

名古屋大学大学院工学研究科  
博士課程(前期課程)

修士学位論文

教育用 LAN のモデル化と通信量解析  
に関する研究

平成 8(1995) 年 3 月

電気工学，電気工学第 2 及び電子工学専攻

石原 進



# 目次

<b>1</b>	<b>はじめに</b>	<b>3</b>
<b>2</b>	<b>ネットワーク性能評価の手法</b>	<b>7</b>
2.1	通信プロトコル	7
2.2	LAN 性能評価の手法	8
2.3	解析手法	9
2.4	シミュレーション技法	11
<b>3</b>	<b>ネットワークシミュレーションモデル</b>	<b>13</b>
3.1	モデルの概要	13
3.2	データリンク層のモデリング	18
3.2.1	IEEE802.3 CSMA/CD のアルゴリズム	18
3.2.2	CSMA/CD 状態遷移モデル	20
3.2.3	キャリアの検出	23
3.2.4	衝突の検出	23
3.2.5	次イベント発生時刻の決定	25
3.2.6	モデルの制限	27
3.3	TCP のモデリング	28
3.3.1	プロトコルの仕様	28
3.3.2	TCP モジュールの構成	32

## 2 目次

---

3.4	上位アプリケーションモデル . . . . .	35
3.5	アクセス集中モデル . . . . .	36
4	実験と考察	39
4.1	データリンク層以下のみのシミュレーション . . . . .	39
4.1.1	シミュレーションの条件 . . . . .	39
4.1.2	性能評価指標 . . . . .	40
4.1.3	シミュレーション結果と考察 . . . . .	41
4.2	情報処理教育環境におけるネットワークシミュレーション . . . . .	48
4.2.1	環境設定 . . . . .	48
4.2.2	性能評価指標 . . . . .	50
4.2.3	シミュレーション結果と考察 . . . . .	51
4.3	教育用 LAN における実システム測定 . . . . .	69
4.3.1	測定方法 . . . . .	69
4.3.2	測定結果と考察 . . . . .	73
5	まとめ	77
5.1	教育用 LAN の構成法 . . . . .	77
5.2	シミュレーションモデルの問題点と今後の展望 . . . . .	79
	謝辞	79
	参考文献	81

# 第 1 章

## はじめに

ローカルエリア・ネットワーク (LAN) の普及，およびワークステーション (WS) の低価格化に伴い，大学等情報処理教育センターのような集合情報処理教育施設においても数百台規模の分散 WS システムを導入する施設が増えている<sup>[1][2][3][4]</sup>．これらの施設ではシステムのすべてのマシンで利用者に同一の利用環境を提供するため，ネットワークファイルシステム (NFS) を用い，一台ないしは数台のファイルサーバ WS で数千人のユーザの個人データや共有データを一元管理している．

このような集合情報処理教育施設の LAN(以下教育用 LAN) の特徴は，時間的，局所的なネットワーク負荷の集中度が高いことである．授業利用時にクライアント WS が一斉に動作すると，全体として非常に大きなネットワークトラヒックが発生する．それらが単一のネットワークに集中すれば，そのネットワークを飽和させてしまうことは十分に予想できる．例えばプログラミングの授業の場合，教官の指示の直後に学生が一斉にコンパイルを始めるという状況が起きる．このとき，コンパイルプログラム，ソースプログラム，ライブラリ等がファイルサーバにおかれているとすると，一時的にファイルサーバへのネットワークアクセスが集中することになる．そうなると，WS 間の通信が阻害され，システム全体としての性能が著しく低くなる恐れがある．

一方，一般の実務や研究用のシステムではネットワーク上の資源へのアクセス

#### 4 第1章 はじめに

---

は時間的に分散しており，ネットワーク負荷が極端に集中することは稀である<sup>[5]</sup>．従来の NFS<sup>[6][7]</sup> はこれら一般のシステムを想定して設計されており，教育用 LAN のように極端にネットワーク負荷が集中するような LAN に対する対策は十分ではない．このため，教育用 LAN で十分な性能を得るためには，教育用 LAN の特性を踏まえて適切な方法で LAN を構成すること，および専用の NFS を構築することが必要である．

また，近年のマルチメディアの普及に伴い，一般の LAN においても画像や動画などの大容量転送が行なわれるようになり，ネットワーク負荷は増大する傾向にある．このため，高負荷 LAN の適切な構成手法の確立は重要な問題である．

本論文では高負荷及び高負荷集中度な LAN においても高精度に通信量解析が可能な性能評価モデルを提案する．更に，教育用 LAN などの高負荷 LAN の適切な構成法を決定するため，教育用 LAN の実利用環境における負荷集中を想定したシミュレーションにより負荷集中時の LAN 通信量を評価し，モデルの検証を行った．

本論文で提案するモデルは OSI 参照モデルの各層に対応したモジュール化がされており，既存のプロトコルのみならず，新規に設計するプロトコルの評価も可能であるという特徴を持つ．

LAN の性能評価モデルは従来，解析手法<sup>[8][9][10][11]</sup> やシミュレーション手法<sup>[12][13][14]</sup> によるモデルが提案されている．解析手法によるモデルでは，簡単化のためパケットの生成率やネットワーク遅延が単純な確率過程に従うものとしてモデル化している．このような手法は負荷が安定している時の平均的なネットワーク性能を予測するには適しており，計算時間も短くてすむ．しかし，極端な負荷の集中や負荷の変動がある場合に正確な評価結果が得られる保証はない．また，クライアント・サーバモデルのようにステーション間のメッセージ依存性がある場合や，ネットワーク上でメッセージが失われるときの再送などを考慮に入れた場合にはモデル化が困難である．

シミュレーション手法では，その記述性の高さから通信プロトコルの動作を忠実にモデル化したものがあるが，単一のプロトコル階層のみをモデル化したものが大部分であり，複数のプロトコル階層を通した通信をモデル化したものはない．

本論文では複数の階層のプロトコルを詳細にモデル化したネットワークノードのモデルを，イベント駆動で並列に動作させるシミュレーションモデル<sup>[15][16][17]</sup>を提案する．本モデルでは基本モデルとしてトランスポート層プロトコルである TCP(Transmission Control Protocol) およびデータリンク層プロトコルの IEEE 802.3 標準の CSMA/CD(Carrier Sense Multiple Access with Collision Detection) の動作を厳密にモデル化し，この基本モデル上に上位のアプリケーションの動作を記述するモデルを組み合わせる．これにより，実利用環境におけるアプリケーション利用時のネットワーク特性を正確に評価できると考える．また，NFS 等のネットワークシステム評価のための基本モデルとしての利用も可能である．なお NFS 設計のための基本モデルを設計することは本研究の目的の一つである．

以下，第 2 章で負荷集中が起きる LAN の性能評価手法について検討を行う．次に第 3 章で今回設計した LAN シミュレーションモデルについて説明する．第 4 章で，実システムでの測定結果および従来手法によるシミュレーション結果と比較することにより，本モデルを用いて行ったシミュレーションモデルの信頼性を検証する．更に，高速 LAN を教育用 LAN に適用した場合のシミュレーション結果について検討を行う．最後に第 5 章で教育用 LAN の適切な構成方法についての検討とシミュレーションモデルの応用分野について述べる．



## 第 2 章

# ネットワーク性能評価の手法

### 2.1 通信プロトコル

OSI(Open Systems Interconnection) では通信プロトコルの 7 階層参照モデル (表 2.1) を提唱している [18] . プロトコルは 7 つの階層に分かれており , それぞれ同じ階層のプロトコル同士が通信を行う . これらのプロトコルの動作のうち , LAN 上の 2 つのステーションのアプリケーション同士の通信時間に影響を与えるのは , (i) 物理層である伝送媒体の伝送速度と (ii) データリンク層が行う物理媒体へのアクセス方式 , (iii) トランスポート層またはセッション層で行われるメッセージ再送信の動作 , および (iv) アプリケーション独自の動作である . また , プロトコルの動作だけでなく計算機の処理能力も通信時間を左右する一因である .

表 2.1: ISO/OSI 参照モデル

	層の名前	機能	プロトコルの例	
7	アプリケーション層	アプリケーション特定の機能	NFS	FTP, TELNET
6	プレゼンテーション層	データ変換	XDR	
5	セッション層	確実な転送	RPC	
4	トランスポート層	プログラム・プログラム間転送	TCP, UDP	
3	ネットワーク層	ホスト・ホスト間転送	IP	
2	データリンク層	ノード・ノード間転送	Ethernet, IEEE802.3 std.	
1	物理層	物理的な接続インターフェース	同軸ケーブル, 光ファイバ	

LAN の性能評価を正確に行うためには上記の要因すべてに対し検討を行わなければならない。

### 2.2 LAN 性能評価の手法

LAN の性能評価を行なうための手法を大別すると，以下の3通りが考えられる [19] 。

- (1) 解析手法．トラヒック理論や待ち行列理論など．
- (2) シミュレーション技法．
- (3) 実システムによる計測

システム設計段階では，(1) あるいは (2) を用いる必要があり，システム導入後では (1) から (3) までのすべてが選択の対象になる．本研究では教育用 LAN の最適構成法の決定のためのネットワーク評価を目的としているから，(1)(2) のいずれかの手法を用いることになる．しかし，(a) 教育用 LAN の実利用時におけるユーザの挙動に関する測定例が少ないこと，(b) 解析手法やシミュレーション手法においてはプログラムのバグは不可避のものであることから，現状の実システムでの計測結果との比較も重要である．

教育用 LAN のような負荷が集中する LAN では，物理媒体アクセスにおける遅延が増大する．特にデータリンク層以下のプロトコルに Ethernet や IEEE 802.3 標準の CSMA/CD 方式を用いた場合，フレーム送信エラーが発生してメッセージが送信されないこともある．従って，信頼性を保証した通信を行うアプリケーションの場合，メッセージの再送が頻発することが予想され，通信時間に大きな影響を与えと考えられる．このため，このような LAN の性能評価を行うには，解析手法，シミュレーション手法のどちらの場合でもメッセージの再送動作およびサーバ・クライアント間のメッセージ依存性をモデルに組み込む必要がある．また，実際のア

アプリケーションの利用時のネットワーク性能を評価するには、ユーザの動作の特性を反映できることも必要である。

以上の条件を踏まえて、従来の解析手法およびシミュレーションモデルによる LAN 性能評価モデルの例について示し、教育用 LAN の評価に適した性能評価手法について検討する。

## 2.3 解析手法

一般には、解析手法の方がシミュレーションよりも評価に要する時間が短い。しかし解析解を得るためには、さまざまな仮定の導入によるモデルの近似および単純化が行われている。

Drakopoulos<sup>[8]</sup> は、クライアント・サーバ記憶システムを閉鎖形待ち行列網でモデル化している。このモデルではイーサネットとトークンリングのネットワークにおいて、パケット到着率がポアソン分布に従うと仮定し、サービス時間をネットワーク平均遅延とした待ち行列でネットワークをモデル化している。このため、長時間の平均的なシステムの性能評価を行なうには適しているが、教育用 LAN で起こるような過渡的な負荷変動時のシステムの挙動を評価するには適していない。

Ibe<sup>[9]</sup> は、ペトリネットにより多数台のクライアントと 1 台のサーバから構成されるクライアント・サーバシステムのモデル化を行っている。このモデルではペトリネットを用いることにより、クライアントのサーバへのリクエストとサーバリプライの依存性をモデルに組み込んでいる。データリンク層以下のプロトコルにはイーサネットとトークン・リングを仮定しているが、イーサネットでの送信エラーはモデルに組み込まれていない。

Bolot<sup>[10]</sup> は、マルコフモデルを用い、混雑したネットワーク上での 2 つのステーション間での通信におけるトランスポート層プロトコルの性能評価を行っている。トランスポート層のプロトコルには、TCP のようにスライディングウィンド

ウ方式を用い確認応答つき再送信の動作を行うものを仮定している。マルコフモデルの状態は、サーバとクライアントの送受信メッセージ数および伝送路上のメッセージ数の組合せとして表される。このため、送受信メッセージ数が多くなったり、プロトコルの実装方式を複雑にすると状態数が莫大になり、数値解を求めるのが困難になる。論文で仮定されたプロトコルは単純化されており、現在実用になっているプロトコル性能改善手法を用いた場合については評価結果が得られていない。また、データリンク層以下のプロトコルの動作は伝送路上に送信されたメッセージ数の単純な関数としてモデル化されている。

Murata<sup>[11]</sup>らは、複数のプロトコル階層を通して通信を行うネットワークをモデル化している。モデルはデータリンク層以下のプロトコルを表すサブモデルと、トランスポート層以上のプロトコルの動作を表すサブモデルによる2階層のモデルとして構成される。すなわち、IEEE802.5標準のようなトークンパッシング方式のデータリンク層プロトコルを、複数待ち行列による周期的サービスモデルとしてモデル化し、スライディングウィンドウ方式を用い確認応答をもつトランスポート層プロトコルを閉鎖形待ち行列網でモデル化している。このモデルは、全二重通信やクライアント・サーバシステムなどの様々な通信方式に応用可能である。

これら解析的手法によるモデルは、負荷が一定の場合の平均的なネットワーク性能値を数値解法で求めるものである。このため、ユーザの動作による負荷の変動が起きた場合のアプリケーション利用時のネットワーク性能を評価するのは困難である。また、複数のプロトコルを通した通信をモデル化する場合、各プロトコルの動作すべてにわたってモデル化を行うと、解析手法は複雑にならざるを得ない。このため、計算を行う際には計算機のメモリの制限を受ける場合が多くなり、前述の例にある通り、モデル化するプロトコルの動作を単純化したり、ネットワークの構成を限定する必要がある。従って、複数のプロトコルを利用したネットワークの構成手法を検討するには不十分である。

## 2.4 シミュレーション技法

前項での検討により，解析モデルによる性能評価は困難であるという見地から，モデル記述性の高いシミュレーションによる性能評価が有用であると考える．

O'Reilly<sup>[13]</sup>らは，データリンク層プロトコルにイーサネットで利用しているCSMA/CD方式を用いた場合のネットワーク評価モデルを提案している．このモデルではネットワーク上のノードをグループ化することにより，特にノード数が多い場合のシミュレーションの高速化とプログラム実行時に必要な記憶容量を削減している．伝送路上でのメッセージの衝突や送信エラーの発生は，動的に計算される確率に従った乱数により決定される．シミュレーション結果は，このような確率計算を行わない厳密にプロトコルを記述したシミュレーションモデル場合のシミュレーション結果と完全に一致している．しかし，各ノードでのパケットの発生はポアソン過程に従うことを前提としているため，データリンク層より上位のプロトコルの動作をモデルに組み込むのは困難である．

Marino<sup>[12]</sup>らは，CSMA/CD方式の動作を完全に記述した，高速なシミュレーションモデルを提案している．このモデルでは各ノードでのパケット送信直後に次のパケット発生時刻をスケジュールすることにより，高速化を実現している．シミュレーションの結果は実測値および<sup>[13]</sup>のモデルの結果と完全に一致している．パケット発生プロセスは，パケットの送信終了直後に次のパケット送信時刻が決定できるものならば種類を問わない．しかし，実際にはデータリンク層より上の階層のプロトコルがパケット送信要求を出す時刻は，上位アプリケーションが通信する相手からの受信パケットに依存する．このためこのモデルでは上位層プロトコルの動作を組み合わせたシミュレーションを行うのは困難と考える．

従来のモデルはデータリンク層のみをモデル化したものが大部分であり，複数の通信プロトコルの相互作用を考慮に入れたモデルはない．そこで本研究では，データリンク層以下のプロトコルとトランスポート層のプロトコルのモデルをそれぞれ

## 12 第2章 ネットワーク性能評価の手法

---

れモジュール化して設計し，これらの二つの組合せで構成されるネットワーク上のノードのモデルをイベント駆動で並列に動作させるシミュレーションモデルを設計した．各プロトコルの動作は厳密にモデル化する．また，ノード間のメッセージをモデル上のノード間で受渡しすることにより，サーバ・クライアント間のメッセージ依存性を完全にモデル化可能とした．このようなモデルに上位層のアプリケーションの動作および計算機の処理能力を記述するモデルを組み合わせることにより，実システムの利用時を想定したシミュレーションが可能である．

## 第 3 章

# ネットワークシミュレーションモデル

本章では教育用 LAN の具体的モデル化手法について述べる．はじめにネットワーク通信プロトコルのモデル化について述べ，次に教育用 LAN のネットワークアクセス集中のモデルについて述べる．

### 3.1 モデルの概要

本手法ではネットワークに一時的に急激な負荷集中が起きた場合のネットワーク通信量を正確に予測するため，対象とする通信プロトコルのアルゴリズムに忠実に従って動作するネットワーク上のノードのモデルを仮定し，それらを並行して動作させる．シミュレーションは，経過時間を各ノードにおける次イベントの発生までの時間とするイベント駆動方式で行う．

今回のモデルでは通信プロトコルとして，現在一般に広く普及している，TCP/IP (Transmission Control Protocol / Internet Protocol)<sup>[20]</sup> と IEEE802.3 標準の伝送媒体アクセス方式を仮定し (表 3.1 参照)，ISO/OSI 参照モデルの物理層からトランスポート層までのプロトコルを基礎モデルとしてモデル化する．この基礎モデルに上位アプリケーションのモデルを組み合わせることにより，種々のプロトコルの組み合わせに応じたシミュレーションシステムを構築することが可能である．なお，ここでは TCP/IP と，IEEE 802.3 標準を実例として用いているが，本論文で提案

表 3.1: 基礎モデルで仮定したプロトコル

階層	プロトコル
トランスポート層	TCP
ネットワーク層	IP
データリンク層	IEEE 802.3 標準 CSMA/CD
物理層	同軸ケーブルまたはより対線

するシミュレーションモデルの基本的枠組みは、これに限定されるものではない。

各通信プロトコルのモデル化手法について以下に概略を示す。

上位アプリケーションがトランスポート層のプロトコルとして TCP を利用する場合、TCP が通信の信頼性を保証するので、上位アプリケーションはデータの受信と送信を行うのみであり、その動作自体がデータ転送時間に及ぼす影響は少ない。このため、上位アプリケーションの動作は (i) TCP のフロー制御の動作に影響を与える受信バッファ内のデータの処理時間と (ii) 送信バッファへのデータの追加の動作のみをモデル化すればよい。一方、トランスポート層のプロトコルに UDP (User Datagram Protocol)<sup>[21]</sup> を用いるものは、UDP が通信の信頼性を保証していないため、アプリケーション自身が通信の信頼性を保証する動作をしたり、データが欠けることを前提とした動作をする。この動作はアプリケーションに依存するため、今回はこのようなアプリケーションの利用は考えていない。

トランスポート層とネットワーク層のプロトコルの動作は以下のように説明できる。トランスポート層の TCP は、信頼性の高い通信を実現するためにパケット順番制御、メッセージ再送信、フロー制御、輻輳制御を行う。このため、過渡的な負荷集中および遅延の変動が生じるネットワークでは TCP の動作がデータ転送時間に与える影響は大きい。ネットワーク層の IP は、パケットの転送経路を決定し、目的のホストまでパケットを送る機能を提供するが、この動作はデータリンク層で規定される 1 セグメント内の通信では通信時間に大きな影響を与えない。通信時間

に影響を与えるのはパケットに含まれる IP ヘッダの長さだけである。

以上の理由から，トランスポート層とネットワーク層のプロトコルが送信時間に与える影響は TCP の動作に集約される．従って，モデルではこれらをまとめて一つのモジュール (TCP モジュール) としてモデル化する．

IEEE 802.3 標準のデータリンク層プロトコルは，CSMA/CD 方式により物理媒体にビット列の送信，衝突検出および送信再試行を行う．これらの動作の結果，ネットワーク遅延およびパケットの消失がおきる．ビットレートおよび信号速度は，物理層の仕様によって決定される．これらデータリンク層以下の機能を一つのモジュール (イーサネットモジュール) としてモデル化する．ここでイーサネットの名前を使用したのは，IEEE 802.3 標準がイーサネットの技術を基礎としたものであり，一般に両者をまとめてイーサネットと呼ぶ場合が多いためである．

図 3.1 に本モデルの概念図を示す．クライアント WS のモデルはそれぞれ TCP モジュールとイーサネットモジュールにより構成される．また，サーバ WS のモデルは一つのイーサネットモジュールと，各クライアントに対応した数の TCP モジュールから構成される．サーバ WS モデル上の TCP モジュールはそれぞれ対応する通信先のクライアント WS のモデルの TCP モジュールと通信を行う．

各モジュールはモジュール内部および外部で計算されるイベントにより動作を開始する．モジュールが動作した結果として，そのモジュールおよび他のモジュールでの次のイベントとその発生時刻が生成される．時刻  $t$  でのすべてのモジュールの動作が終了後，全モジュールでの次イベント発生時刻の最小値まで時間を進める．

イーサネットモジュールの動作は対応する TCP モジュールによる送信要求によって引き起こされる．CSMA/CD アルゴリズムに忠実に従ってフレームの送信動作を行うため，イーサネットモジュールは自分のノードと他のノードのイーサネットモジュールの状態を比較することにより，自分のノードの動作と次イベント発生時刻を決定する．これを積み重ねることにより一つのフレームの送信を完了す

