

高信頼性マルチキャストプロトコルの開発と実装

市村 重博 坂田一拓 倉島 顕尚 前野 和俊

NEC C&C メディア研究所

筆者らは高信頼性マルチキャストプロトコル (SMFTP) を研究開発している。このプロトコルは無線、有線が混合したネットワーク上でのグループ内の資料の交換やプッシュサービスに使用することを目的としており、バルクデータを効率的に送受信できる事の特徴としている。そのために、SMFTP はフローコントロールの機能を実装している。フローコントロールは、送信ホストが実際に送信したサイズを受信ホストに通知し、受信ホストが実際に受信したサイズと比較して受信状態を判断し、受信状態が悪い場合には送信ホストに通知し、送信ホストはその情報に基づいて送信量を増減する事によって行う。また、許容未受信率を設定する事により品質の悪いネットワークにも対応していることを特徴とする。

The Development and Implementation of a Reliable Multicast Protocol

Shigehiro Ichimura, Kazuhiro Sakata, Akihisa Kurashima,

Kazutoshi Maeno

C&C Media Research Laboratories, NEC

The authors are in the process of developing a highly reliable multicast protocol, with the aim of achieving efficient bulk data transfer, such as document exchanges within a group and push services, on wired and wireless networks. The protocol features a flow-control mechanism for improving efficiency in communication over networks. Flow-control is carried out by receiver notifications to the sender, and speed adjustments made by the sender on the basis of the notifications.

1 はじめに

インターネットの発展と携帯端末の普及によってグループ協同作業やプッシュ型の情報サービスが注目されている。これらのサービスは一つの情報を多数の受信ホストに対して送信する。例えば、グループ協同作業の中では参加者に対して資料の配布を行い、またプッシュ型情報サービスではコンテンツを受信ホストに配信するという作業を頻繁に行う。

同じ情報を多数の受信ホストに送信する方法として IP マルチキャストがある [1]。IP マルチキャストは、送信ホストが送信した情報を、あるマルチキャストグループに属しているすべてのホストに配信する。マルチキャストグループ

への参加退席は IGMP [2] によって、マルチキャストグループ参加ホストへのデータの配信経路は DVMRP [3], PIM-SM [4], CBT [5] などのマルチキャストルーティングプロトコルによって扱われる。IP マルチキャストでは送信された情報が途中で失われた場合の補償や、受信されたかどうかの確認はできない。

情報配信サービスでは、データの送信において、信頼性の高い方法が求められる。そこで情報配信を IP マルチキャストを使用して効率的に行った上で、欠落した情報の再送などにより、より高い信頼性をもたらすプロトコルが研究されてきた。これらのプロトコルを総称して高信頼性マルチキャストプロトコル (Reliable Multicast Protocol) と呼ぶ。これまでに RMTP [6], MFTP [7], PGM [8] などが発表

この論文の中に出て来る商標名は各社の登録商標です。

された。

筆者らは高信頼性マルチキャストプロトコル SMFTP を開発・実装してきた。本論文ではこの SMFTP の特徴と機能について述べる。第 2 章では高信頼性マルチキャストプロトコルにおける一般的な課題について述べる。第 3 章では我々が開発した SMFTP について説明する。第 4 章では、SMFTP の大きな特徴であるフローコントロールについて詳細に説明する。

2 高信頼性マルチキャストプロトコルにおける課題

高信頼性マルチキャストプロトコルにおける課題は、信頼性確保をどのような方法で行うかということ、フローコントロール及び輻輳回避の方法である。

2.1 信頼性の確保

信頼性確保には、再送を行う方法と FEC(Forward Error Correction) を使う方法がある。

再送によって行う方法 送信されたパケットが損失して受信できなかったとき、再送によって補う。受信ホストが受信できなかった情報の再送要求を行うときに、これを全部送信ホストに送ると送信ホストには多数のパケットが殺到することになるので(図 1)、それを如何に回避するかが課題である。主な解決策として、受信ホストをいくつかのグループに分けてその中で再送を行う方法や、グループ分けされた受信ホストから再送要求を受け付けてまとめて送信ホストに送る方法 [6]、再送要求を送信するタイミングをずらして、他の受信ホストからの再送要求を受信した受信ホストは再送要求を控える方法などがある。

FEC によって行う方法 あらかじめ送信するパケットに冗長性を与えておき、損失したパケットを他のパケットから復元することによって、信頼性を確保する [9]。適切な冗長度を与えておけばジッターが小さくなるので、音声、映像系の配信に向いている。パケット損失のモデルをどう設定し、冗長度をどのように与えるかが課題である。

