

ECE568 project protocol spec

Group 7

yd171 jx133
ks713 yw479
yz674 jx139
yx236 xh123

Protocol specification (Updated)

The key words "**MUST**", "**SHOULD**" and "**MAY**", in this document are to be interpreted as described in [\[RFC2119\]](#).

- Amazon and UPS **MUST** successfully connect to *the World* server before any further operations. After Amazon connected to *the World*, it sends a **AUConnectedToWorld** message to UPS, which **MUST** include the worldid that Amazon has already connected to. After UPS connects to the same *the World*, it sends a **UAConnectedToWorld** message back, which **MUST** include the same worldid that received from Amazon.
- A customer initiates an order on Amazon's website. Amazon communicates with *the World* to purchase the products with the required description and count. Products are placed into a warehouse when ready.
- Amazon sends **AUOrderCreated** to UPS to notify the creation of an order. A **AUOrderCreated** message **MUST** include orderid, coordinates(x, y) of destination and **MAY** include the username of UPS account.
- Amazon sends **AURequestTruck** command to UPS to request for a truck. A **AURequestTruck** message **MUST** include whnum, x and y to specify the serial number and location of the warehouse, which is the destination of the truck.
- Amazon communicates with *the World* to pack the products. Note that pack operations **SHOULD** start right after products are ready in a warehouse, but **MAY** start before the truck arrives.
- UPS sends a truck to the warehouse specified by Amazon. When it arrives, UPS sends a **UATruckArrived** message to Amazon. A **UATruckArrived** message **MUST** include a truckid that is related to the truck; it **MUST** also include whnum that specifies the destination warehouse.
- After the truck arrives, Amazon communicates with *the World* to load the products to the truck. Amazon sends a **AUOrderLoaded** message to UPS once the load operation is finished. A **AUOrderLoaded** message **MUST** include orderid, truckid packageid and the description of the content of the package.
- After the truck departs for delivery, UPS sends Amazon with **UAOrderDeparture** to notify the departure of the truck, which **MUST** includes orderid, packageid and trackingnum that generated by UPS.
- Before the package is out for delivery, the user **MAY** change the destination of delivery. If that happens, UPS sends a **UADestinationUpdated** message to Amazon, which **MUST** include orderid and the updated destination (x, y).

- After products are loaded, the truck starts the delivery process. Once the package is delivered, UPS sends a **UAOrderDelivered** message to Amazon to notify the completion of the order. A **UAOrderDelivered** message **MUST** include packageid; it **MUST** also include x and y that specify the destination of delivery.
- Each message type mentioned above **MUST** include a seqnum field for further acknowledgement in the response.
- Each message type mentioned above **MUST** be wrapped into either **AUCommands** or **UACommands**. Each **AUCommands/UACommands** **MAY** include zero, one or more specific commands. It **MUST** include an ack field for further acknowledgement in the response. It **MAY** include zero, one or more Err message; an **Err** message **MUST** include an err string that describes the error; it **MUST** also include originseqnum of the source of error, and a seqnum for itself.

Protocol specification (Outdated)

The key words "**MUST**", "**SHOULD**" and "**MAY**", in this document are to be interpreted as described in [\[RFC2119\]](#).

