Lecture 4
LAYERING ARCHITECTURE OF NETWORKS

Topics Covered

- Network Software
- Protocol Hierarchies
- Layers, protocols, and interfaces
- Design Issues for the Layers
- Addressing
- Error Control
- Multiplexing
- Routing
- Connection-Oriented and Connectionless Services

Network Software

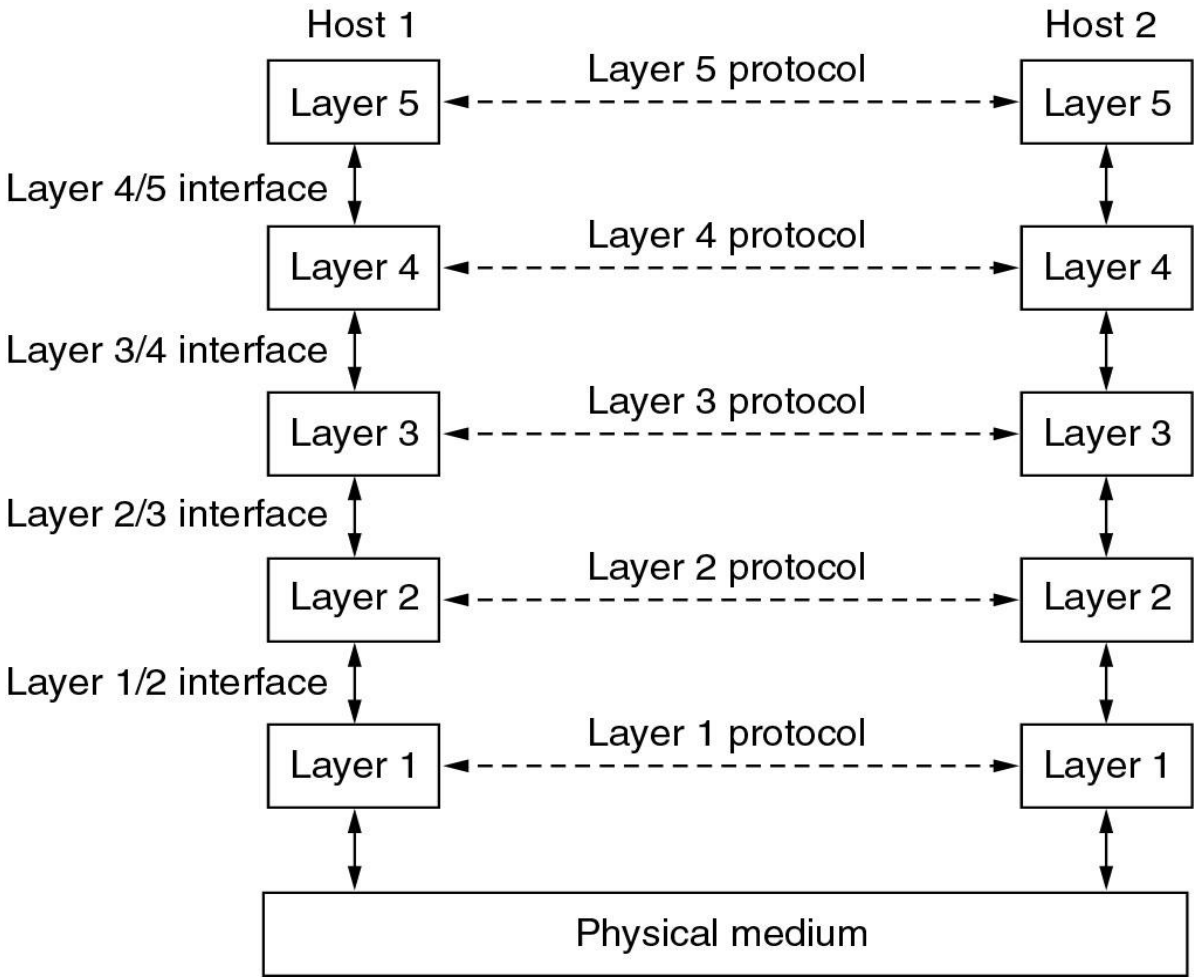
- Protocol Hierarchies
- Design Issues for the Layers
- Connection-Oriented and Connectionless Services
- Service Primitives
- The Relationship of Services to Protocols

