

LM-1

ストリーミング向け帯域制御機構の開発 Bandwidth Control Mechanism for Streaming Servers

内藤 義郎†
Yoshiro Naito

吉澤 康文‡
Yasufumi Yoshizawa

1. はじめに

ストリーミングとは、インターネット上で音声や映像情報をサーバが送信し、クライアントはデータの受信と平行して再生を行う技術である。ストリーミングサーバには複数の異なるクライアントに対して、それぞれ一定の帯域でデータを送信する機能が要求される。しかし、ネットワークの状態によってはデータの再生に必要な帯域を十分確保できないクライアントもある。また、途中でネットワーク帯域が大きく変動する場合もある。

本論文では、上記の問題に柔軟に対応するストリーミングサーバ用の帯域制御機構：BCMS (Bandwidth Control Mechanism for Streaming) と BCMS を用いてストリーミングフローを制御する SSC (Streaming Session Control) の設計と実装について述べる。SSC/BCMS は、ネットワーク帯域が広く安定したクライアントには高品質のストリーミングを提供し、ネットワーク帯域が狭いまたは不安定なクライアントには利用可能な帯域内でなるべく高品質なストリーミングを提供することを目的とする。

2. BCMS による帯域制御

BCMS は複数の固定長パケットからなる「スロット」を単位に図1のような帯域制御を行う。帯域を制御するパラメータは「PPS」(Packet Per Slot)である。

「通常帯域時」はパケットの破棄が起こらない状態であり、サーバが実際に送信する帯域はマルチメディアデータが必要とする帯域(メディア帯域)である。「抑制帯域時」はパケットの破棄が起こる状態であり、送信帯域は通常帯域時の帯域の PPS/S_LEN (スロット長)倍である。図1は、抑制帯域時が通常帯域時の60%のケースである。

BCMS ではスロットの先頭から後ろに行くにつれてパケットの重要度は低くなるものと仮定する。従って、パケットは必ずスロットの後方から破棄される。また、サーバはBCMS の要求に合うようにメディアデータを加工しておく必要がある。

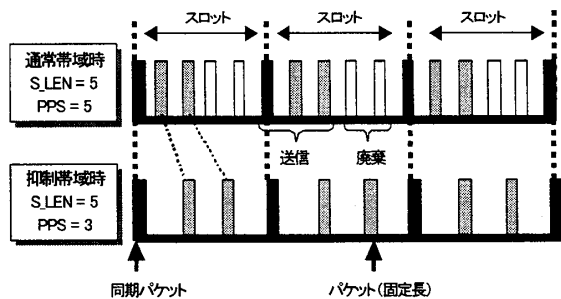


図1 BCMSの帯域制御モデル

BCMS 帯域制御モデルの重要な特性はスロットあたりの送信時間は「通常帯域時」と「抑制帯域時」では変わらないことである。すなわち、スロットの先頭パケット(同期パケット)は PPS を変更しても必ず同じタイミングで送出される。これにより動画の再生時間を保証することが可能になる。

3. SSC/BCMS の構成

3.1 SSC/BCMS の特徴と概要

SSC/BCMS は、次の点を可能にする

- ・一定帯域での送信
- ・「セッション」(フロー)ごとの帯域制御
- ・再生時間を守ったまま帯域幅を減らす
- ・OS 内に実装することで低オーバーヘッドの高速配信

上記の機能を実現するために、SSC/BCMS はストリーミングフローを「セッション」という単位で管理する。つまり、セッションは帯域制御の単位である。本機構は通信方式に UDP を用い、実装は Linux カーネルに対して行った。図2に SSC/BCMS の全体構成を示す。パケットキューはパケットの保存管理を行い、BCMS は帯域制御を行う。また SSC はセッションの管理を行い、SSC ライブラリは SSC のユーザアプリケーションへのインタフェースを提供する。以下、それぞれの構成要素について述べる。

3.2 パケットキュー

パケットキューは固定長である SSC パケットの保存管理を行う。パケットの管理は、FIFO (First In First out) で行う。キューの長さは固定長で、セッション作成時に1度だけ設定可能である。

3.3 BCMS

BCMS では主に帯域制御を行う。BCMS はパケット保存部(Keeper)と送信制御部(Sender)の2つから成る。

Keeper は、ユーザプログラムから受け取ったパケットを送信タイミングが来るまでパケットキューに保存し、満杯のときはブロッキングを行う。Sender はパケットの

