TU/e

CI0121 Computer Networks

Layering and protocol stacks

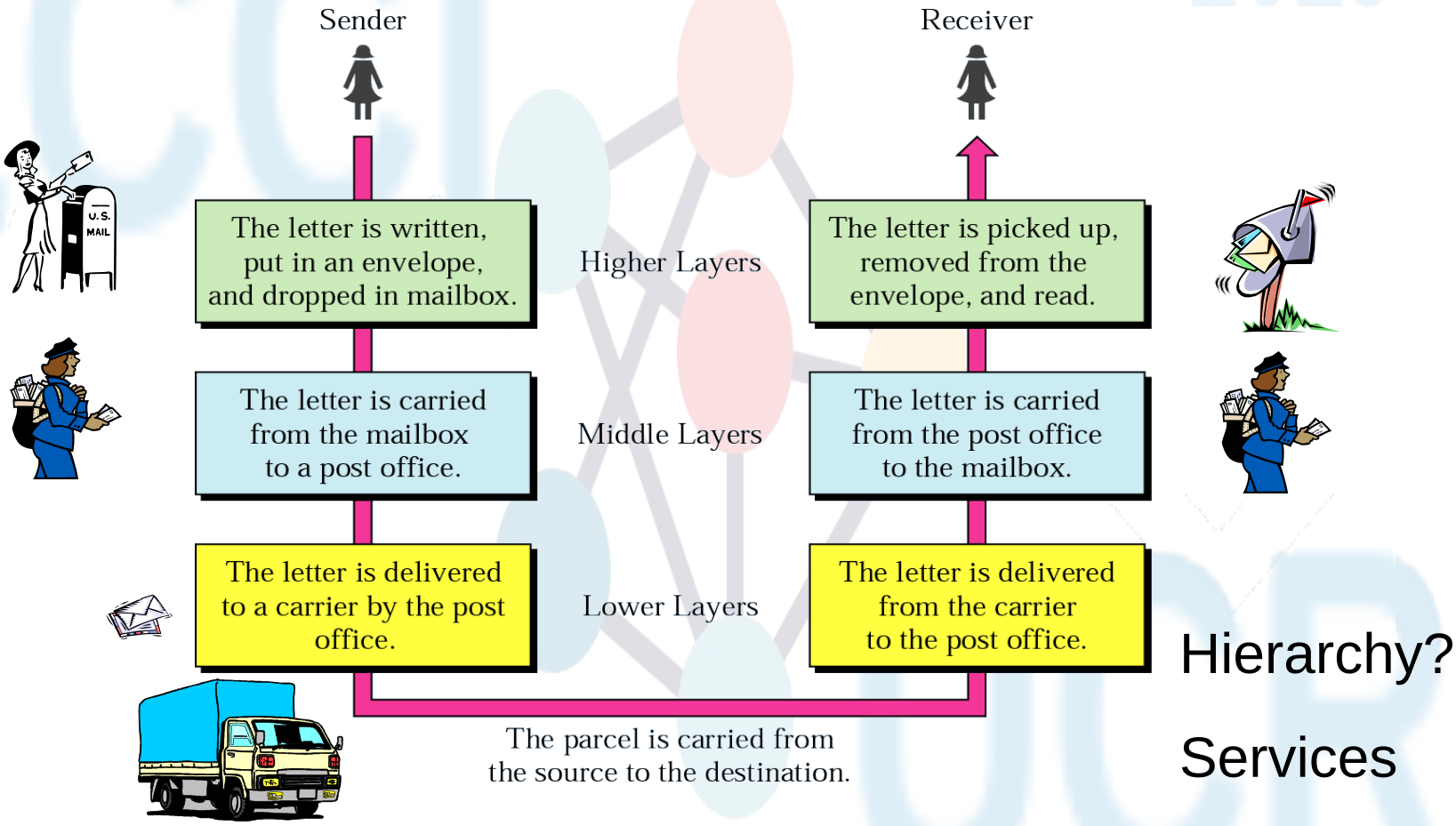
Profesores ECCI

Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives
- OSI layers
- TCP/IP
- Model from the book

Layered tasks

An example from the everyday life



Hierarchy?

Services

Layered communication system

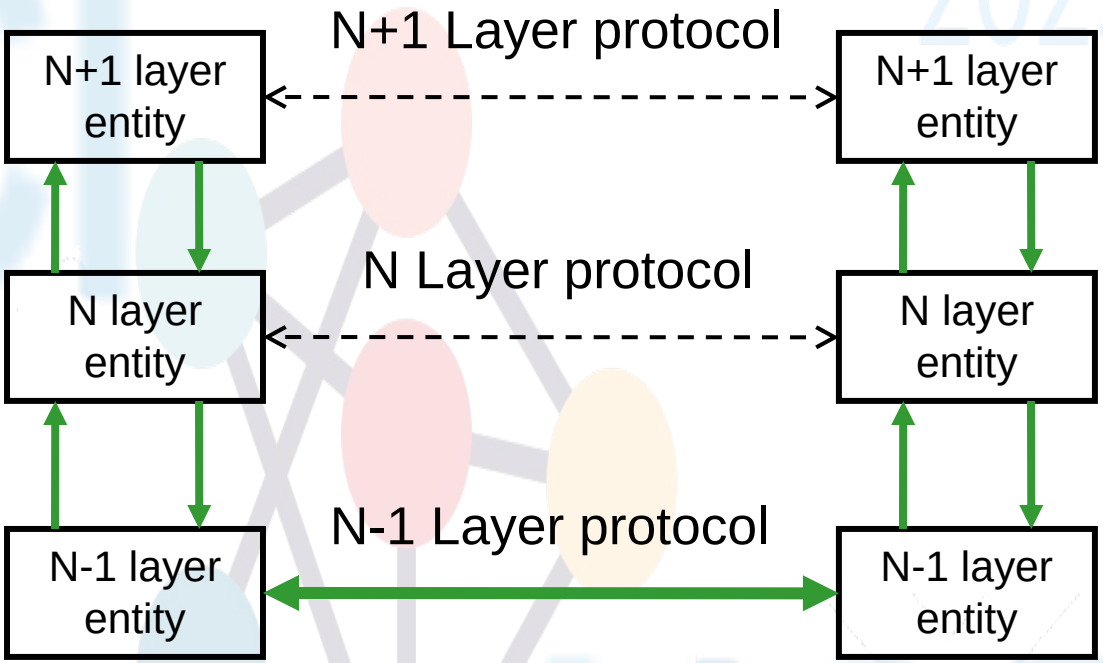
Layer N+1

Interface between layers N+1 and N

Layer N

Interface between layers N and N-1

Layer N-1



Entities from the same layers - peers

A layer

- In telecommunications networks and open system architecture:
  - a group of related functions that are performed in a given level in a hierarchy of groups of related functions.

- In object-oriented design:
  - a *group* of classes that have the same set of (link-time) module dependencies to other modules, i.e. a *collection* of reusable components that are available for reuse in similar circumstances

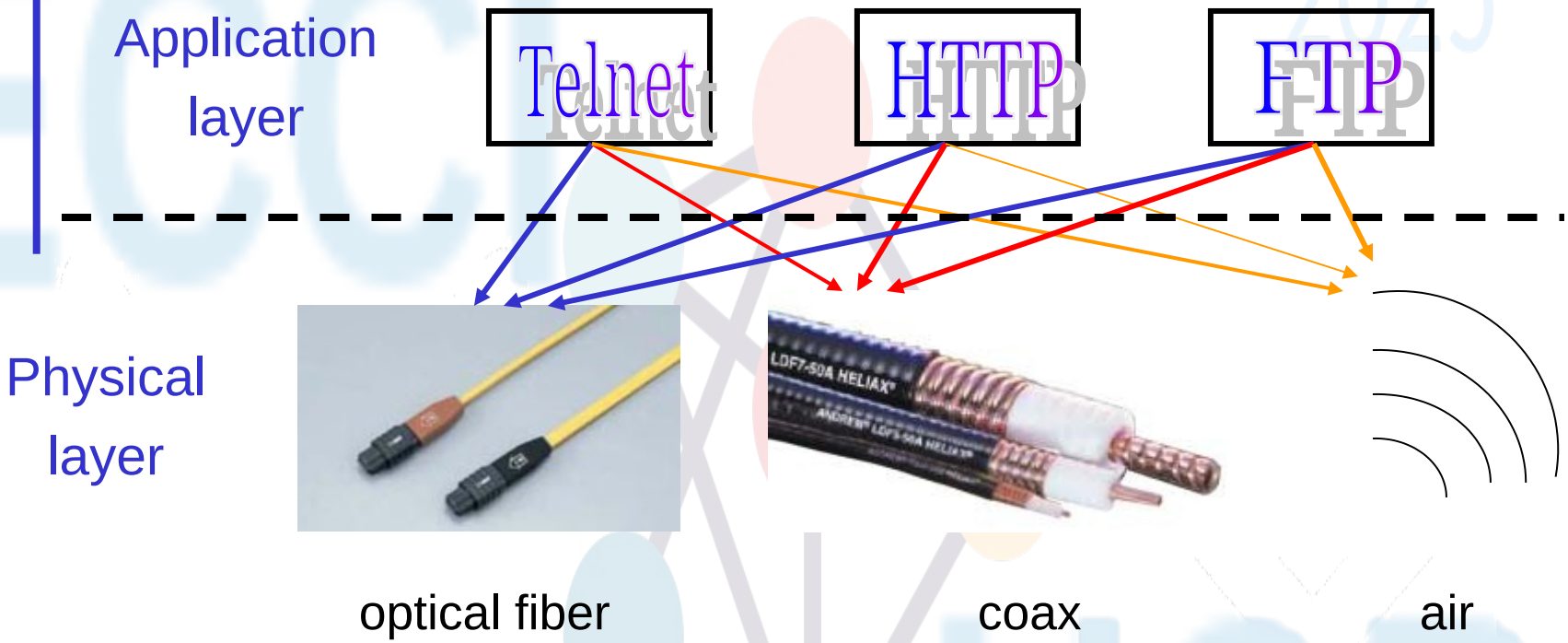
http://en.wikipedia.org/wiki/Layer

Why layered communication?

- To reduce complexity of communication task by splitting it into several layered small tasks
- Functionality of the layers can be changed as long as the service provided to the layer above stays unchanged
  - makes easier maintenance & updating

- Each layer has its own task
- Each layer has its own protocol

Why layering?

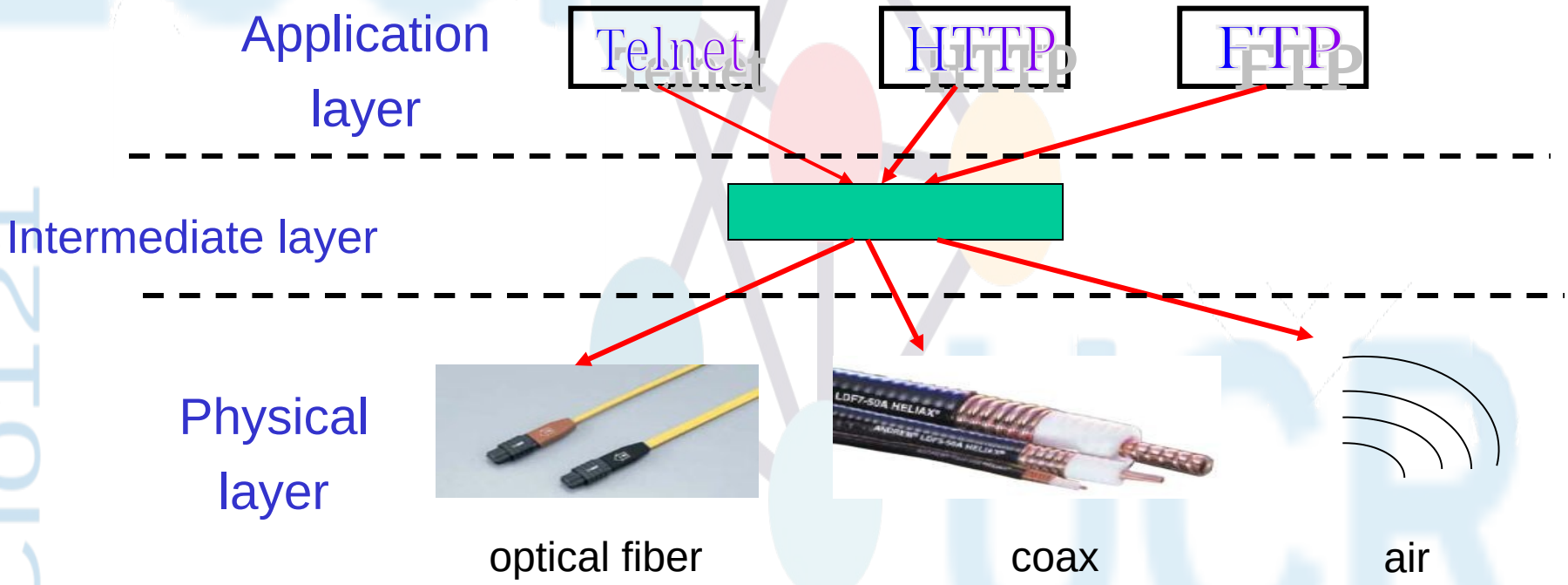


Each new application has to be re-implemented for every network technology

Benefit of layering

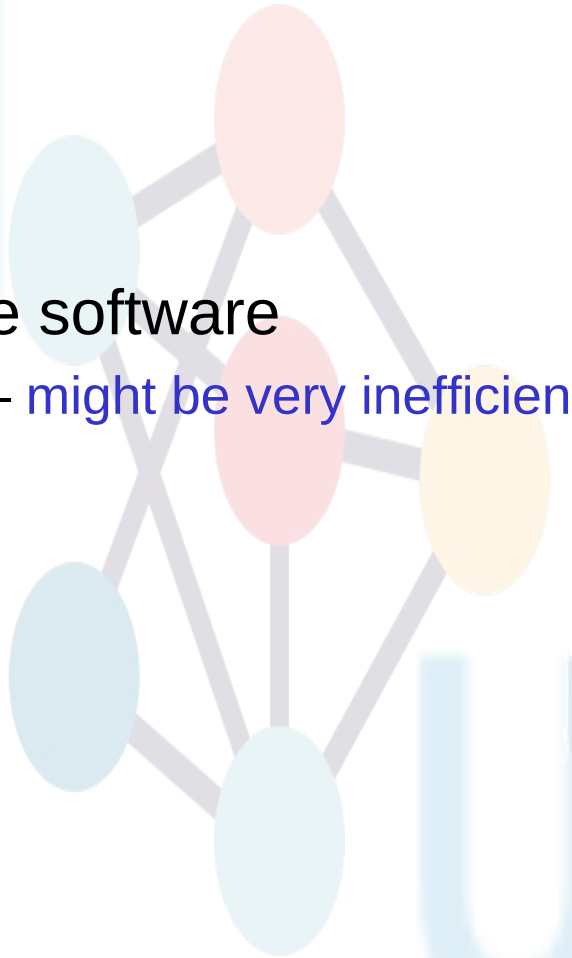
Solution to the problem:

introduce an *intermediate layer* that provides a common abstraction for various network technologies



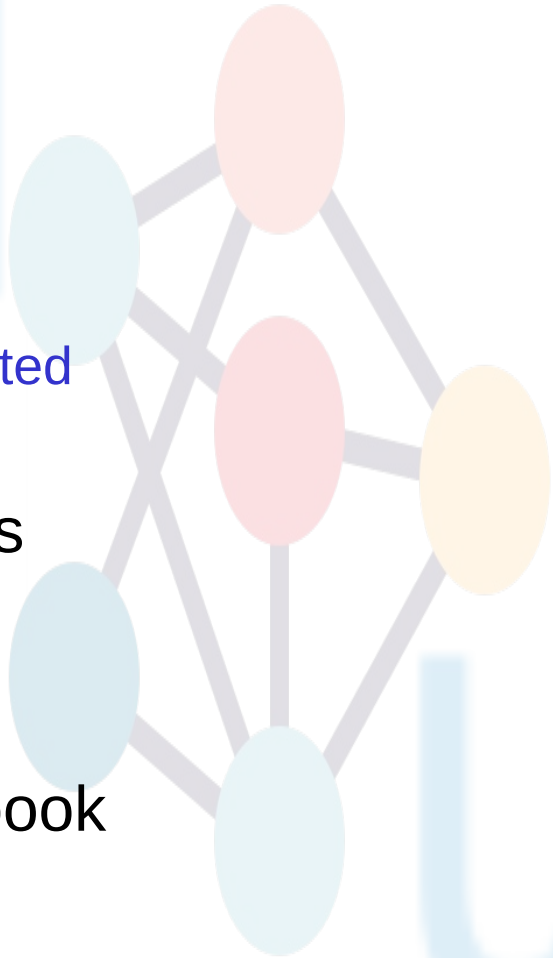
Disadvantages of layering

- Are there any?
- YES!
- Inefficiency of the software
  - Transport layer – might be very inefficient if not aware of the lower layers



Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives
- OSI layers
- TCP/IP
- Model from the book



What is a protocol?

