TCPプロトコルにおけるマルチキャスト機能の実現法

高村 尚彦†　大西 淑雅‡　Bernady O. Apduhan†
末吉 敏則†　有田 五次郎†
(†九州工業大学 情報工学部)
(‡九州工業大学 情報科学センター)

概要

分散処理環境における計算機資源の効率的利用を目指し，現在事実上の標準プロトコルとなっているTCP/IPを対象としたマルチキャスト機能の実現法について述べている．まず，マルチキャストが協調分散アプリケーションに与える効果を示し，次にTCPにおけるマルチキャスト機能の実現法について述べる．

1　はじめに

分散処理環境の発達にともない，ネットワーク通信を用いたアプリケーションが多く開発されており，その例としてテレビ会議システムや協調分散アプリケーションが挙げられる．このようなアプリケーションにおいて大量のデータを複数のホストに送信する場合，マルチキャストは処理効率の改善に有効な方法となる．

マルチキャストの利点には，(1) パケット数の減少によるネットワークトラフィックの減少，(2) プロトコル処理オーバヘッドの削減，(3) 協調分散アプリケーションにおいて各プロセッサ間の同期を取る際のタイムラグの解消などがある．

マルチキャスト機能を持つトランスポート層プロトコルとしてVMTP[4]，XTP[5] 等が代表的である．VMTPは下位層のプロトコルとしてマルチキャスト拡張IP[2] を想定しており，その上位プロトコルとして高い信頼性を保証する．これらのプロトコルに対して，TCPを用いてマルチキャストを行う利点は，事実上の標準プロトコルを用いることにより従来と同様のプログラム記述でマルチキャストを実現でき，プログラムの移植性も高くなる点である．

本稿では，TCPにおけるマルチキャスト機能実現によって協調分散アプリケーションへ与える効果を調査し，マルチキャストの実現手法を提案している．下位層プロトコルとしてマルチキャスト拡張IPを用い，従来と同様の信頼性を提供するために順序制御，誤り検出，再送制御，フロー制御等を行い，更にマルチキャストの宛先となるグループに関する情報を管理する．グループを構成する各ホストでは，マルチキャストパケットを正しく受信すると，それを送信元に知らせるために送達確認を返信する．

以下，2章ではマルチキャストがアプリケーションに与える効果の検証を行い，3章にマルチキャスト機能拡張TCPの設計方針を述べ，4章にパケットフォーマットや各種制御に関する機能仕様を記述している．

2　協調分散処理におけるマルチキャストの効果

本章では，協調分散処理にネットワーク通信が及ぼす影響を，マルチキャストを行う場合と行わない場合について比較し，マルチキャストの効果を示す．

2.1　ユニキャスト対マルチキャスト

まず，複数のホスト（N台）に対し同じ内容のデータを送信する際の，ユニキャストのみの通信とマルチキャストをした場合との比較を行う．

ユニキャストのみの場合，宛先となるホスト各々に計N個のパケットを送信する．TCPのように送信パケットに対し送達確認（ACK）を返す場合には，

An Approach to Realize Multicast Facility in TCP Protocol, by Naohiko Takamura, Yoshimasa Ohnishi, Bernady O. Apduhan, Toshinori Sueyoshi and Itsujiro Arita (Kyushu Institute of Technology)

各々からACKパケットが1個ずつ返信されるため，計2N個のパケットが発行されたことになる[1]．

マルチキャストを用いた場合，宛先グループには共通のグループIDが与えられ，そのグループIDを持つパケットが1個送信される．それに対するACKパケットを考慮に入れると，計(N+1)個のパケットが発行される．従って，ユニキャストのみの場合に対するパケット数の減少率は$\frac{N+1}{2N}$となり，ホスト数の増加に伴いパケット数は約半分に減少する．

2.2 アプリケーション全体から見た効果

次に，アプリケーションが送出する全メッセージに占める，マルチキャスト置換え可能なメッセージの比率とパケット数との関連について解析を行った．以下の様に変数を設定する．

