NWS PROJECT REPORT

Title and Names and roll nos of group members

Pallavi Saxena	CS21MTECH11020
Satvik Padhiyar	CS21MTECH14011
Pradhumn Kanase	CS21MTECH11018

Problem statement

Implement the Modified-Largest Weighted Delay First (M-LWDF) scheduler in the 5G NR module of NS-3. Compare its performance with that of RR, PF, MT

Project description (Summary of the work done)

Add illustrations to better convey your point. Acknowledge source(s) if you are using any figures from other sources

1. Understanding how to integrate the self built MLWDF scheduler

   We found out that we had to place our files in the model folder in the nr folder.We had to also add the newly made mlwdf scheduler files in the wscript of the model folder.

2. Understanding how various ns3 mac files work in collaboration with each other

   We learnt about the various files like ns3-mac-scheduler.cc, ns3-scheduler-ue-info.cc,ns3-tdma-scheduler.cc,ns3-tdma-scheduler-pf.cc etc.We discovered how the different functionalities and parameters were getting inherited from file to file.

3. Finding the desired equation for MLWDF scheduler

   We found an LTE implementation of the MLWDF scheduler. We discovered the equation used for MLWDF scheduler from there.

4. Finding the header libraries for our parameters and how to extract those values

The parameters required for our scheduler were drop probability,target delay,HOL delay,available rate and average rate.

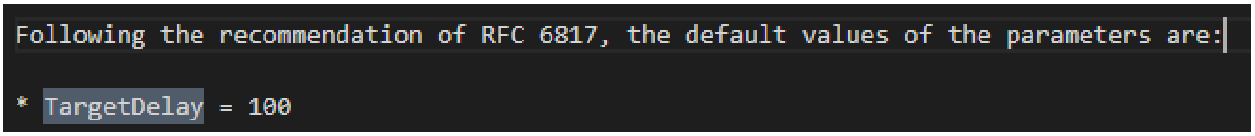
1. Drop Probability

This parameter was present in "pie-queue-disc.h" library.It indicates the chances of a packet to be dropped due to the underlying channel conditions.The exact parameter that contained this value was m_dropprob.It was calculated using “CalculateP()” function which used to update m_dropprob value using variables like channel quality , delay etc.

References:https://www.nsnam.org/doxygen/classns3_1_1_pie_queue_disc.html#a532858948ab061df8daceaa7c798814d

2. Target Delay

After extensive research we got it’s value from the LTE module.We discover it was a constant that was set to the value 100ms. The significance of this parameter it to set a bar for a packet’s waiting time.



```
Following the recommendation of RFC 6817, the default values of the parameters are:

* TargetDelay = 100
```

References:https://github.com/Asthonak/VANET-and-Nodal-Communication-in-a-Adaptive-Environment/blob/dd50b98ee9f5fe82d4b4327d218d632c19c564cf/ns-3-win/src/lte/model/bearer-qos-parameters.cc

3. HOL Delay

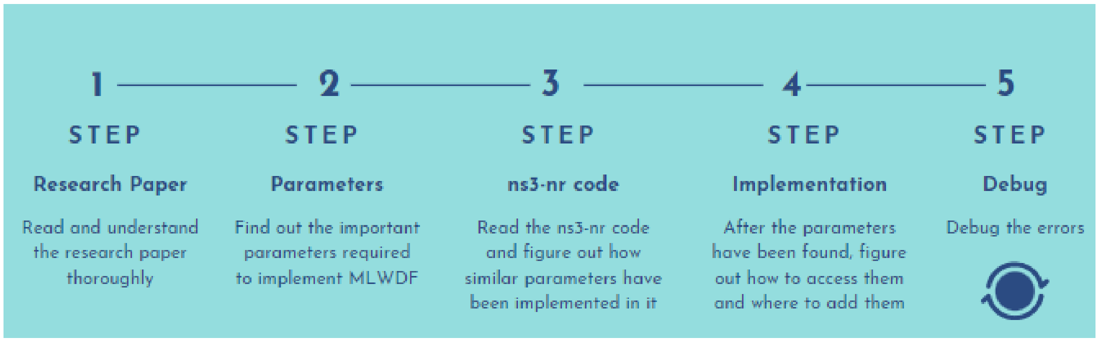
HOL(Head of Line) Delay is calculated by the RLC layer for all the bearers and is passed to the MAC layer scheduler.The HOL delay refers to the interval between the time the first packet to be transmitted pending at the packet transmission queue and the time it is received by the UE.This parameter was present in “nr-mac-sched-sap.h” library and accessible through m_rlcTransmissionQueueHolDelay parameter.

References:https://www.nsnam.org/doxygen/structns3_1_1_ff_mac_sched_sap_provider_1_1_sched_dl_rlc_buffer_req_parameters.html

4. Average Rate/ Available Rate

While researching we found that this parameter is an extension of PF (Proportional Fair scheduler).So we inherited the given parameter from nr-ue-info-pf.h header file.

Stages of MLWDF Scheduler Implementation


1
STEP
Research Paper
Read and understand the research paper thoroughly
2
STEP
Parameters
Find out the important parameters required to implement MLWDF
3
STEP
ns3-nr code
Read the ns3-nr code and figure out how similar parameters have been implemented in it
4
STEP
Implementation
After the parameters have been found, figure out how to access them and where to add them
5
STEP
Debug
Debug the errors

Detailing how the project is implemented in/using NS-3 (Provide details of modifications to NS-3 files and new files created and added)

1. Creating a demo files to be added

   We created replicas of the Proportional Fair scheduling algorithm.

