# COMPUTER COMMUNICATION NETWORKS (UE22EC351A)

Department of Electronics and Communication Engineering

# COMPUTER COMMUNICATION NETWORKS (UE22EC351A)

## UNIT 1: INTERNET ARCHITECTURE AND APPLICATIONS –
## Class 14 – Web & HTTP – Message Formats

Department of Electronics and Communication Engineering

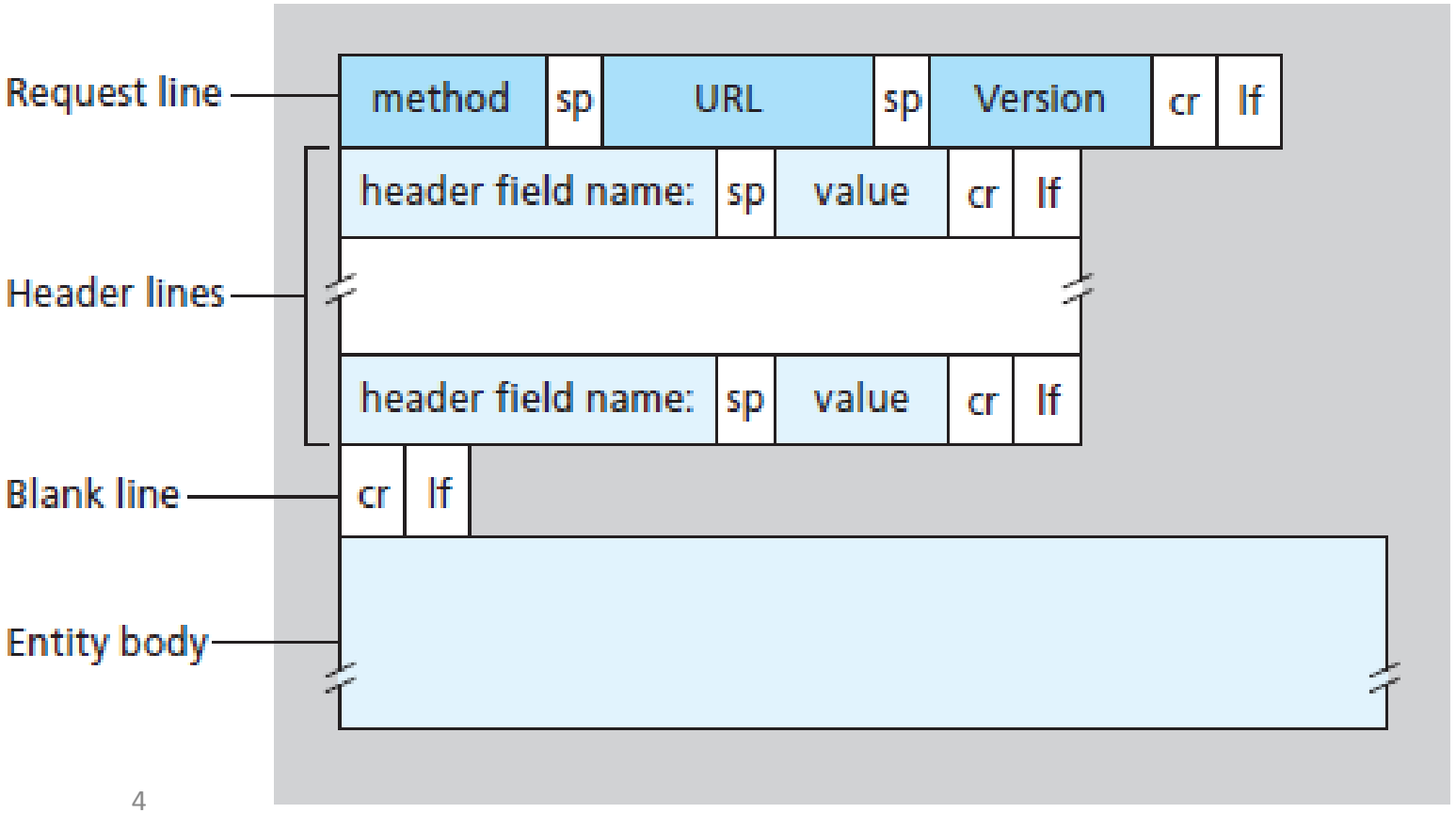
# Unit 1 – Class 14 - Web and HTTP: Message format

- Typical HTTP Request message

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message
  - Entity body is empty (download)  or non-empty (upload)



# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Request line
- The first line of an HTTP request message is called the request line;
- the subsequent lines are called the header lines.
- The request line has three fields: the method field, the URL field, and the HTTP version field.
- The method field can take on several different values, including GET, POST, HEAD, PUT, and DELETE.
- The great majority of HTTP request messages use the GET method. The GET method is used when the browser requests an object, with the requested object identified in the URL field.

