

ECE 358 — Assignment 2

Kevin Carruthers, Lara Janecka, Clarisse Schneider
Spring 2016

Question 1

For any network with some m we can always set $p = 0$. From there, we can build a counter example where $q = 2^{m-1}$. If we put no nodes between p and q (ie. $q - p = 0$) every entry in p 's lookup table will be q . This way we guarantee that q will be the next hop from p . If we set k to some value greater than q , $k - p = 1$. So for any value of m can generate a network with $p = 0$, $q = 2^{m-1}$, and $k = 2^m - 1$. This will always result in $q - p = 0$ and $k - p = 1$ – which is a counter example to the generic case. Therefore this holds for all values of m .

Another counter example is if you always have $p = 0$, $q = 2$, and no node between p and q , and $k = 3$, it doesn't matter how many nodes you add to this network or how you shift the value of m , the equations of $q - p = 0$ and $k - p = 1$ will still hold which violates our claim.

Question 2

a)

We know that you cannot modify more finger tables than actually exist, therefore the worst of all cases is n . Because there is no upper limit on m , we can have a network that is complete (since m can be essentially infinite). In this case, we will have every node in the network containing each other in their finger tables. When we remove any of the nodes in the network, every other node will have to update their finger tables. If we were to add that node back into the network, all those tables would have to be updated again.

b)

When we add a node n into this network, only one table will need to be updated. The node right before n will have to update its table to now have n as its successor. Everyone else will remain the same.

Question 3

a)

Based on our current definition of finger tables we know that each time we hop, we will cover at least half the arc length from our current location to the desired end point. The farthest distance we would have to cover is if we are looking for a node a full circle (length n) away from us. The number of hops that we must make will be $\log n$. The value of n must be small enough to avoid collisions in its finger tables so $n < 2^m - 1$. From this we know that the number of hops will be less than $\log(2^m)$, which works out to being linear with respect to m . This means that there exists a linear equation with respect to m with some constants that are large enough to be bigger than the number of hops required.

b)

We know that the worst case for the number of hops in a network is for it to be full (a node at every value) and for n to be equal to $2^m - 1$.

Base case: In the base case of m being equal to one we have at most two nodes so the value you are looking for will be the only entry in the lookup table so the number of hops will be one.

Assumption: Let's assume that the number of hops required will be equal to m .

Proof: If we increase m by one, we are doubling the number of nodes in the worst case network. This does not matter because each finger table will update to include another entry that covers twice the distance, therefore we will only need one extra hop. With this we will need $m + 1$ hops.

With this we can see that a network will take m hops in the worst case so we can create any function of the form $m - c$ that will be less than the number of hops for the worst case network.

Question 4

1. 1.2.3.4 is in 1.2.3.0/24 but not in 1.2.3.160/28 since the former covers all IPs from 1.2.3.0 to 1.2.3.255 but the latter only covers some IPs above 1.2.3.160.
2. 1.2.3.195 is in 1.2.3.0/24 but not in 1.2.3.160/28 since the former covers all IPs from 1.2.3.0 to 1.2.3.255 but the latter only covers some IPs below 1.2.3.175.
3. 1.2.3.171 is in both 1.2.3.0/24 and 1.2.3.160/28 since the former covers all IPs from 1.2.3.0 to 1.2.3.255 and the latter covers the range from 1.2.3.160 to 1.2.3.175.