NETWORK TECHNOLOGY WITH ADMINISTRATION AND MAINTENANCE

Name : ______________________ ID Number : __________

Building a Network with the OSI Model

The Open Systems Interconnection (OSI) seven-layer model is a guideline, a template that breaks down how a network functions into seven parts called layers. If you want to get into networking you must understand the OSI seven-layer model in great detail.

The OSI seven-layer model provides a practical model for networks. The model provides two things. For network techs, the OSI seven-layer model provides a powerful tool for diagnosing problems. Understanding the model enables a tech to determine quickly at what layer a problem can occur and thus zero in on a solution without wasting a lot of time on false leads. The model also provides a common language to describe networks—a way for us to communicate to each other about the functions of a network. A router operates at Layer 3 of the OSI seven-layer model, for example, so you'll hear techs (and Web sites) refer to it as a "Layer 3 switch." That's a use of the OSI seven-layer model as language.

Biography of a Model

All models are a simplified representation of the real thing. The human model ignores the many different types of body shapes, using only a single "optimal" figure.

Additionally, a model must have at least all the major functions of the real item, but what constitutes a major rather than a minor function is open to opinion.

In modeling networks, the OSI seven-layer model faces similar challenges. What functions define all networks? What details can be omitted and yet not render the model inaccurate? Does the model retain its usefulness when describing a network that does not employ all the layers?

In the early days of networking, different manufacturers made unique types of networks that functioned fairly well. But each network had its own cabling, hardware, drivers, naming conventions, and many other unique features. In fact, most commonly, a single manufacturer would provide everything for a customer: cabling, NICs, hubs, and drivers, even all the software, in one complete and expensive package.

Although these networks worked fine as stand-alone networks, the proprietary nature of the hardware and software made it difficult—to put it mildly—to connect networks of multiple manufacturers. To interconnect networks and improve networking as a whole, someone needed to create a guide, a model that described the functions of a network, so that people who made hardware and software could work together to make networks that worked together well.

The International Organization for Standardization, known as ISO, proposed the OSI seven-layer model. The OSI seven-layer model provides precise terminology for discussing networks

The Seven Layers in Action

Each layer in the OSI seven-layer model defines a challenge in computer networking, and the protocols that operate at that layer offer solutions to those challenges. Protocols define rules, regulations, standards, and procedures so that hardware and software developers can make devices and applications that function properly. The OSI model encourages modular design in networking, meaning that each protocol is designed to deal with a specific layer and to have as little to do with the operation of other layers as possible. Each protocol needs to understand the protocols handling the layers directly above and below it, but it can, and should, be oblivious to the protocols handling the other layers.

The seven layers are:

- Layer 7 Application
- Layer 6 Presentation
- Layer 5 Session
- Layer 4 Transport
- Layer 3 Network
- Layer 2 Data Link
- Layer 1 Physical

Network Hardware and Layers 1-2

Clearly the network needs a physical channel through which it can move bits of data between systems. Most networks use a cable. This cable, known in the networking industry as unshielded twisted pair (UTP), usually contains four pairs of wires that transmit data.

Another key piece of hardware the network uses is a special box-like device called a hub (Figure 1.1), often tucked away in a closet or an equipment room. Each system on the network has its own cable that runs to the hub. Think of the hub as being like one of those old-time telephone switchboards, where operators created connections between persons who called in wanting to reach other telephone users. Layer 1 of the OSI model defines the method of moving data between computers. So the cabling and hubs are part of the Physical layer (Layer 1). Anything that moves data from one system to another, such as copper cabling, fiber optics, even radio waves, is part of the Physical layer. Layer 1 doesn't care what data goes through; it just moves the data from one system to another system. Figure 1.2 shows the OSI seven-layer model.

Figure 1.1 – A typical hub

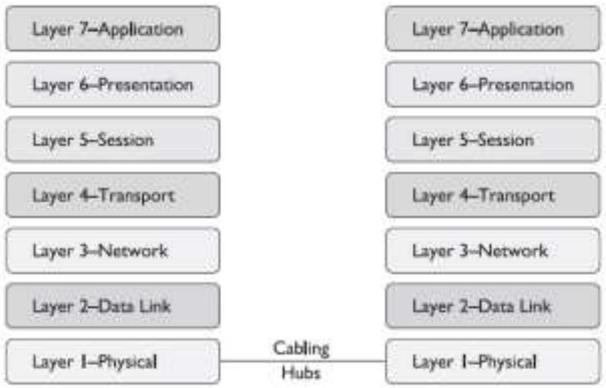


Figure 2.1 – The OSI seven layer model

The real magic of a network starts with the network interface card, or NIC which serves as the interface between the PC and the network. While NICs come in a wide array of shapes and sizes. Figure 1.3 shows a typical NIC.

On older systems, a NIC truly was a separate card that snapped into a handy expansion port, which is why they were called network interface cards. Even though they're now built into the motherboard, we still call them NICs.

When installed in a PC, the NIC looks like Figure 1.4. Note the cable running from the back of the NIC into the wall; inside that wall is another cable running all the way back to the hub.

Cabling and hubs define the Physical layer of the network, and NICs provide the interface to the PC. Figure 1.4 shows a diagram of the network cabling system.

