Applying SoftEther for Networking Education

James T. Yu, Ph.D.

School of Computer Science, Telecommunications,
and Information Systems (CTI)
DePaul University
Chicago, Illinois
jyu@cs.depaul.edu

Abstract

This paper presents the use of SoftEther in supporting networking education. SoftEther is a tool to create a tunnel-based Virtual Private Network (VPN). Unlike Layer 2 tunneling protocol (L2TP) or Point to Point Tunneling Protocol (PPTP), SoftEther uses an upper layer protocol (Secured Socket Layer) to emulate a lower layer interface (Ethernet). This scheme is more flexible and easier to configure than other VPN schemes. It works with almost any Network Address Translation (NAT) router and firewall. With SoftEther, instructors and students can easily create a collaborative network for joint course projects. The tool itself is an effective course object for students to learn various networking concepts, including encapsulation, encryption, and VPN. This paper also presents an extensive experiment of SoftEther on packet trace and performance analysis as measured by round trip delay and throughput.

Introduction

This paper presents the use of SoftEther in supporting networking education. SoftEther [1] is a tool to create a virtual private network (VPN). It has the same tunneling concept as Layer2 tunneling protocol (L2TP) [2] and Point-to-Point Tunneling Protocol (PPTP [3]. The major difference is that SoftEther uses an application layer protocol to emulate a lower layer protocol (Ethernet), while other protocols emulate Point-To-Point (PPP) tunnels. This approach makes SoftEther more flexible in creating a VPN. However, it also makes SoftEther traffic difficult to detect and filter by the firewall [4]. As a result, SoftEther is considered a *pest* by many pest control tools [5] which detect and automatically remove SoftEther from a workstation. On the other hand, the ease-of-use of SoftEther offers a flexible environment to create a Pear-to-Pear (P2P) network [6]. In addition to file sharing supported by most P2P networks, SoftEther provides direct and secured access to each other's workstation. Instructors and students can create a collaborative networking environment and set access accounts to each other's workstation. The SoftEther tool itself is also an effective learning object for students to explore the basic concept of the OSI 7-layer model and data encapsulation.

The purpose of this paper is to present a comprehensive study of this tool, including discussion on protocol stacks and experiment on packet trace and performance analysis. We will also discuss how this tool is used in the classroom environment to help students learn various networking concepts.

SoftEther was developed at University of Tsukuba, Japan, and received a lot of interest in China, Japan, South Korea, and Taiwan. However, there are few documents about SoftEther in English, which implies that this tool is not popular in North America yet.

1 SoftEther Network Configuration

The traditional Ethernet network has one hub and many workstations connected to the hub as illustrated in Figure 1.

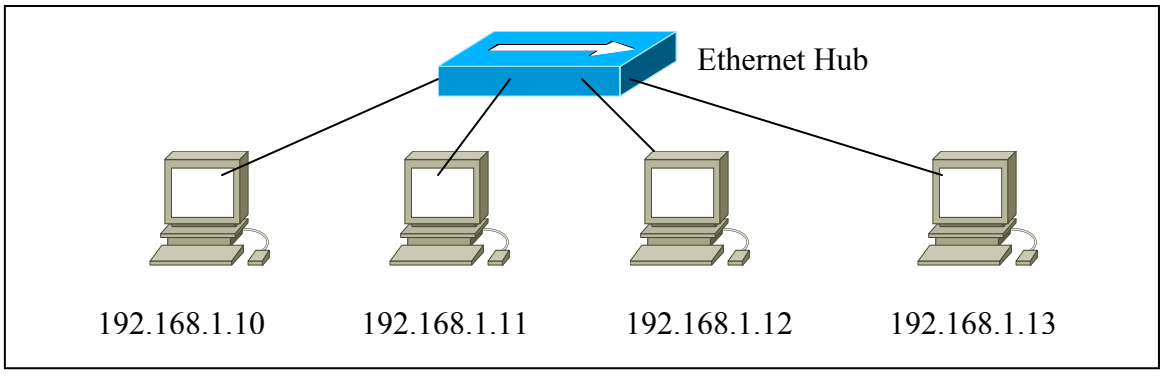

Figure 1. Traditional Ethernet Network

This network has four components:

1. Ethernet Hub

2. Network Interface Card (NIC) on each workstation

3. Ethernet Driver associated with NIC

4. UTP Cable

SoftEther emulates the network illustrated in Figure 1, and it also has four *virtual* components:

1. Virtual Ethernet Hub – The Virtual Ethernet Hub is a dedicated workstation on the *public* Internet, and it is also known as the SoftEther Server.

2. Virtual Network Interface Card (NIC) – The SoftEther NIC is installed on each workstation which could be on the private or public network.

