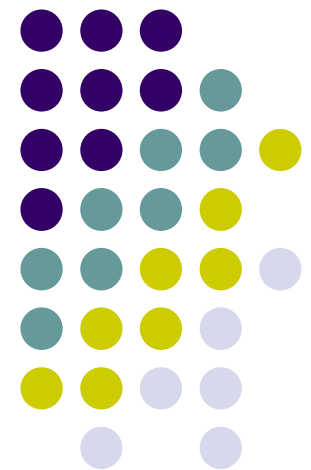


# Lecture 2:

## Basic concepts of computer communication models

---





# Last lecture

- Introduction of the course
- History of the Internet
- Concept of Computer Networks
- Some fundamental concepts: switching, connection oriented, connection less.

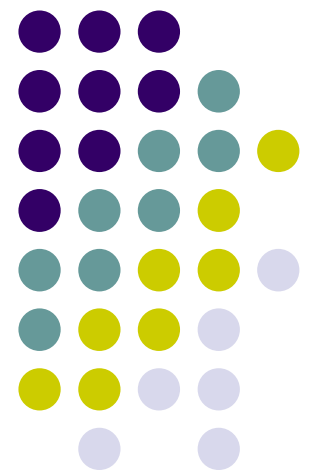


# Content of this lecture

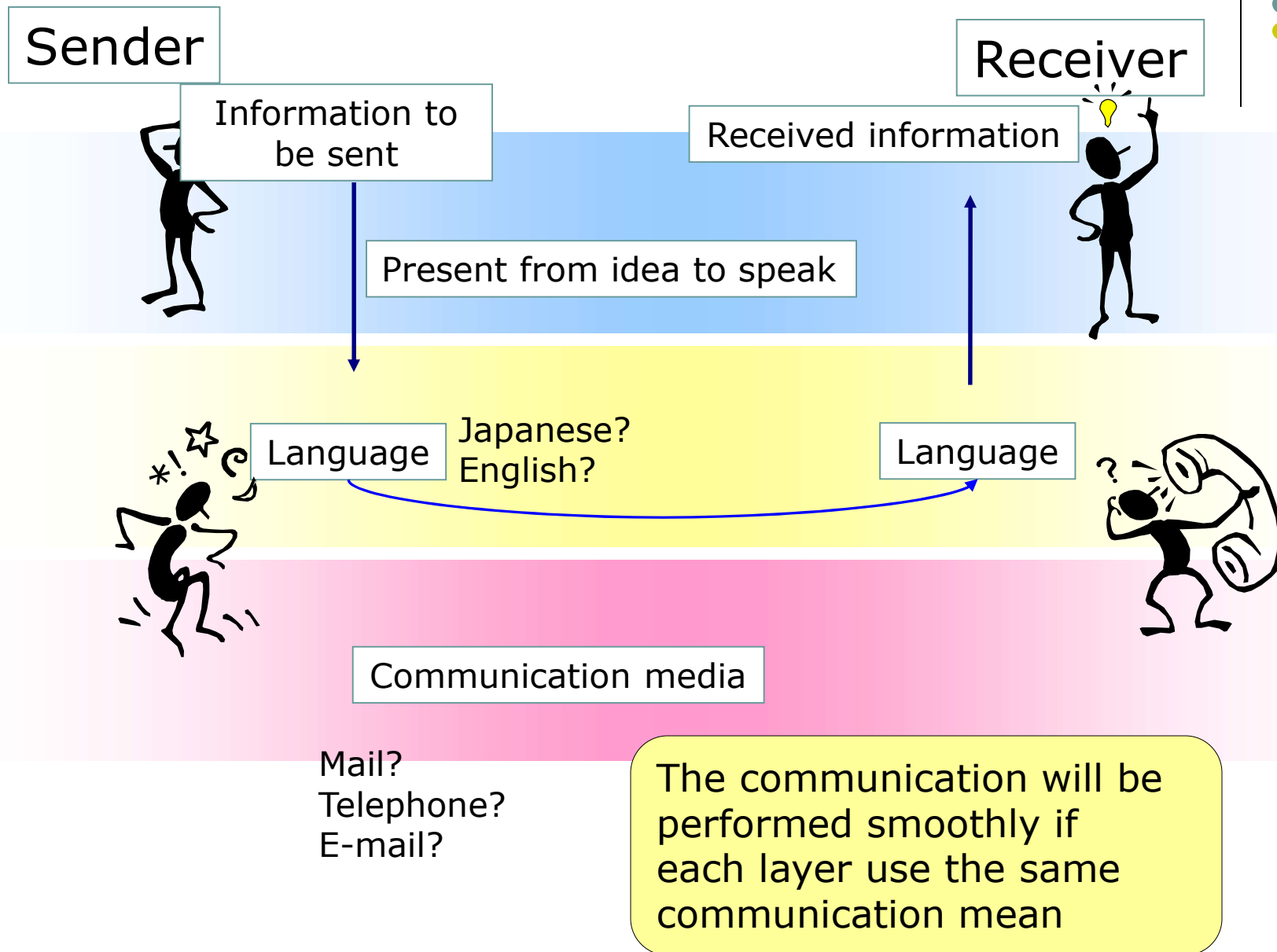
- Layer architecture
- OSI & TCP/IP reference model
- Addressing
- Domain name and conversion/resolution of domain name

# Layer architecture

---



# Example of different functionalities in human communication





## Example of layering (1)

Layered



AV Component

Player  
Speaker  
Amplifier

Not layered



Cassette

All functions are included in  
one cassette.  
Replace all when changing



## Example of layering (2)



ticket (purchase)	<i>ticketing service</i>	ticket (complain)
baggage (check)	<i>baggage service</i>	baggage (claim)
gates (load)	<i>gate service</i>	gates (unload)
runway takeoff	<i>runway service</i>	runway landing
airplane routing	<i>routing service</i>	airplane routing

*layers:* each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

# Why layering?



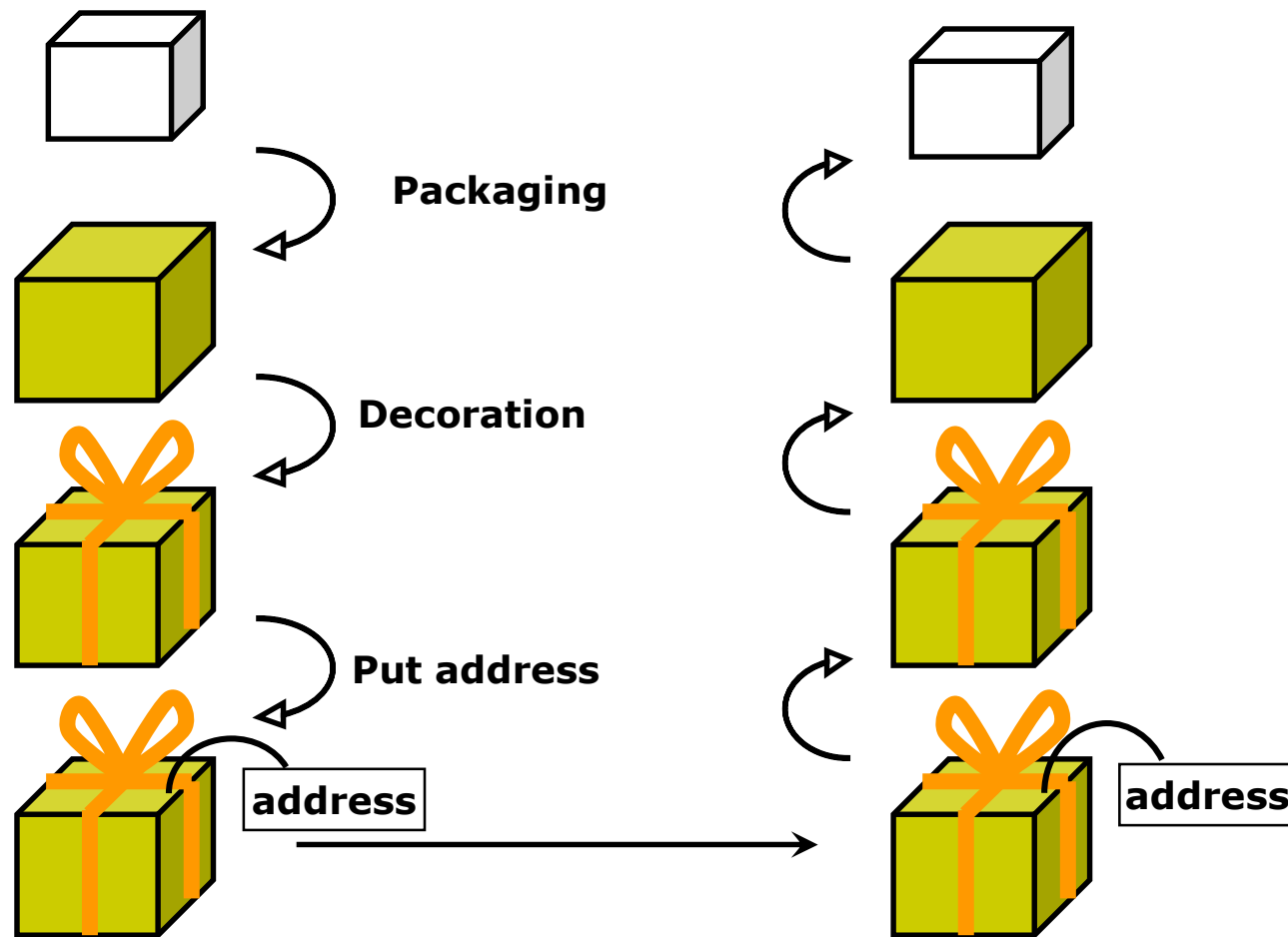
- Dealing with complex systems: *Divide and conquer*
- Explicit structure allows identification, relationship of complex system's pieces
- Modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in DVD player does not impact sound speakers.



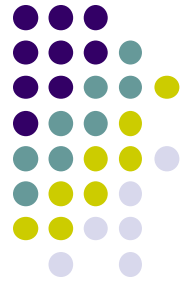
# Data Encapsulation



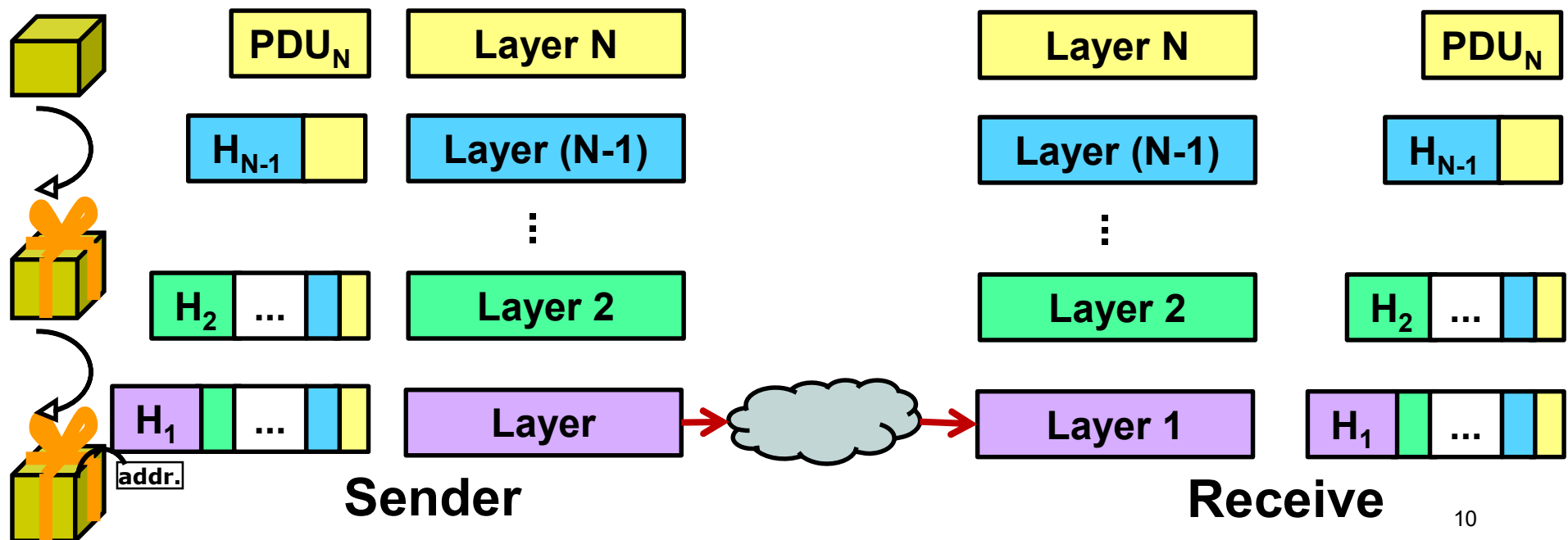
Data encapsulation is similar to a packaging process for a gift.



