する場合に問題となる HOL ブロッキングが発生しない利点も有する。これらの利点は、帯域と遅延時間の積が急速に増大している無線ブロードバンドネットワークにおいて効果が顕著である。また、提案方式はネットワークの輻輳制御など TCP におけるエンドツーエンドの通信を保存する特長も有する。

シミュレーションにより提案方式の性能を他の方式と比較評価した。提案方式は帯域と遅延時間の積が大きな環境において優れた性能を有することを確認した。また SR ARQ の出力パケットを中継するノードのバッファ数が限られている場合、提案方式はバースト的にパケットを出力することがないため、PFRS 方式より優れた性能を達成することを確認した。

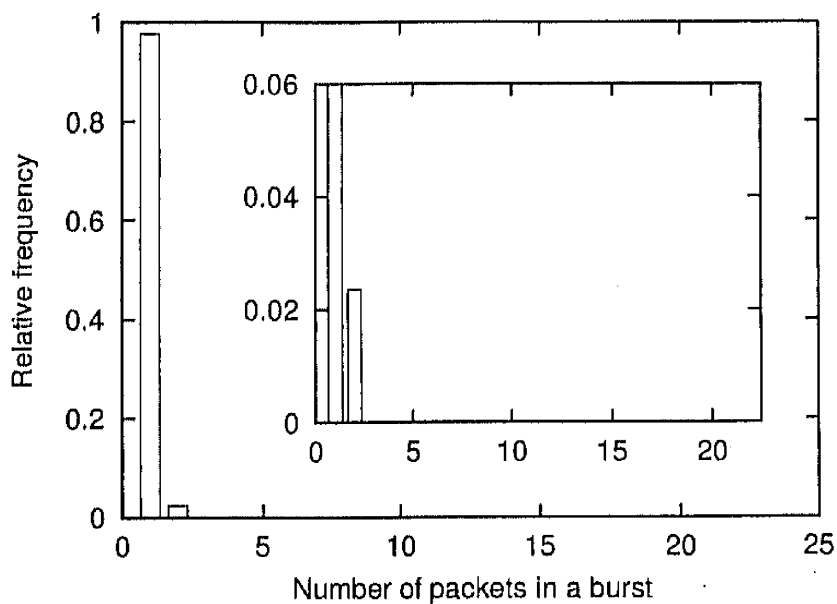


図 6.22 PFRS 方式の場合の TCP の送信側におけるバーストサイズの分布、ビット誤り率は 10^{-5} 、TCP のコネクション数は 10

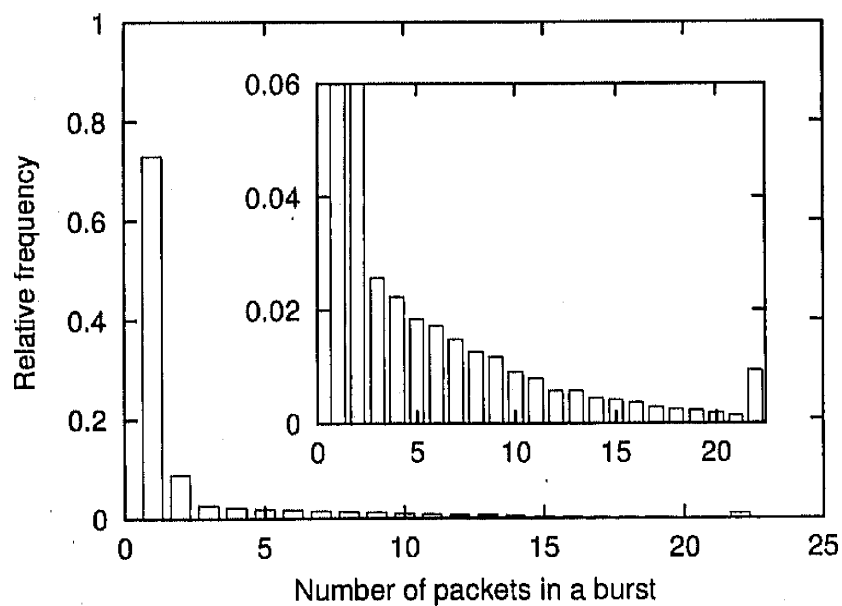


図 6.23 提案方式の場合の TCP の送信側におけるバーストサイズの分布、ビット誤り率は 10^{-5} 、TCP のコネクション数は 10

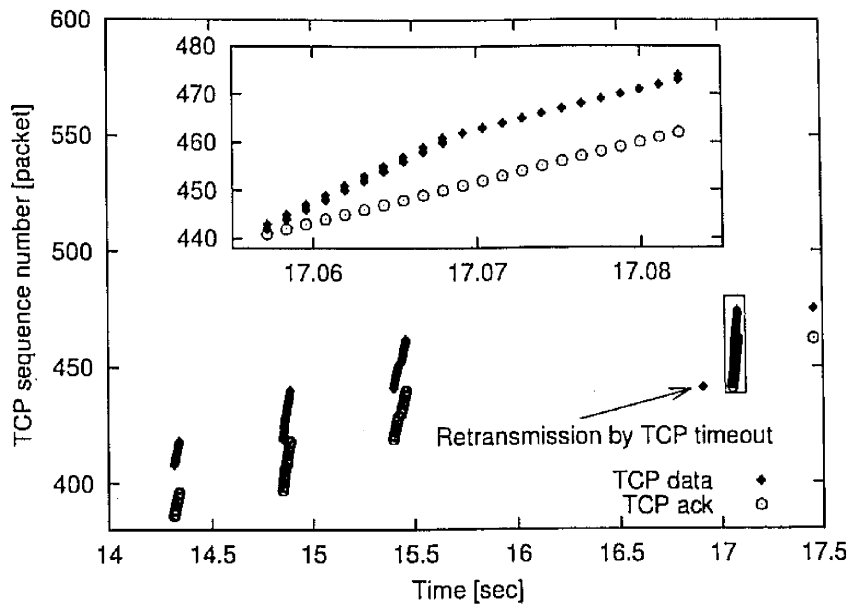


図 6.24 図 6.20 の場合の TCP のデータパケット及び ACK の通番の変化

認した。

PFRS 方式は TCP を意識した処理を必要としない利点を有するが、TCP のコネクション数、通信帯域、TCP のウィンドウサイズに対するスケーラビリティにおいて問題がある。提案方式はこれらに対してスケーラブルであるが、一方で TCP を意識した処理が必要となる欠点もある。PFRS 方式と提案方式は相互補完的な位置づけにあると考えられる。すなわち、プロトコルの種別が不明のフローに対しては PFRS 方式を用いることになる。しかし、TCP が用いられているフローに対しては SR ARQ の受信バッファが限られている場合や、SR ARQ によるバースト的なパケット出力が許容されない場合は提案方式を用いたほうがよい。

6.4.6 で述べたように、提案方式はスプリアスなタイムアウトの発生、輻輳ウィンドウサイズの回復に時間を要すること、TCP におけるバースト的なパケット出力など更なる改善の余地がある。これらの問題に向けた解決方式が今後の課題として残されている。

第7章

結言

この章では、これまで説明した研究内容をまとめるとともに、今後の展開について述べる。

7.1 本研究のまとめ

2章では、SR ARQによるバースト的なパケット出力について検討し、バーストに含まれるパケット数の分布とバーストの発生間隔分布を求めた。また、ある値以上の大きさのバーストが発生する平均間隔を求めた。これらの解析結果をシミュレーションと比較し、解析の妥当性を確認した。これらから、最大再送回数が制限された場合でも、バースト長の分布とバースト間隔分布の評価が可能であることを示した。この章の解析による興味深い結果の一つは、最大長のバーストが発生する確率はパケット誤り率 ϵ が $\epsilon \approx 1/M$ の時に最大となることである。また別の興味深い結果は、バースト長とバースト間隔の変動係数が同様の値でピークをとることである。これらの結果は、パケット誤り率の値がラウンドトリップ時間に送信されるパケット数の逆数に等しい場合に長さの大きなバーストが発生しやすいことを示している。バースト的なパケット出力はパケット誤り率が大きい場合にその影響が顕著となる訳ではなく、帯域と遅延時間の積が大きい場合は、小さなパケット誤り率でもその影響に注意する必要がある。もしパケットの順序逆転が許容されるならば、SR ARQによるバースト的なパケット出力は、SR ARQにおいて順序制御を行わなければ発生しない。また2章では、SR ARQが順序制御を部分的に行う場合や全く行わない場合について順序逆転パケット数の分布についても解析した。SR ARQの上でTCPが用いられる場合、TCPへの順序逆転パケットによる影響を無視できるレベルとするためには完全な順序制御に近い数のパケット保留が必要であることが分かった。

