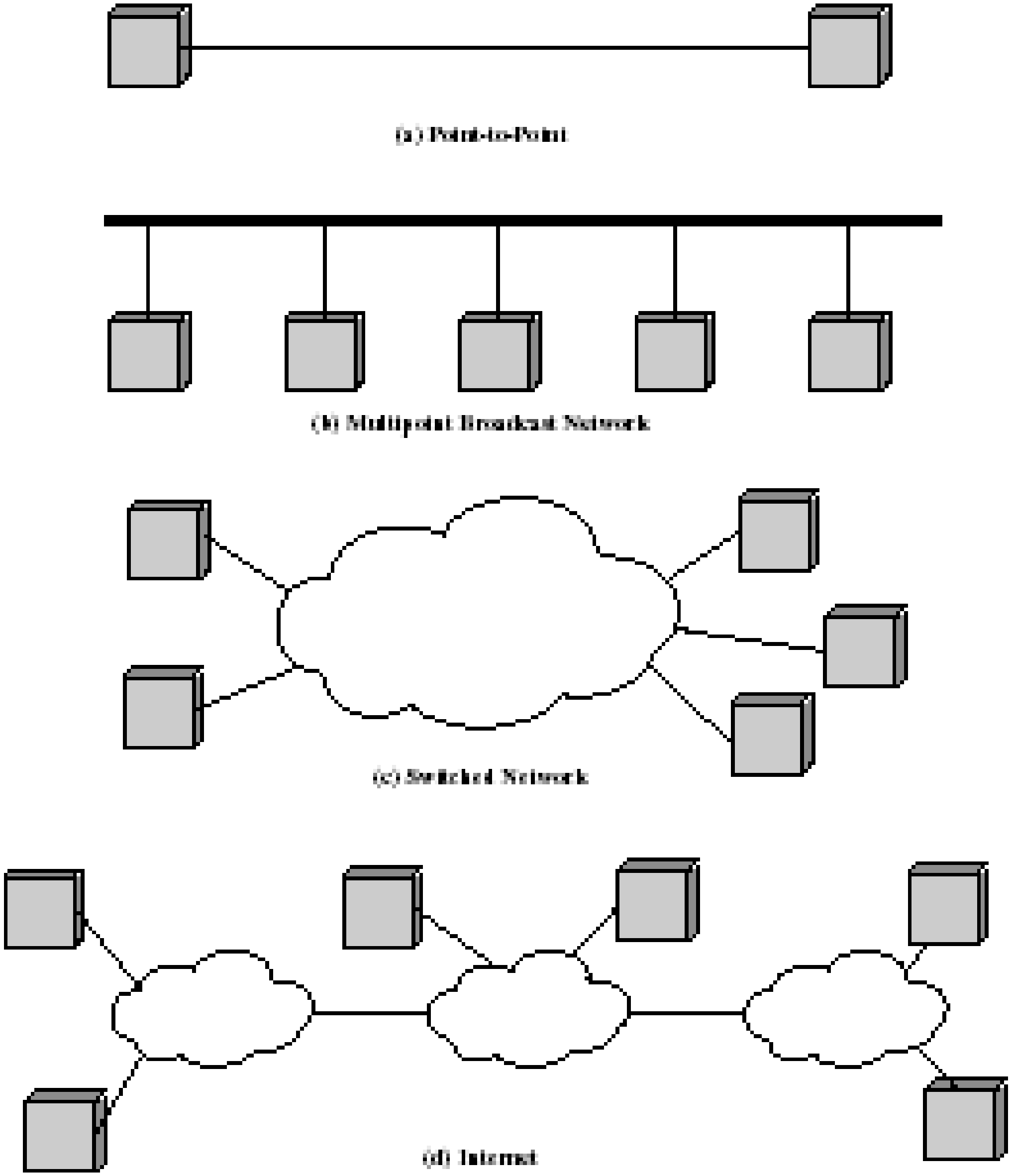
Data and Computer Communications

Protocols and Architecture

Characteristics

- Direct or indirect
- Monolithic or structured
- Symmetric or asymmetric
- Standard or nonstandard

Means of Communication



(a) Point-to-Point

(b) Multipoint Broadcast Network

(c) Switched Network

(d) Internet

Direct or Indirect

- Direct
  - Systems share a point to point link or
  - Systems share a multi-point link
  - Data can pass without intervening active agent
- Indirect
  - Switched networks or
  - Internetworks or internets
  - Data transfer depend on other entities

Monolithic or Structured

- Communications is a complex task
- To complex for single unit
- Structured design breaks down problem into smaller units
- Layered structure

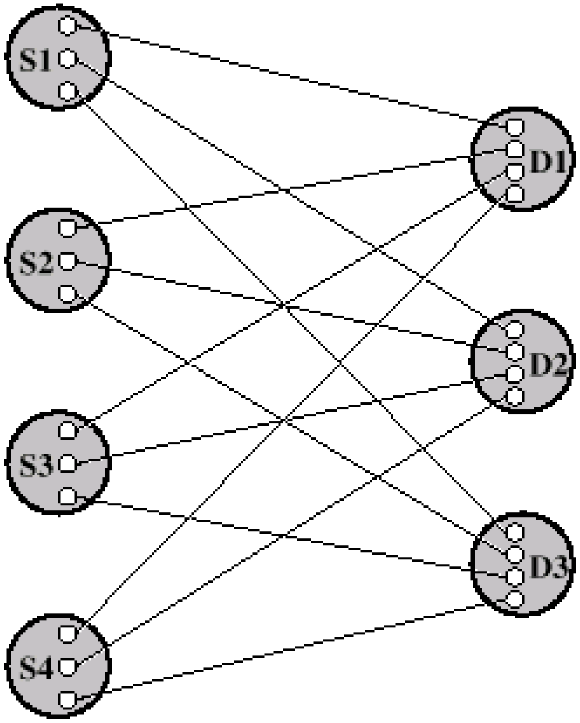
Symmetric or Asymmetric

- Symmetric
  - Communication between peer entities
- Asymmetric
  - Client/server

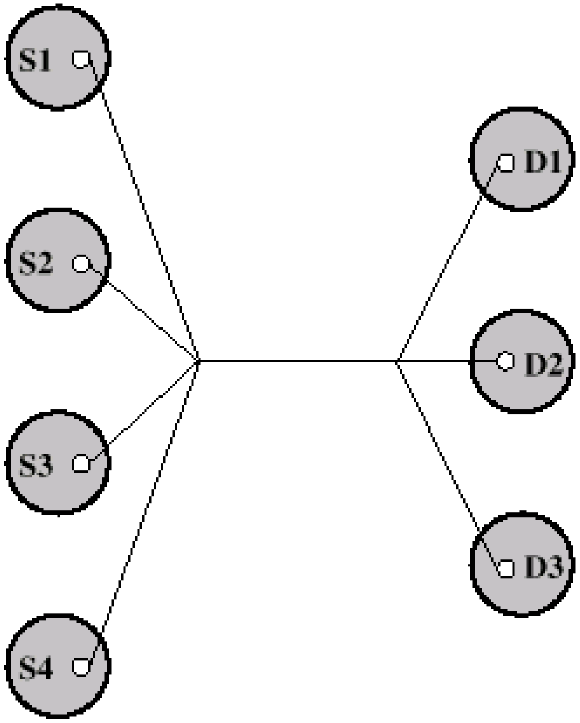
Standard or Nonstandard

- Nonstandard protocols built for specific computers and tasks
- K sources and L receivers leads to K*L protocols and 2*K*L implementations
- If common protocol used, K + L implementations needed

