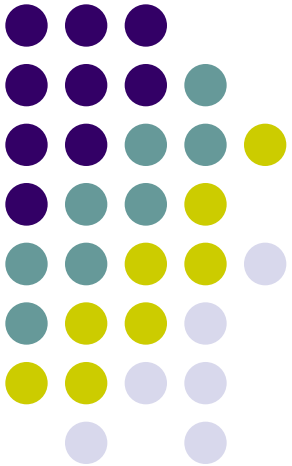
2. Applications and Layered Architectures

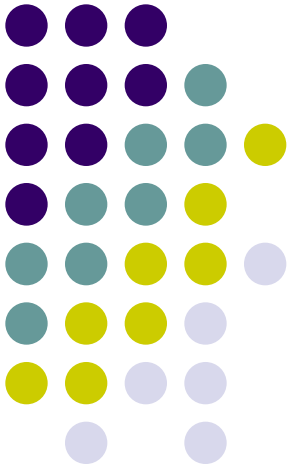
Protocols, Services & Layering
OSI Reference Model
How the Layers Work Together



Network Architectures, Layers and Services

A quick overview of:

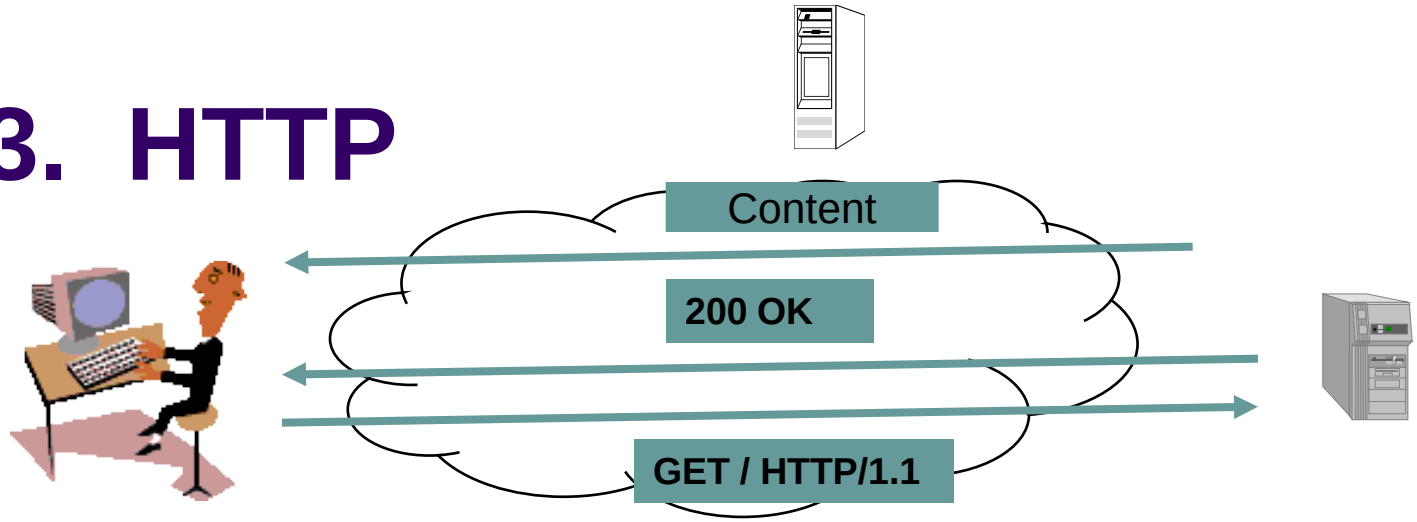
Layering, Services and Protocols



Overview at a glance: Layers, Services & Protocols

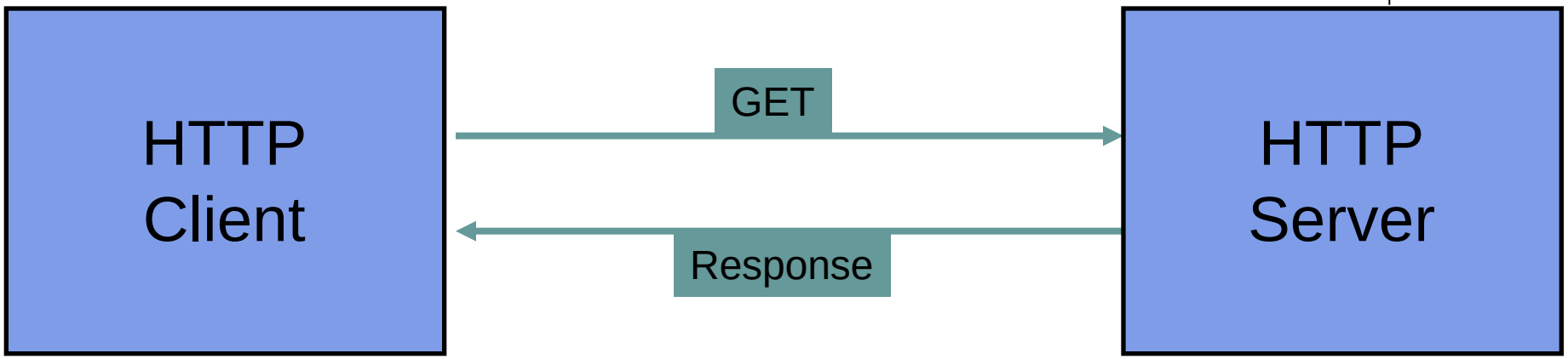
- The overall communications process between two or more communicating nodes connected across one or more networks can become ***very complex***
- ***Layering*** partitions related communications functions into groups that are manageable
- Each layer provides a ***service*** to the **layer above**
- Each layer operates according to a ***protocol***
- Here is an example:

3. HTTP



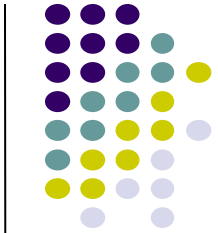
- HTTP client sends its request message: “GET …”
- HTTP server sends a status response: “200 OK”
- HTTP server sends requested file
- Browser displays document
- Clicking a link sets off a chain of events across the Internet!
- Let’s see how protocols & layers come into play…

HTTP Protocol



- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server
- In fact, HTTP client and server are ***processes*** running in two different machines across the Internet
- HTTP uses the reliable stream transfer service provided by TCP (at the Transport layer)

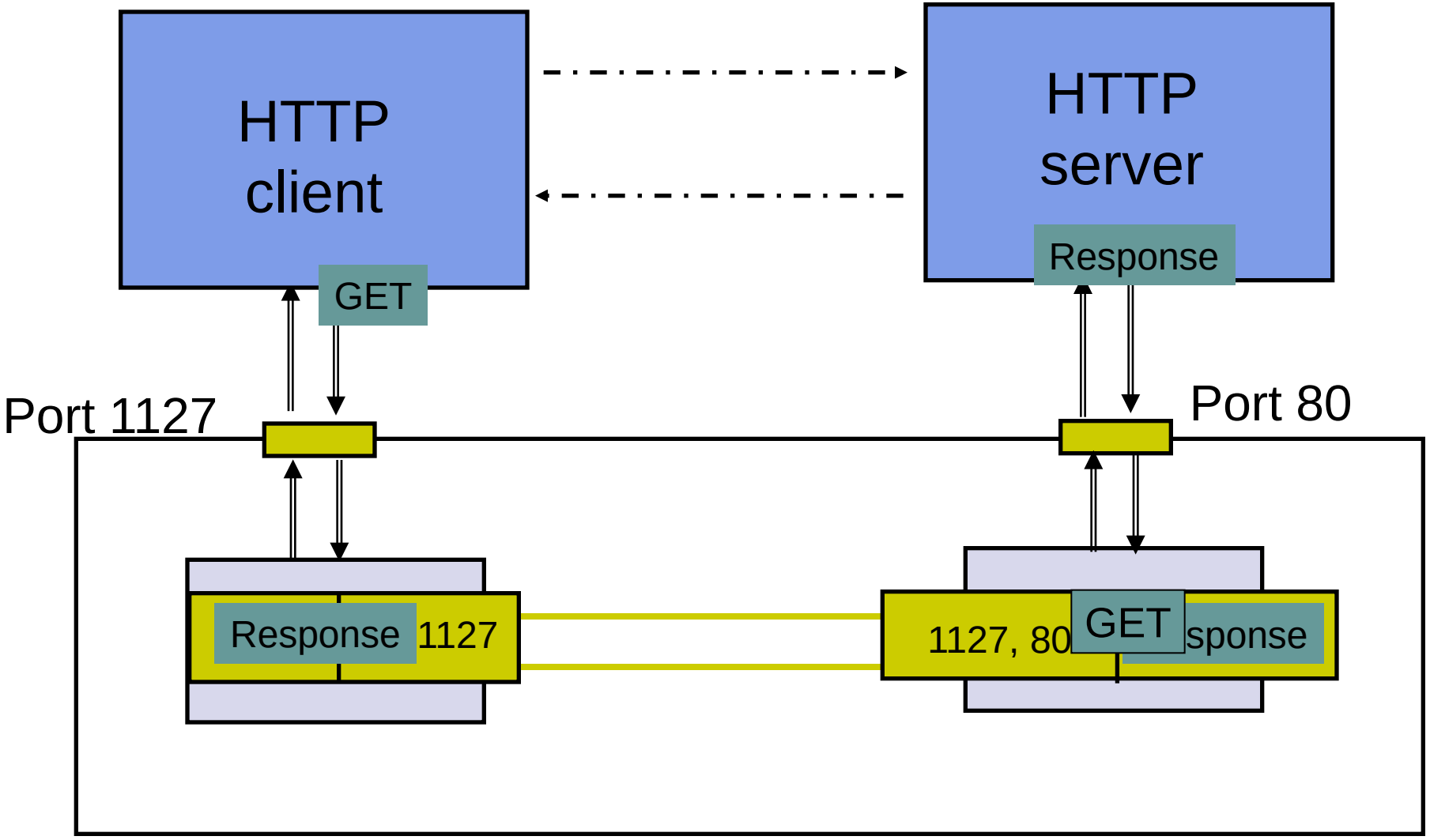
TCP



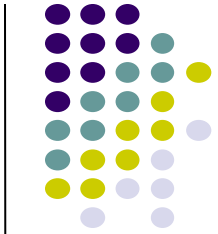
- TCP is a transport layer protocol
- Provides ***reliable byte stream*** *service* between two processes in two computers across the Internet. How to support reliable transmissions?
- TCP is *connection-oriented*: the sender and receiver must first establish an *(logical) association* and set initial sequence numbers before data is transferred.
- Sequence numbers keep track of the bytes that have been transmitted and received
- Error detection and retransmission used to recover from transmission errors and losses
- Connection ID is specified uniquely by

