Application of<br>Mathematica Based Simulation Template to Demand Assigned MAC Described in IEEE P802.11-92/39 ("The IBM MAC Proposal")<br>Submitted By:<br>Larry Van Der Jagt<br>Knowledge Implementations, Inc.<br>32 Conklin Road Warwick, NY 10990<br>Voice: 914-986-3492 FAX: 914-986-6441 EMail: KIILVDJ@attmail.com<br>Main Issues Addressed:<br>29.1 How Does IEEE 802.11 Do Simulation<br>9.1 MAC throughput \& throughput probability

## 图 Commercial Introduction

This notebook contains code that has been developed by Knowledge Implementations, Inc. (KII) to support the IEEE 802.11 effort. The primary business of KII is providing contract engineering services. We have experience in many aspects of networking as well as significant experience in the area of process control. We have been involved with engineering work in the wireless networking fleld since 1987.

In order to help defer the cost of our involvement in generating this code we are requesting that individuals and organizations that obtaln this code and use it pay a shareware fee of $\$ 200.00$ to KII. As an incentive to do this we have encoded the source for the package RadioSupporti.m that contains the main analysis routines that this notebook requires to execute. KII will provide unencoded source for this package to anyone that pays the shareware fee.

Copyright © 1992 Knowledge Implementations, Inc. (KII)
KII MAKES NO REPRESENTATIONS, EXPRESSED OR IMPLIED, WITH RESPECT TO ELECTRONIC MATERIALS, INCLUDING WITHOUT LIMITATIONS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH IS EXPRESSLY DISCLAIMED. KII DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THIS ELECTRONIC MATERIAL WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE PROGRAM WILL BE UNINTERRUPTED OR ERROR FREE.

In addition to the foregoing, you should recognize that all software systems and thelr documentation contain errors and omissions. KII shall not be responsible under any circumstances for providing information on or corrections to errors and omissions discovered at any time in this Electronic Material, whether or not it is aware of the errors or omissions. KII does not recommend the use of this Electronic Material for applications in which errors or omissions could threaten life, injury or significant loss.

IN NO EVENT SHALL KII BE LIABLE FOR ANY LOST PROFITS, LOST BENEFITS, OR ANY CONSEQUENTIAL, INDIRECT, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES, WHETHER IN CONTRACT OR IN TORT, EVEN IF KII HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This Electronic Material is copyrighted by KII.
In return for payment of the shareware fee KII agrees to grant the user a non-exclusive license to use this Electronic Material for personal use only. The user may not use this Electronic Material, directly or indirectly, alone or as part of a larger system, in whole or in part, by transferring it, reselling it or licensing it to others for compensation except for the actual cost of making the physical copy and/or transfer. Any copy transferred by the user to others must be complete and unaltered.

## 图 Overview

In IEEE P802．11－92／123 a detailed computational framework was established for the execution of performance simulations of Media Access Control State machines operated over Physical Layer Entities．This framework allowed the experimenter to locate stations geographically and to assign attributes to those stations．Some of the attributes that were defined were the station Location，TxPower，State and MessageProbability．A random sample from a statistical model was used to simulate the path loss between stations．The path loss matrix that results from this sampling was used to determine the signal level at each station resulting from transmissions from other stations．Each station maintained a stationTxQueue and a stationRxqueue that tracked messages that it desired to send and that it received．The filling and emptying of these queves was performed by traffic generating functions and by a state machine model of MAC operation． In document $92 / 123$ the state machine model used was a very simple ALOHA type model．This document extends the work of $92 / 123$ by implementing state machines for a demand assigned MAC simlar to that described in Documents IEEE P802．11／92－39 and IEEE P802．11／91－74 submitted by K．S．Natarajan et al．．At this time the feature described in $92 / 39$ that allows for direct commaication between mobiles during the C period has not been implemented．Also，at this time the assumption is made that the station is registered with an access point at all times．The addition of registration and direct $c$ period communications would be possible in future work．This document also makes some improvements to the basic framework of document $92 / 123$ that make it easier to use for a wide variety of situations and state machines．

The state machines illustrated in this document continue the process of demonstrating the usefulness of this framework in testing MAC／PHY cholces and evaluating performance．In thelr current state they implement many of the functions described in the previous submittal，they do not implement many of the functions，such as sychronization that will be necessary for the final MAC／PHY design．It should be clear，however，from the functions that are performed that further enhancements would allow these other functions to be implemented and tested．As is the case of any set of state machines that are in the process of being designed，we fully expect that these machines are not perfect and that testing and optimization will reveal bugs both structural and accidental．One area in which some detail not included in the original subrittals that has been added is in the area of the scheduling algorithm and in the choice of when within the $C$ period a station will elect to attempt to gain a bandwidth reservation．Also，as it stands the amount of overhead associated with the various headers is probably larger than it needs to be and some modifications might be done in this area．As was the case with $92 / 123$ we have not made any attempts at improving the code for speed．Our opinion here is that once the scheme that needs to be thoroughly evaluated is implemented that optimization for speed will be in order．In other words，this represents a snapshot of work in progress rather than completed analysis．

## 图I』it1aIIzation

The process of executing a simulation begins with the loading of the support functions that are contained in the package RadioLAN RadioSupporti．Each of these functions will be described in the sections of the notebook that follow．The next cell loads the support packages．This package also uses the standard Mathematica packages
Statistics＇ContinuousDistributions and Statistics＇DiscreteDistributions．These standard packages also need the standard packages＂Statistics＇Normaldistribution＇＂， ＂Statistics｀DescriptiveStatistics＂＂，＂Statistics｀Common｀DistributionsCommon＂＂，and ＂Statistics＂InverseStatisticalFunctions＂＂．The loading of all of these packages is handled automaticaly by the package RadioSupportl．The next cell loads the support packages．

```
<<RadioIAN`RadioSupport1`
```

```
Off[General::spelll];
Off[General::spell];
```

In order to utilize this notebook it is necessary to initialize the fading statistics. This is accomplished using the function SetFadingParameters. This function takes three parameters that are used in calculating the path loss as a function of distance. The large scale fading is determined by using the first parameter and the second parameter as the mean and standard deviation of a normal distribution function and taking a random sample of this distribution to be the $n$ used to calculate the large scale attenuation $1 / d^{\wedge} n$ where $d$ is the distance at which the attenuation is desired. The value of $1 / d^{\wedge} n$ (expressed in $d B$ ) found as the large scale fading value is used as the mean of a second normal distribution with a standard deviation equal to the third parameter. This second distribution models the local fading characteristics. A random sample from this distribution is the value returned for the attenuation at a specific distance.

The cell that follows sets the fading parameters for this specific instance of the simulation. This.cell must be executed prior to the start of the simulation. An example of the output of the fading model is provided in the following few cells. The function GetAttenuation[d] is the actual function that is used when a sample of the statistical model established with SetFadingParameters is required for a given distance.

SetFadingParameters [3, .1,5];
This simulation allows the experimenter the flexibility of setting up the station population that will be involved in the simulation. Each station must have Location (in a three dimensional space), TxPower (in dBmw), MessageProbability (the mean of a Poission Process) and State (initial state of the station of the MAC state machine) specifled. The cell that follows defines an initialization procedure that can be used to set initial attributes for a list of station or access point names. The names used are arbitrary, however, a list of those must be placed in a list assigned to the symbol Accesspoints and Mobiles in order for the simulation to operate. Also, a station should not be named empty since this is the indicator of an empty transmit or receive queue. The initial location of the station is selected from a uniform distribution. The range of this distribution is set by the parameters $x m i n, x m a x, y m i n, y m a x ~ a n d ~ z m i n, z m a x . ~$

```
InitializeAPs[stations_]:=Map[(State[#]^=Hopping;
SynthTimer[吾]^=InitSynthTimer;
TATimer[#]^=0;
TAHTimer[#]^=0;
TBTimer[#]^=0;
TBATimer[#]^=0;
TCTimer[#]^=0;
TCHTimer[#]^=0;
InitBTimer[#]^=0;
InitCTimex[#]^=0;
NetworkID[#]^=InitNetworkID;
BroadcastFlag[#]^=False;
TxPOwer[#]^=InitTxPOwer;
MessageProbability[#]^=InitMessageProb;
StationTxQueve[#]^={empty}; (*In this simulation the AP never originates so this is always emf
StationRxQueue[#]^={empty}; (*This is the list of stations received from during B period*)
RegisteredMobiles[#\mp@subsup{|}{}{\wedge}={{\mathrm{ empty } };}
ReservationQueue[#]^={{empty}};(*All mobile TXQueves that were successful in Cperiod*)
Location[#]^={Random[Statistics'ContinuousDistributions'UniformDistribution[xmin, xmax]],
                        Random[Statistics`ContinuousDistributions' OniformDistribution[ymin,ymax]],
                        Random[Statistics`ContinuousDistributions`OniformDistribution[zmin,zmax]]}
                                    )&,stations];
```

InitializaMobiles[stations_]: =Map[ (State[\#]A=WaitingForAH;
SynthTimer[\#]^=InitSynthTimer;
TBTimer [\#] ${ }^{\wedge}=0$;
TCTimer [\#] ${ }^{\wedge}=0$;
TxPower[\#]^=InitTxPower;
RegisteredAP[\#] A=Accessi;
MessageProbability[\#]^=InitMessageProb;
StationTxQueue[\#]^=\{empty\};
StationRxQueue [\#] ${ }^{\wedge}=\{$ empty $\}$;
BEHeader[\#] $=\{$ disable, none $\}$;
IncomingFailure[\#] ${ }^{\wedge}=0$;
IncomingSuccess[\#] $\wedge=0$;
IncomingFlag[\#] ${ }^{\wedge}=0$;

Lostap [票 $]^{A}=\overline{0}$;
Improper Hop [\#]^=0;
Fallure[\#]^=0;
Success[\#] ${ }^{\wedge}=0$;
Location[\#]^=\{Random[Statistics'ContinuousDistributions'OniformDistribution[xmin, xmax]], Random[Statistics` ContinuousDistributions' DniformDistribution[ymin,ymax]], Random[Statistics'ContinuousDistributions'OniformDistribution[zmin, zmax]]) ) \&, stations];

The following cells initialize a number of parameters and executes the initialization function defined above and also builds the list of stations that is required by other processing steps. The list Transmitonstates is a list of the states in which a device has its transmitter turned on and is used by the CheckForReceive routine in order to determine signal to interference ratio. Note, that the basic unit of time for the simulation is the octettime that is taken to be eight bit times. The hop time in octets is calculated from the number of hops per second and the Octettime parameter. Messages within this simulation are considered to be of fixed length (in octets) with this length set by the symbol messagelength. The parameter minimumcPeriod is used to assure that some bandwidth is set aside for reservation requests. The parameter reservationlength is used to determine the number of octets needed to be transmitted successfully to obtain a reservation for bandwidth (during the $C$ period). The Header overhead parameters are the fixed number of octets of overhead that are assumed to be in each header. Headers that contain station lists (A Header and $B$ Header) grow in size based on the number of stations that are currently being serviced.

```
Hopspersecond=500;
Octettime=8 10^-6;
HopTime=Floor[1/(Octettime Eopspersecond)]
MinimumCPeriod=Floor[.20 HopTime]
messagelength=5;
AReaderOverhead=2;
BHeaderOverhead=2;
CHeaderOverhead=2;
InItSynthTimer=2;
reservationlength=2;
250
50
xmin=0;ymin=0;zmin=0;xmax=20;ymax=20;zmax=20;
InitTxPOWer=10;
InitNetworkID=0;
InitMessageProb=2;
SetAttributes[{Synthtimer, HTimer,TATimer,TAHTimer, TBTimer,TBHTimer,
                            TCTimer,TCHTimer},Listable];
SetAttributes[{State,TxPower,MeseageProbability,StationTxgueue,
                                    StationRxqueue,Location},Listable];
SetAttributes[{BHHeader,NetworkID, BroadcastFlag, ReservationQueue,
                RegisteredAP},LIstable];
SetAttributes[{Failure, IncomingSuccess,IncomingFailure, IncomingFlag,
                    LostAP},Listable];
SetAttributes[{ImproperHop,Success,InitBTimer,InitCTimer},Listable];
AccessPoints={Access1,Access2,Access3};
Mobiles={Station1,Station2,Station3,Station4,Station5};
TransmitOnStates={TransmittingAH,TransmitAPeriod, TransmittingBy,
                                    TransmitBPeriod, TransmitCPeriod,TransmittingCH};
Stations=Join[AccessPoints,Mobiles];
InitializeRPs[AccessPoints];
InitializeMobiles[Mobiles];
```


## ??Access1

Global`Access1
BroadcastFlag[Access1] ${ }^{\wedge}=$ False
InitBTimer[Accessi] $n=0$
InitCTimer[Accessl] $n=0$
Location [Access1] ${ }^{\wedge}=\{13.71997976100826557,10.87329007531652585,9.842681340014111611\}$
MessageProbability[Access1] $\wedge=2$
NetworkID[Accessl] ${ }^{\wedge}=0$
RegisteredMobiles [Accessi] ${ }^{\wedge}=\{\{$ empty\}\}
ReservationQueue[Accessl] $\wedge=\{\{$ empty $\}\}$
State[Accessl] $\wedge=$ Hopping
StationRxQueue [Accessi] $\wedge_{=}$\{empty\}
StationTxqueue [Accessi] $\wedge=\{$ empty\}
SynthTimer[Accessl] $\wedge=2$
TAHTimer[Accessi] $\wedge=0$
TATimer [Accessl] ${ }^{\wedge}=0$
TBHTImer[Accessi] $\wedge=0$
TBTimer [Accessi] $\wedge=0$
TCHTimer[Accessl] $\wedge=0$
TCTimer[Accessi] $\wedge=0$
TxPower[Access1] ${ }^{\wedge}=10$
Once the station parameters and the fading parameters are set the function BuildAttenuationTable can be called with a concatenated list of stations and access points as a parameter and a table of attenuations between entities will be calculated. In this calculation, an assumption is made that reciprocity applies with respect to attenuation of the channel. That is, the path loss between two entities is the same regardless of which station is transmitting and which is receiving. This assumption leads to a symmetric matrix representation for the attenuation between stations. The row and column numbers represent which stations the attenuation value applies to. For instance, the attenuation between the Accessi and Access2 in this example appears in both (row one, column two) and (row two, column one). The cell that follows illustrates the process of generating an attenuation matrix. It should also be pointed out that the attenuation numbers that are generated do not take into account the reduction in antennae aperture as a function of wavelength and hence, tend to understate the probable attenuation at distance in real applications. The impact of this effect, however, will not be important until specific PHY implementations are added to this framework. The symbol Attenuation is assigned to the calculated Attenuation Table for use by other routines, specifically, by GenerateReceivelevels, the routine that calculates the receive level at each station for transmissions originating at each other station and the overall level of receive power at a given station.

