並行処理型待ち行列網シミュレータ D-SSQについて

佐藤 圭　渡辺 尚　中西 暉　真田英彦　手塚慶一
（大阪大学工学部）

1. まえがき

待ち行列網シミュレーションは，待ち行列網が元来有する高い並列性のため，複数の並列処理に分割して並行処理を行うことにより処理能力の改善が期待できる。

ところが，待ち行列網シミュレーションには処理内容に乱数を用いて決定する部分があるため事前の処理スケジュールが行えないという非決定的特性が存在する。このため，画像処理等で多く用いられるようなプロセッサの同期制御[2]を行うと処理待ちによる遊休状態が多発し十分に並列性が利用できないことになる。そこで，このようなプロセッサ間の相互関連を非決定的に発生するジョブに対しては，非同期制御を行うことにより処理効率の大幅な改善が図れると考えられるが，各処理装置の時刻が違うことにより論理矛盾が発生する可能性があるため処理結果の正しさを保証する必要が生じる。

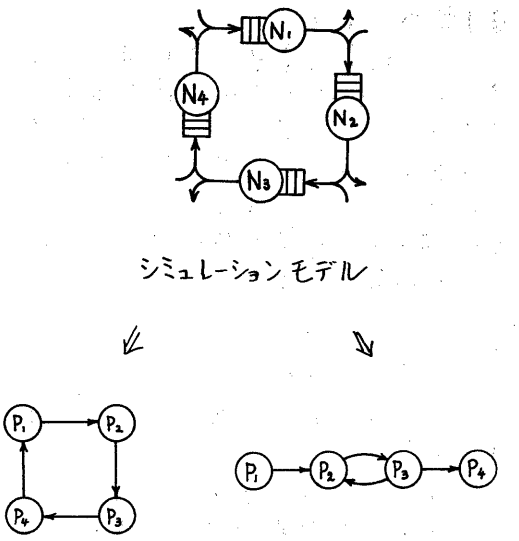
従来提案されている実行制御方式[1][3]は，論理矛盾が発生する可能性がない場合にのみ処理を進行させる方式であるが，このようにすると論理ループがあるモデルではほとんどのプロセッサが遊休状態となり処理の並行度は上がらない。そこで，本研究では，実行制御方式として論理矛盾の発生を許容することにより処理の並行性をあげ，論理矛盾が発生した場合にはその時点で矛盾の解消を行う方法により処理能力の大きな改善が得られると考えられる先行制御方式を採用する。

本稿では，ノード分散事象型待ち行列網シミュレータ D-SSQ（Distributed System Simulator for Queueing network）の実行制御方式である先行制御方式について述べる。さらに，実験システムとシミュレーションにより処理能力の検討を行う。

2. シミュレータの分散化

待ち行列網シミュレーションを分散処理する場合，待ち行列網のノードが並行動作することに着目してノードに対してプロセッサを割り当てるノード分散方式と事象が並列実行可能であることから事象に対してプロセッサを割り当てるプロセス分散方式，さらに，両分散方式を組合せた複合分散方式が考えられる。図1にノード分散とプロセス分散の例を示す。同図で N_i はノード，P_i はプロセッサを表す。

D-SSQにおいてはモデルとの対応がつき易いノード分散を行っている。



シミュレーションモデル

(a) ノード分散　　(b) プロセス分散

図1　シミュレータの分散化

3. 先行制御方式

先行制御方式は，個々のプロセッサが他のプロセッサから受ける制限を無視して処理を行い処理の並行度を高め，このために生じるプロセッサ間通信の際の論理矛盾に対しては，その発生を検出する毎にキャンセル処理により矛盾の解消を行う方式である。

各プロセッサは自プロセッサ内の事象をシミュレーション時刻の小さいものから順に処理する。事象が他プロセッサに対して通信を必要とする場合プロセッサ間通信が行われる。その際，送出されるメッセージにはそのメッセージが受信側プロセッサで処理されるべき時刻（到着時刻）が書き込まれる。受信側プロセッサの時刻が到着時刻よりも進んでいる場合に矛盾が発生する。

