リング型ローカルエリアネットワークにおける
高速化と信頼性向上に関する研究

１９９２年３月

不　破　　泰

目次

序論

大学には，多くの建物に分れた各研究施設に，大型コンピュータやスーパーコンピュータから，ミニコンピュータ，ワークステーション，パーソナルコンピュータまで，多くのコンピュータが設置されている．これらをネットワークで結び，各コンピュータの資源やデータを相互に利用しあい，有効利用を図りたいという要求は大きい．また，電子メール等のネットワーク構築によって生まれる新しいメディアの有効性も大きい．

大学内のネットワークは，1970年代に大学内の計算機センタのコンピュータ上で時分割処理システム(TSS)を動かし，各研究室のTSS端末とセンタのコンピュータとを電話回線を用いて接続することから始まった．続いて，1980年代に各建物内に多くのワークステーションが設置されると，同軸ケーブルを用いたバス型LANの構築が行われるようになった．そして，1984年頃から米国の大学を中心に，この各建物内のLANを相互に接続することで大学全体のコンピュータを接続した，キャンパスネットワークの構築が開始された[MCCR82] [MATU89] [TOKU84].

マサチューセッツ工科大学では，1984年から各LANを結ぶバックボーンネットワークの構築が始まり，全学のネットワーク化に着手している[CLAR82]．スタンフォード大学でも，1984年にSUNetと呼ぶキャンパスネットワーク構築を開始している．カーネギーメロン大学では，1985年よりAndrewと呼ぶ大規模なネットワーク構築を行っている[MORIS86].

日本においても，1980年代後半からいくつかの大学でキャンパスネットワークの構築が行われている．東北大学では，1987年より100Mbpsの光ケーブルによるリング型FDDIネットワークを用いて，各建物のLANを接続するTAINSと呼ぶネットワークを構築している[SAKA90]．京都大学では，デジタル交換機と基幹光ループを用いたネットワークを1987年より構築している[KANA90]．北海道大学でも，1989年より，HINESと呼ぶFDDIを用いたネットワークを構築している[TANA87].

信州大学におけるキャンパスネットワーク(S-net)は，筆者らによって1984年から構築を開始した[FUWA87] [FUWA88b]．最初は研究室内の小規模な実験用ネットワークであったS-netも，現在では2つのキャンパスにまたがり，接続しているコンピュータも450台以上の規模となり，大学内コンピュータの有効利用に貢献している[FUWA91b]．このS-netの構築と7年間におよぶ運用管理，そして利用者に対

するアンケート調査の結果，ネットワークにおいては，その高速化と信頼性向上に対する要求が強いことが明らかとなった．

S-netは，各ノードをリング状に接続するリング型ローカルエリアネットワークである．このリング型ローカルエリアネットワークは，LAN同士を接続する基幹LANとして多く用いられている．本論文は，このリング型ローカルエリアネットワークにおける高速化と信頼性向上に関し，研究を行ったものである．

本論文は，大きく3つに分れる．第1部では，S-netの構成と各機器の設計，実際の運用等について述べる．第2部では，リング型ローカルエリアネットワークの高速化に最適なMACプロトコル(Media access control protocol)についての検討を行う．また新たなプロトコルの提案も行う．第3部では，リング型ローカルエリアネットワークの信頼性向上のための2つの提案を行う．

第1部は，第1章から成る．初めに，S-netの構成と機能について説明する．大学施設は多くの建物に分散し，各建物内に多くのコンピュータが設置されている．S-netは，この大学のコンピュータ環境に合わせた構成としなければならない．また，研究用ネットワークでの使用実験を通して決定した，S-netの機能についても述べる．

次に，S-netの各機器の設計について述べる．S-netでは，ネットワークにRS-232cインターフェースを持つコンピュータを接続するターミナルサーバを開発し，これを用いて多くのコンピュータを建物ごとネットワークに接続する．また，各建物間を接続するゲートも必要である．更に，ネットワーク全体を制御し，各コンピュータに必要な機能を提供するコントローラも必要である．ここでは，これら各機器の設計について述べる．各機器は同時に幾つもの回線からのデータを処理しなければならず，リアルタイム処理と並行処理が必要となるが，このために設計したハードウェアと，ソフトウェアによる簡易な並行処理実現手法についても述べる．そして，実際に製作した機器の能力を調べる．

続いて，S-netの現状について述べる．現在，S-netを信州大学の２つのキャンパスに敷設し，各キャンパスのS-netを，マイクロ回線で接続している．S-netに接続されているコンピュータは，大型計算機からパーソナルコンピュータまで様々であり，台数は２つのキャンパス合わせて400台以上になる．1990年１年間のS-net利用者数は，延べ48934人と多く，ここでは月別，時間別の利用者数等から，大学におけるネットワークの需要を明らかにする．

最後に，利用者に対して行ったアンケートから，今後のキャンパスネットワークに対する利用者の要望を明らかにする．ここでは，利用者の要望の大半は，ネットワークの高速化と信頼性向上の２つに分けられることを示す．

第2部は，第2〜7章から成る．ここでは，高速なネットワークに最適なリング型ローカルエリアネットワークのMACプロトコルについて検討する．

ネットワークの高速化を図るためには，伝送路を高速化するだけでなく，各ノードの処理も高速化する必要がある．伝送速度が100Mbps以上の高速通信では，このノードの処理がボトルネックとなるため，広帯域ISDNで検討されているATM網のように，ノードでの処理を全てハードウェア化する必要がある．このため，MACプロトコルを，ハードウェア処理に適したものとしなければならない．このことをふまえ，第2部では伝送効率の良いレジスタ挿入リングにハードウェア処理が容易であるスロットリングの考えをとりいれたレジスタ挿入型スロットリング(Register-Insertion Type Slotted Ring: RISリング)を提案する．

プロトコルの検討を行う場合，このプロトコルが，基本的にどの様な性能を示し，いかなる特徴を有するかを，定量的に評価しておくことが重要である．さらに，従来から用いられてきている様々なMACプロコトルの性能も同様に評価し，様々な利用環境下に対して，どのMACプロトコルが最適であるのかを明らかにする必要がある．

第2部では，伝送速度が高速な場合において，与えれれた利用環境下で最適なMACプロトコルを明らかにするため，各MACプロトコルの性能解析を行い，その結果を比較する．解析するMACプロトコルはRISリングに加え，従来から用いられている代表的なトークンリング，スロットリング，レジスタ挿入リングの4つである．

従来より，リング型ローカルエリアネットワークのMACプロトコルの性能解析に関する多くの研究が行われてきた．しかし，伝送速度が高速な場合の性能解析は，あまり行れていない．しかも，従来の研究の多くは，各プロトコルごとにモデル化の仮定や解析手法が異なっている．このため，高速ネットワークにおいて同じ条件のもとでの各プロトコルの解析結果を比較して，与えれれた環境のもとではどのプロトコルが最適であるかを調べることは困難である．また，これまでの研究の大半は，解析を簡単にするためネットワークの各装置内のバッファの容量を無限大としている．このため，ネットワークの大切な特性であるバッファのオーバ

フローに関する議論が出来ない．ここでは，リング型ローカルエリアネットワークの各MACプロトコルを，同一の仮定のもとでモデル化する．この際，バッファの容量は有限とした．この結果，各モデルは複雑となり，解析は困難となったが，平衡点解析と呼ぶ手法により解析が可能となった．

第2章では，各MACプロトコルを解析する際の共通の仮定について述べる．

第3章では，提案するRISリングを説明し，これをモデル化してその性能を明らかにする．このモデル化に際しては，RISリングをその動作から2つの場合に分け，各々についてリング中の全ノードの動作を表すマルコフモデルを作成し，最後にそれらを統合する等の工夫を行う．

第4章では，トークンリングの性能解析を行う．解析に際しては，トークンの転送と，トークン受信を待ってパケットの送信を行う各ノードの動作を表したモデルを作成する．

第5章では，スロットリングの性能解析を行う．モデル化に際しては，第3章のRISリングのモデル化と同様に，その動作からスロットリングを2つの場合に分類し，その各々をモデル化して最後に統合する．

第6章では，レジスタ挿入リングの性能解析を行う．レジスタ挿入リングは，他のプロトコルと異なり，各ノードが独自のタイミングでパケットの送信等を行う．そのため，第3〜5章の解析のようにリング中の全ノードの動作を表したモデルの作成は困難である．そこで，レジスタ挿入リングでは各ノードごとに独立したモデルを作成する．さらに，各ノードのモデルも，ノード内の非同期で動作する各モジュールを表す複数のサブモデルからなる構成とし，モデルの簡単化を計る．また，レジスタ挿入リングでは，各ノード内の処理が複雑なため，他のプロトコルのように処理全てをハードウェア化するのではなく，ソフトウェアにより実現することが現実的である．そこで，モデル化に当たっては処理のオーバヘッドを考慮する．また，解析結果の数値例を求めるときは，S-net で用いているノードの実際の処理時間からこのオーバヘッドを求め，現実に沿った性能解析を行う．

第3〜6章の各MACプロトコルのモデルの解析に当っては，総て平衡点解析の手法を用いる．この手法は，既に衛星ネットワークやバス型LANにおける様々なプロトコルの解析に成功している．本論文は，この手法の適用範囲をリング型ローカルエリアネットワークへ拡張するものである．また，解析結果はシミュレーショ

ンによる数値例と比較することで，その精度についても検討する．

第7章では，第2部のまとめとして，高速ネットワークの様々な条件下で各MACプロトコルの解析結果を互いに比較し，各プロトコルの特性について議論する．そして，各プロトコルはどのような条件下で性能が最も優れているかという，プロトコルの適用領域を明らかにする．この際，ノード内バッファのオーバフローを考慮し，オーバフローしてパケットが失われる確率が一定値以下で実用に耐える条件下で適用領域を求める．

第3部は，第8,9章から成る．ここでは，リング型ローカルエリアネットワークの信頼性向上のための手法を，2つ提案する．一つは，ネットワークに接続された各装置の異常を各装置内のCPUが早期に発見して自動的に復旧させるもので(第8章)，もう一つは，このCPU自体の異常を発見して復旧させるものである(第9章)．

第8章で提案する各装置の異常発見と復旧手法は，各装置が定期的にチェック用のパケットを送り合う事で行う．本手法では，各装置にてこのパケットの受信状態と各装置がリング状に接続されているということから，自己の異常を検出するアルゴリズムを提案する．また，異常検出後はこれを復旧し，その後異常情報を任意の装置に報告する機能の提案も行う．本手法は，リング型ネットワーク一般に適用でき，付加するハードウェアが少なく，専用の管理システムを必要としない等の特長を持っている．

第9章では，従来より用いられているウォッチドッグタイマの暴走検出能力を高める方法について提案する．まず，ウォッチドッグタイマのCPU暴走を検出する能力を解析する手法を確立する．これは，ウォッチドッグタイマを品質管理で用いられている抜取り検査の一種とみなし，統計的に解析するものである．解析例として，S-netの各装置で用いられているザイログ社製Z80CPUの暴走時の振舞いを表す状態遷移図を作成し，従来のウォッチドッグタイマの暴走検出能力を明らかにする．続いて，この解析をもとに，暴走検出能力が更に高いウォッチドッグタイマの構成方法を提案するとともに，その暴走検出能力を明らかにする．

第1部
キャンパスネットワーク S-net

第1章 信州大学キャンパスネットワーク S-netの構築と評価

1.1 序

大学においては，キャンパス内あるいはキャンパス間における計算機の分散は顕著であり，それらをネットワークで結び有効に利用したいという要求は大きい．

大学におけるネットワークは，最初計算機センタの大型計算機を構内の電話回線とモデム装置により利用する形態から始まった．続いて各建物内のコンピュータをLANで結ぶ小規模なネットワークが構築された．そして，現在では大学全体のコンピュータを接続できるキャンパスネットワークの構築が，いくつかの大学で行われるようになった[MCCR82] [MATU89] [SAKA90].

カーネギーメロン大学では，学内に数千台のワークステーションを設置し，50〜100台のワークステーションごとに1つのクラスタを作りネットワークで結び，更に各クラスタ間をリング状に接続するという大規模なキャンパスネットワークを構築中である[TOKU84].このカーネギーメロン大学のネットワークでは，接続するワークステーションは全てオペレーティングシステム(OS)としてUNIXを用い，ネットワークにおけるプロトコルとしてTCP/IP方式を用いている．またカーネギーメロン大学のコンピュータサイエンス学科では，多数のワークステーションや計算機が，バス型でMACプロトコルがCSMA/CD方式，上位プロトコルがTCP/IPであるネットワークを用いて接続されている．このネットワークにおいても接続する計算機のOSは全てUNIXである．

しかし，ネットワークに接続したい機器が全て1つのOSに統一されていたり，同一のプロトコルを持っているとは限らない．実際，信州大学においてキャンパスネットワークを考えたとき，接続したい機器は，大学の計算機センタにある汎用大

型計算機，各学部にある様々なOSの計算機，各研究室にあるワークステーションやパーソナルコンピュータ，更に外部ポートを持った計測器等であり，1つのOSに統一することは不可能であった．また，各機器ごとにゲートウェイを開発し，統一したプロトコルでネットワーク化する事も，機器が余りに多く現実的ではなかった．

この様な環境にある大学において有効な新しいキャンパスネットワークを開発し(以下このネットワークをS-netと呼ぶ)，構築し運用したので報告する．

S-netには次の特長がある．

(1) 各建物内の端末や計算機をリング状に接続し，そのリング間を更にリング状に接続する二重のリング構造である．伝送媒体には同軸ケーブル，光ファイバ及びマイクロ波通信を用いる．

(2) 各地域のキャンパスごとにS-netを構築し，このS-netどうしをマイクロ波通信を用いて接続して，各ユーザが任意のキャンパスのS-netを利用できる機能を持つ．

(3) 1台の端末から同時に複数のコンピュータを利用し，互いにデータ交換を行うことができる等様々な計算機の利用形態をサポートする．

(4) パケットのMACプロトコルにリアルタイム性と高負荷時の安定性を図ったレジスタ挿入方式を用る．

(5) 各機器のネットワークへの接続には，ターミナルサーバを開発し，CCITT V.24 (RS-232c)インタフェースを用いる．

第1部では以上の特長を備えたS-netについて述べる．1.2ではS-netの設計について，1.3では実際に製作したS-netの実現方法について述べる．1.4では，このS-netを用いて構築した，信州大学のキャンパスネットワークを紹介する．1.5でこのS-netを評価し，キャンパスネットワークの課題について論じ，1.6で今後のS-netの改良について述べる．

1.2 S-netの設計

1.2.1 キャンパスネットワークに対する要求

筆者はキャンパスネットワークを設計するに当り，ネットワークに対し次のような要求があると考えた．

[1] 大学内の様々なコンピュータや通信機能を持った計測装置と，研究室，教室内の端末とを結び，研究室から任意のコンピュータや計測器を利用したい．

[2] 端末として，既に大学にある様々なパーソナルコンピュータも使用したい．

[3] 複数の計測装置からの測定データを研究室にて収集したり，研究室から装置に指示を送りたい．

[4] 端末から同時に複数のコンピュータへ接続し，個々の計算機を利用したり，互いにデータ交換等を行いたい．

[5] ネットワークの形態が，学部，学科により多くの建物に分れているキャンパスの形態に適したものでありたい．

[6] 接続する機器のトラブルによる，ネットワークへの影響が極力少なくしたい．

1.2.2 S-netの設計

1.2.1で考えたキャンパスネットワークに対する要求を満たすため，次のようにS-netの設計を行った．

[1][2]の要求に対して

既存のネットワークは，例えばバス型でMACプロトコルがCSMA/CDであるEthernet [DEC80] を用いてコンピュータを接続している．Ethernetは比較的簡単な設備で，高速のネットワークが構築できる優れた方式であるが，接続には専用のインタフェースとプロトコルが必要である．S-netを設計した1983年当時，大学の計算センタの大型コンピュータは，通常独自のインタフェースと独自のプロトコルによる通信か，あるいはTSS端末用のRS-232cインタフェースによる通信しかサポートしておらず，ユーザによる新たなインタフェースとプロトコルによる通信は困難であった．

大学に多く設置されているパーソナルコンピュータでは，Ethernetのインタフェースを付け，TCP/IPのプロトコルを用いた通信が可能なものが現れている．しかしまだ高価であり，また全てのパーソナルコンピュータで可能なわけではない．一方，パーソナルコンピュータにはRS-232cインタフェースをサポートし，端末エミュレータのソフトウェアを有するものが多い．

そこでS-netでは，ターミナルサーバを開発して，RS-232cインタフェースによる接続が可能となるようにした．このことにより，大型コンピュータにとって，ネットワークに接続された他の端末等からの使用も，通常のTSS端末からの使用と同じに見えるようにする．この様にして，ネットワーク用にOSを変更する必要を無くした．

RS-232cインタフェースにより接続できるため，外部ポートを持つ計測器も簡単に接続ができる．

[3][4]の要求に対して

既存のネットワークでも，各ノード間で直接パケット交換を行い，遠隔ログイン等のサービスが実現されている．しかし，[3][4]で考えた様々な接続形態の実現には，この様なノード間の直接パケット交換の方式では，パケットの構成やプロトコルが複雑になり，また機能の追加時は，全ノードの処理プログラムを交換しなければならない．

S-netでは，接続した全機器からの入力データをパケットにして1台のコントローラに集め，このコントローラにより様々なサービスを実現することにした．このため，機能の追加はコントローラの処理プログラムを交換するだけですむ．

ただし，このコントローラにデータが集中するため，コントローラでは高速なリアルタイム処理が必要であり，更に多くの接続機器からの要求に同時に答えるため，並行処理も必要である．

[5]の要求に対して

多くの建物に分散するキャンパスの形態に適したネットワークとするため，S-netでは各建物ごとに機器をリングで結び(以下このリングをクラスタリング(Cluster Ring)と呼ぶ)，このクラスタリングどうしを更に大きなリングを用いて接続する(以下このリングをバックボーンリング(Backbone Ring)と呼ぶ) 二重のリング構造とした．これは，カーネギーメロン大学のキャンパスネットワークと同様な考えである．

また，信州大学では長野県各地にキャンパスが点在しているため，各キャンパスごとにS-netを構築し，このS-netどうしをマイクロ波通信を用いて接続し，相互にパケット交換ができる機能を持たせることとした．この機能により，S-netに接続したどの端末からも，任意のS-netを利用できる．

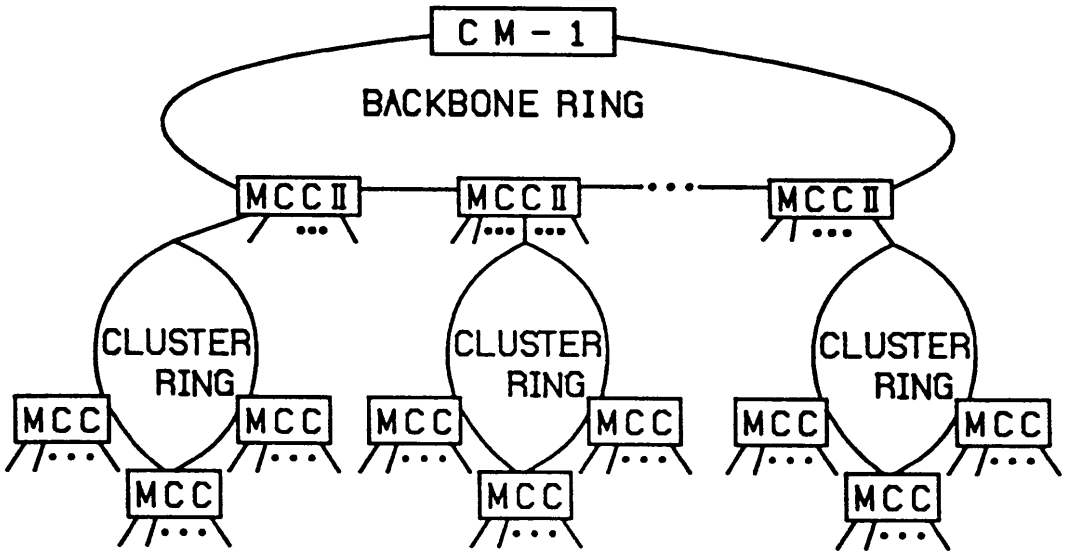


図1.1 S-netのブロック図
Fig.1.1 Block diagram of S-net.

[6]の要求に対して

Ethernetでは，接続する1台の機器のトラブルにより，ネットワーク全体が使用できなくなる事がある．S-netでは，機器はターミナルサーバを用いて接続し，ネットワークの様々なサービスはコントローラが行う．このため，接続する機器のトラブルの影響は，その機器のみか，同じターミナルサーバに接続した機器にしか現れず，ネットワーク全体に影響することはない．

以上をまとめ，S-netの構造は図1.1で示すようにした．図中MCCはクラスタリングにおけるターミナルサーバであり，MCC IIは各クラスタリングと，バックボーンリングとの接続のための機器である．CM-1がコントローラである．端末や大型計算機のチャンネルは全てMCCの下位に接続し，これらの機器からのデータはMCC，MCC IIを経てCM-1に送られる．逆にCM-1からのデータはMCC II，MCCを経て各機器に送られる．さらに図1.2に示すように，複数のS-netのMCCの端末を接続するチャンネルどうし，またはMCC IIのクラスタリングを接続するチャンネルどうしを互いに接続することにより，より大規模なネットワークを構成できるようにした．

このS-netの構成は，建物が各学科ごとに分れているキャンパスの形態に合わせ

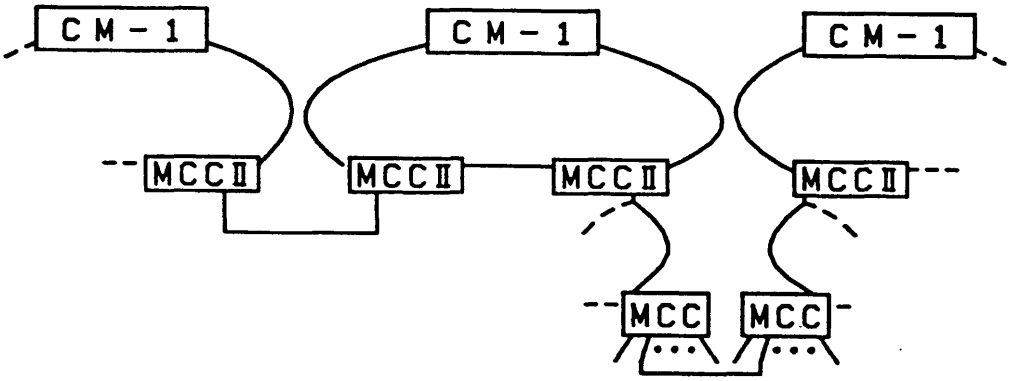


図1.2 S-netどうしの接続
Fig.1.2 S-net to S-net bridge.

たものである．ネットワーク敷設を，建物ごとに独立して行うことができるため，ネットワーク需要の多い建物から順に敷設して利用を開始でき，敷設時に他の建物のネットワークを停止する必要が無いという利点がある．また，キャンパスネットワークは頻繁に接続計算機の増設を行い，ノードの増設を行う．リング型ネットワークではノードの増設時にはリングを切断するため，そのリングの利用ができなくなるが，このリング構成では接続するクラスタリングの利用を停止するだけで，他のクラスタリングの利用は続けられる．

このリング構成の欠点としては，別のクラスタリングに接続された機器間の通信では，間にある中継機器が増加し，中継機器の遅延によりパケットの伝送遅延が大きくなるという問題がある．S-netでは，MCC，MCC IIの中継処理の大半をハードウェアにより行わせることにより，中継機器の遅延を小さくし，伝送遅延の増大を防いだ．

伝送速度は次に示す目標を定めた．

(1) 接続する機器の伝送速度は9600bpsまで対応できるようにする．

(2) クラスタリングは，このリング内の機器が同時に8台まで連続してデータ転送を行うことができるよう76.8Kbpsとする．

(3) バックボーンリングは，データが集中するため，S-net全体で同時に100台以上の機器からのデータを処理できる1Mbpsとする．

また，S-netで実現する機能はとりあえず次に示すものとした．

(1) 2つのチャンネルに接続した2台の機器間で互いにデータの交換を行う機能．従来の遠隔ログイン機能である．

(2) 2つのチャンネルに接続された2台の機器間のデータ交換を，他のチャンネルに接続されたパーソナルコンピュータ等からの指示で行う機能．これは例えばS-net内の2台の計算機間でデータやプログラムの交換を行う際利用する．

(3) S-netに接続された任意の複数の機器からのデータを1つのチャンネルに接続されたパーソナルコンピュータ等に集め，又逆にパーソナルコンピュータから任意の機器にデータを送る機能．これは，S-net内の何台かの測定装置からデータを収集する際等利用する．

(4) メッセージ交換機能．電子メール，電子掲示板を実現する．

以上の各機能は，筆者の研究室において実験的に設置したS-netにおいて，1年間実際に使用しながら利用者の意見を聞き改良が繰り返されたものである．

なお，S-netにはルーチングサービス機能はない．使用者は特定のチャンネルを指定するときは，そのアドレスをコマンド中に直接指定する．他のS-netのチャンネルを指定する場合は，使用者がまず遠隔ログイン機能を用い，使用している端末を他のS-netにログインした後，そのチャンネルを指定する(1.4.2に実行例を示す)．

1.2.3 ネットワークを自作した理由

S-netは，既存の製品を組み合わせて実現するのではなく，新たに各機器を設計開発することで実現した．既存のものを用いなかった理由について述べる．

S-netのアクセスプロトコルとしては，トークンリング，スロットリングと比べ，比較的パケット遅延が小さいことが知られている[STRO87] [LOUC85] レジスタ挿入方式を採用することにした．しかし，S-net敷設を開始した7年前には，既存のものでレジスタ挿入方式を用いたものはほとんど無く，自作するしかなかった．

また，筆者は敷設したネットワークを用いて，利用率，パケット伝送遅延，機器のバッファ内の平均パケット数等の各種データをとり，ネットワークの改良に役立てたいと考えた．この場合，各機器が既成のものでは，詳しいデータがとれない．例えば，パケット伝送遅延を考えると，伝送に大きな影響を与える各機器のソフトウェアのオーバヘッドが明らかでなければ，詳しい解析が出来ない．

価格の問題も大切である．大学の限られた予算の中から，出来るだけ大勢の人が利用できるネットワークを実現するには，自作する方が有利であった．

ただし，自作ではその信頼性に問題が生じる．このため，1.4.3 で述べるようにネットワークの状態を監視して，異常が有れば管理者に知らせる装置を現在使用しているほか，第3部で述べるように信頼性の向上を図る方法についても研究を進めている．

1.3 実現方式

1.2.2で述べた設計に基づき，実際に製作したS-netについて説明する．

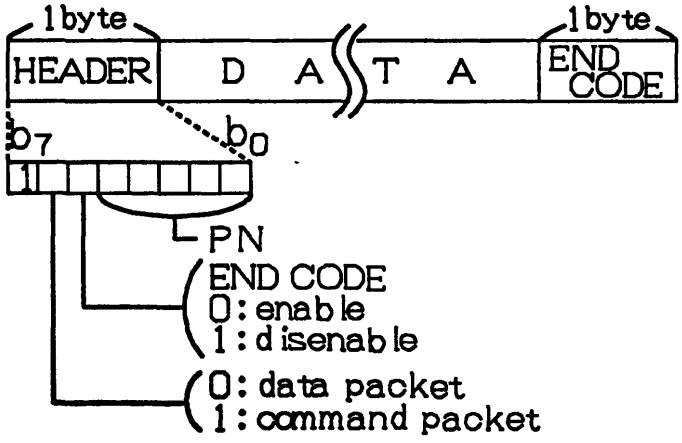


図1.3 クラスタリングのパケット構成
Fig.1.3 Packet format of Cluster ring.

1.3.1 パケット構成

ここでは，S-netで採用したパケットの構成について述べる．図1.3にパケットの構成を示す．パケットは1バイト長のヘッダ部と任意バイト長のデータ部に終端コード1バイト(CRコード:$(0d)_{16}$)を付けた構造とした．各機器とクラスタリングとを接続するターミナルサーバであるMCCは，各機器からのデータをバッファに蓄え，終端コードを受信した時点で，バッファ内のデータをデータ部に置いたパケットをクラスタリングへ出力する．しかし，これでは終端コードを受信するまでいつまでも周辺チャンネルのデータを出力しないことになるので，バッファにデータがありトラフィックが少なく一定時間たっても終端コードを受信しない場合，そこまでのデータをパケットにして出力することにした(現在この時間は1.6msecである)．つまりパケットには，終端コード受信時のパケットでありパケットの終端コードもデータの一部として扱われるものと，終端コードを受信していない時のパケットであり終端コードはデータに含まれないものの2種類がある．パケットのヘッダ部の6ビット目が0の時は終端コードを受信したときのパケットであることを示し，1の時は受信していないときのパケットであることを示す．これは言い換えると，トラフィックが少なくなかなか終端コードを受信しない時は，終端コードまでの真のパケットを複数の仮のパケットに分割することになり，これをメタパケット[NAKA82a]と名付けている．

又周辺チャンネルよりブレーク信号を受信する場合がある．ブレーク信号はコードとして決まっているわけではなく，一定時間以上データ線がアクティブ状態になる事で表すため，パケットのデータ部でこれを表すことができない．そのため，パケットのヘッダ部の7ビット目が1のとき，ブレーク信号を受信したことを表すようにした．

ヘッダ部の下位5ビットはクラスタリングにおいてはそのリング内の機器番号(これを Peripheral channel number:PN と呼ぶ)を示す($PN = 0, 1, \cdots, 31$)．MCCが各機器からの受信データをリングへ出力する時は，データを受信した機器のPNを示す．逆にリングよりMCCがパケットを受信したときは，このパケットのデータ部を出力する機器のPNを示す．

バックボーンリングではヘッダ部の下位5ビットは，そのバックボーンリングに接続されたクラスタリング番号(これを Cluster ring number:CN と呼ぶ) を示す($CN = 0, 1, \cdots, 31$)．バックボーンリングとクラスタリングを接続するMCC IIは，

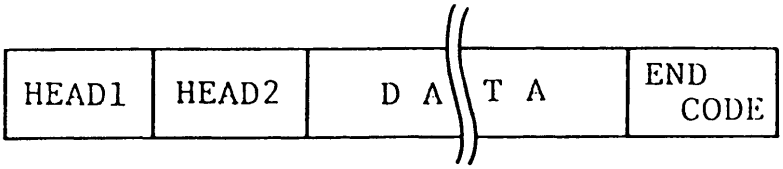


図1.4 バックボーンリングのパケット構成
Fig.1.4 Packet format of Backbone ring.

クラスタリングから入力したパケットの先頭にこのCNを示すヘッダを付け，受信パケットをデータ部に置いたパケットをバックボーンリングへ出力する．このパケットの構成を図1.4に示す．また，バックボーンリングからパケットを受信したMCC IIはそのヘッダ部の下位5ビットが示すクラスタリングへ，データ部と終端コードを出力する．

1つのS-netに接続することができる機器の数は，最大32のクラスタリングごとに最大32台の機器が接続できるため，最大1024台である．

1.3.2 S-netのMACプロトコル

MCCがクラスタリングに対し，又MCC IIがバックボーンリングに対してパケットを送信する際のMACプロトコルとしては，レジスタ挿入方式を用いた．レジスタ挿入方式におけるパケットのリングからの除去は，パケットがリングを1周したときに取除するのではなく，宛先ノードによりリングから取除する方式とした．パケットの送信はトークンや空スロットの受信を待つのではなく随時行われる．その際，隣接ノードから受信した他のノード宛のパケットの送信を，このノードに接続された端末からのパケットの送信(MCC IIの場合は接続されたクラスタリングからのパケットの送信)より常に優先して行う方式とした．

1.3.3 MCC

MCC(Multi-channel Communication Controller)は，各機器をS-netのクラスタリングへ接続するターミナルサーバである．MCCは，クラスタリングに接続した1つのチャンネル(これをホストチャンネルと呼ぶ)と各機器に接続した8つのチャンネル(これを周辺チャンネルと呼ぶ)を持つ．MCCは8つの周辺チャンネルと1つのホストチャンネルからそれぞれ独立に受信するデータを同時に処理しなければならず，リアルタイムで高度な並行処理が要求される．そのため各チャンネルのインタフェースはFIFOメモリを用い，受信データをハードウェアによりFIFOメモリに格納するようにしてソフトウェアの負担を減らした．また，ホストチャンネルのインタフェースは，高速な入出力を可能にするため，簡易プログラマブルアレイロジック(PAL [MONO84]) を利用して設計，制作した．更に制御プログラムは，以前提案した状態遷移法[NAKA82b]と呼ぶ並行処理プログラム設計作成の手法を用い，アセンブリ言語で作成した．

制作したMCCの性能を，すべてのチャンネルが送受信を行うというMCCにとり最も負荷の大きな状態において，データを文字落ちせず送受信できる最高伝送速度により評価した．その結果最高伝送速度はホストチャンネルで81.6Kbps，周辺チャンネルで10.2Kbpsであり，1.2.2で示した目標に達することを確めた．

1.3.4 MCC II

MCC IIはクラスタリングとバックボーンリングとを接続する．構成はMCCと同様に8つの周辺チャンネルと1つのホストチャンネルとを持ち，周辺チャンネルはクラスタリングへ接続し，ホストチャンネルはバックボーンリングへ接続する．

MCC IIはMCC以上の処理能力が必要であるため，MCCのホストチャンネルで用いたPAL回路とFIFOメモリによるインタフェース回路を改良し，CPUからの指示により，受信用のFIFOメモリから1パケットを送信用のFIFOメモリへハード的に転送するコントロール回路と転送バスを設け，処理の大半をハードウェア化した．現在この回路を各チャンネル部分に用い制作したMCC II において周辺チャンネル200Kbps，ホストチャンネル1Mbpsの処理を行うことができ，1.2.2で示した目標を満たした．

1.3.5 CM-1

CM-1はバックボーンリングから送られるパケットを受信し，MCCの周辺チャンネルに接続された最大1024台のパーソナルコンピュータ等の機器に1.2.2で述べたS-netの機能を与えるようにバックボーンリングへ各機器向けのパケットを送信する．

CM-1は図1.4で示した構造のパケットを受信する．ヘッダ2から終端コードまでがMCCが組立てたパケットであり，MCC IIはこのパケットをデータとして受取り，ヘッダ1を付けてCM-1に送る．ヘッダ1の下位5ビットはデータを出力した機器が接続されたクラスタリング番号(CN)を示し，ヘッダ2の下位5ビットはそのクラスタリング内の機器番号(PN)を示しているため，この2つのヘッダからCM-1はこのデータがどの機器からのものか判別する．

CM-1から特定機器へのデータ送信は，その機器が接続されているクラスタリングのCNとその機器のリングにおけるPNをそれぞれヘッダ1，ヘッダ2の下位5ビットに書いて，図1.4の構造のパケットを作り送信することで行う．

チャンネルのインターフェスは，MCC IIで用いたFIFOを利用したハードウェアによる転送回路に，1パケット受信終了時にCPUに対して割込みを発生する機能を付加し，割込み処理において受信パケットを処理するようにした．この結果，CM-1は1Mbpsで送られるパケットを処理でき，1.2.2の目標を満たした．

CM-1のソフトウェアはMCC同様，状態遷移法の手法を用い，多くのPSM (Procedural State Machine)を用いて構成した．図1.5にこのPSMのブロック図を示す．PSM INは1パケット受信時に割込みにより起動するもので，パケットの受信処理を行う．PSM OUTはパケットの出力処理を行う．PSM PROCESS(0)〜(1023)は1024の各チャンネルごとに用意したPSMであり，チャンネルからのデータを処理し，各チャンネルへのサービス処理を行う．

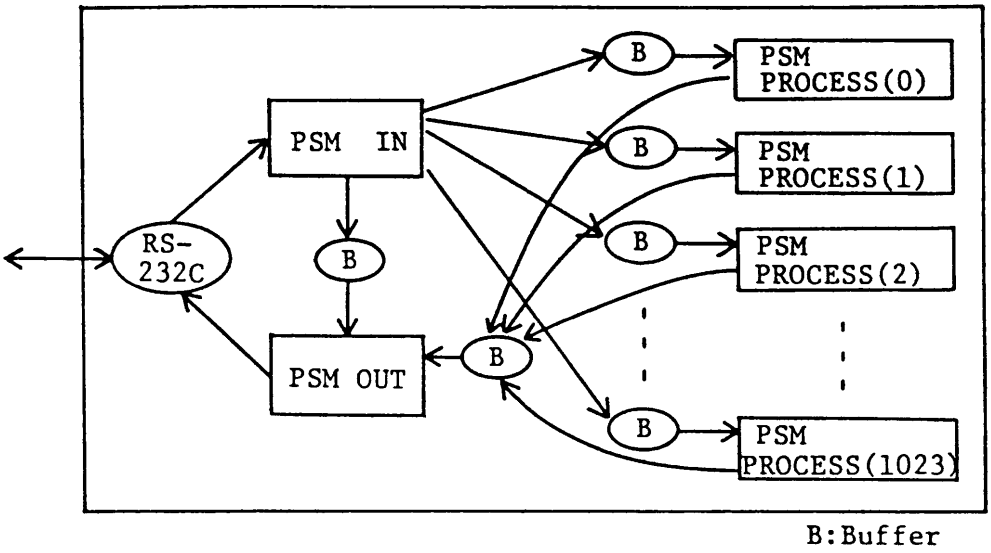


図1.5 CM-1のPSMブロック図
Fig.1.5 Block diagram of PSMs in CM-1.

1.4 構築と運用

1.4.1 構築

これ迄に述べたS-netを実際に信州大学の2つのキャンパスにおいて構築し，大学計算機センターと各研究室を結びキャンパスネットワークを実現したので報告する．

信州大学は，長野県下の長野市，松本市，上田市，伊那市の4つの市に分れ，さらに長野市には2つのキャンパス(工学部と教育学部)があることから，全体で5つのキャンパスに分れている．信州大学では，この5キャンパス間をマイクロ回線で結ぶ計画を立て，現在長野市の工学部と松本市・上田市のキャンパス間が完成している．このマイクロ回線は，7.5GHz帯のデジタルマイクロ波無線回線を用いて各キャンパス間に1.5Mbpsの回線を12回線確保するものである．S-netは，この1.5Mbpsの回線及び多重装置を用いてこの1.5Mbpsの回線へ接続する64Kbpsの回線により，各キャンパスを結んだネットワークを実現する予定である．現在構築を完了し運用されているS-netは，工学部と松本市(本部，人文学部，経済学部，理学部，医学部，教養部)のキャンパスのS-netであり，この2つのS-netをマイクロ回線で結んでいる．

工学部のS-netは，1985年に運用を開始し，現在総チャンネル数295を持つ(図1.6)．各クラスタリングにMCCを介して計算機センタの大型計算機，各学科のミニコンピュータ，各研究室のワークステーション，パーソナルコンピュータを接続している．具体的な接続コンピュータは，日立製大型計算機M-260，CONVEX製スーパーミニコンピュータC1-XP，ワークステーションはSUN，NEWS等であり，さらに各社のパーソナルコンピュータを接続している．このクラスタリングを光ファイバを用いてMCC IIに接続してバックボーンリングにパケットを伝送し，計算機センタに設置したCM-1に送信している．また，一部の学科に敷設されているIEEE 802.3 (Ethernet)によるネットワークとも，ワークステーションを用いて接続され，相互利用を可能としている．MCCに接続する各コンピュータからのチャンネルの規格は，1200bps〜19200bpsのCCITT V.24(RS-232c)であり，MCCにてこれをクラスタリングのパケットにして送信している．クラスタリングのデータ伝送速度は76.8Kbps，バックボーンリングのデータ転送速度は1Mbpsである．

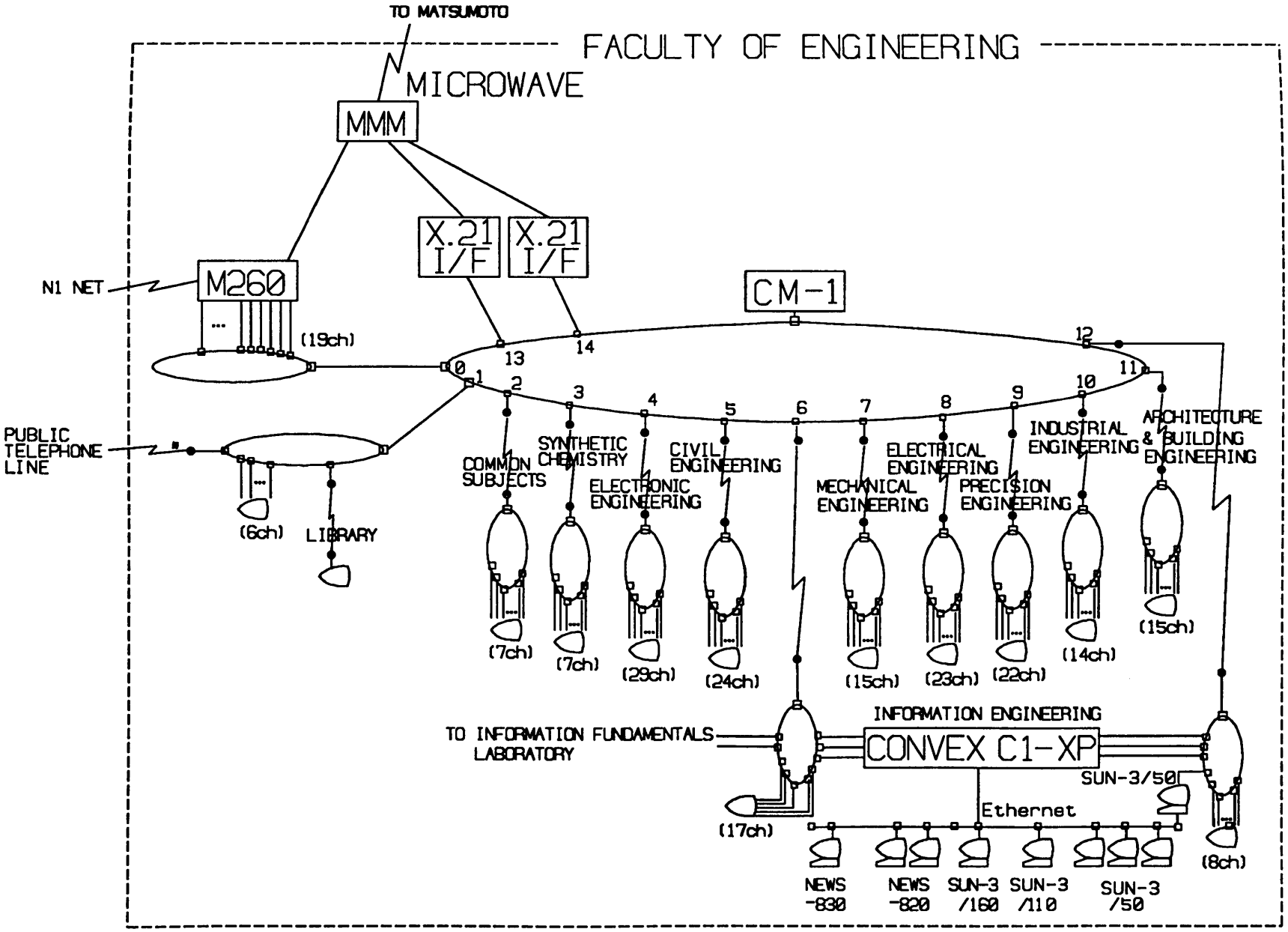


図1.6 S-netの構成(1)
Fig.1.6 Configuration of S-net(1).

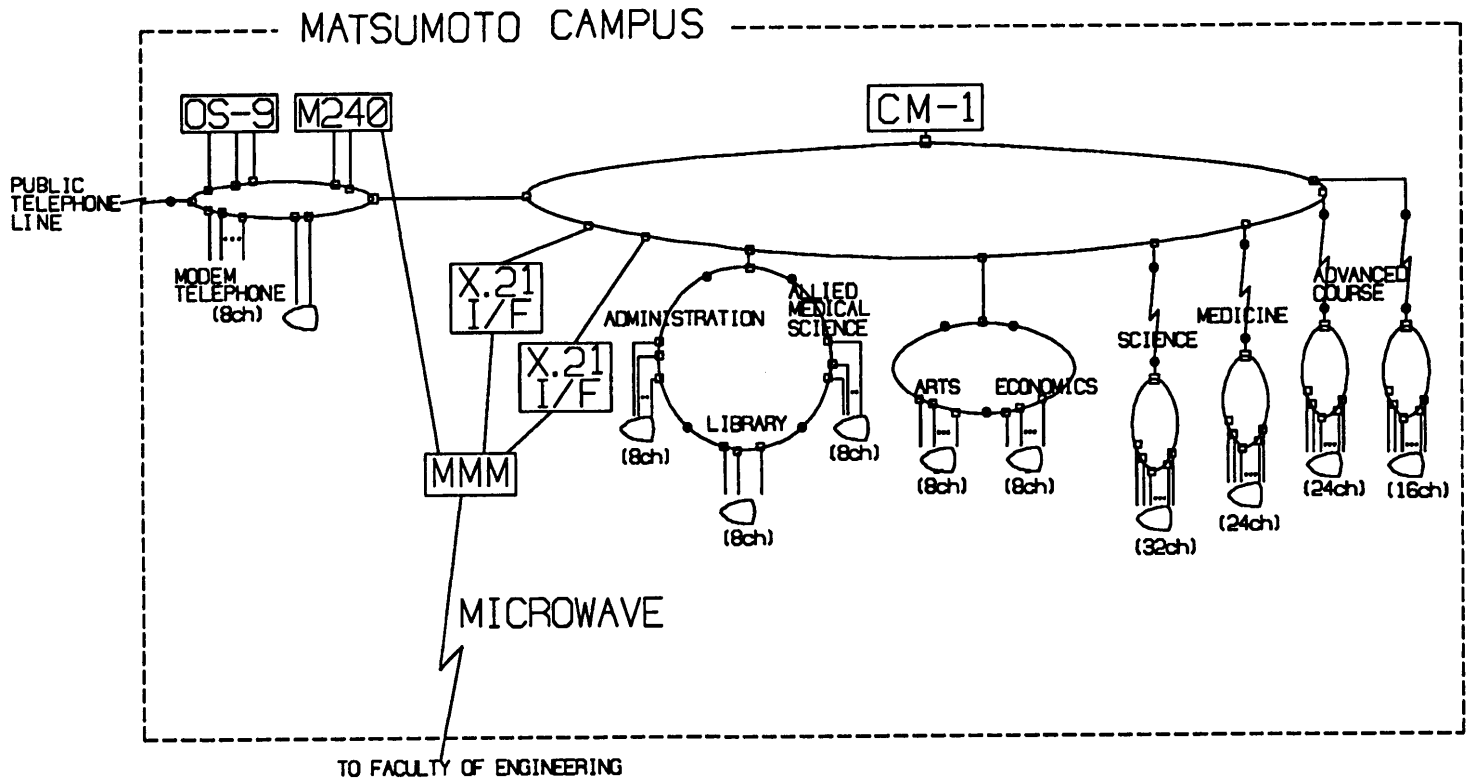


図1.7 S-netの構成(2)
Fig.1.7 Configuration of S-net(2).

松本市のキャンパスのS-netは，1986年に運用を開始し，現在総チャンネル数160を持つ(図1.7)．各クラスタリングにMCCを介して松本地区の大型計算機，ミニコンピュータ，ワークステーション，パーソナルコンピュータを接続している．

この松本キャンパスのS-netと工学部のS-netとは，互いのMCCのチャンネルどうしを前述のマイクロ回線を用いた64Kbpsの回線で接続し，パケット交換を可能とした．このため，互いに相手のS-netに接続されている計算機を利用することが出来る．このマイクロ回線との接続には，CCITT X.21を用いた．また，この両キャンパスのS-netの接続を，1.5Mbpsのマイクロ回線に変更する予定であり，接続用インタフェースの試作を終え，現在テスト中である．この接続の規格は，バイポーラ信号で符号形式はB8ZSである．

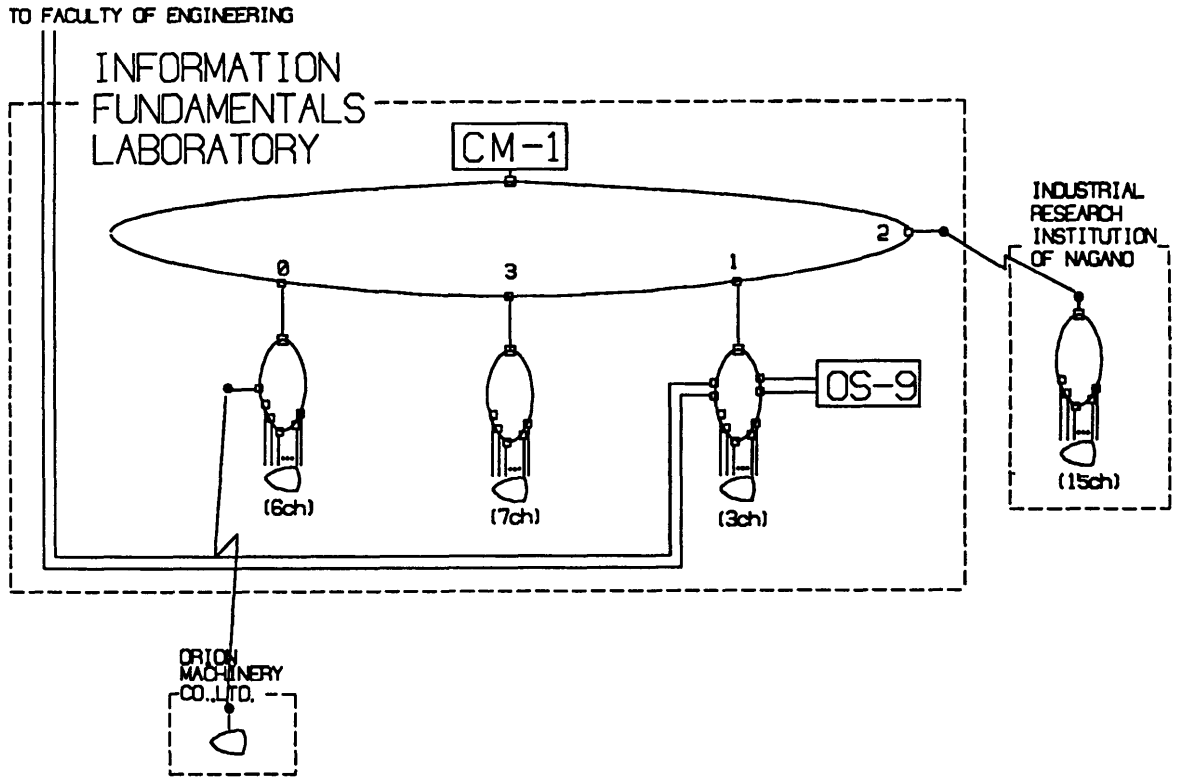


図1.8 S-netの構成(3)
Fig.1.8 Configuration of S-net(3).

また，この工学部，松本キャンパスの2つのS-netのほか，筆者の研究室(工学部情報工学科内)に実験用のS-netを設置している(図1.8)．これは総チャンネル数28の小規模なものである．また，端末1台を非同期無線モデムを用いて工学部北接の企業の研究所に置き，実験に利用している．さらにこのS-netの4つのクラスタリ

ングのうち1つは工学部北接の長野県工業試験場に敷設し，非同期無線モデムを用いて研究室内MCC IIと接続し，試験場にもS-netの端末を設置した．

1.4.2 S-netの使用例

実際にS-netを利用し，1台の端末から複数の計算機を利用する例を示す．ここで用いる端末は，図1.8で示した筆者の研究室に敷設したS-netに接続されたものである．図1.9は，このS-netを用いて研究室内のワークステーションを利用する様子である．1行目はS-netからのメッセージであり，2行目が端末からの入力で，ワークステーションとの接続を指示している(以下図1.9,図1.10において，下線部は端末からの入力を示す)．

```
Welcome to S-net Ver. 2. 1
connect 1-2
Ok
sun3/50 in KISO lab. (sukisol)
login: fuwa
Password:____
Last login: Sat Jan 30 09:49:35 on console
Sun UNIX 4. 2 Release 3. 3 Mon Mar 16 17:59:43 JST 1987
disktop: Started on sd0. Mon Oct 12 17:07:04 JST 1987
sukisol% ls
a. out          install_chgrp  net            test. d
echoc. c        install_login  network
sukisol%
```

図1.9 S-netの使用例(1)
Fig.1.9 Example of using S-net(1).

```
Welcome to S-net Ver. 2. 1
connect 1-4
Ok
Welcome to S-net Ver. 3. 0[KOUGAKUBU]
connect 2-0
Ok
Welcome to S-net Ver. 2. 3[MATUMOTO]
connect 0-0
Ok
JET12012A ENTER USERID -
THM0
JET12026A ENTER PASSWORD FOR THM0 -
#####
JDT2531 THM0 LAST EXECUTION DATE=88. 02. 02 TIME=14. 20
JET1065I TSS THM0 STARTED TIME=14:22:41 DATE=88-02-02
################################################
#  WELCOME M-240H SYSTEM SHINSYU UNIV. MATSUMOTO   #
#  S-net GA SHIYOUDEKIRUYOU NI NARIMASITA          #
#  M-240H SYSTEM NO 24JIKAN UNTEN HA OKONAIMASEN  #
################################################
JET11060I USER COMMAND PROFILE BEING USED
)) LISTC
IN CATALOG:SYS1. CENTCAT
THM0. SYSPROF
```

図1.10 S-netの使用例(2)
Fig.1.10 Example of using S-net(2).

図1.10は，この端末から，松本のS-netに接続された汎用計算機を利用する様子である．工学部のS-netとの接続を行った後(2行目)，松本のS-netとの接続を行う(5行目)．以後，この端末は，マイクロ波通信を経由して松本のS-netの端末として利用できる．この後，汎用計算機との接続を指示し(8行目)，この計算機を利用する．

1.4.3 ネットワーク管理

キャンパスに設置したS-netを運用するにあたっては，S-netの状態を常に監視して，障害が有れば直ちに発見して通知し，復旧させるネットワーク管理が不可欠である．障害は発生しないことが一番であるが，実際には様々な異常が発生する．S-netでも過去6年の間には，CM-1,MCC II,MCCの各装置の故障のほか，雷による装置のラッチアップやMCCの電源の不注意による切断等様々な障害が発生した．

このため，現在S-netでは，各リングの状態を監視する監視システムを開発して，常にS-netの状態を監視している．この監視システムは，各クラスタリングごとにそれぞれ1つの周辺チャンネルを確保して，MCCのこの周辺チャンネルの回線の出力線を入力線に短絡させることでループバックさせ，5分ごとに各クラスタリングのループバック試験を行うことで実現している．またその際，各回線の使用状況も調べて記録し，障害発生時に利用者への連絡に利用すると共に，ネットワークの利用実績を調べることにも用いている．

1.5 評価

この信州大学におけるキャンパスネットワークS-netを評価する．先ず，チャンネル間の伝送能力を調べ，次に利用状況を調べた．さらに，利用者の評価を調べるために，アンケートを行った．

1.5.1 伝送能力

S-netの伝送能力を調べるために，S-netに接続された2台のパーソナルコンピュータ間での伝送実験を行った．伝送実験は，1台のパーソナルコンピュータ(PC1)からもう1台のパーソナルコンピュータ(PC2)へ，102,400byteのデータを伝送する時間を計測する．

伝送実験の前提条件を次に示す．2台のパーソナルコンピュータ間の伝送の上位プロトコルは，プロトコルのオーバヘッドが伝送時間に与える影響が出来るだけ少なくなるように，次に示す簡単なものとした．データの伝送は，まずPC1から伝送開始のためのENQコードを送り，PC2がこれを受信してACKコードを返すことで開始する．その後，データを一定量PC1が送信し(この一定量のデータのかたまりをデータブロックと呼ぶ)，PC2はこのデータブロックを受信ごとにACKコードを返す．PC1はこのACKコード受信後，次のデータブロックを送信する．またPC1は全データの送信終了後終了コードEOT を送る．これに対して，PC2がACKコードを返して伝送を終了する．

2台のパーソナルコンピュータの通信インタフェースは，非同期のRS-232cでこれを19200bpsで用いる．計測する伝送時間は，PC1がENQコードを送るときから，EOTコードに対するACKを受信するまでの時間である．

この時間を，2台のパーソナルコンピュータをケーブルで直結した場合と，工学部のS-netの異なるクラスタリングに接続したMCCのチャンネルにパーソナルコンピュータをそれぞれ接続し，遠隔ログイン機能で互いにデータ交換が可能な状態とした場合について計測した．直結した場合のケーブル長および，MCCとパーソナルコンピュータとを接続したケーブルの長さは，ともに数メートルである．

実験は，データブロックのデータ長を10,240byteとしたとき(つまり，全データを10に分けて送る)と，51,200byteとしたとき(つまり，全データを2回に分けて送る)場合について行った．計測時，このS-netは14人の利用者が遠隔ログイン機能を用いて各自のパーソナルコンピュータを計算機センタの計算機に接続して利用しているという，平均的なS-netの使用状況であった．結果を表1.1に示す．

この表より，S-netを利用しても，伝送時間はほとんど変らず，S-netによる遅延が少ないことが確かめられた．データ分割数が少ない方がより遅延が少ないのは，PC1からのデータブロックとPC2からのACKコードの伝送回数が少ないからであ

表1.1 伝送実験結果
Table 1.1 Results of Transmission Test.

