


QoS を考慮した適応型多重化方式*

神林 隆, 山下 高生, 小野 諭

 NTT ソフトウェア研究所

実時間性を要求するアプリケーションのトラフィックをネットワークに流す場合など、回線が低速の部分では、アプリケーションが要求する QoS を保証することが困難である。本発表では、各トラフィックの特性に応じて、送信すべきフレームを選択し多重化を行ない、要求される QoS を保証する方式を考案したので報告する。本方式は、多様な制御に統一的なメカニズムを提供するだけでなく、ハードウェア化にも適している。

1 はじめに

我々が利用するコンピュータ・ネットワーク・アプリケーションには、さまざまな QoS (Quality of Service: サービスの質) を要求するものがある。画像や音声伝送などの実時間制約の強い QoS を要求するものや、ファイル転送などの高スループットを要求するものや、仮想端末のようにビット・レートは低いが低遅延を要求するもの、などがあげられる。

ところが、現在のネットワークでは、end-to-end の接続性 (connectivity) に主眼が置かれており、データが目的地にきちんと到達することが当面の目的になっているだけである。このようなネットワーク体系では、ネットワーク資源は一元的に取り扱われ、ファイル転送や仮想端末のように、要求する QoS が全く異なるアプリケーションのデータも、同じように first-come-first-served 手法で転送され、そのトラフィックの性質は無視されている。たとえば、ファイル転送により大量のデータがネットワークに流れていると、仮想端末のデータはファイル転送のデータに邪魔され遅延が大きくなるし、音声データはとぎれがちになってしまう。

本発表では、このようなことが起きないように、各トラフィックの特性に応じて、送信するフレームを選択し多重化を行なうことにより、アプリケーションが要求する QoS を保証する方式を考案したので報告することにする。本方式のメカニズムは、多種多様な制御を統一的に取り扱えるようにするだけでなく、将来、ネットワークが

*An adaptive multiplexing method that guarantees QoS

Takashi KAMBAYASHI, Takao YAMASHITA, Satoshi ONO

NTT Software Laboratories

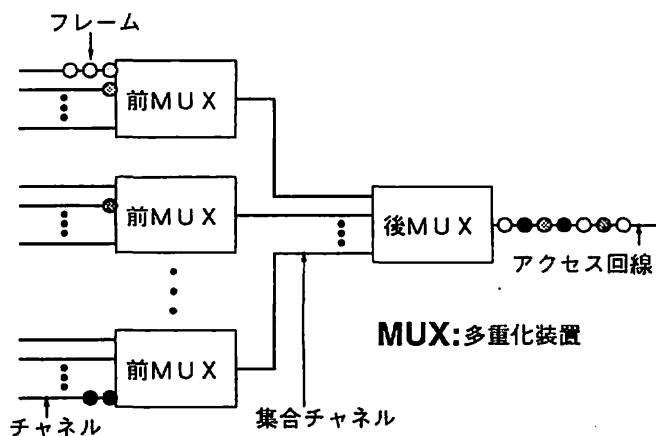


図 1: 本方式による多重化

高速になった場合のハードウェア化にも適している。

本方式では、アプリケーションの要求する QoS 保証に合った多重化を行なう時に、統計的多重化効果により、可変ビット・レートのトラフィックを効率よく転送することが可能である。また、現在、ST-II[1] などの、セッション開始時にバンド幅などの資源予約を行なうプロトコルの研究が行なわれているが、本方式は、このような資源予約をとまなうプロトコルを下位からサポートするものとして開発している。

2 本方式の特徴

本方式による多重化は、図 1 のようになる。各アプリケーションごとにチャンネルが割り当てられ、要求された QoS に従ってパラメータ値が設定されている。本節の以下では、本方式の特徴について説明することにする。

2.1 2 段階の多重化

本方式の特徴の 1 つは、多重化装置 (以下、MUX とする。) が 2 段階になっているという点である。複数のチャンネルが、前 MUX により、集合チャンネルに多重化され、さらに、複数の集合チャンネルが、後 MUX により、物理アクセス回線に多重化される。

前 MUX では、トラフィック特性の似ているチャンネルの多重化を行なう。つまり、画像トラフィックはある特定の前 MUX により多重化され、音声トラフィック、ファイル転送のトラフィック、仮想端末のトラフィックも、同じように、それぞれ別々の前 MUX により多重化され、それぞれの集合チャンネルに送信される。

後 MUX では、異なるトラフィック特性を持つ集合チャンネルを、各トラフィック特性の QoS に応じて送信すべきフレームを選択して、多重化を行ない、アクセス回線に