   We renamed the following files:

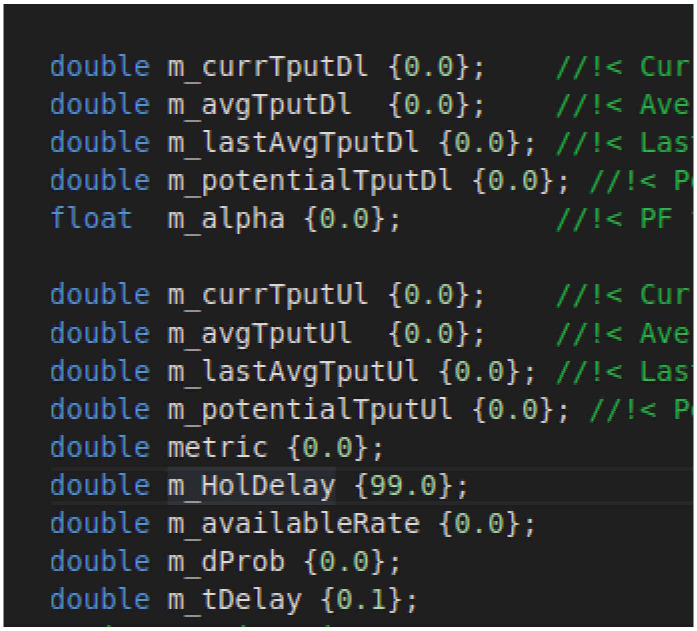
   Nr-mac-scheduler-tdma-pf.cc —> nr-mac-scheduler-tdma-m_lwdf.cc
   Nr-mac-scheduler-tdma-pf.h —> nr-mac-scheduler-tdma-m_lwdf.h
   Nr-mac-scheduler-ue-info-pf.cc —> nr-mac-scheduler-ue-info-m_lwdf.cc
   Nr-mac-scheduler-ue-info-pf.cc —> nr-mac-scheduler-ue-info-m_lwdf.cc

2. Adding those files to ns3-dev/src/nr/model

   We added those files in the wscript and successfully tested it for a single topology.

3. Adding parameters to the code

   We defined all the required parameters apart from available and average throughput as shown in the below figure.



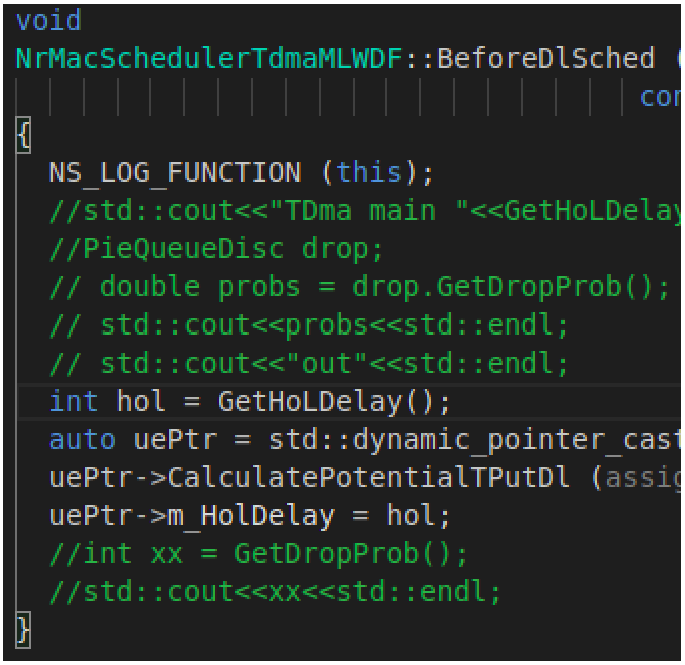
```
double m_currTputDl {0.0};    //!< Cur
double m_avgTputDl  {0.0};    //!< Ave
double m_lastAvgTputDl {0.0}; //!< Las
double m_potentialTputDl {0.0}; //!< P
float  m_alpha {0.0};         //!< PF

double m_currTputUl {0.0};    //!< Cur
double m_avgTputUl  {0.0};    //!< Ave
double m_lastAvgTputUl {0.0}; //!< Las
double m_potentialTputUl {0.0}; //!< P
double metric {0.0};
double m_HolDelay {99.0};
double m_availableRate {0.0};
double m_dProb {0.0};
double m_tDelay {0.1};
```

4. Adding utility functions to the code

   We needed to add various functions in different modules to extract the values of parameters from their respective files

   Adding HOL Delay:

   We realised that the “nr-mac-sched-sap.h” parameters are being passed in before BeforeDlSched function, so it would be easier to access HOL delay from this function rather than modifying “nr-mac-sched-sap.h” module.

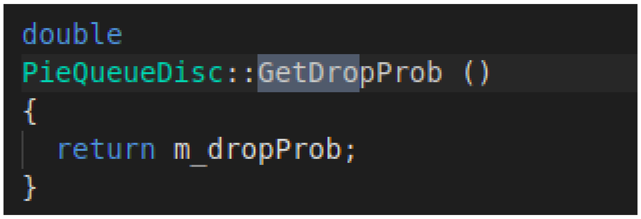


```
void
NrMacSchedulerTdmaMLWDF::BeforeDlSched (
                                       con
{
  NS_LOG_FUNCTION (this);
  //std::cout<<"TDma main "<<GetHoLDelay
  //PieQueueDisc drop;
  // double probs = drop.GetDropProb();
  // std::cout<<probs<<std::endl;
  // std::cout<<"out"<<std::endl;
  int hol = GetHoLDelay();
  auto uePtr = std::dynamic_pointer_cast
  uePtr->CalculatePotentialTPutDl (assig
  uePtr->m_HolDelay = hol;
  //int xx = GetDropProb();
  //std::cout<<xx<<std::endl;
}
```

We defined a Ueptr to access the value of HOL delay as shown in figure.

