Protocols and Architecture

Protocol Architecture.

- Layered structure of hardware and software to support exchange of data between systems/distributed applications
- Set of rules for transmission of data between systems
- One or more common protocols for every layer

Protocols

- Characteristics
  - Direct/indirect communication
    - Point-to-point link
      - Two entities share a link, allowing direct communication
      - Entities may also connect through a number of intermediate hosts, bringing in the issue of access control, making the protocol more complex
    - Switched communications network
      - Entities depend on other entities for data exchange
      - Entities may be connected over local network (Ethernet) or may belong to different networks (internet)
  - Monolithic/structured protocol
    - Monolithic protocol
      - All protocol logic (for every layer) is combined into one unit
      - Problem in changing any aspect of the protocol (such as virtual circuit request)
    - Structured protocol
      - Set of protocols with hierarchical/layered structure
      - Clearly shows the distinction between different layers of logic
    - *Communications architecture*
      - Hardware/software used to implement communications with structured protocol
  - Symmetric/asymmetric protocol
    - Symmetric protocol
      - Involves communication between peer entities
    - Asymmetric protocol
      - Involves client/server type of relationship
  - Standard/nonstandard protocol
    - Standard protocol
      - Commonly accepted protocols that have been agreed on by a standards body
    - Nonstandard protocol
      - Built for specific communications situation
- Functions
  - Basis for all protocols
  - Encapsulation
    - Data as well as control information in each PDU

    * Control information is divided into the following categories:
        1. *Address* of the sender and receiver
        2. *Error detection code* or frame check sequence
        3. *Protocol control* for other protocol functions
- Segmentation and reassembly
    * Segment the data stream into small bounded size blocks or PDUs
    * Reasons for segmentation
        · Communications network may accept data blocks only up to a certain size (53 octets for ATM, 1526 octets for Ethernet)
        · Efficient error control with smaller PDU size; fewer bits retransmitted in the event of failure
        · Better access to shared transmission facilities, with shorter delay; nobody can monopolize the network
        · Smaller buffers at receiver stations
        · Can pause transfer for checkpoint and recovery
    * Disadvantages with segmentation
        · Larger overhead with smaller PDU size
        · More interrupts as PDUs announce their arrival
        · More time spent to process smaller PDUs
    * Segmented data is reassembled into messages appropriate for application level
- Connection control
    * *Connectionless data transfer*
        · Each PDU is independent of other PDUs
    * *Connection-oriented data transfer*
        · Used if stations are to be connected for long time or protocol details are to be worked out dynamically
        · Also known as logical association, or virtual circuit, with three phases
            1. Establish connection
            2. Transfer data
            3. Terminate connection
    * Establish connection
        · One station issues a connection request to the other, with or without involving a central authority
        · Receiver may accept or reject the connection
        · Request may include negotiating syntax, semantics, and timing of protocol
        · Protocol may have some options to be negotiated at connection time, such as PDU size
    * Transfer data
        · Exchange data and control information (flow control, error control)
        · Data flows in one direction while acknowledgements flow in the other
    * Terminate connection
        · Either side may terminate connection by sending a request
        · Connection may be terminated by a central authority
    * Sequencing
        · PDUs are sequentially numbered as they are sent
        · Each side keeps track of outgoing numbers (generated locally), and incoming numbers (generated by other host)
- Ordered delivery
    * PDUs may not arrive in order in which they are sent
    * Connection-oriented protocols require the PDU order to be maintained

    * Number the PDUs sequentially as they are generated
    * Problem if sequence numbers repeat after overflow
    * Preferable to have the maximum number PDU to be twice the maximum number of outstanding PDUs
  - Flow control
    * Function of receiving entity to limit the amount of data sent by transmitter
    * Stop-and-wait
      · Each PDU must be acknowledged before next one can be sent
    * Efficiency requires the transmission of a fixed number of PDUs without acknowledgement
    * Implemented in several protocols
  - Error control
    * Guard against loss or damage of data and control information
    * Implemented as error detection and retransmission
    * Detection of error by receiver makes him discard the PDU
    * No acknowledgement makes the sender retransmit the PDU
    * With error correction code, the receiver may be able to correct the error at destination
    * Implemented in several protocols
  - Addressing
    * Addressing level
      · Level in the communications architecture at which an entity is named
      · Network-level address or IP address used to route a PDU through network (also called Network Service Access Point or NSAP in OSI terminology)
      · Upon arrival at destination, the PDU must be routed to port or Service Access Point (SAP) for the application
    * Addressing scope
      · Global address