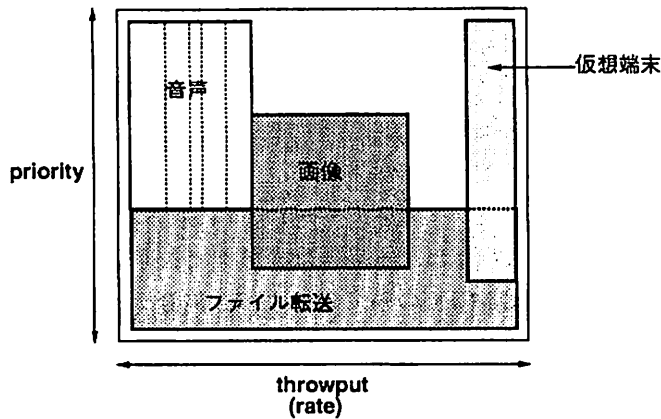


図 2: トラフィック特性に応じた STP の割り当て

送信する。

従来のように、個々のチャンネルを1つのMUXで多重化しようとする、各チャンネルごとに細かな QoS 設定を行なう必要があり、要求された QoS を保証するための admission control が複雑になり、ネットワーク全体の管理が困難になっていた。しかし、本方式のように、ある程度 QoS 特性の似たチャンネルを前 MUX でまとめることにより、後 MUX では、異なる QoS 特性を持つ集合チャンネル同士のスタティックな多重化を行ない、前 MUX では、似かよった QoS 特性を持つチャンネル同士のダイナミックな多重化を行なうだけでよくなる。特に、後 MUX では、各トラフィック特性に合わせた広域的なネットワークの設計を行なうことが可能となる。このように、MUX の目的の違いをはっきりさせて、admission control を容易に行なえるようにしている。

ネットワーク設計のイメージは、図2のようになる。QoS 特性の似たトラフィック (STP: Similar Traffic Pass, 図 2の実線の矩形) は、それぞれトラフィック特性に合わせて、ネットワーク管理者によってスタティックに割り当てられる。たとえば、ファイル転送の STP には、スループットを広く取り、優先度を低く取る。逆に、仮想端末の STP には、優先度を高く、しかも、高く取り、スループットは狭く取る。後 MUX では、このようにして配置された STP の多重化を行なう。各 STP の中では、割り当てられたスループットをダイナミックに細分化 (図 2の破線) して、それを個々のチャンネルに割り当てる。前 MUX では、これらのチャンネルの多重化を行なう。

2.2 後向きの多重化

本方式の特徴の2つめは、多重化が要求駆動型であり、後向きに行なわれるということである。

本方式のように多重化が2段階になっている場合、多重化が前向きに行なわれる場合と、後向きに行なわれる場合の2つが考えられる。多重化が前向きに行なわれる

場合、前 MUX で到着したフレームの多重化を行ない集合チャンネルに送信したとしても、後 MUX ではその集合チャンネルが送信可能になっているとは限らないので、集合チャンネルに送信されたフレームの多重化が待たされたりする。この場合、後 MUX の各集合チャンネルにもキューが必要になる。さらに、後 MUX は、フレームが送信された集合チャンネルにのみ注目して多重化を行なうことになるので、トラフィック特性に応じた、ネットワーク全体のバランスをスタティックに取ることが難しくなる。

逆に、多重化が後向きに行なわれる場合は、アクセス回線にフレームを送信しようとする時に、後 MUX、前 MUX の順で送信すべき集合チャンネル、チャンネルを選択してゆくの、各 MUX でトラフィック特性に応じた QoS を保証することが容易である。まず、後 MUX で、スタティックに割り当てられた各 STP のトラフィック特性に応じて、送信すべき集合チャンネルを選択する。それから、その集合チャンネルに接続されている前 MUX で、ダイナミックに、各チャンネルのトラフィック特性に応じて、送信すべきチャンネルを選択する。このように、スタティックな特性に応じた多重化を行なってから、ダイナミックな多重化を行なうので、ネットワーク全体の管理が行ないやすい。また、後向きの多重化では、集合チャンネルに送信されたフレームが後 MUX で待たされることはない、集合チャンネルでのキューも必要なくなる。

以上の理由により、本方式では後向きの多重化を採用している。

2.3 多様な制御をハードウェア化容易な方法で実現できる多重化

本方式の特徴の最後は、以下に説明する 5 つのパラメータを組み合わせることによって、多種多様な制御の実現が簡単に行なえることである。しかも、各パラメータはレジスタとして実現できるので、ハードウェア化が容易であり、将来の高速ネットワークでも十分適用可能である。