Use of Standard Protocols



(a) Without standards: 12 different protocols;
24 protocol implementations



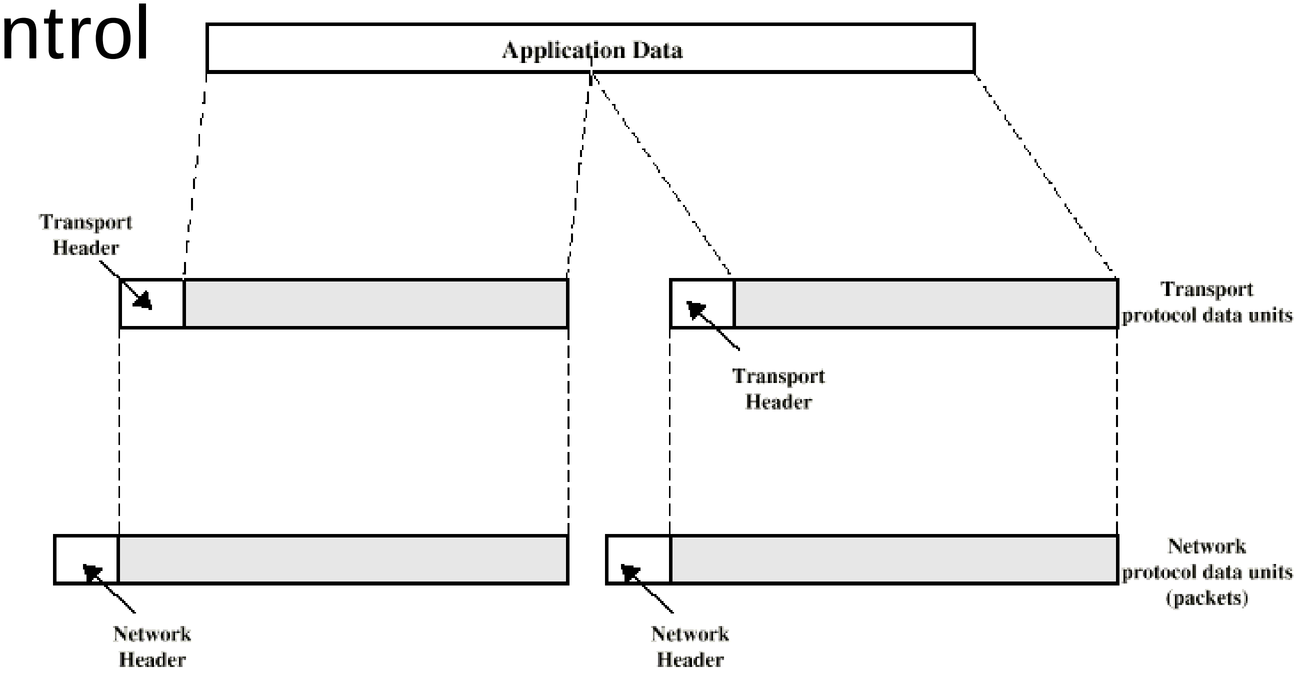
(a) With standards: 1 protocol;
7 implementations

Functions

- Encapsulation
- Segmentation and reassembly
- Connection control
- Ordered delivery
- Flow control
- Error control
- Addressing
- Multiplexing
- Transmission services

Encapsulation

- Addition of control information to data
  - Address information
  - Error-detecting code
  - Protocol control



Segmentation (Fragmentation)

- Data blocks are of bounded size
- Application layer messages may be large
- Network packets may be smaller
- Splitting larger blocks into smaller ones is segmentation (or fragmentation in TCP/IP)
  - ATM blocks (cells) are 53 octets long
  - Ethernet blocks (frames) are up to 1526 octets long
- Checkpoints and restart/recovery

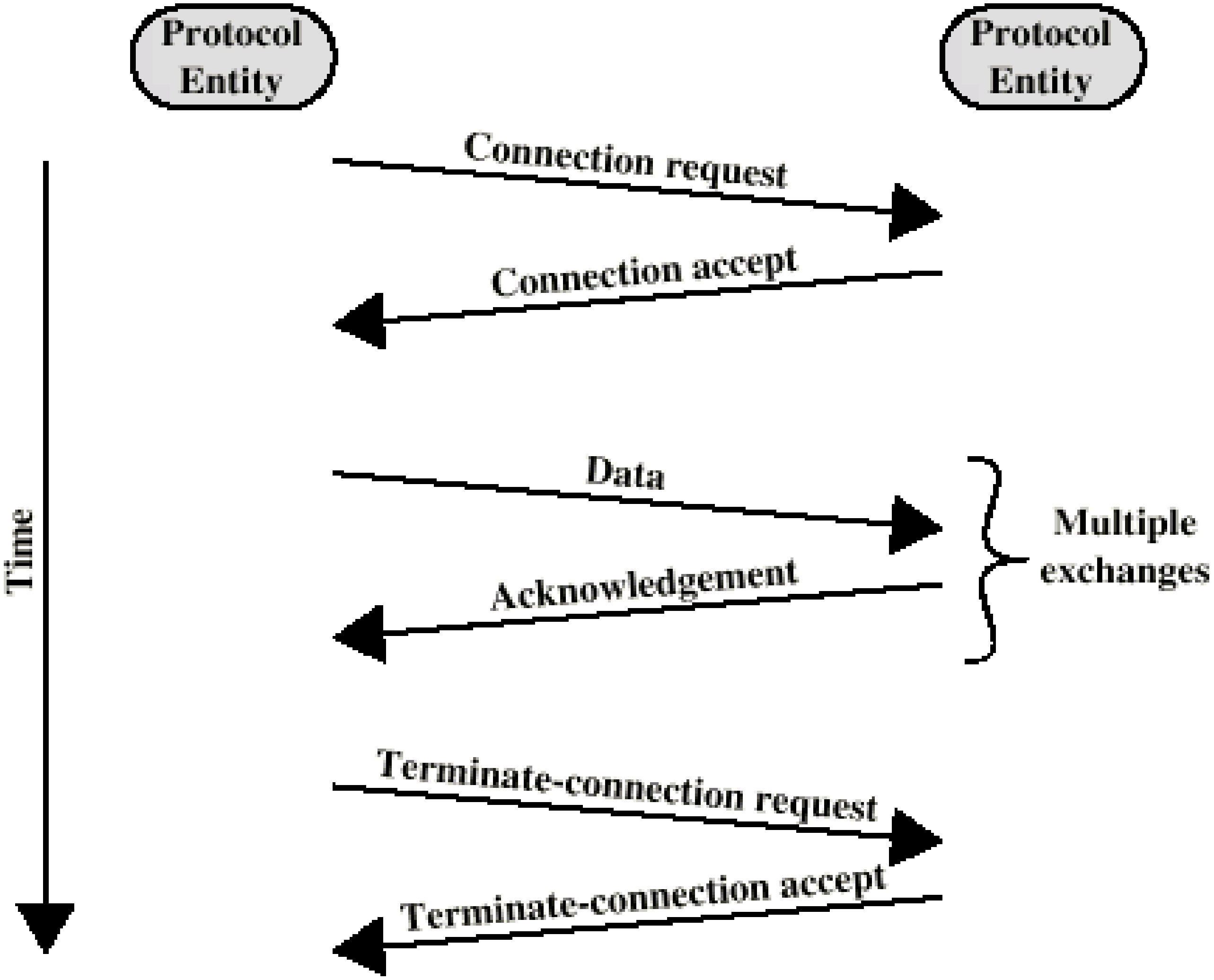
Why Fragment?

