SWAP: Software Wireless Adapter Program for Multiple Connections

Gyan Bahadur, Amit Agarwal

Department of computers science Engineering Sharda University, Greater Noida, Uttar Pradesh, India

Summary
The normal way to use one wireless card is connecting to one wireless network. It is reduces the flexibility and the efficiency. There are no wires; wireless cards could connect to more than one wireless network. It is possible only by using multiple wireless network cards in the device. This is the purpose of my project. This is develop an application capable of communicating with several access points and connecting to their wireless networks. The software wireless adapter program for multiple connections is software for wireless network interface card. Wireless cards can connect to multiple wireless networks by organizing the configuration in the software wireless adapter. Software Wireless Adapters allows a hardware computing device to connect to multiple wireless networks using inbuilt wireless card minimize the need for purchasing other wireless card. This project can enhance the power of home infrastructure network by extending its access to nodes that are out of range of wireless device. The project is introduced by a new functionality. This software wireless adapter program earlier using a single wireless inbuilt card. This is a software based approach, called SWAP which is connections to multiple networks by virtualizing a single wireless card. This virtualizing is intermediate layer below IP, which switches the card across multiple networks.

Keyword:
windows XP, Wi-Fi Adapter.

1. Introduction

Software Wireless Adapters program for Multiple Connections is a software feature for wireless network interface cards (WNIC). Wireless cards can connect to multiple wireless networks by organizing the configurations in Software Wireless Adapters. Software Wireless Adapters allows a Hardware Computing Device to connect to multiple wireless networks using inbuilt wireless card minimizing the need for purchasing other wireless cards. The new functionality introduced by Software Wireless Adapters enables many new applications, which were not possible earlier using a single wireless inbuilt card.

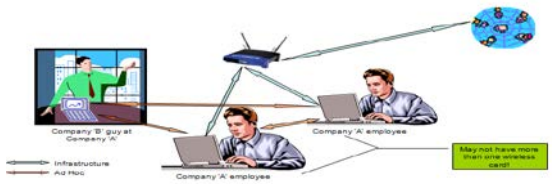

Fig1 Simple Scenario of SWAP

Software Wireless Adapters provides a feature of connecting a Local Area Network to Play Counter Strike over a Local Wireless network, while connecting web applications or websites via another infrastructure network. Software Wireless Adapters provides the feature for connecting local network, which may contain many nodal systems, to the Internet using only one node. Software Wireless Adapters can enhance the power of home infrastructure network by extending its access to nodes that are out of range of your home Wireless Device.

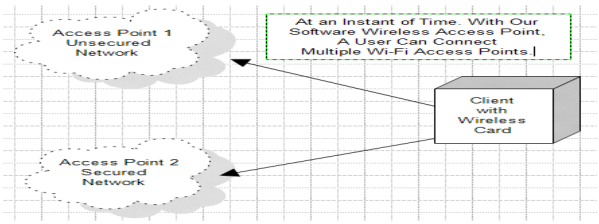

Fig2 multiple access point connected with single wireless card

This diagram clears the purpose of the Project. We are removing the limitation of Multiple Wireless Access Point Connection with a Single Wireless Client Side Card. This is an important project, which can remove a lot of Connection Problems in the Domain of Wireless Connections.
There are several benefits of virtualization. To the best of my knowledge, the benefits of vitalizing a wireless card have been overlooked. In this research/project we propose SWAP, a new virtualization architecture that abstracts a

Manuscript received May 5, 2014
Manuscript revised May 20, 2014

single WLAN card to appear as multiple virtual cards to be simultaneously connected to physically different wireless network. It abstracts a single WLAN card to appear as multiple virtual WLAN cards to the user. The user can then configure each virtual card to connect to a different wireless network. SWAP allows a user to simultaneously connect his machine to multiple wireless networks using one WLAN card. This new functionality introduced by SWAP enables many new applications, which were not possible earlier using a single WLAN card. Software Wireless Adapters enable clients to diagnose the main problems in wireless connectivity. Software Wireless adapters will be used by clients in order to setup an information surface. All clients then exchange wireless configuration information over this information surface and use this information to diagnose the issues of various wireless failures.

