

帯域予約開始までの待ち時間を考慮した RSVP の提案

Proposal of RSVP which consider maximum waiting time for bandwidth reservation

池邊 隆 本多 弘樹 弓場 敏嗣

Takashi, Ikebe Hiroki, Honda Toshitsugu, Yuba

電気通信大学

University of Electro-Communications

概要

従来の IP ネットワークはコネクションレス・ベストエフォート型ネットワークであり、QoS (Quality of Service) の保証が難しかった。End-to-End の QoS を保証するために、IETF (Internet Engineering Task Force) では RSVP (Reservation Protocol) などの帯域予約プロトコルが提案されているが、要求されたすべての帯域予約要求を保証するプロトコルは実現されていない。我々は帯域予約の方法として「待時方式」を考案し、確実に帯域予約要求を満たす方法を検討してきた。この待時方式により、全ての帯域予約通信が特定の経路間を通過し、その経路間に予約可能な帯域以上の量の帯域予約要求が生じるような状況下では、従来の RSVP に比較して良好な結果を得られることが実証された。一方、インターネットなどの広域ネットワークでは、帯域予約は常に特定の経路間を通過するわけではなく、必ずしも待時式帯域予約通信方式が有効ではない。本研究ではこれら課題を解決する方式を提案し、シミュレーションにより本方式の有効性を検証する。

1 はじめに

IETF において、End-to-End での QoS を保証するために、RSVP (Resource Reservation Protocol) [1][2] などの帯域予約プロトコルの提案がなされている。このような帯域予約プロトコルを用いることにより、ルータ等中継ノード間の帯域を予約することが可能である。しかし一時期に総帯域以上の量の帯域予約要求があった場合には、元来呼損系である RSVP ではすべての帯域予約要求が満たされるわけではない。またそのような状況で予約要求を繰り返したとしても、RSVP では早い者勝ちで帯域予約が行われるため、条件によっては広帯域の予約を要求する通信はいつまでも帯域予約できないことさえある。

これに対し、我々は呼損系である RSVP のこの欠点を補完する方法として、待時式帯域予約通信方式[3]を提案した。しかしながら、待時式帯域予約通信方式では、全ての帯域予約通信が特定の経路間を通過し、その経路間において一時的に総帯域以上の量の帯域予約

要求が生じるような状況を前提としており、インターネットなどの広域のネットワークを考えたときに次の理由で必ずしも有効ではない。それは待時式帯域予約通信で経路途中にて順番待ちを行う際、それまでの経路の帯域を確保したままになってしまい、混雑する経路間を使用しない他の帯域予約通信が無用に待たされることになる可能性があるためである。そこで本研究では、待時式帯域予約通信方式のこの問題点を、全リンクの帯域予約の準備が整った時点で一括して帯域予約を行う方法で解決する方式を提案する。さらに本方式を NS-2 (Network Simulator)[4]および RSVP/ns[5] 上に実装し、検証を行っている。

2 RSVP[2]

RSVP はインターネット上の QoS を要求する特定の通信に対して帯域予約を行うことができるプロトコルで、帯域予約要求を経路上のすべて中継ノードに伝達することによって帯域予約を実現する。また RSVP はソフトステートと呼ばれる構造を持ち、帯域予約状態

の生存時間を限定しており、予約状態を維持するには、送信側から Path メッセージを、受信側から Resv メッセージを定期的に送信する必要がある。予約状態の生存時間を過ぎると、予約状態は自動的に削除される。このことにより、ネットワークがすでに使われなくなった帯域予約によって無駄に使用されることはない。また帯域予約ができなければ、ResvErr メッセージを Resv メッセージの送信元に返す。

RSVP の基本的な動作を図 1 と以下の①～⑧に示す。

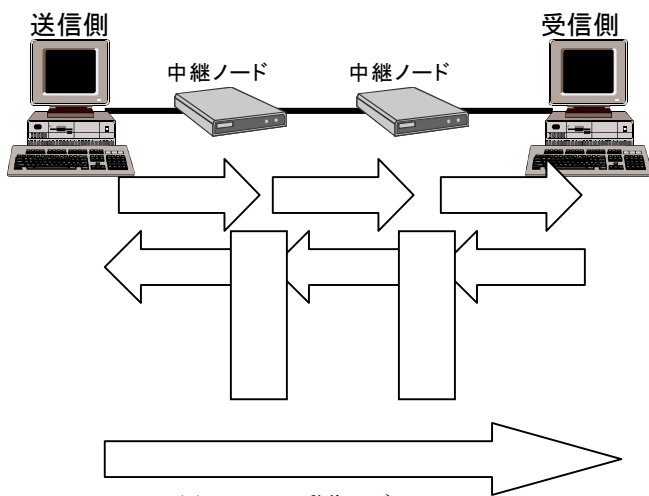


図1 RSVP 動作モデル

- ① まず送信側ホストから受信側ホストに対して Path メッセージを送信する。
 - ② Path メッセージを受信した受信側ホストで中継ノード2に対して Resv メッセージを送信する。
 - ③ 中継ノード2は中継ノード1までの経路の帯域予約を行う（④へ）。
 - ④ 帯域予約が行えない場合には、ResvErr メッセージを受信側ホストへ返す（終了）。
 - ⑤ 帯域予約ができれば、中継ノード2は中継ノード1に対して Resv メッセージを送信する。
 - ⑥ 中継ノード1は送信側ホストまでの経路で□と同様の動作を行う。
 - ⑦ 帯域予約が行えれば、中継ノード1は送信元に対して Resv メッセージを送信する。
 - ⑧ Resv メッセージを受け取った送信側ホストは、データ送信を開始する。帯域予約状態を維持するために、定期的に Path メッセージおよび Resv メッセージを再送する。通信終了後、帯域予約を解除する。（終了）
- ResvErr メッセージを受取った、受信側ホストに

