Low Power Wireless Communication For The Internet Of Things

Diego Oniarti

Anno 2025-2026

Contents

1 IEEE 802.15.4

This is an old (2003) low power standard that is still being updated and improved to this day. It is also the foundations for other standards (e.g. ZigBee and WirelessHard).
It does **not** provide a MAC layer and it operates on a short range (10-15m).

IEEE 802.15.4 operates on the 2.4GHz band, overlapping with WIFI and other protocols that can generate interference.

2 WSN - Wireless Sensor Network

2.1 Applications

There are many possible applications for WSN, some of them being: `Wildlife monitoring`, `Glacier monitoring`, `Volcano monitoring`, `Cattle herding`, `Ocean monitoring`, `Vineyard monitoring`, `Cold chain monitoring`, `Rescue of avalanche victims`, `Vital sign monitoring`, `Tracking vehicles`, `Sniper localization`, `Tunnel monitoring and rescue`, `Industrial processes`, `Smart cities`, and others.

2.2 Classification

Wireless sensor networks can be divided by different criteria:

- Goal: sense-only vs sense-and-react
- Interaction pattern: One-to-one, One-to-many, Many-to-many, Many-to-one
- Mobility: Static, Mobile nodes, Mobile sinks
- Space: Global, Local
- Time: Periodic (fixed or changing period), Event driven

2.3 Operating Systems

The mainstream ones are TinyOS, RIOT, FreeRTOS, Mynewt, Zephyr, and Contiki.

They offer basic run-time support for application programs but not for user interaction.

2.4 Networking Layers

In a classic distributed system the programmer only has to worry about the application layer, the actual program that needs to run on the machines. In WSNs, we often have to deal with the entire stack (i.e. MAC, Routing, transport, and application).

Routing This dictates how the messages are supposed to travel across the network. Do we want a multi-hop mesh? Do we keep one parent or multiple ones? What traffic pattern do we prefer?

MAC This coordinates the link-level communication. The two common approaches are CSMA (carrier-sense multiple access) and TDMA (dime division multiple access).
Two major problems at this level are the need to synchronize sender and receiver, and dealing with changing topologies.

Also, this layer has the responsibility of keeping the radio turned off as much as possible to save energy.

2.5 Typical requirements

The three main requirements (or metrics) that are required by WSNs are

1. Network lifetime
   This is defined as the time until some fraction of nodes runs out of power
2. end-to-end reliability
   This is measured as the percentage of packets being received or lost
3. end-to-end latency
   Some application can even give an hard boundary on this metric

Not all applications require all three of these attributes. Monitoring systems may only care about lifetime and a good reliability, while distributed control systems focus more on latency and a high level of reliability.

2.6 Typical problems

The things that make communication hard usually are:

1. Limited range, leading to multi-hop
2. High power consumption from the radio compared to everything else
3. Instability of the links, both over time and space

3 MAC - Medium Access Control

As the name suggests, the MAC regulates which node has access to the shared channel (or medium).

Medium access control for wireless sensor networks is slightly different from the one for general purpose systems. It trades performance for energy consumption (within reason). Throughput, fairness, and often latency are of secondary importance.

The size of the data packets being sent is often small (in the order of bytes or tenths of bytes even), so the size of the headers has more impact in percentage. This can create a significant overhead, also contributed to by control messages like RTS and CTS.

MAC protocols in some cases (e.g. event-based applications) must account for bursty data patterns, in which a node might remain silent for a long time and sporadically send a burst of data.

3.1 CSMA - Carrier Sense vs Collision Avoidance

Carrier Sense Listen on the channel to check whether or not there is a communication going on already. If that is not the case you can transmit.
Collisions can still occur since the time to start a transmission is not zero. In that case the nodes back-off by a random amount and retry the transmission.

Another problem with this approach is that of the *hidden terminal*. This is a common and well known problem in which two nodes A and B are in range of node C but not of each other. When A and B sense the carrier they see it as being clear (since they are outside of each other's range) so they both transmit to C, which gets two interfering signals.

Collision Avoidance Add some additional control messages (i.e. `request-to-send` and `clear-to-send`) to address the hidden terminal problem.
If a node A wants to transmit it sends a `request-to-send` tagged with its id. C receives it and broadcasts to its neighbors a `clear-to-send` with A's id. Now A knows it can send it's original message, while C knows it can't send anything because it received a `clear-to-send` with someone else's id (A's).

