COMPUTER NETWORKS

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

COMPUTER NETWORKS

Application Layer

Sivaraman Eswaran Ph.D.

Department of Computer Science and Engineering

Unit – 2 Application Layer

HTTP Request Message

- two types of HTTP messages: *request*, *response*
- HTTP request message:
  - ASCII (human-readable format)

carriage return character

line-feed character

request line (GET, POST, HEAD commands)

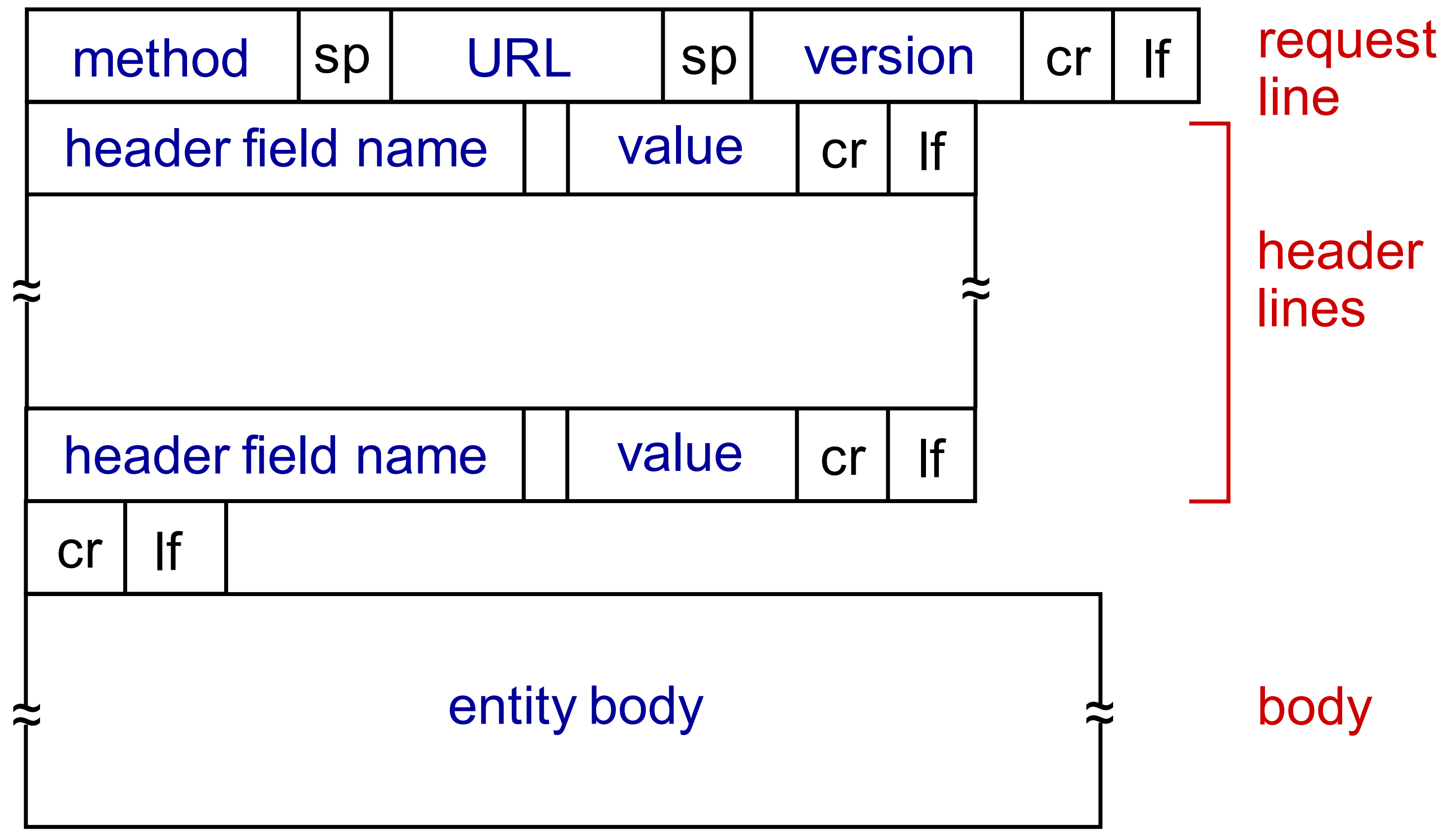
header lines

carriage return, line feed at start of line indicates end of header lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