  (sender port #, sender IP address, receiver port #, receiver IP address)

HTTP uses service of TCP

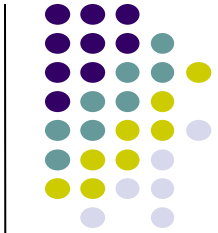


Protocols



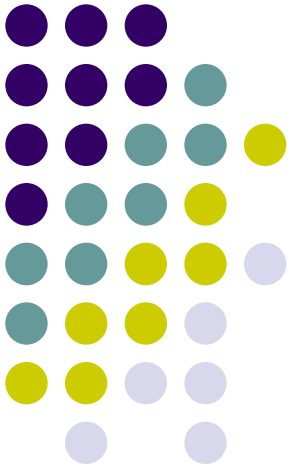
- A *protocol* is a set of rules that governs how two or more communicating entities *in a given layer* are to interact
- Specifies *messages* that can be sent and received
- *Actions* that are to be taken when a certain event occurs, e.g. sending or receiving messages, expiry of timers
- **The purpose of a protocol at a *layer n* is to provide a service to the *layer n+1* above**

Summary



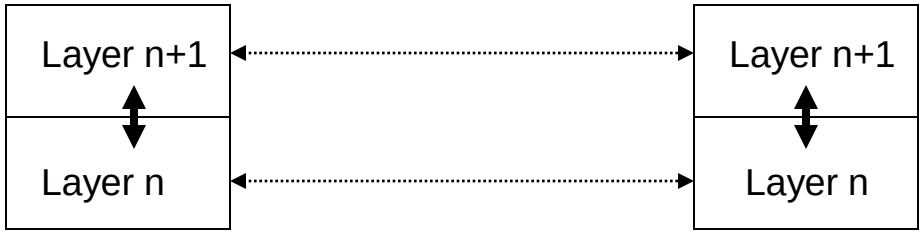
- Layers: related communications functions
  - Application Layer: HTTP, DNS
  - Transport Layer: TCP, UDP
  - Network Layer: IP
- A *protocol* is a set of rules that governs how two or more communicating entities in a layer are to interact
- Services: a protocol provides communication service(s) to the layer above
  - TCP provides connection-oriented reliable byte transfer service
  - UDP provides best-effort datagram service
- Each layer builds on services of lower layers
  - HTTP builds on top of TCP
  - DNS builds on top of UDP
  - TCP and UDP build on top of IP

OSI Reference Model

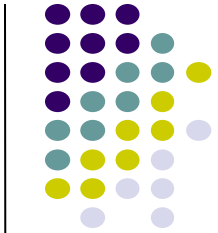


Layered Architecture

- A ***Layer*** defines a set of related communication functions that can be managed and grouped together
- Devices communicate at the same layer with the help of ***Protocols***.
- Layer n+1(upper) acts a *client* to layer n(lower). Layer n is the *server* to layer n+1.



Why Layering?