プロセッサは矛盾を検出すると矛盾解消のためのキャンセル処理を開始する。キャンセル処理はプロセッサの時刻を到着時刻へ戻し，その時刻以後の事象に対して行われた処理を全て未処理の状態へ戻す処理である。

キャンセル対象の事象の中に他のプロセッサへのメッセージの送信を伴う事象を発見すると，そのプロセッサに対してキャンセル要求を送出し，既に送ったメッセージの取消しを要求する。

キャンセル処理は発生経過等により次の3つの場合が考えられる。

(i) モード1
メッセージの到着時刻よりもプロセッサの時刻が進んでいる場合。

(ii) モード2
キャンセル要求を受信した際にキャンセル対象の事象が未処理の場合。

(iii) モード3
キャンセル要求を受信した際にキャンセル対象の事象が既に処理されている場合。

キャンセル処理はプロセッサがシミュレートしているノードの状態を過去のある時刻まで戻す操作である。D-SSQにおいては事象の処理の履歴を保存することにより，矛盾が発生した場合には，ノードの現在状態と戻るべき時刻から現在時刻までの処理の履歴から到着時刻のノードの状態の復元を行う。

しかしながら，過去の履歴をすべて保存するとシミュレーションの進行に伴って必要とされるメモリ量が増大するため，絶対にキャンセル対象となり得ない履歴を消去する必要がある。その際必要な「キャンセルが発生しても絶対に戻り得ない時刻」を“確定時刻”と呼ぶことにする。確定時刻検出方法には集中コントロール法と論理リング巡回法を考え考察した。(4)

