

メタモニタ：適応型ネットワークトラフィック観測機構

八木 哲[†] 小倉 毅[†] 川野 哲生[†]
丸山 充[†] 高橋 直久[†]

本稿では、Gb/s クラスが実用水準に達してきた高速ネットワークの多様な運用管理のために、メタモニタと呼ぶ、比較的低い処理能力とシンプルな構成で、Gb/s クラスの高トラフィックを柔軟に観測する機構について述べる。メタモニタの特徴は次のとおり。1) 複数の異なる性質のモニタで協動的に観測する。2) 観測負荷に応じて全数調査と標本調査を選択する。3) 観測要求や現在の観測結果などの状況に応じて各モニタの観測焦点を変更する。本稿では、まず、メタモニタをモデル化する。次に、観測系とフィードバック系からなるメタモニタについて、観測系の実現方法を述べ、我々が開発を進めている並列分散型高速通信スイッチ“COREswitch”の上に構築したプロトタイプを用いた実験から、基本特性を考察する。さらに、ネットワーク・シミュレータを用いた実験から、実ネットワークへの適応性を考察する。

METAMONITOR: An Adaptive Network-traffic Monitor

SATORU YAGI,[†] TSUYOSHI OGURA,[†] TETSUO KAWANO,[†]
MITSURU MARUYAMA[†] and NAOHISA TAKAHASHI[†]

A simple economical method called METAMONITOR has been developed to monitor Gb/s network-traffic. The METAMONITOR can cooperate with any kind of monitor, select a sampling survey or a census survey according to a load of monitoring and change the focus of cooperating monitors dynamically according to a result of monitoring. The METAMONITOR has an observation system and a feedback system. This paper presents the METAMONITOR model, describes its installation of a observation system, the estimation of the performance of a installed on a parallel distributed high-speed communication switch “COREswitch”, and also the estimation of the adaptability to a real network using a network simulator.

1. はじめに

ネットワークの詳細なトラフィック観測は、障害検出(短期)、経路の制御や設計(中期)、設備設計(長期)など、多様なネットワーク管理のために有意である。これらの用途には、複数のノードの複数のポートを関連付けた観測を、リアルタイムで行う、柔軟な観測機構が望ましい。既存のネットワークでは、SNMPを基に、ネットワークノードをリアルタイムで観測する方法が利用できる¹⁾。遠隔ネットワークが対象の場合は、観測行為に起因するネットワークへの負荷を抑えるため、観測結果をあらかじめ統計処理する方法も利用できる^{2),3)}。また、より詳細な観測のために、パケットヘッダを蓄積し、バッチ的に観測するアプローチ^{4),5)}がある。

一方、ネットワークの高速化は、Gb/s クラスが実用水準に達し、レイヤ 3 スイッチを例として、転送処理のハードウェア化を促している。このような、転送処理のハードウェア化を必要とする、“高”トラフィックを対象にした柔軟な観測機構の実現には、処理能力不足や、高い処理能力を得るためのシステムの複雑化の問題がある。単純に考えれば、ルート検索用のハードウェアを基に、宛先アドレスに加え、発信アドレス、プロトコル種別、ポート番号などの組合せを検索キーとして選択可能にすれば、柔軟な観測が可能である。しかし、ルーティング機構以上に複雑で高コストな機構になり、現実的ではない。たとえば、OC3MON^{6),7)}では、汎用の PC と NIC を基盤とし、光スプリッタを介して OC-3 速度の回線を観測する、シンプルで低コストなシステムを実現している。しかし、Gb/s クラスを対象とすれば、汎用品を基盤とした構成では、バス速度や CPU パワーの制約から、複数ポートを監視することは困難である。また、NMVCsystem では⁸⁾、

[†] NTT 未来ねっと研究所
NTT Network Innovation Laboratories

独自の Gb/s クラスの回線インタフェース・チップ⁹⁾を基に、チップの観測用ポートにマルチプロセッサ構成の処理装置を組み合わせたプローブを、ネットワーク上に複数配置することを想定している。多数の観測点が必要な場合には、処理能力の高い複雑なプローブのために、システムは高コストになる。

これに対し、1) 負荷分散：複数の異なる性質のモニタで協調的に観測する、2) 負荷軽減：統計的な調査の考えに従い観測負荷に応じて全数調査と標本調査を選択する、3) 処理能力の最適配置：観測要求や現在の観測結果などの状況に応じて各モニタの観測焦点を変更することで、観測データ量の削減と観測能力の効率的利用を可能にし、比較的低い処理能力とシンプルな構成で、“高”トラフィックを柔軟に観測する手法を提案する。このような観測機構をメタモニタと呼ぶことにする。

