REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE ÉCOLE NATIONALE SUPERIEURE D'INFORMATIQUE (ESI EX. INI)

Rapport EQ05 G09 RES1 - Partie I
Evaluation des réseaux informatique filaire

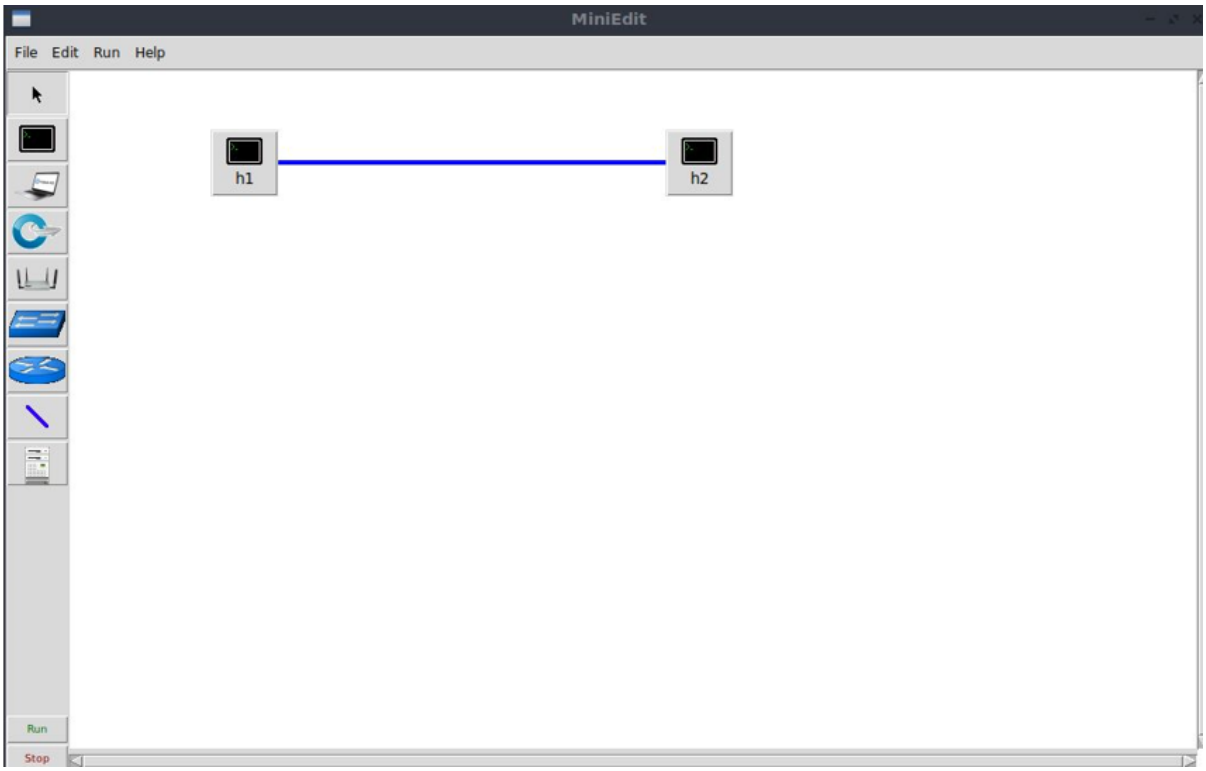
AUTRICES

SID AHMED Sarah - G09
KADID Selssabil - G09
DJOUAHER Yasmine - G09
KHEDDIA Assia G09

Table des matières

1.Mise en place de la topologie

La topologie mise en place :



Script utilisé pour la mise en place de la topologie :

```
  GNU nano 4.8                    topologie.py                    Modif
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.link import TCLink

class DirectLinkTopo(Topo):
    def build(self):

        h1 = self.addHost('h1', ip='10.5.0.6/16')
        h2 = self.addHost('h2', ip='10.5.0.7/16')

        self.addLink(h1, h2, bw=10)

if __name__ == '__main__':
    topo = DirectLinkTopo()
    net = Mininet(topo=topo, link=TCLink)
    net.start()
    CLI(net)
    net.stop()
```

Résultat de l'éxecution de la commande Ping :

```
mininet> h1 ping h2
PING 10.5.0.7 (10.5.0.7) 56(84) bytes of data.
64 bytes from 10.5.0.7: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 10.5.0.7: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.5.0.7: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 10.5.0.7: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.5.0.7: icmp_seq=5 ttl=64 time=0.077 ms
64 bytes from 10.5.0.7: icmp_seq=6 ttl=64 time=0.049 ms
```

Explication des résultats obtenus :

Dans le cadre de notre configuration réseau, les hôtes h1 (serveur) et h2 (client) sont connectés directement par un câble, sans l'utilisation d'un commutateur ou d'un routeur intermédiaire. La commande ping a été utilisée pour évaluer la latence entre ces deux hôtes. Voici les observations détaillées :

1. **Structure de la commande Ping :** La commande ping envoie des paquets ICMP (Internet Control Message Protocol) de h1 à h2 pour mesurer le temps de réponse entre les deux hôtes. Chaque paquet envoyé est de 56 octets, et h1 reçoit une réponse pour chaque requête envoyée, indiquant une bonne connectivité entre les deux hôtes.
2. **Analyse des temps de réponse :** Les résultats montrent des temps de réponse très bas, variant entre 0,048 ms et 0,077 ms. Ces valeurs sont obtenues grâce à la connexion directe entre h1 et h2, sans passer par des équipements intermédiaires, ce qui élimine les retards supplémentaires. Les informations fournies pour chaque paquet sont les suivantes :
   - icmp_seq=n : numéro de séquence du paquet envoyé.
   - ttl=64 : la valeur Time-to-Live, qui indique ici un parcours sans saut intermédiaire.
   - time=x ms : temps en millisecondes pour le trajet aller-retour du paquet.
3. **Interprétation des résultats :** La faible latence mesurée ici est caractéristique d'une connexion réseau directe et optimisée, sans influence de la congestion ou de traitements réseau complexes. Les petites variations dans les temps de réponse sont normales et peuvent être dues à de légères fluctuations matérielles. Une latence aussi basse garantit une transmission rapide et fiable, essentielle dans des applications critiques ou en temps réel.
4. **Importance de ces résultats :** Dans le cadre d'un réseau où une faible latence est cruciale, ces résultats confirment que la connexion directe entre h1 et h2 est optimale. Cette configuration offre un canal de communication rapide et stable, ce qui est indispensable dans des environnements nécessitant une performance en temps réel.

2.Test T1.1

2.1 Courbe de la variation du débit obtenu :

Script pour générer le graphe du débit obtenu en fonction du débit physique :

```
 GNU nano 4.8                                                    script_debit.gnu
set terminal pngcairo enhanced
set output 'variation_debit1.png'

set title "Variation du debit"
set xlabel "debit physique(Mbps)"
set ylabel "debit obtenu(Mbps)"

set grid

plot 'debit.txt' using 1:2 with linespoints title 'debit obtenu' lw 2 pt 7 lc rgb 'red'
```

Le fichier texte contenant les valeurs obtenue il sera utiliser par le script ci dessus :

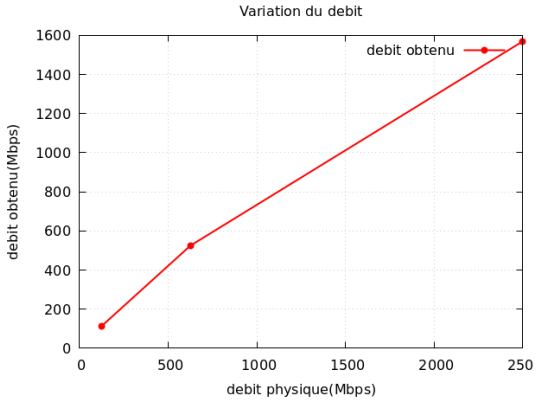
```
 GNU nano 4.8                                                          debit.txt
125 114
625 525
2500 1570
```

Les commandes utilisées :

```
wifi@wifi-virtualbox:~$ nano script_debit.gnu
wifi@wifi-virtualbox:~$ nano debit.txt
wifi@wifi-virtualbox:~$ gnuplot script_debit.gnu
```

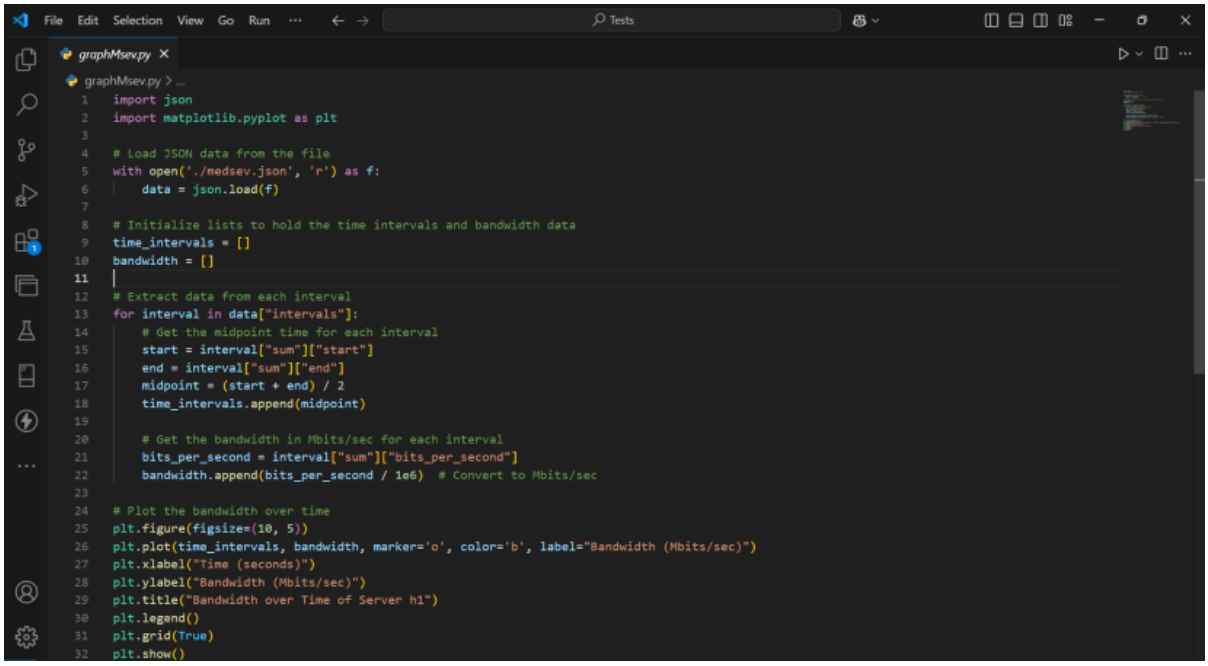
1. Commande 1 crée un fichier nommé “script_debit.gnu” et l’ouvre.
2. Commande 2 crée un fichier nommé “debit.txt” et l’ouvre.
3. Commande 3 exécute le script gnuplot “script_debit.gnu”.

