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

2003－DPS－114　(7)
2003－EIP－ 20　(7)
2003／8／28

IP ネットワークを使った非圧縮 HDTV 対応映像サーバの構成法に関する検討

君山 博之[†]　清水 健司[†]　川野 哲生[†]　小倉 毅[†]　丸山 充[†]

あらまし　高精細な映像データを，ネットワークを介してアーカイブからオンデマンドで取り出しリアルタイムに転送すること，撮影しながらリアルタイムで格納することは，ネットワークを利用した協調型の映像編集を実現するだけでなく，遠隔講義や遠隔医療の実現を容易にする．我々は，2.4Gbps の MAPOS プロトコルによる超高速 IP ネットワークを使って高精細映像を扱えるシステムの開発を行っている．本発表では，1.5Gbps 非圧縮 HDTV までの高精細映像のリアルタイム配信をターゲットに，システムの核となる映像サーバの，汎用 PC と Linux をベースに構成するときに問題となるストレージのアクセス速度とリアルタイム性を確保する実装方式について提案する．この提案方式を用いて映像配信サーバを実現し，1.5Gbps 非圧縮 HDTV 映像の送信試験を行い± 12% 以内のレート変動で安定的に送信できることを確認した．

A study of the implementation of an uncompressed HDTV video server for IP networks

Hiroyuki KIMIYAMA [†], Kenji SHIMIZU [†], Tetsuo KAWANO [†],
Tsuyoshi OGURA [†] and Mitsuru MARUYAMA [†]

Abstract It will be easy to achieve remote collaborative video editing, distance learning, and remote medical work if high-resolution video can be transferred from video archive systems or stored in them via high-speed network in real time. We are developing a 1.5-Gbps uncompressed HDTV video handling system which delivers this HDTV video in real time over a 2.4-Gbps MAPOS IP network. This paper shows how to implement the server's functions of storage access and real-time transmission, which can deliver 1.5-Gbps uncompressed HDTV using an ordinary PC and Linux. It also presents throughput measurements for the server. The measured variation was within 12% when the server's throughput was 1.5 Gbps.

† NTT 未来ねっと研究所
NTT Network Innovation Laboratories

1．はじめに

近年，ネットワークが高速化が進み，ギガビット超クラスの高速ネットワークが利用可能になっており，この高速ネットワークを利用すれば，HDTV のような高精細な映像を圧縮せずにリアルタイムで送ることは十分可能となる．これにより，従来，圧縮することによって行われてきた映像転送が圧縮無しで実現できるため，映像制作の分野で期待されている．また，細部まで鮮明な映像を必要とする遠隔医療や，遠隔講義の分野でも期待されている．

我々は，そのようなニーズに応えるために，物理層に OC-3c ～ OC-48c の SONET/SDH，リンク層に MAPOS（Multiple Access Protocol Over SONET/SDH）([1]-[5]) プロトコルを使った最大 2.4Gbps の通信が可能な IP ネットワーク上で，非圧縮 HDTV 映像をはじめとする高精細映像をリアルタイムに蓄積，配信可能なシステムの開発を行っている [6].

このシステムの 1 つのキーデバイスとなるのが映像配信サーバである．この配信サーバには，Gbps を越える高速な送信レートを実現することは勿論のこと，安定した送信レートを作り出す機能が不可欠である．送信レートが安定しなければ，映像受信側でアンダーフローやオーバーフローが起き，再生映像に乱れを生じる可能性がある．バッファによるレート変動の吸収も可能であるが，1 秒あたり 200MB 弱の非圧縮 HDTV 映像を扱うシステムの場合，バッファを多く持たせることはレスポンスの低下も引き起こし，ユーザビリティーに欠けるシステムとなるだけでなく，システム全体のコストを引き上げることとなる．