3. Virtual Ethernet Driver – The SoftEther driver is similar in function to the Ethernet driver, and it provides the interface to the IP layer. After installing the SoftEther software, a user will see a SoftEther virtual Interface card in the Network Connections Windows as illustrated in Figure 2.

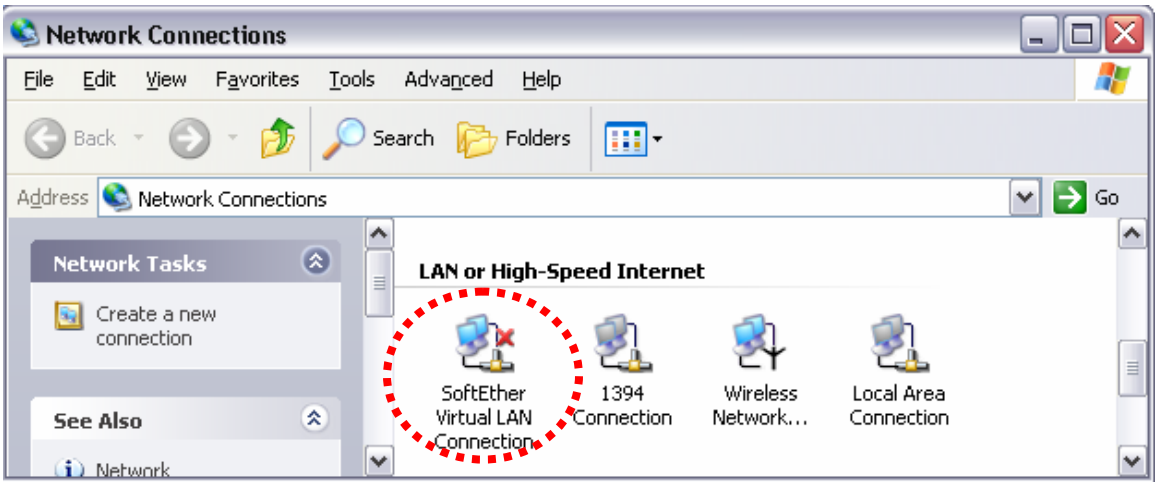

Figure 2. SoftEther Network Interface in the Network Connections Window

A user then uses the same procedure to configure an IP address on this virtual interface as a physical interface. On the SoftEther network, each workstation has multiple interfaces (physical and virtual) and multiple IP addresses, at least one IP address for each interface.

4. Virtual Cable – There is no physical connection of SoftEther network. A virtual connection is an authentication process. The administrator configures user logins on the SoftEther hub, and a user opens the SoftEther connection manager to request a virtual connection. The Windows of SoftEther connection manager is given below:

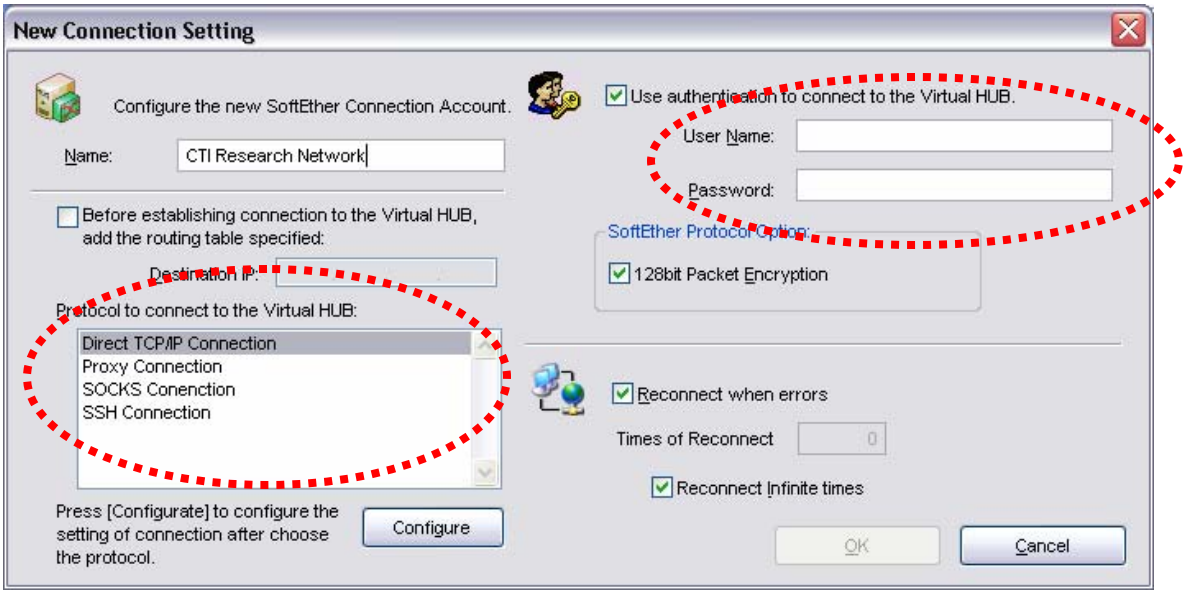

Figure 3. SoftEther Connection Manager

After the login process, a user is virtually connected to the virtual Ethernet hub. All the workstations connected to the virtual hub form a local area network (LAN). In summary, a SoftEther-enabled network has one virtual hub and many SoftEther-enabled workstations as illustrated in Figure 4.