3.2 TDMA - Time Division Multiple Access

Time divided in *frames* which are further divided in *slots*. A typical frame layout has slots for:

- traffic control: meta-information from the base station to the nodes
- downlink: to send application data from the base station to the nodes
- uplink: to send application data from the nodes to the base station
- contention period: to allow new nodes to join the system

3.3 Slotted access

A sort of mix between CSMA and TDMA. All nodes wake up and go to sleep together to ensure synchronization. The awake time is short and the nodes must in some way decide how to handle pending packets during this contention period. This is prone to collision.

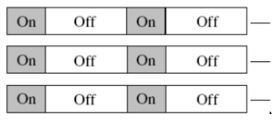

Figure 1: Slotted access

3.4 S-MAC

This is an improvement on slotted access, aiming at reducing collisions.
This is done by roughly synchronizing the wake up clocks of just neighbors, not all the nodes. The active period is divided in two parts: a portion to exchange synchronization messages, and one to exchange RTS, CTS, and data messages.

Furthermore, SYNC messages may only be sent in some awake periods (1 every N for example) to reduce collisions.

This MAC protocol does not guarantee fairness or low latency.

3.5 D-MAC

The *Data-gathering* MAC breaks the ISO/OSI layer abstraction, utilizing routing information to inform the MAC.
The nodes are activated with a slight delay across layers, with the bottom two activating first. Each subsequent layer is activated slightly after the one below.

This approach reduces both latency and contention, but it only really works well in static networks.

3.6 B-MAC / LPL

Low Power Listening is used to limit the power consumption on the listener while moving the blunt of the work on the transmitter. The receiver sends a long preamble (at least longer than the sleep interval)

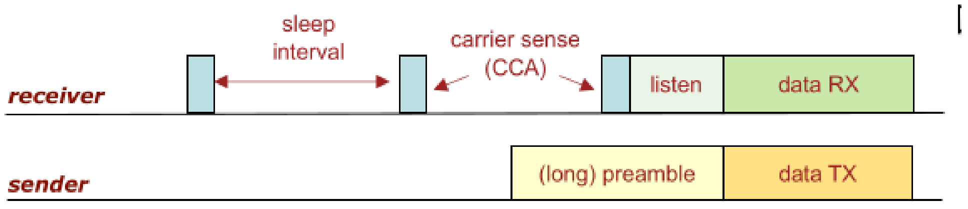

Figure 2: fig:B-MAC

followed by the message. The receiver periodically checks the channel. If it sees the preamble being transmitted it waits for it to end and listens to the message.
The preamble also contains header information, like the message target. So other listeners that wake up can see the preamble destined to another node and resume sleeping.

A long preamble consumes a lot of energy in the sender, while a short preamble requires the listener to wake up more often. Long preambles also make it more likely for collisions between transmitters.

3.7 CCA - Clear Channel Assessment

Checks if the channel is clear for transmission. Must be able to tell if what the node sees on the channel is noise or the transmission from another node.

3.8 LPP - Lop Power Probing

This is the "opposite" of LPL. The receiver sends out periodic beacons (Ready to receive). When a sender wants to transmit it listens for the beacon and then sends its message.

With the long preamble on the sender being replaced by a long listen, this reduces both power consumption and the risk of collision.

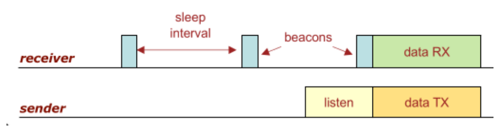

Figure 3: fig:LPP

3.9 A-MAC

Radio probes trigger ACK non-destructively thanks to tight timing.

3.10 Z-MAC

Zebra MAC "switches" between CSMA and TDMA depending on the contention on the channel in each given time.

The two modes formally are HCL (High Contention Level) and LCL (Low Contention Level).
In High Contention the transmitter only sends messages during its allotted time slot (TDMA), while in Low Contention it checks the medium for ongoing transmissions and waist a random backoff if it hears any (CSMA).

4 Routing

In normal contexts, routing would be an invisible part of the network protocol to the developer, simply seeing a socket as an interface to the network. In WSN however, the layers are blurred, requiring the developer to worry about routing as well.

4.1 LEACH

Low-Energy Adaptive Clustering Hierarchy is a routing protocol organized in *rounds*, each one further divided into two *phases*.

Setup Phase

- each node has a random probability of becoming a Cluster Head (CH), and it broadcasts an advertisement.
- Normal nodes get the advertisements and chose an head, effectively joining its cluster.
- The CH defines a TDMA schedule to communicate with the nodes in its cluster and it broadcasts it to them.

