Transmission Control Protocol/Internet Procotol (TCP/IP) to TRM Case Study

US ARMY Tank-Automotive Research,

Development and Engineering Center (TARDEC)

Objective of Case Study

This case study will show how the Transmission Control Protocol (TCP) Internet Protocol (IP) communication suite of protocols uses a layering implementation that can modeled via the DoD Technical Reference Model which uses defined direct and logical interfaces. This case study provides a very general description of how TCP/IP relates to the TRM, and it doesn't provide an in-depth explanation of TCP/IP.

Rationale/Purpose

The Technical Reference Model (TRM) is a evolution between the TAFIM and the GOA model. The TAFIM view defines services within layers, and the General Open Architecture (GOA) standard, SAE AS4893, to define the interfaces between layers. These layers share information with each other through a defined set of protocols. The TRM stack (stacked layers) begins with the physical hardware layer and proceeds through a set of layers to the application layer via a precise set of direct and logical interface definitions that clearly identify the separation of boundaries between the layers. A layered structure provides the following attributes: layer portability, interoperability between associated layers, plug and play implementations, and affordability via contractor competitive competition for the development of each of the independent layers.

The purpose of this case study is to show that the TCP/IP suite of communication protocols can be modeled using the TRM.

Description/Process Used

In the TRM to TCP/IP diagram below, the model is setup as a layered structure. The TRM layers define boundaries that should be used when developing computer hardware and software interfaces. The TRM abstract model helps a software developer recognize interfaces and modularize the software into layers. These layers then pass information amongst other components and systems.

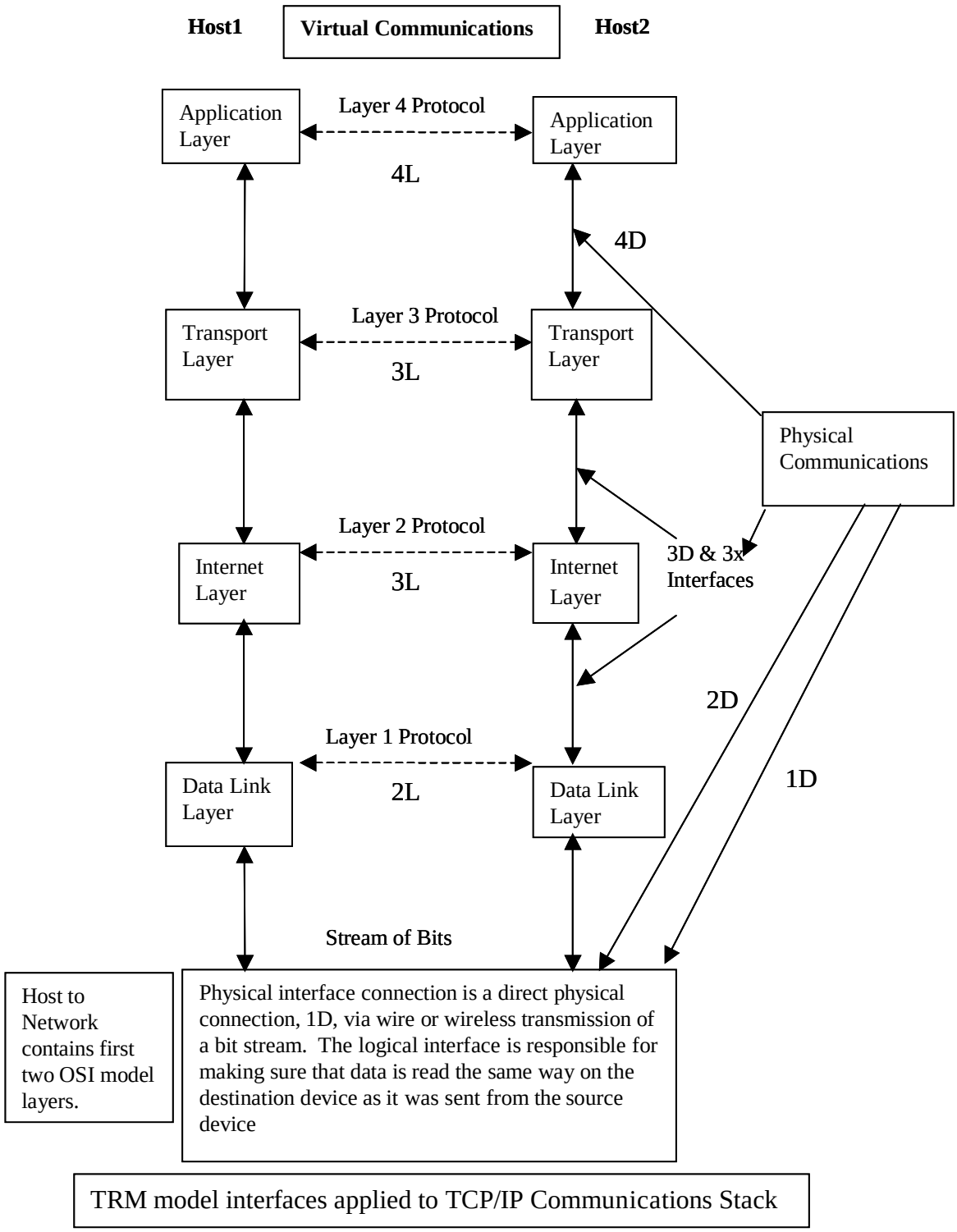
To develop a communication suite of protocols that are portable and easily maintain, the TRM is the modeling tool to help identify the hardware/software layers needed to pass information. A concern in passing information is how to identify and define the data parameters needed to transfer the information from one platform to another platform successfully keeping in mind the separation of boundaries to the input layer, output for that layer, and the logical interface understanding between adjacent components or systems. A software developer must also look at how the passed information will be protected and verified correct upon receipt. Header data, bit insertion techniques, and various algorithms are used to ensure application data within a bit stream is not corrupted. Also within a communications protocol, the receiving layers should send acknowledgements back to the sender of the information letting that layer know that the passed