Adding Drop Prob:

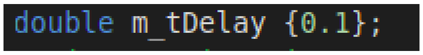
We defined a function GetDropProb in the “pie-queue-disc.h” library to extract the value of drop probability.We couldn’t directly access m_dropprob because of it’s private nature.



```
double
PieQueueDisc::GetDropProb ()
{
  return m_dropProb;
}
```

Adding Target Delay:

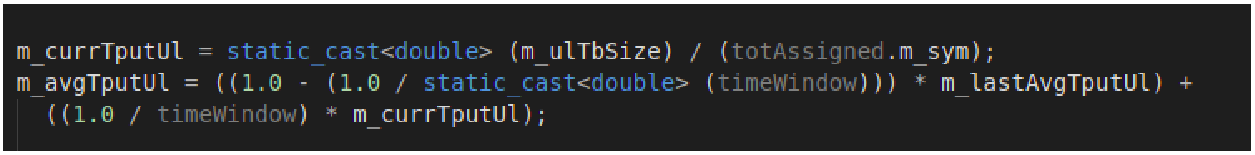
Since it was a constant we initialized it’s value to 100ms at the time of it’s declaration



```
double m_tDelay {0.1};
```

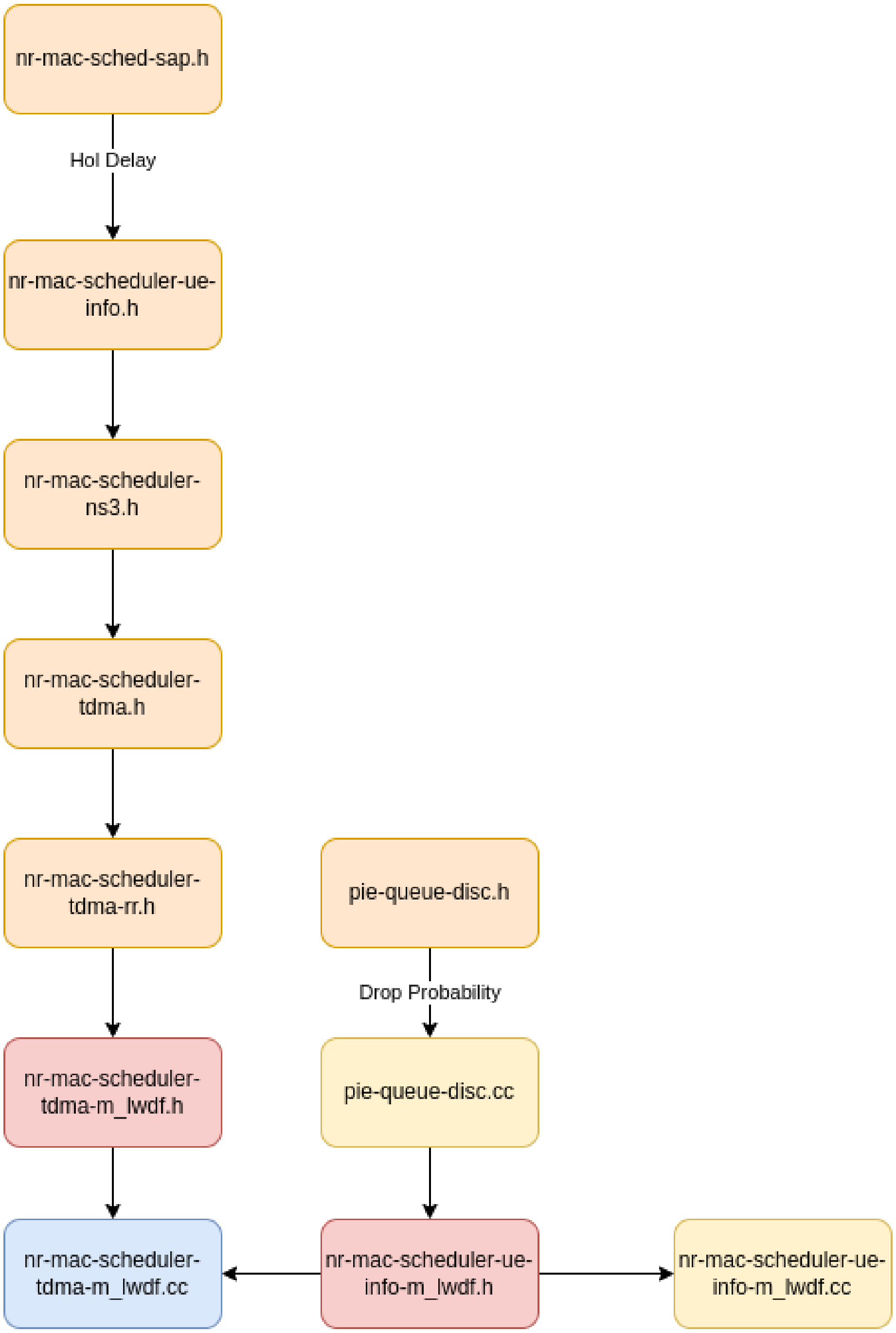
Adding Available and Average Delay:

These parameters were defined in the PF scheduler file and we just added our parameters to the pre existing utility function



```
m_currTputUl = static_cast<double> (m_ulTbSize) / (totAssigned.m_sym);
m_avgTputUl = ((1.0 - (1.0 / static_cast<double> (timeWindow))) * m_lastAvgTputUl) +
  ((1.0 / timeWindow) * m_currTputUl);
```

CODE FLOW


nr-mac-sched-sap.h
Hol Delay
nr-mac-scheduler-ue-info.h
nr-mac-scheduler-ns3.h
nr-mac-scheduler-tdma.h
nr-mac-scheduler-tdma-rr.h
pie-queue-disc.h
Drop Probability
nr-mac-scheduler-tdma-m_lwdf.h
pie-queue-disc.cc
nr-mac-scheduler-tdma-m_lwdf.cc
nr-mac-scheduler-ue-info-m_lwdf.h
nr-mac-scheduler-ue-info-m_lwdf.cc

Discussion about implementation challenges faced and how they are addressed (if applicable)