る。この結果，相手ノードへのフレーム到着時刻が決定され，相手ノードの TCP モジュールでの受信イベントを引き起こす。

TCP モジュールの動作は，イーサネットモジュールによって生成される受信イベント，および上位アプリケーションモデルが生成するデータ送信，受信イベント，および TCP モジュール内部で生成される送信イベントにより引き起こされる。イベントが発生すると，TCP モジュールはモジュール内部処理を行い，現時刻で送信するか否かを判断する。送信すると判断した場合，イーサネットモジュールでの送信イベントを生成する。また，送信するしないに関わらず，TCP モジュール内部の次イベント発生時刻を決定する。

以下次節より，イーサネットモジュール，TCP モジュールおよび上位アプリケーションのモデルの詳しいモデル化の方法について説明する。

なお，シミュレーションモデルのプログラムは C 言語で作成した。

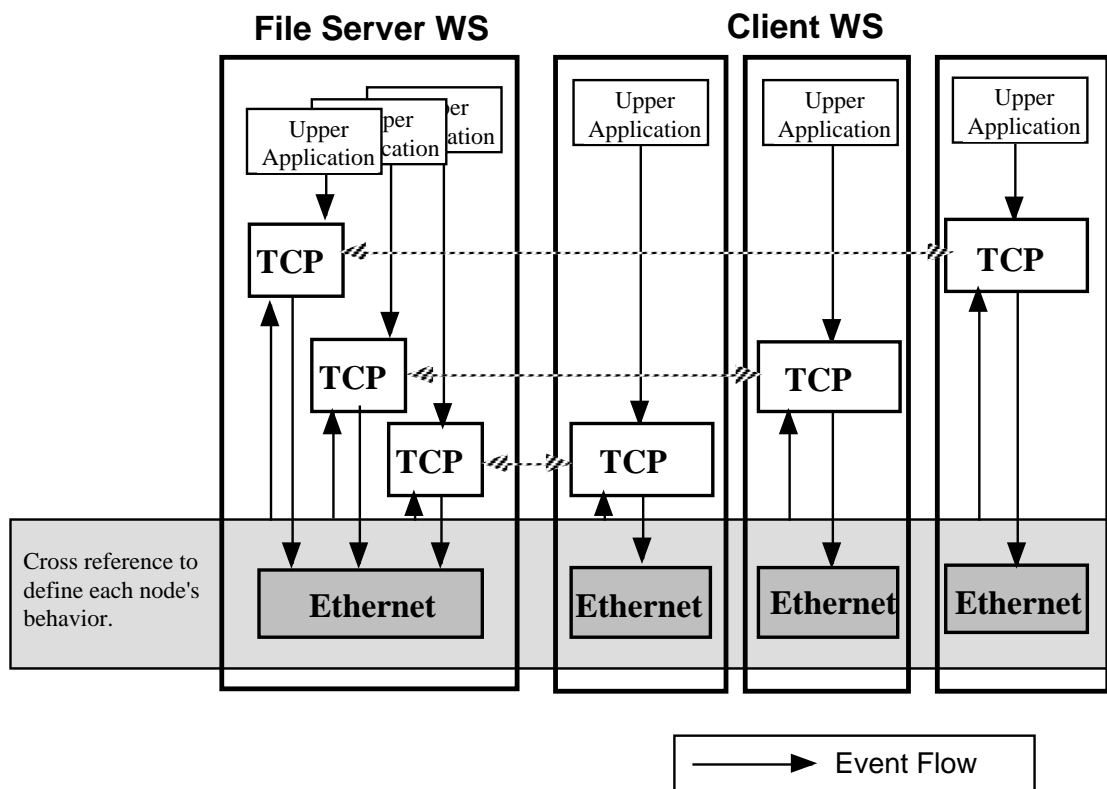


図 3.1: シミュレーションモデルの概念図  
 Fig. 3.1 Overview of simulation model

## 3.2 データリンク層のモデリング

### 3.2.1 IEEE802.3 CSMA/CD のアルゴリズム

データリンク層のプロトコルのモデル化について説明する前に、IEEE802.3 標準の物理媒体アクセス方式である CSMA/CD 方式のアルゴリズム<sup>[22]</sup>について説明する。

CSMA/CD とは共通の伝送路を 2 つ以上の複数のノードが共用しながら、同時には 1 つのノードしかチャンネルをアクセスできないときに、どのノードにアクセス権を与えるかを決める機構である。

CSMA/CD 方式のアルゴリズムを図 3.2 に示す。

ノードがフレームの送信を開始する時、伝送路上に他のノードが発したキャリアの存在を検査する (Carrier Sense)。他のノードが発したキャリアが検出されなくなってからフレーム間ギャップの 96 ビット時間が経過していたら、フレームを送信する。そうでなければキャリアが消えるのを待ち、フレーム間ギャップの時間経過後フレームを送信する。

共通伝送路の空きを検出するだけでは、複数のノードが同時にフレームの送信を開始すると衝突が発生し、正常なフレームが宛先ノードに送達されない。このため各ノードはフレーム送信中には常に衝突を監視 (Collision Detection) している。衝突が検出されると直ちにフレームの送信を休止し、ほかのノードに衝突発生を正しく知らせるため 32 ビットの JAM 信号を送信する。このとき衝突の回数が 15 回以下なら、二進指数バックオフアルゴリズムに従ってフレーム再送信までの待ち時間を計算し、この時間だけ待機する。これは衝突後の再送までの間隔を再送回数に対して指数関数的に増加させることにより、衝突したフレームの再送頻度を下げる手法である。衝突回数が 16 回の場合は送信失敗となる。

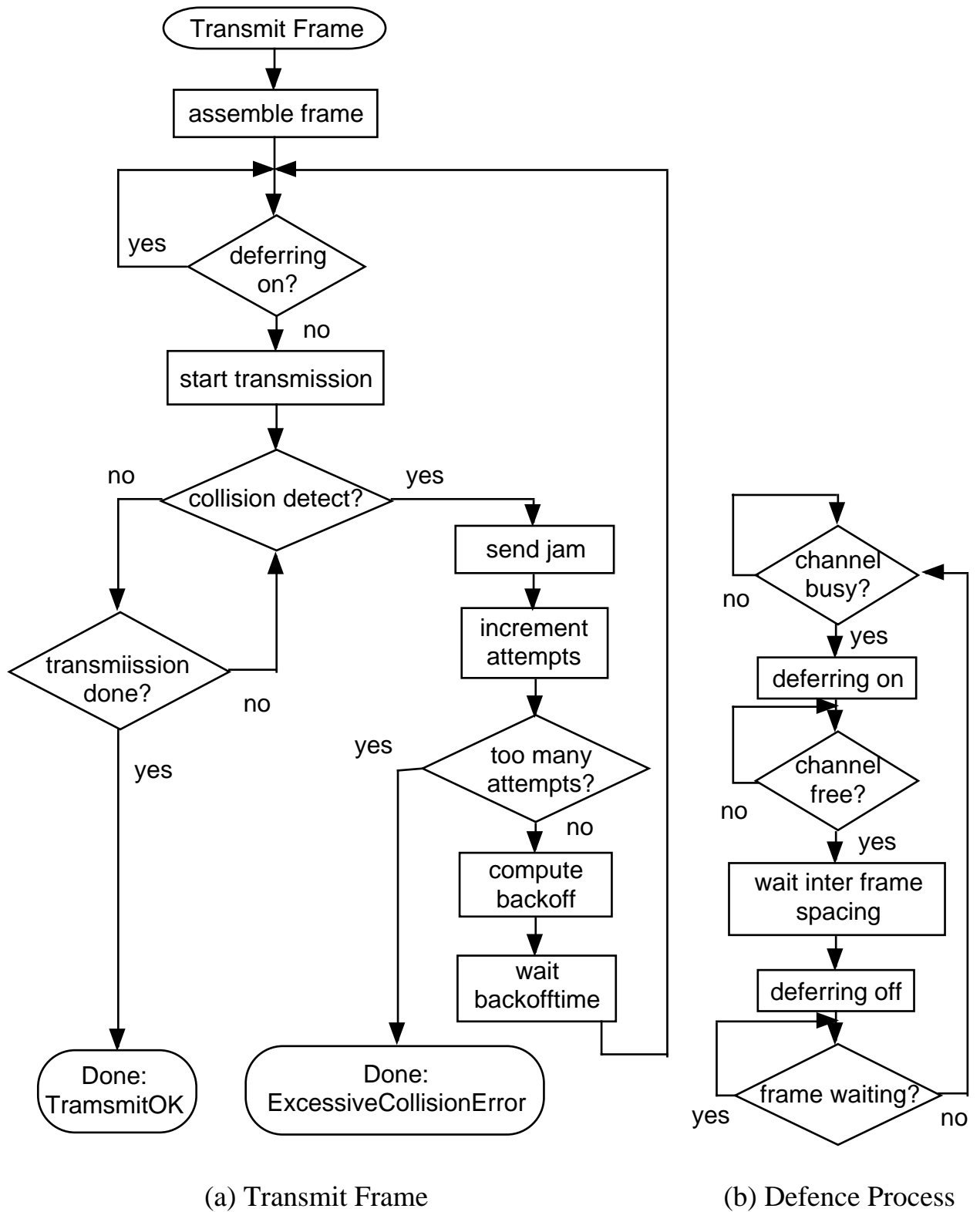


図 3.2: IEEE 802.3 標準の CSMA/CD のアルゴリズム  
 Fig. 3.2. CSMA/CD algorithm of IEEE Standard 802.3.

### 3.2.2 CSMA/CD 状態遷移モデル

CSMA/CD 方式では伝送路上のキャリアの有無がその動作に直接影響を与える。従来のシミュレーションモデルでは、伝送路上のキャリアの検出および衝突の発生を確率モデルとしてモデル化していた。しかし本モデルでは CSMA/CD の動作を厳密にモデル化することを目的としている。そこで CSMA/CD アルゴリズムに従う各ノードの動作を状態遷移モデルとしてモデル化し、各ノードの位置におけるキャリアの存在を独立に判定できるようにした。

モデル上のノード  $i(= 1, \dots, N)$  は、それぞれ送信状態  $S_i \in \{SLP, CS, PR, TR, CD, WT\}$  ノード  $i$  での最も新しいキャリア発生時刻  $t_{cp_i}$ 、キャリア消滅時刻  $t_{cd_i}$ 、および衝突回数  $n_{col_i}$  を保持している。 $S_i \in \{SLP, CS, PR, TR, CD, WT\}$  は IEEE802.3 標準で定義される状態ではなく、本モデルで規定したものである。また、 $t_{cd_i}$  は実際にノードからの信号の送信が終了した時刻ではなく、送信終了後フレーム間ギャップの時間が経過した時刻を意味している。これらをまとめてノード状態  $M_i = \{S_i, t_{cp_i}, t_{cd_i}, n_{col_i}\}$  と表記する。更に、ネットワーク全体の状態を示す変数として、衝突の可能性のある送信中ノードの数  $n_{pr}$  を定義する。

ノード  $i$  は他のノード  $j(= 1 \dots i - 1, i + 1 \dots N)$  の状態  $M_j$  と  $M_i$  および  $n_{pr}$  の比較により、次のイベント発生時刻  $t_{n_i}$  を決定して状態遷移を行う。図 3.3 に状態遷移図、表 3.2 に遷移条件を示す。

以下、送信状態  $S_i$  の詳細について説明する。

SLP(Sleep) 送信バッファが空であり、何も送信をしていない状態。送信バッファにデータが存在すると CS に遷移する。

CS(Carrier Sense) キャリアセンスを行う。伝送路が空であれば PR に遷移する。

PR(Preamble) 送信状態 PR と TR は伝送路上にフレームを送信している状態で

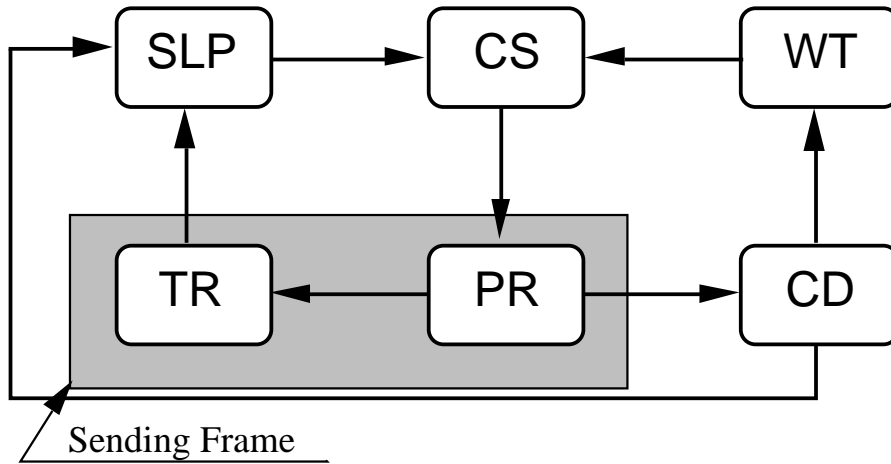


図 3.3: イーサネットモジュールの状態遷移図  
 Fig. 3.3 State diagram of Ethernet module.

表 3.2: イーサネットモジュールの状態遷移条件と動作

Current State	Event	Action	Next State
SLP	送信データ存在	$n_{col_i} \leftarrow 0$	CS
CS	伝送路空き	$t_{cp_i} \leftarrow t$ $n_{pr} \leftarrow n_{pr} + 1$	PR
PR	衝突発生時間経過 衝突なし	$n_{pr} \leftarrow n_{pr} - 1$	TR
	衝突発生時間経過 衝突あり	$n_{pr} \leftarrow n_{pr} - 1$ $n_{col_i} \leftarrow n_{col_i} + 1$ $T_{bi}$ 計算	CD
TR	フレーム送信時間経過	$t_{cd_i} \leftarrow t$	SLP
CD	ジャム信号送信時間経過 $n_{col_i} = 16$	$t_{cd_i} \leftarrow t$	SLP
	ジャム信号送信時間経過 $n_{col_i} < 16$	$t_{cd_i} \leftarrow t$	WT
WT	待ち時間経過		CS

ある．このうち PR は送信中に衝突が発生する可能性がある時間帯として定義する．この状態にあるノードが実際に送信しているのは，衝突が起きない場合フレーム先頭の一部であり，衝突が起きた場合にはフレームの一部と JAM 信号の一部である．

この状態にあるノードが存在する場合，そのノードが送信を開始していても，ノードの発するキャリアが伝送路全体に伝播してはいない．このため，他のノードが伝送路が空いていると判断して，送信を開始するノードが存在する可能性がある．これはすなわち衝突が起きる可能性があることを意味している．

衝突判定(後述)を行なった後，衝突が検出されなければ TR に遷移する．衝突が検出されれば，送信を再試行するまでの待ち時間を二進指数バックオフアルゴリズムに従って計算し CD に遷移する．

TR(Transmission) フレームの送信を行う．衝突の可能性はない．フレームの送信が終了すると SLP に遷移する．

CD(Collision Detect) ジャム信号を送信する．ジャム信号の送信が終了した後，このフレームの送信において衝突回数が  $n_{col} < 16$  の時は，WT に遷移する． $n_{col} = 16$  の時はこのフレームの送信をあきらめ，SLP に遷移する．

WT(Wait) 送信を再試行する時刻まで待っている状態．待ち時間が経過すると CS に遷移する．

実際には，(i)TR → SLP と CD → SLP の遷移後に送信バッファにデータがある，(ii)WT における再送待ち時間がゼロ，の場合には状態遷移はシミュレーションの経過時間ゼロの間に複数の状態遷移が連続して行われる．

### 3.2.3 キャリアの検出

キャリアの検出は状態 CS で行われる。キャリア検出を行うノードを  $i(= 1, 2, \dots, N)$  ,  
その他のノードを  $j(= 1, 2, \dots, i-1, i+1, \dots, N)$  とする。

他のノード  $j$  の送信状態  $S_j \in \{\text{TR}, \text{CD}\}$  のときは、ノード  $j$  の発したキャリアは伝送路全体に伝播しているため、明らかに伝送路が他のノードに使用されている。一方  $S_j \in \{\text{SLP}, \text{CS}, \text{PR}, \text{WT}\}$  の場合、ノード  $j$  の発したキャリアがノード  $i$  で検出されるか否かは、ノード  $j$  がいつキャリアを発し、消滅させたかに依存する。 $S_j = \text{PR}$  のときには、ノード  $j$  が  $t_{cp_j}$  にフレーム送信を開始していても、ノード  $i, j$  間の伝播遅延  $\tau_{ij}$  により、 $t_{cp_j} + \tau_{ij}$  まではノード  $i$  にキャリアが伝播していない(図 3.4)。また、 $S_j \in \{\text{SLP}, \text{CS}, \text{WT}\}$  のときには、ノード  $j$  が現在送信中でなくても、最後にキャリアを消滅させた時刻  $t_{cd_j}$  から  $\tau_{ij}$  が経過していなければ、ノード  $i$  にはキャリアが存在することになる(図 3.5)。

従って、現在時刻  $t$  と他のノード  $j$  の状態  $M_j$  に関して式 (3.1) によりノード  $i$  に他のノード発生したキャリアが存在し、伝送路が使用中であることを判断できる。

$$\begin{aligned} & \text{ノード } i \text{ でキャリアが存在する} \\ = & \left\{ \begin{array}{l} S_j \in \{\text{TR}, \text{CD}\} \text{ を満たす } j \text{ が 1 つ以上存在する} \\ \text{or} \\ S_j = \text{PR} \text{ かつ } t_{cp_j} + \tau_{ij} \leq t \text{ を満たす } j \text{ が 1 つ以上存在する} \\ \text{or} \\ \text{すべての } j (\neq i) \text{ において } S_j \in \{\text{SLP}, \text{CS}, \text{WT}\} \text{ であり} \\ t \leq t_{cd_j} + \tau_{ij} \text{ を満たす } j \text{ が 1 つ以上存在する} \end{array} \right. \quad (3.1) \end{aligned}$$

### 3.2.4 衝突の検出

CSMA/CD アルゴリズムで衝突が発生するのは、あるノード  $i$  が送信を開始しているがそのキャリアが他のノード  $j$  が検出する前にノード  $j$  が送信を開始した場合である。最も早く送信を開始したノードの発したキャリアが全てのノードに伝

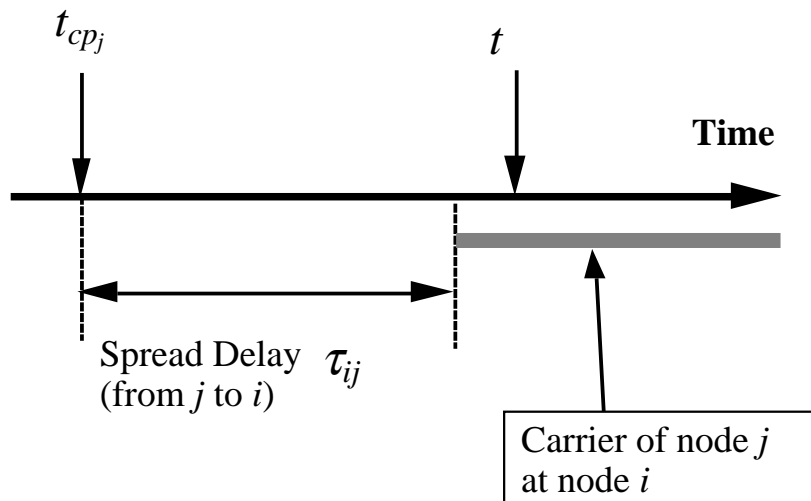


図 3.4: キャリアの検出 : 他のノードが送信中 ( $S_j = PR$ )  
 Fig. 3.4 Carrier sense: other nodes are transmitting

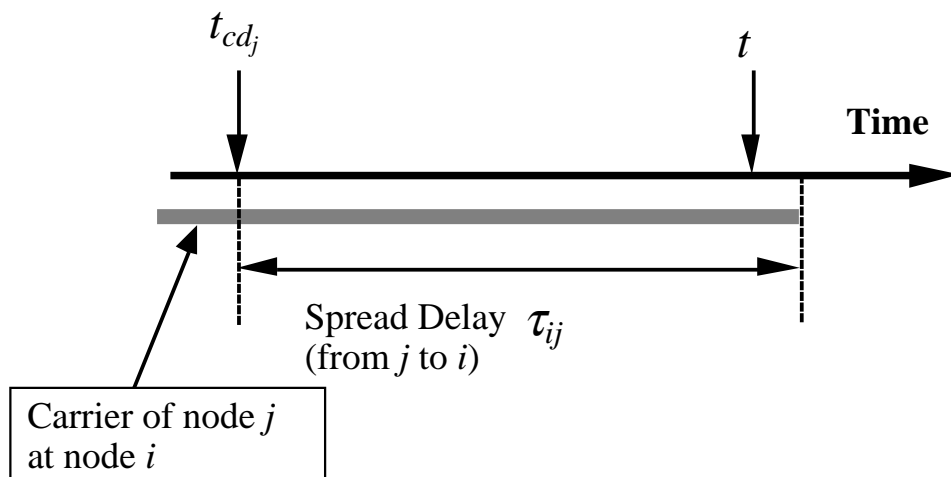


図 3.5: キャリアの検出 : 他のノードの送信終了後 ( $S_j \in \{SLP, CS, WT\}$ )  
 Fig. 3.5. Carrier sense: after other nodes' transmission.

