多チャネルシステム上の送信順序保存放送通信プロトコル

中村章人† 滝沢誠†

本論文では，Ethernet や無線網等の放送通信網上に，信頼性のある放送通信システムを設計する問題について述べる．各エンティティ（実体）が放送したプロトコル・データ単位（PDU）を，送信順に紛失なく受信できる送信順序保存放送通信（OP）サービスを，Ethernet のような放送通信サービスを利用して実現する方法について論じる．本論文では，放送通信サービスの信頼性を明確にし，放送通信サービスのモデルを示す．また，OP サービスを実現するための，分散型制御に基づく，データ転送手順を提案する．

Order-Preserving Broadcast Protocol on Multi-Channel System

Akihito Nakamura† and Makoto Takizawa†

This paper presents a design of a reliable broadcast communication system on less reliable broadcast networks like Ethernet and radio networks. By using the less reliable broadcast service, we design and implement a reliable broadcast communication service, i.e. an order-preserving broadcast (OP) service where every entity receives all the protocol data units (PDUs), i.e. messages broadcast by each entity, in the same order as they were sent. In order to provide reliable broadcast communication for multiple entities on the less reliable broadcast service, we propose a data transmission procedure, whose execution is coordinated by all entities in a distributed scheme. In this paper, we also define reliability of broadcast communication and model of broadcast communication services. The protocol can be useful in designing and implementing distributed systems like groupware systems.

1. はじめに

現在の情報システムは，複数の計算機を通信網により結合した，分散型の形態をとっている．グループウェア・システム[3),9)]等の分散型応用システムを実現するためには，複数の計算機間での協調動作が必要となる．協調動作は，各計算機上の応用実体（OSI のエンティティ[16)]）によって行われ，下位層の通信システムを利用してデータの送受信を行う．特に，各応用実体は，協調動作を行っている複数の応用実体にデータを転送することが必要となる．通信システムにより，複数の応用実体用に，信頼性のある放送通信サービスが提供されているならば，これら分散型応用システムの実現と運用が，容易かつ効率的に行える．

通信プロトコルを設計する上で，プロトコル・データ単位（PDU）が宛先実体で正しく受信されているかどうかを，どう判断するかという問題が重要であり，集中型と分散型の制御方式がある．集中型制御方式では，ある一つの実体（指揮者）が，複数の実体（関係者）での PDU の正しい受信の判断を行う[2),4),5),7),10),17)]．この方式では，各関係者は指揮者からの判断を待たねばならず，指揮者の障害が全体障害を引き起こす場合がある．一方，分散型制御方式では，指揮者と関係者の区別がなく，各実体は受信した PDU に基づいて正しい受信の判断を行う．本論文で提案する放送通信プロトコルは，分散型制御に基づいている．ISIS システム[1)]も，分散型の制御方式を用いているが，各々の PDU については，送信者が正しい受信の判断を行う．本論文で提案するプロトコルは，すべての PDU について，各実体が正しい受信の判断を行うものである．

本論文では，信頼性のある放送通信サービスを，以下の二つの性質を満足するサービスと考える．つまり，放送された各 PDU は，必ず全宛先実体に届くことと，各実体から受信する PDU の順序が，送信順序を保存していることである．放送通信の信頼性の論理的性質については，文献 23) で議論されている．これまでに，全実体で同一の順序で PDU を受信する全順序放送通信（TO）プロトコル[21),23)]を示し，さらに，

† 東京電機大学理工学部経営工学科
Department of Computers and Systems Engineering, Faculty of Science and Engineering, Tokyo Denki University

各 PDU が全実体ではなく，その一部に放送される選択的半順序放送通信（SPO）プロトコル[12)~14)]を示している．本論文では，Ethernet[11)] や無線網等の，PDU の紛失が発生する低信頼な放送通信サービスを用いて，全順序性を保障しないが，各実体からの PDU の送信順序を保持する，分散型制御の送信順序保存放送通信（OP）プロトコルの設計について述べる．

本論文の構成は以下のようである．2章では，放送通信における PDU の正しい受信の概念を明確にし，放送通信サービスのモデルを示す．3章では，OP サービスを提供するための OP プロトコルのデータ転送手続きについて述べる．4章では，OP プロトコルの評価について述べる．

2. サービスモデル

本章では，放送通信における PDU の正しい受信の概念を明確にし，放送通信サービスのモデルを示す．

2.1 正しい受信

群 C を，通信システム内の n 個の実体集合 $\{E_1, \cdots, E_n\}$ とする．各 E_j が送信する PDU には，E_j がそれまでに群内の各実体から受信した PDU に対する受信通知（ACK）が含まれるとする．ある実体 E_i が C 内の全実体宛に放送した PDU p が，C 内の各実体で正しく受信されたかどうかを，各 E_k が判断する基準として，以下の(1)～(3)の三段階が考えられる．

(1) **受理**：E_k が，E_i から p を受信したとき，E_i が p 以前に送信した全 PDU を E_k が既に受信しているならば，p は E_k で受理される．

