Computer Networks

Protocol Architecture and TCP/IP

Amitangshu Pal
Computer Science and Engineering
IIT Kanpur

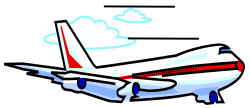
Protocol "layers" and reference models

Networks are complex, with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question: is there any hope of *organizing* structure of network?

Example: organization of air travel



end-to-end transfer of person plus baggage

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

How would you *define/discuss* the *system* of airline travel?

Example: organization of air travel

layers: each layer implements a service

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

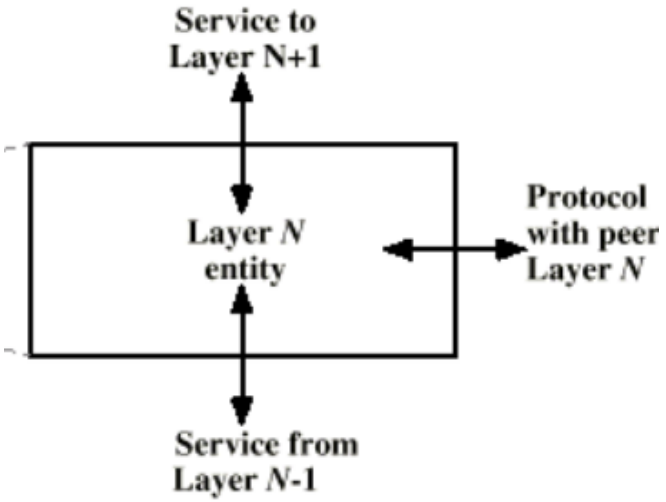
Why layering?

Approach to designing/discussing complex systems:

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

Need for a Protocol Architecture

- Data communications is complex
- Approach:
  - Break the communication tasks into subtasks.
  - Separate layers implement separate subtasks.
  - Layers are arranged in vertical tasks
    - Layer N uses service of layer N-1
    - Layer N provides service to layer N+1
  - Peer layers communicate with a protocol



What Is a Protocol?

- Set of rules that two (or more) peers obey in order to communicate.
- Syntax: Format and types of data blocks or messages.
- Procedures: Set of rules that the peers must follow.
- In order for two devices to communicate → they both needs to understand the same protocol
- Standard: Agreed-upon rules or protocols

Protocol Models

- OSI model: Open System Interconnection
  - Developed by International Standard Organization (ISO)
  - 7 layers

- TCP/IP model:
  - Developed by DARPA for first generation packet switched networks (ARPANET)
  - Used by global Internet