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

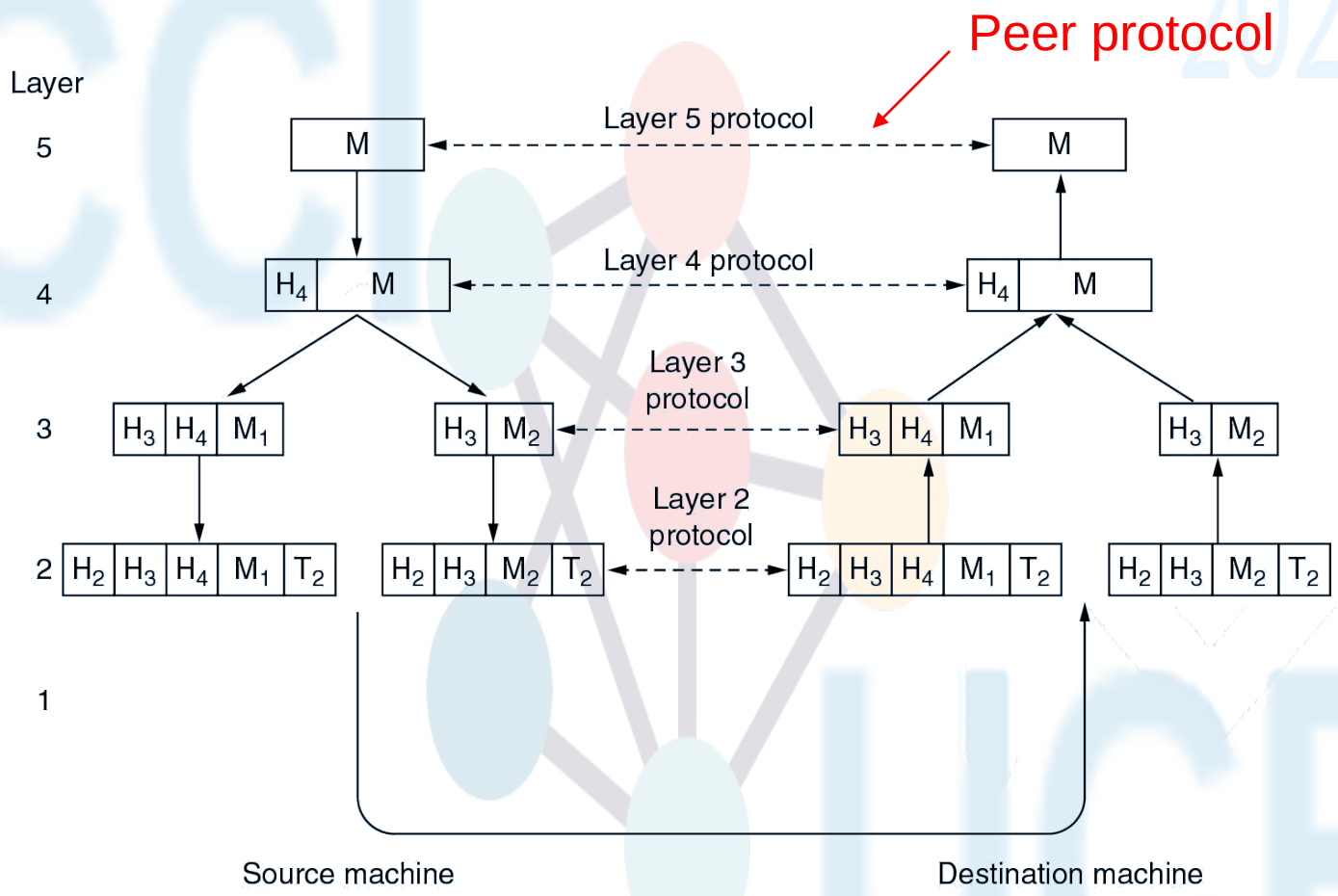
- "A protocol defines format & the order of messages exchanged as well as the actions taken on the transmission/reception of a message." (Kurose, Ross)

Analogy: politician meeting, PhD defense ceremony

A protocol is a set of rules that specify

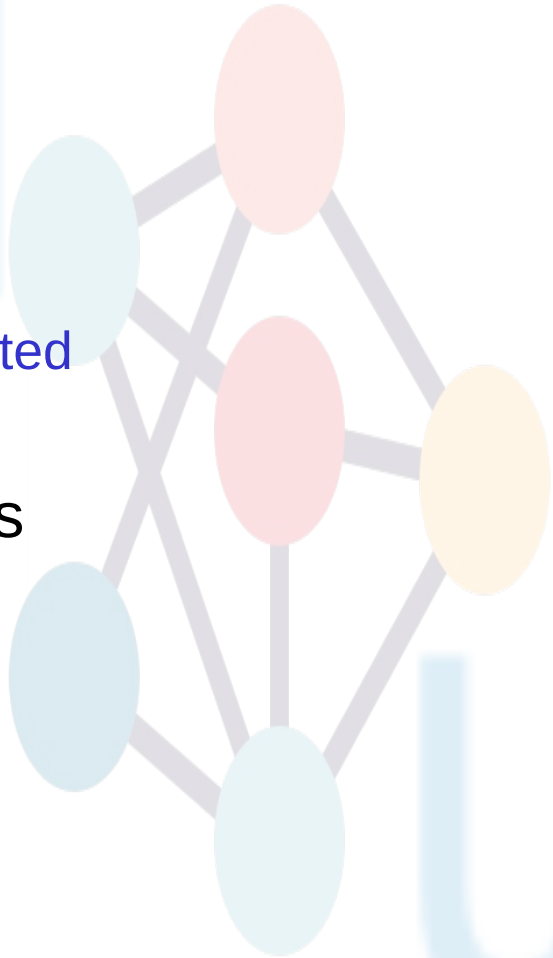
- the format of the messages exchanged
- a number of different protocol states and what messages are allowed to be sent in each state;
- these states determine, among others, the order of the messages, timing constraints and other non-functional properties, if any
- Example: HTTP, FTP, TCP…

Example: Protocol stacks



Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives
- OSI layers
- TCP/IP



What is a service?

WordReference.com Dictionary – “only” 27 definitions

1 an act of help or assistance

2 an organized system of labour and material aids used to supply the needs of the public

example: *telephone service*

example: *bus service*

3 the act or manner of serving food

4 (*Tennis*) (*squash*) the act, manner, or right of serving a ball

the game in which a particular player serves

example: *he has lost his service*

Layered network architecture and services

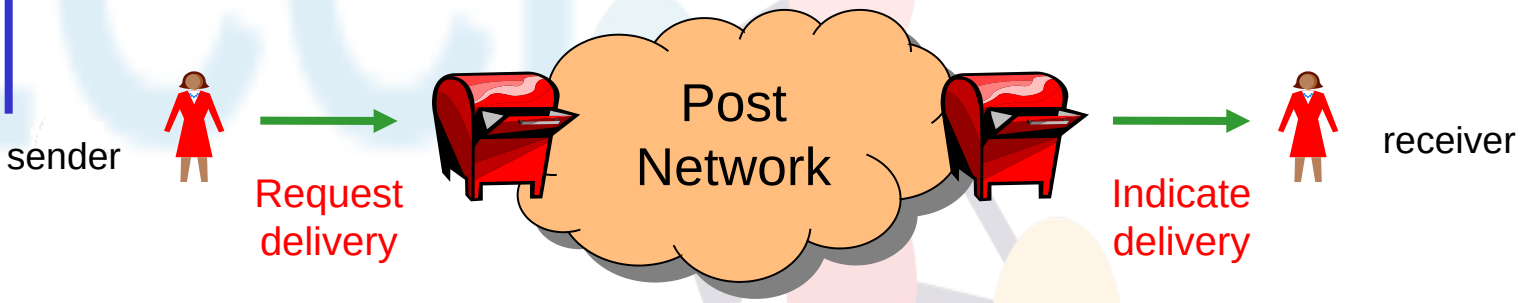
- Layer N **uses** services of layer N-1- service user
- Layer N-1 **provides** services to layer N- service provider
- From the example: service = post delivery

- **Service:** *a contractually specified overall functionality (semantics) of an object*
- **Service quality**: *non-functional properties of a service (e.g. speed, reliability, ...)*
- **Service interface**: *actions ("primitives") and responses that make the service available; these responses can be autonomous ("events")*

Network services

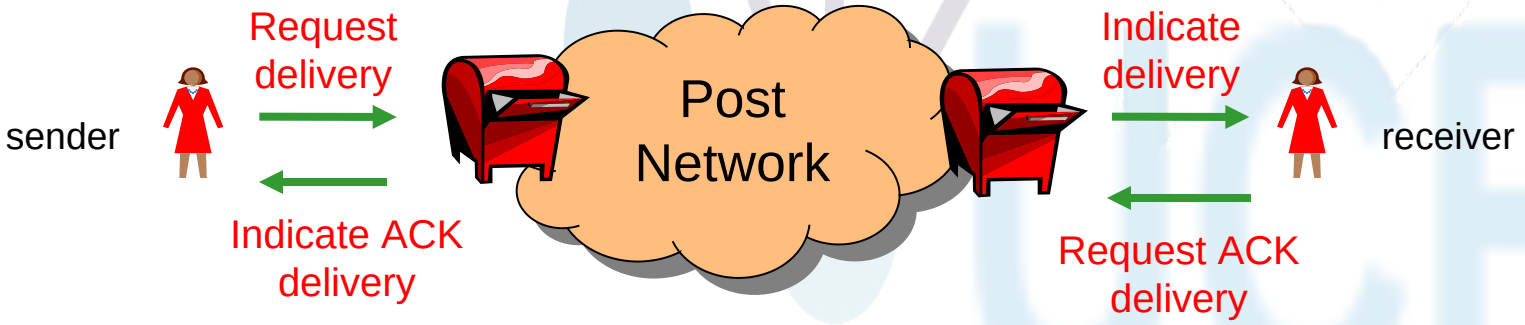
Services provided by different layers

- Unconfirmed (**Best effort**) service: *no feedback if delivery occurs*



- Confirmed (**Acknowledged**) service:

  sender gets a confirmation (acknowledgement) of delivery

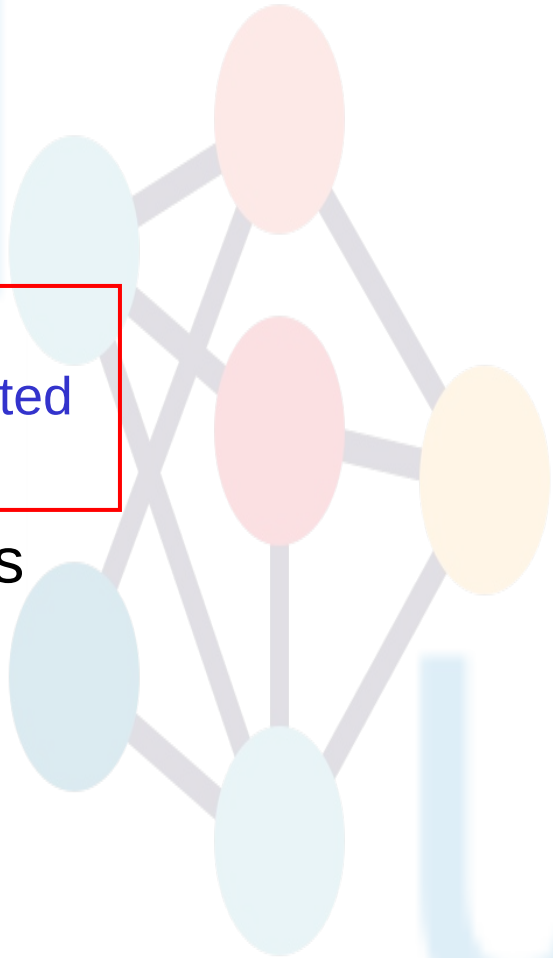


Network services (cnt'd)

- Unreliable services
  - No guaranteed delivery (no acknowledgments)
  - An example: a basic service of datagram networks
- Reliable services
  - Guaranteed delivery
  - Implementation of this service through combination of *timers, acknowledgment and retransmission*
  - An example: FTP, E-mail
- Why would anyone use an unreliable service?

Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives
- OSI layers
- TCP/IP



Connection-oriented services

- Establish connection
- Use connection
- Release connection

- Protocol Data Units (PDUs) are delivered in-sequence of transmission without duplication
- Implementation of this service:
  - Virtual-circuit packet switched network
  - In datagram networks, a connection-oriented service can be accomplished by end systems with sequence numbers, retransmission, and other mechanisms

Example: Service of **TCP** protocol,

Frame relay – for connecting LANs

X.25 – Typically across telephone lines

Connection less service

Example: Postal system

- No guarantee of in-sequence delivery. Losses are possible.

- Implementation of the service
  - default service for datagram communication
  - inefficient to implement in circuit-switching networks and virtual circuit packet switching networks

Example: services of both the IP and UDP protocol

Question: What applications can work with connectionless service?