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Header line
- Now let's look at the header lines in the example.
- The header line Host: www.someschool.edu specifies the host on which the object resides.
- You might think that this header line is unnecessary, as there is already a TCP connection in place to the host. But, the information provided by the host header line is required by Web proxy caches.
- By including the Connection: close header line, the browser is telling the server that it doesn't want to bother with persistent connections; it wants the server to close the connection after sending the requested object.

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Header line
- The User-agent: header line specifies the user agent, that is, the browser type that is making the request to the server.
- Here the user agent is Mozilla/5.0, a Firefox browser.
- This header line is useful because the server can actually send different versions of the same object to different types of user agents. (Each of the versions is addressed by the same URL.)

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Header line

- Finally, the Accept-language: header indicates that the user prefers to receive a French version of the object, if such an object exists on the server; otherwise, the server should send its default version.

- The Accept-language: header is just one of many content negotiation headers available in HTTP.

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Header line
- You may have noticed, however, that after the header lines (and the additional carriage return and line feed) there is an "entity body."
- The entity body is empty with the GET method, but is used with the POST method.
- An HTTP client often uses the POST method when the user fills out a form—for example, when a user provides search words to a search engine.

# Unit 1 – Class 14 - Web and HTTP: Message format

- HTTP Request message – Header line
- With a POST message, the user is still requesting a Web page from the server, but the specific contents of the Web page depend on what the user entered into the form fields. If the value of the method field is POST, then the entity body contains what the user entered into the form fields.

# Unit 1 – Class 14 - Web and HTTP: Message format

- Example-GET message:
  - Request webpage www.gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file.html

```
> Frame 921: 469 bytes on wire (3752 bits), 469 bytes captured (3752 bits) on interface 0
> Ethernet II, Src: HewlettP_7a:62:9d (6c:c2:17:7a:62:9d), Dst: HewlettP_84:19:80 (d4:c9:ef:84:19:80)
> Internet Protocol Version 4, Src: 172.16.175.59 (172.16.175.59), Dst: gaia.cs.umass.edu (128.119.245.12)
> Transmission Control Protocol, Src Port: 56506, Dst Port: 80, Seq: 1, Ack: 1, Len: 415
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate, sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Example-GET message (contd.):
  - Inspecting the raw data of the TCP segment

```
f5 0c dc b7 00 50 1c 15  d2 d7 6c d8 fa eb 50 18   .....P.. ..l...P.
fd b8 b7 19 00 00 47 45  54 20 2f 77 69 72 65 73   ......GE T /wires
68 61 72 6b 2d 6c 61 62  73 2f 68 74 74 70 2d 77   hark-lab s/http-w
69 72 65 73 68 61 72 6b  2d 66 69 6c 65 31 2e 68   ireshark -file1.h
74 6d 6c 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   tml HTTP /1.1..Ho
73 74 3a 20 67 61 69 61  2e 63 73 2e 75 6d 61 73   st: gaia .cs.umas
73 2e 65 64 75 0d 0a 43  6f 6e 6e 65 63 74 69 6f   s.edu..C onnectio
6e 3a 20 6b 65 65 70 2d  61 6c 69 76 65 0d 0a 55   n: keep- alive..U
70 67 72 61 64 65 2d 49  6e 73 65 63 75 72 65 2d   pgrade-I nsecure-
52 65 71 75 65 73 74 73  3a 20 31 0d 0a 55 73 65   Requests : 1..Use
72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
2f 35 2e 30 20 28 57 69  6e 64 6f 77 73 20 4e 54   /5.0 (Wi ndows NT
20 31 30 2e 30 3b 20 57  4f 57 36 34 29 20 41 70    10.0; W OW64) Ap
70 6c 65 57 65 62 4b 69  74 2f 35 33 37 2e 33 36   pleWebKi t/537.36
20 28 4b 48 54 4d 4c 2c  20 6c 69 6b 65 20 47 65    (KHTML,  like Ge
63 6b 6f 29 20 43 68 72  6f 6d 65 2f 35 35 2e 30   cko) Chr ome/55.0
2e 32 38 38 33 2e 38 37  20 53 61 66 61 72 69 2f   .2883.87  Safari/
35 33 37 2e 33 36 0d 0a  41 63 63 65 70 74 3a 20   537.36.. Accept:
74 65 78 74 2f 68 74 6d  6c 2c 61 70 70 6c 69 63   text/htm l,applic
61 74 69 6f 6e 2f 78 68  74 6d 6c 2b 78 6d 6c 2c   ation/xh tml+xml,
61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 6d 6c 3b   applicat ion/xml;
71 3d 30 2e 39 2c 69 6d  61 67 65 2f 77 65 62 70   q=0.9,im age/webp
2c 2a 2f 2a 3b 71 3d 30  2e 38 0d 0a 41 63 63 65   ,*/*;q=0 .8..Acce
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Typical HTTP Response Message

