**PROG8165 – Assignment 1**

Student name: Tri Pham

Student ID: 8060220

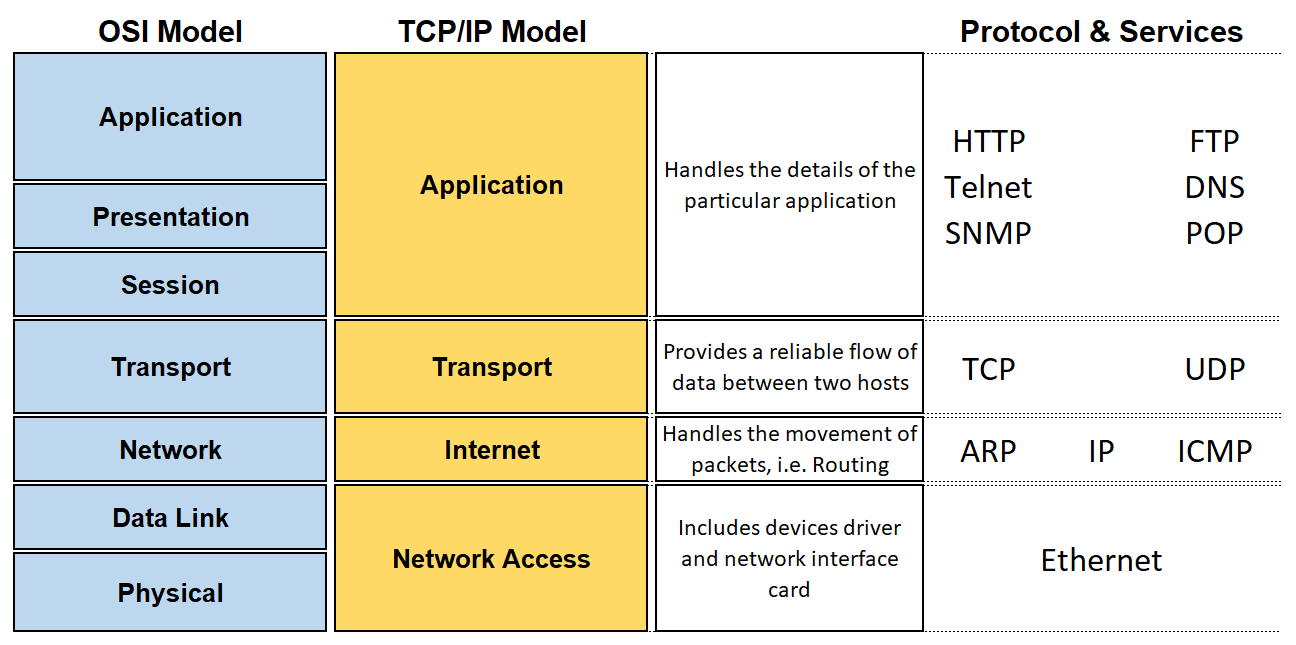
Date of submission:

1. **Using Diagrams/written text do a comparative analysis between the OSI model and the TCP/IP model.** 
   1. **Key Differences between TCP/IP and OSI Model**

|  |  |
| --- | --- |
| **OSI (Open System Interconnection)** | **TCP/IP (Transmission Control Protocol / Internet Protocol)** |
| 1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user. | 1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network. |
| 2. In OSI model the transport layer guarantees the delivery of packets. | 2. In TCP/IP model the transport layer does not guarantees delivery of packets. Still the TCP/IP model is more reliable. |
| 3. Follows vertical approach. | 3. Follows horizontal approach. |
| 4. OSI model has a separate Presentation layer and Session layer. | 4. TCP/IP does not have a separate Presentation layer or Session layer. |
| 5. Transport Layer is Connection Oriented. | 5. Transport Layer is both Connection Oriented and Connection less. |
| 6. Network Layer is both Connection Oriented and Connection less. | 6. Network Layer is Connection less. |
| 7. OSI is a reference model around which the networks are built. Generally it is used as a guidance tool. | 7. TCP/IP model is, in a way implementation of the OSI model. |
| 8. Network layer of OSI model provides both connection oriented and connectionless service. | 8. The Network layer in TCP/IP model provides connectionless service. |
| 9. OSI model has a problem of fitting the protocols into the model. | 9. TCP/IP model does not fit any protocol |
| 10. Protocols are hidden in OSI model and are easily replaced as the technology changes. | 10. In TCP/IP replacing protocol is not easy. |
| 11. OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent. | 11. In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent. |
| 12. It has 7 layers | 12. It has 4 layers |

