The Internet
and
HTTP and DNS Examples

The Internet and an internet

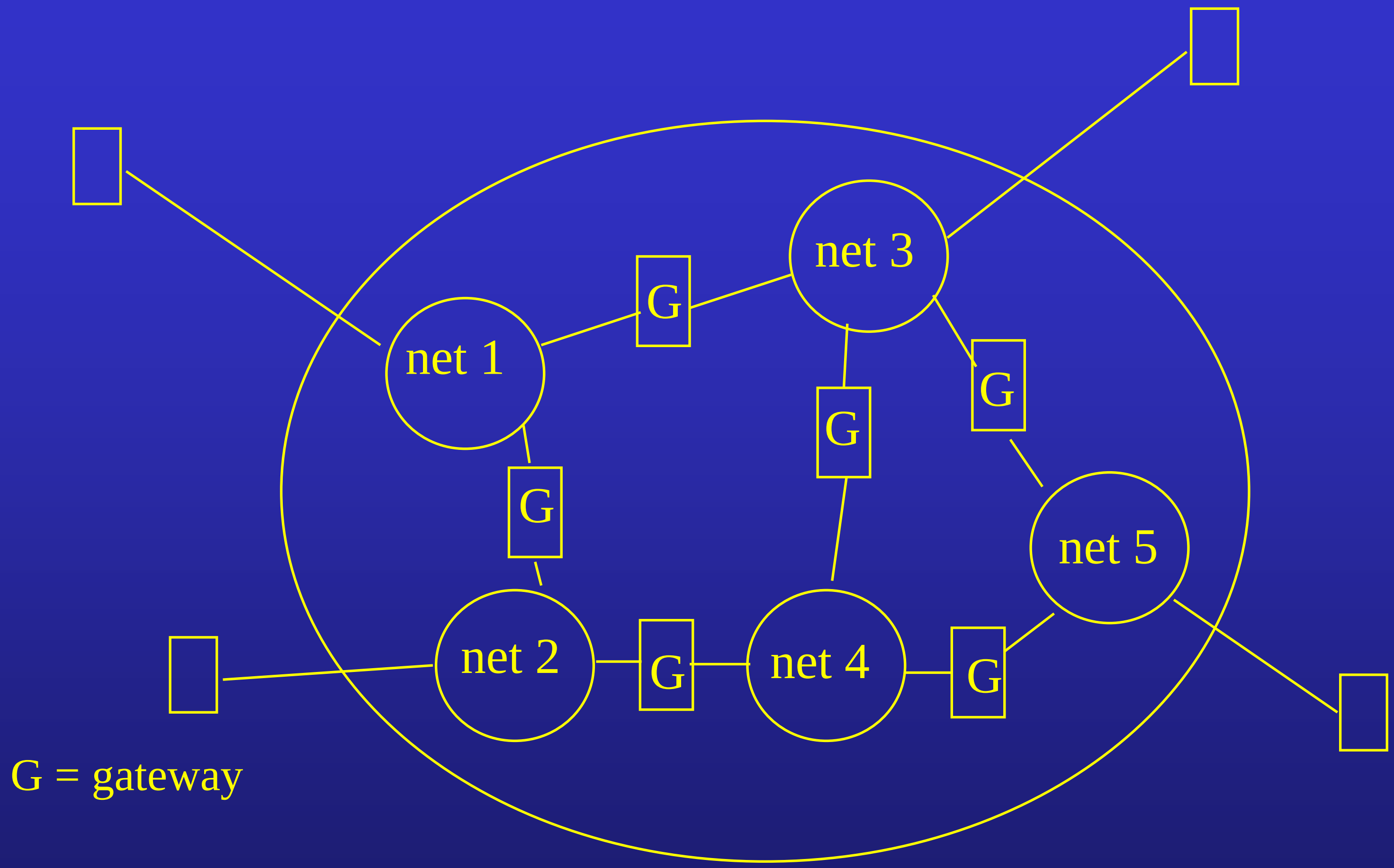
[LG&W pp.26-28]

An **internet** :: involves the *interconnection* of multiple networks into a single large networks.

The **Internet** :: refers to the successor to ARPANET.

IP (the Internet Protocol) :: provides *connectionless* transfer of packets across an internet.