本稿では、まず、メタモニタの枠組みを示すとともに、モデル化する(2章)。メタモニタは、統計的手法を用いるモニタの集合体からなる観測系と、その観測結果を基に、適応的に観測系の観測焦点を変更するフィードバック系に大別でき、フィードバック系のとる戦略は、観測系の観測特性に依存する。そこで、次に観測系の実現法を検討する。実現上の要件は、a) 統計的調査を可能にする標本抽出機構と、b) 観測焦点の自由な変更を可能にする、ネットワークノードとのインタフェースである。a) について、統計的な考えをそのまま具現化する“直接的な実現方法”に対して、実現性を重視した、“簡易的な実現方法”を提案する(3.1節)。b) について、観測系を組み込むネットワークノードの実装に依存するため、まずインタフェースの要件をまとめる(3.2節)。さらに実現例として、COREswitch¹⁰⁾上に構築した実験用プロトタイプを示す(4章)。このプロトタイプで簡単なトラフィックパターンを観測し、基本特性を考察する(5章)。また、ネットワークシミュレータを用いて、より実際のなトラフィックパターンを観測し、実ネットワークへの適応性を考察する(6章)。最後に、本稿の内容をまとめ、今後の課題を示す(7章)。

2. メタモニタ

メタモニタの枠組みを、図1に示す。この枠組みに従って、基本方針である(1) 負荷分散、(2) 負荷軽減、(3) 処理能力の最適配置の、具体化の方法を示す。

(1) 負荷分散：観測対象に選んだトラフィックの特徴を複数のパラメータに分解し、性質の異なる複数の要素モニタと呼ぶモニタで観測する。要素モニタ・

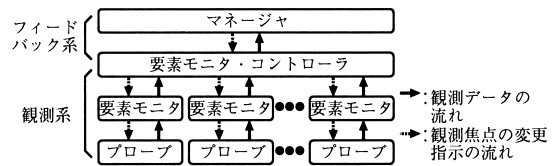


図1 メタモニタの枠組み

Fig. 1 The framework of METAMONITOR.

コントローラと呼ぶ、要素モニタの制御部で、各観測結果からトラフィックの特徴を求める。

- (2) 負荷軽減：要素モニタは、統計的な調査の考えに基づき、観測処理が軽いパラメータの観測には全数調査を適用し、観測処理が重いパラメータの観測には標本調査を適用する。
- (3) 処理能力の最適配置：a) 注目して観測したいポイント、b) 現在の観測結果、c) 観測に必要な負荷、d) 要素モニタの処理能力に基づいて、マネージャは観測焦点の変更内容を決定し、要素モニタ・コントローラを通じて、要素モニタに指示を出す。たとえば、複数ポートの概観から、特定ポートの詳細へ、観測焦点を絞る。

このように、マネージャから指示された、観測対象トラフィック(母集団)と、アドレスやプロトコルなどのパラメータ(母集団変数)に対して、要素モニタは、全数調査と標本調査を選択的に用いて調査を行う。要素モニタ・コントローラは、個々の要素モニタの調査結果から、観測対象トラフィックの様子を推定し、さらに複数の調査結果を組み合わせ、より複雑な条件での、観測対象トラフィックの様子を導出する。これを単位時間ごとに行い、時間変化を観測する。さらにマネージャは、観測結果に応じて、観測対象トラフィックや、パラメータの変更を指示し、観測焦点を変更する。要素モニタと要素モニタ・コントローラが持つ機能を定義し、メタモニタをモデル化する。

[定義1] 観測機能(要素モニタの機能)

ポートの集合 P からの入力パケットを確率 s_p ($0 < s_p \leq 1$, $0 < s_p < 1$ のときは標本調査, $s_p = 1$ のときは全数調査)で抽出し、抽出したパケットのうち、条件 c を満たすパケットの数を期間 s_t ごとに集計する。 P を観測対象ポート、 c をストリーム識別条件、 s_t を抽出期間、 s_p を抽出確率、集計した値の列 R_e を観測結果と呼び、観測機能 m_e を $R_e = m_e(P, c, s_t, s_p)$ で表す。□

観測機構は、指定ポート P からの入力パケットを確率 s_p でサンプリングし、標本を作成する。次に、たとえば行き先 IP アドレスを条件 c として、標本パケットを分類し、これを期間 s_t ごとに集計する。こ

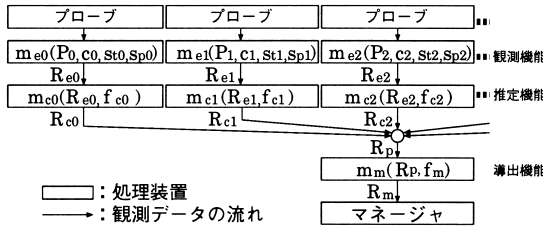


図2 メタモニタのモデル
Fig. 2 The model of METAMONITOR.

