Internet Engineering Assignment

This assignment is for you to (manually, using pencil and paper, or similar) act as a router, and calculate paths through a small network.

You should do this **twice**, once using a Distance Vector protocol (eg: RIP) - that is, using the Bellman-Ford algorithm, and the other time using a Link State protocol (eg; OSPF) - that is, using Dijkstra's *Shortest Path First* algorithm.

To do this you must first download the network specification that corresponds to your enrolment number. Every student's network is slightly different, so do not expect your answers to be the same as any other student's. The data files were randomly generated by excluding some elements (some networks, and one router) from a larger data file.

In the file, you will find a list of 7 routers with names chosen from the set: **A, B, C, D, E, F, G, H** and **J**, Two of those 9 labels will be missing from your data. Each router has one line of the file data, its name is the first character of the line. With each router (on the same line) is a list of the IP addresses of each of its currently working interfaces (from 1 to 4 of them), and the cost of sending (or receiving) a packet via that interface. The cost of using an interface is the number in parentheses (brackets) following the IP address of the interface.

The cost is the metric to be used when determining the results of the routing algorithm – the aim is to minimize the cost of the path to reach every possible network number, where the cost of the path is the sum of the cost of sending a packet on each link that path crosses.

Whether you include the cost of sending to the destination link or not is irrelevant, provided your decision is consistent (you always include it, or always exclude it).

For this project, only IPv4 is used, and you should assume that every network is a Class C equivalent network (that is, that it has 24 network bits, and 8 host bits, in each IP address, or that its network mask is 255.255.255.0). This information will NOT be repeated in the data files.

You will need to determine which links exist, and which routers are connected to each link, from the IP addresses of the routers given in the data file. Each router will have one or more functioning IP addresses, one for each network interface of the router that is currently available for forwarding packets.

Each network will be connected to at least one router, but might be connected to more than one. Most networks will be connected to two routers. (No networks in this data set are connected to more than two.) No explicit information about the networks appears in the data file, you must deduce that from the information given about the routers.

Part One

First, take the data supplied, and draw a diagram of the network, showing the various links (with their network numbers), the routers, and the link costs.

Submit this diagram as part of your assignment results.

The drawing can be done by hand, or using some computer drawing tool. Being correct is important, being pretty is not.

Select one router, one of **A, B, H** or **J** from the network (those routers will be in all data sets). Select only a router that has at least 2 working interfaces (that is one that has at least 2 IP addresses given in your data file). If possible, select a router with 3 or more working interfaces (IP addresses). That router will be the router that you are to perform all routing calculations behalf of (that is, you will be pretending to be that router). Do not choose any of **C – G** for this purpose.

Mark the selected router clearly on your network diagram.

Part Two (Distance Vector Routing)

Show the packets your selected router receives, and sends, and your selected router's internal state (the routing table as it is calculated) at every step. Note that the router starts having no information at all, other than knowing the links to which it is connected, and their costs, Show step by step as the router moves to becoming fully configured using a Distance Vector routing protocol.

You should assume that all the other routers have also just started operating at the same time (the power to the entire network was just restored) so every router starts with knowledge only of its locally connected networks (and costs). You should also assume that all routers process updates at the same speed, using the same algorithms.

When showing the packets, the packet formats are not important. Nor is which distance vector protocol is used. You should show what data the router receives, for the routing protocol only, and from where it is received - other details are NOT important. Similarly, you should show what data is included in packets your router sends as part of the routing protocol, and to where those packets are transmitted, other details are NOT important. Technical protocol details, like the maximum amount of data that fits in one packet, etc, DO NOT matter. You can assume that when you want something sent, something else (not part of this assignment) will take care of getting the details correct.

You also do not need to worry about lost routing packets. For the purposes of this project, you can assume that all packets transmitted are correctly received and processed.

Since the actual protocol being used is not known here, other details such as packet timing are also not known. You should simply assume that all routers examine their current state at (approximately) the same time, each determines what it is to send, and to where, and each sends that data at approximately the same time as all the others. Each router receives data from others sending to it after it has already sent data to its neighbours. The incoming data is received, and processed, and this whole process repeats until there are no more changes (the routing table becomes stable) throughout the network.

