A system for the detection and reporting of wireless modem signals

Ecir Uğur Küçüksille*, Tuncay Yiğit, Ali Güneş and Güncel Sariman

Department of Computer Engineering, Suleyman Demirel University, Isparta, Turkey.

Accepted 25 May, 2010

This study focuses on the detection of wireless network modem signals in the environment using C# language in Visual Studio. Net 2008 editor, obtaining data like the MAC(Media Access Control) address, SSID(Service Set IDentifier), canal used, signal quality level, type of authorization and type of encoding used in detected wireless networks and representative mapping of detected wireless network signals. An application developed here in for this purpose utilizes Native Wifi, an open source code API designed and owned by Microsoft.

Key words: Wireless networks, native wifi, service set identifier, media access control, wireless modem signals.

INTRODUCTION

Local Area Networks (LAN) are networks encompassing many personal computers (PC) connected in a restricted geographical area such as a building, school, hospital, and campus (Erkinay, 2005). In LANs, in case RF (Radio Frequency) or infrared technology is used in place of cables to establish communication between computers and other devices in the network, it is named as Wireless Local Area Networks (WLAN) (Öztürk, 2004).

Wireless technology, in its simplest sense, means the communication of one or several devices with no physical connectivity. Wireless networks; is a flexible communication system applied as an alternative to wired communication and which conducts information exchange through air by using RF technology (Duranoglu, 2004).

Wireless communication technology is a technology with which connection is provided point to point or by a network environment (Duranoglu, 2004). Today, this technology is being used in many places from centers such as company offices, school campuses, airports, train stations or bus terminals to cafes and had a wide usage area. Air is being used as transmission medium in wireless communication technology (Kavas, 2007). As in radio telephones and radios, the communication in wireless communication is provided with radio signals. Wireless communication, also named as WIFI (Wireless Fidelity), is very similar to radio communication. Radio signal is transferred in both situations and these signals are turned into numbers format in a binary system. However, WLAN systems have some significant difference from other systems. WLAN systems use 2.4 GHz or 5 GHz signals. These signal levels are quite higher that the signal level is used by devices such as radio telephones.

This shows us that more data can be transferred via WLAN systems. Distance of WLAN systems is about 25 - 100 m (Öztürk, 2004). There are two common types of WLAN technologies in the world: One of them is American based IEEE 802.11x and 12 and the other is European based HyperLAN system. Current main standard used for establishing a wireless network is IEEE 802.11 (Kavas, 2007) (Table 1).

Detection of wireless network signals in the environment using C# language in Visual Studio.Net 2008 editor, obtaining data like the MAC (Media Access Control) address, SSID (Service Set IDentifier), canal used, signal quality level, type of authorization and type of encoding used in detected wireless networks and representative mapping of detected wireless network signals have been conducted with the application that has been developed.

*Corresponding author. E-mail: ecir@sdu.edu.tr. Tel: +902462111126. Fax: 246 211 1147.
for these purposes.

Wireless network detection and management applications

NetStumbler is one of the application that detects wireless points (wireless connection points), determines signal level and transfer the relevant analysis to the visual interface in a detailed manner. Besides, it conducts functions such as point detection with disconnections, signal power weakening and GPS (Global Positioning System) and measuring signal quality level. It can work with 802.11b, 802.11a and 802.11g (Netstumbler, 2007).

AirSnare program is a network program which detects unauthorized access to wireless network environment. The alarm steps in when an undefined MAC address is entered or secret DHCP (Dynamic Host Configuration Protocol) queries are done. After potential threats are determined, the relevant IP addresses and used ports can be monitored (AirSnare, 2005).

WirelessMon is an application that allows WIFI connection information in the environment. It presents status and locations of real-time wireless connections or other wireless networks in a real-time manner. WirelessMon also keeps the logs in the connection in a separate file and shows the graphics and levels of connection signal together with real-time IPs and also 802.11 WIFI statistics. It allows 802.11 network configurations to be defined accurately. It provides the signal control of WIFI network level control (WirelessMon, 1999).

