The Bowler RPC

A zeroconf protocol for creating PC-microcontroller networks

Ryan Benasutti
Common Wealth Robotics Cooperative

March 2019

1 Overview

1.1 Motivation

1. We want a PC to talk to any Device which is running some Bowler software. We want this software to require minimal to no configuration from the user.
2. We want to support any Resource attached to the Device with minimal to no configuration from the user.
3. We require the communications between the PC and Device satisfy hard real-time requirements: 5ms RTT, 100ms timeout.
4. Therefore, we settle on a zeroconf protocol which can be used by the PC to establish an RPC with the Device.

1.2 Configuration Process

The PC-Device RPC is established in the following order

1. The PC connects to the Device using some Physical Layer implementation.
2. The PC sends Discovery packets to the Device to tell the device which Resources are connected to it. The Device may reject any packet if it deems the Resource invalid.
3. Once all Resources have been discovered, the Discovery process is finished and the PC and Device may use the configured RPC.

2 Discovery

2.1 Packet Format

2.1.1 General Discovery Packet Format

Figure 1 shows what the PC sends the device to initiate a discovery operation. Any additional operation-specific data is sent in the Payload section. The entire packet is 64 bytes.

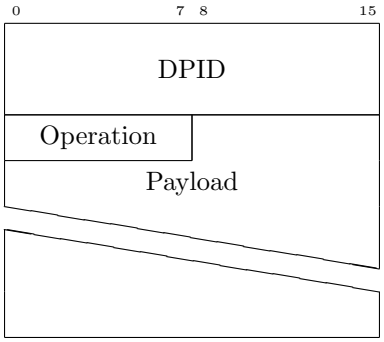

Figure 1: Discovery-time send packet format.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.
- Operation: 1 byte
  - The Operation field states the operation the packet performs.

Figure 2 shows what the device sends the PC to complete a discovery operation. Any additional operation-specific data is sent in the Payload section. The entire packet is 64 bytes.

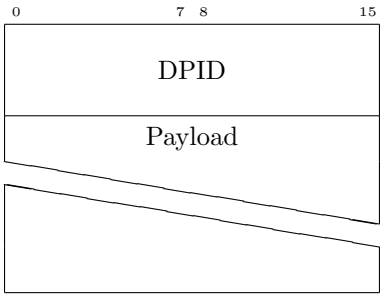

Figure 2: Discovery-time receive packet format.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.

2.1.2 Discovery Packet

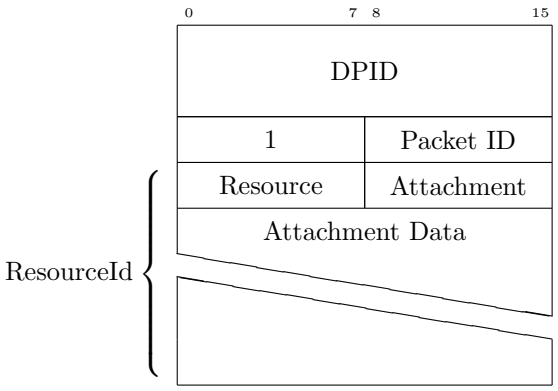

Figure 3: Discovery send packet.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.
- Packet ID: 1 byte
  - The Packet ID field is a new ID for the Packet being discovered.
- Resource: 1 byte
  - The Resource field is the type of the resource. It is the `ResourceId.resourceType.type`.
- Attachment: 1 byte
  - The Attachment field is the type of the attachment point. It is the `ResourceId.attachmentPoint.type`.
- Attachment Data: 1+ bytes
  - The Attachment Data field is any data needed to fully describe the Attachment. It is the `ResourceId.attachmentPoint.data`.

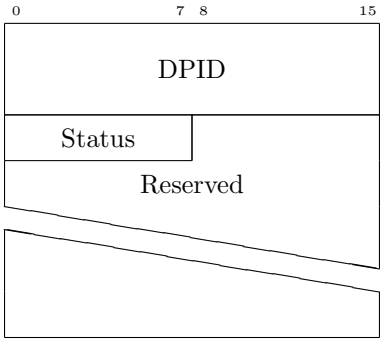

Figure 4: Discovery receive packet.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.
- Status: 1 byte
  - The Status field encodes the status of the discovery operation. 1 = Accepted, 2 = Rejected.