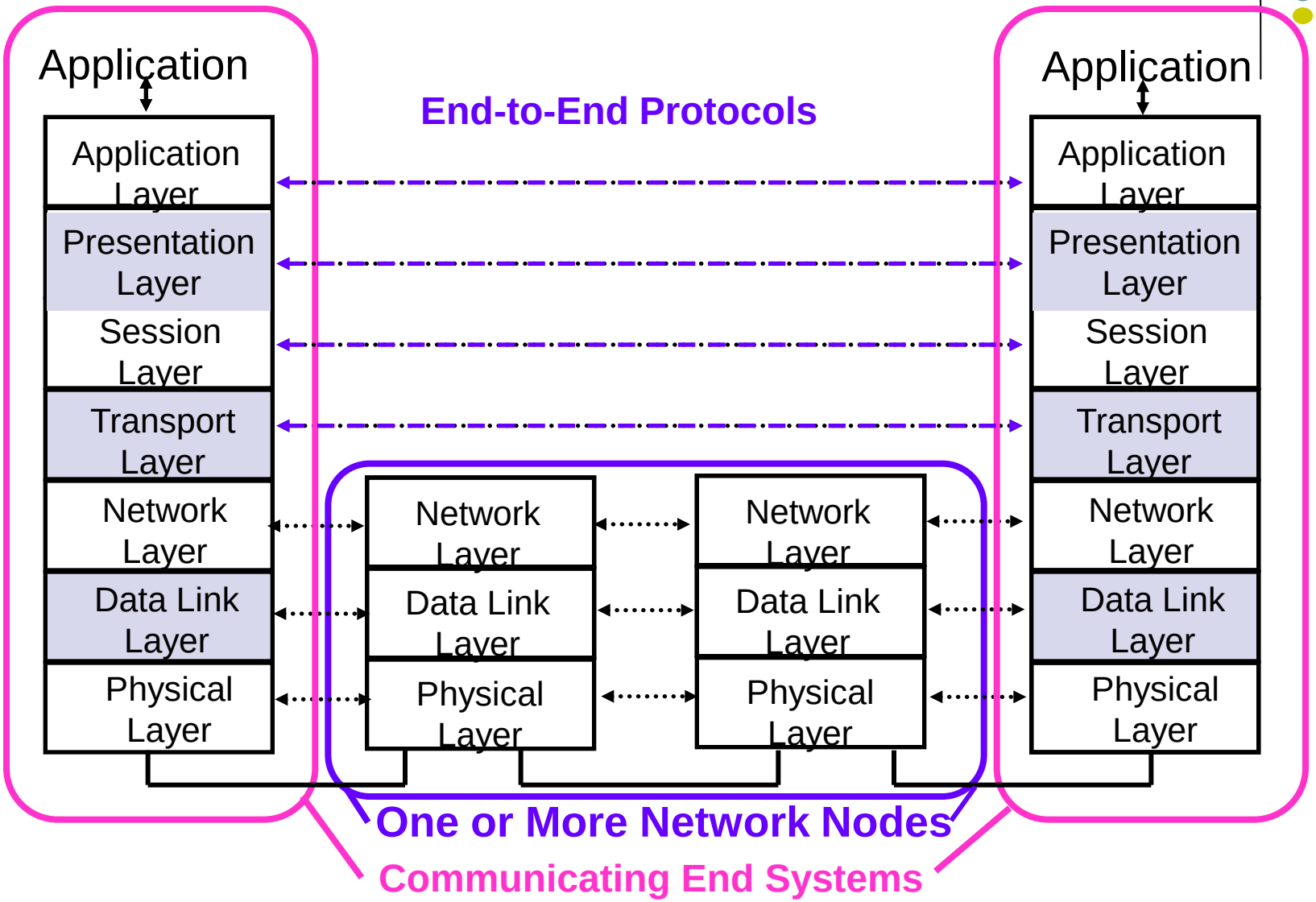


- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts
- Protocol in each layer can be designed separately from those in other layers. Protocol makes “calls” for services from layer below
- Layering provides flexibility for modifying and evolving protocols and services without having to change layers below
- Monolithic non-layered architectures are costly, inflexible, and soon obsolete

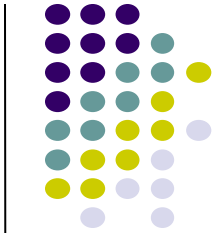
Open System Interconnection (OSI) Reference Model

- Describes a seven-layer abstract reference model for a network architecture
- Purpose of the reference model was to provide a framework for the development of protocols
- OSI also provided a unified view of layers, protocols, and services which is still in use in the development of new protocols
- Detailed standards were developed for each layer, but today many of these are not in use
- TCP/IP protocols preempted deployment of OSI protocols

7-Layer OSI Reference Model



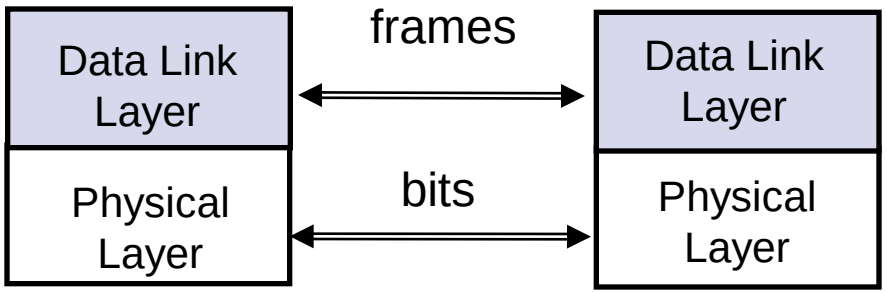
Physical Layer



- Transfers bits across a link
- Definition & specification of the physical aspects of a communications link
  - Mechanical: cable, plugs, pins...
  - Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  - functional/procedural: how to activate, maintain, and deactivate physical links…
- Ethernet, DSL, cable modem, telephone modems…
- Twisted-pair cable, coaxial cable optical fiber, radio, infrared, …

Link Layer

- Transfers *frames* across *direct* connections (point to point)
  - Groups bits into frame
- Detection of bit errors; **Retransmission** of frames
- Activation, maintenance, & deactivation of data link connections
- **Medium access control (MAC)** for local area networks
- **Flow control**

