IS0 REFERENCE MODEL REVIEW

Terry Fox, WB4JFI
President, AMRAD
1819 Anderson Rd.
Falls Church, VA 22043

Introduction

This pa er will review the status and positions 0 4 the various layers in the IS0 (International Standards Organization) Open Systems Interface Reference Model (OSI-RM) as applied to amateur packet radio.

Some amateurs believe that the OSI-RM provides a good basis for the development of computer networking via amateur radio because of the flexibility it allows. Others see that same flexibility as a lot of unnecessary overhead that takes its toll in reduced throughput and added complexity at actual network implementation. Even the most die-hard supporter of OSI-RM must admit that it is less than optimum, especially at the network layer. I believe however, that it is the best game in town at this point, and what we amateurs have implemented so far falls neatly into the OSI-RM architecture.

Overview

The OSI Reference Model for a modern data communications system is broken into seven distinct levels. The terms level and layer are used almost synonymously whenever the OSI-RM or its levels are discussed. Actually, when describing or referring to the function, level is generally considered the correct term, and when calling a particular level by name, layer is more often used. Thus, the first level of the Reference Model, Level 1 is called the Physical Layer. A small point I admit, but one we should keep in mind.

The seven levels that make up the OSI-RM are:

Level 7. Application Layer (highest level)

Level 6. Presentation Layer

Level 5. Session Layer

Level 4. Transport Layer

Level 3. Network Layer

Level 2. Link Layer

Level 1. Physical Layer (lowest level)

Each one of these levels has certain responsibilities to make sure data travels from a source device to a destination device accurately and promptly.

Each of these levels communicates with its peers along the overall network as necessary, using its associated lower level as the communication medium (except for Level 1, which has no lower level). All information received from an upper level by a lower level should be considered as data and not altered beyond what may be done to enhance communication of the data within that level (this includes any headers required by the upper levels).

It should be noted that there is potential in the OSI-RM for a lot of duplicity of functions, depending on what protocol is implemented at each level, and how complex the resulting network is. This is especially true when the affect of multiple levels of multiplexing data paths is considered, as most levels allow. Simpler network systems may leave out certain levels because they just don't apply, or add unnecessarily to the complexity of the overall system. I would recommend that if any level is bypassed, at least one null character is inserted where that level would otherwise go, leaving the network designer with an "out" in case that level is deemed necessary at a future date.

One of the major advantages of the OSI Reference Model is that it will (hopefully) allow substitution at any one of the individual levels, without seriously affecting the other levels of the overall network. This means that one area can use the same Network Layer, for example, as another area, while implementing a totally different Link Layer protocol. This not only allows for creative implementations at any of the levels, but also allows for each level to suit the need of its medium.

A good example of this might be the creation of different Link Layer protocols depending on the communications medium used (meteor scatter likes smaller frame sizes than VHF/UHF terrestial channels), while using the same Network and higher layers.

As mentioned above, this design does have its weaknesses. Sometimes, the levels need to be broken down further than they are (such as the Network Layer into the Network Sublayer and Internetwork Sublayer), while other times there seems to be a problem effectively separating different levels (the Datagram type Internetwork Sublayer relies on the Datagram Transport Layer heavily for proper operation). This paper will discuss the various levels independantly, and try to account for any interdependance as necessary, starting with the lowest level, and working upward. I will also mention some of the alternatives at various levels, and make some recommendations based on my opinions as of the date of this paper.

Level 1, The Physical Layer

Level 1 is the lowest level in the OSI Reference Model. It is concerned primarily with the "real world" part of sending and receiving data. This is not as small a task as initially thought. There are several parts that make up the whole Physical Layer, including:

Voltage levels.

Data and handshaking signals.

Speed of data transmission and reception.

Order of bit transmission and reception.

Modulator/Demodulator (Modem) types.

RF signalling channels.

All of these different parts have to match each other at both ends before any data can be transferred from one location to another.

Typically, data at the Physical Layer is sent over a radio channel in a serial bit stream. The interface between the users terminal or computer is generally also serial, usually asynchronous ASCII, at speeds between 300 and 9600 baud. In serial operation, RS-232C is the common interface for defining voltage levels, data and handshaking signals, the types of connectors used, and their pinouts.

