

# NEGRAV V1.0 API Documentation

## 1 Summary

The NEGRAV protocol is intended to connect several nodes with sensing and moving capabilities inside a wifi Ad-hoc implemented network. The NEGRAV protocol is implemented over TCP. It doesn't implement a simple Client-Server architecture but instead each node is client and server at the same time to have the ability of start a communication in both ways. For that, it uses 2 sockets in each node, the "server" socket ( listening on port 5310) and the "client" socket (transmitting from port 5320).

This protocol uses JSON (RFC 7159) in the format of the messages.

## 2 IP Address definitions

To identify the type or the state of a node, an address pool is used for each case.

- **10.0.0.100 Base Station(BS) IP:** IP of the Base Station.
- **10.0.0.101 to 10.0.0.110 Backup Base Station(BBS) IP pool:** When a new base station is added to the network and the BS IP is already taken, it choose from this pool of address and turn into an Backup Base Station.
- **10.0.0.200 to 10.0.0.255 Base Station Momentary IP pool:** When a BS is turn on, it choose randomly one of this IP address before checking if the BS IP is taken.
- **10.1.0.1 to 10.1.10.255 Sationary node (SN) IP pool:** The IP pool address of all stationary nodes.
- **10.1.100.1 to 10.1.100.100 Stationary Node Momentary IP pool:** The IP pool address of a node before it is added to the network. The selection is made randomly.
- **10.2.0.1 to 10.2.10.255 Mobile Node (MN) IP pool:** The IP pool address of all mobile nodes.
- **10.2.100.1 to 10.2.100.100 Mobile Node Momentary IP pool:** The IP pool of a node before it is added to the network.The selection is made randomly.

10.0.0.100	Base Station IP
10.0.0.101	Backup Base Station IP pool
10.0.0.110	
10.0.0.200	Base Station Momentary IP Pool
10.0.0.255	
10.1.0.1	Stationary node IP pool
10.1.10.255	

10.1.100.1	Stationary node momentary IP pool
10.1.100.100	
10.2.0.1	Mobile Node IP Pool
10.2.10.255	
10.2.100.1	Mobile Node momentary IP pool
10.2.100.100	

## 3 Process Description

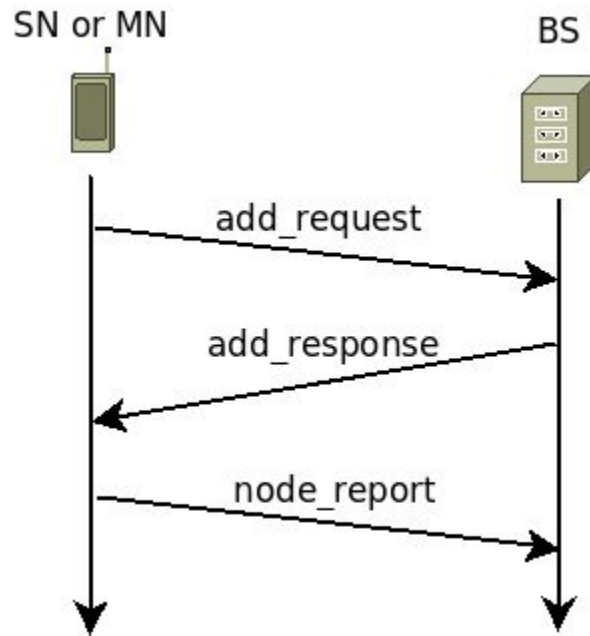
### 3.1 Set-up Process

To init a new network the first thing is to define the Network Identifier (NID). This is a 4 bit number (from 0 to 15) that is set manually in each node of the network (using a dipswitch or in the node software). This NID is used in the SSID of the network. The SSID format is NEGRAV-<NID in hex>. For example if the NID is 10, then the SSID would be NEGRAV-A, or if the NID is 15, then it would be NEGRAV-F.

After that a new Base Station must be set in the network. The use of Backup Base Station are optional but it is recommended to have at least 1. After that, new mobile nodes or stationary nodes can be added to the network.

