

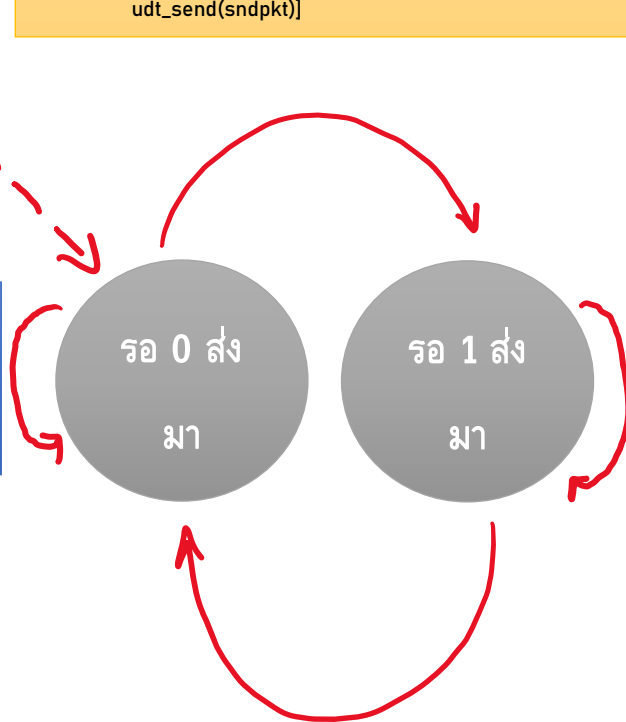
Problem # 03

1. จงออกแบบหลักการของ Reliable Data Transfer ของ Rdt. Version 3.0 ในส่วนของโฮสต์ปลายทาง (Destination Host) โดยใช้ Finite State Machine (FSM) มาโดยละเอียดให้ครอบคลุมทุกกรณี พร้อมอธิบายการทำงาน

Receiver

```
If [rdt_rcv(rcvpkt) && notcorrupt(rcvpkt) && has_seq0(rcvpkt)]  
Then    [extract(rcvpkt,data)  
         deliver_data(data)  
         sndpkt = make_pkt(ACK,0,chksum)  
         udt_send(sndpkt)]
```

```
If [rdt_rcv(rcvpkt) &&  
    (corrupt(rcvpkt) || has_seq1(rcvpkt))]  
Then    [sndpkt = make_pkt(ACK,1,chksum)  
         udt_send(sndpkt)]
```



```
If [rdt_rcv(rcvpkt) &&  
    (corrupt(rcvpkt) || has_seq0(rcvpkt))]  
Then    [sndpkt = make_pkt(ACK,0,chksum)  
         udt_send(sndpkt)]
```

```
If [rdt_rcv(rcvpkt) && notcorrupt(rcvpkt) && has_seq1(rcvpkt)]  
Then    [extract(rcvpkt,data)  
         deliver_data(data)  
         sndpkt = make_pkt(ACK,1,chksum)  
         udt_send(sndpkt)]
```

Rtd3.0 เมื่อฝั่ง Sender ส่ง Packet ไปยังฝั่ง Receiver แล้ว
นั้น ฝั่ง Receiver จะส่ง ACK กลับไปให้เพื่อยืนยันว่า Packet ได้ถูกรับ
แล้ว โดย Rtd3.0 จะต่างกับ Rtd2.0 ตรงที่ใน Rtd3.0 จะมีการเพิ่ม
Time Out เข้ามาในฝั่ง Sender

-เริ่มโดยฝั่ง Receiver รอ Packet Sequence 0 ส่งมาจาก
Sender และถ้าได้รับจริงและไม่ Corrupt จะทำการ Extract นำไปใช้
ต่อไปพร้อมส่ง ACK0 กลับไป สุดท้ายทำการรอ Packet Sequence 1

-หากได้รับ Packet มาแล้ว Corrupt หรือมี Sequence เดิม
จะส่ง ACK0 กลับไปอีกรอบ

-แต่ถ้าได้รับ Packet Sequence 1 และไม่ Corrupt จะทำการ
Extract นำไปใช้ต่อไปพร้อมส่ง ACK1 กลับไป สุดท้ายทำการรอ Packet
Sequence 0

-หากได้รับ Packet มาแล้ว Corrupt หรือมี Sequence เดิม
จะส่ง ACK1 กลับไปอีกรอบ

FSM จะวนกลับไปยังข้อแรกอีกครั้ง