The NIC

To understand networks, you must understand how NICs work. The network must provide a mechanism that gives each system a unique identifier—like a telephone number—so that data is delivered to the right system. That's one of the most important jobs of a NIC. Inside every NIC, burned onto some type of ROM chip, is special firmware containing a unique identifier with a 48-bit value called the media access control address, or MAC address.

Figure 1.3 – Typical NIC

No two NICs ever share the same MAC address—ever. Any company that makes NICs must contact the Institute of Electrical and Electronics Engineers (IEEE) and request a block of MAC addresses, which the company then burns into the ROMs on its NICs. Many NIC makers also print the MAC address on the surface of each NIC, as shown in Figure 1.5.

The MAC address in Figure 1.4 is 004005-607D49, although in print, we represent the MAC as 00-40-05-60-7D-49. The first six digits, in this example 00-40-05, represent the number of the manufacturer of the NIC. Once the IEEE issues to a manufacturer those six hex digits—often referred to as the organizationally unique identifier (OUI)—no other manufacturer may use them. The last six digits, in this example 60-7D-49, are the manufacturer's unique serial number for that NIC; this portion of the MAC is often referred to as the device ID.

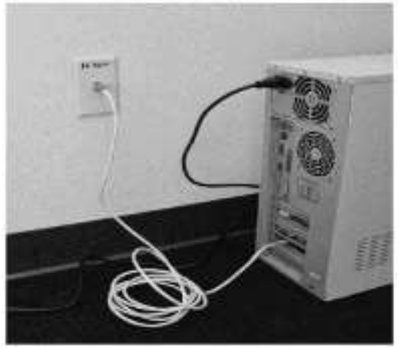


Figure 1.4 – A connected NIC

Every NIC in the world has a unique MAC address, recall that computer data is binary, which means it's made up of streams of ones and zeroes. NICs send and receive this binary data as pulses of electricity, light, or radio waves. The NICs that use electricity to send and receive data are the most common, so let's consider that type of NIC. The specific process by which a NIC uses electricity to send and receive data is exceedingly complicated. If you put an oscilloscope on the wire to measure voltage, you'd see something like Figure 1.6. An oscilloscope is a powerful microscope that enables you to see electrical pulses.

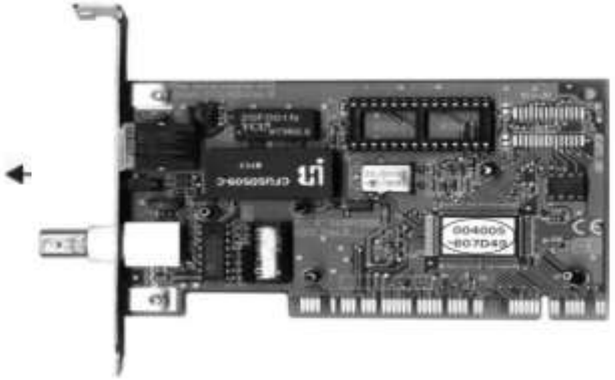
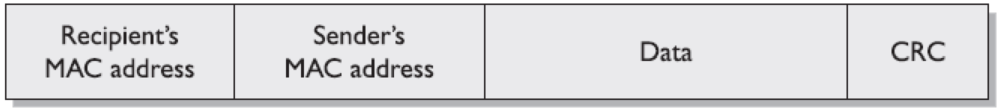


Figure 1.5 – MAC address

All networks transmit data by breaking whatever is moving across the physical layer (files, print jobs, Web pages, and so forth) into discrete chunks called frames. A frame is basically a container for a chunk of data moving across a network. The NIC creates and sends, as well as receives and reads, these frames.

Here's where the MAC address becomes important. Figure 2-17 shows a representation of a generic frame. Even though a frame is a string of ones and zeroes, we often draw frames as a series of rectangles, each rectangle representing a part of the string of ones and zeroes. Note that the frame begins with the MAC address of the NIC to which the data is to be sent, followed by the MAC address of the sending NIC. Then comes the data, followed by a special bit of checking information called the cyclic redundancy check (CRC) that the receiving NIC uses to verify that the data arrived intact.



Recipient's MAC address	Sender's MAC address	Data	CRC

Figure 1.7 - Generic frame

Most CRCs are only 4 bytes long, yet the average frame carries around 1500 bytes of data. Without going into the grinding details, think of the CRC as just the remainder of a division problem. The NIC sending the frame does a little math to make the CRC. Using binary arithmetic, it works a division problem on the data using a divisor called a key. This key is the same on all the NICs in your network—it's built in at the factory. The result of this division is the CRC. When the frame gets to the receiving NIC, it divides the data by the same key. If the receiving NIC's answer is the same as the CRC, it knows the data is good.

The data may be a part of a file, a piece of a print job, or part of a Web page. NICs aren't concerned with content. The NIC simply takes whatever data is passed to it via its device driver and addresses it for the correct system. Special software will take care of what data gets sent and what happens to that data when it arrives.

