

1チャンネル上の全順序放送通信プロトコルにおける データ転送手続き†

滝 沢 誠†† 中 村 章 人††

現在、複数の通信実体の協調動作が分散型データベースシステムの同時実行制御、コミットメント制御、分散型問合せ処理等を実現する上で必要とされており、このためには、複数の実体間での高信頼放送通信が求められている。本論文では、EthernetのMAC層で提供されている低信頼放送通信サービスを利用して、全順序放送通信(TO)サービスを提供するプロトコルについて述べる。本プロトコルは、群という複数のサービスアクセス点(SAP)に対して、同一のデータ単位(PDU)を同一の順序で受信するTOサービスを提供するものである。本論文は、分散型の制御を用いて、Ethernet MAC、無線網の放送通信サービスを用いて、TOサービスを複数の実体に提供するためのデータ転送手続きについて述べる。

1. はじめに

現在の通信網は、OSI¹⁾やTCP/IP²⁾といった一対の実体間でのコネクション指向の高信頼通信サービスを提供し、ファイル転送³⁾、メール³⁾、データベース処理⁴⁾等の応用に用いられてきている。二つの実体間の通信に加えて、現在、二つ以上の複数の通信実体間の協調動作が分散型データベースシステム⁵⁾の同時実行制御、コミットメント制御^{6), 15)}、分散型問合せ処理⁶⁾、分散型デッドロック検出⁷⁾等を実現する上で必要とされている。これらの処理では、全実体(例えば、データベースシステム、副トランザクション)の状態によって、次に行う処理が決定されることから、各実体は他の実体との通信が必要になる。複数の実体間での高信頼放送通信が可能であれば、これらの実現は容易となる。しかし、従来の一対一通信を用いて、高信頼放送通信の実現を行うと、通信の負荷が大きく、現実的でない。

ローカルエリア網(LAN)は、媒体アクセス制御(MAC)層⁸⁾で放送通信を提供しており、ある実体がデータ単位を送信すると、全実体はこれを「同時に」受信できる。しかし、CSMA/CD等のMAC層⁹⁾では、上位層の実体であるプロセス間の高信頼放送サービスが提供されてなく、あるプロセスはデータ単位を受信できない場合がある。本論文では、LANのMAC層の特性を生かして、全実体が同一のデータ単位を同

一の順序で受信できる全順序放送通信(TO)サービスを提供するプロトコルの設計と実現について述べる。

高信頼放送通信については、まだ研究が始まったばかりであり、文献10)~13), 18)等の研究がある。文献11)では、一対一の通信を用いて、通信量をなるべく小さくさせるプロトコルについて述べている。文献10)は、Ethernet⁹⁾を用いたプロトコルについて述べているが、ここでは集中型の制御が用いられている。文献12), 13), 18)は、従来のコネクションの概念を複数のサービスアクセス点(SAP)に拡張した群の概念について述べ、EthernetのMACサービスを用いた群の開設方法について論じている。集中型制御では、一つの主制御実体が正しい受信の決定を行う。集中型制御では、主制御実体の決定を待つことによる性能低下と、主制御実体が障害すると全体の障害となる問題がある。これに対して、分散型制御では、各実体自身が正しい受信の決定を行うために、主制御実体が不要となり、集中型制御の持つ問題を解決できる。現在まで、分散型の制御を用いた放送通信プロトコルの研究は、ほとんどなされていない。本論文で述べるプロトコルは、Ethernet、無線網を用いて、全実体がデータを正しく受信したかどうかの決定を各実体自身で、分散して行うものである。本論文では、実体が障害を起こしたら停止すると仮定し、文献14), 20)のように誤動作しないものとする。

2章では、通信モデルを定義する。3章では、EthernetのMACサービスと無線網を抽象化した1チャンネル(1C)サービスについて述べる。4章では、Ethernet上でTOサービスを提供するTOプロトコルのデータ転送手続きについて述べる。5章では、TOプロトコルの正しさと、性能評価について論じる。

† Data Transmission Procedure of Totally Ordering (TO) Protocol on the One-Channel Service by MAKOTO TAKIZAWA and AKIHITO NAKAMURA (Department of Information and Systems Engineering, Faculty of Science and Engineering, Tokyo Denki University).