## BuildAttenuationTable[Stations]

$$
\begin{aligned}
& \left\{\begin{array}{l}
\{0,-30.9852,-45.7394,-16.3717,-31.6867,
\end{array},-39.9276,-40.685,-29.3515\right\}, \\
& \{-30.9852,0,-27.3462,-29.1486,-25.8994,-33.8268,-34.279,-21.1272\}, \\
& \{-45.7394,-27.3462,0,-30.2465,-16.0334,-13.6978,-34.2907,-35.7278\}, \\
& \{-16.3717,-29.1486,-30.2465,0,-35.2908,-35.9834,-27.5907,-38.1466\}, \\
& \{-31.6867,-25.8994,-16.0334,-35.2908,0,-4.54351,-35.7892,-35.5751\},
\end{aligned}
$$

$$
\begin{aligned}
& \{-39.9276,-33.8268,-13.6978,-35.9834,-4.54351,0,-28.1531,-46.6562\}, \\
& \{-40.685,-34.279,-34.2907,-27.5907,-35.7892,-28.1531,0,-37.822\}, \\
& \{-29.3515,-21.1272,-35.7278,-38.1466,-35.5751,-46.6562,-37.822,0\}\}
\end{aligned}
$$

Having established the parameters associated with the station configuration the next task undertaken is to begin the process of simulating a transmission environment using these stations. The function GenerateTraffic uses the MessageProbability assoclated with each station list in the first parameter to determine how many messages a particular station will generate during this time interval. A station name is selected using a uniform discrete distribution from the possible destinations stations (the second parameter list) that a particular station has available to it (Note: all stations other than itself in this model) for each message that is generated. These stations' names are placed in a list and are appended to the end of the list currently in existance as the StationTxQueue attribute of each station. This queue consists of a list of destination stations for which these messages are intended. The cells that follow illustrate the generation of the initial set of traffic. Each successive call will append traffic to the end of the StationTxQueue for each station. When operating in conjunction with state machines that take messages from the beginning of StationTxqueue, the function GenerateTraffic implements a FIFO style queue for the desired transmissions.

## GenerateTraffic[Mobiles, Mobiles]; <br> StationTxQueue[Mobiles]

$\{\{S t a t i o n 3, ~ e m p t y\}, ~\{e m p t y\},\{e m p t y\}, ~\{e m p t y\},\{S t a t i o n 2, ~ e m p t y\}\}$
The function that evaluates the instantaneous receive level at a given station is GenerateReceivelevels. This function monitors the state of all stations each time it is executed and calculates the receive power of all stations that are in any state listed in the global list Transmitonstates state at all other stations. The level of receive power at a particular station resulting from the transmission of another station is stored In the matrix ArrivingTxPowerTable by GenerateReceiveLevels. In order to avoid negative infinity problems a minimum receive power level is established for all stations. This level is set by setting the symbol MinPower. Also, in order to facilitate evaluation of how this set of stations will operate in the presence of external nolse sources the table ExternalNoiseTable of external noise values at individual stations is required. This table is an array with one entry for the external noise level at each station (note: noise in this array is expressed as power in Watts, not dBrw). The following cell sets the required parameters. The parameters CaptureMargin and retrylimit are used later in the notebook and are only initialized here for convenience. They are not used by GenerateReceivelevels.

```
MinPower=-100;
ExternalNoiseTable=10^({-100,-100, -100,-100,-100}/10).001;
CaptureMargin=12;
retxylimit=3;
```

The support functions used in the execution of the state machines have been modified to allow for more flexible operation. In particular the function CheckForReceive has become a general purpose function for evaluating signal to interference ratio and the job of determining how to adjust transmit and receive queues for acceptable and unacceptable S/I ration has been relegated to being performed in the individual state machines. This change has allowed for the elimination of the routine CheckForResponse. The CheckForReceive function now accepts three parameters, source, destination and a complete list of entities. This function returns or if current conditions would result in a signal to interference ratio in excess of the parameter CaptureMargin for signals arriving at the destination orlginating from the source. It begins by calling GenerateReceivelevels to update the ArrivingTxPowerTable. It then evaluates the sum of the arriving power from all interferers and determines if it is low enough for successful transmission to take place. If it is it returns or if not it returns Nok. The following cell illustrates the use of this function.

State[Stationl]^=TransmitCPeriod;
CheckForReceive[Station1,Access1, Stations]

Evaluating TX Conditions between Stationl and Accessl OK

## N[MatrixForm[ArrivingTxPowerTable], 2]

1. $10^{-13}$
2. $10^{-13}$
3. $10^{-13}$
4. $10^{-13}$
5. $10^{-13}$
6. $10^{-13}$
7. $10^{-13}$
8. $10^{-13}$
9. $10^{-13}$
10. $10^{-13}$
11. $10^{-13}$
12. $10^{-13}$
13. $10^{-13}$
14. $10^{-13}$
15. $10^{-13}$
16. $10^{-13}$
17. $10^{-13}$
18. $10^{-13}$
19. $10^{-13}$
20. $10^{-13}$
21. $10^{-13}$
$9.810^{-6}$
0.000041
$4.310^{-6}$
22. $10^{-13}$
$3.710^{-7}$
$3.510^{-7}$
$8.910^{-7}$
$1.610^{-6}$
23. $10^{-13}$
24. $10^{-13}$
25. $10^{-13}$
26. $10^{-13}$
27. $10^{-13}$
28. $10^{-13}$
29. $10^{-13}$
30. $10^{-13}$
31. $10^{-13}$ 1. $10^{-13}$
32. $10^{-13}$
33. $10^{-13}$
34. $10^{-13}$
35. $10^{-13}$
36. $10^{-13}$
37. $10^{-13}$

## 图 The MAC/PBY State Machines Jsed For This Simulation

Having established the support functions required, the functions that actually process the station state machines can now be presented. For the purposes of this simulation the access polnt is modeled as a simple state machine in which progress from one state to another is based solely on elapsed time. In general, this machine is very simple with most of the transitions being purely timer based. The complexity of the operation has been offloaded to the scheduler function and to the mobiles since within this simulation the mobiles generate all traffic and are responsible for determining whether message transmission is successful or not.

The state machine for the access point in this simulation assumes that any message received during the preceding $B$ period will be retransmitted during the current $A$ period. Upon exit from the state lopping, the length of the A period and the AH header is set based on the length of the StationRxqueue resulting from the previous B period activity. The state Schedule executes the scheduler and transfers the StationRxqueue to the Stationtrqueue. It also resets the recelve queue.
The function ExecuteAcceasPoint that is defined in the next cell implements this state machine for the list of access points that is provided as a parameter.

[^0]```
            Print["ReservationQueues", ReservationQueue[AccessPoints]],
TransmittingAH,
            Print[currentaccess," is TransmittingAR"];
        Which[TAHTimer[currentaccess]==0,
            State [currentaccess]^=TransmitAPeriod,
            True, TAHTimer [currentaccess] ^=TAHTimer[currentaccess]-1],
TransmitAPeriod,
    Print[currentaccess," is TransmitAPeriod"];
    Print["IncomingFlags",IncomingFlag[Mobiles]];
    Which[TATimer [currentaccess]==0,
            (State[currentaccess]^NTransmittingBH;
            TBHTimer[currentaccess]^=Floor[
            (InitBTimer[currentaccess]/messagelength)]+BHeaderOverhead;),
            True,TATimer[currentaccess]^^=TATimer[currentaccess]-1],
TransmittingBH,
    Print[currentaccess," is TransmittingBE"];
    Which[TBHTimar[currentaccess]==0,
            (State[currentaccess] A=RecelveBPeriod;
            TBTimer[currentaccess]^=InitBTimer[currentaceess];),
            True,TBHT1mer[currentaccess]^=TBHTimer[currentaccess]-1],
ReceiveBPeriod,
            Print[currentaccess," is ReceiveBPeriod"];
            Print["TXQueues",StationTxQueue[Stations]];
            Print["RXQueues",StationRxQueue[Stationa]];
            Which[TBTimer[currentaccess]=0,
                    (State[currentaccess]^=TransmittingCH;
                    TCHTimex[currentaccesa]^=CHeaderOverhead;),
                    True, TBTimer[currentaccess]^^TBTimer[currentaccess]-1],
TransmittingCH,
    Print[currentaccess," 1s TransmittingCR"];
    Priat ["TXQueues", Station'TxQueue[Stations]];
    Print["RXQueues", StationRxQueue[Stations]];
    Which[TCHTimer[currentaccess]=00,
                    (State[currentaccess]^=ReceiveCPeriod;
                    TCTImer[currentaccess]^=InitCTimer[currentaccess];),
                    True, TCHTimer[currentaccess] A=TCHTimer[currentaccess]-1],
Receivecperiod,
    (*Print[currentaccess," is ReceiveCPeriod"];*)
    Which[TCTimer[currentaccess] ==0,
                (State[currentaccess]^=Bopping;
                SynthTimer[currentaccess]^=InitSynthTimer;),
                True, TCTimer[currentaccess]^=TCTimer[currentaccess]-1]],
{n,1,Length[accesspoints]}]
```

The overall operation of the simulation is based on the concept that all traffic originates from the mobiles. A mobile's StationTXQueue is a list of all the traffic that the mobile needs to transmit. The process of making a bandwidth reservation involves the mobile verifying that, during the portion of the $C$ period in which it elects to transmit, the $S / I$ ratio at the controlling Access point is sufficient to allow the transfer to be successful. If the $S / I$ ratio is sufficient the StationTxQueue with the station's own name appended as the first element is appended to the ReservationQueue of the Access Polnt. The Scheduler function is executed by the ExecuteAccesspoint state machine at the start of the A period. The scheduler requires that some parameters be set prior to execution. These are the HopTime and the MinimumCPeriod and the header overhead parameters. These parameters along with the length of the A period (indicated by the length of the StationRXQueue at the start of the cycle) are used to determine how much bandwidth is available for the B period. The Schedulex generates the BHEeader that is used to set the TBTimers of the individual mobiles, the length of an individual message as detailed in the messagelength parameter enters into this calculation. Each mobile has its own BHHeader that consists of a list of 2 1tems. These items are the time within the $B$ period to start transmitting and the length of time to transmit. The

Scheduler also sets the value of the symbol allocatod to the total length of the $B$ period allocated. The scheduler function is defined in the following cell.