3章では、SR ARQについて順序制御のみによる遅延時間の確率分布及び再送と順序制

御による遅延時間の確率分布に関する解析を示した。最大再送回数を制限した場合の数値計算が可能で、計算と同じ条件によるシミュレーション結果との一致により解析が正しいことを確認した。本論文で述べた解析により、すべてのパケットの順序を保存する従来の全順序制御方式と HOL ブロッキングを回避した場合の両方の遅延時間分布を求めることが可能である。最大再送回数をある程度大きくすればパケット損失は無視できるレベルとなり、本論文で述べた解析は最大再送回数を制限しない場合の評価にも適用可能と考えられる。計算結果から全順序制御方式において、再送そのものよりも再送に伴う順序制御による遅延時間が支配的であること、SR ARQ 上に多重化される上位層フロー数が増加すると HOL ブロッキングによる遅延時間の影響が大きくなることが分かった。

4章では帯域と遅延時間の積が大きい回線において SR ARQ により誤り回復を行うアーキテクチャについて考察し、多数のフローを SR ARQ に多重化する場合に SR ARQ の受信側での順序制御により遅延時間が大きくなる問題、またこの順序制御によりパケットがバースト的に出力される問題を解決する方式として、パケット消失の検出と再送は上位層のフロー全体で行い、順序制御のみを上位層のフロー個別に行う PFRS 方式を提案した。また、この方式をエンドツーエンドで UDP 及び TCP が用いられる場合についてシミュレーションによる評価を行った。この結果、従来の全フロー ARQ 方式と比較すると、UDP トラヒックの場合は PFRS 方式により平均遅延時間や順序制御のために保留されるバッファ量が減少することが分かった。また TCP トラヒックの場合、出力でのバースト長が小さくなること、またエンドツーエンドで TCP が用いられる場合にスループットが向上することが分かった。更にフローごとに ARQ を用いる個別フロー ARQ 方式と比較すると、UDP トラヒックの場合は PFRS 方式の遅延時間や保留バッファ数が少なく、TCP トラヒックの場合はスループットやバースト性がほぼ同等であることが分かった。PFRS 方式は1本の静的な SR ARQ コネクションを用いればよく、個別フロー ARQ 方式に比べ SR ARQ コネクションの設定や SR ARQ が必要とする順序制御のためのバッファなどのリソース管理が単純となる特長がある。このような特長は、上位層のフローを把握することが難しい IP トラヒックにおいて特に有効である。

5章では、SR ARQ において順序制御を上位層のフローごとに行う PFRS 方式について実現方法を検討し、SR ARQ の通番を利用するポインタ方式を提案した。このポインタ方式では SR ARQ 上で多重化されるフロー数の上限がなくフローごとに順序制御を実現できる特長がある。また SR ARQ の通番の差分を用いるポインタ方式について検討し、値の大きい差分ポインタの頻度が小さくかつ順序制御による遅延時間が短くなる傾向のあることを示した。これらを利用することにより、差分ポインタ値に上限を設けて差分ポインタによるオーバーヘッドを減少させることが可能であることを示した。また上限を超える場合の処置として順序制御を行わない方式と上限値より前のフレームをすべて受信す

るまで保留する方式を比較し、順序制御を行わない方式の方が小さな上限値で済むこと、また実現が容易であることを示した。

6章ではSR ARQにおける順序制御の問題を解決する方式として、SR ARQでは順序制御を行わず、順序制御をエンドのTCPに任せる方式を提案した。提案方式はSR ARQの受信側においてバッファが不要であること、バースト的なパケット出力が発生しないことの利点がある。この方式は多数のTCPコネクションをSR ARQに多重化する場合に問題となるHOLブロッキングが発生しない利点も有する。これらの利点は、帯域と遅延時間の積が急速に増大している無線ブロードバンドネットワークにおいて効果が顕著である。提案方式はネットワークの輻輳制御などTCPにおけるエンドツーエンドの通信を保存する特長も有する。

シミュレーションにより提案方式の性能を他の方式と比較評価した。提案方式は帯域と遅延時間の積が大きな環境において優れた性能を有することを確認した。またSR ARQの出力パケットを中継するノードのバッファ数が限られている場合、提案方式はバースト的にパケットを出力することがないため、PFRS方式より優れた性能を達成することを確認した。

PFRS方式はTCPを意識した処理を必要としない利点を有するが、TCPのコネクション数、通信帯域、TCPのウィンドウサイズに対するスケラビリティにおいて問題がある。提案方式はこれらに対してスケラブルであるが、一方でTCPを意識した処理が必要となる欠点もある。PFRS方式と提案方式は相互補完的な位置づけにあると考えられる。すなわち、プロトコルの種別が不明のフローに対してはPFRS方式を用いることになる。しかし、TCPが用いられているフローに対してはSR ARQの受信バッファが限られている場合や、SR ARQによるバースト的なパケット出力が許容されない場合は提案方式がよい。

提案方式はスプリアスなタイムアウトの発生、輻輳ウィンドウサイズの回復に時間を要すること、TCPにおけるバースト的なパケット出力など更なる改善の余地がある。これらの問題に向けた解決方式が今後の課題として残されている。

以上のように、本論文では今後の高速無線通信においてSR ARQを用いる場合に、従来把握されていなかった出力トラヒックのバースト性とHOLブロッキングの問題点を明らかにし、これらの問題を解決する新し方式を提案し有効であることを示した。

7.2 今後の課題

本論文ではバースト性の評価を行ったが、実システムではバースト性を緩和するためトラヒックシェーピングが用いられると考えられる。このようなトラヒックシェーピング自

体が更に遅延を発生させることが予想される。トラヒックシェーピングを含んだ遅延時間の評価は数学的な取り扱いが難しいと考えられるが、今後の課題である。また、本研究では順序制御による遅延時間に焦点を当てて解析を行ったが、送信側での待ち行列による遅延時間を含めた解析も今後の課題である。提案したポインタによる実現方式では再送回数の制限を設けない場合を評価した。ポインタ方式は、フローごとに最大再送回数を変えて、異なるフローの QoS 要求を満足することが可能である。このような拡張も今後の課題と考えられる。更に、ポインタ方式は複数のリンクを束ねて用いるマルチリンクや TCP で複数のアプリケーションのデータを送る場合（マルチストリーミング）への応用も考えられる。

順序制御はパケット通信における本質的な問題であるが、これまであまり表に出てこなかった。一つの理由は、これまではネットワークでパケットの順序を守る基本方針がとられてきたからと考えられる。SR ARQ は順序制御の問題が顕著となる典型的な場合である。今後、複数経路へのパケットの負荷分散などが用いられるようになると SR ARQ 以外でも類似の問題が発生する。本論文で述べた内容は、このような場合にも有効と考えられる。

付録 A

チャネルの状態確率の導出

(3.1) 式と (3.2) 式で与えられるチャネルの状態確率を導出する。図 3.5 に示す状態遷移図から、平衡状態では各状態における入りフローと出フローが等しくなるので、以下の関係が成立する。

$$\begin{aligned}\alpha p_s(0) &= p_s(1) \\ \varepsilon p_s(1) &= p_s(2) \\ \varepsilon p_s(2) &= p_s(3) \\ &\vdots \\ \varepsilon p_s(N_r - 1) &= p_s(N_r)\end{aligned}$$

これから $p_s(k)$ は $p_s(0)$ により表される。ここで、 $1 \leq k \leq N_r$ 。

$$\begin{aligned}p_s(1) &= \alpha \varepsilon p_s(0) \\ p_s(2) &= \varepsilon p_s(1) = \alpha \varepsilon^2 p_s(0) \\ &\vdots \\ p_s(N_r) &= \varepsilon p_s(N_r - 1) = \alpha \varepsilon^{N_r} p_s(0)\end{aligned}$$

