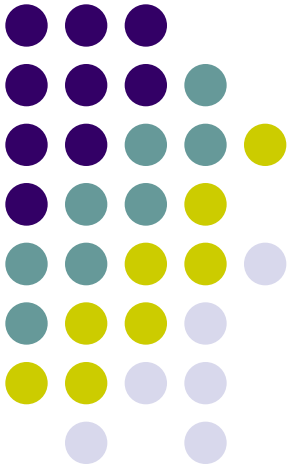
ELEN 4017

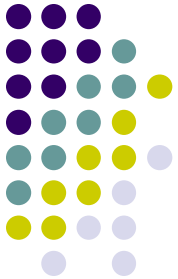
Network Fundamentals

Lecture 5



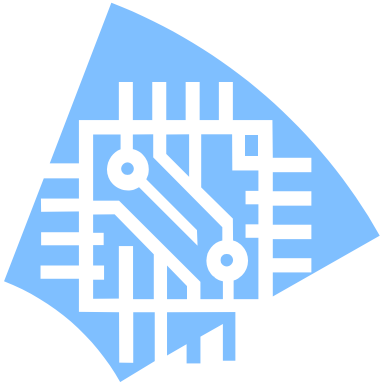
Purpose of lecture

- Continuation of roadmap for course
- Kurose chapter 1



Outline

- Protocol Layers and their Service Models



Protocol “Layers”

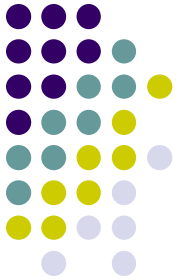
Networks are complex!

- many “pieces”:
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

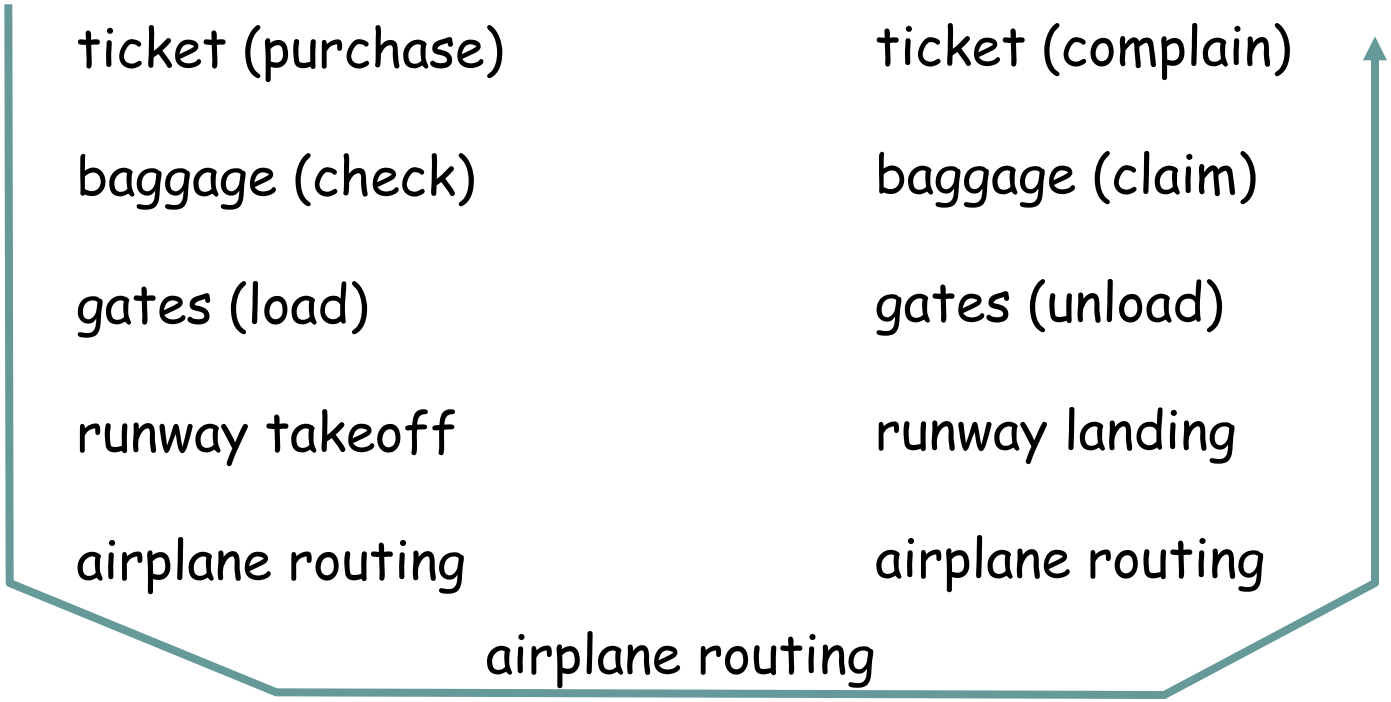
Question:

Is there any hope of *organizing* structure of network?

Or at least our discussion of networks?