### 3.2 Add Process

To add a new node to the network, the Mobile Node or Stationary Node should have the same NID as the Base station. A button (manual or virtual) should be pressed in the node to start the add process. After that the user only have to wait for the status LED to know if the process succeeded or failed.



### 3.2.1 add\_request

Message used for requesting the BS to assign an IP.

Format:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "add_request",
  "source_ip": "<SN or MN momentary IP>"
}
```

Example:

```
{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "add_request",
  "source_ip": "10.1.100.15"
}
```

### 3.2.2 add\_response

Message sent by the BS to assign an IP to the new node.

Format:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "add_response",
  "assign_ip": "<SN or MN new IP>"
}
```

Example:

```
{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "add_response",
  "assign_ip": "10.1.100.15"
}
```

```

    "assign_ip": "10.1.0.3"
}

```

### 3.2.3 node\_report

message used to report all the node capabilities to the BS.

Format:

```

{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "node_report",
  "node_ip": "<node_ip>",
  "type": "<SN or MN>",
  "GPS": [
    "<latitude>",
    "<longitude>",
    "<altitude>"
  ],
  "sensor": [
    {
      "name": "<S1_type>",
      "units": [
        "<unit_1>",
        "...",
        "<unit_n>"
      ],
      "resolution": "<sensor_resolution>",
      "range": [
        "<min_range>",
        "<max_range>"
      ]
    },
    {
      "name": "<S2_type>",
      "units": [
        "<unit_1>",
        "...",
        "<unit_n>"
      ],
      "resolution": "<sensor_resolution>",
      "range": [
        "<min_range>",
        "<max_range>"
      ]
    }
  ]
}

```

Example:

```

{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "node_report",
  "node_ip": "10.2.0.50",
  "type": "MN",
  "GPS": [
    "3.54",
    ".653",
    "210"
  ],
  "sensor": [

```

```

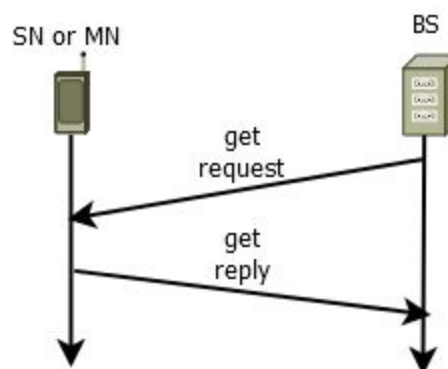
{
  "name": "temp",
  "units": [
    "°C",
    "°F",
    "°K"
  ],
  "resolution": "0.5°C",
  "range": [
    "-50°C",
    "70°C"
  ]
},
{
  "name": "accelerometer",
  "units": [
    "m/s"
  ],
  "resolution": "0.3m/seg2",
  "range": [
    "-5m/seg2",
    "5m/seg"
  ]
}
]
}

```

## 3.3 Get Process

### 3.3.1 Get Request

This message is used by the BS to request MN or SN information. The BS can request for one specific variable data, some variable data or all variables data, the request message contains the kind of request (all or array), and the list of requested variables. The response of this message contains the kind of reply and the list of the variable data requested.



Request format:

```

{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "get",
  "get_type": "<all or array>",

```

```

        "sensor": [
            "<S1_name>",
            "...",
            "<Sn_name>"
        ]
    }

```

**Respond format:**

```

{
    "protocol": "NEGRAV",
    "version": "<current version>",
    "cmd": "get", "get_type": "<all or array>",
    "sensor": ["<S1_value>", "...", "<Sn_value>"]
}

```

**Request example:**

```

{
    "protocol": "NEGRAV",
    "version": "<current version>",
    "cmd": "get",
    "get_type": "<all or array>",
    "sensor": [
        "temp",
        "...",
        "accelerometer"
    ]
}