$\sum_{k=0}^{N_r} p_s(k) = 1$ であるから、 $p_s(0)$ は次のように求まる。

$$p_s(0)(1 + \alpha \varepsilon + \alpha \varepsilon^2 + \dots + \alpha \varepsilon^{N_r}) = 1$$

$$p_s(0) = \frac{1 - \varepsilon}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r+1}}$$

状態確率 $p_s(k)$ が以下のようになる。ここで、 $1 \leq k \leq N_r$ 。

$$p_s(k) = \frac{\alpha(1-\varepsilon)\varepsilon^k}{1-\varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}$$

付録 B

残余受信失敗回数による状態確率の導出

(3.4) 式と (3.5) 式により表される残余受信失敗回数による状態確率を求める。残余受信失敗回数が 0 の場合と 0 より大きい場合の二つの場合を考える必要がある。

残余受信失敗回数が 0 となるのは以下のいずれかの場合である。

- 最初の定義による状態が 0 の場合、パケットが確率 α で送信され正しく受信されるか、残りの確率 $1 - \alpha$ でパケットが送信されない。
- 最初の定義による状態が k の場合 (ここで、 $1 \leq k \leq N_r$)、現在のパケットが次の再送で正しく受信される。
- 最初の定義による状態が N_r の場合、次の再送で状態は確率 1 で 0 に戻る。

これから、確率 $p_r(0)$ は以下のように計算される。

$$\begin{aligned}
 p_r(0) &= \{1 - \alpha + \alpha(1 - \varepsilon)\}p_s(0) + \sum_{k=1}^{N_r-1} (1 - \varepsilon)p_s(k) + p_s(N_r) \\
 &= \frac{(1 - \alpha\varepsilon)(1 - \varepsilon) + (1 - \varepsilon)^2\alpha \sum_{k=1}^{N_r-1} \varepsilon^k + \alpha(1 - \varepsilon)\varepsilon^{N_r}}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \\
 &= \frac{(1 - \varepsilon)\{1 - \alpha\varepsilon + \alpha\varepsilon(1 - \varepsilon^{N_r-1}) + \alpha\varepsilon^{N_r}\}}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}} \\
 &= \frac{1 - \varepsilon}{1 - \varepsilon + \alpha\varepsilon - \alpha\varepsilon^{N_r+1}}
 \end{aligned}$$

残余受信失敗回数が r ($1 \leq r \leq N_r$) となるのは以下のいずれかの場合である。

- 最初の定義による状態が 0 の場合、新規パケットが確率 α で送信され、 r 回の受信

失敗の後に正しく受信される。

- 最初の定義による状態が $k(1 \leq k \leq N_r - r - 1)$ の場合、現在のパケットが r 回の受信失敗の後に正しく受信される。
- 最初の定義による状態が $N_r - r$ の場合、 r 回の受信失敗後に確率 1 で状態が 0 に戻る。

これから確率 $p_r(r)$ は以下の様に計算できる。

$$\begin{aligned}
 p_r(r) &= \alpha \varepsilon^r (1 - \varepsilon) p_s(0) + \sum_{k=1}^{N_r - r - 1} (1 - \varepsilon) \varepsilon^r p_s(k) + \varepsilon^r p_s(N_r - r) \\
 &= \frac{\alpha \varepsilon^r (1 - \varepsilon)^2 + (1 - \varepsilon)^2 \alpha \varepsilon^r \sum_{k=1}^{N_r - r - 1} \varepsilon^k + \varepsilon^r \alpha (1 - \varepsilon) \varepsilon^{N_r - r}}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r + 1}} \\
 &= \frac{\alpha (1 - \varepsilon) \{ \varepsilon^r (1 - \varepsilon) + \varepsilon^{r+1} (1 - \varepsilon^{N_r - r - 1}) + \varepsilon^{N_r} \}}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r + 1}} \\
 &= \frac{\alpha (1 - \varepsilon) \varepsilon^r}{1 - \varepsilon + \alpha \varepsilon - \alpha \varepsilon^{N_r + 1}}
 \end{aligned}$$

謝辞

本研究においてご指導ご鞭撻をいただいた静岡大学情報学部教授 水野忠則博士に深く感謝申し上げます。また、懇切なるご指導をいただいた静岡大学情報学部教授 渡辺 尚博士に深く感謝いたします。

静岡大学大学院博士課程への社会人入学の機会を与えていただいた三菱電機株式会社 情報技術総合研究所所長 肥塚裕至博士に深く感謝します。約 30 年にわたる三菱電機株式会社での研究成果をまとめる機会そして励ましをいただいた前三菱電機株式会社 情報技術総合研究所副所長 風間成介氏に深く感謝します。

本研究の中心テーマである選択再送プロトコルにおける順序制御は 1980 年代前半に担当した衛星利用コンピュータネットワークシステムの開発の中で、その重要性と技術課題を認識したもので、研究の発端は元三菱電機株式会社 松田主税氏の影響によるものです。ここに深く感謝します。

本研究の中核となるアイデアはフランスに所在の三菱電機株式会社海外研究所 ITE 勤務中に考案し特許化しています。フランス勤務中お世話になった前三菱電機株式会社 情報技術総合研究所所長 小野修一氏及び ITE 所長 J. P. Coudreuse 氏に深謝いたします。

本論文は無線通信ネットワークを対象とした再送プロトコルの順序制御を中心に研究成果をまとめていますが、長い研究所勤務で幅広い分野で研究開発を経験し、これらを通じて得られた見識が本論文のベースとして存在します。ご指導いただいた神奈川工科大学情報学部教授 石坂充弘博士、三菱電機情報ネットワーク株式会社常務取締役 青山 滋氏に深く感謝いたします。また研究開発を通して多大な協力をいただいた横谷哲也博士、土田 充氏、妹尾尚一郎氏、河東晴子博士、寺島美昭氏、荒谷和徳氏、宮内直人氏、小林信之氏、山田浩利氏に感謝いたします。

最後に、研究を進めるにあたり協力いただいた家族（弘子、愛子、亮太郎、順子）に感謝します。

参考文献

- [1] 総務省, “e-Japan 戦略,” www.soumu.com/elocalgov/contents/c075.html
- [2] 総務省, “ユビキタスネットワーク,” www.soumu.go.jp/menu_02/ict/u-japan
- [3] M. Weiser, “The Computer for the 21st Century,” *Scientific American*, vol.265, no.3, pp.94-104, Sept. 1991.
- [4] 野村総合研究所, “ユビキタス・ネットワーク,” 野村総合研究所, 2000.
- [5] 笹瀬 巖, “次世代移動通信の動向,” オペレーションズリサーチ学会誌, pp.477-483, August 2004.
- [6] “究極の誤り訂正符号「LDPC」通信, 放送, HDD まで席卷,” 日経エレクトロニクス 2005.1.31.
- [7] 宮原秀夫, 尾家裕二, “コンピュータネットワーク,” 共立出版, 1999.
- [8] 尾家裕二, 後藤滋樹, 小西和憲, 西尾章治郎, “インターネット入門,” 岩波書店, 2001.
- [9] A.S. Tanenbaum, 水野忠則, 相田仁, 東野輝夫, 太田賢, 西垣正勝訳, “コンピュータネットワーク 第4版,” 日経 BP 社, 2003.
- [10] 村山公保, 西田佳史, 尾家裕二, “トランスポートプロトコル,” 岩波書店, 2001.
- [11] 立川敬二監修, “W-CDMA 移動通信方式,” 丸善, 2001.
- [12] 西林泰如, 宇都宮依子, 高木雅裕, “無線 LAN における選択的再送制御を用いた MAC フレームアグリゲーション方式の提案,” 信学技報, IN2004-113, pp.31-36, Nov. 2004.
- [13] P. Loret, M. Luglio, R. Kapoor, J. Stepanek, M. Gerla, F. Vatalaro, and M. A. Vázquez-Castro, “Satellite Systems Performance with TCP-IP Applications,” *MILCOM 2001*, vol.2, pp.811-815, Oct. 2001.
- [14] V. Kawadia and P. R. Kumar, “A Cautionary Perspective on Cross-Layer Design,” *IEEE Wireless Commun.*, vol.12; no.1, pp.3-11, Feb. 2005.
- [15] D. Towsley and J. K. Wolf, “On the Statistical Analysis of Queue Length and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes,” *IEEE Trans. Commun.*, vol.COM-27, no.4, pp.693-702, April 1979.
- [16] S. Lin and P. S. Yu, “An Effective Error Control Scheme for Satellite Communi-