information was successful. If not successfully passes, the sending layer will retransmit the information.

An example of such a communications protocol is the Transmission Control Protocol (TCP) Internet Protocol (IP). TCP/IP uses four stacks or layers to transfer data, because the physical layer and data link layer are combined. In the diagram, I have broken these two layers out to show the distinction between the layers.

The purpose of a layer is to offer certain services to the higher layers, shielding those layers from the details of how the services are actually implemented. Layer n on one-machine carries on a conversation with layer n on another machine via the protocol definition (logical interface definition). A resulting communications protocol stack in formed when data is passed from each individual layer from the lowest layers to the highest layers and the highest layers to the lowest layers. The picture below shows a protocol stack formation, and the commutations between host1 to host2.

The virtual communications between host1 and host2 is the logical interface classes, which establishes the communications understanding or protocol understanding between layers. The physical communications is the direct software layer algorithm to direct software layer algorithm interface, which passes and receives the defined software data parameters based on the logical interface definition. The software layer algorithms are created based on the protocol definition, and each software layer algorithm implementations are independent from other software layer algorithms. Virtual communication (logical interface) would not exist with out the direct interfaces, and the logical interface definition determines the data parameters that the direct interface algorithms utilize.



TRM model interfaces applied to TCP/IP Communications Stack

Below is a description of how the TCP/IP packets (stream of data) of information that are separated into columns of information which form the layers in the TCP/IP stack.

In the TCP/IP communication protocol, the architecture is based on the Internet datagram definition. The make up of the datagram is a stream of bits, 1 and 0's. This stream of bits is further defined by bytes. There are 8 data bits to one byte. The Internet datagram is defined as:

Physical/Data Link	IP Layer Data	TCP or UDP Layer	Application Data	Frame Check Sequence

Note: The Frame Check Sequence is used as error checking in data transmission and UDP is used for error checking when send isolated messages to another system.

A datagram is a finite-length packet with sufficient information to be independently routed from source to destination without reliance on previous transmissions. Each of the separately defined byte lengths represent data that the software layer algorithm understands via the protocol definition.

