CprE 450/550x
Distributed Systems and Middleware

Basics of Computer Networks

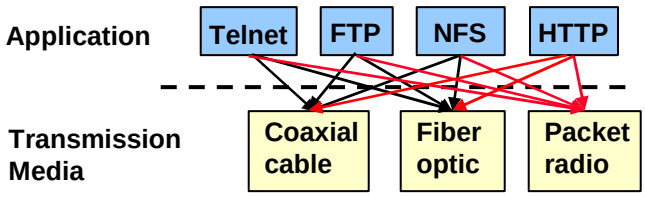
Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu
January 20, 2004

1.1. What is Layering?

- A technique to organize a network system into a succession of logically distinct entities, such that the service provided by one entity is solely based on the service provided by the previous (lower level) entity

1.2. Why Layering?

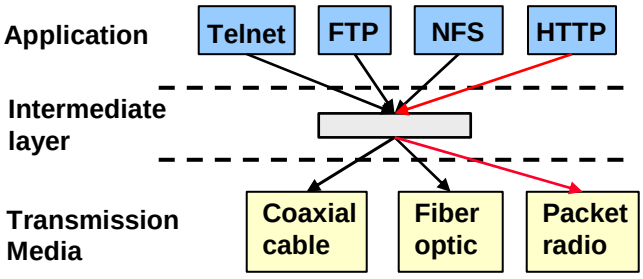
(FTP – File Transfer Protocol, NFS – Network File Transfer, HTTP – World Wide Web protocol)



- No layering: each new application has to be re-implemented for every network technology!

1.3. Why Layering?

- Solution: introduce an intermediate layer that provides a unique abstraction for various network technologies



1.5. Layering

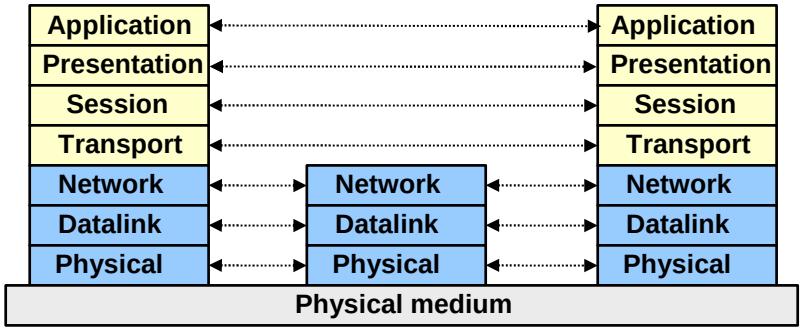
- Advantages
  - Modularity – protocols easier to manage and maintain
  - Abstract functionality –lower layers can be changed without affecting the upper layers
  - Reuse – upper layers can reuse the functionality provided by lower layers
- Disadvantages
  - Information hiding – inefficient implementations

1.6. ISO OSI Reference Model

- ISO – International Standard Organization
- OSI – Open System Interconnection
- Started in 1978; first standard in 1979
  - ARPANET started in 1969;
  - TCP/IP protocols (early versions) ready by 1974
- Goal: a general open standard
  - allow vendors to enter the market by using their own
    - implementation of protocols
    - application level protocols (distributed applications)

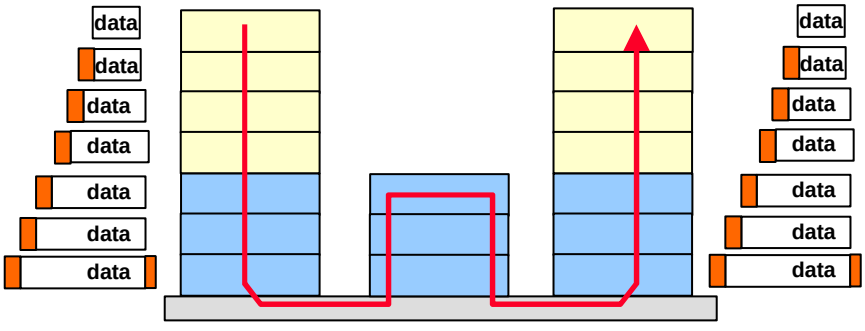
1.7. ISO OSI Reference Model

- Seven layers
  - Lower three layers are peer-to-peer
  - Next four layers are end-to-end