```
Scheduler[accesspoints_,mobiles_]:= DO[(Print["Scheduler running"];
    currentaccess=accesspoints[[立]];
    currentqueve=ReservationQueue[currentaccese];
    totalBPeriod=HopTime-(messagelength
    (Iength[StationRxQueue[currentaccess]]-1))-AHeaderOverhead-
                    BHeaderOverhead-CEeaderOverhead-MinimumCPeriod;
    totalrequested=messagelength
                            (Length[Flatten[currentqueue]]-Length[currentqueue]);
    numberrequested=Length[currentqueue]-1;
        Map[(BEHeader[##^^={disable,none})&,mobiles];
```

    (*If the total requested bandwidth is lest than the total available
    B Period time allocate all of the requests*)
    Which[totalrequested<=totalBPeriod,
        (allocated=0;
            Do[BHHeader[currentqueue[ [k, 1] ] \(]^{\wedge}=\{\) allocated+1, messagelength
                            (Length[currentqueue \([[k]]]-1\) ) \};
                    allocated=allocated+messagelength
                            (Length[currentqueve[ [k]]]-1),
                    \(\{k, 1\), numberrequeated \(\}\);
                    ReservationQueue[currentaccess] \({ }^{\wedge}=\{\) empty \(\}\);
                    InitBTimer [currentaccese]^=allocated+1),
    (*Allocate all available \(B\) period bandwidth among stations when there
    isn't enough*)
            True,
                    (allocated=0; mostper=Floor[totalrequested/ (messagelength
                                    numberxequested)];
            Do[Which[(messagelength (Length[currentqueue[[k]]]-1)) <=mostper,
        (*Current station wants less than the most it can have, so allocate
        it all it wants, reduce the total requirement, and calculate a new
        mostper unless this is the last station in the queve*)
            (BEHeader[currentqueue[[k, 1]] \(]^{\wedge}=\{\) allocated +1 , messagelength
                            (Length[currentqueue[[k]]]-1)];
                allocated=allocated+messagelength (Length[currentqueue[[k]]]-1);
                Reservationqueue[currentaccess]^=
                    ReplacePart[ReservationQueue[currentaccess], \{empty\},k];
                If [k<numberrequested,
                    totalrequested=totalrequested-messagelength
                            (Length[currentqueue[[k]]]-1);
                    mostper=Floor[totalrequested/(messagelength
                            (numberrequested-k))], True]),
                    (*The current station want more than it can have so allocat it
                the most that is allowed and if it is not the last station
                update total requested and mostper*)
                True, (BEHeader[currentqueue[ [k, 1] ] \(]^{\wedge}=\{\) allocated+1, messagelength
                    mostperf;
                        allocated=allocated+mostper;
                        ReservationQueue[currentaccess] \({ }^{\wedge}=\) ReplacePart[ReservationQueue[
                currentaccess], Drop[currentqueue[[k]], mostper],k];
                    If [k<numberrequested,
                    totalrequested=totalrequested-mostper;
                    mostper=Floor[totalrequested/(messagelength
                            (numberrequested-k))],True])],
                    \(\{k, 1\), numberrequested) \(\}\);
                    InitBTimer[currentaccess]^=allocated+1)];
    
# InitCTimer [currentaccess] ${ }^{\wedge}=$ HopTime- (messagelength <br> (Length[StationRxQueue [currentaccess]]-1)) <br> -allocated-AHeaderOverhead-BHeaderOverhead-CHeaderOverhead), $\{n, 1$, Length[accesspoints] $\}]$ 

The state machines for the mobile units are considerably more complex owing to the assumption that has been made that all traffic will emanate from the mobiles. This is fundamentally an assumption that we are dealing with interferring Basic Service Areas in this simulation and that there is no Distribution System.

For the purposes of this simulation, it is assumed that a method for achieving synchronization of hopping patterns is avallable. The implementation of a method for achieving this synchronization within this framework is a subject for future work. In this case a mobile waits in the WaitingforAH state upon becoming active before it starts its hop timing and other functions. When the controllingAP goes into the Transmitting AB state the hop timer is initialized and the hop is begun.

The length of the StationRxqueue at the access point at the end of the $B$ period is the length of the next A period. During the A perlod messages received during the previous $B$ period are transmitted to the destination mobiles. This is accomplished by mobiles with names in the StationTxQueue of the Access point evaluting the $S / I$ conditions for every time period in the A period and if they are acceptable incrementing the IncomingSuccess counter by the number of times its name appears in the stationTrqueue. If the $S / I$ conditions are inadequate the IncomingFailure counter is incremented. In this simulation, the service modeled is connectionless.

A mobile requests enough bandwidth to transmit its entire StationTxQueue, it drops items from this queue when it is allocated bandwidth in the $B$ period. When the StationTxQueue is empty at the start of the $C$ period the traffic generator is run to get new traffic.

During the $B$ period when a station's TBTimer times out it evaluates whether the $S / I$ ratio between the mobile and the access point is OR during the entire time that it is transmitting its allocated messages. If it is it puts its stationTxQueue the portion of its StationTxqueue that it has been authorized to transmit onto the StationRxqueue of the access point and increments the Success counter by the number of messages it was authorized to transmit. If the $S / I$ ratio is NOK at any time during the transmission nothing is place on the stationRxqueue of the access point. In both the success and failure case the items are removed from the tranmsit queue once transmitted. The Failure counter is incremented if the transmission is not successful.

If the station needs to request bandwidth during the $C$ period it selects a slot in the $C$ period at random (from a uniform distribution of the time remaining in the $c$ perlod) and during that period it evaluates whether $S / I$ conditions to the access point are acceptable. If they are it adds to the Reservationqueve as detailed in the scheduler discussion.

The function ExecuteMobile that is defined in the next cell implements this state machine for the list of mobiles that is provided as a parameter.

```
ExecuteMobile[mobiles]]:=Do[currentmobilemmobiles[[n]];
    controllingAP=隹isteredAP[mobiles[[n]]];
    currentnumber=Position[Stations,currentmobile][[1,1]];
    Switch[State[currentmobile],
    Hopping,
    (Print[currentmobile," is Hopping"];
    Which[SynthTimer[currentmobile]==0,
                            State[currentmobile]^=WaitingForAB,
                            True, SynthTimer[currentmobile]^=SynthTimer[currentmobile]-1]),
    WaitingForAB,
    (Print[currentmobile," is WaitingForAH"];
    HTimer[currentmobile]^=HopTime;
    Which[State[controllingAP]===TransmittingAHE&
                    CheckForReceive[controllingAP, currentmobile,Stations]===0R,
                    (State[currentmobile]^=ReceivingAH)]),
ReceivingAB,
    (Print[currentmobile," is ReceivingAE"];
    (*Decrement Hop Timer*)
    HTimer[currentmobile]^=HTimer[currentmobile]-1;
    (*If Hop Timer is expired Hop no matter what*)
    Which[HTimer[currentmobile]==0,
            (State[currentmobile]^=Hopping;
            ImproperHop[currentmobile]^=Improper#op[currentmobile]+1;
            SynthTimer[currentmobile]^=InitSynthTimer;),
(*If its not time to HOp check for AP transmitting AH*)
            State[controllingAP]===TransmittingAH,
            (*If AP is tranBmitting AH check S/I and if it is NOK jump to wait
            for hop and increment the lost access point counter*)
            Which[CheckForReceive[controllingAP, currentmobile, Stations]===NOR,
                (State[currentmobile]^=WaitingForHop;
                LostAP[currentmobile]^=LostAP[currentmobile]+1)],
            (*If everything is OK check to see if AP has completed AB period*)
            State[controlilngAP]===TransmitAPeriod,
            (*If it is done jump to ReceiveAPeriod State*)
            (State[currentmobile]^=ReceiveAPeriod;
            (*If we are expecting a message increment IncomingAttempts and
            set incoming flag to true*)
            IncomingFlag[currentmobile]^=Count[
                Flatten[StationTxQueve[controllingAP]],currentmobile])]),
```

ReceiveAperiod,
(Print[currentmobile," is ReceiveAPeriod"];
(*Decrement EOP Timer*)
ETimer[currentmobile]^=\#Timer[currentmobile]-1;

```
(*If Hop Timer is expired Hop no matter what*)
Which[#Timer[currentmobile]==0,
    (State[currentmobile]^=Hopping;
    ImproperHop[currentmobile]^=ImproperHop[currentmobile]+1;
    SynthTimer[currentmobile]^=InitSynthTimer;),
(*If its not time to Hop check for AP transmitting A*)
    State[controllingAP]===TransmitAPeriod,
    (*If AP is transmitting AH check S/I and lf it is NOK with an
        incoming message expected and jump to wait for hop, also increment
        the lost access point counter and incoming failure counter*)
            Which[CheckForReceive[controllingAP, currentmobile, Stations]===NOK
                &&IncomingFlag[currentmobile]!=0,
                    (State[currentmobile]^=WaitingForHop;
                IncomingFallure[currentmobile]^=IncomingFailure[currentmobile]
                                    +Incomingrlag[currentmobile];
            IncomingFlag[currentmobile]^=0;
            LostAP[currentmobile]^=LostAP[currentmobile]+1)],
        (*If everything is OK check to see if AP has completed A period*)
        State[controllingAP] ===TransmittingBH,
        (*If it is done jump to ReceiveAPeriod State*)
        (State[currentmobile]^=ReceivingBH;
        (*If we are expecting a message increment IncomingSuccess and
        set incoming flag to false*)
Which[IncomingFlag[currentmobile]!=0,
    (IncomingSuccese[currantmobile]^=IncomingSuccess[currentmobile]+
                                    IncomingFlag[currentmobile];
        IncomingFlag[currentmobile]^=0)])]},
```

ReceivingBE,
(Print[currentmobile," is ReceivingBE"]:
(*Decrement Hop Timer*)
HTimer[currentmobile]^=ATimer[currentmobile]-1;
(*If Hop Timer is expired Hop no matter what*)
Which[日Timer[currentmobile]==0,
(State [currentmobile] ${ }^{\wedge}=$ Hopping;
ImproperHop [currentmobile]^=ImproperHop[currentmobile]+1;
SynthTimer [currentmobile]^=InitSynthtimer;),
(*If its not time to Hop check for AP transmitting BH*)
State [controllingAP] ===TransmittingBH,
(*If AP is transmitting BH check $S / I$ and if it is NOK jump to wait
for hop and incremnt the lost access point counter*)
Which[CheckForReceive[controllingAp, currentmobile, Stations] $===N O K$,
(State[currentmobile] ${ }^{\wedge}=$ WaitingForBop;
LostAP[currentmobile] ${ }^{\wedge}=$ LostAP[currentmobile]+1)],
(*If everything is $O K$ check to see if AP has completed $B H$ period*)
State[controllingAP] ===ReceiveBPeriod,
(*If it is done jump to ReceiveBPeriod State*)

```
    If[NumberQ[BEHeader[currentmobile][[1]]],
        State[currentmobile]^=WaitBPeriod;
        TBTimer[currentmobile]^=BHHeader[currentmobile][[1]],
        State[currentmobile]^=WaitingForCl]]),
WaitBPeriod,
    (Print[currentmobile," is InactiveBPeriod"];
    (*Decrement Hop Timer and waiting to transmit timer*)
HTimer[currentmobile]^=HTimer[currentmobile]-1;
TBTimer[currentmobile]^=TBTimer[currentmobile]-1;
(*If Hop Timer is expired Eop no mattor what*)
Which[HTimer[currentmobile]==0,
    (State[currentmobile]^=Hopping;
    ImproperBop[currentmobile]^^=ImproperHop[currentmobile]+1;
    SynthTimer[currentmobile]^=InitSynthTimer;),
(*If its not time to Hop check for time to transmit*)
    TBTimer[currentmobile]==0,
    (*If it is done jump to TransmitBPeriod State*)
    (State[currentmobile]^=TransmitBPeriod;
    TBTimer[currentmobile]^=BHHeader[currentmobile][[2]])]),
TransmitBPeriod,
    (Print[currentmobile," is TransmitBPeriod"];
    (*Decrement EOp Timer*)
    #Timer[currentmobile]^=#Timer[currentmobile]-1;
    TBTimer[currentmobile]^=TBTimer[currentmobile]-i;
    (*If Hop Timer is expired Hop no matter what*)
Which[HTimer[currentmobile]==0,
            (State[currentmobile]^=Hopping;
            ImproperHop[currentmobile]^=ImproperHop[currentmobile]+1;
            SynthTimer[currentmobile]^=InitSynthTimer;),
(*If ita not time to Eop check for expiration of transmit timer and
    everything NOK*)
            TBTimer[currentmobile]!=0&&
            CbeckForReceive[currentmobile, controllingAP, Stations]===NOR,
                Failure[currentmobile]^=Failure[currentmobile]+
                    Floor[BHHeader[currentmobile][[2]]/messagelength];
                StationTxQueue[currentmobile]^=Drop[StationTxQueue[currentmobile],
                    Floor[BHHeader[currentmobile][[2]]/messagelength]];
            State[currentmobile]^=WaitForCH,
(*If it is expired and everything is ok increment success and
        continue*)
            TBTimer[currentmobile]==0,
        Success[currentmobile]^=Success[currentmobile] +
        Floor[BHHeader[currentmobile][[2]]/messagelength];
        StationRxQueue[controllingAP]^=AppendTo[
                            StationRxQueue[controllingAP],
                Take[StationTxQueue[currentmobile],
                    Floor[BHHeader[currentmobile][[2]]/messagelength]]];
        StationTxQueue[currentmobile]^=Drop[StationTxQueue[currentmobile],
                            Floor[BHEeader[currentmobile][[2]]/meseagelength]];
        State[currentmobile]^=WaitingForCH]),
```