Like a canister, a frame can hold only a certain amount of data. Different networks use different sizes of frames, but generally, a single frame holds about 1500 bytes of data. The sending system's software must chop the data up into nice, frame-sized chunks, which it then hands to the NIC for sending. As the receiving system begins to accept the incoming frames, it's up to the receiving system's software to recombine the data chunks as they come in from the network.

When a system sends a frame out on the network, the frame goes into the hub. The hub, in turn, makes an exact copy of that frame, sending a copy of the original frame to every other system on the network. The interesting part of this process is when the copy of the frame comes into all the other systems. Only the NIC to which the frame is addressed will process that frame—the other NICs simply erase it when they see that it is not addressed to their MAC address. Every frame sent on a network is received by every NIC, but only the NIC with the matching MAC address will process that particular frame (Figure 2-19).

The process of getting data onto the wire and then picking that data off the wire is amazingly complicated. For instance, what happens to keep two NICs from speaking at the same time? Because all the data sent by one NIC is read by every other NIC on the network, only one system may speak at a time. Networks use frames to restrict the amount of data a NIC can send at once, giving all NICs a chance to send data over the network in a reasonable span of time. Dealing with this and many other issues requires sophisticated electronics, but the NICs handle these issues completely on their own.

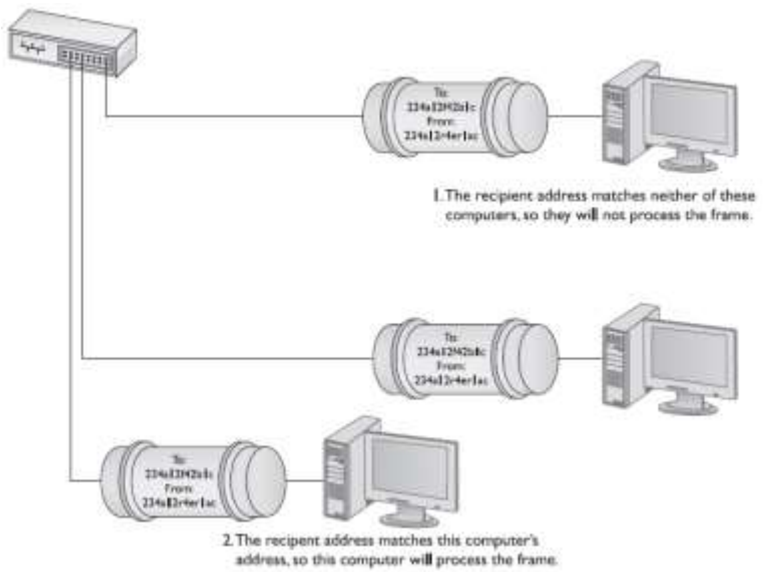

To:
234a1342b1c
From:
234a12r4er1ac
1. The recipient address matches neither of these computers, so they will not process the frame.
To:
234a1342b1c
From:
234a12r4er1ac
To:
234a1342b1c
From:
234a12r4er1ac
2. The recipent address matches this computer's address, so this computer will process the frame.

Figure 1.8 –Incoming Frame

Using the MAC address is a great way to move data around, but this process raises an important question. How does a sending NIC know the MAC address of the NIC to which it's sending the data? In most cases, the sending system already knows the destination MAC address, because the NICs had probably communicated earlier, and each system stores that data. If it doesn't already know the MAC address, a NIC may send a broadcast onto the network to ask for it. The MAC address of FF-FF-FF-FF-FF-FF is the broadcast address—if a NIC sends a frame using the broadcast address, every single NIC on the network will process that frame. That broadcast frame's data will contain a request for a system's MAC address. The system with the MAC address your system is seeking will read the request in the broadcast packet and respond with its MAC address.

- First, the sending system network operating system (NOS) software hands some data to its NIC. The NIC begins building a frame to transport that data to the receiving NIC.
- After the NIC creates the frame, it adds the CRC, and then dumps it and the data into the frame.
- Next, the NIC puts both the destination MAC address and its own MAC address onto the frame. It waits until no other NIC is using the cable, and then sends the frame through the cable to the network.

The frame propagates down the wire into the hub, which creates copies of the frame and sends it to every other system on the network. Every NIC receives the frame and checks the MAC address. If a NIC finds that a frame is addressed to it, it processes the frame; if the frame is not addressed to it, the NIC erases it.

So, what happens to the data when it gets to the correct NIC? First, the receiving NIC uses the CRC to verify that the data is valid. If it is, the receiving NIC strips off all the framing information and sends the data to the software—the network operating system—for processing. The receiving NIC doesn't care what the software does with the data; its job stops the moment it passes on the data to the software.

Any device that deals with a MAC address is part of the OSI Data Link layer. Let's update the OSI model to include details about the Data Link layer (Figure 1.9). Remember that the cabling and the hub are located in the Physical layer. The NIC is in the Data Link layer, but spans two sublayers.

