

Overview

Communication is a subsystem responsible for bridging the two distinct networks present on the rover: the Ethernet Network and the ROS2 Network. Without it, Jonny Boi (the Jetson Orin Nano) would have no way to talk to the hardware microcontrollers or the Basestation, and the internal ROS2 software stack would have no way to act on external inputs or send commands outward.

On the Ethernet side, the Jetson connects to the hardware boards (ArmBoard, DriveBoard, SensorBoard, ContainerBoard) and to the Basestation over WiFi. On the ROS2 side, the Communications node exchanges messages with internal nodes such as the Behavior Node, which handles business logic, and the Controller Node, which manages hardware commands.

All messages sent over the network, whether towards the Basestation or the hardware microcontrollers, are encoded as Protobufs, wrapped inside a [PBEnvelope](#). The PBEnvelope acts as a typed container that lets the receiver identify what kind of message it is receiving before deserializing the payload. These wrapped messages are transmitted as raw bytes inside UDP packets.

The Communications node operates in both directions:

- Inbound (Ethernet -> ROS2): UDP packets received from the Basestation or hardware boards are parsed, the protobuf payload is extracted from the PBEnvelope, and the data is converted into custom ROS2 messages before being published over the appropriate ROS2 topic for other nodes to consume.
- Outbound (ROS2 -> Ethernet): The Communications node subscribes to relevant ROS2 topics, converts the incoming ROS2 messages back into Protobufs, wraps them in a PBEnvelope, and transmits them as UDP packets to the intended destination, either the Basestation or a hardware board.

Consider the Rover Architecture diagram [Rover Architecture diagram](#) to get a visual overview of how the Communications node sits at the center of these data flows.

Before diving into the codebase, it is strongly recommended to read the [ROS2 documentation](#), particularly the entries regarding topics, messages and publishers/subscribers:

<https://docs.ros.org/en/humble/Concepts/Basic/About-Interfaces.html>

You will also need background knowledge in C++ and UDP networking. The following guide is a solid starting point:

<https://beej.us/guide/bgnet/html/split-wide/>

Revision #21

Created 2026-04-14 09:34:51 UTC by Dmytro Khorsun

Updated 2026-04-16 14:16:27 UTC by Andrei Badea