La courbe obtenue :



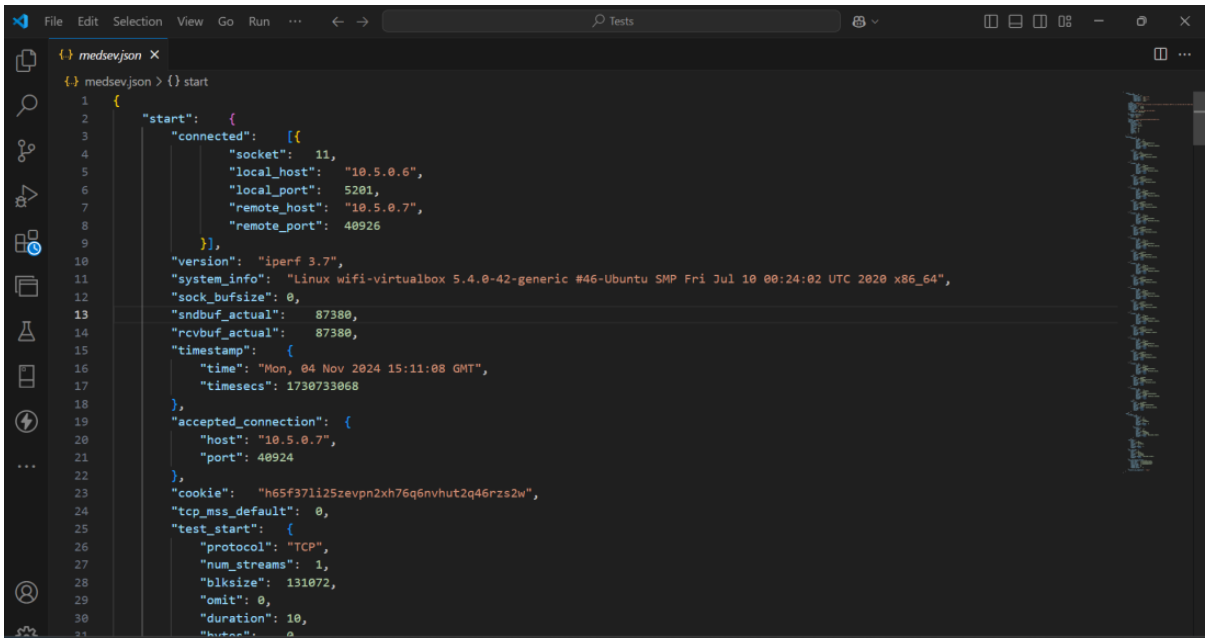
2.2 Une prise d'écran de chaque résultat et une interprétation :

Script pour générer les courbes du débit obtenue en fonction du temps :



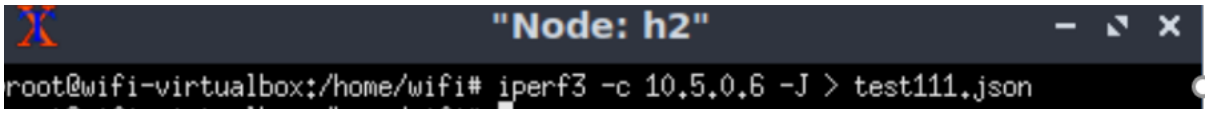
```python
import json
import matplotlib.pyplot as plt

# Load JSON data from the file
with open('./medsev.json', 'r') as f:
    data = json.load(f)

# Initialize lists to hold the time intervals and bandwidth data
time_intervals = []
bandwidth = []

# Extract data from each interval
for interval in data["intervals"]:
    # Get the midpoint time for each interval
    start = interval["sum"]["start"]
    end = interval["sum"]["end"]
    midpoint = (start + end) / 2
    time_intervals.append(midpoint)

    # Get the bandwidth in Mbits/sec for each interval
    bits_per_second = interval["sum"]["bits_per_second"]
    bandwidth.append(bits_per_second / 1e6)  # Convert to Mbits/sec

# Plot the bandwidth over time
plt.figure(figsize=(10, 5))
plt.plot(time_intervals, bandwidth, marker='o', color='b', label="Bandwidth (Mbits/sec)")
plt.xlabel("Time (seconds)")
plt.ylabel("Bandwidth (Mbits/sec)")
plt.title("Bandwidth over Time of Server h1")
plt.legend()
plt.grid(True)
plt.show()
```

Fichier JSON contenant des données provenant d'un test de performance réseau généré par l'outil iperf3 :




```json
{
    "start":    {
        "connected":    [{
                "socket":   11,
                "local_host":   "10.5.0.6",
                "local_port":   5201,
                "remote_host":  "10.5.0.7",
                "remote_port":  40926
            }],
        "version":  "iperf 3.7",
        "system_info":  "Linux wifi-virtualbox 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020 x86_64",
        "sock_bufsize": 0,
        "sndbuf_actual":    87380,
        "rcvbuf_actual":    87380,
        "timestamp":    {
            "time": "Mon, 04 Nov 2024 15:11:08 GMT",
            "timesecs": 1730733068
        },
        "accepted_connection":  {
            "host": "10.5.0.7",
            "port": 40924
        },
        "cookie":   "h65f37li25zevpn2xh76q6nvhut2q46rzs2w",
        "tcp_mss_default":  0,
        "test_start":   {
            "protocol": "TCP",
            "num_streams":  1,
            "blksize":  131072,
            "omit": 0,
            "duration": 10,
```


Les commandes utilisées :

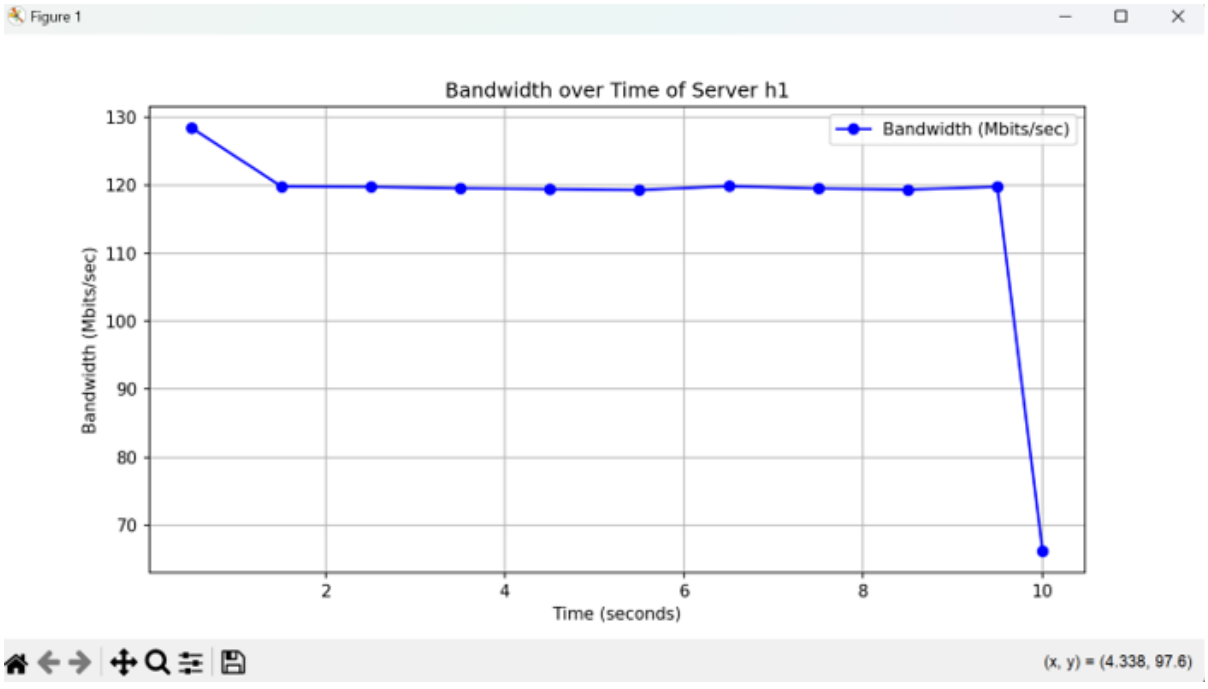


1. sudo python3 topologie.py : exécute le script de la topologie.
2. hx tc qdisc del dev hx-eth0 root : supprime toutes les configurations de file d'attente appliquées à l'interface réseau hx-eth0
3. hx tc qdisc add dev hx-eth0 root tbf rate yMbit burst 1540Kbit latency 100ms : ajoute une règle de contrôle de trafic pour limiter le débit de l'interface hx-eth0 avec les paramètres suivants :
   - Rate yMbit :Limite le débit maximum à y Mbit/s.
   - Burst 1540Kbit : Autorise une rafale (burst) de 1540 Kbit pour compenser les variations de trafic momentanées.
   - Latency 100ms : Permet une latence maximale de 100 ms.
4. h2 iperf3 -c 10.5.0.6 -J : exécute un test de performance réseau en mode client depuis h2 vers le serveur à l'adresse IP 10.5.0.6, et retourne les résultats de bande passante sous format JSON pour une analyse automatisée.
5. h1 iperf3 -s -J : démarre un serveur de test de performance réseau sur h1, qui écoute les connexions entrantes sur toutes ses interfaces, et retourne les résultats en format JSON pour faciliter l'analyse des données.

