

# ECE568 Project Mini Amazon/UPS Interoperability Group 4

## Protocol Spec Document

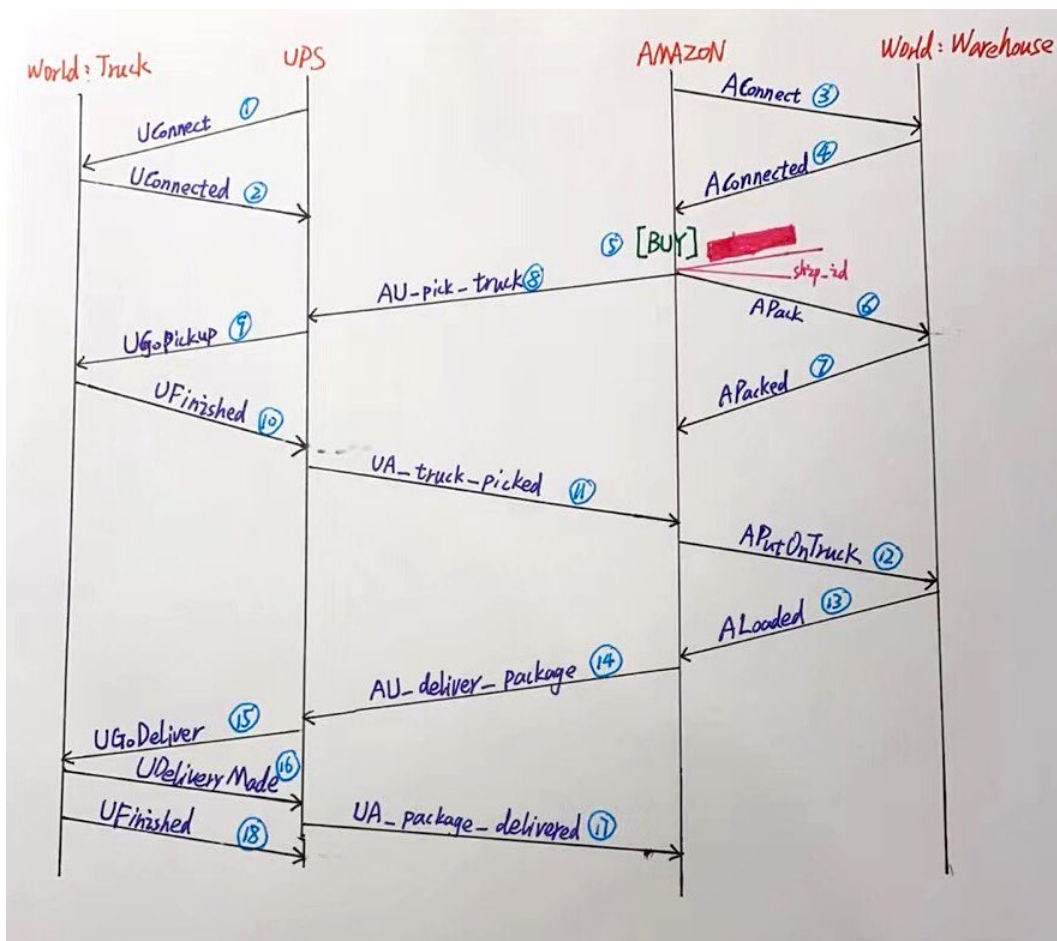
Group Members: Zhiwei Zhou(zz252), Jinxin Ji(jj343), Mengqi Liu(ml607), Peiru Liu(pl202), Jichen Zhang(jz367), Zian Wang(zw142), Hongyang Li(hl404), Hao Wu(hw269)

### 1. Language Requirements

Group members MAY use any language they deem appropriate, as long as that programming language is compatible with Google Protocol Buffer, which is the format of information transmission we chose.

### 2. Basic Timeline

The system consists of three parts: the “world” server, the mini-UPS server as well as the mini-Amazon server. The programs built by group members MUST support the basic system-timeline shown in the following picture.



The things that the system MUST support are:

1. UPS sending message UConnect to World.
2. World sending message UConnected to UPS.
3. Amazon sending message AConnect to World.
4. World sending message AConnected to Amazon.
5. Amazon receiving BUY request from front-end web page.
6. Amazon sending message APack to World.
7. World sending message APacked to Amazon.
8. Amazon sending message AU\_pick\_truck to UPS.
9. UPS sending message UGoPickup to World.
10. World sending message UFinished to UPS.
11. UPS sending message UA\_truck\_picked to AMAZON.
12. Amazon sending APutOnTruck to World.
13. World sending ALoaded to Amazon.
14. Amazon sending AU\_deliver\_package to UPS.
15. UPS sending UGoDeliever to World.
16. World sending UDeliveryMade to UPS.
17. UPS sending UA\_package\_delivered to AMAZON.
18. World sending UFinished to UPS.

### 3. Amazon\_UPS\_protocol Definition

The messages we defined in our .proto file are as follows.

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

```
message AU_pick_truck{
    required int64 shipid = 1;
    required int32 whid = 2;
    required int64 seqnum = 3;
}
```

```
message UA_truck_picked{
    required int64 shipid = 1;
    required int32 truckid = 2;
    required int64 seqnum = 3;
}
```

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

```
message AU_deliver_package{
  required int64 shipid = 1;
  repeated UDeliveryLocation packages = 2;
  required int32 truckid = 3;
  required int64 seqnum = 4;
}
```

```
message UA_package_delivered{
  repeated int64 shipid = 1;
  required int64 seqnum = 2;
}
```

// This is a higher-level class that encapsulates all the messages sent from Amazon to UPS

```
message AU_commands{
  repeated AU_pick_truck pick = 1;
  repeated AU_deliver_package deliver = 2;
  repeated int64 acks = 3;
  repeated err errors = 4;
}
```

// This is a higher-level class that encapsulates all the messages sent from UPS to Amazon

```
message UA_commands{
  repeated UA_truck_picked pick = 1;
  repeated UA_package_delivered deliver = 2;
  repeated int64 acks = 3;
  repeated err errors = 4;
}
```