CS414 SP 2007 Assignment 5

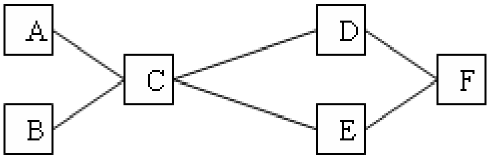
Due April. 11 at 11:59pm Submit your assignment using CMS

1. Discuss whether clients in the following systems can obtain inconsistent or stale data from the file server and, if so, under what scenarios this could occur.
(a) AFS
(b) NFS

2.
(a) We described the algorithm for reliable flooding (used in Link-state routing) in class. Discuss what problem might raise if the field to store sequence number is small (say 6-bit). Do you think this is a concern in nowadays' Link-State routing? Why?

(b) In IP, packets may be fragmented at an intermediate router, but are re-assembled only at the receiving end host. Why aren't packets re-assembled at the router adjacent to the fragmenting router?

(c) Consider the following fish topology: Link A-C, C-D, D-F (all upper links) have bandwidth 1Gbps. Link B-C, C-E, E-F (all lower links) have bandwidth 100Mbps.



Suppose we know that A sends traffic to F at a rate of 1Gbps, and B sends traffic to F at a rate of 100Mbps. It would be nice to engineer the traffic so that A's traffic takes the path A-C-D-F, and B's traffic goes through B-C-E-F. This way we efficiently utilize all the links. Is this possible with IP routing? If so, describe how. If not, explain why.

3. When DNS was originally designed, it was assumed that TTL (Time-to-Live, the cache lifetime) values would be on the order of many hours or even a few days. Later, companies like Akamai started setting the TTL values to very low values---a couple minutes or even less. (This is true only for the lowest level answers. In other words, for a DNS query on host10.akamai.com, the NS (Name Server) record for akamai.com would still have a high TTL, but the A record for host10.akamai.com would have a low TTL). At the time when this happened, some people thought that this would overload the DNS system. Considering the structure of DNS (hierarchy of servers, local DNS resolvers), do these small TTLs overload DNS? Why or why not?

4.

(a) TCP can perform poorly if there is a long end-to-end latency, and there are many losses in the network (say, 5%). If most of the losses happen at a certain link, and that link has a small latency, then performance gains can be had by implementing a retransmission scheme at the link layer. Would doing this violate the end-to-end principle? Why or why not?

(b) TCP sends positive acknowledgments of received packets. Given that most packets arrive successfully, wouldn't it make more sense for TCP to send "negative acks" (that is, indicate which packets weren't received)? What is the problem with a NAK-based scheme.

(c) In TCP slow start, at first the sending host will double the number of packets it sends every time it receives all acknowledgements for the previous packets it sent. In other words, the sending host will send one packet, receive the ACK for that packet, send two packets, receive the ACKs for those packets, send four packets, and so on. This exponential growth continues until there is a packet loss. (Subsequent to this loss, the sending host will increase the number of packets linearly, not exponentially.) Consider the scenario where a host H1 has two connections, one to a host HL on the same LAN, and one to a host HW thousands of miles away. The one-way latency to host HL is 1 microseconds, the one-way latency to host HW is 50 milliseconds and the round-trip latency is twice the one-way latency. Assuming no packet losses, and assuming near infinite bandwidth and processing speeds at all hosts, how long does it take to send the first 1023 packets to each host?