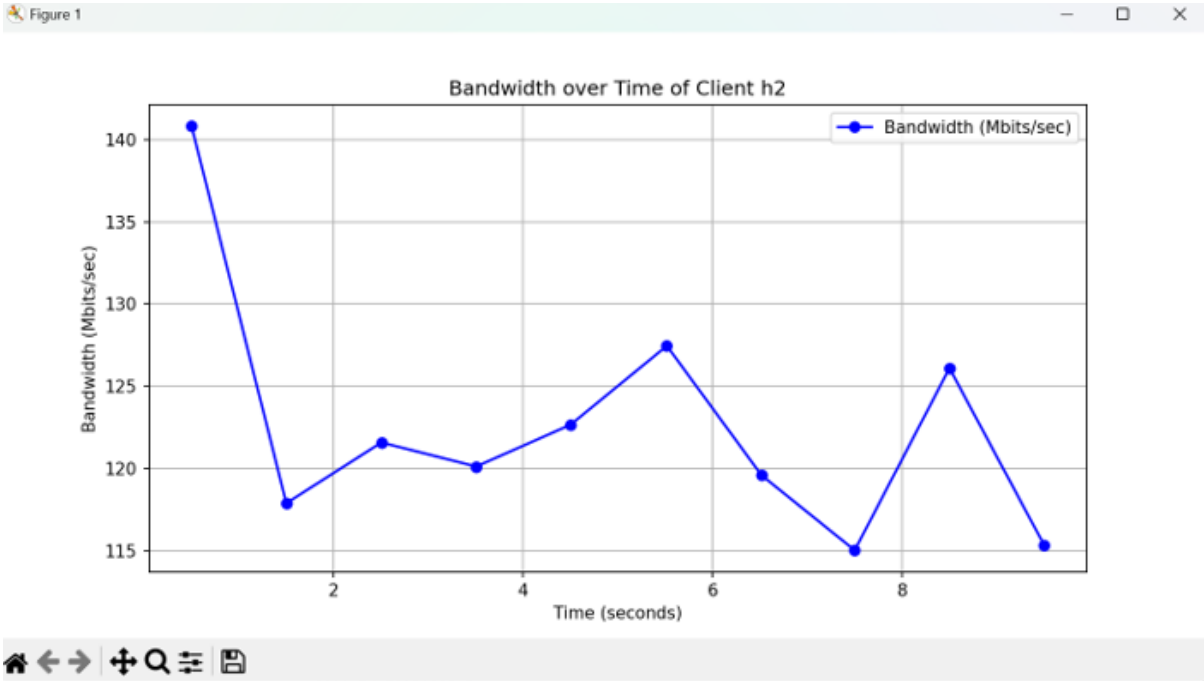
Observation : x est soit 1 ou 2 et y est la valeur du débit physique.

2.2.1 Valeur faible 125Mbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
wifi@wifi-virtualbox:~$ sudo python3 topologie.py
[sudo] password for wifi:
mininet> h1 tc qdisc del dev h1-eth0 root
mininet> h2 tc qdisc del dev h2-eth0 root
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 125Mbit burst 1540kbit latency 100ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 125mbit burst 1540kbit latency 100ms
mininet> h1 iperf3 -s &
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 55304 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  15.4 MBytes   129 Mbits/sec    0    209 KBytes
[  5]   1.00-2.04   sec  11.4 MBytes  92.2 Mbits/sec    0    256 KBytes
[  5]   2.04-3.00   sec  14.0 MBytes   122 Mbits/sec    0    256 KBytes
[  5]   3.00-4.00   sec  14.0 MBytes   117 Mbits/sec    0    256 KBytes
[  5]   4.00-5.00   sec  14.6 MBytes   123 Mbits/sec    0    256 KBytes
[  5]   5.00-6.00   sec  13.0 MBytes   109 Mbits/sec    0    256 KBytes
[  5]   6.00-7.00   sec  14.7 MBytes   123 Mbits/sec    0    269 KBytes
[  5]   7.00-8.00   sec  12.2 MBytes   102 Mbits/sec    0    269 KBytes
[  5]   8.00-9.00   sec  12.6 MBytes   106 Mbits/sec    0    269 KBytes
[  5]   9.00-10.00  sec  14.3 MBytes   120 Mbits/sec    0    269 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   136 MBytes   114 Mbits/sec    0             sender
[  5]   0.00-10.01  sec   135 MBytes   113 Mbits/sec                  receiver

iperf Done.
```

Interprétation des résultats :

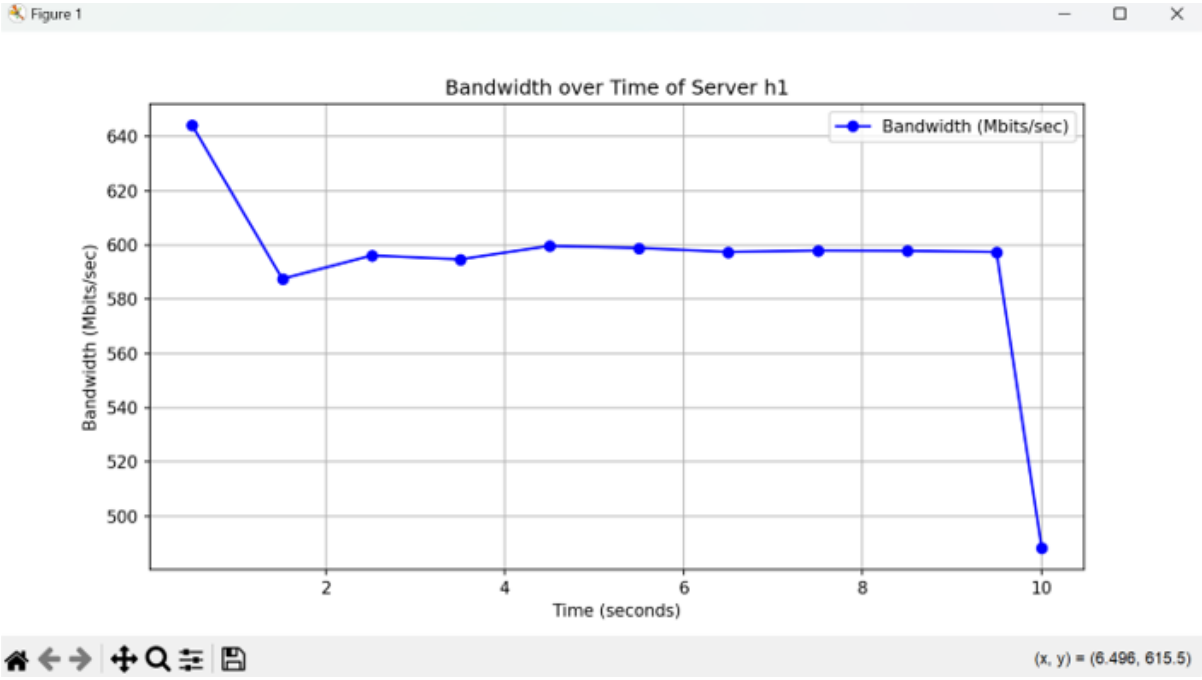
Explication des pertes dues aux protocoles réseau : À faible débit, les frais de communication (overhead) associés aux protocoles de la couche réseau, comme TCP/IP, deviennent significatifs. Chaque paquet TCP/IP inclut des en-têtes qui assurent l'acheminement et l'intégrité des données, mais cet overhead occupe une partie de la bande passante. À ce niveau de débit, cette occupation par les en-têtes réduit le débit effectif mesuré, car une plus grande proportion de la bande passante est consacrée aux informations de contrôle.

Effet minimal de la latence et de la congestion : À ce faible débit, la congestion et la latence ne sont généralement pas problématiques, car le volume de données reste bas, et les tampons (buffers) ne sont pas saturés. Les différences entre le débit configuré et le débit obtenu proviennent donc principalement des protocoles de gestion, et non des limites matérielles ou des retards dans la transmission des données.

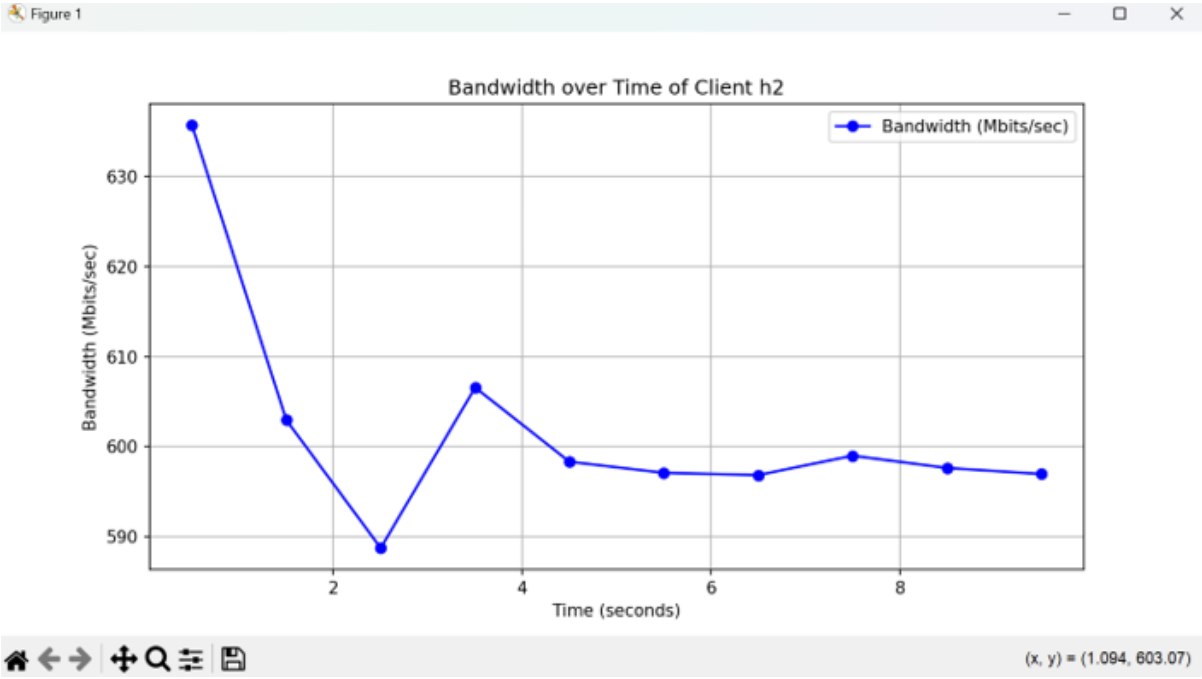
Conclusion pour ce cas : Cette légère réduction de débit est attendue et reste dans des limites raisonnables. Elle s'explique par les protocoles réseau et l'overhead induit, qui impactent légèrement le débit sans compromettre la qualité de la transmission.

2.2.2 Valeur moyenne 625Mbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
wifi@wifi-virtualbox:~$ sudo python3 topologie.py
mininet> h1 tc qdisc del dev h1-eth0 root
mininet> h2 tc qdisc del dev h2-eth0 root
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 625Mbit burst 1540kbit latency 100ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 625mbit burst 1540kbit latency 100ms
mininet> h1 iperf3 -s &
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 55344 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  58.0 MBytes   486 Mbits/sec    0    581 KBytes
[  5]   1.00-2.00   sec  65.5 MBytes   549 Mbits/sec    0    772 KBytes
[  5]   2.00-3.00   sec  66.2 MBytes   556 Mbits/sec    0    772 KBytes
[  5]   3.00-4.00   sec  61.2 MBytes   512 Mbits/sec    0    772 KBytes
[  5]   4.00-5.00   sec  68.8 MBytes   579 Mbits/sec   45    864 KBytes
[  5]   5.00-6.00   sec  63.8 MBytes   535 Mbits/sec    0    867 KBytes
[  5]   6.00-7.00   sec  58.8 MBytes   493 Mbits/sec    0    868 KBytes
[  5]   7.00-8.00   sec  55.0 MBytes   462 Mbits/sec    0    871 KBytes
[  5]   8.00-9.00   sec  62.5 MBytes   524 Mbits/sec    0    871 KBytes
[  5]   9.00-10.00  sec  66.2 MBytes   556 Mbits/sec    0    908 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   626 MBytes   525 Mbits/sec   45             sender
[  5]   0.00-10.01  sec   622 MBytes   521 Mbits/sec                  receiver

iperf Done.
```