†† 東京電機大学理工学部経営工学科

2. 通信モデル

本論文で用いる通信モデルを、OSI 参照モデル¹⁾に基づいて述べる。

通信システムは複数の副システムから構成され、各副システムは複数の階層から構成されている。最下位の階層から N 番目の階層を、 $\langle N \rangle$ 層という。 $\langle N \rangle$ 層は、 n 個の $\langle N \rangle$ 実体 $E_1, \dots, E_n$ から構成されている。 $\langle N \rangle$ 層は、 $\langle N-1 \rangle$ サービスを利用して、 $\langle N \rangle$ 実体の協調動作により、新たな価値を付加して、 $\langle N+1 \rangle$ 層に $\langle N \rangle$ サービスを提供する。 $\langle N \rangle$ 実体を協調動作させるための通信規約が、 $\langle N \rangle$ プロトコルである。各 $\langle N \rangle$ 実体 E_k は、 $\langle N-1 \rangle$ サービスアクセス点 (SAP) S_k において $\langle N \rangle$ プロトコルデータ単位 (PDU) を送受信するとする ($k=1, \dots, n$)。各 $\langle N \rangle$ PDU p に対して、 p . DST を宛先の $\langle N-1 \rangle$ SAP の集合とする。 S_k において送信された各 $\langle N \rangle$ PDU p が、一つの $\langle N-1 \rangle$ SAP のみで受信されるとき、この $\langle N-1 \rangle$ サービスを一对一通信サービスとし、 $n(\geq 1)$ 個の $\langle N-1 \rangle$ SAP で受信されるとき、放送通信サービスとする。 p を放送するために、 E_k が何個の PDU を送信するかは、 $\langle N-1 \rangle$ サービスに依存している。放送通信サービスが提供されていれば、 E_k は一つの PDU を送信すればよい。一对一通信サービスであれば、 E_k は n 個の PDU を送信しなければならない。 S_k において送信された各 PDU p が、 p . DST 内の全実体で重複せず、壊れず、送信順に受信されるとき、このサービスを信頼できるものとする。信頼できないサービスを低信頼であるとする。TCP²⁾ のような従来のコネクション指向のサービスは、一对の実体間に信頼通信サービスを提供している。

従来のプロトコルでは、各実体は PDU p が正しく受信されたことを、 p が到着し、かつ p がある条件を満たすこと、例えば通番のチェック等によって決定する。しかし、放送通信サービスでは、宛先が複数あるために、ある実体が、送信された PDU を受信できない場合を考える必要がある。 $n(\geq 2)$ 個の実体 $E_1, \dots, E_n$ の間で、正しい受信を決定するための一つの方法は、ある一つの主制御実体がこの決定を行う集中型制御である。もう一つは、各実体が互いに通信し合うことによって、各実体自身で正しい受信の決定を行う分散型制御である。各実体は、放送通信を用いることで、1 回の送信により他の全実体に PDU を送信できるので、全体状態の把握が容易となり、分散型の制御が適

したものとなる。分散型の制御を用いることにより、主制御実体の障害または、これからの PDU の障害に対して、他の実体が待ち続けるというブロック問題¹⁶⁾を防ぐことが容易になる。分散型制御での問題は、各実体がどのように全実体の状態を知るかである。

【定義】 各 $\langle N \rangle$ 実体 E_k は、以下の条件をすべて満たすとき、 $\langle N \rangle$ PDU p を正しく受信したとする ($k=1, \dots, n$)。

- (1) E_k は p を受信する。
 - (2) E_k は「 p の宛先内の全実体が p を受信した」ことを知る。
 - (3) E_k は「 p の宛先の各実体は、『全宛先実体が p を受信した』ことを知っている」ことを知る。□
- (1), (2), (3) の条件を満足するとき、それぞれ p は E_k で受信された、前確認された、確認されたとする。 p が前確認されたとき、ある実体 E_j は、 $E_k (k \neq j)$ が既に p を受信しているにもかかわらず、 E_k がまだ p を受信していないと考えるかもしれない。これは、 E_j が E_k から p の受信確認を受信できなかったときに起こる。そこで (3) が必要となる。一对一通信サービスでは、 p が前確認されれば、 p が確認されたと考える。つまり、一方の実体が p を送信して、他方の実体が p の確認通知を送信する。これは、確認通知が障害に遭わないという仮定に基づいている。しかしながら、放送通信サービスでは複数の実体を含んでいるので、ある実体が確認通知を受信できない場合を考慮する必要がある。(3) は、各実体 E_k が、他の実体の受信について誤解することを防止している。

ここで、以下の仮定を設ける。

【仮定】 ある実体が放送した PDU は、その実体自身も受信する。□

信頼放送通信サービスを用いると、ある実体によって放送された PDU を、全宛先実体は同一の順序で受信する。しかし、複数の実体が PDU を放送するとき、各実体は同一の順序でこれを受信するとは限らない。そこで、各実体が同一の PDU を同一の順序で受信できる全順序放送通信サービスを定義する。