- Advantages
  - More efficient error control
  - More equitable access to network facilities
  - Shorter delays
  - Smaller buffers needed
- Disadvantages
  - Overheads
  - Increased interrupts at receiver
  - More processing time

Connection Control

- Connection establishment
- Data transfer
- Connection termination
- May be connection interruption and recovery
- Sequence numbers used for
  - Ordered delivery
  - Flow control
  - Error control

Connection Oriented Data Transfer



Ordered Delivery

- PDUs may traverse different paths through network
- PDUs may arrive out of order
- Sequentially number PDUs to allow for ordering

Flow Control

- Done by receiving entity
- Limit amount or rate of data
- Stop and wait
- Credit systems
  - Sliding window
- Needed at application as well as network layers

Error Control

- Guard against loss or damage
- Error detection
  - Sender inserts error detecting bits
  - Receiver checks these bits
  - If OK, acknowledge
  - If error, discard packet
- Retransmission
  - If no acknowledge in given time, re-transmit
- Performed at various levels

Error Control (Cont.)

- Flow control.
  - Sliding window.
  - Stop-and-wait.
- Error correcting code.
  - Frame = [m data bits + r check bits].
  - N = [m + r] = [n bit codeword].
  - 10001001 XOR 10110001=00111000=3-bits different.
  - Hamming distance=3=d.
  - D single-bit errors will be required.
  - To detect d errors, you had a distance d+1 code because with such a code there is no way that d single-bit errors can change a valid codeword into another valid codeword.
  - Parity bit: a code with a single parity bit has a distance 2.

Error Correcting Code (Cont.)

- To correct d errors, you need a distance 2d+1 code because that way the legal code words are so far apart that even with d changes, the original codeword is still closer than any other codeword, so it can uniquely determined.
- To design a code with m and r that will allow all single errors to be corrected.
- 2^m error messages.
- Each legal message has n illegal code words at a distance 1 from it.
- Each legal message requires n+1 bit patterns dedicated to it.
- Since we have 2^n total bit pattern.
  - $(N+1)2^m \leq 2^n \rightarrow (n+1)2^m \leq 2^{m+r} \rightarrow (n+1)2^m \leq 2^m . 2^r \rightarrow m+r+1 \leq 2^{n.}$
  - M is given, r can be calculated.

Hamming Code

- The bits of the codeword are numbered starting with bit 1 at the left end.
- Bits that are power of 2 (1,2,4,8,16 etc) are check bits.
- The rest (3,5,7,9 etc) are filled up with the m data bits.
- Each check bit forces the parity of some collection of bits, including it, to be even.
- To see which check bits that data bit in position k contributes to, rewrite k as a sum of powers of 2.
- Example: 11=1+2+8.

  29=1+4+8+16.
- When a codeword arrives, the receiver initializes a counter to “zero”.

Hamming Code (Cont.)

- It then examines each check bit, to see if it has correct parity.
- If not, it adds "k" to the counter.
- If the counter is zero after all the check bits have been examined (i.E. If they were all correct) the codeword is accepted as valid.
- If the counter is "nonzero", it contains the number of the incorrect bit.
- Example, if check bits 1,2,8 are in error, the inverted bit is 11, because it is the only one checked by bits 1,2 and 8.
- Hamming codes can only correct single errors.

Hamming Codes to Correct Burst Errors

- A sequence of k consecutive codeword is arranged as a matrix, one codeword per row.
- The data should be transmitted one column at a time, starting with the leftmost column.
- When all k bits have been sent, the second column is sent, and so on.
- When the frame arrives at the receiver, the matrix is reconstructed, one column at a time.
- If a burst arrow of length k occurs, at most 1 bit in each of the k code words will have been affected, but the hamming code can correct one error per codeword, so the entire block can be restored.

Error-detecting Codes

- CRC (polynomial code)(cyclic redundancy code).
- K=bit in the frame.
- Polynomial=x^{k-1} to x^0 [degree k-1].
- Example: frame=110001.

  K= 6.

  Six-term polynomial with coefficients.

  1,1,0,0,0 and 1:$x^5+x^4+x^{0.}$
- The idea is to append checksum to the end of the frame in such a way that polynomial represented by the check summed frame is divisible by g(x) , when the receiver gets the check summed frame, it tries dividing it by g(x).If there is a reminder, there has been a transmission error.