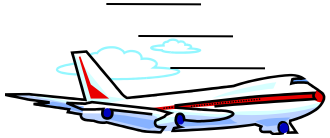


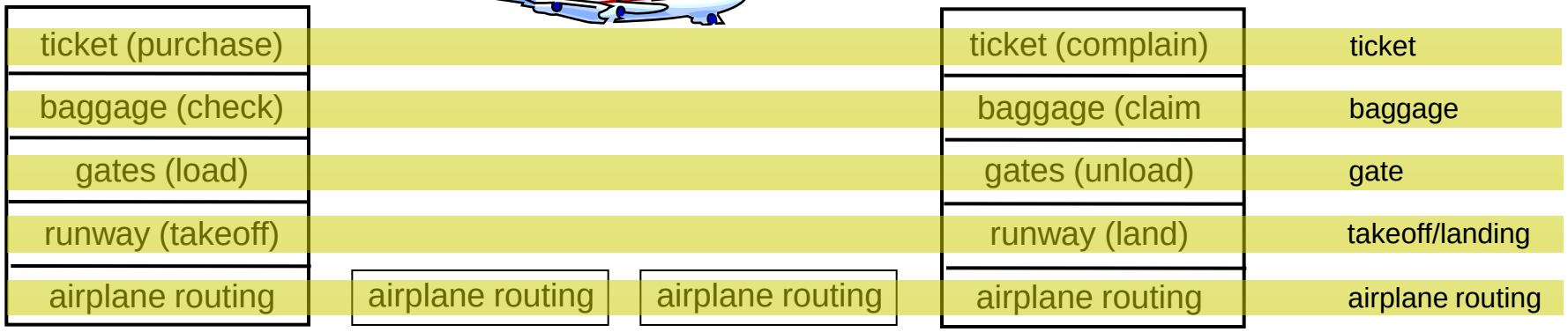
Organization of air travel



- a series of steps

Layering of airline functionality

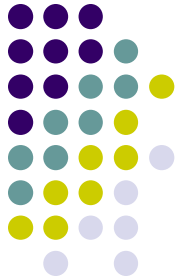




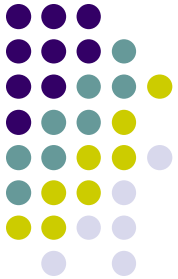
Layers: each layer implements a service

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

Horizontal view



- Each layer, combined with layer below it, provides some service.
  - At ticket layer, airline-counter to airline-counter service is provided.
  - At baggage layer, baggage-check to baggage-claim service is provided. Note: Baggage check provides this service to an **already ticketed person**.
  - And so on …
  - Thus each layer provides:
    - Certain actions within that layer.
    - Uses the services of the layer below it.
    - Consider the baggage check layer.

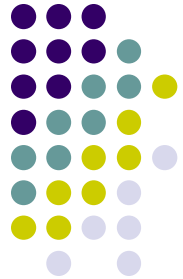


Why layering?

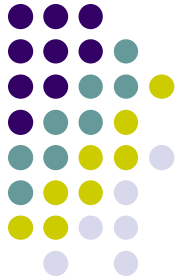
Dealing with complex systems:

- **explicit structure** allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of **implementation** of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system → board people in height order.

