

教育用 LAN における通信量モデリングの一手法

石原進[†], 岡田稔^{††}, 岩田晃^{††}, 櫻井桂一^{†††}

[†] 名古屋大学大学院工学研究科電気系専攻, 名古屋市

^{††} 名古屋大学情報処理教育センター, 名古屋市

^{†††} 愛知県立大学, 名古屋市

あらまし

本論文では、ネットワーク負荷の時間的集中度が高い情報処理教育用 LAN 環境において、ネットワーク通信量を正確に予測可能なネットワークトラフィックモデルを提案する。本モデルでは、利用する通信プロトコルに忠実に従うネットワーク上の各ノードのモデルを仮定し、これらをイベント駆動方式によって、並列に動作させる。各通信プロトコルをモジュール化して設計したことにより、さまざまな通信プロトコルの組み合わせに対応可能である。また、本モデルによるシミュレーションにより、教育用 LAN の 1 セグメントあたりのクライアント数の目安を得ることができた。

キーワード シミュレーション技法、トラフィック解析、LAN、WS システム

A METHOD OF TRAFFIC MODELING ON LOCAL AREA NETWORK FOR EDUCATION

Susumu ISHIHARA [†], Minoru OKADA ^{††}, Akira IWATA ^{††} and Keiichi SAKURAI ^{†††}

[†]Dept. of Electric Eng., School of Eng., Nagoya University, Nagoya, 464-01, Japan

^{††} Education Center for Information Processing, Nagoya University, Nagoya, 464-01, Japan

^{†††}Aichi Prefectural University, Nagoya, 467, Japan

Abstract

In this paper we propose a method of network traffic modeling that the traffic load of LAN in time is very high. In this method, an individual model of nodes on LAN performs target communication protocols consist of datalink layer and transport/network layers. The model works with the event driven state transition, then we can analyse network traffics with integration of transition times. The suitable number of workstations in a segment on high-loaded LAN such as educational computer system is shown by this simulation model.

Keywords Simulation method, Local Area Network, Traffic Analysis, Workstation System

1.はじめに

Local Area Network(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、国内の情報処理教育環境においても大規模分散システムを導入する施設が現われ出した^{[1][2][3][4]}。しかし、教育環境での分散システムの利用の歴史が浅く、また教育環境独自のLAN利用形態が、一般に普及している研究用や実務用のシステムの利用形態と異なることもあり、教育利用に適したLAN構築の手法が確立されるには至っていない。

ところで、情報処理教育環境におけるネットワーク負荷の特徴として、ネットワークへの負荷の時間的な集中度が高いことが挙げられる。例えば、プログラミングの授業においては、教官の指示の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、コンパイラプログラムがネットワーク上のファイルサーバに置かれているとすると、一時的にファイルサーバへのアクセスが集中し、ターンアラウンド時間が大幅に増加する。それに対して研究用や実務用のシステムでは、ネットワーク上の資源へのアクセスは時間的に分散しているため、ネットワーク負荷が極端に集中することは稀である。

Drakopoulosらはネットワークファイルシステムの評価モデルを設計したが^[5]、長時間におけるネットワーク平均遅延をサービス時間とした待ち行列でネットワークをモデル化している。この方法は、長時間の平均的なネットワークの挙動を予測するには適している。しかし、短時間に急激に負荷が上昇した場合、その時点のネットワークの挙動を正確に予測するのは困難である。

本論文では、イベント駆動方式によるネットワークトラフィックモデルを提案する。すなわち、利用する通信プロトコルに忠実に動作する、ネットワーク上の各ノードのモデルを仮定し、これらをイベント駆動方式で並列に動作させるものである。これにより、時間的に高い負荷集中時でのネットワーク通信量を正確に予測できるようにする。

以下、2章では本手法によるシミュレーションモデルと各通信プロトコルの具体的な実装方法について述べる。3章ではシミュレーション結果の例を示す。

表 1. 通信プロトコル

トランスポート層	TCP, UDP
ネットワーク層	IP
データリンク層	Ethernet
物理層	同軸ケーブル 10 Base 5

2.シミュレーションモデル

2.1 モデル化の方法

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を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^{[6][7]}。