Interprétation des résultats :

Limites des équipements et de la couche réseau : À un débit moyen, les équipements réseau, comme les cartes réseau et les processeurs, sont davantage sollicités. À ce stade, les tampons des équipements peuvent rapidement atteindre leur capacité maximale, ce qui entraîne des baisses de débit effectif. Cette surcharge expose les limitations de certains équipements, qui ne sont pas toujours optimisés pour gérer des débits intermédiaires de manière efficace.

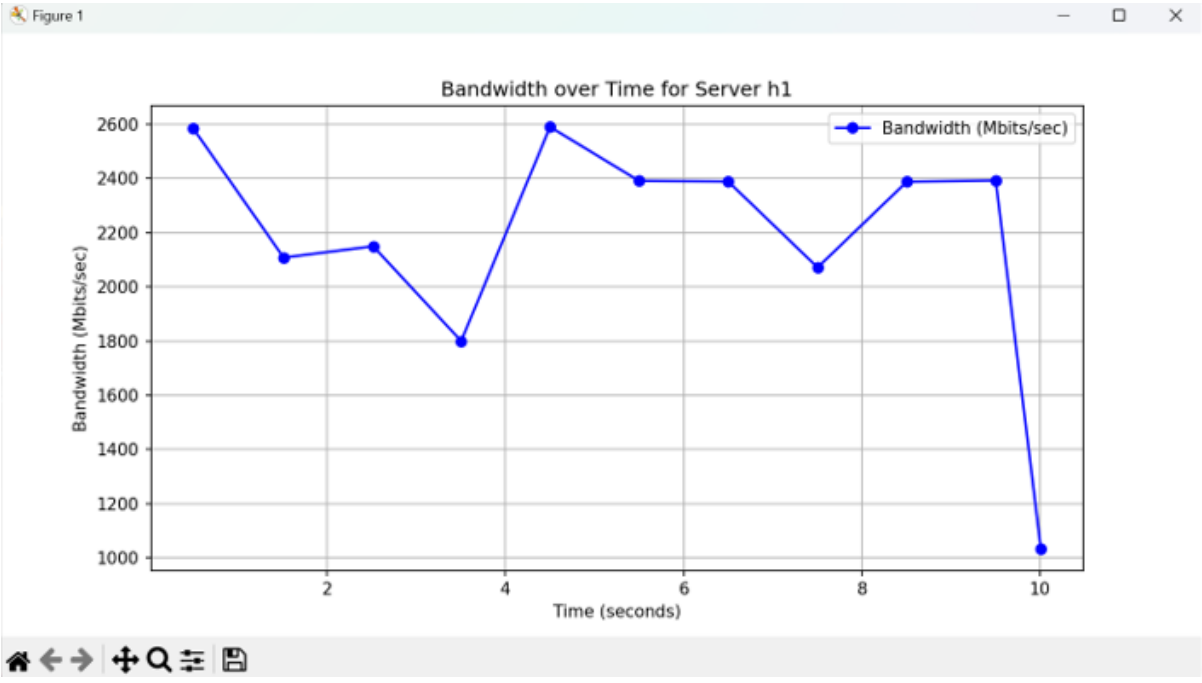
Impact accru de la latence et de ses variances : Avec un débit moyen, la latence et ses variances deviennent plus prononcées. Les fluctuations de la latence peuvent entraîner des réceptions inconstantes et nécessiter des retransmissions, réduisant ainsi le débit mesuré.

Effet des mécanismes de contrôle de flux : Les protocoles de gestion de flux, comme le contrôle de congestion de TCP, deviennent plus actifs lorsque le débit augmente. Ces mécanismes visent à maintenir la stabilité du réseau en limitant la surcharge, mais ils peuvent restreindre le débit disponible pour chaque flux de données. C'est pourquoi l'écart entre le débit configuré et obtenu est plus important à ce niveau.

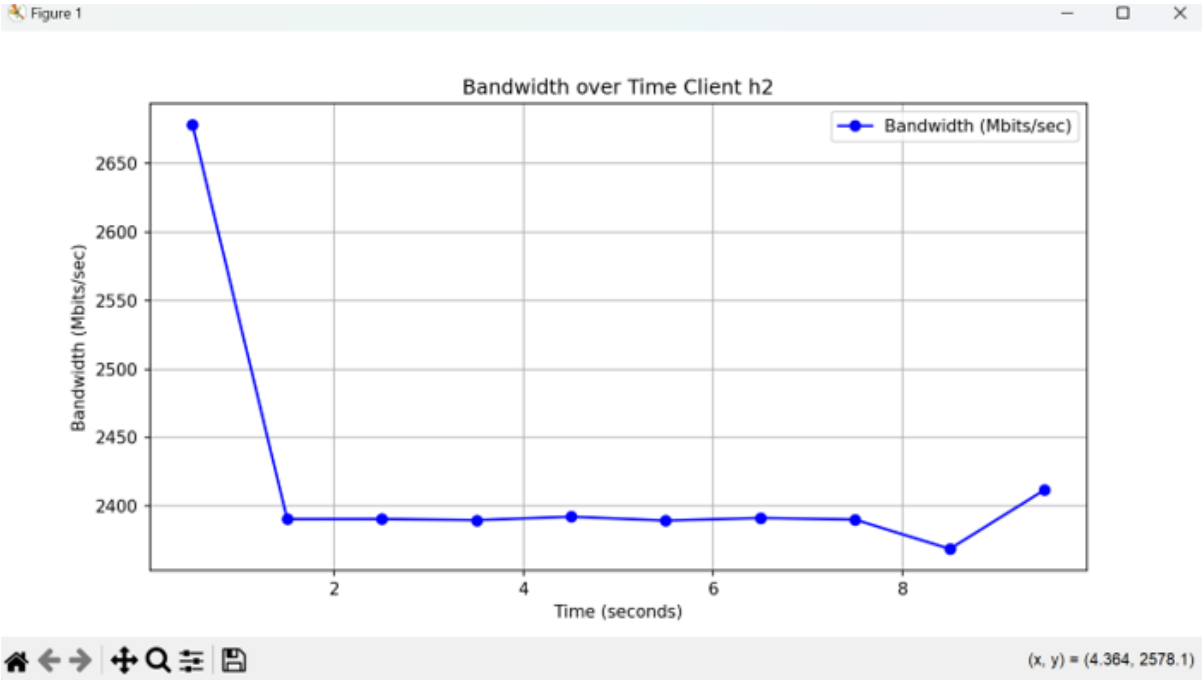
Conclusion pour ce cas : Cette réduction de débit est significative en raison de l'impact combiné de la gestion des flux, des limites des équipements, et de la variabilité de la latence. Les protocoles réseau exercent également une influence accrue sur les performances à mesure que le débit augmente.

