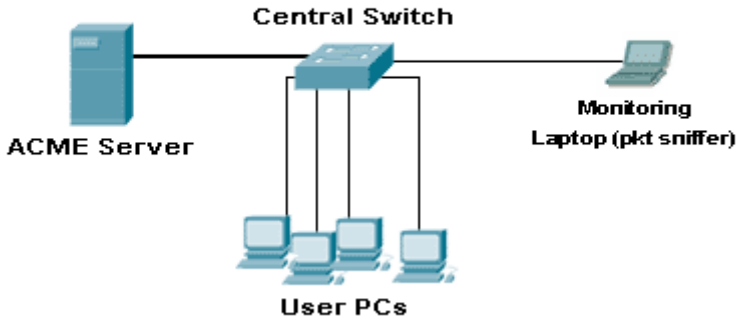
Objectives:

- Explain the need for the Transport layer.
- Identify the role of the Transport layer as it provides the end-to-end transfer of data between applications.
- Identify when it is appropriate to use TCP or UDP and provide examples of applications that use each protocol.

Intro:

ACME Inc. is having network problems again and this time things are little more complicated. According to the report sent to you, it appears that no user PC is able to communicate with the server. ACME Inc. has decided to call you again.

Topology:



The Scenario:

Since none of the user PCs are able to communicate with the server, once inside ACME office you go to check it. The server appears to be ok. ACME software is up and running which leads you to conclude that the server itself is not the problem. You decide to perform a few tests (layer 1 connectivity, layer 2 protocol tests, layer 3 reach ability, etc). Beginning from the layer 1 (physical layer), you perform a connectivity test to ensure the cabling is ok.

After running tests with proper tools, you concluded the cabling was ok.

Next step is to perform layer 2 tests. A few show commands at the console shell on the central switch (the switch which interconnects user PCs and the server) showed the switch is ok. Since the server, the layer 1 and the layer 2 are ok, you move up looking for any layer 3 issues.

From a few different user PCs you issue a ping command towards the ACME server. The output on all of them is similar and it is shown below:

```
C:\>ping 192.168.2.1

Pinging [192.168.2.1] with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=0.820ms TTL=255

Reply from 192.168.2.1: bytes=32 time=0.861ms TTL=255

Reply from 192.168.2.1: bytes=32 time=0.871ms TTL=255

Reply from 192.168.2.1: bytes=32 time=0.869ms TTL=255

Ping statistics for 192.168.2.1:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 0.820ms, Maximum = 0.871ms, Average = 0.859ms

C:\>
```

Obs: 192.168.2.1 is the IP address assigned to the ACME internal server.

Question 1:

Based on the output shown above:

- **a.** Was the ping successful?
- **b.** Is it correct to conclude that there are no layer 3 routing problems? Explain

Since no problems were found on layers 1, 2 and 3 and yet no user PC was able to fully communicate with the server, you decide to check the OSI Layer 4, the Transport Layer.

In order to check layer 4 connectivity you decided to use a packet sniffer to capture network packets being sent within the ACME network.

A packet sniffer is an application written to capture packets flowing within a specific network segment and it is a powerful tool when troubleshooting network problems.

You connected your laptop to a proper location inside ACME network and configured the switch to allow monitoring (monitoring traffic in a switch will be covered later on the course).

A packet sniffer software was started on the laptop. With the packet sniffer still running, you also started the ACME program from one of the user PCs. All packets sent from the user PC towards the server and vice-versa are now being captured by the packet sniffer program running on your laptop.

Below is the relevant portion of the packet sniffer output. Notice only packet headers are being shown:

```
root@laptop:~# tcpdump -n -i eth0 port 25000 and host 192.168.2.1

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

16:44:39.028192 IP 192.168.2.1.52738 > 192.168.2.2.25000: S  733244312:1733244312(0)
win 4128 <mss 1460>

16:44:39.028298 IP 192.168.2.2.25000 > 192.168.2.1.52738: S 2441072844:2441072844(0)
ack 1733244313 win 5840 <mss 1460>

16:44:39.029317 IP 192.168.2.1.52738 > 192.168.2.2.25000: . ack 1 win 4128

16:44:39.029906 IP 192.168.2.2.25000 > 192.168.2.1.52738: R 2441072845:2441072845(0)
win 0

16:44:39.029971 IP 192.168.2.2.25000 > 192.168.2.1.52738: R 2441076973:2441076973(0)
win 0

16:44:39.030000 IP 192.168.2.2.25000 > 192.168.2.1.52738: R 2441085229:2441085229(0)
win 0

16:44:39.050771 IP 192.168.2.2.25000 > 192.168.2.1.52738: P 1:39(38) ack 1 win 5840

16:44:39.051624 IP 192.168.2.1.52738 > 192.168.2.2.25000: R 1733244313:1733244313(0)
win 0
```

Note: tcpdump is a very popular command line packet sniffer on Unix-like systems. Wireshark (and Ethereal) are fully compatible with tcpdump's output file format.

