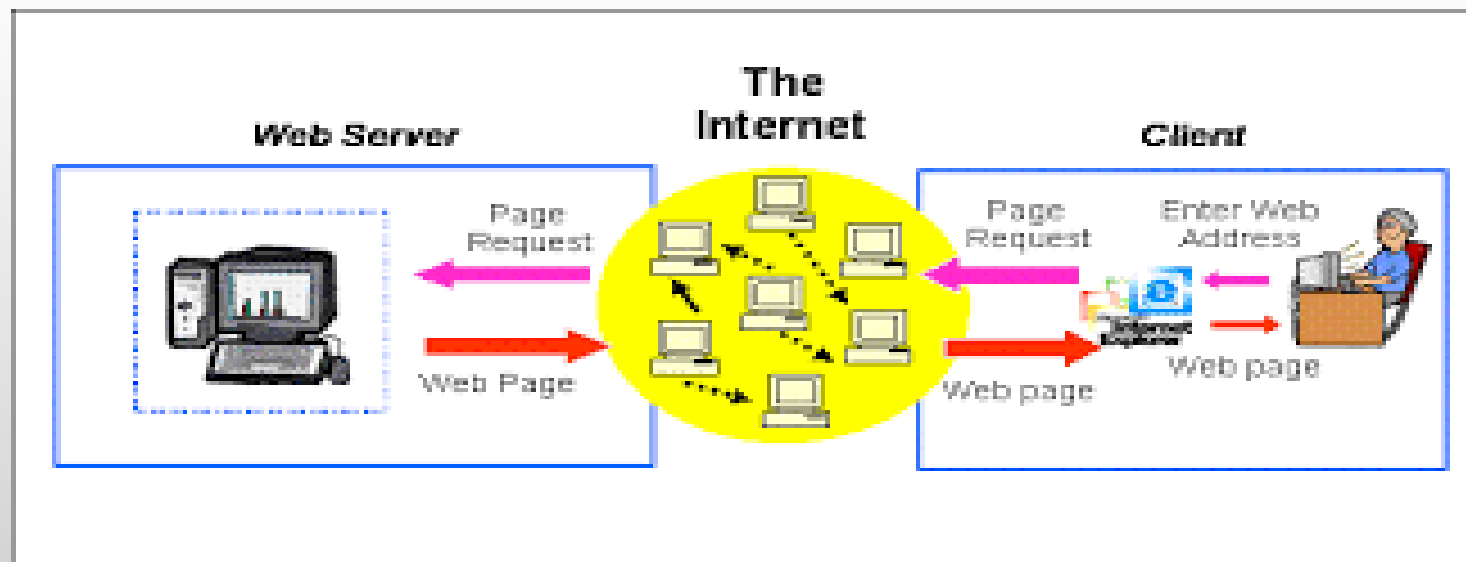


# Introduction to Basic Concepts: Protocols



# Protocols – an Example

- Protocols define sets of rules governing how two or more objects should interact with one another. Protocols serve as specifications rather than implementations of a piece of technology.
- An example of an relevant protocol is the [HyperText Transport Protocol \(HTTP\)](#), which provides an specification (i.e., a vocabulary) that allows client applications to request resources from [Web](#) servers, and Web servers to respond to these requests.



We will talk more about the Client-Server Architecture as we move forward.

Sandra Sampaio

Figure source: [teaching.shu.ac.uk](http://teaching.shu.ac.uk)



- Examples of “verbs” used by client applications and Web servers when communicating via the HTTP protocol over the Internet are shown below.

HTTP verbs	Description
HEAD	Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.
GET	Requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect.
POST	Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.
OPTIONS	Returns a list of the commands supported by this particular server.
DELETE	It is used to delete a resource. It may return the a representation of the removed resource.