(2) **前確認**：E_k が，C 内の各 E_j から p の受信通知を含む PDU を受理する．このとき，p は E_k で前確認されたという．

p の受信通知を含む PDU を，p を前確認する PDU とする．E_k で p が前確認されていても，他のある E_h は，p がある E_j で受理されていることを知らない場合がある．E_j からの p に対する受信通知を含んだ PDU が紛失した場合に，このような状況が発生する．このため，前確認の段階で，p がすべての実体で受理されているかについての決定を行うと，実体間で異なる決定を行う可能性がある．よって，分散型制御下で，各実体が受信の原子性について同一の決定を行うためには，「C 内の全実体が p を前確認している」ことを知る必要がある．

(3) **確認**：E_k が，C 内の各 E_j から，p を前確認する PDU の受信通知を含む PDU を受理する．このとき，p は E_k で確認されたという．

各実体は，受理した PDU p が確認されたとき，p は群内の全実体で正しく受信されたと判断する．

前確認の導入により，各実体が群内での各 PDU の正しい受信の判断を行う分散型の制御を実現できる．これにより，ある一つの実体（制御実体）が判断を行う集中型制御での，判断結果を待つ問題と，制御実体の障害による全体障害，制御実体からの PDU の紛失問題を解決できる．

[正しい受信の例] 群 $C=\{E_1, E_2, E_3\}$ について，E_1 が放送した PDU p が C 内で正しく受信される例を**図1**に示す．

(1) 受理：各実体は，p を受理する．この後，各実体 E_1, E_2, E_3 は，p に対する受信通知を含む（p を前確認する）PDU a, b, c をそれぞれ放送する．

(2) 前確認：PDU a, b, c を受信した各実体は，p が C 内の全実体で受理されていることを知る．つまり，p は各実体で前確認される．その後，各実体は，a, b, c に対する受信通知を含む PDU d, e, f を放送する．しかし，例えば実体 E_3 が，E_2 からの PDU b を受信できなかったとすると，E_3 は，E_2 が p を受理していることを知らない．また，各実体は，a, b, c を受信した場合でも，他の実体が同様に，全実体で p が受理されたことを知っているかどうかはわからない．

(3) 確認：各実体は，PDU d, e, f を受信することにより，C 内の全実体が p を受理していることが明らかになる．この時点で，a, b, c は，d, e, f に

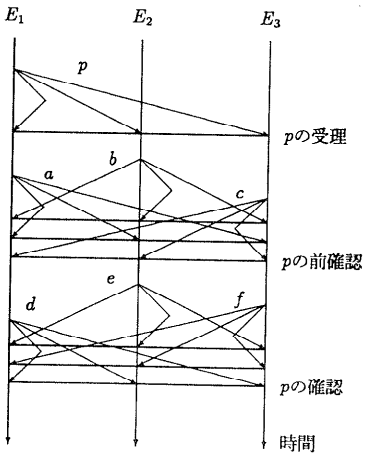


図 1 正しい受信の例
Fig. 1 An example of the correct reccipt.

より前確認される．本論文では，プロトコルの効率を向上させるために，各受信通知を送信するための特別な PDU を用意せず，次のデータ送信用の PDU に付与（ピギーバック）する．□

2.2 放送通信サービスの性質

各実体が利用するサービスを，PDU の系列であるログの集合[21)~23)]としてモデル化する．ここで，m 個の PDU から成るログ L を $<p_1 p_2 \cdots p_{m-1} p_m]$ と書き，$top(L)$ は先頭の p_1，$last(L)$ は最後尾の p_m を示すとする．

群内の各実体 E_i は，送信および受信した PDU の履歴である送信ログ SL_i と受信ログ RL_i を持つ $(i=1, \cdots, n)$．E_i が，p の送信後に q を送信したならば，$p \rightarrow_{SL_i} q$ と書き，p の受信後に q を受信したならば，$p \rightarrow_{RL_i} q$ と書く．副受信ログ RL_{ij} を，RL_i の部分系列で，E_i が E_j から受信した PDU から成るとする．各 RL_i について，p が RL_{ij} 内に含まれるならば，p は必ず SL_j 内に存在するものとする．すなわち，受信された PDU は，ある実体で送信されたものである．

任意の二つの受信ログ RL_i と RL_j の間の関係を，以下に定義する．

[定義]

（1） RL_i と RL_j の両方に含まれる任意の二つの PDU p と q について，$p \rightarrow_{RL_i} q$ ならば $p \rightarrow_{RL_j} q$ であるとき，RL_i と RL_j は順序同値である．

（2） RL_i と RL_j が，同一の PDU を含むならば，RL_i と RL_j は情報同値である．

（3） RL_i と RL_j が，順序同値でかつ情報同値であるとき，RL_i と RL_j は同値である．□

RL_i と RL_j が順序同値である場合，E_i と E_j が共に受信した PDU の受信順序は同一である．しかし，各 PDU が E_i と E_j の両方で受信されているとは限らない．RL_i と RL_j が情報同値である場合，E_i と E_j は同一の PDU を受信しているが，その順序は同一とは限らない．RL_i と RL_j が同値である場合，E_i と E_j は同一の PDU を同一の順序で受信している．

