Trace Processing for HTTP

The program `http_connect` is used to performs an analysis of `tcpdump` output (the ASCII print lines that `tcpdump` generates to `stdout`) and produces a summary of the TCP connections used for HTTP. It assumes that the `tcpdump` has been filtered for packets that are from TCP source port 80 and that the result has been sorted so that packets are in ascending time order within each TCP connection. The script given below will properly prepare the input for this program given a `tcpdump` binary file that may contain more than just HTTP packets (the file extensions are just examples, the program does not make any assumptions about input file names). Note that this filtering produces a UNI-DIRECTIONAL trace containing only those TCP segments sent from the server to the client.

```
#! /bin/sh
tcpdump -n -tt -r $1 tcp src port 80 > $1.http-srv
sort -s -o $1.http-srv-sort +1 -2 +3 -4 +0 -1 -T /tmp $1.http-srv
```

The sorted file can then be processed by the program `http_connect.` This program takes two parameters:

```
  -w file_name        // name for output file
  -r file_name        // name for input file
```

To get usage information, invoke the program with the -h switch.

The output from `http_connect` contains a summary of each HTTP connection in the trace showing the connection initiation, termination, and an entry for each HTTP protocol element: request (REQ) and response (RSP). The integer value following the protocol element type gives the size in bytes of the element. Here is an example that shows a browser using a mix of persistent and non-persistent HTTP connections to request pages from two web servers (a complete description of the output file format is given in the document `http_output_formats.doc`).

```
Timestamp        HTTP client         HTTP server      Type        Size    Timestamp
924200316.592942 5.2.4.133    1046 > 9.3.2.47     80: SYN
924200316.709325 5.2.4.133    1046 > 9.3.2.47     80: REQ          604  924200316.905758
924200318.123634 5.2.4.133    1046 > 9.3.2.47     80: RSP         3108  924200317.346400
924200318.242234 5.2.4.133    1046 > 9.3.2.47     80: FIN               924200318.226707
924200321.413123 5.2.4.133    1047 > 9.3.2.47     80: SYN
924200321.413123 5.2.4.133    1047 > 9.3.2.47     80: TRM               924200321.413123
924200318.173291 5.2.4.133    1048 > 9.3.2.47     80: SYN
924200318.202930 5.2.4.133    1048 > 9.3.2.47     80: REQ          334  924200318.202930
924200318.454963 5.2.4.133    1048 > 9.3.2.47     80: RSP         3238  924200318.202930
924200318.482482 5.2.4.133    1048 > 9.3.2.47     80: REQ          335  924200318.482482
924200319.448150 5.2.4.133    1048 > 9.3.2.47     80: RSP         2750  924200318.482482
924200333.437189 5.2.4.133    1048 > 9.3.2.47     80: FIN               924200333.329735
924200318.173288 5.2.4.133    1049 > 9.3.2.47     80: SYN
924200321.413332 5.2.4.133    1049 > 9.3.2.47     80: REQ          335  924200321.413332
924200321.417483 5.2.4.133    1049 > 9.3.2.47     80: RSP          765  924200321.416469
924200321.434347 5.2.4.133    1049 > 9.3.2.47     80: RST               924200321.434347
924200318.245839 5.2.4.133    1050 > 9.3.2.47     80: SYN
924200318.268149 5.2.4.133    1050 > 9.3.2.47     80: REQ          334  924200318.268149
924200318.305050 5.2.4.133    1050 > 9.3.2.47     80: RSP         2518  924200318.268149
924200318.332915 5.2.4.133    1050 > 9.3.2.47     80: REQ          334  924200318.332915
924200318.560996 5.2.4.133    1050 > 9.3.2.47     80: RSP         2811  924200318.332915
924200332.791114 5.2.4.133    1050 > 9.3.2.47     80: REQ          621  924200332.974109
924200333.342198 5.2.4.133    1050 > 9.3.2.47     80: RSP         3272  924200333.293000
924200333.950575 5.2.4.133    1050 > 9.3.2.47     80: FIN               924200333.891748
924200333.442474 5.2.4.133    1051 > 9.3.2.47     80: SYN
924200333.501751 5.2.4.133    1051 > 9.3.2.47     80: REQ          334  924200333.501751
924200334.395215 5.2.4.133    1051 > 9.3.2.47     80: RSP         3774  924200333.760722
924200348.214294 5.2.4.133    1051 > 9.3.2.47     80: FIN               924200348.214294
924200333.444071 5.2.4.133    1052 > 9.3.2.47     80: SYN
924200333.681838 5.2.4.133    1052 > 9.3.2.47     80: REQ          334  924200333.681838
924200333.907291 5.2.4.133    1052 > 9.3.2.47     80: RSP         2181  924200333.762646
924200348.215111 5.2.4.133    1052 > 9.3.2.47     80: FIN               924200348.215111
```

```
924200350.723585 5.2.4.133    1054 > 4.7.1.71     80: SYN
924200350.759647 5.2.4.133    1054 > 4.7.1.71     80: REQ          283  924200350.759647
924200351.054604 5.2.4.133    1054 > 4.7.1.71     80: RSP        46096  924200350.759647
924200351.990267 5.2.4.133    1054 > 4.7.1.71     80: TRM               924200351.990267
```