わる間に他のノードが送信を開始しなければ、衝突は起こらない。従って衝突の危険があるのは、最も早く送信を開始したノード  $k$  の送信開始  $t_{cp_k}$  から、 $k$  が発したキャリアが全てのノードに伝わるまでの時刻  $t_d = t_{cp_k} + T_{d_k}$  までである。ここで  $T_{d_k}$  はノード  $k$  と他のノードとの間の最大信号伝播遅延であり、 $T_{d_k} = \max(\tau_{kj}; j = 1, 2, \dots, k-1, k+1, \dots, N)$ 。この時間内にノード  $k$  を含む 2 つ以上のノードが送信を開始していれば衝突が起きている。

衝突があったかどうかの判断は式 (3.2) により行う。

$$\text{at } t = t_d \begin{cases} \text{if } n_{pr} > 1 & \text{then 衝突発生} \\ \text{otherwise} & \text{衝突なし} \end{cases} \quad (3.2)$$

実際に各ノードで衝突が検出される時刻は、自分のノードが送信中に他のノードが発したキャリアが到達する時刻である。すなわちノード  $i$  における衝突検出時刻  $t_{col_i}$  は次式で与えられる。

$$t_{col_i} = \min_{j \neq i, S_j = PR} \{t_{cp_j} + \tau_{ij}\} \quad (3.3)$$

### 3.2.5 次イベント発生時刻の決定

イーサネットモジュールの各状態における次イベント発生時刻  $t_{n_i}$  は次のようにして計算される。

SLP 送信バッファにデータがあるかどうかは、TCP モジュールの動作に依存するため、送信バッファ内のデータの有無の判断は同時刻のイーサネットモジュールの動作に先だって行なわれる TCP モジュールの動作終了後に行う。従って  $t_{n_i} = \infty$  とする。

CS 自分のノード  $i$  の位置において他のノード  $j$  の発生したキャリアがなくなるまでの時間を計算する。 $S_j \in \{PR, TR, CD\}$  であれば、現時点でノード  $j$  が出しているキャリアが消滅したことがノード  $i$  に伝わるのは、ノード  $j$  におけるイベ

ント発生後,すなわち状態遷移が起きた後である.従って, $S_j \in \{\text{PR}, \text{TR}, \text{CD}\}$ の状態にあるノード  $j$  が存在するときには  $t_{n_i} = \infty$  となる.自分以外のすべてのノード  $j$  の状態が  $S_j \in \{\text{SLP}, \text{CS}, \text{WT}\}$  であれば, $t_{n_i}$  は式 (3.4) で与えられる.

$$t_{n_i} = \min_{j \neq i} \{t_{cd_j} + \tau_{ij}\} \quad (3.4)$$

PR 最も早く送信を開始したノード  $k$  が発したキャリアが全てのノードに伝わるまでの時刻  $t_d$  まで経過させる.すなわち,

$$t_{n_i} = t_d = t_{cp_k} + T_{dk} \quad (3.5)$$

TR フレームの長さに応じた送信時間を経過させる.従って,現在送信中のパケット送信に必要な時間とフレームギャップの時間の和を  $T_{p_i}$  とすると, $t_{n_i}$  は式 (3.6) で与えられる.ここで  $t_{ts_i}$  はノード  $i$  で  $\text{CS} \rightarrow \text{PR}$  の状態遷移が起きた時刻である.

$$t_{n_i} = t_{ts_i} + T_{p_i} \quad (3.6)$$

また,送信したフレームの到着時刻  $t_{rcv_j}$  はフレーム送信終了後からノード間遅延が経過した時だから,式 (3.7) で計算できる. $t_{rcv_j}$  はTCP モジュールでの受信イベント発生時刻を決定する.

$$t_{rcv_j} = t_{ts_i} + T_{p_i} + \tau_{ij} \quad (3.7)$$

ここで  $j$  はフレームの送信先ノードを示す.

CD 次イベントの発生は衝突検出後の JAM 信号の送信が終了したときである. JAM 信号の送信に要する時間とフレームギャップの時間の和を  $T_{jam}$  とすると, $t_{n_i}$  は式 (3.8) で与えられる.

$$t_{n_i} = t_{col_i} + T_{jam} \quad (3.8)$$

WT 二進指数バックオフアルゴリズムにより決定される送信再試行までの待ち時間  $T_{b_i}$  を経過させる .

$$t_{n_i} = t_{col_i} + T_{jam} + T_{b_i} \quad (3.9)$$

### 3.2.6 モデルの制限

衝突が起きた場合  $t_{col_i} < t_d$  ならば , ノード  $i$  は状態 PR で JAM 信号の一部を送信し ,  $t = t_d$  で CD に遷移し JAM 信号の残りを送信する . しかし

$$t_{col_i} + T_{jam} < t_d \quad (3.10)$$

となる場合 , 状態 PR ですでにノード  $i$  でのキャリアの送出自が終了することになる . このとき ,  $S_i = PR$  でノード  $i$  はキャリアを送出しているという仮定に矛盾が生じる . このため本モデルでは式 (3.10) の状態が発生しないように , モデル化するネットワークを次の条件を満たすものに制限する .

$$\max_{i=1, \dots, N} (T_{d_i}) < T_{jam} \quad (3.11)$$

## 3.3 TCP のモデリング

### 3.3.1 プロトコルの仕様

TCP はトランスポート層のプロトコルとして、信頼性を保証しないプロトコル上で信頼性を保証した全二重通信路を提供する。信頼性を保証し効率の良い通信を行うため、TCP ではスライディングウィンドウ機構を使用している。この機構は、データの順序番号、確認応答メッセージ、送信ウィンドウ、受信ウィンドウを使用することにより実現される。

図 3.6 に TCP のスライディングウィンドウ機構の概略を示す。送信側は送信データをセグメントに分割し、セグメントにデータストリームの先頭の順序番号を付加して送信する。このとき、受信側から知らされている受信側 TCP の受信バッファの空き容量 (送信ウィンドウ) を満たすまでデータストリームを連続して送信できる。送信データは受信側からの確認応答 (ACK) が得られるまで保持される。(図 3.7)

受信側はデータを含むセグメントを受け取ると、ACK セグメントを返送する。ACK セグメントには、セグメントの順序番号と正しく受信できた受信データの順序番号、および受信バッファの空き (受信ウィンドウサイズ) の情報が含まれる。この ACK セグメントは受信側から送信するデータセグメントと同一のものでよい。また、ACK は受信したセグメント一つに対して必ずしも一つ送る必要はなく、正しく受信できたデータに対してまとめて確認応答すれば良い。

受信側から ACK を受け取ると、送信側の TCP は確認応答のあった送信データを送信バッファから解放する。受信したセグメントには受信側 TCP の受信ウィンドウサイズが含まれているから、送信側の送信ウィンドウが更新される。この結果、送信側の送信ウィンドウには空きが生じ、再び送信ウィンドウを満たすまで新しいセグメントを送信することができる。

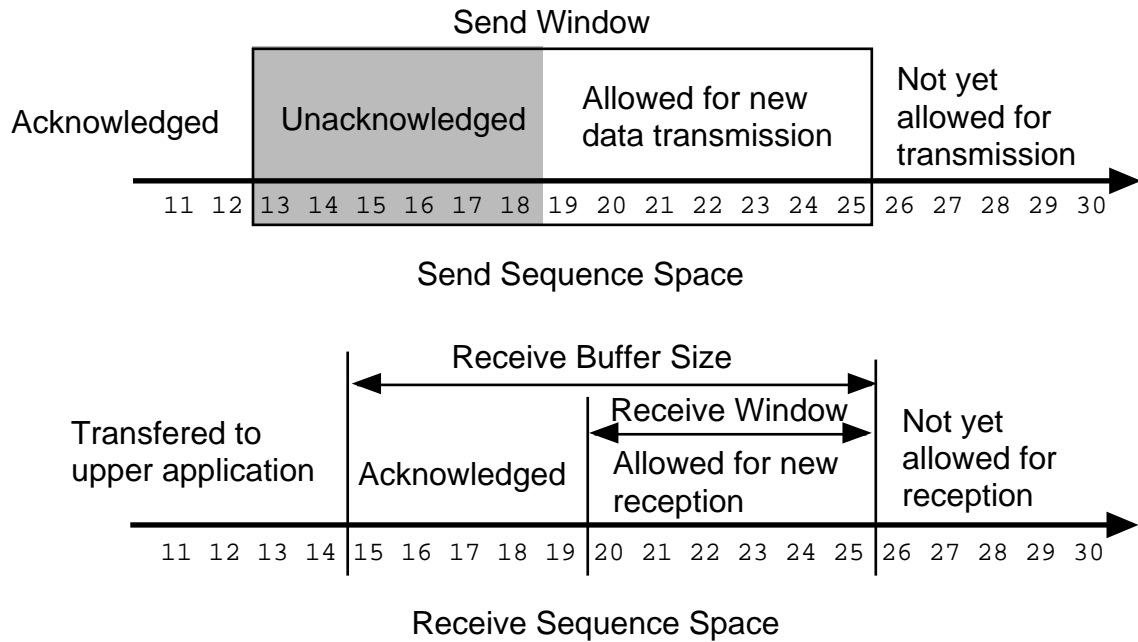


図 3.6: TCP のスライディングウィンドウ機構  
 Fig. 3.6. Sliding window mechanism of TCP.

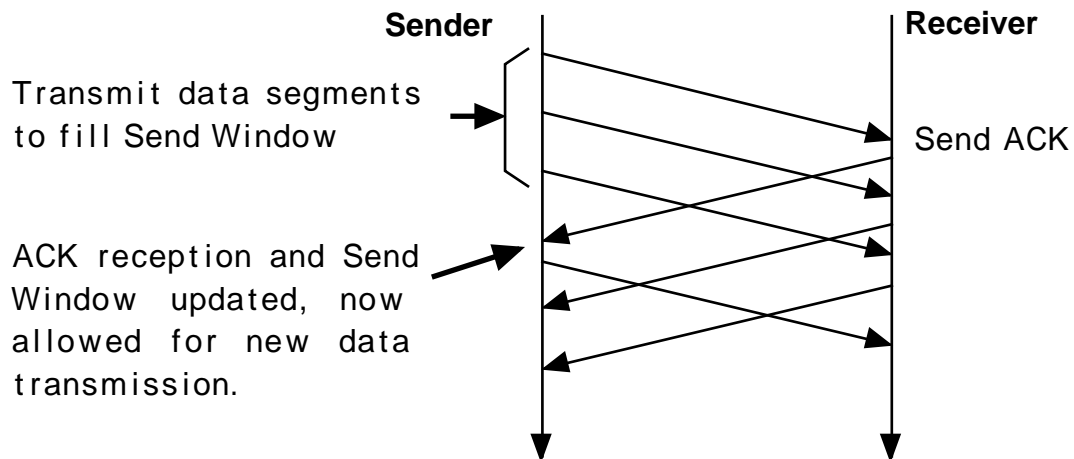


図 3.7: TCP セグメントの送信  
 Fig 3.7 Transmission of TCP segments.

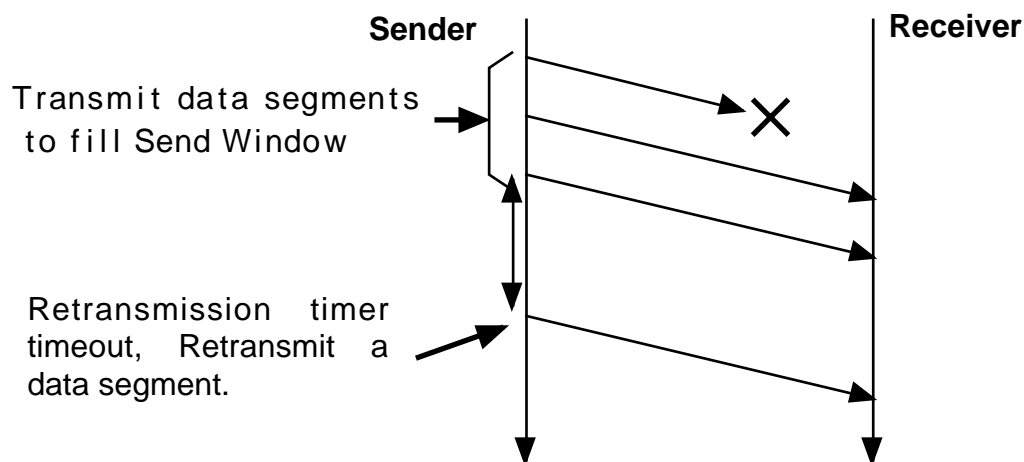


図 3.8: TCP セグメントの再送信  
Fig. 3.8. Retransmission of TCP segments.

もし送信時に設定した再送待ち時間以内に ACK が得られない場合には，確認応答の得られていないデータを再送信する．再送信時には確認応答を受けていないすべてのデータに対して ACK を受けとるまで，セグメントを一つずつ送信する．(図 3.8)

TCP の仕様<sup>[23]</sup>では，セグメントの形式や通信路の確立，切断手法などについては具体的に規定してあるが，TCP の通信性能に影響を与えるスライディングウィンドウ機構の具体的実装方法については規定していない．このため，規定された範囲内で様々な性能向上の手法が提案されている<sup>[24][25][26]</sup>．今回設計したモデルでは，実システム利用時の通信量評価を行うことを目的としているため，UNIX BSD 4.3 の TCP 実装方法<sup>[27]</sup>に準拠するものとした．

### ACK 送信のタイミング

TCP がデータを含むセグメントを受けとった時，受信側 TCP は送信側に ACK を送信する．このセグメントの送信時刻は次のように決定した．

- 200msec の周期で ACK 送信イベントが発生し，このときに ACK を送る必要があると ACK を送信する．
- ユーザアプリケーションが受信バッファからデータを受けた時に，受信バッファの空きがバッファ容量の 1/4 以上であるときには，直ちに ACK を送信する．

#### 再送タイムアウトの設定法

再送タイムアウト値の決定には，滑らかな移動平均として得られるラウンドトリップ時間（データセグメントを送信してから ACK を受けとるまでの時間）の見積もり値を利用する．この方法では，ラウンドトリップ時間  $T_{rtt_i}$  を測定しておき，これを元に式 (3.12) により Smoothed round trip time  $T_{srtt_i}$  を求める．

$$T_{srtt_i} = \alpha T_{srtt_{i-1}} + (1 - \alpha) T_{rtt_i} \quad (3.12)$$

UNIX BSD 4.3 では  $\alpha = 0.9$ ， $\beta = 2$  であり，本モデルでもこの値を使用した．これより再送タイムアウト  $T_{rto_i}$  は式 (3.13) で与えられる．

$$T_{rto_i} = \min\{T_{max}, \max(T_{min}, \beta T_{srtt_i})\} \quad (3.13)$$

$T_{max}$ ,  $T_{min}$  はそれぞれタイムアウトの最大値と最小値であり，それぞれ，64 秒，1 秒に設定した．また， $T_{srtt}$ ,  $T_{rto}$  の初期値は 6 秒とした．

再送時にはバックオフ・アルゴリズムに従って，式 (3.14) で新しい再送タイムアウトを決定する．

$$T_{rto_i} = \min\{T_{max}, \max(T_{min}, 2T_{rto_{i-1}})\} \quad (3.14)$$

また，再送が行われたセグメントに対し確認応答が得られた場合には，次の新しいセグメント送信時の  $T_{rto_i}$  は  $T_{rto_{i-1}}$  を用いる．

### ウィンドウ管理

シリーウィンドウ・シンドローム<sup>[28]</sup>を防ぐため，送信ウィンドウと受信ウィンドウを以下のように制御する．

送信側 送信データが存在するが送信ウィンドウが0で送信できない場合，送信する代わりに持続タイマを  $T_{persist_i} = \min(T_{rto_i}, T_{persistmax})$  にセットし，出力を遅らせる．持続タイマのタイムアウトまでにウィンドウ更新が受信されなければ，1セグメントを送信し，再び持続タイマを  $T_{persist_i} = \min(T_{persist_{i-1}}, T_{persistmax})$  にセットして出力を遅らせる． $T_{persistmax}$  の値は1秒に設定した．

受信側 セグメントの送信時に，受信バッファの空きが受信バッファサイズの1/4または最大セグメントサイズよりも少なければ，ウィンドウサイズを0として送信側に送信する．

TCPの送受信バッファのサイズはそれぞれ，UNIX BSD 4.3でのデフォルト値である4096byteとした．

### 3.3.2 TCP モジュールの構成

3.3.1で述べた仕様に従い，TCPモジュールを図3.9のように構成した．

送受信セグメント TCPセグメントを，仕様書で規定するTCPヘッダに含まれる情報のサブセットとして，i)セグメントの宛先と送信元，ii)正しく受信できたデータの順序番号(ACK)，iii)セグメントの先頭データの順序番号(SEQ)，iv)ウィンドウサイズ(WND)，v)データ長(LEN)のみを含むものとして表した．これをお互いに通信する2つのTCPモジュール間で受け渡す．

受信 通信相手のイーサネットモジュールで送信されたフレームがノードに到着することにより，TCPモジュールでの受信イベントが発生する．このイベントの発

生時刻は式 (3.7) によって決定される  $t_{rcv_j}$  により与えられる。TCP モジュールは受信セグメントの LEN を調べて受信バッファの空き容量を更新する。また ACK, SEQ, WND の値から送信ウィンドウおよび再送バッファの更新を行う。更に再送タイムアウトの決定のため、ラウンドトリップ時間を計算する。

送信 TCP モジュールでの送信イベントは、i) 上位アプリケーションから発生するユーザのデータ送信イベントおよび受信イベント、ii) TCP モジュール内で発生する ACK 送信タイムアウト、iii) 持続タイマタイムアウト、iv) 再送タイマタイムアウトにより発生する。これらのイベントが発生すると TCP モジュールは送信ウィンドウサイズ、ACK を送る必要があるかどうかを調べ、セグメントが送信できる状態ならセグメントを送信する。またセグメントの送信に伴い、各タイマと再送バッファ、送信ウィンドウの更新が行なわれる。この結果各 TCP モジュール内の次イベント発生時刻が計算される。

TCP モジュールでセグメントが送信されると、イーサネットモジュールの送信バッファに送信要求が加えられる。この送信要求にはセグメントのデータ長が含まれており、イーサネットモジュールでのフレーム送信時間の計算に用いられる。また、イーサネットモジュールの送信バッファ内では、TCP の送信要求は FIFO (First In First Out) で処理される。

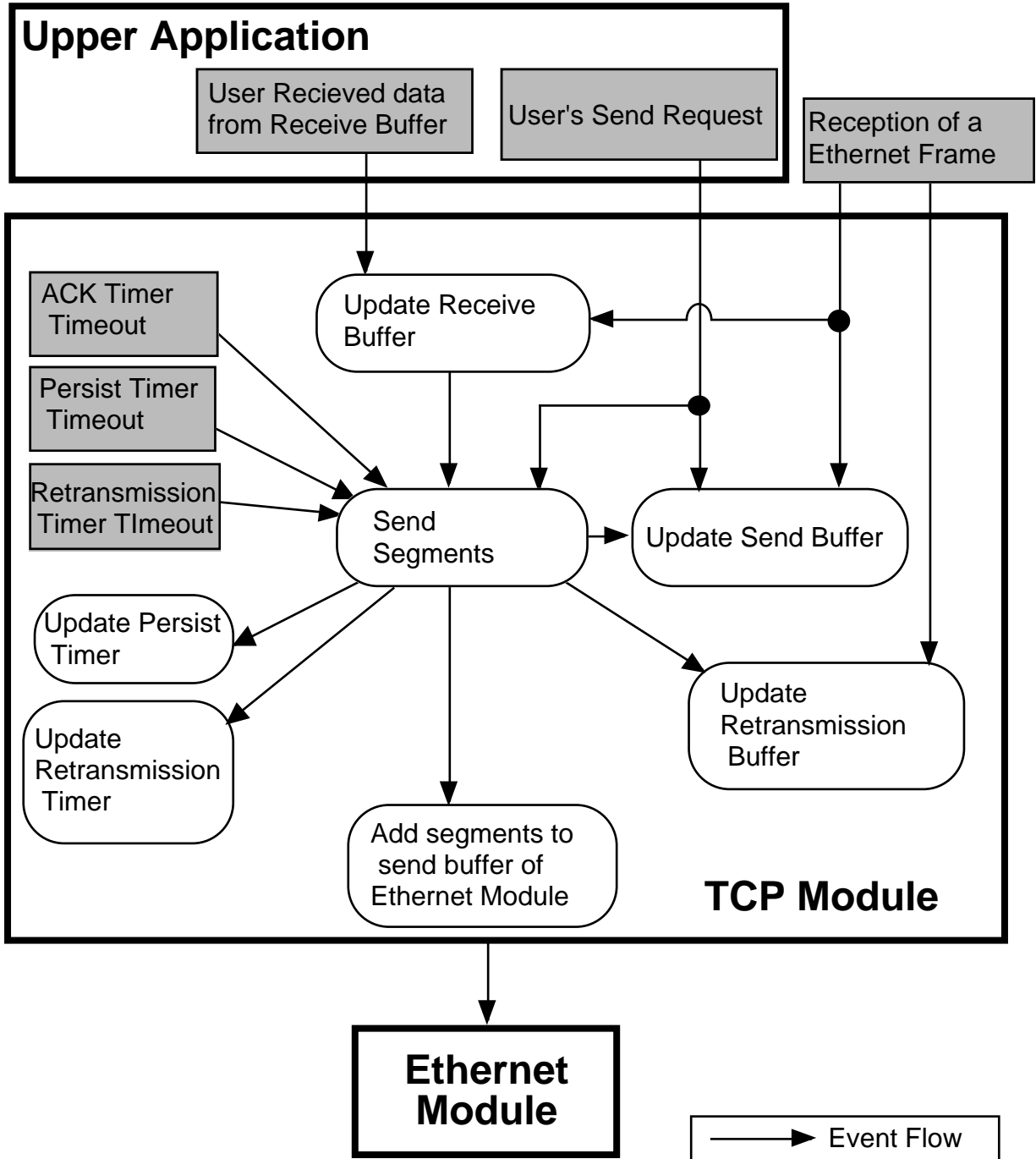


図 3.9: TCP モジュールの構成  
 Fig. 3.9. Structure of TCP module.