N_s	：ユニキャスト時の送信パケット数．
N_{ack}	：受信するACKパケット数．$(=N_s)$
N_{total}	：ユニキャスト時の総パケット数．$(=2N_s)$

N_s個のうちR%のパケットが，構成ホスト数hのグループへのマルチキャストに置き換えられるとき，パケット数は次のようになる．$(r=R/100)$

ユニキャストされるパケット数：

$$N_u' = (1-r)N_s$$

マルチキャストパケット数：

$$N_h' = \frac{N_s}{h}r$$

従って，送信されるパケット数は，

$$N_s' = N_u' + N_h' = (1-\frac{h-1}{h}r)N_s$$

となる．ACKパケット数はN_{ack}と等しい．以上より総パケット数は，

$$N_t' = N_s' + N_{ack} = (2-\frac{h-1}{h}r)N_s$$

となり，ユニキャスト時に対してパケット数は$(\frac{h-1}{h}r)N_s$(個)減少し，パケット数減少率は$(1-\frac{h-1}{2h}r)$となる（図1）．

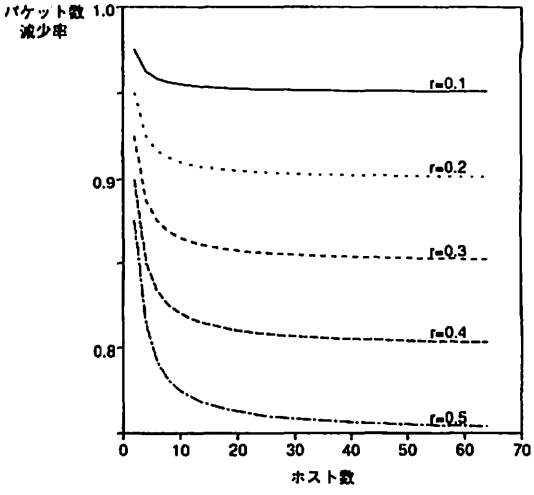

図1．アプリケーションにおけるパケット数減少率

2.3 分散処理システムにおけるマルチキャストの効果

実際のアプリケーションにマルチキャストが与える効果を，DSE(Distributed Supercomputing Environment)[6]上で動作するアプリケーションを対象に検証する．DSEは，図2(a)に示す分散共有メモリ型並列計算機と等価の機能を，図2(b)のようなネットワーク接続された複数のワークステーション間でのメッセージ交換により実現している．DSEアプリケーションは信頼性の高い通信を要求するため，通信プロトコルとしてTCPを用いる．今回の検証はオセロゲームを8台のワークステーションを用いて処理する場合について行っている．

アプリケーションが交換したメッセージの内訳を表1に示す．マルチキャストを適用可能なメッセージは，複数のプロセッサに対し同一内容のデータを送信するものである．アプリケーションが規定したメッセージフォーマットを変更せずにマルチキャストの適用が可能なメッセージは，プロセス起動要求，LWP(LightWeight Process)キューの初期化，LWP起動要求，資源開放要求の4種類であり，これらのメッセージが全体に占める比率は3.17%である．従っ

[1]説明の簡単化のため，ACKはパケットを1個受信する度に返されると仮定している．

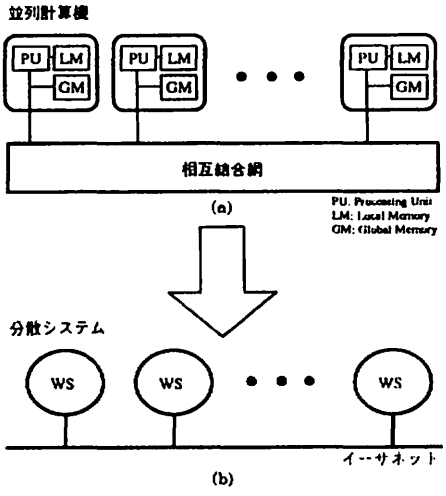

図 2. 協調分散処理システムのモデル

て，パケット数減少率は97%となり，約850個のパケットの減少となる．

また，後述するサブグループへのマルチキャストとメッセージフォーマットの変更を行うことにより，全体の31.1%を占めるFetch&Addへの応答メッセージもマルチキャスト可能となり，この場合にはパケット数は大幅に減少する．