のとき、観測結果 R_e の分解能は、期間 s_t と同じである。

[定義2] 推定機能(要素モニタ・コントローラの機能) 観測結果 R_e の要素であるパケット数に対し、推定関数と呼ぶ関数 f_c を適用して得られたパケット数の列 R_c を、推定した観測結果と呼び、推定機能 m_c を、 $R_c = m_c(R_e, f_c)$ で表す。

推定機構は、たとえば、観測機構の観測結果 R_e が、母集団の $1/n$ の大きさの標本から求められたのであれば、最も単純な方法では、 f_c は単に R_e を n 倍する。

[定義3] 導出機能(要素モニタ・コントローラの機能) n 個の推定した観測結果 $R_{ci} (i = 0, 1, 2, \dots, n-1)$ を要素とする集合 R_p に対し、 R_{ci} の要素であるパケット数に導出関数と呼ぶ関数 f_m を適用して得たパケット数の列 R_m を、導出した観測結果 R_m と呼び、導出機能 m_m を $R_m = m_m(R_p, f_m)$ で表す。

導出機能は、複数の推定機構の調査結果 R_p に、たとえば集合演算を推定関数 f_m として適用し、複数条件に合致するパケット数を求める。

[定義4] メタモニタ

n 個の要素モニタを持つメタモニタは、個々の観測機能を $m_{ei} (i = 0, 1, 2, \dots, n-1)$ 、推定機能を $m_{ci} (i = 0, 1, 2, \dots, n-1)$ 、導出機能を m_m とすれば、以下のように表す。

- $R_{ei} = m_{ei}(P_i, c_i, s_{ti}, s_{pi}) (i = 0, 1, 2, \dots, n-1)$
- $R_{ci} = m_{ci}(R_{ei}, f_{ci}) (i = 0, 1, 2, \dots, n-1)$
- $R_m = m_m(R_p, f_m), R_p \ni R_{ci} (i = 0, 1, 2, \dots, n-1)$

モデルを図2にまとめる。マネージャは、定義に用いた各パラメータの値を決定し、観測焦点を変更する。

3. 実現法

3.1 標本抽出機構

標本抽出の“直接的な実現方法”は、たとえば、データ転送量の削減のために、プローブ部で分布の良い乱数¹²⁾を用いて、確率的にサンプリングすればよい(観

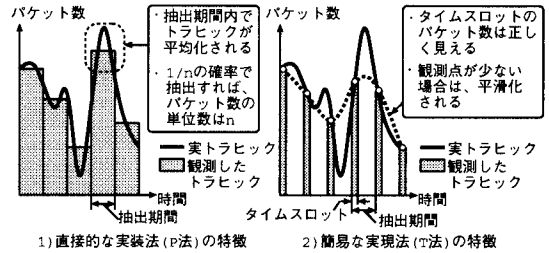


図3 標本抽出の実現方法の比較
Fig. 3 A comparison of sampling method.

測機能の標本抽出部をプローブに移設した形態)。しかし、1) プローブに高速の乱数生成機能が必要、2) 複数ポートを観測する場合、各プローブと観測機能の間の経路を共有すれば、パケットが統計多重されるとはいえ、ブロッキングの可能性がある、経路を独立させれば、複雑な機構になるなどの問題がある。これに対し、実現性の高い“簡易的な実現方法”として、抽出期間を抽出確率に応じた複数のタイムスロットに区切り、1つのタイムスロットを1つの観測対象ポートに割り当て、サンプリングする方法が考えられる。しかしこの方法では、割り当てられたタイムスロットの間に、トラヒック変動の山や谷があった場合には、これを観測できず、平滑化された結果が出る(図3)。

“直接的な実現方法”を確率型標本抽出法(P法)、“簡易的な実現方法”をタイムスロット型標本抽出法(T法)と呼ぶことにする。以降では、実現性は高いが、そのままでは平滑化の問題があるT法を、実用的に使用するための方法を考察する。

- 観測機能の水準：タイムスロットを単位時間として、観測対象ポートからの全入力パケットを計数する。抽出期間ごとに単位時間と累積入力パケット数の相関係数を求めれば、軽い負荷で入力パケット数のパースト変化(平滑化の発生)が推測できる。このとき、コンテキストスイッチの増加など、性能的問題が許せば抽出期間を短くして対処できる。ただし、入力パケットが複数のストリームで構成され、各ストリームの増減が相殺し、結果として累積入力パケット数が一様に増加する場合は、推測できない。
- 推定機能の水準：タイムスロットを単位時間とし、 x 軸に単位時間、 y 軸に割り当てられたタイムスロットで観測したパケット数をとる座標を求める。この座標の補完曲線を求め、抽出期間で積分することで、抽出期間のパケット数を近似する方法がある。多くの座標から補完曲線を求めると、リアルタイム性と処理量に問題が生じる可能性がある。
- 導出機能の水準：ある要素モニタで、抽出期間の

全入力パケットを計数する．この内訳として，他の要素モニタが求めた，割り当てられたタイムスロットの間の，入力パケットの内訳を適用する．分類された入力パケット数の合計が保証できる．