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 18 Aug 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Typical HTTP Response Message
- Let’s take a careful look at this response message.
- It has three sections: an initial status line, six header lines, and then the entity body.
- The entity body is the meat of the message—it contains the requested object itself (represented by data data data data data
- The status line has three fields: the protocol version field, a status code, and a corresponding status message.

- Typical HTTP Response Message

- In this example, the status line indicates that the server is using HTTP/1.1 and that everything is OK (that is, the server has found, and is sending, the requested object).
- Now let's look at the header lines. The server uses the Connection: close header line to tell the client that it is going to close the TCP connection after sending the message.
- The Date: header line indicates the time and date when the HTTP response was created and sent by the server.

- Typical HTTP Response Message

- Note that this is not the time when the object was created or last modified; it is the time when the server retrieves the object from its file system, inserts the object into the response message, and sends the response message.
- The Server: header line indicates that the message was generated by an Apache Web server; it is analogous to the User-agent: header line in the HTTP request message.

# Unit 1 – Class 14 - Web and HTTP: Message format

- Typical HTTP Response Message
- The Last-Modified: header line indicates the time and date when the object was created or last modified.
- The Last-Modified: header, which we will soon cover in more detail, is critical for object caching, both in the local client and in network cache servers (also known as proxy servers).
- The Content-Length: header line indicates the number of bytes in the object being sent. The Content-Type: header line indicates that the object in the entity body is HTML text. (The object type is officially indicated by the Content-Type: header and not by the file extension.)

