L-003

大規模データ収集システムにおける高信頼性マルチキャスト通信
Reliable Multicast Communication for a Large-Scale Data Acquisition System

梶山 真治† 長坂 康史‡
Shinji Kajiyama Yasushi Nagasaka

1. はじめに

素粒子物理学実験における測定器の大型化に伴い、そこで用いられるデータ収集システムの開発が必要不可欠となってきた[1]。この大規模なシステムで問題となるのは、端末間の通信手段である。その中でも特に問題となるのが同一のメッセージを複数の端末に送信する１対多の通信手段である。本研究ではこの１対多の通信手段としてマルチキャスト通信に着目し、大規模データ収集システムのための高信頼性マルチキャスト通信を開発した。

2. システム概要

図１にシステムの構成とデータの流れを示す。このシステムは TR(Trigger)、TS(TriggerServer)、DH(DataHolder)、EB(EventBuilder)、DR(DataRecorder)の 5 つのコンポーネントから構成されている。

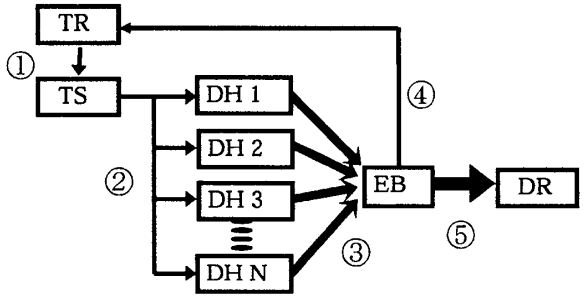

図１．システム構成とデータの流れ

このシステムの目的は、ある事象（イベント）の発生により、イベントフラグメントデータと呼ばれる複数のデータの断片を１ヶ所に集め、イベントごとのデータの集まり（イベントビルトデータ）として保存することである。イベントフラグメントデータはそれぞれの DH で周期的に発生し、DH 内にバッファリングされる。イベント発生が引き金（Trigger）となりイベントフラグメントデータの収集が開始される。あるイベントに対する複数のイベントフラグメントデータをひとまとめとし１つのイベントビルトデータとすることをイベントビルドと呼ぶ。

データの流れは、TR から TS へイベント発生を知らせる Trigger メッセージを送信し（①）、TS は Trigger メッセージを受信すると複数の DH へ Trigger メッセージを送信する（②）。それぞれの DH は Trigger メッセージを受信するとイベントフラグメントデータを EB に送信する（③）。EB はあるイベントに対して全ての DH からのイベントフラグメントデータを受信するとイベントビルドし、TR へイベントビルド応答を送信し（④）、DR へイベントビルトデータを送信する（⑤）。DR は EB からイベントビルトデータを受信するとデータをファイルに書き込む。EB から TR への応答は Trigger メッセージの過剰送信を防ぐために用いられ、DH でのイベントフラグメントデータは周期的に生成される。以上が１つのイベントに対するデータの流れであり、システム稼動時は複数のイベントが並列に処理される。

このシステムで想定しているコンポーネントの台数、コンポーネント間でやり取りされるデータサイズは、DH:1000 台、その他は数台、Trigger メッセージは数十バイト、イベントフラグメントデータは数キロバイトと Trigger メッセージより大きい。

3. 通信システム

このシステムにおける TS-DH 間の通信は同一のメッセージを送信する１対多の通信である。この通信には信頼性と遅延の短さが要求される。この１対多通信の実現方法として２つの通信手段が考えられる。

まず１つめは TCP 通信を用いて TS とそれぞれの DH との間にコネクションを確立し、TS からそれぞれの DH へ１つずつメッセージを送信する手段である。TCP 通信は、２つの端末間でデータを送受信する１対１のユニキャスト通信であり、信頼性を高めるための機能などを提供している。その反面、機能提供に伴いヘッダ処理にかかる時間が大きくなってしまうという欠点を持っている。この方法では、DH の増加に伴いネットワークへ流れるデータ量も増加してしまうことと、それぞれの DH に１つずつメッセージを送信しなければならないため、それぞれの DH でのメッセージ受信時刻が大きくずれてしまい TR-DH 間の遅延が大きくなってしまう。

