マルチメディア通信と分散処理 86－43
グループウェア 26－43
（1998. 1. 30）

インターネットにおけるトラフィックの測定方式

串田高幸

日本アイ・ビー・エム株式会社
東京基礎研究所

インターネットは、世界中のたくさんの組織が 相互接続されているために集中したトラフィックの 測定及び管理を行なうことが不可能である。また インターネットでは、現在、膨大な量のパケットが 転送されていて、トラフィックの状況は、刻々と 変化している。トラフィックの共通パラメーターを 定めて、そのパラメーターにもとづいて各所で 独立してトラフィックを測定することができれば、 インターネットに接続されている各ネットワークが、 どのような状況になっているかわかる。 本稿では、このような測定を行なうために インターネットのトラフィックの測定装置の アーキテクチャとその実装方式について述べる。

The apparatus for the Internet traffic

Takayuki Kushida(kushida@trl.ibm.co.jp)

IBM Research,
Tokyo Research Laboratory

The Internet is widely installed among many countries and many organizations are managed and interconnected with each other. It is difficult to understand the network traffic for the entire Internet. The network traffic is varying at the measurement time. If we have the same set of parameters for the Internet traffic, we can compare the status of the network. This paper describes the architecture of the traffic measurement system for the Internet traffic. Its system is focused on the flow for the Internet.

1. はじめに

インターネットは、世界中のたくさんの組織が相互接続されているために集中したトラフィックの測定及び管理を行なうことが不可能である。またインターネットでは、現在、膨大な量のパケットが転送されていて、トラフィックの状況は、刻々と変化している。もしトラフィックの共通パラメーターを定めて、そのパラメーターにもとづいて各所で独立してトラフィックを測定することができれば、インターネットに接続されている各ネットワークが、どのような状況になっているかを知ることができる。また、今後、インターネットがさらに拡大してくると、トラフィックの解析を行ない、その状況を把握することが、ネットワーク管理の分野において重要な要素となる。

トラフィックを正しく測定することができれば、ネットワークの使用率や課金に対しても有効に利用することができる。さらに複数のネットワークにおいて、トラフィックの共通のパラメータを比べることができれば、個別のネットワークの状況を比較を容易に行なうことができる。

一般にネットワークにおいてトラフィックを解析することによって以下の項目を得ることができる。

- 現在のネットワークの動作状況
- 拡張に対する計画
- パフォーマンスの品質
- サービスの質のチェック
- ユーザーごとの使用量

本稿では、まず、インターネットにおけるトラフィックについて関連研究について述べる。そして、「流れ」を利用したトラフィックの測定方式について述べる。この測定方式を使ってネットワークのトラフィックを収集するためのシステムについて述べる。さらにアプリケーションプログラムとして、ユーザーが動的にその測定パラメータを変化させることによって、多角的に視覚化して解析を行なう方式についても述べる。

2. 関連した研究

B. A. Mah は、ネットワークトラフィックのうちHTTP(Hyper Text Transfer Protocol) に着目して報告している [1]。この報告では、HTTP のトラフィックがパラメータによってどのように変化するかを次の項目について調べた。

1. HTTP の Request の長さの分布
2. Reply の分布
3. 1 ページの長さ (複数の連続した Reply の長さ)
4. 最初に到着したファイルの長さとそれ以外のときに到着したファイルの長さ
5. ユーザーが考えている待ち時間

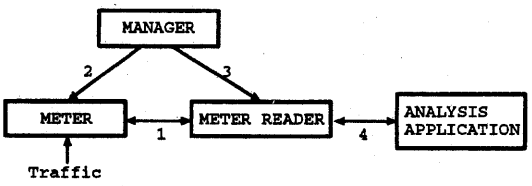

図 1: トラフィック測定装置の構成

この結果、1)HTTP のサイズは、bimodal 分布になっている。2)HTTP の返答のサイズは、heavy-tailed 分布になっていて要求のサイズよりも大きくなっている。80%のドキュメントは、4 つ以下のファイル転送にが行なわれている。最初のページは、それに続くページよりも大きい。さらにサーバーのドキュメントの約 80%は、6 つ以下のページからできていることを報告している。

