MODULE *doris*

This is the list of action that may be enabled

$SendHardMsg(id, list)$	$\triangleq$	Task "*id*" sends a slot message with a reservation list "list".
$SendHardMsg(id)$	$\triangleq$	Task "*id*" sends a slot message (reservation message).
$SendBestMsg(id, s)$	$\triangleq$	Process "*id*" sends a best effort message of size *s*
$SendStopMsg(id, next)$	$\triangleq$	Process "*id*" sends a STOP message to reserve the token for process "next"
$UpdateCounter(id)$	$\triangleq$	The local counter $K[id]$ is incremented by 1.
$UpdateTimer(t, id)$	$\triangleq$	Reset $t[id]$ to 0
$UpdateGamma(id)$	$\triangleq$	Update the *Gamma* tuple when receiving an elementary message with a reservation list
$UpdateState(id, bool)$	$\triangleq$	Update the state variable to the value *bool*
$ReceiveHardMsg(id)$	$\triangleq$	Task *id* receives a slot message
$ReceiveBestMsg(id)$	$\triangleq$	Process *id* receives a best effort message
$ReceiveStopMsg(id)$	$\triangleq$	Process *id* receives a STOP effort message and update his local counter

```
$Timeout(id)$              $\triangleq$  $\wedge\ esTimer[id] = Delta\_es$
                                         $\wedge\ UpdateCounter(id)$
                                         $\wedge\ UpdateTimer(esTimer, id)$

$ElementaryWindow(id)$     $\triangleq$  $\wedge\ 0 \leq t[id] < d$
                                         $\wedge$ IF $K[id] = id$
                                             THEN
                                                $\wedge\ list \triangleq$ IF $state$
                                                                THEN   reservation list for the next cycle
                                                                ELSE   $\{-1\}$
                                                $\wedge\ SendHardMsg(id, list)$
                                             ELSE
                                                $\wedge\ ReceiveHardMsg(id)$
                                                $\wedge$ IF $elementarymessagearrives$
                                                    THEN
                                                       $\wedge\ UpdateGamma(id)$
                                                       $\wedge\ UpdateState(id, true)$
                                                    ELSE  $UpdateState(id, false)$

$ReservationWindow(id)$    $\triangleq$  $\wedge\ delta + d \leq t[id] < delta + 2 * d$
                                         $\wedge$ IF $K[id] = Gamma[id]$
                                             THEN $SendHardMsg(id)$
                                             ELSE $ReceiveHardMsg(id)$

$BestEffortWindow(id)$     $\triangleq$  $\wedge\ 2 * (delta + d) < t[id] < Delta\_es$
```

```
$\wedge$ IF $t[id] < 3 * delta$
   THEN IF  the $MAC$ is iddle
           THEN
             $\wedge$ $beTimer(id) = d$
             $\wedge$ $UpdateTimer(beTimer,\ id)$
             $\wedge$ $UpdateCounter(id)$
             $\wedge$ IF $K[id] = id$
                THEN IF  Process $id$ has “data” to transmit
                       THEN IF $t[id] - sizeOf(data) > 3 * d$
                              THEN $SendBestMsg(id,\ sizeOf(datc$
                              ELSE $SendStopMsg(id,\ id)$
                       ELSE  do nothing (loop)
                ELSE  do nothing (loop)
           ELSE
             $\wedge$ $UpdateTimer(beTimer,\ id)$
             $\wedge$ $ReceiveBestMsg(id)$
             $\wedge$ wait for $EOF$ interrupt
   ELSE
     IF $K[id] = id$
      THEN IF  Process $id$ has “data” to transmit
             THEN $SendStopMsg(id,\ id)$
             ELSE $SendStopMsg(id,\ id + 1)$
      ELSE
        $\wedge$ $ReceiveStopMsg(id)$
```