We faced the following challenges

1. Segmentation Faults

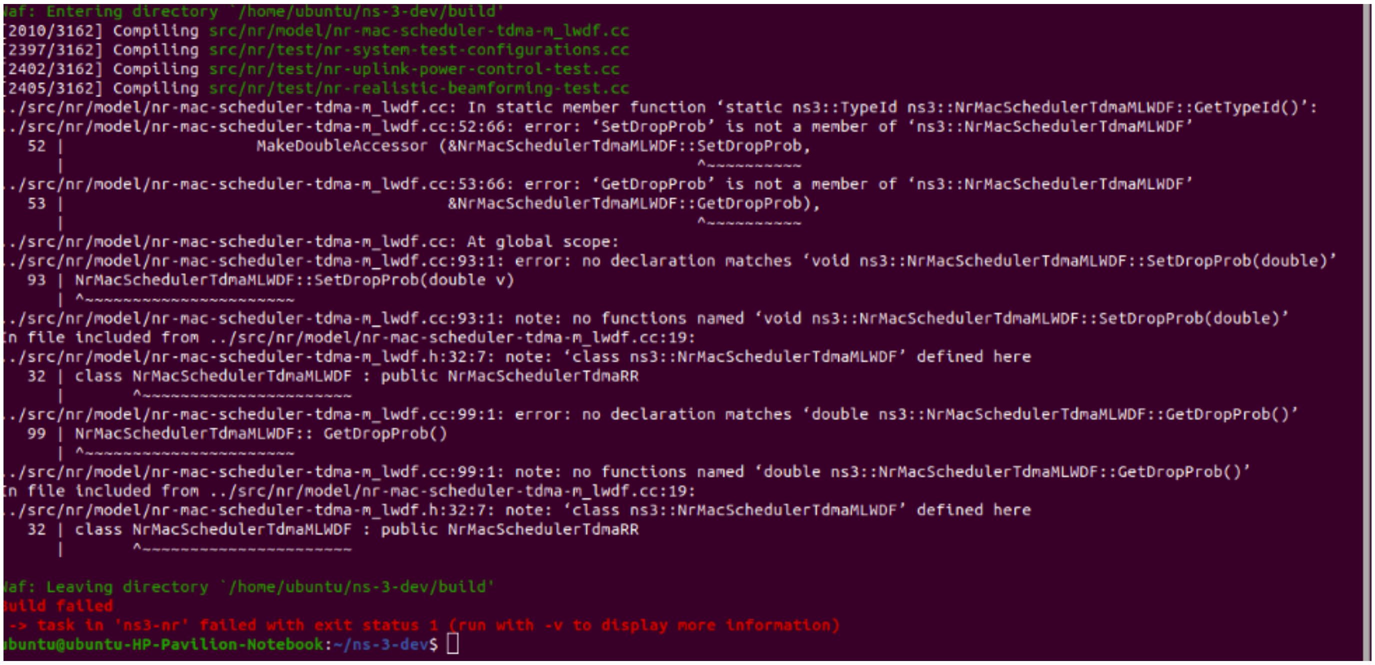
   While accessing the values of the inherited parameters and while defining pointers to access member variables and functions,we encountered segmentation faults

2. Private and undeclared variables

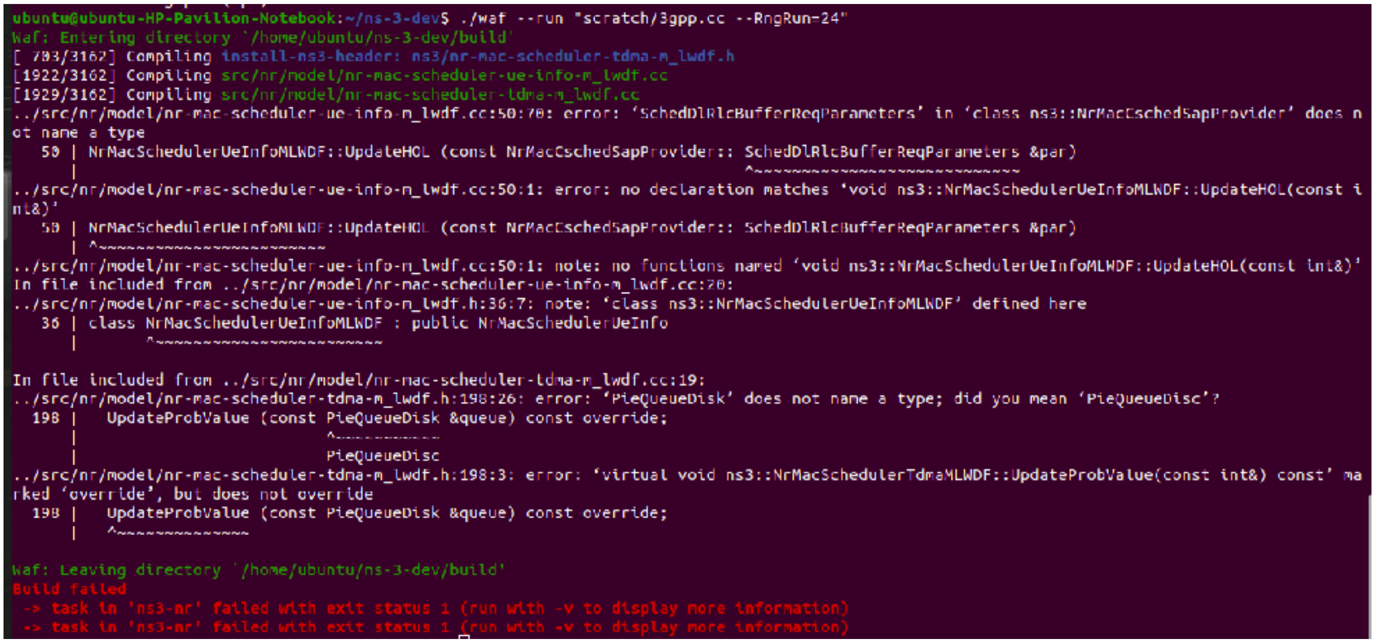
   Retrieving the values of private values from libraries and module required a lot of understanding about the code modules.