さらに，安定した送信レートを実現するためには，高速な読み出し・書き込みが可能なストレージもまた要求される．そこで，我々は，非圧縮 HDTV を手軽に扱えるようにするために，汎用 PC と Linux をベースに映像配信サーバを実現することを目的に，高速なストレージ構成法とリアルタイム送信機構について検討し，映像配信サーバを開発した．汎用 PC と Linux を選択したのは，

(1) ハードウェアが安価であり入手が容易
(2) Ultra 320 SCSI 等の最新の高速インタフェースカードが使用可能

という理由からである．

本発表では，このプラットホームを用いて，映像データを蓄積するための高速なストレージを実現するための方法，リアルタイム送信を実現するための方法について報告する，それらを組み合わせて実装し，1.5Gbps の非圧縮 HDTV 映像をオンデマンドにかつリアルタイムに配信可能なサーバを実現できることを示すとともに，その映像配信サーバのリアルタイム性についての性能評価について報告する．

2．高速ストレージ構成法

高速な映像配信サーバを実現するために重要なポイントの 1 つは，アプリケーションレベルで高速な読み出し・書き込みが可能なストレージをいかに構成するかである．さらに，ストレージを構成を決定する要素は，ハードウェアをどのように構成するかと，どのファイルシステムを使用するかに分けられる．以下に，それぞれについての詳細を記述する．

2.1 ハードウェア構成

まず，映像配信サーバに使用するストレージとして，性能の観点から NAS（Network Attached Storage）タイプのストレージではなく，DAS（Direct Attatched Storage）タイプのストレージを採用することとした．次にストレージと PC 本体とのインタフェースを選択する必要がある．DAS タイプのストレージのインタフェースとして代表的なものには Fibre Channel インタフェースと SCSI がある．Fibre Channel には，ハードディスクドライブを SCSI よりも多く接続できるメリットがある反面，容量あたりのコストが高くなるというデメリットがある．我々は，コストの観点から，最高速度 320MB/s の Ultra 320 SCSI インタフェースをストレージインタフェースとして選択した．

2.2 ファイルシステムの選択

前述した通り，ストレージサブシステムを構成するにあたって，ファイルシステムの選択は重要である．ファイルシステムについては，独自に構成する方法と既存のいくつかのファイル

システムから選択する方法がある．製造とメンテナンス工数のことを考慮するとEXT3やXFSなど既存のファイルシステムの中から選択することとした．Linuxでサポートしているファイルシステムの評価については，いくつかのベンチマークが行われている（例えば[7][8]など）．これらのベンチマークの結果から，我々は，読み取り・書き込み性能が最も高いXFSをファイルシステムとして選択した．

2.3 ストレージパフォーマンス評価

前述したハードウェアとファイルシステムを使用して，実際に十分な読み取り・書き込み速度を得られるかどうかの検証を行った．図1に試験系構成を，表1に使用したハードウェアのスペックを示す．この表に示すように，プラットホームのコンピュータとして，一般的なPCサーバ機で，内部バスの速度で性能が劣化しないように133MHzのPCI-Xバスを持つものを使用した．SCSIカードにも，同様に，PCI-Xをサポートしたものを使用した．ハードディスクドライブには，10000rpm，146GBの容量を持つUltra 320 SCSIドライブを複数台使用した．ハードディスクドライブとして15000rpm，73GBの容量を持つ高速なものも，現在，入手可能である．今回の評価では，非圧縮HDTVを使うことを前提としているため，大容量のストレージを構築する必要があるとともに，ディスク台数をできるだけ減らすことによって，システムのサイズを小さくするだけでなく障害の起こる頻度を下げることも重要であると考えている．そのため，回転数は低いが容量の大きいハードディスクドライブを選択した．OSはRedHat Linux 7.3環境にKernel 2.4.20とXFSバージョン1.2を導入した．