2. Our Approaches

2.1 Virtualization

This diagram illustrates the SWAP virtualization of a wireless card as an abstraction of multiple wireless networks as different always active virtual adapters over a single WLAN card. It should be possible for a user to change individual parameters of each virtual adapter, and these always active adapters should also be able to send and receive packets at any time. SWAP achieves this by multiplexing the wireless card across multiple networks. It uses an adaptive network hopping scheme where a card gets a time slot, called the Activity Period, for each network operating on a particular channel. The sum of the activity periods over all the connected networks is called the Switching Cycle. The virtualization of wireless adapters is implemented by the SWAP protocol driver that is an intermediate layer, between IP and the MAC. This driver exposes the wireless LAN media adapter as multiple always active virtual wireless LAN media adapters, one per desired network. The IP stack sees all the adapters as always active even though at the driver level only one is active (connected) at any given time. Other than exposing virtual adapters, the SWAP Protocol Driver is also responsible for switching the wireless cards across the different networks, and buffering packets for networks that are currently inactive.
The figure shows the virtualization of a WLAN card when a user wants to connect to 3 wireless networks. The SWAP Protocol Driver exposes 3 virtual adapters, and all of them appear active to IP although only Network 2 is active at the instant shown in the figure.

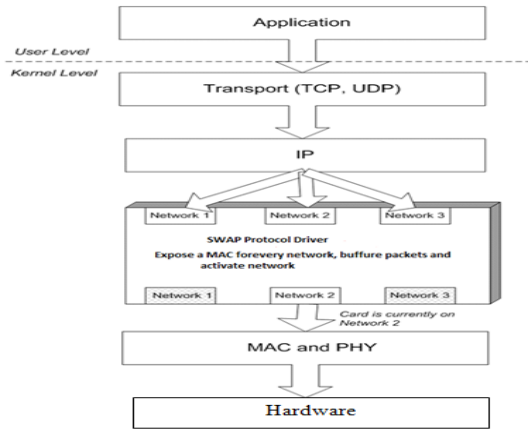

Fig3 Microsoft network deriver for NDIS

2.2 WEP (Wired equivalent privacy)

Wired Equivalent Privacy (WEP) is defined as the purpose for a security protocol for wireless networks. This is encrypting transmitted data without any security. Our data can be intercepted without difficulty.WEP has three settings: Off (no security), 64-bit (weak security), 128-bit (a bit better security)

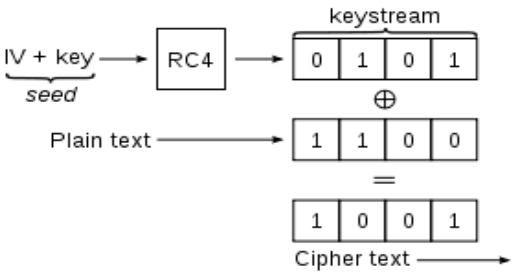

Fig4 Basic WEP Encryption

This is not difficult to crack. It is reduces the performance slightly. If you run a network with only the default security, the WEP is turned off, and this is a very weak security any of your neighbors can immediately log on to your network and use your Internet connection. For wireless devices to communicate, all of them must use the same WEP setting. While there is no extra performance cost to encrypting the longer key, there is a cost to transmitting the extra data over the network. 128-bit security is not much more difficult than 64-bit to crack, so if you are concerned about performance, consider using 64-bit.

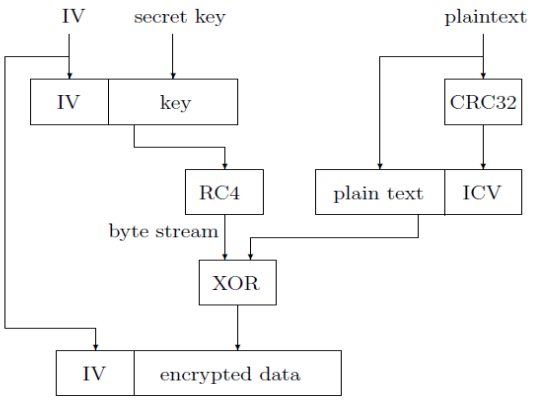

Fig5 WEP Encryption diagram

The encryption algorithm is based in the RC4 algorithm, which generates a byte stream from the final encryption key. This stream is then XORed with the clear data to be encrypted concatenated with an Integrity Check Value (ICV), generating thereby the encrypted data. The process is represented in Figure