Protocol Hierarchies

- In order to *reduce the design complexities* most *networks are organized* as a stack of ***layers*** **or levels**
- The number of layers, the name of each layer, the contents of each layer, and the function of *each layer differ* from network to network.
- The purpose of each layer is to offer certain services to the higher layers hiding the details how the offered services are actually implemented.
- In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

- Layer n on one machine carries a conversation with layer n on another machine
- Rules and conventions used during this conversation are known as Layer n protocol
- Protocol is an agreement between the communicating parties on how communication is to be proceed

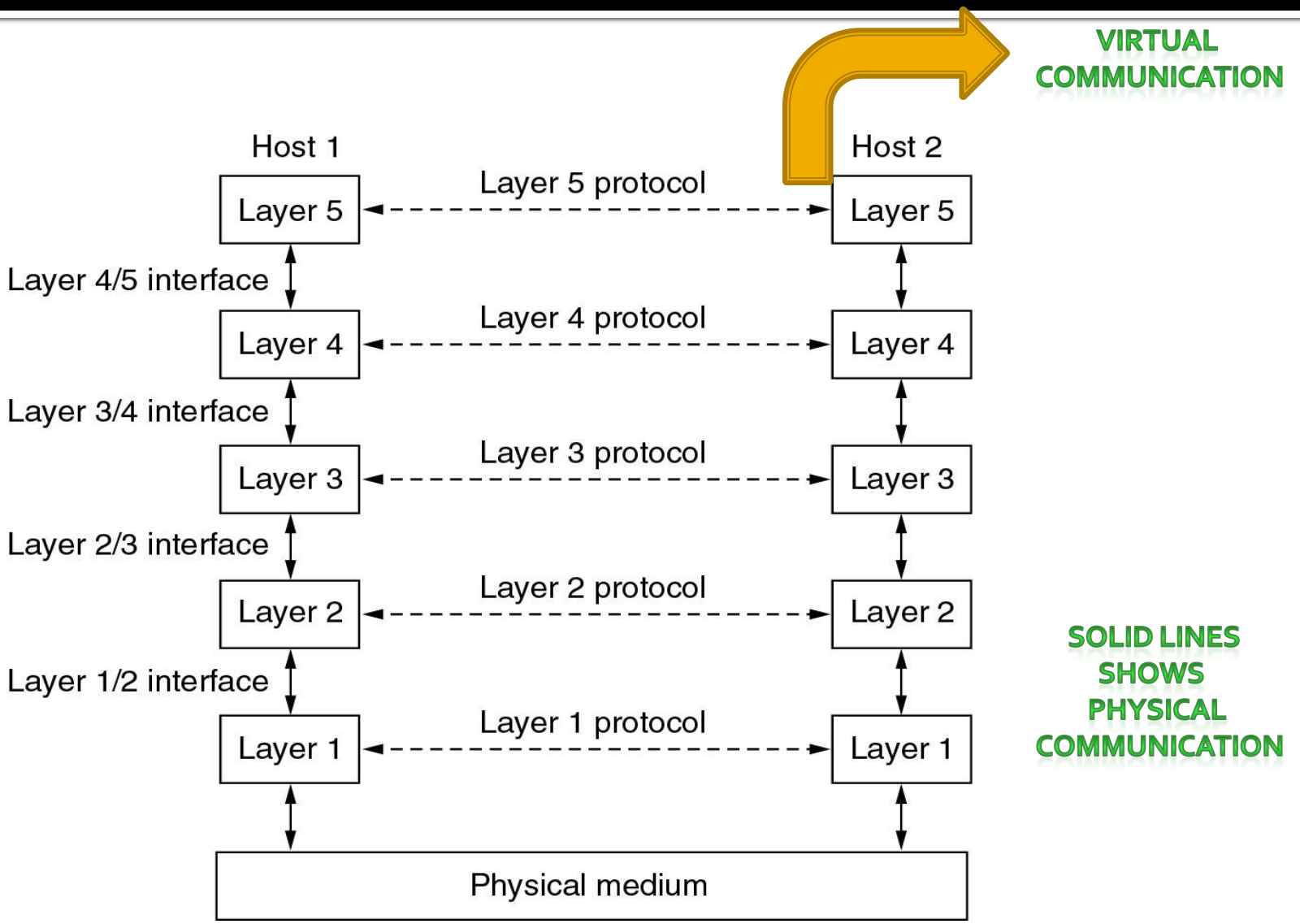
Layers, protocols, and interfaces



In reality

- No data is directly transferred from layer *n on one machine to* layer *n on another machine.*
- Each layer passes data and control information to the layer immediately below it, until the lowest layer is reached.
- Below --- layer 1 is the **physical medium through which actual communication occurs**

Layers, protocols, and interfaces.