- cations," *IEEE Trans. Commun.*, vol.COM-28, no.3, pp.395-401, March 1980.
- [17] A. G. Konheim, "A Queuing Analysis of Two ARQ Protocols," *IEEE Trans. Commun.*, vol.28, no.7, pp.1004-1014, July 1980.
- [18] M. J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Scheme with Finite Receiver Buffer," *IEEE Trans. Commun.*, vol.COM-29, no.9, pp.1307-1315, Sept. 1981.
- [19] E. J. Weldon, Jr., "An Improved Selective-Repeat ARQ Strategy," *IEEE Trans. Commun.*, vol.COM-30, no.3, pp.480-486, March 1982.
- [20] B. H. Sacki and I. Rubin, "An Analysis of a TDMA Channel Using Stop-and-Wait, Block, and Select-and-Repeat ARQ Error Control," *IEEE Trans. Commun.*, vol.COM-30, no.5, pp.1162-1175, May 1982.
- [21] D. M. Chitre, "A Selective-Repeat ARQ Scheme and Its Throughput Analysis," *Int. Conf. Commun.*, vol.3, pp.6G.4.1-6, June 1982.
- [22] F. Baccelli, E. Gelenbe, and B. Plateau, "An End-to-End Approach to the Resequencing Problem," *Journal of the ACM*, vol.31, no.3, pp.474-485, July 1984.
- [23] M. E. Anagnostou and E. N. Protonotarios, "Performance Analysis of the Selective Repeat ARQ Protocol," *IEEE Trans. Commun.*, vol. 34, no.2, pp.127-135, Feb. 1986.
- [24] S. R. Chandran and S. Lin, "A Selective-Repeat ARQ for Point-to-point Communications and Its Throughput Analysis," *ACM SIG-COMM'86*, vol. 16, no. 3, pp. 292-301, 1986.
- [25] H. Bruneel and M. Moeneclaey, "On the Throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions," *IEEE Trans. Commun.*, vol.COM-25, no.5, pp.473-479, May 1977.
- [26] D. Towsley and S. Mithal, "A Selective Repeat ARQ Protocol for a Point to Multipoint Channel," *IEEE INFOCOM*, pp.521-526, 1987.
- [27] N. Shacham and D. Twosley, "Resequencing Delay and Buffer Occupancy in Selective Repeat ARQ with Multiple Receivers," *IEEE INFOCOM'87*, pp.515-524, 1988.
- [28] Z. Rosberg and N. Shacham, "Resequencing Delay and Buffer Occupancy Under the Selective-Repeat ARQ," *IEEE Trans. Inform. Theory*, vol.35, no.1, pp.166-173, Jan. 1989.
- [29] Y. Yoshida, "Resequencing Delay Characteristics of Multilink System in Packet Switched Networks," *ELECTRONICS LETTERS 8th*, vol.26, no.23, pp.1971-

- 1972, Nov. 1990.
- [30] Z. Rosberg and M. Sidi, "Selective-Repeat ARQ: The Joint Distribution of the Transmitter and the Receiver Resequencing Buffer Occupancies," *IEEE Trans. Commun.*, vol.38, no.9, Sept. 1990.
- [31] N. Shacham and B. C. Shin, "A Selective-Repeat ARQ Protocol for Parallel Channels and Its Resequencing Analysis," *IEEE Trans. Commun.*, vol.40, no.4, pp.773-782, April 1992.
- [32] N. T. Plotkin and P. P. Variya, "Performance Analysis of Parallel ATM Connections for Gigabit Speed Applications," *IEEE INFOCOM'93*, vol.3, pp.1186-1193, 1993.
- [33] A. Jean-Marie and L. Gun, "Parallel Queues with Resequencing," *Journal of the ACM*, vol.40, no.5, Nov. 1993.
- [34] J. F. Chang and T. H. Yang, "End-to-End Delay of an Adaptive Selective Repeat ARQ Protocol," *IEEE Trans. Commun.*, vol.42, no.11, Nov. 1994.
- [35] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet Traffic Patterns and Characteristics," *IEEE Network*, vol.11, no.6, pp.10-23, Nov. 1997.
- [36] A. Jean-Marie, M. Tidball, M. Escalante, V. Leoni and H. Ponce de Leon, "On the Influence of Resequencing on the Regularity of Service," *Performance Evaluation* 36-37, pp.115-135, 1999.
- [37] J. G. Kim and M. M. Krunz, "Delay Analysis of Selective Repeat ARQ for a Markovian Source Over a Wireless Channel," *IEEE Trans. Vehicular Technology*, vol.49, no.5, Sept. 2000.
- [38] Y. Xia and D. Tse, "Analysis on Packet Resequencing for Reliable Network Protocols," *IEEE INFOCOM 2003*, vol.2, pp.990-1000, 2003.
- [39] Y. Nebat and M. Sidi, "Resequencing Considerations in Parallel Downloads," *IEEE*, 2002.
- [40] Y. Nebat and M. Sidi, "Analysis of Resequencing in Downloads," *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS*, vol.16, pp.735-757, 2003.
- [41] M. Yoshioka, A. Ohta, and M. Umehira, "A Novel PRIME ARQ for Wireless ATM," *信会技報*, DSP97-152, pp.37-44, 1998.
- [42] Y. Hayashida, A. Maeda, N. Sugimachi, and S. Fujii, "Performance Analysis of Go-Back-N ARQ Scheme With Selective Repeat in Intra-Block," *IEEE Trans. Commun.*, vol.COM-50, no.3, March 2002.

- [43] 藤井俊二, 林田行雄, “コピー伝送選択再送方式の伝送遅延の近似解析とその特性,” 信学論 (B-I), vol.J76-B-I, no.12, pp.967-973, Dec. 1993.
- [44] 藤井俊二, 林田行雄, “コピー伝送選択再送方式の伝送遅延の厳密解析,” 信学論 (B-I), vol.J76-B-I, no.12, pp.974-980, Dec. 1993.
- [45] 藤井俊二, 林田行雄, “マルコフ誤りチャネルにおける選択再送 ARQ 方式の伝送遅延特性,” 信学論 (B-I), vol.J80-B-I, no.5, pp.256-265, May 1997.
- [46] 藤井俊二, 林田行雄, “再送回数に制限のある選択再送 ARQ 方式の厳密解析,” 信学論, vol.J86-B, no.6, pp.876-886, June 2003.
- [47] 藤井俊二, 金子豊久, 林田行雄, “再送回数に制限のあるブロック内選択再送 Go-Back-N ARQ 方式のトラヒック特性,” 信学論, vol.J88-B, no.10, pp.2022-2031, Oct. 2005.
- [48] T. Shikama, Y. Ujihashi, C. Matsuda, and H. Takahashi, “The Layer Model and Error Control for Satellite Communications,” PTC’83, pp.227-234, Jan. 1983.
- [49] 鹿間敏弘, “衛星通信を利用したパケット交換網における誤り回復と順序制御,” 信学会情報ネットワーク研究会, IN87-103, pp.61-66, Dec. 1987.
- [50] 鹿間敏弘, “受信側での順序制御を伴う誤り回復プロトコルの性能,” 電子情報通信学会 部門別全国大会, 546, 1987.
- [51] 鹿間敏弘, “多重選択再送を行う誤り回復プロトコルの順序制御による遅延特性,” 信学会春季全国大会, B-389, March 1988.
- [52] 3GPP TS 25.322 V6.1.0 “Radio Link Control (RLC) Protocol Specification” June 2004.
- [53] H. Koga, T. Ikegami, Y. Hori and Y. Oie, “Out-of-sequence in Packet Arrivals due to Layer 2 ARQ and Its Impact on TCP Performance in W-CDMA Networks” Proc. of 2003 Symposium on Applications and the Internet (SAINT 2003), pp.398-401, Jan. 2003.
- [54] 古閑宏幸, 池永全志, 堀良良彰, 尾家祐二, “Wireless 環境における TCP の性能評価: layer 2 ARQ およびパケット管理による影響,” 信学術報, IN2001-39, pp.19-24, 2001.
- [55] W. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” RFC 2001, Jan. 1997.
- [56] 佐和橋衛, 樋口健一, 新博行, 三木信彦, “W-CDMA 高速パケット無線アクセスと無線リンク特性,” 信学論 B, vol.J84-B, no.10, pp.1725-1745, Oct. 2001.
- [57] 3GPP TR 25.848 V4.0.0 “Physical Layer Aspects of UTRA High Speed Downlink Packet Access,” March 2001.

