

MODUL JARINGAN KOMPUTER

**PANDUAN INSTALASI DAN PENGGUNAAN NS2 (NETWORK
SIMULATOR 2) PADA SISTEM OPERASI WINDOWS**



Disusun Oleh:

Naufal Faiz Alfarizi

20206052001

Modul Pembelajaran

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
2021

KATA PENGANTAR

Puji syukur kehadiran ALLAH SWT yang telah melimpahkan rahmatnya, yang telah memberikan nikmat kesehatan serta kesempatan yang diberikan hingga dapat menyelesaikan pembuatan modul jaringan komputer terkait dengan panduan penggunaan NS2 (Network Simulator 2).

Modul ini disusun bertujuan untuk membantu para mahasiswa dalam mencari referensi Penggunaan dari NS2 (Network Simulator 2), serta menambah ilmu pengetahuan. Didalam modul ini tentunya tidak terlepas dari kesalahan dan kekurangan, maka dari itu diharapkan mahasiswa agar dapat menambah referensi lain dari berbagai sumber.

Akhir kata semoga modul ini dapat bermanfaat bagi mahasiswa dan bagi para pembaca yang berminat pada umumnya. Atas Perhatiannya penyusun mengucapkan terima kasih.

Penyusun

Naufal Faiz Alfarizi

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
DAFTAR TABEL.....	iv
PENDAHULUAN.....	1
NS2 (NETWORK SIMULATOR 2).....	1
1.1 NETWORK SIMULATOR.....	1
1.1.1 Penjelasan Tentang NS2 (Network Simulator 2).....	1
1.1.2 Output NS2.....	2
1.2 TUJUAN UMUM.....	3
PANDUAN INSTALASI NS2.....	4
PADA WINDOWS.....	4
2.1 KEBUTUHAN.....	4
2.2 LANGKAH INSTALASI.....	4
STUDI KASUS.....	10
SIMULASI TCP TAHOE DENGAN MODEL ANTRIAN RED DAN DROPTAIL.....	10
3.1 TCP TAHOE.....	10
3.2 ANTRIAN RED.....	10
3.3 ANTRIAN DROPTAIL.....	11
3.4 PERENCANAAN SIMULASI.....	11
3.5 PERANCANGAN TOPOLOGI.....	12
3.6 VARIABLE YANG DIHITUNG.....	13
3.7 OUTPUT NAM DAN AWK.....	14
LAMPIRAN SCRIPT.....	16
Q AND A.....	24

DAFTAR GAMBAR

Gambar 1. 1 Hubungan C++ dan OTcl.....	1
Gambar 2. 1 Enable Windows Subsystem for Linux.....	5
Gambar 2. 2 Mirosoft Store Ubuntu.....	5
Gambar 2. 3 Ubuntu 20.04 LTS.....	6
Gambar 2. 4 Download Xming X.....	6
Gambar 2. 5 Perintah install NS2.....	7
Gambar 2. 6 Perintah Install NAM.....	7
Gambar 2. 7 Perintah Install Gedit.....	7
Gambar 2. 8 Sample Script.....	8
Gambar 2. 9 Tampilan nam.....	9
Gambar 2. 10 Hasil Akhir Sample NS2.....	9
Gambar 3. 1 Antrian RED.....	10
Gambar 3. 2 Antrian Droptail.....	11
Gambar 3. 3 Perancangan Topologi.....	13
Gambar 3. 4 Output Nam RED.....	14
Gambar 3. 5 Output Trace RED.....	15
Gambar 3. 6 Output Nam Droptail.....	15
Gambar 3. 7 Output Trace Droptail.....	15

DAFTAR TABEL

Tabel 1. 1 Penjelasan Istilah	2
Tabel 3. 1 Simulasi Antrian RED.....	11
Tabel 3. 2 Simulasi Antrian Droptail.....	12

PENDAHULUAN

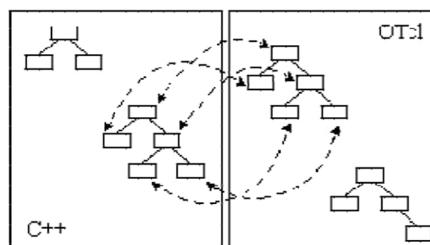
NS2 (NETWORK SIMULATOR 2)

1.1 NETWORK SIMULATOR

1.1.1 Penjelasan Tentang NS2 (Network Simulator 2)

Pada dasarnya NS2 atau Network Simulator 2 merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan simulasi jaringan komputer terkait dengan topologi yang digunakan dan bagaimana proses komunikasi pada suatu jaringan sedang berlangsung. Network Simulator juga dapat digunakan pada jaringan kabel dan komunikasi jaringan wireless atau tanpa kabel. Pada Network Simulator memiliki tampilan dengan node yang bergerak atau juga node yang tidak bergerak yang tentunya tidak sama dengan keadaan yang sebenarnya karena tujuan menggunakan Network Simulator yaitu untuk memberikan gambaran dengan simulasi terkait dengan rancangan jaringan komputer yang akan kita buat.

Network Simulator dibangun dengan menggunakan 2 bahasa pemrograman, yaitu C++ dan Tcl/Tk. C++ digunakan untuk library yang berisi event scheduler, protokol dan network component yang diimplementasikan pada simulasi oleh user. Tcl/Tk digunakan pada skript simulasi yang ditulis oleh NS user dan pada library sebagai simulator objek. Tk juga nantinya berperan sebagai interpreter. Hubungan antara Bahasa pemrograman dapat dilihat pada gambar 1.1.



Gambar 1. 1 Hubungan C++ dan OTcl

Pada gambar 1.1 terkait dengan hubungan antara Bahasa pemrogram pada Network Simulator dapat diperoleh penjelasan istilah seperti pada tabel 1.1:

Tabel 1. 1 Penjelasan Istilah

1. TCL	:	Tool Command Language
2. OTcL	:	Object TCL
3. TK	:	Tool Kit
4. NS-2	:	Network Simulator Versi - 2
5. Nam	:	Network Animator

Bahasa C++ digunakan pada library karena C++ mampu mendukung runtime simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar. Bahasa Tcl memberikan respon runtime yang lebih lambat daripada C++, namun jika terdapat kesalahan syntax dan perubahan *script* berlangsung dengan cepat dan interaktif. User dapat mengetahui letak kesalahannya yang dijelaskan pada konsole, sehingga user dapat memperbaiki dengan cepat. Karena alasan itulah bahasa ini dipilih untuk digunakan pada *Script* simulasi.

