CE6390 – HW1
Seungtack Baek
Feb/22/2012

1. Since Ident field has 16 bits, the maximum number of packets that can exist at the same time is 2^16=65536 packets. Also, each packet is 576 bytes. Combining two results, we know that there will be 65536x576=37748736 bytes in the network. However, a host cannot send another packet for 60 seconds, so the maximum bandwidth is 37748736/60=629145.6 bytes/seconds. (or 629MBps). If this limit were to be exceeded, then there will be two or more packets of same Ident at the same time, and the receiver of the packet will not be able to distinguish two different packets with same Ident.

2. *proxy ARP*

   a) all the ARP messages are as following.

ARP-REQUEST (sent from A)		**ARP-REPLY (sent from B directly)**	
src IP	A's IP address	**src IP**	C's IP address
src Physical addr.	A's Physical address	**src Physical addr.**	B's Physical address
dst IP	C's IP address	**dst IP**	A's IP address
dst Physical addr.	*Empty*	**dst Physical addr.**	A's Physical address

   b) B's routing table looks as the following

DEST	IP Address	Physical Address
A	A's IP address	A's Physical address
C (if B knew of C beforehand..)	C's IP address	C's Physical address
default	0.0.0.0	n/a

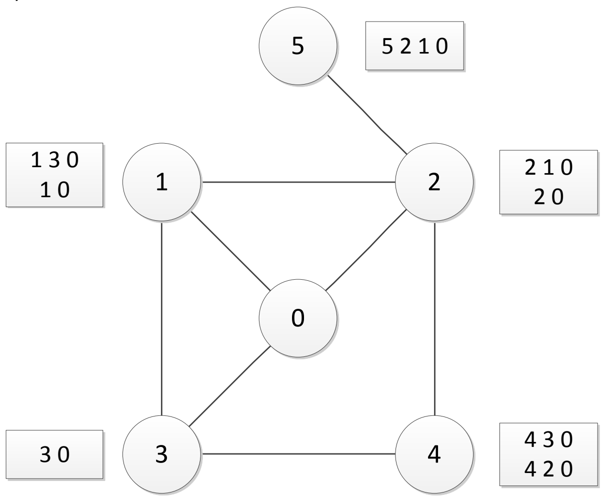
   *NOTE: the peculiarity lies in the B, even though it should not know of the address of A, it knows of A's physical address, along with its IP address.

3.
   a) i – System administrators can "lie" (or inject) that R1 can reach service Provider X and that R2 can reach Service Provider Y.
      ii – Inside the client, routers who can reach R1 faster (yes, now we can choose optimal path over safe path) than R2 will choose R1 and vice versa.
   b) The border routers in each Service Provider AS will inject that they can reach Client. So whenever any router has to send IP to Client, it will go through the border router who speaks BGP.
   c) AS Y will advertise only his address, because its IP address block contains the address block of client.
   d) If X advertises that he can reach client 110.128.0.0/20, then all traffic will go through X. this is because everyone will forward packets via "longest prefix match." Since X offers better accuracy (20 bits, as opposed to 16 bits, which Y advertises), everyone will forward packets to X (if the destination is X)

4. Let's consider two autonomous systems X and Y with IP address 123.0.0.0/8 and 123.123.0.0/16, respectively. Also, X contains another autonomous system Z with IP address 123.123.123.0/24. Then both X and Y will advertise their own address, and only their addresses, to minimize entries in the routing table for the core of the Internet. Now, if others outside this system wish to send to

Z, 123.123.123.0 will match more with 123.123.0.0/16 then with 123.0.0.0/8. Thus the package is misrouted.

5. reverse-engineering SPP,



Since each node in the dispute graph provides paths available, we start linking each node and write down all possible paths next to each node. Then we figure out the rank of path, inferred from dispute graph. For example, to find out which path has higher rank for node 2, we look at the relationship between (2 1 0), (4 2 0) and (2 0). Since (2 0) gives transmission arc to (4 2 0), they should have approximately similar rank. Yet, (2 1 0) gives dispute arc to (4 2 0), which means that (2 1 0) has relatively higher rank than (4 2 0). From this info, we can infer that (2 1 0) should have higher rank than (2 0). We can find out the ranks for node 1 and 4 in similar way.

6. Proof by induction: at any ith iteration (up to |V|+1 iterations), the partially built spanning tree is the solution of sub instance.
At first iteration: We only have x and 0, and they do have converging solution of the following.

$$\begin{tikzcd} X \arrow[r] & 0 \end{tikzcd}$$

Now, we prove that

if T is a solution to corresponding sub-instance, the adding a node that has path consistent with tree T will also be a solution to its own corresponding subinstance, given that

1. the consistent path in u, p has the highest rank for the node added and,
2. where v is the first node to be in the tree T, $p = (u\,v)p'$ and p' is taken by v. (i.e. p' is offered)

Here we introduce two notations:

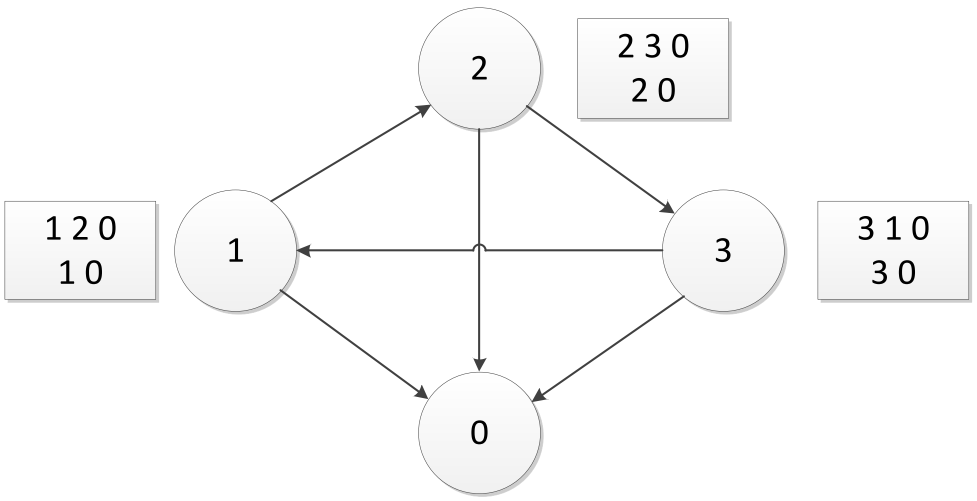
T_i : The tree obtained from i^{th} iteration.

S_i : The corresponding subinstance to T_i (i.e. the subinstance from i^{th} iteration)

Let's say that node u has been added to T_i at $(i+1)^{th}$ iteration to build T_{i+1}. Then, S_{i+1} is nothing but S_i with u is added. However, since the next hop from u, which is v (from 2 above), is taking p', adding u should not change the existing solution since u will take p, which also takes p' after hopping to v. So, given T_i is solution to S_i, T_{i+1} is the solution to S_{i+1}

Without loss of generality, we can say that at any i^{th} iteration, the tree built T_i is the solution to the corresponding subinstance S_i.

7.



This DAG will never converge. Each node prefers the path from their customer than their providers/peers (not shown but can be inserted easily and all available paths (satisfies export policy) are cust_only path of the subsequent nodes. (Note that all of their highest ranked paths give dispute path).

➔ We can infer that DAG needs to be acyclic, in order to guarantee convergence.