Chapter 2

Protocol Architecture/ Multiplexing

What is a protocol?:-

- Protocol is a set of rules the exchange or transmission of data electronically between devices
- To perform a task,
  - The protocol is just a set of rules.
  - Different tasks use different protocols

Examples of Protocol:-

- Two protocols will be studied in your first lab
  - HTTP is the protocol used for browsing website
    - Web browser <----> Web server
    - http://www.youtube.com/watch?v=1IQFjTnDozo&feature=related
  - SMTP is the protocol used for sending emails
    - Email client software < ---- > SMTP server

Need For Protocol Architecture:-

- E.g. File transfer
  - Source must activate comms. Path or inform network of destination
  - Source must check destination is prepared to receive
  - File transfer application on source must check destination file management system will accept and store file for his user
  - May need file format translation

- Task broken into subtasks
- Implemented separately in layers in stack
- (stack is a linear data structure which follows a particular order in which the operation are performed.(LIFO))
- Functions needed in both systems
- Peer layers communicate

Key Elements of a Protocol:-

- Syntax
  - Data formats
  - Signal levels
- Semantics
  - Control information
  - Error handling
- Timing
  - Speed matching
  - Sequencing

Protocol Architecture:-

- Task of communication broken up into modules
- For example file transfer could use three modules
  - File transfer application
  - Communication service module
  - Network access module

Protocol Functions:-

- Small set of functions that form basis of all protocols
- Encapsulation
- Fragmentation and reassembly
- Connection control
- Ordered delivery
- Flow control
- Error control
- Transmission services

Encapsulation:-

- Data usually transferred in blocks
  - Protocol data units (PDUs)
  - Each PDU contains data and control information
  - Some PDUs only control
- Three categories of control
- Address
  - Of sender and/or receiver
- Error-detecting code
  - E.g. frame check sequence
- Protocol control
  - Additional information to implement protocol functions

Fragmentation and Reassembly (Segmentation – OSI):-

- Exchange data between two entities
- Characterized as sequence of PDUs of some bounded size
- Application level message
- Provision(Rule) of checkpoint and restart/recovery operations
- PDU arrival generates interrupt
  - Smaller blocks, more interrupts
- More time processing smaller.

Reassembly:-

- Segmented data must be reassembled into messages
- More complex if PDUs.

Connection Control:-

- Connectionless data transfer
  - Each PDU treated independently
- Connection-oriented data transfer
- Connection-oriented preferred (even required) for lengthy exchange of data
  - Connection establishment
  - Data transfer
  - Connection termination
  - May be interrupt and recovery phases to handle errors

Data Transfer and Termination:-

- Both data and control information exchanged
  - e.g. flow control, error control
- Data flow and acknowledgements may be in one or both directions

Sequencing:-

- Many connection-oriented protocols use sequencing
  - e.g. IEEE 802.11
- PDUs numbered sequentially
- Supports three main functions
  - Ordered delivery
  - Flow control
  - Error control

Transmission Services:-

- Protocol may provide additional services to entities
- Connection basis
- Security:-
- Security mechanisms, restricting access