平成 **16** 年 **3** 月修了

博士（工学）学位論文

（和文題目）**Content Delivery Networks** における 映像コンテンツ配信時の品質維持制御に関する研究

（英文題目）**Study on Sustaining The Quality of Video Content Retrieved from Content Delivery Networks**

平成 **15** 年 **12** 月 **26** 日

高知工科大学大学院 工学研究科 基盤工学専攻（基盤工学コース）

学籍番号 **1056013**

中平 拓司

Takuji Nakahira

目次

論文要旨

本論文は，IP 網の上に構築されたコンテンツ配信ネットワーク（Content Delivery Networks: CDN）を通して提供される映像コンテンツについて，配信時の品質を維持するための制御技術に関する研究成果を述べたものである．論文は，第 1 章「序論」，第 2 章「並列転送による品質維持制御」，第 3 章「コンテンツ配布時における性能改善」，第 4 章「品質維持制御のための利用可能帯域通知法」，第 5 章「結論」の 5 つの章から構成されている．

第 1 章では，本研究に至った背景を整理した．一般に，ネットワークを介して映像を配信するには，その解像度や符号化処理によって決まるビットレートを送受信者間で確保する必要がある．しかし，現在の TCP/IP に基づいたネットワークアーキテクチャでは，ネットワークおよびサーバにかかる負荷が時間と共に変動し，かつその振る舞いを予測するのは困難である．従って，受信，再生される映像の品質を保つことが難しい．そのため，配信対象のコンテンツを複数の広域に分散配置されたサーバ群に複製しておき，クライアントからのアクセスを，ネットワークおよびサーバの状態に応じて最適なサーバに誘導することでネットワークとサーバの負荷分散を図り，品質保証を目指す CDN の概念が現れた．

第 2 章では，CDN によって映像コンテンツが配信される場合，「コンテンツが複数のサーバに複製配置されていることを利用して再生品質を可能な限り保つ配信制御法はないか」という点に着目し，検討を進めた．その結果，1) コンテンツがミラーリングされている環境においてクライアントへの転送量を前回の転送時の転送速度に比例させる方法，2) コンテンツをストライピングすることで記憶容量のコストを下げながらも一定の品質維持に寄与させる方法を提案する．LAN および JGN を利用したテストベッドにおいて，通常転送時にはアンダーフローを起こし，再生が中断してしまう負荷状態であっても，提案方式を採用することにより再生が中断されず，本来の時間・空間解像度での再生が続行されることを実証した．

続いて第 3 章では，上記 1)の方式を，コンテンツのミラーリング時に活用する方法について検討し，次のように利用することを考えた．始めに，配布対象のコンテンツを N 等分し，配布先となる N 台のミラーサーバにストライピン

グする．続いて，各ミラーサーバが残り N-1 個のブロックをオリジンサーバと他のミラーサーバから 1)の方式によって受信する．こうすることにより，ネットワーク品質の変動があってもそれに応じて転送速度が最大化され，ボトルネックに依存せずに配布時間の短縮につながる．前述のテストベッドを用いた実験により，この配布方式が通常の配布に対して 1.2 倍から 1.5 倍の速度向上が達成されることを示した．

本研究は，「クライアントでの再生品質を維持した上で，システム全体の負荷を最も良く分散するには，N 台のサーバのうちどの K 台からどういった分量で配信してもらえば良いか？」という問題に帰着される．第 4 章では，この問題を解くのに要求されるエンドホスト間の利用可能帯域を取得するモデルを検討した．その結果，自律システム（AS:Autonomous System）ごとに経路と利用可能帯域をチェック，保持するモニタリングサーバを置き，送信ホストからのプローブパケットを中継する方式を導いた．提案方式を実装し，既存の方式と比較した結果，測定時間はミリ秒単位，経路を通過するパケット数は 2 個で済み，0.5 秒というプローブ間隔で帯域幅の変動に追従可能であることが明らかとなった．また，モニタリングによるノード負荷およびネットワーク負荷に関しても，0.2 秒程度のモニタリング間隔であれば，システムの運用に支障のない範囲であることがわかった．

最後に，第 5 章において本研究で得られた成果をまとめ，今後の検討課題，将来展望について整理する．

第 1 章

序論

この章では，まず，現在の IP 網における映像配信を支えている要因として，ネットワークの高速化と符号化技術の進展を振り返り，主な配信技術を分類・整理する．次に，映像配信時に要求される品質保証の問題を取り上げ，CDN のアーキテクチャと既存の映像配信リクエスト誘導方法を述べる．

1.1 映像配信の技術的背景

1970 年代に米国で運用が開始されたインターネットでは，当初はファイル転送やニュース購読，電子メール，遠隔操作といったアプリケーションが利用されていた．90 年代に入り，WWW(World Wide Web) が急速に普及，インターネットの商用解放とも相まって，インターネットへの接続端末数・利用者ならびにそのネットワークエリアは世界中に爆発的に広がった [1]．現在では，もはやインターネットは社会基盤の一つとして，企業活動あるいは個人の情報収集・コミュニケーション手段としてなくてはならないものとなっている [2]．

そうした中，従来のテキストあるいは静止画だけでなく，動画像をインターネットを通して配信しようとする試みが始まる [3]．しかし，動画像を安定して配信するには相応の帯域を必要とするため，当初は満足のいく映像コンテンツを提供するのは難しかった．

ところが，ここ数年にわたり，ネットワークの高速化が進み，また，映像のディジタル化と情報量圧縮のための符号化技術の進展といった要因により，映像コンテンツの制作・配信が容易になりつつある．まず，これらの背景技術の進歩を概説する．

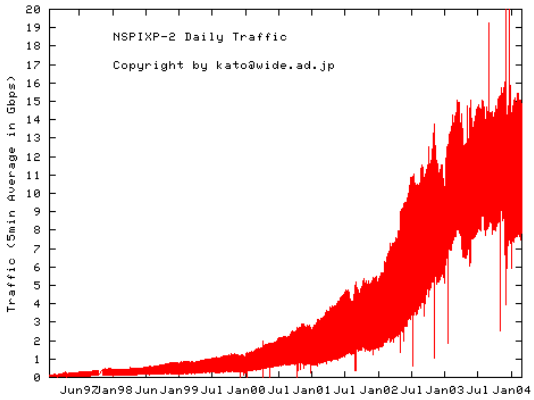


図 1.1　NSPIXP-2 におけるトラヒック量の推移

1.1.1　ネットワークの高速化

インターネットの利用者増加に伴い，そこを流れるトラヒック量も指数関数的に増加し続けている．例として，日本の主要な IX（Internet eXchange）の一つである，NSPIXP-2[4] における 1 日あたりの平均トラヒック量の推移を図 1.1 に示す．これを見ると，トラヒック量は指数関数的に増加しており，2003 年 1 月以降は 15Gbps を超えている．これは，国内において，DSL(Digital Subscriber Line）や CATV，さらには FTTH（Fiber To The Home）といった数 Mbit/s から最大 100Mbit/s の帯域を持ったアクセス網の安価な常時接続が一般的となったためといえる．

総務省が毎年発行している情報通信白書 [5] によると，図 1.2 にあるように，DSL 回線の契約数が大幅に増えており，1000 万人を突破しようとしている．最近では，光ファイバを家屋もしくは集合住宅まで引き込む FTTH の利用者も，首都圏を中心に増加している．

また，他国と比べて日本では，携帯電話からのインターネット接続利用が非常に多いのも特徴である． 図 1.3 を見ると，インターネット接続機能を持った携帯電話の出荷台数は 6000 万台に達している．無線ということで，有線メディアに比べて速度を上げるのが難しかったが，ここ数年，第 3 世代（3G）規格である IMT-2000[6] に対応した端末が市場に流通するようになった．

LAN に目を向けてみると，Ethernet が事実上の標準として定着し，PC や WS に

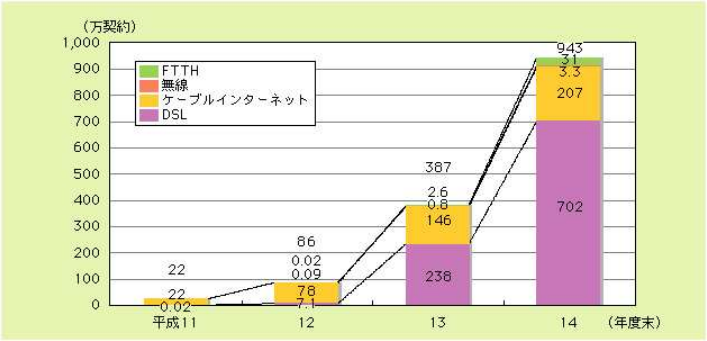


図 1.2　ADSL/CATV/FTTH の加入者数の推移

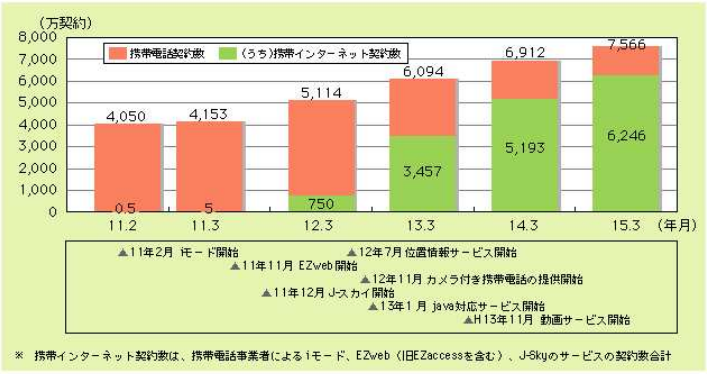


図 1.3　インターネット接続機能を持った携帯電話の契約数

100Base-TX NIC(Network Interface Card) が装備されている状況となった．さらにここ 1,2 年のうちに，理論値で 1Gbit/s の通信速度を提供する Gigabit Ethernet が普及し始めている [7]．Gigabit Ethernet は，表 1.1 に示される規格があり，端末では，既存の UTP ケーブルをそのまま使用できる 1000BASE-T(IEEE802.3ab)，サーバや基幹網には，1000BASE-SX/LX(IEEE802.3z) の導入が進んでいる．2003 年 5 月には，光ファイバを媒体として理論値で 10Gbit/s の伝送速度を持つ 10Gbit Ethernet(IEEE802.3ae) が標準化された [8]．

LAN の場合，無線 LAN の標準化と導入も進んでいる [9]．現在利用されている主な標準を表 1.2 に示す．このうち，IEEE802.11b（11b）が最初に標準化され，無線 LAN カードやベースステーションといった製品が市場に出された．ただし，11b で使われている 2.4GHz

表 1.1　Gigabit Ethernet の種類

名称	媒体	最大伝送距離
1000BASE-T	CAT-5 以上の UTP ケーブル	100m
1000BASE-SX	マルチモード光ファイバ	550m
1000BASE-LX	マルチモード/シングルモード光ファイバ	550m/5km

表 1.2　無線 LAN の種類

名称	周波数帯	最大伝送速度
IEEE802.11b	2.4GHz	11Mbit/s
IEEE802.11a	5GHz	54Mbit/s
IEEE802.11g	2.4GHz	54Mbit/s

帯は，他の電波との干渉があるため，伝送速度は最大 11Mbit/s に抑えられる．さらに，バス方式により帯域を共有しているため，接続台数に比例して速度は低下する．このため，より高速化した規格として，IEEE802.11a（11a），IEEE802.11g（11g）がその後標準化された．11a は 5GHz 帯を使用し，最大伝送速度は 54Mbit/s となっている．一方，11g は 11b と同じ 2.4GHz 帯を使用した上位規格で，11a と同じく，最大伝送速度は 54Mbit/s となっている．

ネットワークの高速化は，上記のようなリンク速度が上がるだけでなく，ルータやスイッチと行ったノードの処理速度がそれに追従して初めて可能となる．特にネットワークの規模が大きくなるにつれて，図 1.4 に示されるように，ルータのパケット転送処理能力が足りなくなることが問題視され始めた．

そのため，従来のソフトウェアによるルーティング処理をハードウェア化し，IP アドレスによるパケットのスイッチングを行うレイヤ 3 スイッチが現れた [10]．最近流通しているレイヤ 3 スイッチは，指定したトラヒックに対して帯域を確保したり，クラスごとにサービス品質（Quality of Service: QoS）を設定する機能を有することが多く．普及が進んでいる．

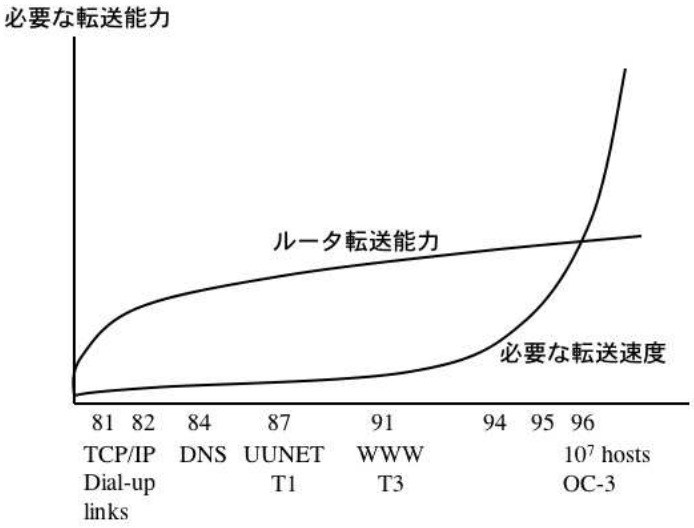


図 1.4　従来型ルータの転送能力の限界

一方，IP アドレスの代わりに「ラベル」と呼ばれる識別子を導入し，これによって高速なスイッチングを行う技術が考案され，IP スイッチ，CSR(Cell Switch Router)，タグ・スイッチといった製品が登場した．これらの技術は後に MPLS(Multi-Protocol LabelSwitching) として標準化された．MPLS の技術については後述する．

1.1.2　符号化技術の発展

ネットワーク技術の進歩により，安価で広帯域な伝送媒体が利用可能となったが，そのまま映像をパケット化して配信すると，SDTV 品質で数 100Mbits/s，HDTV 品質では数 Gbits/s といった帯域が必要となり，多くのユーザが同時にこうしたトラヒックを流すことはまだ難しい．

一般に，図 1.5 にあるように，X 画素 $\times Y$ 画素でディジタル化された動画像のビットレート B[bits/s] は，1 画素あたり a[bits] 使用し，秒間のフレーム数が f[fps] である場合，次式により求められる．

$$B = a \times X \times Y \times f \tag{1.1}$$

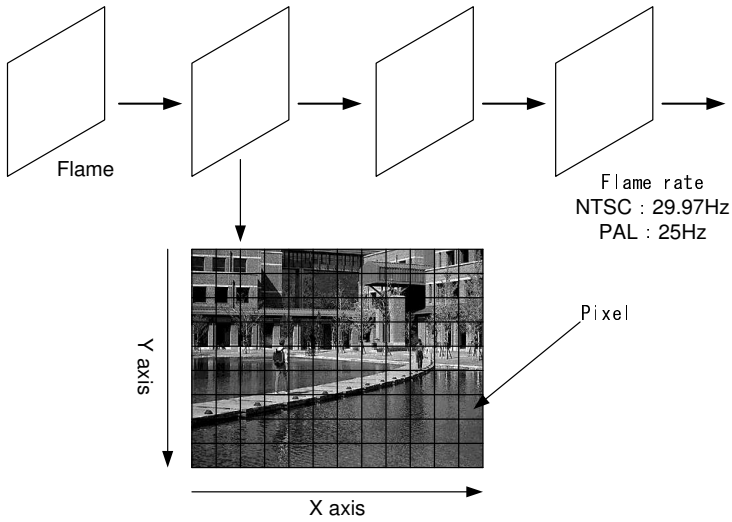


図 1.5　ディジタル動画像のビットレート

ここで，NTSC ビデオを，RGB 各 8bit の和でカラー表現された 720×480 画素の解像度でディジタル化したとすると，B は，a=24, X = 720, Y=480, f=30 を式 (1.1) に代入し，

$$B = a \times X \times Y \times f = 24 \times 720 \times 480 \times 30 = 248.8 Mbit/s \tag{1.2}$$

となる．つまり，通常のテレビジョン品質の映像を伝送するには，約 250Mbit/s の帯域が必要となる．当然，ネットワークだけでなく，端末内部の処理速度もこれ以上でないといけない．

テレビジョン信号をディジタル化する場合，ITU-R 601（旧 CCIR-601）[11] にあるように，カラー表現を，Y（輝度），U（色差 B-Y），V（色差 R-Y）の 3 成分で表す．そして，明るさの変化には敏感だが，色の変化には鈍感であるという人間の視覚特性を利用し，図 1.6 に示すように U, V 成分を X 方向に 1/2 にカットする．このフォーマットを，YUV4:2:2 と呼ぶ．こうすることで，映像のビットレートが 2/3 になる．さらに，図 1.7，図 1.8 に示すように U, V 成分をさらに間引いたものを，YUV4:2:0，YUV4:1:1 といい，これらの場合，ビットレートは元の 1/2 となる．DVD(Digital Versatile Disk) では現画像を YUV4:2:0 で表し，DV(Digital Video)[17] では YUV4:1:1 を利用している．

さらに圧縮率を上げるべく，まず静止画を対象とした符号化の国際標準として JPEG が

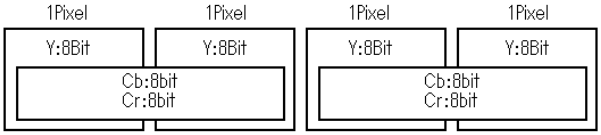


図 1.6　YUV4:2:2

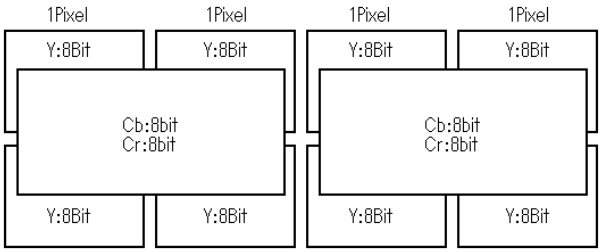


図 1.7　YUV4:2:0

1993 年に採択された [12]．次いで，動画像符号化の国際標準として，MPEG-1 が 1994 年に採択されると，以後，MPEG-2, MPEG-4 と標準化が続いた [13][14]．2000 年には，次世代の JPEG 規格として，JPEG-2000 が標準化された [15]．これらの位置付けと技術的特徴を，表 1.3 に整理する．

JPEG の符号化・復号処理のブロック図を，図 1.9 に示す．まず，入力画像は RGB 成分から YUV 成分に変換され，128 引いた値にレベルシフトされる．次いで，8×8 画素のブ

表 1.3　JPEG/MPEG の種類

名称	使用技術	応用例
JPEG	DCT, ハフマン符号化	ディジタルカメラなど
MPEG-1	DCT, MC, ハフマン符号化	Video CD など
MPEG-2	DCT, MC, ハフマン符号化	ディジタル放送，DVD など
MPEG-4	エラー耐性，オブジェクトベース符号化など	携帯端末への配信など
JPEG-2000	離散ウェーブレット変換，算術符号化，ROI など	適応配信など

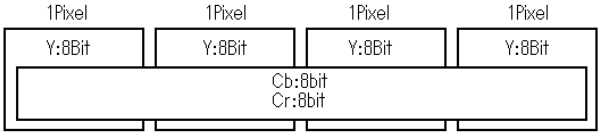


図 1.8 YUV4:1:1

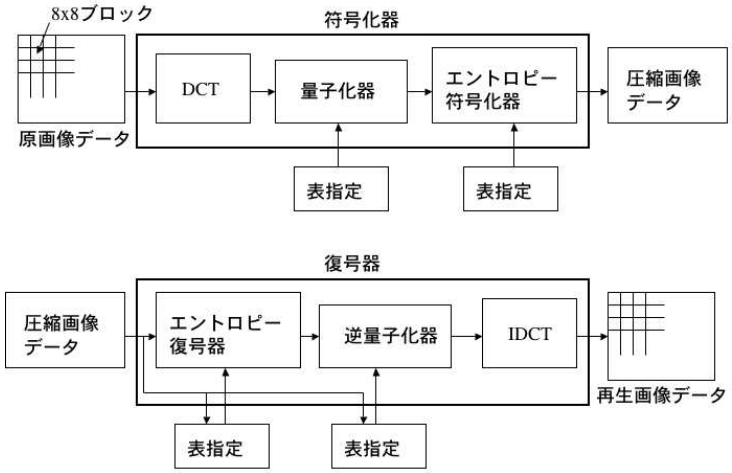


図 1.9 JPEG の処理ブロック

ロックごとに，直交変換の一つである DCT（Discrete Cosine Transform: 離散コサイン変換）が適用される．この場合，$N \times N$ 画素の 2 次元データ $f(x, y)$ の DCT 変換 $F(u, v)$ は式 (1.3) で表され，図 1.10 に示す基底ベクトルの振幅の和として出力される [16].

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos(\frac{\pi(2x+1)u}{2N}) \cos(\frac{\pi(2y+1)v}{2N}) \tag{1.3}$$

また，$F(u, v)$ から $f(x, y)$ への逆変換（Inverse DCT）は，次式に示すようになる．

$$f(x,y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos(\frac{\pi(2x+1)u}{2N}) \cos(\frac{\pi(2y+1)v}{2N}) \tag{1.4}$$

このとき，ブロックの左上の DCT 係数は DC 成分，その他の係数は AC 成分と呼ばれるが，AC 成分は，周波数成分が高くなるほど（ブロックの右下に向かうほど）人間の目に感じにくいため，大幅に削減可能である．このため，DCT 係数の量子化ステップを右下に向

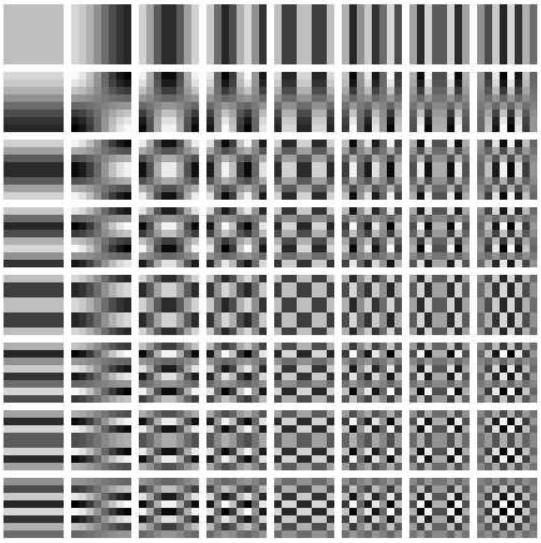
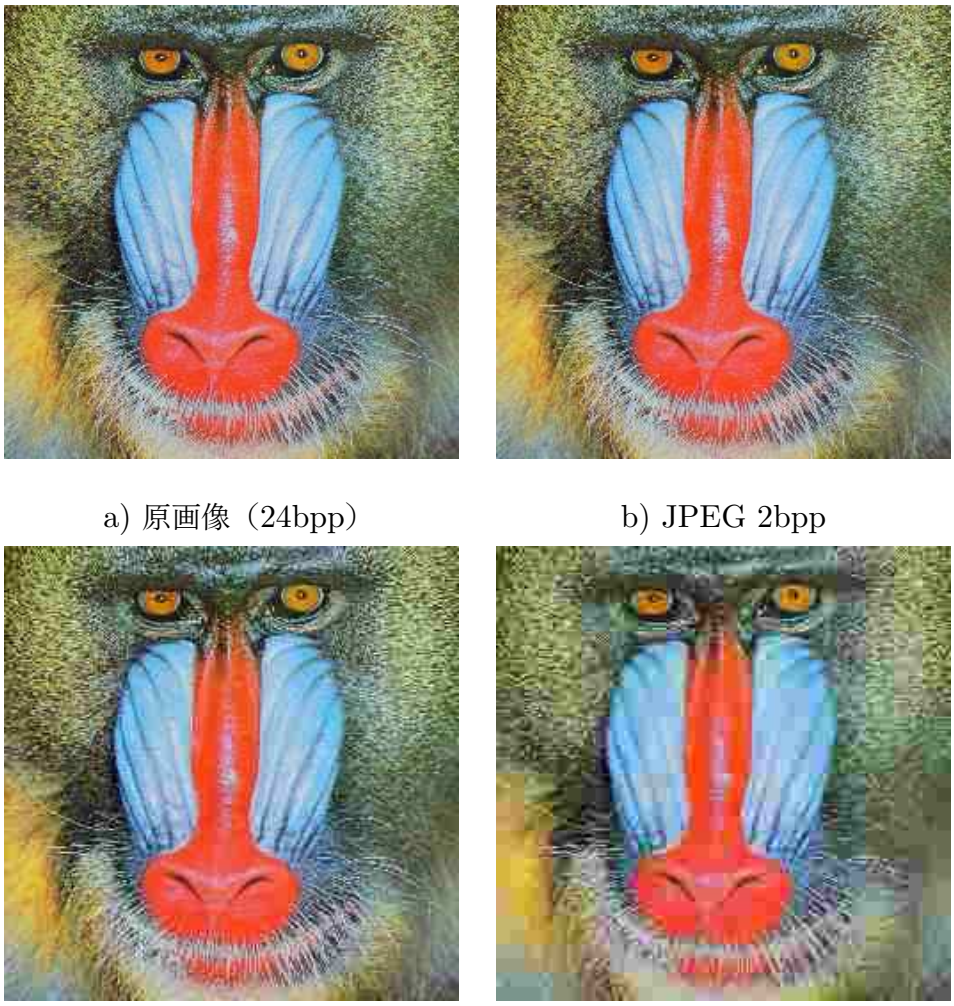


図 1.10　2 次元 DCT の基底ベクトル

かうにつれて大きく取り，量子化する．通常の画像であれば，量子化後の DCT 係数は，右下の基底ベクトル群係数として 0 が続くことになる．そして，出現頻度確率に基づいて定められた符号化表に従ってハフマン符号化され，0 が連続している箇所があるため，大幅に情報量が削減される．JPEG 圧縮後，復号した画像と原画像とを比較したものを図 1.11 に示す．

MPEG では，JPEG の処理を基本としながら，フレーム間の予測符号化を導入している．ここでは，MPEG-1 における符号化器および復号器の処理ブロックを，それぞれ図 1.12 と図 1.13 に示す．原画像の YUV 変換・レベルシフトは JPEG のときと同様であるが，DCT を適用するのは，現在のフレームと前のフレームとの動き補償予測誤差となっている．動き補償予測（Motion Compensation: MC）とは，図 1.14 に示すように，フレーム内の動く物体を検出し，その動きを予測してその結果と現フレームとの誤差を取るものである．これにより，動画像の時間領域における冗長性を除去できる．ゆえに，MPEG の符号化器（Encoder: エンコーダ）には復号部（Decoding Unit: デコーダ部）もある．

MPEG のフレームには，I ピクチャ・P ピクチャ・B ピクチャの 3 種類のピクチャがあり，図 1.15 にあるように，これらの組み合わせによってシーンを構成する．I ピクチャは，他の



a) 原画像（24bpp）　b) JPEG 2bpp

c) JPEG 1bpp　d) JPEG 0.5bpp

図 1.11　JPEG 画像の例（bpp = bit per pixel）

フレームからの予測を行わず，自身のデータからフレーム内相関のみで符号化する．P ピクチャは，過去のフレームから予測を行い，その差分を符号化したもの，B ピクチャは過去と未来のフレームの両方向から予測を行い，その差分を符号化したものである．B ピクチャが最も符号量が少なくなるが，伝送途中のパケットロスやエラーの影響を抑えるため，I ピクチャが周期的に挿入される．I ピクチャから次の I ピクチャまでのピクチャのパターンを GOP（Group Of Pictures）と呼ぶ．

最後に，近年民生用のビデオ機器に普及している DV について述べる．MPEG では，空間領域における冗長性の除去に DCT を使い，時間領域における冗長性の除去には MC を

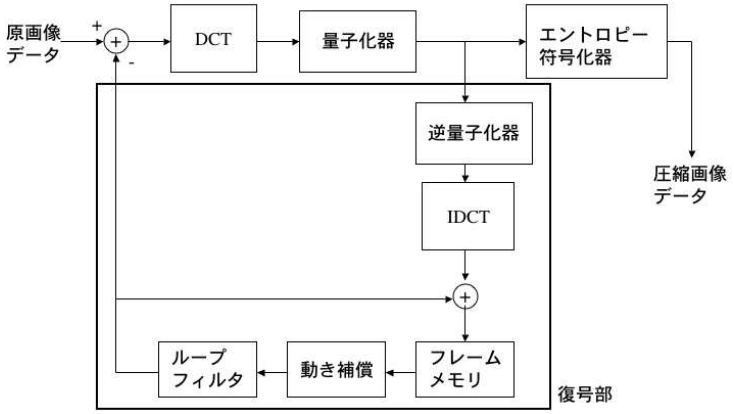


図 1.12　MPEG-1 の処理ブロック（符号化器）

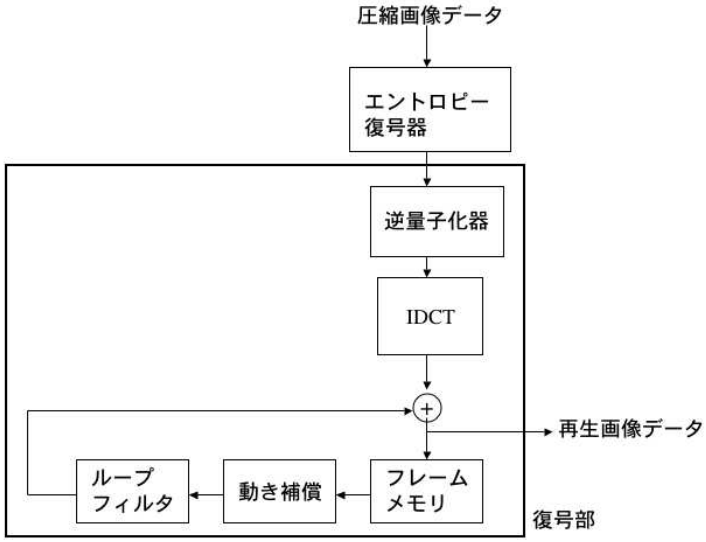


図 1.13　MPEG-1 の処理ブロック（復号器）

使っていた．このため，SDTV 品質の映像であれば 4 Mbit/s から 6Mbit/s に，HDTV 品質でも 20 数 Mbit/s というビットレートにすることが可能となっている．しかし，MC は処理負荷が高い，符号化後のフレームあたりのサイズが可変長であることから編集などの処理に不向きである，といった短所がある．それに対して DV では，符号化後のビットレートは MPEG より高くなるが，フレーム内符号化であり，符号化後のフレームあたりのサイズが固定長であるという点から，ストリーミングなどでの利用が進んでいる．

DV の符号化処理の流れを，図 1.16 に示す．入力画像は 720x480 の大きさで，YUV4:1:1

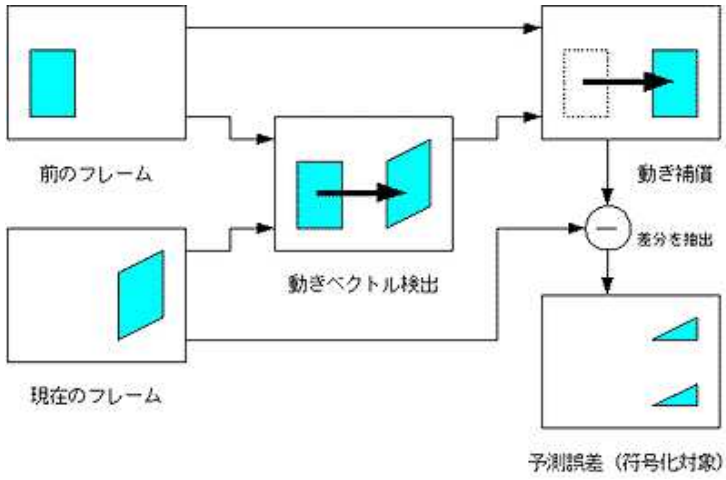


図 1.14　動き補償予測

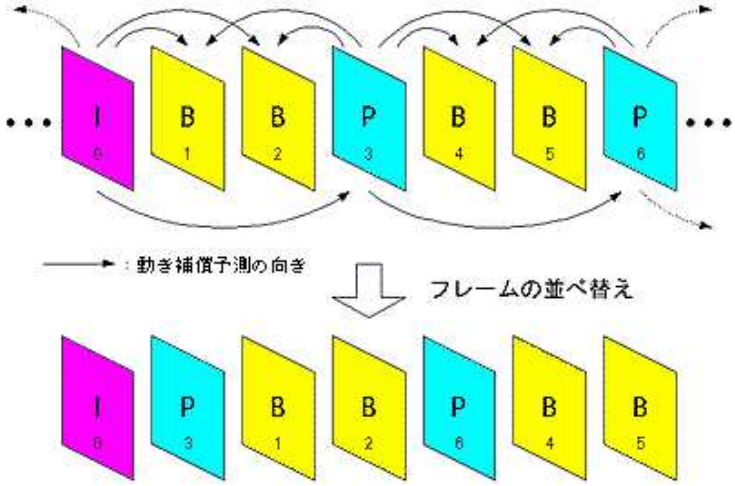


図 1.15　ピクチャの種類

に変換される．次に，JPEG, MPEG のときと同様にブロックごとに DCT が適用される．その後，DCT 係数は量子化され，ハフマン符号化される．DV の符号化データは，80Bytes の DIF Block を基本単位とし，図 1.17 に示すように 1 フレーム分のデータが階層的に出力される [18]．NTSC ビデオの場合，1 フレームは 10 個の DIF Sequence から成り，DIF Sequence は 150 個の DIF Blocks から成り立っている．ゆえに，ビットレート B は，

$$B = 80 \times 150 \times 10 \times 29.97 = 120,000 Bytes/frame \times 29.95 frames/s$$

$$= 3,594,000 Bytes/s \simeq 28.8 Mbit/s \tag{1.5}$$

となる．

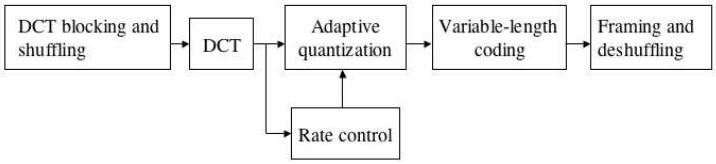


図 1.16　DV の処理ブロック

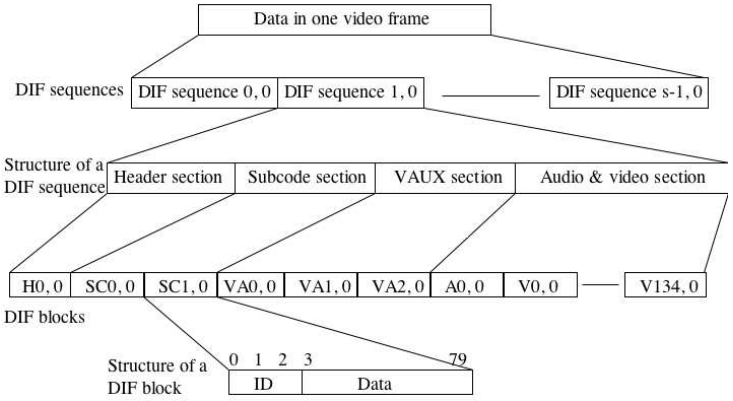


図 1.17　DV のデータフォーマット

1.2　映像配信の種類

このように，ネットワークの高速化と符号化技術の進展によって，IP 網を通した映像配信が現実のものとなってきた．一口に映像配信と行っても様々な種類のものがあるため，ここで，アプリケーションによる分類と，配信手法による分類をしておく．

1.2.1　アプリケーションによる分類

映像を配信することでどういう使い方をするのかという視点で分類すると，現在ある映像配信は，次のいずれかに属すると言える．

1. コンテンツ配信型

映画，テレビ番組，スポーツイベントなど何らかの動画コンテンツを，配信サーバから，要求のあったクライアントに配信するもの．これにはさらに二つの種類がある．一つは，現在進行中の事象をリアルタイムに配信するライブ配信，もう一つは，コンテンツを配信サーバにあらかじめ蓄積しておき，クライアントからの要求に応じて配信するオンデマンド配信である [19]．現在インターネット上で広く使われているアプリケーションとして，Windows Media[20], Real Media[21], Quick Time[22] がある．Windows Media のスクリーンショットを，図 1.18 に示す．

図 1.18　Windows Media/Real Media/Quick Time のスクリーンショット例

2. コラボレーション型

もう一つは，複数のユーザがグループを構成し，その間で互いの映像を送受信し合って，あるタスクを推進するというコラボレーション型である．代表的な例として，ビデオ会議や e-Learning といったものがある．この場合，ユーザ間のコミュニケーションを円滑に取るため，配信される映像（音声を含む）は，遅延なく相手方に到着しなければならない．特に，音声の損失はその大きな妨げとなることが知られている．一般に，許容される音声の遅延時間は 100ms 程度とされている [23]．

アプリケーション例としては，UNIX 系 OS 上で稼働する vic[24] などがある．vic のユーザインタフェース例を図 1.19 に示す．これと合わせて，vat や wb といったツール

図 1.19 vic のユーザインタフェース例

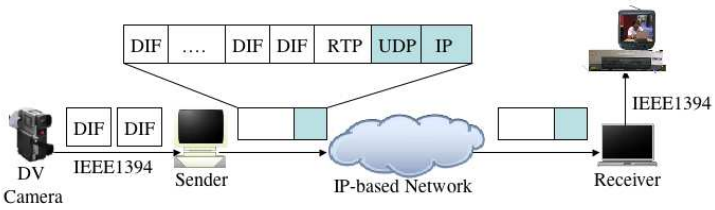


図 1.20 DVTS のシステム構成

も公開されており，複数ユーザ間でのコミュニケーションが可能となっている [25].

