

M2C* Deliverable D1.6

M2C Measurement Tools

Remco van de Meent

`r.vandemeent@utwente.nl`

University of Twente, Enschede
The Netherlands

Abstract

This report gives an overview of the tools developed and/or used within the M2C project. These tools focus on measurement, anonymization and analysis of network traffic, and can be downloaded from <http://m2c-a.cs.utwente.nl/tools/>.

**Measuring, Modelling and Cost Allocation (M2C)* is a project sponsored by the Dutch Telematica Instituut. Partners in this project are the University of Twente (through its CTIT research institute) in Enschede, and the National Research Institute for Mathematics and Computer Science (CWI) in Amsterdam.

1 Introduction

This document describes the tools related to network measurement that have been adapted and developed as part of the M2C project. We first discuss tools related to metering of network traffic, followed by tools for analysis of the gathered data.

Note that, generally, these tools do not result in graphical output, but mere numbers instead. These numbers can be processed by, e.g., *gnuplot* or *Excel*, to create graphs.

The tools that are developed within the M2C project can be downloaded from <http://m2c-a.cs.utwente.nl/tools/>.

2 Metering Tools

In this report we only include a brief description of the metering tools. For information on how these tools have been used in the measurements within the M2C project, see, e.g., [1].

2.1 Traffic Capturing

The tool that has been used for capturing network traffic is the popular *tcpdump* [2] tool. This software can be downloaded from the Internet for free. It captures network traffic, optionally filtering and pretty-printing the data, per the user's request. *Tcpdump* can also be used to write the captured traffic to disc, resulting in *packet traces*, for off-line processing, .

In the M2C project, the GNU/Linux operating system is used for all measurement efforts. One thing that has to be kept in mind, is that GNU/Linux by default imposes a limit of 2 gigabyte on file-sizes. Software that has been compiled with the proper "large file support" does not suffer from this limitation.

2.2 Anonymization

In order to protect the privacy of clients using the network, all packet traces are made anonymous. Various tools exist for anonymization of network traffic; within M2C, *tcpdpriv* [3] is used.

3 Analysis Tools

The tools discussed in this section are all developed within the M2C project¹.

3.1 Throughput on arbitrary time cales

Software like MRTG [4] typically shows the traffic throughput based on 5 minute average rates. Thus, what happens *within* an interval of 5 minutes remains unclear.

Using the packet traces that are collected using tcpdump, which are timestamped with ≤ 10 milliseconds accuracy, and custom built software, we can determine the throughput rates on (more or less) arbitrary time scales. This has been used in the studies presented in [5, 6].

Source: `http://m2c-a.cs.utwente.nl/tools/throughput-alltimescales.pl`

Requires: Perl and Net::Pcap module (available from CPAN)

Usage: `throughput-alltimescales.pl <dumpfile> <dt_sec> <dt_microsec>`

In the following example, the script is used to determine the average throughput rate per 0.5 second. The output format is as follows: first time since start of dump in seconds and microseconds, the number of packets in the preceeding interval, and number of octets sent in the preceeding interval. Note that if the actual throughput rate in bits per second is desired, this has to be calculated seperately (i.e., number of bytes * 8 / interval size).

```
throughput-alltimescales.pl somedumpfile 0 500000
0 500000    334    194600
1      0    238    141132
1 500000    242    157268
2      0    236    143709
2 500000    235    133969
3      0    132     61798
3 500000    121     60594
4      0    199     87675
...
```

¹*Copyright information:* All tools are Copyright (C) Remco van de Meent, University of Twente `r.vandemeent@utwente.nl`. All the tools can be freely distributed and used according to the terms of the GNU General Public License, either version 2 or (at your opinion) any newer version. NO WARRANTY. A full copy of the GNU GPL can be found at <http://www.gnu.org/licenses/licenses.html>

3.2 Flow extraction

The packet traces contain detailed information on single IP packets. Although packets are treated (more or less) independent from each other on the Internet, one can abstract from single packets, and instead look at the *flow* level. We have applied this concept in research presented in [7, 8].

A flow is defined as a group of packets that have certain properties in common. For example, a single flow may consist of all traffic flowing in or out a specific network. Another definition is sometimes referred to as the *5-tuple flow definition*, which means that packets having the following properties in common, are said to belong to the same flow:

- source IP address (e.g., 130.89.12.101)
- destination IP address (e.g., 130.89.1.16)
- protocol number (e.g., 6, for TCP)
- source transport port number (e.g., 1025)
- destination port number (e.g., 80)

We consider flows to be bidirectional, which means that packet in the “other” direction have source and destination addresses and port numbers swapped, still belong to the same flow. With the example numbers above, this means that all packets belonging to the same HTTP-request from a work station to the University of Twente’s web-server, are in the same flow.

To determine the start and end of a single flow, various approaches are possible, e.g., i) look at the transport protocol headers for connection establishment and closure (works for TCP, but not for UDP) or ii) use timeouts: start of a flow is the first packet, and the end is the last packet that is sent (with these properties) followed by an idle period, e.g., 20 seconds. We choose the latter approach in this study; the results have been shown to be similar.