        Global nonambiguity One system to one address but possible to have more then one address for the system

        Global applicability Any system can be identified from anywhere
      · A port may not be unique in the network unless attached to a system (think of SMTP port on every system in a network)
    * Connection identifiers
      · Useful for connection-oriented data transfer (virtual circuit) but meaningless for connectionless data transfer (datagram)
      · Identified by a connection name during the data transfer phase
      · Advantages

        Reduced overhead Data packets can contain just the circuit number after a virtual circuit is established

        Routing Allows the setting up of a fixed route

        Multiplexing More than one connection between entities; incoming PDU can be identified by connection identifier

        Use of state information State information related to the connection; enables flow control and error control using sequence numbers
    * Addressing mode
      · Individual or unicast address – reference to a single system or port
      · Multicast or broadcast address
  - Multiplexing
    * Combining several signals for transmission on some shared medium

    - Possible to have multiple virtual circuits terminating into a single end system
    - Can also be accomplished via port names
    - Upward multiplexing
      - Multiple higher-level connections are multiplexed on a single lower-level connection
      - Connecting your PC to ISP for multiple applications, including web, email, telnet, ftp, ...
    - Downward multiplexing
      - Split a single higher-level connection over a number of lower-level connections
      - Useful for reliability, performance, or efficiency
  - Transmission services
    - Additional services, such as priority, quality of service, and security

OSI

- The OSI model
  - Partitions the communications model into a hierarchical set of layers
  - Each layer is a logical unit to communicate with the corresponding unit at a different host
  - The layer provides a level of abstraction, hiding details of its functions in lower layers and providing service to layers above it
  - Layers should have a clean interface so that changes in one layer do not affect the other layers
  - The goal is to keep each layer small but still, not to have too many layers
  - Seven layers in the model
  - No direct communication between peer layers except at the physical layer level
- Standardization within the OSI framework
  - Functions of each layer are well defined
    - Standards can be developed independently and simultaneously for each layer
    - Speeds up standards making process
  - Well-defined boundaries (interface) between layers
    - Changes in standards in one layer need not affect existing software in another layer
    - Easier to introduce new standards
  - Modular design of layers
    1. Protocol specification
       - Protocol must be precisely specified in terms of PDUs exchanged, semantics of all fields, and allowable sequence of PDUs
    2. Service definition
       - Services provided to next higher layer
       - Functions description of what is provided, and not how it is provided
         - Interaction between two adjacent layers takes place within a single open system and is not a concern of any other open system; *how* pales in comparison to *what*
         - Adjacent layers are usually implemented on the same processor; special hardware features can be exploited to make implementation more efficient
    3. Addressing
       - Network service access point (NSAP) to indicate the transport entity that is user of network service
       - Addressing as SAP allows each layer to multiplex multiple users from the next higher layer
- Service primitives and parameters

- Services between adjacent layers are expressed in terms of primitives and parameters
- Primitive
  * Specifies the function to be performed
- Parameters
  * Data and control information
- Four types of primitives
  1. Request
     * Issued by a service user to invoke some service and to pass the parameters needed to fully specify the service request
  2. Indication
     * Issued by the service provider to:
       (a) Indicate the the primitive has been invoked by the peer service user on the connection and provide associated parameters
       (b) Notify the service user of a provider-initiated action
  3. Response
     * Issued by a service user to acknowledge or complete some primitive previously invoked by an indication to that user
  4. Confirm
     * Issued by service provider to acknowledge or complete some primitive previously invoked by a request by the service user
- The time line indicates the sequence as specified above
- Confirmed service
  * Initiator receives confirmation that the requested service has had the desired effect at the other end
- Nonconfirmed service
  * Initiator receives no confirmation that the requested service has been carried out

- OSI layers
  1. Physical layer
     - Covers the physical interface between devices
     - Identifies the rules to pass bits from source to destination (raw bit stream service)
     - Four important characteristics
       (a) Mechanical
         * Physical properties of the interface to transmission medium
       (b) Electrical
         * Representation of bits in terms of voltage levels
         * Data transmission rates
       (c) Functional
         * Functions of individual circuits of physical interface between a system and transmission medium
       (d) Procedural
         * Sequence of events by which bit streams are exchanged
  2. Data link layer
     - Makes the physical link reliable, through error detection and control
     - Activates, maintains, and deactivates the link
     - Fully functional data link layer obviates the need for error control in higher layers
     - Communication through a number of data link layers may require the higher layers to perform some error control