3.2 ネットワークノードとのインタフェース

観測焦点を自由に変更するためのプローブの要件 (1), (2) と，正しく観測するためのプローブと観測機構間の転送経路の要件 (3), (4) を，以下にまとめる．また，これらの機構はシンプルであることも重要である．

- (1) 任意のポートを等しく対象にできる．
- (2) プローブイフェクトが小さい．
- (3) パケットが破棄されない．
- (4) 通常の転送経路と同程度の遅延．

4. 実験用プロトタイプ

プラットフォームとなる COREswitch¹⁰⁾を 図 4 に示す．COREswitch は，155 Mbps と 622 Mbps の回線 I/F モジュール (CIF)，制御用のプロセッサモジュール (IFP)，クロスバ・スイッチ (XSW)，XSW のアービタ (ABT)，制御用バス (C-bus) からなる．入力パケットがあれば，CIF は内蔵する経路表を検索して宛先 CIF を特定し，XSW を介して転送する．ABT は CIF のリクエストに応じて XSW の調停を行う．IFP は C-bus を介して各 CIF の経路表を管理する．IFP と CIF は，XSW と C-bus に対してシメトリであり，柔軟に構成できる．

本プロトタイプのメタモニタは，IFP 上の 3 つの要素モニタと，WS (図 4) 上の要素モニタ・コントローラからなり，マネージャとして，コマンド入力インタフェースがある．指定ポートの入力パケットの中で，指定の IP アドレスを宛先に持つパケット数を求める．T 法の動作を基本とするが，P 法の動作も限定的に可能である．

- プローブ：XSW と ABT による One-shot 型マルチキャスト機能¹¹⁾を用い，IFP にパケットを取り込む．すなわち，観測対象となる CIF において，IFP にも同じパケットが届くように，宛先 CIF と IFP にマルチキャストする．設定は CIF の経路表

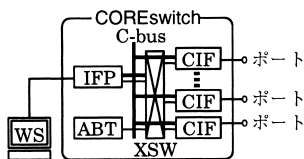


図 4 COREswitch

Fig. 4 The COREswitch.

のエントリにあるフラグの操作で行い，任意のポートを対象にできる．動作はユニキャストと比較して百数十 nsec 程度遅いだけであり，プローブイフェクトは小さい．また，同時に複数のポートを対象としなければ，ブロッキングは生じない．

- 観測機能：T 法の動作の場合，IFP は定期的に CIF の経路表を更新し，各ポートから割り当てられたタイムスロットの間，パケットを取り込む．P 法の動作の場合，1 つのポートの全入力パケットを取り込み，乱数¹²⁾を用いて抽出確率でパケットを取捨する．各要素モニタの観測機能を以下に示す．

パケット数モニタ (要素モニタ 0): 観測対象ポート P に該当する CIF の入力パケット数カウンタを，IFP で定期的にポーリングする．抽出期間 s_t ごとに，観測対象ポート P からの入力パケットの総数を求める，全数調査のモニタ．

$$- R_{e0} = m_{e0}(P, \text{全パケット}, s_t, 1)$$

パケット種別モニタ (要素モニタ 1, 要素モニタ 2): 観測対象ポート P に該当する CIF から，パケットを抽出確率 s_p で IFP に取り込み，宛先 IP アドレスで分類する．抽出期間 s_t ごとに，宛先 IP アドレスが指定 IP アドレスであるパケットの数と，それ以外のパケットの数を求める，標本調査のモニタ．

$$- R_{e1} = m_{e1}(P, \text{宛先 IP} = \text{指定 IP}, s_t, s_p)$$

$$- R_{e2} = m_{e2}(P, \text{宛先 IP} \neq \text{指定 IP}, s_t, s_p)$$

- 推定機能：処理の軽減，リアルタイム性，近似性のために，簡単な補完曲線を用いて推定する．ある抽出期間の推定のために，前後の 2 つずつの 4 点の観測結果から 3 次曲線を求め，台形積分を適用する．

$$- R_{c0} = R_{e0}$$

$$- R_{c1} = m_{c1}(R_{e1}, \text{3次曲線と台形積分})$$

$$- R_{c2} = m_{c2}(R_{e2}, \text{3次曲線と台形積分})$$

- 導出機能：パケット種別モニタが標本調査した入力パケットの内訳が，パケット数モニタが観測した，全入力パケットの内訳を代表すると仮定し，元のトラフィックを導出し，内訳の合計を保証する．

$$- R_m = m_m(\{R_{c0}, R_{c1}, R_{c2}\}, \text{内訳を求める})$$

$$= \left[r_{c0} 0 \frac{r_{c1} 0}{r_{c1} 0 + r_{c2} 0}, r_{c0} 1 \frac{r_{c1} 1}{r_{c1} 1 + r_{c2} 1}, r_{c0} 2 \frac{r_{c1} 2}{r_{c1} 2 + r_{c2} 2} \dots \right]$$

$$r_{c0} i \quad R_{c0}, \quad r_{c1} i \quad R_{c1}, \quad r_{c2} i \quad R_{c2} \quad (i=0, 1, 2, \dots)$$

