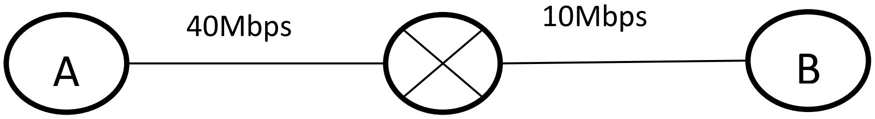
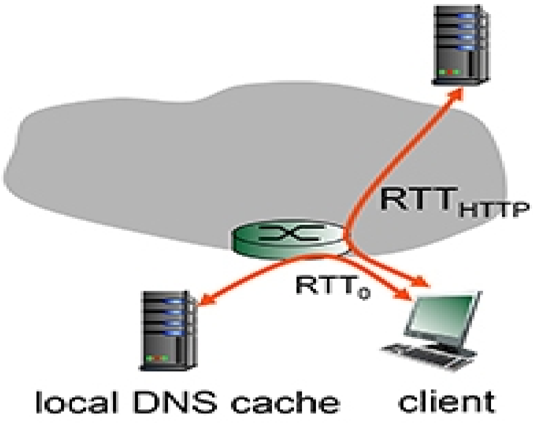
CS5222 Computer Networks and Internets

Tutorial 3 (Week 3)

1. Suppose that A has a file with size of 1 Gbits to send to B through the following path. **How long does it take (in sec) from the moment B receives the first bit of the file until B has received the whole file?**



2. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. It takes RTT_0 = 5 msecs for the host to send a DNS lookup and obtain the IP address. The Web page associated with the link **does not reference any other objects** and consists of a small amount of HTML text. The RTT between the local host and the Web server containing the object is RTT_{HTTP} = 15 msecs.

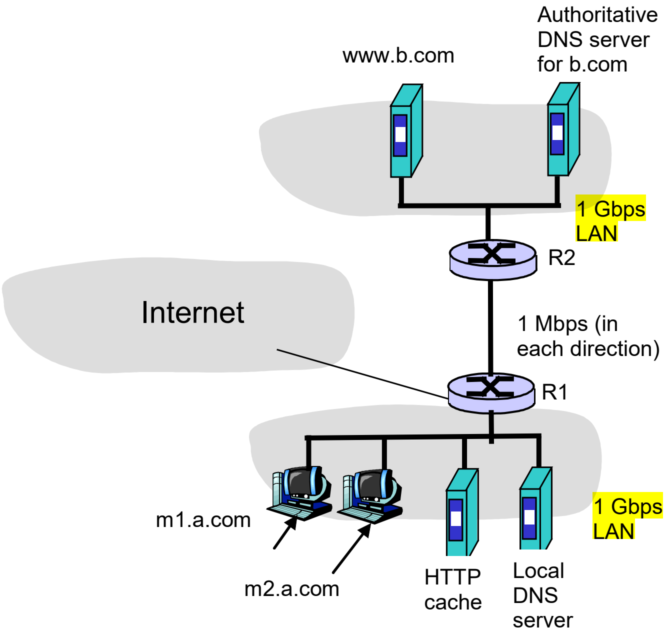


Assuming it takes zero transmission time of the HTML object, how much time elapses from the client clicking on the link to the client receiving the object?

3. Consider the networks shown in the figure below. There are two user machines m1.a.com and m2.a.com in the network a.com. Suppose the user at m1.a.com types in the URL www.b.com/bigfile.htm into a browser to retrieve a 1Gbit (or 1,000 Mbit) file from www.b.com. We have the following assumptions.

    - The packets containing any DNS commands and HTTP commands such as GET are very small compared to the size of the file, and thus their transmission times (but not their propagation times) can be neglected.

- Propagation delays within the LAN are small enough to be ignored. The propagation from router R1 to router R2 is small enough to be ignored.
- The propagation delay from anywhere in a.com to any other site in the Internet (except b.com) is 500 ms.



a) List the sequence of DNS and HTTP messages sent/received from/by m1.a.com as well as any other messages that leave/enter the a.com network that are not directly sent/received by m1.a.com from the point that the URL is entered into the browser until the file is completely received. Indicate the source and destination of each message. You can assume that every HTTP request by m1.a.com is first directed to the HTTP cache in a.com. **Assume that all caches (HTTP cache and local DNS server) are initially empty**. Moreover, all DNS requests are iterated queries. Calculate the time for m1.a.com to receive the file.

b) Now assume that machine m2.a.com makes a request to exactly the same URL that m1.a.com made. List the sequence of DNS and HTTP messages sent/received from/by m2.a.com as well as any other messages that leave/enter the a.com network that are not directly sent/received by m2.a.com from the point that the URL is entered into the browser until the file is completely received. Indicate the source and destination of each message.

4. Assuming that you accessed Amazon.com previously. When you access Amazon.com again, the website lists the items you browsed before and provides some recommendations. Explain how this happens.

5. Suppose that a web browser wants to display a webpage that contains references to 10 objects. Assume that the webpage and the referenced objects are very small, and hence their transmission times can be ignored. For each one of the following scenarios from (a) to (d), answer the following two questions.

- How many HTTP request messages in total does the web browser need to send in order to retrieve all objects?
- How many RTTs does it take when all objects can be received by the client?

Scenarios:

a. The web browser can open up to 5 **parallel TCP connections** to the server over which it can send/receive HTTP messages. Assume that **non-persistent HTTP** is adopted.
b. The web browser can open up to 5 **parallel TCP connections** to the server over which it can send/receive HTTP messages. Assume that **persistent HTTP** is used.
c. The web browser can create a **single TCP connection** to the server over which it can send/receive HTTP messages. Assume that **non-persistent HTTP** is used.
d. The web browser can create a **single TCP connection** to the server over which it can send/receive HTTP messages. Assume that **persistent HTTP** is used.