An internet

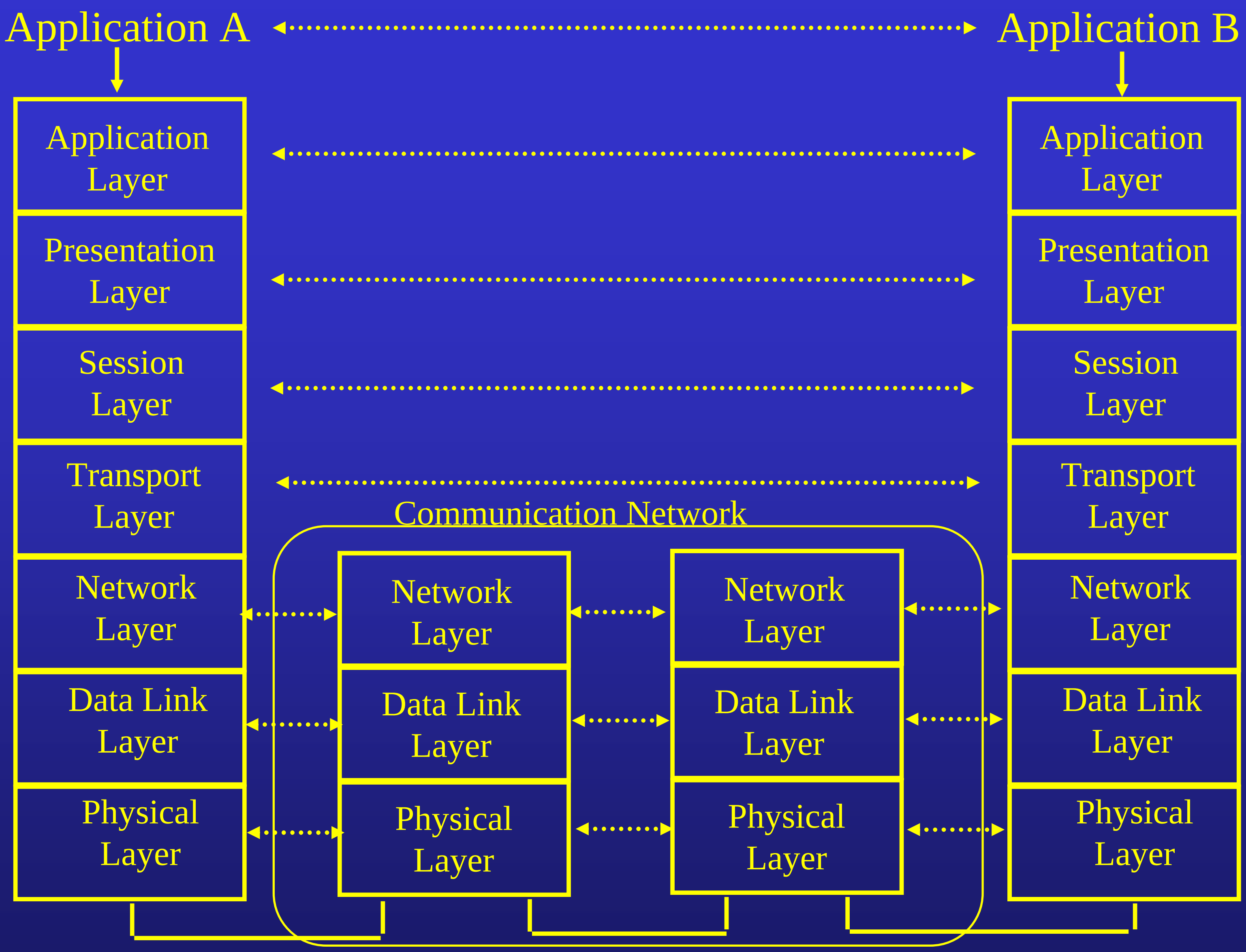


Leon-Garcia & Widjaja: *Communication Networks*

Figure 1.18

The Internet

- Provides a *name space* to refer to machines connected to the Internet (e.g. chablis.cs.wpi.edu).
- The name space is hierarchical, but is only administrative and not used in network routing operations.
- **DNS** (Domain Name Service) provides automatic translation of names to addresses.



