

平成16年度  
研究開発成果報告書

高速高品質コンテンツ配信を実現する自律  
適応型メタコンテンツ・ネットワーク技術  
に関する研究開発

委託先：住友電気工業(株)

平成17年5月

情報通信研究機構

平成16年度 研究開発成果報告書  
「高速高品質コンテンツ配信を実現する自律適応型  
メタコンテンツ・ネットワーク技術に関する研究開発」

目 次

1	研究開発課題の背景.....	1
2	研究開発の全体計画.....	1
2-1	研究開発課題の概要.....	1
2-2	研究開発目標.....	1
2-2-1	最終目標.....	1
2-3	研究開発の年度別計画.....	3
3	研究開発体制 .....	4
3-1	研究開発実施体制.....	4
4	研究開発実施状況.....	5
4-1	最新の高速低遅延順方向誤り訂正技術の国際的調査研究と ネットワーク上の多段利用を想定したクライテリアの確立.....	5
4-1-1	序論.....	5
4-1-2	研究内容.....	5
4-1-3	Digital Fountain 社誤り訂正符号の基本概念.....	6
4-1-4	誤り訂正符号の性能比較.....	8
4-1-5	結論.....	11
4-2	ネットワークのモデル化とシミュレーション技術の開発ならびに 高信頼性コンテンツ配信プロトコルの研究.....	12
4-2-1	序論.....	12
4-2-2	研究内容.....	12
4-2-3	ネットワークのパケットジッタならびに欠損の測定.....	12
4-2-4	インターネットにおけるパケット欠損.....	16
4-2-4-1	実験系.....	17
4-2-4-2	パケット欠損の測定結果.....	19
4-2-4-3	パケット欠損のモデル検討.....	25
4-2-5	結論.....	27
4-3	デバイス化とホームゲートウェイ等評価プラットフォームの試作、 実証試験および I E T F 等へのドラフト提案.....	28
4-3-1	序論.....	28
4-3-2	実証実験内容.....	28

4-3-3	実験結果.....	29
4-3-4	IETF への標準化活動.....	32
4-3-5	結論.....	34
4-4	帯域抑制手段に関する研究、中継装置用超高速低遅延順方向 誤り訂正デバイスの試作・実験.....	36
4-4-1	序論.....	36
4-4-2	研究内容.....	36
4-4-3	IETF の動向.....	36
4-4-4	ライブ FEC サーバの開発.....	38
4-4-4-1	ライブ FEC サーバの仕様.....	38
4-4-4-2	ライブ FEC サーバの構成.....	41
4-4-4-3	ライブ FEC サーバ性能評価試験.....	42
4-4-4-4	エンコード性能評価試験結果.....	43
4-4-4-5	デコード性能評価試験結果.....	45
4-4-5	ライブ FEC サーバの負荷低減を意識したシステム構成に関する考察 .....	46
4-4-6	結論.....	47
4-5	総括.....	48
5	考資料、参考文献.....	49
5-1	研究発表・公演等一覧.....	49

(添付資料)

- 1 研究発表、講演、文献等一覧
- 2 再委託研究報告書(京都大学ならびに大阪大学)



平成16年度 研究開発成果報告書  
「高速高品質コンテンツ配信を実現する自律適応型  
メタコンテンツ・ネットワーク技術に関する研究開発」

## 1 研究開発課題の背景

近年、インターネットによる映像配信の気運が高まっているが、乱れのない映像配信を実現するためには、誤り訂正技術が重要な役割を占めている。これまでの誤り訂正技術は、有限体における多項式演算を用いた手法が主流であったが、復号化処理が符号長の自乗に比例して増大するため、専用のハードウェアを用いて処理する必要があった。高速高品質のコンテンツ配信を実現するためには、ソフトウェア的に処理可能な負荷の軽い誤り訂正技術によって、ネットワークの状況(誤り率)に応じて自律的に適応できるメカニズムを導入することが、重要であると考えられる。

符号理論の歴史は、1948年のシャノンの論文、あるいは1950年のハミング符号の発明から始まる。1960年には、複数のビット誤りを訂正できる BCH (Bose-Chaudhuri-Hocquenghem) 符号やバイト誤りを訂正できる Reed-Solomon 符号が発明された。最近、米国 Digital Fountain 社では、新しい欠損補償符号が開発された。この符号は、バイト誤りを訂正するのではなく、パケット(シンボル)そのものの欠損を補うことの出来る符号であり、これまでの訂正という概念ではなく確率的に元のパケット(シンボル)を復元している。また、この符号は処理時間が符号長に比例するという LDPC (Low Density Parity Check) という誤り訂正符号の流れをくみ、ソフトウェア的に復号可能であるという特徴を持っている。

## 2 研究開発の全体計画

### 2-1 研究開発課題の概要

本研究開発は、最新の超高速低遅延順方向誤り訂正技術を活かした新たな信頼性保証型通信手順によって、従来のフロー制御と廃棄パケットの再送要求やラベルスイッチングによって信頼性を確保するネットワークとはまったく異なる、新たなサービスレベル保証型ネットワークの実現を目指すものであり、基盤技術の研究開発と標準化機関への提案および検証試験により構成される。

### 2-2 研究開発目標

#### 2-2-1 最終目標

- (1) 自律適応型のフロー毎アプリケーション毎サービスレベル毎の順方向誤り訂正機能選択制御機能を持つ多段ネットワークのモデル化とシミュレーション技術の確立
- (2) 自律適応型のフロー毎アプリケーション毎サービスレベル毎の順方向誤り訂正機能選択制御機能を持つ多段ネットワークを実現するための新たなIP上の高信頼性コンテンツ配信プロトコルならびにパケット構造案の提唱
- (3) 最新の超高速低遅延順方向誤り訂正技術の国際的調査研究と、MPEG-1/2/4等のストリーミング配信やダウンロード配信におけるネット

ワーク上での多段処理（エンコード・デコード・トランスコード）に適した同技術のクライテリアの確立

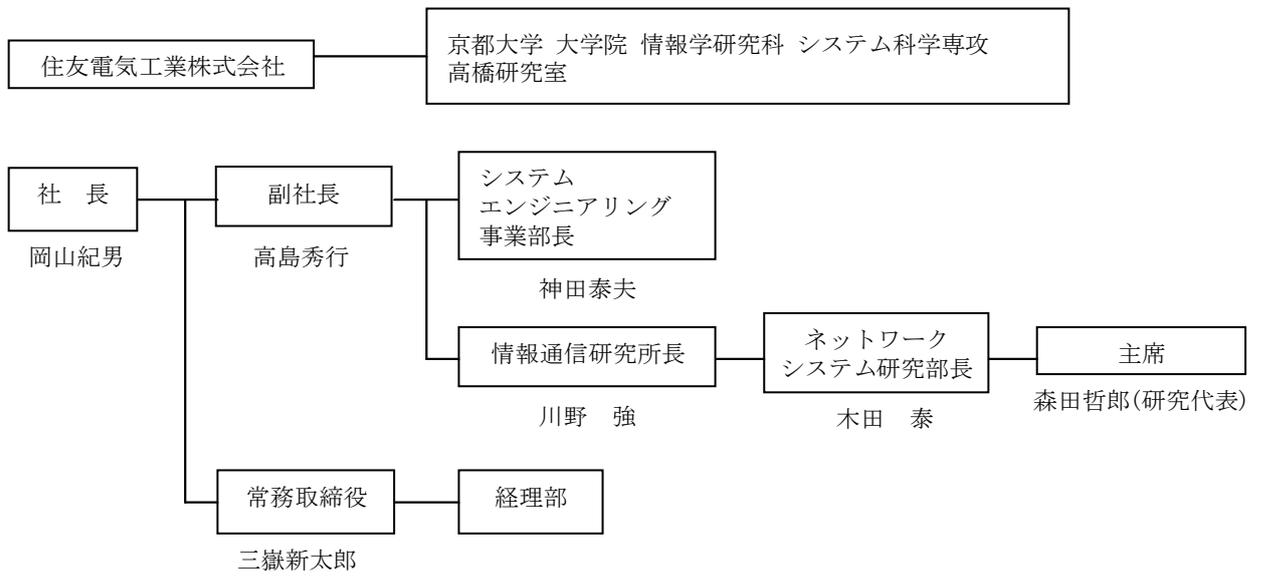
- (4) デファクト標準化を念頭においた、順方向誤り訂正機能およびその選択制御機能のデバイス化と試作評価プラットフォームとしてのサーバ・クライアントおよびホームゲートウェイ等低速低価格中継装置の試作ならびに実証試験
- (5) 実証試験結果に基づく、IETF等への自律適応型のフロー毎アプリケーション毎サービスレベル毎の順方向誤り訂正機能選択制御機能を持つ多段ネットワークを実現するためのあらたな高信頼性コンテンツ配信プロトコルのドラフト提案
- (6) 本研究の信頼性保証型通信手順が、インターネット世界に受け入れられるための、適切な抑制手段に関する研究と提唱。
- (7) 中継装置に適した超高速低遅延の順方向誤り訂正アルゴリズムの調査研究とデバイス試作。
- (8) キャリア・ISP等異業種の参画を得たフィールド実験と有用性の検証

2-3 研究開発の年度別計画

研究開発項目	14年度	15年度	16年度	備考
1) ネットワークのモデル化 <sup>①</sup> とシミュレーション技術の開発 <sup>②</sup> ならびに高信頼性コンテンツ配信プロトコルの研究 <sup>③</sup>	①-----→ ②-----→ ③-----→		-----→	①②③の一部を大阪大学に再委託
2) 最新の高速低遅延順方向誤り訂正技術の国際的調査研究 <sup>④</sup> とネットワーク上の多段利用を想定したクライテリアの確立 <sup>⑤</sup>	④-----→ ⑤-----→	-----→	-----→	④⑤の一部を京都大学に再委託
3) デバイス化 <sup>⑥</sup> とホームゲートウェイ等評価プラットフォームの試作 <sup>⑦</sup> 、実証試験 <sup>⑧</sup> およびIETF等へのドラフト提案 <sup>⑨</sup>	⑥-----→ ⑦-----→	⑧-----→ ⑨-----→	-----→	
4) 帯域抑制手段に関する研究および普及に向けた補完技術に関する研究 <sup>⑩</sup> 、並びに既存の中継装置を補完する高速高品質コンテンツ配信用装置に関する研究と試作 <sup>⑪</sup>		⑩-----→ ⑪-----→	-----→	

### 3 研究開発体制

#### 3-1 研究開発実施体制



## 4 研究開発実施状況

### 4-1 最新の高速低遅延順方向誤り訂正技術の国際的調査研究とネットワーク上の多段利用を想定したクライテリアの確立

#### 4-1-1 序論

インターネット上で、MPEG1/2/4、WMT、Real、QT、MP3等のフォーマットの映像・音声・音楽のストリーミング配信や大容量ファイルの高速ダウンロード配信を行なう場におけるサーバ・クライアント・中継装置を想定したOn\_the\_flyでの順方向誤り訂正エンコード・デコード・トランスコード処理に求められるクライテリアについて整理する。

また、高速低遅延順方向誤り訂正技術に関する最新の国際的研究動向を把握し、本研究開発におけるネットワーク上の利用に適した1つまたは複数の順方向誤り訂正技術候補の選択と、実用化のための課題（ハードウェアを用いた高速化の必要性、チューニングパラメータ、トレードオフの関係にある特性等）について研究する。更に、独自の高速低遅延順方向誤り訂正技術の研究開発の可能性について検討を行なう。

#### 4-1-2 研究内容

誤り訂正符号として、Reed-Solomon 符号を取り上げ、この符号と米国 Digital Fountain 社 Dr. Luby らにより開発された LT 符号について評価検討を行った。LT 符号は、Low-Density Parity-Check (LDPC) 符号の流れをくむ符号であり、復号のアルゴリズムが符号長の一乗に比例する、つまり  $O(n)$  であることを特徴としている。現在、広く用いられている Reed-Solomon 符号は、符号長の自乗に比例する  $O(n^2)$  であり、復号時間を短縮するために専用ハードウェアを使用している。汎用 CPU でソフトウェア的に復号処理ができることが、LT 符号の大きな利点である。ソフトウェア的に処理できるため、広範囲のパケット欠損率に適用することが可能で、今回の研究テーマに対応できる符号であると考えられる。

LDPC (Low-Density Parity-Check) 符号は、シャノンの限界に迫る符号として最近注目を集めている。LT 符号の改良型である Raptor 符号は、LT 符号、LDPC 符号、拡大 Hamming 符号の3つを組み合わせた符号であり、Multi-Stage 符号とも呼ばれている。他の2つの符号と組み合わせた結果、LT 符号において  $10^{-8}$  程度であった復元失敗率が、Raptor 符号では  $10^{-13}$  以下に改良されている。

誤り訂正符号では、バースト誤りに対して、符号長を長くする等の方法を用いて対応することが可能である。ただ Reed-Solomon 符号では、符号長を長くすると、デコードにかかる時間が  $O(n^2)$  であるため、実用的ではない。上記の点を数値計算で比較し、IP ネットワークにおけるパケット欠損の修復において、Digital Fountain 社 LT 符号の改良型である Raptor 符号が、有望である事を確認した。

さらにネットワーク上での(実用化のための)トレードオフに関する研究として、Raptor符号において、ネットワーク上のパケット欠損として二項分布を仮定し、帯域オーバーヘッドを考慮した誤り訂正能力と遅延時間に関する検討を行った。

なお本研究に関して、基礎的な部分は京都大学・高橋研究室に再委託を行った。再委託の研究成果報告書は、本報告書の最後に添付する。

#### 4-1-3 Digital Fountain 社誤り訂正符号の基本概念

符号理論の歴史は、1948年のシャノンの論文、あるいは1950年のハミング符号の発明から始まる。1960年には、複数のビット誤りを訂正できる BCH (Bose-Chaudhuri-Hocquenghem) 符号やバイト誤りを訂正できる Reed-Solomon 符号が発明された。今回取り上げる Digital Fountain 社の誤り訂正符号(LT 符号ならびに、その改良型である Raptor 符号)は、パケット(シンボル)そのものの欠損を補うことの出来る符号であり、これまでの訂正という概念ではなく確率的に元のパケット(シンボル)を復元できるという方法である。

伝送中の欠損パケット(シンボル)を補う方法としては、以下の2つに分類される。

- (1) ARQ (Automatic Repeat reQuest)
- (2) FEC (Forward Error Correction)

ARQ (Automatic Repeat reQuest)は、送信側に欠損した情報の再送を要求することによって欠損パケットを補う方法である。これに対して Forward Error Correction (FEC)方式は、その名前が示す様に再送要求無しで、前方(受信側)で欠損を補う方法である。FEC方式では、送信側で冗長な符号を付加することによって、伝送路でビットエラーやパケット(シンボル)欠損が発生しても、受信側で損失したパケット(シンボル)を回復できる方法である。ARQ方式では、再送要求によるパケットが到着するまでに遅延時間 (RTT : Round Trip Time)の増加が発生するが、FEC方式では遅延が増加しない。一方、本来の情報部分に冗長部分を付与する必要があるため、余分にデータを送信する必要がある。このため「受信時間が長くなる」「余分な伝送帯域が必要となる」などの問題が発生する。また、想定数以上のパケットが欠損すると、復元が不可能になるという問題もある。

本報告書では、 $(n, k, t)$ 欠損復元符号と呼ぶ。欠損復元符号の能力を比較する場合、出来るだけ短い冗長部分 $(n-k)$ で、出来るだけ多くの欠損を復元することが望ましい。従って、評価パラメータとしては、 $t/(n-k)$ が出来るだけ大きい方が良いことが分かる。ただ、 $t$ は $n-k$ を超えられないという Singleton 限界が存在するため、 $t/(n-k)$ の値は、1に近いほど復元能力が高いことが分かる。しかし、単に $t/(n-k)$ だけでは、議論できないため注意が必要である。

図4-1-3-1に、Digital Fountain (DF)社で開発されたLT符号のアルゴリズム(送信側)を示す。この例では、入力データである元のシンボル数は、a~hの8個で、一定数の列を発生するマトリックスによって選ばれたシンボル同士で排他的論理和を取り、これをメタコンテンツとして送信する。平均的な重みは、受信側での復元率が高まるように設定する必要がある。またこの演算の関係は、図4-1-3-1(右)に示すように、グラフ表示することも可能である。例えば、a xor g という出力シンボルは、入力シンボルaとgとに繋がっており、この両者の排他的論理和で構成されていることを示している。また、この接続線の数は、「排他的論理和演算を行った回数+受信メタコンテンツ数」を示しており、符号化/

