

## 待ち行列網シミュレータ D-SSQ について

大阪大学 工学部

佐藤 圭

(Kei Sato)

中西 暉

(Hikaru Nakanishi)

真田 英彦

(Hidehiko Sanada)

千塚 慶一

(Keiichi Tezuka)

## 1. まえがき

待ち行列網シミュレーションは待ち行列網問題を解く有効な手段であり、GPSS, SSQ<sup>(1)</sup>等の大形計算機上で実行されるシミュレータが知られている。しかしながら待ち行列網シミュレーションは、多くのメモリと指数的に増大する実行時間を必要とするため、網規模やサンプル数に大きな制約を受ける。特に大規模網のシミュレーションは、従来の方法ではほとんど実行不可能である。

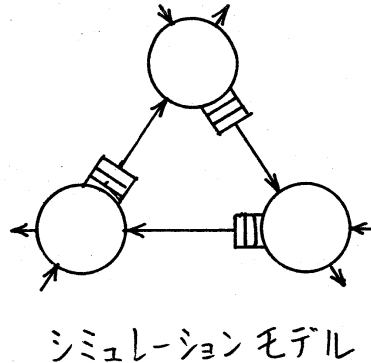
このため、待ち行列網が本来有する並列性に着目してジョブを分割し、マイクロプロセッサに割り当てて並行処理を行う分散処理型シミュレータの研究が各所で行われている。<sup>(2)~(6)</sup>

分散処理型シミュレータにおける重要な問題には、プロセッサ間結合方式、時刻同期方式があり、いずれもシミュレー

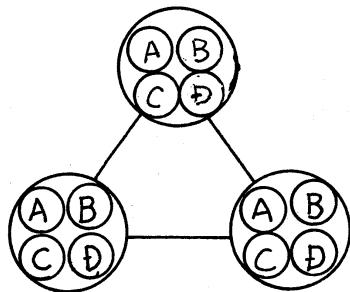
々の効率を左右する重要な問題である。D-SSQは、SSQをもとに分散処理を行う待ち行列網シミュレータであり、時刻同期方式として先行制御方式を用いている。本稿では、D-SSQの構成について述べる。

## 2. シミュレータの分散化

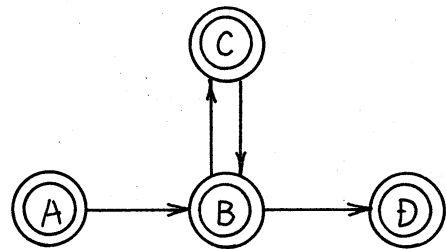
待ち行列網シミュレーションを分散処理する場合、待ち行列網の各ノードが並列動作することに着目し、ノードをプロセッサに割り当てるノード分散方式と、事象が並列動作することに着目し、事象をプロセッサに割り当てるプロセス分散



- Ⓐ : 発生
- Ⓑ : 到着
- Ⓒ : 待ち
- Ⓓ : 退去



ノード分散



プロセス分散

図1 ノード分散とプロセス分散

方式が考えられる。図1にこれらの概念図を示す。さらに、ノード分散とプロセス分散を共に適用する複合分散方式も考えられる。D-SSQにおいては、待ち行列網が本来有する並列性を十分に活用する方法を開発することを目的とするため、ノード分散方式を採る。

### 3. 先行制御方式

D-SSQの時刻同期方式として先行制御方式を用いる。先行制御方式は、個々のプロセッサが他のプロセッサの制限を受けずに処理を先行して行い、プロセッサ間通信の際にシミュレーションパケットの到着すべき時刻よりも受信プロセッサの時刻が進んでいるという時刻矛盾を検出すると、シミュレーションパケットの到着すべき時刻以後に行った処理をキャンセルし、シミュレーションパケットの到着すべき時刻まで受信プロセッサの時刻を戻す方式である。待ち行列網シミュレーションにおいては、処理順序がシミュレーション結果に影響を及ぼす事象の頻度は比較的少ないと考えられるので、先行制御方式を用いることにより、ノード間の並列性を十分に開発できると考えられる。

