1a)

i) Proactive routing maintains routes even when not accessed. Reactive makes the routes on demand.

ii) Distance vector protocols use local knowledge of neighbours while link state protocols have knowledge of the entire network. DIstance vector protocols converge slower and also suffer from the count to infinity problem. Link state protocols use flooding while distance vector protocols don’t.

1b) Beaconing establishes routes between each sender and receiver.

1c) Objective function is used to minimise cost between sender and receiver? Used to update routes when we get a better option and therefore we slowly converge to optimal routes between each sender and receiver.

1d) New\_etx = parent.etx + link\_quality\_to\_parent. RSSI can be one measure of link quality

1e) Reset ETX so we can beacon more frequently and routes can be quickly updated. Could also happen if a problem is detected in the network, or if a node is added or removed from the network.

1. ETX greater from child to parent -> indicates loop, inconsistancy so increase beacon period
2. Parent ETX grows significantly but no loop -> should send more beacons to try and find better route.

1f) i) *ticks*. Refractory period. Adds randomness. Exponential backoff if nothing happens so we save energy.

ii) instance->dio\_next\_delay, instance->dio\_redundancy, instance->dio\_counter

iii) Increase dio\_redundancy and lower delay?

iv) Lower instance->dio\_redundancy? so we transmit less as it is more likely that our neighbours will also have the same data and we reduce congestion

2a) Advantage is we can place them almost anywhere as they don’t require a mains connection. Disadvantage is that the run time is finite and typically short before batteries need to be replaced.

b)

i) 2 + 3 + (18060 \* 3) + (5 \* 3) = 54200mJ

ii) 18060 - (18060\*0.2) \* 3 = 43344mJ 20% duty cycle means it is on 20% of the time.

iii) Energy usage per hour is (3600 \* 8) + 3 = 28803mJ. Solar power per hour is 50J but only starts at 6. So it will die after around 28.1 hours ish

iv) ~84%? Although I didn’t take the initial energy into account which will probably change the answer slightly

c) Again we need to perform the averaging step here and only send 1 message instead of all 10?

3a)

i) Transmissions occur at the same time and the packets ‘collide’. Usually neither packet will get to its intended receiver.

ii) 2 senders can communicate with a receiver but not with each other so they might transmit at the same time causing interference.

iii) Preamble size. Some MACs use preambles to notify receivers that a sender wants to send and to keep the receiver awake. The size of the preamble depends on the duty cycling of the receiver.

iv) sleep-listen schedules of sender and receiver don’t match because they are duty cycling and could have been started at different times.

b) Sender listens to the channel and doesn’t send any preambles. It waits for small beacons to be sent by the receiver. The sender sends these beacons in a period defined by its duty cycle and waits for a response. The beacons act as time slots and can therefore significantly reduce collisions as senders only send in their slot. Because the receiver knows its neighbours from its routing tree it can assign one beacon per sender and maybe cycle these in a round robin manner. This way only 1 neighbour will send at a time.

c) Simple non-persistent CSMA would probably work. When we’re ready to send, check if the channel is idle or busy. If it is idle send immediately. Otherwise wait a random amount of time and repeat.