Network Software Architecture and its Layers and Protocols

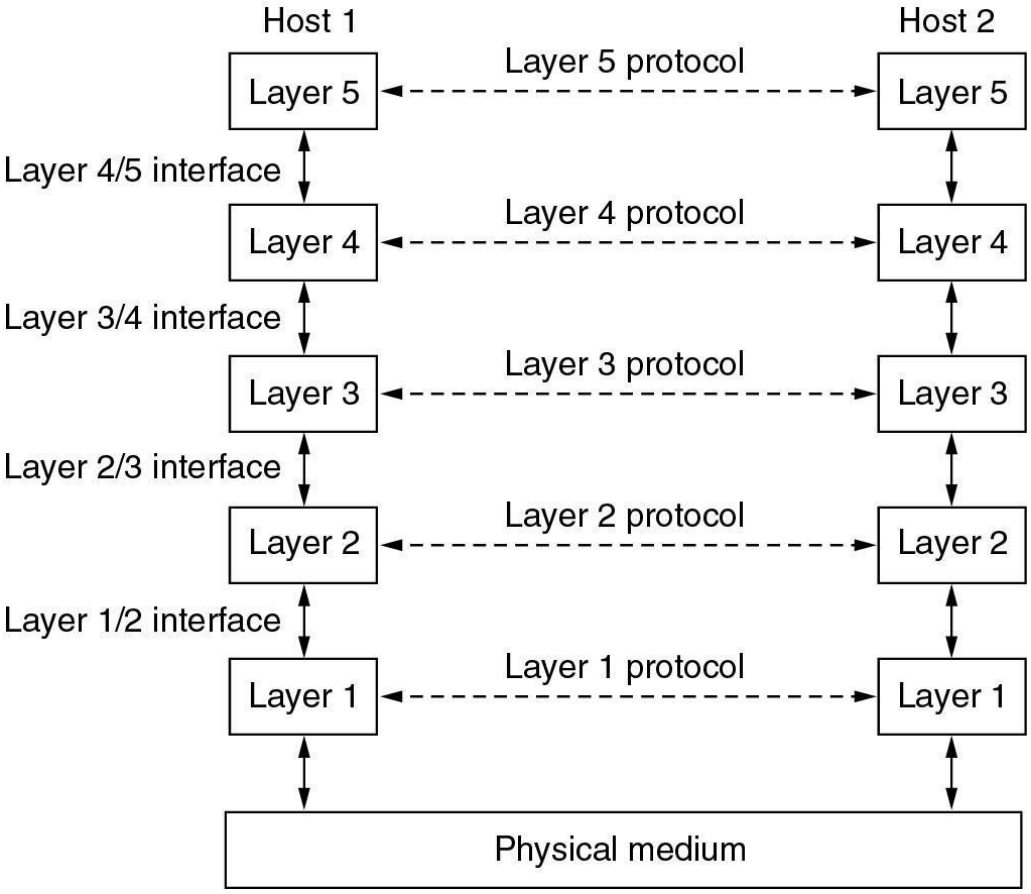
CSE306

Network Software

- Protocol Hierarchies
- Design Issues for the Layers
- Connection-Oriented and Connectionless Services
- Service Primitives
- The Relationship of Services to Protocols