【定義】 各 $\langle N \rangle$ PDU p と q がそれらの共通の宛先の $\langle N-1 \rangle$ SAP で同一の順序で受信されるとき、かつそのときに限り、この $\langle N-1 \rangle$ 信頼放送通信サービスを全順序放送通信 (TO) サービスとする (図 1)。

□

二つの SAP 間の従来のコネクションの概念を $n(\geq 2)$ 個の SAP 間に拡張する。

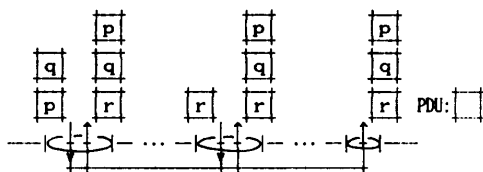


図 1 全順序放送通信 (TO) サービス
Fig. 1 Totally ordering broadcast (TO) service.

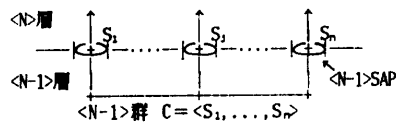


図 2 $\langle N-1 \rangle$ 群
Fig. 2 $\langle N-1 \rangle$ cluster.

【定義】 $\langle N-1 \rangle$ 群 C とは, $n(\geq 2)$ 個の $\langle N-1 \rangle$ SAP $S_1, \dots, S_n$ の組 $\langle S_1, \dots, S_n \rangle$ である (図 2). □
図 2 に示すように, 各実体 E_j は, SAP S_j よりサービスを受けるとき, 実体集合 $\{E_1, \dots, E_n\}$ は群 C を構成するとする.

3. 1 チャンネル (1C) サービス

Ethernet の MAC 層および無線網が提供する通信サービスを論理的に整理する.

【定義】 1 チャンネル (1C) サービスとは, PDU の紛失の可能性はあるが, PDU は全実体で同一の順序で受信される低信頼放送通信サービスである (図 3). □

LAN の MAC 層と無線サービスは 1C サービスを提供している. 1C サービスでは, 各実体は PDU を同一の順序で受信できるが, ある PDU を受信できない場合もある.

L を全順序集合 $(S, \rightarrow)$ とする. つまり, 集合 S 内の要素は, 順序関係 $\rightarrow \subseteq S^2$ により全順序づけられている. S 内のすべての要素 q について, それぞれ $p \rightarrow q$, $q \rightarrow q$ である要素 p と q を各々 L の先頭, 最後尾とし, $\text{top}(L)$ と $\text{last}(L)$ と書く. L 内の要素は, $\text{top}(L)$ から $\text{last}(L)$ まで, $1, 2, \dots, m$ と番号が付いており, m は L の基数である. $L[j]$ を L 内の j 番目の要素

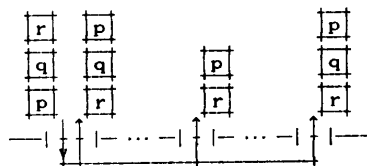


図 3 1 チャンネル (1C) サービス
Fig. 3 One-channel (1C) service.

とし, j をこの索引とする. L^j を次のように定義する. すなわち $L^1 = L[1]$, $L^j = L^{j-1} | L[j] (j \geq 2)$ とする. ここで, $|$ は列の連結を示す. $j \leq k$ のとき, $L[j] \rightarrow L[k]$ であり, $p = L[j]$ で $q = L[j+1]$ のとき, $p \Rightarrow q$ と書く. また, m 個の要素から成る全順序集合 L を, $\langle a_1, \dots, a_m \rangle$ と書くことにし, $a_j = L[j] (j=1, \dots, m)$ で, $a_1 = \text{top}(L)$, $a_m = \text{last}(L)$ である. 例えば, $L^3 = \langle a_1, a_2, a_3 \rangle$ である.

ここで, 送受信される PDU をログによりモデル化する. n 個の実体 $E_1, \dots, E_n$ を考える. また記号 p, q は PDU を示すとする. 各実体 E_k に対して, 送信ログ $SL_k = (SP_k, \rightarrow S_k)$ と, 受信ログ $RL_k = (RP_k, \rightarrow R_k)$ を, それぞれ E_k が放送と受信した PDU の全順序集合とする. ここで SP_k と RP_k は, それぞれ E_k が放送と受信した PDU の集合である. E_k が PDU q 以前に p を受信, 放送したとき, 各々 RL_k 内で $p \rightarrow R_k q$, SL_k 内で $p \rightarrow S_k q$ である.