次に，送信ログに対する受信ログ RL_i の性質を定義する．

[定義]

（1） E_i が，各 E_j から受信した任意の二つの PDU p と q について，$p \rightarrow_{SL_j} q$ ならば，$p \rightarrow_{RL_i} q$ であるとき，RL_i は順序保存である．

（2） RL_i と，$SL_1, \cdots, SL_n$ に含まれる PDU が同じとき，RL_i は情報保存である．□

RL_i が順序保存ならば，E_i は，各 E_j ごとに送信順に PDU を受信している．RL_i が情報保存であるとき，E_i は群内の各実体が送信した全 PDU を受信している．以上から，RL_i については以下の性質を定義する．

[定義]

RL_i が順序保存かつ情報保存であるならば，RL_i は正しい．□

つまり，E_i は，各実体が送信した全 PDU を，送信順に受信している．放送通信サービスを提供する群内の全受信ログが正しいならば，このサービスを，信頼放送通信サービスとする．

2.3 放送通信サービスの種類

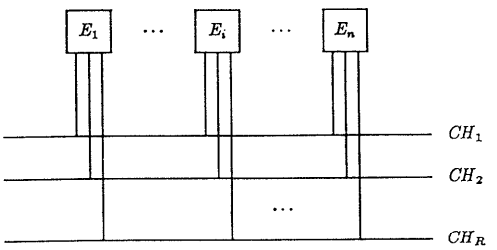
Ethernet や FDDI 等の通信システムは，高い信頼性を提供している．しかし，通信システムが高速化，高信頼化しても，これを利用するプロセスといった実体が，届けられた PDU をすべて受信できない場合がある．バッファの不足，オーバラン等がこの原因である．したがって，実体から見たときの下位層の通信サービスの障害は，PDU の紛失である．ここでは，実体の障害としては，停止だけを考える．

[定義]

（1） 1チャネル (1C) サービスとは，各受信ログが順序保存で，かつ互いに順序同値であるサービスである．

（2） 多チャネル (MC) サービスとは，各受信ログが順序保存であるサービスである．□

1C サービスを利用した場合，各実体は PDU を同一の順序で受信できる．MC サービスでは，各実体ごとに PDU を送信順に受信できる．しかし，これらのサービスでは，PDU が紛失する場合があり，各受信ログは情報保存とは限らない．1C サービスは，Ethernet の MAC 副層[6),11)]のサービスをモデル化し



E_i：実体 $(i=1, \ldots, n)$，CH_j：チャネル $(j=1, \ldots, R)$

図 2 多チャネル (MC) システム
Fig. 2 The Multi-Channel (MC) system.

たものである．MC サービスは，複数の Ethernet で計算機が接続されたシステム［**図 2**］で提供され，各実体は，ある Ethernet を用いて PDU の送信を行うシステムをモデル化したものである．

［**例 2.1**］ 群 C は，三つの実体 E_1，E_2，E_3 から構成されているとする．群 C が，1C サービスと，MC サービスを提供する場合の例を示す．

（1） **図 3** に，C が 1C サービスを提供している場合の，各実体のログを示す．1C サービスでは，各受信ログは順序保存で，かつ互いに順序同値である．しかし，情報保存性が保障されていないので，全 PDU を受信できるとは限らない．例えば，E_2 は c を，E_3 は q を受信していない．

（2） **図 4** に，MC サービスの例を示す．MC サービスでは，各実体は，各々の実体から PDU を送信順に受信できる．しかし，（1）と同様に，PDU の紛失が発生する場合がある． □

信頼放送通信サービスを，以下の二種類に分ける．

［定義］

（1） 送信順序保存放送通信（OP）サービスとは，各受信ログが正しいサービスである．

（2） 全順序放送通信（TO）サービスとは，各受信ログが互いに順序同値である OP サービスである． □

OP と TO サービスでは，各々の実体からの PDU の受信順序は，それらの送信順序を保存しており，PDU の紛失がない．さらに，TO サービスでは，異なる実体から受信した PDU の順序も各実体で同一である．

E_1 RL_1: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_1: $<a\,b\,c\,d]$
E_2 RL_2: $<a\,x\,b\,p\,y\,d\,z\,q]$ SL_2: $<p\,q]$
E_3 RL_3: $<a\,x\,b\,c\,p\,y\,d\,z]$ SL_3: $<x\,y\,z]$

図 3 1C サービスの例
Fig. 3 An example of the 1C service.