おいては、予約解除を示す ResvTear メッセージを送信し、予約状態を解除してベストエフォート型通信を行うか、②～⑦の動作を初めから繰り返すかを選択する。

一般に中継ノードにおいて、送信されるデータは様々な Packet Queuing によって Queuing されて送信される。RSVP と Packet Queuing の動作を図 2 と以下の①～④で示す。

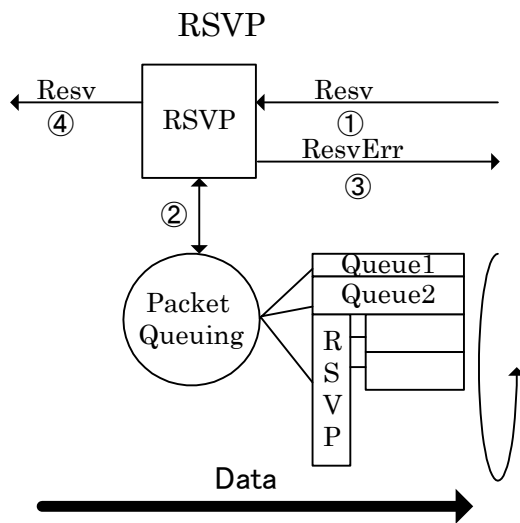


図2 RSVPの詳細図

- ①中継ノードは Resv メッセージを受信。
- ②中継ノードは Resv メッセージが要求する帯域が予約できるかどうかを、現状の Packet Queuing を調べて判断する。要求される帯域を予約可能であれば、Packet Queue を作成する。
- ③予約不可能であれば、Resv メッセージの送信元に ResvErr メッセージを送信する。
- ④Resv メッセージを次の中継ノードへ送信。

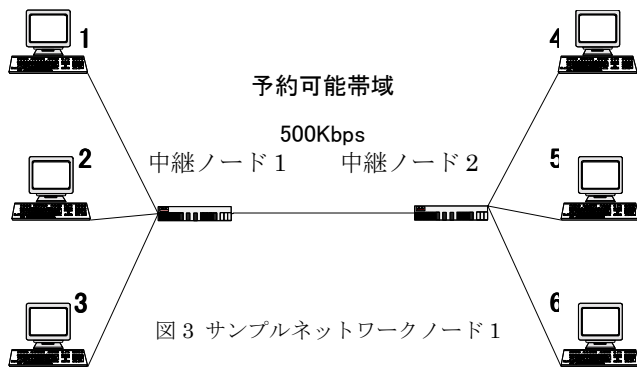
図2では、中継ノードにおいて、Resv メッセージを受信し、次の中継ノードへ Resv メッセージを送信するまでの動作を示している。

3 帯域予約開始までの待ち時間を考慮した RSVP

3.1 RSVP の問題点

RSVP は元来呼損系であるため、複数のホストやアプリケーションが早い者勝ちで帯域予約が行い、帯域予約の総量が予約可能帯域を越えた状況では、それ以

降の帯域予約要求は単に棄却される。要求が棄却された場合、要求送信者は帯域予約をあきらめベストエフォートで通信を行うか、再度予約要求を送信する。要求が受理されるまで要求を繰り返す場合、次に示すように条件によっては広帯域の予約を要求する通信はいつまでも帯域予約できないことがある。



例えば、図3のようなネットワークノードがあるとき、ノード1から4に対して、100kbpsの帯域予約通信を行えば、残りの予約可能帯域は400kbpsとなる。ここにノード2から5に対して450kbpsの帯域予約要求を送信すれば、中継ノード間の帯域予約を行うことができず、ResvErrとなる。ここで、ノード2から5への帯域予約要求を繰り返し送信するとする。その間にノード3から6に対して200kbpsの帯域予約要求が入ってしまうと、帯域予約は成功する。このような小さな帯域予約が繰り返されている限り、ノード2から5に対しての450kbpsの帯域予約はいつまでも待ち続けなくてはならないという問題が生じる。

3.2 待時式帯域予約通信方式[3]

小さな帯域の予約が広帯域の予約より優先されるのは理にかなうことではあるが、一方、広帯域の予約も有限時間内に行えるようにしたいというニーズもある。そこで予約要求を棄却させずに順番待ちさせることにより、有限時間内に帯域予約を行うことができる待時式帯域予約通信方式を提案してきた。これをRSVPの付加サービスとして提供し、しばらく待ち続けるとしても帯域予約を行いたいユーザは、このサービスを選択することにより、前述の問題を解決することができる。