- [58] M. Rossi and M. Zorzi, "Analysis and Heuristics for the Characterization of Selective Repeat ARQ Delay Statistics Over Wireless Channels," *IEEE Trans. on Vehic. Tecn.*, vol.52, no.5, pp.1365-1377, Sept. 2003.
- [59] R. Chakravorty, S. Banerjee, P. Rodriguez, J. Chesterfield, and I. Pratt, "Performance Optimizations for Wireless Wide-area Networks: Comparative Study and Experimental Evaluation," *ACM MobiCom 2004*, pp.159-173, 2004.
- [60] Y. Zhang, D. Delucia, B. Ryu, and S. Dao, "Satellite Communications in the Global Internet: Issues, Pitfalls, and Potential," *INET '97*, June 1997.
- [61] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP Performance over Wireless Networks," *1st ACM International Conference on Mobile Computing and Networking (Mobicom)*, Nov. 1995.
- [62] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya, "Long Thin Networks," *RFC 2757*, Jan. 2000.
- [63] VINT Project, Network Simulator ns-2,
<http://www.isi.edu/nsnam/ns/>.
- [64] T. R. Henderson and R. H. Katz, "Transport Protocols for Internet-compatible Satellite Networks," *IEEE Journal on Selected Areas in Communications*, vol.17, no.2, pp.345-359, Feb. 1999.
- [65] ITU-T Recommendation Q.2110, "B-ISDN Signalling ATM Adaptation Layer — Service Specific Connection Oriented Protocol (SSCOP)," July 1994.
- [66] 加納貞彦監修, 栗林伸一編著, "やさしい ATM ネットワーク信号方式 -マルチメディア通信のための新しい信号方式-, 電気通信協会, 1996.
- [67] T. R. Henderson and R. H. Katz, "Satellite Transport Protocol(STP): An SSCOP-based Transport Protocol for Datagram Satellite Networks," *2nd International Workshop on Satellite-based Information Services (WOSBIS'97)*, 1997.
- [68] X. Xin, T. Kai, and M. YiFei, "Performance Analysis of Transport Protocol in Satellite Network," *Wcc-ICCT2000*, vol.1, pp.935-941, Aug. 2000.
- [69] 赤池武志, 水戸道晴, 井上一郎, "SSCOP プロトコルのトラヒック特性," *信学技報*, SSE-92-125, pp.1-6, 1993.
- [70] G. Fairhurst and L. Wood, "Advice to Link Designers on Link Automatic Repeat reQuest (ARQ)," *RFC 3366*, Aug. 2002.
- [71] 稲村浩, 石川太郎, 渥美幸雄, 高橋修, "W-CDMA 網でのリンク層 ARQ と TCP の特性評価," *情報処理学会論文誌*, vol.43, no.12. pp.3859-3868, Dec. 2002.
- [72] International Standards Organization, "Information Processing Systems — Data

- Communication High-level Data Link Control Procedure — Frame Structure,” IS 3309, 3rd ed., Oct. 1984.
- [73] International Standards Organization, “HDLC Procedures: Elements of Procedures, Amendment 4: Multi-Selective Reject Option,” ISO 4335/4, 1991.
- [74] ETSI, “Radio Link Protocol for Data and Telematic Services on the Mobile Station — Base Station System (MS-BSS) interface and the Base Station System — Mobile Switching Center (BSS-MSC) interface,” GSM Specification 04.22, Ver. 5.0.0, Dec. 1995.
- [75] V. Jacobson, R. Braden, and D. Borman, “TCP Extensions for High Performance,” RFC 1323, 1992.
- [76] E. Blanton, M. Allman, K. Fall, and L. Wang, “A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP,” RFC 3517, April 2003.
- [77] N. Vaidya, M. Mehta, C. Perkins, and G. Montenegro, “Delayed Duplicate Acknowledgments: A TCP-Unaware Approach to Improve Performance of TCP over Wireless,” Technical Report 99-003, Computer Science Dept., Texas A&M University, 1999.
- [78] C. Parsa and J. J. Garcia-Luna-Aceves, “TULIP: a Link-Level Protocol for Improving TCP over Wireless Links,” IEEE WCNC'99, pp.1253-1257, 1999.
- [79] C. Parsa and J. J. Garcia-Luna-Aceves, “Improving TCP Performance over Wireless Networks,” Mobile Networks and Applications Journal, vol.5, no.1, pp.57-71, 2000.
- [80] J. Postel, “Transmission Control Program,” RFC 793, 1981.
- [81] V. Paxson and M. Allman, “Computing TCP’s Retransmission Timer,” RFC 2988, 2000.
- [82] A. Gurtov and R. Ludwig, “Responding to Spurious Timeouts in TCP,” IEEE INFOCOM '03, vol.3, pp.2312-2322, March 2003.
- [83] R. Ludwig and M. Meyer, “The Eifel Detection Algorithm for TCP,” RFC 3522 (experimental), April 2003.
- [84] 小池昌弘, 桑川一也, 加藤聡彦, 伊藤秀一, “A Study on Behavior of TCP in Spurious Retransmission,” 信学会ソサイエティ大会, B-6-4, Sept. 2006.
- [85] K. Pentikousis, “Wired-Cum-Wireless Environments,” IEEE Communications Surveys, vol.4, no.3, <http://www.comsoc.org/pubs/surveys>, 2000.
- [86] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment

- edgment Options," RFC 2018, 1996.
- [87] A. Aggrwal, S. Savage, and T. Anderson, "Understanding the Performance of TCP Pacing," IEEE INFOCOM, vol.3, pp.1157-1165, 2000.
- [88] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286, May 2002.
- [89] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," RFC 2960, Oct. 2000.
- [90] S. Fu, M. Atiquzzaman, and W. Ivancic, "SCTP over Satellite Networks," IEEE CCW 2003, pp.112-116, Oct. 2003.
- [91] S. Fu and M. Atiquzzaman, "SCTP: State of the Art in Research, Products, and Technical Challenges," IEEE Commun. Magazine, vol.42, no.4, pp.64-76, April 2004.
- [92] S. Fu, M. Atiquzzaman, and W. Ivancic, "Evaluation of SCTP for Space Networks," IEEE Wireless Communications, vol.12, no.5, pp.54-62, Oct. 2005.
- [93] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks," RFC 3481, Feb. 2003.
- [94] R. Chakravorty, S. Banerjee, P. Rodriguez, J. Chesterfield, and I. Pratt, "Performance Optimizations for Wireless Wide-area Networks: Comparative Study and Experimental Evaluation," ACM MobiCom '04, pp.159-173, Sept. 2004.
- [95] R. Ludwig, A. Konrad, and A. D. Joseph, "Optimization the End-to-End Performance of Reliable Flows over Wireless Links," ACM MobiCOM '99, pp.113-119, 1999.
- [96] R. Ludwig and B. Rathonyi, "Link Layer Enhancements for TCP/IP over GSM," IEEE INFOCOM '99, vol.2, pp.415-421, March 1999.
- [97] R. Ludwig and R. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," Computer Communication Review, vol.30, no.1, 2000.
- [98] A-F. Canton and T. Chahed, "End-to-end Reliability in UMTS: TCP over ARQ," IEEE Globecom '01, vol.6, pp.3473-3477, Nov. 2001.
- [99] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Enhancing throughput over wireless LANs using Channel State Dependent Packet Scheduling," IEEE INFOCOM'96, vol.3 pp.1133-1140, March 1996.
- [100] L. B. Jiang and S. C. Liew, "An Adaptive Round Robin Scheduler for Head-of-

