

DiffServ 技術を用いた動画配信システム

深澤昌志¹, 八代一浩², 伊藤洋³

¹ 山梨大学 工学研究科

² 山梨県立女子短期大学

³ 山梨大学

概要

近年ネットワークの高速化に伴い実時間通信の需要が高まって来た。しかし現在のインターネットは最善努力型 (best-effort) を基本とした通信が行われているため、実時間処理を行う様な通信は非常に困難である。これらの実現ため QoS(Quality of Service) の要求が高まり、従来の配送系が大きく変わろうとしている。

本研究では QoS 実現のため DiffServ 技術 [1] を用いて実時間通信を行うネットワークシステムの提案を行う。また提案するネットワークシステムの実装を行い、その評価を行う。

An Implementation of an Network Broadcasting System Using DiffServ Technology

Masashi Fukasawa¹, Kazuhiro Yatsushiro², Hiroshi Ito³

¹Faculty of Engineering, Yamanashi University

²Yamanashi Women's Junior Collge

³Yamanashi University

Abstract

Some of recent applications on the Internet require real time communication. However, the present the Internet, it is very difficult to use for real time communication, because of the present Internet communication are based on best-effort. In order to accept these requests, it should be implemented the QoS(Quality of Service) on the network.

In this paper, the network system which performs real time communication using DiffServ technology[1] for QoS realization is proposed. And, we also show the implementation and evaluation of this system.

1 はじめに

現在のインターネットはパケット配送系に信頼性を求めない簡単な仕組みで構成されている。このことはインターネットの拡大や相互接続を容易にし、インターネット普及の大きな要因となった。

近年インターネットの普及、高速化にともない、実時間での動画通信の様な負荷の高いアプリケーションの通信の需要が高まって来た。しかし現在のイン

ターネットでは最前努力型 (best-effort) での通信を基本として行なっている。最前努力型での通信では、帯域の確保、ジッタの抑制などの処理を行うことが非常に困難である。これらの実現ため QoS の要求が高まり、従来の配送系が大きく変わろうとしている。

本研究では QoS 実現のため DiffServ 技術を用いた動画配信のためのネットワークシステムの提案を行い、そのシステムの実装および、評価を行う。

2 DiffServ

ネットワーク上で QoS を実現するための仕組みに DiffServ (Differentiated Services) 技術がある [1]. DiffServ は優先制御によって通信品質を保証するための枠組である。顧客は ISP とサービスレベルの契約 (SLA: Service Level Agreement) を結び、この SLA におおじたサービスを受けることが可能となる。

DiffServ を行うドメインを「DS ドメイン」と名づけ、DS ドメインの境界にあるルータをエッジルータと呼ぶ。DiffServ ではこのエッジルータで SLA にしたがって IP ヘッダの TOS (Type of Service) フィールド (Ver.6 では DS フィールド) に優先度情報の付与 (マーキング) を行う。同一の DS ドメイン内ではマーキング値が同じ IP パケットに対しルータで同一の処理を行う。つまり DS ドメイン内に共通のポリシー (方針) を持たせることによりマーキング値によって同一の差別化を行うことを DiffServ という。

DiffServ の枠組では各ルータ間で特定のキューイング方式やポリシーを限定していないため帯域保証や優先制御、セキュリティ処理の有無などのサービスが可能となる。また、DS ドメインは通常、一つの ISP 単位で考えられる。

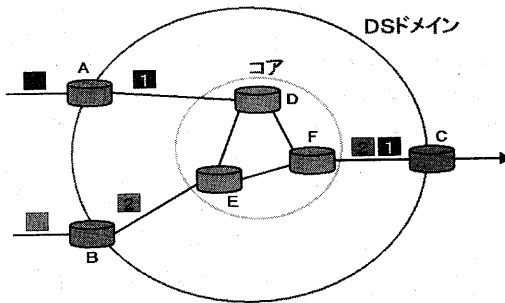


図1 DS の概要 ルーティングポリシーの例

「1」を「2」よりも優先させる

- DS ドメイン : マーキング値が同じパケットに対し同一処理を行う
- エッジルータ : パケットにマーキングを行う (A,B,C)
- コア : マーキング値に基づき差別化を行う (D,E,F)