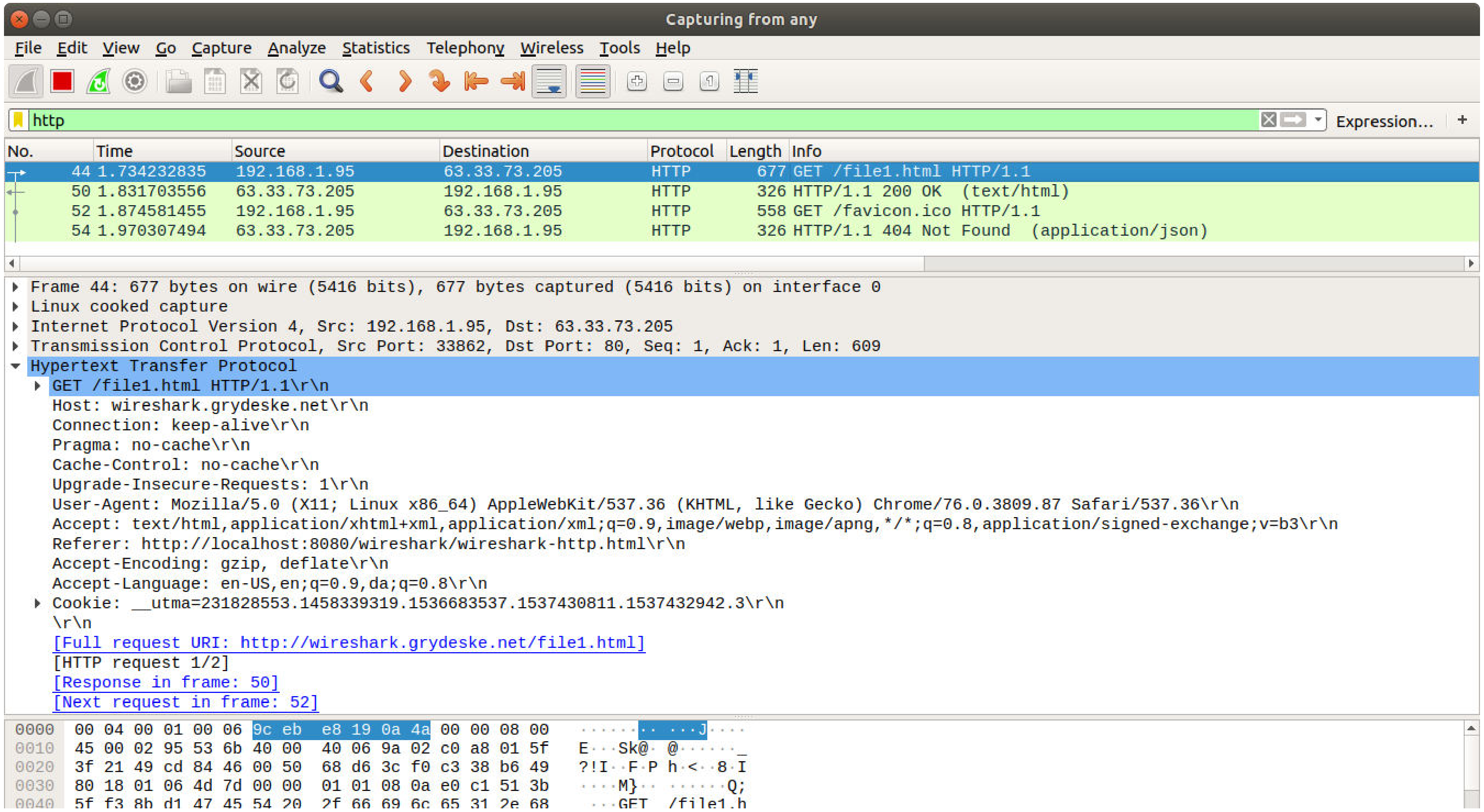
HTTP Request Message: General Format



HTTP specifications [RFC 1945; RFC 2616; RFC 7540]

COMPUTER NETWORKS

HTTP Request Message – Wireshark Capture


Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
http
```

```
No.   Time          Source          Destination     Protocol  Length  Info
   44 1.734232835   192.168.1.95    63.33.73.205    HTTP         677  GET /file1.html HTTP/1.1
   50 1.831703556   63.33.73.205    192.168.1.95    HTTP         326  HTTP/1.1 200 OK  (text/html)
   52 1.874581455   192.168.1.95    63.33.73.205    HTTP         558  GET /favicon.ico HTTP/1.1
   54 1.970307494   63.33.73.205    192.168.1.95    HTTP         326  HTTP/1.1 404 Not Found  (application/json)
```

```
▸ Frame 44: 677 bytes on wire (5416 bits), 677 bytes captured (5416 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.1.95, Dst: 63.33.73.205
▸ Transmission Control Protocol, Src Port: 33862, Dst Port: 80, Seq: 1, Ack: 1, Len: 609
▾ Hypertext Transfer Protocol
  ▸ GET /file1.html HTTP/1.1\r\n
    Host: wireshark.grydeske.net\r\n
    Connection: keep-alive\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.87 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
    Referer: http://localhost:8080/wireshark/wireshark-http.html\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9,da;q=0.8\r\n
  ▸ Cookie: __utma=231828553.1458339319.1536683537.1537430811.1537432942.3\r\n
    \r\n
    [Full request URI: http://wireshark.grydeske.net/file1.html]
    [HTTP request 1/2]
    [Response in frame: 50]
    [Next request in frame: 52]
```