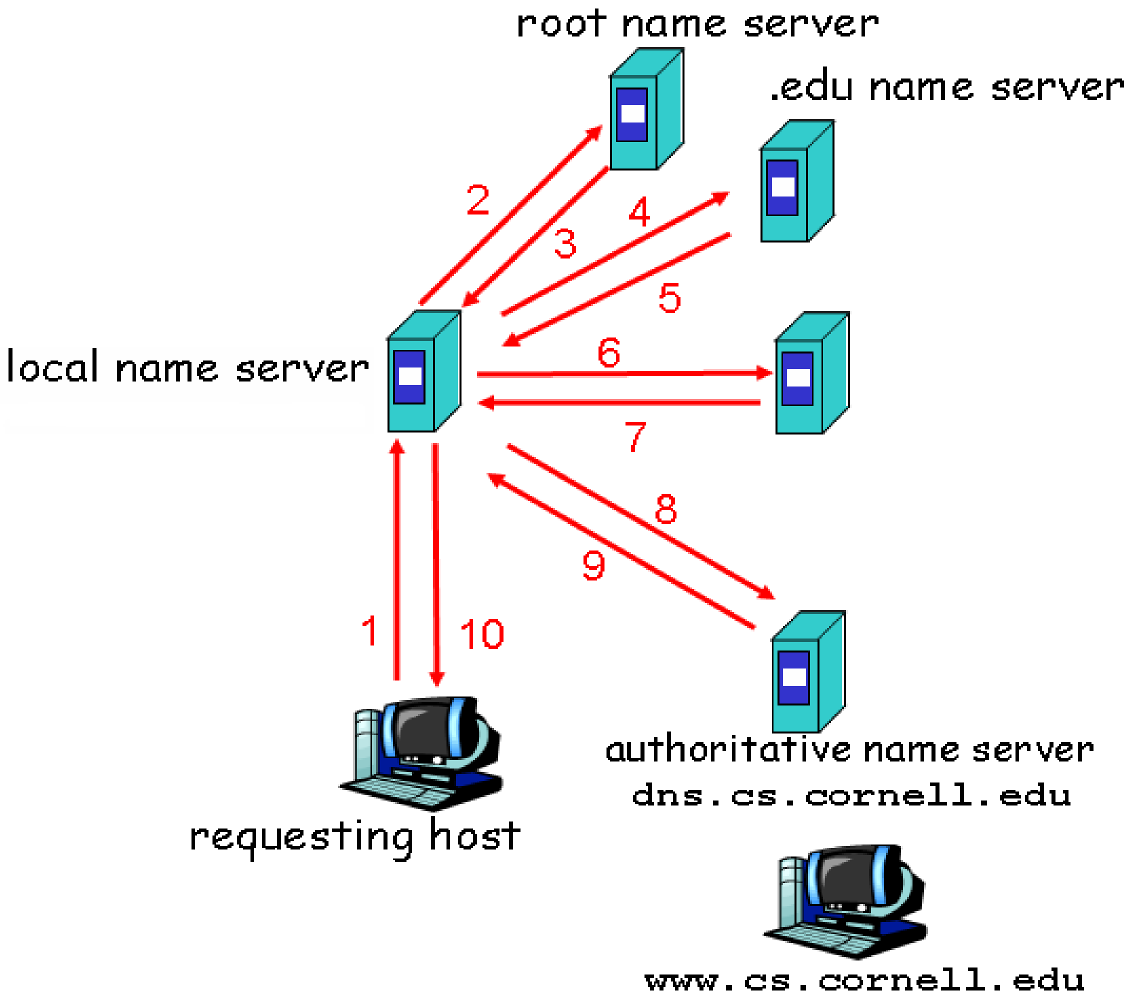
.

5.

(a) The Unix utility **ping** can be used to find the RTT (Round-Trip-Time) to various Internet hosts. Read the man page for **ping**, and use it to find the RTT to www.google.com in California. Measure the RTT value at different times of day, and compare the results. What do you think accounts for the differences?

(b) The Unix utility **traceroute** can be used to find the sequence of routers through which a packet is routed. First, use this to find the path from your site to www.google.com. Paste the output below, and point out which of the routers are within your organization. Then try out a few more sites with **traceroute** and **ping**. How well does the number of hops correlate with the RTT times from **ping**?

(c) The Unix utility **nslookup** is a DNS-lookup tool. Read the man page for **nslookup**. Use **nslookup** to carry out manually a name lookup such as that in figure below, that is, carry out step 2 to 9. Note: you should do this in **nslookup** interactive mode, disable the recursive lookup feature, and specify that queries are for NS (Name Server) records rather than the usual A records. Paste your interactive **nslookup** session below (the command you typed and the output)


root name server
.edu name server
2
3
4
5
local name server
6
7
8
9
1
10
requesting host
authoritative name server
dns.cs.cornell.edu
www.cs.cornell.edu