Smart Communication Environment for Extraterrestrial habitat

Eenesh Chavan*
eenesh@uni-bremen.de
University of Bremen
Bremen, Germany

Dhruv Patel*
dhruv@uni-bremen.de
University of Bremen
Bremen, Germany

Anna Förster
anna.foerster@uni-bremen.de
University of Bremen
Bremen, Germany

***Abstract*—The future extraterrestrial Human activities, e.g. on Mars and Moon will pose several challenges such as transportation, self-sustainable habitat, Earth-to-habitat communication, the safety of the astronauts and probes outside the habitat, as well as the communication between the astronauts and the habitat. We propose a solution for the communication between the astronauts/probes and the habitat using LoRa and IEEE802.11s using OLSR, a proactive Ad-Hoc network routing protocol to implement on the Wi-Fi network between the gateways and the habitat to ensure frequent and reliable communication between the end-devices and the habitat. The initial results of our proposed solution show a viable option for future Human colonies on Mars.**

***Keywords*—Proactive Routing Protocol, OLSR, Pycom LoPy4, LoRa, Raspberry Pi 2, IEEE802.11s, Dragino v1.4, Wi-Fi, habitat, gateway, end-device.**

I. Introduction

Planning to set up long-term and sustainable Human colonies on Mars has several problems which need to be solved. One of them is to ensure the establishment of reliable communication between the habitat and the Astronauts or Probes (end-devices). Initial Human colonies would require a quick setup method to establish reliable communication between the astronauts/rovers and the habitat.

Studies such as Mars communication constellation [1] analyse the possibility of using small spacecraft constellations to enable communication infrastructure for future human settlements. However, using satellites to relay data to the end-devices from the habitat and vice-versa requires devices with high transmission power and is economically expensive. The inability to perform urgent maintenance of the satellites could also lead to communication blackout zones in the network. A well-established Human colony with the necessary infrastructure can use the rigid and stable mobile network infrastructure. However, an initial exploration party will require a quick solution to establish reliable communication, which does not require well-established heavy infrastructure. Thus, using an expensive satellite constellation or rigid mobile network infrastructure is not the ideal solution for the required scenario. Hence, in this paper, we propose an inexpensive solution which enables the monitoring and tracking of mobile devices by installing gateways around the habitat. Fig. 1 shows an example scenario of the proposed network of gateways around the habitat and end-devices. The gateways used will relay the data from the mobile devices to the habitat and vice-versa.

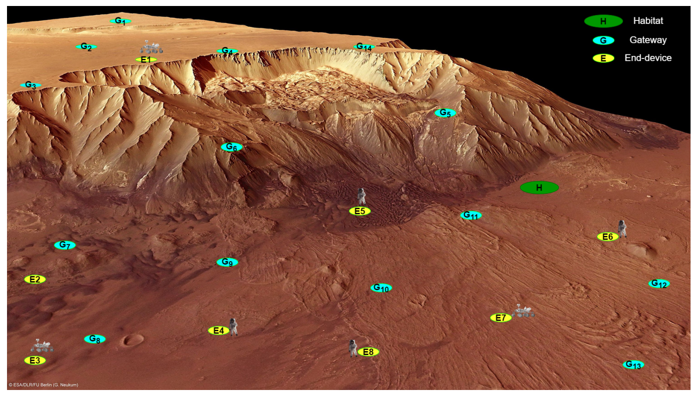


Fig. 1. Scenario depicting the network of gateways around the habitat and end-devices

The Martian environment has proved to be harsh on the previous exploration missions [2]. Large dust storms and high doses of cosmic radiation are the major challenges which can disrupt the communication network. Hence, we propose to use Ad-hoc Network Routing Protocol [3] to avoid a single point of failure and maintain reliable communication. The end-devices on an exploration mission have to be equipped with low power and long-range communication devices to ensure long-lasting and reliable communication with a limited power supply. Therefore, we propose to use Long Range (LoRa) radio communication technique on the end-devices to be more energy efficient and have a higher range.

In this paper, we present the initial results of our implementation of an Ad-hoc network routing protocol on a Wi-Fi network consisting of gateways and habitat while connecting the end-devices to the gateways using LoRa. We built an experimental setup with habitat, gateways and end-devices. The habitat and gateways are *Raspberry Pi 2*[1]. Both of them use the Edimax EW-7811Un Wi-Fi adapters, which are capable of mesh routing. The habitat is

*Both authors contributed equally to this research.

[1] https://www.raspberrypi.com/products/raspberry-pi-2-model-b/

programmed to transmit and receive data from the gateways and store it locally. The gateways also use the *Dragino v1.4 shields*[2] for the LoRa interface and are programmed to relay the data received from the habitat to the end-device and from the end-device to the habitat, and store it locally. The end-devices are *PyCom LoPy4*[3] devices programmed to send beacons & data, receive data from the gateway and store it locally. To evaluate our solution, we identified a test scenario and undertook a performance evaluation.

The remainder of the paper is organized in the following manner: in Section II we describe the scenario, the role of the Ad-Hoc network routing protocols and briefly explain the implementation of Optimized Link State Routing Protocol (OLSR) [4]. In Section III, we describe the system we built, and it's functioning. Afterwards, we describe the scenario used to evaluate the performance and the analysis in Section IV. Section V is a concluding summary of our work and a brief overview of our intended future work.