Application A

Application B

Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Communication Network

Network Layer

Data Link Layer

Physical Layer

Network Layer

Data Link Layer

Physical Layer

Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

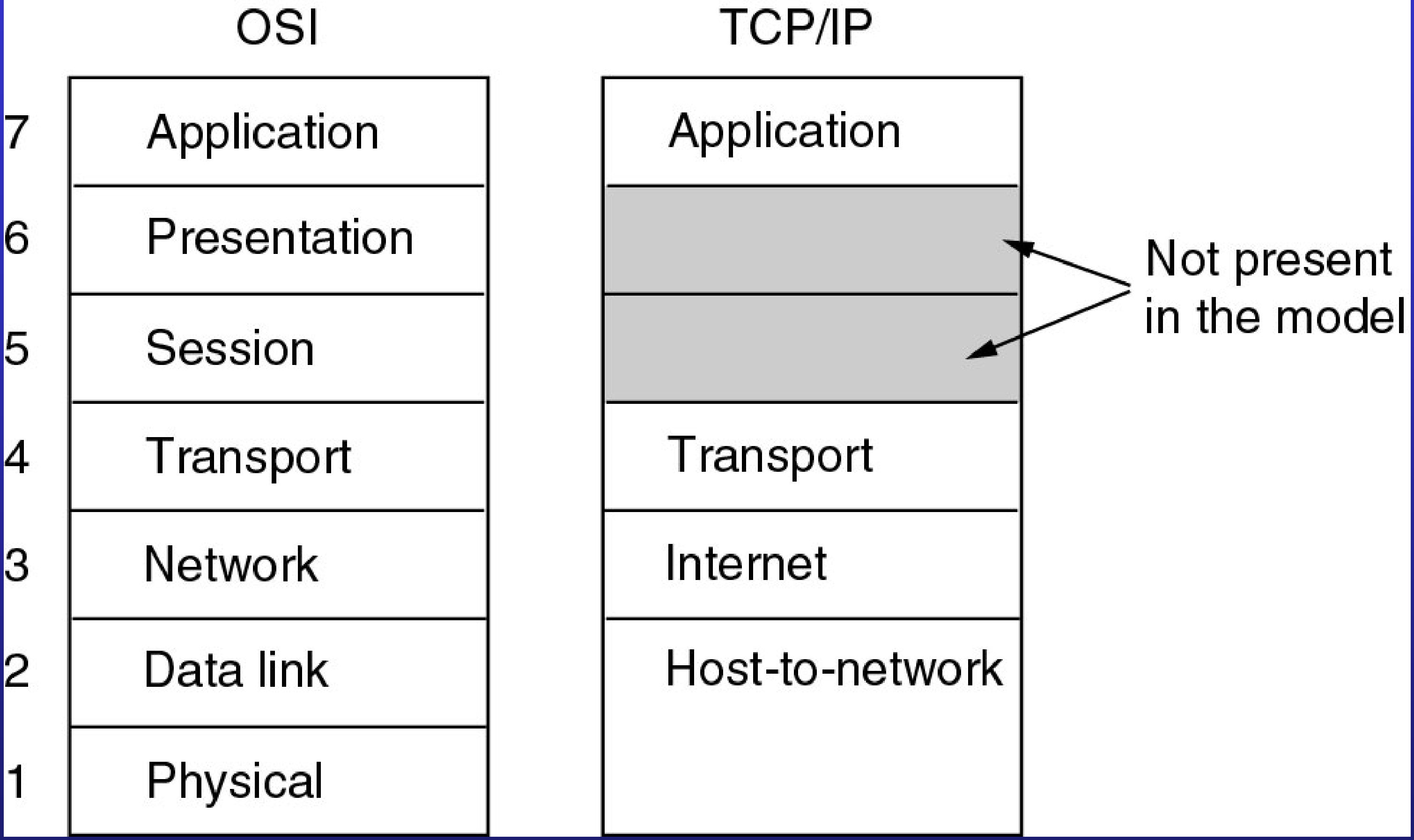
