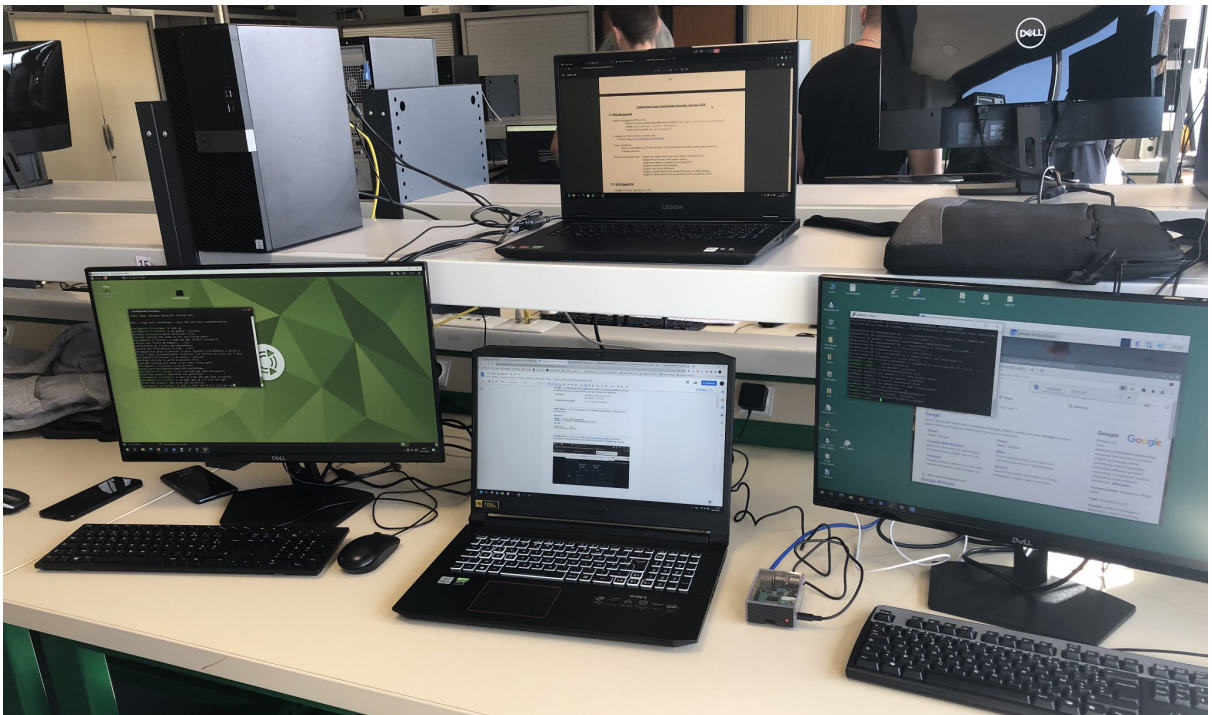


SAé 301 :
Mettre en oeuvre un système de transmission



1) Introduction

Saé, qui signifie *Situation d'apprentissage et d'évaluation* sous forme d'acronyme, est le nom du programme d'évaluation annuel portant sur les compétences acquises par les étudiants. C'est un élément central du processus d'obtention du BUT, nous fournissant la validation des différentes compétences qu'on acquiert tout au long de l'année scolaire dans les différentes ressources. Cette Saé nommée « **Mettre en œuvre un système de transmission** ». Tout au long de cette situation d'apprentissage et d'évaluation, nous avons pu mettre à profit notre savoir-faire informatique/réseau afin de configurer un **Raspberry Pi**, des **machines virtuelles** et mettre en place divers **services** et **serveurs** (Munin, VoD, VPN...). Mais encore de développer les performances de ces derniers de manière à qu'il puissent avoir une meilleure optimisation. Nous avons également effectué certaines **analyses de débit** (données, vidéo, audio) afin de faire des comparaisons et conclure avec un raisonnement logique.

2) Besoin du projet.

Ce **besoin** doit respecter certaines consignes :

- ☐ Lecture de documents techniques ;
- ☐ Config **Raspi 4** et **VM** ;
- ☐ Respect du rendu du cahier des charges ;
- ☐ Mise en place de l'environnement de développement;
- ☐ Réalisation documentée (**rapport + jalons**);
- ☐ **Répartition** des tâches;

Sommaire

I] Système de Transmission :

1. Matériel mis à disposition	Page.....3-4
2. Organisation du projet	Page.....5-6
3. Outils/Services mis en place	Page.....7-8
4. Débit d'une source d'information	Page.....8-9
5. Système d'analyse de débit : Munin - Iperf 3	Page.....10
6. Serveur vidéo à débit adaptatif	Page.....11
7. Serveur VPN	Page.....12
8. Serveur NAS	Page...13-14
9. Datacenter	Page.....14
10. Conclusion	Page.....15

II] Annexe :

1. Ressource utilisé	Page.....16
2. Sources	Page.....16

I. *Système de Transmission :*

1) Matériel mis à disposition

Le matériel mis à notre disposition était très utilisé dans le domaine des réseaux. Nous pouvons séparer celui-ci en deux catégories.

La première catégorie correspond aux équipements réseaux et logiciels. Nous avons eu recours au :

- Portail Captif (**DHCP**)
- **VNC Viewer**
- **Putty**
- **Virtual Box**

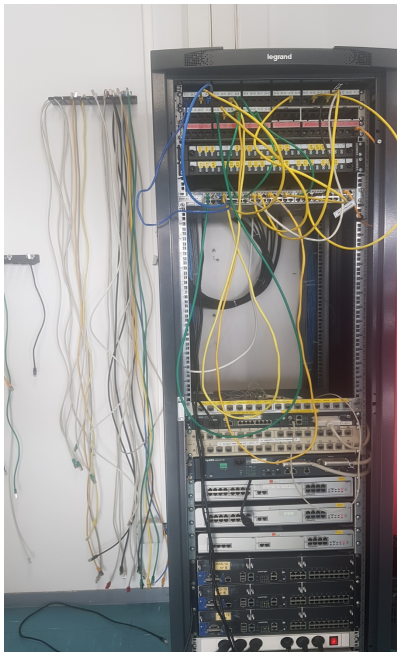


Figure 1 : Le portail captif permet à nos machines d'avoir des adresse attribué par DHCP.



Figure 2 : Ce logiciel permet d'accéder à un bureau d'ordinateur à distance via le protocole RFB.

