**MCSE 1 Lecture 4**

**OSI and TCP/IP models**

**OSI and TCP/IP Models**

When two computers communicate with each other, the message that travels between them must be transformed to facilitate transmission on whatever medium is being used.

This transformation involves many steps. A thorough understanding of this process will be extremely helpful in troubleshooting network problems.

The OSI (Open Systems Interconnection) model is a reference model that helps us to understand how data is transformed from one format to another to make it possible to transfer data between computers. Figure 1 shows the 7 layers in the model. Take note of the data gram names for each layer.

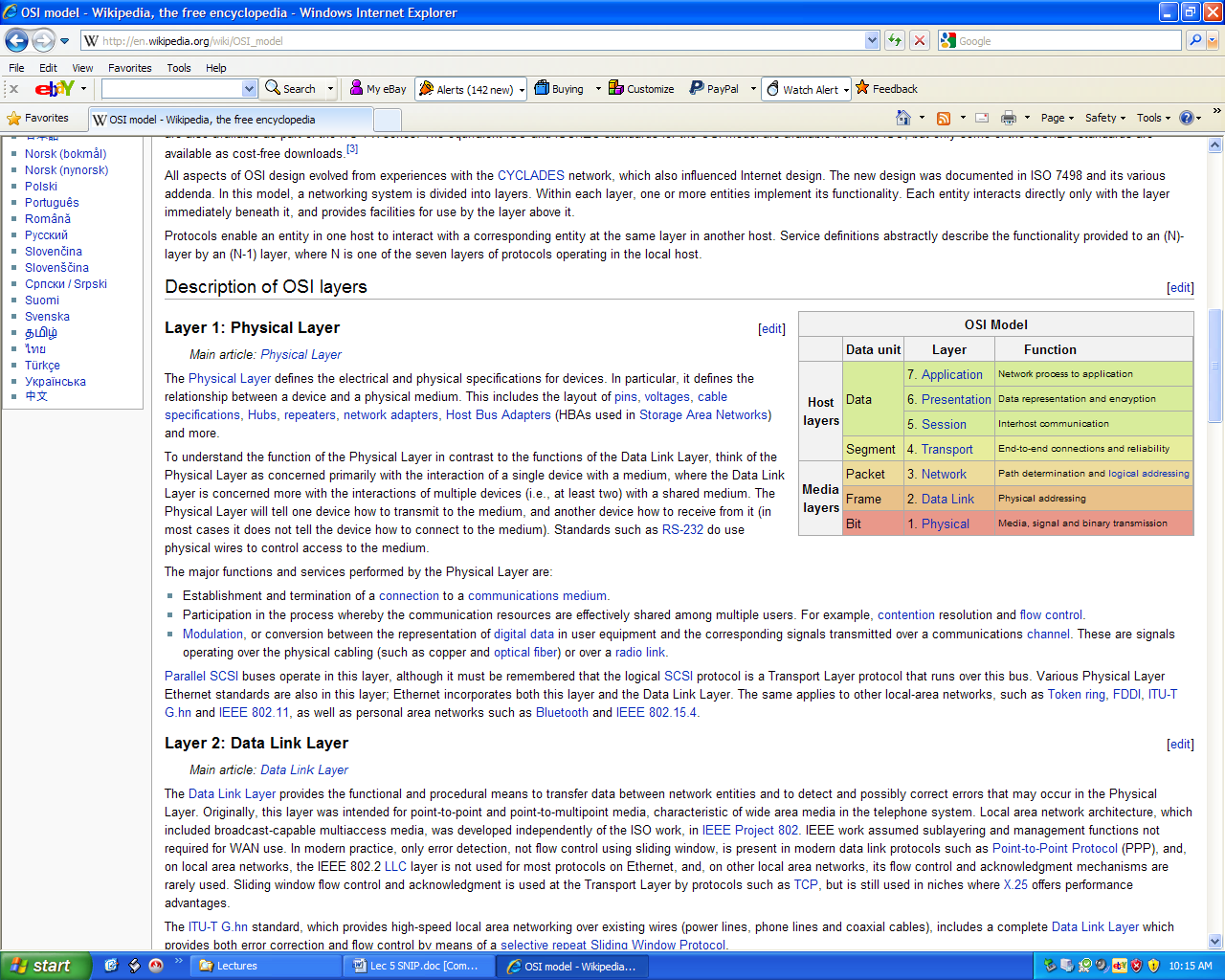


Fig. 1 The OSI model

You can think of the model as being different stages in your computer. Each stage is implemented by software protocols. The network card makes up the physical layer and half of the data link layer.

Each layer serves a different purpose. The function of each layer and some of the protocols found at each layer are explained in table 1.

|  |  |  |
| --- | --- | --- |
| **Layer** | **Protocol** | **Function** |
| **Application** | **DNS, HTTP, FTP, POP, SMTP, DHCP**, Telnet | Interfaces between the user’s application and the protocol stack |
| **Presentation** | MIME, SSL, TLS | Applies services like encoding (ASCII, EBCDIC), encryption, and compression |
| **Session** | NetBIOS, Named Pipes, half duplex, full duplex | Establish, manage, and terminates dialogues between applications running on the local host and the remote host. |
| **Transport** | **TCP, UDP,** PPTP,L2TP | Segment/de-segment the data and provide error checking, flow control and reliability in the form of ACK packets |
| **Network** | **IP, ICMP**, IPsec | Routes the packets between networks. Provides the logical addressing (IP) |
| **Data Link** | **FRAME RELAY, PPP, ARP,Ethernet** | Creates the fields making up the frame.  Provides the physical address. |
| **Physical** | 100Base-Tx, POTS, DSL, Cable modem,  Wireless, fiber | Convert bits into voltages, light or radio waves. Defines the pin-outs of the connections and the timing of the signals |

Table 1 Protocols and function of the OSI layers

**Addressing**

Everyone is familiar with an IP address. Every computer requires an IP address if it is going to communicate with another computer. Addressing is more complicated than just having an IP address. There are 3 different sets of source and destination addresses placed in a frame.