● 動作概要

待時式帯域予約通信方式はRSVPと同様にホップ毎に帯域予約を行い、途中経路における予約可能な通信帯域が枯渇あるいは一定の負荷率に達した場合、各中継ノードで順番待ちを行う。

ただし、事前に静的に決められたリンクの最大予約可能帯域以上の帯域予約要求が行われた場合には、予約要求は棄却される。

● 動作詳細

待時式帯域予約通信方式の機能を実現するためには、各中継ノードが「Session List」とよぶ順番待ちリストをもち、帯域予約の順番待ちの管理を行う。帯域が予約可能になった場合にSession Listに従って帯域予約を行えば、広帯域幅を要求する通信においても、いつまでも待たされることなく、有限の時間内に通信を行うことができる。Session Listは次節で述べる「重み付けソート」により、定期的にソートされる。また中継ノードはSession Listを経路表のそれぞれの経路の数だけ持ち、中継ノードからの出方路の経路の帯域を管理する。順番待ちを行う際、中継ノードはエラーメッセージであるResvErrメッセージを送信するのではなく、順番待ち状態であるResvWaitingメッセージを送信する。ResvWaitingメッセージを受信したユーザサイドのアプリケーション等が順番待ちを行うかどうかを判断する。

また待時式帯域予約通信方式はRSVPと同様にホップ毎に帯域予約を行う。従って経路途中で順番待ちを行う際、すでに帯域予約できた経路に関しては、帯域予約状態を維持する状態になる。そのため順番待ちを行っている間、受信側ホストは定期的にResvメッセージを再送し、帯域予約状態を維持すると同時に、順番待ち状態を維持する。順番待ち状態は、生存時間を持ち、生存時間を過ぎた場合は、順番待ち状態は破棄され、不必要な順番待ちが残ることを防ぐ。

なお、すでに予約されている経路の帯域において、実際のデータの送信は行われていない。一見これは、帯域を無駄遣いしているように思えるが、実際には中継ノードで使用されるPacket Queuing (WFQ or CBQ など)においては空キューがある状態だけになる

ため、単純にスキップされる。このとき、他の帯域予約通信はその帯域を使用する事はできないが、ベストエフォート型通信がその部分の帯域を使用することが出来る。

● 実装詳細

待時式帯域予約通信方式の実装において RSVP との主な変更点は、中継ノードにおいて Resv メッセージを受け取り、その Resv メッセージを処理し、Resv メッセージを次の中継ノードへ送信する際の動作にある。

待時式帯域予約通信方式で Resv メッセージを処理する際の動作例を次の図 4 と以下の①～⑤に示す。

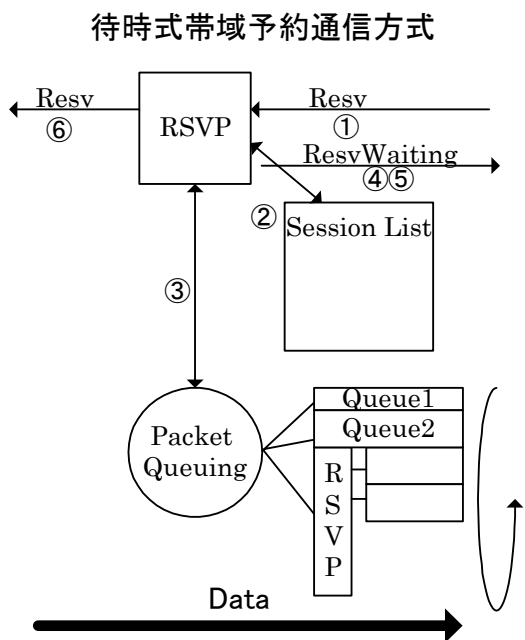


図 4 待時式帯域予約通信方式の詳細図

- ①中継ノードは Resv メッセージを受信する。要求される帯域が、事前に静的に決定された予約可能帯域以上であれば、ResvErr を送信する。
- ②中継ノードは Resv メッセージの情報を Session List に追加する。

Session List の先頭の場合：

Resv メッセージが要求する帯域が予約できるかどうかを Packet Queuing を調べて判断する。

③要求される帯域が予約可能であれば、Packet Queue を作成し Session List から帯域予約情報を消去する。また次の中継ノードに Resv メッセージを送信する。(⑥)

④予約不可能であれば、Resv メッセージの送信元に ResvWaiting メッセージを送信し、Packet Queue

作成の順番待ちを行う。

Session List の先頭ではない場合：

⑤Resv メッセージの送信元に ResvWaiting メッセージを送信し、Packet Queue 作成の順番待ちを行う。

⑥中継ノードは定期的に Session List のソートおよび、帯域の使用状況を調査し、Session List の先頭の帯域予約が要求する帯域が予約可能であれば、Packet Queue を作成し次の中継ノードに Resv メッセージを送信する。

