

# 「なかよし」における高信頼性マルチキャストプロトコルの フロー制御の検討と実装

1 F - 5

市村 重博 坂田一拓 倉島 顕尚 前野 和俊

NEC C&amp;C メディア研究所

## 1 はじめに

筆者らは、モバイルグループウェアシステム「なかよし」を研究開発している。「なかよし」は携帯型のノートPCを持ち寄って、どんな場所でもプレゼンテーション等の協同作業が出来るシステムである。ネットワークには、どんな場所でも持ち寄った端末だけでネットワークを構成できるPHSアドホックネットワークを使用している。協同作業においては、使用する資料をファイルという形で配布する事が多い。筆者らは、協同作業を行う時に使用する資料ファイルの配布を効率的に行うために高信頼性マルチキャストプロトコル(NMFTP)を研究開発してきた[1]。本稿では、この高信頼性マルチキャストプロトコルにおけるフロー制御について検討・評価する。

## 2 フロー制御の必要性

「なかよし」の利用形態としては、客先での営業や、会議室に持ち込んで、有線系のシステムと連携して協同作業を行うといった事が考えられる。したがって受信端末の処理能力が違い、また有線側から資料を送信する場合にはPHS側はスピードが遅いので、送信端末が最大のスピードでマルチキャストで送信すると、多数のパケットが未受信となる。この結果再送が増し、送受信の確実性やネットワークの利用効率が低くなる。

また「なかよし」では手書き・ポインタ・会議プロトコル等の送受信があり、円滑な協同作業のためにはこれらが優先的に送信される必要がある。

さらにファイルの送受信中に途中参加者・途中退席者が発生する事が考えられる。従って、フロー制御の必要機能は以下の通りである。

- (1) 個々の端末の受信能力に適したスピードで送信する。
- (2) 手書き・ポインタ・会議プロトコル等が優先されるように送信する。
- (3) 途中参加者・途中退席者に対応する。

## 3 フロー制御の方法の検討

### 3.1 個々の端末の受信能力に適したスピードで送信する方法

個々の受信端末の受信能力に適したスピードで送信者が送信するには、それぞれの受信端末が受信可能なデータのスピードを送信者に通知すれば良い。この方法として、受信者がある単位時間の間に実際に受信したデータサイズが実際に送信されたデータサイズより小さい場合は、この受信データサイズを受信者の単位

時間あたりの受信可能データサイズとみなして送信者に通知し、もし送信者が通知を受けた場合はその値を前記単位時間で割って受信者の受信可能スピードを計算し、受けていない場合は送信スピードを全受信者の受信可能スピードとみなし、このスピードで送信するという方法が考えられる。

ネットワーク伝送路上の送受信はある確率で送信されたデータが失われると考えられ、この場合、受信端末が実際に受信したデータサイズは実際の受信可能データサイズを表しているとはいえない。一方、受信端末の処理能力不足によるデータの損失や、ネットワークの混雑によるデータの損失がおきている場合の受信データサイズは実際の受信可能データサイズと考える事ができる。この二つの場合を区別できれば、上記の方法はより有効になる。

一般的にいてネットワーク上での通常の損失の割合は、受信者の受信能力不足やネットワークの混雑がおきている場合の損失の割合に比べて小さい。そこで未受信率 $=1 - \text{単位時間あたりの(受信データサイズ/送信データサイズ)}$ とし、この未受信率の閾値を適切に定めれば、二つの場合を区別できる。この閾値を許容未受信率と呼ぶ。許容未受信率以下の未受信率の場合はネットワーク上でのデータの損失とみなして、送信者に対する受信データサイズの通知を行わず、それ以上の場合は受信能力不足によるデータの損失とみなして受信データサイズを送信者に通知するようにすれば、送信者は正しい受信可能スピードを計算出来る。

### 3.2 手書き・ポインタ・会議プロトコル等が優先されるように送信する方法

手書き・ポインタ・会議プロトコル等の送受信が行われた場合は、ネットワークに負荷がかかる。ネットワークが混雑していれば、これらのデータの送受信は円滑ではなくなる。したがって、ネットワークの混雑を検知して回避する事がこれらのデータの送受信を優先することになる。

送信者のローカルネットワークが混雑した場合は、NMFTPのファイルデータの送信スピードが少し低下する。これは送信バッファの溢れという形で検出できる。この場合はNMFTPが送信スピードをより低い値に設定する。受信者のローカルネットワークが混雑した場合は、受信者のファイルデータの受信能力が低下し、前節の方法により送信者に通知され送信スピードが低下する。つまりNMFTPは手書き・ポインタ・会議プロトコル等の送信中は送信スピードを減らし続け、それらのデータが優先的に送受信されるようにする。