# Data transmission in layering systems



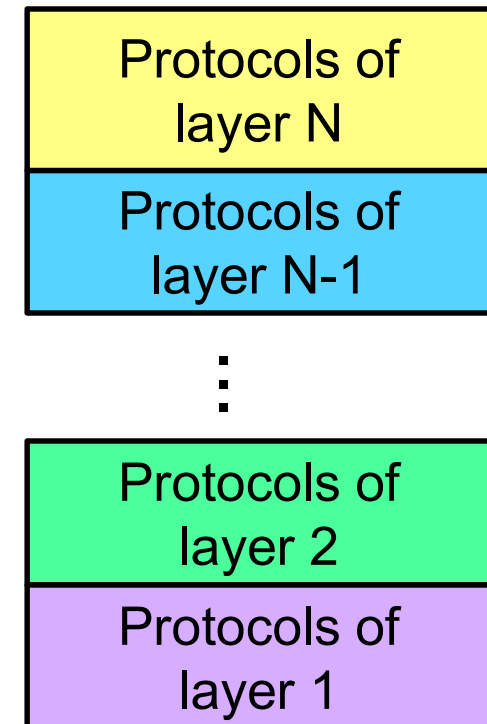
- Sender: add the header containing information for processing data at that layer and forward to the below layer (Đóng gói dữ liệu – Encapsulation)
- Receiver: process data by reading the right header, remove the header and forward to the upper layer





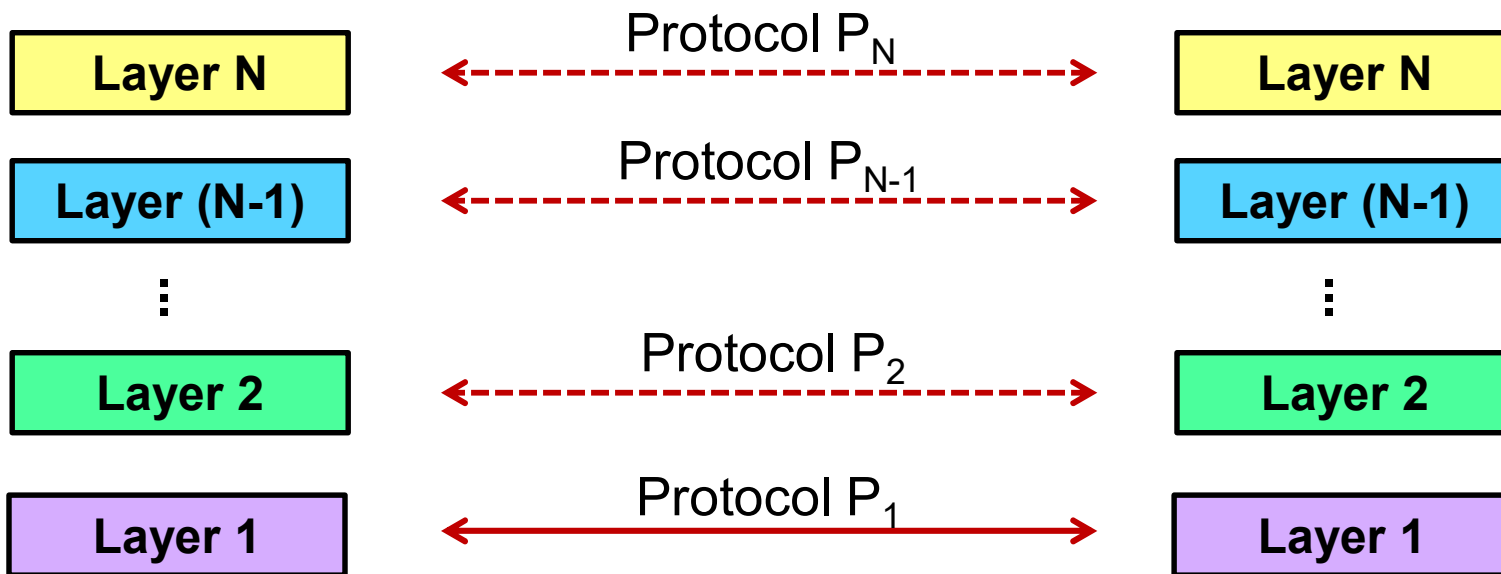
# Protocol stack

- Functions are splitted to layers
- Each layer has multiple ways to implement its functions → multiple protocols
- Protocol stack: stack protocols based on layering systems
- Protocols at each layer include:
  - Call a service of the below layer
  - Provide service(s) for the upper layer





# Data transmission in layering systems



- Layers (same level) on each side use the same protocol to exchange logical information
  - Two way to communicate logical information at the same logical level: connectionless or connected-oriented

# Connection oriented transmission vs. connectionless

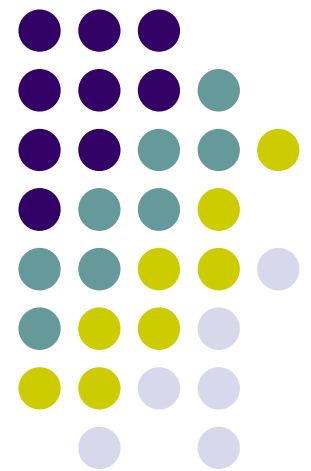


- Connection oriented transmission:
  - Data are transmitted over a connection already established
  - 3 working phases: Establishing a connection, data transmission, teardown the connection.
  - Reliable
- Connectionless transmission
  - No connection establishing phase
  - Only data transmission phase
  - Not reliable - “Best effort”

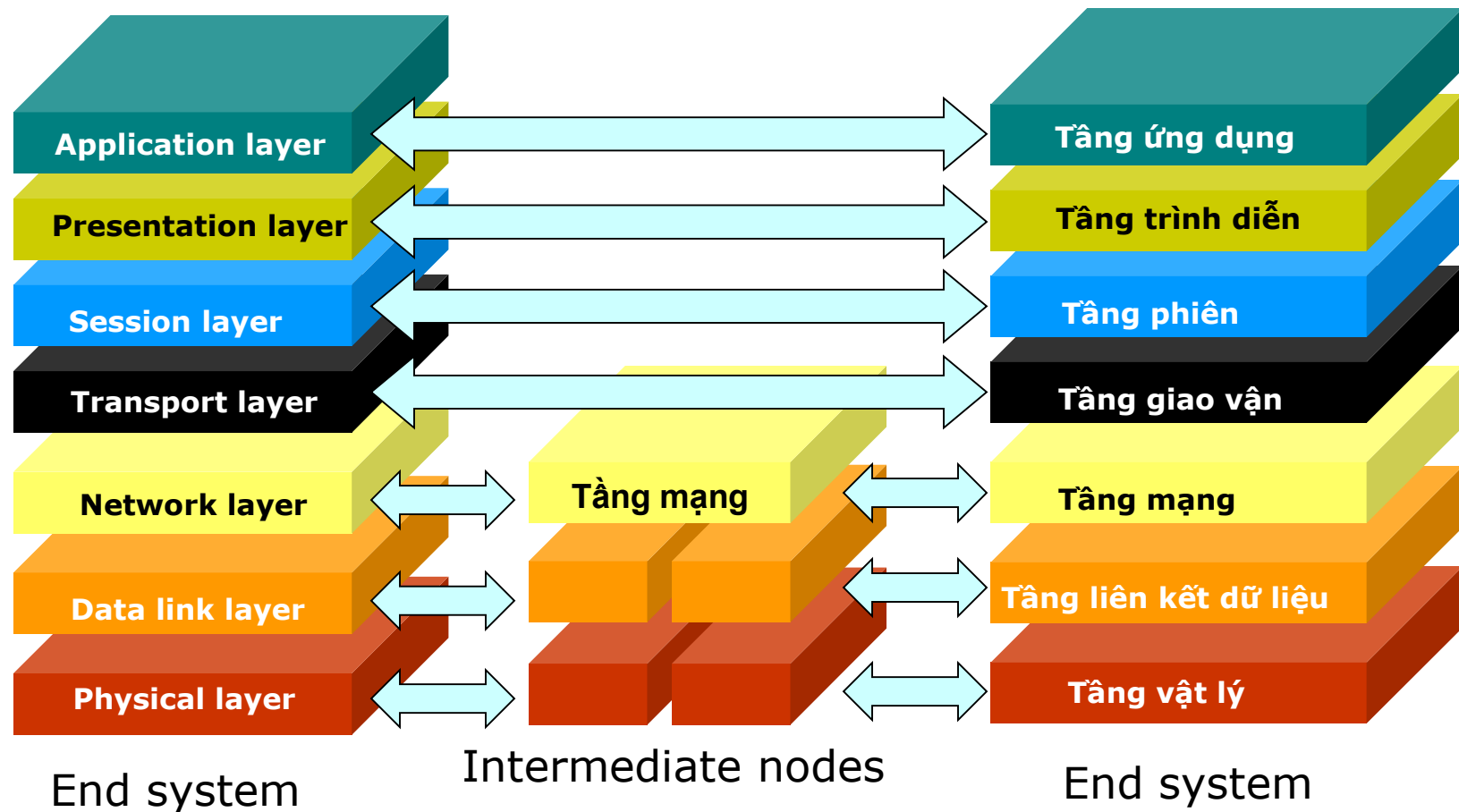
# Reference models

---

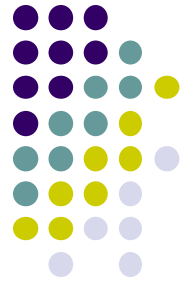
OSI  
TCP/IP



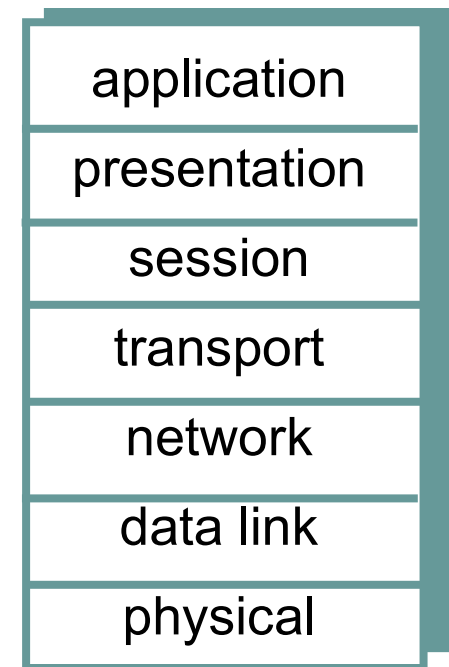
# OSI - Open System Interconnection: 7 layers



# The main functionality of each layers



- **Physical layer:** Transferring bits “over medium”
- **Datalink layer:** Transferring data between direct connected elements in the networks.
- **Network layer:** Routing, forwarding data from the source to the distant destination
- **Transport:** Transmitting data between applications
- **Session :** synchronization, check-point, recovery of transmission process
- **Presentation:** data encoding, compression, data conversion...
- **Application:** Supporting communications between distant parts of an application.

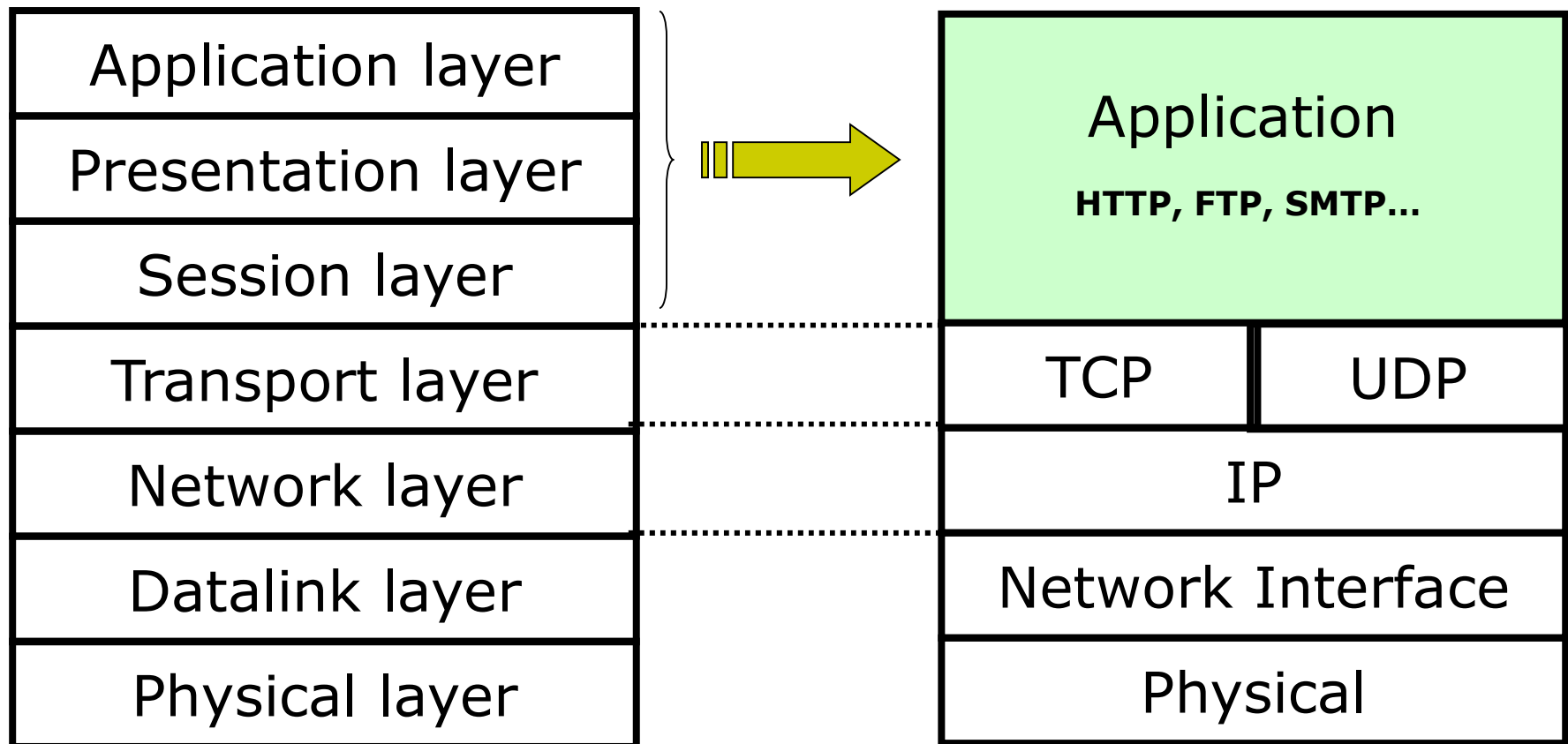






# Models OSI and TCP/IP

In the TCP/IP model of the Internet, the functionalities of 3 first layers are combined in a single layer.

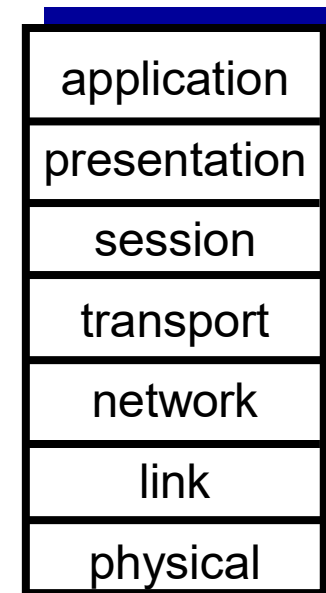




# ISO/OSI reference model

Two layers not found in Internet protocol stack!