復号化に必要な処理時間と関係していることが分かる。

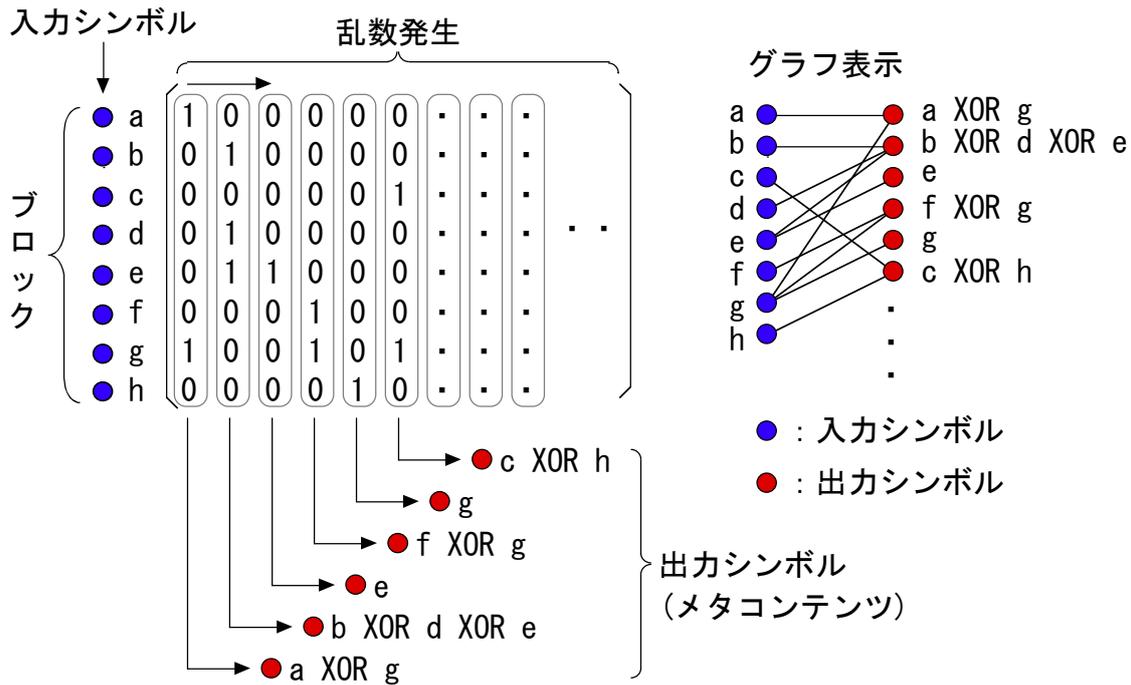
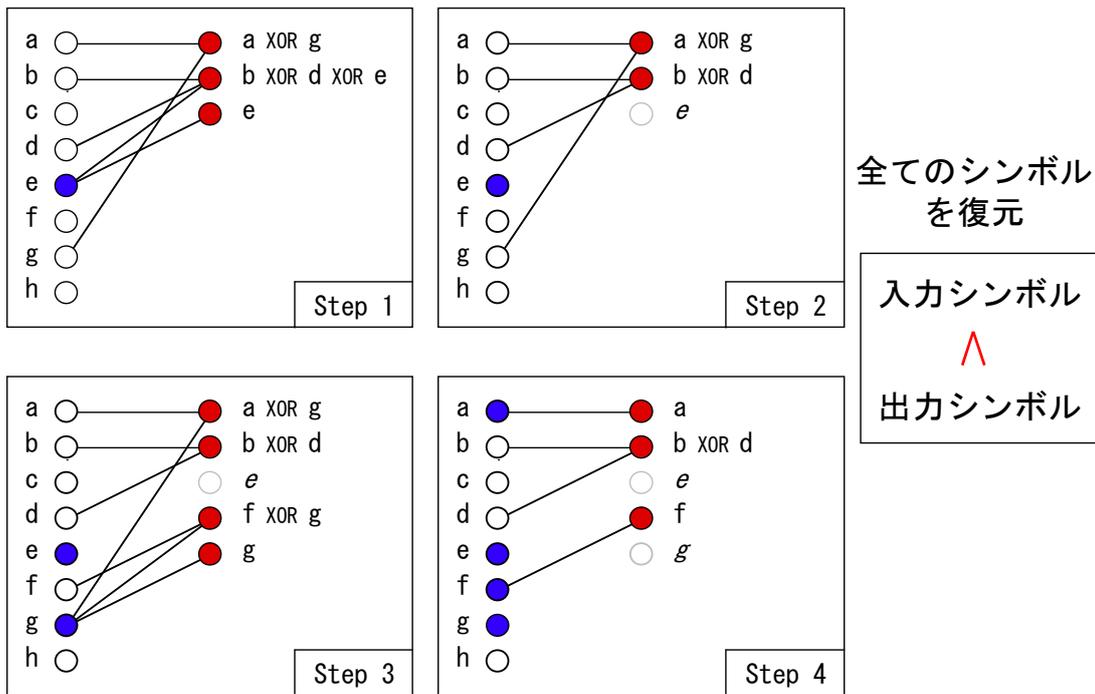


図4-1-3-1 LT 符号の符号化アルゴリズム

図4-1-3-2は、受信側での復号アルゴリズムである。図4-1-3-1で生成されたメタコンテンツシンボルのうち、最初の2つを受信した状態では、どの入力シンボルも復元することは出来ない[Step1]。ここで、重み1のシンボルであるeを受信すると、元のシンボルeを復元することが出来ると同時に、eというシンボルを含んだ他のメタコンテンツシンボルの重みを1つ下げることが出来る。この例では、 $b \text{ xor } d \text{ xor } e$ というメタコンテンツを、 $b \text{ xor } d \text{ xor } e \text{ xor } e = b \text{ xor } d$ という演算によって、 $b \text{ xor } d$ にすることが出来る[Step2]。また排他的論理和演算によって、接続線の数も減少する。さらに、gという重み1のメタコンテンツを受信することによって、gを参照している $a \text{ xor } g$ 、 $f \text{ xor } g$ という2つのメタコンテンツから、aならびにfを復元することが出来る[Step3, 4]。この処理を続けることによって、最終的に全てのシンボルを復元することが可能となる。ただこれは確率的な処理であり、何個のメタコンテンツを受信する必要があるかは、一意には決まらない。メタコンテンツは連立方程式であるため、少なくとも元のシンボル数以上のメタコンテンツシンボルを受信する必要があるが、その中には同じメタコンテンツシンボルを2回以上重複して受信している場合があり、必要数は確率的に決定される。



○ : 未復元シンボル      ● (赤) : 出力シンボル      ● (青) : 復元された入力シンボル

図4-1-3-2 LT 符号の復号化アルゴリズム

#### 4-1-4 欠損補償符号の性能比較

今回我々が取り上げた Digital Fountain 社の欠損補償符号(LT 符号と、その改良型である Raptor 符号)と、一般に広く用いられている Reed-Solomon 符号の性能について比較を行う。それぞれの符号は、符号ごとに異なる特徴を持ち、用途に対して性能が異なるため、直接的に比較することは難しい。今回は、送出されたパッケージが欠損した場合の復元能力という点で、Reed-Solomon 符号と Raptor 符号を比較する。

表 4-1-4-1 は、Raptor 符号と Reed-Solomon 符号を比較した結果を、簡単にまとめたものである。Reed-Solomon 符号は、連続したビット誤り(つまりバイト誤り)を訂正するように設計された符号であり、消失したパッケージを復元する機能はない。また符号化/復号化に際しては、有限体の中で多項式演算を行うことによって、誤り訂正に必要な冗長部分を計算するため、必要な処理時間が情報長の自乗に比例するという特徴を持っている。処理時間がかかると、符号化/復号化の処理が終了しないうちに次のデータが到着するという問題が発生する。従って、処理速度が情報長に比例する( $O(n)$ である)Raptor 符号が、処理能力において優れていることが分かる。

表 4-1-4-1 誤り訂正符号の比較

	Raptor符号	Reed-Solomon符号
符号化による効果	欠損パッケージを復元	バイト誤りを訂正

符号化／復号化	ソフトウェア $O(n)$	ハードウェア $O(n^2)$
設定変更	容易	困難

平均的なパケット欠損率を横軸に取り、ブロック化を行った場合のブロック復元失敗率を縦軸に取って比較することにより、誤り訂正(欠損補償)符号の能力を把握することが出来る。誤り訂正を行わない場合は、パケットの(平均)欠損率とブロックの復元失敗率の関係は、傾き1の直線になるような気がするが、実際には、ブロック内に含まれるパケット数  $N_s$  (統計でいうサンプル数)により、復元失敗率が増加することが分かる。パケットの(平均)欠損率(統計学でいう母集団の欠損率)を  $f_p$  とすると、誤り訂正を行わない場合のブロック復元失敗率  $f_B$  は

$$f_B = 1 - (1 - f_p)^{N_s}$$

となり、 $f_p$  が小さい時には、

$$f_B \cong N_s f_p \quad (f_p \ll 1)$$

$f_p$  が1に近づくと、

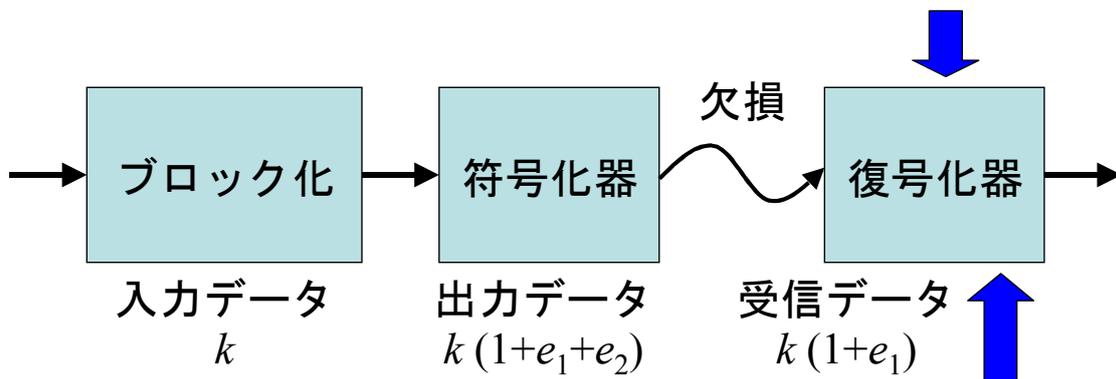
$$f_B \rightarrow 1 \quad (f_p \rightarrow 1)$$

となることが分かる。受信側で誤り訂正を行うと、ブロック復元失敗率  $f_B$  は低下し、この結果から符号の誤り訂正能力を求める。

Raptor 符号では、入力データをシンボルという単位に分割し、そのシンボル同士の排他的論理和(XOR)を取ることによって、出力シンボルを作成し、それをパケットとしてまとめて送出している。さらにこのパケットをブロック化することによって、パケット欠損に対応している。特別な場合は、1シンボルを1パケットで送出し、この場合シンボルとパケットは、同等のものと考えることが出来る。受信側に必要な出力シンボル数は、入力シンボル数以上必要となり、([余分に必要なシンボル数]/[入力シンボル数]-1)をオーバーヘッドと呼ぶ。今回の計算では、オーバーヘッドが5%の場合、 $10^{-13}$ の失敗率で元の入力シンボルを復元できると仮定して計算を進める。

図4-1-4-2に、Raptor符号を用いた場合の伝送系を模式的に示している。まず設計パラメータとして、伝送路での平均的なパケット欠損率を設定する。さらにブロックを構成するパケット数を決めると、ブロック内で発生する欠損パケット数の分布が決まる。この分布は、二項分布であることが知られている。ブロックサイズを大きくすると、ブロック内での欠損パケット数の分布は、平均的なパケットの欠損率に収束する。つまり欠損パケット数の分布は、インパルス関数となる。ブロック内のパケット数を少なくすると、二項分布が広がり、伝送路に設定された(平均パケット欠損率)以上のパケット欠損率が発生する割合が増加し、欠損パケットを復元できない確率が増大する。

復元可能数を越える欠損が発生する確率： $f_1 = f(p, n, t)$



Raptor符号自体の復元失敗率： $f_2 = 10^{-13}$

初期オーバーヘッド： $e_1 = 5\%$

図 4-1-4-2 Raptor 符号を用いた伝送系

Reed-Solomon 誤り訂正符号は、誤り訂正個数  $t$  と情報長  $k$  ならびに符号長  $n$  との間に、下記のような「Reiger の限界式」が存在している。

$$t = \frac{n - k}{2}$$

Reed-Solomon 符号の誤り訂正能力  $t/n$  は、以下のように表される。

$$\frac{t}{n} = \frac{(n - k) / 2}{n} = \frac{1}{2} \left(1 - \frac{k}{n}\right) = \frac{1}{2} (1 - R)$$

また Raptor 符号の場合は、以下のようになる。

$$\frac{t}{n} = \frac{n - (1 + e_1)k}{n} = 1 - (1 + e_1) \frac{k}{n} = 1 - (1 + e_1)R$$

つまり市販の Reed-Solomon 符号用 IC チップ(ハードウェア構成の IC チップ)は、誤り訂正用の冗長部分の半分の誤りしか訂正できない。この結果は、符号化率を 0 に近づけても全体の半分しか、誤りを訂正できないことを意味する。一方 Raptor 符号は、符号化率を小さくすることで、符号全体の誤りを訂正することが出来る。ただ Raptor 符号には、初期オーバーヘッドが存在し、この値によって符号化率が影響を受ける。

図 4-1-4-3 は、Raptor 符号の初期オーバーヘッドをパラメータとして、Reed-Solomon 符号と Raptor 符号の誤り訂正能力を比較したものである。Raptor 符号の初期オーバーヘッドを 5%と仮定すると、符号化率が 90%以下では、Raptor 符号の方が、高い誤り訂正能力を有していることが分かる。逆に、初期オーバーヘッドが大きくなると、Reed-Solomon 符号にとって有利な領域が拡大することが分かる。この結果から符号化率を 0 に近づけると、Raptor 符号の方が有利で

あることが分かる。ただ上記の比較は、ハードウェア化された誤り訂正用 Reed-Solomon チップを用いて、欠損復元を行った場合である。欠損復元用のソフトウェア Reed-Solomon を用いた場合には、Reed-Solomon 自体が Singleton 限界を満たす符号であるため、5%のオーバーヘッドを必要とする Raptor 符号は、必ず下に来ることになる。

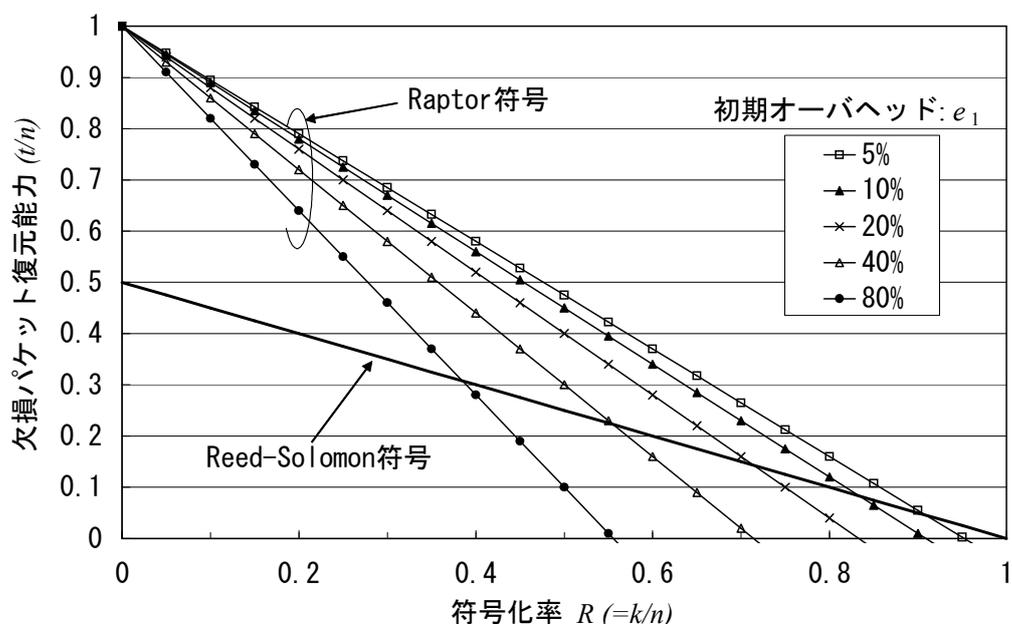


図 4-1-4-3 符号化率と誤り訂正能力の関係

#### 4-1-5 結論

誤り訂正符号として、ハードウェア組み込み型のReed-Solomon符号を取り上げ、この符号と米国Digital Fountain社Dr. Lubyらにより開発されたLT符号について評価検討を行った。LT 符号は、Low-Density Parity-Check (LDPC) 符号の流れをくむ符号であり、復号のアルゴリズムが符号長の一乗に比例する。直接的に両者の誤り訂正能力を比較することは出来ないためReed-Solomon符号はインターリーブを用いると、下記のような点で、Raptor符号の方が有利であることが分かる。

今回の評価で、Digital Fountain社のRaptor符号(LT 符号の改良型欠損符号)が、有望であるという結果を得た。

なお本研究に関して、基礎的な部分は京都大学・高橋研究室に再委託を行った。再委託の研究成果報告書は、本報告書の最後に添付する。

## 4-2 ネットワークのモデル化とシミュレーション技術の開発ならびに高信頼性コンテンツ配信プロトコルの研究

### 4-2-1 序論

メタコンテンツ型の順方向誤り訂正技術を、所定のアプリケーションに対して、フロー毎にサービスレベルに応じて適用するネットワークモデルにおける QoS のシミュレーション技術の研究開発と検証を行なう。

ブロードバンドコンテンツのライブあるいはオンデマンド配信サービスを提供するネットワークシステムにおいて、順方向誤り訂正機能を有さないサーバ、順方向誤り訂正機能を有するサーバ、順方向誤り訂正機能を有さないクライアント、および、順方向誤り訂正機能を有するクライアントが混在する可能性のある、ヘテロジニアスなネットワーク環境下において、所定のアプリケーションに対してフロー毎に、サービスレベルに応じた順方向誤り訂正機能を自律適応的に、あるいは、利用者の要求に応じて、選択的に適用・制御するためのプロトコル（パケット構造および通信シーケンス）案に関する、研究・開発をおこなう。

### 4-2-2 研究内容

多段ネットワークのモデル化を行う第一歩として、一段のネットワークモデルを考え、このシミュレーションを行っている。ネットワーク上にサポートサーバと呼ばれる機器を設置し、このサーバによってネットワーク上で発生したパケット損失を補うというモデルである。この結果を、電子情報通信学会全国大会で発表を行った。

また、Reed-Solomon 符号と DF 社の Raptor 符号の処理速度を比較し、その差を比較した。また、インターネット上で発生するパケット欠損とジッタの関係を調査し、その関係をモデル化することを試みた。

### 4-2-3 ネットワークのパケットジッタならびに欠損の測定

実際のインターネット環境で、パケットの欠損率とジッタ値を測定した。図 4-2-1 に、実験系の構成を示す。表 4-2-1 は、使用した機器の一覧である。まず 1028byte の UDP パケットを作成し、トラフィックジェネレータ (Sniffer 1) から 250msec 間隔で 10,000 個のパケットを送出した (約 32.896Kbps)。受信側は BB ルータで、Ethernet ならびに光回線の伝送速度は 100Mbps である。