集中コントロール法は図2に示すように，確定時刻検出用プロセッサを設ける方式である。

論理リング巡回法は図3に示すように全プロセッサで論理リングを構成し，この論理リング上に確定時刻検出パケットを巡回させておく方式である。

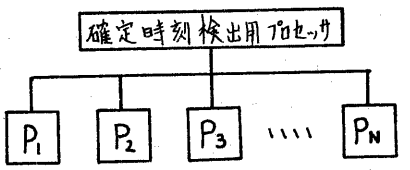


図2. 集合コントロール方式

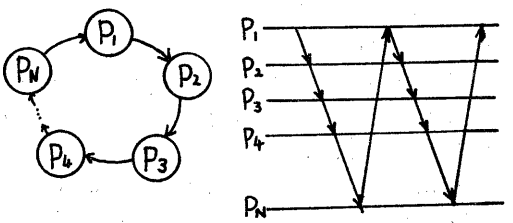


図3. 論理リング巡回方式

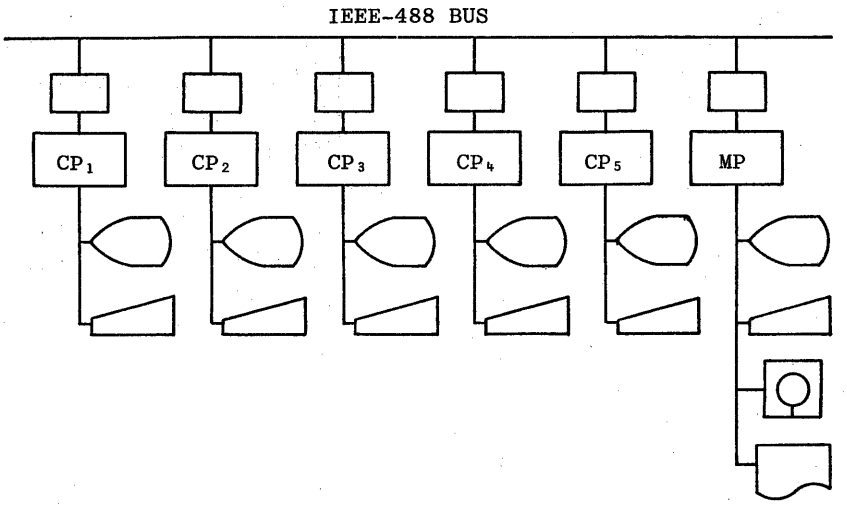


図４．実験システムのハードウェア構成

4.実験システムによる処理能力評価

D-SSQの処理能力等の諸特性を評価するために実験システムを構成した。図４に実験システムのハードウェア構成を示す。１台のＭＰ（Master Processor）と５台のＣＰ（Computing Processor）がＩＥＥＥ－488通信バスにより接続された構成である。

ＭＰはバスの制御，初期値入力，統計量出力，確定時確の検出を行う。ＣＰはノードの疑似動作，統計量の収集，ＭＰへの転送を行う。確定時刻検出方式は集中コントロール方式を採用しておりＭＰが確定時刻の検出を行う。