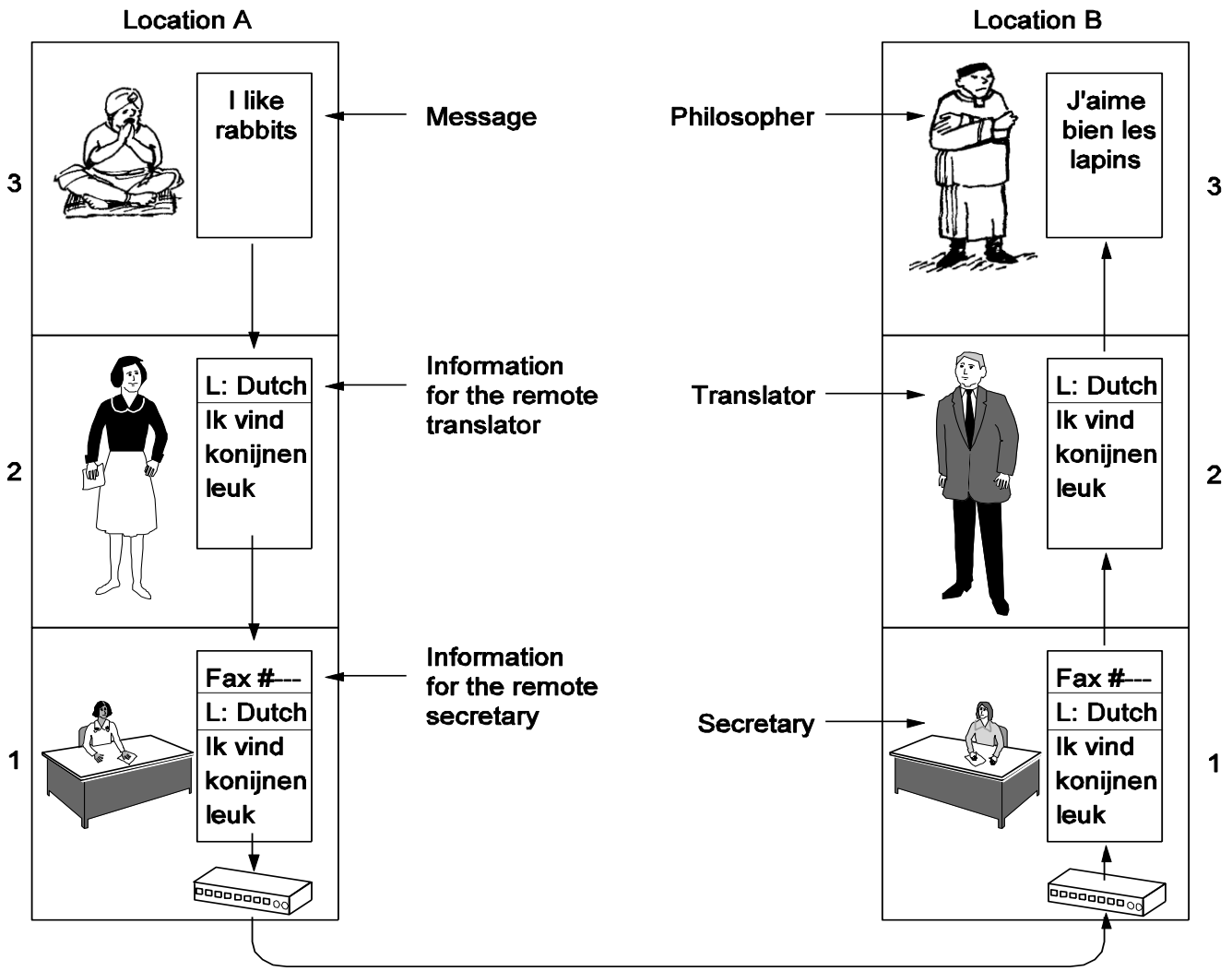


EACH LAYER PERFORMS
A SPECIFIC FUNCTION
AND
THERE IS A CLEAN
INTERFACE BETWEEN THE
LAYERS

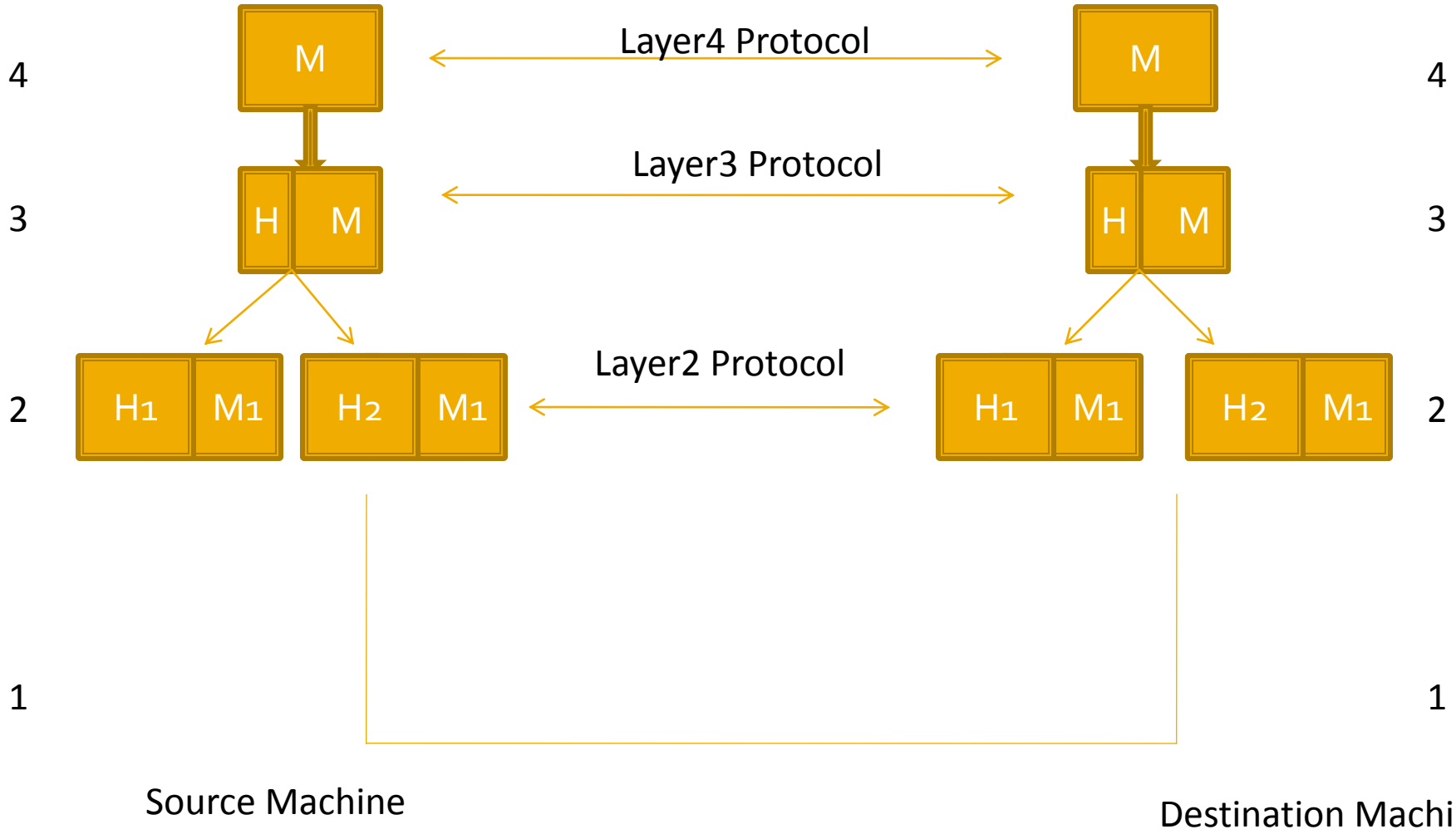
EXAMPLE

- Two philosophers (layer 3), one of whom speaks Urdu and English and one of whom speaks Chinese and French.
- Since they have no common language, they each engage a translator (layer 2)
- Translators in turn contacts a secretary (layer 1).
- Philosopher 1 passes a message (in English) **across the 2/3 interface** to his translator, saying ‘‘Ilike rabbits,’’
- The translators **have agreed on a neutral language** known to both of them, Dutch, so the message is converted to ‘‘Ik vind konijnen leuk.’’ **The choice of language is the layer 2 protocol and is up to the layer 2 peer processes.**
- The translator then gives the message to a secretary for transmission, by, for example**, fax (the layer 1 protocol).**
- When the message arrives, it is translated into French and passed across the 2/3 interface to philosopher 2.
- **Each protocol is completely independent of the other ones**
- The translators can switch from Dutch to say, HINDI, provided that they both agree, and neither changes his interface with either layer 1 or layer 3.
- Similarly, the secretaries can switch from fax to e-mail or telephone without disturbing (or even informing) the other layers.
- Each process may add some information intended only for its peer. This information is not passed upward to the layer above.

The philosopher-translator-secretary architecture.



EXAMPLE


