Statistics & Measurements: From Network Research to Network Operations

Hervé Guy & René Hatem, CANARIE Inc.

Abstract – An innovative network measurement analysis system has been developed as part of the CA*net3 advanced networks program. The system, which builds on the measurement capability of a number of low-cost and innovative M&P tools, is intended to assist network operators in a real world networking environment to diagnose problems quickly and effectively. A large DBMS structure for collecting all relevant data in near real time is the core part of the architecture.

I. INTRODUCTION

Measurement and performance (M&P) analysis of Internet networks has taken on increasing importance to ensure network reliability, identify potential problems and ensure customer's contractual requirements are being met. First line responsibility for ensuring the network operates in the most efficient and effective manner lies with Network Operations, an area which has not fully been able to benefit from advances in network measurement instrumentation, in large part due to the overhead required to analyze the resulting large amounts of data. In certain cases network operators are fed pre-defined measurement reports which provide little or no value in dealing with real-time management of their networks.

A number of network statistics and performance measurement tools have come to be over the last few years. These tools, many emanating from the research community, have enabled network researchers, engineers, and operators to better understand their networks and the Internet. Specific examples of these tools are the OCxMon [1], the Multi Router Traffic Grapher [2], the Surveyor infrastructure [3], AMP [4], and Cflowd [5]. With some exceptions (an obvious one is MRTG), interpretation and analysis of collected data into its final form is usually done through post-analysis. This process might be automated to some extent, but this implies reliance on programming resources. While this is fine for researchers and design & planning engineers, it is far from ideal for network operators.

CANARIE Inc. is Canada's advanced Internet development organization. It was established in 1993 and has been working with government, industry, and the research and educational communities to enhance Canada's advanced Internet infrastructure, applications development and use. CANARIE and its CA*net3 program are supported by Industry Canada.

At the CA*net3 Advanced Research and Development Network Operations Center (ARDNOC), the problem of providing next generation NOCs with rapid measurement data analysis capabilities is an area of research focus. Although much work remains, an approach has been implemented.

At the base of the solution lies a common database management system (DBMS). Coupled with existing tools, modified where possible to provide data on regular short intervals, the aim of the work is to provide network operators with near real time "exception" reporting based on information acquired through the variety of measurement tools in service in the network, and based on criteria network operators themselves establish and modify as seen fit.

This paper will examine the problem of providing a flexible M&P analysis capability to a network operations environment. We will examine the approach taken by the ARDNOC as well as the architecture and implementation details of the solution.

Preliminary results will be presented in the form of an example using real CA*net3 cflowd measurements, followed by a discussion on future work.

II. PROBLEM

Today there exists a number of Internet statistics and measurement tools[6]. These tools come in different shapes and sizes and vary in the measurements they provide, in the mechanisms by which they acquire network data, by which they process the data, and by which they output and format results. In addition to the above, differences can exist with respect to the topological location of the measurement tool and the underlying lower layer encapsulation schemes.

For an instrumented network, especially a heterogenously instrumented one, these differences can significantly complicate the analysis of measurements, resulting in large delays between the time the measurements are taken and the time the measurement analysis is completed and available.

This makes for a poor NOC tool. The need to automate analysis as much as possible is critical for the efficient operation of a NOC. To accomplish this end, one can speed

things up through the use of scripts, but in a NOC environment this entails reliance on programming resources which are often external to the Operations Group. The construction of a script to do exactly what is required might also take several iterations as the operator attempts to get his/her exact requirement across to the programmer. More importantly, creating a new query, making changes to an old one, or modifying a report format will take time, and ultimately could discourage the use of the measurement system by the NOC.

An alternative to the custom script approach is to use a DBMS to store all measurement tool data, independent of the measurement tools specifics, and provide a common interface for creating queries and specifying report formats.