5. 基本特性

COREswitch に 3 台の端末を接続し，端末 a から端末 b と，端末 a から端末 c へ UDP を用いてトラフィックを生成する．端末 a が接続されたポートで

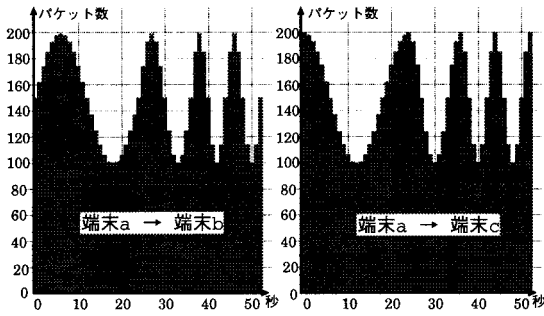


図5 生成したトラフィック
Fig. 5 Generated traffic.

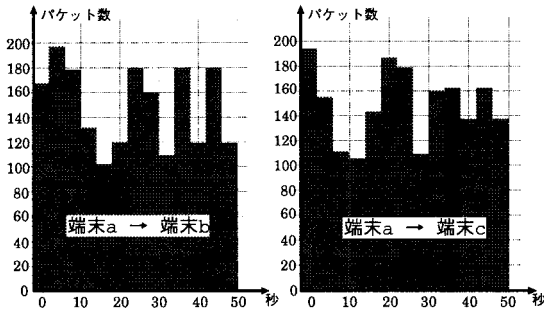


図6 平均化した結果
Fig. 6 Generated traffic on average.

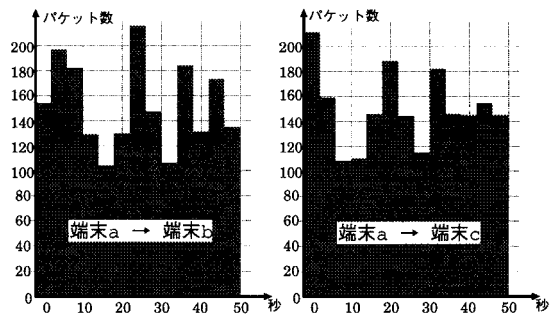


図7 P法による観測結果
Fig. 7 A record with the P method.

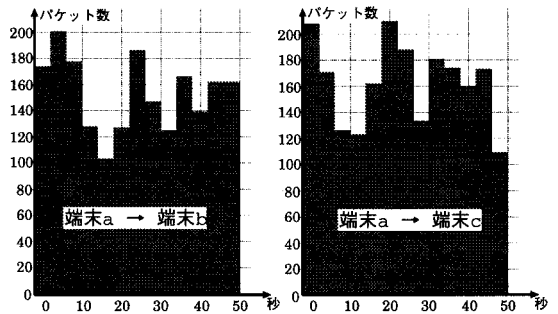


図8 T法による観測結果
Fig. 8 A record with the T method.

ケットの行き先を観測し、基本的な特性を考察する。COREswitchの最大ポート数が16のため、全ポートを観測対象にできるように、抽出期間を4秒 ($s_t=4$ 秒)、タイムスロットを1/4秒 ($s_p=1/16$)とした。2つのトラフィックは、周期を24, 12, 8, 8秒と短くしながら、1秒単位でパケット数を変化させた(図5)。さらに、抽出期間(4秒)で平均化し、図6に示す。

図6と図7のP法による観測結果を比べると、時間同期がとれていないため、少し時間軸をずらして平均化した結果になっているが、近似性が分かる。図6と図8のT法による観測結果を比べると、変動周期が抽出期間の3倍から2倍程度まで短くなる、30秒を経過した付近から増減が鈍っており、平滑化の発生が推測される。

これに対して、各抽出期間ごとに、タイムスロットを単位時間とし、単位時間と観測対象ポートからの総累積入力パケット数とで相関係数を求める(図9)。図9では、30秒経過した付近から値が小さくなり、平滑化の発生を示唆している。このように平滑化の発生が分かる場合には、システムの性能が許す範囲で、抽出期間を短くして対応できる。

この相関係数を用いる方法では、パケット数の合計が一定のまま、割合が変化する場合には、平滑化の発生は推測できない。このような場合の簡単な実験を行

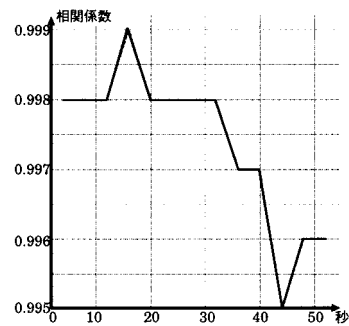


図9 抽出期間ごとの相関係数
Fig. 9 A value of correlation coefficient every sampling period.