- Amazon and UPS **MUST** successfully connect to *the World* server before any further operations.
- A customer initiates an order on Amazon's website. Amazon communicates with *the World* to purchase the products with the required description and count. Products are placed into a warehouse when ready.
- Amazon sends **AURequestTruck** command to UPS to request for a truck. A **AURequestTruck** message **MUST** include whnum, x and y to specify the serial number and location of the warehouse, which is the destination of the truck.
- Amazon communicates with *the World* to pack the products. Note that pack operations **SHOULD** start right after products are ready in a warehouse, but **MAY** start before the truck arrives.
- UPS sends a truck to the warehouse specified by Amazon. When it arrives, UPS sends a **UATruckArrived** message to Amazon. A **UATruckArrived** message **MUST** include a truckid that is related to the truck; it **MUST** also include whnum, x and y that specify the destination warehouse.
- After the truck arrives, Amazon communicates with *the World* to load the products to the truck. Amazon sends a **AUOrderLoaded** message to UPS once the load operation is finished. A **AUOrderLoaded** message **MUST** include truckid and packageid; it **MUST** also include x and y to specify the destination of delivery.
- After products are loaded, the truck starts the delivery process. Once the package is delivered, UPS sends a **UAOrderDelivered** message to Amazon to notify the completion of the order. A **UAOrderDelivered** message **MUST** include packageid; it **MUST** also include x and y that specify the destination of delivery.
- Each message type mentioned above **MUST** include a seqnum field for further acknowledgement in the response.
- Each message type mentioned above **MUST** be wrapped into either **AUCommands** or **UACommands**. Each **AUCommands/UACommands** **MAY** include zero, one or

more specific commands. It **MUST** include an ack field for further acknowledgement in the response.

- An **Err** message **MUST** include an err string that describes the error. It **MUST** also include originseqnum of the source of error, and a seqnum for itself.
- After receiving a **AUCommands/UACommands** message, the receiver **MUST** reply with a **AUResponse/UAResponse** message, which **MUST** include the original commands; it **MAY** include a boolean field finished to represent the status, and one or more **Err** message to describe the errors. It **MUST** include one or more acks corresponding to the same field in the received commands as an acknowledgement.

Flow Chart

<https://drive.google.com/file/d/13raWjyKHoxPXIEmJ38U0uNzmKj3BX74b/view?usp=sharing>

Source Code (Updated)

```
syntax = "proto2";
message AUConnectedToWorld {
    required int64 worldid = 1;
    required int64 seqnum = 2;
}

message UAConnectedToWorld {
    required int64 worldid = 1;
    required int64 seqnum = 2;
}

message AUOrderCreated {
    required int32 orderid = 1;
    required int32 destinationx = 2;
    required int32 destinationy = 3;
    optional string upsaccount = 4;
    required int64 seqnum = 5;
}

message UADestinationUpdated {
    required int32 orderid = 1;
    required int32 destinationx = 2;
    required int32 destinationy = 3;
    required int64 seqnum = 4;
}

message AURequestTruck {
    required int32 whnum = 1;
    required int32 x = 2;
```

```
    required int32 y = 3;
    required int64 seqnum = 4;
}

message UATruckArrived {
    required int32 truckid = 1;
    required int32 whnum = 2;
    required int64 seqnum = 3;
}

message AUOrderLoaded {
    required int32 orderid = 1;
    required int32 truckid = 2;
    required int64 packageid = 3;
    required string description = 4;
    required int64 seqnum = 5;
}

message UAOrderDeparture {
    required int32 orderid = 1;
    required int64 packageid = 2;
    required int64 trackingnum = 3;
    required int64 seqnum = 4;
}

message UAOrderDelivered {
    required int64 packageid = 1;
    required int32 destinationx = 2;
    required int32 destinationy = 3;
    required int64 seqnum = 4;
}

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

message AUCommands {
    repeated AUConnectedToWorld connectedtoworld = 1;
    repeated AUOrderCreated ordercreated = 2;
    repeated AURequestTruck requesttruck = 3;
    repeated AUOrderLoaded orderloaded = 4;
    repeated int64 acks = 5;
    repeated Err error = 6;
}
```

```
message UACommands {  
    repeated UAConnectedToWorld connectedtoworld = 1;  
    repeated UADestinationUpdated destinationupdated = 2;  
    repeated UATruckArrived truckarrived = 3;  
    repeated UAOrderDeparture orderdeparture = 4;  
    repeated UAOrderDelivered orderdelivered = 5;  
    repeated int64 acks = 6;  
    repeated Err error = 7;  
}
```