* 1. **Key similarities between TCP/IP and OSI Model**

1. Similar architecture: Both the models share the similar or equal architecture and it can be defined by the fact that both of them are constructed with layers.
2. Share common Application layer: Both models share a common “application layer”. However, this layer contains different services, which depends upon each model.
3. Both the models have comparable “transport” & “network” layers: this can be defined by the fact that whatever “functions” are performed b/w the presentation and network layer of OSI model similar or equal function are performed at the “transport layer” of the TCP/IP model.



1. **Describe using diagrams/written text the Transmission Control Protocol in detail, i.e. the use of ports, the TCP header, checksum, etc.**

#### **TCP**

In the TCP/IP protocol suite, TCP is the intermediate layer between IP below it, and an application above it. Using TCP, applications on networked hosts can establish reliable connections to one another. The protocol guarantees in-order delivery of data from the sender to the receiver.

* 1. **Basic Protocol Operation**

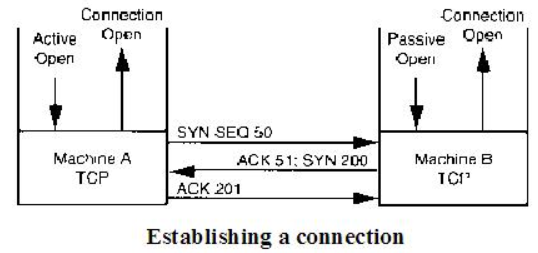
The Transmission Control Protocol is connection-oriented, meaning user data is not exchanged between TCP peers until a connection is established between the two end points. This connection exists for the duration of the data transmission.

TCP connections have three phases:

1. Connection establishment
2. Data transfer
3. Connection termination
   1. **Connection Establishment**

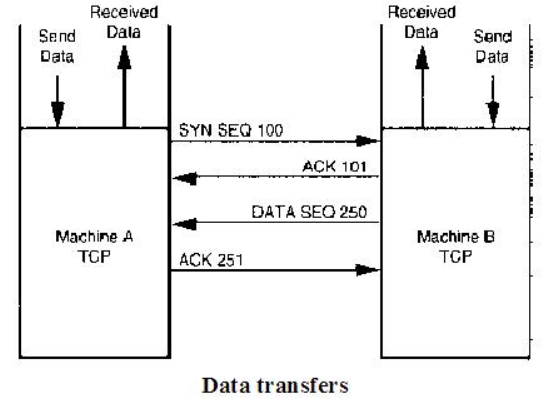
To establish a connection, TCP uses a 3-way handshake. Before a client attempts to connect with a server, the server must first bind to a port to open it up for connections. This is called a passive open. Once the passive open is established, a client may initiate an active open. The server then sends an acknowledgement to the client. At this point, both the client and server have received an acknowledgement of the connection.

Figure below shows a flow diagram for a TCP open. The process begins with Machine A's TCP receiving a request for a connection from its ULP, to which it sends an active open primitive to Machine B. The segment that is constructed will have the SYN flag set on (set to 1) and will have a sequence number assigned. The diagram shows this with the notation SYN SEQ 50 indicating that the SYN flag is on and the sequence number (Initial Send Sequence number or ISS) is 50. (Any number could have been chosen.)



* 1. **Data Transfer**

Transferring information is straightforward, as shown in Figure below. For each block of data received by Machine A's TCP from the ULP, TCP encapsulates it and sends it to Machine B with an increasing sequence number. After Machine B receives the message, it acknowledges it with a segment acknowledgment that increments the next sequence number (and hence indicates that it received everything up to that sequence number). Figure shows the transfer of only one segment of information - one each way.

