1. Describe the routing problem
I'm not sure what it's asking here exactly, so I ha ve made some assumpthons.
The routing problem is as follows: what is the sho rtest path that a Router can send a packet to anot her router in the network?

2. Describe the solution implemented in your progra m
I used a python class Router which encapsulated a dictionary called routingTable where each item was a Distance Vector of it's own and it's direct nei ghbors. Also a very important part of this was usi ng a forwarding table which I didn't figure out ti ll much later. One thing you have to account for i s to be able to always account for your neighbors changing, basically you can have a link from A B w ith cost 4 and A C cost 5 and C B 4, but what if y our link from A B cost goes up to a high number 30 ? then the forwarding table is what tells you in o rder to send a packet (the fact that your distance vector is updating) you must send that U protocol message through C in order to get to B, also at t hat link cost there is convergence i.e if I were t o do a link cost change of A B 64, then A would ke ep exchanging messages with C until convergance ho wever I have also implemented poison reverse to be able to stop this, essentially C would be telling A a white lie that C is actually 64 (infinity) a way from B. Actually, in my example lets say C B w as 50 Therefore A would just stop after a single i teration and know that the fastest way would be to go through 50 + 5.

3. Are the timeouts occuring properly every 10 seco nds?
Yes they are, with a few negliggibale millisecond d ifference which can be seen

4. For each of the three test cases (test1, test2, and cti)

-*****NOTE: test4 is the same as test2 so we were t
old to ommit that one

cti- 5 nodes all links were 1, every node was at mo
st 2 away from farthest node.

test1- 5 nodes again except AB, AE, BE links forme
d a triangle where A initially only uses E as a ho
p to get to every other node in the network

test2- This one was a crazy one, there's ALOT of l
inks in this one, one t look at is the HB and HG l
ink, always routes through B and so I used that fo
r testing a lot. The main thing about this is all
nodes have at most 4 links so this really tests th
e dynamic capabilities of your program.

5. What routing tables did your algorithm converge
to? Are they correct?
I have provided an OUTPUT.txt file, and I have loo
ked over all the answers and yes they are all corr
ect.

6-7.In test1, what happens when link cost BE change
s to 2, the algorithm is
allowed to converge, and then BE changes back to 8.
 Is the behavior
different if the -p flag is used (poisoned reverse)
?
o In cti, what happens when the link cost AE link c
ost CD change to 64 and
the system is given time to converge? Is the behavi
or different when the -p
flag is used? Now change the two links back to cost
 1. What happens?

I have put these in the OUTPUT.txt numbered 4., 4.
 with poison reverse, and 5. I will explain it ov
er here as well

6. When BE changes to 2 the algorithm converges, a
nd then we switch BE back to 8 it takes a few extr

a iterations to finally get B to stop thinking it can get to A through B. However when we use poison reverse B right away knows it can get to A through E with 9 and then right away after all the other nodes update knows it can get there even less through hop C. So basically what the difference between poison reverse in this case is, it takes less iterations because B doesn't have to talk to c since c is essentially feeding B a "white lie" as described in the textbook.

7. In cti, what happens when the link cost AE link cost CD change to 64 and
the system is given time to converge? Is the behavior different when the -p
flag is used? Now change the two links back to cost 1. What happens?
without poison reverse you can see it slowly converging iteration-wise going up until the nodes A, B, C reach 64 to E, D and E, D reach 64 for A,B,C. When using poison reverse it drastically reduces these iterations so that you can see the nodes knowing of their entries right away (the ones I listed earlier). It is pretty cool how big of a difference poison reverse made in this case, versus the last case where it wasn't as big of a deal.

8. Describe any additional experiments you have performed with your router and the
results of those experiments.
I did SOOOOOOOOOOOOOO much testing, I tested a lot with all 3 tests but especially with test2, I would try to break the algorithm as much as I could and I mainly just wanted a perfectly working algorithm. For instance, on test2, I would make H to B into 64 then H to G into 64 then C to D into 30. And just almost all the other links that mattered a higher number. I made a test3 to fix an issue i was having with test2, basically it was just nodes H to A to B with counting-to-infinity issue. For cti, instead of cutting to infinity i tried cutting AE and CD to 5 each and was seeing what would ha

ppen and then for test1 I test a lot on cutting A's links to 64 and seeing how everything would behave. I did a lot more tests than just that but the main thing that helped me was test2, it's really what made me realize the importance of the forwarding table and why it's so important in the case of nodes having so many different links, just testing with cti and test1 is definitely not enough if you want something good.