A tool has been written to extract flows, consisting of one or more (possibly millions) packets, from the packet traces. The output of this tool is a list of information about flows in a packet trace, containing information like: time of first and last packet in the flow, total amount of bytes and packets, port numbers, etc.

Source: `http://m2c-a.cs.utwente.nl/tools/flow-extract.c`

Requires: GLib (see <http://www.gtk.org/>) (≥ 2.0) and libpcap (comes with tcpdump [2])

Usage: `flow-extract -f <dumpfile> -o /dev/null [-t <flowtimeout>]`

Example output:

```
key: 36750669323322059084060008004328
src ip: db0d1234 [hex]
dst ip: c602994c [hex]
proto: 6
src_port: 80
dst_port: 4328
bytes_in: 3317
bytes_out: 103088
bytes_total: 106405
pkts_in: 43
pkts_out: 77
ts_first: 1060959602 107166 [sec usec]
ts_last: 1060959603 643147 [sec usec]
duration: 1 535981 [sec usec]
```

Although this output format is “suitable” for human reading, a less versatile format is handy for post-processing by computer programs. For that purpose, the following script converts the output into one-line descriptions: <http://m2c-a.cs.utwente.nl/tools/flow-extract-convert.pl>.

3.3 Visualization Tools

In order to avoid duplicate work, and to create a flexible architecture for visualizing network traffic characteristics, the Visualization Tools package has been developed. See [7] for an extensive description and manual. A live version of the package is accessible at <http://m2c-a.cs.utwente.nl/bsc-visual/>. The Visualization Tools operate on flows, as described above. All access is via a web-interface, and the output of the package are graphs.

Source: <http://m2c-a.cs.utwente.nl/tools/visualizationtools.tar.gz>

Requires: Apache, MySQL, PHP

Usage: Unpack, edit config.php, access via web

The following traffic characteristics are currently available:

- 10 second average throughput rates, including peak rates within the 10 second intervals
- traffic rates of large (“elephants”) compared to small (“mice”) flows

- variance in throughput across various time scales (burstiness)
- flow arrival, duration and concurrency characteristics

The Visualization Tools internally access a MySQL database containing information about the flows. In order to store the flow information generated by the flow-extract(-convert) tool in the MySQL database, a small utility has been developed (see also [8]):

Source: <http://m2c-a.cs.utwente.nl/tools/flowtosql.tar.gz>

Requires: Java, MySQL

Usage: Unpack, edit `src/flowtosql/FlowToSql.java`, compile, run (see `execute.bat`)

3.4 Analysis of Unknown Traffic

One can recognize a trend that a growing fraction of all Internet traffic cannot be identified to be generated by a specific application: *unknown traffic*. This has motivated the development of algorithms, and implementations of those algorithms in a tool, which tries to identify unknown traffic by relating unknown traffic flows to known traffic flows. See [8] for an extensive description of the algorithms and the tool.

This tool, just like the Visualization Tools, operates on flows and internally uses the MySQL database with flow information. The output of the tool consists of HTML files, giving indications of the applications that are used.

Source: <http://m2c-a.cs.utwente.nl/tools/flowanalyzer.tar.gz>

Requires: Java, MySQL

Usage: Unpack, then see `compile.bat` and [8] for further instructions

4 Conclusions

Within the M2C project we have developed a number of tools in order to get a better understanding of network traffic. These tools help to analyse large quantities of data, and can be reused in other measurement environments as well. The tools are freely available, under the GNU GPL.

References

- [1] R. van de Meent, “M2C Measurement Data Repository,” tech. rep., University of Twente, December 2003. M2C Project Deliverable 1.5.
- [2] Lawrence Berkeley National Laboratory Network Research, “TCPDump: the Protocol Packet Capture and Dumper Program,” 2003. <http://www.tcpdump.org/>.
- [3] Ipsilon Networks, “tcpdpriv,” 1997. <http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html>.
- [4] T. Oetiker, “MRTG: Multi Router Traffic Grapher,” 2003. <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>.
- [5] R. van de Meent, A. Pras, M. Mandjes, H. van den Berg, and L. Nieuwenhuis, “Traffic Measurements for Link Dimensioning: A Case Study,” in *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM2003)* (M. Brunner and A. Keller, eds.), no. 2867 in Lecture Notes in Computer Science (LNCS), pp. 106–117, October 2003.
- [6] R. van de Meent, A. Pras, M. Mandjes, H. van den Berg, F. Roijers, P. Venemans, L. Nieuwenhuis, F. Roijers, and P. Venemans, “Burstiness predictions based on rough network traffic measurements.” Submitted.
- [7] R. van de Meent and A. Pras, “Visualization Tools,” tech. rep., University of Twente, November 2003. M2C Project Deliverable 1.3.
- [8] R. van de Meent and A. Pras, “Assessing Unknown Network Traffic,” tech. rep., University of Twente, November 2003. M2C Project Deliverable 1.4.