

## 因果関係順序付け選択的グループ通信プロトコル\*

7 U-3

坂元 紫穂子 滝沢 誠†

東京電機大学‡

e-mail{shihoko,taki}@takilab.k.dendai.ac.jp

### 1はじめに

ワークステーションと通信技術の発達により、複数のコンピュータを通信網により相互接続させた分散型システムが広く用いられている。分散型システムでは、従来の一対一型通信に加えて、複数プロセス間でのグループ通信が必要となる。応用によっては、グループ内の全プロセスを宛先とするグループ通信に加えて、宛先をグループ内のプロセスに特定できる選択的グループ通信[3]が要求される。また、グループ内で送信されるメッセージを、各プロセスが因果関係順に受信できる必要がある。例えば、CSCWでワークステーションのウィンドウを共有するとき、ウィンドウ操作要求は因果関係順に各ワークステーションに届けられる必要がある。本論文では、因果関係順の受信を行なえる選択的グループ通信を提供する因果関係順序付け選択的グループ通信(SCO)を考える。さらに、メッセージ紛失が存在する下で、SCOサービスを提供するプロトコルを考える。本研究では、完全分散型の制御を用いる。[2, 3, 4]では放送網を用いたが、ここでは一対一型高速網を利用する。

2章では、通信サービスのモデルを示す。3章では、データ転送手続きを述べ、4章では、評価を示す。

### 2 モデル

通信システムは、図1に示す3階層から構成される。システム層の各プロセス  $E_i$  は、網層が提供する高速な通信サービスを利用して、応用層に高信頼なグループ通信サービスを提供する。高速網では、通信速度が処理速度より速いことから、プロセスがメッセージの受信に失敗し、また、オーバーフローによりメッセージが紛失する場合がある。

$E_i$  におけるメッセージ  $m$  の送信事象と受信事象をそれぞれ、 $send_i(m)$  と  $receive_i(m)$  で表す。事象間の先行関係  $\rightarrow$ [1]を以下のように定義する。

[定義] 任意の事象  $e_1$  と  $e_2$  間で以下のいずれかが成り立つならば、 $e_1 \rightarrow e_2$  である。

(1)  $E_i$ において、 $e_1$  が  $e_2$  に先行する。

(2) ある  $E_i$  と  $E_j$  と  $m$ について、 $e_1 = send_i(m)$ かつ  $e_2 = receive_j(m)$  である。

(3)  $e_1 \rightarrow e_3$ 、 $e_3 \rightarrow e_2$  である  $e_3$  が存在する。□

次に、メッセージ間の因果関係[2]を定義する。

[定義] 任意のメッセージ  $p$  と  $q$  について、 $send_i(p) \rightarrow send_j(q)$  ならば、 $p$  は  $q$  に因果先行する ( $p < q$ )。□ 任意のメッセージ  $p$  と  $q$  について、 $p < q$  であるならば、 $p$  と  $q$  の共通の宛先は、 $p$  の後に  $q$  を受信する選択的グループ通信を因果関係順序付け選択的(SCO)グループ通信とする。

\*Selective Causally Ordering Group Communication Protocol  
†Shihoko Sakamoto and Mokoto Takizawa

‡Tokyo Denki University

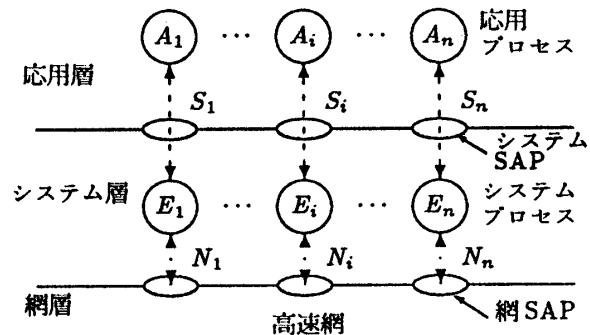


図1: システムモデル

### 3 データ転送手続き

網層は、メッセージ紛失の可能性のある、高速な通信サービスをシステム層に提供する。システム層は網サービスを利用し、メッセージ紛失のない、因果関係順序付けされた、選択的グループ通信サービスを応用層に提供する。グループ通信では、メッセージは、全宛先で受信されたか、もしくはされなかったかのどちらかである原子性が必要となる。このため、 $E_i$  は、メッセージが全宛先で受信されたかどうかの確認を行なう必要がある。ここでは、プロセスは高信頼とする。

