

# SAÉ - 30.1

*Mettre en oeuvre un système de transmission*



*Kurul - Rabergeau- Kut*

17/10/2023

## Présentation de la SAE

---

Une S.A.É représente une phase de Situation d'Apprentissage et d'Évaluation.

La saé 3.01 "Mettre en œuvre système de transmission" a débuté le 13/10/23 et s'est terminée le 20/10/23.

Le travail englobait l'installation de machines et leur paramétrage, la mise en service d'un serveur Web Nginx et l'application de :

- Transmission vidéo grâce à un serveur vidéo DASH à débit variable ;
- Transmission sécurisée en établissant un serveur VPN ;
- Transmission de médias via la gestion d'un serveur NAS OpenMediaVault.

Nous avons aussi employé des programmes pour examiner les transmissions réseau et avons utilisé ces données pour mieux les appréhender.

Pour ce projet, nous avons été amenés à utiliser :

- Un RaspberryPi avec sa carte micro-SD, son adaptateur d'alimentation ainsi qu'un câble Ethernet pour la connexion via le portail captif.
- Deux PC sous Debian, connectés au réseau par le portail captif, équipés de VirtualBox.

Les buts de cette Saé étaient multiples. Tout d'abord, ils visaient à nous enseigner la mise en place de services au sein du réseau, tout en nous encourageant à réfléchir sur la qualité de la transmission des informations dans le réseau. De plus, ils nous incitent à développer la capacité de faire des interprétations similaires à celles d'un technicien spécialisé dans le domaine des Réseaux et Télécommunications. Un objectif majeur de cette expérience était de renforcer notre autonomie, en nous apprenant à chercher des informations dans la documentation (même si elle n'est pas nécessairement en français) et à collaborer efficacement en groupe, une compétence cruciale pour notre avenir en tant que techniciens.

Dans le cadre de ce rapport, nous vous présenterons le projet que nous avons réalisé et partagerons nos réflexions et interprétations concernant notre travail.

## Sommaire :

1. Notre organisation
2. Config VM / RPI
3. Analyse matériel VM / RPI
4. Systèmes d'analyse de débit: Munin
5. Application à une transmission vidéo : Serveur vidéo à débit adaptatif
6. Application à une transmission multimédia : Serveur NAS
7. Application à une transmission sécurisée : Serveur VPN
8. Partie HTTPS Mise en place d'une sécurisation sur le serveur Vod
9. Conclusion

# Notre organisation

---

Pendant cette Saé, nous avons appliqué les approches agiles que nous avons abordées lors de la ressource Gestion de Projet. Ces approches nous ont permis de démontrer notre agilité, réactivité, transparence et capacité à anticiper. En conséquence, notre équipe est devenue plus unie, capable de travailler en étroite collaboration et de progresser de manière plus fluide dans notre travail.

Les méthodes agiles nous ont donné la possibilité de prendre du recul par rapport à notre travail et d'effectuer des ajustements, notamment en collaborant avec le client, afin d'obtenir des résultats plus précis par rapport à leurs attentes.

Étant donné que notre groupe était récemment formé, il a fallu un certain temps pour s'adapter. La mise en œuvre des méthodes agiles s'est avérée bénéfique pour maintenir une cohésion solide et favoriser une collaboration harmonieuse.

## 1. Méthodologie Agile :

Pour démarrer notre projet, nous avons opté pour la mise en place d'un cadre Scrum. Scrum est une méthodologie agile couramment employée pour la gestion de projets. Elle est structurée autour de cycles de travail appelés "sprints", au cours desquels chaque membre de l'équipe s'engage à réaliser des tâches préalablement définies. À la clôture de chaque sprint, l'équipe procède à une revue de sprint, au cours de laquelle chacun présente ce qu'il a accompli.

## 2. Technique Pomodoro :

Pour optimiser notre productivité et maintenir une concentration constante, nous avons mis en place la méthode Pomodoro. Chaque "pomodoro" consistait en une période de travail ininterrompu de 25 minutes, suivie d'une pause de 5 minutes. Ce schéma nous a grandement aidés à maintenir notre concentration tout en gérant la fatigue.

### 3. Tableau Niko-Niko :

Afin de maintenir un suivi constant du moral de l'équipe et de repérer rapidement tout signe de stress ou de problèmes potentiels, nous avons adopté la méthode du tableau de niko-niko. Chaque jour, chaque membre de l'équipe plaçait un visage souriant, neutre ou triste pour refléter son état émotionnel du jour. Cette approche nous a permis d'établir des canaux de communication ouverts et de traiter les préoccupations avant qu'elles ne prennent de l'ampleur et ne se transforment en problèmes majeurs.

#### Voici comment fonctionne le tableau Niko-Niko :

- Icônes d'émotion : À la fin de chaque journée, chaque membre de l'équipe place une icône sur le tableau pour refléter son humeur de la journée.

