

負荷集中 LAN における クライアントサーバシステムの評価

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

〒464-01 名古屋市千種区不老町

Tel 052-789-3906

susumu@ecip.nagoya-u.ac.jp, okada@ecip.nagoya-u.ac.jp

あらまし ネットワークへの過渡的な負荷集中が起こるクライアントサーバシステムの一つとして, 集合情報処理教育施設におけるシステムをとりあげ, 実際のユーザ利用時におけるネットワーク特性をシミュレーションにより評価した. シミュレーションモデルは各層のプロトコル動作を詳細に記述したモデルをイベント駆動で動作させるモデルであり, プロトコルには Ethernet および TCP/IP 上で動作する NFS と FTP を仮定した. 両者間での負荷集中時の Ethernet の状態変化を本論文で定義するストール率によって比較し, Ethernet コリジョンドメイン内におけるクライアント数の実用限界とシステム破綻限界を示した.

キーワード クライアント・サーバシステム, Ethernet, NFS, TCP/IP, シミュレーション, ネットワーク設計

Performance Evaluation of a Client-Server System on High-Loaded LAN

Susumu ISHIHARA[†] and Minoru OKADA^{††}

[†]Graduate school of Engineering, Nagoya University,

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

Nagoya University, Nagoya, 464-01 JAPAN

Tel +81-52-789-3906

susumu@ecip.nagoya-u.ac.jp, okada@ecip.nagoya-u.ac.jp

Abstract In this paper we evaluated a performance of a client-server system on a hi-loaded LAN with behaviors of user applications. We selected an educational client-server system as a sample of such systems. This simulation was performed by an event-driven simulation model that have detailed models of multi-layer protocols (Ethernet, TCP/IP, NFS and FTP). We evaluated the Ethernet states under the use of NFS and FTP by average stall rate which defined in this paper. From this result, the maximum numbers of client workstations for practical use and extreme use in an Ethernet collision domain are shown.

key words client-server system, Ethernet, NFS, TCP/IP, simulation, network design

1 はじめに

最近のネットワークの普及により大規模な分散システムを導入する施設が増えている。大学の情報処理教育センター等の集合情報処理教育施設においても例外ではなく、数百台のワークステーション (WS) と数千人のユーザを含むシステムの例 [1] がある。このようなシステムでは、全てのユーザに同一の利用環境を提供するために、一台ないしは複数台のファイルサーバで全ユーザの個人データを管理している。しかし、教育用のシステムでは授業利用時にネットワークおよびファイルサーバに負荷が集中するため、この負荷を正確に予測し適切な応答時間を提供することがシステム設計上の問題点となる。一方、実務用のシステムでは、このような極端な負荷集中が頻繁に起こることはほとんどない。

数百台規模のシステムのユーザ利用時のピーク負荷を予測し、実システムの評価を行うことは困難である。このためこれらのシステムでは、解析的手法やシミュレーションによる評価を行うのが現実的である。従来、クライアントサーバシステムにおける性能評価手法として多くの解析的評価手法 [2][3][4] や評価ツール [5] が提案されてきた。しかし、これらの多くは各階層の通信プロトコルについてのみ述べたものが多く、実際のユーザ利用時のトータルなシステム特性を評価したものは少ない。また従来の研究ではデータリンク層以下のネットワーク性能については、平均的なネットワーク特性に基づいてプロトコルをモデル化しており、高負荷時におけるネットワーク特性の変化は考慮されていない。アプリケーションによる Ethernet 上のパケット発生率の違いを考慮したシミュレーションが [6] により行われているが、実務環境における平均的なネットワーク負荷を対象に行ったものであり、教育環境のようなきわめて負荷集中度が高いシステムを対象にした例はない。

そこで、筆者らは実システムにおける負荷集中時のネットワーク特性を正確に評価するため、各階層のプロトコルおよびユーザの動作を忠実にモデル化する一方法を提案してきた [7]。本研究ではこのモデルによるシミュレーションにより、教育環境におけるネットワーク負荷集中時のネットワーク特性評価を複数のプロトコルを対象に行った。