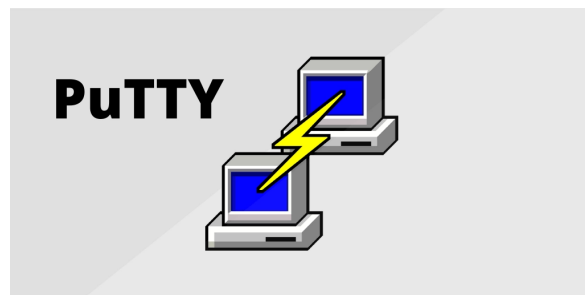


Figure 3 : PuTTY est un émulateur de terminal pour Windows permettant la connexion à une machine diistante par protocole ssh.



VirtualBox

Figure 4 : VirtualBox est le logiciel de virtualisation d'Oracle. Celui-ci permet d'héberger une ou plusieurs machines virtuelles, avec des systèmes d'exploitation différents.

La deuxième catégorie serait le matériel de bureautique tel que :

- ❖ Les **2 ordinateurs**
- ❖ Le **matériel de stockage** avec une carte micro-SD de 16go contenant une image de type **BUSTER**
- ❖ Un **Raspberry Pi 4** utilisant la carte micro-SD citée précédemment
- ❖ Une **alimentation** USB-C 5V pour le Raspberry Pi
- ❖ Un **écran** ainsi que le câble **HDMI** vers **micro HDMI**, pour voir l'interface graphique du Raspberry Pi 4



Figure 5 : Le Raspberry Pi 4 est un nano-ordinateur qui se présente sous la forme d'une carte de processeur. Il peut se brancher directement sur un écran.

2) Organisation du projet

L'organisation est une **clé majeure** pour une bonne réussite du projet. Nous avons alors pris le temps de répartir les tâches du projet et d'attribuer à chacun sa propre section. Cela a conduit à des progrès réussis car nous avons donné la priorité à notre **travail d'équipe** plutôt qu'à la tâche solitaire. Attribuer des sections individuelles aux seins du groupe s'est avéré être un moyen efficace de vérifier la compréhension des instructions.

Afin de mener à bien ce projet, nous avons utilisé plusieurs **outils d'organisation** :

Dans un premier temps, le logiciel **Discord**. Grâce à cet outil, nous avons pu créer un serveur de discussion permettant le **travail collaboratif** et les **réunions** du soir pour le compte rendu. Dans ce serveur, plusieurs salons ont été créés (voir fig. 6). Le salon qu'on a le plus utilisé a été le salon **brouillon**, comme son nom l'indique nous avons une totale liberté dessus, puisque nous y mettons tout type de traces (photo, screen, texte, et bien autre).

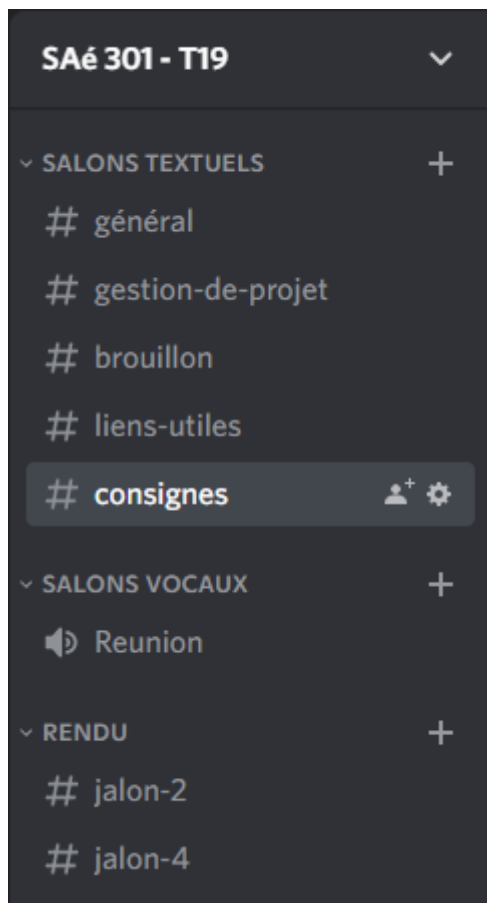


Figure 7 : Discord est un logiciel propriétaire gratuit de VoIP et de messagerie instantanée. Il fonctionne sur les systèmes d'exploitation Windows, macOS, Linux, Android, iOS ainsi que sur les navigateurs web

Figure 6 : Le salon reunion nous à permis un travail en distanciel le soir, un partage d'écran étant possible, nous avions une image claire du travail de chacun d'entre nous

Nous avons également utilisé **Monday.com**, qui est un logiciel très adapté pour la gestion de projet. Grâce à ce site Web, nous sommes en mesure de mettre à jour l'avancement des travaux restants et d'ajouter plus d'informations, le **statut** de la tâche est renseigné avec un **code couleur**.

Pour chaque division nous y renseignons :

- Un **groupe d'éléments**, ce qui nous permet de différencier et séparer les tâches dans le **bon ordre** ;
- Son **état** d'avancement ;
- Quelles sont les **personnes** ayant réalisées la pratique ;
- Date de **début/fin** ;
- Statut du **rendu** du jalon ;