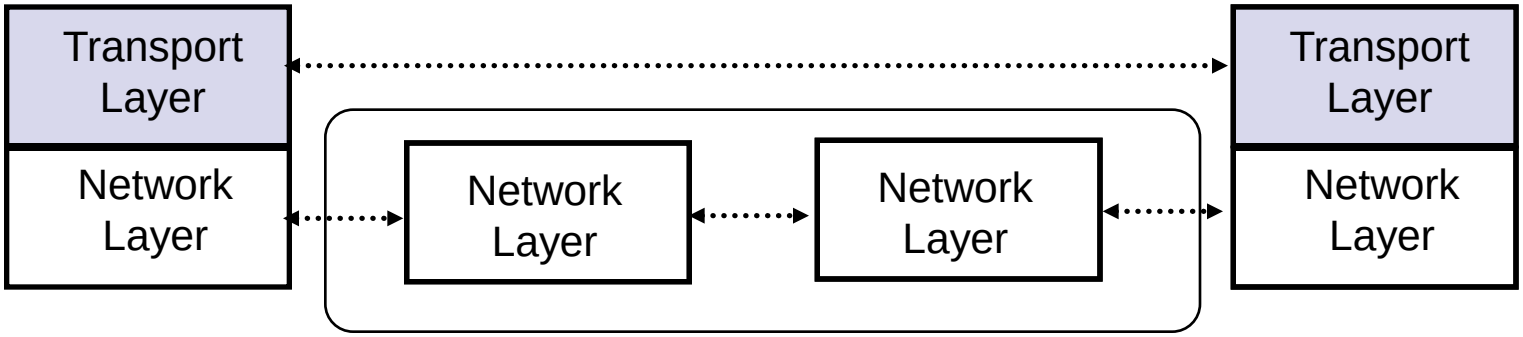


Network Layer

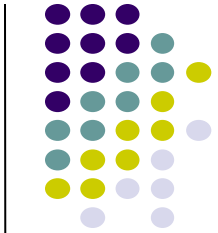
- Transfers *packets* across ***multiple links*** and/or ***multiple networks***
- Addressing must scale to large networks
- Nodes *jointly* execute routing algorithm (*distributed algorithm*) to determine paths across the network
- Forwarding transfers packet across a node
- **Congestion control** to deal with traffic surges
- Connection setup, maintenance, and teardown when connection-based

Transport Layer

- Transfers data **end-to-end (or host-to-host/server)** from a process in a machine to a process in another machine
- Reliable stream transfer or quick-and-simple single-block transfer
- Port numbers enable multiplexing
- **Message segmentation and reassembly**
- **Session** setup, maintenance, and release