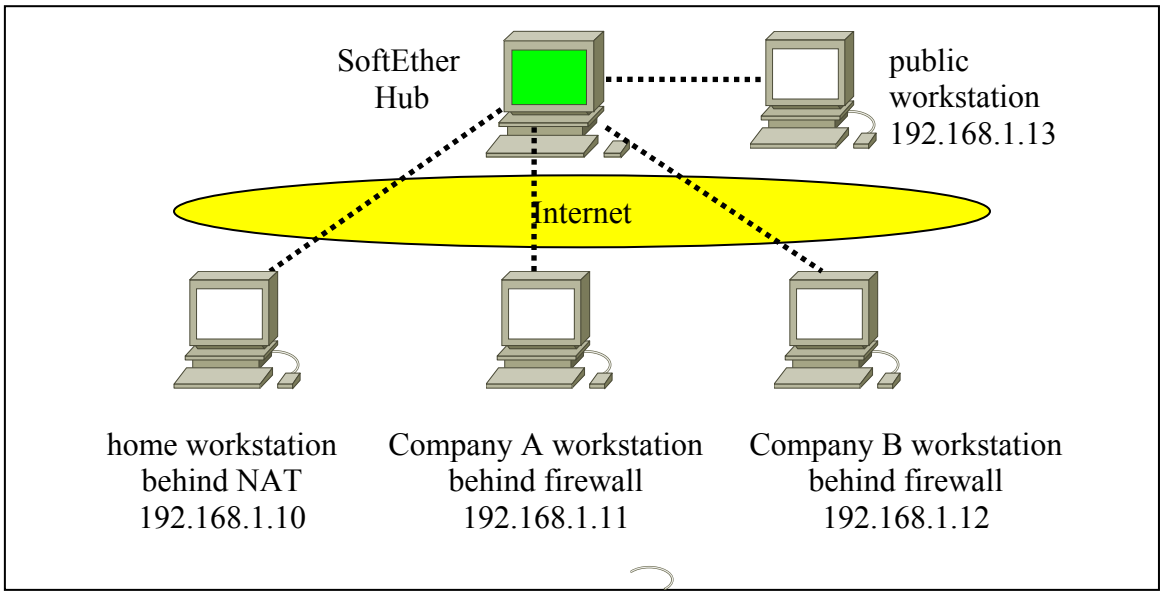

Figure 4. SoftEther Network Architecture

The above network diagram shows that a home workstation behind Network Address Translation (NAT) router can communicate with office workstations behind the enterprise firewall at different companies. The only requirement is that they are virtually connected to a SoftEther hub.

Because Windows XP and Linux both support the network bridging and routing configurations[7][8], we may extend the network to more workstations without installing SoftEther on them. A bridged configuration to extend SoftEther reach is illustrated in Figure 5.

