

1ère Année Cycle Supérieur (1CS)

Projet Réseau II

Partie 01:

Evaluation des performances d'un réseau informatique

Equipe n° = 02:

- LAOUCHEDI Karim (G-3)
- ABOUD Rayane (G-4)
- BOUDAUD Sid Ahmed (G-2)
- NAILI Noufel (G-8)
- BELLI Bilal (G-5)

I. Table de Matiere:

Table de Matiere	2
PARTIE 01	3
Introduction	3
Configuration de la Topologie	3
PARTIE DÉCOUPÉE SOUS RÉSEAU	3
Choix du réseau	3
Schéma du réseau	3
Explication du découpage	4
Explication de la réalisation	4
TEST DE LA CONFIGURATION	6
TEST 1	6
TEST 2	6
TEST 3	7
TEST 4	7
TABLES DE ROUTAGE	8
Série Test 1	10
TEST 1.1	10
TEST 1.2	11
TEST 1.3	12
Graph	13
Explication	14
TEST 2.1	14
TEST 2.2	16
TEST 2.3	17
Graph	18
Explication	18
TEST 3.1	19
TEST 3.2	20
TEST 3.3	21
Graph	22
Explication	22
TEST 4	23
Graph	24
Explication	24
Synthèse globale	25
Conclusion	26
Ressources et références utilisées	26

II. PARTIE 01:

A. Introduction:

L'architecture d'internet, ainsi que l'ensemble des réseaux existants, étant une des plus complexe au monde il est bien important de connaître l'ampleur de ses performances vis à vis des différentes liaisons constituant ce réseau énorme, et ceci dans le but d'essayer de quantifier les coût et délais pouvant caractériser les différents réseaux pour ainsi avoir une optique précise de ce que un prestataire de ces services peut offrir à ses clients.

Et pour obtenir cette mesure de caractéristique nous effectuerons un éventail de tests destinés à mettre les équipements d'un sous réseau sous stress et comparer les valeurs obtenus aux valeurs attendues (valeurs idéales).

Dans ce qui suit nous allons nous consacrer à effectuer cette batterie de tests dans une topologie réseaux simplifiée afin de faire apparaître au mieux possible l'effet de différents paramètres tels que: la perte de paquets, la latence et le débit.

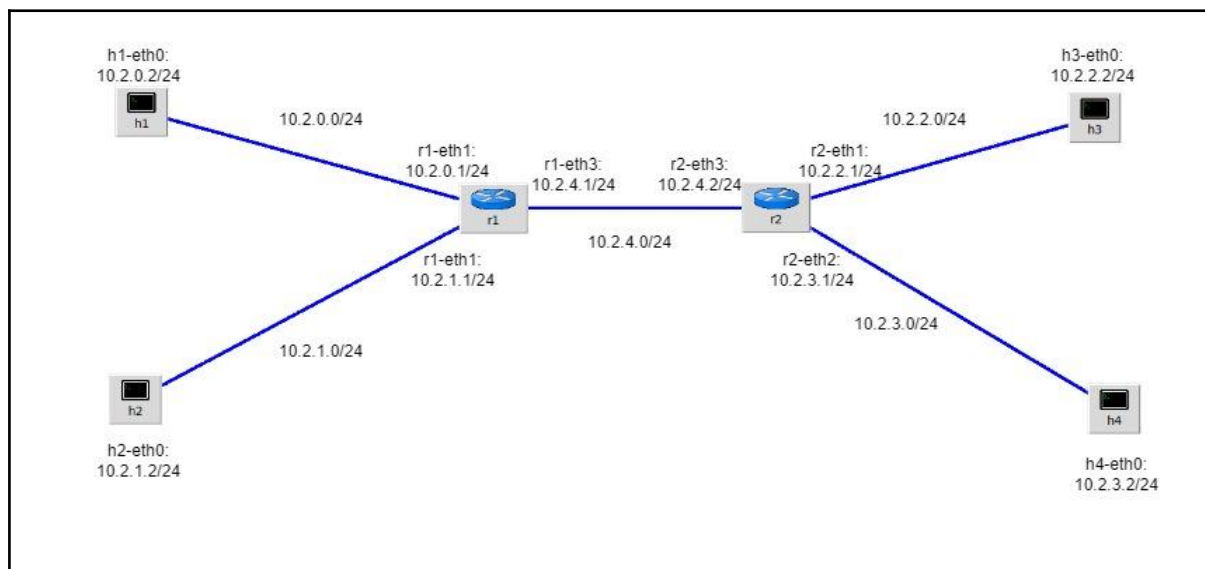
B. Configuration de la Topologie:

PARTIE DÉCOUPÉE SOUS RÉSEAU:

Choix du réseau :

Étant demandé dans l'énoncé de prendre des adresses réseau 10.x.0.0/16 avec x représentant le numéro de l'équipe, donc notre adresse réseau utilisée est 10.2.0.0/16.

Schéma du réseau :



Explication du découpage :

Dans la topologie au dessus les machines h1, h2, h3 et h4 sont des machines appartenant aux 4 sous-réseaux différents connectés aux interfaces des routeurs r1, r2.

L'adressage des différents sous-réseaux et interfaces est illustré dans le tableau suivant :

Equipement	Interface	Adresse IP de l'interface
h1	h1-eth0	10.2.0.2/24
h2	h2-eth0	10.2.1.2/24
h3	h3-eth0	10.2.2.2/24
h4	h4-eth0	10.2.3.2/24
r1	r1-eth1	10.2.0.1/24
	r1-eth2	10.2.1.1/24
	r1-eth3	10.2.4.1/24
r2	r2-eth1	10.2.2.1/24
	r2-eth2	10.2.3.1/24
	r2-eth3	10.2.4.2/24

Explication de la réalisation :

Pour effectuer la configuration citée dans ce qui précède (et à cause de défaillances que présente le logiciel), nous avons décidé de le faire à travers l'API qu'offre Mininet pour le langage python.

Dans ce qui suit nous allons expliquer de manière détaillé comment nous avons effectué l'étape de configuration au moyen de ce script :

```

class NetworkTopo( Topo ):
    def __init__( self, ):
        Topo.__init__( self )

        defaultIp='10.2.'
        defaultMask='/16'
        subnetsMask='/24'

        h1 = self.addHost('h1',ip=defaultIp+'0.2'+subnetsMask, defaultRoute='via ' + defaultIp+'0.1')
        h2 = self.addHost('h2',ip=defaultIp+'1.2'+subnetsMask, defaultRoute='via ' + defaultIp+'1.1')
        h3 = self.addHost('h3',ip=defaultIp+'2.2'+subnetsMask, defaultRoute='via ' + defaultIp+'2.1')
        h4 = self.addHost('h4',ip=defaultIp+'3.2'+subnetsMask, defaultRoute='via ' + defaultIp+'3.1')

        r1 = self.addHost('r1',cls=LinuxRouter, ip=defaultIp+'0.1'+subnetsMask)
        r2 = self.addHost('r2',cls=LinuxRouter, ip=defaultIp+'2.1'+subnetsMask)

        # h1 r1 and h2 r1 links
        self.addLink(h1,r1,intfName2='r1-eth1', params2={'ip': defaultIp+'0.1' +subnetsMask}) # 'ip': '10.2.0.1/24'
        self.addLink(h2,r1,intfName2='r1-eth2', params2={'ip': defaultIp+'1.1' +subnetsMask})

        # h3 r2 and h4 r2 links
        self.addLink(h3,r2,intfName2='r2-eth1', params2={'ip': defaultIp+'2.1' +subnetsMask})
        self.addLink(h4,r2,intfName2='r2-eth2',params2={'ip': defaultIp+'3.1' +subnetsMask})

        # r1 r2 link
        self.addLink(r1,r2,intfName1='r1-eth3',params1={'ip':defaultIp+'4.1'+subnetsMask},intfName2='r2-eth3',params2={'ip':
defaultIp+'4.2'+subnetsMask})

```

Dans la figure 1, nous effectuons par le biais d'une classe la création de notre topologie ainsi que les différents sous réseaux et ceci à avec la commande “**addHost**”, en spécifiant l'adresse des différentes interfaces en paramètres.