```
WaitingForCB,
    (Print[currentmobile," is WaitingForCH");
    (*Decrement Hop Timer and waiting to transmit timer*)
    HTimer[currentmobile]^=⿴Timer[currentmobile]-1;
    (*If Hop Timer is expired Hop no matter what*)
    Which[日Timer[currentmobile]==0,
            (State[currantmobile]^=Hopping;
            ImproperHop[currentmobile]^=ImproperHop[currentmobile]+1;
        SynthTimer[currentmobile]^=InitSynthTimer;),
    (*If its not time to Hop check for AP tranmsitting cH & OK*)
            State[controllingAP]===TransmittingCE,
            (*If AP is transmitting AH check S/I and if it is NOK jump to wait
            for hop and increment the lost access point counter*)
            Which[CheokForReceive[controllingAP, currentmobile, Stations]===NOK,
                    (State[currentmobile]^=WaitingForHop;
                    LostAP[currentmobile]^=LOstAP[currentmobile]+1)],
            (*If everything is OK check to see if AP has completed CH period*)
            State[controllingAP]===ReceiveCPeriod,
                Stata[currentmobile]^=SetupCPeriod]),
SetupCPeriod,
    (Print[currentmobile," is SetupCPeriod"];
    (*Decrement Hop Timer and waiting to transmit timer*)
    HTimer[currentmobile]^=BTimer[currentmobile]-1;
(*If Hop Timer is expired Hop no matter what*)
    Which[HTimer[currentmobile]==0,
        (State[currentmobile]^^=Hopping;
        ImproperHop[currentmobile]^=ImproperHop[currentmobile]+1;
    SynthTimer[currentmobile]^=InitSynthTimer;),
    (*If the TXQueue goes empty generate traffic and request bandwidth*)
        StationTxQueue[currentmobile]==={empty},
        (GenerateTraffic[{currentmobile},mobilea];
        Print["Traffic Generator Executed"];
            Which[StationTxQueue[currentmobile]==={empty},
                State[currentmobile]^=WaitingForHop,
            (*There is new traffic a reservation is required, pick a slot*)
        True,
            TCTimer[currentmobile]^=Random[
            Statistics`DiscreteDistributions`DiscreteOniformDistribution[
            (Floor[HTimer[currentmobile]/meseagelength]-1) messagelength]];
            Print[TCTimer[mobiles]];
            State[currentmobile]^=WaitCPeriod]),
            True,
            State[currentmobile]^=WaitingForHop]),
WaitCPeriod,
    ( (*Print[currentmobile," is InactiveCPeriod"];*)
    (*Decrement Hop Timer and waiting to transmit timer*)
```

HT1mex[currantmobile]^=ETimer[currentmobile]-1;
TCTimer[currantmobile]^=TCTimer[currentmobile]-1;
(*If Hop Timer is axpirad Hop no matter what*)
Which[HTimex[currentmobile] $==0$,
(State[currentmobile] ${ }^{\wedge}=$ Hopping;
ImpropexHop [currentmobile] $=$ Improper Hop [currentmobile]+1;
Synthtimex[curxantmobile]^=InitSynthTimer;),
(*If its not time to Hop check for time to transmit*)
TCT1mer[currentmobile] $==0$,
(*If it is done jump to TranemitcPeriod state*)
(State[currentmobile] $=$ TranemitcPeriod; TCTimex[currentmobile]^=reservationlength)]),

Transmitcperiod,
((*Print[currantmobile," is Transmitcperiod"];*)
(*Decrement Hop Timer and waiting to transmit timar*)
ATimer[currantmobile]A=ETimer[currentmobile]-1;
TCTimer[currentmobile]^=TCTimer[currentmobile]-1;
(*If Hop Timer is expired Hop no matter what*)
Which[aTimer[currentmobile] $=0$,
(State[currentmobile]^=Hopping:
Impropertop[currentmobile]^=ImproperHop[currentmobile]+1;
SyathTimer[currentmobile]^=InitSynthTimer;),
(*If TC Timer expires try to make reservation*)
TCTimer[currentmobila]! $=0$, Which[
CheckrorRecelve[currentmobile, contirollingAp, stations] ===NOR,
(Failure[currentmobile]^=Failure[currentmobile] +
Length[StationTxQueve[currentmobile]];
StationTxQueue[currentmobile] $\wedge=$ \{empty\};
State[currentmobile]^=WaitingForlop)].
(*If everything is ok increment success and continue*)
TCTimer[currentmobile] $=0$,
(Reservationqueue [controllingAP]^=
PrependTo[ReservationQuque[controllingAP],
Prepend[Drop[StationTxQueue[currentmobile],-i], currentmobile]];
State[currentmobile]^=WaitingForlop)]),
WaitingFoxtop,
((*Print[currentmobile," is WaitingForHop"];*)
Which[日Timer[currentmobile] $==0$,
(State[currentmobile]^=Hopping;
SynthTimer[currentmobile] =InitSynthTimer;), True, HTimer[currentmobile]^=
$\{n, 1$, Length[mobiles]\}]
(2) Simulation Execution and Results

Finally, the stage is set for performing an actual simlation. The following cell executes 1000 cycles of the defined simulation and provides diagnostic information as it is executing as well as summary statistics upon completion.
maxtime $=1000$;
Do [ExecuteAccesePoint [\{Accessi\}]; ExecuteMobile[Mobiles], \{tt, 1, maxtime\}]
Output of this command detalled in Appendix
Succass [Mobiles]
Failura[Mobiles]
IncomingSuccess [Mobiles]
IncomingFailura[Mobiles]
Lostap [Mobiles]
$\{7,5,7,10,8\}$
$\{0,2,0,2,0\}$
$\{3,12,5,2,4\}$
$\{0,0,0,0,0\}$
$\{0,0,0,0,0\}$

## 图 Conclusion

This document has expanded upon the framework for evaluating MAC/PHY performance presented in $92 / 123$. This expansion has taken the form of a preliminary implementation of one of the MACs that has been presented to the MAC group. The requested action from this submittal is to close issue 29.1 and to direct the parties working on channel models, MACs and MAC/PHY throughput presentations to begin working with a common set of tools developed in Mathematica.

## Accass1 is Hopping

Stationi ts WaltingForAH
Station2 is WaitingForAH
Station3 is WaitIngForAH
Stationt le WaltingForAH
Station5 is WaitingForAH
Accessi is Hopping
Station 1 is WaitingForAH
Station2 is WaitngForAH
"tation3 is WaitingForAH
station4 is WaitingForAH
Station5 is WaitingForAH
Access 1 is Hopping
Stationt is WaitingForAH
Statlon2 is WaltingForAH
Station3 is WaitingForAH
Stations is WaitingForAH
Station5 is WaitingForAH
Access1 is Scheduling
TXQueues\{(empty\}, (empty), \{empty), \{empty), \{empty\}, (empty), \{empty\}, (empty)\}
RXQueves ( $e \mathrm{mpty}$ ), \{empty), (empty), (empty), (empty), (empty), (empty). (empty])
ReservationQueues $\{\{$ emply\}\}, \{\{empty]!, \{(empty\}!!
Scheduler running
Allocations \{\{disable, none), (disable, none), \{disable, none\}, (disable, none\}, [disable, none)]
TXQueues \{(empty), \{empty), \{empty), \{empty\}, \{empty\}, \{empty), \{empty), \{empty]\}
RXQueves ( $e m p t y$ ), (empty), (empty), (empty), (empty), (empty), \{empty), (emply)\}
ReservationQueues\{(empty), [\{empty]), [\{empty\}]\}
Station1 is WaitingForAH
Evaluating TX Conditions between Access 1 and Station
Station2 is WaitingForAH
Evaluating TX Conditions between Access1 and Station2
Stations is WaitingForAH
Evaluating TX Conditions between Access 1 and Station3 Station4 is WaithngForAH
Evaluating TX Conditions between Access 1 and Station 4
Station5 is WaitingForAH
Evaluating TX Conditions between Access1 and Station5
Access 1 is TransmittingAH
Station 1 is ReceivingAM
Evaluating TX Conditions between Access 1 and Station
Station2 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station2
Station3 is ReceivingAH
Evaluating TX Conditions between Access1 and Station3 ationt is ReceivingAH
-valuating TX Conditions between Access1 and Station4
Station5 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5
Accessi is TransmittingAH
Station is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station2 Station3 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3 Stationt is PeceivingAH
Evaluating TX Conditions between Access 1 and Station 4
Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5
Access 1 is TransmittingAH
Slationt is ReceivingAH
Station2 is ReceivingAH
Station3 is ReceivingAH
Station4 is ReceivingAH
Stations is ReceivingAH
Access! is Transmitaperiod
IncomingFlags $\{0,0,0,0,0\}$
Station1 is ReceiveA Period
Evaluating TX Conditions between Access 1 and Station1 Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions belween Access 1 and Station3
Station4 is ReceiveAPeriod
Evaluating TX Conditions beween Access 1 and Station4 Station5 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access 1 is TransmitAPeriod
IncomingFlags $\{0,0,0,0,0\}$
Station is ReceiveAPeriod
tion2 is ReceiveAPeriod
Ition3 is ReceiveA Period
Station 4 is ReceiveA Period

## Station5 is ReceiveAPeriod

Access 1 is TransmittingBH
Station1 is ReceivingBH-
Evaluating TX Conditions between Access1 and Stationi
Station2 is FecelvingBH
Evaluating TX Conditions between Accesal and Station2
Stations is ReceivingeH
Evaluating TX Conditions between Accessi and Station3 Station4 is ReceivingBH
Evaluating TX Conditions between Access1 and Station 4
Station5 is ReceivingBH
Evaluating TX Conditions between Access1 and Station5
Accessi is TransmittingBH
Station1 is RecelvingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station2
Slation3 is RecelvingBH
Evaluating TX Conditions between Access 1 and Stations Station4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4 Station5 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station5
Accoss 1 is TransmittingEH
Station1 is RecelvingBH
Station2 is ReceivingBH
Stations is ReceivingBH
Station4 is Receiving8H
Station5 is ReceivingBH
Accessi ls RecelveBPeriod
TXQueues (\{empty), \{empty), (empty), (empty), (empty\}, \{empty\}, (empty), \{empty)]

Station! is WaitingForCH
Station2 is WaitingForCH
Station3 is WaitingForCH
Stations is WaitingForCH
Station5 is WaitingForCH
Accessi is ReceiveBPeriod
TXQueues \{\{emply\}, \{empty\}, \{empty\}, \{empty\}, (empty\}, \{empty), [empty), (empty)]
RXQueues(\{empty), \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{emply\}, \{empty), (empty\})
Slation1 is WaitingForCH
Evaluating TX Conditions betwoen Access 1 and Station 1
Station2 is WaitingForCH
Evaluating TX Conditions between Access1 and Station2
Station3 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station3
Station4 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station4
Station5 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station5
Access 1 is TransmittingCH
TXQueues\{\{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}; \{empty\}, \{empty\}\}
RXQueues (empty), \{empty), \{empty\}, \{empty\}, \{empty\}, (empty). \{empty\}, \{empty)\}
Station1 is WaitingForCH
Evaluating TX Conditions between Access1 and Station 1
Station2 is WaitingForCH
Evaluating TX Conditions between Access1 and Station2
Station3 is WaitingForCH
Evaluating TX Conditions between Access1 and Station3

## Stations is WaitingForCH

Evaluating TX Conditions between Access1 and Stationd
Station5 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmittingCH
TXQueues \{(empty), \{empty\}, \{empty\}, \{empty\}, \{empty), \{empty), (empty\}, (empty\}]
RXQueues [\{empty\}, \{empty\}, \{empty), \{empty\}, \{empty\}, \{empty), \{empty), \{empty]\}
Stationt is WaitingForCH
Evaluating $T \times$ Conditions between Access 1 and Station 1
Station2 is WaitingForCH
Evaluating TX Conditions between Access1 and Station2
Station3 is WaitingForch
Evaluating TX Conditions between Access 1 and Station3
Station4 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station 4
Station5 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station5
Accessy is TransmittingCH
TXQueues ( Rempty\}, \{empty), \{empty\}, \{empty\}, \{empty\}, \{empty\}. (empty), (empty)\}
RXQueues [\{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}]
Station1 is WaitingForch
Station2 is WaitingForCH
Station3 is WaitingForCH
Station4 is WaitingForCH
Station5 is WaitingForCH

Stationl is SetupCPeriod
Tratic Generator Executed
$\{225,0,0,0,0\}$
Station2 is SetupCPerlod
Tratfic Generator Executed
$\{225,30,0,0,0\}$
Stations is SetupCPeriod
Traffic Generator Executed
$(225,30,188,0,0)$
Station4 is SetupCPeriod
Traffic Generator Executed
$\{225,30,188,66,0\}$
Station5 is SetupCPeriod
Traffic Generator Executed
[225, 30, 189, 66, 119]
Evaluating TX Conditions between Station2 and Access1
Evaluating TX Conditions between Station 4 and Access 1
Evaluating TX Conditions between Station5 and Access1
Evaluating TX Conditions between Station3 and Accessi
Evaluating TX Conditions between Stationt and Access1
Station 1 is Hopping
Station2 Is Hopping
Station3 is Hopping
Station4 is Hopping
Stations is Hopping
Stationt is Hopping
Station2 is Hopping
Stations is Hopping
Stationt is Hopping
Station5 is Hopping
Station1 is Hopping
Station2 is Hopping
Station3 is Hopping
Stations is Hopping
Station5 is Hopping
Station1 is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Stationl Is WaitingForAH
Station2 is WaitingForAH
Staton3 is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Station 1 is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Station 4 is WaitingForAH
Station5 is WaitingForAH
Station1 is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Accass 1 is Hopping
Station 1 is WaitingForAH
Station2 is WaitingForAH
Stations is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Access1 is Hopping
Station1 is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Stationt is WaitingForAH
Station5 is WaitingForAH
Access1 is Hopping
Station! is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Access 1 is Scheduling
TXQueues \{\{empty\}, \{empry), (empty), \{Station5, Station3, empty\}, \{Station3, empty\}, (Station2, empty), (Station5, Station2, Station5, Station2, Station2, Station2, empty). (Station2, Station2, Station4, empty])
RXQueues\{(empty\}, (empty\}, (empty\}, (empty\}, (empty), \{empty), \{empty\}, \{empty\}\}
ReservationQueues [ (SStation1, Station5, Station3], (Station3, Station2),
(Station5, Station2, Station2, Station4),
(Station4, Station5, Station2, Station5, Station2, Station2, Station2),
(Station2, Station3\}, empty), \{(empty\}), (|empty)]\}