また、J.A. Zinky らは、ネットワークのトラブルを解析するためのツール SpyGlass の開発について報告している [2]。このツールでは、パケットの解析に対して異なったタイプのダイヤグラムを用意して、その異なったダイヤグラムのオブジェクト間の関連を表示するためにリンクをハイライトしている。また、生のパケットからテキスト形式のダイヤグラムを生成する環境についても述べている。しかし、このツールは、トラフィックの解析への使用というよりも、むしろプロトコルの解析を主な目的にしている。

我々は、以前よりインターネットのトラフィックを収集して解析する研究をしてきた [3][4] [5][6]。これらの一連の研究では、収集解析装置を使用して、一旦、すべてのパケットを蓄積して、そのデータを解析した。

一方、IETF(Internet Engineering Task Force) では、その分科会として、RTFM(Realtime Traffic Flow Measurement)- WG がある。この分科会では、トラフィックの収集を実時間で行なうことをめざしている。アーキテクチャとして Experimetal RFC(Request For Comments) の RFC2063 が発行されている [7]。この RFC は、トラフィックを「流れ」として捉えて、その測定方式のアーキテクチャについて述べている。

本稿では、この RFC2063 を基本にして、今までの研究を加えてトラフィックを測定装置をどのように実装するかについて述べる。

3. 構成

ネットワークのトラフィック測定装置の構成として大きく 4 つのエンティティに分割する。図 1が、トラフィック測定装置の構成である。これらの 4 つのエンティティは、1. マネージャー (Manager)、2. メーター (Meter)、3. メーターリーダー (Meter Reader)、そして 4. 解析ア

プリケーション (Analysis Application) である。これらのエンティティについてそれぞれ機能を説明する。

1. マネージャー (Manager)
   このエンティティは、メーターの設定とメーターリーダーの制御を行なう。また、それぞれのメーターを適切な設定にするために解析アプリケーションが持っているデータを利用することもある。
2. メーター (Meter)
   メーターは、トラフィックの測定点に置かれる。それぞれのメーターは、設定により選択的にネットワークのトラフィックを収集する。そして、その収集されたデータをメーターリーダに転送する。処理して保存されたデータは、「使用量データ」と呼ばれる。
3. メーターリーダー (Meter Reader)
   メーターリーダーは、メーターから使用量データを解析アプリケーションに転送するためのエンティティである。
4. 解析アプリケーション (Analysis Application)
   解析アプリケーションは、ネットワーク管理に有効な情報や報告を出すために「使用量データ」を処理するエンティティである。

各コンポーネント間のインタラクション

4つのエンティティが定義されると、そのエンティティ間のインタラクションを定義する必要がある。ここでは、次の4つのエンティティ間のインタラクションについて述べる。

1. メーターとメーターリーダーの間 (図1の1)
2. マネージャとメーターの間 (図1の2)
3. マネージャとメーターリーダーの間 (図1の3)
4. メーターリーダーとアプリケーションの間 (図1の4)

また、「流れ」を識別するためにアドレス属性のうち次の一つを使用する。

- インターフェイス番号
  トラフィックを測定したインターフェイスの番号
- 隣接した階層のアドレス
  「流れ」のパスにおいてn-1層のソースと目的のアドレス。
- ピアアドレス
  パケットレベルでのソースと目的のアドレス。
- トランスポート・アドレス
  トランスポート層のソースと目的のポート番号。

測定の精度

「流れ」の精度は、次のような調整をして制御する。

- アカウントの可能なアドレス属性ごと。
- パケットの属性ごと。

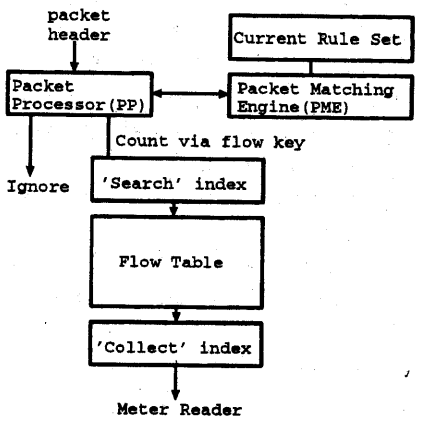

図 2: メータの構造

- 「流れ」の生成時間及び「流れ」の間隔ごと (報告される間隔は、精度で測定するために十分に短い必要がある。)