This list is not complete. For a complete list and for more information about each Representational State Transfer (REST) HTTP command, refer to <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>



# More about the HTTP Protocol: Statelessness

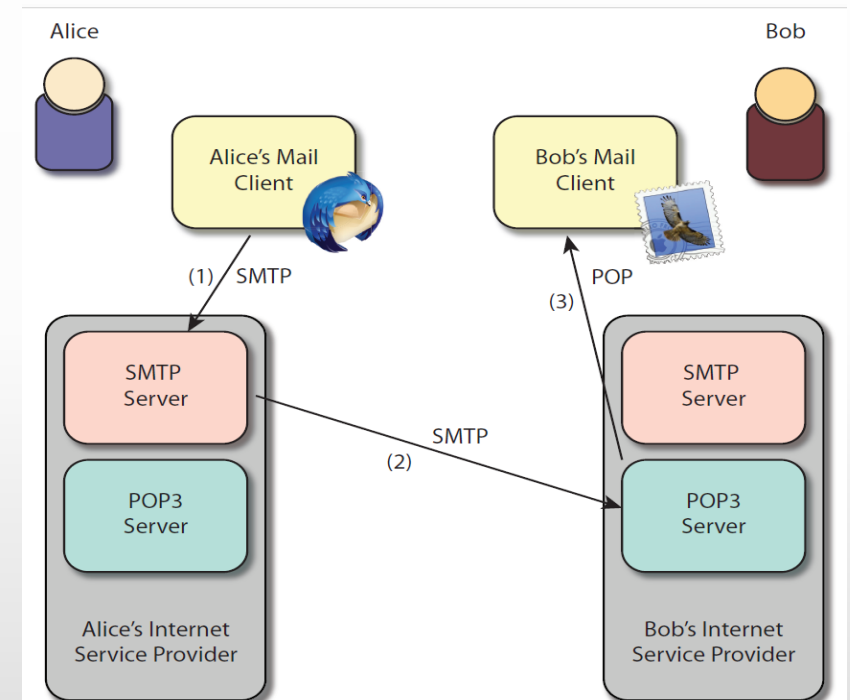
- By following the HTTP protocol, Web servers are said to be **stateless**, meaning that once a request from a client application is fulfilled, the Web **server disconnects from the client and “forgets” that the client ever connected.**
- The stateless nature of the communication between client and server allows the system to treat each request for content as **an independent transaction** that can be completed.
- However, there are ways in which state information can be preserve. These are outside of the scope of the protocol, but encoded in applications and servers that use it.

Techniques for preserving state information implemented in client applications and Web sites is out of the scope of this course.



# Email

- Electronic mail (email) seems like a simpler system than ‘the Web’, in many ways it is more complex.
- The expectation is that an email has to be in exactly one place at any one time.
  - If it is in two places at the same time then it has been duplicated accidentally.
  - If it is in zero places then it has been lost ‘in the system’, in detriment of the sender and the recipient of that email.



Standard components involved in an email exchange.



# Email Associated Protocols

## The Simple Mail Transport Protocol (SMTP)

- Like HTTP, SMTP is a text-based protocol. But unlike HTTP, it is 'connection based', meaning that a client (in this case, the Mail User Agent) can issue multiple consecutive comments to the SMTP server, and should explicitly terminate its connection when its finished.

SMTP vocabulary used in the communication between client (C) and server (S).

01 S: 220 smtp.example.com ESMTP Postfix  
02 C: HELO relay.example.org  
03 S: 250 Hello relay.example.org, I am glad to meet you  
04 C: MAIL FROM:<bob@example.org>  
05 S: 250 Ok  
06 C: RCPT TO:<alice@example.com>  
07 S: 250 Ok  
08 C: RCPT TO:<theboss@example.com>  
09 S: 250 Ok  
10 C: DATA  
11 S: 354 End data with <CR><LF>.<CR><LF>  
12 C: From: "Bob Example" <bob@example.org>  
13 C: To: "Alice Example" <alice@example.com>  
14 C: Cc: theboss@example.com

A typical email exchange – part 1.



- At the end of this exchange, Bob's email has successfully been moved from his mail client, to his 'outgoing mail server'.
- There are other, more recent, email protocols, such as IMAP. Information about the protocols can be found at:
  - POP3: <http://www.faqs.org/rfcs/rfc1939.html>
  - SMTP: <https://tools.ietf.org/html/rfc5321>
  - IMAP: <https://tools.ietf.org/html/rfc3501>

15 C: Date: Tue, 15 January 2008 16:02:43 -0500

16 C: Subject: Test message

17 C:

18 C: Hello Alice.

19 C: This is a test message with 5 header fields and 4 lines in the message body.

20 C: Your friend,

21 C: Bob

22 C: .

23 S: 250 Ok: queued as 12345

24 C: QUIT

25 S: 221 Bye

{The server closes the connection}

A typical email exchange – part 2.