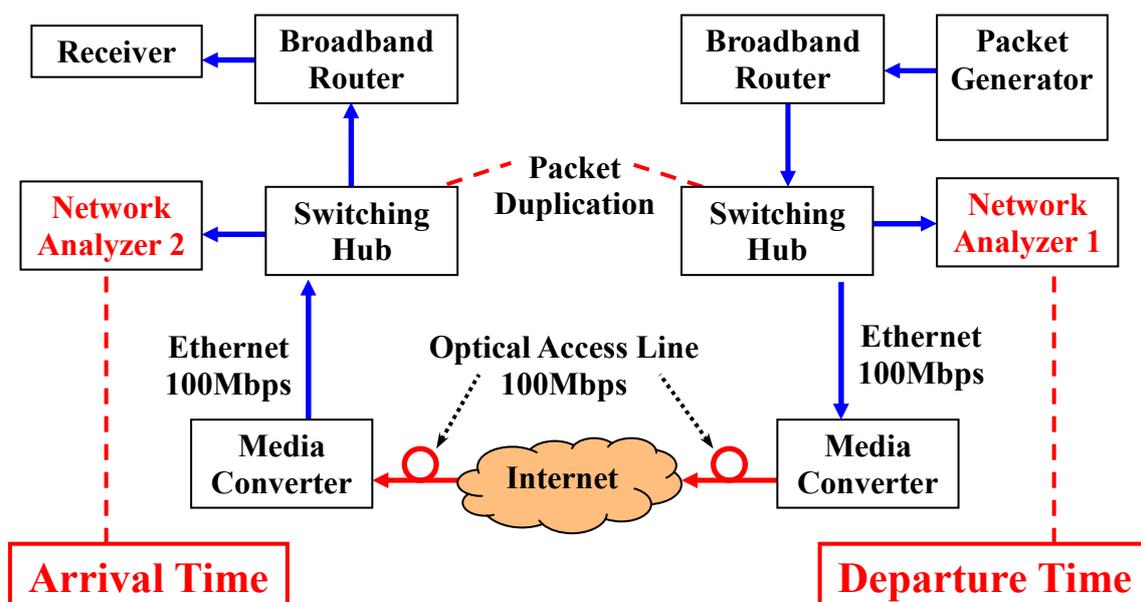


図 4-2-1 実験系の構成

表 4-2-1 使用した機器

	機器名称
BB ルータ	Corega BAR SW-4P Pro (10BASE-T/100BASE-TX に対応した WAN ポートと 4 ポートのスイッチング HUB)
Cat 2950	Cisco Catalyst 2950 (WS-C2950-24) ワイヤスピード : 3.6Mpps

図 4-2-1 の実験系で、Sniffer 2 ならびに Sniffer 3 でのパケット到着時刻を測定し、この両者を比較することによって、Internet のパケット欠損率ならびにジッタ値を測定した。Sniffer 1 からは、10,000 個のパケットが送出されるため、Sniffer 3 でのパケット到着数と比較すると、Internet でのパケット欠損数が分かる。今回の測定では、65 個のパケットが欠損し、欠損率は、0.65%であることが分かる。

図 4-2-2 は、ジッタを測定する方法を示したものである。パケットの出発時刻 (Sniffer 2 への到着時刻) を縦軸に取り、パケットの到着時刻 (Sniffer 3 への到着時刻) を横軸に取る。ジッタがなく、出発時間と到着時間を測定する装置のクロックが一致していれば、両者の関係は傾き 1 の直線で表される。ジッタがあると、結果は折れ線グラフとなる。この折れ線グラフから、一次の最小自乗法を用いて理想的な到着直線を算出する。この直線の傾きは、クロックの違い等によって 1 からずれる。実際の到着時刻と理想的な到着直線からの横方向のずれが、遅延時間 (ジッタ) を表す。この折れ線の傾きは、(パケットが等間隔で送出されている場合には) 伝送速度の大きさに比例し、傾きが緩やかな場合は、伝送速度が低下していることを示す。

このグラフでは、横方向のずれである遅延時間をグラフ上に表現することが難しいので、その値を差し引いて図 4-2-3 (下) の様に、縦軸をジッタ値に変更する。今後は、このグラフを用いてジッタを評価する。到着時刻の測定に Sniffer 3 を用いているため、±0.6msec 程度の測定誤差が発生するが、パケットの送出

間隔が 250msec と大きいので、実験結果に大きな影響を与えないと考えられる。

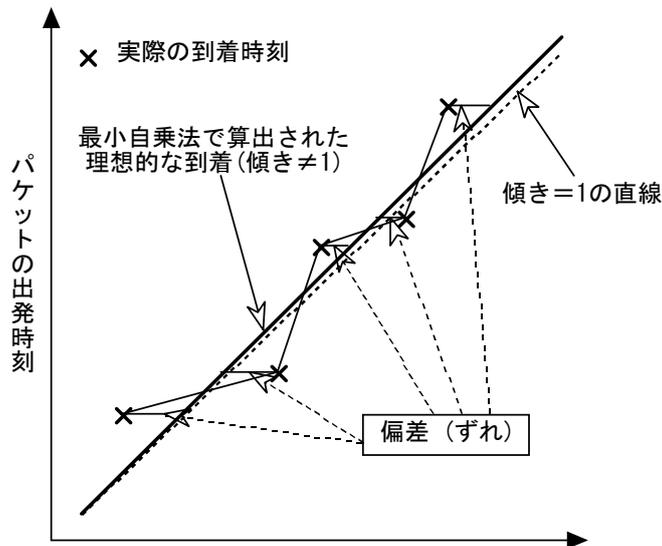


図 4-2-2 ジッタの測定方法

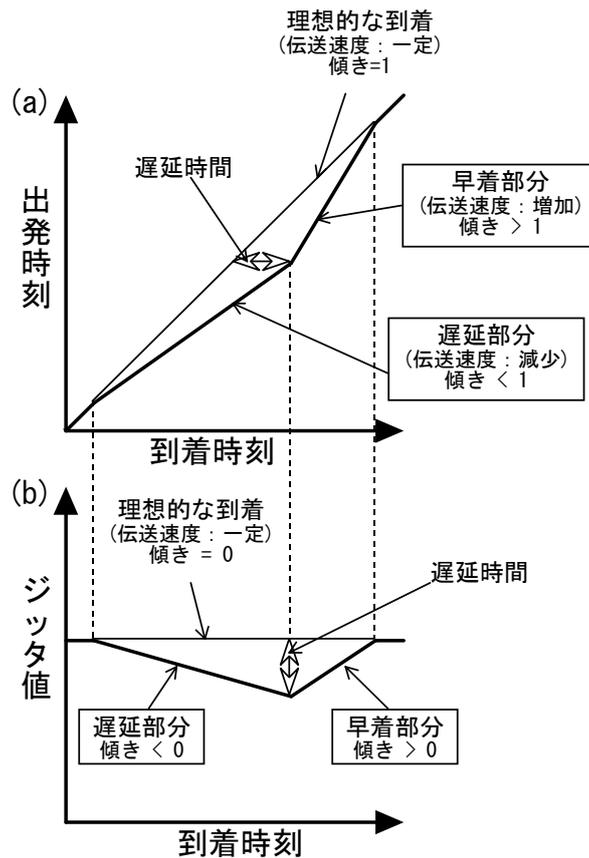


図 4-2-3 ジッタの評価方法

図 4-2-4 は、測定結果全体を示したものである。全部で 10,000 個のパケットを送出しているので、2500 秒=41 分 40 秒の測定時間となる。ジッタ値が正の

側に振れているのは、パケットが理想的な到着時刻より早く到着していることを示している。測定時間 2,500 秒の間に、±10msec の緩やかなジッタがあることが分かる。ジッタ値が 0 のところ(横軸上)に示されている白丸は、その付近でパケットが欠損したことを示している。パケット自体が消滅しているため、その到着時刻を知ることは出来ないため、最小自乗法で求めた直線の式にパケットの出発時刻を代入し、理想的な到着時刻を求めた。

ジッタ値が小さいところ(正の領域)では、途中の経路上でのバッファに溜まるパケット数が少ないことに対応しているが、この領域でも多くのパケット欠損が発生していることが分かる。例えば、800~900 秒付近では、遅延時間が極小になっているが、この部分で集中的にパケット欠損が発生していることが分かる。ネットワークの混雑が緩和している領域で、多くのパケットが欠損しているため、このメカニズムを調べる必要がある。また図 4-2-4 には、緩やかなジッタ以外に、突発的に遅延時間が増加していることが分かる。これは、他のユーザからのパケットが急激に増加し、遅延時間が増加したと考えられる。しかし、この短期的なジッタとパケット欠損も、対応していないと思われる。

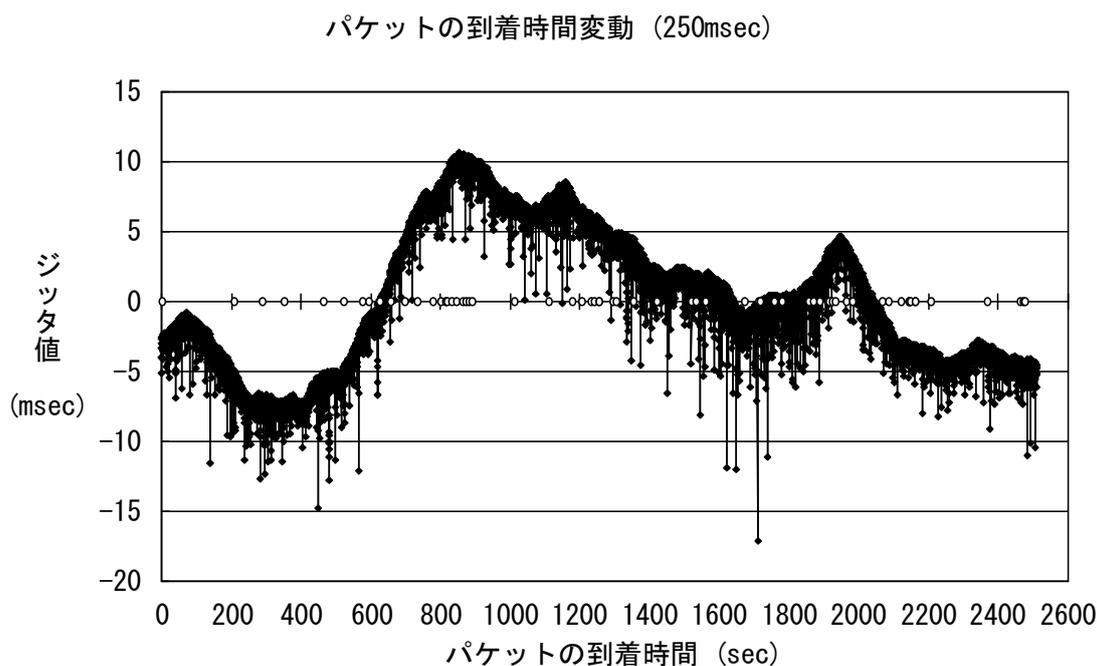


図 4-2-5 は、インターネットでのパケット欠損を模式的にあらわしたものである。水の流れは、パケットストリームに相当し、タンクはルータに設置されたバッファに相当する。バッファのサイズは、大きいタイプ(B)と小さいタイプ(A, C)の2種類ある。大きいタイプ(B)は、幹線系に設置されたルータのバッファに相当し、小さいタイプ(A, C)は支線系に設置されたルータのバッファに相当する。濃い色で示された「My Stream」は、A, B, Cの3つのタンクを經由し、外に流れ出す。A, B, Cのタンクを經由する際に、他のストリームと合流、分離を繰り返す。例えば、タンク A に流れ込む水流は、支線系であるためユーザ数

が少なく、水流が大きく変化する。またタンク A の大きさも、幹線系にあるタンク B と比べて小さいため、A1 からのストリームの変化に対応できず、タンクから水が溢れることになる。これが、パケット欠損に相当する。一方、幹線系に位置するタンク B は、統計多重効果によって B1 ストリームの変化が小さく、タンクのサイズも大きいので、バッファの溢れが発生しにくい。図 5-2-5 の結果と比較すると、緩やかなジッタ値の変化は、タンク B に溜まる水量の変化だと考えられ、突発的に発生するジッタの変化は、支線系にあるタンク A, C に溜まる水量の変化であると考えられる。タンク A, C からのバッファ溢れは、A1, C1 からの水流の変化によって発生し、突然発生するため、図 4-2-4 の緩やかなジッタの変化から予測することは困難である。

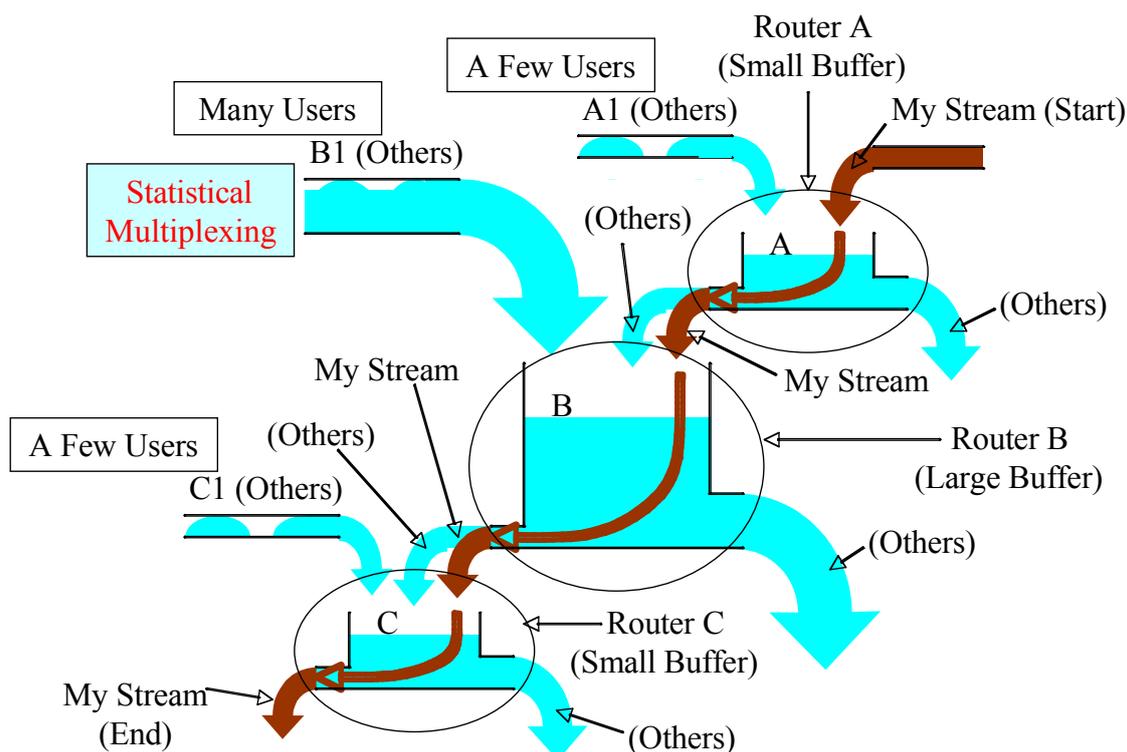


図 5-2-4 インターネットでのジッタ発生モデル

#### 4-2-4 インターネットにおけるパケット欠損

インターネット上で映像を配信する際にパケット欠損が映像品質に大きな影響を与えることは良く知られているが、実際のパケット欠損率やパケット欠損パターンに関してはあまり知られていない。また欠損を復元して映像品質を良好に保つために FEC は有力な手段ではあると考えられるが、FEC を有効に適用するためにはパケット欠損状況を把握する必要がある。本節では、以下の目的から実インターネットにおけるパケット欠損の測定を行った。

1. 実ネットワークにおけるパケット欠損率のレベル・変動の測定
2. FEC の適用に重要なバーストパケット欠損の測定
3. パケット欠損のモデル化

#### 4-2-4-1 実験系

測定系の概要を図 4-2-4-1 に示す。調査対象は、100Mbps の光回線「B フレックス」とした。ISP(Internet Service Provider)については、現状の映像コンテンツ配信サービスの多くが ISP が自社の顧客に対して提供する形で、同一 ISP 内のネットワークにおけるデータ送受信となることから、送信側・受信側ともに同一の ISP を選択した。本測定では大阪と東京の拠点にパケット送受信用のサーバを配置し、2 拠点間のネットワークの状態を測定することにした。これは、一つの拠点で同一 ISP 内の通信を行った場合、ISP 内の直近(BB ルータの接続先)もしくは近接のルータで折り返されるなど、実インターネット環境の多段のルータをほとんど通過しない短絡したネットワーク構成になる可能性があるためである。

次に使用した機器(表 4-2-4-1)の動作について述べる。まず一方のサーバを送信モードで動作させ、RTP パケットを一定速度(6Mbps)で送信する。他方のサーバは受信モードで動作させ、パケットの IP アドレスとポート番号から送信サーバから送られてきたパケットを判定し、パケットのタイムスタンプと RTP シーケンス番号及び受信時刻を 1 パケットごとにメモリに保存する。一定時間(約 20 分間)パケットを送信後、送信・受信を中断し、受信側では SDRAM に記録したデータをハードディスクに保存する。各サーバはプログラム停止しない限り以上の動作を 20 分単位で繰り返す。一定期間測定後、受信側に保存されたデータを回収し、パケット欠損率やバースト欠損長などを解析した。

測定条件については表 4-2-4-2 に記載する。測定期間は 2004 年 6 月から 10 月初旬までの約 4 ヶ月間であり、最初の 1 ヶ月については東京から大阪に向けてデータを送信し、7 月以降は大阪から東京に向けてデータの送信を行った。

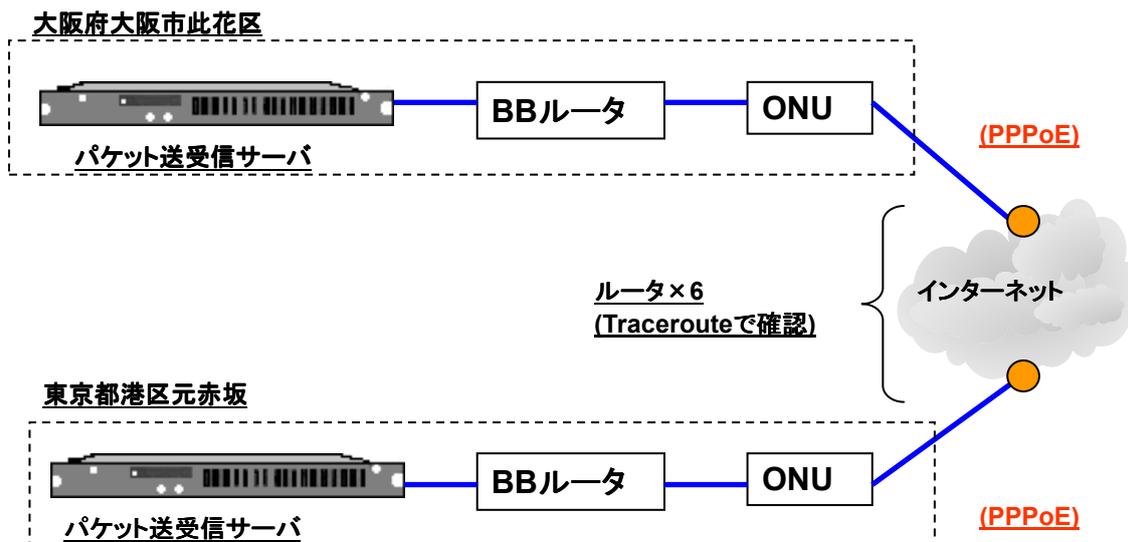


図 4-2-4-1 パケット欠損測定系

表 4-2-4-1

使用機器	仕様
パケット送受信サーバ	<p>[ハードウェア仕様]            HP 製 ProLiant DL360 G3            CPU : Intel 製 Xenon プロセッサ 3.2GHz            メモリ : 1GB, HDD : 36.4GB</p> <p>[ソフトウェア仕様]            OS : Redhat Linux 8.0            パケット送受信ソフト            送信モードと受信モードがあり、どちらか一方で動作可能。            ・送信モード時            指定したビットレートで固定長のパケットを送信            ・受信モード時            送信側から送られてきたパケットにキャプチャ時のタイムスタンプを付加してシーケンス番号部分を保存</p>