### 3.4 上位アプリケーションモデル

上位アプリケーションの動作は、(i)TCP の送信バッファにデータを加える動作と (ii) 受信バッファからデータを読みとる動作に大別される。

(i) の動作は、送信バッファにデータを加える動作が起きる間隔  $T_{send}$  と加えるデータのサイズ  $L_{send}$  をランダムに発生するモデルによりモデル化できる。例えば、TELNET のように小さなデータが比較的長い間隔で送信されるアプリケーションの場合には、 $T_{send}$  の平均が大きく、 $L_{send}$  の平均が小さくなるようにこれらを乱数で発生すれば良い。逆に FTP のように大きなデータを連続して送信するアプリケーションの場合、 $T_{send}$ 、 $L_{send}$  を大きくなるようにする。

(ii) の動作は、受信バッファのデータを読みとる間隔  $T_{read}$  と読みとるデータのサイズ  $L_{read}$  をランダムに発生するモデルによりモデル化できる。アプリケーションによる動作の違いは (i) の場合と同様にモデル化できる。

### 3.5 アクセス集中モデル

この節では、教育用 LAN におけるネットワークアクセス集中のモデルについて説明する。

教育用 LAN で最もネットワークアクセスが集中するのは、授業利用時に教官の指示のもとに学生が一斉にネットワークアクセスを伴う操作を行う場合である。このとき、学生が教官の指示に反応して WS の操作を行うまでの時間には個人差があるため、学生が使用しているクライアント WS からサーバ WS にファイル転送要求が送信される時刻にはばらつきが生じる。そこでシミュレーションでは各クライアント WS が送信を開始する時刻  $t_{s_i}$  を式 (3.15) により与えられる密度関数  $f(t_s)$  に従う乱数により決定した。

$$f(t_s) = \begin{cases} \left[ \int_0^\infty \exp\left\{-\frac{(\xi-m)}{2\sigma^2}\right\} d\xi \right]^{-1} \exp\left\{-\frac{(t_s-m)^2}{2\sigma^2}\right\} & (t_s > 0) \\ 0 & (t_s \leq 0) \end{cases} \quad (3.15)$$

式 (3.15) は正規分布  $N(m, \sigma^2)$  の密度関数において確率変数が非正になる確率をゼロとしたものを表している。

パラメータ  $m, \sigma$  の決定には、教育環境におけるユーザの反応速度の実測結果を用いた。川村<sup>[29]</sup>は、本学情報処理教育センターを利用中の学生 146 人を対象に、口頭による合図に対して学生がマウスのダブルクリックで応答するまでの時間を計測した。図 3.10 に反応時間 20 秒までの応答人数のヒストグラムを示す。学生の反応時間の多くは合図のあった時刻から 6 秒以内に集中している。この範囲内での反応時間の平均は 2.2 秒、標準偏差は 1.6 秒である。キーボードによってコマンドを入力した場合については反応時間の測定データは得られていないが、マウスによる場合に比べてコマンド入力に要する時間は長くなり、その時間の個人差は大きくなると考えられる。以上の理由から、モデル M) マウスによるコマンド入力を行なう場合、モデル K) キーボードによるコマンド入力を行なう場合として、表 3.3 のよ

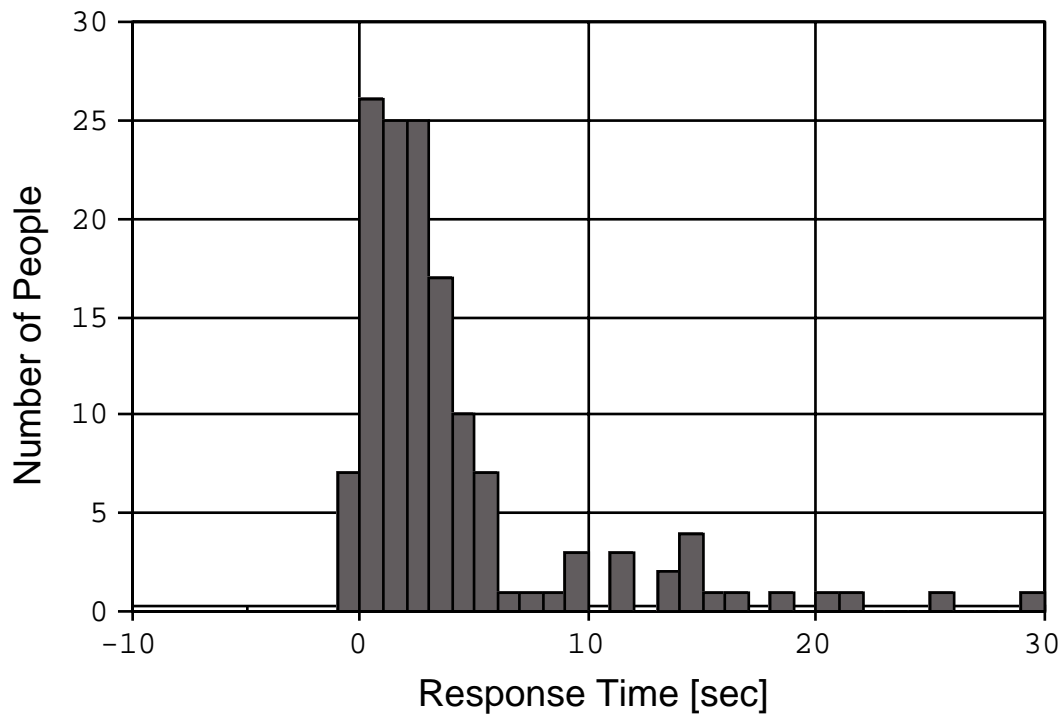


図 3.10: 学生の反応時間 (マウスのダブルクリックによる応答時)  
 Fig. 3.10. Response time of students. (by double clicking)

うにパラメータを決定した．このパラメータを用いた密度関数を図 3.11 に示す．

表 3.3: アクセス集中モデルのパラメータ

モデル名	想定したコマンド入力方法	$m$ (秒)	$\sigma$ (秒)
M	マウス	2.2	1.6
K	キーボード	6.3	10.6

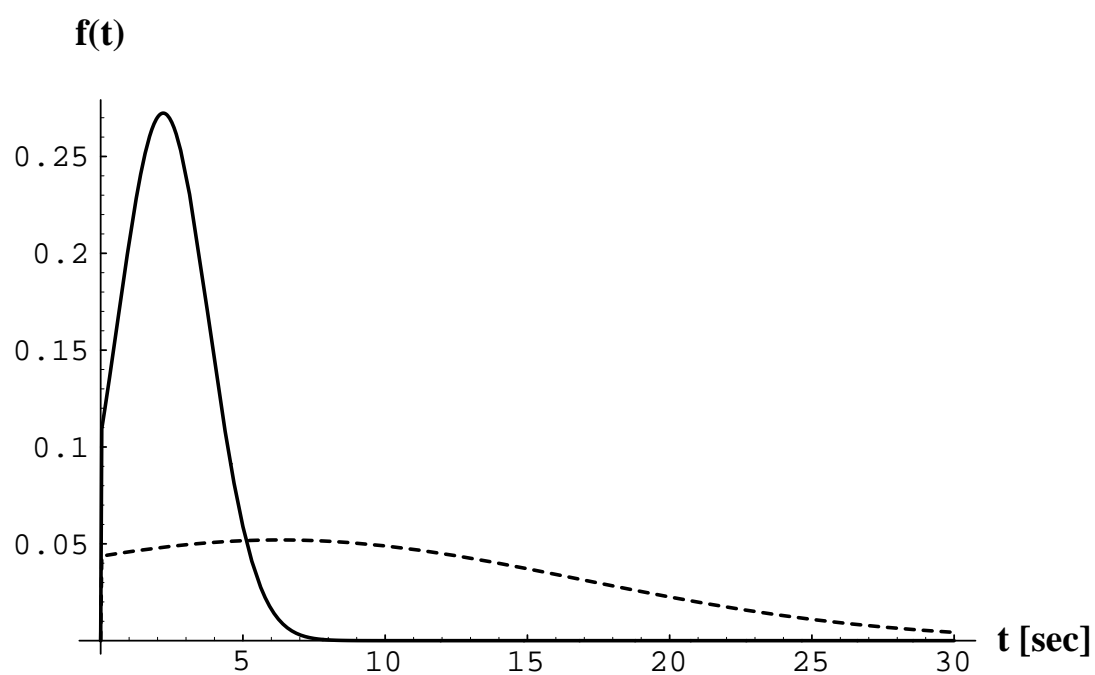


図 3.11: アクセス集中モデルの  $t_s$  決定に用いる密度関数 (実線: Model M, 破線: Model K)

Fig. 3.11 Probability density function to define  $t_s$  of access congestion model.  
(Solid line: Model M, Broken line: Model K)

## 第 4 章

# 実験と考察

### 4.1 データリンク層以下のみのシミュレーション

本節ではイーサネットモジュール単体でのシミュレーションを行い，Marino らによるイーサネット CSMA/CD シミュレータ (以下 Marino モデル)<sup>[12]</sup> による結果との比較によりシミュレーションモデルの基本的な信頼性を検証する．

#### 4.1.1 シミュレーションの条件

イーサネットで接続された  $N$  個のノードでランダムに送信要求が発生し，伝送路上にフレームを送信すると仮定する．この条件は，モデル上ではイーサネットモジュールの上位に TCP モジュールではなく，ランダムに送信要求が発生するモジュールをおくことで実現できる．以下，各パラメータの設定値について説明する．

- 各ノードにおけるイーサネットモジュールへの送信要求の発生間隔は平均  $1/\lambda$  の指数分布に従う．
- ネットワーク入力負荷  $\rho$  は式 (4.1) で計算される．

$$\rho = \frac{N\lambda}{\mu} \quad (4.1)$$

ここで  $1/\mu$  は 1 フレームの送信時間とフレームギャップ時間の和である．今回のシミュレーションでは伝送速度を 10Mbps，送信フレームのデータ部の

長さを 512byte とした . このとき , フレームの長さはプリアンプルを含めて 4304bit であり ,

$$\frac{1}{\mu} = 0.440 \text{ msec} \quad (4.2)$$

である .

- 各ノードは HUB を介してスター型に接続されているものとし , ノード間の遅延はすべて 1.140msec とした . これは 10BASE-T ネットワークで 10m のケーブルで HUB と各ステーションを接続した場合に , ケーブル以外の遅延を無視した値である .

Marino モデルと本モデルでのシミュレーション条件が異なるのは , i) ネットワーク上のノードの配置 , ii) フレーム長の二点である . Marino モデルではノードの配置を 2480m の同軸ケーブル上 ( $\alpha = 0.016$ ) に 32 のノードが均等に配置されているものとしている . また , フレーム長は文献<sup>[12]</sup>では明記されていないが , Ethernet を仮定しているのでプリアンプルを含んで 512 bit (64 byte) から 12134 bit (1518 byte) の範囲内の値であり ,  $0.060 \leq 1/\mu \leq 1.224 \text{ msec}$  である .

#### 4.1.2 性能評価指標

性能評価指標について以下のように定義する .

平均送信遅延  $d$  送信が成功した全フレームの送信遅延  $d_{ij} (i = 1, \dots, N, j = 1, \dots, n_{f_i})$

の平均 . すなわち ,  $d = (\sum_{i=1}^N \sum_{j=1}^{n_{f_i}} d_{ij}) / (\sum_{i=1}^N n_{f_i})$  . ここで  $n_{f_i}$  は  $i$  番めのノードが送信したフレームのうち送信に成功したフレームの数である .  $d_{ij}$  はフレームの送信要求からフレームの送信が成功し , 終了するまでの時間である . ここでは  $d_{ij}$  を  $1/\mu$  の倍数として表す . フレーム送信要求の発生時に伝送路が空いており , 衝突がなかった場合にはフレームの送信遅延は再短であり ,  $d_{ij} = 1$  である .

衝突率  $c$  全送信成功フレーム数に対する衝突数の割合．すなわち， $n_{c_i}$  を  $i$  番めのノードで送信したフレームが衝突した回数の総和とすると， $c = (\sum_{i=1}^N n_{c_i}) / (\sum_{i=1}^N n_{f_i})$  である．

### 4.1.3 シミュレーション結果と考察

ここでは本シミュレーションモデルによる結果と Marino モデルによる結果との比較，およびシミュレーション結果の信頼区間から本モデルの信頼性を検証する

#### Marino モデルとの比較

シミュレーションは，反復実施法に基づきノード数およびネットワーク入力負荷が異なる条件で，乱数発生 of 初期値が異なる独立した試行を 30 回繰り返した．各試行のシミュレーション上の時間は 20 秒間とした．シミュレーション実行に要した時間は， $N = 32, \rho = 0.5$  のとき S-4/2000(処理能力 657MIPS) を用いて 1 回の試行で 18 秒である．一方 Marino モデルではクロック 12MHz の AT-286 で 70 秒である．計算機の能力を考慮すると Marino モデルに較べて本モデルの方がシミュレーションに多くの時間を要するといえる．この原因は，Marino モデルが各ノードがフレームの送信終了時に次のフレーム送信時刻を決定可能であるとしてシミュレーションの経過時間ステップを長くとしているのに対し，本モデルでは常にフレームの送信時刻はデータリンク層では予測不能であるとして，細かい経過時間ステップでイーサネットモジュールで全ノードの状態遷移を行なっているからである．

ノード数  $N = 32$  のときの平均送信遅延  $d$  と衝突率  $c$  を表 4.1，図 4.1，4.2 に示す．図 4.1，4.2 から， $\rho$  が増大するに従い  $c$  は増大し， $\rho = 0.6$  付近で  $d$  が急激に増大することがわかる．本モデルと Marino モデルの結果を比較すると，両者にシミュレーション条件の若干の違いがあるものの，期待通り  $d, c$  とともに両者の結果はほぼ一致している．

表 4.1: 平均送信遅延と衝突率 ( $N = 32$ )

Load $\rho$	平均送信遅延 $d$		衝突率 $c$	
	Proposed Model	Marino Model	Proposed Model	Marino Model
0.10	1.055		0.018	
0.20	1.139		0.068	
0.20		1.20		0.10
0.30	1.269		0.166	
0.40	1.480		0.317	
0.42		1.70		0.40
0.50	1.955		0.556	
0.56		2.45		0.84
0.60	3.340		0.891	
0.61		3.80		1.08
0.70	9.008		1.331	
0.71		9.10		1.61
0.80	38.946		1.772	
0.90	119.335		1.621	
1.00	317.607		1.038	

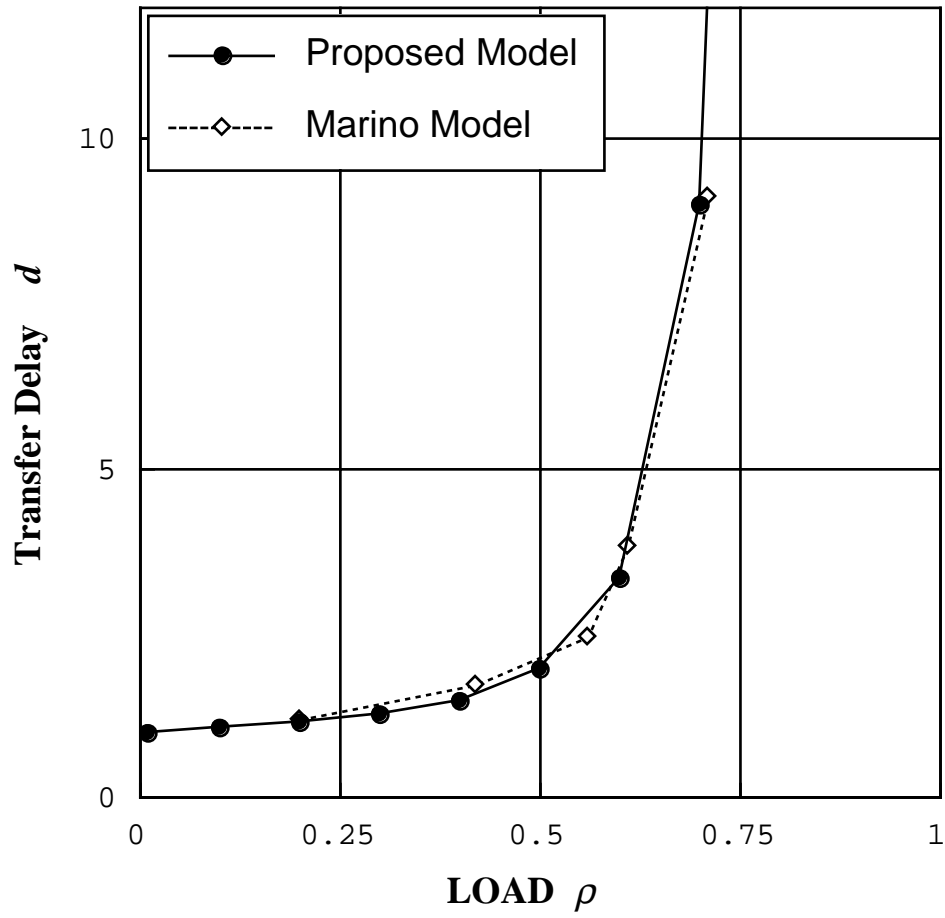
図 4.1: 平均送信遅延 ( $N = 32$ )

Fig. 4.1. Mean transfer delay.

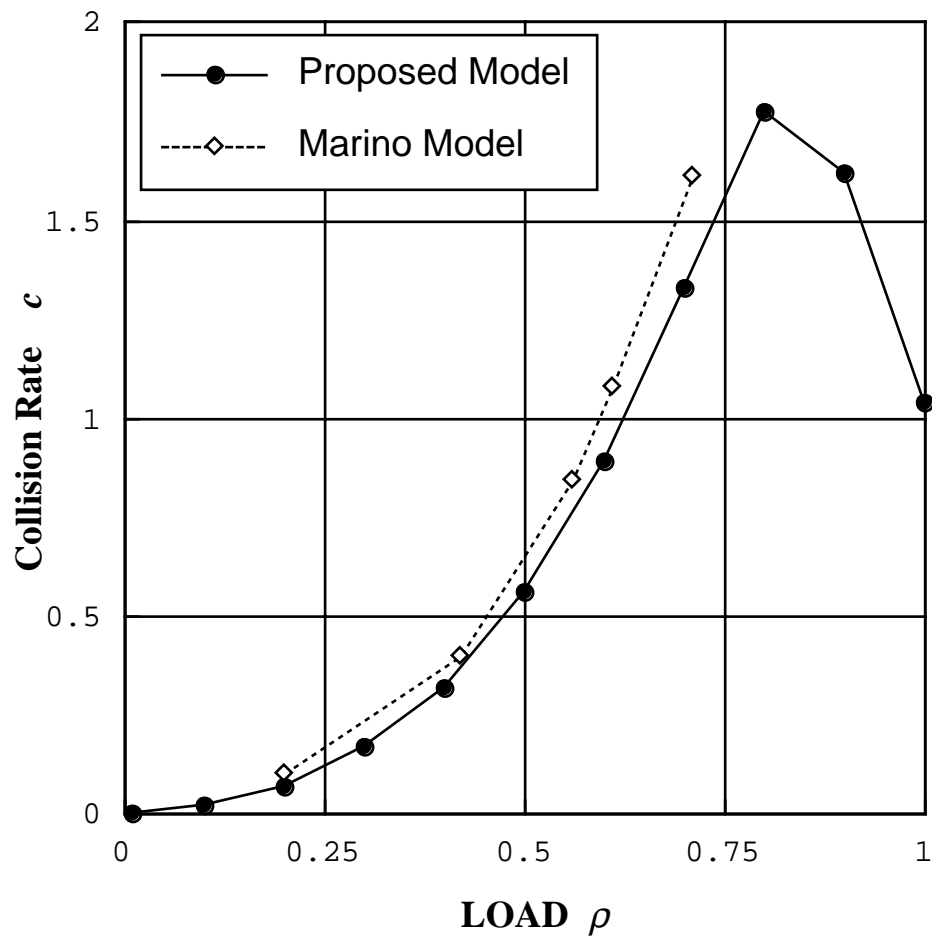


図 4.2: 衝突率 ( $N = 32$ )  
Fig. 4.2. Mean collision rate.