4. メーター

メーターは、トラフィックを測定する場所にある装置である。そして、その内部は、図2のような構造になっている。

- 収集されたパケットヘッダー (Packet Header) は、パケットプロセッサー (Packet Processor,PP) に転送される。
- PP からパケット照合エンジン (Packet Matching Engine, PME) が呼び出される。
- PME は、現在の規則の集合 (Current Rule Set) を設定した仮想マシンが走っている。そして、PME は、パケットにどのような処理をすればよいかを PP に返す。
- この処理のあとにパケットのいくつかは、「IGNORE 状態」になり PP から除去される。
- パケットが一致したら「流れ」を記述するために FLOW キーを返す。
- FLOW キーは、「流れ」のテーブルの中の「流れ」のエントリーの場所を見つけ出すために使用される。新しいエントリーは、「流れ」が最初に見つけ出された時に生成される。
- メーターリーダーは、どのようなときでも「流れ」のテーブルからデータを集めることができる。また、収集のためにインデックスを使用してもよい。

「流れ」のテーブル

この節では、図2の「流れ」のテーブル (Flow Table) について説明する。トラフィックメーターは、検出された

「流れ」のレコードに対するテーブルを持っている。これが「流れ」のテーブルである。「流れ」のレコードは、「流れ」に対して下記のような属性を持っている。

- 「流れ」のソースと目的アドレス。アドレスとマスクからできている。パケットが、収集された箇所のインターフェイスの数。
- 「流れ」において、パケットが最初と最後に検出された時間。
- 「流れ」を「forward」(ソースから目的) と「backward」(目的からソース) にそれぞれカウントする項目。
- その他の属性。(例えば、「流れ」のレコードの状態。) 「流れ」は、次の状態を持っている。

  INACTIVE メーターでは、まだ使用されていない状態。

  CURRENT レコードが使用中の状態。

  IDLE レコードが使用中の状態。しかし、ある周期において測定されたデータが何も来ない状態。

パケットの取扱い

「流れ」のテーブルを使った処理は、以下の方法で行なわれる。

- パケットヘッダーから同じ属性値を使ってキーを生成する。
- 生成されたキーが、現在の規則の集合と比べて一致しているかどうかを調べる。

パケットの取扱いにおいては、以下のような考慮すべき点がある。これらは、

- パケットを「IGNORE」として認識すること
- 「流れ」の両方向でパケットを一致させること
- 「流れ」の一方向だけでパケットを一致させること

である。また、パケットを取り扱うためのアルゴリズムは、次の機能である。ここで $A->B$は、ソースから目的への「流れ」という意味である。

- 一致 ($A->B$): Ignore, Fail, Suc と 3 つの状態がある。
- 現在 ($A->B$): $A->B$が現在の「流れ」の場合
- 生成 ($A->B$): テーブルに $A->$B の「流れ」が加わるとき
- カウント ($A->B$、前方向): 「流れ」$A->B$に対する forward カウンターを増やす
- カウント ($A->B$、逆方向): 「流れ」$A->B$に対する reverse カウンターを増やす

このようにこのメーターでは、通常、両方向にそれぞれパケットを数えることができる。

収集

規則の集合は、メーターの中にテーブルとして記憶される。このテーブルの一列が、「現在の規則」になる。テーブルの一列と収集されたパケットが一致するかどうかを調べる。テーブルの構造は、次のようになっている。

```
+-------- test ---------+   +---- action -----+
attribute & mask = value:   opcode,  parameter;
```

opcode には二つのフラグ「goto」と「test」がある。PME は、test indicator という論理値の識別子を保持して、その初期値は on なっている。もし識別子が、on ならばテストを行なう。もし off ならば、opcode を動作させる。この opcode の動作の種類は、全部で 15 種類ある。

「流れ」のテーブルの保存

「流れ」のテーブルは、一つ一つの「流れ」のレコードの列が集まってできている。そして、メーターが動作しはじめたときに「流れ」は何もなく、すべての「流れ」のレコードは、「INACTIVE」状態になっている。

測定するときにそれぞれのパケットは「流れ」として認識される。しかし、「流れ」は、現在の「流れ」のレコードとしてセットされていないこともある。もし「流れ」のレコードが、現在の「流れ」としてセットされていれば、その状態を「CURRENT」と呼ぶ。新しい「流れ」が見つけられたとき、メーターは、テーブルの中の「INACTIVE」状態の「流れ」のレコードを検索する。また、テーブルの中では、「流れ」のレコードの順番に特別な差はない。

5. メーターリーダー

使用量のデータは、メモリーが許す限り、通常、報告する間隔でメーターからメーターリーダーに収集される。この報告する間隔は、マネージャが定義する。収集されたデータは、使用量のレコードの列として「FLOW DATA FILE」という名前でディスクに保管される。

