CPS221 Lecture: Layered Network Architecture

last revised 9/10/12

Objectives

1. To discuss the OSI layered architecture model
2. To discuss the specific implementation of this model in TCP/IP

Materials:

1. Projectable of OSI Layers
2. Projectable and examples from Forouzan ch. 2 (p 19)
3. Projectable showing OSI encapsulation
4. Projectable of correspondence between TCP/IP and OSI Layers

I. **Introduction**

A. As you recall, last class we introduced the notion of a protocol What do we mean by a protocol?

ASK

B. Networks in general, and the Internet in particular, make use of what are sometimes called protocol stacks or layered protocols.

PROJECT: OSI Layers

1. The motivating factor for the use of a layered architecture is the incredible diversity of systems and physical devices - e.g.

a) Diverse platforms - computers running various OS's, but also devices such as smart phones, entertainment systems, etc.

b) Diverse forms of physical connection - all sorts of wired connections, wireless connections, fiber optics, etc.

2. The basic idea is that each layer is responsible for a particular kind of functionality. Each layer depends on the layers below it for other functions, and provides services to the layers above it.

a) Example: one book uses the following example

Discuss examples and projectables from Forouzan ch. 2

b) Example: A while back we made a reservation for a campground that provided four different mechanisms for making a reservation:

(1) One could download the reservation form, fill it out, and attach it to an email

(2) One could download a reservation form, fill it out, and fax it

(3) One could download a reservation form, fill it out, and mail it

(4) One could telephone the facility, where a person would fill out the form based on information supplied on the phone

In all cases, the result was a form which would then go to the person responsible for actually entering the reservation.

In effect, the reservation form defined a protocol that made use of a lower layer implemented by email, fax, postal mail, or a person on a telephone

3. One major goal of a layered architecture is to factor out various services so that a given service can be used by multiple versions of the layer above it and can make use of multiple versions of the layer below it.

Example: the same reservation form "protocol" can be used whether the information is communicated via email, fax, postal mail or telephone. Conversely, fax, say, can be used for many other purposes besides transmitting reservation forms!

II. **The ISO/OSI Reference Model**

   A. In the early days of networking, each networking platform used its own protocol stack. In 1984, the ISO standardized a model called the "Open Systems Interconnection" reference model.

      1. Note that the standards body is the ISO; the reference model can either be called OSI or ISO/OSI

      2. The OSI reference model is as a <u>model</u> that can be used to describe protocol stacks.

      3. At the time the OSI model was first produced, there were a large number of different protocol stacks in use by different vendors, including the TCP/IP protocol used by the Arpanet. Today, TCP/IP has become the dominant protocol, and is where we will focus our attention. However, we will briefly look at the ISO/OSI model.

   B. The ISO/OSI model has 7 layers. Rather than discussing them in detail, let's consider a (hypothetical) example on a system which can be described using the ISO/OSI model (not a TCP/IP system of sort that is typically used!)

   (We'll consider the layers top-down, so the order of our discussion will be the reverse of the way the layers are conventionally numbered)

   Suppose I wanted to make an ecommerce purchase using my credit card.

      1. (Layer 7). The Application layer deals with matters like accepting my order and receiving needed information like my name, address, credit card number, etc.

2. (Layer 6). The Presentation Layer deals with matters like encryption. In the ISO/OSI model, this is factored into a separate layer because

   a) Not all uses of an application like a web browser need encryption

   b) Facilities like encryption are useful for many different kinds of application

3. (Layer 5). The Session Layer deals with matters like authentication. Many ecommerce sites require you to establish a session by logging in, and expect you to terminate the session by logging out. In the ISO/OSI model, this is factored into a separate layer because

   a) Not all uses of an application like a web browser need the notion of a session.

   b) Facilities like authentication are useful for many different kinds of application

4. (Layer 4). The Transport layer deals with error-free transmission of information between my computer and the server - including dealing with the possibility that some information may be lost in transit and thus must need to be retransmitted.

   a) The transport layer deals with <u>logical</u> units of information (e.g. web pages or forms) which - if long - may need to be broken into smaller units for handling by the lower layers. In this case, the transport layer on the sending end needs to segment the information, and that on the receiving end needs to reassemble it, dealing with issues like lost, duplicated, or out-of-order pieces - all of which are handled at this layer so that higher layers don't need to worry about them.

b) The transport layer deals with communication between a specific <u>process</u> on the sender and one on the receiver.

5. (Layer 3). The Network layer is actually responsible for the transmission of information over the network, including, in particular, issues of choosing a route from the source node to the destination.

   The Network layer deals with communication between specific <u>hosts</u> (systems), which may be the sender or the receiver or some intermediate host in the network

6. (Layer 2). The Data Link layer deals with reliable transmission of information between pairs of systems (e.g. it deals with errors). What else this entails depends, of course, on what sort of physical connection exists between a given pair of systems.

7. (Layer 1). The Physical layer corresponds to the medium over which the information is physically transmitted (e.g. various forms of wired or wireless connection.