The data speed and modem type are related to the RF signalling channel used in amateur packet radio communications. It is very difficult to design a modem that will allow data transfer at a rate of 56kbps (kilo-bits-per-second) over a data path using the HF frequencies. It is beyond this paper to specify optimal data rates and modem types for all aspects of amateur packet radio, rather, I will discuss some of the more common systems presently being used or being actively discussed.

VHF/UHF Operation

There is only one commonly used standard on VHF/UHF at the moment. It is the Bell 202 modem, running at 1200 bps. This is an extremely popular standard in that it affords a relatively fast speed of operation (compared to 60 wpm Baudot), yet does not require special radios or other difficult to obtain equipment. There are a lot of surplus 202 type modems available, along with several simple modem designs. There are even single-chip modems becoming available (such as the AMD 7910) that do the whole modem magic in one IC.

Even satellite operation is being experimented with, using the above mentioned 202 standard. Some users are finding that some modem designs (sucn as the phase-locked-loop modems) are not functioning as well as others, primarily due to the inferior signal-to-noise ratio SSB over a satellite gives as opposed to VHF FM.

There is some experimentation going on with higher speeds, particularly on the 220 MHz band, where we are allowed to run up to 56 kbps. The present experimentation generally consists of speeds up to 9600 bps (the speed where most HDLC chips internal clock recovery circuits start to die), using different modulation and demodulation techniques. One of these is to not use the classic concept of a modulator and demodulator, but rather shift the RF carrier some specified amount at the transmitting end, and take the output of the discriminator output directly from the receiver, before any audio processing. This technique is actually quite old (relatively speaking), some of the early packet experiments in Canada used this technique quite well at speeds up to 4800 bps. The drawback to this system is that it requires the modification of the radios to be used, and may not give enough of an increase in speed to warrant a long-term commitment of time and materials necessary to develop the system.

There is the potential for a lot more experimentation in the VHF/UHF area, including extremely high speeds using microwave RF technology such as gunnplexers.

Meteor Scatter

Some experiments using meteor scatter are in the design stage. These tests will probably be conducted on 6 meters, with stations about 600 to 900 miles apart (optimum meteor scatter range). Due to the extremely short duration of meteor scatter paths, high speeds and small packet sizes will be the order of the day. This may cause special protocols to be developed to reduce the amount of overhead required, and take into account the sporadic nature of this RF medium.

HF Operation

There is some HF packet operation going on now, with the promise of a lot more in the near future. HF can allow a major jump of physical space in a single hop, if the correct frequency of operation is chosen. HF does have its own set of peculiarities to deal with, such as narrowness of the channel bandwith, selective fading of different frequencies within the channel, and intersymbol distortion due to the RF energy taking multiple paths to reach the other end.

Some of the initial tests were conducted on 40 meters using the VHF standard 202 type modem, running at 300, 150, and even 75 bps. The reason for this initial choice was that the equipment was already hooked up and operating, but it was found that this system leaved a lot to be desired. The major problem in this system was the wide bandwidth necessary to be clear of interference (202 modems use FSK with one tone being 1200 Hz and the other being 2200 Hz, resulting in a shift of 1000 Hz, requiring almost the whole 1000 Hz to be devoid of other signals, no small feat on 40 meters).

One answer to this modem problem is the Packet Adaptive Modem designed by Paul Rinaldo, W4RI, and Robert Watson. This modem uses a relatively new technique to amateurs, Minimum Shift Keying or MSK, for the transmission of data. It will eventually be able to run up to 1200 bps with a channel bandwidth equivalent to a 600 Hz shift FSK modem. The design is completed, and some of the boards are being tested now. The finished system will be written up in an upcoming issue of QST.

Another set of experiments being conducted uses a 200 Hz s'hift FSK modem running at 300 bps. Bob Bruninga, WB4APR is among the group testing this system on a regular basis on the 10 MHz band, using surplus Bell 1.03 type modems. The 30 meter band has some real. advantages to the packet user, the main one being the lack of QRM. Bob routinely maintains connections for up to several hours at a time now, implying this may be a reliable method of transferring packets over a medium distance.

