フロー制御機能を備えたアプリケーションの開発

4U－5

青柳好織*, 石田慶樹†, 牛島和夫*
*九州大学大学院システム情報科学研究科
†九州大学大型計算機センター

1　はじめに

現在、コンピュータ・ネットワークは広く普及し、利用者も著しく増加している。そのような中で、実時間通信を行なうアプリケーションへの要求が高まってきている。

実時間通信を行なうアプリケーションでは、転送される連続データの時間的制約を保持しつつ情報をいかに伝送するかが問題となる。時間的制約を満たす方法として資源予約というものがある[1]。しかし資源予約を行なったとしても、予約された資源を正しく利用しなければ、時間的制約を満足できない。また、資源を無駄に使うことにもなりかねない。そのような事態を避けるためには、予約された資源を利用するデータの流れ(フロー)を制御する必要がある。更に、各アプリケーションで取り扱うデータの特性を考慮しつつデータを伝送するためには、アプリケーションレベルでのフロー制御が必要である。

本研究では、フローを制御できるアプリケーションとして、音声および画像データを転送しつつフロー制御を行なう多対多のテレビ会議ツールの開発を目標とし、まず音声データについてフロー制御を行なうツールを開発し、アプリケーションによるフロー制御の有用性を示す。

2　実時間通信を行なうアプリケーション

実時間通信を行なうアプリケーションの性能は、次の二点で評価できる[2]。

一つは、送信側がデータを送信してから受信側がデータを受け取るまでにかかる時間である。特に遅延時間が問題となる。もう一つは、送信者側から送られたデータが正確に受信者側で再生できているかという点である。これは送信側がデータをデータグラム化して転送した時のデータグラムの遅延時間のばらつきの問題となる。

従って、実時間通信を実現するためには、ネットワークが遅延時間と遅延時間のゆらぎであるジッタをアプリケーションの要求する範囲内に抑え、アプリケーションの要求する帯域を確保しなければならない。つまり、

ユーザまたはアプリケーションが要求するQoSを保証するために、ネットワーク資源を予約する必要がある。

Development of a multimedia application with flow controler
Yoshio Aoyagi*, Yoshiki Ishida† and Kazuo Ushijima*
*Graduate School of Information Science and Electrical Engineering,Kyushu University
†Computer Center,Kyushu University

3　フロー制御

フローとは、ある一つのユーザ活動から生じ、同じQoSを要求する単方向の関連したデータグラムの流れのことをいう。

資源予約機構を用いて、アプリケーションがあらかじめ通信に必要なネットワーク資源を確保することで、取り扱うデータの時間的制約は満たされ、実時間通信が実現できる。そこで、まずアプリケーションは通信に必要な資源を明確にするためにフローの仕様を定義する必要がある。定義したフローの仕様に基づき確保すべき資源の量を決定し資源を予約する。しかし、予約した帯域以上のデータを一度に送信するなどしてデータ送信時に資源の利用状況を守れない場合、自分のフローだけでなく同じ経路上の他のフローにおいても遅延時間の増大やデータ落ちなどの不具合が発生することになる。

そこで、フロー中に転送されるデータの品質劣化を防ぐとともに他のフローのデータを不当に排除しないようにするために、ネットワーク内での予約された資源の不適当な利用を抑制する必要がある。これをフロー制御と呼ぶ。資源予約の利用による実時間通信の実現には、フロー制御は不可欠である。

フローは送信側のアプリケーションからOSを介してネットワーク上に送り出される。そして、受信側はネットワーク上のフローをOSを介してアプリケーションで受け取る。従って、ネットワーク・OS・アプリケーションの三つのレベルにおいてフロー制御を行なう必要がある。

ネットワークレベルでのフロー制御は、パケットのスケジューリングやクラス化によって行なわれる。現在、それらの機能を持つルータやスイッチが開発され、実験ネットワークが構築されつつある。

またOSレベルにおいては、複数のアプリケーションから生じたフローを一つにまとめてネットワークに送り出す作業を行なう必要がある。OSレベルでのフロー制御は、その一つにまとめられたフローに対して行なわれる。OSのレベルにおいてフローを一つにまとめて制御するのであれば、アプリケーションレベルでのフロー制御は不必要にも思える。しかし、OSだけがフロー制御を行なった場合、OSがフロー制御する時点で複数のアプリケーションのフロー間での競合が起こり、アプリケーションの特性及びアプリケーションが扱うデータの特性に依らないデータ落ちや遅延が発生する恐れがある。それは個々のデータの特性をOSが分かり得ないことに起因している。一方アプリケーションは自分が扱うデータ