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とEthernetを仮定し、OSI階層モデルの物理層からトランスポート層までをモデル化している(表1)。この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。

各通信プロトコルは次のようにモデル化する。

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動作を行うので、上位アプリケーションの動作自体がデータ転送時間に及ぼす影響は少ない。そこで、上位アプリケーションの動作は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を処理する時間のみ関係するものとして、これを固定率サービスセンタとしてモデル化する。一方、トランスポート層のプロトコルにUDP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行う。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、再送信フロー制御、輻輳制御を行う。このため、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 Ethernet の 1 セグメント内で考えた場合に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。

以上より、ネットワーク層以上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、一つのモジュール (TCP モジュール) として扱う。

データリンク層の Ethernet は、CSMA/CD 方式により伝送路にビット列の送信、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。Ethernet のビットレートおよび信号速度は、物理層である同軸ケーブルによって決まる。同軸ケーブルに 10 Base 5 を用いた場合には、ビットレートは 10Mbps であり、信号速度 $v = 0.66c$ (c は光速) である。Ethernet の機能は一つのモジュール (Ethernet モジュール) として扱う。

図 1 に本モデルの概念図を示す。クライアントの各ノードはそれぞれ TCP モジュールと Ethernet モジュールを持つ。サーバノードは一つの Ethernet モジュールと、各クライアントに対応した TCP モジュールを持ち、それぞれの TCP モジュールがクライアントの TCP モジュールと通信を行う。モデルにおいて時間経過の最小ステップを決定するのは Ethernet モジュールである。Ethernet モジュールは、自分のノード i と他のノード j の状態から、ノード i における次イベント発生時刻 t_{ni} を決定する。TCP モジュールは Ethernet モジュールの動作に対して受動的であり、その動作は各ノードで独

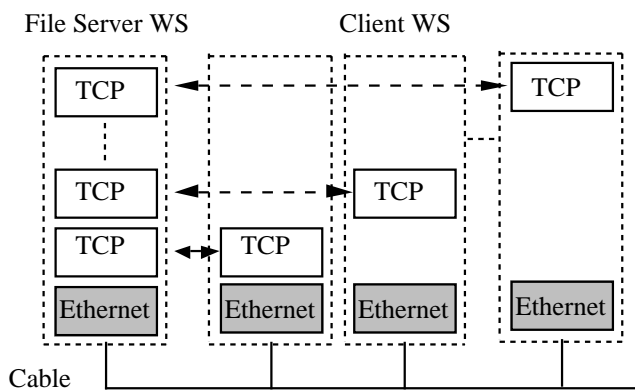


図 1 モデル概念図

Fig. 1 An outline of communication and network model.

立である。

2.2 Ethernet のモデル化

ここではデータリンク層のプロトコルである Ethernet のモデル化について説明する。ネットワーク上の各ノードの Ethernet モジュールは CSMA/CD 方式のアルゴリズムに従って状態遷移を行い伝送路にパケットを送信する^[8]。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。本モデルでは動作を厳密に模擬するため、伝送路上のキャリアの有無をネットワーク上のすべてのノードに共通な一つのパラメータでは判断せず、各ノード位置でのキャリアの有無をそれぞれ独立に判定する。

(a) 状態遷移

各ノードは図 2 の状態遷移図および図 3 の遷移条件に従って状態遷移を行う。ノード i の状態 $S_i \in \{SLP, CS, PR, TR, CD, WT\}$ は Ethernet の状態ではなく、本モデルで規定したものである。

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。受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。キャリアがないと PR に遷移する。

PR(Preamble): パケットの送信を開始する。この状態では、伝送路上に送り出したパケットは衝突の可能性がある。Ethernet の同軸ケーブル 10Base5 の一セグメント内での通信で衝突の危険があるのは、送信を開始してから、(1) 式で与えられる時間 T_d 以内であり、この値はジャム信号の送信時間とフレームギャップの時間の和 T_{jam} より小さい。

$$T_d = \frac{2l_{max}}{v} \quad (1)$$

ここで l_{max} は最長ノード間距離である。

CS→PR の状態遷移があつてから T_d だけ経過した時点で、この間に衝突があると CD に遷移する。衝突がなければ TR に遷移する。

TR(Transmission): パケットの送信を行う。衝突の可能性はない。パケット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、このパケットの送信において衝突回数が $n_{col} < 16$ の時は、送信を再試行するまでの待ち時間を計算し、WT に遷移