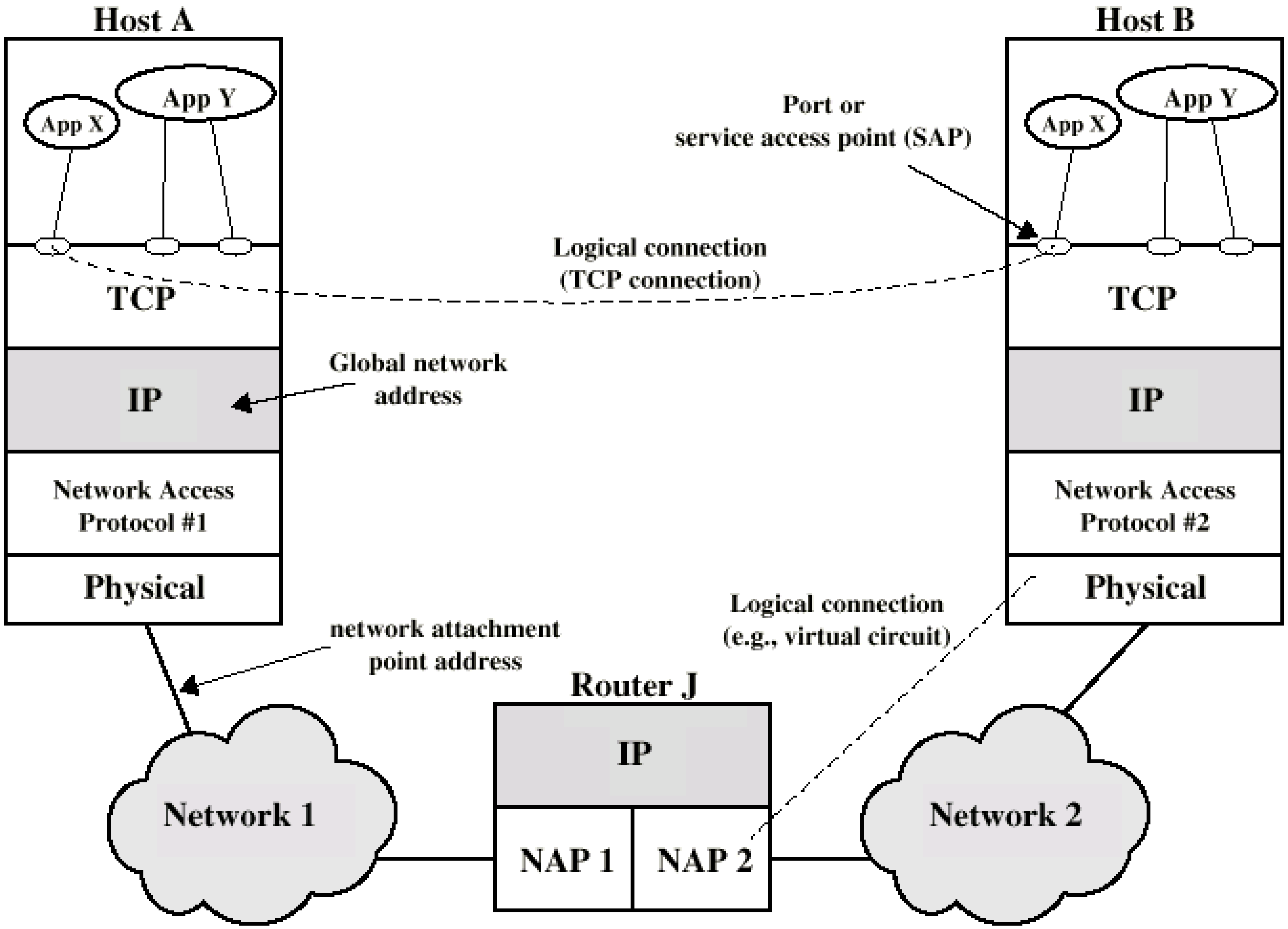
Addressing

- Addressing level
- Addressing scope
- Connection identifiers
- Addressing mode

Addressing level

- Level in architecture at which entity is named
- Unique address for each end system (computer) and router
- Network level address
  - IP or internet address (TCP/IP)
  - Network service access point or NSAP (OSI)
- Process within the system
  - Port number (TCP/IP)
  - Service access point or SAP (OSI)

Address Concepts



Addressing Scope

- Global nonambiguity
  - Global address identifies unique system
  - There is only one system with address X
- Global applicability
  - It is possible at any system (any address) to identify any other system (address) by the global address of the other system
  - Address X identifies that system from anywhere on the network
- E.G. MAC address on IEEE 802 networks

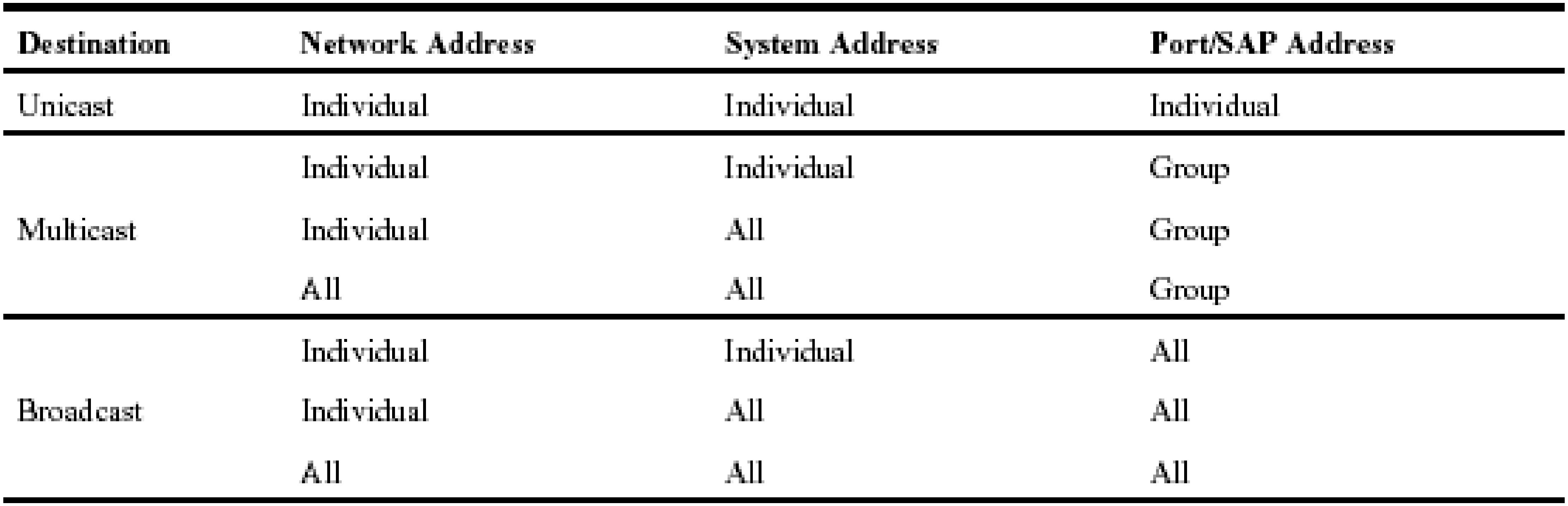
Connection Identifiers

- Connection oriented data transfer (virtual circuits)
- Allocate a connection name during the transfer phase
  - Reduced overhead as connection identifiers are shorter than global addresses
  - Routing may be fixed and identified by connection name
  - Entities may want multiple connections - multiplexing
  - The end systems can maintain state information relating to the connection

Addressing Mode

- Usually an address refers to a single system
  - Unicast address
  - Sent to one machine or person
- May address all entities within a domain
  - Broadcast
  - Sent to all machines or users
- May address a subset of the entities in a domain
  - Multicast
  - Sent to some machines or a group of users

Addressing Mode (Cont.)



Destination	Network Address	System Address	Port/SAP Address
Unicast	Individual	Individual	Individual
Multicast	Individual	Individual	Group
	Individual	All	Group
	All	All	Group
Broadcast	Individual	Individual	All
	Individual	All	All
	All	All	All

Multiplexing

- Supporting multiple connections on one machine
- Mapping of multiple connections at one level to a single connection at another
  - Carrying a number of connections on one fiber optic cable
  - Aggregating or bonding ISDN lines to gain bandwidth

Transmission Services

- Priority
  - e.g. control messages
- Quality of service
  - Minimum acceptable throughput
  - Maximum acceptable delay
- Security
  - Access restrictions

OSI - The Model

- A layer model
- Each layer performs a subset of the required communication functions
- Each layer relies on the next lower layer to perform more primitive functions
- Each layer provides services to the next higher layer
- Changes in one layer should not require changes in other layers

Principles Used in Defining the OSI Layers

- Do not create so many layers as to make the system engineering task of describing and integrating the layers more difficult than necessary.
- Create a boundary at a point where the description of services can be small and the number of interactions across the boundary are minimized.
- Create separate layers to handle functions that are manifestly different in the process performed or the technology involved.
- Collect similar functions into the same layer.
- Select boundaries at a point which past experience has demonstrated to be successful.
- Create a layer of easily localized functions so that the layer could be totally redesigned and its protocols changed in a major way to take advantage of new advances in architecture, hardware, or software technology without changing the services expected from and provided to the adjacent layers.