```
0000  00 04 00 01 00 06 9c eb  e8 19 0a 4a 00 00 08 00   ········ ···J····
0010  45 00 02 95 53 6b 40 00  40 06 9a 02 c0 a8 01 5f   E···Sk@· @······_
0020  3f 21 49 cd 84 46 00 50  68 d6 3c f0 c3 38 b6 49   ?!I··F·P h·<··8·I
0030  80 18 01 06 4d 7d 00 00  01 01 08 0a e0 c1 51 3b   ····M}·· ······Q;
0040  5f f3 8b d1 47 45 54 20  2f 66 69 6c 65 31 2e 68   ····GET  /file1.h
```

Other HTTP Request Messages

POST method:

- web page often includes form input
- user input sent from client to server in entity body of HTTP POST request message

HEAD method:

- requests headers (only) that would be returned *if* specified URL were requested with an HTTP GET method.

GET method (for sending data to server):

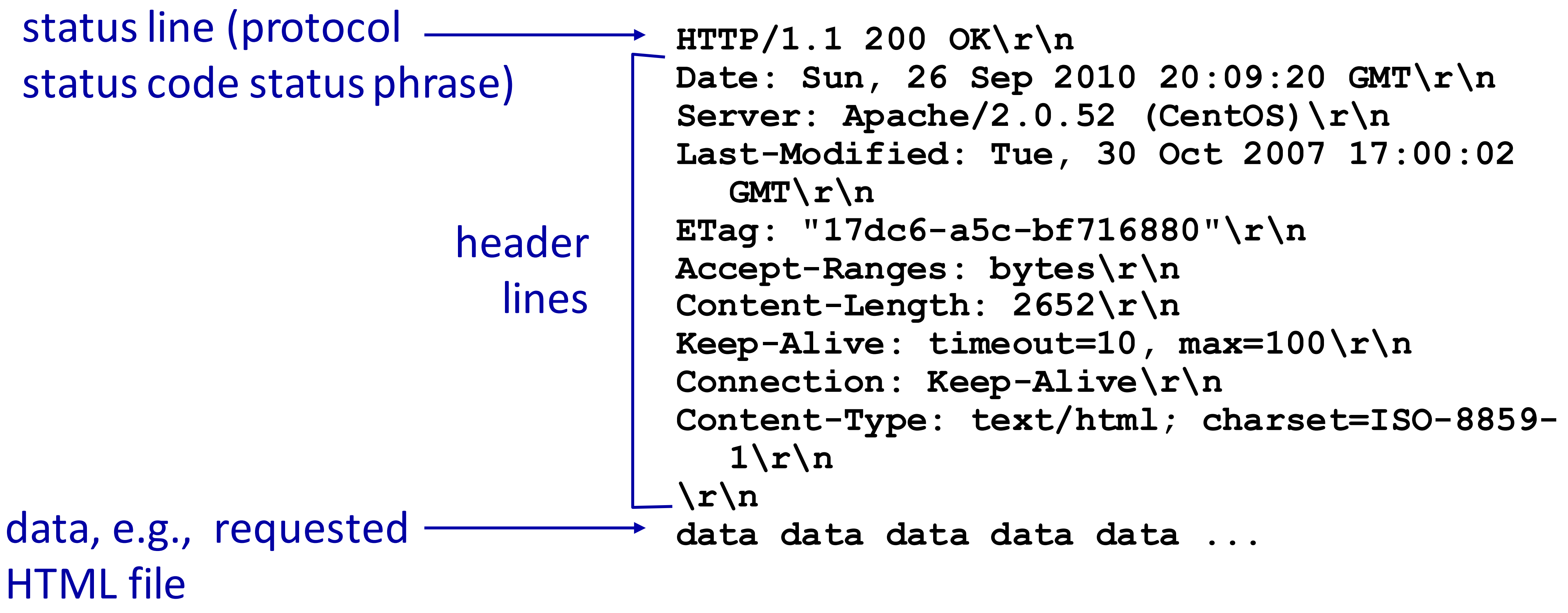
- include user data in URL field of HTTP GET request message (following a '?'):