う。2つのストリームに対して、パケット数の合計を一定にし、割合(0から100%)と、その割合が継続する時間(0から8秒)を乱数¹²⁾で変化させる。抽出期間が、最大継続時間の50%, 100%, 200%となる4, 8, 16秒の場合について、160秒間観測する。各抽出期間での観測結果の誤差を求め、誤差を累積度数グラフに示す(図10)。抽出期間が短いほどグラフの立ち上がりが早く、誤差の少ない抽出期間の数が多いことが分かる。観測誤差が20%以内である抽出期間の数が、全抽出期間の数に占める割合を求めれば、表1のとおりである。

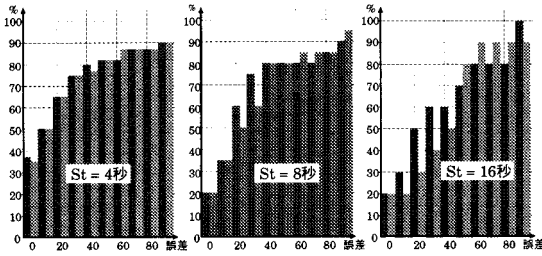


図 10 抽出期間と誤差の関係

Fig. 10 A relationship between sampling period and error range.

表 1 観測精度

Table 1 A precision of observation.

抽出期間の長さ(秒)	4	8	16
誤差 20%以下の割合(%)	50	35	25

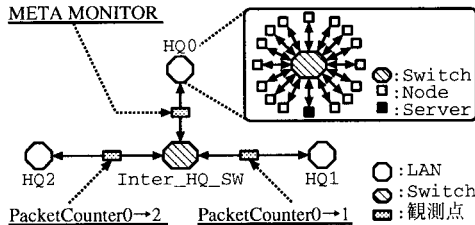


図 11 仮想ネットワーク
Fig. 11 A virtual network.

6. 実ネットワークへの適応性

ネットワークシミュレータ(OPNET)を用い、実際に近いトラフィックを仮想ネットワークで発生させる。これをパケットカウンタとメタモニタで観測し、その結果を比較することで、メタモニタの適応性を考察する。仮想ネットワーク(図 11)は、HQ0, HQ1, HQ2, の3つのLANからなり、各LANは15台のノードと1台のサーバを含む。リンクはすべてギガビットイーサである。また、個々のノードがLAN規模のトラフィックを生成する設定とした(表 2)。

仮想ネットワークの、3つの観測点の内容を表 3 に示す。2つはパケットカウンタであり、1つはメタモニタである。観測点“METAMONITOR”では、推定機能を使用しない代わりに、導出機能で4回分の観測結果の平均を求める。これは推定機能で補間曲線を用いるのと同様に、前後の観測結果を考慮する方法である。抽出期間は1秒であるが、観測結果の分解能は4秒になる。

HQ0からの総パケット数を図 12 に示す。図 12 から、実際には細かなトラフィック変動があることが分かる。これは、トラフィック生成パターンが次の特徴を持つためだと考えられる。

表 2 トラフィック生成の設定
Table 2 A configuration of traffic generation.

プロトコル	パラメータ	設定値
HTTP	サーバ選択	ランダム
	閲覧ページ数/サーバ	10 ページ
	閲覧ページ数/時間	3000 ページ
	オブジェクト数/ページ	10 オブジェクト
	平均オブジェクトサイズ	12 Kbyte
FTP	最大接続数	50 コネクション
	サーバ選択	ランダム
	ファイル転送数/時間	500 回
	平均サイズ	10 Kbyte
e-mail	Get:Put	1:1
	サーバ選択	ランダム
	受信メール数/時間	500 通
	送信メール数/時間	500 通
	平均サイズ	2 Kbyte

表 3 観測点

Table 3 Observation points.

観測点	内容
Packet Counter0 1	HQ0 から HQ1 への全パケット数を計数するカウンタ。
Packet Counter0 2	HQ0 から HQ2 への全パケット数を計数するカウンタ。
META MONITOR	HQ0 からのパケットの中で HQ1 へのパケット数と、HQ2 へのパケット数を観測するメタモニタ。 $R_{e0} = m_{e0}(P, HQ0 \text{ からのパケット}, 1 \text{ 秒}, 1)$ $R_{e1} = m_{e1}(P, \text{宛先 IP} = HQ1, 1 \text{ 秒}, 1/5)$ $R_{e2} = m_{e2}(P, \text{宛先 IP} = HQ2, 1 \text{ 秒}, 1/5)$ $R_m = m_m(\{R_{e0}, R_{e1}, R_{e2}\})$; 内訳を求めさらに 4 回分の計算結果の平均をとり最終的な観測結果とする)

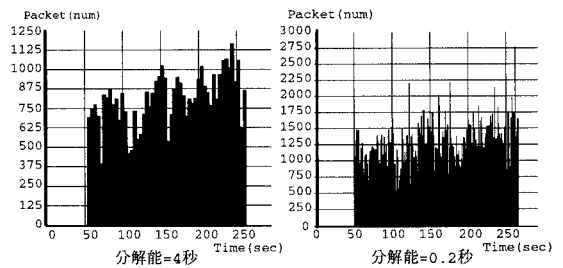


図 12 HQ0からのトラフィック
Fig. 12 Traffic from HQ0.