C. A few things to note:

1. A layer does not correspond to a single protocol, but rather to a family of protocols.

   a) For example, http:. smtp:. ftp:. ssh: ... are all Application Layer protocols.

   b) For example, the Data Link and Physical Layers can be implemented using many different types of medium, each of which has its own Data Link protocol. (Indeed, using an appropriate protocol, one could implement the physical layer using trained monkeys with flags - but the performance wouldn't be very good!)

2. The upper layers are usually implemented 100% in software; the lowest layer is 100% hardware, and the layers just above it may be implemented in a mixture of hardware and software.

3. Unless the sending and receiving systems are physically connected to one another, information is typically transmitted between systems via a series of hops involving various intermediate systems.

   a) Compare traveling Gordon to the airport - a friend might drive you to the train station, from where you take a commuter rail train to North Station, from where you take the Orange Line to State Street Station, from where you take the Blue Line to Airport Station, from where you take a bus to your terminal.

   b) The lowest three layers exist on each system participating in the transmission; but the highest four layers exist only on the sending and receiving systems.

   c) Systems participating in the transmission (other than the sender and receiver) will have multiple instances of the Physical and Data Link layers - one for each system connected to. (The sender and receiver could have multiple instances of these layers as well, but usually don't; only one is part of transmitting any given message in any case.)

4. Each layer (except the Physical layer) depends only on the services of the layer immediately below it.

   a) On the sending end, the Application layer passes a message to the Presentation layer, which does whatever is needed (e.g. encryption) and then passes it down to the Session layer which passes it down to the Transport layer which passes it down (possibly broken up into smaller pieces) to the Network layer

which passes it down to the Data Link Layer which uses the Physical Layer to send it to another system.

b) On the intermediate systems, one Data Link layer instance receives information from its Physical Layer and deals with any errors. The Network Layer then sends the same information out over a different Data Link Layer instance to the next system enroute to the destination.

c) On the receiving end, the Data Link Layer receives information from the Physical Layer. After dealing with any errors, it passes this information up to the Network Layer which passes it up to the Transport Layer (which may need to reassemble pieces into a complete message), which passes it up to the Session Layer, which passes it up to the Presentation layer which does whatever is needed (e.g decryption) and then passes it up to the Application layer.

5. On the sending end, each layer takes the information from the layer above it and "wraps" it with additional layer-specific information, which is stripped off at the other end. This is called <u>encapsulation</u>.

   PROJECT Encapsulation in OSI Layers

   a) One thing not shown in this diagram is the fact that a level 4 data unit may be broken up into several level 3 data units to be reassembled at the other end..

   b) The result is that a lower level data unit contains headers from the layers above it as well as the original data - e.g, in this diagram, D2 might actually look like this:

   T2 D7 H7 H6 H5 H4 H3 H2

6. The software at the higher layers, though physically depending only on the layer below, logically behaves as if communicating

with its peer on the other system - e.g.

A web browser physically communicates with a web server through the various layers on both systems, but uses the http protocol as if it were communicating directly with the server.

D. On the Internet, messages normally do not pass directly from one end point host to the other; instead, they pass through a series of intermediate nodes.

PROJECT: OSI Layers again - note flow of messages through intermediate notes

1. In the original ARPANET model, these intermediate nodes were called IMPs (Interface Message Processors).

2. Today, this functionality is normally provided by devices known as routers.

3. As indicated in the example projected, these intermediate nodes typically run only the lowest layers of the model (physical, data link, and network) since their task is only to forward messages between the end-points.

III. **TCP/IP**

A. TCP/IP was initially developed as part of the ARPANET in the 1970's.

1. The name is constructed from the names of two protocols:

a) TCP: Transmission Control Protocol - which is one of several possible protocols in the TCP/IP suite that roughly corresponds to the Transport Layer of the ISO/OSI model.   (It turns out to be possible to use the TCP/IP protocol suite without actually using TCP!)

b) IP: Internet Protocol - which roughly corresponds to the Network layer of the ISO/OSI model.

2. Support for TCP/IP networking was incorporated as part of Berkeley Unix in the early 1980's, a very influential system.

3. The fact that that the ARPANET is the precursor of the Internet, and the influential role of BSD Unix have resulted in TCP/IP becoming for all intents and purposes <u>the</u> networking protocol stack in today's systems.

B. Looking at TCP/IP in terms of the ISO/OSI reference model

PROJECT Correspondence between TCP/IP and OSI Layers

1. The lowest layers (Physical and Data Link) are provided by a variety of media-specific protocols.   New protocols corresponding to new types of communication media can be added at any time.

   In TCP/IP, the basic unit of information transmitted at this level is properly called a <u>frame</u> (though sometimes the term packet is used).