制御方法の例としては、優先度制御、time-window、packet-spacing[2]、work-conserving/non-work-conserving[3]などがあげられる。優先度制御は、フレームに優先度を与え、優先度の高いものから送信するというものである。time-window では、ある一定時間内に送信できるデータ量を $a\Delta t + b$ と決めて、それを越えないようにフレームの送信を行なう。packet-spacing では、フレームを送信し終ったあと、一定時間はフレームを送信しないようにして、フレームの送信間隔をあけるものである。work-conserving とは、フレームがキューイングされている限り、フレームを送信しようとするものであり、non-work-conserving では、たとえフレームがキューイングされていたとしても、フレームが送信可能状態になるまで、送信を行なわない。このように、さまざまな制御方法が現実存在しているわけだが、本方式では、これらの制御を実現するための統一的なメカニズムを提供しており、パラメータ値の設定を変更することにより、これらを容易にシミュレートすることが可能である。

本方式では、各アプリケーションが要求する QoS を表すために、2 段階の MUX

¹ここで、 a は平均レート、 b はバースト量である。

の入力回線である各チャンネル／集合チャンネルに、5つの制御パラメータ値を設定する。この5つの制御パラメータは、MUXの出力回線である集合チャンネル／アクセス回線に送信するフレームを選択するときに、選択の基準となる。後MUXの入力回線である集合チャンネルには、トラフィック特性の異なるSTPに対してQoS制御のためのパラメータをスタティックに与え、前MUXの入力回線であるチャンネルには、各STP内の個々のチャンネルのQoSに応じてパラメータを設定する。

各チャンネル／集合チャンネルは、それぞれ、チャンネル時間／集合チャンネル時間と呼ばれる論理的な時間を持っている。チャンネル時間／集合チャンネル時間の最も小さいチャンネル／集合チャンネルが、優先度が最も高く、送信すべきチャンネル／集合チャンネルとして選択される。また、アクセス回線もアクセス回線時間と呼ばれる論理的な時間を持っている。アクセス回線時間は、リアルタイムで進行する時間であり、MUX全体の時間を表している。チャンネル時間／集合チャンネル時間は、チャンネル／集合チャンネルのその時点までの総送信量を表すものと考えることができ、フレームを送信するたびにチャンネル時間／集合チャンネル時間は、フレームを送信した分だけ進められる。

以下、説明を簡単にするために、前MUXに対するチャンネル、後MUXに対する集合チャンネルを入力回線と呼び、前MUXに対する集合チャンネル、後MUXに対するアクセス回線を出力回線と呼ぶことにする。

- **rate** は、入力回線の平均送信レートを定義しており、入力回線からフレームを送信するたびに、入力回線時間は、(送信するフレーム長 / rate) だけ増加する。

rateの値の大きい入力回線ほど、フレームを送信しても入力回線時間の増え方がゆっくりなので、たくさんのフレームを高速に送信することが可能である。高スループットを要求するアプリケーションに対応する入力回線は、rate値を大きくする必要がある。

rateは、time-windowを実現する時に、 $a\Delta t + b$ の a として用いられる。

- **credit** は、入力回線にフレームがバースト的に到着した場合、その入力回線のフレームを出力回線に送信できるバースト量を表している。入力回線時間と出力回線時間との差がcreditを越えた入力回線は、フレームがキューイングされていたとしても送信に規制がかけられ、出力回線時間が進行して、入力回線時間と出力回線集合との差がcreditより小さくなるまで、フレームを送信できない。バースト性の高いデータを転送するアプリケーションは、credit値を大きくする必要がある。

creditは、time-windowを実現する時に、 $a\Delta t + b$ の b として用いられる。この他にも、maxと組み合わせて、packet-spacingやwork-conserving/non-work-conservingのためにも用いられる。

- **max** は、入力回線時間と出力回線時間との差がとりうる最大値である。

maxをcreditより小さい値に設定することにより、入力回線時間と出力回線時間との差はmaxで押えられるので、creditを越えることはなくなる。この場

合、フレームの送信が待たされることがなくなるので、その入力回線は work-conserving で多重化される。

逆に、max を credit より大きい値に設定することにより、たくさんのフレームを送信した入力回線では、入力回線時間と出力回線時間の差が credit を越えてしまう時がある。この場合、出力回線時間が進んで、入力回線時間と出力回線時間との差が credit より小さくなるまで、フレームを送信できないので、non-work-conserving になる。また、max と credit の差を、packet-spacing における、フレームの送信間隔と見なすことができる。