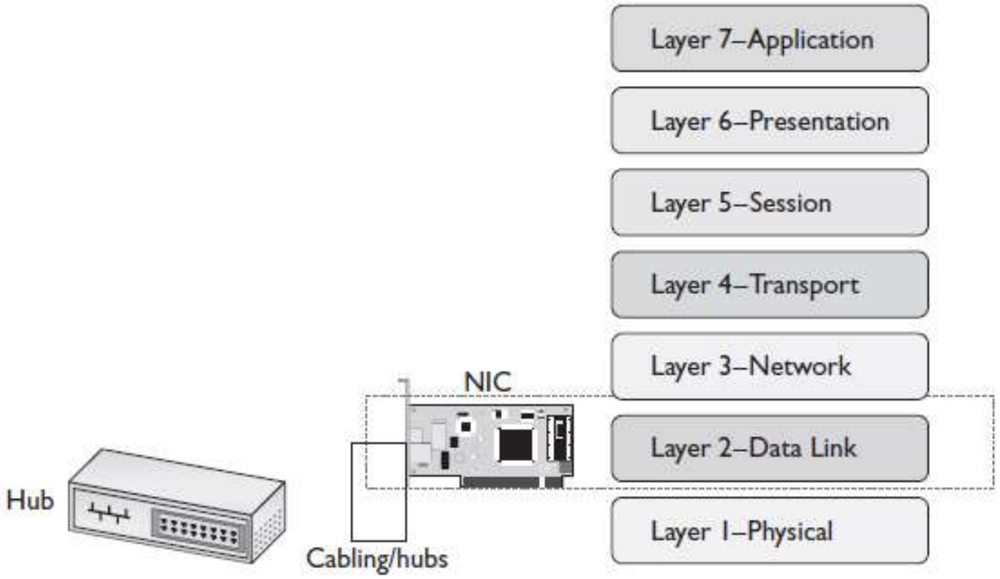


Figure 1.9 -Layer 1 and Layer 2 are now properly applied to the network

The Two Aspects of NICs
On one end, frames move into and out of the NIC's network cable connection. On the other end, data moves back and forth between the NIC and the network operating system software. The many steps a NIC performs to keep this data moving—sending and receiving frames over

the wire, creating outgoing frames, reading incoming frames, and attaching MAC addresses—are classically broken down into two distinct jobs.

The first job is called the Logical Link Control (LLC). The LLC is the aspect of the NIC that talks to the operating system, places data coming from the software into frames, and creates the CRC on each frame. The LLC is also responsible for dealing with incoming frames: processing those that are addressed to this NIC and erasing frames addressed to other machines on the network.

The second job is called the Media Access Control (MAC), it remembers the NIC's own MAC address and handles the attachment of MAC addresses to frames. Remember that each frame the LLC creates must include both the sender's and recipient's MAC addresses. The MAC also ensures that the frames, now complete with their MAC addresses, are then sent along the network cabling.

NICs solely in the Data Link layer is the obvious other duty of the NIC—putting the ones and zeroes on the network cable.

Network Software and Layers 3-7

Getting data from one system to another in a simple network (defined as one in which all the computers connect to one hub) takes relatively little effort on the part of the NICs. But one problem with simple networks is that computers need to broadcast to get MAC addresses. It works for small networks, but what happens when the network gets big, like the size of the entire Internet? Just imagine millions of computers all broadcasting. No data could get through. When networks get large, you can't use the MAC addresses anymore. Large networks need a logical addressing method that no longer cares about the hardware and enables us to break up the entire large network into smaller networks called subnets. Figure 1.10 shows two ways to set up a network. On the left, all the computers connect to a single hub. On the right, however, the LAN is separated into two five-computer subnets.

To move past the physical MAC addresses and start using logical addressing requires some special software, usually called a network protocol. Network protocols exist in every operating system. A network protocol not only has to create unique identifiers for each system, but must also create a set of communication rules for issues like how to handle data chopped up into multiple packets, and how to make sure that those packets get from one subnet to another. Let's take a moment to learn a bit about the most famous network protocol—TCP/IP—and its unique universal addressing system.

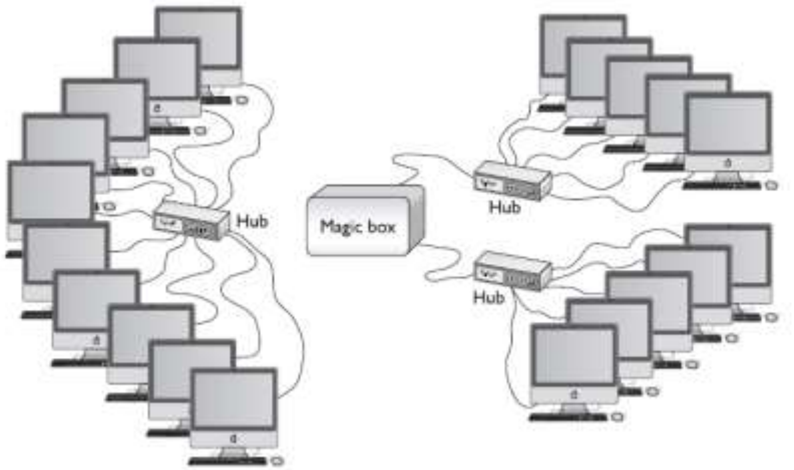


Figure 1.10 - Large LAN complete (left) and broken up into two subnets (right)

To be accurate, TCP/IP is really several network protocols designed to work together— but two protocols, TCP and IP, do so much work the folks who invented all these protocols named the whole thing TCP/IP. TCP stands for Transmission Control Protocol, and IP stands for Internet Protocol..