II. Scenario and Ad-Hoc Network Routing Protocols

A failure of the central access point in a communication network leads to a total shutdown of the network. On Mars, this type of failure could hinder a critical exploration mission and could be lethal to human lives. Hence, a Mobile Ad-hoc Network (MANET) [3] can be implemented to avoid dependency on a central access point in the communication network. By implementing an Ad-hoc network routing protocol, we propose an experimental scenario for duplex communication between the habitat and the end-devices. Fig. 2 shows the scenario where multiple end-devices are connected to the gateways. They can send and receive data from the gateways. The gateways can send and receive data from the habitat using the ad-hoc network of gateways and habitat. To understand this scenario, let us consider an example in which end-device ID734 wants to send some data to the habitat. To do so, ID734 sends data to the connected gateway 10.10.10.6 using LoRa and then the gateway 10.10.10.6 forwards the packet to the habitat using the ad-hoc network of the gateways. Similarly, the habitat sends the data to the end-device ID734 by first sending it to the gateway 10.10.10.6 using the routing protocol and then the data is forwarded by 10.10.10.6 to ID734. In this ad-hoc network of gateways and the habitat, each node (gateway/habitat) participates in routing by forwarding the data to other nodes. The nodes forwarding the data are determined on the basis of routing algorithms.

The ad-hoc network routing algorithms/protocols are broadly classified into two categories:-

1) Reactive routing protocol: A reactive routing protocol is also known as an On-demand routing protocol [5]. In this type of routing, a route is discovered only when required. The route is mainly discovered by flooding the entire network with a Route Request Packet (RREP). The routing consists of two main phases: (i)route discovery, and (ii) route maintenance. DSR [6], AODV [7] and LAR [8] are the reactive routing protocols considered in this paper and briefly explained as follows:-

- **Distance State Routing (DSR [6]):** DSR computes the routes using source routing and allowing each node to dynamically discover a source route to any destination across multiple hops. The data packet sent carries the entire ordered list of the intermediate nodes in its header. It allows packet routing loop-free and mitigates the need for up-to-date routing information in the intermediate nodes through which the packet is forwarded.
- **Ad-hoc On-demand Distance Vector (AODV [7]) routing:** AODV also uses broadcast route discovery method like DSR. However, it dynamically establishes route table entries at intermediate nodes instead of source routing. The most recent routing information between the nodes is maintained using monotonically increasing sequence numbers to mitigate the counting to infinity looping problem.
- **Location Aided Routing (LAR [8]):** LAR uses the location information to improve the performance of the route discovery. The entire network is flooded with *route request* packets if the source does not have information about the destination. However, on availability of the previous location of a mobile destination node limits the flooding area with *route request* packets. In this case, the source expects the destination to be present in an expected zone. This zone is created based on the previous location and the velocity of the destination node. A request zone

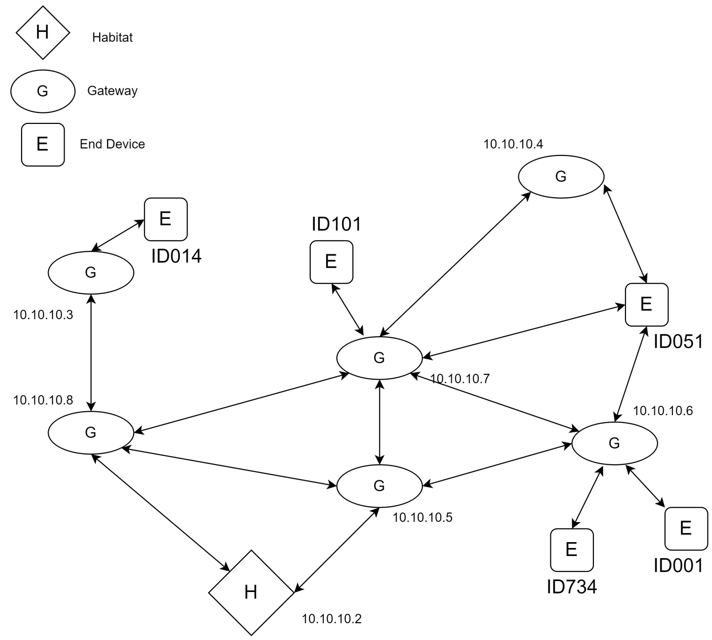


Fig. 2. Scenario

[2]https://wiki.dragino.com/index.php/Lora_Shield
[3]https://pycom.io/

is then defined by the source for the route request, with the expected zone and source being part of this zone. The *route request* is forwarded only by the nodes belonging to this zone. Hence, limiting the area of route discovery.

2) Proactive routing protocol: The second class of Ad-hoc network routing protocols are proactive routing protocols, also known as table-driven routing protocols [9]. In proactive routing protocols, each node/gateway maintains its own routing table containing the route information to all the nodes connected in the network. These tables are updated periodically to have information of the latest network topology. Two such proactive routing protocols considered are DSDV [10] and OLSR [4] and briefly explained as follows:-

- **Destination-Sequenced Distance-Vector Routing (DSDV [10]):** Based on the distance-vector algorithm, in DSDV, each node maintains a routing table listing all available destination nodes and the number of hops to each of them. Each node monitors the cost of outgoing links and broadcasts periodically to each of its neighbours to keep the distance estimates up to date. To mitigate the counting to infinity looping problem, each routing table entry is tagged with a sequence number to distinguish stale routes from the new ones.
- **Optimized Link State Routing (OLSR [4]) protocol:** OLSR is based on link state routing. Each node in this case monitors the network topology, with a cost for each link. Each node maintains a routing table listing the available destinations nodes. To minimizes the flooding of the network, each node selects some of its neighbours as multipoint relays, which are allowed to relay data packets into the network and eventually to the destination node.

The reactive and proactive protocols can be broadly distinguished based on the delay, overhead, topological changes, mobility of the nodes and routing method. Table I shows this comparison between Proactive and Reactive routing protocols:-

TABLE I
PROACTIVE VS. REACTIVE ROUTING

Proactive	Reactive
Table driven routing	Uses route discovery method
Less delay	High delay
Better for static and frequent communication	Better for mobile devices
Slow reaction to topological changes	Agile to changes in topology
High overhead	Low Overhead

Using the above comparison we deduce that the reactive routing protocols such as AODV, DSR and LAR do not maintain a routing table but only discover a route on-demand. Therefore, they are agile to the topological changes but have high delays. However, a typical network on Mars will consist of gateways which do not move frequently and the end-devices such as probes, astronauts and rovers which are mobile but slow. Therefore, the network will require the data to be forwarded with low delay and is not prone to frequent topological changes. LAR can minimize the flooding of the network but initial mars settlements will not have a widespread GNSS system to provide accurate locations. Therefore, LAR is not a viable option. Considering the above drawbacks posed by the Reactive routing protocols, for low-density and low-speed scenarios, Proactive routing protocols such as OLSR and DSDV are proven to have better performance [11]. Hence, a brief comparison between OLSR and DSDV is presented.