【定義】 RL_k 内のすべての PDU p と q に対して, ある SL_j 内で $p \Rightarrow S_j q$ であるとき, $p \rightarrow R_k q$ ならば, RL_k は正しいとする. □

つまり, 各実体 E_k は, ある実体 E_j が放送した順に PDU を受信する場合である.

【定義】 受信ログ $RL_1, \dots, RL_n$ に対して, 以下の条件が成り立つとき, PDU 索引 f を障害点とする.

(1) すべての E_j と E_k について, $RL_j^{f-1} = RL_k^{f-1}$ で, RL_j^{f-1} は正しい.

(2) ある E_j について, RL_j^f は正しくない. □

1C サービスを用いた場合, ある実体 E_k がある PDU p を受信できない場合がある. このとき, RL_k は他の $RL_j (j \neq k)$ とは異なり, 障害点が存在する. 例として, 実体 E_1, E_2, E_3 の受信ログ RL_1, RL_2, RL_3 と送信ログ SL_1, SL_2, SL_3 を図 4 に示す. 例で, E_2 は三つの PDU c, h, i をこの順に放送している. ここで, E_2 と E_3 はそれぞれ PDU g と i を受信していない. よって, $RL_1^6 = RL_2^6 = RL_3^6$ で, $RL_1^7 \neq RL_2^7$ であり, RL_2^7 は正しくないで, 障害点は 7 である.

図 4 と同じ送信ログについて, 図 5 の受信ログを考える. E_1 が PDU e を放送したにもかかわらず, E_1 を含むどの実体も e を受信していない. g を受信したとき, $b \Rightarrow S_1 g$ が満たされないで, $RL_j^6 (j=1, 2, 3)$ は正しくない. よって, 障害点は 6 である.

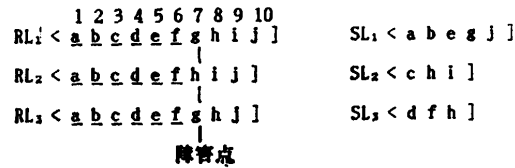


図 4 障害点 (a)

Fig. 4 Failure point (a).

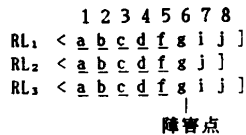


図 5 障害点 (b)

Fig. 5 Failure point (b).

4. 1チャネル (1C) サービス上での全順序 放送通信 (TO) プロトコル

n 個の実体 $E_1, \dots, E_n$ に対して, 1C サービスを用いて全順序放送通信 (TO) サービスを提供する分散型のプロトコルについて述べる. 以下, 記号 p, q, r, s, t は PDU を示すとする.

4.1 変数

各実体 E_k が放送する PDU p は, HDLC¹⁷⁾ や TCP²⁰⁾ のように, E_k が既に受信した PDU の確認通知等の以下の情報を含む.

- $p.SRC = p$ を放送する実体, つまり E_k .
- $p.SEQ = p$ の通番.
- $p.A_j = E_k$ が次に E_j から受信予定の PDU の通番 ($j=1, \dots, n$).
- $p.BUF = E_k$ 内で利用可能なバッファ数.

各実体 E_k から放送される PDU には一意な通番が与えられる. E_k で p の放送後に q が放送されれば ($p \rightarrow s_k q$), $p.SEQ < q.SEQ$ である. また, $p \Rightarrow s_k q$ ならば, $q.SEQ = p.SEQ + 1$ とする. $p.A_j$ は, 「 $q.SEQ < p.A_j$ なる PDU p を, E_k が既に E_j から受信している」ことを意味している.

各 E_k は, 変数 SEQ と REQ_j と, PDU の通番を検査するための行列 $AL_{kj} (k, j=1, \dots, n)$ を持つ.

$SEQ = E_k$ が次に放送する予定の PDU の通番.

$REQ_j = E_k$ が E_j から次に受信予定の PDU の通番 ($j=1, \dots, n$).

$AL_{kj} = E_j$ が E_k から次に受信予定の, E_k が知っている PDU の通番 ($k, j=1, \dots, n$).

$\min AL_j$ を, $AL_{j1}, \dots, AL_{jn}$ 中の最小値とする. これは, 全実体が $g.SEQ < \min AL_j$ なる PDU g

を, E_j から既に受信していることを意味している. ISS_j を, 各実体 E_j の初期通番とする. 最初は, $SEQ = ISS_k$ で, $REQ_j = AL_{jk} = ISS_j$ ($k, j=1, \dots, n$) である.