もう一つ，より高画質な映像を伝送することを目的としたツールとして，DVTS(Digital Video Transport System) がある [26]．DVTS は，図 1.20 に示すように，DV 映像を IEEE1394 経由で取り込み，その DIF ブロックを RTP パケットとしてリアルタイムに伝送する [27][28]．既存の民生用機器と廉価な PC で高画質・低遅延な配信が実現でき，オープンソース化されていることから，各方面で利用される機会が多い．

1.2.2 配信手法による分類

次に，映像データをどうやって相手方に届けるかという視点で分類すると，次の 3 通りのものがある．

1. ダウンロード

蓄積配信の場合，最も簡単な方法は，配信サーバにコンテンツをファイルとして保存しておき，クライアントからの要求があればそのダウンロードを認めるものである．通常，HTTP や FTP が転送プロトコルとして使われる [29]．ユーザは，ファイルのダウンロードが完了してから，コンテンツファイルを再生する．

2. ストリーミング

ダウンロードでは，コンテンツファイルのサイズが大きいと，クライアントにも大きなディスク空き容量を要求することになる．また，ファイルの転送時間が再生までの待ち時間となってしまう．さらに，ライブ配信の場合，遅延なくクライアントにパケットを送信しなければならない．そこで，図 1.21 に示すように，クライアントのメモリ領域の一部をバッファとして，送信側がコンテンツのビットレートと同じ速度でパケットを転送する．そして，バッファに決められた量だけのデータが溜まったら，再生を開始する．以後．再生と受信を並行する．この技術をストリーミングと呼ぶ．

ストリーミングのためのトランスポートプロトコルとして，RTP(Real-time Transport Protocol) が IETF の AVT Working Group によって定められている [30]．RTP は通常 UDP の上位層として，図 1.22 に示すフォーマットのヘッダを付与し，送信する．パケットの順序を示すシーケンス番号（Sequence Number）や再生時の同期を取るためのタイムスタンプ（Timestamp），どういった種類のデータがペイロードに入っているか識別するペイロードタイプ（Payload Type: PT）などを入れる．

3. プログレッシブダウンロード

ダウンロードとストリーミングとの中間に位置する配信手法が，プログレッシブダウンドードである．ストリーミングの場合，UDP を使って，送信側が一定のレートでパケットを送信することが多いが，UDP トラヒックの通過を拒否するようにファイアウォールが設定されていたり，レートが大きいと帯域を共有している他の TCP トラヒックのスループットを大幅に下げてしまうといった問題がある．そのため，HTTP や FTP によりコンテンツのダウンロードを開始するが，受信次第再生を始めてしまうというものである．Windows Media Technology や Quick Time などは，この手法に対応して

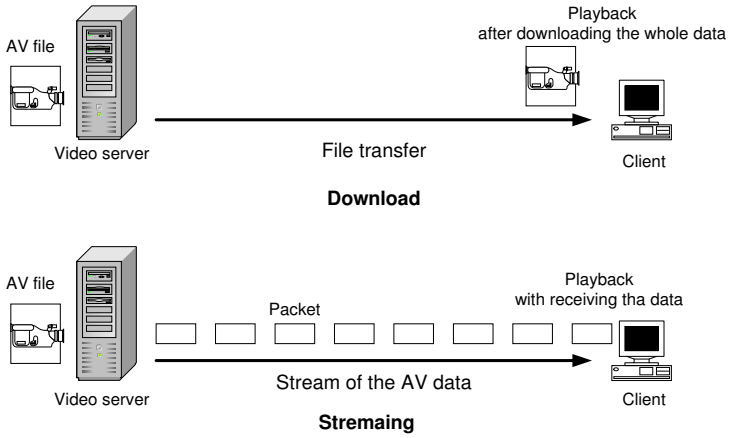


図 1.21　ストリーミング

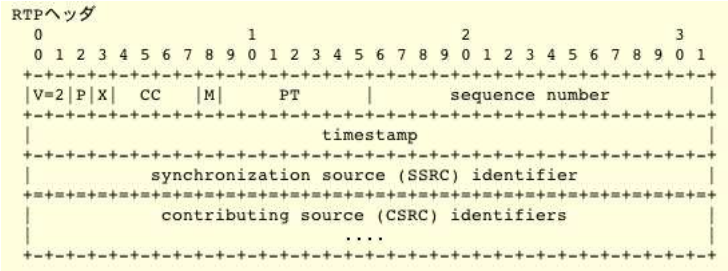


図 1.22　RTP ヘッダフォーマット

いる．

1.3　品質保証への対応

1.3.1　TCP/IP の問題点

コンテンツ配信において，ストリーミングあるいはプログレッシブダウンロードによって品質が低下せずに再生が続けられる条件は，「パケットが制限時間内に到着すること」となる．別の言い方をすると，「受信レートがコンテンツのビットレート，すなわち再生レートを下回らないこと」となる．

現在の TCP/IP プロトコルスタックに基づいたインターネットアーキテクチャでは，あらゆる種類のトラヒックが IP パケットに分割されて転送される．このとき，電子メールの

ように正確にメッセージが到達することが重要であって時間の制約は緩いものもあれば，映像のように，多少の情報の欠落は許されるが遅延に厳しいものもある．図 1.23 に示すように，IP ではそれらの区別はされずにパケットはルータにキューイングされ，到着した順に次のポートへと出力される．ここで，ルータの処理能力を上回る数のパケットが集中的に到着すると，パケットはキューに入れられるため，レシーバへの到着が遅れることになる．さらにトラヒック量が増すと，キューの容量は有限であるため，パケットの廃棄が始まる．これは，レシーバにパケットが届かないことを意味する．

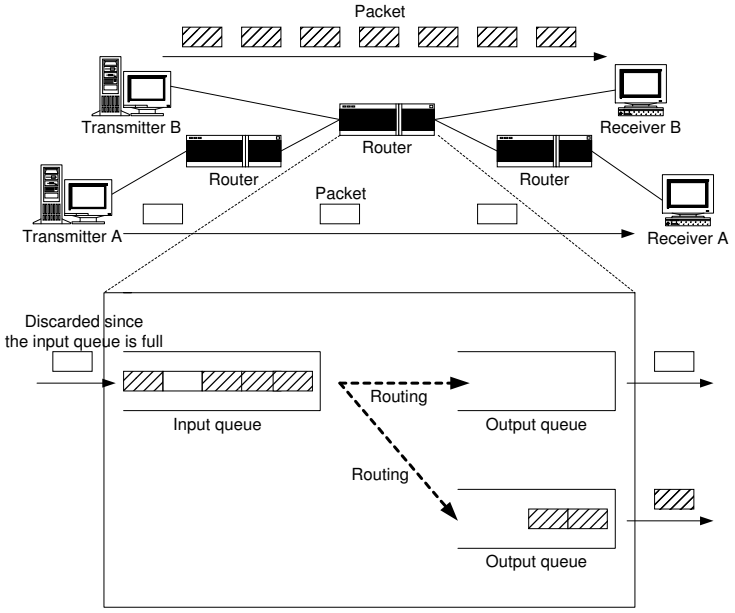


図 1.23　ルータにおけるキューイング

パケット到着の信頼性を保証するのは TCP の役割であり，確認応答（ACK）を送信ホストに返すことでパケットの到着を通知している．もし，所定の時間が経過しても ACK が返ってこないときは，パケットが途中で廃棄されたか，パケット内のデータに誤りがあったものと判断し，そのパケットを再送する [31]．このため，ネットワーク帯域を上回る量のトラヒックが発生すると，再送の回数が増え，遅延が増大する．プログレッシブダウンロードにより映像コンテンツを配信することを考えると，こうした状況では，パケットの到着が再生に間に合わず，再生が中断されてしまう．

UDP を使ったストリーミングの場合には，送信ホストから一定のレートでパケットが送

出されるが，上記のように途中のルータでパケットが廃棄されても再送はされず，受信ホストには届かない．一般に，パケットロス率が 10^{-5} を上回ると，再生される画品質に劣化が見られるとされている [32]．また，廃棄されなくとも，キューイングによる遅延の揺らぎが大きくなると，本来のフレームレートにそった再生ができなくなってしまう．揺らぎを伴ったパケットの遅延を図 1.24 に示す．

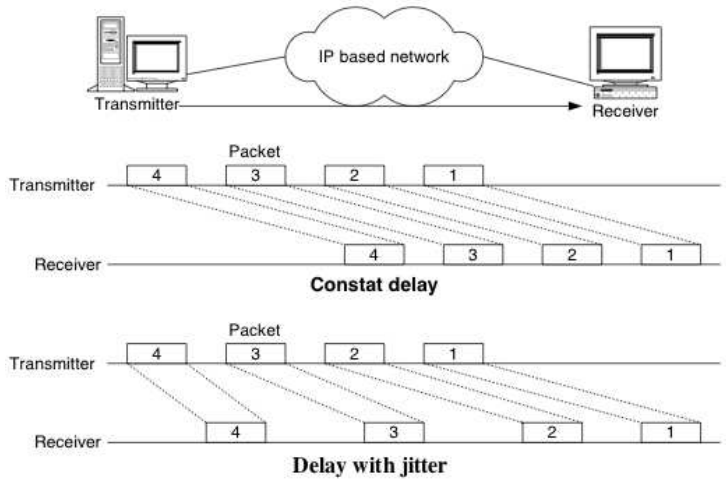


図 1.24　パケットの遅延と遅延揺らぎ

このように，従来の TCP/IP では通信品質が保証されないという問題があり，映像配信を含め，インターネットの利用範囲が広がるにつれて，大きな課題となった．ゆえに，品質保証への対応を考慮した技術が活発に研究され，実装されるようになった．こうした技術のうち，代表的なものを以下に述べる．

1.3.2　ネットワーク層での技術

最初に，ネットワーク層における品質保証技術を整理する．これには，大きく分けて 2 つの考え方がある．一つは，コネクションの確立前にあらかじめ資源を確保しておくというものである．代表的なものとして，IntServ が標準化された [33]．しかし，フロー数やルータ数が増加するほど処理の手間がかかるだけでなく，全てのルータがこのアーキテクチャに対応している必要があるため，実際のインターネット環境で広く普及することはなかった．

もう一つは，フロー単位ではなく，トラヒックをある条件によって複数のクラスに分類し，クラス単位での資源割り当てを行うというものである．この考え方は，DIffServ として標準化された [34]．DiffServ に対応したネットワーク（DS ドメイン）に IP パケットが到着すると，エッジノード（Edge Node）において図 1.25 にあるように，IP パケットがどのクラスに属するかを分類する．次に，IP ヘッダの ToS(Type of Service) フィールド内に，DSCP(DiffServ Code Point) と呼ばれる値をマークし，該当するキューに送る．DS ドメイン内の DiffServ 対応ルータ（DS Router）では，DSCP を参照してパケットを分類し，定められた処理を行う．DiffServ におけるサービスレベルとして，固定帯域を優先的に割り当てるプレミアムサービス（Premium Service），プレミアムサービスとベストエフォートの間の品質を提供するタイアードサービス（Tiered Service），残された資源を用いてパケットを転送するベストエフォートサービス（Best-effort Service）の 3 種類が規定されている．

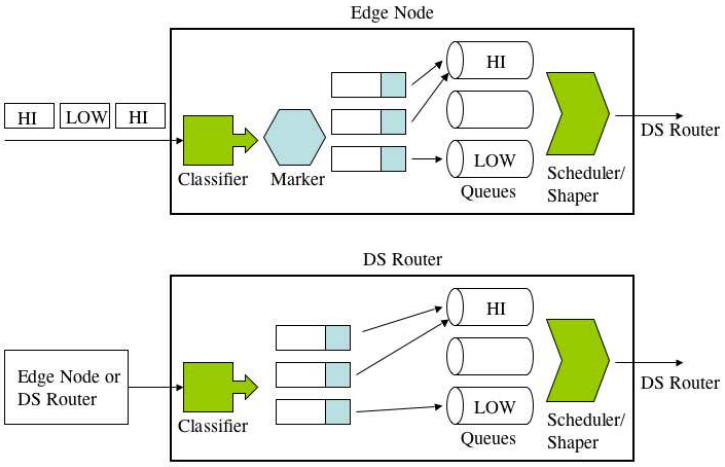


図 1.25　Diffserv のアーキテクチャ

両者の考え方の中間に位置するものとして，近年標準化され，実装が進んでいる MPLS(Multi-Protocol Label Switching) がある [35]．MPLS の概要を図 1.26 に示す．MPLS を実装したルータを LSR(Label Switching Router) と呼ぶ．MPLS ネットワークにパケットが転送されると，その入口となる LSR(Edge LSR) において，ネットワーク層ヘッダとデータリンク層ヘッダとの間にラベルが付与される．その後の LSR(Core LSR)

では，宛先 IP アドレス（Destination IP Address）から経路表を参照する従来の処理に代わって，ラベルを参照してパケットをスイッチングする．最後に，出口となる Edge LSR においてラベルが除去され，通常の IP パケットとして転送される．ラベルとインタフェースの関連付けと隣接 LSR 間でのラベル情報の交換には，LDP(Label Distribution Protocol)，TDP(Tag Distribution Protocol) などが利用される [36].

MPLS の従来の転送処理に対する利点には，次のものがある．

- パケットに下位ネットワークが認識可能なラベルを付与して転送することで，複数のネットワーク層プロトコル・複数のデータリンク層プロトコルに対応できる．
- ラベルへの IP アドレスのマッピングにより，任意の粒度のフロー集約が可能である．
- LSR では，入力パケットの固定長ラベルを読んで出力ラベルを取り出す処理が 1 回のメモリアクセスで完了できるため，パケット転送処理が高速になる．
- 経路を明示的に指定することができ，例えば，複数の経路の使用率を平滑化させることで輻輳を回避しやすくすることができる．

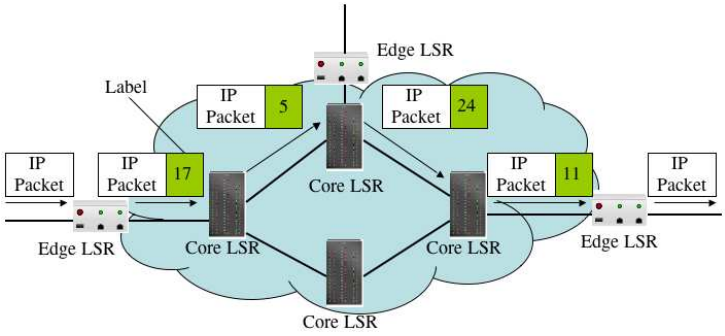


図 1.26　MPLS の概要

キューイングの仕方を変えることで，品質を向上させるという手段もある．その一例として，RED(Rondom Early Ditection)[38] によるキュー管理機構を，図 1.27 に示す．RED では，平均待ちパケット数が下限値未満であればすべてのパケットを受け付け，下限値と上限値との間であれば確立 $p(x)$ でパケットを廃棄する．上限値以上となれば，すべてのパケットを廃棄する．こうすることで，TCP フローの連続的なパケット損失を防ぎ，スループッ

トの向上に役立つ．

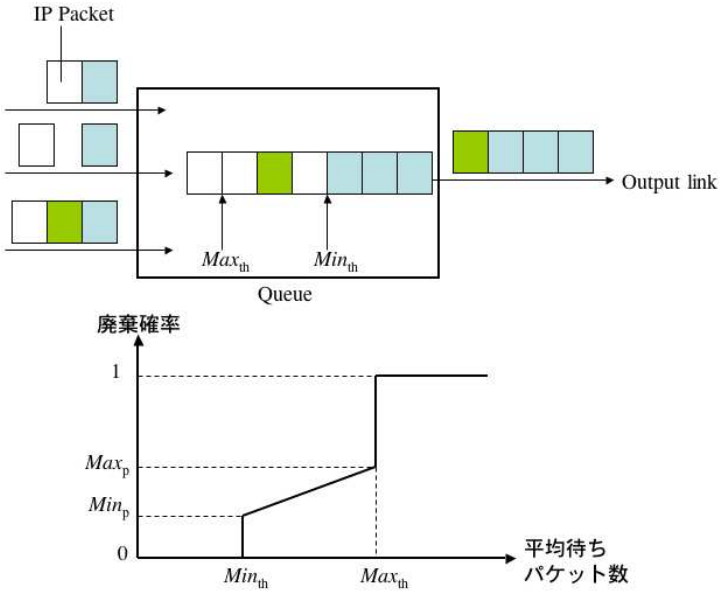


図 1.27　RED によるキューイング

1.3.3　上位層での技術

上位層において通信品質を保証もしくは向上させる技術もある．昨今の CPU やメモリ，ディスクといった装置の処理能力向上は著しいが，あるサーバからコンテンツ配信を実行するとき，同時に提供できるクライアント数には限りがある．これに対処するには，配信サーバを複数台用意し，クライアントからのアクセスを分散させればよい．

最も簡単なアクセス分散方法は，最初のアクセスはサーバ#1 に，2 番目のアクセスはサーバ#2 に,...，N 番目のアクセスはサーバ#N に，N+1 番目のアクセスは再びサーバ#1 に，と機械的に順番にアクセスを割り振ることである．これは，ラウンドロビン（round-robin）と呼ばれる．図 1.28 に示すように，一つのドメイン名に複数の IP アドレスを対応させておき，名前解決の際に順に応答する手法を，DNS ラウンドロビンという [39]．ただし，DNS ラウンドロビンだと，クライアントのキャッシュが残っている間，そのサーバに引き続きアクセスしてしまう，あるサーバの障害時や保守時にもアクセスさせてしまう点が問題となる．

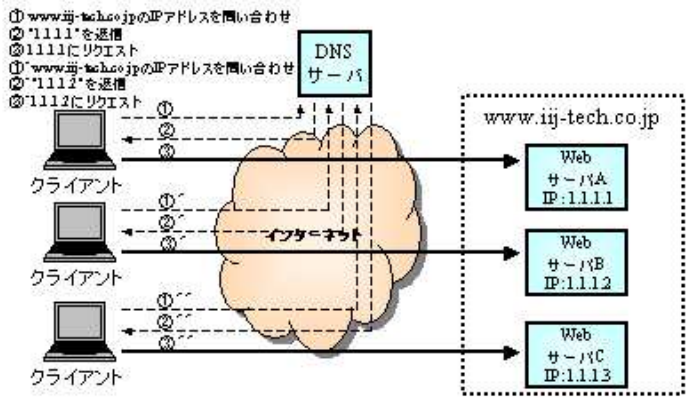


図 1.28　DNS ラウンドロビン

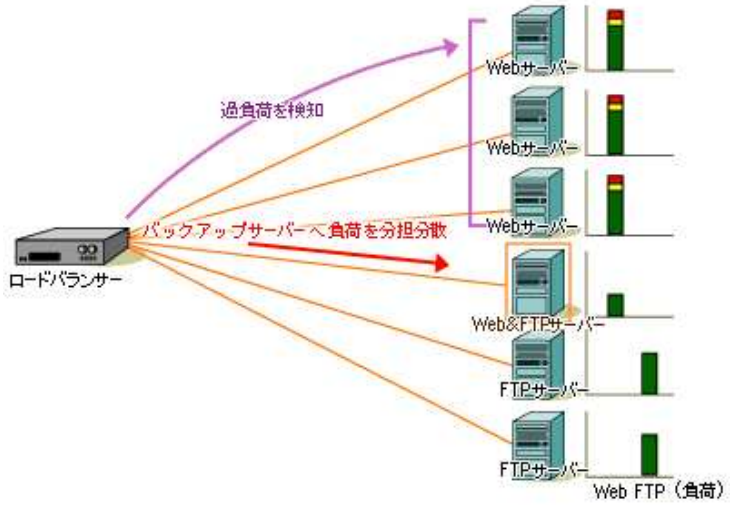


図 1.29　負荷分散装置の導入

そこで，直接サーバにアクセスさせずに，図 1.29 に示すように，サーバ群の手前に負荷分散装置（Load Balancer: LB）を置き，クライアントからのアクセスを一旦この LB にさせておき，負荷分散装置がサーバの CPU 使用率やディスクの使用率といった指標をもとにアクセスを転送すべきサーバを決定する方法がある [40]．LB を導入することにより，負荷を分散させるだけでなく，あるサーバに障害が発生したとしても，システム全体を止めることなくそれを切り離し，復旧するまでの間は他の正常に稼働しているサーバを選択させることができる．

クライアントからのアクセスはLBで終端しているため，図1.30に示すように，IPアドレス・ポート番号が適宜変換される．ポート番号はレイヤ4で用いられる概念であることから，レイヤ4スイッチとも呼ばれる．さらに，HTTPヘッダなどアプリケーション層のヘッダ情報を基にサーバを選択するものも登場しており，これはレイヤ7スイッチと呼ばれている．LBにおける主なサーバ選択基準を，表1.4に示す．

表1.4　LBにおけるサーバ選択基準

基準	仕組み
ラウンドロビン	リクエスト順に振り分ける
重み付けラウンドロビン	サーバの性能で重み付けしたラウンドロビン
接続数	最も接続数の少ないサーバを選択
応答時間	測定用パケットによりサーバの応答時間を測って選択

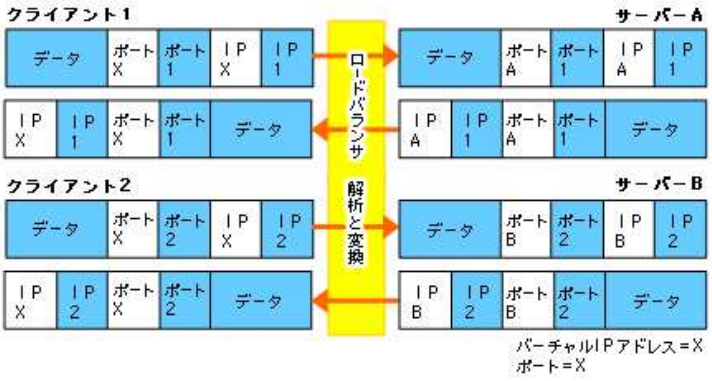


図1.30　負荷分散装置におけるヘッダ変換処理

一方，クライアント側でアクセス負荷を減らそうとしたのが，プロキシサーバの設置によるキャッシングである[41]．既にWWWコンテンツにアクセスするときには一般的となっているが，その概念を図1.31に示す．まず，クライアント側のネットワーク内に，プロキシサーバと呼ばれるゲートウェイを設置する．プロキシ（proxy）とは代理という意味であり，クライアントが直接目的のサーバにアクセスする代わりに，プロキシサーバにアクセスしてもらい，受信したコンテンツをクライアントに転送してもらう．これにより，過去にアクセ

スされたコンテンツは，プロキシサーバ内にキャッシュとして保存されることになり，すぐにコンテンツを取得できる．ゆえに，外部ネットワークに出てゆくトラヒック量を減らし，帯域幅の節約が可能となる．

もちろん，プロキシサーバのストレージ容量は有限であるので，何らかの判断によりキャッシュするコンテンツ・消去するコンテンツを決めなければならない．コンピュータ内部のキャッシュメモリと同様に，記録した日時の古い順に消去する方法や，直前のアクセス間隔が長いものから順に消去する方法が一般的である．WWW のプロキシサーバの実装として広く利用されているものに，Squid[42] がある．

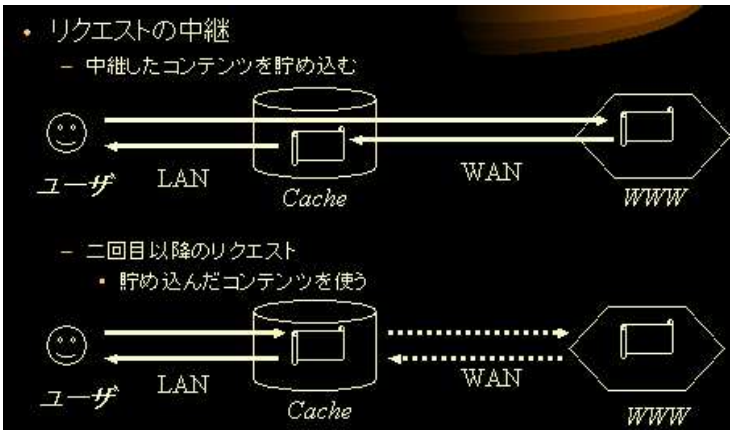


図 1.31　プロキシサーバによるキャッシング

1.4　CDN による負荷分散

1.4.1　CDN のアーキテクチャ

上記の負荷分散技術により，迅速なコンテンツの配信が可能となるが，近年，ネットワークの高速化に合わせて，配信されるコンテンツのファイルサイズも増える傾向にある．例えば，ソフトウェアパッケージであれば数 MB から数 10MB といったサイズが一般的であり，無償で提供される OS の ISO イメージファイルでは 640MB になる．今後，ますます多くのユーザが高速ネットワークからアクセスすることを考えると，アクセス網より上位のネッ

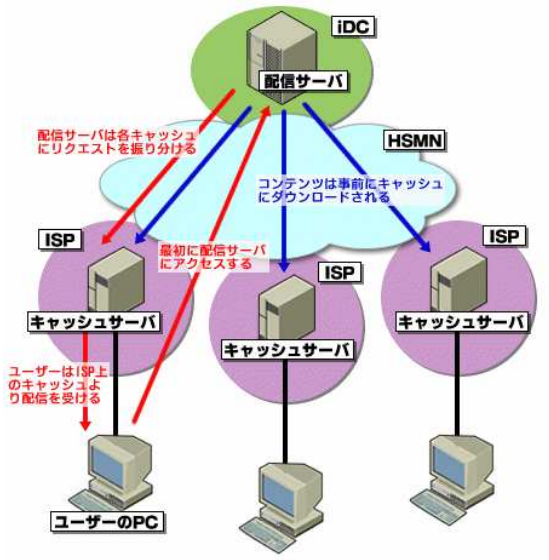


図 1.32　CDN のアーキテクチャ（CDN JAPAN の場合）

トワークに負荷が集中することが予想される．そうなると，従来の負荷分散技術だけでは，ネットワークの負荷まで分散させることは難しい．

そのため，新たなコンテンツ配信網（Content Delivery Networks: CDN）として，図 1.32 に示すように，配信サーバの代理サーバ（Surrogate）をネットワークの各地に広域に配置し，クライアントからのリクエストを適切なサーバに誘導する仕組みが研究され始めた [43]．これは，従来の負荷分散技術を，より広域のネットワークに対して適用したものと考えられる．CDN は，以下に挙げる複数の機能から成り立っている．

- コンテンツ配布

  オリジナルのコンテンツファイルを，代理サーバに配布する．実際には，すべてのサーバにすべてのコンテンツを複製するのはコスト面で問題があるため，どのコンテンツをどのサーバに配布すればより良い負荷分散が実現できるかという配布戦略が重要となる．

- リクエスト誘導

  クライアントからのコンテンツ配信要求をある基準により選択したサーバに転送し，配信を実行する．どういった基準でリクエストを誘導するかが課題となる．

- 認証・課金

  コンテンツを要求したクライアントが正当なユーザであるか認証し，条件に応じて定められた課金処理を行う．

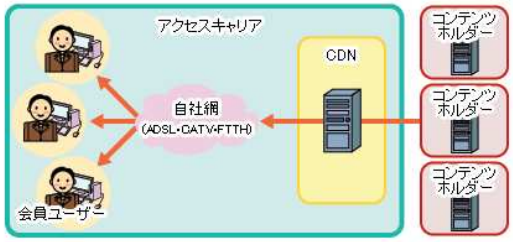
- DRM(Digital Rights Management)

  最近，音楽や映画のディジタルコンテンツが違法にコピーされ，流通することが大きな社会問題となっている．そうした状態では，コンテンツを提供してくれる制作者が増えない．ゆえに，電子透かし [44] など，コンテンツの不正使用を防止する技術を導入することが求められている．こうしたディジタルコンテンツに関わる知的財産権を管理する仕組みは DRM と呼ばれている．

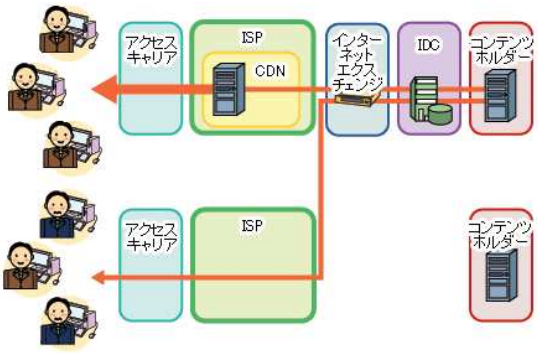
現在提供されている CDN によるコンテンツ配信サービスのモデルは，アクセスキャリアモデル，ISP モデル，CDN 事業者モデルに大きく分類される．これらの形態を，それぞれ図 1.33a), 図 1.33b), 図 1.33c) に示す．アクセスキャリアモデルは，アクセス網を提供しているキャリアが自社網内に CDN を構築し，会員向けに有料コンテンツを提供するモデル，ISP モデルは，ISP（Inernet Service Provider）が自社ネットワーク内に CDN を構築し，自社ユーザに対してコンテンツを配信するモデルで，CDN 事業者モデルは，CDN 事業者が独自のインフラを構築し，アクセスキャリアや ISP に対してコンテンツを配信するモデルである．

1.4.2 リクエスト誘導の種類

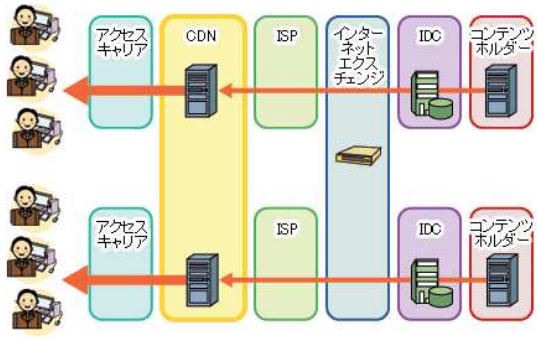
現在，CDN の研究は，コンテンツ配布戦略とリクエスト誘導戦略の二つが中心となっている．前者の研究例には，[45]-[47] がある．ここでは，後者のリクエスト誘導に関する研究例を紹介する．最も単純な方法として，クライアントにサーバの一覧を見せて近くのサーバを手動で選択してもらうというものがある．これは，近くのサーバであれば RTT(Round-Trip-Time) が小さいため，TCP のスループットが上がるという前提に立っている．Linux カーネルの配布サイト [48] や SourceForge.net[49] などは，世界各地にミラーサーバを配置



a) アクセスキャリアモデル



b) ISP モデル



c) CDN 事業者モデル

図 1.33　CDN によるコンテンツ配信サービスモデル

し，ユーザにどのサーバから配信してもらうか選択させている．例として，Sourceforge.net に公開されているファイルをダウンロードする際に表示されるサーバ選択の様子を，図 1.34 に示す．

これに対して，システムの方で自動的にサーバを選択してリクエストを誘導する手法として，DNS Balance[50]，TENBIN[51] がある．両者共に Ring Server Project のサイトで運用されている．Ring Server では，図 1.35 に示すグラフを提示し，ユーザにサーバを選択し

てもらっていたが，これを自動化した DNS サーバが DNS Balance である．TENBIN は，多様なサーバ選択ポリジーを利用可能な DNS サーバの実装例である．現在の Ring Server サイトでは，サーバ負荷によるミラーサーバの選択に前者を，距離によるミラーサーバの選択に後者を使用している．また，TENBIN と経路情報を収集するサーバとを組み合わせ，日食などの天体イベントを広域に配信することにも成功している [52].