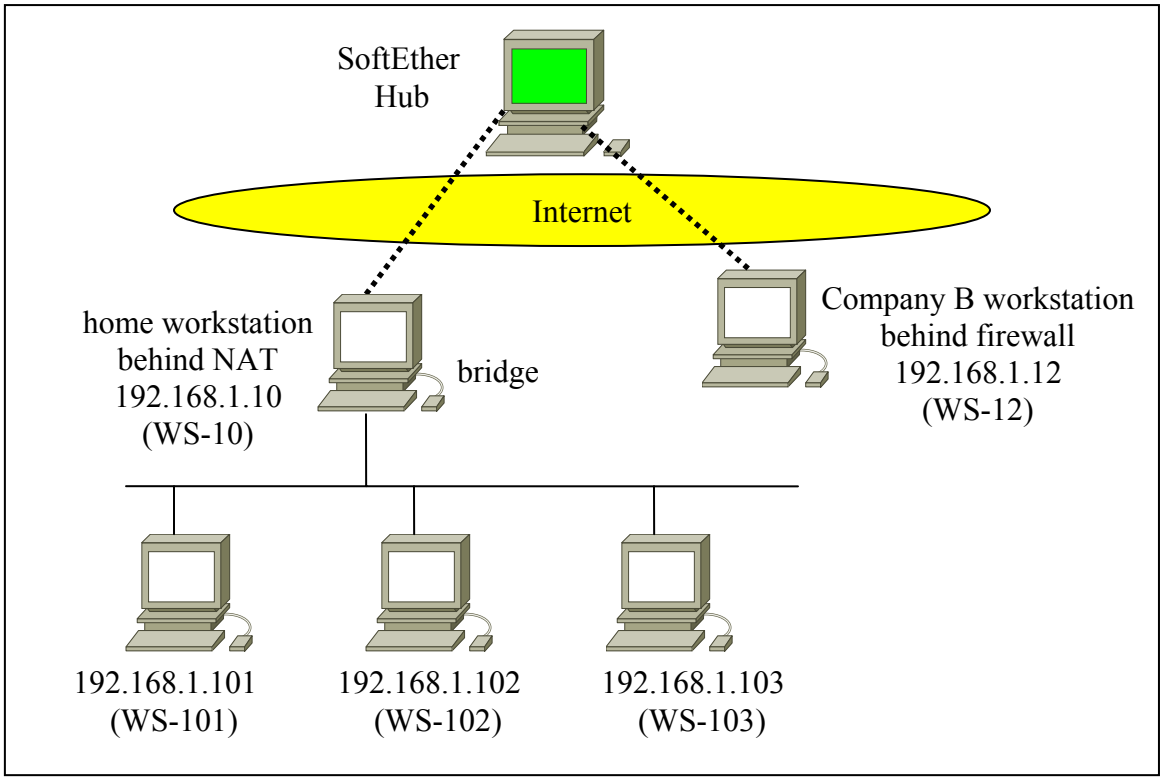

Figure 5. Bridged Configuration of SoftEther Network

In Figure 5, the bridged workstation (WS-10) needs two NICs, one NIC is for SoftEther and the other NIC is for the LAN connecting to workstations 101, 102, and 103. These three workstations (101, 102, and 103) do not have SoftEther installed on them, and they can use WS-10 as the bridge to connect to the SoftEther hub and WS-12. These three workstations do not know (and do not need to know) whether WS-12 is local or remote. As we can see from this bridged example, a SoftEther NIC is *functionally* the same as a physical NIC, and all stations are on *the same IP subnet*.

2 SoftEther Protocol Stacks

A major different between SoftEther and other tunneling protocols is that SoftEther is a *broadcast* protocol while others are based on *point-to-point* connection. The emulation of SoftEther is a virtual Ethernet network over a Wide Area Network (WAN), and offers the concept of a global and virtual LAN. Another important distinction is that SoftEther is using an application layer protocol to emulate Ethernet while other protocols are using the IP protocol to emulate a PPP tunnel. The current configuration setting of SoftEther supports four connection types: (1) direct TCP connection, (2) SSL/SSH connection, (3)

Socket connection, (4) proxy connection (see Figure 3). The protocol stacks of Secured Socket Layer (SSL) communication between clients are given as follows:

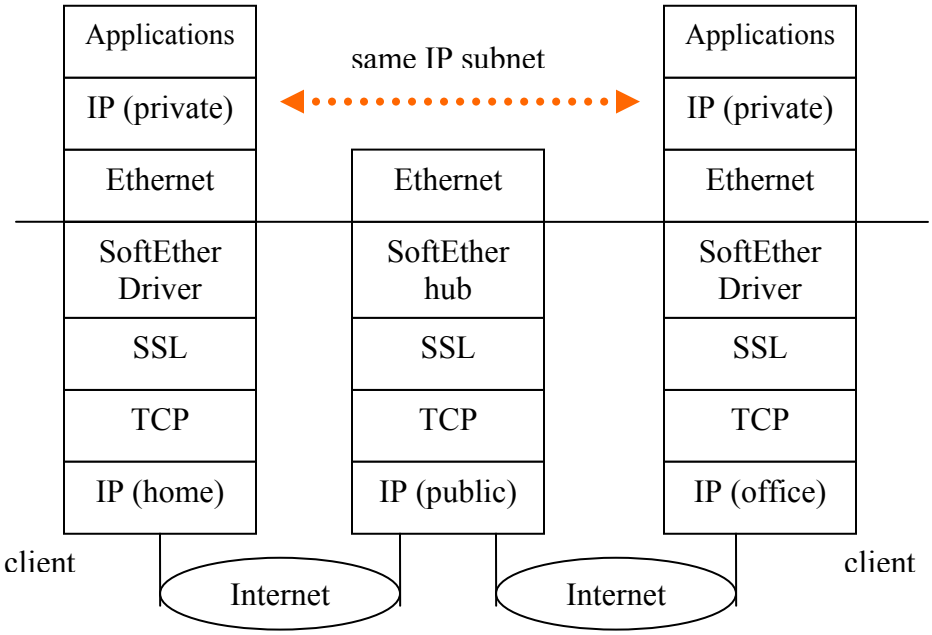

Figure 6. SoftEther Protocol Stacks

If we send a 32-bytes ICMP packet, the actual size of Ethernet frame would be 74 bytes and the actual size of SoftEther frame would be 157 bytes as illustrated in Figure 7.