- Connection-oriented service typically provided by the layer above

Synchronous

- Service with fixed delays and known error rate
- Implementation of the service:
  - This service is the “native service” of a circuit-switched network
  - Very difficult to implement in a packet-switched network
- Example: Telephony and Television

Quality of service (QoS)

- Some services must guarantee bounds to one of the following performance parameters
  - End-to-end delay (mean or maximum)
  - Delay variations (delay jitter)
  - Throughput
  - Loss-rate
- Implementation of the service
  - In a packet switched network, requires algorithms for admission control, traffic conditioning and packet scheduling
- Example: Multimedia applications require QoS guarantees

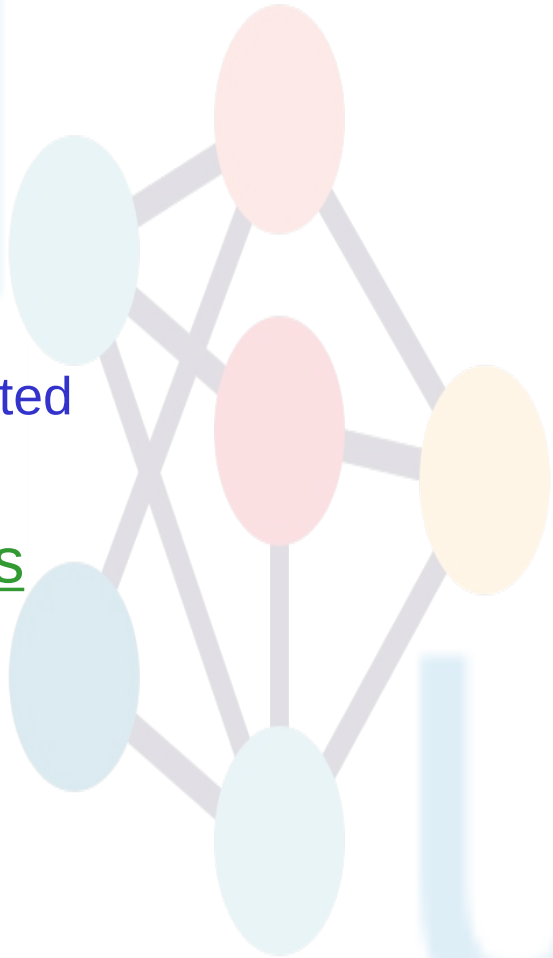
Service types

	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

Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives

- OSI layers
- TCP/IP

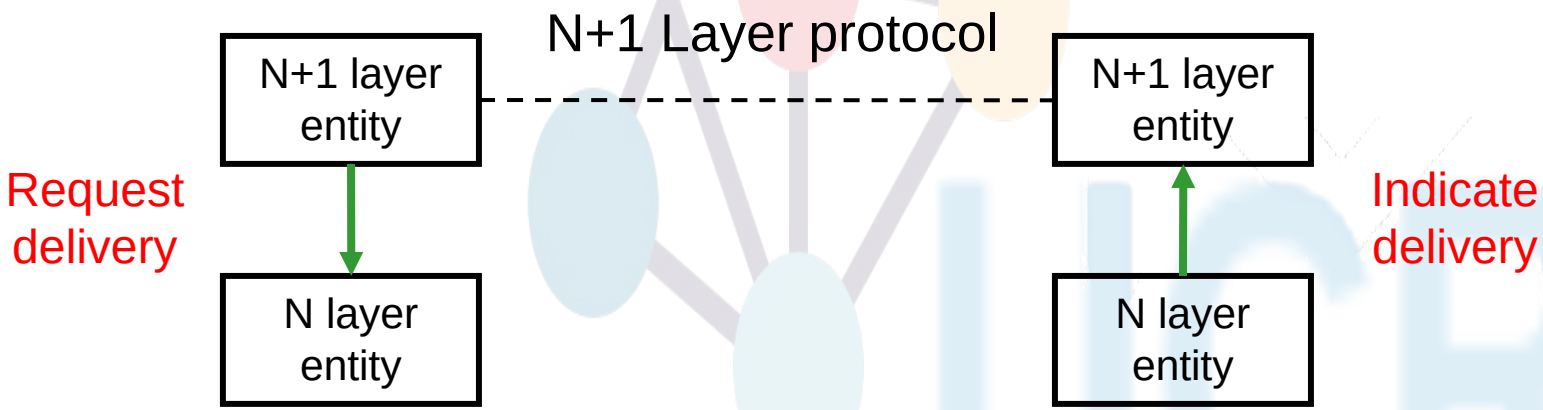


Service primitives

Usually system calls

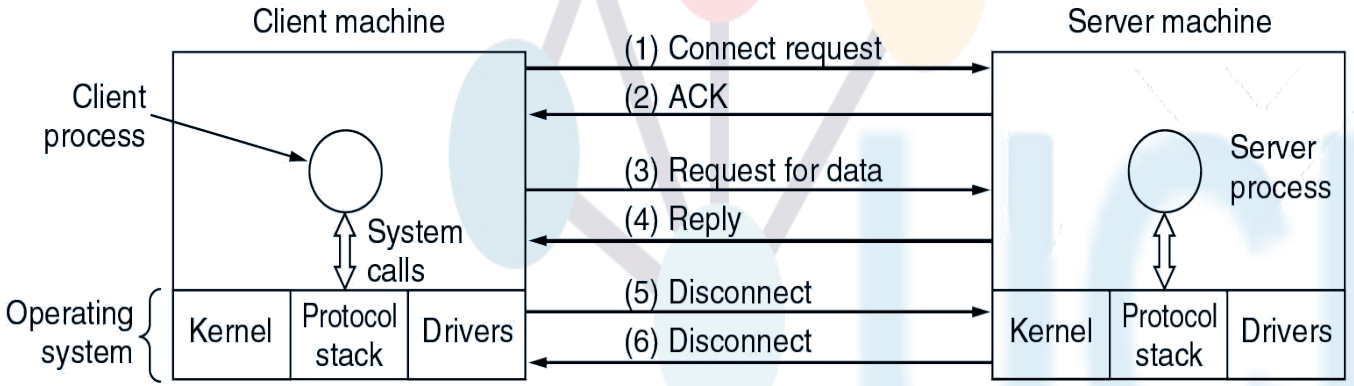
"..A **service** is a set of **primitives** (operations) that a layer provides to the layer above it…" (Tanenbaum)

Service defines what operations are to be performed but says nothing how they are performed



Example:
primitives for connection-oriented service

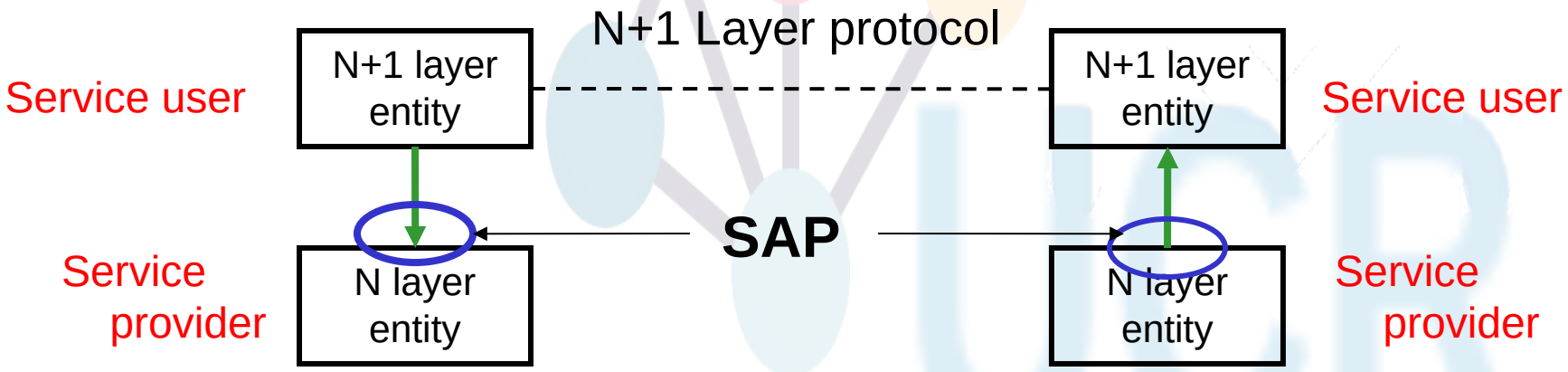
Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection



Service access points

A point at which a designated service may be obtained

- The term for the component of a network address which identifies the individual application on a host which is sending or receiving a packet
- TCP/IP's equivalent term is “port"
- Different SAPs distinguish between different services or applications on a host, e.g. e-mail, FTP, HTTP (from FELDOC)



Analogy

Question

Define service, service access point and quality of service at the gasoline station

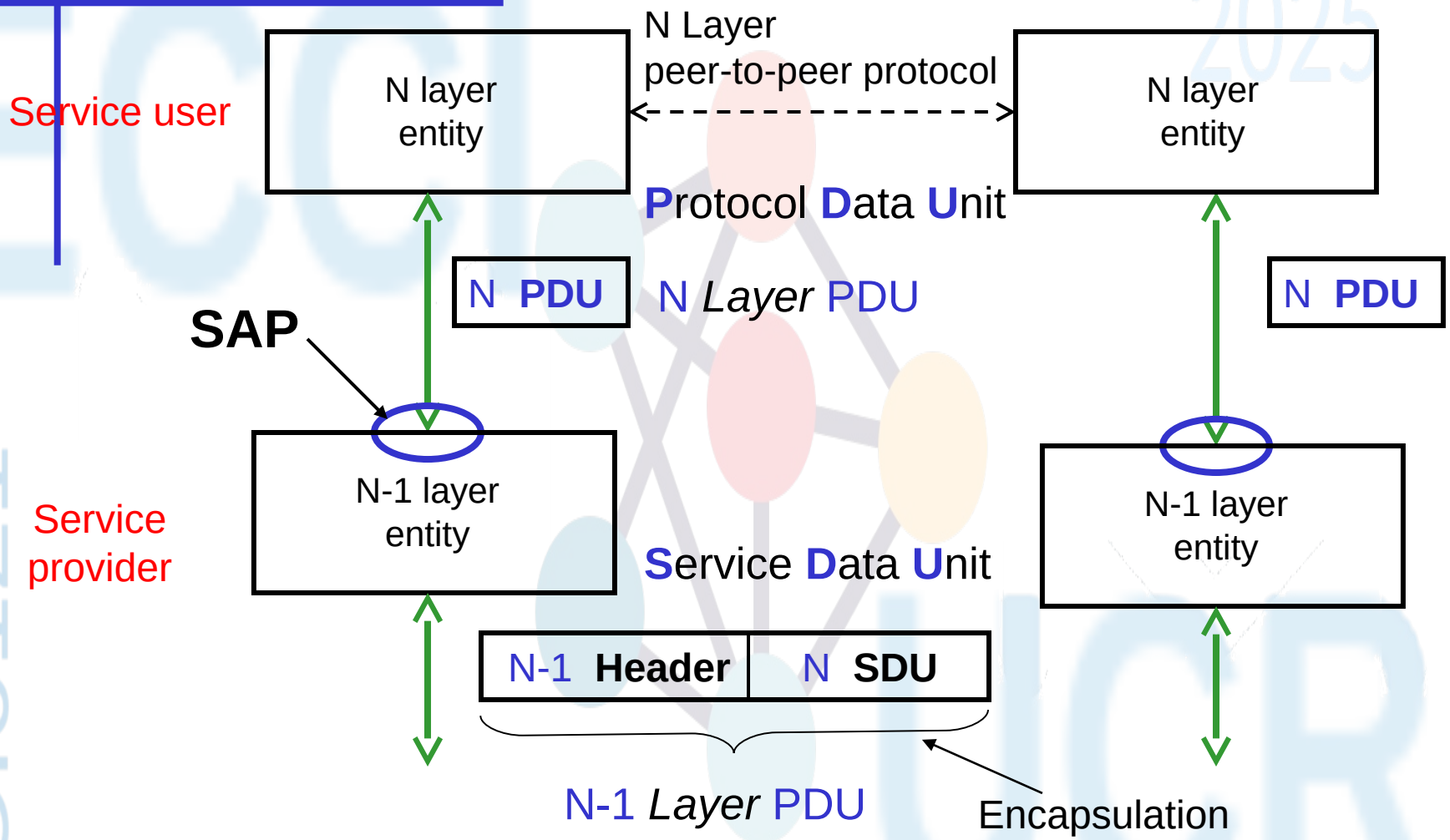
Answer

Service: fuel distribution, car-washing

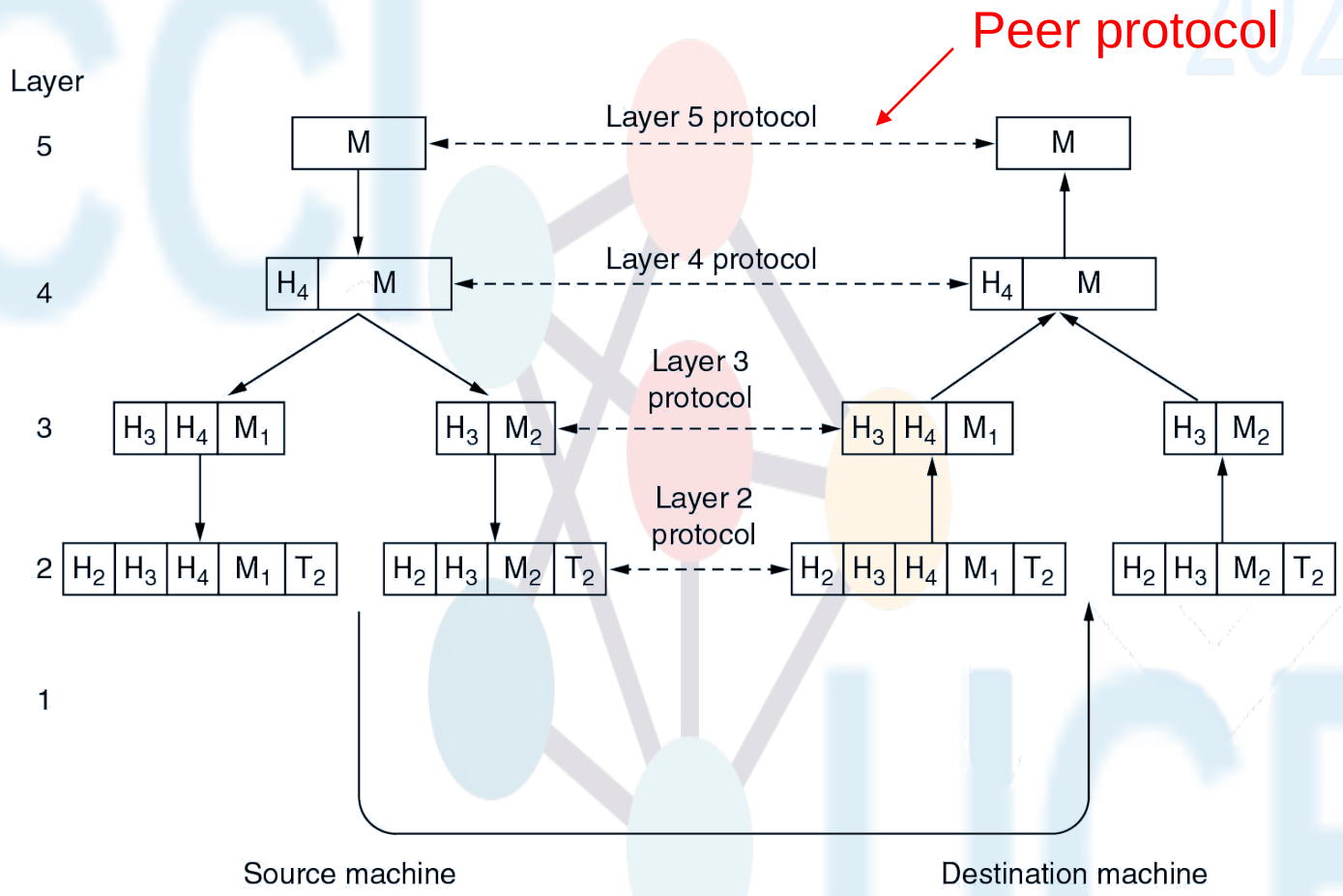
Service Access Point: Fuel machine, washing room