Data Block Length			
10240 bytes		51200 bytes	
Direct	Via S-net	Direct	Via S-net
53.5 sec	54.0 sec	53.5 sec	53.7 sec

る．コンピュータを接続するMCCは，コンピュータからの受信データを蓄えておくバッファが空の状態のとき，1byte受信後直ちにこれをパケット化して送信すると，パケット数が非常に多くなってしまう．そこで，受信しても直ちにこれをパケット化せず，一定時間(1.6msec)続くデータを待つようにした．このため，新たなデータの伝送のたびに，パケットのアクセス遅延の他に1.6msecの遅延が生じる．

1.5.2 利用状況

工学部のS-netの1990年1年間の利用者数は，のべ48,734人と多い．これを月別に示すと，図1.11のようになる．卒業研究で忙しくなる10月から2月までの間の利用者は特に多い．この時期S-netに接続されている計算機センタの大型計算機のチャンネルは，いつも遠隔ログインにより各研究室のパーソナルコンピュータと接続されており，空いているチャンネルがほとんど無い状態が続く．

最も利用者の多い1月について，時刻別の利用者数を図1.12に示す．図より利用者は16時をピークとして，一日中利用されていることが分る．22時に利用者が増えるのは，計算機センタが閉館し，センタの大型計算機はS-netを用いなければ利用できなくなるからである．ちなみに，工学部からのセンタ計算機の利用の内，1990年では66%がS-netを用いての利用である．

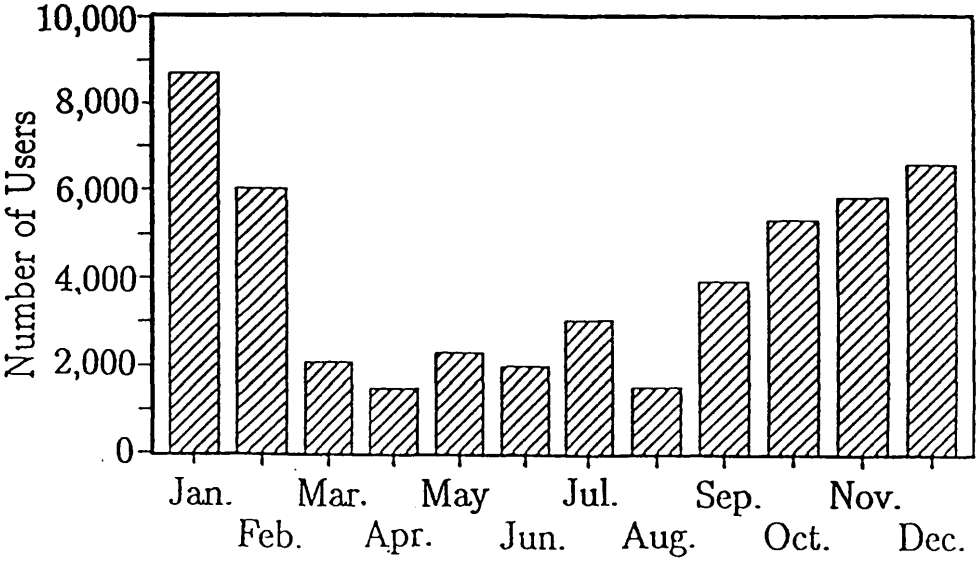


図1.11 S-netの月別利用者数(1990年)
Fig.1.11 Breakdown of S-net Users by Month (1990).

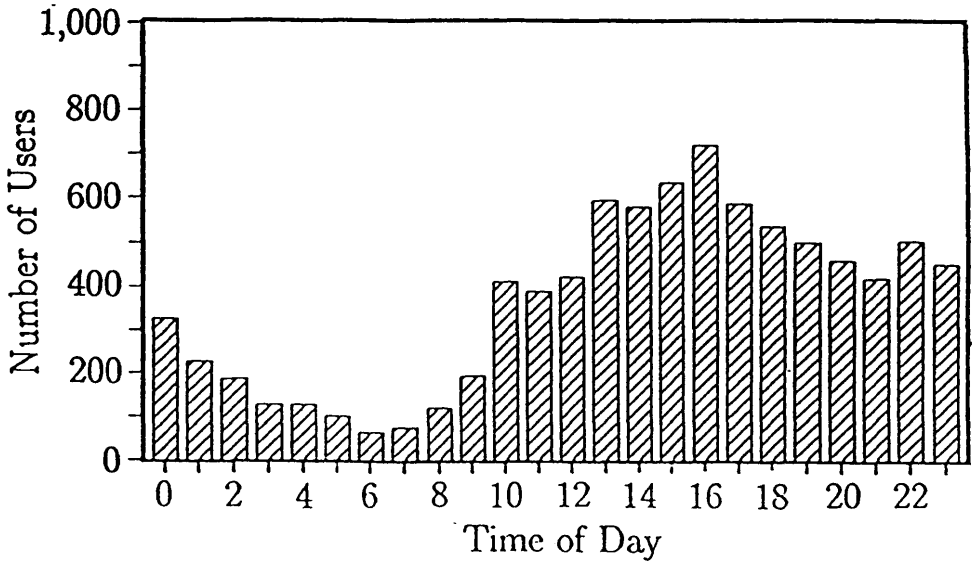
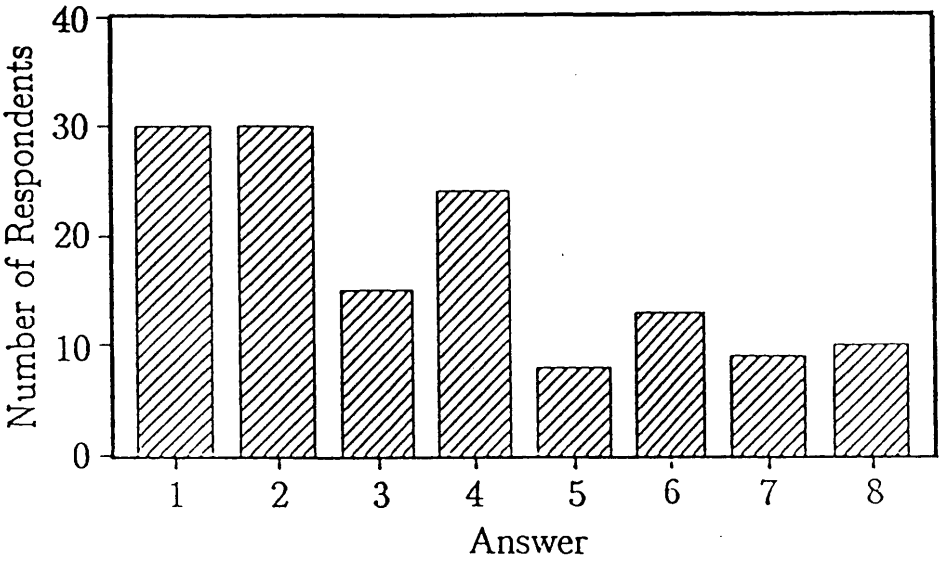


図1.12 S-netの時刻別利用者数(1990年1月)
Fig.1.12 Breakdown of S-net Users by Hour (Jan. 1990).

1.5.3 アンケート

工学部のS-netを利用できる人(S-netの回線が引かれている研究室の教官，技官，学生)全員に，S-netについてのアンケート調査を行った．アンケートをお願いしたのは254人であり，内139人より解答があった．



1. Almost never
2. Less than 30 min
3. 30 min ~ 1 hour
4. 1~2 hours
5. 2~3 hours
6. 3~4 hours
7. 4~5 hours
8. More than 5 hours

図1.13 S-netの1日平均利用利間
Fig.1.13 The Average Daily Usage of S-net.

まず，一日の平均S-net利用時間を調べた結果を，図1.13に示す．図より，回線が研究室にあってもほどんど利用しないひとも22%いるが，多くの人(78%)はほぼ毎日S-netを利用していることが分る．また，毎日1時間以上も利用する人が，全体の46%もおり，キャンパスネットワークの需要が大きいことがわかる．

さらに，この1日の利用時間を教官と学生別に調べてみると，教官は短時間の利用者が多く，学生は長時間の利用者が多いことが分る．教官は，30分以内の利用者が全体の80%を占めるのに対し，学生の54%は1時間以上利用している．教官は主にネットワークを電子メールの利用等短時間の処理に利用しており，学生は研究に利用しているためと考える．さらに，毎日3時間以上利用しているものは学生の27%にもなり，多くの学生の研究にS-netが欠かせないものになっていることを示している．

次に，計算機センタの大型計算機を利用するときに，S-netを用いる割合を調べた結果を，図1.14に示す．図より90%はS-netを利用するという人が最も多く，常にS-netを利用するという人も多い．半分以上S-netを利用する人は75%もいる．

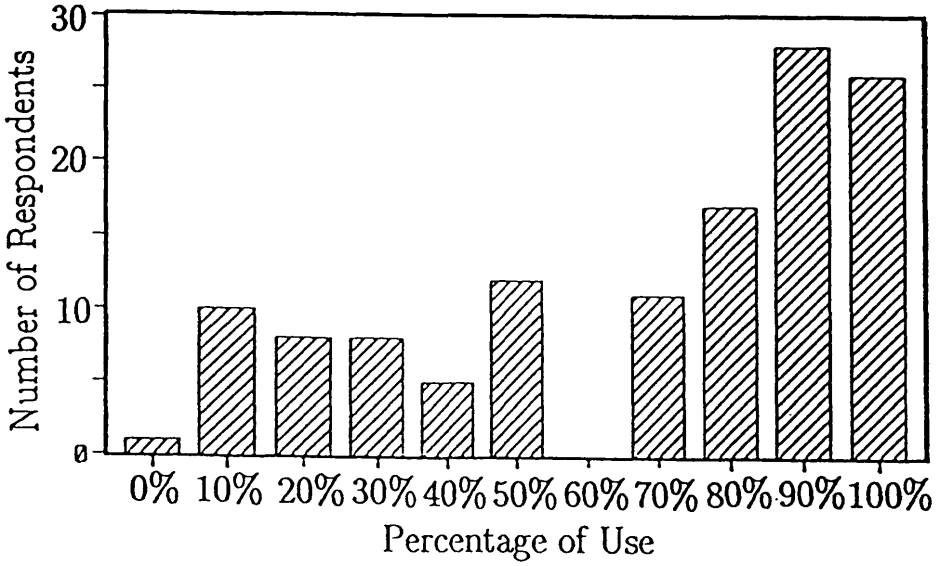
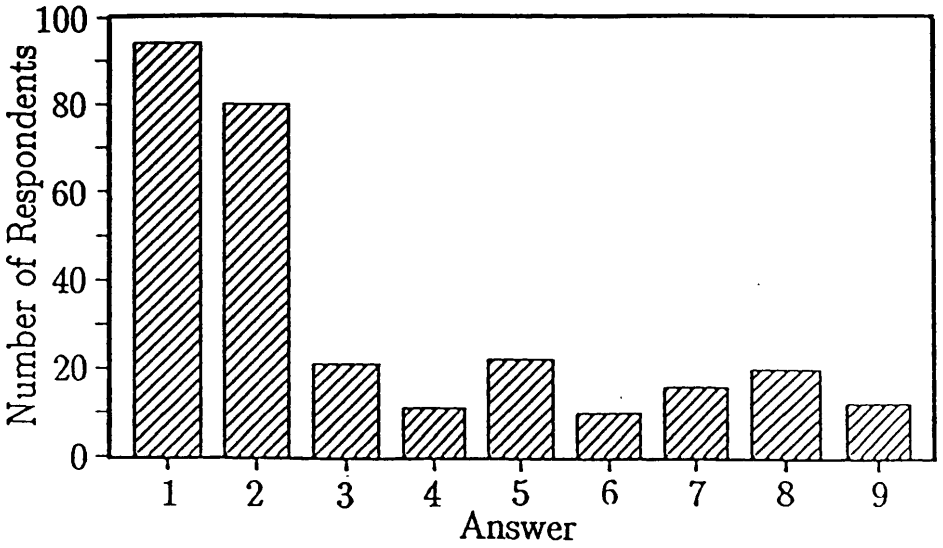


図1.14 センタ計算機使用時の，S-net 利用率
Fig.1.14 Rate of Computer Center Access via S-net.

図1.15 S-netに対する満足度
Fig.1.15 User's Level of Satisfaction with Regard to S-net.

このS-netに対する各自の満足度を調べた(図1.15)．図より72%の人が満足もしくはほぼ満足と答えているが，不満足と思う人も多い．この不満足である理由の大半は，計算機センタの大型計算機に接続するチャンネルの不足である．現在S-netに接続されている大型計算機のTSSチャンネルは，9600bpsが16回線，4800bpsが6

回線，1200bpsが14回線であり，利用者が多い9600bps回線は，常に遠隔ログイン機能により各研究室より使用され，空きがほとんど無い状態が続いている(1991年1月におけるこの9600bps チャンネルの使用率は，14時から15時の間では平均86%である)．



1. Increase in the number of M-260 channels
2. Improvement in network reliability
3. Connections with other networks
4. Transmission of audio
5. Transmission of still pictures
6. Transmission of moving pictures
7. Electronic mail exchange, foreign and domestic
8. Connections with other PC networks, database centers, etc.
9. Wireless access to the S-net

図1.16 S-netに対する要望
Fig.1.16 The Expectation of Future S-nets.

このことは，今後のS-netに対する希望調査を示す図1.16からも分る．様々な希望がある中でも，この計算機の9600bpsチャンネルの増設はアンケート解答者の68%にあたる94人が希望している．また，その他の希望として，ネットワークの信頼性向上が多かったことは注意すべきであり，今後のS-netの改良時においては，信頼性を最大限に考慮すべきであることを示している．また，学外の様々なネットワークとの接続や，音声・画像の伝送といった要求も多いことが分り，キャンパス

ネットワークでも他のネットワークと同様に，WANやマルチメディアネットワークにたいする需要があることがわかる．このため，S-netの高速化も重要な課題である．

1.6 今後のS-netの改良

現在計画しているS-netの今後の発展について述べる．

1.6.1 接続可能台数の増加

現在のS-netに接続可能な計算機は，最大で1024台である．今後，S-net利用者の増加に伴い，この制限が問題となる．特に，1つのクラスタリングに接続できる計算機が32台までであるため，多くの計算機を接続したい学科では1つのクラスタリングに収容しきれなくなる．

この問題を解決するために，S-netに接続出来る計算機数の大幅な増加を計画している．接続台数の増加のためには，2つの問題を解決する必要がある．一つはパケットのヘッダ部の構成についてであり，もう一つはCM-1の処理についてである．

接続可能な計算機数を制限しているものの一つは，パケットのヘッダ部構成である．各クラスタリングに収容されるコンピュータを識別するチャンネル番号(CN)も，クラスタリングを識別するチャンネル番号(PN)も，共に現在のパケットのヘッダ部では5bitで表される．接続可能台数を増やすためには，このビット数を増やすため，ヘッダ部の拡張を行う必要がある．この際，現在のMCC, MCC IIも利用できるように，ヘッダ部の拡張は現在のヘッダ構成の上位互換性を保証する必要がある．

この点を考慮して決定した新しいパケットのアドレス部構成を図1.17に示す．この構成は，CNとPNの両方共通である．図中(1)が現在のヘッダ構成であり，(2)が拡張したヘッダ構成である．拡張は，現在のヘッダ構成で用いられていない上位3ビットのパターンを用い，このパターンの時にヘッダ部を2バイト長とするものである．この結果，CNとPNは1バイト目の5bit($PN1$)に加え，2バイト目の7bit($PN2$)が加わり，合計12bitとなる．そして，PNは0〜4095となり(内0〜31のときは現在の(1)の構成を用いる)，1つのクラスタリングに接続できるコンピュータは最大4096

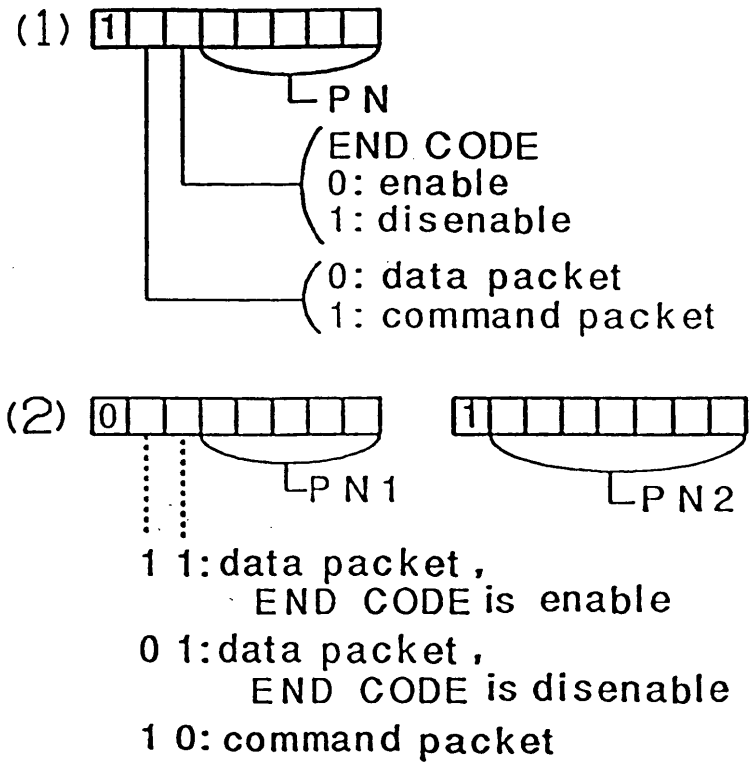


図1.17 新しいヘッダ構成
Fig.1.17 New Header Structure.

台，S-net中のクラスタリング数も4096リングとなり，接続可能なコンピュータの台数は16,777,216台と大幅に増える．

CM-1は，接続された各計算機に対する処理を行うために，多くの表を用いる．例えば，ある表は各計算機からの入力データを格納するバッファとして用い，またある表は各計算機に対して現在行っている処理の状態を記憶するのに用いている．接続可能台数の増加は，この表を大きくする必要があるが，CPUで扱うことができるメモリ空間の制約があると共に，表の拡大によりソフトウェアによる表アクセスのオーバヘッドも大きくなる．このため，表をメモリと制御回路から構成されるハードウェアで実現してCPUのメモリ空間から切り離し，表への書き込み読み出しも，CPUがこのハードウェアにコマンドを送ることで実現する方式を採用することとした．現在，このハードウェアを製作し，試験を行っている．

1.6.2 高速化

S-netの利用範囲を広げるためには，伝送速度の高速化は不可欠である．今後普及が予想される狭帯域ISDNと接続するためには1.5Mbpsの速度が，また既存のLANとの接続をスムーズにするためには10Mbpsの速度が必要である．また，動画の伝送まで可能とするマルチメディアネットワークとするためには，100Mbps以上必要となる．

現在，S-netを高速化するために，高速伝送のためのインターフェース回路の設計を行っている．この設計には，受信データのビット同期回路やキャラクタ同期回路の設計・試作から，リング型ネットワークにおいて，リング中の全ノードが同期確立を完了したことを確認してパケット送信を開始するまでのシーケンス[FUWA89b]といったことまで含まれる．

一方，高速なパケット交換を可能とするためのアクセスプロトコルについての検討も重要である．このことについては，**第2部**で詳しく論じる．

1.6.3 信頼性の向上

現在のS-netでは，各ループの障害チェックは行っているが，障害の復旧処理は人手で行っている．今後S-netを信州大学の各キャンパスに拡張するにあたっては，この復旧処理を自動的に行わせることが不可欠となる．このリング型ネットワークにおける信頼性の向上については，**第3部**で論じる．

1.7 結語

簡単な機器構成で柔軟なネットワークが構築でき，ネットワーク内の様々なシステムを利用することが出来る，キャンパスネットワークS-netを提案し，使用する各機器を設計制作した．更に，これを信州大学の2つのキャンパスに構築し，多くの利用者を得た．設計，製作した各機器は，設計目標の性能を満たし，現在支障なく動作している．

第1部では，このS-netの現状と利用状況，利用者の意見についても述べた．また，この利用者の意見と6年間のS-netの運用経験より考える今後のS-netの改良点についても述べた．

S-netは信州大学独自のキャンパスネットワークであるが，本論文によって明らかになったキャンパスネットワークに対する需要や要求は，全てのキャンパスネットワークに対して共通のものであり，キャンパスネットワークのあるべき姿を考える上での貴重な資料である．

特に，高速化，信頼性向上がキャンパスネットワークでは重要であることが示された．そこで，第2部では高速化について，第3部では信頼性向上について検討する．

第2部
高速化に適したリング型ネットワークのMACプロトコル

第2章 課題と諸仮定

2.1 課題

近年，ネットワークを介して大量の情報を高速に転送することに対する要求が，飛躍的に高まってきている．この要求は，LAN，WANともに強く，特にリング型のネットワークは，各LAN間を結んだバックボーンネットワークとしても利用されることから，より高速化が望まれている．

筆者は，信州大学におけるキャンパスネットワークとしてS-netを提案し，運用をしてきた．S-netは，レジスタ挿入型のリングを2段に階層化した構成をとっている．このS-netでも，利用者に対するアンケートの結果，高速化に対する要望が多い．

ネットワークの高速化を図るためには，伝送路を高速化するだけでなく，各ノードの処理も高速化する必要がある．伝送速度が100Mbps以上の高速通信では，このノードの処理がボトルネックとなるため，広帯域ISDNで検討されているATM網のように，ノードでの処理を全てハードウェア化する必要がある．そして，パケットの構成やMACプロトコル等を，ハードウェア処理に適した構成としなければならない．S-netでもこういったことから，ハードウェア処理が容易であるスロットリング[FLIN83]の考えをとりいれて，レジスタ挿入型リングのパケット長を固定長とし，送信時にデータが無い場合は空スロットを送信する方式（以後これをレジスタ挿入型スロットリングRegister-Insertion Type Slotted Ring: RISリングと呼ぶ）の採用を検討し，現在同期確立の方式やそのためのハードウェア等について研究を進めている[FUWA89b]．

一方，このレジスタ挿入型スロットリングが，基本的にどの様な性能を示し，いかなる特徴を有するかを，定量的に評価しておくことも重要である．さらに，従来から用いられてきている様々なMACプロコトルの性能も同様に評価し，様々な利用環境下に対して，どのMACプロトコルが最適であるのかを明らかにする必要がある．

第2部では，伝送速度が高速な場合において，与えれれた利用環境下において最適なMACプロトコルを明らかにするため，各MACプロトコルの性能解析を行い，その結果を比較する．解析するMACプロトコルはRISリングに加え，リング型ネットワークの代表的なプロトコルであるトークンリング，スロットリング，レジスタ挿入リングの4つである．

従来から，リング型ネットワークの様々なMACプロトコルの性能解析が行われてきた[BUX83] [LOUC85] [KOSI87] [LIU88] [HAMM86] [BHUY89] [TANAB89] [ABDU84]．これらは，いずれも対象とするネットワークの性能を良く表現できるモデルを構築し，各ネットワークの性能を明らかにした優れた研究である．しかしながら従来からの解析は，次の点で問題がある．

(1)各ノードのバッファ段数を無限と仮定している．

これは，本来有限であるバッファ段数を解析の簡単化のために無限とするもので，プロトコルの重要な特性であるバッファのオーバフローや平均パケット数について解析することが出来ない．

本研究では，バッファの段数がn段($n = 1, 2, 3, \cdots$)と有限の場合について解析し，これらの特性も扱える様にする．

(2)隣接ノードからのパケットや空のスロット到着を，不自然な仮定により簡単化している．

従来の解析では，隣接ノードからのパケット到着や空スロット受信に関し，ノード間でのパケット到着の相関性を厳密にモデル化するのが困難なために，様々な仮定を設けてモデルを簡単化している．この到着をポアソン分布と仮定したり[HAMM86] [TANAB89]，この仮定による解析結果とシミュレーション結果の差を小さくするために，1台前のノードにおける隣接ノードからのパケット到着と端末からのパケット発生から，このノードのパケット到着を求める解析もある[BUX83]．後者の場合も，1台前のノードのパケット到着はポアソン分布としているため，解

析結果は高負荷時にシミュレーション結果と一致しない．また，パケット到着を回線のスループットを用いた幾何分布で表せるとして解析する研究もあるが[LOUC85] [KOSI87] [BHUY89]，この場合も高負荷時に解析結果がシミュレーション結果と一致しなくなる．さらに，ノードの発生パケットと空スロットの到着との間に，フローバランスが成立しているとして解析する研究もある[LIU88]．しかし，常にフローバランスが成立するとはかぎらず，通常自ノード発生パケットの方が到着する空スロットより多いときはノードのオーバフローが発生する．これを常にフローバランスが成立しているとしたのでは，ノードにおけるオーバフロー問題を扱えなくなる．

本論文ではモデルを簡単にするための上記のような仮定を設けず，隣接ノードからのパケット到着は各ノードの振舞いを表すマルコフモデルを解くことで，求めることにする．

(3)個々のMACプロトコルごとにモデル化の為の仮定や解析方法が異なっており，解析結果の比較が出来ない．

最適MACプロトコルを求めるためには，各MACプロトコルを同一の仮定の元で解析し，これを同一の方法で解析する必要がある．そこで，本研究では4つのMACプロトコルを出来るだけ同一の仮定の元でモデル化する様，共通の仮定を設けた．また，共通の解析手法としては，モデルが複雑な有限状態マルコフ連鎖モデルであっても解析ができる，平衡点解析[TASA86]の手法を用いることとする．この手法は，既に衛星通信網とバス型LANにおける様々な形態のネットワーク解析に成功しており，本研究はこの平衡点解析の適用領域をリング型ネットワークにも拡張するものである．

以下本論文では，まず2.2で共通の仮定を述べる．そして，第3章, 第4章,第5章, 第6章でそれぞれRISリング，トークンリング，スロットリング，レジスタ挿入リングのモデル化と解析について述べる．最後に第7章で，各MACプロトコルの解析結果を比較して，最適プロトコルについて議論する．

2.2 諸仮定

各MACプロトコルを解析する際の共通の仮定を説明する．筆者は，この共通の仮定をもとにプロトコルをモデル化し，解析を行った．しかし，モデル化と解析を容易にする目的で，各プロトコルごとにいくつかの近似を行い，そのための仮定を導入した．各プロトコルのおける近似は，第3〜6章でそれぞれ述べる．

解析に際し導入した共通の仮定は，次の６つである．

(A1) リングに接続されているノードの台数をM台とする．

(A2) 時間をT_a/Mごとの時間単位(“ステップ”と呼ぶ)に細分化する．

(A3) 時間L_t,T_a,T_hは，隣接ノード間の伝搬遅延を含む．

(A4) 各ノードの端末バッファの段数は有限で，蓄えることが出来るパケット数をTとする．

(A5) 各端末はステップ当り確率σでパケットを1つ生成する．このパケットの発生はステップの最後の時点で起きる．発生したパケットは，ノードの端末バッファに蓄えられる．その際端末バッファが一杯であるときはオーバフローとなり，そのパケットは破棄される．

(A6) 端末バッファに蓄えられるパケットの宛先は，一様に確率$1/(M-1)$で自ノード以外の$M-1$台の各ノード宛である．

なお，ここでT_aは各ノードが1パケット(スロットリングでは1スロット) を送信(もしくは受信)するのに要する時間である．T_hは，受信したパケット(スロット)を次のノードへ転送する際の，ノード内での遅延時間である．L_tは，トークンリングにおいて各ノードがトークンを受信するのに要する時間である．

第3章 RISリングの性能解析

3.1 RISリング

3.1.1 RISリングの動作

レジスタ挿入型スロットリングについて説明する．ここで説明のため，M台の各ノードは1からMまでの番号を付ける．任意のノードに番号1を付けて“ノード1”と呼び，パケットの流れる方向で次に接続されているノードを“ノード2”と呼ぶ．同様に他のノードも接続順に“ノード3”，“ノード4”，…，“ノードM”と呼ぶ．

RISリングは，スロットリングとレジスタ挿入リング双方の特徴を持つ．スロットリングのように，各ノードは一定時間間隔でパケットを送信することが出来る．またレジスタ挿入リングと同様，各ノードは図3.1で示す様に，リングバッファ(Ring Buffer)・端末バッファ(Terminal Buffer)と呼ぶ2種類のバッファを備える．

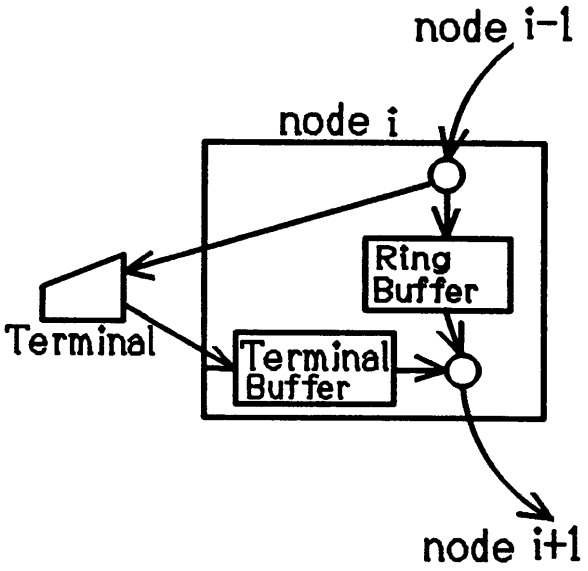


図3.1 ノードの構成
Fig.3.1 Basic structure of a node.

パケットは固定長であり，各ノードはこのパケットを一定時間間隔ごとにある送信タイミングに合わせて送信する．この送信可能なタイミングの間隔を“スロッ

ト”と呼ぶ．パケットはヘッダ部とデータ部から成る．ヘッダ部の先頭の1ビットは，データ部にデータが書いてあるか空かを示すフラグである．データ部にデータが書いてあるパケット（これをフルパケットと呼ぶ）のヘッダ部には，このパケットを受け取るノードのアドレスが書かれている．

隣接ノードから送られてきたパケットのうち，フルパケットはその宛先が調べられる．そして，このパケットが自ノード宛の場合は直ちにこのパケットをこのノードに接続された端末へ送り，パケットをリングから取り除く．自ノード宛でない場合はこのパケットをリングバッファに格納する．端末から発生するパケットは，端末バッファに蓄えられる．スロットの初めにリングバッファ又は端末バッファにあるパケットを一つ送信するが，この時常にリングバッファからの送信を優先させる．つまり，端末バッファからの送信は，リングバッファが空である時にのみ行われる．両バッファが共に空の場合は，空パケットを送信する．

各ノードの送信タイミングとして，筆者はまず全てのノードが同時に送信を開始するものを提案した[FUWA90a]．そして，より効率的なタイミングとして，隣接ノードから受信したパケットのヘッダ解析後，直ちに次のノードへ転送を開始できるように設定したものを提案した[FUWA90d]．本論文では，伝送遅延が少ない後者のタイミングを考えるものとする．そのタイミングを，ノード数が4の場合について図3.2に示す．ここで，時間T_aは1パケットの送信時間であり，T_hが一つ前のノード(node)から送られてきたパケットの受信を開始してからヘッダ部の受信を完了してこれを解析し，宛先が自ノードであるかどうかを識別するまでの時間である．高速のネットワークではこの解析処理は全てハードウェア化するので，T_hはパケットのヘッダ部を受信する為の時間にほぼ等しくなる．

各ノードは，パケット受信開始後T_h時間後にパケットの送信を開始できるように，タイミングが決められる．ただし，全てのノードでこのようなタイミングが決められるとは限らず，リング上1台のノードだけはT_h時間以上待つ場合がある．図3.2でも，ノード1はノード4の送信開始後$T_a - T_h \times 3 \quad (> T_h)$時間待つことになる．

ここで，パケットが送信を開始してから目的ノードに受信され始める迄の時間を“パケット転送時間”と呼ぶことにする．各パケットのパケット転送時間はこのノード1を通るか通らないかで異なったものとなる．例えば，図3.2においてパケットAとBは，共に2台のノードにより中継されて目的ノードに達する．しかし，パ

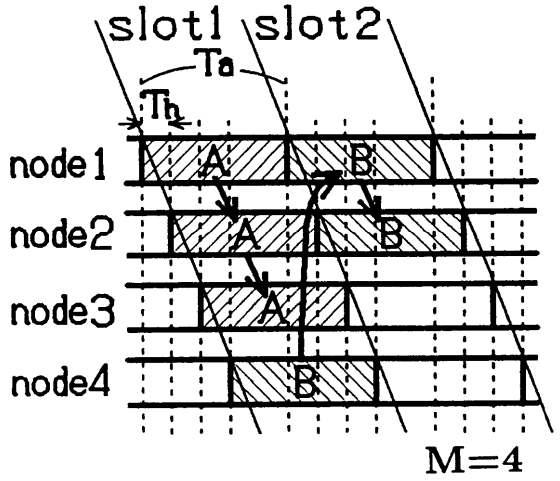


図3.2 パケット送信タイミング($M \leq a$)
Fig.3.2 Timing chart of sending packets in the case of $M \leq a$.

ケットAのパケット転送時間が$2 \times T_h$であるのに対してパケットBのパケット転送時間は$T_h + T_a - (M-1) \times T_h$(ここで$M = 4$)となる.

3.1.2 RISリングの分類

ここでは,RISリングにおけるパケットの流れがいくつかのグループに分れることを示し,このグループの数により,RISリングを分類できることについて述べる.

以下の説明において,記号aを次の式を満たす実数として定義する.

$$a = \frac{T_a}{T_h} \tag{3-1}$$

RISリングは,ノード数Mとこのaの値によって,$M \leq a$の場合と$M > a$場合の2通りに分類できる.図3.2は$M \leq a$の場合例であり,図3.3は$M > a$場合の例である.

図3.3では,ノード1を除く全てのノードはパケット受信後T_h時間後にパケットを送信する.この図で,ノード6からノード5宛のパケットBはスロット1でノード7からノード1へ転送された後,ノード1でスロット2のパケット出力に間に合わないため,スロット3まで待って出力される.このとき,間にあるスロット2では,別のパケットを転送できる.図では,このスロット2でパケットCとパケットDを転送している.パケットDは,パケットBと同様にノード1まで転送された後,1スロッ

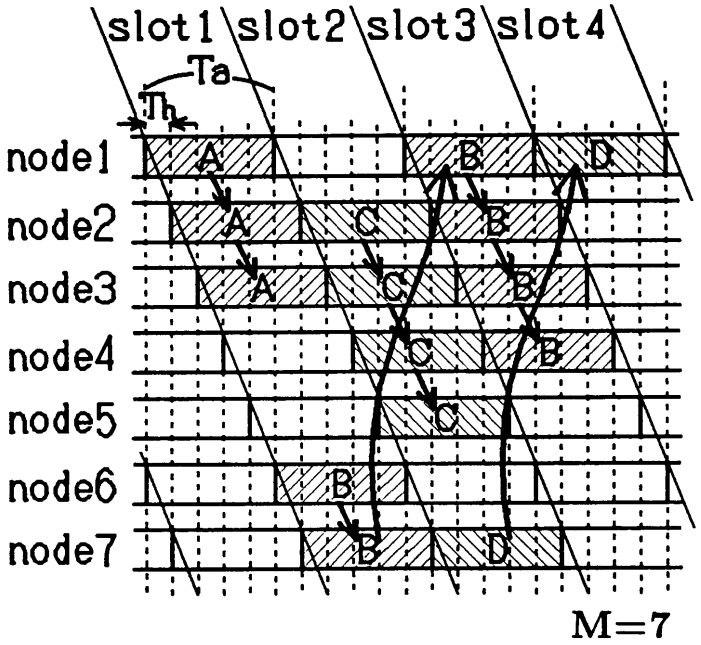


図3.3 パケット送信タイミング($M > a$)
Fig.3.3 Timing chart of sending packets in the case of $M > a$.

ト空けて転送される．つまり，この場合はスロット1,3,5,･･･での転送と，スロット2,4,6,8,･･･での転送という別の二つの系列に分れている．

この系列の数は，簡単に求められる．図3.4では，パケットの流れは矢印で表される．(a)は$a < M \leq 2a$の場合で，この場合は2系列に分れる（図3.3もこの場合であった）．一方(b)のように$2a < M \leq 3a$の場合は，3系列に分れる．一般に，パケットの流れは$\lceil M/a \rceil$の系列に分れる（$\lceil x \rceil$は，xの小数点以下を切り上げた整数）．

以後RISリングを平衡点解析の手法で解析するため，ネットワークにおけるM台のノードの各々がとりうる状態をモードとして表し，このモード間の遷移の関係を表現した離散時間のマルコフモデルを作成する．このモデル化は，$M \leq a$の場合と$M > a$の場合に分けて行う．

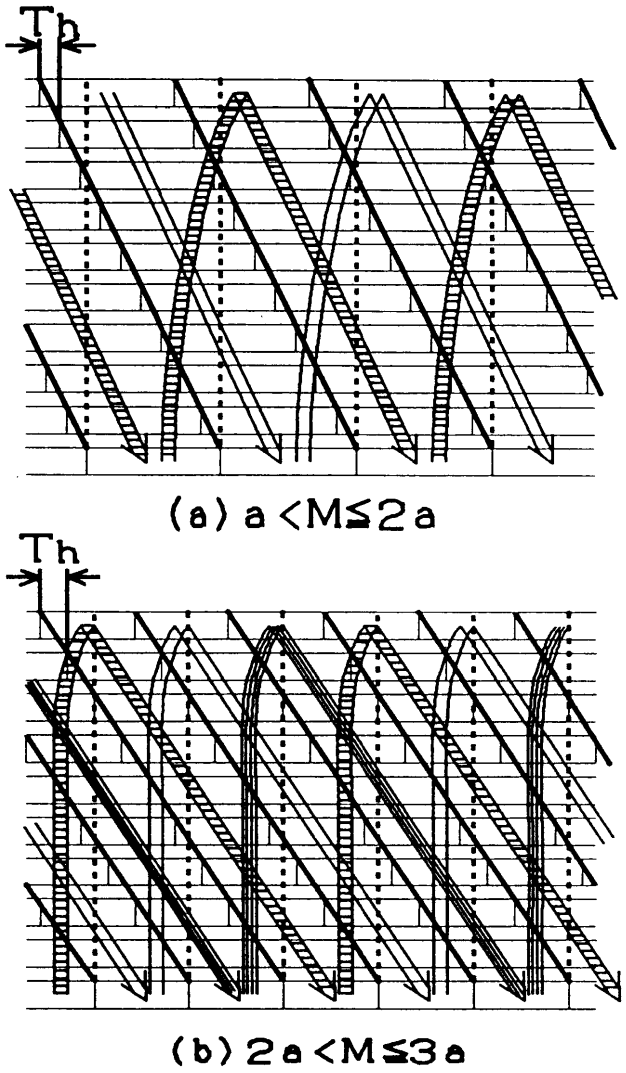


図3.4 パケットの流れ($M > a$)
Fig.3.4 Flow of packets in the case of $M > a$.

3.2 $M \leq a$の場合のモデル化

3.2.1 近似

図3.5(a)に示すように，ノード1だけは，パケット受信を開始してからパケット送信を開始するまでの遅れが$T_a - T_h \times (M-1)$となり，これ以外のノードはこの遅れがT_hである．この遅れの不均一をそのままモデル化すると，モデルはモード数が多く複雑なものとなる．

また，仮定(A6)よりパケットの宛先が一様分布となることは，モデルをさらに複雑なものとする．さらに，各ノードでの端末からのパケット発生についても，仮定(A5)より任意のステップでの発生を表すと，モデルはモード間の遷移が多くなり，解析が困難になる．

そこで，モデルを簡単なものとするために，以下の作業を行う．

[1] パケットの宛先の変形

パケットの宛先を一様分布とする仮定(A6)の代りに，宛先を幾何分布とする次の仮定(RI1)を導入し，モデルを簡単にする．

(RI1) 隣接ノードからパケットを受信したノードでは，このパケットの宛先が確率$E = 2/M$で自ノード宛であるとする．

ここで確率Eの値を$2/M$としたことは，次の理由による．パケットが送信元ノードから送信されてから，宛先ノードに到着するまでに，途中平均して何台のノードにより中継されるかを考える．この値は，宛先を一様分布とする仮定(A6)では$(M-2)/2$で与えられる．一方宛先を幾何分布とする仮定(RI1)では，$(1-E)/E$で与えられる．この2つの値を等しくするとEの値は次のようになる．

$$E = \frac{2}{M} \tag{3-2}$$

[2] 送信タイミングの変形

ただ1つのノード(ノード1)だけがパケット送信開始のタイミングが異なるということが，モデル作成を複雑にしている．このモデル作成の複雑さを回避するため，パケット送信開始のタイミング(パケット受信を開始してから送信を開始するまでの遅れ)が全て均一になるように，モデルでは近似を行う．図3.5(b)に近似した送信タイミングを示す．図で示すように，全てのノードでのパケット送信の遅れは，T_a/Mであると近似するのである．この遅れT_a/Mは，ちょうど1ステップに相当する．このタイミングの変形により，全ノードは1ステップごとに接続順にパケット送信を開始するようになりモデル化が容易になる．なお，このタイミングの変形は，モデルにのみ適用するものであり，実際のシステムのタイミングを変形するものではない．

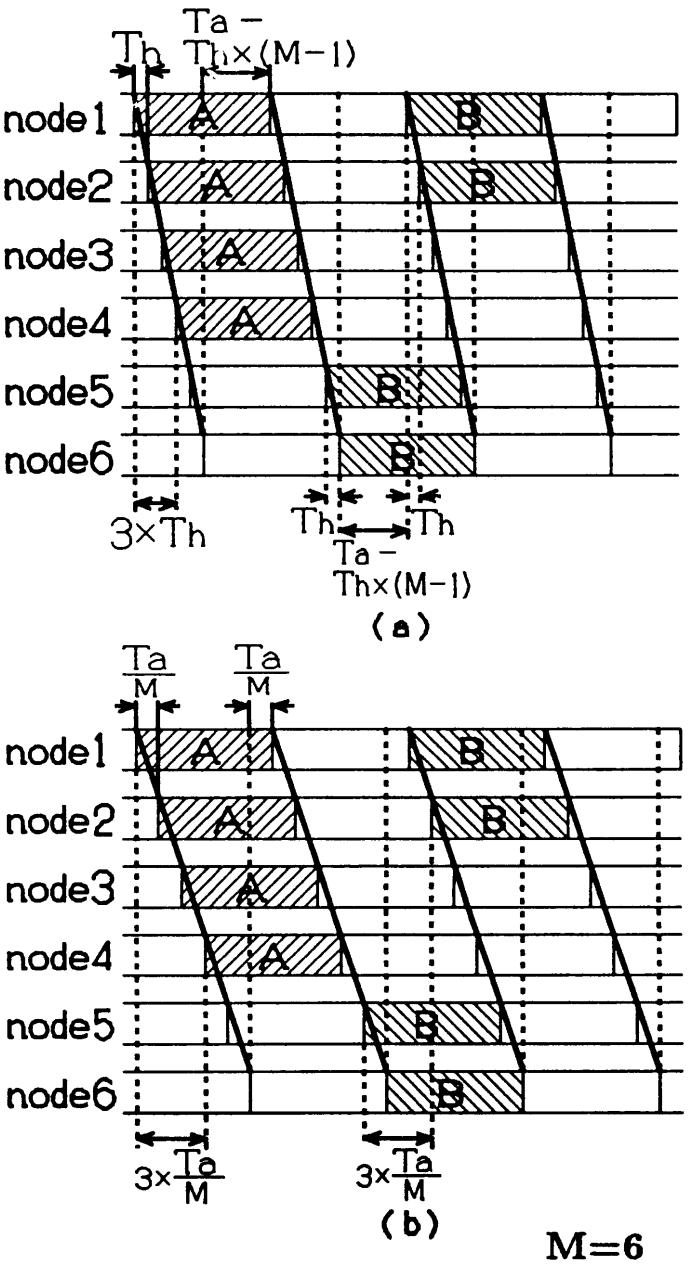


図3.5 パケット送信タイミングの変形($M \leq a$)
Fig.3.5 Modification of timing of sending packets in the case of $M \leq a$.

[3] パケット発生タイミングの変形

仮定(A5)より，各ノードに接続した端末は，任意のステップで新たなパケットを発生するが，これをそのままモデル化すると，モード間の遷移が多くなり，解析が困難になる．

そこで，このパケット発生タイミングについて，モデルでは仮定(A5)の代りに，次の仮定(RI2)を導入する．

(RI2) 各端末は，接続したノードが新たなパケットの送信を開始する直前に，確率$M\sigma$でパケットを生成し，端末バッファに蓄える．すなわち，Mステップ=1スロットの間に高々1個のパケットが確率$M\sigma$で発生する．

1スロットの間に2個以上のパケットが発生する確率については，通常$\sigma << 1$であるため無視する．

なお，これまでに示した各変形は，あくまでもモデルの簡単化のために行うものであり，実際のシステムを変形するものではない．

[4] パケット転送時間の補正

ここで，3.1.1で定義したパケット転送時間(パケットがその発生元のノードから送信され始めた時点から，宛先ノードで受信開始されるまでの時間) について考える．

[2]のパケット送信タイミングの変形は，このパケット転送時間に関して，モデルより求めた値と実際のシステムにおける値とで，くい違いが生じる．これを図3.5を用いて説明する．図のパケットA，Bは，共に途中3台のノードにより転送され，目的ノードへ到着するパケットである．ただし，パケットBはノード1により中継されるが，パケットAはノード1は中継しない．このとき，実際のシステムではパケットAのパケット転送時間は $3 \times T_h$ となり，パケットBの転送時間は$2 \times T_h + T_a - T_h \times (M-1)$となる(図3.5(a)参照)．一方，タイミングを変形したモデルでは，転送時間はパケットA，B共に$3 \times (T_a/M)$となる(図3.5(b)参照)．

このため，モデルから得られたパケット転送時間D_mを，次の式により実際のシステムにおけるパケット転送時間D_2に変換する必要がある（付録A参照）．

$$D_2 = \frac{M}{a} D_m + \frac{(M-2)(a-M)}{2M} T_h \qquad (3-3)$$

3.2.2 モデル化

ノードから送信されるパケットについて見ると，実際のシステムでは送信中はこのパケットの一部は次のノードにあり，残りは送信しているノードにある．しかし，この一時的に両方のノードにまたがっている状態は，モデル化を困難にするため，モデルの状態遷移の観点からは，モデルでは送信を開始すると，直ちにこのパケットは次のノードに移ると考えて，ノードの状態を変化させることにする．

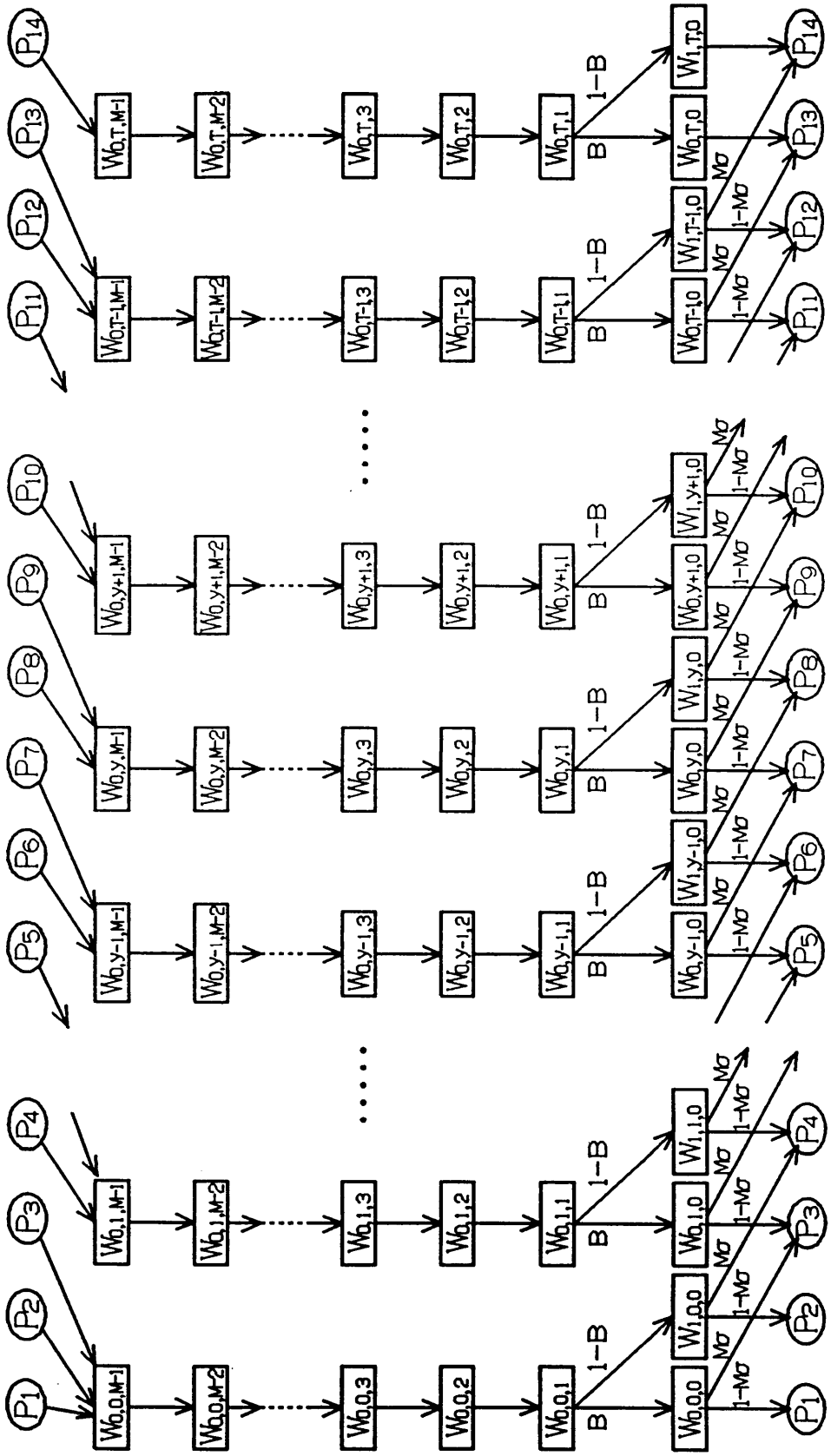


図3.6 RISリングの近似モデル
Fig.3.6 An approximate model of a RIS ring.