Scheduler running
Allocations $\{(1,10\},\{61,5\},\{11,5\},\{31,30\},\{16,15\}\}$

TXQueves\{(empty\}, \{empty\}, \{empty), (Stations, Station3, empty\}, \{Station3, empty\}, (Station2, empty), (Station5, Station2, Station5, Station2, Station2, Station2, empty), (Station2, Stajon2, Station4, empty)]
RXQueves (\{empty), (empty), (empty\}, (empty), \{empty), (empty), (empty), \{empty)\}
ReservationOueues(\{empty\}, \{(empty)\}, \{(empty)]\}
Stationt is WaltingForAH
Evaluating TX Conditions between Access1 and Station1
Station2 is WaitingForAH
Evaluating TX Conditions between Access 1 and Station2
Stations is WaitingForAH
Evaluating TX Conditions between Access 1 and Station3
Stationt is WaitingForAH
Evaluating TX Conditions between Access1 and Station4
Stations is WaitingForAH
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmittingAH
Station1 is RoceivingAH
Evaluating TX Conditions between Accoss1 and Station1
Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2
Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3
Stationt is ReceivingAH
Evaluating TX Conditions between Access1 and Station 4
Stations is ReceivingAH
Evaluating TX Conditions between Access1 and Station5
Accesst is TransmittingAH
Stationt is ReceivingAH
Evaluating TX Conditions between Access 1 and Station1
Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3
Stationt is ReceivingAH
Evaluating TX Conditions between Access 1 and Station4
Station5 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 5
Accass1 is TransmittingAH
Stationt is ReceivingAH
Station2 is ReceivingAH
Station3 is ReceivingAH
Stationt is ReceivingAH
Stations ta ReceivingAH
Access1 is TransmitaPeriod
IncomingFlags $\{0,0,0,0,0\}$
Station! is ReceiveAPeriod
'Evaluating TX Conditions belween Access 1 and Station1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceiveAPeriod
Evaluating TX Conditions between Accesst and Station3
Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 4
Stations is ReceiveAPeriod
Evaluating $T X$ Conditions between Access 1 and Station5
Access 1 is Transmitaperiod
IncomingFlags $\{0,0,0,0,0 \mid$
Stationt is ReceiveAPeriod
Station2 is ReceiveAPeriod
Station3 is ReceiveAPeriod
Stationt is ReceiveAPeriod
Stations is ReceiveAPeriod
Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3
Stationt is ReceivingBH
Evaluating TX Conditions between Access 1 and Station 4
Stations is Receiving8H
Evaluating TX Conditions between Access1 and Station5
Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3
Station 4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4
Station5 is Receiving BH
Evaluating TX Conditions between Access1 and Station5

Accessi is TransmittingBH
Station1 is RecoivingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access 1 and Stations Station4 is ReceivingBH
aluating TX Conditions between Access 1 and Station4 ation5 is ReceivingBH
Evaluating TX Conditiona between Access 1 and Station5
Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station
Station2 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station2
Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3 Station4 ts ReceivingBH
Evaluating TX Conditions between Access1 and Station4 Station5 is ReceivingBH
Evaluating TX Conditions between Access1 and Station5 Accesse 1 is TransmittingBH
Stationt is ReceivingBH
Evaluating TX Conditions between Access 1 and Station1 Station2 is ReceivingBH
Evaluating TX Conditions belween Access1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station3 Station4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4 Station5 is ReceivingBH
Evaluating TX Conditions between Access1 and Station5 Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station 1 Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions betwoen Access1 and Station3 Station4 is PeceivingBH
Evaluating TX Conditions between Access 1 and Stationd Stations is Receiving BH
Evaluating TX Conditions between Access 1 and Station5

- eass1 is TransmittingBH
fon1 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station 1
Station2 is Receiving BH
Evaluating TX Conditions beween Accoss 1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3 Stations is ReceivingBH
Evaluating TX Conditions beween Accessy and Station4 Station5 is Receiving BH
Evaluating TX Conditions between Accoss1 and Station5 Access 1 is TransmittingBH
Stationt is ReceivingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3 Slation 4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Stationd Stations is ReceivingBH
Evaluating TX Conditions between Access 1 and Station5 Access 1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station 1 Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station3 Stationd is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4 Station5 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station5 Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access1 and Station 1 ct=+ion2 is ReceivingBH

Jating TX Conditions between Access 1 and Station2 Station3 is ReceivingBH
Evaluating TX Conditions between Access1 and Station3

## Station4 is Receivinger

Evaluating TX Conditions between Access1 and Station4
Station5 is Receiving BH
Evaluating TX Conditions batween Access1 and Station5
Accessi is TransmittingBH
Station1 is RecelvingBH
Evaluating TX Conditions between Access1 and Station1
Stationt is ReceivingBH
Evaluating TX Conditions between Accesat and Station2
Station3 Is Recelving日H
Evaluating TX Conditions between Acceasy and Station3
Station4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4 Stations is ReceivingBH
Evaluating TX Conditions between Accese1 and Station5
Accessi is TransmittingBH
Stationt is ReceivingBH
Evaluating TX Conditions between Accese1 and Station
Station2 Is ReceivingBH
Evaluating TX Conditions between Accasa1 and Station2
Station3 la RecelvingBH
Evaluating TX Conditions between Accasal and Station3
Strition4 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4
Station5 is ReceivingEH
Evaluating TX Conditions between Access1 and Station5
Accessi is TransmittingBH
Stationt is ReceivingBH
Evaluating TX Conditions between Access 1 and Station
Station2 is ReceivingBH
Evaluating TX Conditions between Access1 and Station2
Stations is RecoivingBH
Evaluating TX Conditions between Access 1 and Station3
Stationt is ReceivingBH
Evaluating TX Conditions between Access 1 and Station 4
Stations is Receiving BH
Evaluating TX Conditions bewween Access 1 and Station5
Access1 is TransmittingBH
Stationi is RecaivingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is Receiving BH
Evaluating TX Conditions between Access 1 and Station2
Stations is RecoivingBH
Evaluating TX Conditions between Access 1 and Station3
Station4 is Receiving BH
Evaluating TX Conditions between Access1 and Station4 Station5 is Receiving BH
Evaluating TX Conditions between Access1 and Station5
Access1 is TransmittingBH
Station1 is ReceivingBH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station2
Station3 is ReceivingBH
Evaluating TX Conditions between Access 1 and Stations
Stations is ReceivingBH
Evaluating TX Conditions between Access 1 and Station4
Station5 is ReceivingBH
Evaluating TX Conditions between Access 1 and Station5
Access1 is Transmitting BH
Station1 is ReceivingBH
Station2 is ReceivingBH
Station3 is ReceivingBH
Station4 is ReceivingBH
Station5 is ReceivingBH
Access1 is ReceiveBPeriod
TXQueues \{(empty], (empty), \{empty), \{Station5, Station3, empty\}, \{Station3, empty), (Station2, empty). (Station5, Station2, Station5, Station2, Station2, Station2, empty), \{Station2, Station2, Station4, empty)\}
RXQueues, (emply), \{empty), (empty\}, (empty), \{empty), (empty), \{empty), (empty]\}
Station1 is InactiveBPeriod
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station4 is InactiveBPeriod
Station5 is InactiveBPeriod
Access 1 is ReceiveBPeriod
TXQueues (|empty), \{empty), (empty). \{Station5, Station3, empty\}, \{Station3, empty), \{Station2, empty\}, (Station5, Station2, Station5, Station2, Station2, Station2, empty), (Station2, Station2, Station4, empty)]
RXQueues \{\{empty\}, \{empty). (emply\}, \{empty\}, (empty\}, \{empty\}, \{empty), \{empty]\}
Station1 is Transmite Period
Evaluating TX Conditions between Station1 and Access 1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod

Station4 is InactiveBPeriod
Station5 Is InactiveBPeriod
Access1 is Receive日Period
TXQueves \｛（empty），（empty），（empty\}, (Station5, Station3, empty), \{Station3, empty), ［Station2，empty），（Station5，Station2，Station5，Station2，Station2，Station2， empty），（Station2，Station2，Station4，empty）］
RXQueues（ $e m p t y\}$ ，（empty），\｛empty\}, \{empty\}, \{empty\}, (empty\}, \{empty\}, \{empty])
Station1 is TransmitBPeriod
Evaluating TX Conditions betwoen Stationt and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Stationt is InactiveBPeriod
Station5 is InactiveBPeriod
Access1 is ReceiveBPeriod
TXQueues \｛（emply），\｛empty），［empty\}, [Station5, Station3, empty), \{Station3, empty), ［Station2，empty］，（Station5，Station2，Station5，Station2，Station2，Station2， emply），（Station2，Station2，Station4，empty）］
RXQuaues；（empty），（emply\}, (empty), \{empty), \{empty\}, (empty), \{empty\}, (empty)\}
Station1 is TransmitBPeriod
Evaluating TX Conditions between Station1 and Accass1
Statlon2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station4 is Inactive日Period
Station5 is InactiveBPeriod
Access1 is RecelveBPeriod
TXQueves\｛（empty\}, (emply), \{empty), \{Station5, Station3, empty\}, \{Station3, empty), （Station2，empty），（Station5，Station2，Station5，Station2，Station2，Station2， empty），\｛Station2，Station2，Station4，emply）］
RXQueves（\｛empty），（empty），（empty\}, (empty), (empty), (empty), (empty), (empty)]
Station1 is TransmitBPeriod
Evaluating TX Conditions between Station 1 and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station4 is InactiveBPeriod
Station5 is InactiveBPeriod
Access1 is RecelveBPeriod
TXQueves \｛（empty），（empty），\｛empty］，［Station5，Station3，empty］，［Station3，empty）， ［Station2，empty\}. \{Station5, Station2, Station5, Station2, Station2, Station2, emply），（Station2，Station2，Station4，empty｜］
RXQueues（ $e$ empty），［empty），（empty），（empty），（empty］，（empty），（empty），［empty］）
Station1 is TransmitBPeriod
Evaluating TX Conditions between Station1 and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Stationt is InactiveBPeriod
Station5 is InactiveBPeriod
Access1 is ReceiveBPeriod
TXQuewes（｜empty），\｛empty），\｛empty），（Slation5，Station3，empty），（Station3，empty）， \｛Station2，emply），（Station5，Station2，Station5，Station2，Station2，Station2， empty\}. \{Station2, Station2, Station4, empty)\}
RXQueues｜（empty），（empty），\｛empty），\｛empty），\｛empty），\｛empty），（empty），（empty）］
Station1 is TransmitBPeriod
Evaluating TX Conditions between Station1 and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station 4 is Inactive BPeriod
Station5 is InactiveBPeriod
Access1 is ReceiveBPeriod
TXQueves \｛\｛empty\}, \{empty), \{empty), \{Station5, Station3, emply), \{Station3, empty\},
（Station2，empty），\｛Station5，Station2，Station5，Station2，Station2，Station2，
empty\}, \{Station2, Station2, Station4, empty]\}
RXQueues（ $e m p t y$ ），（empty），\｛empty），（empty），（empty\}, \{empty\}, (empty\}, \{empty)\}
Station 1 is TransmifBPeriod
Evaluating TX Conditions between Station1 and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station4 is InactiveBPeriod
Station5 is InactiveBPeriod
Access 1 is ReceiveBPeriod
TXQueues \｛（empty），（empty\}, (empty], [Station5, Station3, empty), \{Station3, empty\}, \｛Station2，empty\}, |Station5, Station2, Station5, Station2, Station2, Station2, emply），［Station2，Station2，Station4，empty）］
RXQueves（\｛empty），（empty\}, \{empty\}, \{empty), \{empty\}, \{empty\}, \{empty\}, \{empty]\}
Station1 is TransmitBPeriod
Evaluating TX Conditions between Station1 and Access1
Station2 is InactiveBPeriod
Station3 is InactiveBPeriod
Station4 is InactiveBPeriod
Station5 is InactiveBPeriod
Access 1 is ReceiveBPeriod
TXQueues（\｛empty），（empty），保pty），\｛Station5，Station3，empty），（Station3，empty\}, ［Station2，empty），｜Slation5，Station2，Station5，Station2，Station2，Station2， empty），\｛Station2，Station2，Station4，empty］\}
RXQueues\｛（empty\}, \{empty), \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}, \{empty\}\}

## Station1 is TransmitBPeriod

Evalueting TX Conditions between Station1 and Accoas1
Station is InectiveBPeriod
Stations is InactiveBPeriod
Stationt is InactiveBPeriod
Station＇s in InactiveBPeriod
Accessi is ReceiveBPeriod
TXQuaves \｛（empty\}, \{empty), \{empty), \{Station5, Station3, emply), (Station3, empty), （Stalion2，empty），（Station5，Station2，Station5，Station2，Station2，Station2，
emply），（Station2，Station2，Station4，empty）］
RXQueces \｛empty），\｛emply），（empty\}, (empty), (empty), (empty), (empty), (empty)]
Stationt is TransmitePeriod
Statione is InactiveBPeriod
Stations is InactiveBPeriod
Stationd is InactiveBPeriod
Stations is Inactive日Period
Access1 is ReceiveBPeriod
TXQueves\｛（empty），（empty），（empty），\｛empty），（Station3，empty），（Station2，empty）， ［Station5，Station2，Statlon5，Station2，Station2，Station2，empty\},
（Staion2，Station2，Station4，empty］）
RXQueves［\｛empty，［Station5，Station3］\}, \{empty), \{empty], \{empty], (emply), (empty), ［empty］，（empty］）
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Stations is TransmitBPerioo
Evaluating TX Conditions between Station3 and Access1
Stationt is InactiveBPeriod
Stations is InactiveBPeriod
Access1 is ReceiveBPeriod
TXQueves（（empty），\｛empty），\｛empty），\｛empty），（Station3，emply），（Station2，empty），
（Station5，Station2，Station5，Station2，Station2，Station2，empty），