The Physical Layer is the only level that maintains an actual "physical" or ''electrical" connection with its peers. The rest of the levels communicate with their respective peers through assigned "logical" or 'virtual' connections. Since these logical connections aren't part of the real, physical world but rather system concepts implemented in computer programs, there must be an actual computer device used to implement these protocols. These computer programs run either as a portion of a mainframe program, or, more frequently, in a smaller, dedicated computer.

Level 2, The Link Layer

All this leads us to the Link Layer. This level is responsible for receiving and sending data from the higher level protocols and sending that data to or receiving the data from the the Physical Layer, respectively. Part of this responsibility is to make sure that data integrety is maintained through the physical devices implemented, and recovering from any errors occuring in the physical world.,

Figure 1 shows several types of devices interconnected as a portion of an amateur packet radio network. Note that there is a separate link layer that corresponds to each Physical Layer.

In order to insure data integrity over the Physical Layer, the Link Layer does several things to the data it receives from the higher levels. Most Link Layer protocols start by taking the data received from the higher level and creating small groups of data, called frames, then sending these frames to the Physical Layer for actual transmission. Most link protocols add a certain amount of overhead at the beginning and end of the actual data to be sent. This overhead usually consists of an assigned number of the frame, frame type identifiers, frame source and/or destination identifiers, and some sort of mathematically derived number that is used as a check to make sure both sides of the physical interface have the same data. These basic functions are described in an ISO standard (ISO 3309), commonly referred to as the High-level Data Link Control protocol, or HDLC.

There are two versions of Link Layer protocols commonly used in amateur packet radio today. Both follow the HDLC standard for the addition of flags, address, control, and Frame Check Sequence (FCS) fields. The flags are used to indicate the beginning and end of the frame, the address field is used to indicate who the frame is from and/or going to, the control field is used to show what type of frame is being conveyed, and the FCS field is a cyclic-redundancy check calculated on the data between the opening and closing flags.

In order to assure t'he flag character (01111110) does not appear anywhere in a frame except at the beginning or end, anytime five or more one bits are found in the data, a zero bit is added after the fifth one bit. The receiving end will realize that the zero was added, and delete it.

The first thing most Link Layer protocols do is to establish a "virtual" connection between the two devices wishing to communicate. This allows both devices to know what mode each is in at any given time. In order to make and maintain this connection, certain types of frames are sent back and forth that don't carry any user data, but rather perform command or supervisory functions related to the status of the link. There are also supervisory link functions to make sure one device doesn't "overload" the other with data faster than the receiver can handle it.

Vancouver Protocol

The first Link Layer developed for use on the ham bands was based on the IBM variation of HDLC, called SDLC. This protocol was developed by Doug Lockart, VE7APU, the "father" of packet radio on the ham bands. It is connection oriented, and uses eight-bit address and control fields, along with the standard CRC for the FCS. There are a few supervisory frames necessary for creating and maintaining the connection, along with flow control frames to prevent overloading. The level 2 Vancouver protocol works fine, and its overhead is minimal.

AX.25 Level 2

After the AMRAD group used the Vancouver protocol for a while, it became obvious that there were some limitations to this protocol. The main limitation had to do with the addressing information imbedded in each frame. The Vancouver protocol uses eight bits for the addressing information. Some implementers of the Vancouver protocol modified it so that the addition of "digital repeaters" or digipeaters could be used. These additions took up two of the eight bits in the address field, leaving six bits for actual addressing. This meant that only 64 users could be addressed before overflow was reached. In addition, someone in each group had to assign these numbers to stations, and make sure that numbers weren't assigned twice.

AX.25 took care of this by installing the amateur's callsign in the address field. One more addition saw both the source and destination addresses in the address field. This meant that the address field of a frame jumped from one byte to 14 bytes in a single bound! A further addition saw first one, and now up to eight digital repeater addresses in the address field. Talk about overhead! Unfortunately, in order to design a system that hams can use easily, a system like this is almost a necessity.

In addition, AX.25 added more supervisory frames, and is designed to be more flexible in higher speed and full duplex systems. Most amateurs using packet radio today are using the AX.25 Level 2 standard, and all packet systems available today can support the AX.25 Level 2 protocol.