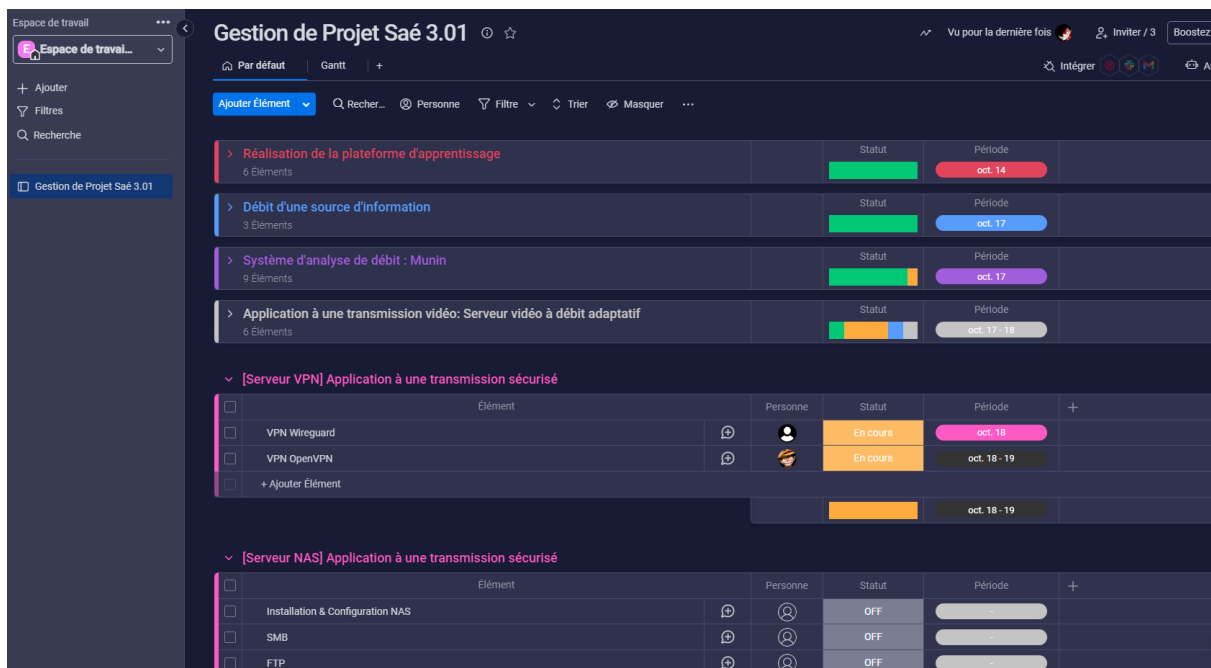


Figure 8 : Voici un exemple de notre actuelle avancé sous forme de tableau daté du 18/10/2022



Figure 9 : Le même tableau de la Figure 6 sous forme de gant daté du 18/10/2022

3) Outils/Services mis en place

Nous avons mis en place l'outil de surveillance **Munin**, qui fournit des **résultats/informations** pertinentes sous formes de **graphique** en page web, il permet de personnaliser et faire les informations fournies par l'installation de plugins.

Serveur : Munin : Centralise les infos fournies par les nœuds et crée des graphes.

Client : Munin-node : collecte les infos auprès de l'élément à supervisor.

Par défaut le logiciel Munin prélève les valeurs toutes les 5min, mais pour **diminuer** le travail du serveur et **optimiser** ce fonctionnement on utilise **FCGI** afin de générer des informations seulement à la demande.

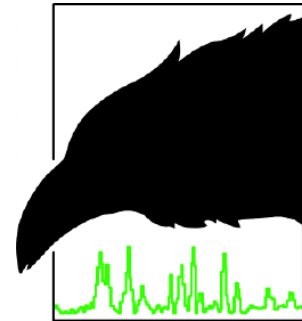


Figure 10 : Munin

Nous avons utilisé Nginx comme serveur web à défaut d'utiliser Apache. On l'a utilisé pour nos deux sites, c'est -à -dire Munin sur le **port 80** et le Serveur VoD sur le **port 8080**.
(Le serveur Vod est désormais disponible en https sous le port 443)



Figure 11 : Nginx

192.168.33.119

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Figure 12 : Page d'accueil Nginx

L'étape suivante était de mettre en place un **Serveur VoD** à débit adaptatif **DASH**.
DASH est un protocole de **streaming adaptatif** : il permet de changer le débit de la vidéo en fonction des **performances réseau**, afin que la vidéo ne soit pas interrompue lors de la lecture.
C'est la qu'entre en jeu **Shaka-player**, qui à été initialisé en lien depuis l'index vers un mpd externe.
Shaka-player permet d'**adapter la résolution vidéo** en fonction du débit reçu.



Figure 13 : Video On Demand

4) Débit d'une source d'information

Nous avons réalisé ici différentes mesures sur le débit ainsi que leurs variations :

- La **radio** a un débit constant et a un petit débit comme afficher en haut à gauche

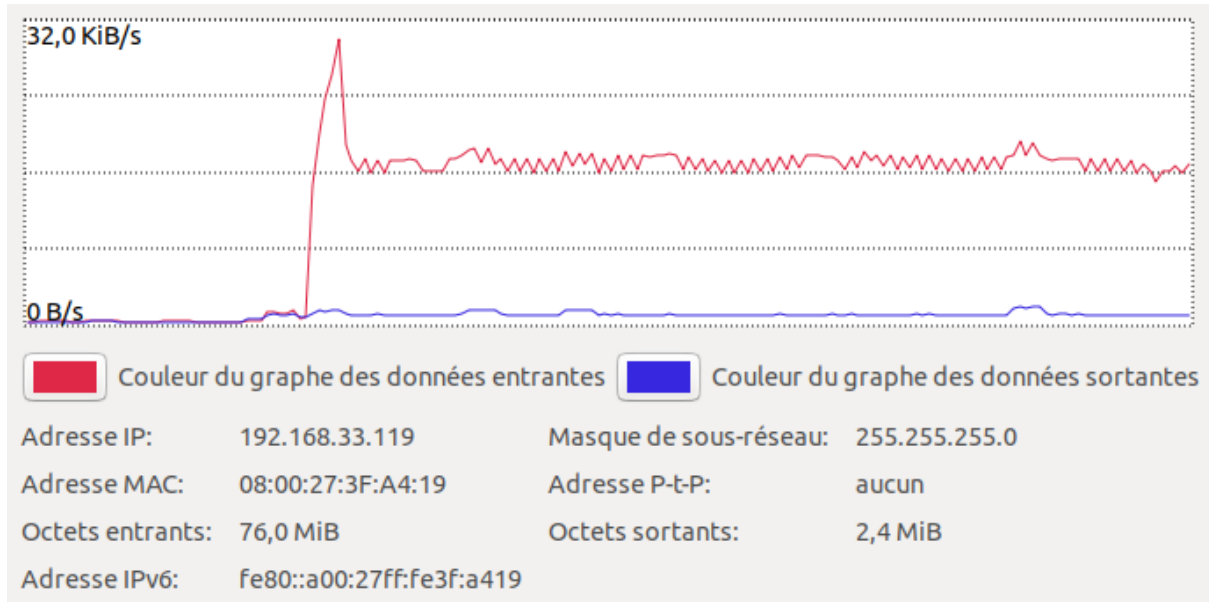


Figure 14 : mesure avec la radio.

