A-036

高完全性マルチキャストの提案

Proposal of High Integrity Multicast

金田 直樹† 塩見 格一†
Naoki Kanada Kakuichi Shiomi

1 はじめに

航空に係る業務において一般に，見落とし，聞き間違い，思い込みといった誤りを避けなければならない．そのため，複数の人間により互いに確認を行うこと一般的に行われている．例えば航空管制官はパイロットに対し，管制指示を必ず復唱させる．また，航空管制の場合，全ての空域には隣接空域があるが，空域と関係なく航空機は飛行する．そのため隣接空域の担当者との調整業務が必然的に発生する．また，航空機が通過を予定していれば隣接していない空域とも調整業務は発生するため，調整業務は広範囲にわたる．本発表は，航空管制において重要である意思決定を助けるために，情報を矛盾なく共有する方法について提案を行う．

現在の情報共有は，クライアント・サーバ方式により，サーバにある唯一のデータベースからクライアントが情報を取得することにより情報共有を実現しているが，複数の計算機上にあるデータベースを完全に同期させることは容易ではなく，サーバの冗長構成は難しい．

サーバを使用しない情報共有方式としては Peer-to-Peer(P2P) システムが代表的なものであり，Pure P2P によるファイル共有が有名である．Pure P2P システムは完全自立分散システムのため耐障害性が高いが，各ノードが保持するデータが無矛盾であることを保証することが困難である．メタ情報をサーバに持たせる Hybrid P2P 方式は，この問題点はある程度解決されるがサーバが必要になるため，クライアント・サーバ方式の問題点を部分的に引き継いでしまう．この問題を解決する方法として，CAN[1] や Chord[2] のような分散ハッシュ表 (Distribution Hush Table, DHT) によるデータ検索方法が知られている．しかし，DHT は検索を効率的に行う仕組みであり，ある特定のデータに対して多くのノードからアクセスがあった場合，要求されたデータを持つ計算機にアクセスが集中してしまうという問題は依然として残る．

ブロードキャストやマルチキャストなど 1 対多の通信を利用した情報共有はリアルタイム性を持ち，拡張性が高く，部分的な故障に対して強いという利点がある．また，通信媒体によっては 1 対多の通信を 1 対 1 の通信路を多数張るより効率的に行うことができる．そのような媒体の例として，ブロードキャストを基本とする無線通信や，歴史的に 1 対多通信の容易な Ethernet (IEEE802) がある．しかし，通常の 1 対多通信は受信者による確認応答 (Ack) が全くないか，すべての受信者が送信者に確認応答を返すかのどちらかである．確認応答のないものは信頼性に欠け，すべての受信者が送信者に確認応答を返す方法は確認応答が送信者に集中する (Ack Explorsion) という問題がある．そこで本論では，リアルタイム性を持つ情報共有の方法として，確認応答つき 1 対多通信の改良を提案する．

†独立行政法人 電子航法研究所 管制システム部
Traffic Management Systems Division, Electronic Navigation Research Institute

2 航空管制に係る通信の性質

航空管制に係る通信はいくつかの性質がある．第一に，情報が正しく相手に届いたかどうかは重要である．そのため，必ず確認をとる．例えば，管制官はパイロットに管制指示を復唱させる．第二に，管制指示のように重要度が高い情報はリアルタイム性が高く情報量は少ない．逆に，気象情報のような比較的重要度が低い情報はリアルタイム性は低く情報量が多い．第三に，数万の航空機が同時に飛行している状況は少なくとも我が国では近い将来も含め考えにくい．第四に，フェールセーフ，またはフェールソフト性が求められる．例えば 1 つのノードが故障する場合，ある部分がすべて故障する場合，そして自分以外の全部が故障している，すなわちスタンドアロン運転のそれぞれの場合につき，可能な限り故障していない部分による縮退運転を行うことができることが望ましい．今回の想定は地対地通信であるので，通信手段としては，Ethernet と Internet Protocol (IP) を用いた通信を仮定することとした．

3 提案手法

3.1 前提条件

リアルタイム性がなく情報量が多い通信は従来手法のカバーする範囲であると考える．そこで本論では重要度の高い，情報量が少なくリアルタイム性と完全性を必要とする通信をどのように行うかについて論じる．完全性

とはシステムが使用できないときに警報を発する能力のことである.