表 4-2-4-2

	測定条件
測定期間	2004年 6月～2004年 10月 6月6日～7月1日 : 東京 => 大阪 7月25日～10月6日 : 大阪 => 東京
送信データ	プロトコル : RTP/UDP/IP パケット長 : 1086Byte, ビットレート : 6Mbps

#### 4-2-4-2 パケット欠損の測定結果

##### (1) パケット欠損率の推移

###### (1-1) 概要

6月～10月の測定により得られた結果を、1週間ごとに得られた結果を示す。詳細データは、添付資料参照。

##### 1. パケット欠損率

横軸：時間、縦軸：個別パケット欠損率

約20分間の測定(総送信パケット数 782,609)について送信パケット数と受信パケット数から算出された個別パケット欠損率をプロットした。ただし、1回の測定中にパケット欠損が発生しなかった場合には1.0E-7にプロットした。横軸は左端をその週の日曜日・午前0時としている。

東京から大阪へパケットを送信する場合を測定 A

大阪から東京へパケットを送信する場合を測定 B

とし、測定データと測定期間の対応を次の表 4-2-4-2-1 に示す。

平均欠損率①：個別パケット欠損率の1週間の平均値

平均欠損率②：個別パケット欠損率のうち、10%を超える測定結果を除いた1週間の平均値

表 4-2-4-2-1 測定期間

A-1	2004年6月第2週(6/6～6/12)
A-2	2004年6月第3週(6/13～6/19)
A-3	2004年6月第4週(6/20～6/26)
A-4	2004年6月第5週(6/27～7/1)
B-1	2004年7月第4週(7/25～7/31)
B-2	2004年8月第1週(8/1～8/6)
B-3	2004年8月第3週(8/16～8/21)
B-4	2004年8月第4週(8/22～8/27)
B-5	2004年8月第5週(8/29～9/4)
B-6	2004年9月第1週(9/5～9/11)
B-7	2004年9月第2週(9/12～9/17)
B-8	2004年10月第1週(10/1～10/2)
B-9	2004年10月第2週(10/3～10/6)

上記の結果の中で、本文中には A1 と B1 の結果のみを示す。詳細データは、添付資料参照。

###### (1-2) 測定結果

A. 伝送方向 東京 => 大阪

(A-1)

測定期間：2004年6月第2週(6/6～6/12)

平均欠損率①  $1.44E-3$

平均欠損率②  $4.95E-5$

特異点：6月12日 01:00 開始の測定でパケット欠損率 68.9%

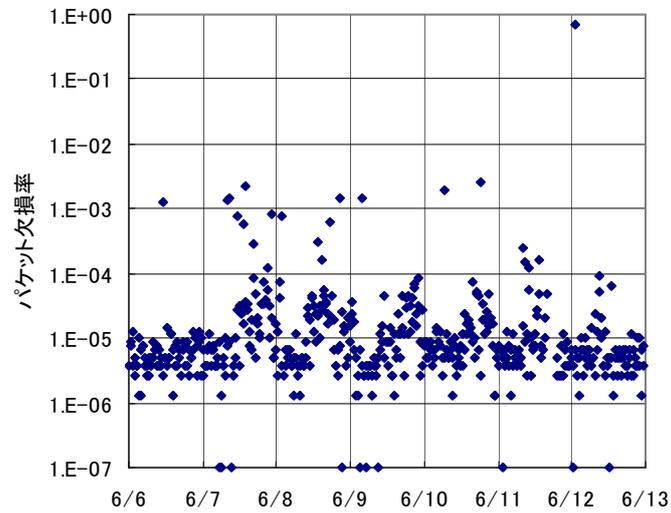


図 4-2-4-2 パケット欠損率の推移

B. 伝送方向 大阪 => 東京

(B-1)

測定期間：2004年7月第4週(7/25～7/31)

平均欠損率① 6.99E-3

平均欠損率② 3.29E-5

特異点：7月25 19:00～の測定でパケット欠損率 91.1%

7月27 02:00～の測定でパケット欠損率 91.2%

7月28 18:00～の測定でパケット欠損率 82.4%

7月30 10:00～の測定でパケット欠損率 81.9%

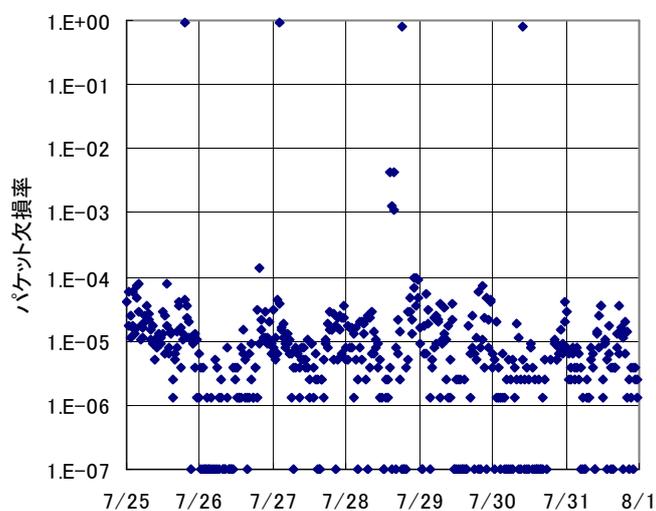


図 4-2-4-3 パケット欠損率の推移

(1-3) 測定結果の考察

(1-3-1) 東京から大阪へパケットを送信した場合

・測定日、時期による差異が大きく、時間帯による変動（図 4-2-4-5）は、明確な傾向は得られなかった。

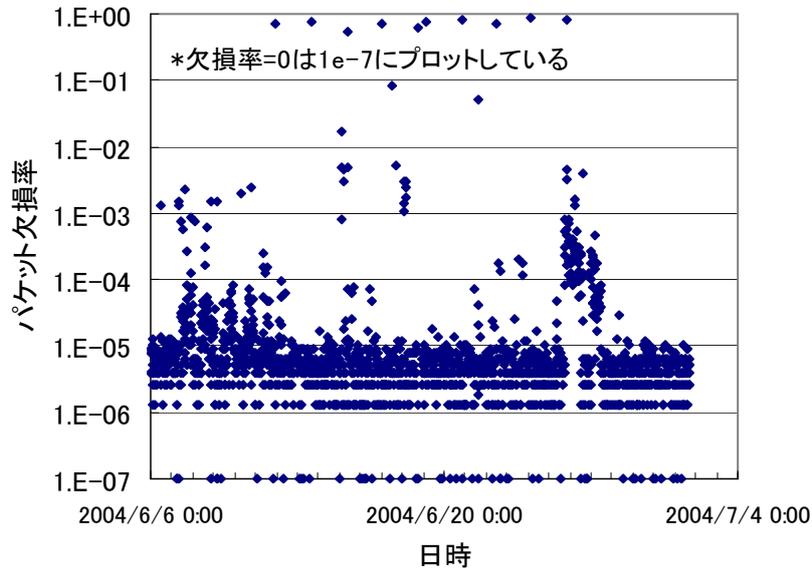


図 4-2-4-6 東京から大阪へデータ送信時のパケット欠損率の推移

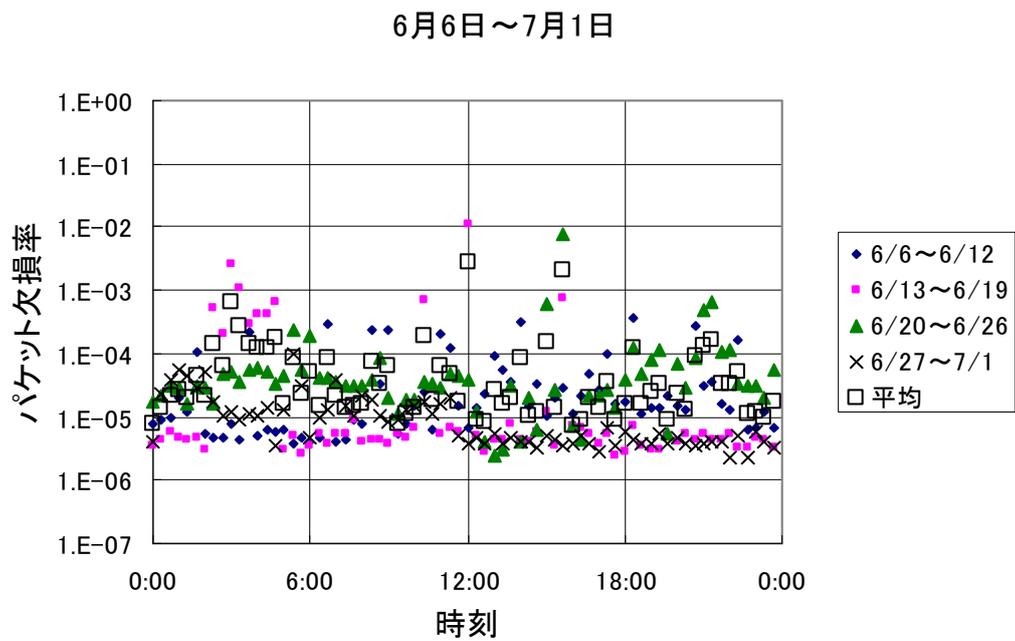


図 4-2-4-7 時間帯によるパケット欠損率の変動(A-1～A-4)

(1-3-2) 大阪から東京へパケットを送信した場合

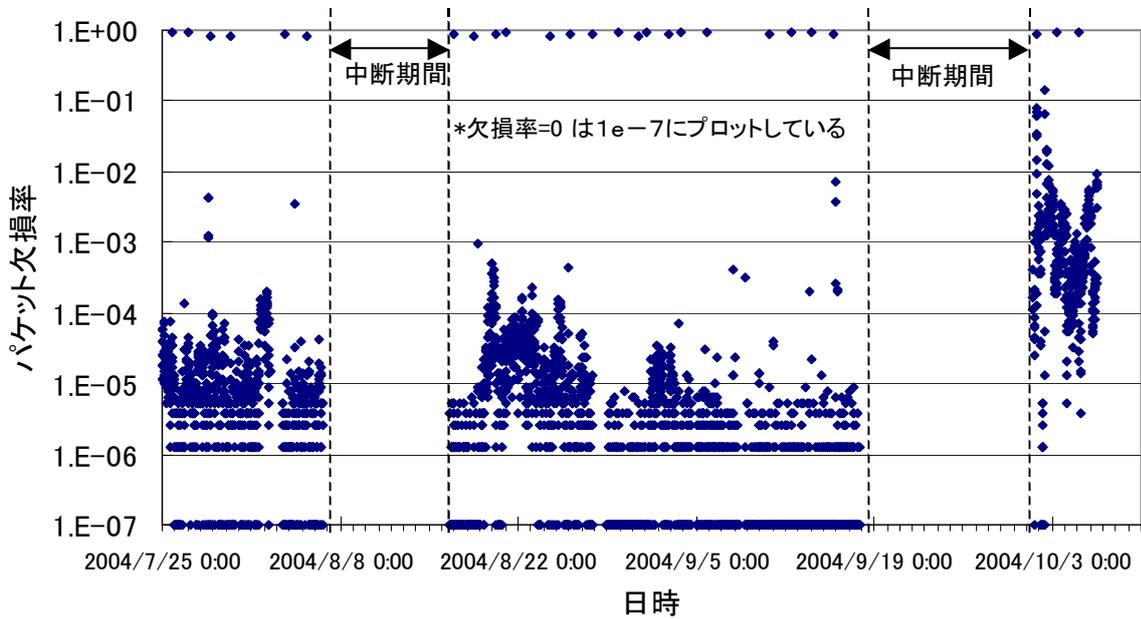


図 4-2-4-5 大阪から東京へデータ送信時のパケット欠損率の推移

7月25日～9月17日

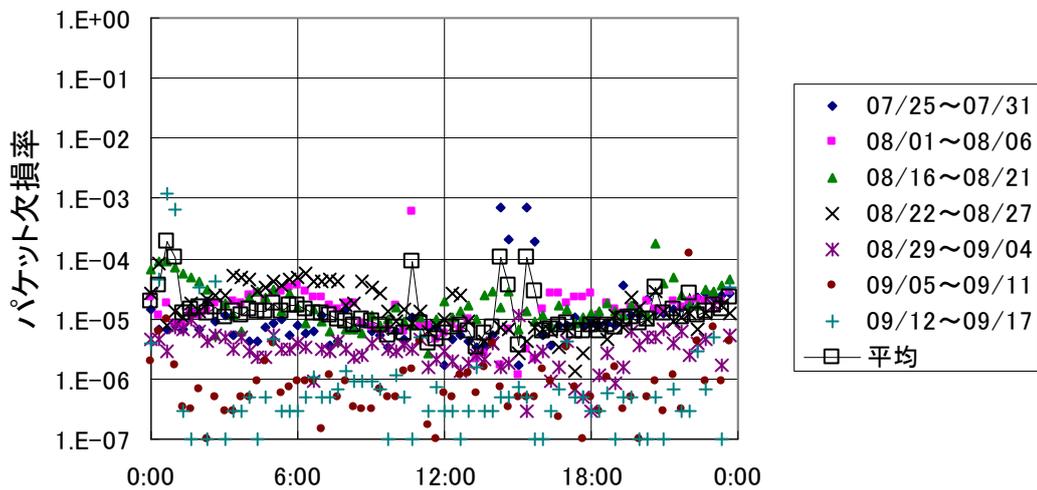


図 4-2-4-6 時間帯によるパケット欠損率の変動(B-1～B-7)

(2) バースト欠損特性

(2-1) 概要

測定において発生したバースト欠損の頻度をヒストグラムにして示す。横軸はバースト欠損長、縦軸は発生回数である。(ただしパケット欠損率が10%以上発生していた測定はグラフに含めていない)

(2-2) 測定結果

A. 伝送方向 東京 => 大阪

(A-1)

測定期間：2004年6月第2週(6/6～6/12)

最大バースト欠損長 387

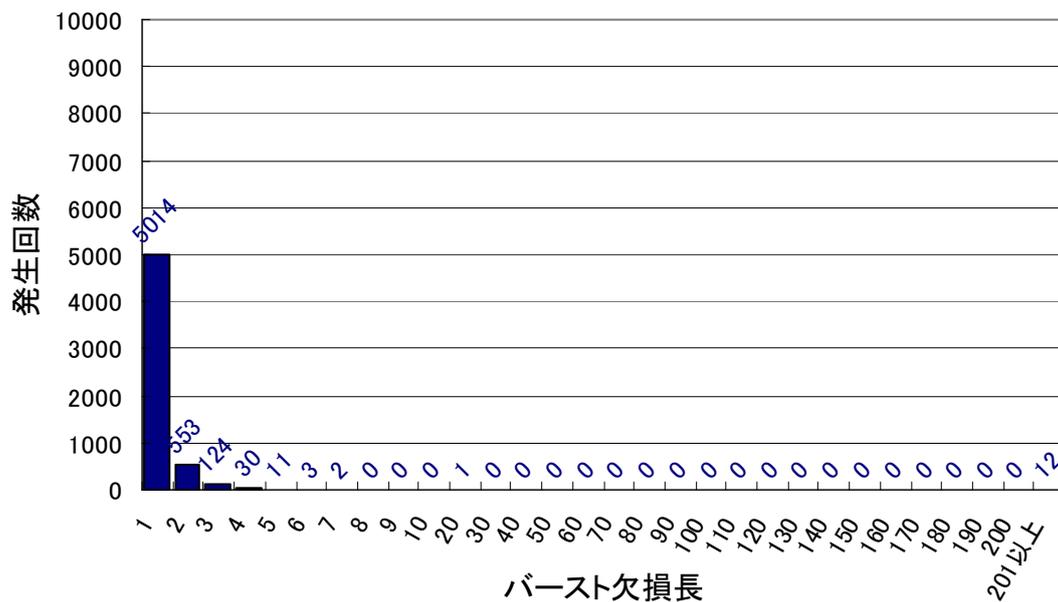


図 4-2-4-8 バースト欠損の発生回数(A-1)

B. 伝送方向 大阪 => 東京

(B-1) 測定期間：2004年7月第4週(7/25~7/31)

最大バースト欠損長 10

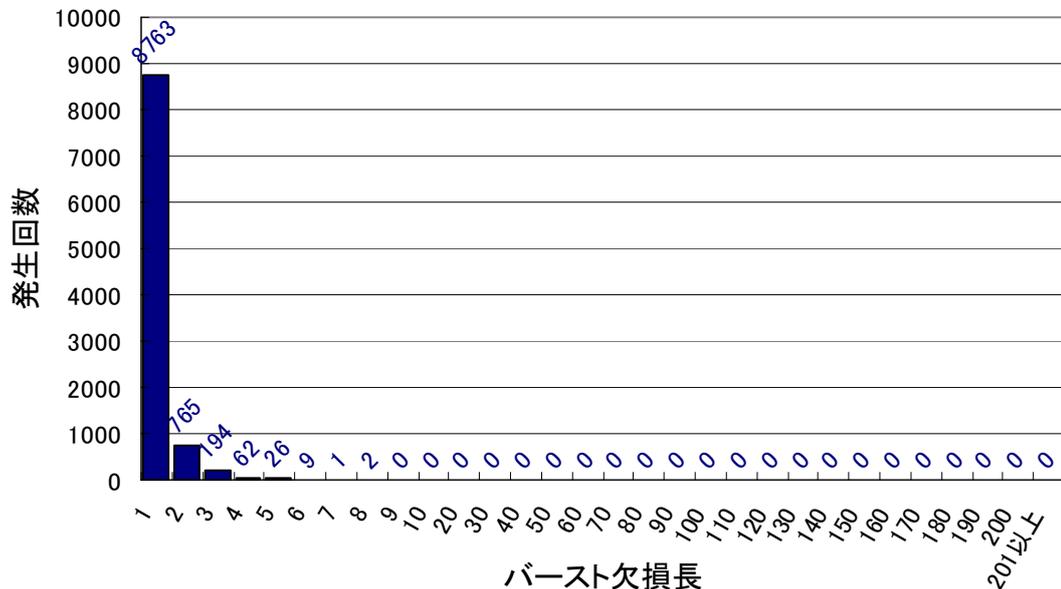


図 4-2-4-9 バースト欠損の発生回数(B-1)