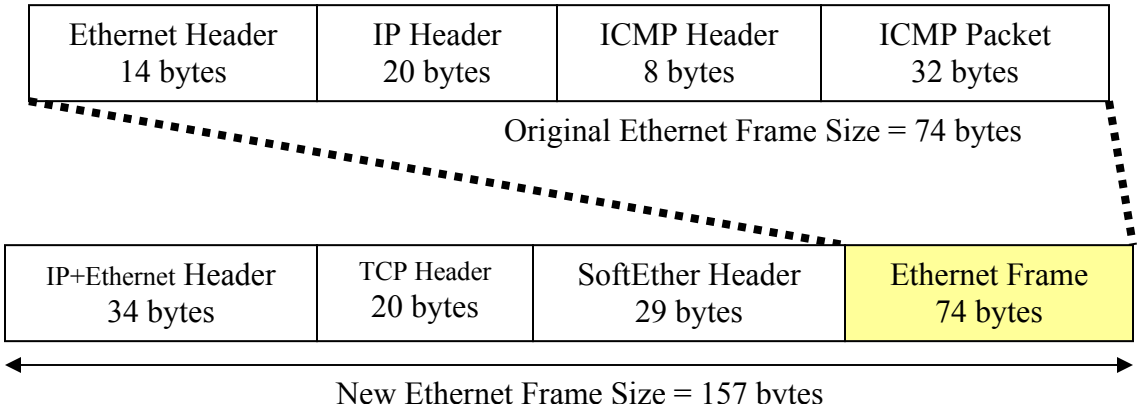

Figure 7. SoftEther Frame Format

3 SoftEther Experiments

5.1 Experiment Configuration

We conducted a comprehensive experiment to study the functions and performance of SoftEther. The network is between a home workstation (private IP address) and a public workstation (public IP address at a DePaul University Lab). The public workstation is both the SoftEther hub and a SoftEther station. The IP subnet of 192.168.1.0/24 is used exclusively for the SoftEther network. The home workstation is on a private IP network (192.168.0.0/24). The network diagram is illustrated in Figure 8.

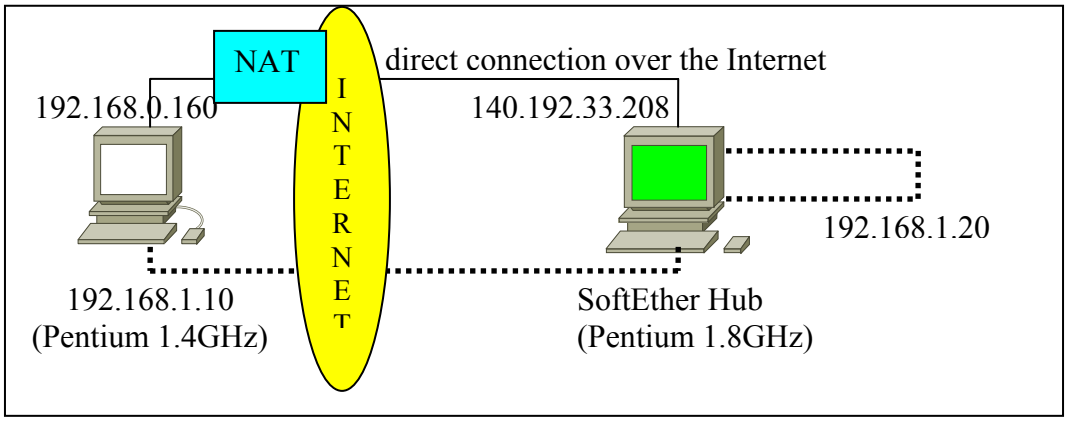

Figure 8. Network Configuration of SoftEther Experiment

The home workstation can **ping** the lab station in two ways: ping its private IP address (192.168.1.20) and ping its public IP address (140.192.33.208). If we ping the private IP address, it traverses the complete SoftEther protocol stack of Figure 6. If we ping the public IP address, the packet traverses only the IP over Ethernet protocol which is the bottom layer of Figure 6.