Les icônes courantes incluent:

- 😊 (Heureux / Satisfait)
- 😐 (Neutre / OK)
- 😞 (Insatisfait / Contrarié)

Date	Fatih	Nicolas	Suha
13/10	😊	😊	😊
17/10	😞	😊	😊
18/10	😊	😊	😐
19/10	😊	😞	😊
20/10	😐	😊	😞
21/10	😊	😊	😊

## Mise en place du RPI :

Tout d'abord, la première étape consistait à graver la carte micro-SD du Raspberry Pi avec le système d'exploitation Raspberry Pi OS (64 bit). Pour ce faire, nous avons besoin d'un logiciel nommé "win32diskimager" et "RaspberryPi imager"



Il était également important de créer un fichier (sans extension) nommé "SSH" dans la partition boot de la carte micro-SD. Cette action avait pour effet d'activer le protocole SSH, permettant ainsi une configuration à distance. Le SSH permet l'utilisation d'un terminal à partir d'une machine distante de manière sécurisée grâce à la cryptographie asymétrique, qui autorise l'échange de clés pour assurer une communication sécurisée par la suite.

Ensuite, nous avons procédé à la configuration réseau en utilisant un serveur DHCP pour attribuer une adresse IP au Raspberry Pi. Cela nous a permis de configurer l'interface réseau du Raspberry Pi via SSH. Nous avons choisi l'adresse IP 192.168.33.214 (le .14 était déjà pris) et avons ajouté les informations du routeur par défaut ainsi que les serveurs DNS.

Une étape cruciale consistait à activer VNC via raspi-config, car cela était nécessaire pour se connecter au portail captif et installer des paquets à distance avec une interface graphique.

Nous avons également modifié le mot de passe de l'utilisateur "pi" à l'aide de la commande passwd.

Enfin, nous avons suivi la recommandation de modifier le nom de la machine en modifiant le fichier "/etc/hostname" et en nous assurant que la configuration était toujours valide après le redémarrage.

## Configuration de la première machine virtuelle sous VirtualBox :

Il était nécessaire de créer la machine virtuelle en suivant la configuration recommandée :

- 4 processeurs,
- 4 Go de RAM
- Disque dur de 40 Go.

Cette configuration visait à obtenir un bon encodage vidéo (pendant le serveur de lecture adaptative). Mais vu que nous allons créer un autre VM, il ne fallait pas consommer trop de CPU.

La machine virtuelle devait être installée avec le système d'exploitation Ubuntu MATE, qui était une option intéressante car la version MATE était maintenue à jour et enrichie en fonctionnalités tout en conservant une faible consommation de ressources mémoire et CPU.

Par la suite, nous avons entamé la procédure d'attribution d'une adresse IP en effectuant les actions suivantes :

Nous avons commenté le fichier `/etc/network/interfaces` conformément aux directives trouvées sur Internet.

De plus, nous avons ajusté le fichier `/etc/netplan/01-network-manager-all.yaml` en fonction des indications que nous avons trouvées en ligne.

```
GNU nano 6.4
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.33.114/24
      gateway4: 192.168.33.1
      nameservers:
        addresses: [192.168.33.1,8.8.8.8]
```

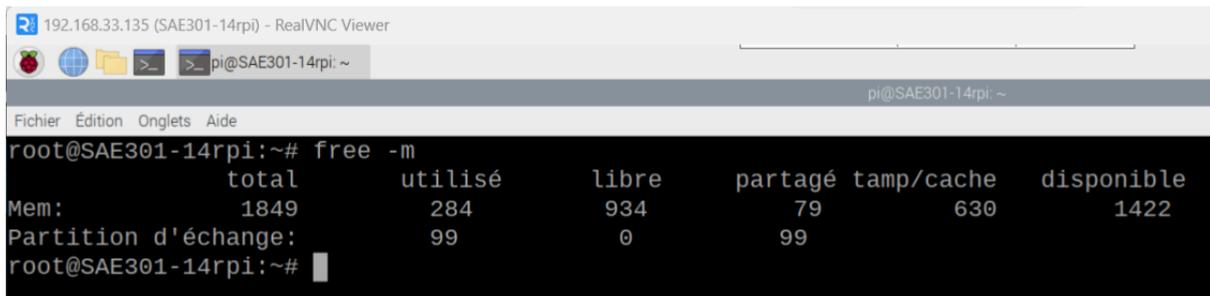
- L'adresse IP de la VM : 192.168.33.114
- La passerelle pour l'interface sur "192.168.33.1" DNS : 192.168.33.1 (serveur portail captif) et "8.8.8.8" (serveur DNS public de Google).