本論文ではまず、評価に用いたシミュレーションモデルの概要について述べる。次に教育環境における負荷集中を表すモデルについて述べる。最後に異なる障害対策をもつプロトコルを用いた二つのアプリケーション NFS と FTP における負荷集中時のファイル

転送についてシミュレーションした結果を示し、両者の特性の違いを検討する。

2 シミュレーションモデルの概要

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。

各ノードのモデルは、1) ネットワークモデル、2) アプリケーションモデル、3) ユーザモデルの 3 つのモデルから構成される。ネットワークモデルは OSI 第 4 層以下のプロトコルの動作を記述したモデルである。この上に任意のアプリケーションモデルを加えることにより、様々なアプリケーション利用時のネットワーク特性の評価が可能である。ユーザモデルは、クライアント WS のユーザがネットワークアクセスを伴う操作を行うタイミングを記述するモデルである。

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。

3 ネットワークモデル

ネットワークモデルは、通信に使用する各階層のプロトコルに忠実に動作し、送受信間で交換される個々のパケットの動きを完全にシミュレートする。このため、パケットの損失やラウンドトリップ時間の変化によって引き起こされる再送信などのプロトコル動作がユーザへの応答時間に及ぼす影響を正確に予測できる。なお本モデルではネットワークプロトコルの動作のみをモデル化しており、それによって消費される CPU 処理時間の影響は考慮していない。

使用プロトコルには、TCP/IP プロトコルと 10Mbps および 100 Mbps の Ethernet [8] を仮定した。汎用化のため、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。

3.1 Ethernet サブモデル

Ethernet サブモデルは CSMA/CD の動作を状態遷移モデル (図 2) でシミュレートする。本モデルの

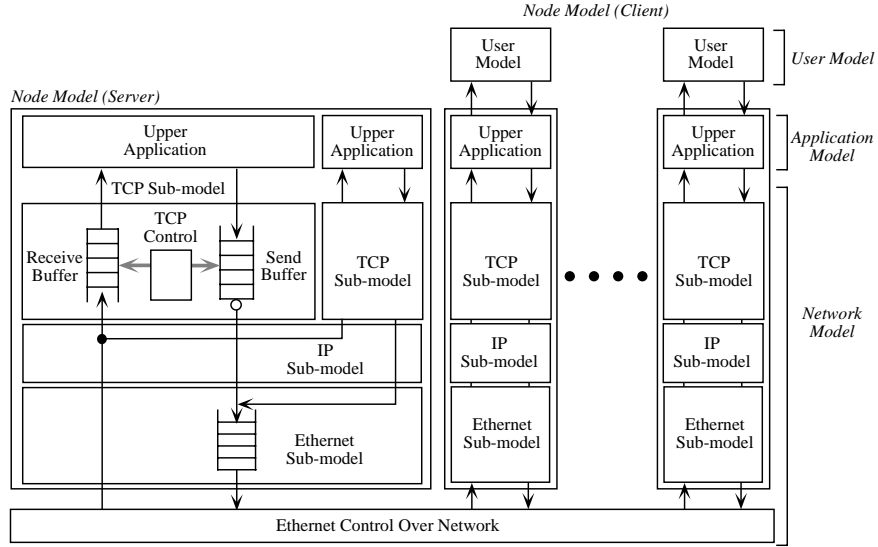


図 1: シミュレーションモデルの構成

詳細については [7] に譲り，ここではその概要について述べる．

ノード i の状態は 6 つの状態 $S^{(i)} \in \{\text{SLP}, \text{CS}, \text{CR}, \text{TR}, \text{CD}, \text{WT}\}$ によって表される．以下に個々の状態の持つ意味について示す．