- line-Blocking problem in Wireless LANs," IEEE WCNC 2005, vol.2, pp.1219-1224, March 2005.
- [101] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry," RFC 3449, Dec. 2002.
- [102] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, April 1999.
- [103] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 3782, April 2004.
- [104] A. Gurtov and S. Floyd, "Modeling Wireless Links for Transport Protocols," ACM CCR, vol.34, no.2, pp.85-96, April 2004.
- [105] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering," ACM CCR, vol.32, no.1, pp.20-30, Jan. 2002.
- [106] K. Fall and S. Floyd, "Simulation-based Comparison of Tahoe, Reno, and SACK TCP," ACM CCR, vol.26, no.3, pp.5-21, July 1996.
- [107] K. Baharath-Kumar and P. Kermani, "Analysis of a Resequencing Problem in Communication Networks," IEEE INFOCOM, pp. 303-310, April 1983.
- [108] T.-S. P. Yum and T. Ngai, "Resequencing of Messages in Communication Networks," IEEE Trans. Commun., vol.COM-34, no.2, Feb. 1986.
- [109] G. Harrus and B. Plateau, "Queueing Analysis of a Reordering Issue," IEEE Trans. on Software Engineering, vol.SE-8, no.2, pp.113-123, March 1982.
- [110] F. Kamoun, L. Kleinrock, and R. Muntz, "Queueing Analysis of the Reordering Issue in a Distributed Database Concurrency Control Mechanism," Proc. of the 2nd International Conference on Distributed Computing Systems, pp.12-23, April 1981.
- [111] S. Chowdhury, "An Analysis of Virtual Circuits with Parallel Links," IEEE Trans. on Commun., vol.39, no.8, pp.1184-1188, August 1991.

研究業績リスト

学協会誌論文

- [1] 石坂充弘, 鹿間敏弘, “時分割多重型 LAN におけるトークンパッシング方式とその性能評価,” 電子情報通信学会和文論文誌 B-I, vol.J73-B-I, no.1, pp.34-43, Jan. 1990.
- [2] T. Yokotani, H. Sato, S. Nakatsuka, and T. Shikama, “The New Slot Reutilization Schemes in DQDB MAN,” IEICE Transactions, vol.E 74, no.9, pp.2728-2736, Sept. 1991.
- [3] T. Yokotani and T. Shikama, “Bandwidth Allocation for Connectionless Service in Private Networks Based on ATM Technology,” IEICE Trans. Commun., vol.E77-B, no.3, pp.386-395, March 1994.
- [4] 妹尾尚一郎, 厚井裕司, 貞包哲男, 中谷直司, 馬場義昌, 鹿間敏弘, “生体認証によるネットワーク個人認証システム,” 情報処理学会論文誌, vol. 4, no.4, April 2003.
- [5] 鹿間敏弘, 水野忠則, “Selective-Repeat ARQ における順序制御方式の提案と性能評価,” 電子情報通信学会和文論文誌 B, vol.J88-B, no.4, pp. 718-727, April 2005.
- [6] 鹿間敏弘, 水野忠則, “Selective-Repeat ARQ の通番を用いるフロー単位の順序制御方式,” 電子情報通信学会和文論文誌 B, vol.J88-B, no.6, pp. 1017-1028, June 2005.
- [7] 寺島美昭, 別所雄三, 宮内直人, 中川路哲男, 鹿間敏弘, 福岡久雄, 佐藤文明, 水野忠則, “差分更新を実現する分散オブジェクト再構成ミドルウェアの実装と検証,” 情報処理学会論文誌, vol.46, no.9, Sept. 2005.
- [8] T. Shikama, T. Munaka, T. Watanabe, and T. Mizuno, “A TCP-Aware Link Layer Protocol Based on Selective-Repeat ARQ with No Resequencing,” 情報処理学会論文誌, (2005 年 11 月採録決定) .
- [9] T. Shikama, S. Seno, T. Watanabe, and T. Mizuno, “Delay Analysis of the Selective Repeat ARQ Protocol with the Per-Flow Resequencing Scheme,” 情報処理学会論文誌, (2005 年 11 月採録決定) .
- [10] 鹿間敏弘, 荒谷和徳, 渡辺 尚, 水野忠則, “衛星回線でのシャドウイングによる影響を

低減するパケット通信方式,” 静岡大学情報学研究, March 2006(採録決定).

- [11] 山田浩利, 中村 浩, 北市隆一, 古谷信司, 鹿間敏弘, 大島一能, “高スループットを実現する新スケジューリングアルゴリズムを採用した入出力バッファ型パケットスイッチアーキテクチャ,” 電子情報通信学会和文論文誌 B, (条件付採録)

国際会議論文

- [1] H. Matsunaga, C. Matsuda, M. Kajiwara, and T. Shikama, “The Performance of Protocols in the Satellite Channel,” Proc. 5th ICC, pp.445-450, 1980.
- [2] T. Shikama, Y. Ujihashi, C. Matsuda, and H. Takahashi, “The Layer Model and Error Control for Satellite Communications,” PTC’83, pp.227-234, Jan. 1983.
- [3] Y. Ujihashi, T. Shikama, K. Watanabe, K. Oshima, and S. Aoyama, “An Architecture for Connectionless Data Service in B-ISDN,” GLOBECOM’90, pp.1739-1743, Dec. 1990.
- [4] T. Shikama, Y. Ujihashi, and K. Watanabe, “An Architecture for LAN Interconnection in B-ISDN,” CCECE, pp.55.2.1-55.2.4, Sept. 1990.
- [5] A. Otsuka, M. Tsuchida, and T. Shikama, “Loopless Routing Method for Connectionless Mode Packet Switching Networks,” ISITA’90, pp.1027-1030, Nov. 1990.
- [6] T. Yokotani and T. Shikama, “The Erasure Nodes Control Scheme on DQDB,” GLOBECOM’92, pp.1214-1218, 1992.
- [7] T. Yokotani and T. Shikama, “Changeover without Slot Loss of the Erasure Nodes on DQDB,” International Conference on Local and Metropolitan Communication Systems, pp.365-376, Dec. 1994.
- [8] T. Fujino, M. Miyake, K. Murakami, A. Shibuya, F. Ishizu, H. Kubo, T. Shikama, and B. Penther, “Baseband Signal Processing Technologies for 64 Mbit/s Radio Transmission for Mobile Broadband Systems,” ACTS Mobile Commun. Summit ’97, pp.507-512, Oct. 1997.
- [9] T. Fujino, M. Miyake, K. Murakami, A. Shibuya, F. Ishizu, H. Kubo, T. Shikama, B. Penther, F. Munsche, and U. Lambrette, “Design of Baseband Signal Processing Unit in SAMBA Trial Platform for Mobile Broadband Applications,” ACTS Mobile Commun. Summit ’98, June 1998.
- [10] A. Shibuya, N. Fukui, T. Yamauchi, H. Aikawa, N. Kikuchi, K. Murakami, T. Shikama, and B. Penther, “Performance of TDMA Slot Synchronization for Mobile Broadband System Using 40GHz Radio Channel, IEEE Vehicular Technology

Conference, VTC 1999 - Fall, 1999.

- [11] T. Shikama, "New Selective-Repeat ARQ Architecture for High Speed IP Traffic over a Satellite Channel," AIAA-2003-2333, 21st International Communications Satellite Systems Conference and Exhibit, Yokohama, April 2003.
- [12] S. Seno, T. Sadakane, Y. Baba, T. Shikama, Y. Kouji, and N. Nakaya, "A Network Authentication System with Multi-Biometrics," APCC 2003, the 9th Asia-Pacific Conf., vol. 3 pp.914-918, Sept. 2003.
- [13] H. Kawahigashi and T. Shikama, "Estimation of the Number of Radio Channels of Ad-hoc Networks in Analogy to the Coloring Problem," ANTAC 2003, Dec. 2003.
- [14] T. Shikama and T. Mizuno, "Resequencing Schemes for Selective-Repeat ARQ and Their Performance," AINA'05, vol.2, pp.491-494, March 2005.
- [15] T. Shikama, T. Watanabe, and T. Mizuno, "Delay Analysis of the Selective-Repeat ARQ with the Per Flow Resequencing," IEEE ICC 2005, vol.1, pp. 26-32, May 2005.

