Analysis and Verification of Concurrent Systems UFCFYN-15-M

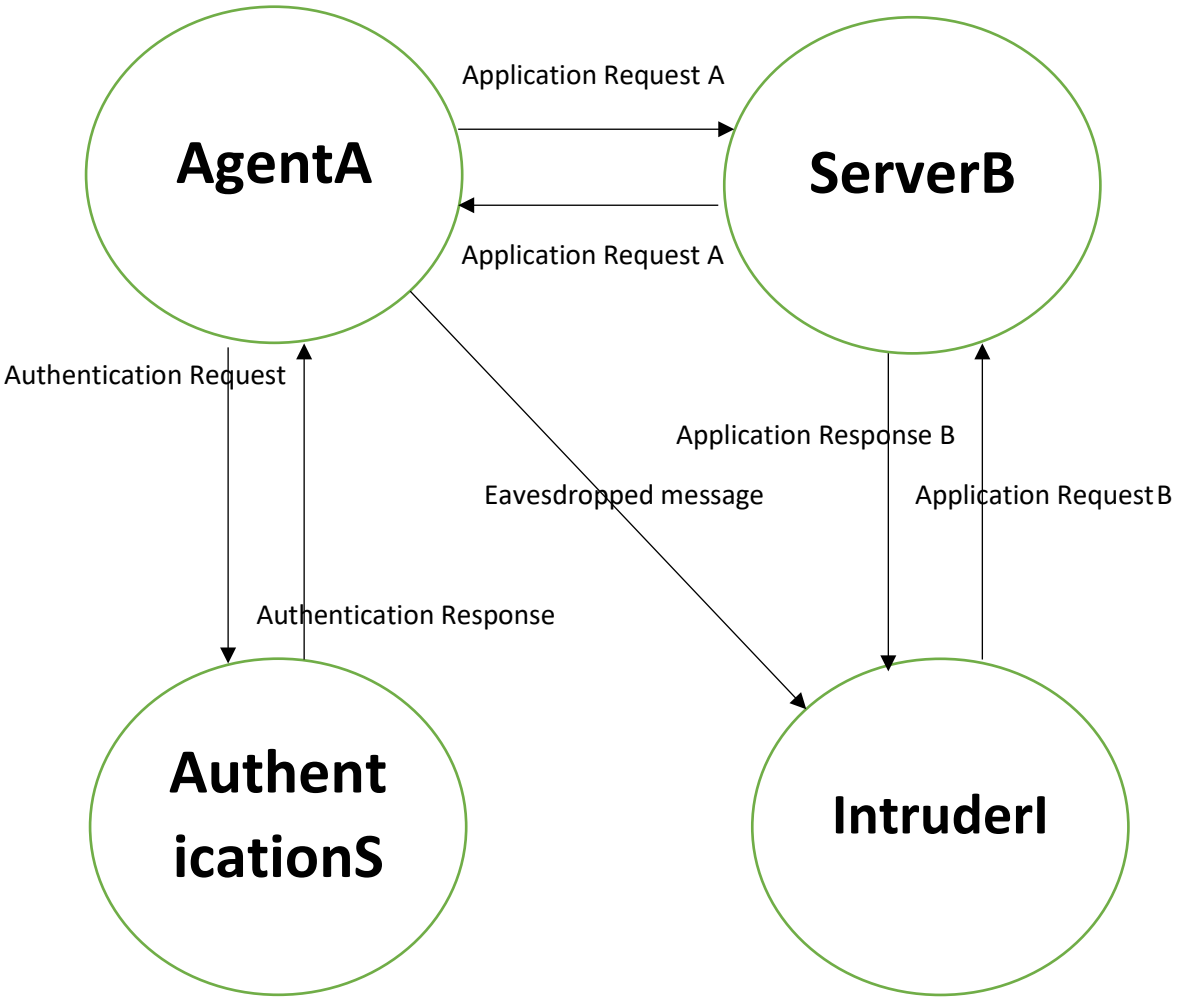
Course Work Assignment

By Michael Onileowo

Task 1:

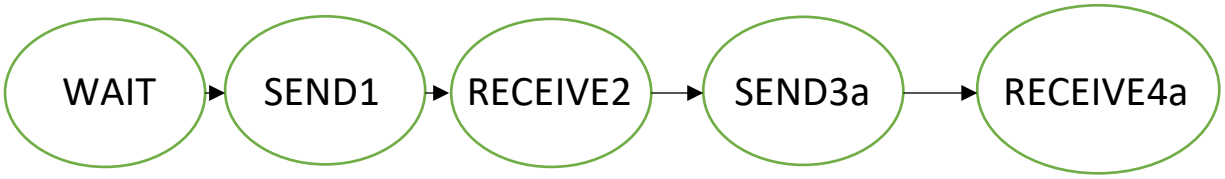
Design and draw a state transition diagram of the system considering four agents.

