PROTOCOL LAYERING

David L. Grebe
Apogee Labs, Inc.

ABSTRACT

The advent of COTS based network-centric data systems brings a whole new vocabulary into the realm of instrumentation. The Communications and computer industries have developed networks to a high level and they continue to evolve. One of the basic techniques that has proven itself useful with this technology is the use of a "layered architecture." This paper is an attempt to discuss the basic ideas behind this concept and to give some understanding of the vocabulary that has grown up with it.

KEY WORDS

Layered Protocol, Protocol Stack, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol (IP)

GENERAL BACKGROUND

Protocol layering is a common technique to simplify designs by dividing them into functional layers, each concerned primarily with one task. While the terminology derives from networking designs it can be broadly applied to general system designs dealing with data acquisition and delivery. Protocol layering should not be confused with so-called 'top-down' design techniques. Successive layers do not necessarily address more elemental or primitive concepts. Rather, each layer is designed to perform a well-defined function while interacting directly with only the layer immediately beneath it to provide facilities for use by the layer above it. This means that each layer is **independent of the others** and is therefore **replaceable**. Within each layer one or more entities may implement its functionality. For any given application only the layers required need be assembled, which holds the promise of reusability without imposing costs inherent with monolithic solutions that need to address all anticipated requirements.

Taken together these protocol layers make up the "communications protocol stack." There are several general things about layered protocol stacks that are often misunderstood by people not in the business of setting up communications systems between computers. One of the misconceptions is that there is a single "standard" protocol stack. We hear a lot about the ISO seven-layer model and it seems to be some kind of a standard, and it is, but the fact of the matter is that no one builds systems that use that model.

Well if nobody uses it then why does everybody talk about it? It is not true that nobody "uses" it just that nobody builds systems that fully comply with it. It has become the model against which everything else is measured. You will find a communications protocol described in relationship to the layer or layers that it would represent if it really was an implementation of the ISO model. This is a useful function since it gives people a frame of reference and improves communications between people regarding what is in a given protocol layer. If someone says that this is a "layer three" protocol a person who understands this technology knows what to expect that protocol to do. It is very much like describing a device to an instrumentation engineer as a pressure transducer. There are still many things about it that are still unknown but he now knows what questions to ask to determine the exact nature of the device.

Another thing that should to be noted is that this protocol stack is only concerned with communications and not any application. The application is generally thought of as existing "above" the communications protocol stack. This is true even if the protocol stack has an "application" layer. This is the layer that interfaces the communications protocol stack to the various applications. There will often be many applications running at the same time but only one communications protocol stack. The other thing that we will note at this point is that there may be several protocols running at the same layer of the protocol stack. Transmission Control Protocol (TCP) and User Datagram protocol (UDP) are a pair of prime examples of this. Both are layer four protocols, both may be present in the same protocol stack at the same time and one application may be using TCP while another is using UDP. We commonly talk about TCP/IP as if the two protocols were a matched set but it would be just as correct to talk about UDP/IP since Internet Protocol (IP) is a layer three protocol that both TCP and UDP often use. To try to diagram all of the protocols that may exist in a system at a given time is a bit like trying to describe a maze. This maze will have many ways through it and an application can use the ones that provide the exact functionality that is required at the time.

For example a data acquisition system might want to use TCP to transport the setup information. This would guarantee that the set up file is received correctly but at the cost of the network bandwidth required to acknowledge each part of the setup file and to retransmit it if there was any question about it being correctly received. That same system might use UDP to transmit the data because it could not afford to pay the bandwidth penalty required by TCP and could afford occasional lost samples of data.

The most common example of a protocol stack in use today is the Internet protocol stack. It is generally said to have four layers but if you try to identify all of the components that exist in the Internet protocol you will find dozens of protocols not just four. Many of them co-exist at each layer of the protocol stack.

The following discussion is based on a four layer model that is the basis for the current Internet. It was developed in the 1970's as a DARPA project that succeeded beyond its wildest dreams. The four layers (Figure 1) operate together as follows:

The **PROCESS LAYER** implements user-level functions such as mail delivery, file transfers and remote login. Typical implementations are TELNET, FTP, and SMTP programs, provided by different vendors with different nuances, but interchangeable as far as the rest of the protocol stack is concerned.