情報量の少ない通信は, パケットに連番を付与することによるパケット損失の検出は難しい. なぜなら, パケットに連番を付与する方法によるパケット損失の検出は, 到着したパケットの連番が非連続であることによりパケットの損失を検出するからである. このとき, 最後のパケットは後続のパケットがないためパケット損失が検出できない. また, 不完全ながら通信が行えている場合にはパケット損失を検出できるが, 全く通信ができない場合にパケット損失を検出できない. そこで本論では逆に, 1パケットに必要なデータがすべて入るものと仮定する. これはかなり強い仮定である. 前述のような航空管制に係る通信の特徴より, 重要度とリアルタイム性が高い情報の情報量は小さいと仮定することができる. また, 通信相手のアドレスは事前にすべて知っていること, 通信相手のリストはすべての参加者が共有していること, 故障率は十分小さく, 通信相手が一斉に故障しないこと, 通信相手に頻繁な増減はないことを仮定する.

3.2 従来技術

IP による 1 対多の通信はマルチキャストとブロードキャストの2種類がある. ブロードキャストはルータを超えた通信ができないので広範囲の通信には向いていない. 信頼性の高いマルチキャストを実現するための Reliable Multicast が研究されている [3, 4]. Reliable Multicast は TCP[5] と部分的に同様の機能を持つマルチキャストの実装を目指している. 誤り訂正符号を付加することによって伝送路の信頼性を得る方法, 確認応答をルータにより集約する Tree Based Ack, 情報が届かなかったことを送信者に通知する Nack (Not Ack) などが提案されている. しかし, Tree Based Ack は途中経路のルータが全て対応しなければ使えないこと, 再送要求を送出する NACK は完全に通信路が切断されたことを検出できないという問題点がある. また, これらはマルチキャストの利点のうち, 従来から着目されている利点である, 多くのユーザに対して映像のように大規模なデータを配信することを目的とした技術であり, リアルタイム性を求めているわけではない. また, 誤り訂正符号の付加により伝送路の信頼性を得ても相手に情報が伝達できたかどうかはわからないことから, TCP の完全な代替にはなり得ない.

3.3 情報共有手順の提案

本論では, 集合に関する以下の標準的な記法を用いる. 集合 A の要素数を $|A|$ で表す. $\forall i \in A$ は「集合 A に属す, すべての要素 i(に対して)」 を意味する. 2^A は集合 A の巾集合を表す. 例えば $A = \{0,1,2\}$ なら, $2^A = \{\{\},\{0\},\{1\},\{2\},\{0,1\},\{0,2\},\{1,2\},\{0,1,2\}\}$ である.

ここでは n 個のノードが相互に接続されたネットワークを考える[1]. 各ノードに番号 $0,1,2,\ldots,n-1$ を振る. 各ノードは自分の番号を知っているものとする. $A = \{0,1,\ldots,n-1\}$ とする. 各ノードの送信先アドレスは別の方法を使用し, すべてのノードで共有しているものとする.

以上の状況において, ノード 0 が情報を発信するとして一般性を失わない. 提案する手順は以下の通りである. Step 1 を単に 1 と略記する.

