# MOBILE COMPUTING - 20XW61

## LAZY CACHING

SAI DINESH B
20PW30

### Need of Caching :

- Every time a user x is called, x's location is cached at the VLR in the caller's zone, so that any subsequent call to x originated from that zone can reuse this information.
- Caching is useful for those users who receive calls frequently relative to the rate at which they relocate.
- To locate a user, the cache at the VLR of the caller's zone is queried first. If the location of the user is found at the cache, then a query is launched to the indicated location without contacting the user's HLR. Otherwise, the HLR is queried.
- In eager caching, every time a user moves to a new location, all cache entries for this user's location are updated. Thus, the cost of move operations increases for those users whose address is cached.

### Lazy caching :

- In lazy caching, a move operation signals no cache updates.
- when at lookup a cache entry is found there are two cases:
    - User is still in the indicated location and there is a cache hit.
    - User has moved out, in which case a cache miss is signaled.

- In the case of a cache miss, the usual procedure is followed: the HLR is contacted and after the call is resolved , the cache entry is updated. Thus, in lazy caching, the cached location for any given user is updated only upon a miss.

Let μ be the Location Area crossing rate (From the formula of CMR ratio) and C be the location updating cost of the callee whenever the mobile device moves to a new location area.

Then, updating cost of location area = $\mu * C$

Then , paging cost = $\lambda ( p* C_v +(1-p)* ( C_v + C_h) )$

$= \lambda (C_v+ (1-p) * C_h)$

Where λ is the call arrival rate, p is the probability of cache hit , $C_v$ is the cost of updating/querying the cache in the caller's VLR , $C_h$ is the cost of querying the HLR in the callee.

$$p = \frac{\lambda}{\lambda + \mu}$$

### COMPARISON OF EAGER CATCHING COST AND LAZY CATCHING COST :

EAGER COST : $C_e = \mu (C + C_v ) +\lambda C_v$

LAZY COST : $C_l = \mu C +\lambda [ C_v + (1-p) C_h ]$

$$
\begin{aligned}
C_e - C_l &= \mu c_v - \lambda(1-p)c_h \\
&= \mu c_v - \frac{\lambda\mu}{\lambda+\mu} c_h \\
&= \mu\left(c_v - \frac{\lambda}{\lambda+\mu} c_h\right) \\
&= \mu\left(c_v - \frac{CMR}{CMR+1} c_h\right)
\end{aligned}
$$

$$
\begin{aligned}
C_e - C_l &= \mu\left(c_v - \frac{CMR}{CMR+1} c_h\right) \\
&= \begin{cases} \mu c_v & \text{if } CMR \to 0 \\ \mu\left(c_v - c_h\right) & \text{if } CMR \to \infty \end{cases}
\end{aligned}
$$

If CMR is small, lazy caching has a lower cost.

If CMR is large, eager caching has a lower cost.