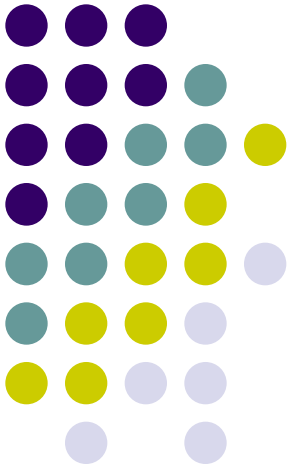


Upper Layers

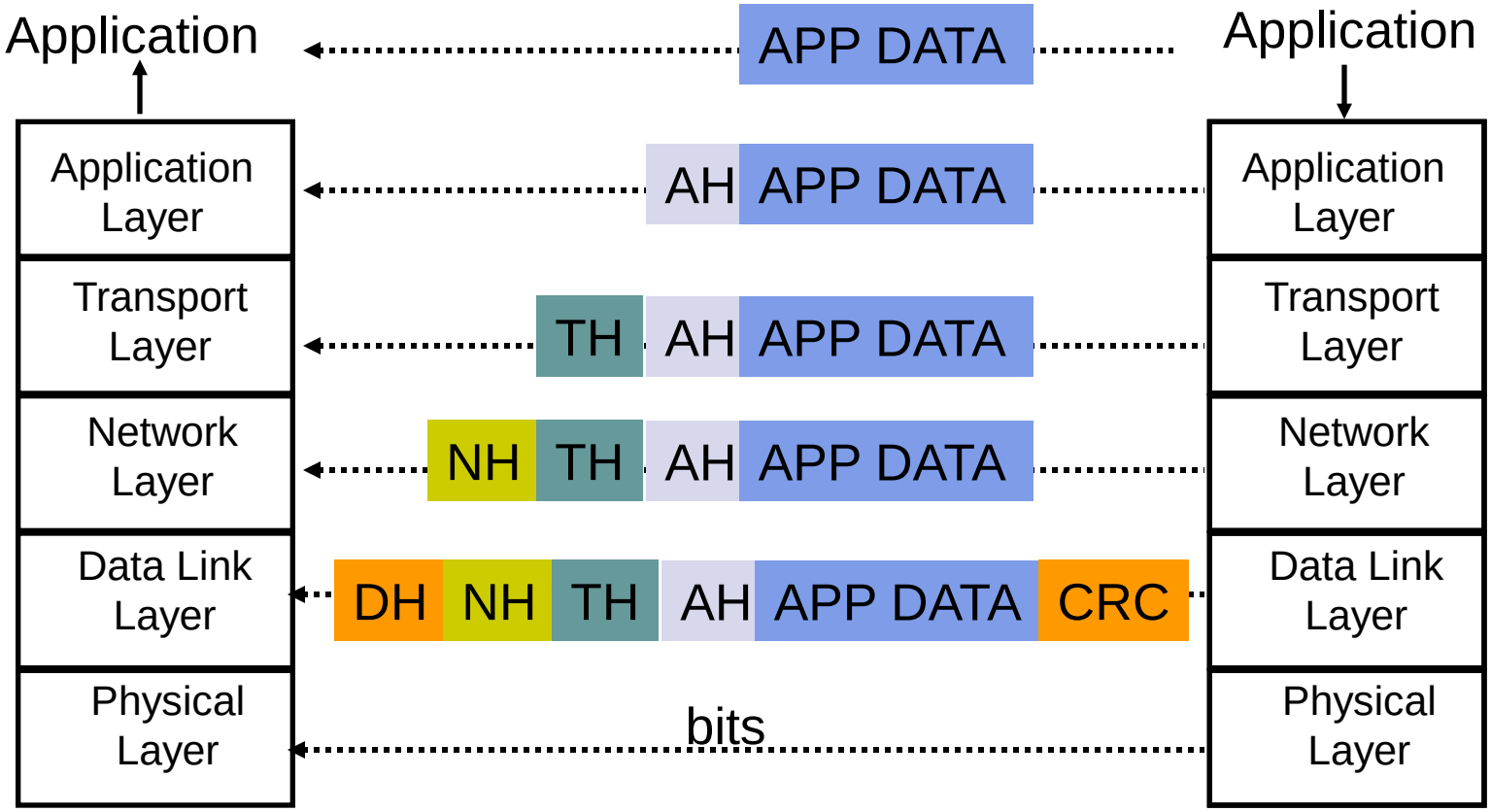


- Session Layer: Allows users of different machine to establish a session between them
- Presentation Layer: Concerned with the syntax and semantics of information transmitted
- Application Layer: Contains variety of protocols that are needed by users, e.g. hyper text transfer protocol (HTTP), File Transfer Protocol (FTP), etc.

How the layers work together

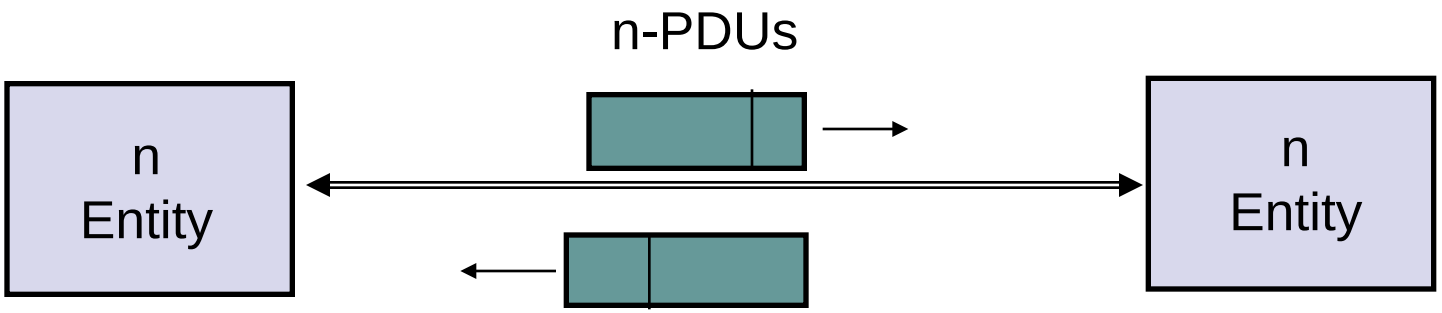


Headers and Trailers



OSI Unified View: Protocols

- Layer n in one machine interacts with layer n in another machine to provide a service to layer n +1
- The entities comprising the corresponding layers on different machines are called *peer processes*.
- The machines use a set of rules and conventions (the *layer-n protocol)*.
- Layer-n peer processes communicate by exchanging *Protocol Data Units* (PDUs)

