1. We use a link for the shortest path computation only if it exists in our database in both directions because the wireless communication needs the bidirectional connection in order to have that reliable connection. For example if A->B is linked but B->A is not then A can hear B,but B can’t hear A. If we used a directed link AB when BA doesn’t exist packets could be sent but B wouldn’t be able to reply or acknowledge the message was delivered.
2. Yes our routing algorithm produces symmetric routes because it’s a spanning tree and since there is no loops the nodes only talk linearly to each other.
3. If a node advertised itself as having neighbors, but never forwarded packets the packets would just be dropped. Since the network thinks the routes exist but they don’t actually work. The way to solve this issue would be to have acknowledgments that a neighbor receives a packet but doesn’t forward it or if there are no replies after a couple of tries just to mark that specific route as a failed route.
4. If link-state packets are lost or corrupted the nodes that are missing LSA wont appear in the database, some packets may go through routes that are not the most efficient.
5. If a node alternated between advertising and withdrawing a neighbor every few milliseconds it would cause routing tables to be constantly changning and there could be packet loss because the routes would be changing. In order to deal with this would what we already have in our code with rate limiting LSA generation at 10 seconds with the linkstatetimer.