Nous avons aussi, pour définir les liens entre les machines utilisé la fonction **addLink** en spécifiant les machines, les interfaces concernées et optionnellement les adresses ip des interfaces si il y a besoin de les créer.

```

def run():

    topologie = NetworkTopo()
    net= Mininet(topo=topologie,link=TCLink)
    net.start()

    r1=net['r1']
    r2=net['r2']
    h1=net['h1']
    h2=net['h2']
    h3=net['h3']
    h4=net['h4']

    info(r1.cmd('ip route add 10.2.2.0/24 via 10.2.4.2 dev r1-eth3'))
    info(r1.cmd('ip route add 10.2.3.0/24 via 10.2.4.2 dev r1-eth3'))

    info(r2.cmd('ip route add 10.2.0.0/24 via 10.2.4.1 dev r2-eth3'))
    info(r2.cmd('ip route add 10.2.1.0/24 via 10.2.4.1 dev r2-eth3'))

    # change link settings for both ports
    r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit')
    r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit')

    r2.cmd('sudo tc qdisc add dev r2-eth2 root netem rate 5gbit')
    r1.cmd('sudo tc qdisc add dev r1-eth2 root netem rate 5gbit')

    r2.cmd('sudo tc qdisc add dev r2-eth1 root netem rate 5gbit')
    r1.cmd('sudo tc qdisc add dev r1-eth1 root netem rate 5gbit')

    h1.cmd('sudo tc qdisc add dev r1-eth0 root netem rate 5gbit')
    h2.cmd('sudo tc qdisc add dev r1-eth0 root netem rate 5gbit')
    h3.cmd('sudo tc qdisc add dev r1-eth0 root netem rate 5gbit')
    h4.cmd('sudo tc qdisc add dev r1-eth0 root netem rate 5gbit')

    CLI(net)

```

Dans la figure 2 nous créons la topologie à l'aide du constructeur de la classe mininet depuis la topologie définie dans la figure précédente, puis nous ajoutons les routes au tables du routage des routeurs r1 et r2 à l'aide de la commande “**ip route add**” (une route pour chacun des sous réseaux), nous exécutons ensuite la commande tc sur chacune des interfaces des routeurs (r1-eth3 et r2-eth3) pour spécifier le débit de lien entre r1 et r2.

TEST DE LA CONFIGURATION:

Afin d'assurer un bon fonctionnement des configurations implémentés on a prévu de faire un plan de test qui nous a amené à vérifier notre topologie.

Si la configuration fonctionne parfaitement on aura :

1. Le succès d'envoyer des paquets entre les machines du même réseau.
2. Les mêmes valeurs configurées.

TEST CONFIGURATION 1:

Commande utilisée
h3 ping h1 -c4
Résultat
<pre>mininet> h3 ping h1 -c4 PING 10.2.0.2 (10.2.0.2) 56(84) bytes of data. 64 bytes from 10.2.0.2: icmp_seq=1 ttl=62 time=0.080 ms 64 bytes from 10.2.0.2: icmp_seq=2 ttl=62 time=0.074 ms 64 bytes from 10.2.0.2: icmp_seq=3 ttl=62 time=0.076 ms 64 bytes from 10.2.0.2: icmp_seq=4 ttl=62 time=0.076 ms --- 10.2.0.2 ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3059 ms rtt min/avg/max/mdev = 0.074/0.076/0.080/0.002 ms</pre>
Explication
<ul style="list-style-type: none">• la commande envoie des paquets de test de connectivité de la source (h3) vers sa destination (h1), le paquet est envoyé avec succès car les routeurs peuvent acheminer correctement les destinations.

TEST CONFIGURATION 2:

Commande utilisée
h4 ping h2 -c4
Résultat

```
mininet> h4 ping h2 -c4
PING 10.2.1.2 (10.2.1.2) 56(84) bytes of data.
64 bytes from 10.2.1.2: icmp_seq=1 ttl=62 time=0.075 ms
64 bytes from 10.2.1.2: icmp_seq=2 ttl=62 time=0.076 ms
64 bytes from 10.2.1.2: icmp_seq=3 ttl=62 time=0.081 ms
64 bytes from 10.2.1.2: icmp_seq=4 ttl=62 time=0.105 ms

--- 10.2.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063 ms
rtt min/avg/max/mdev = 0.075/0.084/0.105/0.012 ms
```

Explication

- la commande envoie des paquets de test de connectivité de la source (**h4**) vers sa destination (**h2**), le paquet est envoyé avec succès car les routeurs peuvent acheminer correctement les destinations.

TEST CONFIGURATION 3:

Commande utiliser

h3 traceroute h1

Résultat

```
mininet> h3 traceroute h1
traceroute to 10.2.0.2 (10.2.0.2), 30 hops max, 60 byte packets
 1  _gateway (10.2.2.1)  0.946 ms  0.858 ms  0.819 ms
 2  10.2.4.1 (10.2.4.1)  0.785 ms  0.720 ms  0.677 ms
 3  10.2.0.2 (10.2.0.2)  0.628 ms  0.567 ms  0.511 ms
```

Explication

- La commande trace le chemin suivi par le paquet de sa source (**h3**) vers sa destination (**h1**), il passe premièrement par sa passerelle par défaut du (**r2**) et après vers le prochain saut vers (**r1**) et enfin vers la machine destinée.

TEST 4:

Commande utilisée

h4 traceroute h2

Résultat			
<pre>mininet> h4 traceroute h2 traceroute to 10.2.1.2 (10.2.1.2), 30 hops max, 60 byte packets 1 _gateway (10.2.3.1) 0.092 ms 0.031 ms 0.024 ms 2 10.2.4.1 (10.2.4.1) 0.067 ms 0.052 ms 0.042 ms 3 10.2.1.2 (10.2.1.2) 0.068 ms 0.055 ms 0.053 ms</pre>			
Explication			
<ul style="list-style-type: none"> La commande trace le chemin suivi par le paquet de sa source (h4) vers sa destination (h2), il passe premièrement par sa passerelle par défaut du (r1) et après vers le prochain saut vers (r2) et enfin vers la machine destinée. 			

TABLES DE ROUTAGE:

Routeur: r1

Commande utilisée			
r1 route			
Résultat			
<pre>mininet> r1 route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface 10.2.0.0 0.0.0.0 255.255.255.0 U 0 0 0 r1-eth1 10.2.1.0 0.0.0.0 255.255.255.0 U 0 0 0 r1-eth2 10.2.2.0 10.2.4.2 255.255.255.0 UG 0 0 0 r1-eth3 10.2.3.0 10.2.4.2 255.255.255.0 UG 0 0 0 r1-eth3 10.2.4.0 0.0.0.0 255.255.255.0 U 0 0 0 r1-eth3</pre>			

Pour atteindre le réseau de destination	Prochain nœud passerelle	Via l'interface
10.2.0.0/24	Direct	r1-eth1
10.2.1.0/24	Direct	r1-eth2
10.2.2.0/24	10.2.4.2/24	r1-eth3
10.2.3.0/24	10.2.4.2/24	r1-eth3
10.2.4.0/24	Direct	r1-eth3

Routeur: r2

Commande utilisée
r2 route

Résultat							
<pre>mininet> r2 route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface 10.2.0.0 10.2.4.1 255.255.255.0 UG 0 0 0 r2-eth3 10.2.1.0 10.2.4.1 255.255.255.0 UG 0 0 0 r2-eth3 10.2.2.0 0.0.0.0 255.255.255.0 U 0 0 0 r2-eth1 10.2.3.0 0.0.0.0 255.255.255.0 U 0 0 0 r2-eth2 10.2.4.0 0.0.0.0 255.255.255.0 U 0 0 0 r2-eth3</pre>							

Pour atteindre le réseau de destination	Prochain nœud passerelle	Via l'interface
10.2.0.0/24	10.2.4.1/24	r2-eth3
10.2.1.0/24	10.2.4.1/24	r2-eth3
10.2.2.0/24	Direct	r2-eth1
10.2.3.0/24	Direct	r2-eth2
10.2.4.0/24	Direct	r2-eth3

Remarques :

Les tests effectués sont sur le système d'exploitation linux, donc les commandes sont spécifiques à cet OS.