# Unit 1 – Class 14 - Web and HTTP: Message format

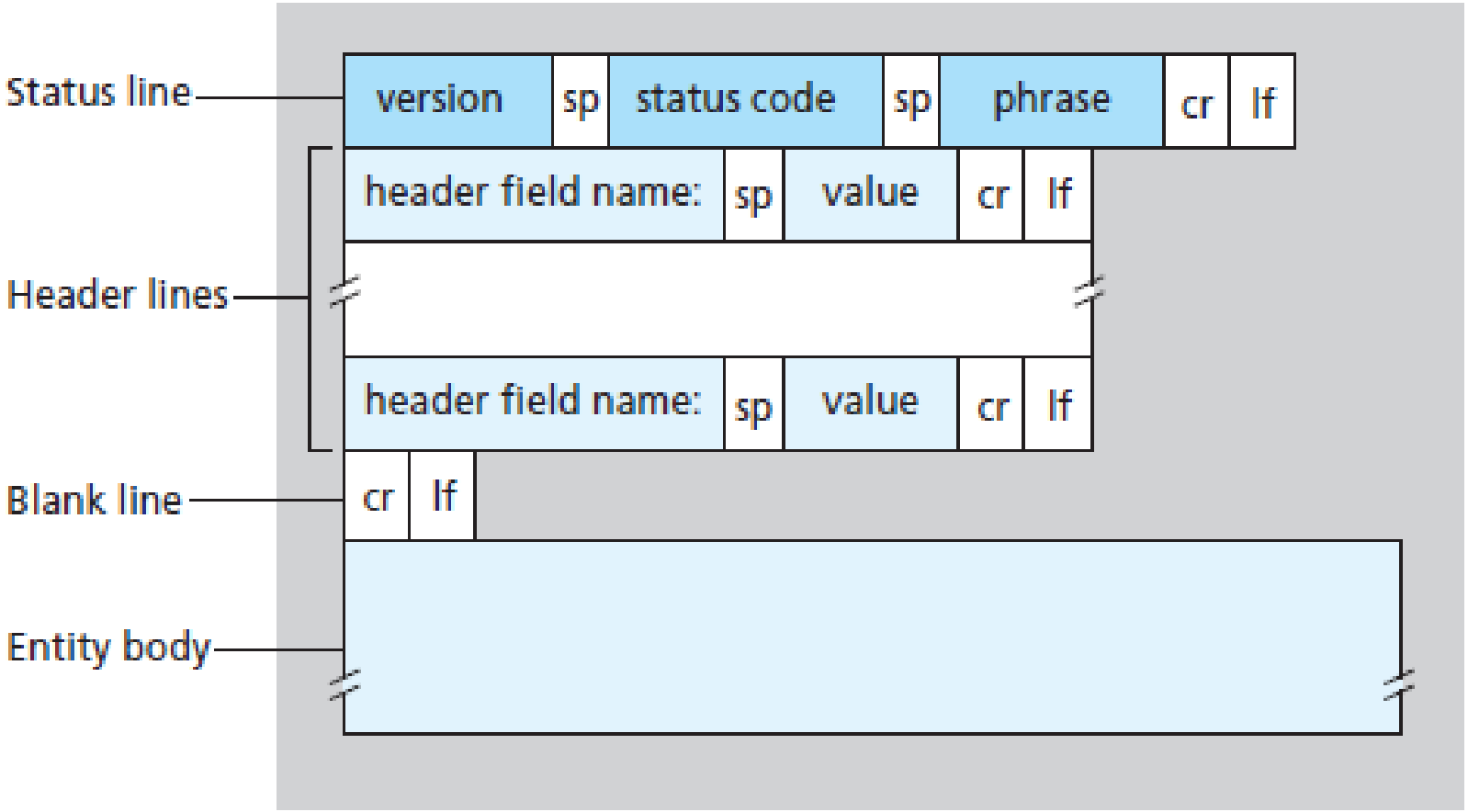
- Common Status Codes
- 200 OK – Request Succeeded & the information is returned in the response
- 301 Moved Permanently: Requested object has been permanently moved; the new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL
- 400 Bad Request: This is a generic error code indicating that the request could not be understood by the server.
- 404 Not Found: The requested document does not exist on this server

# Unit 1 – Class 14 - Web and HTTP: Message format

- Common Status Codes
- 505 HTTP Version Not Supported: The requested HTTP protocol version is not supported by the server.