フロー制御を行うために, E_k が放送する各 PDU p は, E_k 内の空バッファ数を他の実体に通知するための情報 $p.BUF$ を持つ. 各実体 E_k は変数 $F_1, \dots, F_n$ を持つ. F_j は, E_k が知っている E_j 内の空バッファ数である. $\min F$ を $F_1, \dots, F_n$ 中の最小値とする.

以上の変数について, 以下の制約が成り立つ.

【制約】(1) $\min AL_k \leq REQ_k \leq SEQ$,

(2) $AL_{kj} \leq REQ_k$ ($j=1, \dots, n$). □

4.2 送受信

PDU の送受信の手続きについて考える.

4.2.1 受信

各実体 E_k は, E_j から PDU p が到着し, p が以下の条件を満たせば, p を受信する ($k, j=1, \dots, n$).

【受信条件】(1) $p.SEQ = REQ_j$.

(2) $AL_{kj} \leq p.A_k \leq REQ_k$ ($k=1, \dots, n$). □

(1) は, E_k が E_j から p を受信予定であることを示す. (2) は, E_j が既に受信した PDU に対する確認通知を示す. p が受信されたとき, E_k は以下の動作を行う.

【受信動作】(1) $REQ_j := REQ_j + 1$, $AL_{kj} := p.A_k$ ($k=1, \dots, n$).

(2) $F_j := p.BUF$.

(3) p を RL_k の最後尾に追加し, 受信印を付ける. □

ここで, RL_k 内で受信印の付いた PDU から成る部分系列を RPL_k とする.

4.2.2 送信

p を, E_k が送信しようとする PDU とする. このとき, 以下のフロー条件が充足されれば, E_k は p を放送する. ここで, $C(\geq 1)$ と W は正定数である.

【フロー条件】 $\min AL_k \leq SEQ < \min AL_k + \min(W, \min F / (C * n^2))$. □

【送信動作】(1) $p.A_j := REQ_j$ ($j=1, \dots, n$).

(2) $p.SEQ := SEQ$, $SEQ := SEQ + 1$.

(3) $p.BUF :=$ 現在利用可能なバッファ数.

(4) p を SL_k の最後尾に追加し, p を放送する. □

フロー条件が充足されなければ, E_k は放送を行わない. 各 E_k では少なくとも $C * n^2$ 個の PDU を記憶できるだけのバッファが必要である. 空バッファ数

が $C \cdot n^2$ 以下ならば、 E_k は「受信できない」ことを示す RNR (Receive Not Ready) PDU を放送する。 E_k から RNR を受信したら、各 E_j は E_k から「受信可能である」ことを示す RR (Receive Ready) PDU を受信するまで、放送を停止する。十分なバッファを得られれば、 E_k は RR を放送する。RR も RNR も、通常の PDU と同様に確認通知 $A_j (j=1, \dots, n)$ を含む。

4.3 前 確 認

【定義】 各 E_k の受信ログ RL_k 内の PDU p (ただし、 $p.SRC=E_j$) と q ($q.SRC=E_k$) に対して、 $p \rightarrow_{RL_k} q$, $p.SEQ < q.SEQ$ ならば、 q を E_k について p を前確認する PDU とする。□

つまり、 q を受信したとき、 E_k は p が E_k で受信されていることがわかる。

【前確認条件】 $p.SEQ < \min AL_j (p.SRC=E_j)$. □

【補題 4.1】 E_k が E_j から受信した PDU p が前確認条件を充足すれば、 p は E_k で前確認される。

【証明】 E_k は、 $p.SEQ < \min AL_j$ なる PDU p (ただし、 $p.SRC=E_j$) を各実体が既に受信していることを知っている。よって、 p は E_k で前確認される。■

【補題 4.2】 ある実体 E_k と受信ログ RL_k について、部分ログ RL_k^A が正しく、 $p = \text{last}(RL_k^A)$ が前確認条件を満たすならば、 $g \rightarrow_{RL_k} p$ である各 PDU g も前確認条件を満たす。

【証明】 g を、 RL_k 内の $g \rightarrow_{RL_k} p$ である任意の PDU とする (図 6)。 p は前確認されているとする。 p を前確認する各 PDU は g も前確認する。なぜなら、 g は各実体で p 以前に受信されているからである。■

よって、 RPL_k 内の PDU p が前確認条件を満たせば、 $g \rightarrow_{RL_k} p$ なる PDU g も前確認条件を満たす。前確認条件を満たす PDU p に、前確認印が与えられ

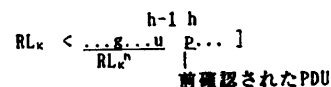


図 6 ログ内の PDU
Fig. 6 PDU log.