Les valeurs des paramètres utilisés:

	DF	DL	LA	TP	TD	WIN	DW	LA_TCP	TP_TCP	TP_UDP	LA_UDP
E2	5	10	5	2	2	16	4	10	2	3	15

	Valeur faible	Valeur moyenne	Valeur élevée
T1 (Débit)	0,5	2,5	5
T2 (LA)	5 ms	25 ms	50 ms
T3 (TP)	2	8	16
T4	DF = 2,5 ; LA = 5 ms ; TP = 2		

C. Série Test 1:

TEST 1.1:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 0.5gbit ')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 0.5gbit ')
```

Résultat

```
"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53342
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53344
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  1.00-2.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  2.00-3.00   sec    57.0 MBytes   479 Mbits/sec
[ 7]  3.00-4.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  4.00-5.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  5.00-6.00   sec    57.0 MBytes   479 Mbits/sec
[ 7]  6.00-7.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  7.00-8.00   sec    57.0 MBytes   478 Mbits/sec
[ 7]  8.00-9.00   sec    57.0 MBytes   479 Mbits/sec
[ 7]  9.00-10.00  sec    57.0 MBytes   478 Mbits/sec
[ 7] 10.00-10.14  sec    7.89 MBytes   477 Mbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.14  sec    578 MBytes   478 Mbits/sec
-----
Server listening on 5201
[ ]

"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53344 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec    68.1 MBytes   571 Mbits/sec    0   2.92 MBytes
[ 7]  1.00-2.00   sec    57.5 MBytes   482 Mbits/sec    0   5.76 MBytes
[ 7]  2.00-3.00   sec    56.2 MBytes   472 Mbits/sec    0   8.36 MBytes
[ 7]  3.00-4.00   sec    57.5 MBytes   482 Mbits/sec    0   8.36 MBytes
[ 7]  4.00-5.00   sec    56.2 MBytes   472 Mbits/sec    0   8.36 MBytes
[ 7]  5.00-6.00   sec    57.5 MBytes   482 Mbits/sec    0   8.36 MBytes
[ 7]  6.00-7.00   sec    57.5 MBytes   482 Mbits/sec    0   8.36 MBytes
[ 7]  7.00-8.00   sec    56.2 MBytes   472 Mbits/sec    0   8.36 MBytes
[ 7]  8.00-9.00   sec    57.5 MBytes   482 Mbits/sec    0   8.36 MBytes
[ 7]  9.00-10.00  sec    56.2 MBytes   472 Mbits/sec    0   8.36 MBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec    581 MBytes   487 Mbits/sec    0
[ 7]  0.00-10.14  sec    578 MBytes   478 Mbits/sec
-----
iperf Done.
```

Observation

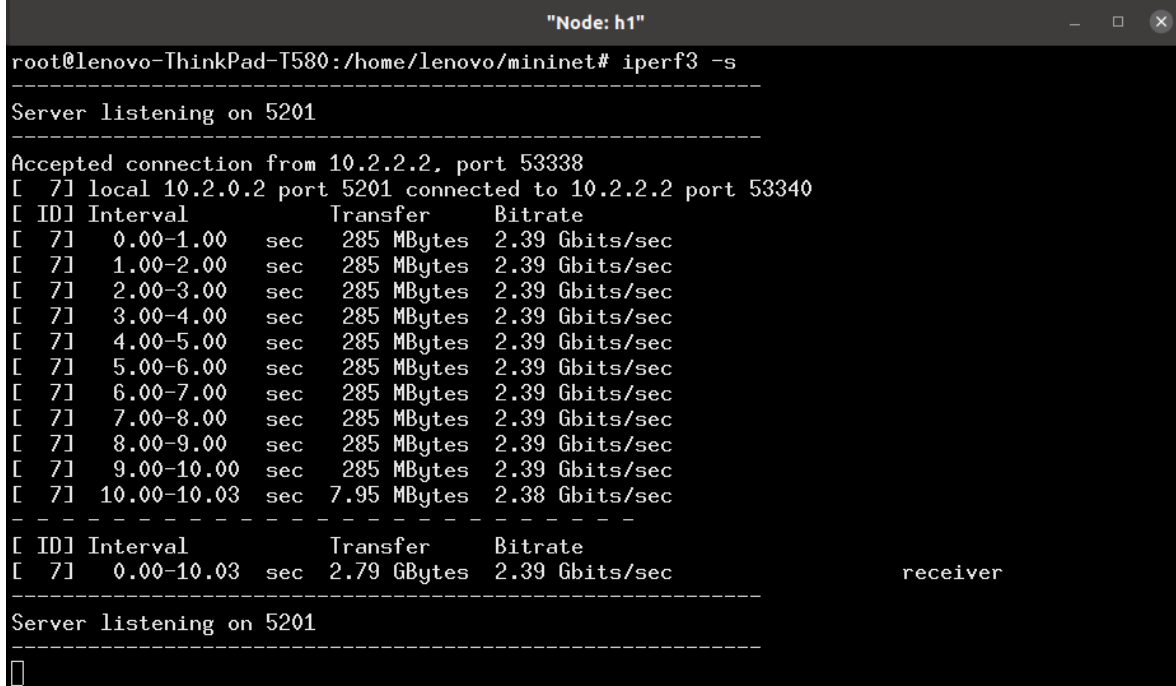
- On remarque que le débit “**bitrate**” reste très stable durant tout le long de la transmission (en moyenne 478 mb/s pour h3 et 487 mb/s pour h1) étant donné que nous avons plafonné le débit à 0.5 gb/s, on remarque une petite perte dû à d’autres aléas tel que la perte de paquets ou les acquittements.

TEST 1.2:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 2.5gbit ')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 2.5gbit ')
```

Résultat



```

"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53338
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53340
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  1.00-2.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  2.00-3.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  3.00-4.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  4.00-5.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  5.00-6.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  6.00-7.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  7.00-8.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  8.00-9.00   sec      285 MBytes  2.39 Gbits/sec
[ 7]  9.00-10.00  sec      285 MBytes  2.39 Gbits/sec
[ 7] 10.00-10.03  sec       7.95 MBytes 2.38 Gbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.03  sec       2.79 GBytes  2.39 Gbits/sec
-----
Server listening on 5201
-----

```

```

"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53340 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00      sec    296 MBytes  2.49 Gbits/sec    0   8.37 MBytes
[ 7]  1.00-2.00      sec    285 MBytes  2.39 Gbits/sec   32   8.37 MBytes
[ 7]  2.00-3.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  3.00-4.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  4.00-5.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  5.00-6.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  6.00-7.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  7.00-8.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  8.00-9.00      sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
[ 7]  9.00-10.00     sec    285 MBytes  2.39 Gbits/sec    0   8.37 MBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00      sec    2.79 GBytes  2.40 Gbits/sec   32
[ 7]  0.00-10.03      sec    2.79 GBytes  2.39 Gbits/sec
sender
receiver
iperf Done.

```

Observation

- On remarque que le débit “bitrate” reste très stable durant tout le long de la transmission (en moyenne 2.40 gb/s pour h3 et 2.39 gb/s pour h1) étant donné que nous avons plafonné le débit à 2,5 gb/s, on remarque une petite perte dû à d’autres aléas tel que la perte de paquets ou les acquittements.

TEST 1.3:

Commandes utilisées

```

# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit ')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit ')

```

Résultat

```

"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53334
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53336
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  1.00-2.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  2.00-3.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  3.00-4.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  4.00-5.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  5.00-6.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  6.00-7.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  7.00-8.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  8.00-9.00   sec    570 MBytes  4.78 Gbits/sec
[ 7]  9.00-10.00  sec    570 MBytes  4.78 Gbits/sec
[ 7] 10.00-10.01  sec    8.20 MBytes  4.76 Gbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.01  sec   5.57 GBytes  4.78 Gbits/sec
-----
Server listening on 5201

