Virginia Tech ■ ECE/CS 4570: Wireless Networks and Mobile Systems ■ Spring 2006
In-Class Laboratory Exercise 13 (L13)

Part I – Objectives and Laboratory Materials

Objectives:

The objectives of this laboratory are to:

- ❑ provide an experience with the design concept of broadcast-based data services to mobile users
- ❑ improve understanding of the performance and scalability characteristics of broadcast-based data services versus traditional client-server data services; and
- ❑ introduce the implementation of broadcast-based data services.

After completing the lab exercise, you should be able to:

- ❑ compare and contrast broadcast-based data services vs. traditional client-server data services;
- ❑ understand how to use UDP to implement broadcast-based and client-server data services;
- ❑ evaluate the response time performance metric of broadcast-based data services versus traditional client-server data services.

Hardware to be used in this laboratory exercise:

- ❑ Dell notebook computer with IEEE 802.11b card (with a fully charged battery)
- ❑ iPAQ with an IEEE 802.11b card and cradle (with a fully charged battery)
- ❑ Intel Wireless Gateway (one for each group)

Software to be used in this laboratory exercise:

- ❑ Notebook, iPAQ/Pocket PC, PocketSoap v1.5
- ❑ Microsoft Visual Studio .NET

Part II – Pre-laboratory Assignment

This portion of the assignment must be completed prior to the in-class laboratory session.

Reading Assignment:

- ❑ Browse through the following article on UDP programming:
  - o Using UDP Services:

    http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconusingudpservices.asp

Tasks:

- ❑ Install PocketSoap on your iPAQ, using the steps below, if it is not yet installed. You will use PocketSoap to access a web service running on your notebook computer and evaluate the response time metric for this traditional client-server data service.
  - o Your notebook should be pre-installed with PocketSoap. To install PocketSoap on your iPAQ, go to ActiveSyn->Tools-Add/Remove Programs and select PocketSoap v1.5 and then hit the "OK" button.

- ❑ Install the broadcast-based data service demonstration software using the following procedure.
  - o Download the server code (implemented in Visual Studio .NET C#) "**server.zip**" from the class web page into your notebook's c:\wnms\labs\lab_11 folder and unzip it. Source code for both a broadcast-based data service and a client-server data service (web service) is included.

  Note: In the in-class laboratory session, you will experiment with a broadcast-based data service demonstration that includes two parts: (i) a data dissemination server that broadcasts data items (stock prices) via a wireless channel; and (ii) a client-side application that listens and retrieves desired data items. The server will broadcast UDP packets via port 8888 while the client application will listen to data broadcast on this specific UDP port. The client application also contains a simple UI to allow a user to input a stock symbol and subsequently display the stock price broadcast from the server.

- ❑ Configure the traditional client-server data service StockQuoteService on your notebook.
  - o Go to **Start->Control Panel->Administrative Tools->Internet Information Services.** Under the Internet Information Services, click on the machine name (wnms*xx* where *xx* is your group number) and then click "+" to show the default Web Site. Click on **Default Web Site** on the right panel, and right click and select "New" to create a "Virtual Directory." After following "Next," enter "StockQuoteService" in the Virtual Directory Alias. Select a directory, c:\wnms\labs\lab_11\Server\wsquoter. Test the web service by entering URL, http://localhost/StockQuoteService/service1.asmx.

- ❑ Install the client software for measuring response time
  - o Download the client-side application "**client.zip**" from the class web page into your notebook's c:\wnms\labs\lab_11 folder and unzip it. Copy the application "PPCClient" to your PocketPC under "Program Files".

Part III – In-class Laboratory Assignment

1. **Experiment with broadcast-based data services in wireless environments via the demonstration software**
   - ❑ Set up a WLAN in infrastructure mode
     - o Each group will set up its own Intel Wireless Gateway in this laboratory session with the following configuration.
       - ▪ SSID = **WNMS*xx*** where ***xx*** is your group number
       - ▪ WEP enabled with key = **ABCDEF4570**
     - o Setup your iPAQ and notebook IEEE 802.11b network interface cards in infrastructure mode.
       - ▪ Configure the IP address of the wireless interface on your notebook computer manually as "192.0.2. 116/255.255.255.0."
       - ▪ Configure the IP address of your iPAQ as "192.0.2.166/255.255.255.0."
       - ▪ Verify the network connection by pinging from your notebook to your iPAQ and vice versa.

