CAIDA CoralReef Exercises

Timestamps –more than meets the eye

(Unix Help Version)

Level

Introductory.

Prerequisites

1. an ability to perform simple tasks using Unix
2. introductory networking concepts including a notion of packets and cells

System Resources

1. gnuplot
2. CoralReef or WWW access

Timestamps

Conceptually a CoralReef trace is a series of records. Each record describes a cell or a packet recorded from a communications link. Each record contains a timestamp that records the time at which this cell was captured by the monitor. This exercises examines these timestamps.

Preparation

Obtain or locate the following traces:

- `ODU-926843183.crl.enc.gz`
- `ACK-dag-19990708-121553-0-160000-161000.crl`
- `ACK-dag-19990708-121553-1-160000-161000.crl`
- another coral trace of your choice

A large number of "packet header traces", containing the first 48 bytes[1] of each IP packet are available at http://moat.nlanr.net/Traces/ In addition there are some traces that have been selected specifically for these exercises. Your instructor may have made these available locally. They are also available at: http://www.caida.org/t̃onym/traces/

Typically the traces include data from two interfaces, one collecting data from each direction.

The names of these traces consist of a three letter code, a Unix timestamp, and an extension indicating the format of the trace. The three letter code identifies the location of the monitor. For example `ODU-926843183.crl.enc` refers to the Old Dominion University vBNS link collected at 20:26 on Sunday May 16 1999.

If you need to convert the Unix timestamp to a date and time try `date -r time_value` or

```
perl -e 'print scalar(localtime(time_value));'
```

Note that these commands will give date and time in *your* local time zone.

The Waikato University WAND group also collects traces using their DAG hardware. These are named differently to the NLANR traces. The DAG traces start with a three letter code identifying the trace, followed by `-dag-` identifying them as DAG traces, followed by the date and time of the trace, followed by the interface.

[1] Why 48 bytes?

For example ACK-dag-19990708-121553-0-160000-161000.crl was collected at 12:15 on Jul 8 1999 on interface 0 of the monitor.

1 Printing timestamps

The coral application `crl_timestamps` prints timestamp information about each cell. The three columns of its output are:

1. The interface from which this cell or packet was recorded.[2]
2. The timestamp indicating the cell was recorded.
3. The difference between the current timestamp and the timestamp for the previous cell on this interface.

1. Run `crl_timestamps` on the ODU trace and examine the output to confirm that it is working correctly and that you understand its output.

   You might like to redirect the output to `more` (by adding `|more`) to the end of the command. This takes the output of `crl_timestamps` and processes it with the command `more` which allow you to page through it a page at a time.

2. Count the number of cells and work out the average number of cells per second.

   To count the number of cells redirect the output to the `wc -l` command.

[2] Most monitors that record data are connected passively to the communications channel that they are recording. For example a monitor recording cells from an OC12 ATM link may be connected to the link using a fiber splitter which takes a small percentage of the light energy and redirects it to an ATM interface on the monitor. In this way the monitor can record the cells it sees. Each interface is able to record one stream of data. Most connections are full duplex so two interfaces are needed the record the data for both directions.

2 A closer look

Some difficulties arise when working with traces because of their size. Some of these difficulties are related to the time it takes to process the trace. This is especially true if interpreted languages like perl are used (which is the case with `crl_timestamps`).

A more significant problem is the difficulty of verifying that the data collected is reasonable. It is valuable to develop strategies for making some simple validity checks on the data. In the case of timestamps we know they should always increase in value and under without any sudden changes.

One way to verify this is to plot the timestamp value against the cell number. To do this you need to:

3. Extract just the timestamp from the output of `crl_timestamps`. Awk can be used to do this. For example redirect the output through `awk ' print $2 '` The result must be put into a file so that they can be plotted. Use `> filename` to redirect output to a file. That is run something like:
   `crl_tmestamps ODU-... | awk ... > filename`
4. Plot the result. Either use your favorite plotting program or `gnuplot.`[3].

[3]Gnuplot is a simple to use plotting program from the GNU project. Grace (also known as xmgr) is an alternative which some people prefer.

Gnuplot plots data in a file. In the simplest case the data should be lines with pairs of xvalue yvalue separated by a space or a comer. If there is only one value per line it is assumed that the x values start at 0 and increase by 1 each line.

