Seminar Report: Namy

Albert Bausili, Noa Yu Ventura, Pol Verdura

May 17, 2023

Upload your report in PDF format.
Use this LaTeX template to format the report.

1 Open questions

Try to answer all the open questions in the assignment. When asked to do so, perform experiments to support your answers.

a) What happens if a client repeats the same query on experiment MS1?
Because the TTL equals 0, it won't be cached and the resolver will have to request the URL again to the root server.

b) What is the observed behavior on experiment MS2?
The host is unreachable, it is not found. This is because with our implementation it is assumed that hosts never fail, which means the resolution for a name will always be the same. In this case, a host fails which means the address the parent has is wrong, it is sending the old address the host had, and then delivers it to the client. When the client resolver tries to contact this host using the old address the host does not reply, because it does not receive the request, since it has a new address now.

c) What is now the observed behavior on experiment MS2?
The parent server gets a notification that its child has died, and removes it from its own cache, so now it knows the child died and will remove its entry. This way the parent doesn't send misinformation to the cient.

d) What is the observed behavior on experiment MS3?
The resolver doesn't find the host we shut down, because the TTL is so long the client asks to the cache of the resolver, and it is valid, so it searched for the last position of the host, which is now not the same since we shut it down and then restarted it. If we wait for the TTL to reach 0 then we don't have this problem, since the entry now is invalid and we query the root server.

e) What happened with cached information in the resolver?
The cached information is wrong and will be until the TTL expires. Only when it is requested again and TTL reaches 0 it will be updated.

f) Which nodes have been notified about the host movement?
The parent, it sends a deregister message.

g) **When will the client find the host correctly in the new location?**
When the client does a query to the resolver and it hasn't the entry cached, the servers will return the names they know. Since a host is deregistered when it dies, the server parent won't have information about this fallen host, which means it will look for it again and get the new address. Then the parent server will give the new address of the host that died to the client.

h) **Derive a theoretical quantification of the amount of messages needed for name resolution without and with the cache (assume a resolver that repeats the same query about a host at depth D in the namespace every F seconds for a total duration of R seconds, being the TTL equal to T seconds).**
Without cache it's always messages = (((D*2)+2)/F)*R from resolver to the depth D host. With cache if T <= F, then it will always be like without cache. With cache, if R <= F, then it will always be messages = (D*2)+2 (only one query). With cache if T > F and R > F, then we have two cases: hit cache or miss cache. The proportion of misses is F/T, the time we need to wait for the response of a query is R/2*F, 2*(D+1) are the messages that the client sends to the resolver, and the amount of messages of a query with cache miss is (D*2)+2, and with cache hit is 0. With cache we have messages = (F/T)*(R/2*F)*((D*2)+2) + 2*(D+1) = (R/2*T)(D*2+2) + 2*(D+1)

i) **Which resolution can exploit caching better? Justify why.**
Recursive, because the hosts also learn something and can cache themselves, unlike iterative that they only know they children.

2 Personal opinion

Give your opinion of the seminar assignment, indicating whether it should be included in next year's course or not.

- **Albert**:
I found it really confusing writing the code without almost any clue, and the explanation was not specific enough. The testing of the code was really annoying to debug or even to know if the changes were right, and I feel like we lost a lot of time on those testings rather than coding or understanding the purpose of the lab.

- **Noa**:
I'm not sure if there's a better way to test the code, but I find it very time-consuming, boring and tedious to open so many terminals and so many commands to test if your code works well. It would be nice to have a script that executes all of them, so all you need to do is check the ping's output. It was interesting to know how DNS works, but unlike other topics of the course, this theory and practice is pretty straightforward and doing the practice doesn't reinforce the theory knowledge we know unlike other labs we did before (talking about groupy and ordy specially). I would like the pdf to be more precise on what we need to do too, we spent a lot of time

doing this lab compared to others, and it's not like this topic is harder than others we did before.

- **Pol**:
  It was hard to test because you had to maintain various terminals simultaneously. Moreover, if a small change was made, you had to restart the same configuration and it was really slow. On the other side, the seminar was interesting and helps you see how TTL in DNS works.