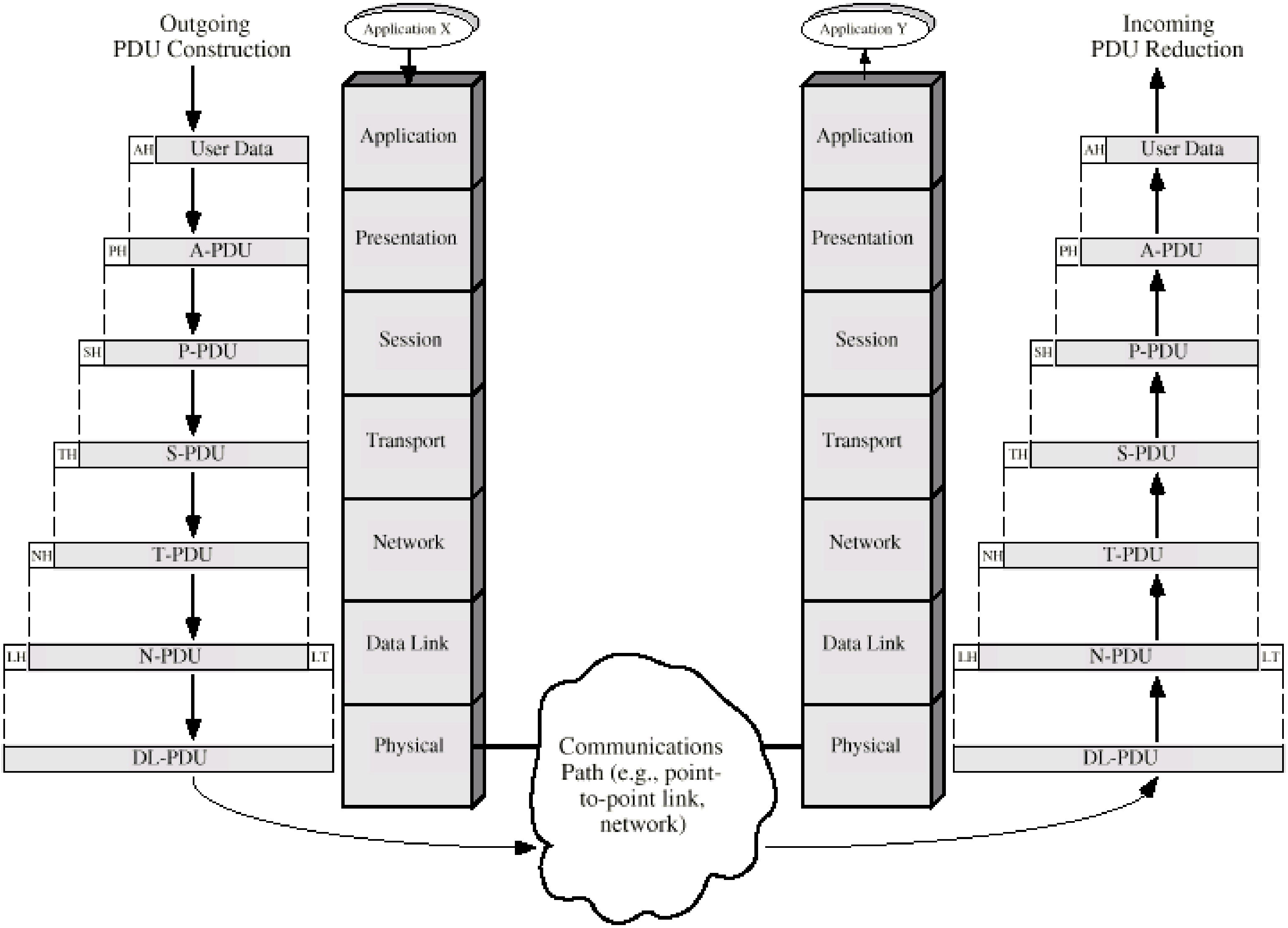
Principles Used in Defining the OSI Layers

- Create a boundary where it may be useful at some point in time to have the corresponding interface standardized.
- Create a layer where there is a need for a different level of abstraction in the handling of data, for example morphology, syntax, semantic.
- Allow changes of functions or protocols to be made within a layer without affecting other layers.
- Create for each layer boundaries with its upper and lower layer only.

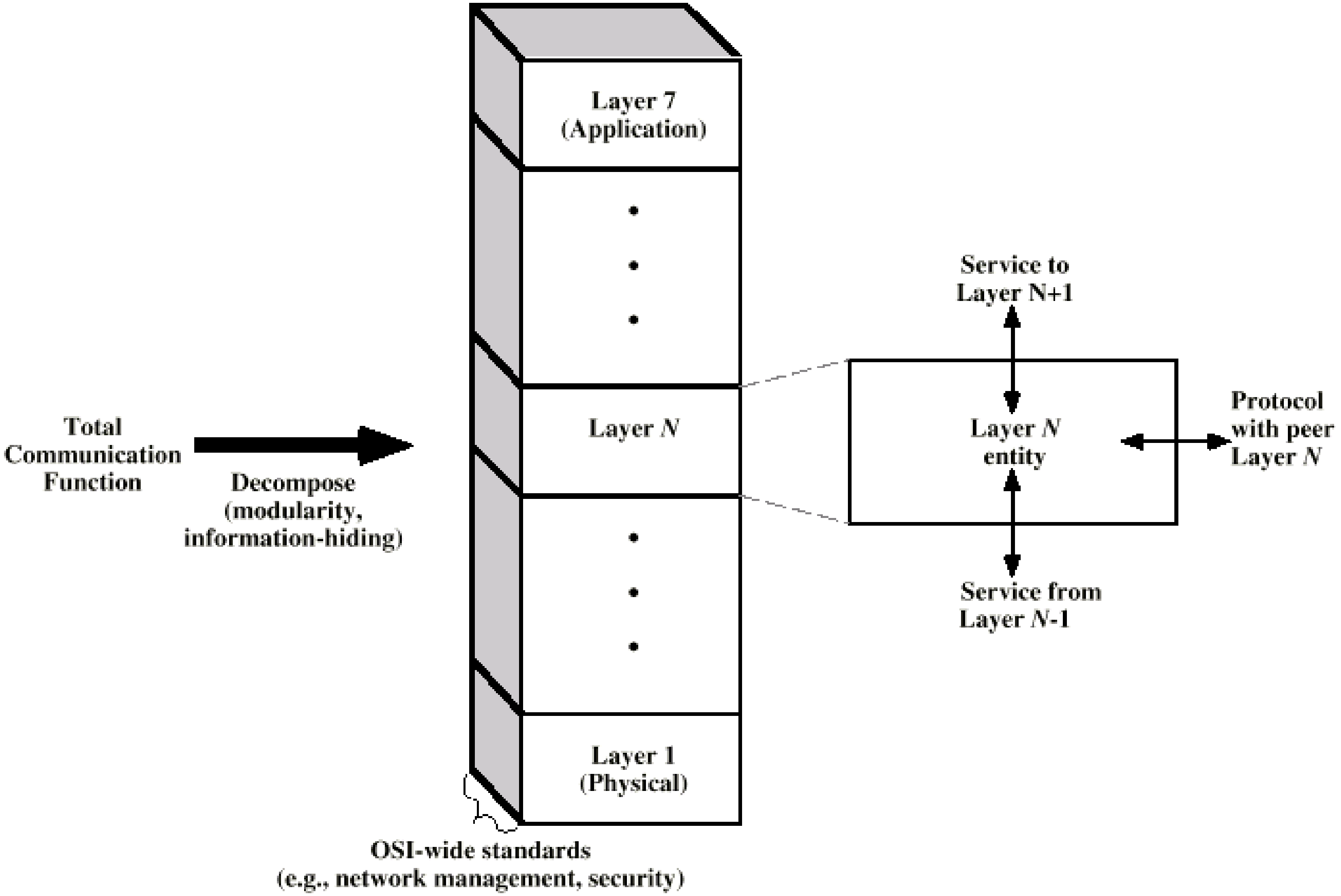
Similar principles have been applied to sublayering:

- Create further subgrouping and organization of functions to form sublayers within a layer in cases where distinct communication services need it.
- Create, where needed, two or more sublayers with a common, and therefore minimal functionality to allow interface operation with adjacent layers.
- Allow bypassing of sublayers.

