

高負荷多セグメントネットワークにおけるトラフィック評価

石原 進[†], 岡田 稔^{††}

[†]名古屋大学大学院工学研究科, ^{††}名古屋大学情報処理教育センター

あらまし 大学等の集合情報処理教育施設における LAN では、授業利用時におけるネットワークおよびサーバに対する負荷の時間的集中を避けることがシステム設計上の焦点となる。筆者らは、このような LAN の性能評価を効率良く行うイベント駆動によるシミュレーションモデルを開発した。本稿ではこのシミュレーションモデルのブリッジ、ルータ等を含む多セグメントネットワークへの拡張方法を示す。本方法では、ネットワーク層およびデータリンク層レベルでの経路表を用いることにより、多セグメントネットワークのトポロジを記述する。

Evaluation of Traffic on High-Loaded Multisegment Network

Susumu ISHIHARA[†], Minoru OKADA^{††}

[†]Graduate School of Engineering, Nagoya University

^{††}Education Center for Information Processing, Nagoya University

Abstract The traffic of LAN on large distributed workstation system for educational use has the high-load and high-trangiciency. We developed an event-driven simulation-model for analysis of such a high-loaded network. It has detailed models of multi-layer protocols, applications and user behavior model. In this paper a method to apply this model to multi-segment networks is discussed. Routing tables of network layer and datalink layer are used for describing the network topology of the multi-segment network.

1 はじめに

近年の LAN の普及により、大学等の集合情報処理教育施設において大規模分散システムを導入するケースが多くなった。集合情報処理教育施設の LAN (以下教育用 LAN) で特徴的なのは、授業利用時におけるネットワークおよびサーバへの時間的な負荷集中である。

このため教育用 LAN のシステム設計においては、サーバおよびクライアントにおける適切なファイル配置やネットワークの分割が必要である。しかし教育用 LAN の特性を考慮してネットワークトラフィックに関して調査した文献はほとんど無い。実システム上のアプリケーション利用時のネットワーク特性に関しては、実システム上の測定に基づくアプリケーション毎の packets 発生特性モデルの検討が [2] により行われているが、ユーザの動作特性を

考慮に入れたトラフィック評価は十分に行われていないのが現状である。

筆者らは教育用 LAN における負荷集中時においても、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可能なシミュレーションモデルを設計した [3][7]。このシミュレーションモデルでは、各階層のプロトコルモデルをイベント駆動で並列に動作させることにより、アプリケーション利用時の正確なネットワーク特性を評価可能である。しかし、このモデルではシミュレーション可能な Ethernet ネットワークの論理セグメントが 1 つに限定されており、ブリッジやルータ等を含む多セグメントネットワークの評価を行うことができなかった。本稿では、まず単一セグメントにおける本シミュレーションモデルの概要を示し、このシミュレーションモデルのブリッジ、ルータなどを含む多セグメントネットワークへの拡張について述べる。

本稿で示すシミュレーションシステムの拡張では、[3]のシミュレーションシステムが持つ、アプリケーションおよびユーザの動作を考慮に入れたシミュレーションが可能であると言う利点を失うことなく、このシステムを多セグメントネットワークに適用可能にする。これによって、トラフィック分割のために多セグメントに分割した教育用 LAN や遠隔端末室を持つ教育用システムなどの評価が可能になる。

2 シミュレータの概要

シミュレーションモデルは、ネットワークに過渡的な高負荷がかかった場合の特性を正確に予測するため、各ノードのユーザ、アプリケーション、通信プロトコルの動作を詳細に記述したノードのモデルをイベント駆動で並列に動作させるものである。シミュレーションモデルは3つの階層（ネットワークモデル、アプリケーションモデル、ユーザモデル）から構成されている。ネットワークモデルは、トランスポート層以下の階層のネットワークプロトコルの動作を表すモデルである。またアプリケーションモデルはアプリケーションの動作に応じた送信メッセージの発生を行うモデルであり、ユーザモデルによってその動作開始タイミングが指定される。

図1にシミュレーションモデルの全体構成を示す。各モデルはそれ自身が発生するイベント及び他のモデルから渡されるイベントによって動作を開始する。動作を終了すると必要があれば他のモデルにデータを受け渡し、他のモデルおよび自身のモデルに対する新しいイベント発生時刻をスケジュールする。各モデルはそれぞれ C および C++ で記述されており、現在モデル毎にライブラリ化作業が行われている。