1.1.2 Output NS2

Setelah melakukan simulasi dengan Network Simulator, maka NS2 akan memberikan output berbasis text-based yang berisi tentang hasil dari simulasi yang dideklarasikan. Terdapat 2 jenis file output antara lain sebagai berikut:

1. File Trace yang akan digunakan untuk analisa numerik
2. File NamTrace yang digunakan untuk menampilkan input tampilan grafis yang dibaca oleh Nam.

1.2 TUJUAN UMUM

Pada modul jaringan komputer tentang panduan penggunaan NS2 sebagai Network Simulator, memiliki tujuan umum sebagai berikut:

1. Mahasiswa dapat menggunakan NS2 sebagai Network Simulator yang running pada Windows.
2. Mahasiswa dapat menerapkan sample studi kasus yang digunakan pada modul ini sebagai gambaran atau contoh simulasi.
3. Mahasiswa dapat membuat Simulasi Jaringan pada model jaringan wired dengan TCP Variants.

PANDUAN INSTALASI NS2 PADA WINDOWS

2.1 KEBUTUHAN

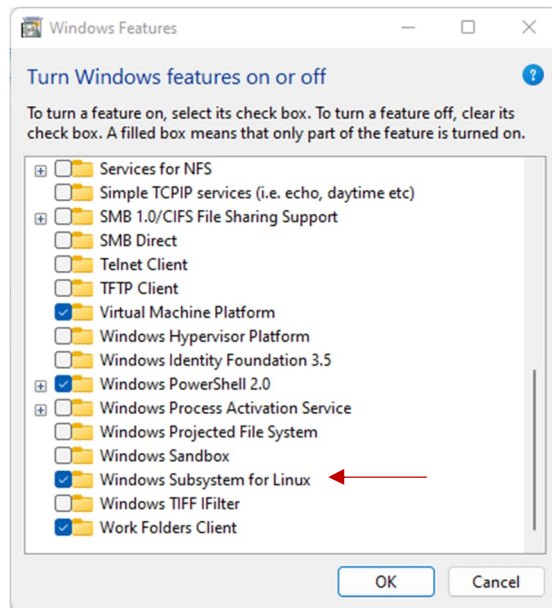
Pada dasarnya NS2 berjalan diatas environment Linux namun, dapat juga berjalan pada windows dengan menggunakan subsystem linux yaitu WSL (Windows Subsystem Linux) yang merupakan dukungan dari microsoft secara resmi mulai Oktober 2018. Singkatnya, fitur ini menawarkan kemampuan menjalankan terminal UNIX di dalam Windows. Beberapa kebutuhan yang perlu disiapkan sebelum melakukan instalasi NS2 pada windwos antara lain sebagai berikut:

1. Sistem Operasi Windows 8 atau diatasnya
2. Ubuntu LTS (on Windows Store)
3. Xming sebagai perantara visual
4. NAM
5. Gedit untuk melakukan editing script yang ditulis.

2.2 LANGKAH INSTALASI

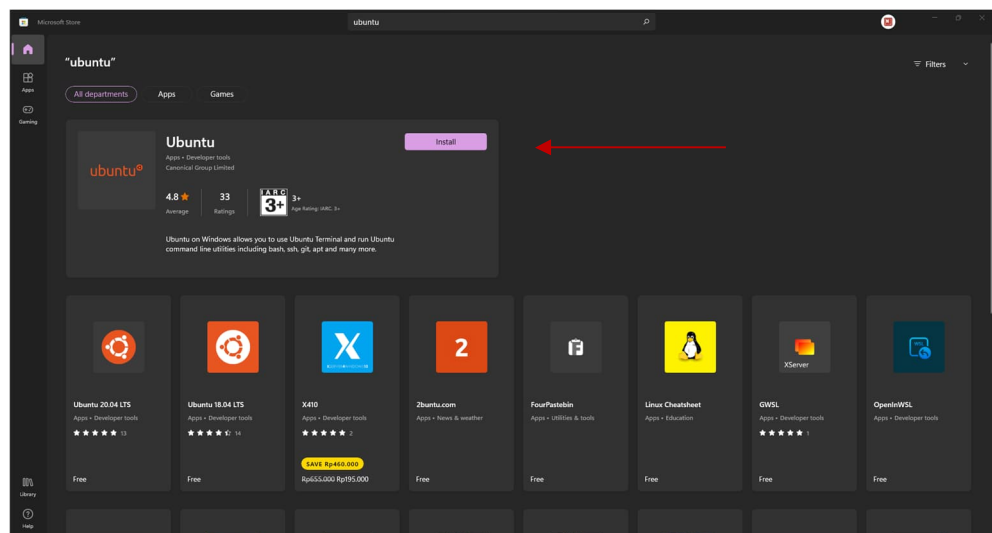
Dalam panduan Langkah instalasi pada modul ini akan menjelaskan mulai dari enable Windows Subsystem Linux pada feature program hingga running sample sederhana. Dengan Langkah – Langkah yang dapat dijabarkan seperti berikut:

1. Pertama yang perlu dilakukan adalah enable Windows Subsystem Linux pada feature program dengan cara masuk ke **Control Panel > Programs > Programs and Features > Turn Windows features on or off > berikan centang pada “Windows Subsystem for Linux”**. Seperti pada gambar berikut:



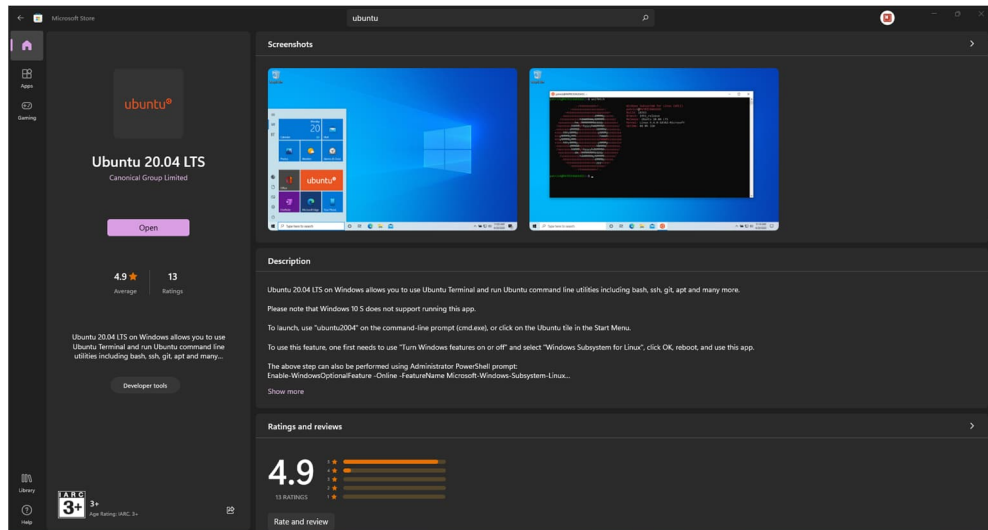
Gambar 2. 1 Enable Windows Subsystem for Linux

2. Selanjutnya masuk pada Microsoft store untuk melakukan proses download dan install Ubuntu. Seperti gambar berikut:



Gambar 2. 2 Mirosoft Store Ubuntu

3. Untuk pemilahan versi ubuntu bebas saja karena yang disediakan pada Microsoft store semuanya sudah support untuk menjalankan NS2 dan yang saya gunakan adalah versi Ubuntu 20.04 LTS.



Gambar 2. 3 Ubuntu 20.04 LTS

4. Setelah selesai melakukan instalasi Ubuntu, selanjutnya melakukan instalasi Xming X Server for Windows dapat didownload pada link berikut <https://sourceforge.net/projects/xming/>



Gambar 2. 4 Download Xming X

5. Setelah semua kebutuhan sudah disiapkan, selanjutnya buka ubuntu untuk melakukan proses instalasi NS2, Nam dan Gedit pada Ubuntu yang berjalan diatas environment Windows.

6. Perintah Install NS2 (sudo apt-get install ns2)

```
xloyd@Xloyd-PC: ~  
xloyd@Xloyd-PC:~$ sudo apt-get install ns2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libotcl1 libtcl8.6 libtclcl1 libtk8.6 libxss1  
Suggested packages:  
  tcl8.6 tk8.6 gnuplot  
The following NEW packages will be installed:  
  libotcl1 libtcl8.6 libtclcl1 libtk8.6 libxss1 ns2  
0 upgraded, 6 newly installed, 0 to remove and 86 not upgraded.  
Need to get 4149 kB of archives.  
After this operation, 22.6 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libtcl8.6 amd64 8.6.10+dfsg-1 [902 kB]  
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 libotcl1 amd64 1.14+dfsg-4 [22.1 kB]  
Get:3 http://archive.ubuntu.com/ubuntu focal/universe amd64 libtclcl1 amd64 1.20-9build1 [63.2 kB]  
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 libxss1 amd64 1:1.2.3-1 [8140 B]  
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 libtk8.6 amd64 8.6.10-1 [714 kB]  
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 ns2 amd64 2.35+dfsg-3build1 [2440 kB]  
Fetched 4149 kB in 28s (146 kB/s)  
Selecting previously unselected package libtcl8.6:amd64.  
(Reading database ... 32223 files and directories currently installed.)  
Preparing to unpack .../0-libtcl8.6_8.6.10+dfsg-1_amd64.deb ...  
Unpacking libtcl8.6:amd64 (8.6.10+dfsg-1) ...  
Selecting previously unselected package libotcl1:amd64.  
Preparing to unpack .../1-libotcl1_1.14+dfsg-4_amd64.deb ...  
Unpacking libotcl1:amd64 (1.14+dfsg-4) ...  
Selecting previously unselected package libtclcl1:amd64.  
Preparing to unpack .../2-libtclcl1_1.20-9build1_amd64.deb ...  
Unpacking libtclcl1:amd64 (1.20-9build1) ...  
Selecting previously unselected package libxss1:amd64.  
Preparing to unpack .../3-libxss1_1:1.2.3-1_amd64.deb ...  
Unpacking libxss1:amd64 (1:1.2.3-1) ...  
Selecting previously unselected package libtk8.6:amd64.  
Preparing to unpack .../4-libtk8.6_8.6.10-1_amd64.deb ...  
Unpacking libtk8.6:amd64 (8.6.10-1) ...  
Selecting previously unselected package ns2.  
Preparing to unpack .../5-ns2_2.35+dfsg-3build1_amd64.deb ...  
Unpacking ns2 (2.35+dfsg-3build1) ...  
Setting up libtcl8.6:amd64 (8.6.10+dfsg-1) ...  
Setting up libxss1:amd64 (1:1.2.3-1) ...  
Setting up libotcl1:amd64 (1.14+dfsg-4) ...  
Setting up libtk8.6:amd64 (8.6.10-1) ...  
Setting up libtclcl1:amd64 (1.20-9build1) ...  
Setting up ns2 (2.35+dfsg-3build1) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...  
xloyd@Xloyd-PC:~$
```

Gambar 2. 5 Perintah install NS2

7. Perintah Install NAM (sudo apt-get install nam)

```
xloyd@Xloyd-PC: ~  
xloyd@Xloyd-PC:~$ sudo apt-get install nam  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  nam  
0 upgraded, 1 newly installed, 0 to remove and 86 not upgraded.  
Need to get 196 kB of archives.  
After this operation, 695 kB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 nam amd64 1.15-5build1 [196 kB]  
Fetched 196 kB in 2s (89.3 kB/s)  
Selecting previously unselected package nam.  
(Reading database ... 32599 files and directories currently installed.)  
Preparing to unpack .../nam_1.15-5build1_amd64.deb ...  
Unpacking nam (1.15-5build1) ...  
Setting up nam (1.15-5build1) ...  
Processing triggers for man-db (2.9.1-1) ...  
xloyd@Xloyd-PC:~$
```

