568 Final Project
Protocol Specification

Group 5:
yc557, yj193, za53, xw164, zs149, fq18, ys385, zx109, sz279, cy141

amazon_ups.proto:

```
syntax = "proto2";


// request
message AUCommand{
 repeated AUPickupRequest pickupRequests = 1;
 repeated AUDeliverRequest deliverRequests = 2;
 repeated int64 acks = 3;
 repeated Err error = 4;
}


// response
message UACommand{
 repeated UALoadRequest loadRequests = 1;
 repeated UADelivered delivered = 2;
 repeated int64 acks = 3;
 repeated Err error = 4;
}

// UPS to Amazon: UPS creates a world for Amazon to connect to
message UAstart {
 required int64 worldid = 1;
 required int64 seqnum = 2;
}

// Amazon to UPS: when Amazon received a Buy command, it send APickupRequest to
UPS to prepare a truck sent to target warehouse
message AUPickupRequest{
 required int64 seqNum = 1;
 required int64 shipId = 2;
 required int32 warehouseId = 3;
 required int32 x = 4; // location of the warehouse
 required int32 y = 5; // location of the warehouse
 required int32 destinationX = 6;
 required int32 destinationY = 7;
 optional string upsName = 8;
 required string items = 9;
}


// U -> A Arrived, ready to load
message UALoadRequest{
 required int64 seqNum = 1;
 required int32 truckId = 2;
 required int64 shipId = 3;
}



// A -> U: when all ready, make UPS deliver the package
message AUDeliverRequest{
 required int64 seqNum = 1;
 required int64 shipId = 2;
}



// U -> A: delivered
message UADelivered{
 required int64 seqNum = 1;
 required int64 shipId = 2;
}


message Err{
 required string err = 1;
 required int64 originSeqNum = 2;
 required int64 seqNum = 3;
}

// ack response

message Response {
 repeated int64 acks = 3;
 repeated Err error = 4;


}
```

Possible Interaction Scenarios:

1. Every request **MUST** have a sequence number. All the requests are wrapped within UACommand and AUCommand.

2. When server starts, ups **SHOULD** send a UAstart to amazon, to ensure the connection if established
3. When Amazon receives a purchase order, it **SHOULD** send an AUPickupRequest to UPS, which **MUST** have a shipId, location of the warehouse, location of the destination, and name of the item. The request **MAY** have a ups account name, in order to request a truck to be sent to the targeted warehouse.
4. When UPS sends the truck to the target warehouse, it **SHOULD** send UALoadRequest to Amazon to load the packages with shipId on the truck referenced by the corresponding truckId.
5. When Amazon finishes loading, it **SHOULD** send AUDeliverRequest to deliver packages.
6. When UPS delivers packages, it **SHOULD** send UADelivered to indicate package delivered.
7. Instead of receiving Response in our previous design, we put the content of these reponses in the AUcommand and UAcommand, and only send and recv AUcommand/UAcommand