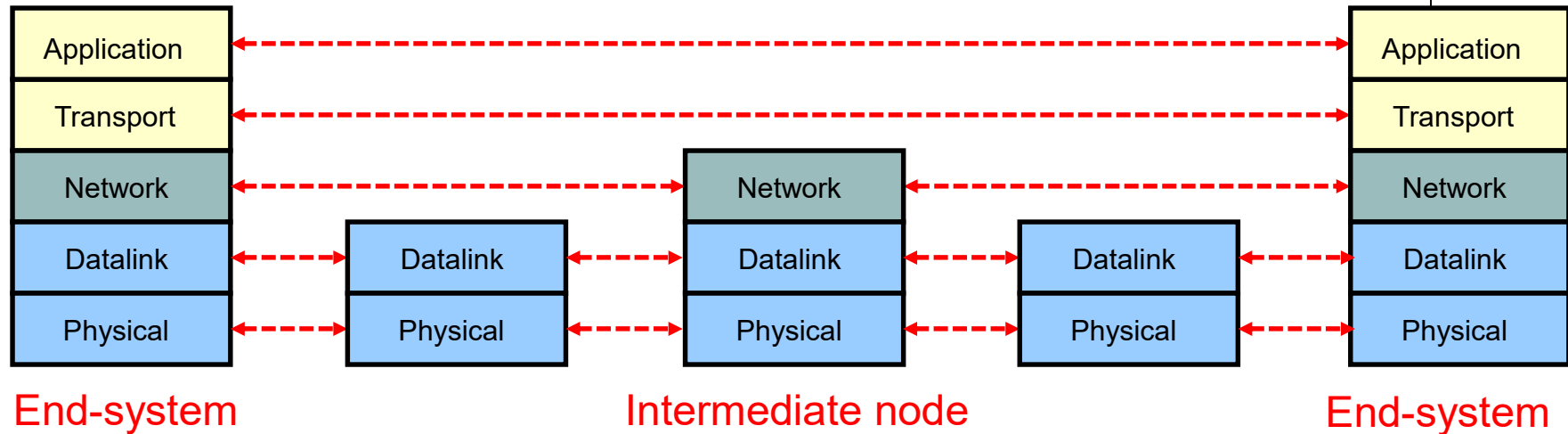
- *presentation*: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session*: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application
  - needed?



The seven layer OSI/ISO reference model



# Implementing the layering system

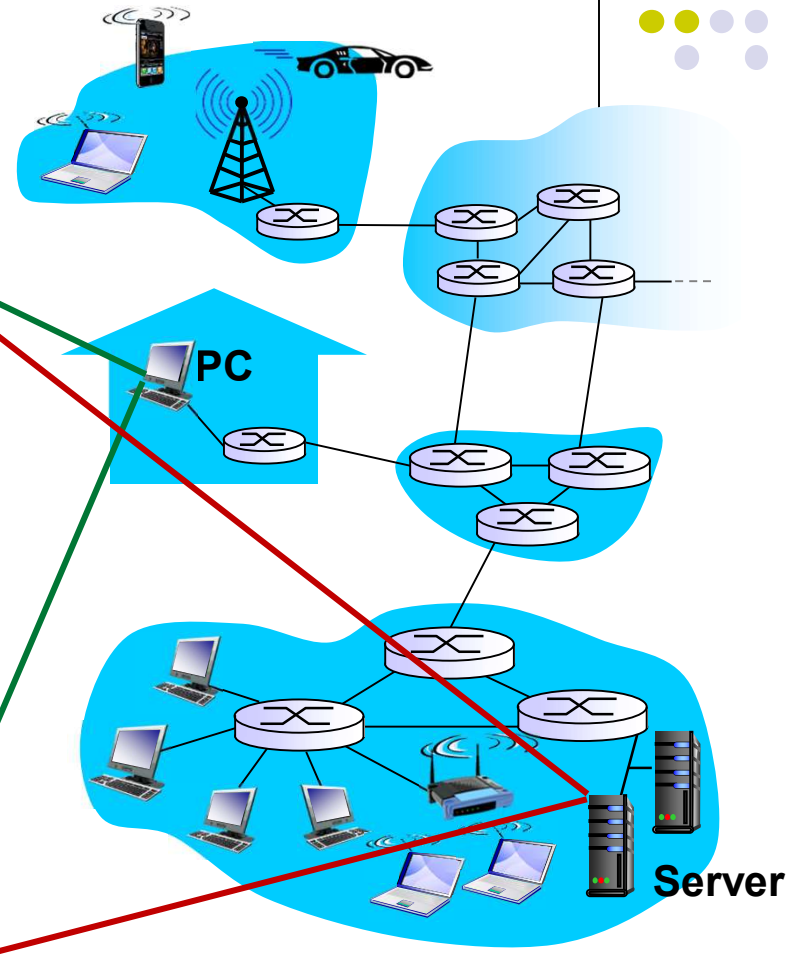


- End-systems: PC, server, smartphone...
- Intermediate node: Hub, Switch, Router

# Implementing the layering system

End systems (server, PC, smartphone...)

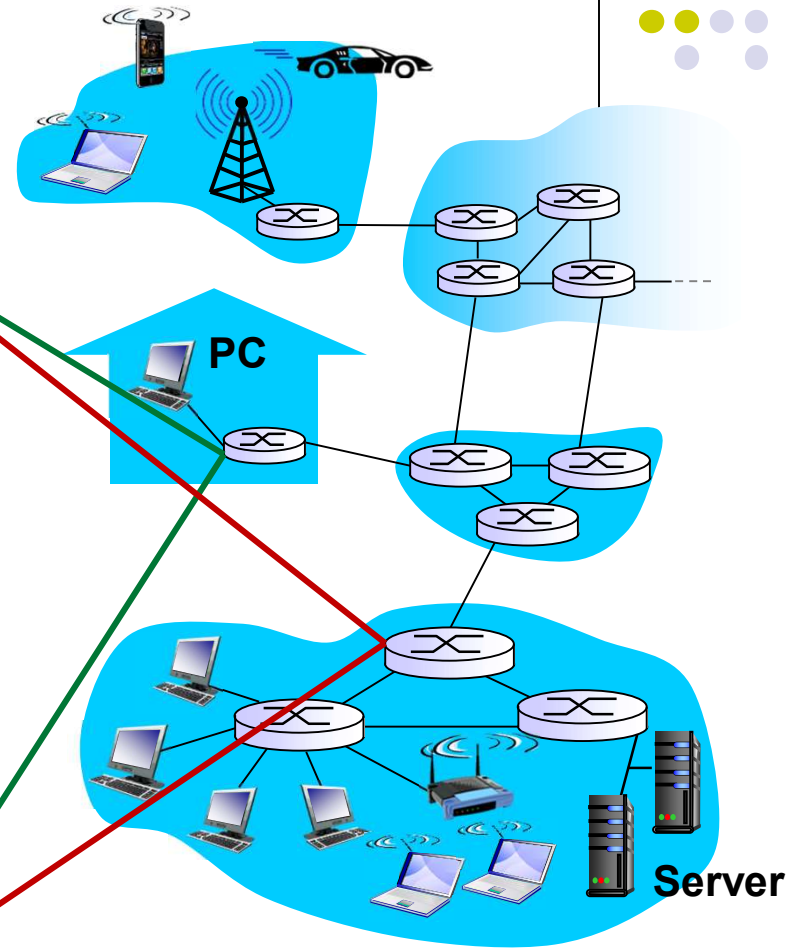
✓	Applications provide services for users
✓	Control data transmission between applications
✓	Routing, forwarding data
✓	Control data transmission on physical connections
✓	Convert data to physical signal and transfer



# Triển khai kiến trúc phân tầng

Intermediate router

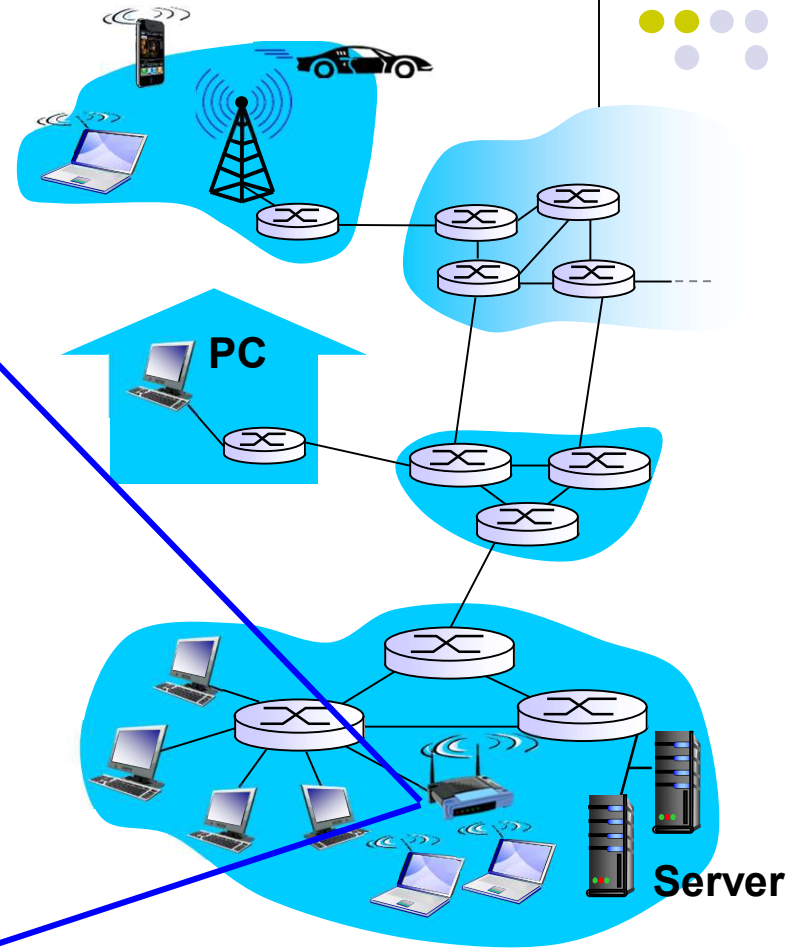
✗	Applications provide services for users
✗	Control data transmission between applications
✓	Routing, forwarding data
✓	Control data transmission on physical connections
✓	Convert data to physical signal and transfer

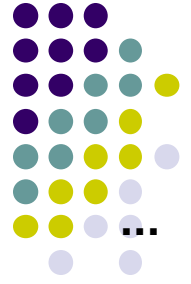


# Triển khai kiến trúc phân tầng

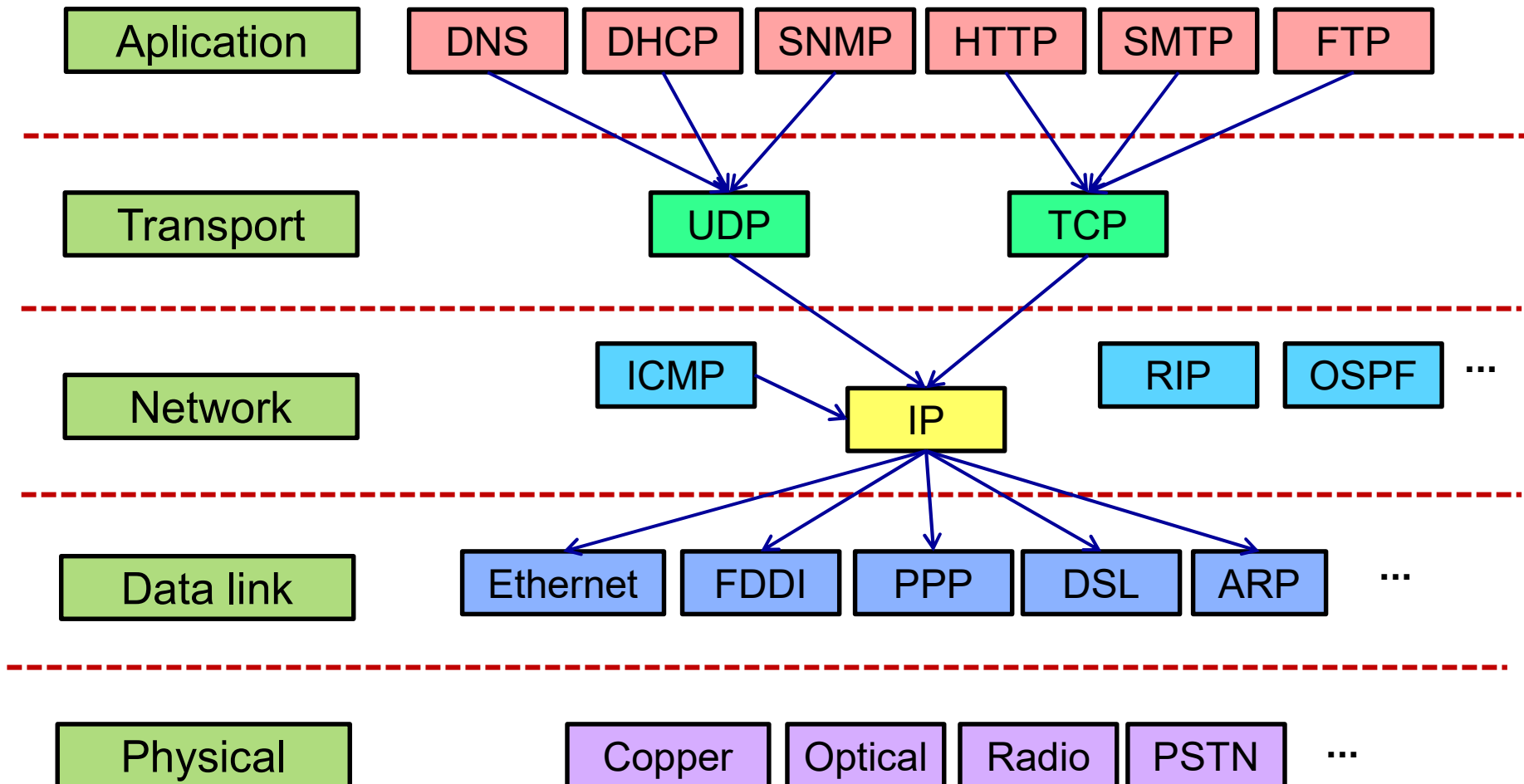
Intermediate switch

✗	Applications provide services for users
✗	Control data transmission between applications
✗	Routing, forwarding data
✓	Control data transmission on physical connections
✓	Convert data to physical signal and transfer