## 信頼区間

信頼区間は反復実施法に基づき求めた、 $n$  回の試行間における独立性と特性値  $X$  が正規分布に従って分布していることを仮定すれば、サンプルの平均値を  $\bar{X}$  とすると次の確率変数

$$t = \frac{\bar{X} - m}{(S'/\sqrt{n})} \quad (4.3)$$

は自由度  $(n - 1)$  の  $t$  分布

$$f(t) = \frac{1}{\sqrt{(n-1)\pi}} \frac{\Gamma(n/2)}{\Gamma((n-1)/2)} \left(1 + \frac{t^2}{n-1}\right)^{-n/2} \quad (4.4)$$

に従って分布している。ここで、 $m$  は母集団の平均であり、 $S'$  はサンプルの不偏分散である。これより、 $m$  の  $100 \times (1 - \alpha)\%$  信頼限界  $\mu_{\alpha}^{+}$ 、 $\mu_{\alpha}^{-}$  が式 (4.5) で与えられる。

$$\left. \begin{array}{l} \mu_{\alpha}^{+} \\ \mu_{\alpha}^{-} \end{array} \right\} = \bar{X} \pm \frac{S'}{\sqrt{n}} t_{1-\alpha/2} \quad (4.5)$$

$t_{1-\alpha/2}$  は自由度  $(n - 1)$  の  $t$  分布の  $100 \times (1 - \alpha/2)\%$  値である。

表 4.2、図 4.3 に  $n = 30$ 、 $N = 32$  のときの  $d$  と  $c$  の 95% 信頼区間幅のサンプル平均に対する割合  $(\mu_{0.05}^{+} - \bar{X})/\bar{X}$  を示す。  $0.2 < \rho \leq 1.0$  の範囲では  $d$ 、 $c$  の両方で 95% 信頼区間は平均値の  $\pm 5\%$  以内である。

以上の検討より、本モデルはネットワーク動作を正確にシミュレーションするモデルとして有効であることが確かめられた。

表 4.2: 95%信頼区間  
サンプル平均に対する割合で示した。

Load $\rho$	95%信頼区間 [%]	
	平均送信遅延 $d$	衝突率 $c$
0.1	0.18	8.21
0.2	0.20	3.36
0.3	0.35	2.24
0.4	0.42	1.12
0.5	1.33	0.75
0.6	2.01	0.75
0.7	3.36	0.75
0.8	2.27	0.37
0.9	1.32	0.37
1.0	3.24	1.49

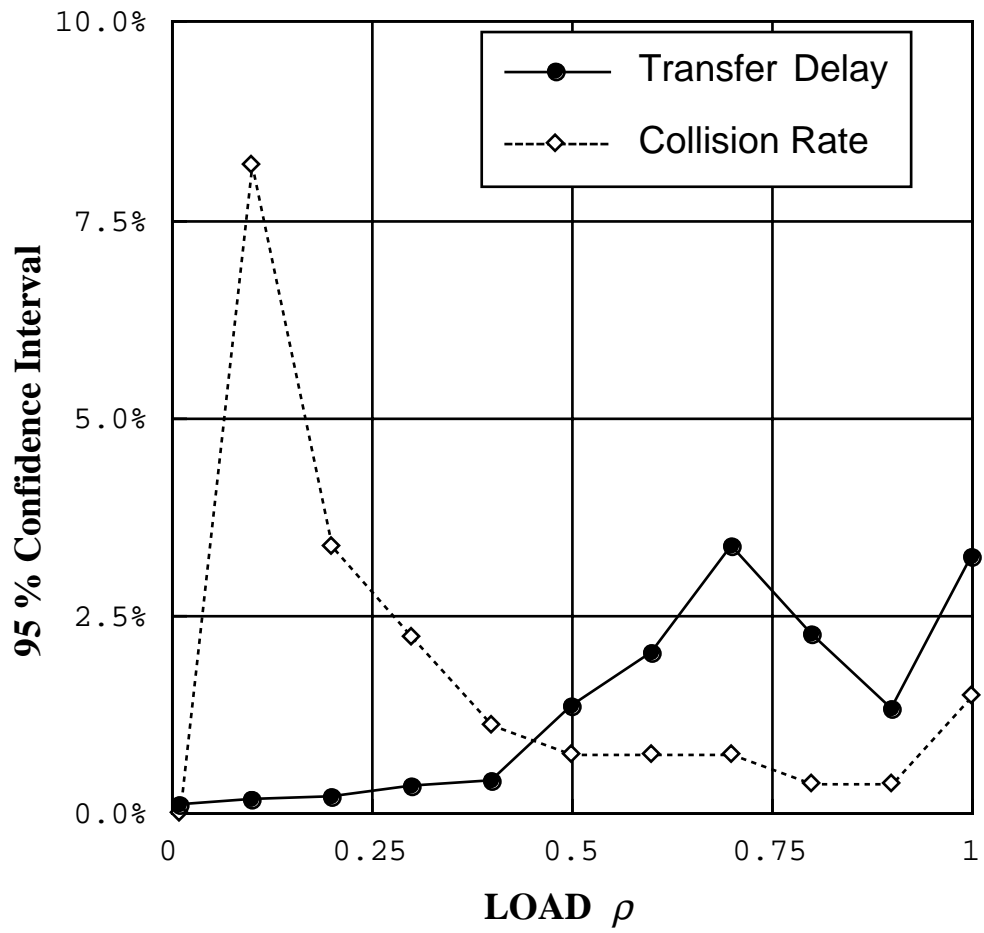


図 4.3: 95%信頼区間 ( $N = 32$ )  
 Fig. 4.3. 95 % confidence interval.

## 4.2 情報処理教育環境におけるネットワークシミュレーション

この節では教育用 LAN でのクライアントサーバシステムにおいて、ファイルサーバにクライアントからのファイル転送要求が集中した場合を想定したシミュレーションの方法と結果について述べる<sup>[15][30]</sup>。

### 4.2.1 環境設定

#### ネットワーク環境

図 4.4 に示すように、より対線で 1 台の HUB に 1 台のファイルサーバ WS(SWS) と  $N$  台のクライアント WS(CWS) が接続されたネットワークを仮定した。物理層、データリンク層の仕様は伝送速度 10Mbps の 10BASE-T または 100Mbps の 100BASE-TX<sup>[31]</sup> とする。より対線の長さはすべて 10m とした。この条件は HUB の遅延を無視した場合 3 章で規定したモデルの制限を満たす。

各 CWS は 3.5 節で定義したアクセス集中モデルに従い、一斉に SWS にファイル転送要求を送信する。SWS はクライアントからの要求に応じてファイルを CWS に送信する。

#### 上位アプリケーションのモデル

上位アプリケーションにファイル転送プログラムを使うので、上位アプリケーションの動作を以下のようにモデル化した。

1. ファイル転送開始時刻になると、CWS の上位アプリケーションモデルは送信要求メッセージを表す 20byte のデータを TCP モジュールの送信バッファに加える。

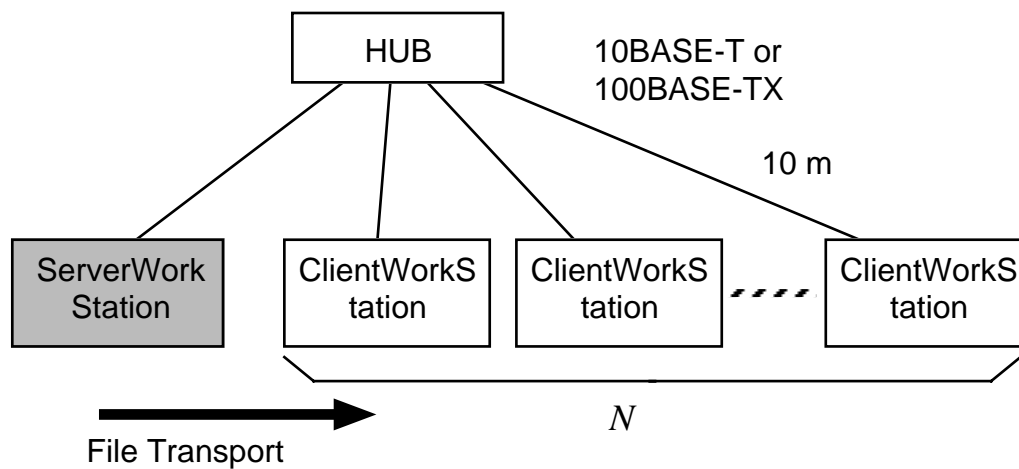


図 4.4: 仮定したネットワーク環境  
Fig. 4.4. Supposed network environment.

2. SWS の上位アプリケーションモデルは TCP モジュールの受信バッファから送信要求メッセージサイズの 20byte を読みとると、転送ファイルサイズ分のデータを TCP の送信バッファに加える。
3. SWS での送信データがすべて TCP モジュールに渡され、CWS で TCP の受信バッファからファイルサイズ分のデータが読みとられた時、通信は終了する。

また、上位アプリケーションモデルにおける TCP モジュールの送受信バッファへの読み書き動作は、以下のようにモデル化した。

- WS の処理速度は十分に高速なものであるとし、上位アプリケーションのモデルは、送信データがなくなるまで常に TCP モジュールの送信バッファをいっぱいにする。すなわち上位アプリケーションの送信動作は、CWS の場合ファイル転送要求メッセージサイズ、SWS の場合転送ファイルサイズによってのみ決められる。

- 上位アプリケーションの受信動作は，平均 0.7msec の指数乱数で決定される読みとり間隔で TCP の受信バッファを空にすることによりモデル化する．

#### 4.2.2 性能評価指標

性能評価指標について以下のように定義する．

平均ファイル転送時間  $\overline{T}_{trs}$   $i$  番めの CWS におけるファイル転送要求発生時刻から，すべてのファイルを受信するまでの時間  $T_{trs_i}$  ( $i = 1, \dots, N$ ) の全 CWS における平均．すなわち， $\overline{T}_{trs} = \frac{\sum_{i=1}^N T_{trs_i}}{N}$

平均実効ファイル転送速度  $\bar{v}$   $i$  番めの CWS における単位時間あたりのファイル転送量  $v_i$  の全 CWS における平均．すなわち， $\bar{v} = \frac{\sum_{i=1}^N v_i}{N}$ ．また， $s$  を転送ファイルサイズとすると， $v_i = T_{trs_i}/s$  である．

平均フレーム衝突率  $\bar{c}$  SWS を含むすべてのノード  $i = (0, \dots, N)$  での，フレーム送信試行回数  $n_{a_i}$  にしめるフレーム衝突回数  $n_{c_i}$  の割合をすべてのノードで平均したもの．すなわち， $\bar{c} = \frac{\sum_{i=0}^N (n_{c_i}/n_{a_i})}{N+1}$

平均フレーム送信エラー率  $\bar{\epsilon}$  SWS を含むすべてのノード  $i = (0, \dots, N)$  での，送信に失敗したフレームを含む全送信フレーム数  $n_{s_i}$  に占める，送信に失敗したフレーム数  $n_{e_i}$  の割合の平均．すなわち， $\bar{\epsilon} = \frac{\sum_{i=0}^N (n_{e_i}/n_{s_i})}{N+1}$ ．

平均データセグメント再送率  $\bar{r}$   $i$  番めの CWS と SWS の TCP コネクションにおいて，全く再送がない場合にサーバ側 TCP が送信するべきデータセグメント数を  $m_{s_i}$  とし，再送したデータセグメント数を  $m_{r_i}$  とする．このとき， $r_i = \frac{m_{r_i}}{m_{s_i} + m_{r_i}}$  をデータセグメント再送率と定義する． $\bar{r}$  は全 TCP コネクションにおける  $r_i$  の平均であり， $\bar{r} = \frac{\sum_{i=1}^N r_i}{N}$

### 4.2.3 シミュレーション結果と考察

シミュレーションの実行には S-4/2000 を用いた。計算機の処理能力は 657MIPS である。シミュレーションの実行時間は  $N$  とアクセス集中モデル，物理伝送路の伝送速度，転送ファイルサイズにより大きく変化する。 $N = 20$ ，アクセス集中モデル M，物理伝送速度 10Mbps，転送ファイルサイズ 1MB の場合，シミュレーションの 1 回の試行には，39 秒を要した。

各条件でのシミュレーションの繰り返し回数を表 4.3 に示す。繰り返し回数は  $N$  に応じて変化させた。その他の条件が異なる場合でも  $N$  が同じならば，繰り返し回数は同じである。

以下，特別に記述しない限り，シミュレーション結果は転送ファイルサイズ 1MB のときのものである。

表 4.3: シミュレーションの繰り返し回数

クライアント WS 台数 $N$	繰り返し回数
1, 2, 4, 5, 8, 10	20
12, 14, 16, 18, 20, 25, 30, 35	40
40, 50	20
75	10
100	5

## 物理層伝送速度 10Mbps の場合

シミュレーション結果を表 4.4, 4.5 に示す.

## 平均ファイル転送時間

転送ファイルサイズを 1MB, 500KB としたときの  $N$  に対する平均ファイル転送時間  $\overline{T_{trs}}$  の変化を図 4.5, 4.6 に示す. マウスによるコマンド入力を仮定したアクセス集中モデル M では, 転送ファイルサイズが 1MB, 500KB の両方の場合について,  $\overline{T_{trs}}$  は  $N$  に対してほぼ線形に増加している. 一方, キーボードによるコマンド入力を仮定したモデル K では,  $N$  が小さい時には  $\overline{T_{trs}}$  の増加率は少ないが, ファイルサイズ 1MB のときは  $N > 20$ , 500KB のときは  $N > 30$  で急激に増加している. この時の増加率はモデル M の場合とほぼ同じである.

## 平均実行ファイル転送速度

$N$  に対する平均実効ファイル転送速度  $\bar{v}$  の変化を図 4.7 に示す. 見やすくするため, 軸は対数でとった. モデル M の場合には  $\bar{v}$  は  $N$  にほぼ反比例の関係で減少する. モデル K では,  $N < 20$  までは緩やかに減少し,  $N = 20$  付近から急激に減少する. しかし, モデル M, K どちらの場合も,  $N = 100$  のときでも CWS 一台当たりの物理層伝送速度 0.1Mbps 以上の  $\bar{v}$  が得られている.

ここで,  $\bar{v}$  の増加率が変化するときの  $N$  を  $N_t$  と定義する. 上記の結果から, ファイルサイズが大きい場合, アクセス集中度が大きい場合には  $N_t$  が小さくなることがわかる.  $N < N_t$  の場合, 各 CWS の SWS との通信が相互に干渉することなく通信することができ, 十分実用的な速度で通信が可能である.  $N > N_t$  の場合, 各 CWS と SWS との通信が相互に干渉しあい, 通信速度は低下する. このことから,  $N_t$  の値は物理ネットワーク 1 セグメントあたりのクライアント数を決定する指標になるといえよう. すなわち, 教育用 LAN の 1 セグメントに 1 台の SWS に接続する CWS の台数  $N$  は  $N < N_t$  を満たすようにすれば, すべての CWS で十

表 4.4: 平均ファイル転送時間と実効ファイル転送速度 (伝送速度 10Mbps)

N	平均ファイル転送時間 [sec]				実効ファイル転送速度 [Kbps]			
	File Size: 1MB		File Size: 500KB		File Size: 1MB		File Size: 500KB	
	Model M	K	M	K	M	K	M	K
1	1.083	1.087	0.544	0.542	7748	7717	7708	7735
2	1.362	1.085	0.597	0.544	6802	7736	7269	7717
4	2.138	1.340	0.821	0.609	4903	6895	6090	7350
6	3.504	1.395	1.024	0.641	3279	6801	5463	6987
8	4.725	1.658	1.612	0.854	2294	6157	4308	6566
10	5.867	1.998	2.235	0.716	1817	5518	3355	6744
12	7.455	2.099	2.618	0.850	1414	5253	2769	6560
14	8.975	2.323	3.398	0.857	1162	5048	2057	6298
16	10.420	3.068	3.861	0.955	947	4310	1794	5960
18	11.932	3.264	4.503	0.930	853	4058	1420	6113
20	13.531	3.802	4.973	1.041	733	3533	1318	5697
25	17.505	6.091	6.695	1.311	552	2421	921	5086
30	21.622	8.666	8.633	1.546	439	1643	717	4549
35	26.151	13.089	10.492	2.135	355	1073	541	3872
40	30.796	16.146	12.652	2.724	302	818	436	3342
50	38.960	25.207	16.078	4.655	239	528	345	2086
75	57.649	45.221	26.823	13.262	155	278	181	648
100	74.469	65.650	37.108	24.092	115	177	131	308

分実用的な速度で通信が可能になる。但し今回シミュレーションを行った  $N < 100$  の範囲では、 $N = 1$  の場合をのぞき、 $\bar{v}$  は最低でも CWS 一台当たりの物理層伝送速度を下回ることはない。

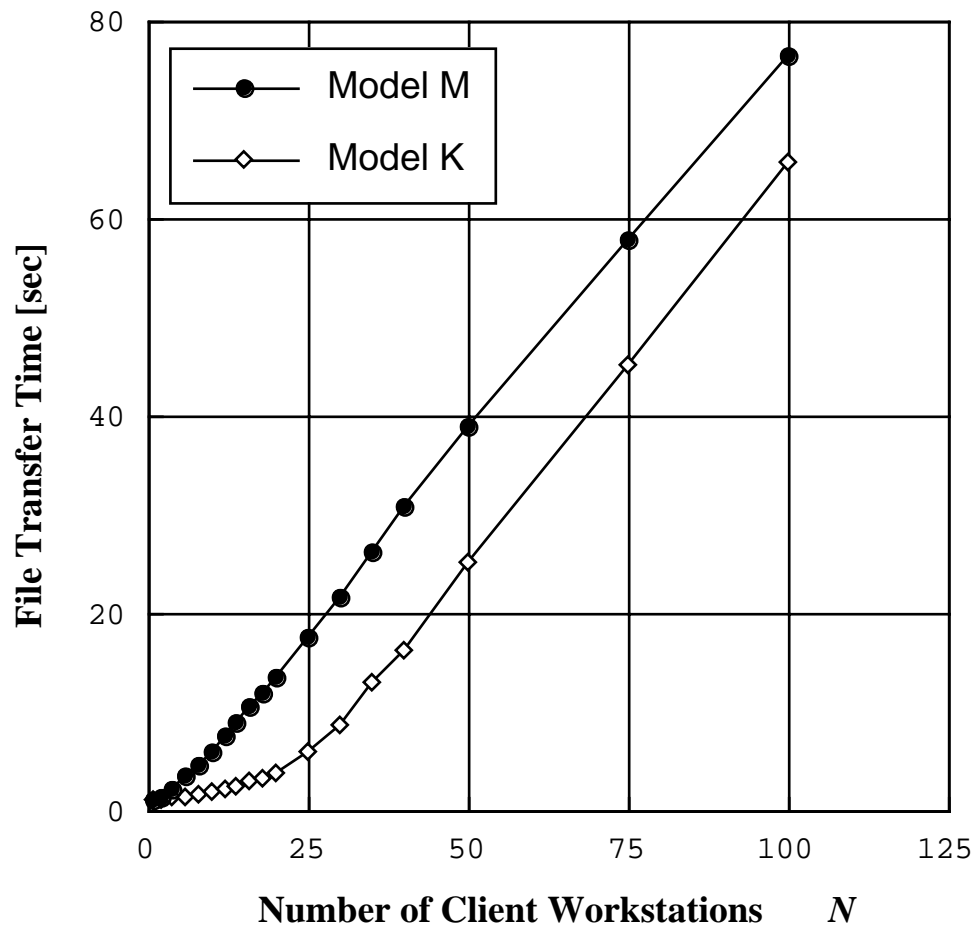


図 4.5: 平均ファイル転送時間 (伝送速度 10Mbps, ファイルサイズ 1MB)  
Fig. 4.5. Mean file transfer time. (line speed 10Mbps, file size 1MB)