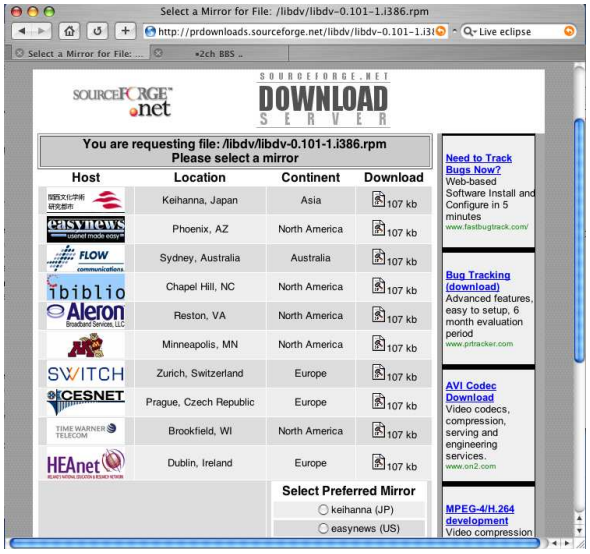


図 1.34　手動によるサーバ選択の例

図 1.35　サーバ負荷の表示

一方で，クライアント側が何らかの基準によりサーバを選択する方法も提案されており，これは次の三種類に大別される．

1. traceroute などを用いてサーバまでのホップ数を測り，経路ホップ数が最小のサーバを選択する．
2. ping などによってサーバまでの往復遅延時間，物理容量などの品質を測り，測定品質パラメータ群による評価値が最良のサーバを選択する．
3. サーバ側が過去の品質情報を保持し，それを参照して最良のサーバを選択する．

1 の方法は，現在の IP 網では品質が変化しうることを考慮すると現実的でない．2 の方法は，測定した瞬間の状態で決定するため，その後品質が悪化することがあり得る．3 の方法は，ネットワーク設備の変更や異常トラヒックの発生といった状態変化に対応困難である．最近は，まず 3 の方法により 2,3 台のサーバに絞りこみ，その後 2 の方法によって最良の結果を得た 1 台を選択する方式が研究されている [53]-[55]．

1.5　まとめ

本章では，まず，インターネット，特にアクセス網がここ数年の間に急速に広帯域化したこと，映像のディジタル化とその情報量を削減する符号化技術が進み，コンピュータの処理能力の向上もあって映像コンテンツの配信が増えてきたことを述べた．次に，映像配信の種類を整理し，従来ベストエフォートであった IP 網においても円滑に映像が配信できるよう，サービス品質を保証するための技術が登場し，主な技術を紹介した．その中から，より柔軟で広域に渡るな負荷分散技術として CDN の概念が現れたことを導き，その成功の鍵を握るリクエスト誘導技術について，DNS と連携し，クライアントからの名前解決のときに適切なサーバの IP アドレスを返す方式と，クライアント側がある基準に従ってサーバを選択する方式に分類されることを述べた．

第 2 章

並列転送による再生品質維持制御

この章では，まず，CDN において映像コンテンツを配信するときの問題点として，既存のサーバ選択方式では，コンテンツの再生品質を維持できない場合があることを明らかにする．これに対し本研究では，複数のサーバから並列にクライアントに対してコンテンツデータを転送することで，配信時の再生品質を維持することを提案する．提案方式は二種類ある．一つは，コンテンツがミラーリングされている状態において，各サーバからの転送量を転送速度に比例配分させる方式である．もう一つは，コンテンツを決まった単位量で分割してサーバ群に分散配布させておく（以後ストライピングと呼ぶ）することで，各サーバからクライアントまでの経路に要求する帯域を抑える方式である．後半では，DV ファイルを対象とした配信実験について報告し，提案方式の有効性を示す．

2.1 研究の動機

2.1.1 CDN における映像配信時の問題点

前章において，CDN におけるリクエスト誘導方式を述べたが，ここで，ビットレート B_f[Mbit/s] で再生時間が T_f[sec] である映像コンテンツが，図 2.1 に示すように配置されている場合を考える．

この環境で，クライアント C がこのコンテンツの視聴を要求したとすると，既存の方式では，手動でのサーバ選択，ラウンドロビンによるサーバ選択，RTT・サーバ負荷・過去の QoS に関する統計などを指標としたサーバ選択といったリクエスト誘導が行われる．その後，選択されたサーバ S_i から C に対して配信が実行される．

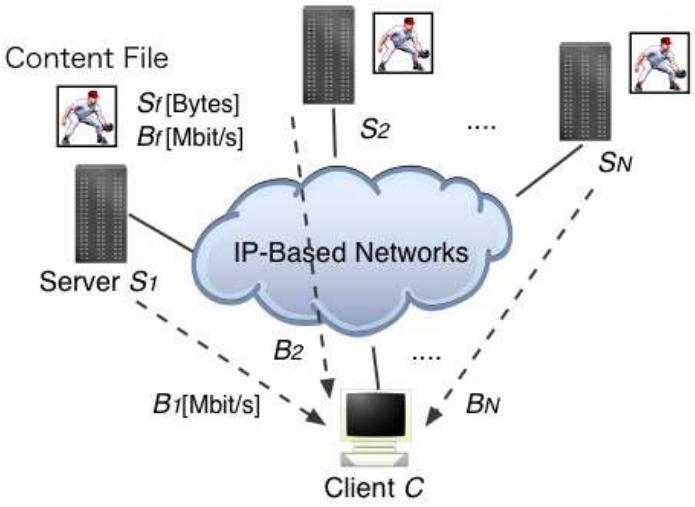


図 2.1　CDN における映像コンテンツ配信

ここで，コンテンツのファイルサイズ S_f[Bytes] は，次式により求められる．

$$S_f = \frac{B_f T_f}{8} \tag{2.1}$$

選択されたサーバ S_i から C に至る経路の帯域幅が B_i[Mbit/s] であったとすると，このコンテンツをダウンロードするのに要する時間 T_{dl}[sec] は，

$$T_{dl} = \frac{8S_f}{B_i} \times 10^{-6} \tag{2.2}$$

となる．T_{dl} が短く，かつ C に S_f[Bytes] の空き容量があれば，ダウンロードによる配信が簡単といえる．

今後，ネットワークの広帯域化が進むにつれて，配信されるコンテンツもより高品質，長時間のものが一般化することが予測される．そうすると S_f が大きくなり，日常的に映像コンテンツを配信してもらおうとすると，クライアントのストレージ容量を圧迫してしまう．また，クライアントもコンテンツ全体を視聴したいとは限らない．そのため，現在ではオンデマンド配信はプログレッシブダウンロードで，ライブ配信はストリーミングで行うのが一般的となっている．この場合，即時に再生が始まりかつ停止せずに最後まで視聴できるための条件は，

$$B_i \geq B_f \tag{2.3}$$

となる．既存のサーバ選択方式ではこの条件が成り立つかどうか判断することが難しい．例えば，TCP のスループットは RTT に反比例するが，では RTT がいくらのときにスループットは何 Mbit/s 得られるかという関連付けを経路ごとに行うのは現実的でない．

さらに，B_i は時間と共に変動するため，仮に選択されたサーバから配信を開始したときに式 (2.3) が成り立っていたとしても，これが配信の最後まで成り立つことは保証されない．もし，途中で式 (2.3) が成り立たなくなれば，再生が中断され，B_i が小さくなるほど再生が再開されるまで待たされる時間が長くなってしまう．UDP によってストリーミングしていたとすると，パケット到着の遅延揺らぎの増大，パケットの損失率の上昇が起こり，画品質も劣化する．

2.1.2 従来の対応

現在インターネット上で公開されている映像コンテンツは，図 2.2 に示されるように，あらかじめ数種類のビットレートで符号化しておき，ユーザにどのレートで配信するか選択してもらうことが多い．これも一つの手段だが，経路内のリンクあるいはノードが高負荷になるとやはり品質が低下する．

映像配信時の帯域幅不足が起きたとき，次のいずれかの方法で帯域幅に合わせたビットレートで送信し，帯域幅の変動に適応する技術が研究されている．

1. 時間解像度，すなわちフレームレートを下げる．例えば，30fps で再生されるものを 15fps に下げればビットレートは 1/2 になる．ただし，下げすぎると滑らかさが失われる．
2. 空間解像度，すなわち 1 フレームあたりの情報量を減らす．量子化係数を調整して 1 画素あたりのビット数を下げたり，サブサンプリングにより送信する画素数を減らすことが該当する．

図 2.2　ビットレートの選択画面の例

ライブ配信の場合，RTP/UDP を用いるのが一般的だが，図 2.3 に示すように，RTCP(RTP Cotnrol Protocol) を利用して，送受信ホスト間でパケット損失率や遅延揺らぎといった QoS パラメータを定期的に交換し合い，送信ホストが動的にビットレートを制御できるようになっている．RTCP の仕様は，RTP と合わせて [30] に定められている．RTCP のフォーマットには，図 2.4 および図 2.5 に示すように，送信ホストから受信ホストに対して送られる RTCP-SR(Sender Report) と，受信ホストから送信ホストに対して送られる RTCP-RR(Receiver Report) の二種類がある．

オンデマンド配信の場合，単に符号化・蓄積しただけだと帯域幅の不足に適応できない．そのため，図 2.6 に示すように符号化データを階層構造とし，帯域幅に合わせて送信する階層数を調整する階層符号化や，図 2.7 に示すように，送信ホスト内部あるいは経路内の中継ノードにおいてビットレートを変換するトランスコーディングと行った技術が研究されている．

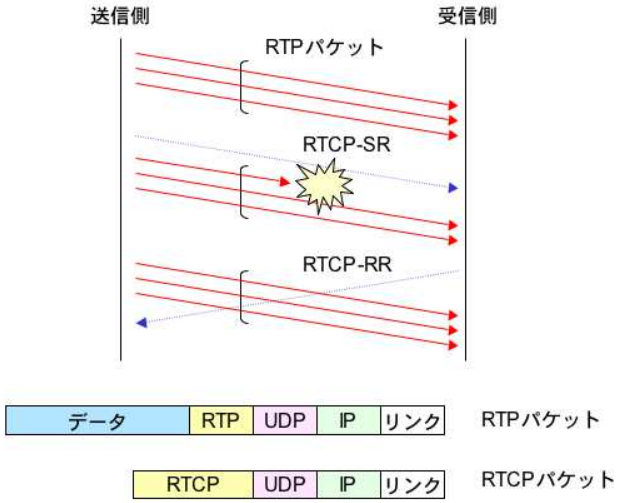


図 2.3　RTCP の利用

図 2.4　RTCP-SR のフォーマット

2.1.3　着眼点

このように，映像コンテンツを配信する場合，帯域幅の変動に適応する必要があるが，いずれの方法をとっても結局は再生品質を下げることになる．これは，CDN においても既存のサーバ選択方式を採用する限り変わらない．しかし，地理的に分散した複数台のサーバにコンテンツが配置されていることを利用して，再生品質を下げる前に本来の品質を維持する

図 2.5　RTCP-RR のフォーマット

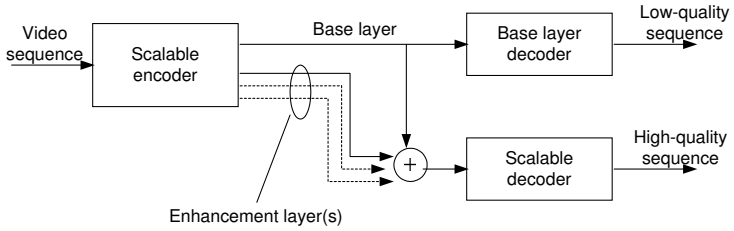


図 2.6　階層符号化

ことができないか，検討を進めた．

その結果，コンテンツがミラーリングされている N 台のサーバから分担してデータを配信してもらうことで，式 (2.3) の条件がすべての経路で成り立たないとしても，別の条件が

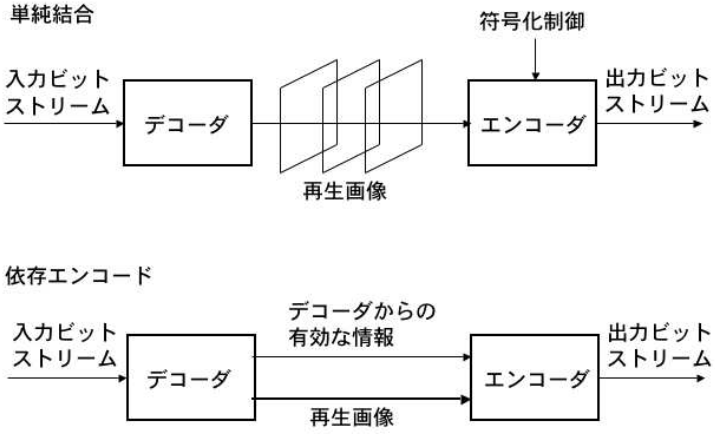


図 2.7　トランスコーディング

成立する限り再生品質を維持できる配信方式を提案する．提案方式は二種類ある．一つは，コンテンツがミラーリングされている状態において，各サーバからの転送量を転送速度に比例配分させる方式である．もう一つは，コンテンツを決まった単位量でサーバ群にストライピングすることで各サーバからクライアントまでの経路に要求する帯域を抑える方式である．以下，提案方式の概念と実装にあたっての仕様を説明する．

2.2 ミラーリングされたコンテンツの並列転送

2.2.1 概念

いま，図 2.8 に示すように，ある映像コンテンツがファイルとしてサーバ $S_i(i = 1, 2, ..., N)$ にミラーリングされているものとする．このコンテンツのファイルサイズを S_f[Bytes]，再生時間を T_f[sec]，ビットレートを B_f[Mbit/s] と仮定し，クライアント C からの要求に応じて配信する．ここで，C は S_i に所望のコンテンツが存在することを既に知っているものとする．

S_i から C に至る経路の帯域幅を B_i[Mbit/s] とし，C のネットワーク受信速度ならびに受信したデータの再生処理速度は，B_i の総和以上あるものとする．このとき，S_i から C に転送してもらうコンテンツデータの量 s_i[Bytes] を，次のように定める．

$$s_i = \frac{B_i}{\sum_{i=1}^{N} B_i} \times S_f \tag{2.4}$$

すなわち，1 台のサーバで S_f[Bytes] すべてを転送する代わりに N 台で分担し，各サーバが負う送信量は，転送速度に比例配分させる．このとき，S_i から C へのコンテンツデータ転送時間 T_{di}[sec] は，

$$T_{di} = \frac{s_i}{B_i} = \frac{\frac{B_i}{\sum_{i=1}^{N} B_i} \times S_f}{B_i} = \frac{S_f}{\sum_{i=1}^{N} B_i} \tag{2.5}$$

となり，実効的な転送速度は B_i の総和となる．ゆえに，次式が成立すれば，コンテンツをストリーミングおよびプログレッシブダウンロードしたときに，品質が劣化せずに，再生を

続けることができる．

$$\sum_{i=1}^{N} B_i \geq B_f \tag{2.6}$$

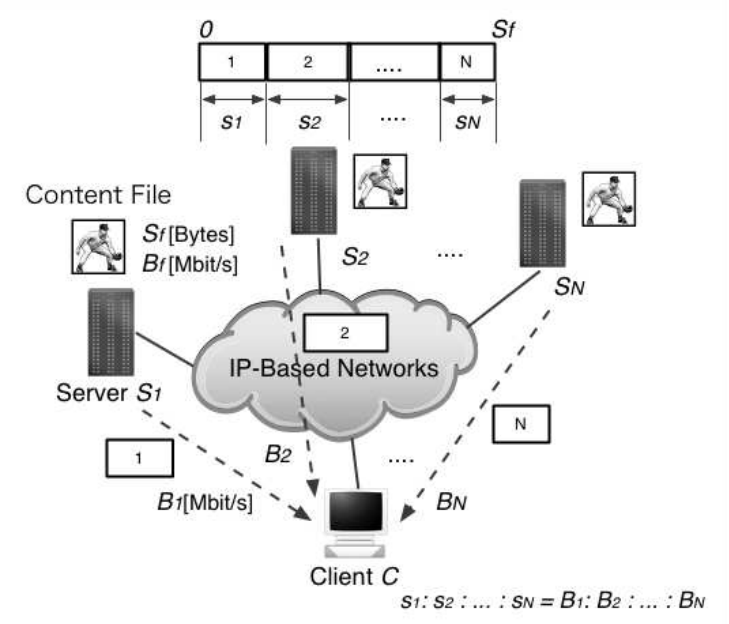


図 2.8　ミラーリングされたコンテンツの並列転送

2.2.2　仕様

実際のネッワークでは，転送速度 B_i は時間と共に変化するため，これを $b_i(t)$ と表す．仮に式 (2.4) に示したように，配信開始時に S_i からの送信量を割り当てたとしても，時間の経過と共に $b_i(t)$ の大小関係が変わることで，かえって転送速度が低下するおそれがある．

ストリーミングにより配信する場合，送信レート $b_{snd}(t)$ で送出されたストリームの受信レートが $b_{rcv}(t)$ だったとすると，図 2.9 に示すように，ネットワークの帯域幅 $b_i(t)$ の状態によって，次の関係が成り立つ．

$$b_{rcv}(t) = b_{snd}(t) \qquad if b_i(t) \geq b_{snd}(t) \tag{2.7}$$

$$b_{rcv}(t) < b_{snd}(t) \quad if \quad b_i(t) < b_{snd}(t) \tag{2.8}$$

よって，$b_{rcv}(t) = b_{snd}(t)$ であれば $b_{snd}(t)$ を上げ，$b_{rcv}(t)$<$b_{snd}(t)$ であれば $b_{snd}(t)$ を下げることで $b_i(t)$ を推定することはできる．`pathload` は，この原理により，エンドホスト間の利用可能帯域を測定している．しかし，これにより $b_i(t)$ を求めるにはトラヒックを余分に発生させるだけでなく，推定に 10 秒から 30 秒かかることがわかっているため，実用的でない．さらに，複数のホストペアによってボトルネックを共有していたとすると，推定値どおりのレートでストリームを送信すると輻輳してしまう，既存 TCP トラヒックのスループットを下げてしまうといった問題もある．

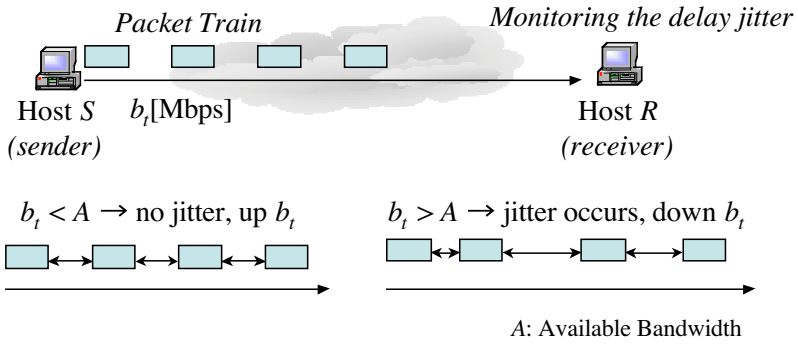


図 2.9　帯域幅と受信レートの関係

これに対して，プログレッシブダウンロードにより配信することを考えると，トランスポート層に TCP を用いるため，輻輳を回避するためのフロー制御が適用された状態で $b_{rcv}(t)$ が得られるため，これを $b_i(t)$ とみなすことができる．ただし，そのままダウンロードを開始すると，$s_i(t)$ を修正することができない．

ゆえに，図 2.10 にあるように，S_f を大きさ U[Bytes] のブロックが M 個集まったものとみなし，クライアント C から順次ブロックを要求し，S_i がそれに応じて転送することを考える．転送プロトコルには，HTTP（トランスポート層は TCP）を使うものとする．

このとき，ブロック 1 個分のコンテンツデータ受信を m 回の転送で，$u = U/m$[Bytes] ごとに行う．そして，k 回目の転送時の S_i からの送信量 s_{ik}[Bytes] を，式 (2.10) および式 (2.10) により求める．

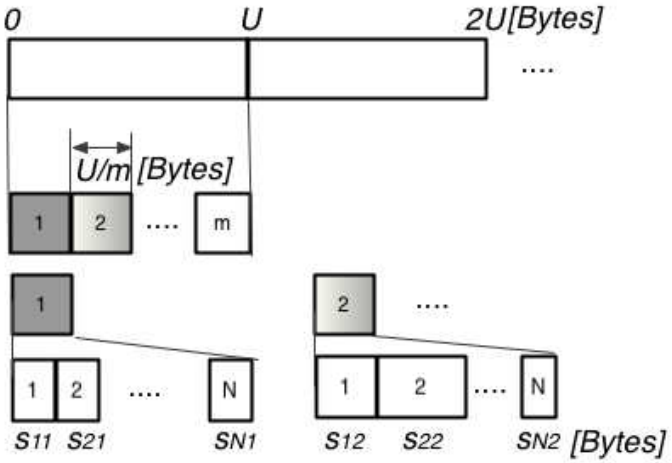


図 2.10　ミラーリングされたコンテンツのブロック単位での並列転送

$$s_{i1} = \frac{u}{N} \quad (k = 1) \tag{2.9}$$

$$s_{ik} = \frac{b_{ik-1}}{\sum_{i=1}^{N} b_{ik-1}} \times u \quad (2 \leq k \leq m) \tag{2.10}$$

ここで，b_{ik} は，S_i からの k 回目の転送時における転送速度を指す．b_{ik} は，s_{ik} を転送に要した時間で割ることで得られる．これにより，$b_i(t)$ の変動に追従しながら，b_{ik} の総和が B_f 以上である限り，再生品質が維持されることを目指す．

2.3　ストライピングされたコンテンツの並列転送

2.3.1　概念

前節で述べた転送方式は，コンテンツが N 台すべてにミラーリングされていることを利用したものであり，仮にすべてのサーバからの転送速度が B_f を下回ったとしても，合計で B_f 以上あれば本来の品質を保つことが可能というものであった．ただ，高品質で再生時間が長時間にわたるコンテンツの場合，昨今ストレージの低価格化が進んでいるとはいえ，ミラーリングするコストもそれなりにかかる．

そこでもう一つの方法として，図 2.11 に示すように，コンテンツファイルを U_s[Bytes]

ごとに S_i に順次振り分ける，つまりストライピングすることを提案する．これは，RAID レベル 0 と同様のデータ格納方法であるが，各ディスクがネットワーク上に分散している場合にも適用したものといえる．この場合，S_i に必要なストレージ容量は，S_f/N[Bytes] とミラーリング時の $1/N$ で良いことになる．

クライアントへの配信は，S_1 から S_N すべてが自身が持っている U_s[Bytes] のブロックをクライアント C に送信する，C は受信したデータを本来の順序に整列する，次のブロックを要求する，といった手順を繰り返すことにより行われる．この場合，クライアント C において，ビットレート B_f を維持してコンテンツが再生されるための条件は，

$$b_i(t) \geq \frac{B_f}{N} \quad (i = 1, 2, ..., N) \tag{2.11}$$

となる．ただし，$b_i(t)$ は，任意の時刻 t における S_i から C へのコンテンツデータ転送速度である．

$b_i(t)$ のうちどれか一つが B_f/N を下回ると，たとえ合計で B_f 以上あったとしても，再生品質は維持できないことになる．よって，ネットワーク品質が低下したときの耐性はミラーリングによる並列転送方式よりも劣るが，その分コンテンツ配布のコストを下げることができる．それでも，B_f といった速度は出ないが，B_f/N 以上であれば定常的に得られるといった場合には有効な手段となる．

2.3.2 仕様

ここでも，ミラーリングによる並列転送と同様に，コンテンツをブロックごとに取得・再生することにする．コンテンツがストライピングされた場合，U[Bytes] のブロック単位で受信・再生するには，ストライピングの単位 U_s[Bytes] ごとに S_i から受信して整列するという手順を，U[Bytes] 揃うまで繰り返せば良い．

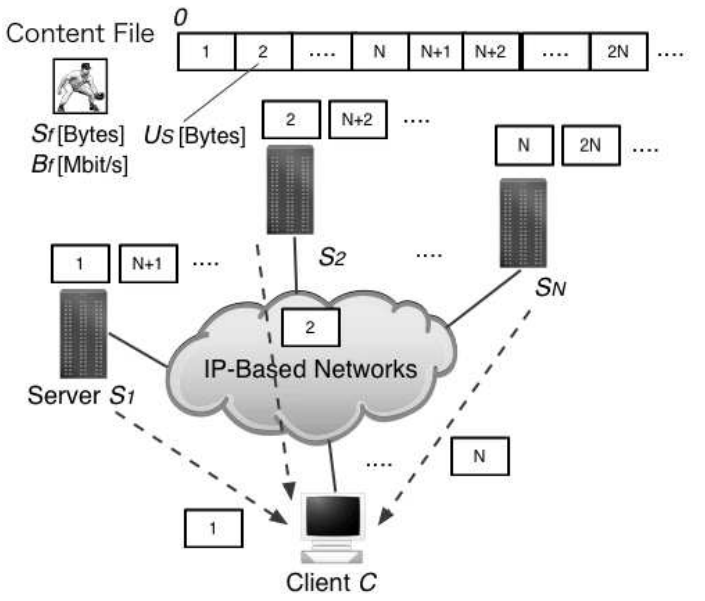


図 2.11　ストライピングされたコンテンツの並列転送

2.4　プロキシサーバへの適用

ここまで，1 台のクライアントがあるコンテンツを要求するというモデルで議論を進めてきたが，図 2.12 に描かれているように，同一の自律システム（AS）内に複数のクライアント $C_1, C_2, ..., C_M$ がある場合を考える．

単純に全てのクライアントが独立にコンテンツの配信を要求すると，サーバ S_i がある AS と C_i がある AS との間を通るトラヒックは M 倍になるため，ネットワークおよびサーバにかかる負荷が問題となる．このため，プロキシサーバを設置し，受信するコンテンツデータをキャッシングすることが有効な対策となる．

プロキシサーバを利用する場合でも，次のようにして，提案方式を適用することができる．

1. C_i から，プロキシサーバ S_p にコンテンツがキャッシングされているか照会する．
2. キャッシングされていれば，それを受信・再生する．もし存在しなければ，S_p が S_i からコンテンツデータを提案方式により受信する．
3. 所定の量を受信したら C_i に転送し，C_i は再生を開始する．S_p は次のデータの受信を

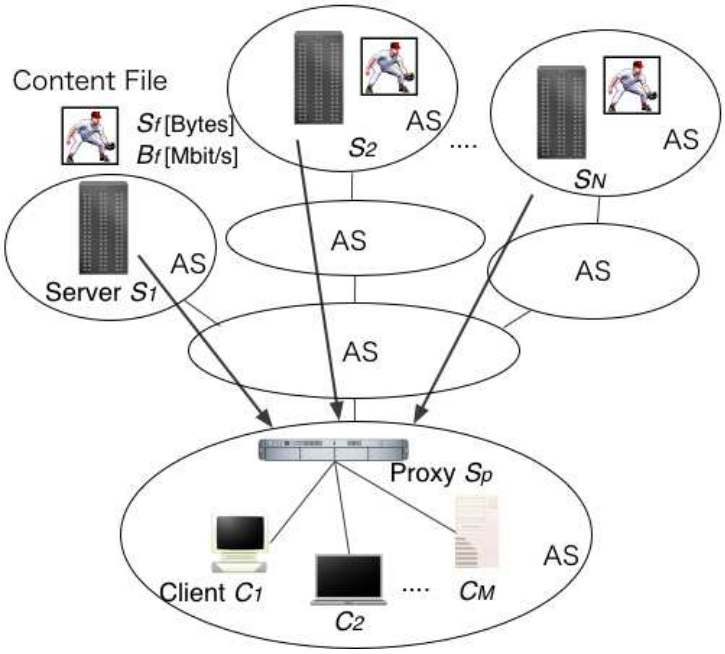


図 2.12　プロキシサーバが行う並列転送

引き続き行う．

こうすることで，同じコンテンツを要求するクライアントが 2 台以上あった場合にも，S_i から S_p に至る経路を通るトラヒック量はコンテンツ一個分ですむことなる．

2.5　DV 配信実験

提案方式の有効性を検証するため，前述の仕様に沿ったプレイヤーを実装し，DV ファイルをミラーリングして並列転送した場合と，ストライピングして並列転送した場合の再生品質を評価した．実験用のネットワークとして，LAN 上の特定マシンに擬似的にネットワーク品質を設定したものを用いた．

2.5.1　実験環境

構築した実験環境を図 2.13 に示す．配信対象として，再生時間 T_f=8 分間の DV ファイルを用意した．NTSC 形式の場合，DV データの 1 フレームあたりのサイズは 120,000Bytes

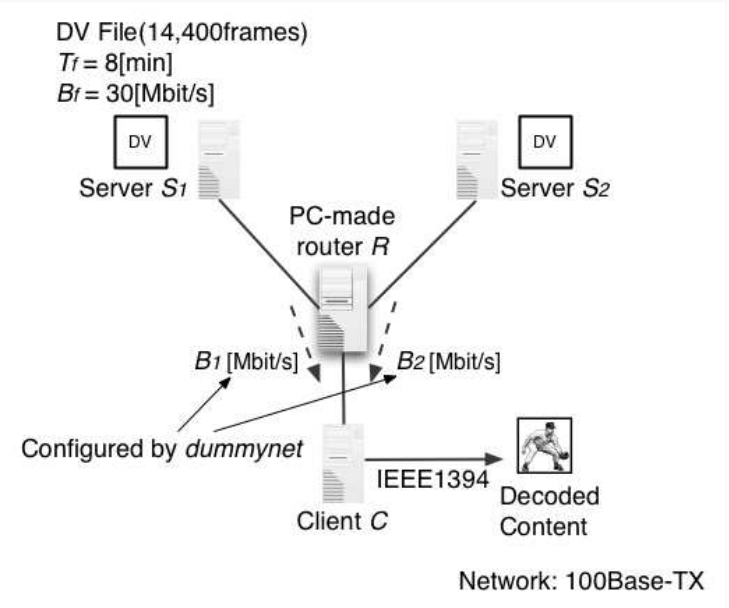


図 2.13 実験環境

となる [18] ため，ファイルサイズ S_f は，S_f = 120,000Bytes/frame × 30frames/sec × 480sec = 1,728,000,000Bytes ≃ 1.6GB[*1] となる．

サーバは 2 台，PC-AT 互換機に Linux 2.4.22 を導入したものに HTTP サーバとして `apache-1.3.29`[*2] を稼働させる．クライアントには，PC-AT 互換機に FreeBSD 5.1-RELEASE を導入したものを選んだ．コンテンツデータの受信には `libcurl-7.10.8`[*3] を利用し，再生には，IEEE1394 インタフェースを介した DV データの送受信が行える `fwcontrol`[*4] を利用した．

サーバ，S_1, S_2 およびクライアント C は，100Base-TX によってもう 1 台の PC（R）と接続される．この R がルータとして動作し，パケットが転送される．R には dummynet[56] を導入し，擬似的にネットワーク品質を設定できるようにした．

[*1] $1GB = 2^{30} Bytes$ として計算
[*2] http://www.apache.or.jp/
[*3] http://curl.haxx.se/
[*4] /usr/src/usr.sbin/fwcontrol にソースコードがある．

2.5.2 コンテンツの再生手順

コンテンツの再生手順を図 2.14 に示す．始めに，S_f[Bytes] を U[Bytes] のブロックに分けて考え，そのうち最初の 2 個のブロックを受信し，ファイルとして保存する．2 個目のブロックを受信し終わったら，最初のブロックの再生（IEEE1394 インタフェースへの出力）を開始する．そして最初のブロックを再生し終わったら，次のブロックの再生に移ると共に，3 個目のブロックの受信を開始する．以下，これを最後のブロックまで繰り返す．今回の実験では，U=36,000,000Bytes（300 フレーム・再生時間 10 秒に相当）とした．

転送速度が足りずに，次に再生すべきフレームがまだ受信されていないときにアンダーフローと判定し，再生を中断する．そして，そのブロックの残りデータと次のブロック全体を受信し終わってから再生を再開する．よって，再生が中断されることを品質の劣化とする．空間解像度については，TCP をトランスポートプロトコルとしているため，DV フォーマットへの符号化・復号過程で生じる以外の劣化はないものとする．

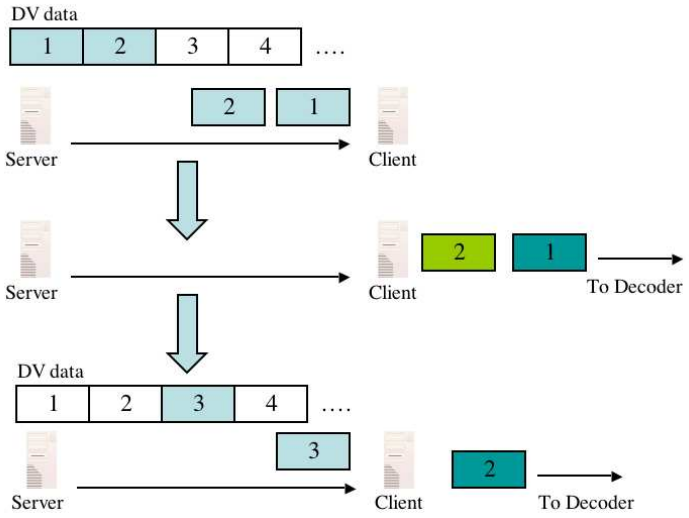


図 2.14　再生手順

2.5.3 実験シナリオ

図 2.13 の構成において，(1)S_1 からすべて転送する，(2)S_1, S_2 両方にファイルを置き，並列転送する，(3)S_1, S_2 にストライピングし，並列転送する場合について再生品質を比較する．(2) においては，ブロックをさらに 8 個の小ブロックに分けて，式 (2.10), 式 (2.10) に従って転送する．(3) では，ストライピングの単位を U_s=3,600,000Bytes（30 フレーム・再生時間 1 秒に相当）とする．

配信時には，S_1, S_2 から C に至る経路の帯域幅 B_1, B_2 を，dummynet により，(I) 一定の値に設定する，(II) 決められた周期で一様に変動する，という二通りの制御を実行した．DV のビットレート B_f は 30Mbit/s 弱[*5]となることから，前者の場合は，B_1, B_2 を 10Mbit/s から 40Mbit/s の間に設定し，後者においては，B_1, B_2 は 10Mbit/s から 40Mbit/s の範囲で一様に 分布させ，変動の周期 T_r を 5, 10, 20, 30 秒とした．