しかしながら、無制限に先行処理を行うと、プロセッサの並行度は最大となるが、明らかにキャンセルが発生すると予

相される場合でも処理を行うため、キャンセルが頻発し無効処理量が大きく、処理効率は悪くなる。一方、先行に規制を加えると、キャンセルの発生は減少するが、プロセッサの空き時間が生じるため不利である。そこで、この空き時間に統計量収集等の処理を行うことによりプロセッサの並行度を低下させずに先行を適度に規制する方式が考えられる。これを規制先行制御方式という。

D-SSQにおいて先行制御を行うために次のような機能をSSQに付加している。

- (i) プロセッサのシミュレーション履歴の保存
- (ii) トランザクションのキャンセル
- (iii) 確定時刻検出
- (iv) シミュレーション履歴からの統計量収集

以下にこれらの機能について述べる。

#### 4. シミュレーション履歴の保存

先行制御を行う場合にトランザクションに対して施した処理をキャンセルし、プロセッサの時刻を戻し、キャンセル発生時刻のプロセッサ状態を復元する必要がある。D-SSQにおいてはプロセッサの状態を復元するためにシミュレーション履歴を必要とする。すなわち、キャンセル発生時のシミ

ュレーション時刻のノード状態とキャンセルにより戻るキャンセル発生時刻からキャンセル発生時のシミュレーション時刻までのノードの状態変化が得られれば、キャンセル発生時刻のノード状態が復元できるからである。

SSQとD-SSQのシミュレーションの実行過程を図2に示す。SSQにおいては未処理のトランザクションが時刻順に並べられたタイミングチェーンが構成されており、最も先に処理されるトランザクションがタイミングチェーンの先頭にあり、このトランザクションがタイミングチェーンから切断され処理される。一方、D-SSQにおいては履歴トランザクションと未処理トランザクションがタイミングチェーンを構成し、未処理トランザクションの先頭のトランザクシ

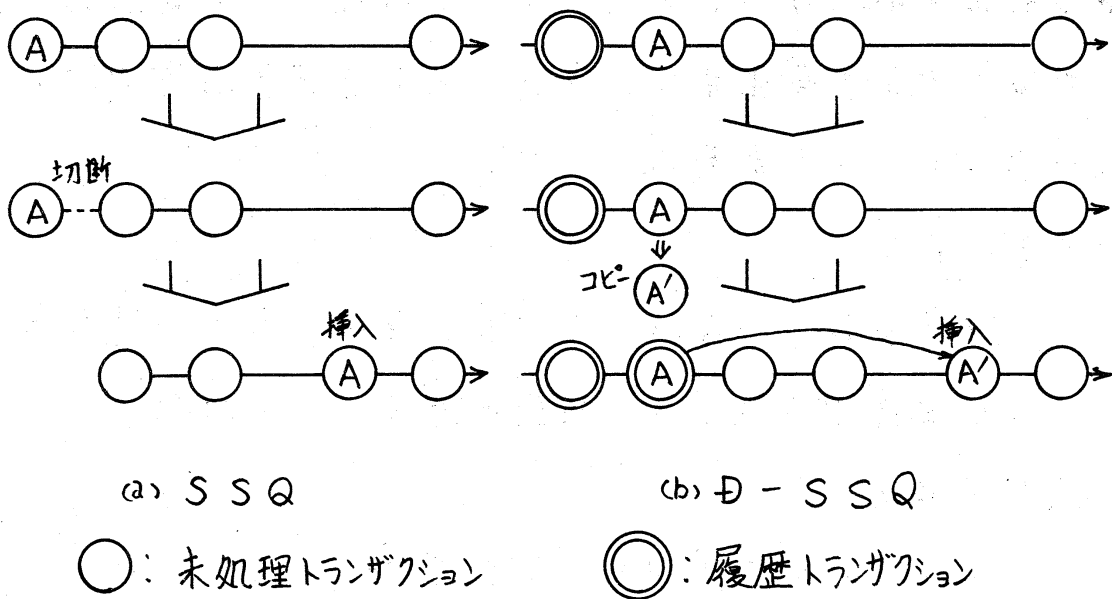


図2. シミュレーションの実行過程

オンがコピーされ、そのコピーが処理される。もとのトランザクションはコピーが受けた処理内容を記入され、シミュレーション履歴となる。さらに、コピーはもとのトランザクションと同一のメッセージであることを示すパケットポインタにより結合される。