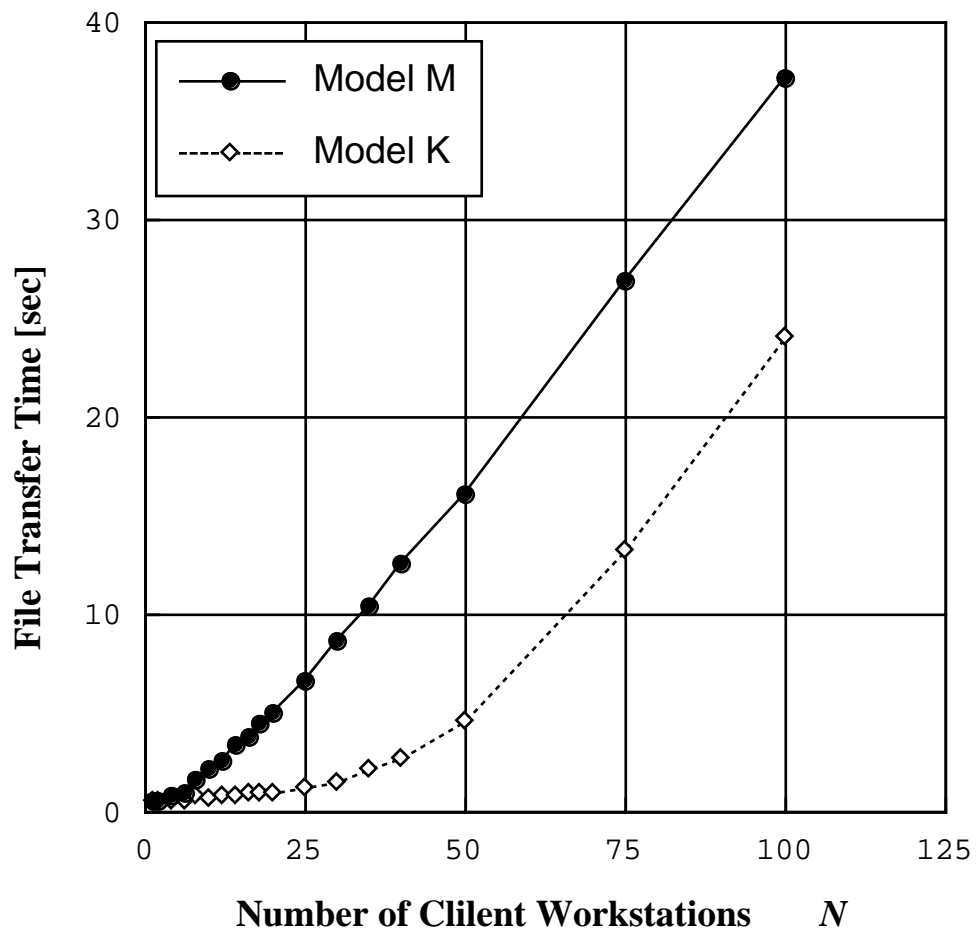


図 4.6: 平均ファイル転送時間 (伝送速度 10Mbps, ファイルサイズ 500KB)

Fig. 4.6. Mean file transfer time. (line speed 10Mbps, file size 500KB)

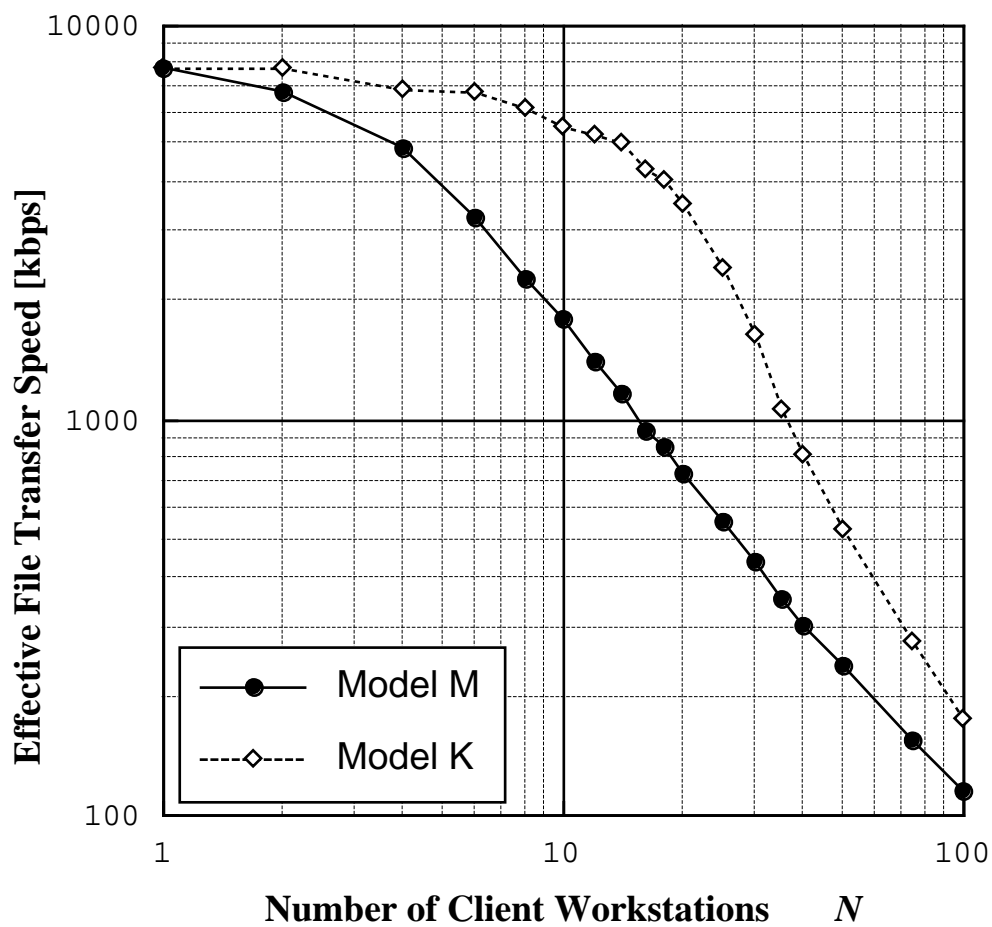


図 4.7: 平均実効ファイル転送速度 (伝送速度 10Mbps, ファイルサイズ 1MB)  
Fig. 4.7. Mean effective file transfer speed. (line speed 10Mbps, file size 1MB)

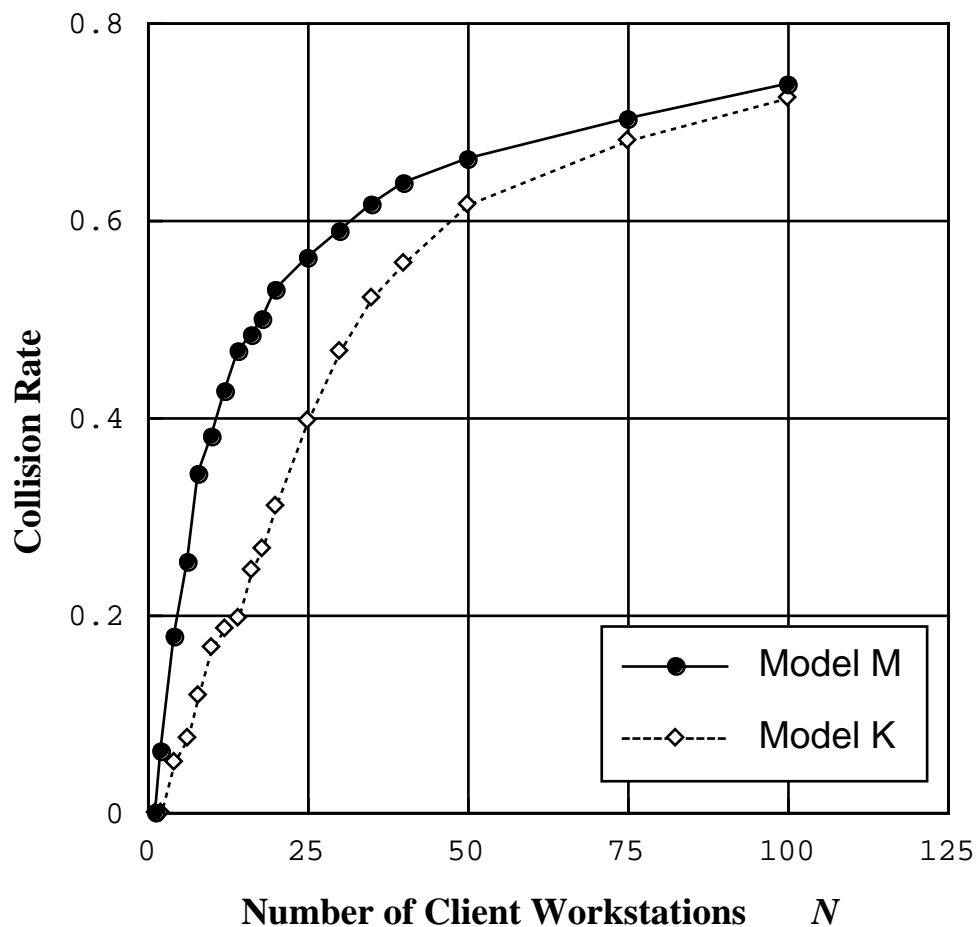


図 4.8: 平均フレーム衝突率 (伝送速度 10Mbps, ファイルサイズ 1MB)  
 Fig. 4.8. Mean frame collision rate. (line speed 10Mbps, file size 1MB)

平均フレーム衝突率  $N$  に対する平均フレーム衝突率  $\bar{c}$  の変化を図 4.8 に示す。  $N$  が小さい場合には  $\bar{c}$  は  $N$  の増加にともない急激に増加するが、  $\bar{c} \geq 0.5$  になると緩やかに増加する。

#### 平均フレーム送信エラー率

$N$  に対する平均フレーム送信エラー率  $\bar{\epsilon}$  の変化を図 4.9 に示す。

エラー率の増加傾向はファイル転送時間と同様の傾向を示している。アクセス集中モデル M の場合、  $\bar{\epsilon}$  は  $N$  に対してほぼ線形に増加する。一方、モデル K の場

表 4.5: 平均フレーム衝突率, 送信エラー率, データセグメント再送率 (伝送速度 10Mbps, ファイルサイズ 1MB)

N	平均フレーム衝突率		平均フレーム送信エラー率		データセグメント再送率	
	Model M	K	M	K	M	K
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0622	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.1792	0.0525	0.0011	0.0002	0.0000	0.0000
6	0.2538	0.0756	0.0025	0.0003	0.0001	0.0000
8	0.3439	0.1196	0.0048	0.0006	0.0007	0.0000
10	0.3819	0.1688	0.0062	0.0010	0.0012	0.0000
12	0.4280	0.1870	0.0089	0.0011	0.0019	0.0000
14	0.4673	0.1974	0.0120	0.0013	0.0023	0.0000
16	0.4837	0.2475	0.0136	0.0021	0.0029	0.0001
18	0.5007	0.2692	0.0162	0.0026	0.0034	0.0001
20	0.5286	0.3119	0.0194	0.0034	0.0041	0.0002
25	0.5612	0.3966	0.0252	0.0071	0.0055	0.0013
30	0.5889	0.4663	0.0319	0.0113	0.0067	0.0022
35	0.6149	0.5207	0.0402	0.0184	0.008	0.0039
40	0.6367	0.5578	0.0465	0.0238	0.0092	0.0049
50	0.6608	0.6164	0.0601	0.0398	0.0106	0.0080
75	0.7027	0.6812	0.0840	0.0714	0.0128	0.0112
100	0.7365	0.7242	0.1068	0.0911	0.0143	0.0135

合， $N$  が小さい場合には  $\bar{\epsilon}$  は小さく保たれるが， $N > 20$  で急激に増加する．このときの  $N$  の値は図 4.7において  $\bar{v}$  が急増する時の  $N$  の値と同じである．

この結果から，図 4.8での  $\bar{c}$  の増加率の変化を説明することができる． $N$  が小さい場合  $N$  の増加にともない，同時に多数の WS がフレームの送信を開始する頻度が高くなり，頻繁に衝突が起こるようになる．一つのフレームの送信において連続した衝突の発生は送信エラーを引き起こす．このため， $\bar{c} \geq 0.3$  になると送信エラーが発生するようになる．送信エラーの発生はそのフレームの送信再試行を行わないことであるから，この結果  $\bar{c}$  の増加は抑制される．また，送信エラーの発生により，SWS が送信する TCP データセグメントや CWS からの確認応答が失われることになり， $\bar{v}$  が急激に増大する．

#### 平均データセグメント再送率

図 4.10に  $N$  に対する平均データセグメント再送率  $\bar{r}$  の変化を示す．

データセグメントの再送は，伝送路上でのデータセグメントまたは確認応答の消失が起きた場合と，送信遅延が長いために再送タイムアウトが起きた場合に行われる．このため， $N$  に対する  $\bar{r}$  の変化は  $\bar{\epsilon}$  の場合とほぼ同じ傾向を示している．しかし， $N$  が小さく送信エラーが起きていてもその頻度が少ない場合には，再送が起きていない．このことは次のように説明できる．TCP の確認応答はデータセグメント 1 つに対して必ず一つ送信される必要がなく，正しく受信できたデータの順序番号のみを返せばよいものである．このため，少量の確認応答が TCP より下層のプロトコルで失われても再送が起きないのである．

また，再送率の増加率はアクセス集中モデル M の場合に  $N = 40$ ，モデル K の場合に  $N = 50$  付近で緩やかになっている．これは TCP の再送タイマー制御が適切に働き，確認応答が失われなかったが送信側に到着するのが遅かった場合に，再送タイムアウトにより無駄な再送信を行うケースが減少したためと思われる．

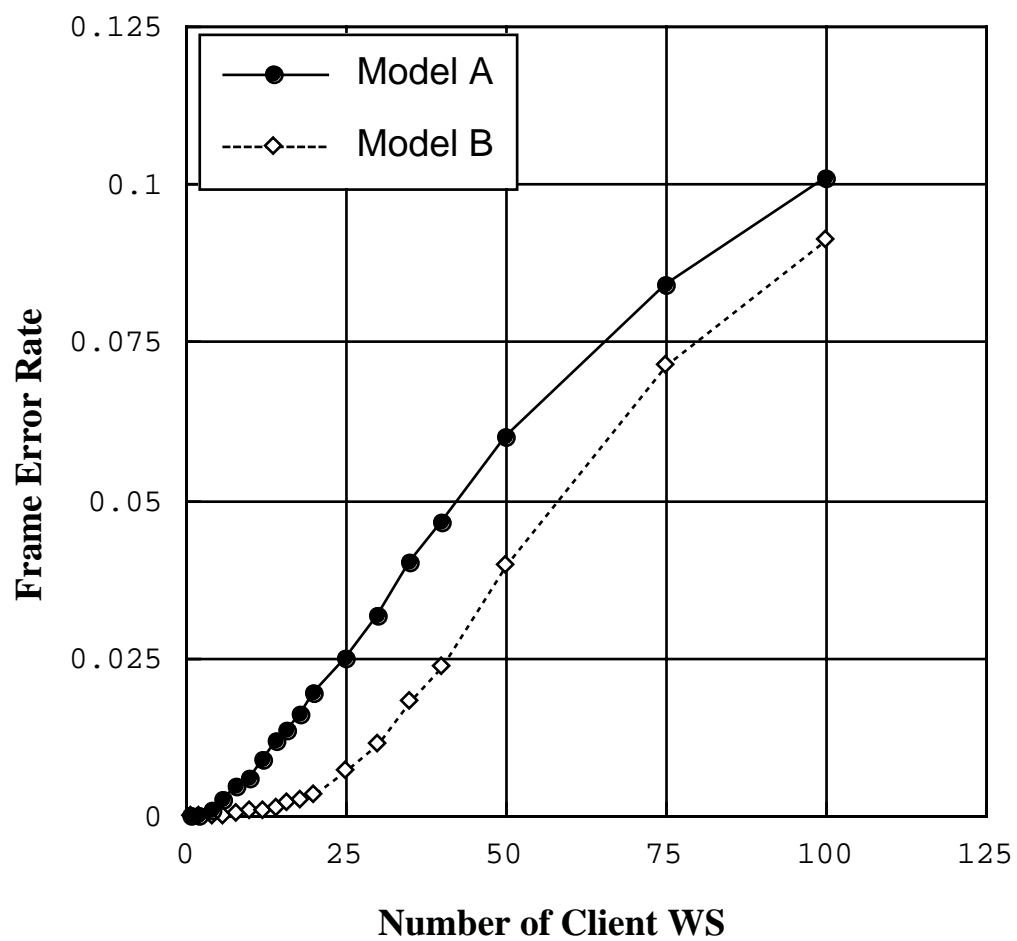


図 4.9: 平均フレーム送信エラー率 (伝送速度 10Mbps , ファイルサイズ 1MB)  
Fig. 4.9 Mean frame transmission error rate.(line speed 10Mbps, file size 1MB)

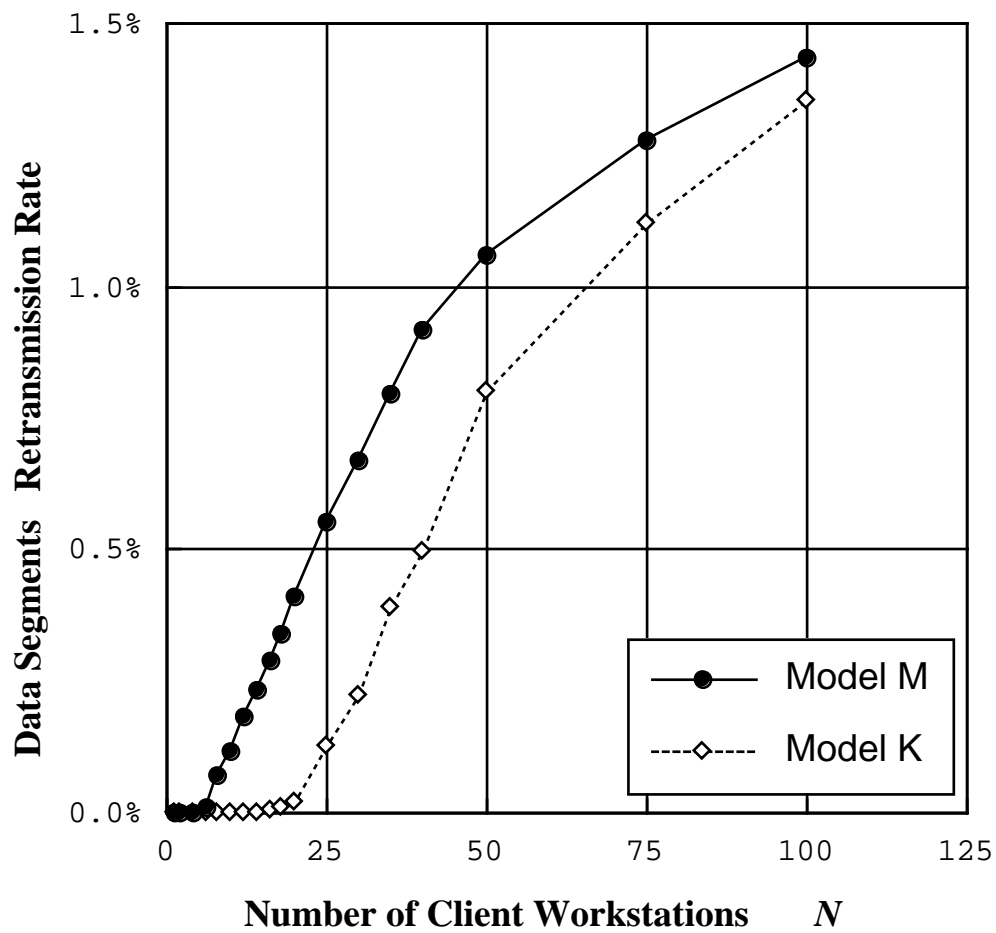


図 4.10: 平均データセグメント再送率 (伝送速度 10Mbps , ファイルサイズ 1MB)  
Fig. 4.10. Mean data segment retransmission rate. (line speed 10Mbps, file size 1MB)

## 物理層伝送速度 100Mbps の場合

測定結果を表 4.6 , 4.7 に示す .

## 平均ファイル転送時間と平均実効ファイル転送速度

$N$  に対する平均ファイル転送時間  $\overline{T_{trs}}$  の変化を図 4.11 に , 平均実効ファイル転送速度  $\bar{v}$  の変化を図 4.12 に示す . ファイルアクセス集中モデル K の場合 ,  $\overline{T_{trs}}$  ,  $\bar{v}$  は  $N$  が増加しても増加しない . これは大部分の CWS の通信が他の CWS の通信と干渉することなく通信できたためである . 一方アクセス集中モデル M の場合の結果は物理層伝送速度 10Mbps の場合のシミュレーションでのモデル K の結果と良く似ている .  $N < 20$  では  $\overline{T_{trs}}$  は  $N = 1$  のときとほぼ同じ値を保っており ,  $N > 20$  で急激に増加が始まる . しかし ,  $\overline{T_{trs}}$  はそれが最も大きくなる  $N = 100$  のときにも 4.4 秒程度であり , 実用的な範囲に含まれる .

このことから , 教育用 LAN で伝送速度 100Mbps の伝送媒体を用いる場合 , 1 セグメントに 1 台の SWS とともに 100 台以上の CWS を接続しても , 実用的な通信速度を保つことがわかる . 但し , 本シミュレーションでは , SWS の処理速度を十分に速いものとしているため , 実際には通信速度はこのシミュレーション結果よりも低くなると考えられる .