E_1 RL_1: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_1: $<a\,b\,c\,d]$
RL_{11}: $<a\,b\,c\,d]$
RL_{12}: $<p\,q]$
RL_{13}: $<x\,y\,z]$
E_2 RL_2: $<p\,x\,a\,b\,y\,z\,d\,q]$ SL_2: $<p\,q]$
RL_{21}: $<a\,b\,d]$
RL_{22}: $<p\,q]$
RL_{23}: $<x\,y\,z]$
E_3 RL_3: $<a\,x\,p\,y\,b\,c\,z\,d]$ SL_3: $<x\,y\,z]$
RL_{31}: $<a\,b\,c\,d]$
RL_{32}: $<p]$
RL_{33}: $<x\,y\,z]$

図 4 MC サービスの例
Fig. 4 An example of the MC service.

［**例 2.2**］ 例 2.1 と同じ送信ログを用いて，OP サービスと，TO サービスを提供する群 C の例を示す．

（1） C が OP サービスを提供している場合，各実体は，各々の実体から PDU を送信順に受信できる．例えば，**図 5** で，各実体は E_1 が送信した a，b，c，d をこの順序で受信する．

（2） **図 6** に，C が TO サービスを提供している場合の，各実体のログを示す．TO サービスでは，各受信ログは順序保存で情報保存，かつ互いに順序同値である．つまり，各実体は全 PDU を同一の順序で受信している． □

グループウェアシステム[3),9)]等の新しい分散型応用システムでは，複数の利用者，計算機間でのグループ通信が必要となる．こうしたシステムの実現と運用を行う上で，OP サービスと TO サービスの利用は有効である．データ転送の信頼性と順序を保つことを考慮せずに，応用を設計し運用できるからである．例として，ワークステーション（WS）を利用した，分散型会議システムを考える．一人一台の WS を利用し，各利用者のウィンドウが，他のすべての WS にも表示されているとする．こうしたウィンドウ上に，文書や画像等を表示する．ここで，各利用者の行ったウィンドウに対する変更が，全 WS に反映されねばならない．ウィンドウごとに，その表示内容の変化が全 WS で同一でなければならないとすると，各 WS からウィンドウの変更を，OP または TO サービスを利用して放送できる．この場合，ウィンドウごとの更新の順序だけが重要であり，WS ごとの更新の全順序性は必要ないので，TO サービスよりも OP サービスの

E_1 RL_1: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_1: $<a\,b\,c\,d]$
RL_{11}: $<a\,b\,c\,d]$
RL_{12}: $<p\,q]$
RL_{13}: $<x\,y\,z]$
E_2 RL_2: $<p\,x\,a\,b\,y\,c\,z\,d\,q]$ SL_2: $<p\,q]$
RL_{21}: $<a\,b\,c\,d]$
RL_{22}: $<p\,q]$
RL_{23}: $<x\,y\,z]$
E_3 RL_3: $<a\,x\,p\,y\,b\,c\,z\,d\,q]$ SL_3: $<x\,y\,z]$
RL_{31}: $<a\,b\,c\,d]$
RL_{32}: $<p\,q]$
RL_{33}: $<x\,y\,z]$

図 5 OP サービスの例
Fig. 5 An example of the OP service.

E_1 RL_1: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_1: $<a\,b\,c\,d]$
E_2 RL_2: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_2: $<p\,q]$
E_3 RL_3: $<a\,x\,b\,c\,p\,y\,d\,z\,q]$ SL_3: $<x\,y\,z]$

図 6 TO サービスの例
Fig. 6 An example of the TO service.

方が効率が良い（後述の障害復旧を参照）．しかし，一つのウィンドウを，全 WS で共有して操作するためには，操作の順序が各 WS で同一でなければならない．このような場合には，TO サービスを利用できる．

3. 多チャネル（MC）サービス上の送信順序保存放送通信（OP）プロトコル

本章では，送信順序保存放送通信（OP）プロトコルのデータ転送手続きについて述べる．OP プロトコルは，下位層から提供される MC サービスを利用し，上位層に OP サービスを提供する．

3.1 変　　数

OP プロトコルの PDU 形式を，**図7**に示す．ここで，あるPDU p について，記法 p^i は，p が E_i により送信されたことを示すときに用い，記法 $p^i.F$ は p^i 内の項目 F を示す．$p^i.$SRC は，p^i を送信する実体 E_i である．$p^i.CID$ は，E_i が属する群の識別子である．群識別子は，各群を一意に識別し，群確立手続き[19),20)]によって決定される．p^i は，通番 $p^i.SEQ$ を持つ．$p^i.ACK_j$ は，E_i が E_j から次に受信予定の PDU の通番である（$j=1,\cdots,n$）．これは，E_i が $q^j.SEQ<p^i.ACK_j$ である全 PDU q^j を，E_j から既に受信していることを示す確認通知である．$p^i.BUF$ により，E_i が利用可能なバッファ数が各実体に通知され，各実体はこれに基づいてフロー制御を行う．

