

## ネットワーク下位プロトコルを考慮した 分散アルゴリズムシミュレータ<sup>†</sup>

6E-9

若林真一      有永憲一      細谷好志      小出哲士      吉田典可  
 広島大学 工学部

### 1 はじめに

複数のプロセスが互いに協調動作する分散システムを評価する場合、シミュレーションによる実験的評価が重要である。これまでに著者らは、分散アルゴリズムの上位プロトコルレベルでの実験的評価を行うための分散アルゴリズムシミュレータを開発してきた [2]。しかし、高速な LAN などを使ったシステムではネットワーク下位プロトコルによる影響を無視することができず、それを考慮に入れた分散アルゴリズムの詳細な評価が必要である。本稿では、分散アルゴリズムシミュレータにおけるネットワーク下位プロトコルを考慮したシミュレーション機能とシミュレーションデータ解析機能の実現、及び相互排除分散アルゴリズムのシミュレーションとその実験的評価を行った結果について報告する。

### 2 分散アルゴリズムシミュレータ

シミュレータが対象とする分散処理モデルは、プロセスの集合とプロセス間通信のための通信リンクの集合から構成されるものとする。各プロセスは共有メモリを持たず、プロセス間通信はメッセージバッシングのみとし、リンクでのメッセージの損失はない。通信リンクの方向性、通信遅延、及び通信容量は任意に設定可能である。ネットワークは、実際に通信遅延や通信容量を任意に設定できる物理ネットワークと評価対象アルゴリズムが仮定する論理ネットワークをそれぞれ個別に設定することができ、シミュレータ内で自動的に対応づけられる。ネットワーク下位プロトコルは、イーサネットプロトコルを考える。

本シミュレータは各プロセスが実行するアルゴリズムを機械語命令単位で詳細にシミュレーションするシステムである。シミュレーションは主に、各プロセスを矛盾なく同時実行させるスケジューラ及び各プロセスの命令実行とプロセス間のメッセージ移動を行う命

令実行部をシミュレーション制御部が集中的に制御することによって行われる。シミュレーションデータ解析部は、シミュレーションから得られたパフォーマンスデータを解析し、システムの負荷特性等を出力する。

### 3 シミュレーションの実現方法

イーサネット型ネットワークにおけるシミュレーションでは、データリンク層プロトコルとして CSMA/CD を考慮している。図 1 に実現モデルを示す。

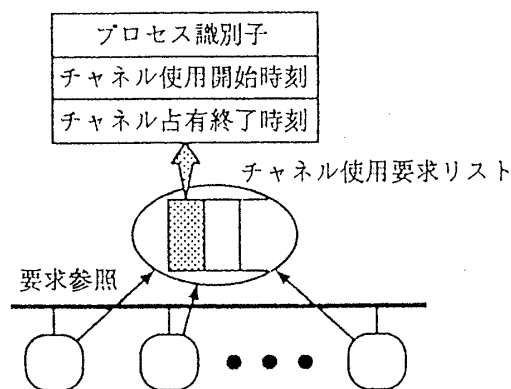


図 1 イーサネットプロトコルの実現

チャンネルはリスト構造でモデル化されており、各リストはチャンネルを使用したいプロセスのチャンネル使用要求、つまり各プロセスがチャンネルを占有する時刻を保持している。あるプロセスがチャンネルを使用したい場合、まずこのチャンネル要求リストを参照し、他のプロセスがいつまでチャンネルを占有するか調べる。その時、リスト内に他のプロセスのチャンネル使用要求がなければ、チャンネルは空いているのでチャンネル要求リストに要求を入れる。一方リスト内に他のプロセスのチャンネル使用要求がある場合、そのプロセスの時刻が要求リスト内の送信開始時刻からスロットタイム以内であれば、いずれ衝突が起こることになるため、そのプロセスはただちに送信を中断し、切捨てバイナリエクスポネンシャルバックオフアルゴリズム (truncated binary exponential back off) にしたがって再送信を試みる (衝突検出)。また、そのプロセスの現在時刻が要求リスト

<sup>†</sup>"A distributed algorithm simulator considering low level network protocols", Shin'ichi WAKABAYASHI, Ken'ichi ARINAGA, Takashi HOSOTANI, Tetsushi KOIDE, and Noriyoshi YOSHIDA, Faculty of Engineering, Hiroshima University.