再生品質については，次の評価尺度を用いる．まず，アンダーフローにより再生が中断した回数を「再生中断回数（Number of Stalling: NoS）」とする．これは，0 が理想的となる．次に，中断してから再生が再開されるまでに要した時間を「再生再開待ち時間（Time to Resumption: TR）」とする．これも，再生中断回数が 0 の場合は 0 となる．続いて，これらの値から,「平均再生中断間隔（Mean Time Between Stalling: MTBS）」と「平均再生再開待ち時間（Mean Time To Resumption: MTTR）」および「連続再生率（Consective Playback Ratio: CPR）」を求める．これらは，次のように定義される．

- 平均再生中断間隔（MTBS）[sec]

  中断せずに連続して再生される平均時間．コンテンツの再生時間を T_f としたとき，MTBS = T_f / (NoS+1) となる．
- 平均再生再開待ち時間（MTTR）[sec]

  再生が中断されてから再開されるまでに要する平均時間．MTTR = $\sum_{i=1}^{NoS} TR_i$/NoS となる．

[*5] 120000Bytes/sec × 30frames/sec = 3,600,000Bytes/sec = 28,800,000bit/s = 28.8Mbit/s となる．

表 2.1 帯域幅一定時の再生品質（S_1 からのみ転送，MTBS/MTTR の単位は sec）

B_1[Mbit/s]	NoS	MTBS	MTTR	CPR
10	23	20.000	48.339	0.292
20	19	24.000	16.841	0.587
25	11	40.000	11.835	0.771
30	0	480	0	1.000

表 2.2 帯域幅一定時の再生品質（ミラーリング，MTBS/MTTR の単位は sec）

B_1[Mbit/s]	B_2[Mbit/s]	NoS	MTBS	MTTR	CPR
10	20	11	40.0	10.004	0.799
10	25	1	240	8.067	0.967
20	20	0	480	0	1.000

- 連続再生率（CPR）

  コンテンツの再生時間のうち連続して再生できている割合．CPR = MTBS/(MTBS+MTTR) として算出する．0 から 1 の値を取り，1 に近いほど良い評価尺度である．

合わせて，コンテンツデータのブロック転送速度 B_u を求める．B_u は，ブロックのサイズ U を，ブロック転送をクライアントが要求してから最後のバイトを受信するまでの時間で割って算出する．

表 2.3 帯域幅一定時の再生品質（ストライピング，MTBS/MTTR の単位は sec）

B_1[Mbit/s]	B_2[Mbit/s]	NoS	MTBS	MTTR	CPR
10	20	22	20.869	16.021	0.565
10	25	23	20.0	15.145	0.569
20	20	0	480	0	1.000

2.5.4 結果

B_1, B_2 を一定の値に設定したときの各転送方式における再生中断回数 NoS，平均連続再生時間 MTBS，平均再生再開待ち時間 MTTR，連続再生率 CPR を，表 2.1 から表 2.3 にそれぞれ示す．

S_1 からのみデータを転送したとき（表 2.1），$B_1 \leq 25Mbit/s$ になると再生が中断され，B_1 が小さくなるにつれて指標が悪化している．それに対して，S_1, S_2 両方にコンテンツファイルを配置してから並列転送した場合（表 2.2）には，B_1=20Mbit/s・B_2=20Mbit/s のとき NoS=0，および B_1=10Mbit/s・B_2=25Mbit/s のとき NoS=1 となっている．B_1, B_2 いずれもコンテンツのビットレート B_f を下回っているが，S_1, S_2 両方から転送することで再生品質を維持できている．S_1, S_2 にコンテンツファイルをストライピングして並列転送したとき（表 2.3）には，B_1=20Mbit/s・B_2=20Mbit/s において NoS=0 となっているが，B_1=10Mbit/s・B_2=25Mbit/s，B_1=10Mbit/s・B_2=20Mbit/s においては，連続再生率が 0.57 弱となっている．

次に，ブロック転送速度 B_u の推移を図 2.15 から図 2.17 に示す．図 2.15 を見ると，B_1=30Mbit/s のときは安定した転送速度が得られている．コンテンツのビットレートは，B_f=30Mbit/s であることから，B_1=30Mbit/s では，現在再生中のブロックがすべて再生し終える前に次のブロックの受信が完了しているためといえる．

B_1=20, 25Mbit/s では，k 個目のブロックを再生し終えた時点で，まだ次の $k+1$ 個目のブロックの受信が残っている．さらに，k 個目のブロックの再生が終わると，$k+2$ 個目の

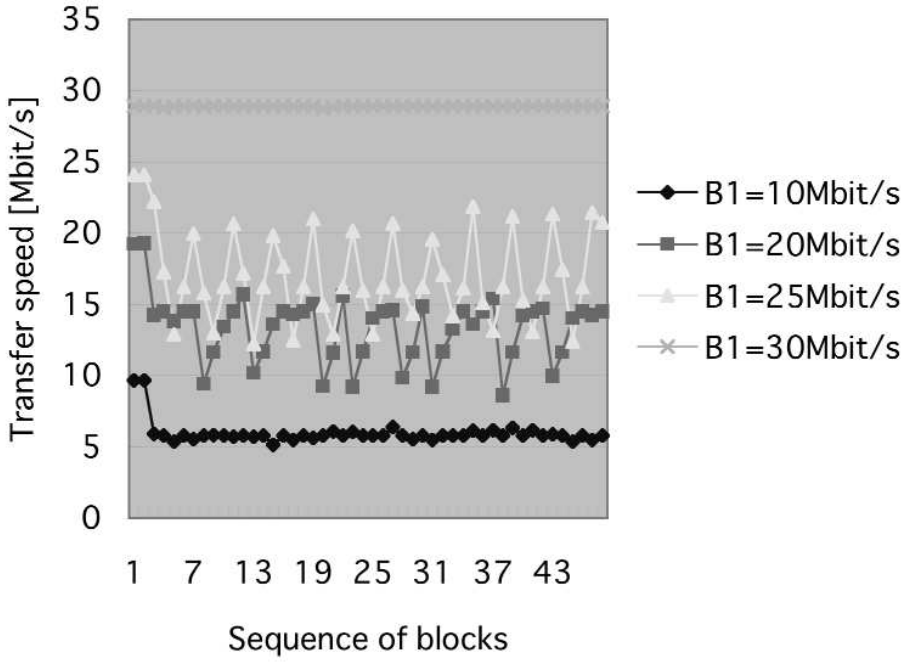


図 2.15 帯域幅一定時のブロック転送速度（S_1 からのみ転送）

ブロックの受信を始めるため，$k+1$ 個目のブロック転送速度が半分近くまで下がる．その後 $k+1$ 個目のブロックの受信が完了し，$k+2$ 個目のブロック転送速度は上向くことになる．その結果，図 2.15 にあるように B_u は振動していると考えられる．

B_1=10Mbit/s になると，一つのブロックを受信するのに 3 ブロック分の時間を要する計算になり，アンダーフローとなってしまう．この場合だと，10 秒相当のブロック 1 個を受信するのに 30 秒かかることになる．

一般化すると，クライアントが一つのブロックを再生し終えたら次のブロックを要求するという配信形態である場合，NoS=0 となる条件は，

$$B_u \geq \frac{B_f}{2} \tag{2.12}$$

となる．よって，提案方式を用いてブロック単位でコンテンツを転送するとき，ミラーリング時には

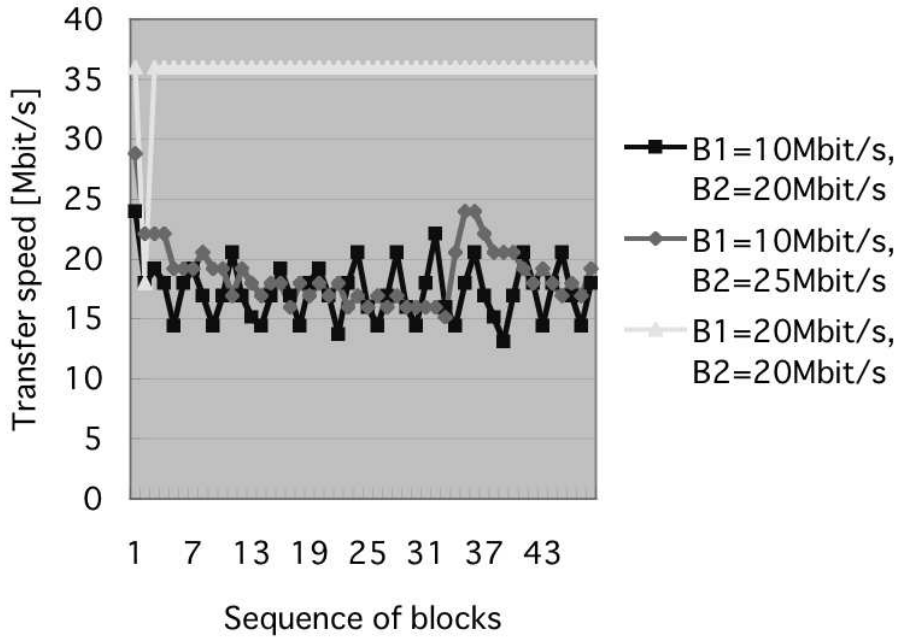


図 2.16 帯域幅一定時のブロック転送速度（ミラーリング）

$$\sum_{i=1}^{N} B_i \geq \frac{B_f}{2} \quad (i = 1, 2, ..., N) \tag{2.13}$$

一方，ストライピング時には，

$$B_i \geq \frac{B_f}{2N} \quad (i = 1, 2, ..., N) \tag{2.14}$$

が成り立つ限りにおいて，B_f[Mbit/s] での連続再生が可能となる．ここでは，その境界条件が，B_u=15Mbit/s となっている．

ミラーリングあるいはストライピングして並列転送した場合，図 2.16・図 2.17 にあるように，図 2.15 と同様のパターンで転送速度が推移しているが，値そのものは増加しており，ミラーリング時には，ほぼすべての転送で B_u ≥15Mbit/s となっている．ゆえに，アンダーフローを起こさずに再生が続けられたといえる．

次に，帯域幅 B_1, B_2 を 10Mbit/s から 40Mbit/s の範囲で一様に変動させたときの再生品質を，表 2.4 から表 2.6 に示す．B_1, B_2 は，T_r=5, 10, 20[sec] ごとにその値を変えている．ミラーリングしてから並列転送したときには，常に NoS=0 を保っている．ストライピ

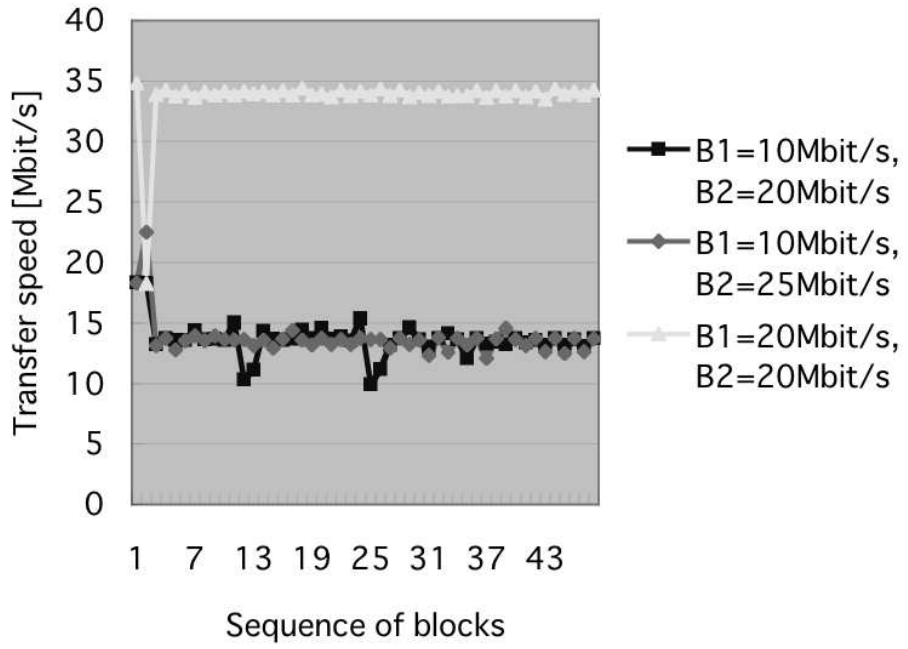


図 2.17 帯域幅一定時のブロック転送速度（ストライピング）

表 2.4 帯域幅変動時の再生品質（S_1 からのみ転送，MTBS/MTTR の単位は sec）

T_r[sec]	NoS	MTBS	MTTR	CPR
5	7	60	14.706	0.803
10	9	48	16.6	0.743
20	5	80	15.745	0.844
30	3	120	15.141	0.887

ングしてから並列転送した場合には，T_r=20, 30[sec] において 2 回再生が中断しているが，T_r=5,10[sec] において．NoS=0 を達成している．

このときのブロック転送速度の平均値（AVG）・最小値（MIN）・最大値（MAX）・標準偏差（SD）を表 2.7 から表 2.9 に，毎回の値の推移を図 2.18 から図 2.21 にそれぞれ示す．帯域幅を一定にしたときと同様に，ブロック転送速度は，ミラーリングしてから並列転送した場合に最も大きくなり，平均で S_1 からのみ転送したときの約 2 倍となっている．次いでストライピングしてから並列転送したとき，S_1 からのみ転送したときの順となっている．

表 2.5 帯域幅変動時の再生品質（ミラーリング，MTBS/MTTR の単位は sec）

T_r[sec]	NoS	MTBS	MTTR	CPR
5	0	480	0	1.000
10	0	480	0	1.000
20	0	480	0	1.000
30	0	480	0	1.000

表 2.6 帯域幅変動時の再生品質（ストライピング，MTBS/MTTR の単位は sec）

T_r[sec]	NoS	MTBS	MTTR	CPR
5	0	480	0	1.000
10	0	480	0	1.000
20	2	160	13.626	0.921
30	2	160	44.594	0.782

図 2.18 から図 2.21 と表 2.4 から表 2.6 に示した再生品質とを見比べると．B_u=15Mbit/s を上回っている限り NoS=0 となっており，B_u がこの値を下回った回数が多いほど，再生品質が悪くなっていることがわかる．

表 2.7 ブロック転送速度の統計値（S_1 からのみ転送）

T_r[sec]	AVG	MIN	MAX	SD
5	20.5	10.9	38.5	7.09
10	20.4	8.55	38.5	8.48
20	24.8	9.82	38.5	8.96
30	26.2	10.88	38.5	9.10

表 2.8 ブロック転送速度の統計値（ミラーリング）

T_r[sec]	AVG	MIN	MAX	SD
5	39.5	26.2	57.6	8.34
10	41.9	28.8	72.0	10.2
20	41.6	24.0	57.6	9.05
30	42.4	26.2	57.6	6.64

表 2.9 ブロック転送速度の統計値（ストライピング）

T_r[sec]	AVG	MIN	MAX	SD
5	32.7	19.5	53.6	7.99
10	34.4	17.0	60.6	9.57
20	33.6	10.8	57.9	12.2
30	34.9	13.2	54.6	10.3

2.6 JGN を利用した実証実験

以上のように，提案方式による品質維持制御の有効性を示すことができた．次に，実ネットワークにおける実証実験として，JGN(Japan Gigabit Network)[*6][57] を利用した系を構築し，提案方式による DV 映像の配信を行った．以下，その内容を述べる．

2.6.1 実験環境

構築した実験系の構成を図 2.22 に示す．JGN への接続地点として，通信・放送機構の高知通信トラヒックリサーチセンター（高知），幕張ギガビットリサーチセンター（幕張），岡山情報通信研究開発支援センター（岡山）の 3 カ所を選んだ．

[*6] http://www.jgn.tao.go.jp/
JGN の運用は 2004 年 3 月で終了するが，後継のテストベッドが計画されている．

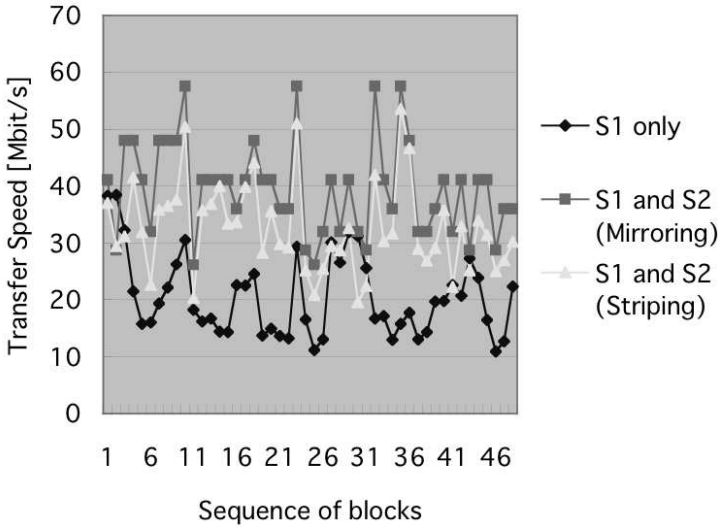


図 2.18 ブロック転送速度の推移（T_r=5[sec]）

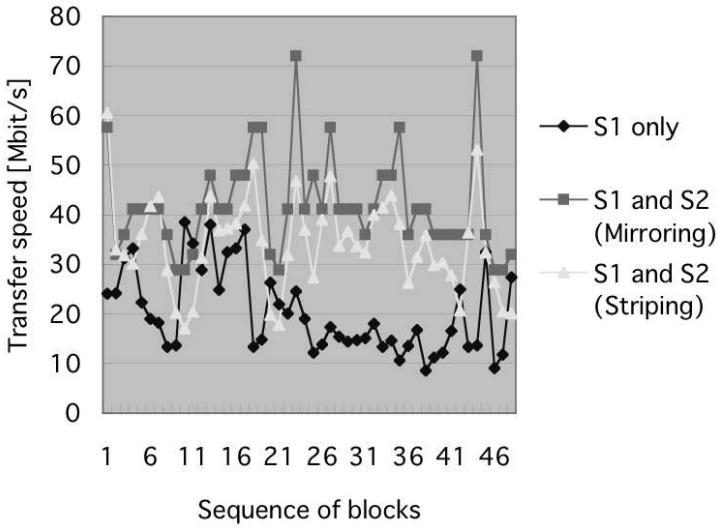


図 2.19 ブロック転送速度の推移（T_r=10[sec]）

使用した機材と主な仕様を表 2.10 に示す．各地点に PC ルータを 1 台ずつ配置し，ATM-PVC(Permanent Virtual Circuit) を設定，互いに通信できるようにした．ルータの内側には，映像配信サーバを岡山と幕張に置き，高知には，クライアント 2 台と負荷トラヒック発生用のホスト 1 台を置いた．これらの外観を，図 2.23 と図 2.24 にそれぞれ示す．配信サーバからクライアントに至る経路は，配信サーバ → PC ルータ → JGN → PC ルータ → クライアントとなる．

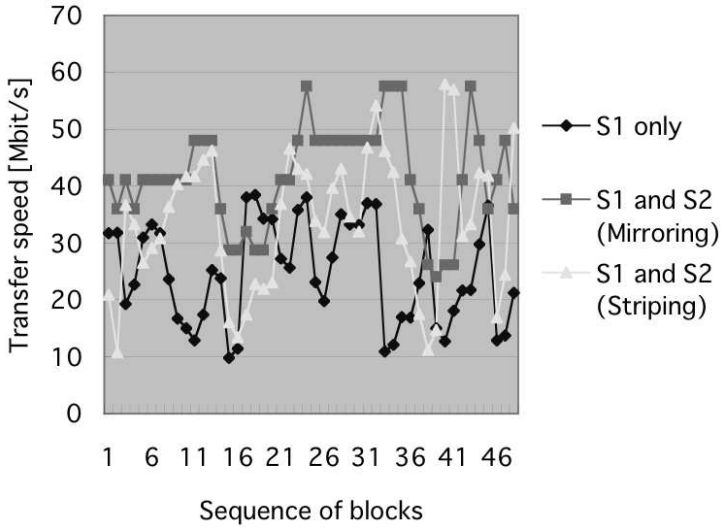


図 2.20 ブロック転送速度の推移（T_r=20[sec]）

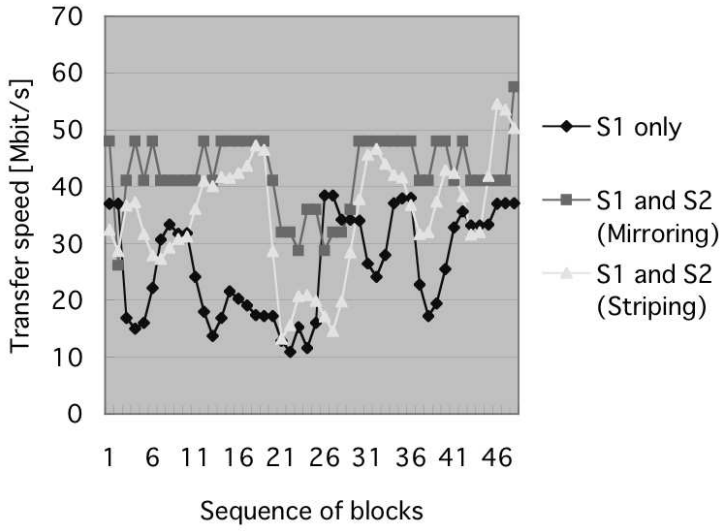


図 2.21 ブロック転送速度の推移（T_r=30[sec]）

2.6.2 実験シナリオ

実験シナリオとしては，前節と同様に再生時間 8 分間の DV ファイル（1,728,000,000bytes）を岡山のサーバ S_1 及び幕張のサーバ S_2 に置き，クライアントが HTTP により転送，再生する．ここでは，2 つの DV ファイル F_1, F_2 を S_1, S_2 に置き，高知のクライアント C_1 が F_1 を，C_2 が F_2 をそれぞれ配信させる．HTTP サーバには Apache を，HTTP クライア

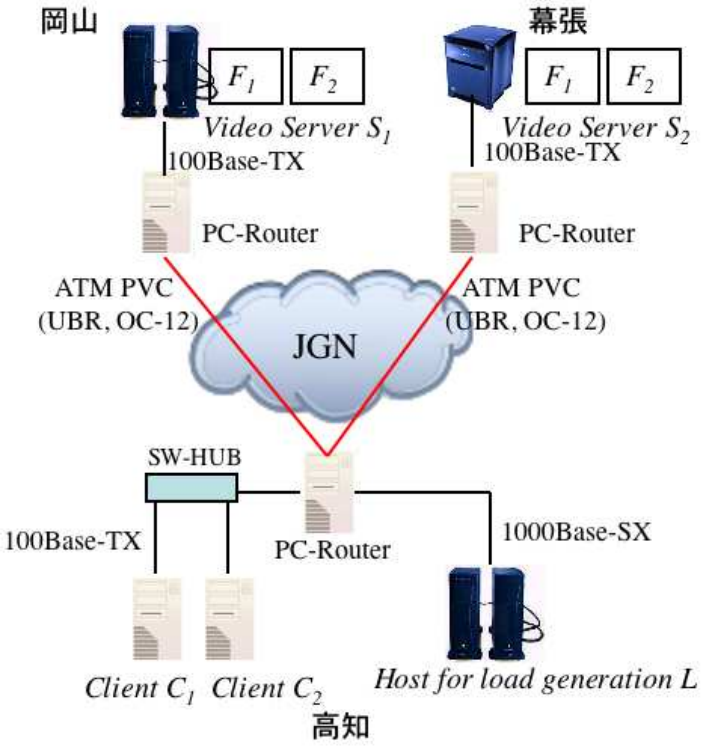


図 2.22　JGN 実験系

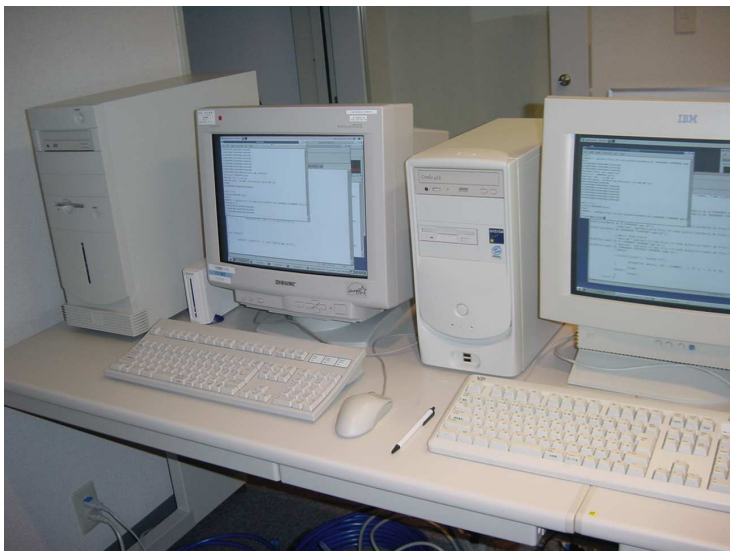


図 2.23　クライアントの外観

表 2.10 使用した機材と主な仕様（JGN 実験）

用途	機種名	OS	CPU	RAM
配信サーバ S_1	SGI Origin200	IRIX 6.5.21m	R10000 200MHz × 2	512MB
配信サーバ S_2	SGI Origin2000	IRIX 6.5.21m	R10000 200MHz × 2	1024MB
PC ルータ	PC/AT	WindowsNT4.0	Pentium-III 600MHz	128MB
クライアント C_1	PC/AT	FreeBSD 5.1-RELEASE	Athlon 1.2GHz	384MB
クライアント C_2	PC/AT	FreeBSD 5.1 5.1-RELEASE	Celeron 800MHz	512MB
負荷発生ホスト L	SGI Origin200	IRIX 6.5.21m	R10000 200MHz × 2	512MB

ントには libcurl を用いた．C_1, C_2 の受信バッファサイズは 192Kbytes に，L の受信バッファサイズおよび S_1, S_2 の送信バッファサイズは 512Kbytes とした．クライアントにおける DV コンテンツ再生の様子を図 2.25 に示す．

DV データ配信のパターンは，次の 5 通りを用意した．

1. C_1, C_2 ともに S_1, S_2 から F_1, F_2 を通常転送
2. C_1 は S_1 から F_1 を通常転送，C_2 は S_1, S_2 から F_2 を並列転送
3. C_1, C_2 ともに S_1, S_2 から F_1, F_2 を並列転送
4. C_1 は S_1 から F_1 を通常転送，C_2 は S_1, S_2 にストライピングされた F_2 を並列転送
5. C_1, C_2 ともに S_1, S_2 にストライピングされた F_1, F_2 を並列転送

図 2.24　負荷発生ホストの外観

図 2.25　DV 再生の様子

これらを図示したものを，図 2.6.2 から図 2.6.2 に示す．いずれのパターンも，300 フレーム（36,000,000bytes）を 1 ブロックとした転送を行う．ストライピングの単位は 30 フレーム（3,600,000bytes）とした．受信した DV データは IEEE1394 インタフェースへと出力され，再生される．

配信は，まず C_1 から開始し，C_1 が再生を開始したら C_2 への転送が始まる．その後，負

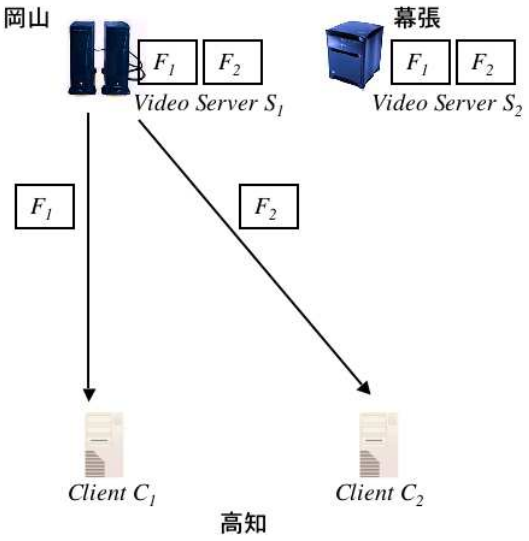


図 2.26 配信パターン 1

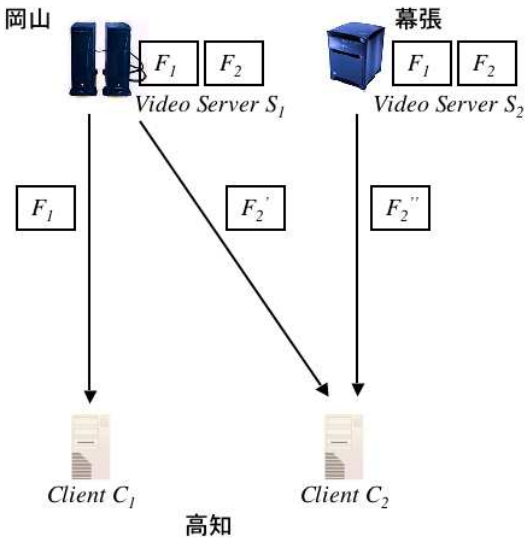


図 2.27 配信パターン 2

荷発生用ホスト L が，図 2.31 に示すように，S_1 および S_2 にある 4 つのファイルを HTTP により同時に転送する．これを 30 秒間続けたら転送を打ち切り，30 秒間休止した後これを繰り返す．S_1, S_2 から PC ルータへは 100Base-TX を通っているのに対し，高知の PC ルータから L へは 1000Base-SX を通っているため，C_1, C_2 への DV データ転送速度がこのトラヒックにより低下する．この状況において，通常転送時，ミラーリングによる並列転送時，

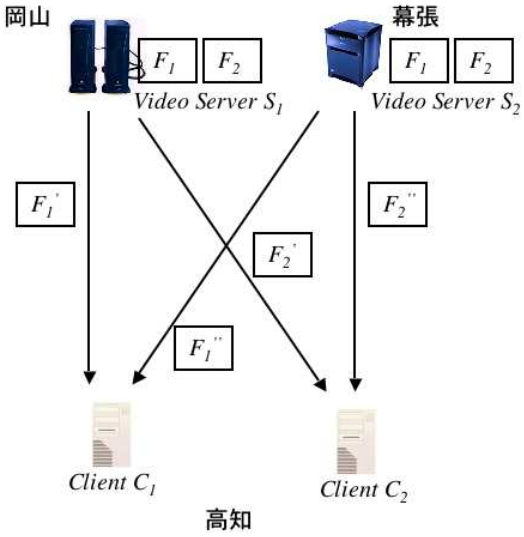


図 2.28 配信パターン 3

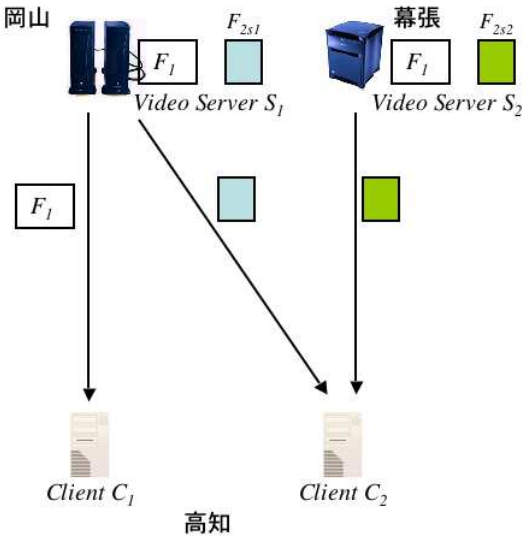


図 2.29 配信パターン 4

ストライピングによる並列転送時の C_1, C_2 での再生品質と転送速度を評価した.