TCP/IP Layers

User Processes


Operating Systems

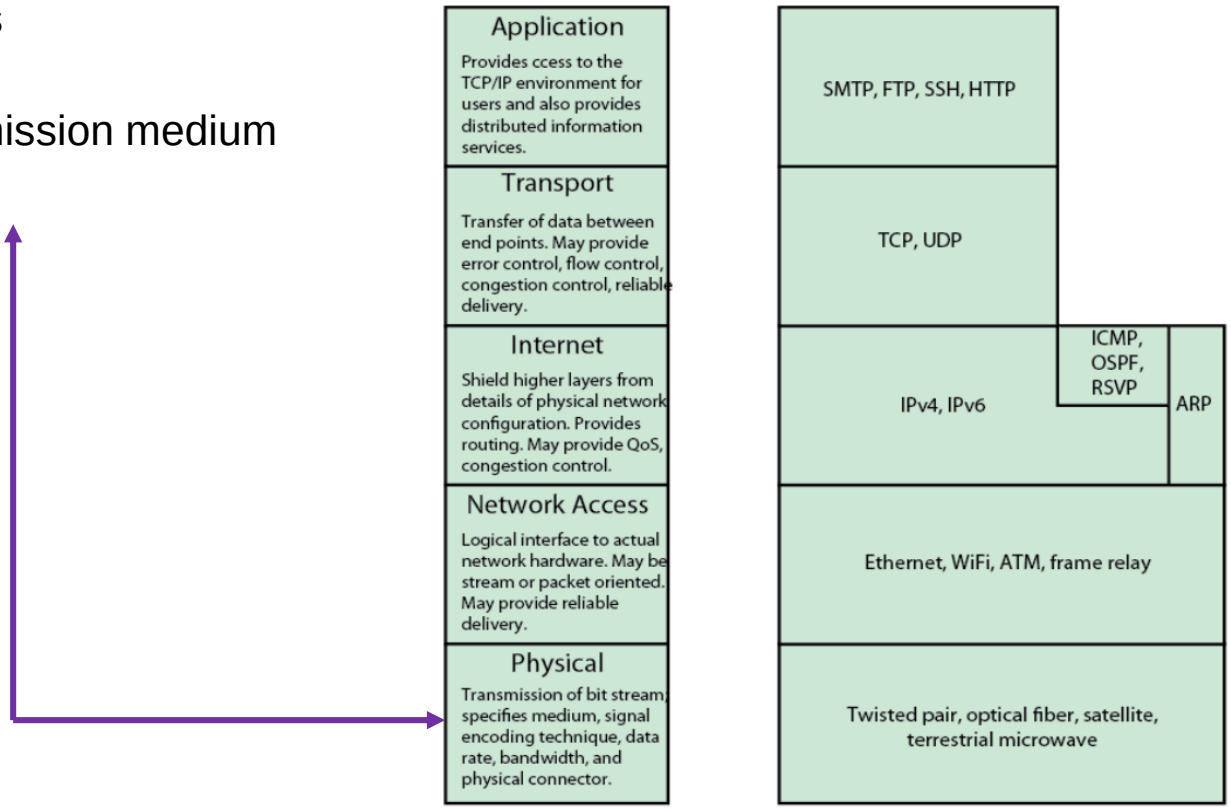


Network Interface Cards


Application Provides ccess to the TCP/IP environment for users and also provides distributed information services.	SMTP, FTP, SSH, HTTP
Transport Transfer of data between end points. May provide error control, flow control, congestion control, reliable delivery.	TCP, UDP
Internet Shield higher layers from details of physical network configuration. Provides routing. May provide QoS, congestion control.	IPv4, IPv6 / ICMP, OSPF, RSVP / ARP
Network Access Logical interface to actual network hardware. May be stream or packet oriented. May provide reliable delivery.	Ethernet, WiFi, ATM, frame relay
Physical Transmission of bit stream, specifies medium, signal encoding technique, data rate, bandwidth, and physical connector.	Twisted pair, optical fiber, satellite, terrestrial microwave

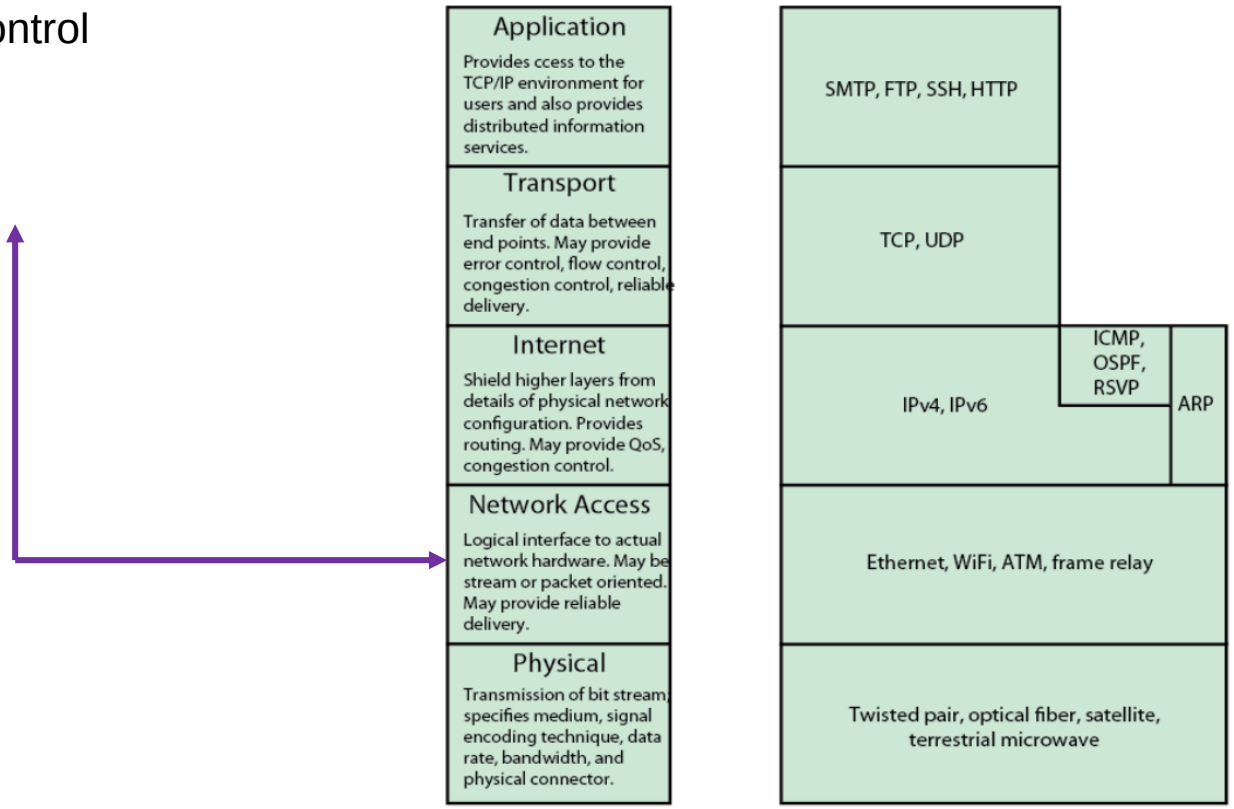
Physical Layer

- Physical layer:
  - Covers the physical interface between a data transmission device (workstation, computer) and a transmission medium or network
  - Specify the characters of the transmission medium
  - The nature of the signals
  - The data rate
  - Characteristics of transmission medium



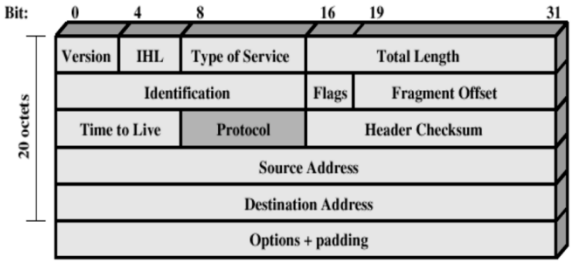
Network Access Layer

- Network Access layer:
  - Transmission of data over the link to which the device is attached
  - Allows layers above to ignore the details of the links
  - May provide reliable delivery
  - Flow control and error control
  - Sometimes called:
    - Data Link Layer
    - MAC Layer
    - Link Layer
    - Hardware Layer