これを処理時間の点から見ると，処理完了までには約54.7秒かかっている．同一データが他の7台のプロセッサに配送されるまでに平均7msec.，最大195msec.かかることが分かっており，マルチキャストにより，この配送の時間差を累積した約1.6秒は確実に処理時間が短縮される．更に，その間のプロセッサ稼働率の上昇のために，全体としての処理時間は更に短くなることが予想される．

3 設計方針

本章では，マルチキャスト機能を持つTCP（以後マルチキャストTCPと呼ぶ）の設計方針について述べる．マルチキャストTCPの特徴は以下のようなものである．

- マルチキャスト拡張IPの上位層プロトコルとして，信頼性の高いマルチキャストを支援．

表1. DSEアプリケーション（オセロゲーム）に使われたメッセージの内訳

メッセージの種類	個数	マルチキャストの適用	全体に占める割合(%)
プロセス起動要求	8	○	0.03
LWPキューの初期化	8	○	0.03
LWP起動要求	877	○	3.08
資源開放要求	8	○	0.03
Fetch&Add	8856	×	
〃 応答	8856	△	(31.1)
共有メモリ読み出し等	8364	×	
共有メモリ書き込み等	246	×	
セマフォ関連	1230	×	
合計	28453		3.17

LWP: LightWeight Process

- TCPの規格に準拠し，従来とほぼ同様の手続きでマルチキャスト機能を利用可能．

以下，グループの定義，インタフェース，他のプロトコルとの関係，信頼性の高い通信の実現，コネクションの開設と閉鎖，データ通信について，その設計方針を述べる．

3.1 グループの定義

マルチキャストは，共通のグループIPアドレスを持つホスト群（グループ）へセグメントを送信することである．クラスDのIPアドレス[2]がグループIPアドレスとして使用される．

1グループのメンバー数の上限のデフォルト値は20となっており（[2]参照），また，1ホストが複数のグループの構成ホストとなることも可能である．グループの構成例を図3に示す．グループの構成はコネクション開設時に確定している必要があり，そのグループに関するコネクションが全て閉じられるまで構成の変更は行えない．ここで以降の説明に使用する用語を定義しておく．

[2] 111で始まるIPアドレス．十進区分で表すと，224.0.0.0～239.255.255.255．

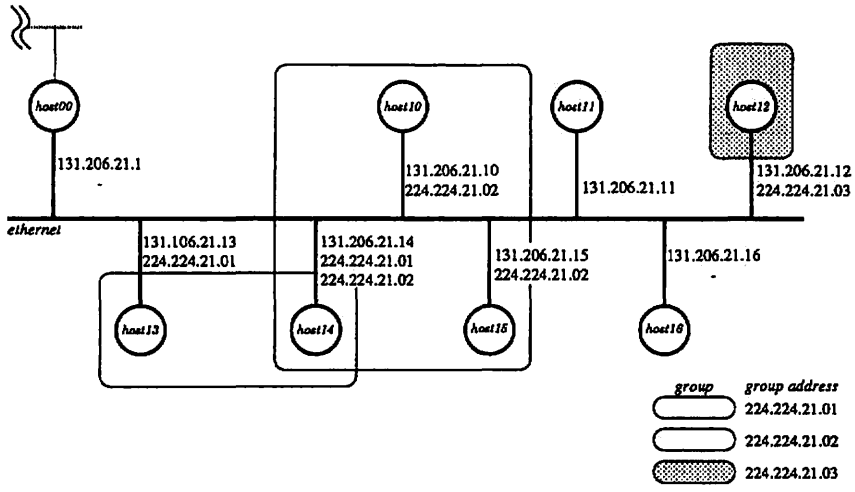

図 3. グループ構成例

グループ	共通のグループ IP アドレスを持つホスト群.
グループ構成ホスト	グループのメンバーとなっているホスト.
データ送信ホスト	グループにデータを送信するホスト.