以下各モデルの概要について述べる。詳細については [3][7] を参照されたい。

2.1 ネットワークモデル

本モデルでは 100 Mbps および 10 Mbps の Ethernet 上の TCP/IP ネットワークを仮定している。しかし、各階層のプロトコルはそれぞれモジュール化されているため、プロトコルモジュールを交換することにより様々なプロトコルの組合せを想定したシミュレーションが可能である。

2.1.1 Ethernet サブモデル

Ethernet サブモデルは、Ethernet の1つの論理セグメントごとに複数の入力待ち行列と一つの出力を持つ。入力是一位層モデルからのそのセグメントへのネットワーク出力である。Ethernet モデルはセグメント内で CSMA/CD に基づいてシミュレーションを行い、最終的な出力としてセグメント内にネットワーク入力をもつ上位層モデルに対する到着イベントを生成し、その上位層モデルに到着メッセージを渡す。

Ethernet モデルでは、各入力待ち行列が表すノード間の遅延と各ノード送信開始時刻に基づいて、正確に CSMA/CD のシミュレーションを行う。各ノードのモデルは、それぞれ最新の送信開始時刻、送信終了時刻、衝突回数、および送信状態を保持している。これらの値を全ノードモデル間で比較することにより、各ノード位置での伝送路状態および衝突の有無を判定して各ノードモデルにおける次のイベント発生時刻を決定する。

シミュレーションにおいて、その経過時間ステップがもっとも小さくなるのが Ethernet モジュールである。本モデルでは衝突の発生を効率よく判定するために、シミュレーション可能なネットワークのコリジョンドメインを以下の式で表される規模に制限している。

$$2 \min_{i \neq j} \{\tau^{ij}\} + \tau_{JAM} > 2 \max_{i \neq j} \{\tau^{ij}\} \quad (1)$$

τ^{ij} はノード i, j 間の伝播遅延であり τ_{JAM} はジャム信号の送信時間とフレームギャップ時間の和である。この制限内のネットワークであれば、フレーム送信中のノードを、衝突する可能性があるノードと衝突の可能性があるノードに明確に分けることができる。これにより、衝突の判定を衝突の可能性があるノードにのみに対して行えば良く、Ethernet サブモデル内の経過時間ステップを長くすることができ、効率的なシミュレーションが可能になる。

2.1.2 IP サブモデル

IP はネットワーク層のプロトコルとして、経路制御及びデータグラムのフラグメント化とリアセンブルを行う。単一セグメントのモデルでは、データグラムのフラグメント化の機能のみを持つ。

送信側の IP サブモデルは上位層モデルからの

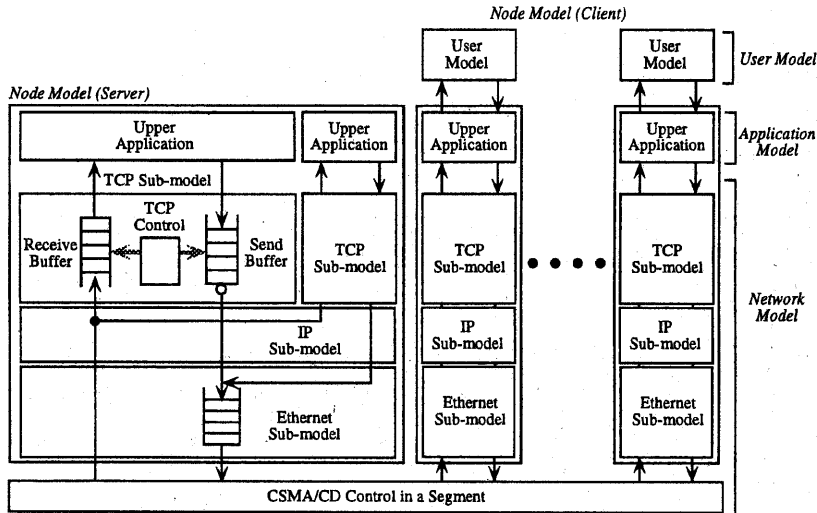


図 1: シミュレーションモデル

送信要求を受けると、対応する Ethernet サブモデルの送信待ち行列に宛先の IP サブモデルを指定して送信メッセージを加える。この時、必要であればデータグラムのフラグメント化が行われる。受信側の IP サブモデルでは、Ethernet サブモデルが発生する到着イベントを受けとり、必要であればデータグラムをリアセブンブルして上位層モデルに渡す。