1. **Transport layer Addressing (Port numbers)**

The transport layer places both a source and destination port number in each segment. For instance, if you use Windows Explorer to access a WEB site, the transport layer places the “well known” port number 80 for HTTP (web service protocol), in the **destination port** field. (**Well known port #’s = 1 to 1023**).

The local computer generates a random **source port** number between **1024 and 65,535** and places that in the field for the source port number.

The WEB server may be providing other services such as DNS, DHCP, FTP, etc. When a frame arrives the server must discover which type of service to give the frame to. This is where the destination port number comes into play. Each of the application layer protocols has its own “well known” port number associated with it. When the frame arrives at the remote WEB server, the destination port number 80 tells the server to give this segment to the HTTP protocol which in turn will pass it to the WEB service.

When the WEB server responds, it must place the random port number used by the originating host, in the destination port field for the segment making up the response message. The originating host might be running multiple sessions such as WEB browsing, mail, and ftp downloading. When the reply arrives from the WEB server, the destination port number tells the local host which protocol originated the communications. The reply can then be handed off to the WEB browser rather than the mail or ftp applications.

**2. Network Addressing (logical addressing)**

The network layer receives the segments from the transport layer and produces **packets** by adding fields for the source and destination IP addresses. (There are other forms of logical addressing such as Novell’s IPX and Apple’s AppleTalk but they are rapidly being phased out in favor of IP).

**The IP addresses never change from source to destination**. The destination IP address ensures delivery to the correct host while the source address ensures the remote host will reply to the correct local host.

It is up to the Administrator or DHCP server to configure the local host with an IP address.

**3. Data Link Addressing (physical addressing)**

At layer 2, the data link layer, the packets received from the network layer are transformed into frames by adding the source and destination physical addresses to the header of the frame and a CRC error checking number to the trailer of the frame.