2.2.3 Valeur élevée 2.5Gbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
wifi@wifi-virtualbox:~$ sudo python3 topologie.py
mininet> h1 tc qdisc del dev h1-eth0 root
mininet> h2 tc qdisc del dev h2-eth0 root
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 2.5gbit burst 1540kbit latency 100ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 2.5gbit burst 1540kbit latency 100ms
mininet> h1 iperf3 -s &
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 55932 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   182 MBytes  1.53 Gbits/sec  122   1.43 MBytes
[  5]   1.00-2.00   sec   246 MBytes  2.07 Gbits/sec   45   1.44 MBytes
[  5]   2.00-3.00   sec   171 MBytes  1.43 Gbits/sec  135   1.06 MBytes
[  5]   3.00-4.00   sec   171 MBytes  1.44 Gbits/sec    0   1.10 MBytes
[  5]   4.00-5.02   sec   216 MBytes  1.78 Gbits/sec   90   1.24 MBytes
[  5]   5.02-6.01   sec   136 MBytes  1.15 Gbits/sec  135   1.24 MBytes
[  5]   6.01-7.00   sec   198 MBytes  1.67 Gbits/sec   45   1.27 MBytes
[  5]   7.00-8.01   sec  93.8 MBytes   778 Mbits/sec   32   1.28 MBytes
[  5]   8.01-9.00   sec   240 MBytes  2.04 Gbits/sec  109    947 KBytes
[  5]   9.00-10.00  sec   214 MBytes  1.79 Gbits/sec    0    687 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.82 GBytes  1.57 Gbits/sec  713             sender
[  5]   0.00-10.00  sec  1.82 GBytes  1.56 Gbits/sec                  receiver

iperf Done.
```

Interprétation des résultats :

Limites matérielles des interfaces réseau : À un débit aussi élevé, les limitations des interfaces réseau (comme les cartes réseau et les bus de données) deviennent évidentes. Si les équipements ne sont pas spécifiquement conçus pour supporter de tels débits, le débit mesuré sera nettement inférieur au débit configuré. Par exemple, une carte réseau classique ou une machine virtuelle peut ne pas être capable de gérer efficacement 2,5 Gbit/s.

Saturation et gestion des tampons : Avec un débit élevé, les tampons réseau atteignent leur capacité maximale très rapidement, provoquant des pertes de paquets et des retransmissions fréquentes, ce qui réduit le débit effectif. Dans les environnements virtualisés, où les ressources sont partagées, cette saturation des tampons est amplifiée, entraînant une baisse encore plus marquée du débit.

Augmentation des coûts de gestion des protocoles : À des débits élevés, les protocoles réseau (par exemple, TCP/IP) exigent davantage de ressources pour traiter les grandes quantités de données. Le réseau doit gérer des processus supplémentaires pour assurer l'intégrité des paquets et éviter la congestion. Ces opérations de gestion des protocoles consomment une partie significative de la bande passante, réduisant ainsi le débit effectif.

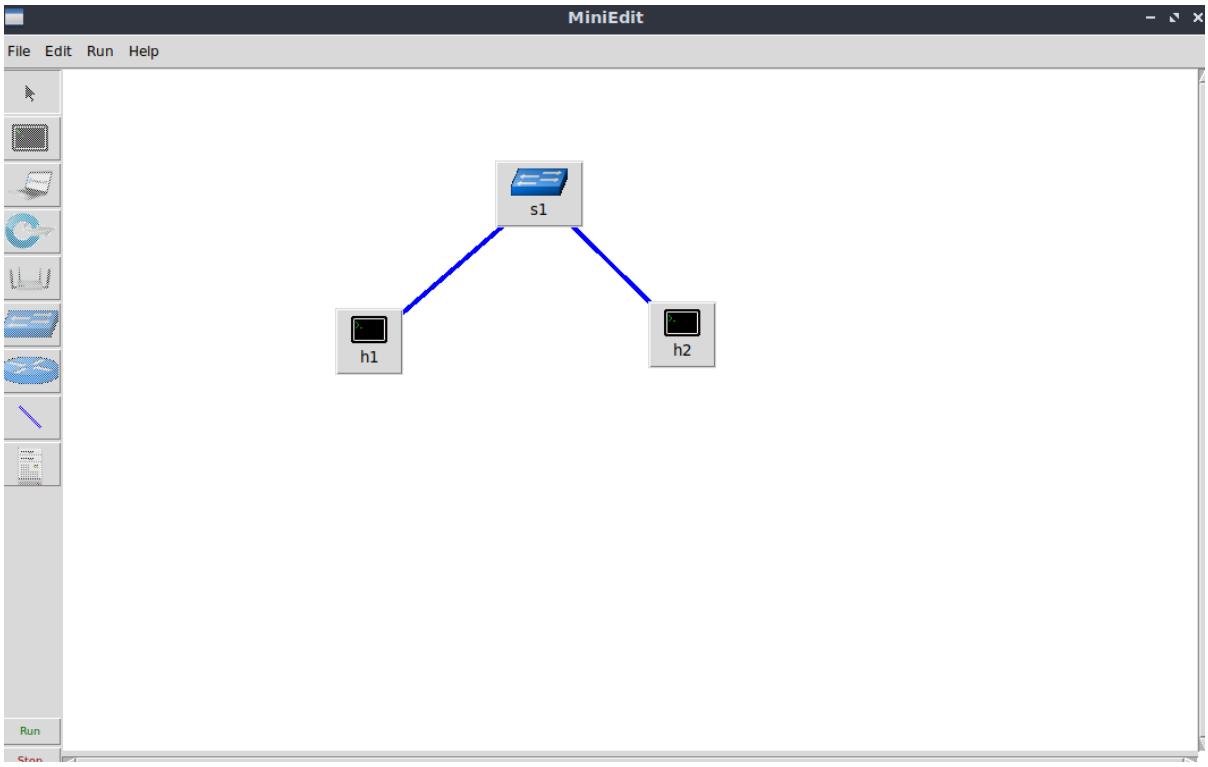
Conclusion pour ce cas : La réduction notable du débit obtenu à 1,57 Gbit/s s'explique par les limites matérielles, la saturation des tampons, et les coûts élevés de gestion des protocoles. Ces facteurs limitent l'atteinte du débit maximal configuré.

Interprétation générale : Les résultats montrent qu'à mesure que le débit configuré augmente, l'écart entre le débit physique théorique et le débit effectif augmente également. Ce phénomène s'explique par plusieurs facteurs :

1. **Limitations matérielles :** Les équipements réseau standard ne sont pas toujours optimisés pour gérer des débits élevés. Cela nécessite parfois l'utilisation de matériel spécialisé (ex. : cartes réseau haute performance).
2. **Gestion des protocoles :** Les protocoles de contrôle, comme ceux de la couche de transport dans le modèle OSI, imposent des frais de gestion (overhead) plus importants à mesure que le débit augmente.
3. **Contraintes d'infrastructure :** Dans les environnements virtualisés, les ressources sont partagées, ce qui peut causer des saturations et réduire les performances.

3.Test T1.2

Mise en place de la topologie :



3.1 Courbe de la variation du débit obtenu :

Script pour générer le graphe du débit obtenu en fonction du débit physique :

```
 GNU nano 4.8                                                script_debit2.gnu
set terminal pngcairo enhanced
set output 'variation_debit2.png'

set title "Variation du debit"
set xlabel "debit physique(Mbps)"
set ylabel "debit obtenu(Mbps)"

set grid

plot 'debit2.txt' using 1:2 with linespoints title 'debit obtenu' lw 2 pt 7 lc rgb 'green'
```

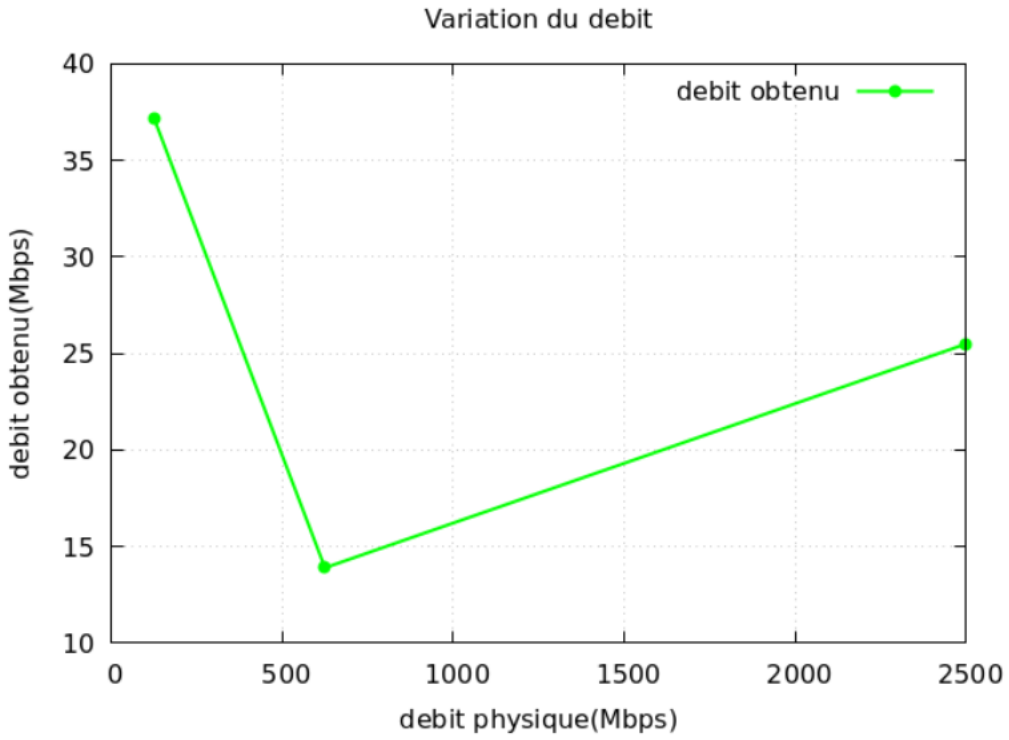
Le fichier texte contenant les valeurs obtenue il sera utiliser par le script ci dessus :

```
  GNU nano 4.8                          debit2.txt                          Modified
125 37.2
625 13.9
2500 25.5
```

Les commandes utilisées :