Physical Layer

The TRM physical layer is responsible for generating and actually passing the physical electrons, stream of information, from sending hardware component to the receiving hardware component. The physical layer contains the hardware, sensors, microcontroller, microprocessors, data buses, electromagnetic waves, and electrical interface requirements, which is the 1D direct interface.

- Logical Interface

  An example of the logical interfaces associated with the physical layer bit stream of information being passed from the sending hardware to the receiving hardware is shown below.

  011111 11011111 010 Data Bits

  0111110110111110010 Stream of Information being Transferred

  Extra Bits Inserted and Removed by the Hardware

  The sending hardware inserts a zero bit (0) after every five consecutive one-bit pattern (1). This procedure is called zero bit insertion or bit stuffing. The receiving hardware will remove the inserted zero bits upon receipt of the data stream. The two hardware components must have already defined the logical interface definition before bit insertion. So, the sending and receiving hardware already have an understanding on how they will communicate with each other before sending and receiving information data. This logical interface definition is normally used so duplication for the user data (application data) is not corrupted from sending hardware to receiving hardware.

- Direct Interface

  The direct interface is a bus connecting the two hardware pieces.

Data Link Layer

The task of the data link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Various framing methods are used, including character count, character stuffing, and bit stuffing. Data link protocols can provide error control to retransmit damaged or lost frames.

Internet Protocol Layer

The Internet Protocol (IP) layer routes data between the hosts, other systems. The information data may be passed to a single network or may be relayed across several networks in the internet. IP address data routes its traffic without caring which application to application interaction a particular datagram belongs to.

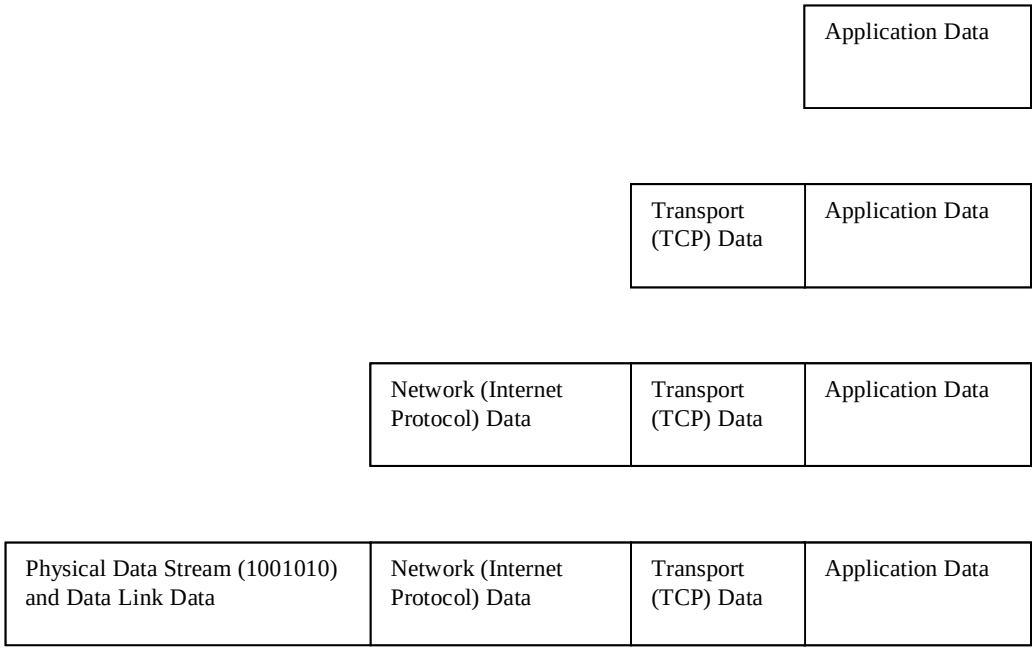
Transmission Control Protocol (TCP)

The TCP provides a reliable data connection service to applications. TCP has as an algorithm, which guarantee that data is error free, complete, and in proper sequence for recombining packets of data. TCP ensures data is sent and received by sending acknowledgement message between the sending TCP layer system and receiving TCP layer system. It is a logical interface or virtual communications interface.