また、個々の「流れ」を識別するために、スタート時間、規則の集合の ID とサブスクリプト (流れテーブルの列の番号) を使用する。

使用量のレコード及び流れのデータファイル

収集された使用量のデータは、メーターリーダーごとに別々のデータファイルとして記録しておく。収集されたメーターが、すぐにわかるように「流れ」のデータファイル (測定された使用量のデータ) には、メーターの情報をもつようにする。

また「流れ」の測定量は、一つの流れにおけるバイト数とパケット数である。

使用量のレコードは、「流れ」を表している。使用量のレコードは、メーターのエンティティに対する 1) 識別子 (ネットワークアドレス)、2) 時間スタンプそして 3) メーターの識別子 (流れのデータレコード) の 3 つからできている。

```
-------- -------- -------- -------- --------
1. RECORD ID.
   Meter ID, Timestamp, Collection Rules ID
--------
2. Flow ID.
   Address List, Subscriber ID, Attributes
--------
3. COUNTERS
   Packet Count, Byte Count, Flow Start/Stop Time
-------- -------- -------- -------- --------
```

メーターリーダーの使用量のファイルは、使用量レコード (上記の 1 から 3 まで) を順次つけたして増やす。この方法によって周期的に集められた「流れ」の使用量レコードのファイルを作る。このファイルを「FLOW DATA FILE」と呼ぶ。

メーターからメーターリーダーの転送

メーターからメーターリーダーへの使用量のレコードの転送は、正確で信頼性が高く、さらに十分なセキュリティが確保されている必要がある。もしメーターに MIB が定義されていれば、SNMP を使ってこの操作を行なうができる。この SNMP を使用する方法は簡単な例であって、他のアクセス方法を使ってもよい。

通常の操作において、メーターは、「流れ」の精度に必要な度数の桁数を用意しておく。そのため、規則のファイルにその精度を定義しておかなければならない。最悪の場合、「流れ」のトラフィックがメーターで使用できるメモリよりも多くなって動作しているような状況がある。そのような場合に対処するため、自動的に精度を低くする機能を入れておく。

6. マネージャ

マネージャは、メーターを設定をしたり、メーターリーダーを制御したりする。マネージャは、以下の項目のようなインタラクションを介してこれらの作業を行なう。

マネージャとメータ：制御機能

- 規則の集合をダウンロードする
- 規則の交換
  規則がダウンロードされれば、マネージャは、どの規則の集合が現在走っているか、そしてどれをスタンバイさせるかメーターに知らせることができる。
- ハイウオーターマークのセット
  このときの比率の値は、スタンバイしている規則の集合にスイッチしたときに、メータにおいて解釈される。「流れ」の測定精度をよくするためにメーターの「流れ」に対する記憶容量を十分に残しておく。
- 流れの終了パラメータのセット
  メーターは、資源がなくなるとメーターのテーブルから「流れ」のレコードを消去する。これは、次のような場合である。
  - メーターリーダーに最後に報告した後、何もトラフィックがない状態
  - 古い流れがまだある状態
  - 報告していないパケットが少しだけ流れている状態
- 「INACTIVE」からのタイムアウトのセット
  このパラメータは、流れのパケットを最後に見つけてからの待ち時間 (秒単位) である。「流れ」のレコードにから、待ち時間が範囲内であってそしてアイドルであることがわかれば、パケットを現在の収集の規則に従って集める。

マネージャとメータリーダー：制御機能

トラフィックの「流れ」の測定では、機能を適切に動作させるためにセットしなければならないたくさんのパラメーターがある。この操作は、静的なメーターの構成とはまったく逆の動的なネットワークの管理である。

この操作は、メモリとのトレードオフになる。もしメモリが少ないと、メーターの中にあるデータの記録 (使用量) の間隔を増加させる必要がある。

また、通常、ネットワークのバンド幅は、ユーザのデータを運ぶために使用される。そのため、使用量の間隔の変更は、メータとメータリーダーの間の仮想リンクになっているネットワークのバンド幅を制限する。この制限されたところにおいて、ネットワーク管理の操作は、次のようになる。