```
www.somesite.com/animalsearch?monkeys&banana
```

PUT method:

- uploads new file (object) to server
- completely replaces file that exists at specified URL with content in entity body of POST HTTP request message

HTTP Response Message



status line (protocol status code status phrase)

header lines

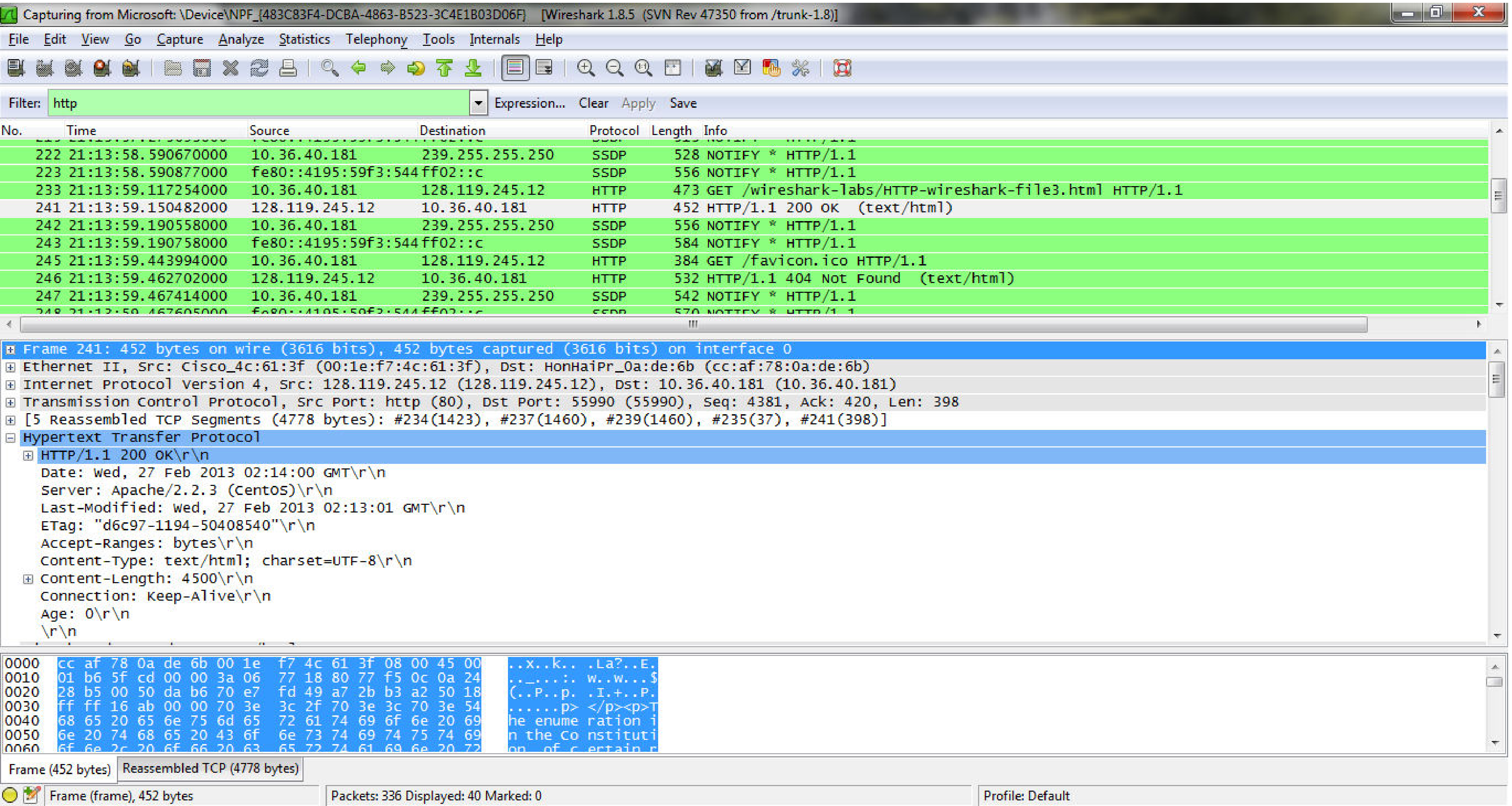
data, e.g., requested HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
   GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\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
\r\n
data data data data data ...
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

COMPUTER NETWORKS

HTTP Response Message – Wireshark Capture


Capturing from Microsoft: \Device\NPF_{483C83F4-DCBA-4863-B523-3C4E1B03D06F} [Wireshark 1.8.5 (SVN Rev 47350 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save