図1は DiffServ における差別化の例を示している。A,B からそれぞれ DS ドメイン内に入ってきたパケットが、ルータ C を通り、出て行くことを考える。A から入って来たパケットに「1」、B から入って来たパケットに「2」をエッジルータである A,B でそれぞれマーキングする。コアルータ E,D,F のどのルータでも同じ処理 (ここでは1の値がついたパケットを優先して送る) を行う。よってルータ C から送出される時には必ず「1」の値がついたパケットが優先されて出てくる。

3 動画配信システム

3.1 動画配信システム

DiffServ 技術を用いた動画配信システムとして図2に示すようなシステムの構築を行った。

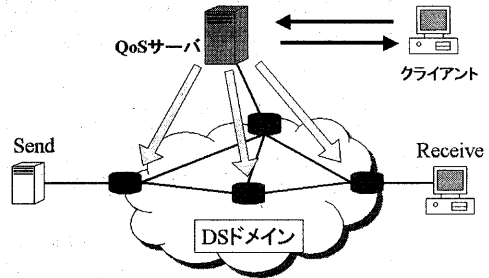


図2 動画配信システム

DiffServ 技術を用いた動画配信システムの構築には QoS サーバと呼ばれるネットワーク管理サーバの設置を行う。Send から Receive へ動画を配信する際、まず QoS サーバはクライアントからの帯域確保要求の受け付ける。QoS サーバにはネットワークの資源状況を常に把握させるため、受け付けた要求をデータベースによって管理させる。これらの情報を元に DS ドメイン内のルータへ動画配信に必要な設定を行う。

QoS サーバの役割を分けると以下の様な 11 のモジュールに分類できる。これらのモジュールは、(1)~(8) のような一連の動きをする。その様子を図3に示す。

- ユーザインタフェース クライアントの帯域予約の受け付け。
- 認証データベース 帯域制御利用可能者リストデータベース。
- レジストレータ 予約機能。
- リソースデータベース ネットワーク資源データベース。
- リソースマネージャ ネットワーク資源管理。
- スケジュールデータベース 帯域制御要求データベース。
- アドミッションコントローラ 網内への予約受け付け管理。
- スケジューラ スケジューリング。
- アカウンティングデータベース 利用履歴データベース。
- アカウンティングマネージャ 利用履歴計算。
- デバイスコントローラ ネットワーク機器の設定。

- (1) クライアントがブラウザを利用した、ユーザインタフェース(図4)から、QoSサーバへ帯域制御要求を送る。
- (2) ユーザインタフェースから送られてきた要求を受けたレジストレータは、認証データベースを検索し、要求を送った利用者の認証を行う。認証が行えた場合は、その要求をアドミッションコントローラへ送り、属してない場合は、要求を受けることができないことを利用者に告げる。
- (3) アドミッションコントローラは、スケジュールデータベースを検索する。さらに、リソースマネージャにも問い合わせを行い帯域制御が行えるか調べ、その結果を利用者に告げる。
- (4) リソースマネージャは、リソースデータベースを検索し、現在使用するのネットワークにその要求を満たす帯域などが残っているかを検索する。要求を満たすことができるなら、アドミ

ッションコントローラに返し、要求を満たすことができない場合は、要求を受けることができないことを利用者に告げる。

- (5) アドミッションコントローラは、送られてきた要求をスケジュールデータベースに加える。もし解除の要求ならば消去する。
- (6) スケジューラは、スケジュールデータベースの情報をアカウンティングマネージャへ送る。
- (7) アカウンティングマネージャは、送られてきた情報を元に利用履歴を計算し、その結果をアカウントデータベースに加算する。その後、デバイスコントローラへ送る。
- (8) デバイスコントローラは、送られてきた情報を基に帯域制御の設定を必要なルータへ送り、キューイング方式などの変更を行う。