```
wifi@wifi-virtualbox:~$ nano debit2.txt
wifi@wifi-virtualbox:~$ nano script_debit2.gnu
wifi@wifi-virtualbox:~$ gnuplot script_debit2.gnu
```

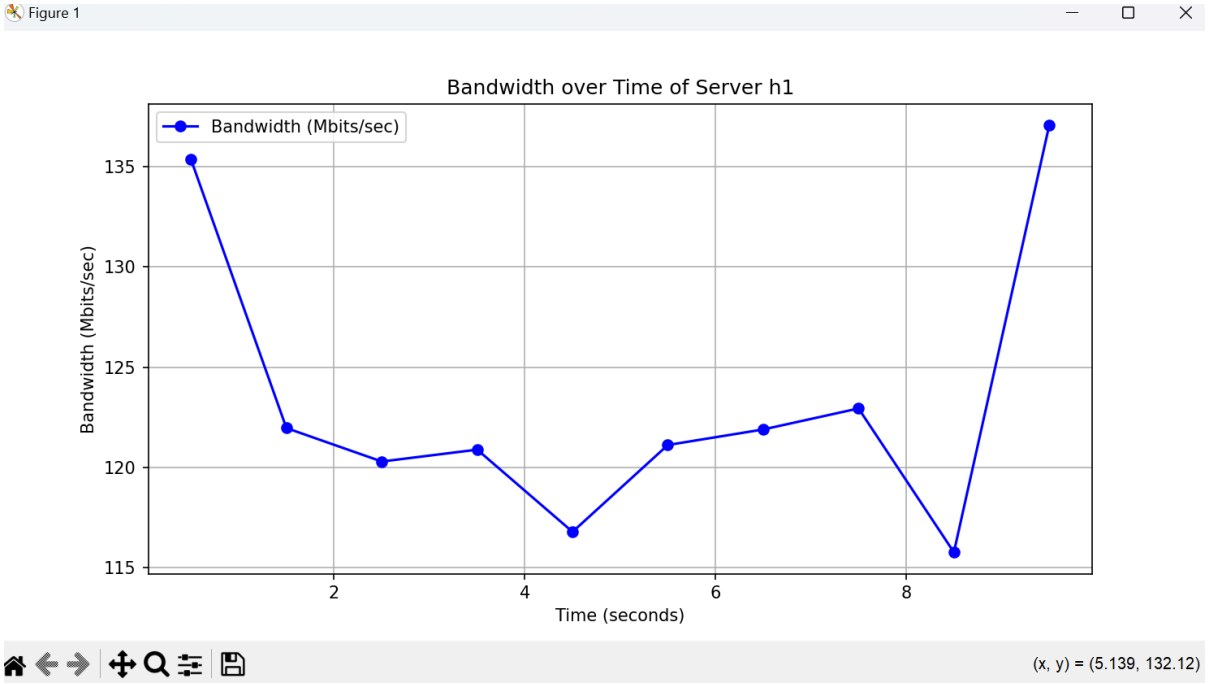
La courbe obtenue :



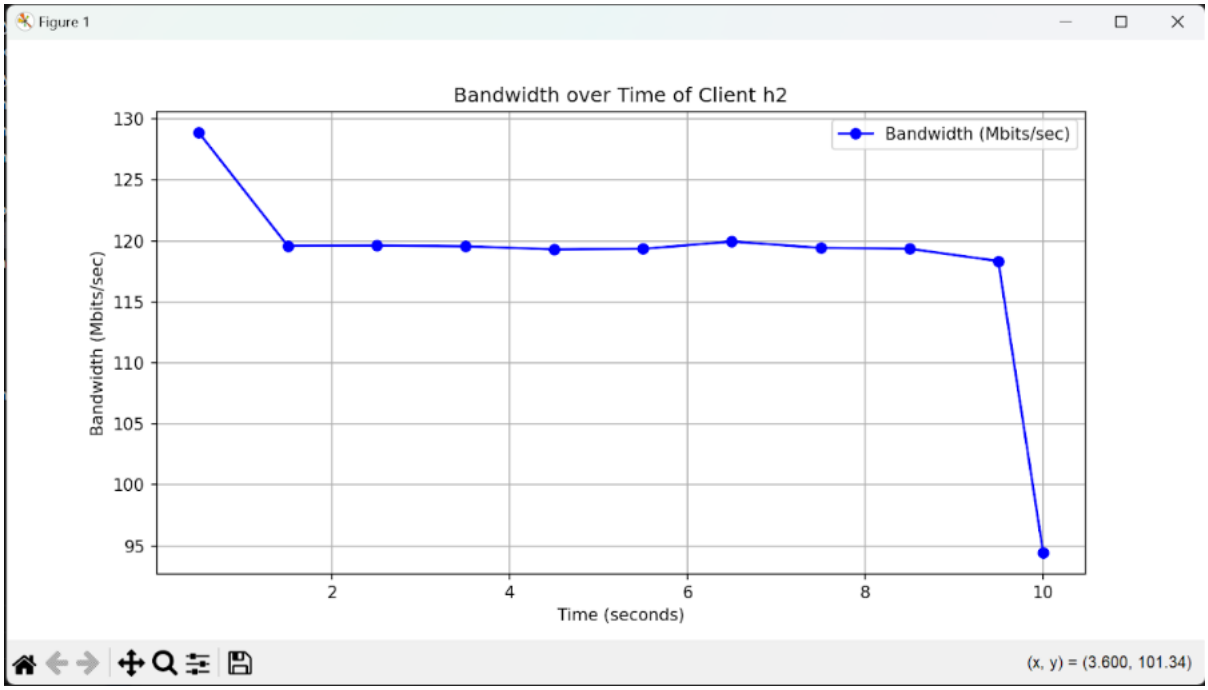
3.2 Une prise d'écran de chaque résultat et une interprétation :

3.2.1 Valeur faible 125Mbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 1000Mbit burst 1540 latency 10ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 125Mbit burst 1540 latency 10ms
mininet> h1 iperf3 -s &
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 49484 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  6.07 MBytes  50.9 Mbits/sec    0   63.6 KBytes
[  5]   1.00-2.00   sec  5.61 MBytes  47.0 Mbits/sec    0   63.6 KBytes
[  5]   2.00-3.01   sec  5.02 MBytes  41.8 Mbits/sec    0   67.9 KBytes
[  5]   3.01-4.00   sec  5.33 MBytes  45.0 Mbits/sec    0   70.7 KBytes
[  5]   4.00-5.00   sec  5.65 MBytes  47.4 Mbits/sec    0   70.7 KBytes
[  5]   5.00-6.00   sec  4.69 MBytes  39.2 Mbits/sec    0   70.7 KBytes
[  5]   6.00-7.00   sec  4.67 MBytes  39.3 Mbits/sec    0   82.0 KBytes
[  5]   7.00-8.00   sec  4.10 MBytes  34.4 Mbits/sec    0   56.6 KBytes
[  5]   8.00-9.04   sec  1.12 MBytes  9.06 Mbits/sec    0   56.6 KBytes
[  5]   9.04-10.00  sec  2.05 MBytes  17.8 Mbits/sec    0   74.9 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  44.3 MBytes  37.2 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  44.0 MBytes  36.9 Mbits/sec                  receiver

iperf Done.
```

Débits définis pour les hôtes :

— la bande passante de h1 est limitée à 1000 Mbits/s, avec une latence de 10 ms et un burst de 1540 kbits.

— h2, en revanche, est limité à 125 Mbits/s, avec les mêmes paramètres de latence et de burst.

Cette configuration crée un goulet d'étranglement où h2 ne peut pas recevoir des données aussi rapidement que h1 peut les envoyer, ce qui met en évidence les effets de limitation de débit et de congestion.

Observations sur les Graphes de Bande Passante :

Fluctuations :

— Dans le premier graph, la bande passante montre une baisse progressive, oscillant autour de 120 à 135 Mbits/s, ce qui correspond étroitement à la limitation imposée pour h2.

— le deuxième, qui montre une stabilisation autour de 120Mbit et une baisse Lorsque h1 tente d'envoyer plus rapidement.

Ces fluctuations et cette diminution progressive sont des signes typiques d'un réseau TCP en train de gérer la congestion. Les protocoles TCP adaptent dynamiquement le débit en fonction des réponses de l'autre hôte. Dans ce cas, la bande passante réelle se stabilise autour du débit maximal que h2 peut gérer (125 Mbits/s). Lorsque h1 tente d'envoyer plus rapidement, la congestion se produit et TCP réduit automatiquement le débit pour éviter la perte de paquets.

Effet de la Limitation Asymétrique de Bande Passante :

— La différence de bande passante maximale entre h1 et h2 impose une restriction où h2 ne peut absorber qu'une fraction de ce que h1 pourrait potentiellement envoyer.

— Ce goulet d'étranglement force h1 à ajuster son débit pour éviter de saturer h2.

Dans un environnement avec des limites asymétriques, le débit effectif du réseau finit par s'ajuster pour correspondre à la capacité de l'hôte le plus lent (ici h2). Les fluctuations dans le graphe sont dues aux tentatives du protocole TCP d'optimiser le débit tout en évitant de dépasser la capacité de h2.

Latence et Burst :

— La latence de 10 ms est relativement faible, ce qui limite les délais de transmission, mais la limitation de débit de h2 reste le facteur dominant.

— Le paramètre de burst, qui définit la quantité maximale de données transmises en une seule rafale, permet de compenser les pertes momentanées de débit, mais il est insuffisant pour compenser la limitation de débit de h2.

La faible latence contribue à maintenir un flux de données constant, mais la bande passante de h2 reste insuffisante pour permettre un débit plus élevé. Le burst est là pour aider en cas de fluctuations temporaires, mais comme le débit est constamment limité, son impact est minimal.

Conclusion :

— Le goulet d'étranglement provoqué par la limitation de débit de h2 force le réseau à s'adapter à une bande passante bien inférieure à la capacité maximale de h1.

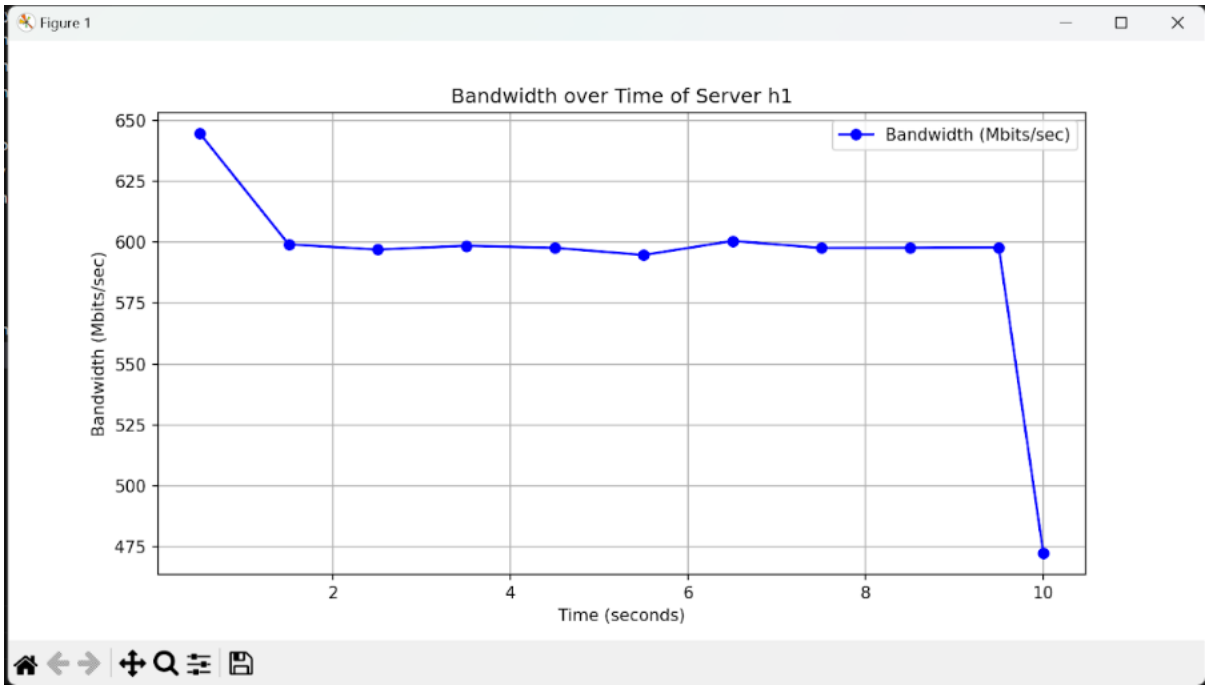
— Le protocole TCP tente d'ajuster automatiquement la bande passante, ce qui entraîne des cycles de congestion et de récupération visibles dans les graphes.