Source Code (Revised)

```
syntax = "proto2";

message AUConnectedToWorld {
  required int64 worldid = 1;
  required int64 seqnum = 2;
}

message UAConnectedToWorld {
  required int64 worldid = 1;
  required int64 seqnum = 2;
}

message AUOrderCreated {
  required int32 orderid = 1;
  required int32 destinationx = 2;
  required int32 destinationy = 3;
  optional string upsaccount = 4;
  required int64 seqnum = 5;
}

message UADestinationUpdated {
  required int32 orderid = 1;
  required int32 destinationx = 2;
  required int32 destinationy = 3;
  required int64 seqnum = 4;
}

message AURequestTruck {
  required int32 whnum = 1;
  required int32 x = 2;
  required int32 y = 3;
  required int64 seqnum = 4;
}

message UATruckArrived {
  required int32 truckid = 1;
  required int32 whnum = 2;
  required int32 int x = 3;
  required int32 int y = 4;
  required int64 seqnum = 3;
}

message AUOrderLoaded {
  required int32 orderid = 1;
  required int32 truckid = 2;
  required int64 packageid = 3;
```

```

    required string description = 4;
    required int64 seqnum = 5;
}

message UAOrderDeparture {
    required int32 orderid = 1;
    required int64 packageid = 2;
    required int64 trackingnum = 3;
    required int64 seqnum = 4;
}

message UAOrderDelivered {
    required int64 packageid = 1;
    required int32 destinationx = 2;
    required int32 destinationy = 3;
    required int64 seqnum = 4;
}

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

message AUCommands {
    repeated AURequireTruck requiretruck => 1;
    repeated AUOrderLoaded orderloaded = 2;
    repeated int64 acks = 3;
    repeated AErr error = 4;
}

message AUResponses {
    repeated AURequireTruck requiretruck = 1;
    repeated AUOrderLoaded orderloaded = 2;
    optional bool finished = 3;
    repeated AErr error = 4;
    repeated int64 acks = 5;
}

message UACommands {
    repeated UATruckArrived truckarrived = 1;
    repeated UAOrderDelivered orderdelivered = 2;
    repeated int64 acks = 3;
    repeated AErr error = 4;
}

```



```
message UAResponses {  
  repeated UATruckArrived truckarrived = 1;  
  repeated UAOrderDelivered orderdelivered = 2;  
  optional bool finished = 3;  
  repeated AErr error = 4;  
  repeated int64 acks = 5;  
}
```

Source Code (Outdated)

```
syntax = "proto2";  
message AURequestTruck {  
  required int32 whnum = 1;  
  required int32 x = 2;  
  required int32 y = 3;  
  required int64 seqnum = 4;  
}  
  
message UATruckArrived {  
  required int32 truckid = 1;  
  required int32 whnum = 2;  
  required int32 int x = 3;  
  required int32 int y = 4;  
  required int64 seqnum = 5;  
}  
  
message AUOrderLoaded {  
  required int32 truckid = 1;  
  required int64 packageid = 2;  
  required int32 x = 3;  
  required int32 y = 4;  
  required int64 seqnum = 5;  
}  
  
message UAOrderDeparture {  
  required int64 packageid = 1;  
  required int64 trackingnum = 2;  
  required int64 seqnum = 3;  
}  
  
message UAOrderDelivered {  
  required int64 packageid = 1;
```

```
    required int32 x = 2;
    required int32 y = 3;
    required int64 seqnum = 4;
}

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

message AUCommands {
    repeated AURequireTruck requiretruck = 1;
    repeated AUOrderLoaded orderloaded = 2;
    repeated int64 acks = 3;
}

message AUResponses {
    repeated AURequireTruck requiretruck = 1;
    repeated AUOrderLoaded orderloaded = 2;
    optional bool finished = 3;
    repeated AErr error = 4;
    repeated int64 acks = 5;
}

message UACommands {
    repeated UATruckArrived truckarrived = 1;
    repeated UAOrderDelivered orderdelivered = 2;
    repeated int64 acks = 3;
}

message UAResponses {
    repeated UATruckArrived truckarrived = 1;
    repeated UAOrderDelivered orderdelivered = 2;
    optional bool finished = 3;
    repeated AErr error = 4;
    repeated int64 acks = 5;
}
```