まず，最初の評価として，複数台のハードディスクドライブを使用して，ソフトウェアRAID（レベル0）ボリュームを構築し，XFSファイルシステムで初期化し，読み取り速度と書き込み速度評価を行った．速度の評価方法として，このファイルシステム上でボリュームの容量と同じサイズの1つのファイルを作成し，そのファイルに対する読み取り速度と書き込み速度を，ファイル内の全データに対して計測する方法を用いた．1.5Gbpsという高速なコンテンツを扱うことから，1回の読み取り，書き込みサイズが大きくなることが想定されることから，この測定では，ランダムアクセスではなくシーケンシャルアクセス速度を計測した．1回の読み取りサイズは128MBとし，読み取りのためのreadシステムコールの時間を測定して，その値から読み取り速度を求めた．書き込み速度も同様である．

図2にハードディスクドライブ台数と読み取り・書き込み速度との関係を示す．図中の読み

表1　ストレージ評価試験環境

PC本体	CPU	Intel Xeon 2.4GHz
	Memory	512MB
	Mother Board	Supermicro P4DL6
	SCSI HBA	Adaptec 39320D
ストレージ	HDD	Seagate ST3146807LC
	JBOD	Adaptec DuraStor 412R
OS	RedHat Linux 7.3 + Kernel 2.4.20	

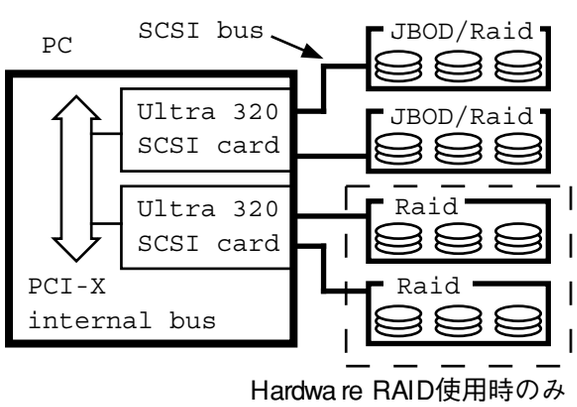


図1　ストレージ評価試験系構成

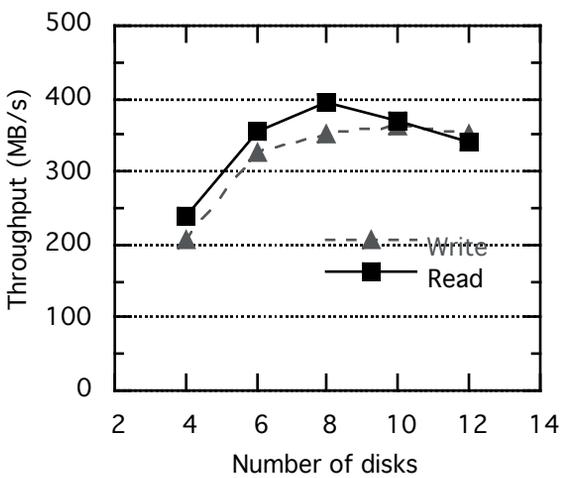


図2　ソフトウェアRAIDのみの性能評価結果

取り速度は，計測した全域の速度の平均値である．読み取り速度はドライブ８台で，書き込み速度はドライブ１０台で最大となることが確認された．一般的に 128MB のデータ全てが連続セクタに格納されることは稀であり，１回の読み取りの間に何回かのディスクヘッドのシークが入る．また，データが書き込まれたディスク上の位置によっても，読み取り速度は変動する．映像配信サーバとして使用する場合，問題となるのはその速度の最小値である．この値が映像レートよりも小さい場合は，受信側でバッファリングしない限りアンダーフローを起こす．表２にドライブ８台使用時の読み取りおよび書き込み速度の平均，最小，最大値を示す．この表からドライブ８台で非圧縮 HDTV 映像配信サーバ実現のための，十分な速度を持つストレージが構築できることが判明した．