2.1.3 TCP および UDP サブモデル

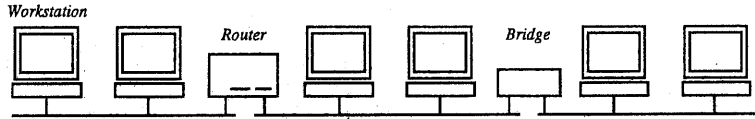
TCP はトランスポート層のプロトコルとして、全二重の論理通信路の管理、パケット順番制御、確認応答付き再送信、フロー制御、輻輳制御を行う。モデルでは簡単のため論理通信路はあらかじめ確立されていると仮定し、TCP ウィンドウフロー制御による動作のみをシミュレートする。TCP の仕様書 [4] ではウィンドウフロー制御の具体的な実装方法は述べられていないが、その実装方法によって TCP の性能が大きく左右されることは広く知られている。本モデルでは 4.3 BSD UNIX [5] における実装方法を採用した。

UDP はアプリケーションが IP を直接利用するためのトランスポート層プロトコルである。UDP には TCP の様な障害対策機能がないので、UDP モデルでは、単純に上位層からの送信データのデータ長に UDP ヘッダの長さを加えて IP モデルに渡す機能のみを持つ。

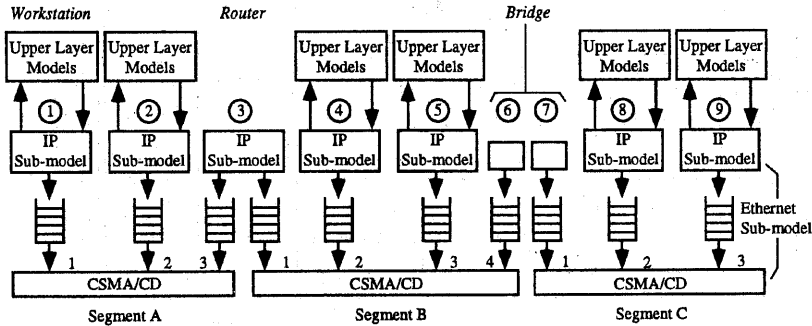
2.2 アプリケーションモデル

アプリケーションのモデルとして、クライアントからの要求によりサーバからのファイル転送を行う。2つのアプリケーション FTP と NFS のモデルが作成されている。FTP はトランスポート層プロトコルとして TCP を用いる。このため、パケット損失などの障害対策として TCP による確認応答付再送信が行われる。一方、NFS はトランスポート層プロトコルとして UDP を使い、上位層のプロトコル XDR および RPC 上で動作する。UDP は TCP と異なり障害対策を行わない。NFS における障害対策は NFS 自身による単純なタイムアウト & リトライ方式である。

各アプリケーションモデルの動作は単純化し、クライアント・サーバ間で交換されるメッセージは、ファイル転送要求とそれに対する応答メッセージのみとした。クライアント側のアプリケーションモデルは、ユーザモデルが発生するユーザの操作イベントに従って要求メッセージを生成する。ユーザの1回の操作に対して、複数の要求メッセージの送信が必要な場合、アプリケーションモデル内部で、要求メッセージを生成する。NFS クライアントにおける要求メッセージの再送や、一回の要求で処理が終らないファイル読み込み要求の送信などはこのように行われる。サーバ側のアプリケーションモ



(a) Multisegment Network



(b) Simulation Model of Multisegment Network

図 2: 多セグメントネットワークモデル

丸数字は MID を表す。

デルは、クライアント側の要求メッセージを受けとり、要求に応じた応答メッセージを生成してクライアント側に送信する。

なお、NFS の実装方式は SunOS 4.1 の実装方式に従っている。

3 多セグメントネットワークへの拡張

3.1 多セグメントネットワークモデル

前述のシミュレーションモデルをルータおよびブリッジを含む多セグメントネットワークへ対応できるように拡張する。この拡張は案であり、拡張モデルの実装は現在終了していない。

図 2 に多セグメントネットワークの例とそのシミュレーションモデルの概要を示す。多セグメントネットワークのモデルでは、ルータは 1 つの IP サブモデルが複数のセグメントの Ethernet サブモデルの入力待ち行列に入力を持つユニットとして表現される。これに伴い、多セグメントネットワークのモデルでは IP サブモデルに経路制御の機能を追加する。一方、ブリッジは、データリンク層で動作するネットワーク接続機器であるため、ブリッジのモ