IP - Layer 3, the Network Layer

The IP protocol is the primary protocol that TCP/IP uses at Layer 3 (Network) of the OSI model. The IP protocol makes sure that a piece of data gets to where it needs to go on the network. It does this by giving each device on the network a unique numeric identifier called an IP address. An IP address is known as a logical address to distinguish it from the physical address, the MAC address of the NIC.

Every network protocol uses some type of naming convention, but no two protocols use the same convention. IP uses a rather unique dotted decimal notation (sometimes referred to as a dotted-quad numbering system or dotted-octet numbering system) based on four 8-bit numbers. Each 8-bit number ranges from 0 to 255, and the four numbers are separated by periods. A typical IP address might look like this: 192.168.4.232.
No two systems on the same network share the same IP address; if two machines accidentally receive the same address, they won't be able to send or receive data. These IP addresses must be configured by the end user (or the network administrator).

Take a look at Figure 1.10. What makes logical addressing powerful are the magic boxes—called routers—that separate each of the subnets. Routers work like a hub, but instead of forwarding packets by MAC address they use the IP address. Routers enable you to take one big network and chop it up into smaller networks. Routers also have a second, very important feature. They enable you to connect networks with different types of cabling or frames. Figure 1.11 shows a typical router. This router enables you to connect a network that uses MAC addresses—a small subnet—to a cable modem network.

Figure 1.11 - Typical router

What's important here is understanding that in a TCP/IP network, each system has two unique identifiers: the MAC address and the IP address. The MAC address (the physical address) is literally burned into the chips on the NIC, while the IP address (the logical address) is simply stored in the software of the system. MAC addresses come with the NIC, so we don't configure MAC addresses, whereas we must configure IP addresses through software. Figure 1.12 shows a network diagram with the MAC and IP addresses displayed for each system.

This two-address system enables IP networks to do something really cool and powerful: using IP addresses, systems can send each other data without regard to the physical connection. This capability requires more than the simple assignment of an IP address for each computer. The network protocol must also know where to send the frame, no matter what type of hardware the various computers are running. A network protocol also uses frames.

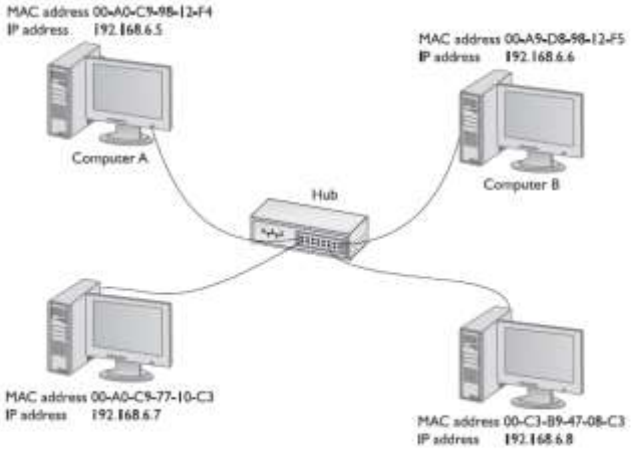

MAC address 00-A0-C9-98-12-F4
IP address 192.168.6.5
Computer A
MAC address 00-A9-D8-98-12-F5
IP address 192.168.6.6
Computer B
Hub
MAC address 00-A0-C9-77-10-C3
IP address 192.168.6.7
MAC address 00-C3-B9-47-08-C3
IP address 192.168.6.8

Figure 1.12 – A network addressing

Anything that has to do with logical addressing works at the OSI Network layer. At this point there are only two items that operate at the Network layer—routers and the part of the network protocol on every computer that understands the logical addressing (Figure 1.13).

Visualize the network protocol software as a layer between the system's software and the NIC. When the IP network protocol gets hold of data coming from your system's software, it places its own frame around that data. We call this inner frame an IP packet, so it won't be confused with the frame that the NIC will add later. Instead of adding MAC addresses to its packet, the network protocol adds sending and receiving IP addresses. Figure 1.14 shows a typical IP packet; notice the similarity to the frames you saw earlier.

But IP packets don't leave their PC home naked. Each IP packet is handed to the NIC, which then encloses the IP packet in a regular frame, creating, in essence, a packet within a frame. A more conventional drawing would look like Figure 1.15.