次に，蓄積容量アップと信頼性の向上のため，表３に示すハードウェア RAID を複数台使用して評価を行った．ハードウェア RAID のインタフェースは表３に示すように Ultra 160 SCSI であるが，Ultra 320 SCSI と接続可能であるため，表１の構成を使用して評価を行った．ハードウェア RAID 毎に RAID レベル５の LUN を構築し，その LUN を使ってソフトウェア RAID（レベル０）ボリュームを構築し，その速度を測定した．１回の読み取り・書き込みサイズ，測定方法は前述した通りである．ハードウェア RAID の台数と読み取り・書き込み速度の関係を図３に示す．この結果から，ハードウェア RAID を３台以上使用すれば，目的とする性能が得られることが判った．ハードウェア RAID を使用した方が書き込み速度が早いのは，ハードウェア

表２　HDD ８台使用時の読み取り・書き込み速度（MB/s）

	平均値	最小値	最大値
read	394	270	405
write	331	263	354

表３　ハードウェア RAID スペック

RAID 機種名	Adaptec DuraStor 6320
RAID コントローラ	１台
キャッシュメモリー	128MB
HDD	Seagate ST3146807LC ７台
外部インタフェース	Ultra 160 SCSI

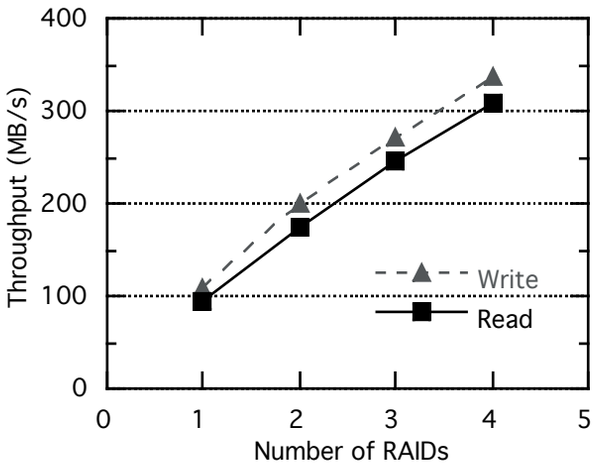


図３　ハードウェア RAID による性能評価結果

RAID コントローラ上のキャッシュメモリの効果であると思われる．

３．疑似リアルタイム処理

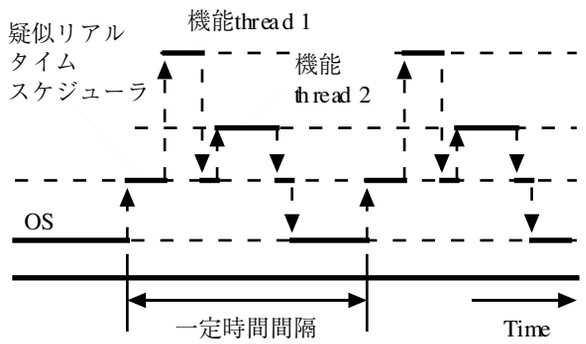
3.1　処理概要

受信側で映像データがアンダーフローやオーバーフローを起こさないためには，リアルタイムにデータをディスクから読み出し，送信する必要がある．我々は，このリアルタイム制御を次の方式を用いて実現した．それは，まず，映像配信処理に必要のないアプリケーション停止するとともに，映像配信サーバ機能を１つのアプリケーションとして実装し，そのアプリケーション内に独自のスケジューラを構築して，処理スケジュールをアプリケーション内で制御し，リアルタイム性を確保する方式である（以下，アプリケーションレベルスケジューリング方式という）．この方式は，アプリケーションレベルで実装することにより，製造やデバッグが容易であり，OS に依存しないので移植性にも優れている．また，汎用 OS を利用できるためリアルタイム OS を用いて実装する場合に比べて，利用可能なハードウェアが多いメリットがある．その反面，完全なリアルタイムが保証されていないため，アプリケーション内の各処理が遅延することによって，最終的に，受信側でアンダーフローを引き起こす可能性がある．したがって，この遅延がどの程度なのかを事前に評価し，その遅延を吸収可能なバッファを受信側に用意する必要がある．

