Project: Mini-Amazon/ Mini-UPS

ECE 568 – Spring 2019

Interoperability Group 1
Team 1 (yx139, qc36)
Team 9 (rg241, yh218)
Team 17 (kw283, zh89)
Team 25 (yl543, cn104)

2019 April 8

Protocol Specification

We would use Google Protocol Buffer as the Amazon-UPS communication protocols.

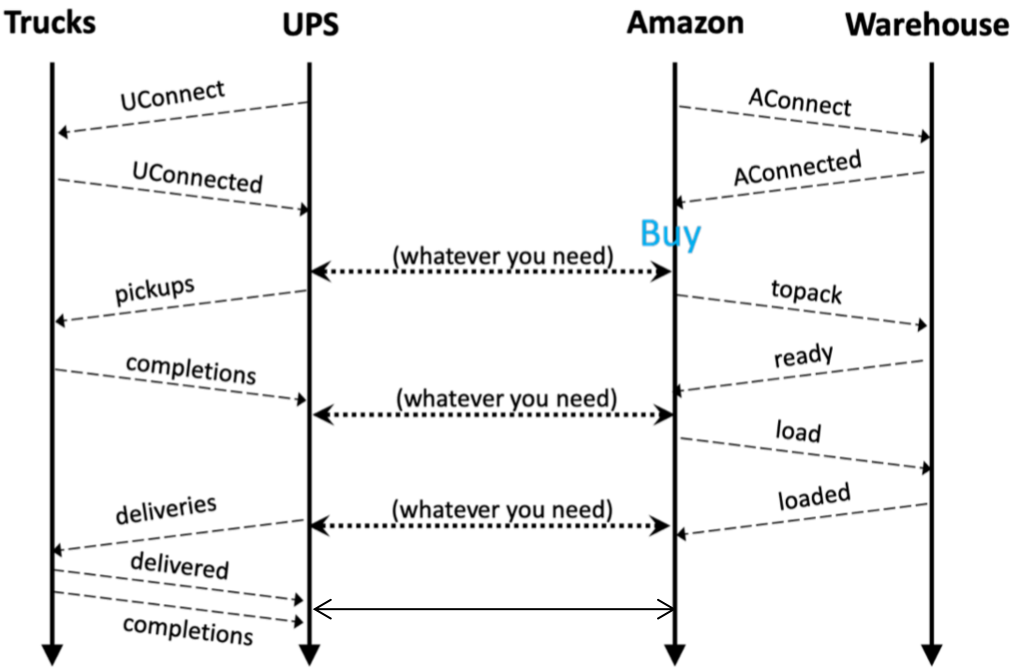

Fig.1. Amazon-UPS communication

The communications occur in four time points: order placed, truck arrives at Warehouse, loading finished and package delivered as shown in Fig.1.

Protocol Details

Amazon and UPS should send A/U Communicate. All the details are embedded in the A/U Communicate like how it works in the communication with the world.

A/U Communicate implement ack numbers as well. For each request inside A/U Communicate, there is a seqnum. When the other side receives the commands from one side, it will check the seqnum of each request. Then it will process the request and return responses with acks of those seqnums.

Overall, the A/U Communicate commands are as follow:

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

message ACommunicate{
    repeated AReadyForPickup areadypick = 1;
    repeated AOrderPlaced aorderplaced = 1;
    repeated ALoadingFinished aloaded= 2;
    repeated int64 acks = 3;
}

message UCommunicate{
    repeated UReadyForPickup ureadypick = 1;
    repeated UOrderPlaced uorderplaced = 1;
    repeated UArrivedAtWarehouse uarrived= 2;
    repeated UPackageDelivered udelivered=3;
    repeated int64 acks = 4;
}
```

1. **Package Ready for Pickup**

Amazon -> UPS:

```
message Product{
    required string name =  1;               //name of item
    required string description = 2;         //description
    required int32 count = 3;                //amount
}
message AOrderPlaced{
    required int32 whid = 1;                 //warehose id
    required int32 x =2;
    required int 32 y =3;                    //delivery address
    optional int64 packageid = 4;            //world package ID
    required int64 packageid = 4;            //world package ID
    repeated Product things = 5;             //products in one package
    optional int64 UPSuserid =6;             //UPS user id
    required int64 seqnum = 7;               //sequence number for ACK
}
```

The message sent from Amazon to UPS MUST have warehouse id to identify which warehouse the truck should go to; MUST have delivery address to identify the destination of the truck; MAY have package id to identity which package the truck should load; MAY have the UPS user id to identify which user made the order; Must have the sequence number to make sure the message is delivered and acknowledged.

UPS -> Amazon:

```
message UOrderPlaced{
    required int64 packageid = 1;             // world package ID
    required int64 tracknum = 2;             //tracking number
    required int32 truckid = 2;              //truck id
    required int64 seqnum = 3;               //sequence number for ACK
}
```

When UPS receives Ready_For_pickup message from Amazon, UPS MUST send ACK to Amazon.

And the message sent from UPS MUST contain the package id to identity which package the truck has loaded; must has a tracking number to check the shipment; Must have the sequence number to make sure the message is delivered and acknowledged.

2. Truck Arrives at Warehouse

Amazon -> UPS

When truck arrives at the warehouse, the UPS site should send the truck information to the Amazon site, indicating that the truck has arrived at the warehouse and waiting for the package loading. And the Amazon site MUST acknowledge that it has received the information.

UPS -> Amazon

```
message UArrivedAtWarehouse{
    required int32 truckid = 1;              //truck id
    required bool arrived = 2;               //arrived at warehouse
    required int64 seqnum = 2;               //sequence number for ACK
}
```

When UPS arrives at the Warehouse, UPS MUST send truck id to identify which truck should be loaded; MUST send one bool to indicate it has arrived and is ready to be loaded; Must have the sequence number to make sure the message is delivered and acknowledged.

3. Loading Finished

Amazon -> UPS

```
message ALoadingFinished{
    required bool loaded = 1;                //loaded
    required int64 packageid = 1;            //world package ID
    required int32 truckid = 2;              //truck id
    required int64 seqnum = 3;               //sequence number for ACK
}
```

When Amazon loads package successfully, it MUST send one bool to indicate that it has finished loading; MUST send truck id to identify which truck is ready to deliver; Must send the sequence number to make sure the message is delivered and acknowledged.

UPS -> Amazon

When received loading finished message from Amazon, the UPS MUST response with ACK to confirm.

4. Package Delivered

Amazon -> UPS

When received the package delivered message from UPS, the Amazon MUST response with ACK to confirm.

UPS -> Amazon

```
message UPackageDelivered{
    required int64 tracknum = 1;              //tracking number
    required int64 packageid =1;              // world package ID
    required int64 seqnum = 2;                //sequence number for ACK
}
```

When Package delivered successfully, UPS MUST send the tracking number to identify which package has been delivered; Must have the sequence number to make sure the message is delivered and acknowledged.