SCOプロトコルは、分散型制御を用いる。各  $E_i$  は、メッセージ  $p$  を受信したならば、 $p$  の受信通知を  $p$  の全宛先に送信する必要がある。メッセージを受信する毎に、受信通知を送信すると、メッセージ数が増大してしまう。また、受信通知をビギーバックさせると、 $E_j$  からメッセージを受信しても  $E_j$  に送信するデータがないと、受信通知を  $E_j$  に届けられない。このために、遅延時間が増加してしまう。分散型制御で、メッセージ数を減少させ、遅延時間を増大させないために、以下の方式を用いる。

(1) 送信メッセージに受信通知をビギーバックさせる。

(2) 一定時間内に、受信したメッセージで受信通知を送信していないものがあるとき、一定時間後に、受信通知を送信する(遅延確認)。

$E_i$  は図2のような FIFO キューを持つ。網層より受けとったメッセージ  $p$  は、まずメッセージ紛失の検出が行なわれる。紛失がなければ、キュー  $RRQ_k$  ( $p$  の送信元プロセスを  $E_k$  とする)に入れられる。次に、因果関係に基づいて順序付けがされ、 $CRQ$  に移される。最後に、全宛先で受信されたかどうかの確認がなされたら、 $ARQ$  に移される。応用層のプロセスは  $ARQ$  よりメッセージを取り出す。

各メッセージは図3のような形式を持つ。 $src$  は送信元プロセス、 $dst$  は宛先プロセスの集合を表し、 $sqn$  は

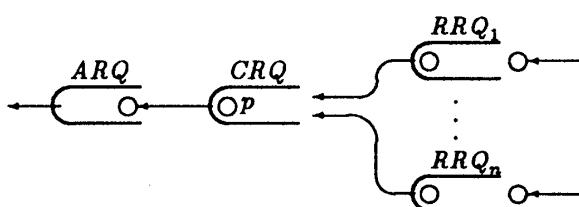


図 2: キューとメッセージの動き

src	dst	sqn	lsn <sub>1</sub> , ..., lsn <sub>n</sub>	ack <sub>1</sub> , ..., ack <sub>n</sub>	csn <sub>1</sub> , ..., csn <sub>n</sub>	data
-----	-----	-----	--	--	--	------

図 3: メッセージの形式

メッセージ通番を表す。 $sqn$  は、送信元プロセスがメッセージを送信する度に、1 加算される。 $lsn_i$  は、各  $E_i$  に対する副通番で、送信元プロセスが  $E_i$  にメッセージを送信する度に、1 加算される。各  $E_i$  は、 $lsn$  によりメッセージ紛失を検出する。 $ack_i$  は送信元プロセスの受信通知であり、送信元プロセスが  $E_i$  からのメッセージで、最後に受理したメッセージの通番を表す。メッセージを受信した各宛先プロセスは、 $ack$  により、送信元プロセスが受理したメッセージを知ることができ、確認を行なう。 $csn_i$  は因果情報であり、 $E_i$  からのメッセージに対し、最後に因果先行するメッセージの通番を表す。データ転送手続きの基本手順を図 4 に示す。

各  $E_i$  は、 $csn$  により順序付けを行なう。

[定理] 2つのメッセージ  $p$  と  $q$  について、 $p \prec q$  ならば、かつそのときに限り、以下が成り立つ。

- (1)  $p.src = q.src$  ならば、 $p.sqn < q.sqn$ 。
- (2)  $p.src \neq q.src$  ( $p.src = E_j$ ) ならば、 $p.sqn < q.csn_j$ 。