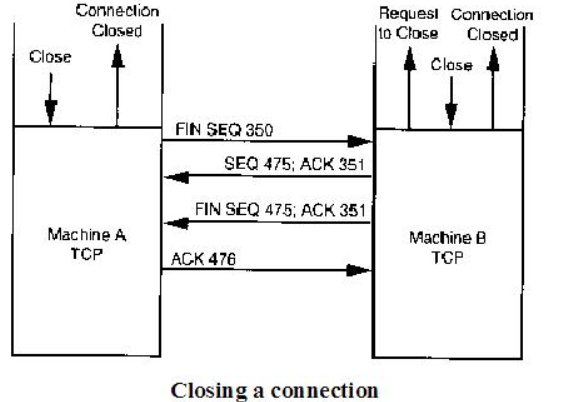


* 1. **Connection Termination**

The connection termination phase uses, at most, a four-way handshake, with each side of the connection terminating independently. When an end point wishes to stop its half of the connection, it transmits a special packet with a flag indicating it is finished. The other end acknowledges the flag. A typical connection termination includes this two-phase handshake from both ends of the connection.

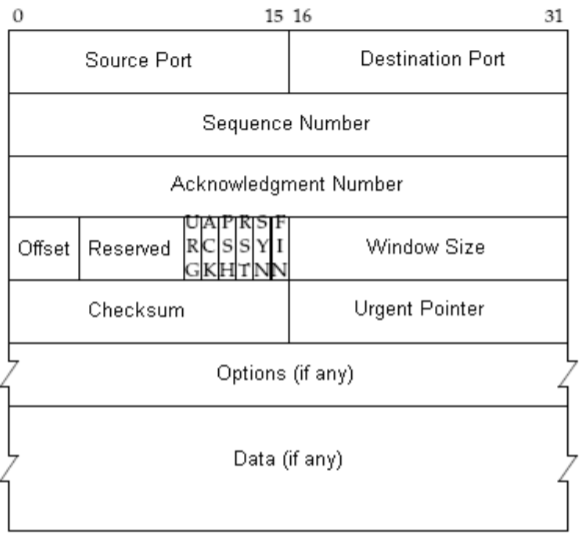
This is shown in Figure below. In the figure, Machine A's TCP sends the request to close the connection to Machine B with the next sequence number. Machine B will then send back an acknowledgment of the request and its next sequence number. Following this, Machine B sends the close message through its ULP to the application and waits for the application to acknowledge the closure. This step is not strictly necessary; TCP can close the connection without the application's approval, but a well-behaved system would inform the application of the change in state.

After receiving approval to close the connection from the application (or after the request has timed out), Machine B's TCP sends a segment back to Machine A with the FIN flag set. Finally, Machine A acknowledges the closure and the connection is terminated.



* 1. **TCP Packet Structure**

A TCP packet consists of two sections, header and data. All fields may not be used in every transmission. A flag field is used to indicate the type of transmission the packet represents and how the packet should be interpreted.



The header consists of 11 fields, of which 10 are required:

• **Source port**—Identifies the sending application.

• **Destination po**rt—Identifies the destination application.

• **Sequence number**—Used for assembling segmented data in the proper order at the receiving end.

• **Acknowledgement number**—The sequence number the sender (the receiving end) expects next.

• **Data offset**—The size of the TCP header, it is also the offset from the start of the TCP packet to the data portion.

• **Reserved**—Reserved for future use, should be set to zero.

• **Flags** (also known as control bits)—contains 6 1-bit flags:

- URG—Urgent pointer field is significant.

- ACK—Acknowledgement field is significant.

- PSH—Push function.

- RST—Reset the connection.

- SYN—Synchronize sequence numbers.

- FIN—No more data from sender.

• **Window**—The number of bytes the sender is willing to receive starting from the acknowledgement field value.

• **Checksum**—used for error-checking of the header and data.

1. **Describe using diagrams/written text the Internet Protocol in detail, i.e. Subnets, IP header, IP Addressing, etc.**

## **The Internet Protocol (IP)**

IP stands for Internet Protocol and describes a set of standards and requirements for creating and transmitting data packets, or datagrams, across networks. The Internet Protocol (IP) is part of the Internet layer of the Internet protocol suite. In the OSI model, IP would be considered part of the network layer. IP is traditionally used in conjunction with a higher-level protocol, most notably TCP. The IP standard is governed by RFC 791.

