1. We use a link for the shortest path computation only if it exists in our database in both directions because bidirectional links require both nodes to broadcast to each other. Just because A can hear B doesn't mean that B can hear A. This is why both directions need to exist. If we used a directed link AB when the link BA does not exist, it would cause a unidirectional link meaning that A could talk to B, but B can't say anything back. This means acks may not be received and cause inconsistencies.
2. Yes my routing algorithm does follow a symmetric route, this is because the topology that is being used is a tree and its connected in a line, so there is only one possible direction that the packets can flow.
3. If a node advertised as having neighbors but never forwards, it would cause a "black hole" meaning packets would flood to it but then never come out. To deal with this, you could add a timeout time, so packets timeout and drop instead of piling up.
4. A corrupted or lost link state packet would cause some inconsistencies in the LSDB, but since it is constantly updating, the inconsistency will be resolved by itself.
5. If a node alternated between advertising and withdrawing a neighbor, every few milliseconds it would cause a bunch of LSA to be sent. To deal with this, you could put timers to limit the amount of forwarding packets that can be sent within a certain time frame.