http_active program

The program http_active is used to create an activity trace (summary form) of web browsing clients with respect to three types of activity: client sending request data, server sending response data, client is idle (no request or response). Identification of idle periods is used to infer user “think” times between requests for new top-level pages. A client is defined by a single IP address.

"Idle" is defined as a period of time greater than a threshold value ("idle_limit" with a default of 2000 milliseconds) during which a client has no requests outstanding. A request is outstanding from the start time of a request until the end time (normal or terminated) of the corresponding response. The input to this program is the SORTed output from http_connect. The sort to be applied is produced with the following shell script:

```
   sort -s -o $1.sort +1 -2 +0 -1 -T /tmp $1
```

This sorts all the records for a given client IP address in timestamp order.

The sorted file can then be processed by the program http_active. This program takes three parameters:

```
  -w file_name        // name for output file
  -r file_name        // name for input file
  -I idle_limit       // threshold time value (ms) to distinguish idle periods
```

To get usage information, invoke the program with the -h switch.

The output is also time-ordered with respect to a single client (IP address). It consists only of client request entries (in the same format as the input) ordered by start time, server responses (in the same format as the input) ordered by end time, and client idle periods (giving the elapsed idle time) ordered by the end of the idle period. Here is the same example given above as output from http_active (a complete description of the output file format is given in the document http_output_formats.doc).

```
Timestamp        HTTP client         HTTP server      Type         Size  Timestamp
924200316.709325 5.2.4.133    1046 > 9.3.2.47     80: REQ          604  924200316.905758
924200318.123634 5.2.4.133    1046 > 9.3.2.47     80: RSP         3108  924200317.346400
924200318.202930 5.2.4.133    1048 > 9.3.2.47     80: REQ          334  924200318.202930
924200318.268149 5.2.4.133    1050 > 9.3.2.47     80: REQ          334  924200318.268149
924200318.305050 5.2.4.133    1050 > 9.3.2.47     80: RSP         2518  924200318.268149
924200318.332915 5.2.4.133    1050 > 9.3.2.47     80: REQ          334  924200318.332915
924200318.454963 5.2.4.133    1048 > 9.3.2.47     80: RSP         3238  924200318.202930
924200318.482482 5.2.4.133    1048 > 9.3.2.47     80: REQ          335  924200318.482482
924200318.560996 5.2.4.133    1050 > 9.3.2.47     80: RSP         2811  924200318.332915
924200319.448150 5.2.4.133    1048 > 9.3.2.47     80: RSP         2750  924200318.482482
924200321.413332 5.2.4.133    1049 > 9.3.2.47     80: REQ          335  924200321.413332
924200321.417483 5.2.4.133    1049 > 9.3.2.47     80: RSP          765  924200321.416469
924200332.791114 5.2.4.133       * > *              * IDLE       11373  924200321.417483
924200332.791114 5.2.4.133    1050 > 9.3.2.47     80: REQ          621  924200332.974109
924200333.342198 5.2.4.133    1050 > 9.3.2.47     80: RSP         3272  924200333.293000
924200333.501751 5.2.4.133    1051 > 9.3.2.47     80: REQ          334  924200333.501751
924200333.681838 5.2.4.133    1052 > 9.3.2.47     80: REQ          334  924200333.681838
924200333.907291 5.2.4.133    1052 > 9.3.2.47     80: RSP         2181  924200333.762646
924200334.395215 5.2.4.133    1051 > 9.3.2.47     80: RSP         3774  924200333.760722
924200350.759647 5.2.4.133       * > *              * IDLE       16364  924200334.395215
924200350.759647 5.2.4.133    1054 > 4.7.1.71     80: REQ          283  924200350.759647
924200351.054604 5.2.4.133    1054 > 4.7.1.71     80: RSP        46096  924200350.759647
```