2.6.3 結果: 再生品質

再生品質として，前節の実験と同様に，再生中断回数（NoS），平均再生中断間隔（MTBS），平均再生再開待ち時間（MTTR），連続再生率（CPR）を求めた．各配信

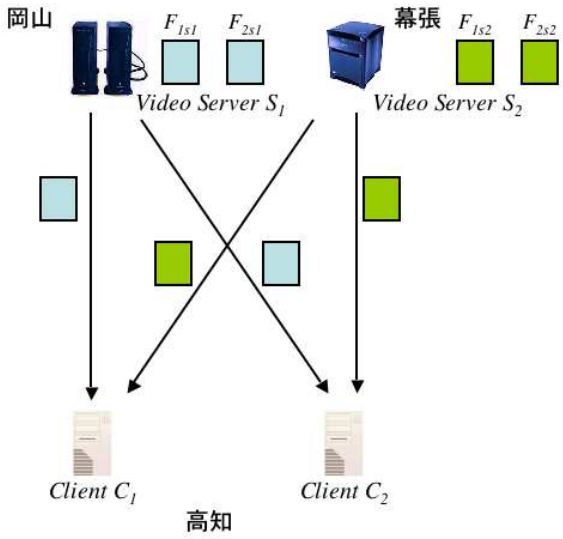


図 2.30　配信パターン 5

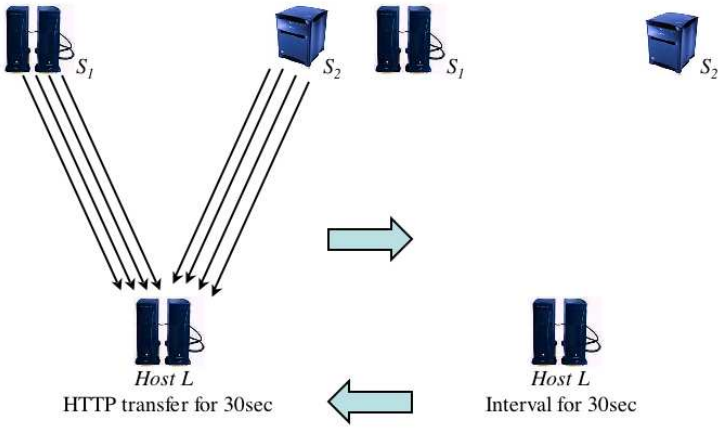


図 2.31　負荷トラヒック

パターンにおける C_1, C_2 でのこれらの値を，表 2.11，表 2.12 にそれぞれ示す．

パターン 1 では再生の中断が頻発するのに対し，パターン 2 からパターン 5 では 0 ないし 1 回の再生中断ですんでいる．パターン 2 およびパターン 4 は，片方が通常転送，もう片方が並列転送したときであることから，一部のクライアントが提案方式を採用したときにも，システム全体の負荷分散に貢献していることがわかる．

表 2.11 C_1 における再生品質（MTBS/MTTR の単位は sec）

パターン	NoS	MTBS	MTTR	CPR
1	10	43.64	11.81	0.787
2	1	240	8.076	0.967
3	0	480	0	1.000
4	0	160	0	1.000
5	1	240	13.14	0.948

表 2.12 C_2 における再生品質（MTBS/MTTR の単位は sec）

パターン	NoS	MTBS	MTTR	CPR
1	6	68.57	12.11	0.850
2	1	240	0	1.000
3	0	480	13.626	1.000
4	0	480	44.594	1.000
5	1	240	13.626	0.946

2.6.4 結果: 転送速度

次に，各配信パターンにおける C_1, C_2 への DV データ転送速度の統計値を表 2.13，表 2.14 に，その推移を図 2.32 から図 2.36 にそれぞれ示す．パターン 1 では負荷トラヒックの発生に合わせて転送速度が低下し，負荷発生後は 10Mbit/s から 25Mbit/s の間を上下している．前節と同じ条件で再生しているため，転送速度が 15Mbit/s を下回った結果，アンダーフローを起こすことが確認できる．

これに対し，パターン 2 からパターン 5 ではいずれも再生品質は良好であったが，転送速度の推移を見ると，ほとんどのブロック転送時に 20Mbit/s 以上の転送速度が得られており，再生品質の維持に繋がっていることがわかる．全体として，パターン 3 のときに最も安定し

表 2.13 C_1 における転送速度の統計値（単位は Mbit/s）

パターン	平均値	最小値	最大値	標準偏差
1	27.26	9.73	91.05	22.40
2	42.88	13.7	90.97	22.10
3	39.09	19.2	72.0	11.45
4	40.22	16.29	90.74	17.97
5	38.64	13.66	65.59	13.87

表 2.14 C_2 における転送速度の統計値（単位は Mbit/s）

パターン	平均値	最小値	最大値	標準偏差
1	26.48	9.79	91.02	20.87
2	31.93	13.71	72.0	13.43
3	40.32	20.57	72.0	11.49
4	42.65	17.27	73.45	14.79
5	41.19	13.21	73.25	15.56

た転送が行われており，転送速度が押し上げられている．

2.7 まとめ

本章では，CDN によって映像コンテンツの配信を行おうとしたとき，最適と思われるサーバを 1 台選択するだけでは，配信途中のネットワーク品質の劣化に合わせてクライアントでの再生品質も劣化するという問題点を指摘した．その解決法として，複数台のサーバがあることを利用した並列転送方式を提案した．提案方式を採用することにより，各経路の転送速度がコンテンツのビットレートを下回っていても，解像度及びフレームレートを下げることなく品質を維持できることを実証した．

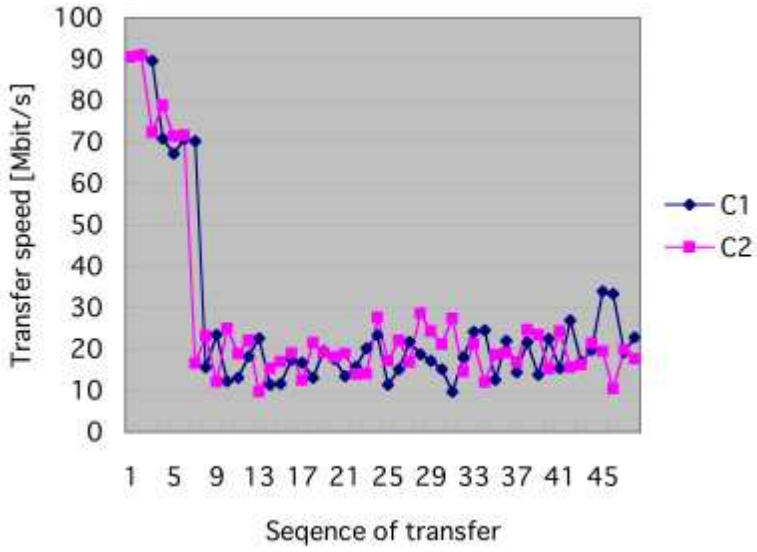


図 2.32　転送速度の推移（JGN 実験パターン 1）

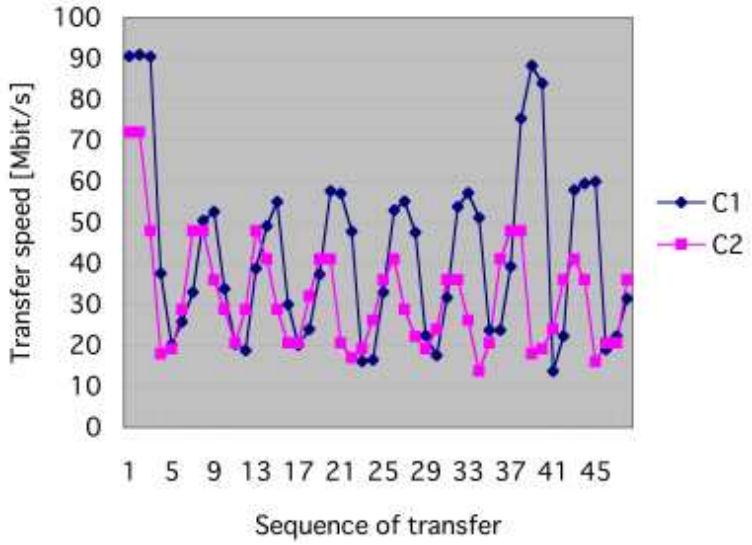


図 2.33　転送速度の推移（JGN 実験パターン 2）

ここでの実験環境は，100Mbit/s の容量に対して 30Mbit/s のビットレートという組み合わせであったが，この値は今後より大きくなることが予想される．現に，LAN の端末では 1000Base-T が普及しつつあり，LAN バックボーンあるいは WAN では 10G Ethernet が間もなく導入されようとしている．ビットレートが上がるにつれてそれを維持するのは難しいため，提案方式による「上位層でフローを分散する」という概念はさらに重要性を増すも

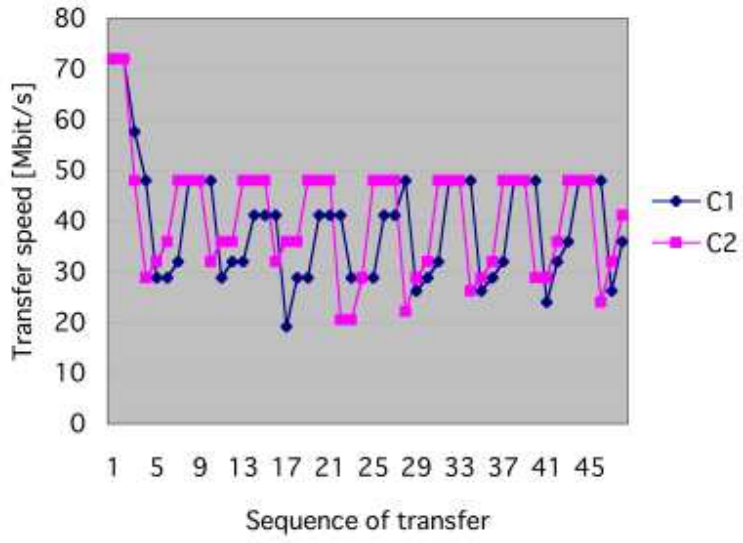


図 2.34　転送速度の推移（JGN 実験パターン 3）

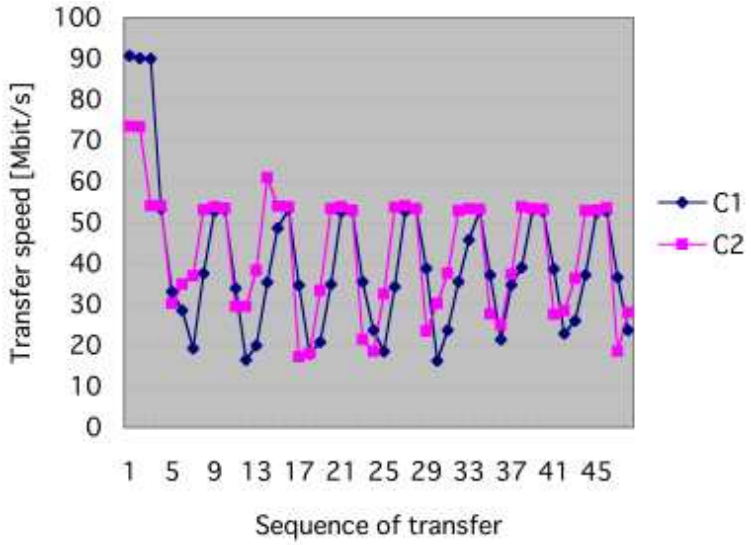


図 2.35　転送速度の推移（JGN 実験パターン 4）

のと考える．

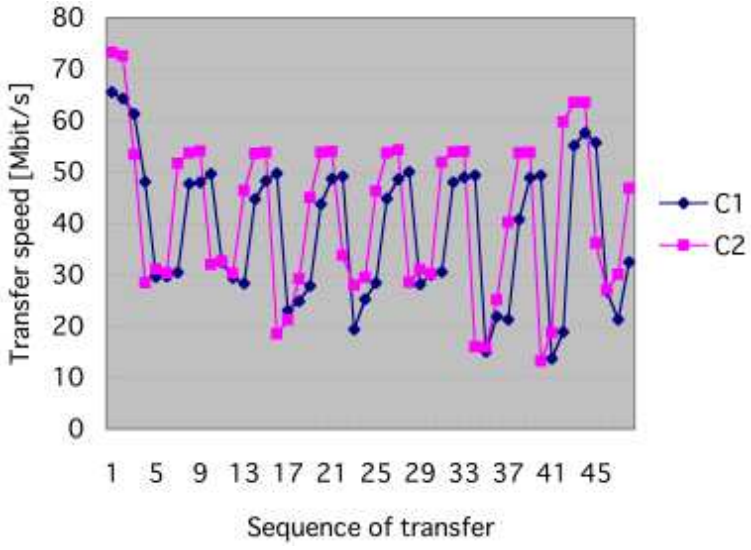


図 2.36　転送速度の推移（JGN 実験パターン 5）

第 3 章

コンテンツ配布時における転送効率の改善

前章では，複数サーバからクライアントへのコンテンツ配信について議論した．この章では，オリジンサーバに投入されたコンテンツをサロゲートに配布する段階に焦点を当て，前章で示した並列転送を応用したコンテンツ配布方式を提案する．提案方式により，従来の同報ベースの配布に対してネットワーク品質の変動に左右されずに転送効率が改善されること，配布先のサロゲート数の増加に対して改善効率が上がることを理論的に導く．続けて，実験により提案方式が有効であることを実証する．

3.1 従来のコンテンツ配布方式

3.1.1 ユニキャストによる配布

始めに，本論文のおけるコンテンツ配布を，「オリジンサーバに投入されたコンテンツファイルを，対象とする N 台のサロゲート（分散配置されたミラーサーバ）に複製すること」と定義する．最も簡単にコンテンツを配布するには，図 3.1 に示すように，コンテンツファイルをユニキャストにより転送すればよい．このとき，オリジンサーバ S_0 の帯域幅を B_0[Mbit/s]，S_0 からサロゲート $S_i \quad (i = 1, 2, ..., N)$ に至る経路の帯域幅 B_i[Mbit/s] としたとき，次の関係が成り立つものとする．

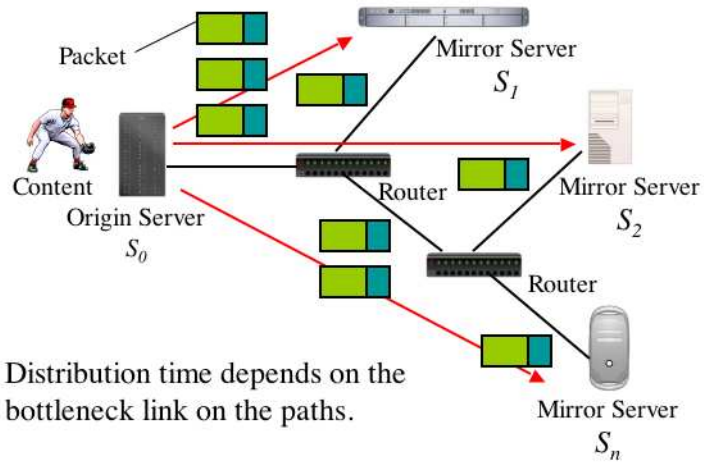


図 3.1　ユニキャストによるコンテンツ配布

$$B_0 \geq \sum_{i=1}^{N} B_i \tag{3.1}$$

また，S_0 から S_i に至る経路間ではボトルネックを共有していないとすると，S_f[Bytes] のコンテンツファイルを S_i に配布するのに要する時間 T_{du}[sec] は，

$$T_{du} = \max(\frac{S_f}{B_1}, \frac{S_f}{B_2}, ..., \frac{S_f}{B_N}) = \frac{S_f}{\min(B_i)} \tag{3.2}$$

となる．転送速度で言えば，B_i の最小値と等しいことになる．

ユニキャストによるコンテンツ配布は，HTTP や FTP など標準的なプロトコルで容易に実行可能である．しかし，N に比例して S_0 で消費される帯域が増えると共に，ネットワーク内を通るパケット数も N に比例して増加してしまう．

代替案として，同時に転送せずに，最初 S_1 に転送し，次に S_2，S_3，...，最後に S_N と順番に配布する手法もある．こうすれば，S_0 およびネットワーク内で消費される帯域を抑えることができるが，配布時間 T_{du} は，次式に示すようになる．

$$T_{du} = \frac{S_f}{B_1} + \frac{S_f}{B_2} + ... + \frac{S_f}{B_N} = S_f \sum_{i=1}^{N} \frac{1}{B_i} \tag{3.3}$$

結局，順番に配布すると，T_{du} は N の増加につれて大きくなってしまう．

3.1.2　IP マルチキャストによる配布

ユニキャストによる配布では，中身が同一のパケットを重複して送信するため帯域の無駄使いとなる．このように，決められたホストの集合に対して効率的に同報（ブロードキャスト）する手段として，IP マルチキャスト（IP Multicast: 以下マルチキャスト）が研究されている．マルチキャストでは，図 3.2 に示すようにパケットが転送される [58].

まず，送信端末からパケットが 1 個だけ直近のルータ R_1 に送られる．R_1 は，このパケットを必要な数だけ複製し，下流のルータ R_2, R_3 に転送する．R_2, R_3 も同様にパケットを複製し，受信端末に転送する．こうして，ルータがパケットを適宜複製することでパケットの重複を無くし，受信端末数に関わらずに一定の帯域しか使わないですむという利点がある．

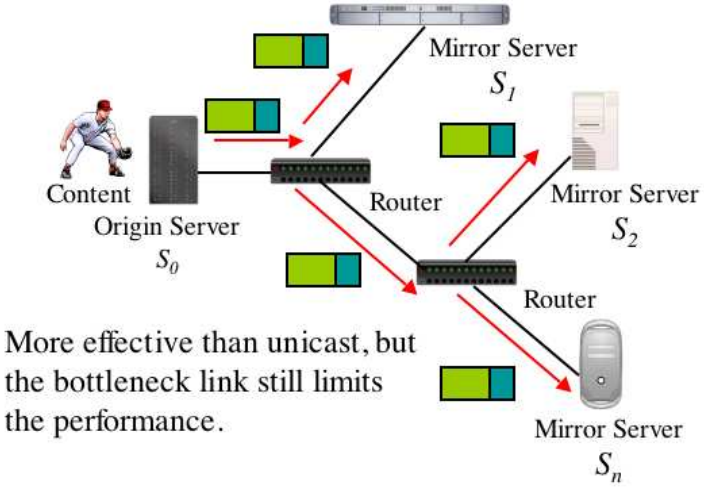


図 3.2　IP マルチキャストによるコンテンツ配布

実際のプロトコルでは，送受信端末間でグループを形成し，グループの識別子としてマルチキャストアドレスを設定する．送信端末は，このマルチキャストアドレスを宛先にしてパケットを送信する． IPv4 では，図 3.3 に示すように，マルチキャストアドレス用としてクラス D が設けられており，224.0.0.0 から 239.255.255.255 までの範囲が該当する．

IPv6 では，マルチキャストの機能は標準としてすべてのルータ・ホストが実装すべきものとされている．IPv4 と比べ，どの範囲までマルチキャストパケットを転送するか決める

図 3.3　IPv4 アドレスの各クラス

概念として「スコープ（Scope)」を導入している点，IGMP が ICMP メッセージの一つとして統合されている点などが異なっている [59]．IPv6 マルチキャストアドレスの構成を，図 3.4 に示す．

図 3.4　IPv6 マルチキャストアドレス

マルチキャストを実現するには，受信端末がマルチキャストグループへの参加・離脱を意思表示を受付け，それを基にしてパケットを転送すべきルータの構成（マルチキャストツリー）を保守し，ルータ間でのマルチキャストパケットの転送を実効する必要がある．前者

のプロトコルとして，図 3.5 に示す IGMP(Internet Group Management Protocol)[60] がある．後者のプロトコルはマルチキャストルーティングプロトコルと呼ばれる．代表的なマルチキャストルーティングプロトコルを表 3.1 に示す [61].

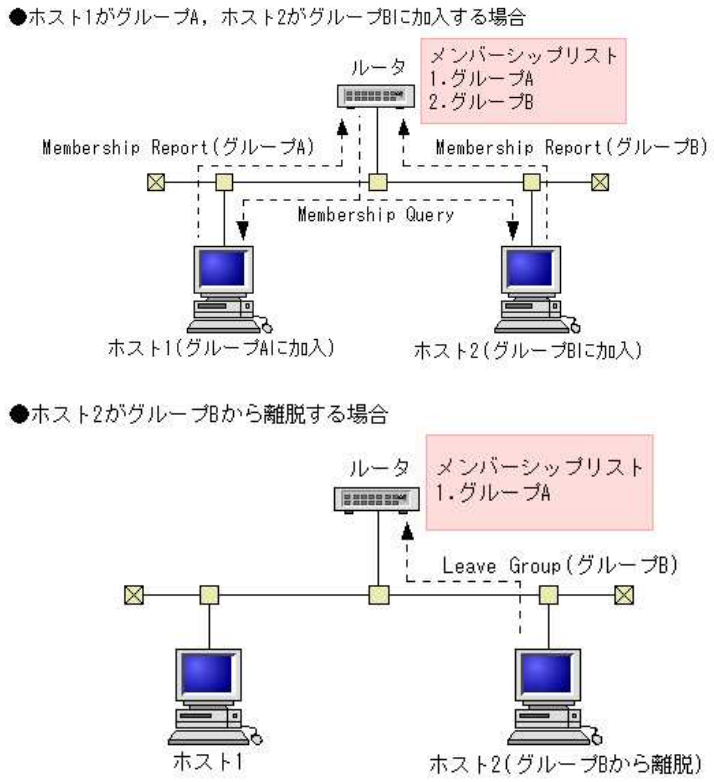


図 3.5　IGMP の動作

表 3.1　主なマルチキャストルーティングプロトコル

プロトコル	概要	仕様
DVMRP	距離ベクトル型	RFC1075
MOSPF	OSPF を拡張	RFC1584
CBT	共有ツリーによる配送	RFC2189
PIM-SM	ユニキャストルーティングプロトコルに依存しない，分散型	RFC2362

IP マルチキャストは，あくまで IP の機能のひとつであって通信の信頼性は保証していない．そのため，ファイル配布など，信頼性が求められる場合に対応するために，信頼性マル

チキャストプロトコル（Reliable Multicast Protocol）が提案されている [62]．信頼性マルチキャストプロトコルで提案されている信頼性付与の方式を，表 3.2 に示す．現在のところ，それぞれの方式にまだ問題点が残っており，これを改善する研究が進められている．

表 3.2　信頼性マルチキャストプロトコルの方式

方式	概要	問題点
ACK-based	単純に ACK を返す	ACK が溢れる
Tree-based ACK	ACK をツリー上に集約する	ツリーの管理が煩雑になる
NACK-based	否定応答を返し再送してもらう	確実性に欠ける
FEC*1	誤り訂正符号を付けて送信する	余分な帯域が必要

マルチキャストによってコンテンツを配布した場合の配布時間 T_{dm}[sec] は，

$$T_{dm} = \max(\frac{S_f}{B_i}) = \frac{S_f}{\min(B_i)} \tag{3.4}$$

となり，式 (3.2) と同じである．ただし，式 (3.2) は，各経路間でボトルネックが共有されていないことが条件であるのに対し，マルチキャストの場合は，同一リンクにおけるパケットの重複がないため，常に式 (3.4) が成り立つ．

3.2　提案する配布方式

3.2.1　従来の配布方式の問題点

これまで，ユニキャストおよびマルチキャストによるコンテンツ配布を論じたが，配布に要する時間は，B_i の最小値で決定することになる．これは，他のサロゲートがコンテンツデータをすべて受信し終えていたとしても，残り一つの配布が遅いままという状態になる．実際には，前章で述べたように，エンドホスト間の帯域幅は時間によって変動するため，その振る舞いによってはさらに配布に時間がかかる可能性がある．早く配布し終えた方がそれだけ早くユーザにコンテンツを提供できるため，ネットワーク品質に依らずに性能を確保で

きる配布方式が望まれる．

3.2.2　ストライピングと並列転送を組み合わせた配布方式

この問題に対して，前章で提案した，並列転送によってコンテンツのビットレートを維持する方式を応用することで，コンテンツ配布時の効率を改善することを提案する．提案するコンテンツ配布方式は，次の二つのステップを踏む．

1. S_i にコンテンツデータをストライピングする
2. ストライピングされた断片を複数の S_i から並列に取得する

まず，コンテンツデータをストライピングする処理を，図 3.6 に示す．コンテンツのサイズを S_f[Bytes] としたとき，これを N 等分し，それぞれを S_i に転送する．ストライピングされた断片を P_i とする．そのサイズは，S_f/N[Bytes] となる．

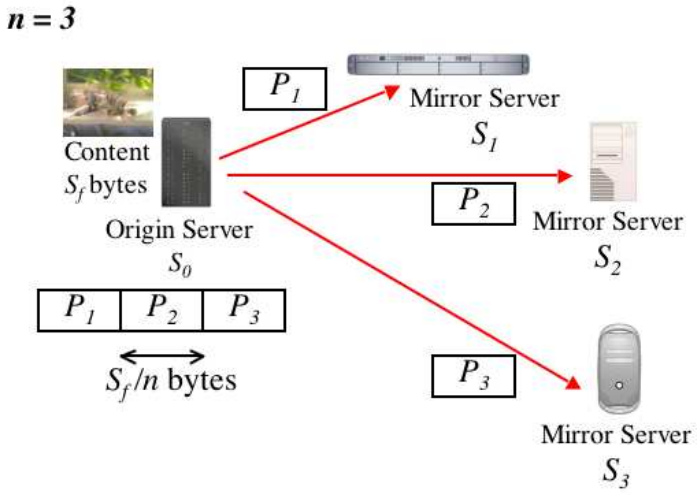


図 3.6　コンテンツのストライピング

これが終わった時点で，S_1 には P_1 が，S_2 には P_2 が，以下同様に S_N には P_N があることになる．次のステップは，残り $N-1$ 個の断片を取得することになる．最初に，S_1 が P_2 を，S_2 が P_3 を，以下同様にして S_N が P_1 を受信する．このとき，P_2 は S_0 と S_2 にあり，P_3 は S_0 と S_4 にあり，以下同様に P_1 は S_0 と S_1 にあることに着目し，図 3.7 に示すよう

に，2 台のサーバから並列に P_i を転送する．ここで言う並列転送は，前章で述べたミラーリングされたコンテンツの並列転送と同様に行うものとする．

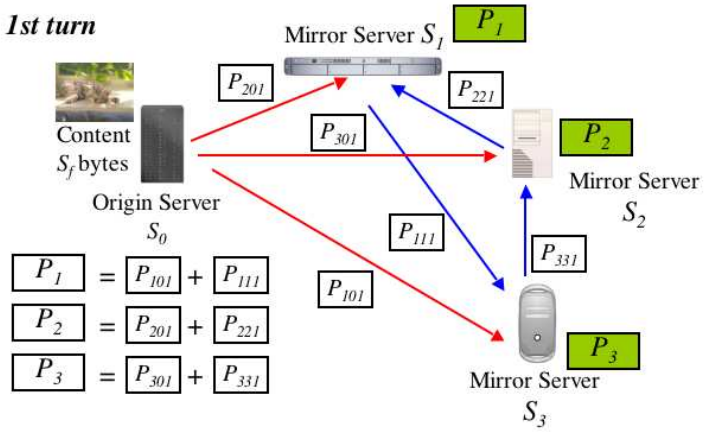


図 3.7　2 台のサーバからの並列転送

この転送が完了すると，今度は，S_i に 2 つの断片がある状態になる．ゆえに，次の断片を 3 台のサーバから並列に転送，受信する．例として，$N = 3$ のときの配布転送を図 3.8 に示す．この場合，S_1 は P_3 を S_0, S_3, S_2 から，S_2 は P_1 を S_0, S_1, S_3 から，S_3 は P_2 を S_0, S_2, S_1 からそれぞれ並列に転送，受信する．

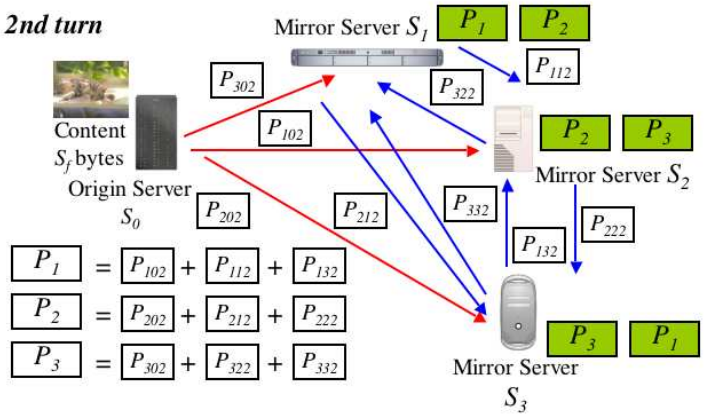


図 3.8　3 台のサーバからの並列転送

以下，k 回目の転送時には $k+1$ 台のサーバから並列転送を受け，$N-1$ 回の転送ですべての断片が S_i に配布されることになる．

次に，この方式による配布時間 T_{dp}[sec] を考える．T_{dp} は，最初のストライピングに要する時間 T_{st} と，その後の $N-1$ 回の転送時間 $T_{pr}(i)$ の和で表される．

$$T_{dp} = T_{st} + \sum_{i=1}^{N-1} T_{pr}(i) \tag{3.5}$$

ここで，T_{st} は，次式により求めることができる．

$$T_{st} = \frac{S_f/N}{\min(B_{0i})} \tag{3.6}$$

一方，$T_{pr}(1), T_{pr}(2), ..., T_{pr}(N-1)$ は，$S_i(i = 0, 1, ..., N)$ から $S_j(j = 0, 1, ..., N)$ に至る経路の帯域幅を S_{ij}[Mbit/s] とすると，次のようになる．

$$T_{pr}(1) = \max(\frac{S_f/N}{B_{01} + B_{21}}, \frac{S_f/N}{B_{02} + B_{32}}, ..., \frac{S_f/N}{B_{0N} + B_{1N}}) \tag{3.7}$$

$$T_{pr}(2) = \max(\frac{S_f/N}{B_{01} + B_{31} + B_{41}}, \frac{S_f/N}{B_{02} + B_{42} + B_{52}}, ..., \frac{S_f/N}{B_{0N} + B_{2N} + B_{3N}}) \tag{3.8}$$

$$T_{pr}(N-1) = \max(\frac{S_f/N}{B_{01} + B_{21} + ... + B_{N1}}, \frac{S_f/N}{B_{02} + B_{12} + ... + B_{N2}}, ..., \frac{S_f/N}{B_{0N} + B_{1N} + ... + B_{N-1N}}) \tag{3.9}$$

上の式と式 (3.6) を比較すると，分子は S_f/N と同じで，分母が大きくなる分，

$$T_{pr}(i) \leq T_{st} \quad (i = 1, 2, ..., N-1) \tag{3.10}$$

が成り立つ．

ゆえに，式 (3.5) と式 (3.10) から，次の関係が成り立つ．

$$T_{dp} \leq \sum_{i=1}^{N-1} T_{st} = N \times \frac{S_f/N}{\min(B_{0i})} = \frac{S_f}{\min(B_{0i})} \tag{3.11}$$

すなわち，提案方式によるコンテンツ配布時間は，ユニキャスト・マルチキャストによる配布時間を下回ることがわかる．B_ij が時間と共に変動するとしても，$T_{pr}(i)$ の分母が B_{ij} の和を取っているため，式 (3.11) は引き続き成り立つ．

3.2.3 提案方式の利点

提案方式によるコンテンツ配布の利点として，以下のものが挙げられる．

1. 配布時間の短縮

   上記のとおり，$1/N$ の大きさでコンテンツデータを分散させておいてからオリジンサーバとサロゲートの両方から残りのデータを転送することで，ユニキャスト・マルチキャストのときと比べて配布時間を短縮できる．

2. 帯域変動時の耐性

   B_{ij} が時刻 t の関数 $b_{ij}(t)$ であったとき，ユニキャスト・マルチキャストによる配布では，その影響を直接被ってしまい，配布時間がさらに延びてしまう．しかし，提案方式では，そのときの帯域幅に比例させて各サーバから転送されるデータ量を配分するため，理論上，常に転送速度が各経路の帯域幅の和となる．ゆえに，ある経路の帯域幅が一時的に低下しても，一定以上の転送速度が保たれる．

3. 追加の実装が不要

   マルチキャストは効率的な通信手段であるが，現在のところ広く普及していない．これは，マルチキャストルーティングプロトコルの実装が煩雑であること，ネットワークの規模が大きくなるとマルチキャストツリーを維持するのにコストがかかることが主な原因と考えられる．一方，提案方式では，個々の転送は通常のユニキャストですむため，追加の実装が不要である．

4. 規模適応性がある

   ネットワークなど対象システムの規模が大きくなったときにもプロトコルが効果的に運用できるとき，そのプロトコルには規模適応性（スケーラビリティ: Scalability）があ

るという．提案方式の場合，T_{dp} は N が増えるにつれて小さくなることがわかり，この点で規模適応性があると言える．

このうち，規模適応性についてさらに検討を進める．例として，$B_{ij} = B$ と一定であると仮定すると，T_{dp} は，次のようになる．

$$T_{dp} = \frac{S_f/N}{B} + \frac{S_f/N}{2B} + ... + \frac{S_f/N}{NB} = \frac{S_f}{NB}\sum_{k=1}^{N}\frac{1}{k} \tag{3.12}$$

ここで，

$$\ln(N) \leq \sum_{k=1}^{N}\frac{1}{k} \leq \ln(N+1) \tag{3.13}$$

となることが知られており [63]，式 (3.12) は，N の単調減少関数となる．

3.3 実験

提案方式によるコンテンツ配布を，前章で述べた JGN を利用した実験系で検証した．以下，その内容と結果について述べる．

3.3.1 実験環境

この実験で用いた実験環境を図 3.9 に，使用した機材を表 3.3 にそれぞれ示す．基本的な構成は前章の実験のときと変わらないが，S_1, S_2 のネットワークインタフェースに Gigabit Ethernet(1000BASE-SX) を利用する．そして，高知にあるサーバ S_0 をオリジンサーバと見なし，岡山および幕張のサーバ S_1, S_2 にファイルを配布するものとする．ここで，S_0 は前章の実験でいう負荷発生用ホスト L と同じ機種であり，S_1, S_2 も同様である．

最初に，実験系の基本特性として，S_0-S_1 間，S_0-S_2 間，S_1-S_2 間の RTT と TCP スループットを測定した．ping コマンドにより RTT を測定した結果，図 3.10 に示すようになった．次に，Iperf[64] により得られた各パスの最大スループットを，図 3.11 に示す．ウィンドウサイズを 192Kbytes としたときの FTP および HTTP によるファイル転送速度は，図

3.12 に示すように，高知ー岡山間で 120Mbit/s, 高知ー幕張間で 80Mbit/s, 岡山ー幕張間で 60Mbit/s となった．これはちょうど RTT に反比例した形となっている．

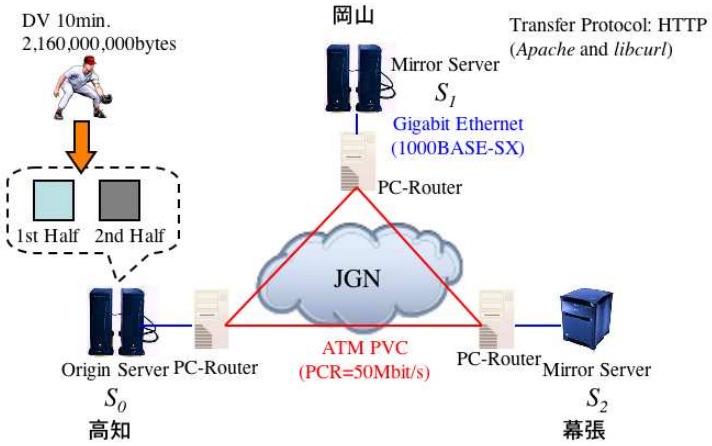


図 3.9　JGN を利用したコンテンツ配布実験環境

表 3.3　コンテンツ配布実験の使用機材

用途	機種名	OS	CPU	RAM
オリジンサーバ S_0	SGI Origin200	IRIX 6.5.21m	R10000 200MHz × 2	512MB
ミラーサーバ S_1	SGI Origin200	IRIX 6.5.21m	R10000 200MHz × 2	512MB
ミラーサーバ S_2	SGI Origin2000	IRIX 6.5.21m	R10000 200MHz × 2	1024MB
PC ルータ	PC/AT	WindowsNT4.0	Pentium-III 600MHz	128MB