各 E_i は，以下の変数を持つ（$j,k=1,\cdots,n$）．

- $SEQ_i=E_i$ が，次に送信予定の PDU の通番．
- $REQ_j=E_i$ が，E_j から次に受信予定の PDU の通番．
- $AL_{kj}=$「E_j が，E_k から次に受信予定である」と E_i が知っている PDU の通番．
- $min\,AL_k=AL_{k1},\cdots,AL_{kn}$ の最小値．
- $PAL_{kj}=$「E_j が，E_k からの PDU で，次に前確認予定である」と E_i が知っている PDU の通番．
- $BUF_j=E_j$ 内の利用可能なバッファ数．
- $min\,BUF=BUF_1,\cdots,BUF_n$ の最小値．

AL_{kj} には，E_i が E_j から受信した PDU p^j の確認通知 $p^j.ACK_k$ が記録される．これにより，E_i は，$q^k.SEQ<AL_{kj}$ である PDU q^k を E_j が受信していることがわかる．$min\,AL_k$ は，群内の各実体が，$g^k.SEQ<min\,AL_k$ である PDU g^k を E_k から既に受信していることを意味する．群内の全実体は，群確立手続き[19),20)]により，各実体 E_i の SEQ_i と BUF_i の初期値 ISS_i と IBF_i についての合意が取れている．群が確立されたとき，各 E_i 内で，$SEQ_i=ISS_i$, $REQ_j=AL_{jk}=ISS_j$, $BUF_j=IBF_j$ である（$j,k=1,\cdots,n$）．

SRC	CID	SEQ	$ACK_1 \ldots ACK_n$	BUF	DATA

図 7　PDU の形式
Fig. 7　PDU format.

E_i $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ E_j

$\qquad ARL_{ij} \qquad PRL_{ij} \qquad RRL_{ij}$

RL_{ij}: $< a_1 \ldots a_x]$ $< p_1 \ldots p_y]$ $< r_1 \ldots r_{z-1}]$ $\leftarrow r_z$ — SL_j: $< \ldots r_z]$

図 8　受信ログ
Fig. 8　Receipt log.

各 E_i は，送信ログ SL_i と副受信ログ RL_{ij} を持つ（$j=1,\cdots,n$）．各副受信ログ RL_{ij} は，**図8**のように三つの副ログ ARL_{ij}, PRL_{ij}, RRL_{ij} に分けられる．それぞれ，確認，前確認，受理された PDU が格納される．

3.2 送 受 信

各 E_i は，応用実体からデータ送信要求を受け，以下のフロー条件を充足するならば，送信手続きに従って PDU を送信する．ここで，W は最大ウィンドウサイズ，H は最大 PDU 長である．各実体は，少なくとも $O(n^2)$ 個の PDU を記憶できるだけのバッファを持たねばならない．これは，最悪の場合，一つの PDU が確認されるまでに，n^2 個の PDU の送受信が必要[21)]であることによる．

[フロー条件] $min\,AL_i \leq SEQ_i < min\,AL_i + min(W, min\,BUF/(H*n^2))$. □

[送信手続き]

（1） $p^i.SEQ := SEQ_i$, $SEQ_i := SEQ_i + 1$.

（2） $p^i.ACK_j := REQ_j$ $(j=1,\cdots,n)$, $p^i.BUF :=$ E_i が現在利用可能なバッファ数．

（3） p^i を SL_i に追加し，p^i を送信する． □

E_i が，p^j を受信したとする．このとき，p^j は以下の受理手続きに従って，受理される．

[p^j の受理条件] $p^j.SEQ=REQ_j$. □

[受理手続き]

（1） $REQ_j := REQ_j + 1$, $AL_{kj} := p^j.ACK_k$ $(k=1,\cdots,n)$, $BUF_j := p^j.BUF$.

（2） p^j を RRL_{ij} に追加する． □

E_i は，E_j から PDU を送信順に受信する．E_j から p^j を受理するごとに，AL_{kj} が，$p^j.ACK_k$ により変更される（$k=1,\cdots,n$）．

3.3 前 確 認

次に，受理した PDU を前確認する方法を示す．p

を，E_j が送信した PDU とする．E_i が受理した p について，群内の各実体から p に対する確認通知を受信することで，p は前確認される．E_i が，$p.SEQ<q.ACK_j$ である PDU q を E_k から受信することにより，E_k で p が受理されていることがわかる．各 AL_{jk} は，「$p^j.SEQ<AL_{jk}$ なる p^j は，既に E_k で受理されている」ことを示す $(j, k=1, \cdots, n)$．したがって，$p^j.SEQ<minAL_j$ ならば，群内の全実体が既に p^j を受理していることがわかる．したがって，E_i で受理された各 p^j は，以下の手続きにより，前確認される．

[前確認条件] $p^j.SEQ<minAL_j$. □

[前確認手続き] $j=1, \cdots, n$ について，$p^j=top(RRL_{ij})$ が前確認条件を充足する間，{p^j を RRL_{ij} から取り出し，PRL_{ij} に追加する．$PAL_{kj}:=p^j.ACK_k$ $(k=1, \cdots, n)$．$i=j$ ならば，p^j を SL_i から削除する．}． □