- La **vidéo** n'a pas un débit constant car la vidéo se charge avec un cache, quand le cache a fini de se charger, le débit **s'arrête** ou **diminue** on remarque que le débit est plus élevé que celui de la radio

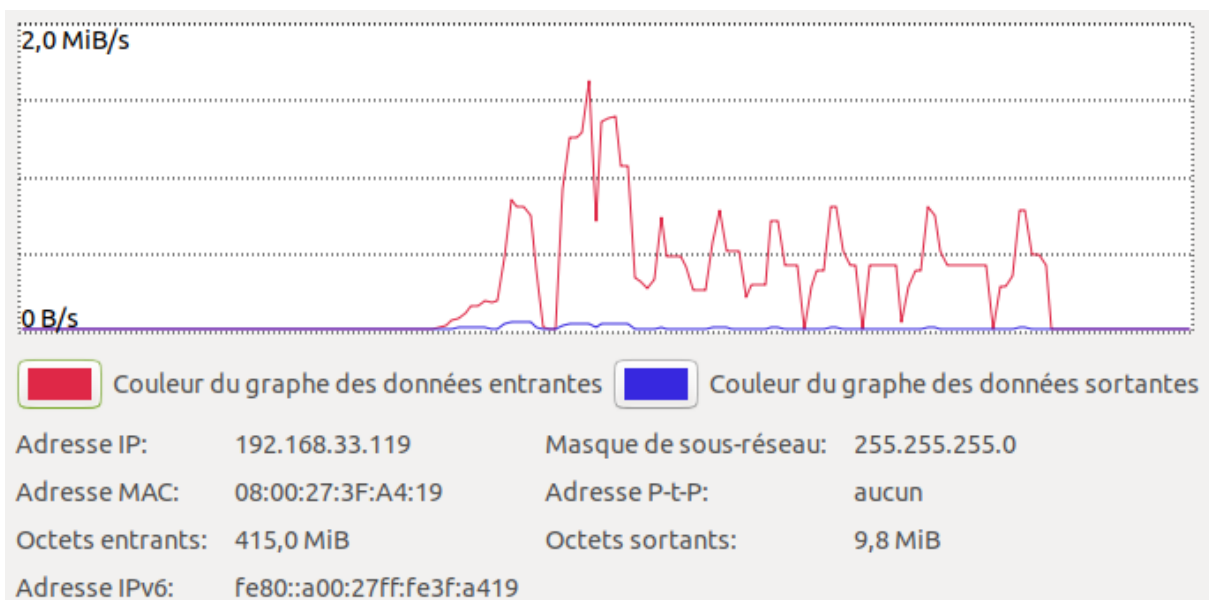


Figure 15 : mesure avec la vidéo

- Le **live** est **constant** comme la radio mais a un **débit plus important** que celle ci

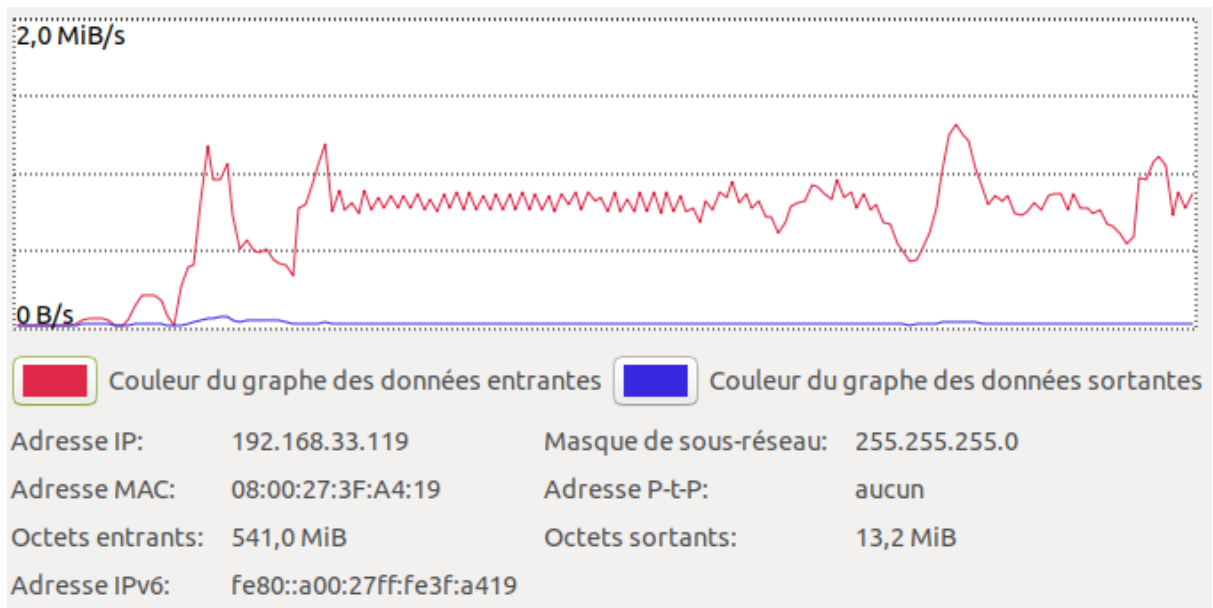


Figure 16 : mesure avec le live (en direct)

- Le **téléchargement** est la mesure avec le **plus haut débit** mais il n'est pas constant