MATERIALS AND METHODS

Developed wireless network application

This wireless network application has been developed with Visual Studio.Net 2008 editor using C# language. The Native Wifi API contains functions, structures, and enumerations that support wireless network connectivity and wireless profile management. Native WIFI API, which has been published by Microsoft as open source, has been used for determining the signals in this working environment, obtaining the qualities of detected network devices, connecting to network environment and disconnecting the connection and also a Interop class has been used for controlling wireless network card of the computer. The developed application basically consists of two parts. First part includes the functions such as listing MAC addresses of the network devices together with features such as SSID, the channel that is used, signal quality level, authorization type and the encoding type that is used by means of signals determined in the environment under the control of a listview, providing connection for these network devices and disconnecting them and building up an XML (Extensible Markup Language) profile for the connected network shows the XML profile about connected to a wireless network in (Figure 1). In second part, graphical demonstration of wireless network devices has been conducted on a representative mapping together with its SSID names.

Wireless network detection and reporting

The first procedure while compiling the developed application is the detection of wireless network devices and listing them under listview control together with all their features. These features are the MAC address, SSID, the channel that is used, signal quality, authorization type, encoding type, standard of the network that is being used and speed level. In addition to those quality level of the signal is shown in this part.

The most important device for scanning wireless network signals in the environment is Timer control. Scanning time can be changed by setting this control to the desired time. One of the detected wireless modems is chosen and the network is connected with entering the relevant password after scanning procedure is completed. Firstly a profile in XML format is created for the wireless modem that is chosen to be used in WlanAPI class for this connection procedure. XML tags of this profile consist of device features such as MAC, SSID, signal level and encoding type that belongs to chosen wireless modem. After the modem profile is created for the modem to be connected, this profile is sent as parameter to SetProfile() method of WlanAPI class. SetProfile method of WlanAPI class is as follows:

Public Wlan. Wlan Reason Code Set profile (Wlan. Wlan profile Flags, String ProfileXml, bool overwrite)
(Wlan. Wlan reason Code reason Code;
(Wlan. ThrowlfErro)
return reasonCode;

After that connect () method of WlanAPI class is called with required parameters and connection is made with wireless modem. Connect () method of WlanAPI class is as follows.