- インタラクティブなアプリケーション。
- 回線速度に対して操作あたりの転送量が少ない。
- 映像や音などの継続転送がない。

HQ0 から HQ1 へのパケット数について、観測点“METAMONITOR”と“PacketCounter0 1”の観測結果を図 13 に示す。2つの観測点の観測結果を比較すれば、A-1 から A-5, B-1 から B-5 と、ともに5つのピークがあり、A-1 と B-1 といった対応する各ピークについても、近似性が確認できる。また、HQ0

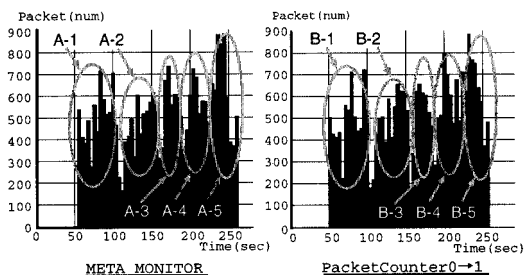


図 13 HQ0 から HQ1 へのトラフィック
Fig. 13 Traffic from HQ0 to HQ1.

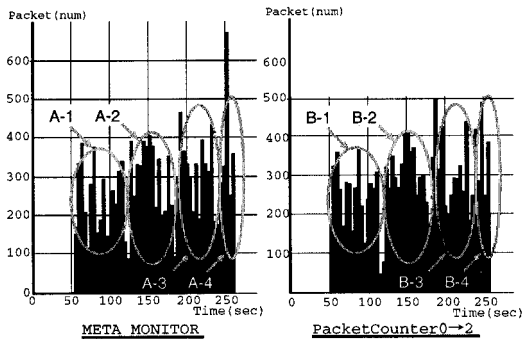


図 14 HQ0 から HQ2 へのトラフィック
Fig. 14 Traffic from HQ0 to HQ2.

から HQ2 へのパケット数について、観測点“META-MONITOR”と“PacketCounter0 2”の観測結果を比較すれば(図 14)、同様に近似性が確認できる。

図 12 が示すような、実ネットワークで起こりうる細かな変動に対して、前節で示した抽出期間を短くする方法や抽出確率を上げる方法だけで対応することは、処理能力上好ましくない。しかし、図 13, 14 から、観測結果の平均をとることで、統計多重効果を利用して見かけ上の変動を減らし、対応可能なことが分かる。また、平均をとる観測結果数を増やせば、抽出確率を低くしても、近似性を保てる傾向がある。この方法は、計算量が少ない反面、分解能が落ちる欠点がある。しかし、一般的なトラフィック測定ツールである MRTG¹⁾の標準分解能の 5 分と比べれば、分解能は十分に高いといえる。

7. おわりに

本稿では、Gb/s クラスが実用水準に達してきたネットワークの多様な運用管理のために、1) 負荷分散: 複数の異なる性質のモニタで協調的に観測する、2) 負荷軽減: 観測負荷に応じて全数調査と標本調査を選択する、3) 処理能力の最適配置: 観測要求や現在の観測結果などの状況に応じて各モニタの観測焦点を変更することで、比較的低い処理能力とシンプルな構成で、柔

軟に“高”トラフィックを観測をする、メタモニタをモデル化した。また、観測系とフィードバック系からなるメタモニタの観測系について、実現法、基本特性、実ネットワークへの適応性を示した。5 章、6 章の実験結果から、メタモニタ方式は、トラフィックの概観観測に耐えらるゝと考える。障害解析など、詳細観測が必要な場合について、1 ポート以上の全入力パケットを指定キーで分類可能な処理能力があれば、これを概観観測時は各ポートに分散させ、詳細観測時は 1 ポートに集中させることで対応できる。

Gb/s クラスの高速ネットワークを対象に、特に複数ポート関連付けてトラフィックを観測するためには、OC3MON^{6),7)}のような、汎用システムによるアプローチでは性能限界が低く、NMVCsystem⁸⁾のような、専用システムによるアプローチでは高コストになるといった問題があった。これは、観測対象ポートの全入力パケットを処理対象とするためである。メタモニタ方式では、統計的調査の考え方に基づき、処理対象パケットを $1/n$ にすることで、必要な処理能力を抑制している。またこのために、パケットの分類など、柔軟性が求められる処理に、ソフトウェア処理の導入が容易である。