Quality of service: Vary

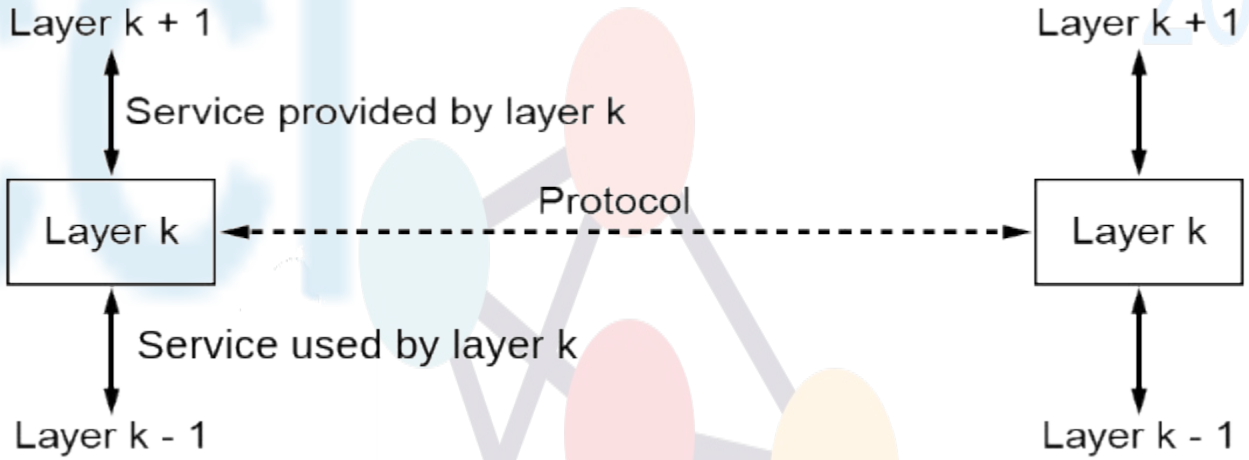
Data exchange



Protocol stacks



Compare service & protocol



Service: **what** a layer does

Interface between different layers: **how** to access service

Protocol: set of rules; packets between the same layers

- **how** the service is implemented (even more than this)

Services

- Telecom definition of services
  - The same as application
    - "IT product whose *value* resides mostly in functionality & content, rather then transport or connectivity."
      - » Q: Does this apply to most of the Internet applications today?
    - "Value added functionality provided by operators to network users"
    - Services are what customers use & pay for
- Web services
  - "A software system designed to support interoperable machine-to-machine interaction over a network." wikipedia
  - provide a very loose coupling between an application that uses the web service and the web service itself
- NGN services
  - "Software applications that provide a useful & well-defined functionality to the user."
    - Can be realized using interactive distributed software components

Telecom vs. Web services (applications)

- Web services:
  - Visual & presentation oriented
  - Users use multimedia machines to access those
  - Information is presented to user for consumption
    - Pushed from a Web server to a browser for display
  - Stress is on presentation
  - User expectations
    - Prepared to wait a few more seconds
    - No problem with downloading
  - Deployment
    - quick
  - Security
    - Less secure
  - Billing
    - Flat

- Telecom services
  - No pushing of content
  - Presentation is more aural
  - The main trust is auditory
  - Stress is on timing
  - No intermediary
  - User expectations
    - Downloading before making 911 call is unacceptable!
  - Improved privacy
  - Deployment
    - slow
  - Security
    - More secure
  - Billing
    - per service

Homework assignment

Define:

Layers,

Services,

Service providers,

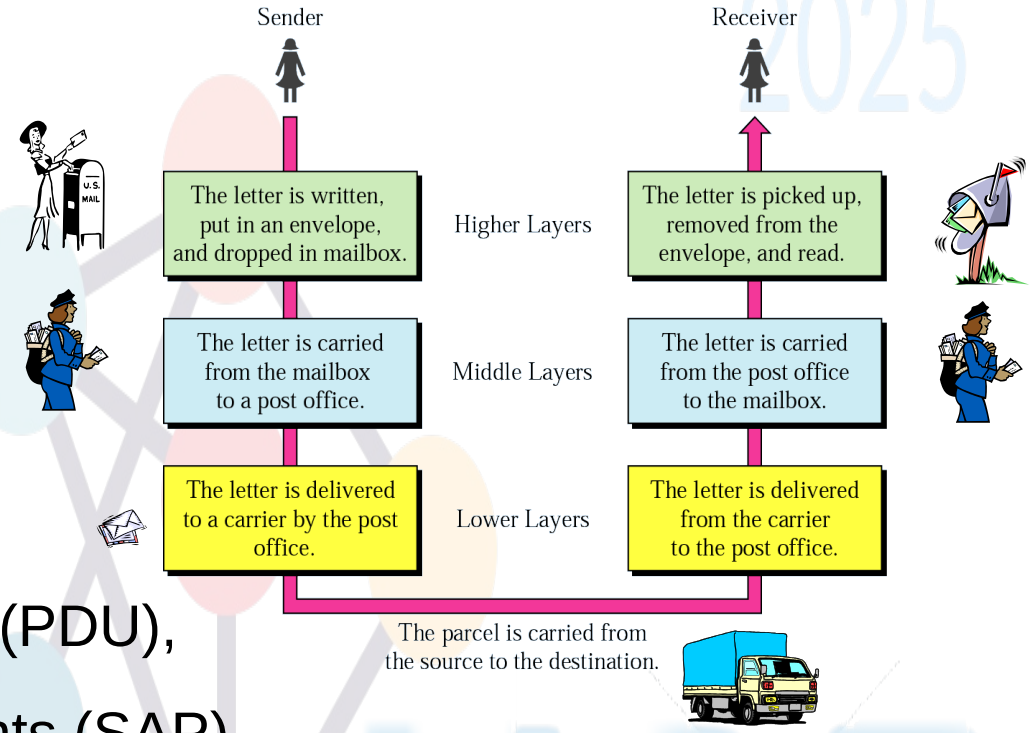
Service users,

Protocol Data Units (PDU),

Service Access Points (SAP),

Peer protocols,

in the example given at the beginning of the class



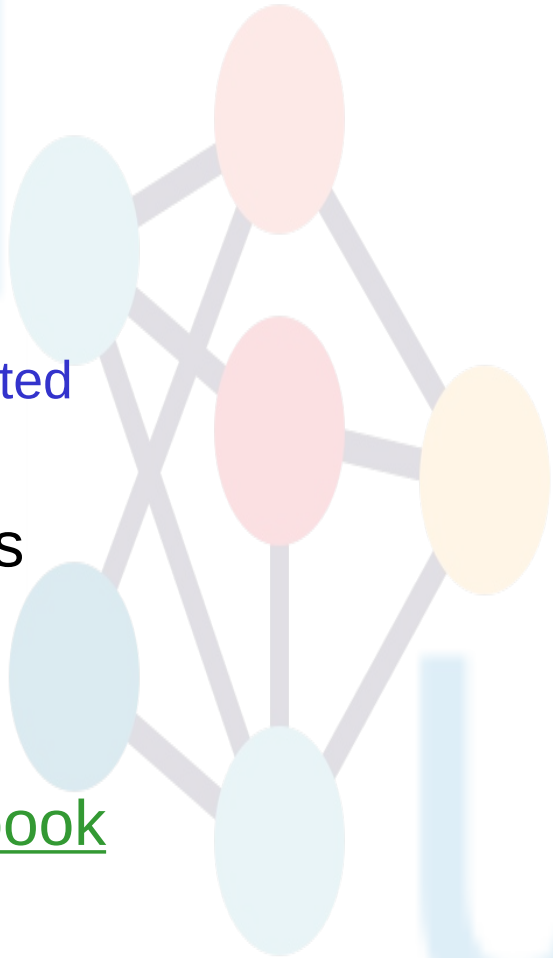
Homework assignment (cnt'd)

- Think for a moment about the restaurant. You go there for a dinner. You enter the restaurant, sit by a table trying to feel yourself comfortable. You need to communicate with the waiter. To get you served the waiter may communicate with someone else.
- Define:
  - layers
  - protocols
  - service provider(s)
  - service users
  - service(s)
  - service primitive(s)
  - access point(s)
  - Quality of Service

  if any.
- Redo the exercise in case you want to buy a coffee to a person sitting at another table because you would like to meet her/him.

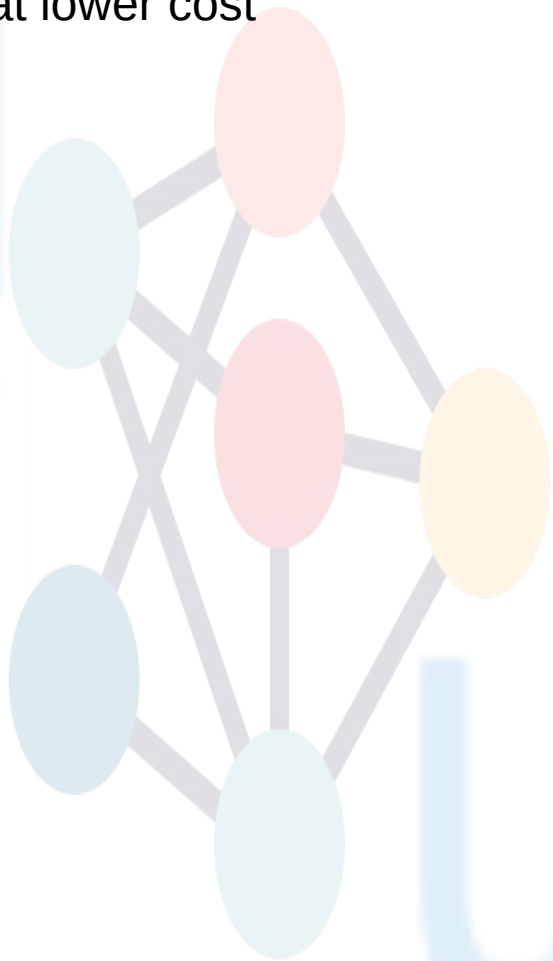
Content

- Layering
- Protocols
- Services
  - connectionless
  - connection oriented
  - synchronous
- Service primitives
- OSI layers
- TCP/IP
- Model from the book



Issues, to be resolved by the layers

- Larger bandwidth at lower cost
- Error correction
- Flow control
- Addressing
- Multiplexing
- Naming
- Congestion control
- Mobility
- Routing
- Fragmentation
- Security
- ....



Reference models

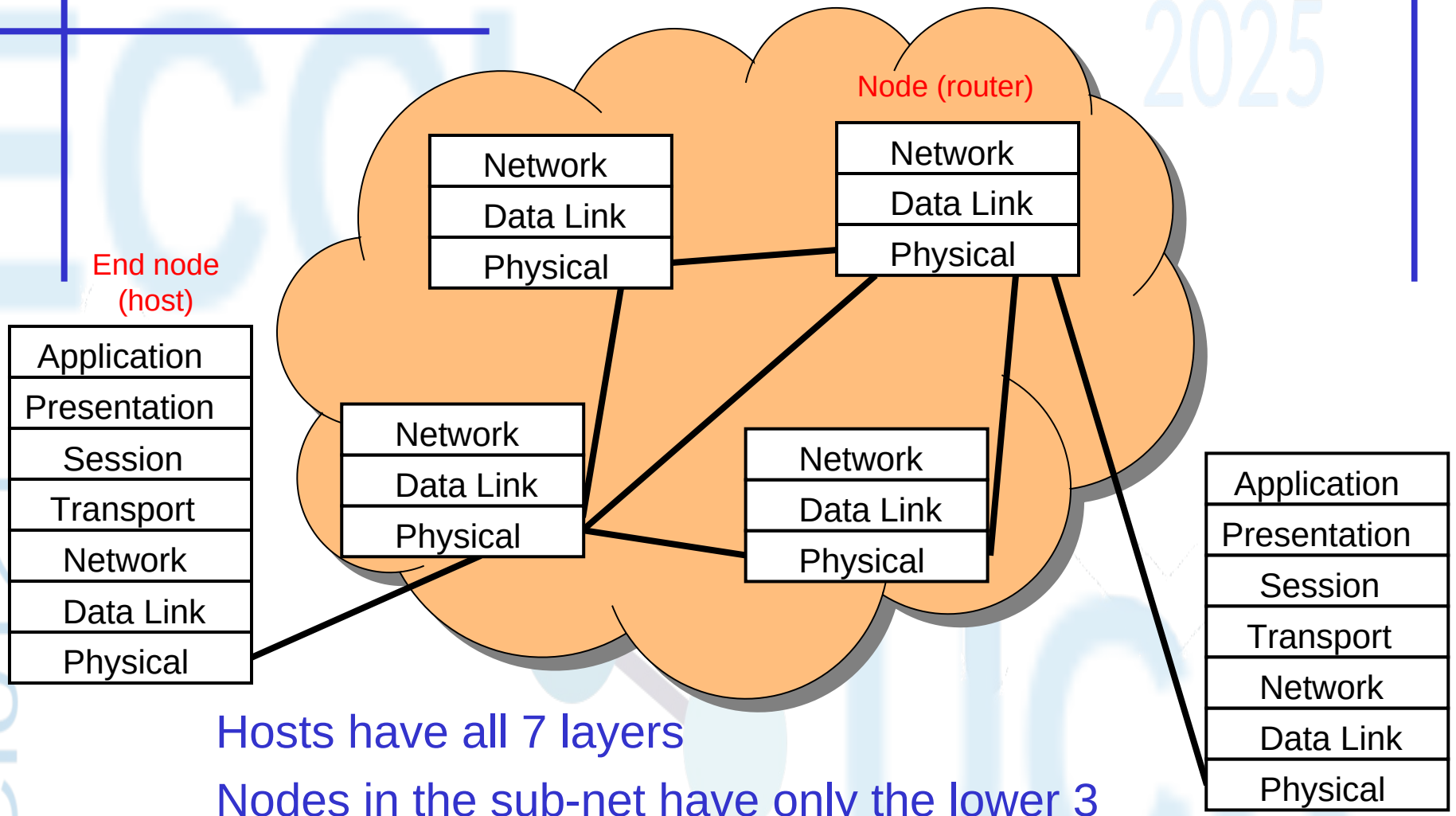
- OSI reference model
- TCP/IP

OSI Reference model

- Open System Interconnection
- 7 layers

1. Crate a layer when different abstraction is needed
2. Each layer performs a well define function
3. Functions of the layers chosen taking internationally standardized protocols
4. Minimize information flow across the interfaces
5. Number of layers – large enough to avoid complexity
6. Please Do Not Throw the Sausage Pizza Away