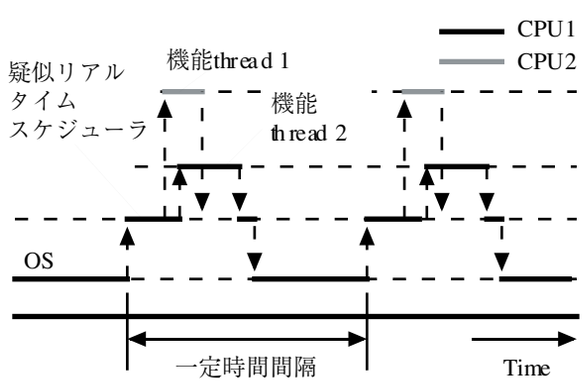
3.2 アプリケーションレベルスケジューリング方式の実装

アプリケーションレベルスケジューリング方式の概略を図4に示す．まず，図4(a)に示すように，アプリケーションでの処理を予め機能ごと，例えば，データ送信，ディスクアクセスなどに分割し，threadとして実装する（以下機能threadと呼ぶ）．また，これらの機能は，予め処理時間を評価しておき，後述するスケジュール機能の起動時間間隔以内に終了できるように実装する．さらに，分割した処理の優先順位を決めるためのスケジューラ機能を実装する．この機能はOSのインターバルタイマシステムコールを用いて一定時間間隔で起動され，各機能threadをその優先順位に応じて順番に起動する．

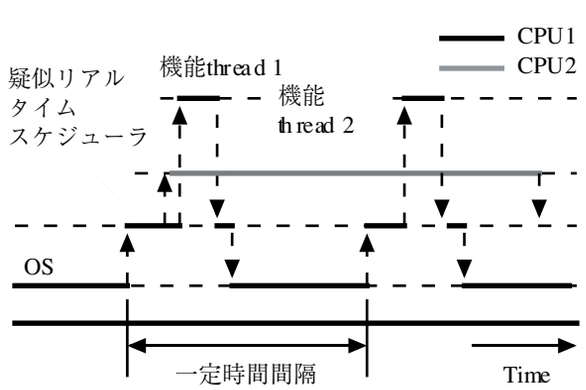
マルチプロセッサ機能が使える場合は，2つの機能threadを同時に走らせることが可能なため，図4(b)のように実行させることが可能である．さらに、2つの機能threadの間でリソースの競合が無ければ図4(c)のように，1回の処理時間がインターバルタイマの時間間隔を越えるような機能threadも動作させることが可能となる．



(a) Single processorによる機能thread起動



(b) Multi processorを使った機能thread同時起動



(c) 長い処理時間の機能thread同時起動

図4　疑似リアルタイムスケジューラタイミングチャート概略図

3.3 リアルタイム性評価

この方式のリアルタイム性を検証するために，図5に示すシステムを構築し，前述したアプリケーションレベルスケジューリング方式を実装した疑似リアルタイムスケジューラと，機能threadとして，データ送信処理を機能を実装（図6）し，送信側PCから受信側PCへ0.5Gbps～2Gbpsの一定レートで送信し，受信側で受信