```

```

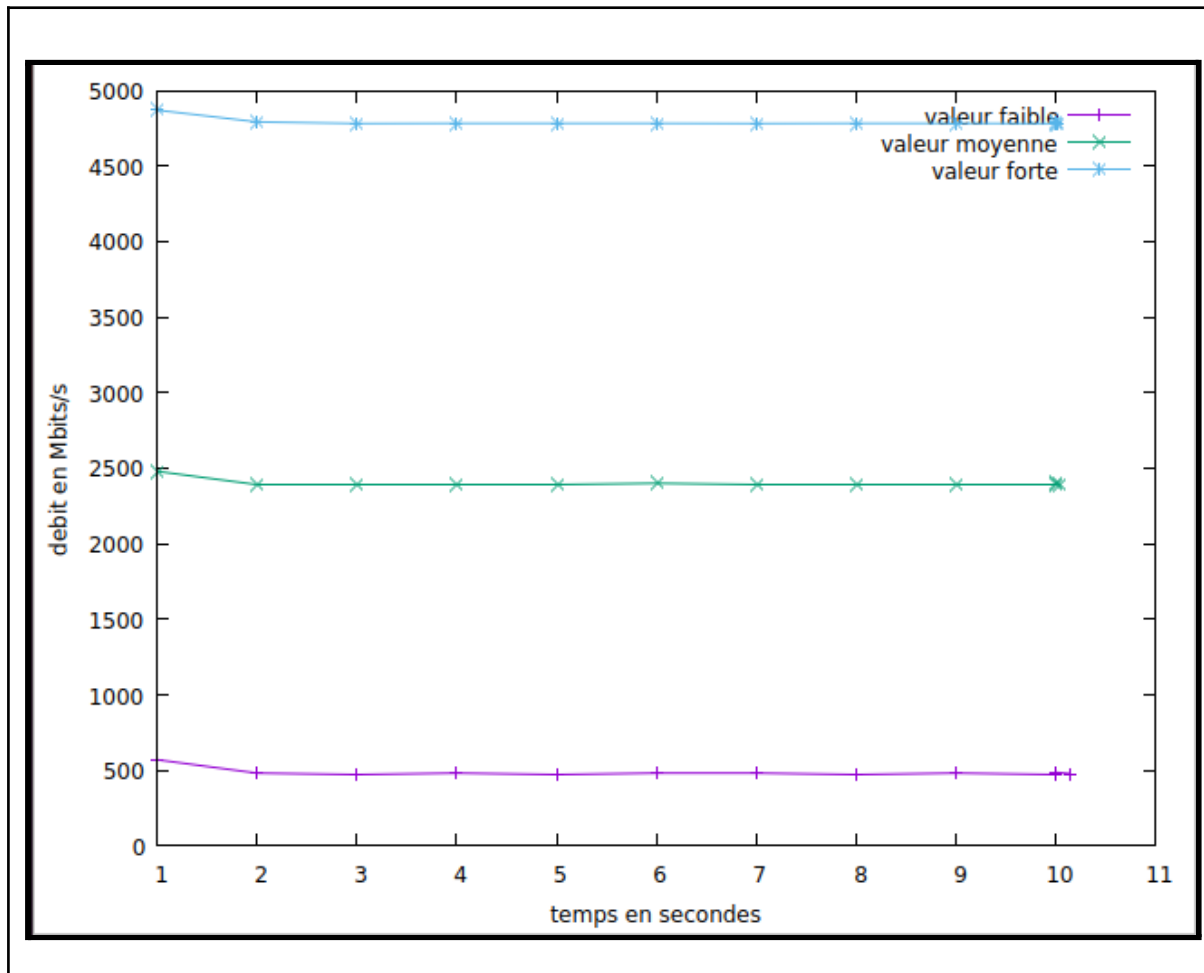
"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53336 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec    581 MBytes  4.87 Gbits/sec    0   8.37 MBytes
[ 7]  1.00-2.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  2.00-3.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  3.00-4.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  4.00-5.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  5.00-6.00   sec    570 MBytes  4.78 Gbits/sec   32   8.37 MBytes
[ 7]  6.00-7.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  7.00-8.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  8.00-9.00   sec    570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7]  9.00-10.00  sec    571 MBytes  4.79 Gbits/sec    0   8.37 MBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec   5.58 GBytes  4.79 Gbits/sec   32
[ 7]  0.00-10.01  sec   5.57 GBytes  4.78 Gbits/sec
-----
iperf Done.

```

Observation

- On remarque que le débit “bitrate” reste très stable durant tout le long de la transmission (en moyenne 4.79 gb/s pour h3 et 4,78 gb/s pour h1) étant donné que nous avons plafonné le débit à 5 gb/s, on remarque une petite perte dû à d’autres aléas tel que la perte de paquets ou les acquittements.

Graph:



Explication

Dans le graph ci-dessus, décrivant le débit applicatif du point de vu client(h3) lors de l'échange de packet entre h1 et h3, nous remarquons une constance dans les courbes des débits, de plus pour chacun des trois tracés nous constatons qu'il sont juste au-dessous des débits spécifiés dans les tests 1,1 , 1,2 et 1,3 (0,5 gb/s , 2,5 gb/s et 5 gb/s). Même si les autres liaisons entre h1-r1 et h3-r2 ont elles comme débit le débit maximale de 5 gb/s, mais le fait que les paquets passent forcément par la liaison r1-r2 les constraints à adopter le débit dans la liaison en question donc le débit applicatif entre h1 et h3 est réduit par le débit de la liaison r1-r2 ce qui fait que, dans le graph, pour chaque valeur de débit applicatif dans la liaison r1-r2 (faible, moyenne et élevée) nous retrouvons cette valeur même pour la liaison h1-h3.

T

TEST 2.1:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit delay 5ms')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit delay 5ms')
```

Résultat

```
"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53346
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53350
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-1.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 1.00-2.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 2.00-3.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 3.00-4.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 4.00-5.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 5.00-6.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 6.00-7.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 7.00-8.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 8.00-9.00      sec   570 MBytes  4.78 Gbits/sec
[ 7] 9.00-10.00     sec   570 MBytes  4.78 Gbits/sec
[ 7] 10.00-10.01    sec   7.95 MBytes  4.74 Gbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-10.01     sec   5.57 GBytes  4.78 Gbits/sec
-----
Server listening on 5201
[ ]

"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53350 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate    Retr  Cwnd
[ 7] 0.00-1.00      sec   581 MBytes  4.87 Gbits/sec    0   8.37 MBytes
[ 7] 1.00-2.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 2.00-3.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 3.00-4.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 4.00-5.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 5.00-6.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 6.00-7.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 7.00-8.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 8.00-9.00      sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
[ 7] 9.00-10.00     sec   570 MBytes  4.78 Gbits/sec    0   8.37 MBytes
-----
[ ID] Interval           Transfer     Bitrate    Retr
[ 7] 0.00-10.00     sec   5.58 GBytes  4.79 Gbits/sec    0
[ 7] 0.00-10.01     sec   5.57 GBytes  4.78 Gbits/sec
-----
iperf Done.
sender
receiver
```

Observation

- On remarque une baisse très légère du débit qui a été mis à 5 gb/s pour le lien entre r1 r2, et la nouvelle valeur est de moyenne de 4,78 gb/s, ceci est dû partiellement à

la latence qui à été introduite et qui est de 5ms, et aussi dû au fait qu'il peut y avoir des pertes de paquets et des acquittements.

TEST 2.2:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit delay 25ms')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit delay 25ms')
```

Résultat

```
"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53356
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53358
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00   sec    74.1 MBytes    622 Mbits/sec
[ 7]  1.00-2.00   sec    159 MBytes    1.33 Gbits/sec
[ 7]  2.00-3.00   sec    154 MBytes    1.29 Gbits/sec
[ 7]  3.00-4.00   sec    157 MBytes    1.32 Gbits/sec
[ 7]  4.00-5.00   sec    158 MBytes    1.33 Gbits/sec
[ 7]  5.00-6.00   sec    152 MBytes    1.28 Gbits/sec
[ 7]  6.00-7.00   sec    159 MBytes    1.33 Gbits/sec
[ 7]  7.00-8.00   sec    157 MBytes    1.32 Gbits/sec
[ 7]  8.00-9.00   sec    152 MBytes    1.27 Gbits/sec
[ 7]  9.00-10.00  sec    159 MBytes    1.33 Gbits/sec
[ 7] 10.00-10.05  sec    7.86 MBytes    1.31 Gbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.05  sec    1.46 GBytes    1.24 Gbits/sec
-----
Server listening on 5201
[

"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53358 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr      Cwnd
[ 7]  0.00-1.00   sec    84.6 MBytes    710 Mbits/sec    0    16.0 MBytes
[ 7]  1.00-2.00   sec    160 MBytes    1.34 Gbits/sec    0    16.0 MBytes
[ 7]  2.00-3.00   sec    154 MBytes    1.29 Gbits/sec   45    16.0 MBytes
[ 7]  3.00-4.00   sec    158 MBytes    1.32 Gbits/sec    0    16.0 MBytes
[ 7]  4.00-5.00   sec    158 MBytes    1.32 Gbits/sec    0    16.0 MBytes
[ 7]  5.00-6.00   sec    152 MBytes    1.28 Gbits/sec    0    16.0 MBytes
[ 7]  6.00-7.00   sec    159 MBytes    1.33 Gbits/sec    0    16.0 MBytes
[ 7]  7.00-8.00   sec    158 MBytes    1.32 Gbits/sec    0    16.0 MBytes
[ 7]  8.00-9.00   sec    152 MBytes    1.28 Gbits/sec    0    16.0 MBytes
[ 7]  9.00-10.00  sec    159 MBytes    1.33 Gbits/sec    0    16.0 MBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr      Cwnd
[ 7]  0.00-10.00  sec    1.46 GBytes    1.25 Gbits/sec   45    16.0 MBytes
[ 7]  0.00-10.05  sec    1.46 GBytes    1.24 Gbits/sec    0    16.0 MBytes
-----
iperf Done.
```

Observation

- On remarque baisse conséquente (drastique) du débit qui a été mis à 5 gb/s pour le lien entre r1 r2, et la nouvelle valeur est de moyenne de 1,24 gb/s, ce qui est énorme comme baisse, ceci est majoritairement dû à la latence qui est de 25 ms et qui cause la perte de paquets au cours de la transmission.