A first attempt at implementing this approach was undertaken at the CA*net3 NOC for the generation of CA*net3 weekly traffic reports[7]. A system using a small popular DBMS populated with Cflowd data records was set up and gave the ARDNOC the capability to quickly analyze, aggregate, reorganize, and present this data.

The real power of the approach, however lies in its ability to abstract the complexities of the underlying measurement infrastructure. A DBMS approach makes it relatively easy to reorganize data in any which way, correlate results of different tool origins, correlate results with network events, perform aggregation, maintain historical records, create and modify queries, and create and modify report formats.

The DBMS solution described above lacks certain features that are desireable for a NOC, namely real-time output, and real-time alert generation. The current design of the DBMS allows anomalous events, as defined by the operators, to trigger notifications in real or near real time.

III. Approach & Architecture

A DBMS is the central element of the CA*net3 network measurement system. The central database acts as a repository for all network measurement data, independent of measurement tool origin.

One of the difficulties involved with analyzing data from a diverse set of tools is that each tool output format is different. The measurement tools themselves are deployed across the network and store data, in either binary or text format.

The DBMS solution we describe below can be applied to any network measurement tool which produces a text file output. Presently the main tools in use in the CA*net3 NOC are Cflowd, MRTG and an OC3Mon.

Cflowd relies on Cisco's Netflow Export feature to provide inbound router interface flow information (for those router interfaces where the flow-switching feature is enabled). Using the Arts++ library, cflowd data is transformed into text files providing AS matrix information, network matrix information, protocol and application port information among others[8]. The text files are periodically generated through the use of scripts and the UNIX cron daemon.

MRTG runs on a separate UNIX host and stores SNMP get responses in log files. In the ARDNOC MRTG is used mainly for calculating 5-minute average traffic rates on a router interface (or sub-interface) basis. The log files are by default updated every 5 minutes.

The OC3Mon in use in CA*net3 runs on a DOS box and produces pre-defined statistics based on the AAL5 and TCP/IP header information that passes on the optical link it sniffs[9]. Again using cron, the results of the stats, in text format are downloaded to a UNIX box.

In order to populate the database, a generic scanner program is used to read measurement data text files. Access to these files is through NFS. A tool-output-format specific encapsulator reads the text file data fields and populates the database tables. A different encapsulator is required for each different data format.

Figure 1 below provides a high level view of the system using cflowd as the measurement data source.

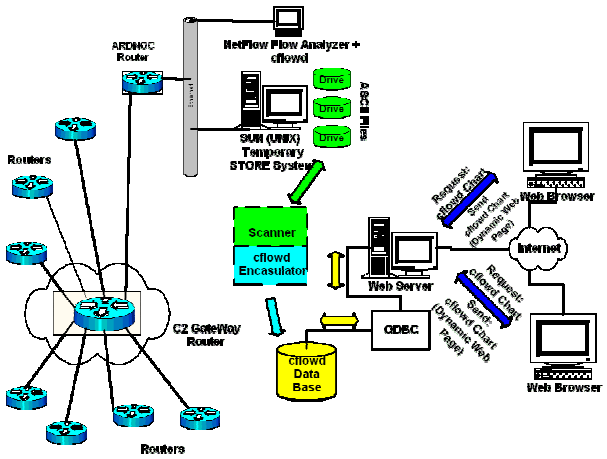


Figure 1: Measurement System using CFlowd

The scanner is a C++ object oriented based application. It contains many generic objects that deal with the general behavior of the scanners. It also contains specific objects that are dealing the specificity of the file to encapsulate. Thanks to the object oriented programming (OOP) inheritance principle, it is easy to develop and add new specific objects to the scanner in order to adapt it the a new type of files to scan.

The scanner/encapsulator can contain several applications (or jobs to perform). We can define the following parameters for each application defined:

- name,

- type (specific to the type of data collected, e.g. cflowd, mrtg, oc3mon, etc.),
- scanning period
- destination database to populate describing by the ODBC string (such as cflowd db, mrtg db, oc3mon db, etc.)
- filter to specify the file extensions to scan,
- directories of the source files to scan, and
- scanned files status list box.