OSI Reference Model (cnt'd)



Hosts have all 7 layers

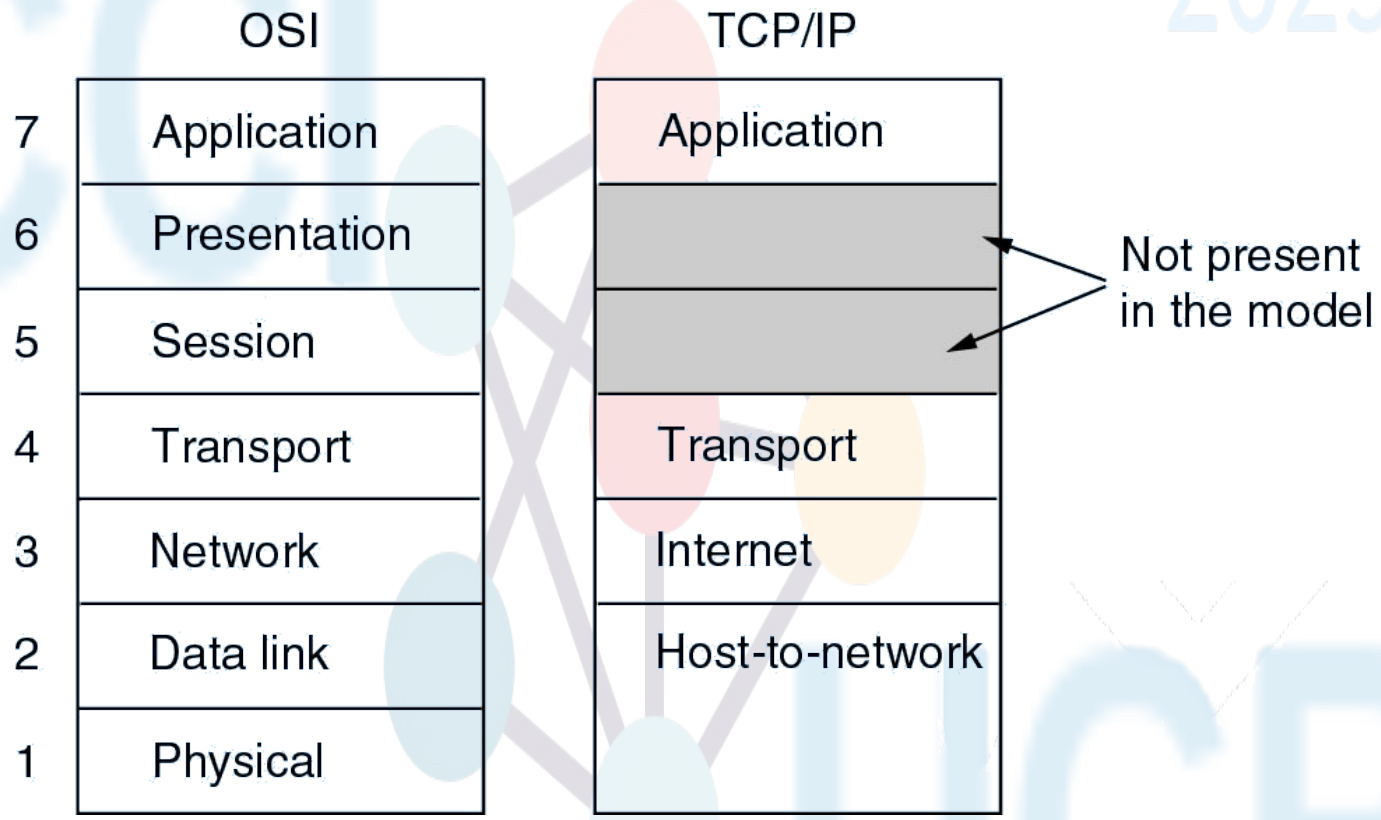
Nodes in the sub-net have only the lower 3

Some stacks and the OSI model

Protocol Stacks in Relationship to the OSI Model

OSI Layer	Apple Computer	Banyan Systems	DEC DECnet	IBM SNA	Microsoft Networking	Novell NetWare	TCP/IP Internet	Xerox XNS	OSI Protocols
Application Layer 7	Application Programs and Protocols for file transfer, electronic mail, etc.								
Presentation Layer 6	AppleTalk Filing Protocol (AFP)	Remote Procedural Calls (Net RPC)	Network Management Network Application	Transaction Services Presentation Services	Server Message Block (SMB)	NetWare Core Protocols (NCP	(Telnet, FTP, SMTP, etc.)	Control and Process Interaction	ISO 8823
Session Layer 5	AppleTalk Session Protocol (ASP)		Session	Data Flow Control	Network Basic Input/Output System (NetBIOS)	Network Basic Input/Output System (NetBIOS)			ISO 8327
Transport Layer 4	AppleTalk Transaction Protocol (ATP)	VINES InterProcess Communications (VIPC)	End Communications	Transmission Control	Network Basic Extended User Interface (NetBEUI)	Sequenced Packet Exchange (SPX)	Transmission Control Protocol (TCP), User Datagram Protocol (UDP)	Sequenced Packet Protocol (SPP)	ISO 8073 TP0-4
Network Layer 3	Datagram Delivery Protocol (DDP)	VINES Internet Protocol (VIP)	Routing	Path Control		Internet Packet Exchange (IPX)	Internet Protocol (IP)	Internet Datagram Protocol (IDP)	ISO 8473 (CLNP)
Data Link Layer 2	Network Interface Cards: Ethernet, Token-Ring, ARCNET, StarLAN, LocalTalk, FDDI, ATM, etc. NIC Drivers: Open Datalink Interface (ODI), Network Independent Interface Specification (NDIS)								
Physical Layer 1	Transmission Media: Twisted Pair, Coax, Fiber Optic, Wireless Media, etc.								

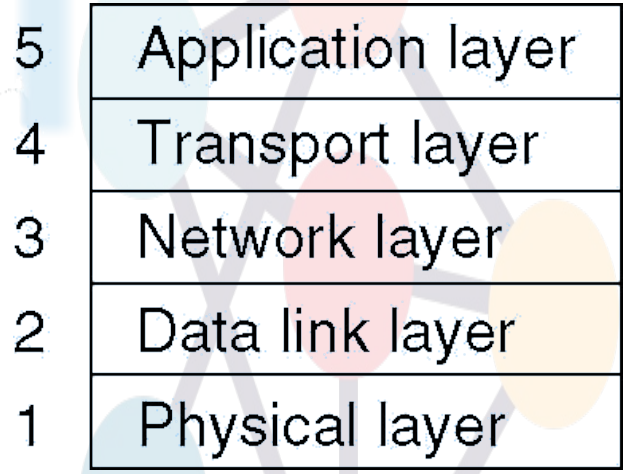
TCP/IP Model


OSI
TCP/IP
7
Application
6
Presentation
5
Session
4
Transport
3
Network
2
Data link
1
Physical
Application
Transport
Internet
Host-to-network
Not present
in the model

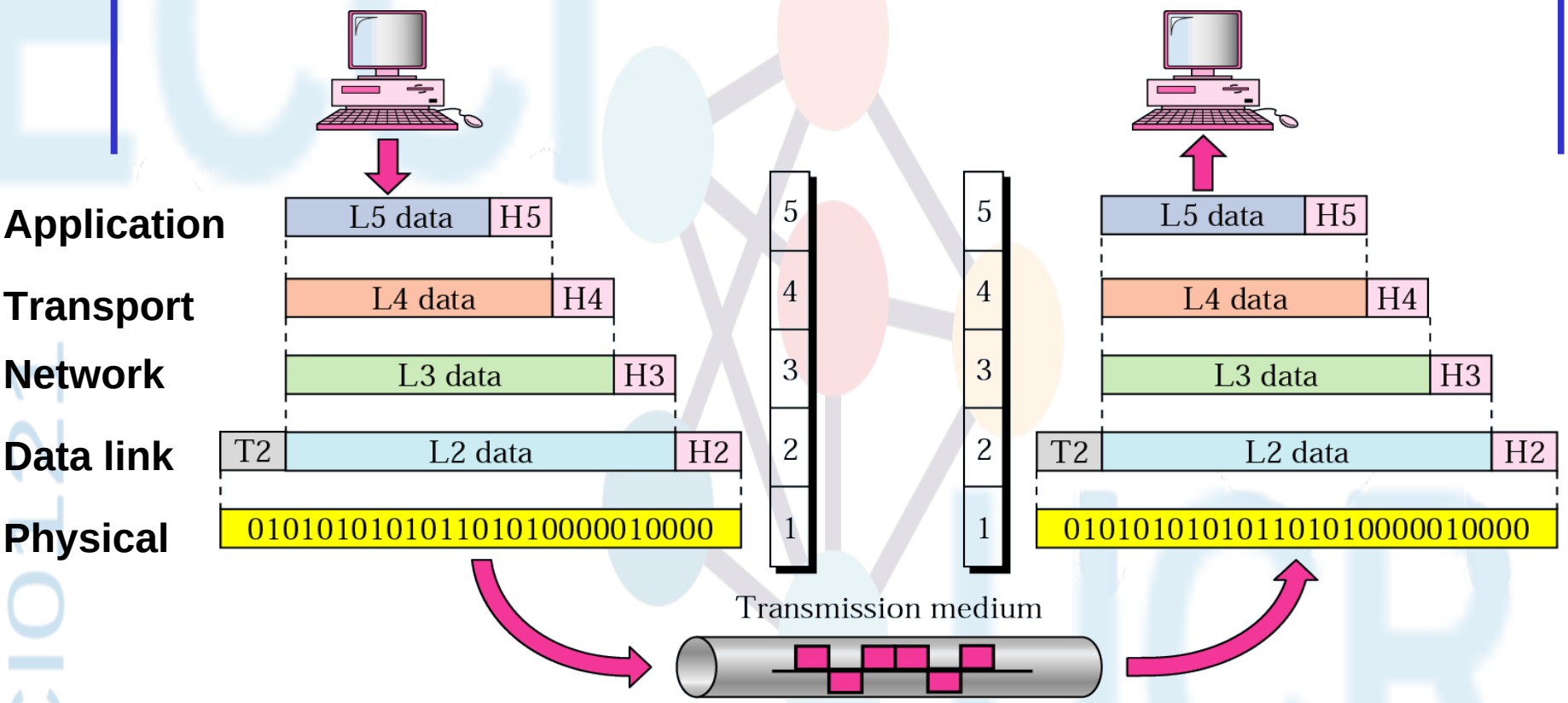
Comparing different models

- TCP/IP ‘model’
  - named after the dominating protocols
  - model constructed after the fact
  - host-to-network
    - not really a layer; just get IP packets across
- OSI model
  - well-discussed, before designing protocols
- OSI protocol stack
  - not widely used

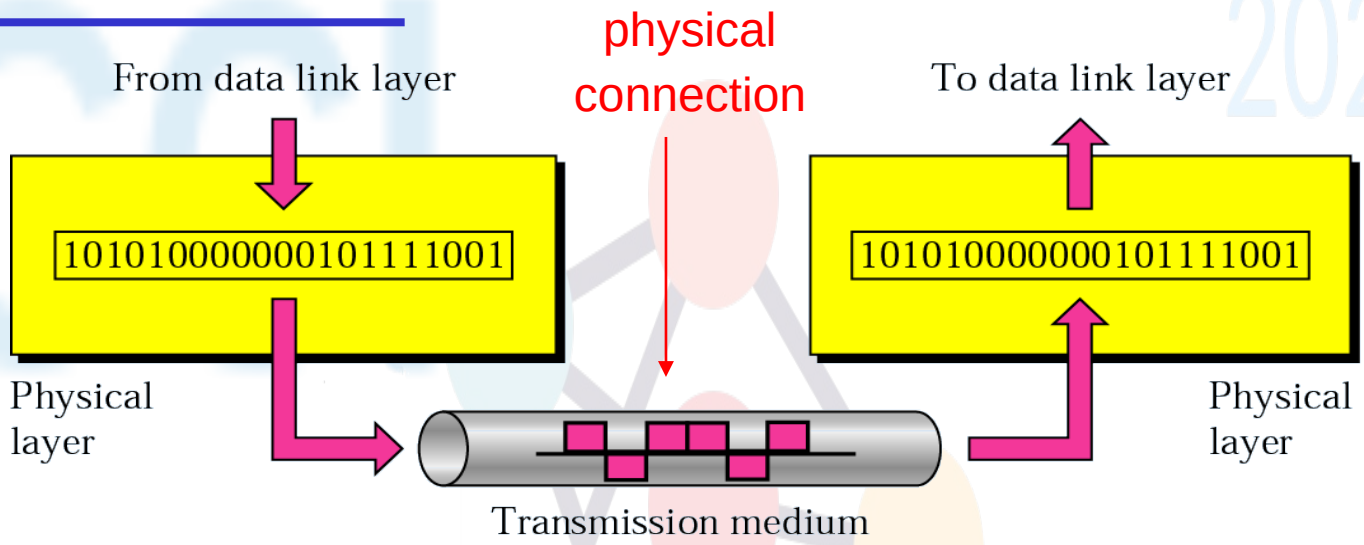
Model in the book...



Model in the book…(cnt'd)



Physical layer



Transporting bits from one end node to the next

- type of the transmission media (twisted-pair, coax, optical fiber, air)
- bit representation (voltage levels of logical values)
- data rate (speed)
- synchronization of bits (time synchronization)

Physical layer

QUESTION

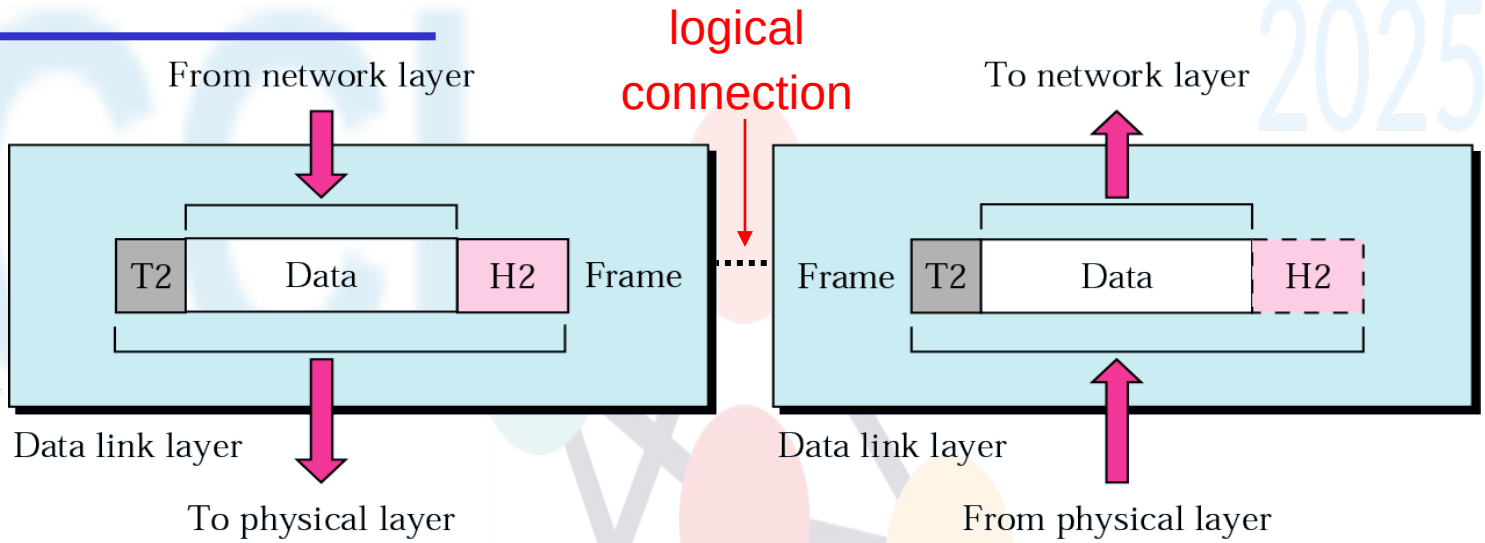
Define the *service* and the *protocol* in the Physical layer

ANSWER

Service: moving bits between two systems on the same link

Protocol: coding scheme, duration of bit, voltage levels, etc.