TEST 2.3:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit delay 50ms')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit delay 50ms')
```

Résultat

```
"Node: h1"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 53362
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 53364
[ ID] Interval           Transfer    Bitrate
[ 7]  0.00-1.00   sec    2.45 MBytes  20.6 Mbits/sec
[ 7]  1.00-2.00   sec   71.1 MBytes  596 Mbits/sec
[ 7]  2.00-3.00   sec   79.5 MBytes  667 Mbits/sec
[ 7]  3.00-4.00   sec   79.5 MBytes  667 Mbits/sec
[ 7]  4.00-5.00   sec   79.5 MBytes  667 Mbits/sec
[ 7]  5.00-6.00   sec   77.7 MBytes  652 Mbits/sec
[ 7]  6.00-7.00   sec   76.9 MBytes  645 Mbits/sec
[ 7]  7.00-8.00   sec   79.5 MBytes  667 Mbits/sec
[ 7]  8.00-9.00   sec   79.5 MBytes  667 Mbits/sec
[ 7]  9.00-10.00  sec   79.5 MBytes  667 Mbits/sec
[ 7] 10.00-10.10  sec    7.95 MBytes  666 Mbits/sec
-----
[ ID] Interval           Transfer    Bitrate
[ 7]  0.00-10.10  sec   713 MBytes  592 Mbits/sec
-----
Server listening on 5201
[ ]
```

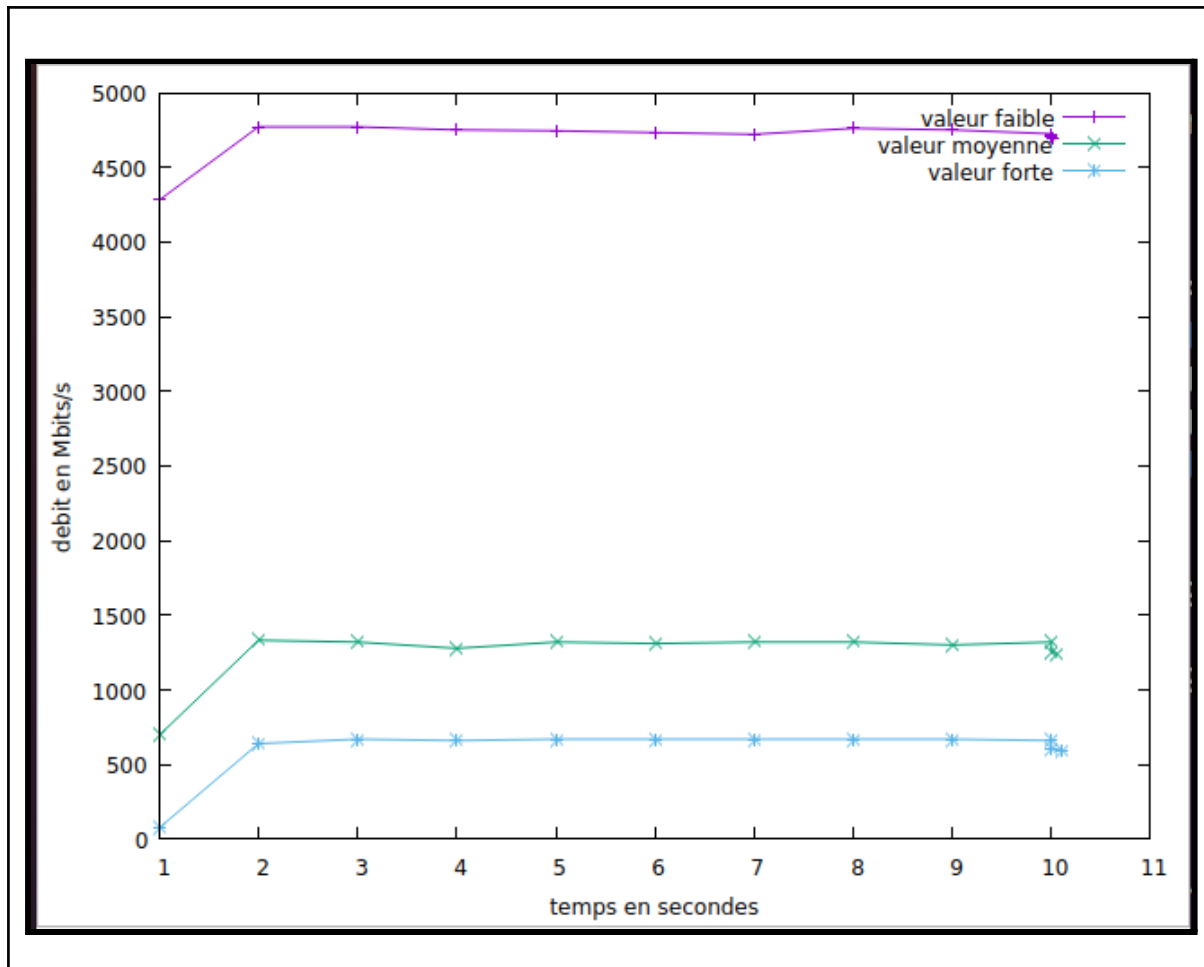
```
"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 53364 connected to 10.2.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 7]  0.00-1.00  sec   9.16 MBytes  76.8 Mbits/sec    0   2.45 MBytes
[ 7]  1.00-2.00  sec   75.0 MBytes  629 Mbits/sec    0  16.0 MBytes
[ 7]  2.00-3.00  sec   80.0 MBytes  671 Mbits/sec    0  16.0 MBytes
[ 7]  3.00-4.00  sec   80.0 MBytes  671 Mbits/sec   45  16.0 MBytes
[ 7]  4.00-5.00  sec   78.8 MBytes  661 Mbits/sec    0  16.0 MBytes
[ 7]  5.00-6.00  sec   77.5 MBytes  650 Mbits/sec    0  16.0 MBytes
[ 7]  6.00-7.00  sec   77.5 MBytes  650 Mbits/sec    0  16.0 MBytes
[ 7]  7.00-8.00  sec   80.0 MBytes  671 Mbits/sec    0  16.0 MBytes
[ 7]  8.00-9.00  sec   78.8 MBytes  661 Mbits/sec    0  16.0 MBytes
[ 7]  9.00-10.00 sec   80.0 MBytes  671 Mbits/sec    0  16.0 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 7]  0.00-10.00  sec   717 MBytes  601 Mbits/sec   45      sender
[ 7]  0.00-10.10  sec   713 MBytes  592 Mbits/sec           receiver

iperf Done.
```

Observation

- .On remarque une baisse conséquente (drastique) du débit qui a été mis à 5 gb/s pour le lien entre r1 r2, atteignant la moyenne de 592 mb/s pour h1 et 601 mb/s pour h3, ce qui est énorme comme baisse, et qui est aussi dû à la latence introduite et qui est de 50 ms.

Graph:



Explication :

Dans le graph ci dessus, représentant les tracés des variations des débits de la liaison h1-h3 pour différentes valeurs de latences appliquées au lien r1-r2. Nous remarquons que les différents tracés sont tous quasiment constant, sauf peut-être pour le début (moment de première connexion au serveur), et pour la même valeur de débit (5 gb/s) nous constatons une diminution accrue du débit applicatif dans la liaison h1-h3 au fur et à mesure que la latence du lien r1-r2 est augmentée à travers le test (les différentes valeurs faible, moyenne et élevée), ceci est dû au fait que tous les paquets qui sont transmis de h3 à h1 (ou vice versa), passent systématiquement par la liaison r1-r2. De plus l'impact de la latence sur le trafic réseau est majeur car elle empêche les machines de tirer pleinement profit du débit de la liaison en question, faisant ainsi ralentir le passage des paquets de h3 vers h1 et donc entraînant la diminution du débit.

TEST 3.1:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit loss 2%')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit loss 2%')
```