# Protocol stack of TCP/IP

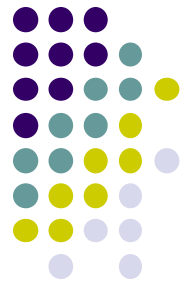




# Protocol stack of TCP/IP

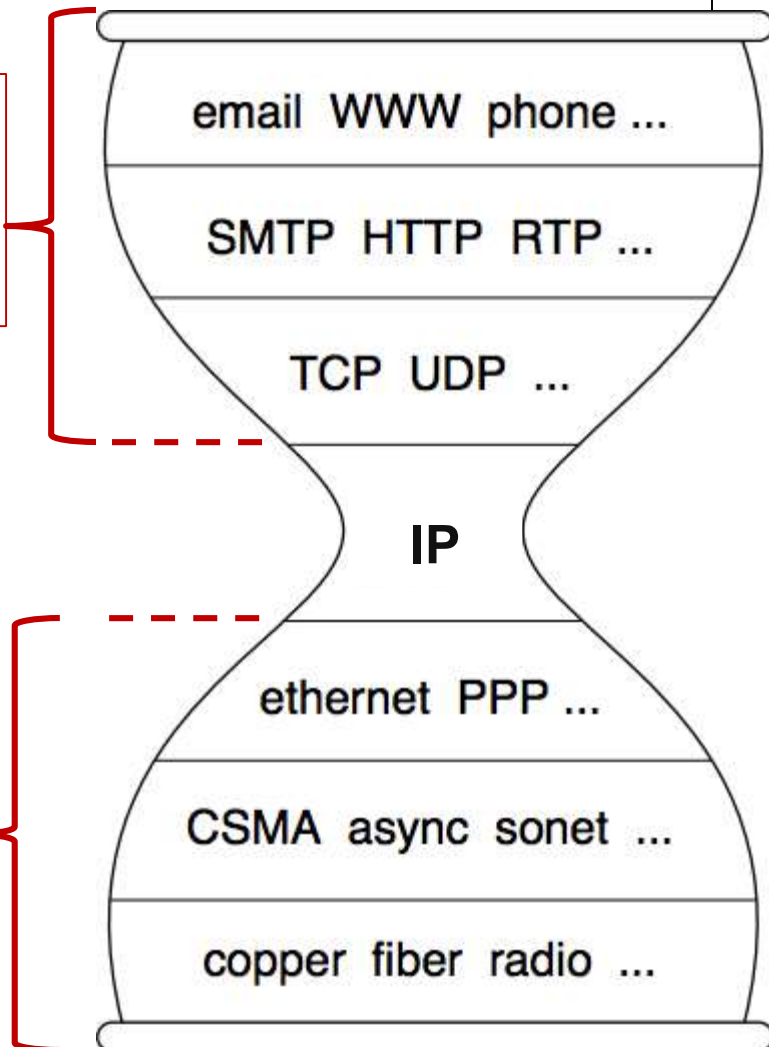
- “Hourglass”: only use IP – Internet Protocol at the network layer:
  - Allow a network use a new technology and connect to an existing one
  - Separate developing applications from technologies of lower layers
    - ➔ IP-based application: VoIP...
  - Support parallel modifications/ upgrading on both upper and lower layers
- However, it is extremely hard to upgrade to a new IP version (converting IPv4 to IPv6)





# Install TCP/IP on networks

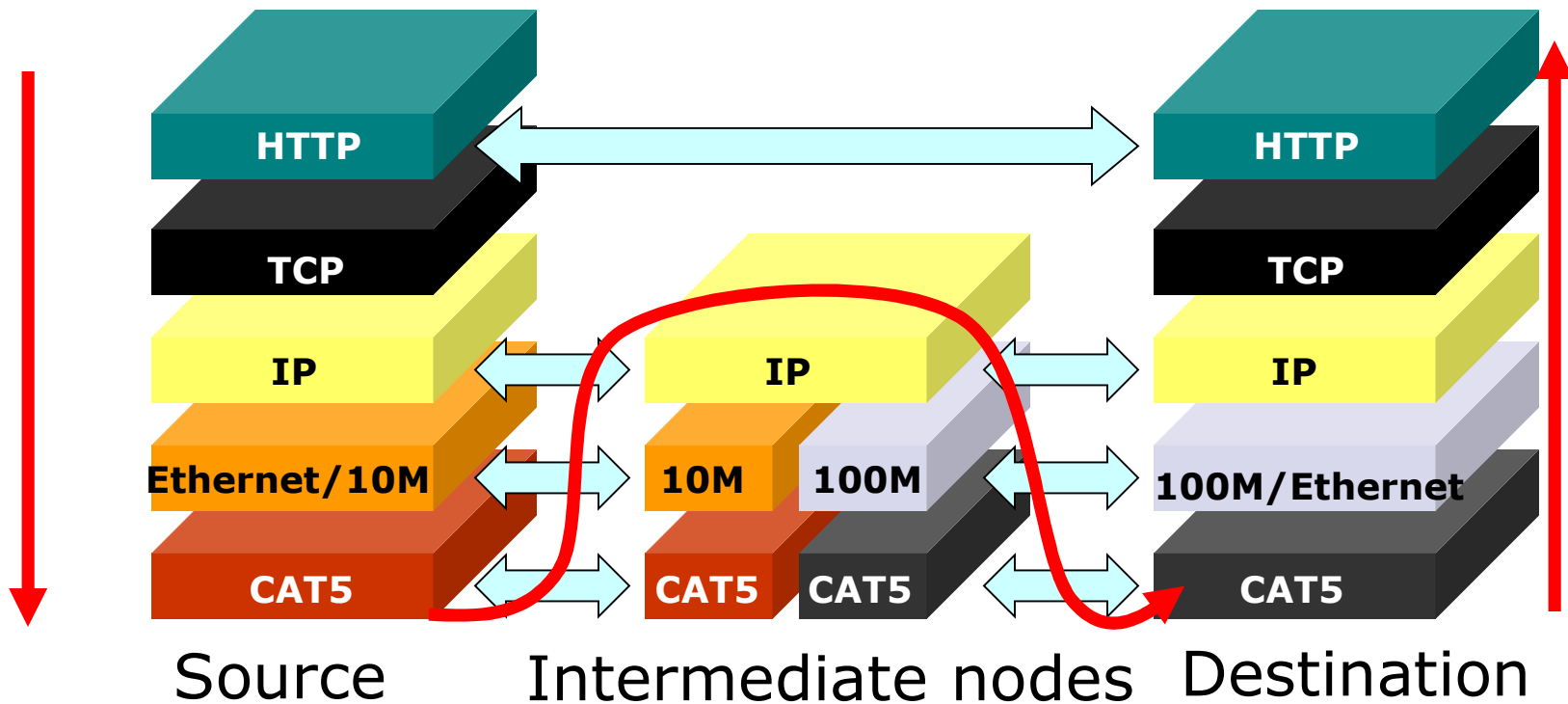
- End systems
  - Different on different applications
- 
- Same as every nodes
- 
- Same as every nodes
  - Different at different physical connections



# Layering model of the Internet



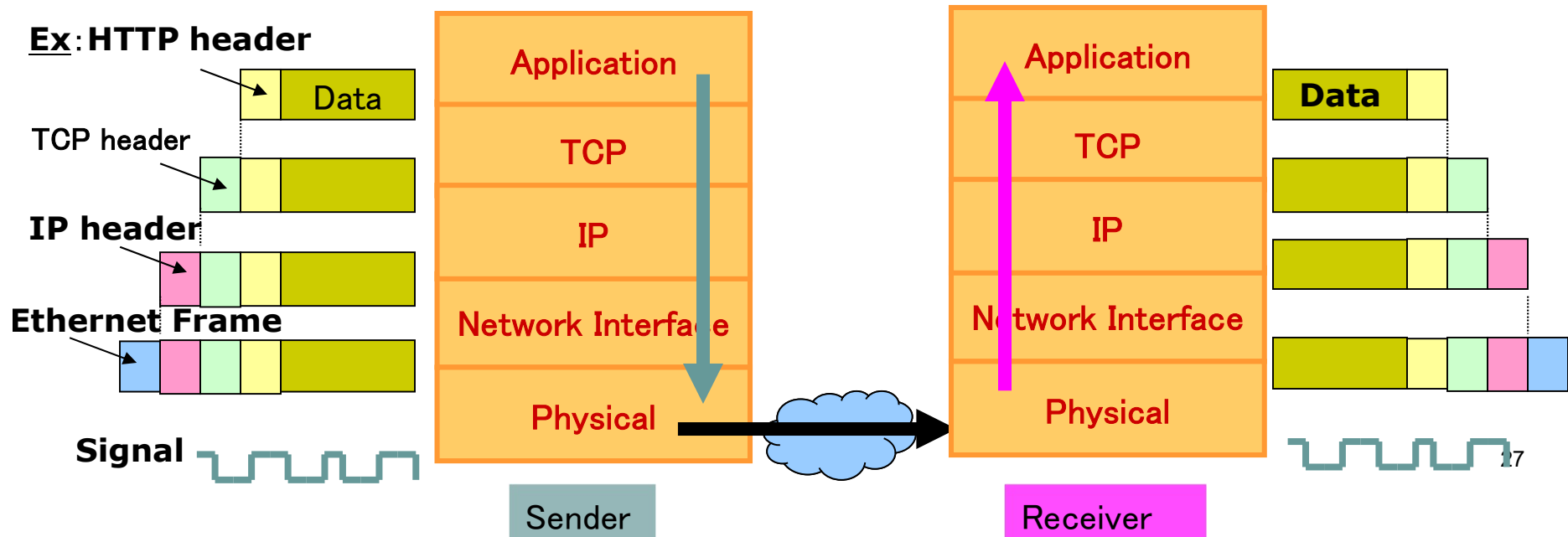
Example of data transmission from a source to a destination through intermediate nodes (router)

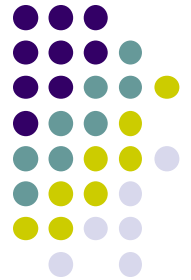


# The protocols TCP/IP and encapsulation process

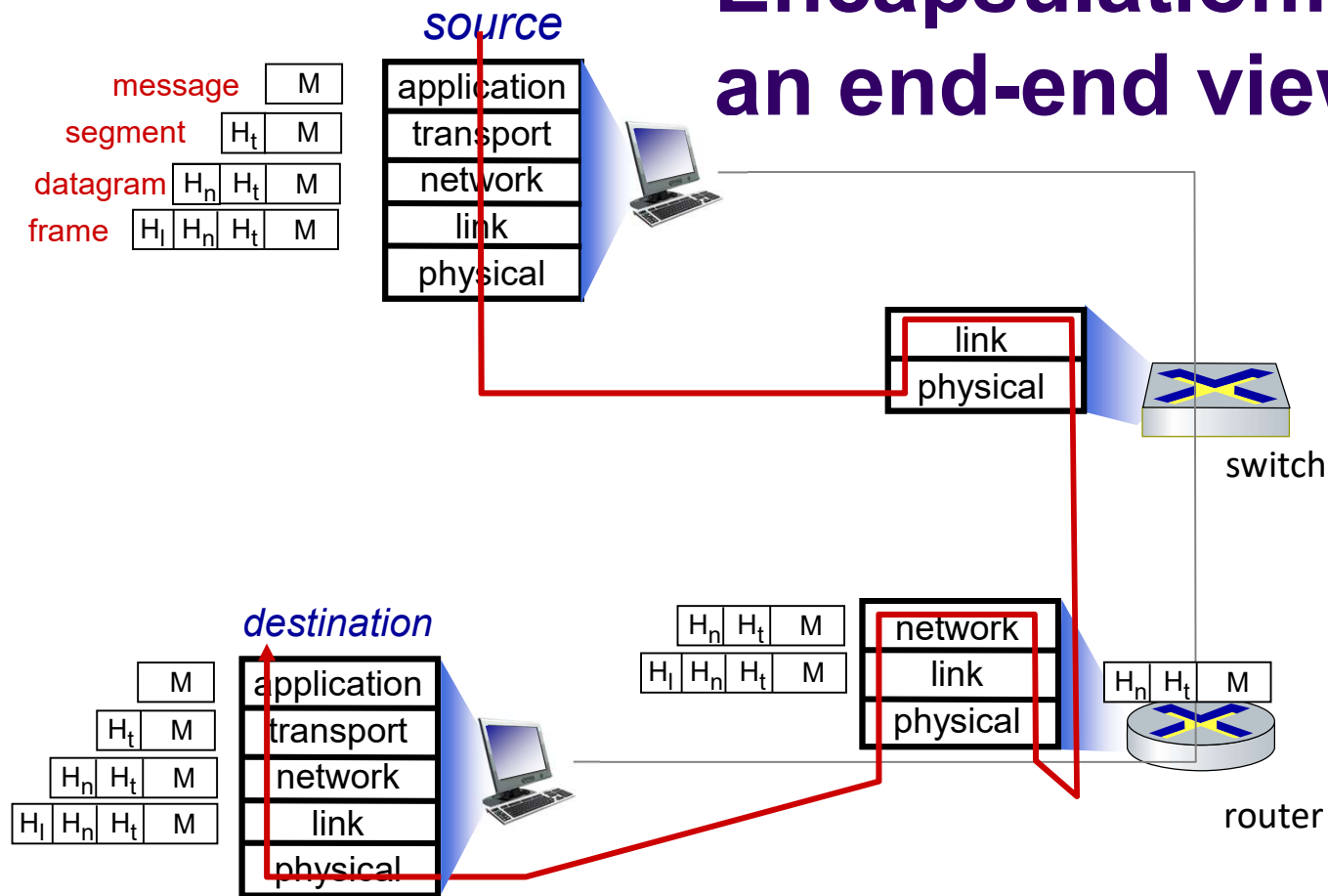


- At sender
  - Each layer add control information to the header of packet and transfer to the lower layer.
- At receiver
  - Each layer process packet according to the information of the header, then remove the corresponding header and deliver the remaining data to the upper layer.