```
No.  Time                Source                  Destination      Protocol Length Info
 222 21:13:58.590670000  10.36.40.181            239.255.255.250  SSDP     528 NOTIFY * HTTP/1.1
 223 21:13:58.590877000  fe80::4195:59f3:544     ff02::c          SSDP     556 NOTIFY * HTTP/1.1
 233 21:13:59.117254000  10.36.40.181            128.119.245.12   HTTP     473 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
 241 21:13:59.150482000  128.119.245.12          10.36.40.181     HTTP     452 HTTP/1.1 200 OK  (text/html)
 242 21:13:59.190558000  10.36.40.181            239.255.255.250  SSDP     556 NOTIFY * HTTP/1.1
 243 21:13:59.190758000  fe80::4195:59f3:544     ff02::c          SSDP     584 NOTIFY * HTTP/1.1
 245 21:13:59.443994000  10.36.40.181            128.119.245.12   HTTP     384 GET /favicon.ico HTTP/1.1
 246 21:13:59.462702000  128.119.245.12          10.36.40.181     HTTP     532 HTTP/1.1 404 Not Found  (text/html)
 247 21:13:59.467414000  10.36.40.181            239.255.255.250  SSDP     542 NOTIFY * HTTP/1.1
```

```
⊞ Frame 241: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
⊞ Ethernet II, Src: Cisco_4c:61:3f (00:1e:f7:4c:61:3f), Dst: HonHaiPr_0a:de:6b (cc:af:78:0a:de:6b)
⊞ Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 10.36.40.181 (10.36.40.181)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 55990 (55990), Seq: 4381, Ack: 420, Len: 398
⊞ [5 Reassembled TCP Segments (4778 bytes): #234(1423), #237(1460), #239(1460), #235(37), #241(398)]
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Date: Wed, 27 Feb 2013 02:14:00 GMT\r\n
    Server: Apache/2.2.3 (CentOS)\r\n
    Last-Modified: Wed, 27 Feb 2013 02:13:01 GMT\r\n
    ETag: "d6c97-1194-50408540"\r\n
    Accept-Ranges: bytes\r\n
    Content-Type: text/html; charset=UTF-8\r\n
  ⊞ Content-Length: 4500\r\n
    Connection: Keep-Alive\r\n
    Age: 0\r\n
    \r\n
```