4-2-4-3 パケット欠損のモデル検討

4-2-4-2 で得られたデータをギルバートモデル(図 4-2-4-10)への適用を行った。表 4-2-3-1 にギルバートモデルのパラメータを、図 4-2-4-11, 図 4-2-4-12 にバースト欠損の発生頻度の測定値及び、ギルバートモデルによる理論値を示す。本測定では1週間当たりのパケット欠損発生回数は $1.0E+03 \sim 1.0E+06$ 程度のオーダーでしかないので、ある程度の発生頻度の得られる10パケット未満のバースト欠損についてプロットして測定値と理論値の比較と考察を行った。

図 4-2-4-11 の測定値と図 4-2-4-12 のギルバートモデルの結果は、一致していないことがわかる。特に実測した結果では、バースト欠損が非常に高い確率で発生するが、理論値では測定値ほどのバースト欠損は発生していない。これは測定時にバースト欠損長が200パケットを越えるケースが何度も発生したケースや、パケット欠損の発生回数に対して非常に大きなバースト欠損が発生するなどしており、ギルバートモデルのランダム性から大きく外れる現象が発生したこと、及びパケット欠損率が低い(=パケット欠損のサンプル数が少ない)ことが原因と考えられる。一方、図 4-2-4-3-4 と図 4-2-4-3-5 から、200パケット以上のバースト欠損が発生している時もあるが、ギルバートモデルによる理論値と非常に良く一致している。

以上から、測定期間内に大きなバースト欠損が発生していない場合は理論値と測定値は短いバースト長ではよく似た傾向が得られた。また、大きなバースト欠損が発生している場合においても、平均欠損率が高くパケット欠損の発生

回数自体が多ければ測定値と理論値は非常に良く似た傾向を示した。しかし、(A-4)や(B-1)のように、大きなバースト欠損長は10 パケット未満であるが、理論値と大きくずれる測定結果も存在する。従って今後は、測定サンプル数を増やして大きなバースト欠損への適合性の確認、2次以上のギルバートモデルを適用してモデルの再検討などを行う必要があると考えられる。

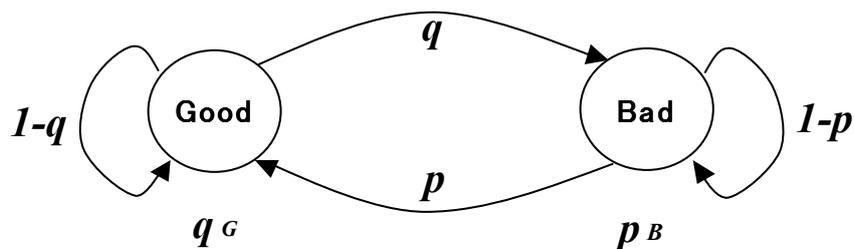


図 4-2-4-10 ギルバートモデル

表 4-2-4-3-1

測定期間	測定結果	平均バースト長	$p$	$p_B$ (平均欠損率)	$q_G$	$q$
6/6~6/12	A-1	4.78	0.21	4.95E-05	0.99995	1.04E-05
6/13~6/19	A-2	22.01	0.045	2.77E-04	0.99972	1.26E-05
6/20~6/26	A-3	14.09	0.071	1.71E-04	0.99983	1.22E-05
6/27~7/1	A-4	1.04	0.96	1.27E-05	0.99987	1.22E-05
7/25~7/31	B-1	1.12	0.89	3.29E-05	0.99997	2.93E-05
8/1~8/6	B-2	3.13	0.32	3.03E-05	0.99997	9.68E-06
8/16~8/21	B-3	1.09	0.92	2.62E-05	0.99997	2.40E-05
8/22~8/28	B-4	1.04	0.96	2.03E-05	0.99998	1.95E-05
8/29~9/4	B-5	1.21	0.82	3.28E-06	0.999997	2.70E-06
9/5~9/11	B-6	1.23	0.81	2.96E-06	0.999997	2.40E-06
9/12~9/17	B-7	2.80	0.36	2.96E-05	0.999970	1.06E-05
10/1~10/2	B-8	2.39	0.42	5.80E-03	0.994198	2.45E-03
10/3~10/9	B-9	1.20	0.84	1.16E-03	0.998838	9.72E-04

バースト欠損の発生頻度(実測値)

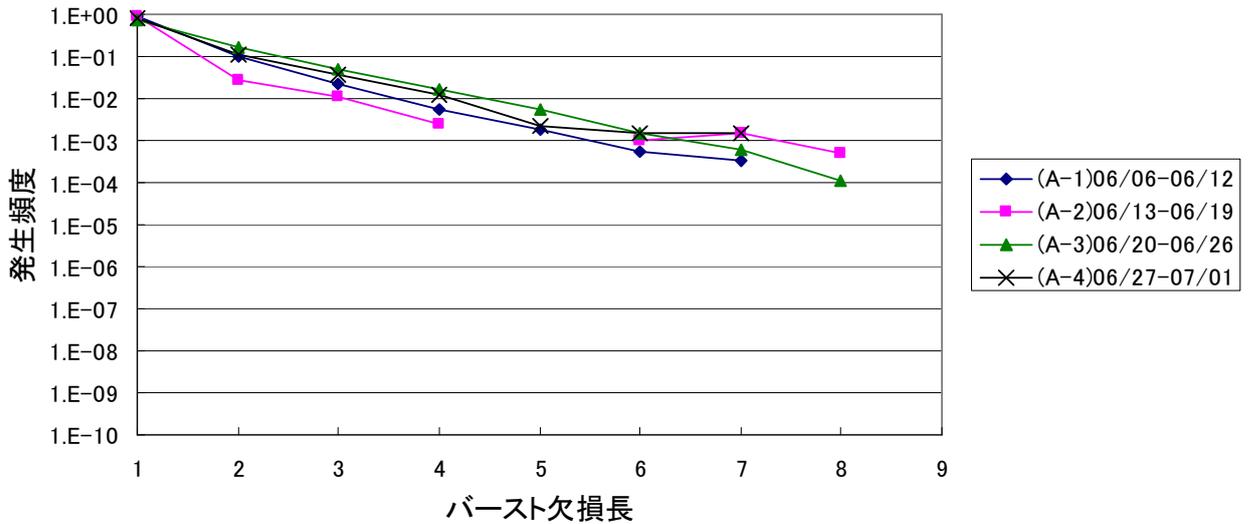


図 4-2-4-11 バースト欠損の発生頻度(測定値 A)

バースト欠損の発生頻度(ギルバートモデル)

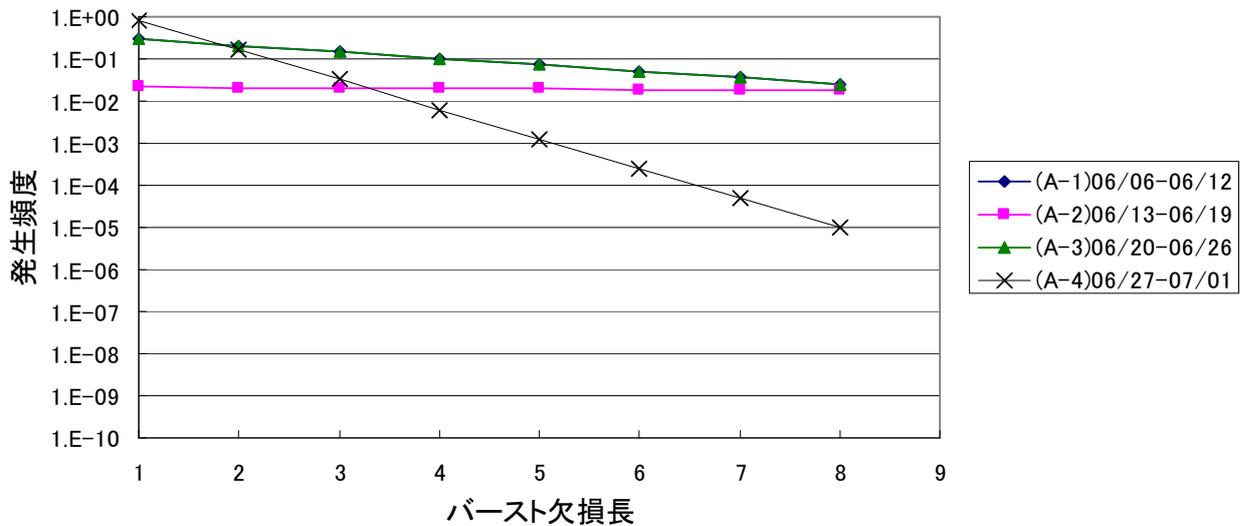


図 4-2-4-12 バースト欠損の発生頻度(理論値 A)

#### 4-2-5 結論

インターネットのジッタには、長い周期のジッタと短い周期のジッタが存在し、緩やかなジッタは測定時間 2,500 秒の間に、±10msec 変動していることが分かった。1%程度のパケット欠損が発生し、この影響によって映像の乱れが発生していると考えられる。また今回の結果をギルバートモデルに当てはめると、比較的よく一致していることがわかった。

## 4-3 デバイス化とホームゲートウェイ等評価プラットフォームの試作、実証試験および I E T F 等へのドラフト提案

### 4-3-1 序論

4-2 で述べたように、インターネット上でパケット欠損が発生していることがわかった。FEC (Raptor™) 機能を持たない VOD サーバにリアルタイムに符号化できる FEC サーバを配備し、復号化機能を有する STB を用いて、B2C 型サービスにおける FEC 技術の有用性・機能性の実証試験を行った。その結果、STB 向け VOD サービスを対象に商用ネットワーク上で FEC の有効性を確認した。

### 4-3-2 実証実験内容

NTT-BBのバックボーンネットワーク内に住友電工ネットワークス製のFECサーバとNEC製の配信サーバを設置し、NTT東日本の地域IP網経由で本サービス用にモニターにSTBを貸与し、FEC技術を適用したMPEG-2 TSデータを配信した。ネットワーク構成は、図4-3-2-1のとおりである。なお、配信サーバ～STB間は、HSAC1.0(\*1)に準拠している。

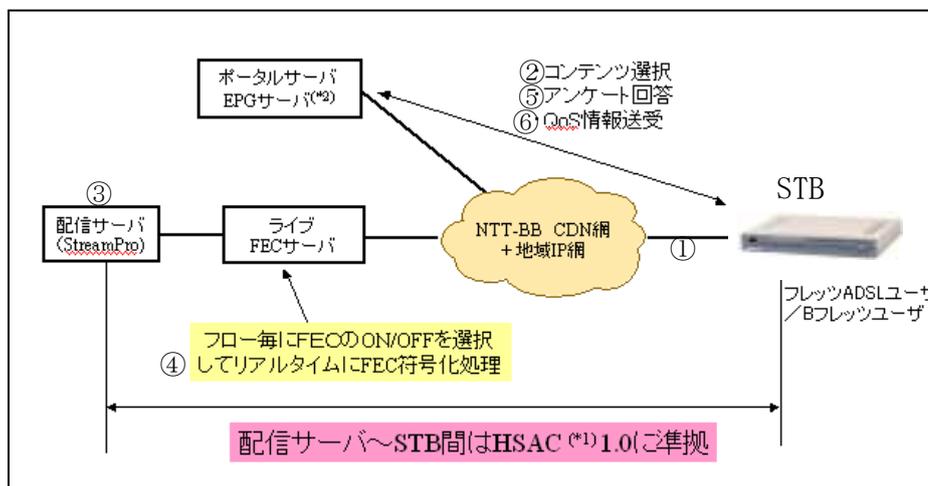


図 4-3-2-1 システム基本構成

(\*1) 光サービスアーキテクチャコンソーシアム

(\*2) Electronic Program Guide (電子番組表)

モニター期間は、平成 15 年 12 月 1 日～平成 16 年 1 月 31 日の 2 ヶ月間であった。ただし、12 月 1 日からモニター機器を順次発送したため、収集データは、12 月 8 日からのものを採用した。

モニターは、東京都 (一部エリアを除く) において、NTT 東日本の提供する「フレッツ・ADSL (モア II、モア、8M タイプ)」、「B フレッツ (ベーシック、ニューファミリー、ファミリー、マンション)」を利用し、実効速度が 4Mbps 以上の利用環境にあるユーザに限定して募集した。

モニターは、都内在住の合計 299 人を対象とする。また実験に使用したコンテンツは、「アニメ」、「映画」、「音楽」の大きく分けて 3 種類、約 100 本のコン

コンテンツを提供した。コンテンツの中には、FEC-ON・FEC-OFF が混在する。

実験には、定量的(客観)評価と定性的(主観)評価の2つの評価方法を用いた。

#### (1) 定量的(客観)評価

STBにQoS評価用エージェントを実装し、コンテンツ視聴毎にQoSデータおよびFEC機能の運用データを収集した。QoS評価用サーバに対してサーバ・クライアント間のQoS情報及びFEC機能の運用情報を通知する。

##### 【評価項目】

- ・パケット欠損
- ・FEC修復失敗エラー
- ・FEC適用ステータス

#### (2) 定性的(主観)評価

定性的評価では、視聴毎にSTBによる簡易なアンケートと、Webによる詳細なアンケートを採った。

### 4-3-3 実験結果

サーバへの同時アクセス数については、昼間より夜間の方が多傾向にある。また、総アクセス数と同様、週末にアクセスが多い。同時アクセス数のピーク数は、14クライアント(全体の約5%)である。年末年始のアクセス傾向については、週末と同程度のアクセスがあると想定していたが、意外に少なかった。

#### (2) パケット欠損率と視聴時間の関係を比較

モニターのパケット欠損率と視聴時間の関係をグラフにしてみると、FEC-ONの場合(図4-3-3-3)には、パケット欠損が10%以下であれば、多くのモニターが20分以上視聴しているのが分かる。一方、FEC-OFFの場合(図4-3-3-10)は、パケット欠損が1%以下でも、視聴に耐えられず、ほとんどのモニターが視聴を止めてしまうことが分かる。つまり、FECの効果によって、モニターの視聴時間が長くなる傾向にある。

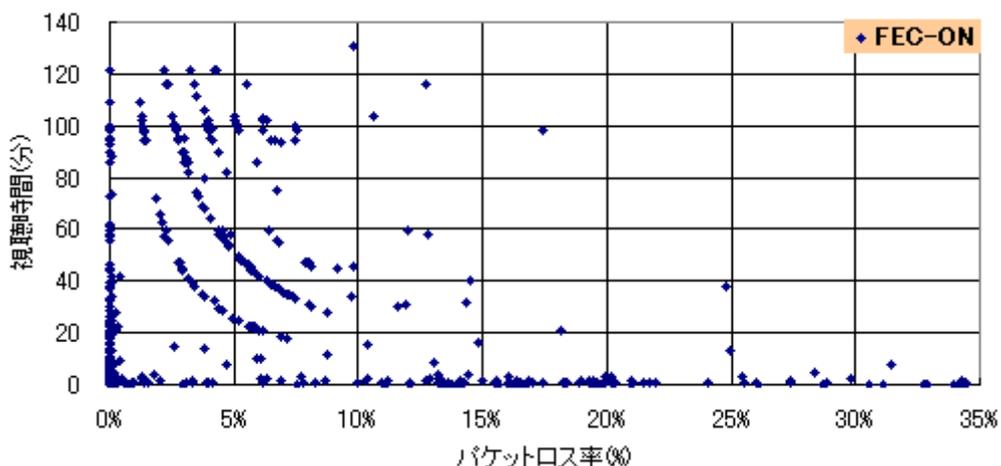


図 4-3-3-1 モニターのパケット欠損率と視聴時間の関係 (FEC-ON)

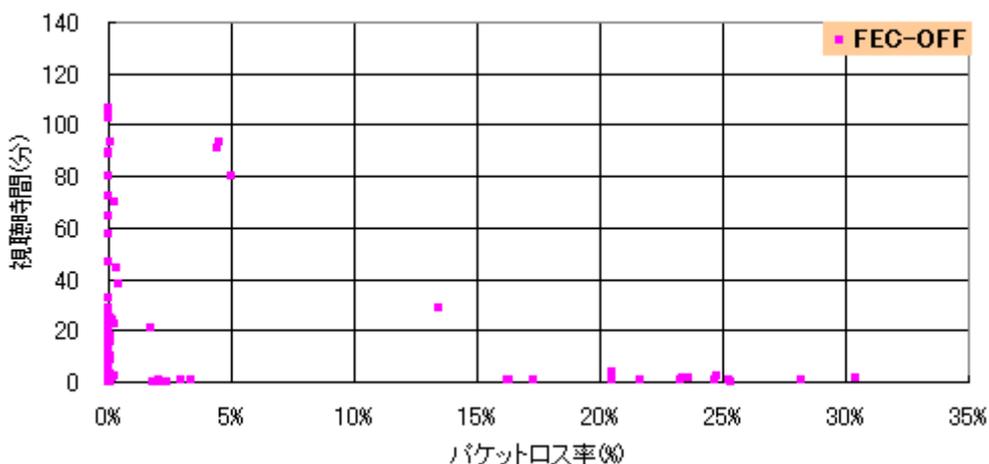


図 4-3-3-2 モニターのパケット欠損率と視聴時間の関係 (FEC-OFF)

(2) パケット欠損率と視聴回数の関係を比較

各モニターのパケット欠損率と(3分以上)の視聴回数(回)の関係を図 4-3-3-11 に示す。FEC-ON の場合には、パケット欠損が 20%以下でも 100 以上の視聴回数があるのに対して、FEC-OFF の場合は、パケット欠損が 5%を超えてしまう(グラフ上は 5%刻みで表記しているが実際は 1%以下)と、ほとんどアクセスがなくなってしまう。つまり、FEC の効果によって、モニターの視聴回数が増える傾向にある。