Physical Layer

Electrical and/or Optical Signals

Leon-Garcia & Widjaja: *Communication Networks*

Figure 2.6

OSI versus TCP/IP

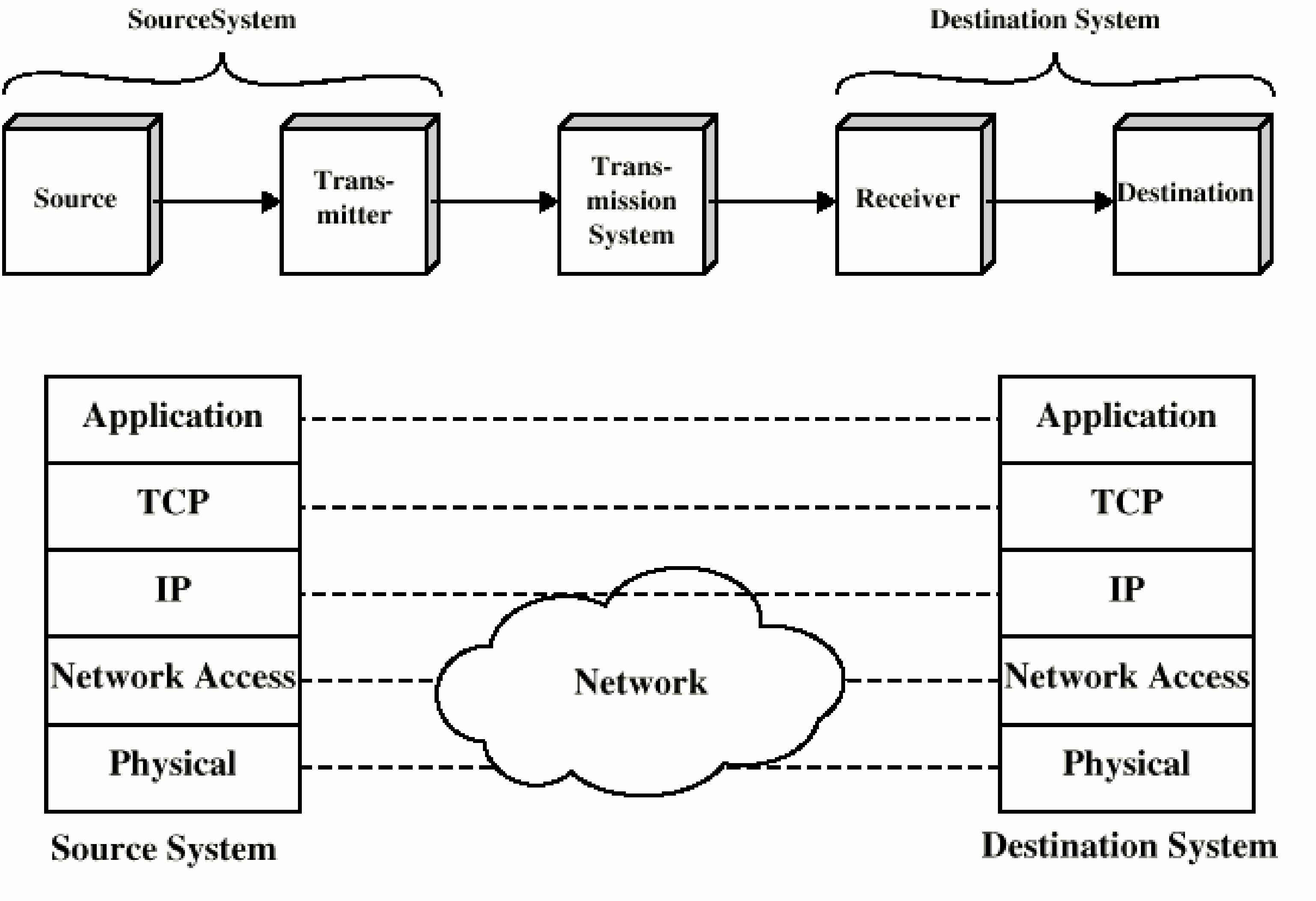


	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data link	Host-to-network
1	Physical	