Public void Connect (Wlan. WlanConnectionMode Connection Mode, Wlan.Dot11Bsstype, stringprofile).
Figure 1. XML profile for connected wireless network.

```csharp
[Wlan. WlanConnectionParameters Connectionparams = new Wlan.
WlanConnectionParameter();
connectionParams. WlanConnectionMode = ConnectionMode;
connectionParams. profile= profile;
connectionParams.dot11BssType = bsstype;
connectionParams.flags = 0;
Connect(connectionparams);
}

After connection is established with the wireless modem that has
been chosen, all the features of the device such as MAC address,
SSID, the channel that is used, signal level, authorization type and
encoding type are written to textbox and label controls in the
interface of developed application and shown to the user. Range of
visuals in signal level indicator is only considered as an estimate.
The Connect Method () and WlanState() with which these
procedures are done is as follows:

private void connect()
WlanClient.WlanInterfaceWlaniface = Client.interfaces[0]; if
(Wlaniface.interfaceState = Wlan.WlanInterfaceState.Connected II
wlanface.interfaceState==WlaninterfaceState.AdHocNetworkForme
d)
textBox1.Text = wlanface.CurrentConnection.profileName;
textBox4.Text = wlanface.CurrentConnection.wlanAssociationAttributes.wlanSignal
Quality + "%";
textBox5.Text = ... +
wlanface.CurrentConnection.wlanSecurityAttributes.securityEnabled;
textBox6.Text = wlanface.CurrentConnection.wlanSecurityAttributes.dot11A
uthAlgorithm.TosString();
textBox7.Text = wlanface.CurrentConnection.wlanSecurityAttributes.dot11CipherAl
orithm.TosString();
int signalquality =
Covert.ToInt32(wlanface.CurrentConnection.wlanAssociationAttribu
tes.wlanSignalQuality);
if (signalquality > 79)
pictureBox2.Image = Properties.Resources._5;
if (signalquality > 59 && signalquality < 79)
pictureBox2.Image = Properties.Resources._4;
if (signalquality > 39 && signalquality < 59)
pictureBox2.Image = Properties.Resources._3;
if (signalquality > 19 && signalquality < 39)
pictureBox2.Image = Properties.Resources._2;
if (signalquality > 0 && signalquality < 19)
pictureBox2.Image = Properties.Resources._1;
WlanState();
private void WlanState()
WlanClient.WlanInterface wlaniface = Clinet.Interfaces[0]; if
(wlaniface.InterfaceState = Wlan.WlanInterfaceState.Connected II
wlanface.InterfaceState==WlaninterfaceState.AdHocNetworkForme
d)
WlanClient.WlanInterface wlaniface = Client.interfaces[0]; if
(wlanface.InterfaceState = Wlan.WlanInterfaceState.Connected II
wlanface.InterfaceState==WlaninterfaceState.AdHocNetworkForme
d)
Switch (wlanface.CurrentConnection.isState)
case Wlan.WlanInterfaceState.AdHocNetworkFormed:
label11.Text = 'Now connecting' +
wlanface.CurrentConnection.profileName; break;
case Wlan.WlanInterfaceState.Authenticating:
label11.Text = "Authenticating..."; break;
case Wlan.WlanInterfaceState.Connected:
label11.Text ="Connected to " +
```
quality level and used encoding type of connected network has been written on textbox controls on the developed form by means of objects derived from WlanClient class. Besides, signal quality level has been visualized by means of picture box control on the form. Five (5) different pictures have been used for network signal level. Quality indicator pictures of picture box control have been changed by comparing the detected signal level with estimated values. Signal quality over 79 is excellent among the values estimated for the visualization of signal quality and values between 0 and 19 are considered very low shown in (Figure 2).

**Representative mapping**

Graphical demonstration of wireless modems has been made on a representative mapping in the second part of the developed application, together with their SSID names. Wireless Sensor Network simulators have been used for this mapping system. All of the detected wireless modems are shown on the mapping in a circular manner and SSID names are written on the circles that are drawn. The method that conducted the mapping procedure is as follows:

```csharp
Private Void mapaccesspoint(){
    int signicount = 0, lines = 0;
    signicount = listview2.items.count;
    if (signicount < 6)
        lines = 420;
    if (signicount > 6 & signicount < 10)
        lines = 350;
    if (signicount > 10)
        lines = 200;
    network = new WirelesssensorNetwork(signicount, 25 10, lines, 10,
        (float)50/100.0f, 15, 10, false,
        0.0, 0.1, picNetwork. Width - 5, picNetwork. Height - 5, picNetwork.
        Width - 45);
    iSetupDisplay = 0;}
```

**Experimental results**

Modems in the environment have been detected with the application that has been developed and listed as it is shown in (Figure 3). Besides, features of the connected network have also been listed in the project and signal data have been visualized. Representative drawing of the detected modems on the mapping is shown in (Figure 4). Mapping procedure is done on picture box control. Wireless Sensor Network has been used for the placement of the wireless modems on the mapping. Objects belonging to Drawing class have been used as well as objects derived from the classes in this simulator.
RESULTS

Wireless network signals in the environment have been successfully detected and all the detected network devices have been listed under a list view control together with their features in the first part of the study that has been conducted. Detected wireless networks have been visualized on a representative mapping in the second part. Mapping procedure is done in a representative manner in this study. GPS technology may be added to the study that will be developed for real-time mapping. It was observed at the end of the experimental studies that the application which has been developed had success-fully detected the signals in the environment and all data relevant to detected wireless networks have been obtained.

REFERENCES


Durangoğlu Ö (2004). Kablosuz Ağlar; HomeRF, Bluetooth Teknolojileri. Department of Computer Engineering, Faculty of Engineering and Architecture, Beykent University, Turkey.