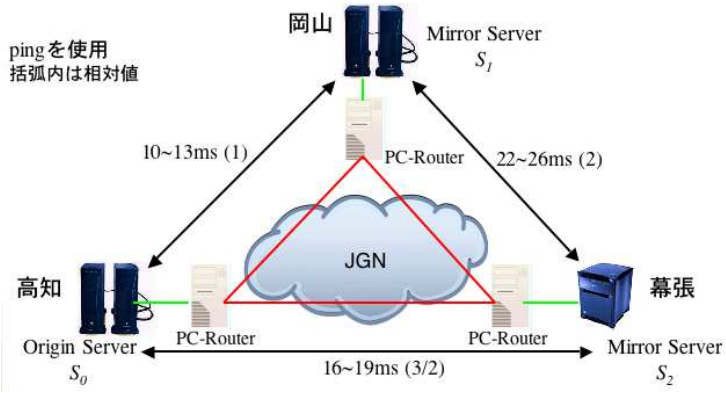


図 3.10　サーバ間の RTT

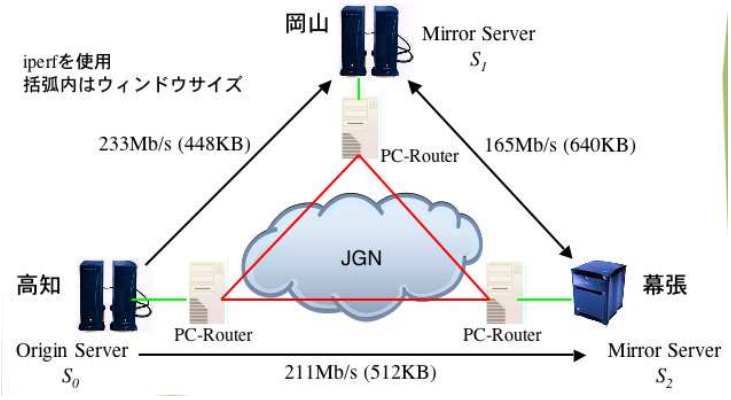


図 3.11　サーバ間の TCP スループット

3.3.2　実験シナリオ

この実験系において，再生時間 10 分間の DV コンテンツを 5 分ずつ 2 つのファイル F_1, F_2 に分けて S_0 に置き，S_1, S_2 に配布する．F_1, F_2 のサイズ S_f は，

$$S_f = 120,000 bytes/frame \times 30 frames/sec \times 300 sec = 1,080,000,000 bytes \quad (3.14)$$

となる．

配布のパターンとして，まず，図 3.13 に示すように，両者とも通常のユニキャストにより転送するパターン 1 を実行する．次に，提案方式に従い，図 3.14 に示すように，S_1 に F_1 を，S_2 に F_2 をまず転送し，次に図 3.15 に示すように，S_0 と S_2 から F_2 を S_1 に，S_0

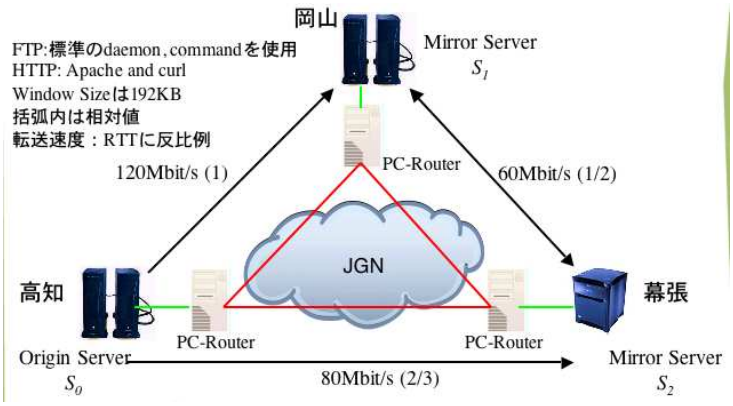


図 3.12　サーバ間の FTP/HTTP スループット

と S_1 から F_1 を S_2 に並列転送するパターン 2 の二つを実行する．これらの転送速度と転送時間を比較する．並列転送時のブロックサイズは 64Mbytes とした．転送プロトコルには HTTP を用い，HTTP サーバには Apache を，HTTP クライアントには libcurl を利用した．

エンドホストのリンク速度がネットワーク内のリンク速度を上回るという条件を満たすよう，高知，岡山，幕張に置かれた PC ルータを結ぶ PVC の PCR(Peak Cell Rate) を上り下りとも 50Mbit/s に設定した．他に競合するトラヒックがなければ，F_1 あるいは F_2 の転送時間 T_d は，

$$T_d = \frac{8 \times 1,080,000,000 bits}{50 \times 10^6 bit/s} = 172.8 sec \tag{3.15}$$

となる．ただし，実際に測定したところ，PCR を 50Mbit/s に設定したときの実効転送速度は 40Mbit/s から 42Mbit/s 程度であった．

この DV データ転送と合わせて，図 3.16 に示す負荷トラヒックを発生させる．負荷トラヒックの方向は，S_0 から S_1 に向かうもの，S_0 から S_2 に向かうもの，S_1 から S_2 に向かうもの，そして S_2 から S_1 に向かうものとする．いずれも Iperf により，T[sec] 周期で 1 本の TCP フローの発生と休止を繰り返す．

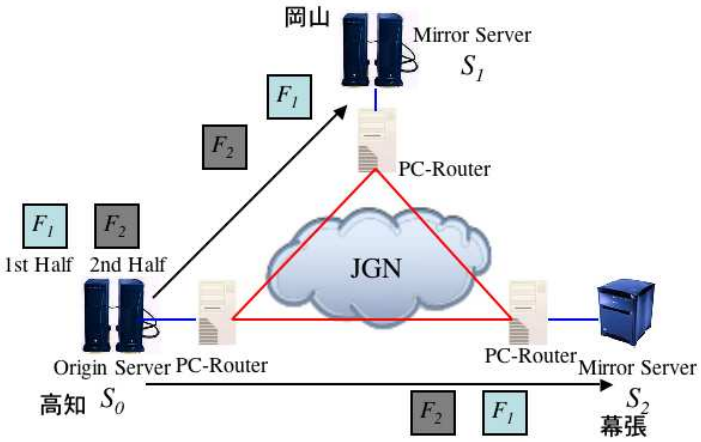


図 3.13　コンテンツ配布パターン 1（通常転送）

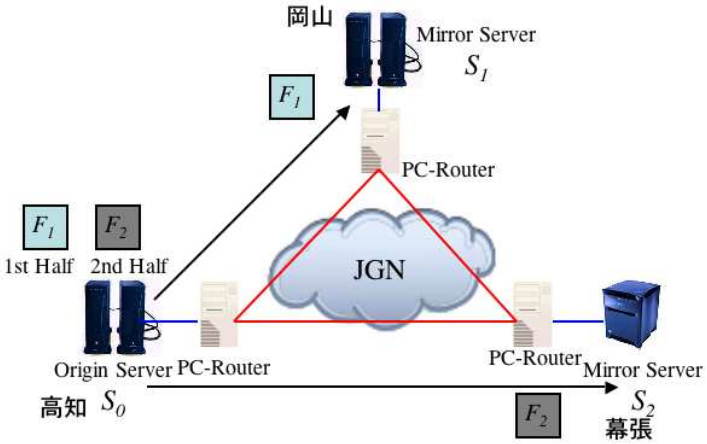


図 3.14　コンテンツ配布パターン 2（提案方式）

3.3.3　結果

負荷トラヒックの発生周期 T を 15sec, 30sec, 60sec としたときの DV ファイルの転送速度と転送時間を表 3.4 に示す．前半のファイル配布については，パターン 1，パターン 2 ともほぼ同じ転送速度，転送時間となっている．これは，両者とも通常のユニキャストによる転送の結果であり，妥当といえる．

後半のファイル配布を比較すると，T=15sec ではパターン 2 の転送速度はパターン 1 のそれと比べて 15%向上している．さらに，T =30sec では 52%，T=60sec では 65%転送速

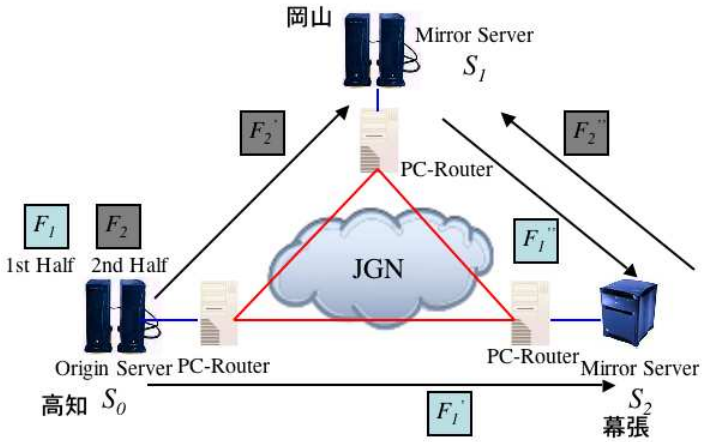


図 3.15　コンテンツ配布パターン 2 続き（提案方式）

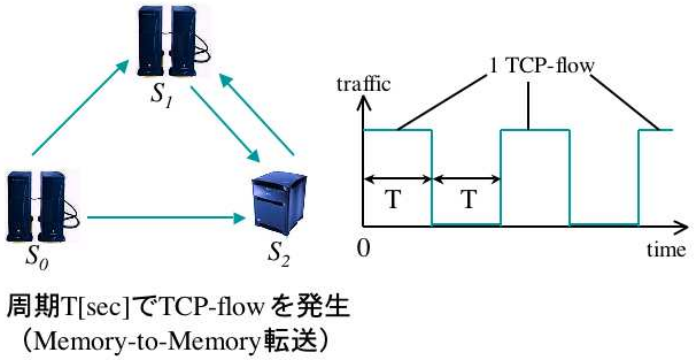


図 3.16　負荷の発生パターン

度が向上している．表 3.4 に示した全体とは，配布対象のデータを転送し始めてから最後のバイトを受信し終えるまでを指しており，T=30sec のときに 1.17 倍，T = 60sec のときに 1.22 倍の速度向上が見られた．

次に，パターン 2 において，S_1 が S_0 と S_2 から F_2 を並列転送したきの各サーバからの転送量，転送速度，および転送時間の推移を，図 3.17 から図 3.19 にそれぞれ示す．これらを見ると，T=15sec のとき，S_0 からの転送量と S_2 からの転送量が交互に入れ替わる形となっている．同じパターンが転送速度と転送時間に関しても見られることから，ブロックサイズが負荷の周期に対して大きすぎて追従できていないと考えられる．

逆に，T =30sec, 60sec になると，転送時間が一致しているのがわかる．この場合は，ブ

表 3.4 コンテンツ配布実験の結果

パターン	T	1 回目	2 回目	全体
1	15sec	270sec 32.0Mbit/s	272sec 31.8Mbit/s	542sec 31.9Mbit/s
2	15sec	296sec 29.2Mbit/s	236sec 36.6Mbit/s	532sec 32.5Mbit/s
1	30sec	280sec 30.9Mbit/s	265sec 32.6Mbit/s	545sec 31.7Mbit/s
2	30sec	280sec 30.9Mbit/s	175sec 49.4Mbit/s	465sec 37.2Mbit/s
1	60sec	288sec 30.0Mbit/s	274sec 31.6Mbit/s	562sec 30.8Mbit/s
2	60sec	294sec 32.0Mbit/s	166sec 52.0Mbit/s	460sec 37.6Mbit/s

ロックサイズが負荷の周期に対して十分小さいため，その変動に追従することが可能となり，1.5 倍から 1.6 倍の転送速度が得られたといえる．

この結果から，提案方式の効果を上げるには，負荷の変動周期に比例してブロックサイズを決定することが有効であると予測される．ただし，小さくし過ぎると，TCP のスロースタートアルゴリズム [65] の影響で，転送速度が最大になるずっと手前で転送が終わってしまい，コンテンツのサイズが大きくなるにつれて，通常の転送時の方が早く配布が終わるという状況になってしまう．

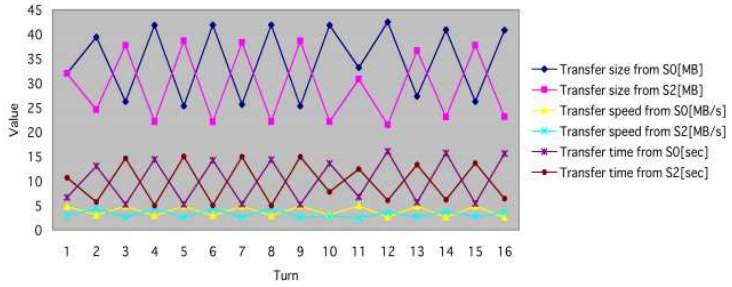


図 3.17　$T = 15sec$ のときの並列転送のトレース

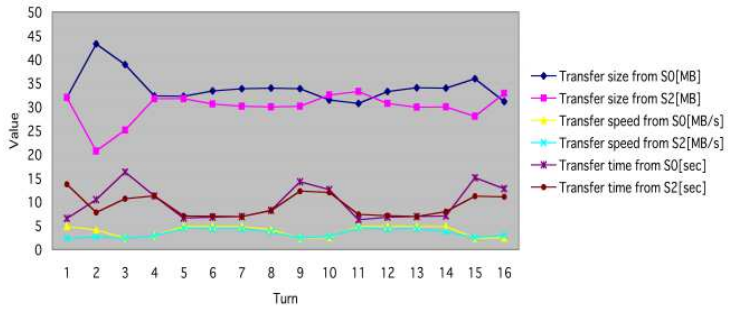


図 3.18　$T = 30sec$ のときの並列転送のトレース

3.4　コンテンツ配布方式の改善案

3.4.1　提案方式の課題

先の実験により，提案方式により転送効率が向上することが示されたが，ブロックサイズの導出など，より良い配布方式にするための課題も存在することがわかった．提案方式の課題として，以下のものが挙げられる．

1. 負荷トラヒックの特性や TCP の振る舞いによって最適なブロックサイズが変わってしまう．
2. 最初のストライピング時に転送速度の偏りがあった場合，最後の転送が終わり並列転送に移るまでの待ち時間が無駄使いとなる．
3. 転送前のファイル分割および転送後のファイル連結処理が必要であり，これらの処理がオーバーヘッドとなる．また，ディスクの負荷も増す．

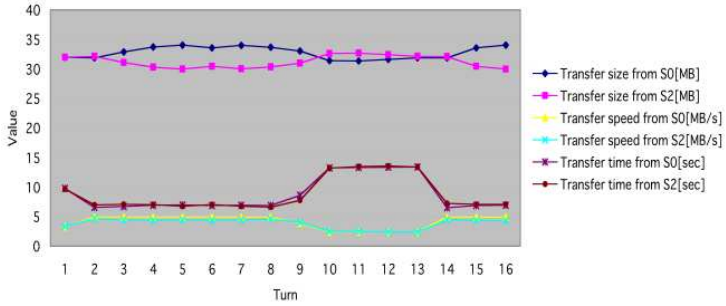


図 3.19　$T = 60sec$ のときの並列転送のトレース

実験では，オリジンサーバにすでに蓄積済みの複数ファイルを配布するという形態を取ったため，3 番目のオーバーヘッドを考慮せずに済んだ．しかし，配布対象のコンテンツの再生時間およびビットレートが増すと，それをサロゲートにおいても一つのファイルとして扱おうとすると，上記の処理回数が増えることになり，効率の低下と負荷の増大が懸念される．

3.4.2　改善案

提案方式の利点を残しつつ前述の課題に対処した改善案の試みとして，図 3.20 に示す配布方式を検討する．まず，オリジンサーバ S_0 に蓄積済みのコンテンツファイルを S_1 から S_N のサロゲートに配布することを考える．コンテンツファイルは，再生時間 T_f[sec]，ビットレート B_f[Mbit/s]，ファイルサイズ $S_f = B_f \times T_f$[bytes] であるとする．

始めに，S_0 においてファイルを開き，データを S_1 から S_N にユニキャスト転送する．このとき，各サロゲートへのパスの帯域幅が異なっているか，あるいは負荷の有無により，転送の完了時刻には一般にばらつきが生じる．先の提案方式では，このばらつきの大小に関係なく，すべてのサロゲートが転送を終えてから次のステップである並列転送に移ったため，この待ち時間が有効に使われていなかった．

そこで，転送が完了したサロゲート S_k は，オリジンサーバ S_0 にその旨を報告する．報告を受けた S_0 は，S_k 以外の残りのサロゲートの中から，最も受信量の少ないサロゲート S_{min} に対し，S_0 と S_k から残りのデータを並列に転送するよう指示する．以後，すべてのサロゲートが受信を完了するまでこの手続きを繰り返す．このように，帯域幅が狭いもしく

は負荷が大きいパスを通過しているサロゲートに対して既に転送が完了したサロゲートが加勢することで，空き時間を無くしかつ全体の転送速度を向上させることが期待できる．また，100%受信してから他のサロゲートへのデータ送信に移るため，煩雑なファイル処理から解放され，実装も簡易になる．

また，コンテンツデータがリアルタイムにオリジンサーバに入力されており，それを蓄積しながら同時にサロゲートにも配布したいときには，図 3.21 に示すように，ある一定のサイズごとにサロゲートへの配布を実行すれば，コンテンツ全体の保存を待たずに配布が始められる．

転送が完了したサロゲートが，まだ転送が完了していないサロゲートのうちどれに対して転送を支援するかについては，上記以外の選択肢が存在するため，どの手法が最も効率が良くなるか，今後こうした視点の検討を行う必要がある．

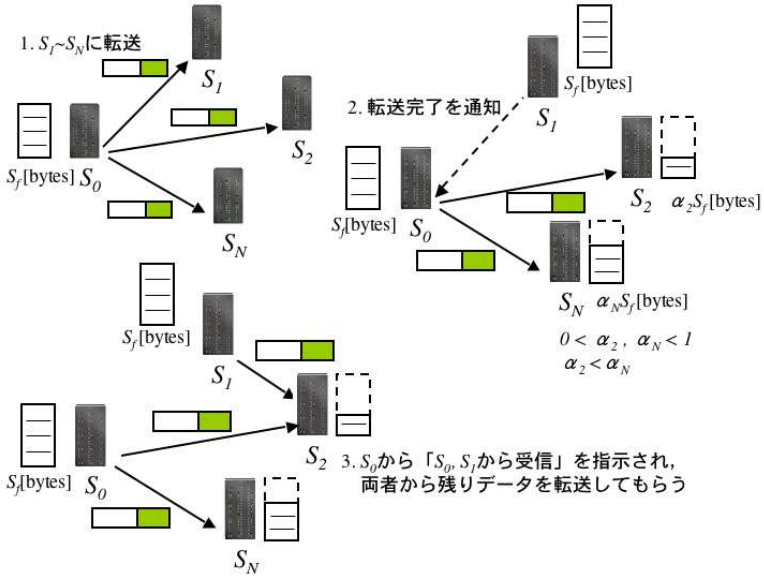


図 3.20　コンテンツ配布方式の改善案

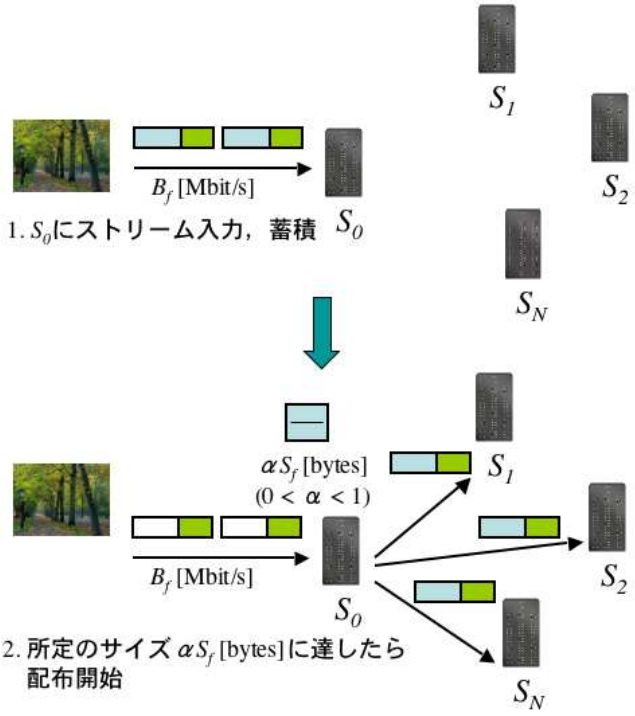


図 3.21　ライブコンテンツの配布

3.5　まとめ

本章では，第 2 章で提案した品質維持制御方式を，オリジンサーバからサロゲートへのコンテンツ配布に応用できることを述べ，従来のユニキャストおよび IP マルチキャストによる配布におけるボトルネックによる性能限界を打破し，負荷への耐性を高める効果があることを実証した．

また，実験結果から，提案方式の課題として，並列転送時の最適なブロックサイズの導出，ストライピング完了までの待ち時間が浪費されること，ファイル処理のオーバーヘッドがあることを明らかにした．改善策として，始めは通常のユニキャストによる転送を行うが，先に転送が完了したサロゲートが他のまだ転送が完了していないサロゲートに対する配信サーバとなり，オリジンサーバと並列にコンテンツデータを転送する方式を提案した．後者の方式は，蓄積済みのファイルを配布するときだけでなく，リアルタイムにコンテンツが投入されているときの配布にも適用することが可能である．

第 4 章

品質維持制御のための利用可能帯域通知法

第 2 章において，コンテンツをミラーリングあるいはストライピングして並列転送することでクライアントでの再生品質を維持できることを示したが，システム全体から見た負荷の偏りはそのまま残ってしまう．また，サーバやネットワークの障害時の対応も課題となる．本章では，再生品質を維持した上で，より効果的な負荷分散を実現するため，エンドホスト間の利用可能帯域をネットワーク側から通知してもらうことを提案し，その有効性について議論する．

4.1 利用可能帯域取得の必要性

第 2 章で示した並列転送方式は，クライアントから TCP によりコンテンツデータを分担して取得していた．これは，TCP にフロー制御機構が働いており，得られる転送速度が他の TCP フローと帯域幅を共有した結果であり，そのまま現在のサーバ・クライアント間の帯域幅とみなしてよいことが背景にある．このため，帯域幅を推定するためにコンテンツデータ以外の余分なトラヒックを流さずに済み，ネットワーク内を流れる他のトラヒックとも協調できる．しかし，第 2 章で提案した転送方式には，経路間の帯域利用率に偏りがあってもそれが考慮されず，システム全体で負荷が分散されない場合があるという課題がある．

例えば，図 4.1 に示すように，ビットレート 30Mbit/s の DV コンテンツを 2 台のサーバからクライアント C に配信するときに，S_1 から経路 P_1 は容量 100Mbit/s で利用率は

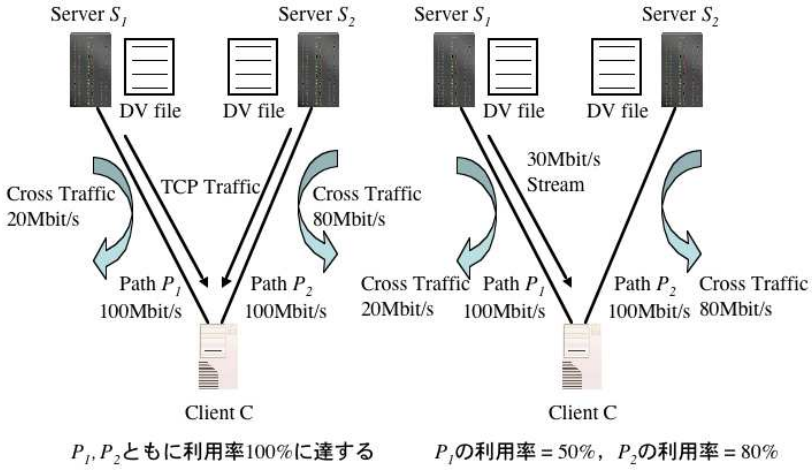


図 4.1　利用率の偏り

20%，S_2 からの経路 P_2 は容量 100Mbit/s で利用率は 80%であったする．このとき，ミラーリングおよびストライピングによる並列転送であれば，P_1, P_2 ともに利用率が 100%となる．

一方，S_1 から 30Mbit/s ですべてのコンテンツデータを転送したとすると，P_1 の利用率は 50%，P_2 の利用率は 80%のままとなり，ともに利用率に余裕が残る．このように，TCP は可能な限り帯域を使おうとするため，システムの負荷が上手に分散されるとは限らない．

もう一つのトランスポートプロトコルである UDP を使えば，アプリケーション側で送信レートを制御できるため，上記のように必要な分だけ帯域を使うといったことができる．ところが，UDP にはフロー制御機構がないため，そのまま送信すると，リンクを共有している TCP フローのスループットを抑圧してしまうことが知られている [66]．ではどれだけのレートで送信してよいか知ろうとすると，第 2 章で述べたようにオーバーヘッドが生じてしまう．

また，TCP 自身が持つ問題もある．TCP には様々な実装があるが，基本的には式 (4.1) に示す関係が成り立つことが知られている．

$$Throughput = \frac{WindowSize}{RTT} \tag{4.1}$$

この関係から，図 4.2 に示すように，ウィンドウサイズが経路の帯域幅と遅延の積（帯域幅

遅延積：Bandwidth Delay Produt:BDP）よりも小さいと，その帯域幅を使いきれないことがわかる．

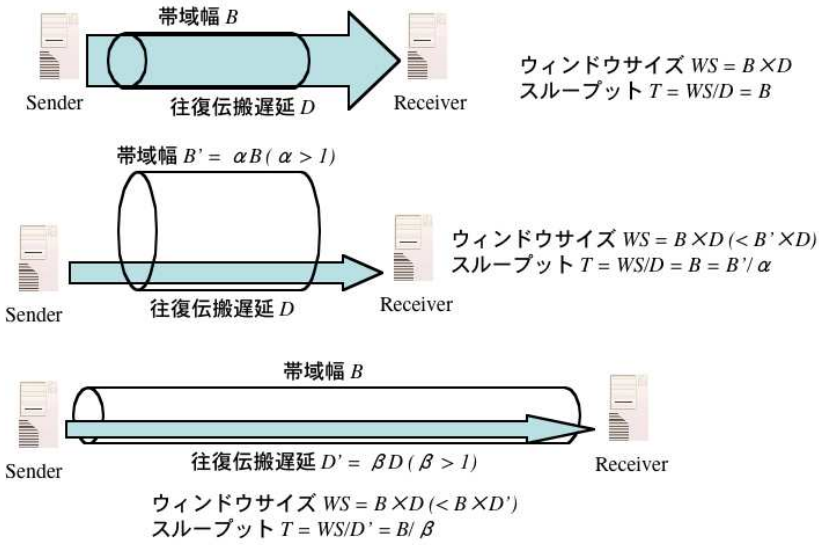


図 4.2　帯域幅遅延積の大きい通信路

今後，BDP とコンテンツのビットレートはますます大きくなると考えられるが，そうした場合，前述の利用率の問題とは逆に，ネットワークは十分な転送能力を持っているにも関わらず，TCP では必要な転送速度が得られないことが懸念される．

帯域幅 B[Mbit/s] のリンクを n 個の TCP フロー f_i が流れている場合，f_i のスループット TP_i[Mbit/s] は，すべてのフローのウィンドウサイズと RTT が等しい場合，

$$TP_i = \frac{B}{n} \quad (i = 1, 2, ..., n) \tag{4.2}$$

となる．しかし，実際には，ウィンドウサイズ・RTT ともフロー間で異なっているため，TP_i に格差が生じることが問題となっている．ゆえに，ある映像コンテンツが複数のクライアントに配信されるとき，再生品質が維持されるクライアントと，スループットが伸びずに再生が中断されるクライアントとに分かれるおそれがある．

こうした TCP の問題を克服しようとすべく，近年研究が盛んに行われている．BDP とスループットの問題については，[68] で述べられている．既存の TCP に替わる新たなトラ

ンスポートプロトコルも評価されている [69]．公平性の問題については，[70] などで議論されている．

整理すると，現在のトランスポートプロトコルを用いた場合，

1. TCP
   - BDP が小さい場合，最適な負荷分散が図れず，フロー間で不公平が生じる．
   - BDP が大きすぎると，再生品質を維持するのに必要な転送速度が得られない．
2. UDP
   - フロー制御がないため既存の TCP フローのスループットを抑圧する．輻輳しやすい．
   - 適切な送信レートを推定するには，余分なトラヒックと時間がかかりすぎる．

という問題を抱えている．

こうした問題を克服し，適切な負荷分散を行いながら個々のクライアントでの再生品質を維持することが，本研究の次の目標といえる．そのためには，「ネットワークはブラックボックスであり，パケットを転送するのみ，制御はエンドホストが行う」という従来のアーキテクチャを変えなければならない．そこで，「エンドホストからネットワークに対して，経路の利用可能帯域を問い合わせる」というモデルを考える．ここで適切な値を得ることができれば，利用する経路の利用率の差を最小にするといった制御目標に対して，どのサーバからどれだけコンテンツデータを転送すればよいか導くことが可能となる．

4.2 提案するモデル

4.2.1 利用可能帯域の定義

まず，この章で議論する「利用可能帯域」を定義する．図 4.3 に示すように，ホスト T_x からホスト R_x に向かうパケットが，$R_1, R_2, ..., R_N$ と N 個のルータを経由するものとする．R_i の出力リンク L_i の容量を C_i[Mbit/s]，利用率（Utilization）を $\rho_i (0 \leq \rho_i \leq 1)$ と

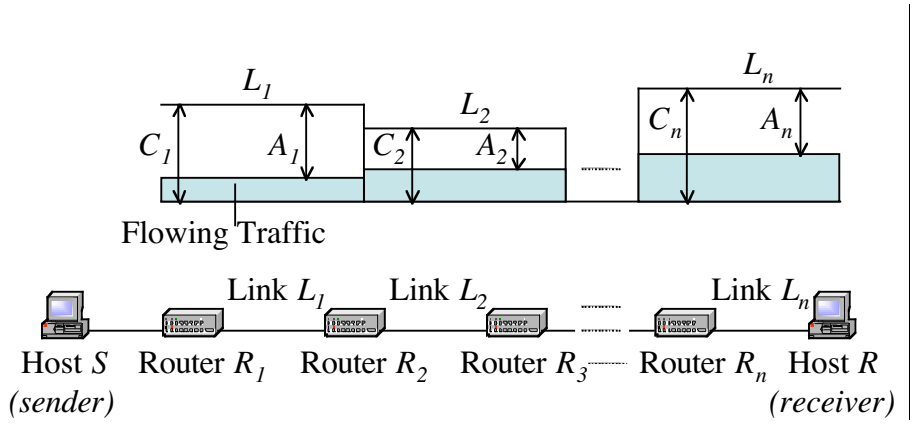


図 4.3 利用可能帯域の定義

すると，このリンクの利用可能帯域 A_i[Mbit/s] は，

$$A_i = C_i(1 - \rho_i) \tag{4.3}$$

ここで，L_i を通過しているトラヒック量を T_i[Mbit/s] とすると，$\rho_i = \frac{T_i}{C_i}$, $A_i = C_i - T_i$ となる．

このとき，T_x から R_x に向かう経路の利用可能帯域 A[Mbit/s] は，次式で定義される．

$$A = \min(A_1, A_2, ..., A_N) \tag{4.4}$$

4.2.2 既存の技術

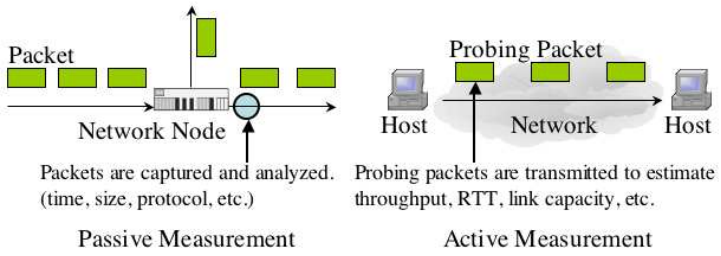


図 4.4 通信品質の測定方法

IP 網において通信品質を測定する技術は，大きく二つに分類される [71]．一つは，あ

る観測点においてそこを通過するパケットを収集・分析するもので，受動的計測（Passive Measurement）と呼ばれる．もう一つは，試験用のパケット群（プローブ: Probe）を送信し，その振る舞いから現在の品質を推定するもので，能動的計測（Active Measurement）と呼ばれる．これらの概要を図 4.4 に示す．前者の例としては，tcpdump[72], Ethereal[73], MRTG[74] などがあり，後者の例としては，ping, traceroute, Iperf といったものがある．他にも多くの実装が公開されており，無償で利用可能である [75].