2.2 フローコントロール

マルチキャストにおいては受信ホストは多数でかつ様々であり、受信ホストまでのネット

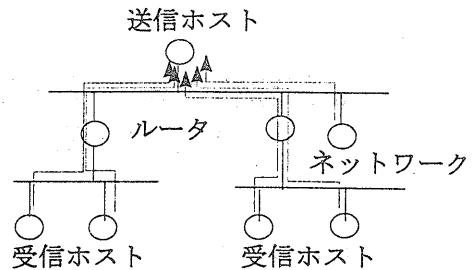


図 1: 再送要求の集中

ワークに流れる他のトラフィックも存在するので、送信ホストのみの都合で送信量を決定して送信した場合、受信ホストの処理能力不足やネットワークの混雑のためにこれらの情報の一部が配送・受信されず途中で欠落することがある。再送を行って信頼性を確保する高信頼性マルチキャストプロトコルにおいては、これらの欠落した情報は再送されることになり、送信ホストや受信ホスト、ネットワークに対してより大きな負荷をかけることになる。これを防ぐためにネットワークに流れるパケットの流量をどのように調整するかが課題である。

送信ホストが送信量を制御する方法、受信ホストの退席によって受信ホストまでのパケットの配送に影響を与え、それによってコントロールする方法 [10] などが提案されている。

しかしながらこれまでに提案されたプロトコルは、グループ内での資料の交換や情報配信サービスにおけるバルクデータ転送を行うときに必要な、能力の大幅に異なる受信端末の存在や、受信端末の増減、品質の悪いネットワークの存在について考慮されていなかった。

3 SMFTP

3.1 特徴

SMFTP はグループ内での資料の配布やプッシュ型の情報サービスに対して効率的なバルクデータ転送の手段を提供することを目的としている。そのために以下の特徴を持っている。再送によって信頼性を確保 信頼性の確保は欠落情報の再送によって行っている。また再送は、初めの送信が全部終了した後に行っている。これは SMFTP はバルクデータ転送のような送受信に時間のかかる場合を想定しているため、送信中に新たな利用者が受信を開始する事が十分に考えられ、この場合、途中で何回か再送する

種類	概要	送受信方向
PRO_INIT	イニシャライズ	送 → 受
PRO_INIT_ACK	イニシャライズ確認	送 ← 受
PRO_DATA	データ送信	送 → 受
PRO_DATA_END	データ受信結果要求	送 → 受
PRO_DATA_ACK	データ受信確認	送 ← 受
PRO_DATA_NAK	再送要求	送 ← 受
PRO_FLOW_REQ	受信状態要求	送 → 受
PRO_FLOW_RES	受信状態通知	送 ← 受

図 2: SMFTP の PDU

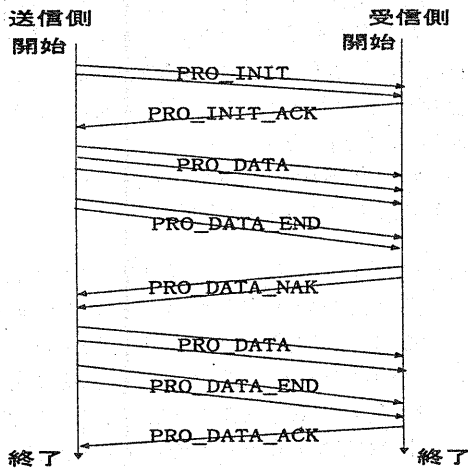


図 3: SMFTP での送受信のシーケンス

よりも最後に再送を行った方が再送数が少なくなりネットワークの利用効率が良くなるからである。

送信ホストが再送 再送の要求は全て送信ホストに送られ、送信ホストのみが再送を行う。SMFTP では動的に受信ホストが変化する状況を想定しており、グループ化して再送要求を集積するなどの方法は適当でないので、送信ホストのみが再送する方式とした。

フローコントロール フローコントロールの機能を実装している。許容未受信率を設定する事により品質の悪いネットワークにも対応している。後に詳説する。

3.2 PDU とシーケンスの概要

図 2 に SMFTP の PDU の種類を示す。図 3 にこれらの PDU を使用した送受信のシーケンスを示す。なお PRO_FLOW_REQ と PRO_FLOW_RES は後に詳しく説明するために省いてある。次章で SMFTP におけるフローコントロールについて説明する。

4 フローコントロール

4.1 課題

高信頼性マルチキャストにおいて高速な送受信を行うには、送信したパケットの損失を防ぎ、再送を減らさなければいけない。送信したパケットが損失する原因は以下の3つである。

