Why does a node need to know who are its neighbors?
- So that a node can transmit its packet to the sink node (or Base Station).
- Just knowing your neighbors is not enough to get your packet routed to the sink node.
- Knowing which neighbors is "closer" to the B.S. than yourself would help in routing the packet to the B.S.
- "Closer"?
  - Geometric (Euclidean) distance.
  Or is it in terms of something else?
"distance"
- # of hops
- consider also congestion.

There are situations in which forwarding the packet to closest neighbor doesn't work.
To construct a Routing Tree rooted at the BS.

How to construct this routing tree?

Topology graph
- Construct a Spanning tree
- Minimum cost
- Spanning tree MST.
Since topology can change due to node failures & mobility we need a distributed & dynamic protocol to construct a routing tree.

1. Approach 1: Wave propagation
   - BS starts the tree construction protocol by sending its neighbor TREE_Construct message.
   - Other nodes: If I heard a TREE_Construct (d) then send TREE_Construct (d+1) \( \Leftarrow \) do this only on the 1st receipt of TREE_Construct message & ignore all others.
How does a node know that the TREE-CONSTRUCT message is a "fresh" or "latest" one and not an "old" message?

- Not a problem - everyone (node) has a clock that keeps track of the periodicity of TREE-CONSTRUCT messages. Every \( t \) period, a node can expect to receive a fresh TREE-CONSTRUCT message.
TREE_CONSTRUCT (hop distance, wave, $+t_{BS}$)

$D = 0$

$D = 1$

energy v/s latency

issue?
WSN is a distributed system
- we assume that there is no global clock and so relying on the assumption that clocks are perfectly synchronized for proving correctness of your distributed protocol is not a good idea.
To make tree routing protocol more resilient to faults:
- maintain a list of "closer" neighbors
- if my primary "forwarder" fails, then use other closer neighbors on the list to forward my message to the B.S.

Stop & Wait protocol for reliable transmission:

Sender recv.

If # of retries exceeds $K$ times, then give up.

Wait for ack if time out, resend the msg.

msg → ack
Longer you transmit the more # less transmission you possibly interfere with.

Longer transmissions are not good in terms of energy consumption as well as interference with other transmissions.
Power control nodes can adjust their transmission power.

How to minimize your transmission power while ensuring that the network doesn’t get disconnected.