— Les fluctuations autour de 120-135 Mbits/s représentent l'équilibre trouvé par le protocole TCP pour maintenir une transmission stable sans saturer h2.

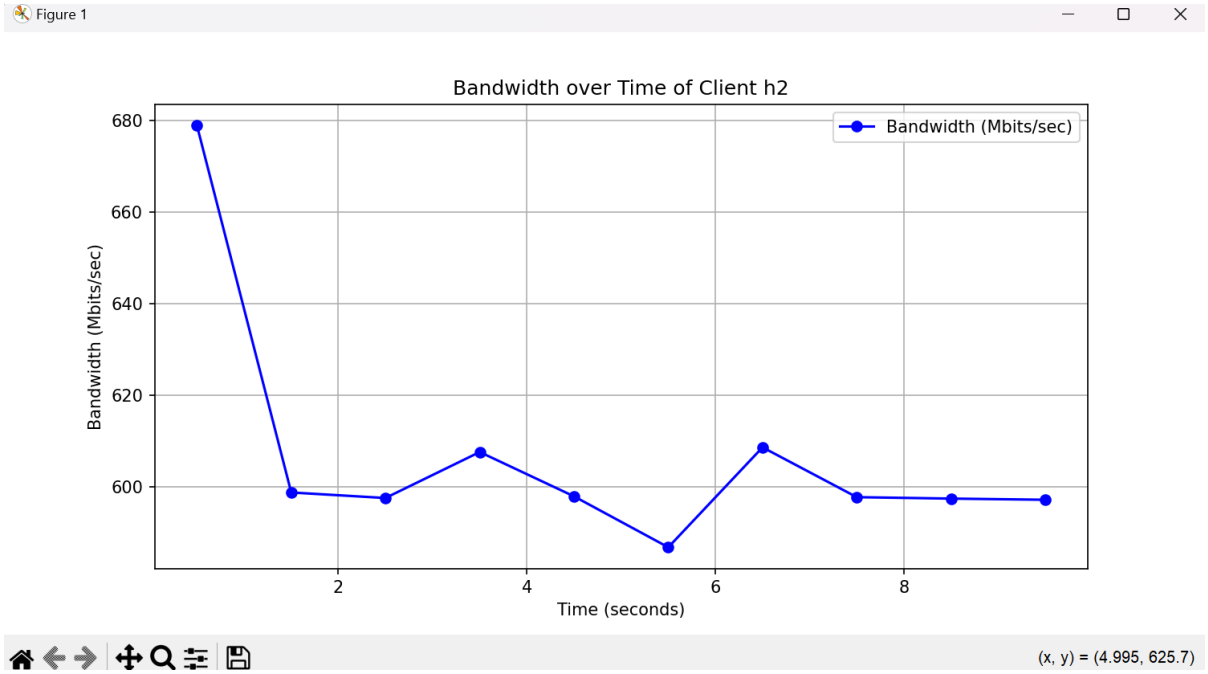
Ces résultats montrent que la gestion de la congestion par le protocole TCP est efficace, mais que la limitation de débit impose une réduction significative des performances globales du réseau, même si h1 pourrait théoriquement supporter un débit bien plus élevé.

3.2.2 Valeur moyenne 625Mbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 1000Mbit burst 1540 latency 10ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 625Mbit burst 1540 latency 10ms
mininet> h1 iperf3 -s &
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 49488 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.02   sec  4.27 MBytes  35.1 Mbits/sec    0   67.9 KBytes
[  5]   1.02-2.14   sec   954 KBytes  6.98 Mbits/sec    0   67.9 KBytes
[  5]   2.14-3.22   sec   954 KBytes  7.22 Mbits/sec    0   67.9 KBytes
[  5]   3.22-4.01   sec   573 KBytes  5.99 Mbits/sec    0   67.9 KBytes
[  5]   4.01-5.03   sec   573 KBytes  4.60 Mbits/sec    0   46.7 KBytes
[  5]   5.03-6.00   sec   764 KBytes  6.43 Mbits/sec    0   60.8 KBytes
[  5]   6.00-7.00   sec  2.24 MBytes  18.8 Mbits/sec    0   65.0 KBytes
[  5]   7.00-8.00   sec  1.12 MBytes  9.37 Mbits/sec    0   65.0 KBytes
[  5]   8.00-9.00   sec   382 KBytes  3.13 Mbits/sec    0   65.0 KBytes
[  5]   9.00-10.00  sec  4.85 MBytes  40.7 Mbits/sec    0   67.9 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  16.6 MBytes  13.9 Mbits/sec    0             sender
[  5]   0.00-10.01  sec  16.2 MBytes  13.6 Mbits/sec                  receiver

iperf Done.
mininet>
```

Interprétation :

Effet de la Limitation de Débit : Le débit moyen configuré à 625 Mbit/s limite la bande passante des transmissions entre les machines h1 et h2. Cette limite fixe un plafond pour la quantité de données pouvant transiter par seconde, ce qui se reflète dans les fluctuations autour de cette valeur dans les graphes. Les variations sont normales dans un environnement simulé, surtout lorsque le système alterne entre les périodes de burst (pic) et de latence.

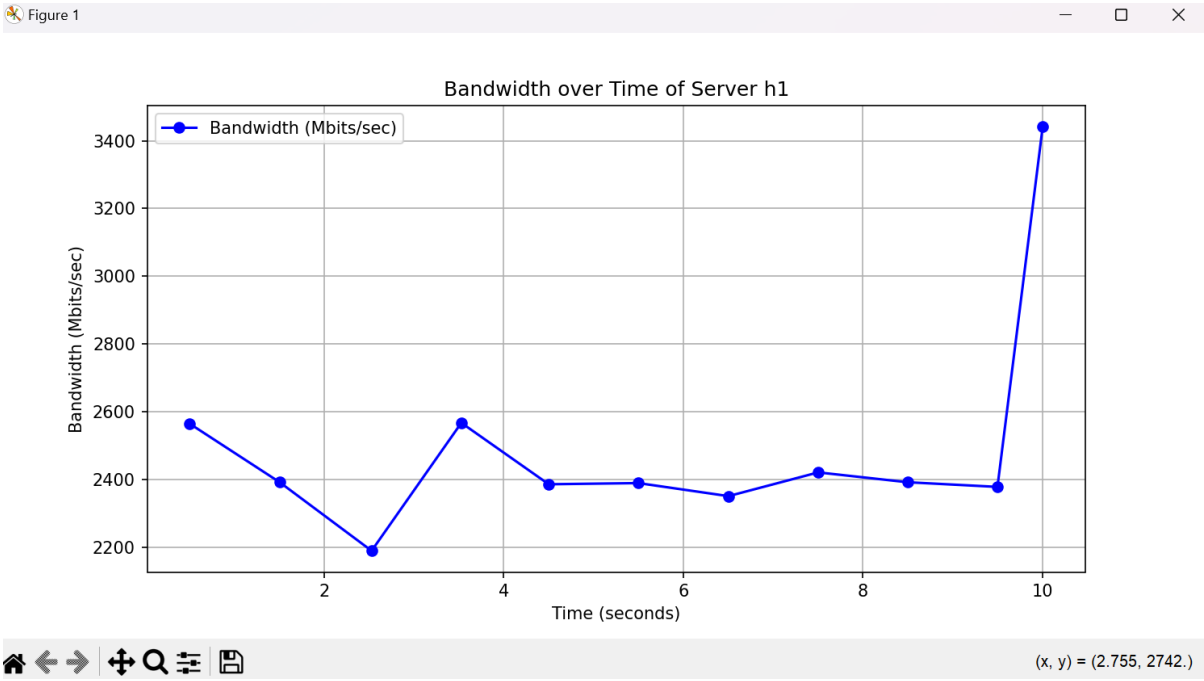
Impact de l'Overhead des Protocoles : À un débit moyen, l'overhead lié aux protocoles réseau (comme TCP/IP) reste présent et réduit légèrement le débit effectif. Les en-têtes et autres informations de contrôle consomment une partie de la bande passante, ce qui explique en partie les légères baisses dans les mesures de débit, car elles occupent de la place sans transmettre de données utiles.

Influence de la Latence et de la Congestion : Dans cette configuration, la latence et la congestion restent faibles mais peuvent affecter les mesures de débit de façon plus visible qu'à faible débit. Cela s'observe dans les variations entre les points de mesure, où l'on note que le débit peut descendre en dessous de 625 Mbit/s. En effet, à ce débit intermédiaire, il est probable que les buffers réseau se remplissent temporairement avant d'être libérés, générant des pics et des creux dans le graphe.

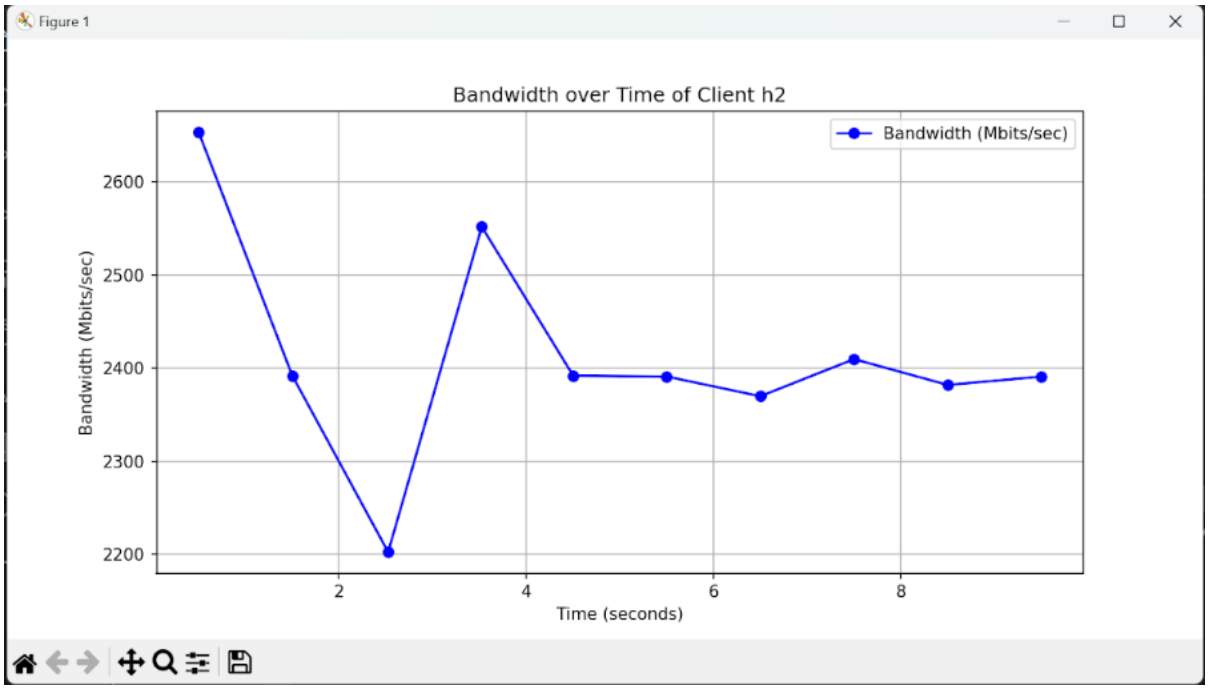
Conclusion :
La configuration de débit intermédiaire montre un comportement où le débit reste globalement stable autour de 625 Mbit/s, avec des fluctuations dues à l'overhead et à l'effet du burst. Les variations observées restent dans les limites attendues, et l'impact de la latence et de la congestion est faible mais plus visible qu'à faible débit, illustrant ainsi comment le débit intermédiaire gère des volumes de données modérés sans saturer les buffers.

3.2.3 Valeur élevée 2500Mbit :

Variation du débit en fonction du temps sur le serveur h1 :



Variation du débit en fonction du temps sur le client h2 :



Résultat de la commande iperf3 :