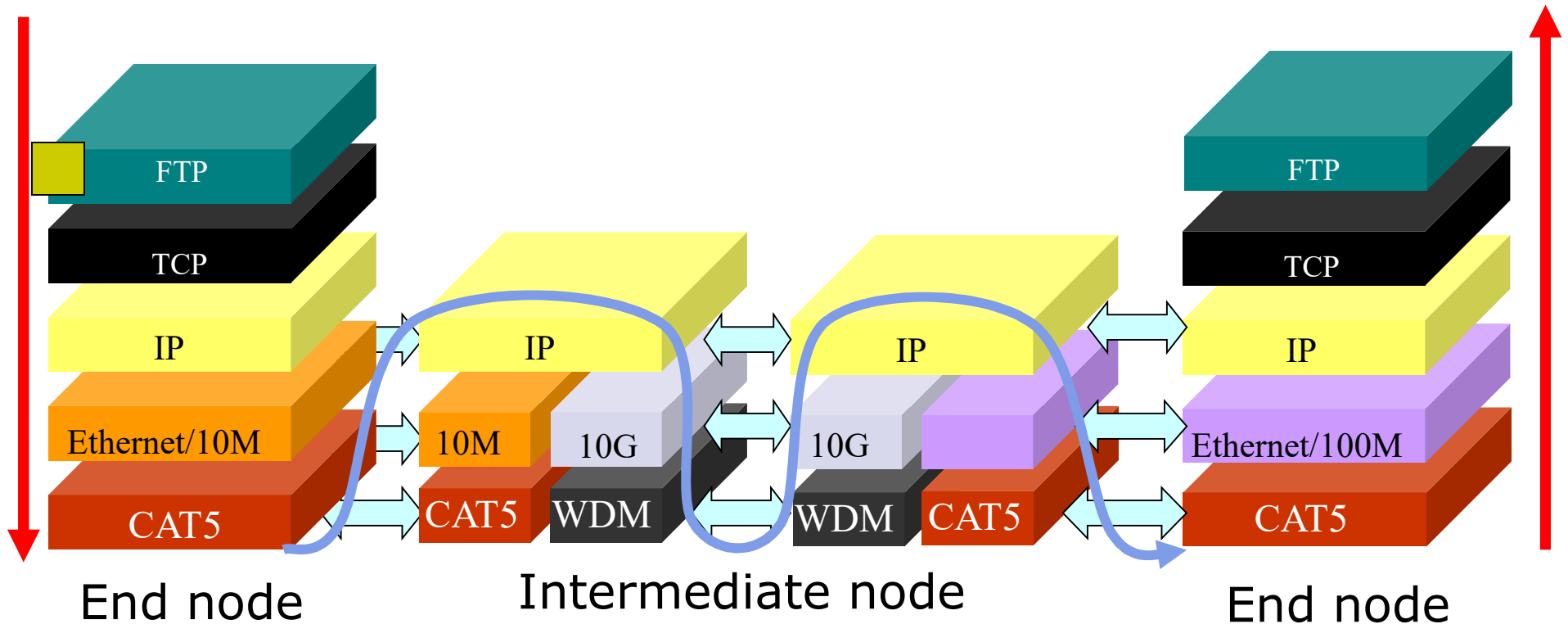




# Encapsulation: an end-end view

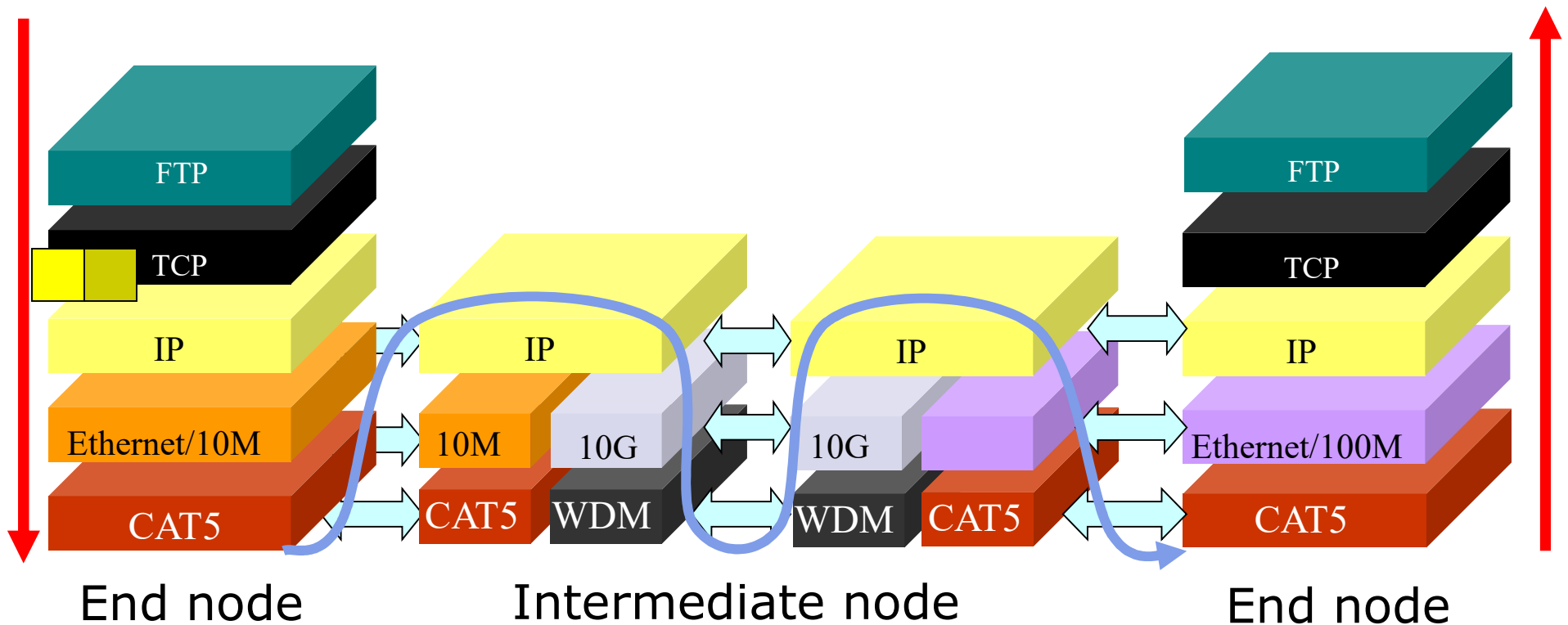


# Protocol stack and encapsulation



Dữ liệu - payload

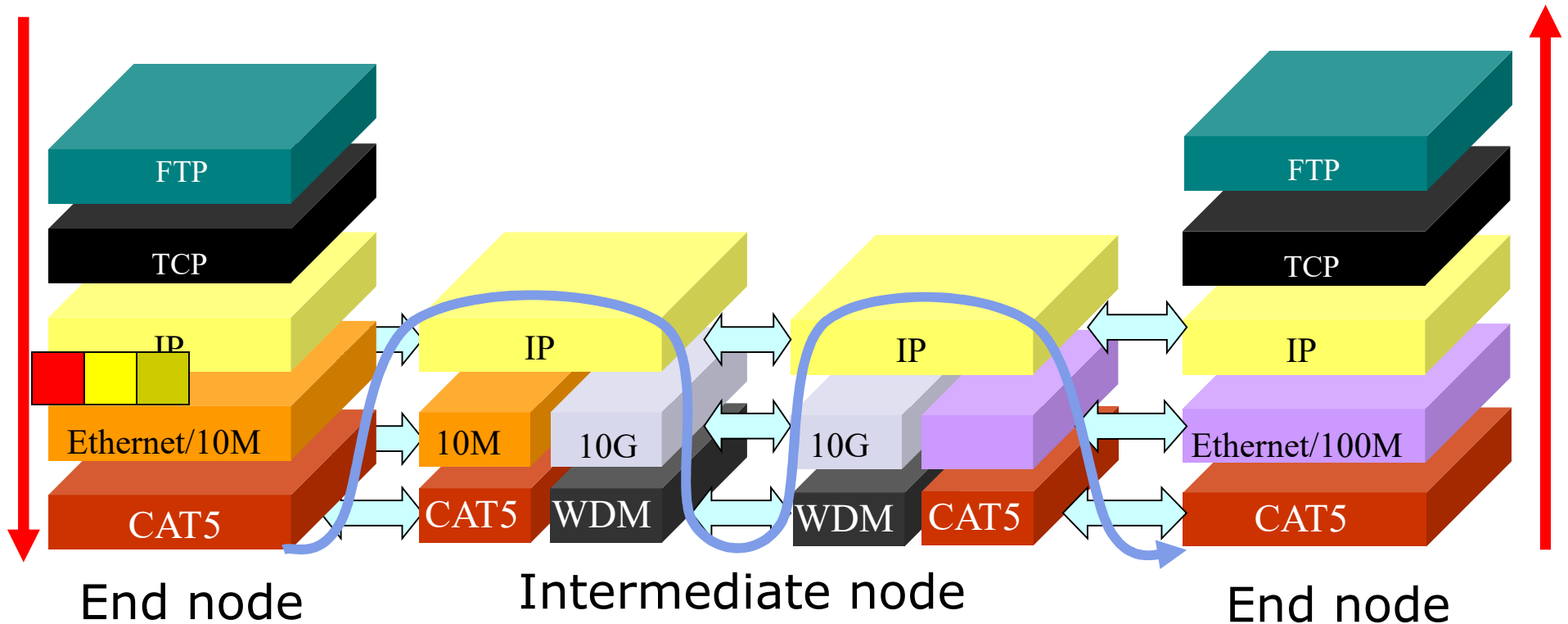
# Protocol stack and encapsulation



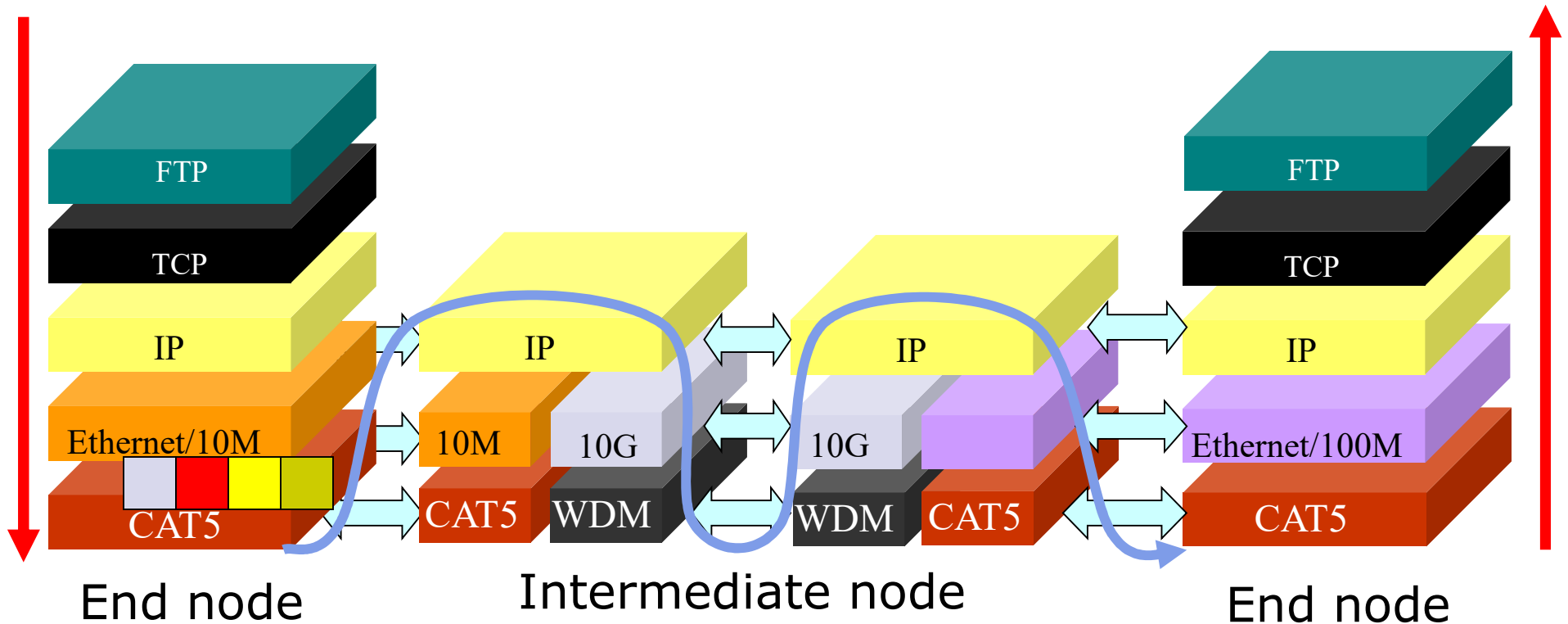
TCP header

Dữ liệu - payload

# Protocol stack and encapsulation

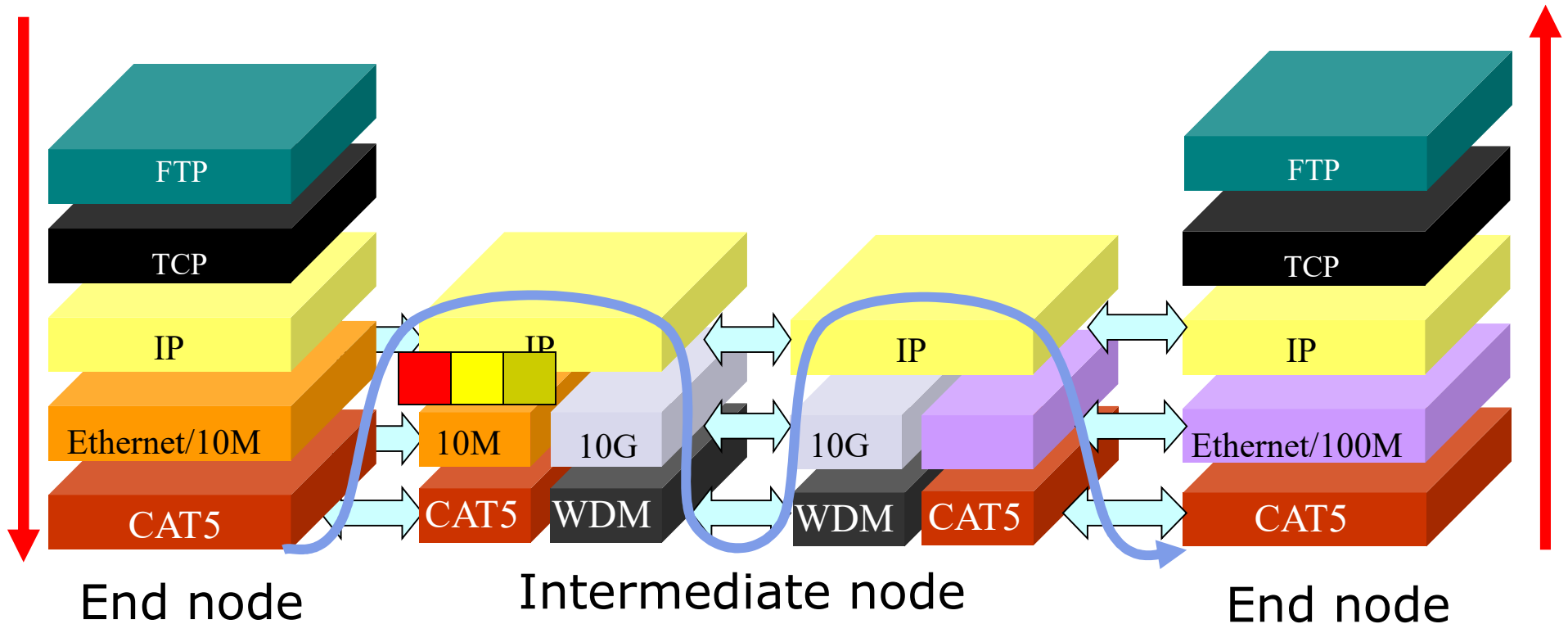


# Protocol stack and encapsulation

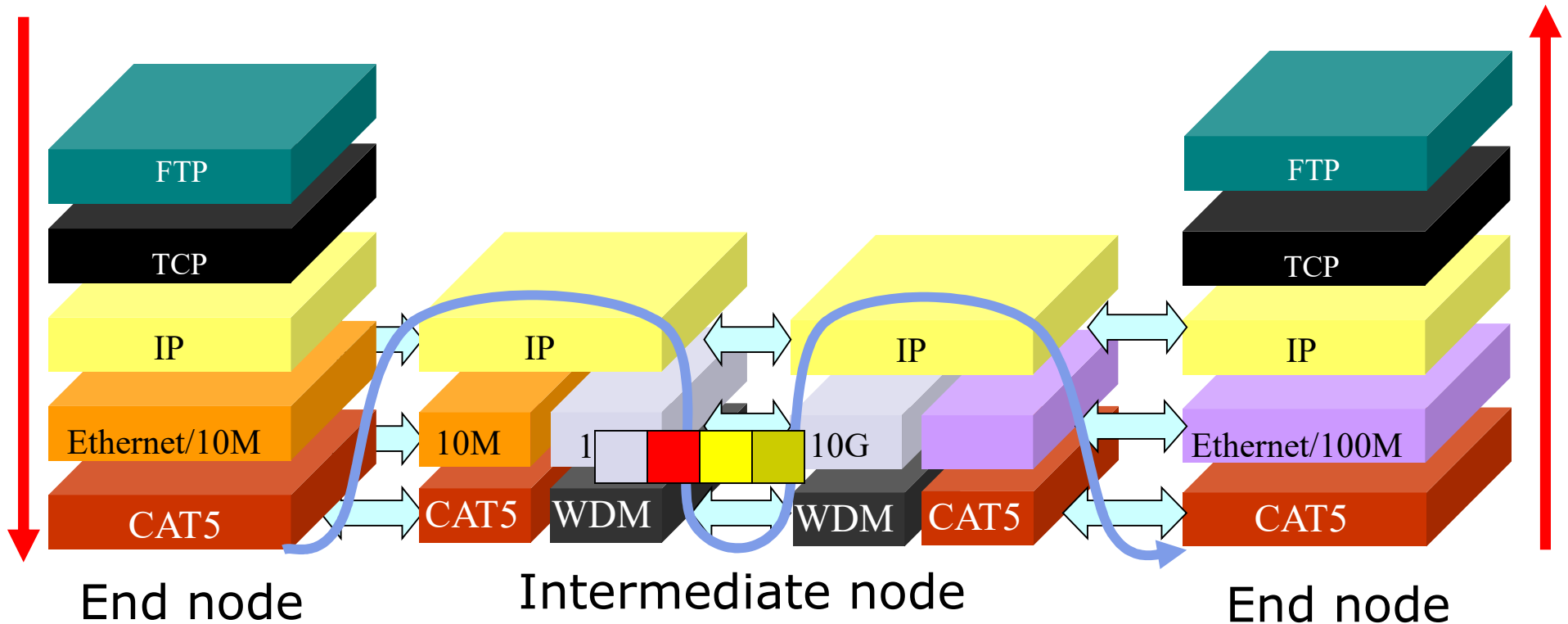




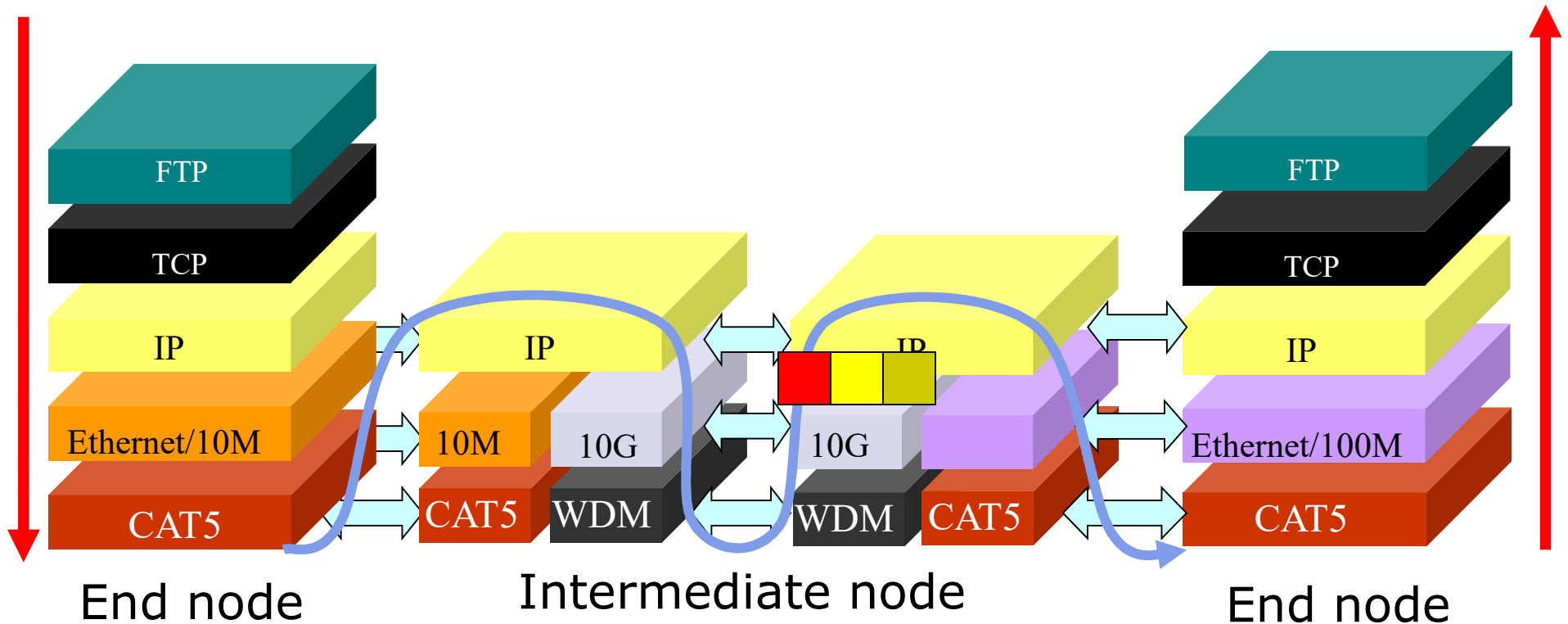
# Protocol stack and encapsulation



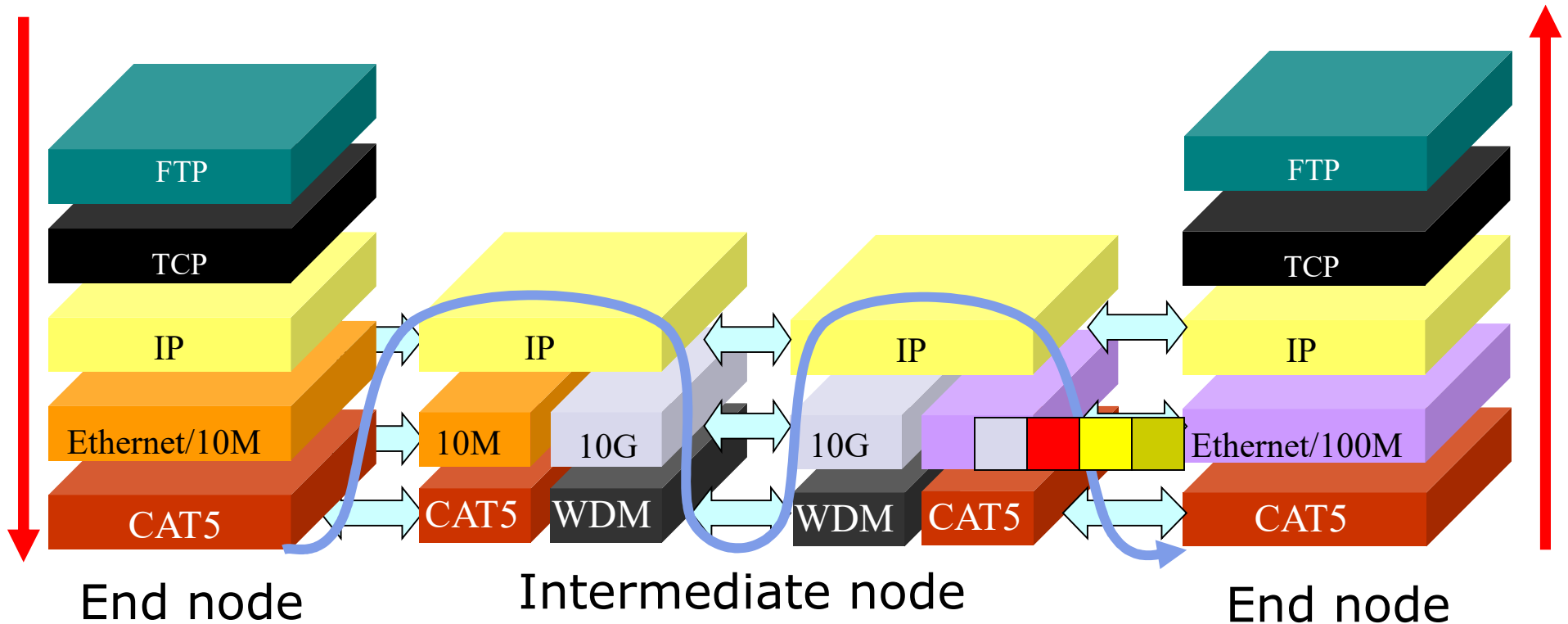
# Protocol stack and encapsulation



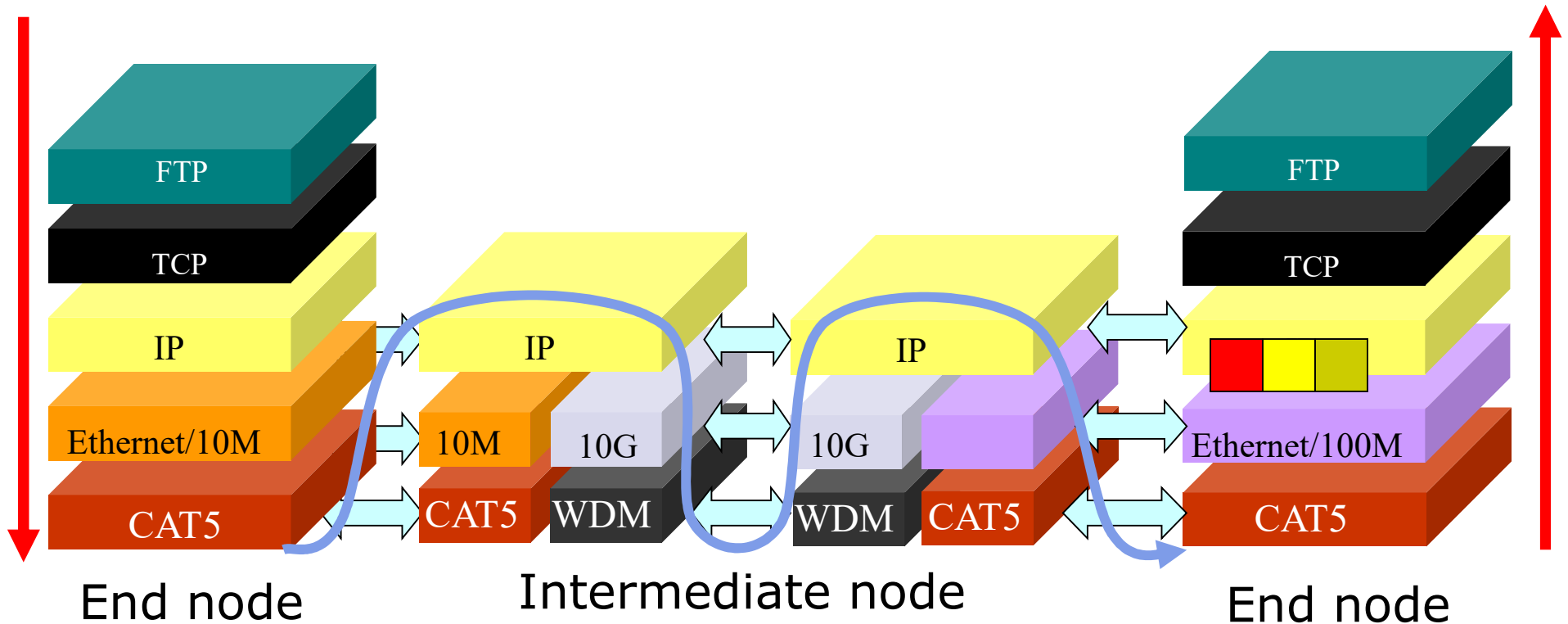
# Protocol stack and encapsulation



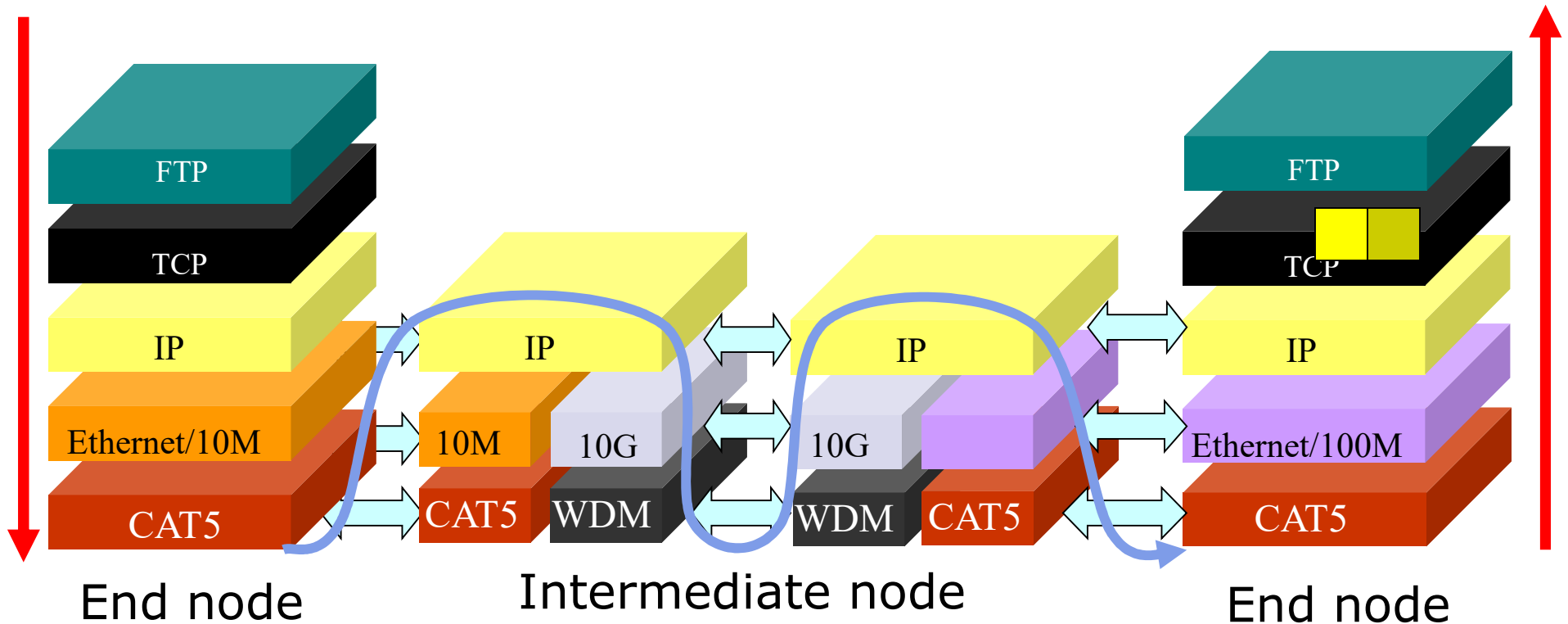
# Protocol stack and encapsulation



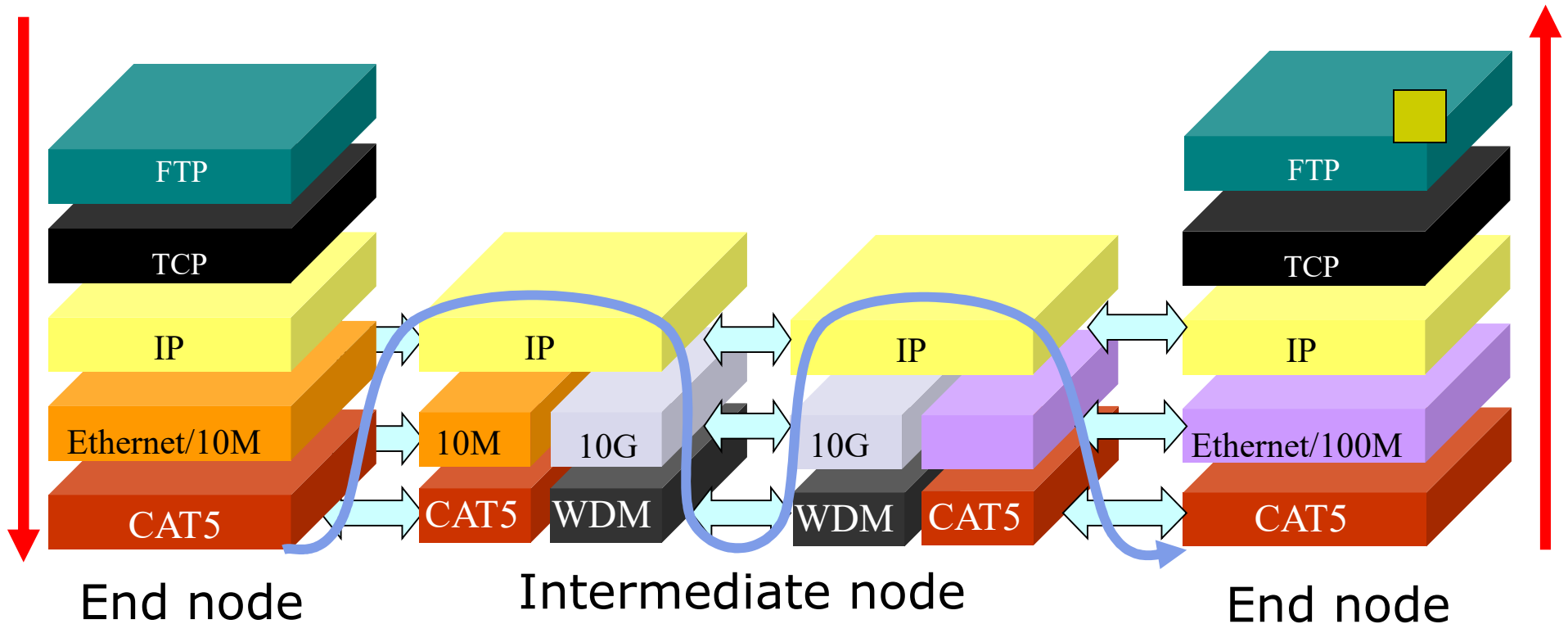