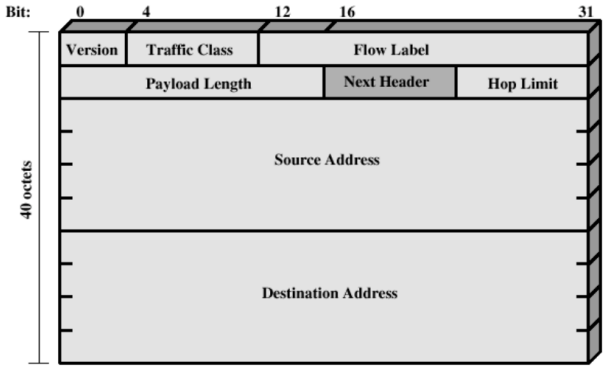


Internet Layer

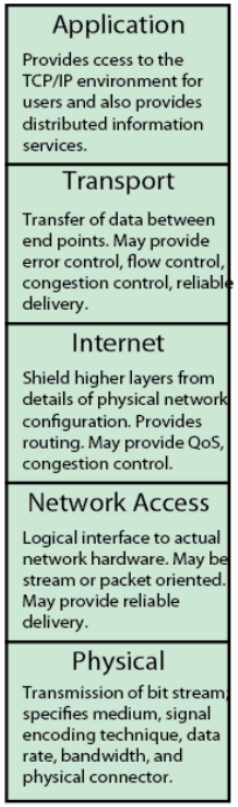
- Internet layer:
  - Provide Routing function across multiple networks
  - Uses Internet Protocol (IP) to provide routing functions
  - May provide QoS, congestion control etc
  - Router:
    - A processor that relays packets from one network to other

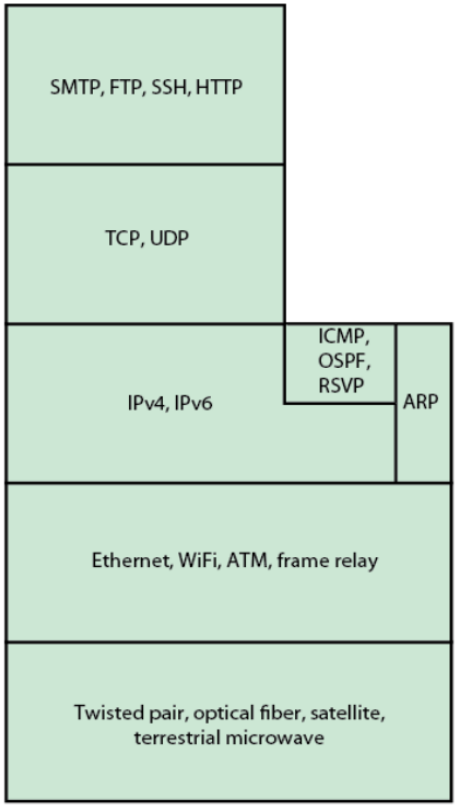


- IPv4 header:
  - 32 bit source and destination address



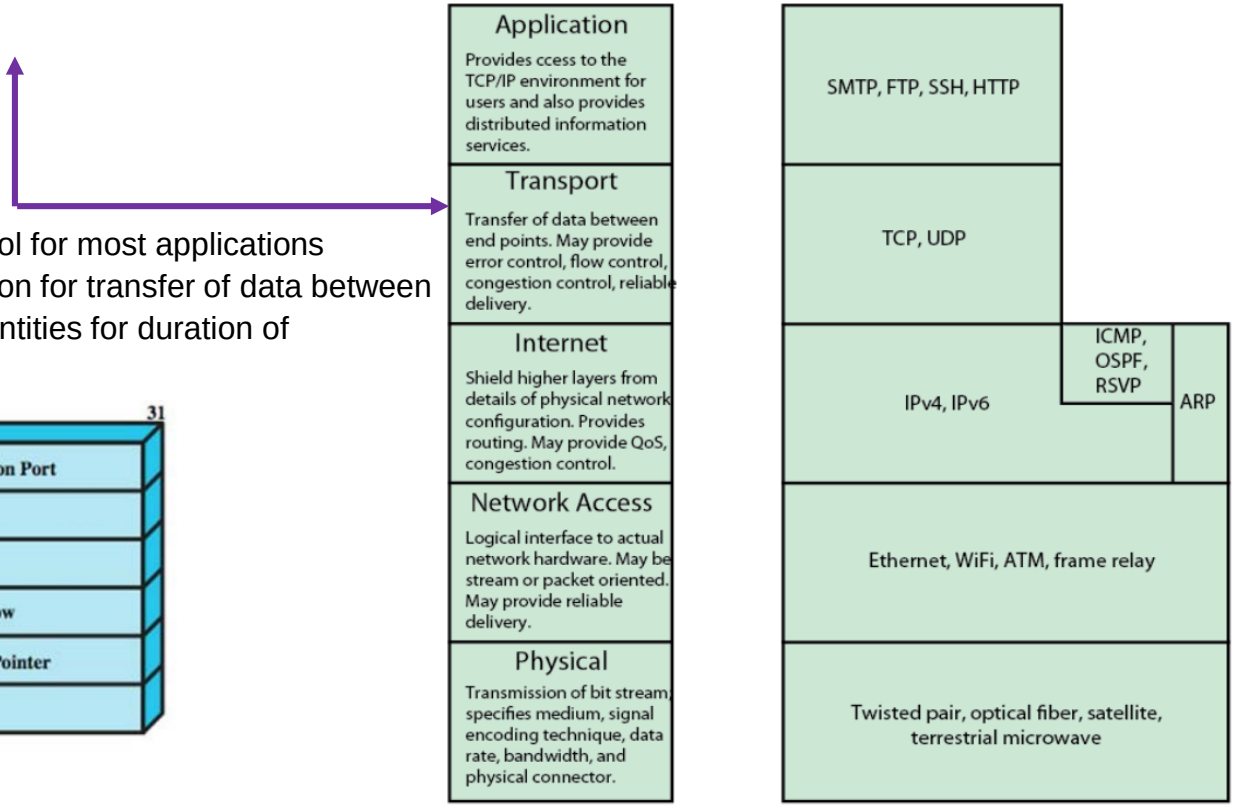
- IPv6 header:
  - 128 bit source and destination address