3.4 確　　認

下位層で1Cサービスが提供されている場合，ある PDU p を前確認する PDU の受信順序は，各実体で同一である．文献 19), 20), 23)では，この1Cサービスの性質を利用して，各応用実体が，PDU を同一順序で受信できる全順序放送通信 (TO) プロトコルを示した．TO プロトコルでは，受信の全順序性（すなわち，受信ログの同値性）を保つために，受信ログは一つである．ある PDU q が受信されたときに，p が前確認されたとする [**図 9** (1)]．1C サービスを用いているので，p を前確認する各実体からの PDU は，q 以前に受信されている．したがって，q が前確認されれば，p を前確認している全 PDU が前確認されている [図 9 (2)]．しかし，MC サービスを用いる OP プロトコルでは，順序保存性を保てばよいので，各 E_i は，各 E_k ごとに受信ログ RL_{ik} を持つ $(k=1, \cdots, n)$．p を確認するためには，p を前確認する各実体からの PDU が，すべて前確認されている必要がある．このために，本論文では，AL に加えて，行列 PAL を用いる．各 PAL_{jk} は，「E_j からの PDU の中で，E_k が次に前確認すると E_i が知っている PDU の通番」を表している $(j, k=1, \cdots, n)$．$minPAL_j$ を，$PAL_{j1}, \cdots,$ PAL_{jn} の最小値とする．E_i は，$q^j.SEQ<minPAL_j$ となるとき，群内の各実体が q^j を既に前確認していることがわかる [**図 10**]．以上から，PRL_{ij} 内の PDU は，以下の確認手続きにより確認される．

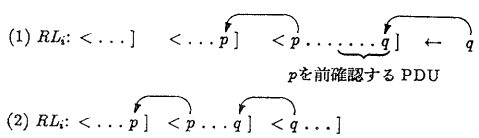


図 9 TO プロトコルでの確認
Fig. 9 Acknowledgment in the TO protocol.

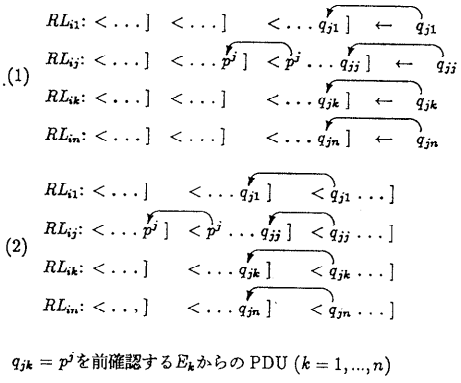


図 10 OP プロトコルでの確認
Fig. 10 Acknowledgment in the OP protocol.

[確認条件] $p^j.SEQ<minPAL_j$. □

[確認手続き] $j=1, \cdots, n$ について，$p^j=top(PRL_{ij})$ が確認条件を充足する間，{p^j を PRL_{ij} から取り出し，ARL_{ij} に追加する．}． □

3.5 障害復旧

MC サービスを利用した場合，PDU の紛失が発生する．各 E_i は，以下の障害条件によって，PDU の紛失を検出できる．

[障害条件] [**図 11**]

(1) p^j を受信したとき，$REQ_j<p^j.SEQ$ ならば，$REQ_j \leq q^j.SEQ<p^j.SEQ$ である q^j を紛失してる．

(2) q^h を受信したとき，ある $j(\neq h)$ について $REQ_j<q^h.ACK_j$ ならば，$REQ_j \leq q^j.SEQ<q^h.ACK_j$ である q^j を紛失している． □

次に，紛失 PDU の復旧方法を考える．PDU の再送方法には，go-back-n 方法と，選択的再送方法がある．go-back-n 方法では，紛失した PDU 以降に受信した PDU を廃棄し，これらを再送する．この方法で

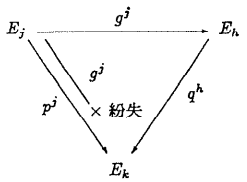


図 11 障害の検出
Fig. 11 Detection of lost PDUs.

は，手順は簡単であるが，正しく受信されたPDUも廃棄され，再送されるために，効率の問題がある．一方，選択的再送方法では，PDUの廃棄を行わず，紛失したPDUのみを選択的に再送する．この方法は，効率は良いが手順が複雑であり，go-back-n方法と比較してより多くのバッファ容量を必要とする．TOプロトコルでは，受信ログ間の同値性を保つために，go-back-n方法を用いている[21]．しかし，OPプロトコルでは，受信ログを各実体ごとの副ログとして保持しており，各副ログ内のPDUの位置は，それらの通番（SEQ）により，一意に決定できる．このことから，選択的に再送したPDUを受信ログの適当な位置に挿入することができ，PDUの廃棄をなくし，再送PDUを減少できる．紛失したPDU以降に受信したPDUは，紛失したPDUが再送されるまで，前確認されない．したがって，これらのPDUを記憶しておくためのバッファ容量が，余分に必要となる．この量は，最大で，各実体での送信ウィンドウ幅の和である．しかし，現在メモリの大容量化が可能となってきているので，問題とはならないと考える．