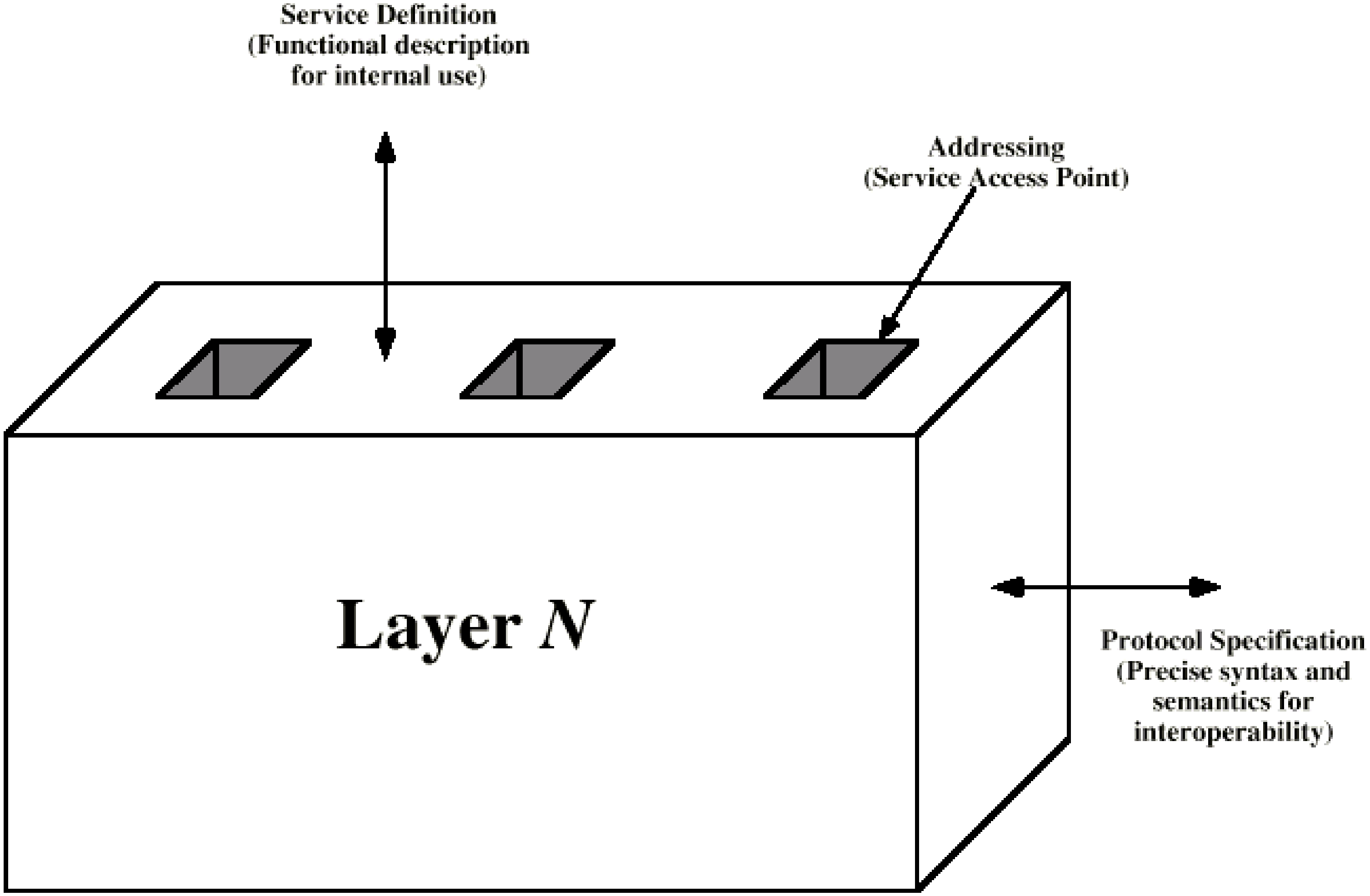
The OSI Environment



OSI As Framework for Standardization



Layer Specific Standards



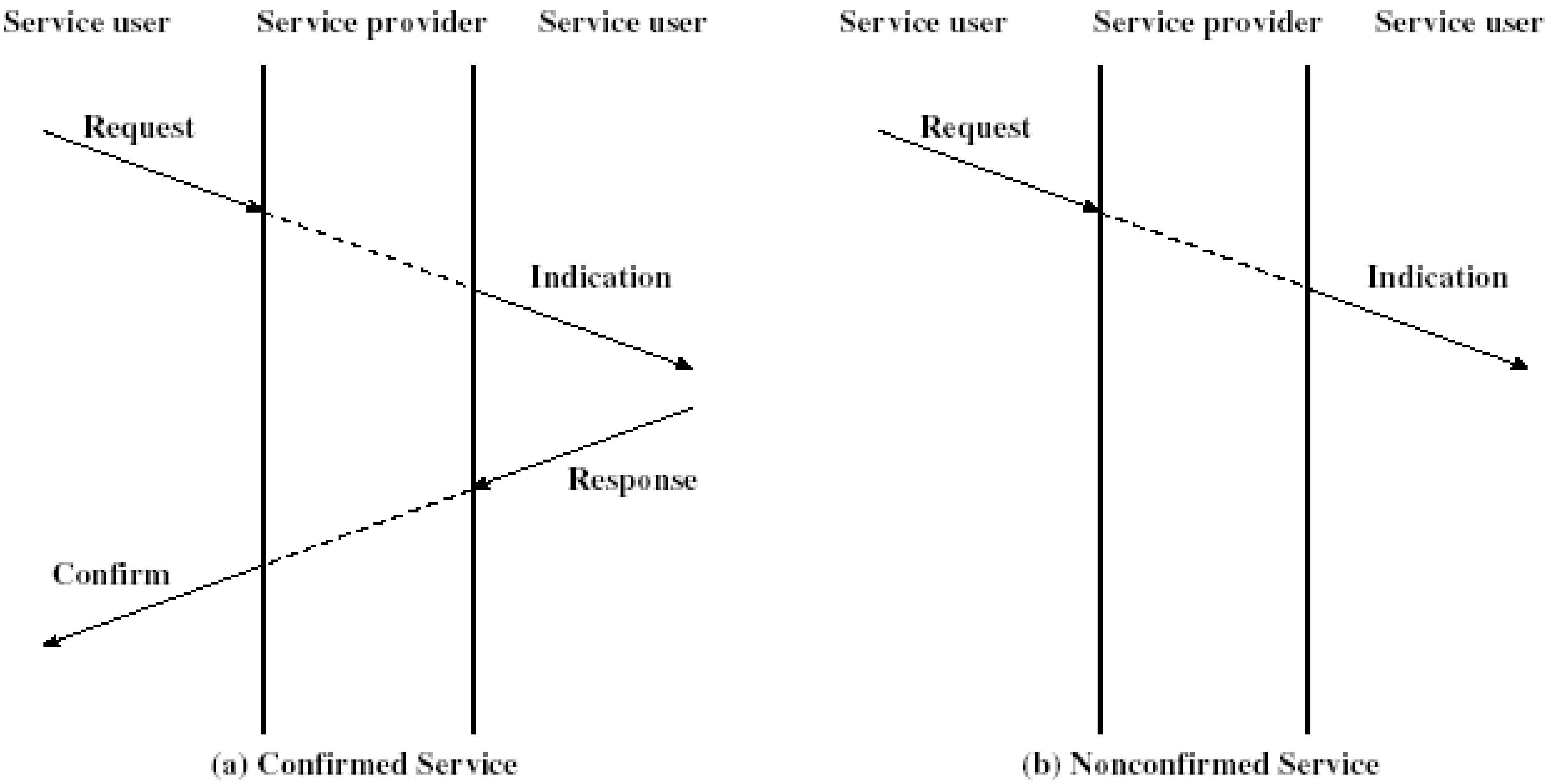
Elements of Standardization

- Protocol specification
    - Operates between the same layer on two systems
    - May involve different operating system
    - Protocol specification must be precise
        - Format of data units
        - Semantics of all fields
        - Allowable sequence of PCUs
- Service definition
    - Functional description of what is provided
- Addressing
    - Referenced by saps