5.2 Packet Trace

The first experimental study is to trace packets on both the physical NIC and the SoftEther NIC. We use the ping command to generate the traffic from the home workstation to the lab workstation. On the home workstation, we turned on Ethereal [9], a network sniffer tool, to trace the packet. As shown in Figure 2, the workstation has both a physical interface and a virtual SoftEther interface. As a result, we can sniff the packets on either the physical interface or the virtual SoftEther interface. When we sniff the packers on the *virtual interface*, we can recognize individual packets of ICMP echo request and reply messages. This packet trace on the virtual interface is illustrated in Figure 9.

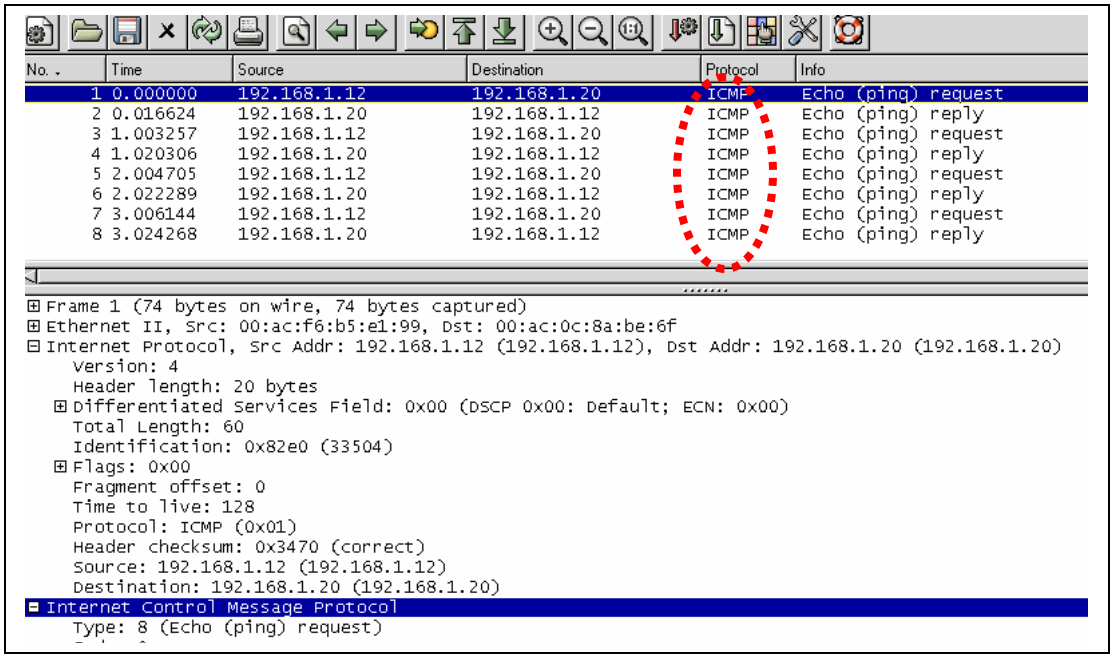

Figure 9. Packet Trace on the Virtual SoftEther NIC

If we generate the same traffic (ping the SoftEther address of the lab station from the home station) and sniff on the *physical interface*, we will see many TCP packets as illustrated in Figure 10.