Destination-Sequenced Distance-Vector Routing (DSDV) is a type of distance-vector algorithm. Each node maintains a routing table listing all the available destinations and the number of hops to the destination node. Each node broadcasts its routing table, hence flooding the entire network. On the other hand, Optimized Link State Routing (OLSR) is a link-state algorithm. Each node maintains a view of the topology with a cost for each link. In OLSR, each node assigns some of its neighbours as Multi-point Relays (MPR) based on the number of second neighbours connected and, unlike DSDV, uses these MPRs for broadcasting the messages rather than flooding the entire network. Table II shows a brief comparison between OLSR and DSDV. It is seen that OLSR has better throughput and low delay compared to DSDV. One of the drawbacks of Distance Vector algorithms is *counting-to-infinity* problem in large-size networks, which is not the case in Link State Algorithms. Because a human colony will require a large network of gateways, OLSR suits us better compared to DSDV for our implementation.

TABLE II
OLSR VS. DSDV

	OLSR	DSDV
Routing Algorithm	Link State	Distance Vector
Flooding of network	No	Yes
Throughput	High	Low
Delay	Low	High
Frequency of updates	Event/time driven	Time driven

A. *OLSR (Optimized Link State Routing) protocol*

OLSR protocol is an optimization of the pure link-state protocol for MANET. Each node selects a set of nodes in its neighbourhood, called the Multipoint Relays (MPRs). Only the MPRs are allowed to retransmit their packets. The MPRs reduce the duplicate retransmissions of the broadcast packets in the same region, therefore minimizing the flooding of broadcast packets in the network. The remaining nodes in the neighbourhood which are not in the MPR set, read and process the packets. They do not retransmit the broadcast packets. Each node selects its MPR set among its one-hop neighbours fulfilling the following condition: every two-hop neighbour of the node must have a bidirectional link with the MPR node. The smaller the multipoint relay set is, the optimal routing protocol is.

1) ***Neighbor sensing and Multipoint relay selection****:* Each node checks the links with its neighbour nodes to validate the bi-directionality of the link. The nodes accomplish this by periodically broadcasting *HELLO* messages which contain the information about its neighbours and their link status. This *HELLO* message contains: (i) the list of addresses of the first neighbours to which a bi-directional link exists, (ii) the list of addresses of the first neighbours which are heard by the node but not validated to have a bi-directional link. These *HELLO* messages allow each node to the knowledge of its neighbourhood up to two hops, therefore selecting its multipoint relays. The selected multipoint relays are then marked as MPR in the link status of the *HELLO* messages. Each node upon receiving the *HELLO* message, constructs the *MPR Selector* table with the nodes who have selected it as an MPR. Each node also maintains a neighbour table to record the information about its one-hop neighbour, a list of two-hop neighbours and the link status of these one-hop neighbours which can be either uni-directional, bi-directional or an MPR. The neighbour table also maintains sequence number value to specify the latest MPR set of the node. The sequence number is incremented to a higher value when a node selects or updates its MPR set. The multipoint relay set is re-calculated when:

- when a bi-directional link is added with a new neighbor or fails in the neighborhood; or
- two-hop neighbor set with bi-directional link changes.

*2) **Topology and Routing Table calculation**:* To build the *intra-forwarding* database for routing packets, each node broadcasts a *Topology control (TC)* message periodically to declare its *MPR Selector* set, which is forwarded as a usual broadcast message in the entire network. The information obtained from the TC messages is then recorded in a topology table maintained by each node in the network. The node records information about the multipoint relays of other nodes in this table and uses this table along with the neighbor table to calculate the routing table.

III. Functioning of the network

At initialization of the network, all the gateways sync their clocks with the habitat's clock using the Network Time Protocol (NTP) [12] over the Wi-Fi network. In this way, the habitat's clock can be considered as the network's clock. After initialization, we classify the functioning of the network into two separate communication channels as follows:-

- Enddevice-to-Habitat (E2H) communication
- Habitat-to-Enddevice (H2E) communication

In E2H communication, the packet is sent from the end-device to the habitat via the gateways, while in the H2E communication, the packet is sent from the habitat to the end-device via the gateways.

*A. **E2H communication***

The E2H communication happens through two steps:

1) end-device to gateway over LoRa.
2) gateway to habitat over the Wi-Fi mesh network with OLSR routing protocol. If a gateway doesn't have a direct connection with the habitat, then the data is sent with Multihop.

*1) **From End-device to Gateway (E2G)**:* There are three types of packets sent by the end-device: (i) **Time-synchronization packet**, (ii) **Beacon packet** and (iii) **Data packet**. The time-synchronization packet is broadcasted only once at the initialization of an end-device. The time-synchronization packet is used to receive a gateway-to-enddevice time packet (G2E time packet) from a gateway to synchronize the clock on the end-device with the network's clock. The time-synchronization function enables the end-device to synchronize it's time to only one gateway, therefore discarding the G2E packets received from other gateways. Fig. 3 shows the LoRa packet sent by the end-device for time-synchronization. The time-synchronization packet includes the epoch time stamp of the unsynchronized end-device.