1.8. Encapsulation

- A layer can use only the service provided by the layer immediate below it
- Each layer may change and add a header to data packet



1.9. OSI Model Concepts

- Service – says what a layer does
- Interface – says how to access the service
- Protocol – says how is the service implemented
  - a set of rules and formats that govern the communication between two peers

1.10. Physical Layer (1)

- Service: move the information between two systems connected by a physical link
- Interface: specifies how to send a bit
- Protocols: coding scheme used to represent a bit, voltage levels, duration of a bit

- Examples: coaxial cable, optical fiber links; transmitters, receivers

1.11. Datalink Layer (2)

- Service:
  - framing, i.e., attach frame separators
  - send data frames between peers
  - others:
    - arbitrate the access to common physical media
    - ensure reliable transmission
    - provide flow control
- Interface: send a data unit (packet) to a machine connected to the same physical media
- Protocols: layer addresses, implement Medium Access Control (MAC) (e.g., CSMA/CD)…

1.12. Network Layer (3)

- Service:
  - deliver a packet to specified destination
  - perform segmentation/reassemble
  - others:
    - packet scheduling
    - buffer management
- Interface: send a packet to a specified destination
- Protocols: define global unique addresses; construct routing tables

1.13. Transport Layer (4)

- Services:
  - provide an error-free and flow-controlled end-to-end connection
  - multiplex multiple transport connections to one network connection
  - split one transport connection in multiple network connections
- Interface: send a packet to specify destination
- Protocols: implement reliability and flow control
- Examples: TCP and UDP

1.14. Session Layer (5)

- Service:
  - full-duplex
  - access management, e.g., token control
  - synchronization, e.g., provide check points for long transfers
- Interface: depends on service
- Protocols: token management; insert checkpoints, implement roll-back functions

1.15. Presentation Layer (6)

- Service: convert data between various representations
- Interface: depends on service
- Protocol: define data formats, and rules to convert from one format to another

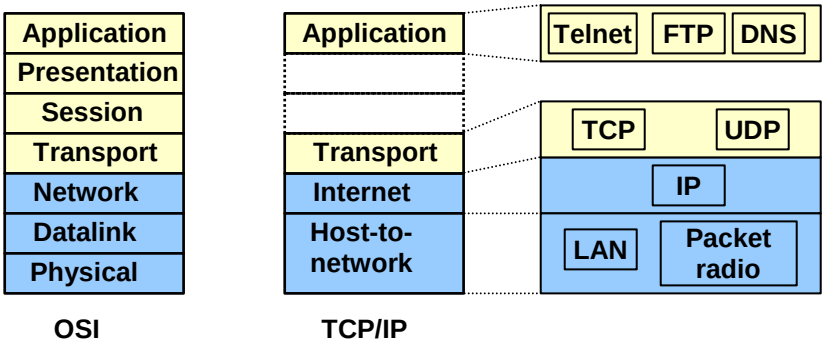
1.16. Application Layer (7)

- Service: any service provided to the end user
- Interface: depends on the application
- Protocol: depends on the application

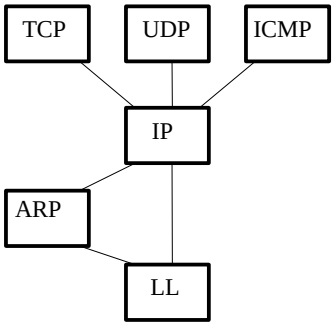
- Examples: FTP, Telnet, WWW browser

1.17. OSI *vs*. TCP/IP

- OSI: conceptually define services, interfaces, protocols
- Internet: provide a successful implementation