作成したモデルを図3.6に示す．四角で囲まれたものが状態を示す．各ノードの状態は，リングバッファ内のパケット数$x(x=0,1)$と，端末バッファ内のパケット数$y(y=0,1,\cdots,T)$，更にzステップ後$(z=0,1,\cdots,M-1)$に新たにパケット送信を開始できるという，合計3個のパラメータで表すことが出来る．モデルは，この3つのパラメータを用いた$W_{x,y,z}$の合計$(M+1)(T+1)$個のモードから構成される．

ここで，このモデルの状態の観察，パケットの発生と送信，モードの遷移のタイミングは，次のようになる．

(T1) 状態の観察は各ステップの最初に行う．

(T2) モードの遷移は各ステップの最後に行う．

(T3) $W_{x,y,0}$モードにあるノードは，現ステップの終了直前に送信を行う（但し(T2)の直前）．

(T4) $W_{x,y,0}$モードにあるノードは，現ステップの終了直前にパケットを1つ発生できる（但し(T3)の直前）．

zパラメータは，このステップの最後にパケットの送信を開始するノードが$z=0$であり，以後ノードのzの値は接続順に1から$M-1$までの値となり，同じ値を持つ複数のノードは存在しない．iステップ後に送信を開始するノードは，端末バッファ内のパケット数yに応じて図中水平に並んだ$W_{x,y,i}$のモードのいずれか一つにあることになる．そして，1ステップごとにzの値は1ずつ減り，$z=0$のノードはステップの最後にパケット送信を開始して，$z=M-1$になる．

$z=0$であるノードのパケット送信には，次の3つの場合がある．

(C1) リングバッファ，端末バッファ共に空の場合．

(C2) リングバッファは空で，端末バッファは空でない場合．

(C3) リングバッファが空でない場合．

(C1)の場合，空のパケットの送信を開始する．この場合は，$W_{0,0,0}$モードにあるノードが確率$1-M\sigma$でパケットを発生しない時起き，このノードは$W_{0,0,M-1}$モードへ遷移する（図3.6中P_1）．

(C2)の場合，端末バッファからフルパケットの送信を開始する．$W_{0,y,0}$ モード$(y=1,2,\cdots,T-1)$にあるノードが確率$1-M\sigma$でパケットを発生しなかった場合と，$W_{0,y-1,0}$モード$(y=1,2,\cdots,T)$にあるノードが確率$M\sigma$でパケットを発生

した場合，送信時に端末バッファにy個パケットがある．そこで，このノードは端末バッファからのパケット送信を開始し，$W_{0,y-1,M-1}$モードへ遷移する（図3.6中P_3, P_5, P_7, P_9, P_{11}と$W_{0,T-1,0}$からのP_{13}）．$W_{0,T,0}$モードにあるノードは，端末バッファが一杯であり，これ以上端末バッファ内のパケット数は変化しない．このため，$W_{0,T,0}$モードにあるノードは確率1で端末バッファからのパケット送信を開始し，$W_{0,T-1,M-1}$モードへ遷移する（図3.6中$W_{0,T,0}$ からのP_{13}）．

(C3)の場合は，端末バッファの状態に関わらず，リングバッファ内のパケットの送信を開始する．$W_{1,y,0}$モード$(y = 0,1,\cdots,T-1)$にあるノードが確率$1 - M\sigma$でパケットを発生しないとき，リングバッファ内のパケットを送信して$W_{0,y,M-1}$モードへ遷移する．一方このノードが確率$M\sigma$でパケットが1つ発生したときは，リングバッファ内のパケットの送信と端末バッファ内のパケット数の増加となり$W_{0,y+1,M-1}$モードへ遷移する（図3.6中P_2, P_4, P_6, P_8, P_{10}, P_{12}と$W_{1,T-1,0}$モードからのP_{14}）．また，$W_{1,T,0}$にあるノードは，これ以上端末バッファ内のパケット数は増えないため，確率1で$W_{0,T,M-1}$へ遷移する（図3.6中$W_{1,T,0}$からのP_{14}）．

この$z = 0$のモードから送信されたパケットは，隣接するノードが受信する．この受信するノードは，次に送信を開始する$z = 1$のノードであり，このステップの最後にパケットを受信し，$z = 0$のモードへ遷移する．このとき，パケットの宛先が受信したノード宛であれば，このパケットはこのノードに取り込まれるためリングバッファには入らないが（$W_{0,y,1}$モードから$W_{0,y,0}$モードへの遷移），宛先がこの受信ノード宛でなければこのパケットはリングバッファに入り，このノードのxパラメータの値は1になる（$W_{0,y,1}$モードから$W_{1,y,0}$モードへの遷移）．なお，既に述べたように，モデルにおいてはリングバッファのパケットは，送信が開始されると直ちに次のノードへ移ると考える．このため，$x = 1$となるのは$z = 0$のときに限ることに注意されたい．

ここで，この$W_{0,y,1}$からの遷移確率を求めるため，確率変数$A(\mathrm{n})$を次の様に定義する（nはシステムの状態ベクトルである．これについては，3.4.1で定義する）．

$$A(\mathrm{n}) = \begin{cases} 1 & \text{遷移の際，z=0のモードにあるノードは} \\ & \text{フルパケットを送信} \\ 0 & \text{遷移の際，z=0のモードにあるノードは} \\ & \text{空パケットを送信} \end{cases} \qquad (3-4)$$

$z = 1$のノードのリングバッファにパケットが入らないのは，$A(\mathrm{n}) = 1$（空パケットを受け取る）であるか$A(\mathrm{n}) = 0$(フルパケットを受け取る）であっても確率Eでその宛先が自ノードである場合である．このため，$W_{0,y,1}$モード$(y = 0,1,\cdots,T)$

にあるノードが$W_{0,y,0}$モードへ遷移する確率Bは，次式で与えられる．

$$B = A(\mathrm{n}) + \{1 - A(\mathrm{n})\}E \qquad (3-5)$$

逆に，$z = 1$のノードのリングバッファにパケットが入るのは，$A(\mathrm{n}) = 0$（空でないパケットを受け取る）でありかつ確率$1 - E$でその宛先が自ノードでない場合である．このため，$W_{0,y,1}$から$W_{1,y,0}$への遷移確率は$1 - B$となる．

3.3 $M > a$の場合のモデル化

3.3.1 近似

3.1で示したように，$M > a$の時システムはパケットの転送については次式で求められるg個の独立した系に分けて考えることが出来る．

$$g = \lceil M/a \rceil \qquad (3-6)$$

そこで，モデル化においてもシステムをこのg個の系に分けて各々をモデル化することとし，$M \leq a$の場合と同様にモデルの簡単化のため，次に示す作業を行う．

[1] **パケットの宛先の変形**

$M \leq a$の場合と同様，モデルにおいては仮定(A6)の代りに仮定(RI1)を用いることとする．

[2] **送信タイミングの変形**

各系のモデル化にあたっては，$M \leq a$の場合と同様，各ノードでのパケット送信開始のタイミングの遅れを同間隔にするため，タイミングの変形を行う．図3.7に$M = 8$，$a = 3$の例を示す．

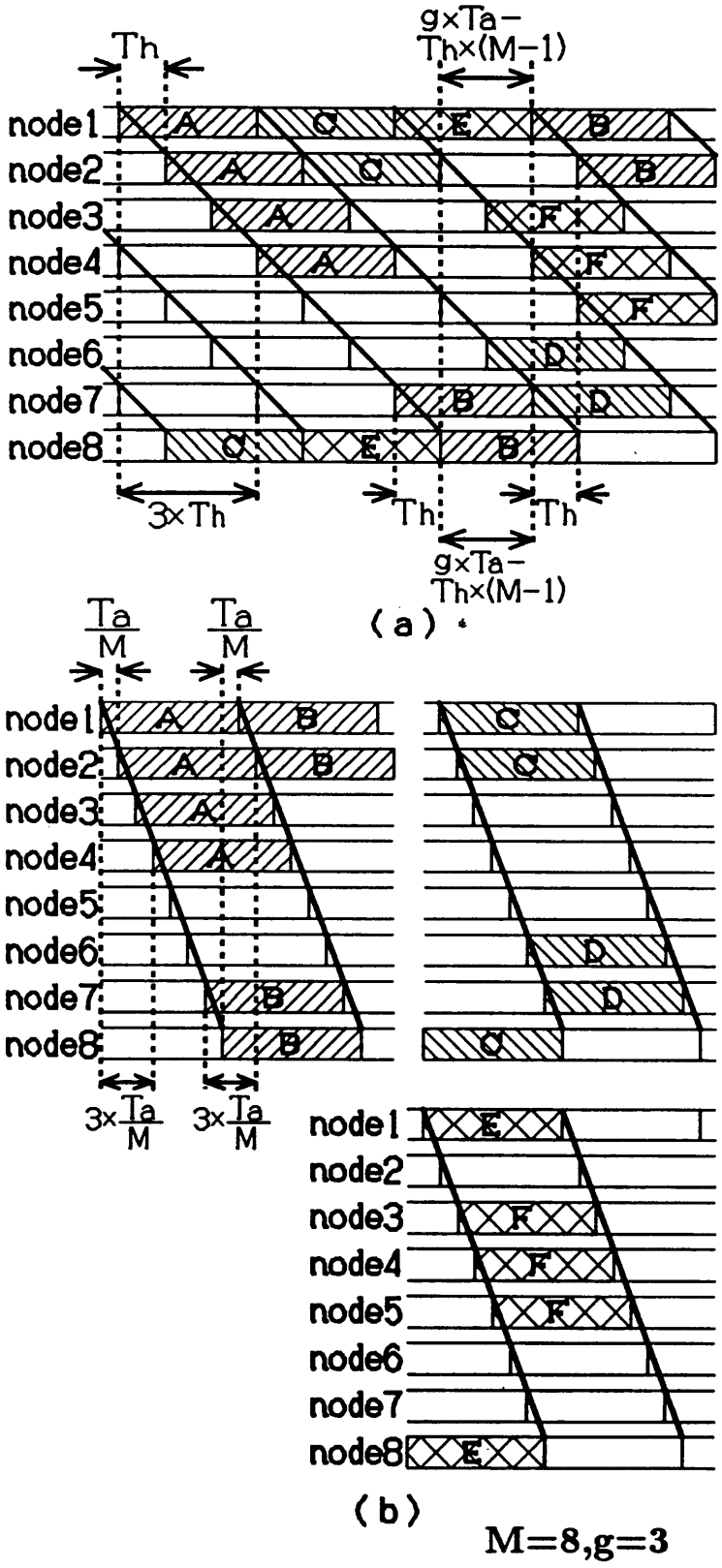


図3.7 パケット送信タイミングの変形($M > a$)
Fig.3.7 Modification of timing of sending packets in the case of $M > a$.

実際のシステム（図中(a)）では，パケット受信を開始してからパケット送信を開始するまでの時間は，ノード1のみが$g \times T_a - T_h \times (M-1)$であり，他のノードでは$T_h$である．

図中(b)に変形したタイミングを示す．システムをg個の系(ここではg=3) に分け，更にこの遅れが全て均一になるよう，遅れをT_a/Mに揃えるよう各ノードの送信タイミングを変形する．

[3] パケット発生タイミングの変形

$M \leq a$の場合と同様，各系のモデル化にあたっては，パケット発生タイミングについて，仮定(A5)の代りに，仮定(RI2)を用いる．

[4] パケット転送時間の補正

$M \leq a$の場合と同じ議論により，モデルから得られた転送時間D_mを用いて，実際のシステムにおける転送時間D_2を次のように表現することが出来る．

$$D_2 = \frac{M}{a} D_m + \frac{(M-2)(g \times a - M)}{2M} T_h \qquad (3-7)$$

$M \leq a$の時はgの値は1 $(= \lceil M/a \rceil)$ であり，$g = 1$を式(3-7)に代入すると式(3-3)と等しくなる．このことは，$M \leq a$の場合もこの式(3-7)で転送時間の補正が出来ることがわかる．

3.3.2 モデル化

各系ごとのシステムは，$M \leq a$の場合と同じ図3.6に示すモデルで表すことが出来る．各系においてノード間を転送されるパケットは系ごとに独立しており，系間をまたがって転送されることはない．このため，転送されるパケットについては，この系ごとのモデルで完全に記述できる．しかしながら，端末バッファ内のパケットについては，系ごとに独立していない．

図3.6において，$z=0$のモードにいるノードが$z=M-1$のモードに遷移するときを考える(このとき，このノードは次の系へ移る)．このとき，ノードがP_1,P_2,P_3の経路で遷移する場合は，遷移後の端末バッファが空であるため，端末バッファ内パケットの問題はない．しかし，P_4からP_{14}の経路で遷移するノードは，遷移後端末バッファにパケットが残る．ここで，この端末バッファ内のパケットは，次のパケット送信開始時点にリングバッファが空であれば送信を開始するため，この遷移は，この系の$W_{0,y,M-1}$へ遷移するのではなく，実際のシステムにおいて次に送信が開始される系の$W_{0,y,M-1}$へ遷移する．このため，この遷移による系のつながりを考慮すると，全体のモデルは，例えば$g=3$の場合，図3.8のようになる．

ここで，この図について見てみると，3つの系のモデルは同一であるため，各モデルでの平衡点においては，系間をまたがるフローは全て同じとなる．そのため，平衡点において解析するならば，次の系への出力と前の系からの入力は等しくなり，結局自己のP_4からP_{14}からの出力をそのまま入力として考えて$W_{0,y,M-1}$へ遷移させても，解析結果は同じであることになる．このことは，$M > a$の場合でも，図3.6が示す$M \leq a$の場合のモデルによって解析が出来ることを示す．

この考察と，パケットの宛先に関する確率Eの値が常に式(3-2)を用い，さらに転送時間の訂正が常に式(3-7)で可能であることから，レジスタ挿入型スロットリングの解析は，図3.6によるモデルと式(3-2) による確率Eの計算，式(3-7)による転送時間の訂正により，全ての場合において可能であることが示された．このため，以後は$M \leq a$ の場合と$M > a$の場合に分けることなく，解析を進めていくことにする．

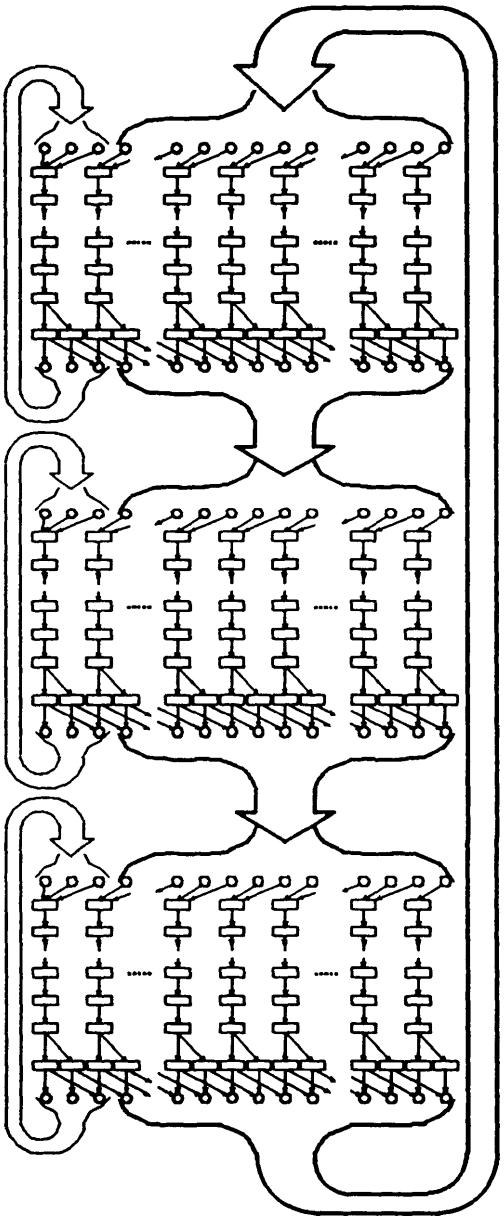


図3.8 端末バッファ内パケットの流れ
Fig.3.8 Flow of packets in terminal buffers.

3.4 モデルの解析

3.4.1 平衡点方程式

作成したモデルの解析を行う．このモデルにおいて，モード$W_{x,y,z}$にあるノー

ド数を表す確率変数を$n_{x,y,z}$とおく．モデルの定義より明らかに次の式が成り立つ．

$$\sum_{y=0}^{T} n_{0,y,0} + \sum_{y=0}^{T} n_{1,y,0} = 1 \qquad (3-8)$$

$$\sum_{y=0}^{T} n_{0,y,z} = 1 \qquad (z = 1, 2, \cdots, M-1) \qquad (3-9)$$

モデルの状態ベクトルnを次のように定義する．

$$\begin{aligned}\mathbf{n} = \{&n_{0,0,0}, n_{0,0,1}, \cdots, n_{0,0,M-1}, n_{1,0,0},\\ &n_{0,1,0}, n_{0,1,1}, \cdots, n_{0,1,M-1}, n_{1,1,0},\\ &n_{0,2,0}, n_{0,2,1}, \cdots, n_{0,2,M-1}, n_{1,2,0},\\ &\vdots\\ &n_{0,T,0}, n_{0,T,1}, \cdots, n_{0,T,M-1}, n_{1,T,0}\}.\end{aligned}$$

このベクトルnは，既約な有限状態マルコフ連鎖となる．

このマルコフ連鎖を，平衡点解析の手法を用いて解析する．平衡点解析では，システムは常に平衡点に留っていると仮定する．平衡点は，各モードにおける平均増加ノード数が0となる点と定義される[TASA86]．そのため，状態nが与えられたとき，各モードへ流入する平均ノード数と各モードから流出する平均ノード数とが等しくなるように，次の平衡点を求める式を得る．

$$Bn_{0,y,1} = n_{0,y,0} \qquad (y = 0, 1, \cdots, T) \qquad (3-10)$$

$$(1-B)n_{0,y,1} = n_{1,y,0} \qquad (y = 0, 1, \cdots, T) \qquad (3-11)$$

$$n_{0,y,z+1} = n_{0,y,z} \quad (y = 0, 1, \cdots, T)(z = 1, 2, \cdots, M-2) \qquad (3-12)$$

$$n_{0,0,0} + (1 - M\sigma)(n_{1,0,0} + n_{0,1,0}) = n_{0,0,M-1} \qquad (3-13)$$

$$\begin{aligned}&M\sigma(n_{1,y-1,0} + n_{0,y,0}) + (1 - M\sigma)(n_{1,y,0} + n_{0,y+1,0})\\ &= n_{0,y,M-1} \qquad (y = 1, 2, \cdots, T-2)\end{aligned} \qquad (3-14)$$

$$\begin{aligned}&M\sigma(n_{1,T-2,0} + n_{0,T-1,0}) + (1 - M\sigma)n_{1,T-1,0} + n_{0,T,0}\\ &= n_{0,T-1,M-1}\end{aligned} \qquad (3-15)$$

$$M\sigma n_{1,T-1,0} + n_{1,T,0} = n_{0,T,M-1} \qquad (3-16)$$

ここで，式(3–10)は$W_{0,y,0}$ $(y=0,1,\cdots,T)$に関する式であり，式(3–11)は$W_{1,y,0}$ $(y=0,1,\cdots,T)$，式(3–12)は$W_{0,y,z}$ $(y=0,1,\cdots,T)$ $(z=1,2,\cdots,M-2)$，式(3–13)は$W_{0,0,M-1}$，式(3–14)は$W_{0,y,M-1}$ $(y=1,2,\cdots,T-2)$，式(3–15)は$W_{0,T-1,M-1}$，式(3–16)は$W_{0,T,M-1}$に関する式である．

次に，ノード間を転送されていくパケットの数について考える．$z=0$のモードにあるノードは，図3.6のモデルにおいてP_1で示す遷移以外の遷移を行うときに1個パケットを次の$z=1$のモードにあるノードへ出力する．その平均数は，

$$1-(1-M\sigma)n_{0,0,0} \tag{3-17}$$

となる．このパケットは確率Eで受信したノード宛であるので，$z=1$ のモードのノード宛でないパケットの数は，

$$\{1-(1-M\sigma)n_{0,0,0}\}(1-E) \tag{3-18}$$

となる．これは，$z=1$のモードのノードのリングバッファへ入るパケットの数になる．$W_{0,y,1}$モード$(y=0,1,\cdots,T)$にあるノードが$W_{1,y,0}$モードへ遷移するとき，パケットがリングバッファへ入る．そのため，リングバッファへ入るパケット数は，

$$\sum_{y=0}^{T}(1-B)n_{0,y,1}=1-B \tag{3-19}$$

となる．

ここで，平衡点においては式(3–19)のパケット数は式(3–18)のパケット数と等しくなければならないので，次の式を得る．

$$\{1-(1-M\sigma)n_{0,0,0}\}(1-E)=1-B \tag{3-20}$$

この式(3–8)〜(3–20)を解くことで，次の平衡点の各要素の値が得られる．

$$\begin{aligned}\mathbf{n_e}=\{&n_{0,0,0e},n_{0,0,1e},\cdots,n_{0,0,M-1e},n_{1,0,0e},\\&n_{0,1,0e},n_{0,1,1e},\cdots,n_{0,1,M-1e},n_{1,1,0e},\\&n_{0,2,0e},n_{0,2,1e},\cdots,n_{0,2,M-1e},n_{1,2,0e},\\&\vdots\\&n_{0,T,0e},n_{0,T,1e},\cdots,n_{0,T,M-1e},n_{1,T,0e}\}\end{aligned}$$

なお，今後の式の展開においては$n_{x,y,ze}$を単に$n_{x,y,z}$と書く．

式(3-8)〜(3-16)より，次の式が得られる．

$$n_{0,y,0} = X^y n_{0,0,0} \qquad (y = 0, 1, \cdots, T-1) \tag{3-21}$$

$$n_{0,T,0} = \frac{M\sigma(1-B)}{B} X^{T-1} n_{0,0,0} \tag{3-22}$$

$$\begin{aligned} n_{0,y,M-1} &= n_{0,y,M-2} = \cdots = n_{0,y,2} = n_{0,y,1} \\ &= \frac{1}{B} X^y n_{0,0,0} \qquad (y = 0, 1, \cdots, T-1) \end{aligned} \tag{3-23}$$

$$n_{1,y,0} = \frac{1-B}{B} X^y n_{0,0,0} \qquad (y = 0, 1, \cdots, T-1) \tag{3-24}$$

$$\begin{aligned} n_{0,T,M-1} &= n_{0,T,M-2} = \cdots = n_{0,T,2} = n_{0,T,1} \\ &= \frac{1}{B} \frac{M\sigma(1-B)}{B} X^{T-1} n_{0,0,0} \end{aligned} \tag{3-25}$$

$$n_{1,T,0} = \frac{1-B}{B} \frac{M\sigma(1-B)}{B} X^{T-1} n_{0,0,0} \tag{3-26}$$

ここで，Xは次式で定義される．

$$X = \frac{M\sigma(1-B)}{B(1-M\sigma)} \tag{3-27}$$

そして，式(3-9)に式(3-23)と式(3-25)を代入して，次の$n_{0,0,0}$とBに関する式を得る．

$$n_{0,0,0} = \frac{B}{\frac{B(1-M\sigma)}{B-M\sigma} + \left\{ \frac{M\sigma(1-B)}{B} - \frac{M\sigma(1-B)}{B-M\sigma} \right\} X^{T-1}} \tag{3-28}$$

この式(3-28)を式(3-20)へ代入して，最後にBに関する次の式が得られる．

$$\begin{aligned} &\left(1 - \frac{B(1-M\sigma)}{\frac{B(1-M\sigma)}{B-M\sigma} + \left\{ \frac{M\sigma(1-B)}{B} - \frac{M\sigma(1-B)}{B-M\sigma} \right\} X^{T-1}} \right) \\ &\quad \times (1-E) + B = 1 \end{aligned} \tag{3-29}$$

$E \leq B \leq 1$という条件(これは，$0 \leq A(\mathrm{n}) \leq 1$ であることと式(3-5)より得られる)において，式(3-29)を満たすB すなわち平衡点におけるBの値を得ることができる．このBの値を，式(3-21)〜(3-28)に代入して，$\mathbf{n}_e$の要素が全て求まる．

3.4.2 スループット

リング型のネットワークでは，M台のノード間にM個の伝送路がある．ここで考えるスループットとは，送られるパケットのうちフルパケットが送られる割合を各伝送路ごとに求め，この平均をとったものとする．$z=0$のモードにあるノードが，フルパケットを送信し始めるのは，図3.6でP_1の経路以外で遷移した場合である．その確率は，与えられたnに対して，

$$1-n_{0,0,0}(1-M\sigma) \tag{3-30}$$

となる．この式は，Mチャネルある伝送路全てに対して成立するので，nが与えられたとき，このモデルの条件付スループット$S(\mathrm{n})$は，

$$S(\mathrm{n})=1-n_{0,0,0}(1-M\sigma) \tag{3-31}$$

となる．$\bar{S}$を$S(\mathrm{n})$のnに関する期待値とする．平衡点解析では$\bar{S}$は$S(\mathrm{n_e})$で近似するため，式(3-28)を用いて次のスループットの式が得られる．

$$\begin{aligned}\bar{S} &= 1-n_{0,0,0e}(1-M\sigma)\\ &= 1-\frac{B(1-M\sigma)}{\frac{B(1-M\sigma)}{B-M\sigma}+\left\{\frac{M\sigma(1-B)}{B}-\frac{M\sigma(1-B)}{B-M\sigma}\right\}X^{T-1}}\end{aligned} \tag{3-32}$$

3.4.3 平均パケット遅延

平均パケット遅延は，パケットが発生してから宛先ノードにそのパケットが完全に受信されるまでの平均時間と定義する．平均パケット遅延を求めるに際しては，この遅延を次の3つの要素に分けて考え，各要素ごとに求めていくこととする．

(D1) パケット発生から送信が開始されるまでの，端末バッファでの待ち時間 D_1.

(D2) 送信開始後，このパケットが宛先のノードまでの途中にあるノードを転送される時間 D_2.

(D3) 宛先のノードにてパケットを受信する時間 D_3.

このうち，D_3の時間は，Mステップと一定である．

(D1)の端末バッファでの待ち時間D_1は，モデルの解析結果を用いて，以下のようにリトルの公式により求める．

平衡点におけるシステムの端末バッファ内にある全パケット数N_1を求める．端末バッファ内のパケットは，$W_{0,y,z}$ $(y=1,2,\cdots,T)$ $(z=0,1,\cdots,M-1)$および$W_{1,y,0}$ $(y=1,2,\cdots,T)$のモードにあるノードが各々y個蓄えているので，N_1は次のようになる．

$$N_1=\sum_{y=1}^{T}\Big\{\sum_{z=1}^{M-1}yn_{0,y,ze}+y(n_{0,y,0e}+n_{1,y,0e})\Big\}$$
$$=\frac{M}{B}\sum_{y=1}^{T}yn_{0,y,0e} \qquad (3-33)$$

ここで，式(3–10)〜(3–12)を用いた．この式(3–33)に式(3–21)と式(3–22) を代入すると次式が得られる．

$$N_1=\frac{M}{B}\Big\{\frac{X-X^T}{(1-X)^2}-\frac{(T-1)X^T}{1-X}$$
$$+T\frac{M\sigma(1-B)}{B}X^{T-1}\Big\}n_{0,0,0e} \qquad (3-34)$$

端末バッファで出力を待っているパケットは，$W_{0,y,0}$ $(y=1,2,\cdots,T)$ のモードにあるノードが遷移する際に1個出力される．さらに，$W_{0,0,0}$のモードにあるノードが遷移する際にも，確率$M\sigma$でパケットが端末バッファを経て出力される（この場合は，端末バッファ内での待ち時間は0 ということになる）．このため，端末バッファ内のパケットから各ステップごとに出力されるパケットの個数λ_1は次のように与えられる．

$$\lambda_1=M\sigma n_{0,0,0e}+\sum_{y=1}^{T}n_{0,y,0e} \qquad (3-35)$$

ここへ，式(3–21)と式(3–22)を代入して次の式を得る．

$$\lambda_1=\Big\{M\sigma+\frac{X-X^T}{1-X}+\frac{M\sigma(1-B)}{B}X^{T-1}\Big\}n_{0,0,0e} \qquad (3-36)$$

さらに，端末バッファへのパケットの発生を，$z=0$の時にMステップ分まとめて発生させているので，実際の待ち時間はこのモデルから得られる値に，$M/2$ステップ加算する必要がある．

平衡点解析では，D_1で表される平均待ち時間は，平衡点における待ち時間で近似する．以上から，端末バッファ内での平均待ち時間D_1は，リトルの公式を用いて，次の式で与えられる．

$$D_1=\frac{N_1}{\lambda_1}+\frac{M}{2} \qquad (3-37)$$

次に，(D2)の時間D_2を求める．送信を開始したパケットが，宛先のノードに到着するまでに，途中中継されるノード数の平均をGとする．Gは，パケットの宛先が仮定(RI1)の幾何分布で表されることから，次の式で与えられる．

$$G = \sum_{i=0}^{\infty} iE(1-E)^i = \frac{1-E}{E} \qquad (3-38)$$

1台のノードにより中継される際に要する時間は1ステップである．更に式(3-7)を用いた補正を行い，実際のシステムにおける(D2)の平均パケット遅延D_2を与える次の式が得られる（単位：ステップ）．

$$D_2 = \frac{M}{a} \times \frac{1-E}{E} + \frac{(M-2)(g \times a - M)}{2M} \times \frac{M}{a} \qquad (3-39)$$

全体のパケット遅延Dは，上記の(D1)(D2)(D3)の各要素の合計として，次の式で与えられる．

$$D = D_1 + D_2 + M \qquad (3-40)$$

3.4.4 端末バッファオーバフロー確率

パケットが端末で発生したときに，端末バッファが一杯で，このパケットが破棄される確率を，端末バッファオーバフロー確率P_{ov}と定義する．

モデルにおいては，このオーバフローはノードが$W_{0,T,0}$もしくは$W_{1,T,0}$のモードにあるときに起きる．このため，モデルから次のP_{ov}を求める式が得られる．

$$P_{ov} = n_{0,T,0e} + n_{1,T,0e} \qquad (3-41)$$

3.4.5 端末バッファ内パケット数

各ノードの端末バッファに蓄えられているパケットの数について検討する．送信開始直前の時点での端末バッファ内パケット数を考えることで，与えられた条件(ノード数，端末バッファ段数，パケット発生確率) における端末バッファ内のパケット数の最大値を得ることが出来る．以後，端末バッファ内パケット数とは，この送信開始直前の時点における端末バッファ内パケット数を指すこととする．

モデルにおいて，送信開始直前とは$z=0$のノードが遷移する直前である．ノードが$z=0$のモードにあるときに，ノードの端末バッファ内のパケット数がkである確率を$P[\,k\,]$とする．

$k=0$であるのは，このノードが$W_{0,0,0}$モードか$W_{1,0,0}$モードにあり，確率$1-M\sigma$で端末バッファにパケットを発生しない場合である．そこで，$P[\,0\,]$は次のようになる．

$$P[0]=(1-M\sigma)(n_{0,0,0e}+n_{1,0,0e})=(1-M\sigma)n_{0,0,1e} \qquad (3-42)$$

$k=i\ \ (i=1,2,\cdots,T-1)$であるのは，ノードが$W_{0,i-1,0}$モードか$W_{1,i-1,0}$のモードにあり，確率$M\sigma$で端末バッファにパケットが発生する場合（但し，$W_{0,i-1,0}$モードの場合は，端末バッファに一時的に$i$個のパケットを持つが，次のステップでは$i-1$個になる）と，ノードが$W_{0,i,0}$か$W_{1,i,0}$のモードにあり，確率$1-M\sigma$で端末バッファにパケットが発生しない場合である．そこで，$P[\,i\,]$は次のようになる．

$$\begin{aligned} P[\,i\,] &= M\sigma(n_{0,i-1,0e}+n_{1,i-1,0e})+(1-M\sigma)(n_{0,i,0e}+n_{1,i,0e})\cdot \\ &= M\sigma n_{0,i-1,1e}+(1-M\sigma)n_{0,i,1e} \end{aligned} \qquad (3-43)$$

$k=T$であるのは，ノードが$W_{0,T-1,0}$か$W_{1,T-1,0}$のモードにあり，確率$M\sigma$で端末バッファにパケットが発生する場合と，ノードが$W_{0,T,0}$か$W_{1,T,0}$のモードにある場合である（後者の場合は，既に端末バッファが一杯であり，これ以上パケットはバッファに入らない）．そこで，$P[\,T\,]$は次のようになる．

$$\begin{aligned} P[T] &= M\sigma(n_{0,T-1,0e}+n_{1,T-1,0e})+(n_{0,T,0e}+n_{1,T,0e}) \\ &= M\sigma n_{0,T-1,1e}+n_{0,T,1e} \end{aligned} \qquad (3-44)$$

式(3-42)〜(3-44)を用いると，端末バッファ内パケットの平均値と分散を求めることができる．

3.5 数値例

ここでは，これまでに検討したモデルによる解析結果をシミュレーションによる結果と比べると共に，レジスタ挿入型スロットリングの性能について論じる．

ここでは，ノード数Mを10, 30, 100とし，T_aとT_hの比である値aを15として解析する．またシミュレーションは，仮定(A1)〜(A6)を用い，その他の近似のための仮定は用いていない．シミュレーション時間は3.5.2を除き330,000Mステップとする．さらに，シングルラン法[KOBA78] により，95%の信頼区間を求める．しかし，以後示す各図においてはいずれも，シミュレーション値を示す記号（○△□▽）の中に，この95%の信頼区間の範囲が含まれてしまうため，特にグラフには記入しない．

3.5.1 スループットと平均パケット遅延

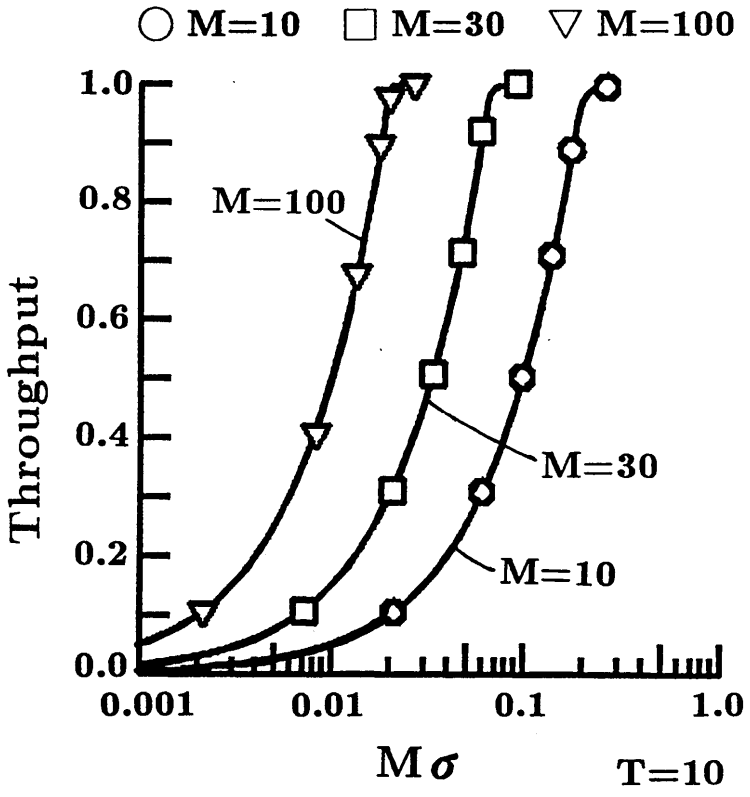


図3.9 スループットと$M\sigma$の関係($T = 10$)
Fig.3.9 Throughput versus $M\sigma$ for $T = 10$.

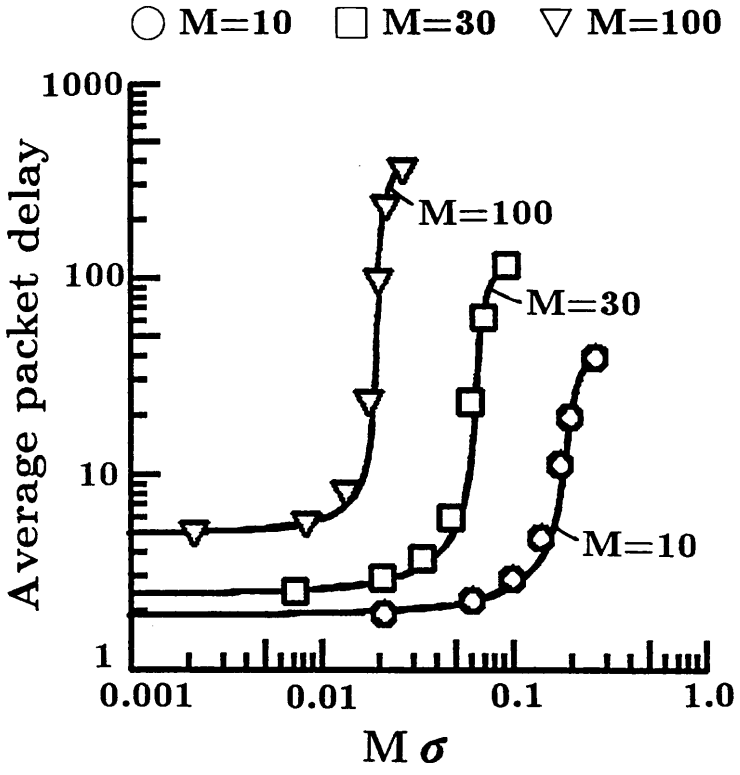


図3.10 平均パケット遅延と$M\sigma$の関係($T = 10$)
Fig.3.10 Average packet delay versus $M\sigma$ for $T = 10$.

端末バッファの段数$T = 10$の場合について，$M\sigma$とスループットとの関係を図3.9に，$M\sigma$と平均パケット遅延（パケット伝送時間で正規化）との関係を図3.10に，スループットと平均パケット遅延との関係を図3.11に示す．図中実線が解析によって得られた値を示し，○□▽がそれぞれ$M = 10, 30, 100$のシミュレーション結果を示す．図3.11から，スループットが0.5以下の時には解析値とシミュレーション結果とは良く一致していることがわかる．高負荷時においては，平均パケット遅延の解析値はシミュレーション値に比べてやや低くなる．これは，高負荷時に発生する端末バッファのオーバフローにより，平衡点解析におけるポテンシャルの井戸[TASA86]が非対称になることが原因と考える．

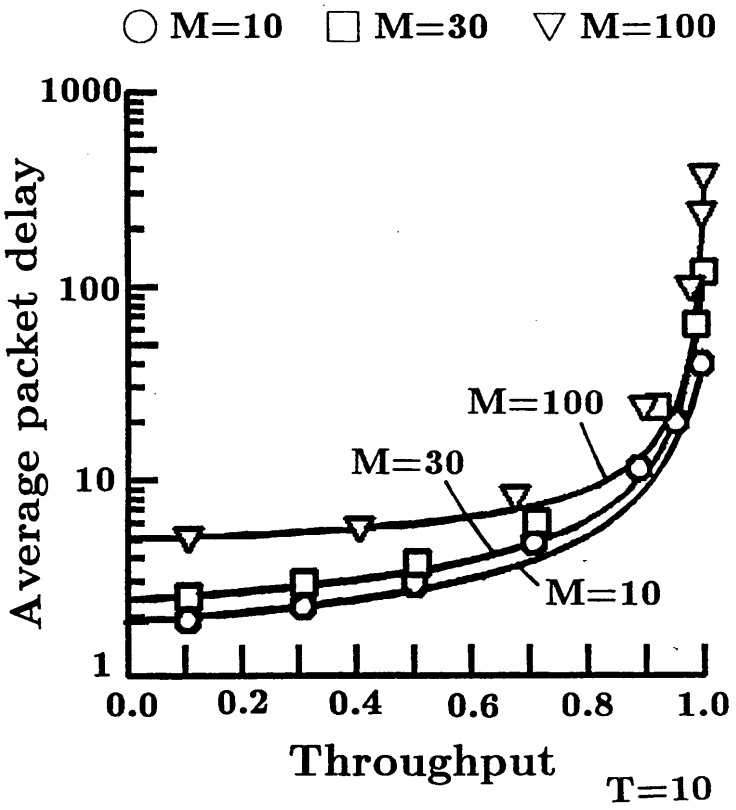


図3.11 平均パケット遅延とスループットの関係($T = 10$)
Fig.3.11 Average packet delay versus throughput for $T = 10$.

次に，$M = 30$として端末バッファ数Tが1の場合と10の場合について，$M\sigma$とスループットとの関係を図3.12に，$M\sigma$と平均パケット遅延との関係を図3.13に示す．図中実線が解析によって得られた値を示し，○△がそれぞれ$T = 1, 10$のシミュレーション結果を示す．解析値はシミュレーション結果と良く一致している．

図3.12と図3.13から，$T = 10$の時の方が$T = 1$の時に比べ，スループット，平均パケット遅延共に大きいことがわかる．これは，端末バッファサイズの増加が，端末バッファのオーバフロー発生確率を低下させ，この結果パケットが端末バッファ内で待つ時間を大きくしたことを示している．そこで，次にTの値を変えたときの端末バッファにおけるオーバフロー確率を見ることとする．

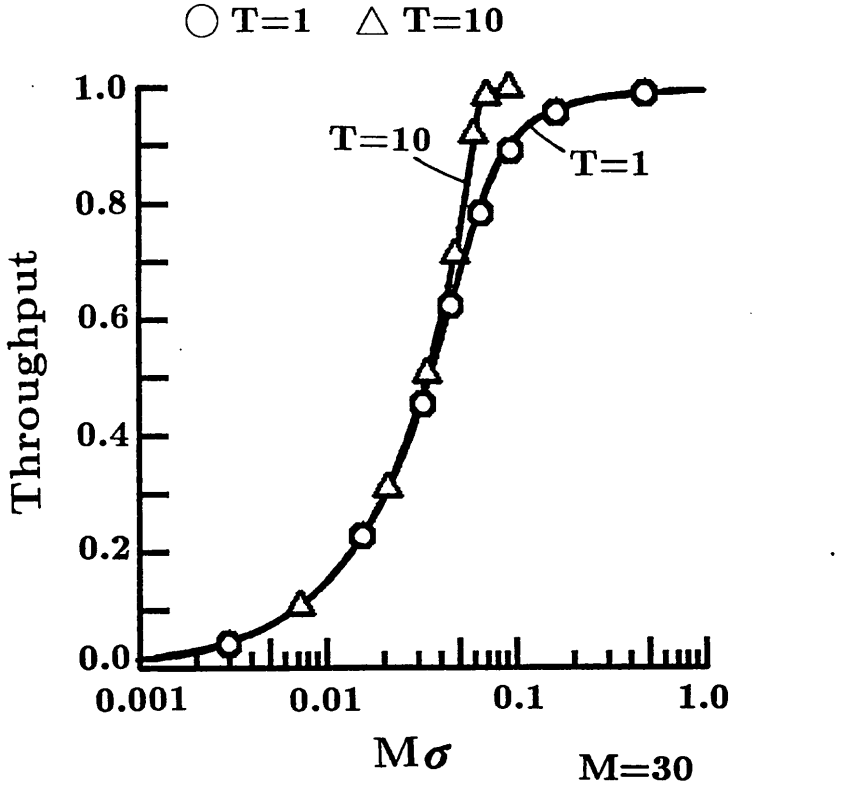


図3.12 スループットと$M\sigma$の関係($T = 1, T = 10$)
Fig.3.12 Throughput versus $M\sigma$ for $T = 1$ and $T = 10$.

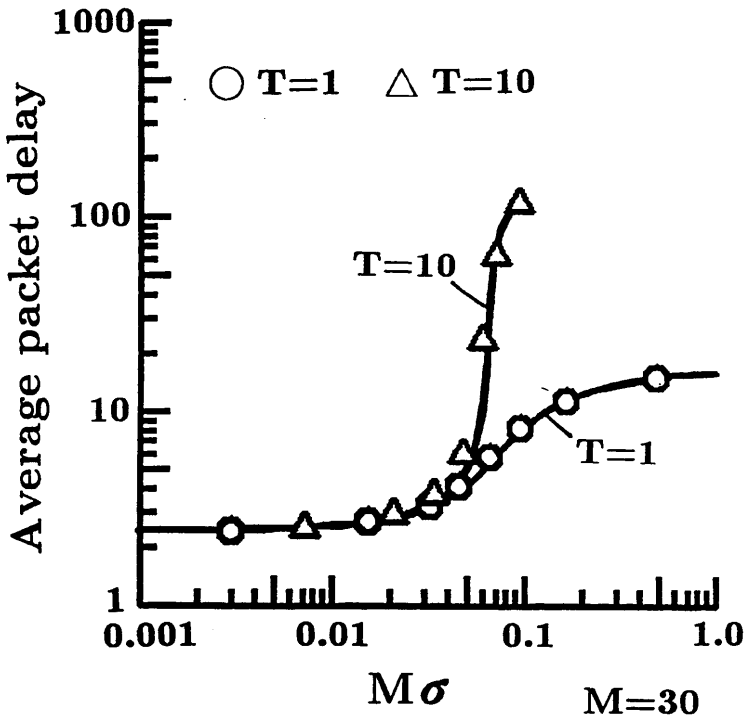


図3.13 平均パケット遅延と$M\sigma$の関係($T = 1$,$T = 10$)
Fig.3.13 Average packet delay versus $M\sigma$ for $T = 1$ and $T = 10$.

3.5.2 端末バッファオーバフロー確率

図3.14に，$M = 30$のときの$M\sigma$と端末バッファのオーバフロー確率の関係を，端末バッファの段数Tの値が2, 3, 5, 10の場合について，それぞれ示す．図中実線が解析によって得られた値を示し，記号○△□▽はそれぞれ$T = 2, 3, 5, 10$の場合のシミュレーション結果を示す．ただし，この時のシミュレーション時間は，オーバフロー確率が小さい場合でも求められる様に，他より長い120,000,000ステップとする．このため，確率が1.0×10^{-6}以上の場合について，求めることができた．解析値とシミュレーション結果とは，ほぼ一致している．

オーバフローしたパケットは破棄されるため，ここで示すオーバフロー確率とパケットの破棄率とは一致する．このパケット破棄率をデータや映像のパケットに求められる1.0×10^{-9} 以下[VERB87]とするためには，$T = 2, 3, 5, 10$の場合それぞれ$M\sigma$の値を0.0015($\sigma = 0.00005$)，0.0078($\sigma = 0.00026$)，0.026($\sigma = 0.00087$)，0.048($\sigma = 0.0016$) 以下にする必要があることがわかる．

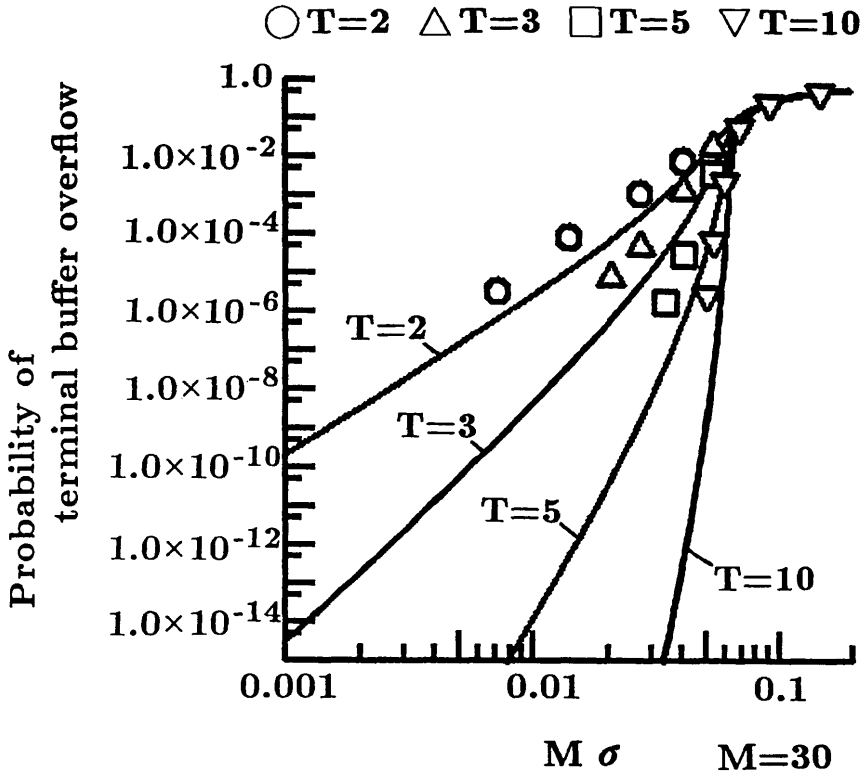


図3.14 端末バッファオーバフロー確率と$M\sigma$の関係($M = 30$)
Fig.3.14 Probability of terminal buffer overflow versus $M\sigma$ for $M = 30$.

3.5.3 端末バッファ内パケット数

$T = 10$の場合，$M\sigma$と端末バッファ内パケット数の平均との関係を図3.15に，$M\sigma$と端末バッファ内パケット数の分散との関係を図3.16に示す．図中実線が解析によって得られた値を示し，○□▽がそれぞれ$M = 10, 30, 100$ のシミュレーション結果を示す．

平均，分散とも，$M\sigma$が増えるにつれて増大する．しかし，高負荷時において分散の値が逆に減少する．これは，オーバフローが発生し，バッファ内パケット数がバッファ段数付近に偏ってきた事を表している．

平均，分散ともに解析値とシミュレーション値が良く一致している．特に分散が高負荷時に減少する部分においても，解析値はシミュレーション値と同様の傾向を正しく示している．このことから，本解析を用いて，端末バッファ内パケット数について議論できることがわかる．

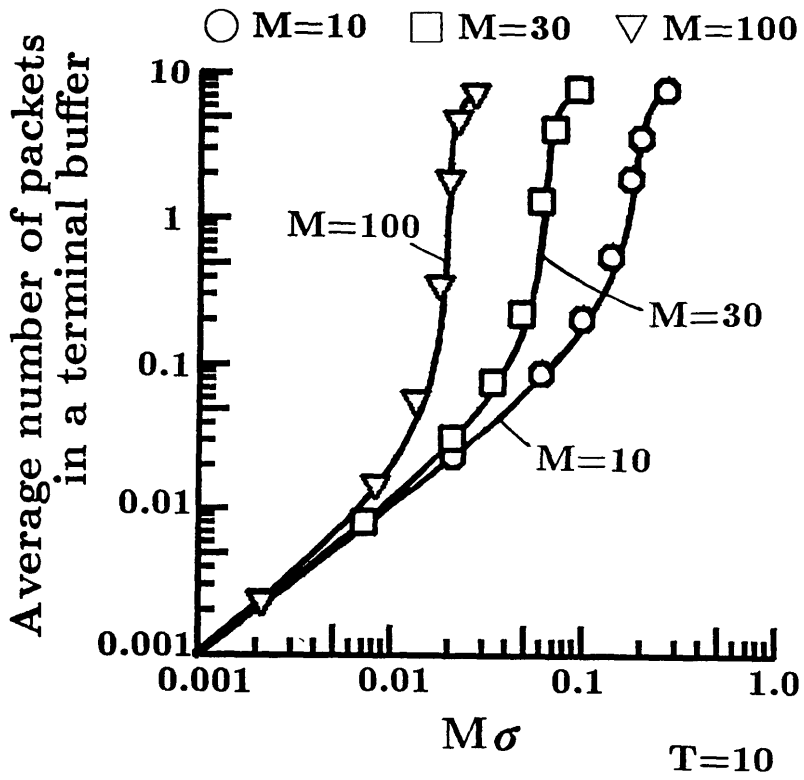


図3.15 端末バッファ内パケット数の平均と$M\sigma$の関係($T = 10$)
Fig.3.15 Average number of packets in a terminal buffer versus $M\sigma$ for $T = 10$.

付録A 式$(3-3)$の導出

まず，パケットがノード1により中継されない場合，実際のシステムにおける転送時間D_{21}をD_bから求める式は，次のようになる．

$$D_{21} = \frac{T_h}{\frac{T_a}{M}} D_m = \frac{M}{a} D_m \qquad (A-1)$$

次に，パケットがノード1により中継される場合，実際のシステムにおける転送時間D_{22}をD_mから求める式は，次の様になる．

$$\begin{aligned} D_{22} &= (D_m - \frac{T_a}{M})\frac{M}{a} + T_a - (M-1)T_h \\ &= \frac{M}{a} D_m + (a-M)T_h. \end{aligned} \qquad (A-2)$$

ここで，任意のパケットがノード1により中継される確率P_pを求める．仮定(A6)より，実際のシステムにおいて各ノードが生成するパケットの宛先は，自ノード以外の全てのパケットに対し一様であるので，M台ある各ノードは宛先が$M-1$通り

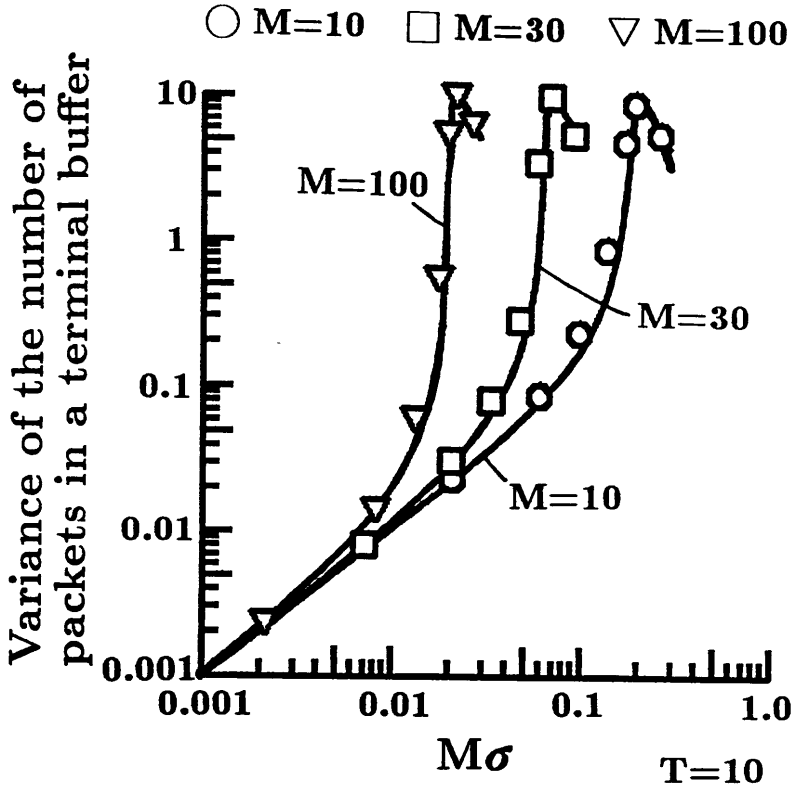


図3.16 端末バッファ内パケット数の分散と$M\sigma$の関係($T = 10$)
Fig.3.16 Variance of the number of packets in a terminal buffer versus $M\sigma$ for $T = 10$.

のパケットを生成し，ノード全体でパケットは$M(M-1)$通りとなる．このうち，ノード1により中継されるパケットの種類の数を考える．ノード1とノード2が出すパケットには，ノード1により中継されるものは無い．ノード3が出すパケットでは，ノード2宛のパケット1種類だけがノード1により中継される．ノード4が出すパケットでは，ノード2とノード3宛の2種類がノード1により中継される．このようにして，最後のノードMが出すパケットでは，ノード2, 3, 4, ⋯, $M-1$ 宛の$M-2$種類のパケットがノード1により中継される．このことと，仮定(A6)より各パケットは同確率で発生することから，P_pは次のようになる．

$$P_p = \frac{\sum_{k=3}^{M}(k-2)}{M(M-1)} = \frac{M-2}{2M}. \qquad (A-3)$$

D_mからD_2への変換は，式(A-1)で行う場合の確率が$1-P_p$であり，式(A-2)で行う場合の確率がP_pであるので，次式が得られる．

$$D_2 = D_{21}(1-P_p) + D_{22}P_p. \qquad (A-4)$$

式(A–1)〜(A–3)を式(A–4)に代入して，式(3–3)が得られる．

第4章 トークンリングの性能解析

4.1 トークンリング

本章で解析するトークンリングの動作を説明する．このトークンリングは，文献[TROP81]に基づくものである．

各ノードに接続した端末から発生するパケットは，ノード内の端末バッファと呼ぶバッファに蓄えられる．リングに接続されている全てのノードが送信すべきパケットを持っていないときは，1つのフリートークンが循環している．パケットを持つノードは，このフリートークンの受信を待ち，受信後，ビジートークンに続けてパケットを1個のみ送信する．そして，パケットの送信を終了した後，フリートークンを次のノードへ送る．他のノードは，ビジートークンとこれに続けて受信するパケットを次のノードへ転送する．この際パケットの宛先アドレスを調べ，自ノード宛であればパケットを複製して，ノードに接続した端末へ送る．パケットの送信元のノードは，自分が送信したビジートークンとそれに続くパケットがリングを一周し受信したとき，リングから取り除く(図4.1参照)．

解析に際しては，フリートークンとビジートークンは同じ長さで，ノードが受信に要する時間をL_tとする．パケットは固定長で，ノードが受信に要する時間をT_aとする．

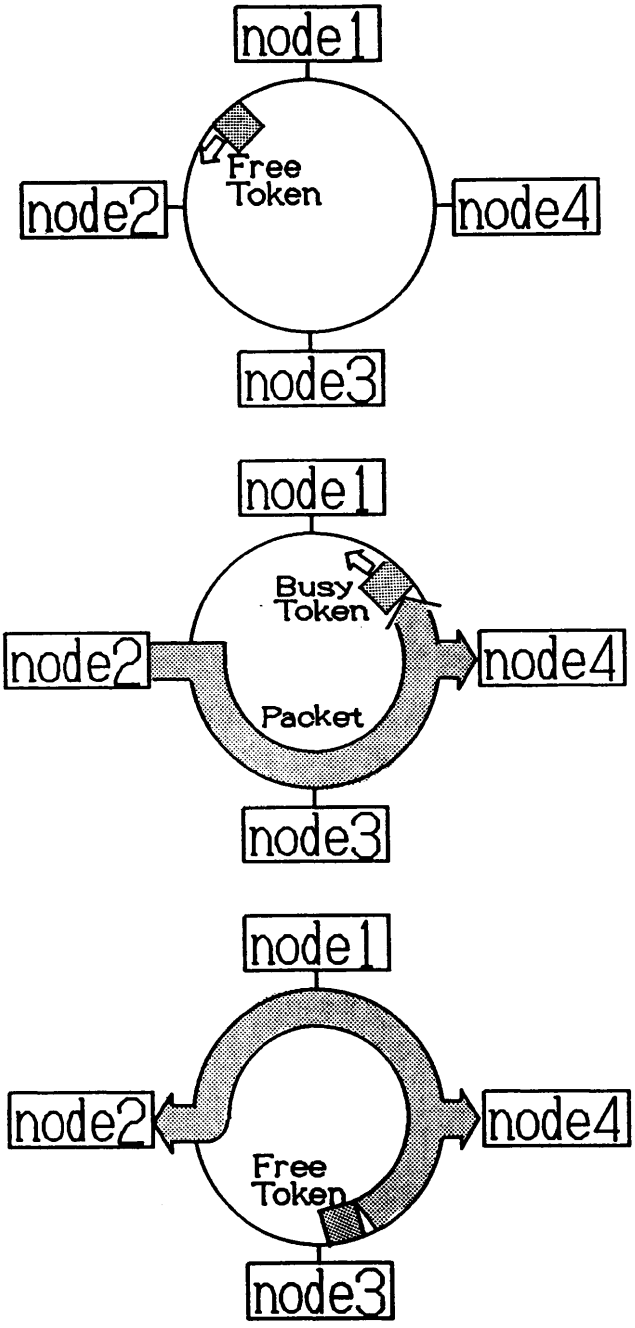


図4.1 トークンリング
Fig.4.1 Token ring.