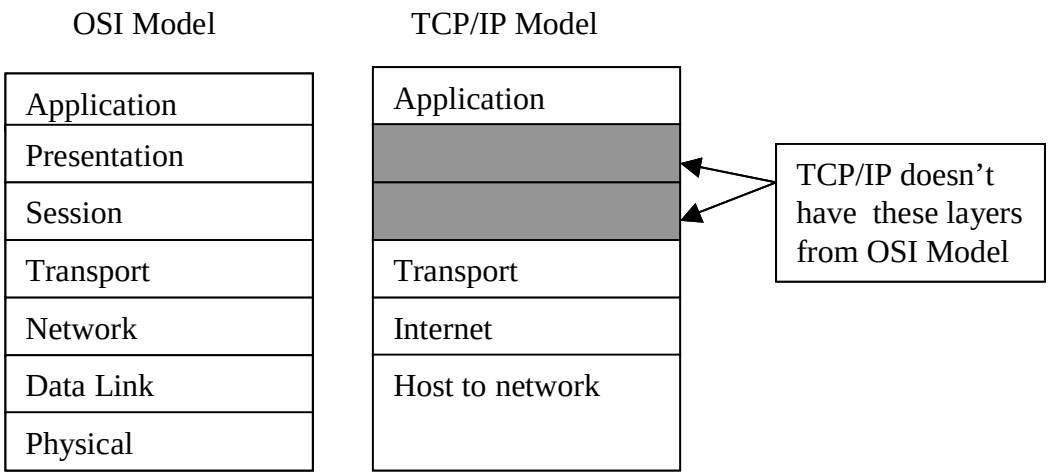
Application Layer

Application used by TCP/IP are: File Transfer Protocol, Simple Mail Transfer Protocol, Telnet terminal access, Domain Name System (DSN) directory services, and program to program communications.

The two diagrams below show how the information data can be separated into segments and represented as a layers/stacks.