2.1.3 Group Discovery Packet

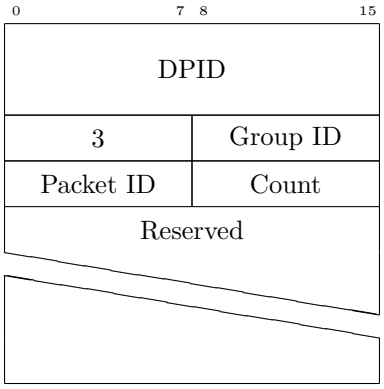

Figure 5: Group discovery send packet.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.
- Group ID: 1 byte

- The Group ID field is the ID for the group being made. Future Group Member Discovery Packets will need this ID to add Resources to the correct group.

- Packet ID: 1 byte
  - The Packet ID field is the ID for the packet the Group will use. All Resources in the Group get packed into one packet.

- Count: 1 byte
  - The Count field is the number of Resources that will be added to the group.

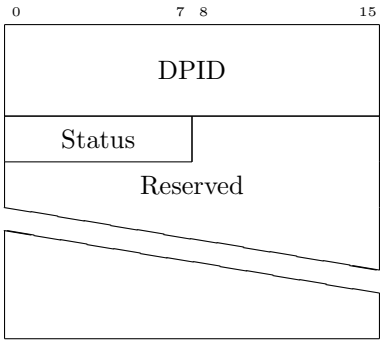

Figure 6: Group discovery receive packet.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.

- Status: 1 byte
  - The Status field encodes the status of the discovery operation. 1 = Accepted, 2 = Rejected.

2.1.4 Group Member Discovery Packet

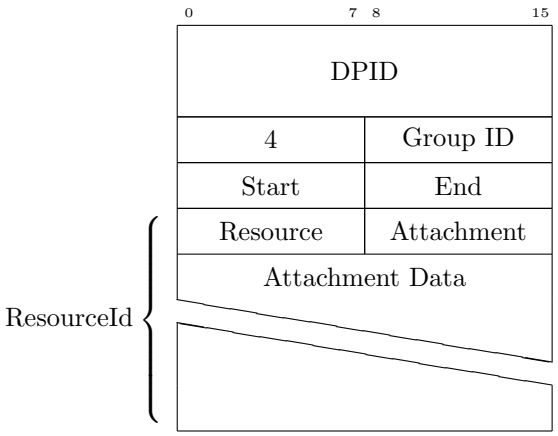

Figure 7: Group member discovery send packet.

- Group ID: 1 byte
  - The Group ID field is the ID for the Group that this Resource will be added to.
- Start: 1 byte
  - The Start field is the starting byte index in the response Payload for this Resource's response data.
- End: 1 byte
  - The End field is the ending byte index in the response Payload for this Resource's response data.
- Resource: 1 byte
  - The Resource field is the type of the resource. It is the `ResourceId.resourceType.type`.
- Attachment: 1 byte
  - The Attachment field is the type of the attachment point. It is the `ResourceId.attachmentPoint.type`.
- Attachment Data: 1+ bytes
  - The Attachment Data field is any data needed to fully describe the Attachment. It is the `ResourceId.attachmentPoint.data`.

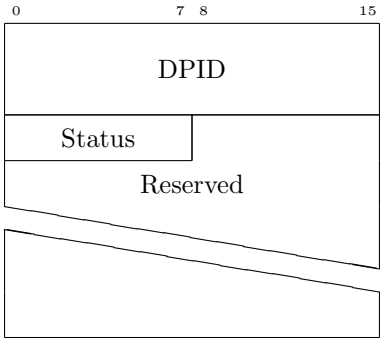

Figure 8: Group member discovery receive packet.

- DPID (Discovery Packet ID): 4 bytes
  - The DPID field is typically filled by SimplePacketComs and contains the ID for the packet it is contained in.
- Status: 1 byte
  - The Status field encodes the status of the discovery operation. 1 = Accepted, 2 = Rejected.

2.2 Discovery Process

2.2.1 Discovery

Sequence diagram for

1. Send discovery packet and get response (accepted).
2. Send discovery packet and get response (rejected).

2.2.2 Group Discovery

Sequence diagram for

1. Send group discovery packet and get response (accepted). Send multiple group member discovery packets and get responses (accepted).
2. Send group discovery packet and get response (accepted). Send multiple group member discovery packets and get responses (most accepted, some rejected).
3. Send group discovery packet and get response (rejected). Send multiple group member discovery packets and get responses (rejected).

3 RPC

3.1 Packet Format

3.1.1 Non-Group

Packets for non-Group Resources correspond to a single Resource. These Resources do not have any timing constraints.

3.1.2 Group

Packets for Group Resources correspond to multiple Resources whose responses are packed into a single packet. These Resources can have timing constraints and are therefore put into a Group.