２つめはマルチキャスト通信を用いる手段である。この通信はトランスポート層のプロトコルとして UDP を利用し、TS から１回メッセージを送信するだけで複数の DH へ一斉にメッセージを届けることができるという１対多の通信手段である。DH が増加してもネットワークへ流れるデータ量は一定で、それぞれの DH でのメッセージ受信時刻のずれも小さい。さらに、マルチキャスト通信は UDP 通信を用いているため、TCP 通信のような再送制御やフロー制御といった信頼性を高める機能をサポートしていない。このためヘッダ処理にかかる時間が軽減される利点があるが、もしメッセージが何らかの原因で DH へ届かない場合、その通知もせず、復元も不可能となるので信頼性が低いという欠点がある。

両手段の問題点を比較した場合、TCP 通信を用いた通信手段の問題点は TCP 通信の仕様に大きく影響を受けており、変更が比較的困難であることから、マルチキャスト通信を用いた通信手段の開発を行った。

4. 高信頼性マルチキャスト通信

マルチキャスト通信の信頼性の低さから、TS-DH 間の通信にこの通信をそのまま用いることはできない。そこでマルチキャスト通信にアプリケーションレベルの再送制御を付加することにより高信頼化を図った。その再送制御の形態には大きく分けて２つある。両者の違いは再送制御を送信側（TS）で行うか、受信側（DH）で行うかである。図２に両者のデータの流れを示す。

前者の再送制御を送信側で行う場合の流れは、まず TS からマルチキャストメッセージを送信する（①）。それぞれの DH でメッセージが正確に受信されると、TS へ ACK（肯定応答）を返信する（②）。図２の DH3 のように、何らかの原因でメッセージが DH に届かず、ある時間が過ぎても ACK

† 広島工業大学大学院工学研究科情報システム工学専攻、
Graduate School of Engineering, Hiroshima Institute of Technology

‡ 広島工業大学工学部知的情報システム工学科、
Faculty of Engineering, Hiroshima Institute of Technology

が返ってこない場合、その DH に対して再送する（③）。このように再送のきっかけとなるメッセージの損失を送信側で発見するものが送信側再送制御方式（Transmission-side Resend control System: TRS）である。

一方、再送制御を受信側で行う場合の流れは、まず TS からマルチキャストメッセージを送信する（①）。そして DH でメッセージを受信し、メッセージの内容から損失が発生したか調べ、もし損失が発生したとみなしたら、NACK（否定応答）を返信する（②）。TS は NACK を受信すると、再送を行う（③）。損失発生を調べる方法は、メッセージには識別番号が割り振られており TS から番号順に送信されてくるので、受信したメッセージのシーケンス番号が番号順でない場合、損失が発生したと認識する。このようにしてメッセージの損失を受信側で発見し送信側に知らせるものが受信側再送制御方式（Reception-side Resend control system: RRS）である。

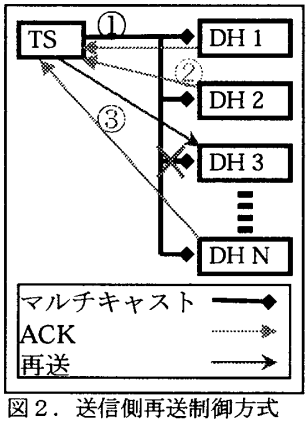

図２．送信側再送制御方式

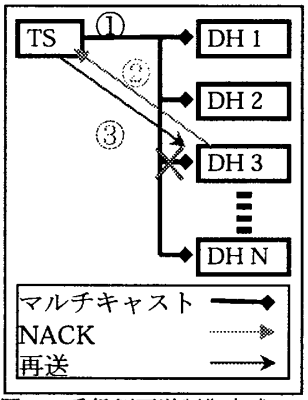

図３．受信側再送制御方式

TRS は DH が増加するにつれ、DH からの ACK の総数が増加し、TS への負荷が増加してしまうという点からスケーラビリティに関して問題があるといえる。一方 RRS では DH が増加した場合でも、TS への負荷の増加が抑えられる。本研究は大規模なシステムを対象としているためスケーラビリティも重要となる。そこで、マルチキャスト通信に RRS を付加した高信頼性マルチキャスト通信システムの開発を行った。