4.2 トークンリングのモデル化

トークンリングを平衡点解析の手法で解析するため，ネットワークにおけるM台のノードの各々がとりうる状態をモードとして表し，このモード間の遷移の関

係を表現した離散時間のマルコフモデルを作成する．

まず，トークンリングのモデル化と解析を容易にする目的で導入した仮定を説明する．

導入した仮定は，パケット長とトークン長との関係に関するものと，モデルの時間単位(モデルの状態を観察する周期)に関するものである．これらの仮定をもとに近似モデルIを作成する．更に，このモデルに対してパケット発生についての仮定を置き，遷移が少なく解析可能な近似モデルIIを作成する．

なお，ここで述べる各仮定は，あくまでもモデル上の仮想的なものであり，実際のシステムの動作に関するものではないことに，注意されたい．

4.2.1 パケット長の仮定と時間単位の変形

解析を容易にする目的で，次の仮定を置く．

(TO1) パケット長は，トークン長の整数倍である．つまり，$H = T_a/L_t$を満たす整数Hが存在する．

また，モデルの時間単位を，ステップ(T_a/M)とするよりも，トークン送信に要する時間L_tとした方が，モデルが簡単になる．これは，各時間単位ごとにフリートークンがノード上を接続順に転送されていくことになるためである．そこで，仮定(A2)の代りに時間単位をL_tとする次の仮定を導入し，モデルを作成する．

(TO2) 時間をL_tごとの時間単位(この単位を“トークン時間”と呼ぶ)に細分化する．

この時間単位の変形により，仮定(A5)のステップごとのパケット発生に関する仮定を，次のトークン時間ごとのパケット発生の仮定(TO3)で置き換える必要がある．

(TO3) 各端末はトークン時間当り確率λでパケットを1つ生成し，端末バッファに蓄える．ここで，$\lambda = \sigma L_t M/T_a$である．

λの値は，σにトークン時間とステップ時間の比を乗じたものである．トークン時間がステップの時間の2倍以上の場合もあるが，$\sigma \ll 1$であるため，1トークン時間に2パケット以上発生する確率を0とした．

4.2.2 近似モデルI

これまでの仮定に基づき作成したモデルを図4.2に示す．各ノードが取るモードは，フリートークンの受信を待つかフリートークンを転送中であるWモードと，フリートークンを持ちパケットを送信中であるSモードとに分れる．

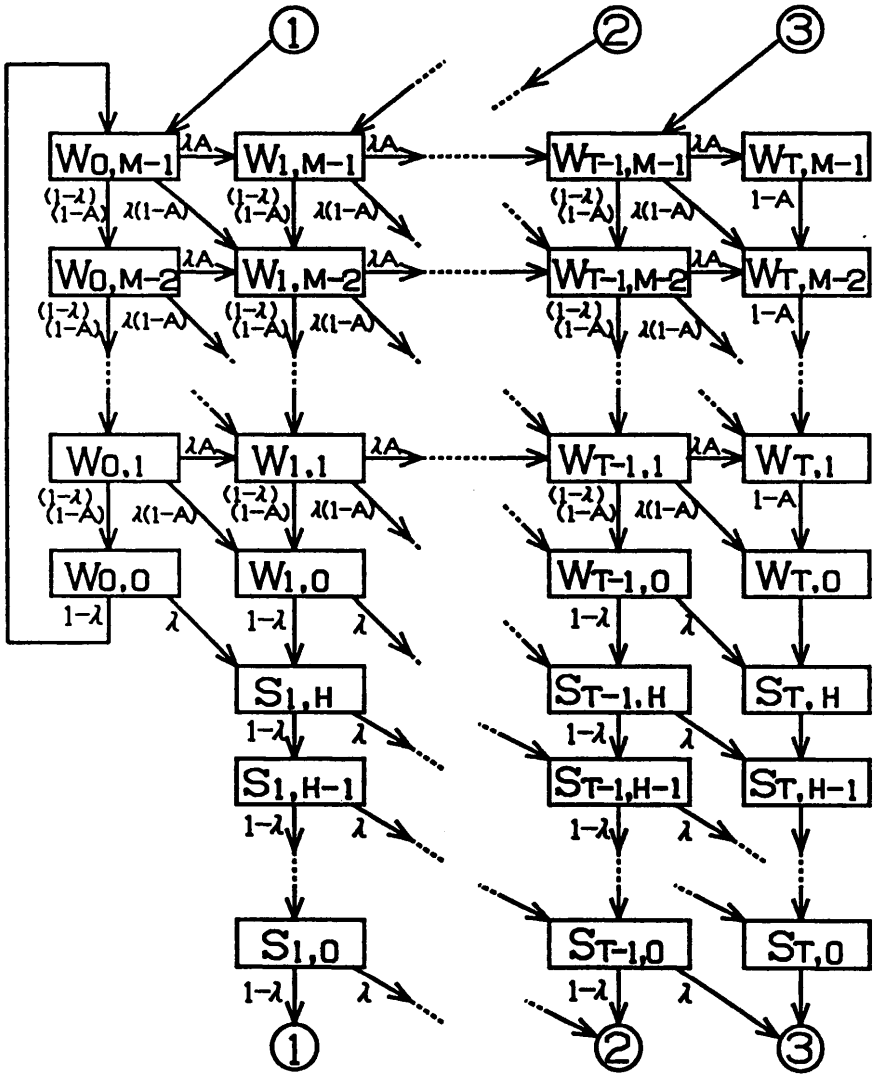


図4.2 トークンリングの近似モデルI
Fig.4.2 Approximate model II of a token ring.

ここで，このモデルの状態の観察，パケットの発生と送信，モード遷移のタイミングを，次のようにする．

(T1) 状態の観察は各トークン時間の最初に行う．

(T2) モードの遷移は各トークン時間の最後に行う.

(T3) トークンとパケットの送信は，各トークン時間の最後に行う(但し(T2)の直前).

(T4) 端末からのパケット発生は各トークン時間の最後に起きる(但し(T3)の直前).

次に，確率変数$A(\mathrm{n})$を定義する．ここでnは，4.5で定義するモデルの状態ベクトルである.

$$A(\mathrm{n}) = \begin{cases} 1 & \text{フリートークンを持つノードは，遷移時にトークンを持ち続ける} \\ 0 & \text{フリートークンを持つノードは，遷移時にトークンを送信する} \end{cases} \qquad (4-1)$$

$A(\mathrm{n}) = 1$であるときは，フリートークンを持っているノードは，遷移時にパケットの送信を開始するか送信中のどちらかである．$A(\mathrm{n}) = 0$ であるとき，フリートークンを持っているノードは，遷移時にフリートークンを受信しても送信パケットが無いか，前回の遷移時まで送信中であったパケットの送信を今回の遷移時に終了するかのどちらかである．確率変数$A(\mathrm{n})$の値は，モデルを解析することにより求める．なお図4.2では，この確率変数$A(\mathrm{n})$を，簡単のために単にAと書いた.

Wモードにあるノードの状態は，2つのパラメータで決まる．一つは，そのノード内の端末バッファにv個 $(v = 0, 1, \cdots, T)$ のパケットがあるというvパラメータである．もう一つは，パケットの流れと逆の方向に，リング上をxノード$(x = 0, 1, \cdots, M-1)$さかのぼったノードが，現在フリートークンを持っているというxパラメータである．すなわち，xパラメータはあと何ノードで自分にフリートークンがまわって来るかを表すパラメータである．vパラメータの値がi，xパラメータの値がjのWモードを，$W_{i,j}$と書く.

Sモードにあるノードの状態も，2つのパラメータにより決まる．一つは，送信中のパケットを含めて端末バッファ内にy個$(y = 1, 2, \cdots, T)$ のパケットを有するというyパラメータである．もう一つは，パケットの送信を完了するまでに，あとzトークン時間$(z = 0, 1, \cdots, H)$要するというzパラメータである．yパラメータの値がi，zパラメータの値がjのSモードを，$S_{i,j}$と書く.

$x = 0$のWモードにあるノードは，フリートークンを持っている．このとき，ノードが$W_{0,0}$モードにあって遷移時に確率$1-\lambda$でパケットが発生しない場合，次のノードへフリートークンの送信を開始し，$W_{0,M-1}$モードへ遷移する．一方，ノードが$W_{i,0}(i \neq 0)$モードにあるか，$W_{0,0}$モードにあっても遷移時に確率λでパケットが発生した場合，ビジートークンの送信を開始してSモードへ遷移する．ここでSモー

ドへの遷移は，ノードが$W_{i,0}(i=1,2,\cdots,T)$にあって確率$1-\lambda$でパケット発生がなかった場合と，ノードが$W_{i-1,0}$にあって確率λでパケット発生があった場合，共に遷移時にi個のパケットを有しており，$S_{i,H}$モードへ遷移する．また，ノードが$W_{T,0}$モードにあるときは，パケットが発生してもオーバフローとなってこれ以上パケットは増えず，確率1で$S_{T,H}$へ遷移する．

パケット送信中を表すSモードにあるノードのzパラメータの値は，トークン時間ごとに1ずつ減る．その際，確率λでパケットが発生すると，yパラメータの値が1増える．但し，$y=T$であるノードでは，パケットが発生してもオーバフローとなってこれ以上パケットは増えず，yパラメータの値は変化しない．ノードが$S_{y,0}$のモードにある時は，パケットの送信を全て完了しており，次の遷移でフリートークンの送信を開始して，$W_{y,M-1}$又は$W_{y-1,M-1}$モードへ遷移する．

$x\neq 0$のWモードにあるノードは，遷移時に確率$1-A(\mathrm{n})$でxパラメータの値を1ずつ減らす．これは，フリートークンを持っているノードがそのトークンを次ノードへ送信する場合に対応する．一方，確率$A(\mathrm{n})$でフリートークンの送信が無い場合は，xパラメータの値は変らない．また，確率λでパケットが発生すると，vパラメータの値は1増える．しかし，vパラメータの値がTのときはパケットが発生してもその値は変化しない．

4.2.3 パケット発生の変形

図4.2のモデルは，各トークン時間でのパケットの発生を許すものであるため，モード間の遷移が多く，その解析が困難である．そこで，パケット発生の仮定(TO3)を次に示す仮定(TO4)と(TO5)で更に置き換え，遷移の少ない近似モデルIIを作成する．

まず，遷移時にxパラメータの値が変化しない場合($A(\mathrm{n})=1$のとき)，$x\neq 0$のWモードにあるノードのパケット発生について考える．この場合，xパラメータの値は$1\sim M-1$の$M-1$通りあるが，その各々の場合のパケット発生を$x=M-1$の時にまとめる．すなわち，

(TO4) $W_{v,M-1}$モードにあるノードは，$A(\mathrm{n})=1$の時，遷移時に確率$(M-1)\lambda$で1パケット発生する．

次に，$A(\mathrm{n})=0$の場合のWモードにあるノード(この場合，遷移時にxパラメータの値が1減る)について，パケット発生を考える．xパラメータの値は$0\sim M-1$

のM通りある．このM通りの各々のパケット発生を次のように$x = 0$の時にまとめる．

(TO5–1) $W_{v,0}$モードにあるノードは，遷移時に確率$M\lambda$で1パケット発生する．

Sモードにいるノードは，$H+1$回遷移後Wモードに戻る．そこで，上記と同様に各遷移時のパケット発生を次のように$W_{v,0}$モードからSモードへ遷移する時にまとめる．

(TO5–2) $W_{v,0}$モードにあるノードがSモードへ遷移する場合，遷移時に確率$(H+1)\lambda$で1パケット発生する．

(TO5–1)と(TO5–2)は，共に$W_{v,0}$モードにあるときのパケット発生に関する仮定である．そこで，これら2つの仮定を一つにまとめる．

$W_{i,0}$モード$(i \neq 0)$のときは，既に端末バッファにパケットを持つため，確率1でSモードへ遷移する．このとき更にパケットが発生しない確率は，(TO5–1)と(TO5–2)より，$(1-M\lambda)\{1-(H+1)\lambda\}$である．また，パケットが1つ発生する確率は$(1-M\lambda)(H+1)\lambda + M\lambda\{1-(H+1)\lambda\}$，パケットが2つ発生する確率は$M(H+1)\lambda^2$となる．ここで，$\lambda \ll 1$であるため$\lambda^2 = 0$と近似し，改めてパケットが発生しない確率を$1-(M+H+1)\lambda$，パケットが1つ発生する確率を$(M+H+1)\lambda$，パケットが2つ以上発生する確率を0と近似する．

$W_{0,0}$モードのときは，(TO5–1)により確率$M\lambda$でパケットが1つ発生し，Sモードへ遷移する．このSモードへ遷移する場合は，(TO5–2)より確率$(H+1)\lambda$でパケットが更に1つ発生し，結局確率$M(H+1)\lambda^2$でパケットが2つ発生する．しかし，$W_{i,0}$モードの場合と同様に，$\lambda^2 = 0$と近似し，パケットが1つ発生する確率を改めて$M\lambda$と近似する．

以上の議論により，仮定(TO5–1)と(TO5–2)をまとめ，次の仮定(TO5)を置く．

(TO5) $W_{0,M-2} \sim W_{0,1}$モードにあるノードは，パケットを発生しない．一方，$W_{0,0}$モードにあるノードは，遷移時に確率$M\lambda$でパケットを1つ発生する．また，$W_{i,M-2} \sim W_{i,1}(i \neq 0)$モードにあるノードも，パケットを発生しない．しかし，$W_{i,0}$モードにあるノードは，遷移時に確率$(M+H+1)\lambda$でパケットを1つ発生する．

このパケット発生タイミングの近似により，遷移が少ない図4.3の近似モデルIIを得る．

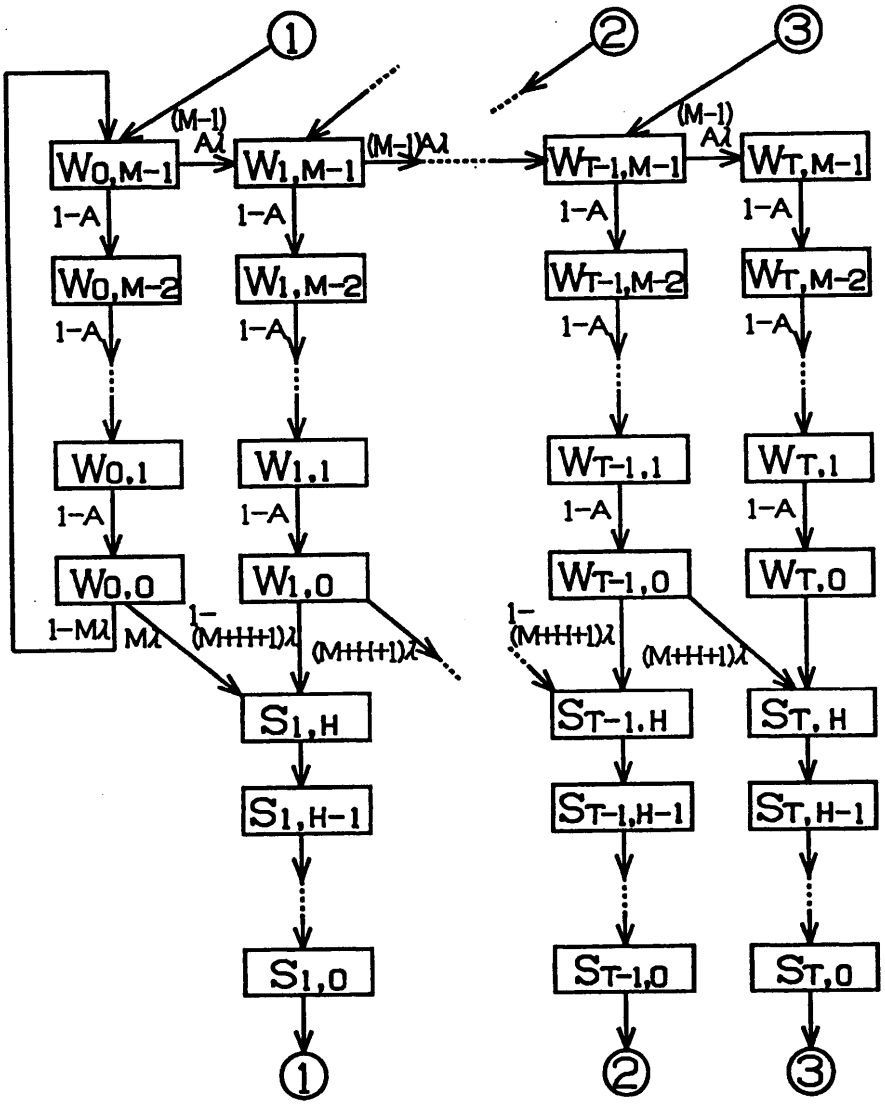


図4.3 トークンリングの近似モデルII
Fig.4.3 Approximate model II of a token ring.

このモデルにより各ノードが遷移する様子を図4.4に示す(図では$M=3, H=2$). 図は各トークン時間ごとに送信されるフリートークン(FT)とビジートークン(BT)及びパケット(PACKET)とモードを各ノード別に書き，▼はノードに接続した端末がパケットを発生したことを示している.

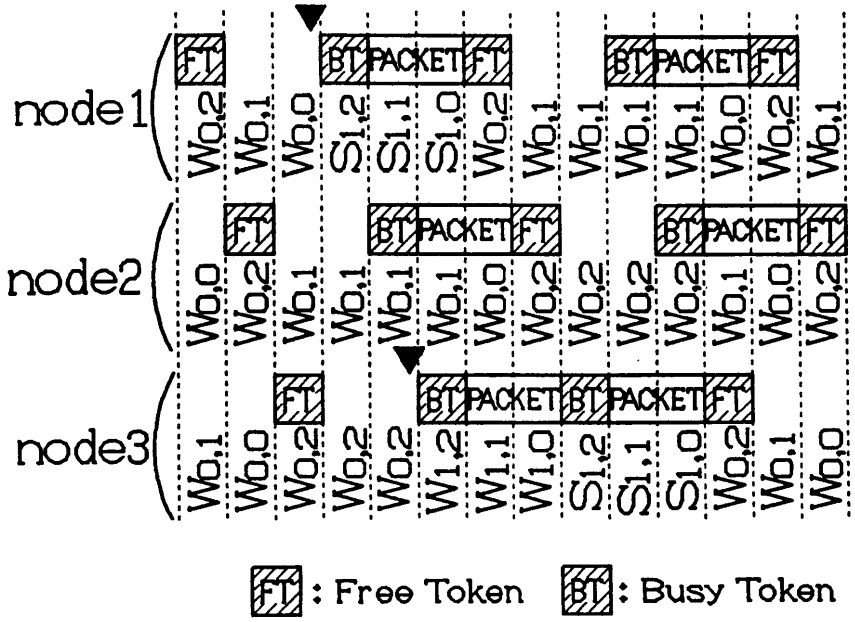


図4.4 モード遷移の例
Fig.4.4 An example of mode transitions.

4.3 トークンリングの解析

4.3.1 平衡点方程式

図4.3のモデルの解析を行う．このモデルにおいて，$W_{v,x}$モードにあるノード数を表す確率変数を$w_{v,x}$，$S_{y,z}$モードにあるノード数を表す確率変数を$s_{y,z}$とおく．モデルの定義より，明らかに次式が成り立つ．

$$\sum_{v=0}^{T} w_{v,x} = 1 \qquad (x = 1, 2, \cdots, M-1) \tag{4-2}$$

$$\sum_{v=0}^{T} w_{v,0} + \sum_{y=1}^{T} \sum_{z=0}^{H} s_{y,z} = 1 \tag{4-3}$$

モデルの状態ベクトルnを次のように定義する．

$$\begin{aligned} \mathbf{n} = (&w_{0,0}, w_{0,1}, \cdots, w_{0,M-1}, w_{1,0}, w_{1,1}, \cdots, w_{1,M-1}, \\ &\cdots \quad , w_{T,0}, w_{T,1}, \cdots, w_{T,M-1}, \\ &s_{1,0}, s_{1,1}, \cdots, s_{1,H}, s_{2,0}, s_{2,1}, \cdots, s_{2,H}, \\ &\cdots \quad , s_{T,0}, s_{T,1}, \cdots, s_{T,H}) \end{aligned}$$

このベクトルnは，既約な有限状態マルコフ連鎖となる．

このマルコフ連鎖を，平衡点解析の手法を用いて解析する．平衡点解析では，システムは常に平衡点に留っていると仮定する[TASA86]．そこで，各モードへ流入する平均ノード数と各モードから流出する平均ノード数とを等しく置くと，以下の式(4–4)〜(4–12)を得る．

$$(1-A(\mathbf{n}))w_{v,1} = w_{v,0} \qquad (v = 0,1,\cdots,T) \tag{4-4}$$

$$(1-A(\mathbf{n}))w_{v,j+1} = (1-A(\mathbf{n}))w_{v,j}$$
$$(v = 0,1,\cdots,T)(j = 1,2,\cdots,M-2) \tag{4-5}$$

$$(1-M\lambda)w_{0,0} + s_{1,0} = Bw_{0,M-1} \tag{4-6}$$

$$(M-1)A(\mathbf{n})\lambda w_{i-1,M-1} + s_{i+1,0} = Bw_{i,M-1}$$
$$(i = 1,2,\cdots,T-1) \tag{4-7}$$

$$(M-1)A(\mathbf{n})\lambda w_{T-1,M-1} = (1-A(\mathbf{n}))w_{T,M-1} \tag{4-8}$$

$$s_{y,j+1} = s_{y,j} \qquad (y = 1,2,\cdots,T)(j = 0,1,\cdots,H-1) \tag{4-9}$$

$$M\lambda w_{0,0} + Cw_{1,0} = s_{1,H} \tag{4-10}$$

$$(1-C)w_{i-1,0} + Cw_{i,0} = s_{i,H} \qquad (i = 2,3,\cdots,T-1) \tag{4-11}$$

$$(1-C)w_{T-1,0} + w_{T,0} = s_{T,H} \tag{4-12}$$

但し，BとCは，次式で定義される．

$$B = 1 - A(\mathbf{n}) + (M-1)A(\mathbf{n})\lambda \tag{4-13}$$

$$C = 1 - (M+H+1)\lambda \tag{4-14}$$

ここで，式(4–4)は$W_{v,0}$に関する式であり，式(4–5)は$W_{v,j}$ $(j = 1,2,\cdots,M-2)$，式(4–6)は$W_{0,M-1}$，式(4–7)は$W_{i,M-1}$ $(i = 1,2,\cdots,T-1)$，式(4–8)は$W_{T,M-1}$，式(4–9)は$S_{y,j}$ $(j = 0,1,\cdots,H-1)$，式(4–10)は$S_{1,H}$，式(4–11)は$S_{i,H}$ $(i = 2,3,\cdots,T-1)$，式(4–12)は$S_{T,H}$に関する式である．

次に，A(n)に関する式を得るために，ノード間を転送されるフリートークンの数について考える．あるノードがフリートークンを出力するのは，そのノードが

$W_{0,0}$モードにあって確率$1-M\lambda$で$W_{0,M-1}$モードへ遷移するか，そのノードが$S_{y,0}$モードにあって確率1で$W_{y-1,M-1}$モードへ遷移するときである．トークン時間当りのその平均数は，

$$(1-M\lambda)w_{0,0}+\sum_{y=1}^{T}s_{y,0}$$

となる．このフリートークンは次のノードが受信するが，そのとき次のノードは$W_{v,1}$モードから$W_{v,0}$モードへ遷移する．トークン時間当りのその平均数は，

$$\sum_{v=0}^{T}w_{v,1}(1-A(\mathbf{n}))=1-A(\mathbf{n})$$

となる．ここで，式の変形に式(4-2)を用いた．この2つの平均数は等しくなることから，次式を得る．

$$A(\mathbf{n})=1-(1-M\lambda)w_{0,0}-\sum_{y=1}^{T}s_{y,0} \qquad (4-15)$$

式(4-2)〜(4-15)を解き，平衡点におけるベクトル$\mathbf{n}$(これを$\mathbf{n}_e$と書く)の各要素を求める．式(4-4)〜(4-12)より，次の式(4-16)〜(4-21)が得られる．

$$w_{i,0}=X^i w_{0,0} \qquad (i=0,1,\cdots,T-1) \qquad (4-16)$$

$$w_{T,0}=CX^T w_{0,0} \qquad (4-17)$$

$$w_{i,j}=\frac{1}{1-A(\mathbf{n})}X^i w_{0,0} \qquad (i=0,1,\cdots,T-1)$$
$$(j=1,2,\cdots,M-1) \qquad (4-18)$$

$$w_{T,j}=\frac{C}{1-A(\mathbf{n})}X^T w_{0,0} \qquad (j=1,2,\cdots,M-1) \qquad (4-19)$$

$$s_{1,z}=\left\{\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}+M\lambda\right\}w_{0,0} \quad (z=0,1,\cdots,H) \qquad (4-20)$$

$$s_{i,z}=\left[\left\{1+\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}\right\}X^{i-1}-\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}X^{i-2}\right]w_{0,0}$$
$$(i=2,3,\cdots,T)(z=0,1,\cdots,H) \qquad (4-21)$$

但し，Xは，次式で定義される．

$$X=\frac{(M-1)A(\mathbf{n})\lambda}{C(1-A(\mathbf{n}))} \qquad (4-22)$$

そして，式(4–15)に式(4–20),(4–21)を代入して$w_{0,0}$について解いた式と，式(4–3)に式(4–16),(4–17),(4–20),(4–21)を代入して$w_{0,0}$について解いた式とから，$A(\mathbf{n})$に関する次の式が得られる．

$$\begin{aligned}&1+\frac{X-X^T}{1-X}+\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}X^{T-1}\\ =&(1-A(\mathbf{n}))\Big[\frac{1-X^T}{1-X}+CX^T+(H+1)\Big\{\frac{X-X^T}{1-X}\\ &+M\lambda+\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}X^{T-1}\Big\}\Big]\end{aligned}\qquad(4-23)$$

与えられた$\lambda\ (=\sigma L_t M/T_a)$，$M$，$H\ (=T_a/L_t)$，$C\ (=1-(M+H+1)\lambda)$から，この式(4–23)を満たす$A(\mathbf{n})$すなわち$A(\mathbf{n}_e)$を求め$(0\le A(\mathbf{n}_e)\le 1)$，この$A(\mathbf{n}_e)$と式(4–15)〜(4–21)より$\mathbf{n}_e$の要素が全て求まる．

4.3.2 スループット

リング型のネットワークでは，M台のノード間にM個の伝送路がある．ここで考えるスループット$\bar{S}$とは，パケット伝送が行われる割合を各伝送路ごとに求め，この平均をとったものとする．ノードが$S_{y,j}\ (j=1,2,\cdots,H)$ モードにあるとき，遷移の際パケットを送信する．このため，このモデルの条件付スループット$S(\mathbf{n})$は，

$$S(\mathbf{n})=\sum_{y=1}^{T}\sum_{j=1}^{H}s_{y,j}\qquad(4-24)$$

となる．$\bar{S}$は$S(\mathbf{n})$の$\mathbf{n}$に関する期待値である．平衡点解析では$\bar{S}$は$S(\mathbf{n}_e)$ で近似するため，式(4–20),(4–21)を用いて次のスループットの式が得られる．

$$\bar{S}=H\Big[\frac{X-X^T}{1-X}+M\lambda+\frac{(M-1)A(\mathbf{n})\lambda}{1-A(\mathbf{n})}X^{T-1}\Big]w_{0,0}\qquad(4-25)$$

4.3.3 平均パケット遅延

平均パケット遅延Dを，パケットが生成されたときから，宛先ノードにこのパケットが全て受信されるまでの時間の平均と定義する．平均パケット遅延を求めるに際しては，この遅延を次の3つの要素に分けて考え，各要素ごとに求める．

(D1) パケット発生から送信が開始されるまでの，端末バッファでの待ち時間D_1.

(D2) 送信開始後，このパケットが宛先のノードまでの途中にあるノードを転送される時間D_2.

(D3) 宛先のノードにてパケットを受信する時間D_3.

このうちD_3は，Hトークン時間と一定である．

端末バッファでの待ち時間D_1は，モデルの解析結果を用いて，以下のようにリトルの公式により求める．

システムの端末バッファ内にある全パケット数の平均値N_1を，平衡点における値で近似すると，次のようになる．

$$N_1 = \sum_{i=1}^{T}\sum_{x=0}^{M-1} iw_{i,x} + \sum_{y=1}^{T}\sum_{z=0}^{H} ys_{y,z} \qquad (4-26)$$

端末バッファ内のパケットは，ノードが$S_{y,H}$モードから$S_{y,H-1}$モードへ遷移する際に1個出力を開始する．このため，端末バッファから各トークン時間ごとに出力されるパケット数の平均値Q_1は次式で近似される．

$$Q_1 = \sum_{y=1}^{T} s_{y,H} \qquad (4-27)$$

リトルの公式より，N_1/Q_1で得られる値は，パケットが発生してから，そのパケットの送信を終了するまでの時間である．そこでパケットの送信を開始してから，終了するまでに要するHトークン時間を引いて，D_1は次の式で与えられる(単位:トークン時間).

$$D_1 = \frac{N_1}{Q_1} - H + \frac{M}{2} \qquad (4-28)$$

この式で，$M/2$トークン時間を加算している．これは，xパラメータが1減る遷移ごとのパケット発生を，$x=0$の時にまとめて確率$M\lambda$で起こると近似しているための補正である．ここでの補正は，負荷が小さく，リトルの公式で得られる値が小さいときに有効である．負荷が大きくて得られる値が大きくなると，補正が結

果の精度に与える影響はわずかになる．パケット発生の近似は，これ以外にも行っている．しかし，他の近似は負荷が小さいとき時(つまり，あまりパケットが発生しないため$A(\mathrm{n})=1$にならず，Sモードへも遷移しない時)には，補正量が少ない．このため，ここでは他の近似に対する補正は行わない．

次に，D_2を求める．まず，送信が開始されたパケットが，宛先のノードで受信されるまでに，途中中継されるノード数の期待値Gを求める．パケットの宛先が仮定(A6)の一様分布で表されることから，Gは次のようになる．

$$G=\sum_{k=0}^{M-2} k\times\frac{1}{M-1}=\frac{M-2}{2} \qquad (4-29)$$

1台のノードにより中継される際に要する時間は1トークン時間である．このため，D_2は次式で表される(単位:トークン時間)．

$$D_2=\frac{M-2}{2} \qquad (4-30)$$

以上より，平均パケット遅延Dは，次式で与えられる(単位:トークン時間)．

$$D=D_1+D_2+H \qquad (4-31)$$

4.3.4 端末バッファオーバフロー確率

パケットが端末で発生したときに，端末バッファが一杯でこのパケットが破棄される確率を，端末バッファオーバフロー確率P_{ov}と定義する．

モデルにおいては，このオーバフローはM台のノードが各々$W_{T,x}$モードもしくは$S_{T,z}$モードにあるときに起きる．このため，モデルから次のP_{ov}を求める式が得られる．

$$P_{ov}=\frac{\sum_{x=0}^{M-1} w_{T,x}+\sum_{z=0}^{H} s_{T,z}}{M} \qquad (4-32)$$

4.4 数値例

4.3で得られた解析結果の精度を調べるため，解析値とシミュレーションによる結果とを比べる．

数値例は，次の条件下で求める．パケット長は256ビット（内ヘッダ長は16ビット）とする．トークン長は，フリートークン，ビジートークンともに8ビットとする（$H = 256/8 = 32$）．各ノード内の端末バッファの段数Tは10段とし，通信速度は100Mbpsとする．

シミュレーションは仮定(A1)〜(A6)を用い，他の仮定は用いていない．シミュレーション時間は，スループットと平均パケット遅延を求める場合は$330,000M$ステップとする．端末バッファオーバフロー確率を求める場合は，確率が小さい場合も求められる様に，より長い120,000,000 ステップとする．更に，シングルラン法[KOBA78]により，95%の信頼区間を求める．しかし，以後示す各図においてはいずれも，シミュレーション値を示す記号(○△□▽)の中にこの95%の信頼区間の範囲が含まれてしまうため，特にグラフには記入していない．

また，トークンリングで端末バッファの段数が1である場合は，厳密な解析が可能である(文献[TAKA90]等)．そこで，付録B において，厳密解析，EPAによる近似解析，シミュレーションのそれぞれの結果を比較し，EPAおよびシミュレーションの結果の精度について検討する．

4.4.1 スループットと平均パケット遅延

端末バッファの段数$T = 10$の場合について，$M\sigma$とスループットとの関係を図4.5に，$M\sigma$と平均パケット遅延(T_aで正規化)との関係を図4.6に示す．図中実線が解析によって得られた値を示し，▽△○がそれぞれ $M = 10,30,100$のシミュレーション結果を示す．解析値とシミュレーション結果とは良く一致している．

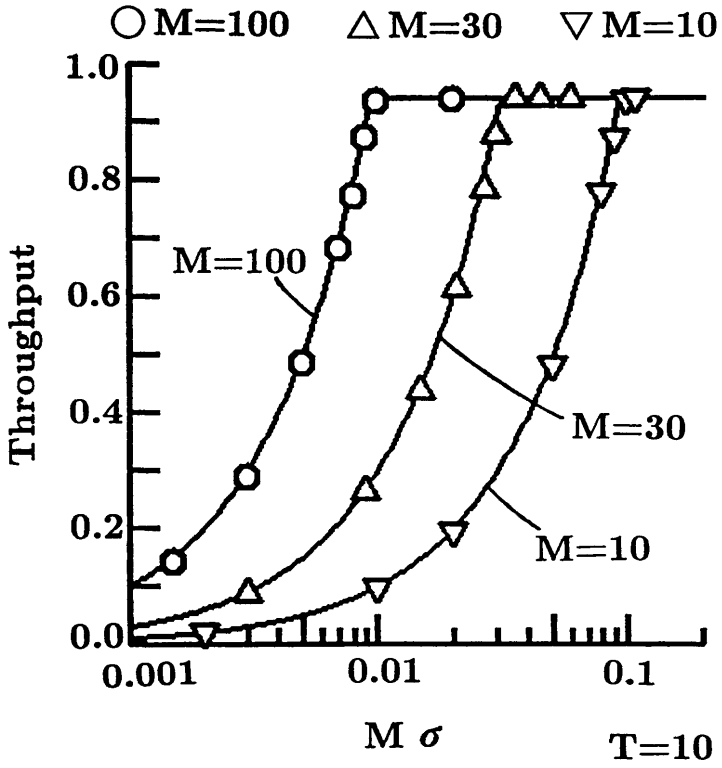


図4.5 スループットと$M\sigma$の関係($T = 10$)
Fig.4.5 Throughput versus $M\sigma$ for $T = 10$.

図4.5から，スループットは負荷を大きくしても0.97以上にはならないことが分る．これは，常に各伝送路がパケットを連続して送り続ける場合でも，各パケットの間にビジートークンが入り，全体としてパケットの伝送が占める割合が256/(256＋8)となることから，説明できる．

4.4.2 端末バッファオーバフロー確率

図4.7に，$M = 30$のときの$M\sigma$と端末バッファのオーバフロー確率の関係を，端末バッファの段数Tの値が2,5,10の場合について示す．図中実線が解析によって得られた値を示し，記号○△▽はそれぞれ$T = 2,5,10$の場合のシミュレーション結果を示す．解析値とシミュレーション結果とは，良く一致している．

オーバフローしたパケットは破棄されるため，オーバフロー確率とパケットの破棄率とは一致する．このパケット破棄率をデータや映像のパケットに求められる1.0×10^{-9} 以下[VERB87]とするためには，$T = 2,5,10$の場合それぞれσの値を0.000008, 0.0005, 0.0009以下とする必要があることがわかる．

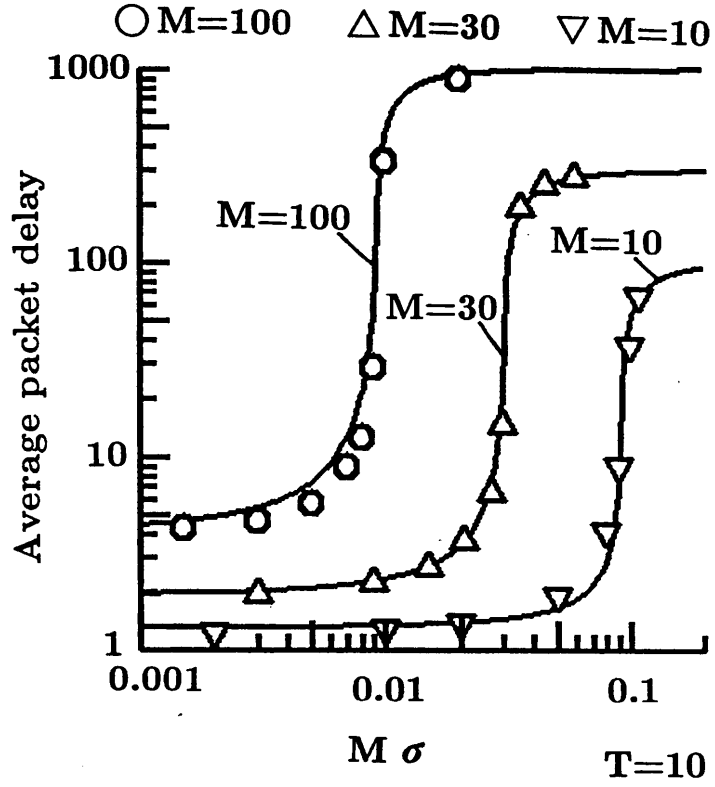


図4.6 平均パケット遅延と$M\sigma$の関係($T=10$)
Fig.4.6 Average packet delay versus $M\sigma$ for $T=10$.

付録B 厳密解析との比較

文献[TAKA90]で示されたポーリングシステムにおける性能解析の結果から，端末バッファの段数Tが1である場合の厳密な解析値が得られる．文献[TAKA90]では，1ポーリングサイクル当りの平均送信メッセージ数$E[Q]$ は，次式で与えられる．

$$E[Q]=\frac{M\sum_{n=0}^{M-1}\binom{M-1}{n}\Pi_{j=0}^{n}\{e^{\lambda(Mr+jb)}-1\}}{1+\sum_{n=1}^{M}\binom{M}{n}\Pi_{j=0}^{n-1}\{e^{\lambda(Mr+jb)}-1\}} \qquad (B-1)$$

ここで，rはポーリングをあるノードから次のノードに移すのに要する時間であり，bは送信メッセージを持つノードにポーリングを行ってから，次のノードにポーリングが移るまでの時間の平均である．トークンリングでは，これらの時間は次式で与えられる．

$$r=L_t \qquad (B-2)$$

$$b=L_t+T_a \qquad (B-3)$$

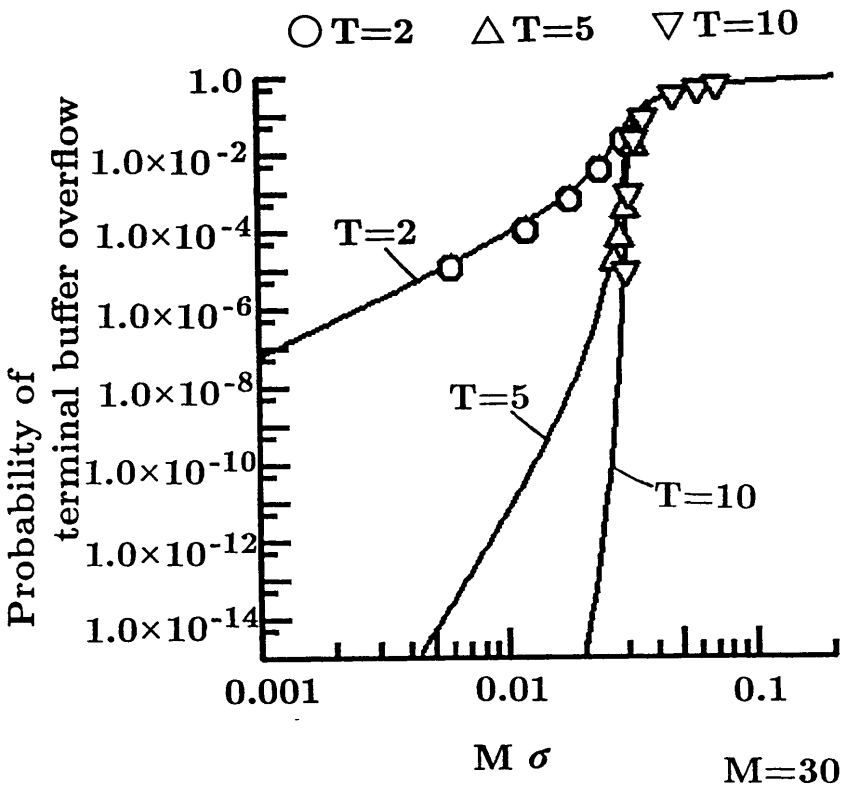


図4.7 端末バッファオーバフロー確率と$M\sigma$の関係($M = 30$)
Fig.4.7 Probability of terminal buffer overflow versus $M\sigma$ for $M = 30$.

また，λは各ノードにおけるメッセージ発生のポアソン分布のパラメータである．時間tの間にk個メッセージが発生する確率$P_k(t)$は$P_k(t) = \frac{(\lambda t)^k}{k!}e^{-\lambda t}$で表されるため，1ステップ時間$(T_a/M)$に1個パケットが発生する確率$\sigma$と$\lambda$の値の関係は，次式により与えられる．

$$\sigma = \lambda \frac{T_a}{M} e^{-\lambda T_a/M} \qquad (B-4)$$

この$E[Q]$を用い，次式によりメッセージのバッファ内平均待ち時間$E[W]$を得る．

$$E[W] = (M-1)b - \frac{1}{\lambda} + \frac{MR}{E[Q]} \qquad (B-5)$$

この$E[W]$は，本論文におけるD_1にあたる．D_2,D_3の値は常に一定であり，本論文の値を用いる．単位時間における平均送信パケット数γは，

$$\gamma = \frac{M}{E[W] + b + 1/\lambda} \qquad (B-6)$$

で与えられ，この値にパケット長を乗じると本論文の定義によるスループットが得られる．

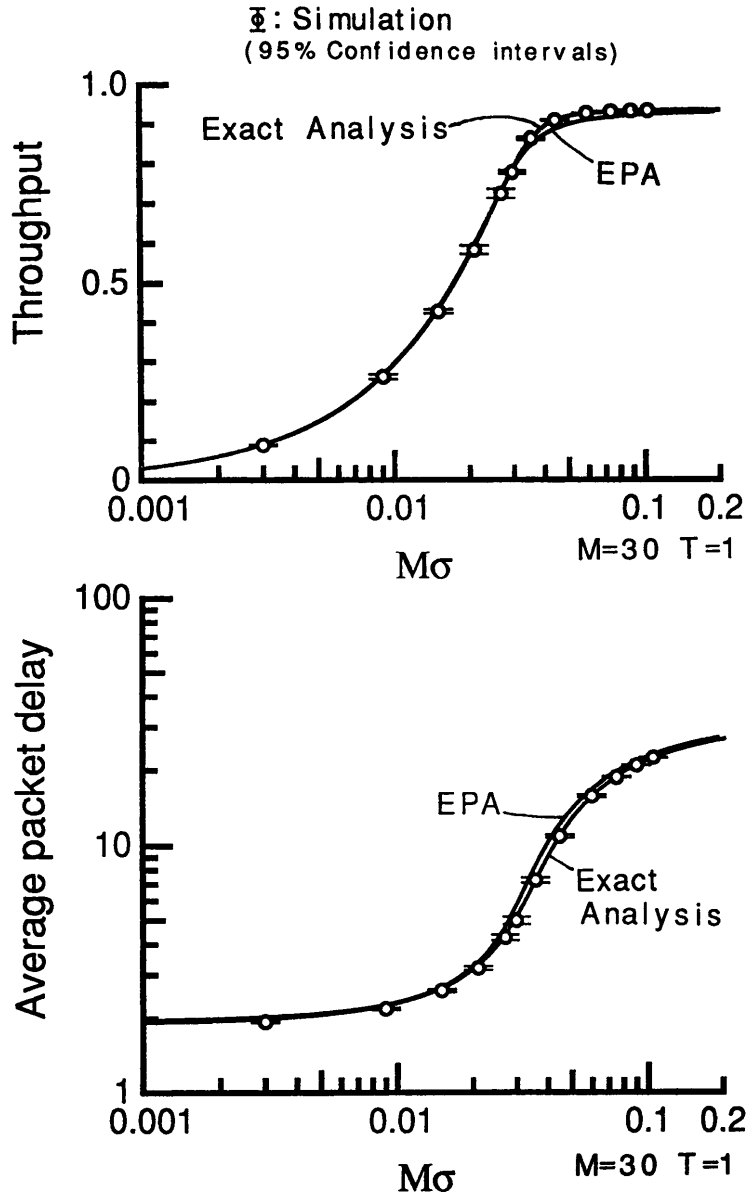


図4.8 EPA，厳密解析，シミュレーションによる結果の比較($M = 30, T = 1$)
Fig.4.8 Comparison of EPA, exact analysis and simulation results for $M = 30$ and $T = 1$.

以上の式を用いた厳密な解析とトークンリングのEPAによる解析，さらにシミュレーションによるそれぞれの結果を，数値例を示して比較する(図4.8)．ここでは，ノードの台数$M = 30$，端末バッファの段数$T = 1$とし，他のパラメータは，4.4と同じである．この結果より，シミュレーションと厳密解析の結果は一致し，シミュレーションプログラムの正当性が示された．また，EPAによる近似解析と厳密解析の結果もほぼ一致し，EPAによる解析結果の精度が良いことが示された．

第5章 スロットリングの性能解析

5.1 スロットリング

解析するスロットリングについて述べる．

図5.1にスロットリングの構成例を示す．リング上には，1台のタイミングの調整を行うノード(以後このノードをマスタノードと呼ぶ)と，他の複数のノードがある．図5.1では，ノード1(node1)がマスタノードである．リング上には，固定長のスロットが循環している．スロットは，先頭の1ビットのフラグと，それに続くデータ部から成る．フラグは，続くデータ部にパケットが書き込まれている(Full)か，空(Empty)なのかを示す．以後，フラグがFullのスロットをフルスロット，フラグがEmptyのスロットをエンプティスロットと呼ぶ．

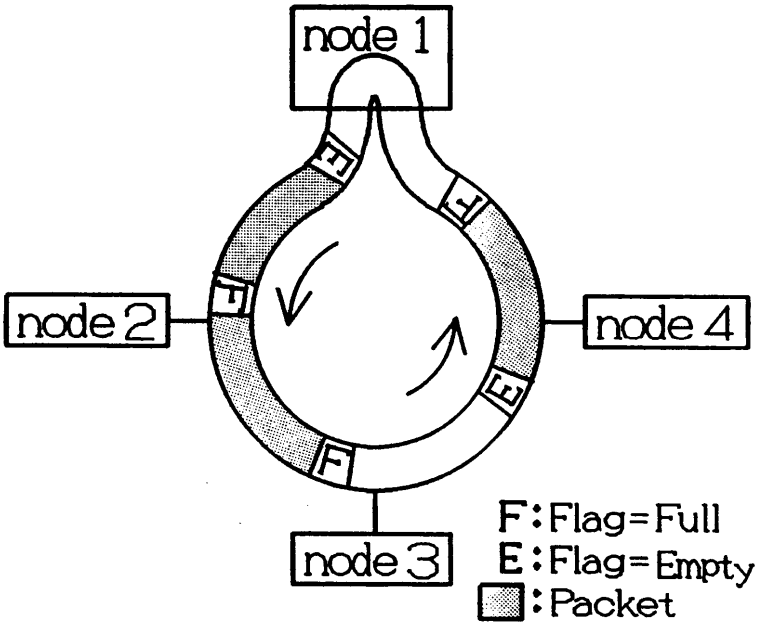


図5.1 スロットリング
Fig.5.1 Slotted ring.

各ノードに接続した端末から発生するパケットは，ノード内の端末バッファと呼ぶバッファに蓄えられる．端末バッファにパケットを持つノードは，エンプティスロットの受信を待ち，受信するとフラグをFullに書き換え，データ部にパケットを書き込んで送信する．

各ノードは，フルスロットを受信すると，そのデータ部のパケットを複製して内部に取り込みながら，スロットを次ノードへ転送する．このパケットの宛先が自ノードの時は，ノードに接続した端末にこのパケットを送り，宛先が自ノードではない時は，このパケットを破棄する．

エンプティスロットを受信したノードが，送信すべきパケットを持たないときは，このスロットをそのまま次ノードへ転送する．

フルスロットに書き込まれたパケットは，このスロットがリングを一周してきた時に，送信元のノードが取り除く．各ノードは，リング上を循環しているスロットの総数をあらかじめ知っており，自分がパケットを書き込んだスロットがリングを一周し受信した時，フラグを受信した時点でそのことがわかる．そこで，この時端末バッファ内にパケットが無ければ，このフラグをFullからEmptyに戻し，直ちにこのスロットを次ノードへ転送する．また，端末バッファ内にパケットがあれば，フラグはFullのままで転送し，続いてバッファ内パケットをデータ部に書き込んで転送する．

リング上を循環するスロットの総数は，各ノードにおいてスロットを受信してから次ノードへ送信するまでの転送遅延，ノード数，ノード間の伝搬遅延から定まるリングの収容ビット数を，スロット長で割った値となる．このとき，割り切れないときは，マスタノード内に遅延バッファを設けることでリングの収容ビット数を増やし，ちょうど割り切れるようにする．つまり，この場合はマスタノードは他のノードよりスロットを転送する際のノード内遅延を大きくして，収容しきれないスロットの一部を内部に蓄えることとなる．

例えば，ノードの転送遅延を1ビット送信時間，ノード数を100台，スロット長を30ビット，ノード間の伝搬遅延を無視できるとする．このときは，マスタノードの遅延バッファを21ビットとしてリングの収容ビット数を120ビットにし，スロットの総数を4とする．

マスタノードは，このリングの収容ビット数を調整することと，初期時にエンプティスロットの送信を行い全ノードの送信タイミングの同期をとる事を除いては，パケットの送受信等の処理は他のノードと同じである．又，他のノード内のスロット転送の遅延は，フラグの書き換えを行う時間を保証する1ビット送信時間だけでよい．

各ノードの送信タイミングの例を図5.2に示す．この図は，各ノードごとにそのノードから送信されるスロットの状態を表したものである．図では，ノード1からノード4宛のパケットAと，ノード4からノード3宛のパケットBとが送信されている．パケットは，宛先ノードに到着後，さらにリングを一周して，発信元のノードにより取り除かれる．ここで，時間T_aは1スロットの送信時間であり，T_hが1ビット送信に要する時間である．各ノードは，スロット受信開始後T_h時間後にスロットの送信を開始できるように，タイミングが決められる．ただし，マスタノードだけはT_h時間以上待つ場合がある．このため，パケットがスロットに書き込まれて送信を開始してから宛先ノードに受信され始める迄の時間は，このマスタノードであるノード1を通るか通らないかで異なったものとなる．例えば，図5.2においてパケットAとBは，共に2台のノードにより中継されて宛先ノードに達する．しかし，パケットAが送信を開始して $2 \times T_h$時間後に宛先ノードに受信され始めるのに対して，パケットBは$T_h + T_a - 3 \times T_h$ 時間後に宛先ノードに受信され始める．

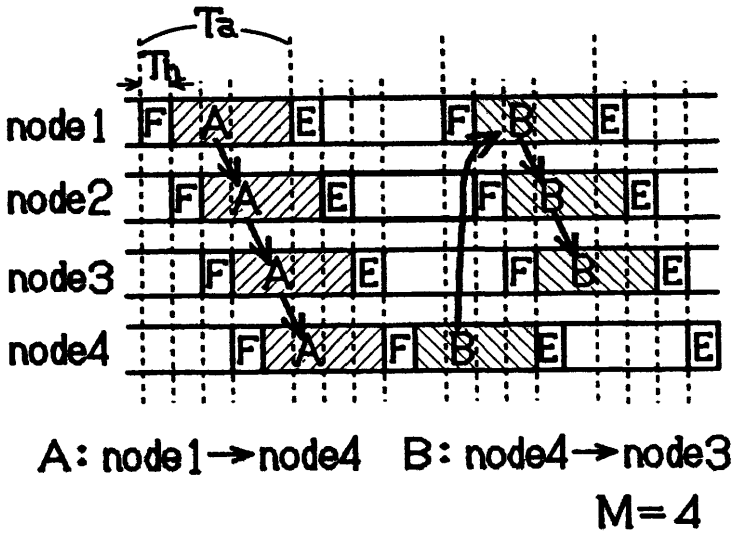


図5.2 パケット送信タイミング例
Fig.5.2 Example of timing chart of sending packets.

なお，以下の説明において，記号aを次の式を満たす整数として定義する(aはビットを単位とするスロット長である)．

$$a = \frac{T_a}{T_h} \tag{5-1}$$

リング上のスロット数は，このaとMとから，$\lceil M/a \rceil$で求められる（$\lceil A \rceil$は，Aの小数点以下を切り上げた整数）．

5.2 モデル化

5.1で説明したスロットリングのネットワークを平衡点解析の手法で解析するため，ネットワークにおけるM台のノードの各々がとりうる状態(モード)の遷移の関係を表現したモデルを作成する．

先ずモデル化を，$M \leq a$でリング上のスロット数が1の場合と$M > a$でリング上のスロット数が複数である場合に分けて行い，その後この2つのモデルは統合して考えることが出来ることを示す．

5.2.1 $M \leq a$の場合のモデル化

[1] モデル化の考え方

モデル作成の基本的な考え方を説明する．

図5.3(a) に示すように，マスタノードであるノード1だけは，スロット受信を開始してからスロット送信を開始するまでの遅れが$T_a - T_h \times (M-1)$となり，他のノードはこの遅れがT_hである．この遅れの不均一をそのままモデル化すると，モデルはモード数が多く複雑なものとなる．

また，巡回しているフルスロットがリングを一周して元のノードが受信したときに，スロット中のパケットを取り除くことをそのままモデルに表そうとすると，分布に記憶が生じ，モデルがより複雑なものとなる．

さらに，各ノードでの端末からのパケット発生についても，任意のステップでの発生を表すと，モデルはモード間の遷移が多くなり，解析が困難になる．

そこで，モデルを簡単にする以下の作業を行う．

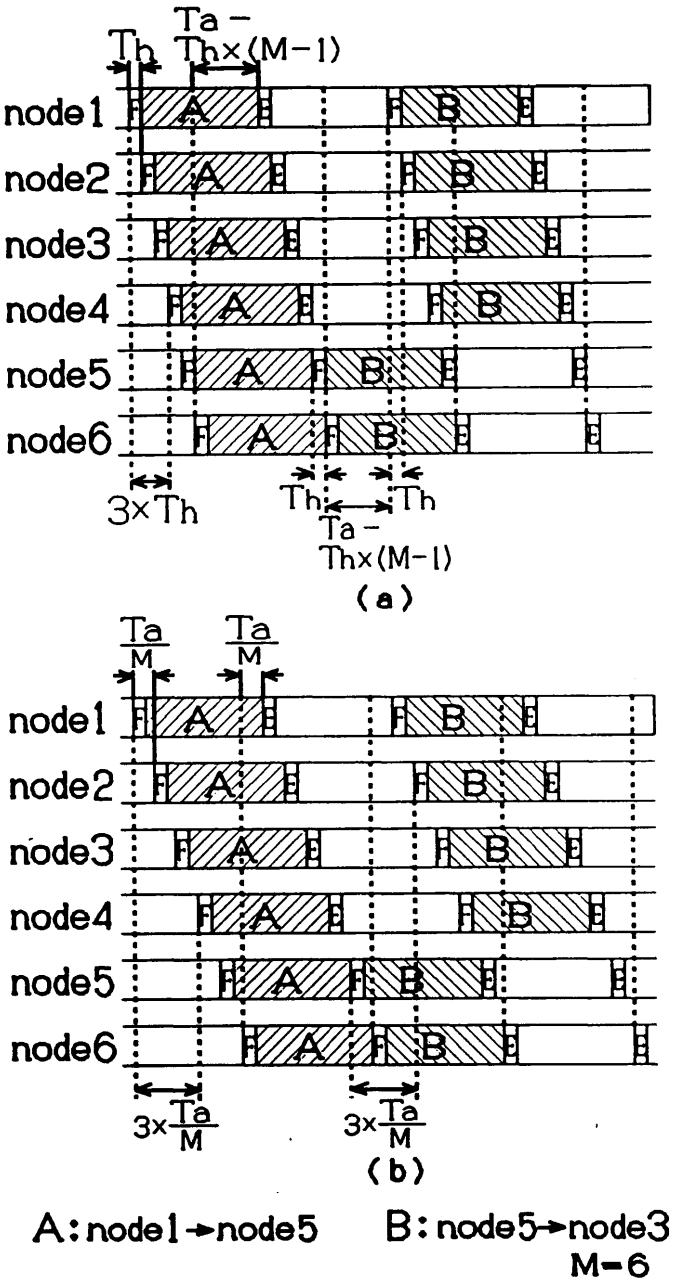


図5.3 パケット送信タイミングの変形($M \leq a$)
Fig.5.3 Modification of timing of sending packets in the case of $M \leq a$.