3.2 インタフェース

マルチキャスト TCP とユーザ間のインタフェースは，コネクションの開設 (OPEN)，コネクションの閉鎖 (CLOSE)，データ送信 (SEND)，データ受信 (RECEIVE)，コネクション状態の獲得 (STATUS) といった操作に加え，

- グループへの参入 (JOIN)
- グループからの脱退 (LEAVE)

といったものを提供する．JOIN, LEAVE 操作はグループ構成に関する操作であり，OPEN 操作に先だって行われ，コネクションを既に確立したグループに対して JOIN 操作や LEAVE 操作を行うことはできない．

マルチキャスト TCP とネットワーク側のインタフェースは，ネットワークシステム内の任意のマルチキャスト TCP モジュールを指定したデータグラムの送受信を行う手続きを供給する．

3.3 信頼性の高い通信

TCP コネクション上では，データの送信順序と信頼性が保証され，そのために順序番号と送達確認 (ACK) が使われる．マルチキャスト TCP においてデータを含むセグメントがグループに送信された場合，そのデータは再送キューにコピーされ，再送タイマがセットされる．再送タイマが切れるまでに全てのグループ構成ホストから ACK を受信すれば，該当データは再送キューから取り除かれる．全ての ACK を受信する前に再送タイマが切れた場合，グループに対してデータの再送が行われる．

フロー制御のために，グループ構成ホストは各々のウィンドウ値をデータ送信ホストに知らせる．ウィンドウは各グループ構成ホストが受信可能なオクテット数であり，送信ホストでは各グループ構成ホストのウィンドウを比較し，その最小値に従って送信を行う．

3.4 コネクションの開設と閉鎖

TCP が扱うデータストリームを，ホスト上の各プロセスに分配するためにポート識別子が使われる．ポート識別子は IP アドレスと組み合わされ，ネットワーク全体で一意な識別子となる．グループ構成ホストは共通のポート識別子を用いてコネクションの開設を行い，グループ IP アドレスとポート番号によりグループの識別は一意になされる．

コネクションに関する情報はTCP制御ブロックに蓄えられる．データ送信ホストにおいて個々のグループ構成ホストに関する情報を保持する必要があるため，従来のTCP制御ブロックを拡張した（図4）．コネクションの開設には3-way handshakeと

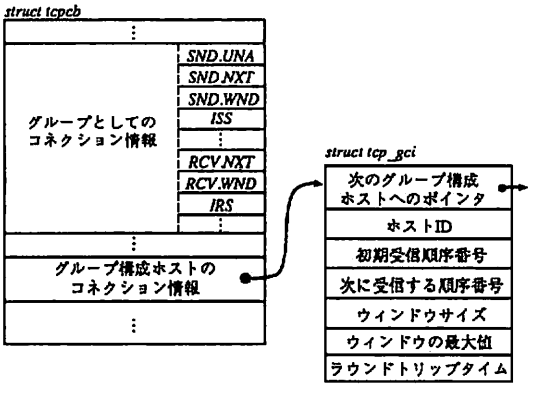

図4. TCP制御ブロックの拡張

呼ばれる方法が使われ，順序番号の同期や，その他の制御パラメータの交換を行う．SYNフラグを含むセグメントにより，双方向の順序番号の同期が完了すればコネクションは確立する．また，コネクションの閉鎖はFINフラグを含むセグメントを交換して行われる．

4 機能仕様

4.1 ヘッダフォーマット

マルチキャストTCPのヘッダは，従来のTCPのヘッダの一部を変更したものである（図5）．

変更されたフィールドについて以下に述べていく．

順序番号フィールド

グループ構成ホストから見た送信順序番号（データ送信ホストからは受信順序番号）フィールドには，そのコネクション開設時の初期送信順序番号（又は初期受信順序番号）からのオフセット値が書き込まれる．このオフセット値と，個々のグループ構成ホストの初期順序番号を加算し，送信順序番号（受信順序番号）の実効値を得る．グループ構成ホストからのデータ送信は行われないため，オフセット値が変更されるのは，コネクション開設時のSYNパケット送信時と，コネクション閉鎖時のFINパケット送信時に限られ，その度にオフセット値は1ずつ加算される．