The **HOST-TO-HOST LAYER** handles connection establishment, flow control, retransmission of lost data. Typical program implementation here is TCP and UDP.

The **INTER-NET LAYER** formats data for delivery across a number of different physical networks. The most widely used 'data unit' at this layer is the IP datagram, handled by the Internet Protocol (IP). This protocol also provides an address that is independent of any address used by a lower layer. This allows a datagram to be sent over many different media as it goes from its source to its destination without regard to the addressing used over the physical media.

The **NETWORK ACCESS LAYER** is responsible for moving the data over hardware interconnect media. This could be Ethernet, 'T' carriers and Fiber Optic links.

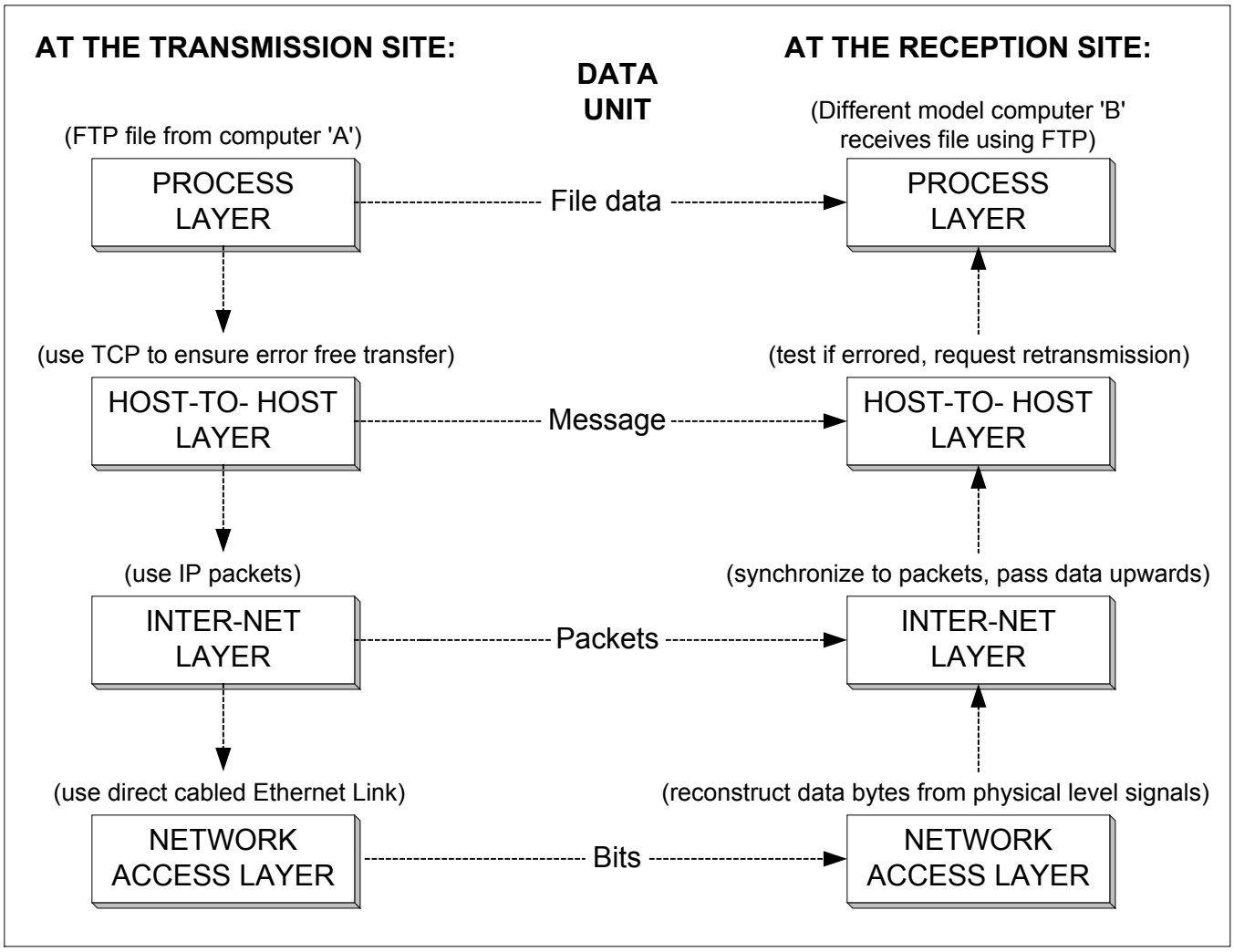

Figure 1: Internet Protocol Layering Model

Note that as far as each layer is concerned it is connected with its corresponding level at the other site without regard to HOW that connection came to be made by its underlying layer(s). If after some time of

use it becomes necessary to implement a higher speed link, only the Network Access Layer would need to be changed for example. Note also that the standard interface allows connection between all types of computers that subscribe to the standards chosen.