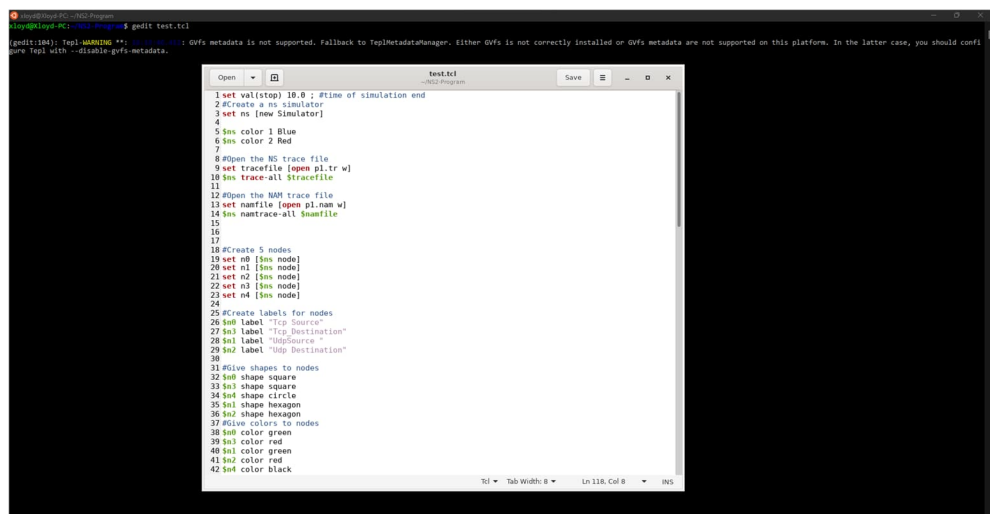
Gambar 2. 6 Perintah Install NAM

8. Perintah Install Gedit (sudo apt-get install gedit)

```
xloyd@Xloyd-PC: ~  
xloyd@Xloyd-PC:~$ sudo apt-get install gedit  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:
```

Gambar 2. 7 Perintah Install Gedit

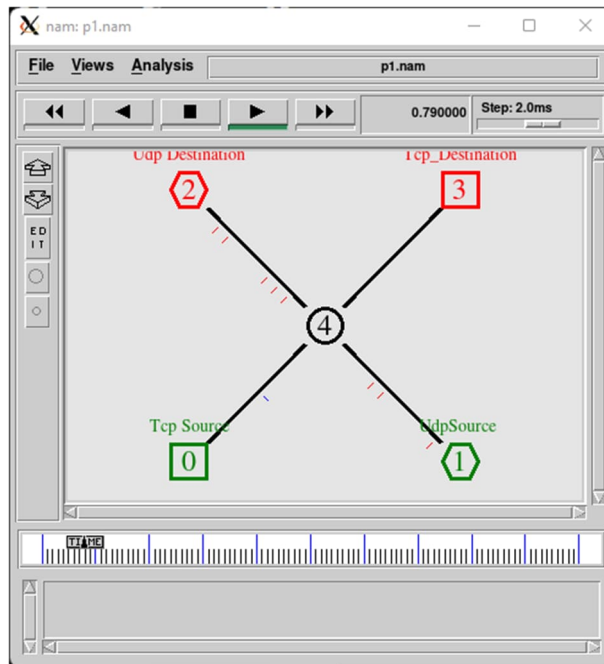
9. Jika sudah terinstall semua, kemudian buat folder dengan perintah `mkdir [nama_file]` lalu masuk ke directory tersebut untuk membuat script dengan ekstensi `*.tcl`.
10. Kemudian kita membuat script tersebut dengan menggunakan gedit dengan contoh perintah `gedit test.tcl` maka selanjutnya akan diarahkan ke workspace kosong yang akan diisi sample script dari simulasi jaringan komputer. Script tersebut saya lampirkan pada halaman lampiran.



```
1set val(stop) 10.0 ; #time of simulation end
2create a ns simulator
3set ns [new Simulator]
4
5$ns color 1 Blue
6$ns color 2 Red
7
8#Open the NS trace file
9set tracefile [open p1.tr w]
10$ns trace-all $tracefile
11
12#Open the NAM trace file
13set namfile [open p1.nam w]
14$ns namtrace-all $namfile
15
16
17
18#create 5 nodes
19set n0 [$ns node]
20set n1 [$ns node]
21set n2 [$ns node]
22set n3 [$ns node]
23set n4 [$ns node]
24
25#create labels for nodes
26$ns label "tcp Source"
27$ns label "tcp Destination"
28$ns label "udpSource "
29$ns label "udp Destination"
30
31#Give shapes to nodes
32$ns shape square
33$ns shape square
34$ns shape circle
35$ns shape hexagon
36$ns shape hexagon
37#Give colors to nodes
38$ns color green
39$ns color red
40$ns color green
41$ns color red
42$ns color black
```

Gambar 2. 8 Sample Script

11. Setelah script selesai dibuat, lalu save kemudian execute script tersebut dengan perintah `ns test.tcl`. hingga menampilkan tampilan seperti pada gambar 2.9 yang merupakan hasil dari simulasi jaringan komputer yang telah dibuat pada script tersebut apabila tampilan belum muncul bisa melakukan pengecekan apakah file nam sudah tersedia jika sudah bisa execute manual dengan perintah `nam [namfile.nam]`.



Gambar 2. 9 Tampilan nam

12. Hasil akhirnya adalah output 2 buah file yang pertama merupakan file .nam dan yang kedua merupakan file trace dengan ekstensi .tr yang dapat dibuka dan dibaca menggunakan script awk dengan perintah **gawk -f [namafilename.awk] [namafilename.tr]** untuk menghitung hasil dari simulasi jaringan komputer sederhana yang telah dibuat.

```
xloyd@Xloyd-PC: ~/NS2-Program
xloyd@Xloyd-PC:~/NS2-Program$ ls -lt
total 4360
-rw-r--r-- 1 xloyd xloyd 1787 Nov 17 17:53 openAwk.awk
-rw-r--r-- 1 xloyd xloyd 3078054 Nov 17 16:21 p1.nam
-rw-r--r-- 1 xloyd xloyd 1263487 Nov 17 16:21 p1.tr
-rw-r--r-- 1 xloyd xloyd 2583 Nov 17 16:21 test.tcl
xloyd@Xloyd-PC:~/NS2-Program$ gawk -f openAwk.awk p1.tr

-----
Throughput TCP = 25.415 Mbit
Avg throughput TCP = 0.0508301 Mbps
-----
```