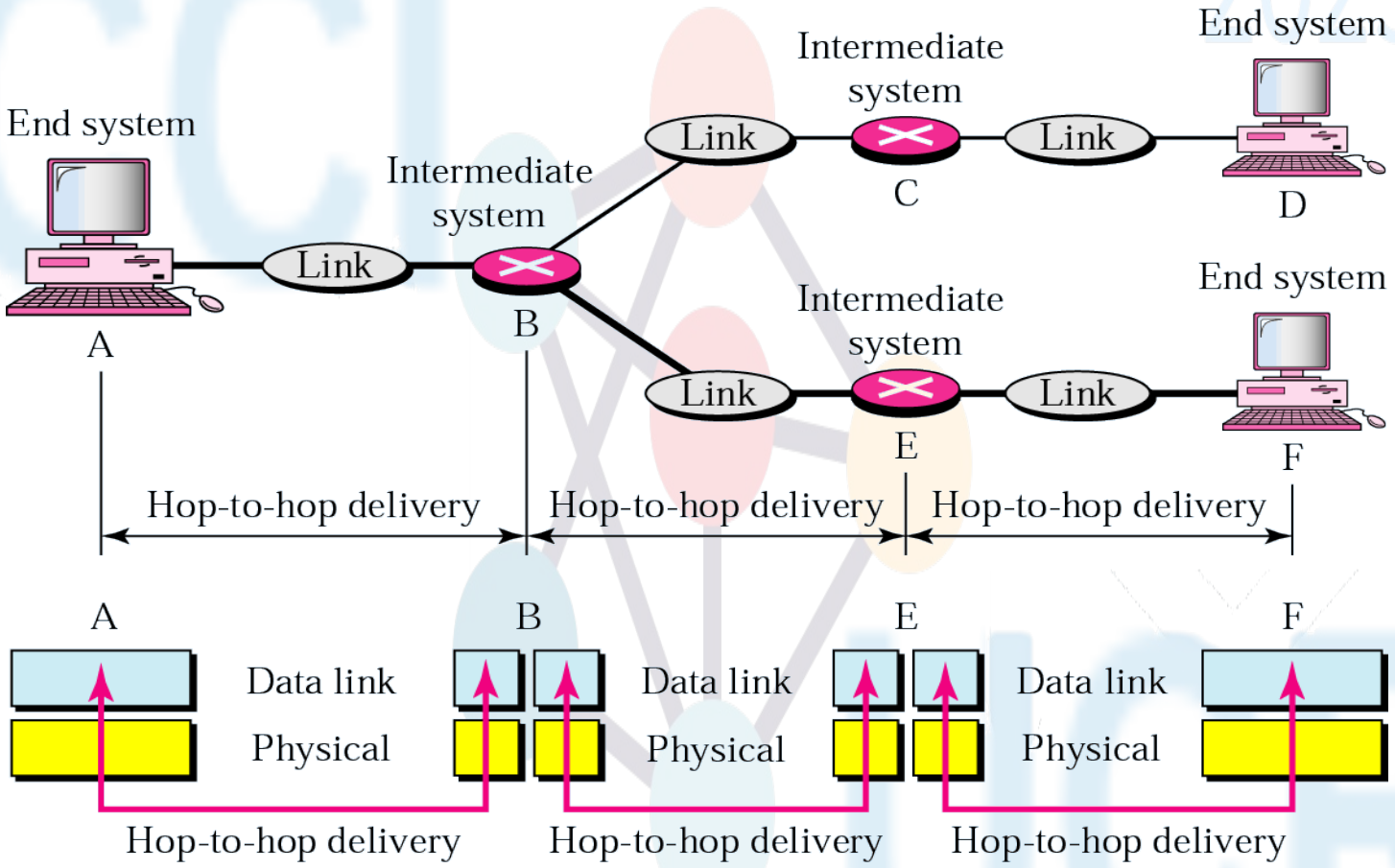
Data Link layer



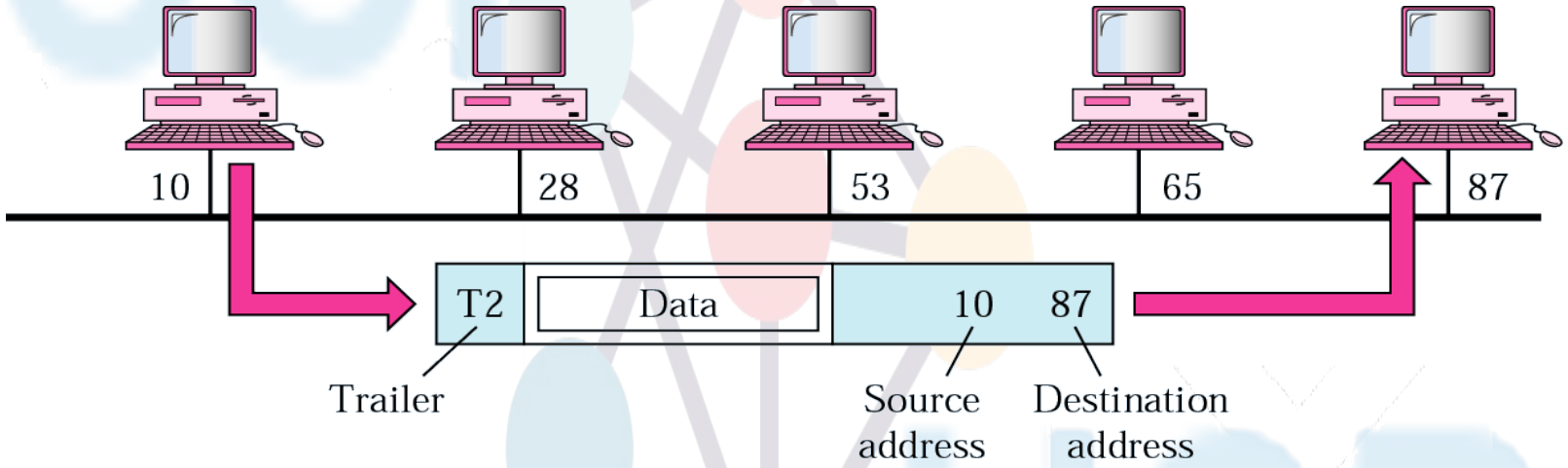
Transporting frames from one end node to the next one

- framing
- physical addressing
- flow control
- error control
- access control (broadcast networks)

Data Link layer
- hop-to-hop delivery-



Data Link layer
- example-



Data Link layer

QUESTION

Define the *service* and *protocol* in the Data Link layer

ANSWER

Service: attach frame separator; send data between peers

arbitrates the access on the common media, flow control

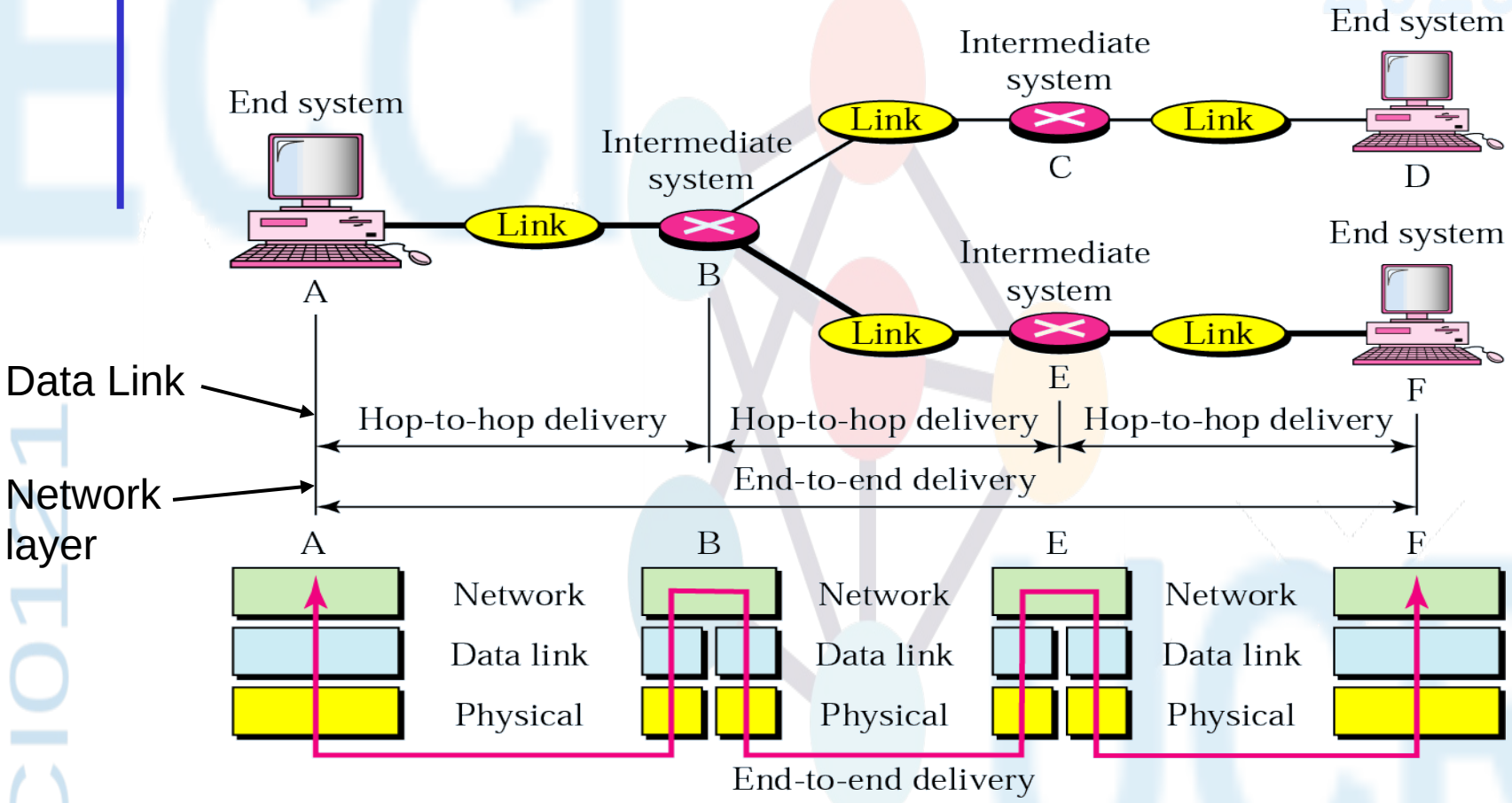
Protocol: MACA, CSMA/CD, Bluetooth…

Network layer

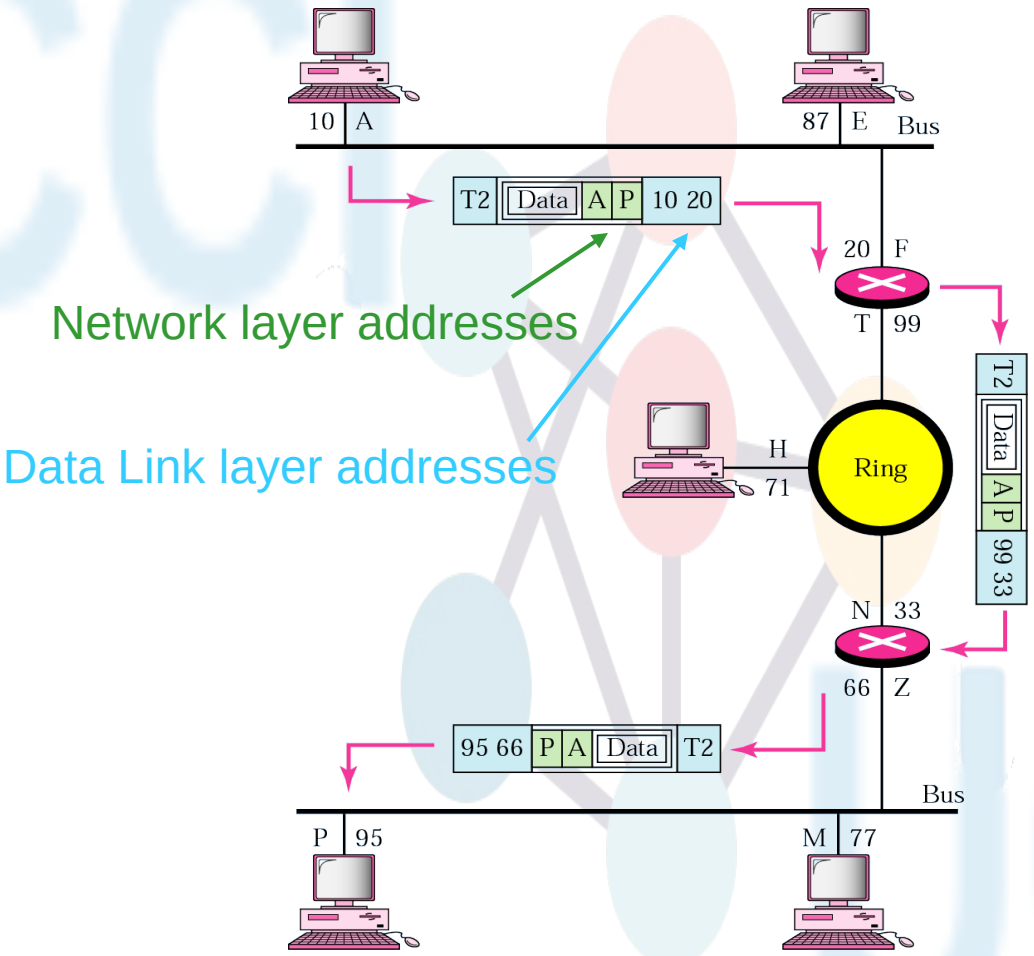
Not a message

- End-to-End **packet** delivery
  - From the original source to a destination
- Needed when 2 devices are attached to different networks
  - What is the network definition here?
- Main duties:
  1. Logical addressing
  2. Routing
  3. Switching
  4. Congestion control and QoS