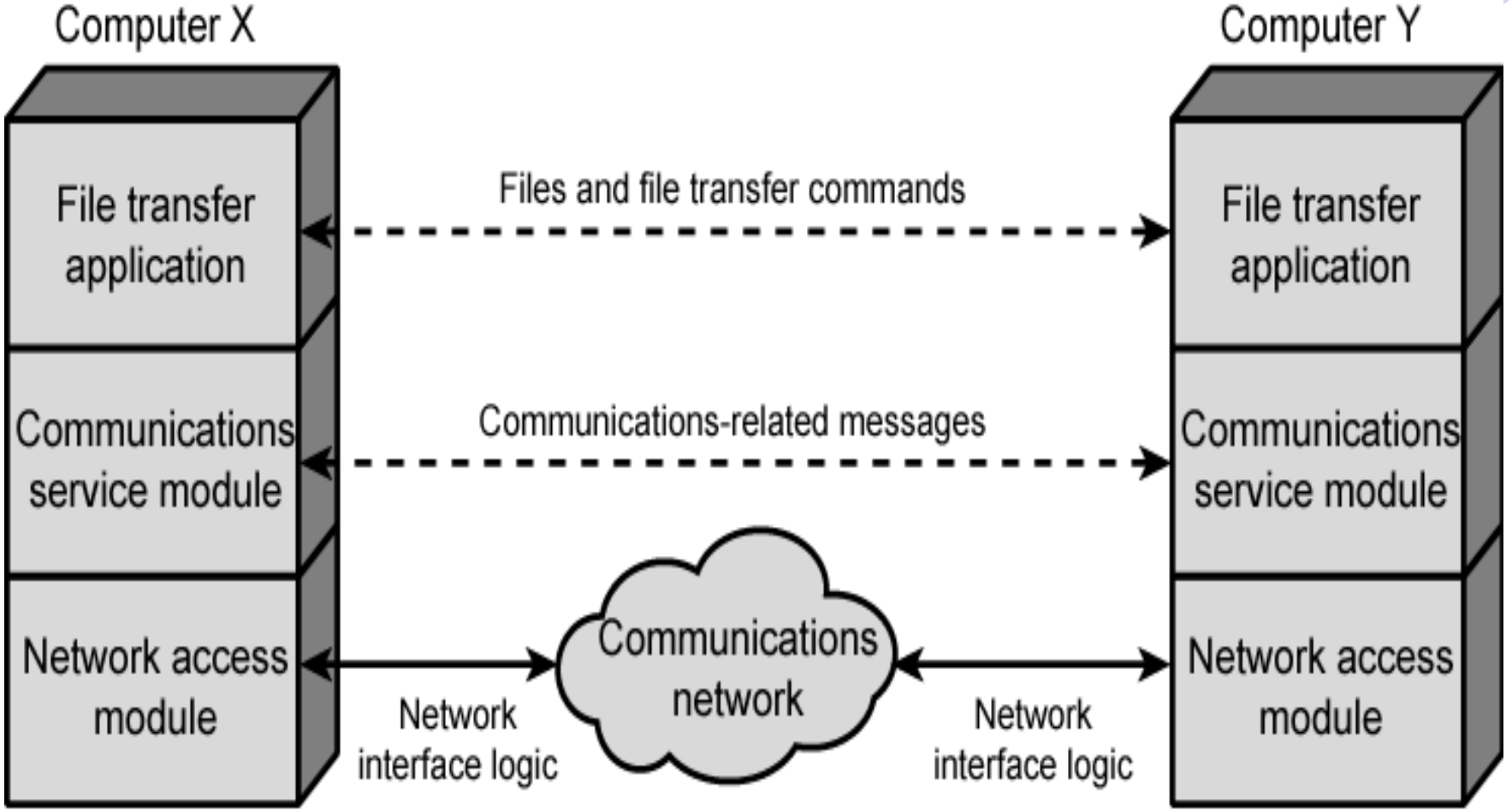
Benefits of layers - Decoupling



- By decomposing into layers, you make the **interfaces** explicit i.e. layer x expects a certain interface of layer y.
- Thus you have a **contract** between x and y.
- Since interface is the only dependency between x and y, the systems are not tightly coupled. Thus a change inside layer x wont affect y.



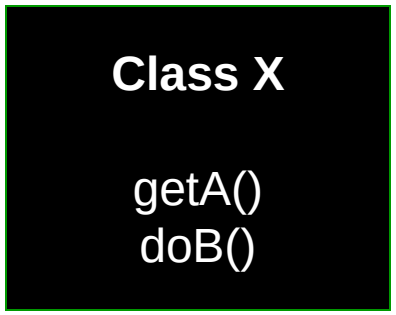
- data exchange can involve complex procedures.
- Divide the problem into sub-tasks
- implemented separately in layers in stack
  - each layer provides functions needed by higher layers to achieve communications.
  - In turn, it uses functions provided by layers below
- peer layers communicate with a protocol