Steady Phase Normal nodes send their data to their local head. The head collects messages from the nodes, merges them in one bigger message, and sends this fused message to the base station.

4.2 Directed Diffusion

Data-centric routing for many-to-one communication with multiple topics.
This approach supports data in `key:value` format, with multiple sinks that may be **interested** in different attributes.

Each sink floods the system with its interest, allowing the nodes to build a reverse path tree for each interest (gradients). When a sensor has to report some data in just sends it along the “gradient” for that specific key/topic.

The trees must be periodically rebuilt/reinforced through periodic flooding. Another important decision is the metric with which a node chooses its parent. Some common options are

- hop count
- link quality
- workload and residual every
- application information

Workload can greatly affect the lifetime of a node. Nodes closer to the sink tend to route messages from more nodes, leading to higher battery consumption.
One solution might be load balancing, making different nodes take turns based on their estimated lifetime, but this can be complex to implement. Another solution can be to simply instal bigger batteries on the nodes closest to the sink. Lastly, we can use in-network aggregation to reduce the number of communications.

4.3 MintRoute

This is the routing protocol in TinyOS 1. It periodically sends out beacons from the sink that are then retransmitted by the lower nodes. The metric is

$$m = m_p + \frac{1}{LQI_p^3}$$

where m_p is the metric transmitted by the parent and LQI_p is the Link Quality between the node and the parent.

4.4 CTP - Collection Tree Protocol

The routing protocol in TinyOS 2. It uses ETX instead of LQI to determine which parent to follow.

4.5 MUSTER

Routing protocol used for multi-source multi-sink routing. Usually in this scenario each sink would have its own dedicated tree, but this solution is very expensive.

MUSTER aims at overlapping as much of the trees as possible, and collect multiple readings in the same message, to reduce energy consumption on heavy forwarders.

4.6 RPL

A quite heavy protocol that supports `multipoint-to-point`, `point-to-point`, and `point-to-point` communication. It stores the routes as a DODAG (Destination Oriented Directed Acyclic Graph) and keeps track of an objective function metric similar to the CTP one.

The protocol can operate in two modes: Storing mode, and non-storing mode.

Storing Mode: Every node stores the routing table for its own sub-DODAG. This can take up a lot of memory on the nodes.

Non-storing Mode: Only the sink keeps a routing table of all the nodes. To collect information the messages need to accumulate the path that they have taken, leading to possibly very large messages.

Point-to-point routing differs slightly in the two modes. In storing mode a message from node A to node B only needs to travel up to the lowest common parent between the two. In non-storing mode the messages have to travel from A up to the sink, and from the sink down to B.

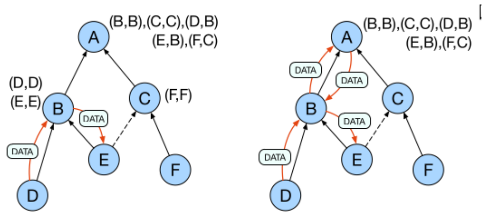

Figure 4: fig:RPLptp

Problems This protocol is made quite heavy by its need to support IPv6, which adds to the size of the packets, along with the memory occupied by the routing tables.

4.7 Orchestra

Protocol used in industrial scenarios that tries to be both flexible and reliable. It does this by autonomously generating TSCH schedules informed by the routing state.

The TSCH aspect makes it reliable, while the RPL information makes it more flexible.

3 slot types

1. Rendez-vous: this one is for discovery and routing. It is contention based.
2. Receiver-based shared: This one is contention based too. Each node gets a single receive slot and it's used for unicast.
3. sender-based shared (/dedicated): Also for unicast. Each node gets several receive slots, potentially wasting energy but reducing contention.

4.8 Opportunistic forwarding

Routing technique based on broadcasting instead of unicast. Each node broadcasts its messages, and if someone else can receive them, they forward them further. If a node receives a message and sees another node already forwarding it, it drops it.

4.9 ORW - Opportunistic Routing in WSN

It still relies on a routing metric, except this time it is EDC (Expected Duty Cycled Wakeups).