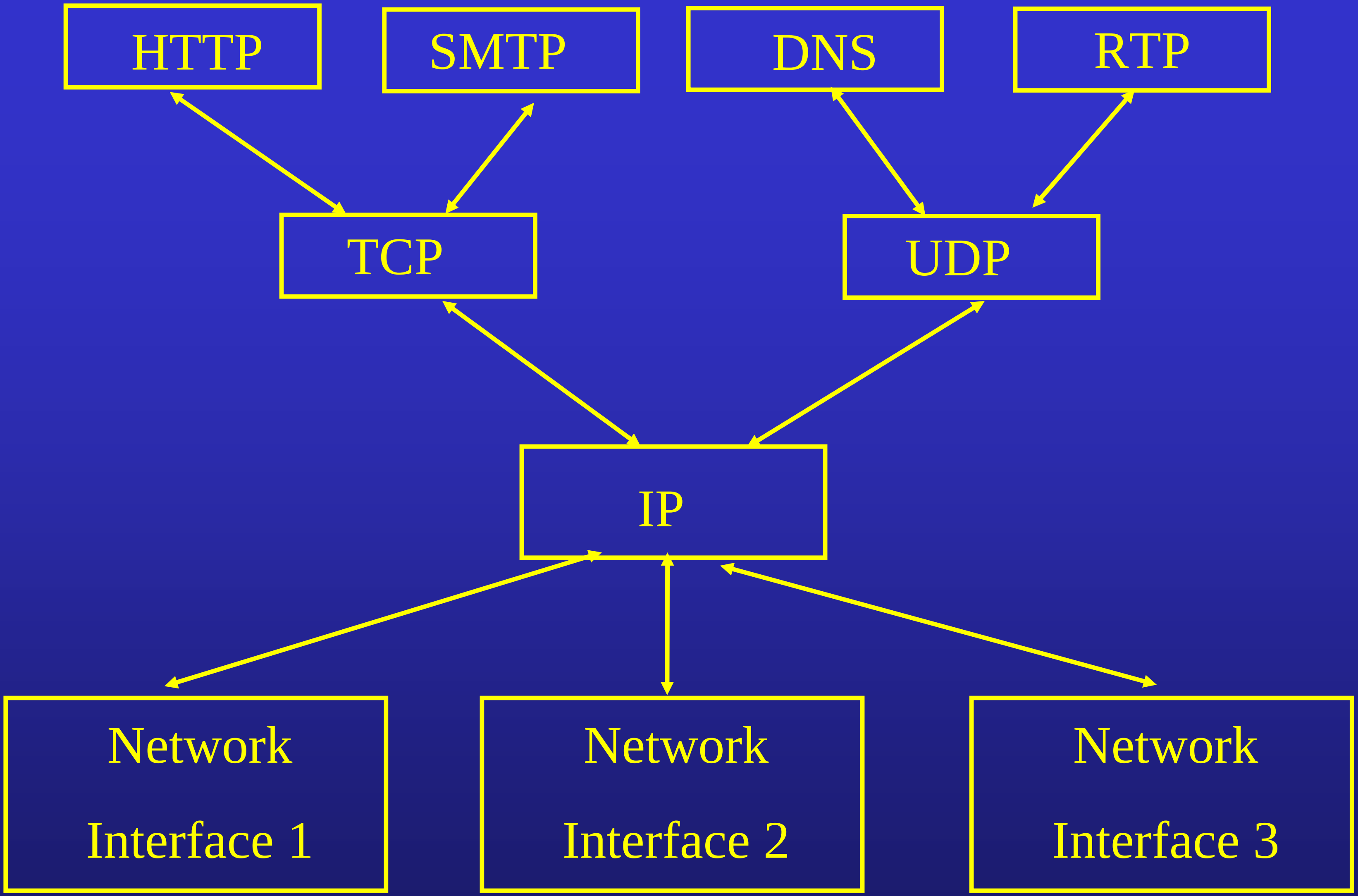
Not present in the model

Figure 1-21. The TCP/IP reference model.

TCP/IP Architectural Model



DCC 6th Ed., W. Stallings Figure 1.9



Leon-Garcia & Widjaja: *Communication Networks*

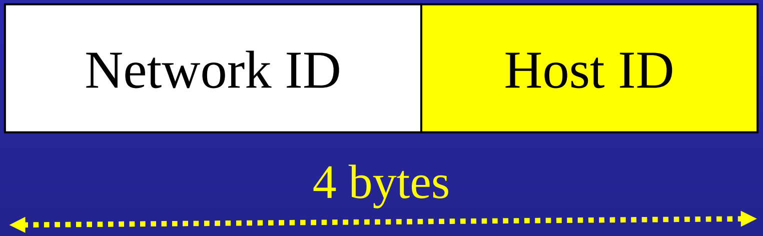
Figure 2.12

IP

- Currently IP provides *best-effort service.*
  - – packets may be lost (i.e., IP is unreliable).
- General IP design philosophy
  - – Keep internal operations simple by relegating complex functions to the **edge** of the subnet.
  - – IP can operate over any network
  - – This design allows IP to scale!!!
  - – *The end-to-end mechanisms are responsible for recovery of packet losses and congestion control.*

IPv4

- Uses *hierarchical address space* with location information embedded in the structure.