3. Definition and declaration of functions

   The functions would show up undeclared and undefined because they weren’t declared in the right file,



```
af: Entering directory `/home/ubuntu/ns-3-dev/build'
[2010/3162] Compiling src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc
[2397/3162] Compiling src/nr/test/nr-system-test-configurations.cc
[2402/3162] Compiling src/nr/test/nr-uplink-power-control-test.cc
[2405/3162] Compiling src/nr/test/nr-realistic-beamforming-test.cc
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc: In static member function ‘static ns3::TypeId ns3::NrMacSchedulerTdmaMLWDF::GetTypeId()’:
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:52:66: error: ‘SetDropProb’ is not a member of ‘ns3::NrMacSchedulerTdmaMLWDF’
   52 |                     MakeDoubleAccessor (&NrMacSchedulerTdmaMLWDF::SetDropProb,
      |                                                                  ^~~~~~~~~~~
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:53:66: error: ‘GetDropProb’ is not a member of ‘ns3::NrMacSchedulerTdmaMLWDF’
   53 |                                       &NrMacSchedulerTdmaMLWDF::GetDropProb),
      |                                                                  ^~~~~~~~~~~
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc: At global scope:
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:93:1: error: no declaration matches ‘void ns3::NrMacSchedulerTdmaMLWDF::SetDropProb(double)’
   93 | NrMacSchedulerTdmaMLWDF::SetDropProb(double v)
      | ^~~~~~~~~~~~~~~~~~~~~~~
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:93:1: note: no functions named ‘void ns3::NrMacSchedulerTdmaMLWDF::SetDropProb(double)’
n file included from ../src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:19:
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.h:32:7: note: ‘class ns3::NrMacSchedulerTdmaMLWDF’ defined here
   32 | class NrMacSchedulerTdmaMLWDF : public NrMacSchedulerTdmaRR
      |       ^~~~~~~~~~~~~~~~~~~~~~~
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:99:1: error: no declaration matches ‘double ns3::NrMacSchedulerTdmaMLWDF::GetDropProb()’
   99 | NrMacSchedulerTdmaMLWDF:: GetDropProb()
      | ^~~~~~~~~~~~~~~~~~~~~~~
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:99:1: note: no functions named ‘double ns3::NrMacSchedulerTdmaMLWDF::GetDropProb()’
n file included from ../src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:19:
./src/nr/model/nr-mac-scheduler-tdma-m_lwdf.h:32:7: note: ‘class ns3::NrMacSchedulerTdmaMLWDF’ defined here
   32 | class NrMacSchedulerTdmaMLWDF : public NrMacSchedulerTdmaRR
      |       ^~~~~~~~~~~~~~~~~~~~~~~

af: Leaving directory `/home/ubuntu/ns-3-dev/build'
uild failed
 -> task in 'ns3-nr' failed with exit status 1 (run with -v to display more information)
buntu@ubuntu-HP-Pavilion-Notebook:~/ns-3-dev$ 
```

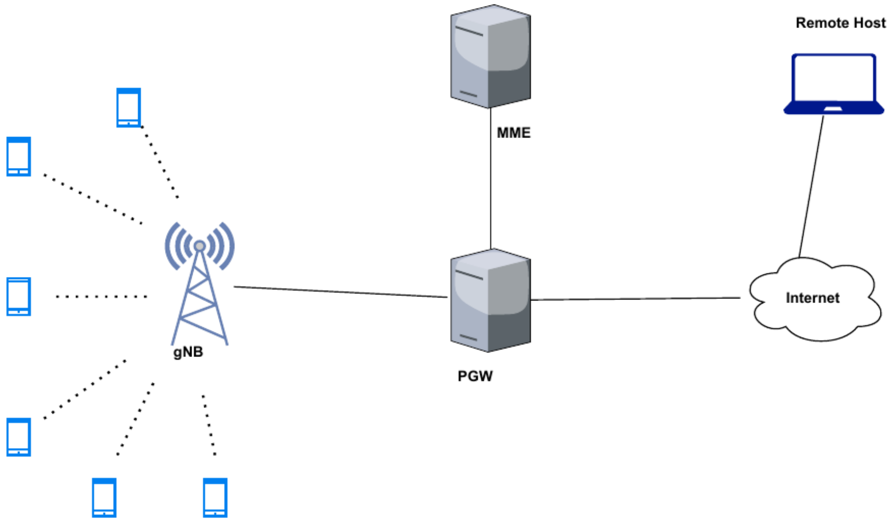


```
ubuntu@ubuntu-HP-Pavilion-Notebook:~/ns-3-dev$ ./waf --run "scratch/3gpp.cc --RngRun=24"
Waf: Entering directory `/home/ubuntu/ns-3-dev/build'
[ 703/3162] Compiling install-ns3-header: ns3/nr-mac-scheduler-tdma-m_lwdf.h
[1922/3162] Compiling src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.cc
[1929/3162] Compiling src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc
../src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.cc:50:70: error: ‘SchedDlRlcBufferReqParameters’ in ‘class ns3::NrMacCschedSapProvider’ does n
ot name a type
   50 | NrMacSchedulerUeInfoMLWDF::UpdateHOL (const NrMacCschedSapProvider:: SchedDlRlcBufferReqParameters &par)
      |                                                                      ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~
../src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.cc:50:1: error: no declaration matches ‘void ns3::NrMacSchedulerUeInfoMLWDF::UpdateHOL(const i
nt&)’
   50 | NrMacSchedulerUeInfoMLWDF::UpdateHOL (const NrMacCschedSapProvider:: SchedDlRlcBufferReqParameters &par)
      | ^~~~~~~~~~~~~~~~~~~~~~~~~
../src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.cc:50:1: note: no functions named ‘void ns3::NrMacSchedulerUeInfoMLWDF::UpdateHOL(const int&)’
In file included from ../src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.cc:20:
../src/nr/model/nr-mac-scheduler-ue-info-m_lwdf.h:36:7: note: ‘class ns3::NrMacSchedulerUeInfoMLWDF’ defined here
   36 | class NrMacSchedulerUeInfoMLWDF : public NrMacSchedulerUeInfo
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~

In file included from ../src/nr/model/nr-mac-scheduler-tdma-m_lwdf.cc:19:
../src/nr/model/nr-mac-scheduler-tdma-m_lwdf.h:198:26: error: ‘PieQueueDisk’ does not name a type; did you mean ‘PieQueueDisc’?
  198 |   UpdateProbValue (const PieQueueDisk &queue) const override;
      |                          ^~~~~~~~~~~~
      |                          PieQueueDisc
../src/nr/model/nr-mac-scheduler-tdma-m_lwdf.h:198:3: error: ‘virtual void ns3::NrMacSchedulerTdmaMLWDF::UpdateProbValue(const int&) const’ ma
rked ‘override’, but does not override
  198 |   UpdateProbValue (const PieQueueDisk &queue) const override;
      |   ^~~~~~~~~~~~~~~

Waf: Leaving directory `/home/ubuntu/ns-3-dev/build'
Build failed
 -> task in 'ns3-nr' failed with exit status 1 (run with -v to display more information)
 -> task in 'ns3-nr' failed with exit status 1 (run with -v to display more information)
```