## 5. キャンセル

キャンセルはプロセッサ間通信の際にシミュレーションパケットの到着時刻よりも受信プロセッサの時刻が進んでいる場合に発生するだけでなく、キャンセルを行っている際に他プロセッサへシミュレーションパケットを送信したという履歴がある場合、送出したパケットのキャンセルを要求するキャンセル要求パケット送出され、このパケット受信によってもキャンセルが発生する。キャンセルは発生の原因等により次の3つのモードが考えられる。

### ◦モード1

シミュレーションパケットを受信し、その到着時刻よりも受信プロセッサの時刻が進んでいる場合。

### ◦モード2

キャンセル要求パケットを受信し、キャンセル対象のトランザクションが未処理の場合。

・モード3

キャンセル要求パケットを受信し、キャンセル対象のトランザクションが既に処理されている場合。

図3にモード1のキャンセルの例を示す。同図(a)はシミュレーションパケット到着直前の状態、(b)はキャンセル前処理の終了した状態、(c)はキャンセル終了後の状態である。

(a)のようにA・B・C・D・Eの5個のトランザクションがタイミングチェーン上にあり、A・B・Cが既処理、D・Eが未処理である場合にAとBの間の時刻にパケット $\alpha$ が到着すると、 $\alpha$ がタイミングチェーンに挿入され、 $\alpha$ 直後のBにキャンセルポイントが設定される(b)。この後、キャンセルポイントが時刻順にトランザクションをスキャンし、対象となるトランザク

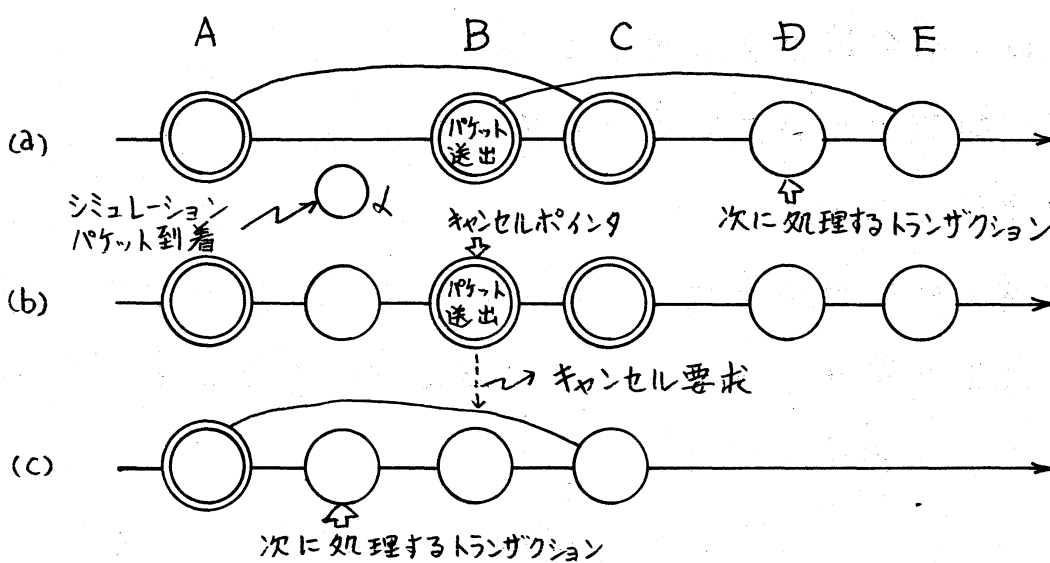


図3. モード1のキャンセル

ションがなくなるとキャンセルを終了する。(c)では、BとCが未処理の状態に戻され、DとEが消去されている。また、Bはパケット送出手の履歴であるのでキャンセルの際にキャンセル要求が送出される。