Today, just about all local area networks (LANs) use Ethernet. The physical address is called a **MAC** address. Every Ethernet card has a MAC address burnt into an EEPROM chip on the NIC. No two MAC addresses can be the same in the whole world.

When the frame has to be sent outside the local network as it would be when it is sent to the Internet, the Ethernet header and trailer have to be discarded and the packet has to be repackaged with the header and trailer that corresponds to the new layer 2 protocol. This **WAN** protocol might be frame relay. In this case the layer 2 address is called a **DLCI** instead of a MAC address. The administrator must configure the source and destination DLCI numbers through software. The service provider decides what DLCI numbers the customer will use, just like the telephone company decides what telephone number you will use.

Figure 2 illustrates how data is sent between a WEB browser on the local machine and a WEB server on the remote machine.

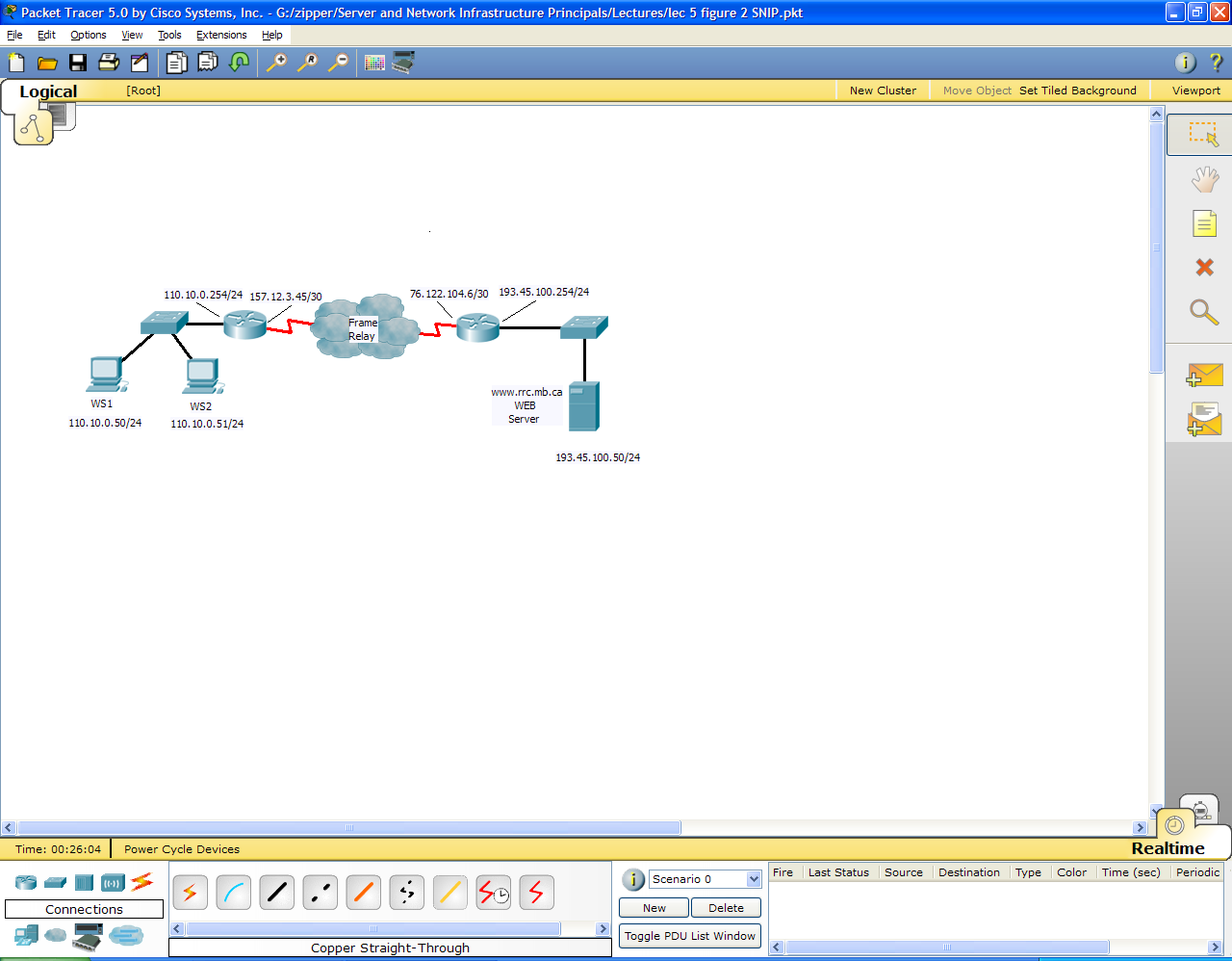


Fig. 2 Network to illustrate WEB services

What happens when a user at WS1 opens the Internet Browser and types www.rrc.mb.ca into the URL?

The first thing that happens is the fully qualified domain name, www.rrc.mb.ca is translated into the IP address 193.45.100.50 by DNS. (DNS will be explained in an up-coming chapter).

The browser must then send a request for a WEB page to the WEB server. The browser calls up HTTP at the application layer to accept the request and pass it down the protocol stack to the presentation layer. The presentation layer performs the necessary formatting and then passes the data to the session layer.

The session layer initiates a connection request and passes the data to the transport layer. The transport layer uses the TCP protocol to break the data into smaller chunks. TCP formats each segment with a destination port of 80 for WEB services. The local host then generates a random source port between 1024 and 65,535. Let’s say it generates a source port number of 60000. The data and segments appear as shown below in figure 3.