の特性を知っている。従ってOSにデータをわたす前に各アプリケーション毎にフロー制御を行なうことによって、OSでのフロー制御の負荷を軽減できるとともに、各アプリケーションの特性及びアプリケーションが扱うデータの特性に合わせたフロー制御が可能になり、転送データの品質をさらに向上させることができるようになる。やはりOSだけでなくアプリケーションのレベルにおいてもフロー制御は必要である。

アプリケーションレベルでのフロー制御として行なう必要があることは、送信するデータをフローの仕様に合うように調節することである。音声データのような固定長のデータの場合は、予約した帯域に合う大きさにしたデータを一定時間毎に送信することで、フローを一定に保つことができる。また動画のように様々な外部要因によってデータ量が変化する場合もある。そこでアプリケーションではフローを一定に保つために、入力データ量の変化に合わせて入力データを適切に処理してから送信しなければならない。

4 フローを制御する音声会議ツールの実装と評価

4.1 ツールの実装

本研究では、フローを制御するアプリケーションとして、音声と映像をネットワーク上に転送しつつ、それぞれのフローを制御するテレビ会議ツールの開発を目指し、今回は音声による会議ツールの開発・評価を行なった。

本研究が目標とするテレビ会議ツールは、音声取込部・音声再生部・映像取込部・映像再生部・フロー制御部・ネットワーク・インターフェース部から構成されている。この中で今回実装したのは、映像取込部と映像再生部を除く四つのモジュールである。

まず、音声取込部が音声入力装置から音声データを取り込み、フロー制御部にわたす。音声データは固定長なので、一定間隔で正しく送信することによってフローは一定に保たれる。フロー制御部は、音声取込部から受け取ったデータを一定時間毎にネットワーク・インターフェース部にわたす。今回は一回の送信当りのデータ量を800バイト、データ送信時間間隔を0.1秒とし、実時間インターバルタイマを用いて実装を行なった。ネットワーク・インターフェース部は通信の準備とデータの送受信を行なう。音声再生部は、ネットワーク・インターフェース部からデータを受け取り再生する。もし一定時間間隔より早く到着した場合は、その間隔になるまで待った後再生する。

4.2 実験および評価

LAN上の2台のマシンの間で音声通信を行ない、その時のパケットの送信時間間隔を計算することで、一定時間毎に送信できているかを確認する実験を行なった。送信時刻の取得にはtcpdumpを用いた。

4663個のパケットを送信し、それから送信時間間隔を計算して4662個のサンプルを得た。その送信時間間隔の分布をグラフにしたものが図1である。横軸が送信時間間隔、縦軸が該当するサンプル数を表わしている。送

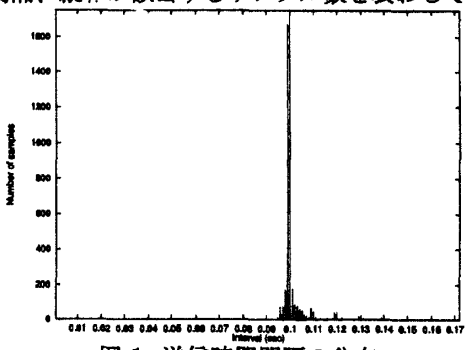

図1 送信時間間隔の分布

信時間間隔の平均値は0.102秒で、分散は4.979×10^{-5}であった。全体のうち77.13%が0.1±0.01秒台に集中しているが、タイマの精度を考慮しても、残りの約23%は、0.1秒という時間間隔から0.01秒以上ずれていることになる。このずれの原因としては、次のようなことが考えられる。

まず、OSがパケットをネットワークにわたすときの遅れ。これは、そのときの待ち行列の状態が関係するものと推測できる。次に、ネットワーク上でのパケットの衝突による遅れ。また、OSでのアプリケーションのスケジューリングによるずれが生じることもあり得る。

今回の実験では、誤差の主な原因が上記で挙げたもののうちどれであるのか、もしくは別の原因であったのかを確認するまでには至らなかった。

5 おわりに

本研究では、資源予約機構を用いて実時間通信を実現する際のアプリケーションでのフロー制御の必要性について述べた。また、実際にフローを制御するアプリケーションの開発を行ない、アプリケーションでも、フローをある程度一定に保つことは十分に可能であることを示した。

今後の課題としては、実験を重ねることによる有用性の確認と誤差の原因の追及、さらに映像データをフロー制御しながら送受信する機能の実装がある。また、本ツールは本来、資源予約された経路上での使用を前提にしているので、RSVPなど資源予約機構が実装されたネットワーク上での実験も行なっていきたいと考えている。

参考文献

[1] WIDEプロジェクト, "1994年度WIDEプロジェクト研究報告書", pp. 55–94, March 1994.

[2] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.