```

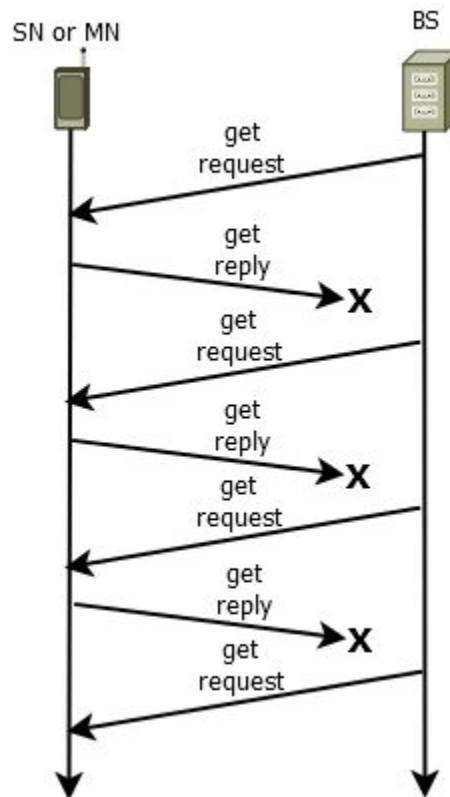
**Respond format:**

```

{
    "protocol": "NEGRAV",
    "version": "<current version>",
    "cmd": "get", "get_type": "<all or array>",
    "sensor": ["45", "...", 3]
}

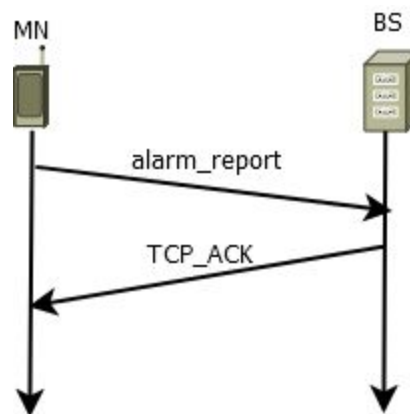
```

When the BS get request, fails it retries 2 more times.



### 3.4 Alarm Process

This process happen when a sensor from a node is out of the configured normal range.



#### 3.4.1 alarm\_report

This message is sent from a node to the BS to report an anomaly in one of the sensors.

Format:

```

{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "alarm_report",
  "node_ip": "<node_ip>",
  "sensor": "<sensor type>",

```

```

    "value": "<sensor_value>"
  }

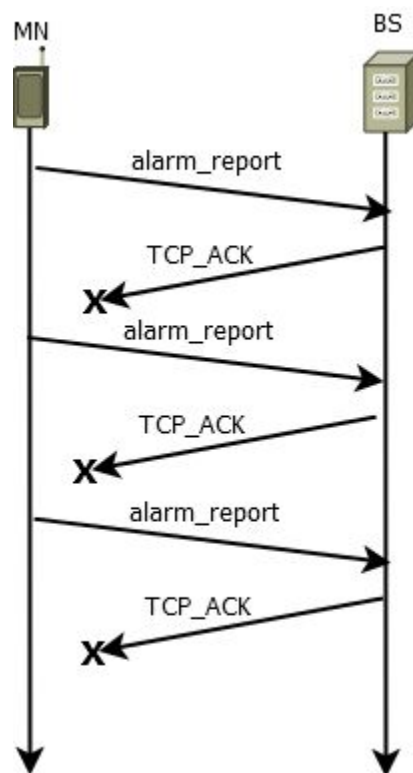
```

Example:

```

{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "alarm_report",
  "node_ip": "10.1.0.85",
  "sensor": "battery",
  "value": "5%"
}

```



## 3.5 Configuration Process

This process is used to remotely configure things in the nodes. Clock, alarms, sampling periods are some of the things that can be changed.

### 3.5.1 Config\_request

Format:

```

{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "node_configure",
  "assign_ip": "<node_ip>",

```



```

    "set_node_time": "<node_clock>",
    "sensor": [
        {
            "S1_name": "<S1_type>",
            "S1_period": "<sample_period>",
            "S1_alarms": [
                "<HI_alarm>",
                "<LO_alarm>"
            ]
        },
        {
            "S2_name": "<S2_type>",
            "S2_period": "<sample_period>",
            "S2_alarms": [
                "<HI_alarm>",
                "<LO_alarm>"
            ]
        }
    ]
}

```

Example:

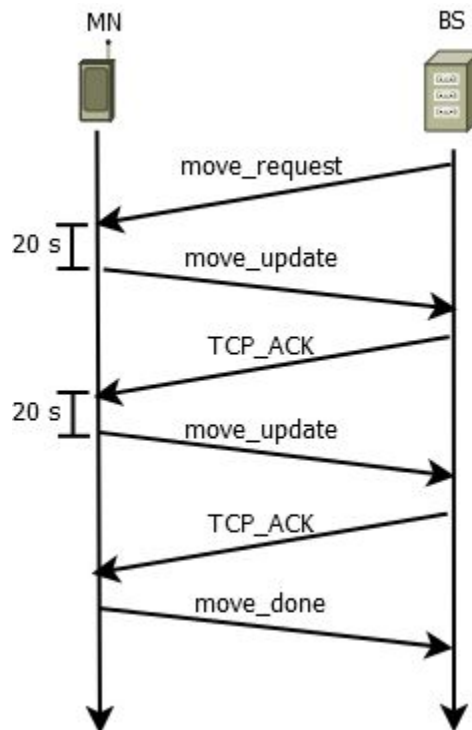
```

{
    "protocol": "NEGRAV",
    "version": "<current version>",
    "cmd": "node_configure",
    "assign_ip": "<node_ip>",
    "set_node_time": "<node_clock>",
    "sensor": [
        {
            "S1_name": "temp",
            "S1_period": "10",
            "S1_alarms": [
                "100",
                "0"
            ]
        },
        {
            "S2_name": "battery",
            "S2_period": "5",
            "S2_alarms": [
                "5",
                "1"
            ]
        }
    ]
}

```

### 3.6 Move Process

The mobile stations are made to move in the field. The action is starter by the base station asking the MN to move to a certain location. The MN start to send notifications each 20 seconds to check if he is still inside the network. In case the communication failed, he return to the last place it was able to communicate and notify the BS that he was not able to reach its destiny. In case the BS want the MN to stop, it should send a move\_request to its current location.



### 3.6.1 move\_request

This is the message sent by the BS to the MN to start the moving process. It sends the target location and optionally sends a road map, a series of point that it should go to arrive the target.

Format:

```

{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "move_request",
  "target_location": [
    "<latitude>",
    "<longitude>"
  ],
  "road_map": [
    [
      "<lat_1>",
      "<long_1>"
    ],
    [
      "<lat_2>",
      "<long_2>"
    ],
    [
      "..."
    ],
    [
      "<lat_n>",
      "<long_n>"
    ]
  ]
}
  
```

Example:

```

{
  
```

```

    "protocol": "NEGRAV",
    "version": "v1.0",
    "cmd": "move_request",
    "target_location": [
        "4.13124",
        "4.3243"
    ],
    "road_map": [
        [
            "1.234",
            "1.12312"
        ],
        [
            "2.13214",
            "2.31241"
        ],
        [
            "3.125",
            "3.51242"
        ]
    ]
}

```

### 3.6.2 move\_update

This message is used to send updates in the movement. Optionally it sends the current point in the road map it is heading.

Format:

```

{
    "protocol": "NEGRAV",
    "version": "<current version>",
    "cmd": "move_update",
    "target_location": [
        "latitude",
        "longitude"
    ],
    "move_delta": "<move_delta>",
    "battery": "<battery_percentage>",
    "current_target": [
        "<lat>",
        "<long>"
    ]
}

```

Example:

```

{
    "protocol": "NEGRAV",
    "version": "1.0",
    "cmd": "move_update",
    "target_location": [
        "4.234",
        "4.14235"
    ],
    "move_delta": "2.3m",
    "battery": "15%",
    "current_target": [
        "2.2134",
        "2.12414"
    ]
}

```

```
    ]
}
```

### 3.6.3 move\_done

This message is sent by the MN when stop moving either because it arrived its destination or because it encounter a problem.

Those info messages can be:

- destination\_reached: when it arrives to its target location.
- out\_of\_range: When a problem with range occurred.
- no\_movement: When after a while it is trying to move but the move\_delta is zero.