[選択的再送の例] E_1が，PDU a, b, c, d, e, f, gをこの順に送信し，E_2はこれらのすべてを受信したが，E_3はcとdを受信できなかったとする[**図12**]．TOプロトコルでは，E_3は，c以降に受信したPDUを廃棄し，E_1がc, d, e, f, gを再送する．この方法では，受信したPDUの廃棄と再送により，効率が低下するので，紛失したcとdのみを再送することを考える．cとdのみを再送しても，各PDUの通番の大小関係から，E_3はこれらを受信ログの適当な位置に挿入できる．つまり，$b.SEQ<c.SEQ<d.SEQ<e.SEQ$により，再送されたcとdを，この順にbとdの間に挿入できる．□

PDU p^jの紛失を発見したE_iは，送信実体E_jに，再送を要求する．また，再送要求を受信した実体では，要求のあったPDUのみを再送する．このために，再送要求PDUは，E_jに対して，再送すべきPDU通番の範囲を示すための項目$GAPS_j$, $GAPE_j$を含む．障害条件(1)または(2)が充足されたとき，$GAPS_j=REQ_j$とし，(1)ならば$GAPE_j=p^j.SEQ$，または(2)ならば$GAPE_j=q^h.ACK_j$となる．E_jが，再送要求PDUを受信したとき，$GAPS_j \neq NIL$ならば，$GAPS_j$から$GAPE_j$までのPDUを再送する．図12の例では，$GAPS_1=REQ_1$ $(=c.SEQ)$，$GAPE_1=e.SEQ$となる．

また，各E_iは，最後に送信したPDU pが一定時間内に前確認されないならば，pまたはpに対する確認通知を運ぶPDUが紛失したと判断し，pを再送する．

$RL_{21}: <a\ b\ c\ d\ e\ f\ g]$　　$SL_1: <a\ b\ c\ d\ e\ f\ g]$

$RL_{31}: <a\ b\ e\ f\ g]$

図12 選択的再送の例

Fig. 12 An example of the selective retransmission.

4. 評　　価

OPプロトコルを，Ethernetで接続されたSunワークステーション（SPARC 2, 28.5 MIPS）上で，UNIXのユーザプロセスとして実装し，OPプロトコルの性能評価を行った．各ワークステーションごとに，一つのOPモジュールを実装した．EthernetのMACインタフェースには，NIT[18]を用いた．プログラムは，C言語で約5,000ステップであり，実行形式の大きさは約50 KBである．

評価は，OPの応用実体が，ファイル転送のように，常に送信すべきPDUを持つ高負荷の状況で行った．PDU長は固定（180バイト）とし，各実体が同一のデータ量を送信するものとした．この時，実体数nを変化させた場合の，1 PDUを受信するために要したCPU時間（T_{op}）と，応用実体間での遅延時間（T_{ap}）を，**図13**に示す．T_{op}は，OPプロトコルモジュールが，上位層（応用実体）からデータを受け取ってから，これを確認するまでに要したCPU時間の合計である．T_{ap}は，応用実体がデータを放送してから，これが群内の応用実体で受信されるまでに要した実時間である．図からわかるように，nに対して

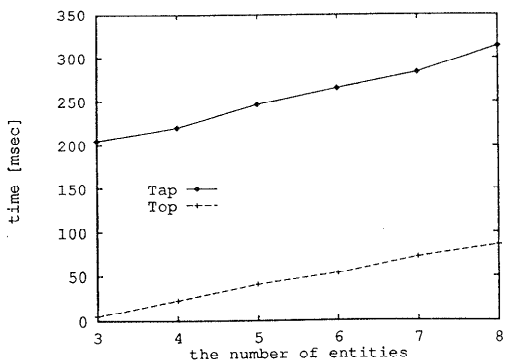


図13 応用実体間の伝送遅延時間と1PDU当たりのCPU時間

Fig. 13 Transmission delay time and CPU time for receipt processing per PDU.

T_{op} は線形であり，n=5 のとき，約 40 msec である．T_{ap} も，n に対して，線形に増加していることがわかる．文献12)，13)，22) で述べたように，CPU と PDU についての計算量は $O(n)$ であることがわかる．

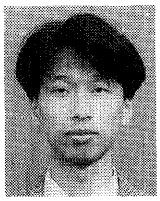
5. まとめ

本論文では，Ethernet 等の低信頼な放送通信サービスを利用して，信頼性のある放送通信サービスを提供する OP プロトコルの設計について述べた．OP プロトコルによって提供される OP サービスを利用することにより，各実体から送信された PDU は，群内の全実体に送信順に届けられる．OP プロトコルは，分散型制御と，群という概念に基づいている．群とは，複数の実体の集合であり，コネクション[16]を拡張したものである．MC サービス上で，OP プロトコルが OP サービスを提供することを示し，プロトコルの実装を行い，性能評価を行った．OP プロトコルを利用することにより，グループウェア等での協調動作を容易に実現し，効率的に実行できる．今後は，他の放送通信プロトコル[14),15),21)]を含めた，総合的な評価を行う予定である．