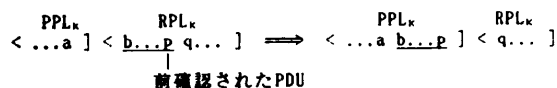


図 7 前確認動作
Fig. 7 Pre-acknowledgment action.

る。 RL_k 内で、前確認印の付いた PDU の部分系列を PPL_k とする。

【前確認動作】 RPL_k 内の先頭から p までの受信印の付いた PDU を RPL_k から取り出し、前確認印を付けて PPL_k の最後尾に追加する (図 7)。□

4.4 確 認

次に、PDU をどのように確認するかを考える。

【定義】 各 RL_k 内の PDU p と q に対して、 q が E_k について p を前確認し、 E_k について p を前確認する $g \rightarrow_{RL_k} q$ なる PDU g が存在しないとき、 q は E_k について p を最初に前確認するとする。また、 $q (=q.SRC)$ が E_k について最初に p を前確認する PDU で、 $q \rightarrow_{RL_k} g$ なる p を最初に前確認する g が存在しないとき、 q は p を最後に前確認するとする。□

【確認条件 1 (ACK 1)] PPL_k 内の PDU p に対して、 E_k で p を最初に前確認するすべての PDU が前確認される。□

【補題 4.3】 PPL_k 内の PDU p が確認条件 1 を充足するならば、 p は E_k で確認される。

【証明】 p を前確認する PDU q の受信によって、 $E_j (=q.SRC)$ が p を既に受信していることがわかる。 q の前確認の意味は、各実体が、 E_j は p を受信したことを知っていることである。よって、 p を最初に前確認する全 PDU が前確認されれば、 E_k は、各実体が他の全実体が p を受信したことを知っていることがわかる。■

1C サービスでは、PDU は各実体で同一の順序で受信される。よって、 g が E_k で p を最後に前確認する PDU であるとき、 g が E_j で受信されていれば、 g は E_j においても p を最後に前確認する PDU である。よって、確認条件は以下のように単純化できる。

【確認条件 2 (ACK 2)] PPL_k 内の PDU p に対して、 p を最後に前確認する PDU が前確認される。□

【補題 4.4】 1C サービスが用いられ、かつ PPL_k 内の PDU p が確認条件 2 を充足するならば、 p は確認される。■

PDU p が確認条件を満たすならば、 E_k は p に確認印を与える。 RL_k 内で確認印の付いた PDU の部分系列を APL_k とする。

【確認動作】 PPL_k 内の先頭から p までの PDU を PPL_k から取り出し、確認印を与え APL_k の最後尾に追加する (図 8)。□

E_k は p を受信していないが, E_j は p と q をこの順序で受信している. E_k は p を受信していないので, p を前確認する PDU を放送しない. 仮定より, E_j で p は APL_j , または PPL_j 内にある. 補題 4.1 より, E_j は全実体から p を前確認する PDU を受信している. これは矛盾である. ■

この定理は, ある実体で受信されていない PDU は, どの実体でも前確認されないことを示している.

【定理 5.2】 各実体 E_j と E_k に対して, APL_j は $APL_k | PPL_k$ の前置である.

【証明】 FP が存在しない場合は, 明らかである. 定理 5.1 より, FP は APL_k にも PPL_k にも含まれない. p は E_j で APL_j 内にあり, p はある E_k では前確認されていないとする. q を, p を最後に前確認する PDU とする. E_j で p が確認されるためには, q が前確認されなければならない. q が, E_j で既に前確認されているということは, q は全実体で前確認されているか, または受信されていることである. 仮定より, p は E_k で前確認されていない. これは, $p \rightarrow_{R_j} r \rightarrow_{R_j} q$ なる PDU r が存在し, E_k は r を受信していないことを意味する. 補題 4.2 より, E_j で q が前確認されているので, r も E_j で前確認されている. しかし, 定理 5.1 より, FP, つまり r が各 E_l の RPL_l 内に存在する. これは, 矛盾する. ■

この定理は, ある実体で各 PDU p が確認されるときは, p は各実体で確認されるか, または前確認されることを示している.

【定理 5.3】 障害点 FP の発見後, 再設定手続きを適用するならば, 各実体 E_k は受信ログとして RL_k^{FP-1} を持つ.