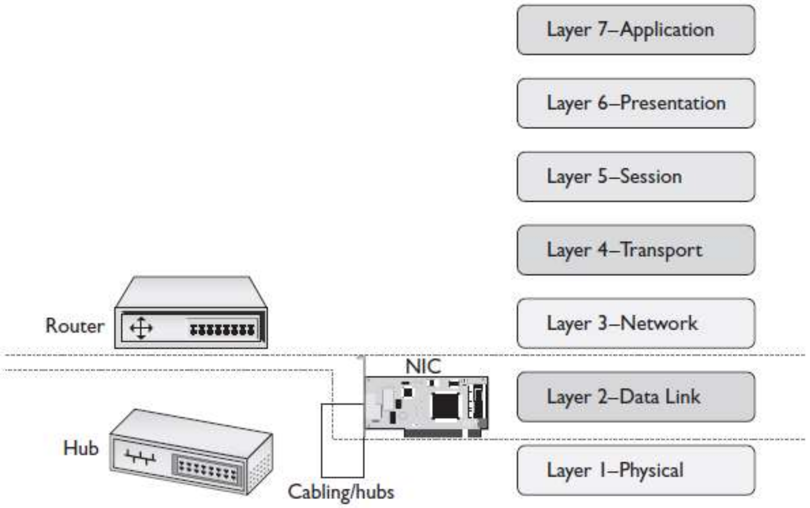


Figure 1.13 - Router now added to the OSI model for the network

Data type	Packet Count	Recipient's IP address	Sender's IP address	Data

Figure 1.14 - IP packet

Frame	Packet	Data	CRC

Figure 1.15 - IP packet in

The router has two connections (See figure 1.16). One is just a built-in NIC that runs from the router to the hub. The other connection links the router to a cable modem. Cable networks don't use MAC addresses. They use their own type of frame that has nothing to do with MAC addresses. If you tried to send a regular network frame on a cable modem network, it wouldn't work. When a router receives an IP packet inside a frame added by a NIC, it peels off that frame and replaces it with the type of frame the cable network needs (Figure 1.17).

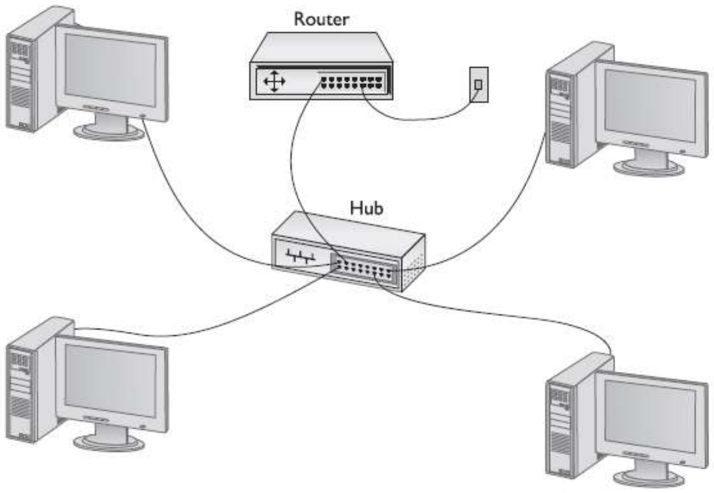


Figure 1.16 - Adding a router to the network

Once the network frame is gone, so are the MAC addresses! Thus, you need some other naming system the router can use to get the data to the right computer—and that's why you use IP addresses on a network. After the router strips off the MAC addresses and puts on whatever type of addressing used by the cable modem network, the frame flies through the cable modem network, using the IP address to guide the frame to the router connected to the receiving system. At this point, the process reverses. The router rips off the cable modem frame, adds the MAC address for the receiving system, and sends it on the network, where the receiving system picks it up (Figure 1.18).

The receiving NIC strips away the MAC address header information and passes the remaining packet off to the software. The networking software built into your operating system handles all the rest of the work. The NIC's driver software is the interconnection between the hardware and the software. The NIC driver knows how to communicate with the NIC to send and receive frames, but it can't do anything with the packet. Instead, the NIC driver hands the packet off to other programs that know how to deal with all the separate packets and turn them into Web pages, e-mail messages, files, and so forth.

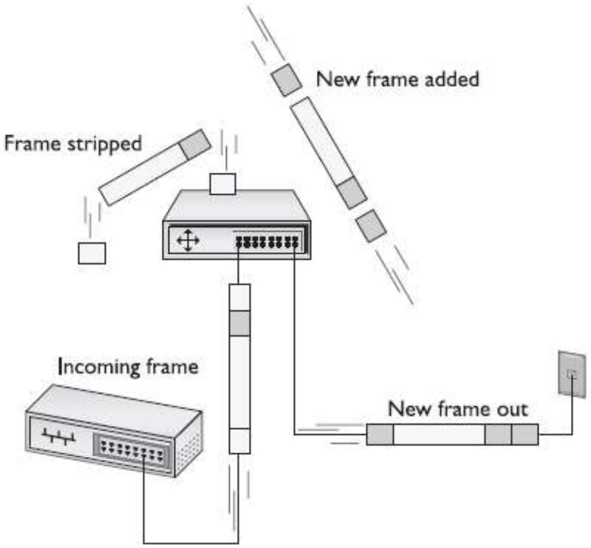


Figure 1.17 - Router removing network frame and adding one for the cable line

The Network layer is the last layer that deals directly with hardware. All the other layers of the OSI seven-layer model work strictly within software.

Layer 4, the Transport Layer

Because most chunks of data are much larger than a single frame, they must be chopped up before they can be sent across a network. When a serving computer receives a request for some data, it must be able to chop the requested data into chunks that will fit into a packet (and eventually into the NIC's frame), organize the packets for the benefit of the receiving system, and hand them to the NIC for sending. The receiving system must be able to recognize a series of incoming packets as one data transmission, reassemble the packets correctly based on information included in the packets by the sending system, and verify that all the packets for that piece of data arrived in good shape.