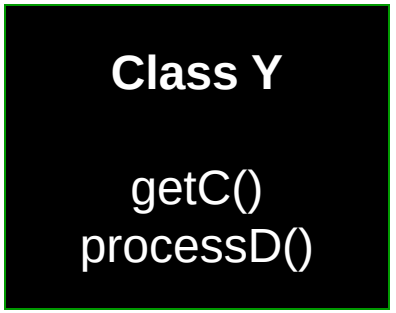


Stallings, W – Data and Computer Communications

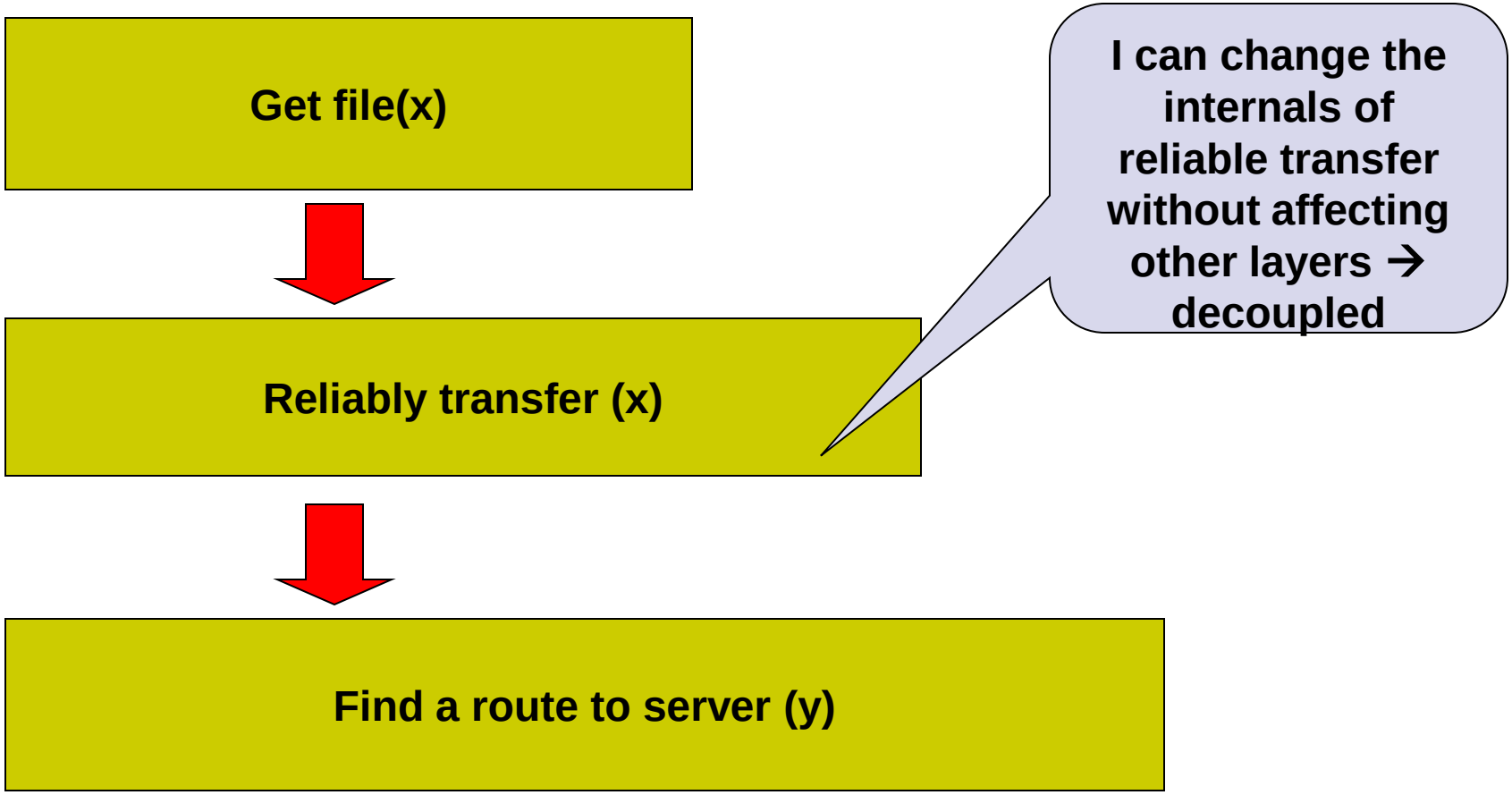
Interfaces – OO software

- In OO software, interfaces used extensively to reduce dependencies.
- Classes are black boxes, and only reveal their public interface.
- Insides of class can change, as long as interface is respected.
- This is just tip of the iceberg in terms of power of OO in large systems.
- Interfaces truly appreciated in large systems.

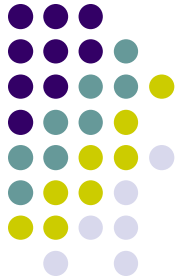




Simplified network stack



Layering – divide and conquer

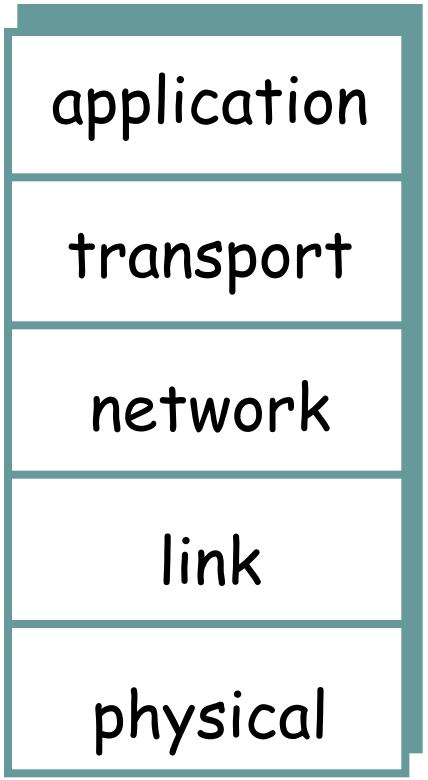


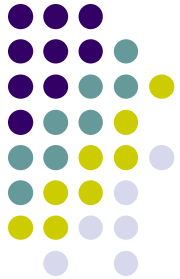
- Simplify the problem by splitting responsibilities
- Layer 1 – responsible for reliable transfer
- Layer 2 – responsible for finding a route
- Layer 3 – responsible for specifics of the network media e.g. copper, wireless, fibre

- Does Layer 2 care about reliable transfer ?
- Does Layer 1 care about the specifics of the medium ?

Internet protocol stack

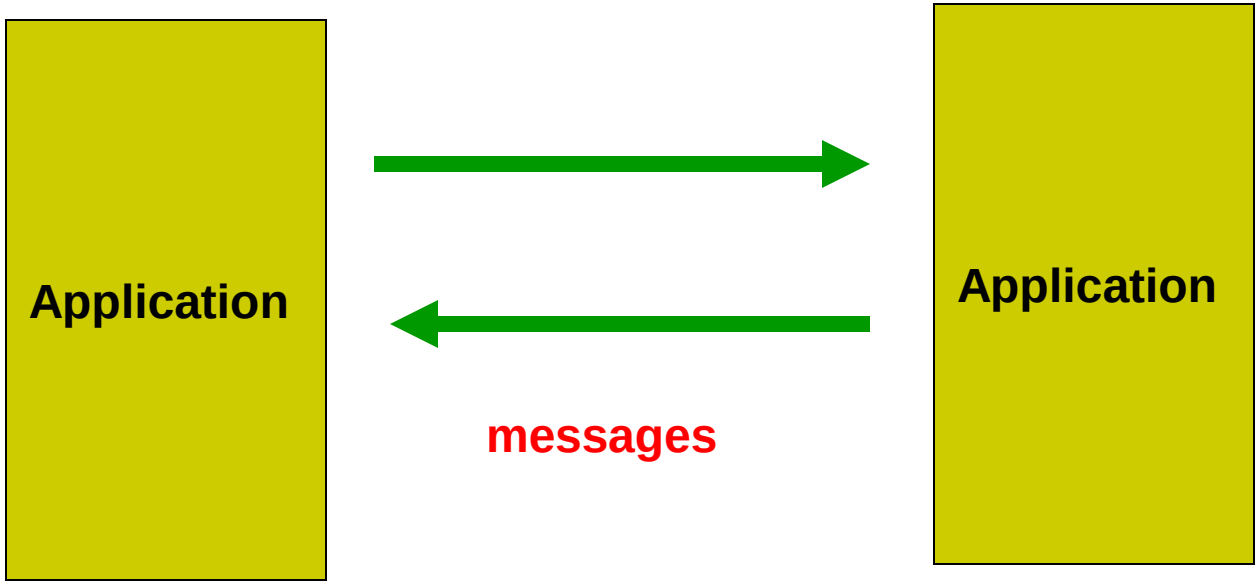
- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits “on the wire”