- **offset** は、到着したフレームが実際に到着した時間より早く到着したように見せかけるパラメータである。これにより、低レートでスパースにフレームが到着するようなアプリケーションに割り当てられた入力回線のフレームを優先的に送信することが可能になる。

音声データを転送するチャンネルに大きな offset 値を設定することにより、音声データのフレームを画像データに邪魔されることなく、送信することが可能になる。また offset は、仮想端末のようなスパースにデータ転送を行なうアプリケーションの場合も、ネットワークが混雑している場合に、優先度を高くして、遅延を短くする制御が可能である。

offset は、主に、優先度制御を実現する時に用いられる。

- **min** は、入力回線時間と出力回線時間との差がとりうる最小値である。
min は、優先度制御に用いられる。

3 本方式の詳細

3.1 MUX の構成

前 MUX と後 MUX は同様な構成をしており、入力回線であるチャンネル/集合チャンネルごとに以下の値を持っている。

- 前節で説明した5つのパラメータの値
- チャンネル時間/集合チャンネル時間 (チャンネル/集合チャンネルの論理的な時間を表しており、フレームの到着、送信のたびに新しい値に更新される。)
- not-empty フラグ (送るべきフレームがチャンネル/集合チャンネルにあるかどうかを表している。チャンネルの場合は、バッファにフレームがキューイングされているかどうかで判断し、集合チャンネルの場合は、多重化するチャンネルの少なくとも1つが not-empty であるかどうかで判断する。)
- within-credit フラグ (入力回線時間と出力回線時間との差が credit より小さいかどうかを表している。チャンネルの場合は、チャンネル時間と集合チャンネル時間との差を計算し、集合チャンネルの場合は、アクセス回線時間と集合チャンネル時間との差を計算する。)

- work-conserving フラグ (work-conserving であるかどうかを表している。本方式においては、後 MUX は non-work-conserving で多重化を行なうので、集合チャンネルでは常に 0 であり、前 MUX は work-conserving で多重化するので、チャンネルでは常に 1 である。)

また MUX は、すべてのチャンネル／集合チャンネルで共有、参照するアクセス回線時間を持っている。アクセス回線時間は、リアルタイムで進行する時間であり、MUX 全体の時間を表していると考えられる。

各チャンネル／集合チャンネルが、ready(送信可能である)かどうかは、以下のよう求められる。

$$\text{ready} = \text{not-empty} \wedge (\text{work-conserving} \vee \text{within-credit})$$

つまり、MUX のチャンネル／集合チャンネルのどれかにフレームがキューイングされており (not-empty)、かつ、フレームがキューイングされている限り送信しようとし (work-conserving)、non-work-conserving だとしても、入力回線時間と出力回線時間の差が credit 値より小さければ (within-credit)、そのチャンネル／集合チャンネルは ready となり、集合チャンネル／アクセス回線への送信が可能となる。

3.2 前 MUX にチャンネルからフレームが到着した時の動作

前 MUX にチャンネル i からのフレームが到着した場合、フレームはそのチャンネル i のバッファにキューイングされ、チャンネル i の not-empty フラグを立て、接続されている集合チャンネル j の not-empty フラグも立てる。さらに、その時点でのチャンネル時間 i と集合チャンネル時間 j との差が credit_i より小さい場合は、チャンネル i の within-credit フラグも立てる。また、チャンネル時間 i は、その時点でのチャンネル時間 i と、集合チャンネル時間 j からチャンネル i の offset 値を引いた値の大きい方に更新される。

前 MUX :

```

if チャンネル  $i$  にフレームが到着 then
    フレームをバッファ  $i$  にキューイング ;
    not-empty $_i$  ← 1 ;
    接続されている集合チャンネル  $j$  の not-empty $_j$  ← 1 ;
    if (チャンネル時間  $i$  - 集合チャンネル時間  $j$ ) < credit $_i$  then
        within-credit $_i$  ← 1 ;
    チャンネル時間  $i$  ←
        max(チャンネル時間  $i$ , (集合チャンネル時間  $j$  - offset $_i$ )) ;

```

3.3 後 MUX がフレームをアクセス回線に送信する時の動作

フレームをアクセス回線に送信する時、つまり、実際に多重化を行なう場合、まず、後 MUX で、各集合チャンネルが ready であるかどうかを調べ、ready 状態になっている集合チャンネルの中で最も集合チャンネル時間の少ない集合チャンネル j を選択