Figure 2 shows the dB architectures. Two levels can be seen:

Top Level 1: One central database that represents the common and generic objects describing the networks related components; and

Bottom Level 2: Several databases where each one represents specific objects related to the monitoring technique used.

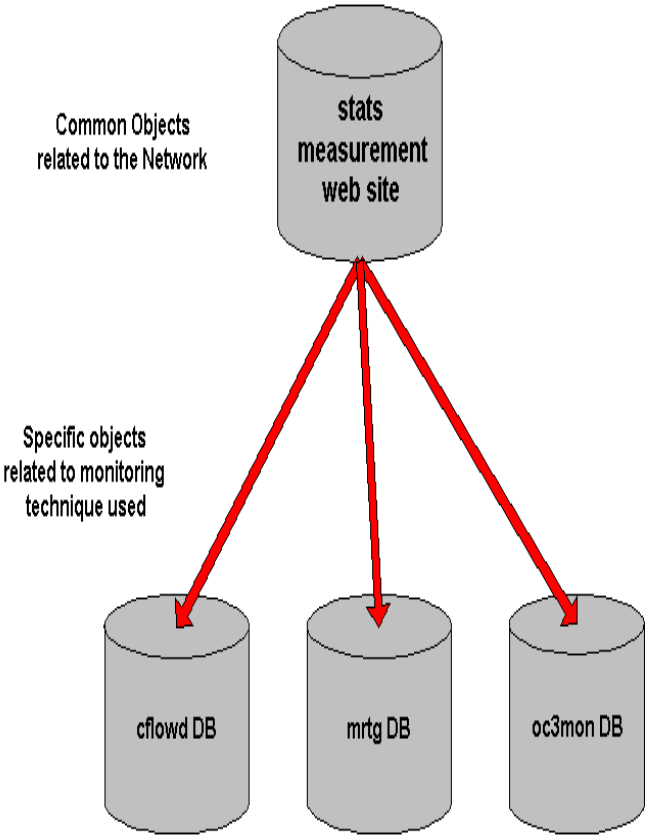


Figure 2: Database architecture

By linking the generic database's objects (peer/GigaPOP, router, etc.) to the specific database's objects (Interface number, Protocol, AS, Net, Nexthop, Count of Bytes, Bytes/second, etc.) one may easily query the data and get the useful results by key word common to a network operations environment.

Moreover, if one needs to display the data collected under other parameters (e.g. by network policy statements), all that is required is to add those new related objects to the generic database, to link them to the specific ones and to generate the appropriate output.