Network ID

Host ID

4 bytes

- IP address is usually expressed in *dotted-decimal notation* (e.g., 128.100.11.56).

Applications and Layered Architectures

[LG&W pp.43-49]

- In the 1970’s vendor companies (IBM and DEC) developed *proprietary networks* with the common feature of grouping communication functions into related and manageable sets called **layers**.

network architecture :: a set of protocols that specify how every layer is to function.

Advantages of Layering Design

- Provides an abstraction for functional locality.
- Simplifies the design process.
- Led to flexibility in modifying and developing network architectures.
- Accommodates incremental changes.

Layering Examples

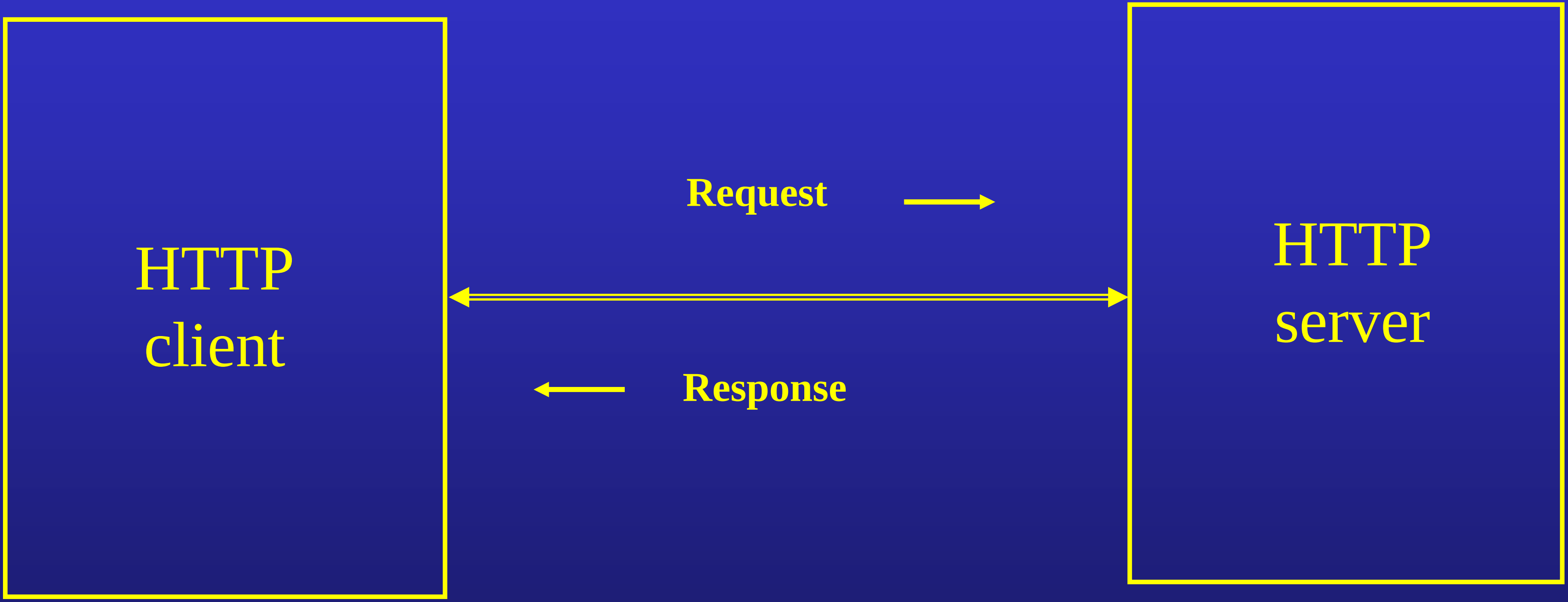
Client/server relationship

- Server process waits for incoming requests by listening to a **port**.
- Client process makes *requests* as required.
- Server process provides *responses* to these requests.
- The server process usually runs in the background as a **daemon** **(e.g. httpd is the server daemon for HTTP).**

HTTP Example

- HTTP (HyperText Transfer Protocol) specifies rules by which the client and the server interact so as to retrieve a document.
- The protocol assumes the client and the server can exchange messages directly
- The client software needs to set up a two-way connection prior to the HTTP request.