利用可能帯域を求めるという点では，指定時間内に送信されたパケットのサイズと数からスループットを算出するのが最も簡単な手法である．S[Bytes] のパケットを T[sec] の間送信したとき，受信されたパケットの数が N 個であれば，スループット TP[Mbit/s] は，

$$TP = \frac{8NS}{T} \times 10^{-6} \tag{4.5}$$

となる．Iperf などのツールがこれに該当する．

これとは別の原理による測定ツールとして，pathload[*1][76] がある．pathload の測定原理を図 4.5 に示す．これは，送信ホスト T_x から N 個のパケット列を B_{snd}[Mbit/s] のレートで受信ホスト R_x まで送信し，R_x での受信レート B_{rcv}[Mbit/s] を求める．そして，利用可能帯域 A を次のように推定する．

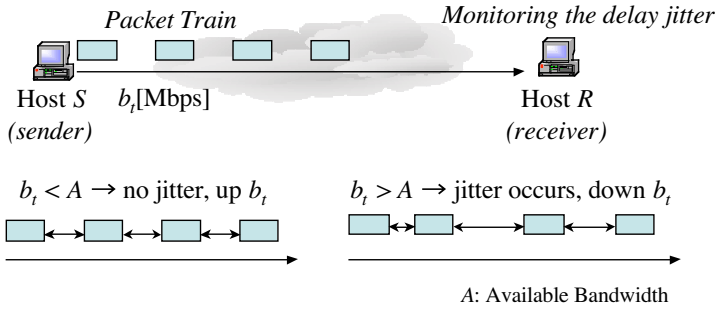


図 4.5　pathload の測定原理

[*1] http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathload.html から入手可能（2003 年 12 月現在）

- $B_{rcv} = B_{snd}$ ならば $A \geq B_{snd}$ とし，B_{snd} を上げる．
- B_{rcv}<B_{snd} ならば A<B_{snd} とし，B_{snd} を下げる．

これを，所定の収束誤差範囲内に収まるまで繰り返し．A の範囲を求める．

同様にパケット列を送信し，パケット同士の関係から利用可能帯域を推定する方法として，[77]-[79] が提案されている．これらはいずれも Active Measurement であり，エンドホストにプログラムを導入するだけで実行できる点が利点といえる．しかし，測定精度が他のトラヒックに影響されること，時間がかかること，プローブを実行するホストペアの増加による帯域の消費が大きくなることが欠点である．ゆえに，データ転送に時間の制約がある映像コンテンツの配信時に適用し，適切な負荷分散を行うには不向きである．

一方，Passvie Measurment により，ホスト間の利用可能帯域を求める方法として，図 4.6 に示すように，経路上のルータ R_i に対して SNMP により，通過するインタフェースのトラヒック量を取得することが考えられる．SNMP エージェントは事実上全ての既製品ルータに実装されており，エンドホストが SNMP マネージャを導入すればよい．ここで，前回の測定で得られた R_i のトラヒック量を $S_{old}(i)$[bytes]，それから T[sec] 後に再び取得したトラヒック量を $S_{new}(i)$[bytes] とすると，この経路の利用可能帯域 A は，次式により求められる．

$$A = \min(\frac{S_{new}(1) - S_{old}(1)}{T}, \frac{S_{new}(2) - S_{old}(2)}{T}, ..., \frac{S_{new}(N) - S_{old}(N)}{T}) \qquad (4.6)$$

しかし，この手法は以下の問題がある．まず，測定前にどのルータのどのインタフェースを通るか，traceroute 等によって把握する必要がある．これが測定時間を増やす原因となり，また，セキュリティの観点から，外部ネットワークから来た ICMP メッセージを廃棄するよう設定されたルータもあるため，常にどこを通るかわかるとは限らない．次に，ルータ数に比例して利用可能帯域を求めるのに発生するパケット（SNMP query と reply メッセージ）数も増えてしまう．さらに，SNMP は，本来ネットワーク機器の様々な情報を簡単に取得し，管理を容易にすることを目指したプロトコルであり，通常，ネットワーク管理者に

よってコミュニティ名と呼ばれるパスワードを管理対象機器に設定している．そうしたプライベートな情報を外部の者に教えることは好ましくない．

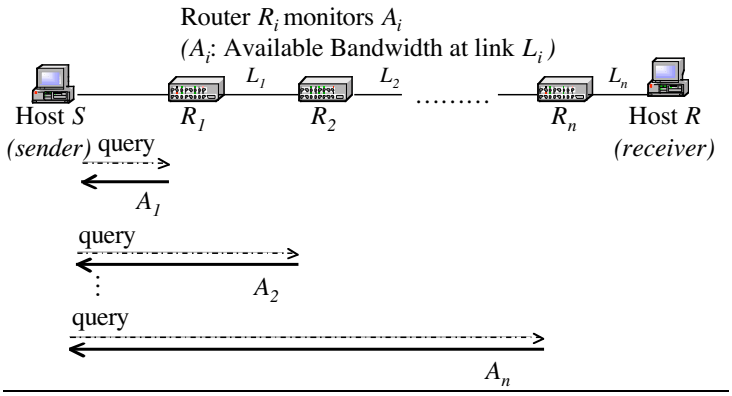


図 4.6 SNMP を用いた利用可能帯域の測定

4.2.3 提案方式の仕様

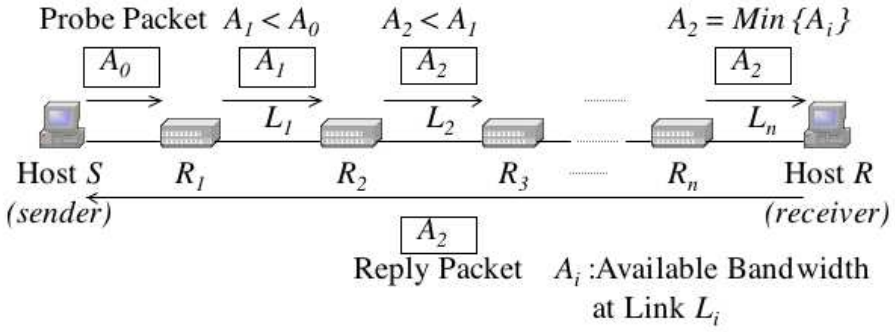


図 4.7 提案する利用可能帯域測定モデル

既存の Active Measurement 技術によりエンドホスト間の利用可能帯域を測定するには，ネットワークに与える負荷・推定値の精度・測定時間に問題があることがわかった．そこで，Active Measurement と Passive Measurement を組み合わせた形で軽負荷・高精度・低遅延の測定モデルを提案する．提案方式の動作概念を，図 4.7 に示す．

このモデルにおいて，送信ホスト T_x が受信ホスト R_x までの経路の利用可能帯域 A を知りたいとすると，次の手順により，A を取得する．

1. T_x が図 4.8 に示されたフォーマットのパケット（プローブパケット: Probe Packet）を 1 個 R_1 に送信する．このとき，プローブパケットの Source Address/Destination Address フィールドには T_x と R_x の IP アドレスを，Link Cpacity/Available Bandwidth フィールドには，T_x の出力インタフェースのリンク容量 C_0 と利用可能帯域 A_0 をそれぞれ記す．
2. R_1 がプローブパケットを受信すると，その宛先アドレスから次にパケットを転送するルータ R_2 を決定する．次に，その出力リンクの利用可能帯域 A_1 とプローブパケットに記された A_0 とを比較する．
3. もし，A_1 が A_0 を下回っていれば，プローブパケットの Available Bandwidth フィールドを A_1 に書き換え，Link Capacity フィールドも C_1 に書き換えた後，R_2 に転送する．A_1 が A_0 以上であれば，そのまま R_2 にプローブパケットを転送する．
4. 同様の処理を R_2 から R_N まで繰り返し，R_N から R_x にプローブパケットが送られる．
5. R_x は，プローブパケットの Source Address フィールドを参照し，T_x に結果を返信するパケット（リプライパケット: Reply Packet）を作成し，プローブパケットと同様のフォーマットで T_x に送信する．
6. T_x がリプライパケットを受信し，その Available Bandwidth フィールドに書かれた値が A となる．

これは，A が A_i の最小値と同値であるということを利用している．各ルータにおいて，それまでの A_i の最小値と現在の値を比較し，小さい方を残して次のルータに転送することで，R_x に到着した時点でプローブパケットに記された値が A となる．

ここで，プローブパケットの各フィールドを，次のように規定する．

1. IP バージョン（IP Version） 1byte

   IP のバージョン番号を記す．この値が 4 であれば，後述の送信元アドレスおよび宛先アドレスの長さは 4bytes となり，6 であれば，図 4.9 に示すように 16bytes となる．
2. ユニット（Unit） 1byte

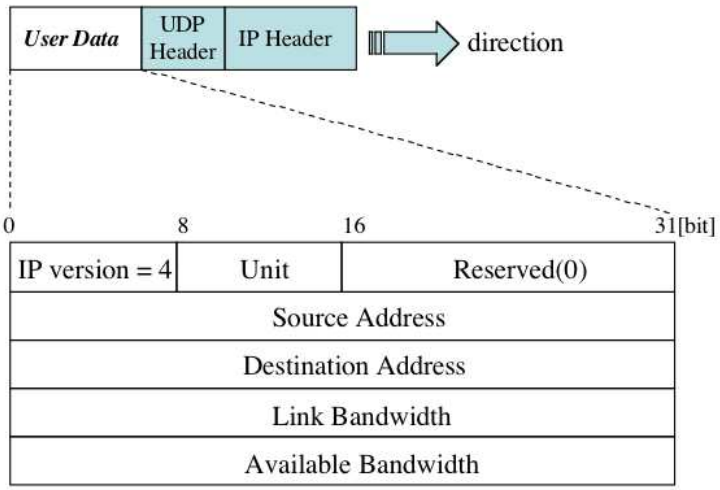


図 4.8 プローブパケットのフォーマット

このパケットに記述されるリンク帯域幅および利用可能帯域幅の単位を定める．このフィールドは，後で説明するように，帯域幅パラメータの指数を示す．

3. Reserved 2bytes

   将来の仕様拡張のための空き領域とする．

4. 送信元アドレス（Source Address）4bytes or 16bytes

   プローブパケットの送信元となる IP アドレスを記す．

5. 宛先アドレス（Destination Address）4bytes or 16bytes

   プローブパケットの宛先となる IP アドレスを記す．ルータは，ここを参照して転送先を決定する．

6. リンク帯域幅（Link Bandwidth）4bytes

   利用可能帯域幅フィールドを書き換えるとき，そこのリンク帯域幅をこのフィールドにも書くようにする．これにより，パスのボトルネックリンクの使用率を算出できる．フィールド長が 4bytes であるため，単純に扱えば 4Gbit/s までしか記述できない．このため，前述のユニットフィールドと組み合わせることで，任意の大きさの値を保持できるようにする．ユニットフィールドの値を U，リンク帯域幅および利用可能帯域幅フィールドに記された値を，それぞれ C, A とすると，実際のリンク帯域幅は，

$C \times 10^U$[bit/s]，利用可能帯域幅は $A \times 10^U$[bit/s] と解釈する．

7. 利用可能帯域幅（Available Bandwidth）4bytes

ルータにおいて求められた使用リンクの利用可能帯域幅が，このフィールドに記された値を下回っていれば，その値に書き換えてから次のルータに転送する．

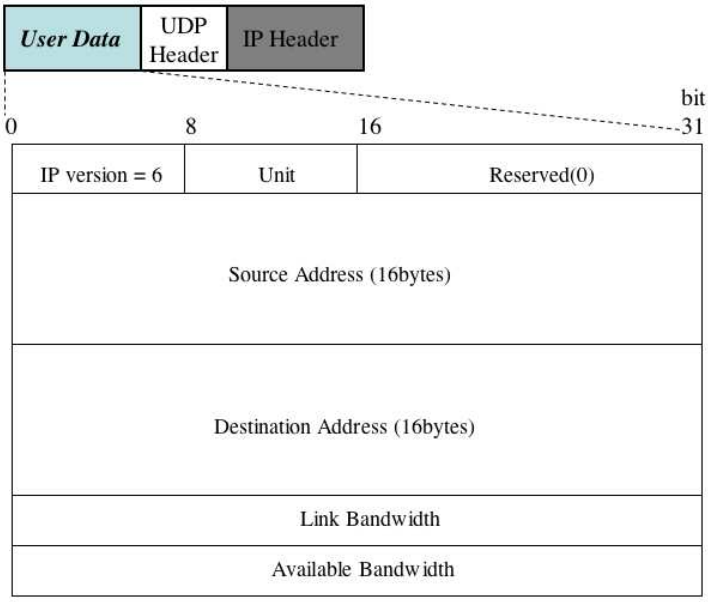


図 4.9　IPv6 ネットワーク下でのプローブパケットのフォーマット

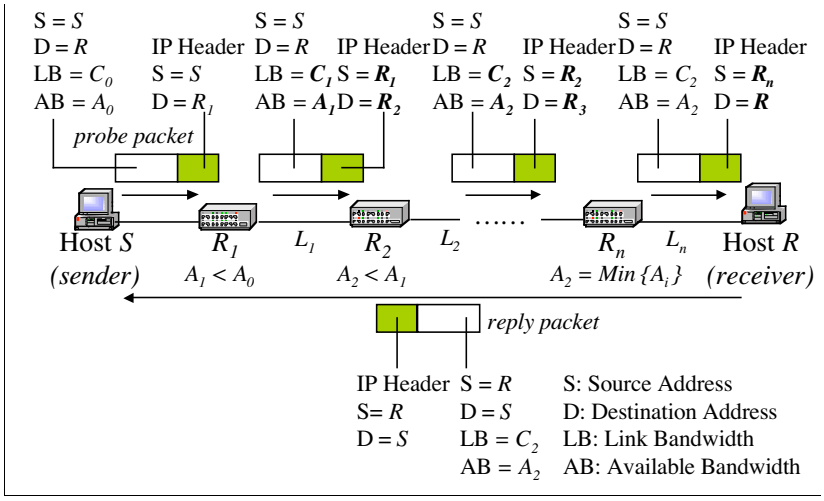


図 4.10　プローブパケット転送の詳細

このフォーマットでプローブパケットを転送する手順の詳細を，図 4.10 に示す．まず，送信ホスト S がプローブパケットを生成する．このとき，プローブパケットの送信元アドレスには自身の IP アドレス（以下単にアドレス）を，宛先アドレスには受信ホスト R の IP アドレスを記す．また，リンク容量，利用可能帯域は，プローブパケットを送信するネットワークインタフェースの値を書く．これらの値を，それぞれ C_0, A_0 とする．残りのフィールドにもそれぞれ対応する値をセットしたら，これを R に向かう最初のルータ R_1 に送信する．すなわち，プローブパケットに付与される IP ヘッダの送信元アドレスは S のアドレスに，宛先アドレスは R_1 のアドレスになる．

一方ルータでは，送信ホストと共通の UDP ポートを開いて，プローブパケットの到着を待つ．プローブパケットが R_1 に到着したら，R_1 は以下の処理を行う．

1. 宛先アドレスが自身のものと一致していれば，次の Reply Packet 返信を実行する．そうでなければ，宛先アドレスと自身の持つルーティングテーブルを参照して，次の転送先ルータ R_2 と出力インタフェース E_1 を決定する．
2. E_1 の利用可能帯域 A_1 を求め，Probe Packet に書かれてある A_0 と比較する．もし，A_1 が A_0 を下回っていれば，A_1 の値を利用可能帯域フィールドに上書きし，E_1 のリンク容量 C_1 もリンク容量フィールドに上書きする．その後，チェックサムを再計算する．A_1 が A_0 以上であれば，この処理は行わない．
3. IP ヘッダの送信元アドレスを R_1 に, 宛先アドレスを R_2 とし，R_2 に転送する．

この処理を，後続のルータ $R_2.R_3, ..., R_n$ において繰り返し，最終的に目的地である R まで届けれらる．その後，R からリプライパケットが S に返信される．

受信ホスト R は，プローブパケットを受信したら，宛先アドレスが自身のアドレスと一致しているか確認する．一致していれば，リプライパケットを生成する．リプライパケットの IP バージョン，単位はプローブパケットと同じものとし，送信元アドレスは R のアドレスに，宛先アドレスは S のアドレス（プローブパケットに書かれていた送信元アドレス）にセットする．

そして，自身の NIC の利用可能帯域がプローブパケットに書かれた値を下回っていれば，リプライパケットの利用可能帯域をその値にセットし，さらにリンク容量も NIC のものにセットする．そうでなければ，これらのフィールドはプローブパケットに書かれた値をそのまま移し，S に向かって送信する．このとき，リプライパケットの IP ヘッダにおける送信元アドレスには R のアドレスが，宛先アドレスには S のアドレスがセットされ，途中ルータでは，通常の転送処理が行われる．

4.2.4 利点

このモデルにより利用可能帯域を測定する利点を述べる．まず，経路の距離や種類に関係なく，往復 2 個のパケットで 1 回の測定が行えるため，これまでの Active Measurement で問題となったオーバーヘッドが無視できる．次に，A_i は R_i がモニタリングすることにより更新できるため，測定値の正確さが増す．さらに，測定に要する時間は，ルータにおける処理遅延が伝搬遅延に対して十分小さいと仮定すると，1-RTT ですむことになる．既存の技術では早くても数秒，遅いときには 30 秒ほどかかるのに比べると，映像コンテンツ配信時に適用するのに十分な性能といえる．

4.2.5 実装

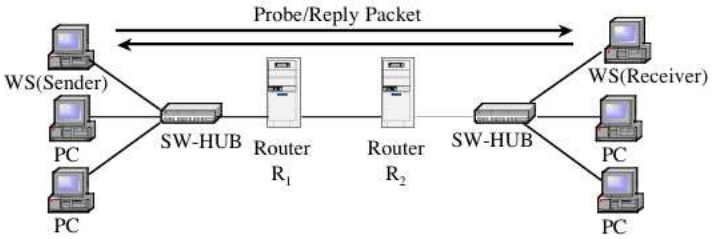


図 4.11 利用可能帯域測定方式の実験環境

この利用可能帯域測定方式を PC 上に実装し，精度と負荷および測定時間を評価した．試験に用いた環境を図 4.11 に示す．図中の Sender から Receiver に向かって 20bytes のプ

ローブパケットが送信され，同じく 20bytes のリプライパケットが Receiver から返される．両者の間には 2 台のルータ R_1, R_2 があり，それぞれプローブパケット転送処理を実行する．これらは，PC に NIC を追加してルータとして動作させている．同時に R_1, R_2 では，Net-SNMP[80] を利用して 1 秒ごとにインタフェースのトラヒック量を取得しており，インタフェースの利用可能帯域を更新している．

この環境において，Sender, Receiver と同じサブネットに PC をそれぞれ 1 台ずつ置き，一定レートのトラヒックを発生させる．以上の機器はすべて Fast Ethernet（リンク容量 100Mb/s）によって接続した．トラヒックの送信レートを 0Mbit/s から 80Mbit/s まで変えたときの Sender-Receiver 間の利用可能帯域の測定値と測定時間を表 4.1 に示す．また，プローブパケットと同じサイズの ICMP Echo Request メッセージを ping コマンドにより送信したときの RTT も合わせて示してある．提案方式による測定時間と ping RTT は，100 サンプルの平均値を取っている．このように，リンクの利用率の上下に関わらず，正確に利用可能帯域が取得できている．測定時間を見ると，同じパケットサイズの ping を送信したときと同じオーダーであり，迅速な処理が要求される映像配信時においても，提案手法が実用的であるといえる．

表 4.1 測定値と測定時間

負荷	測定値	測定時間	ping RTT
0Mbit/s	99-100Mbit/s	0.620ms	1.120ms
10Mbit/s	88-90Mbit/s	0.697ms	1.128ms
20Mbit/s	77-78Mbit/s	0.750ms	1.232ms
40Mbit/s	58-61Mbit/s	0.806ms	1.238ms
60Mbit/s	35-46Mbit/s	0.869ms	1.415ms
80Mbit/s	14-22Mbit/s	0.991ms	1.348ms

次に，プローブパケットが集中してルータに到着したときの処理負荷を見るため，プローブパケットを定められた間隔で連続して 180 秒間送信した．このときのルータにおける転

送処理プロセスの CPU 使用率（最大値）を表 4.2 に示す．1ms 間隔でプローブパケットを送信したときの CPU 使用率は，最大 3%となり，間隔なしで連続して送信した場合，最大 10.5%となった．この結果から，1000PPS(Packet Per Second) 以下のレートであれば，プローブパケット送信時のルータに与える負荷は無視できると予測できる．

ただし，リプライパケットを受信せずに，プローブパケットのみ間隔なしで連続して送信すると，プローブパケットの送信レートは 131kPPS となり，CPU 使用率は R_1 で最大 99.0%，R_2 で最大 55.0%となった．これは，もし悪意を持ったユーザが大量にプローブパケットを送信することで，ルータの処理を不能にしてしまう DoS(Denail of Services) 攻撃に繋がる危険性があることを示唆しており，この点を考慮したプローブパケット転送処理を検討する必要がある．

表 4.2 ルータの CPU 使用率（最大値，PPS は Packet Per Second）

送信間隔	送信レート	R_1 CPU 使用率	R_2 CPU 使用率
10ms	206PPS	0.10%	0.00%
1ms	1125PPS	3.22%	3.17%
0	3062PPS	10.5%	10.1%
0（Reply なし）	131kPPS	99.9%	55.0%

4.3 モデルの見直し

4.3.1 モニタリングサーバの導入

提案方式の有効性が先の実験により実証されたが，提案方式の欠点として，経路内のすべてのルータがこのプロトコルを実装し，プローブパケットを適切に処理しなければいけないという点がある．現在のインターネットでは，送信されたパケットは，数十個のルータを経由して目的のホストまで到達するのが普通であり，また，世界中で稼働している無数のルータが，一斉に新しいプロトコルを導入するのは無理である．実際，QoS 保証アーキテクチャ

の一つとして提案された IntServ は，同様の問題があり，普及していない．また，実験により，リプライパケットの受信をしないで大量のプローブパケットを送信することで，ネットワークの性能低下あるいは使用不能に陥るおそれがあることも明らかとなった．

こうした問題を考慮し，提案方式の利点を残しつつ，より導入を容易にするものとして，図 4.12 に示すモデルを提案する．先のモデルでは，ルータごとに A_i の更新とプローブパケットの処理を行っていたのに対し，図 4.12 のモデルでは，自律システム（AS: Autonomous System）ごとにモニタリングサーバ MS_k を設置し，これに処理を行わせる．モニタリングサーバの機能は次の二つである．

- 経路情報の収集とトラヒック量のモニタリング
- プローブパケットの転送処理

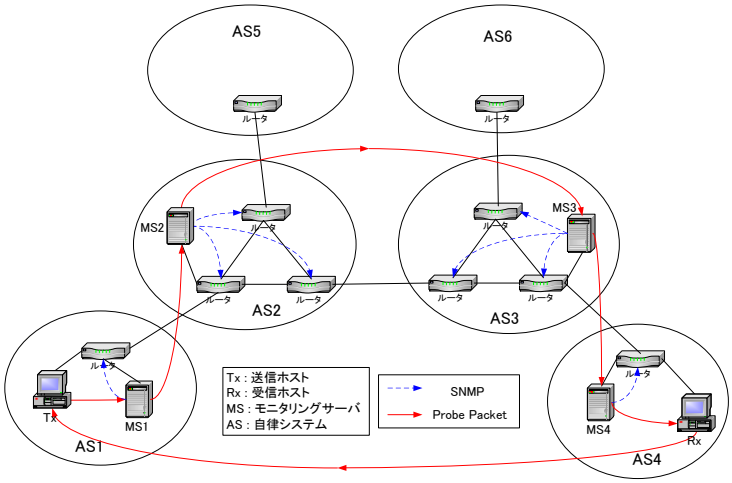


図 4.12　自律システムごとにモニタリングサーバを置いたモデル

4.3.2　経路情報の収集とトラヒック量のモニタリング

モニタリングサーバは，図 4.13 に示すように，自 AS にあるルータからインタフェース情報・ルーティングテーブルを取得し，そこから存在する経路（どのルータから入ってきてどのルータから出て行くか）を抽出する．続いて，経路を構成するルータのトラヒック量を

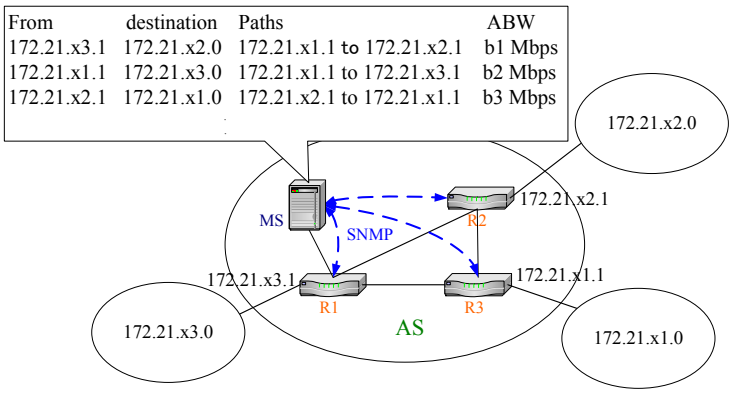


図 4.13　モニタリングサーバによる利用可能帯域の更新処理

定期的に取得し，経路の利用可能帯域を更新・保存する．ルーティングテーブルとトラヒック量の取得には，SNMP を使う．SNMP はほぼすべてのネットワーク機器が実装している標準的なプロトコルであり，本モデルの導入も容易となる．

具体的には，次のようにして情報の取得と更新を行う．

1. あらかじめ与えられた AS 内のルータのリストを読み込む．このリストには，表 4.3 に示された内容が含まれている．
2. 読み込んだルータのリストから，SNMP を使って各ルータのインタフェース数，インタフェース速度，ルーティングテーブルを収集する．
3. 収集した情報から，ルータ間のリンクの一覧を作成し，それぞれに ID を割り振る．この一覧に，リンク速度やトラヒック量，利用可能帯域も書き込まれる．
4. AS 内にある境界ルータ（隣接する AS と接続されたルータ）全てを 2 つずつ組み合わせたペアを両端をするパスの一覧を作成する．この一覧には，AS への入口となる境界ルータから出口となる境界ルータまでのルータ全ての IP アドレスと，その間にある全てのリンク ID が含まれる． また，ルーティングテーブルからそれぞれのパスを使う宛先 IP アドレス（ネットワークアドレスもしくはホストアドレス）を割り出して列挙する．
5. パスごとにリンクの一覧からそのパスに含まれるリンクの利用可能帯域を調べ，最小値

をそのパスの利用可能帯域とする．

6. 以後定期的に書くリンクのトラヒック量を SNMP で取得し，リンクの一覧とパスの一覧，利用可能帯域を更新する．

表 4.3 モニタリングサーバが読み込むルータのリスト内容

項目	説明
IP アドレス	SNMP での問い合わせに使う IP アドレス
コミュニティ名	SNMP のコミュニティ名
境界ルータ識別子	そのルータが隣接 AS との境界に位置するか
境界インタフェース IP アドレス	境界ルータである場合はその境界と成っているインタフェースの IP アドレス

4.3.3 プローブパケットの転送処理

送信ホスト T_x が受信ホスト R_x に向かう経路の利用可能帯域 A を取得するには，前述のモデルと同様に，図 4.14 に示すプローブパケットを定期的にモニタリングサーバに送信する．トランスポート層には UDP を使用し，ペイロード部分には次の情報を格納する．

- Tx の IP アドレス (T_x, 4bytes)
- R_X の IP アドレス（R_x, 4bytes）
- AS の入口となるルータの IP アドレス（ER, 4bytes）
- 通信開始，継続，終了を示すフラグ（F, 1byte）
- モニタリングサーバで書き込まれるパスの利用可能帯域（ABW, 4bytes）

ここでのプローブパケット転送の具体的な動作手順は，次のとおりである．

1. T_x から，T_x が属する AS にあるモニタリングサーバ MS にプローブパケットを送信する．このとき，プローブパケットの利用可能帯域フィールドには，T_x の送信インタ

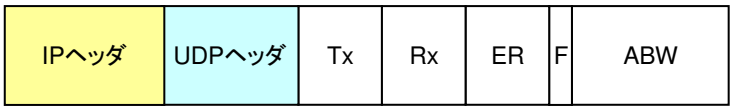


図 4.14　モニタリングサーバを経由するプローブパケットのフォーマット

フェースの利用可能帯域（b_0）を記す．MS_1 の IP アドレスは，あらかじめ T_x が知っているものとする．

2. MS は，手前のモニタリングサーバで書き込まれた自 AS の始点ルータの IP アドレスと R_x の IP アドレスを，モニタリングで作成したパスの一覧と照らし合わせて，どのパスを通過するか特定する．
3. 特定したパスの利用可能帯域（b_2）と b_0 を比較し，b_2 の方が小さければ b_2 をプローブパケットに書き込む．
4. R_x の IP アドレスを見て，プローブパケットが次に通過する AS の入口となるルータの IP アドレスをプローブパケットに書き込む．
5. プローブパケットを次の AS のモニタリングサーバに転送する．このとき，モニタリングサーバには，あらかじめ隣接する AS のモニタリングサーバに関する情報が登録されているものとする．
6. 2 から 5 までの処理を，プローブパケットが R_x の属する AS に到達するまで繰り返す．
7. R_x が属する AS のモニタリングサーバは，2 の処理を行った後，R_x にプローブパケットを転送する．
8. R_s は，受信したプローブパケットを参照し，T_x にパケットを返信する．
9. 返信されたパケットに記載された値が，T_x から R_x に至るパスの現在の利用可能帯域となる．

MS_1 は，プローブパケットの宛先アドレスから T_x と R_x との通信で使われる経路を特定し，その利用可能帯域 A_1 を参照する．そして，A_1 がプローブパケットに記された値 A_0 を下回っていれば書き換え，隣の AS のモニタリングサーバ MS_2 にプローブパケットを転送する．以下，図 4.15 に示すように，同様の処理を繰り返す．ここで，AS_k のモニタリング

サーバ MS_k は，隣接する AS のモニタリングサーバ MS_{k-1} と MS_{k+1} の所在を知っているものとする．

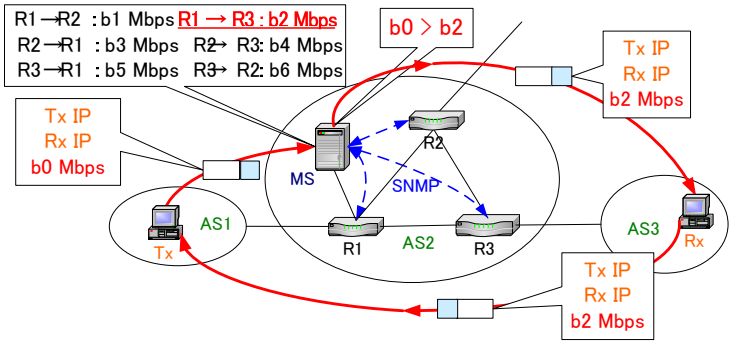


図 4.15　モニタリングサーバにおけるプローブパケット転送処理

4.3.4　改善点

このモデルにより，ルータは従来通りのパケット転送処理を行い，モニタリングサーバがエンドホストからのプローブパケット転送処理を行うため，ルータにかかる余分な負荷を軽減できる．発生するパケット数は任意の経路において往復 2 個で済み，迅速な応答時間が得られるという当初のモデルの利点は残しつつ，モニタリングサーバにより 1 つの AS をカバーすることで導入を容易にしている．さらに，ここでの処理はすべてアプリケーション層のものであり，下位のプロトコルは既存のままで良い．

セキュリティ対策については，モニタリングサーバを利用できるユーザをあらかじめ登録しておき，実際にプローブパケットが到着したときに認証処理を行うことが有効となる．モニタリングサーバが管理するのは自身の属する AS までであり，外部にプライベートな情報を流さなくてすむ．例えば，SNMP のコミュニティ名は，従来通り管理単位ごとに設定しておくことが可能である．もし不正に利用され，大量のプローブパケット送信によりモニタリングサーバが使用不能になったとしても，本来のルータを経由した経路は稼働しており，通常のパケット転送は行われる．

4.4 実験

続いて，モニタリングサーバを導入した利用可能帯域通知システムを実装し，通知帯域の正確さ，測定時間，実行時のシステムにかかる負荷を評価した．また，ネットワーク内を流れるトラヒック量が時間と共に変動した場合の通知帯域の追従性についても評価した．

4.4.1 実験環境

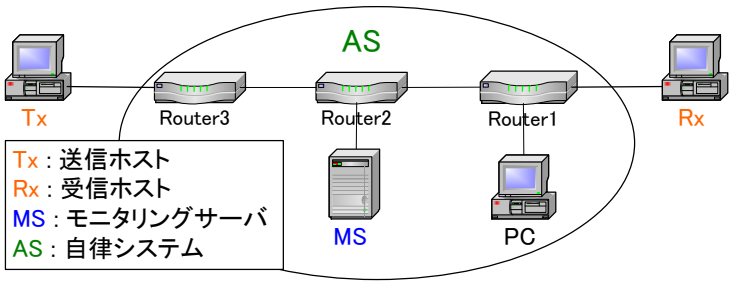


図 4.16 モニタリングサーバを導入した実験環境

構築した実験系を図 4.16 に，使用した機材の仕様を表 4.4 に示す．各ホストは，100Base-TX によって接続されている．モニタリングサーバにおけるルータ情報のモニタリング，プローブパケットの送信と転送，返信処理は，Perl スクリプトとして作成した．また，SNMP を使用する部分では，公開されている Perl モジュールの Net-SNMP 4.1.0[81] を組み込んだ．ルータには，SNMP デーモンとして Net-SNMP 5.1 が導入されている．

4.4.2 通知帯域の精度

最初に，Iperf により $T_x - R_x$ 間の利用可能帯域を測定した場合，pathload により測定した場合，そして提案方式によって利用可能帯域を通知してもらった場合に通知される帯域の精度を比較した．Iperf による計測では，送信ホスト，受信ホスト共に Red Hat Linux を使い，TCP スループットを求めた．pathload による計測では，現在公開されているバージョン 1.1 を使用し，同じく Red Hat Linux 上で実行した．

最初に，通知帯域の精度について述べる．ネットワーク上に他のトラヒックが流れていな

表 4.4 モニタリングサーバを導入した実験系の仕様

	T_x	MS	Router1
OS	Windows XP Red Hat Linux 9.0	Windows2000 Server	Fedora Core1
CPU	Celeron 750MHz	Celeron 2.5GHz	Celeron 2.5GHz
RAM	256MB	512MB	512MB
HDD	20GB	80GB	80GB
	Router2	Router3	R_x
OS	Red Hat Linux 9.0	Windows2000 Red Hat Linux 9.0	Windows XP Red Hat Linux 7.3
CPU	pentium-II 450MHz	Athlon 1.2GHz	Celeron 600MHz
RAM	192MB	512MB	256MB
HDD	12GB	12GB+20GB	10GB

い場合と．Iperf を用いてルータ 1 とルータ 3 との間で 30Mbit/s および 50Mbit/s の UDP トラヒックを発生させた場合における，各方式の通知帯域を求めた．トラヒックが流れていないときの結果を表 4.5 に，30Mbit/s および 50Mbit/s の UDP トラヒックが流れているときの結果を，表 4.6 と表 4.7 にそれぞれ示す．ここでは，5 回計測した結果とその平均を表している．

トラヒックが流れていないとき，提案方式では 100Mbit/s，Iperf および pathload では 93Mbit/s から 96Mbit/s の値が通知されている．これは，計測する仕組みの違いによるものと思われる．UDP フローが流れているとき，提案方式と Iperf では妥当な結果が得られているのに対し，pathload では上限と下限との間の差が大きくなっている．このため，pathload に固有の問題があると考えられる．

表 4.5 無負荷時の通知帯域（単位は Mbit/s）

	Iperf	pathlaod	提案方式
1	93.5	96.1-96.1	100
2	93.5	95.42-96.26	100
3	93.6	96.18-96.18	100
4	93.3	95.42-96.26	100
5	93.3	96.06-96.06	100
平均	93.836	95.836-96.172	100

表 4.6 負荷 30Mbit/s 時の通知帯域（単位は Mbit/s）

	Iperf	pathlaod	提案方式
1	64.4	38.5-74.83	67.68
2	64.1	36.82-36.82	68.15
3	63.8	37.5-84.5	68.15
4	69.4	37.95-57.79	68.15
5	65.2	38.84-63.11	67.68
平均	65.38	37.922-63.41	67.962

4.4.3 測定時間

次に，各方式において利用可能帯域を得るのに要した時間を，表 4.8 に示す．Iperf は，測定時間をオプションで指定できるが，ここではデフォルトの 10 秒間とした．ここでは，10 回の測定値とその平均を示している．提案方式では，T_x において Ethereal を起動し，キャプチャされた送受信パケットの時間間隔を測定時間とした．

1 回の測定にかかった時間は，pathload では平均 8.1 秒，提案方式では平均 3.1ms となり，提案方式の優位性が際立っている．提案方式は，プローブパケットが 1 往復するだけで