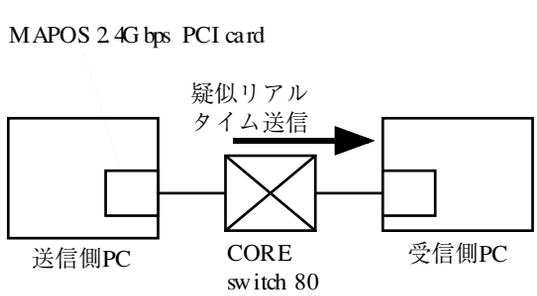


図5　疑似リアルタイム評価試験系構成

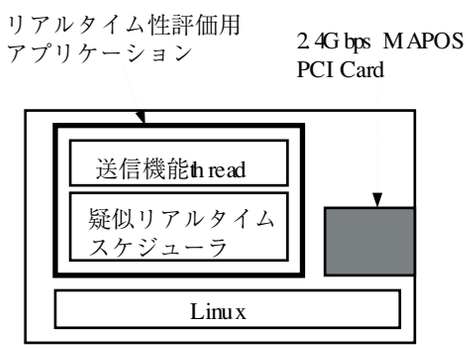


図6　疑似リアルタイム評価試験用アプリケーション構成

レートを測定した．NIC として 2.4Gbps MAPOS PCI カード，データリンクプロトコルに MAPOS を使用するとともに，受信側と送信側の間には低遅延の MAPOS プロトコルをサポートした CORE switch 80 を図 5 のように配置した．この 2.4Gbps MAPOS PCI カードは，約 64KByte の長大 MTU が利用可能な MAPOS プロトコルと組み合わせることによって，2.4Gbps のワイヤーレートまでの通信性能を実現することが可能である [10][11].

図 7 に受信側 PC で測定した受信レートの時間変化を示す．この図のように受信レートは，ほぼ一定となることが確認でき，アプリケーションレベルでもリアルタイム性が実現できることが確認できた.

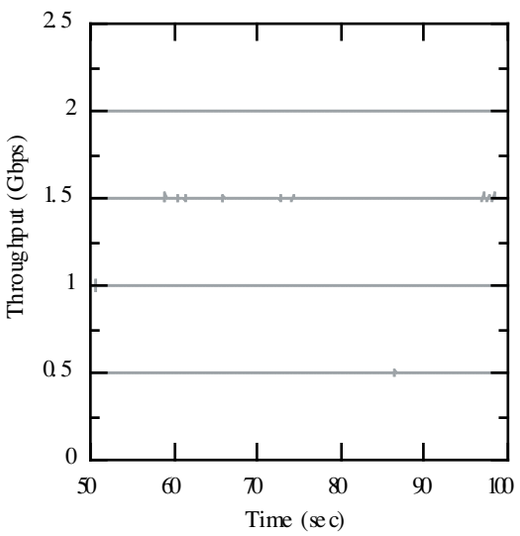


図 7　疑似リアルタイム評価試験結果

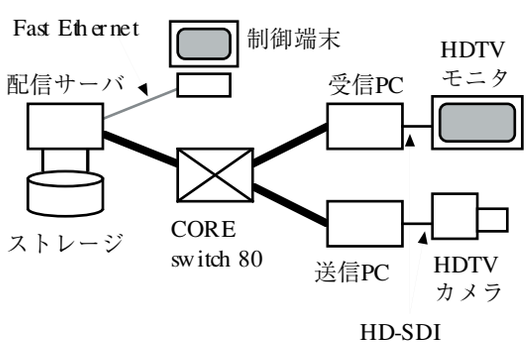


図 8　非圧縮 HDTV 映像配信システム概念図

4．サーバの構成と評価

4.1　配信システム概要

配信サーバを含む非圧縮 HDTV コンテンツ配信システムの基本構成を図 8 に示す．図中の受信 PC は，映像コンテンツをネットワークから受信し，HD-SDI インタフェースに出力する．送信 PC は HD-SDI から入力された映像データをネットワーク経由で配信サーバに送信する．また，図中の制御端末ではサーバからのストリームの送信，停止を制御する．配信サーバと受信 PC，送信 PC との間は前述した MAPOS 2.4Gbps のネットワークで結ばれている．制御端末と配信サーバの間は Fast Ethernet を使用し，制御メッセージのやり取りを行う．なお，上位プロトコルとして，映像コンテンツは UDP/IP，制御メッセージは RTSP[12]/TCP/IP を使用している．蓄積・配信可能な映像コンテンツは，SMPTE 292M に準拠した 1080i/60 非圧縮 HDTV 以下のレートの映像である.

4.2　配信サーバの構成

配信サーバのハードウェアとして，表 1 に示したスペックの PC を使用した．OS も同様に表 1 に示したものを使用した．ストレージとしては，前述した評価結果から Seagate 社 ST3146807LC 8 台を用いてソフトウェア RAID（レベル 0）を構築し，XFS で初期化したものを使用した.

