Computer Networks

27-1-15

- Considerations for layer division in a network implementation

  → New layer is created when a conceptually different level of abstraction is needed

  → Layer must perform a well defn set of fns; no fn overlap between layers

  → Layer fn chosen/defn to fully support internationally standardized layer protocols

  → Layer boundaries should be well defined and chosen or ~~on~~ designed to minimize the volume of data flow across inter-layer interface, but provide just the needed level of information interchange

  → Number of layers, ~~layer~~ should be large enough to accomodate conceptually/semantically distinct fn in distinct layers & small enough to avoid unwieldy/unmanageable design

  → No. of layers, layer naming and layer protocol(s) should conform with international standards, if any

- The OSI reference model - proposed by ISO (also called ISO OSI model)

- Open System Interconnect model for interconnecting open systems (any vendor, any H/W, any OS/software platform) - it has seven layers

- The OSI reference model - not a complete description
  it does not specify
  - → Exact service primitives for each layer
  - → Exact inter layer interfaces
  - → Exact set of protocol / protocol suite to be implemented at each level, but does provide broad guidelines for semantic level at each layer
  - → ISO has seperate standards for each layer (service primitives, interfaces, protocols)
  - → These are separate standards and not necessarily part of OSI model specifications

- OSI is definitely not the best possible design
  - → Some layers are very complex; have to provide complex services, i.e., D/L, N/W & to some extent, transport layer
  - → Some layers are very lightly loaded (have to provide relatively simple services) eg., session and Presentation layers - some implementations even avoid having these layers

- OSI model is very relevant
  - → Standardization of layers by a non-commercial international body
  - → Provide protocol stds for each layer (published as separate standards, distinct from network specification)
  - → These layer protocols are extensively used in existing operational networks

The OSI reference model - proposed by ISO

fig ~~1-30~~ 1-20

→ Physical layer

- Txm. / reception of data bit / byte streams ; shared or P2P links ; prone to errors, noise, bit dropping, etc.
- Phy representation of '1s' & '0's - voltage / current levels, current direction, high low or low high ~~tx~~ transitions
- tx^m rules : simplex, half duplex, full duplex, etc.
- HW details - electrical & procedural mechanical, connector type pin loc. & fn

→ D.L. Layer

- Error control - uses varied types of error prone, noisy & unreliable phy. tx^m medium. Provides error free (almost), reliable (ack.) data tx^m to network layer
- Logical data unit - creates simplest logical data unit - data frame ; N/W layer packets split into smaller size frames, each separated by a frame bdry.
- Sequence control - frames arriving out of order stored (store & fwd) packet formed by reordering after all frames (error-free) are received
- Flow control : sender - receiver speed diff managed
  - Handshaking
  - Pre decide tx^m speed upper bound
- Congestion control : too many frames in 1 dir. i.e. Ack frames from A and B (for B to A data) & A to B data frames, many re tx^m because of high tx^m error rate (wireless sat. links)
  - Problem reduced partially by flow control & 'piggybacking' of "Ack" frames on regular data frames

- Diff. QOS: links with diff levels of reliability, error detection and recovery, robustness, speed, etc. realized from inherently error prone and unreliable layer links

- Two sublayers
  - Medium access control (MAC): creates reasonably error-free and reliable virtual pt to pt links from shared multiaccess broadcast phy. links
  - Logical link control (LLC): responsible for fast reliable & efficient tx^{m} of frames over pt. 2 pt. links (phy & virtual)

→ <u>Network layer</u>

- Virtual end to end channel – hide comm subnet from transport layer & create a reliable, error free virtual comm. channel for every host pair

- Routing: The task of finding an ordered set of links & routers in comm. subnet to form a path from source host to dest host (& back for Ack packets)

- Routing types: based on static routing tables built into the network, i.e., a ckt switching approach where actual switching is never done

  - Path setup (a connection) before a conversation, eg, a terminal session (byte stream) or file transfer (message stream) and dismantled after it is over i.e., ckt switching for conn.-oriented service

  - Path setup one link ~~&~~ at a time & recomputed each time a packet arrives at a new router in subnet, i.e., store & fwd/packet switching for conn. less service (ack. or unacknowledge)