Because we are not concerned with changes to the network state, (that is, all functioning networks remain functioning, and no broken networks are repaired) it is not important whether techniques such as split horizon are used or not (and poison reverse will never be needed). You can select to implement split horizon, or ignore it.

SHOW ALL STEPS in the processing for your router. DO NOT omit any processing until the system has reached stable state (the state where nothing changes for any of the input packets, and the same output packets are sent over and over again.)

Remember that the states of all the other routers will also be changing at each step – you are not required to show all of that state (though you may), but you must take into account what is happening at all the other routers in order to correctly determine what will be received from your router's neighbors. So, for example, if your selected router is X, and X has Y as a neighbor, and Y has Z as a neighbour (but Z is not a neighbour of X) then what Z sends Y will (partly) determine what Y sends to X, and that itself will be determined by what Z has received from its (other) neighbours.

For each data item received by your selected router from any other router (that is, for each single route entry in the update packet) you should show how that data is processed – if it is ignored for some reason, you should say why. If it is entered into your routing tables, you should show what data is entered, and what this will cause your router to do.

After showing the processing of the first 2 or 3 data packets from all neighbours in complete detail, you may, if you choose, skip some details for all later data packets – but be sure to always show the contents of each incoming and outgoing data packet, and the routing protocol data structure(s) as it (or they) exist after each incoming packet has been processed.

When the algorithm completes (when the routing data structures are stable) you should show the forwarding table from your router to every known network. (Known networks are those listed in your data file.)

Part Three (Link State Routing)

Show the complete Link State Database your router would have received if it were using a Link State routing protocol. (For this, assume the Distance Vector protocol of Part Two of this project never existed.)

Do not show the link state advertisements. That is, you can assume the process of distributing link state advertisements has completed, and the full link state database exists at your router before this part of your project begins.

Show how Dijkstra's Shortest Path First algorithm is used to calculate the paths from your selected router to all destinations that can be reached.

SHOW ALL STEPS.

Note this is important: **SHOW ALL STEPS**. Do not omit anything.

At each step the contents of the three data elements of the SPF algorithm (R, E and O) must be made explicit, and the next operation(s) to be performed explained.

When the algorithm terminates, you should have the lowest cost (shortest) paths from your router to every possible destination. Destinations are the various networks that exist in your network data file.

Hints.

In order to complete this project, particularly the part using the Distance Vector protocol, you will need to not only investigate what happens at your selected router, but at all other routers in the network. You can include all of that work in your submission.

You may notice that the first octet (byte) of the IP addresses assigned to the different links is never the same as that assigned to any other link. You may make use of that fact and record network numbers using just their first octet while processing the various data. (That is: you may use simplified single number network identifiers.)

You might also notice that the host ID part (last 8 bits) of each IP address identifies the router (A == 1, B == 2, etc) though this may not be of much assistance (you may however make use of this any way that helps.) Note, router J uses 9, not 10, as there is no router called I.

Submitting Results.

Students may submit the results of this project in a PDF format file (not MS-Word or anything like that) attached to an e-mail to kre@coe.psu.ac.th with a Subject header of

Internet Engineering Assignment 491234567

where the number is your student ID. The file name of the PDF file should also include your student ID – so use something like 491234567.pdf, Assignment-491234567.pdf, 491234567_internet_engineering_assignment.pdf, or my_name-491234567.pdf. The exact name is not important as long as it contains your student ID. It would be better if PDF file names do not include spaces, but that is not required.

Alternatively, you may submit your work on paper (either printout, or hand written). Or, you may submit part electronically and part on paper. In this last case, be sure to make it clear with each part that there is more of your project than the single submission.

If you submit more than once, make it clear whether you intend the later submission to replace the earlier, or perhaps the later submission is just a repeat of the earlier one because you are not sure that the earlier one was not lost, or that the later submission is an addition to the earlier one (so they should both be considered together) – in this case please use different filenames for the two submissions if submitting electronically.

For each e-mail submission you should receive an e-mail acknowledgment within a day or two. This will simply state that your e-mail was received. If this fails to arrive, you should take steps to confirm the arrival of your submission, or resend it.

In all cases, please make CERTAIN that your student number (and name) appear clearly on EVERYTHING submitted. If you submit paper, please write your student number on EVERY PAGE.

Deadline.

The deadline for this project is Sunday March 2, 23:59 (ie: before the start of Monday March 3rd).