Gambar 2. 10 Hasil Akhir Sample NS2

STUDI KASUS

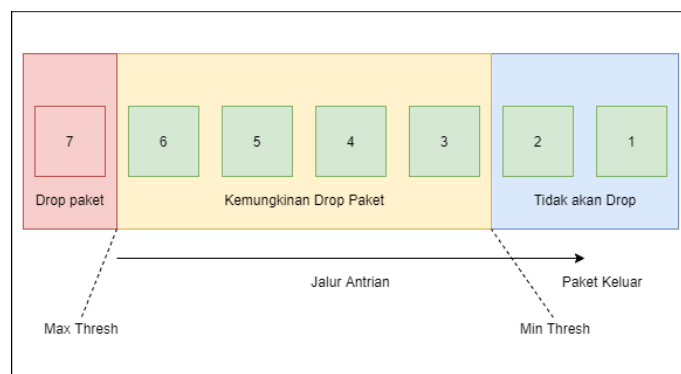
SIMULASI TCP TAHOE DENGAN MODEL ANTRIAN RED DAN DROPTAIL

3.1 TCP TAHOE

Pada modul ini akan diberikan contoh studi kasus dengan membuat simulasi menggunakan model TCP Tahoe. TCP Tahoe adalah algoritma yang lebih sederhana dari varian TCP lainnya, TCP Tahoe mengacu pada algoritma congestion control sebagai TCP yang disarankan oleh Van Jacobson. TCP Tahoe didasarkan tiga algoritma congestion control yaitu, Slow Start (SS), Congestion Avoidance (CA) dan fast retransmit. Tetapi, TCP Tahoe tidak menggunakan algoritma fast recovery. Pada fase congestion avoidance, TCP Tahoe memberlakukan duplikasi tiga ACK sama dengan time-out. Ketika proses duplikat tiga ACK diterima, TCP Tahoe akan menggunakan fast retransmit dengan menurunkan CongWin menjadi 1 dan mulai masuk ke fase Slow Start.

3.2 ANTRIAN RED

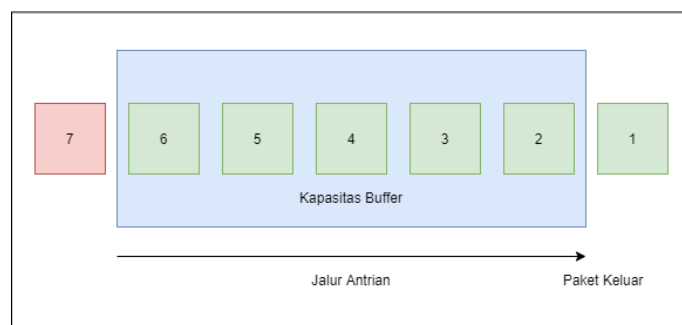
Random Early Detection merupakan mekanisme antrian yang dapat melakukan packet drop sebelum buffer penuh dengan cara menentukan nilai min thresh dan max thresh. Jika data kurang dari min thresh, maka data akan dilayani atau diproses. Jika data diantara min thresh dan max thresh, maka data akan ditandai dan di drop secara acak. Dan jika data melebihi max thresh maka data langsung di drop. Untuk ilustrasi atau gambaran mengenai model antrian Random Early Detection dapat dilihat pada gambar 3.1.



Gambar 3. 1 Antrian RED

3.3 ANTRIAN DROPTAIL

Antrian droptail merupakan bagian dari penjadwalan FIFO (First In First Out) dimana data yang datang terlebih dahulu akan keluar terlebih dahulu juga. Manajemen antrian droptail menggunakan penjadwalan FIFO ketika ruang antrian buffer sudah penuh maka paket akan langsung di drop dengan tidak ada perlakuan khusus terhadap paket. Semua paket dianggap sama dan tidak ada prioritas untuk paket tertentu. Ilustrasi atau gambaran mengenai model antrian droptail dapat dilihat pada gambar 3.2.



Gambar 3. 2 Antrian Droptail

3.4 PERENCANAAN SIMULASI

Perancangan simulasi dalam penelitian bertujuan untuk menentukan parameter yang digunakan. Pada dasarnya parameter digunakan sebagai nilai acuan yang berfungsi untuk melakukan proses komputasi ketika simulasi dilakukan. Parameter simulasi dalam penelitian ini dapat dilihat pada tabel 3.1 untuk metode antrian Random Early Detection dan tabel 3.2 untuk metode antrian Droptail

Tabel 3. 1 Simulasi Antrian RED

No	Parameter Simulasi	Nilai
1	Varian TCP	TCP Tahoe
2	Model Antrian	Random Early Detection
3	Link Node n0-n2, n1-n2	Bandwith: 10 Mbps Delay Propagation: 10 Mbps
4	Link Node n2-n3	Bandwith: 2 Mbps Delay Propagation: 5 Mbps

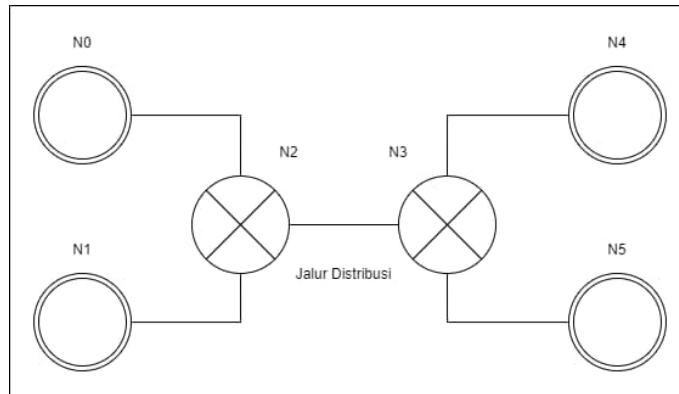
5	Link Node n3-n4, n3-n5	Bandwith: 5 Mbps Delay Propagation: 10 Mbps
6	Nilai Min tresh dan Max tresh	Min Thresh: 10 (1x Pengujian) Max Thresh: 60
7	Buffer Size	50 Paket
8	Waktu Simulasi	300 Detik