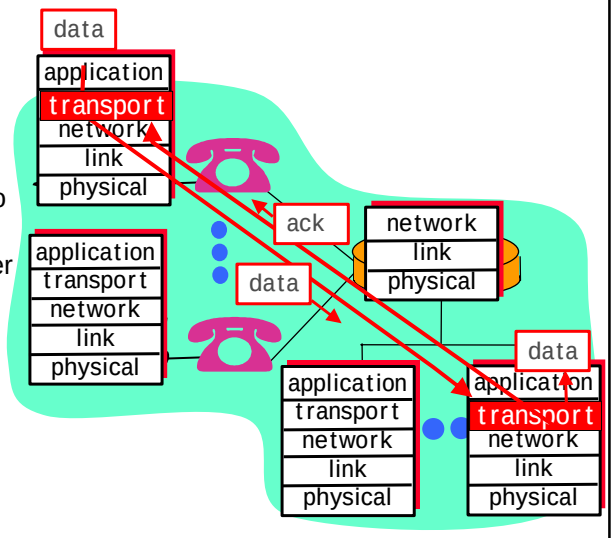
1.18. Protocol Graphs



TCP/IP Protocol Graph

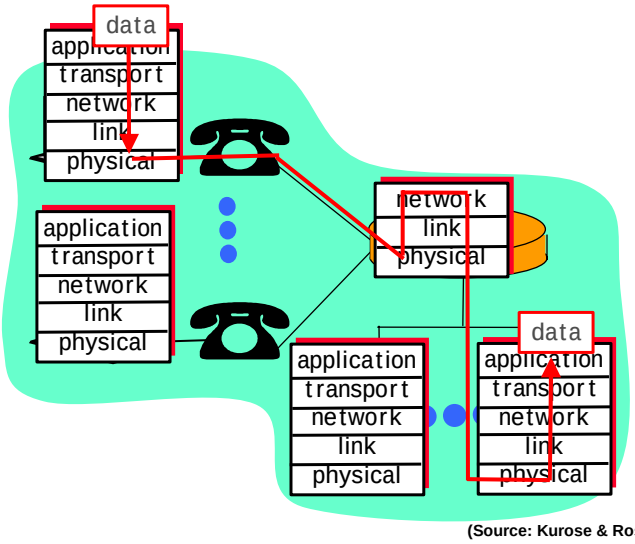
1.20. Example: Transport Protocol (Logical Communication)

- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



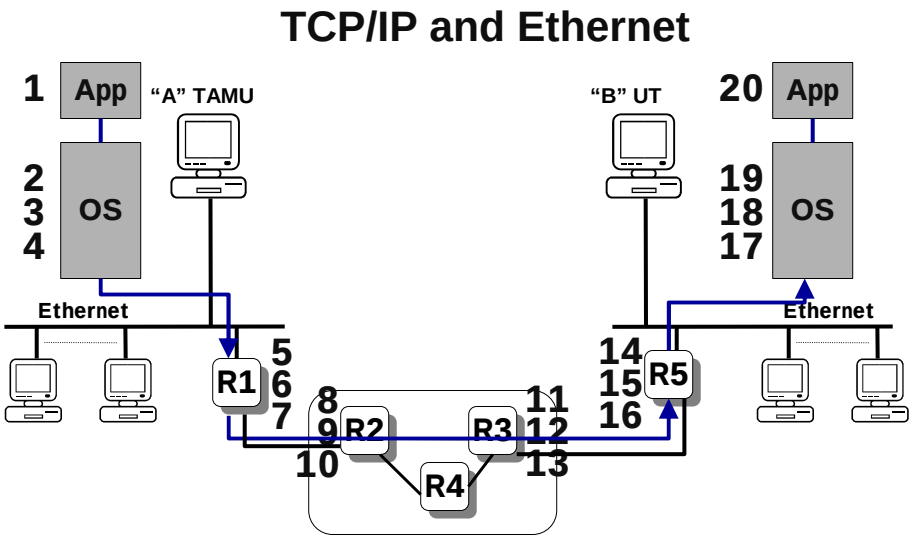
(Source: Kurose & Ross)