SLP 送信待ち行列が空であり，何も送信していない．

CS キャリアセンス中．

CR フレームの送信中．衝突が起きる可能性がある．

TR フレームの送信中．衝突は生じていない．

CD 衝突を検出し，JAM 信号を送信している．

WT JAM 信号を送信後，次の送信再試行まで待機している．

CSMA/CD においては，送信元における伝送路上のキャリアの検出と衝突の検出がその動作の要となる．このため，各ノードのモデル i は最後にキャリアを発生した時刻 $t_s^{(i)}$ ，キャリアを消した時刻 $t_e^{(i)}$ ，および送信中のフレームに関する衝突回数 $n_c^{(i)}$ を保持している．これらのパラメータおよび $S^{(i)}$ を全ノード間で比較することによって，各ノードにおけるキャリアの検出時刻，および衝突の検出時刻を計算できる．

なお，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．

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

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の状態を衝突する可能性があるノード ($S^{(i)} = \text{CR}$) と衝突の可能性がないノード ($S^{(i)} \in \{\text{TR}, \text{CD}\}$) に明確に分けることができる．このため，衝突判定は $S^{(i)} = \text{CR}$ となるノードの数を判定することのみによって行うことができる．送信中のノードの数を数えて衝突の判定を行うシミュレーション手法は他にも存在するが [9]，これらの手法では上位層の動作を考慮に入れておらず，各ノードがフレームの送信を開始する時刻があらかじめ分かっていることを前提としている．本研究では上位層の動作は Ethernet サブモデルでは予測不可能であることを前提としているため，これらの手法を直接利用することはできない．

3.2 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．今回のシミュレーションでは単一のネットワークを考慮しているので，経路選択の機能はモデルには必要ない．データグラムのフラグメント化の機能も単一の LAN では必要のないことが多いが，今回はアプリケーションとして想定した NFS が同一 LAN 内でも積極的にフラグメント化を利用するため，モデルにフラグメント化の機能を組み込んだ．

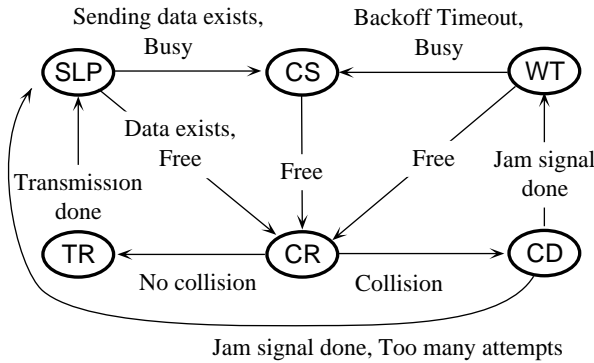


図 2: Ethernet サブモデルの状態遷移図

フラグメント化およびリアセンブルのための IP サブモデルの動作の流れは以下になる。

1. 送信側の IP サブモデルは、上位層のモデルからデータグラムを受け取ったときにフラグメント化の必要があると、データグラムをフラグメント化し、Ethernet サブモデルの送信待ち行列に加える。
2. 受信側の IP サブモデルは、フラグメントを受信するとそのフラグメントが属するデータグラムを表す表のエントリにそれを加える。フラグメントの到着によってデータグラムが完成されると、上位層サブモデルに到着イベントを発生させる。

3.3 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し仮定し、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した。

TCP の仕様書 [10] 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 [11] における実装方法を採用した。TCP サブモデルの具体的な実装方法とパラメータについて表 1 に示す。

UDP はアプリケーションの送信データグラムをそのまま IP に渡すプロトコルである。モデルでは、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。

表 1: TCP サブモデルの実装パラメータ

送信バッファサイズ	4096 octet
受信バッファサイズ	32768 octet
最大セグメントサイズ	1460 octet
再送タイムアウト最大値	64 sec
再送タイムアウト最小値	1 sec
最大再送回数	無限大