また，ＣＰ，ＭＰのソフトウェアはＢＡＳＩＣにより記述されており．図５にＣＰのソフトウェア構成，図６にＭＰのソフトウェア構成を示す。

本実験システムを用いてD-SSQの処理能力標価を行った。処理能力を次のように定義する。

処理能力
$$= \frac{\text{1プロセッサシミュレータにおける処理時間}}{\text{マルチプロセッサシミュレータにおける処理時間}}$$

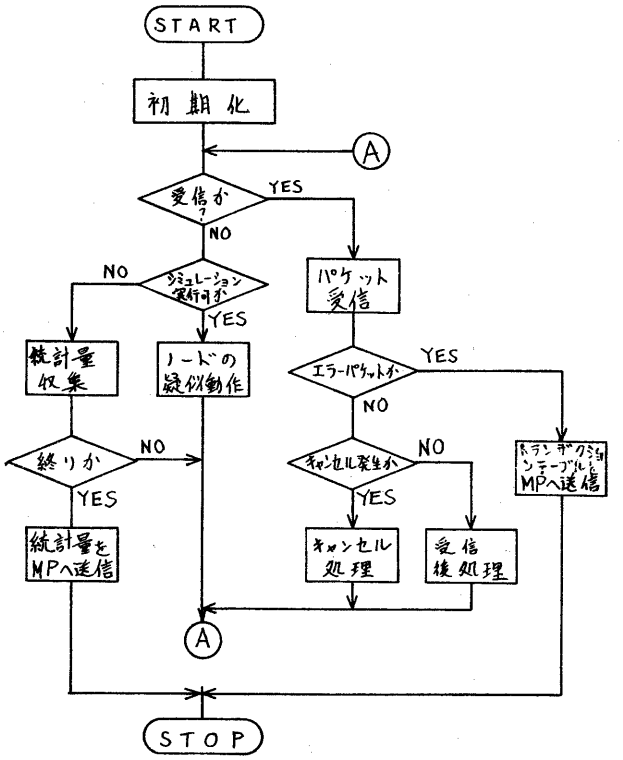


図５．ＣＰのソフトウェア構成

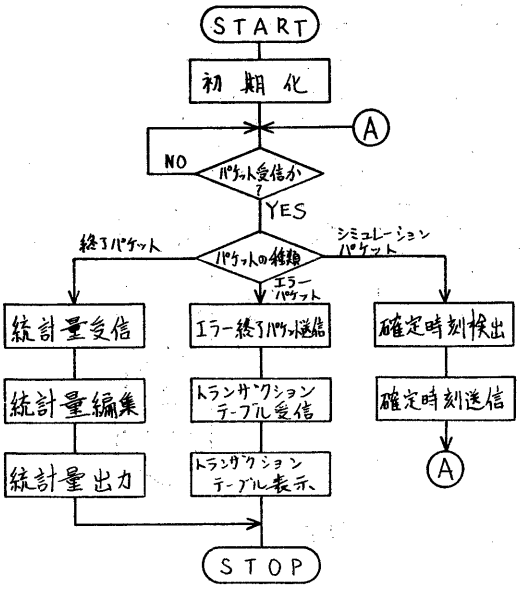


図6. MPのソフトウェア構成

図7は実際に作った2種のプログラムのそれぞれのオーバーヘッドをパラメータとする処理能力特性である。表1にこのオーバーヘッドの実測値を示す。同図より処理能力はプロセッサ台数に対して線形に増加すること，分散化オーバーヘッドが小さい方が処理能力が大きくまた処理能力の傾きも大きいことがわかる。

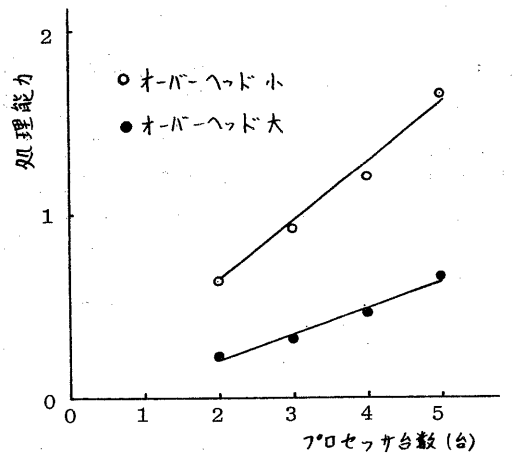


図7 処理能力特性

表1 分散化オーバーヘッド

分散化オーバーヘッド	小	大
通信を行わない事象	1.5	1.7
通信を行う事象	2.3	3.6
パケット受信	1.0	2.0
キャンセル処理(モード1)	2.0	4.5
キャンセル処理(モード2)	0.4	0.4
キャンセル処理(モード3)	1.6	3.5

但1プロセッサシミュレータの事象の処理時間を1とする。

5. シミュレーションによる検討

多数のプロセッサを使用する場合のD-SSQの処理能力の検討を行うため大型計算機上にD-SSQのシミュレータを構成しシミュレーションを行った。仮定は次のとおりである。

(i) 網形態は完全網でありプロセッサ数nをパラメータとする。
(ii) 通信を要求する事象は発生率rで発生する。
(iii) 時刻更新間隔は平均1の指数分布とする。
(iv) プロセッサの処理時間は表1のオーバーヘッド小の値を用い，指数分布とする。
(v) プロセッサ間通信に要する時間dは0.01単位時間必要とする。

シミュレーションでは実時刻を500単位時間実行しシミュレータが到達した時刻を求めた。

図8はr=1の場合の処理能力特性である。同図より処理能力はプロセッサ台数に対し線形に増加することがわかる。またd=0.1の場合もほとんど処理能力に差がないことがわかる。ところが，d=0.1, n=40の場合通信バスの利用率は約0.9という非常に高い値となるため，プロセッサ数がもう少し増加すると通信バスのふくそうによる急激な処理能力の低下がおこると考えられる。

図9はプロセッサがメッセージを受信した際の各処理の発生割合を示す。図10はその際の到着時刻と受信側プロセッサの時刻の差を示す。両図より，モード1のキャンセルの発生率はほぼ一定していることがわかる。さらに，プロセッサ数が少ないとキャンセルの伝搬範囲が限られるために，プロセッサ数の増加とともにモード2，モード3のキャンセル量が増加する。一方，プロセッサ数が大きくなると各処理の発生率はそれぞれ一定値に収束する。これは，プロセッサ数が大きくなってもキャンセルの伝搬長はある値以上大きくならないことを示している。換言すると，プロセッサ数が増大しても先行制御により発生するプロセッサ当りのキャンセルのオーバーヘッドはそれほど大きくならない。このことは，処理能力がプロセッサ台数に対し線形に増加することを裏づけている。