今後、特にフィードバック系について、観測系の特性をふまえたうえで、適応的に観測焦点の変更を可能にする、メタモニタのパラメータの記述方法、それらを制御する API、観測シナリオの記述方式などを検討し、メタモニタを高度化する。これらの良否判定は、管理者やシステムが、ネットワークを観測するときの戦略に依存し、ドメイン知識によるところが大きく、より現実的な環境での評価を要する。

謝辞 日頃ご指導いただく皆様方に深謝いたします。

参考文献

- 1) MRTG (Multi Router Traffic Grapher), <http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html>
- 2) Waldbusser, S.: Remote Network Monitoring Management Information Base, RFC1757 (1995).
- 3) Waldbusser, S.: Remote Network Monitoring Management Information Base Version2 Using SMIV2, RFC2021 (1997).
- 4) 串田高幸: インターネットの TCP トラフィックの解析, 情報処理学会研究会報告, 97-DSP-84-4, pp.19-24 (1997).
- 5) 小松原, 鈴木, 茂木, 三上: インターネット・トラフィックの短期的特性の分析, 情報処理学会研究会報告, 97-DSM-7-3, pp.13-18 (1997).

- 6) Apisdorf, J., Claffy, K., Thompson, K. and Wilder, R.: OC3MON: Flexible, Affordable, High-Performance Statistics Collection, *Proc. INET97'* (1997).
- 7) OC3MON, <http://www.nlanr.net/NA/Oc3mon/>
- 8) Parulkar, G., Schmidt, D., Kraemer, E., Turner, J. and Kantawala, A.: An Architecture for Monitoring, Visualization and Control of Gigabit Networks, *IEEE Network*, Sep./Oct., pp.34-43 (1997).
- 9) Dittia, Z.D., Cox Jr, J.R. and Parulkar, G.M.: Design of the APCI: A High Performance ATM Host-Network Interface Chip, *Proc. IEEE INFOCOM'95*, pp.179-187 (1995).
- 10) 丸山, 小倉, 八木, 川野, 村上, 高橋: 並列分散型高速通信スイッチ COREswitch, 信学技報告, CPSY98-156 (1999).
- 11) 小倉, 高橋, 丸山, 八木, 川野: COREswitchにおけるマルチキャスト方式, 第56回情報処理学会全国大会論文集(3), pp.446-447 (1998).
- 12) Matsumoto, M. and Nishimura, T.: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, *ACM Trans. Modeling and Computer Simulation*, Vol.8, No.1, pp.3-30 (1998).
- 13) 八木, 高橋, 丸山, 小倉, 川野: 高速ネットワーク向け統計的トラフィック観測機構の提案, 第57回情報処理学会全国大会論文集(3), pp.417-418 (1998).
- 14) 八木, 小倉, 川野, 丸山, 高橋: メタモニタ: 適応型ネットワークトラフィック観測機構, 情報処理学会研究会報告, 99-DPS-92-2, pp.7-12 (1999).
- (平成11年5月10日受付)
(平成11年12月2日採録)



八木 哲 (正会員)

昭和41年生。平成4年大阪大学大学院工学研究科電子工学専攻博士課程前期修了。同年日本電信電話(株)入社。ソフトウェア工学, 交換機シミュレータ, 計算機ネットワークシステム, 広域データベース等の研究開発に従事。現在, NTT未来ねっと研究所所属。



小倉 毅 (正会員)

平成4年神戸大学工学部システム工学科卒業。平成6年同大学院工学研究科システム工学専攻修士課程修了。同年日本電信電話(株)入社。現在, NTT未来ねっと研究所にて, 高速通信システムアーキテクチャの研究に従事。



川野 哲生 (正会員)

平成8年九州大学大学院総合理工学研究科情報システム学専攻修士後期課程修了。同年日本電信電話(株)入社。現在, NTT未来ねっと研究所にて, 高速通信プロトコル, および, 高速通信システムアーキテクチャの研究に従事。博士(工学)。電子情報通信学会会員。



丸山 充 (正会員)

昭和58年電気通信大学電気通信学部応用電子工学科卒業。昭和60年同大学院修士課程修了。同年日本電信電話(株)入社。以来, 高精細画像情報提供システム, ビデオ・オン・デマンドシステム, コンピュータネットワークと実時間並列分散アーキテクチャの研究開発に従事。現在, NTT未来ねっと研究所主任研究員。工学博士。電子情報通信学会, 日本ソフトウェア科学会, IEEE, ACM各会員。



高橋 直久 (正会員)

昭和26年生。昭和49年電気通信大学電気通信学部応用電子工学科卒業。昭和51年同大学院修士課程修了。同年日本電信電話公社武蔵野電気通信研究所入所。以来, 機能分散型並列計算機, データフロー型計算システム, ソフトウェア工学, コンピュータ・ネットワーク等の研究に従事。現在, NTT未来ねっと研究所並列分散アーキテクチャ研究グループリーダー。工学博士(東工大)。電子情報通信学会, 日本ソフトウェア科学会, ACM各会員。