- Congestion control - attempts to regulate packet flow and avoid too many packets on the subnet, i.e., keep total packet flow rate less than comm. subnet ~~b/w~~ between

- Congestion control & routing complement each other
  - Good routing avoids ~~alread~~ already congested routes and sends packets through lesser used ones (still achieving fewer delays)
  - This reduces congestion on heavily used routes and increases utilization of used ones

- Accounting / billing f^n.
  - Comm. subnet normally owned and operated by 3rd party or renumeration basis

  - Renumeration can be based on diff. parameters i.e., volume of data transfer speed & bandwidth usage duration, reliability, accuracy level, etc

  - Simplest mechanism would require N/W layer to keep track of no. of packets routed from / to each host

  - For a network spanning diff nations accounting gets more involved (billing rationale varies between countries)

- Transparent internetworking - packet travelling through diff N/Ws
  - Address scheme / format may (&do) vary between N/Ws
  - Packet size and format may vary

P.T.O

- Layer protocol may vary (more than one protocol specified for some layers on OSI model)
- N/W layer takes care of these & related aspect to implement transparent interconnection of heterogenous networks and hide details of transport layer

Transport layer

- ~~or~~ True source to dest or end to end layer
- ~~All~~ A tr. layer entity on a host is conn. to corr. entity on all other hosts in the N/W through direct virtual links
  - Diff from lower layers - layer entities of diff hosts connected by chain of virtual links (layer protocols) and routers

---

3-2-15

- Msg to packets:
  - Source host entity accepts variable (large) sized msg. (SPDU) from session layer and breaks into smaller sized units (TPDU) before handing down to N/W layer
- Packets to message:
  - Dest. host entity ensures reliable and error free receipt of all packets (from N/W layer) puts them in sequence and hands over complete msg. to session layer
- Connection multiplexing - Session layer request for connection (transport)

- Normal throughput -
  N/W layer requested to establish a single source dest. logical path (conn. oriented or ~~connection~~ connection-less, i.e, ckt or packet switched, unack. or ack), ~~as~~ usually full. duplex

- High throughput - packets multiplexed through several normal throughput transport connections, multiplexing at source host and demultiplexing at dest. host done by transport layer

- Low throughput - several low throughput connections realized with a single normal throughput N/W connection, multiplexing / demultiplexing overhead borne by transport layer

• Several types of services offered to session layer
  - Fully reliable, error free strict message order end to end connection
  - Highly reliable and error tolerant (not 100%) connection that maintains message ordering (i.e., unacknowledged connection oriented)
  - Fully reliable, error free conn. with no guarantee on msg. ordering (ack, conn-less)
  - Limited reliability and error tolerant conn. with no guarantee on msg. ordering
  - Broadcast / multicast ~~connection~~ connection

• Flow control
  - Avoids buffer ~~flow~~ overflow in fast source - slow dest-host pair or underflow in slow source - fast dest. pair
  - Implemented at a higher semantic level than the D/L layer, but same general handshaking ~~protocol~~ approach used

~~Sessio~~

Session Layer

- Allows host peer processes to establish sessions
  - Client process on one host logging on to server process on remote host, a request reply session followed by a log out
  - A file transfer between 2 hosts
- Allows usual end-to-end byte/msg stream, but at higher semantic level compared to transport layer
- Provides enhanced services to facilitate sessions b/w remote host processes
  - Dialogue control - regulate control flow/direction provide arbitration, keep track of whose turn it is to speak next
  - Token mgt. - provides synchronization & mutual exclusion when required by layer protocols
    - Session layer provides token to requesting host process; token can be exchanged between peer processes, process holding token can perform critical op. requiring mutual exclusion (eg: distributed DB access)
    - ~~For~~ Peer processes synchronized by exchanging token (process holding token can run, while others wait for its exchange)
  - Rollback / Crash recovery - provides checkpoint and file backup abilities
    - Checkpoints inserted during long file transfers and system crash during such transfer taken care of by restarting transfer from last successful checkpoint
    - Remote file/database update done only after successful data backup; system crash managed by restarting update on uncorrupted backup copy after creating ~~set~~ another backup