参考文献

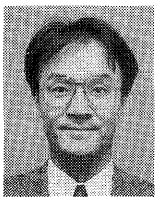
1) Birman, K., Schiper, A. and Stephenson, P.: Lightweight Causal and Atomic Group Multicast, *ACM Trans. Comput. Syst.*, Vol. 9, No. 3, pp. 272-314 (1991).
2) Chang, J. M. and Maxemchuk, N. F.: Reliable Broadcast Protocols, *ACM Trans. Comput. Syst.*, Vol. 2, No. 3, pp. 251-273 (1984).
3) Ellis, C. A., Gibbs, S. J. and Rein, G. L.: Groupware: Some Issues and Experiences, *Comm. ACM*, Vol. 34, No. 1, pp. 38-58 (1991).
4) Garcia-Molina, H. and Kogan, B.: An Implementation of Reliable Broadcast Using an Unreliable Multicast Facility, *Proc. of the 7th IEEE Symp. on Reliable Distributed Systems*, pp. 428-437 (1988).
5) Garcia-Molina, H. and Spauster, A.: Message Ordering in a Multicast Environment, *Proc. of IEEE ICDCS-9*, pp. 354-361 (1989).
6) IEEE: IEEE Project 802 Local Network Standards-Draft (1982).
7) Kaashoek, M. F., Tanenbaum, A. S., Hummel, S. F. and Bal, H. E.: An Efficient Reliable Broadcast Protocol, *ACM Operating Systems Review*, Vol. 23, No. 4, pp. 5-19 (1989).
8) Luan, S. W. and Gligor, V. D.: A Fault-Tolerant Protocol for Atomic Broadcast, *IEEE Trans. Parallel and Distributed Systems*, Vol. 1, No. 3, pp. 271-285 (1990).
9) 松下 温(編著)：図解グループウェア入門，オーム社 (1991).
10) Melliar-Smith, P. M., Moser, L. E. and Agrawala, V.: Broadcast Protocols for Distributed Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol. 1, No. 1, pp. 17-25 (1990).
11) Metcalfe, R. M.: Ethernet: Distributed Packet Switching for Local Computer Networks, *Comm. ACM*, Vol. 19, No. 7, pp. 395-404 (1976).
12) Nakamura, A. and Takizawa, M.: Reliable Broadcast Protocol for Selectively Partially Ordering PDUs (SPO Protocol), *Proc. of IEEE ICDCS-11*, pp. 239-246 (1991).
13) Nakamura, A. and Takizawa, M.: Design of Reliable Broadcast Protocol for Selectively Partially Ordering PDUs, *Proc. of IEEE COMPSAC 91*, pp. 673-679 (1991).
14) 中村章人，滝沢 誠：多チャネル上の選択的放送通信プロトコルのデータ転送手続き，情報処理学会論文誌，Vol. 33, No. 2, pp. 223-233 (1992).
15) Nakamura, A. and Takizawa, M.: Priority-Based Total and Semi-Total Ordering Broadcast Protocols, *Proc. of IEEE ICDCS-12*, pp. 178-185 (1992).
16) ISO: Open Systems Interconnection—Basic Reference Model, ISO 7498 (1987).
17) Schneider, F. B., Gries, D. and Schlichting, R. D.: Fault-Tolerant Broadcasts, *Science of Computer Programming*, Vol. 4, No. 1, pp. 1-15 (1984).
18) Sun Microsystems: Sun OS Reference Manual, Revision A of 27 March (1990).
19) Takizawa, M.: Cluster Control Protocol for Highly Reliable Broadcast Communication, *Proc. of the IFIP Conf. on Distributed Processing*, pp. 431-445 (1987).
20) Takizawa, M.: Design of Highly Reliable Broadcast Communication Protocol, *Proc. of IEEE COMPSAC 87*, pp. 731-740 (1987).
21) 滝沢 誠，中村章人：1チャ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，情報処理学会論文誌，Vol. 31, No. 4, pp. 609-617 (1990).
22) Takizawa, M. and Nakamura, A.: Partially Ordering Broadcast (PO) Protocol, *Proc. of the 9th IEEE INFOCOM*, pp. 357-364 (1990).
23) Takizawa, M. and Nakamura, A.: Reliable Broadcast Communication, *Proc. of IPSJ Int'l Conf. on Information Technology (Info-Japan)*, pp. 325-332 (1990).

(平成4年4月24日受付)
(平成4年10月8日採録)



中村 章人（正会員）

1966年生．1989年東京電機大学理工学部経営工学科卒業．1991年同大学院工学研究科修士課程修了．現在，同大学院理工学研究科博士後期課程在学中．分散型システム，通信プロトコル等に興味を持つ．



滝沢 誠（正会員）

1950年12月6日生．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，1987年より同助教授．工学博士．1989年9月より1年間ドイツ国立情報処理研究所（GMD）客員教授．1990年7月よりKeele大学（英国）客員教授．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），「データベースシステム入門技術解説」（ソフトリサーチセンタ）．