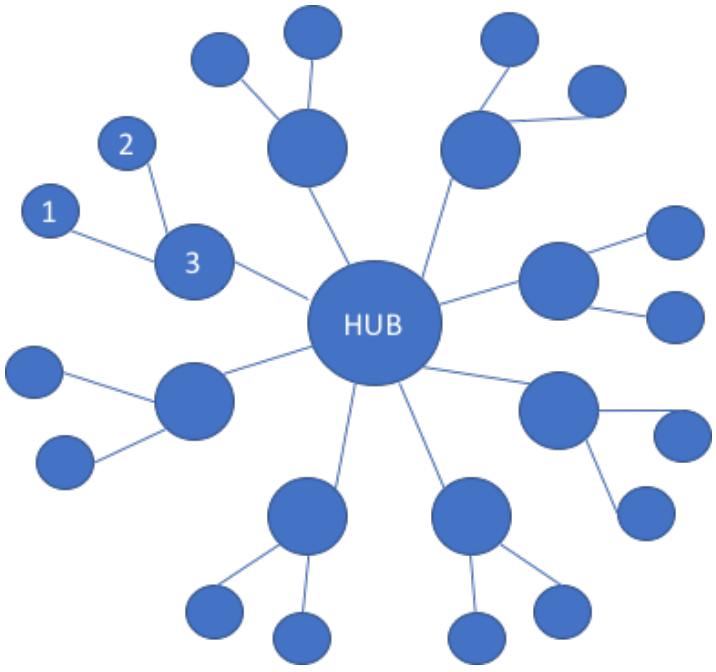
ECE 6110
Project 3

For this project, you will be using the distributed simulation capabilities of ns-3 to model the spread of an internet worm. You will also gain experience creating a custom point-to-point-layout helper and use ns-3's nix vector routing features.

1 You need to create a Point-to-point layout helper class modeled after the existing ones in ns-3. This helper will create the topology described below.
2 Your simulation must run on up 4 MPI instances and be configurable via command line. Details below.
3 You must enable nix-vector routing in your simulation
4 You will create 3 graphs (diagrams).
    a. Show the WALLCLOCK execution time for both conservative synchronization algorithms for each of the required MPI configurations.
    b. Show the WALLCLOCK execution time for both conservative synchronization algorithms, using 4 MPI ranks, varying the lookahead time.
    c. Show the infection rate as a function of time and scanning rate for the three different scanning patterns described in the paper. See the paper for examples.
5 You will follow the file naming specifications and hand in guidelines I give so I don't have to dock you silly points for not reading the instructions.

Base Topology



In the sample code provided, I have given you files to create your layout helper in. DO NOT RENAME THEM also DO NOT RENAME THE MAIN PROJECT FILE, please. The helper should accept 2 integers and 2 point-to-point helpers. For your submission, you can assume the integers are always going to be 8 for the inner and 2 for the outer which will provide the topology shown above. The Hub has 8 nodes connected to it and each of those nodes has 2 children. Use one p2phelper for the links from the hub to the first set of nodes and use the other for connecting those nodes to their children.

The links should be configured as following:

Hub->Inner	100mbps / 5 ms
Inner->Child	100mbps / 8 ms
Backbone (Hub->Hub)	1gbps / varied delay

The defaults are fine for the rest of the attributes on the device and the channels.

We need to create the illusion of "holes" in our IP range so the worm doesn't always correctly pick a real target. Use ns-3's uniform random number generator when creating your node layout. Assume each HUB to inner node link occurs 80% of the time and each inner node to child node occurs 60% of the time. If your worm decides to send a packet to a node that is not there, the packet must be routed as far as it can be before being dropped. For example, in the above topology, if the worm picks the IP address on node 1 and node 1 isn't there, the packet must route up to node 3 prior to it being dropped. You probably want to create functions in your topology helper class to assist you with this. You can assign IP address however you wish, but you should assume they would be contagious if your topology were "full." Not missing any nodes. Remember the IP address are on the network device not the node.

For your script, you should create 4 instances of your topology and connect their hubs with pointToPoint links which will be the backbone of the network. You should assign your topologies to MPI instances as following:

- If the number of MPI instances is 1, the four instances should be on that one rank
- If the number of MPI instances is 2, two should be on one and two on the other
- If the number of MPI instances is 4, each MPI rank should get one

You will need to write code so your script can detect the number of MPI ranks and assign your topology instances correctly.

Use the paper to help guide your modeling of the worm. We will only be considering UDP worms so don't worry about TCP. You can also disregard varying the infection rate. Choose one size. Look around to see if you can find an appropriate size. It doesn't matter what port you use. You do need to model the 3 different Scanning patterns described in the paper. Use ns-3's random number capabilities where needed.

I recommend that you use Google or grep to search the ns-3 codebase for examples. They do exist. If you don't know how to use grep, it's worth your time to learn. Lots of examples online. Use Google to find them.

Your simulation script has to accept the following command line arguments. I will be using a script to run your code so if you don't name them correctly, or don't capitalize them correctly, your code won't run and you will lose points.

ScanRate	Rate to generate worm traffic (5, 10, 20) ms, default 5
ScanPattern	Scanning pattern (Uniform, Local, Sequential), default Uniform
BackboneDelay	ms delay for backbone links (lookahead), pick a valid default
SyncType	Conservative algorithm (Yawns, Null), default Yawns

Assume all of the instances of your topology are on the planet Earth and pick 5 appropriate values when varying the delay on your backbone.

The provided skeleton code should be extracted into your scratch folder. Creating a subfolder in scratch is how you link multiple files into one program in ns-3's build system. When you are done, put your graphs into this folder with your completed code, compress it and submit on TSquare. USE .ZIP or .TAR.GZ, DO NOT USE .RAR. We're not downloading warez on newsgroups. You submission should be p3.(zip/tar.gz).

Have fun! Ask questions. Apologies for any typos, I've caught my son's cold and my head feels numb. Ask for clarification if something isn't clear. I'll make an announcement if anything changes.

ps

Installing OpenMPI

MAC:	brew install openmpi
Ubuntu:	sudo apt install openmpi openmpi-dev
Windows:	Ha! Yeah right.

Configuring ns-3 with mpi

./waf configure --enable-mpi