The state transition diagram for AgentA, ServerB, AuthenticationS and IntruderI is as below



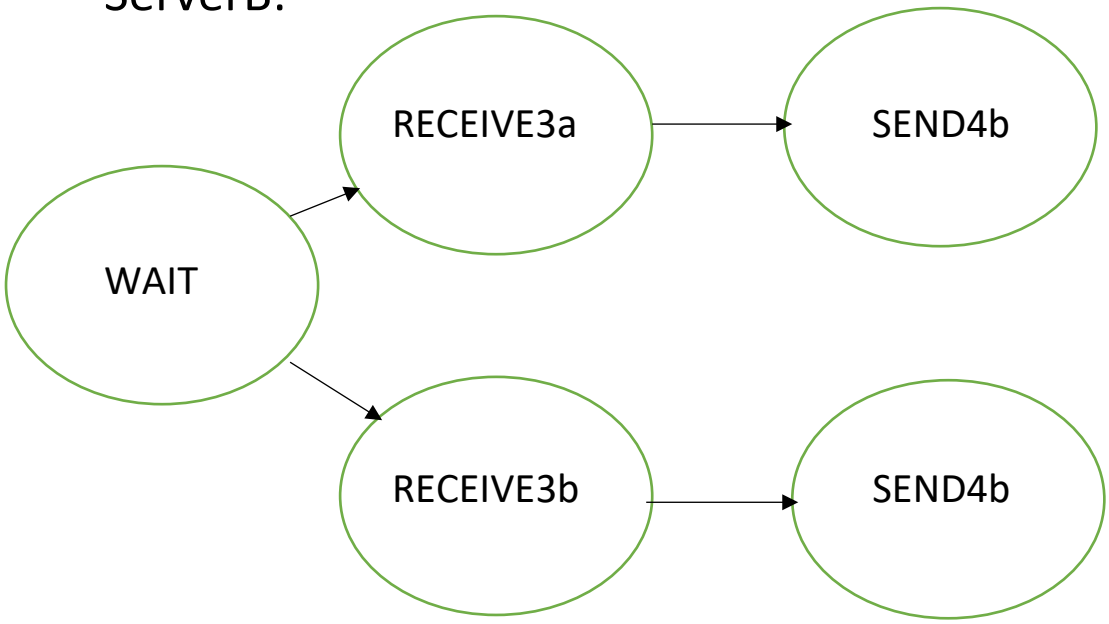
This is also represented as below to help in writing the NuSMV code

AgentA:

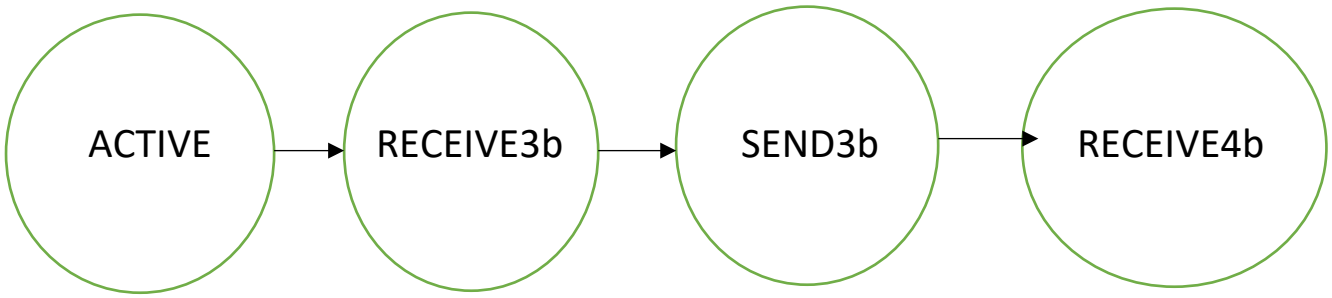


AuthenticationSB:

ServerB:



IntruderI:



Task 2:

In your NuSMV model (code using the SMV language) all the agents should work concurrently, and in an asynchronous manner.

agent.smv

Task 3 and 4:

Identify and express five authentication and secrecy properties using both LTL and CTL
and Verify all the properties identified above

1. SPEC AG!(clientA.count_A_B_s < (masterB.count_B_A_r) + (masterB.countB_B_I_r))

This is to specify that the count of messages received by ServerB can be more than the count of messages sent by AgentA because IntruderI is also able to send messages to ServerB.

```
NuSMV > check_ctlspec -p "AG!(clientA.count_A_B_s < (masterB.count_B_A_r) + (masterB.count_B_I_r))"
-- specification AG !(clientA.count_A_B_s < masterB.count_B_A_r + masterB.count_B_I_r)  is true
NuSMV >
```

2. SPEC AG!(clientA.status = receive4b)

This is to specify that Always Globally, AgentA status will not get to the status of receive4b. This is the message that has been eavesdropped by IntruderI.

```
NuSMV > check_ctlspec -p "AG!(clientA.status = receive4b)"
-- specification AG !(clientA.status = receive4b)  is true
NuSMV >
```

```
NuSMV > check_ctlspec -p "AG(clientA.status = receive4b)"
-- specification AG clientA.status = receive4b  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 4.1 <-
    clientA.status = wait
    clientA.count_A_S_s = 0
    clientA.count_A_S_r = 0
    clientA.count_A_I_s = 0
    clientA.count_A_B_s = 0
    clientA.count_A_B_r = 0
    authServerS.status = wait
    authServerS.count_S_A_s = 0
    authServerS.count_S_A_r = 0
    masterB.status = wait
    masterB.count_B_A_s = 0
    masterB.count_B_A_r = 0
    masterB.count_B_I_s = 0
    masterB.count_B_I_r = 0
    rogueI.status = idle
    rogueI.count_I_A_r = 0
    rogueI.count_I_B_s = 0
    msgM.msg = m1
NuSMV >
```

3. SPEC AG(clientA. count_A_S_s >= authServerS. count_S_A_r)

This verify that Always Globally, number of messages sent from AgentA to AuthenticationS must greater or equal to the count messages sent from AuthenticationS to AgentA.

```
NuSMV > check_ctlspec -p "AG(clientA. count_A_S_s >= authServerS. count_S_A_r)"
-- specification AG clientA.count_A_S_s >= authServerS.count_S_A_r  is true
NuSMV >
```

4. SPEC AG(clientA.count_A_B_s >= masterB.count_B_A_r)

This will verify that the count of messages sent from ServerB to AgentA will always be equal or greater than the count of the messages received by AgentA from ServerB.

```
NuSMV > check_ctlspec -p "AG(clientA.count_A_B_s >= masterB.count_B_A_r)"
-- specification AG clientA.count_A_B_s >= masterB.count_B_A_r  is true
NuSMV >
```

5. SPEC AG(rogueI.count_I_A_r = clientA.count_A_I_s)

This verifies that IntruderI cannot have more messages than the number of messages sent by AgentA.

```
NuSMV > check_ctlspec -p "AG!(rogueI.count_I_A_r = clientA.count_A_I_s)"
-- specification AG !(rogueI.count_I_A_r = clientA.count_A_I_s)  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 6.1 <-
    clientA.status = wait
    clientA.count_A_S_s = 0
    clientA.count_A_S_r = 0
    clientA.count_A_I_s = 0
    clientA.count_A_B_s = 0
    clientA.count_A_B_r = 0
    authServerS.status = wait
    authServerS.count_S_A_s = 0
    authServerS.count_S_A_r = 0
    masterB.status = wait
    masterB.count_B_A_s = 0
    masterB.count_B_A_r = 0
    masterB.count_B_I_s = 0
    masterB.count_B_I_r = 0
    rogueI.status = idle
    rogueI.count_I_A_r = 0
    rogueI.count_I_B_s = 0
    msgM.msg = m1
NuSMV >
```

6.