# Protocol stack and encapsulation



# Protocol stack and encapsulation

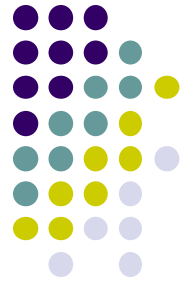


# Protocol stack and encapsulation



Dữ liệu - payload

# Summary: Advantage of layering architecture



- Layering architecture allows to divide the functionalities of networks into small components
- Layers are independent:
  - An upper layer makes use of the functionality of its right bellow layer but does not care about further layer.
- Extensibility/Scalability
- Flexible
  - It is possible to upgrade the communication system by upgrading the technology of some layers: Ex:
    - ADSL→FTTH
    - IPv4→IPv6
- Without layering:
  - Any change in the system requires changing the whole systems.



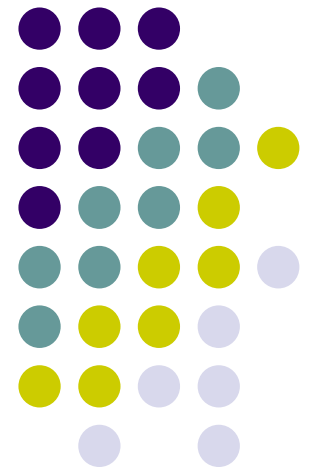
# Identification in the Internet

---

MAC Address

IP Address

Port number