**Data from the session layer arrives at the transport layer**

|  |  |  |
| --- | --- | --- |
| Source port  60000 | Destination Port  80 | data |

1st segment 2nd 3rd 4th 5th

Fig. 3 Data from the session layer is broken into segments at the transport layer

Each segment is then passed to the network layer where it gets a set of source and destination IP addresses. This is shown in figure 4.

|  |  |  |
| --- | --- | --- |
| Source port  60000 | Destination Port  80 | data |

1st segment 2nd 3rd 4th 5th

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source**  **IP**  **110.10.0.50** | **Destination**  **IP**  **193.45.100.50** | Source  port  60000 | Destination  Port  80 | data |

1st packet 2nd 3rd

Fig. 4 Segments are turned into packets at the network layer

The network layer then passes the packet to the Data Link layer where the physical address is added to the header and the CRC is added to the trailer to form an Ethernet frame. Figure 5 illustrates the creation of a frame.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source**  **IP**  **110.10.0.50** | **Destination**  **IP**  **193.45.100.50** | Source  port  60000 | Destination  Port  80 | data |

1st packet 2nd 3rd

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dest.  MAC | Source  MAC | **Source**  **IP**  **110.10.0.50** | **Destination**  **IP**  **193.45.100.50** | Source  port  60000 | Destination  Port  80 | data | CRC |

1st frame

Fig. 5 Packets are turned into frames at the Data Link layer

The MAC address is unique to Ethernet. Ethernet addresses are only significant on the local segment. In figure 2 the Source MAC would be the MAC address burnt into the NIC of WS1. The Destination MAC would be the MAC address burnt into the NIC on the gateway. The gateway in this case is the router interface with IP address 110.10.0.254.

The MAC address is a 48 bit address represented as 6 groupings of 2 hexadecimal digits. Figure 6 shows the MAC address in the output of

ipconfig /all.