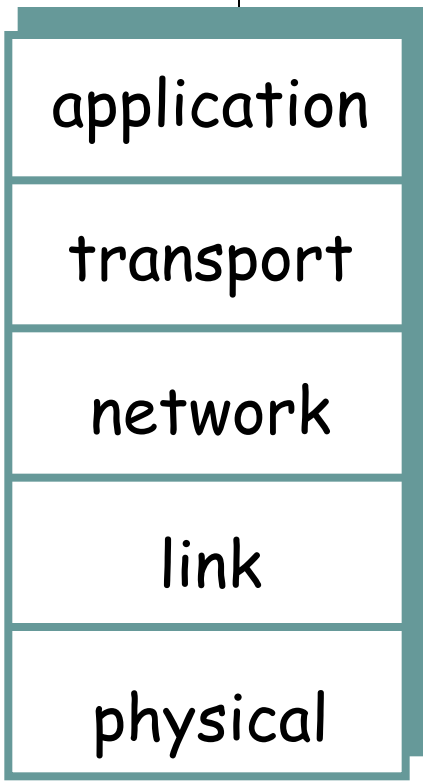




- Application Layer:
  - Where applications and associated protocols reside
    - http – responsible for request and transfer of web documents
    - ftp – transfer of files between 2 end systems.
- Packets exchanged by applications are called **messages**.

Application peers communicating

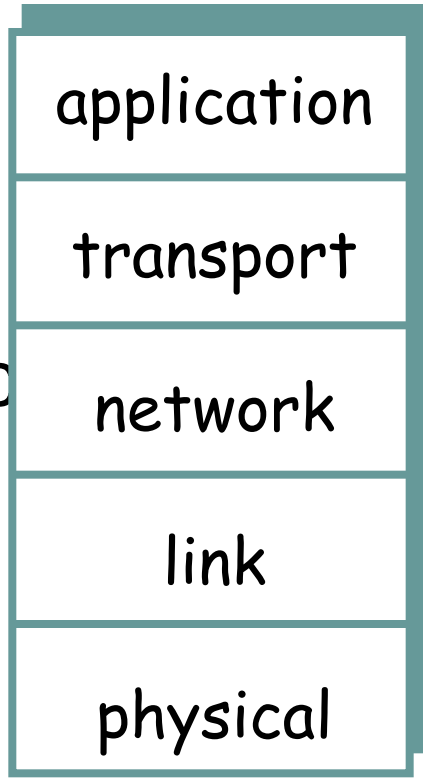


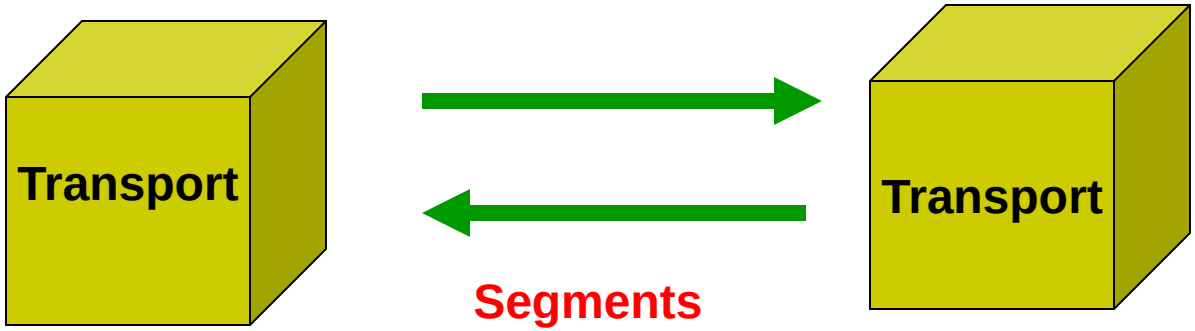


Application considers that it is communicating directly with its peer.

Transport layer

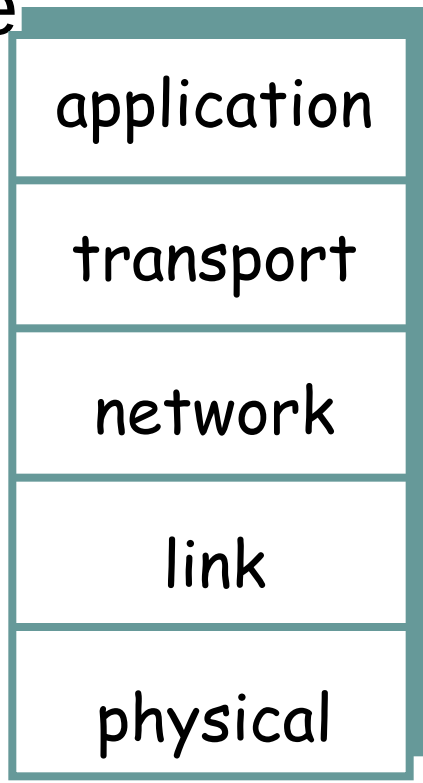
- Transports application layer messages.
- Provides guaranteed delivery service (TCP).
- Provides a connection oriented servic to application (TCP).
- Packet is called a **segment**.

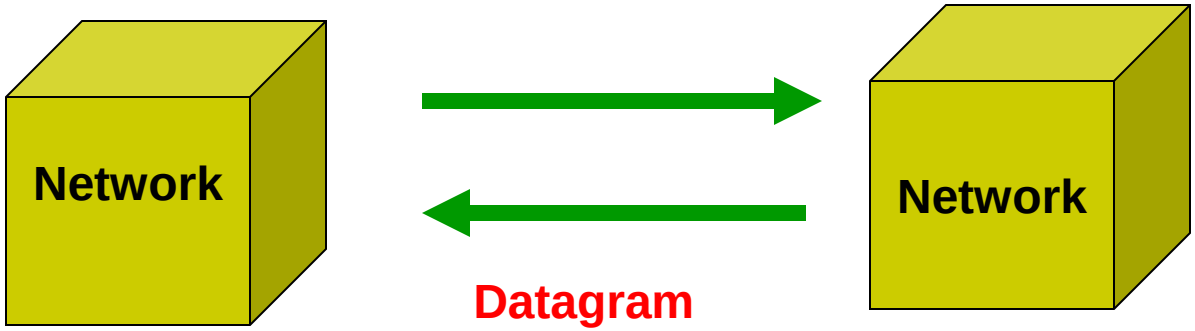




Network layer (IP)