On the output above, is possible to see the TCP 3-Way handshake (the three first packets shown) but right after the last step of the 3-Way handshake (the last acknowledge packet), a computer with IP Address 192.168.2.2 sends out a sequence of TCP RESET packets to the server.

Question 2:

a. What portion of the output (which TCP flags within the first 3 lines) identifies the TCP 3-Way Handshake?
b. What portion of the output identifies the TCP Reset packets?
c. Are TCP RESET packets suppose to appear at this point?

TCP is a connection oriented protocol and therefore, it establishes sessions between source and destination. The TCP RESET packets sent by 192.168.2.2 are suddenly terminating the TCP session and therefore, no upper layer communication happens. This explains why pings are successful but the ACME software does not work: Because the software needs to communicates reliably, all its communication is based on the TCP protocol but all TCP sessions are being terminated before any data is transferred.

The problem is isolated but the cause is still unknown. Some further investigation is necessary.

According to the packet sniffer output shown, the computer 192.168.2.2 is generating the TCP RESET packets. 192.168.2.2 is the IP address of the user PC you are using to perform the tests. This PC could be the one generating the malicious traffic but it is still hard to tell.

Question 3:

a. Which would be the best way to identify the source of the malicious traffic?
b. Is this information available? Where?

As stated before, the packet sniffer output only shows packets headers but all the packet information is available on the captured data. This means the MAC address of the PC generating the malicious traffic is embedded within the packets, too.

After a deeper analysis on the TCP RESET packets captured by the packet sniffer, you conclude the MAC Address of the PC generating the traffic is 00:C0:A8:7C:ED:EA

The MAC address of the PC you performed the test is 00:12:3F:E8:8B:A8 and thus it is not the source of the malicious traffic. There is another PC on the network which is sending out packets with a fake source IP Address. This technique of faking source IP address of a packet is called *IP Spoofing.*

Question 4:

a. Would be possible to respond to a query which was sent within a source IP spoofed packet? Explain.

Because MAC addresses have a flat address space (no hierarchy) there is no easy way to find the source of the malicious traffic. You must check every computer on the network until the MAC address 00:C0:A8:7C:ED:EA is found.

Since all the user PCs within ACME Inc. are running Windows XP, the command **ipconfig /all** can be used to check the MAC address of the PCs. A Sample output is shown below:

```
C:\>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : USERPC-07

        Primary Dns Suffix  . . . . . . . : acme.com

        Node Type . . . . . . . . . . . . : Broadcast

        IP Routing Enabled. . . . . . . . : No
```

```
        WINS Proxy Enabled. . . . . . . . : No

        DNS Suffix Search List. . . . . . : acme.com

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :

        Description . . . . . . . . . . . : Broadcom NetXtreme 57xx Gigabit Controller

        Physical Address. . . . . . . . . : 00-12-3F-E8-88-AE

        Dhcp Enabled. . . . . . . . . . . : Yes

        Autoconfiguration Enabled . . . . : Yes

        IP Address. . . . . . . . . . . . : 192.168.2.21

        Subnet Mask . . . . . . . . . . . : 255.255.255.0

        Default Gateway . . . . . . . . . : 192.168.2.1

        DHCP Server . . . . . . . . . . . : 192.168.2.1

        DNS Servers . . . . . . . . . . . : 24.25.5.150

                                            24.25.5.149

        Lease Obtained. . . . . . . . . . : Monday, September 10, 2007 10:03:01

AM

        Lease Expires . . . . . . . . . . : Tuesday, September 11, 2007 10:03:01

 AM

C:\>
```

After a few tries the PC generating the malicious traffic was found. A quick look reveals the anti-virus software is outdated and the PC is infected by a virus which terminates new TCP sessions. The virus was monitoring the network searching for TCP SYN packets. When they were found, the virus faked the IP address used by the source of that specific TCP session and sent a TCP RESET to the real destination as if it was the real source. The real destination (ACME Server) would accept the TCP reset packet and terminate the session because the virus replicated all the TCP flags within the packet. Because the real source (other ACME user PCs) would never receive data traffic from the server (the real destination) the TCP timers on the user PCs would eventually expire and the source would also assume the session terminated.

No TCP sessions would be established while this virus was running.

You removed the virus and updated the anti-virus. With no virus, the network goes back to normal operation.

Answers:

Q1a. *Yes.*

Q1b. *Yes. Since the pings were successful, the network layer is routing packets correctly.*

Q2a*.*

The three first packets happening at:

16:44:39.028192: SYN from the client to the server

16:44:39.028298: SYN+ACK from the server to the client

16:44:39.028317: ACK from the client to the server.

The flag S identifies the packets as TCP SYN packets.

Q2b.

The flag R defines the packet as a TCP Reset packet.

Q2c.

TCP Reset packets are used to immediately terminate a TCP connection.

Q2d.

No. No data was even sent.

Q3a.

The MAC Address makes it possible to identify the source of the malicious traffic.

Q3b.

Yes. The output shown only show packet headers but the packet sniffer captured entire packets.

Q4a.

No. Because the source IP address is not real, the answer would reach the device configured with the spoofed ip address, not the device which actually sent the packet with the spoofed IP address.