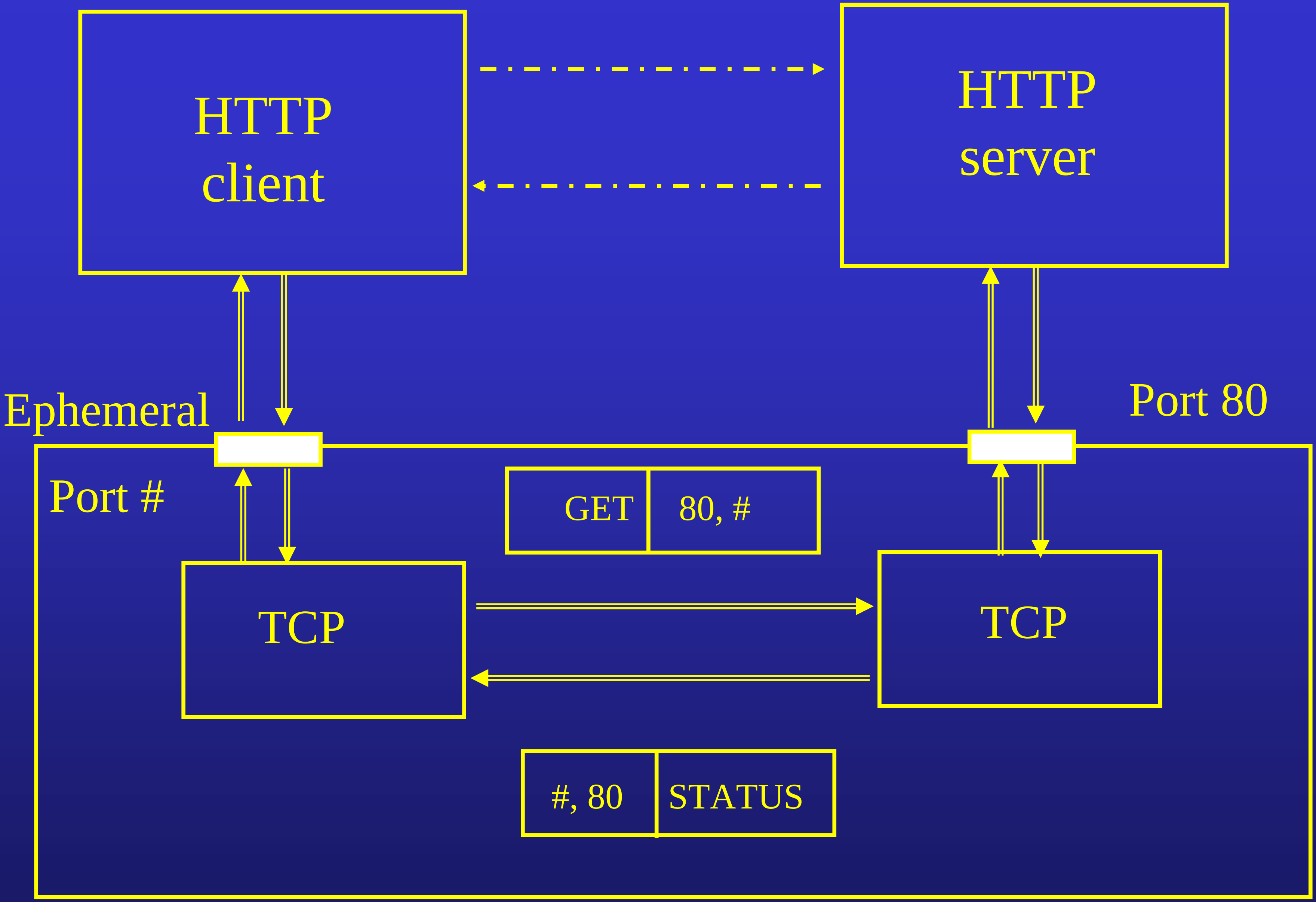
HTTP Client/Server Interaction



Copyright ©2000 The McGraw Hill Companies

Leon-Garcia & Widjaja: *Communication Networks*

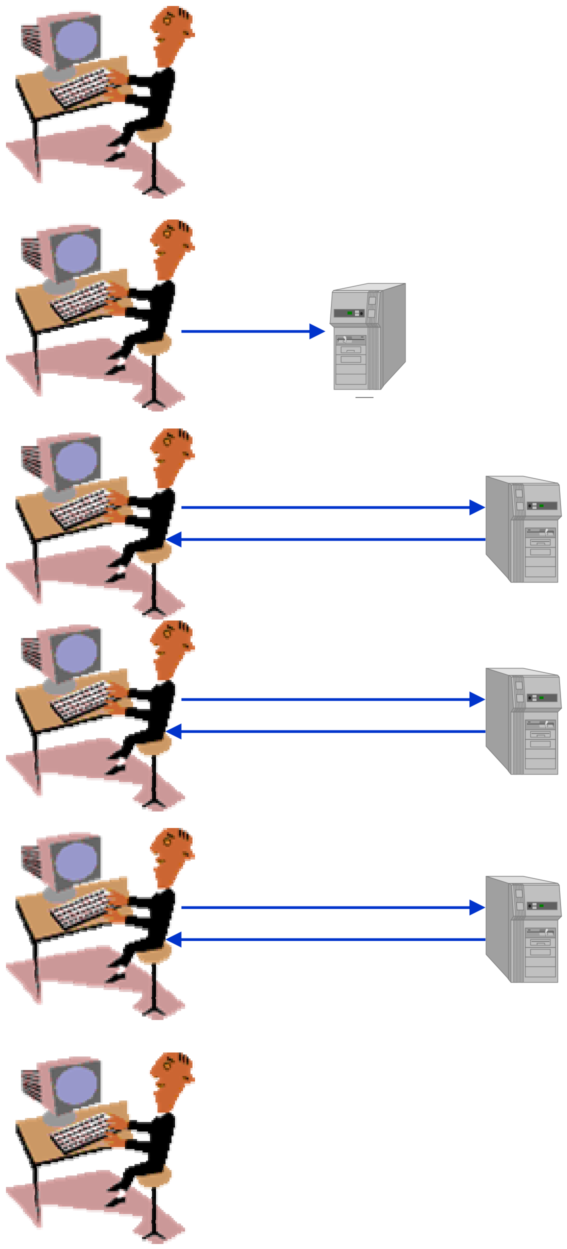
Figure 2.1



Leon-Garcia & Widjaja: *Communication Networks*

Figure 2.2

Retrieving a Web Page



1. The user clicks on a link to indicate which document is to be retrieved.

2. The browser must determine the address that contains the document. It does this by sending a query to its local name server.

3. Once the address is known the browser establishes a connection to the specified machine, usually a TCP connection. In order for the connection to be successful, the specified machine must be ready to accept TCP connections.

4. The browser runs a client version of HTTP, which issues a request specifying both the name of the document and the possible document formats it can handle.

5. The machine that contains the requested document runs a server version of HTTP. It reacts to the HTTP request by sending an HTTP response which contains the desired document in the appropriate format.

6. The TCP connection is then closed and the user may view the document.

Figure 1.4

Leon-Garcia & Widjaja: *Communication Networks*

Retrieving a document from the Web

1. User selects document