The time-synchronized packet thus received by the gateway

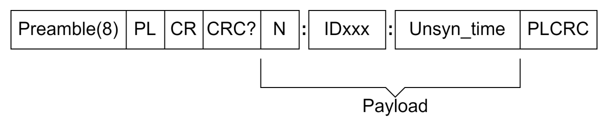


Fig. 3. Packet format for the Time-synchronization from end-device

PL: Payload Length (max 251 octets)
CR: Coding Rate
CRC?: Cyclic Redundancy Check present?
PLCRC: Payload Cyclic Redundancy Check
N: Time-synchronization packet identifier
IDxxx: end-device ID
Unsyn_time: Epoch time on unsynchronized end-device.

is then processed, and the gateway sends a G2E time packet which includes the synchronized time of the gateway with the habitat (Syn_time), time at which the time-synchronization packet was received from the end-device (Recv_time) and the Unsyn_time of the end-device. Fig. 4 shows the time-synchronization packet sent by the gateway to the end-device. The G2E time packet is then received at the end-device, where it is used to calculate the offset (Eq. 1) [13] between the end-device clock and gateway clock using the ***Syn_time***, ***Recv_time***, ***Usyn_time*** and the time at which the G2E time packet is received (***Enddevice_Recv_time***). The offset thus calculated is then added to the unsynchronized time of the end-device to synchronize it with the gateway and the entire network.

$$offset = \frac{(Recv\_time - Usyn\_time) + (Syn\_time - Enddevice\_Recv\_time)}{2} \quad (1)$$

The second type of packets sent by an end-device are beacon packets which are broadcasted periodically at an interval of 30 seconds over LoRa in a packet format as shown in Fig. 5. This beacon packet enables a gateway to learn the end-devices connected to it. The gateway uses the beacon packet to maintain and update a *Geography Table (GT)* shown in Table III. The Geography Table contains a list of all the connected end-device IDs along with the corresponding

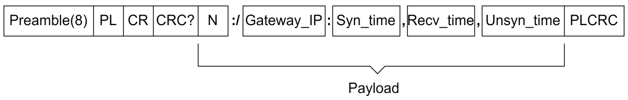


Fig. 4. Packet format of the G2E time packet sent by the gateway to an end-device.

PL: Payload Length (max 251 octets)
CR: Coding Rate
CRC?: Cyclic Redundancy Check present?
PLCRC: Payload Cyclic Redundancy Check
N: Time-synchronization packet identifier
Gateway_IP: IP address of the sender gateway
Syn_time: Synchronized time on gateway
Recv_time: Time at which time-synchronization packet received on gateway
Unsyn_time: Epoch time on unsynchronized end-device.

*Received Signal Strength Indicator (**RSSI**).*

The data packets shown in Fig. 6 are periodic or non-periodic packets which include the data (such as text messages, sensor values, etc.) sent by the end-devices to the gateway and forwarded immediately to the habitat. The data packet is also used to update the Geography Table of the gateway by adding or updating the end-device ID and corresponding RSSI value.

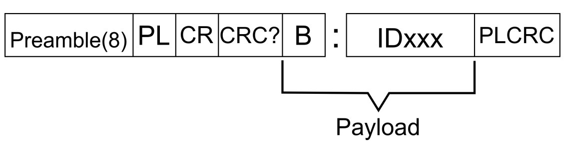


Fig. 5. Packet format of the Beacons from end-devices

PL: Payload Length (max 251 octets)
CR: Coding Rate
CRC?: Cyclic Redundancy Check present?
PLCRC: Payload Cyclic Redundancy Check
B: Beacon packet identifier
IDxxx: end-device ID

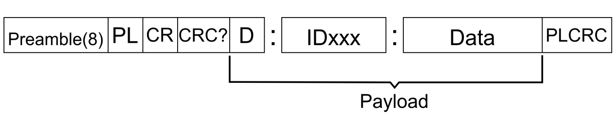


Fig. 6. Packet format of Data packets from end-devices

PL: Payload Length (max 251 octets)
CR: Coding Rate
CRC?: Cyclic Redundancy Check present?
PLCRC: Payload Cyclic Redundancy Check
D: Data packet identifier
IDxxx: end-device ID

The explicit LoRa packet format is used for both the Beacons and the Data packets. Hence, the maximum payload in a LoRa packet is 251 octets.

2) ***From Gateway to Habitat (G2H)**:* The *Geography Table* (see Table III) maintained by the gateway, is then sent to the habitat as a Geography Table message (GT message) at an interval of 30 sec. over the Wi-Fi network with TCP/IP. The interval can be adjusted based on the traffic of the end-devices, to limit the load on the WiFi network. If a Gateway is expected to handle mobile end-devices often (maybe near a transport route) then the interval should be reduced whereas if a Gateway is expected to handle stationary end-devices often (end-devices like soil monitoring sensors) then the interval should be increased. The GT message thus sent by the gateway has the packet format as shown in Fig. 7 and is used by the habitat to update its own Geography Table (Table IV) and learn the end-devices connected to the gateways. This information is later used by the habitat to send data packets to the end-device. When a gateway receives a Data packet shown in Fig. 6 from an end-device over LoRa, it generates a TCP packet shown in Fig. 8 and instantly forwards it to the habitat.Using TCP allows for good failure recovery and good Throughput, and it also doesn't interrupt any existing services. At habitat, we have set a limit of 5 sockets to be queued before further incoming socket connections are rejected. It is done to avoid the overload on the Habitat and to reduce the delay between G2H communication.