4
3
2
1
M
H
M
H1
M1
H2
M1
Layer4 Protocol
Layer3 Protocol
Layer2 Protocol
M
H
M
H1
M1
H2
M1
4
3
2
1
Source Machine
Destination Machine

Design Issues for the Layers

- Addressing
- Error Control
- Flow Control
- Multiplexing
- Routing

Addressing

- A Network has many computers
- Some means is needed to specify with whom sender wants to talk.
- Since multiple destinations are there, -----some form of **addressing** is needed in order to specify a specific destination.

- **Rules for data transfer**
  - In some systems, data only travel in one direction; in others, data can go both ways
  - The protocol must also determine how many channels the connection corresponds to and what their priorities are.
  - Many networks provide at least two channels per connection, one for normal data and one for urgent data.

Error Control

- **Error control is an important issue because physical communication circuits** are not perfect.
- Many error-correcting codes are known, but both ends of the connection must agree on which one is being used.
- Also the receiver must have some way of telling the sender which messages have been correctly received and which have not.

Issues like----

- Not all communication channels preserve the order of messages sent on them.
- To deal with a possible loss of sequencing, the protocol must make explicit **provision for the receiver to allow the pieces to be reassembled properly**.
- An obvious solution is to number the pieces

Another issue is.....

- Fast Sender and Slow receiver

- Solns like acknowledgement
- Other solutions -------
  - limit the sender to an agreed-on transmission rate. This subject is called **flow control.**

- Inability to accept long messages.
- This property leads to mechanisms for disassembling, transmitting, and then reassembling messages

Multiplexing

- To set up a separate connection for each pair of communicating processes is inconvenient or expensive
- the underlying layer may use the same connection for multiple, unrelated conversations
- Multiplexing is needed in the physical layer

Routing

- When there are multiple paths between Source & Destination– A Route must be chosen.
- Sometimes this decision must split over two or more Layers.
- High Level Decision Vs. Low Level Decision based on current traffic load, Known as Routing.
- Following a proper route to come to conclusion.

Connection-Oriented and Connectionless Services

- Layers can offer two types of service to the layers above them

Connection oriented service

- Modeled after Telephone System
- You pick up phone---dial num---talk—n hang up
- Similarly connection oriented service first establish the connection---uses the connection and then releases it
- In most cases bits arrive in the same order as released.
- In some cases sender and receiver negotiate about parameters like maximum message size, quality of service etc

Connectionless Service

- Modeled after a postal service
- Each message carries full destination address
- Each one is routed through the system independent of all the others
- Order may not be necessarily followed