# Mesh Protocol

## Server

* The server constantly checks for new nodes and that all of the known ones are still in range (using the request broadcast).
* When it receives a reply, it checks if it has already been registered.
  + If a node is unique, it is registered and an IDOK message is sent with the relevant node ID.
  + If a node has already been registered with that ID an IDNOTOK is sent. To differentiate between a new node registering and the same node’s message being bounced around the network, the content field will hold another random number which will be different from both nodes. The IDNOTOK message’s content field will hold the random number received by the rejected node.
* When it needs to send a message to a node it uses flooding to propagate the packet through the network.
* There can only be one server in a network and its ID is zero.

## Nodes

* When it receives the general broadcast from the server it replies with an acknowledgement that includes a randomly generated node ID and a randomly generated content field. The server will reply with an IDOK or IDNOTOK. For IDNOTOK with the same random number in the content field the node will resend a new, randomly generated ID and content to be checked again.
  + The randomly generated ID can be any 8-bit number except zero
* Flooding is used to send the packet throughout the network:
  + When it receives a packet, it checks the target ID against its own node ID.
  + If there is a match the packet is processed and an acknowledge reply is sent with a destination of zero.
  + If the target ID does not match it next checks the time-to-live field. If this is not zero it decrements it by one and rebroadcasts the packet. If it is zero it does nothing.
* Server request broadcasts and acknowledge replies also use this flooding technique.

## Packet Structure

* Message Types (value):
  + Ack (0) – Respond to req message with a node ID. This ID is either a requested ID or an ID already registered with the server.
  + Req (1) – To discover nodes and ensure current nodes are still available.
  + LED (2) – Toggles LED on target device.
  + IDOK (3) – Sent by server to indicate to a node that the requested ID is valid.
  + IDNOTOK (4) – Sent by server to indicate to a node that the requested ID is not valid and a new one must be generated.
* Packet structure:
  + Message Type – type of message
  + Destination ID – Location to send the message to.
  + Time-to-live – Number of hops before the message expires.
  + Packet ID – unique packet ID
  + Opcodes – to identify the message as part of this network, can be any two numbers but must be consistent across the network.
  + Timestamp – Time when the packet was sent in seconds from the Unix epoch (1 January 1970 00:00:00). Packets are ignored 2 seconds after this.
  + Payload structure:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| [0] Message type | [1] Destination ID | [2] TTL | [3] Packet ID | [4] Message | [5] Opcode1 | [6] Opcode2 |

|  |  |  |
| --- | --- | --- |
| Type | Destination ID | Message |
| Req | - | - |
| Ack | 0 | For requesting ID:  Random number to differentiate from same ID requests from other nodes |
| LED | Node ID | - |
| IDOK/IDNOTOK | Node ID/ requested node ID | Content field of received ack |

* Field values for the different message types (TTL and opcodes are constant and packet ID is the counter from the node/server):