# Communication protocol

This section will explain how to get nodes communicate with each other.

As stated in the system overview, when a node is too far away from the central node to get the signal from the central node, we need an “intermediate node” called repeater. This repeater will be responsible for capturing data from the central node and deliver this data to the far-away node or vice versa. To be general in all cases, we will define a data frame to send back and forth among nodes.

When a node wants to send information to other nodes, it has to send the information compressed in the following frame:

|  |  |  |  |
| --- | --- | --- | --- |
| **Route** | **Action** | **Data** | **End of package** |

1. Route

The **Route** of the frame will define the path of the transmission from the sender to the target. It has the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte1** | **Byte2** | **…** | **ByteN** |

**Byte0:** indicates how many nodes there are in the path (exclusive the sender).

**Byte2 – ByteN:**

Indicates node IDs the information must go through from the sender to the destination node.

**Example:**

If we want to send data from the central node (ID 1) to a far-away node (ID 3), we will put one repeater (ID 2) between the node 1 and node 3. So, data from node 1 first goes to node 2, and then node ID 2 sends this data to node ID 3. We have:

* Byte1 = 0x02
* Byte2 = 0x02
* Byte3 = 0x03

So, the content of **Route** part is:

|  |  |  |
| --- | --- | --- |
| **Byte1** | **Byte2** | **Byte2** |
| 0x02 | 0x02 | 0x03 |

1. Action

**Action** part contains one byte with one of the following characters:

* ‘M’: the sender sends the map to the destination node so that the destination node can know the way to send data back to the sender. This is used only when central node sends data to other nodes.
* ‘R’: the sender requests data from the destination node.
* ‘D’: the sender sends data to the destination node.

### Data:

**Data** depends on the action the sender has sent in the previous byte.

1. Action is ‘M’: the first byte of **Data** will be the number of nodes in the path from the destination back to the central node. The next bytes will be IDs of these nodes.
2. Action is ‘R’: **Data** will contain one byte standing for names of sensors that sender is requesting. This byte will be one of the following characters:

* ‘H’: humidity
* ‘T’: temperature
* ‘L’: light
* ‘O’: occupancy
* ‘W’: water level
* ‘Z’: all sensors

1. **Action is ‘D’**: **Data** contains bytes which are information from sensors. It has the following format:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte1 | Byte2 | Byte3 | … | Bytem | Bytem+1 | Bytem+2 | … | Bytek | … |
| # of sensors | Type of 1st sensors | # of data bytes | | | Type of 2st sensors | Number of data bytes | | |  |

1. End of package:

**End of package** is the “$” character. It indicates the end of the package sent by the sender.

When a node receives information from another node, it will first decrease content of Byte1 of **Route** by one and then check whether this content of this byte is zero or not. If this is a zero, this node is the final destination. Therefore, it will have appropriate respond to the sender based on the **Action** and **Data** in the frame. If this is not a zero, clearly this node is a repeater. Thus, it first deletes Byte2 of Route which is its own ID, shifts all other bytes on the right of the deleted byte to the left by one byte, then continue to send modified frame to the next node in the path.

In summary, with this communication protocol, we are able to use just one program but applicable for all nodes in the wireless network. It does not matter the nodes are near the central nodes (no repeater required) or very far away from the central node (many repeater required). Regularity and modularity are the key advantages of this protocol.