3.3 Session List のソートアルゴリズム

Session List に保持された順番に帯域予約を行う場合、ある要求Aより Session List 上で後ろの要求Bに対しては、仮に回線にその要求Bの要求帯域分の空きがあっても、帯域予約を行うことができない。この状況を回避するために、各要求に対して要求する帯域幅 b と、その中継ノードでの待ち時間 T を組み合わせた重み w を式 1 に従ってつけ、その重みをキーとして Session List のソート(以降、重み付けソートと呼ぶ)を行うようにする。

$$w = T + P \times \left(\frac{B}{b}\right) \quad (1)$$

- w : 重み
- T : 中継ノードでの待ち時間
- B : 経路の最大予約可能帯域
- b : 通信が予約する帯域
- P : 各リンクの特性に応じた重み

ここで、 B/b が 1 を越える場合には、予約要求が予約可能帯域を超えることになり、その要求は棄却される。また各リンクの特性に応じた重み P は、そのリンクをどの程度の帯域予約通信が通信するかにより決める。

この重み P は、各リンク毎に決定することが可能であるが、たとえば、企業ネットワークのような複数のノードが集合する単位でポリシーに従って決定されるのが望ましい。また複数のソートアルゴリズムを任意に組み合わせることも可能である。

3.4 待時式帯域予約通信方式の課題

呼損系であるRSVPの欠点を補完することが可能な待時式帯域予約通信方式であるが、待時式帯域予約通信方式は全ての帯域予約通信が特定の経路間を通過し、その経路間において混雑が生じるような条件下での有効な動作を行うように考案されている。

これは、例えば企業間ネットワークなどで、遠隔地に離れた企業内LANをVPNなどで接続し、使用する場合には有効である。しかし、インターネットなどの広域のネットワークを考えた場合、全ての帯域予約通信が特定の経路間を通過するわけではない。

この場合、次の理由で必ずしも待時式帯域予約通信は有効ではない。待時式帯域予約通信はRSVPと同様にResvメッセージに伴ってホップ毎に帯域予約を行う。そのため経路途中で順番待ちを行う際、それまでの経路の帯域を確保したままになり、混雑する経路間を使用しない他の帯域予約通信が、確保されている帯域を使用できず、無用な待ちとなる問題が発生する。

次の表1及び図5で、この無用な待ちの問題を例示する。

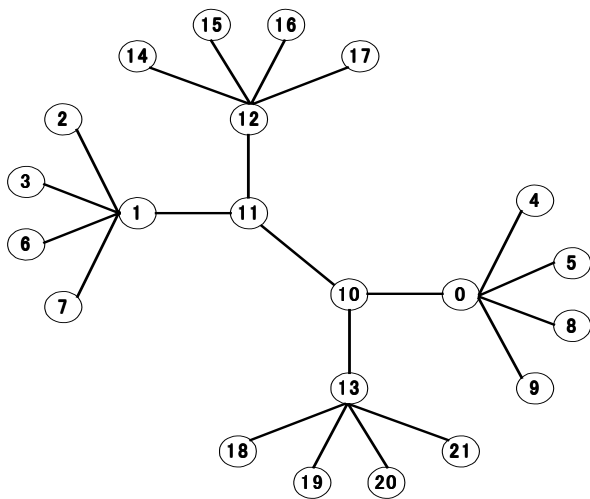


図5 サンプルネットワークノード2

表1 サンプルネットワークノード2の条件

帯域予約通信	受信側ノード	送信側ノード
A	2	4
B	3	14
その他	ノード11-ノード10間を使用	

いずれの帯域予約通信も最大予約可能帯域を予約す

るものとする。

表1、図5の条件で、まず帯域予約通信Aが受信側のノード2から、ホップ毎に帯域予約を行う。

ノード11-ノード10間で、すでに他の帯域予約通信が帯域を使用しているため、帯域予約通信Aはノード11-ノード10間で順番待ちを行う。

ここで帯域予約通信Bが受信側ノード3からホップ毎に帯域予約を行う場合、ノード1-ノード11間において、すでに帯域予約通信Aが帯域予約を行っているためにノード1-ノード11間において順番待ちを行う。この状態では、仮にノード11-ノード12間及び、ノード12-ノード14間が帯域予約可能な状態でも、通信をしていない帯域予約通信Aが、ノード1-ノード11間の帯域を確保しているために、帯域予約通信Bが通信を行うことができない。

この問題は、本質的に待時式帯域予約通信方式がRSVPと同様、Resvメッセージに伴ってホップ毎に帯域予約を行うことにより生じる。

3.5 帯域予約開始までの待ち時間を考慮したRSVP

前述の待時式帯域予約通信方式の課題を解決するためには、Resvメッセージに伴ってホップ毎に帯域予約を行う動作を変更する必要がある。

本研究では、この課題を解決することを可能とする「帯域予約開始までの待ち時間を考慮したRSVP」を提案する。また本方式は、IntServ[8][9]のGuaranteed Service [10]を想定している。

RSVP および、待時式帯域予約通信方式との本質的な動作の違いとしては、Resvメッセージに伴ってホップ毎に帯域予約を行うのではない点、またデータ送信を開始する条件が異なる。以下にこの点を詳述する。

動作概要

本方式においては各中継ノードにおける帯域予約をResvメッセージでホップ毎に行うのではなく、各中継ノードにおいては帯域予約の可能性がどうかの判別を行い、全経路帯域が予約できることを確認した後、データ送信を行う直前に全経路の帯域予約を一括して行う。