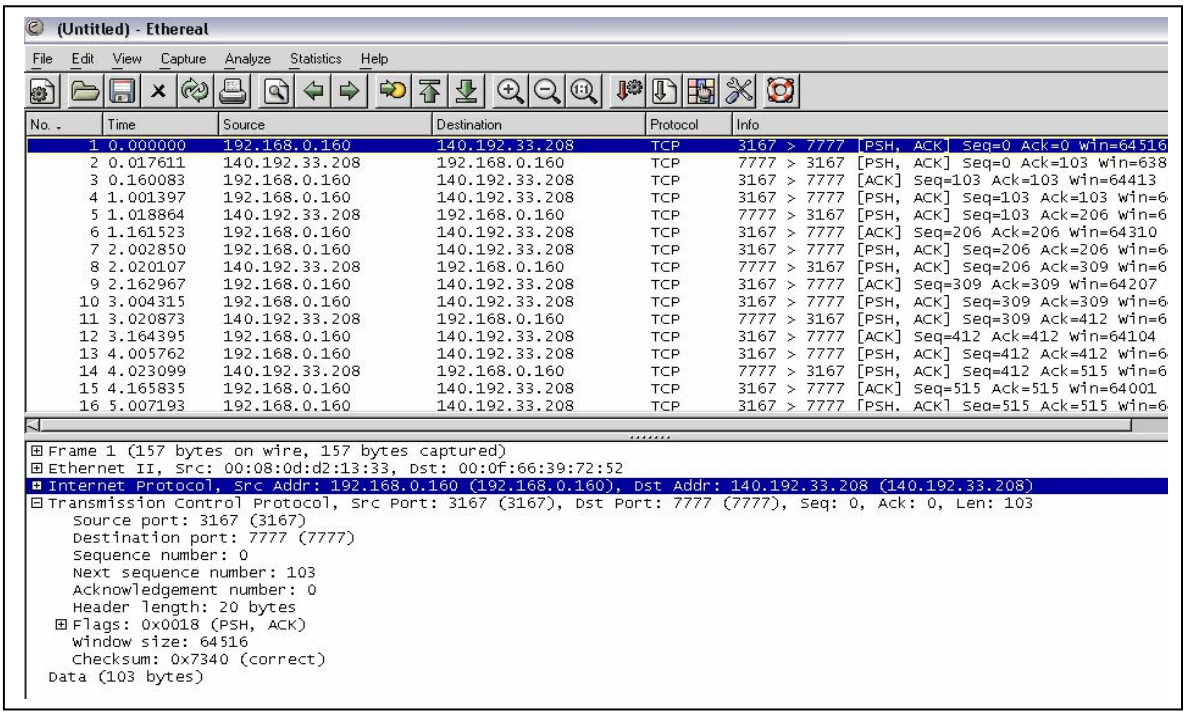

Figure 10. Packet Trace on the Physical NIC

To identify the ICMP traffic, we look for Ethernet frame with size = 157 bytes (see Figure 7). In this packet trace, we cannot tell what information or type of messages are carried as the packet information is encrypted. The header information simply shows that the tunnel is created by the TCP port=7777 which is the default SoftEther port.

5.3 Performance Analysis

Our second experimental study is on the performance of SoftEther. We follow the benchmark methodology of RFC 2544 [10]. Our first measurement is round trip delay. The measurement of *direct connection* is to **ping** the *public* IP address of the lab station from the home workstation. The measurement of SoftEther connection is to **ping** the *private* IP address. The results are given in Table 1, which shows a *constant* longer delay of SoftEther for about 3 ms. This is the CPU processing time of SoftEther protocol stacks on SoftEther hub and two workstations. The additional delay of 3 ms is not an issue of most data applications, but it may raise some concerns for real-time applications, such as Voice over IP (VoIP).

Table 1. Round Trip Delay of SoftEther and Direct Connection

ICMP Packet Size	Direct Connection (ms)	SoftEther Connection (ms)	Difference (ms)
64 bytes	14	17	3
128 bytes	18	20	2
256 bytes	21	24	3
512 bytes	30	32	2
1024 bytes	45	49	4
1450 bytes	59	64	4

The results from Table 1 show that SoftEther has a constant overhead of 2-3 ms for each packet. By analyzing the protocol stacks of Figure 6 where a packet goes through the protocol stacks 8 times for each ICMP packet, we estimated that the overhead of SoftEther protocol on each station is 0.5 ms on a Pentium III (1.4Ghz) machine. For the purpose of comparison, we also tried to conduct an experiment using Point-to-Point Tunneling Protocol (PPTP), and measured the round trip delay. Because PPTP on Windows supports a compression algorithm, we observed a constant delay of 15 ms, regardless of the packet size. This compression algorithm makes it difficult to compare the performance between SoftEther and PPTP.[1]

The third experiment is to measure throughput. The home Internet connection is ADSL.lite, and has asymmetric download and upload speeds of 1,544 kbps and 378 kbps. The experiment is based on File Transfer Protocol (FTP) between the home workstation

[1] The experiment was using SoftEther Release 1.0. SoftEther Release 2.0 is in beta, and it supports data compression.

and the lab station. We transferred both a large file (6MByte) and a small file (855Kbyte). The throughput data is given in Table 2.

Table 2. Throughput of SoftEther and Direct Connection

	File Size	Direct Connection (kbps)	SoftEther Connection (kbps)	Difference
Download	5.95 MB	998.9	876.4	12.3%
Download	855 KB	998.3	878.7	12.0%
Upload	5.95 BM	309.6	291.5	5.8%
Upload	855 KB	308.8	291.5	5.6%

The data of Table 2 shows that SoftEther has 12% more overhead for the downlink and 6% more overhead for the uplink as measured by the throughput. Because the downlink is 5 times faster than the uplink, the effect of SoftEther overhead is more significant on the downlink (12.3% vs. 5.8%). The data also shows that the file size is not a factor in performance variation. The benchmark test standard, RFC2544, also includes the performance test on packet loss However, we did not observe any packet loss in the experiment.

4 SoftEther for Networking Education

Our study and experiment of SoftEther show that this tool supports the teaching of various networking concepts.

