NOTES

Chapter 16a
Protocol Layering

Network Architectures

Problem:

Network Systems are complex. How do you DESIGN and ORGANIZE such a complex system?

Network Architectures: (continued)

Solution:

1. Partition the complex communication problem into smaller subproblems.
2. Layers: Group related subproblems together into a conceptual service layer (i.e., the services provide by the code that implements a layer solve the subproblems of that layer). Order the layers so that each layer builds on the services of the layers below it (i.e., develop a “depends-on” relationship between layers). Similar to layered OS designs.
3. Protocols: Create protocols where each protocol handles one (possibly more) of the many subproblems we described earlier. Protocols that handle the same problem are grouped together into a Layer.

- What functionality should reside in each layer?
- Two popular approaches: (1) ISO OSI (2) TCP/IP

NOTES

Layering Concept

Network software and hardware on each machine is typically divided into pieces and layered. Each layer takes responsibility for handling one part of the networking problem. Conceptually, sending a message means passing the message down through the successive layers of the protocol stack on the sender's machine, transfering the message across the network, and then passing the message back up through the layers on the reciever's machine.

Layering Picture

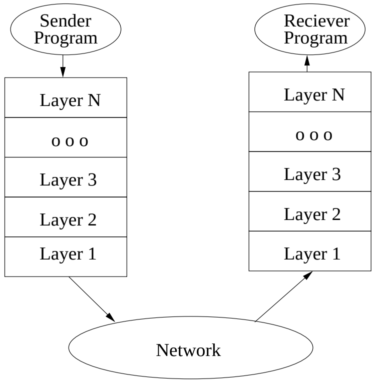

Figure 1: Layering Concept

NOTES

Layers Defined In General Terms

- Each layer defines, in very general terms, what types of services that layer should offer. Essentially it defines the minimum goals (services) provided by a layer. Note that the layering model does not specify

- In other words, layering is a conceptual idea that only loosely defines the organization of the networking software.

Network Layering Principal

1. Layer N at the destination recieves exactly the same object (message) sent by layer N at the source. (treat lower layer protocols like a black box that delivers exactly what it was handed).
2. Layer independence: Layer N does not know anything about layer N-1 or layer N+1 (this allows us to mix and match protocols, in particular to run multiple protocols over the same base protocol, cause the base protocol will not assume anything about what it is carrying).

NOTES

Protocols

- A code module that implements a layer's services is called a Protocol . In other words, a protocol provides a communication service .
- There may be many different protocols at each layer. Each protocol implements the conceptual layer's services in a different way.
- A Protocol Specification defines:

  service interface : the services that is offers its users.

  peer interface : how it communicates with other nodes to implement the service interface.
- A Protocol Stack is defined by selecting one protocol from each layer to create a complete communication channel.

Example Protocol Stack

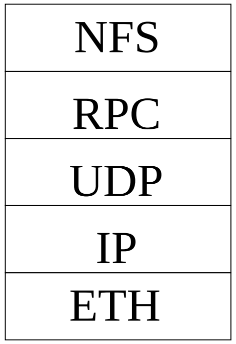

Figure 2: Example Protocol Stack (NFS = Sun's Network File System)

NOTES

ISO/OSI 7-layer Reference Model

ISO: International Standards Organization developed

OSI: Open Systems Interconnection reference model

Reference Model: means a network layering model (term tries to relay the fact that it is a conceptual model)

X.25/400/500: Most popular (example) OSI protocols, developed by

ITU-TS: Telecommunications Section of the International Telecommunication Union, which was previously called

CCITT: Consultative Committee on International Telephony and Telegraph (that is the English translation of the French)

The OSI Network Layers

Physical Layer: responible for getting raw bits from one node to another (electical characteristics, voltages, current, distance, etc.)

Data Link Layer: responsible for creating Frames (groups of bits) and getting them from one node to another. Local Network addresses may be defined by this layer (or the physical layer, or not at all)

Network Layer: provides host-to-host communication and defines the basic unit of transfer (called a network layer packet) (may imply fragmenation/reassembly) network level addressing, and (possibly) routing. These may all be defined just like the link level (e.g. packet = frame).

Transport Layer: provides process-to-process communication and the transport level unit of transfer (often called a message), process addressing, and may add other end-to-end services like reliability. We typically consider the process the endpoint of communication (even if more levels of processing are necessary).

NOTES

OSI Layers: (continued)

Session: No well defined. Used to set up remote communication sessions (rlogin). Can also tie together multiple transport streams into a single "session".

Presentation Layer: Grab bag of stuff. Data format exchange (XDR, Abstract Syntax Notation ASN.1), Compression/Decompression, Encryption/Decryption, Image conversion, ...

Application Layer: Application specific communication protocols (ftp, telnet, http, ...).

- Some services are often performed at multiple layers (e.g., error detection/correction, reliable delivery, ordering, duplicate suppression, ...).
- Every node in the system must implement the physical - network layer.

OSI Figure

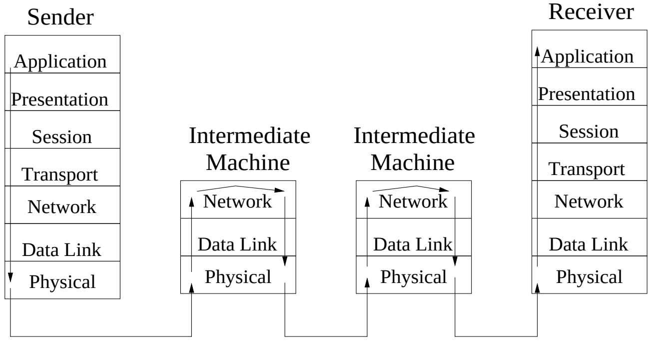

Figure 3: Even intermediate machines must implement the Network Layer

NOTES

TCP/IP layering model

Can be forced into OSI, but really a 4 layer + hardware model.

Hardware: Not an official layer (but like physical layer in OSI).

Network Interface Layer: The layer responsible for transmitting IP packets over whatever network technology the computer is hooked up to. The underlying network could be a complex network provider or a simple directly attached communication channel (like a serial line or tin can and string).

Internet Layer: host-to-host communication, internet addressing and routing, fragmenation/reassembly. Only protocol is IP! Thats why we can be pretty clear about what this layer does.

Transport Layer: process-to-process communication, process-to-process addressing, possibly other end-to-end services.

Application Layer: application specific protocols (may actually be several layers of protocols - its up to the application).

Non-obvious Differences between OSI and TCP models.

1. Many OSI protocols provide reliability at all levels. TCP assumes reliability is an end-to-end problem. Reliability only at transport level (via TCP). Maybe at hardware level, but not required.
2. Who's in charge?
   - OSI protocols (like X.25) often adhere to the philosophy that the network vendor control network service, controlling access, monitoring traffic, deciding routes, insuring reliability, billing, ... (DUMB HOSTS, INTELLIGENT NETWORK)
   - TCP allow hosts to participate in routing decisions and reliability. (INTELLIGENT HOST, DUMB NETWORK)