RA5 – 25100198

Q1. I disagree with Emaan as she states that a cache-miss due to an out-of-order or dropped packet will cause extra workload on the cache via the web-server trying replicate an already existing value. This is incorrect, as the paper states that “but web servers will skip inserting entries into *memcached* after querying for data to avoid putting additional load on a possibly overloaded network or server.” As such, Emaan’s concerns are not founded in the design of the algorithm, which specifically avoids this problem.

Q2. This problem is solved via the *memcached* system using *leases.* Essentially, when a web-server returns a value to a client for setting the cache, it includes a 64-bit token called a lease. This token can be used by the cache to verify whether or not this value should be stored in the cache. Importantly, this token allows a cache to determine if some value is stale.

The exact mechanism through which it does this verification is not mentioned and is likely proprietary knowledge. However, of the limited intelligence a *memcached* cache has, the ability to verify a data’s validity using *leashes* is part of it. According to the paper, this form of verification is similar to how the load-linked/store conditional works in low-level systems programming, which aims to check if some data item has changed since the time it was last loaded.