一般的に知られている高信頼性マルチキャストプロトコルのなかでは SRM(Scalable Reliable Multicast)が最も RRS に類似している[2]。この両者の共通点は、受信側でシーケンス番号の抜けからパケットの損失を発見し NACK で再送を要求する点である。相違点として NACK と再送の通信手段が挙げられる。SRM では NACK をマルチキャストで送信し、この NACK にマルチキャストセッションに参加している誰もがマルチキャストで再送できるのに対して、RRS では NACK をユニキャストで TS のみに送信し、TS はユニキャストで NACK を送信してきた DH に再送する。これは、このシステムではマルチキャストメッセージを受信する DH は EB へイベントフラグメントデータを送信しなければならなく DH に余分な負荷をかけないためである。DH の増加により TS への NACK が増加した場合は、TS の台数を増やすことで負荷分散を施す。

5. 性能評価

開発したシステムの性能を評価するため、システムを動作させ性能試験を行った。TS－DH 間の通信にマルチキャスト通信に RRS を付加した高信頼性マルチキャスト通信、TCP 通信によるユニキャスト通信をそれぞれ用いたシステムを動作させ性能比較を行った。測定は Dual Athlon 1.6GHz の CPU、256Mbyte のメモリ、Intel e100 の NIC を装着し、OS に Linux(カーネルバージョン 2.4.18)が動作している PC を使用した。ネットワーク環境は PCI GX-08SX に CISCO Catalyst3500 series XL と REPOTEC RP-G3240U が接続されている。システムパラメータとして、DH の数を 8、16、32、48、58 台、イベントフラグメントデータサイズを 256、512、1024、1536 Byte とそれぞれ増加させ、Trigger 送信レートを 100 Hz に固定した。

図４に両システムの TR における Trigger メッセージを送信してから、イベントビルド応答を受信するまでの経過時間を示す。DH の数が 16 台以下の場合、目立った違いはないが、16 台以降では DH の増加に伴い、両システムとも経過時間が大きくなっているが、RRS のほうが TCP に比べ増加率が抑えられている。これはマルチキャスト通信を用いることにより TS-DH 間の重複したメッセージ送信処理を削減できたためである。この結果から１対多の通信にマルチキャスト通信に RRS を付加した高信頼性マルチキャスト通信を用いることにより、システムの効率が向上し、スケーラビリティに関してもよいといえる。

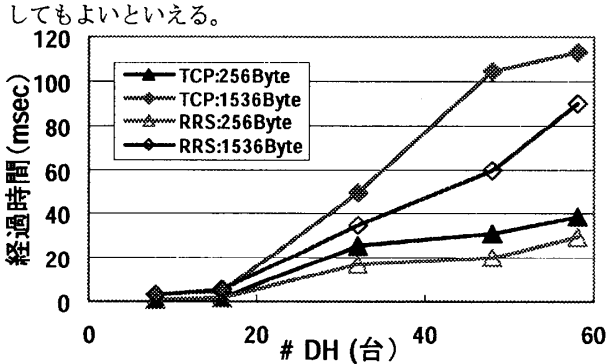

図４．TR における Trigger メッセージを送信してイベントビルド応答を受信するまでの経過時間

6. 再送制御の下位層への移行

これまでの再送制御はアプリケーションレベルでの制御であった。この再送制御をより下位層のトランスポート層で行うことによりさらなる効率化を図った。トランスポート層で再送制御を行うことにより、カーネル空間とユーザ空間との間で無駄なデータのコピーがなくなり、処理が軽減される。そして、メッセージの損失をすばやく発見できるなど、処理の軽減や再送制御の効率化が期待できる。

7. まとめ

大規模データ収集システムにおける１対多の通信に着目し、この通信手段としてマルチキャスト通信を用いたデータ収集システムの開発を行った。その中でマルチキャスト通信の欠点である信頼性の低さを補うため、独自に開発した受信側再送制御方式を付加した高信頼性マルチキャスト通信を開発した。性能試験の結果、この高信頼性マルチキャスト通信を用いることにより、ネットワークに流れる重複メッセージを削除でき、システムの効率が向上することを確認した。さらに再送制御をアプリケーション層から、より下位層のトランスポート層で行うことによりシステムの性能が向上した。

8. 参考文献

[1] ATLAS Collaboration, “ATLAS High Level Trigger, Data Acquisition and Controls”, ATLAS Technical Design Report, ATL-DAQ-2003-052, 2003

[2] S. Floyd, V. Jacobson, C-G. Liu, S. McCanne, L. Zhang: "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEE/ACM Transactions on Networking, Dec. 1997.