表 2: データリンク層レベルの経路表

Ethernet Segment	Destination MID							
	1	2	3	4	5	8	9	
A	1	2	3	3	3	3	3	
B	3	3	3	4	5	7	7	
C	6	6	6	6	6	8	9	

(MID = 6, 7 はブリッジなので宛先の MID には含まれない。)

デルには IP サブモデルが含まれない。ブリッジはメッセージを単純に Ethernet サブモデルの入力待ち行列に加えるモデル (ブリッジサブモデル) 2 つで表される。各 IP サブモデルおよびブリッジサブモデルはそれぞれユニークな ID (以下 MID) を保持する。

多セグメントネットワークのシミュレーションにおいて問題となるのは、モデル上でのネットワークポロジの記述方式である。本モデルで、ネットワークポロジの記述に必要なのは、ルータを挟んだ異なるネットワーク間の経路情報とブリッジを挟んだ異なるセグメント間の経路情報である。前者はネットワーク層における経路情報に相当し、後者はデータリンク層における経路情報に相当する。

表 1: ネットワーク層レベルの経路表

括弧内は Ethernet サブモデルの入力待ち行列

Source MID	Destination MID						
	1	2	3	4	5	8	9
1	1 (-)	2 (A1)	3 (A1)	3 (A1)	3 (A1)	3 (A1)	3 (A1)
2	1 (A2)	2 (-)	3 (A2)	3 (A2)	3 (A2)	3 (A2)	3 (A2)
3	1 (A3)	2 (A3)	3 (-)	4 (B1)	5 (B1)	8 (B1)	9 (B1)
4	3 (B2)	3 (B2)	3 (B2)	4 (-)	5 (B2)	8 (B2)	9 (B2)
5	3 (B3)	3 (B3)	3 (B3)	4 (B3)	5 (-)	8 (B3)	9 (B3)
8	3 (C2)	3 (C2)	3 (C2)	4 (C2)	5 (C2)	8 (-)	9 (C2)
9	3 (C3)	3 (C3)	3 (C3)	4 (C3)	5 (C3)	8 (C3)	9 (-)

(MID = 6, 7 はブリッジなので宛先, 送信元の MID に含まれない.)

3.2 データリンク層レベルの経路情報

多セグメントネットワークのモデルでは, 異なるセグメント間でのメッセージ送信時に, 送信側セグメントの Ethernet サブモデルからどの上位モデルに到着イベントを発生させるかを指定する方法が必要である. 単一のセグメントモデルであれば, 送信時に指定された宛先の IP サブモデルに直接到着イベントを発生させればよい. しかし, 多セグメントネットワークのモデルでは, セグメント間の通信を行う時に, 宛先の IP サブモデルに対して直接到着イベントを発生させることができない場合がある. 図 2 のモデルで, 送信側のセグメント B の Ethernet サブモデルが宛先 MID = 9 のメッセージを処理する場合を考える. この場合, MID = 9 の IP サブモデルに直接到着イベントを発生させることはでない. 到着イベントが発生すべきモデルは MID = 7 のブリッジサブモデルである.

ワークステーションおよびルータのモデルの IP サブモデルで指定できる宛先は, IP サブモデルを持つ MID であり, ブリッジを指定することはできない. これはブリッジはデータリンク層で動作するネットワーク機器であり, ネットワーク間のルーティングを行うものではないからである.

そこで多セグメントモデルの Ethernet サブモデルでは, 入力待ち行列に加えられたメッセージの宛先 MID から, データリンク層レベルでのネットワークトポロジに従って, 次に到着イベントを発生させるブリッジサブモデルおよび IP サブモデルの MID を判定する. この判定のためにデータリンク層レベルでのネットワークトポロジを記述した経路表を用いる. この経路表はデータリンク層レベルでの経路情報を記述したものであり, 一般に IP ルー

タが用いるネットワーク層レベルでの経路表とは異なる. 図 2 のネットワークのデータリンク層レベルの経路表の例を表 2 に示す.

表 2 から, 前出のセグメント B の Ethernet サブモデルが宛先 MID = 9 のメッセージの処理を行う場合には, 到着イベントが発生するのは MID = 7 のブリッジサブモデルであることがわかる. MID = 7 のブリッジサブモデルに到着したメッセージは宛先 MID を変更せずにそのままセグメント C の Ethernet サブモデルの送信待ち行列に加えられる. 表 2 よりこの後到着イベントが発生するのが本来のメッセージの宛先である MID = 9 の IP サブモデルであることがわかる.