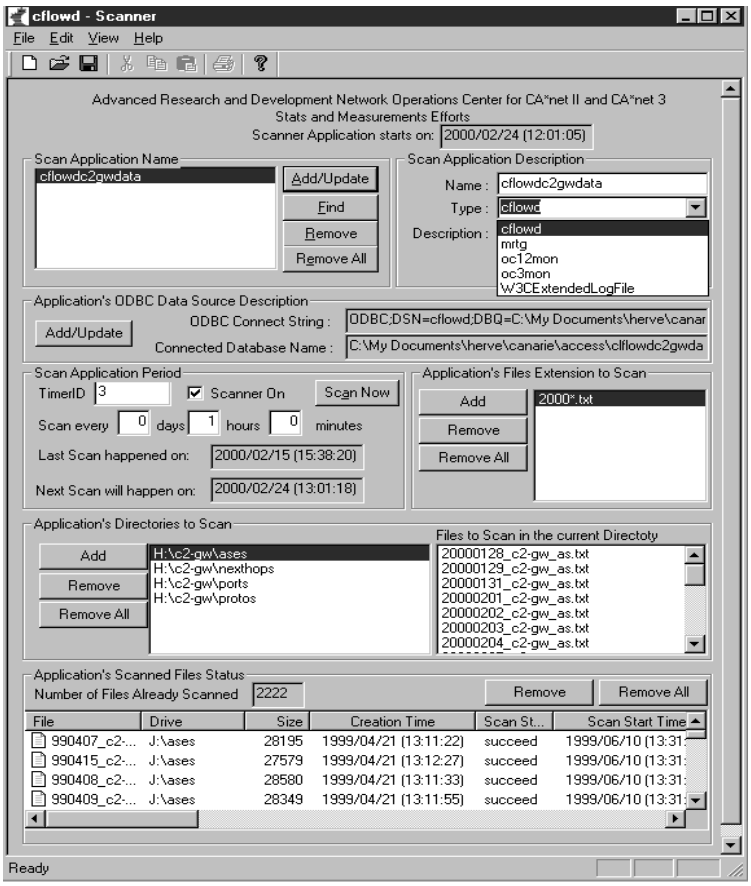


Figure 3: Scanner/encapsulator interface

For reasons of technological familiarity and cost, the database implementation chosen was MS SQL Server 7 running on four Intel Pentium Zeon 550MHz processors and 1 GB of RAM. The storage devices are four in number, can hold each up to 18 GByte of capacity and are controlled through an Ultra3 RAID controller.

In order to analyze and/or publish the collected measurement data, two approaches can be used.

The first approach is used for real-time analysis and alert generation (where primary clients are NOC operators).

In order to achieve real time analysis and alert generation, the trigger feature of the SQL server can be used. A Trigger is a special type of store procedure that is executed automatically as part of a data modification statement. A trigger is created on a table (e.g. cflowd net matrix object) and associated with one or more data modification actions such as the SQL INSERT (when new rows are added), UPDATE (when existing rows are modified) and DELETE (when existing rows are removed) statements.

Consequently, when the scanner scans a data file (e.g. cflowd net file) and encapsulates the data file into its related database table (e.g. cflowd net table) by inserting the record, the defined trigger is fired automatically. The trigger can run a specific store procedure that performs complex data integrity checks based on certain criteria predefined by the NOC operators. If those pre-defined criteria are met, the trigger may execute another job or store procedure in cascade, which will ultimately alert the NOC operator that something has met the specified criteria.

The second approach is to use common SQL statements to query the database and generate, on the fly, the appropriate analysis output report (where e.g. the primary information clients are company executives). The CA*net3 weekly traffic report generation process is an example of this sort of usage of a DBMS system[7].

IV. Example Result

Below we examine an example case of how the DBMS network measurement system can be used to identify network events.

Step one - Observation:

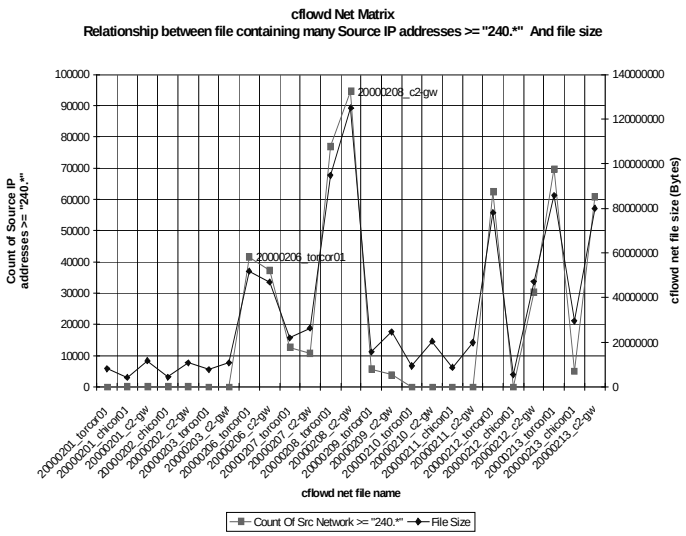
The cflowd net matrix output file size can be orders of magnitude greater than usual.

