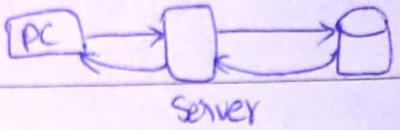
1). Client - Server Architecture:

* Always on host called server which requests services from other hosts called client.
* Client hosts can always on or sometimes on.
* With client server archi, clients do not directly communicate with each other
* Server has fixed well known add. (IP add.)



2). P2P:

* No always on server
* End systems directly communicate with each other
* Greatest strength is its scalibility.
* Because of highly distributed and de-centralized nature, it is difficult to mangge.

What services does an App. need?

1). Reliable Data Transfer: Email, file tranfer, web doc. transfers, and financial applications require fully reliable data transfer.

2). Bandwidth: Some applications must be able to transmit data at certain rate in order to be effective.

3). Timing: Some apps require tight timing constraints on data delievery in order to be effective

4). Security: encryption etc.

Mon Tue Wed Thu Fri Sat

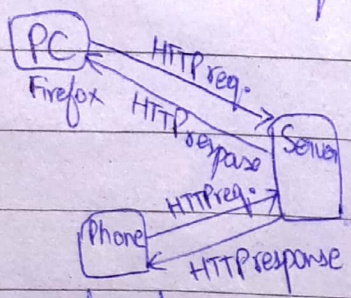
Date: __/__/20

TCP

TCP congestion control mechanism throttles sending process when network is congested b/w sender and receiver. The throttling of transmission rate can have very harmful effect on real time audio or video applications that have minimum required bandwidth constraint. For these reasons, real time apps usually run over UDP rather than TCP.

HTTP:

* Uses TCP as its underlying transport protocol. HTTP client first initiates connection with server. Once connection is established, browser and server processes access TCP through their socket interface and TCP connection closed.
* Stateless protocol, as server maintains no info about past client requests.



Non persistent: In nonpersistent conn., one TCP conn. is made for each request/response.

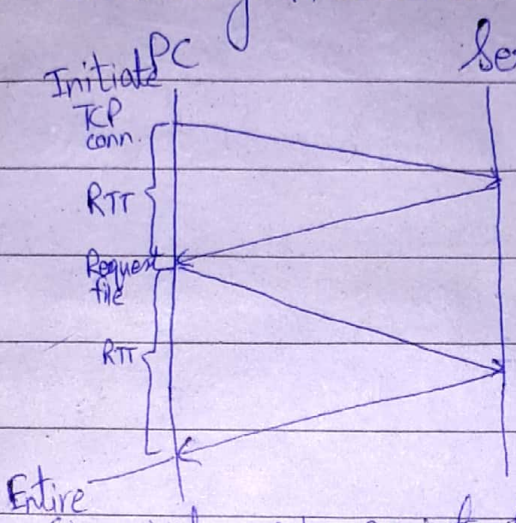
* Downloading multiple objects required multiple connections.

Tue Wed Thu Fri Sat

Date: / /20

RTT: Time it takes for small packet to travel from client to server and then back to client.

* Three way handshake.



High overhead on server because server need N diff buffers and requires slow start procedure

Non-persistent Response Time = 2RTT + file transmission time

Persistent:

* By default, HTTP uses this
* In persistent conn, server leaves conn. open for more requests after sending response. Server can close conn. at request of client or if timeout has been reached.
* As little as one RTT for all referenced objects

Web Caching (Proxy Server):

It is network entity that satisfies HTTP requests without involving origin server.

It has its own disk storage and keeps copies of recently requested objects in this storage.

Once browser is configured, request for an object is first directed to web cache.

If object in cache: then cache returns object
else cache requests object from origin server then returns

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

object to client.

* Cache acts as both server and client at same time

Q Why Web Caching?