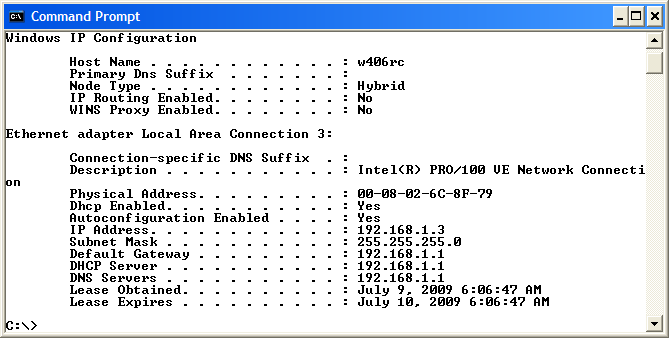


Fig. 6 The MAC address of a computer as shown with **ipconfig /all**

WS1 wants to send the Ethernet frame to the MAC address of the gateway but, it does not know the MAC address of the gateway. It does know the IP address of the gateway because either the administrator statically configured it on WS1 or DHCP sent the gateway address to WS1.

WS1 sends out an **ARP** (Address Resolution Protocol) request as a broadcast on the segment. The broadcast frame says “whoever 110.10.0.254 is, please send me your MAC address”. Since it is a broadcast, all hosts including the gateway, accept the frame as if it were addressed to that host. When they all inspect the data request, only the gateway with address 110.10.0.254 will respond. It sends WS1 a frame that tells it what the gateway’s MAC address is.

You can check the MAC table by typing **arp –a** at the DOS prompt. This is illustrated in figure 7.

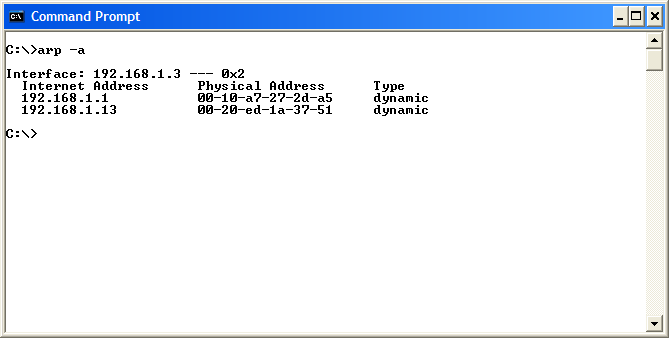


Fig. 7 Using the ARP –a command to view the MAC table

Now WS1 can send the WEB page request to the gateway’s MAC address. When the router (gateway) gets the frame from WS1, it strips off the layer 2 Ethernet header and trailer and passes the packet to the network layer.

At the network layer, the router logically AND’s the subnet mask with the destination IP address to extract the network address. The router then checks its routing table to see if it knows where that network is. In this case, the network is out the serial interface facing the internet. This interface is using a different layer 2 protocol. It is using frame relay instead of Ethernet. The router adds a new layer 2 header and trailer that are required by frame relay frames. This is shown in figure 8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dest.  MAC | Source  MAC | **Source**  **IP**  **110.10.0.50** | **Destination**  **IP**  **193.45.100.50** | Source  port  60000 | Destination  Port  80 | data | CRC |

Ethernet Frame arrives from WS1 at the Ethernet interface

The router strips off the Ethernet header and trailer and adds the Frame Relay header and trailer.

Router

Frame exits the serial interface as a Frame Relay frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dest.  DLCI | Source  DLCI | **Source**  **IP**  **110.10.0.50** | **Destination**  **IP**  **193.45.100.50** | Source  port  60000 | Destination  Port  80 | data | CRC |

Fig. 8 Packets are turned into frames at the Data Link layer

The cloud, in figure 2 represents the Internet. The frame relay frames are passed from router to router until they arrive at the destination router’s serial interface designated by the IP address 76.122.104.6.

This router reverses the process performed by the originating router. It strips off

the layer 2 frame relay header and trailer. The router logically AND’s the subnet mask with the destination IP address to get the network address (193.45.100.0). It checks its routing table to see if there is an entry for 193.45.100.0. The router finds that network is associated with the Ethernet interface.

Now the router knows the destination host (WEB server) is on the Ethernet segment. The router wants to forward the frames to the WEB server but it does not know the MAC address of the WEB server. The router does know the IP address of the WEB server because that is the destination IP address on the frame it just received.