```
0000  cc af 78 0a de 6b 00 1e  f7 4c 61 3f 08 00 45 00   ..x..k.. .La?..E.
0010  01 b6 5f cd 00 00 3a 06  77 18 80 77 f5 0c 0a 24   .._...:. w..w...$
0020  28 b5 00 50 da b6 70 e7  fd 49 a7 2b b3 a2 50 18   (..P..p. .I.+..P.
0030  ff ff 16 ab 00 00 70 3e  3c 2f 70 3e 3c 70 3e 54   ......p> </p><p>T
0040  68 65 20 65 6e 75 6d 65  72 61 74 69 6f 6e 20 69   he enume ration i
0050  6e 20 74 68 65 20 43 6f  6e 73 74 69 74 75 74 69   n the Co nstituti
```

Frame (452 bytes)
Reassembled TCP (4778 bytes)
Frame (frame), 452 bytes
Packets: 336 Displayed: 40 Marked: 0
Profile: Default

HTTP Response Status Codes

- status code appears in 1st line in server-to-client response message.
- some sample codes:

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (in Location: field)

400 Bad Request

- request msg not understood by server

404 Not Found

- requested document not found on this server

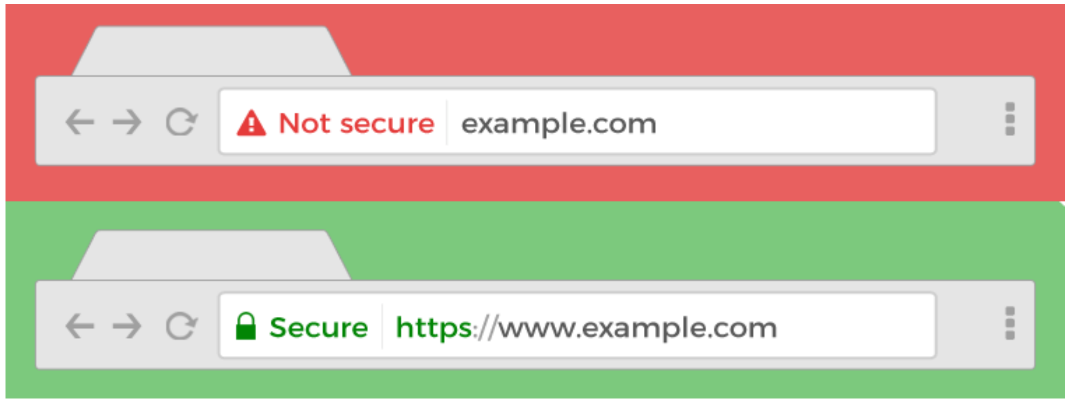
505 HTTP Version Not Supported

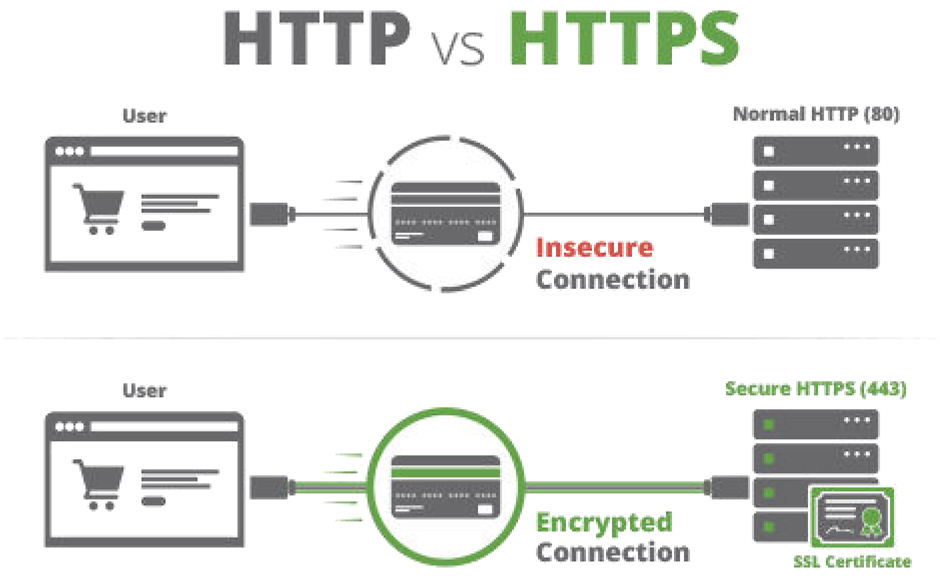
COMPUTER NETWORKS

Application Layer

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

HTTP vs HTTPS

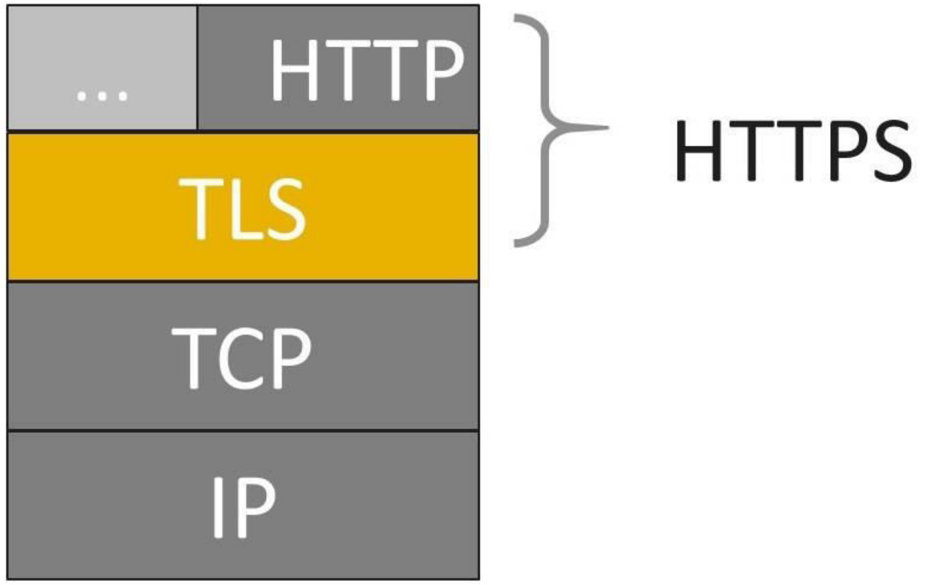




- HTTPS is HTTP with encryption – All communications between browser and server are encrypted (bi-directional).
- ‘S’ refers ‘Secure’ or HTTP over Secure Socket Layer.