する。 $n_{col} = 16$ の時はこのパケットの送信をあきらめ、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。

(b) キャリアおよび衝突の検出

図 2 の状態遷移において CD でキャリアの検出、PR で衝突の検出を行う。衝突の検出は自分のノード以外が発生したキャリアによる信号波形の乱れを検出することであるから、モデル上ではキャリアの検出と同様の手法を用いることができる。ここでキャリア検出を行うノードを i 、その他

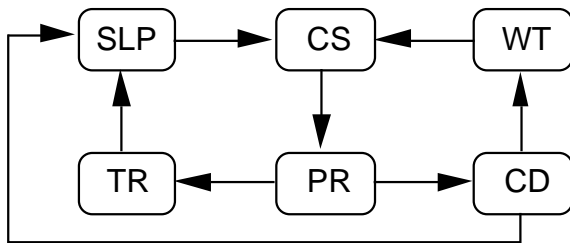


図 2 Ethernet モデルの状態遷移図

Fig. 2 State diagram of Ethernet model.

のノードを j とし、また、ノード j が最後にキャリアを発生した時刻を t_{cpj} 、キャリアを消滅させた時刻を t_{cdj} とする。ノード i 、ノード j の間の距離を l_{ij} とすると、現在時刻 t に関して次の条件が満たされるとき、ノード i にノード j が発生したキャリアが存在する (図 4 参照)。

$$t_{cdj} + \frac{l_{ij}}{v} < t_{cpj} + \frac{l_{ij}}{v} \leq t \quad \vee$$

$$t_{cpj} + \frac{l_{ij}}{v} \leq t \leq t_{cdj} + \frac{l_{ij}}{v} \quad (2)$$

状態 PR では衝突の検出とともに実際に衝突が発生した時刻 t_{col_i} を計算する。 $S_i = PR$ のとき、ノード i に最初に到着する他ノード j のキャリア到着時刻 t_{ca_i} は (3) 式で与えられる。

$$t_{ca_i} = \min_{j \neq i, t_{cdj} < t_{cpj}} \left\{ t_{cpj} + \frac{l_{ij}}{v} \right\} \quad (3)$$

このとき、次の条件を満たせば衝突が起きている。

$$t_{ts_i} \leq t_{ca_i} \quad (4)$$

		Next State					
		SLP	CS	PR	TR	CD	WT
Current State	SLP	/	Data exist in Send Buffer	/	/	/	/
	CS	/	/	No carrier $t_{ts_i} \leftarrow t$	/	/	/
	PR	/	/	/	$t_{n_i} = t_{ts_i} + T_d$ No Collision $n_{col} \leftarrow 0$	$t_{n_i} = t_{ts_i} + T_d$ Collision Compute T_{b_i} $n_{col_i} \leftarrow n_{col_i} + 1$	/
	TR	Packet Transmission complete	/	/	/	/	/
	CD	Jam end $n_{col_i} = 16$	/	/	/	/	Jam end $n_{col_i} < 16$
	WT	/	Waiting Retry done	/	/	/	/

note

Condition
Action

図 3 遷移条件と動作

Fig. 3 Condition and Action of state transmission.

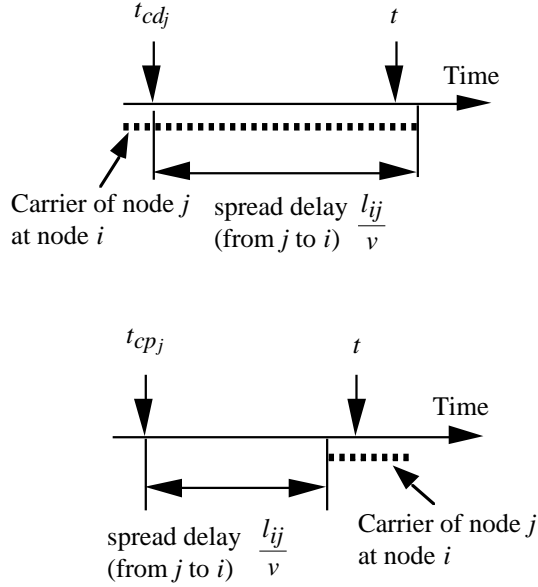


図 4 キャリアの存在の判定
Fig. 4 Carrier Sense.