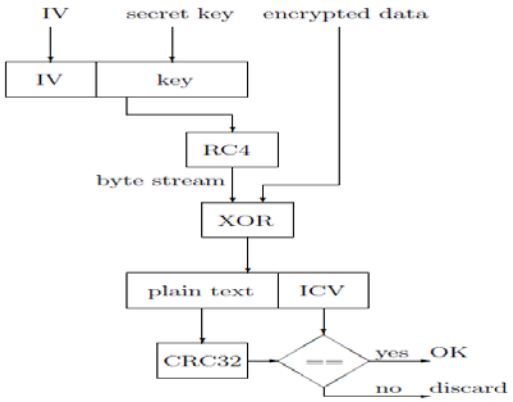

Fig6 WEP decryption diagram

The decryption algorithm is the inverse algorithm. With the final encryption key, the same byte stream, as in the encryption process, is generated with the RC4 algorithm as well. This stream is then XORed with the encrypted data. This gives us the plain text and the ICV. We can verify the integrity of the data recalculating the ICV and comparing it with the received one.

3. Methodology And Planning

Write a code for the driver. Driver program starts by registering to an NDIS as an intermediate driver. Microsoft offers a miniport adapter. We then register miniport with NDIS. Miniport is not a protocol. At this part miniport will act like a driver. We need to register the miniport before the BindAdapter handler is initialized. We will specify send handlers and send packet handlers. If all goes well we will register the protocol. The name of the protocol (SWAP) and the service for it will have the same name. This will help NDIS to find accurate binding and it will send the binding call to bind miniport. We have to register an IOCTL interface. The routine for IOCTL registration will be done all the times, whenever a new miniport instance is initialized.

a. **SWAP Executable**: Software Wireless Adapter Program Install, Uninstall & Connect program.

b. **SWAP Service**: Windows Service to automate the startup process for SWAP. It will have three possible modes: Enabled, Disabled & Automatic.

c. **SWAP Performance Monitor**: This program monitors the performance of the Software Wireless Adapter. This may correct some faults in network communications. This can be implemented by using some fault detection algorithms.

d. **SWAP Console**: The console will act like a command line command centre program. Developers or command line users can operate SWAP from anywhere in the command line.

e. **SWAP Registry Management Console**: This module will monitor all the necessary changes to be done inside windows registry to perform proper functions. This will handle creation of registry keys, management of registry keys & deletion of registry keys for SWAP.

f. **SWAP Driver**: We need to program for the driver which can register with NDIS as an intermediate driver. We need to consider system driver object structure. We can do this using an object for driver. The driver will have to manage the registry entries as well. A return message with a Success Flag or Error Code Flag should be returned on successful or failure at initialization.

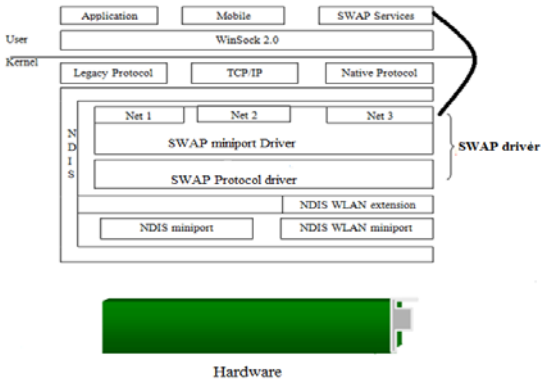

Fig7 Windows network stack

4. Implementation

This diagram shows that and we have implemented SWAP on windows XP. The windows provide a network driver interface specification (NDIS) as an intermediate layer between the network device drivers and IP. NDIS provides transport independence for the network card since all the upper layer protocols call the NDIS interface to access the network. We implement the SWAP as a combination of an NDIS intermediate driver and services. The service has the buffering and switching logic, its passes instruction to the driver, which implements the mechanics for this buffering and switching. We require all the wireless nodes to implements SWAP. However, no change is requiring in the wired node for SWAP works.

4.1 SWAP Driver

To the windows NDIS interface we added an intermediate driver, which we refer to as the SWAP driver. NDIS requires the lower edge of the network protocols driver to bind to a network miniport driver, and the upper edge of miniport drivers to bind to a protocol driver. A miniport driver directly manages a network interface card (NIC) and provides an interface to higher-level drivers. Therefore the SWAP driver comprises two components: the SWAP protocol driver that binds at the lower edge to the network card miniport driver, and the SWAP miniport driver that binds at the upper edge to the network protocols, such as TCP/IP. The SWAP protocol driver expose the virtual adapter for each network to which the wireless card is connected. The SAP miniport driver maintains the state for each virtual adapter. The advantage of this architecture is that there is a different IP address for each network. The network stack sees each virtual adapter as a different wireless card interface, each of these virtual adapters should have a distinct MAC address. Each virtual adapter is given the MAC address of the underlying wireless card.