## 6. 確定時刻検出と統計量収集

確定時刻とは絶対にキャンセルの起さない時刻であり、例えば、各々のプロセッサのシミュレーション時刻の最小値である。確定時刻検出の目的は、規制先行制御を行う際の先行の程度を知るための情報を得ることと、シミュレーション履歴から統計収集しトランザクションを開放するための情報を得ることである。特に前者はシミュレータの効率に大きな影響を及ぼすため、きめ細かい規制先行制御を行うためには効率よい確定時刻検出が必要である。

確定時刻検出には次の方法が考えられる。

### (i) 集中法 (図4)

確定時刻検出専用のプロセッサを設け、プロセッサ間通信を監視することにより確定時刻を検出する方法

### (ii) ループ2周法 (図5)

シミュレータ上の全プロセッサで論理リンクを構成し確定時刻を必要とするプロセッサがシミュレーション



時刻観測要求を論理リング上に1周させる。その後、  
 確定時刻検出パケットを1周させて確定時刻を求める  
 方法。

集中法は、プロセッサ間通信のみに注目しているため、き  
 め細かい確定時刻の検出が行えないが、通信量が増加しない  
 こと、集中管理型のバス形態で実現し易いことが利点である。  
 一方、リング2周法は、完全な分散形であるが、確定時刻  
 検出のための通信が必要である。また、網規模が大きくなる  
 とループ1周の時間が大きくなり、確定時刻が各々のプロセ

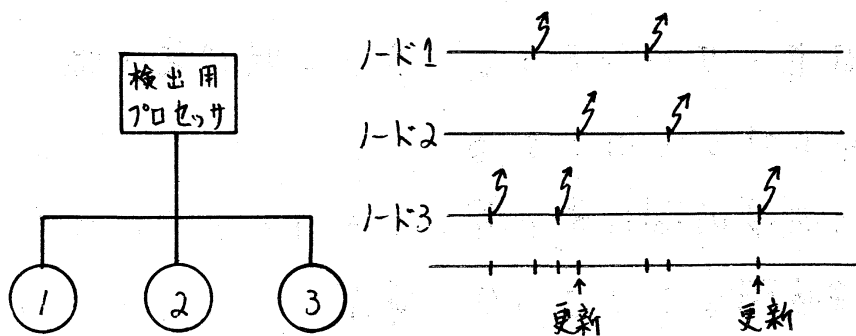


図4. 集中法

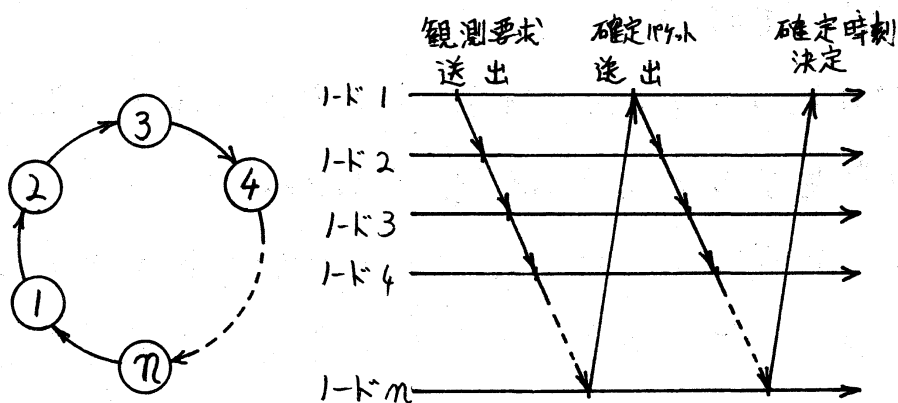


図5. リング2周法

ッサのシミュレーション時刻の最小値(確定時刻の上限値)よりかなり小さな値となる。規制先行制御においては、確定時刻は先行の程度を知る重要な情報であるが、網規模が大きくなれば確定時刻の正確な値の検出が困難となるため、隣接プロセッサの情報等の局所的な情報の利用も必要である。