Source to destination delivery



Network layer
- example -



Network layer

QUESTION

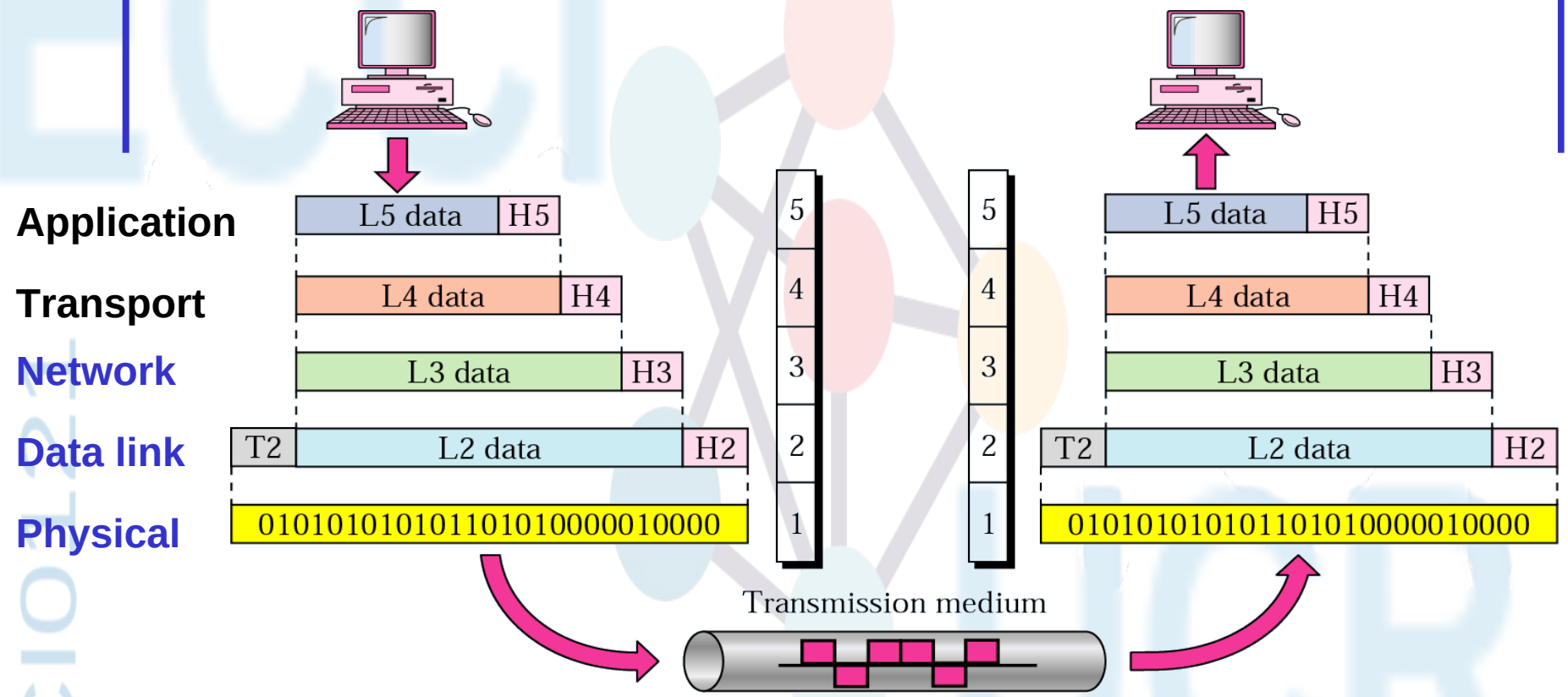
Define the *service* and the *protocol* in the Network layer

ANSWER

Service: packet delivery to the destination; fragmentation; reassembly (what)

Protocol: global addressing; construct routing tables; packet reassembly/fragmentation (how)

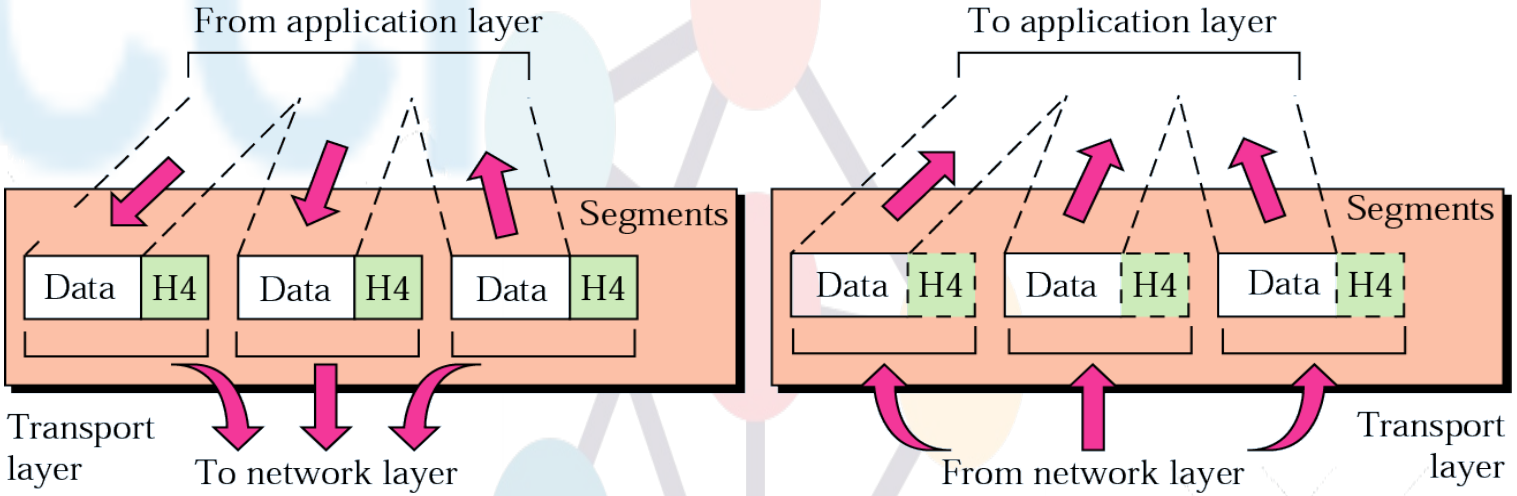
A reminder


Application
Transport
Network
Data link
Physical
L5 data
H5
L4 data
H4
L3 data
H3
T2
L2 data
H2
01010101010110101000010000
5
4
3
2
1
Transmission medium

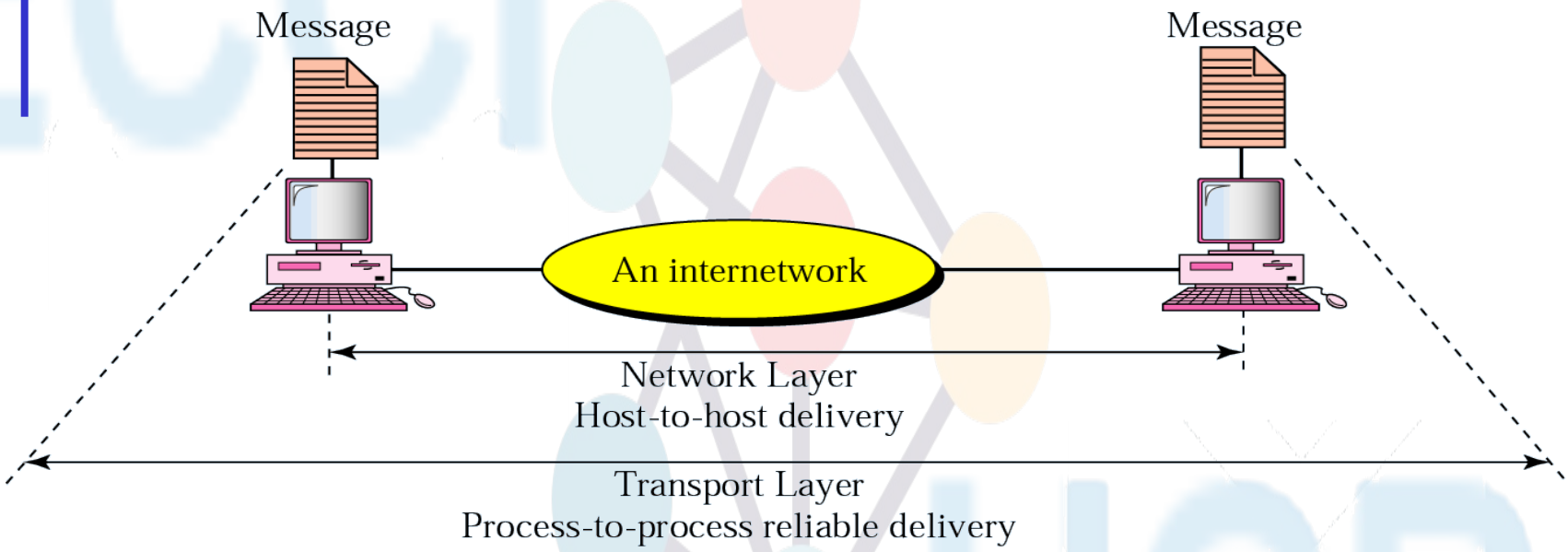
Transport layer

- Process-to-Process delivery of the entire **message**
  - From the original source to a destination
- Needed when several processes (running programs) active at the same time
- Main tasks:
  - Port addressing
  - Segmentation and reassembly
  - Congestion control
  - Flow control
  - Error control

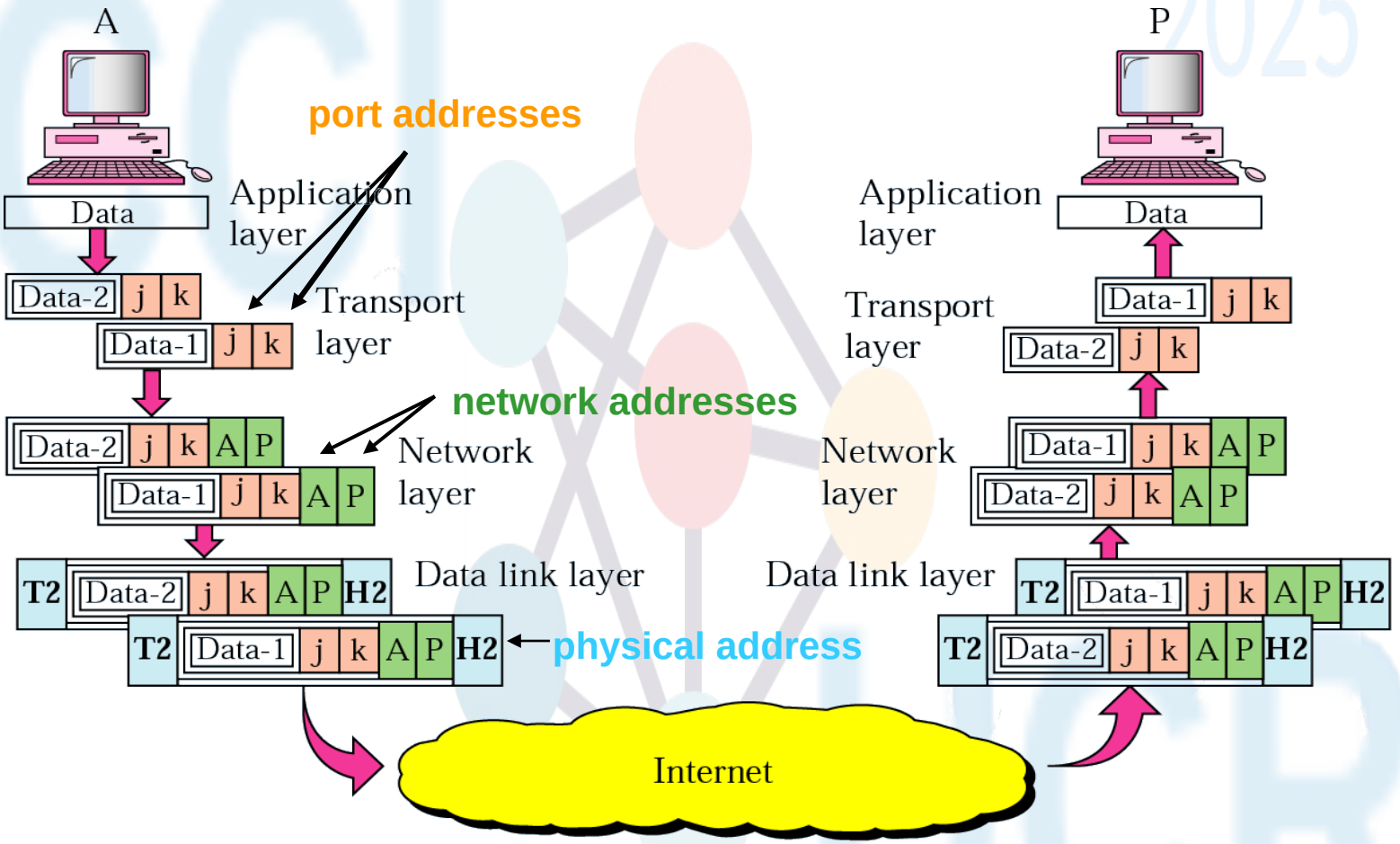
Transport layer



Transport layer
-an example of a reliable delivery -



Transport layer
-example 2 -



Transport layer

QUESTION

Define the *service* and *protocol* in the Transport layer

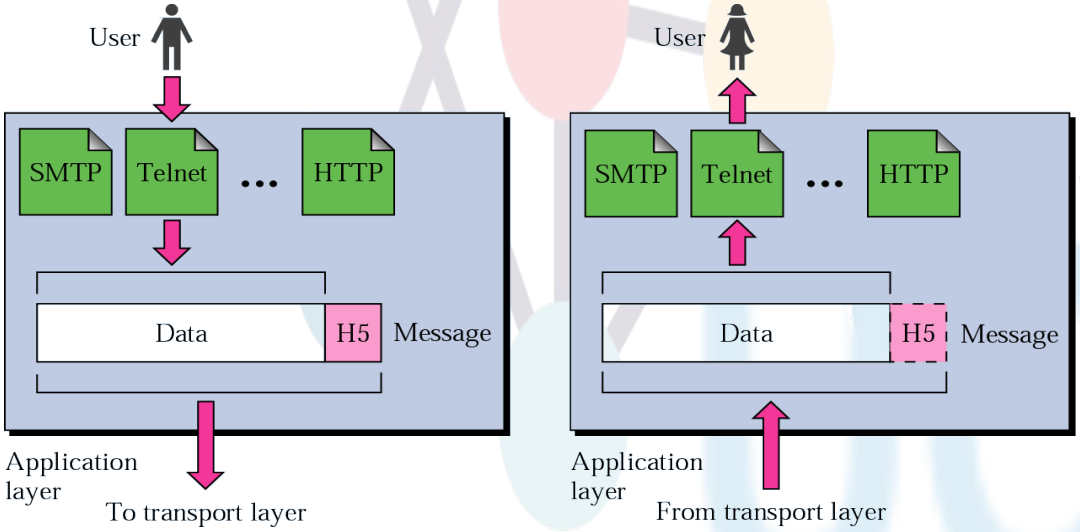
ANSWER

Service: in-order, error free, flow & congestion controlled end-to-end connection

