

Network Software

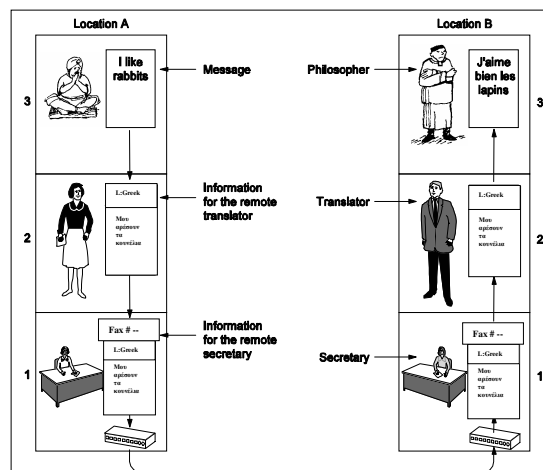
Layers
Messages
Design
Primitives

- Network Layers
- Message
- Design Issues
- Service Primitives

Sample Communications

Layers
Messages
Design
Primitives

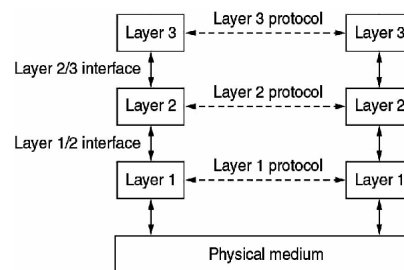
- Philosopher
 - Protocol
- Translator
 - Services
 - Interfaces
 - Protocols
- Secretary
 - Services
 - Interfaces
 - Protocols



Network Layers

Layers
Messages
Design
Primitives

- Network Software is organized into layers to...
 - Reduce design complexity,
 - Allow layers to be independent of one another and hence easily changed
- Each layer
 - Provides services to the layer above
 - Uses services from the layer below
- Interfaces for the services make available primitives from the lower layers to the upper.
- From an end-to-end perspective communication is between peers at the same protocol layer.

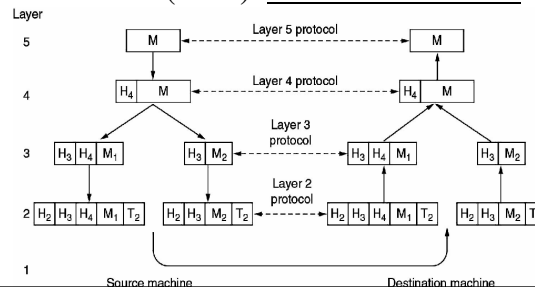


3

Messages

Layers
Messages
Design
Primitives

- Control info
 - Each layers adds a header and possible a tail of control information to the message.
 - Used by the corresponding layer in the receiver and then stripped off the message before being passed to the next highest layer.
- Protocol Data Unit (PDU): Control Info + Data



4

Some Design Issues (1)

Layers
Messages
Design
Primitives

- Overall
 - Type of comms: Simplex Communication, Half-Duplex, Full-Duplex Communication
 - Connection: Connection Oriented (telephone) vs. Connectionless (postal letters).
 - Confirmation: May be required even by connectionless services.
 - Quality of service: Needs consideration. Reliable or unreliable?
- Error control: How to locate them and what to do with them.

Some Design Issues (2)

Layers
Messages
Design
Primitives

- Messages
 - Addressing: We need some way to identify the station to which we are transmitting.
 - Routing: How do we get a message from A to B?
 - Max Length: Messages may need to be fragmented because of restrictions places by different layers.
- Flow control
 - Overloading: The receiver can be slower that the sender causing overloading.
 - Ordering: particularly where they have been broken up.
 - Multiplexing: As communication channels are shared, it is often necessary to put messages together for transmission down a line and separate them at the other end.

Sample Service Primitives

Layers
Messages
Design
Primitives

- Connection-oriented or connectionless comms?
- Primitives:
 - Listen: Block waiting for an incoming connection.
 - Connect: Request a connection with a waiting peer.
 - Receive: Block waiting for an incoming messages.
 - Send: Send a message to a peer.
 - Disconnect: Terminate a connection.