AX.25 also allows multiple link connections, so that several stations can be interconnected. This includes connecting to one's self, allowing testing of packet software if there are no other stations around (as long as there is a repeater available).

Those wishing to read more about these protocols should refer to the following:

Vancouver protocol available from:

Vancouver Amateur Digital Communications
Group (VADCG)
C/O Doug Lockhart, VE7APU
9531 Odlin Road
Richmond, B.C. V6X 1E1

AX.25 Level 2 protocol specification:

Second ARRL Amateur Radio Computer Networking Conference Proceedings available from the ARRL for $9.00.

Updates on the AX.25 Level 2 protocol are available in the AMRAD Newsletter.

Digital Repeaters

Both the modified Vancouver protocol and the AX.25 Level 2 protocol support devices called "digital repeaters" or "digipeaters". These type of repeaters differ from the normal voice type repeater in that they generally operate as time-domain, or store-and-forward repeaters rather than the frequency-domain system used by voice systems. What this means is that a repeater will listen to a frequency for frames it should repeat. When it hears one, it pulls it into its memory, checking to be sure there are no errors, and then waits for the sender to drop its transmitter. The repeater then re-transmits the frame on the same frequency. This allows several packet stations to communicate over a single frequency that might not otherwise be able to hear each other. Since a single frequency is used, spectrum usage is cut in half. In addition, the repeater is usually a very simple device, since no cavities or filters are required.

Level 3. The Network Laver

The next level up the ISO-RM is the Network Layer. The units transferred at the Network Layer are called "packets". This level probably should have been split into two distinct levels. The lower level, sometimes called the Network Layer or Level 3A, maintains control over a single, smaller network of users. The upper portion, called the Internet Layer or Level 3B, interconnects these smaller groups into a larger network, allowing individuals or systems in one group to communicate with others in different groups if they want.

At this point, I think it would be advantageous to discuss for a moment the two basic types of network designs, the connection oriented, and the connectionless (clever name) or Datagram type. These two systems differ greatly in their design philosophy but either can be used in place of the other without adverse affects.

Some think that a whole network and internetwork must be the same type, or communications cannot happen, but with the proper separation of functions, gateways can be built allowinn different systems at almost any level. A gateway is a device that transforms one type of protocol that exists on one side of it to a different type protocol being used at the other side of it. When properly designed, gateways are capable of interfacing two completely different style protocols to each other, as if the difference didn't exist.

Getting back to the two types of networks, I will first discuss the connection oriented network, followed by the connectionless type.

Connection Oriented Network

The connection oriented network operates very similarly to the Link Layer protocol. In order to transfer any user data across the network, a "connection" must first be made from one user to the other. This involves passing between the two stations (and any network controller that may exist) a connection request and acknowledgement. Once this connection is made, any data travelling between the two users must travel through the path established at the time the connection was created. If any unrecoverable errors occur, the defective connection must be torn down, and a new connection must be made, if possible.

Some of the advantages of a connection oriented protocol are:

1. Lower overhead per packet once a connection is made, since all information about who is communicating and what path is being used is sent only once (when the connection is being generated). This lower overhead usually simplifies the software necessary to implement the protocol.

2. Out-of-sequence packets *generally aren't allowed, again simplifying the software needed to implement the network protocol, and also simplifying the higher level protocols.

3. Connection oriented protocols are generally easier to implement than datagram type protocols.

4. Once a connection is made, the routing of packets doesn't have to be recalculated over and over (and over and over) and over again.

Some of the disadvantages of the connection oriented network protocols are:

1. Since the route of data flow is established at connect time, if there is any failure along the path chosen, the connection must be torn down and re-established using a different path. This implies that any network using a connection oriented protocol should be as reliable as possible. Unreliable networks may take a longer amount of time to keep the network running than actually pass data.

2. If part of the network becomes overly congested, since there is no way to dynamically alter the path used in a connection, the congestion will become worse as time progresses, unless there is a way to automatically tear down and re-establish connections around the congested portion.

3. Out-of-sequence packets aren't normally allowed, causing Accurately received packets to be rejected because of badly received earlier packets. This could- cause an increase in channel occupation, reducing effective channel throughput.