● **動作詳細**

中継ノードは Resv メッセージを受信すると、その情報を一旦 Session List に追加し、要求された帯域幅がその時点で予約可能帯域以下の場合、中継ノードは帯域予約を行うのではなく、Bandwidth Control と呼ぶ、帯域予約管理部分に仮想 Queue を作成することによって帯域予約の可能性の判別を行い、Resv メッセージを次の中継ノードへ送信する。仮想 Queue が作成されたことによって、その時点での予約可能帯域は変化しない。その時点での予約可能帯域には実際に帯域予約が行われた帯域のみを反映する。

要求された帯域幅がその時点での予約可能帯域以上の場合、その中継ノードにおいて、Bandwidth Control に仮想 Queue を作成するための順番待ちをおこない、ResvWaiting メッセージを送信する。

中継ノードは定期的に Session List のソートおよび、帯域の使用状況を調査し、Session List の上位から要求された帯域幅がその時点での予約可能帯域幅以下のものがあるかを調べ、予約可能帯域幅以下のものがあるならば Bandwidth Control に仮想 Queue を作成する。

最終的に送信側まで Resv メッセージが到達すると、送信側ノードは中継ノードへ帯域予約を行わせる「帯域予約命令」を各中継ノードにブロードキャストし、各中継ノードは帯域予約命令を受信して各経路の帯域予約を行う。

中継ノードは一定期間内に受信した帯域予約命令に対し Session List の先頭から帯域予約を行っていく。

全経路の帯域予約が行われると、送信側ノードは受信側ノードへデータ送信を開始する。データ送信終了後、各 Session List から帯域予約情報を、Bandwidth Control から仮想 Queue を削除する。

仮想 Queue は、Resv メッセージを受け取った時点での帯域予約の可能性判別に基づいて作成されているので帯域予約命令を受信した時点では帯域予約が行えないものも出てくる。そのような場合には他の中継ノードでは一旦、経路の帯域予約を開放し、再度 Session List のソートに従って、帯域予約を行う。

● **実装詳細**

本方式の実装において待時式帯域予約通信方式との主な実装に置ける変更点は、中継ノードにおいて Resv

メッセージを受け取り、その Resv メッセージを処理し、Resv メッセージを次の中継ノードへ送信する際の動作にある。

図 6 と以下の①～⑦に中継ノードにおける本方式の動作過程を示す。

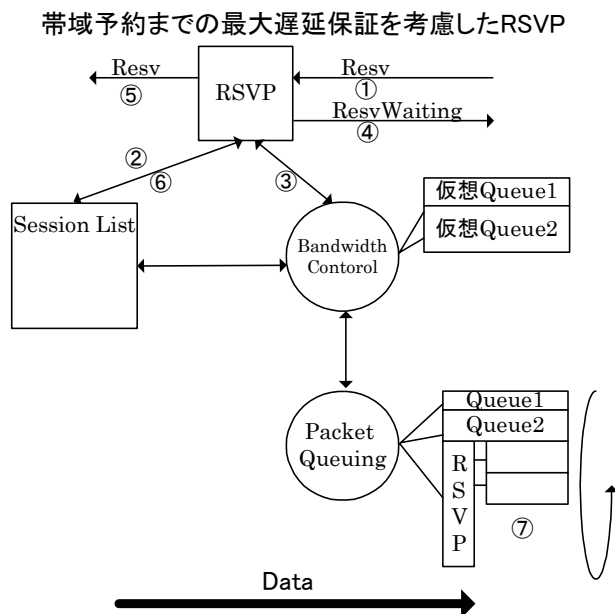


図 6 帯域予約までの最大待ち時間を考慮した RSVP の詳細図

- ①中継ノードは Resv メッセージを受信。要求される帯域が、事前に静的に決定された予約可能帯域以上であれば、ResvErr を送信する。
- ②Resv メッセージを Session List に追加。
- ③帯域予約が要求する帯域が、その時点での予約可能帯域以下であれば、中継ノードは Bandwidth Control に仮想 Queue を作成し、Resv メッセージを次の中継ノードへ送信する。(⑤)
- ④帯域予約が要求する帯域が、その時点での予約可能帯域以上であれば Resv メッセージの送信元に ResvWaiting メッセージを送信し、仮想 Queue 作成のための順番待ちを行い、Resv メッセージを次の中継ノードへ送信。(⑤)

各中継ノードは定期的に帯域予約命令を受信したもので Session List の先頭から Packet Queue の作成を試み、Packet Queue の作成の可否を送信側ノードへ通知する。

全経路の Packet Queue が作成できればデータ送信を開始する。データ送信終了後、Session List から帯域予約情報を、Bandwidth Control から仮想 Queue

を削除する。

以下は中継ノードの定期的な動作となる。

⑥Session List は定期的な定められたアルゴリズムによって、ソートされる。

⑦中継ノードは、一定期間内に帯域予約命令を受信したもので Session List の上位から Packet Queue の作成を行い、送信側ノードへ通知を行う。

このときすでに作成された Packet Queue でデータ転送が行われていないものは一旦削除され、帯域予約を開放し、再度 Session List のソート後、Session List の上位から Packet Queue の作成を行う。