解説

- [1] 鹿間敏弘, "ISDNの現状と将来展望," 電学誌, vol.110, no.4, pp.301-304, April 1990.

その他論文

- [1] 松永 宏, 松田主税, 鹿間敏弘, 谷口 順, 斎藤幸夫, "新データ網接続端末のインタフェース及び試験装置," 三菱電機技報, vol.53, no.10, pp.754-758, Oct. 1979.
- [2] 高橋寛子, 松永 宏, 松田主税, 鹿間敏弘, 飯田 登, "衛星利用コンピュータネットワーク実験システム - I. システム構成とネットワーク制御方式 -, " 三菱電機技報, vol.55, no.12, pp.837-841, Dec. 1981.
- [3] 藤原謙一, 石井康一, 石坂充足, 関川一彦, 高木 宏, 原田永康, 鹿間敏弘, 小川啓一, 藤田清志, "三菱高速デジタル多重化装置 MX-7100 シリーズの基本方式とシステム構成," 三菱電機技報, vol.61, no.12, pp.10-13, pp.754-758, Dec. 1987.
- [4] M. Ishizaka and T. Shikama, "A Token-Passing Scheme in a Time-Division Multiplexed Local Area Network and its Performance Evaluation," Electronics and Computers in Japan: Part 1, vol.73, no.12, pp.32-44, 1990.

- [5] 松崎 正, 杉本 明, 鹿間敏弘, “高速電力線通信の現状と展望,” 三菱電機技報, vol.78, no.7, pp.440-444, July 2004.
- [6] T. Shikama and T. Mizuno, “A Proposal of the Re-sequencing Scheme for the Selective-Repeat ARQ and Its Performance Evaluation,” Electronics and Communications in Japan: Part 1, (in press).

研究会発表

- [1] 鹿間敏弘, “衛星通信を利用したパケット交換網における誤り回復と順序制御,” 信学会情報ネットワーク研究会, IN87-103, pp.61-66, Dec. 1987.
- [2] 松川康一, 大塚 晃, 鹿間敏弘, 青山 滋, “移動体通信におけるハンドオフを優先したチャネル割り当てに関する一提案,” 回路とシステムに関する第4回軽井沢ワークショップ, 1991.
- [3] 横谷哲也, 鹿間敏弘, “CLSF における処理方式と性能特性,” 日本オペレーションズリサーチ学会 待ち行列研究部会 情報通信ネットワークに関する評価モデルシンポジウム, 1993.
- [4] 村上 謙, 鹿間敏弘, 北山健志, 別所雄三, 横谷哲也, 一番ヶ瀬 広, “ATM-PON システムにおける動的帯域割当制御方式の性能評価,” 信学技報, SSE2000-217, vol.100, no.610, pp.43-48, Feb. 2001.
- [5] 村上 謙, 横谷哲也, 小崎成治, 一番ヶ瀬 広, 鹿間敏弘, “G-PON TC プロトコルの性能評価とマルチサービスのための提案,” 信学技報, NS2002-15, vol.102, No.19, pp.57-60, April 2002.
- [6] 横谷哲也, 村上 謙, 吉田俊和, 鹿間敏弘, 菊地克昭, 大島一能, “ATM-PON 上での VLAN サービスの実現方法,” 信学技報, NS2002-30, vol.102, no.52, pp.17-20, May 2002.
- [7] 北山健志, 横谷哲也, 鹿間敏弘, “APON における上り方向トラヒックの CDV について,” 信学技報, NS2002-31, vol.102, no.52, pp.21-24, May 2002.
- [8] 鹿間敏弘, “Selective-Repeat ARQ プロトコルの順序制御による出力トラヒックのバースト性評価,” 信学技報, NS2002-81, vol.102, no. 202, pp.25-28, July 2002.
- [9] 山田佳子, 牛迫幸雄, 鹿間敏弘, 井上誠也, 井口政昭, 二川憲夫, 木村和宏, “3C1 準天頂システムにおけるハンドオーバ技術,” 第47回宇宙科学技術連合講演会, pp.1312-1316, Nov. 2003.
- [10] 河東晴子, 鹿間敏弘, “地形情報を利用したアドホックネットワーク設計方式,” 信学技報, IN 研究会, IN2003-120, vol.103, no.421, pp.41-46, Nov. 2003.

-
- [11] 宮内直人, 寺島美昭, 鹿間敏弘, “ネットワーク障害の故障診断方式に関する検討,” 信学技報, TM 研究会, TM2003-105, vol.103, no.700, March 2004.
 - [12] 荒谷和徳, 鹿間敏弘, “移動端末の衛星通信のブロッキングによる通信断影響提言のためのパケット通信制御検討,” 信学技報, NS 研究会, NS21003-350, pp.283-286, March 2004.
 - [13] 山田浩利, 中村浩, 北市隆一, 古谷信司, 勝田喬雄, 小林信之, 鹿間敏弘, 大島一能, “10Gbps 8x8 入出力バッファ型パケットスイッチの開発,” 信学技報, IN 研究会, IN2003-331, vol.103, no.692, pp.417-422, March 2004.
 - [14] 寺島美昭, 宮内直人, 鹿間敏弘, 佐藤文明, 水野忠則, “分散オブジェクトシステムの再構築を実現する構成管理方式の検討,” マルチメディア・分散・強調とモバイル (DICOMO2004) シンポジウム, vol. 2004, no.7, pp.683-686, 2004.
 - [15] 鹿間敏弘, 水野忠則, “Selective-Repeat ARQ における順序制御方式の性能評価,” 情報学ワークショップ 2004, pp.232-239, July 2004.
 - [16] 鹿間敏弘, 渡辺 尚, 水野忠則, “フロー単位に順序制御を行う Selective-Repeat ARQ の遅延時間解析,” 信学技報, CS 研究会, CS2004-95, vol.104, no.381, pp.43-48, Oct. 2004.
 - [17] 鹿間敏弘, 水野忠則, “Selective-Repeat ARQ における順序制御方式の提案,” 信学技報, NS 研究会, NS2004-148, vol.104, no.433, pp.17-20, Nov. 2004.
 - [18] 鹿間敏弘, 水野忠則, “SR ARQ の通番を用いるフロー単位の順序制御実現方式の提案,” 信学技報, NS 研究会, NS2004-314, vol.104, no.690, pp.345-348, March 2005.
 - [19] 荒谷和徳, 宮内直人, 鹿間敏弘, “移動端末の衛星通信のブロッキングによる通信断影響低減のためのパケット通信制御検討,” 信学技報, NS 研究会, NS2004-317, vol.104, no.690, pp.357-360, March 2005.
 - [20] 宮内直人, 高野 啓, 寺島美昭, 鹿間敏弘, 中川路哲男, “ネットワーク障害診断ツールの開発,” 信学技報, TM 研究会, TM2004-106, vol.104, no.707, pp.25-30, March 2005.
 - [21] 鹿間敏弘, 渡辺 尚, 水野忠則, “Selective-Repeat ARQ の順序制御による出力トラヒックのバースト性とパケットの順序逆転特性,” 信学技報, CS 研究会, CS2005-11, vol.105, no.241, pp.1-6, August 2005.
 - [22] 鹿間敏弘, 荒谷和徳, 渡辺 尚, 水野忠則, “衛星回線でのシャドウイングによる影響を低減するパケット通信方式,” 情報学ワークショップ 2005, pp.53-58, Sept. 2005.

