R2.

Network architecture refers to the organization of the communication process into layers in order to run in different end systems and communicate each other. Application architecture is designed by an application developer who decide the building structure of the application (e.g., client-server or P2P). From the application developer's perspective, the network architecture is fixed and provides a specific set of services to applications.

R3.

* Client process: process that initiates communication (create a message to send into the network)
* Server process: process that waits to be contacted (receiving a message and possibly responding back with a message)

R4.

In the Internet, the host is identified by its IP address.

The sending process must also identify the receiving process running in the host. This information is needed because in general a host could be running many network applications.

many processes running on same host. A destination port number serves this purpose.

* The IP address of the destination host
* the port number of the socket in the destination process.

R10.

The process by which two devices initiate communications. Handshaking begins when one device sends a message to another device indicating that it wants to establish a communications channel. The two devices then send several messages back and forth that enable them to agree on a communications protocol. A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other. SMTP uses handshaking at the application layer whereas HTTP does not (uses but in another layer)

R12.

When the user first visits the site, the server creates an identification number that is unique for every user, creates an entry in its back-end database, and then returns this identification number as a cookie number. This cookie number is stored on the user’s host, in the user browser. When the user return again in the site and have done some purchase the browser send the cookie number back to the site. Like this, the user has different cookie numbers for different action on this site.

R13.

Web caching can bring the desired content “closer” to the user, possibly to the same LAN to which the user’s host is connected. browser sends all HTTP requests to cache

* + if object in cache: cache returns object
  + else cache requests object from origin server, then returns object to client and keeps a copy

Web caching can reduce the delay for all objects, even objects that are not cached, since caching reduces the traffic on links.

P4.

a) The URL of the document requested by the browser was http://gaia.cs.umass.edu/cs453/index.html. The Host field : gaia.cs.umass.edu indicates the server's name and /cs453/index.html indicates the file name wanted by the client.

b) The browser is running HTTP version 1.1 (1st line after Method –URL - **version**)

c) persistent connection **(Connection:keep-alive)**

d) This info is not contained in an HTTP message (Only available in TCP that carried the HTTP GET request)

e) Browser = Mozilla/5.0 . Browser type information is needed by the server to send different version of the same object to different types of browsers. For example , the same object will be requested from a google chrome and internet explorer but the server will send a different version to those browsers.( Each of the versions is addressed by the same URL.)

P5.

1. The code 200 indicatesthat server was able to find the document and object of the document will be send. The reply was provided on Tuesday, 07 Mar 2008 12:39:45 GMT.

b) Saturday 10 Dec 2005 18:27:46 GMT

c) 3874 bytes (Ranges: bytesContent-Length: 3874)

d) The first five bytes of the returned document are : <!doc

The server agreed to a persistent connection (Connection: Keep-Alive field)

P6.

1. either the client or the server can indicate to the other that it is going to close the persistent connection. It does so by including the connection-token "close" in the Connection-header field of the http request/reply.

HTTP/1.1 devices may close the connection at any time, though servers should try not to close in the middle of transmitting a message and should always respond to at least one request before closing

1. HTTP does not provide any encryption services.
2. “Clients that use persistent connections should limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy
3. “A client might have started to send a new request at the same time that the server has decided to close the "idle" connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress

P10.

1. Document time = (3.5 x RTT) + (1Kbyte)/(1 Mbps) = (3.5 x 100) + (1024x8)/(1000000)= 350ms + 8.192ms= 358.192ms

Image Time = 5x ((3.5 \* RTT) + (50x1Kbyte)/(1 Mbps))= 5x ((3.5 \* 100) + (50x1024x8)/(1000000))= 5 (350ms + 409.6ms)= 3798ms

Total time= Document Time + Image Time = 358.192ms + 3798ms = **4156.192ms**

1. 1st Connection with 2 parallel -> 2 object requested at the same time:

Document time = (3.5 x RTT) + (1Kbyte)/(0.5 Mbps) = (3.5 x 100) + (1024x8)/(0.5x1000000) = 350ms + 16.384ms = 366.384ms

Image time = ((3.5 \* RTT) + (50x1Kbyte)/(0.5Mbps)) = ((3.5 \* 100) + (50x1024x8)/(0.5x 1000000))= 350ms + 819.2ms=1169.2ms

2nd & 3rd Connection each with 2 parallel -> 2 object requested at the same time:

Image time double = 2x ((3.5 \* RTT) + (50x1Kbyte)/(0.5Mbps)) = 2x ((3.5 \* 100) + (50x1024x8)/(0.5x1000000))= 2(350ms + 819.2ms)= 2338.4ms .

In the 1st connection I can request two object parallel (document,image)

In the 2nd connection I will request two images

Same in the 3rd connection

For the 1st connection I have to take the time of the object that is higher

Total time= max (Document Time , Image Time ) + Image time for 2nd&3rd connection = 1169.2ms + 2338.4ms = **3507.6ms**

1. Document time = (3.5 x RTT) + (1Kbyte)/(Mbps/6) = (3.5 x 100) + (1024x8)/(1000000/6) = 350ms + 49.152ms = 399.152ms

Image time = ((3.5 \* RTT) + (50x1Kbyte)/(Mbps/6)) = ((3.5 \* 100) + (50x1024x8)/(1000000/6))= 350ms + 2457.6ms=2807.6ms

Total time= max (Document Time , Image Time ) = **2807.6ms**

1. Without Pipeline :

Document time = RTT + (1Kbyte)/(Mbps) = 100ms + (1024x8)/(1000000) = 100ms + 8.192ms = 108.192ms

Image time = 5 \* (RTT + (50x1Kbyte)/(Mbps)) = 5(100) + (50x1024x8)/(1000000))= 5 (100ms + 409.6ms) =2548ms

Total time = 2.5RTT + Document Time + Image Time = 250ms + 108.192ms + 2548ms = **2906.192ms**

**With Pipeline :**

Document time = (1Kbyte)/(Mbps) = (1024x8)/(1000000) = 8.192ms = 8.192ms

Image time = 5\*(50x1Kbyte)/(Mbps)) = 5((50x1024x8)/(1000000))= 5\*409.6ms =2048ms

Total time = 3.5RTT + Document Time + Image Time = 350ms + 8.192ms + 2048ms = **2406.192ms**

P29.

Yes, you can configure your browser to open multiple simultaneous connections to a website.

**Advantages:**

* Download the file faster

**Disadvantages:**

* Download speed for other users will be slower
* Extra traffic