ここで、 t_{ts_i} はノード i が CS→PR への状態遷移を行った時刻である。

t_{col_i} は (5) 式で与えられる。

$$t_{col_i} = t_{ca_i} \quad (5)$$

(c) 次イベント発生時刻の決定

Ethernet 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 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、 $S_j \in \{PR, TR, CD\}$ 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 $S_j \in \{SLP, CD, WT\}$ であれば、 t_{n_i} は (6) 式で与えられる。

$$t_{n_i} = \max_{i \neq j} \left\{ t_{cd_j} + \frac{l_{ij}}{v} \right\} \quad (6)$$

PR: 衝突の危険がある時間 T_d だけ経過させ

る。従って、 t_{n_i} は (7) 式で与えられる。

$$t_{n_i} = t_{ts_i} + T_d \quad (7)$$

TR: パケット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 (8) 式で与えられる。

$$t_{n_i} = t_{ts_i} + T_{p_i} \quad (8)$$

CD: 次イベントの発生は JAM 信号の送信が終了したときであるから、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 (9) 式で与えられる。

$$t_{n_i} = t_{col_i} + T_{jam} \quad (9)$$

WT: 乱数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。

$$t_{n_i} = t_{col_i} + T_{jam} + T_{b_i} \quad (10)$$

2.3 TCP のモデル化

TCP はトランスポート層のプロトコルとして、全二重の論理通信路の確立、パケット順番制御、再送信、フロー制御、輻輳制御を行う。

TCP の仕様書^[9]では、TCP の具体的な実装方法は提案という形で述べられており、細かい規定はなされていない。今回のモデルでは基本的な仕様および実装方法は仕様書に提案されているものに従うこととした。また、モデル化に必要なない緊急のデータ送信、論理通信路の異常からの復帰などの機能は除外した。

以下では、各機能についてモデル化の方法について述べる。

(a) TCP によるデータ送受信

TCP は 3 ウェイハンドシェイクと呼ばれる方法で、クライアント、サーバ間の論理通信路を確立し、その上で全二重通信を行う。論理通信路の確立、切断では、状態遷移を行いながらそれらに必要なセグメントをサーバ、クライアント間でやりとりする。モデルではこれらのセグメントによる通信量の増加も正確に予測するため、論理通信路確立と切断の動作を TCP のプロトコルに忠実にやっている。

TCP における通信の信頼性の保証は、送信側 TCP が送信したセグメントに対する応答を受信側 TCP より受け取ることによりなされる。もし、送信時にあらかじめ定めた時間内に受信応答が得られない場合 (再送タイムアウト) には、送信側が同

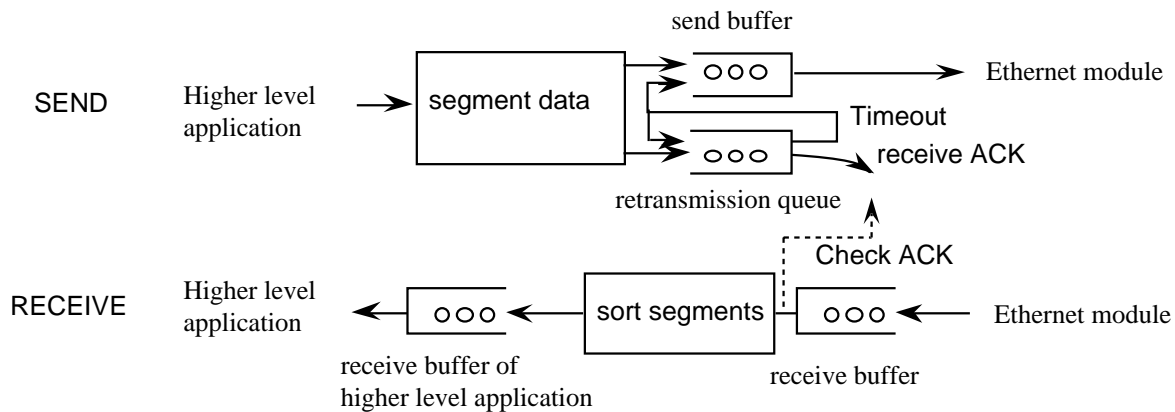


図 5 TCP モジュールでのデータフロー
Fig. 5 Dataflow of TCP module.