また，図8において処理能力の絶対値が非常に小さい。これは，分散化オーバーヘッドが非常に大きいこと，先行制御を無制限に行っているため，論理矛盾の発生が予測できるような場合でも先行処理を行うので無効処理量がかなり多いことが原因であると考えられる。

分散化オーバーヘッドが大きいことに対しては先行制御方式に適したデータ構造の検討が必要である。すなわち，現在実験システムのデータ構造は1プロセッサシミュレータのものを基本にしているため，履歴保存，キャンセルなどを行う先行制御には適さないと考えられる。図11にオーバーヘッドを10%減少させた場合の処理能力特性を示す。図より分散化オーバーヘッドが処理能力の傾きに大きく影響を及ぼすことがわかる。

無制限の先行による無効処理量の増

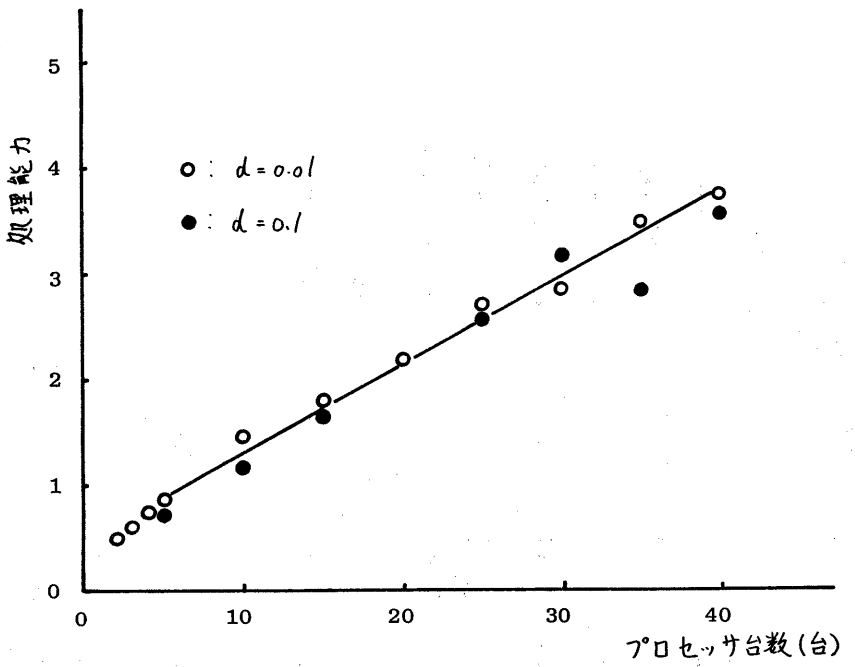


図8　処理能力特性

大に対しては適当に先行の規制を行うことにより処理能力の改善ができると考えられる。規制先行制御方式は，適当なアルゴリズムにより先行規制時刻を決定し，その時刻を越えての処理の先行は行わず，他の時刻に関係しない処理を行いながら他のプロセッサの処理の進行を待ち，プロセッサの並行度を低下させずにキャンセルの発生を抑えようとする方式である。

図12は確定時刻にある一定値を加えたものを規制時刻とする場合の処理能力特性である。プロセッサ数が大きい場合処理能力は向上しているがプロセッサ数が小さい場合逆に処理能力が低下している。これは，規制時間幅に最適値が存在するためで，規制が強過ぎる場合はプロセッサが遊休状態になることが多いからであると考えられる。