3. Network layer
   - Transfers information across communications network, performing switching and routing functions
   - Hides underlying data transmission and switching technologies
   - Highest layer in a network node
   - System interacts with network
     * Specification of destination address
     * Request for network services like priority
   - In direct point-to-point network, there is no need for network layer as data link layer manages the link
   - Systems could be connected across a single network, using circuit switching or packet switching techniques
     * Packet level of X.25 standard
4. Transport layer
   - Mechanism for exchange of data between end systems
   - Ensures that data are delivered error-free, in sequence, and with no losses or duplication
   - May optimize the use of network services
   - Provides a requested quality of service to session entities, based on acceptable error rates, maximum delay, priority, and security
   - Size and complexity depend on the reliability of underlying layers
5. Session layer
   - Mechanism to control the dialogue between applications in end systems
   - Key services include
     * Dialogue discipline
       · Full duplex or half duplex
     * Grouping
       · Mark data to define groups of data
     * Recovery
       · Checkpoint to allow retransmission of all data since last checkpoint due to failure
6. Presentation layer
   - Format of data to be exchanged between applications
   - Defines syntax used between application entities
   - provides for selection and modification of the representation used
   - Data compression and encryption
7. Application layer
   - Interface between application programs and OSI environment
   - Management functions and other useful mechanisms for distributed applications support

TCP/IP protocol suite

- Reasons for TCP/IP's success
  - Time; appeared on the scene before the OSI model
  - Support from the DOD
  - Internet foundation
- The TCP/IP approach

  - Modular and hierarchical like the OSI model
  - Descriptive in nature compared to prescriptive nature of OSI
    * Allows multiple protocol functionality in a single layer
  - Does not require strict use of all layers
    * Application level protocols may directly run on top of IP

- Operation of TCP and IP
  - Computer is connected to network using a network access protocol such as ethernet
    * Enables host to send data across the network to another host or to a router to be transmitted to another network
  - Internet protocol
    * Implemented in all the end systems and routers
    * Acts as a relay to move data from one host to another, possibly through router(s)
  - Transmission control protocol
    * Implemented in the end systems only
    * Keeps track of data blocks to ensure reliable delivery to appropriate applications
  - Two levels of addressing
    * Unique host address over global internet, used by IP
    * Unique process (port) address within host, used by TCP
  - TCP header
    * Control information for data blocks generated by user application
    * Items in the header

      Destination port
      · Address to whom data is to be delivered

      Sequence number
      · Sequence number assigned to segment to keep track of segment order
      · Destination TCP entity may use it to reorder segments

      Checksum
      · Code to check error during transmission
  - IP datagram
    * Created by adding IP header to each segment
    * Items in header include destination host address
    * Presented to network access layer for transmission
  - Packet or frame
    * Created by network access layer by adding its own header to the IP datagram
    * Packet header contains information for network to transfer data across the network
    * Items in packet header are

      Destination network address
      · Device address for packet delivery

      Facilities request
      · Request for use of network facilities, such as priority

- Applications
  - Simple mail transfer protocol (SMTP)
    * Basic email facility

    * Mechanism to transfer messages across hosts
    * Features include mailing lists, return receipts, and forwarding
    * Does not specify message creation; just the transfer of message using TCP
  - File transfer protocol (FTP)
    * Transfer files across systems under user commands
    * Can accommodate both text and binary files
    * Upon request, sets up a TCP connection to target system for exchange of control messages
    * Connection allows user to send authentication and files with desired file actions
    * Upon approval, a second TCP connection is opened for actual data transfer
    * Second connection avoids the overhead of control information at the application level
    * After file transfer is complete, control connection is used to signal completion and accept new commands
  - Telnet
    * Remote logon capability
    * Designed to work with simple scroll-mode terminals
    * Implemented in two modules
      1. User telnet
         · Interacts with terminal I/O module to communicate with a local terminal
         · Converts characteristics of real terminals to network standards and vice versa
      2. Server telnet
         · Interacts with an application, acting as a surrogate terminal handler
         · Makes remote terminal appear as local to the application
    * Traffic between user and server telnet is carried on a TCP connection