#### **How IP Works**

IP is designed to work over a dynamic network. This means that IP must work without a central directory or monitor, and that it cannot rely upon specific links or nodes existing. IP is a connectionless protocol that is datagram-oriented., so each packet must contain the source IP address , destination IP address, and other data in the header to be successfully delivered.

Combined, these factors make IP an unreliable, best effort delivery protocol. Error correction is handled by upper level protocols instead. These protocols include TCP, which is a connection-oriented protocol, and UDP, which is a connectionless protocol.

Most internet traffic is TCP/IP

## **IP Versions**

There are two versions of IP in use today, IPv4 and IPv6. The original IPv4 protocol is still used today on both the internet, and many corporate networks. However, the IPv4 protocol only allowed for 232addresses. This, coupled with how addresses were allocated, led to a situation where there would not be enough unique addresses for all devices connected to the internet.

IPv6 was developed by the Internet Engineering Task Force (IETF), and was formalized in 1998. This upgrade substantially increased the available address space and allowed for 2128 addresses. In addition, there were changes to improve the efficiency of IP packet headers, as well as improvements to routing and security.

## **What is an IP address?**

An IP address (internet protocol address) is a numerical representation that uniquely identifies a specific interface on the network. Addresses in IPv4 are 32-bits long. This allows for a maximum of 4,294,967,296 (232) unique addresses. Addresses in IPv6 are 128-bits, which allows for 3.4 x 1038 (2128) unique addresses.

The total usable address pool of both versions is reduced by various reserved addresses and other considerations.

IP addresses are binary numbers but are typically expressed in decimal form (IPv4) or hexadecimal form (IPv6) to make reading and using them easier for humans.

#### **Subnet Masks**

A single IP address identifies both a network, and a unique interface on that network. A subnet mask can also be written in dotted decimal notation and determines where the network part of an IP address ends, and the host portion of the address begins.

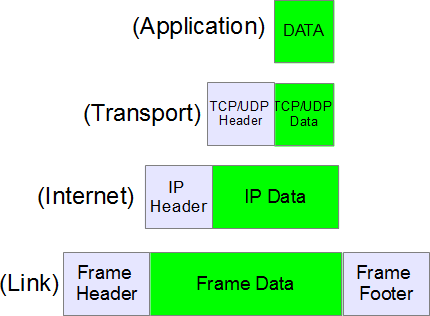
When expressed in binary, any bit set to one means the corresponding bit in the IP address is part of the network address. All the bits set to zero mark the corresponding bits in the IP address as part of the host address.

The bits marking the subnet mask must be consecutive ones. Most subnet masks start with 255. and continue on until the network mask ends. A Class C subnet mask would be 255.255.255.0.

#### **IP Address Classes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Leading bits** | **Size of *network number*bit field** | **Size of *rest* bit field** | **Number of networks** | **Addresses per network** | **Total addresses in class** | **Start address** | **End address** |
| Class A | 0 | 8 | 24 | 128 (27) | 16,777,216 (224) | 2,147,483,648 (231) | 0.0.0.0 | 127.255.255.255 |
| Class B | 10 | 16 | 16 | 16,384 (214) | 65,536 (216) | 1,073,741,824 (230) | 128.0.0.0 | 191.255.255.255 |
| Class C | 110 | 24 | 8 | 2,097,152 (221) | 256 (28) | 536,870,912 (229) | 192.0.0.0 | 223.255.255.255 |
| Class D (multicast) | 1110 | not defined | not defined | not defined | not defined | 268,435,456 (228) | 224.0.0.0 | 239.255.255.255 |
| Class E (reserved) | 1111 | not defined | not defined | not defined | not defined | 268,435,456 (228) | 240.0.0.0 | 255.255.255.255 |

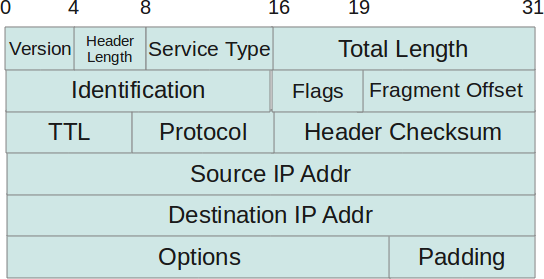
### **IP Layer in TCP/IP Suite**



As can be seen from the image above, the IP protocol sits at the layer-2 of TCP/IP protocol suite ie the Internet layer. Another point worth noting here is how the data is packed in TCP/IP suite. If we analyze the figure above, we see :