Tabel 3. 2 Simulasi Antrian Droptail

No	Parameter Simulasi	Nilai
1	Varian TCP	TCP Tahoe
2	Model Antrian	Droptail
3	Link Node n0-n2, n1-n2	Bandwith: 10 Mbps Delay Propagation: 10 Mbps
4	Link Node n2-n3	Bandwith: 2 Mbps Delay Propagation: 5 Mbps
5	Link Node n3-n4, n3-n5	Bandwith: 5 Mbps Delay Propagation: 10 Mbps
6	Nilai Min tresh dan Max tresh	-
7	Buffer Size	20 Paket (1x Pengujian)
8	Waktu Simulasi	300 Detik

3.5 PERANCANGAN TOPOLOGI

Perancangan topologi dalam penelitian ini menggunakan jenis topologi dumb – bell karena topologi ini pada umumnya dapat digunakan untuk mengamati, mempelajari terkait efek penyempitan bandwith yang diatur pada 2 jalur pendistribusian paket yaitu jalur pengiriman paket dan jalur penerimaan paket yang dapat dilihat pada gambar 3.3.



Gambar 3. 3 Perancangan Topologi

3.6 VARIABLE YANG DIHITUNG

Variable yang dihitung dalam studi kasus ini 4 yaitu *Packet Delivery Ratio*, *Throughput*, *End to End Delay* dan *Packet Drop* dengan penjelasan seperti berikut:

1. *Packet Delivery Ratio*

Packet Delivery Ratio merupakan parameter yang digunakan untuk menghitung presentase keberhasilan tingkat paket diterima oleh TCP receiver dengan menggunakan perhitungan sebagai berikut:

$$PDR = \frac{\text{Paket Diterima}}{\text{Paket Dikirim}} \times 100\%$$

2. *Throughput*

Secara umum, *Throughput* merupakan nilai banyaknya jumlah paket yang diterima oleh TCP Reciver dalam waktu simulasi yang sudah ditentukan. Sehingga dapat dilakukan perhitungan dengan menggunakan cara seperti berikut:

$$\text{Throughput} = \frac{\text{Paket Diterima}}{\text{Waktu Pengiriman}} \times \text{Ukuran}$$

3. *End to End Delay*

End to End Delay merupakan waktu yang diperlukan sebuah paket dalam pendistribusiannya dari TCP sender hingga dapat diterima oleh TCP receiver sebagai tujuannya. Dengan melakukan cara perhitungan seperti berikut:

$$\text{Delay} = \frac{\text{Waktu Diterima} - \text{Waktu Dikirim}}{\text{Jumlah Paket Dikirim}}$$

4. *Packet Drop*

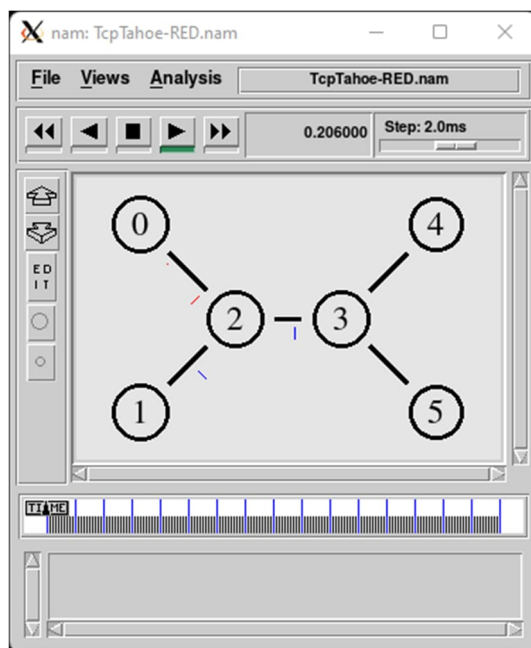
Packet Drop merupakan paket yang hilang ketika proses pengiriman atau pendistribusian paket yang dilakukan oleh TCP sender ke TCP receiver. Jumlah paket yang hilang tersebut dapat dihitung dengan menggunakan cara seperti berikut:

$$\text{Delay} = \text{Paket Dikirim} - \text{Paket Diterima}$$

3.7 OUTPUT NAM DAN AWK

Pengujian berikut menghasilkan dua buah output diantaranya Output Simulasi jaringan komputer dalam bentuk Nam dan file trace yang dihitung menggunakan script awk pada masing masing model antrian yaitu RED dan Droptail.

1. Output Nam Tcp Tahoe antrian RED



Gambar 3. 4 Output Nam RED

2. Output Trace Tcp Tahoe antrian RED

```

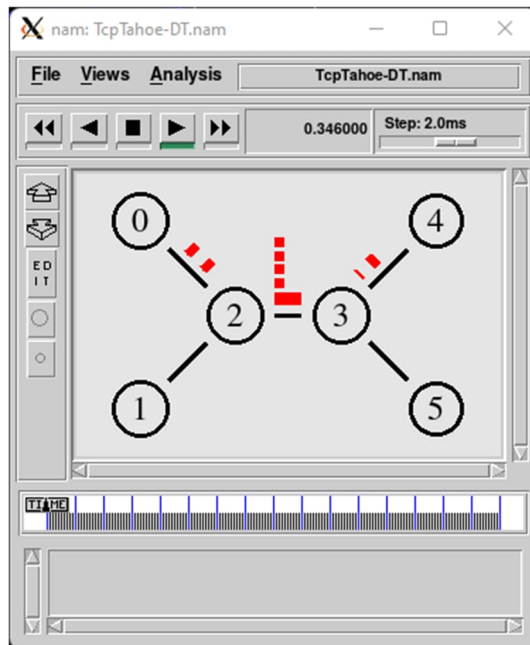
xloyd@Xloyd-PC: ~/NS2-Program/TcpTahoe/RED
xloyd@Xloyd-PC:~/NS2-Program/TcpTahoe/RED$ ls
TcpTahoe-RED.nam  TcpTahoe-RED.tcl  TcpTahoe-RED.tr  all.q  openAwk.awk  temp.a  temp.q  temp.queue
xloyd@Xloyd-PC:~/NS2-Program/TcpTahoe/RED$ gawk -f openAwk.awk TcpTahoe-RED.tr

-----
Throughput TCP = 254.216 Mbit
Avg throughput TCP = 0.508432 Mbps
-----
Packet Delivery Ratio = 98.8675%
-----
TCP Sent Packet = 32407
TCP Packet Drop = 367
-----
Avg end to end delay = 0.0520787 s
-----
xloyd@Xloyd-PC:~/NS2-Program/TcpTahoe/RED$