- (1) 受信ホストの受信能力不足
受信ホストの受信能力が充分でないと配信されてくる情報を全て受信できずに、パケットの損失が起きる。
- (2) ネットワークの輻輳
受信ホストまでの配信経路が幾つかのネットワークによって構成されていて、その途中に帯域の狭いネットワークがある場合などに、中継ノードでパケットの中継を行う事ができなくなり、パケットの損失が発生する。また、経路上のネットワークには別のトラフィックが存在し、そのために使用可能な帯域が減少するのでパケットの損失が発生する。
- (3) ネットワーク品質の問題
ノイズなどによりビット誤りやバースト誤りなどが発生し、それらが訂正できない場合にパケットの損失が起きる。無線ネットワークで特に起きやすい。

上記の原因は動的に変化するもので動的に対応する必要がある。それぞれの原因の判定方法について以下に検討する。

受信ホストの受信能力の判定 ネットワークでの欠落を考えないとして、ある時間の間に送信ホストが送信した情報がすべて受信できた場合は、その時の送信スピードは受信能力に対して余裕があると考えられ、一部でも受信できなかった場合は受信能力の限界であると考えられる。したがって送信された量と受信した量を比較することによって、送信量に対する受信能力の充分さが判定できる。

ネットワークの輻輳の判定 送信ホストは送信しているネットワークのローカルな輻輳を送信が困難になることによって検出できる。ネットワークの品質を無視すれば、送信ホストから受信ホストの間までのネットワークにおいて輻輳が発生した場合、送信ホストが送信した情報のうち一部がルーター等で欠落し受信ホストまで

配送されない事になる。したがって受信ホストが受信した情報に欠落を検出すれば輻輳を検出したことになる。

ネットワーク品質の判定 ネットワークの品質が悪い場合は、受信したパケットの検査によって破壊されたパケットが破棄されるので、この割合を計算する事で経路上のネットワークの品質を知ることができる。

4.2 SMFTP における提案方法

受信ホストの受信能力と利用可能なネットワークの帯域幅のどちらか低い方が、パケットの損失を起こさずに送受信可能な帯域幅と考えられ、送信したパケットのうち、この帯域幅から溢れた部分は受信ホストでのパケットの損失として検知される。パケットの損失があった場合に送信速度を低下させ、ない場合には増加させることによって、送信速度を動的に変化する送受信可能な帯域幅に追従させることができ、パケットの損失が少ない高速な送受信が可能になる。

許容未受信率 SMFTP では送信された情報のうち、ある一定の割合までの損失を許容しておりこの値を許容未受信率と呼んでいる。

無線環境などのエラー率の高いネットワークでは、送信端末の送信速度に関わらず、ある確率で送信されたデータが失われると考えられる。この場合、送信速度を低下させても情報の損失はなくなるので、送信速度が漸減するという問題がある。したがってネットワークの品質によるデータの損失と、受信端末の処理能力不足やネットワークの輻輳によるデータの損失を区別する必要がある。一般的に比べて小さいので、ある割合までの損失はネットワーク品質に起因するものと見なし、これを区別するために許容未受信率という値を設定し、受信ホストは、この許容未受信率以上データを受信できなかった場合のみ応答するようにすれば、前記の問題を解決できる。

以上の考察より、SMFTP では次のような方法でフローコントロールを実現した。

4.3 実現方式

フローコントロール用の PDU として PRO_FLOW_REQ と PRO_FLOW_RES を定

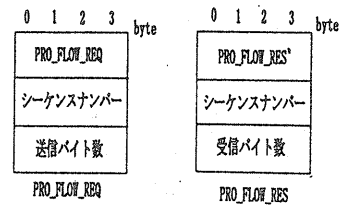


図 4: フローコントロール用の PDU

義する。(図 4)。

PRO_FLOW_REQ は、送信ホストが送信スピードと応答タイミングを受信ホストに通知するために使用する。PRO_FLOW_RES は、受信ホストが受信速度を送信ホストに通知するために使用する。