5.1 Virtual Private Network

VPN is a private network over a public network, and it is common to use L2TP or PPTP to build a tunnel among VPN clients and a VPN gateway. In the past, the easiest way to build a VPN is to use Windows 2003 server and Windows XP clients. SoftEther offers a simpler way to build a VPN by using Windows clients only. The only requirement is to have a workstation with a public IP address. In our lab environment, we have several Windows 2003 servers; however, students are not given administration privilege on the servers. As a result, they cannot configure VPN at the lab. With SoftEther, students can create an ad hoc VPN and explore its capability. In addition, the instructor can setup a SoftEther hub at the lab, and students can create an ad hoc Pear-to-Pear (P2P) network for collaborative projects.

5.2 Data Encapsulation

The traditional OSI 7-layer model uses a lower layer protocol to encapsulate a high layer Protocol Data Unit (PDU). The most common example is IP over Ethernet (a layer-3 protocol over a layer-2 protocol). SoftEther takes a reverse direction by using a higher layer protocol (SSL) to encapsulate a lower layer protocol (Ethernet). We acknowledge that this is not a revolutionary approach as we already implemented layer-2 tunneling

protocols which encapsulate PPP packets (layer-2) in IP packets (layer-3). From an educational perspective, SoftEther shows that we can *technically* encapsulate any protocol at any network layer using any other protocol at any layer. It adds new meaning to the layered-approach to the OSI network model, and helps students explore the interfaces among protocol layers. This is also a perfect example of an *overlay* network for students to learn various protocol functions. From a practical perspective, SoftEther identifies an effective solution to address the issue of traffic blocked by firewall and NAT.

5.3 Network Security

SoftEther supports various encryption protocols RC4-MD5, AES128, and AES256, and students can configure different protocols and observe performance variations. Students may also use a network sniffer, such as Ethereal, to study encrypted packets.

SoftEther also creates a new security challenge for network administrators. Traditionally, a firewall can block any undesirable traffic based on information in the TCP/UDP/IP headers. However, a firewall would not block web traffic as it is the most common application required in many enterprise environments. SoftEther can emulate any application-layer traffic and can also use HTTPS to encrypt the packets. As a result, a firewall cannot distinguish a SoftEther packet from a *real* HTTPS packet. Although there are several studies on this issue [4], no viable solutions can address this issue yet. Currently, many network administrators simply consider SoftEther as a pest [5] and use a pest control tool to remove it from workstations in the enterprise environment

5.4 Virtual VLAN

Virtual LAN (VLAN) as specified in IEEE 802.1Q, is a well-defined network technology. SoftEther takes one step further of creating virtual VLAN to support collaborative work. Students can create their own virtual VLAN and share their private web sites and file folders with others for collaborative projects. We see many potential applications in this area. One example is to setup a public SoftEther hub, and users connect their Voice over IP (VoIP) phones to the hub using a private IP address. With this configuration, users can efficiently create a private voice network with direct dialing each other.

5 Conclusions

This paper presents how SoftEther works from both theoretical and empirical perspectives. It also shows practical applications of using SoftEther for designing VPN. The experimental results show that the SoftEther overhead is insignificant for most data applications, but it needs to be improved for delay-sensitive applications. Our conclusion is that SoftEther is more efficient and effective than L2TP and PPTP in creating a VPN. Using SoftEther, we can easily create a collaborative environment for instructors and students over any NAT router or firewall. Given that SoftEther is being labeled as a *pest* by many security tools, we do not recommend students to use SoftEther from their office environment unless they are granted permission from their IT office. In addition, we see many potential applications of SoftEther. One example is *public* SoftEther hub with commercial services on it. It is a new paradigm of portal service yet to be explored.

REFERENCES

[1] http://www.softether.com

[2] Point-to-Point Tunneling Protocol, RFC 2661, http://www.ietf.org, August, 1999

[3] Layer-2 Tunneling Protocol, RFC 2637, http://www.ietf.org, July 1999

[4] "Controlling and Monitoring SoftEther traffic Using the NetEnforcer," http://www.allot.com/media/news/Controlling%20SoftEtherv%20TN%20v3a%20ENG.pdf

[5] http://www3.ca.com/securityadvisor/pest/pest.aspx?id=453078890

[6] Rikitake, K., Nogawa, H., Tanaka, T., Nakao, K. and Shimojo, S.: Internet Security Management on Teleworking Environment, *Proceedings of the Sixth Japan Telework Society Conference*, Japan Telework Society, pp. 85-90 (2004).

[7] Chibiao Liu and James T. Yu, "Applications and Performance Analysis of Bridging with L3 Forwarding on Wireless LANs," CTI Research Symposium, November 2004

[8] J. T. Yu, "Performance Evaluation on Linux Bridge," Telecommunications System Management Conference 2004, Louisville, Kentucky, April 2004

[9] Ethereal, http://www.ethereal.com

[10] Benchmarking Methodology for Network Interconnect Devices, RFC-2544.