4. If a station is moving through areas covered by connection oriented networks, it could have a problem when the time comes to leave one area and go into another. How a roving station can be passed from one network to another in connection oriented networks isn't a big problem presently, but it could become a problem as the use of packet radio increases.

There are more advantages and disadvantages for the connection oriented protocols, but those mentioned above are the most important.

Connectionless Protocols

The connectionless type of protocols (called the datagram type from here on) operate in a different manner than the connection type. In a datagram protocol, all information needed to get a packet from the source to its destination is included in the header of each packet. Obviously, this will cause the header to become larger than the equivalent packet of a connection oriented network. In addition, each packet's routing must be decided independantlv from others either preceeding or succeeding it, causing a lot of additional operating overhead while each packet switch decides the b&t way for this packet-to go. This can come in handy when a network is not too reliable, or when a portion of a network becomes congested, since the path taken by packets can be dynamically altered. This doesn't come cheaply however, it usually takes more computer power to make sure a datagram type network functions properly.

As the last paragraph illustrates, the advantages and disadvantages between connection oriented networks and datagram type networks are generally just the opposite of each other.

Level 3A. The Network Sublaver

The Network Sublayer is responsible for taking data from the higher level protocols, packetizing it, and sending it to the Link Layer for actual transmission through the Physical Layer. While the Link Layer is responsible for making sure the user data accurately transverses the physical link between two stations, the Network Layer is responsible for making sure that user data passes through any intervening nodes, such as metropolitan network controllers or packet switches. Along with this, the Network Layer makes sure that any data from another network either passes through the network successfully, or reaches the destination station if that station is part of the metropolitan network.

When I first began to study protocols above level 2, I was impressed by the datagram type of network. It seemed to have a lot going for it. especially in a relatively unstructured and potentially unreliable amateur radio packet network. Datagram networks are very forgiving by nature, allowing for the voluntary nature of amateur stations showing up, and then disappearing.

Then we found out how people were implementing datagrams, and on what type of machines. It seemed that most people were implementing datagrams on large computers or mini-computers. There didn't seem to be a practical implementation of a datagram network based on microcomputers.

In addition, the two major commercial data networks seemed to be implementing connection oriented networks very effectively, including the use of microcomputers in their implemenations. This is when I started taking a second look at the CCITT standard X.25, both at level 2 and level 3.

About the same time, Gordon Beattie Jr, N2DSY, was coming to the conclusion that X.25 would be a good place to start on establishing a standard protocol for levels 2 and 3. In the summer of 1982, Gordon came down to the Washington area, and we had a conference with Eric Scace, K3NA, at Telenet.

Eric became a valuable asset in our discussions, since in addition to working at Telenet, he served on the CCITT committee on X.25. It turns out that there can be a large difference between what a protocol document appears to say, and how the protocol is actually implemented. This is where Eric really helped, by giving us insight not only into w'hat the protocol designers meant, but also how the real world networks implemented the protocol.

As a result of these meetings, we came up with drafts of protocols for both levels 2 and 3. Level 2 eventually grew into the AX.25 Level 2 that most packeteers are now using. Level 3 is a much larger, more sophisticated protocol, and as such, requires more careful analysis to see what we need and what we don't in the amateur community. As with level 2, we can't just throw out portions of the protocol without making sure they wont be needed in the future.

A separate paper in these proceedings discusses the level 3 protocol in some detail, so I wont get describe it in detail here. It is based on the CCITT X.25 Level 3 protocol, using "virtual circuits". Permanent virtual circuits weren't deemed to be useful, at least at this point, in the amateur service, and the Datagram service of X.25 was eliminated by the CCITT because of lack of interest.

One of the main arguments used against connection oriented networks is that they aren't very forgiving in unreliable enviroments. It seems that most metropolitan networks should be reliable enough to support connections without major problems. Since connections require less channel overhead than datagrams, this should also allow more efficient use of RF frequencies.

The recommendation to go AX.25 at the Network Sublayer is not cast in stone, but it appears that this is the best compromise protocol to use at the local or metropolitan level.