## 7. 先行制御方式の効率の定量的検討

先行制御方式の効率を評価するための簡単なシミュレーションを行った。以下に仮定を示す。

- (i) 網形態は  $n$  局環状網とする。
- (ii) 各プロセッサのシミュレーション時刻更新幅は  $1/\lambda$  の指数分布に従うものとする。
- (iii) 事象の処理時間は全て等しいものとする。以後事象1個の処理時間を1クロックと呼ぶ。
- (iv) プロセッサ  $i$  からプロセッサ  $j$  への通信は通信確率  $p_{ij}$  に従って行なう。
- (v) プロセッサ  $i$  からプロセッサ  $j$  への通信の際に時刻矛盾が発生すると  $L = \text{int}((t_j - t_i) * \lambda * \alpha)$  クロックの間キャンセル処理としてプロセッサの時刻更新を停止する。
- (vi) プロセッサ  $i$  でキャンセルが発生した場合、キャンセル発生時刻以後に他プロセッサ送信が行われていると、受

信プロセッサにおいてもキャンセルが発生する。

(iii) 通信装置による待ち行列はないものとする。

図6に通信確率  $r=0.4$  における処理能力特性を示す。処理能力はプロセッサ数に対して線形に増加することがわかる。これは、通信装置による待ち行列が発生しないと仮定しているためであり、実際のシステムではプロセッサ数の増加に連れて通信処理能力が飽和すると考えられる。通信オーバーヘッドはプロセッサが送受信を行う際に要求される処理及び待ちによる遊休期間である。通信オーバーヘッドが増加すると、処理能力は大きく劣化する。通信オーバーヘッドは分散化によるオーバーヘッドであり、これを小さく抑える必要がある。

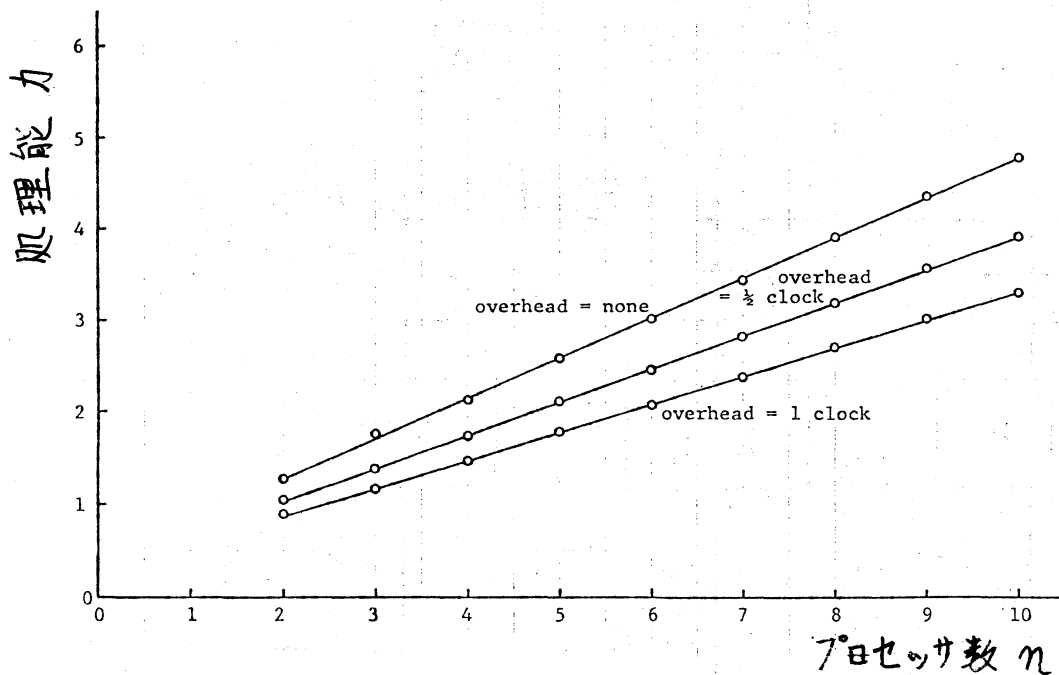


図6. 処理能力特性

## 8. 実験システムの構成

実験システムはD-SSQの確定時刻検出アルゴリズム、規制先行制御アルゴリズム等の開発を目的として構成したもので、ハードウェア構成を図7、ソフトウェア構成を図8に示す。マスタープロセッサ(MP)は統計量の収集、編集、入出力、バスのコントロールを行う。ノードプロセッサ(NP)は待ち行列網のノードの疑似動作を行う。