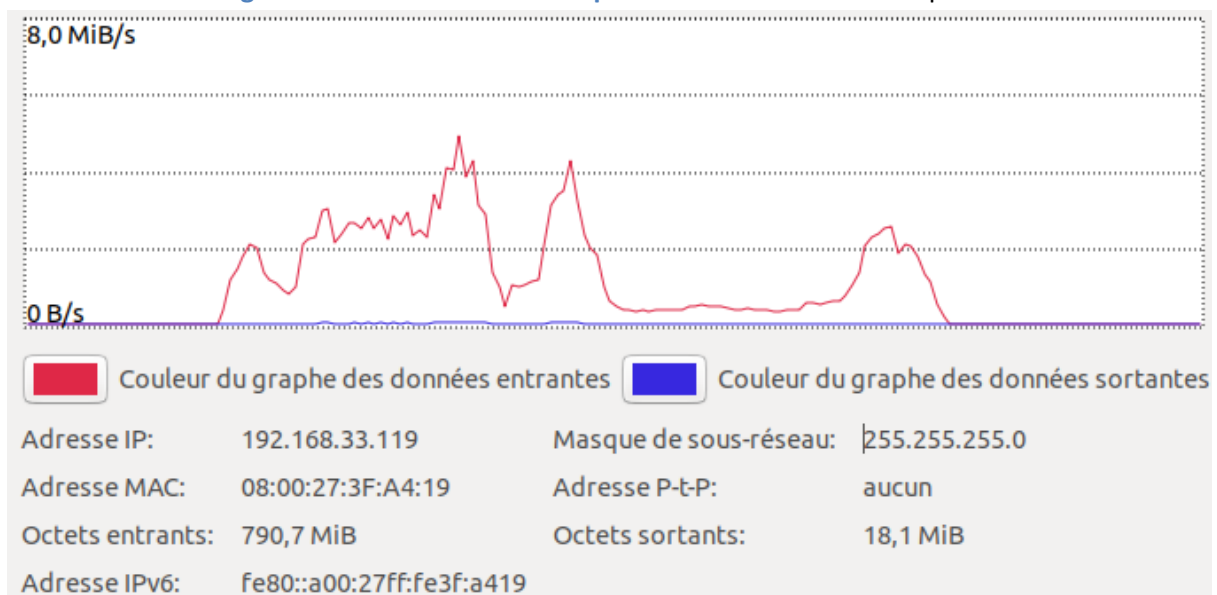


Figure 17 : mesure du téléchargement

5) Système d'analyse de débit : Munin | Iperf3

Iperf est un logiciel informatique permettant la **mesure** de différentes variables d'une connexion **réseau IP**, en effet durant cette SAE nous avons dû l'utiliser à maintes et maintes reprises durant les différentes mesures. Cela nous a permis de créer une plateforme de développement pour les **échanges/débit**, du serveur VoD et des différents VPN.

```
daelk@daelk-VirtualBox:~$ iperf3 -c 192.168.33.19
Connecting to host 192.168.33.19, port 5201
[ 5] local 192.168.33.119 port 40892 connected to 192.168.33.19 port 5201
[ ID] Interval           Transfer     Bitrate      Retr    Cwnd
[ 5]  0.00-1.00      sec   14.5 MBytes  122 Mb/s      0    604 KBytes
[ 5]  1.00-2.00      sec   11.2 MBytes  94.4 Mb/s      0   1.15 MBytes
[ 5]  2.00-3.00      sec   11.2 MBytes  94.4 Mb/s      0   1.71 MBytes
[ 5]  3.00-4.00      sec   11.2 MBytes  94.4 Mb/s      0   2.26 MBytes
[ 5]  4.00-5.00      sec   10.0 MBytes  83.9 Mb/s      0   2.71 MBytes
[ 5]  5.00-6.00      sec   11.2 MBytes  94.4 Mb/s      0   2.71 MBytes
[ 5]  6.00-7.00      sec   11.2 MBytes  94.4 Mb/s      0   2.71 MBytes
[ 5]  7.00-8.00      sec   11.2 MBytes  94.3 Mb/s      0   2.71 MBytes
[ 5]  8.00-9.00      sec   11.2 MBytes  94.4 Mb/s      0   2.71 MBytes
[ 5]  9.00-10.00     sec   11.2 MBytes  94.4 Mb/s      0   2.71 MBytes
- - - - -
[ ID] Interval           Transfer     Bitrate      Retr
[ 5]  0.00-10.00      sec   115 MBytes  96.1 Mb/s      0
[ 5]  0.00-10.26      sec   114 MBytes  93.4 Mb/s
sender
receiver
```

Figure 18 : test iperf3

Munin est un **outil de surveillance** système et réseau, voici donc un screen d'une analyse sur Munin, nous avons ici mesuré le débit d'Iperf 3 sous forme de **graphique** :

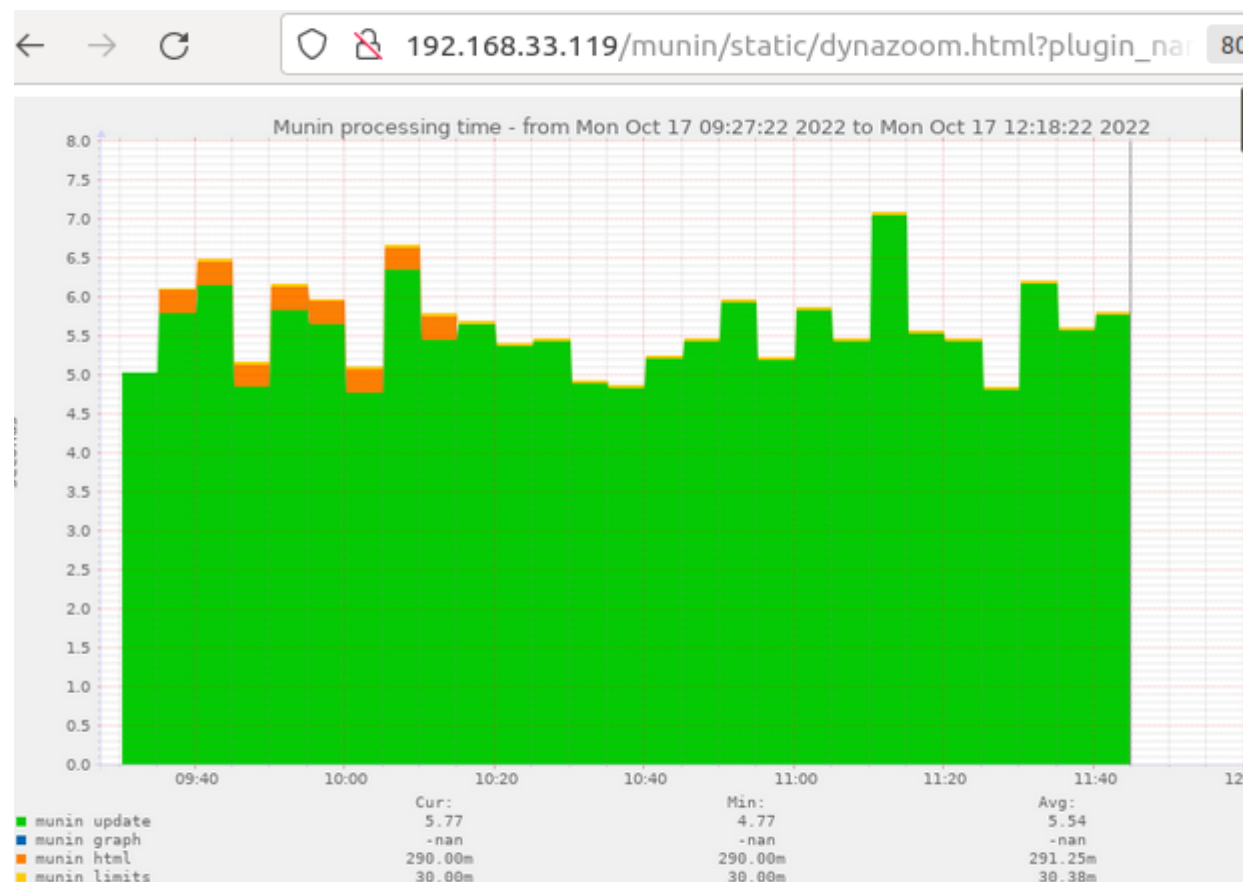


Figure 19 : Graphique Munin

6) Serveur vidéo à débit adaptatif

La **segmentation** était une partie importante pour la finalité du serveur VoD. L'encodage en **différente résolution** a été utile pour l'adaptation de la qualité vidéo sur le serveur VoD.