(1)フルスロット中のパケットの除去

受信したフルスロット中のパケットを除去するノードの分布を幾何分布とする次の仮定(SL1)を導入し，モデルを簡単にする．

(SL1) 隣接ノードからフルスロットを受信したノードは，確率$E = 1/M$でこのフルスロットを作ったのは自ノードであるとし，パケットを取り除く．

ここで，Eの値を$1/M$としたことは，次の理由による．パケットが取り除かれるまでにいくつのノードを中継されるかという，中継ノード数の期待値Hは，次の式で与えられる．

$$H = \sum_{k=0}^{\infty} kE(1-E)^k = \frac{1-E}{E}$$

実際は，この中継ノード数は常に$M-1$であるので，Hが$M-1$となるようEの値を次のようにする．

$$E = \frac{1}{M} \tag{5-2}$$

(2)送信タイミングの変形

各ノード間のスロット送信開始の遅れが全て均一になるように，モデルにおいては，遅れをT_a/Mに揃えた送信タイミングに変形する(図5.3(b))．この遅れT_a/Mは，2.で定義した1ステップに相当する．このタイミングの変形により，全ノードは1ステップごとに接続順にスロット送信を開始するようになりモデル化が容易になる．

(3)パケット発生タイミングの変形

仮定(A5)より，各ノードに接続した端末は，任意のステップで新たなパケットを発生するが，これをそのままモデル化すると，モード間の遷移が多くなり，解析が困難になる．そこで，このパケット発生タイミングについて，モデルでは仮定(A5)の代りに，次の仮定(SL2)を導入する．

(SL2) 各端末は，接続したノードが新たなスロットの送信を開始する直前に，確率$M\sigma$でパケットを生成し，端末バッファに蓄える．

すなわち，Mステップ=1スロットの間にたかだか1個のパケットが発生する．1スロットの間に2個以上のパケットが発生する確率については，通常$\sigma \ll 1$であるため無視する．

なお，これまでに示した各変形は，あくまでもモデルの簡単化のために行うものであり，実際のシステムを変形するものではない．また，ノードから送信されるスロット内のパケットについて見ると，実際のシステムでは送信中はこのパケットの一部は次のノードにあり，残りは送信しているノードにある．しかし，この一時的に両方のノードにまたがっている状態は，モデル化を困難にするため，モデルでは送信を開始すると直ちにこのパケットは次のノードに移ると考えて，ノードの状態を変化させることにする．

(4)パケット転送時間の補正

ここで，一つのパケットがその発生元のノードから送信され始めた時点から，宛先ノードで受信を開始するまでの時間を，パケット転送時間と呼ぶことにする．

(2)の変形は，このパケット転送時間に関して，モデルより求めた値と実際のシステムにおける値とで，くい違いを生じさせる．これを図5.3を用いて説明する．図のパケットA，Bは，共に途中3台のノードにより転送され宛先ノードへ到着するパケットである(パケットAはノード1からノード5宛，パケットBはノード5からノード3宛)．ただし，パケットBはマスタノードであるノード1により中継されるが，パケットAはノード1は中継しない．このとき，パケットAの転送時間は$3 \times T_h$となるが，パケットBの転送時間は$2 \times T_h + T_a - T_h \times (M-1)$となる．一方，タイミングを変形したモデルでは，転送時間はパケットA,B共$3 \times (T_a/M)$ となる．

このため，モデルから得られた転送時間D_mを，次の式により実際のシステムにおける転送時間D_2に変換する必要がある．この式については，RISリングの解析について述べる**第3章の付録A**を参照されたい．

$$D_2 = \frac{M}{a} D_m + \frac{(M-2)(a-M)}{2M} T_h \qquad (5-3)$$

[2] モデル化

作成したモデルを図5.4に示す．各ノードの状態は，次ノードへ転送すべきフルスロットを受信したかどうかを示すパラメータx($x = 1$のとき転送すべきフルスロットを受信したことを示し，それ以外の時は$x = 0$)と，端末バッファ内のパケット数$y(y = 0, 1, \cdots, T)$，更にzステップ後($z = 0, 1, \cdots, M-1$)に新たにスロット送信を開始するという，合計3個のパラメータで表すことが出来る．各モードは，この3つのパラメータを用い，$W_{x,y,z}$と書く．

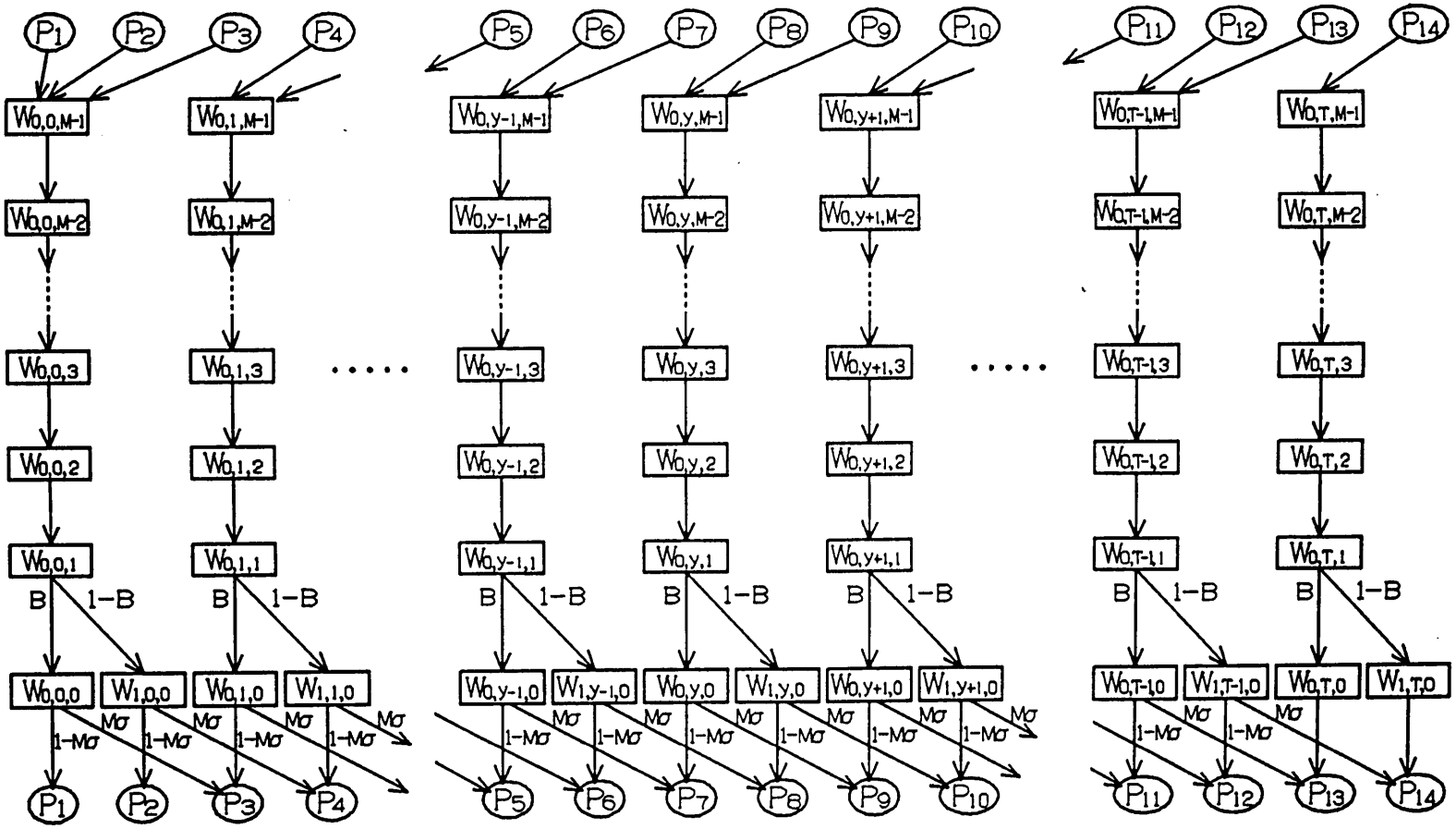


図5.4 スロットリングの近似モデル
Fig.5.4 An approximate model of a slotted ring.

このモデルの状態の観察，パケットの発生とスロットの送信，モードの遷移のタイミングは，次のようになる．

(T1) 状態の観察は各ステップの最初に行う．

(T2) モードの遷移は各ステップの最後に行う．

(T3) 送信を開始するステップにあるノードは，ステップの終了時に送信を行う(但し(T2)の直前)．

(T4) 端末からのパケット発生はそのノードが送信を開始するステップの終了時に起きる(但し(T3)の直前)．

zパラメータは，このステップの最後にスロットの送信を開始するノードが$z=0$であり，以後ノードのzの値は接続順に1から$M-1$までの値となる．iステップ後に送信を開始するノードは，端末バッファ内のパケット数yに応じて図中水平に並んだ$W_{0,y,i}$のモードのいずれか一つにあることになる．そして，1ステップごとにzの値は1ずつ減り，$z=0$のノードはステップの最後にスロット送信を開始して，$z=M-1$になる．

$z=0$のノードのスロット送信には，3つの場合がある．

第1は，転送すべきフルスロットが無く，端末バッファにパケットがない場合で，エンプティスロットを送信する．転送すべきフルスロットが無いというのは，エンプティスロットを受信したか，フルスロットを受信しても確率Eでこのスロット中のパケットを取り除いた場合である．この時の遷移は，ノードが$W_{0,0,0}$にある場合確率$1-M\sigma$で$W_{0,0,M-1}$へ遷移する．

第2は，転送すべきフルスロットが無く，端末バッファにパケットがある場合で，端末バッファ内のパケットをデータ部に書いたフルスロットの送信を開始する．この場合の遷移は，ノードが$W_{0,y,0}$ $(y=1,2,\cdots,T-1)$にある場合は確率$1-M\sigma$で，$W_{0,y-1,0}$ $(y=1,2,\cdots,T)$にある場合確率$M\sigma$で，送信時に端末バッファにy個パケットがあり，$W_{0,y-1,M-1}$へ遷移する．またノードが$W_{0,T,0}$にある場合は，これ以上端末バッファ内のパケット数は変化しないため，確率1で$W_{0,T-1,M-1}$へ遷移する．

第3は，転送すべきフルスロットが有る場合で，この時は端末バッファの状態に関わらず，このフルスロットの送信を開始する．この場合の遷移は，ノードが$W_{1,y,0}$ $(y=0,1,\cdots,T-1)$にある場合確率$1-M\sigma$で端末バッファの状態は変

化せずに$W_{0,y,M-1}$へ遷移し，確率$M\sigma$で端末バッファ内のパケット数が1つ増えて$W_{0,y+1,M-1}$へ遷移する．また，ノードが$W_{1,T,0}$にある場合は，これ以上端末バッファ内のパケット数は増えず，確率1で$W_{0,T,M-1}$へ遷移する．

送信されたスロットは，隣接する$z=1$のノードが受信し，$z=0$のモードへ遷移する．このとき，スロットのフラグがFullならば，仮定(SL1)より確率Eでスロット中のパケットを取り除く．また確率$1-E$でこのフルスロットは次ノードへ転送すべきスロットとし，このノードのxパラメータの値は1になる．

ここで，この$W_{0,y,1}$からの遷移確率を求めるため，確率変数$A(\mathrm{n})$を次の様に定義する(ベクトルnは状態ベクトルである．これについては，5.3.1で説明する)．

$$A(\mathrm{n})=\begin{cases}1 & \text{遷移の際，スロット送信ノードは}\\ & \text{エンプティスロットを送信}\\ 0 & \text{遷移の際，スロット送信ノードは}\\ & \text{フルスロットを送信}\end{cases} \tag{5-4}$$

$z=1$のノードが次の遷移で$x=0$であるのは，$A(\mathrm{n})=1$(エンプティスロットを受け取る)であるか$A(\mathrm{n})=0$(フルスロットを受け取る)であっても確率Eでそのスロット中のパケットを取り除く場合であり，その確率は$A(\mathrm{n})+\{1-A(\mathrm{n})\}E$となる．以後

$$B=A(\mathrm{n})+\{1-A(\mathrm{n})\}E \tag{5-5}$$

とする．これは，$W_{0,y,1}$から$W_{0,y,0}$への遷移確率である．逆に，$z=1$のノードが次の遷移で$x=1$になるのは，$A(\mathrm{n})=0$(フルスロットを受け取る)でありかつ確率$1-E$でそのスロットを次ノードへ転送すべきスロットとする場合でありその確率は$1-B$となる．これは，$W_{0,y,1}$から$W_{1,y,0}$への遷移確率である．

5.2.2 $M>a$の場合のモデル化

[1] モデル化の考え方

$M>a$の場合は，パケットは$\lceil M/a\rceil$個のスロットにより送信される．以後，

$$g=\lceil M/a\rceil \tag{5-6}$$

とおく．そこで，モデル化においてもシステムをスロットごとにg個の系に分けて各々をモデル化することとし，$M\leq a$の場合と同様にモデルの簡単化のため，次の作業を行う．

(1)フルスロット中のパケットの除去

$M \leq a$の場合と同様，モデルにおいては仮定(SL1)を用いることとする．ここで，確率Eの値は，$M \leq a$の場合と同じ，式(5–2)を用いる．

(2)送信タイミングの変形

$M \leq a$の場合と同様，各ノードでのスロット送信開始のタイミングの遅れを同間隔にするため，タイミングの変形を行う．図5.5で示す例により説明する．

実際のシステム（図中(a)）では，マスタノードであるノード1を基準として，各ノードのスロット送信開始のタイミングをT_hずつ遅らせる．しかし，ノード8とノード1との間だけは，遅れが$g \times T_a - T_h \times (M-1)$となる．

モデル化に際してはg個の系に分け，更にこの遅れが全てT_a/Mに揃うように各ノードの送信タイミングを変形する（図中(b)）．

(3)パケット発生タイミングの変形

$M \leq a$の場合と同様，パケット発生タイミングについて，仮定(A5)の代りに，仮定(SL2)を用いる．

(4)パケット転送時間の補正

$M \leq a$の場合と同じ議論により，モデルから得られた転送時間D_mを用いて，実際のシステムにおける転送時間D_2を次のように求められる．

$$D_2 = \frac{M}{a} D_m + \frac{(M-2)(g \times a - M)}{2M} T_h \qquad (5-7)$$

ここで，$M \leq a$の場合の変換式(5–3)と，この式(5–7)とを比べると，$M \leq a$の時は$g = \lceil M/a \rceil = 1$であることから，$M \leq a$の場合もこの式(5–7)で転送時間の補正が出来ることがわかる．

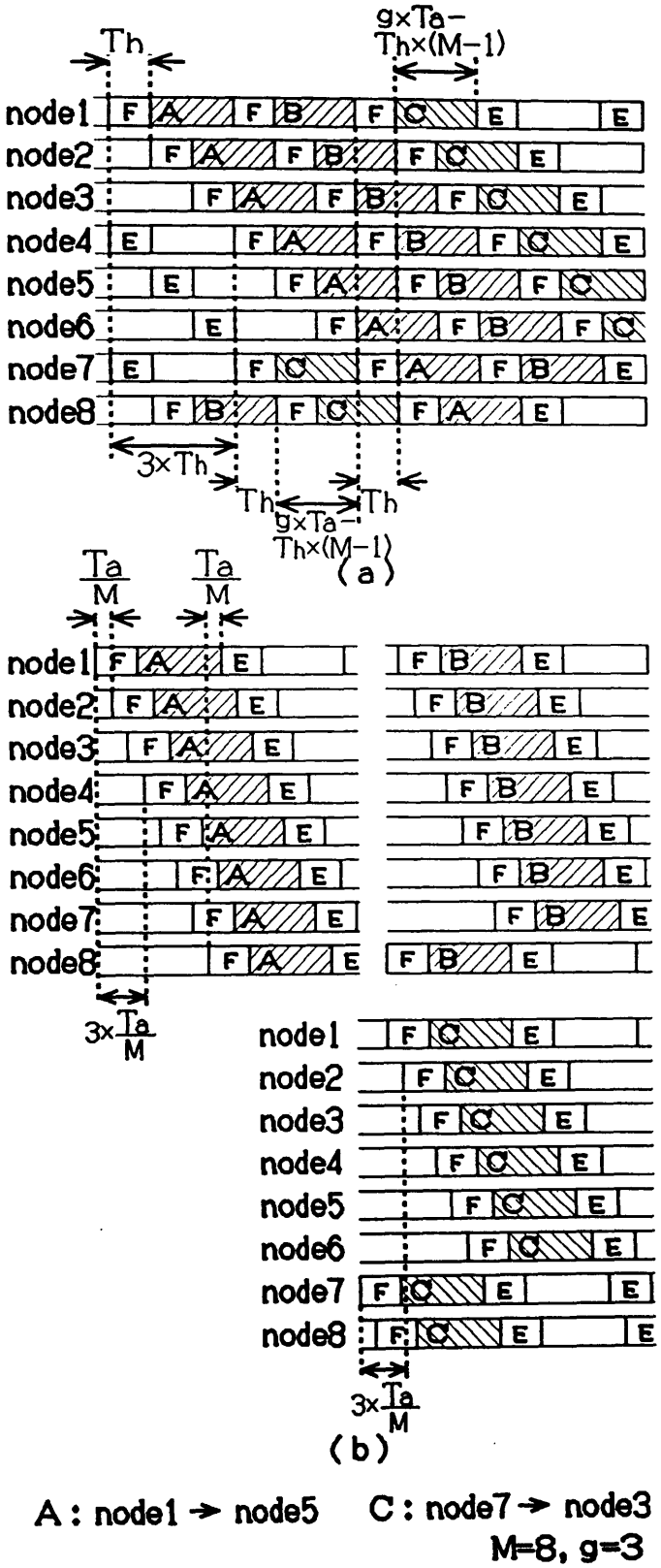


図5.5 パケット送信タイミングの変形($M > a$)

Fig.5.5 Modification of timing of sending packets in the case of $M > a$.

[2] モデル化

各系ごとのシステムは，$M \leq a$の場合と同じ図5.4に示すモデルで表すことが出来る．各系においてノード間を転送されるパケットは系ごとに独立し，系間をまたがって転送されることはない．このため，転送されるパケットについては，この系ごとのモデルで完全に記述できる．しかし，端末バッファ内のパケットについては，系ごとに独立していない．

図5.4において，$P_1P_2P_3$の経路で遷移するノードは，遷移後の端末バッファが空であるため，端末バッファ内パケットの問題はない．しかし，$P_4 \sim P_{14}$の経路で遷移するノードは，遷移後端末バッファにパケットが残る．この端末バッファ内のパケットは，次のスロット送信開始時点にxパラメータの値が0であれば送信を開始するため，この遷移は，この系の$W_{0,y,M-1}$へ遷移するのではなく，実際のシステムにおいて次に送信が開始される系の$W_{0,y,M-1}$へ遷移する．ここで，各系のモデルは同一であるため，平衡点においては系間をまたがるフローは全て同じとなる．そのため，平衡点において解析するならば，次の系への出力と前の系からの入力は等しくなり，結局自己の$P_4 \sim P_{14}$からの出力をそのまま入力として考えて$W_{0,y,M-1}$へ遷移させても，解析結果は同じとなる．このことは，$M > a$の場合でも，図5.4が示す$M \leq a$の場合のモデルによって解析が出来ることを示す．

この考察と，フルスロット中のパケットを取り除く確率Eの値が常に式(5–2)を用い，さらに転送時間の訂正が常に式(5–7)で可能であることから，スロットリングの解析は，図5.4によるモデルと式(5–2)による確率Eの計算，式(5–7)による転送時間の訂正により，全ての場合において可能であることが示された．このため，以後は$M \leq a$の場合と$M > a$の場合に分けることなく，解析を進めることにする．

5.3モデルの解析

5.3.1 平衡点方程式

作成したモデルの解析を行う．このモデルにおいて，モード$W_{x,y,z}$にあるノード数を表す確率変数を$n_{x,y,z}$とおく．モデルの定義より明らかに次の式が成り立つ．

$$\sum_{y=0}^{T} n_{0,y,0} + \sum_{y=0}^{T} n_{1,y,0} = 1 \qquad (5-8)$$

$$\sum_{y=0}^{T} n_{0,y,z} = 1 \qquad (z = 1, 2, \cdots, M-1) \qquad (5-9)$$

モデルの状態ベクトルnを次のように定義する．

$$\begin{aligned}\mathbf{n} = \{&n_{0,0,0}, n_{0,0,1}, \cdots, n_{0,0,M-1}, n_{1,0,0},\\ &n_{0,1,0}, n_{0,1,1}, \cdots, n_{0,1,M-1}, n_{1,1,0},\\ &\vdots\\ &n_{0,T,0}, n_{0,T,1}, \cdots, n_{0,T,M-1}, n_{1,T,0}\}\end{aligned}$$

このベクトルnは，既約な有限状態マルコフ連鎖となる．

このマルコフ連鎖を，平衡点解析の手法を用いて解析する．平衡点解析では，システムは常に平衡点に留っていると仮定する．平衡点は，各モードにおける平均増加ノード数が0となる点と定義される[TASA86]．そのため，状態nが与えられたとき，各モードへ流入する平均ノード数と各モードから流出する平均ノード数とが等しくなるように式をたて，平衡点を求める式を得る．

$$Bn_{0,y,1} = n_{0,y,0} \qquad (y = 0, 1, \cdots, T) \qquad (5-10)$$

$$(1-B)n_{0,y,1} = n_{1,y,0} \qquad (y = 0, 1, \cdots, T) \qquad (5-11)$$

$$\begin{aligned}&n_{0,y,z+1} = n_{0,y,z}\\ &\qquad (y = 0, 1, \cdots, T)(z = 1, 2, \cdots, M-2)\end{aligned} \qquad (5-12)$$

$$n_{0,0,0} + (1 - M\sigma)(n_{1,0,0} + n_{0,1,0}) = n_{0,0,M-1} \qquad (5-13)$$

$$\begin{aligned}&M\sigma(n_{1,y-1,0} + n_{0,y,0}) + (1 - M\sigma)(n_{1,y,0} + n_{0,y+1,0})\\ &\qquad = n_{0,y,M-1} \qquad (y = 1, 2, \cdots, T-2)\end{aligned} \qquad (5-14)$$

$$M\sigma(n_{1,T-2,0}+n_{0,T-1,0})+(1-M\sigma)n_{1,T-1,0}$$
$$+n_{0,T,0}=n_{0,T-1,M-1} \qquad (5-15)$$

$$M\sigma n_{1,T-1,0}+n_{1,T,0}=n_{0,T,M-1} \qquad (5-16)$$

次に，ノード間を転送されていくパケットの数を考える．$z=0$のモードにあるノードは，図5.4のモデルのP_1で示す遷移以外の遷移を行うときにパケットを次の$z=1$のモードにあるノードへ出力する．その平均数は，

$$1-(1-M\sigma)n_{0,0,0} \qquad (5-17)$$

となる．このパケットは確率Eで受信したノードが取り除くので，$z=1$のモードのノードが次へ転送するパケットの数は，

$$\{1-(1-M\sigma)n_{0,0,0}\}(1-E) \qquad (5-18)$$

となる．これは，$W_{0,y,1}$モードにあるノードが$W_{1,y,0}$モードへ遷移する際に行われたため，このパケット数は，

$$\sum_{y=0}^{T}(1-B)n_{0,y,1}=1-B \qquad (5-19)$$

となる．式(5–19)のパケット数は式(5–18)のパケット数と等しくなければならないので，次の式を得る．

$$\{1-(1-M\sigma)n_{0,0,0}\}(1-E)=1-B \qquad (5-20)$$

この式(5–8)〜(5–20)を解くことで，平衡点

$$\mathbf{n}_e=\{n_{0,0,0e},n_{0,0,1e},\cdots,n_{0,0,M-1e},n_{1,0,0e},$$
$$n_{0,1,0e},n_{0,1,1e},\cdots,n_{0,1,M-1e},n_{1,1,0e},$$
$$\vdots$$
$$n_{0,T,0e},n_{0,T,1e},\cdots,n_{0,T,M-1e},n_{1,T,0e}\}$$

の各要素が得られる．なお，今後の式の展開においては$n_{x,y,ze}$を単に$n_{x,y,z}$と書く．

式(5–8)〜(5–16)より，次の式が得られる．

$$n_{0,y,0}=\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{y}n_{0,0,0}$$
$$(y=0,1,\cdots,T-1) \qquad (5-21)$$

$$n_{0,T,0} = \frac{M\sigma(1-B)}{B}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{T-1} n_{0,0,0} \qquad (5-22)$$

$$n_{0,y,M-1} = n_{0,y,M-2} = \cdots = n_{0,y,2} = n_{0,y,1}$$
$$= \frac{1}{B}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{y} n_{0,0,0} \quad (y = 0,1,\cdots,T-1) \qquad (5-23)$$

$$n_{1,y,0} = \frac{1-B}{B}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{y} n_{0,0,0}$$
$$(y = 0,1,\cdots,T-1) \qquad (5-24)$$

$$n_{0,T,M-1} = n_{0,T,M-2} = \cdots = n_{0,T,2} = n_{0,T,1}$$
$$= \frac{1}{B}\,\frac{M\sigma(1-B)}{B}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{T-1} n_{0,0,0} \qquad (5-25)$$

$$n_{1,T,0} = \frac{1-B}{B}\,\frac{M\sigma(1-B)}{B}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{T-1} n_{0,0,0} \qquad (5-26)$$

そして，式(5–9)に式(5–23)と式(5–25)を代入して$n_{0,0,0}$について解いた式を式(5–20)に代入し，Bに関する次の式が得られる．

$$\left(1 - \frac{B(1-M\sigma)}{\frac{B(1-M\sigma)}{B-M\sigma} + \left\{\frac{M\sigma(1-B)}{B} - \frac{M\sigma(1-B)}{B-M\sigma}\right\}\left\{\frac{M\sigma(1-B)}{B(1-M\sigma)}\right\}^{T-1}}\right)$$
$$\times (1-E) + B = 1 \qquad (5-27)$$

与えられたσ,M,$E(=2/M)$についてこの式からBを求め，このBを式(5–20)〜(5–26)へ代入して$\mathbf{n_e}$の要素が全て求まる．

5.3.2 スループット

リング型のネットワークでは，M台のノード間にM個の伝送路がある．ここで考えるスループットとは，送られるスロットのうちフルスロットが送られる割合を各伝送路ごとに求め，この平均をとったものとする．$z=0$のモードにあるノードがフルスロットを送信し始めるのは，図5.4でP_1の経路以外で遷移した場合である．このため，このスループット$\bar{S}$は次式で得られる．

$$\bar{S} = 1 - n_{0,0,0}(1-M\sigma) \qquad (5-28)$$

5.3.3 平均パケット遅延

平均パケット遅延を求めるに際しては，この遅延を次の三つの要素に分け，各要素ごとに求めていく．

(D1) パケット発生から送信が開始されるまでの，端末バッファでの待ち時間D_1．

(D2) 送信開始後，このパケットが宛先のノードまでの途中にあるノードを転送される時間D_2．

(D3) 宛先のノードにてパケットを受信する時間D_3．

このうち，(D3)の受信時間D_3は，Mステップと一定である．

(D1)の端末バッファでの待ち時間D_1は，モデルの解析結果を用いて，以下のようにリトルの公式により求める．

平衡点におけるシステムの端末バッファ内にある全パケット数N_1を求める．端末バッファ内のパケットは，$W_{x,y,z}$モードにあるノードがy個蓄えているので，N_1は次のようになる．

$$N_1 = \sum_{y=1}^{T}\Big\{\sum_{z=1}^{M-1} y n_{0,y,z} + y(n_{0,y,0} + n_{1,y,0})\Big\} \qquad (5-29)$$

端末バッファで出力を待っているパケットは，$W_{0,y,0}$のモードにあるノードが遷移する際に1個出力される．さらに，$W_{0,0,0}$のモードにあるノードが遷移する際にも，確率$M\sigma$でパケットが端末バッファを経て出力される(この場合は，端末バッファ内での待ち時間は0)．このため，端末バッファ内のパケットから各ステップごとに出力されるパケットの個数Q_1は次のように与えられる．

$$Q_1 = M\sigma n_{0,0,0} + \sum_{y=1}^{T} n_{0,y,0} \qquad (5-30)$$

さらに，端末バッファへのパケットの発生を，$z=0$の時にMステップ分まとめて発生させているので，$M/2$ステップ加算する必要がある．

以上から，端末バッファ内での平均待ち時間D_1は，リトルの公式を用いて，次の式で与えられる．

$$D_1 = \frac{N_1}{Q_1} + \frac{M}{2} \qquad (5-31)$$

次に，(D2)の時間D_2を求める．送信を開始したパケットが，宛先のノードに到着するまでに，途中中継されるノード数の平均をGとする．Gは，パケットの宛先

が仮定(A6)の一様分布で表されることから，次の式で与えられる．

$$G = \sum_{k=0}^{M-2} k \times \frac{1}{M-1} = \frac{M-2}{2} \qquad (5-32)$$

モデルにおいて1台のノードにより中継される際に要する時間は1ステップである．更に式(5-7)を用いた補正を行い，実際のシステムにおける(D2)の平均パケット遅延D_2を与える次の式が得られる(ここで単位をステップにするための補正を加えた)．

$$D_2 = \frac{M}{a} \times \frac{(M-2)M}{2T_a} + \frac{(M-2)(g \times a - M)}{2M} \times \frac{M}{T_a} \qquad (5-33)$$

全体のパケット遅延Dは，各要素の合計として，次の式で与えられる．

$$D = D_1 + D_2 + M \qquad (5-34)$$

5.3.4 端末バッファオーバフロー確率

パケットが端末で発生したときに，端末バッファが一杯で，このパケットが破棄される確率を，端末バッファオーバフロー確率P_{ov}と定義する．

モデルにおいては，このオーバフローはノードが$W_{0,T,0}$もしくは$W_{1,T,0}$のモードにあるときに起きる．このため，モデルから次のP_{ov}を求める式が得られる．

$$P_{ov} = n_{0,T,0} + n_{1,T,0} \qquad (5-35)$$

5.4 数値例

これまでに検討したモデルによる解析結果をシミュレーションによる結果と比べ，解析の精度を調べる．

ここでは，ノード数Mを10,30,100とし，スロットのビット数であるaを257ビット(フラグ1ビット，データ部256ビット)として解析する．シミュレーションは仮定(A1)～(A6)のみを用い，他の簡単化のための仮定は用いていない．シミュレーション時間は5.4.2を除き330000Mステップとする．さらに，シングルラン法[KOBA78]により，95%の信頼区間を求める．しかし，以後示す各図においてはいずれも，シミュレーション値を示す記号(○△▽)の中に，この95%の信頼区間の範囲が含まれてしまうため，特にグラフには記入しない．

5.4.1 スループットと平均パケット遅延

端末バッファの段数$T = 10$の場合について，$M\sigma$とスループットとの関係を図5.6に，$M\sigma$と平均パケット遅延(パケット伝送時間で正規化)との関係を図5.7に示す．スループットと平均パケット遅延との関係は，$M = 100,30,10$の全てを1つの図にすると殆ど重なるため，図5.8,5.9,5.10にそれぞれ分けて示す．図中実線が解析によって得られた値を示し，○□▽がそれぞれ$M = 100,30,10$のシミュレーション結果を示す．各図とも解析値とシミュレーション結果とは良く一致しており，本解析の精度が高いことがわかる．

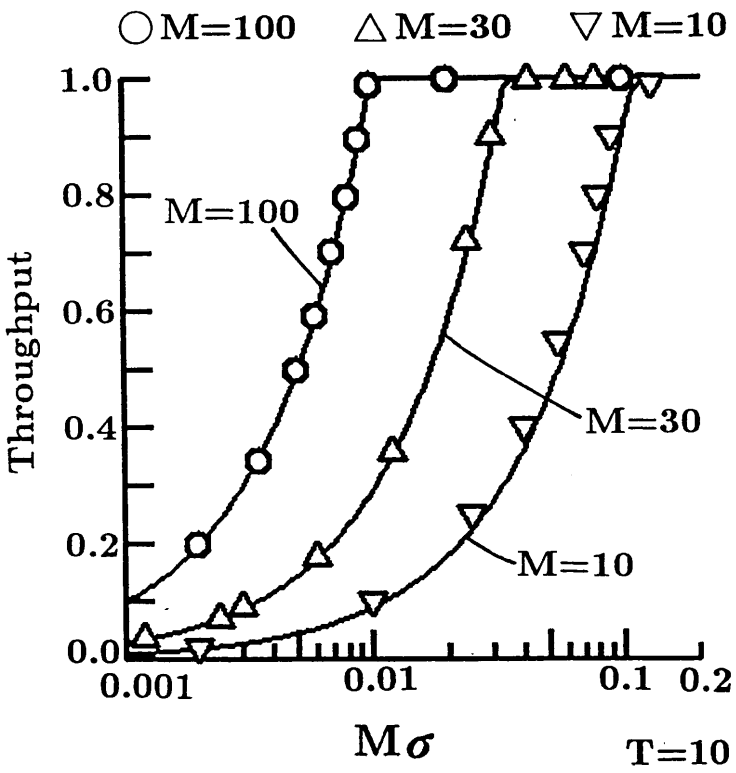


図5.6 スループットと$M\sigma$の関係($T = 10$)
Fig.5.6 Throughput versus $M\sigma$ for $T = 10$.

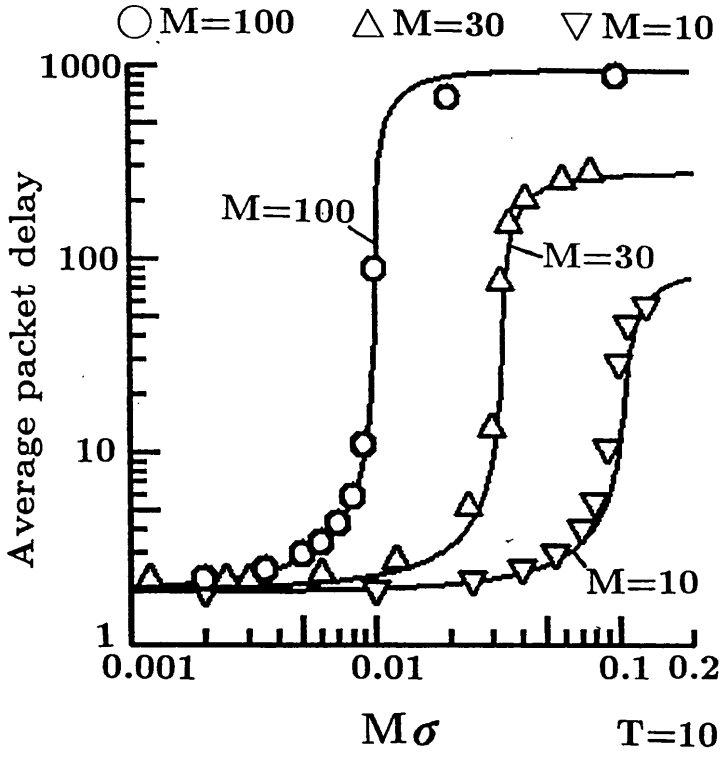


図5.7 平均パケット遅延と$M\sigma$の関係($T = 10$)
Fig.5.7 Average packet delay versus $M\sigma$ for $T = 10$.

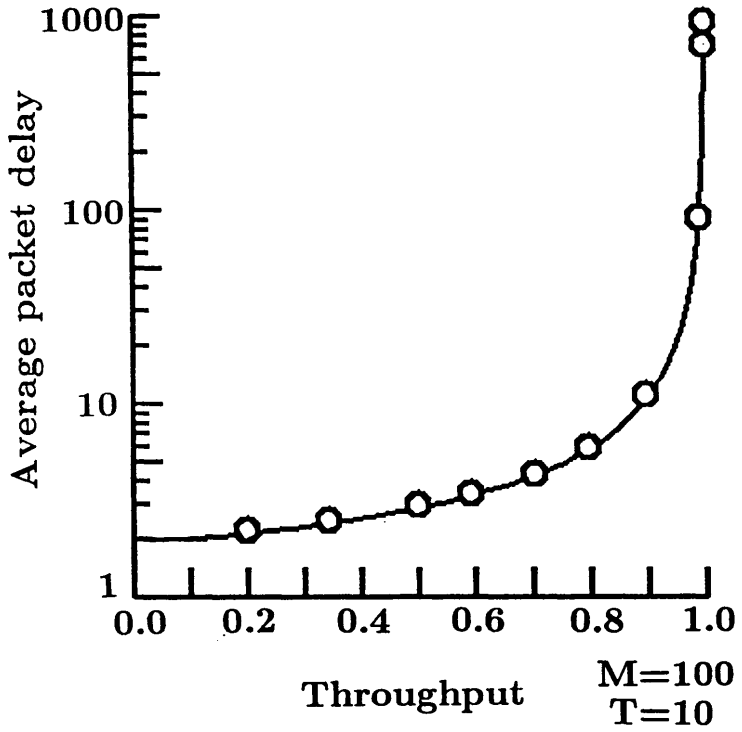


図5.8 平均パケット遅延とスループットの関係($M = 100, T = 10$)
Fig.5.8 Average packet delay versus throughput for $M = 100$ and $T = 10$.

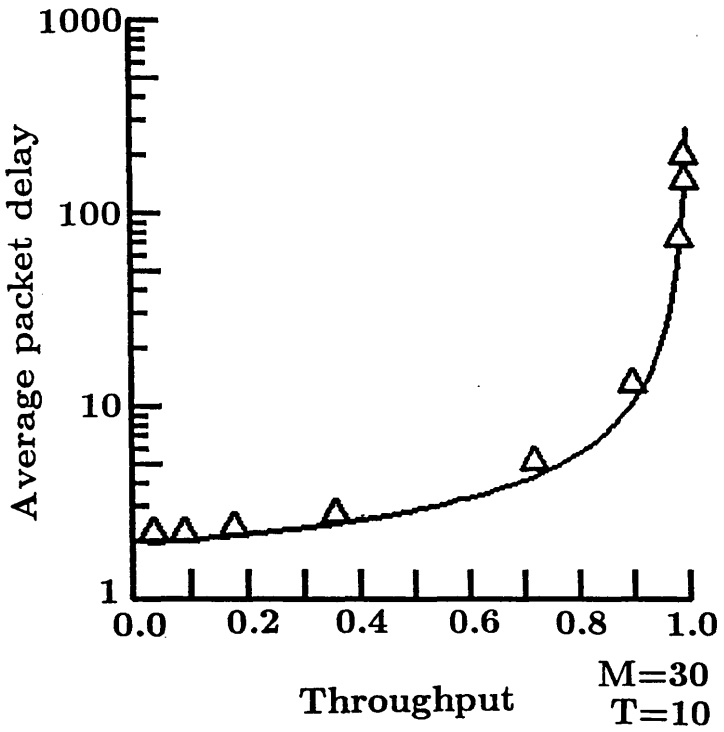


図5.9 平均パケット遅延とスループットの関係($M = 30, T = 10$)
Fig.5.9 Average packet delay versus throughput for $M = 30$ and $T = 10$.

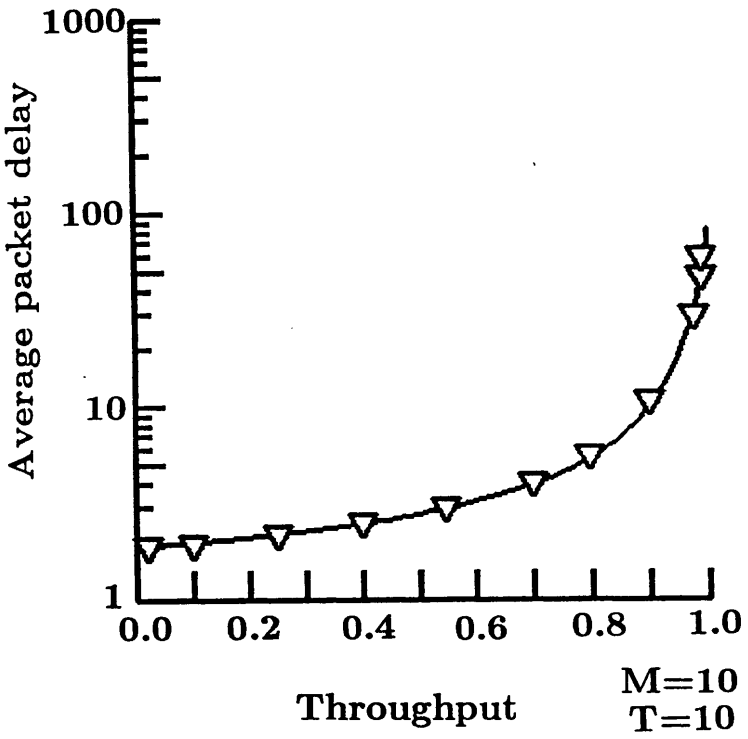


図5.10 平均パケット遅延とスループットの関係($M = 10, T = 10$)
Fig.5.10 Average packet delay versus throughput for $M = 10$ and $T = 10$.

5.4.2 端末バッファオーバフロー確率

図5.11に$M = 30$のときの$M\sigma$と端末バッファのオーバフロー確率の関係を，端末バッファの段数Tの値が2,5,10の場合について，それぞれ示す．図中実線が解析によって得られた値を示し，記号○△▽はそれぞれ$T = 2,5,10$の場合のシミュレーション結果を示す．シミュレーション時間は，オーバフロー確率が小さい場合も求められる様に，他より長い120000000ステップとする．解析値とシミュレーション結果とはよく一致しており，本解析により高い精度でバッファオーバフロー確率が得られることが分る．

オーバフローしたパケットは破棄されるため，ここで示すオーバフロー確率とパケットの破棄率とは一致する．このパケット破棄率をデータや映像のパケットに求められる1.0×10^{-9}以下[VERB87]とするためには，$T = 2,5,10$の場合それぞれσの値を0.000035, 0.00057, 0.00096以下とする必要があることがわかる．

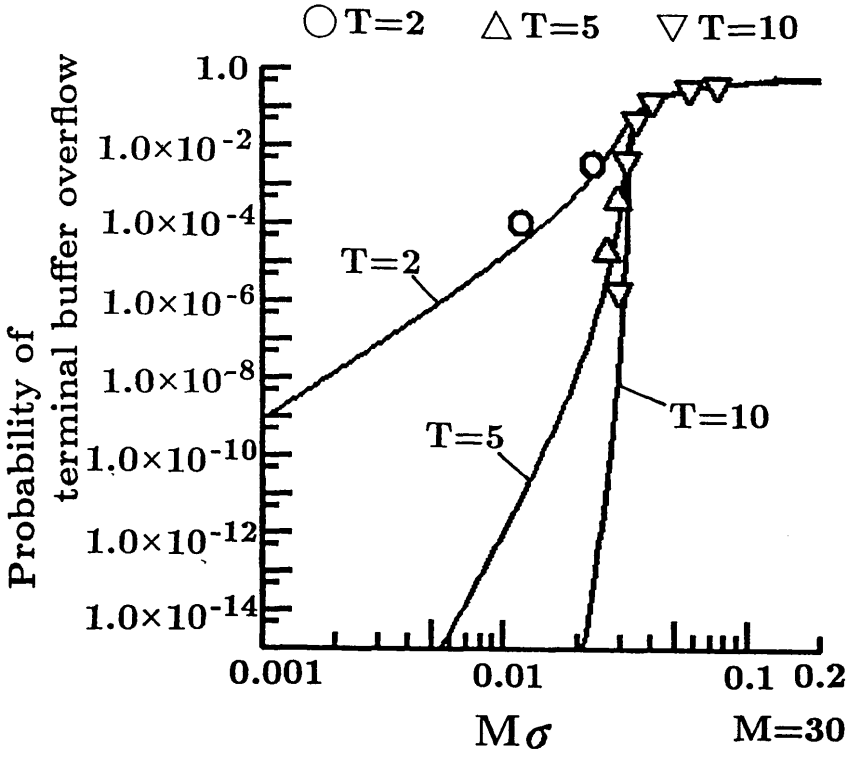


図5.11 端末バッファオーバフロー確率と$M\sigma$関係($M = 30$)
Fig.5.11 Probability of terminal buffer overflow versus $M\sigma$ for $M = 30$.

第6章 レジスタ挿入リングの性能解析

6.1 レジスタ挿入リング

解析するレジスタ挿入リングについて説明する．

各ノードの構成を図6.1に示す．ノードにはリングバッファ(Ring Buffer)，端末バッファ(Terminal Buffer)と呼ぶ2つのバッファを備える．

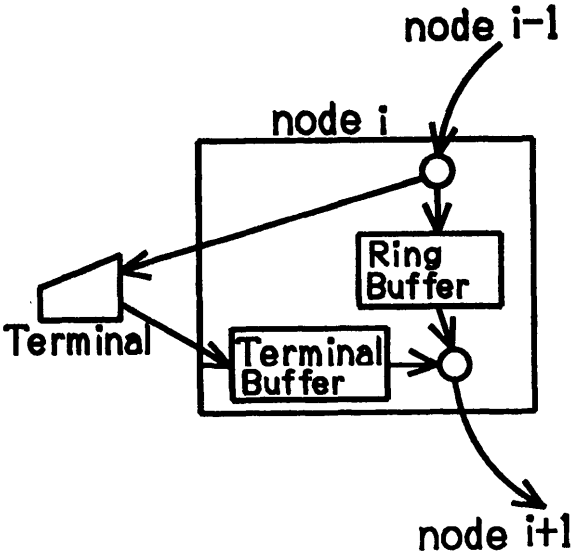


図6.1 ノードの構成
Fig.6.1 Block diagram of node.

隣接ノードから受信したパケット(パケットは固定長とする)は，ヘッダ部を受信した時点でその宛先が調べられる．そして，このパケットが自ノード宛の場合は，このパケットをこのノードに接続した端末へ送り，パケットをリングから取り除く．宛先が自ノード宛でない場合は，このパケットをリングバッファへ格納する．端末から発生するパケットは，端末バッファへ蓄える．

送信が可能で，リングバッファまたは端末バッファにパケットがある場合，直ちにパケットを1つ送信するが，この時，常にリングバッファからの送信を優先する．ここで，リングバッファからのパケット送信は，パケット全体のリングバッファへの格納を待たず，受信中のパケットの格納と同時に行うことができる．このため，

他ノード宛のパケットを次のノードへ転送する際の自ノード内での遅延は，最小，でパケットのヘッダ部受信時間とヘッダ解析時間の合計となる．

ここで，隣接ノードからのパケット受信は，スロットリングの様に一定時間ごとにあるわけではない．また，端末バッファにパケットがあり送信が可能であれば，直ちに送信を開始するため，送信処理と受信処理は並行して別々のタイミングで行われる．

このタイミングの複雑さは，他のMACプロトコルに無いものであり，このことによりノード内の全ての処理をハードウェア化することは困難となるか，ハードウェアの規模が大きくなる．そこで，CPUを用いて処理の一部をソフトウェアで実行することが一般的である．このため，性能解析にはこのソフトウェアによる処理のオーバヘッドを考慮する必要がある．

システムを解析するにあたっては，パケットの受信に要する時間をT_aとする．パケットの受信を開始してから，そのパケットのアドレス解析を終了するまでの時間をUステップとする．この時間には，ソフトウェアによる処理のオーバヘッドが含まれる．

6.2 モデル化

レジスタ挿入リングのネットワークを平衡点解析の手法で解析するため，ノードがとりうる状態(モード)の遷移の関係を表現したモデルを作成する．

6.2.1 モデル化の基本的な考え方

まず，モデル作成の基本的な考え方について述べる．

レジスタ挿入リングでは，各ノードが独自のタイミングでパケット送信を開始する．このため，トークンリングやスロットリングの解析で用いたモデルのように，ネットワーク上のM台ある全てのノードの振舞いを1つのモデルで表すことは，レジスタ挿入リングでは困難である．このため，モデルでは1台のノードの振舞いのみを表し，ネットワーク全体をM個のモデルで表すこととした．

また，レジスタ挿入リングにおいては，各ノードの状態も，パケットの送信と受信の状態，リングバッファ，端末バッファの状態と4つの状態の組合せから成り，多くのモードが必要になる．さらに，受信バッファのアドレス解析中に端末からパケットが発生し，このパケットの送信を開始したり，送信中に隣接ノードからパケットを受信し，このアドレス解析を送信と並行して行う等の動作を考えると，モード間の遷移は大変複雑なものになる．

このため，各ノードの振舞いを表すモデルを，さらに4つのサブモデルに分け，各サブモデルではそれぞれ，送信，受信，リングバッファ，端末バッファの振舞いを別々に表すことで，モデルの簡単化を図った．

6.2.2 パケット宛先の近似

パケットの宛先に関し，仮定(A6)のもとでは一様分布のため，分布に記憶が生じ，モデル化が極めて複雑になる．そこで，モデル化を容易にするために，宛先アドレスを幾何分布で近似し，無記憶性を保証した次の仮定(RE1)を，仮定(A6)の代りに導入する．

(RE1) 隣接ノードからパケットを受信したノードでは，このパケットの宛先が確率$E = 2/M$で自ノード宛であるとする．

ここで確率Eの値は，パケットが送信を開始されてから宛先ノードに達するまで，途中パケットを中継するノード数の平均が，仮定(A6)と(RE1)とで一致するように定め，次式を得た．

$$E = \frac{2}{M} \tag{6-1}$$

なお，この変形は，あくまでもモデルの簡単化のために行うものであり，実際のシステムを変形するものではない．

6.2.3 モデル化

作成したモデルを図6.2に示す．モデルは，リングバッファの振舞いを示すサブモデル(SUB1)，端末バッファの振舞いを示すサブモデル(SUB2)，隣接ノードからのパケット受信の振舞いを示すサブモデル(SUB3)，パケット送信の振舞いを示すサブモデル(SUB4)からなる．

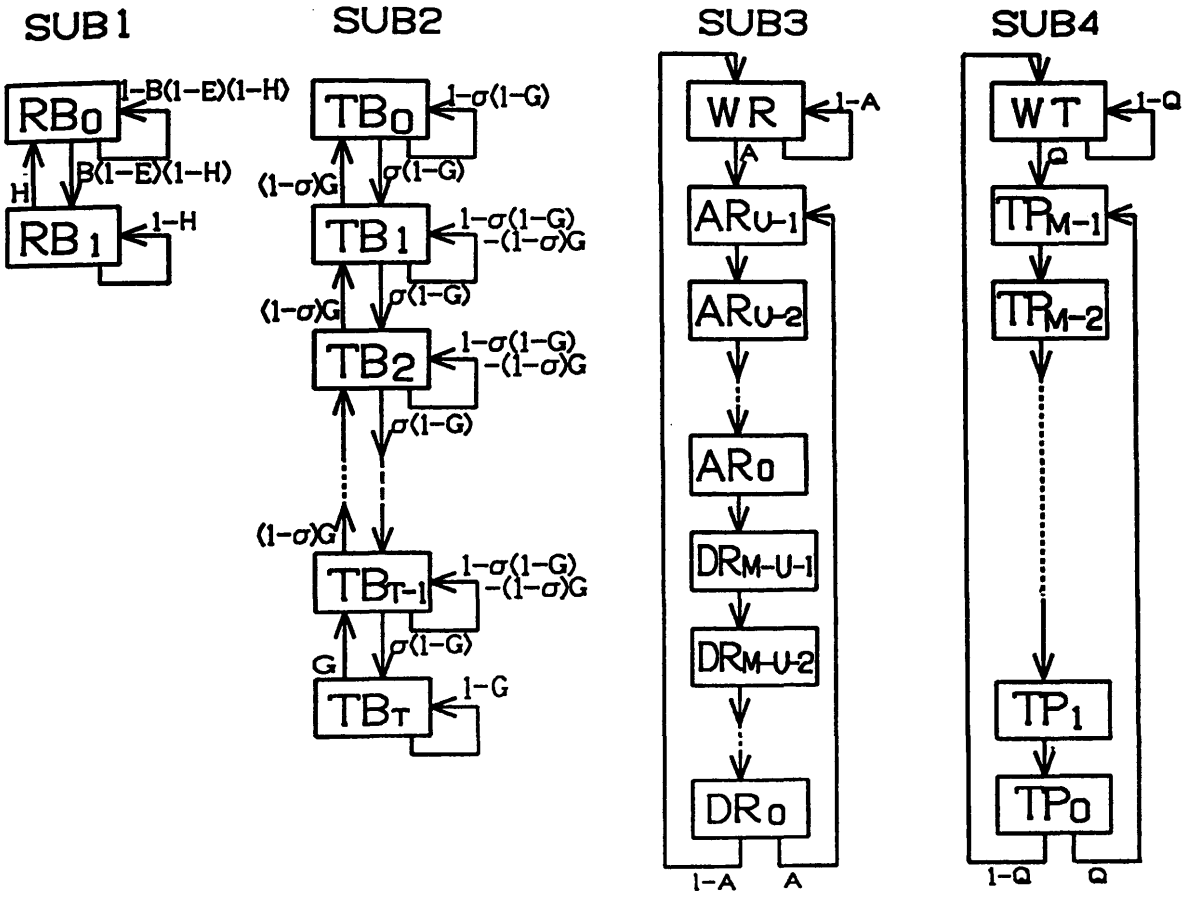


図6.2 レジスタ挿入リングの近似モデル
Fig.6.2 An approximate model of a register insertion ring.