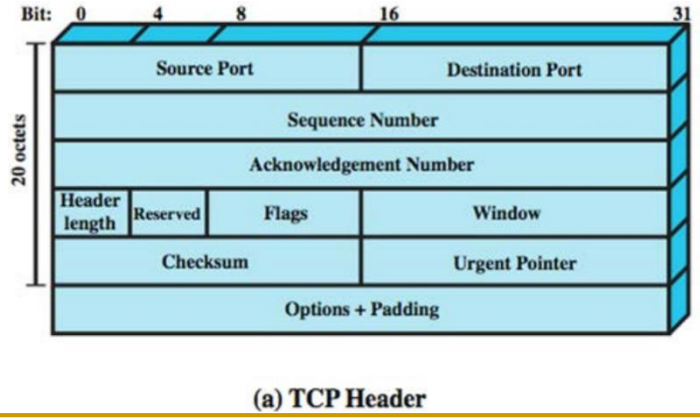


Transport Layer

- Transport layer:
  - Provide End-to-end (reliable, i.e. No error, In order) delivery of data
  - TCP, UDP
  - Also known as host-to-host layer



- TCP:
  - TCP is the transport layer protocol for most applications
  - TCP provides a reliable connection for transfer of data between
  - TCP tracks segments between entities for duration of each connection

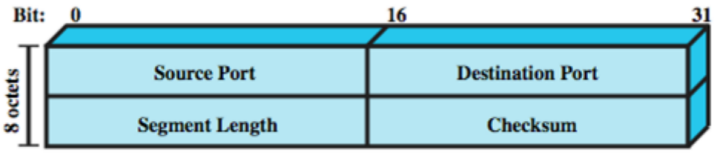


(a) TCP Header

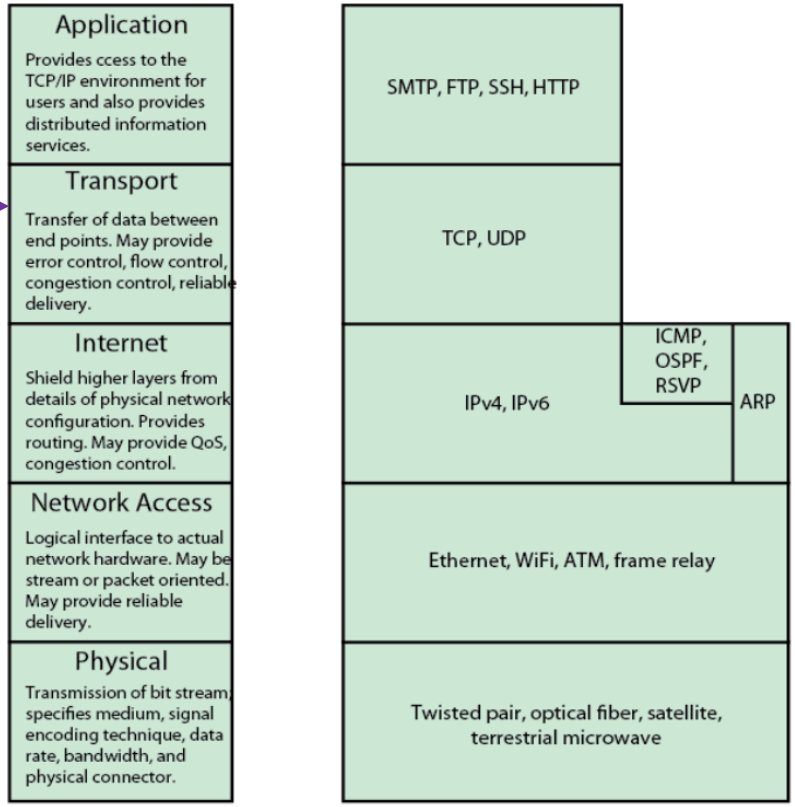
Transport Layer

- Transport layer:
  - Provide End-to-end reliable (No error, In order) delivery of data
  - TCP, UDP
  - Also known as host-to-host layer

- UDP:
  - Alternative to TCP
  - Does not guarantee delivery, preservation of sequence, or protection against duplication
  - Adds port addressing capability to IP
  - Used with Simple Network Management Protocol (SNMP)