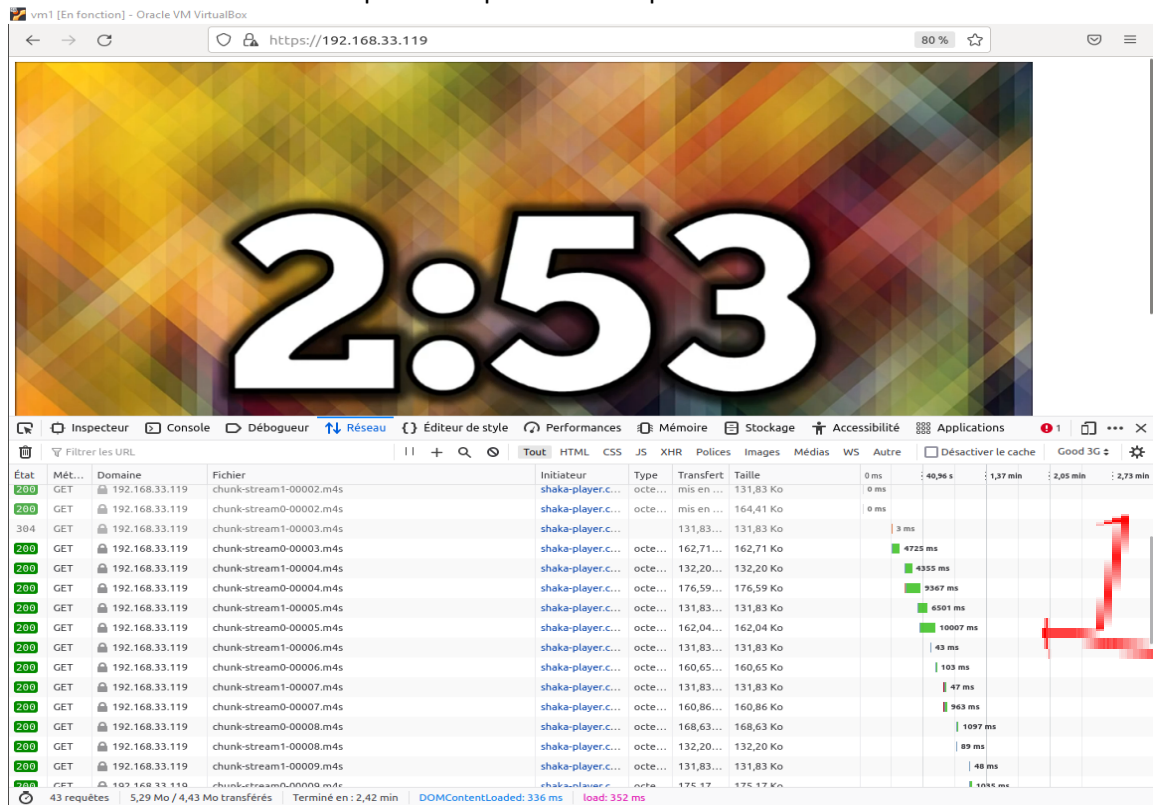


Figure 20 : Page serveur VOD

Le fait de changer le débit a un impact sur la latence du réseau ce qui influence la fluidité de la vidéo. Comme on peut le voir ci-dessous, la latence augmente car le débit a été modifié.

	131,83...	131,83 Ko		3 ms					
octe...	162,71...	162,71 Ko		4725 ms					
octe...	132,20...	132,20 Ko		4355 ms					
octe...	176,59...	176,59 Ko		9367 ms					
octe...	131,83...	131,83 Ko		6501 ms					
octe...	162,04...	162,04 Ko		10007 ms					
octe...	131,83...	131,83 Ko		43 ms					
octe...	160,65...	160,65 Ko		103 ms					

Figure 21 : Évolution de la latence

7) Serveur VPN

Nous avons mis en place **2 serveur VPN** et nous avons pu les comparer, donc **Wireguard** à une vitesse supérieure à **OpenVPN**

```
pi@SAE31-19rpi:~ $ sudo wg
interface: wg0
  public key: 2wtqSA1HlycvSPpRx8DoZSfyEr0Ug/eFvCjzyuObDBU=
  private key: (hidden)
  listening port: 51608

peer: GeKSUzQYwIrTcUUtdOq/B4vhDJwXnoEj9x+KrfOtXRQ=
  endpoint: 192.168.33.119:56314
  allowed ips: 10.0.0.1/32
  latest handshake: 4 minutes, 15 seconds ago
  transfer: 2.11 KiB received, 2.17 KiB sent
pi@SAE31-19rpi:~ $
```

Figure 22 : Screen de la config de Wireguard

```
daelk@daelk-VirtualBox:~$ iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 5] local 10.0.0.1 port 46852 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 5] 0.00-1.00 sec  12.5 MBytes 105 Mb/s   0     577 KBytes
[ 5] 1.00-2.00 sec  12.1 MBytes 101 Mb/s   0     1.10 MBytes
[ 5] 2.00-3.00 sec  10.0 MBytes 83.9 Mb/s  0     1.63 MBytes
[ 5] 3.00-4.00 sec  11.2 MBytes 94.4 Mb/s  0     2.17 MBytes
[ 5] 4.00-5.00 sec  11.1 MBytes 93.0 Mb/s  0     2.69 MBytes
[ 5] 5.00-6.00 sec  10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
[ 5] 6.00-7.00 sec  11.2 MBytes 94.4 Mb/s  0     2.71 MBytes
[ 5] 7.00-8.00 sec  10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
[ 5] 8.00-9.00 sec  11.2 MBytes 94.4 Mb/s  0     2.71 MBytes
[ 5] 9.00-10.00 sec 10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 5] 0.00-10.00 sec  110 MBytes 91.9 Mb/s   0
[ 5] 0.00-10.25 sec  109 MBytes 89.6 Mb/s
```

Figure 23 : Test Iperf3 sur le débit serveur VPN Wireguard

Figure 24 : Test Iperf3 sur le débit serveur VPN OpenVPN

The image shows two terminal windows side-by-side. The left window shows the output of an iperf3 test on a Wireguard server, and the right window shows the output of an iperf3 test on an OpenVPN server.