じセグメントを再送する。

論理通信路が開設されているときの TCP モジュール内のデータフローを図 5 に示す。

送信側 上位層からデータを受け取るとデータをセグメントに分割する。次に、送信バッファの空き容量、および受信側の上位層受信バッファの空き容量 (送信ウィンドウサイズ) が十分にあれば、ヘッダを付加してセグメントを送信バッファに送る。このとき送信セグメントが受信側からの応答確認を要する (データおよび同期、切断要求を含む) 場合にはセグメントを再送待ち行列に入れる。受信側からの応答確認が再送タイムアウトの時間内に受信できたときは、そのセグメントを待ち行列から取り除く。再送タイムアウトが起きたときは再送を行うため、そのセグメントを再送待ち行列から取り出し、送信バッファおよび再送待ち行列に入れる。

受信側 下位層から受信セグメントを受け取ると、セグメントの順序を整列して上位層受信バッファへ送る。次のセグメント送信時に、上位層に送ることができたセグメントまでの応答、および受信ウィンドウサイズ (上位層受信バッファの空き容量) を、セグメントのヘッダに付加して送信側 TCP モジュールに送る。

(b) 再送タイムアウトの設定

再送タイムアウトの値は通信効率に大きな影響を与えるため、その設定方法には多くの手法が提案されている。本方法では TCP の仕様書^[9]に参考として示されている指数平均による方法を用いる。この方法では、送信したセグメントに対する応答を受信するまでの時間 (round trip time) T_{rtt_i} を

測定しておき、これを元に (11) 式を与える。

$$T_{srtt_i} = \alpha T_{srtt_i} + (1 - \alpha) T_{rtt_i} \quad (11)$$

これより再送タイムアウト T_{rto_i} は (12) 式で与えられる。

$$T_{rto_i} = \min \{ T_{max}, \max \{ T_{min}, \beta \cdot T_{srtt_i} \} \} \quad (12)$$

また、再送が行われたときにはバックオフすなわち、そのセグメントの T_{rto_i} を再送が行われる度に 1.5 倍する方法をとる。

(c) パラメータ

TCP の動作は、モデル上では以下のパラメータにより決定される。

- ・ 上位アプリケーションの受信バッファサイズ
- ・ 送、受信バッファサイズ
- ・ 上位層が受信バッファのデータを処理するのに要する時間
- ・ データのセグメント化に要する時間
- ・ 受信セグメントのソートなどに要する時間
- ・ 再送タイムアウトを決定するパラメータ α, β ((11) (12) 式参照)

(d) 次イベント発生時刻の決定

TCP モジュールにおける次イベント発生時刻は、各 TCP モジュールにおける再送タイムアウト時刻、time wait timer 終了時刻、送信開始時刻、受信バッファの容量が空く時刻の最小値として与えられる。

3. シミュレーション

3.1 場面設定

ここでは本手法で作成されたモデルによるシミュレーション結果の一例を示す。

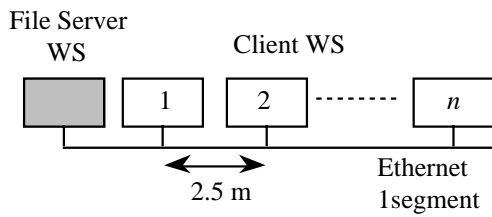


図 6 ネットワーク環境
Fig.6 Network Environment.