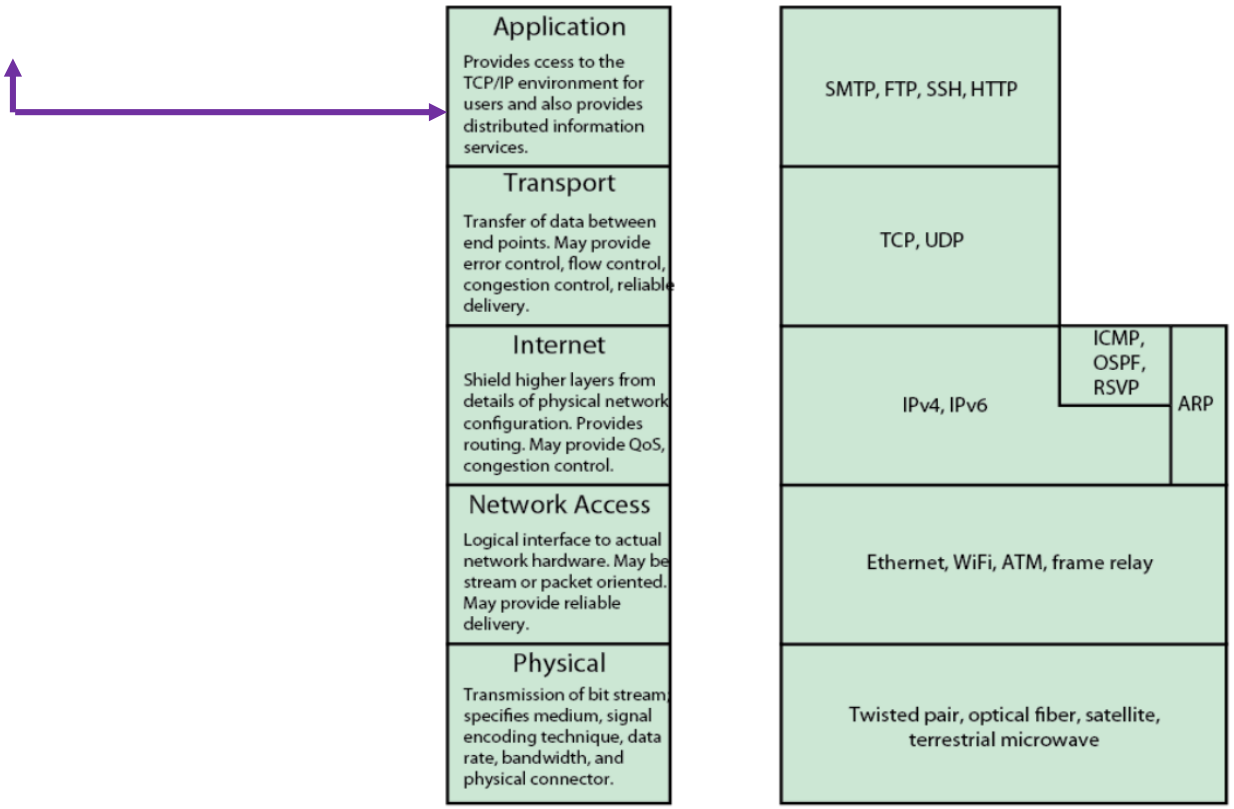


(b) UDP Header



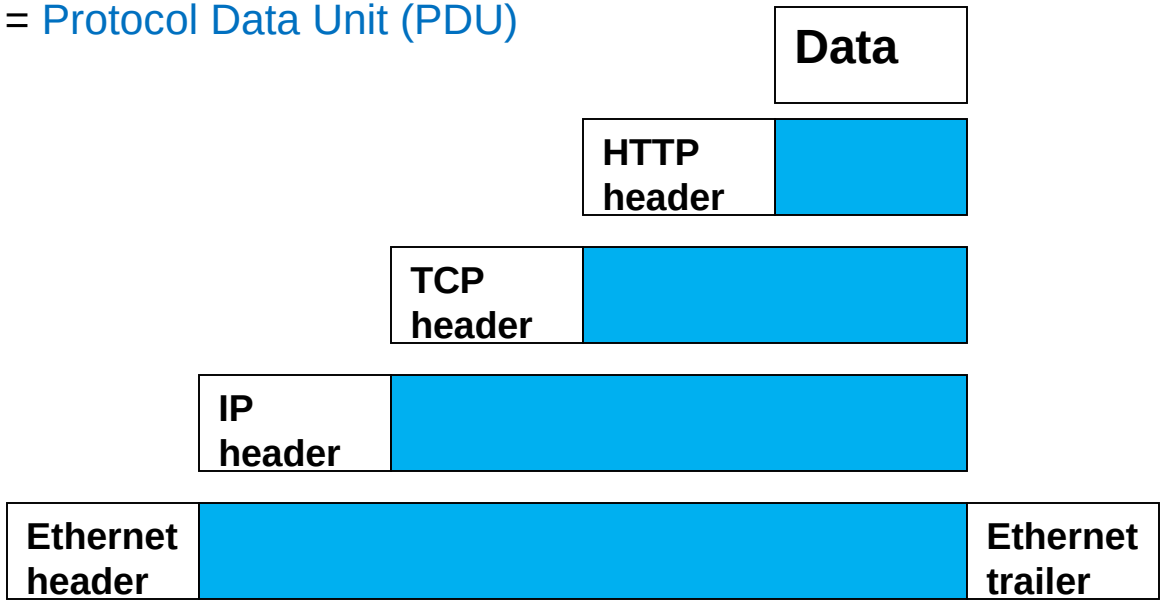
Application Layer

- Application layer:
  - Support user application
    - Web browsing: HTTP
    - File Transfer: FTP
    - Email: SMTP
    - Secure remote login: SSH