ここで，このモデルの状態の観察，パケットの発生と送信，モードの遷移のタイミングは，次のようになる．

(T1) 状態の観察は各ステップの最初に行う．

(T2) モードの遷移は各ステップの最後に行う．

(T3) 送信を開始するノードは，ステップの終了時に送信を行う(但し(T2)の直前)．

(T4) 端末からのパケット発生はステップの終了時に起きる(但し(T3)の直前)．

[1] サブモデルSUB1

リングバッファの振舞いを表すSUB1は，リングバッファが空である状態のRB_0モードと，リングバッファ内にパケットがある状態のRB_1モードの2つのモードからなる．ここで変数Bは，隣接ノードから受信したパケットのアドレス解析が，現在のステップにおいて終了する確率を示す．また変数Hは，現在のステップ終了時に，リングバッファからパケットの送信を開始できる状態となる確率である．つまり，パケットの送信を行っていないか，ステップ終了時にパケットの送信を終了し，新たなパケット送信が可能となる確率である．

ノードがRB_0モードにあるとき，確率Bでパケットのアドレス解析を終了し，$1-E$の確率でこのパケットは他のノード宛である．このとき，確率Hでこのパケットは直ちに次ノードへ送信が開始されるが，確率$1-H$でこのパケットはリングバッファに蓄えられる．このため，ノードがRB_0モードにあるとき，確率$B(1-E)(1-H)$で，RB_1モードへ遷移する．

ノードがRB_1モードにあるとき，確率Hでリングバッファ内にあるパケットの送信を開始し，RB_0モードへ遷移する．

[2] サブモデルSUB2

端末バッファの振舞いを表すSUB2は，端末バッファ内のパケット数がi $(i = 0, 1, \cdots, T)$のとき，そのノードはTB_iモードとなる．ここで変数Gは，現在のステップ終了時に，端末バッファからパケットの送信を開始できる状態となる確率である．つまり，パケットの送信を行っていないか，ステップ終了時にパケットの送信を終了し新たなパケット送信が可能となり，さらにリングバッファからパケットの送信を開始しない確率である．

ノードがTB_0モードにあるとき，確率σでパケットが発生すると，確率Gでこのパケットは直ちに次ノードへ送信が開始されるが，確率$1-G$でこのパケットは端末バッファに蓄えられる．このため，ノードがTB_0モードにあるとき，確率$\sigma(1-G)$でTB_1モードへ遷移する．

ノードが$TB_i(i = 1, 2, \cdots, T-1)$モードにあるとき，確率$\sigma$でパケットが発生した場合について考える．このとき，このパケットは端末バッファに蓄えられるが，確率Gで端末バッファからバッファ内の先頭にあるパケットの送信が開始され，結局端末バッファ内のパケット数は変らず，このノードはTB_iモードに留る．一方確率$1-G$で端末バッファからパケット送信が開始されないときは，端末バッファ内のパケットは1つ増え，このノードはTB_{i+1}モードへ遷移する(確率$\sigma(1-G)$)．また，確率$1-\sigma$でパケットが発生しない場合，確率Gで端末バッファからのパケット送信が開始され，ノードはTB_{i-1}モードへ遷移する(確率$(1-\sigma)G$)．

ノードがTB_Tモードにあるときは，パケットが発生してもオーバフローとなり，これ以上パケットは端末バッファへ入らず，確率Gでパケット送信を開始してTB_{T-1}モードへ遷移する．

[3] サブモデルSUB3

パケットの受信状態を示すSUB3は，隣接ノードからのパケット到着を待つ状態を示すWRモード，パケットのアドレス部の受信と解析中である状態を示すAR_iモード$(i=0,1,\cdots,U-1)$，アドレス解析終了後，残りの部分を受信中であることを示すDR_iモード$(i=0,1,\cdots,M-U-1)$からなる．ここで変数Aは，現在のステップ終了時に，隣接ノードからのパケットの受信を開始する確率である．

ノードがパケットの到着待ちの状態でWRモードにあるとき，確率Aでパケットの受信を開始し，AR_{U-1}モードへ遷移する．以後，パケットのアドレス部の受信と解析を終了するまでのUステップの間，AR_{U-1}モードからAR_0モードまで，確率1で遷移する．ノードがAR_0モードにあるとき，このステップでアドレス解析を終了し，以後パケットの残りを受信するため，DR_{M-U-1}モードからDR_0モードまで，確率1で遷移する．

ノードがDR_0モードにあるとき，このステップでパケットの受信を終了する．このとき，確率Aで隣接ノードから新たなパケットが到着してAR_{U-1}モードへ遷移し，確率$1-A$で到着待ちのWRモードへ遷移する．

[4] サブモデルSUB4

パケットの送信状態を示すSUB4は，送信していない状態WTモードと，送信中の状態$TP_i(i=0,1,\cdots,M-1)$モードからなる．ここで変数Qは，現在のステップ終了時に，新たなパケット送信を開始する確率である．

ノードがWTモードにあるとき，確率Qでパケットの送信を開始し，TP_{M-1}モードへ遷移する．以後，パケットの送信を終了するまでのMステップの間，TP_{M-1}モードからTP_0モードまで，確率1で遷移する．ノードがTP_0モードにあるとき，このステップでパケット送信を終了し，確率Qで更に新たなパケット送信を開始してTP_{M-1}モードへ遷移し，確率$1-Q$で送信をしていないWTモードへ遷移する．

6.3 モデルの解析

6.3.1 平衡点方程式

作成したモデルの解析を行う．このモデルにおいて，RB_0, RB_1, TB_h $(h = 0,1,\cdots,T)$, WR, AR_i $(i = 0,1,\cdots,U-1)$, DR_j $(j = 0,1,\cdots,M-U-1)$, WT, TP_k $(k = 0,1,\cdots,M-1)$ の各モードにあるノード数を表す確率変数を，それぞれr_{b0}, r_{b1}, t_{bh}, w_r, a_{ri}, d_{rj}, w_t, t_{pk}とおく．このとき，モデルの状態ベクトルnを次のように定義する．

$$\begin{aligned}\mathbf{n} = \{&r_{b0}, r_{b1},\\ &t_{b0}, t_{b1}, \cdots, t_{bT},\\ &w_r, a_{r0}, a_{r1}, \cdots, a_{rU-1}, d_{r0}, d_{r1}, \cdots, d_{rM-U-1},\\ &w_t, t_{p0}, t_{p1}, \cdots, t_{pM-1}\}\end{aligned}$$

このベクトルnは，既約な有限状態マルコフ連鎖となる．

モデルの定義より明らかに次の式が成り立つ．

$$r_{b0} + r_{b1} = 1 \tag{6-2}$$

$$\sum_{i=0}^{T} t_{bi} = 1 \tag{6-3}$$

$$w_r + \sum_{i=0}^{U-1} a_{ri} + \sum_{i=0}^{M-U-1} d_{ri} = 1 \tag{6-4}$$

$$w_t + \sum_{i=0}^{M-1} t_{pi} = 1 \tag{6-5}$$

このマルコフ連鎖を，平衡点解析の手法を用いて解析する．平衡点解析では，システムは常に平衡点に留っていると仮定する．平衡点は，各モードにおける平均増加ノード数が0となる点と定義される．そのため，状態nが与えられたとき，各モードへ流入する平均ノード数と各モードから流出する平均ノード数とが等しくなるように式をたて，平衡点を求める次の式を得る．

$$H\ r_{b1} = B(1-E)(1-H)r_{b0} \tag{6-6}$$

$$(1-\sigma)G\ t_{b1} = \sigma(1-G)t_{b0} \tag{6-7}$$

$$(1-\sigma)G\,t_{bi+1}+\sigma(1-G)t_{bi-1}$$
$$=\{\sigma(1-G)+(1-\sigma)G\}t_{bi}\quad(i=1,2,\cdots,T-2)\qquad(6-8)$$

$$G\,t_{bT}+\sigma(1-G)t_{bT-2}$$
$$=\{\sigma(1-G)+(1-\sigma)G\}t_{bT-1}\qquad(6-9)$$

$$\sigma(1-G)t_{bT-1}=G\,t_{bT}\qquad(6-10)$$

$$(1-A)d_{r0}=A\,w_r\qquad(6-11)$$

$$A(w_r+d_{r0})=a_{rU-1}\qquad(6-12)$$

$$a_{ri+1}=a_{ri}\qquad(i=0,1,\cdots,U-2)\qquad(6-13)$$

$$a_{r0}=d_{rM-U-1}\qquad(6-14)$$

$$d_{ri+1}=d_{ri}\qquad(i=0,1,\cdots,M-U-2)\qquad(6-15)$$

$$(1-Q)t_{p0}=Q\,w_t\qquad(6-16)$$

$$Q(w_t+t_{p0})=t_{pM-1}\qquad(6-17)$$

$$t_{pi+1}=t_{pi}\qquad(i=0,1,\cdots,M-2)\qquad(6-18)$$

ここで式(6–6)はRB_0とRB_1モードに関する式であり，式(6–7)はTB_0モード，式(6–8)はTB_i $(i=1,2,\cdots,T-2)$モード，式(6–9)はTB_{T-1}モード，式(6–10)はTB_Tモード，式(6–11)はWRモード，式(6–12)はAR_{U-1}モード，式(6–13)はAR_i $(i=0,1,\cdot,U-2)$モード，式(6–14)はDR_{M-U-1}モード，式(6–15)はDR_i $(i=0,1,\cdots,M-U-2)$モード，式(6–16)はWTモード，式(6–17)はTP_{M-1}モード，式(6–18)はTP_i $(i=0,1,\cdots,M-2)$モードに関する式である．

次に，ノード間を転送されるパケットの数について考える．サブモデルSUB4において，WTモードもしくはTP_0モードにあるノードは，確率QでTP_{M-1}モードへ遷移し，パケットの送信を開始する．そのステップごとの平均数は$Q(w_t+t_{p0})$となる．一方，次のノードにおいて，サブモデルSUB3のWRモードもしくはDR_0モードにそのノードがある場合，確率AでAR_{U-1}モードへ遷移し，パケットの受信を開始する．そのステップごとの平均数は，$A(w_r+d_{r0})$である．この2つの値は等しくなるので，次式を得る．

$$Q(w_t+t_{p0})=A(w_r+d_{r0})\qquad(6-19)$$

ステップごとにアドレス解析が終了するパケット数の平均はサブモデルSUB3よりa_{r0}であり，その内$1-E$の割で宛先が他のノードとなるため，リングバッファへ送られるパケット数の平均は$(1-E)a_{r0}$となる．一方，サブモデルSUB1よりステップごとにリングバッファに送られるパケット数の平均は$B(1-E)r_{b0}$となる．この2つの値は等しくなるので，次式を得る．

$$(1-E)a_{r0} = B(1-E)r_{b0} \tag{6-20}$$

ステップごとの送信開始パケット数の平均は，サブモデルSUB4より$Q(w_t+t_{p0})$である．この数は，リングバッファからの送信数と端末バッファからの送信数の合計である．リングバッファからのステップごとの平均送信開始パケット数は，サブモデルSUB1より$H\ r_{b1}+B(1-E)H\ r_{b0}$である．端末バッファからのステップごとの平均送信パケット数は，サブモデルSUB2より$\sigma G\ t_{b0}+G\sum_{i=1}^{T} t_{bi}$となり，これは式(6-3)より$G\{1+(\sigma-1)t_{b0}\}$となる．このことから，次の式を得る．

$$\begin{aligned} Q(w_t+t_{p0}) = {} & H\ r_{b1}+B(1-E)H\ r_{b0} \\ & +G\{1+(\sigma-1)t_{b0}\} \end{aligned} \tag{6-21}$$

Gは，各ステップの最後に，端末バッファからパケットの送信を開始できる状態となる確率である．つまり，パケットの送信を行っていないか，ステップ終了時にパケットの送信を終了し，新たなパケット送信が可能となり，さらにリングバッファがパケットの送信を開始しない確率である．サブモデルSUB1とSUB4より，Gを求める次の式が得られる．

$$G = \{1-B(1-E)\}r_{b0}(w_t+t_{p0}) \tag{6-22}$$

Hは，各ステップの最後に，リングバッファからパケットの送信を開始できる状態となる確率である．つまり，パケットの送信を行っていないか，ステップ終了時にパケットの送信を終了し，新たなパケット送信が可能となる確率である．サブモデルSUB4より，Hを求める次の式が得られる．

$$H = w_t+t_{p0} \tag{6-23}$$

この式(6-2)〜(6-23)を解くことで，平衡点における状態ベクトル$\mathbf{n}_e$の各要素が得られる．

式(6–6)〜(6–18)より，次の式が得られる(以下，$X = \frac{\sigma(1-G)}{(1-\sigma)G}$とおく)．

$$r_{b1} = \frac{B(1-E)(1-H)}{H} r_{b0} \tag{6-24}$$

$$t_{bi} = X^i t_{b0} \quad (i = 1, 2, \cdots, T-1) \tag{6-25}$$

$$t_{bT} = \frac{\sigma(1-G)}{G} X^{T-1} t_{b0} \tag{6-26}$$

$$d_{ri} = \frac{A}{1-A} w_r \qquad (i = 0, 1, \cdots, M-U-1) \tag{6-27}$$

$$a_{ri} = \frac{A}{1-A} w_r \qquad (i = 0, 1, \cdots, U-1) \tag{6-28}$$

$$t_{pi} = \frac{Q}{1-Q} w_t \qquad (i = 0, 1, \cdots, M-1) \tag{6-29}$$

この式(6–24)〜(6–29)を式(6–2)〜(6–5)へ代入して，次の式を得る．

$$r_{b0} = \frac{H}{H + B(1-E)(1-H)} \tag{6-30}$$

$$t_{b0} = \frac{X-1}{\frac{G+\sigma(1-G)}{G} X^T - \frac{\sigma(1-G)}{G} X^{T-1} - 1} \tag{6-31}$$

$$w_r = \frac{1-A}{1+(M-1)A} \tag{6-32}$$

$$w_t = \frac{1-Q}{1+(M-1)Q} \tag{6-33}$$

以上の各式を用い，次式が得られる．

$$G = \frac{1 + (M+E-2)Q - (1-E)(M-1)Q^2}{\{1+(M-1)Q\}^2} \tag{6-34}$$

$$\frac{-E\,Q}{1+(M-1)Q} + G\left[1 + (\sigma-1)\frac{X-1}{\frac{G+\sigma(1-G)}{G} X^T - \frac{\sigma(1-G)}{G} X^{T-1} - 1}\right] = 0 \tag{6-35}$$

式(6–35)へ式(6–34)を代入して得た式に対して，与えられたM,$E(=2/M)$,σから，この式を満たすQ，すなわち平衡点におけるQの値Q_eが求まる$(0 \le Q_e \le 1)$．この値と各式から，$\mathrm{n_e}$の要素が全て求まる．

6.3.2 スループット

リング型のネットワークでは，M台のノード間にM個の伝送路がある．ここで考えるスループットとは，パケットが送られる割合を各伝送路ごとに求め，この平均をとったものとする．

サブモデルSUB4のモードが，WT以外になっているとき，そのノードは送信中である．その確率は，与えられたnに対して，$1-w_t$となる．この式は，Mチャネルある伝送路全てに対して成立するので，次のスループットの式が得られる．

$$\bar{S} = 1 - w_t \tag{6-36}$$

6.3.3 平均パケット遅延

平均パケット遅延を求めるに際しては，この遅延を次の三つの要素に分けて考え，各要素ごとに求めることとする．

(D1) パケット発生から送信が開始されるまでの，端末バッファでの待ち時間D_1.

(D2) 送信開始後，このパケットが宛先のノードまでの途中にあるノードを転送される時間D_2.

(D3) 宛先のノードにてパケットを受信する時間D_3.

このうち，(D3)の受信時間D_3は，Mステップと一定である．

(D1)の端末バッファでの待ち時間D_1は，モデルの解析結果を用いて，以下のようにリトルの公式により求める．

各ノードの端末バッファ内にある平均パケット数N_1を求める．端末バッファ内のパケットは，$TB_i(i=1,2,\cdots,T)$モードにあるノードが各々i個蓄えているので，N_1は次のようになる．

$$N_1 = \sum_{i=1}^{T} i\, t_{bi} \tag{6-37}$$

端末バッファから各ステップごとに出力されるパケットの平均個数λ_1は，式(6-21)を得たときに求めたように，次式で与えられる．

$$\lambda_1 = G\{1 + (\sigma - 1)t_{b0}\} \tag{6-38}$$

以上を用い，端末バッファ内での平均待ち時間D_1は，リトルの公式より，次の式で与えられる(単位:ステップ)．

$$D_1 = \frac{N_1}{\lambda_1} \tag{6-39}$$

次に，(D2)の時間を求める．送信を開始したパケットが，宛先のノードに到着するまでに，途中中継されるノード数の平均をLとする．Lは，パケットの宛先が仮定(RE1)の幾何分布で表されることから，次の式で与えられる．

$$L = \sum_{k=0}^{\infty} k\, E(1-E)^k = \frac{1-E}{E} \tag{6-40}$$

1台のノードにより中継される際に要する時間の平均D_aを，モデルの解析結果を用いて，次のように求める．

ノードがRB_1 モードにあるときに，リングバッファ内に1個パケットを蓄えるので，リングバッファ内の平均パケット数N_aは，次の式で得られる．

$$N_a = r_{b1} \tag{6-41}$$

リングバッファから各ステップごとに出力されるパケットの平均個数λ_aは，式(6-21)を得たときに求めたように，次式で与えられる．

$$\lambda_a = H\, r_{b1} + B(1-E)H\, r_{b0} \tag{6-42}$$

式(6-41)(6-42)より，リトルの公式を用い，リングバッファ内の平均待ち時間はN_a/λ_aで得られる．さらに，中継ノードではアドレス解析時間Uステップが必要である．以上より，D_aは次の式により得られる(単位:ステップ)．

$$D_a = \frac{N_a}{\lambda_a} + U = \frac{1-H}{H} + U \tag{6-43}$$

(D2)の転送時間の平均D_2は，式(6-40)(6-43)より，次式で与えられる．

$$D_2 = \left(\frac{1-H}{H} + U\right)\frac{1-E}{E} \tag{6-44}$$

全体のパケット遅延Dは，上記の(D1)(D2)(D3)の各要素の合計として，次の式で与えられる．

$$D = D_1 + D_2 + M \tag{6-45}$$

6.3.4 端末バッファオーバフロー確率

パケットが端末で発生したときに，端末バッファが一杯で，このパケットが破棄される確率を，端末バッファオーバフロー確率P_{ov}と定義する．

モデルにおいて，このオーバフローはノードがTB_Tモードにあるときに起きる．このため，モデルから次のP_{ov}を求める式が得られる．

$$P_{ov} = t_{bT} \tag{6-46}$$

6.4 数値例

これまでに検討したモデルによる解析結果をシミュレーションによる結果と比べ，解析の精度を調べる．

ここでは，ノード数Mを10,30,100とし，パケット長を256ビット(内ヘッダ長を16ビット)として解析する．また，通信速度は100Mbpsとする．

レジスタ挿入リングのヘッダ解析に要するCPU処理のオーバヘッド時間は，筆者が設計，製作したノードのプログラムから求めた．このノードは，レジスタ挿入リングによるネットワークS-net[FUWA87][FUWA88b]において用いられている．このプログラムでは，パケット受信と並行して宛先アドレスや送信の状況等を調べ，パケットのヘッダ部を受信した時には，あとは転送用のハードウェアに対する指示パルスを出すだけとなるように，設計を行った（アドレスの比較はハードウェアにより行う）．この結果，オーバヘッド時間はパルス出力の命令実行時間である500nsecとなる（使用CPUは，ザイログ社製Z80でこれを20MHzのクロックで使用）．伝送速度が100Mbpsの場合，この500nsecは50ビットを受信する時間に等しい．このため，ヘッダ受信時間とオーバヘッド時間の合計Uは，$\frac{(16+50)}{256}M$ステップとなる．

シミュレーションは仮定(A1)〜(A6)を用い，他のモデルの簡単化のための仮定は用いていない．シミュレーション時間は6.4.2 を除き330000Mステップとする．さらに，シングルラン法[KOBA78]により，95%の信頼区間を求める．しかし，以後示す各図においてはいずれも，シミュレーション値を示す記号(○△▽)の中に，この95%の信頼区間の範囲が含まれてしまうため，特にグラフには記入しない．

6.4.1 スループットと平均パケット遅延

端末バッファの段数$T = 10$の場合について，$M\sigma$とスループットとの関係を図6.3に，$M\sigma$と平均パケット遅延(パケット伝送時間で正規化)との関係を図6.4に示す．図中実線が解析によって得られた値を示し，○□▽がそれぞれ$M = 100,30,10$のシミュレーション結果を示す．各図とも解析値とシミュレーション結果とは良く一致しており，本解析の精度が高いことがわかる．

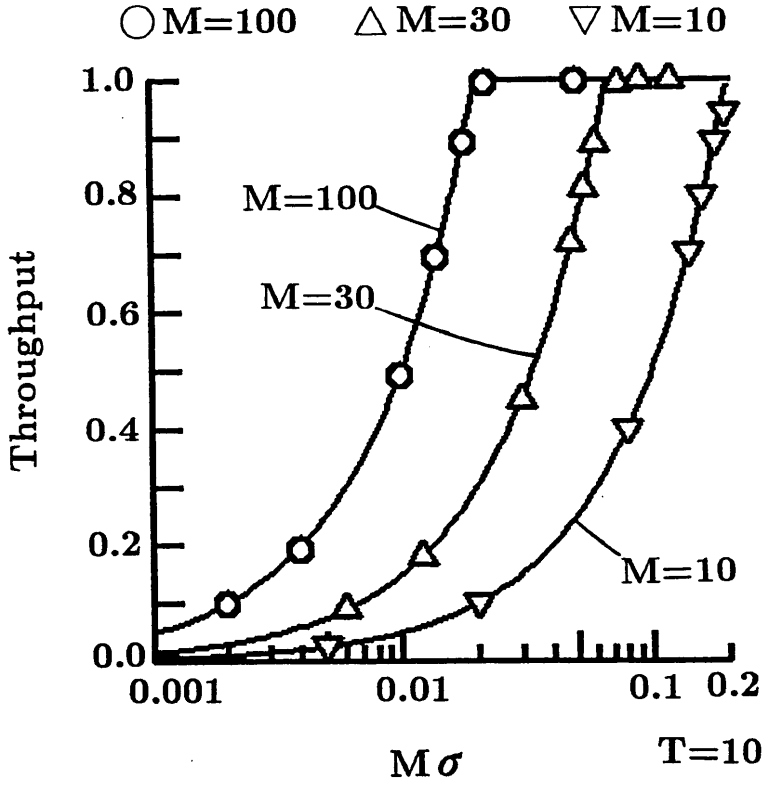


図6.3 スループットと$M\sigma$の関係($T = 10$)
Fig.6.3 Throughput versus $M\sigma$ for $T = 10$.

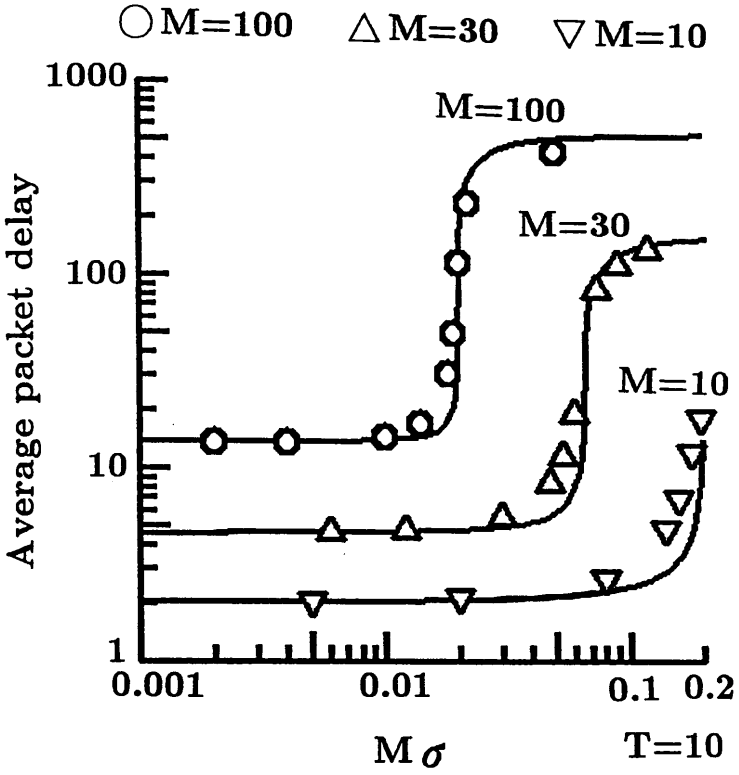


図6.4 平均パケット遅延と$M\sigma$の関係($T = 10$)
Fig.6.4 Average packet delay versus $M\sigma$ for $T = 10$.

6.4.2 端末バッファオーバフロー確率

図6.5に$T = 10$のときの$M\sigma$と端末バッファのオーバフロー確率の関係を，ノード数Mの値が100,30の場合について，それぞれ示す．図中，実線が解析によって得られた値を示し，記号○△はそれぞれ$M = 100,30$の場合のシミュレーション結果を示す．シミュレーション時間は，オーバフロー確率が小さい場合も求められる様に，他より長い120000000ステップとする．解析値とシミュレーション結果とはほぼ一致しており，本解析により得られたオーバフロー確率の有効性が確かめられた．

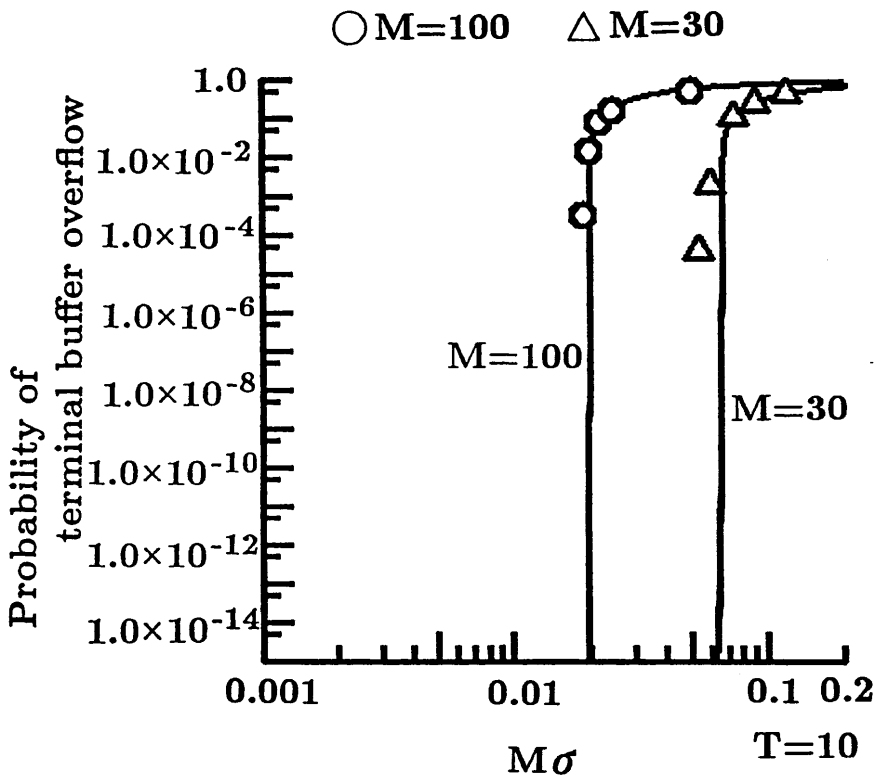


図6.5 端末バッファオーバフロー確率と$M\sigma$関係($T = 10$)
Fig.6.5 Probability of terminal buffer overflow versus $M\sigma$ for $T = 10$.

第7章 MACプロトコルの適用領域

第2部では，トークンリング，スロットリング，レジスタ挿入リング，RISリングの性能解析を行った．最後に，これまでの解析結果を用いて，様々な条件下での解析の各プロトコルの数値例を示す．そして各MACプロトコルの性能比較を行い，各プロトコルが各々どの様な条件下でもっとも性能がよいかという，各プロトコルの有効適用領域を明らかにする．

7.1 数値例

数値例は，次の条件下で求める．パケット長は各プロトコルとも256ビット（内ヘッダ長は16ビット）とする．トークンリングにおけるトークン長は，フリートークン，ビジートークンともに8ビットとする（$H = 256/8 = 32$）．各ノード内の端末バッファの段数Tは10段とし，通信速度は100Mbpsとする．

スロットリングにおけるスロット長は，パケット長に1ビットのフラグを付けて257ビットとする．この結果，スロットリング以外のプロトコルにおけるT_aが256ビット受信時間であるのに対し，スロットリングにおけるT_aは257ビット受信時間となる．そこで，ステップ(T_a/M)当りのパケット発生確率σは，スロットリングでは$\sigma' = \frac{257}{256}\sigma$で置き換えることとする．

レジスタ挿入リングのヘッダ解析に要するCPU処理のオーバヘッド時間は，5.で述べた様に筆者が設計，製作したノードのプログラムから求め，ヘッダ受信時間とオーバヘッド時間の合計Uは，$\frac{(16+50)}{256}M$ ステップとする．

シミュレーションは仮定(A1)〜(A6)を用い，他の仮定は用いていない．シミュレーション時間は，スループットと平均パケット遅延を求める場合は$330,000M$ステップとする．端末バッファオーバフロー確率を求める場合は，確率が小さい場合も求められる様に，より長い120,000,000 ステップとする．更に，シングルラン法[KOBA78]により，95%の信頼区間を求める．しかし，以後示す各図においてはいずれも，シミュレーション値を示す記号(○△□▽)の中にこの95%の信頼区間の範囲が含まれてしまうため，特にグラフには記入していない．

7.1.1 スループットと平均パケット遅延

ノードの台数$M = 30$の場合について，ネットワークの負荷である$M\sigma$とスループット，平均パケット遅延(パケット伝送時間で正規化) との関係を図7.1に示す．図中，実線が解析値を示し，○△□▽がそれぞれトークンリング，スロットリング，レジスタ挿入リング，RISリングのシミュレーション結果を示す．

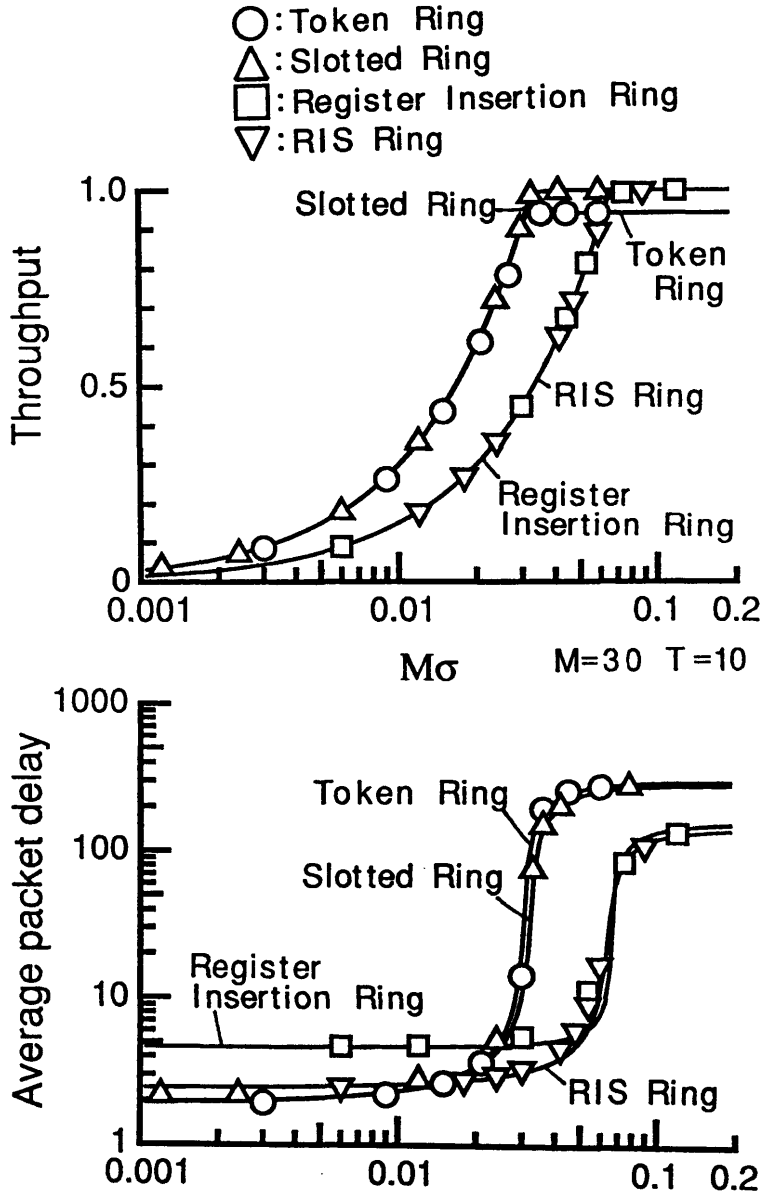


図7.1 スループット，平均パケット遅延と$M\sigma$の関係($M = 30, T = 10$)
Fig.7.1 Throughput and average packet delay versus $M\sigma$ for $M = 30$ and $T = 10$.

解析値とシミュレーション結果とは良く一致しており，解析の精度が各プロトコル共高いことが分る．

スループットの結果では，トークンリングとスロットリングの値，レジスタ挿入リングとRISリングの値が，それぞれほぼ同じ値となっている．また，トークンリングとスロットリングの値が，常にレジスタ挿入リングとRISリングより大きいことも分る．これはパケットをリングから除去する方法が異なるからである．すなわち，トークンリングとスロットリングはパケットの送信元のノードがリングを一周してきたパケットを取り除く．これに対して，レジスタ挿入リングとRISリングはパケットの宛先ノードが取り除く．つまり，トークンリングとスロットリングのスループットには，宛先ノードに達した後送信元ノードに戻るパケットという無効なトラヒックが含まれる（この送信元に戻るパケットを用いて，宛先ノードが受信確認のためのACKを返す場合は，無効なトラヒックとはならないが，本論文ではMACプロトコルによるACKは考えない）．又，図でトークンリングのスループットの最大値が1にならないのは，各パケット間に必ずトークンが入るためである．

平均パケット遅延の結果からは，次のことが分る．

負荷が小さいときは，トークンリングとスロットリングの遅延が小さい．スロットリングでは，各ノードがパケットを次ノードへ転送する際の転送遅延時間がフラグ受信に要する1ビット受信時間，トークンリングではトークン受信に要する8ビット受信時間ですむ．これに対して，RISリングではヘッダ受信に要する16ビット受信時間，レジスタ挿入リングではヘッダ受信時間に加えてCPUによる処理のオーバヘッド時間も必要となる．負荷が軽い場合には，これらの転送時間がパケット遅延の支配的な要因となるため，上記の結果が得られる．

一方負荷が大きくなると，レジスタ挿入リングとRISリングのほうが遅延が小さくなる．トークンリングとスロットリングではパケットが必ずリングを一周するためトラヒックが多くなり，このためノードの端末バッファ内での待ち時間が大きくなる．レジスタ挿入リングとRISリングでは，パケットは宛先ノードでリングから除去されるため，トラヒックは他の２つほど多くならず，従って端末バッファ内での待ち時間も他の２つほど大きくならない．

7.1.2 端末バッファオーバフロー確率

図7.2に$M = 30$のときの$M\sigma$と端末バッファのオーバフロー確率の関係を示す．解析値とシミュレーション結果とはほぼ一致しており，解析により得られたオーバフロー確率の有効性が確かめられた．

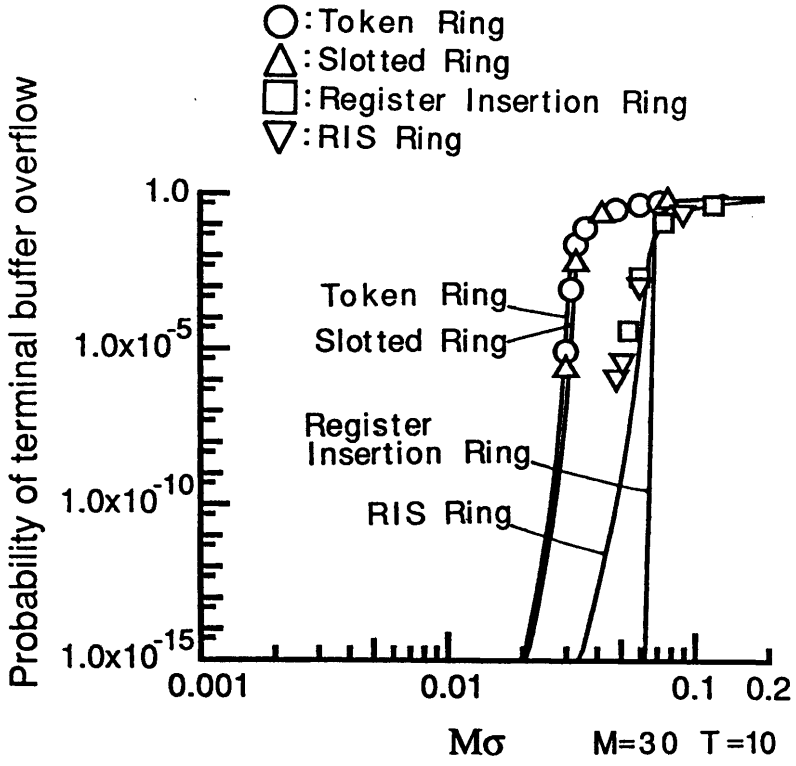


図7.2 端末バッファオーバフロー確率と$M\sigma$の関係($M = 30, T = 10$)
Fig.7.2 Probability of terminal buffer overflow versus $M\sigma$ for $M = 30$ and $T = 10$.

オーバフロー確率は，任意のタイミングでバッファからのパケット送信を開始できるレジスタ挿入リングが最も小さく，ついでRISリングが小さい．このRISリングの確率は，負荷が大きくなるとレジスタ挿入リングの確率とほぼ等しくなる．

7.2 性能比較と結語

ここでは，4つのプロトコルの解析結果を互いに比較し，与えられた条件下で最適なプロトコルを考える．

ノードの台数Mと各ノードで1秒間に発生するパケット数とを変えながら，平均パケット遅延と端末バッファオーバフロー確率を各プロトコルごとに計算する．そして，オーバフロー確率が1.0×10^{-9}以下であるプロトコルの中から，パケット遅延が最小のプロトコルを求め，これを最適なプロトコルとする．1.0×10^{-9}の値は，データ伝送で通常求められるパケットの許容破棄率である．なお，パケット長，Tの値等その他の条件は，7.1と同じとした．

その結果，各プロトコルが最適となる領域を図7.3に示す．図中“None”とあるのは，全てのプロトコルにおいて端末バッファオーバフロー確率が1.0×10^{-9}より大きくなる領域である．図より，トークンリングはMが小さくパケット発生も少ないときに優れていることが分る．スロットリングはMが大きくなりパケット発生が少ないときに優れている．これらのことは，7.1.1の議論からも明らかである．

パケット発生が多くなると，パケットが宛先ノードで取り除かれるレジスタ挿入リングとRISリングが有利になる．RISリングは，レジスタ挿入リングのパケット送信タイミングをスロット化することで処理をハードウェア化したものである．このため，パケットの送信は一定周期ごとに限られ，レジスタ挿入リングに比べて端末バッファ内での待ち時間が増える．一方，パケット転送に要するオーバヘッドがレジスタ挿入リングより小さい．このことから，Mが小さいときは端末バッファ内での待ち時間が少ないレジスタ挿入リングが最適となる．また，Mが大きいときは，パケット転送時の遅延が小さいRISリングが最適となる．

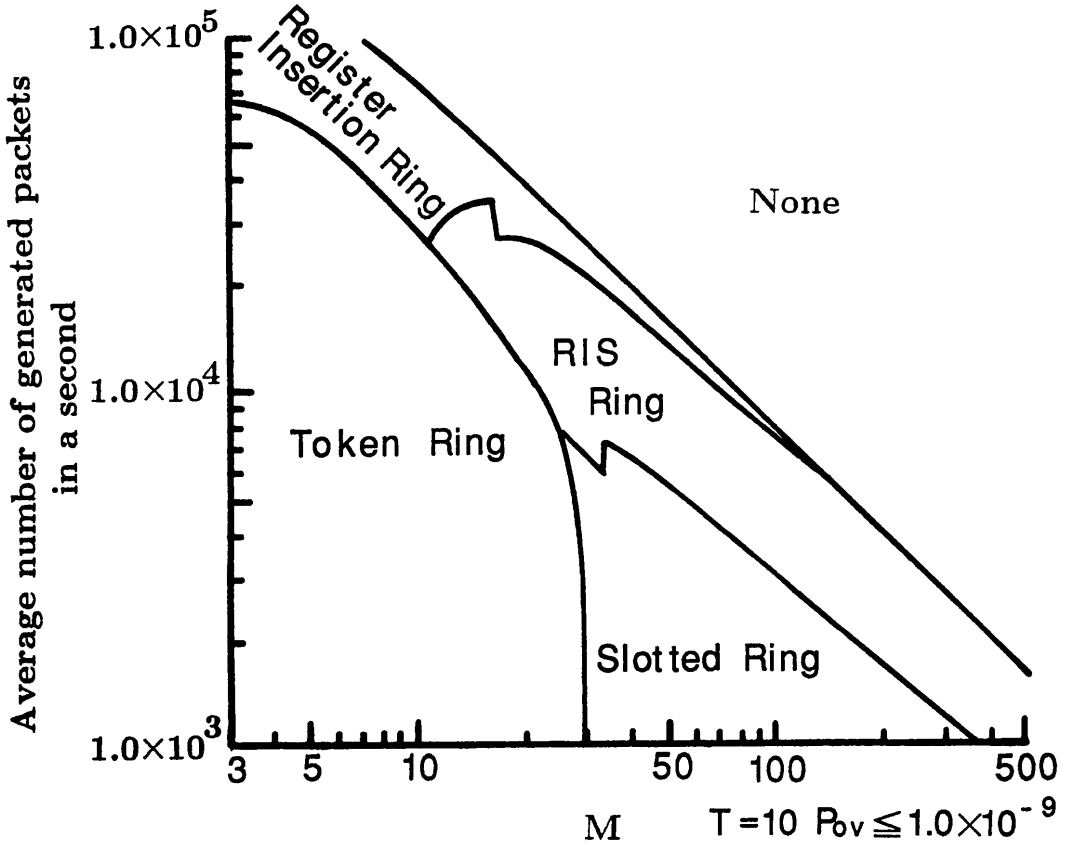


図7.3 トークンリング，スロットリング，レジスタ挿入リング，
RISリングの有効適用領域($T = 10, P_{ov} \leq 1.0 \times 10^{-9}$)

Fig.7.3 Suitable regions for token rings, slotted rings, register–insertion rings and RIS rings at $T = 10$ under the condition that $P_{ov} \leq 1.0 \times 10^{-9}$.

第3部
リング型ネットワークの信頼性向上

第8章 リング型ネットワークにおけるノードのハングアップ検出と復旧法

8.1 序

近年ネットワークが様々なところで敷設されるにつれ，その信頼性が重要な問題となってきている．これまでにも，ネットワークの信頼性を向上させ，障害が発生したときに，その障害を発見し，障害を回避したり復旧するための様々な研究が行われてきた．

例えば，リング状のネットワークを階層化して構築し，各々の階層間にバイパスを設けるもの[PIER72][KROP72][COKE72]，リング状のネットワークにおいてケーブルとノード内のモジュールを各々二重化し，さらにネットワークを監視するスーパバイザを階層化して設置するもの[EBIH85]，ループ状のネットワークにループバック機能を設け，ノードがパケットを送信した後，NAC信号を受け取るか，もしくは一定時間たってもACK信号を受信しない場合，別ルートで再送するもの[PAUL80]，専用の管理システムが定期的に各装置をポーリングして障害情報を収集し，ルートの障害を発見し，障害を回避する別ルートを設定するもの[NAKAM87]，バイパス機能を持った多重多段のリング状ネットワークにおける信頼性の評価の研究[EBIH88]等が行われている．

一方筆者は，信州大学内に，独自に開発したリング型キャンパスネットワークS-netを構築し，運用，改良，保守を行い，障害発生時には復旧，対策を行ってきた．これまでに発生した障害の記録を分析したところ，LANの様に狭い範囲にネットワークを構築した場合は，広範囲のネットワークと異なり，ケーブル等ルートに

障害が発生することはほとんどなく，障害の大半はノードのハングアップが原因であることが分った．

ここで，ハングアップとはシステムを構成する様々なハードウェアで発生する障害と定義する．このハングアップには，ドライバ・レシーバ用ICの障害の他，様々なハードウェアの障害があり，一方その原因には，ラインのノイズ，熱暴走，雷のサージ，ICのラッチアップ等さまざまなものがあった．S-netは，ハングアップ発生の都度その原因を追求して対策を施し，最近ではハングアップはほとんど発生していないが，発生するとネットワークに重大な影響を与えることから，万一発生したときにこれを早期に発見し復旧させることは重要である．このことは，ちょうどCPUの暴走対策に関し，暴走に対する様々な対策をとったうえで，それでも万一暴走したときのためにウォッチドッグタイマを用い暴走の検出と復旧を図ることと同様である．また，以後の対策のために，復旧後ハングアップ箇所を管理装置に報告する機能も必要である．

従来のネットワーク管理の方法には，ネットワーク全体を定期的に監視し，障害を発見し，復旧を指示する専用の管理装置をネットワークに接続するものがある．しかし，この方法はノードのハングアップの検出・復旧に関して次の点で問題がある．

(1) ループバック可能なようにラインの二重化がなされていなければ，ハングアップしているノードの特定はできない．

(2) ハングアップしているノードを発見しても，ハングアップがノードのデータ受信回路部で発生している場合，管理装置からの復旧の指示をこのノードは受信できず，従ってこのノードは復旧できない．

本論文においては，リング型ネットワークにおけるノードのハングアップを自動的に検出し，かつ自動的に復旧し，その後ハングアップ箇所を報告する手法について提案を行う．これは，リング型ネットワークの形状的な特性を利用して，各ノードが定期的にチェック用のパケットを送り合い，このパケットの受信状態から自己のハングアップの有無を検出するもので，検出後自動的に復旧させ，さらにチェック用パケットの情報をもとにハングアップ箇所を任意のノードに接続した装置に報告する機能をもつ．この方法は，リング型ネットワークに広く適用でき，付加するハードウェアが少なく，さらに検出・復旧・報告は各ノードにて行うため，専用の管理システムを必要としないという特長を持っている．

8.2ではS-netにおける障害について分析し，8.3においてリング型ネットワークのノードのモデル化を行う．8.4では，提案するハングアップ検出・復旧の手法について，さらに8.5でハングアップ箇所の報告の手法について述べ，8.6でその評価を行う．

8.2 S-netにおける障害

筆者は，キャンパスネットワークとして，2段に階層化した，レジスタ挿入リング方式を用いたリング型ネットワークS-netを提案し，信州大学の２つのキャンパスに構築し，運用をしてきた[FUWA87][FUWA88b]．このS-netには，450台以上のパーソナルコンピュータや汎用計算機を接続し，多くの利用者を得ている．筆者は，このS-netの運用にあたりその保守を行い，障害発生時にはその復旧と対策を行なうとともに，発生した障害を記録してきた．ここではこの記録をもとにS-netにおける障害の分析をおこなう．

障害には様々なものがあったが，それを原因別に分類してみると，表8.1の様になる．障害は1986年〜1988年の3年間で52件発生し，そのなかで原因として最も多いのはノードのハングアップであり，42件(81%)発生した．ハングアップにはドライバ・レシーバICや通信用LSIのハングアップの他，様々なハードウェアのハングアップがあった．ハングアップの原因には，ラインのノイズ，熱暴走，雷のサージ，ICのラッチアップ等さまざまなものがあり，発生時ごとに原因を追求し，その都度対策を施すことにより，最近ではほとんど発生していない．しかし，ハングアップが発生するとその影響は大きく，ネットワーク全体におよぶ場合もある．このため，ハングアップの有無を常に監視し，万一発生したときには早期に検出して復旧させることが重要である．

一方，復旧の方法であるが，ハングアップを起こした場合，リセット信号を与えることで復旧できる素子もあるが，ドライバ・レシーバやゲートがラッチアップした場合のように，復旧するには電源を入れ直すしか方法のない素子も多い．このため，ノードを確実に復旧させるためには，電源の入れ直しを考えなくてはならない．

従来障害の対策としては，ルートを二重化し，ノード等各機器も二重化し，障害発生時には別ルート，別の機器を使用するか，または障害発生のルート，機器を

表8.1 S-net障害の原因別分類(1986年10月～1988年10月)

原　　因	件数	割合
ノードのハングアップ	42	81%
ICの不良	5	9%
電源装置異常	2	4%
ソフトウェアのバグ	1	2%
基板の半田付け不良	2	4%
合　　計	52	100%

切り離す等の対策が行われているが，二重化は費用の問題があり，障害発生部分の切り離しは，その部分に接続された機器からはネットワークを利用できないという問題がある．

この分析に基づき，定期的にネットワークをチェックしてハングアップしているノードを発見し，そのノードの電源を自動的に入れ直す方法が，万一のハングアップによるネットワーク障害に対して有効であることがわかる．

また復旧後，今後の対策のためにハングアップ箇所を任意のノードに報告し，そのノードに接続した機器で記録をとれるようにすることも重要である．

8.3 リング型ネットワークにおける ノードのモデル化

ここでは，リング型ネットワークにおけるノードのハングアップ検出・復旧法を考えるにあたり，このノードのモデル化を行う．

図8.1にリング型ネットワークにおけるノードの構成を示す．ノードは，ラインから入力するパケットをRECEIVERモジュールが受信してヘッダの送り先アドレスを求め，このアドレスから自己のアドレスと一致すればパケットを取り込み，そのノードに接続された機器へ送る．またこのパケットを次のノードへ転送する場合は，そのパケットをTRANSLATOR，DRIVERの各モジュールを経て，再びラインへ送信する．またこのノードに接続されたコンピュータからのパケットは，ネットワークのアクセス方式に従い，DRIVERモジュールからラインへ送信する．これらの処理は，マイクロプロセッサ等CPUが制御して行う．

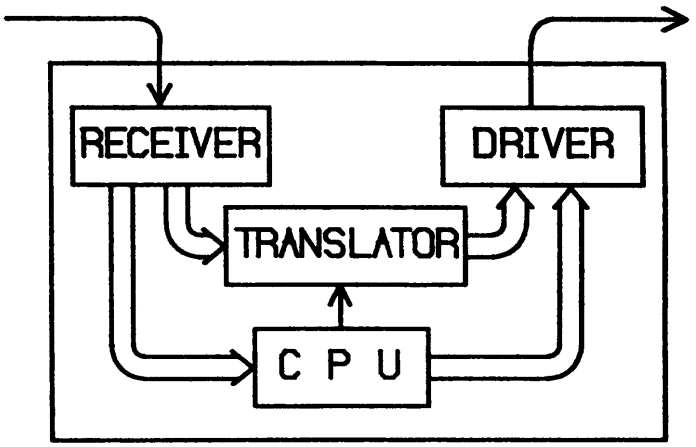


図8.1 ノードの構成
Fig.8.1 Block diagram of node.

このためリング型ネットワークにおけるノードは，制御を行うCPU，データの受信を行うRECEIVER，送信を行うDRIVER，パケットの転送を行うTRANSLATORの各モジュールに分けることができる．

8.4 提案する障害対策

8.3で示したノードのモデルに基づき，提案するリング型ネットワークのハングアップ検出・復旧の方法を説明する．

モデル化したノードにおいて，CPU，RECEIVER，DRIVER，TRANSLATORの各モジュールのハングアップが問題となるが，CPUのハングアップは従来からあるウォッチドッグタイマの手法により，対策を立てることができる．筆者は，CPUのハングアップを検出する能力が高いウォッチドッグタイマについての提案を行った[FUWA88a]．このことについては，第9章で論じる．そこで，本章ではCPUはハングアップしないものと仮定して，残りのRECEIVER，DRIVER，TRANSLATORの各モジュールのハングアップについて考える．

8.4.1 基本的な考え方

図8.2にリング型ネットワークのモデルを示す．ノードは図のようにn個あるとする(説明のために1 ～ nの番号を付ける)．このn個のノードにおいて，モデル化した3つの部分のいずれかがハングアップしたときに，これを検出して復旧させ，さらに報告する手法を提案する．

この手法は次の基本的な考えに基づいている．

(1) 各ノードは独立して，一定時間ごとにハングアップ検出のためのチェック用パケットをリングを一巡するように送る．

(2) 各ノードは，自分が出したチェック用パケットと，リングのパケットの流れる方向に対して一つ前のノードおよび一つ後のノードが出したチェック用パケットの受信状況とその内容から，自分のノード内の3つのモジュールのハングアップの有無を一定時間ごとに判定する．

(3) 自己内にハングアップしているモジュールは無いと判定したノードは，パルスを復旧回路へ出力する．復旧回路ではこのパルスの到着があらかじめ定めた時間以上遅れれば，電源を切り，ノードとは独立したタイマにより一定時間後に再立ち上げを行い，復旧させる．

(4) 復旧後，一つ前のノードと，一つ後のノードが出したチェック用パケットの内容から，ハングアップ箇所を調べ，その結果を任意のノードに接続した機器に送る．

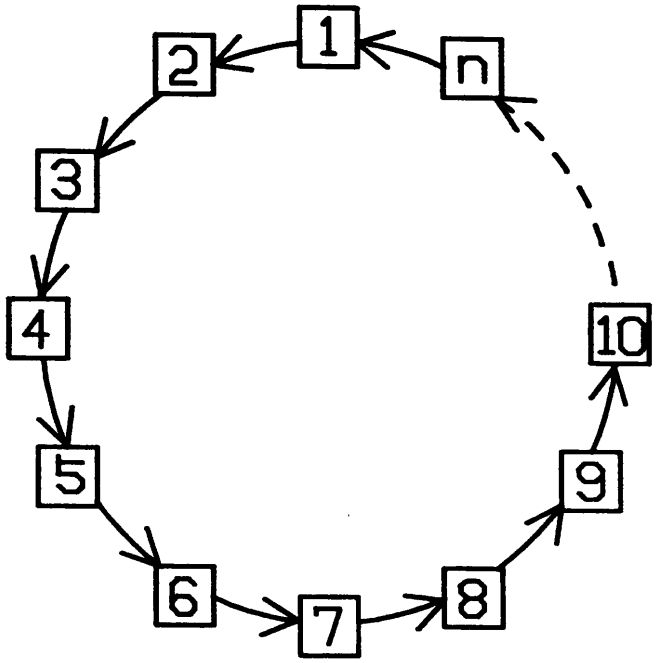


図8.2 リング型ネットワーク
Fig.8.2 Network of ring topology.

なお，報告機能については8.5で述べる．

また，この手法を考えるに際し，ハングアップに関する次の仮定を行った．

(1) ハングアップは同時に2箇所以上では発生しない．

(2) ノードが復旧のため電源を再立ち上げした後，T時間(8.4.2で述べるハングアップの有無を判定する間隔で，10秒程度)はどのノードにもハングアップは発生しない．

(3) ノードの台数は3台以上とする．

この仮定は，S-netにおけるハングアップの発生状況からも，手法の一般性を損うものではない．

8.4.2 タイミング

図8.3に各ノードのチェック用パケット送信と，ハングアップの有無の判定，復旧のための電源再立ち上げのタイミングを示す．判断はT時間ごとに行い，チェック用パケットは$T/3$時間ごとに出力する．ハングアップが無いと判定し，復旧回路にパルスを出力してからW_T時間($T < W_T < 4T/3$)以内に，再びハングアップが無いと判定してパルスを出力しないときは，復旧回路において電源をO_T時間($W_T + O_T < 4T/3$)切断し再び立ち上げる．

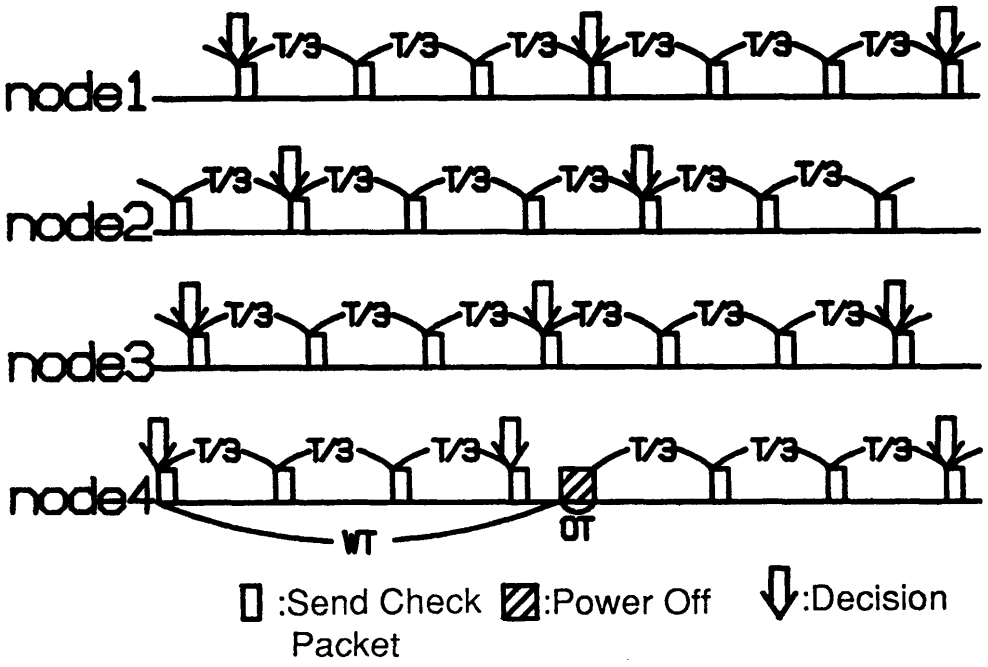


図8.3 チェック用パケット送信・判定・復旧タイミング
Fig.8.3 Timing chart of sending packets, diagnosing and restoration.