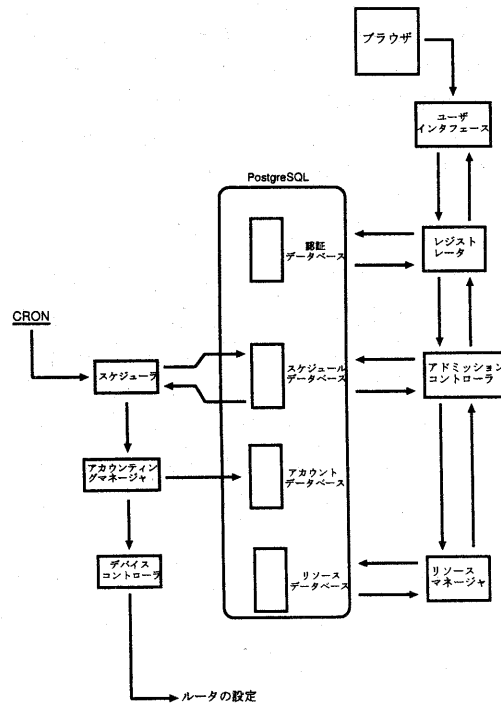


図3 QoSサーバシステム

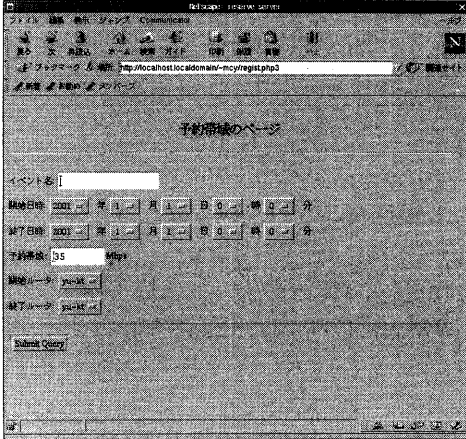


図 4 登録ページ

4 システムの評価

4.1 実験ネットワーク

DiffServ を使った動画配信システムを評価するため図 5 の様なネットワークを構築した。Send から DV 映像を Receive に配送する。これを測定用トラヒックと呼ぶ。またネットワークに負荷を与えるために UDP サーバから QoS サーバに UDP パケットを送信する。これを負荷トラヒックと呼ぶ。その際、測

定用トラヒックと負荷トラヒックにそれぞれマーキングを行ない、測定用トラヒックの帯域を確保するようルータ A で差別化を行う。つまりルータ A はマーキングと共に差別化も行うこととなる。

ルータ A,B におけるキューイングには CBQ(Class Base Queueing) を使用し、コンディショナーにはトークンバケット方式を用いる。またこれらのキューイングを実装するため ALTQ[2] を用いる。

DV 映像の転送には DVTS[3] を用いる DVTS は約 30Mbps の帯域を使い DV 映像を実時間で転送する。キューイングの設定は表 1 に、各マシンのスペックを表 2 に示す。

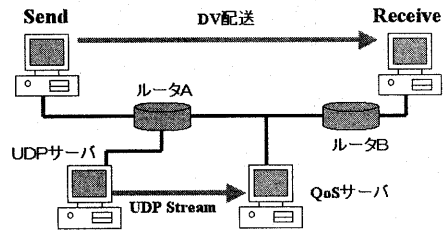


図 5 実験用ネットワーク図

表 1 キューイングの設定

パケットの種類	与える帯域
DV パケット	35Mbps
UDP パケット	空いている部分

表 2 マシンスペック

	CPU	メモリ	OS	ソフトウェア
Send	PentiumIII 1GHz	256M	FreeBSD-3.4	DVTS-0.3.2
Receive	PentiumIII 600MHz	256M	FreeBSD-3.4	DVTS-0.3.2
ルータ A	PentiumIII 600MHz	256M	FreeBSD-3.4	ALTQ-2.1
ルータ B	AMD K6-II 400MHz	128M	FreeBSD-3.4	ALTQ-2.1
QoS サーバ	PentiumIII 1GHz	256M	Vine Linux-2.1	postgreSQL
UDP サーバ	PentiumII 200MHz	128M	FreeBSD-3.4	

4.2 評価方法

評価は全 30 分で行い、ネットワークの状況に伴い、10 分ずつ 3 つの状態にわけらる。

- 状態 A (0~10 分)

Send から Receive まで DV データの配送を行う。この状態ではネットワーク的に負荷がない (DV トラフィックを妨害するトラフィックがない) ため、安定した送信が行われると考えられる。つまり DiffServ を使ったときの目標状態となる。

- 状態 B (10~20 分)

次の 10 分間で UDP サーバから QoS サーバへの UDP パケットの転送を開始する。この時までルータ A において差別化は行わない。よって Best-effort 状態での転送となる。

- 状態 C (20~30 分)

最期の 10 分間でルータ A において差別化を開始する。このとき、DV トラフィックの帯域を確保し、他のトラフィック (ここでは UDP トラフィック) にはその他使っていない部分を与える。