【証明】 E_k について, $RL_k[FP]=q$ で, E_j について, $RL_j[FP](=p) \rightarrow_{R_j} q$ であるとする. つまり, E_k は p を受信していない.

(1) E_k は, $p(p.SRC=E_k)$ の粉失を, タイムアウト, または FP 条件によって発見する. E_k は, $r.A_i=p.SEQ$ なる RST r を放送する. r を受信したとき, E_j は, $q(q.SRC=E_k)$ を含む $p \rightarrow_{R_j} g$ なる PDU g と p を廃棄する. E_j は, $s.A_i=s.SEQ$ なる RST-
PACK s を放送する. s を受信したとき, E_k は q を廃棄する. ここで, E_j と E_k は同一の受信ログ RL_j^{FP-1} と RL_k^{FP-1} を持つ.

(2) p を前確認する PDU を, E_k からある時間内に受信しないことにより, E_j が, E_k は p を受信していないことを発見した場合を考える. E_j は, RPL_j

内の $p \rightarrow_{R_j} g$ なる PDU g と p を廃棄し, $r.A_i=p.SEQ$, $r.A_i=q.SEQ$ なる RST r を放送する. r の受信によって, E_k は RPL_k 内の $q \rightarrow_{R_k} g$ なる PDU g と q を廃棄する. ここで, E_j と E_k は同一の受信ログ RL_j^{FP-1} と RL_k^{FP-1} を持つ.

(1) と (2) より, 再設定手続きの適用後, 各実体 E_j は同一の受信ログ RL_j^{FP-1} を持つ. ■

【定理 5.4】 1C サービス上での TO プロトコルは, 群内の各実体に全順序放送通信 (TO) サービスを提供する.

【証明】 PDU の紛失がないときは明らかである. PDU の紛失がある場合を考える. 定理 5.2 から, 障害点より後に受信された PDU は前確認されない. また, 各実体は FP 条件, またはタイムアウトにより, FP を発見できる. 定理 5.3 より, 再設定手続きの実行後, 各実体は同一の正しいログを持つ. ゆえに, 定理は成り立つ. ■

5.2 性能

本プロトコルの性能を, PDU 数と時間により評価する. 前者では, ある PDU p を確認するために SAP で送信される PDU 数を考える. SAP から宛先までの最大遅延時間をウランドとし, 後者についてはラウンド数を考える.

まず, PDU 数について考える. 最良の場合, 各実体は, PDU p を最後に前確認する PDU g を受信してから, g を前確認する PDU を放送する. このとき, $1+(n-1)+n=2n$ 個の PDU が放送され, RPL と PPL の長さは $O(n)$ である. 最悪の場合, p を前確認する PDU g を受信するごとに, 各実体は g を前確認する PDU を放送する. ここで, $1+(n-1)+(n-1)^2=n^2-n+1$ 個の PDU が放送される. RPL と PPL の長さはそれぞれ $O(n)$ と $O(n^2)$ である.

次に, ラウンド数について考える. 最良の場合, p を前確認する PDU は並行して放送される. ここでラウンド数は 3 である. 一方, Ethernet MAC サービスのように PDU を並行して放送できない場合, つまり, 一つのチャンネル上に, 同時に高々一つの PDU しが存在できない場合, 最良 $1+(n-1)+n=2n$ ラウンド, 最悪 $1+(n-1)+(n-1)^2=n^2-n+1$ ラウンド必要である. PDU 長は, 実体数だけの確認 A_i を持つので, $O(n)$ である.

6. おわりに

本論文では, 放送通信における信頼性について定義

し, Ethernet のような低信頼放送通信サービスを用いて, 全順序放送通信サービスを提供する TO プロトコルのデータ転送手続きについて述べた. このプロトコルは, 群という概念に基づく分散型制御を用いている. TO プロトコルは, Ethernet の MAC サービスのような 1 チャンネル (1C) サービスを用いて, 群内の各実体が同一の PDU を同一の順序で受信できる TO サービスを提供する. このプロトコルを用いることにより, 分散型データベースシステムでの, コミットメント制御¹⁵⁾, 分散型デッドロック検出, 分散型問合せ処理¹²⁾等を容易に実現できる.