```

Gambar 3. 5 Output Trace RED

3. Output Nam Tcp Tahoe antrian Droptail



Gambar 3. 6 Output Nam Droptail

4. Output Trace Tcp Tahoe antrian Droptail

```

xloyd@Xloyd-PC: ~/NS2-Program/TcpTahoe/Droptail
xloyd@Xloyd-PC:~/NS2-Program/TcpTahoe/Droptail$ ls
TcpTahoe-DT.nam  TcpTahoe-DT.tr  TcpTahoe-Droptail.tcl  openAwk.awk  queue.tr
xloyd@Xloyd-PC:~/NS2-Program/TcpTahoe/Droptail$ gawk -f openAwk.awk TcpTahoe-DT.tr

-----
Throughput TCP = 273.386 Mbit
Avg throughput TCP = 0.546772 Mbps
-----
Packet Delivery Ratio = 99.2711%
-----
TCP Sent Packet = 34710
TCP Packet Drop = 253
-----
Avg end to end delay = 0.0799484 s
-----

```

Gambar 3. 7 Output Trace Droptail

LAMPIRAN SCRIPT

```
Test.tcl

set val(stop) 10.0 ; #time of simulation end
#Create a ns simulator
set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red

#Open the NS trace file
set tracefile [open p1.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open p1.nam w]
$ns namtrace-all $namfile


#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create labels for nodes
$n0 label "Tcp Source"
$n3 label "Tcp_Destination"
$n1 label "UdpSource "
$n2 label "Udp Destination"

#Give shapes to nodes
$n0 shape square
$n3 shape square
$n4 shape circle
$n1 shape hexagon
$n2 shape hexagon
#Give colors to nodes
$n0 color green
$n3 color red
$n1 color green
$n2 color red
$n4 color black

#Create links between nodes
$ns duplex-link $n0 $n4 100.0Mb 40ms DropTail
$ns queue-limit $n0 $n4 5 ; # default queue limit is 50

$ns duplex-link $n4 $n3 100.0Mb 40ms DropTail
$ns queue-limit $n4 $n3 5

$ns duplex-link $n1 $n4 100.0Mb 40ms DropTail
```

```

$ns queue-limit $n1 $n4 5

$ns duplex-link $n4 $n2 100.0Mb 40ms DropTail
$ns queue-limit $n4 $n2 5

$ns duplex-link-op $n4 $n2 queuePos 0.5 ;# give reading when queue content is 50%
$ns duplex-link-op $n4 $n0 queuePos 0.5

#Give node position (for NAM)
$ns duplex-link-op $n4 $n0 orient left-down
$ns duplex-link-op $n1 $n4 orient left-up
$ns duplex-link-op $n3 $n4 orient left-down
$ns duplex-link-op $n2 $n4 orient right-down

#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp0 $sink3
$tcp0 set packetSize_ 1000

#Setup a UDP connection
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null2 [new Agent/Null]
$ns attach-agent $n2 $null2
$ns connect $udp1 $null2
$udp1 set packetSize_ 1000

#Assign flow-id
$tcp0 set fid_ 1
$udp1 set fid_ 2

#Setup a CBR Application over TCP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $tcp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 3.0Mb
$cbr0 set random_ null
$ns at 0.01 "$cbr0 start"
$ns at 9.9 "$cbr0 stop"

#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 1000
$cbr1 set rate_ 2.0Mb
$cbr1 set random_ null
$ns at 0.1 "$cbr1 start"
$ns at 9.0 "$cbr1 stop"

#Define a 'finish' procedure

```

```

proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam p1.nam &
    exit 0
}

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

```

TcpTahoe-RED.tcl

set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
set nf [open TcpTahoe-RED.nam w]
$ns namtrace-all $nf
set nd [open TcpTahoe-RED.tr w]
$ns trace-all $nd
#cwndtcp1
#$ns namtrace-all $nf
#set cwnd1 [open cwndsk1_1.tr w]
#cwndtcp2
#$ns namtrace-all $nf
#set cwnd2 [open cwndsk1_2.tr w]
#*****
#prosedur "finish"
proc finish {} {
    global tchan_
    set awkCode {
        {
            if ($1 == "Q" && NF>2) {
                print $2, $3 >> "temp.q";
                set end $2
            }
            else if ($1 == "a" && NF>2)
                print $2, $3 >> "temp.a";
        } }
    set f [open temp.queue w]
    puts $f "TitleText: red"
    puts $f "Device: Postscript"
    if { [info exists tchan_] } {
        close $tchan_
    }
    exec rm -f temp.q temp.a
    exec touch temp.a temp.q
    exec awk $awkCode all.q
    puts $f "\"queue"
    exec cat temp.q >@ $f

```

```

puts $f \n\"ave_queue
exec cat temp.a >@ $f
close $f
#exec xgraph -bb -bg white -tk -x time -y queue temp.queue &
exit 0
}
#*****
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
#kapasitas link dan mekanisme antrian
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms RED
$ns duplex-link $n3 $n4 5Mb 10ms DropTail
$ns duplex-link $n3 $n5 5Mb 10ms DropTail
#orientasi node
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
#batasan jumlah ruang antrian
$ns queue-limit $n2 $n3 10
$ns duplex-link-op $n2 $n3 queuePos 0.5
#setting antrian RED
set redq [$ns link $n2 $n3] queue
set tchan_ [open all.q w]
$redq set bytes_ false
$redq set queue_in_bytes_ false
$redq trace curq_
$redq trace ave_
$redq attach $tchan_
$redq set thresh_ 10
$redq set maxthresh_ 60
$redq set q_weight_ 0.002
$redq set linterm_ 10
#Agent TCP dan aplikasi FTP
#TCP1
set tcp [new Agent/TCP]
$tcp set class_ 1
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$tcp set window_ 1000
#tcp trace
$tcp attach $nd
$tcp tracevar cwnd_

```