* The application layer sends the data (to be transferred to remote destination ) to the transport layer.
* The transport layer puts its header in the beginning and sends this complete packet (TCP-header + app-data) to the IP layer.
* On the same lines, The IP layer puts its header in front of the data received from TCP (Note that data received from TCP = TCP-header + app-data).
* So now the structure of IP datagram becomes IP-header + TCP-header + app-data.
* This IP datagram is passed to the ethernet layer which on the same lines adds its own header to IP datagram and then the whole packet is transmitted over network.

### **IP Header**



* **Protocol Version(4 bits)** : This is the first field in the protocol header. This field occupies 4 bits. This signifies the current IP protocol version being used. Most common version of IP protocol being used is version 4 while version 6 is out in market and fast gaining popularity.
* ***Header Length(4 bits)***: This field provides the length of the IP header. The length of the header is represented in 32 bit words. This length also includes IP options (if any). Since this field is of 4 bits so the maximum header length allowed is 60 bytes. Usually when no options are present then the value of this field is 5. Here 5 means five 32 bit words ie 5 \*4 = 20 bytes.
* **Type of service(8 bits)** : The first three bits of this field are known as precedence bits and are ignored as of today. The next 4 bits represent type of service and the last bit is left unused. The 4 bits that represent TOS are : minimize delay, maximize throughput, maximize reliability and minimize monetary cost.
* **Total length (16 bits)**: This represents the total IP datagram length in bytes. Since the header length (described above) gives the length of header and this field gives total length so the length of data and its starting point can easily be calculated using these two fields. Since this is a 16 bit field and it represents length of IP datagram so the maximum size of IP datagram can be 65535 bytes. When IP fragmentation takes place over the network then value of this field also changes. There are cases when IP datagrams are very small in length but some data links like ethernet pad these small frames to be of a minimum length ie 46 bytes. So to know the exact length of IP header in case of ethernet padding this field comes in handy.
* **Identification(16 bits)**: This field is used for uniquely identifying the IP datagrams. This value is incremented every-time an IP datagram is sent from source to the destination. This field comes in handy while reassembly of fragmented IP data grams.
* **Flags(3 bits)**: This field comprises of three bits. While the first bit is kept reserved as of now, the next two bits have their own importance. The second bit represents the ‘Don’t Fragment’ bit. When this bit is set then IP datagram is never fragmented, rather its thrown away if a requirement for fragment arises. The third bit represents the ‘More Fragment’ bit. If this bit is set then it represents a fragmented IP datagram that has more fragments after it. In case of last fragment of an IP datagram this bit is not set signifying that this is the last fragment of a particular IP datagram.
* **Fragment offset(13 bits)**: In case of fragmented IP data grams, this field contains the offset( in terms of 8 bytes units) from the start of IP datagram. So again, this field is used in reassembly of fragmented IP datagrams.
* **Time to live(8 bits)**: This value represents number of hops that the IP datagram will go through before being discarded. The value of this field in the beginning is set to be around 32 or 64 (lets say) but at every hop over the network this field is decremented by one. When this field becomes zero, the data gram is discarded. So, we see that this field literally means the effective lifetime for a datagram on network.
* **Protocol(8 bits)**: This field represents the transport layer protocol that handed over data to IP layer. This field comes in handy when the data is demultiplex-ed at the destination as in that case IP would need to know which protocol to hand over the data to.
* **Header Checksum(16 bits)**: This fields represents a value that is calculated using an algorithm covering all the fields in header (assuming this very field to be zero). This value is calculated and stored in header when IP data gram is sent from source to destination and at the destination side this checksum is again calculated and verified against the checksum present in header. If the value is same then the datagram was not corrupted else its assumed that data gram was received corrupted. So this field is used to check the integrity of an IP datagram.
* **Source and destination IP*(32 bits each)***: These fields store the source and destination address respectively. Since size of these fields is 32 bits each so an IP address os  maximum length of 32 bits can be used. So we see that this limits the number of IP addresses that can be used. To counter this problem, IP V6 has been introduced which increases this capacity.
* **Options(Variable length)**: This field represents a list of options that are active for a particular IP datagram. This is an optional field that could be or could not be present. If any option is present in the header then the first byte is represented as follows :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| copy flag | option class | | option num | | | | |