表 4.6: 平均ファイル転送時間と実行ファイル転送速度 (伝送速度 100Mbps , ファイルサイズ 1MB)

N	平均ファイル転送時間 [sec]		平均実効ファイル転送速度 [Kbps]	
	Model M	K	M	K
1	0.272	0.272	30866	30932
2	0.269	0.269	31183	31313
4	0.276	0.270	30575	31111
6	0.273	0.271	30778	30983
8	0.289	0.273	29952	30789
10	0.281	0.272	29988	30926
12	0.294	0.271	29469	31026
14	0.326	0.277	28824	30808
16	0.340	0.273	28429	30841
18	0.331	0.283	28076	30726
20	0.442	0.275	26496	30758
25	0.503	0.275	25147	30575
30	0.750	0.290	23083	30305
35	0.835	0.289	21221	30159
40	1.067	0.293	18450	29864
50	1.537	0.301	14917	29576
75	3.174	0.342	7292	28059
100	4.409	0.469	4651	26604

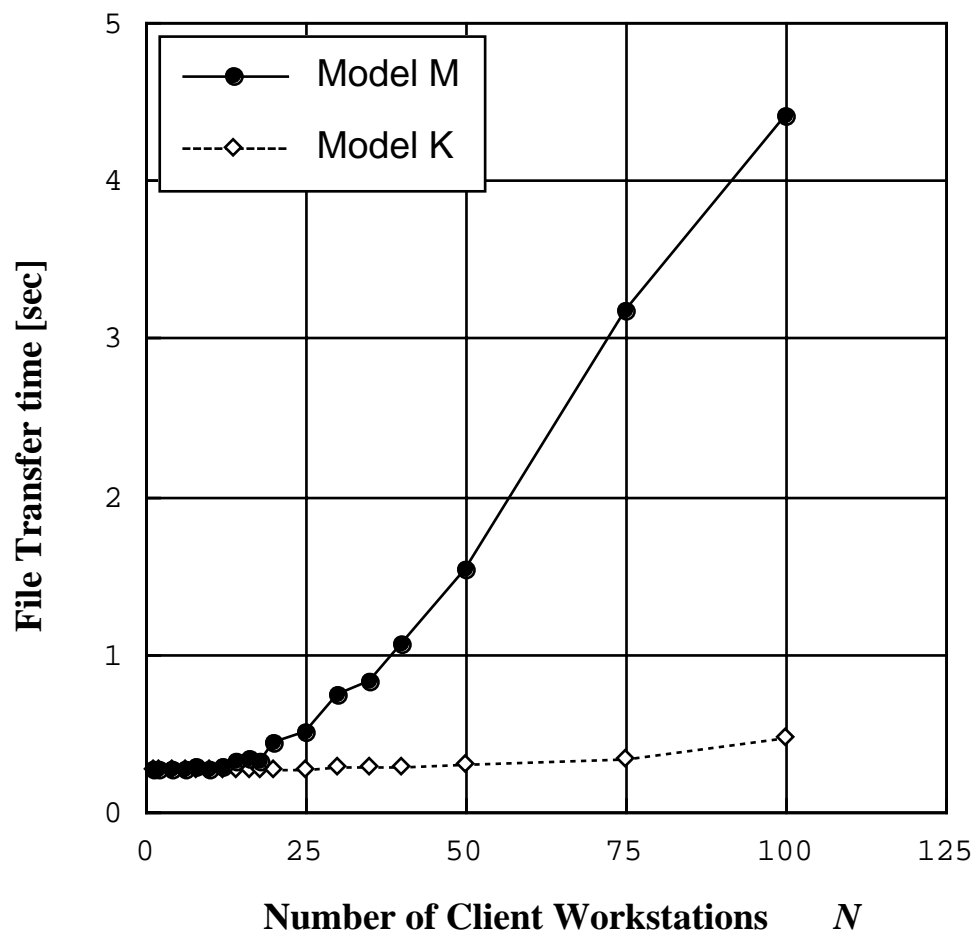


図 4.11: 平均ファイル転送時間 (伝送速度 100Mbps, ファイルサイズ 1MB)  
Fig. 4.11. Mean file transfer time. (line speed 100Mbps, file size 1MB)

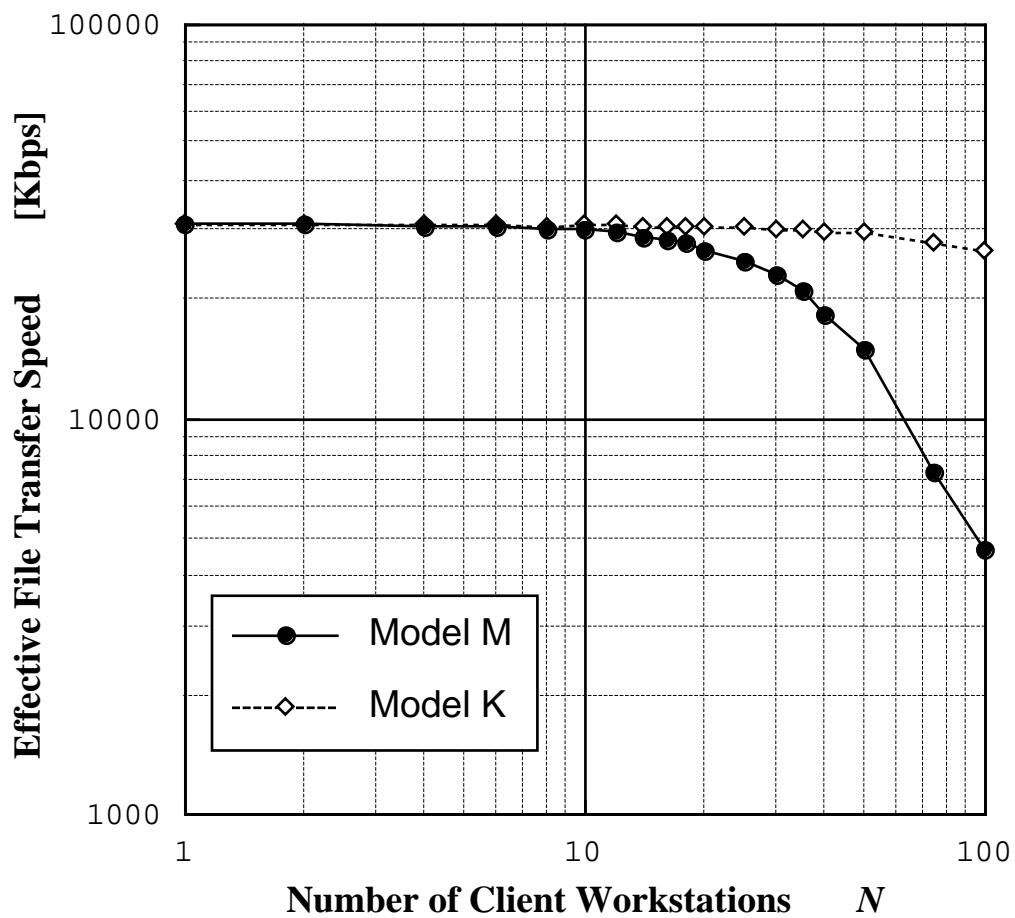


図 4.12: 平均実効ファイル転送速度 (伝送速度 100Mbps, ファイルサイズ 1MB)  
Fig. 4.12. Mean effective file transfer speed. (line speed 100Mbps, file size 1MB)

表 4.7: 平均フレーム衝突率と平均フレーム送信エラー率 (伝送速度 1Mbps, ファイルサイズ 1MB)

N	平均フレーム衝突率		平均フレーム送信エラー率	
	Model M	K	M	K
1	0.0000	0.0000	0.0000	0.0000
2	0.0144	0.0064	0.0000	0.0000
4	0.0515	0.0077	0.0000	0.0000
6	0.0577	0.0168	0.0000	0.0000
8	0.0965	0.0436	0.0000	0.0000
10	0.1156	0.0319	0.0000	0.0000
12	0.1523	0.0291	0.0000	0.0000
14	0.1844	0.0371	0.0001	0.0000
16	0.2043	0.0452	0.0001	0.0000
18	0.2389	0.0482	0.0001	0.0000
20	0.2724	0.0519	0.0002	0.0000
25	0.3196	0.0731	0.0004	0.0000
30	0.3574	0.0869	0.0006	0.0000
35	0.4037	0.1052	0.0009	0.0000
40	0.4457	0.1192	0.0014	0.0000
50	0.4811	0.1363	0.0022	0.0000
75	0.5474	0.2221	0.0050	0.0001
100	0.5782	0.2883	0.0071	0.0003

## 平均フレーム衝突率と平均フレーム送信エラー率

$N$  に対する平均フレーム衝突率の変化を図 4.13 に, 平均フレーム送信エラー率の変化を図 4.14 に示す. アクセス集中モデル M の場合には  $N$  の増加とともに  $\bar{c}$  が増加し,  $\bar{c} > 0.3$  になった時に送信エラーが発生するようになる. この傾向は物理伝送速度 10Mbps の場合と同じである. しかし,  $\bar{c}$  の値は 10Mbps の場合の 1/10 程度の低い値である. アクセス集中モデル K の場合には  $N = 100$  でも  $\bar{c} \leq 0.3$  であり, 送信エラーはほとんど発生していない.

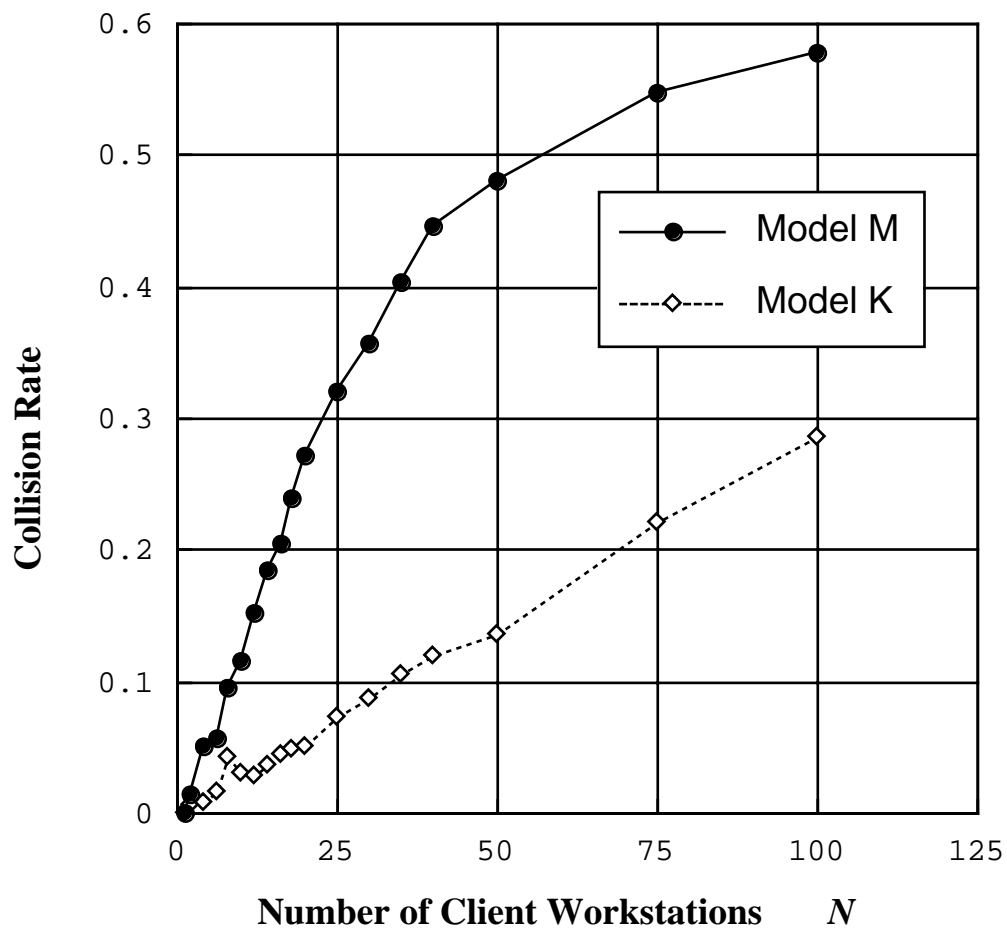


図 4.13: 平均フレーム衝突率 (伝送速度 100Mbps, ファイルサイズ 1MB)  
Fig. 4.13. Mean frame collision rate. (line speed 100Mbps, file size 1MB)

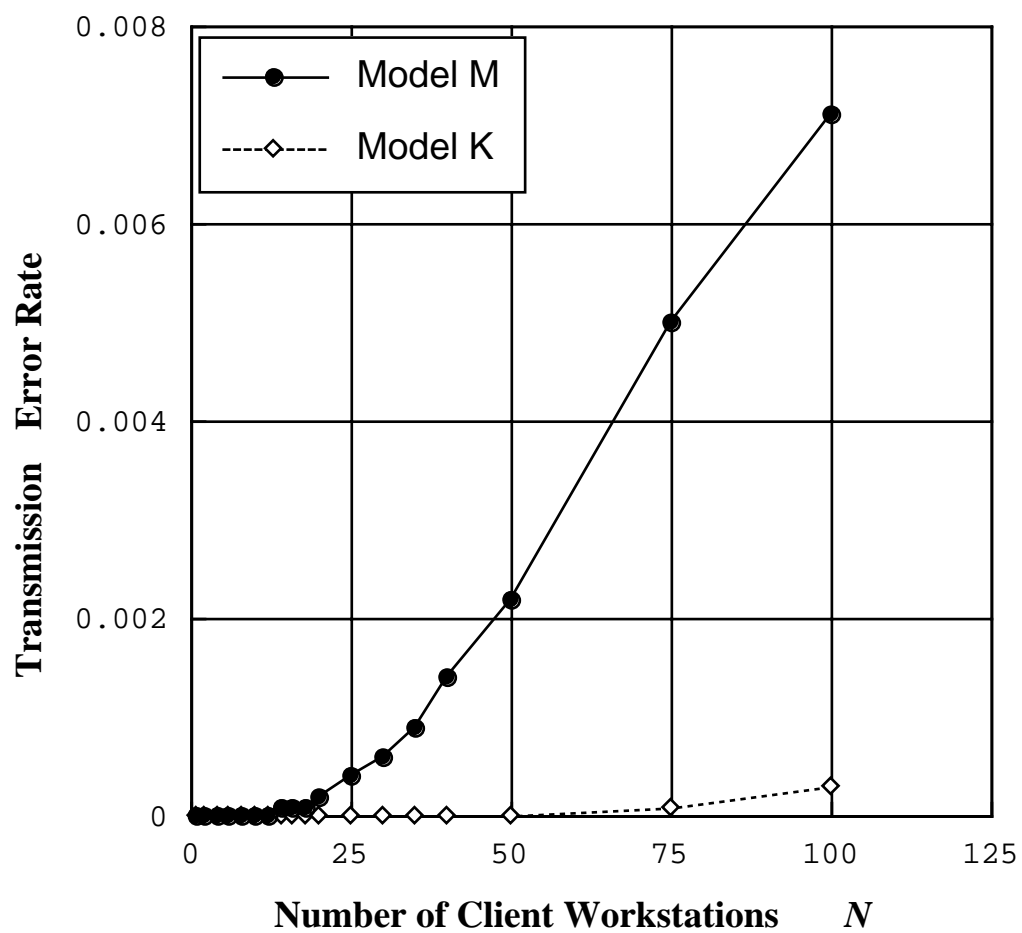


図 4.14: 平均フレーム送信エラー率 (伝送速度 100Mbps, ファイルサイズ 1MB)  
Fig. 4.14. Mean frame transmission error rate. (line speed 100Mbps, file size 1MB)

## 4.3 教育用 LAN における実システム測定

今回設計したシミュレーションシステムの評価を行うため、教育用 LAN の実システムにおけるネットワークアクセス集中時のファイル転送時間を測定した<sup>[32][33]</sup>。測定には名古屋大学情報処理教育センターの設備<sup>[2][4]</sup>を利用した。

### 4.3.1 測定方法

#### 測定プログラム

一般にはトランスポート層に TCP を用いるファイル転送プロトコルとして FTP (File Transfer Protocol)<sup>[34]</sup> が用いられている。しかし、FTP プログラムが実行時に表示するファイル転送時間は、必ずしも実際にファイル転送に要した時間を正確に表示するものではない。このため今回の測定は、新たにファイル転送時間計測用のプログラムを作成して行った。なお測定にあたっては、ntp(network time protocol)により各 WS の時刻同期を行った。

測定用プログラムは、(i) ファイル転送要求を受けてファイルを送信するサーバプログラムと (ii) ファイル転送要求を行いファイル転送要求からファイル転送の終了までの時間を計測するクライアントプログラム、の二つから構成されている。図 4.15 にプログラムの通信手順を示す。以下、測定用プログラムの動作について説明する。

1. まずサーバプログラムはクライアントからの TCP コネクションの確立要求を受けするため待機している。
2. クライアントは与えられた送信開始時刻になると、サーバに TCP コネクションの確立を要求する。また、この時刻をファイル転送開始時刻として記録する。
3. サーバはクライアントからのコネクション確立要求に応答する。その後 3 Way Handshake によりクライアントとの通信路が確立される。

4. クライアントはサーバにファイル転送要求を送信する．プログラムが実際に送信するのは，14 byte のデータを含む1つの TCP セグメントである．
5. サーバはクライアントからの要求に応じてファイルを送信する．
6. クライアントはファイルをすべて受信すると，コネクションを閉じる．現在時刻をファイル転送終了時刻として記録する．
7. サーバはファイルをすべて送信し終わるとコネクションを閉じる．TCP の機能により，クライアント・サーバの両方がコネクションを閉じた後にコネクションは切断される．

実際にはサーバへのアクセスが集中した場合，クライアントプログラムがサーバとのコネクションの確立を要求しても，サーバでコネクションの確立が拒否される場合がある．このときはクライアントプログラムはコネクションが確立されるまで，2でコネクションの確立を試みる．また，シミュレーションで測定した各 CWS のファイル転送時間  $T_{trs_i}$  は4から6までの時間に対応しているが，実システムでは  $T_{trs_i}$  として2から6までの時間を測定している．このため，実システムでの  $T_{trs_i}$  の測定値はシミュレーションで得られた  $T_{trs_i}$  よりも若干大きな値になっている．

クライアントのファイル転送開始時刻およびファイル転送時間の精度は  $\pm 16.667$  msec である．

#### 測定環境

測定は本学情報処理教育センターの演習室の環境で行った．各演習室には20～50台のクライアントWSと1台のプリントサーバWSが伝送速度10 Mbpsの10BASE-5により相互接続されている．各演習室のネットワークは物理的に完全に別のネットワークになっており，それぞれがマルチネットワークインターフェースをもつファイルサーバWSに接続されている．ファイルサーバWSには施設の全利用者

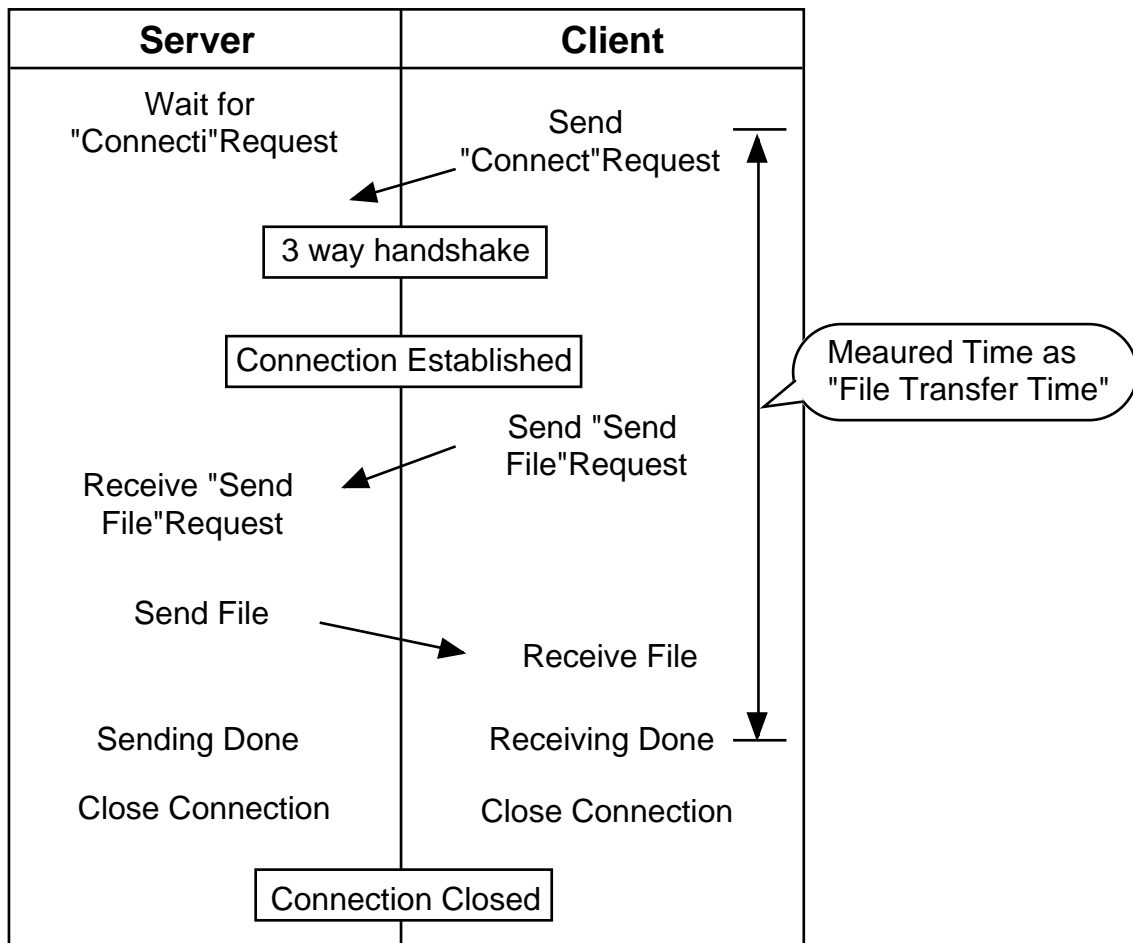


図 4.15: ファイル転送時間計測プログラムの通信手順

Fig. 4.15. Communication procedure of file transfer time measuring program.