To plot data start `gnuplot` and use a command line: `plot "filename" with lines` Don't forget the quotes around the file name. If you prefer you can specify `with points` or `with linespoints`. Extra lines can be added by including more filenames (with or without a `with` clause) separated by a comer. For example:
`plot "foo", "bar" with linespoints`

To print a guplot graph set the output type to the type of printer with the set printer command and send the output to a file with the set output command. For example:

```
set terminal postscript color
set output "ts.ps"
```

There is a lot more gnuplot can do for you if you are adventurous. Try the online help.

You will notice that there is a discontinuity near the start of the trace. At about cell 1300 the timestamps return to 0 and start again. Most of the NLANR traces have this discontinuity near the start.

To understand this behaviour you need to know a little about the way the monitor is constructed. Most monitors have two cards ATM interface cards, one to monitor each direction of the link. The cards are operated by firmware which is downloaded to the card at the beginning of a measurement. Once this firmware is loaded the card starts recording data. It is desirable to have the timestamps as close as possible on the two cards so once both are operational the monitor system resets both cards timestamps at the same time, effectively synchronising them.

Because of this behaviour any analysis that uses timestamps should discard the early part of the trace. The average cell rate you calculated earlier will be wrong.

5. edit the timestamp file you created to remove the cells before the clock reset and recalculate the cell rate. Redirect input from the file to the `wc` command. That is, do something like: `wc -l < filename`.

While the difference between the correct and incorrect calculations is not large in this case it should be clear that similar hidden behaviour in cell traces could have a significant effect on the analysis.

3 Another trace

6. Repeat the timestamp graph with the `ACK-dag-19990708-121553-0-160000-161000.crl` trace.

The timestamps in the ACK traces are derived from an external real time reference (the global positioning system (GPS) is used to provide an accurate time reference).

E.G. `help plot`

There are a number of 0 timestamps at the end of the trace because the raw file format used for CoralReef is based around blocks of or cells but the DAG monitor is not. A partly filled block of cells is padded with zeros by the conversion software.

7. remove the zero timestamps from the end of the file and re plot the ACK trace.

4 Inter-cell time distribution

Researchers are often interesting in the typical time between packets or the distribution of packet inter-arrival times. For example the designers of a high speed switch might want to know the shortest and most common inter-packet times and the percentages of packets at those times so that they can design a switch that can switch at the appropriate number of packets per second.

8. Plot a graph of inter-arrival time against frequency of inter-arrival time. You will need perform the following steps:
   (a) select the cell time delta from the output of `crl_timestamps` (probably using `awk` or `cut -f`) Note that because these traces only include the first cell of each packet the cell interarrival time is the same as the packet interarrival time.
   (b) sort the deltas into numeric order (using `sort -n`)
   (c) count the number of occurrences of each value (using `uniq -c`)
   (d) swap the columns so that the count (the x value on the plot) is first. (use `awk '{ print $2 " " $1 }'`
   (e) Put the result into a file and plot.

   You can do all this in a single command, piping the output of each stage to the next stage. Your command will look something like: `crl_timestamps ACK-... | awk ... | sort -n | uniq -c | awk ... > filename`

   If you are using gnuplot you might like to experiment with different plot styles including `with impulses`.

9. What are the most common and the smallest inter-cell times in the `ACK-dag-19990708-121553-0-160000-161000.crl` trace? **Note:**The smallest inter-cell time is *not 0*. You will need to zoom in on parts of the graph to discover the full structure. If you are using `gnuplot` use `set xrange [min:max]` e.g. `set xrange [0:1.0]` and/or `setscale`.

5 Combining traces

Because of the external time source, which synchronises both interface cards in a DAG monitor, the cells in the traces collected for the two directions can be combined and, if sorted on timestamp, the cells will be correctly ordered between the two traces. This is not the case for traces collected where the two cards are independent because the separate clocks in the cards will drift slightly over time and cells that are *reported* in one order might have actually occurred in a different order.

This ordering problem makes some kinds of analysis impossible. For example with correctly ordered cells retransmission can be detected when data is seen *after* it has been acknowledged. If the ordering of cells can not be relied on this analysis will not be meaningful.

If two or more traces are listed on the `crl_timestamps` command line (and most other coral applications) coral will merge the data from the two files as the data is processed so that the cells or packets are processed in order.

10. Rerun `crl_timestamps` with both ACK traces as parameters. Note the interface number now varies and the timestamp deltas are relative to the interface of the cell, not the immediately preceding cell if that was from the other interface.

6 Conclusion

The concept of timestamps and card interfaces is simple but practical considerations make timestamps more difficult to deal with. For example the large size of traces requires care to ensure that the methodology accounts for all the events in the trace, not just the most common events.