Solution of Assignment 1

Due Date: November 24, 2017

November 22, 2017

1 Prove that any acyclic graph/network has a longest simple path.

Proof: Given any acyclic graph $G = (V, E)$,we first prove that any path cannot pass one node more than once. Without loss of generosity, if there a path $A_1, A_2 ... A_k$, contains node A_i for more than once, the graph must forms a ring with $A_i A_{i+1} ... A_i$. Thus any path in an acyclic graph cannot contain one node for more than once. Thus the length of the path satisfies:

$$L(p) < |V|.$$

Moreover, Since a path cannot contain one node for more than once, the number of paths in this graph satisfies:
$N(G) \leq 2^{|v|}$
Thus the number of the paths in the graph is limited. Thus we have a finite path set P, and ∀p ∈P, $L(p) < |V|$, there must exists a longest simple path.

2 Write the Routing Algorithm for complete binary tree.(Suppose nodes are numbered from top to down, from left to right, and beginning from root.)

Algorithm 1: Routing Algorithm for a complete binary tree

Input: a complete binary tree T, source label s,destination label D
Use Dijistra algorithm to find the shortest path length for each node
Initialize a distance array d(,);
for i=1 to $|V|$ **do**
Q $\leftarrow \emptyset$
for j=1 to $|V|$ **do**
d(i,j) $\leftarrow \infty$;
Q $\leftarrow$ Q $\cup$ j;
end for
d(i,i) $\leftarrow$ 0;
While$Q \neq \emptyset$ **do**
j $\leftarrow$ $arg\ min_j\ d(i,j)$;
remove j from Q;
for each edge (j,k) of j **do**
if d(i,k) > d(i,j) + 1 **do**
d(i,k) $\leftarrow$ d(i,j) +1
end if
end for
end while
RoutingAlgorithm
r $\leftarrow$ s;
while $r \neq D$ **do**
if $d(r/2, d) < d(r, d)$ **then**
r $\leftarrow$ $r/2$ $Routetotheparentnode$
else if $d(2r, d) < d(r, d)$ **then**
r $\leftarrow$ 2r; Route to the left-son node
else if $d(2r + 1, d) < d(r, d))$ **then**
r $\leftarrow$ $2r + 1$; Route to the right-son node
end if
end while

---

3 Prove that hypercube is optimal in fault tolerance.

Definition of Fault tolerance: Fault tolerance FT is defined as:
FT = arg $m_k$ax The graph is still connected after removing k nodes.
A graph with degree d has optimal fault tolerance if it's fault tolerance is $d - 1$
We prove it by induction.
In a q-cube, the degrees of each node is q.

- q=1 When we remove the node, there remains only one node. The case is trivial.
- Suppose when q=k, hypercube has optimal fault tolerance. That means a k-cube is connected after removing k-1 nodes.
  When $q = k + 1$, we try to prove a (k+1)-qube is still connected after removing k-nodes. From the definition of hypercube, a (k+1)-hypercube is composed of two k-hypercubes. Thus the deleted nodes can be assumed as $k = n_i + n_2$, where n_1, n_2 represents the deleted nodes in each sub-hypercube, respectively.
  If $0 < n_1 < k$, each sub-hypercube is still connected by induction. Moreover, Since there are 2^k edges between two sub-hypercubes, and we delete at most k edges between two hypercubes, the two sub-hypercubes are inner-connected.
  Thus the $(k + 1)$ cube is still connected.
  if $n_1 = 0$ or $n_1 = k$, the deleted nodes are in one sub-hypercube. Suppose the two sub-hypercubes are H_1 and H_2, we know that each vertex in H_i .

  1. Any two vertexs in H_2 are still connected for we don't remove any vertex in H_2;
  2. A vertex in H_1 and a vertex in H_2 are connected according to the definition of (k+1)-qube;
  3. Any two vertex in H_1 are connected because each vertex in H_1 connects to a vertex in H_2 and H_2 inself is connected.
     Hence, the (k+1)-qube is still connected after removing k nodes.
     Hence proved that the hypercube is optimal fault tolerance