The router ARPs for the MAC address on the Ethernet segment. The WEB server responds with its MAC address. The router encapsulates the layer 3 packet that just arrived with a new layer 2 Ethernet header and trailer. The Ethernet frame is then sent to the WEB server.

At the WEB server, the frame is accepted by layer 2 because the destination MAC address is correct. The server strips off the layer 2 header and trailer and passes the packet to the network layer. The network layer checks the destination IP address and finds it is correct. The server strips off the layer 3 header and passes the segment to the transport layer.

At the transport layer, the destination port number is checked and found to be 80. The transport layer passes the segment up the stack to the HTTP protocol which then passes it to the WEB service.

The WEB server then responds with the requested page and the whole process repeats itself as the data travels from the WEB server back to WS1.

**Local vs Remote Messaging**

When a computer wants to send a frame to another computer the other computer may be on the same segment (same network), or it may be on a distant segment, (different network).

How does the sending computer know to send the message locally or send it to the gateway to be delivered to a distant network?

The computer must apply the subnet mask to the destination IP address to obtain the network of the targeted computer. Then the computer applies the subnet mask to its own IP address to obtain its own network address. It compares the two networks. If they are the same the message can be sent locally. The computer will ARP for the targeted computer’s MAC address and then deliver the package directly to the targeted computer.

In figure 2, if WS1 pings WS2, WS1 applies the subnet mask as shown below:

IP of WS2 = 110.10.0.51 = 01101010.00001010.00000000.00110011

Subnet Mask = 255.255.255.0 = 11111111.11111111.11111111.00000000

Network = 01101010.00001010.00000000.00000000

The network address of WS2 = **110.10.0.0 Network addresses**

**are the same**

Now WS1 applies the subnet mask to itself

IP of WS1 = 110.10.0.50 = 01101010.00001010.00000000.00110010

Subnet Mask = 255.255.255.0 = 11111111.11111111.11111111.00000000

Network = 01101010.00001010.00000000.00000000

The network address of WS1 = **110.10.0.0**

WS1 now knows WS2 is on the same segment since the network addresses turned out to be the same.

**Remote Messaging**

If WS1 pings the WEB server the results are different.

IP of server = 193.45.100.50 = 11000001.00101101.01100100.00110010

Subnet Mask = 255.255.255.0 = 11111111.11111111.11111111.00000000

Network = 11000001.00101101.01100100.00000000

The network address of server = **193.45.100.0 Network addresses**

**are different**

Now WS1 applies the subnet mask to itself

IP of WS1 = 110.10.0.50 = 01101010.00001010.00000000.00110010

Subnet Mask = 255.255.255.0 = 11111111.11111111.11111111.00000000

Network = 01101010.00001010.00000000.00000000

The network address of WS1 = **110.10.0.0**

Because the network address of the server is different than the network address of WS1, WS1 knows it must send the package to the gateway to be delivered. WS1 ARPs for the MAC address of the gateway and then sends the ping to the gateway. The gateway then delivers the ping to the next router as it travels to the server.

**TCP/IP Model**

The OSI model is a great educational tool to aid in understanding communications problems but, it does not relate well to the actual implementation of the processes as they are found in a computer.

The TCP/IP model originated with a military/research organization called DARPA. It lends itself more closely to the actual processes found in the protocol stack. The two models are compared below in figure 9.