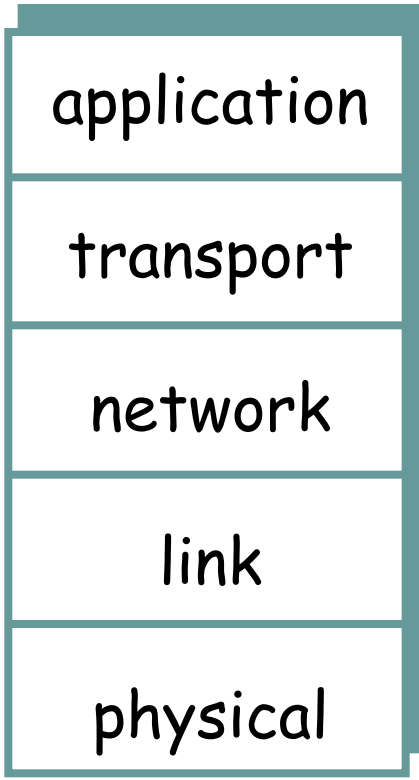
- Move network layer packets from one host to the next.
- For inter host transfer will have to traverse a number of networks.
- Packet is called a **datagram.**

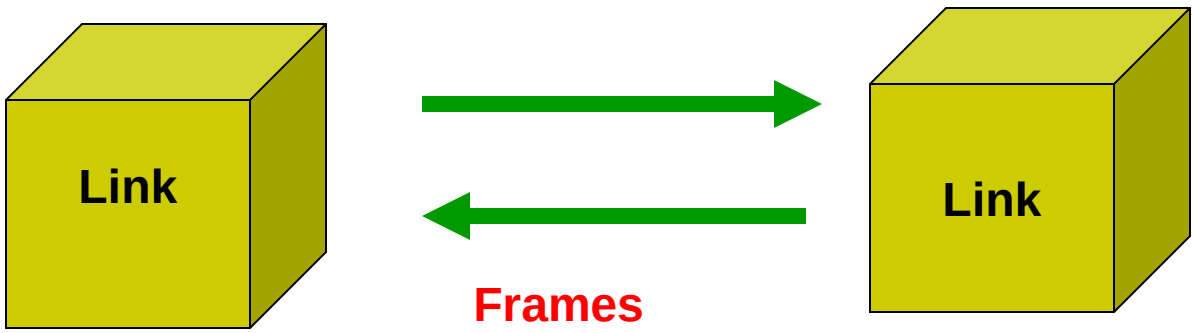




Link layer

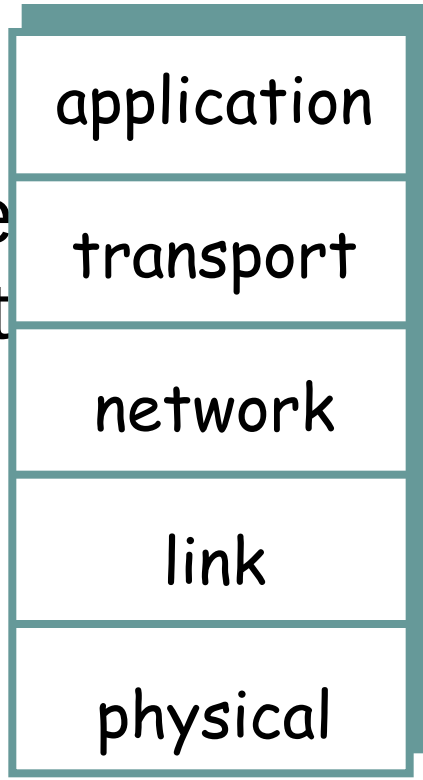
- Transport across a single link.
- For end to end routing, a packet will traverse a number of links e.g. coppe fibre, wireless.
- Packet is called a **frame**.