Format:

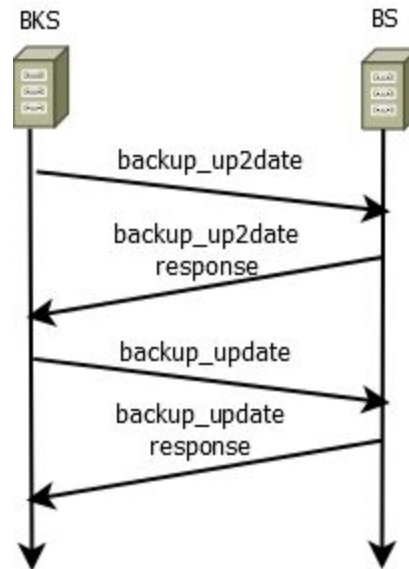
```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "move_done",
  "current_location": [
    "<latitude>",
    "<longitude>"
  ],
  "reason": "<destination_reached or out_of_range or no_movement>",
  "battery": "<battery_percentage>"
}
```

Example:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "move_done",
  "current_location": [
    "2.12124",
    "2.151412"
  ],
  "reason": "out_of_range",
  "battery": "15%"
}
```

## 3.7 Backup Process

The backups are periodically asking the BS the version of the data it posses. It check the version number with its internal version and if it differ it ask the BS for the update This information includes the other BBS IP addresses. In case the BS does not respond to the version check, the BBS with the lower IP number will take the IP of the BS and become the next BS.



### 3.7.1 backup\_up2date

This message is send to the BS to request the version ID. The response uses the same message including the version field.

Request Format:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "backup_up2date",
  "bkup_ip": "<Backup_IP>"
}
```

Response Format:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "backup_up2date",
  "bkup-version": "<version_ID>"
}
```

Request Example:

```
{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "backup_up2date",
  "bkup_ip": "10.0.0.101"
}
```

Response Example:

```
{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "backup_up2date",
  "bkup-version": "123512ASF332523G"
}
```

### 3.7.2 backup\_update

This message is used by the BSS to request an information update. The response of this message contains the new version number, the list of all backups and the list of all the nodes in the network, with all the information.

#### Request format

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "backup_update",
  "bkup-ip": "<backup-ip>"
}
```

#### Response format:

```
{
  "protocol": "NEGRAV",
  "version": "<current version>",
  "cmd": "backup_update",
  "bkup-version": "<backup-version>",
  "bkup-list": [
    "<bkup_1_ip>",
    "<bkup_2_ip>",
    "...",
    "bkup_2_ip"
  ],
  "nodes": [
    {
      "node_ip": "<node_ip>",
      "type": "<SN or MN>",
      "GPS": [
        "<latitude>",
        "<longitude>",
        "<altitude>"
      ],
      "sensor": [
        {
          "name": "<S1_type>",
          "units": [
            "<unit_1>",
            "...",
            "<unit_n>"
          ],
          "resolution": "<sensor_resolution>",
          "range": [
            "<min_range>",
            "<max_range>"
          ]
        },
        {
          "name": "<S2_type>",
          "units": [
            "<unit_1>",
            "...",
            "<unit_n>"
          ],
          "resolution": "<sensor_resolution>",
          "range": [
            "<min_range>",
            "<max_range>"
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

#### Request example:

```

{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "backup_update",
  "bkup-ip": "10.0.0.101"
}

```

#### Respond example

```

{
  "protocol": "NEGRAV",
  "version": "v1.0",
  "cmd": "backup_update",
  "bkup-version": "123GSDF324123",
  "bkup-list": [
    "10.0.0.101",
    "10.0.0.105",
    "10.0.0.109"
  ],
  "nodes": [
    {
      "node_ip": "10.1.0.15",
      "type": "SN",
      "GPS": [
        "3.4342",
        "1.12321",
        "345"
      ],
      "sensor": [
        {
          "name": "temp",
          "units": [
            "°C",
            "°F"
          ],
          "resolution": "1°C",
          "range": [
            "-50°C",
            "75°C"
          ]
        }
      ]
    }
  ]
}

```