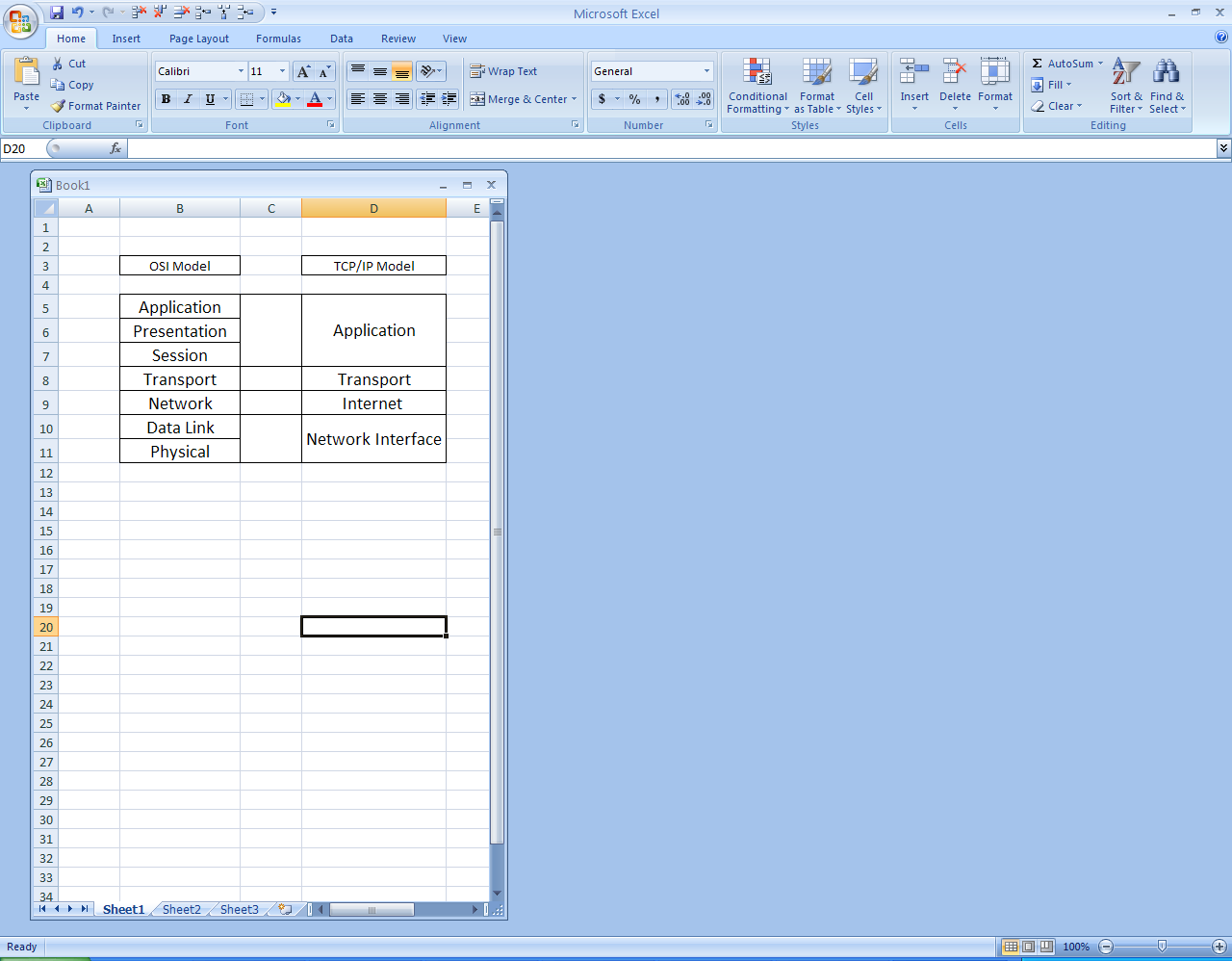


Fig. 9 The 7-layer OSI model and the 4-layer TCP/IP model

The Application layer in the TCP/IP model does the same job as the Application, Presentation and Session layers of the OSI model

The Transport layer is the same for both models.

The Network layer in the OSI model is called the Internet layer in the TCP/IP model. Both layers perform the same task.

The Data Link layer and Physical layer in the OSI model has been combined into the Network Interface layer in the TCP/IP model. The network Interface layer is actually the NIC.

For educational purposes, the OSI model is usually preferred. Since both models appear quite frequently in literature, it is important to be aware of both of them.