ネットワーク環境は次のように仮定する。図 6 に示すように、同軸ケーブル 10Base5 の 1 セグメント(500m) の一端にファイルサーバが接続され、そこから等間隔に 1~99 台のクライアント WS が接続されている。各クライアント WS は、上位層プロトコルに ftp を用いて、一斉にサーバ WS から各クライアント WS にファイル転送を行う。

また、教育環境でのネットワークアクセスの集中を模擬するため、各クライアントの送信開始時刻について以下のような仮定をする。情報処理教育の教室で、教官の指示の後に学生が一斉にネットワークにアクセスすることを考えると、このときの学生の反応にはばらつきがある。従って、学生の利用するクライアントにおけるファイル転送開始時刻 t_{si} は図 7 のように分布すると考えられる。ここで t_{ei} はファイル転送終了時刻である。モデルでは、 t_{si} の分布が図 8 に示すような正規分布 $N(\mu, \sigma^2)$ の正の部分に従うと仮定して、シミュレーションを行った。

TCP モジュールのパラメータは次のように設定した。

- ・上位アプリケーションの受信バッファサイズ

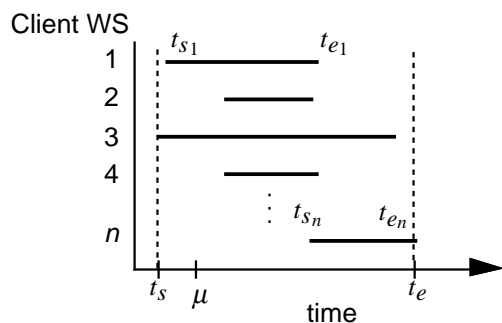


図 7 クライアント WS におけるファイル転送時間の分布

Fig. 7 Distribution of file transfer time on Client WS.

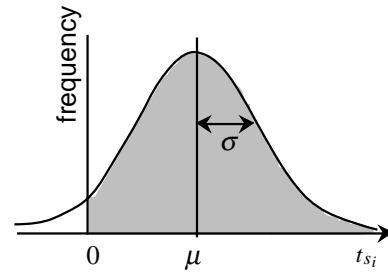


図 8 クライアント WS のファイル転送開始時刻の分布

Fig. 8 Distribution of start time of file transfer.

42 Kbytes

- ・TCP モジュールの送信バッファ、受信バッファの容量は十分に大きい。
- ・データのセグメント化、受信セグメントのソートに要する時間は 0
- ・再送タイムアウトを決定するパラメータ $\alpha = 0.8, \beta = 1.3$ ((11) (12) 式 参照)

3.2 シミュレーション結果

サーバから n 台のクライアントへのファイル転送時間の平均 $\overline{t_{ei} - t_{si}}$ の計算結果を図 9 に示す。ここでは、転送ファイルのサイズを 1Mbytes、クライアント WS の送信開始時刻の分布の平均 μ を 1.0 [sec] とし、標準偏差 σ を変化させてシミュレーションを行った。 $n = 8 \sim 16$ のあたりでファイル転送時間が急増していることがわかる。また、 σ が小さく、ネットワークアクセスの集中度が大きいときには、少ない n でのファイル転送時間の増加が見られる。さらに、ファイル転送期間 $[t_s, t_e]$ のサーバにおける全 Ethernet パケット送信試行回数を S_{packet} 、全衝突回数を S_{col} として、サーバにおける Ethernet パケット衝突率 S_{col} / S_{packet} を図 10 に示す。 n がある値を超えると衝突確率が急激に増加しており、ファイル転送時間と同様の傾向を示している。 S_{col} / S_{packet} の目安を 10% とすると、10 20 台あたりが衝突が多発する変化点と見られる。