Résultat

```
"Node: h1"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -s
Server listening on 5201
Accepted connection from 10.2.2.2, port 56122
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 56124
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-1.00    sec 67.0 MBytes 562 Mbits/sec
[ 7] 1.00-2.00    sec 249 MBytes 2.09 Gbits/sec
[ 7] 2.00-3.00    sec 95.2 MBytes 799 Mbits/sec
[ 7] 3.00-4.00    sec 64.8 MBytes 543 Mbits/sec
[ 7] 4.00-5.00    sec 35.7 MBytes 299 Mbits/sec
[ 7] 5.00-6.00    sec 140 MBytes 1.18 Gbits/sec
[ 7] 6.00-7.00    sec 133 MBytes 1.11 Gbits/sec
[ 7] 7.00-8.00    sec 152 MBytes 1.28 Gbits/sec
[ 7] 8.00-9.00    sec 73.2 MBytes 615 Mbits/sec
[ 7] 9.00-10.00   sec 110 MBytes 921 Mbits/sec
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-10.00   sec 1.09 GBytes 939 Mbits/sec
receiver

"Node: h3"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 56124 connected to 10.2.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr    Cwnd
[ 7] 0.00-1.00    sec 70.9 MBytes 595 Mbits/sec 946    45.2 KBytes
[ 7] 1.00-2.00    sec 249 MBytes 2.09 Gbits/sec 3391   41.0 KBytes
[ 7] 2.00-3.00    sec 94.1 MBytes 790 Mbits/sec 1209   42.4 KBytes
[ 7] 3.00-4.00    sec 64.8 MBytes 543 Mbits/sec 867    22.6 KBytes
[ 7] 4.00-5.00    sec 35.7 MBytes 299 Mbits/sec 320    1.41 KBytes
[ 7] 5.00-6.00    sec 140 MBytes 1.18 Gbits/sec 2187   50.9 KBytes
[ 7] 6.00-7.00    sec 132 MBytes 1.11 Gbits/sec 2076   8.48 KBytes
[ 7] 7.00-8.00    sec 153 MBytes 1.28 Gbits/sec 2620   11.3 KBytes
[ 7] 8.00-9.00    sec 74.0 MBytes 621 Mbits/sec 1115   60.8 KBytes
[ 7] 9.00-10.00   sec 109 MBytes 913 Mbits/sec 1665   66.5 KBytes
[ ID] Interval      Transfer    Bitrate    Retr
[ 7] 0.00-10.00   sec 1.10 GBytes 942 Mbits/sec 16396
[ 7] 0.00-10.00   sec 1.09 GBytes 939 Mbits/sec
sender
receiver
iperf Done.
```

Observation

- On remarque une baisse très significative du débit pendant tout le test mais aussi une variation significative du débit entre les intervalles du temps de 299 Mbits/s à 2,09 Gbits/s .

TEST 3.2:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit loss 8%')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit loss 8%')
```

Résultat

```
"Node: h1"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 56136
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 56138
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-1.00      sec  2.83 MBytes  23.7 Mbits/sec
[ 7] 1.00-2.00      sec  2.25 MBytes  18.9 Mbits/sec
[ 7] 2.00-3.00      sec  3.20 MBytes  26.9 Mbits/sec
[ 7] 3.00-4.00      sec  2.09 MBytes  17.5 Mbits/sec
[ 7] 4.00-5.00      sec  1.70 MBytes  14.3 Mbits/sec
[ 7] 5.00-6.00      sec  1.06 MBytes   8.90 Mbits/sec
[ 7] 6.00-7.00      sec  4.97 MBytes  41.7 Mbits/sec
[ 7] 7.00-8.00      sec    891 KBytes  7.30 Mbits/sec
[ 7] 8.00-9.00      sec  3.38 MBytes  28.4 Mbits/sec
[ 7] 9.00-10.00     sec  2.32 MBytes  19.5 Mbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-10.01     sec  24.7 MBytes  20.7 Mbits/sec                      receiver

"Node: h3"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 56138 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7] 0.00-1.00      sec  3.02 MBytes  25.3 Mbits/sec  196   2.83 KBytes
[ 7] 1.00-2.00      sec  2.36 MBytes  19.8 Mbits/sec  110   2.83 KBytes
[ 7] 2.00-3.00      sec  3.17 MBytes  26.6 Mbits/sec  192   1.41 KBytes
[ 7] 3.00-4.00      sec  2.17 MBytes  18.2 Mbits/sec  152   19.8 KBytes
[ 7] 4.00-5.00      sec  1.68 MBytes  14.1 Mbits/sec   80   17.0 KBytes
[ 7] 5.00-6.00      sec  1018 KBytes   8.34 Mbits/sec   90   15.6 KBytes
[ 7] 6.00-7.00      sec  5.16 MBytes  43.3 Mbits/sec  240   2.83 KBytes
[ 7] 7.00-8.00      sec  1.06 MBytes   8.86 Mbits/sec   89   19.8 KBytes
[ 7] 8.00-9.00      sec  3.36 MBytes  28.1 Mbits/sec  189   4.24 KBytes
[ 7] 9.00-10.00     sec  2.30 MBytes  19.3 Mbits/sec  134   4.24 KBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7] 0.00-10.00     sec  25.3 MBytes  21.2 Mbits/sec  1472
[ 7] 0.00-10.01     sec  24.7 MBytes  20.7 Mbits/sec
-----
iperf Done.
```

Observation

- Avec une perte de paquets du 8% on remarque que la diminution du débit est encore plus signifiante, mais la variation du débit n'est pas très signifiante en la comparant avec la précédente (2%), elle est entre 8,34 Mbits/s et 43,3 Mbits/s.

TEST 3.3:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 5gbit loss 16%')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 5gbit loss 16%')
```