それぞれの状態においてトラフィック量、パケットロス、ジッタの計測を行う。

4.3 結果および評価

- トラフィックの様子

状態 A では DV 転送の必要量である約 30Mbps を安定して転送を行っている。状態 B では UDP トラフィックによりネットワークに負荷がかかってしまい、ルータ A においてパケットが破棄されてしまうため、DV トラフィックの帯域が十分確保できていない。状態 C で DiffServ を使うことによって DV の帯域を状態 A と同様程度、確保できている。また UDP トラフィックの帯域は DV トラフィックに帯域を譲った分帯域が落ちてしまっているが、ネットワーク全体 (DV トラフィックと UDP トラフィックの合計) で見たとき状態 B の時と変わらない帯域を使用できている。つまり DiffServ を使用してもネットワーク的な資源を無駄にすることなく、通信を行うことができると言える。

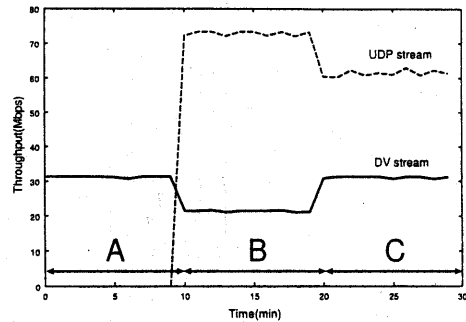


図 6 トラフィックの推移

- パケットロス

各状態のパケットロスの一秒毎の平均値を表 3 に示す。ここで注目したいのは損失率である。状態 B では 31.4% あった損失率が DiffServ を使うことにより 0.03% まで軽減させることができた。

表 3 パケットロスの合計と平均 (個)

	状態 A	状態 B	状態 C
reach	7492.22	5137.07	7490.05
loss	0.04	2354.88	2.20
損失率 (%)	0	31.4	0.03

reach: 1 秒間で到着したパケットの平均

loss: 1 秒間で失ったパケットの平均

損失率: $\text{loss} / (\text{reach} + \text{loss})$

- ジッタの計測

パケットの到着間隔のばらつきであるジッタを測定した。DV トラフィックのジッタを計測したものを図 7 に示す。ジッタの計測にはビデオサーバから 1 秒ずつ DV と同じマーキングを行ったパケットを送信する。ジッタの算出は
$$\text{Jitter} = T_i - T_{i-1}$$
 T_i : i 番目のパケットが到着した時間 (受信ホストで測定) T_{i-1} : i-1 番目のパケットが到着した時間 (受信ホストで測定) という形で行う。

実時間でのビデオ配送では帯域保証と共にジッタがビデオの品質に大きく影響を与える。図 7 はジッタの推移を表す。状態 B に比べ状態 C では大きく改善できていることがわかる。

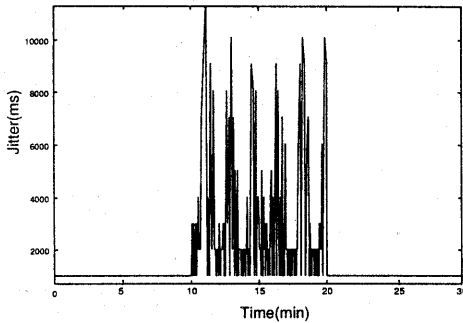


図7 ジッタの推移

● 動画像の状況

状態 B では UDP トラフィックの影響のため実際の動画像にはブロックノイズが現れてしまう。これは動画像の packets が届かなかった場合、その packets 部分が表示できないため起こる現象である。また画像と同様、音声にも影響がでてしまう。しかし状態 C、つまり DiffServ を使用することにより、トラフィックの保証、パケットロスの減少、ジッタの抑制を行うことができ、そのため動画像の画質、音声を状態 A と同等のレベルでの転送を行うことができた。

表 3 のパケットロスの値から毎秒約 2 パケット

程の損失がある。しかし実際の画像、音声に影響は与えられていない。

5 まとめ

今回の研究で DiffServ を用いた動画配信システムの構築および配送系の評価を行った。今後の課題としてリソースマネージャと、トラフィックエンジニアリングの開発があげられる。さらに個々のシステムを利用したビデオ会議システムなどのアプリケーションの開発を行いたい。

参考文献

- [1] S.Blake, "An Architecture for Differentiated Services," RFC2475, 1998
- [2] <http://www.csl.sony.co.jp/person/kjc/programs.html#ALTQ>.
- [3] Akimichi Ogawa, Katsushi Kobayashi*, Kazunori Sugiura, Osamu Nakamura, Jun Murai
"Design and Implementation of DV based video over RTP,"
<http://www.sfc.wide.ad.jp/DVTS/>, 2000.