When we look inside the file we may find that the network matrix contains many entries with a source host addresses (/32) beginning with a 240 or greater (up to 255) prefix and where the number of Bytes involved is equal or less than 1500.

Step two - Correlation test with respect to the cflowd net file size and the strange behavior observed:

The following table and chart are the result of a query perform on the cflowd net table:

Cflowd Net File Name	Count Of Cflowd Source Network >= 240.*.*.*	Cflowd Net File Size	Cflowd Net File Size Order
20000208_c2-gw	94500	124895728	1
20000208_torcor01	77038	94682166	2
20000213_torcor01	69776	85749028	3
20000212_torcor01	62448	78085000	5
20000213_c2-gw	60976	79978842	4
20000206_torcor01	41645	51827418	6
20000206_c2-gw	37247	46927433	8
20000212_c2-gw	30416	47145070	7
20000207_torcor01	12631	22075152	12
20000207_c2-gw	10731	26439673	10
20000209_torcor01	5744	15739041	15
20000213_chicor01	5072	29433542	9
20000209_c2-gw	3981	24622996	11



Both confirm the strong relationship between the strange event observed and the size of the cflowd net file.

Step three - Determination of victim:

Cflowd Net File Name	Peer Name	Count Of Cflowd Source Network
20000206_torcor01	c2-c3 connection	41645
20000207_torcor01	c2-c3 connection	12631
20000208_torcor01	c2-c3 connection	77038
20000209_torcor01	c2-c3 connection	5743
20000212_torcor01	c2-c3 connection	30707
20000213_torcor01	c2-c3 connection	62602
20000206_c2-gw	Netera	37247
20000207_c2-gw	Netera	10731
20000208_c2-gw	Netera	94500
20000209_c2-gw	Netera	3980
20000212_c2-gw	Netera	30415
20000213_c2-gw	Netera	60976
20000212_torcor01	Onet	31741
20000213_torcor01	Onet	7174
20000213_chicor01	Startap	5072

Step four - Final investigation: To determine which destination network addresses are involved and victim of that strange behavior:

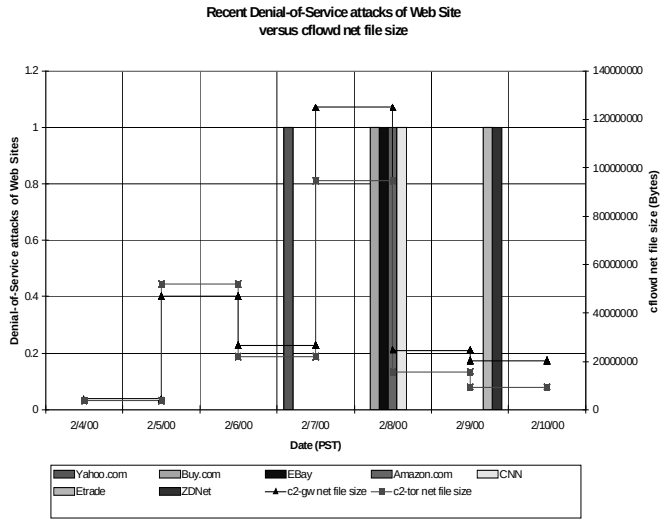
File Name	Peer Name	Destination Network	Count Of Cflowd Destination Network
20000208_c2-gw	netera	128.138.0.0/16	452
20000208_c2-gw	Netera	128.32.0.0/16	6348
20000208_c2-gw	Netera	130.241.0.0/16	21279
20000208_c2-gw	Netera	138.23.0.0/16	42124
20000208_c2-gw	Netera	152.16.0.0/16	3402
20000208_c2-gw	netera	207.58.0.0/18	20892
20000208_c2-gw	netera	207.58.63.1/32	3

Step five – Definition of the appropriate trigger based on criteria observed in steps one to four.