The network protocol breaks up the data into packets and gives each packet some type of sequence number. Embedded into the data of each packet is a sequencing number. By reading the sequencing numbers, the receiving system knows both the total number of packets and how to put them back together.

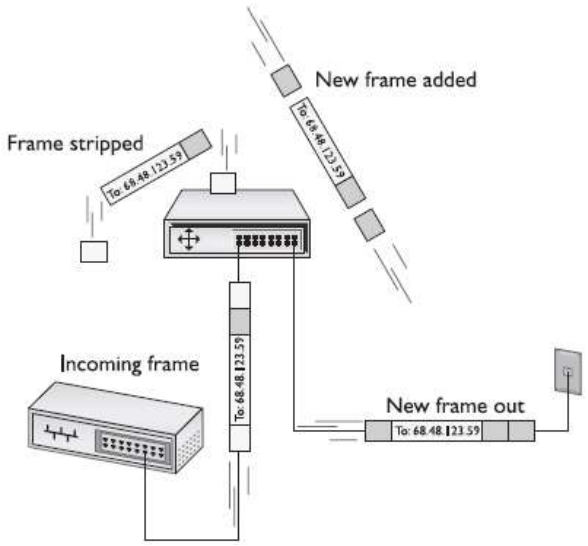


Figure 1.18 - Router in action

Layer 4, the Transport layer of the OSI seven-layer model, has only one big job: it's the assembler/disassembler software. As part of its job, the Transport layer also initializes requests for packets that weren't received in good order (Figure 1.19).

Layer 5, the Session Layer

In a network, any one system may be talking to many other systems at any given moment. Additionally, the operating system must enable one system to make a connection to another system to verify that the other system can handle whatever operation the initiating system wants to perform. If user A wants to send a print job to another user B’s printer, it first contacts A’s system to ensure that it is ready to handle the print job. The session software handles this part of networking.

Layer 5, the Session layer of the OSI seven-layer model, handles all the sessions for a system. The Session layer initiates sessions, accepts incoming sessions, and opens and closes existing sessions. The Session layer also keeps track of computer naming conventions, such as calling your computer SYSTEM01 or some other type of name that makes more sense than an IP or MAC address (Figure 1.20).

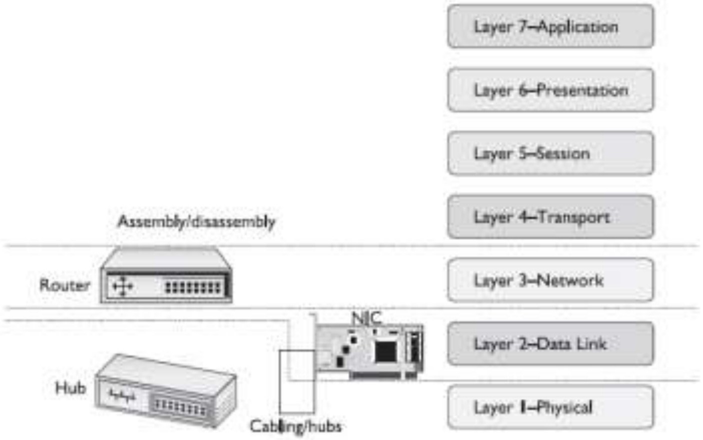


Figure 1.19 – OSI updated

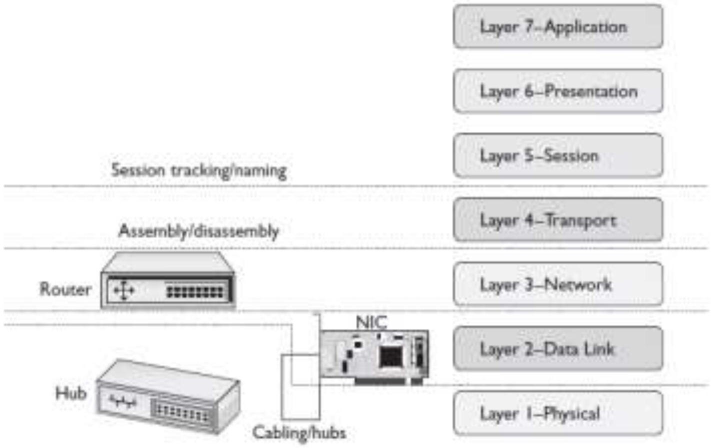


Figure 1.20 – OSI updated

Layer 6, the Presentation Layer

One of the most powerful aspects of a network lies in the fact that it works with (almost) any operating system. Today's networks easily connect, for example, a Macintosh system to a Windows PC, despite the fact that these different operating systems use different formats for many types of data. Different data formats used to drive us crazy back in the days before word processors (like Microsoft Word) could import or export a thousand other word processor formats (Figure 1.20).