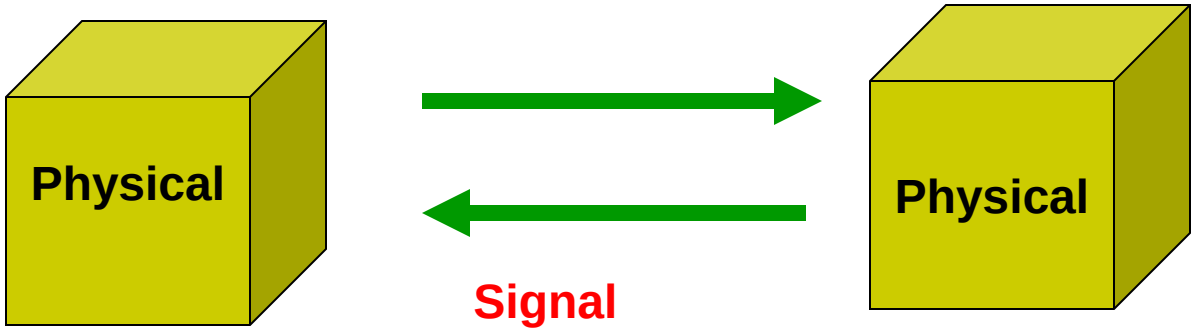




Physical layer

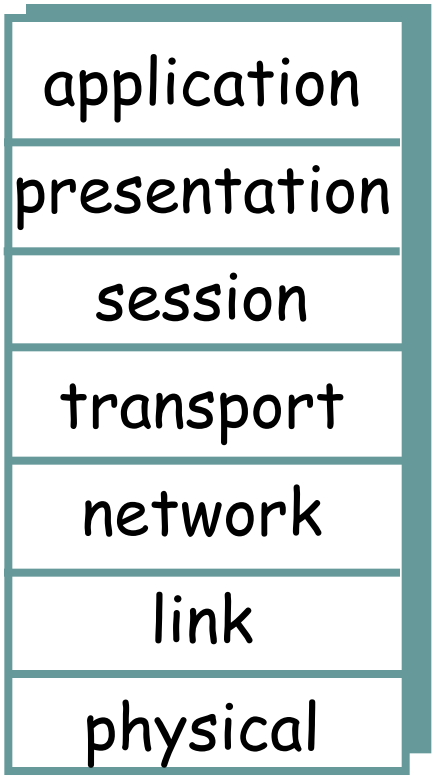
- Transports of physical bits (signal) across medium
- Ethernet (LAN) protocol has a numbe of physical layer protocols dependent on the medium e.g. copper, fibre.



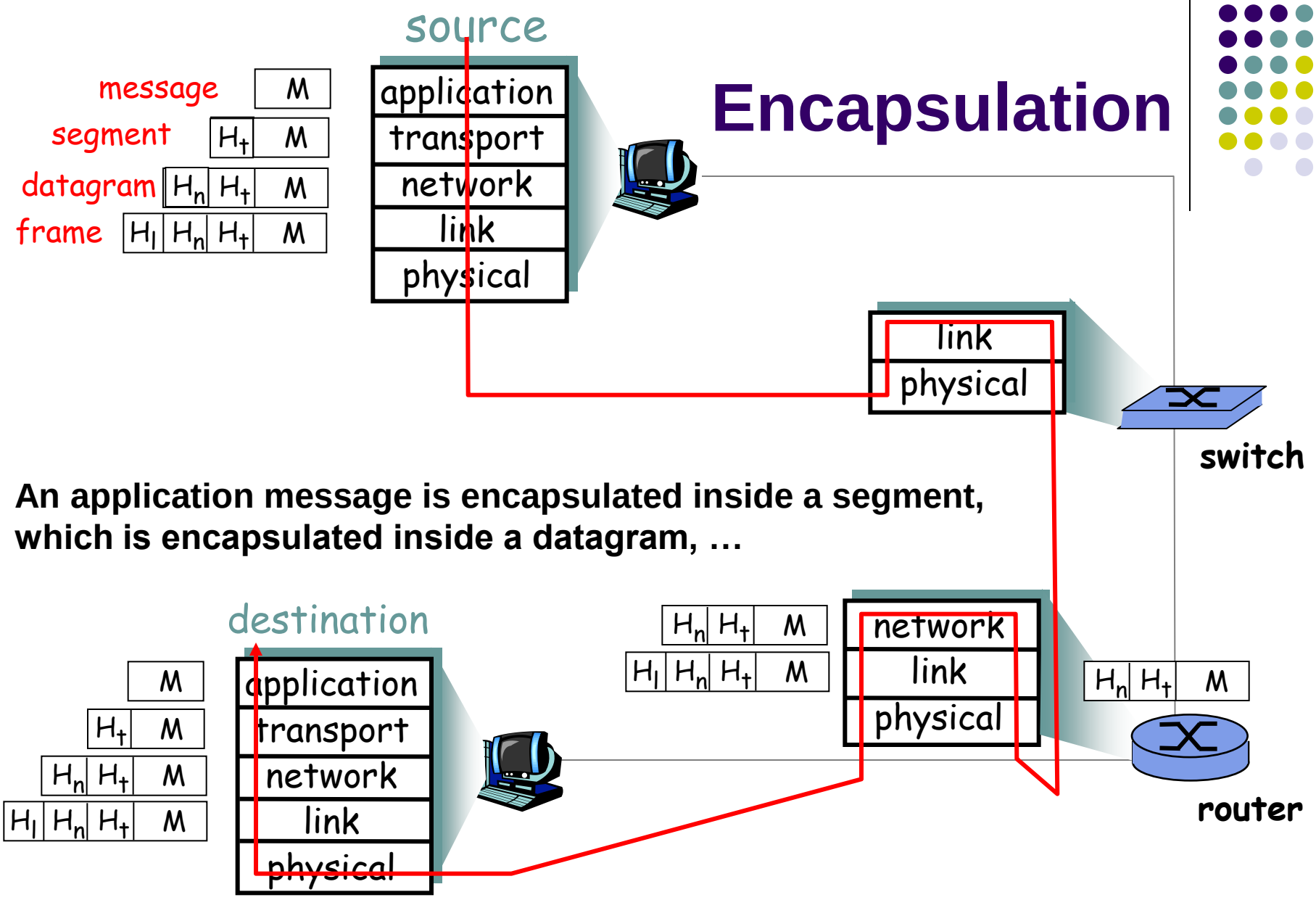


ISO/OSI reference model

- 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.