* In the description above, the ‘copy flag’ means that copy this option to all the fragments in case this IP datagram gets fragmented. The ‘option class’ represents the following values : 0 -> control, 1-> reserved, 2 -> debugging and measurement, and 3 -> reserved. Some of the options are given below :

|  |  |  |  |
| --- | --- | --- | --- |
| **class** | **number** | **length** | **description** |
| 0 | 0 | – | end of option list |
| 0 | 1 | – | no operation |
| 0 | 2 | 11 | security |
| 0 | 3 | var. | loose source routing |
| 0 | 9 | var. | strict source routing |
| 0 | 7 | var. | record route |
| 0 | 8 | 4 | stream id |
| 2 | 4 | var. | INTERNET time stamp |

* **Data**: This field contains the data from the protocol layer that has handed over the data to IP layer. Generally this data field contains the header and data of the transport layer protocols. Please note that each TCP/IP layer protocol attaches its own header at the beginning of the data it receives from other layers in case of source host and in case of destination host each protocol strips its own header and sends the rest of the data to the next layer.

1. **Examine the different Protocols that are used in various layers of the models (UDP, ARP, SMTP, HTTP, PPP etc) pick at least 10 different ones and using simple diagrams/written text describe these succinctly, no more than one to three sentences each.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Application layer** | | | |
|  | Protocol | Common Port | Description |
|  | HTTP | 80 | Hypertext Transfer Protocol is often called the protocol of the Internet.  HTTP received this designation because most Internet traffic is based on HTTP.  When a user requests a Web resource, it is requested using HTTP. |
|  | SMTP | 25 | Simple Mail Transfer Protocol is a standard electronic-mail protocol that handles the sending of mail from one SMTP to another SMTP server. |
|  | FTP | 20, 21 | The **File Transfer Protocol** is used to connect to remote computers, list shared files, and either upload or download files between local and remote computers. |
|  | POP3 | 110 | Post Office Protocol 3 is used for electronic messaging across the Internet.  POP3 is a protocol that involves both a server and a client.  A POP3 server receives an e-mail message and holds it for the user. |
| **Transport layer** | | | |
|  | Protocol | Description t | |
|  | TCP | Transmission Control Protocol uses a reliable delivery system to deliver layer 4 segments to the destination. This would be analogous to using a certified, priority, or next-day service with the Indian Speed Post;Service. | |
|  | UDP | UDP uses a best-effort delivery system, similar to how first class and lower postal services of the Indian Postal Service work. With a first class letter (post card), you place the destination address and put it in your mailbox, and hope that it arrives at the destination. | |
| **Network layer** | | | |
|  | Protocol | Description | |
|  | ARP | The Address Resolution Protocol (ARP) is an Internet layer protocol that helps TCP/IP network components find other devices in the same broadcast domain. | |
|  | IPX/SPX | Internetwork Packet Exchange/Sequenced Packet Exchange developed by Novell and is used primarily on networks that use the Novell NetWare network operating system.  The IPX and SPX protocols provide services similar to those offered by IP and TCP.  Like IP, IPX is a connectionless network layer protocol. | |
| **Data link layer** | | | |
|  | Protocol | Description | |
|  | FDDI | Fiber Distributed Data Interface, shares many of the same features as token ring, such as a token passing, and the continuous network loop configuration. | |
|  | PPP | The Point-to-Point Protocol (PPP) is used to establish a direct connection between two nodes.  It connects two routers directly without any host or any other networking device in between.  It can provide connection authentication, transmission encryption and compression. | |
| **Physical layer** | | | |
|  | Protocol | Description | |
|  | ISDN | Integrated Services Digital Network adapters can be used to send voice, data, audio, or video over standard telephone cabling. | |
|  | Modems | Modems can be external, connected to the computers serial port by an RS-232 cable or internal in one of the computers expansion slots.  Modems connect to the phone line using standard telephone RJ-11 connectors. | |