Simulation/experimental Setup

Test scenario created/used to conduct the experiments

We created a minor topology to test out the scheduler behaviour.It consisted of 6 UEs places at (1000,0) (3000,0) (-1000,0) (-3000,0) (-10,0) (10,0). In this way we have behaviours of the scheduler in the best and worst case scenarios.We also incorporated mobility models to check the behavior of the schedule in mobile Ues.



Configuration Parameters

Simulation Parameter	Value
Number of UEs	6; 1 Downlink UDP Flow per UE from the Remote Host.
Number of gNBs	1
Locations of UE (in meters)	(10,0), (1000,0), (3000,0), (-10,0), (-1000,0), (-3000,0)
Base Station position	(0,0)
gNB Tx Power	23 dBm
S1-U Link Delay between gNodeB and P-GW	2 ms
P2P link between P-GW and Remote Host	Data Rate: 10 Gbps Link Delay: 5 ms
Channel model	3GPP, LoS

Channel bandwidth	**50 MHz**
Central frequency	**6 GHz for numerologies 0,1,2** **28 GHz for numerology 3**
Scenario	**UMa_LoS**
Shadowing	**disabled**
Application Type	**UDP Client and UDP Server**
BandWidth Part	**1 bandwidth part (1 for DL). Create one component carrier (CC) and set the parameters “Numerology”, ”Pattern”, and ”TxPower”.**
RLC MaxTxBufferSize	**999999999**
Antennas for all the UEs	**NumRows: 2** **NumColumns: 4** **AntennaElement: IsotropicAntennaModel**
Antennas for all the gNbs	**NumRows: 4** **NumColumns: 8** **AntennaElement: ThreeGppAntennaModel**
BeamformingMethod	**DirectPathBeamforming**
Error Model	**NrEesmIrT1**

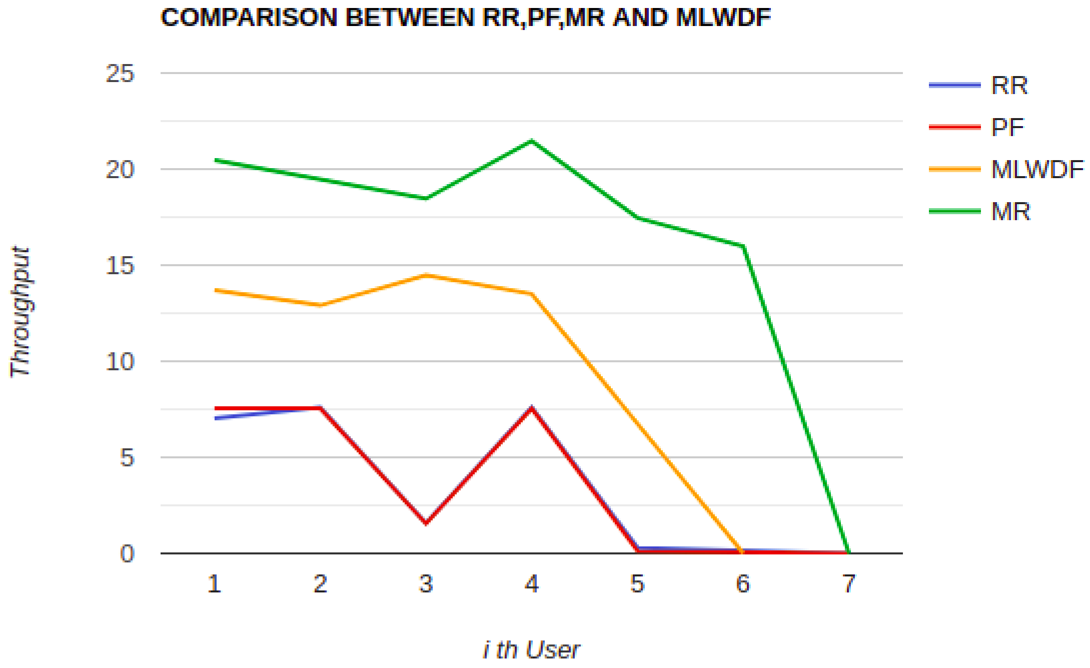
Packet Size	**1500 bytes**
AmcModel	**ShannonModel**
Height of Base Station and UE	**10 Meters / 1.5 Meters**
Full buffer case (UDP Traffic)	**Set minimum 30 Mbps per each DL flow (1500 Byte packets, 2500 packets per sec)**
Non Full buffer case (UDP Traffic)	**Set maximum 12 Mbps per each DL flow (1500 Byte packets, 1000 packets per sec)**
Total simulation time	**5 seconds (Static Scenario)**