制御ビットフィールド

マルチキャストTCPでは，TCPで定められた6種類の制御ビット（URG, ACK, PSH, RST, SYN, FIN）に加え，マルチキャストパケットであることを示すMLT(MuLTicast)フラグが設定される．MLTフラグとACKフラグが同時にセットされた場合，マルチキャストに対するグループ構成ホストからのACKパケットであることを示し，このとき送信データは含まれない．

また，MLTビットの追加により予約領域は5 bitsになる．

オプションフィールド

サブグループへのマルチキャストのためにオプションフィールドを用いるため，オプションのリストは表2のようになる．

表2. マルチキャストTCPのオプションリスト

種別	長さ	意味
0	-	オプションリストの終り
1	-	無操作
2	4	最大セグメント長
3	4	サブグループマスク

サブグループマスク

サブグループへのマルチキャストを行った場合，コネクションを張っている相手グループ内の指定されたホストではデータが上位層に渡され，指定外のホストでは順序番号の更新とACKの送信を行い，データは上位層に渡されない．このオプションを用いる利点は，既にコネクションを張っているグループに含まれるホスト群（サブグループ）と，新たにコネクションを張ることなく通信を行える点である．

サブグループの指定に使われるのがサブグループマスクである．各グループ構成ホストには，グループ内で一意に決定されるホストIDがアプリケーションから割り当てられる．マルチキャスト時に，アプ

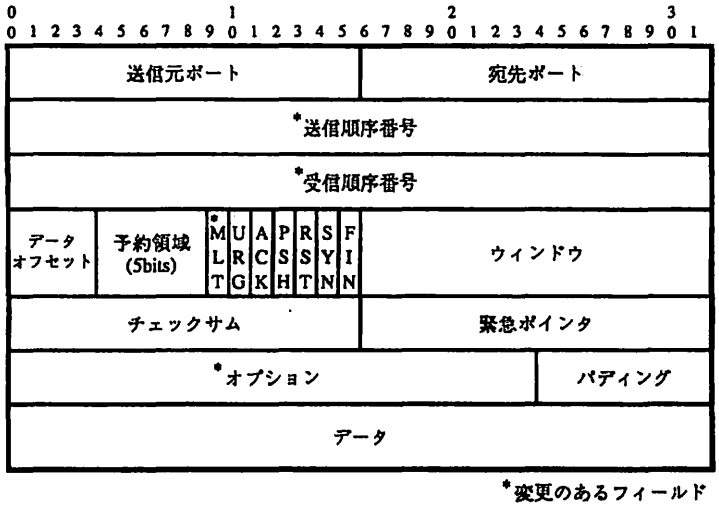

図 5. ヘッダフォーマット

リケーションがサブグループマスクをセットした場合，受信ホストでは，自ホストに割り当てられたホスト ID とオプションフィールドのサブグループマスクとの論理演算を行い，そのセグメントの処理を判断する（図 6）．尚，図中では説明の便宜のためにサブグループマスクは 8bits にしているが，実際のマスクは 16bits である．

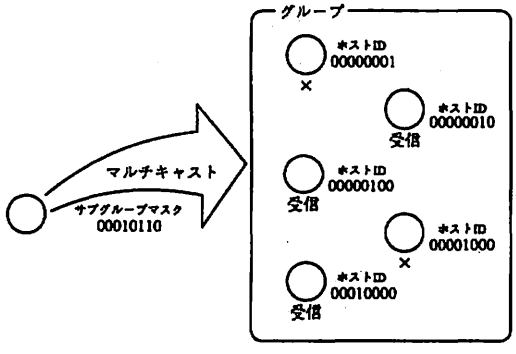

図 6. サブグループへのマルチキャスト

4.2 順序制御

TCP 制御ブロックにはコネクション維持のために必要な幾つかの変数を保持している．TCP 制御ブロックが保持する変数は，ローカル及びリモートのソケット番号，コネクションの優先度とセキュリティ変数，ユーザ側の送信バッファ及び受信バッファへのポインタ，再送キュー及び現在のセグメントへのポインタ，送信順序番号及び受信順序番号へのポインタといった変数である．マルチキャスト TCP では以上のものに加え，グループを構成する各々のホストに関してコネクション情報を保持する．以下に，説明に用いる用語についてまとめておく．