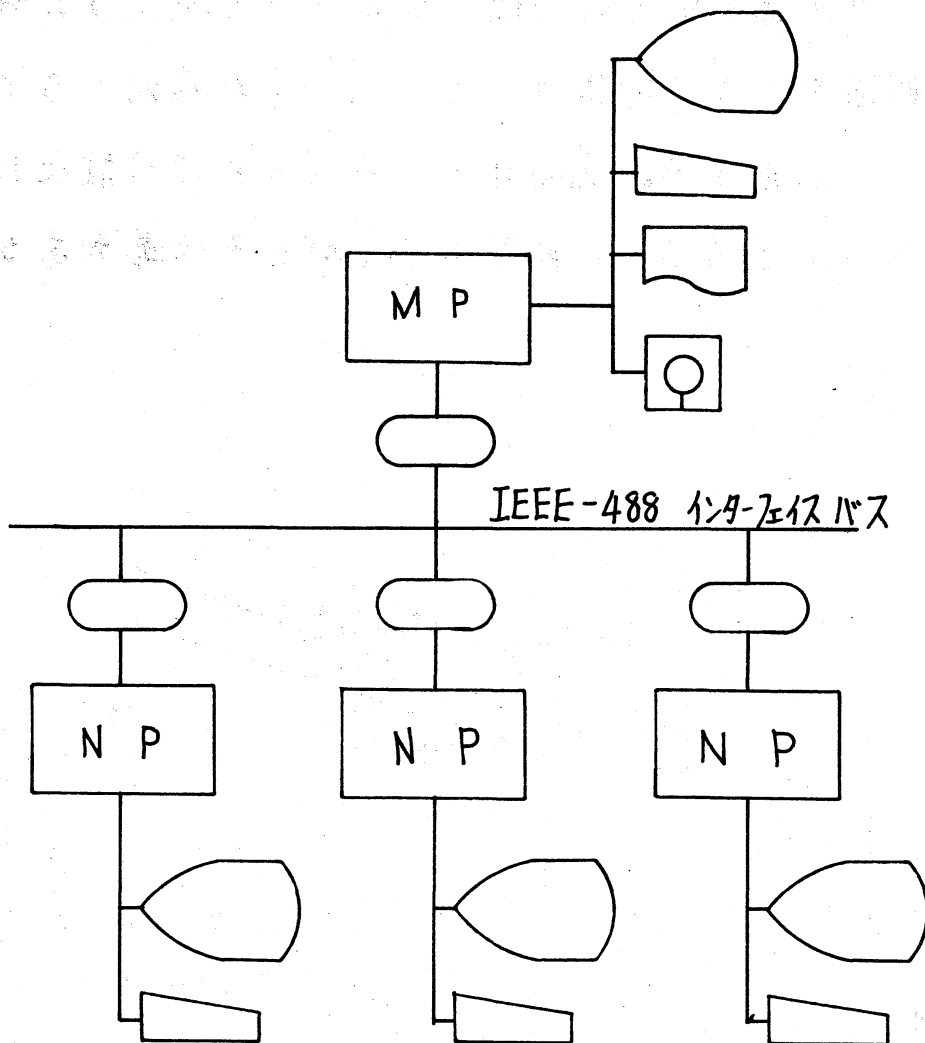


図7. ハードウェア構成

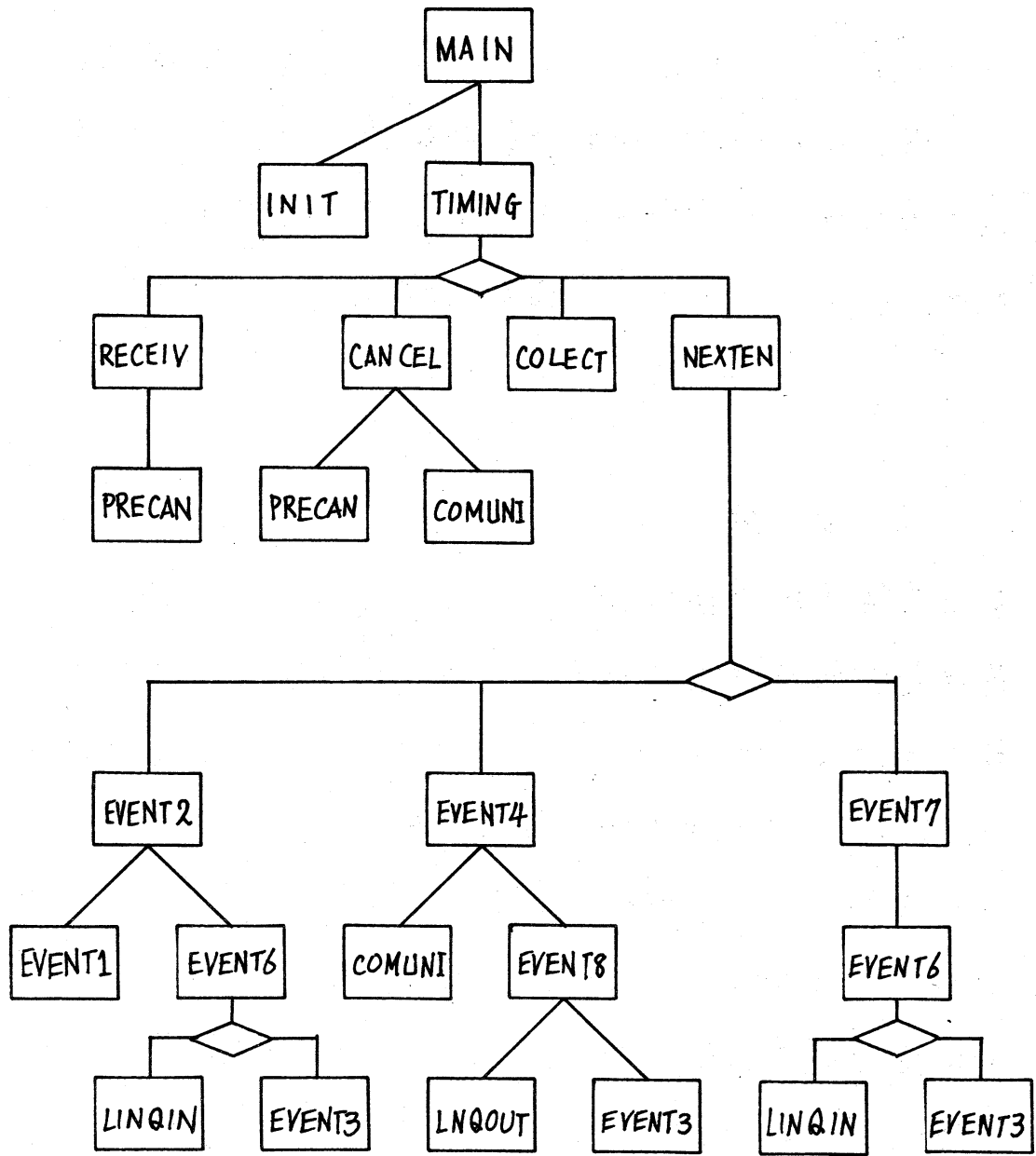


図8. ソフトウェア構成

## 9. まとめ

本稿では先行制御を許すノード分散事象型待ち行列網シミュレータHSSQについて述べた。今後、実験システムを用いて確定時刻検出、規制先行制御等のアルゴリズムを開発し、その評価を行い、それをもとに16bitマイクロプロセッサを用いた実用システムの構成を行う予定である。

## 参考文献

- (1) 真田 “待ち行列網シミュレータHSSQとその成果について” 数理解析研究所講究録452, 待ち行列理論とその応用 '82.2
- (2) 稲森, 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” 電子計算機研究会資料 EC79-78
- (3) 中川, 長谷川他 “待ち行列網システムシミュレーションにおける並列処理” 計算機アーキテクチャ研究会34-3 1979.
- (4) 西田, 宮原他 “待ち行列網シミュレータHASS-QN” 電子通信学会研究会資料 EC80-52
- (5) J.K. Peacock et al. “Distributed simulation Using a Network of Processors” Computer Networks 3(1979) P44~56
- (6) 中川 “待ち行列シミュレーションの並列処理” 情報理論とその応用研究会資料 1981