TABLE III
GEOGRAPHY TABLE MAINTAINED ON GATEWAY

end-device ID	RSSI (in dBm)
ID014	-90.86
ID001	-96.2
ID051	-69.53

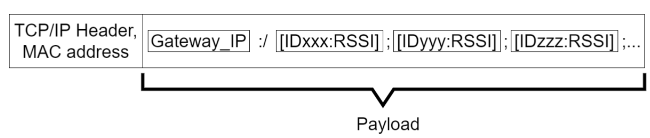


Fig. 7. Packet format of the Geography Table (GT) message

gateway_IP: IP adderess of the sender gateway
IDxxx: end-device ID
RSSI: Received Signal Strength Indicator of the IDxxx

TABLE IV
HABITAT GEOGRAPHY TABLE

gateway IP	end-device ID	RSSI (in dBm)
10.10.10.3	ID014	-90.86
10.10.10.6	ID001	-96.2
10.10.10.6	ID051	-65.33
10.10.10.6	ID734	-83.27
10.10.10.7	ID101	-43.72
10.10.10.7	ID051	-98.34
10.10.10.4	ID051	-59.53

*B. **H2E communication***

In H2E communication, the data is sent by the habitat to an end-device. To send data from the habitat to an end-device, the habitat should know with which gateways the end-device

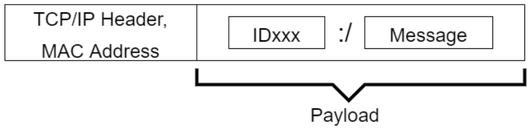


Fig. 8. Packet format of the forwarded message
IDxxx: end-device ID

has a LoRa connection. The Geography Tables thus sent by the gateways to the habitat are then stored on the habitat's own Geography Table (Table IV). The habitat Geography Table consists of the available gateways, end-device ID connected to the respective gateways and their corresponding RSSI values sensed on the gateways.

The H2E communication takes place in two steps:

1) habitat to gateway over the Wi-Fi mesh network with OLSR routing protocol. If the habitat doesn't have direct connection with a gateway, then the data is sent with Multihop.
2) gateway to end-device over LoRa.

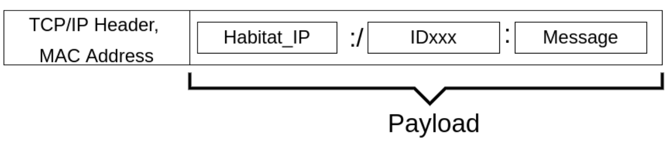


Fig. 9. Packet format of the message sent from habitat to gateway
habitat_IP: IP address of the habitat
IDxxx: Destination end-device ID

*1) **From Habitat to Gateway (H2G):*** An end-device could have a LoRa connection with multiple gateways. Therefore, the habitat can send data packets to multiple gateways to reach an end-device, which can cause unnecessary flooding of the network for H2E communication. To prevent this, the habitat's Geography Table is used. Let us consider the end-device "ID051", which is connected to multiple gateways as shown in the Table IV. The habitat first tries to send the data to the gateway "10.10.10.4" that maintains the best RSSI with the destination end-device "ID051". The data is sent with TCP over the Wi-Fi network in the packet format as shown in Fig. 9. If the gateway "10.10.10.4" is unavailable and not updated on the habitat Geography Table, then the habitat tries to send the data to the gateway "10.10.10.6", which has the second-best RSSI with the destination end-device "ID051". If the gateway 10.10.10.6 is also not reachable, then the habitat tries to send the data using the next gateway with the best RSSI to the end-device and so on until the data is sent to a reachable gateway. In this way, the data is sent reliably without flooding the network.

*2) **From Gateway to End-device (G2E):*** The Wi-Fi network used in the experiment has a higher data rate compared to

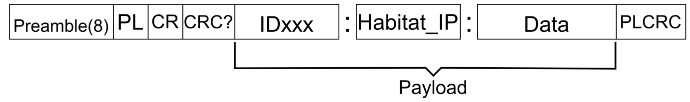


Fig. 10. Packet format of Data packets from gateway to end-devices
PL: Payload Length (max 251 octets)
CR: Coding Rate
CRC?: Cyclic Redundancy Check present?
PLCRC: Payload Cyclic Redundancy Check
IDxxx: Desitnation end-device ID
habitat_IP: IP address of the habitat

the LoRa network. To bridge this gap, the data from the habitat over Wi-Fi is first saved on the gateway and then forwarded to the end-device over LoRa. The data is sent from the gateway to the destination end-device in the packet format shown in Fig. 10. An end-device only receives the data if the end-device's ID is stated in the packet.

IV. Evaluation

The system developed is evaluated using an experimental setup. This section describes the experimental setup, the scenario used and the analysis of the results.

A. Experimental Setup and the Evaluation scenario

We are using commercially available *Raspberry Pi 2* as the habitat and the gateways with *Edimax EW-7811Un* WiFi adapters configured to a power level of 0dBm to limit the range. The gateways also use the *Dragino v1.4* LoRa shield. All the gateways have the same Spreading Factor (SF = 7), coding rate (CR = 4/5) and power level (2dBm) for the LoRa shield. According to the EU region specifications, the gateways are set to an operating frequency of 868 MHz and Class C LoRa devices, allowing the radio to listen to the incoming packets at all times. All the gateways are operated on power banks. The end-devices can be both short- and long-term users of the network. Some of the applications identified for the end-devices are shown in Table V. The test was performed with one habitat, 6 gateways and 4 end-devices. Fig. 11.

TABLE V
TEST CASES EVALUATED IN THE EXPERIMENTAL SCENARIOS

end-device type	Description
Short-term	An astronaut/rover on a short-term exploration mission with asynchronous short text messaging service.
Long-term	Beacons from a hibernating probe
Long-term	Temperature probe to measure atmospheric temperature

The gateways are placed such that at least one of them have a Line of Sight (LOS) and direct connection with the habitat. The experiment was conducted into two parts: (i) E2H and (ii) H2E communication. For E2H communication, 30 data packets were sent from each of the end-devices to the habitat. Additionally, the beaconing interval for end-devices is set to