Protocol Data Unit

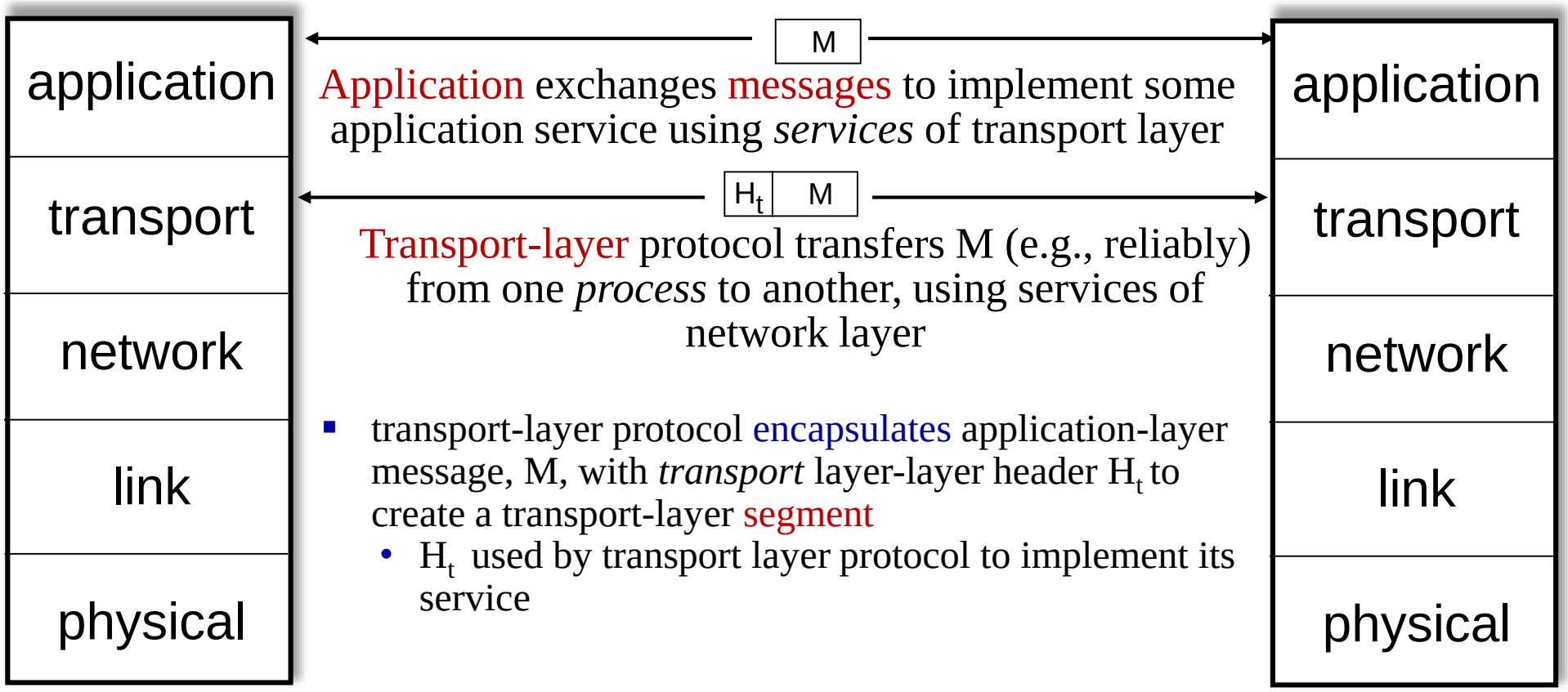
- Headers are added to carry the control informations, referred to as encapsulation
  - Control informations are source/destination address, sequence number, error-detection code
- Header + Data = Protocol Data Unit (PDU)



Addressing

- Internet Address: Each host in a sub-network must has an unique Global Internet Address
  - IP address: IPv4, IPv6
- Port Address: Each process with a host must have an address (known as ports) that is unique within the host
- Another address is also used in this context
- Hardware Address:
  - Ethernet LAN: 48 bit address

