Lecture 3

Implementation of Protocol Stack

<table>
<thead>
<tr>
<th>Layer</th>
<th>Subroutine Lib, User Code (in OS Commit Protocol)</th>
<th>Subroutine Library</th>
<th>System Process Outside OS or Subroutine Lib</th>
<th>InOperating System</th>
<th>Network Adapter Card</th>
<th>Chipset(IEEE802.x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataLink</td>
<td>Subroutine Lib, User Code (in OS Commit Protocol)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>Subroutine Lib, User Code (in OS Commit Protocol)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Subroutine Lib, User Code (in OS Commit Protocol)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Real Picture

- User 7
- Libs 7,6,5,4,3
- OS 7,3,2
- Hardware 1,2
Layering (see Fig 1.8, Bertsekas Gallager)

- General Concepts:
  -- Modules
    --- I/O description - blackbox
    --- off the shelf reusability
  -- Service
  -- Layered Protocol View
  -- Distributed Black Box
  -- Protocol = distributed algorithm

Physical Layer
- Virtual Bit Pipe (characterized by R bits/sec with error rate of 10^-4)
- Modems = Interface (noop’s if digital medium)
- DL interface:
  -- sync bit
  -- async bit
  -- async char
- Simple Protocol needed
  -- how many wires, voltages, pins
  -- how many microsec’s a bit lasts, if simultaneous, 2-way communication is ok.
  -- how does communication start and end?
  -- RS232C: example (fig 1.9), - each end is Finite State Machine and stores state
DTE

Req to Send

DCE

DCE Ready

Clear to Send

CaDet

One Wire per Signal (D.C.)
- **Data Link Control Layer (or Data Link Layer)**
  - Provides virtual, reliable, asynchronous variable length packet pipe on unreliable bit pipe

  - Packet = string of bits from layer 3

  - Asynchronous pipe in two ways
    -- Asynchronous delay through pipe due to:
      1) error recovery
      2) variable length packet

    -- Asynchronous insertion of packets due to:
      1) No data
      2) pipe full, due to errors
Frame:

<table>
<thead>
<tr>
<th>Header</th>
<th>Layer 2 bits</th>
<th>Trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of data</td>
<td></td>
<td>Checksum</td>
</tr>
</tbody>
</table>

-- 3 types
--- Data Frame
--- Ack Frame
--- Piggyback

DLC:
-- may regulate traffic (slow sender to receive rate or if no receive buffer)
-- always guarantees ordering
-- usually (not always) guarantees reliability (detect corrupted, lost, duplicated frames)

Question: If layer 2 guarantees reliable pipe, does layer 3 & above need error detection?

Answer: Yes, since it resets if 1 endpoint’s layer 2 crashes. Also if multihop packet.

Runs over
-- Point-to-point
-- Multiaccess (add source, destination address to header)
(satellite, radio, cable, optical fiber)
- In layer 2:

- Layer 2 Summary of Functions:
  - Error detection and recovery
  - Flow Control
  - Provides asynchronous interface for layer 3 (if layer 2 is synchronous or intermittent synchronous)
  - MAC (if multiaccess medium in layer 1)
  - Addressing (if multiaccess medium in layer 1)
• Layer 3: Network Layer

Could bridge 2 LANs or, be a backbone switch
Encapsulation:

- Network Layer’s jobs:
  -- Routing (which outgoing link?)
  -- Flow control

- Miscellaneous
  -- Network gets hit by packets from 2 directions:
    layer 4 and layer 2
Network can generate its own control packets:
--- Source quench
--- Say if a link failed
--- Net management
--- Establish, close connection

VC Networks operation - Layer 3 must:
-- Select path/route for VC upon connection establishment
-- Insure during connection that all packets follow selected path
-- Control packets to optimize route selection

Datagram Networks - Layer 3 must:
-- Route each packet individually

Network layer can provide:
-- Virtual connection-oriented service
   --- with virtual circuit op or datagram op
   --- Always in LAN, FIFO

-- Datagram connectionless service
   --- with datagram op
   --- only in WAN, Reordering

Examples:
--- ISO - connection-oriented and connection-less, both on X.25 VC
--- TCP (connection-oriented) and UDP (connection-less) both on IP (using datagrams)

Flow Control
-- Local access
   --- In MAC layer for an endpoint
--- In network layer in subnet node

-- Global problem
    --- Congestion control/management

-- Techniques
    --- Buffer management
    --- Access control to subnet
    --- Routing to spread load
    --- Bits in packets to pass info on congestion
    --- RTD measurement to detect congestion

-- Connection-oriented - easier, versus
    Connection-less - harder flow control

-- In higher data-rate, higher link capacity networks:
    --- no feedback before transmission ends maybe
    --- errant node can still flood net
    --- expect same utility, because new applications
        may eat up bandwidth

-- Internets (Internet sublayer)
    --- needed because there are multiple LAN
        protocols
    --- TCP/IP - SNA gateway

-- Question: If LAN uses 1500 byte DL packets, LAN2 uses 512, which layer splits/assembles packets?
• Transport Layer (layer 4)

- Reliable end-to-end delivery (if network is unreliable) (TCP)

- Breaks messages into packets, reassembles

- Multiplex low data rate sessions onto same network session

- Split one high data rate transport session into multiple network sessions

- End-to-end flow control, recovery from link/node failures

- Sometimes transport flow-control/error recovery overlaps/fights network, DLC (want integrated solution)

- Operating system implements transport
Layer 5: Session layer

- Allows ordinary data transport (as layer 4 does) with enhancement

- Who pays for service?

- Load sharing on multiple processors

- Access rights (remote login)

- 411 service: look up where in network a service is available (IP address, TCP-port)

- Note: "Session" initiation divided between layers 5, 4, 3 (following from Tanenbaum 7.1)

- Only the OSI suite has it; other protocols either don’t provide its functions or do it in other layers

- Seven groups of OSI service primitives:
  1&2) Connection establishment and release (like layer 4 except):
      --- one TL connection can be used for many short sessions
      --- release is "orderly" at layer 5, "abrupt" at layer 4

  3) Data transfer (like layer 4)

  4) Token management (whose turn is it to talk?)
5) Synchronization: move session entities back to known state if an upper layer error occurs (layer 4 only recovers from common errors in layer 1 - 3). Roll back to last synchronization point and re-start.

6) Activity management (partition message stream into files)

7) Exception reporting.

• Layer 6: Presentation
  - Translation to/from universal data representation (float, char) (Suns, XDR)
  - Encrypt, decrypt
  - Data compression, decompression (video file, image, multimedia network)

• Layer 7: Application
  - File transfer
  - Mail exchange
  - Transaction protocol
  - RPC (requires processes to marshall data)
  - Virtual terminal (how to move cursor)