- マネージャとメーターリーダーの識別子
  マネージャは、メーターが収集しているトラフィックのうち、正しい集合をだけを報告する。これは、使用情報への許可されていないアクセスを妨ぐためである。
- 報告の間隔を制御する
  通常の報告の間隔は、トラフィックのパターンにより選択される。しかし、報告されている間隔を正しくセットにしたときでさえ、トラフィックの量が爆発的に増加すれば、メモリーを使い切ってしまう。メモリーを使い切って (高いウオーターマークの位置) しまうような危険な状態があることをマネージャに知らせるためにメーターに対するなんらか

の機構がなければならない。

- 精度の制御
  測定の精度の制御は、トラフィックのすべての情報を記録するために高い信頼性を持つ必要がある。また精度は、測定しているシステムの能力に依存する。精度は、それぞれのインターフェイスに対して「流れ」のIDを決めて制御する。
  メーターの現在の規則の集合により精度が制御されているとき、マネージャーは、違う規則の集合に切り替えることをメーターに要求する。新しい規則の集合が必要になったときは、マネージャからダウンロードするか、あるいは、メーターの初期設定の一部としてダウンロードする。
- 「流れ」の生存時間の制御
  「流れ」のタイムアウトのパラメータには、1)「流れ」の最大の生存時間と2) テーブルから「流れ」を除去するタイムアウトのの2つがある。

7. アプリケーション

解析を行なうアプリケーションは、メーターリーダーから使用量のデータを取得して、そのデータを解析するエンティティである。また、解析を行なうために使用量のデータを取得する。しかし、その使用目的によりデータの表現方法がまったく異なる。

例えば、次のような項目がこのアプリケーションから出力される。

- トラフィックの流量の行列
  可能な限り多くのパスに対する「流れ」の総合的な行列。
- 流量の頻度分布
  時間によってどのように流量が変化しているかの分布。
- 使用量データ
  特定のホストにより送受信されたすべてのデータ。

視覚化

ネットワークのトラフィックが膨大な量になれば、数値で表現したデータを理解するができない。そのため、視覚化して利用する方が考えられる。トラフィックを視覚化するには、2次元あるいは3次元のグラフ用いて行なわれる。

このようなデータを視覚化する場合、以下のようにいくつかの方法がある。

- 数値での表示
- 表
- グラフ
- アニメーション

しかし、結果を単純に視覚化するだけでは、トラフィックの解析にならない。色々な尺度からあるいは、いくつかのパラメーターを変更することによって、データを再表示することが解析に有効である。そのため、解析アプリケーションには、インタラクティブなインターフェイスがとなる。さらにこのような操作を行なうためにメーターリーダーから解析アプリケーションには、必要なデータが順次送られる必要がある。

メーターリーダーとアプリケーション

メーターリーダーとアプリケーションの間のインタラクションは、アプリケーションからは、データの要求のためのメッセージが出され、またメーターリーダーからは、その要求の返答にあたるデータがかえる。この2つの間では、必要なデータをアプリーケーションで表示するためにデータのやりとりが行なわれることになる。

8. おわりに

本稿では、トラフィック測定装置について、その概要を述べた。このトラフィック測定装置を実装することによって、ネットワークの流れを実時間で測定することができる。さらにアプリケーションプログラムにより動的にパラメータを変更しながら視覚化されたトラフィックを見て、ネットワークを調べることができる。

参考文献

[1] Bruce A. Mah, An Empirical Model of HTTP Network Traffic, *InfoComm'97 Kobe, pp.592-pp.600 ,1997.*

[2] J. A. Zinky and F. M. White, Visualizing Packet Traces *ACM SIGCOMM'92 MD, USA, pp.293-pp.304, 1992.*

[3] 串田, インターネットのトラフィックを測定及び解析するためのツールの設計及び開発, 情報処理学会マルチメディア通信と分散処理ワークショップ,*1995* 年 *10* 月, *1995.*

[4] 串田, 佐藤, 山内, インターネットにおけるトラフィックの収集と解析情報処理学会マルチメディアと分散処理研究会, *1996* 年 *3* 月, *1996.*

[5] 串田, TCP/IP ネットワークのトラフィックの特徴情報処理学会マルチメディアと分散処理ワークショップ, *1996* 年 *10* 月, *1996.*

[6] 串田, インターネットのTCP トラフィックの解析情報処理学会マルチメディアと分散処理研究会, *1997* 年 *9* 月, *1997.*

[7] N. Brownlee, et. al., Traffic Flow Measurement: Architecture, *Request For Comments 2063, January 1997*