Service Primitive Types

- Request
  - A primitive issued by a service user to invoke some service and to pass the parameters needed to specify fully the requested service.
- Indication
  - A primitive issued by a service provider either to:
  - Indicate that a procedure has been invoked by the peer service user on the connection and to provide the associated parameters, or
  - Notify the service user of a provider-initiated action.
- Response
  - A primitive issued by a service user to acknowledge or complete some procedure previously invoked by an indication to that user.
- Confirm
  - A primitive issued by a service provider to acknowledge or complete some procedure previously invoked by a request by the service user.

Service Primitive Types (Cont.)



OSI Layers (1)

- Physical
  - Physical interface between devices
    - Mechanical
    - Electrical
    - Functional
    - Procedural
- Data link
  - Means of activating, maintaining and deactivating a reliable link
  - Error detection and control
  - Higher layers may assume error free transmission

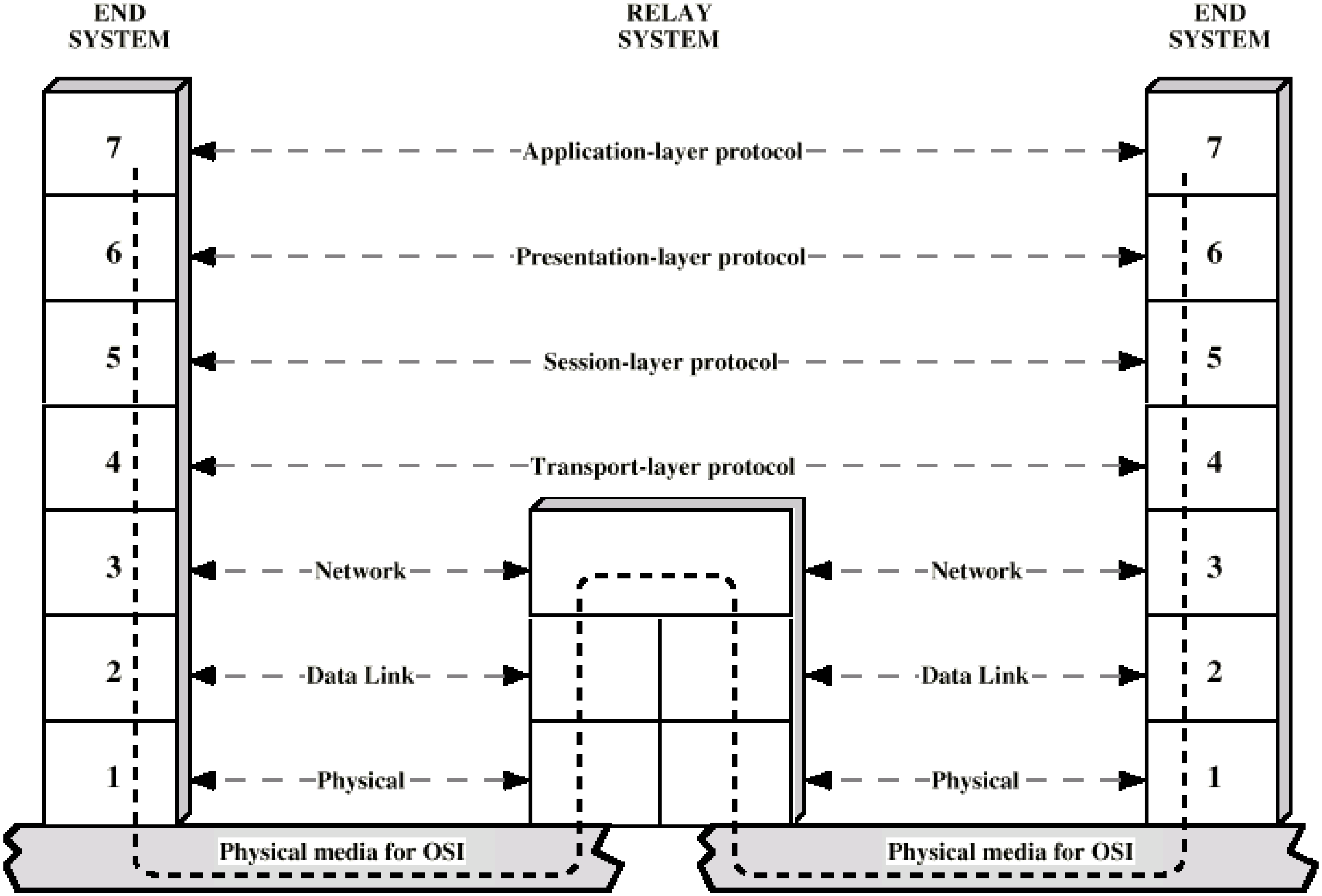
OSI Layers (2)

- Network
  - Transport of information
  - Higher layers do not need to know about underlying technology
  - Not needed on direct links
- Transport
  - Exchange of data between end systems
  - Error free
  - In sequence
  - No losses
  - No duplicates
  - Quality of service

OSI Layers (3)

- Session
  - Control of dialogues between applications
  - Dialogue discipline
  - Grouping
  - Recovery
- Presentation
  - Data formats and coding
  - Data compression
  - Encryption
- Application
  - Means for applications to access OSI environment

Use of a Relay



TCP/IP Protocol Suite

- Dominant commercial protocol architecture
- Specified and extensively used before OSI
- Developed by research funded US department of defense
- Used by the internet