3.3 ネットワーク層レベルの経路情報

ネットワーク層レベルの経路情報は, 実際の IP と同様にネットワーク層レベルの経路表 (表 1) を定義することにより与えることができる. この経路表では上位層モデルが指定する宛先の IP サブモデルの MID から, 中継ノードの IP サブモデルの MID およびそれに対応したセグメントの Ethernet サブモデルの入力待ち行列を指定する. IP サブモデルはこの経路表を参照して上位層からの送信メッセージおよび Ethernet サブモデルから受けとった中継メッセージを処理する.

MID = 9 の IP サブモデルが上位層が指定した宛先 MID = 1 のメッセージを処理する場合, 表 1 から中継ノードが MID = 3 のルータであることがわかる. この中継ノードの MID がセグメント C Ethernet サブの入力待ち行列 3 に宛先として渡される. Ethernet サブモデルは表 2 を参照し, この宛先に対応した到着イベントを発生させるのが

MID = 6 のブリッジサブモデルであることを知る。ネットワーク層レベルの経路表はデータリンク層レベルの経路表とは独立である。このため、IP サブモデル内で動的に経路表を書き換えることにより、ネットワーク層における動的な経路制御のシミュレーションが可能である。

4 検討

Ethernet サブモデルが保持するデータリンク層レベルの経路表と IP サブモデルが保持するネットワーク層レベルの経路表、および Ethernet サブモデルが保持するセグメント内ノード間の遅延情報によって、多セグメントネットワークのトポロジを完全に記述することができる。しかし、数百台規模のネットワークを考えた場合、前節で述べた経路表を人間の手で作成するのは困難である。また、動的な経路制御が行われる場合には、経路表はあくまでモデル内の内部表現とし、モデル外部から与えるネットワークトポロジの表現はより洗練された形にするべきである。現在、ネットワークトポロジの効率的な外部表現方式として、S 式を用いた方式やビジュアルプログラミングによる方式を検討している。

本モデルでは、経路表をネットワーク層およびデータリンク層で異なるものを使うようにしているが、これら 2 つの経路表をまとめて経路制御を行うことも可能である。データリンク層の経路表をネットワーク層の経路表に組み込み、IP サブモデルがブリッジサブモデルの MID を直接宛先として指定できるようにすれば良い。この場合、ブリッジサブモデルおよび IP サブモデルは、経路表を参照して次に到着イベントを発生させるモデルの MID を判定して Ethernet サブモデルに渡すことになる。しかし、動的な経路制御をモデル内で行う場合には、IP サブモデルが宛先に対する次の中継ノードの変更を行った時には、その経路に含まれるブリッジサブモデルが次にメッセージを渡すべき MID も変更される可能性がある。ひとつの IP サブモデル内での動的な経路変更の影響がモデル全体の経路情報に及ぶため、モデルの実装は複雑になると考える。

5 まとめ

本稿では、教育用 LAN などの時間的集中度が高いネットワークの性能評価を行うためのシミュレ-

ションモデルとして、各階層のネットワークプロトコルのモデルを並列に動作させるイベント駆動方式のシミュレーションモデルを紹介し、本モデルの多セグメントネットワークへの拡張方法について述べた。本方法ではネットワーク層およびデータリンク層での経路表をモデルに与えることによって、多セグメントネットワークのトポロジを記述する。

多セグメントネットワークに拡張されたシミュレーションモデルの実装は現在終了していない。今後の課題は、シミュレーションモデルの実装とネットワークトポロジの効率的な外部記述方式の検討である。

参考文献

- [1] 岡田, 櫻井, 岩田, “教育用大規模分散型 WS システムの一構成方法,” 情処論, vol. 37, no. 12, pp. 2447-2456, 1996
- [2] 石田, 高原, “イーサネット上のトラフィック特性によるバケット発生モデルについて,” 信学論, vol. J78-B-I, no. 11, pp. 664-671, 1995
- [3] 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, vol. J-78-A, no. 8, pp. 961-964, 1995
- [4] J. Postel, “Transmission control protocol,” RFC 793, 1981
- [5] S. J. Leffler and Marshall Kirk McKusick et al., “The design and implementation of the 4.3BSD UNIX operating system,” Addison-Wesley, 1989
- [6] 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
- [7] 石原, 岡田, “高負荷 LAN におけるクライアント・サーバシステムの性能評価,” 信学技報, IN96-129, pp. 9-16, 1997