4 関連研究

現在までに、RSVP において帯域を調整する仕組みについて、多くの研究[6][7]がなされているが、これらは現状の帯域を「どのように分配していくか」という点に着目したものであり、予約可能帯域が枯渇すると、従来の RSVP と同様にそれ以降の帯域予約要求に対して、要求を満たすことができない。

5 シミュレーションによる評価

本方式の評価として、NS-2(Network Simulator) 及び、RSVP/ns を用いて本方式をシミュレーションし、既存の待時式帯域予約通信方式及び RSVP と比較する。

シミュレーションの内容は、全ての帯域予約通信が特定のリンクを通過する場合と、全ての帯域予約通信が特定のリンクを通過しない場合のシミュレーションを行う。

5.1 全ての帯域予約通信が特定のリンクを通過する場合

全ての帯域予約通信が特定のリンクを通過する場合のシミュレーションを以下に示す。

シミュレーションには図 7 のネットワークトポロジーを使用した。このネットワークトポロジーは、全ての帯域予約通信が特定のリンクを通過するもので、最も簡潔なネットワークトポロジーであると考えられる。

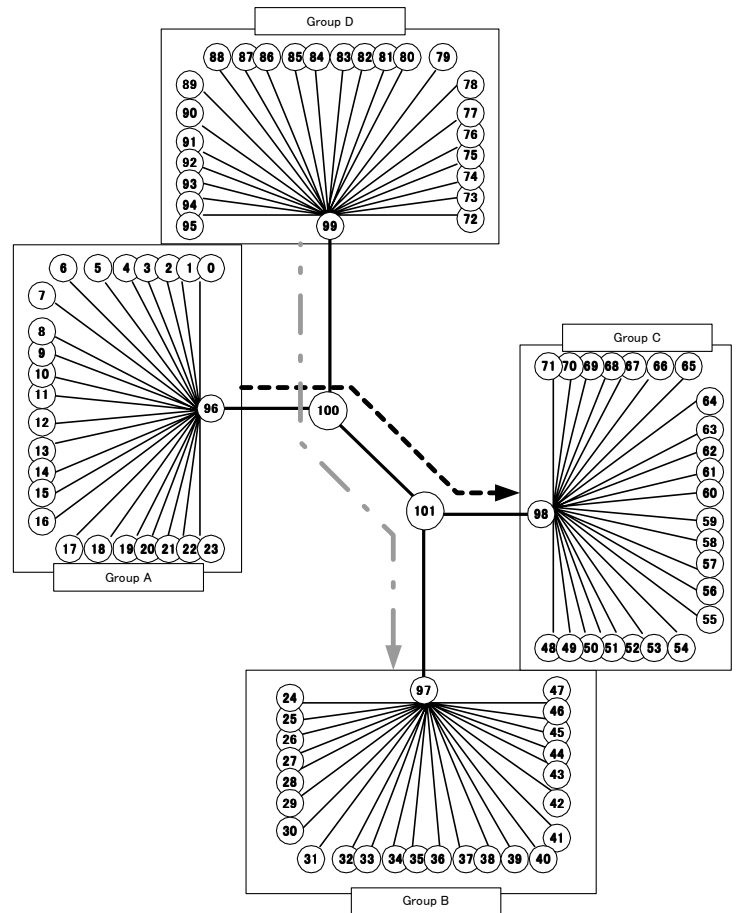


図 7 シミュレーションに使用したネットワークトポロジー-1

図 7 のネットワークトポロジーにおいて、特定のリンクはノード 100 - ノード 101 間である。

本方式、待時式帯域予約通信方式、および RSVP による通信のシミュレーションをそれぞれ 1000 回行った。シミュレーションの条件は以下に示す。

- それぞれのノード間のリンクは、帯域 2Mbps の全二重回線とし、その中での予約可能帯域は 500kbps とした。

これは帯域予約に必要な各種メッセージが使用する帯域、及び、本シミュレーションでは使用していないが、ベストエフォート型通信が使用する帯域を考え、2Mb/s 中、500Kb/s のみを予約可能とした。

- 通信の組み合わせ：

ノード 96 に繋がるノード 0~23 までを 1 つのグループ A とし、そのグループ A からノード 100 - ノード 101 間を通過して、ノード 98 に繋がるノード 48~71 のグループ C に対しての通信を

行った。同様にノード 99 に繋がるノード 72~95 のグループ D から、ノード 97 に繋がるノード 24~47 のグループ B へ通信を行った。予約する帯域は、100kbps、200kbps、300kbps、500kbps をそれぞれ平等に 6 通信ずつ、それぞれのグループ間の通信で行った。

- 本シミュレーションに用いた通信は、UDP フローを用い、パケットの到着過程は CBR を用いた。パケット長はそれぞれ 100 オクテット、200 オクテット、300 オクテット、500 オクテットとし、パケット送信間隔は 0.008 秒とした。帯域予約後、30Mbit のデータ送信を行った。
- それぞれの中継ノードは WFQ を Packet Queuing として用いた。
- それぞれの帯域予約通信は、シミュレーション開始から 20000 秒以内にランダム順に生起させた。
- すべての通信の優先度は同じ優先度とした。
- 予約要求開始から帯域予約までの時間を測定した。
- このシミュレーションにおいて、各リンクに使用した重み値 P は 17 を使用した。
- 本来 RSVP は呼損系であるが、シミュレーションにおいて、データを送信するアプリケーションは帯域予約が成功するまで、帯域予約を繰り返す条件とした。