パケットロス率と視聴回数の相関関係

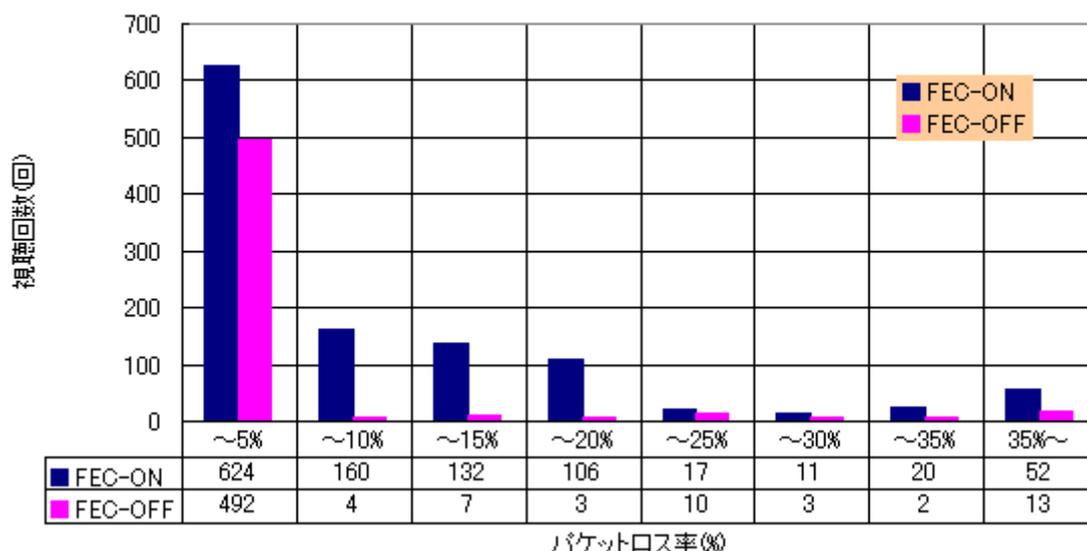


図 4-3-3-3 パケット欠損率と視聴回数の関係

(2) パケット欠損率と安定度評価の関係を比較

安定度が「よい」と答えたモニターのパケット欠損率の内訳を算出したところ、FEC-ON では、パケット欠損 5%までが 65%、5~10%が 33%であった。一方、FEC-OFF では、パケット欠損 5%まで(実際の測定値は 1%以下)が 99%であった。つまり、FEC-ON では、約 10%までのパケット欠損発生しても安定したサービスが提

供できるが、FEC-OFF では、パケット欠損が 1 %以下でも、サービス提供が困難となってしまう。

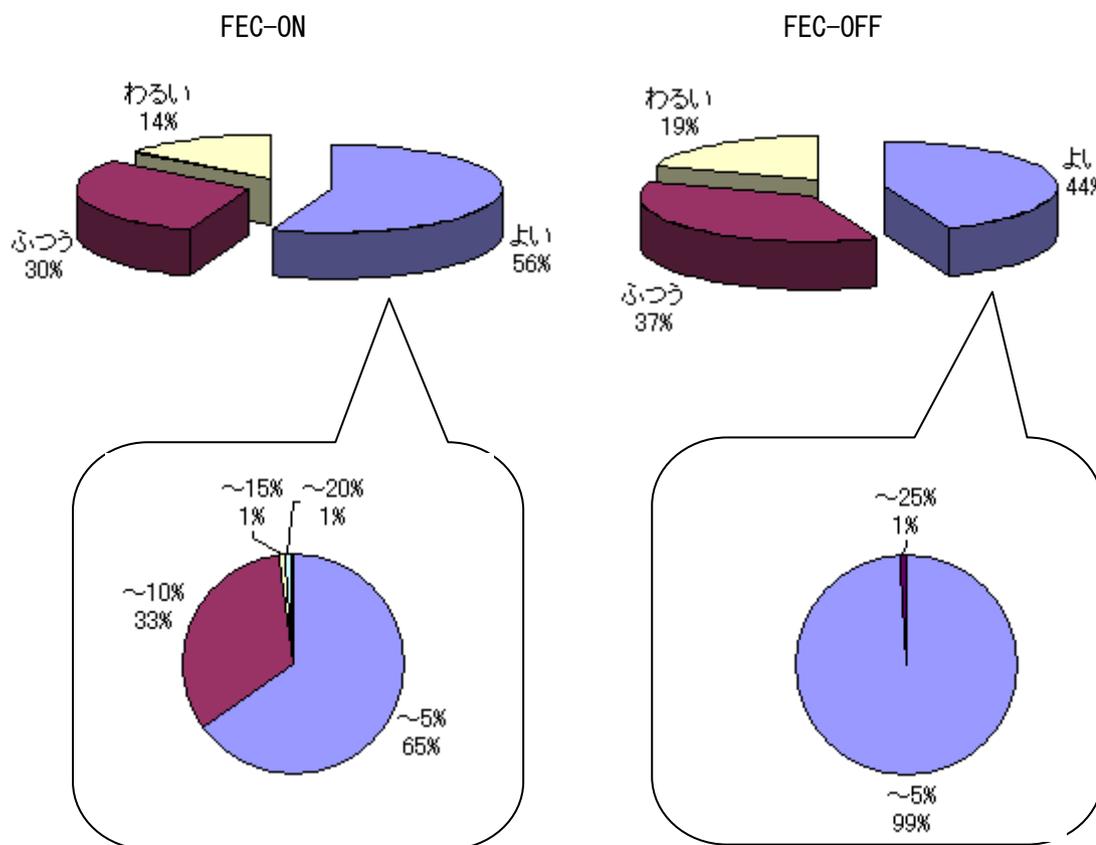


図 4-3-3-4 安定度が「よい」と答えたモニターのパケット欠損率の内訳

#### (4) 利用回線種別のパケット欠損率分布

モニターのパケット欠損率と視聴時間の関係を利用回線種別にグラフにしてみると、ADSL では、パケット欠損が 35%までに散在しているが、B フレッツでは、8%までに収まっている。

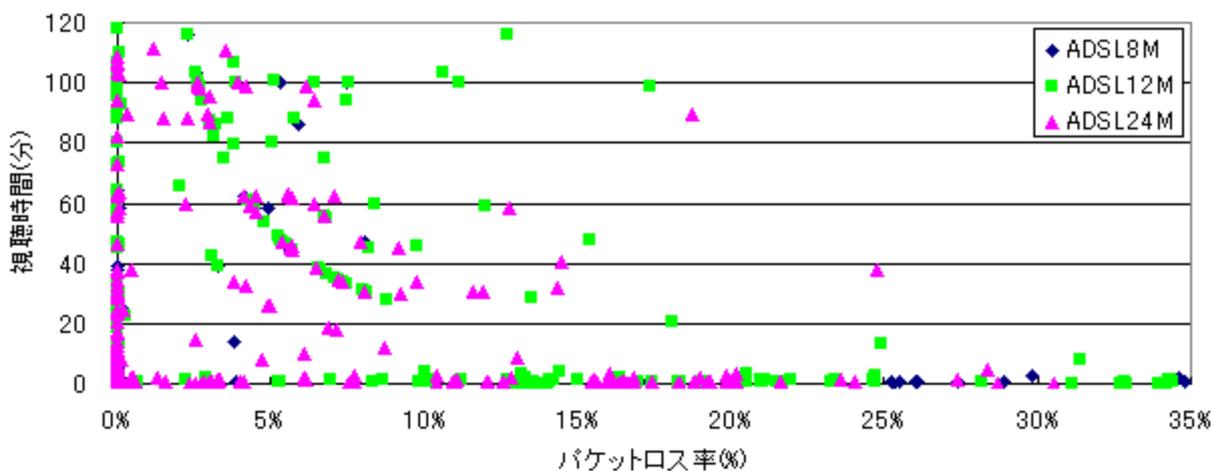


図 4-3-3-5 パケット欠損率分布(ADSL)

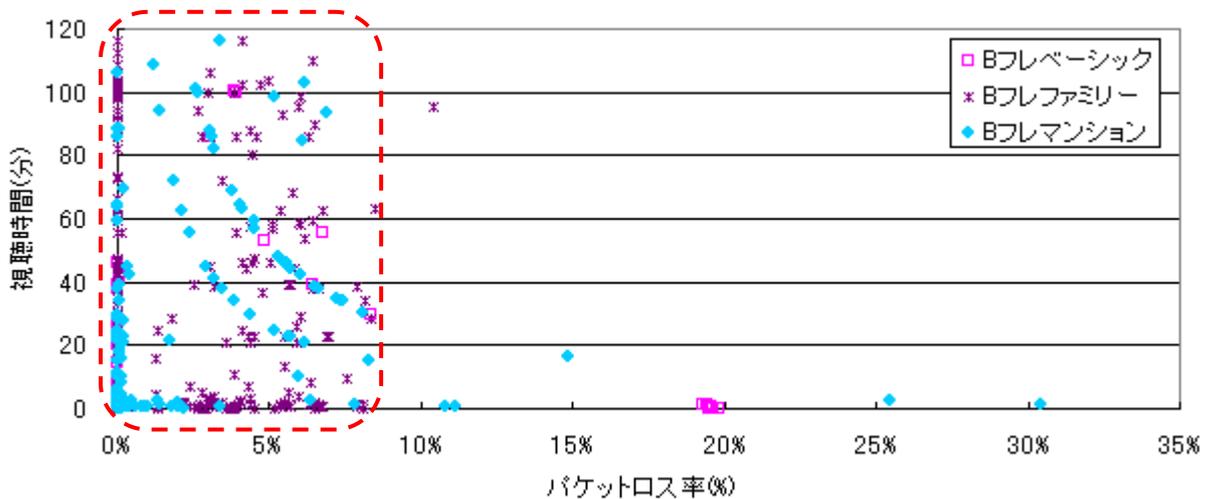


図 4-3-3-6 パケット欠損率分布(B フレツ)

#### 4-3-4 IETF への標準化活動

##### (1) FEC 機能付き RTSP プロトコル

映像伝送のための RTSP プロトコルに FEC 機能を付加したものを実装し、実証実験で検証を行い、その結果をもとにして IETF にて標準化の提案を行うという手順で標準化プロセスをすすめる。

提案するプロトコルは、RTSP を拡張し、ストリームサーバとクライアントの間で FEC パラメータの交換を可能にしている（以下、提案するプロトコルを「拡張 RTSP」と呼ぶ）。今回の実証実験では、FEC サーバを使用したか、提案するプロトコルでは FEC サーバなしの形態も考慮し、その場合 FEC サーバの機能はストリームサーバ自身に取り込まれることになる。

拡張 RTSP では、次のような方法で FEC パラメータの交換を実現している。

(a) クライアント (STB) は SETUP メッセージ中に FEC パラメータの記述を追加する。

パラメータ記述には、

```
Fec-method: method=<fec method>
```

```
Fec-param: <fec parameters>
```

という拡張ヘッダを使用し、ここに FEC パラメータの詳細を記述する。

(b) FEC サーバ (FEC サーバを使用しない形態では、ストリームサーバ) は、SETUP メッセージ中の FEC パラメータを解釈し、その応答をクライアントに返す。

(c) クライアントは、応答メッセージ中の FEC パラメータに対する応答を検出し、FEC を使用したストリーム伝送を行う。

実際のシーケンスを、図 4-3-4-1 および図 4-3-4-2 に示す。

なお、FEC サーバを使用した場合、拡張 RTSP はクライアントと FEC サーバ間

で適用され、ストリームサーバから見れば通常の RTSP と変わらず上位互換性を確保するようになっている。

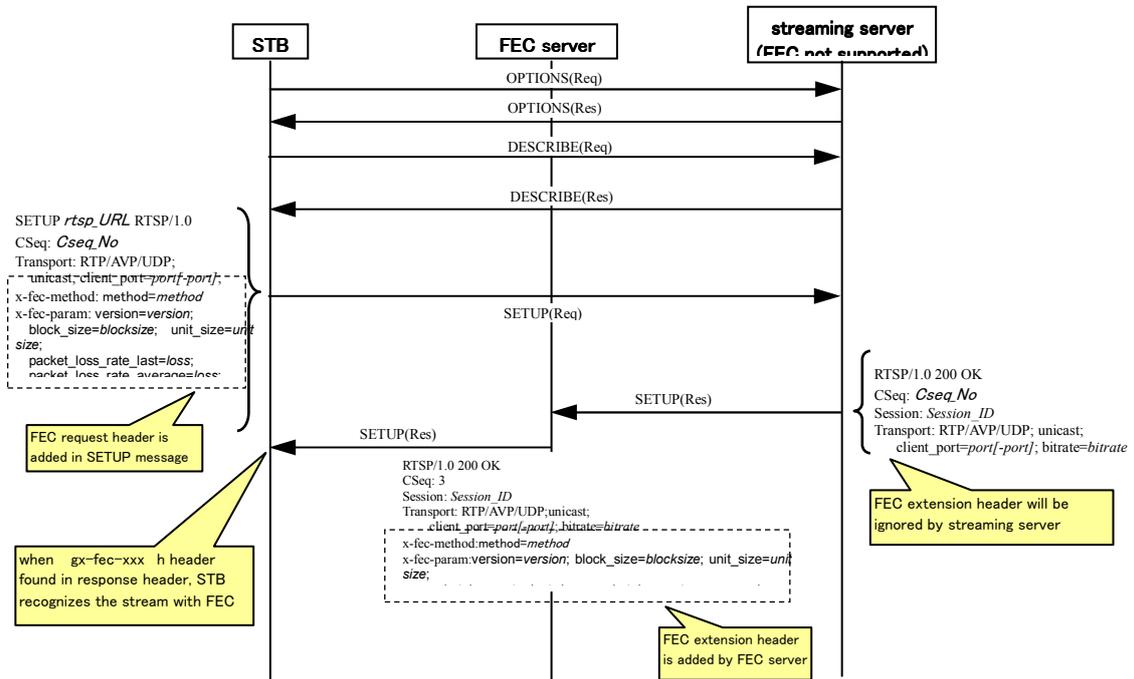


図4-3-4-1 拡張RTSPプロトコル (FECサーバあり)

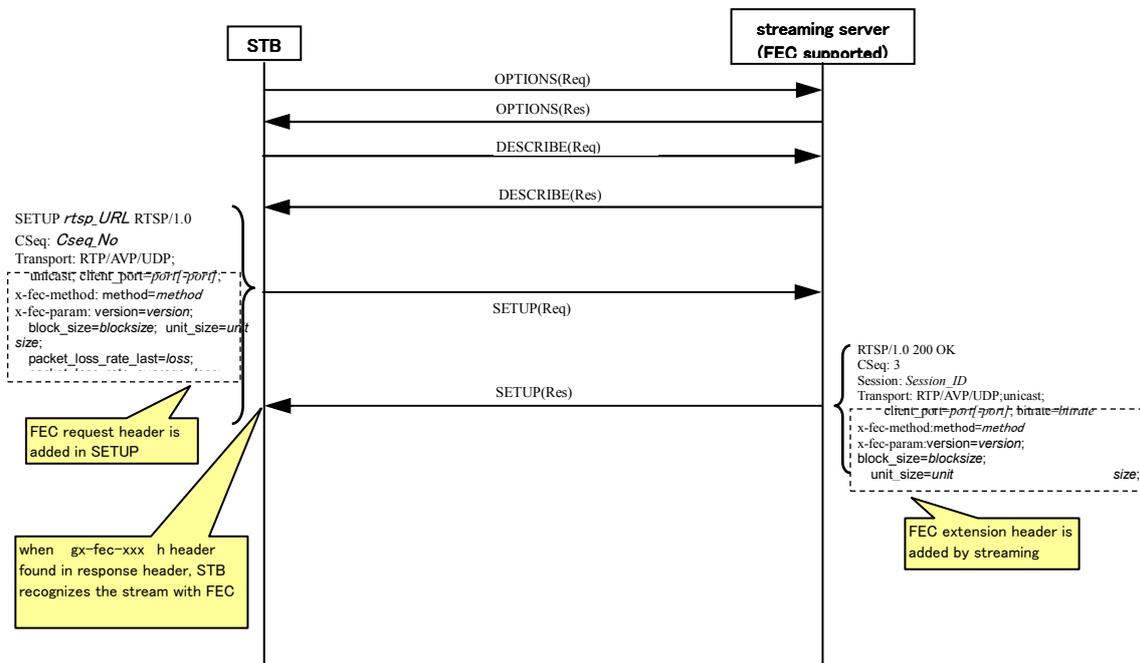


図4-3-4-2 拡張RTSPプロトコル (FECサーバなし)

## (2) 拡張 RTSP の IETF ドラフト案

上記拡張 RTSP を元に関係者でレビューし、IETF に提案するための作業を進めている段階である。

#### 4-3-5 結論

##### (1) FEC の有用性について

本実験では、FEC の有用性について、次のことが実証できた。

###### ① FEC の効果によって、モニターの視聴時間が長くなる

FEC-ON の場合には、パケット欠損が 10%以下であれば、多くのモニターが 20 分以上視聴していた。一方、FEC-OFF の場合は、パケット欠損が 1%以上発生すると、視聴に耐えられず、ほとんどのモニターが視聴を止めてしまうことがわかった。

###### ② FEC の効果によって、モニターの視聴回数が増える

FEC-ON の場合には、パケット欠損が 20%以下でも 100 以上の視聴回数があるのに対して、FEC-OFF の場合は、パケット欠損が 1%以下でもほとんどアクセスがなくなってしまうことがわかった。

###### ③ FEC なしでは、事業が成り立たない

FEC-ON では、約 10%までパケット欠損が発生しても安定したサービスが提供できるが、FEC-OFF では、パケット欠損が 1%以下でも、サービスが実質不可能となってしまうことがわかった。

##### (2) ネットワーク構成と FEC 適用領域について

各種利用回線のモニターを抽出し、ネットワーク構成別に主観評価を行ったところ、6M を視聴できたのは B フレッツ回線のモニターが中心で、ADSL 回線では 3M 止まりであり、回線速度の違いが顕著にあらわれた。また、FEC が有効な場合、パケット欠損が 5%発生した状況では、65%のモニターが安定度がよいと答え、パケット欠損が 10%発生した状況では、33%のモニターが安定度がよいと答えた。つまり、パケット欠損が 10%程度発生した状況でも FEC が視聴品質確保の一助となっていることがわかった。

##### (3) FEC 有無での操作性について

FEC-ON の操作性・応答性に関しては、30%近くのモニターが「よい」と答え、「ふつう」も含めると約 80%を占めている。また、FEC を適用することによって、トリックプレイ(早送りや巻き戻し)の操作性が向上した。

##### (4) アクセス傾向について

アクセス数は、総アクセス数、同時アクセス数ともに週末に多い傾向にある。同時アクセス数のピーク数は、14 クライアント(全体の約 5%)であった。実際のサービスを提供するにあたって、これらを考慮したネットワーク・サーバ構成が必要になる。