Network Software

Protocol Hierarchies



- Layers, protocols, and interfaces- **Network Architecture**

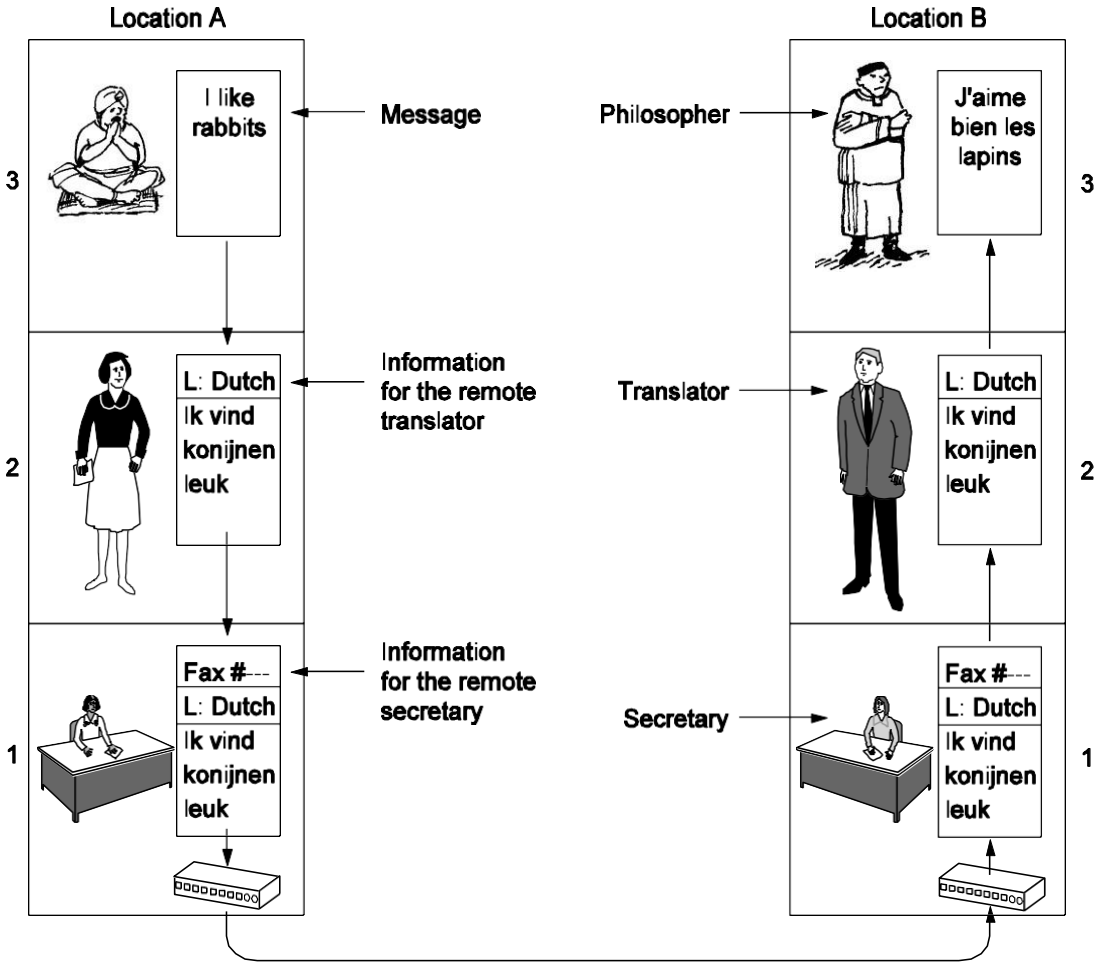
- A **protocol** is an agreement between the communicating parties on how communication is to proceed.

- The entities comprising the corresponding layers on different machines are called **peers.** The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

- A list of the protocols used by a certain system, one protocol per layer, is called a **protocol stack.**

