DTN Installation Instructions:

- Download .tgz files for DTN2 and Oasys from:
  http://sourceforge.net/projects/dtn/files/
- Download .tar.gz Berkeley DB 5.1 (versions higher that 5.3 and lower than 5.0 are not compatible) from:
  http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index-082944.html
- Extract the above downloaded files and place the created folders in a folder, say in a folder "project" in /home/username/Documents
- Rename the folder extracted from oasys-1.6.0.tgz (or any other version) from "oasys-1.6.0" to simply "oasys".
- Thus /home/username/Documents/project now contains three folders named "db-5.1.29", "dtn-2.9.0" and "oasys".
- Note: While executing the below commands, use "sudo" wherever required.
- Now we build the Berkeley DB. Change the working to directory to /home/username/Documents/project/db-5.1.29/build_unix and execute the two commands:
  $ ../dist/configure
  $ make
  This will build the Berkeley DB library.

To install the Berkeley DB library, enter the following command:
$ make install

- Change working directory to /home/username/Documents/project/oasys in the terminal and execute:
  $ sh configure
  $ make
- Change working directory to /home/username/Documents/project/dtn-2.9.0 in the terminal and execute:
  $ sh configure -C --disable-ecl --disable-edp
  $ make
- Now we need to choose a directory that will hold the files dtnd needs. Say this folder is called "dtn" and placed in Home. Thus the directory created is /home/username/dtn.

- Copy the file “dtn.conf” from /home/username/Documents/project/dtn-2.9.0/daemon to /home/username/dtn. Now edit this /home/username/dtn/dtn.conf and set the “payloaddir” to /home/username/dtn/bundles and set the “dbdir” to /home/username/dtn/db.
  i.e. change the lines
  storage set payloaddir $dbdir/bundles
  storage set dbdir $dbdir/db
  set dbdir “”

  to

storage set payloaddir /home/username/dtn/bundles
storage set dbdir /home/username/dtn/db
set dbdir “/home/username/dtn/db”

The eid(used for identifying endpoints i.e. computers) can be changed by editing the line:
route local_eid “dtn://john.dtn”

- Change the working directory to /home/username/Documents/project/dtn-2.9.0 and initialize the database by executing:
  $ sudo daemon/dtnd -c /home/username/dtn/dtn.conf --init-db

  Upon successful execution, the terminal should look like:
  $ sudo daemon/dtnd -c /home/username/dtn/dtn.conf --init-db

```
[1178218919.248562 /dtnd notice] random seed is 248552
[1178218919.248660 /dtnd notice] DTN daemon starting up... (pid 3268)
[1178218919.263195 /dtnd notice] initializing persistent data store
[1178218919.263343 /dtn/storage notice] creating new database directory
/var/tmp/dtn/db
[1178218921.130742 /dtnd notice] closing persistent data store
```

- The setup is now ready and we can start the dtn daemon. Change the woring directory to /home/username/Documents/project/dtn-2.9.0 and execute the command:
  $ sudo daemon/dtnd -c /home/username/dtn/dtn.conf

  This should give us a dtn% prompt. This means the server is up and running and awaiting commands.

- Leave this terminal open to keep the daemon runnning and open another terminal to execute dtn application codes.

- Some application codes are provided in the directory /home/username/Documents/project/dtn-2.9.0/apps

- More information about the DTN setup and how to use the above mentioned application codes is provided in /home/username/Documents/project/dtn-2.9.0/doc/manual/index.html

  In this manual, the relevant sections in the Table of Contents are:
  1) Getting Started -> Compiling DTN2
  2) Tutorials -> Start dtnd and sending a ping
  3) Tutorials -> DTN2 applications
  4) Tutorials -> Building a bigger DTN

  Manual for Berkeley DB is provided in /home/username/Documents/dtn/db-5.1.29/docs/index.html

Troubleshooting tips:

- While executing make for oasys, tcl library is required. Install it using "`sudo apt-get install tcl8.5-dev`".

- The first observable (ie to test the cost property of dtslr routing algorithm) can only be tested when both the links (from machine1) are in state –OPEN (and NOT even AVAILABLE) .You can check the state using "link dump" command.
  This is because if any link is in state AVAILABLE, then irrespective of its cost, it link will be used for further communications.
  Note: A link's state changes from open to available when a communication happens through it.

- More than often, when setting single/multiple links between machines for the second time onwards ("second time" is when- after having the setup made for the first time, closing the daemon and then next time when needed, again starting the daemon and setting the links up), an error will be thrown because the daemon remembers previously used link names and IP endpoints and in order to maintain consistency in log, imposes many restrictions (on names and endpoints) while setting up the links again.
  The best way to avoid this is to assign previously unused IP addresses to the endpoints/nodes/computers EACH time you want to start the daemon and setup the DTNetwork.

Assignment Question:

Setup a DTNetwork of 3 computers (same can be extended to add more nodes)

- Say we have three computers/machines running dtn daemons with their EIDs and IP addresses as:
  dtn://mach1.dtn ->192.168.229.141 :we will call this machine1
  dtn://mach2.dtn ->192.168.229.142 :we will call this machine2
  dtn://mach3.dtn ->192.168.229.143 :we will call this machine3

- Now we want to send a message/file from machine1 to machine3 via machine2. Thus machine2 acts here as intermediate node/router and forwards bundles/data received from machine1 to machine3.

- All the machines use DTLSR (Delay Tolerant Link State Routing) algorithm.

- Thus the network topology looks like:
  |machine1|-------✉|machine2|---------✉|machine3|

  Note: all the commands to configure routes, links, etc. are entered in the dtn% prompt. The ***route***, and ***link*** commands can be used to setup the topology and demonstrate the observables.

Observables:

- Using the application “dtnsend” on machine1, send a message from machine1 to machine2. Verify the bundle has actually reached machine2 by reading dtn% console or by running the application “dtnrecv” on machine2 and receiving the message. Now use “link dump” on machine1.The status of one of the two links should change to “available”. Verify that this is the link with the lower cost ie link_one.

- Now, close both the links between machine1 and machine2 on machine1 by either physically disconnecting the two computers or using commands to close the simulated links “link_one” and “link_alt”:
  > link close link_one
  > link close link_alt

- Again, using the application “dtnsend” on machine1, send a message from machine1 to machine2. A corresponding bundle should be generated which still resides on machine1 as there is no path to machine2.Verify using the command “bundle list” in dtn% prompt. Now open any link and the message should reach machine2. After opening any link, no bundles should remain on machine. Verify this by again using “bundle list” on machine1.

- Keep atleast one link from machine1 to machine2 open and keep link from machine2 to machine3 also open. Now send a message from machine1 to machine3 using “dtnsend” application. If the setup described above is correctly done, the message should reach machine3. Verify using application “dtnercv” on machine3.