これらの PDU を使用した動作シーケンスは以下の通りである (図 5 参照)。

- (1) 送信ホストは PRO_DATA で情報の送信を開始するとともに、ある時間毎に PRO_FLOW_REQ を送信する。この時にシーケンスナンバー、及び前回 PRO_FLOW_REQ の送信から今回の送信までに送信したデータのサイズを送信バイト数に設定して送る。
- (2) 受信ホストは PRO_DATA を受信した場合、そのサイズを積算していく。
- (3) PRO_FLOW_REQ を受信した受信ホストは、送信ホストが送信したデータサイズが分かり、前記積算した値から、その間に実際に受信したデータサイズが分かり、送信されたデータの損失率がわかる。損失率が許容未受信率以下であれば、何も応答しない。もし許容未受信率以上であれば、PRO_FLOW_RES を送信ホストに送る。このとき PRO_FLOW_REQ のシーケンスナンバーの値、および実際に受信したデータサイズをそれぞれ設定する。
- (4) PRO_FLOW_RES を受信した送信ホストは、それを PRO_FLOW_REQ の送信間隔で割って、受信ホストの受信可能スピードを計算する。複数の受信ホストから PRO_FLOW_RES を受信した場合は、計算された受信可能スピードのうちの最低なものを受信ホストの受信可能スピードとして保存する。

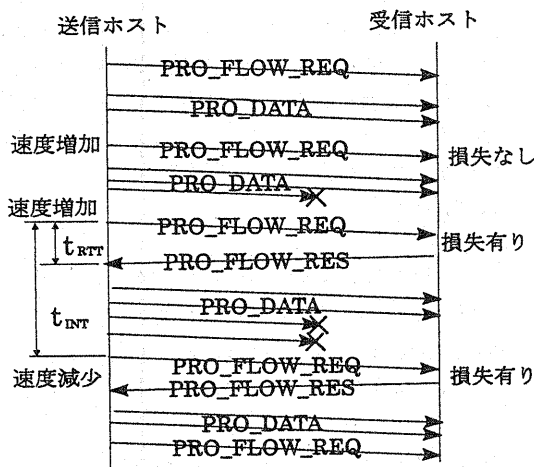


図 5: フローコントロールの動作

- (5) 次に PRO_FLOW_REQ を送信するタイミングで送信速度の変更を行う。送信ホストはもし PRO_FLOW_RES を受信しているならば、送信速度をその値に変更する。もし PRO_FLOW_RES を受信していないならば、送信速度を増加させる。

4.4 実装

実装は Windows 上で行った。IP マルチキャストは Winsock を通して使用している。

実装する場合には PRO_FLOW_REQ の送信間隔 (t_{INT})、その間に送信するパケットの数とその一つずつのサイズが問題となる。 t_{INT} が短ければ短いほどネットワークの輻輳状態や受信ホストの受信能力の増減に対する追従が良くなる。IP マルチキャストでの損失はパケット単位で起こるので、ある程度の精度をもって受信能力やネットワークの速度を測定するためには、パケットの大きさが小さく数が多くなければならない。しかし短い間隔で送られている PRO_FLOW_REQ の間に多数の小さなパケットを送受信することは、プロトコル上のオーバーヘッドの増大や、処理の負荷の増大を招く。 t_{INT} を長くした場合は上記の逆になり、その間に送るパケットのサイズと数の組み合わせには自由度が生まれるがネットワークの変化に対する追従性が悪くなる。そこで実装では、送信数の最大値と最小値を定めて、その間に収まるように調整している。

t_{INT} は PRO_FLOW_REQ を送信してからその応答が返ってくると予想される時間 (t_{RTT} と呼ぶ) までの間隔より大きい必要がある。なぜなら応答が返ってくる前に新たな PRO_FLOW_REQ を送信することは本来送信速度を減少させるべき場面での送信速度の増大を招き、その結果多数の情報の欠落の原因となるからである。実装においては、PRO_FLOW_REQ, PRO_FLOW_RES に含まれるシーケンスナンバーを利用して t_{RTT} を計測し、 t_{INT} を t_{RTT} より大きくなるように調整している。

パケットの損失がない場合の送信速度の増加率は 10% とした。また、ネットワークの能力に素早く適応するために、パケットの損失が起き始めるまでは速度の増加率は 100% とした。初期送信速度は 2KB/s、初期送信間隔は 500ms である。

予備的なテストの結果、通常の状態ではパケットの損失の確率は数%程度であったので、許容未受信率を 20% とした。

4.5 評価

2 台のデスクトップパソコン、2 台のノートパソコンの 4 台を 10MEthernet の LAN で接続して送受信のテストを行い評価した。OS はすべて Windows95 である。