表 2 に 1000 回のシミュレーションを行った際の帯域予約までの待ち時間の平均を示す。

このシミュレーションにおいては、すべての帯域予約通信は同ホップ数の経路を通過するので、異なる部分は予約する帯域のみとなる。

表 2 シミュレーション結果 1
単位はいずれも $\times 10^3$ Sec

予約帯域	RSVP	待時式帯域予約通信	本方式
100kbps	2.2	6.2	7.7
200kbps	3.0	6.8	8.0
300kbps	5.1	7.1	8.2
500kbps	11.6	7.7	8.6

表 2 の結果より、待時式帯域予約通信方式において、広帯域幅(500kbps)を要求する通信では、RSVP に比べ待ち時間が短くなっていることが確認できる。一方狭帯域を要求する通信では、待ち時間が長くなっているが、これは、先に広帯域を要求する帯域予約が帯域を確保し、通信を終了するまでの間、狭帯域を要求する帯域予約が待つためである。本方式においてはこれらの待ち時間は重み値 P に依存する形になる。今回のシミュレーションにおいては重み値 P を 17 としているが、重み値をより小さくすれば、最大の待ち時間をより短く設定することができる。

この結果より本方式においても、待時式帯域予約通信方式と同様に最大待ち時間を考慮するアルゴリズムが有効に動作しているのが確認できる。

本シミュレーションは、全ての帯域予約通信が特定のリンクを通過するシミュレーションであるので、本方式と待時式帯域予約通信方式において本質的な差はでないはずである。しかしながら若干の差が出ている。これは方式の動作自体が異なるために生じている差だと考えられる。具体的には、方式的に待時式帯域予約通信方式よりも処置が複雑で、メッセージが多い点などが考えられる。

5.2 全ての帯域予約通信が特定のリンクを通過しない場合

次に、全ての帯域予約通信が特定のリンクを通過するのではない場合のシミュレーションを行い、本方式において、待時式帯域予約通信方式の課題が解決されているか否かを以下のシミュレーションにおいて確認する。シミュレーションには図 8 のネットワークトポロジーを使用した。このネットワークトポロジーは、全ての帯域予約通信が特定のリンクを通過しない場合において、最も簡潔なネットワークトポロジーであると考えられる。

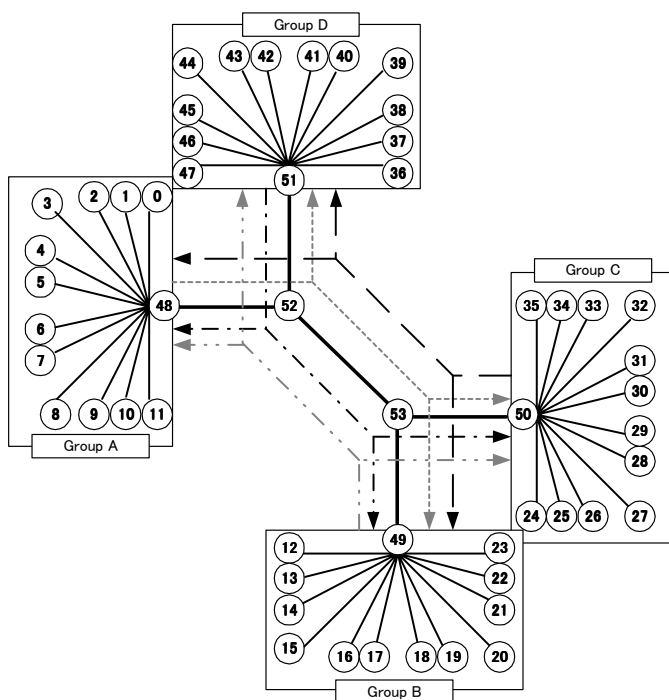


図 8 シミュレーションに使用したネットワークポロジ-2

図 8 のネットワークポロジ-2において、特定のリンクはノード 52 - ノード 53 間である。

本方式、待時式帯域予約通信方式、および RSVP による通信のシミュレーションをそれぞれ 1000 回行った。シミュレーションの条件を以下に示す。

- それぞれのノード間のリンクは、帯域 2Mbps の全二重回線とし、その中での予約可能帯域は 500kbps とした。

これは帯域予約に必要な各種メッセージが使用する帯域、及び、本シミュレーションでは使用していないが、ベストエフォート型通信が使用する帯域を考え、2Mbps 中、500Kbps のみを予約可能とした。

- 通信の組み合わせ：

それぞれのグループから、4 通信ずつ他のグループに対して帯域予約通信を行う。予約帯域は、ノード 52 - ノード 53 間を通過する帯域予約通信は 500kbps、通過しないものは 100kbps とする。例えばグループ A からグループ D へは 100kbps で帯域予約通信を行い、グループ A からグループ B、グループ D へは 500kbps の帯域予約通信を行った。
- 本シミュレーションに用いた通信は、UDP フ