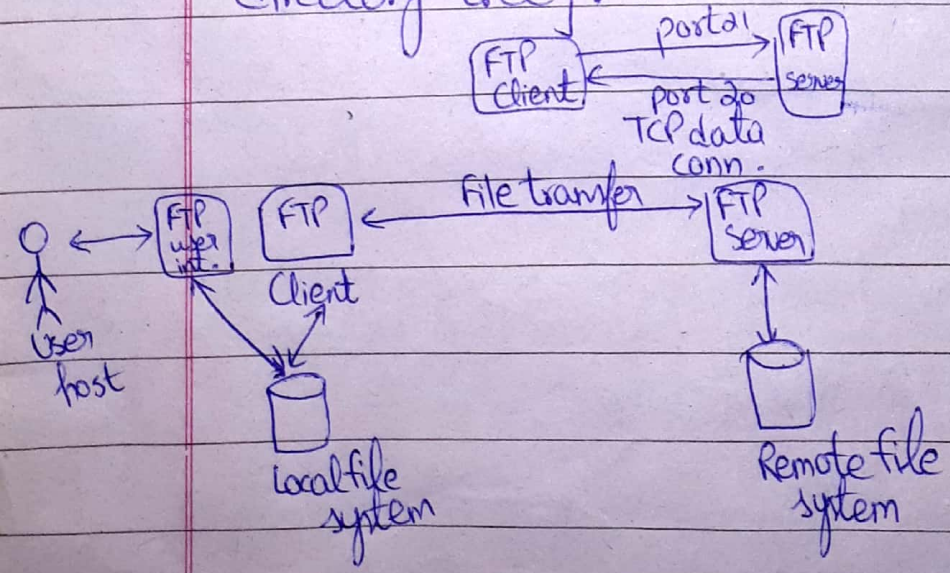
Ans 1). Reduce response time for client requests.

2). Reduce traffic on institutions access link.

FTP: Provided by TCP/IP for copying file from one host to another .or transferring files from one system to other.

It differs from other client/server apps in that it establishes two conn. b/w hosts. One conn. is used for data transfer, other for control info. (commands and response).

FTP server maintains state. [must keep track of user current directory as user wanders about remote directory tree].



Tue Wed Thu Fri Sat

Pg#2

Date: __/__/20__

SMTP: [Simple Mail Transfer Protocol]

* Used by client to send email to servers.
* SMTP is an application layer protocol
* It uses TCP [ensurity, ACK, password, connec oriented]
* Uses TCP Port 25.
* It is push protocol.
* Requires each message in 7bit ASCII

① SMTP SMTP POP, IMAP
User Agent → Mail server Alice → Mail server Bob → User Agent Bob

* Uses persistent connections

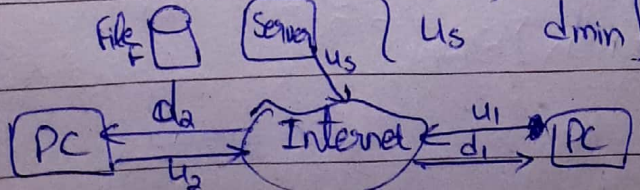
P2P File distribution:

1). Using Client Server archi.:

i). Server must transmit one copy of file to each of N peers. Thus server must transmit NF bits. Since server upload rate is u_s. Time to distribute file is NF/u_s

ii). Let dmin denote download rate of peer with lowest download rate. Peer with lowest download rate cannot obtain all F bits of file in less than F/dmin. So, minimum distribution time is at least F/dmin.

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$$



Mon Tue Wed Thu Fri Sat

Date: __/__/20

2). By using P2P:

1). At beginning of distribution only server has file. To get this file. Min. dist. time = $\frac{F}{u_s}$

2). F/d_{min} 3). $NF/(u_s+u_1+--u_N)$

$$D_{p2p} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s+\sum_{i=1}^{N} u_i}\right\}$$

P22

$F = 15\text{Gbits} = 15\times 1024\text{Mbits}$

$u_s = 30\text{Mbps}$

$d_i = 2\text{Mbps}$

For $N = 10, 100, 1000$

$u = 300\text{Kbps}, 700\text{Kbps}, 2\text{Mbps}$.

For Client Server:

u \ N	10	100	1000
300Kbps	7680		
700Kbps	7680		
2Mbps	7680		

For $N=10, u=300\text{Kbps}$

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$$

$$= \max\left\{\frac{15\times 1024\times(10)}{30}, \frac{15\times 1024}{2}\right\}$$

$$= \max\{5120, 7680\}$$

Mon Tue Wed Thu Fri Sat

Date: __/__/20

For $N=100, u=300kbps$

$$D_{cs} = \max\left\{\frac{NF}{Hs}, \frac{F}{d_{min}}\right\}$$

$$= \max\left\{\frac{15\times1024\times100}{30}, \frac{15\times1024}{2}\right\}$$

$$= \max\{51200, 7680\}$$

For PaP:

$N=10, u=300kbps$

$$D_{pap} = \max\left\{\frac{F}{Hs}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

$$= \max\left\{\frac{15\times1024}{30}, \frac{15\times1024}{2}, \frac{15\times1024\times(10)}{30+\frac{300}{1024}+\frac{700+2}{1024}}\right\}$$

$$= \max\{512, 7680, 4657\}$$