内の送信終了時刻よりも前ならば、チャンネルが空くまで送信を見合わせる。チャンネルが空けば、一定時間待つて再度送信を試みる(キャリア検知)。

シミュレーションデータ解析部は、ユーザの入力したプロセス数、バケット長、通信容量、伝搬遅延時間等の設定パラメータ、及びシミュレーション結果より得られる送信要求の分布等のパフォーマンスデータよりチャンネルのスループット、平均待ち時間等のイーサネットのトラフィック特性を出力する。これにより、上位プロトコルレベルでは得られない下位プロトコルレベルの影響を考慮した解析結果を得ることができる。

#### 4 実験結果

本シミュレータは NEC EWS4800/220 上で C 言語を用いて実現されている。実験では Ricart と Agrawala の相互排除アルゴリズム RA[1] をシミュレータ上でシミュレートし、資源の要求頻度とチャンネルのスループット及び平均資源獲得時間との関係をそれぞれ評価した。RA は、資源を使用したいプロセスがネットワークに属するすべてのプロセスに REQUEST を送信し、すべてのプロセスから REPLY を受信すれば資源を使用することにより相互排除を行うアルゴリズムである。

図 2 はプロセス数 10, 50 の場合の資源要求確率とチャンネルのスループットの関係を示したもので、図 3 は資源要求確率 0.01, 0.05, 0.10, 1.00 の場合のプロセス数と平均資源獲得時間の関係を示したものである。

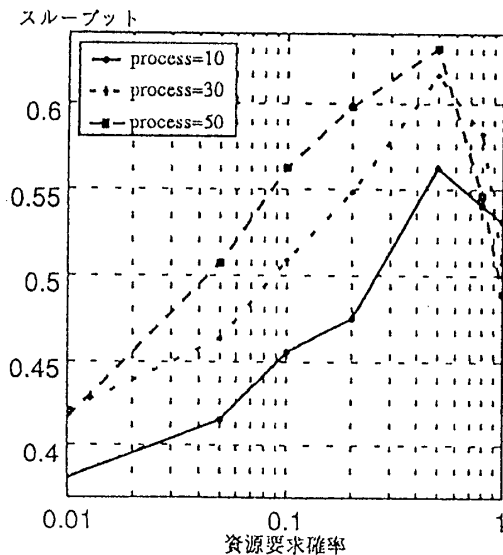


図 2 資源要求確率とスループット

図 2 より、資源要求確率の増加に伴ってチャンネルのスループットが増加するが、ある確率以上になるとスループットが減少することが分かる。これは、資源要

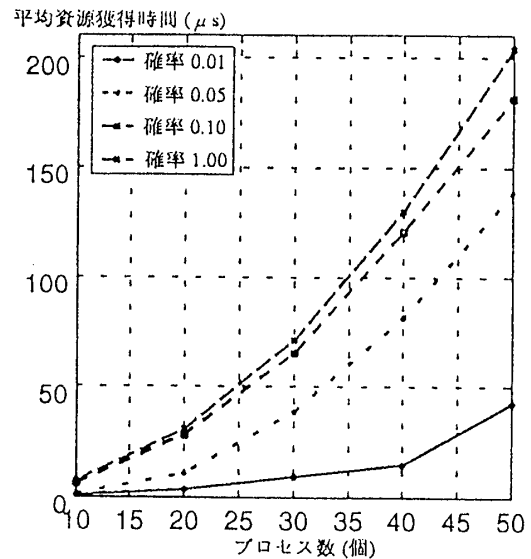


図 3 プロセス数と平均資源獲得時間

求確率がある程度以上になるとチャンネルでのメッセージの衝突回数が増加するためであると考えられる。また、プロセス数が多いほどチャンネルのスループットが急激に増加し、減少の割合も急激であることが分かる。

図 3 より、プロセス数の増加及び資源要求確率の増加に伴って平均資源獲得時間が増加することが分かる。これは、プロセス数が多くなるとそれだけ単位時間当たりの資源要求が多くなり、チャンネルでのメッセージの衝突が多くなるためであると考えられる。また、資源要求確率がある程度以上高くなると、平均資源獲得時間の増加の割合が少なくなっている。

#### 5 あとがき

本稿では、分散アルゴリズムシミュレータにおけるネットワーク下位プロトコルを考慮したシミュレーション機能の実現を行った。実験より本分散アルゴリズムシミュレータは、ネットワーク下位プロトコルとしてイーサネットプロトコルを考慮したシミュレーションが可能であることが確認された。

今後の課題として、メッセージ到着間隔を考慮に入れた分散アルゴリズムの評価などが挙げられる。

本研究の成果の一部は文部省科学研究費補助金一般研究(B)(課題番号 04452195)による。

#### 文献

- [1] 亀田, 山下: “分散アルゴリズム”, 出版予定.
- [2] 若林, 他: “ネットワークの形状と通信遅延モデルを任意に設定可能な分散アルゴリズムシミュレータ”, 情処第 45 回全国大会, pp. 5-279-5-280 (1992).