# Identification

- Identification allows identify a person or an object
  - Name
    - Nguyen Thuc Hai
  - Address
    - 1 Dai Co Viet, Hai Ba Trung, Ha Noi
  - Telephone number
    - 8680896
  - Email
    - hai--xxx@it.hut.edu.vn

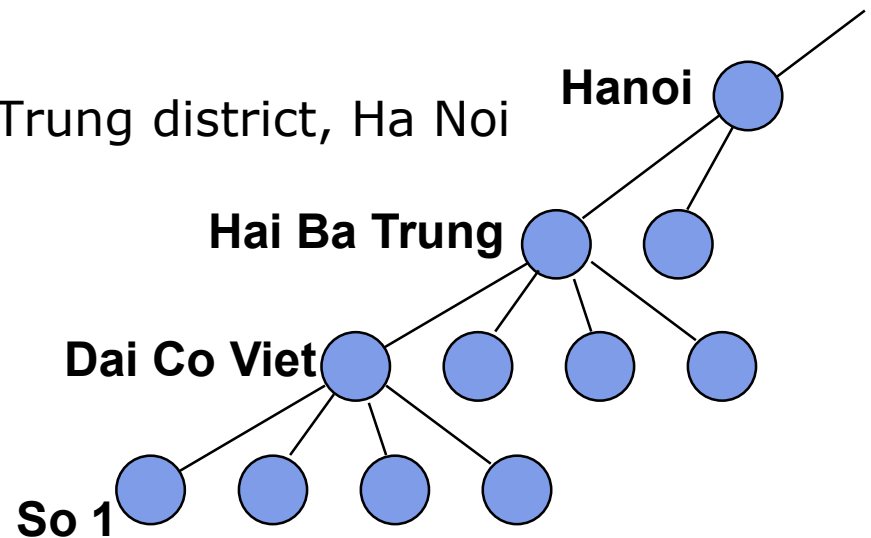


# Identification

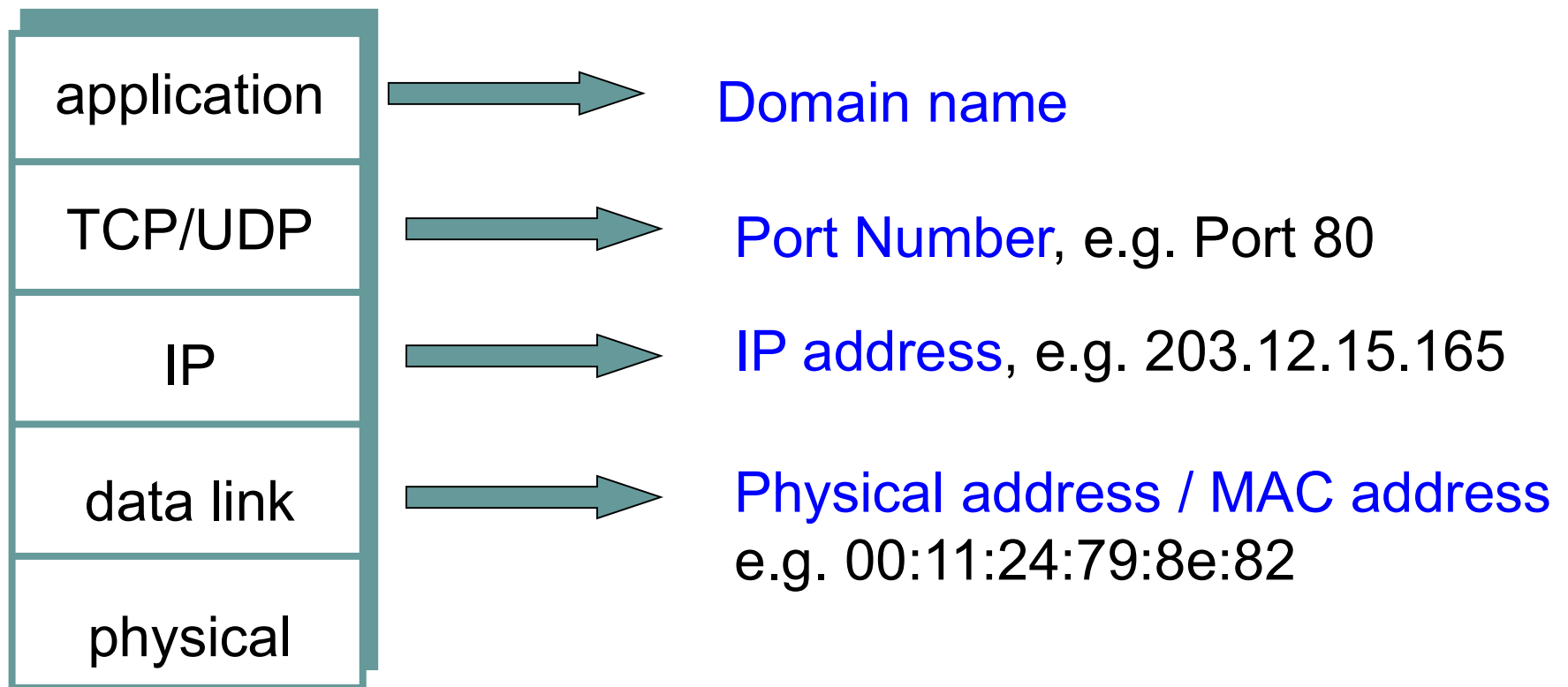
- Identification usually has hierarchical structure
  - Allow to manage efficiently a large addressing space
  - Scalability

- Example of hierarchy

- Address
  - 1 Dai Co Viet street, Hai Ba Trung district, Ha Noi
- Telephone number
  - +84-(4) 868-08-96



# Identification in the Internet and the relationship between layers





# Addressing in the Datalink layer

- Physical address/ MAC address
  - Using in Datalink layer
  - Fixed on NIC (Network Interface Card)
  - Used for identifying machine in broadcast network segment.