1. ノード 0 はすべてのノード $\forall i \in A$ に対して情報を送信する. これはマルチキャストとして実現される.
2. すべてのノード $\forall i \in A$ はそれぞれ, 写像 $adj : A \to 2^A$ により, 隣のノードの集合 $B_i = adj(i)$ を選定する. adj の詳細については後述する.
3. ノード i はすべての隣のノード $\forall j \in B_i$ に対して 0 からの情報が届いたかどうか, ユニキャストにより問い合わせを行う.
4. ノード i はすべての隣のノード $\forall j \in B_i$ それぞれから 0 からの情報が届いたかどうかユニキャストにより確認応答を受ける. ここで, 自分と隣が同じ情報を持っているかどうか確認を行うために MD5[6] や SHA-1[7] のようなハッシュ関数により計算されたハッシュ値を確認応答に含める. これによりノード i は隣が同じ情報を持っていることを確認する.
5. ノード i はすべての隣のノード $\forall j \in B_i$ より確認応答を得た場合以外, ノード 0 に再送要求をユニキャストにより送信する. 再送要求のパケットには確認応答を得られなかったノード (故障ノード) の番号を若い順にパケットに入る範囲で含める.
6. ノード 0 は再送要求が来るか, 確認応答が来なかった場合に全ノード A に対して再送を行う. タイムアウト時間の決定はいくつかの研究が知られている. ここではパケットにタイムスタンプを付加することによる往復時間 (Round Trip Time, RTT) の計測による方法 [8] を利用することが妥当であると考えられる. ここで $0 \in A$ より, ノード 0 も上記 2,3,4 の作業を行うことに注意せよ.
7. 再送回数が別に定める $maxfail$ 回を超えたら再送を終了し, ノード 0 は, ノード k の故障報告の回数が, $maxfail$ 回であれば k をマルチキャストグループから外す (disjoin).

[1] ここではマルチキャストに参加するホストをノードと定義しており, マルチキャストに参加していないクライアントや途中のルータ等は含んでいないのでネットワーク全体をグラフ $G = (V, E)$ としてモデル化したときに $|V| \neq n$ であることに注意せよ.

3.4 利点と問題点

上記情報共有手順の利点は，完全な分散システムであるため，Pure P2P システムと同様，頑健であり，耐障害性が高いこと，一部のノードが故障しても他の部分に影響を与えづらいフェールソフト性を持つこと，そして効率性と拡張性を両立していること，システムが使用できないときに警報を発する，という完全性を持つこと，などが挙げられる．

問題点としては前述したとおり，大容量の情報配送に向かないことが挙げられるが，この手順は重要でリアルタイム性が高く，少ない情報を共有するための手順として設計しているので，事前に意図していることであり，制約事項ではあるが問題点ではないと考えている．

また，この手順はノードが「存在しない」ことを検出できるので，故障などに起因する意図しない disjoin を検出するプロトコルとして応用することも可能である．

この手順は写像 adj の選び方により性質が異なる．以下，その点について考察する．

3.4.1 決定的な場合

$adj(i) = \{0\}$ (if $i = n-1$), $\{i+1\}$ (otherwise) である場合，動作，パケット総数の期待値と成功率について考える．このとき，$adj(i) = \{i+1 \bmod n\}$ であり，ノード $i+n$ を i と同一視すると扱いが簡単なので，この節に限り $\bmod n$ をとって考えることにする．なお，$\bmod n$ の記述は省略する．

この場合，ノード i は ノード $i+1$ に情報到達の問い合わせを行う．そのため，ノード $i+1$ に情報が届いておらず，ノード i に情報が届いていれば，ノード i はノード $i+1$ に情報が到達していないことをノード 0 に報告する．そのため，情報が届かないノードの存在をノード 0 は知ることができる．番号の連続したノード $k, k+1, \ldots, k+l-1$ が故障している場合はノード $k+l$ がノード $k+l-1$ の故障を報告するので，やはり，故障ノードが存在していることをノード 0 は知ることができる．