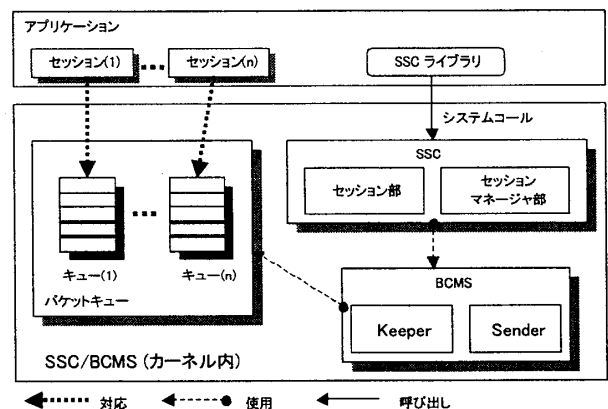


図2 SSC/BCMSの全体構成

†東京農工大学大学院工学研究科

‡東京農工大学工学部

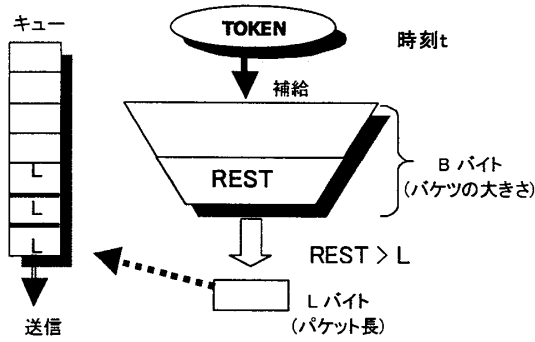


図3 Senderの送信制御

送信タイミングの制御を行う。

Sender は単一のセッションを制御し、タイマによって一定のスケジューリング間隔で呼び出される。Sender では帯域制御にトークンパケットを用いる(図3)。トークンの大きさは1回のSenderの呼び出しで送出すべきデータ量である。通常帯域時(PPS = S_LEN)のトークンの大きさをDEF(byte), Senderの呼び出し間隔をS_PERIOD(sec), メディア帯域をM_BWとすると、トークンの大きさは式(1)で求められる。

$$DEF = M_BW \times S_PERIOD$$

$$TOKEN(byte) = DEF \times (PPS/S_LEN) \dots (1)$$

このモデルでは平均データレートはトークンの補給スピード(トークンレート)で決まり、最大バースト長はパケットの大きさ(Bバイト)で決まる。

3.4 SSC

SSC はセッション部とセッションマネージャ部から構成される。SSC ではストリーミングフローを「セッション」という単位で管理する。セッションでは一つのUDPコネクションを保持し、BCMSを用いて帯域制御を行う。セッションマネージャはセッションテーブルの管理とセッションで共有するデータや状態の管理を行う。

3.5 SSC ライブラリ

SSC ライブラリはカーネル内のSSCのフロントエンドであり、ユーザプログラムはSSCが提供するインタフェースを通してストリーミングに関する制御を行う。

(1)セッションオブジェクトの生成、破棄

- ・ ssc_create(local_host, remote_host, local_port, remote_port)
- ・ ssc_destroy(ssc_fd)

ssc_create()はセッション管理構造体をカーネル内部に作成する。引数はUDP通信のエンドポイントを取る。戻り値は成功したらファイルディスクリプタを返し、失敗であれば-1を返す。

(2)セッションの確立

- ・ ssc_session(ssc_fd, bandwidth, pkt_size, slot_len, q_len)

この関数ではセッションの確立を行う。引数としてSSCのファイルディスクリプタ、メディア帯域(byte/sec)、パケットサイズ(byte)、スロット長(packet)、キュー長(packet)を指定する。

(3)パケットの送受信

- ・ ssc_transmit(ssc_fd, data)
- ・ ssc_recieve(ssc_fd, data, packetinfo)

SSCパケットの送受信を行う。引数はファイルディスクリプタとパケットへのポインタである。

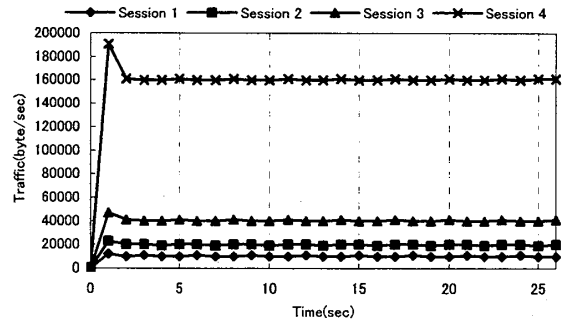


図4 一定帯域(通常帯域)で送信

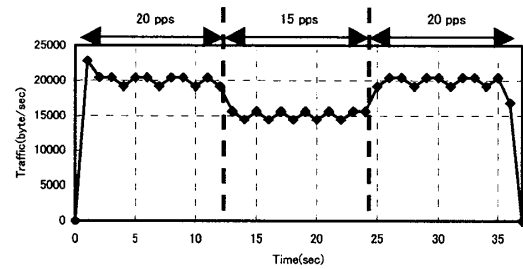


図5 PPSを途中で変更

ssc_receive()のpacketinfoという引数にはSSCパケットに付加されていたヘッダ情報が格納される。

(4)ppsの変更

- ・ ssc_change_pps(ssc_fd, pps)

この関数では、ppsの変更を行う。SSCのファイルディスクリプタと新しいppsを指定する。新しいppsはスロット長以下の値でなければならない。

4. 実験と評価

2つの実験ケースの結果を示す。第1のケースは4台のパソコンにそれぞれセッションを割り当て、10,000, 20,000, 40,000, 160,000 byte/secのメディア帯域を設定して送信を行った。パケット長は1,200バイトでパケットの破棄行わない。この結果は図4で、一定帯域で送信できていることが確認できた。第2のケースは10スロットごとにPPSを変更して計測した。結果は図5で、帯域が制御できているのが確認できる。また同じ10スロットでは送信時間が同じであることも確認できる。

5. おわりに

本論文では、ストリーミングサーバ用の帯域制御機構：BCMSとBCMSを用いてストリーミングフローを制御するSSCの設計と実装について述べた。今後はより詳細な評価とSSC/BCMS用へのメディアフォーマット変換プログラムの作成を予定している。

参考文献

- [1] 毛利 公一, 吉澤 康文, “リアルタイム通信向け予約型帯域保証機構の開発,” 第42回プログラミング・シンポジウム報告集, pp. 91-98, Jan 10 2001.