次に，配信サーバのアプリケーションソフトウェアの構成は，前述した疑似リアルタイムスケジューラを使用し，配信サーバの機能を 3 つの機能 thread（端末間制御機能，データ送信処理機能，ディスク読み取り機能）に分割し，実装した. 各機能 thread の起動時間間隔を表 4 に示す．OS から疑似リアルタイムスケジューラを起動する時間間隔は 10msec である．ディスク読み取り

表 4　機能 thread の起動時間間隔

機能 thread	起動間隔（msec）
疑似リアルタイムスケジューラ	10
端末間制御機能	10
データ送信処理機能	10
ディスク読み取り機能	330/340

機能に関しては，読み取り速度を上げるため１回の読み取り単位を 10 フレーム単位とした．表２に示した読み取り速度の最小値から，1080i/60 の非圧縮 HDTV の場合，10 フレーム分のデータに対する最大読み取り時間は，約 218msec と推定される．この数値は，インターバルタイマの時間間隔の 10msec を越えるが，マルチプロセッサ機能により，前述した通り問題なく動作可能であると考えられる．

4.3　評価

前述した疑似リアルタイム処理が前述した通り動作するかどうかを実証すると同時に，受信 PC に必要なバッファ量を見積もるため，図８のシステムを用いて配信サーバから，実映像を送信し，受信 PC で受信レートどのようになるかを評価した．配信サーバから受信 PC へ 1080i/60（約 1.5Gbps）の映像コンテンツを５分間送信し，受信 PC で１フレームを受信する時間間隔を継続的に測定した．その結果から求めた受信レートの時間に対する推移を図９に示す．また，図 10 に１フレーム受信の時間間隔の分布を示す．この結果の最大，最小時間間隔はそれぞれ 37.1msec，29.4 msec であり，フレームの到着時間の揺らぎを± 4 msec（± 12%）以内に押さえることが出来た．UNIX の時間分解能が 10msec であることを考えれば，十分小さい値であると言える．この結果から，汎用 PC と Linux を使って，前述したストレージの構成法と疑似リアルタイム処理方式を組み合わせることにより，1.5Gbps という超高速な送信レートで安定的に配信可能な映像配信サーバを実現可能であることを確認した．