4.2 SWAP Services

The SWAP Driver, the other major software component is the SWAP Service. This service is implemented at the user level and implements the buffering and switching logic. It interacts with other SWAP nodes, and passes signaling messages to the SWAP Driver to either start or stop a switching and buffering action. The SWAP Service is responsible for signaling the switching time to the SWAP protocol driver. This signal indicates the time to switch the card, and activate another network.

5. SYSTEM DESIGN

Physical Design: Explain relationship between various components (processes, input, output & entities) of the system. Draw DFDs and other diagrams.
Software Wireless Access Point has the following required factors:

1. Entry Point for the Service.
2. Initializing the service.
3. Control Service function for Service Control Manager
4. A function to set current status of the service. Send a status report to the service control manager.
5. Error Message Log manage function.
6. Service installation code
7. Service stop code
8. Service remove code
9. Code to run the service as a console application
10. Function to control event handling from the console
11. Code to convert error messages into string and display.

The following methods will be implemented for clients:

1. Create a UDP socket
2. Connect to the server
3. Allocation of IP address for connectivity
4. Send the packet to the server
5. Program Delay to start other clients
6. Drop the connection

The following methods will be implemented for server side:

1. Start listening for connections
2. Spin forever handling requests
3. Wait for a connection and accepting on arrival
4. Send acknowledgement packet to client
5. Bounce connection to client and close the connection
6. Setup a listener on give interface and port
7. Acknowledge incoming packets to the clients
8. Call the main function again
9. Shutdown the socket

Flow Chart of the Service

Stage 1: Main Service

Stage 2: Main Service

Stage 3: Main Service

Fig8 The diagram shows the process execution of our Software Wireless Access Point.

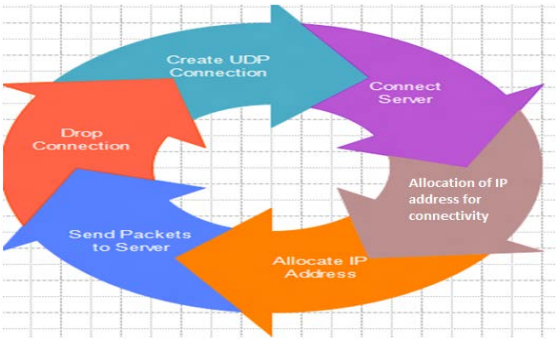

Fig9 Client Side Processing for Software Wireless Access Point

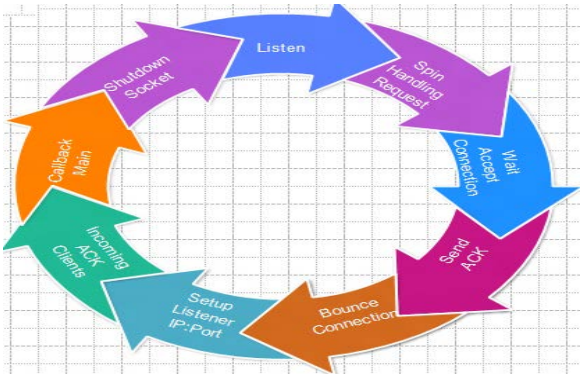

Fig10 Server Side Processing for Software Wireless Access Point

6. System Performance And Result

Testing & Debugging: Use Past Data to check whether the programmers work as intended by

6.1 Module Testing

The SWAP Project is divided into three major modules. The installation of all driver. Installation of service to handle security and data transfer. The Wi-Fi Access Point Main Interface to connect and transmit data. The module testing are done independently and we have received successful results of the specified modules as designed.

6.2 System Testing:

Software Wireless Access Point is tested in a Laboratory Environment with the following devices:

1. Client Wi-Fi Card, Connected to Test System.
2. Wi-FI Access Point { D-Link } with WEP Security
3. Android Mobile Phone with Portable Wireless Access Point Feature.

Test Reports for System Testing Graphically. The Representations are the CPU Usages scheme, to analyze performance of SWAP. We get successful connection results while System Testing. SWAP is successfully tested as a feasible solution for multiple wireless connections.