［Station2，Station2，Station4，emply）］
RXQueves \｛（empty，（Station5，Station3\}), (empty], \{empty), \{empty), (empty\}, (empty), （emply），\｛emply］\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is TransmitBPeriod
Evaluating TX Conditions beween Station3 and Access1
Stationt is InactiveBPeriod
Stations is Inactive日Period
Accesst is Recoive BPeriod
TXQueuss／\｛empty），（emply\}, \{empty\}, \{empty\}, (Station3, empty), \{Station2, emply\},
［Station5，Station2，Station5，Station2，Station2，Station2，emply），
［Station2，Station2，Station4，empty）］
RXQueves（\｛empty，［Station5，Station3］\}, \{empty), \{empty], (empty), \{empty), \{empty'
\｛empty）．（emptyl）－－
Stationt is WaitingForCH
Station2 is Inactive日Period
Station3 is TransmitBPeriod
Evaluating TX Conditions between Station3 and Access1
Stationd is InactiveBPeriod
Station5 is inactiveBPeriod
Access1 is ReceiveBPeriod
TXQueues \｛（empty\}, \{empty\}, \{emply\}, \{empty\}, \{Station3, empty\}, \{Station2, empty\}, （Station5，Station2，Station5，Station2，Station2，Station2，empty）， ［Station2，Station2，Station4，empty）］
RXQueves（\｛empty，\｛Station5，Station3\}), \{empty\}, \{empty\}, \{empty), \{empty], (empty\}, （emply），（empty）］
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is TransmitBPeriod
Evaluating TX Conditions between Station3 and Access1
Station4 is InactiveBPeriod
Station5 is InactiveBPeriod
Access1 is ReceiveBPeriod
TXQueues（\｛empty），［empry），\｛empty\}, (empty\}, [Station3, empty], \{Station2, empty\},
（Station5，Station2，Station5，Station2，Station2，Station2，empty），
［Station2，Station2，Station4，empty］\}
RXQueves［（empty，\｛Station5，Station3］\}, \{empty), \{empty), \{empty), (empty), \{empty), \｛emptyl．［empty］］
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is TransmitBPeriod
Station4 is inactiveBPeriod
Station5 is InactiveBPeriod
Access1 is ReceiveBPeriod

（Station5，Station2，Station5，Station2，Station2，Station2，empty），
［Station2，Station2，Station4，empry］）
RXQueves（｜empty，\｛Station5，Station3\}, \{Station2\}\}, \{empty\}, \{empty\}, \{empty\},
（empty），
\｛emply），（empty），（empty）］
Station1 is WaitingForCH
Station2 is InactiveBPeriod

## Station3 is WaitingForCH

Stationt is inactivesPeriod
Station5 is TransmitsPerlod
Evaluating TX Conditions batween Station5 and Access1
Accass Il ReceiveBPeriod
TXQueves (fempty), (empty), (empty), (empty), (Station3, empty), (empty), [Station5, Station2, Statlon5, Station2, Station2, Station2, empty]. [Station2, Station2, Station4, empty]]
nxoueves,(empty, (Station5, Station3), (Station2)], (empty), (empty), (empty). IPy),
(empty), (empty), (empty)]
Station1 is WaitingForCH
Station2 is InactiveBPariod
Stations is WatingForch
Stationt is InactiveBPeriod
Station5 Is TransmitBPeriod
Evaluating TX Conditions between Station5 and Accoss1
Accoss 1 is ReceiveBPeriod
TXOveues (fempty), \{empty), \{empty), \{empty), (Stations, empty), (empty),
(Station5, Station2, Station5, Station2, Station2, Station2, empry),
[Station2, Station2, Station4, empty)]
RXOUeues(femply, \{Station5, Station3), (Station2)], \{empty), (empty), (empty), (empty),
(emply), \{empty), (emptyl)
Station1 is WaitingForCH
Station2 is inactiveBPeriod
Station3 is WaltingForCH
Stationt is InactiveBPeriod
Station5 is TransmitBPerlod
Evaluating TX Conditions between Station5 and Access1
Access 1 is ReceiveBPeriod
TXQueues (lemply), (empty), (empty), (empty), (Station3, empty), \{empty),
[Station5, Station2, Stations, Station2, Station2, Station2, emply),
(Station2, Station2, Station4, empty)]
RXOUeues)(empty, (Station5, Station3), (Station2)], (empty), \{empty), (emply), (empty).
[empty), (empty), (empty)]
Station 1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is InactiveBPeriod
Stavons is TransmitsPeriod
Evaluating TX Conditions between Station5 and Access 1
Access 1 is ReceiveBPeriod
TXQueues(femply), (empty), \{empry), (emply), (Station3, empty), (empty), itation5, Station2, Station5, Station2, Station2, Station2, emply),
\{Station2, Station2, Station4, empty)\}
RXCuoues ((empty, (Station5, Station3), (Station2)], (empty), (empty), (empty),
(empty).
(empty), (empty), (emptyll
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForch
Station4 is InactiveBPeriod
Stations is TransmitBPeriod
Evaluating TX Conditions between Station5 and Access1
Accoss il ReceiveBPeriod
TXOueues (|empty), (ermpty), (emply), (empty), (Station3, empty), (empty),
(Station5, Station2, Station5, Station2, Station2, Station2, emply),
[Station2, Station2, Station4, empty])
RXQueues (|empry, [Station5, Station3), (Station2|), (empty), (empty), (emply),
(empty).
[empty], (empty), (empty]\}
Station is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is InactiveBPeriod
Station5 is Transmite Period
Evaluating TX Conditions between Station5 and Access 1
Access1 is ReceiveBPeriod
TXQueues [(empty), (empty), (empty), (empty), \{Station3, empty), (empty),
[Station5, Station2, Station5, Station2, Station2, Station2, empty).
(Slation2, Station2, Station4, empry)]
RXOueues(|empty, (Station5, Station3), (Station2)], (empty), (empry), (empty),
(empty).
(empry\}, (empty), \{empty)\}
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Siation4 is InactiveBPeriod
Station5 is TransmitBPeriod
yating TX Conditions between Station5 and Access1
ss1 is ReceiveBPeriod
TXOUeves ( (empty), (empty), (empty), (empty), (Station3, empty), (empty).
[Station5, Station2, Station5, Station2, Station2, Station2, empty],
[Station2, Station2, Station4, empty|]
RXCueves (\{empty, (Station5, Stations), (Station2)], (empty), (emply), (empty),
(empty).
(empty). \{emply), (empty)]
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForch
Stationt is InactiveBPeriod
Station5 is TranamitBPeriod
Evaluating TX Conditions between Slation5 and Access1
Access 1 is Receive BPeriod
TXOueves ([empty), \{empty), \{emply\}, \{empty), \{Station3, empty), (empty), (Stations, Station2, Station5, Station2, Station2, Station2, emply). (Station2, Station2, Station4, empty])
RXQueves (|empty, (Station5, Station3), (Station2]), (empty), (empty), (empty).
(emply).
[empty), (empty), (empty)]
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForch
Stationt is InactiveBPeriod
Station5 is TransmitBPeriod
Evaluating TX Conditions between Station5 and Access 1
Accass1 is RecoiveBPerico
TXOueues(|empty), (empty), (empty), (empty), \{Station3, empty), (empty), [Station5, Station2, Station5, Station2, Station2, Station2, emply).
[Station2, Station2, Station4, emptyl]
RXQueues (\{empty, (Stations, Station3), (Station2|), (empty), fempty), (empty),
(empty).
(empty), (empty), |emptyl)
Stationt is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Stations is InactiveBPeriod
Station5 is TransmitBPeriod
Evaluating TX Conditions between Station5 and Access 1
Access1 is Receive日Period
TXOueuss ([empty), (emply), (empry), \{emply), (Station3, empty), (empty), [Station5, Station2, Station5, Station2, Station2, Station2, empty]. [Slation2, Station2, Station4, empry|]
RXQueues ( (empty, [Station5, Station3), \{Station2)\}, (empty), (empty), (emply),
(empty).
[empty), (empty), \{empty)\}
Stationt is WaitingForch
: Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is InactiveBPeriod
Station5 is TransmitBPeriod
Evaluating TX Conditions between Station5 and Accoss 1
Access 1 is ReceiveBPeriod
TXQueues ((empty), (empty), (empty), (empty), (Station3, emply), (emply), \{Station5, Station2, Station5, Station2, Station2, Station2, empty), [Station2, Station2, stationd, empry])
RXOueues(empty, (Station5, Station3), [Station2)], (empty), (emply), \{empty), (emply), (empty), (empty), (empty)]
Station is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Stationd is InactiveBPeriod
Stations is TransmitBPeriod
Evaluating TX Conditions between Station5 and Access1
Access1 is ReceiveBPeriod
TXOueues ((emply), (empty), (empty), (empty), (Station3, empty), (empty), (Station5, Station2, Station5, Station2, Station2, Station2, empty), [Station2, Station2, Station4, empty])
RXQueves (empty, (Station5, Station3), (Station2)\}, (empty), (empty), (empty), (empty).
(empty), (empty), (emptyl)
Station is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is InactiveBPeriod
Station5 is TransmitBPeriod
Evaluating TX Conditions between Stations and Access1
Access 1 is ReceiveBPeriod
TXQueues \{(empty), (empty), (empry), (empty), (Station3, empty), (empty),
[Station5, Station2, Station5, Station2, Station2, Station2, empty),
[Station2, Station2, Station4, empry)]
RXQueues (\{empry, \{Station5, Station3), \{Station2|\}, \{empty), \{empty\}, \{empty\}, (empry),
(empty), \{empty), (empry)\}
Station1 is Waitingforch

Station2 is InactiveBPerlod
Stations is WaltingForCH
Stationt la InactiveBPeriod
Station5 is TransmitBPerlod
Evaluating TX Conditions between Station5 and Access1
Access1 is ReceiveBPerlod
TXOUeves (|empty), (empty), \{empty), \{empty), [Station3, empty), \{empty\}. (Station5, Station2, Station5, Station2, Station2, Station2, empty), [Station2, Station2, Station4, empty]\}
RXQueves \{(empty, (Slation5, Station3], [Station2]\}, \{empty), (empty), (empty), \{empty),
(empty\}, \{emply), (empty)\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForCH
Stationd is InactiveBPeriod
Station5 is TransmitBPeriod
Access 1 is ReceiveBPeriod
TXQueves (\{empty), \{empty), \{empty), \{empty\}, \{Station3, empty), \{empty\},
(Station5, Station2, Station5, Station2, Station2, Station2, empty), (emply)]
RXQueves (|empty, \{Station5, Station3), (Station2), (Station2, Station2, Station4\}),
\{empty), (empty), \{empty\}, (empty), \{empty\}, \{empty\}, (empty)\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForCH
Station4 is TranamitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Accessi is ReceiveBPeriod
TXQueves \{\{empty\}, \{empty\}, \{empty\}, \{empty), \{Station3, empty\}, \{empty\},
(Station5, Station2, Station5, Station2, Station2, Station2, emply), \{empty])
RXQueves \{(emply, (Station5, Station3\}, (Station2), (Station2, Station2, Station4\}),
(empty), (empty), (empty), (empty), \{empty\}, (empty), \{empty\}]
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForch
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station 4 and Access 1
Station5 is WaitingForCH
Accass1 is ReceiveBPeriod
TXQueves \{(empty\}, (empty), (empty), \{empty). (Station3, empty), \{empty\},
\{Station5, Station2, Station5, Station2, Station2, Station2, empty), \{empty]\}
RXQueves \{ (empty, \{Station5, Station3\}, \{Station2\}, \{Station2, Station2, Station4\}\}, (empty), (empty), (empty), (empty), \{empty), \{empty), (empty]\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Access1 is ReceiveBPeriod
TXQueues (\{empty), \{empty), (empty), (empty), (Station3, empty), \{emply),
\{Station5, Station2, Station5, Station2, Station2, Station2, empty), (empty)\}
RXQueves \{\{emply, \{Station5, Station3\}, \{Station2\}, \{Station2, Station2, Station4\}),
(empty), (empty), (empty), (empty), \{empty\}, (empty), \{empty\}\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions betwoen Station4 and Access 1
Station5 is WaitingForCH
Access 1 is RecaiveBPeriod
TXQueves \{\{empty\}, (empty\}, (empty), \{empty), \{Station3, empty\}, (empty\}, \{Station5, Station2, Station5, Station2, Station2, Station2, empty), \{empty]\}
RXQueves \{\{empty, (Station5, Station3), \{Station2), (Station2, Station2, Station4)\}, (empty), (empty), (empty), (empty), \{empty\}, \{empty), (empty]\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForch
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Accessi
Station5 is WaitingForch
Access 1 is ReceiveBPeriod
TXQueves \{(empty), (empty), \{empty\}, \{empty), (Station3, empty), \{empty),
[Station5, Station2, Station5, Station2, Station2, Station2, emply), (empty]]
RXQueues \{(empty, \{Station5, Station3), [Station2], [Station2, Station2, Station4]\},
(empty), (empty), (empty\}, \{empty\}, (empty), (empty), (empty])
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Stations is WaitingForCH
Access1 is ReceiveBPeriod

TXQueues ffempty\}, \{empty\}, \{empty\}, \{empty\}, \{Station3, empty\}, \{empty\},
(Stalion5, Station2, Station5, Station2, Station2, Station2, empty), (empty)]
RXQueues \{ (emply, (Station5, Station3\}, \{Station2\}, \{Station2, Station2, Station4\}\},
(empty). (empty), \{empty), (empty), (empty), (empty), (empty])
Stationt is WaitingForCH
Stationd is InactiveBPeriod
Stations in WaitingForCH
StationA is TransmitaPeriod
Evaluating TX Conditions between Stationd and Access1
Stations in WaitingForCH
Accesst is ReceiveBPeriod
TXOueues (\{empty), \{empty], \{empty\}, \{empty\}, [Station3, empty\}, \{empty\},
[Station5, Station2, Station5, Station2, Station2, Station2, empty), (empty)]
RXQueues(\{emply, [Station5, Station3), (Station2\}, (Station2, Station2, Station4]),
(empty), (empty), (empty), \{empty), (empty), (empty), (empty])
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForCH
Station is TransmitBPeriod
Evaluatine TX Conditions between Station4 and Accoss1
Station5 it WaitingForCH
Access1 is ReceiveBPeriod
TXQueves \{(emply\}, \{empty\}, \{empty\}, \{empty\}, \{Station3, empty\}, \{empty),
[Stations, Station2, Station5, Station2, Station2, Station2, empty\}, (empty])
RXOueues(fempty, (Station5, Station3), (Station2), (Station2, Station2, Station4)],
(empty), \{empty), (empty\}, (empty). \{empty), (emply), \{empty)\}
Stations in WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForCH
Stationd is TransmitBPeriod
Evaluating TX Conditions between Stationd and Access 1
Station5 is WaitingForCH
Access1 is ReceiveBPeriod
TXQueves (|empty), (empty), (empty), \{empty], (Station3, empty), \{empty),
