Department of Computing

**Hong Kong Polytechnic University**

**Comp 5527 Mobile Computing and Data Management**

**Tutorial Three Sample Solutions**

Q1: Sample Solutions:

In broadcast disk caching, hot items will be broadcast very frequently and it does not make good use of the limited cache to cache them. On the other side, cold items will be broadcast very sparsely but there are too many of them to be cached, and each cached cold item only contributes marginally due to low access rate. Broadcast disk adopts a cost-based replacement mechanism to take into account of the factors above.

PIX cost function for replacement: access Probability Inverse broadcast frequency (X).

Cost = P / X (where X is frequency of the item being broadcast).

In this problem, access probability for items in group A, B and C are q**A** = 4/21, q**B** = 1/21, q**C** = 1/168.

Three broadcast disks A, B and C, with broadcast frequency 4:2:1

Assume that the cache is of size 3, with a data item access sequence of 2, 5, 4, 7, 5, 3, 5, 9.

PIX**2** = PIX**9** = 1/21; PIX**5** = PIX**7** = 1/42; PIX**3** = PIX**4** = 1/168.

With 2, 5, 4 arriving, cache contains **2, 5, 4** in that order.

With 7 coming, remove 4 with lowest PIX and put in 7: **2, 5, 7**.

With 5 coming, no action.

With 3 coming, remove 5 with lowest PIX and oldest: **2, 7, 3**.

With 5 coming, remove 3 with lowest PIX: **2, 7, 5**.

With 9 coming, remove 7 with lowest PIX and oldest: **2, 5, 9**.

Q2. Sample Solutions:

Mobile wireless networks can be broadly classified as either infrastructure based or ad hoc based. In the former scheme, as shown in Figure 1, a fixed network device such as a mobile support station (MSS) forwards messages that mobile hosts (MHs) send or receive. The MSS is similar to the server in a traditional client-server distributed system in that all source data is deployed on it. Other MHs retrieve data from the MSS and can cache a replica by themselves. In contrast, ad hoc based mobile networks, like that shown in Figure 2, do not store data on the MSS but use it only as the access point to the Internet. Ad hoc based mobile networks disperse all data items for searching and querying across the mobile hosts.

In an infrastructure-based mobile network, caching consistency can be achieved through one-hop Invalidation Report (IR) from MSS. However, in an ad hoc based mobile network, the MSS may be unreachable in one single hop thus is undependable for caching invalidation. Meanwhile, all source data are dispersed across the network and each mobile host may have both the source data and cache copies of some data items. This makes the cache invalidation much more complex. Therefore, new cache invalidation techniques are needed for ad hoc based mobile network.

Q3. Sample Solutions:

In data cache schema, a node caches a passing-by data item locally when it finds that the data item is popular and it has enough free cache space.

*a)* both N6 and N7 request di through N5.

N6, N7 will cache the data locally. Moreover, N5 knows that di is popular and caches it locally. Future requests by N3, N4, or N5 can be served by N5 directly.

Data cache needs extra space to save the data, it should be used prudently. Suppose the data center receives several requests for di forwarded by N3 − N4 − N5. All these three nodes may think that di is a popular item and should be cached. However, it wastes a large amount of cache space. To avoid this, a conservative rule should be followed: *a node does not cache the data if all requests for the data are from the same node.*

*b)* both N1 and N2 request a data di through N3-N4-N5.

All requests received by N5 are from N4, which in turn are from N3. With the new rule, N4 and N5 do not cache di. The requests received by N3 are from different nodes N1 and N2, so it will cache the data. N1 and N2 also cache the data.

*c)* N1 request a data di through N3-N4-N5

N3 , N4 and N5 do not cache di . N1 cache di

In path cache schema, a node only records the data path when it is closer to the caching node than the data center.

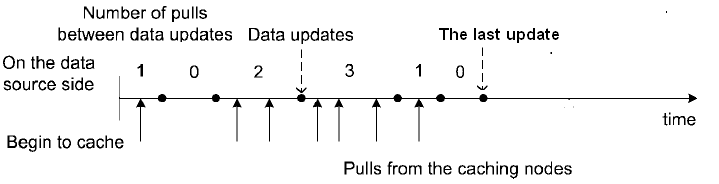
*d)* N1 request a data di through N3-N4-N5.

Node N1 has requested a data item di from N11. When N3 forwards the data di back to N1, N3 knows that N1 has a copy of di so cache the path. Later, if N2 requests di, N3 knows that the data center N11 is three hops away whereas N1 is only one hop away. Thus, N3 forwards the request to N1 instead of N4.

N4 and N5 need not cache the path information of di since N4 and N5 are closer to the data center than the caching node N1.

Q4. Sample Solutions:

We calculate the number of pulls between consecutive two updates and get the history: 102310



So we predicate the probability that a number of *v* caching nodes will pull the data after the last update.

p(0|102310) =0.3, p(1|102310) =0.3

p(2|102310) =0.1, p(3|102310) =0.1

Since the sum of the all probabilities is not equal to 1, it means that some possible values in *v* cannot be predicated by current history information. Here we use a simple estimation

p(4|102310)=1-0.3-0.3-0.1-0.1=0.2.

4 is used because it is the smallest integer in the set *N*-{0,1,2,3} where *N* is the non-negative integer set.

The expected value of the number of predicated pulls is

0\*0.3+1\*0.3+2\*0.1+3\*0.1+4\*0.2=1.6

So if *th*=1.5, the source node needs to push the update to the caching nodes. If *th*=2, it has no need to push.