Internet Sublayer

The Internet Sublayer is the next step (or half-step in this case> up the ladder to the user. This level isn't necessary for purely local or metropolitan communications, since the data at that level isn't intended to go outside the individual network. The Internet Sublayer is only necessary when data must flow outside a single network s boundry.

Since the Internet Sublayer is responsible for the transfer of data across individual networks to the destination network, there must be enough addressing information in the level 3B header to make sure the packet can be successfully routed to its destination. The internet protocol is also responsible for making sure any fragmentation of large packets into smaller packets is done in an orderly fashion.

The amateur community is very inventive, and often likes to use whatever is invented locally rather than using a “standard” foisted on us by some outside group. Keeping this in mind, and also keeping in mind the potential of some networks being not as reliable as others, I propose that we use a datagram type of internet protocol. Even though datagrams might require more computer power to implement, not every user will be required to have this overhead, since the internet protocol is used to interconnect the individual networks, not each user.

Among the datagram type internet protocols available are the DARPA internet protocol and the National Bureau of Standards (NBS) internet protocol. These two are very similar, in fact the NBS standard grew out of the DARPA one. It seems that either of these might suit our needs with some slight “adjustments” for amateur peculiarities. The main difference between these two is that the NBS version has longer address fields (which we may need). The DARPA internet adds a minimum of 20 bytes of header (more for options), while the NBS version adds a minimum of 28 bytes. Otherwise, both look almost identical. Figure 2 shows the outline of an NBS internet header. Unfortunately, it is beyond the scope of this paper to discuss the operation of these protocols.

One important thing to keep in mind when discussing internet protocols, particularly the datagram type, is that the internet protocol must work very closely with the next level protocol, the Level 4, or Transport Layer protocol. The datagram type internet assumes that a rather large transport protocol resides above it, making sure that any alterations of data that might crop up due to internet operations (such as packets arriving out of sequence) are properly corrected. This interdependence is why the internet and transport levels are often referred to as one combination protocol (such as TCP/IP which means Transmission Control Protocol/Internet Protocol). It is important to keep this in mind when designing or implementing an internet protocol.

As mentioned before, just because a datagram type of protocol is chosen for the Internet Sublayer, this DOES NOT mean that a datagram Network Sublayer must also be implemented. This is just NOT true. In fact, included in the NBS documents on the NBS internet protocol is software describing an interface to an X.25 Network Sublayer. The two are separate items, and should be delt with as such.

Level 4, The Transport Layer

The main function of the Transport Layer is to make sure the data passed on from the higher levels at one side of a group of networks interconnected using an internet protocol is received at the intended destination correctly.

Part of this responsibility is to make sure the data is received in the same order as it was sent. In datagram protocols, it is possible for one packet sent before another to arrive at the destination network after the second one. This could cause big problems if left uncorrected. The Transport Layer must make sure all packets are in the correct order before sending them on up the ISO-RM ladder to the higher levels. This may involve buffering the packets for a period of time, potentially requiring large amounts of memory.

Another responsibility of the Transport Layer is to notify the originating network that the packet successfully reached the destination network. In addition, the Transport Layer may impose flow control procedures on packets as necessary.

As mentioned earlier, the Transport Layer works very closely with the Internet Sublayer. This means that if the DARPA internet is used, the DARPA transport protocol should also be used. The DARPA transport protocol adds an additional 20 bytes minimum of overhead as a transport header. If the NBS internet is chosen, the NBS transport protocol should also be implemented. The NBS version is more complicated than the DARPA version, and some of it might have to be thrown out if it is to be used on a microcomputer system.

Level 5, The Session Layer

Now that the data has transversed the network successfully, it is ready to be used by the intended destination device. If that destination device is a larger computer, capable of running several programs simultaneous1y, there must be a way of telling which program the received data is intended for. This is part of the responsibility of the Session Layer.

One example of this might be Dave, K8MMO's system having someone running an orbit prediction program the same time as another person is editing a document, both running under MP/M II. The other example might be having several -different people using the same program, such as a bulletin board program, at the same time.