### 3.3 途中参加者・退席者への対応

途中参加者の受信能力が低い場合は、NMFTP自体のプロトコルも未受信になりやすくなり、NMFTP

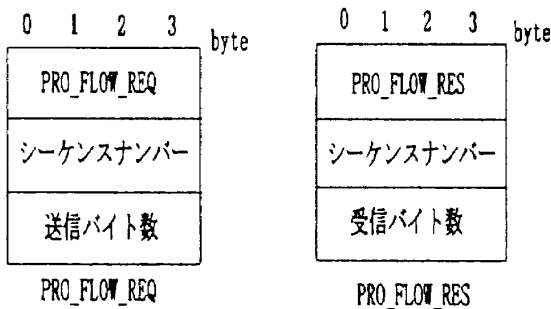


図 1: フロー制御用のプロトコル

の動作の確実性が損なわれる。また受信能力の低い受信者が途中退席した場合はより速やかに送信スピードを増加させる事が必要である。そこで NMFTPResetFlowControl という API を用意した。この API が呼ばれると NMFTP は送信スピードを最低に設定し、さらに送信スピードを増加させる場合の増加率をより大きくする。これによって NMFTP 自体の動作の確実性が向上し、より早く速い送信スピードに達するようになる。資料ファイルを配布する AP は途中参加者・途中退席者があった場合にこの API を呼ぶ事によって、より効率的かつ確実な資料の配布を行う事が出来る。

#### 4 実装

以上の検討からフロー制御を以下の様に実装した。送受信者間でデータ転送スピードを把握するためにこのプロトコルとして PRO\_FLOW\_REQ と PRO\_FLOW\_RES を用いる。それぞれのプロトコルの構造を図 1 に示す。このプロトコルを用いて送信者と受信者は以下のように動作する。

- (1) 送信端末は資料ファイルのある決まった大きさに分割して受信者に対してマルチキャストする。この送信を行っている間、ある一定時間毎に PRO\_FLOW\_REQ をマルチキャストする。この時にシーケンスナンバー、また前回の PRO\_FLOW\_REQ の送信から今回の送信までの間に送信したデータのサイズを設定して送る。
- (2) PRO\_FLOW\_REQ を受信した受信者は、この中に含まれている送信データサイズと、前回の PRO\_FLOW\_REQ の受信から今回の受信までの間に実際に受信したデータサイズを用いて未受信率を計算する。未受信率がもし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の場合には PRO\_FLOW\_RES を送信者に送る。このとき PRO\_FLOW\_REQ のシーケンスナンバーの値、及び実際に受信したデータサイズをそれぞれ設定する。
- (3) PRO\_FLOW\_RES を受信した送信者は、それを PRO\_FLOW\_REQ の送信間隔で割って、受信者の受信可能スピードを計算する。
- (4) 送信者はもし PRO\_FLOW\_RES を受信しているならば、送信スピードを計算した値に変更する。もし PRO\_FLOW\_RES を受信していないならば、送信スピードを増加させる。

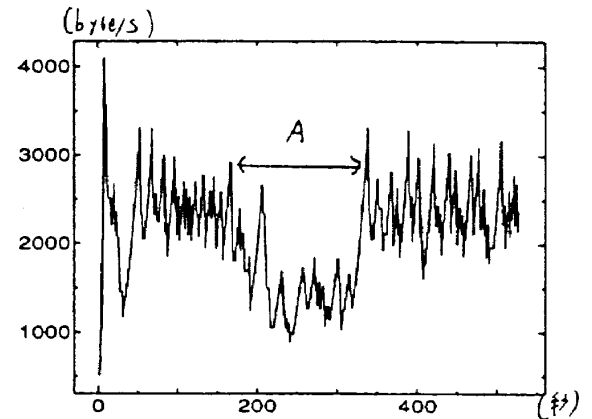


図 2: 送信スピードの経過

また送信バッファの溢れを検知した場合は送信スピードを減少させる。

#### 4.1 評価

2台のノートパソコンで PHS アドホックネットワークを構成し、室内及び 25m 離れた距離で 256 バイト/s でファイルの送受信テストを行った。このときの未受信率は大体 1% から 5% までであった。このスピードに対しては端末の処理能力は十分であり、また PHS の伝送速度に対して送信スピードが遅いのでネットワークの混雑もおきていないと考えられる。したがってこの値は伝送路上のデータの損失確率と考えられる。このテストの結果から許容未受信率は 0.2 とした。

次にファイル送受信を行っている時の手書き・ポインタ等の操作感を評価した。

PRO\_FLOW\_REQ の送信間隔は 1 秒、最低送信スピードから最初の PRO\_FLOW\_RES の受信、もしくは送信バッファの溢れを検知するまでの送信スピードの増加率は 100%、それ以外の送信スピードの増加率は 10%、減少させる場合の減少率は 20% とした。

ノート PC4 台を同一の部屋において会議を開催し、協同作業中にファイルの送受信を行った。そのときの NMFTP の送信スピードを図 2 に示す。

図 2 に示された通り、最初は急激に送信スピードが上昇したあとだいたい 2.5kbyte/s 程度で安定している。もし許容未受信率が適当な値でない場合は、送信スピードは減少しづけるはずであるので、設定した許容未受信率の値は適当であると考えられる。

図の中程 A の部分は、プレゼンテーションプログラムを起動して手書きポインタ等の送受信を行った時である。実際の手書き・ポインタの操作感覚は、表示がすこし遅くなる程度であり、またファイルの送信者、受信者の端末の間でほとんどタイムラグなく表示された。

よって目的とするフロー制御が出来ていると考えられる。

#### 参考文献

- [1] 市村他, “モバイルグループウェアシステム「なかよし」における信頼性マルチキャストプロトコルの実装” 情処 55 全大 2H-5(1998).