1.21. Example: Transport Protocol (Physical Communication)



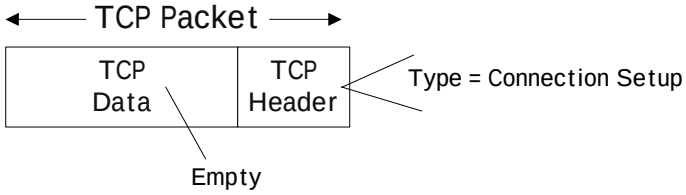
(Source: Kurose & Ross)

2. Example: FTP over the Internet



2.1. In the sending host

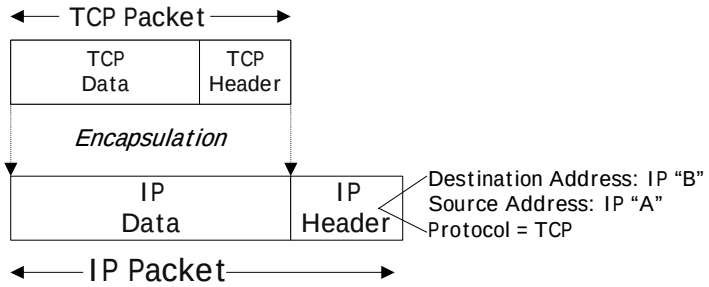
1. **Application-Programming Interface (API)**
   - Application requests TCP connection with "B"
2. **Transmission Control Protocol (TCP)**
   1. Creates TCP "Connection setup" packet
   2. TCP requests IP packet to be sent to "B"



2.2. In the sending host (2)

3. Internet Protocol (IP)

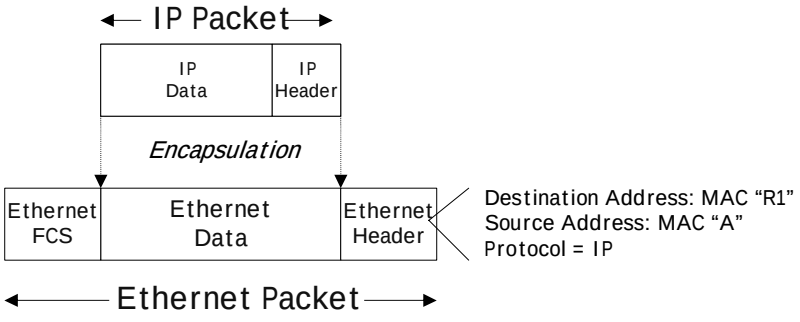
- Creates IP packet with correct addresses.
- IP requests packet to be sent to router.



2.3. In the sending host (3)

4. Link ("MAC" or Ethernet) Protocol

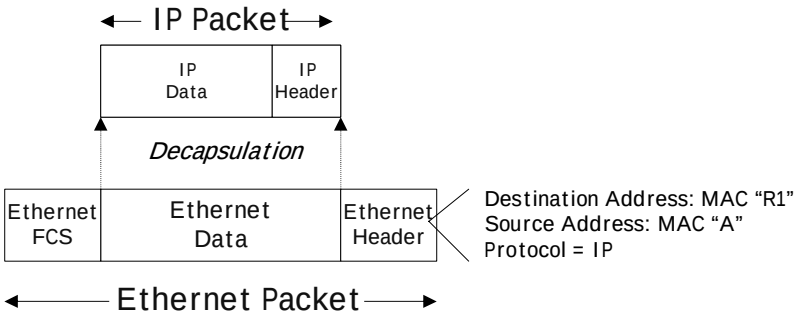
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