# Unit 1 – Class 14 - Web and HTTP: Message format

- Web server  sends the response message which could have the requested object



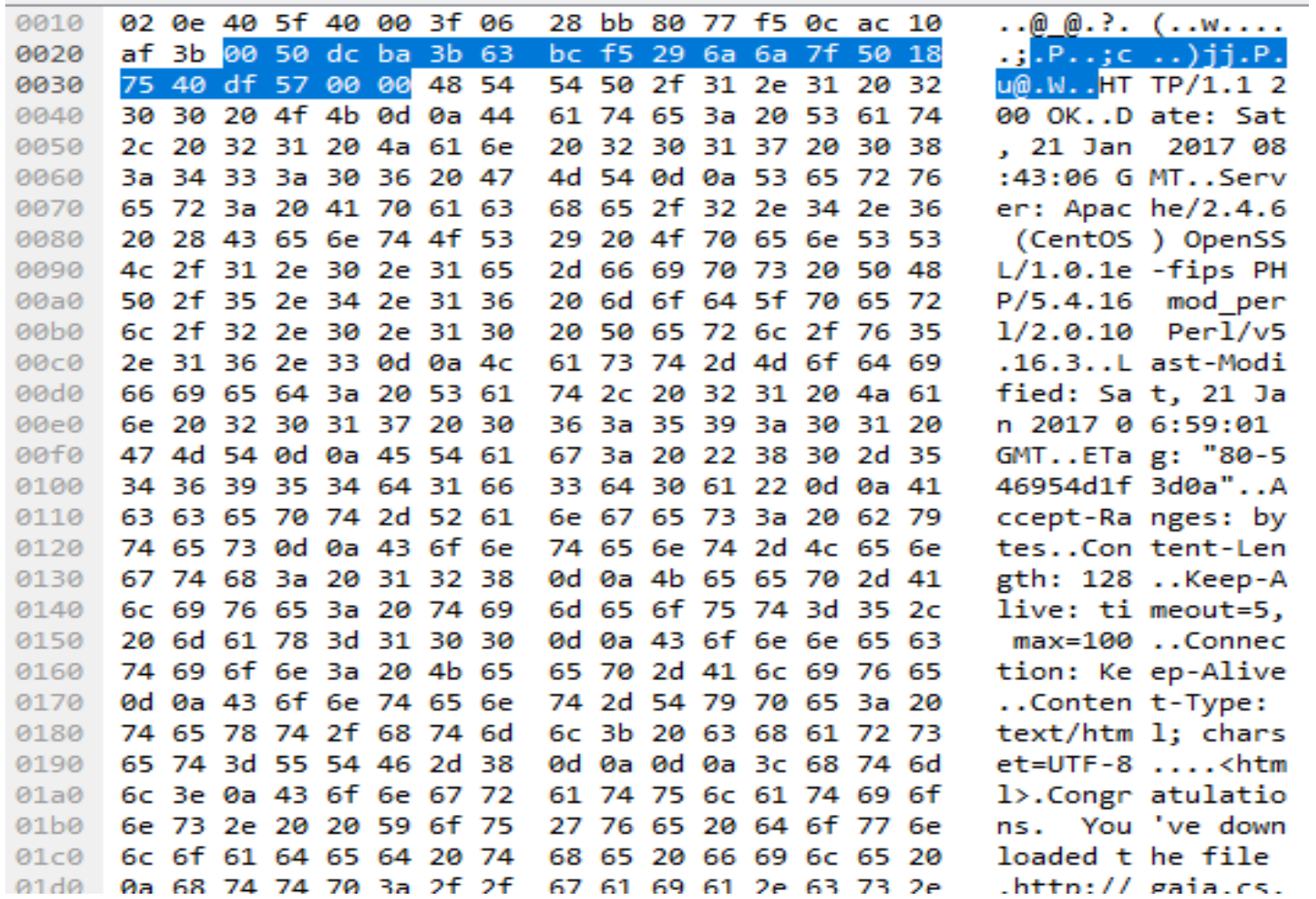
# Unit 1 – Class 14 - Web and HTTP: Message format

- Example-HTTP response message:

```
Internet Protocol Version 4, Src: gaia.cs.umass.edu (128.119.245.12), Dst: 172.16.175.59 (172.16.175.59)
Transmission Control Protocol, Src Port: 80, Dst Port: 56506, Seq: 1, Ack: 416, Len: 486
Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Sat, 21 Jan 2017 08:43:06 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3\r\n
    Last-Modified: Sat, 21 Jan 2017 06:59:01 GMT\r\n
    ETag: "80-546954d1f3d0a"\r\n
    Accept-Ranges: bytes\r\n
  > Content-Length: 128\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.555626000 seconds]
    [Request in frame: 921]
    File Data: 128 bytes
Line-based text data: text/html
    <html>\n
    Congratulations.  You've downloaded the file \n
    http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html!\n
    </html>\n
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Example-HTTP response (contd.):