全国大会発表

- [1] 鹿間敏弘, 田中雄三, 松田主税, “フロー制御のウィンドウ・サイズとバッファ容量についての一考察,” 信学会全国大会, 1323, 1978.
- [2] 高橋寛子, 松田主税, 飯田 登, 梶原 誠, 鹿間敏弘, “CS 利用コンピュータ・ネットワーク実験システムにおける衛星回線シミュレータ,” 信学会通信部門全国大会, 536, 1980.
- [3] 谷口 順, 松田主税, 鹿間敏弘, “マルチマイクロプロセッサシステムにおけるメモリ構成の一考察,” 情報処理学会第 21 回全国大会, 7D-4, 1980.
- [4] 松田主税, 鹿間敏弘, 梶原 誠, “衛星利用データ通信のプロトコル性能評価,” 情報処理学会第 22 回全国大会, 3E-5, 1981.
- [5] 鹿間敏弘, 松田主税, 松永 宏, “衛星利用コンピュータ・ネットワークの方式に関する一考察,” 情報処理学会第 23 回全国大会, 4D-8, 1981.
- [6] 鹿間敏弘, “受信側での順序制御を伴う誤り回復プロトコルの性能,” 電子情報通信学会部門別全国大会, 546, 1987.
- [7] 鹿間敏弘, “多重選択再送を行う誤り回復プロトコルの順序制御による遅延特性,” 信学会春季全国大会, B-389, 1988.
- [8] 杉山直行, 鹿間敏弘, 青山 滋, 岩田 忠, “時分割多重伝送システムの相互接続に関する検討,” 信学会秋季全国大会, B-188, 1988.
- [9] 杉山直行, 鹿間敏弘, 青山 滋, “ビット／オクテット多重化方式による動的帯域割当ての性能評価,” 信学会秋季全国大会, B-240, 1988.
- [10] 鹿間敏弘, 大島一能, 青山 滋, 北村春夫, “企業における広帯域通信網構築に関する検討,” 信学会春季全国大会, B-439, 1989.
- [11] 山中秀昭, 鹿間敏弘, 大島一能, 青山 滋, “企業内 ATM 網における CBO サービスの一検討,” 信学会春季全国大会, B-440, 1989.
- [12] 山中秀昭, 大島一能, 鹿間敏弘, 青山 滋, “分割共通バッファ形 ATM スイッチの検討,” 信学会秋季全国大会, B-192, 1989.
- [13] 山中秀昭, 三浦摂子, 宇治橋義弘, 鹿間敏弘, 大島一能, 青山 滋, “ATM ノード装置の基本機能試作,” 信学会春季全国大会, B-462, 1990.
- [14] 宇治橋義弘, 渡辺 潔, 鹿間敏弘, 青山 滋, “ATM による LAN 間接続の一検討,” 信学会春季全国大会, B-456, 1990.
- [15] 鹿間敏弘, 宇治橋義弘, 渡辺 潔, “ATM を利用した LAN 間接続におけるルーチングとリアセンブル方式,” 信学会秋季全国大会, B-522, 1990.

- [16] 渡辺 潔, 宇治橋義弘, 鹿間敏弘, “ATM による FDDI 相互接続装置の試作,” 信学会春季全国大会, B-456, 1991.
- [17] 横谷哲也, 鹿間敏弘, “DQDB におけるスロット消去ノードの配置制御,” 信学会春季大会, B-434, 1992.
- [18] 横谷哲也, 鹿間敏弘, “CLSF における処理方式と性能特性,” 信学会秋季大会, B-342, 1992.
- [19] 松川康一, 大塚晃, 鹿間敏弘, “移動体通信におけるゾーン半径と呼損率に関する一考察,” 信学会秋季大会, SA-1-6, 1991.
- [20] 佐藤浩之, 青柳秀典, 工藤勝彦, 沢田圭一, 矢野 稔, 鹿間敏弘, 大鳥一能, “150M/600Mbps VPI 変換用 BiCMOS-LSI の開発,” 信学会春季大会, B-498, 1993.
- [21] 宇治橋義弘, 中島宏一, 鹿間敏弘, “並列巡回符号回路 AAL タイプ 3/4 への適用,” 信学会春季大会, B-505, 1993.
- [22] 鹿間敏弘, 山崎研多, “CLSF における交換方式の性能評価,” 信学会秋季大会, B-391, 1993.
- [23] 堀内栄一, 村上 謙, 横谷哲也, 佐藤浩之, 宇治橋義弘, 鹿間敏弘, “B-ISDN における UNI 信号プロトコル試験ツールの開発,” 信学会秋季大会, B-420, 1993.
- [24] 横谷哲也, 村上 謙, 堀内栄一, 佐藤浩之, 宇治橋義弘, 鹿間敏弘, “呼制御機能を備えた ATM 網シミュレータの開発,” 信学会秋季大会, B-421, 1993.
- [25] 沼倉 歩, 佐藤浩之, 上田広之, 小浜茂樹, 鹿間敏弘, “ATM セルトラヒックパタン生成 LSI の開発,” 信学会秋季大会, B-550, 1993.
- [26] 佐藤浩之, 桑原信也, 鹿間敏弘, “フレームリレーによる LAN 間通信の検討,” 信学会秋季大会, B-576, 1993.
- [27] 服部 寛, 土田 充, 鹿間敏弘, “ATM 網によるフレームリレーサービスの性能評価,” 信学会秋季大会, B-578, 1993.
- [28] 志村雅人, 横谷哲也, 佐藤浩之, 鹿間敏弘, “B-ISDN の CLSF におけるアドレス検索方式の検討,” 信学会春季大会, B-506, 1994.
- [29] 村上 謙, 横谷哲也, 鹿間敏弘, “半蓄積交換におけるブロッキングに関する検討,” 信学会春季大会, B-511, 1994.
- [30] 土田 充, 横谷哲也, 福田雅裕, 山中秀昭, 清水圭一, 吉村 博, 鹿間敏弘, “ATM-LAN の機能と技術課題,” 信学会春季大会, SB-5-2, 1994.
- [31] 横谷哲也, 村上 謙, 市橋立機, 鹿間敏弘, “低速 ATM における統計多重に関する特性評価,” 信学会総合大会, B-628, 1995.
- [32] 鹿間敏弘, 小林信之, “Selective-Repeat ARQ の出力パケットを固定帯域で中継する場合の遅延時間,” 2002 年信学会ソサイエティ大会, B-6-26, 2002.

- [33] 山田浩利, 古谷信司, 小林信之, 鹿間敏弘, 村上圭司, “10Gbit/s 8 × 8 入出力バッファ型パケットスイッチの開発,” 2002 年信学会通信ソサイエティ大会, B-6-93, 2002.
- [34] 古谷信司, 山田浩利, 小林信之, 鹿間敏弘, 村上圭司, “高スループットの入力バッファ型スイッチ向け新スケジューリングアルゴリズム,” 2002 年信学会通信ソサイエティ大会, B-6-94, 2002.
- [35] 河東晴子, 妹尾尚一郎, 鹿間敏弘, “位置情報を利用したアドホックネットワークのルーティング,” 2002 年信学会通信ソサイエティ大会, SB-3-8, 2002.
- [36] 河東晴子, 石橋孝一, 妹尾尚一郎, 鹿間敏弘, “自律分散ネットワークのフラッディング方式の一検討,” 2003 年信学会総合大会, B-5-127, 2003.
- [37] 河東晴子, 鹿間敏弘, “地形情報を利用したアドホックネットワーク設計方式,” 2003 年信学会通信ソサイエティ大会, B-5-112, 2003.
- [38] 鹿間敏弘, 水野忠則, “Selective-Repeat ARQ における順序制御方式の提案,” 2004 年信学会通信ソサイエティ大会, B-3-19, 2004.
- [39] 鹿間敏弘, 水野忠則, “SR ARQ の通番を用いるフロー単位順序制御の実現方式提案,” 2005 年信学会総合大会, B-6-77, 2005.

雑誌および講習会での発表

- [1] 鹿間敏弘, “OSI をはじめとする通信プロトコルの解説書,” 日経コンピュータ, 1987.7.6.
- [2] 鹿間敏弘, “ISDN の現状と将来動向,” 第 40 回システム制御情報講習会「ここまできた高度情報化技術」, pp.18-24, Feb. 1991.
- [3] 鹿間敏弘, “B-ISDN 技術とネットワーク,” 日本工業技術センターセミナー, June 1992.