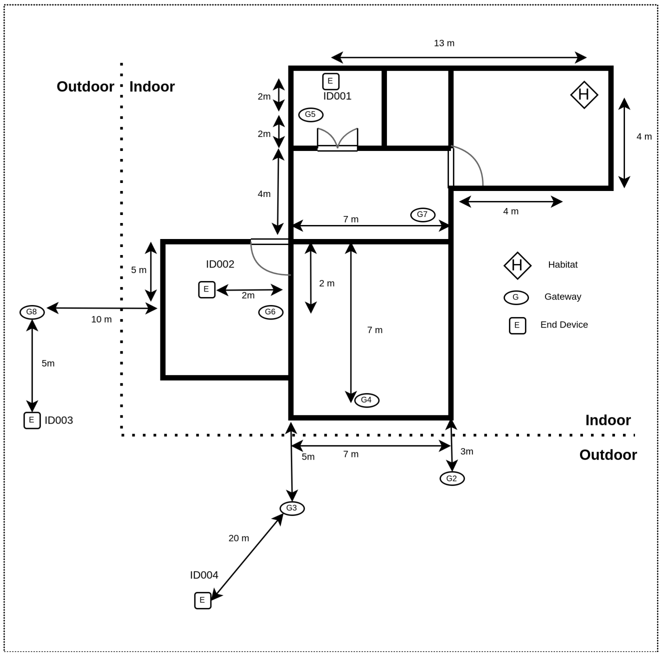
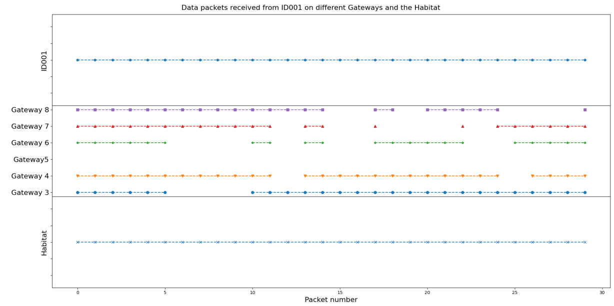


Fig. 11. Experimental scenario (caveat: approximate distances, not drawn to scale)

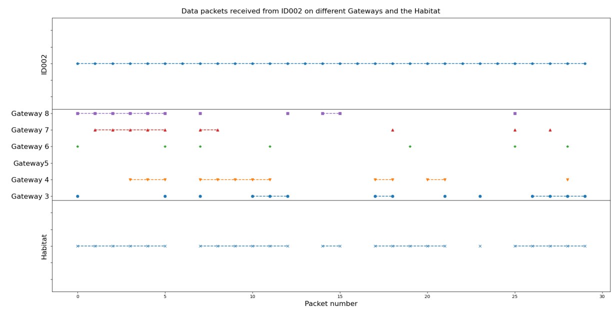
30sec, and the interval for periodic GT messages is set to 30sec for the gateways. For H2E communication, 30 data packets were sent from the habitat to each of the four end-devices.

B. Results and Analysis

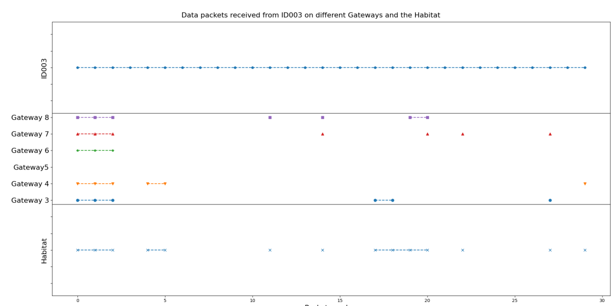
To evaluate the overall performance, (i) **Delivery Ratio** and (ii) **Average Delay** are used as the performance metrics. We first analyze the performance metrics for the E2H communication. Fig. 12 represents the data packets sent by the end-devices which are received by multiple gateways and subsequently received by the habitat. The gateway 5 was offline in E2H communication and hence does not receive any packet from the end-devices (see Fig. 12). For ID001 (Fig. 12(a)), we observe that each gateway does not receive all the 30 packets from the end-device, but each packet is received by at least one or more gateways and sent to the habitat, therefore all the packets are received by the habitat giving us a delivery ratio of 1 for E2H communication as seen in the plot for delivery ratio in Fig. 13(a) & 13(c). For ID002 and ID004 the delivery ratio is not 1 for E2H communication (see Fig. 13(c)) which is also observed in Fig. 12. The overall delivery ratio for ID003 is 0.47 which represents the irregular connection of the end-device with the gateways. In G2H communication if the habitat's queue is full then it will reject any incoming socket connections and it is evident from the Fig. 13(b) that some of the packets were lost due to the incoming socket connection rejection at the habitat and therefore the delivery ratio for G2H communication is not always 1 for all the gateways.



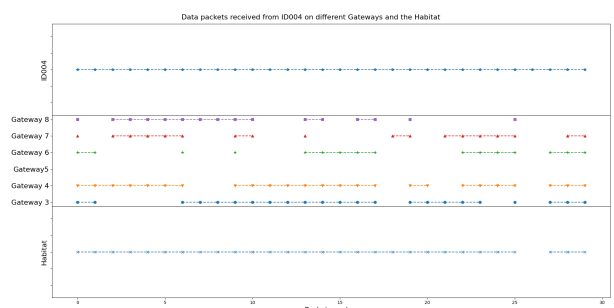
(a) ID001



(b) ID002



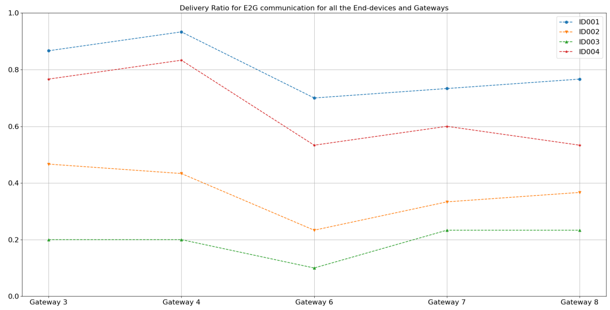
(c) ID003



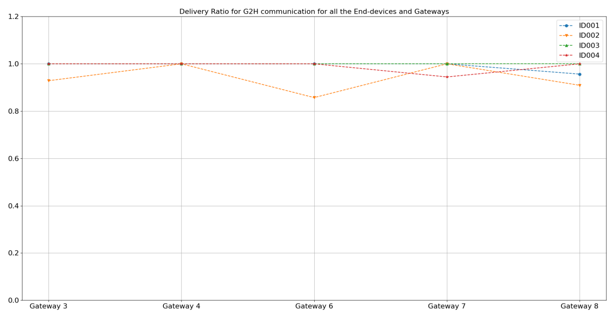
(d) ID004

Fig. 12. Plots representing the Data packets sent by the end-devices to the habitat via multiple gateways.