Services, Layering and Encapsulation

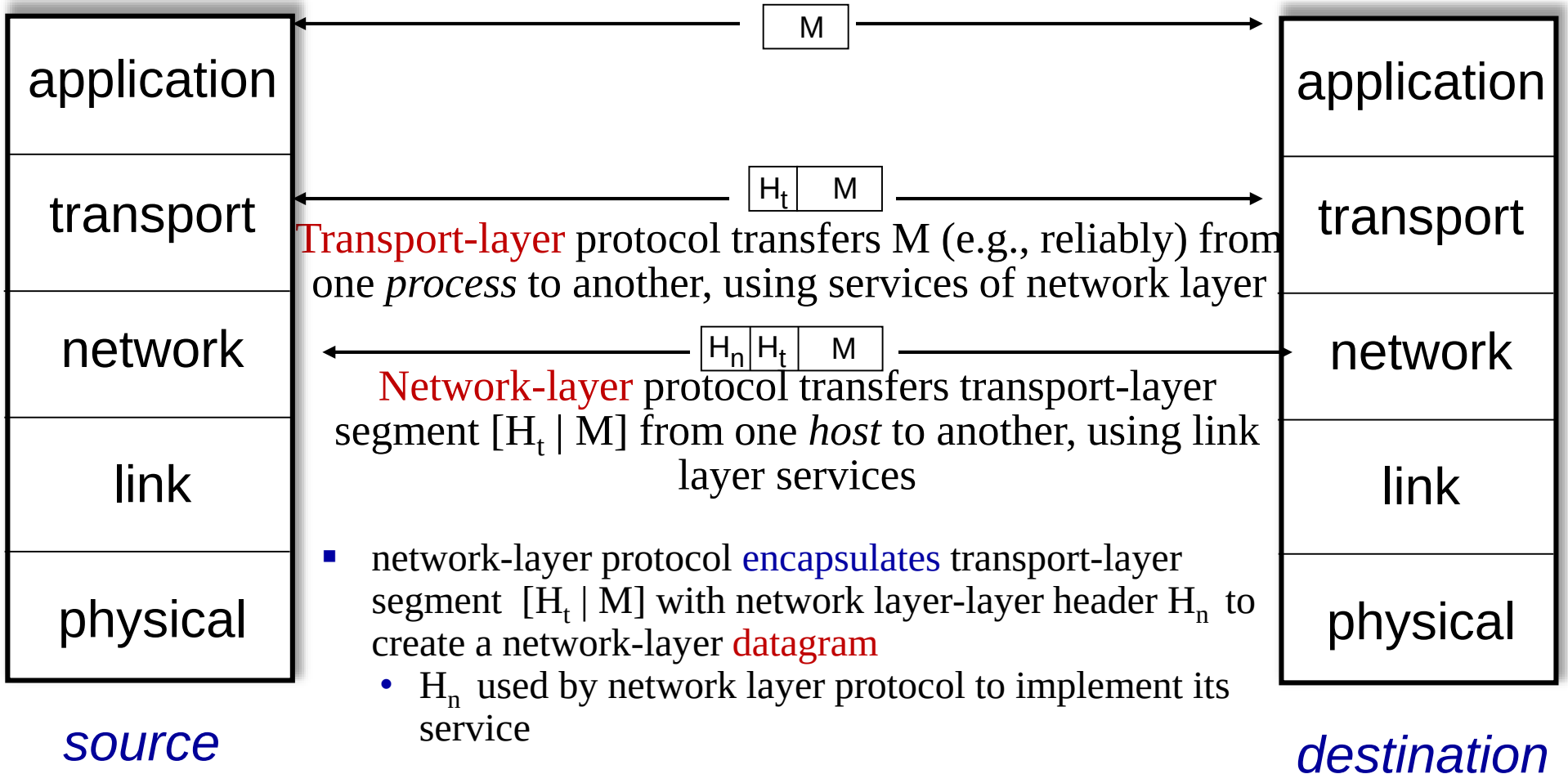


- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

source

destination

Services, Layering and Encapsulation

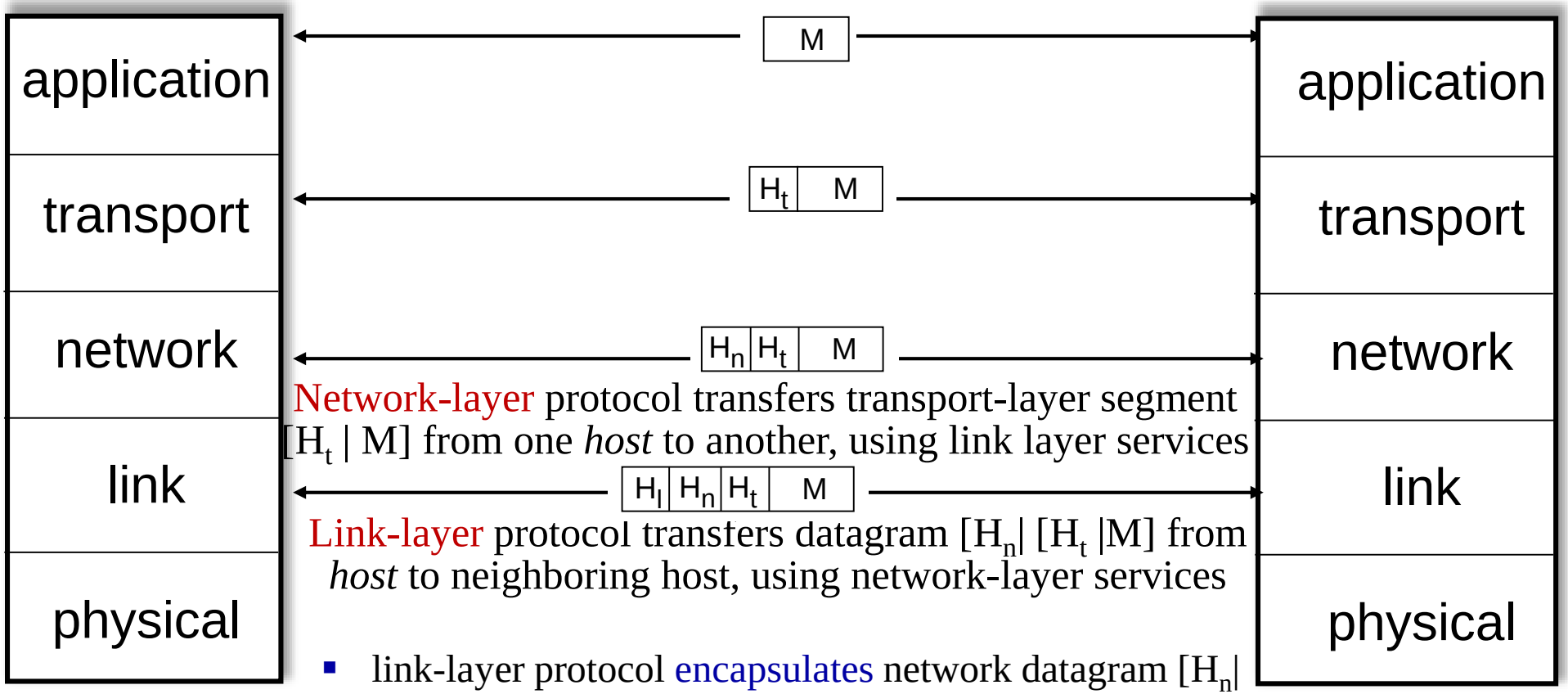


Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

- network-layer protocol encapsulates transport-layer segment [H_t | M] with network layer-layer header H_n to create a network-layer datagram
  - H_n used by network layer protocol to implement its service

