Network Working Group
Request for Comments: 2721
Category: Informational

N. Brownlee
The University of Auckland
October 1999

RTFM: Applicability Statement

Status of this Memo

Copyright Notice

Abstract

This document provides an overview covering all aspects of Realtime Traffic Flow Measurement, including its area of applicability and its limitations.

Table of Contents

1 The RTFM Documents

The RTFM Traffic Measurement System has been developed by the Realtime Traffic Flow Measurement Working Group. It is described in six other documents, as follows:

[ACT-BKG] Internet Accounting: Background (Informational)

Sets out the requirements for a usage reporting system for network traffic. Sketches out the RTFM Architecture (meters, meter readers and managers) allowing for multiple meters and meter readers, with asynchronous reading from the meters. Proposes methods of classifying traffic flows, the need for flows to be bi-directional (with separate sets of counters for each direction) and the need for each packet to be counted in a single flow (the ' count in one bucket' principle).

[RTFM-ARC] RTFM Architecture (Informational)

Defines the RTFM Architecture, giving descriptions of each component. Explains how traffic flows are viewed as logical entities described in terms of their address-attribute values, so that each is defined by the attributes of its end-points. Gives a detailed description of the RTFM traffic meter, with full details of how flows are stored in the meter's flow table, and how packets are matched in accordance with rules stored in a ruleset.

[RTFM-MIB] RTFM Meter MIB (Proposed Standard)

Describes the SNMP Management Information Base for an RTFM meter, including its flow table, rule table (storing the meter's rulesets) and the control tables used for managing a meter and reading flow data from it.

[RTFM-SRL] SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups (Informational)

An RTFM ruleset is an array of rules, used by the meter to decide which flows are of interest, which end-point is the flow source, and how much detail (i.e. what attribute values) must be saved for each flow. SRL is a high-level language providing a clear, logical way to write rulesets. It should also be useful for other applications which select flows and perform actions upon them, e.g. packet-marking gateways, RSVP policy agents, etc.

[RTFM-NEW] RTFM New Attributes (Experimental)

There has been considerable interest from users in extending the RTFM Architecture so as to allow a meter to report on an increased number of flow-related measures. This RFC documents work on specifying such measures (the 'new' attributes) and reports on experience of implementing them.

[RTFM-NTM] RTFM: Experiences with NeTraMet (Informational)

NeTraMet is a free software implementation of the RTFM Architecture which has been available since 1993. This RFC records RTFM implementation experience gained with NeTraMet up to late 1996. One particularly important result is the realisation that groups of rules which test the same attribute using the same mask can be implemented as a single hashed comparison, allowing the meter to rapidly determine whether a packet belongs to one of a large number of networks.

2 Brief Technical Specification (TS)

RTFM provides for the measurement of network traffic 'flows', i.e.

- a method of specifying traffic flows within a network
- a hierarchy of devices (meters, meter readers, managers) for measuring the specified flows
- a mechanism for configuring meters and meter readers, and for collecting the flow data from remote meters

RTFM provides high time resolution for flow first- and last-packet times. Counters for long-duration flows may be read at intervals determined by a manager. The RTFM Meter is designed so as to do as much data reduction work as possible, which minimizes the amount of data to be read and the amount of processing needed to produce useful reports from it.

RTFM flow data can be used for a wide range of purposes, such as usage accounting, long-term recording of network usage (classified by IP address attributes) and real-time analysis of traffic flows at remote metering points.

3 Applicability Statement (AS)

To use RTFM for collecting network traffic information one must first consider where in the network traffic flows are to be measured. Once that is decided, an RTFM Meter must be installed at each chosen measurement point.

At least one Meter Reader is needed to collect the measured data from the meters, and a single Manager is needed to control the meters and meter readers.

RTFM Meters may be single- or multi-user hosts running a meter program (one such program is available as free software, a second is under development at IBM Research). Alternatively, meters could be run as firmware in switches or routers. A hybrid approach in which an RTFM meter takes raw traffic data from a router provides another useful implementation path.

RTFM Managers are programs running on a host, communicating with meters and meter readers via the network. For this purpose meters are SNMP agents implementing the RTFM Meter MIB, and managers are SNMP clients using the Meter MIB to store and access the flow data.