2. Network software of client locates the server host and establishes a two-way connection.
3. HTTP client sends message requesting document.
4. HTTP daemon listening on TCP port 80 interprets a message.
5. HTTP daemon send a result code and a description of the information that the client will receive

```
GET  /infocom/index.html HTTP 1.0
```

```
HTTP/1.1 200 OK
Server: Apache/1.3.23 (Unix)
Content-Length:  414
Content-Type:     text-html
```

Retrieving a document from the Web

6. HTTP daemon reads the file and sends the requested file through the TCP port.

7. Text is displayed by client browser, which interprets the HTML format.

8. HTTP daemon disconnects the connection after the connection is idle for some timeout period.

```
<html>
<head>
<title>Infocom ’99</title>
<font face =“Arial”>The Future Now
     …      </font>
```

DNS Query and Response

1. Application requests name to address translation.
2. Resolver composes query message.
3. Resolver send UDP datagram encapsulating the query message.
4. DNS server looks up address and prepares response.
5. DNS sends UDP datagram encapsulating the response message.

Header: OPCODE=SQUERY

Question:

QNAME= tesla.comm.toronto.edu.,
QCLASS=IN, QTYPE=A

HEADER: OPCODE=SQUERY, RESPONSE
AA

Question: QNAME=

Tesla.comm.toronto.edu.,
QCLASS=IN, QTYPE=A

Answer: telsa.cmm.toronto.edu.

86400 IN A 128.100.11.56