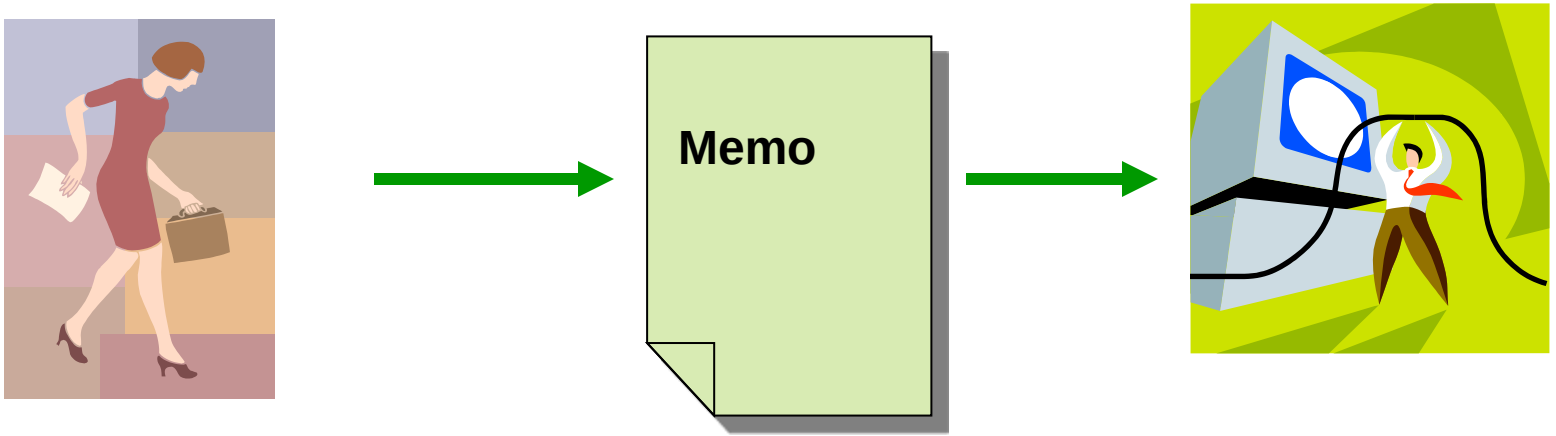


Encapsulation

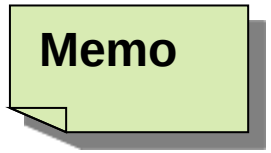


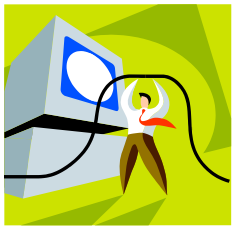
A human analogy

- Sending a corporate memo from one branch to another, using public postal system.
- Alice in one branch office wants to send memo to Bob in another branch.

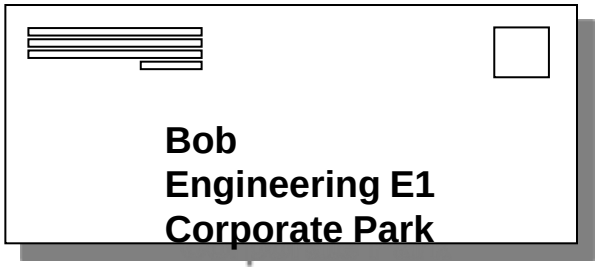


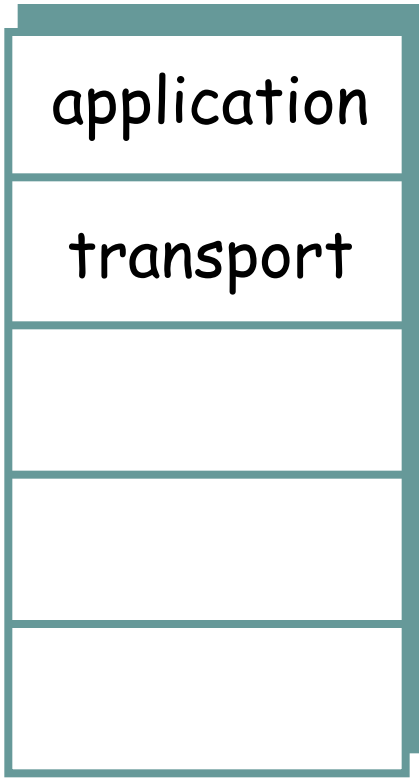
Application to transport





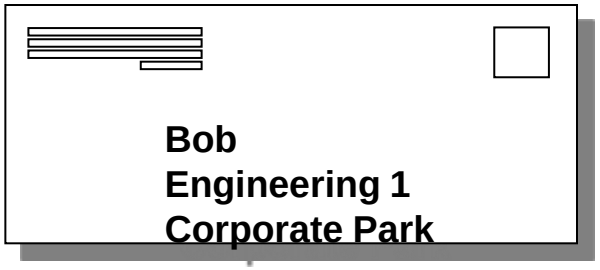
Inter office envelope



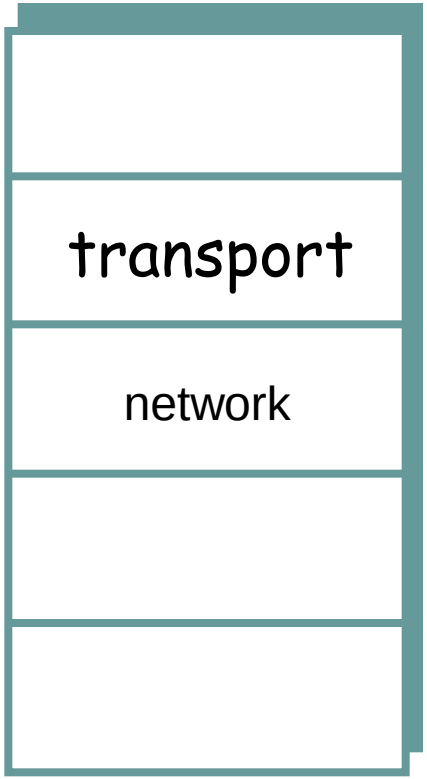


Transport to Network

Inter office envelope



Postal envelope



Link layer

- Video – Warriors of the net
- http://www.youtube.com/watch?v=Ve7_4ot-Dzs