4 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リクエストによりサーバからファイル転送を行う 2 つのアプリケーション、FTP と NFS を仮定した。FTP はトランスポート層のプロトコルに TCP を使って通信を行う。このため、パケット損失などの障害対策は全て TCP によって行われる。すなわち FTP を使った場合、障害対策としてウィンドウフロー制御および確認応答付き再送信が行われることになる。一方、NFS はプレゼンテーション層プロトコル XDR とセッション層プロトコル RPC 上で動作し、トランスポート層には UDP を使用する。UDP は TCP とは異なり、障害対策を行わず、障害対策は NFS により行われる。NFS による障害対策は単純なタイムアウト & リトライ方式である。なお、NFS モデルは XDR および RPC のプロトコル動作を含んでいる。

4.1 FTP モデル

実際の FTP によるファイル転送では、一つのファイル転送を始める前に、サーバ・クライアント間で小さなメッセージ交換が数回行われる。しかし、今回のシミュレーションではアプリケーションレベルでのメッセージ交換は、クライアントからの一つのリクエストメッセージとそれに対する一つのサーバからの応答メッセージのみと仮定した。

以下に FTP モデルの動作の流れを示す。

1. ユーザモデルがコマンド送信イベントを発生すると、クライアントの FTP モデルは 20 byte (トランスポート層以下のヘッダ長を除く) のリクエストメッセージをサーバに送信する。
2. サーバの FTP モデルはクライアントからのリクエストメッセージを受け取ると、リクエストメッセージによって指定されたファイルサイズ分のデータをクライアントに送信する。

4.2 NFS モデル

FTP がファイル単位でデータの転送要求を行うのに対し、NFS では一つの要求メッセージで最大 8 KB のデータ転送を要求する。サーバ側は要求されたサイズのデータを一つのメッセージでクライアントに返送する。サーバ側の NFS モデルが送信する応答メッセージが Ethernet の 1 フレームに納まらない場合、応答メッセージは IP モデルでフラグメント化される。以下に NFS モデルの動作の流れを示す。

1. ユーザモデルがコマンド送信イベントを発生すると、クライアントの NFS モデルは 120 byte (トランスポート層以下のヘッダ長を除く) のリクエストメッセージにユニークな ID を付けてサーバに送信する。
2. サーバの NFS モデルはリクエストメッセージを受け取ると、要求されたサイズのデータを含む応答メッセージにリクエストメッセージと同じ ID を付加してクライアントに送信する。
3. クライアントは自分が最後に送信したリクエストと同じ ID の応答メッセージを受け取ると、新しくユニークな ID を持つリクエストメッセージを生成してサーバに送信する。2 ~ 3 の動作を受信データサイズがユーザが要求するファイルサイズに達するまで続ける。

もし、リクエストを送信してから、定められた再送タイムアウト時間内に応答メッセージを受け取ることができなければ、クライアントは同じ ID でメッセージをサーバに送信する。

なお、NFS の再送タイムアウトの設定方法には指数バックオフなどの手法があるが、今回は再送タイムアウトの時間は固定値とし、0.7 秒とした。また、最大再送回数は無限大であるとした。

5 ユーザモデル

ユーザモデルは、クライアント WS のユーザの挙動を記述したモデルである。教育用システム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ときである。このとき、学生が教官の指示に反応して操作を行うための時間にはばらつきが生じる。そこで本研究では、各クライアント WS がサーバ WS に対してリクエストを送信する時刻 t_s を