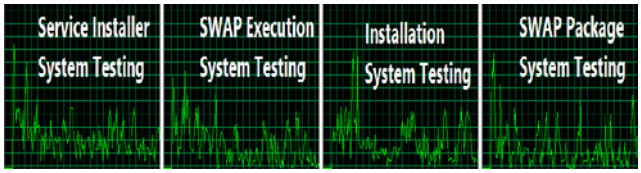

Fig11 System performance

This is the output interface of our project

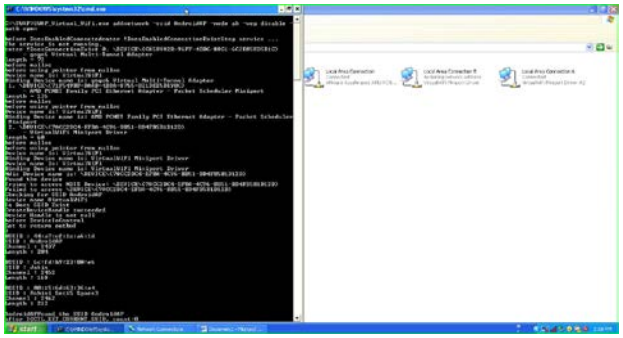

Fig11 Android Access Point connected with wireless adapter

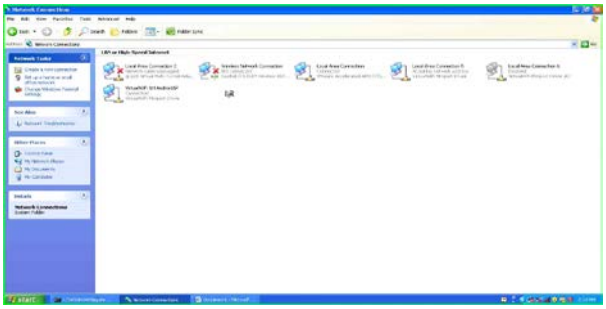

Fig12 connected to the Network using Our Software Enabled Interface

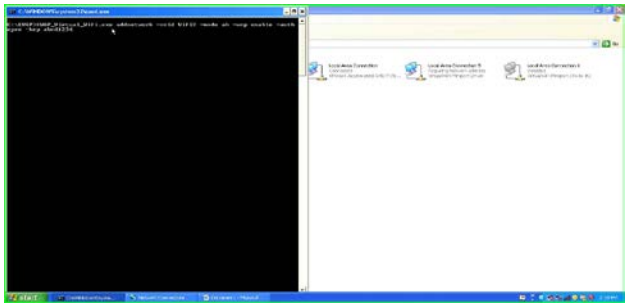

Fig13 Addnetwork command of our project

7. Conclusion

This is implementing on windows XP. We have seen, we can deal with more than one wireless network with just a single wireless adapter. This makes the wireless network more versatile than before, since we virtually have no limit of wireless networks to connect to. This is works on the both ad hoc network and infrastructure network. The future scope of this research is SWAP application only supports WEP encryption. This method is not a very secure. There are two other methods that provide more security such as Wi-Fi protected Access (WPA) and the IEEE 802.11i standard. As a conclusion, it can be said that we are still in the very beginning of the wireless networks, so they will take some more time to be fully developed.

References

[1] Chandra, R.; Bahl, P.; Bahl P., MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. IEEE Infocom, Hong Kong, 2004.

[2] Dheeraj Sanghi Samir Goel, Improving TCP performance over wireless links. In IEEE Region 10 International Conference on Global Connectivity in Energy: Computer, Communication and Control Volume: 2, 1998, pp. pages: 332-335.

[3] IEEE 802.11 Wireless Local Area Networks http://grouper.ieee.org/groups/802/11/, 2009.

[4] F. J. Corbato, SYSTEM REQUIREMENTS FOR MULTIPLE -ACCESS, TIME-SHARED COMPUTERS, Massachusetts Institute of Technology, Cambridge, MA, 1964.

[5] Jim T. Geier, Wireless LANs: Implementing Interoperable Networks, New Riders Publishing, Thousand Oaks, CA, 1998.

[6] Ramjee Prasad, Luis Munoz, WLANs and WPANs Towards 4G Wireless (Universal Personal Communications Series), Artech House, Inc., Norwood, MA, 2003.

[7] IEEE 802.11 Wireless Local Area Networks http://grouper.ieee.org/groups/802/11/, 2009.

Gyan Bahadur received the B.E. degree from RTM Nagpur University in 2008. Currently doing M.Tech in Sharda University 2011-2013.