(Station5, Station2, Station5, Station2, Station2, Station2, emply), (emply)\}
RXOueles;(empty, [Station5, Station3\}, [Station2), \{Station2, Station2, Station4]\},
(emply), \{empty), (empty), (empty), (empty), \{empty), (empty)\}
Station1 is WaitingForch
Station2 is InactiveBPeriod
Station3 in WaitingForCH
Stationd is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access 1
Station5 ie WaitingForch
Access1 is ReceiveBPeriod
TXQueves \{(empty), (emply), \{empty\}, \{empty), \{Station3, empty), (empty),
[Stations, Station2, Station5, Station2, Station2, Station2, empty), (emply)\}
RXOueves|\{emply, (Station5, Station3\}, [Station2\}, [Station2, Station2, Station4]],
[empty), (empty], (empty), (empty\}, (empty), \{empty), (empty])
Stationt is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Accessi is ReceiveBPeriod
TXQueues ([emply), \{empty), \{empty), \{empty), \{Station3, empty], (empty\},
[Station5, Station2, Station5, Station2, Station2, Station2, empty], \{empty]\}
RXOueves [\{empty, [Station5, Station3\}, (Station2], \{Station2, Station2, Station4]\},
\{empty). (empty\}, \{empty), (empty), \{empty\}, (empty), (empty\})
Slation1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Access1 is ReceiveBPeriod
TXQueves \{\{empty), \{empty\}, \{empty\}, \{empty\}, \{Station3, empty\}, \{empty),
(Station5, Station2, Station5, Station2, Station2, Station2, empty), (empty)]
RXQueues (|empty, \{Station5, Station3), (Station2), (Station2, Station2, Station4)|,
(empty). (empty), (empty), (empty), (empty), \{empty), (empty)\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Access 1 is ReceiveBPeriod
TXQueues \{ empty\}, \{empty\}, \{empty\}, (empty), \{Station3, empty], (empty),
[Station5, Station2, Station5, Station2, Station2, Station2, empty\}, (empty)
RXOueles \{(empty, [Station5, Station3), [Station2], \{Station2, Station2, Station4]\},
\{empty). (empty), \{empty), (empty\}, \{empty), (empty], [empty]\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForCH

## Stationt is TransmitBPerioc

Evaluating TX Conditions berween Stationt and Accoss1
Station5 is WaitingForch
Access1 is RecoiveBPertod
TXQueves（（empty），（empty），\｛empty），（empty），\｛Station3，empty），（empty），
［Station5，Station2，Station5，Station2，Station2，Station2，empty），（empty）］
RXQueues（［emply，（Station5，Station3），［Station2），［Station2，Station2，Station4）］，
（empty）．（empty），（empty），（empty），（empty），\｛empty），（empty）\}
tion1 is WaitingForch
tion2 is IractiveBPerlod
Station3 is WaitingForCH
Stationt is TransmitePeriod
Evaluating TX Conditions between Station 4 and Access1
Station5 is WaitngForCh
Access1 is ReceiveBPeriod
TXQueuss（emply），（emply），（emply），（empty），（Station3，empty），\｛empty\},
（Station5，Station2，Station5，Station2，Station2，Station2，emply），（empty）］
RXQueues（［empty，［Station5，Station3），\｛Station2），［Station2，Station2，Station4）］， （empty）．（empty），（empty），（empty），（empty），（empty），\｛empty）］
Stationt is WaitingForch
Station2 is ：nactiveBPeriod
Stations is WaitingForCH
Station4 is TransmitePPeriod
Evaluating TX Conditions between Station4 and Access 1
Station5 is WaitingForCH
Access1 is ReceiveBPeriod
TXQueuss（｜emply），（empty），（empty），（empyly，［Station3，empty），（empty），
［Station5，Station2，Station5，Station2，Station2，Station2，empty）．（empty）］
RXOUeves（｜empty，（Station5，Station3），IStation2），（Station2，Station2，Station4）］， （emply），（empty），（empry），（empty），（empty），（empty），（empiy）］
Stationt is WaitingForch
Station2 is InactiveBPeriod
Stations is WaitingForch
Stationt is TransmitsPeriod
Evaluating TX Conditions between Station4 and Access1
Stations is WaitingForCH
Access 1 is ReceiveBPeriod
TXQueues（（empty），（empty），（empty），（empty），（Station3，empty），（empty），
（Station5，Station2，Station5，Station2，Station2，Station2，empty），\｛empty）］
RXQueves（｜empry，（Station5，Station3），（Station2），（Station2，Station2，Station4）］， （empty）．（empty），（ernpty），（empty），（empty），（empty），（empty）］
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Stations is WaitingForch
－Yont is Transmitesperiod
．uating TX Conditions between Station4 and Access 1
Stations is WaitingForCH
Access 1 is ReceiveBPeriod
TXQueues \｛（emply），（empty），（empty），（empty），\｛Station 3 ，empty），（ernpty）， （Station5，Station2，Station5，Station2，Station2，Station2，empty），（empry）］
RXQueves（［empry，（Station5，Station3），（Station2），［Station2，Station2，Station4）］， （empty），（empty），（empty），（empty），（empty），（empty），\｛empty）］
Stationt is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForch
Slationt is TransmitBPeriod
Evaluating TX Conditions between Station4 and Accass1
Stations is WaitingForCH
Access1 is ReceiveBPeriod
TXQueues（｜empty），（empty），（emply），\｛empty），\｛Station3，empty），（empty］， ［Station5，Station2，Station5，Station2，Station2，Station2，empry），（empty）］
RXQueves（（empty，\｛Station5，Station3），（Station2），（Station2，Station2，Station4］）， （empty），\｛empty），（empty），（empty），（empty），（empty），\｛empty）\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station 4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Stations is WaitingForCH
Accesst is ReceiveBPeriod
TXQueues（［empty），（empty），（empty），（empty），［Station3，empty），\｛empty］，
［Station5，Station2，Station5，Station2，Station2，Station2，empy），（empty）］
RXQueues（（empty，\｛Station5，Station3），（Station2），\｛Station2，Station2，Station4）］，
（empty），（empty），（empty），（empty），（empty），\｛empty），（empty）］
Station is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitsPeriod
Evaluating TX Conditions between Station 4 and Access 1
Stations is WaitingForCH
A－ass 1 is ReceiveBPeriod
，Ueues \｛（empty］，（empty），\｛empty），\｛empty\}, \{Siation3, empty\}, (empty\}, \｛station5，Station2，Station5，Station2，Station2，Station2，empry），（empty）｜
RXQueues（\｛emply，［Station5，Slation3\}, \{Station2\}, [Station2, Station2, Station4]\},
（empty），（empty），（emply），（emply），（empty），（empty），（empty）］
Stationt ls WaltingForch
Station2 is InactiveBPeriod
Station3 is WaltingForCH
Stationt is TranamitBPeriod
Evaluating TX Conditions between Station4 and Accoss1
Station5 is WaitingForCH
Access1 is RecaiverPeriod
TXQueves（\｛empty），（empty），（empty），（empty），（Stations，emply），（empty），
（Station5，Station2，Station5，Station2，Station2，Station2，empty），（emply）］
RXQueles（\｛empty，（Station5，Station3），（Station2），［Station2，Station2，Station4）］，
［empty），（empty），（emply），（empty），（empty），（emply），（emply）］
Stationl is WaitngForCH
Station2 is inactiveBPeriod
Station3 is WaitingForch
Station 4 is TransmitBPeriod
Evaluating TX Conditions betwoen Station4 and Accoss1
Station5 is WaltingForCH
Accoss 1 is ReceiveBPeriod
TXQueues（iempty），（empty），（empty），\｛empty），\｛Station3，empty），（empty），
［Station5，Station2，Station5，Station2，Station2，Station2，emply），（empty）］
RXQueues（［empty，（Station5，Staion3），（Station2），［Sation2，Station2，Station4］）， （empty），（emply），\｛empty），（empty），（empty），（empty），（empty）］
Stationt is WaitngForch
Station 2 is InactiveBPeriod
Stations is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Accosal
Stations is WaitingForch
Access 1 is ReceiveBPeriod
TXQueues（｜empty），（empty），（empty），｜empty），｜Station3，empty），｜empty），
［Slation5，Station2，Station5，Station2，Station2，Station2，empty），（empty）］
RXQueues（｜emply，IStation5，Station3），［Station2），\｛Station2，Station2，Station4\}], \｛empty），\｛empty），（empty），（empty）．（empty），（empty），（empty）］
Stationt is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Stationt is TransmitBPeriod
Evaluating TX Conditions berween Station4 and Accoss1
Station5 is WaltingForCH
Access 1 is Receive日Period
TXQueves（｜empty），（empty），（emply）．（empty），\｛Station3，empty），（empty），
［Station5，Station2，Station5，Station2，Station2，Station2，empty），（emply）］
RXQueues（［empty，\｛Station5，Station3\}, (Station2), (Station2, Station2, Stationt)], （empty），（empty），（empty）．（empty），（empty），（emply），（emptyl）
．Stationl is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitePeriod
Evaluating TX Conditions between Slation4 and Access1
Station5 is WaltingForCH
Access1 is ReceiveBPeriod
TXQueues（\｛empty），（empty），（empty），\｛empty），\｛Stations，empty），\｛empty）， ［Station5，Station2，Station5，Station2．Station2，Station2，empty），（emply）］
RXQueues［｜empty，［Station5，Station3\}, \{Station2\}, [Station2, Station2, Station4)\},
（empty），（empty），\｛emply），（empty），（empty），（emply），（empty）\}
Station1 is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access1
Station5 is WaitingForCH
Accoss 1 is RecriveBPeriod
TXQueues（iemply），（empty），（empty），（empty），（Station3，empty），（emply）． ［Station5，Station2，Station5，Station2，Station2，Station2，empty］，（empty）］
RXQueues（｜empty，（Station5，Slation3），（Station2），（Station2，Station2，Slation4）］，
（empty），（empty），（empty），（empty），（empty），（empty），（empty）］
Stationt is WaitingForch
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Evaluating TX Conditions between Station4 and Access 1
Station5 is WaitingForCH
Accoss1 is Receive日Period
TXQueles（iempry），（empty），（empty），（empty），（Station3，empty），（empty），
［Station5，Station2，Station5，Station2，Station2，Station2，empty］，（empry］）
RXQueues（［empty，（Station5，Station3），（Station2），｜Station2，Station2，Station4］），
\｛empry）．（empty），（empty），（empty），（empty），（empty），（empty）］
Stationt is WaitingForCH
Station2 is Inactive日Period
Station3 is WaitingForCH
Station4 is TransmitePeriod
Evaluating TX Conditions between Station4 and Access1
Stations is WaitingForCH

Accessi is ReceiveBPeriod
TXQueves (\{empty), \{empty), \{empty), \{empty\}, \{Station3, empty), \{empty\},
[Station5, Station2, Station5, Station2, Station2, Station2, empty\}, \{empty)\}
RXQueues ((empty, (Station5, Station3), (Station2), (Station2, Station2, Station4]), \{empty), \{empty\}, (empty\}, (empty), \{empty\}, \{empty), \{empty\}\}
Station1 is WaitingForCH
Station2 is InactiveBPeriod
Station3 is WaitingForCH
Station4 is TransmitBPeriod
Stations is WaitingForCH
Access1 is ReceiveBPeriod
TXQueues \{(empty), (emply), \{empty), (empty), (Station3, empty), (empty), (empty), \{empty])
RXQueues (\{empty, (Station5, Station3), (Station2), (Station2, Station2, Station4), (Station5, Station2, Station5, Station2, Station2, Station2)], (empty), (emply), (empty), (empty), \{empty\}, \{empty\}, \{empty)\}
Station1 is WaitingForCH
Station2 is TransmitBPeriod
Evaluating TX Conditions between Station2 and Access1
Station3 is WaitingForCH
Station4 is WaltingForCH
Stations is WaitingForch
Accesst is Receive日Period
TXQueues (\{empty), \{empty), (empty), \{empty), (Station3, empty), (empty), (empty), [empty]]
RXQueves \{(empty, (Station5, Station3), \{Station2\}, [Station2, Station2, Station4], [Station5, Station2, Station5, Station2, Station2, Station2)], (empry), \{empty\}, (empty), \{empty), (empty), (empty\}, \{empty)\}
Station1 is WaitingForCH
Station2 is TransmitBPeriod
Evaluating TX Conditions between Station2 and Access1
Station3 is WaitingForCH
Station4 is WaitingForCH
Station5 is WaitingForCH
Access 1 is Receive BPeriod
TXQueues\{(empty\}, \{empty\}, \{empty), \{empty). \{Station3, empty\}, (empty). (empty), [empty])
RXQueves \{(empty, (Station5, Station3), (Station2), (Station2, Station2, Station4], \{Station5, Station2, Station5, Station2, Station2, Station2]\}, \{empty], (empty), (empty), (empty), \{empty), (empty\}, (empty)\}
Station1 is WaitingForCH
Station2 is TransmitBPeriod
Evaluating TX Conditions between Station2 and Access1
Station3 is WaitingForCH
Station4 is WaitingForCH
Station5 is WaitingForCH
Access 1 is ReceiveBPeriod
TXQueues (\{empty), \{empty], \{empty\}, \{empty\}, (Station3, empty), \{empty), \{empty\}, [empty]\}
RXQueves \{\{empty, \{Station5, Station3\}, \{Station2\}, \{Station2, Station2, Station4\}, (Station5, Station2, Station5, Station2, Station2, Station2)\}, (emply), (empty), (empty\}, (empty), (empty), (empty), (empty\}\}
Station1 is WaitingForCH
Station2 is TransmitBPeriod
Evaluating TX Conditions between Station2 and Access1
Station3 is WaitingForch
Station4 is WaitingForCH
Station5 is WaitingForCH
Accessi is ReceiveBPeriod
TXQueues\{\{emply\}, \{empty\}, (empty\}, \{empty\}, \{Station3, empty\}, (empty\}, \{empty\}, [empty])
RXQueues \{(empty, [Station5, Station3), \{Station2], \{Station2, Station2, Station4\}, (Station5, Station2, Station5, Station2, Station2, Station2]), \{empry), \{empty), \{empty\}, (empty), (empty\}, \{empty), \{empty\}\}
Stationt is WaitingForCH
Station2 is TransmitBPeriod
Station3 is WaitingForCH
Station4 is WaitingForCH
Station5 is WaitingForCH
Access1 is ReceiveBPeriod
TXQueves\{(empty\}, \{empty), (empty\}, \{empty\}, \{empty\}, (empty\}, (empty), (empty)\}
RXQueues(\{emply, (Station5, Station3), \{Station2), (Station2, Station2, Station4\},
[Stations, Station2, Station5, Station2, Station2, Station2], [Station3]], \{empry],
(empty), (empty], \{empty), (empty\}, (empty), (empty)]
S:ation1 is WaitingForCH
Evaluating TX Conditions between Access1 and Station 1
Station2 is WaitingForCH
Evaluating TX Conditions beween Access 1 and Station2
Station3 is WaitingForCH
Evaluating TX Conditions beween Access1 and Station3
Siation4 is WaitingForCH
Evaluating TX Conditions between Access1 and Station4
S:ation5 is WaitingForCH
Evaluating TX Conditions between Access 1 and Station5

## Access1 is TransmittingCH

TXQueves |(empty), \{empty\}, (empty), \{empty\}, \{empty\}, \{empty\}, \{empty), \{empty\}\}
RXQueves (\{empty, (Station5, Station3), [Station2), (Station2, Station2, Station4],
\{Station5, Station2, Station5, Station2, Station2, Station2), (Station3)], (empty), (empty), (empty), (empty), (empty), (empty), (empty)]
Stations is WaitingForCH
Evalueting TX Conditions betwoen Access 1 and Station1
Stations is WaitingForCH
Evalueting TX Conditions between Access1 and Station2
Stations is WaitingForCH
Evaluating TX Conditions between Access1 and Stations
Stationt is WaitingForCH