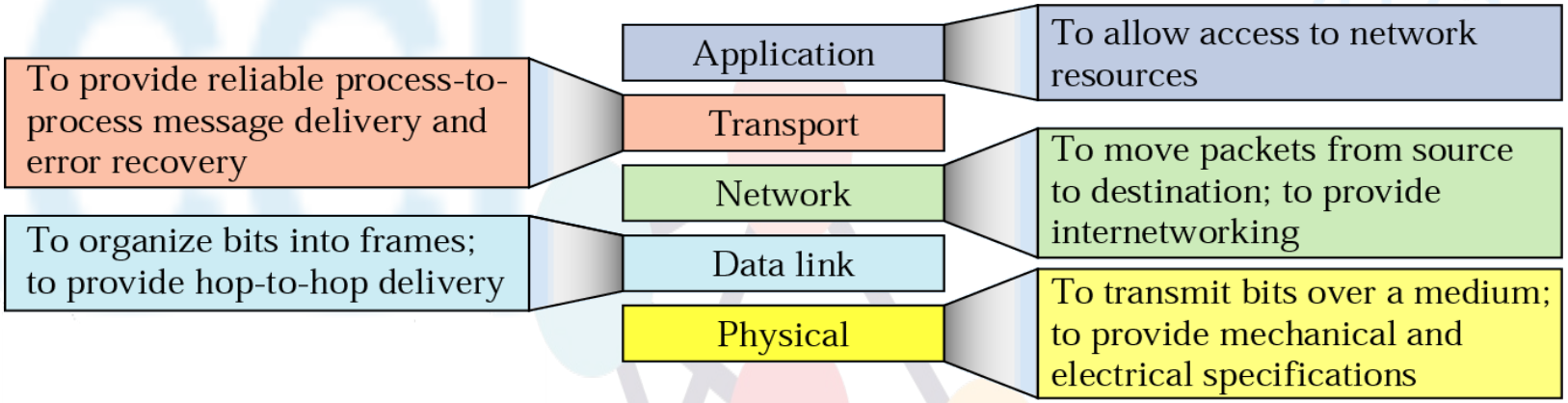
Protocol: implements reliability and flow & congestion control

Application layer

- Enables user to access the network
- Provides services to a user
  - E-mail
  - Remote file access and transfer (Telnet, FTP)
  - Access to WWW (HTTP)



Summary of layers and protocols

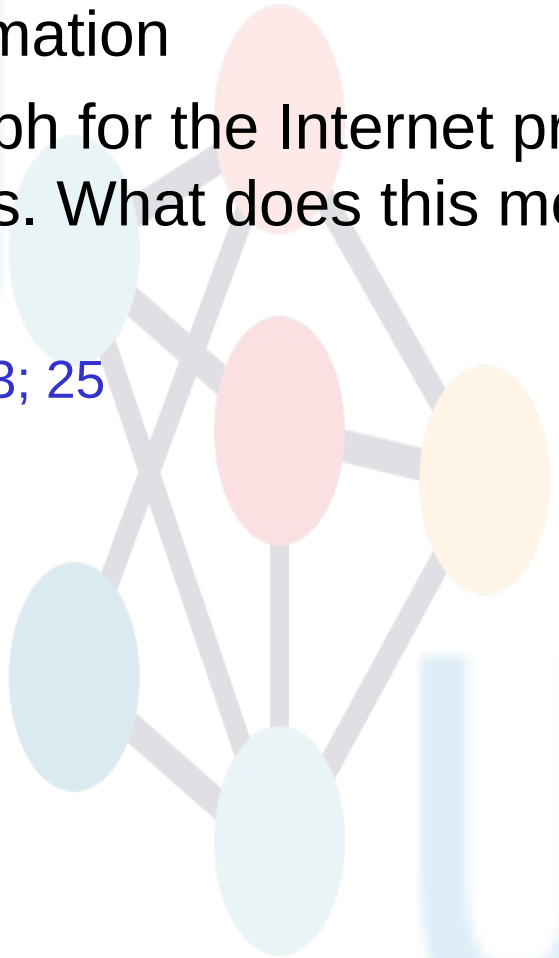


- Low-level protocols define the electrical and physical standards to be observed, bit- and byte-ordering and the transmission and error detection and correction of the bit stream

- High-level protocols deal with the data formatting, including the syntax of messages, the terminal to computer dialogue, character sets, sequencing of messages

Homework assignment

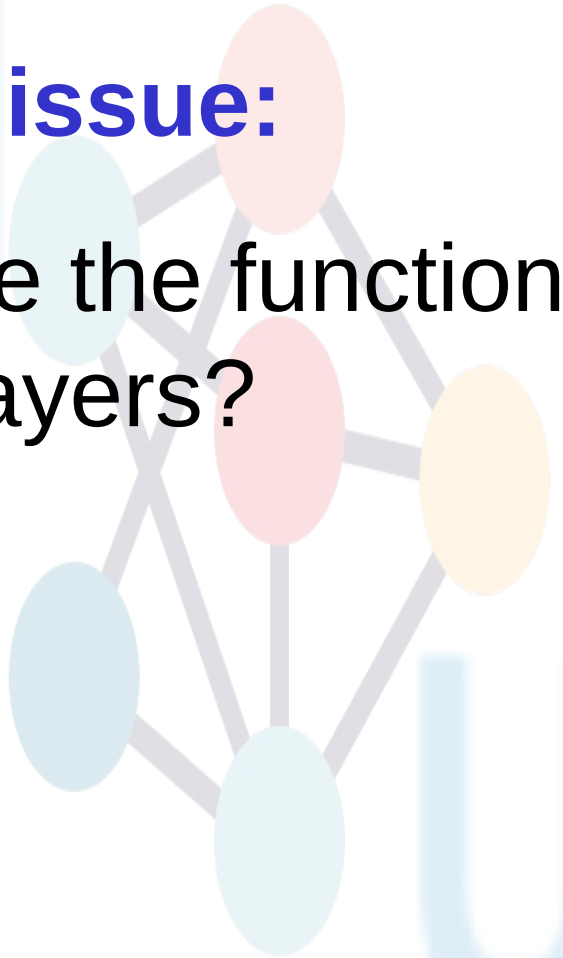
- Read extra information
- The protocol graph for the Internet protocols looks like an hour glass. What does this mean in practice?
- Chapter 1:
  - Exercises: 12; 13; 25



Extra information

Key design issue:

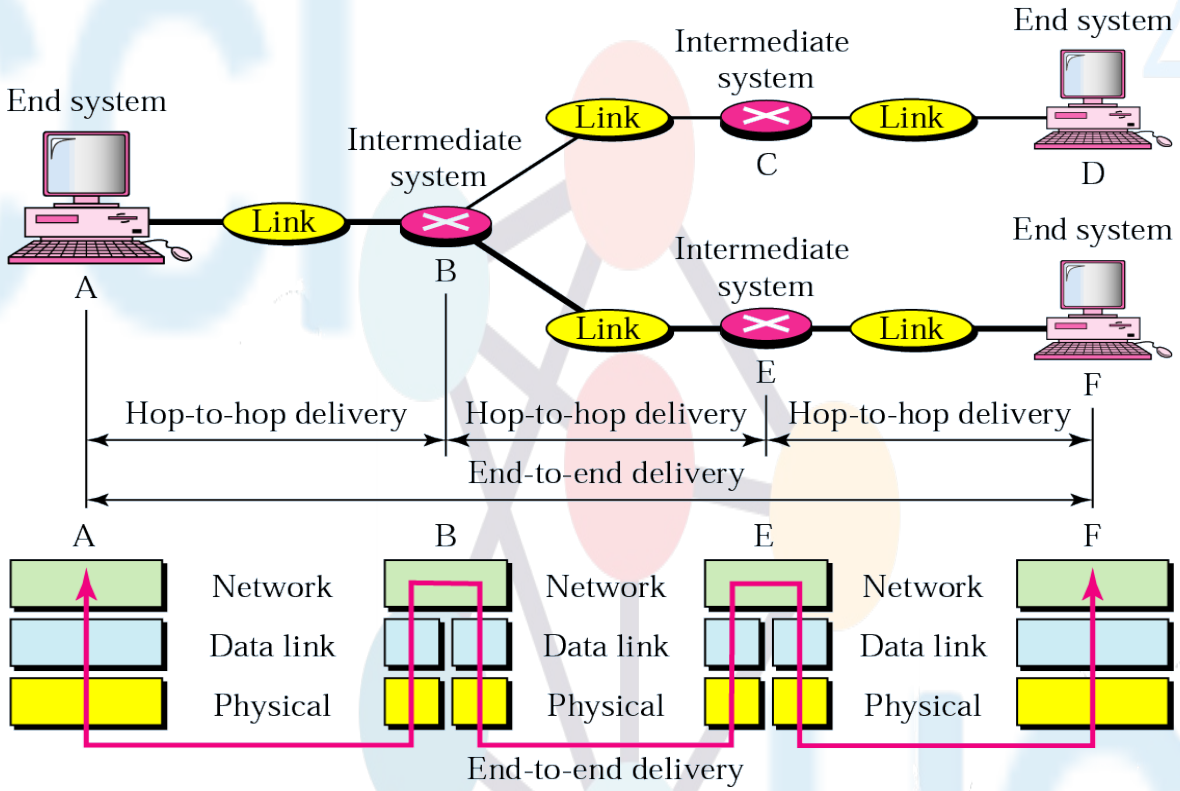
How to divide the functionality among the layers?



End-to-End argument

- The application knows the requirements best, place functionality as high in the layer as possible
- If the application can implement a functionality correctly, implement it at a lower layer only as a performance enhancement
- Think twice before implementing a functionality that you believe that is useful to an application at a lower layer

Example: Reliability

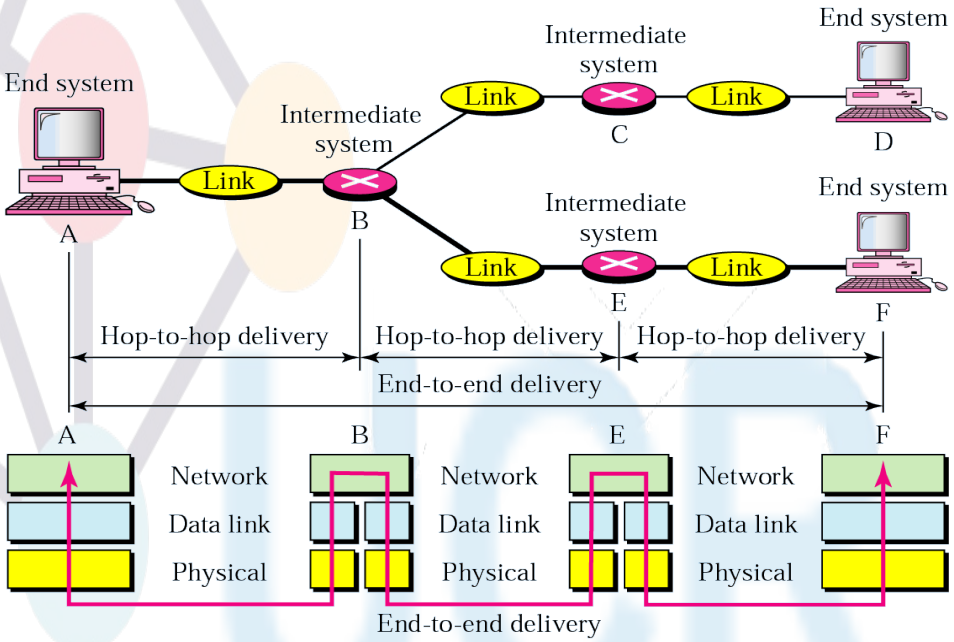


Solution 1: Make hop-to-hop delivery reliable and concatenate them

Solution 2: End-to-End control and re-transmission

Example: Reliability (cnt'd)

- The receiver has to do the check anyway!
- Thus, full functionality can be entirely implemented at the upper layer; no need for reliability from lower layers
  - Is this always correct?



Question

Is there any need to implement reliability at lower layers?

ANSWER

Yes, but only to improve performance

Example: in a high error-rate communication network a reliable communication service at lower layers might help

Trade-off layering

- Higher layers, closer to application, organize lower-level network resources to achieve application-specific design goals efficiently
- Lower layers, which support many independent applications, should provide effective sharing of resources
- Preserving low-cost options to innovate outside the network, while keeping the core network services and functions *simple and cheap*
- If the application can do it, don't do it at a lower layer -- the application knows the best what it needs

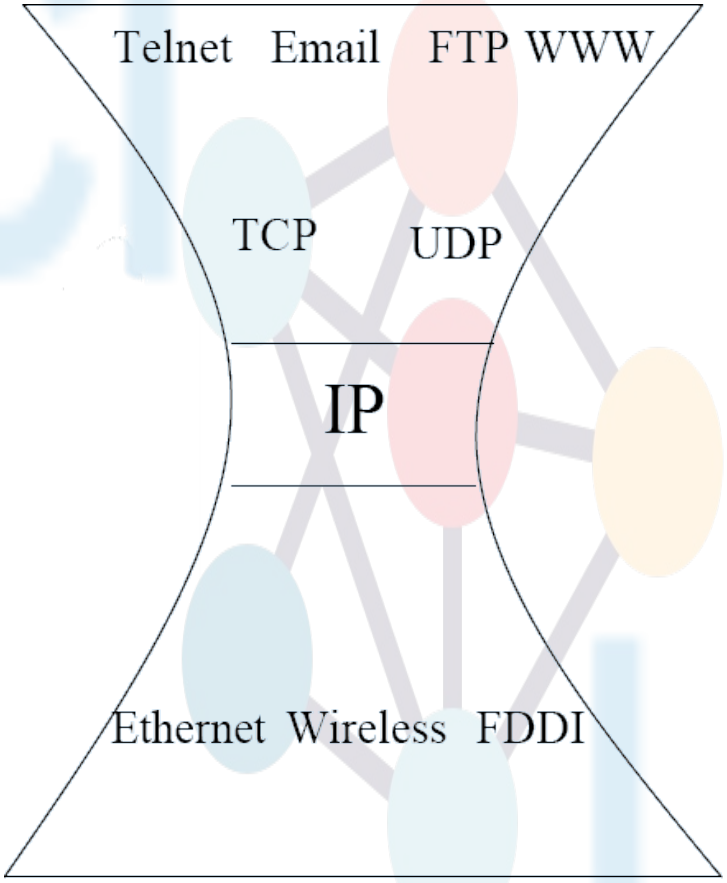
- Add functionality in lower layers if it
  - is used and improves performance of a large number of (current and potential future) applications, and
  - does not hurt (too much) other applications

Example: Error control

Question:

- E2E or localized control
- Localized error control
  - May increase propagation delay: store-and-forwarding done multiple times
  - Confines packet propagation
  - Reduce propagation of erroneous packet
  - Reduce propagation of retransmission
  - Reducing delay (rapid detection/recovery) and network load
  - Require functionality within network: buffering, error checking, time-outs for ACKs

The hourglass architecture of the Internet



The Internet model

- Component physical networks cooperate to form a virtual network called an internet

- All component networks are equal (Uniformity)
- Component networks can be added to existing internets and smaller internets can be combined to form bigger internets (Scalability)
- Any 2 hosts on the internet can communicate with each other (Universal connection)
- Interconnection is performed at the network level instead of the application level (Interconnection abstraction)

- Two networks are interconnected via a device or a computer called an *internet gateway* or an *internet router* (Network interconnection)