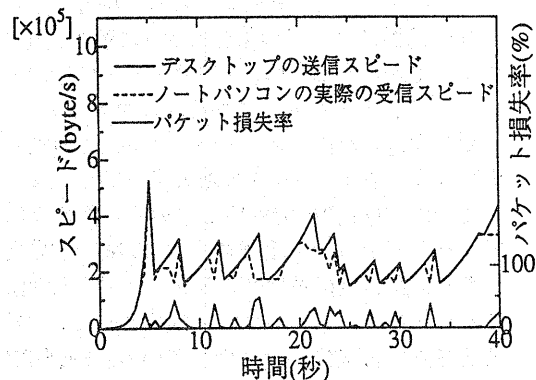


図 6: 受信能力の低い受信ホストがいる場合の動作

デスクトップが送信、残り 3 台が受信 デスクトップから 15MB のファイルを送信した時の送信速度、ノートパソコンの受信速度、パケットの損失率の変化を図 6 に示す。図

6によって分かるように、最初スロースタートにより大きく送信スピードがあがったあと、エラー率の増減によって送信スピードが変化しており、平均して200KB/s程度の送受信スピードになっている。またパケットの損失率は送信スピードによって変化している。先の評価でデスクトップ同士での送受信スピードは最大で1MB/sであるのでノートパソコンの受信能力が低い事が低い送受信スピードの理由であり、端末の受信スピードに応じた送信スピードに適應するという動作になっている。

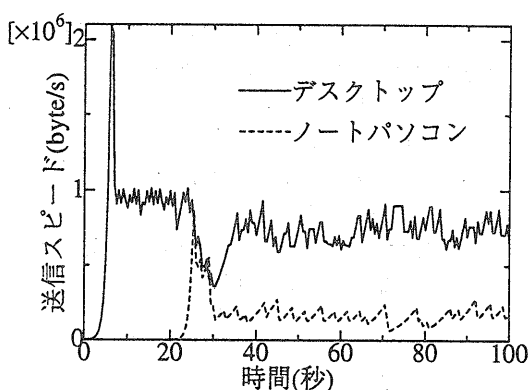


図 7: ネットワークが混雑した場合の送信スピードの変化

デスクトップ同士、ノートパソコン同士が送受信 デスクトップで送信を行っている最中に、ノートパソコンでも送受信を行った場合のそれぞれの送信スピードの変化を図 7 に示す。図 7 によって分かるように、デスクトップの送信スピードは、デスクトップ間の送受信が1MB/s程度で安定した後、ノートパソコン同士の送受信によるトラフィックの増大に対応して減少して800KB/s程度で安定している。先の評価でノートパソコンの受信能力は200KB/s程度であり、実験の結果ではデスクトップの送受信スピードは、ノートパソコンの送受信によって消費された帯域分だけ減少している。SMFTPのフローは共存できることが分かる。

5 おわりに

高信頼性マルチキャストプロトコルを開発し、フローコントロールを実装した。今後の課題としてはTCPとの親和性が挙げられる。よく知られているようにTCPもフローコント

ロールを行っている。フローコントロールを行っている別々のプロトコルが同時に動作した場合には、両方のプロトコルが均等に帯域を使うようには動作せず、一方が最大、もう片方が最小のスピードでの送受信になりかねない。これを解決するのが今後の課題である。

参考文献

- [1] S.Deering, Host Extension for IP Multicast, RFC1112, August 1989.
- [2] W.Fenner, Internet Group Management Protocol Version 2, RFC2236, November 1997.
- [3] D.Waitzman, C.Partridge and S.Deering, Distance Vector Multicast Routing Protocol, RFC1075, November 1988.
- [4] D.Estrin, D.Farinacci, A.Helmy, D.Thaler, S.Deering, M.Handley, V.Jacobson, C.Liu, P.Sharma and L.Wei, Protocol Independent Multicast-Sparse Mode, RFC2362, June 1998.
- [5] A.Ballardie, Core Based Tree Multicast Routing, RFC2189, September 1997.
- [6] B.Whetten, M.Basavaiah, S.Paul, T.Montgomery, N.Rastogi, J.Conlan and T.Yeh, THE RMTP-II PROTOCOL, Internet Draft, April 1998.
- [7] K.Miller, K.Robertson, A.Tweedly and M.White, StarBurst Multicast File Transfer Protocol, Internet Draft, April, 1998.
- [8] Tony Speakman, Dino Fainacci, Steven Lin and Alex Tweedly, PGM Reliable Transport Protocol Specification, Internet Draft, January 1998.
- [9] D.Rubenstein, J.Kurose and D.Towsley, Real-Time Reliable Multicast Using Proactive Forward Error Correction, NOSSDAV 1998.
- [10] L.Vicisano, L.Rizzo and J.Crowcroft, TCP-Like Congestion Control for Layered Multicast Data Transfer, INFOCOM 1998, August 1997.