Instructions for JNW2 Project LAN1 – Backoff for Contention LAN

See chapter 6 of *Understanding Internet Protocols.* Project LAN1 is basically the same as given at the end of Chapter 6. However, in JNW2 the working of the LAN is in ContentionLan.java and LanInterface.java and the binary exponential backoff is in Backoff.java.

Your assignment is to complete Backoff.java based on the algorithms in the comments it contains, which are essentially the same as those in *Understanding Internet Protocols.* Your assignment is to code method binaryExponentialBackoff() for the JNW2 contention LAN. To understand what is needed, first read the comments in ContentionLan.java and LanInterface.java to learn how the JNW2 contention LAN works. Then read through carefully all of Backoff.java. Then you should be ready to code and test binaryExponentialBackoff().

To debug your Backoff.java, right-click on the JNW2 project in NetBeans and select "Properties" and then "Run". Use the Main Class option to select JNW2.interfaces.Backoff. This will let you test with the main() in Backoff.java, which prints out the results of repeatedly invoking binaryExponentialBackoff().

After debugging, you will run JNW2 as a simulator to complete LAN1. To test your Backoff.java in a simulation, right-click on the JNW2 project in NetBeans and select "Properties" and then "Run". Use the Main Class option to select JNW2.RunSimulation; also, type under Arguments "LANCollisions.txt" to load the appropriate network configuration.

When it is working correctly, the first two "email" messages start at the same time so there is a collision, but the messages are delivered properly due to the backoff function. JNW2 output identifies the collisions and resulting jamming frames. Also each message is displayed by JNW2 when sent and also when received at the application layer. The collisions and the data packets email are counted in the statistics at the end of the simulation run.

You will see several kinds of Events that JNW2 uses to simulate the contention LAN. One is LanSendRecycleEvent, which is how JNW2 does a retry if the LAN was busy or contention is experienced at a sending interface. A second is DlcSendWakeupEvent, which is how JNW2 revisits the DLC Send function to check for frames waiting in the queue. A third is Jamming Frame, consisting of 48 characters (alternating 1 and 0), which the interface that detects the collision sends to itself immediately after cancelling the frame it was sending. You should trace through the simulation to understand how each event triggers a new one in Discrete Event Simulation. That is the way JNW2 works.

Submit your Backoff.java and a copy of the output produced when you run the simulation with the LANCollisions.txt configuration. Be sure to include your name in the code comment provided.