ローを用い、パケットの到着過程は CBR を用いた。パケット長はそれぞれ 100 オクテット、500 オクテットとし、パケット送信間隔は 0.008 秒とした。帯域予約後、30Mbit のデータ送信を行った。

- それぞれの中継ノードは WFQ を Packet Queuing として用いた。
- それぞれの帯域予約通信は、シミュレーション開始から 1000 秒以内にランダム順に生起させた。
- すべての通信の優先度は同じとした。
- 予約要求開始から帯域予約までの時間を測定した。
- このシミュレーションにおいて、各リンクに使用した重み値 P は 17 を使用した
- 本来 RSVP は呼損系であるが、シミュレーションにおいて、データを送信するアプリケーションは帯域予約が成功するまで、帯域予約を繰り返す条件とした。

表 3 に、1000 回のシミュレーションを行った際の帯域予約までの平均待ち時間を示す。

データの比較として、特定のリンクであるノード 52 - ノード 53 間を通る予約帯域 500kbps の帯域予約通信と、特定のリンクであるノード 52 - ノード 53 間を通らない、予約帯域 100kbps の帯域予約通信での帯域予約までの待ち時間の平均に着目する。

表 3 シミュレーション結果 2
単位はいずれも $\times 10^3$ Sec

	RSVP	待時式帯域予約通信	本方式
ノード 52-53 間を通過 しない 100kbps	1.8	6.9	0.45
ノード 52-53 間を通過 500kbps	6.5	9	6.1

表 3 の結果より、待時式帯域予約通信方式において、隣のグループに対する 100kbps の帯域予約通信において、帯域予約までの待ち時間が非常に長くなっていることがわかる。一方、本方式においては、隣のグループに対する帯域予約までの平均待ち時間が大幅に短縮されており、このとき、待時式帯域予約通信方式に見られる、無用な待ちが解消されていることが確認で

きる。

6 結論と今後の課題

本方式を使用することにより、RSVP では実現できなかった、すべての帯域予約通信に対する帯域予約を、待時式によって、実現することが可能になる。

これは、待時式帯域予約通信方式においても可能であったが、待時式帯域予約通信方式では、広域ネットワークなどの全ての帯域予約通信が特定の経路間を使用しない場合を考慮した際に、無用な待ちも大きいという問題があった。これに対し本方式を使用すれば、この問題を解決し、無用な待ちを大幅に削減することが可能となり、かつ、帯域予約までの最大待ち時間をコントロールすることができる。

しかしながら、全ての帯域予約通信が特定の経路間を使用する場合、例えば企業間ネットワークなどで、遠隔地に離れた企業内 LAN を VPN などで接続し、使用する場合には待時式帯域予約通信方式の方がより短い待ち時間となる結果となっている。

現時点では、それぞれの中継ノードにおける計算量等をシミュレーションの待ち時間には反映できておらず、その点からも、例えば大多数の帯域予約通信が特定の経路間を使用する場合には待時式帯域予約通信を使用し、大多数の帯域予約通信が特定経路間を使用しない場合には本方式を使用するというように、ネットワークの形態によって使い分けが必要となると考えられる。

また、帯域予約までの待ち時間を決定するのはソートアルゴリズムとなるが、現状では単一のアルゴリズムによるソートであり、重み値自体も、手動で算出した値であり、個々の中継ノードにおいて、独立したソートを行っている。このソートアルゴリズムにおいて、複数のアルゴリズムを組み合わせたり、重み値を自律的な決定を行ったり、個々の中継ノードを連携させていくことにより、より帯域予約までの最大待ち時間とそのパフォーマンスのバランスの良いネットワークを実現することが可能になると考えられる。

またこの問題は、ネットワーク全体の最適化問題と考えることも出来る。各ノードで帯域の割り当てを分散処理するのではなく、COPS(Common Open Policy

Service)[11]を用いた集中制御も検討し、今後の課題としたい。

参考文献

- [1] L. Delgrossi, L. Berger, "Internet Stream Protocol Version 2 (ST2) Protocol Specification -Version ST2+", RFC 1819, August 1995.
- [2] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin. "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.
- [3] 池邊,本多,弓場,三木 "IP ネットワークにおける待時式帯域予約通信方式の評価", 信学技報,IN2001-1(2001-05) pp1-6,2001
- [4] <http://www-mash.cs.berkeley.edu/ns/>
- [5] Marc Greis, RSVP/ns: An Implementation of RSVP for the Network Simulator ns-2
- [6] 荒川,渥美,"RSVP を利用した適応的な帯域制御方式の検討",信学技報, IN98-108, pp37-44,1998.
- [7] 勝又ほか, "通信品質保証のための帯域管理機能の提案", 信学技報,IN97-168(1998-02).
- [8] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, September 1997.
- [9] S. Shenker, J. Wroclawski, "Network Element Service Specification Template", RFC 2216, September 1997.
- [10] S. Shenker, C. Partridge, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.
- [11] D. Durham, J. Boyle "The COPS (Common Open Policy Service) Protocol", RFC 2478, January 2000.