データ送信ホストの TCP 制御ブロック

SND.UNA	送信済みだが ACK 未受信の領域の先頭．$SND.UNA_{grp}$ はグループ全体としての値を示し，$SND.UNA_{host[i]}$ はグループ構成ホスト i の値を示す．以下の変数についても同様である．
SND.NXT	次に送信するデータの先頭．
SND.MAX	送信済みデータの送信順序番号の最大値．
SND.WND	送信ウィンドウサイズ．
ISS	送信順序番号の初期値．
RCV.NXT	次に受信するセグメントの順序番号の期待値．

RCV.WND　受信ウィンドウサイズ.

IRS　受信順序番号の初期値. 各グループ構成ホストで異なる値を保有し, グループとしての値はグループ構成ホストのものに依存しない.

更に, データ送信ホストで処理中のセグメントに対し以下の変数を用いる.

SEG.SEQ　セグメントの送信順序番号.

SEG.ACK　セグメントの受信順序番号. この場合, 初期値からのオフセット.

SEG.LEN　セグメント長.

SEG.WND　セグメントのウィンドウ値.

順序番号

TCP コネクション上で送信されるデータにはオクテット毎に順序番号が付けられる. ACK はパケット毎に送られるのではなく累積して送られ, セグメントの受信順序番号が X であれば送信順序番号 (X-1) までのデータを全て受信したことを意味する.

TCP では順序番号を比較して以下の事項を調べる.

- ACK 未受信のセグメントへの ACK であるか.
- 全送信セグメントに対して ACK を受信したか.
- 受信セグメントが, 期待している順序番号を含んでいるか.

ACK 受信時には, まず対応するグループ構成ホストに関するコネクション情報を更新し, その後グループ全体のコネクション情報との比較を行い, 必要ならば更新を行う. 各々の変数の更新は以下のように行う.

SND.UNA：

$$SND.UNA_{grp} = min(SND.UNA_{host[1]}, \cdots, SND.UNA_{host[n]})$$

SND.NXT：次に送信するセグメントの順序番号.
SEG.ACK：SND.UNA≤SEG.ACK≤SND.MAX ならば受理.
SEG.LEN：セグメントに含まれるデータ長. SYN, FIN を含むセグメントの場合 1.

また, 受信領域の変数に関して以下の処理を行う.

$$RCV.NXT_{grp} = min(RCV.NXT_{host[1]}, \cdots, RCV.NXT_{host[n]})$$

$$RCV.WND_{grp} = min(RCV.WND_{host[1]}, \cdots, RCV.WND_{host[n]})$$

初期順序番号の決定

順序番号の同期には 3-way handshake が用いられる. マルチキャストを行う場合, 複数のグループ構成ホストと同期を取るため, 各ステップにおいて全グループ構成ホストからの ACK を受信しなければ, 次のステップに進むことはできない.

データ送信ホストの ISS (グループ構成ホストからは IRS に相当) の決定は従来と同様の方法で行う. 一方, グループ構成ホストの ISS (送信ホストからは $IRS_{host[i]}$) 決定は, 各グループ構成ホストで別々に行われ, SYN セグメントに対する ACK(+ SYN) セグメント送信時に SEG.SEQ として ISS を返す. 以後の通信においてグループ側が用いる SEG.SEQ は, ISS_{grp}からのオフセット値が使われ, 送信順序番号の実効値は, データ送信ホストのコネクション情報に含まれる該当ホストの $IRS_{host[i]}$ にオフセット (SEG.SEQ) を加算して求められる.

4.3 データ通信

コネクションが確立されると, データ送信ホストからグループに向けてデータが送信される. グループからデータ送信ホストへのデータ送信は, 順序番号の一貫性を保持するために行わない. セグメントはエラー (チェックサムの失敗) やネットワークの輻輳のために破棄される可能性があり, 全セグメントの確実な配送を保証するために再送が行われる. 再送によりセグメントが重複して到着するかもしれず, セグメントの順序番号の確認により重複は検出される.

