Embedded nano mesh – network principle explanation.

1 – Protocol goal.
2 – Network topology
3 – Node description
4 – Node operation principle
5 – Packet structure and configuration

1 – Protocol goal
This protocol is designed to be general use, multi purpose low speed messaging protocol between embedded devices. No-STD friendly.

2 – Network topology
The network is designed to be one ranged – Mesh topology. It is supposed to be self-healing, self-configurable network. All nodes are equal in operational functionality.
Each device, which runs the protocol – is later called Node.

3 – Node description
Each node – is the part of mesh network. Each node has it's own numeric address. Nodes are communicating by using structures named – Packets.
This protocol is also friendly to be ported to many platforms.

Packet can travel trough the network with a limited number of hops. This limit is set during sending any type of message by setting parameter "lifetime" parameter to specific value.

Hop – means the process of the packet being sent by node trough the radio ether to other node, and then received by other node successfully.

The message routing is done in the dumbest way as possible. The packet is spreaded trough the network in the similar way – as the wave is spreaded in the pool, bouncing out of the edges and fading down. Fading is made by the Lifetime parameter, which is kept as the field of the Packet. Lifetime is decreased by 1 during each hop. Once the Lifetime is debleaded – the packet dies and no longer re-transmitted trough the network.

Protocol uses USART interface in order to use radio modules to speak trough the radio ether. All radio modules should be configured to communicate on same frequency. All the other routing will be handled by the protocol.
In order to port protocol to the new platform PlatformSerial interface should be implemented also for this platrofm.

Protocol uses timing counting in milliseconds since the program start. It uses timing counting in order to know what time to keep between packets being sent to ether. This timings are needed to give some time for other nodes to transmit into the ether in order to void collisions. So Node sends the packet once at a time and then listens for incoming packets during the "Listen period".
In order to port protocol to the new platform PlatformTime interface should be implemented also for this platrofm.

Node can communicate with other nodes in several ways:

1. By sending packed message to specific node address.
2. By multicasting packed message. Multicasting means, that the message is addressed to all the nodes of the network. Multicasting also means, that the node, which receives multicast – treats the packet as addressed to the current node, keeps a copy, and then sends a copy further to other nodes of the network.
3. By sending packed message with “ping” flag being set. This will force receiving device to unpack and keep the message as usual, and to provide automatic response with “pong” flag being set.
4. By sending packed message with “send_transcation” flag being set. This will force receiving device to unpack and key sending packed message witep the message as usual, and to provide automatic response with “finish_transaction” flag being set.

4 – Node operation principle

Node relies on interfaces PlatformTime and PlatformSerial being implemented and provided during call of some methods as generic parameters. It is made to be statically linked. It is made to speed up code on some weak platforms, as arduino, etc..

Node – is the program structure, that has few internal queues for the messages:

1. Queue for the messages to be sent from the current Node. (Sending queue).
2. Queue for the messages, that were caught by the current node and were addressed to other nodes. (Transit queue).

Node also provides few interfaces to operate:

- “new” – Interface to configure new instance of “Node”. This method sets few important things of the Node:
  - Listen period
  - Address (numerical address identifier)
- “send” – Interface to (send the message to specific device \ multicast the message to all devices).
- “send_ping_pong” – Interface to send message, and force receiving device answer back with “pong” at least once. This interface requires PlatformTime and PlatformSerial generic parameters.
- “send_with_transaction” – Interface to send message, that guarantees, that the receiving device has acted on sent message only once. This interface requires PlatformTime and PlatformSerial generic parameters.
- “update” – this is the most important method. It does all internal work with internal queues. It sends stacked messages from the queues, and routes the packets. This interface requires PlatformTime and PlatformSerial generic parameters.

5 - Packet structure and configuration

Configuration of types of fields can be found in: src/mesh_lib/node/packet/types.rs
Configuration of size of data to be sent can be found in: src/mesh_lib/node/packet/constants.rs
The packet consist of next fields:

- source_device_identifier – this field contains the address of device, which has sent that packet. Currently is used to filter out packet duplications.
- destination_device_identifier – this field contains the target address, which shall receive the packet.
- Id – this field contains packet serial number. Serial number is used to distinguish packets from each other in the ether. Currently is used to filter out packet duplications.

- Lifetime – this field contains the number of hops, that the packet is still able to make.
- Flags – Integer type binary flag field. Constants, describing the fields are placed in src/mesh_lib/node/packet/constants.rs file.
- data_length – Field that contains numeric value, that says how many bytes are packet within the packet.
- Data – vector of bytes of transferring data.
- Checksum – is the sum, that is calculated to check – if the packet contains correct data. This significally reduces chance of the data being corrupted during the transmission.

Important! In case of change data types of fields of Packet, serialization and de-serialization parts also should be checked in order to keep consistency of packet on all devices. All nodes should have same version of firmware installed in order to be able to communicate.