- Uses TLS (SSL) to encrypt normal HTTP requests and responses.
- Attackers can't read the data crossing the wire and you know you are talking to the server you think you are talking too.

HTTP vs HTTPS (more)

- HTTP + TLS -> Encrypted
- Uses port no. 443 for data communication.
- HTTPS is based on public/private-key cryptography.
  - The public key is used for encryption
  - The secret private key is required for decryption.
- SSL certificate is a web server's digital certificate issued by a third party CA.
  - Create an encrypted connection and establish trust.
- Is my certificate SSL or TLS?



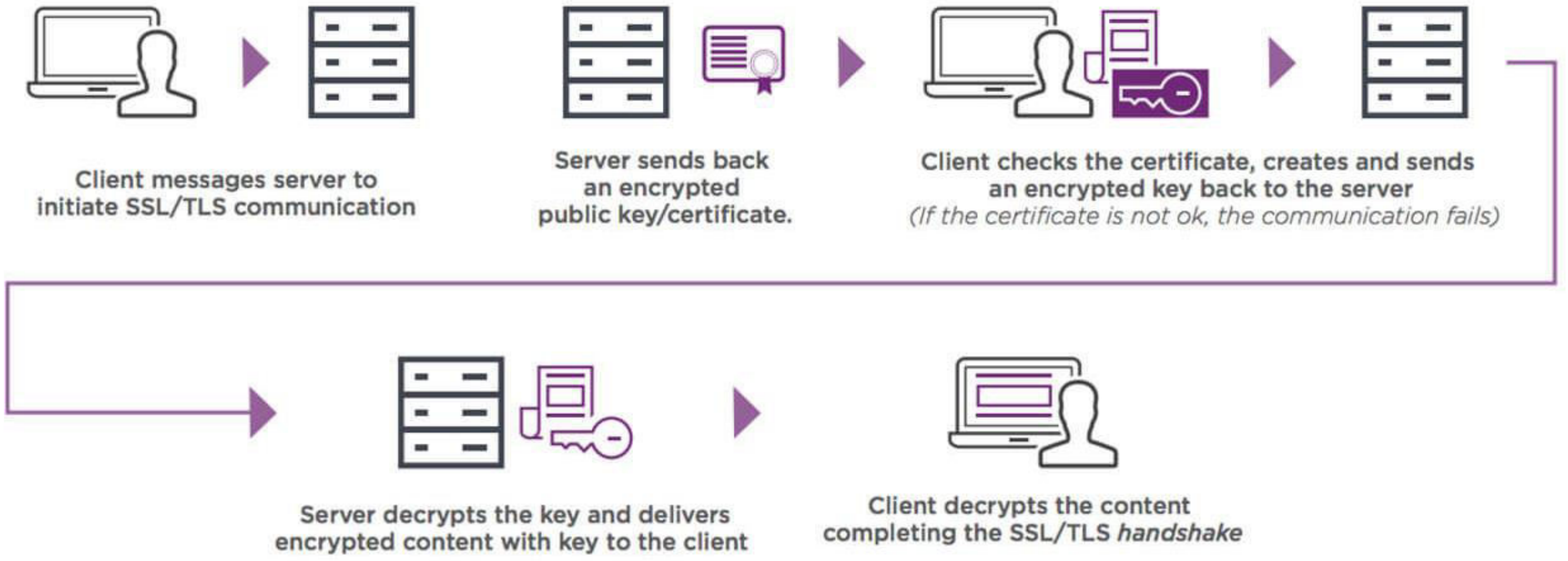
Any message encrypted with Bob's public key can be only decrypted with Bob's private key.

How does SSL works?

- Step 1: Browser requests secure pages (HTTPS) from a server.
- Step 2: Server sends its public key with its SSL certificate (digitally signed by a third party – CA).
- Step 3: On receipt of certificate, browser verifies issuer's digital signature. (green padlock key)
- Step 4: Browser creates a symmetric key (shared key), keeps one and gives a copy to server. Encrypts it using server's public key.
- Step 5: On receipt of encrypted secret key, decrypts it using its private key and gets browser's secret key.

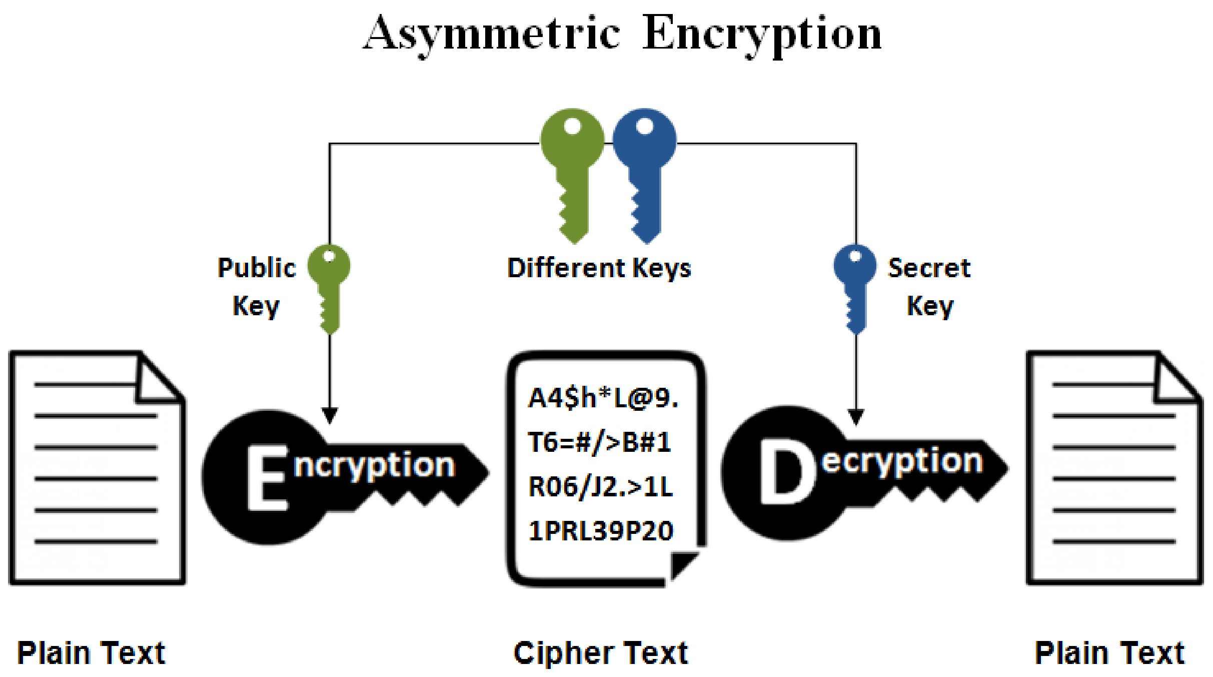
- Asymmetric and Symmetric key algorithms work together.