次にプロセッサの相互関連の強さが処理能力に及ぼす影響について述べる。

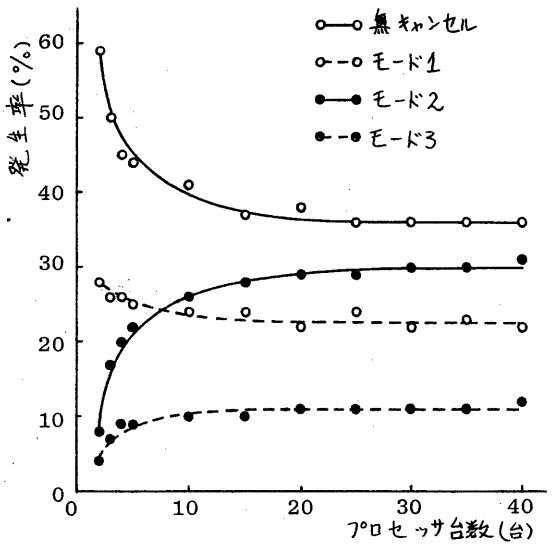


図9．到着メッセージの要求処理

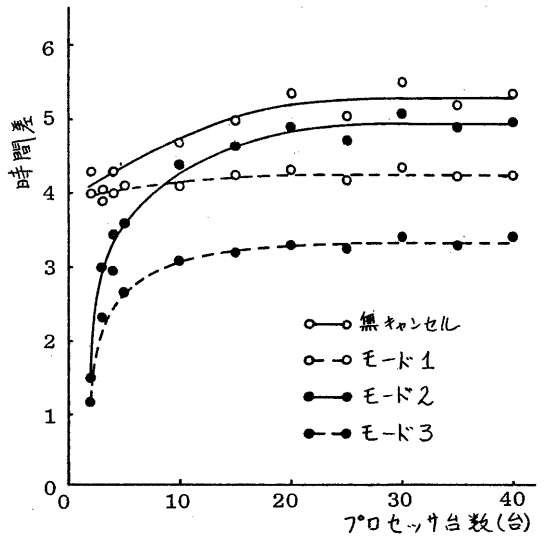


図10．到着時刻との時間差

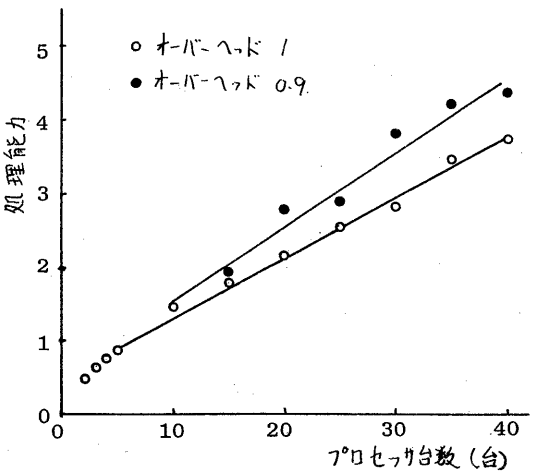


図11．処理能力特性

図13はプロセッサ数 $n=40$ で通信を伴う事象の発生率 r を横軸とした場合の処理の有効率である。図より処理の有効率は r に対して指数関数的に減少することがわかる。また，D-SSQのシミュレーションでは通信を伴う事象の発生率は0.5であるので処理の有効率はおよそ20％となる。

図14はプロセッサがメッセージを受信した際の各処理の発生割合を示す。キャンセルが発生しない場合とモード3のキャンセルは r に無関係に一定となる。モード1のキャンセルは r の増加に伴って減少するがモード2のキャンセルは逆の傾向を示す。すなわち，r の増加と共にメッセージを受け取った際に矛盾が発生する確率は減少するが，一度キャンセル処理が始まると，結果としてモード2のキャンセルを引き起こすキャンセル要求が多く出されることがわかる。