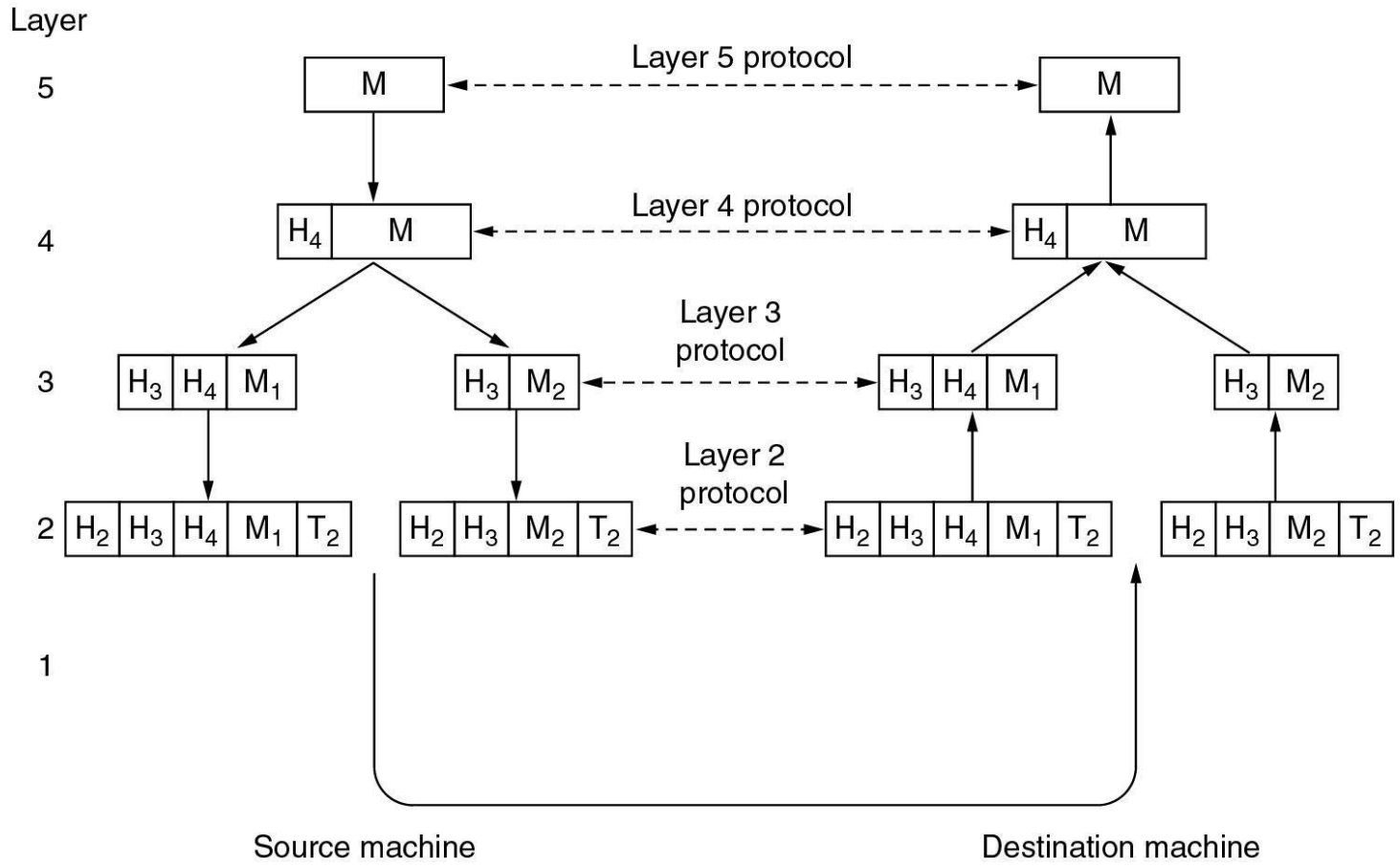
Protocol Hierarchies (2)

- The philosopher-translator-secretary architecture.



Protocol Hierarchies (3)

- Example information flow supporting virtual communication in layer 5.