HEX      00:11:24:79:8e:82

BIN      00000000 00010001 00100100 01111001 10001110 10000010

            └────────────────┘      └────────────────┘

            OUI                      Assigned by manufacture

OUI (Organizationally Unique Identifier):  
Each Manufacture have an some OUI unique



# Addressing in the Internet

- IP address
  - Used in IP-**Internet Protocol** (network layer)
  - Value depends on the networks. Each network interface card should be assigned an IP address.
  - Used for identifying a machine in an IP network, example:
    - 133.113.215.10 (ipv4)
    - 2001:200:0:8803::53 (ipv6)



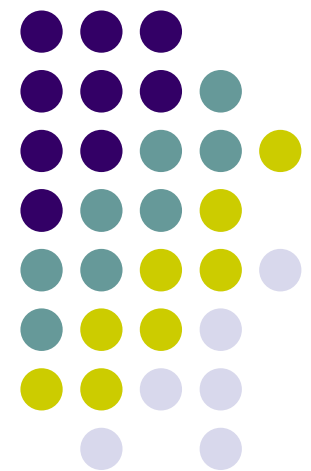
# Addressing in transport layer

- Port number
  - On each machine, there may be several applications running.
  - Applications of the same machine are distinguished by port number.
  - An application instance in the internet is identified by the IP address of the host and port number on which it runs
  - Similar to the address of a room in a building
    - Building address: B1 Building, 1 Dai Co Viet, Ha Noi => **similar to IP address**
    - Room number 325 => **Similar to port number**
- E.g. HTTP runs on port 80, FTP runs on ports 20, 21 ...

# Mapping to address

---

Domain name  
Domain name resolution  
nslookup  
arp



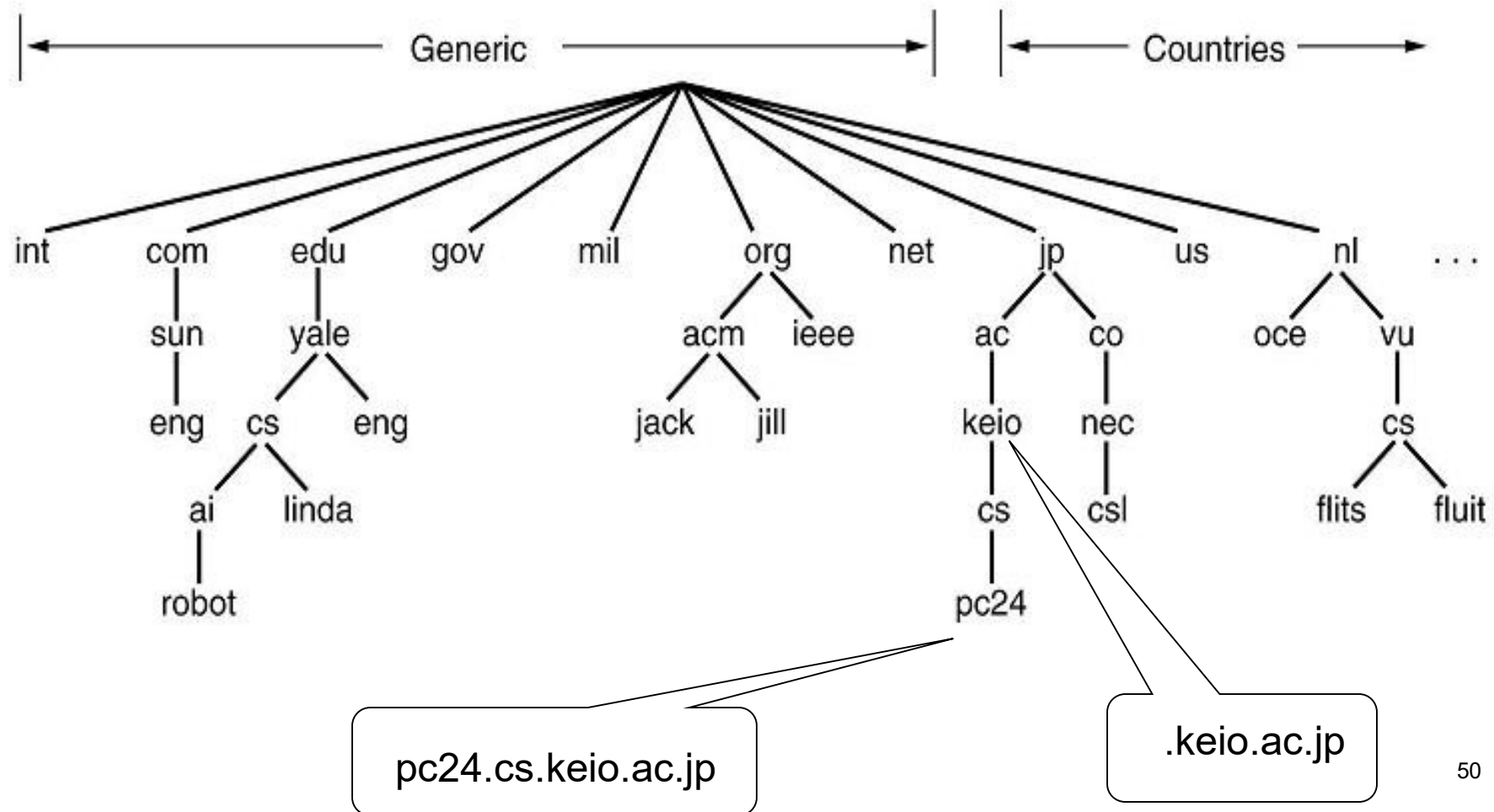
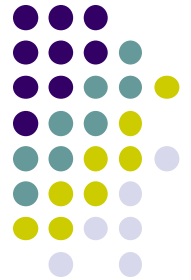




# Domain name

- Domain Name  
(FQDN: Fully Qualified Domain Name)
  - Domain name is the name of a computer or a network using alphabet and numbers
    - www.keio.ac.jp
    - www.hedspi.hut.edu.vn
    - .hut.edu.vn

# Domain name space



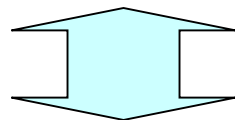


# Domain name and IP address

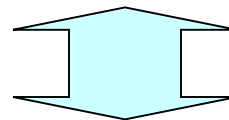
- For sending data to a host/machine, the host must be identified
  - By an IP address
  - By a domain name (easy to be memorized by human)
- name
  - Variable length
  - easy to be memorized by human
  - Nothing to do with the location of the host
- IP address
  - Fixed length (32 bits or 128 bits)
  - Computer process address more easily
  - Used for routing purpose

203.162.7.194

www.hedspi.hut.edu.vn



www.hust.edu.vn



202.47.142.40

# Conversion/resolution of address



- Computer prefers numbers
- Human prefers names



Need address conversion



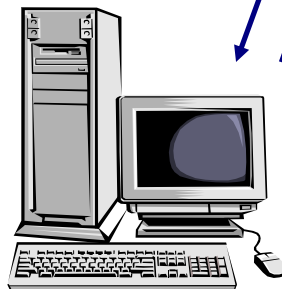
User

I want to see the webpage  
www.hust.edu.vn

Need to access to  
202.47.142.40



Domain name server



Webserver of  
www.hust.edu.vn  
202.47.142.40

*If user knows the IP address of  
the webserver, he can access  
via IP address from Browser*

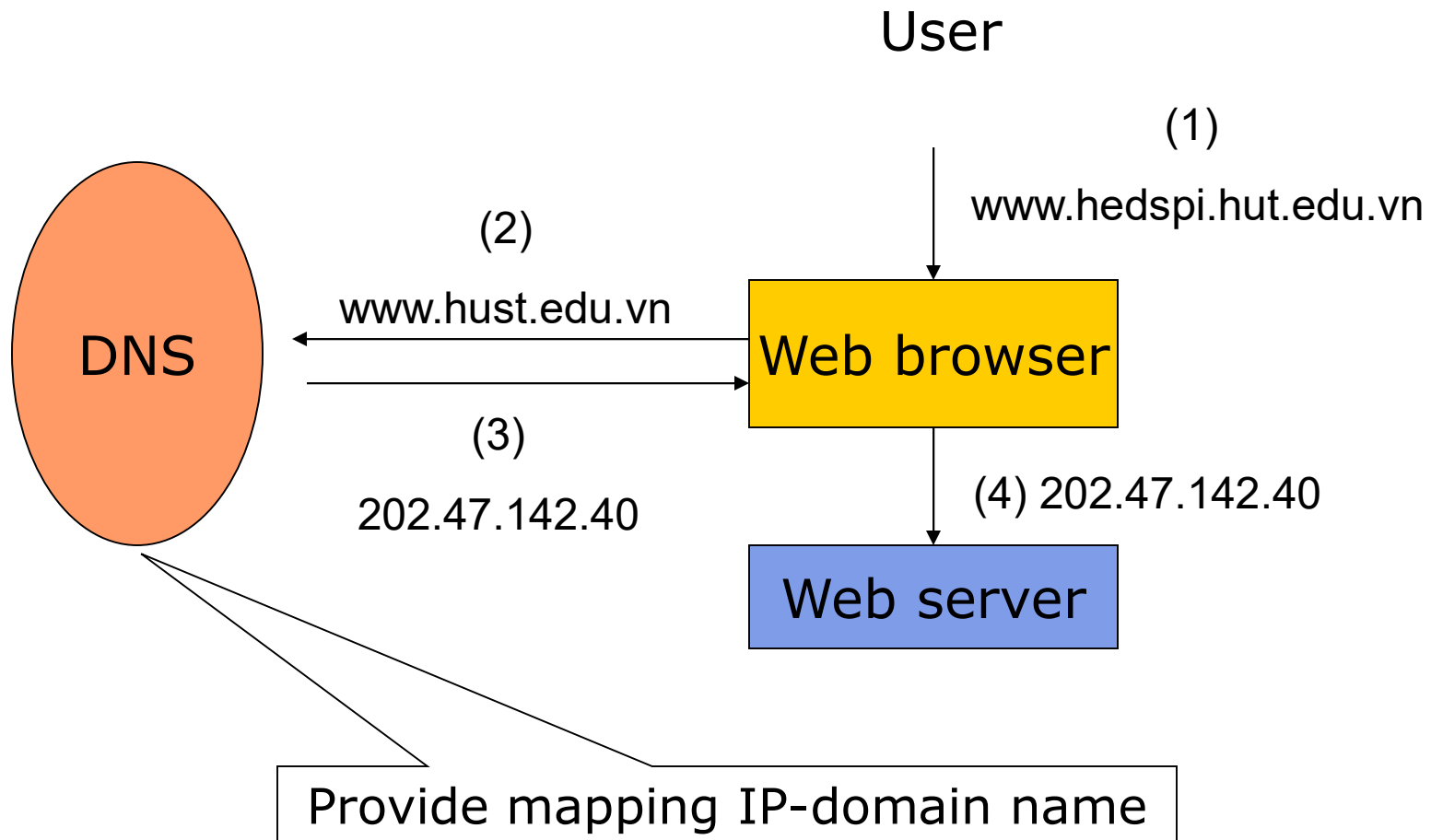


# Address resolution/conversion

- Concept
  - Mechanism finding address IP from a domain name and vice versa.
  - There is no mathematical formula for this conversion.
- Domain name server (DNS)
  - Store the mapping of IP address and Domain name of the same host in a database
  - Answer requests of resolution of IP or domain name from users.
  - Widely used in the Internet



# Example



# Nslookup tool on Windows, Linux



- nslookup www.hedspi.hut.edu.vn
- Conversion "name $\Leftrightarrow$  IP address"

```
C:\Documents and Settings\hongson>nslookup www.hedspi.hut.edu.vn
```

```
Server:
```

```
Address: 192.168.1.1
```

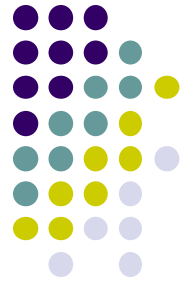
```
Non-authoritative answer:
```

```
Name: www.hedspi.hut.edu.vn
```

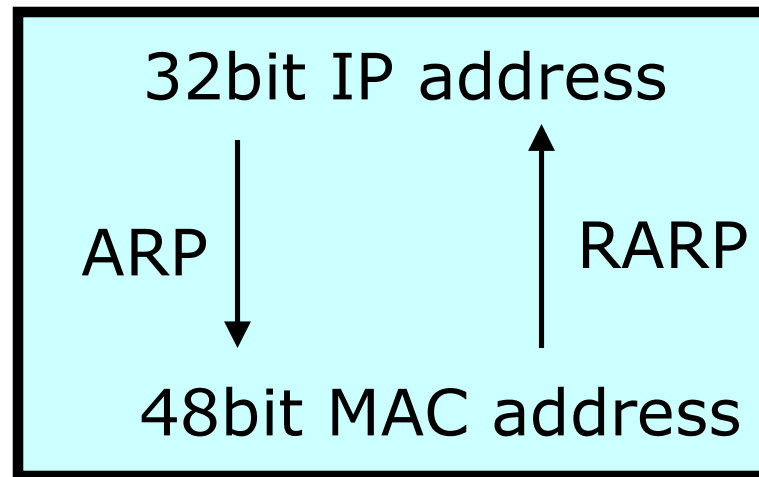
```
Address: 202.47.142.140
```

```
C:\Documents and Settings\hongson>
```

# ARP Conversion of Mac address and IP address

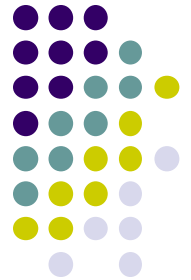


- Address Resolution Protocol
- MAC and IP are both used for identifying a NIC.
- ARP allows to find MAC address from IP address





# Example: ARP table (on Windows)



```
C:\Documents and Settings\hongson>arp -a
```

```
Interface: 192.168.1.34 --- 0x2
```

Internet Address	Physical Address	Type
192.168.1.1	00-02-cf-75-a1-68	dynamic
192.168.1.33	08-00-1F-B2-A1-A3	dynamic

**IP address**

```
C:\Documents and Settings\hongson>
```

**MAC address**



# Summary

- Layer architecture
  - Why layering
  - Model TCP/IP vs. Model OSI
  - Encapsulation, PDU. SAP
- Addressing on Internet
  - Address IP, MAC, domain name, port
  - Address conversion

# Quizz