個々のノードでハングアップの有無の判定から次の判定までのT時間の間に，全てのノードが少なくとも1つはチェック用パケットを送信する必要がある．各ノード間では同期をとらず独立して動作しているため，この条件を満たすには，チェック用パケットの送信間隔は$T/2$以下である必要がある．しかし$T/2$では，任意のノードが電源を切っている間に別のノードがパケットを送信すると，このパケットは失われ，このノードからは次の判定までにチェック用パケットを1つも送れないことになる．仮定より，復旧後T時間はハングアップが発生しないため，パケット送信の間隔を$T/3$にして，T時間に3パケット送信すれば，$W_T + O_T < 4T/3$であること

からこの問題は解決できる.

8.4.3 チェック用パケット

チェック用パケットは，各ノードにて組立てられ送信される．このパケットは，各ノードを次々と転送され，その都度各ノードにて情報の読み出し，書き込みを行い，最後にパケットを送信したノードに戻る(もちろん途中にハングアップしたモジュールがあれば，パケットは失われる)．パケットの送信転送方式は，ネットワークにより次の２種類に分れる．

(1) 各ノードは，自分宛ではないパケットでも，その内容を読んだり書き込んだりできるネットワークの場合.

この場合は，各ノードは宛先を自分宛にしたチェック用パケットを送信すればよい.

(2) 各ノードは，自分宛ではないパケットの内容を読み書きできないネットワークの場合.

この場合は，ノードは宛先を一つ先のノードにしたチェック用パケットを送信し，各ノードでこのパケットを受け取ると必要なデータ処理の後，宛先をまた一つ先に書き換えてバケツリレー式に次々最初のノードまで転送する.

チェック用のパケットの構成は，各々リング型ネットワークのパケットの構成に従えばよい．必要なことは，このパケットがチェック用であることを示し，さらに次の$S_0 \sim S_3$の各1bitの情報を含むことである.

S_0 :送信，転送時に各ノードでIFLの値が1であれば，このビットを1にする.

S_1 :送信時，ノードの$Stable$の値が1011であれば1．そうでなければ0.

S_2 :送信時，ノードの$Stable$の値が1000であれば1．そうでなければ0.

S_3 :送信時，ノードの$Stable(1)$の値が1であれば1．そうでなければ0.

(IFL,$Stable$については8.4.4で述べる)

S_1，S_2については，一度1になると$Stable$の値が各々1011，1000以外になってもあと2パケットの間，1を保つ.

8.4.4 ノード内変数

各ノードは，ハングアップの有無の判定，復旧後の報告のために，以下の変数を持つ．

$Stable(0 \sim 3)$

(0) :一つ前のノードが出したチェック用パケットを受信したとき，1にする．

(1) :自分が出したチェック用パケットを受信したとき，1にする．

(2) :一つ後のノードが出したチェック用パケットを受信したとき，1にする．

(3) :一つ後のノードが出したチェック用パケットを受信したとき，そのパケットのS_3の値をセットする．

$Itable(0 \sim 3)$

(0) :一つ前のノードのDRIVERがハングアップしていたことを確認したら，1にする(確認方法については8.4.6で説明する)．

(1) :一つ前のノードが出したチェック用パケットを受信したとき，そのパケットのS_1の値をセットする．

(2) :一つ後のノードが出したチェック用パケットを受信したとき，そのパケットのS_2の値をセットする．

(3) :一つ前，自分自身，一つ後のノードが出したパケットを受信した際，各パケットのS_0の値が1のとき，1にセットする．

C_1,C_2:カウンタとして使用する(8.4.6で説明)．

IFL:電源投入後最初のT時間の間1．それ以外は0．

各変数の電源投入時の初期値は，$Stable(1)$とIFLが1で，他は全て0である．

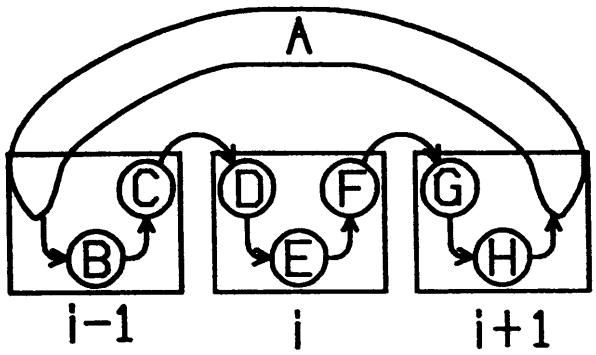


図8.4 ネットワークのモデル
Fig.8.4 Model of network.

8.4.5 ハングアップ時の*Stable*の値

リング型ネットワークにおいて，あるノード内のあるモジュールがハングアップしたときに，図8.2における$i(1 \leq i \leq n)$番目のノード(以後このノードを"ノードi"と呼ぶ)の*Stable*の値がどのようになるのかを，図8.4を用いて考える．

この図において，B，Cは各々ノード$i-1$($i=1$のときはn)のTRANSLATOR，DRIVERの各モジュールであり，D，E，Fは各々ノードiのRECEIVER，TRANSLATOR，DRIVERモジュール，同様にG，Hは各々ノード$i+1$($i=n$のときは1)のRECEIVER，TRANSLATORモジュールである．また，Aはノード$i-1$のRECEIVERモジュールとノード$i+1$のDRIVERモジュール，及び他のノードの各モジュール全体を示す．ここで，A～Hが各々ハングアップした場合にノードi内の*Stable*の値がどうなるか考えるが，Aがハングアップしたというのは，Aに含まれるモジュールの内，少なくとも1つがハングアップしたことを表すこととする．

表8.2にA～Hが各々ハングアップした場合の，*Stable*の値を示す．

(1)Aがハングアップした場合

ノードiはノード$i-1$が出すパケットを受信することはできるが($Stable(0)=1$)，自分自身が出すパケット及びノード$i+1$が出すパケットは受信できない($Stable(1 \sim 2)=0$)．また，ノード$i+1$が出すパケットを受信できないため，$Stable(3)$の値は初期値0のままである．

表8.2 A~Hがハングアップしたときのノードiの$Stable$の値

ハングアップ箇所	Stable (0)	Stable (1)	Stable (2)	Stable (3)
A	1	0	0	0
B	1	0	0	0
C	0	0	0	0
D	0	0	0	0
E	1	1	1	0
F	1	0	1	0
G	1	0	1	0
H	1	0	1	1
正　常	1	1	1	1

(2)Bがハングアップした場合

ノード$i-1$が出すパケットはBを経由せずCから出力されるため，ノードiは受信できるが($Stable(0)=1$)，ノードiとノード$i+1$が出すパケットはBを通過できず，ノードiは受信できない($Stable(1\sim 3)=0$).

(3)Cがハングアップした場合

ノード$i-1$は全くパケットを送信できず，ノードiはノード$i-1$，ノードi，ノード$i+1$が出すパケットを全て受信できない($Stable(0\sim 3)=0$).

(4)Dがハングアップした場合

ノードiはどのノードが出すパケットも受信できない($Stable(0\sim 3)=0$).

(5)Eがハングアップした場合

ノードiはパケットの転送は行えないが，パケットの受信，パケット送信は行える．このため，ノードiはノード$i-1$，ノードi，ノード$i+1$からのパケットは全て受信できる($Stable(0 \sim 2) = 1$)．$Stable(3)$の値は，ノード$i+1$内の$Stable(1)$の値であり，ノード$i+1$におけるEモジュールはノードiにおけるBモジュールに相当するため，表中B行の$Stable(1)$の値から0になる．

(6)Fがハングアップした場合

ノードiはパケットの送信ができないため，自己のパケットは受信できないが($Stable(1) = 0$)，ノード$i-1$，ノード$i+1$が出すパケットは受信できる($Stable(0) = 1$,$State(2) = 1$)．$Stable(3)$の値は，表中C行の$Stable(1)$の値から0になる．

(7)Gがハングアップした場合

ノードiは，自己が出したパケットは受信できないが($Stable(1) = 0$)，ノード$i-1$，ノード$i+1$が出すパケットは受信できる($Stable(0) = 1$,$State(2) = 1$)．$Stable(3)$の値は，表中D行の$Stable(1)$の値から0になる．

(8)Hがハングアップした場合

ノードiは自己が出すパケットは受信できないが($Stable(1) = 0$)，ノード$i-1$，ノード$i+1$が出すパケットは受信できる($Stable(0) = 1$,$Statel(2) = 1$)．$Stable(3)$の値は，表中E行の$Stable(1)$の値から1になる．

(9)ハングアップがない場合

ノードiは全てのノードが出すパケットを受信できる($Stable(0 \sim 2) = 1$)．また，ノード$i+1$も同様であるため,$Stable(3)$の値は1となる．

8.4.6 判定

表8.2に基づき提案する判定のアルゴリズムを，図8.5に示す．8.4.1で述べたように，提案する手法は各ノードが自己内のモジュールのハングアップを判定するものであり，表8.2中ノードiが判定しなくてはならないのは，D～Fのハングアップである．そこで，判定時に$Stable$の値から，このD～Fを判定することを考える．なお，図8.5中$Stable = 1000$のときの処理は，同時に2つ以上のモジュールがハングアップしたときでも，復旧が行えるようにするためのもので，このことについては付録Cで述べる．

```
switch(Stable[0]<<3+Stable[1]<<2+Stable[2]<<1+Stable[3])
{
case 0xf:          /* Stable = 1111 (all NODEs are correct) */
  if(C1==1)
    Itable[0]=1;       /* set Itable(Hang-Up occurs at NODE-C)*/
  /******** Send a report packet(Hang-Up occurred) ********/
  if(Itable[0]==1)
    report("Hang-Up occurred at DRIVER in NODE(i-1)");
  else if(Itable[0]==0 && Itable[1]==1 && IFL==1)
    report("Hang-Up occurred at TRANSLATOR in NODE(i)");
  else if(Itable[0]==0 && Itable[1]==0 &&
        Itable[2]==1 && Stable[2]==1 && IFL==1)
    report("Hang-Up occurred at RECEIVER in NODE(i)");
  for(j=0;j<=3;j++) Itable[j]=0;                  /* clear Itable */
  C1=0; C2=0;                                    /* clear counter */
  break;
case 0x0:          /* Stable = 0000
                      (Hang-Up may occur at NODE-C or -D) */
  if(C1==0) send_pulse();              /* send a pulse to WDT */
  C1++; C2=0;
  Itable[3]=0;
  break;
case 0xe:          /* Stable = 1110
                      (Hang-Up may occur at NODE-E) */
  if(!(IFL==0 && Itable[3]==0))
    send_pulse();                          /* send a pulse to WDT */
  if(C1==1)
    Itable[0]=1;       /* set Itable(Hang-Up occurs at NODE-C)*/
  Itable[3]=0;
  C1=0;
  break;
case 0xa:          /* Stable = 1010
                      (Hang-Up may occur at NODE-F or -G) */
  if(Itable[3]!=0)
    send_pulse();                          /* send a pulse to WDT */
  if(C1==1)
    Itable[0]=1;       /* set Itable(Hang-Up occurs at NODE-C)*/
  Itable[3]=0;
  C1=0;
  break;
case 0x8;          /* Stable = 1000 (Hang-Up may occur
                          at several NODEs simultaneously) */
  if(C2==0) send_pulse();              /* send a pulse to WDT */
  C2++;
  if(C1==1)
    Itable[0]=1;       /* set Itable(Hang-Up occurs at NODE-C)*/
  C1=0;
  Itable[3]=0;
  break;
default:         /* else */
  if(C1==1)
    Itable[0]=1;       /* set Itable(Hang-Up occurs at NODE-C)*/
  C1=0; C2=0;
  Itable[3]=0;
  break;
}
for(j=0;j<=3;j++) Stable[j]=0;                    /* clear Stable */
IFL=0;
```

図8.5 判定・報告処理
Fig.8.5 Procedure of diagnosing and reporting.

(1)$Stable = 1110$のとき

このときは，表8.2よりE(自己内のTRANSLATOR)がハングアップしていると判定できる．

但し，$Itable(3) = 1$のときは無効とし，ハングアップは無いと判定する．これは，ネットワークにおいていずれかのノードが復旧のため電源を再立ち上げしたときは，$Stable$のあるビットは立ち上げ前，あるビットは立ち上げ後と$Stable$の内容にくい違いが生じる可能性があるためである．$Stable$のビットを構成するノード$i-1$，ノードi，ノード$i+1$のパケットのS_0の値が１つでも1であれば，$Itable(3)$の値は1となり，このくい違いの可能性を示す．

また，$IFL = 1$のときも無効とした．これは，立ち上げ後T時間後までは自己のパケットを受信しないときも$Stable(1)$の値は初期値の1となっており，$Stable$の値が1110となることがあるためである(このことについては，(2)でさらに説明する)．

(2)$Stable = 1010$のとき

このときは，表8.2よりFかGがハングアップしていることを示す．この２つはノードiのDRIVERとノード$i+1$のRECEIVERにあたり，特別なハードウェアの付加なしにはこの２つの判別は不可能である．このため，とりあえずDRIVERがハングアップしていると判定し，再立ち上げを行う．この場合も$Itable(3) = 1$のときは無効とする．

また，実際にはGがハングアップしていたときは，立ち上げ後もこの状態は変らず$Stable$の値は再び1010となるが，立ち上げ後T時間までは$Stable(1)$ の値は初期値の1であり，$Stable$は1110となる．このためEがハングアップしていると誤って判断するため，$IFL = 1$のときは$Stable$の値が1110 でも無効とする．

(3)$Stable = 0000$のとき

このときは，表8.2からCかDがハングアップしていることを示している．これは，ノード$i-1$のDRIVERとノードiのRECEIVERとにあたり，これも特別なハードウェアを用いない判別は不可能である．

しかしこの場合，ノード$i-1$では$Stable = 1010$となり，前述の電源再立ち上げを行っている．このため，ノードiでは，とりあえずハングアップは無いと判定し，T時間待ち(この待ちに変数C_1を用い，待っているときに$C_1 = 1$にする)，次の判定時に続けて$Stable$の値が0000となったとき，Dがハングアップしていると判定する．また，$C_1 = 1$のとき$Stable$ の値が0000以外になった場合は，ノード$i-1$のDRIVERがハングアップしていたことになるため，$Itable(0) = 1$にする．

8.4.7 復旧

図8.6に復旧のための回路を示す．この回路は，ウォッチドッグタイマの回路と似た構造であり，ノードとは独立して動作するクロックによりカウントアップするカウンタと，このカウンタの桁あふれ信号により動作するタイマ付リレーとからなる．ノードにおいてT時間ごとに行う判定時にハングアップが無いと判定した場合は，ポートからカウンタにリセットパルスを送り，カウンタをクリアする．このパルスがW_T時間以上来ないとカウンタは桁あふれを起こし，この桁あふれ信号によりタイマ付リレーが働き電源を切り，O_T時間後再立ち上げを行う．タイマ付リレーは様々なものが市販されている．

この回路の特長は，たとえノードのポートがハングアップしても，復旧作業が行われる点であり，フェイルセーフの構成になっている．またこの回路自身にも自己検査性を持たせることにより，より信頼性を上げることができる[NAMI85]．

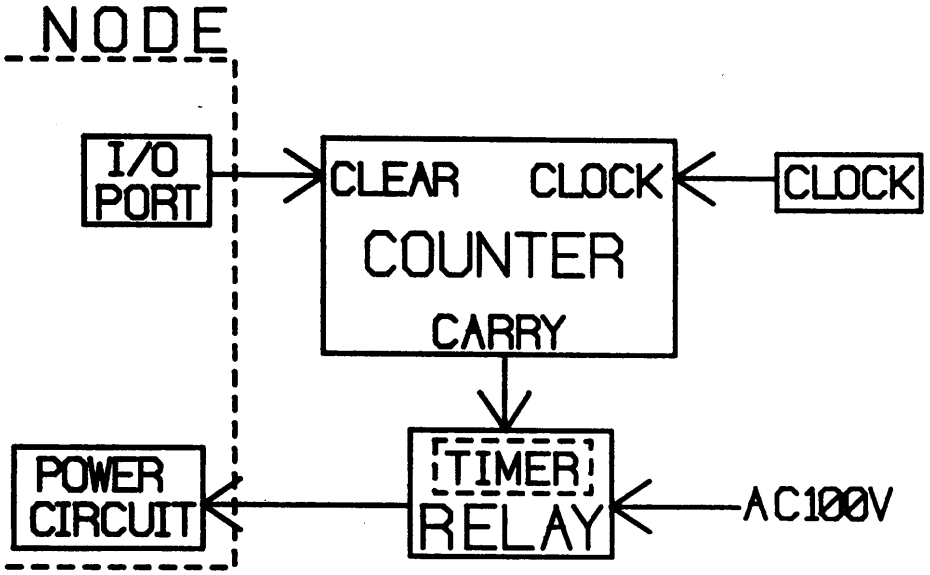


図8.6 復旧回路
Fig.8.6 Restoration circuit.

8.5 報告機能

報告はハングアップ復旧後にどのノードのどのモジュールでハングアップが発生したのかを，任意のノードに接続した機器にパケットにして知らせる機能である．報告は，電源を切り再立ち上げ後，全てのモジュールが正常になったときにのみ行えるため，電源を切る前の状態は消え，これを用いることはできない．CMOSのメモリとバッテリを用い，電源を切る前の状態を保存する方法もあるが，回路の付加が伴い，この回路自身の信頼性を考慮した新たな対策が必要になることから問題がある．ここでは，電源立ち上げ後の$Stable$の値と，$Itable$の値を用いることにし，図8.5において$Stable$の値が1111のときに$Itable$の値を用いて報告を行う．

まず，$Itable(0)=1$のときは8.4.6にて示したとおり，ノード$i-1$のDRIVERがハングアップしていたことを示しており，この報告を行う．

次に，$IFL=1$(つまり，自己内にハングアップのおそれがあり，電源再立ち上げを行った直後)であり，かつ$Itable(1)=1$の場合について述べる．$Itable(1)=1$ということはノード$i-1$から$S_1=1$のパケットを受信したことを示している．これはノード$i-1$の$Stable$の値が1011であったことを示している(S_1は8.4.3で示した

ように一度1になると$Stable$の値が1011 以外になっても2パケットの間は1を保つ). ノードの$Stable$の値が1011 ということは，表8.2からHがハングアップしていることを示し，ノード$i-1$ から見たHはノードiのTRANSLATORにあたるため，この報告を行う.

$IFL=1$で$Itable(1)=0$かつ$Itable(2)=1$の場合は，ノード$i-1$から$S_1=0$のパケットを受信し，ノード$i+1$から$S_2=1$のパケットを受信したことを表す. これはノード$i-1$の$Stable$の値が1011ではなく，ノード$i+1$の$Stable$の値が1000であったことを示す. ノードの$Stable$ の値が1000ということは，表8.2からノード$i+1$から見てAかBがハングアップしたことを示し，このことからノードiのRECEIVERかTRANSLATORがハングアップしたことを示している(8.4.6のアルゴリズムから，ノード$i+1$ から見てAに属するモジュールでノードiのRECEIVER以外のモジュールのハングアップでノードiが電源の再立ち上げをすることはない). 一方ノード$i-1$の$Stable$の値は，ノードiのTRANSLATORはハングアップしていないことを示しているため，この場合はノードiのRECEIVERがハングアップしていたことになり，この報告を行う.

8.6 評価

提案した方法により，ハングアップが発生してから平均何秒後に復旧されるかをシミュレーションを用いて調べた. シミュレーションは，ノード数を6，W_Tを$T+1$秒，O_Tを2秒，各モジュールのハングアップの発生確率を二項分布として，Tの時間を様々に変えて行った. この結果を図8.7に示す. Tを10秒とした場合，ハングアップ発生から回復までの平均時間は19.4秒であった(これには電源を切っている時間2秒を含んでいる).

リングに接続したノード数をnとし，このときチェック用パケットが1秒間に何個必要であるか，各ノードが行う送信／受信回数は何回かを検討する. ここでパケット数とは，最初にノードが生成するパケットの数を意味し，各ノードが他のノード宛のパケットを受信したときに次のノードへ転送するパケットは数に含めないこととする. チェック用パケットは，各ノードが独立に$T/3$secごとに1個送信するため，ネットワーク全体の1secあたりのチェック用パケット数N_cは，

$$N_c=\frac{3n}{T} \qquad (8-1)$$

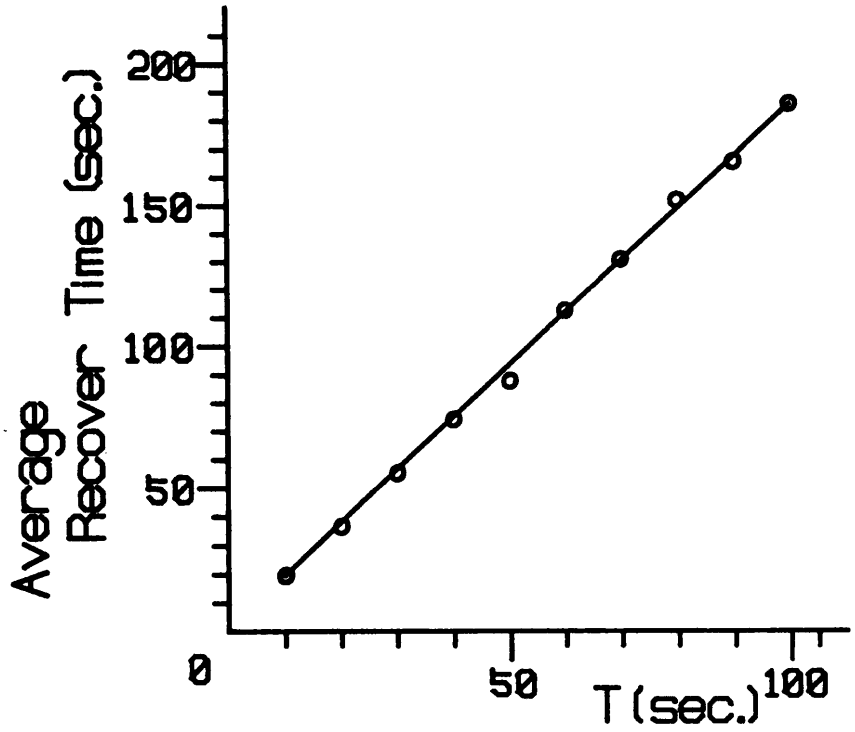


図8.7 チェック間隔と平均復旧時間の関係
Fig.8.7 Interval of diagnosing versus average recover time.

となる．またこのパケットは，生成したノードが送信した後，各ノードで受信・送信を繰り返しながら最初のノードに受信されるまでリングを一巡するため，結局各ノードが送信／受信するチェック用パケットの数は，送信・受信とも式(8-1)のネットワーク全体のチェック用パケットの数と等しい．このことから，例えばノードの数が100と比較的大規模なリングの場合，チェック用パケットの数と各ノードの送信／受信パケット数は$T = 10$secのときで，30個/secとなる．

このことを用い，S-netにおけるチェック用パケットの数と，各ノードの送信／受信数を検討する．S-netは，図8.8に示すように複数のCLUSTER RINGと，BACKBONE RINGとを階層化した2重のリング構成になっている．ここで，CM-1はネットワーク全体のコントローラであり，MCC IIはBACKBONE RINGとCLUSTER RING とを接続する機器である．BACKBONE RINGには最大4台のMCC IIが接続でき，1台のMCC IIには8個のCLUSTER RINGを接続する．CLUSTER RINGにはターミナルサーバであるMCCを最大4台接続でき，各MCCには8台の端末を接続できる．

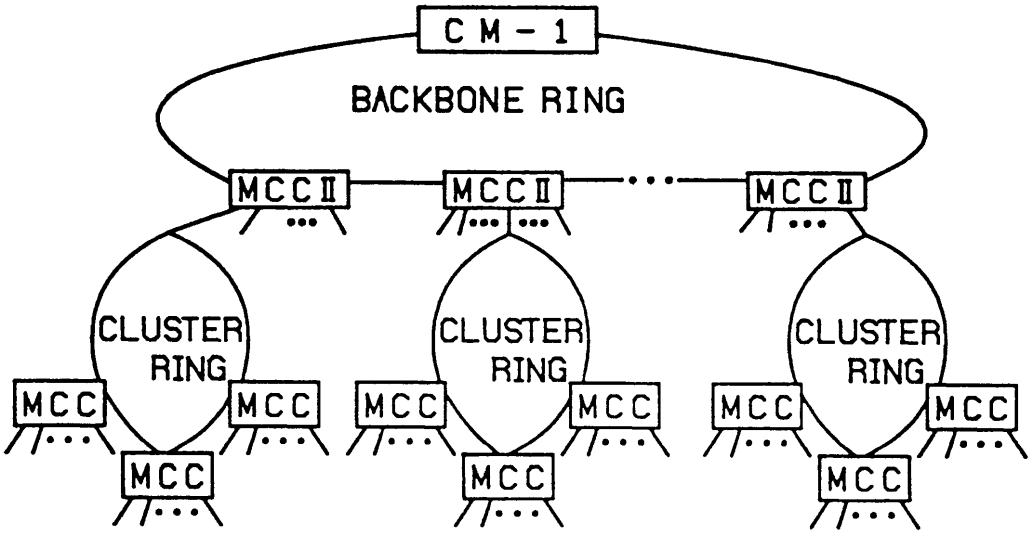


図8.8 S-netのブロック図
Fig.8.8 Block diagram of S-net.

このS-netにおいては，各リングごとに独立して本提案の手法を用い，CM-1，MCC II，MCCのハングアップを検出する．BACKBONE RINGでは，本提案でのノードにあたるのはMCC IIとCM-1(チェック用パケットに関しては，CM-1も1台のノードとして動作する)であり，MCC IIの台数を$n(1 \leq n \leq 4)$ とするとノードの数は$n+1$となり，式(8-1)よりBACKBONE RINGでの1secあたりのチェック用パケット数N_{c1}は，

$$N_{c1} = \frac{3(n+1)}{T} \qquad (8-2)$$

となる．CLUSTER RINGでは，ノードにあたるのは，MCCとMCC IIである(MCC IIは，チェック用パケットに関しては接続した各CLUSTER RINGに対してそれぞれ1台のノードとして動作し，これによりMCC IIの各CLUSTER RING用のインタフェースのハングアップを検出する)．このため，CLUSTER RINGに接続したMCCの平均台数をm台$(1 \leq m \leq 4)$とすると，CLUSTER RINGでの1secあたりのチェック用パケットの数N_{c2}は，

$$N_{c2} = \frac{3(m+1)}{T} \qquad (8-3)$$

となる．CM-1とMCCが送信／受信するチェック用パケットの数は，各々N_{c1}，N_{c2}と等しい．MCC IIには1つのBACKBONE RINGと8つのCLUSTER RINGを接続

するため，MCC IIが送信／受信する1secあたりのチェック用パケットの数NはN_{c1}，N_{c2}から，

$$M = \frac{3(n+1)}{T} + 8\frac{3(m+1)}{T} = \frac{3n+24m+27}{T} \qquad (8-4)$$

となる．現実のS-netでは，$n=3$，$m=2$程度であり(現在総チャネル数240)，このことから，$T=10$secのときBACKBONE RINGにおけるチェック用パケット数とCM-1の送信／受信数は1.2個／sec，CLUSTER RINGにおけるチェック用パケットの数とMCCの送信／受信数は0.9個／sec，MCC IIの送信／受信数は8.4個／secとなる．

ネットワークが高負荷で動作しているときに本方法による復旧時間がどう変動するかを考える．本方法では$T/3$時間($T=10$秒の時は3秒程度) でパケットがリングを一巡すれば，復旧時間は変動しない．これは使用するネットワークのスループットと伝送遅延との関係によるが，例えばS-netの場合について，シミュレーションにより検討した．シミュレーションでは，4台のノード(各ノードは8個のチャネルを持ち，総チャネル数は32)を200m間隔でリング状に接続し，各チャネルからパケットがポアソン分布に従う到着確率で入力し，これをリングへ送信するものとした．伝送速度は1$Mbps$，伝送路の信号伝搬遅延は5$\mu s/km$，ノード内でのパケットの遅延は10bitとし，パケットがリングを一巡するのにかかる平均伝送時間とスループットとの関係を求めた．この結果，スループットが0.9と高負荷時でも伝送時間は0.425msec (パケット長で正規化して8.49)となり，高負荷時でも復旧時間に影響がないことがわかる．

また，復旧のために電源を切っているときと再立ち上げをしたときの影響であるが，電源を切っているときは，パケットはそのノードを通過できないが，これは切る前もハングアップの影響で同様にパケットは通過できないため，問題はない．一方，再立ち上げ時にラインに発生するノイズは，データリンク層により処理を行い，誤ってパケットとして受信しないようにする．

また，本方法ではハングアップが同時に2箇所以上では発生しないとの仮定を行っているが，万一2箇所以上のモジュールが同時にハングアップした場合でも，このアルゴリズムにより動作が不安定にならず，復旧が行われることを，付録Cにて明らかにする．

本方法には，次の特長がある．

(1) 特定のプロトコルに依存していないため，リング型のネットワーク一般に適用できる．

(2) 本方式のために必要とするハードウェアは，簡単な復旧回路のみである．

(3) 検出は各ノードが個々に行うため，リング全体を管理する特別な管理装置を必要としない．

8.7 結語

リング型ネットワークにおけるノードのハングアップを自動的に検出し，復旧させ，その後ハングアップ箇所の報告を行う手法を提案した．この方法は，リング型ネットワークの形状的な特徴を利用して，各ノードが定期的にチェック用パケットを送り，この受信状況から自己内のハングアップを検出するもので，必要なハードウェアの付加が少なく，特別な管理装置を必要としない等の特長を持つ．また，本方法は必要とするパケットの通信量も少なく，通常のパケット交換への影響も殆どない．さらに，同時に複数のハングアップが発生しても誤動作しない．

本方法を用いることにより，万一ノードにハングアップが発生した場合に，これを自動的に発見し復旧することができるため，ハングアップによるネットワーク障害の早期回復を図ることができる．

付録C 2箇所以上のモジュールが同時にハングアップしたときの安定性

本論文では，ハングアップは同時に2箇所以上では発生しないと仮定して，ハングアップの判定・復旧・報告のアルゴリズムを提案した．この仮定は，ネットワークにおける実際のハングアップの発生状況からみても，提案するアルゴリズムの一般性を損うものではないが，ここでは万一2箇所以上のモジュールが同時にハングアップした場合でも，このアルゴリズムにより動作が不安定にならず，復旧が行われることを，以下に明らかにする(但し，報告は正しく行えない)．

本文中図8.4のモデルにおいて，ノードiの*Stable*の値を，A～Hのモジュールが2箇所以上同時にハングアップした場合の全ての組合せについて調べ，各場合の*Stable*の値が，表8.2のどのモジュールがハングアップしている場合と同じになるかを求めた．

その結果，2箇所以上でハングアップが起きたときは，自己のノード内のRECEIVERモジュールであるEからパケットの流れと逆の方向にモジュールをたどったとき，ハングアップしているモジュールの内，Hを除いて最初にあるモジュールのみがハングアップしているときと*Stable*の値が同じになることが分った．2箇所以上のモジュールがハングアップし，パケットの流れと逆にたどり最初にあるハングアップしているモジュールがHのときは，*Stable*の値はFかGがハングアップしているときと同じになった．

例えば，Dがハングアップしていれば他のどのモジュールがハングアップしていても，*Stable*の値はDのみがハングアップしているときと同じになる．同様に，例えばBとHとGが同時にハングアップしても，*Stable*の値はBのみがハングアップしているときと同じになる．

ノードiが判定・復旧しなければならないのはD，E，Fのモジュールについてである．そこで以下において，D，E，Fを含んで2つ以上のモジュールが同時にハングアップする全ての組合せについて，いずれの場合も復旧が行われることを明らかにする．

[1]Dを含む2つ以上のモジュールがハングアップしたとき

このときは，常にDがハングアップしていると判定し，ノードiは復旧処理を行う．

[2]Fがハングアップし，さらにD以外の１つ以上のモジュールが同時にハングアップしたとき

[2–1]ハングアップしたモジュールにCが含まれている場合

CもしくはDがハングアップしていると判定し，この判定がもう一度あれば復旧処理を行う．一方ノード$i-1$では，Fのハングアップの影響で*Stable*の値は1000となるか，もしくはノード$i-1$のRECEIVERかノード$i-2$のノードのDRIVERのモジュールがハングアップしていれば*Stable*の値は0000となり，いづれにしてもノード$i-1$は直ちに復旧処理は行わないため，ノードiは次の判定時に復旧処理を行う．

[2–2]Cは正常で，ハングアップしたモジュールにAもしくはBが含まれている場合

この場合*Stable*の値は1000となり，AもしくはBがハングアップしていると判定する．このため例えば互いに間に1つのノードを持って接続した2つのノードのDRIVERが同時にハングアップすると，この2つのノードの*Stable* の値はどちらも1000となる．図8.5に示したように，1000のパターンが2回続けば復旧処理を行うようにして，この場合でも復旧できるようにした．

[2–3]A,B,Cは正常で，ハングアップしたモジュールにHが含まれている場合

ノードiは，FもしくはGがハングアップしていると判定し，復旧処理を行う．

[2–4]A,B,C,Hは正常で，ハングアップしたモジュールにGが含まれている場合

ノードiは，FもしくはGがハングアップしていると判定し，復旧処理を行う．

[2–5]Fの他にEがハングアップしている場合

ノードiは，Fがハングアップしているとして，復旧処理を行う．

[3]Eがハングアップし，さらにD,F以外の1つ以上のモジュールが
ハングアップしたとき

[3–1]ハングアップしたモジュールにCが含まれる場合

ノードiはCまたはDがハングアップしていると判定する．ノード$i-1$は，[2]で検討したようにこの判定時か，次の判定時に復旧処理を行う．ノード$i-1$が次の判定時に復旧処理を行うときは，ノードiは次の判定時に復旧処理を行う．一方ノード$i-1$がこの判定時に復旧処理を行えば，ノードiはさらにF以外にハングアップしているモジュールがない場合，Eのハングアップを判定して復旧処理を行い，F以外にまだハングアップしているモジュールがある場合，[3–2]，[3–3]もしくは[3–4]に従う．

[3–2]Cは正常で，ハングアップしたモジュールにAもしくは
Bが含まれる場合

ノードiは，AもしくはBがハングアップしていると判定し，[2–2]同様この判定が2回連続したときに，復旧処理を行う．

[3–3]A,B,Cは正常で，ハングアップしたモジュールに
Hが含まれる場合

ノードiは，FもしくはGがハングアップしていると判定し，復旧処理を行う．

[3–4]Fの他にGがハングアップしているとき

ノードiは，FもしくはGがハングアップしていると判定し，復旧処理を行う．

第9章 ウォッチドッグタイマの有効性に関する統計的考察

9.1 序

コンピュータ搭載車に代表されるような，人命にかかわるシステムでは，何重にもフェイルセーフ機能が働くよう設計されなければならない．また，最近重要性が増しているコンピュータネットワークにおいても，ネットワークが異常となるとその影響は大きく，異常を早期に発見して復旧させるシステムが必要となる．このため，ラインのノイズや雷のサージ等様々な原因によるCPUの暴走を検出し，復旧させることが重要である．

CPUの暴走を検出する手段として，ウォッチドッグタイマ[RAMA74]がよく知られている(以下WDTと略す)．これは，基本的にはプログラムループ中に挿入したWDT用の信号出力命令によるパルスが一定時間以内に出力されるかを調べるものである．この出力が時間内に無い場合，CPUが暴走しているとして，必要な処置をする．自動車に搭載されているコンピュータシステムでも，このWDTによりCPUの暴走を検出し，暴走状態から正常動作に復帰しようとしている．

このWDTは，様々なコンピュータシステムにおいて用いられており，数多くの研究が行われ，多くの特許が毎年出願されている．さらに，暴走検出時のロールバックに関する研究[UPAD86]や，WDTのハードウェアの信頼性を向上されるため，順序回路を用いた自己検査性WDTの研究もなされている[NAMI85]．

本論文では，このWDTの暴走検出能力の限界について統計的に考察を行い，さらにより暴走検出能力の高いWDTの提案を行う．WDTの統計的考察に際しては，WDTを品質管理で用いられている抜き取り検査の一種と見なし，解析した．また，CPUが暴走した際の振舞いについて，実際のCPUを元にシミュレーションを行い，提案するWDTの暴走検出能力を検討した．この結果，本論文で提案するWDTは，従来のWDTに比べ約500倍程度の検出能力があることがわかった．

なお，本論文では，CPUが暴走した場合は，CPUのプログラムカウンタ値は，プログラム内の値を一様にとると仮定する．この仮定は，車搭載用CPU基板やネットワーク用システムにおけるCPUの暴走原因として考えられるラインのノイズと

外部の放電ノイズによる暴走に対しては，妥当性があることを，付録Dで示す．

9.2 CPU暴走の種類と原因

本章では，CPUの暴走とはプログラマが考えたものと異なる動作をCPUが行うことと定義する．ここでは，WDTが検出すべきCPUの暴走の種類と原因について述べる．なお，本論文で考えるシステムは，プログラムがあらかじめROMに入っているものとし，外部記憶装置からプログラムをRAMに読み込み使用するものではないとする．

9.2.1 CPU暴走の種類

CPUの暴走には次の2種類があると考える．

(1) 長時間正常ルーチンに戻らないもの

(2) 短時間(数命令)異常な命令を実行し，または数命令スキップした後，正常ルーチンに戻るもの

(1)は長い間誤ったプログラムを実行し続け，正常ルーチンに戻らないものである．一般的にシステムが暴走していると言われるのはこの場合であり，従来のWDTはこの状態を検出しようとするものが多い．

(2)は数命令だけ誤ったプログラムを実行し，そのあとすぐに(例えば従来のWDTにおける暴走検出タイマの設定時間以内に)正常ルーチンに戻るか，または数命令をスキップするだけで，その後正常ルーチンを実行し続ける場合である．しかし，数命令の暴走であっても，その間にデータを書き換えたり，ポート出力等を行い，システムに重大な影響を及ぼすことや，9.2.2 で述べるように，書き換えられたデータにより別の暴走が発生する事がある．また数命令のスキップの場合も，必要なデータ更新を行わなかったり，ポートの設定を行わない等，システムに重大な影響を及ぼす事がある．このため，(2)の暴走に対しても検出する必要がある．

9.2.2 CPU暴走の原因

CPUの暴走には様々なものがあるが，ここではその原因を大きく5つに分けて考察する．

(1)CPU自体の暴走

これは，CPU自体に故障が生じたり，CPUの動作環境が悪化し，またはCPUにバグがあり，暴走するものである．この原因としてはCPU自体の故障，CPUのバクの他，CPUに与えている電源，クロック，バス等が不安定になり暴走することも考えられる．CPUの内部は同期式順序回路で構成されており，クロックの立ち上がり，立ち下がりに同期して状態を遷移させながら動作しているため，クロックのデューティが狂ったときや，周波数が規定値より高くなった場合，正しい遷移が行われず，暴走する事がある．また，CPUの命令フェッチ時にデータバスが不安定になり，誤った命令をフェッチし，暴走する場合もある．

(2)プログラムメモリによる暴走

プログラムを格納しているメモリから，CPUがプログラムを正しく読み出せずに暴走するものである．この原因としては，メモリ素子そのものの故障の他，メモリ素子に与える電源の不調，読みだし速度が規定値以上で読み出しが不安定であること等も考えられる．

(3)データメモリによる暴走

データを格納しているメモリから，正しいデータを読み出せずに暴走したり無限ループに陥るものである．この原因は(2)と同様の事があげられる．例えばプログラムがデータの値から次の命令実行番地を計算する様な構成になっていたとき，その値が狂い，予期しない値であった場合，CPUは誤った番地からプログラムを実行し暴走する．また，データをリスト構造で格納していたとき，次のレコード位置を示すポインタが狂った場合，特定のレコードをうまくたどることができず，無限ループに陥ることもある．この場合は，CPU自体は正常ルーチンを正しく実行しており，CPUの暴走ではないが，システムとしては正常な処理が停止してしまう．

(4)JUMP TABLEによる暴走

命令の実行番地を格納してあるJUMP TABLEの内容を正しく読み出せずに暴走するものである．この原因も(2)と同様である．システムには，次の実行番地をそのときの状態からJUMP TABLEにより求める様に指定するものがある．また，割込みが発生したときの割込み処理ルーチンの開始番地をTABLEで持たせているシステムも多い．このJUMP TABLEの内容が狂い正しい命令を実行しなくなり，CPUが暴走する場合がある．又割込みのJUMP TABLEが狂ったときは，割込みが発生するごとに暴走を繰り返すことになる．

(5)ソフトウェアのバグによる暴走

ソフトウェアのバグが原因で暴走する場合もある．これは，例えばプログラムのループの終了判定部分にバグがあり無限ループになる場合や，次の命令実行番地をデータから計算する形式のプログラムでは，その計算部分のバグでCPUが誤った番地からプログラムを実行し暴走する場合である．また，JUMP TABLEの内容をプログラムによりCPUが書き込む形式の場合も，書き込み部分にバグがあれば，そのJUMP TABLEを用いて実行番地を変えたときにCPUは暴走する．

9.3 従来のウォッチドッグタイマと問題点

ここで，従来の代表的なWDTについて考察し，その限界について考える．WDTについては，多くの方法が特許申請されている．そこで，公開特許公報の中からWDTに関するものを取り出し，主だったWDTの方式を調べた．従って，各方式が具体的に特定の一申請を示すものではない．

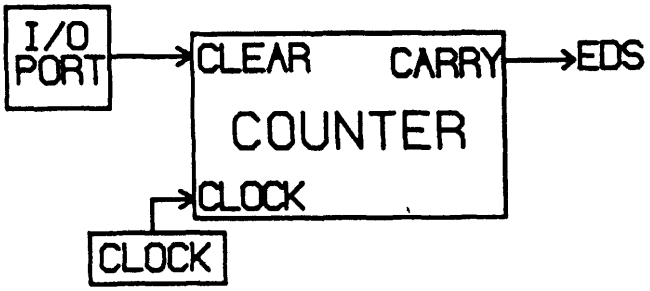


図9.1 従来のWDT(1)
Fig.9.1 Conventional WDT(1).

[1]従来のウォッチドッグタイマ(1)

最も一般的なWDTは図9.1に示すように，CPUとは無関係なクロックによってカウントアップするカウンタを用いるもので，このカウンタのリセット端子にCPUのI/Oポートを接続し，プログラムのメインループに記述したOUT命令によりカウンタの値をクリアするものである．CPUが暴走してOUT命令を一定時間以上実行しないとカウンタが桁あふれをおこし，この桁あふれ信号をEDS(Error Detection Signal)として用い，暴走を検出する．この方法は簡便な方法であるが，次の３つの問題がある．

(1)暴走した結果，OUT命令を含む閉鎖ループを実行した場合

この場合，常にOUT命令によりタイマはクリアされ，桁あふれによる暴走検出は作用しない．

(2)割込み処理を含んだプログラム構成の場合

この場合，割込みプログラムの実行開始番地TABLEが壊れたり，割込みプログラムの内容が変化して暴走が発生しても，割込みプログラム内で閉鎖ループが発生しない場合，メインプログラムは関知せず実行を続け，OUT命令を実行するため，暴走検出が作用しない．

(3)メインプログラム内で暴走しても，すぐ通常のプログラム実行に戻った場合

9.2.1の(2)の暴走の場合である。この場合は，暴走が一瞬であり，すぐに正常ルーチンに戻りOUT命令を実行してカウンタをクリアするため，タイマが

桁あふれせず，WDTはこの暴走を検出できない．

[2]従来のウォッチドッグタイマ(2)

OUT命令によりカウンタをクリアするのではなく，図9.2に示す様にCPUが命令をメモリから取り込む際のアドレスバスの値を比較回路に入力し，この値があらかじめ定めた値(メインループ内のアドレス値)であればカウンタのリセット信号を出力するWDTもある．この方法は上記[1]の(1)の問題に対して，解決を試みるものである．(1)の問題では，プログラムがROMに入っている場合，閉鎖ループはROM上にはできずRAM上にできるので，たとえRAM上で閉鎖ループができ，どの様な命令を実行しようとも，アドレスバスを用いたリセット出力信号回路は働かず，カウンタのクリアは行われない．つまり，OUT命令等特定の命令を用いてリセット信号を出力するのは，メインループの特定の場所をCPUが実行したのを検出するためであるが，この特定の命令が他の場所(例えばデータエリア)に存在する場合もあり，アドレス情報を直接用いる方が確実であることになる．ただしこの方法も[1]の(2)(3)の問題には対応できない．

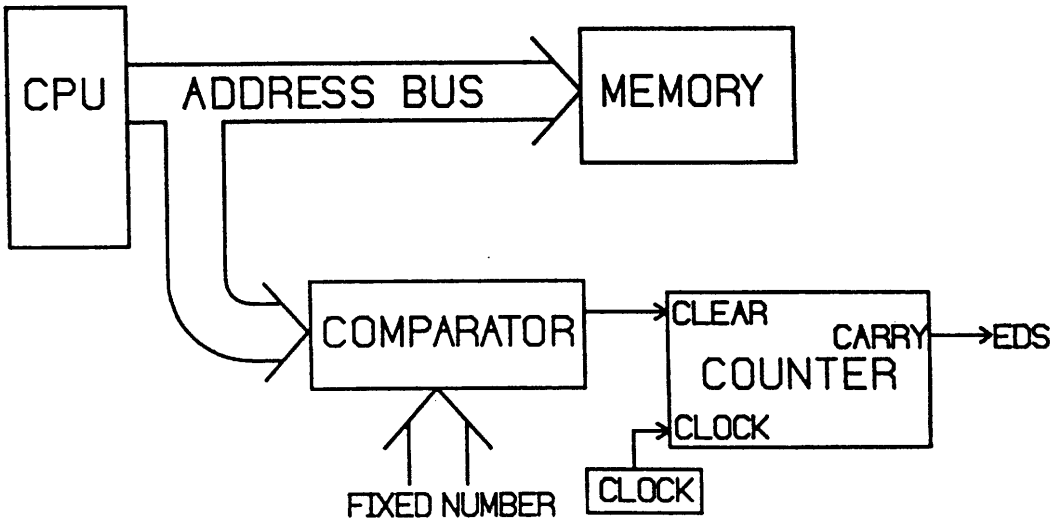


図9.2 従来のWDT(2)
Fig.9.2 Conventional WDT(2).

[3]従来のウォッチドッグタイマ(3)

外部の割込み信号により割込み処理を起動し処理を行っているシステムで，図9.3の様にこの割込み処理によりカウントアップし，割込み処理中にカウンタのリセット信号を出力するようにしているものもある．この方式は，自動車のコントロールシステムに多くみられ，エンジンの回転パルスを割込み信号に用いている．これは暴走検出時間をダイナミックに変化させることが出来，エンジンが高速で回転しているときの暴走検出時間が短くなるという特長を持つ．しかし，この方法では，メインルーチンや他の割込み処理内で暴走が起きたときでも，割込み信号により該当する割込み処理を正常に実行し，リセット信号を出力するため，暴走検出が作用しないという問題がある．

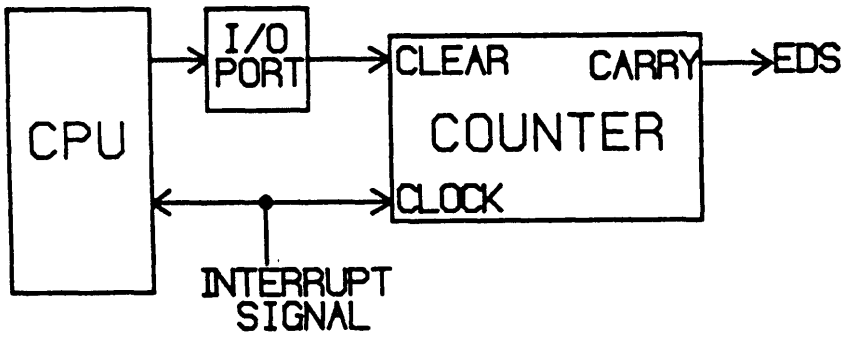


図9.3 従来のWDT(3)
Fig.9.3 Conventional WDT(3).

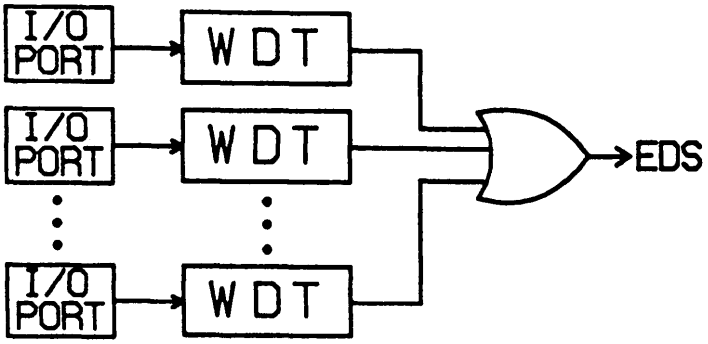


図9.4 従来のWDT(4)
Fig.9.4 Conventional WDT(4).

[4]従来のウォッチドッグタイマ(4)

割込み処理を用いているシステムにおいて，WDTをメインルーチン用だけでなく，各割込みルーチンごとに設ける方式もある．この場合は，複数あるWDTの各暴走検出端子のORをとり，その出力をシステムの暴走検出端子として用いる(図9.4)．これは[1]の(2)の問題を解決する．しかし，(1)(3)の問題に対しては効果がない．

以上まとめると，[1]の方式を基本とし，他の方式はこれを改良して用いている．[2]の方式は[1]の(1)の問題を解決し，[4]の方式は[1]の(2)の問題を解決している．[3]の方式はWDTの検出時間をダイナミックに状況に応じて変化させる事ができるが，WDTの暴走検出能力自体を低下させている．

このように見ると，これまでのWDTの研究は[1]の(1)(2)の問題を解決しようとする方向にあり，[1]の(3)の問題が残されてきたように思える．つまり，一瞬暴走してすぐ正常ルーチンに戻ったようなときに対する処置が不十分であるといえる．そこで9.4において，WDTの暴走検出能力について統計的に考察し，9.5において[1]の(1)(2)のみでなく(3)の問題も解決するWDTについて考察する．

9.4 WDTの暴走検出能力と暴走時のCPUの動作

本章では，WDTを抜き取り検査の一種としてWDTの暴走検出能力について考察し，次に暴走したときのCPUの動作について調べる．

9.4.1 WDTの暴走検出能力

WDTの原理は，プログラム中のある特定の命令(例えばOUT命令)が正しく実行されていることをカウンタを用いて調べ，その命令が正しく実行されていれば，プログラム全体も正しく実行されていると見なす事である．このことは，品質管理で広く用いられている抜き取り検査[MAKA74][MORI80]と同じ考えである．そこで，WDTを抜き取り検査と見なし，その暴走検出能力を解析する．これは文献[MAKA74][MORI80]等で述べられた抜き取り検査の検出能力と同様の解析である．

WDTを抜き取り検査としたとき，WDTは抜き取り検査の内，計数抜き取り検査における1回抜き取り検査と考えられる．これは，ロットの中からn個の試料を抜き取り，この中の不良品の数をx，あらかじめ定めた合格判定個数をcとし，$x \leq c$のときロットは合格，$x > c$のとき不合格とするものである．WDTの場合，WDTをメインプログラムと各割込みルーチンごとに設けるとすると，ロットはそのWDTが属するメインプログラム又は割込みルーチンであり，合格判定個数cは0ということになる．

ここで，プログラムのステップ数をNとし，暴走した結果実行を飛ばしてしまうステップ数をNp(ここで，抜き取り検査ではpは不良率であり，WDTの場合1回の暴走で飛ばすステップ数が全ステップ数に占める割合にあたる)とする．このとき，WDT用のOUT命令等特定の命令の個数をn(nは抜き取り個数にあたり，1とは限らない)，この内暴走によって飛ばされてしまう命令の数(これがn個抜き取った内の不良品の数にあたる)がxになる確率$P(x)$は次の超幾何確率で表される．

$$P(x) = \frac{\binom{Np}{x}\binom{N-Np}{n-x}}{\binom{N}{n}} \qquad (9-1)$$

ここで，$\frac{n}{N} \leq \frac{1}{10}$とき$P(x)$は近似的に2項確率で置き換えられ，次の様に表されることが知られている[MAKA74][MORI80]．

$$P(x) = \binom{N}{x} p^x (1-p)^{n-x} \qquad (9-2)$$

計数抜き取りで1回抜き取り検査の場合，ロットの不良率がpのとき合格する確率$L(p)$は，

$$L(p) = \sum_{x=0}^{c} P(x) \qquad (9-3)$$

であり，WDTの場合は$c = 0$であるので，

$$L(p) = \binom{N}{0} p^0 (1-p)^n \qquad (9-4)$$

となる．この式から，nの値が1 ～ 20までについてpと$L(p)$の関係をグラフにすると図9.5の様になり，検査用の命令数をnとしたときのWDTの暴走検出特性(暴走しても検出できない確率)を表す．暴走検出特性は理想的には$p = 0$のとき$L(p) = 1$，$p \neq 0$のとき$L(p) = 0$であることが望ましく，nの値を増やせば理想に近くなるが，$p < 0.2$のときに検出能力に問題があることがわかる．そこで，暴走したときのCPUの動作を解析し，プログラムにおけるpの値について検討する．

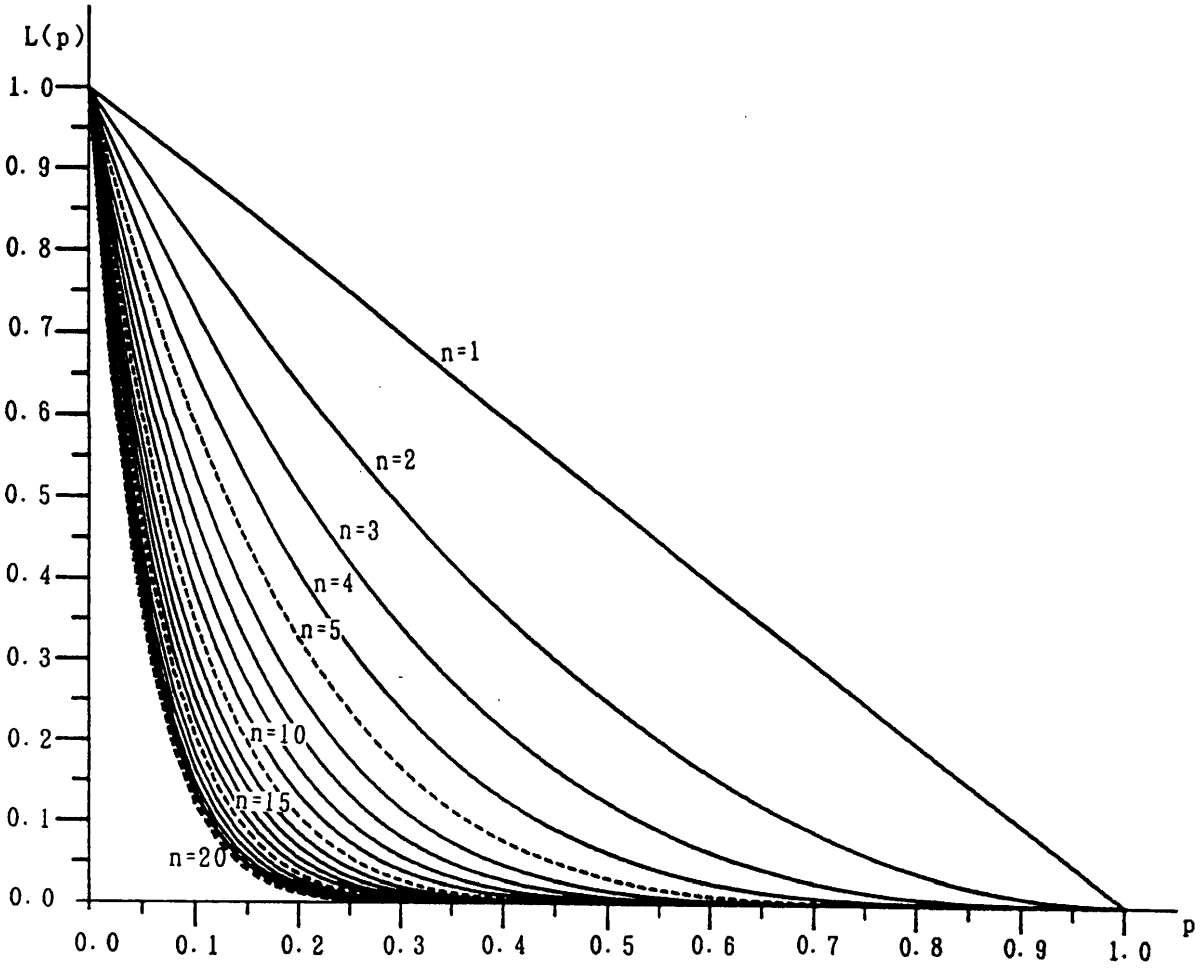


図9.5 ウォッチドッグタイマの有効性
Fig.9.5 Validity of watchdog timer.