```
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 2500Mbit burst 1540 latency 10ms
mininet> h2 tc qdisc add dev h2-eth0 root tbf rate 2500Mbit burst 1540 latency 10ms
mininet> h1 tc qdisc del dev h1-eth0 root
mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 1000Mbit burst 1540 latency 10ms
mininet> h1 iperf3 -s &
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
mininet> h2 iperf3 -c 10.5.0.6
Connecting to host 10.5.0.6, port 5201
[  5] local 10.5.0.7 port 49492 connected to 10.5.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  3.18 MBytes  26.6 Mbits/sec    0   77.8 KBytes
[  5]   1.00-2.00   sec  4.10 MBytes  34.5 Mbits/sec    0    119 KBytes
[  5]   2.00-3.12   sec  3.42 MBytes  25.6 Mbits/sec    0    134 KBytes
[  5]   3.12-4.00   sec  2.17 MBytes  20.7 Mbits/sec    0    136 KBytes
[  5]   4.00-5.00   sec  4.85 MBytes  40.7 Mbits/sec    0    144 KBytes
[  5]   5.00-6.01   sec  3.17 MBytes  26.4 Mbits/sec    0    151 KBytes
[  5]   6.01-7.01   sec  2.05 MBytes  17.1 Mbits/sec    0    165 KBytes
[  5]   7.01-8.00   sec  1.93 MBytes  16.3 Mbits/sec    0    165 KBytes
[  5]   8.00-9.00   sec  3.85 MBytes  32.3 Mbits/sec    0    277 KBytes
[  5]   9.00-10.00  sec  1.68 MBytes  14.1 Mbits/sec    0    277 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  30.4 MBytes  25.5 Mbits/sec    0             sender
[  5]   0.00-10.06  sec  29.3 MBytes  24.4 Mbits/sec                  receiver
```

Interpretation :

Overhead des Protocoles : Bien que l'overhead lié aux protocoles réseau (comme TCP/IP) ait un impact, cet effet est relativement négligeable à un débit aussi élevé. En effet, une grande partie de la bande passante est disponible pour le transfert des données utiles, et l'impact des en-têtes de paquets devient proportionnellement moins important.

Congestion et Latence Minimales : À ce niveau de débit, les buffers réseau peuvent être légèrement plus sollicités, mais la latence reste faible en raison de la capacité élevée du lien. Par conséquent, il y a peu d'impact sur la qualité de la transmission des données, et les fluctuations observées sont mineures.

Variabilité et Stabilité : On observe de petites variations dans le débit, mais elles sont relativement faibles, et on remarque une stabilité des variations pour le client h2 autour de 2400Mbits ce qui montre une bonne stabilité globale de la configuration à haut débit.

Interprétation Générale des Trois Cas (Faible, Moyenne et Élevée) :

1. **Effet de l'Overhead des Protocoles :** Dans les trois configurations, l'overhead introduit par les en-têtes TCP/IP a un impact plus visible à faible débit, car chaque bit d'overhead prend une proportion plus importante du total de la bande passante. À mesure que le débit augmente (625 Mbit/s puis 2500 Mbit/s), cet overhead devient négligeable par rapport à la quantité de données transmises, permettant un débit plus proche de la valeur configurée.

2. **Impact de la Latence et de la Congestion :** Dans le cas d'un faible débit, la latence est quasi-inexistante et les risques de congestion sont minimes. À débit moyen, les buffers peuvent légèrement s'emplir mais sans impacts majeurs. Avec un débit élevé, la latence et la congestion peuvent commencer à apparaître si le trafic est intense, mais la capacité du lien permet de les gérer efficacement.

3. **Stabilité et Variabilité :** Pour chaque configuration, la bande passante fluctue légèrement autour de la valeur configurée, plus les variations sont moins importantes en proportion, montrant une meilleure stabilité globale du lien.

4.Comparaison des résultats de T1.1 et T1.2

4.1 Comparaison :

Dans le premier test, h1 et h2 sont directement connectés, avec les débits configurés et les débits obtenus suivants :

- 125 Mb/s configuré → 114 Mb/s obtenu.
- 625 Mb/s configuré → 525 Mb/s obtenu.
- 2,5 Gb/s configuré → 1,57 Gb/s obtenu.

Dans le second test, h1 et h2 sont connectés via un commutateur, avec h1 configuré à un débit constant de 1 Gb/s, et les débits de h2 variant :

- 125 Mb/s configuré → 37,2 Mb/s obtenu.
- 625 Mb/s configuré → 13,9 Mb/s obtenu.
- 2,5 Gb/s configuré → 25,5 Mb/s obtenu.

Les résultats montrent que dans le second test, les débits obtenus sont bien plus faibles pour chaque configuration de h2 par rapport au premier test, et l'écart entre le débit configuré et le débit obtenu est nettement plus important.

4.2 Explication :

1. **Effet du Débit Fixe de h1 (1 Gb/s) :** Dans le second test, le débit constant de h1 à 1 Gb/s crée une limitation intrinsèque qui empêche h2 de recevoir plus que cette valeur, indépendamment de la capacité configurée de h2. Par exemple, même si h2 est configuré à 2,5 Gb/s, il ne peut exploiter qu'une fraction de ce débit en raison de la restriction imposée par h1. Ce goulot d'étranglement réduit le débit maximal potentiel que h2 pourrait atteindre, accentuant la différence entre le débit configuré et le débit obtenu.
2. **Impact du Commutateur :** L'ajout d'un commutateur dans le second test introduit une couche supplémentaire dans le chemin de communication. Le commutateur, en fonction de ses capacités, peut ajouter de la latence, créer des files d'attente et engendrer des pertes de paquets, particulièrement à des débits élevés. Ces limitations réduisent le débit effectif entre h1 et h2, contribuant aux résultats significativement inférieurs dans le second test par rapport au premier. Dans les environnements de réseau où des commutateurs de faible capacité sont utilisés, ces pertes peuvent être amplifiées, particulièrement pour des flux de données exigeants en bande passante.
3. **Comparaison des Pertes de Débit dues aux Protocole et à l'Infrastructure :** Dans le premier test, les pertes de débit sont principalement dues aux frais de protocole (overhead) et aux limitations des équipements en lien direct. En revanche, dans le second test, ces pertes sont exacerbées par l'addition du commutateur et par la configuration asymétrique des débits, où h1 est limité à 1 Gb/s. Cela engendre des limitations matérielles et une gestion supplémentaire des flux de données, qui réduisent davantage le débit obtenu.

5.Conclusion

En résumé, les résultats montrent que le débit constant de h1 à 1 Gb/s, combiné aux limitations introduites par le commutateur, engendre une réduction plus marquée du débit dans le second test par rapport à une connexion directe. Ces facteurs démontrent l'importance de la cohérence entre les débits configurés et les capacités des équipements intermédiaires, ainsi que de la symétrie des débits entre les machines pour optimiser la performance réseau.
[3, 2, 1].

Bibliographie

[1] OVERLEAF. *Learn LaTeX in 30 minutes*. Accessed : 2024-11-06. URL : https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

[2] CCNA REPONSES. *Introduction aux réseaux - Module 4 : Couche physique*. Accessed : 2024-11-06. URL : https://ccnareponses.com/introduction-aux-reseaux-modules-4-couche-physique/.

[3] Cisco SYSTEMS. *Troubleshooting ICMP Ping*. Accessed : 2024-11-06. URL : https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html.