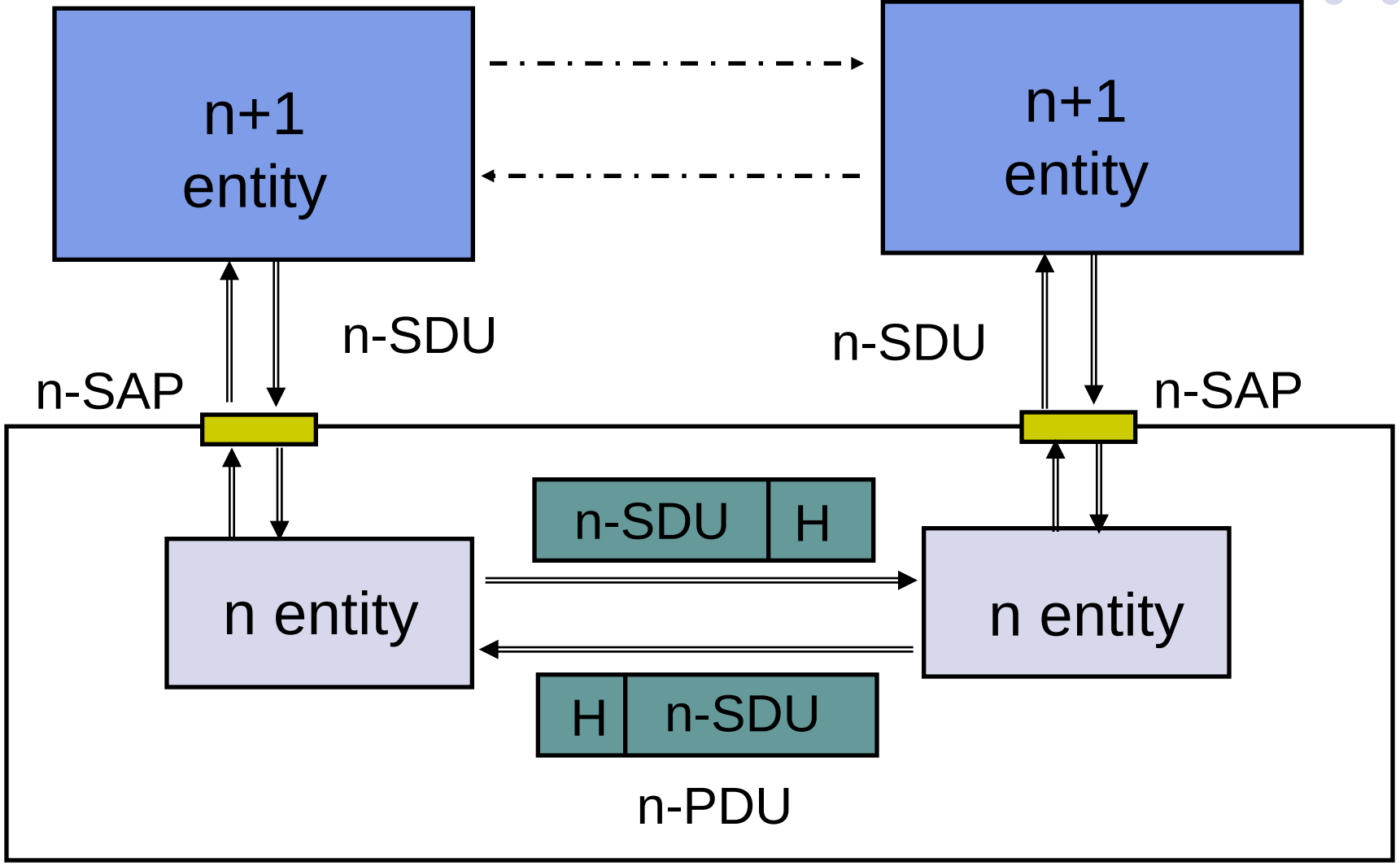


Layer n peer protocol

OSI Unified View: Services

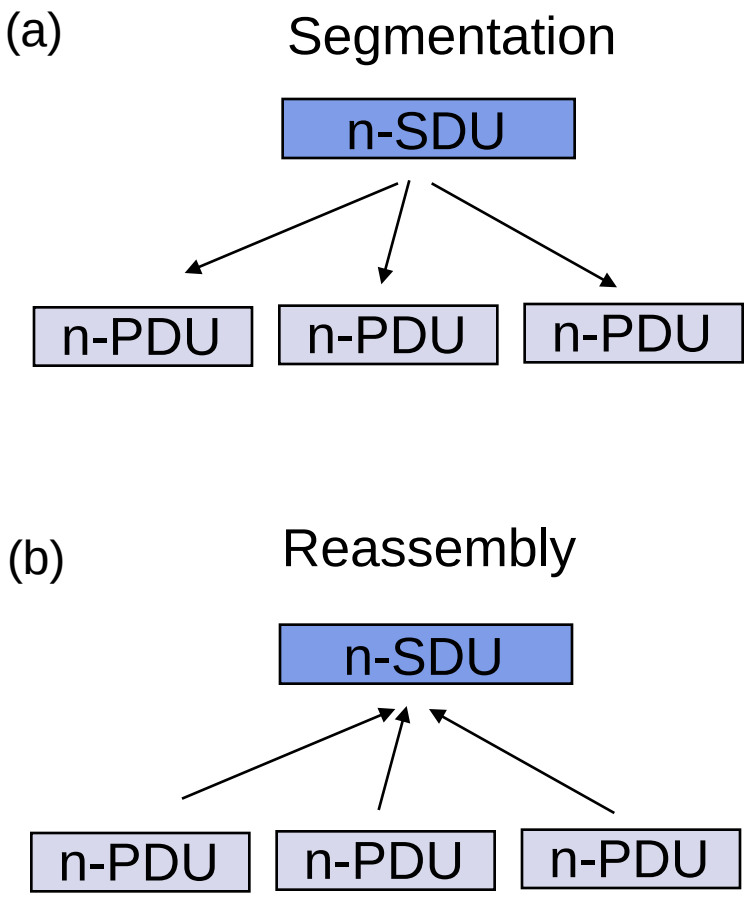
- Communication between peer processes is virtual and actually indirect
- Layer n+1 transfers information by invoking the services provided by layer n
- Services are available at *Service Access Points (*SAP's)
- Each layer passes data & control information to the layer below it until the physical layer is reached and transfer occurs
- The data passed to the layer below is called a *Service Data Unit* (SDU)
- SDU's are *encapsulated* in PDU's