4 Limitations

RTFM is designed to measure traffic flows for traffic passing a point in a network. If packets for a flow pass the metering point in both directions the meter will match them up, providing counters for each direction. If packets only pass in one direction the meter can only provide counts for that direction.

Users of RTFM should note that installing meters, meter readers and managers merely provides one with the capability to collect flow data. Further installation work will be needed to develop configuration files (RTFM rulesets) for each meter, data processing applications to analyse the flow data, and various scripts, cron jobs, etc. so as to create a useful production-quality measurement system which suits a user's particular needs.

One of the strengths of RTFM is its ability to collect flow data at whatever level of detail (or 'granularity') is required. It can be tempting to simply collect 'all possible data', but there are severe resource constraints. If one tries to save the complete address-attribute value for all attributes of every possible flow a very large amount of data may be produced rapidly, but the meter has only a finite amount of memory for its flow table. A better approach is to save the minimum amount of data required to achieve the measurement system goals.

For example, to collect usage data so as to bill subscribers identified by their IP address one could just save the full IP address, nothing more. The RTFM meter would produce flow data for each subscriber IP address, with PDU and Octet counts for data sent and received, which would be the minimum needed to produce bills. In practice one would probably want to save at least part of the

Destination IP address, which would allow the production of usage logs showing subscriber activity over time.

The simplest way to determine how much detail can be collected is to create an initial ruleset which collects the minimum amount, then to modify it step by step, gradually increasing the amount of information saved for each flow. An RTFM meter ought to provide some measures of its own performance (e.g. number of active flows, percentage idle processor time, packets metered, packets not metered). Such measures will be implementation-specific, but should allow a user to assess the impact of each change to the ruleset.

If the network data rate is too high, i.e. the meter reports that it cannot meter all the packets even with the initial ruleset above, one may be able to use other strategies. For example one could

- run the meter on a faster computer, e.g. move from a DOS PC to a workstation, or perhaps use a meter implemented in firmware within a switch or router.

- use sampling. The details of such sampling are not defined within the RTFM Architecture, but the Meter MIB provides one simple method by allowing one to specify that only every nth packet on an interface will be metered. This would probably not be acceptable for producing billing data, but might well be acceptable for traffic engineering purposes.

5 Security Considerations

These are discussed in detail in the Architecture and Meter MIB documents. In brief, an RTFM Meter is an SNMP agent which observes a network and collects flow data from it. Since it doesn't control the network directly, it has no direct effect on network security.

On the other hand, the flow data itself may well be valuable - to the network operator (as billing data) or to an attacker (who may wish to modify that data, or the meter's ruleset(s)). It is therefore important to take proper precautions to ensure that access to the meter and its data is sufficiently secure.

For example, a meter port attached to a network should be passive, so that it cannot respond to login attempts of any kind. Control and data connections to a meter should be via a secure management network. Finally, suitable security should be established for the meter, as it would be for any other SNMP agent.

Meters may, like any other network component, be subjected to Denial of Service and other attacks. These are outside the RTFM Architecture - countermeasures for them are available, but are also outside RTFM.

6 Policy Considerations

When collecting traffic data, one must have well-defined operations policies covering points such as:

- Exactly what data is to be collected, at what level of detail?
- How long will the data be kept?
- What may the data be used for?
- Who will be allowed to see the raw data?
- May summaries of the data be shown to other people?

Policy issues such as these should normally be considered as part of an organisation's Network Security Policy.

Other policy issues relating more directly to the traffic data are essentially part of the measurement system design, such as:

- How much time resolution is required for the data?
  (Less resolution implies longer collection intervals, but that may require more memory in the meters to hold flow data between collections).

- What level of hardware redundancy is needed?
  (A single meter and meter reader is generally enough. For greater reliability, meters and meter readers can be duplicated).

- Who is allowed to use the system?
  (Approved users will need permissions to download rulesets to the meters, and to collect their data, possibly via their own meter readers).

7 Soundness

NeTraMet, the first implementation of the RTFM Architecture, has been in use worldwide since 1994. Currently there are many organisations, large and small, using it to collect traffic data for billing purposes.