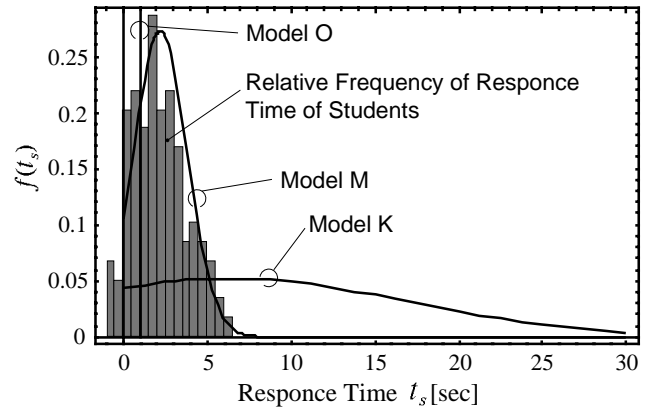


図 3: ユーザモデル

次式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。

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

式 2 は正規分布 $N(\mu, \sigma^2)$ の密度関数において、確率変数が非負になる確率を 0 とした確率分布である。

シミュレーションでは、ユーザの応答時間の集中がコマンドの入力方法によって異なると考え、パラメータ μ, σ の組合せが異なる 3 つのモデルを使用した。すなわち、コマンド入力の方法がマウスによる場合 (Model M)、キーボードによる場合 (Model K) およびクライアントが完全に同時にリクエストの送信を開始した場合 (Model O, $\mu = 1.0, \sigma = 0$) である。

パラメータ μ, σ の決定には、教育環境におけるユーザの応答時間の実測結果を用いた。[13] では学生 146 人を対象に、口頭による合図に対して GUI で応答するまでの応答時間を測定している。この結果では、学生の応答時間の 80% は 7.0 秒以内に集中しており、この範囲内の応答時間の平均値および標準偏差はそれぞれ 2.2 秒、1.6 秒であった。この値より、Model M では、 $\mu = 2.2\text{sec}, \sigma = 1.6\text{sec}$ とした。一方、CUI での応答時間の分布は [13] では測定されていない。そこで、キーボードでの入力はマウス入力に較べて時間を要し、その時間のばらつきも大きいことから、Model M に較べてより広がりのある分布として $\mu = 2.2\text{sec}, \sigma = 1.6\text{sec}$ という値を仮定した。この分布は 7 文字程度のコマンドライン入力を行うことを想定している。図 3 に、[13] による測定結果 (7.0 秒以下) と Model M, K, O における t_s の分布を示す。

6 シミュレーション結果

シミュレーションは以下の状況を仮定して行った。1台のファイルサーバと N 台のクライアント WS が 1 台の HUB に 10 m のケーブルを介して接続されている 10BASE-T (10 Mbps) および 100BASE-TX (100 Mbps) のネットワークを仮定する。この環境下で、5 で定義したアクセス集中モデルに従って全てのクライアント WS がサーバ WS に対して NFS および FTP によるファイル転送要求を行う。ファイルサイズは、中規模のアプリケーションをファイルサーバから読みとることを想定し、1 MB とした。なお、FTP で接続する場合、クライアントがリクエストを送信する前に TCP におけるサーバ・クライアント間の接続は確立されているとした。

6.1 ファイル転送時間

図 4 に NFS および FTP における全クライアントの平均ファイル転送時間を示す。

同図より、100BASE-TX を利用した場合、NFS, FTP とともに Model M, K で 1 MB のファイル転送時間は最大でも 5 秒以内であり、十分な実用域である。一方、10BASE-T の場合、NFS, FTP とともに Model M では $N = 5$, Model K では $N = 20$ でファイル転送時間が急激に増加している。このことから、10 Mbps の Ethernet では教育環境における Ethernet の 1 つのコリジョンドメイン内のノード数は、GUI, CUI のコマンド入力を行う場合、それぞれ 5 台程度、20 台程度が実用限界であるといえよう。また、100 Mbps の Ethernet を用いた場合、実用限界は少なくとも 100 台以上である。

また、10BASE-T の場合、Model M, O で NFS によるファイル転送時間が $N > 90$ で急激に増加していることが分かる。これはネットワーク入力負荷の増大によってネットワークが完全に破綻していることと捉えることができる。このときの N が 10 Mbps の Ethernet の破綻限界と言える。

6.2 ストール率による Ethernet の状態解析

6.2.1 ストール率の定義

混雑時の Ethernet の状態を説明するため、ここでストール率という概念を導入する。クライアントのノード i においてそのノードが Ethernet サブモデルで定義されるノードの状態 $S^{(i)} \in \{CS, CR, TR, CD, WT\}$ となる時間を $T_{XMT}^{(i)}$ とする。