Therefore, a trigger may be defined that sends an alert to the network operations personnel via an e-mail message indicating that such and such Peer network with destination addresses may be targets of some strange probing ... or other activity.

Such an alert generation capabilty could be used for early detection of DoS attack waves such as the ones that occurred on February 7, 8 and 9, 2000 and disabled popular Web sites like Yahoo, eBay, Amazon, Buy.com, Etrade and ZDNet[10].

It is interesting to note that some degree of correlation between certain CA*net3 cflowd net matrix file sizes and the dates when those popular Web sites were disabled seemingly exists – see chart below.



Is it a coincidence or not? Actually we don't know… Further investigation is required to establish correlation.

Again this is just one example of how the DBMS can be used to spot network events and/or strange activity. Similar approaches can be used to spot SMURF attacks or UDP floods. Having the DBMS query the cflowd protocol table data for higher-than-normal levels of ICMP or UDP traffic entering an interface AND also querying the cflowd network matrix file for many spoofed source addresses to a same host into that same interface with two positive results could indicate a high probability of the occurence of a SMURF or UDP flood attack.

V. Summary of Ongoing and Future Work

In many instances it is difficulty to obtain real-time data through the use of certain tools, such as CFlowd for example. Presently, the ARDNOC relies on Cflowd information for reporting network usage. Work in progress is to modify Cflowd to roll over arts++ data files every hour, and then every half-hour, as first steps, instead of every 24 hours. Similar modifications may need to be performed on other tools to provide results in time frames appropriate for the desired degree of real-time responsiveness. Since attacks can last several hours, a half-hour alert capability may be acceptable.

Three other areas are targetted for future work. The tweaking of various system parameters in order to minimize the time it takes the scanner/encapsulator to read tool text output files and import the desired fields into the database, run the queries and generate an alert or report is important for near real time operation. This is an ongoing exercise.

Most Service Provider network operators work on a daily basis with BGP routing policy in order to balance traffic between links or perform other traffic engineering feats. ARDNOC ops have identified a need for network measurement data to be displayed by BGP router policy. This can be easily implemented with the current DBMS based measurement system.

Finally, a more in depth study of the scalability and sytstem performance limitations of this DBMS based measurement system is required.

VI. Conclusion

In order to enable network operators to be proactive in their roles, supporting network measurement systems must be made available that can quickly analyze network measurement data, and be able to correlate data originating from different tools and with known events.

Through an inductive development process, the ARDNOC has arrived at a DBMS based solution that provides rapid data analysis of measurements, provides the capability to correlate measurements obtained through different tools and in different formats (i.e. is tool independent), and provides for easy mechanisms for end users (e.g. network operators) to customize queries (pattern searches), reports and reporting intervals.

VII. Acknowledgements

The authors would like to thank Bill St-Arnaud for his encouragement and guidance.

VIII. References

1. vBNS Engineering, OCxMon Overview, http://www.vbns.net/stats/flows/html/overview.html
2. http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
3. Advanced Network & Services, Surveyor Home Page, http://www.advanced.org/surveyor/
4. NLANR MOAT, Introduction to the NLANR AMP Project for HPC sites, http://watt.nlanr.net/AMP/
5. Dwm/kc, cflowd, http://www.caida.org/Tools/Cflowd/
6. CAIDA Measurement Tool Taxonomy, http://www.caida.org/Tools/taxonomy.html
7. http://www.canet3.net/nettraffic/Weeklyreports.html
8. Daniel W. McRobb, cflowd design, http://www.caida.org/Tools/Cflowd/design/design.html
9. CAIDA, DOS OC3Mon raw data description http://www.caida.org/Tools/Coral/flowoutput.html
10. James Niccolai, IDG News Service, 02/11/00, NetworkWorldFusion Web magazine, http://www.nwfusion.com/news/2000/0211hacker.html?nf