Evaluating TX Conditions between Access1 and Station4
Stations is WaitingForCH
Evalueting TX Conditions between Access1 and Station5
Access1 is TransmittingCH
TXQueces ((empty), \{empty\}, |empty), (empty), \{empty), \{empty\}, \{empty\}, \{empty\})
RXCueues (\{empty, \{Station5, Station3\}, (Station2\}, \{Station2, Station2, Station4),
[Stalion5, Station2, Station5, Station2, Station2, Station2\}, (Station3)], \{empty\},
(emply). (empty), (empty), (empty), (empty), (empty])

## Stationt is WaitingForCH

Evalueting TX Conditions between Access1 and Station1
Stationz is WaitingForch
Evaluating TX Conditions between Access1 and Station2
Stutions is WaltingForch
Evaluafing TX Conditions between Access 1 and Station3
Stationd is WaitingForCH
Evaluating TX Conditions between Access1 and Station4
Stations is WaitingForch
Evaluating TX Conditions between Access 1 and Station5
Accossi is TransmittingCH
TXCueves (|empty), (empty), \{empty), (emply\}, \{empty\}, \{empty\}, (empty), \{empty)\}
RXQueues(\{emply, [Station5, Station3), [Station2], \{Station2, Station2, Station4\},
(Station5, Station2, Station5, Station2, Station2, Station2), (Station3)\}, (empty),
(emplyt, \{empty), (empty\}, (empty), \{empty\}, (empty)]
Stationt is WaitingForch
Station2 is WaitingForch
Stations is WaitingForch
Station is WaitingForCH
Stations is WaitingForch
Stationt is SetupCPeriod
Tralfic Generator Executed
(95, 0, 0, 0, 0)
Station2 is SetupCPeriod
Tratfic Generator Executed
$\{95,74,0,0,0\}$
Station3 is SetupCPeriod
Traffic Generator Executed
(95, 74, 133, 0, 0)
Station is SetupCPeriod
Traffic Generator Executed
\{ $95,74,133,59,0\}$
Station5 is SetupCPeriod
Trafic Generator Executed
$\{95,74,133,59,124\}$
Evalualing TX Conditions between Station4 and Access1
Evaluating TX Conditions between Station2 and Access1
Evaluating TX Conditions between Station1 and Access 1
Evaluating TX Conditions between Station5 and Access 1
Evaluating TX Conditions between Station3 and Access 1
Stationt is Hopping
Station2 is Hopping
Station3 is Hopping
Stationt is Hopping
Stations is Hopping
Stationt is Hopping
Station2 is Hopping
Station3 is Hopping
Stationt is Hopping
Station5 is Hopping
Station1 is Hopping
Station2 is Hopping
Station3 is Hopping
Station4 is Hopping
Station5 is Hopping
Station 1 is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Station4 is WaitingForAH
Station5 is WaitingForAH
Station! is WaitingForAH
Station2 is WaitingForAH
Station3 is WaitingForAH
Stations is WaitingForAH

Stations is WaitingForAH Station is WaitingForAH Station2 is WaitingForAH Stations is WaitingForAH Stations is WaitingForAH Stations is WaitingForAH Stationt is WaitingForAH
Station2 is WaitingForAH
r ins is WaitingForAH
in 4 is WaitingForAH
Stations is WaitingForAH Stationt is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station 4 is WaltingForAH Stations is WalingForAH Stationt is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Stationt is WaitingForAH Station5 is WaitingForAH Station 1 is WaitingForAH Station2 is WaltingForAH
Station3 is WaltingForAH Station4 is WaitingForAH Station5 is WaitingForAH Station 1 is WaltingForAH Station2 is WaitingForAH Stations is WaitingForAH Stationt is WaitingForAH Station5 is WaitingForAH Station 1 is WaltingForAH Station2 is WaitingForAH Station3 is WaitingForAH Stationd is WaitingForAH Stations is WaitingForAH Stationt is WaitingForAH Station2 is WaitingForAH Stations is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Slation is WaitingForAH Stationt is WaitingForAH Statlon3 is WaitingForAH Stron is WaitingForAH
S $\quad 25$ is WaitingForAH Struon 1 is WaitingForAH Station2 is WaitingForAH Stations is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Station1 is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Station1 is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Stationt is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station4 is WaitingForAH Stations is WaitingForAH Station1 is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Station1 is WaitingForAH Station2 is WaitingForAH Station3 is WaitingForAH Station4 is WaitingForAH Station5 is WaitingForAH Access 1 is Hopping Station1 is WaitingForAH Station2 is WaitingForAH Stations is WaitingForAH Statian 1 is WaitingForAH
Sta is WaitingForAH Acctisil is Hopping Stationy is WaitingForAH

Station2 is WaltingForAH
Stations is WaitingForAH
Stations is WaitingForAH
Station5 is WaitingForAH
Accessi is Hopping
Stations is WaltingForAH
Station2 is WaltingForAH
Stations is WaltingForAH
Stationt Is WaitingForAH
Stations is WaltingForAH
Access1 is Scheduling
TXQueues ([empty), (emply), (empty), (Station4, Station2, Stations, empty, (Stations, Station5, emply), (Station2, Station1. Station2, emply),
[Station2, Station2, Stationi, Station3, emply], (Station1, empty)]
RXQueues \{\{empty, \{Station5, Station3\}, \{Station2\}, \{Station2, Station2, Station4\}, [Station5, Station2, Station5, Station2, Station2, Station2), \{Station3]), (empty),
(empty), (empty), (empty\}, \{empty\}, (empty), \{empty]\}
ReservationQueues (|IStation3, istration2, Station1, Station2], [Station5, Stationt],
(Station1, Station4, Station2, Station3), (Station2, Station3, Station5),
(Station4, Station2, Station2, Station1, Station3), empty), ([emply|\}, \{(empty)]\}
Schoduler running
Allocations $\{(21,15),(36,10\},\{1,15\},\{46,20),\{16,5\}\}$
TXOueles (\{empty, \{Station5, Station3], [Station2\}, [Station2, Station2, Station4], (Station5, Station2, Station5, Station2, Station2, Station2), (Station3)], (empty), (empty), (Station4, Station2, Station3, emply), (Station3, Station5, empty), [Station2, Station1, Station2, emply),
(Station2, Station2, Station1, Stations, empty), (Station1, empty)]
FXXQueues\{(empty), (empty), (empty), (empty), \{empty), (empty], \{empty), (empty]\}

Station1 is WaitingForAH
Evaluating TX Conditions between Access1 and Station 1
Station2 is WaitingForAH
Evatuating IX Conditions between Accass 1 and Station2
Stations is WaitingForAH
Evaluating TX Conditions between Access1 and Station3
Stationt is WaitingForAH
Evaluating TX Conditions between Access1 and Station 4
Station5 is WaitingForAH
Evaluating TX Conditions between Access1 and Statton5
Accessi is TransmittingAH
Stationt is ReceivingAH
Evaluating TX Conditions between Access1 and Station1
Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2
Stations is ReceivingAH
'Evaluating TX Conditions between Access1 and Station3 Stationt is ReceivingAH
Evaluating TX Conditions between Access 1 and Station4
Station5 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5
Accesst is TransmittingAH
Station1 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 1 Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2 Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3 Station4 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station4 Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5 Access 1 is TransmittingAH
Station1 is ReceivingAH
Evaluating TX Conditions between Access1 and Station1 Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2 Slations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3 Stationt is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 4 Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5 Access 1 is TransmittingAH
Station1 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 1 Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2 Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station3 Station4 is ReceivingAH
Evaluating TX Conditions beween Access1 and Station4 Station5 is ReceivingAH
Evaluating TX Conditions between Access1 and Station5 Access1 is TransmittingAH

Station1 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceivingAH
Evaluating TX Conditions between Access1 and Station3 Station4 is ReceivingAH
Evaluating TX Conditions between Access1 and Station4 Station5 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmittingAH
Station1 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station1 Station2 is ReceivingAH
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceivingAH
Evaluating TX Conditions between Access1 and Station3 Stations is ReceivingAH
Evaluating TX Conditions between Access 1 and Station 4 Station5 is ReceivingAH
Evaluating $T \times$ Conditions between Access1 and Station5
Access1 is TransmittingAH
Station 1 is ReceivingAH
Evaluating IX Conditions belween Access 1 and Station1
Siation2 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station2
Station3 is ReceivingAH
Evaluating TX Conditions between Access1 and Station3 Station4 is ReceivingAH
Evaluating TX Conditions between Access 1 and Station4 Station5 is ReceivingAH
Evaluating TX Conditions between Access1 and Station5
Access1 is TransmittingAH
Station is ReceivingAH
Station2 is ReceivingAH
Station3 is ReceivingAH
Station4 is ReceivingAH
Station5 is ReceivingAH
Access1 is TransmitAPeriod
IncomingFlags $\{0,7,2,1,3\}$
Station1 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station3 Station4 is ReceiveA Period
Evaluating TX Conditions between Access 1 and Station4
Station5 is ReceiveA Period
Evaluating TX Conditions between Access 1 and Station5
Access1 is Transmita Period
IncomingFlags ( $0,7,2,1,3$ )
Station 1 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station3 Stationd is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station4 Station5 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access 1 is TransmitAPeriod
IncomingFlags $(0,7,2,1,3)$
Station1 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 1 Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Slation2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station3 Station4 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station4 Station5 is ReceiveAPeriod
Evaluating TX Conditions belween Access1 and Station5 Access1 is TransmitaPeriod
IncomingFiags (0, 7, 2, 1, 3)
Station 1 is ReceiveA Period
Evaluating TX Conditions between Access 1 and Station1 Station2 is Receiveaperiod
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station3
Station4 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 4

Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access1 is Transmitaperiod
IncomingFlags $(0,7,2,1,3)$
Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station3
Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Stationd
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station5
Accessi is TranemitaPeriod
IncomingFlags $\{0,7,2,1,3\}$
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station2
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station3 Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Stationd
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access 1 is TransmitAPeriod
IncorningFlags $\{0,7,2,1,3\}$
Stationt is RecaiveA Period
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceiveAPeriod
Evaluating TX Conditions belween Access1 and Station2
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station3
Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station4
Station5 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmitaPeriod
IncorningFlags $\{0,7,2,1,3\}$
Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station3
Stationt is RecoiveAPeriod
Evaluating TX Conditions between Access1 and Station4 Station5 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Accoss1 is TransmitAPeriod
IncomingFlags $\{0,7,2,1,3$ )
Stations is ReceiveAPeriod
Evaluating TX Conditions between Aocess 1 and Station 1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station2 Station3 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station3 Stationt is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station4
Stations is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmitAPeriod
IncomingFlags $(0,7,2,1,3)$
Station 1 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2 Slation3 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station3
Station4 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 4
Station5 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station5
Access1 is TransmitaPeriod
IncomingFlags $0,7,2,1,3 \mid$
Station1 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station 1
Station2 is ReceiveAPeriod
Evaluating TX Conditions between Access1 and Station2
Station3 is ReceiveA Period
Evaluating TX Conditions between Access 1 and Station3
Station4 is ReceiveAPeriod
Evaluating TX Conditions between Access 1 and Station4


[^0]:    ExecuteAccessPoint[accesspoints_]:=Do[currentaccess=accesspoints[[n]]; Switch[State[currentaccess],

    Eopping,
    Print[currentaccess," is Hopping"];
    Which[SynthTimar[currentaccess] $=0$,
    (State[currentaccess]^=Schedule;
    TAHTimer [currentaccess] ${ }^{\wedge}=$ Length $[$
    StationRxQueue [currentaccess]]-1+AHeaderoverhead;
    TATimer[currentaccess] ${ }^{\wedge}=($ messagelength (Length [ StationRxqueue [currentaccese]]-1)) +1 ; ),

    True,
    SynthTimer[currentaccess]^=SynthTimer[currentaccess]-1],
    Schedule, Print[currentaccess," is Scheduling"];
    Print ["TXQueues", StationTxQueue[Stations]];
    Print["RXQueves", StationRxQueve[Stations]];
    Print["ReservationQueues", ReservationQueue[AccessPoints]];
    Scheduler[\{currentaccess), Mobiles];
    State[currentaccess]^=TransmittingAH;
    StationTxQueue[currentaccess] $\wedge$ =StationRxqueue [currentaccess];
    StationRxQueue [currentaccess] ${ }^{\wedge}=\{$ empty $\}$;
    Print["Allocations", BHHeader[Mobiles]];
    Print ["TXQueues", StationTxQueue[Stations]];
    Print["RXQueues", StationRxQueue[Stations]];