One example of these is Kawaihiko, the New Zealand Universities' Network, which has seven RTFM meters located at sites throughout New Zealand. One of the sites is NZIX, the New Zealand Internet eXchange at the University of Waikato, where Kawaihiko has a meter (attached to a 100baseT network) observing traffic flows across the exchange to

each of Kawaihiko's three international Internet Service Providers. 5-minute Octet counts are collected from all the Kawaihiko meters by a single meter reader at Auckland. Traffic data from the meters is used to determine the cost per month for each of the Kawaihiko sites.

It is difficult to estimate how many organisations are using RTFM traffic measurement. There are about 250 people on the NeTraMet mailing list, which often carries questions like 'why doesn't this ruleset do what I meant'? Once new users have the system running, however, they tend to simply use it without further comment.

From time to time the list provides useful feedback. For example, early in 1998 there were two very significant user contributions:

- Jacek Kowalski (Telstra, Melbourne) described an improved hash algorithm for NeTraMet's flow table, which provided almost an order of magnitude improvement in packet-handling performance.

- Kevin Hoadley (JANET, U.K.) reported having problems with very large rulesets. These were resolved, and better methods of downloading rules developed, allowing NeTraMet to work well for rulesets with more than 32,000 rules.

Perhaps one reason why there is little discussion of NeTraMet's use in collecting billing data is that users may consider that the way collect their data is a commercially sensitive matter.

8 Appendix A: WG Report on the Meter MIB

The Meter MIB (in its current form) was developed early in 1996. It was produced as an SNMPv2 MIB, following a number of detailed (and continuing) discussions with David Perkins beginning at the Dallas IETF meeting in December 1995.

There are two current implementations:

- NeTraMet (Nevil Brownlee, The University of Auckland)

- IBM Meter (Sig Handelman & Stephen Stibler, IBM Research, N.Y, Bert Wijnen provided further help with SNMP)

The NeTraMet meter is a stand-alone SNMP agent using an SNMPv2C implementation derived from CMU SNMPv2.

The IBM meter runs as a sub-agent on an AIX system. All the meter code has been written by Stephen Stibler - it was not derived from the NeTraMet code. Stephen has found it useful to use nifty, one of NeTraMet's manager/reader programs, to test the IBM meter.

As indicated above, there have only been two implementors to date, and the Working Group consensus has been very strong.

The MIB has one unusual aspect: the method used to read large amounts of data from its Flow Table. An earlier SNMPv1 version of the MIB was in use from 1992 to 1997; it used opaque objects to read column slices from the flow table for flows which had been active since a specified time. This was very non-standard (or at least very application-specific).

With the change to SNMPv2 we were able to use 64-bit counters for PDUs and Octets, RowStatus variables for control tables and GETBULK requests to read rows from the flow table. We also use the TimeFilter convention from the RMON2 MIB to select flows to be read; this gives the meter MIB a strong resemblance to RMON2.

The current MIB introduces a better way of reading large amounts of data from the flow table. This is the 'DataPackage' convention, which specifies the attribute values to be read from a flow table row. The meter returns the values for each required attribute within a BER-encoded sequence. This means there is only one object identifier for the whole sequence, greatly reducing the number of bytes required to retrieve the data. The combination of

TimeFilter: to select the flows to be read
DataPackage: to select the attributes required for each flow
GetBulk: to read many flows with a single SNMP PDU

provides a very effective way to read flow data from a traffic meter.

9 References

[ACT-BKG] Mills, C., Hirsch, G. and G. Ruth, "Internet Accounting Background", RFC 1272, November 1991.

[RTFM-ARC] Brownlee, N., Mills, C. and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999.

[RTFM-MIB] Brownlee, N., "Traffic Flow Measurement: Meter MIB", RFC 2720, October 1999.

[RTFM-NEW] Handelman, S., Stibler, S., Brownlee, N. and G. Ruth, "RTFM: New Attributes for Traffic Flow Measurement", RFC 2724, October 1999.

[RTFM-NTM] Brownlee, N., "Traffic Flow Measurement: Experiences with NeTraMet", RFC 2123, March 1997.

[RTFM-SRL] Brownlee, N., "SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups", RFC 2723, October 1999.

10 Author's Address

Nevil Brownlee
Information Technology Systems & Services
The University of Auckland
Private Bag 92-019
Auckland, New Zealand

Phone: +64 9 373 7599 x8941
EMail: n.brownlee@auckland.ac.nz

11 Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.