CS439 – Wireless Networks and Mobile Computing
Spring 2015
Project on Network Measurements and Analysis
Professor: Maria Papadopouli

Part A

The purpose of this assignment is to capture network traffic using tcpdump or wireshark and the then analyze it with matlab.

At you PC, generate various types of traffic (e.g., HTTP, torrent, video streaming, VOIP calls) for at least an hour and capture it using wireshark or tcpdump. Then, for each protocol, save trace files with all exchanged packets and import them in matlab.

Helpful tips on how to import data collected from wireshark into matlab:

1. Apply filters to separate the packets of each protocol.
2. Save the data in a .txt file (file-->export-->as plain text file). From the emerging window select "packet summary" and "all packets".
3. Save the file into your working folder in matlab.
4. Open matlab and execute the following commands:

```
fid=fopen('filename.txt');
C = textscan(fid, '%d %f %s %s %s %d %*[^\n]', 'headerLines', 1);
fclose(fid);
```

The argument %*[^\n] indicates to matlab to ignore everything that exists after the sixth column in each line.

The cell array C contains six attributes. Select the ones that contain the timestamp and length of packets.

Part B

Analyze the traces collected in Part A in order to obtain a better understanding of the user demand in your network.

Produce a bar plot that indicates the total number of bytes transferred by each protocol (e.g., HTTP, TCP, and UDP) during your experiment. Tip: use the matlab command "bar" to construct a bar plot.

Plot the cumulative distribution function (CDF) of the packet size for each protocol. Report the mean, median, and standard deviation of the packet size for each protocol. Comment on the results. Tip: to construct a cdf plot use the

command “ecdf” of matlab, while to estimate the mean, median, and standard deviation, use the commands “mean”, “median”, and “std”, respectively.

Part C

The file “Packet_header.txt” contains flow-level packet header traces (see Appendix 1 for more detailed information).

For each of the Access Points 167, 183, 91, 143, and 469, plot the CDF of the flow duration (sixth column in file “packet_headers.txt”).

For each of the Access Points 167, 183, 91, 143, and 469, plot the CDF of the flow size in bytes (fifth column in file “packet_headers.txt”).

Plot the flow-size in bytes versus the duration of each flow. Is there a correlation (Appendix II)? Explain.

What is the mean, median, and standard deviation of flow size for each of the Access points 167, 183, 91, 143, and 469? Is the traffic load heavy or light in general? Which AP has the lightest and which AP has the heaviest traffic load?

The file “http.txt” contains monitored HTTP traffic (see Appendix 1 for more detailed infromation).

Consider a 5-minute time scale. Plot the number of HTTP Requests versus time for each of the categories specified in the sixth field of the HTTP Requests trace.

Useful tips: Suppose that time instances t1 and t2 indicate the start and end of the experiment and d is the size of the considered time intervals (e.g., 5 minutes). If you save the timestamps of HTTP requests in a vector “x” and execute the command “y = ceil((x - t1)/d)”, you will get a vector “y”. This vector indicates the ID of the interval in which each request was sent. Then, you can use the command “accumarray” and the vector “y” in order to compute the total number of requests that were performed in each interval. Read the documentation of “accumarray” for more information.

Produce a bar plot that shows the total number of HTTP Requests for the ten more frequent categories that appear in the sixth field of the HTTP trace. Comment on the plot. Is there a category that dominates the others? What types of sites do the clients browse more?

IMPORTANT NOTE: You must submit all the MATLAB code you wrote for the trace analysis with sufficient comments.

Appendix I

Packet header traces

The file entries includes the following fields:

1. Timestamp of the first packet of the flow.
2. ID of wireless client that initiated the flow.
3. ID of the Access point to which the wireless client was connected.
4. Total number of packets transferred during the flow.
5. Total number of bytes transferred during the flow.
6. Duration of the flow (estimated by measuring first to last packet arrival times).
7. Duration of the flow (estimated on the basis of WLAN-originated packets, still measuring first to last packet time arrivals, only now the "last" packet is the last one with a useful payload).
8. Type of TCP termination:
    a. Not_Closed: no termination packets (FIN or RST) observed in any direction.
    b. LAN_Closed: termination packets only observed from the LAN.
    c. WAN_Closed: termination packets only observed from the WAN.
    d. Complete: termination packets observed in both directions.
9. Port number of the LAN side.
10. Port number of the WAN side.

HTTP requests

Each line represents an HTTP request found in a packet captured by a server monitoring traffic in and out of the wireless network. HTTP requests were extracted after processing the original packet headers traces

HTTP user request logs look like this:

```
1044559890 972962 mdO+1om0AIt2x28foCcTA1qkNA4Q
hFJXMOBElyLJ0/JDx6/J+JnYvWV8 pfifPZXra3Z0Tpj657m6w6Lnsbmg

1044559895 915535 mS9mdo76HsLlW4YrxT9WzajaLU4g
hu7q19RyZWOFlNUfHfN0woGy9bOU pNWlImhuPtc+qVkvT0vwvQPANLU0
News/Breaking_News/
```

Each line contains the following columns:

1. Time in seconds since the Unix epoch that the packet was captured by the server.
2. Number of microseconds that had elapsed since the beginning of that second.

3. MAC address of the wireless device that made the request always starting with an 'm'.
4. Host as drawn from the HTTP header always starting with an 'h'.
5. Requested path always starting with a 'p'.

   Recall that a standard HTTP/1.1 request may look like:

   ```
   GET/index.jsp HTTP/1.1
   Host: www.foo.com
   ```

   For this case, the path is "/index.jsp" and the host would be "www.foo.com".

   All of these are SHA-1 hashed of the actual data.

6. The sixth field, if provided, identifies a category in the DMOZ database that closely matches the site specified. If the sixth field is provided, then the seventh field will identify how closely the request matches the DMOZ entry. (The DMOZ was downloaded in approximately April, 2003.)

Appendix II

Notes on cross-correlation between two signals

The cross-correlation between two sampled signals is defined as:

$$R_{xy}(m) = \frac{1}{N}\sum_{n=1}^{N-m+1} y(n)x(n+m-1)$$

$m = 1, \dots, N$ and N is the number of samples

In matlab you can use the function xcorr(x,y,'coeff') in order to estimate the cross-correlation between x and y. Using the option 'coeff' you get the cross-correlation normalized between 0 and 1, where 0 means that the signals are uncorrelated.

Example (MATLAB)

X= randn(100,1); % generate the signal X

Y= sin([1:100]); % generate the signal Y

[c lags] = xcorr(X,Y,'coeff'); %compute the cross-correlation

plot(lags,c) % plot the cross-correlation