この場合の利点は，第一に adj が全射であるので，すべてのノードに対し情報送信のたび必ず確認を行うこと，第二に動作が単純で判りやすいこと，第三に効率的に配送確認を行うことができることである．これは以下の理由による．近年普及しているレイヤ 2 スイッチによる Ethernet を用いたネットワークでは各ノードが行う確認作業である Step 3 及び Step 4 はそれぞれ並列処理が可能である．そのため処理時間は，Step 1, Step 3 及び Step 4 を合わせて最大の RTT の 3/2 倍しか必要としないことが期待される．また，情報送信が成功した場合のパケット数は Step 1 で 1, Step 3 で n, 及び Step 4 で n であり，合計 $2n+1$ パケットである．これはノード 0 が TCP によって $n-1$ 個のノードとセッションを確立するために必要なパケット数 $3(n-1)$ よりも少ない．

3.4.2 確率的な場合 1

$|adj(i)| = 1$ であり，その選び方が一様ランダムである場合について，パケット総数の期待値と成功率について考える．この場合は，ノード i は一様ランダムに選ばれたノードに対して情報到達の確認を行う．

パケット数は決定的な場合に関する考察の場合と同じである．情報を送信できていないノードが存在するにも関わらず正しく送信が行われたと誤る確率の下界 $P(n)$ は以下のように与えられる．

$$P(n) = \left(1 - \frac{1}{n}\right)^n$$

$P(n)$ は n に関して単調増加であるため $n \geq 6$ ならば

$$\frac{1}{3} < \left(\frac{5}{6}\right)^6 \leq P(n) < \lim_{n\to\infty} P(n) = \frac{1}{e}$$

が成り立つ．また，$n \geq 2$ ならば $P(n) \geq 1/4$ である．この場合，決定的な場合のように確率 1 の送達確認はできない．しかし，故障して応答しないノードが存在する場合，1 度情報を送信するごとに 1/4 より大きな確率で故障ノードを発見できる．この確率は $n \geq 6$ ならばこの確率は 1/3 に改善され，どんなにノード数が増加しても 1/2 になることはない．情報の送信を行うごとにこの検査を独立試行として行うので，期待値 4 回 ($n \geq 6$ なら 3 回) の情報送信により故障ノードを発見できる．

この手法の利点と欠点は，決定的な場合と異なり，確率的な確認になるので確率 1 での検出はできない．しかし確率的な検査は想定外の故障に対して頑健であり，予想されない故障に対処可能な可能性が大きいという重要な利点がある．例えば，ノード 0 から ノード 1 と 2 の両方に情報が到達していても，前節で説明した決定的な場合，ノード 1 と 2 の間で通信ができなければ，ノード 2 はノード 1 の故障をノード 0 に報告してしまう．確率的な方法では，ノード 1 の検査を行うノードは確率的に決まるため，ノード 1 と通信が可能なノードがノード 1 の検査を行う場合はそのような誤報告を行わない．また，複数のノードが同時に故障した場合，故障ノード数が n よりも十分小さければ，故障ノードを同時に発見できる可能性もある．この利点がありながら，期待値わずか 3 回で故障ノードを発見できるという効率の低下は完全性を確保するためには問題ではない可能性もある．なぜなら，高完全性システムの目的はシステムが使用できないとき

警報を発することであり，そのときの対処は人間に任されるからである．

3.4.3 確率的な場合 2

$|adj(i)| = 1$ のとき，高い確率で近くのノードを選び，低い確率で遠くのノードを選択することとする．このときのパケット数の期待値は決定的な場合と同じであるが，ネットワークをパケットが通過するコストは安くなることが期待される．

3.4.4 複数のノードへの確認

$|adj(i)| > 1$ の場合は隣のノードが複数存在する．このとき，自分及び隣のノードの持っている情報のハッシュ値を比較し多数決によりどれが正しいかを決定することができる．正しい情報を持っているノードはもはや再送要求を出す必要はない．しかし，完全性を確保するため故障ノードを報告するという意味において送信元であるノード 0 への報告は引き続き意味を持つ．

