```
AsyncReceive(Aid, mb, data_r, comm_r) ==
  /\ Aid \in Actors
  /\ mb \in Mailboxes
  /\ data_r \in Addr
  /\ comm_r \in Addr
  /\ pc[Aid] \in ReceiveIns
      (* If a matching "send" request exists in the mailbox mb, choose the
      oldest one and, complete the receiver's fields and set the communication
      to the "ready" state *)
  /\ \/ \exists c \in mailbox(mb):
          /\ c.status="send"
          /\ \forall d \in mailbox(mb): d.status="send" => c.id <= d.id
          /\ Communications' =
               (Communications \ {c}) \cup {[c EXCEPT
                                       !.status = "ready",
                                       !.dst = Aid,
                                       !.data_dst = data_r]}
          (* Use c's existing communication id *)
          /\ memory' = [memory EXCEPT ![Aid][comm_r] = c.id]


      (* Otherwise (i.e. no matching AsyncSend communication request exists),
        create a "receive" request and push it in the Communications. *)
     \/ /\ ~ \exists c \in mailbox(mb): c.status = "send"
        /\ LET comm ==
                 [id |-> Cardinality(Communications)+1,
                  status |-> "receive",
                  dst |-> Aid,
                  data_dst |-> data_r]
           IN
             /\ Communications' = Communications \cup {comm}
             /\ memory' = [memory EXCEPT ![Aid][comm_r] = comm.id]
  /\ \E ins \in Instr : pc' = [pc EXCEPT ![Aid] = ins]
  /\ UNCHANGED <<waitingQueue,Requests>>
```