4. むすび

本論文では教育用 LAN における通信量を評価するためのモデリングの一手法として、通信アルゴリズムに忠実に従うノードモデルを並列に動作させる方法を提案した。

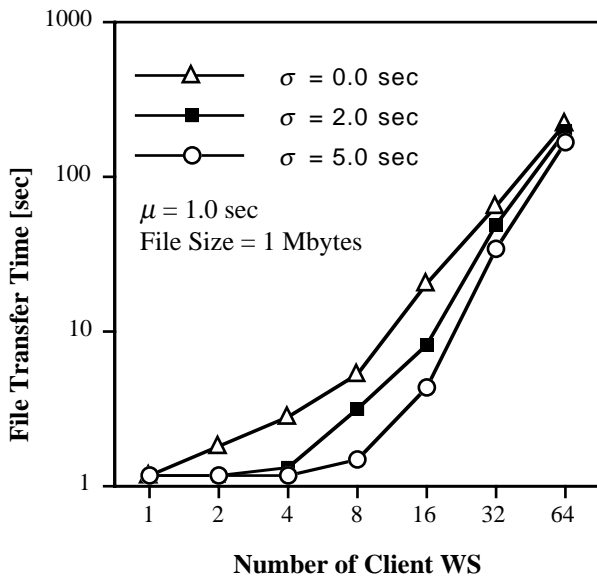


図 8 サーバからクライアントへのファイル転送時間の平均
Fig. 9 Average file transfer time from a server to clients.

この結果、教育環境においては 10Mbps の媒体ではクライアント数 15 台前後が一つの目安となることが示された。

現在は特定のプロトコルでのモデルで動作確認を行っている段階である。今後は扱うプロトコルの範囲を拡大し、リピータ、ルータ等を含む多様なネットワークポロジに対応できる通信量評価システムを構築する予定である。

またこのシステムを用いて、本学情報処理教育センターで設計する予定である教育用ネットワークファイルシステムの評価を行う予定である。

謝辞

本研究の機会を与えて戴く本学情報処理教育センター長毛利佳年雄教授、並びに討論戴く岡田研究室諸氏に感謝する。また、基礎検討^[6]を行った現 NTT データ通信の岡崎香織氏に感謝する。

参考文献

- [1] 中山仁, 大西淑雅, 末永正, 有田五次郎: "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", 情処論, Vol. 35, No. 11, pp. 2225-2238 (1994)
- [2] 岡田稔, 岩田晃, 松本哲也, 池田幹男: "教育用大規模分散型 WS システムの構成方法", 情処研資, DSM-9411020 (1994)
- [3] 荻原剛志, 山口英, 大川直, 松浦敏雄, 福岡秀和: "情報処理教育におけるマルチメディアコ

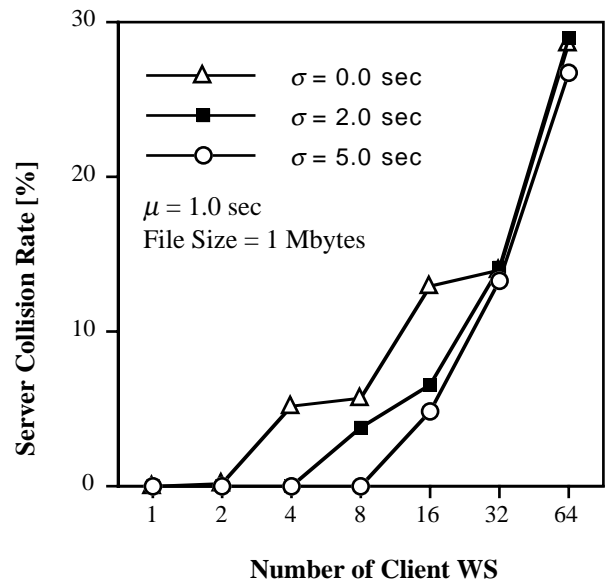


図 10 ファイル転送時のサーバ WS における Ethernet におけるパケットの衝突率
Fig. 10 Ethernet collision rate in file transfer at File Server.

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, No. 16, pp. 25-34 (1992)

- [4] 松本哲也, 池田幹男, 岩田晃, 岡田稔: "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] E. Drakopoulos and M. J. Merges: "Performance Analysis of Client-Server Storage Systems," *IEEE Trans. Comput.*, Vol. 41, No. 11, pp. 1442-1452 (Nov 1992)
- [6] 岡崎香織: "情報処理教育を指向したネットワークファイルシステム", 名古屋大学卒業論文 (Mar 1994)
- [7] 石原進, 岡田稔, 櫻井桂一: "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, p. 401 (1994)
- [8] ANSI/IEEE 802.3-1988, ISO 8802-3 (1989)
- [9] Information Science Institute University of Southern California: "Transmission Control Protocol," *RFC 793* (Sep 1981)