9. Point out decisions that either I or you have made that could significantly affect
the behavior of this distributed routing algorithm
The select pattern doing timeouts every 10 second, I would probably give a higher timeout, because if you think about it it is important that we do give other nodes messages automatically but 10 seconds is a little short for things to constantly be messing up and it could affect the behavior. Another decision I made myself was using forwarding tables, I think the professor may have gone over it in class but I didn't have a textbook or do research on it and I basically just came up with the idea on my own- I called it something else but I obtained a textbook later because I was confused about some terminology. The forwarding table is so important for the dynamic aspect of the algorithm that if you were just sending it to every direct link neighbor without a forwarding table, you run into SO many issues which I was doing at first.

10. Point out any discrepancies between the behavior you observe in your program
and the behavior of D-V routing described in the textbook. How do you explain
the differences (if any).

The book is really vague, it says that a router is supposed to it's distance vector to all neighbors but it is too implicit about the fact that your direct link neighbors will not always be your neighbors in the sense of a forwarding table, your forw

arding table can potentially constantly change and the book is not explicit enough about this. It was really frustrating just having to take so much time to figure out the whole forwarding table issue I was having, my main issue with that was I wasn't updating my forwarding table every time I received a message which was the biggest key to fixing my problem and I spent way too much time on it. They account for such a simple example with 3 nodes and things can go so differently with a network like test2. Two things the book helped me a lot on was terminology and poison reverse, first I was confused what exactly distance vectors in relation to a routing table was, the book cleared it up. A distance vector is just a vector/array/row inside a routing table, you have your own and your neighbors. Getting that terminology down initially could have saved me much more time. Another thing was the whole concept of poison reverse being a "white lie" I think that sentence in itself really just hit me instantly and made me realize what the point of poison reverse was and why it worked to solve some problems of counting to infinity.

11. In your conclusion, assess how realistically your router models the function of a
real internet router. What simplifying assumptions have we made? What aspects
of the environment are different and might have significant influence on the
performance of a real router? Again, I encourage you to read RFC 1058 as
background.

If we are comparing RIP protocol, RIP protocol's infinite threshold is 16 hops and not just the link cost of 16, it uses a different concept than link cost and each router keeps track of those hops. Something we simplified very much was just sending U protocol messages and L protocol messages, one thing we don't take into account is how many hops do we want to be at a certain node if both paths ar

e the same cost? RIP protocol takes the lowest amo
unt of hops into account. It calls this "metric".
A real router sends packets and not byte streams,
they have a different packet format with that metr
ic present in it along with command, version, addr
ess family identifier. The datagram contains aroun
d 5 different commands for version 1 anyways. Also
, the RIP packet formats do not distinguish among
various types of address, in our case- we do know
the exact addresses of our neighbors. Also it uses
 30 second timeouts rather than 10 seconds. There
is also a garbage-collection timer that is set to
120 seconds.

12. Finally, please tell me how long each member of
 your team spent on this project â€“
individually and in group sessions.

I spent 3 days(not sure how many hours)... this wa
s mainly dealing with issues but I am proud that I
 was able to do this with 0 help from anyone, incl
uding 0 lecture or 0 professor, although I did ask
 him how poison reverse worked but if I had the bo
ok it would make more sense. Anyways, I made so ma
ny mistakes on day one and did a lot of useless te
sting but on day 3 I just did SO much testing and
found so many issues and was able to fix them. I w
orked extremely hard on this project, and got very
 little sleep the past 3 days but if I knew everyt
hing I knew now I could finish this in a few hours
. I made some really dumb assumptions in the begin
ning but I learned from my mistakes, and I'm glad
this project made me go and get the textbook. I al
so put a lot of good debugging code which helped m
e realized every single action every router was ta
king at any specific time, what messages they were
 sending, receiving. This helped with my testing i
mmensly.

I'm just glad I'm graduating this semester, with a
 pretty amazing project done even though I'm a lit
tle burnt out, this project reminded me why I love

computer science so much.

Workload: I worked on my own with 0 help as listed ^

Thanks for reading all of this, Kudra or Professor! :)