The Session Layer adds its own overhead to make sure the proper application (be it a program or another user) gets the correct user data. The Session Layer introduces a new term for the block of data it deals with, the "messaig”. Within the overhead that the Session Layer adds is some sort of routing information to insure that the data received from the network is sent to the proper program within the computer, which is referred to as a “port”. These ports are generally assigned names by the application being run.

The Session Layer also makes sure that an 0perating session between a user at one end and the program at the other end is handled smoothly. If the user should suddenly disappear from the system, it is up to the Session Layer to inform the application of this problem, so that the application can take any action deemed necessary. This implies that the Session Layer not only handles data between the network (via the Transport Layer) and any applications involved, it also passes some status and control information between the network and the application in question.

The Session Layer is not a necessity in a lot of instances, such as two people typing back and forth ala RTTY mode. In this case, the Session Layer overhead could be considered unnecessary and eliminated.

The Session Layer is a subject that needs further study at this time, as there are several versions ou t (DARPA, NBS, CCITT. BX.25 etc). Since there aren’t a lot of mainframes on the a packet networks so far (there isn’t even a network as such), there is time to study this level carefully before making a commitment to any particular standard.

Level 6. The Presentation Laver

The Presentation Layer is responsible for making sure that the data passed from one end of a hook-up to the other end makes some sense, and is displayed in an orderly fashion. It specifies things such as what character code is used and screen and printer display control sequences (such as cursor addressing).

The Presentation Layer can be a very complicated system, or it can be a null level, depending on what type of devices are being used at each end.

If, for example, a glass TTY (such as is used for the hearing impaired) is to be used with a version of a word processor set for a Heath H-19 terminal, the Presentatin Layer would be very complicated. Not only would there be code conversions required (ASCII vs Baudot), but also screen formatting characters would have to be converted, along with other problems. The end that would do the conversion would depend on what type of Presentation Layer protocol had been agreed to by the users of the system.

If a different user was to use the same word processor with a Heath H-19, and the Presentation Layer protocol agreed to was the H-19 running ASCII, the Presentation Layer at both ends could end up being a null level, since the same protocol is implied at both ends.

Level 7, The Application Layer

Application Layer protocols are primarily concerned with how a particular program is operated by the user of the program. The application protocols are established so that users (be they individual or another program) will know how to correctly use a program through the network.

The Application Layer, being the top of the system, would normally be the last area to look at for standardization. Since there are a myriad of programs that could be run as application programs over the amateur packet radio network (and a lot more not even thought of, or written yet) this could end up being the hardest set of protocols to come up with.

Two types of programs that should have standard protocols written for fairly quickly though. They are the message system (generic name) and the file transfer programs.

There are many message system programs available to the amateur today. It seems that every one of these systems uses different commands to operate it, along with a different message format. It would help greatly if there could be a single, standard set of commands available, along with a standard message format. Then, each message system along a network could potentially access other message systems along the network, and automatically grab off any pertinent data. Also, there could then be defined within this message system protocol, a way of automatically forwarding messages along the network from a source message system to the destination message system.

There are many different message systems, and many different message system "standards" already in existance. DARPA has a standard, so does NBS, and the CCITT just ot name a few. This is an area I haven't delved into too far yet, so I have no feeling at this time as to which protocol would best suit our needs. Some initial work is being done by Paul Rinaldo, W4RI, Hank Magnuski, KA6M, Larry Kayser, WA3ZIA, along with the AMSAT and VITA contingent on message system standardization.

The other Presentation Layer protocol that needs almost immediate attention is the file transfer protocol. A lot of us are presently using the Ward Christensen protocol so prevalent among CP/M users for exchanging CP/M files using modems over the phone lines. In fact, this protocol has been implemented in many computer systems other than CP/M, including 6800 type computers and (rumor has it) DEC minis.

One of the faults with the CP/M file transfer protocol is that it uses a very simple checksum on the data transferred to make sure no errors crept into the transfer. There has been some modifications made in this area recently, some versions of this transfer program now allow either the original checksum routine or a more sophisticated CRC type calculation. Since there is so much redundant checking of data at the lower levels of a network, the more sophisticated version may not be needed.

There is also a protocol for file transfer floating around that was developed by the NBS but I haven't had a chance to study it carefully enough yet to see if it would fit our needs.