Another useful construct that falls out is the ability to interject a small subset of the model between the two sites shown above in order to take advantage of a different physical layer that may not be directly supported by the equipment at each site. For example, suppose that after initial laboratory testing of these two co-located computer based systems the transmission site needs to be moved several hundred miles away. The Ethernet link could be broken by a new function at each site that would convert from the Ethernet back to the IP protocol, and then re-encode the packets into a long haul T1 link. A corresponding function at the receive site converts the received 'T1 bits' back to 'Ethernet bits'.

Note also that as the capabilities of layers evolve over time a controlled, 'graceful' upgrade migration path is available without the need necessarily to upgrade all layers simultaneously.

ONE MODEL OR MANY?

There are several other protocol models.

The Seven Layer OSI Model defines layers as:

7. APPLICATION: Network services such as e-mail and file transfer
6. PRESENTATION: Formatting, encryption, and compression of data
5. SESSION: Setup and management of end-to-end connection
4. TRANSPORT: End-to-end delivery of messages
3. NETWORK: End-to-end delivery of datagrams over intervening networks
2. DATA LINK: Transmission of packets on a given link
1. PHYSICAL: Transmission of bits

Networks of IBM computers utilized a System Network Architecture (SNA) protocol model:

TS, Transaction Services
PS, Presentation Services
DFC, Data Flow Control
TC, Transmission Control
PC, Path Control
DLC, Data Link Control
PH, Physical Control

Open Data Network (ODN) was proposed as a ***framework*** within which Telephone, Computer, and CATV networking can be studied. This was not proposed to be a standard like the OSI model, but rather a tool to collect and compare the technologies and services applicable to the markets to be addressed by each application. This approach may be a good parallel to our efforts within the range/DoD community. The point being there is no one best protocol model, but rather protocol layering has been a powerful tool to allow for connection between all types of equipment and networks from many vendors and

service providers. It succeeds because it provides a ***set*** of standard procedures, data units, and exchange techniques that are not tied to specific instantiations. Beyond supporting interoperability at peer-to-peer levels, it provides the flexibility to adapt new techniques without complete systems re-design. Equally important is the ability to 'layer the layers' and provide for protocol conversions using subsets of protocol stacks.

ISSUES IN RANGE SYSTEMS

Perhaps we should start here with a bit of history. In the dim and distant past when the test ranges were set up, or even before, the problem of the testing engineers was to measure physical parameters on board a vehicle, transmit that data to a display system of some sort and to be able to reproduce these parameters for evaluation. In these early days this often consisted of manometer boards and movie cameras to record the readings. As the technology evolved, the photo recorder was replaced with systems that could record the data on magnetic tape and even transmit it to the ground so that it could be viewed in real time. These techniques improved with time and the analog recording and transmitting techniques were replaced with digital techniques but the problem remained the same. Measure physical parameters on the vehicle and pass that information to the display facility for processing and display. IRIG 106 PCM is a perfect example of the state of this art. However, with the advent of the computer onboard the vehicle, the problem changed. The computers sometimes transmit parameters between each other and sometimes they transmit messages. Believe it or not, a text string is not a parameter no matter how much we try to pretend it is and to fit it into our comfortable model. Today most data acquired on a vehicle comes from the computers not transducers that we put onboard. Video and audio systems are also important data types on the test ranges. They differ from parameterized data mainly in the data rates and the way that the data is processed.