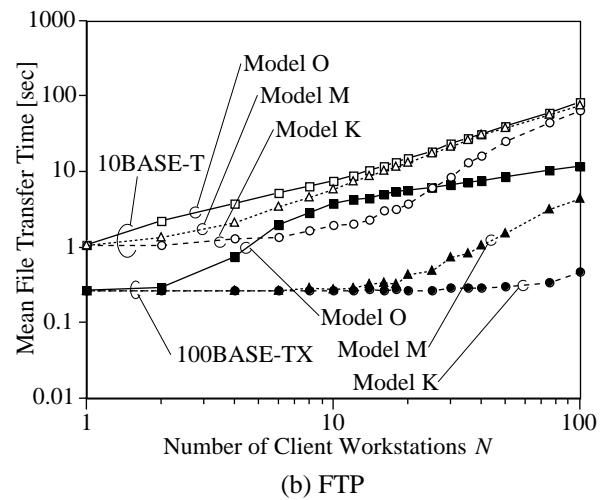
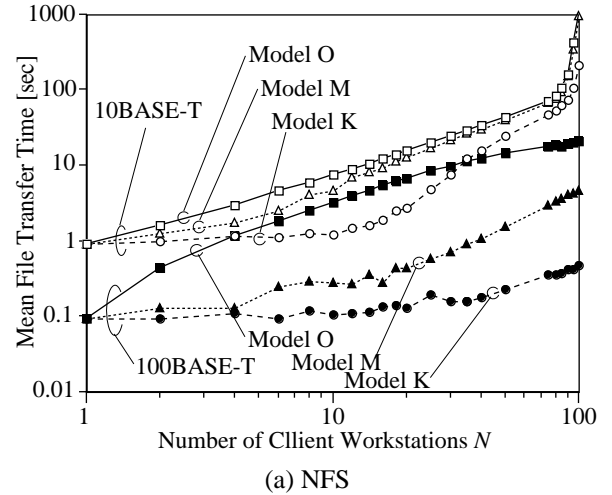


図 4: ファイル転送時間

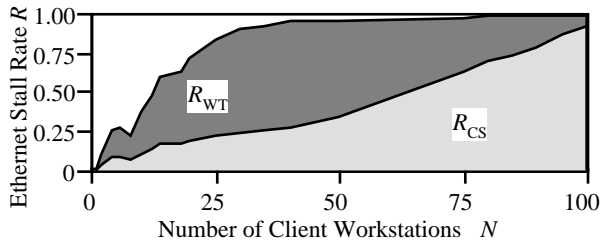
また、 $S^{(i)} = CS$ となる時間を $T_{CS}^{(i)}$, $S^{(i)} = WT$ となる時間を $T_{WT}^{(i)}$ とする。これよりクライアント台数が N の時の平均ストール率 R を以下のように定義する。

$$R_{CS} = \frac{1}{N} \sum_{i=1}^N \frac{T_{CS}^{(i)}}{T_{XMT}^{(i)}} \quad (3)$$

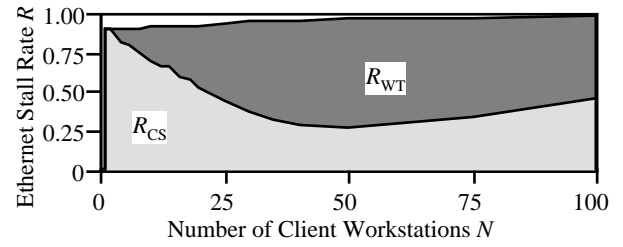
$$R_{WT} = \frac{1}{N} \sum_{i=1}^N \frac{T_{WT}^{(i)}}{T_{XMT}^{(i)}} \quad (4)$$

$$R = R_{CS} + R_{WT} \quad (5)$$

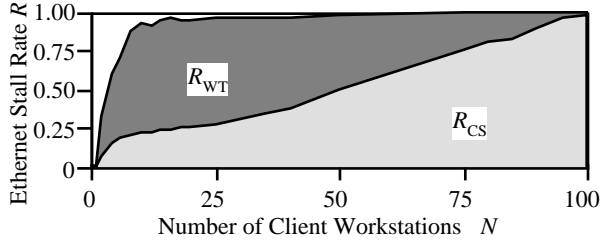
ストール率 R は、1つのフレーム送信時間に対して、実際には信号を送信していない時間が占める割合である。また、 R_{CS} および R_{WT} はそれぞれ、信号を送信していない時間のうちキャリアセンスをしている時間と送信再試行待ち時間が占める割合を示す。