- ❑ Run the data dissemination server software “bcaster.exe” located under the c:\wnms\labs\lab_11\Server\bcaster\Release folder.
  - o To emulate a broadcast environment in which only one broadcast server exists, run the data dissemination server on your notebook during the in-class laboratory session
  - o Examine the server code “Class1.cs” in folder c:\wnms\labs\lab_11\Server\bcaster on your notebook computer and understand at the code level how the server provides broadcast-based data services as follows.
    - ▪ The total number of data items that will be broadcast by the server periodically is 1000. Each data item is formatted as “C### PRICE” where C### is the stock symbol and PRICE is the stock price.
    - ▪ All data items are broadcast in a “flat” organization.
    - ▪ The server broadcasts data items using UDP and broadcast with each UDP packet containing 100 data items organized in a back-to-back manner using the “|” character as the separator. Each UDP packet size is approximately 1K bytes in length.
    - ▪ The broadcast channel bandwidth is approximately 80 Kbps. To not overload the broadcast channel with UDP packets, the time interval between which UDP packets are sent is calculated as 1K*8/80k = 0.1s.
    - ▪ The server uses 255.255.255.255 as the broadcast address. For cases in which you want to multicast, a UdpClient needs to join a multicast group. Read the reading material “Using UDP Services” for information about the range of multicast group address to be used.
- ❑ Run the client-side application on your iPAQ
  - o The client-side code is implemented with eMbedded Visual C++. You are not given the source code because you are not required to modify the client-side code in the in-class laboratory assignment. (***Note:*** You will be asked to modify the server-side code in C# in E14.) On the client user interface screen (see Figure 1), button “Quote” is used to get a quote for a particular stock symbol (C###), button “Client/Server” is used to measure the average response time for the traditional client/server data service, button “Broadcast Data” is used to measure the average response time for in-class broadcast data services (flat), buttons “Hot Data” and “Cold Data” are used to measure the response time for hot data and cold data (broadcast disk), respectively. You only need to use the last two buttons in E14.
  - o Start the application PPCClient on your iPAQ. After verifying the client-side application functions correctly on your iPAQ, note the stock prices returned for the following stock symbols:
    - ▪ C234
    - ▪ C125
    - ▪ C640

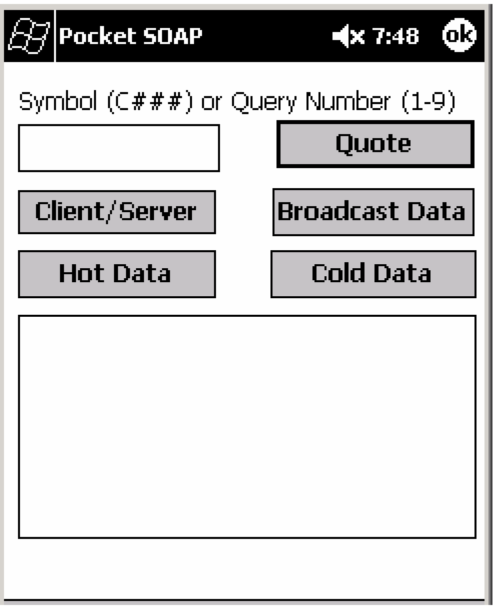

Figure 1. Client user interface screen.

2. **Measure the response time metric for the broadcast-based data service demonstration application**.
   - Run PPCClient on your iPAQ again. Then input *N* in the range of 1 to 9, inclusive, into the edit-box. The application will randomly choose *N* stock symbols and tune to the broadcast channel to retrieve their respective prices, thus emulating *N* clients listening to the broadcast channel simultaneously. It will then report the average response time of *N* queries in the list-box. The emulation time will last 5 seconds.
   - Record the average response times observed in the table below. The data will be compared with those from the traditional on-demand client-server data service application in which a client will send a query to the server on-demand and then wait for a reply sent back from the server. (Try three times and get the grand average of them.)
   - Terminate the broadcast program once you collect the average response time data.
   - Draw a conclusion in terms of the sensitivity of the response time with respect to the number of clients (*N*) simultaneously accessing the broadcast-based data service based on the response time values collected.

# of clients (N)	1	2	3	4	5	6	7	8	9
Grand average response time (ms)									

3. **Measure the response time metric for the traditional client-server data service demonstration application**.

   To avoid interference among groups, each group should set up its own Intel Wireless Gateway with SSID = **WNMS*xx*** and WEP key = **ABCDEF4570**, where ***xx*** is your group number. Configure the IP address of your notebook manually as 192.0.2.116/255.255.255.0.

For traditional on-demand client-server data services, we will use the PPCClient application again to measure the average response time as a function of the number of simultaneous queries, *N*.

- Make sure your iPAQ is connected to the on-demand stock quote server "StockQuoteService" running on your notebook at IP address 192.0.2.116 by pinging the server.
- Run PPCClient on your iPAQ. Then input *N* in the range of 1 through 9, inclusive, into the edit-box. The application will create *N* threads to emulate *N* clients. Each client will randomly generate a valid symbol, send the symbol to the stock quote server, and receive a price returned from the server. This process will run for about 5 seconds. Then the average response time per request will be calculated and displayed in the list-box. Try three times and calculate the grand average of them to improve the accuracy. If any average response time output is found to be abnormally higher than others, you can discard it as an outlier (perhaps due to short-lived interference) without using it for the grand average calculation.
- Record the average response times observed in the table below. The data will be compared with those from the broadcast-based client-server data service application.
- Draw a conclusion in terms of the sensitivity of the response time with respect to the number of clients (*N*) simultaneously accessing the broadcast-based data service based on the response time measurement data collected.

# of clients (N)	1	2	3	4	5	6	7	8	9
Grand average response time (ms)									