This means that we now have multiple fundamentally different types of data in our systems. We have the classic parameterized data and we have messages. These two types of data require different handling and thus different standards. First lets consider the parameterized data. In this case there will be many different sources of data but they all need to come together at the display facility and be processed in much the same fashion. We do not want to build display facilities that need a different process to display the data from each sensor. This implies that we need a standard way to put the parameterized data into the system so that we can process it using the same hardware and software in the display facility. For sensors that we put on the vehicle this should be rather straightforward. That is not to say that it will be easy to get agreement on all details but it should be possible. Parameterized data that comes from messages transmitted between computers pose a slightly different problem and there we need to decide whether to provide the processes in our display facilities to interpret this data or to try to move it onto the vehicle. However, the parameterized computer-to-computer information is normally in one of a few formats so it is probably practical to have different display processes for each different format on the display system rather than change formats on the vehicle.

To handle this parameterized data there are some significant differences between the processing that is needed in our display systems and what is normally seen in computer-to-computer communications. Most computer-to-computer communications are concerned with passing the entire array of data from one place to another. With parameterized data it is normal to pass only a selected subset to the display (the recorder will probably want it all) and to be able to do this without the source knowing what subset

is going to be required. In other words the source will broadcast the data and the receivers will each take just the parts that they want. Often the part that is desired does not include all of the data within a packet so it is not just a problem of receiving some packets and discarding others it needs to go deeper than that to accepting or rejecting individual parameters. COTS products will normally want to pass an entire packet. We may need a specialized protocol that is not commonly found in COTS products. This can be done in a central specialized facility or on each display station and economics will probably dictate the choice.

As we have already noted there are messages passed between computers on the vehicle that do not contain parameterized data. There are text strings, error messages of various types and an endless string of other possibilities. The problem is how to deal with these messages. In most cases specialized logic exists in both the computer on the vehicle generating the message and in the computer intended to receive the message that is used to interpret these messages. There are two possible ways to approach this problem. We can try to interpret these messages on the vehicle and put them into some displayable format or we can relegate that problem to specialized software on the ground. This may very well mean that the test community tells its customers that it will acquire the relevant data, transmit in some form to the display facility but will simply pass it through to the customer for any further processing. The customer may bring a display processor into the display facility and process the data there, the customers software could be loaded on one of the display system computers or any of a number of other solutions but the display problem for this type of data should be in the customers hands. This means that we need a standard way to "package" the message so that we can pass it to our customer. Another way of describing this process is that we need to define a standard set of protocols to handle message traffic that can then be implemented in the data acquisition system and in the display and customers system to facilitate this traffic. These protocols would not know or care what was in the message but they could deliver it to the system that does.

AN EXAMPLE

The following table is an example of how a layered protocol stack for a network-centric data acquisition system might look.

<table>
<tr><td>Top Layer</td><td colspan="2">Encryption protocol</td><td colspan="2">Compression Protocol</td></tr>
<tr><td>Process Layer</td><td>Parameterized data protocol</td><td>Message data protocol</td><td>Audio Protocol</td><td>Video protocol</td></tr>
<tr><td>Host-to-Host layer</td><td colspan="2">TCP</td><td colspan="2">UDP</td></tr>
<tr><td>Internetwork Layer</td><td colspan="2">IP</td><td colspan="2">Address Resolution Protocol (ARP)</td></tr>
<tr><td>Network Access Layer</td><td colspan="4">Fibre Channel</td></tr>
</table>

Note that the column that a particular protocol is listed in does not indicate which higher layer protocols may use that protocol. The encryption protocol can use the audio protocol that can in turn use TCP. The Address Resolution Protocol (ARP) may not be used by any higher-level protocol since it is used to determine how to route a message from the source to the destination.

FIRST, THE OPPORTUNITES THAT THIS PROVIDES

One of the first things to note is that this is an opportunity to examine the way that we package data for transmission, recording and display. With the IRIG 106 type systems there was in general a single way to put data into a system. We could get creative but within limits. If we are to abandon these systems that means that it is time to examine the advantages and disadvantages that we had with the old PCM systems and to make sure that we retain as many of the advantages as possible while eliminating many of the disadvantages that we faced. We need to be very careful when moving to a new standard that we do not "throw the baby out with the bath water." But we really do need to make a careful examination of the basic processes and capabilities to define a system that will stand us in as good a stead as PCM has for so many years.