Conclusion

The OSI-RM appears to be taking shape in the amateur packet radio network. There is some protocol development work being done at almost all levels of the Reference Model, with most people working from the ground up at this point.

One of the disadvantages of the OSI-RM is thet there is a lot of added overhead, as mentioned at the beginning of this paper. This is primarily because multiplexing of different data paths is allowed at each level, causing multiple flow control procedures and addresses to be required at each level.

An alternative to the OSI-RM system might be to break the overall network design at different places. Eliminating the redundant capability of multiplexing of operations at each level would reduce the total amount of overhead required. This would have to be done very carefully.

It is hoped that this paper will help the newcomer to amateur packet radio understand how a data network is designed and implemented using the OSI Reference Model to allow a maximum of flexibility to the designers and implementers. I further hope that this paper stirs interest in the more advanced packet radio enthusiasts by stating my opinions and suggestions on recommendations at the various levels of the OSI-RM.

Comments or suggestions regarding any portion of this paper should be addressed to the author at the above address, or be sent to the Amateur Radio Research and Development (AMRAD) Newsletter for publication at the following address:

Amateur Radio Research and Development
PO Drawer 6148
McLean VA 22106-6148

```
                                   Octets
                         1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-----------------------------------------------------------------
! Version !  IHL  !  Grade of Service  !        Data Unit 'Length        !
|-----------------------------------------------------------------|
|         Identifier          !Flags !        Fragment offset        |
|-----------------------------------------------------------------|
|     Lifetime     |  User Protocol   |        Header Checksum        |
|-----------------------------------------------------------------|
|                          Source Address                          |
|-----------------------------------------------------------------|
|                    Source Address (continued)                    !
|-----------------------------------------------------------------!
|                        Destination Address                       !
|-----------------------------------------------------------------!
|                 Destination Address (continued)                  !
|-----------------------------------------------------------------!
|              Options as needed               |      Padding      |
|-----------------------------------------------------------------|
|                              Data                                |
|              Note: This field is of variable length              |
|                           End of Data                            |
-----------------------------------------------------------------
```

Figure 2. NBS Internet Header Format

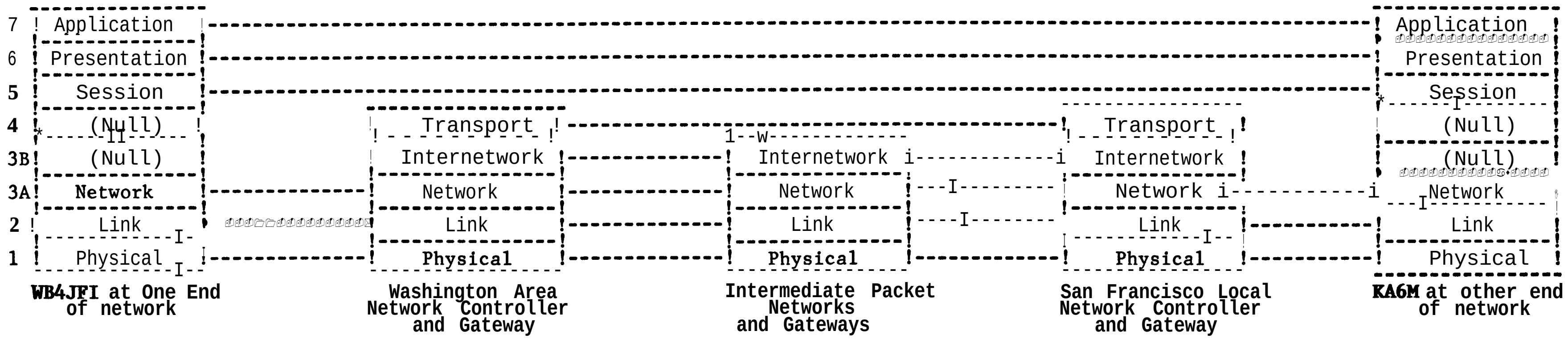

Figure 1. Example of IS0 Interconnections in an Amateur Network
Shown is a Theoretical Packet Connection Between WB4JFI in the Washington DC
Area, and KA6M in the San Francisco Area