##### (5) コンテンツについて

コンテンツの内容に関しては、半数以上のモニターが満足している。画質・音質については、不満と答えたモニターは 10~20%程度と大変少なかったため、問題はなかったと考えられる。回線環境によるバラつきも見られなかった。画質・音質の乱れについては、無料なら多少我慢するが、有料だったら絶対

に我慢できないという回答が見受けられた。有料サービスにした場合には、品質保証が必須である。

(6) STB サービス全体について

サービス全体の感想としては「いくつか気になる点はあるが、概ね良い」が圧倒的に多く、この視聴形態自体は受け入れられることが分かった。

今後の課題としては、本実験では、FEC 適用率は全てのモニターに対して一律であったが、FEC 適用率を ADSL と B フレッツで変えたり、ユーザのネットワーク状況に合わせるといったことが考えられる。

## 4-4 帯域抑制手段に関する研究、中継装置用超高速低遅延順方向誤り訂正デバイスの試作・実験

### 4-4-1 序論

デファクト標準化を念頭においた、順方向誤り訂正機能およびその選択制御機能のデバイス化と評価プラットフォームとしてのサーバ・クライアントおよびホームゲートウェイ等低速低価格中継装置の試作を行なう。

### 4-4-2 研究内容

インターネット上でのコンテンツ配信に用いられているプロトコルは、TCP/IP が一般的で、伝送途中で発生したパケット欠損を、ARQ (Automatic Repeat reQuest) という再送方式で補っている。UDP を用いたリアルタイムの映像伝送を行う場合は、伝送路途中で発生するパケット欠損に対しては、前方(順方向)誤り訂正技術 (Forward Error Correction : FEC) を用いて、受信側(前方)で欠損パケットを補う方法が用いられる。この理由は、再送による遅延時間の予測が難しいためである。FEC を用いると、想定された数以下のパケット欠損であれば、受信側で元のパケットを復元することが出来る。しかし、元のパケットを復元するためには、一定時間に余分なパケットを送信する必要があり、伝送速度の増加によってルータの処理帯域を圧迫してしまい(輻輳が発生し)、パケット欠損が増加する。ルータでパケット欠損が発生すると、TCP 系のパケットは、受信確認が出来ず、結果的に伝送速度が低下する(輻輳制御)。一方 UDP 系のパケットには、帯域を抑制する機能がついていないため、結果的に TCP 系の通信を押しよけて通すこととなる。インターネット上の均衡を保つためには、UDP パケットに対しても何らかの抑制メカニズムを考えておく必要がある。この際、課金モデルやインターネット上の仕掛けに対する考察も、必要となる。

### 4-4-3 IETF の動向

IETF (Internet Engineering Task Force)では、継続時間の長い UDP 系の通信に輻輳制御機能を付加しようという動きがある。

#### (1) DCCP (Datagram Congestion Control Protocol)

DCCP はまだ RFC にはなっておらず、作業中であるが、IETF のホームページにいくつかのドキュメントが提出されている。基本的には、輻輳制御機能をもったトランスポート層プロトコルとして DCCP を作り、UDP に替わる方式を普及させていこうとしている。再送制御はしないものの、受信側から送信側に受信確認を返すことで、送信側で輻輳状況を判断し、送信を抑制するメカニズムである。具体的に、2つのタイプが提案されている。

#### (A) TCP-like な方式

TCP と同様に Windowing (受信確認を受け取る前に送信可能なデータ量を制限すること) を行い、このサイズをパケット欠損によって制限していくやり方。短時間での送信レートがパケット欠損によって急激に変動する性質があり、ストリーミング映像配信には向かない。

#### (B) TCP-Friendly な方法

パケット欠損率を測定し、その欠損率における TCP の性能を推定して、それと同程度に平均送信レートを調整する方法。送信レートの変動が平滑化される

ので、ストリーミング映像配信に適している。この方式は (TCP Friendly Rate Control : TFRC) は、RFC3448 に詳しい記述があり、アルゴリズムが示されている。この中に、パケット欠損率から TCP 性能 (スループット) を算定する式が示されており (出典: "Modeling TCP Throughput: A Simple Model and its Empirical Validation", Proc. ACM SIGCOMM 1998)、パケットの往復にかかる時間 (往復遅延) (Round Trip Time : RTT) とパケット欠損率によって TCP 系に適用されるの伝送速度が算出される。パケット欠損率と TCP Friendly な伝送速度の関係を、図 4-4-1 に示す。この結果から、100ms 程度の往復遅延がある場合には、パケット欠損率 0.1%以上では、映像伝送速度 4Mbps で伝送することが出来ない。

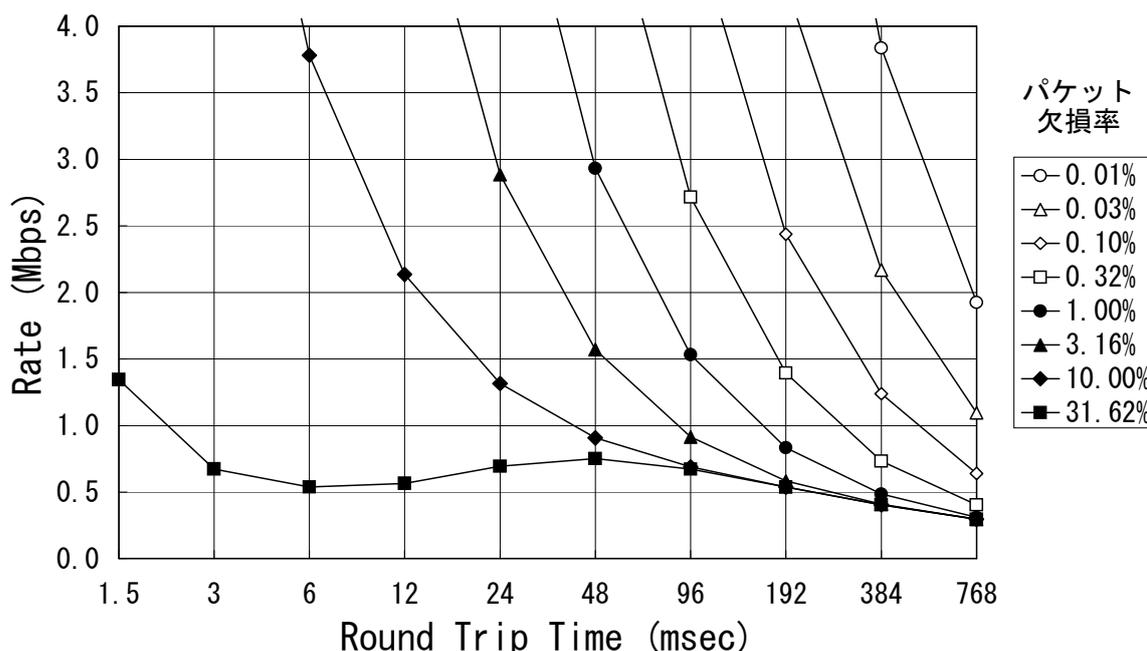


図 4-4-1 パケット欠損率と TCP Friendly な伝送速度の関係

またパケット欠損率：1%，RTT：100ms の状況では、伝送速度の 1.5Mbps と算出される。この状態で、映像伝送速度 4Mbps を確保しようとするれば、基準となる伝送速度より 2.5 倍程度の伝送帯域を過剰に確保していることになる。

TFRC に基づく計算では、インターネットでの映像配信の速度を増加させようとする、インターネット上でのパケット欠損率を低減する必要がある。つまり「映像配信の速度：増加」→「パケット欠損率：低減」→「FEC による欠損パケットの復元」→「映像配信速度：増加」という正帰還がかかり伝送速度が発散してしまうため、FEC を用いる事が出来なくなる。しかし、ベストエフォート型の定額サービスでは、何らかの抑制メカニズムが必要であり、その基準の考え方として、この方法は意味があると考えられる。

ベストエフォート型の定額サービスで上記の方法を用いる場合には、ユーザの契約内容に応じて、基準となる伝送速度の何倍までの帯域を許すかという設定を事前に行う方法が考えられる。この許可された帯域内に映像伝送速度が収まるように、サーバ(FEC エンコーダ)側で映像伝送速度を動的に調整する。映像

伝送速度が、許容された帯域より大きい場合には、映像自体の伝送速度を低下させる方法も、考慮する必要がある。

#### 4-4-4 ライブ FEC サーバの開発

##### 4-4-4-1 ライブ FEC サーバの仕様

性能評価試験で使用したライブ FEC サーバの機器仕様について表 4-4-4-1～表 4-4-4-4 に示す。また図 4-4-4-1 に概観図を示す。

表 4-4-4-1 インタフェース仕様

項目	仕様
<b>Main ポート</b>	
ポート数	RJ45・・・2ポート
仕様	10/100/1000Base-T、Auto-MDI-X、Auto-Negotiation
物理インタフェース	RJ-45
<b>Ext ポート</b>	
ポート数	RJ45・・・2ポート
仕様	10/100/1000Base-T、Auto-MDI-X、Auto-Negotiation
物理インタフェース	RJ-45
<b>保守ポート (イーサポート)</b>	
ポート数	RJ45・・・1ポート
仕様	10/100Base-T、Auto-MDI-X、Auto-Negotiation
物理インタフェース	RJ-45
<b>保守ポート (シリアルインタフェース)</b>	
ポート数	2ポート
規格等	RS232C 信号
伝送速度	9600bps
物理インタフェース	D-sub 9ピンコネクタ
用途	ポート1：メインCPUに直結 ポート2：FEC用CPUに接続 (ロータリ SW2 で切替)

表 4-4-4-2 一般仕様

項目	仕様
外形寸法	44mm(H) x 430mm(W) x 420mm(D) (突起物含まず) ラックマウント型 本機器はEIA規格19インチラックに搭載可能です。
重量	5.5Kg
電源仕様	AC100V ±10V (50Hz 及び60Hz)
消費電力	12.5W以下
動作周囲温度・湿度条件	温度(°C) 0 ~ 40°C 湿度(%) 5 ~ 85%
放射電界強度	VCCIクラスA 適合
雑音端子電圧	VCCIクラスA 適合
電源電圧変動	AC100V ±10V (50Hz 及び60Hz)

表 4-4-4-3 性能仕様

項目	仕様
帯域幅	300Mbps
カスケード機能	最大3段

表 4-4-4-4 SW/LED 仕様

SW	
Reset	リセットボタン
ロータリ SW 1	16bit 0: 通常運用モード 1: 検査モード C: 運用構成定義強制消去 F: ロータモード (FTP によるファームウェアダウンロード)
ロータリ SW 2	16bit (FEC 用シリアルコネクタ切替) 0~9: 10 個の FEC 用 CPU の SIO ポートを切替
表示 LED	
PWR (緑色)	<ul style="list-style-type: none"> <li>・ 緑点灯・・・通電中</li> <li>・ 消灯・・・非通電中</li> </ul>
ALM (赤色)	<ul style="list-style-type: none"> <li>・ 赤点灯・・・セルフテスト異常検出 (致命的障害による運用停止)</li> <li>・ 赤点滅・・・セルフテスト実行中 (縮退動作中)</li> <li>・ 消灯・・・正常状態</li> </ul>
DIAG (赤色)	<ul style="list-style-type: none"> <li>・ 赤点灯・・・セルフテスト異常検出 (デバッグ用)</li> <li>・ 赤点滅・・・セルフテスト実行中 (デバッグ用)</li> <li>・ 消灯・・・正常状態</li> </ul>
OPR (緑色)	メイン CPU のステータスを確認する LED。 <ul style="list-style-type: none"> <li>・ 緑点滅・・・メイン CPU 動作状態</li> <li>・ 緑点灯/消灯・・・CPU ハングアップ状態</li> </ul>
メインポート/EXT ポート LINKLED (緑色)	<ul style="list-style-type: none"> <li>・ 緑点灯・・・Ether リンクアップ</li> <li>・ 消灯・・・Ether リンクダウン</li> </ul>
メインポート/EXT ポート ACTLED (緑色)	<ul style="list-style-type: none"> <li>・ 緑点滅・・・データ送受信中</li> <li>・ 消灯・・・非データ通信中</li> </ul>
保守ポート LINK (緑色)	緑点灯: リンクアップ 消灯: リンクダウン
保守ポート ACT (緑色)	緑点灯/点滅: データ通信 消灯: 非データ通信
FEC 用 CPU-LED × 20 個 (2 色)	Active-LED) <ul style="list-style-type: none"> <li>・ 緑点滅・・・FEC 動作中</li> <li>・ 緑点燈・・・FEC 動作停止中</li> <li>・ 消灯・・・CPU 非通電時</li> </ul>
	ALARM-LED) <ul style="list-style-type: none"> <li>・ 赤点滅・・・エラー発生。</li> <li>・ 消灯・・・CPU 通常動作。</li> </ul>



図 4-4-4-1 FEC サーバ概観図

#### 4-4-4-2 ライブ FEC サーバの構成

##### 4-4-4-2-1 概要

ライブ FEC サーバは RTSP (Real Time Streaming Protocol) で制御するストリーミングサービスにおいて、FEC のエンコード・デコードと帯域制御を実現する装置である。FEC (Forward Error Correction) 技術として、高速・低遅延の packets 欠損回復技術である「Raptor」を使用する。また RTSP により、ストリーミング開始・終了を制御する。

##### 4-4-4-2-2 全体構成

図 4-4-2 にライブ FEC サーバのブロック図を示す。

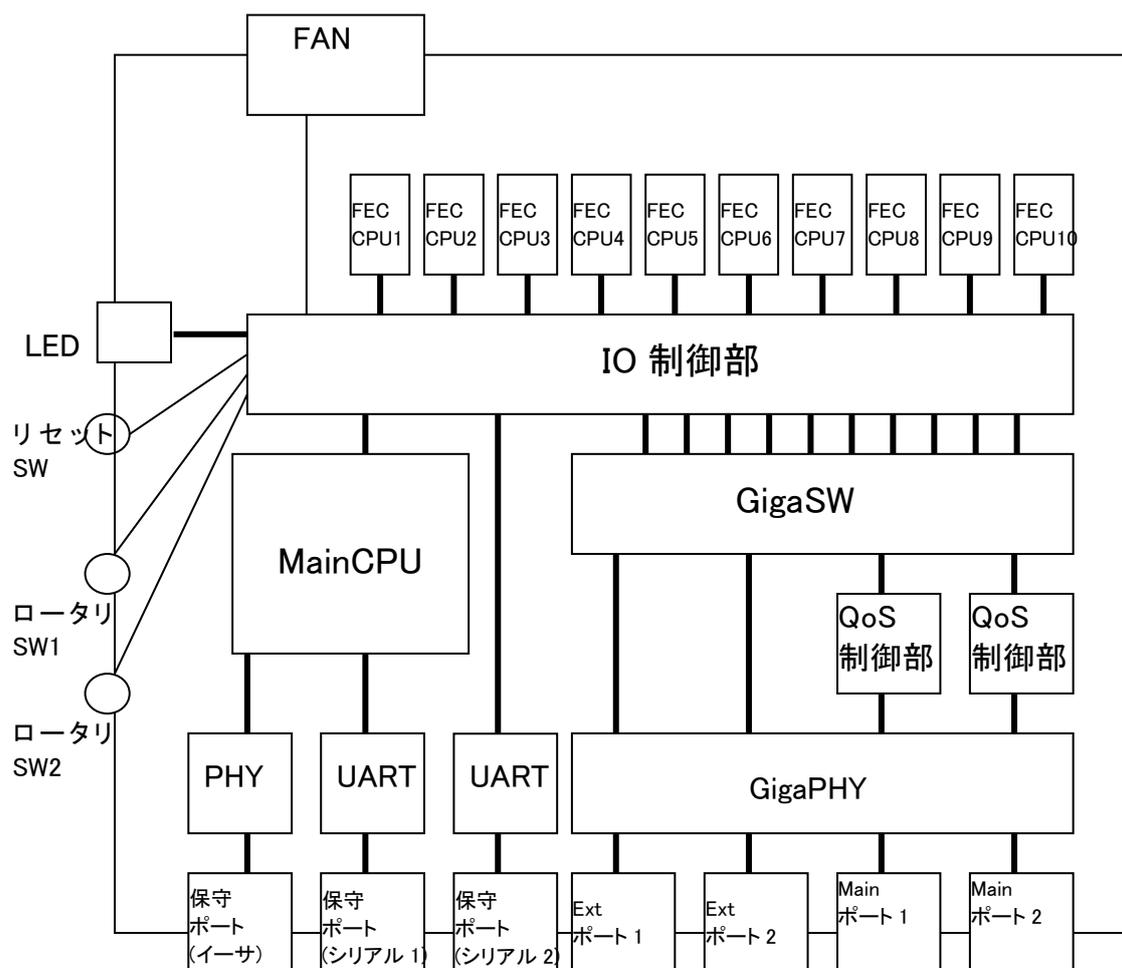


図 4-4-2 ライブ FEC サーバブロック図

Main ポートには 2 つのポートがあり (ポート 1、ポート 2)、構成定義により送信ポート、受信ポートに設定される。ストリーミングサーバから配信されたストリームを Main ポートで受信し、FEC 用 CPU にて FEC デコード・FEC エンコードを行う。また、FEC サーバからの送信ポートにシェーピング機能を持った QoS 制御部を有し、高品質な通信を可能とする。

1 台のライブ FEC サーバは 10 個の FEC 用 CPU を所持しており、複数のストリームを特定の CPU に負荷が生じないように分散してエンコード・デコード処理を行う。1 台のライブ FEC サーバで最大 100 のストリームを同時に処理ができる。

さらに最大 3 段までライブ FEC サーバを Ext ポートからカスケード接続することができ、その場合 30 個の FEC 用 CPU で処理し、最大 300 のビデオストリームを同時に処理することができる。

保守ポートとしてシリアルインタフェースが 2 ポート、イーサポートが 1 ポートある。シリアルインタフェースはメイン CPU に直結、FEC 用 CPU にはロータリースイッチを切替えることで 10 個の FEC 用 CPU のうちひとつを選択して接続する。シリアルインタフェースは機器のデバッグで使用する。イーサポートではファームウェアのダウンロード、FEC 処理の条件を構成する種々のパラメータの設定を行う。