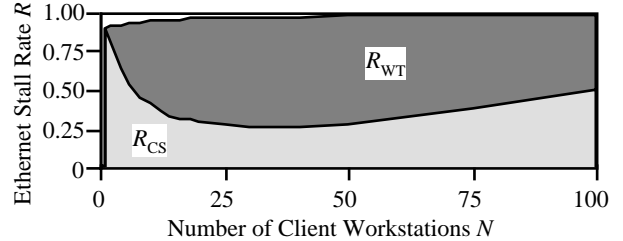
(a) NFS, Model K



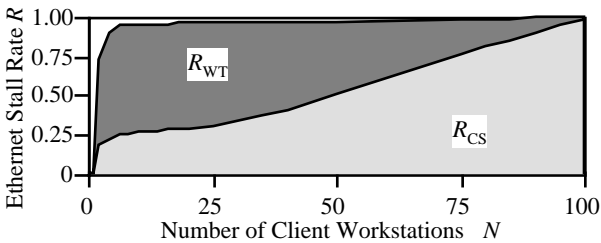
(d) FTP, Model K



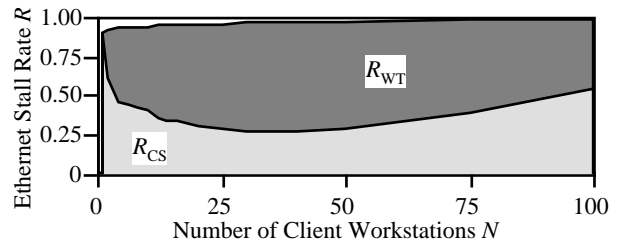
(b) NFS, Model M



(e) FTP, Model M



(c) NFS, Model O



(f) FTP, Model O

図 5: クライアントの平均 Ethernet ストール率 (10BASE-T)

6.2.2 Ethernet の状態解析

図 5に、10BASE-T 使用時における NFS と FTP でのクライアント数 N に対する平均ストール率 R の変化を示す。100BASE-TX の場合の結果は、モデル O を除き 10BASE-T の場合の結果の $0 < N < 10$ の部分を $0 < N < 100$ に伸張したものと同様の傾向を示した。

NFS ノード数が多くなるにつれて R が増加している。 $R = 0.9$ に達した時の N はファイル転送時間が急激に増加する最初の N に一致している。また、 N の増加に伴い、 R_{CS} が増加している。特にモデル M と O では $N = 100$ は $R_{CS} > 0.9$ であり、そのときのファイル転送時間は急激な増加傾向を見せている。このことから、NFS の場合 R および R_{CS} が 0.9 になるときの N がそれぞれネットワークの実用限界と破綻限界を示していると言えよう。

FTP N が小さい場合にも R は大きな値を保ち、 N の増加に対しても大きな変化は見られない。NFS の場合と異なり、 N が小さいときに R_{CS} の値が大きい。

これは受信側の TCP が、受信すべき全てのデータサイズに関係なく、連続したデータを受け取った場合にすぐに確認応答を送信するためである。しかし、実際にクライアント側から確認応答が Ethernet で送信されるのはサーバの送信終了後であり、 N が小さいときには他のクライアントが送信する確認応答と競合することがないため、衝突は起きにくい。 N が増加すると、サーバの送信終了後に複数のクライアントが同時に確認応答を送信することになり、 R_{WT} が増加する。