データ送信ホストは, セグメント送信の際に SND.NXT を進める. 同様に, グループ構成ホストがセグメントを受信したら RCV.NXT を進める. 送信ホストがグループ構成ホストの一つから ACK を受信したら $SND.UNA_{host[i]}$を進め, それにより $SND.UNA_{grp}$が更新される場合もある.

セグメントの再送

再送タイムアウト値の決定にはRTT(Round Trip Time)値が使われるが，グループが複数ホストから構成される場合，当然各ホストによりRTT値にばらつきがある．従ってRTTの測定は，グループ構成ホスト個々について行われ，その最大値を対象グループに対するRTT値として再送タイムアウト値の計算を行う．

セグメント送信後，それに対してACKを受け取る前に再送タイマが切れると，セグメントの再送を行う．再送はグループに対してgo-back N方式で行われ，既にセグメントを正しく受信したグループ構成ホストでは，新たに受信したセグメントの順序番号を確認することにより重複セグメントを検出し，破棄する．

ウィンドウの管理

SEG.WNDには，そのSEG.ACKから始まる受信領域の大きさがオクテット単位で示される．グループ構成ホストから送られるACKセグメントにはそれぞれ異なるウィンドウ値が示されており，データ送信ホストでは，各グループ構成ホストのウィンドウ値と受信順序番号値から最小のウィンドウ値を算出し，それを$SND.WND_{grp}$とする．

また，SEG.WNDが0であるセグメントを受信した場合，データ送信ホストからデータを含むセグメントの送信は一定時間行えない．但し，再送セグメントやACKセグメントの送信は除外する．

5 おわりに

本論文では，TCP/IPを用いたマルチキャストの実現法について述べた．TCPはコネクション指向であるため，マルチキャストを実現するためには，従来のTCPで管理していた以上のコネクション情報が必要となる．特に，グループを構成する個々のホストに関するコネクション情報量はグループ構成ホストの増加に比例するため，コネクション情報をより効率的に管理しなければ，プロトコル処理速度の大きな低下を招いてしまう．

また，アプリケーションが行う通信全体に占めるマルチキャスト可能部の比率が大きく，マルチキャストの宛先となるグループを構成するホスト数が多いほど，マルチキャスト機能を実装時の効果は大きくなり，この点と処理オーバヘッドの増加との兼ね合いが大きな意味を持つ．

TCP/IPを用いてマルチキャストを行うことにより，多くの既存のアプリケーションの一部を書き換えるだけで容易にマルチキャストを実現でき，ソフトウェア資源の有効利用の点からも，他の通信プロトコルを用いるのに比べ有利となる．

今後，マルチキャストTCPの実装，及び性能評価を行い，その結果に基づいてより効率的にマルチキャストを実現できるような技法を検討していくことが課題である．

参考文献

[1] J. Postel, *Transmission Control Protocol*, RFC-793, SRI Network Information Center, Menlo Park, CA, September 1981.

[2] S. E. Deering, *Host Extensions for IP Multicasting*, RFC-1054, SRI Network Information Center, Menlo Park, CA, May 1988.

[3] S. J. Leffler, M. K. McKusick, M. J. Karels, and J. S. Quarterman, *The Design and Implementation of the 4.3BSD UNIX Operating System*, Addison-Wesley Publishing Company, Inc., 1989.

[4] D. Cheriton, *Versatile Message Transaction Protocol(VMTP)*, RFC-1045, SRI Network Information Center, Menlo Park, CA, February 1988.

[5] G. Chesson, *XTP/PE Overview*, Proceedings in 13th Conference of Local Area Networks, pp. 292-296, 1988.

[6] T. Tezuka, K. Ryokai, B. O. Apduhan and T. Sueyoshi, *Implementation and Evaluation of a Distributed Supercomputing Environment on a Cluster of Workstations*, Proceedings in 1992 International Conference on Parallel and Distributed Systems, pp.58-65, December 1992.