表 4.7 負荷 50Mbit/s 時の通知帯域（単位は Mbit/s）

	Iperf	pathlaod	提案方式
1	44.8	37.78-41.59	47.35
2	44.8	37.7-45.24	47.69
3	44.7	38.97-49.22	47.69
4	44.6	39.25-41.43	48.04
5	44.5	39.13-39.13	48.04
平均	44.68	38.566-43.322	47.762

あり，その往復伝搬遅延とモニタリングサーバでの処理時間の和が測定に要する時間と見なせる．この実験系の $T_x - R_x$ 間の RTT を求めたところ，平均 0.7ms 程度であった．ゆえに，これを指し引いた 2.4ms が，モニタリングサーバでの処理時間といえる．実際のネットッワークでは，これがモニタリングサーバの数だけ整数倍された時間がかかると考えられる．ただし，モニタリングサーバの実装に改良の余地が多くあり，今後この処理時間は短縮できる可能性が高い．

4.4.4 システムにかかる負荷

続いて，提案方式を実行したときのモニタリングサーバ，エンドホスト，ルータおよびネットワークにかかる負荷を調べた．モニタリタングサーバにかかる負荷は，OS に付属しているシステムモニタにより計測した．1 秒あたりにモニタリングサーバに到着するプローブパケットの数を 1, 2, 10, 20, 100 個としたときのモニタリングサーバの CPU 使用率は，図 4.17 に示すようになった．プローブ間隔が 0.01 秒になると CPU 使用率が 25%以上を記録しており，多くのエンドホストが短時間に集中してプローブを実行すると，モニタリングサーバに高い負荷がかかるものと考えられる．今後，実装の見直しや，モニタリングサーバの増設といった対策が必要と言える．

表 4.8　測定時間（単位は sec）

	Iperf	pathlaod	提案方式
1	10	6.74	0.002977
2	10	9.15	0.003016
3	10	7.82	0.003083
4	10	9.16	0.003086
5	10	10.73	0.003127
6	10	7.35	0.003138
7	10	10.87	0.003169
8	10	6.75	0.003181
9	10	8.82	0.003184
10	10	4.00	0.03218
平均	10	8.139	0.0031179

次に，各方式を実行しているときエンドホストにかかる CPU 使用率を，表 4.9 に示す．ここでは，10 回の計測結果とその平均値を表している．Iperf を実行したときには 50%前後の CPU 使用率となっており，pahtload を実行したときには平均で 63%となった．これに対

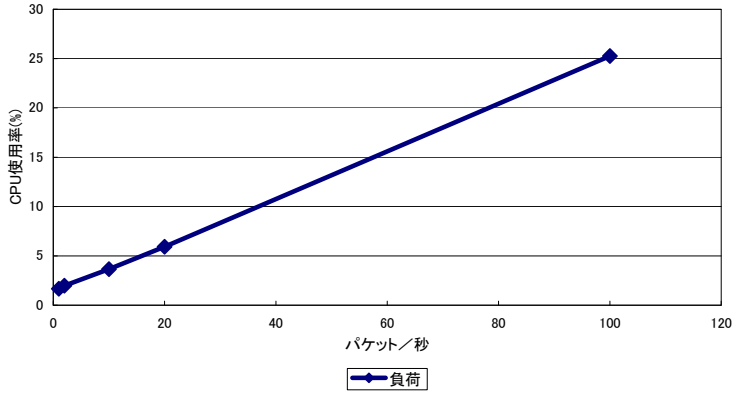


図 4.17　モニタリングサーバにかかった負荷

し，提案方式を実行したときには，送信ホスト T_x および受信ホスト R_x とも 1%から 5%という他のプロセスに比べて無視できるほどの負荷しかかかっていない．

表 4.9 エンドホストの CPU 使用率（単位は%）

	Iperf	pathlaod	提案方式 (T_x)	提案方式（R_x）
1	44	65	1	2
2	45	66	4	2
3	50	67	2	2
4	50	65	2	2
5	52	67	3	2
6	52	47	1	2
7	57	56	4	2
8	52	68	1	2
9	47	66	5	3
10	48	66	1	2
平均	48.96	63.3	2.4	2.1

実験系の 3 台のルータのうち，ルータ 1 における利用可能帯域測定中の CPU 使用率は，表 4.10 に示すように，Iperf では平均 32.2%となったのに対し，pathload では平均 4.8%，提案方式では平均 1.3%という低い値となった．Iperf は，出来る限り多くのパケットを送信しようとするため，ルータにかかる負荷も大きくなっている．

合わせて，各方式によってネットワークに送出されるパケット数を比較したものを，表 4.11 に示す．ここでの値は，T_x において Ethereal によってキャプチャされたパケットの数となっており，5 回計測した．Iperf，pahtload では数 1000 個以上のパケットが発生しているのに対し，提案方式では，その仕様から常に 2 個のパケットしかネットワークに送出されないため，他方式に対して大きなアドバンテージとなることがわかる．

表 4.10 ルータの CPU 使用率（単位は%）

	Iperf	pathlaod	提案方式
1	30	6	2
2	26	7	1
3	37	8	1
4	40	4	1
5	29	5	3
6	35	7	1
7	38	3	1
8	27	3	1
9	30	2	1
10	24	3	1
平均	32.3	4.8	1.3

4.4.5 トラヒック変動に対する通知帯域の追従性

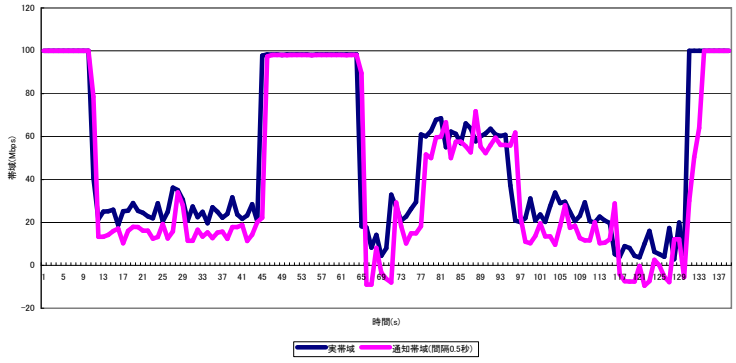


図 4.18 トラヒック変動時の通知帯域（プローブ間隔 0.5 秒）

最後に，提案方式によって $T_x - R_x$ 間の利用可能帯域をプローブしている間にトラヒック量が変動したとき，その変化に通知帯域がどれだけ追従することができるか調べた．具体的には， 提案方式により利用可能帯域をプローブし続け，その間 60 秒にわたって Iperf によ

表 4.11　ネットワークに送出されたパケット数

	Iperf	pathlaod	提案方式
1	58354	4420	2
2	63416	7166	2
3	65097	6005	2
4	71254	10893	2
5	67833	9563	2
平均	65191	7609	2

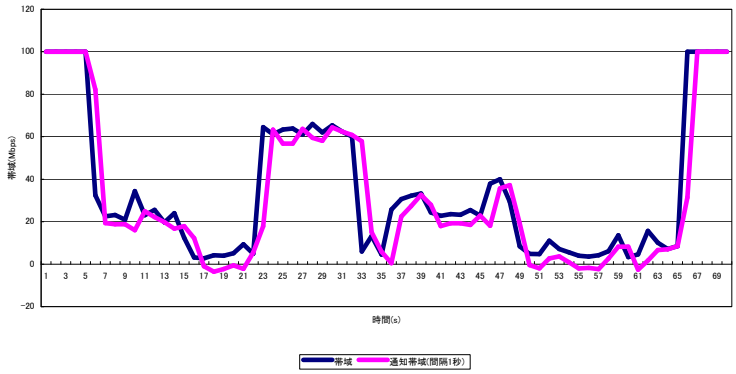


図 4.19　トラヒック変動時の通知帯域（プローブ間隔 1 秒）

る複数の TCP フローをランダムに発生させる．

プローブ間隔およびモニタリング間隔を 0.5 秒，1 秒，2 秒としたときの通知帯域の推移を，図 4.18，図 4.19 および図 4.20 にそれぞれ示す．これらのグラフは，1 分間ランダムに発生させた TCP フローによって上下した実際の利用可能帯域とそのときの通知帯域を，プローブ間隔ごとにプロットしている．これらの結果から，常に細かく変動する TCP トラヒックに対しておおむね追従できていることがわかる．ただし，プローブ間隔が 1 秒になると，通知帯域が実際の帯域に対して遅れることがわかる．

また，利用可能帯域が 10Mbit/s を下回るという高負荷な状態では，通知帯域がマイナスになっている．これは，ルータでのバイトカウントに実際の値とのずれが生じている可能性

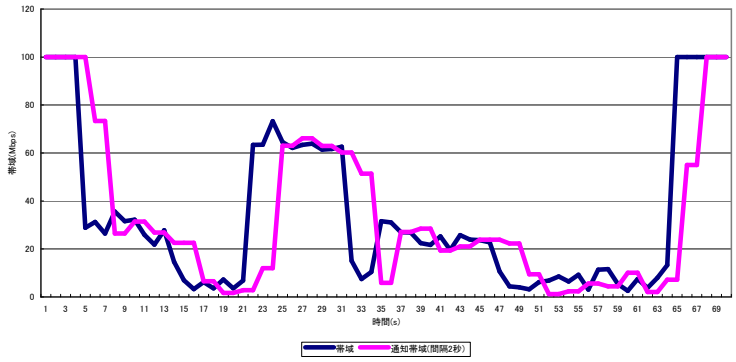


図 4.20　トラヒック変動時の通知帯域（プローブ間隔 2 秒）

があり，詳しい原因については，現在調査中である．ただ，使用率 90%というかなり高い状態でのみ起こる現象であり，実際の運用時には，利用可能帯域が 0 と通知して差し支えないと思われる．

4.5　まとめ

ここでは，再生品質を維持しつつより効果的な負荷分散を図るために必要な，エンドホスト間の利用可能帯域を取得する技術について議論した．ここ数年，Active Measurment によって利用可能帯域を推定する研究が盛んになりつつあるが，測定時間やシステムに与える負荷を考慮すると，映像配信制御にそのまま利用するのはまだ難しいと言える．

これに対して本研究では，高い精度で迅速にかつ最小限の負荷で利用可能帯域を得る方法として，それが各リンクの最小値で決まることに着目したプローブの仕組みを提案した．モニタリングサーバをネットワークに導入し，隣接する AS 間でプローブパケットを転送することで，通知帯域・処理時間・負荷が大幅に改善され，帯域変動時の追従性にも優れた方式であることを実証した．

第 5 章

結論

ここまで，第 1 章で本研究の技術背景を俯瞰し，第 2 章から第 4 章において，CDN における映像配信時および配布時の品質維持方式を提案し，その有効性を実証した．この章では，本研究の成果と意義を整理し，今後の課題を述べ，本論文の結論とする．

始めに，本研究によって得られた成果をまとめると，配信制御技術として，従来の「最適なサーバを選択する」から一歩進んで「配信期間中の品質を維持する」という方向を示すことができた．これは，過去に発表された研究にはない観点であり，本研究の新規性を示している．コンテンツ配布についても，「どのファイルをどこに転送するかが決まったして，ではどうすればそれをできるだけ高い品質で実行できるか」という新たな視点を持ち込んだ．そして，分散させてフローを転送することで，同報ベースから生じる一本のボトルネックによって全体の性能が制限されるという問題点が克服されることを実証できた．さらに，個々のクライアントに対して最善の品質を提供しつつシステム全体の負荷分散を最適化することを目指して，パスの利用可能帯域を通知する機構を提示した．これは，既存の Active Measurement と Passive Measurement を組み合わせることにより，両者の欠点を補っている．この方向性にも，新規性があると言える．

本論文では，提案方式を汎用 PC および UNIX サーバに実装し，実ネットワークを用いて有効性を示している．いずれも，特別なハードウェアあるいはソフトウェアは必要としておらず，Apache, libcurl, fwcontrol, Net-SNMP など既存のオープンソースプログラムを利用している．また，実験では DV 映像をサンプルに選んだが，提案方式そのものは特定のコーデックには依存しておらず，他の環境への移植性，プログラムの保守性にも優れている．

今後，ますます広帯域化が進み，配信されるコンテンツのビットレートも上がるにつれて，

エンドホスト間に要求される品質も厳しくなることが予測される．それに伴い，本研究で提案した分散型のコンテンツ配信の意義が増すものと考える．合わせて，利用可能帯域を得て映像配信に効果的に応用することも重要になるであろう．

次に，本研究の波及性について考えると，業務での利用という点では，地上波放送の設備を有している既存放送事業，IP 網での映像配信事業において期待が持てる．地上波放送は，2003 年 12 月から 3 大都市圏においてディジタル放送が開始され，2010 年の完全ディジタル化に向けて設備の更新が進んでいる．これを機に，例えば，キー局で制作されたコンテンツを系列各局に配布する，配信事業者あるいは組織間での提携によりコンテンツの配置範囲を広げてより多くの視聴者を得る，といった応用が考えられる．IP 網での映像配信事業も，コンピュータの進歩とネットワーク技術の発展という恩恵に授かり，既存メディアにはない多様なコンテンツを比較的容易に提供できるとされていたが，品質の点で問題があった．しかし，完全に品質が保証されない現在の IP 網をインフラにした場合でも，本研究のような品質維持技術が発展することで，それを克服することが期待できる．

もう一つ，個人での映像利用という点で考えると，最近の技術動向として，以下の要素が今後大きな影響を与えると思われる．

1. ハードディスクの大容量化と低価格化が進んだことで，ハードディスク搭載型のビデオレコーダや，PC に TV チューナを搭載することで，長時間の映像コンテンツを記録することが容易にできるようになった．
2. ネットワーク機器および汎用 OS の IPv6 への対応が進み，FTTH など上下対称の高速なアクセス系が普及している．
3. 集中管理するサーバを持たずにユーザ端末同士が直接ファイルを交換する Peer-to-Peer(P2P) 技術が広まった．

こうした点から，ユーザ端末同士で直接映像コンテンツの送受信を行うというモデルが考えられる．実際，従来のファイル交換に加えて，最近では PeerCast[82] やシェアキャスト [83] といった P2P 技術に基づいた AV ストリーム配信アプリケーションも登場し，研究も

されている [84]．本論文では，対等な立場の複数のサーバが連携してユーザ端末にコンテンツを提供するというモデルを取ったが，提案方式自体は，複数のサーバ間であっても複数のユーザ端末間であっても適用可能であり，汎用性があると言える．

最後に，本研究の今後の課題として，以下のものが挙げられる．

- 第 2 章で示したミラーリングによる並列転送時のブロックサイズ，ストライピングによる並列転送時の分割単位とネットワーク品質，再生品質との関係を導く．
- ネットワークの規模が大きくなったとき，並列転送するサーバ数が増えたとき，クライアントが増えたときの上記提案方式の有効性を検証する．
- DVCPRO や D1，さらに HD などより高いビットレートの映像コンテンツを配信するときの上記提案方式の有効性を検証する．
- 第 3 章で示したコンテンツ配布方式の改善案を吟味し，先に実験した方式や既存技術に対する優位性の有無を評価する．
- 第 4 章で示した利用可能帯域通知法において，複数のホストペアがリンクを共有しているとき，各送信ホストがどのようにして送信レートを決定すれば輻輳せずに帯域幅を最大限使えるか，検討する．
- 上記提案システムにおいて，どういったアルゴリズムに従ってどのサーバからどれだけのレートで送信してもらうか吟味し，それがユーザから見たときの品質維持およびシステム全体の負荷分散に寄与できるかどうか評価する．

謝辞

本研究を進めるにあたり，博士前期課程在籍時を含めた 5 年間叱咤激励していただき，指導教員および論文審査の主査として，常に有益なご助言ご尽力を賜った島村和典教授に心から感謝致します．本大学院の 1 期生として入学して以来，まさにゼロからの出発であり戸惑うことも多々ありましたが，研究のことのみならず，大学院生活全般に渡ってご指導を賜りました．改めてありがとうございました．

岡田守教授，木村義政教授，清水明宏教授，岩田誠教授には，副査として，研究に対する有益なご助言をいただき，また研究のみならず，TA や学内行事等においても大変お世話になりました．お礼申し上げます．

さらに，情報システム工学コースの他の教員の皆様，寺田浩詔教授，坂本明雄教授，篠森敬三教授，菊池豊助教授，福本昌弘助教授，浜村昌則講師，任向実講師，妻鳥貴彦講師，大森洋一助手，水科晴樹教育講師，ならびに秘書室の村上加織さん，片山さおりさんにもいろいろご支援いただきました．感謝申し上げます．

本大学院入学と時を同じくして発足した通信・放送機構 高知通信トラヒックリサーチセンターの加藤寛治研究員，高松希匠研究員，林秀樹研究員，神田敏克研究フェロー，同事務局の楠瀬智佐さん，和田典子さんには，研究者あるいは人生の先輩として，大きな力をいただきました．お礼申し上げます．

同じく幕張ギガビットリサーチセンターの榎本正研究員，小倉孝夫研究員，石丸勝洋研究員，鈴木純司研究フェロー，岡山情報通信研究開発支援センターの古川繁雄研究指導員には，実験系の構築と設備の利用を快諾していただきました．また，本社研究推進部の皆様には，追加機材の調達へのご理解とご尽力をいただきました．ここに感謝致します．

また，大学院の同期生として，故 井上富幸さん，橋本学さんのお二人には，博士前期課程時代から様々な面でお世話になりました．短い年数でしたが，密度の濃い有意義な時間を過ごすことができたことを嬉しく思います．ありがとうございました．

日頃の研究活動の拠点となった島村研究室では，博士後期課程の山口巧さん，同前期課程の中平和友君，谷岡亮介君，三谷乙弘君，山田敦君の諸氏にお世話になりました．ありがとうございました．特に，中平君と学部生の小松義幸君，都築洋平君には，日頃から研究の推進にあたって議論を重ね，その進展に貢献してくれました．皆さんを始め，島村研究室の現役生ならびに卒業生の皆さん，そして研究室の長である島村教授と有益な時間を過ごせたことに，深く感謝致します．

参考文献

[1] Internet Software Consortium, http://www.isc.org/ds/hosts.html

[2] NUA Internet Surveys, http://www.nua.ie/surveys/

[3] Kevin Savetz, Neil Randall and Yves Lepage, “MBONE: Multicasting Tomorrow’s Internet,” John Wiley & Sons Inc, Mar., 1996.

[4] NSPIXP, http://nspixp.wide.ad.jp/

[5] 総務省，“情報通信白書平成 15 年版，” http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/h15/index.html

[6] 大橋 正良，“IMT-2000 インターネットサービス，”情報処理， Vol.40, No.3, pp.299-302, Mar., 1999.

[7] Stephen Saunders 著，井早 優子，米沢 寿員，林田 朋之訳，“ギガビットイーサネット徹底解説，”日経 BP 社，1999 年.

[8] 石田 修，瀬戸 康一郎 監修，“IDG 情報通信シリーズ: 10 ギガビット Ethernet 教科書，”IDG ジャパン，2002 年.

[9] 松江 英明，守倉 正博 監修，“IDG 情報通信シリーズ: 802.11 高速無線 LAN 教科書，”IDG ジャパン，2003 年.

[10] Paul Izzo, “Gigabit Networks,” John Wiley & Sons Inc, Mar., 1996.

[11] ITU Recommendataion BT.601, “Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios,” Oct., 1995.

[12] 越智 宏，黒田 英夫，“JPEG&MPEG 図解でわかる画像圧縮技術，”日本実業出版社，1999 年.

[13] 映像情報メディア学会 編，“総合マルチメディア選書 MPEG，”オーム社，1996 年.

[14] 三木 弼一，“MPEG - 4 のすべて－多彩な映像、音声を自在に符号化する，”工業調査会，1998 年.

[15] 小野 定康，鈴木 純司，“わかりやすい JPEG2000 の技術，”オーム社，2003 年.

[16] 小野 文孝，渡辺 裕，“高度映像技術シリーズ 1: 国際標準画像符号化の基礎技術，”コロナ社，1998 年.

[17] SMPTE 306M, “6.35-mm type D-7 component format - video compression at 25Mb/s -525/60 and 625/50,” 2002.

[18] SMPTE 314M, “Data structure for DV-based audio and compressed video 25 and 50Mb/s,” 1999.

[19] 笠原 久嗣，“マルチメディア解説シリーズ 6 マルチメディアビデオオンデマンド，”昭晃堂，1999 年.

[20] Microsoft Windows Media 9 Series, http://www.microsoft.com/japan/windows/windowsmedia/default.aspx

[21] Real Networks, http://www.jp.realnetworks.com/

[22] Apple Computer, http://www.apple.co.jp/quicktime/index.htm

[23] Craig Partridge 著，樊 瑞雪，吉井 考伸，増本 健三訳，“ギガビットネットワーク—超高速マルチメディアネットワークの基礎と応用，”ソフトバンク，1995 年.

[24] Steven McCanne and Van Jacobson, “vic: A Flexible Framework Framework for Packet Video,” ACM Multimedia 95, Nov., 1995.

[25] LBNL’s Network Research Group, http://www-nrg.ee.lbl.gov/

[26] Akimichi Ogawa, Katsushi Kobayashi, Kazunori Sugiura, Osamu Nakamura and Jun Murai, “Design and Implementation of DV based video over RTP,” Packet Video Workshop 2000, APr., 2000.

[27] K. Kobayashi, A. Ogawa, S. Casner and C. Bormann, “RTP Payload Format for DV (IEC 61834) Video,” IETF Request for Comments 3189, Jan., 2002.

[28] K. Kobayashi, A. Ogawa, S. Casner and C. Bormann, ”RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio,” IETF Request for Comments 3190, Jan., 2002.

[29] 砂原 秀樹，知念 賢一，中田 秀基，松岡 聡，後藤 滋樹，“岩波講座インターネット 4 ネットワークアプリケーション，”岩波書店，2003 年.

[30] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” IETF Request for Comments 1889, Jan., 1996.

[31] 村山 公保，西田 佳史，尾家 裕二，“岩波講座インターネット 3 トランスポートプロトコル，”岩波書店，2001 年.

[32] Martn J. Riley and Iain E. G. Richardson, “Digital Video Communications,” Artech House Publishers, 1997.

[33] J. Wroclawski, “Specification of the Controlled-Load Network Element Service,” IETF Request for Comments 2211, Sep., 1997.

[34] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, “An Architecture for Differentiated Services,” Dec., 1998.

[35] E. Rosen, A. Viswanathan and R. Callon, “Multiprotocol Label Switching Architecture,” IETF Request for Comments 3031, Jan., 2001.

[36] Uyless D. Black, “MPLS and Label Switching Networks,” Prentice Hall, 2000.

[37] Bruce Daive and Yakov Rekhter 著，トップスタジオ 訳，池尻 雄一 監修，“MPLS 入門，”翔泳社，2002 年.

[38] 村田 正幸，山口 英，塚本 昌彦，塚田 晃司，星 徹，下條 真司，佐藤 哲司，名和 小太郎，篠崎 彰彦，尾家裕二，“岩波講座インターネット 6 社会基盤としてのインターネット，”岩波書店，2001 年.

[39] 田代秀一，西角直樹，西木健哉，小島富彦，神原顕文，“インターネットコンテンツ配送技術の最新動向，” 情報処理，vol.42, no.11, pp.1082-1091, Nov. 2001.

[40] Tony Bourke 著，鍋島 公章，横山 晴庸，上谷 一 訳，“サーバ負荷分散技術，”オライリージャパン，2001 年.

[41] Kyung-Ah AHN,Hoon CHOI and Won-Ok KIM, “Architecture of a VOD System

with Proxy Servers,” IEICE Trans. Commun. Vol.E83-B, No.4, pp.850-857, Apr., 2000.

[42] Squid Web Proxy Cache, http://www.squid-cache.org/

[43] D. C. Verma,“ CONTENT DISTRIBUTION NETWORKS: An Engineering Approach,” John Wiley & Sons Inc., Dec. 2001.

[44] 画像電子学会 編集，小松 尚久，田中 賢一 著，“電子透かし技術ーディジタルコンテンツのセキュリティ，”東京電機大学出版局，2004 年.

[45] Yan Chen, Lili Qiu, Weiyu Chen, Luan Nguyen and Randy H. Katz, “” Efficient and Adaptive Web Replication Using Content Clustering,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 21, NO. 6, pp.979-994, August 2003.

[46] 村田 啓，國近 洋平，甲藤 二郎，“コンテンツ配信ネットワーク (CDN) におけるサロゲート間協調動作の検討，”電子情報通信学会 技術研究報告 NS2003-337, Mar. 2004.

[47] 阿部 泰丈，中庭 明子，池田 大輔，榎原 博之，岡田 博美，“CDN におけるコスト重視型コンテンツ配置更新法 -SxS 最適配送法-，”電子情報通信学会 技術研究報告 NS2003-339, Mar. 2004.

[48] The Linux Kernel Archives, http://kernel.org/

[49] SourceForge.net, http://sourceforge.net/

[50] Load-Balancing DNS server DNS Balance, http://openlab.jp/dns_balance/dns_balance.html

[51] 下川俊彦，吉田紀彦，牛島和夫，“多様な選択ポリシーを利用可能なサーバ選択機構，”信学論 (D-I)，J84-D-I, no.9, pp.1396-1403,Sep. 2001.

[52] 下川俊彦，中川郁夫，山本文冶，吉田紀彦，“広域分散 Web サーバにおける経路情報を用いたサーバ選択，” 電子情報通信学会 技術研究報告 IA-2001-26, pp.49-56, Nov., 2001.

[53] S. G. Dykes, K. A. Robbins and C. L. Jeffery,“ An Empirical evaluation of client-

side server selection algorithm,” Proc. IEEE Infocom 2000, 2000.

[54] 間瀬 憲一，栗林 孝行，津野 昭彦，“QoS 統計データを利用した動的サーバ選択法，”電子情報通信学会論文誌 B，Vol. J86-B, No.3, pp.499-510, Mar., 2003.

[55] 和泉 勇治，宇津江 康太，加藤 寧，根元 義章，“リンクの輻輳状態を考慮した動的なミラーサーバ選択方式，”情報処理学会論文誌 Vol. 45, No. 1, pp.65-73, Jan. 2004.

[56] 山口英，“UNIX Communication Notes 182 品質改善,”UNIX MAGAZINE, pp.59-64, Aug. 2003.

[57] 江崎 浩 編，“特集 ジャパンギガビットネットワーク，”情報処理， Vol.43, No.11, pp.1149-1203, Nov. 2002.

[58] Dave Kosiur 著，苅田 幸雄 監訳，“マスタリング TCP/IP IP マルチキャスト編，”オーム社，1999 年.

[59] クリスチャン・ウイテマ 著，村井 純 監修，WIDE プロジェクト IPv6 分科会 監訳，松島 栄樹 訳，“IPv6 次世代インターネット・プロトコル，”ピアソンエデュケーション，1997 年.

[60] B. Cain, S. Deering, I. Kouvelas, B. Fenner and A. Thyagarajan, “Internet Group Management Protocol, Version 3,” IETF Request for Comments 3376, Oct. 2002.

[61] 永見 健一，“マルチキャスト経路プロトコル，”情報処理， Vol.42, No.8, pp.749-753, Aug. 2001.

[62] 木下 真吾，“リライアブルマルチキャスト技術の最新動向，”電子情報通信学会論文誌 B，Vol.J85-B, No.11, pp.1819-1842, Nov. 2002.

[63] 森口 繁一，一松 信，宇田川 兼久，“級数・フーリエ解析 岩波 数学公式，”岩波書店，1987 年.

[64] NLANR/DAST: Iperf 1.7.0 - The TCP/UDP Bandwidth Measurement Tool, http://dast.nlanr.net/Projects/Iperf/

[65] W. Richard Stevens 著，橘 康雄 訳，井上 尚司 監訳，“詳解 TCP/IP Vol.1 プロトコル，”ピアソンエデュケーション，2000 年.

[66] J. Widmer, R. Denda and M. Mauve,“ A survey on TCP-friendly congestion control,” IEEE Network, vol. 15, no. 3, pp. 28 - 37, May/June 2001.

[67] 竹内 茂樹，古閑 宏幸，飯田 勝吉，門林 雄基，山口 英,“ DDCP を用いた実時間通信のための輻輳制御機構の検討，” 電子情報通信学会技術研究報告 NS2003-355, Mar. 2004.

[68] 鶴 正人，熊副 和美，尾家 祐二,“ 長距離高速通信のための TCP 性能改善技術の動向，”情報処理， Vol.44, No.9, pp.951-957, Sep. 2003.

[69] 熊副 和美，堀 良彰，鶴 正人，尾家 祐二,“ JGN を利用した高速トランスポートプロトコルの評価，”電子情報通信学会技術研究報告 NS2003-354, Mar. 2004.

[70] 永田 晃，山本 和徳，松田 崇弘，山本 幹，池田 博昌,“ レート通知を用いたネットワーク介在型 TCP ふくそう制御方式，”電子情報通信学会論文誌 B，Vol.J85-B, No.8, pp.1402-1410, Aug. 2002.

[71] 鶴 正人，尾家 祐二,“ インターネットの特性計測技術とその研究開発動向，”情報処理，Vol.42, No.02, pp.192-195, Feb. 2001.

[72] TCPDUMP public repository, http://www.tcpdump.org/

[73] Ethereal: A Network Protocol Analyzer, http://www.ethereal.com/

[74] MRTG: The Multi Router Traffic Grapher, http://people.ee.ethz.ch/ oetiker/ webtools/mrtg/

[75] Internet Tools Taxonomy, http://www.caida.org/tools/taxonomy/

[76] M. Jain and C. Dovrolis,“End-to-End Available Bandwidth:Measurement Methodology, Dynamics, and Relation With TCP Throughput,”IEEE/ACM Transactions on Networking, vol. 11, no. 4, August 2003.

[77] 勝山恒男，安達基光,“ トラヒック測定技術: NEPRI，”FUJITSU, Vol.51, No.6, pp.391-395, Nov. 2000.

[78] M. Miyasaka, T. Iwai and H. Kasahara,“Network-friendly Estimation of Available Bandwidth,”5th Asia Pacific Symposium on Information and Telecommunication

Technoloeis (APSITT) 2003 Proceedings, pp.479-484, Nov. 2003.

[79] N. Hu and P. Steenkiste,“Evaluation and Characterization of Available Bandwidth Probing Techniques,”IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 21, NO. 6, pp.879-894, August 2003.

[80] The NET-SNMP Project Home Page, http://www.net-snmp.org/

[81] Modules on CPAN alphabeticaly, http://www.cpan.org/modules/01modules.index.html

[82] PeerCast, http://www.peercast.org/jp/

[83] シェアキャスト，http://www.scast.tv/scast/

[84] 浜 崇之，中里 秀則，浅谷 耕一，富永 英義，“離脱耐性のある P2P ライブ・ストリーミングシステムの提案，”電子情報通信学会技術研究報告，NS2003-367, Mar. 2004.

付録 A

業績リスト

2001 年 4 月から 2004 年 3 月までの対外発表を示す.

A.1　査読付き論文

[1] 中平 拓司，島村 和典，“ 映像コンテンツ配信時における再生品質維持を目指した並列転送方式，” 電子情報通信学会和文論文誌 B（査読中）平成 16 年 10 月号発行予定，インターネットアーキテクチャ技術特集号

A.2　査読付きレター

[2] 中平 拓司，島村 和典，“ ICMP を応用した End-to-End 利用可能帯域測定法，” FIT 情報技術レターズ，Vol.1, pp.207-208, Sep. 2002.

A.3　査読付き国際会議

[3] T. Nakahira and K. Shimamura, “ A Studey of AV File Adaptive Distribution Based on Content Synchronization, ” Proceedings of 4th Asia-Pacific Symposium on Information and Telecommunication Technologies, pp.91-pp.95, Nov. 2001.

[4] T. NAKAHIRA and K. SHIMAMURA, “ A MEASUREMENT SCHEME FOR END-TO-END AVAILABLE BANDWIDTH ON VIDEO DELIVERY SYSTEMS, ” Proceedings of International Conference on Advanced Image Technology 2003, pp.61-64,

Jan. 2003.

[5] T. NAKAHIRA, Y. KOMATSU and K. SHIMAMURA, “ A Replication Method Considering The Dynamics of Network Load in Content Distribution Networks, ” Proceedings of 5th Asia-Pacific Symposium on Information and Telecommunication Technologies, pp269.-pp.274 , 2003.

A.4　研究会発表

[6] 中平 拓司，中平 和友，島村 和典，“ 同期参照による網負荷適応型 AV ファイル配送制御の一検討，” 情報処理学会研究報告，DSM22-2, Jul. 2001.

[7] 中平 拓司，山岡 徹也，島村 和典，“ CDN における映像コンテンツ巡回配布方式-Relaycast-の提案，” 電子情報通信学会技術研究報告 CS2002-117, pp.13-16, Dec. 2002.

[8] 中平 拓司，島村 和典，“ End-to-End 利用可能帯域測定による映像配信制御方式の検討，”電子情報通信学会技術研究報告 NS2002-262, pp.13-18, Mar. 2003.

[9] 中平 拓司，島村 和典，“ 帯域変動時の再生品質維持を目指した映像コンテンツ並列転送実験，”電子情報通信学会技術研究報告，NS2003-336, Mar. 2004.

A.5　全国大会・支部連合大会発表

[10] 中平 拓司，島村 和典，“ IP 網における同期性を考慮した複数 AV データ送信方式の検討，” 2001 年電子情報通信学会ソサイエティ大会講演論文集 B-16-3, Sep. 2001.

[11] 中平 和友，中平 拓司，島村 和典，“ 線形補間を用いた網負荷適応画像符号化法の一検討」平成 14 年度電気関係学会四国支部連合大会講演論文集，” pp.317, Oct. 2002.

[12] 小松 義幸，中平 拓司，島村 和典，“ モニタリングサーバを利用したエンドホスト間利用可能帯域測定法の提案，”平成 15 年度電気関係学会四国支部連合大会講演論文集，pp.193, Oct. 2003.

[13] 中平 拓司，島村 和典，“映像配信時における再生品質維持を目指した並列転送方式，”

2004 年電子情報通信学会総合大会講演論文集 B-7-46, Mar. 2004.

[14] 中平 和友，中平 拓司，島村 和典，“ Bayer-Matrix を応用した網負荷適応画像符号化法の一検討，” 2004 年電子情報通信学会総合大会講演論文集 D-11-20, Mar. 2004.

A.6 表彰

[15] 中平 拓司，島村 和典，“ 同期性を考慮した複数 AV ファイル配信システムの構築，” 2001 画像電子学会年次大会一般セッション 14, Jun. 2001. （2002 年 6 月 研究奨励賞受賞）