また，ノードの検査を何度も行うことと等価であるので故障ノードの発見率を高めることができる．決定的な場合は複数のノードが同時に故障したときの発見を，確率的な場合は故障ノードを発見するまでの試行回数の期待値の低下を，それぞれ期待することができる．

4 基本要件の検討

まず，前述のように，ブロードキャストやマルチキャストといった 1 対多通信が容易でなければならない．前述のように，最初の実装としては全てのノードが完全に対称であるという本提案手法が活かせること，普及率等を考え，Ethernet が最適であると考える．しかし，最終的には無線通信への応用も考慮している．例えば，無線通信の中でも 1 対多通信能力が非常に高い衛星と，通信速度は低いものの通信コストの安いモバイルアドホックネットワークを組み合わせることにより，本提案手法は非常に有用な衛星通信の手段となりうる．航空管制においても洋上を飛行する航空機と管制官の間でインマルサット衛星を利用したデータ通信による管制はすでに開始されており，本提案手法の有用性は決して小さくないと考える．

IP による 1 対多の通信は TCP ではなく，User Datagram Protocol (UDP) によるブロードキャストまたはマルチキャストになる．このとき，本提案手法は IPv6 によるマルチキャストを利用して実装することが妥当であると考えられる．理由は以下の通りである．第一に，ブロードキャストパケットはルータを越えられないのでマルチキャストを行う必要があるが，IPv4 ではマルチキャスト非対応のルータが多く，本手法においては途中のルータがすべてマルチキャスト対応にしなければならないため難しい．第二に，本提案手法では 1 パケットに情報が全て入るという仮定をおいているが，1 パケットで送信できる情報量の最大値である Maximum Transmission Unit (MTU) の最小値は IPv4[9] では 576 オクテットと比較的小さいが，IPv6[10] では 1280 オクテットと比較的大きく，1 パケットで実用的な大きさの情報を送信可能である．さらに途中のルータによるパケットの分割 (フラグメント化) は基本的にないこと，すべてのホストに対してマルチキャストの実装が義務づけられていることも挙げられる．

5 おわりに

本論では確認応答を分散したマルチキャストの提案を行った．分散した確認応答により通信路及び各ノードが正しく動作しているかどうか常時検査を行うため，高い完全性と送信者に対する確認応答の集中回避を両立できる．これによりリアルタイム性を持ち，情報の矛盾がなく拡張性が高い情報共有を実現することができる．

将来の課題は以下のものがある．第一に実装を行い，実験することにより *maxfail* のような未決定のパラメータを決定する必要がある．第二に，マルチキャストグループへの join / disjoin が発生したときにノード番号を振り直す方法が未解決である．第三に，多くのノードが一斉に故障しないことを仮定しているため，通常は再送要求は少ないと考えているが，再送要求が膨大になったときにノード 0 に再送要求が殺到することへの対策が必要である．

参考文献

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A Scalable Content-Addressable Network," Proceedings of the ACM SIGCOMM, pp.161–172, ACM Press, August 2001.

[2] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," IEEE Transactions on Networking, vol.11, pp.17 – 32, February 2003.

[3] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, and M. Luby, "The reliable multicast design space for bulk data transfer." RFC2887, August 2000.

[4] J.C. Lin and S. Paul, "Rmtp : A reliable multicast transport protocol," Proceedings of IEEE INFOCOM, pp.1414–1424, 1996.

[5] J. Postel, "Transmission Control Protocol." RFC793, September 1981.

[6] R.L. Rivest, "The MD5 Message-Digest Algorithm." RFC1321, April 1992.

[7] D.E.E. 3rd and P.E. Jones, "US Secure Hash Algorithm 1." RFC3174, September 2001.

[8] V. Jacobson, B. Braden, and D. Borman, "TCP Extensions for High Performance." RFC1323, May 1992.

[9] J. Postel, "Internet Protocol." RFC791, September 1981.

[10] S.E. Deering and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC2460, December 1998.