```

#TCP2
set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
$ns attach-agent $n1 $tcp2
set sink2 [new Agent/TCPSink]
$ns attach-agent $n5 $sink2
$ns connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set type_ FTP
$tcp2 set window_ 1000
#tcp trace
$tcp2 attach $nd
$tcp2 tracevar cwnd_
#Pengeplotan data cwnd dengan file akhir .xg
proc plotWindow {tcpSource outfile} {
    global ns
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $outfile "$now $cwnd"
    $ns at [expr $now+0.1] "plotWindow $tcpSource $outfile"
}
#set outfile [open "cwndsk1_tcp1.xg" w]
#$ns at 0.0 "plotWindow $tcp $outfile"
#set outfile [open "cwndsk1_tcp2.xg" w]
#$ns at 0.0 "plotWindow $tcp2 $outfile"
#*****
$ns at 0.1 "$ftp start"
$ns at 0.1 "$ftp2 start"
$ns at 300.1 "$ftp stop"
$ns at 500.1 "$ftp2 stop"
$ns at 502.0 "finish"
$ns run

```

TcpTahoe-Droptail.tcl

```

set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
set nf [open TcpTahoe-DT.nam w]
$ns namtrace-all $nf
set nd [open TcpTahoe-DT.tr w]
$ns trace-all $nd
#cwndtcp1
#$ns namtrace-all $nf
#set cwnd1 [open cwnd1.tr w]
#cwndtcp2
#$ns namtrace-all $nf
#set cwnd2 [open cwnd2.tr w]
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exit 0
}

```

```

}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
#kapasitas link dan parameter
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail
$ns duplex-link $n3 $n4 5Mb 10ms DropTail
$ns duplex-link $n3 $n5 5Mb 10ms DropTail
#orientasi/letak node
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
#limitasi queue
$ns queue-limit $n2 $n3 20
$ns duplex-link-op $n2 $n3 queuePos 0.5
#TCP 1 Source agent dan aplikasi || sink
set tcp [new Agent/TCP]
$tcp set class_ 1
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$tcp set window_ 1000
#tcp trace
$tcp attach $nd
$tcp tracevar cwnd_
#TCP 2 Source agent dan aplikasi || sink
set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
$ns attach-agent $n1 $tcp2
set sink2 [new Agent/TCPSink]
$ns attach-agent $n5 $sink2
$ns connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set type_ FTP
$tcp2 set window_ 1000
#tcp trace
$tcp2 attach $nd
$tcp2 tracevar cwnd_
proc plotWindow {tcpSource outfile} {
global ns
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $outfile "$now $cwnd"

```

```

$ns at [expr $now+0.1] "plotWindow $tcpSource $outfile"
}
#PLOT WINDOW DEGAN XGRAPH
#set outfile [open "cwndtcp1.xg" w]
#$ns at 0.0 "plotWindow $tcp $outfile"
#set outfile [open "cwndtcp2.xg" w]
#$ns at 0.0 "plotWindow $tcp2 $outfile"
set qfile [$ns monitor-queue $n2 $n3 [open queue.tr w] 0.031131312]
[$ns link $n2 $n3] queue-sample-timeout;
$ns at 0.1 "$ftp start"
$ns at 5.0 "$ftp2 start"
$ns at 300.0 "$ftp stop"
$ns at 300 "$ftp2 stop"
$ns at 300.1 "finish"
$ns run

```

openAwk.awk

```

BEGIN {
countcwnd = 0
sendsize = 0
recvdSize = 0
startTime = 0.1
stopTime = 200
num_sample = 0;
total_delay = 0;
sent_1 = 0;
drop_tcp1 = 0;
tcp1 = 0;
}
{
event = $1
time = $2
Node_id = $3
layer = $4
flags = $5
sequence_number = $6
cwnd = $6
cwndvalue = $7
packet_type = $7
packet_size = $8;
timecwnd = $1
fromNode = $3
toNode = $4
#avgcwnd
if ($2=="0" && $4=="4" && $6=="cwnd_" && $7=="1.000") {
css1=css1+1;
}
if ($1=="+" && $3=="0"){
sizetcp1=sizetcp1+$6;
}
if ($2=="1" && $4=="5" && $6=="cwnd_" && $7=="1.000") {
css2=css2+1;
}
}

```

```

if ($1 == "+" && $3 == "1"){
sizetcp2=sizetcp2+$6;
}
#throughput
if ($1=="r" && $4==4){
tcp1=tcp1+$6; }
#packetsent
if (Node_id == 0 && layer == 2 && event == "+")
sent_1++;
#packetdrop
if (packet_size ==1 && event == "d")
drop_tcp1++;
#e2edelay
if (event == "+" && time >=0.1 && Node_id == 0 &&
flags=="tcp" && packet_size==1){
p_id=$12
t_dep[$12] = time
}
if (event == "r" && time >=0.1 && layer == 4 && flags == "tcp" && packet_size==1){
if (p_id = $12){
t_arr[$12] = time
num_sample++
delay = t_arr[$12] - t_dep[$12]
total_delay += delay
} }
}
END {
print "\n-----"
print " Throughput TCP = " (tcp1*8/1024/1024) " Mbit";
print " Avg throughput TCP = " tcp1*8/1024/1024/500 " Mbps";
#print" Total cwnd = " sizetcp1/1024" kB"
#print" Avg Byte per cwnd = "(sizetcp1/1024)/css1 " KB/cwnd"
print "-----"
print " Packet Delivery Ratio = " ((sent_1-drop_tcp1)/sent_1)*100 "%";
print "-----"
print " TCP Sent Packet = " (sent_1)
print " TCP Packet Drop = " (drop_tcp1)
avg_delay = total_delay/num_sample;
print "-----"
print " Avg end to end delay = " avg_delay " s";
print "-----\n"
}

```

Q AND A



Langkah penyelesaian problem tersebut:

Buat yang ngalamin kendala serupa bisa ikutin langkah berikut:

1. Cek versi wslnya di windows powershell dengan perintah **wsl -l -v**
2. Apabila versi yang tampil adalah versi 2 maka downgrade ke versi 1 karena versi 2 belum mendukung untuk sharing display itu sebabnya koneksi selalu ditolak, gunakan perintah **wsl --set-version Ubuntu 1**
3. Buka ulang wsl dan pastikan xming running di port 0.0
4. Ketik perintah **export DISPLAY=:0.0**
5. Buka ulang gedit [namafilename.tcl]
6. Selesai