メイン CPU では RTSP からストリーミング情報を取得し、各 FEC 用 CPU へのストリームの振り分けを制御する。全ての FEC 用 CPU のパラメータを管理しており、必要に応じて各 FEC 用 CPU にパラメータをダウンロードする。GigaSW の各種カウンタにより各ポートの統計情報を定期的に収集し保持する。保守用ポート、Main ポート、Ext ポートを監視しポート情報を保持する。その他、QoS 制御部、各 LED の制御、FAN 状態の監視を行う。

I/O 制御部では FEC 用 CPU、メイン CPU と GigaSW のインタフェースを接続するための SMII と MII の変換ロジック、メイン CPU と LED など各デバイスとの中継機能を有する。

QoS 制御部ではシェーピング機能があり、受信したストリームの転送レートにあわせてパケット送信間隔を設定する。QoS をハード化することにより各パケット送信後のウェイト処理を 20ns 単位での設定が可能となり、ストリームの転送レートに応じてきめ細かい設定ができる。パケット送信間隔はストリーム毎に設定できる。シェーピング方式については次項で述べる。

#### 4-4-4-3 ライブ FEC サーバ性能評価試験

##### (1) 概要

今回開発したライブ FEC サーバを FEC エンコーダ、FEC デコーダとして使用した際の性能評価を行った。映像配信サーバに対して多数のクライアントが配信を要求する状況を模擬した評価環境を構築し、配信コンテンツの種類や FEC サーバの FEC パラメータを様々に変化させた際の接続限界本数を調べる試験を行い、またライブ FEC サーバ内に組み込んだハードシェーパの機能評価試験も行った。

##### (2) 評価試験構成

評価試験構成を図 5-4-3-3-1 に示した。最大 100 本のストリームを配信するために配信サーバを 2 台用意し、また FEC オンのストリームと FEC オフのストリームを同時に流す試験のために負荷シミュレータを 2 台用意した。GigaSwitch としては Cisco 製 Catalyst3550 を、パケットキャプチャやパケット間隔測定にはアンリツ製 Data Quality Analyzer MD1230A を使用した。

##### (3) ライブ FEC サーバ基本設定

性能評価を行うに当たって、以下の設定を基本設定とした。

<配信コンテンツに関するパラメータ>

- ① コンテンツレート：6 Mbps
- ② コンテンツ長：10 分

<FECに関するパラメータ>

- ③ Raptor プロトコルバージョン：7
- ④ Encoding Unit Size：1024Byte
- ⑤ Source Block Size：32KB
- ⑥ Packet Loss Rate：2%
- ⑦ Target Loss Rate：1.00e-09

<システム構成に関するパラメータ>

- ⑧ 使用 FEC CPU 数：10 (MAX)
- ⑨ FEC オフのストリーム：なし

(4) 性能評価試験項目

(3)の基本設定のもとで、以下の各項目に関してパラメータを変化させた際にエンコード処理、デコード処理の接続限界本数がどのように変化するかを試験した。

- ① コンテンツレート：3 Mbps、4 Mbps、5 Mbps、6 Mbps
- ② コンテンツ長：1 分、10 分
- ③ Raptor プロトコルバージョン：7、9
- ④ Encoding Unit Size：256Byte、512Byte、1024Byte
- ⑤ Source Block Size：32KB、64KB、128KB
- ⑥ Packet Loss Rate：1%、2%、5%
- ⑦ Target Loss Rate：1.00e-08、1.00e-09、1.00e-10
- ⑧ 使用 FEC CPU 数：1、5、10
- ⑨ FEC オフのストリーム数：0、10、20、30、40、50  
(⑨はエンコーダ性能評価試験のみ)

(5) 性能評価方法

<エンコード性能評価試験・デコード性能評価試験>

(4)の各項目に対して接続限界本数を調べる試験は以下のように行った。

1. 負荷シミュレータを用いてコンテンツを10秒ごとに、FEC CPUが過負荷になるまで追加し、短時間での接続限界本数を調べる
2. 上記接続限界本数において1時間のエイジングを行って、ライブ FEC サーバの各種ログを確認し、特に FEC CPU やシェーパでパケット廃棄や処理エラーが起きていないことを確認して接続限界本数とした  
(デコーダの試験の際は対向で用いたエンコーダ側でもパケット廃棄や処理エラーが起きていないかも併せて確認した)

<ハードシェーパ機能評価試験>

FEC サーバ(エンコーダ)からの出力パケット間隔を Data Quality Analyzer にて計測し、ハードシェーパ設計仕様から計算されるパケット間隔との比較を行った。

4-4-4-4 エンコード性能評価試験結果

(1) 評価試験構成

ライブ FEC サーバエンコード性能評価試験における試験構成を図 5-4-3-3-1 に示した。

## (2) 各評価試験項目結果

まず、前述のライブ FEC サーバ基本設定の下でエンコード接続限界本数を測定したところ、6Mbps コンテンツ×50 本であった。基本設定の元で各項目のパラメータを変化させたときの接続限界本数の変化は以下のようになった。

### ① コンテンツレート

接続限界本数はコンテンツレートにほぼ反比例する。

### ② コンテンツ長

1 分コンテンツ、10 分コンテンツ共に接続限界本数は 50 本である。RTSP のセッションを切断、接続する回数が 10 倍になるが、RTSP を処理する部分を Main CPU が行い、FEC 処理を 10 個の FEC CPU が行うというように処理が分担されているため、セッション切断、接続回数によって接続限界本数は減少しにくいと考えられる。

### ③ プロトコルバージョン

Raptor 7 と Raptor 9 では限界接続本数には差は見られない (共に 50 本)。

### ④ Encoding Unit Size

エンコーディング・ユニット・サイズが小さくなるほど、処理するパケット数が増加し、FEC CPU への負担が増加するので接続限界本数は減少する。

### ⑤ Source Block Size

FEC 処理はソースブロックサイズ分のパケットをためてから開始するため、ソースブロックサイズが大きくなるほど FEC CPU の負荷が上がり、接続限界本数は減少する。

### ⑥ Packet Loss Rate

パケット欠損率が高いほど、Protection Amount が増加し FEC エンコード処理負荷が高くなるため、接続限界本数は減少する。

### ⑦ Target Loss Rate

ターゲット欠損率の変化による Protection Amount の変化はほとんどない (Target Loss Rate=1.00e-08 の場合 45、Target Loss Rate=1.00e-09、1.00e-10 の場合 46) ため、接続限界本数には変化は見られない。

### ⑧ FEC CPU 数

(1 つの FEC CPU でエンコード処理できる限界本数×FEC CPU 数) が接続限界本数となると予想され、実際評価結果はそのように線形となっている。

### ⑨ FEC エンコードしないストリーム数

FEC エンコードしないストリームは FEC CPU を通らず Main CPU から出力ポートへスルーされるため、Main CPU での処理オーバーにならない限り、FEC オ

フのストリーム数は、FEC 処理の接続限界本数には影響しないと考えられ、そのような結果になっている。ただし、FEC エンコードしないストリーム数が 40、50 本の際には、FEC サーバ内のシェーパにて Discard がわずかに(40 本の場合送信パケット数全体の約 0.00002%、50 本の場合約 0.00008%)発生した。これは、配信サーバ (StreamPro) が配信する際、配信本数が多くなるとバースト的に送信される割合が高くなり、シェーパの Queue あふれが発生することが原因と考えられる。

#### 4-4-4-5 デコード性能評価試験結果

##### (1) 試験構成

ライブ FEC サーバデコード性能評価試験における試験構成は、添付資料を参照。

##### (2) 各試験項目結果

まず基本設定の下でデコード接続限界本数を測定したところ、6Mbps コンテンツ×30 本であった。次に、基本設定の元でコンテンツレートを変化させたときの接続限界数の変化を示す。この結果から、接続限界本数はコンテンツレートにほぼ反比例する事が分かる。

その他のパラメータを変化させた時の結果は、添付資料参照。

#### 4-4-4-6 ハードシェーパ機能評価試験結果

##### (1) 試験構成

負荷シミュレータにつながる GigaSW のポートをミラーリングし、Data Quality Analyzer を用いてパケット間隔の測定を行った。

##### (2) 試験結果

基本設定において、6 Mbps コンテンツ 1 本を配信し、Data Quality Analyzer を用いてライブ FEC サーバ (エンコーダ) から送出されるパケットの間隔の測定を行った。1 分間の測定を行ってヒストグラムにしたものは、添付資料参照。最も頻出のパケット間隔 (0.8ms 以上 0.9ms 未満) が、シェーパから連続して送られるパケット送信間隔を表し、3.5ms 以上に現れているパケット間隔は、StreamPro からの配信方法における、500ms 中の送信休み時間に対応している。

また、基本設定以外にも、以下の 8 項目について 6 Mbps コンテンツ 1 本を流したときの FEC サーバからの送出パケット間隔のヒストグラムデータ測定を行った。

- **Encoding Unit Size : 256Byte、512Byte**
- **Source Block Size : 64KB、128KB**
- **Packet Loss Rate : 1%、5%**
- **Target Loss Rate : 1.00e-08、1.00e-10**

パケット送信間隔は、

$N \text{ 値} \times 20 \text{ [ns]} \times \text{パケットサイズ [Byte]}$

で計算される。FEC パラメータ等の設定に対応してシェーパ送出ビットレートが

決まり、シェーパ送出ビットレートに対応してN値が決まる。よって FEC パラメータの設定が変わるとパケット送信間隔は変化する。パケット送信間隔の計算値と測定値（ヒストグラム中最も頻度が高いパケット送信間隔）を比較したものが表 5-4-3-3-5 である。

また、基本設定にて限界接続本数を配信し、そのうちの 1 本を Data Quality Analyzer にてパケット間隔測定してヒストグラムにまとめたものは、添付資料参照。1セッションの時に比べて、セッションが多くなると、シェーパの Queue からの出力が重なることで出力 Delay が発生することにより、頻出のパケット間隔以外のところが増加していると考えられる。他の 11 項目についても同じような傾向が見られた。

#### 4-4-5 ライブ FEC サーバの負荷低減を意識したシステム構成に関する考察

ライブ FEC サーバの性能評価試験結果より、FEC ありと FEC なしが混在すると、FEC なしの処理でも CPU リソースがかなり使われてしまうことが分かった。そのため、ライブ FEC サーバ自体に FEC なしのデータが流れない構成にして、高価なライブ FEC サーバの設置数を減らし、処理可能なストリーム数を増やすことが考えられる。

##### 【考えられるシステム構成】

前提：コンテンツ自体には FEC は入っておらず、ライブ FEC サーバで透過的に FEC を付与する

① ライブ FEC サーバが RTSP をスヌープして FEC の有無を判別する方式(現状の方式)

- ・メリット：ストリームサーバの構成を変えることなく、**シンプルな構成**。
- ・デメリット：FEC サーバに、FEC ありと FEC なしが混在し、**パフォーマンスが低下する**。

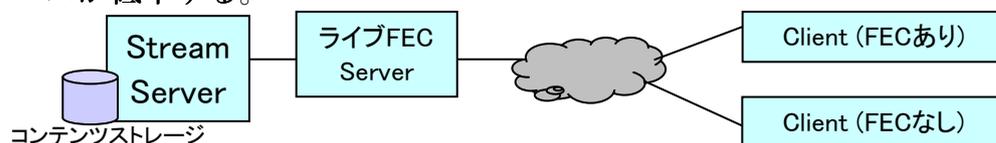


図 4-4-6-1：現状の方式

② ストリームサーバは 1 台だが、FEC ありの時となしの時でストリームサーバにアクセスする IP アドレスが異なる方式

- ・メリット：ストリームサーバの構成を一部変更するが、**シンプルな構成**。
- ・デメリット：ライブ FEC サーバに、FEC ありと FEC なしが混在し、**パフォーマンスが低下する**。

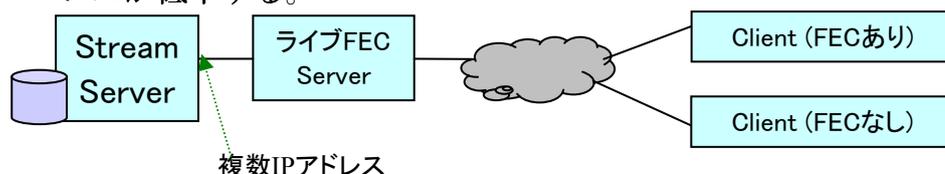


図 4-4-6-2：複数 IP アドレスを付与する方式

- ③ ストリームサーバに複数の NIC を組み込み、それぞれに別の IP アドレスを付与する方式
- メリット：ライブ FEC サーバには、FEC ありのデータしか流れないため、パフォーマンスが低下しない。
- ・デメリット：ストリームサーバのインタフェースを増やすなど、構成変更が必要。
  - ・ストリームサーバがボトルネックになる可能性がある。

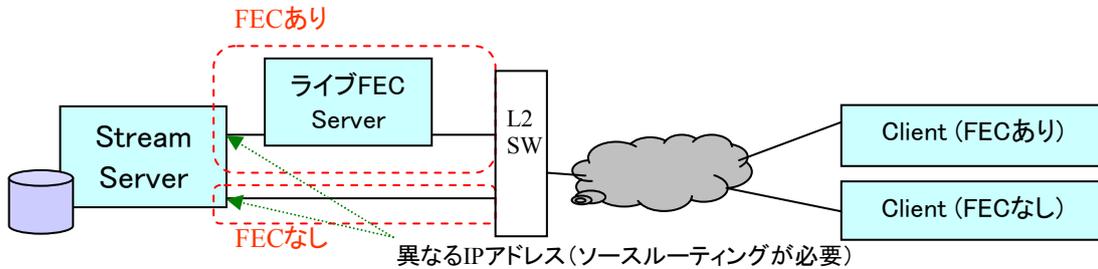


図 4-4-6-2 複数の NIC を組み込む方式

- ④ FEC ありの時と FEC なしの時でアクセスするストリームサーバを別にする方式
- ・メリット：ライブ FEC サーバには、FEC ありのデータしか流れないため、パフォーマンスが低下しない。ストリームサーバを増やすことで、ストリームサーバがボトルネックになりにくい。
  - ・デメリット：ストリームサーバを増設しなければならない。システムが複雑になる。

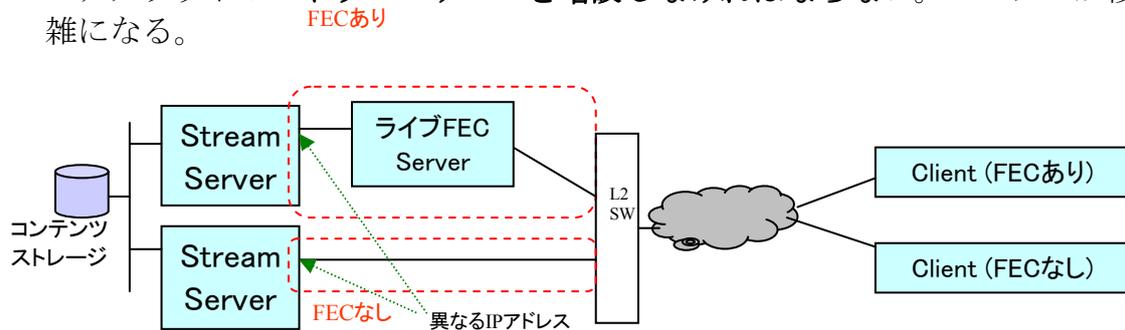


図 4-4-6-3：複数のストリームサーバを用意する方式

①と②は、あまり違いがない。したがって、負荷分散には③か④のどちらかの手法が適切である

#### 4-4-3-6 今後の方針

今回は実施しなかったが、ライブ FEC サーバ 2 台、3 台をカスケード接続して限界ストリーム数の調査を今後検討する必要がある。またハードとソフト併用によるシェーピング方式の検討がある。

#### 4-4-6 結論

TCP 系の配信と UDP 系の配信を、インターネット上で混在させる場合の方法に

ついて調査を行い、TFRC という方法が有効であるとの結果を得た。

#### 4-5 総括

誤り訂正符号に関する理論的な検討から、Digital Fountain 社の誤り訂正符号が有力であるとの結論を得た。実機で、Reed-Solomon 符号と Raptor 符号の処理速度の比較を行った結果、Raptor 符号が有利であるとの結論を得た。インターネット上のパケット欠損とジッタの関係をモデル化し、その関係について検討を行った。また、商用ネットワーク上で FEC を用いた映像配信の実証実験を行い、FEC の有効性を確認した。

## 5. 参考資料、参考文献

### 5-1 研究発表・講演等一覧

#### ・学会発表

- (1) “インターネット映像配信における Reed-Solomon 誤り訂正符号と Multi-Stage 符号との比較検討”, 森田 哲郎, 西本 裕明, 佐々木 隆志, 高橋 豊, 電子情報通信学会 信学技報 CQ2003-36.
- (2) “XOR-based coding に基づく効率的なマルチキャスト映像配信制御の提案”, 高橋潤, 戸出秀樹, 村上孝三, 電子情報通信学会 信学技報 NS2003-209 PN2003-37.
- (3) “サポート配信機能を有するメタコンテンツのマルチキャスト配信制御”, 久野和英, 高橋潤, 戸出秀樹, 正城敏博, 村上孝三, 電子情報通信学会 信学技報 NS2003-210 PN2003-38
- (4) “Erasure Code を用いたマルチサーバ配信方式の検討”, 藤枝 太一, 久野 和英, 高橋 潤, 戸出 英樹, 正城 敏博, 村上 孝三, 電子情報通信学会 2004 年 総合大会

#### ・論文発表 (査読あり)

- (1) “Comparison of Loss Resilient Ability between Multi-Stage and Reed-Solomon Coding,” Tetsuo Morita, Hiroaki Nishimoto, Takashi Sasaki, and Yutaka Takahashi, 11th International Conference on Telecommunication Systems.
- (2) “Performance Analysis of HC/RC for Rich Content Distribution,” Kaesinee Thongsisod, Takashi Sasaki, Tetsuo Morita, and Yutaka Takahashi, 11th International Conference on Telecommunication Systems.
- (3) “Packet Loss and Delay Jitter Characteristics of the Internet -VoD Trial Service with Packet Recovery-,” Tetsuo Morita, Yutaka Majima, Toshihiro Takashima, Takahiro Kusumoto, Takashi Sasaki, and Yutaka Takahashi, 12th International Conference on Telecommunication Systems, Monterey (USA), July 24 (2004).

#### ・ニュース解説

- (1) “ブロードバンドコンテンツ配信用の高速パケット欠損補償技術を開発 -フィールドトライアルサービスにおいて有効性を実証-”, 電子情報通信学会 ニュースレター, 2004 年 8 月号

以上