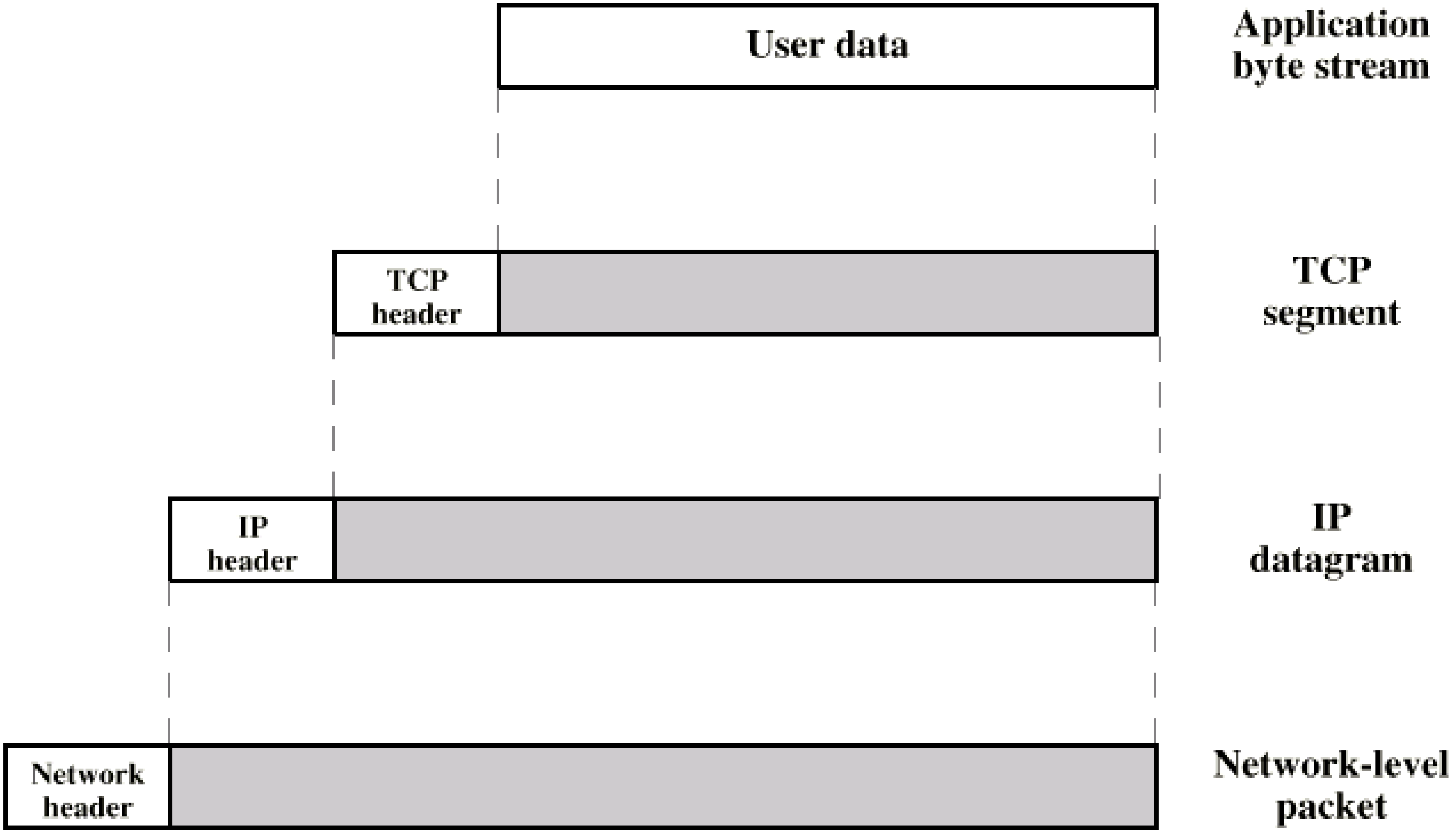
TCP/IP Protocol Architecture(1)

- Application layer
  - Communication between processes or applications
- End to end or transport layer (TCP/UDP/...)
  - End to end transfer of data
  - May include reliability mechanism (TCP)
  - Hides detail of underlying network
- Internet layer (IP)
  - Routing of data

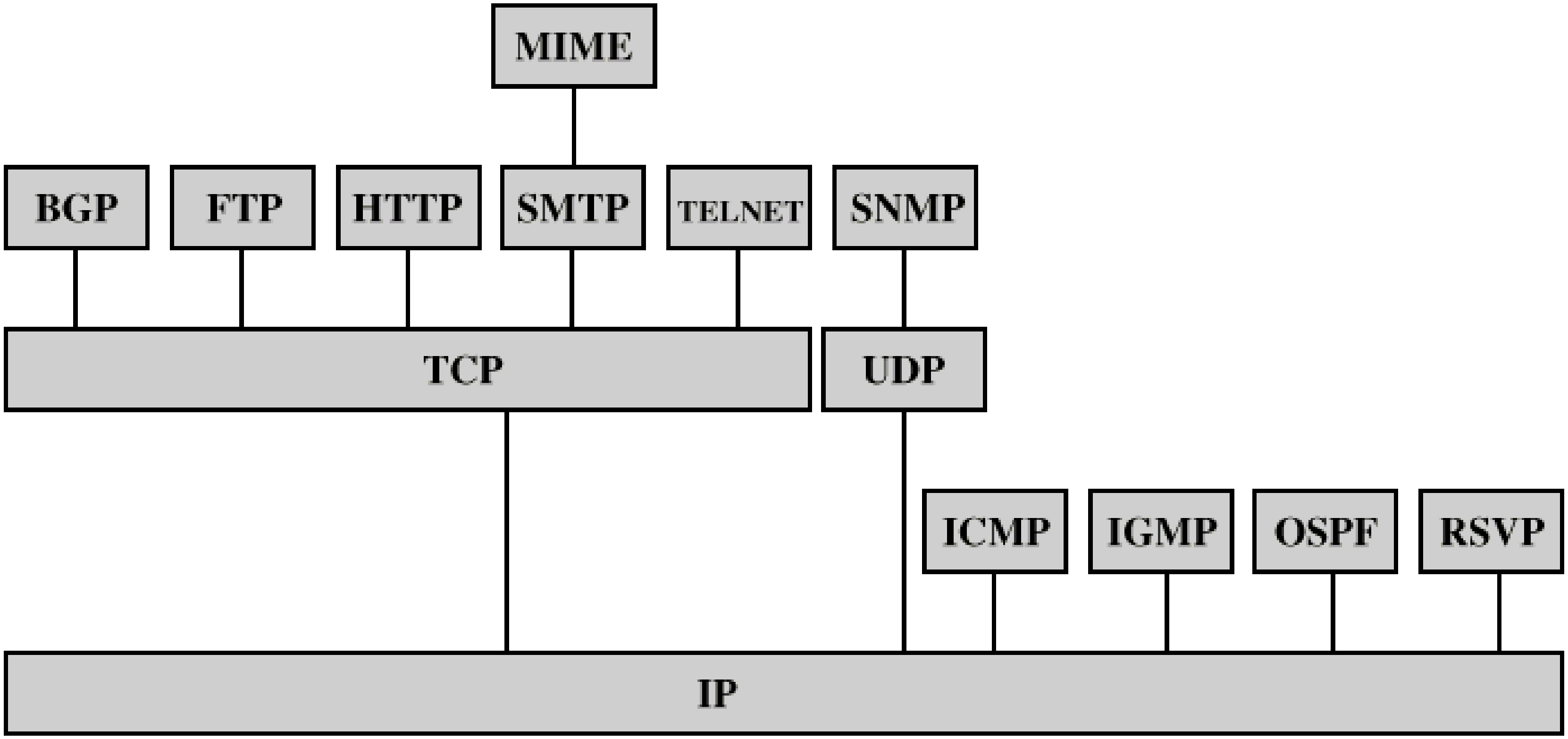
TCP/IP Protocol Architecture(2)

- Network layer
  - Logical interface between end system and network
- Physical layer
  - Transmission medium
  - Signal rate and encoding

PDUs in TCP/IP



Some Protocols in TCP/IP Suite



BGP	=	Border Gateway Protocol
FTP	=	File Transfer Protocol
HTTP	=	Hypertext Transfer Protocol
ICMP	=	Internet Control Message Protocol
IGMP	=	Internet Group Management Protocol
IP	=	Internet Protocol
MIME	=	Multi-Purpose Internet Mail Extension
OSPF	=	Open Shortest Path First
RSVP	=	Resource ReSerVation Protocol
SMTP	=	Simple Mail Transfer Protocol
SNMP	=	Simple Network Management Protocol
TCP	=	Transmission Control Protocol
UDP	=	User Datagram Protocol