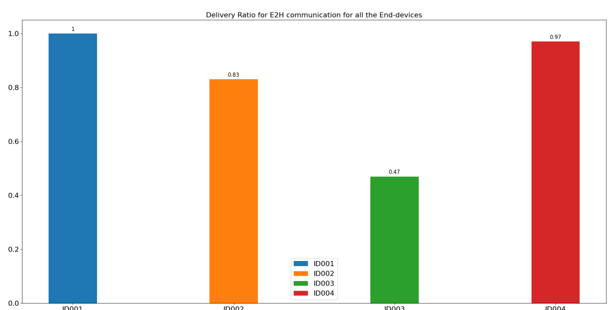
Fig. 14 shows the plots of average delay for E2G, G2H and E2H communication for all the end-devices. The E2G average delay plot here considers the delay from the point of transmission of the data packets from the end-device to the time stamp at which the data packet is received and stored at the gateway. Similarly, G2H average delay considers the delay



(a) Delivery ratio for E2G communication

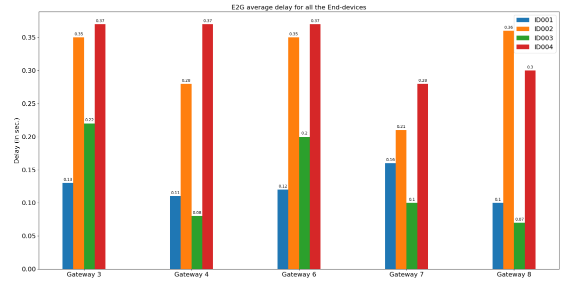


(b) Delivery ratio for G2H communication

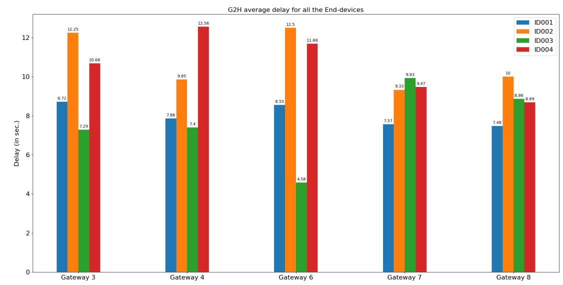


(c) Delivery ratio for E2H communication

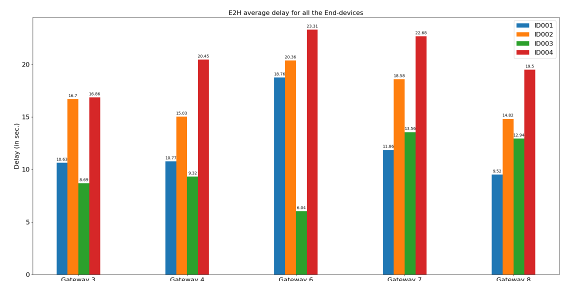
Fig. 13. Delivery ratio for E2G, G2H and E2H communication for all the end-devices and gateways.



(a) Average delay in the E2G communication



(b) Average delay in the G2H communication



(c) Average delay in the E2H communication

Fig. 14. Average delay (a) for receiving data packets at the gateway from the end-devices, (b) for receiving data packets from the gateways at the habitat and (c) overall average delay for receiving the data packets from the end-device at the habitat.

from the point of transmission of the data packets from the gateway to the time stamp at which the packet is received and stored at the habitat. In Fig. 14(a), it is observed that all end-devices have an average delay in the range of 0.07 - 0.37 sec. The G2H average delay is relatively high compared to the E2G delay and in the range of 4.58 - 12.56 sec. (see Fig. 14(b). It is due to the fact that the packet sent from the gateway first waits in the queue at the habitat server. The E2H delay should be more than the sum of E2G and G2H delay because it also includes the processing time on the gateways to forward the received data packet and it is clearly observed in Fig. 14(c) that it is more than the sum of the E2G average delay (see Fig. 14(a)) and G2H average delay (see Fig. 14(b)).

For H2E communication, we observe that the habitat is sending data to specific gateways in Fig. 15 which are then forwarding data to specific end-devices. The ID001, ID002, ID003 and ID004 receives data packets from Gateway 5, 6, 8 and 3, respectively. Fig. 16 shows the delivery ratio for H2E communication, representing a delivery ratio of 1 for all the end-devices. Hence, all the packets sent by the habitat are received by the end-devices.

Fig. 17 shows the plots for average delay for H2G, G2E

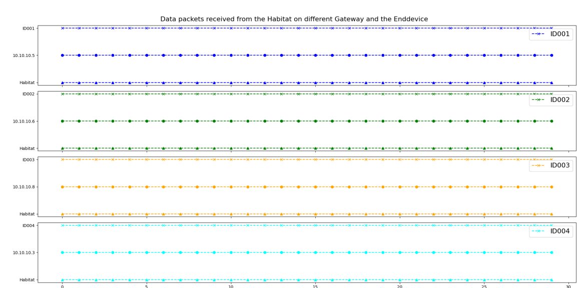


Fig. 15. Plots representing the Data packets sent by the habitat to the end-devices via specific gateways.