```
0010  02 0e 40 5f 40 00 3f 06  28 bb 80 77 f5 0c ac 10   ..@_@.?. (..w....
0020  af 3b 00 50 dc ba 3b 63  bc f5 29 6a 6a 7f 50 18   .;.P..;c ..)jj.P.
0030  75 40 df 57 00 00 48 54  54 50 2f 31 2e 31 20 32   u@.W..HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 53 61 74   00 OK..D ate: Sat
0050  2c 20 32 31 20 4a 61 6e  20 32 30 31 37 20 30 38   , 21 Jan  2017 08
0060  3a 34 33 3a 30 36 20 47  4d 54 0d 0a 53 65 72 76   :43:06 G MT..Serv
0070  65 72 3a 20 41 70 61 63  68 65 2f 32 2e 34 2e 36   er: Apac he/2.4.6
0080  20 28 43 65 6e 74 4f 53  29 20 4f 70 65 6e 53 53    (CentOS ) OpenSS
0090  4c 2f 31 2e 30 2e 31 65  2d 66 69 70 73 20 50 48   L/1.0.1e -fips PH
00a0  50 2f 35 2e 34 2e 31 36  20 6d 6f 64 5f 70 65 72   P/5.4.16  mod_per
00b0  6c 2f 32 2e 30 2e 31 30  20 50 65 72 6c 2f 76 35   l/2.0.10  Perl/v5
00c0  2e 31 36 2e 33 0d 0a 4c  61 73 74 2d 4d 6f 64 69   .16.3..L ast-Modi
00d0  66 69 65 64 3a 20 53 61  74 2c 20 32 31 20 4a 61   fied: Sa t, 21 Ja
00e0  6e 20 32 30 31 37 20 30  36 3a 35 39 3a 30 31 20   n 2017 0 6:59:01
00f0  47 4d 54 0d 0a 45 54 61  67 3a 20 22 38 30 2d 35   GMT..ETa g: "80-5
0100  34 36 39 35 34 64 31 66  33 64 30 61 22 0d 0a 41   46954d1f 3d0a"..A
0110  63 63 65 70 74 2d 52 61  6e 67 65 73 3a 20 62 79   ccept-Ra nges: by
0120  74 65 73 0d 0a 43 6f 6e  74 65 6e 74 2d 4c 65 6e   tes..Con tent-Len
0130  67 74 68 3a 20 31 32 38  0d 0a 4b 65 65 70 2d 41   gth: 128 ..Keep-A
0140  6c 69 76 65 3a 20 74 69  6d 65 6f 75 74 3d 35 2c   live: ti meout=5,
0150  20 6d 61 78 3d 31 30 30  0d 0a 43 6f 6e 6e 65 63    max=100 ..Connec
0160  74 69 6f 6e 3a 20 4b 65  65 70 2d 41 6c 69 76 65   tion: Ke ep-Alive
0170  0d 0a 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ..Conten t-Type:
0180  74 65 78 74 2f 68 74 6d  6c 3b 20 63 68 61 72 73   text/htm l; chars
0190  65 74 3d 55 54 46 2d 38  0d 0a 0d 0a 3c 68 74 6d   et=UTF-8 ....<htm
01a0  6c 3e 0a 43 6f 6e 67 72  61 74 75 6c 61 74 69 6f   l>.Congr atulatio
01b0  6e 73 2e 20 20 59 6f 75  27 76 65 20 64 6f 77 6e   ns.  You 've down
01c0  6c 6f 61 64 65 64 20 74  68 65 20 66 69 6c 65 20   loaded t he file
01d0  0a 68 74 74 70 3a 2f 2f  67 61 69 61 2e 63 73 2e   .http:// gaia.cs.
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Consider the HTTP messages exchanged between the client (web browser) and the web server as given below.

```
GET /ethereal-labs/lab2-3.html HTTP/1.1\r\n
Host: gaia.cs.umass.edu\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0
Accept-Language: en-us, en;q=0.50\r\n
Accept-Encoding: gzip, deflate, compress;q=0.9\r\n
Accept-Charset: ISO-8859-1, utf-8;q=0.66, *;q=0.66\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n

HTTP/1.1 200 OK\r\n
Date: Tue, 23 Sep 2003 05:37:02 GMT\r\n
Server: Apache/2.0.40 (Red Hat Linux)\r\n
Last-Modified: Tue, 23 Sep 2003 05:37:01 GMT\r\n
ETag: "1bff2-1194-96813940"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 4500\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
```

# Unit 1 – Class 14 - Web and HTTP: Message format

- Answer the following questions: (1 mark each)
- 1. What is the "method" used by the client?
- 2. What is name of the object requested by the client?
- 3. What is the name of the web server on which the object is to be located?
- 4. What is the type of TCP connection preferred by the client?
- 5. What are the status code and phrase returned by the web server?
- 6. What is the version of HTTP supported by the web server?
- 7. What is the time at which the HTTP message was generated by the web server?
- 8. When was the object last updated at the web server?
- 9. What is the size of object sent by the web server?
- 10. What is the server process used by the web-server?

# Unit 1 – Class 14 - Web and HTTP: Message format

- Solution –
- 1. GET
- 2. lab2-3.html
- 3. gaia.cs.umass.edu
- 4. Persistent (keep alive)
- 5. 200 and OK
- 6. HTTP 1.1
- 7. Tue, 23 Sep 2003 5:37:02 GMT
- 8. Tue, 23 Sep 2003 5:37:01 GMT
- 9. 4500 bytes
- 10. Apache /2.0.40

# THANK YOU

Department of Electronics and communication engineering