9.4.2 CPUの動作

本論文では，CPUが何らかの原因で暴走したあと，CPUのプログラムカウンタは任意の値をとると仮定する(この仮定の正当性については，付録Dで実際にCPUが暴走したときのプログラムカウンタの値を調べることで確かめる)．例えばザイログ社のZ80CPUの場合，プログラムカウンタは16bitであるので$(0000)_{16}$ ～ $(FFFF)_{16}$の任意の値をとる．このとき，プログラムカウンタの値が実際のシステムにおけるプログラムの範囲外であれば，図9.6に示す回路を用いてCPUが命令をメモリから取り込む際のアドレスバスを監視して，暴走をチェックする事ができる．しかし，暴走した後のプログラムカウンタの値がプログラムの範囲内の場合，図9.6の回路は効果がなく，WDTにより暴走を検出しなくてはならない．

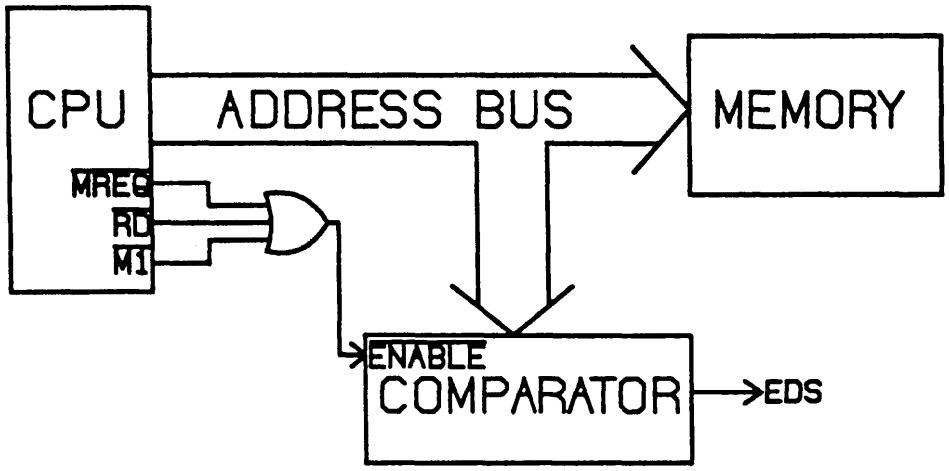


図9.6 アドレスバス監視回路
Fig.9.6 Circuit of checking the address bus.

ここでは，この回路を用い，暴走した時のプログラムカウンタの値がプログラムの範囲外であれば確実に検出できるものとし，カウンタ値がプログラムの範囲内である場合について考える．このとき，CPUはプログラムカウンタが指すプログラム内の任意のアドレスから実行するが，このアドレスが複数バイト命令の第1ワードでなかった場合更に暴走を続け，暴走の結果プログラムカウンタが命令の第1ワードを指したときから正常状態に戻ることになる．

9.4.1においてWDTの暴走検出能力は不良率pに大きく依存することが明らかに

なったため，暴走した結果プログラムカウンタの値がプログラムの範囲内になり，図9.6の回路が働かないときのpの値を解析する．この場合のpの値は，暴走した結果スキップされたり正常に実行されなかったプログラムの部分がプログラム全体に占める割合である．

まず，暴走の結果プログラムカウンタ値が狂ったため実行されなかった命令についてであるが，ここではカウンタ値がプログラム内の値になった場合のみを考えている．カウンタ値はこのプログラム内の値を一様にとると仮定しているので，pの値は0.5ということになる．

次に，狂ったプログラムカウンタ値から実行し，正常なプログラム実行に戻るまでにスキップされたり正常に実行されないプログラム数を，その際のCPUの振舞いをシミュレートすることにより求める．このCPUの振舞いは各CPUによりそれぞれ異なるため，ここでは制御システムで多く用いられているザイログ社のZ80CPUについてシミュレートした．まず，異常時のCPU動作を表す状態遷移図を作成し，次にその遷移確率を求め，この遷移図に従いシミュレーションを行った．

[1] 状態遷移図

Z80CPUが狂ったカウンタ値から実行したときの例を図9.7に示す．ここでは，$(1000)_{16}$番地から始まるプログラムを，カウンタ値が$(1002)_{16}$になったため，3ワード命令の3ワード目から実行が開始された場合の様子を示している．プログラム作成者が意図したのはMNEMONIC(1)に示す処理であるが，CPUはMNEMONIC(2)に示す動きをする．このため，5つの命令を正常に実行できず，$(1009)_{16}$番地のOR命令から正常な命令実行に戻っている．

シミュレーションにあたっては図9.8に示す状態遷移図を作成した．この遷移図は各状態が暴走したときの各CPUの状態に対応している．状態0は正常にプログラムを実行している状態であり，状態1が暴走して，カウンタ値が狂った瞬間を示す．シミュレーションは状態を1にしてから，また0に戻るまでに正常に実行されなかったプログラム数をカウントする．状態2はカウンタが2ワード命令の2ワード目を指している状態を，状態3はカウンタが3ワード命令の2ワード目を指し，状態4はカウンタが3ワード命令の3ワード目を指している状態を表している．これらは全て暴走状態であり，プログラム作成者が書いたプログラムとは異なる動作をCPUはする．状態5は遷移図の都合上用いた状態で，暴走して異なる動作をしているCPUが

ADDRESS	MNEMONIC(1)	OP CODE	MNEMONIC(2)
1000	ld hl,(2244h)	2A	
1001		44	
1002		22	ld (2AEBh),hl
1003	ex de,hl	EB	
1004	ld hl,(2245h)	2A	
1005		45	ld b,l
1006		22	ld (7E19h),hl
1007	add hl,de	19	
1008	ld a,(hl)	7E	
1009	or a	B7	or a
⋮	⋮	⋮	⋮

図9.7 暴走の例
Fig.9.7 Example of fault.

実行した命令が3ワード命令のとき，飛ばした2ワード目にある命令が本来何ワード命令であったかを調べるのに用いている．

CPU暴走時の振舞いを，この状態遷移図が状態を遷移しながらシミュレートする様子を，図9.7で示した暴走の例を用いて説明する．この例では，暴走したとき，3バイト命令の3バイト目から実行を開始するので，遷移図では状態が0 → 1 → 4と遷移する．次に3バイト命令の2バイト目を実行するので状態3へ遷移し，さらに3バイト命令の3バイト目を実行するので状態4へ遷移をし，この後正常に戻るので状態0へ遷移する．つまり，この暴走の場合は，0 → 1 → 4 → 3 → 4 → 0と遷移することになる．

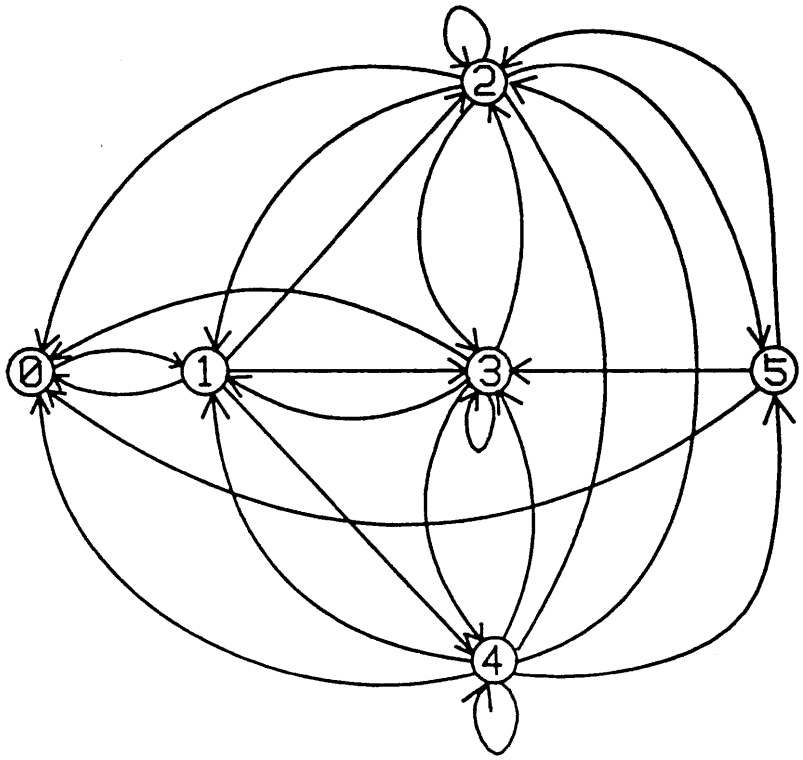


図9.8 CPU暴走時の状態遷移図
Fig.9.8 State transition diagram of a CPU in case of fault.

[2] 遷移確率

この状態遷移図の各遷移条件を満たす確率を求め，コンピュータによりシミュレーションを行う．ここでこの確率を求めるため，まずプログラムの中に含まれる1,2,3ワード命令の割合を実際のプログラムをいくつか調べる事により求めた．この結果を表9.1に示す．

ここでは任意に選んだ4つのプログラムを用い，その総命令数と1,2,3ワード命令の数を調べた．なおここでAのプログラムはロボットの制御ソフト，Bは誤り符号訂正のソフト，C,Dはネットワークのサーバのソフトである．表9.1によると，各ソフトにより1,2,3ワードの命令の割合は，ある程度似たものであることがわかる．もちろんこの割合と異なるソフトもあるがここではこの表より1,2,3ワードの命令の割合を4：4：2とした．プログラム中に各ワード数の命令が占める割合は，この割合にワード数を掛けて4：8：6になり，暴走したときプログラムカウンタが1ワード命令，2ワード命令の1ワード目又は3ワード命令の1ワード目を指す確率は

表9.1 プログラム中の1,2,3バイト命令の比

プログラム	総命令数	1byte 命令数 (割合)	2byte 命令数 (割合)	3byte 命令数 (割合)
A	6459	2488 (0.39)	2262 (0.35)	1709 (0.26)
B	948	453 (0.48)	395 (0.42)	100 (0.10)
C	560	252 (0.45)	194 (0.35)	114 (0.20)
D	54	16 (0.30)	25 (0.46)	13 (0.24)

$(4+4+2)/(4+8+6) = 10/18$になる．これは状態1から0に遷移する確率である．

同様に状態1から2に遷移する確率は，暴走したときプログラムカウンタが2ワード命令の2ワード目を指す確率であるから，4/18．状態1から3への遷移確率は，3ワード命令の2ワード目を指す確率で2/18．状態1から4 への遷移確率は，3ワード命令の3ワード目を指す確率で2/18となる．

状態2からの遷移は，2ワード命令の2ワード目が1ワード命令であるか，2ワード命令であるか，3ワード命令であるか，ジャンプ命令であるか，また以後の命令が何ワードの命令であるかによって決定する．この2ワード目の値はこの命令のオペランド値であり，データもしくはアドレス情報が書かれているため，$(00)_{16} \sim (FF)_{16}$の値が一様に表れると考え，Z80の全ニーモニック中に占める1,2,3ワード命令とジャンプ命令の割合を調べ，この値と前記プログラム中に占める各命令の割合から状態2からの遷移確率を求めた．他の状態についても同様に求め，表9.2に示すような遷移確率を得た．この表は状態iから状態jに遷移する確率を示す．

表9.2 CPU暴走時の状態遷移確率

i \ j	0	1	2	3	4	5
0	-	-	-	-	-	-
1	10/18	-	4/18	2/18	2/18	-
2	4094/5120	540/5120	172/5120	86/5120	76/5120	152/5120
3	582/5120	540/5120	152/5120	76/5120	3770/5120	-
4	4094/5120	540/5120	172/5120	86/5120	76/5120	152/5120
5	4/10	-	4/10	2/10	-	-

[3] シミュレーション結果

[2]で求めた確率に従いコンピュータでシミュレーションを行い1回の暴走で正常に実行されないプログラム数と，この暴走中に実行するジャンプ命令の数を調べた．10万回暴走を繰り返し起こしその平均をとったところ，1回の暴走で正常に実行されないプログラムの数は0.512命令，実行するジャンプ命令は0.065回となった．正常に実行されないプログラムの命令数は非常にすくなく，CPUはすぐに正常ルーチンに戻ることがわかる．これは，Z80の場合1ワード命令が大変多い(全命令の約79%)ためだと考えられる．

このシミュレーション結果から，プログラムにおける不良率pを求める．pは，暴走した結果スキップしたり正常に実行されなかったプログラムの部分が，プログラム全体に占める割合である．このスキップしたり正常に実行されなかった部分とは，暴走の結果プログラムカウンタが狂いスキップした部分と，狂ったプログラムカウンタ値から実行し，正常なプログラム実行に戻るまでにスキップしたり正常に実行しない部分とから成る．前者は前述のようにプログラム全体の50%にあたり，後者はこのシミュレーション結果より，正常に実行されない0.512命令と0.065回のジャンプ命令によりスキップする命令の合計となる．この後者がプログラム全体に占める割合は，0.065回のジャンプ命令による3.25%(0.065 × 50%)と0.512命

令がプログラム全体に占める割合となり，通常のプログラムのようにプログラムが数Kバイトから成る場合は0.512命令の割合は無視できる．このことから，プログラムにおける不良率pは，$0.5+0.0325$ となり，およそ0.53となる．

このことから，図9.5より従来のように$n=1$のときは暴走検出能力があまり高くないのに対し($n=1$のとき$L(0.53)=0.47$であり，暴走しても見逃す確率が0.47ある)，$n \geq 7$のときはその約100倍の暴走検出確率($n=7$のとき$L(0.53)=0.0051$，$n=8$のとき$L(0.53)=0.0028$)が得られることがわかる．

なお，この検出能力は，9.2.1で考えた2種類の暴走についての能力であり，実際には9.2.1の(2)の一瞬の暴走により危険な事態が起きることはほとんどない．このことについては後述する．

9.5 暴走検出能力の高いWDT

これまでのWDTに対する考察をふまえて，暴走検出能力の高いWDTを提案する．ここでは，従来のWDTを用いたシステムにハードウェアの追加を最小限にして，暴走検出能力の向上を実現することを考える．

これまでの考察により，図9.6で示すアドレス監視回路を設けた上で，検査箇所を1箇所だけではなく例えば7箇所以上メインループ内に入れておくと，暴走検出能力が100倍以上向上することがわかった．

しかし，ただ出力命令を7箇所入れるだけでは$n=7$にはならない．$n=7$とするためには，7箇所の出力命令が順に重複する事なく実行されていることを確認する機能を付加しなくてはならない．そのために，図9.9で示すアルゴリズムを用いる．図中Xは8bitのRAM上に割り当てた変数であり，図では$n=8$となっている．このアルゴリズムは各チェック箇所ごとにXの決められたbitを反転し，Xの値が正常かどうか判定し，正常ならばカウンタのクリアパルスを出力する様になっている．反転するbitはチェック箇所が実行される順に1bit目から1 づつずらせ，最後のチェック箇所は8bit目を反転する．

このため，たとえば2番目に実行されるチェック箇所においては，2bit目を反転した結果，Xの値は暴走して他のチェック箇所を飛ばしていない限り$(00000011)_2$または$(11111100)_2$ということになる．これが，例えば1番目のチェック箇所を暴走の結果飛ばしていた場合，Xの値は$(00000010)_2$または$(11111101)_2$となり，反転し

た結果が1でも0でも検出できることになる．またこの2番目のチェック箇所を実行した直後に暴走して再び2番目のチェック箇所を実行した場合には，Xの値は更に2bit目が反転して$(00000001)_2$または$(11111111)_2$となり，この場合も検出できることになる．

割込み処理があるシステムでは，各割込み処理ルーチンごとに別のWDTを用意するがこの各ルーチンごとのWDTもこの方法で実現する．

図9.9では，Xの値が異常なときは，何もしていないが，この場合は暴走処理ルーチンへ処理を移してもよい．

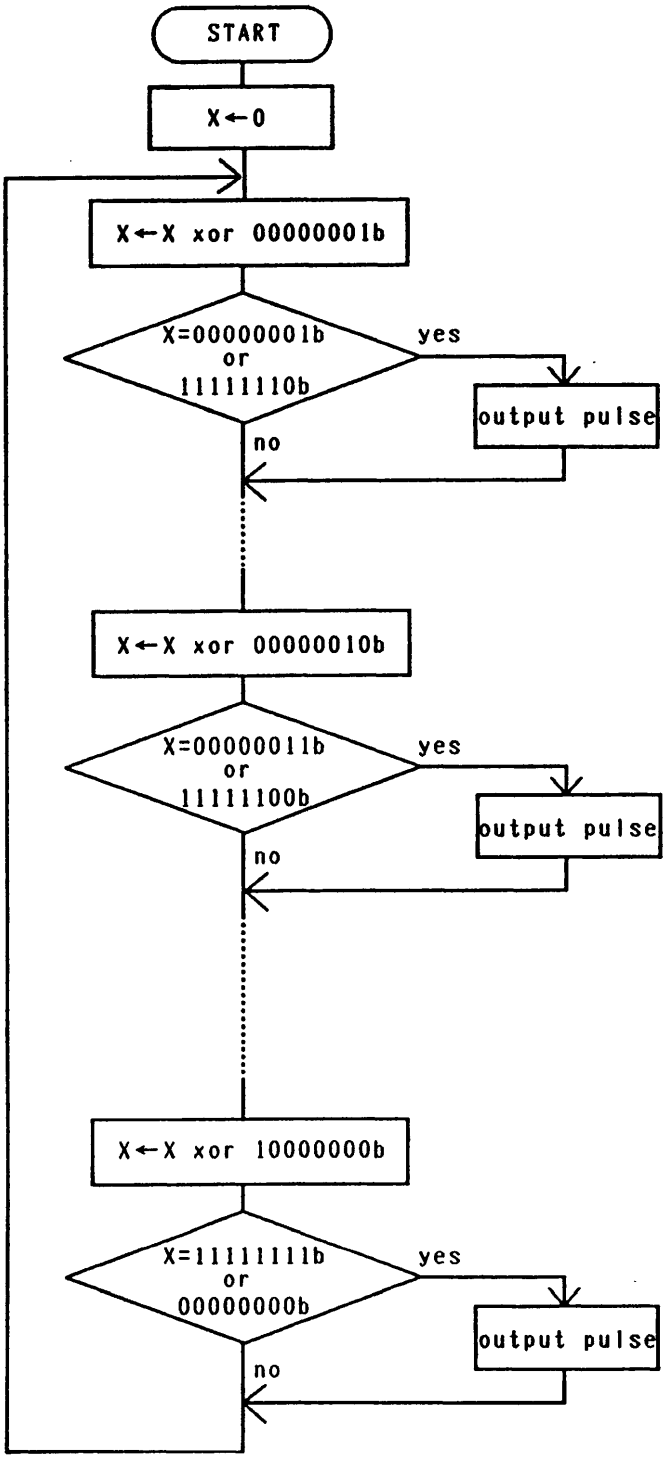


図9.9 提案するWDTのフローチャート
Fig.9.9 Flow chart of proposed WDT.

9.6 考察

提案した，図9.6の回路と図9.9のアルゴリズムを付加したWDTの，暴走検出能力について考える．図9.6の回路により，暴走しプログラムカウンタ値がプログラムの範囲外になった場合は，暴走を確実に検出できる．このため，例えばメモリ空間が64KBのCPUの場合，プログラムが16KBであれば，この回路により，暴走の3/4は検出できることになる．更に，図9.9のアルゴリズムを用いると，9.4.2によりプログラムカウンタ値がプログラム範囲内になった場合の検出能力は，$L(0.53) = 0.0028$ ($n = 8$の時)であるため，図9.6の回路と組み合わせて，暴走したときその暴走を検出できない確率は$0.0028 \times (1/4)$で約0.1%となる．これは，従来のWDT(例えば図9.1の回路を用いたもの)と比べ，約500倍能力が向上したことになる．

なお，一瞬暴走したとしても，実際に危険な事態が起こる確率は小さい．ただ，その暴走が危険であるかどうかの判定は，システム全体を見なければプログラムだけを見ていたのではわからないため，危険な事態が起こる確率を正確に算定することは不可能である．ここでの確率0.1%は，この一瞬の暴走の検出も考えた値であり，従って危険な確率の上限値と解釈すべきである．

また，本論文ではWDTのハードウェアについては特に触れていないが，文献[NAMI85] で述べられた様な，自身のハードウェア故障を外部に知らせる機能を持ったWDTを用いることが好ましい．

9.7 結語

従来のWDTを調べ，暴走検出が完全ではなく，暴走しても検出できず暴走を続ける場合があることを明らかにした．またWDTが抜き取り検査と見なせるとして検出能力について統計的に検証し，さらに暴走時のCPUの振舞いのシミュレーション結果から，より暴走検出能力の高いWDTを提案した．このWDTは従来のものに簡単なハードを追加し，ソフトウェアを一部変更するだけあるが，暴走検出能力は飛躍的に高まることを示した．今後，アドレス監視回路をゲートアレイICを用いて1チップ化し，提案したWDTのより容易な実現を図る予定である．

付録D 暴走時のプログラムカウンタ値

本論文では，暴走が起きた場合，その飛び先がプログラム全体に一様であるという仮定の下で解析を行った．そこで，実際のコントローラにおいて暴走した場合の飛び先を調べる実験を行い，この仮定の妥当性を調べた．実験に用いたコントローラは，実際に大学内のネットワークに用いられているもので，これにCPUのバスを監視する回路と，暴走した時のバスの状態を記録するFIFOメモリ回路から成る測定回路を取り付けた．この測定回路に測定用のコンピュータを接続して，暴走した時の飛び先を，FIFOメモリに記録されたバスの状態から求めた．実験は２種類の暴走原因に対して行った．

１つはCPUのクロックにノイズを与えて暴走させるもので，1000回暴走させた時のそれぞれの飛び先を図9.10に示す．２番目の実験は，図9.11に示すようにCPU基板の面に対し平行に，基板上のCPUの端子から5mm離した所で，7Kvの電位差を持つように帯電した一対の電極を，CPUの端子の並びと直角方向に約5mmの距離に近づけ放電させ，CPUを70回暴走させるもので，この時の飛び先を図9.12に示す．

図9.10，図9.12は飛び先の番地の上位8bitを横軸に，下位8bitを縦軸にとり，その交点に丸を書いており，どちらも飛び先が一様である様子が分る．様々な暴走原因全てを試すことは出来ず，限られた実験であるが，この図からはこの仮定の妥当性が確認できる．

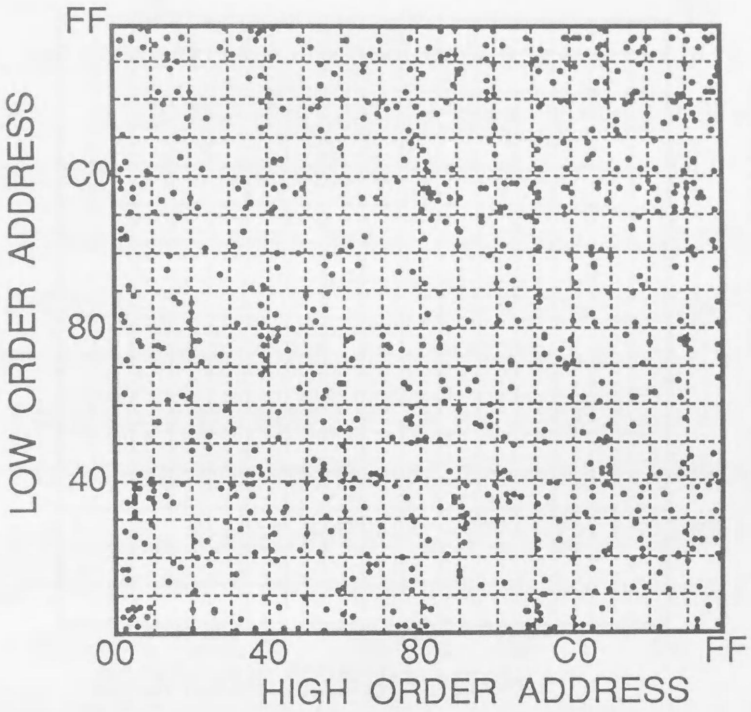


図9.10 暴走時のプログラムカウンタ値(1)
Fig.9.10 Value of program counter in case of fault(1).

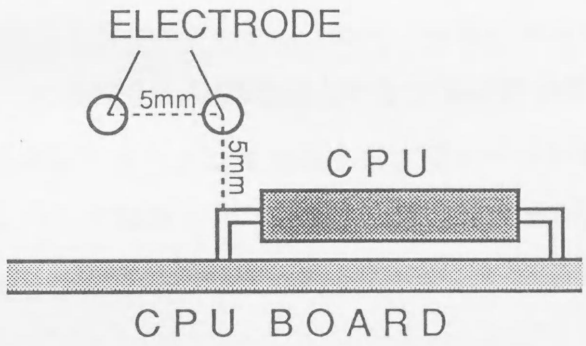


図9.11 実験システムの構成
Fig.9.11 Diagram of the experimental system.

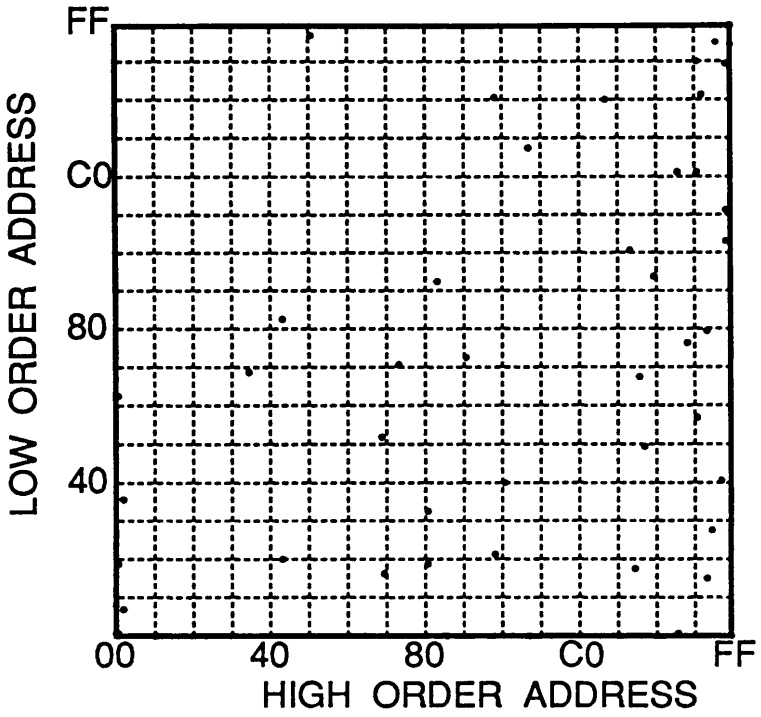


図9.12 暴走時のプログラムカウンタ値(2)
Fig.9.12 Value of program counter in case of fault(2).

結論

本論文では，第1部(第1章)でリング型キャンパスネットワーク(S-net)の構築と運用について述べ，第2部(第2〜7章)でリング型ローカルエリアネットワークの高速化，第3部(第8,9章)でその信頼性向上について論じた．

第1章では，初めにS-netの構成と機能について説明した．研究施設が複数の建物に分散し，各建物内に多くのコンピュータが設置されている大学環境に合わせ，S-netは建物毎にリングを敷設してこのリングに建物内のコンピュータを接続し，さらに各リング間をリング状に接続する二重のリング構造とした．また，研究用ネットワークでの使用実験より，遠隔端末機能やデータ交換機能，電子メール機能等を持たせることとした．S-netの各機器は，同時に幾つもの回線からのデータを処理しなければならず，リアルタイム処理と並行処理が必要となるため，状態遷移法と呼ぶ並行処理実現手法を用いて設計した．

続いて，S-netの現状について述べた．現在，S-netを信州大学の2つのキャンパスに敷設し，各キャンパスのS-netをマイクロ回線で接続している．S-netに接続されているコンピュータは400台以上であり，1990年1年間のS-net利用者数は延べ48934人，24時間常に利用者がいる．論文では，この7年間のS-netの利用実績を示すと共に，大学におけるネットワークの需要を明らかにした．

最後に，利用者に対して行ったアンケートから，今後のキャンパスネットワークに対し，ネットワークの高速化と信頼性向上の2つの要望が多いことを示した．

第2部では，高速なネットワークに最適なリング型ローカルエリアネットワークのMACプロトコルについて検討した．これは，伝送速度が高速な場合において，様々なMACプロトコルの性能解析を行い，各プロトコルの適用領域を明らかにしたものである．

第2章では，各MACプロトコルを解析する際の共通の仮定について述べた．最適なMACプロトコルを求めるために，本論文では同一条件下で各MACプロトコルの解析結果を比較出来るよう，各MACプロトコルの解析を，この共通の仮定のもとで行った．この仮定の中では，リング中の各ノード内のバッファ容量を有限としたことが特長である．従来の研究においては，モデル化の簡単化のためにこの容量を無限大としていたが，有限とすることで，ネットワークの大切な特性であるバッファのオーバフローに関する議論が出来るようになった．

第3章では，まずRISリングと呼ぶ新しいMACプロトコルを提案し，次にその性能解析を行った．これは，伝送効率の良いレジスタ挿入リングにハードウェア処理が容易であるスロットリングの考えをとりいれたものである．解析のためのモデル化にあたっては，リング中のスロットの数が1つの場合と複数の場合に分けた．そして，各々の場合について，リング中の全ノードの動作を表すマルコフモデルを作成し，最後にそれらを統合した．

第4章では，トークンリングの性能解析を行った．各ノードにおけるパケットの発生，トークン受信を待ってのパケットの送信，及びトークンの転送の各状態を表したマルコフモデルを作成して，解析した．

第5章では，スロットリングの性能解析を行った．スロットリングのモデルはRISリングとほぼ同じである．異なるのは，パケットのリングからの除去と，各ノードにおけるパケットの転送遅延の2つである．RISリングではパケットが宛先ノードに到着した時にリングから取り除かれるのに対して，スロットリングでは宛先ノードでは取り除かれず，送信元ノードに戻ったときに取り除かれる．また，各ノードでのパケットの遅延は，RISリングではヘッダを受信する時間だけ必要となるのに対し，スロットリングでは1bitのフラグ受信時間のみでよい．

第6章では，レジスタ挿入リングの性能解析を行った．レジスタ挿入リングでは，各ノードごとに独立したモデルを作成し，さらに，各ノードのモデルも，ノード内の非同期で動作する各モジュールを表す複数のサブモデルからなる構成とし，解析可能なモデルを作成した．また，レジスタ挿入リングではソフトウェアによる処理のオーバヘッドを考慮しなければならないが，S-net で用いているノードの実際の処理時間からこのオーバヘッドを求め，現実に沿った性能解析を行った．

第3〜6章の各MACプロトコルのモデルの解析に当っては，すべて平衡点解析の手法を用いた．既に衛星ネットワークやバス型LANにおける様々なプロトコルの解析に成功しているこの平衡点解析の適用範囲を，本論文はリング型ネットワークへ拡張したことになる．また，シミュレーションによる数値例との比較により，解析結果の精度が高いことを確かめた．

第7章では，第2部のまとめとして，高速ネットワークの様々な条件下で各MACプロトコルの解析結果を互いに比較し，各条件下でパケットの伝送時間がもっとも少ない最適なプロトコルを明らかにした．この際，ノード内バッファのオーバフローを考慮し，オーバフローしてパケットが失われる確率が一定値以下で実用に

耐える条件下で最適プロトコルを求めた．

ノードの台数が少なく，パケット発生も少ないときにはトークンリングが最適である．ノードの台数が多く，パケット発生が少ないときにはスロットリングが最適である．

パケット発生が多くなると，パケットが宛先ノードで取り除かれるレジスタ挿入リングとRISリングが有利になる．RISリングは，レジスタ挿入リングのパケット送信タイミングをスロット化することで処理をハードウェア化したものである．このため，パケットの送信は一定周期ごとに限られ，レジスタ挿入リングに比べて端末バッファ内での待ち時間が増える．一方，パケット転送に要するオーバヘッドがレジスタ挿入リングより小さい．このことから，ノードの台数が少ないときは端末バッファ内での待ち時間が少ないレジスタ挿入リングが最適となる．また，ノードの台数が多いときは，パケット転送時の遅延が小さいRISリングが最適となる．

第3部では，リング型ローカルエリアネットワークの信頼性向上のための手法として，ネットワークに接続された各装置の異常を各装置内のCPUが早期に発見して自動的に復旧させる手法と(第8章)，このCPU自体の異常を発見して復旧させる手法(第9章)を提案した．

第8章では，リング型ネットワークにおけるノードのハングアップを自動的に検出し，復旧させ，その後ハングアップ箇所の報告を行う手法を提案した．この方法は，リング型ネットワークの形状的な特徴を利用して，各ノードが定期的にチェック用パケットを送り，この受信状況から自己内のハングアップを検出するもので，必要なハードウェアの付加が少なく，特別な管理装置を必要としない等の特長を持つ．また，本方法は必要とするパケットの通信量も少なく，通常のパケット交換への影響も殆どない．さらに，同時に複数のハングアップが発生しても誤動作しない．

第9章では，従来より用いられているウォッチドッグタイマの暴走検出能力を高める方法について提案した．まず，ウォッチドッグタイマを品質管理で用いられている抜取り検査の一種とみなして，ウォッチドッグタイマのCPU暴走の検出能力を解析する手法を確立した．そして，S-netの各装置で用いられているザイログ社製Z80CPUの暴走時の振舞いを表す状態遷移図を作成し，従来のウォッチドッグタイマの暴走検出能力とその問題点を明らかにした．さらに，抜取り検査では，サンプル数が多くなると検出能力が飛躍的に向上することを用い，暴走検出能力が

更に高いウォッチドッグタイマの構成方法を提案した．提案したウォッチドッグタイマは，従来のものに比べて，検出能力が約500倍となった．

本論文の特長として，リング型のキャンパスネットワークS-netを実際に設計し，7年間にわたり運用してきた経験をもとに，高速化と信頼性向上の議論を行っていることがあげられる．接続計算機数が数百台と大規模なキャンパスネットワークの7年にわたる運用経験をもつ大学は少なく，本論文によって明らかになったキャンパスネットワークに対する需要や要求は，キャンパスネットワークのあるべき姿を考える上での貴重な資料である．また，各MACプロトコルの性能解析にあたっても，実際のノードの処理状態を反映したモデル作成が可能となった．更に，高速ネットワークの性能解析では，ノード内処理のオーバヘッドをどのような値に見積るかが重要な問題となるが，本論文ではS-netで実際に使用しているノードのオーバヘッドの値を用ることができ，現実にそった解析が可能となった．

また，本論文中で提案したRISリングだけでなく，リング型ネットワークの代表的な3つのMACプロトコルすべてを，共通の仮定のもとで共通の手法を用いて解析したことも，本論文の特長である．特に，各ノード内のバッファの容量を有限とした仮定で解析が出来たことにより，従来の研究では求められなかったバッファのオーバフロー確率や，バッファ内のパケット数を求めることが可能となった．このため，オーバフロー確率を考慮した上で，高速ネットワークにおける様々な条件における，最適なMACプロトコルを求めることができた．

以上，本論文の結果は，今後のリング型ネットワーク設計における基礎として，重要な資料となると考えている．

今後，本論文の結果をふまえて，次の研究を行う予定である．

まず，S-netの拡張，高速化，信頼性の向上を早急に行う予定である．その際，各ノードのパケット発生状況を調べることで，実際のネットワークの負荷を求め，その結果を第7章で示した各MACプロトコル毎の有効適用領域にあてはめて，最適なMACプロトコルを決定する必要がある．また，第9章で述べたWDTをゲートアレイICを用いて1チップ化し，実装を容易にする必要もある．

リング型ネットワークにおけるMACプロトコルの解析に関しては，各ノードにおけるパケットの発生過程を，複数の発生源があるものに拡張して，今後需要の増大が予想されるマルチメディアネットワークの環境下におけるプロトコルの性

能を解析する予定である．また，MACプロトコルだけでなく，上位の論理リンク制御(LLC)副層でのプロトコルについても，検討する予定である．

謝辞

本論文まとめるにあたり，終始熱心な御指導と御教示を賜った，名古屋工業大学電気情報工学科 川口 喜三男教授に，厚く御礼申し上げます．

本論文まとめるにあたり，数々の御教示を賜った，名古屋工業大学電気情報工学科 池田 哲夫教授，生産システム工学科 小和田 正教授に，厚く御礼申し上げます．

本研究に関し，懇切なる御指導ならびに御助言を賜った，名古屋工業大学電気情報工学科 田坂 修二助教授に，厚く感謝いたします．

本研究を開始する機会を与えて下さり，常に御指導，御教示を賜り，本論文をまとめるにあたりご支援頂いた信州大学工学部情報工学科 中村 八束教授に，厚く感謝いたします．

本研究の基礎となった信州大学キャンパスネットワークの構築，運用は，信州大学工学部情報工学科基礎講座の皆様の多大なる協力があってはじめてできたものです．特に，芹内 美樹氏(現日立超LSIエンジニアリング(株))，水野 貴博氏(現日本IBM(株))，清水 英孝氏(現長野県情報技術試験場)，新村 正明氏(現長野県企業局)，相浦 広国氏(現日本電信電話(株))及び現在大学院修士課程在学中の和崎 克己君，大河原 輝雄君は，キャンパスネットワークの設計に際しても大いに協力頂きました．ここに，深く御礼申し上げます．

参考文献

[ABDU84] F. Abdul-Ghani and P.A. Davies: "High-speed optical-fibre ring network with a register insertion access protocol", IEE PROCEEDINGS, Vol.131, Pt. H, No.2 (Apr. 1984).

[BHUY89] Laxmi N.Bhuyan, Dipak Ghosal and Qing Yang: "Approximate analysis of single and multiple ring networks", IEEE Trans. Comput., Vol. C-38, No.7, pp.1027-1040 (July 1989).

[BUX83] Werner Bux and Marcel Schlatter:"An approximate method for the performance analysis of buffer insertion rings", IEEE Trans. Commun., Vol. COM-31,No.1, pp.138-146 (Jan. 1983).

[CLAR82] Dave Clark:"MIT compus network - inplementation planning document", Massachusetts institute of technology (1982).

[COKE72] C. H. Coker: "An experimental interconnection of computers through a loop transmission system", Bell system technical journal, Vol.51, No.6, pp.1167-1175 (June 1972).

[DEC80] "The ethernet, a local area network: data link and physical layer specifications, version 1.0", DEC, Intel, XEROX (Sep.1980).

[EBIH85] Yoshihiko Ebihara, Katsuo Ikeda, Tomoo Nakamura, Shigeo Nakatsuka and Michihiro Ishizaka: "Fault diagnosis and automatic reconfiguration for a ring subsystem", Computer Networks and ISDN systems, Vol.10, No.2, pp.97-109 (1985).

[EBIH88] 海老原 義彦,劉 暁明,池田 克夫:"バイパス機能を持つ多重多段リングシステムの信頼性",情報処理学会論文誌, Vol.29, No.6, pp.614-622 (June 1988).

[FLIN83] David C.Flint:"The data ring main : an introduction to local area networks",Wiley Heyden Ltd. (1983).

[FUWA87] 不破 泰,中村 八束,清水 英孝:"無手順型キャンパスネットワークS-netについて", 電子情報通信学会情報ネットワーク研究会 IN87-72 (Nov. 1987).

[FUWA88a] 不破 泰,中村 八束:"ウォッチドッグタイマの有効性に関する統計的考察",電子情報通信学会論文誌D, Vol.J71-D, No.11, pp.2414-2423 (Nov. 1988).

[FUWA88b] 不破 泰,中村 八束,清水 英孝:"2重のリング構造から成るキャンパスネットワーク -S-netについて-", 電子情報通信学会論文誌B, Vol.J71-B, No.12, pp.1672-1681 (Dec. 1988).

[FUWA89a] 不破 泰,中村 八束:"リング型ネットワークにおけるノードのハングアップ検出と復旧法について", 電子情報通信学会論文誌B-I, Vol.J72-B-I, No.9, pp.730-740 (Sep. 1989).

[FUWA89b] 不破 泰,中村 八束,新村 正明:"レジスタ挿入型スロットリングにおける同期の確立アルゴリズムについて", 電子情報通信学会情報ネットワーク研究会 IN89-68 (Sep. 1989).

[FUWA90a] 不破 泰,田坂 修二:"レジスタ挿入方式を用いたスロットリングの性能解析", 電子情報通信学会論文誌B-I, Vol.J73-B-I, No.3, pp.179-190 (March 1990).

[FUWA90b] 不破 泰,田坂 修二:"有限バッファを持つリング型ネットワークの性能解析 -トークンリング-", 電子情報通信学会情報ネットワーク研究会 IN90-57 (Sep. 1990).

[FUWA90c] 不破 泰,田坂 修二:"有限バッファ端末を持つリング型ネットワークの性能解析 -スロットリング-", 電子情報通信学会情報ネットワーク研究会 IN90-62 (Oct. 1990).

[FUWA90d] 不破 泰,田坂 修二:"レジスタ挿入型スロットリングにおけるスロットタイミングの高効率化とその性能解析", 電子情報通信学会論文誌B-I, Vol.J73-B-I, No.11, pp.813-824 (Nov. 1990).

[FUWA90e] 不破 泰,田坂 修二:"有限バッファ端末を持つリング型ネットワークの性能解析 -レジスタ挿入リング-", 電子情報通信学会情報ネットワーク研究会 IN90-68 (Dec. 1990).

[FUWA91a] Yasushi Fuwa and Shuji Tasaka:"Register-insertion type slotted rings: a performance analysis", Proc. IEEE INFOCOM, pp.191-201 (April 1991).

[FUWA91b] Yasushi Fuwa, Yatsuka Nakamura and Hirokuni Aiura:"Utilization of the shinshu university campus network -S-net-", IEICE Transactions on Communications, Vol.E74, No.9, pp.2756-2764 (Sep. 1991).

[HAMM86] Joseph L.Hammond and Peter J.P.O'Reilly: " Performance analysis of local computer networks", Addison-Wesley Publishing Company (1986).

[KANA90] 金沢 正憲,石橋 勇人:"京都大学情報通信システムの構成概念と応用",電子情報通信学会情報ネットワーク研究会 IN90-19 (May 1990).

[KOBA78] Hisashi Kobayashi:" Modeling and analysis", Chap.4, Addison-Wesley Publishing Company, Inc. (June 1978).

[KOSI87] R. Kositpaiboon R. and N.D.Georganas: "Performance of integrated circuit/packet slotted rings", Conf. Rec. IEEE GLOBECOM '87,

pp.1804–1808 (Nov. 1987).

[KROP72] W. J. Kropfl: "An experimental data block switching system", Bell system technical journal, Vol.51, No.6, pp.1147–1165 (June 1972).

[LIU88] Ming–Kang Liu and David G.Messerschmitt: "Skew time slot switching and slotted–ring in a metropolitan area network", Proc. IEEE INFOCOM, pp.568–575 (March 1988).

[LOUC85] Wayne M.Loucks, V.Carl Hamacher, Bruno R.Preiss and Luke Wong: "Short–packet transfer performance in local area ring networks", IEEE Trans. Comput., Vol. C–34,No.11, pp.1006–1014 (Nov. 1985).

[MAKA74] 真壁 肇:"品質管理",朝倉書店, pp.239–249 (1974).

[MATU89] 松方 純:"大学における大規模LANの構築",情報処理学会論文誌, Vol.J30, No.11, pp.25–35 (Jan. 1989).

[MCCR82] John W. McCredie:"Strategies for campus computing", Perspectives in Computing, Vol.2, No.3, pp.4–13(Oct.1982).

[MONO84] Monolithic Memories, Inc.: "PAL handbook",(1984).

[MORI80] 森口 繁一:"品質管理",岩波書店, pp.59–86 (1980).

[MORIS86] James H. Morris, et al:"Andrew: a distributed computing environment", Communications of the ACM, Vol.29, No.3, pp.184–201 (Mar 1986).

[NAKA82a] 中村 八束,野田 昭繁:"回線交換・パケット交換のハイブリッドネットワークシステムについて",電子通信学会論文誌B, Vol.J65–B, No. 6, pp.679–686 (Jun. 1982).

[NAKA82b] Yatsuka Nakamura and Yasushi Fuwa:"A simple programming method of state transition diagrams for parallel processings", Journal of information processing, Vol.5, No.3, pp.148–154 (Sep. 1982).

[NAKAM87] 中村 勤,新内 浩介,鈴木 三知男,佐々木 良一,角田 知明:"高速ディジタル網集中管理システムにおける障害管理方式", 電子情報通信学会情報ネットワーク研究会 IN87–52 (Oct. 1987).

[NAMI85] 並木 好広,古賀 義亮:"自己検査性ウォッチドックタイマの一構成法",電子情報通信学会論文誌D, Vol.J68–D, No.8, pp.1543–1544 (Aug. 1985).

[PAUL80] Daniel J. Paulish:"A fail–soft distributed processing system", COMPCON 80 Fall, pp.179–184 (1980).

[PIER72] J. R. Pierce: "Network for block switching of data", Bell system technical journal, Vol.51, No.6, pp.1133–1145 (June 1972).

[RAMA74] C. V. Ramamoorthy,R. C. Cheung and K. H. Kim: "Reliability and integrity of large computer programs", Computer Systems Reliability, pp.617–709, Infortech Information (1974).

[SAKA90] 坂田 真人,根元 義章,野口 正一:"東北大学総合情報ネットワークシステムTAINSの構築",情報処理学会論文誌,Vol.J31, No.11, pp.1661–1671 (Nov. 1990).

[STRO87] Norman C. Strole:"The IBM token–ring network", IEEE NETWORK, Vol.1, No.1, pp.23–30 (Jan.1987).

[TAKA90] Hideaki Takagi: "Queueing analysis of polling models: an update", in Stochastic Analysis of Computer and Communication Systems, Hideaki Takagi (Editor), pp.267–318,Elsevier Science Publishers B.V. (North–Holland), IFIP (1990).

[TANA87] 田中 譲:"HINES計画に寄せて",北海道大学大型計算機センターニュース, Vol.19, No.5, pp.20–34 (1987).

[TANAB89] 田邊正雄,畝本和夫,松尾直樹:"レジスタ挿入型優先制御方式のスロットリングLANへの適用",電子情報通信学会情報ネットワーク研究会 IN89–11(May 1989).

[TASA86] Shuji Tasaka: " Performance analysis of multiple access protocols", MIT Press, Cambridge, MA (1986).

[TOKU84] 徳田 雄洋:"大学や研究所の壁を越えて発展する米国の電子コミュニティ",日経エレクトロニクス, No.353, pp.275–298 (Oct.1984).

[TROP81] Carl Tropper: "Local computer network technologies", Academic Press, New York, NY (1981).

[UPAD86] J. Shambhu Upadhyaya, Kewal K. Saluja:"A watchdog processor based general rollback techniquewith multiple retries", IEEE Trans Software Eng., Vol.12, No.1, pp.87–95 (Jan. 1986).

[VERB87] W. Verbiest, M. Dupponcheel, "Video coding in an ATD environment", GSLB–Seminar on Broadband Switching, pp.317–326 (Jan.1987).

研究業績

論文

[1] Yatsuka Nakamura and Yasushi Fuwa:"A simple programming method of state transition diagrams for parallel processings", Journal of information processing, Vol.5, No.3, pp.148–154 (Sep. 1982).

[2] 中村 八束,西田 秀次,不破 泰:"周期性部分の圧縮と無限規約コード利用の音声合成法", 電子通信学会論文誌D, Vol.J66–D, No.2, pp.135–142 (Feb. 1983).

[3] 中村 八束,山崎 靖夫,不破 泰:"機器組み込み用並行処理言語coroutine Cの設計とその処理系の実現", 情報処理学会論文誌, Vol.28, No.1, pp.74–81 (Jan. 1987).

[4] 不破 泰,中村 八束:"ウォッチドッグタイマの有効性に関する統計的考察", 電子情報通信学会論文誌D, Vol.J71–D, No.11, pp.2414–2423 (Nov. 1988).

[5] 不破 泰,中村 八束,清水 英孝:"2重のリング構造から成るキャンパスネットワーク –S–netについて–", 電子情報通信学会論文誌B, Vol.J71–B, No.12, pp.1672–1681 (Dec. 1988).

[6] 中村 八束,不破 泰:"情報数学教育における数学記述言語システムの利用", 電子情報通信学会論文誌A, Vol.J72–A, No.4, pp.711–717 (Apr. 1989).

[7] 不破 泰,中村 八束:"リング型ネットワークにおけるノードのハングアップ検出と復旧法について", 電子情報通信学会論文誌B–I, Vol.J72–B–I, No.9, pp.730–740 (Sep. 1989).

(Yasushi Fuwa and Yatsuka Nakamura:"Diagnosing hang–ups and automatic restoration for nodes in networks of ring topology", Electronics and Communications in Japan, Part 1, Vol.73, No.6, pp.73–84 (Jun 1990))

[8] 不破 泰,田坂 修二:"レジスタ挿入方式を用いたスロットリングの性能解析", 電子情報通信学会論文誌B–I, Vol.J73–B–I, No.3, pp.179–190 (March 1990).

[9] 不破 泰,田坂 修二:"レジスタ挿入型スロットリングにおけるスロットタイミングの高効率化とその性能解析", 電子情報通信学会論文誌B–I, Vol.J73–B–I, No.11, pp.813–824 (Nov. 1990).

[10] Yasushi Fuwa and Shuji Tasaka:"Register–insertion type slotted rings: a performance analysis", Proc. IEEE INFOCOM, pp.191–201 (April 1991).

[11] Yasushi Fuwa, Yatsuka Nakamura and Hirokuni Aiura:"Utilization of the shinshu university campus network –S–net–", IEICE Transactions on Communications, Vol.E74, No.9, pp.2756–2764 (Sep. 1991).

報文

[1] 中村 八束,中西 雅之,不破 泰:"パーソナルコンピュータの簡易ネットワーク装置とその応用について", 信州大学工学部紀要, No.53, pp.31–43 (Dec. 1982).

[2] 中村 八束,不破 泰,水野 貴博:"無手順通信方式を中心とするシステム統合型ネットワーク", 信州大学工学部紀要, No.58, pp.9–25 (Sep. 1985).

[3] 中村 八束,不破 泰,高野 洋之:"既約コード法による音声合成・編集システム", 信州大学工学部紀要, No.61, pp.1–7 (Feb. 1987).

[4] 中村 八束,不破 泰,牛丸 利治:"無手順型パーソナルコンピュータネットワークにおける資源共有システム", 信州大学工学部紀要, No.61, pp.9–18 (Dec. 1987).

[5] Yatsuka Nakamura, Yasushi Fuwa, Hidehiko Kitahara and Naohisa Morimoto:"A self-supporting quadruped walking-robot", Journal of the faculty of engineering, Shinshu University, No.63, pp.19–27 (Feb. 1988).

[6] Yasushi Fuwa, Yatsuka Nakamura and Yoshihiro Koshisaka:"An algorithm for solving alive sisuation pussles in GO game", Journal of the faculty of engineering, Shinshu University, No.65, pp.17–24 (Feb. 1989).

[7] Yatsuka Nakamura, Yasushi Fuwa and Hiroshi Imura:"A theory of finite topology and image processing", Journal of the faculty of engineering, Shinshu University, No.69, pp.11–24 (Sep. 1989).

学会発表

[1] 不破 泰,中村 八束,清水 英孝:"無手順型キャンパスネットワークS-netについて", 電子情報通信学会情報ネットワーク研究会 IN87–72 (Nov. 1987).

[2] 不破 泰,中村 八束:"ウォッチドッグタイマの有効性に関する統計的考察", 情報通信学会フォールトトレラントシステム研究会 FTS87–28 (Jan. 1988).

[3] 中村 八束,不破 泰:"情報数学教育に於ける数学記述言語システムの利用", 電子情報通信学会教育工学研究会 ET88–1 (Apr. 1988).

[4] 不破 泰,中村 八束:"リング型ネットワークにおけるノードのハングアップ検出と復旧法について", 電子情報通信学会論文誌B–I, Vol.J72–B–I, No.9, pp.730–740 (Sep. 1989).

[5] 不破 泰,中村 八束,新村 正明:"レジスタ挿入型スロットリングにおける同期の確立アルゴリズムについて", 電子情報通信学会情報ネットワーク研究会 IN89–68 (Sep. 1989).

[6] 中村 八束,不破 泰,森本 直久,力 尚宏:"Pascalプロシジャによる学生モデルと誤答分析システムの実現", 電子情報通信学会教育工学研究会 ET89–69 (Sep.

1989).

[7] 不破 泰,田坂 修二:"有限バッファを持つリング型ネットワークの性能解析–トークンリング–", 電子情報通信学会情報ネットワーク研究会 IN90–57 (Sep. 1990).

[8] 不破 泰,田坂 修二:"有限バッファ端末を持つリング型ネットワークの性能解析 –スロットリング–", 電子情報通信学会情報ネットワーク研究会 IN90–62 (Oct. 1990).

[9] 不破 泰,田坂 修二:"有限バッファ端末を持つリング型ネットワークの性能解析 –レジスタ挿入リング–", 電子情報通信学会情報ネットワーク研究会 IN90–68 (Dec. 1990).

[10] 中村 八束,不破 泰,今 利寿,力 尚宏:"算術式計算における誤答モデルとその評価", 電子情報通信学会教育工学研究会 ET91–85 (Oct. 1991).

[11] 中村 八束,不破 泰,西山 隆也:"多段順序回路における信号伝播のタイミング問題", 電子情報通信学会回路とシステム研究会 CAS91–85 (Nov. 1991).

付記　研究業績と本論文との関係

本論文の各章と研究業績の関係を以下に示す.

第1章: **論文**[1][5][11]
報文[1][2]
学会発表[1]

第2章: **論文**[8][9][10]
学会発表[7][8][9]

第3章: **論文**[8][9][10]
学会発表[5]

第4章: **学会発表**[7]

第5章: **学会発表**[8]

第6章: **学会発表**[9]

第7章: **論文**[8][9][10]
学会発表[7][8][9]

第8章: **論文**[7]
学会発表[4]

第9章: **論文**[4]
学会発表[2]