We need to recognize where fundamental differences exist in the data that we are trying to collect, record, transmit and process and to provide a set of standards that will allow each type of data to be handled in a cost effective manner. Where multiple data types exist we will need multiple ways to deal with that data within a common framework. One aspect of that common framework must be the Internet. It is too much a part of our systems to ignore without driving our costs up instead of down. We will need to think of things like telemetry transmitters and data recorder as ways to bridge or route the data across a different media. We will need to look at interfaces to avionics busses as gateways or bridges that take messages from the avionics bus and allow them to be routed across the data acquisition network. If we can accomplish the development of new standards where they are required and the adoption of existing standards where they will do the job we will be able to simplify the instrumentation, transmission, recording and display of data and that will result in more efficiently run test programs. There is a caveat here as well. If we try to "force fit" our task into existing standards just because they exist, we may be making the standards development process easier at the expense of day-to-day operations and that is false economy.

WHAT RESOURCES EXIST TODAY?

There are many technologies that can be used that have been developed by the communications and computer industries. For example the use of fiber optics for communications allows for almost unlimited bandwidth for data transmission both on the vehicle and on the ground. The weak point here is in the real-time transmission link and in the capabilities of the onboard data recorders. Neither of these elements in the system can match the transmission media. The recorders can provide adequate bandwidth; it is the capacity that provides the limits there. Fibre is not the only physical layer media that is available to use when we implement our systems. As long as we implement the systems using commercial standards and layered protocol techniques, the possibilities are almost endless. One of the beauties of this approach is that to use a different media does not mean redesigning the entire system just the parts that will profit by the change.

Processing power is available and it is becoming available at any level in the system from the display system all the way down to the sensors. This has the potential to provide the capability to acquire "information" rather than just "data." However, this needs to be done in a few standard ways; we cannot afford to address it piecemeal. The biggest problem with doing this is not our ability to accomplish it, but to accomplish it reliably. To do that requires a thorough understanding of each problem and a

correctly implemented solution. It is like aliasing in our present systems. The data can "look" good but have artifacts in it that we cannot detect.

WHAT IS UNIQUE IN RANGE APPLICATIONS?

Much of the answer to this question has already been discussed in earlier parts of this paper and the author is not capable of coming up with an exhaustive list. However we can introduce some of the areas of differences between commercial applications and the range applications.

Most data passed between computer systems today is not "time critical" in the sense that range data is time critical. If someone sends an MS word file the only time critical factor is the patience of the sender and receiver. With range data lives can be at stake as well as millions of dollars in a test vehicle. The ability to make a decision based on the data can be extremely critical. This makes the timely delivery of the data with a minimum of latency much more critical than it normally is in commercial applications.

Since the latency in a packetized data system is a variable with only the maximum latency specified, this leads to another problem. That is that there is no longer a fixed relationship between when a sample is acquired and when it appears in the data stream. Even worse is that the data that appears on the network is not in time order. Effect may come before the cause! This will force the data to be time tagged with an accuracy and resolution that commercial protocols do not need. The necessity to be able to time tag the data further adds to the overhead that is associated with putting the data into packets. So a new protocol that is not in common use with computer systems will need to be used.

Another difference is that the ranges are used to transmitting data as a continuous stream. When the data is packetized, its nature moves from being a continuous stream to a burst availability mode. This and the additional overhead required by the packets will further tax the bandwidth that is available at the test ranges. The lack of enough bandwidth to telemeter the data will require innovative solutions. Data compression and two-way communications with the vehicle to allow real-time control of what is transmitted are possibilities that will need to be evaluated. However, these are issues that can be handled in a layered protocol stack and implemented only when required.

Some real thought about where in the system to tackle the various issues is needed. As smart sensors become available they will be able to implement some of the needed layered protocols. This will be especially true for high bandwidth signal sources such as audio and video. In other cases it will fall into the realm of intelligent data collectors or concentrators. For avionics bus data the bridge that takes the avionics data and places it on the data acquisition network will need to put the information in the protocol that is required by the data acquisition network.

CONCLUSION

Like the Open Data Network *framework,* it may be very useful to adapt a suite of techniques that can be applied to each of the practices encountered in Range Applications. A requirement must then be established that implementations operate within the adapted layer definitions and techniques. Thus, data acquisition subsystems only acquire, data distribution systems distribute and recorders only record. To support this, a new generation of bridges and routers can provide the required interconnectivity.

ACKNOWLEDGMENTS

The author wishes to acknowledge the valuable assistance of Mr. Lee Eccles in preparing this paper. Mr. Eccles' extensive background in telemetry and data acquisition systems contributed greatly to the discussion concerning the range applications.