Design Issues for the Layers

- Addressing or naming
- Error Control
- Flow Control---*Congestion*
- Statistical Multiplexing
- Routing
- Scalable
- QoS---*real time*
- Reliability
- Security

Connection-Oriented and Connectionless Services

- A **circuit is another name for a connection with associated resources,** such as a fixed bandwidth.
- This dates from the telephone network in which a circuit was a path over copper wire that carried a phone conversation.
- In contrast to connection-oriented service, **connectionless service is modeled** after the postal system.
- Each message (letter) carries the full destination address, and each one is routed through the intermediate nodes inside the system independent of all the subsequent messages.
- **Store or forward switching**
- **Cut through switching**

- Each kind of service can further be characterized by its reliability. Some services are reliable in the sense that they never lose data.
- Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived.
- Reliable connection-oriented service has two minor variations: **message sequences and byte streams.**
- The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.
- One such application is digitized voice traffic for voice over IP.
- Unreliable (meaning not acknowledged) connectionless service is often called **datagram service.**

Connection-Oriented and Connectionless Services

- Six different types of service.

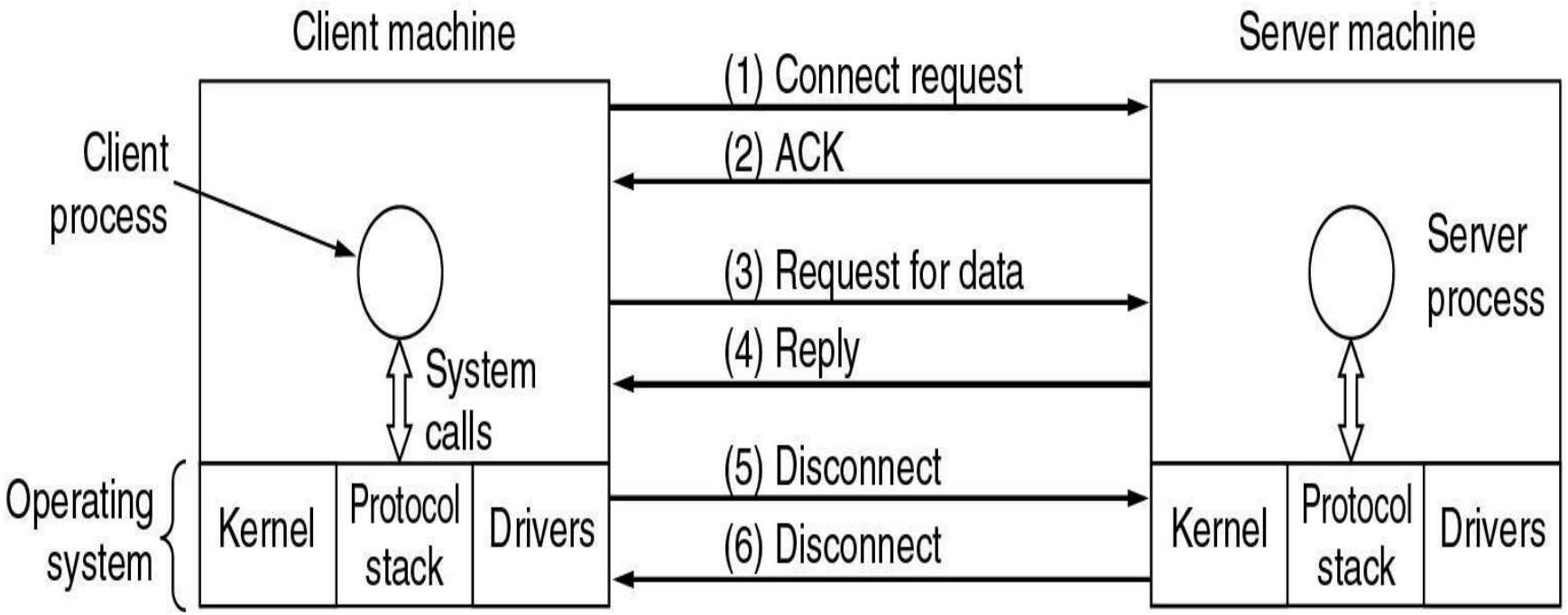
	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Service Primitives

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

- Five service primitives for implementing a simple connection-oriented service.

Service Primitives (2)



- Packets sent in a simple client-server interaction on a connection-oriented network.

Services to Protocols Relationship

- The relationship between a service and a protocol.