Group4 Protocol Specification

Jl1251 hl490 jm907 zz315 sl846 sx83 hg161 zw297

1. Introduction

This protocol specification defines the communication protocol between Amazon (A) and UPS (U) systems, facilitating the exchange of information for transportation logistics.

2. Terminology

- **A (Amazon)**: Represents the Amazon system.
- **U (UPS)**: Represents the UPS system.

3. Messages that can be sent

3.1. AUCommands

```
message AUCommands {
  repeated AUNeedATruck need = 1;
  repeated AUTruckCanGo go = 2;
  repeated Err errors = 3;
  repeated int64 acks = 4;
}
```

This is the message Amazon sends to UPS, and all the details are embedded in this message, which we will discuss in the next section.
For each field in this message, they MAY be provided. However, for the entire message, it MUST contain at least one field, in other words, the entire message MUST NOT be empty (there is no point in sending an empty message).

3.2. UACommands

```
message UACommands {
    repeated UATruckArrived arrived = 1;
    repeated UADelivered delivered = 2;
    repeated UAChangeAddr changeAddr = 3;
    repeated Err errors = 4;
    repeated int64 acks = 5;
}
```

This is the message UPS sends to Amazon, and all the details are embedded in this message, which we will discuss in the next section.
For each field in this message, they MAY be provided. However, for the entire message, it MUST contain at least one field, in other words, the entire message MUST NOT be empty (there is no point in sending an empty message).

3.3. UAInitConnect

```
message UAInitConnect{
  required int64 worldid = 1;
}
```

- Purpose: UPS sends this message to let Amazon know which world it should connect to.
- Fields:
  - worldid: Indicates the world id.
- Requirements:
  - worldid MUST be provided.

3.4. AUConfirmConnect

```
message AUConfirmConnect{
  required int64 worldid = 1;
  required bool connected = 2;
}
```

- Purpose: Confirms that Amazon now knows which world to connect to.
- Fields:
  - worldid: Indicates the world id that amazon and ups should connect to.
  - connected: Indicates whether the connection is successful (true) or not (false).
- Requirements:
  - worldid MUST be provided.
  - connected MUST be provided.

4. Message Definitions and Usage Scenarios

4.1. A Messages (starts with AU, means Amazon sends to UPS)

4.1.1. AUNeedATruck

```
message AUNeedATruck{
  required Pack pack = 1;
  required int64 seqnum = 2;
}
```

- Scenario: When Amazon needs UPS to send a truck to the warehouse to load packages, it sends UPS this message
- Purpose: Requests a truck for transportation from UPS.
- Fields:
  - pack: The Pack stucture contains package information.
  - seqnum: Sequence number for message identification.
- Requirements:
  - pack MUST be provided.
  - seqnum MUST be provided.

4.1.2. AUTruckCanGo

```
message AUTruckCanGo{
  required int32 truckid = 1;
  required int64 seqnum = 2;
}
```

- Scenario: When the load is complete, Amazon sends this message to UPS to tell that the truck is ready to leave.
- Purpose: Indicates that a truck is ready to leave with the loaded orders.
- Fields:
  - truckid: Identifier of the truck.
  - seqnum: Sequence number for message identification.
- Requirements:
  - truckid MUST be provided.
  - seqnum MUST be provided.

4.2. U Messages (starts with UA, means UPS sends to Amazon)

4.2.1. UATruckArrived

```
message UATruckArrived{
  required int32 truckid = 1;
  required string trackingid = 2;
  required int32 wh_id = 3;
  required int64 seqnum = 4;
}
```

- Scenario: When a UPS truck arrives at an Amazon warehouse, UPS sends this message to Amazon.
- Purpose: Notifies Amazon that a truck has arrived at warehouse.
- Fields:
  - truckid: Identifier of the truck.
  - trackingid: Identifier of the package should be loaded.
  - wh_id: Identifier of the truck has arrived at which warehouse.
  - seqnum: Sequence number for message identification.
- Requirements:
  - truckid MUST be provided.
  - trackingid MUST be provided.
  - wh_id MUST be provided.
  - seqnum MUST be provided.