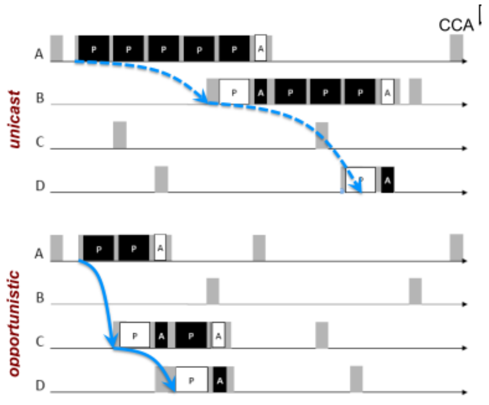

Figure 5: fig:ORW

4.10 Glossy

This protocol broadcasts messages without building any topology first. The messages are simply broadcast from one node, then all the nodes that received the message broadcast it as well. This protocol requires *accurate network-wide synchronization*, so that the messages may collide non destructively.

This makes the propagation of messages fast and reliable (due to the inherit redundancy), but requires synchronization assumptions. It is also tolerant of mobility.

5 Data Prediction

In some circumstances we can reduce the amount of data being sent over the air by only sending

enough to build a *model*. The aggregator can sim-
ply assume data will keep coming in conformance to that model, and the sensors only need to send outliers.

5.1 DBP - Derivative Based Prediction

This approach works well with linear data (or data that is linearisable over short spans of time). The model that is built and predicted only requires a point and an angle, so it is quite simple. Determining of a data point is in the error margin is also computationally simple, since we only need the distance from the point to the line.

We set a *value tolerance* and a *time tolerance*. We start sending data again when we get values outside of the tolerance for too long of a time.

One drawback with this approach is the face that the data being transmitted is

- unpredictable
- bursty
- critical. Since losing a message has a great impact on the system

All these properties are well server by the glossy stack.

5.2 Glossy

Glossy is a network stack designed to collect sparse and aperiodic data. This type of traffic can be generated by data prediction algorithms or by networks used for event handling and command issuing instead of data collection.

The system is built atop the glossy floods. The nodes flood their data and the receiver floods its acknowledgment. If a node receives the ack to its own message it stops transmitting. If it receives the ack to the message of another node it transmits its own message again.

6 LPWAN - Low Power Wide Area Network

Low power WANs can serve large areas (in the tens of kilometers range) for multiple years, but with very *low data throughput* and *high latency*.
They often use simple star topologies, where many nodes can be served by the same base station. This makes the network stack more simple, not requiring multi-hop, and puts the heavy work on the base station instead of the devices.

Narrowband Transmitting over a narrow frequency band allows for more links to efficiently share the spectrum, and makes noise have a smaller impact on the signal's power. This also allows for simpler receivers and antennas.

Ultra-narrow band signals take this to an extreme, increasing latency significantly as the signal is kept in the hundreds of hertz range.

Spread Spectrum takes the opposite approach to narrowband, spreading the signal over a much wider band of the spectrum. This makes the signal appear almost like noise, making it less susceptible to jamming and actual noise, but requiring a more sophisticated decoder on the receiver's end.

The two common implementations of this are DSSS (Direct Sequence Spread Spectrum) and CSS (Chirp Spread Spectrum).

7 Localization

7.1 Localization vs Detection

Proximity detection only requires to measure or determine the distance between two actors, while localization needs to provide (x, y, z) coordinates of a actor in a given coordinate space.

7.2 Anchors

Localization techniques assume the existence of some reference point(s) whose location is known. These beacons or anchors are used to build the coordinate system the localization will take place in.

7.3 Device-based vs Device-free

Device-based localization requires the actor (usually a person, animal, or robot) be equipped with a device that can communicate with the anchors.

Device-free localization is usually based on the way an actor itself disturbs the wireless communication between beacons. Radar and lidar are also device free localization methods.

7.4 RF-based technologies

RF-id is cheap and passive, meaning it does not require a battery. It's range is very poor, some times requiring direct contact.

WiFi is ubiquitous and supports both ranging and communication, but it consumes a lot of power.

Bluetooth is ubiquitous in some contexts, can transmit data as well as ranging, and consumes less energy than WiFi, but its localization is rougher and its range is shorter.

Ultra-wide band is accurate and operates over a larger range, but it requires dedicated infrastructure to work.

7.5 Computing the position

The position of an actor is usually calculated starting from its distance from the anchors. If we plug this value into the equation of a circle $(x_i - x)^2 + (y_i - y)^2 = d^2$ (or into that of a sphere depending on context) we can build a system of equations. This is then solved with the least squares method to get the actor's position.

$$residue = \frac{\sum_{i=1}^{n} \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} - d_i}{n}$$

If the residue is too large the measurement is discarded.