	20 seconds (Mobile scenarios)

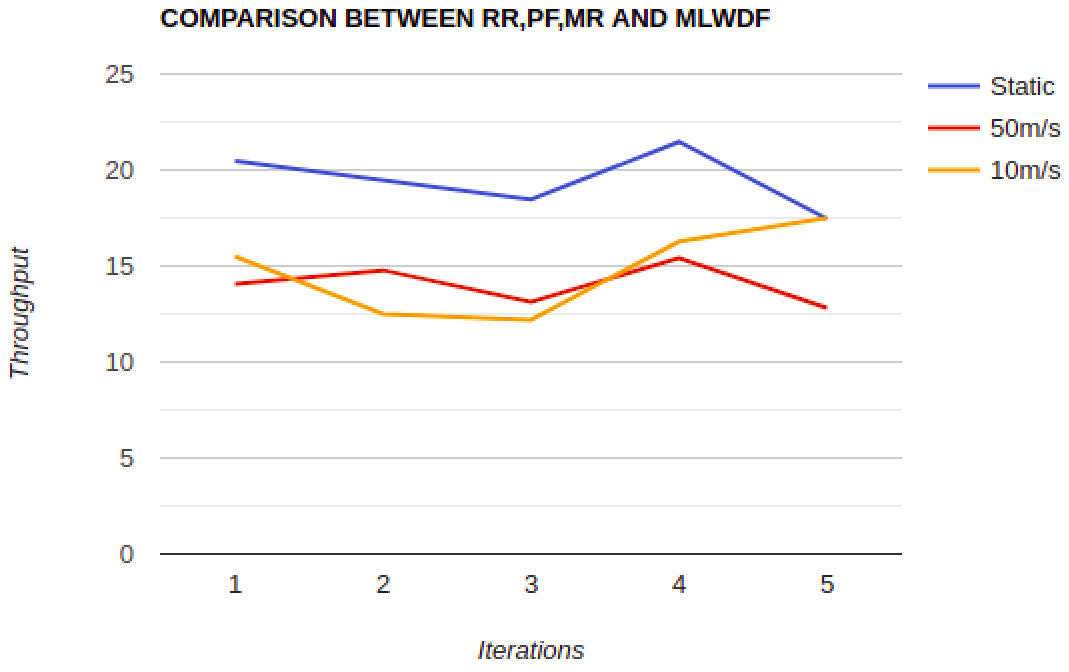
Performance Metrics

The below mentioned are the different test case observations and

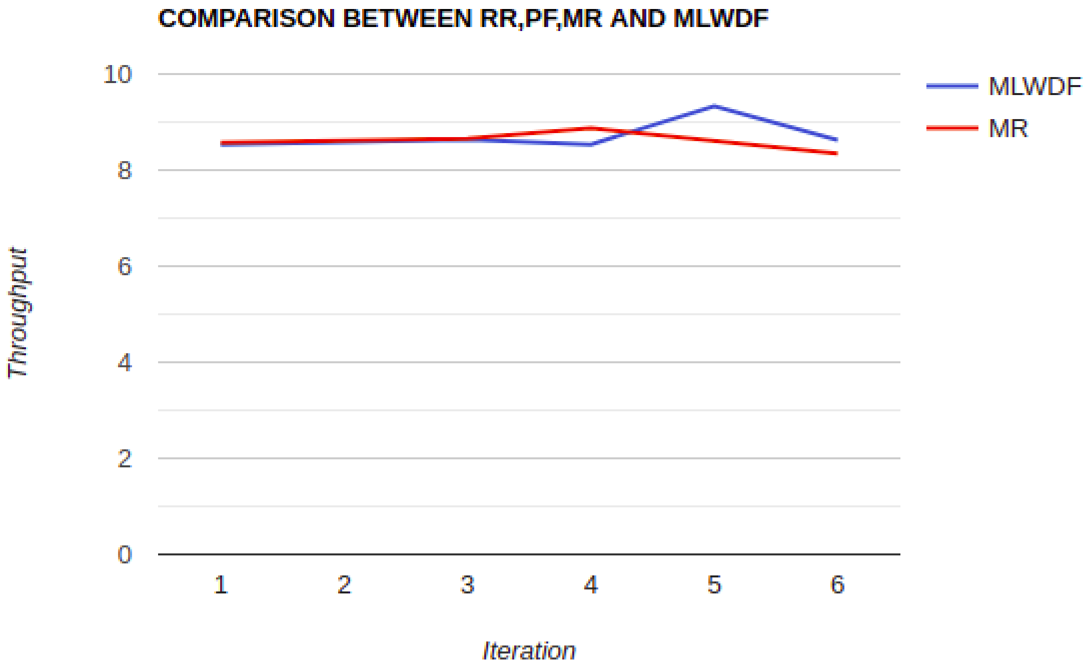
A)Comparison based on the placement of UEs



