

Bottom line: we want to be able to send information from a source to a destination reliably.

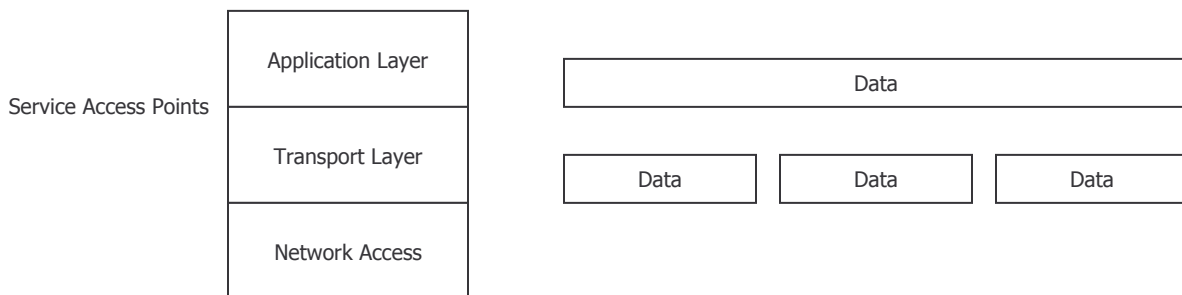
Ensuring Reliability in Transmission

- Handshaking (a form of flow control)
- Error checking
 - CRCs – Cyclic redundancy checks
 - Checksums
 - Parity
- Error correction
 - Ability to request re-transmission
 - Ex: HDLC – High level data link control
- Correct transmission rate
 - If transmitter is sending at 9600 baud, the receiver must be set up to receive at that rate. If the rates don't match, a **framing** error can occur.

We don't want all this handled in one big monolithic program. The application which sends e-mail doesn't want to know the details of how it gets to the destination. It just wants to send it reliably. Thus, the big program is broken into smaller modules, usually referred to as **layers**.

These layers are independent of one another and are arranged in what looks like a stack. Each layer communicates only with the layer above or below it.

Hypothetical Protocol Stack



Application Layer – different programs on the computer that want to communicate with another computer. Example: game, e-mail, etc.

Port Addresses are used by the transport layer to know which application to send the data to. The combination of a port address and the computer address (ie: IP address) is a **socket**.

Transport Layer – takes the big chunk of data from the application layer and breaks it down into small chunks (packets) that are small enough to be sent across the network being used. Adds header information, stuff like the port address.

When receiving data, it is responsible for reassembling the packets into one big chunk of data (in the proper order!!!).

Performs error checking and correction.

Network Access – (or physical layer) – the layer that is actually connected to the physical network. Is responsible for getting the data off and on to the physical medium.
Handles the exchange of data between the network and the computer.

Needs to know what the bit rate is.

Encodes the data at this point, if necessary.