Résultat

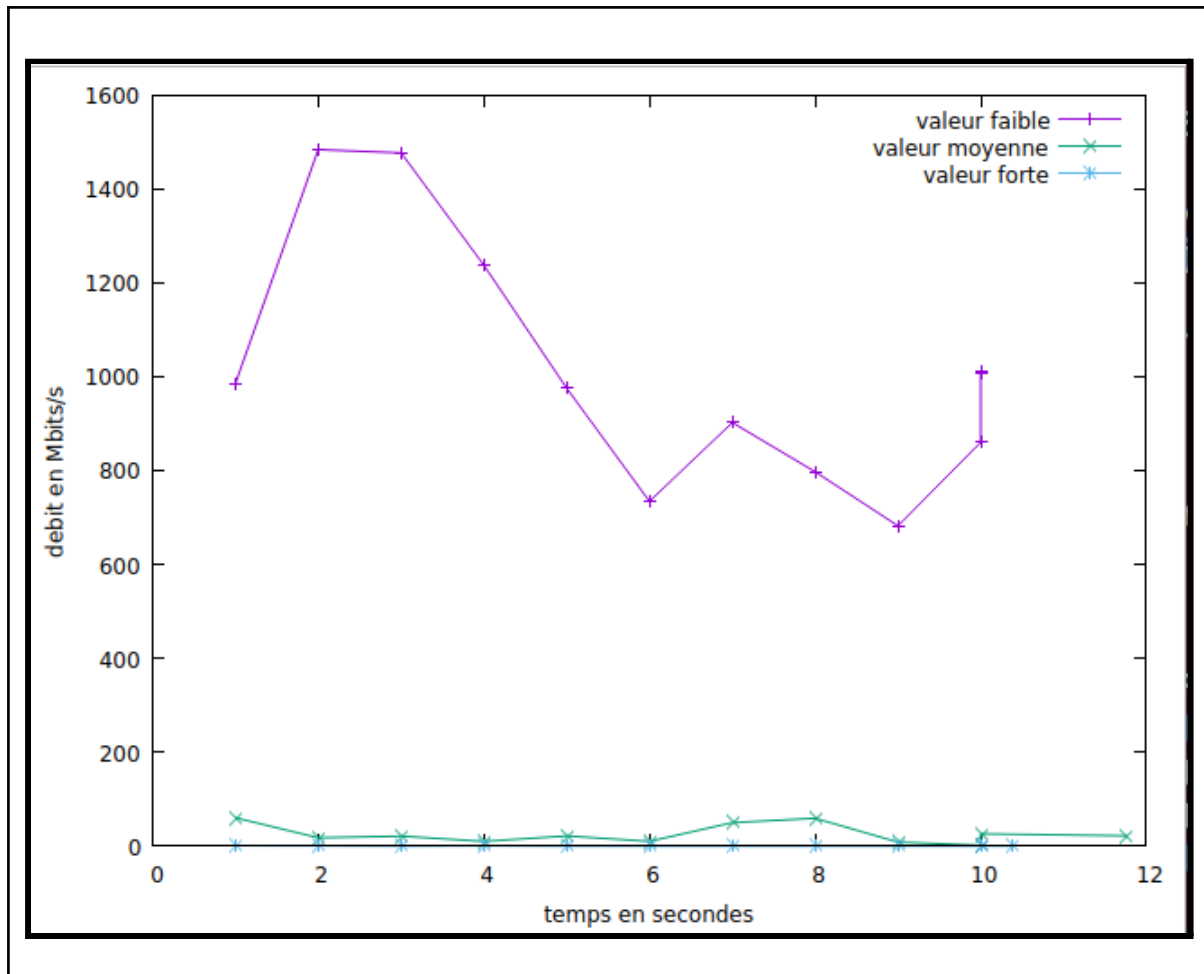
```
"Node: h1"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.2.2.2, port 56158
[ 7] local 10.2.0.2 port 5201 connected to 10.2.2.2 port 56162
[ ID] Interval      Transfer    Bitrate
[ 7]  0.00-1.00    sec  1.24 MBytes  10.4 Mbits/sec
[ 7]  1.00-2.00    sec  77.8 KBytes  637 Kbits/sec
[ 7]  2.00-3.00    sec  80.6 KBytes  660 Kbits/sec
[ 7]  3.00-4.00    sec  143 KBytes  1.17 Mbits/sec
[ 7]  4.00-5.00    sec  9.90 KBytes  81.1 Kbits/sec
[ 7]  5.00-6.00    sec  9.90 KBytes  81.1 Kbits/sec
[ 7]  6.00-7.00    sec  94.7 KBytes  776 Kbits/sec
[ 7]  7.00-8.00    sec  127 KBytes  1.04 Mbits/sec
[ 7]  8.00-9.00    sec  35.4 KBytes  290 Kbits/sec
[ 7]  9.00-10.00   sec  116 KBytes  950 Kbits/sec
[ 7] 10.00-10.20   sec  163 KBytes  6.68 Mbits/sec
-----
[ ID] Interval      Transfer    Bitrate
[ 7]  0.00-10.20   sec  2.07 MBytes  1.71 Mbits/sec
receiver
```

```
"Node: h3"
root@creponne:/home/creponnekarim/my_progs/py_progs/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 56162 connected to 10.2.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 7]  0.00-1.00    sec  1.80 MBytes  15.1 Mbits/sec  82  1.41 KBytes
[ 7]  1.00-2.00    sec  0.00 Bytes  0.00 bits/sec  15  1.41 KBytes
[ 7]  2.00-3.00    sec  255 KBytes  2.09 Mbits/sec  28  2.83 KBytes
[ 7]  3.00-4.00    sec  0.00 Bytes  0.00 bits/sec  5  1.41 KBytes
[ 7]  4.00-5.00    sec  0.00 Bytes  0.00 bits/sec  2  4.24 KBytes
[ 7]  5.00-6.00    sec  0.00 Bytes  0.00 bits/sec  3  1.41 KBytes
[ 7]  6.00-7.00    sec  0.00 Bytes  0.00 bits/sec  19  1.41 KBytes
[ 7]  7.00-8.00    sec  255 KBytes  2.08 Mbits/sec  19  2.83 KBytes
[ 7]  8.00-9.00    sec  0.00 Bytes  0.00 bits/sec  6  1.41 KBytes
[ 7]  9.00-10.00   sec  255 KBytes  2.08 Mbits/sec  31  4.24 KBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 7]  0.00-10.00   sec  2.55 MBytes  2.14 Mbits/sec  210
[ 7]  0.00-10.20   sec  2.07 MBytes  1.71 Mbits/sec
sender
receiver
iperf Done.
```

Observation

- Avec une perte de paquets de 16% on remarque une baisse drastique du débit en atteignant 0 bits/s souvent au côté du serveur.

Graph :



Explication :

Comme nous pouvons le remarquer dans le schéma ci dessus, qui représente la fluctuation des différents débits dans la liaison h1-h3 correspondant à différentes valeurs de perte de paquets (faible, moyenne et élevée), l'introduction ne serait-ce qu'un pourcentage très minime tel que taux de perte fait diminuer drastiquement et fait fluctuer le débit de manière considérable. Comme nous pouvons le constater pour une valeur faible de taux de perte nous avons un débit applicatif qui fluctue entre les valeurs 600 mb/s et 1600 mb/s, nous voyons aussi que pour les valeurs moyennes et élevées du taux de perte une diminution du débit applicatif qui avoisine le zéro. Ceci est dû au fait que dans la liaison r1-r2 nous avons introduit un paramètre de perte de paquet qui cause donc la perte de paquet envoyé de h1 à h3 et de paquets envoyé de h3 à h1, par conséquent ceci se traduit par une diminution et une fluctuation du débit applicatif.

TEST 4:

Commandes utilisées

```
# change link settings for both ports
r2.cmd('sudo tc qdisc add dev r2-eth3 root netem rate 2.5gbit delay 5ms loss 2%')
r1.cmd('sudo tc qdisc add dev r1-eth3 root netem rate 2.5gbit delay 5ms loss 2%')
```