Left Window (Wireguard):

```
daelk@daelk-VirtualBox:~$ iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 5] local 10.0.0.1 port 46852 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 5] 0.00-1.00 sec  12.5 MBytes 105 Mb/s   0     577 KBytes
[ 5] 1.00-2.00 sec  12.1 MBytes 101 Mb/s   0     1.10 MBytes
[ 5] 2.00-3.00 sec  10.0 MBytes 83.9 Mb/s  0     1.63 MBytes
[ 5] 3.00-4.00 sec  11.2 MBytes 94.4 Mb/s  0     2.17 MBytes
[ 5] 4.00-5.00 sec  11.1 MBytes 93.0 Mb/s  0     2.69 MBytes
[ 5] 5.00-6.00 sec  10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
[ 5] 6.00-7.00 sec  11.2 MBytes 94.4 Mb/s  0     2.71 MBytes
[ 5] 7.00-8.00 sec  10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
[ 5] 8.00-9.00 sec  11.2 MBytes 94.4 Mb/s  0     2.71 MBytes
[ 5] 9.00-10.00 sec 10.0 MBytes 83.9 Mb/s  0     2.71 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 5] 0.00-10.00 sec  110 MBytes 91.9 Mb/s   0
[ 5] 0.00-10.25 sec  109 MBytes 89.6 Mb/s
```

Right Window (OpenVPN):

```
pi@SAE31-19rpi:~/Desktop$ iperf3 -c 10.0.0.1
Connecting to host 10.0.0.1, port 5201
[ 5] local 10.0.0.2 port 45852 connected to 10.0.0.1 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 5] 0.00-1.00 sec  9.16 MBytes 76.8 Mb/s   0     502 KBytes
[ 5] 1.00-2.00 sec  10.6 MBytes 88.9 Mb/s   91     288 KBytes
[ 5] 2.00-3.00 sec  10.6 MBytes 88.9 Mb/s   0     310 KBytes
[ 5] 3.00-4.00 sec  10.5 MBytes 87.8 Mb/s   1     235 KBytes
[ 5] 4.00-5.00 sec  10.5 MBytes 87.9 Mb/s   64     195 KBytes
[ 5] 5.00-6.00 sec  10.5 MBytes 87.8 Mb/s   18     126 KBytes
[ 5] 6.00-7.00 sec  10.5 MBytes 87.9 Mb/s   0     176 KBytes
[ 5] 7.00-8.00 sec  9.92 MBytes 83.2 Mb/s   31     170 KBytes
[ 5] 8.00-9.00 sec  10.5 MBytes 87.9 Mb/s   0     208 KBytes
[ 5] 9.00-10.00 sec  9.86 MBytes 82.7 Mb/s   5     187 KBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 5] 0.00-10.00 sec  104 MBytes 87.5 Mb/s   210
[ 5] 0.00-10.04 sec  103 MBytes 86.3 Mb/s
```

8) Serveur NAS

Un serveur NAS (Network Attached Storage), est un périphérique de stockage, il est connecté au réseau, ce qui lui permet de stocker de grandes quantités de données. Ceux-ci se trouvent généralement dans les entreprises et les maisons. Nous arrivons donc sur openmediavault :

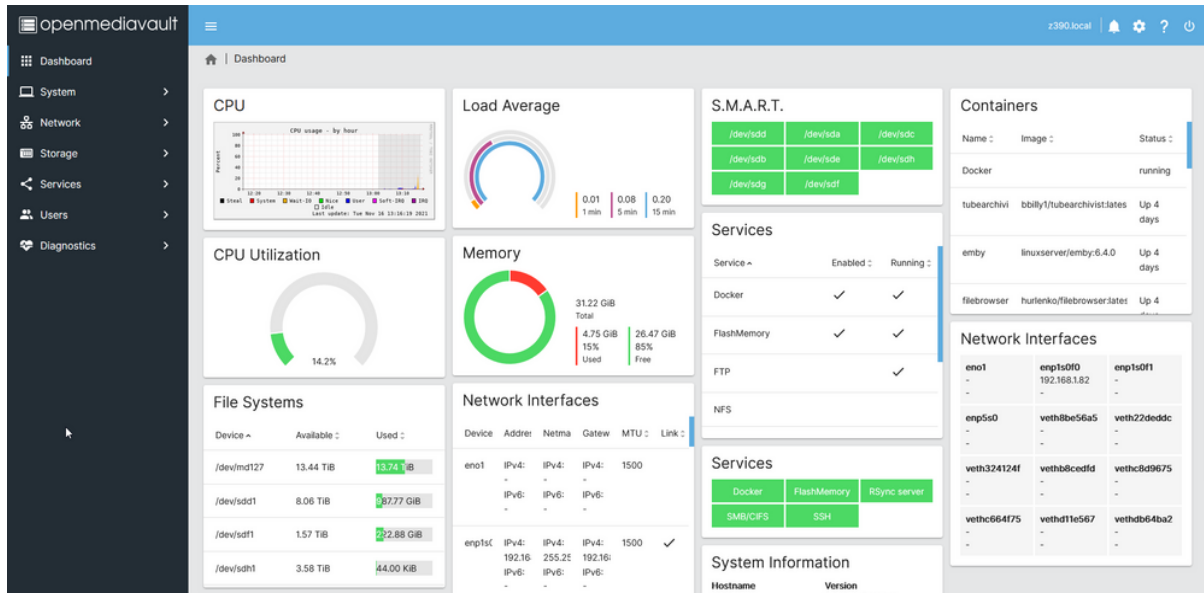


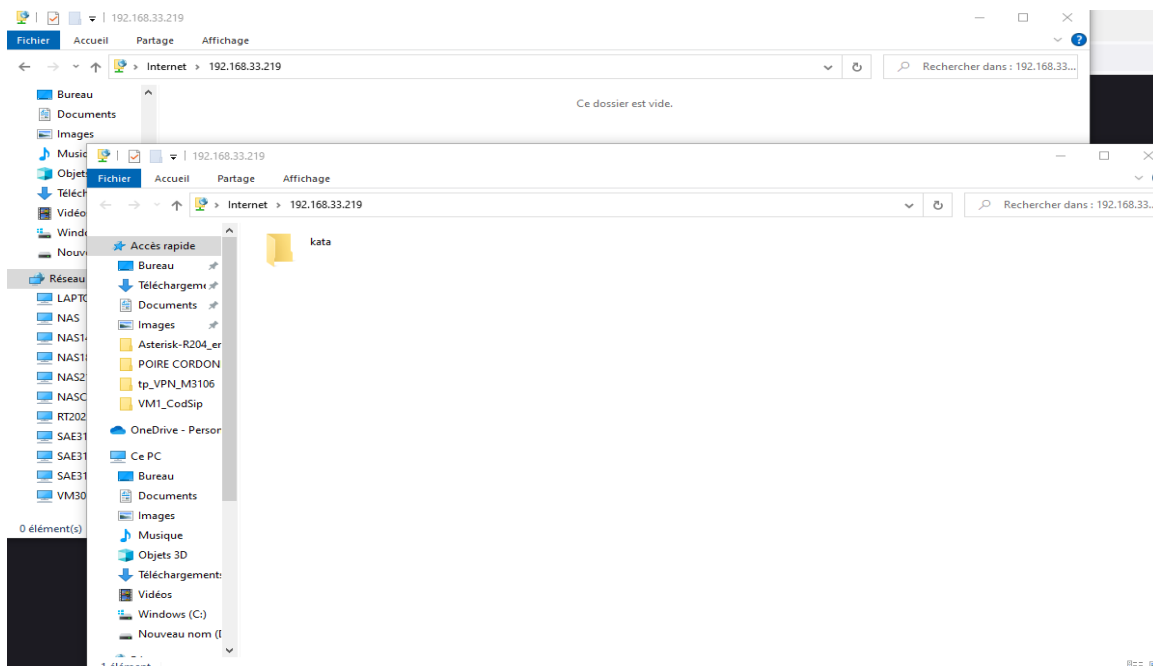
Figure 25 : OpenMediaVault

La réalisation d'un partage de fichier **samba** et **ftp** était la première étape.