2.4. In Router R1

5. Link ("MAC" or Ethernet) Protocol

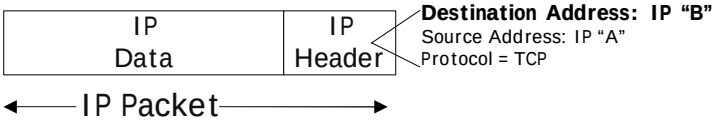
- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



2.5. In Router R1

6. Internet Protocol (IP)

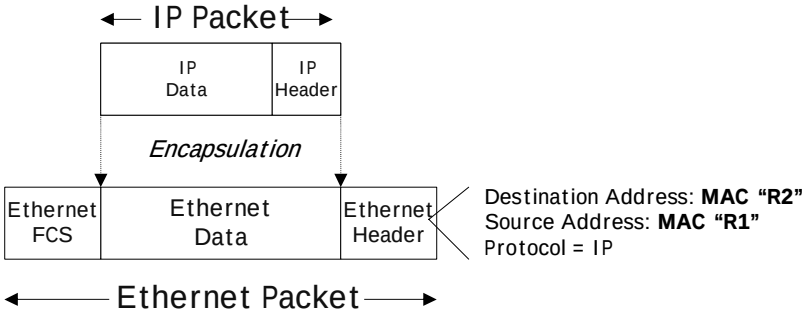
- Use IP destination address to decide where to send packet next ("next-hop routing").
- Request Link Protocol to transmit packet.



2.6. In Router R1

7. Link ("MAC" or Ethernet) Protocol

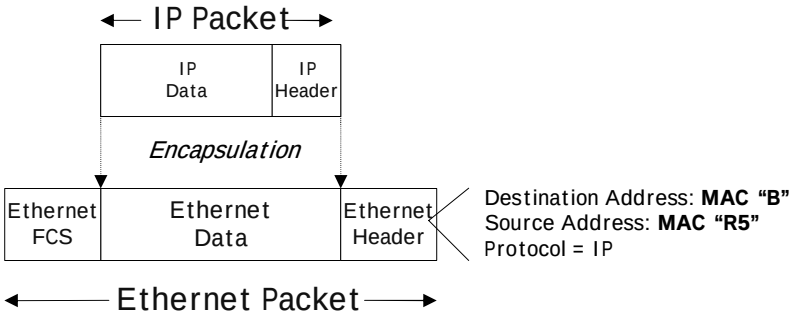
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



2.7. In Routers R2, R3, R5 (*Same operations as Router R1)*

16. Link ("MAC" or Ethernet) Protocol

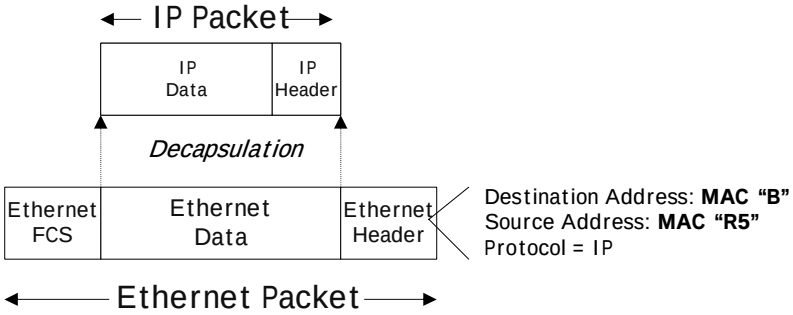
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