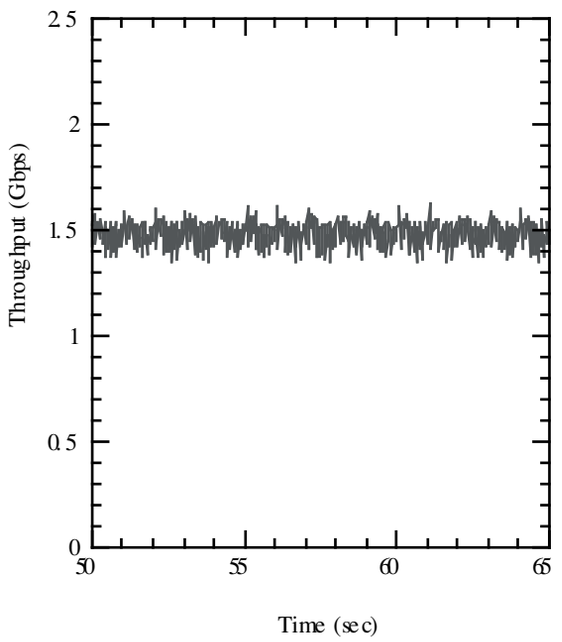


図９　受信 PC におけるスループットの推移

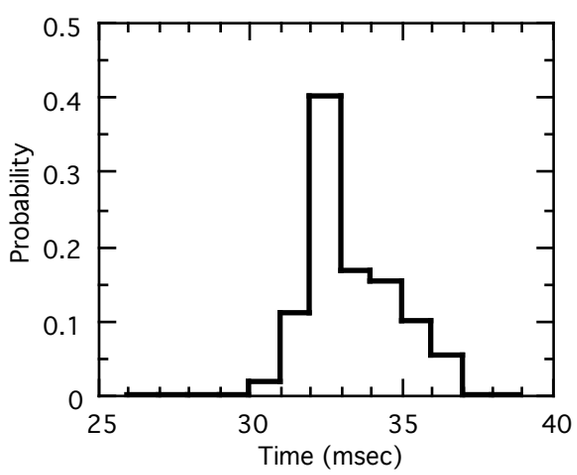


図 10　１フレーム受信時間間隔分布

さらに，この実験で得られた時間間隔データと１フレームの時間間隔 33.33msec との差を順番に加算し，その最大値と最小値を求めた．受信 PC に必要となる最低のバッファ量はこれらの値の絶対値の大きい方である．最大値，最小値は，それぞれ 7.0 msec，-9.3 msec であった．このことから，受信 PC に必要なバッファ量は 10 msec 分であることが判った．

５．まとめ

汎用 PC と Linux をベースに，非圧縮 HDTV をリアルタイムで送信可能な高速な映像配信サーバの構成するために必要な高速なストレージの構成法とリアルタイム処理の実現方法を記述した．これらを元に，配信サーバを実装し，ネットワークを通じて，1.5Gbps の非圧縮 HDTV 映像データを送信したときの受信側の受信レートを測定した．この結果から，受信レートの揺らぎについて評価し，レートの揺らぎが± 4 msec 以内に収まることを確認した．これにより，1.5Gbps の超高速なレートでも，小さいレートのゆらぎで安定的に送信可能な映像配信サーバを汎用 PC と Linux により実現できることを確認した．また，この結果から，受信側に持たせる必要がある最低バッファ量を評価し，この値が１フレームの

時間間隔よりも短い 10 msec であることを確認した．

今後は，Fibre Channel を使用した SAN（Storage Area Network）を適用することによって大容量化と高可用性の向上を図るとともに，動画像編集システムとのインテグレーションを図っていく予定である．

参考文献

[1] K. Murakami and M. Maruyama: MAPOS - Multiple Access Protocol over SONET/SDH Version 1, RFC-2127 (1997).

[2] M. Maruyama and K. Murakami: MAPOS Version 1 Assigned Numbers, RFC-2172 (1997).

[3] K. Murakami and M. Maruyama: A MAPOS Version 1 Extension - Node Switch Protocol, RFC-2173 (1997).

[4] K. Murakami and M. Maruyama: A MAPOS Version 1 Extension - Switch-Switch Protocol, RFC-2174 (1997).

[5] K. Murakami and M. Maruyama, MAPOS 16 - Multiple Access Protocol over SONET/SDH with 16 Bit Addressing, RFC-2175 (1997).

[6] NTT: i-Visto Internet Video Studio System for HDTV Production, http://www.i-visto.com/ .

[7] R. Bryant, R. Forester and J. Hawkes: Filesystem Performance and Scalability in Linux 2.4.17, Proceeding of the 2002 Usenix Annual Technical Conference (2002).

[8]R. Dunlap: Linux Journaling Filesystems and Workloads, http://www.osdl.org/docs/linux_journaling_filesystems_and_workloads.pdf, Aug. (2002).

[9] SGI: Developer Central Open Source ¦ Linux XFS, http://oss.sgi.com/projects/xfs/ .

[10] 清水，川野，小倉，丸山：MAPOS 対応 OC-48c PCI カードの実現と性能評価，信学技報 NS2002-55, pp.9-12 Jun. (2002).

[11] 川野，小倉，清水，丸山，小柳：非圧縮 HDTV over IP システムにおける高速プロトコル処理技術，信学技報 NS2002-51, pp.47-50, Jun. (2002).

[12] H. Schulzrinne, A. Rao and R. Lanphier: Real Time Streaming Protocol (RTSP), RFC-2326 (1998).