Layers, Services, and Protocols



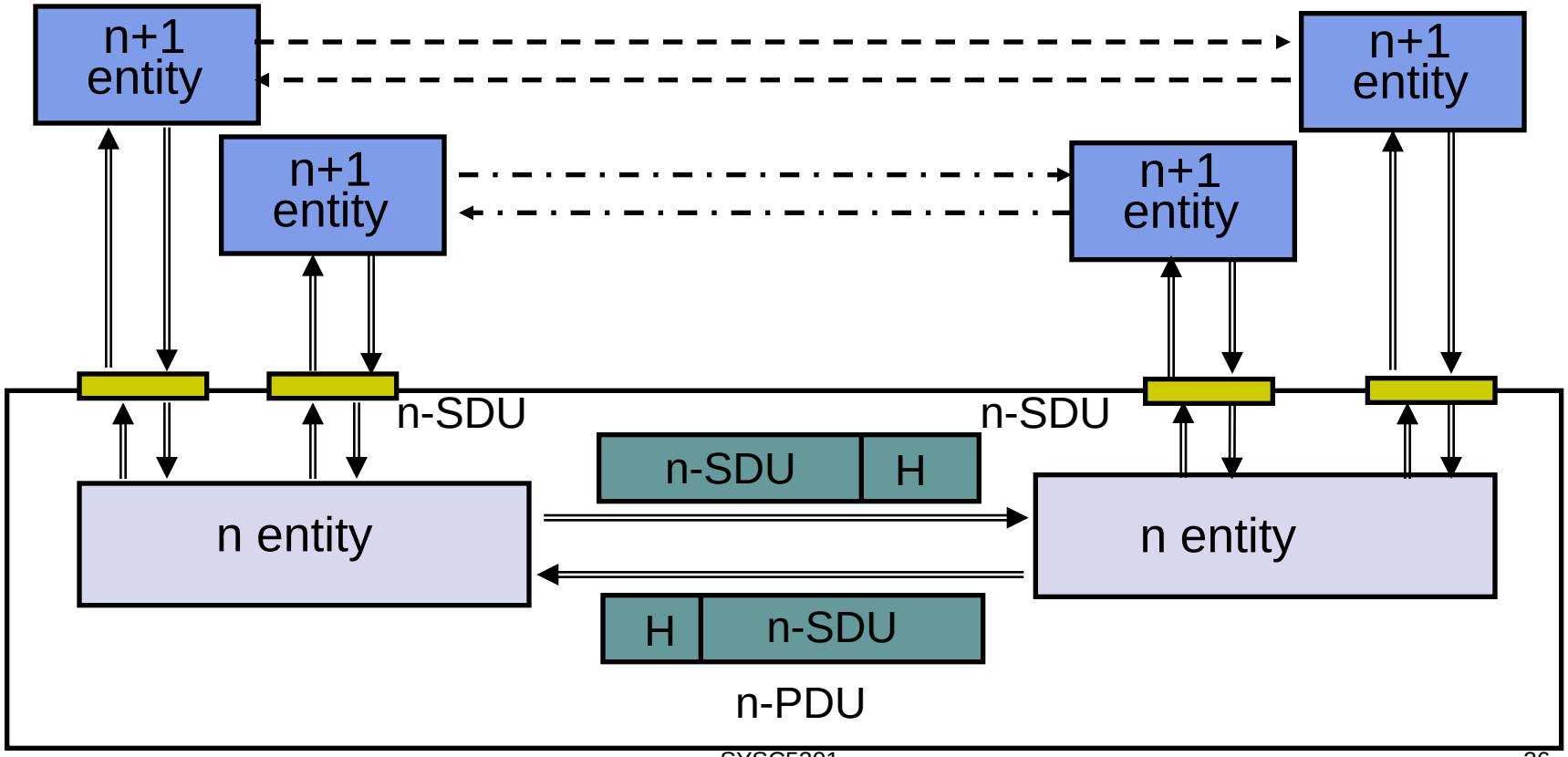
Segmentation and Reassembly

- A layer may impose a limit on the size of a data block that it can transfer for implementation or other reasons
- Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1)
- Sender side: SDU is segmented into multiple PDUs
- Receiver side: SDU is reassembled from sequence of PDUs

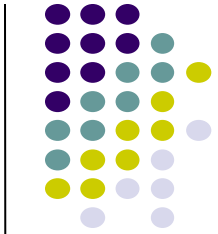


Multiplexing

- Sharing of layer n service by *multiple* layer n+1 users/processes
- Multiplexing tag or ID required in each PDU to determine which users/processes an SDU belongs to



Connectionless & Connection-Oriented Services



- Connection-Oriented
  - Three-phases:
    1. Connection setup between two SAPs to initialize state information
    2. SDU transfer
    3. Connection release
  - E.g. TCP, ATM
- Connectionless
  - Immediate SDU transfer
  - No connection setup
  - E.g. UDP, IP
- Layered services need not be of same type
  - TCP operates over IP
  - IP operates over ATM

  (although ATM was not designed for IP originally…!)