2. IP (The Internet Protocol) constitutes the Network Layer.

   a) The basic unit of information transmitted at this level is properly called a <u>datagram</u> (though sometimes the term packet is used)<u>.</u>   Depending on what is used at the two lower layers, a single datagram may need to be fragmented into multiple frames.

   b) This layer makes no guarantees about reliable transmission. In particular

(1) A datagram may be lost in transmission. (In fact, if it is fragmented and one of its fragments fails to arrive within a fixed period of time, the whole datagram is simply dropped.)

(2) A datagram may be corrupted in transmission.

(3) A datagram may be delivered more than once.

(4) Multiple datagrams sent from one system to another system may be delivered in an order different from the order in which they were went.

3. There are actually several protocols at the Transport Layer level. Higher-level code specifies which protocol to use when communication is initiated.

   a) TCP (Transmission control protocol) is the most commonly used protocol.

   (1) The basic unit of information transmitted using TCP is called a <u>segment</u>.

   (a) Several requests to transmit information from the application layer may be grouped together into a single segment.

   (b) Conversely, a request from the application layer to transmit information larger than some maximum segment size results in it being broken up into several segments.

   (c) Each TCP segment is encapsulated into an IP datagram.

   (2) TCP is stream-oriented.

   (a) That is, it does not keep track of boundaries between requests issued by the application layer of the protocol,

(b) In effect, the receiver sees a stream of data coming from the sender, with no indication of how the sender originally divided this information.

Example: the application layer at the sender may have performed a single "send" operation involving 10,000 bytes - but the receiver may receive this in two operations involving 5000 bytes each.

Example: the application layer at the sender may perform 2 "send" operations each involving 10 bytes, but the receiver may receive this in a single operation involving all 20 bytes.

(3) TCP is connection oriented - that is, when two systems communicate using TCP a connection is established by one system contacting the other; then information is exchanged; then the connection is closed.

(4) TCP achieves reliable transmission by an acknowledgement mechanism, and by associating a sequence number with each segment sent over a connection.

(a) If the sender does not get acknowledgement for a transmission that it has sent within a specified period of time, it will retransmit it.

(b) This number may used to request retransmission of a segment that has become corrupted in transmission - in which case the receiver sends a negative acknowledgement (request for retransmit) instead of an ordinary acknowledgement.

(c) This number may be used to detect duplicate datagrams for a given segment.

(d) This number is used to ensure that the receiver sees segments in the same order as they were transmitted.

b) UDP (User datagram protocol) is a simpler protocol with more limited capabilities.

(1) In effect, what it provides is access to the raw IP level.

(2) The basic unit of information transmission using UDP is called a <u>user datagram</u>, A user datagram is encapsulated in an IP datagram an the next level.

(3) UDP is message oriented. Each request to transmit data from the application layer becomes a single datagram, and is received at the other end as a single unit. (This is sometimes an advantage over TCP)

(4) UDP is not connection-oriented. A sender simply transmits information, without needing to first establish a connection.

(5) UDP does not guarantee correct, in order delivery of datagrams.

c) Sometimes TCP and UDP are compared by using the following analogy.

(1) TCP resembles communication by telephone, in the sense that a connection must first be established, that information is received in the order transmitted, and that there are no formal boundaries between segments. (But TCP also guarantees error-correction)

(2) UDP resembles postal mail in the sense that no connection need to be established before sending a letter, letters are separate messages, and there is no guarantee that a letter will

not be lost, damaged, or that multiple letters will be delivered in the order sent. (But UPD can also duplicate datagrams!)

d) SCTP (Stream Control Transmission Protocol) is just starting to show up (e.g. it is part of the most recent versions of Java 7)

(1) It is a message-oriented protocol (like UDP) that includes the reliability guarantees like TCP.

(2) It has a number of other features to support high-performance transmission of large amounts of data.

(3) It was developed to support applications for which neither TCP or UDP is really suitable - e.g. telephony.

(4) However, we will not discuss it further.

4. There is no TCP/IP equivalent to the Presentation and Session Layers in the ISO/OSI model.

a) Their functionality may be subsumed in the application layer

Example: http does not include the notion of a session. Web applications that require this notion (e.g. ecommerce sites requiring a login) typically obtain this function by the use of cookies or any one of a number of add-ons like Cold Fusion (used at Gordon)

b) Sometimes, their functionality is provided by "add-on" protocols such as the secure socket layer (SSL) which provides encryption. (When used with the web, the resulting combination of http: plus SSL is treated as a distinct protocol - https).

5. TCP/IP assumes the existence of an Application layer that builds on top of it. (That is, Application layer protocols are not considered part of TCP/IP, though many protocols such as http are commonly used as application layer protocols with TCP/IP).

C. Ports

1. Recall that the network layer handles <u>host-to-host</u> transmission of data, whereas the transport layer handles <u>process-to-process</u> transmission of data.

    a) In general, a single network connection may be servicing several different applications running on a given computer.

    b) Every IP datagram includes a port number - an integer in the range 0 ... 65535

    c) The transport layer uses this to distinguish between processes - when a datagram arrives over the network, the port number is used to decide which process it is delivered to.

2. We will discuss this further when we talk about the transport layer.