現在, TO プロトコルを Ethernet MAC サービスを用いて, SunOS と UNIX System V 上で C 言語で実現している. Ethernet 上には, 大型機 M380 (Facom), ミニコン A400 (Facom), 3 台の Sun 3 ワークステーションが接続された構成で, TO プロトコルの性能評価中である. 問題はバッファサイズで, 本プロトコルでは従来の一対一通信に比べて, 最悪の場合 n^2 倍以上のバッファが必要になる. しかし, 今日のハードウェア技術により記憶装置のコストは非常に低くなっているので, この問題は数 M バイトの記憶装置を用いることで解決できると考える. 実現結果とその評価については, 別の論文で報告したい. 現在, 群を構成する実体に変化できる群, 一般の低信頼放送サービス上の信頼性, 安全性のある放送通信プロトコル¹⁹⁾等の検討を行っている.

参 考 文 献

- 1) ISO: Data Processing—Open Systems Interconnection—Basic Reference Model, ISO 7498 (1987).
- 2) Defense Communications Agency: DDN Protocol Handbook, Vol. 1-3, NIC 50004-50005 (1985).
- 3) ISO: Message Oriented Text Interchange System, ISO 8883 (1987).
- 4) ISO: Remote Database Access, ISO/JTC 1/SC 21 (1989).
- 5) Takizawa, M.: *Distributed Database System—JDDBS, JARECT*, North-Holland and Ohmsha (1985).
- 6) Gray, J.: Notes on Data Base Operating Systems, *Operating Systems: An Advanced Course*, Bayer, R. ed, Springer-Verlag (1979).
- 7) Edgar, K.: Deadlock Detection in Distributed Databases, *ACM Comput. Surv.*, Vol. 19, No. 4, pp. 303-328 (1987).
- 8) IEEE Project 802 Local Network Standards—Draft (1982).
- 9) Metcalfe, R. M.: Ethernet: Distributed Packet Switching for Local Computer Networks, *CACM*, Vol. 19, No. 7, pp. 395-404 (1976).
- 10) Chang, J. and Maxemchuck, M. F.: Reliable Broadcast Protocols, *ACM TOCS*, Vol. 2, No. 3, pp. 251-273 (1984).
- 11) Schneider, F., Gries, D. and Schlichting, R. D.: Fault-Tolerant Broadcasts, *Sci. Comput. Programming*, Vol. 4, pp. 1-15 (1984).
- 12) Takizawa, M.: Design of Highly Reliable Broadcast Communication Protocol, *Proc. of IEEE COMPSAC 87*, pp. 731-740 (1987).
- 13) Takizawa, M.: Cluster Control Protocol for Highly Reliable Broadcast Communication, *Proc. of the IFIP Conf. on Distributed Processing*, Amsterdam, pp. 431-445 (1987).
- 14) Bracha, G. and Toueg, S.: Asynchronous Consensus and Broadcast Protocols, *JACM*, Vol. 32, No. 4, pp. 824-840 (1985).
- 15) Takizawa, M. and Shin, S.: Commitment Control by Using Reliable Broadcast Communication, to appear in *Proc. of the 2nd Scandinavia-Japanese Seminar* (1989).
- 16) Skeen, D.: Nonblocking Commitment Protocols, *ACM SIGMOD*, pp. 133-147 (1982).
- 17) ISO: Consolidation of HDLC Elements of Procedures, ISO 4335 (1978).
- 18) 中村章人, 滝沢 誠: 多チャンネル上での全順序放送通信プロトコルについて, 情報処理学会マルチメディア通信と分散処理研究会資料, 39-1, pp. 1-8 (1989).
- 19) 滝沢 誠: 安全放送通信, 情報処理学会マルチメディア通信と分散処理研究会資料, 42-6, pp. 39-46 (1989).
- 20) Lamport, L., Shostak, R. and Pease, M.: The Byzantine Generals Problem, *ACM TOPLAS*, Vol. 4, pp. 382-401 (1982).

(平成元年 5 月 29 日受付)

(平成 2 年 2 月 13 日採録)

**滝沢 誠 (正会員)**

1950年生。1973年東北大学工学部応用物理学科卒業。1975年東北大学大学院工学研究科応用物理学専攻修士課程修了。同年(財)日本情報処理開発協会入社。1986年東京電機大学理工学部経営工学科講師。現在同助教授。1989年9月より1年間西独国立情報処理研究所(GMD-IPSI (F4)) 客員教授。工学博士。分散型データベースシステム、通信網、分散型システム、知識ベースシステム等の研究に従事。電子情報通信学会、人工知能学会、ACM、IEEE 各会員。「知識工学基礎論」オーム社(共著)。

**中村 章人 (学生会員)**

1966年生。1989年東京電機大学理工学部経営工学科卒業。現在、同大学大学院理工学研究科システム工学専攻在学中。分散型データベースシステム、通信プロトコル等に興味を持つ。