4.2.2. UADelivered

```
message UADelivered{
  required string trackingid = 1;
  required int32 truckid = 2;
  required int64 seqnum = 3;
}
```

- Scenario: When a package is delivered, UPS sends this message to let Amazon know so that Amazon can update package status.
- Purpose: Indicates that a truck has delivered an order to the user.
- Fields:
  - trackingid: Identifier of the package has been delivered.
  - truckid: Identifier of the truck.
  - seqnum: Sequence number for message identification.
- Requirements:

- trackingid MUST be provided.
- truckid MUST be provided.
- seqnum MUST be provided.

4.2.3. UAChangeAddr

```
message UAchangeAddr{
  required string trackingid = 1;
  required int32 dest_x = 2;
  required int32 dest_y = 3;
  required int64 seqnum = 4;
}
```

- Scenario: When the user chooses to change the shipping address on the UPS website, if the address is allowed to be changed and is changed, UPS sends this message to remind Amazon.
- Purpose: Alert Amazon shipping address has been modified.
- Fields:
  - trackingid: Identifier of the package has been delivered.
  - dest_x: The x-coordinate of the new shipping address
  - dest_y: The y-coordinate of the new shipping address
  - seqnum: Sequence number for message identification.
- Requirements:
  - trackingid MUST be provided.
  - dest_x MUST be provided.
  - dest_y MUST be provided.
  - seqnum MUST be provided.

4.3. Shared Messages

4.3.1. Pack

```
message Pack{
  required int32 wh_id = 1;
  repeated Product things = 2;
  required string trackingid = 4;
  required int64 packageid = 5;
  optional int32 upsaccount = 6;
  required int32 amazonaccount = 7;
  required int32 dest_x = 8;
```

```
    required int32 dest_y = 9;
}
```

- Purpose: Represents an package to be transported, containing essential information such as coordinates, tracking ID, and account details.
- Fields:
  - wh_id = The warehouse containing the package.
  - things = The products in this package.
  - trackingid = The trackingid of this package.
  - packageid = The packageid of this package.
  - upsaccount = The UPS account of the user.
  - amazonaccount = The Amazon account of the user.
  - dest_x = The x_coordinate of shipping address of the package.
  - dest_y = The y_coordinate of shipping address of the package.
- Requirements:
  - wh_id MUST be provided.
  - things MUST be provided.
  - trackingid MUST be provided.
  - packageid MUST be provided.
  - Upsaccount MAY be provided.
  - amazonaccount MUST be provided.
  - dest_x MUST be provided.
  - dest_y MUST be provided.

4.3.2. Product

```
message Product{
    required int64 id = 1;
    required string description = 2;
    required int32 count = 3;
}
```

- Purpose: Represents an Product, this is just inherited from world amazon protocol.
- Fields:
  - id = The unique id of the product.
  - description = The description of the product.
  - count = The quantity of the product.
- Requirements:

- id MUST be provided.
- description MUST be provided.
- count MUST be provided.

4.3.3. Err

```
message Err{
  required string msg = 1;
  required int64 originseqnum = 2;
  required int64 seqnum = 3;
}
```

- Purpose: Represents an error message, indicating a failure or problem encountered.
- Fields:
  - msg: Description of the error message.
  - originseqnum: Sequence number of the original message that triggered the error.
  - seqnum: Sequence number for message identification.
- Requirements:
  - msg MUST be provided.
  - originseqnum MUST be provided.
  - seqnum MUST be provided.

5. overall communication process

When Amazon receives an order and the warehouse has inventory, it will send topack to the world and send **AUNeedATruck** to UPS. After receiving this message, UPS should reply an **ack** to Amazon as confirmation. When the UPS truck arrives at Amazon's warehouse, it will send **UATruckArrived** to inform Amazon that the truck has arrived, and Amazon will reply **ack** to confirm. At the same time, if the package is already packed, Amazon can send load to the world to start loading. After the packages are loaded, Amazon will send the message **AUTruckCanGo** to UPS to inform UPS that the packages have been loaded and the truck can leave. UPS should send **ack** as confirmation. Whenever a package is delivered, UPS sends **UADelivered** to Amazon to inform it of the situation, and Amazon sends **ack** as confirmation..