NFS, FTP どちらの場合でも、 N が大きいときには R に対して R_{CS} が占める割合が高い。この状態は、サーバからの連続した全てのクライアントに対する応答メッセージ送信、および再送タイムアウトによる再送信によって伝送路が占有され、クライアントは伝送路の空きをみて新たにメッセージを送信することができなくなっている状態を示していると考えられる。各ノードが送信を開始することができないため、必然的に衝突が起きにくくなり、 R_{CS} に対して R_{WT} は減少する。この傾向は NFS の場合に顕著である。

NFS の場合、障害対策のために送信要求メッセージの再送信を行うのはクライアントであり、再送タイ

ムアウトの値も固定値である。このため、 N が多くなるにつれて再送リクエストによるパケットの増加がネットワークの混雑を増長していると考えられる。一方 TCP の場合、再送を行うのはサーバである。受信側であるクライアントの TCP は送信側からデータを受信しない限り確認応答を送信しないので、クライアントの増加によるパケットの発生は NFS の場合ほど多くはならない。また、TCP では送信側の再送待ち時間を適応制御しているため、NFS の場合のように頻繁に再送が行われることがない。

7 むすび

極度の時間的負荷集中が発生するクライアント・サーバシステムの例として、集合情報処理教育施設のシステムをとりあげ、実際のアプリケーション利用時を想定したシミュレーションを行った。新たに定義したストール率により、教育環境における Ethernet の一つのコリジョンドメイン内のクライアント数の実用限界と破綻限界を示した。また、ネットワーク混雑時の Ethernet の状態について、FTP と NFS 利用時における両者の比較を行った。この結果、NFS を利用した場合の方がコリジョンドメイン内のクライアント数の増加によるネットワーク混雑への影響が高いことを示した。

今回行ったシミュレーションでは、対象とするネットワークを Ethernet の一つのコリジョンドメインに絞って行ったが、本シミュレーションモデルにネットワーク接続機器のモデルを加えることにより、複雑なトポロジのネットワークの評価も可能である [14]。また、今回のシミュレーションでは CPU の処理時間が通信時間に及ぼす効果についてはモデルに組み込まれていない。スイッチングハブや高速 LAN を用いた場合には、サーバ負荷による影響が無視できないと考える。今後は、CPU 負荷をモデルに組み込み、スイッチングハブ、ルータ等を含む複雑なネットワークの評価を行う予定である。

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] O. C. Ibe, H. Choi and K. S. Trivedi, “Performance evaluation of client-server systems,”

- IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 11, pp. 1217–1229, 1993
- [3] M. Murata and H. Takagi, “Two-layer modeling for local area networks,” *IEEE Trans. Commun*, vol. 36, no. 9, pp. 1022–1034, 1988
- [4] C. M. Woodside, J. E. Neilson *et al.*, “The stochastic rendezvous network model for performance of synchronous client-server-like distributed software,” *IEEE Trans. Comp*, vol. 44, no. 1, 1995
- [5] G. Franks, A. Hubbard *et al.*, “A toolset for performance engineering and software design of client-server systems,” *Performance evaluation*, vol. 24, pp. 117–136, 1995
- [6] 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, vol. J78-B-I, no. 11, pp. 664–671, 1995
- [7] 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
- [8] ANSI/IEEE Standard 802.3, “Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications”, *IEEE Press*, 1993
- [9] P. Marino and A. del Rio, “An accurate and fast CSMA/CD simulator,” *Microprocessing and Microprogramming*, vol. 39, pp. 187–190, 1993
- [10] J. Postel, “Transmission control protocol,” *RFC 793*, 1981
- [11] S. J. Leffler and Marshall Kirk McKusick *et al.*, “The design and implementation of the 4.3BSD UNIX operating system,” *Addison-Wesley*, 1989
- [12] Sun Microsystems, Inc, “NFS: Network file system protocol specification,” *RFC 1094*, 1989
- [13] 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
- [14] 石原, 岡田, “高負荷多セグメントネットワークの性能評価モデル,” 信学全大 97, 1997 (submitted)