Ensuite nous devons changer le nom de la machine **openmedia** ainsi que plusieurs paramètres, on devait créer un **fichier de partage** pour après créer des **user** pour tester si le système de partage de fichier fonctionnait.

Son utilisation reste plutôt simple, on avait besoin de 2 clic pour faire fonctionner samba et ftp

Figure 26 : Screen répertoire de partage



Nous pouvons voir dans le réseau notre **partage de fichier** qui est **VM301-19**, en cliquant dessus, nous pourrions nous identifier à l'aide du login et mot de passe préalablement défini, afin d'accéder aux différents fichiers.

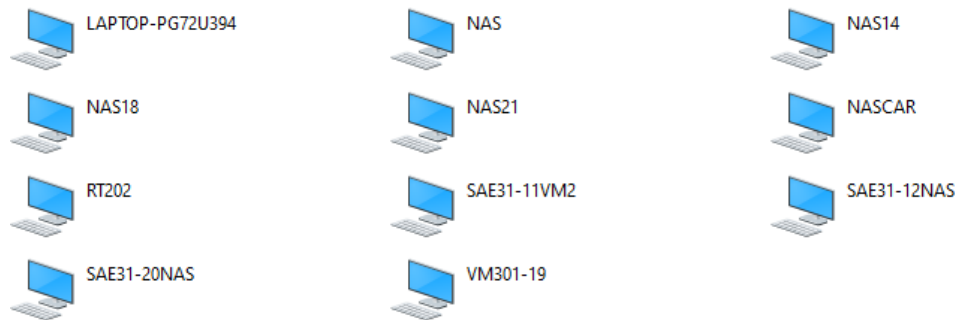
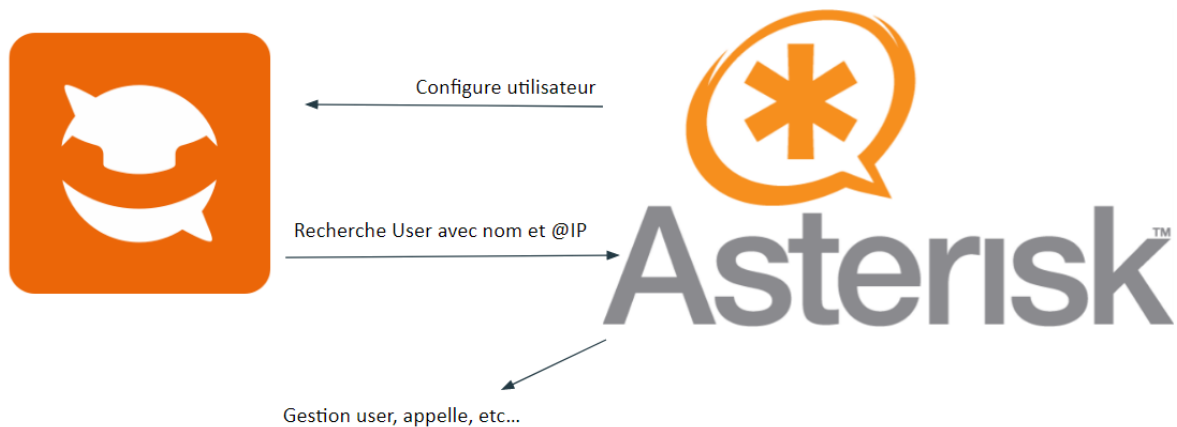


Figure 27 : mesure avec le live (en direct)

8) Bonus : Datacenter

Le bonus consiste à installer Asterisk sur la VM située dans Proxmox suite à cela il faudra créer des **utilisateurs** sur le serveur **Asterisk**, il faut modifier son IP afin de le mettre dans le même réseau. Ensuite sur la VM Ubuntu il faut installer **Linphone** pour pouvoir se connecter aux différents Users créés afin de pouvoir appeler grâce au serveur asterisk qui ce dernier gèrera les différents appels.



9) CONCLUSION

Cette SAÉ nous a permis de développer nos connaissances/compétences acquis l'année passée, notamment dans le domaine des réseaux et système de transmission. Mais également la gestion de projet, car dans une Saé de cette ampleur, il faut savoir s'organiser et répartir équitablement le travail afin de réaliser les rendus dans le temps imparti. Les ressources de gestion de projet, d'Expression-Culture-Communication et de Projet Personnel et Professionnel nous ont permis de réaliser des jalons clairs ainsi qu'un dossier compréhensible. Certains problèmes nous ont empêché d'avancer aussi rapidement que nous le voulions, c'est pour cela qu'il a fallu faire preuve de flexibilité et de coordination dans le groupe. Cependant lors de cette Saé, les problèmes ont été moins importants que dans les précédentes Saé, ce qui vient d'une meilleure méthodologie de travail. Cela nous montre l'évolution durant nos précédentes SAE.

II. Annexe :

1) Ressources utilisées

Les ressources utilisées sont les suivantes :

- [R303] | Services réseaux avancés
- [R305] | Chaînes de transmissions numériques
- [R311] | Anglais professionnel
- [R312] | Expression-Culture-Communication professionnelles : Savoir collaborer
- [R313] | Projet Personnel et Professionnel
- [R315] | gestion de projet

2) Sources

Sources des legendes :

- Figure 2 → [Qu'est-ce qu'un VNC Viewer](#)
- Figure 3 → <https://fr.wikipedia.org/wiki/PuTTY>
- Figure 4 → https://fr.wikipedia.org/wiki/Oracle_VM_VirtualBox
- Figure 5 → [Raspberry Pi 4 : sélection de matériel](#)
- Figure 7 → [https://fr.wikipedia.org/wiki/Discord_\(logiciel\)](https://fr.wikipedia.org/wiki/Discord_(logiciel))
- Figure 8 & 9 → <https://monday.com/>
- Figure 10 → <https://munin-monitoring.org/>
- Figure 11 → <https://www.nginx.com/>
- Figure 12 à 27 → Screen issue de notre travail lors du projet