□

以上の定理を用いて、因果関係に基づき、順序付けを行なう。

#### 4 評価

本プロトコルを、1つのメッセージが確認されるまでのメッセージ数と遅延時間により評価する。グループ内のプロセス数  $n$  を 10 とする。平均宛先数  $m$  ( $\leq n$ ) を変化させたときのメッセージ数と遅延時間を、集中型制御  $C(m)$  と、SCO プロトコル  $D(m)$  について、図 5 と図 6 に示す。図 5 より、 $C(m) < D(m)$  である。図 6 より、集中型に対して、分散型の SCO プロトコルが遅延時間を減少できることがわかる。

#### 5 おわりに

本研究では、一对一型高速網を用いた、因果関係順序付け選択的グループ通信(SCO)プロトコルの設計を行ない、評価を示した。集中型に対して、SCO プロトコルがメッセージ数と遅延時間を減少できることを示した。

#### References

- [1] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System," *Comm. ACM*, Vol.21, No.7, 1978, pp.558-565.
  - [2] Nakamura, A. and Takizawa, M., "Causally Ordering Broadcast Protocol," *Proc. of IEEE ICDCS-14*, 1994, pp.48-55.
  - [3] Nakamura, A. and Takizawa, M., "Reliable Broadcast Protocol for Selectively Partially Ordering PDUs (SPO Protocol)," *Proc. of IEEE ICDCS-11*, 1991, pp.239-246.
  - [4] Tachikawa, T. and Takizawa, M., "Selective Total Ordering Broadcast Protocol," *Proc. of the 2nd IEEE ICNP*, 1994 pp.212-219.
- ```

send_i(p) (k=1, ..., n)
  p.dst:=p の宛先; p.src:=E_i;
  p.sqn:=SQN; SQN:=SQN+1;
  p.lsn_k:=LSN_k;
  if E_k ∈ p.dst then LSN_k:=LSQ_k+1;
  p.csn_k:=CSN_k;
  p.ack_k:=AL_k;
receive_j(p) (p.src は E_i, k=1, ..., n)
  if p.lsn_j=LRN_i {
    LRN_i:=LRN_i+1;
    p を RRQ_i 入れる; CSN_i:=p.sqn;
    CSN_k:=max(CSN_k, p.csn_k);
    while (RRQ_k ≠ φ) {
      q:=causality();
      q を RRRQ_k から CRQ に移す;
      AL_j:=q.sqn; }
    AL_k:=p.ack_k;
    while (acknowledge() ≠ φ) {
      r:=acknowledge();
      r を CRQ から ARQ に移す; } }
  else 再送要求;
causality()
  top(RRQ_k) の中で最も因果先行するメッセージ q を探す;
  (q.sqn < p.csn_k, q.src=E_k)
acknowledge()
  r:=top(CRQ);
  if r.sqn < min(AL_hj) (E_h ∈ r.dst) then return(r);
  else return(φ);

```
- 図 4: データ転送手続き
- | m  | D(m) | C(m) |
|----|------|------|
| 1  | 1.0  | 1.0  |
| 2  | 0.8  | 1.0  |
| 3  | 0.6  | 1.0  |
| 4  | 0.5  | 1.0  |
| 5  | 0.4  | 1.0  |
| 6  | 0.35 | 1.0  |
| 7  | 0.3  | 1.0  |
| 8  | 0.28 | 1.0  |
| 9  | 0.26 | 1.0  |
| 10 | 0.24 | 1.0  |
- 図 5: メッセージ数
- | m  | D(m) | C(m) |
|----|------|------|
| 1  | 1.0  | 1.0  |
| 2  | 0.8  | 1.0  |
| 3  | 0.65 | 1.0  |
| 4  | 0.55 | 1.0  |
| 5  | 0.45 | 1.0  |
| 6  | 0.38 | 1.0  |
| 7  | 0.32 | 1.0  |
| 8  | 0.28 | 1.0  |
| 9  | 0.25 | 1.0  |
| 10 | 0.22 | 1.0  |
- 図 6: 遅延時間