表 4.8: 測定に使用した WS の性能値

	型名	処理速度	主記憶
クライアント WS	S-4/EC	21 MIPS	16MB
ファイルサーバ (プリントサーバ WS を代用)	S-4/IX	28.5 MIPS	16MB

の個人データ及び共有データが格納されており，通常の利用時には各演習室のクライアント WS からファイルサーバ WS にファイルアクセスが行われる．

今回の測定では，演習室のプリントサーバ WS をファイルサーバとし，同演習室のクライアント WS からプリントサーバ WS に対してファイルアクセスを行なうものとした．ファイルサーバ WS を測定時に利用しなかったのは，他の演習室や外部からのログインによるファイルサーバ WS の負荷の変動による測定結果への影響を最小限にするためである．表 4.8に今回使用した WS の性能値を示す．

#### 測定条件

以下の条件に従って，ファイルアクセス集中時の各クライアントのファイル転送時間を測定した．

クライアント WS 台数  $N$  1 ~ 50 台．

転送ファイル 容量 1 MB のランダムなバイト列．

測定回数  $N < 25$  のとき 10 回，これより多い時には 5 回の測定．

アクセス集中モデル 3.5で示したアクセス集中モデルに従い，シミュレーションモデルと同じルーチンを使用して，各 CWS での通信開始時刻を決定した．

### 4.3.2 測定結果と考察

表 4.9, 図 4.16, 4.17に平均ファイル転送時間  $\overline{T_{trs}}$  の実システム測定結果とシミュレーションによる結果を示す.

実システムの測定結果とシミュレーションの結果は, アクセス集中モデル M, K どちらの場合にも同様の傾向を示した. しかし,  $N$  が大きい場合 (モデル M のとき,  $N > 10$ , モデル K のとき  $N > 25$ ) には両者の結果の開きが大きい. これには以下のような理由が考えられる.

- 実システムでの測定プログラムでは, クライアントからのコネクション確立要求がサーバから許されなかった場合に, コネクションが確立するまで連続してコネクション確立要求メッセージを送り出す. このため, ネットワークに余分なメッセージが流れることにより, 全体の通信時間が長くなった.
- シミュレーションモデルではイーサネットモジュールの送信バッファの容量と, サーバ上で同時に動作する上位アプリケーションの数に制限を設けなかった. 実システムではこれらの値には制限があり, 今回の測定時にも, サーバプログラムのプロセスが同時に動作していなかったことが考えられる.

表 4.9: 実システムでの測定結果  
平均ファイル転送時間 [sec]

N	平均ファイル転送時間 [sec]			
	Model M		Model K	
	実測値	シミュレーション	実測値	シミュレーション
1	1.74	1.083	1.72	1.087
2	1.92	1.362	1.85	1.085
4	2.56	2.138	1.84	1.340
6	3.55	3.504	2.12	1.395
8	5.07	4.725	2.22	1.658
10	6.94	5.867	2.55	2.000
12	8.72	7.455	2.62	2.099
14	11.22	8.975	2.96	2.323
16	13.21	10.420	3.19	3.068
18	15.43	11.932	3.59	3.264
20	17.60	13.531	4.41	3.802
22			5.53	
24			6.82	
25	23.13	17.505		6.091
26			9.19	
30	28.84	21.622	13.13	8.666
35	34.04	26.151	19.31	13.089
40	38.84	30.796	25.75	16.146
45	44.07		31.22	
50	45.49	38.960	36.63	25.207
75		57.649		45.221
100		74.650		65.650

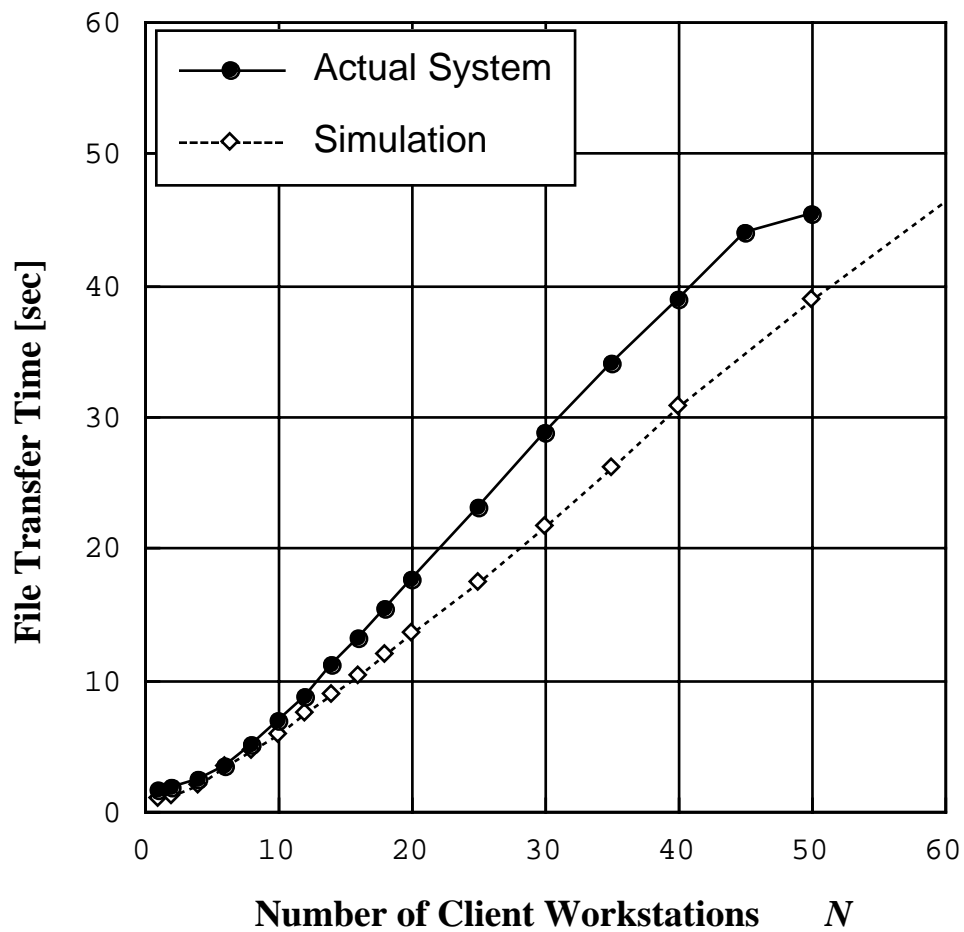


図 4.16: 実システムによる平均ファイル転送時間の測定結果 (Model M)

Fig. 4.16 Measured value of mean file transfer time on actual system. (Model M)

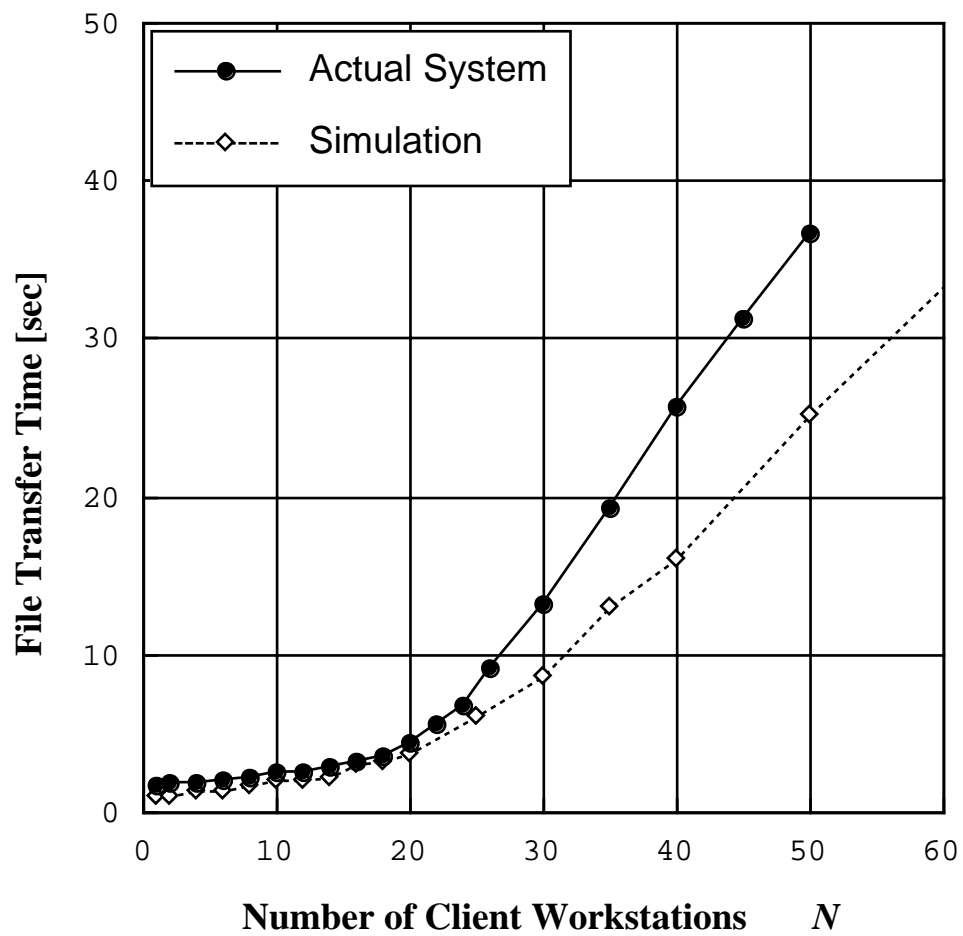


図 4.17: 実システムによる平均ファイル転送時間の測定結果 (Model K)

Fig. 4.17 Measured value of mean file transfer time on atual system. (Model K)

# 第 5 章

## まとめ

負荷が時間的に集中した場合でも，複数の通信プロトコルを通して通信を行うネットワークの通信量を正確に評価するモデルとして，イベント駆動によるシミュレーションモデルを提案した．このモデルでは ISO/OSI 参照モデルのトランスポート層以下のプロトコル，TCP/IP および IEEE 802.3 標準の CSMA/CD をそれぞれモジュール化して設計した．各モジュールはプロトコルの動作を厳密にモデル化するものである．これにより，システム実利用環境におけるアプリケーション利用時のネットワーク特性を正確に評価することを可能にした．また，シミュレーションモデルの信頼性は，従来モデルとの比較実験，および実システム測定実験との比較により確かめられた．

以下，本モデルによるシミュレーション結果から得られる結論とモデルの問題点，今後の展望について述べる．

### 5.1 教育用 LAN の構成法

本モデルを用い，教育用 LAN におけるネットワーク負荷集中時のシミュレーションを行った結果，教育用 LAN における適切なネットワーク構築手法として以下のような結論が得られた．

教育用 LAN の負荷集中時の特性 Ethernet LAN で構成されたクライアント・サーバシステムで、ファイルサーバにファイル転送要求が集中した場合、実効ファイル転送速度はファイル転送を要求するクライアント数  $N$  がしきい値  $N_t$  を越えると急激に減少する。 $N_t$  の値は Ethernet の伝送速度とファイルアクセス集中率の比および転送ファイルサイズにより変化する。

伝送速度 10Mbps の LAN 転送ファイルサイズの上限が 1MB、ユーザのコマンド入力方法がキーボードの場合は  $N_t = 20$  である。またコマンド入力方法がマウスの場合は  $N_t = 4$  である。実際の利用環境ではシミュレーションで想定した場合のように、すべてのクライアントで同一サイズのファイルを一度に転送する機会はそれほど多くないと考えられる。このため、教育用 LAN の 1 セグメントに一台のサーバに接続するクライアント WS の台数の目安は 10 数台程度であるといえよう。これ以上のクライアントを接続する場合には、ネットワークを複数のセグメントに分割する必要がある。

伝送速度 100Mbps の LAN マウスによるコマンド入力が行われるときでも、伝送速度に対するアクセス集中度の比は伝送速度 10Mbps の場合に比べて低くなる。 $N_t$  の値は伝送速度 10Mbps の場合に比べ大きくなり、 $N_t = 20$  である。但し、伝送速度が速いため  $N = 100 > N_t$  でも十分実用的な実効ファイル転送速度が得られる。

以上の結論はファイルサーバの能力が十分に高い場合を想定したシミュレーションにより得られている。実際には、100BASE-TX のように物理層の伝送速度が高速な場合には、サーバの処理能力が通信速度に大きく影響を与えられ、従って更に信頼性における評価を行うためには、サーバの処理能力を上位層モデルに組み込んでシミュレーションする必要がある。

## 5.2 シミュレーションモデルの問題点と今後の展望

今回設計したモデルではシミュレーションの正確さを重視して、各通信プロトコルを詳細にモデル化した。このためシミュレーション実行時間の削減のために物理伝送路の総延長に制限を設けている。実際の LAN ではブリッジやルータなどを含む複雑なネットワーク構成がなされており、伝送路の総延長は長い。このようなシステム全体の性能評価を行うにはイーサネットモジュールの構成を再検討する必要がある。また、本モデルではイーサネットモジュールのキャリアセンスを行なう部分で計算量がノード数  $N$  に対して  $O(N^2)$  である。数百台規模のシステムの評価を行うにはシミュレーション実行時間の削減のため、各プロトコルのモデルの単純化や、複数のステーションのモデルをグループ化してモデル化するなどの検討が必要である。

今後は以上の問題点を検討した上で、教育用高負荷 LAN における NFS の基本モデルとして考える階層化 NFS の設計と評価への応用を考えている。



# 謝 辞

本研究を進めるに当たって、多くの方々の御指導、御協力がありました。この場を借りて深く感謝の意を表したいと思います。

本研究の機会を与えて戴いた、名古屋大学情報処理教育センター長の毛利佳年雄教授に感謝します。毛利教授には私の博士進学審査の主査を担当して戴きました。あわせて感謝の意を示します。

指導教官として研究指導を行ってくださった、名古屋大学情報処理教育センターの岡田稔助教授に感謝します。また研究指導を行って戴いた、名古屋大学情報処理教育センターの岩田晃助手、愛知県立大学の櫻井桂一教授に感謝の意を示します。

名古屋大学情報処理教育センターの池田幹男助手、松本哲也助手には討論の場で研究に対し有用な助言を与えて戴きました。また、名古屋大学情報処理教育センター岡田研究室の大学院生、卒業研究生の皆さんにも討論の場で貴重な意見を戴きました。みなさんに感謝します。

平成4年度卒業生で現NTTデータ通信の岡崎香織氏と平成5年度卒業生で現三井物産の川村尚央氏の卒業研究は、本研究の基礎データとして大変有用でした。お二人に感謝の気持ちを示します。



## 参考文献

- [1] 中山, 大西, 末永, 有田: “工学系学生のための情報処理集合教育環境の設計と構築”, 情処学論, **35**, 11, pp. 1442–1452 (1994).
- [2] 岡田, 岩田, 松本, 池田: “教育用大規模分散型 WS システムの構成方法”, 国立大学情報処理教育センター協議会・情報処理教育に関する研究会, DSM–11020, (1994).
- [3] 荻原, 山口, 大川, 松浦, 福岡: “情報処理教育におけるマルチメディアコミュニケーション”, 国立大学情報処理教育センター協議会・情報処理教育に関する研究会, **16**, pp. 25–34 (1992).
- [4] 松本, 池田, 岩田, 岡田: “教育用大規模分散型 WS システムの利用状況”, 情処研資, DSM–9411021, (1994).
- [5] 石田, 高原: “イーサネット上のトラヒック特性によるパケット発生モデルについて”, 信学論, **J78–B–I**, 11, pp. 664–671 (1995).
- [6] Sun Microsystems, Inc.: “NFS: Network file system protocol specification”, RFC 1094 (1989).
- [7] E. Levy and A. Silberschatz: “Distributed file systems: Concepts and examples”, ACM Computing Surveys, **22**, 4, pp. 321–374 (1990). (邦訳 天海良治,

- 分散ファイルシステム：その概念と実例, bit 別冊コンピュータサイエンス, 共立出版, pp. 141–191, (1992)).
- [8] E. Drakopoulos and M. J. Merges: “Performance analysis of client-server storage systems”, *IEEE Trans. Comput.*, **41**, 11, pp. 1442–1452 (1992).
- [9] O. C. Ibe, H. Choi and K. S. Trivedi: “Performance evaluation of client-server systems”, *IEEE Trans. Parallel and Distributed Systems*, **4**, 11, pp. 1217–1229 (1993).
- [10] J. C. Bolot, A. U. Shankar and B. D. Plateau: “Performance analysis of transport protocols over congestive channels”, *Performance Evaluation*, **11**, 1, pp. 45–66 (1990).
- [11] M. Murata and H. Takagi: “Two-layer modeling for local area networks”, *IEEE Trans. Commun.*, **36**, 9, pp. 1022–1034 (1988).
- [12] P. Marino and A. del Rio: “An accurate and fast CSMA/CD simulator”, *Microprocessing and Microprogramming*, **39**, pp. 197–190 (1993).
- [13] P. J. P. OReilly and J. L. Hammond: “An efficient simulation technique for performance studies of CSMA/CD local networks”, *IEEE J. Sel. Areas Commun.*, **SAC-2**, 1, pp. 238–249 (1994).
- [14] A. M. Law and M. G. McComas: “Simulation software for communications networks: The state of the art”, *IEEE Commun. Mag.*, **32**, 3, pp. 44–50 (1994).
- [15] 石原, 岡田, 櫻井: “情報処理教育用 LAN のネットワーク負荷解析”, *電気関係学会東海連大*, **801**, (1994).

- 
- [16] 石原, 岡田, 岩田, 櫻井: “教育用 LAN における通信量モデリングの一手法”, 信学技報, CAS94-88, pp. 29-36 (1995).
- [17] 石原, 岡田, 岩田, 櫻井: “イベント駆動方式による LAN 通信量解析モデル”, 信学論, J78-A, 8, pp. 961-964 (1995).
- [18] The Wollongong group: “Internetworking: An introduction”, The Wollongong group. Inc. (1989). (邦 監訳 白鳥則朗, “インターネットワーキング入門: TCP/IP から OSI へ, オーム 社, 1989).
- [19] 村田, 宮原: “通信トラヒック理論とその応用 システムのモデル化と 性能評価”, 信学誌, 77, 9, pp. 968-975 (1994).
- [20] J. Postel: “Internet protocol”, RFC 791, Information science institute/ University of Southern California (1981).
- [21] J. Postel: “User datagram protocol”, RFC 768, Information science institute/ University of Southern California (1981).
- [22] ANSI/IEEE Standard 802.3: “Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications”, IEEE Press (1993).
- [23] J. Postel: “Transmission control protocol”, RFC 793, Information science institute/ University of Southern California (1981).
- [24] R. Braden: “T/TCP – TCP extensions for transactions functional specification”, RFC 1644 (1994).
- [25] V. Jacobson, et al.: “TCP extensions for high performance”, RFC 1323 (1992).
- [26] J. Neagle: “Congestion control in IP/TCP internetworks”, RFC 896 (1984).

- [27] S. J. Leffler, M. K. McKusick, et al.: “The design and implementation of the 4.3BSD UNIX operating system”, Addison-Wesley publishing company, Inc. (1989). (邦訳 中村明 他, UNIX 4.3 BSD の設計と実装, 丸善 (1991)).
- [28] D. D. Clark: “Window and acknowledgment strategy in TCP”, RFC 813, MIT laboratory for computer science (1982).
- [29] 川村: “ネットワーク通信量の計測とその検討”, 名古屋大学工学部卒業論文 (1995).
- [30] 石原, 岡田, 岩田, 櫻井: “シミュレーションモデルによる教育用 LAN の通信量解析”, 50 回情処全大, 1, pp. 147–148 (1995).
- [31] LAN MAN standards committee of the IEEE computer society: “IEEE draft standard for carrier sense multiple access with collision detection (CSMA/CD) access method & physical layer specifications: MAC parameters, physical layer, medium attachment units and repeater for 100 Mb/s operation (version 5.0)” (1995).
- [32] 石原, 岡田, 岩田, 櫻井: “教育用 LAN の通信量解析モデルの評価”, 電気関係学会東海連大, 610, (1995).
- [33] 石原, 岡田, 岩田, 櫻井: “教育用 LAN の通信量解析”, 1996 年情報学シンポジウム, pp. 17–23 (1996).
- [34] J. Postel: “File transfer protocol”, RFC 765 (1985).