する。本方式においては、後 MUX は non-work-conserving で多重化を行なうので、work-conserving フラグは常に下りている。つまり、アクセス回線時間と集合チャネル時間の差が credit を越えている場合は、送るべきフレームがあっても決して ready にはならないので、その集合チャネルが選択されることはない。その後、後 MUX は、選択した集合チャネル j を出力回線にもつ前 MUX j に、選択されたことを通知する。

後 MUX :

各集合チャネルが ready かどうかを調査 ;
ready で、最も集合チャネル時間の少ない集合チャネル j を選択 ;
接続されている前 MUX j に選択したことを通知

通知を受けた前 MUX j は、各チャネルが ready であるかどうかを調べ、ready 状態になっているチャネルの中で最もチャネル時間の少ないチャネル i を選択する。本方式では、前 MUX は work-conserving で多重化を行なうので、work-conserving フラグは常に立っている。つまり、集合チャネル時間とチャネル時間の差が credit を越えていても、そのチャネルにフレームがキューイングされている限りは、常に ready になっている。

このようにして、最終的に、アクセス回線に多重化するべきフレームを持つチャネル i が選択され、チャネル i の中からは、キューイングされているフレームの中で、最も早く到着した、古いフレームが選択される。前 MUX j は、選択されたフレームを集合チャネル j に送信し、多重化する。その時に、チャネル i のチャネル時間に、選択されたフレームを送信するのに必要な時間 (送信するフレーム長 / rate i) を足し込み、チャネル時間を進める。

この送信により、チャネル i にキューイングされているフレームがなくなった場合、前 MUX j はチャネル i の not-empty フラグを下ろす。さらに、各チャネルが not-empty であるかどうかを調べ、各チャネルの not-empty フラグがすべて下りている場合にのみ、出力回線である集合チャネル j の not-empty フラグを下ろす。

前 MUX j :

各チャネルが ready かどうかを調査 ;
ready で、最もチャネル時間の少ないチャネル i を選択 ;
チャネル i の最も古いフレームを集合チャネル j に送信 ;
チャネル時間 $i \leftarrow$ チャネル時間 $i +$ (送信するフレーム長 / rate i) ;
if チャネル i にキューイングされているフレームの数 = 0 then
 not-empty $i \leftarrow$ 0 ;
 各チャネルが not-empty かどうかを調査 ;
 if すべての not-empty = 0 ;
 not-empty $j \leftarrow$ 0 ;
 end
end
end

後 MUX は、集合チャンネル j からそのフレームを受けとり、アクセス回線に送信、多重化する。その時に、集合チャンネル j の集合チャンネル時間に、フレームを送信するのに必要な時間 (送信するフレーム長 / rate_j) を足し込み、集合チャンネル時間を進める。

後 MUX :

集合チャンネル j に送信されたフレームをアクセス回線に送信 ;

集合チャンネル時間 $j \leftarrow$

集合チャンネル時間 $i + (\text{送信するフレーム長} / \text{rate}_j)$

4 おわりに

本発表では、さまざまな QoS 特性をもつアプリケーションのトラフィックをネットワークに転送する場合に、各トラフィックの特性に応じて、送信すべきフレームを選択し多重化を行ない、要求された QoS を保証する方式について報告した。本方式は、従来の多重化方式と比べて、

- 多重化装置が 2 段階になっており、後 MUX では、トラフィック特性の異なる STP を多重化し、前 MUX では、特性の似たトラフィックを多重化する。
- 多重化が要求駆動型で行なわれ、まず、後 MUX でスタティックな多重化を行なってから、前 MUX でダイナミックな多重化を行なう。
- パラメータ値の設定を組み合わせることより、現実に存在するさまざまな制御を実現することが可能であり、かつ、ハードウェア化も容易である。

などの特徴がある。また、本方式による多重化装置の構成と動作アルゴリズムについても報告を行なった。

今後は、本手法をシミュレート実験し、その有効性について、評価を行なう予定である。

謝辞

本研究をご支援くださる、NTT ソフトウェア研究所の後藤滋樹ソフトウェア基礎技術研究部長、および、小川裕グループリーダーに感謝いたします。

参考文献

- [1] C. Topolcic: "Experimental Internet Stream Protocol, Version 2 (ST-II)", CIP Working Group, *FRC 1190*, 1990.
- [2] Sylvie Dupuy, Wassim Tawbi and Eric Horlait: "Protocols for high-speed multimedia communications networks", *Computer Communications*, Vol.15, No.6, pp.349-357, 1992.
- [3] Hui Zhang and Srinivasan Keshav: "Comparison of Rate-Based Service Disciplines", *SIGCOMM '91 CONFERENCE*, pp.113-121, 1991