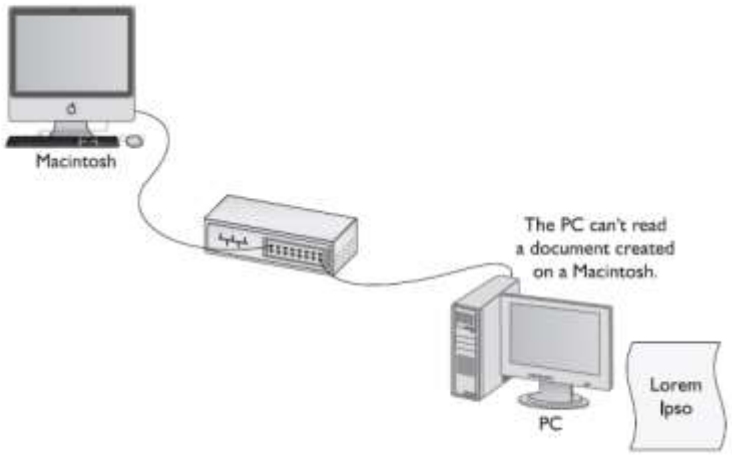


Figure 1.20 – Different data formats were often unreadable between systems

This created the motivation for standardized formats that anyone—at least with the right program—could read from any type of computer. Specialized file formats, such as Adobe's popular Portable Document Format (PDF) for documents and PostScript for printing, provide standard formats that any system, regardless of the operating system, can read, write, and edit.

Layer 6, the Presentation layer of the OSI seven-layer model, handles converting data into formats that are readable by the system. Of all the OSI layers, the high level of standardization of file formats has made the Presentation layer the least important and least used (Figure 1.21).

Layer 7, the Application Layer

The last, and most visible, part of any network is the software applications that use it. If you want to copy a file residing on another system in your network, you need an application like Network in Windows 7 (or My Network Places in earlier versions of Windows) that enables you to access files on remote systems. If you want to view Web pages, you need a Web browser like Internet Explorer or Mozilla Firefox. The people who use a network experience it through an application. A user who knows nothing about all the other parts of a network may still know how to open an e-mail application to retrieve mail.

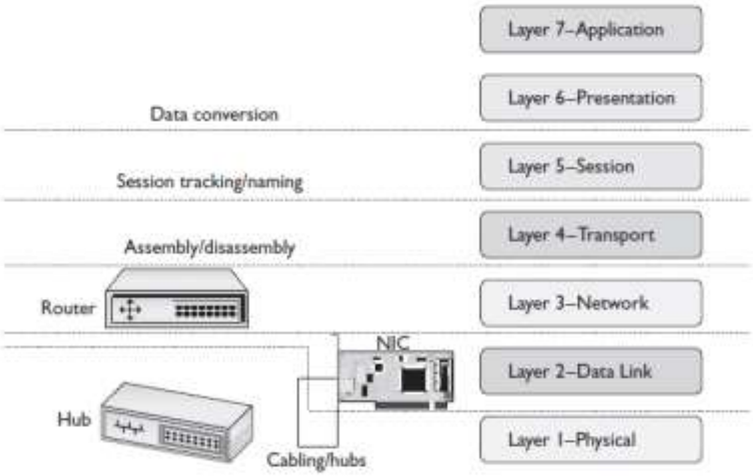


Figure 1.21 – OSI updated

Applications may include a number of additional functions, such as encryption, user authentication, and tools to control the look of the data. But these functions are specific to the given applications. In other words, if you want to put a password on your Word document, you must use the password functions of Word to do so.

Layer 7, the Application layer of the OSI seven-layer model, refers to the code built into all operating systems that enables network-aware applications. All operating systems have Application Programming Interfaces (APIs) that programmers can use to make their programs network aware (Figure 2-45). An API in general provides a standard way for programmers to enhance or extend an application's capabilities.

The OSI seven-layer model provides you with a way to conceptualize a network to determine what could cause a specific problem when the inevitable problems occur. Users don't need to know anything about this, but techs can use the OSI model for troubleshooting.

By understanding how network traffic works throughout the model, you can troubleshoot with efficiency. You can use the OSI model during your career as a network tech as the basis for troubleshooting.

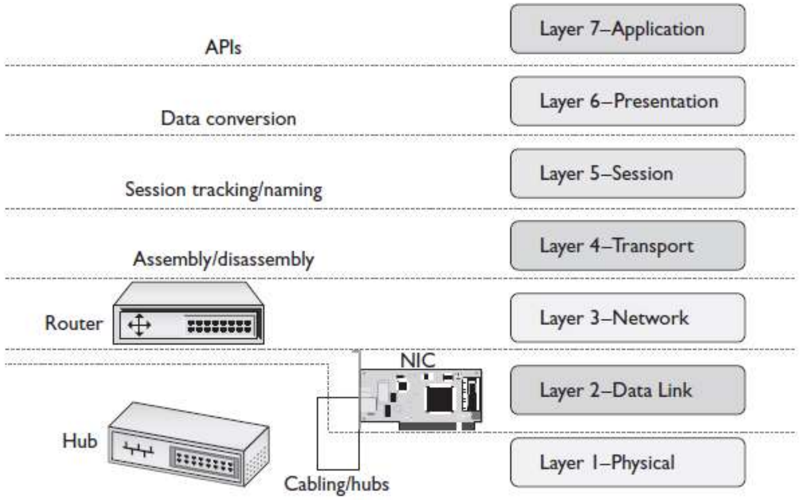


Figure 1.22 – OSI updated