以上より先行制御においてはプロセッサ間通信量を最少とするような分散化を行うべきであることがわかる。

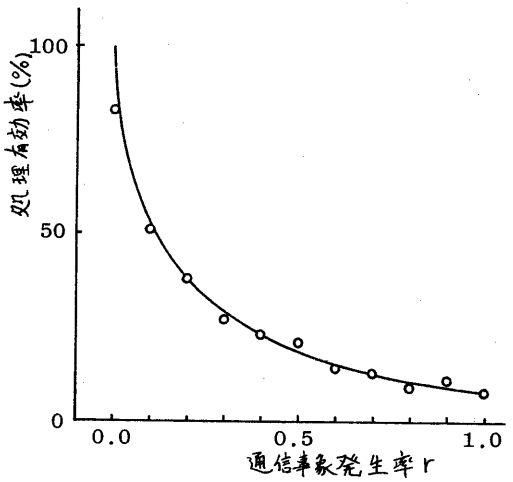


図13. 処理有効率

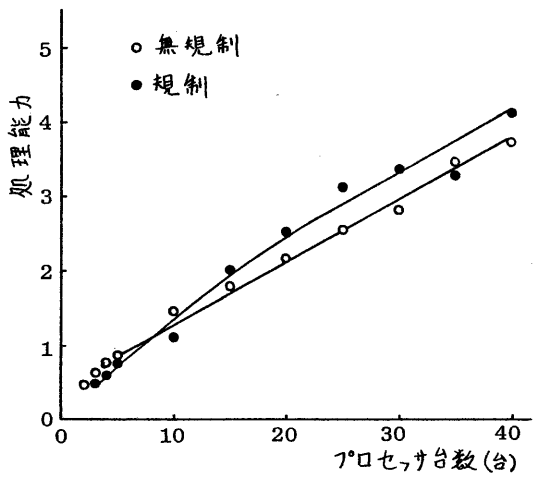


図12. 処理能力特性

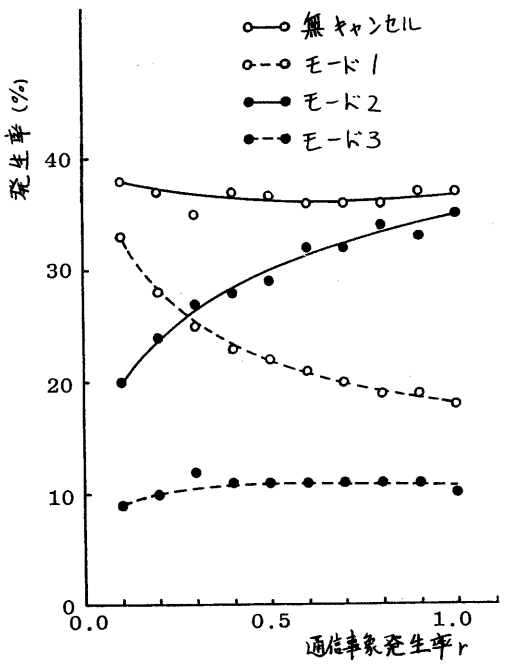


図14. 到着メッセージの要求処理

6. まとめ

実行制御方式として先行制御方式を採る並行処理事象型待ち行列網シミュレータD-SSQの処理能力を実験システムと大型計算機シミュレーションを用いて検討を行った。その結果，処理能力はプロセッサ台数に対して線形に増加すること，先行制御量が過大であるため無効処理量が大きいことと分散化オーバーヘッドがかなり大きいことから処理能力の絶対値は大きくないこと，先行規制や分散化オーバーヘッドの減少により処理能力の改善が図れることを示した。

今後，先行規制方式のより詳細な検討，先行制御に適するデータ構造の検討，先行制御方式の有効範囲の考察，D-SSQの実用レベルシステムの構成が課題として残されている。

[参考文献]

(1) J.K. Peacock, J.W. Wong, and E.G. Manning "Distributed Simulation Using a Network of Microprocessors", Computer networks 3, 1979

(2) 舘森，清水他 "複合マイクロプロセッサによる並列処理形通信網シミュレータ" 信学技報 EC79-78

(3) 中川，長谷川，他 "待ち行列システムシミュレーションにおける並列処理情処，計算機アーキテクチャ 34-38

(4) 渡辺，佐藤，中西，真田，千塚，"並行処理事象型待ち行列網シミュレータD-SSQについて"，信学技報 CS83-102