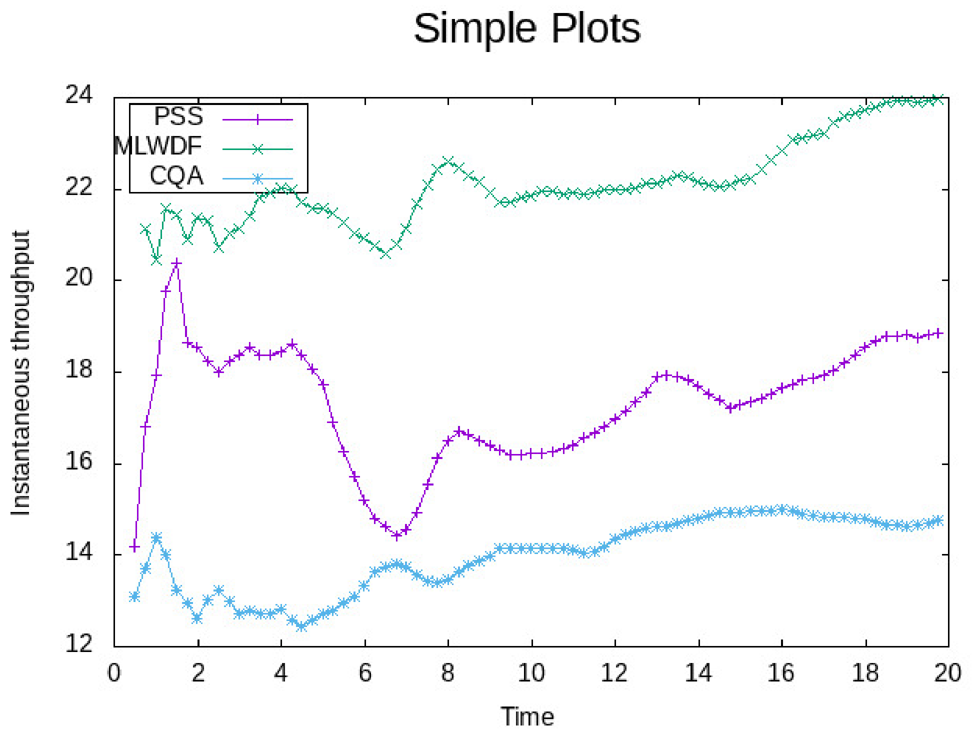
B)Comparison based on the mobility model



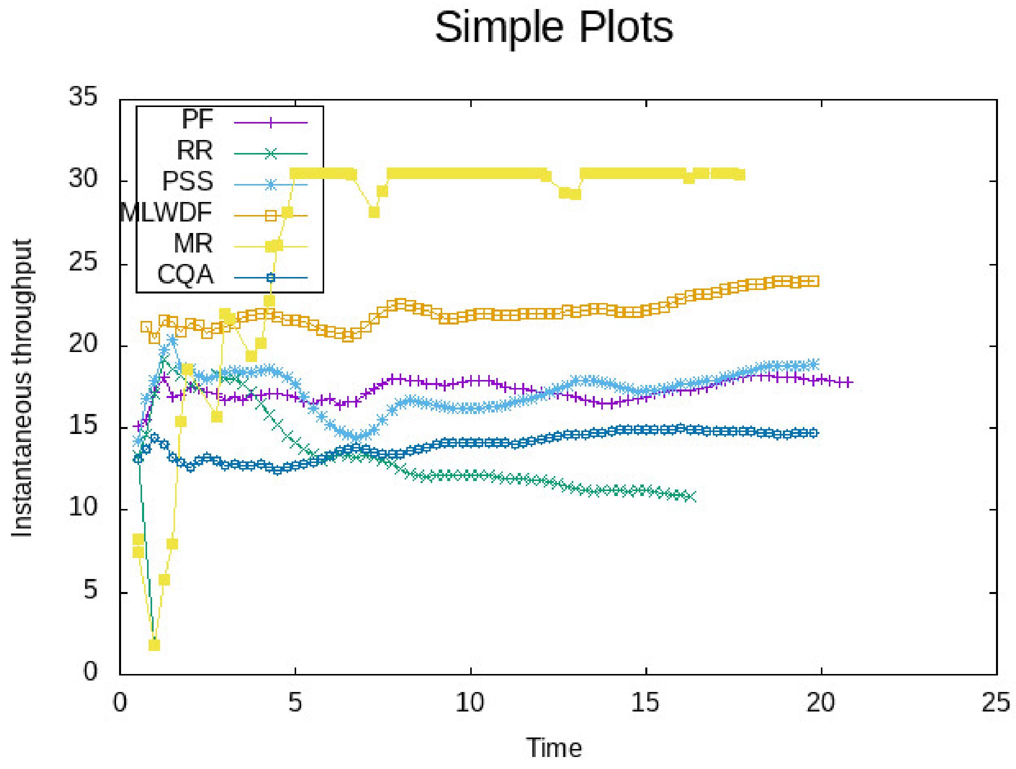
C)Comparison based on the average delay



D)Comparison with other implementations of schedulers done in our batch


Simple Plots
PSS
MLWDF
CQA
Instantaneous throughput
24
22
20
18
16
14
12
0
2
4
6
8
10
12
14
16
18
20
Time

Conclusion


Simple Plots
PF
RR
PSS
MLWDF
MR
CQA
Instantaneous throughput
35
30
25
20
15
10
5
0
0
5
10
15
20
25
Time

1. MLWDF outperforms PF and RR

   This happens because MLWDF considers a lot more parameters than average and available throughput.It also doesn’t blindly allocate resources like RR. The drop probability and HOL delay conveys about the channel condition and traffic conditions observed at different UEs. Hence it could establish a better relationship between fairness and throughput.

2. MLWDF cannot outperform MR

   This is because the motivation of MLWDF also includes fairness so there is a possibility that it might not be selecting the highest throughput providing UEs all the time unlike MR.

Member Contribution

1)Pallavi Saxena

- Researching MLWDF equation
- Figuring out how to add parameters and functions to the base file
- Debugging

2)Satvik Padhiyar

- Incorporating HOL parameter
- Research Paper analysis
- Debugging

3)Pradhumn Kanase

- Incorporation of Drop Probability
- Integration of MLWDF module
- Debugging

Combined effort: We didn’t have any work segregation.We all used to have a google meet and shared our understanding about the code and how to proceed further.

- 1)Making report
- 2)Reasoning observations
- 3)Giving and making presentation
- 4) Construct topological model for testing

References

Mac scheduling algorithm files present in Ns3-dev

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5340336

https://opus.lib.uts.edu.au/bitstream/10453/10894/1/2009000660.pdf

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1400171