Relationship between OSI and TCP/IP Models

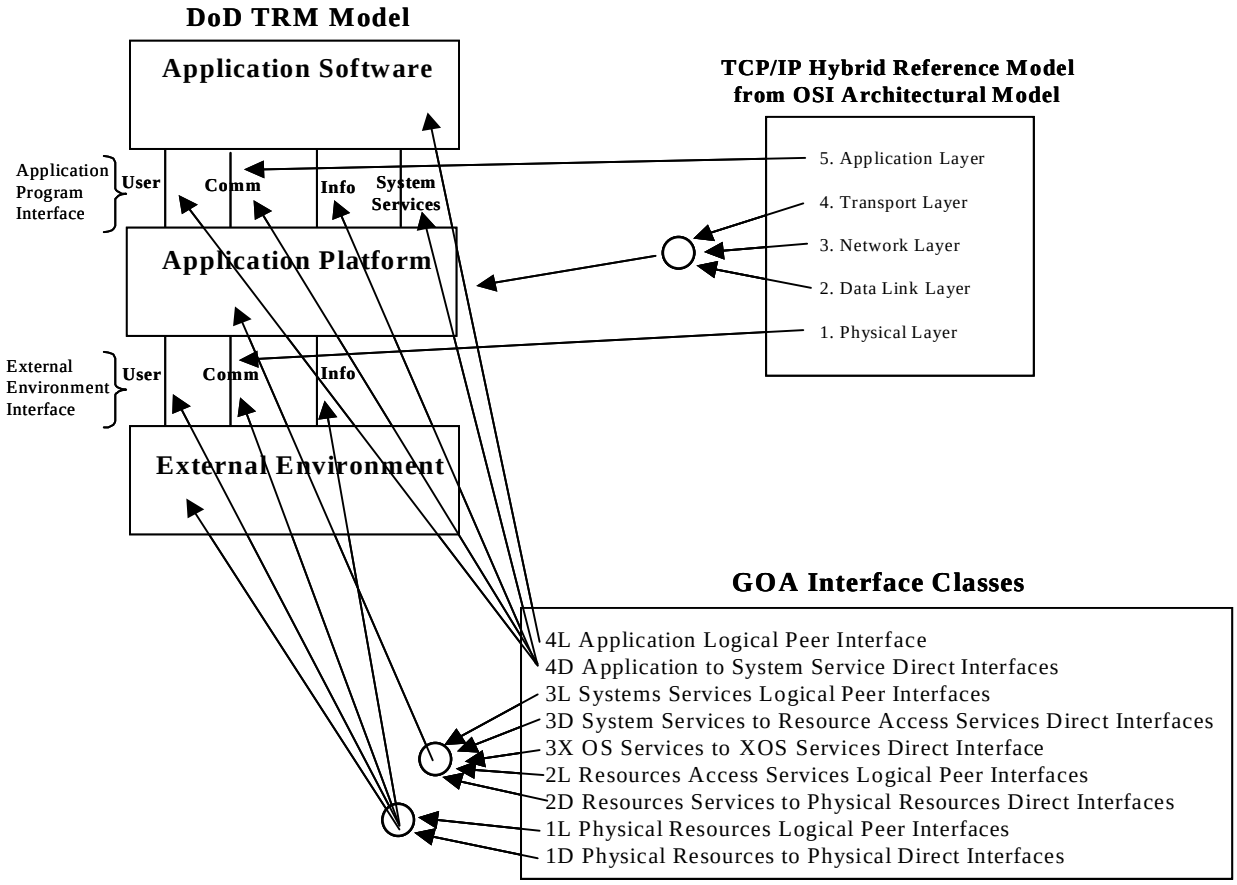


Results/Findings

In the comparison between the TRM, OSI, GOA and TCP/IP, I have broken down the TCP/IP model using the TRM and OSI model concepts. The TCP/IP model doesn't make a distinction between the physical and data link layers. Each layer in the TCP/IP model uses the fields in the datagram above to check for error transmissions and router destinations while protecting the application data. Each of the defined fields, bytes, are stripped away via a particular layer

algorithm as defined by the protocol being used. When the remaining datagram bit stream reaches the application layer, the bit stream contains some header information for error checking and the application data. The application software can use the application data without having any knowledge of how the data arrived.

The picture below is another view of the relationship between the TRM model, TCP/IP and the GOA interface classes. The TCP/IP model can easily be represented via TRM model and OSI model.



Definitions

Destination Address: This field contains the destination information (used by the network Layer).

Source Address: This field contains source information (used by the network Layer).

User Data field: This field contains data used by the application (used by application).

Frame Check Sequence: Error checking for data transmission

Service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented.

Protocol is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer. Layers use protocols in order to implement their service definitions.

Host to Network (first two layers):

- The physical layer is concerned with the transmission characters of wire, fiber optics, and wireless communications. It passes the steam of bits in the form of a datagram.
- The data link layer delimits the start and end of frames to be used by the network layer.

The network layer is concerned with getting packets from the source to the destination.

The transport layer provides reliable, cost-effective data transport from the source machine to the destination machine, independent of the physical network.

The application layer is concern with security, naming within the internet, network management, and the applications such as electronic mail, net news, multimedia, and world wide web.

Expectations

The TCP/IP communications protocol is an example of how to define, using the TRM model, interfaces within an system. Defining the interfaces between layers makes the software portable. A common understanding (datagram definition) should be established between system layers for the purpose of interoperability. The TRM model provides the layer concept needed to support interoperability, portability, and affordability. The TRM model is a good tool to use for computer and software developers.

Main domains use the TCP/IP in their communications with each other. (DOD, Industry, Various Countries). Interoperability between these domains is achieved because they use the same standard (TCP/IP standard). JTA accomplishes the same interoperability goals mandating standards that are to be used by the weapons community.

Reference

1. TCP/IP (Architecture, Protocols and Implementation) by Sidnie Feit
2. Computer Networks, Third Edition, by Andrew S. Tanenbaum
3. DOD TRM Version 1.0, Nonember 5, 1999
4. SAE AS4893