- Asymmetric key algorithm – verify identity of the owner & its public key -> Establish trust.
- Once connection is established, Symmetric key algorithm is used to encrypt and decrypt the traffic.

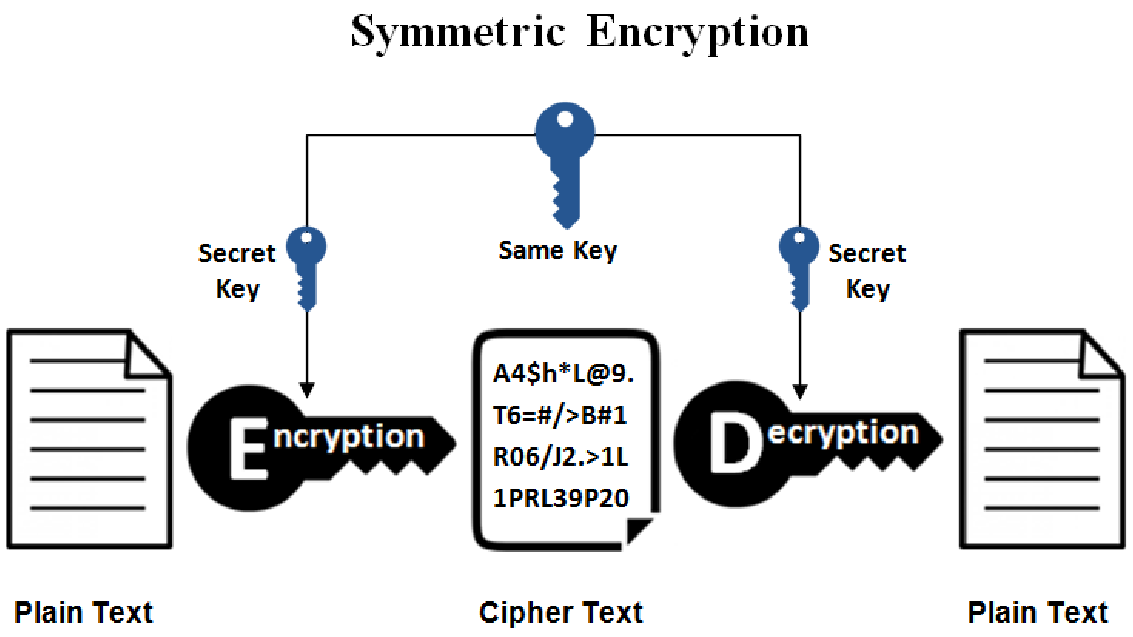
How does SSL works?


Client messages server to initiate SSL/TLS communication
Server sends back an encrypted public key/certificate.
Client checks the certificate, creates and sends an encrypted key back to the server
(If the certificate is not ok, the communication fails)
Server decrypts the key and delivers encrypted content with key to the client
Client decrypts the content completing the SSL/TLS *handshake*

Benefits of HTTPS over HTTP using SSL Certificates

- Stronger Google ranking.
- Updated browser labels.
- Improved security.
- Increased customer confidence / safer experience.
- Build customer trust and improve conversions.


Asymmetric Encryption
Public Key
Different Keys
Secret Key
Encryption
A4$h*L@9.
T6=#/>B#1
R06/J2.>1L
1PRL39P20
Decryption
Plain Text
Cipher Text
Plain Text


Symmetric Encryption
Secret Key
Same Key
Secret Key
Encryption
A4$h*L@9.
T6=#/>B#1
R06/J2.>1L
1PRL39P20
Decryption
Plain Text
Cipher Text
Plain Text

PES
UNIVERSITY
ONLINE

THANK YOU

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

sivaramane@pes.edu

+91 80 6666 3333 Extn 834