Protocol Specs

Group 6

Introduction

We will use Google protocol buffer for communication between amazon and UPS. Each inter-communication MUST involve three-way handshake

RFC

- Amazon MUST open a listening socket at port 6666 and UPS MUST connect to that Socket. UPS MUST connect to the simulator to create a new world and MUST then send World ID to let amazon know the world it refers to, then Amazon MUST send a response indicating whether the connection to that world is Success or Failure back to UPS:
  Request:

```
message UAConnect{
    required int64 worldid = 1;
    required int64 seqnum = 2;
};
```

  Response:

```
message AUConnected{
    required bool world_connection_status = 1;
    required int64 seqnum = 2;
};
```

- After every purchase, Amazon SHOULD notify UPS about the warehouse to pickup packages and MUST send Warehouse ID and a single package info. Note that UPS does not need to know the warehouse location for pickup, this detail is handled by the world simulator and is hidden from the UPS side. Furthermore UPS should ignore any subsequent AURequestForPickup message with same warehouse id if it detects there is already a truck en route to that warehouse, and collect that packageInfo with the same warehouse id (to be sent back with UAReadyForPickup):

```
message AUPackageInfo{
    required int64 packageid = 1;
    optional string upsAccount = 2;
    required string description = 3;
    required int32 destx = 4;
    required int32 desty = 5;
    required AUDeliveryLocation packagelocation = 2;
};
```

```
message AURequestPickup{
        required AUPack pack = 1;
        required int64 seqnum = 2;
        };

message AUPack{
        required Apack package = 1;
        optional string upsAccount = 2;
        required int32 destx = 3;
        required int32 desty = 4;
};
```

- For each packageInfo, the UPS MUST include one UAIsAssociated message in the response, this is sent inside the UACommand together with the acks for AURequestForPickup:

```
        Message UAIsAssociated{
                Required int64 packageid = 1;
                Required bool checkResult = 2;
        };
```

- UPS SHOULD notify Amazon that it has arrived at the warehouse and is ready to pick up the package and MUST send the warehouse ID and the TruckID of the truck assigned, UPS should also include every packages that is ready to be loaded in that warehouse according to packageinfo sent with AUReadyForPickup.

```
        message UAReadyForPickup{
                required int32 whnum = 1;
                repeated int64 packageid = 2;
                required int32 truckid = 3;
                required int64 seqnum = 4;
        };
```

- Amazon MUST notify UPS that all the packages have been loaded to the truck and for each package in the truck Amazon MUST send related information : For each package, Amazon MUST include packageid, description, location for delivery; additionally, Amazon MAY include the UPS username should the customer chooses to relate the purchase with his UPS account- Package delivery address, WarehouseID.

```
        message AUDeliveryLocation{
                required int64 packageid = 1;
                required int32 x = 2;
                required int32 y = 3;
        };

        message AUReadyForDelivery{
                required int32 truckid = 1;
```

```
        repeated AUPackageInfo amazonpackageInfo = 2;
        required int64 seqnum = 2;
    };
```

- UPS MUST notify Amazon about the status of the packages whenever the package is delivered and MUST send related PackageID.

```
    message UAPackageDelivered{
        required int64 packageid = 1;
        required int64 seqnum = 2;
    };
```

Command Wrapper:
All the above message(except UAConnect) needs to be send in this form

```
    message UACommand{
        repeated UAReadyForPickup pickupReady = 1;
        repeated UAPackageDelivered packageDelivered =2;
        repeated UAIsAssociated linkResult =3;
        repeated Err error = 4;
        repeated int64 acks = 5;
    };

    message AUCommand{
        repeated AURequestPickup pickupRequest = 1;
        repeated AUReadyForDelivery deliveryReady =2;
        repeated Err error = 3;
        repeated int64 acks = 4;
        };

    message Err{
        required string errorInfo = 1;
        required int64 originseqnum = 2;
        required int64 errorSeqnum = 3;
    };
```

Note that packageID and seqnum must be UNIQUE at all times!

Lucid Chart
Page 1 : UML diagram
Page 2 : Stade machine

https://lucid.app/lucidchart/d2e239e6-d711-436d-ac2a-eac336c1608f/edit?invitationId=inv_d46b2712-7c5d-461f-b2d3-b3929d91aefa