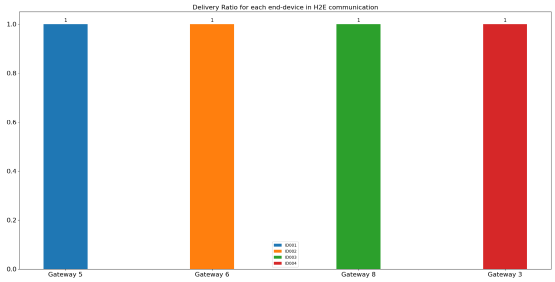
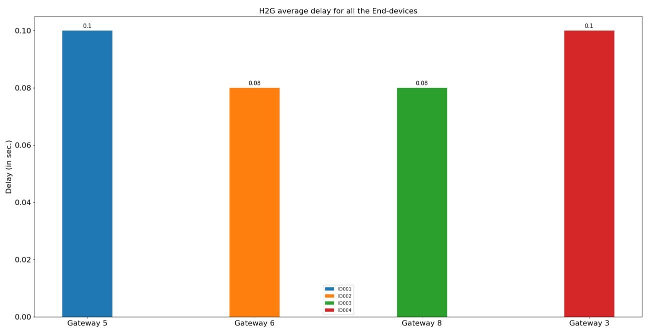
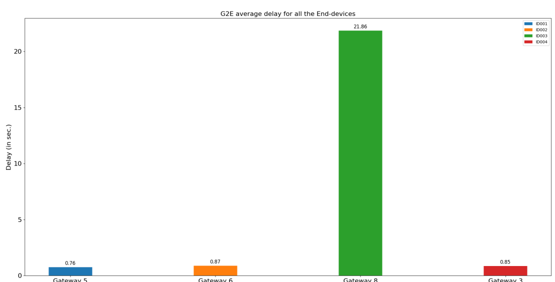


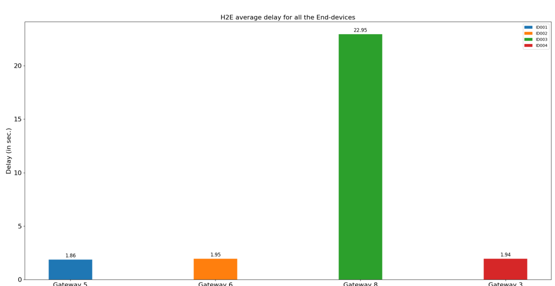
Fig. 16. Delivery ratio for H2E communication



(a) Average delay in the H2G communication



(b) Average delay in the G2E communication



(c) Average delay in the H2E communication

Fig. 17. Average delay (a) for receiving data packets at the gateways from the habitat, (b) for receiving data packets from the gateways at the end-devices and (c) overall average delay for receiving the data packets from the habitat at the end-devices.

and H2E communication. The H2G average delay plot here considers the delay from the point of transmission of the data packets from the habitat to the time stamp at which the data packet is received and stored at the gateway. Similarly, G2E average delay considers the delay from the point of transmission of the data packets from the gateway to the time stamp the packet is received and stored at the end-device. The delays for H2G (see Fig. 17(a)) and G2E (see Fig. 17(b)) are on the scale of less than 1 sec. except for ID003 in the case of G2E communication (see Fig. 17(b)). The high delay for ID003 in G2E communication is due to the miscalculation in the offset of the time-synchronization. To calculate the offset on the end-devcies, decimal library [14] was used. The decimal library sometimes gives incorrect results after calculation with big numbers because it is mainly limited by available memory and process power on the end-devices [14]. This error in the average delay of ID003 is then propagated to the overall H2E average delay (see Fig. 17(c)).

The above results represent that the H2E communication has lower average delay and better delivery ratio compared to the E2H communication. It is justified by the fact that in G2H communication, data is sent from multiple gateways in the form of GT messages and data packets to a single habitat through TCP sockets, which then wait in the incoming socket connection queue at the habitat.

V. Conclusion and Future Work

In this preliminary work, we presented the implementation of a communication infrastructure enabling astronauts and probes to send and receive data from the habitat. The network includes gateways and habitat using OLSR as a routing protocol. The end-devices use the gateways to relay the information to the habitat using multiple hops in the Network and vice versa.

The experimental scenario and the results show the working of our system, with end-devices being able to send data to the habitat and habitat capable of using the geography table to send data to a specific end-device. The end-devices were capable of synchronizing time with the network using specific gateways. For future work, we intend to support E2E communication in the network, unicast data packets from an end-device to a gateway, store data on gateway buffer in case of unavailability of the habitat, support Class A and Class B LoRa end-devices & introduce ACK in E2G and G2E communication.

References

[1] F. Castellini, A. Simonetto, R. Martini, and M. Lavagna, “A mars communication constellation for human exploration and network science,” *Advances in Space Research*, vol. 45, no. 1, pp. 183–199, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117709006826

[2] L. McMillon-Brown, T. J. Peshek, A. Pal, and J. McNatt, “A study of photovoltaic degradation modes due to dust interaction on mars,” *Solar Energy Materials and Solar Cells*, vol. 221, p. 110880, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927024820304785

[3] G. Kaur and P. Thakur, “Routing protocols in manet: An overview,” in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, 2019, pp. 935–941.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” pp. 62–68, 2001.

[5] K. Lois, "Comparative study of reactive routing protocols for manets." [Online]. Available: https://www.ijana.in/Special%20Issue/C32.pdf
[6] D. Johnson, D. Maltz, and J. Broch, "Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Networking*, vol. 5, 01 2002.
[7] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," pp. 90–100, 1999.
[8] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," p. 66–75, 1998. [Online]. Available: https://doi.org/10.1145/288235.288252
[9] P. Theerthagiri, P. Rajakumar, and A. Pitchaikkannu, "Overview of proactive routing protocols in manet," 04 2014, pp. 173–177.
[10] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, 05 1999.
[11] E. E. Akkari Sallum, G. dos Santos, M. Alves, and M. M. Santos, "Performance analysis and comparison of the dsdv, aodv and olsr routing protocols under vanets," pp. 1–7, 2018.
[12] D. Mills, *Computer Network Time Synchronization: The Network Time Protocol*. CRC Press, 2006. [Online]. Available: https://books.google.de/books?id=pdTcJBfnbq8C
[13] T. W. of Oz. Ntp timestamp calculations. [Online]. Available: https://www.eecis.udel.edu/~mills/time.html
[14] mpy dev, "micropython-decimal-number," https://github.com/mpy-dev/micropython-decimal-number, 2021.