Services, Layering and Encapsulation

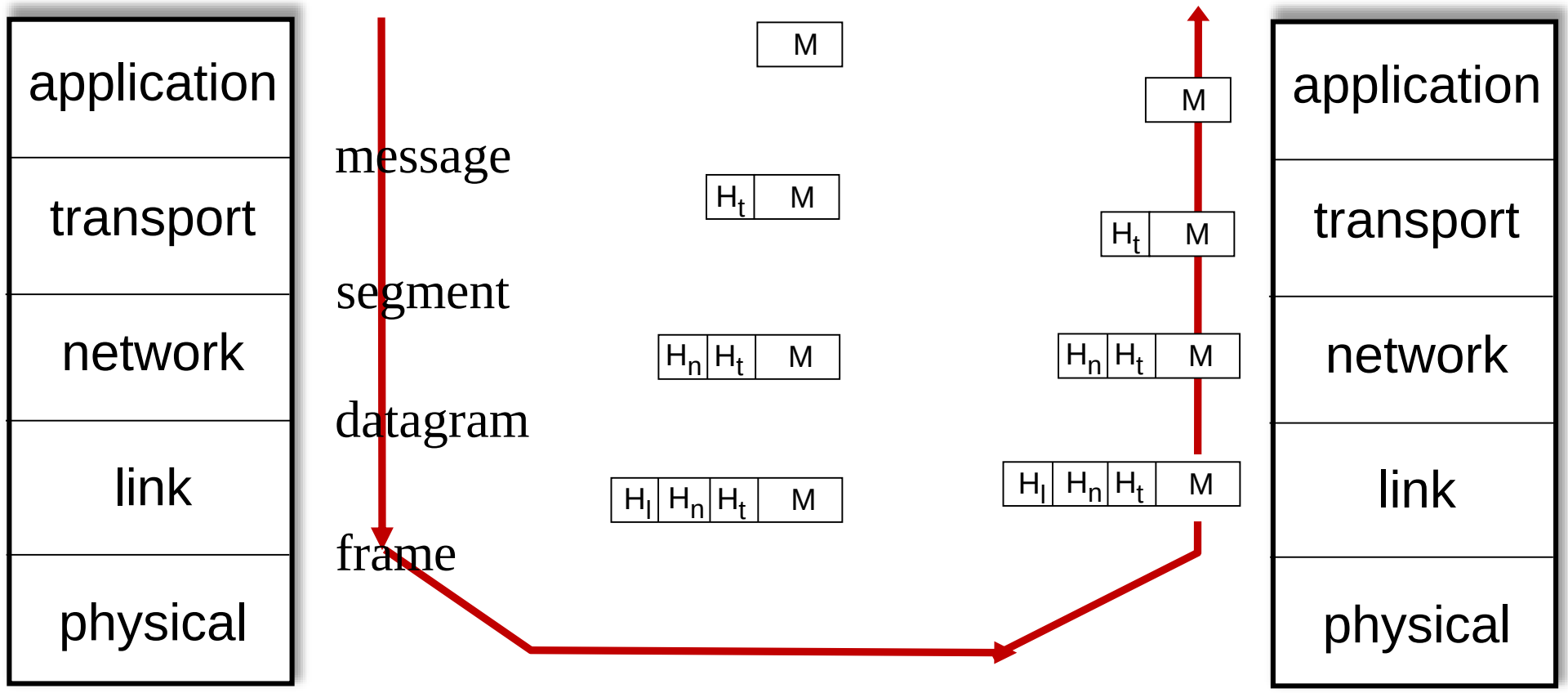


Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

Link-layer protocol transfers datagram [H_n| [H_t |M] from *host* to neighboring host, using network-layer services

- link-layer protocol encapsulates network datagram [H_n| [H_t |M], with link-layer header H_l to create a link-layer frame

Services, Layering and Encapsulation



Encapsulation: an end-end view