Résultat

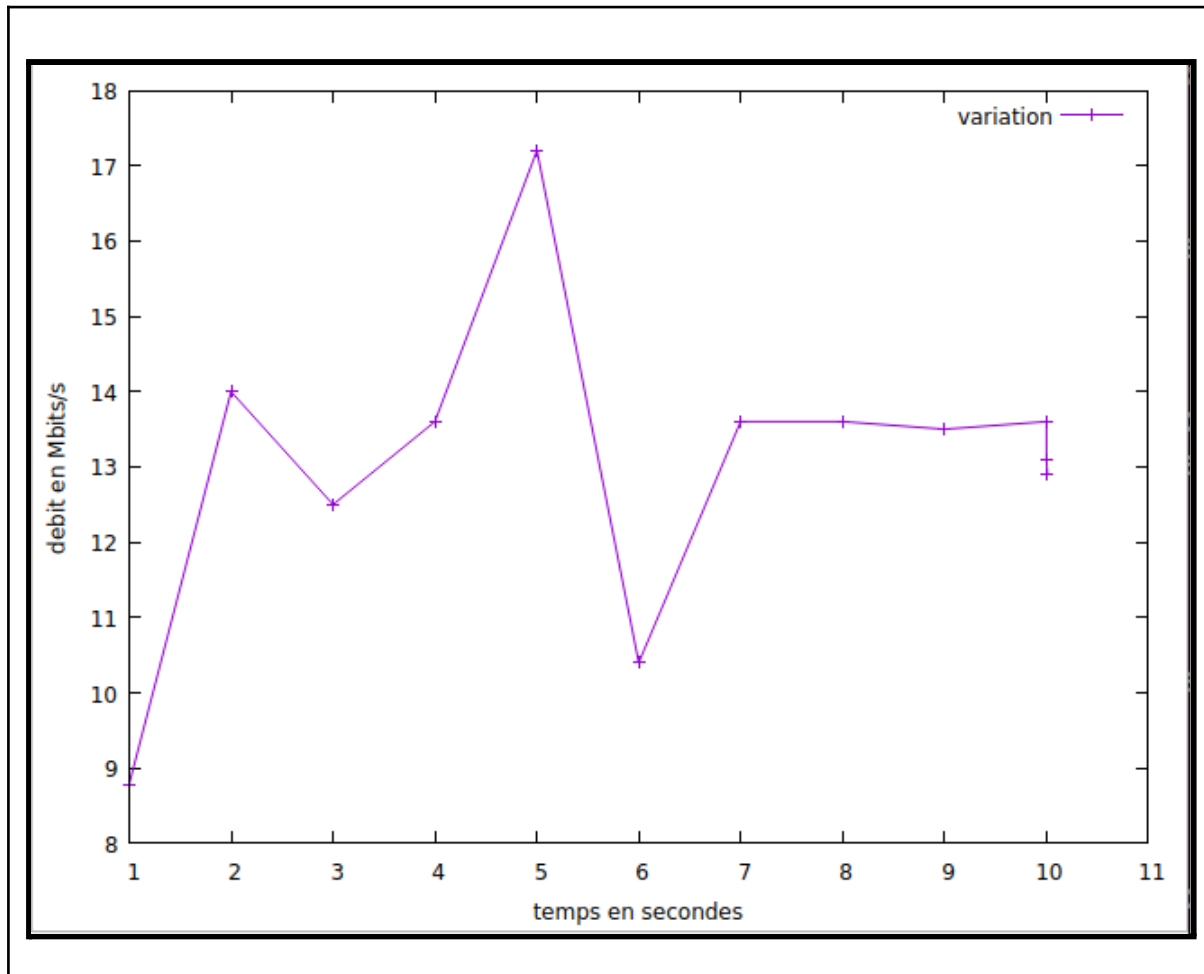
```
"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 51696 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr    Cwnd
[ 7]  0.00-1.00    sec  43.8 MBytes  367 Mbits/sec  547    12.7 KBytes
[ 7]  1.00-2.00    sec  1.25 MBytes  10.5 Mbits/sec   16    24.0 KBytes
[ 7]  2.00-3.00    sec  1.25 MBytes  10.5 Mbits/sec   22    25.5 KBytes
[ 7]  3.00-4.00    sec  1.25 MBytes  10.5 Mbits/sec   20    19.8 KBytes
[ 7]  4.00-5.00    sec  1.25 MBytes  10.5 Mbits/sec   21    22.6 KBytes
[ 7]  5.00-6.00    sec  1.25 MBytes  10.5 Mbits/sec   23    19.8 KBytes
[ 7]  6.00-7.00    sec  1.25 MBytes  10.5 Mbits/sec   21    21.2 KBytes
[ 7]  7.00-8.00    sec  1.25 MBytes  10.5 Mbits/sec   28    7.07 KBytes
[ 7]  8.00-9.00    sec  1.25 MBytes  10.5 Mbits/sec   18    12.7 KBytes
[ 7]  9.00-10.00   sec  1.25 MBytes  10.5 Mbits/sec   14    11.3 KBytes
-- -- --
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00    sec  55.1 MBytes  46.2 Mbits/sec  730
[ 7]  0.00-10.01    sec  43.9 MBytes  36.8 Mbits/sec
sender
receiver
iperf Done.
```

```
"Node: h3"
root@lenovo-ThinkPad-T580:/home/lenovo/mininet# iperf3 -c 10.2.0.2
Connecting to host 10.2.0.2, port 5201
[ 7] local 10.2.2.2 port 51696 connected to 10.2.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr    Cwnd
[ 7]  0.00-1.00    sec  43.8 MBytes  367 Mbits/sec  547    12.7 KBytes
[ 7]  1.00-2.00    sec  1.25 MBytes  10.5 Mbits/sec   16    24.0 KBytes
[ 7]  2.00-3.00    sec  1.25 MBytes  10.5 Mbits/sec   22    25.5 KBytes
[ 7]  3.00-4.00    sec  1.25 MBytes  10.5 Mbits/sec   20    19.8 KBytes
[ 7]  4.00-5.00    sec  1.25 MBytes  10.5 Mbits/sec   21    22.6 KBytes
[ 7]  5.00-6.00    sec  1.25 MBytes  10.5 Mbits/sec   23    19.8 KBytes
[ 7]  6.00-7.00    sec  1.25 MBytes  10.5 Mbits/sec   21    21.2 KBytes
[ 7]  7.00-8.00    sec  1.25 MBytes  10.5 Mbits/sec   28    7.07 KBytes
[ 7]  8.00-9.00    sec  1.25 MBytes  10.5 Mbits/sec   18    12.7 KBytes
[ 7]  9.00-10.00   sec  1.25 MBytes  10.5 Mbits/sec   14    11.3 KBytes
-- -- --
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00    sec  55.1 MBytes  46.2 Mbits/sec  730
[ 7]  0.00-10.01    sec  43.9 MBytes  36.8 Mbits/sec
sender
receiver
iperf Done.
```

Observation

- On remarque une baisse drastique au niveau de débit en comparant entre 2.5 Gbits/s qui est le débit de lien spécifié et le débit applicatif qu'on a obtenu en lançant le test qui est environ 10.5 Mbits/s, ce qui est complètement inacceptable et inutilisable.

Graph:



Explication :

Le graph ici présent, représente la variation du débit applicatif dans la liaison h1-h3 par rapport au temps après avoir appliqué une série de paramètres (cités dans les commandes plus haut) à la liaison interne r1-r2. Nous remarquons une fluctuation du débit que nous pouvons attribuer au paramètre de taux de perte (simple déduction du test numéro 3), de plus la diminution du débit est causé par la limitation de ce dernier sur le lien r1-r2 (conclusion du test numéro 1), mais elle est aussi dû au fait que nous avons introduit une latence de 5 ms (déduction du test numéro 2) et aussi au taux de perte (test numéro 3).

III. Synthèse globale

Dans les tests précédemment effectués nous pouvons remarquer et tirer une série de conclusions assez intéressantes pour chacun des paramètres spécifiés : le débit du lien, la latence, et le taux de perte des paquets.

Nous avons remarqué à travers le premier test (celui du débit) que nous pouvons contrôler le débit applicatif d'un lien juste en réduisant le débit d'une partie de ce lien, limitant ainsi la fenêtre du débit qui passe par le lien, en l'occurrence dans notre cas le débit entre h1 et h3.

Pour le second test, le test concernant la latence, nous avons déduit qu' en modifiant la latence d'une partie du lien c'est tout le lien qui est impacté, mais comment? L'effet de la latence dans les tests effectués se résume en une réduction du débit applicatif spécifié sans pour autant le faire fluctuer, en d'autres termes la latence réduit le débit spécifié de manière stable, et plus elle est grande plus le débit applicatif est réduit.

Pour ce qui est du troisième test, le test de taux de perte, nous pouvons aussi constater à travers le test que le taux de perte appliqué au lien r1-r2 impacte, comme dans les autres tests, tout le lien h1-h3. L'effet qu'a le taux de perte sur le débit du lien spécifié est assez varié: non seulement il fait diminuer le débit applicatif (plus le taux de perte est grand plus le débit du lien diminue), mais il fait aussi fluctuer le débit grandement car la perte de paquets dépend tout d'abord de la taille du paquet qui peut impacter grandement le débit si le paquet perdu est un grand paquet, et au contraire peut réduire le débit que de peu si le paquet perdu est petit.

Pour le quatrième test, il représente la concaténation de toutes les déductions et conclusions que nous avons listé plus en haut, faisant ainsi subir au réseau tous les aléas correspondant aux paramètres détaillés précédemment.

IV. Conclusion

En conclusion, on peut dire que cette partie du projet nous a permis d'examiner en détail la variation et les différentes caractéristiques du débit applicatif dans une connection client-serveur en interne à un réseau informatique, et ceci à travers la variation de plusieurs paramètres à savoir le débit de lien entre les routeurs, la latence et le taux de perte de paquets. Les conclusions qu'on a pu ressortir vont nous aider à concevoir des réseaux ou nous pouvons avoir un grand contrôle sur le débit et pouvoir ainsi garder dans la majorité du temps un débit convenable.

Et dans ce qui suivra du projet nous allons nous attaquer à une autre série de test qui nous permettra à son tour, d'explorer les paramètres du protocole TCP (visiblement le plus utilisé de tous les protocoles) susceptibles d'impacter les performances d'un réseau, pour ainsi, non seulement comprendre de manière détaillé la façon avec laquelle nous pouvons concevoir un réseau en prenant en compte le protocole TCP, mais aussi accroître notre compréhension du protocole lui même.

V. Ressources et références utilisées:

- NTP Lab Séries
- [Iperf - Mininet](#)
- csie.nqu.edu.tw
- intronetworks.cs.luc.edu
- [Python Mininet Addhost method examples](#)
- [Mininet Custom Topologies](#)
- [Introduction to Mininet](#)