2.8. In the receiving host

17. Link ("MAC" or Ethernet) Protocol

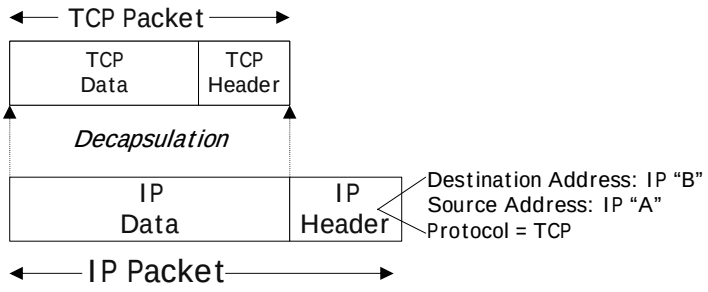
- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



2.9. In the receiving host (2)

18. Internet Protocol (IP)

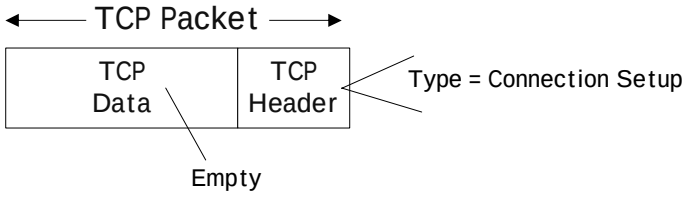
- Verify IP address.
- Extract/decapsulate TCP packet from IP packet.
- Pass TCP packet to TCP Protocol**.**



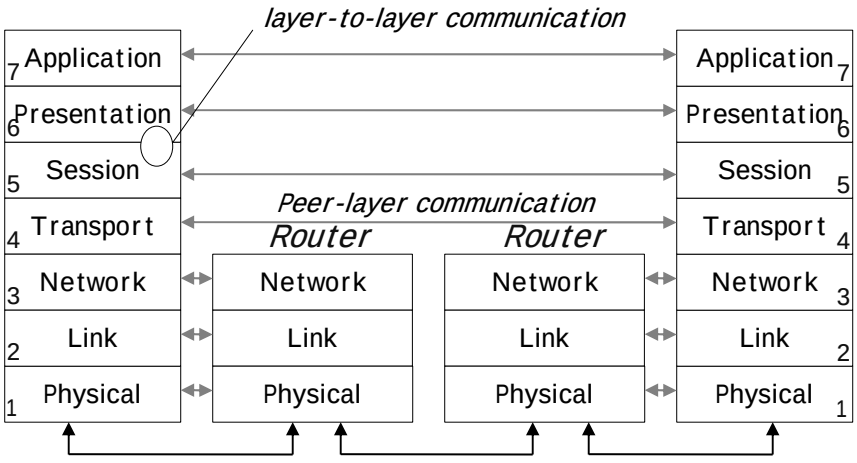
2.10. In the receiving host (3)

19. Transmission Control Protocol (TCP)
- Accepts TCP “Connection setup” packet
- Establishes connection by sending “Ack”.

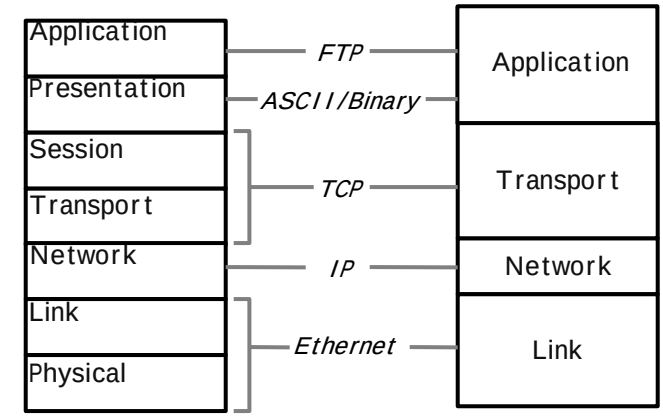
20. Application-Programming Interface (API)
- Application receives request for TCP connection with “A”.



2.11. Layering: The OSI Model



2.12. Layering: Our FTP Example



The 7-layer OSI Model

The 4-layer Internet model

Any Questions?

See you next time.