## Analyse du matériel et de ses performances (Raspberry pi) :

Nous avons relevé les composants du Raspberry pi et de la VM afin d'effectuer le jalon 2 :

- **Type des Raspberry Pi:** Raspberry pi 4 Model B (Raspberry pi 2018), le modèle se trouve sur la machine directement
- **Version mémoire RAM:** 1849 Mo de RAM

Avec la commande `free -m` (on utilise `-m` pour avoir l'unité en Mo) :



```

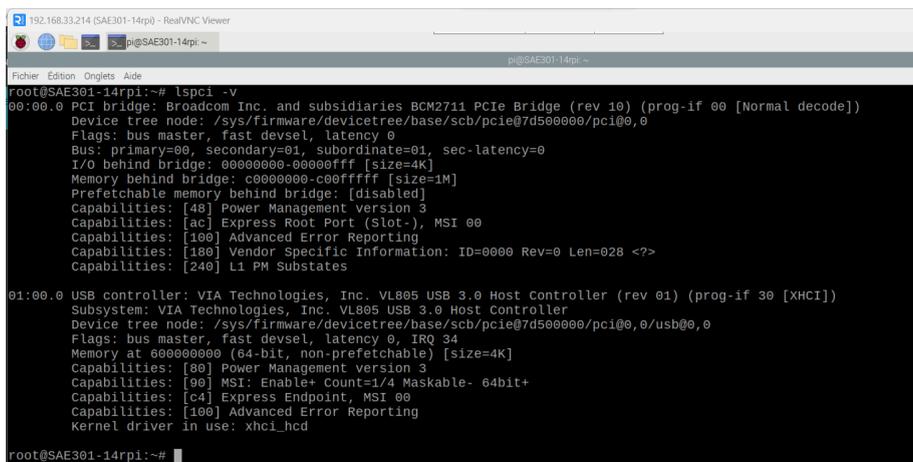
root@SAE301-14rpi:~# free -m
              total        utilisé          libre       partagé  tamp/cache    disponible
Mem:           1849           284           934            79           630           1422
Partition d'échange:           99             0            99
root@SAE301-14rpi:~#

```

- **total :** La quantité totale de mémoire physique (RAM) disponible sur le système.

Nous avons obtenu des informations matérielles détaillées à l'aide de commandes telles que `lshw` pour le Raspberry Pi, `lspci -v` pour les composants réseau, `lsusb -t` pour les ports USB et `lshw -C memory` pour les informations sur la RAM.

- **Carte réseau:** `lspci -v` nous donne un aperçu détaillé des composants matériels connectés via pci-e dans notre ordinateur.



```

root@SAE301-14rpi:~# lspci -v
00:00.0 PCI bridge: Broadcom Inc. and subsidiaries BCM2711 PCIe Bridge (rev 10) (prog-if 00 [Normal decode])
Device tree node: /sys/firmware/devicetree/base/scb/pcie@7d500000/pci@0,0
Flags: bus master, fast devsel, latency 0
Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
I/O behind bridge: 00000000-0000ffff [size=4K]
Memory behind bridge: c0000000-c00fffff [size=1M]
Prefetchable memory behind bridge: [disabled]
Capabilities: [48] Power Management version 3
Capabilities: [ac] Express Root Port (Slot-), MSI 00
Capabilities: [100] Advanced Error Reporting
Capabilities: [180] Vendor Specific Information: ID=0000 Rev=0 Len=028 <?>
Capabilities: [240] L1 PM Substates

01:00.0 USB controller: VIA Technologies, Inc. VL805 USB 3.0 Host Controller (rev 01) (prog-if 30 [XHCI])
Subsystem: VIA Technologies, Inc. VL805 USB 3.0 Host Controller
Device tree node: /sys/firmware/devicetree/base/scb/pcie@7d500000/pci@0,0/usb@0,0
Flags: bus master, fast devsel, latency 0, IRQ 34
Memory at 600000000 (64-bit, non-prefetchable) [size=4K]
Capabilities: [80] Power Management version 3
Capabilities: [90] MSI: Enable+ Count=1/4 Maskable- 64bit+
Capabilities: [c4] Express Endpoint, MSI 00
Capabilities: [100] Advanced Error Reporting
Kernel driver in use: xhci_hcd
root@SAE301-14rpi:~#

```

*lshw -C network* nous donne un aperçu détaillé des composants réseau

```

pi@SAE301-14rpi:~$ sudo lshw -C network
*-network:0
  description: Ethernet interface
  identifiant matériel: 1
  nom logique: eth0
  numéro de série: dc:a6:32:2b:2b:8a
  taille: 100Mbit/s
  capacité: 1Gbit/s
  fonctionnalités: ethernet physical tp mii 10bt 10bt-fd 100bt 100bt-fd 1000bt 1000bt-fd autonegotiation
  configuration: autonegotiation=on broadcast=yes driver=bcmgenet driverversion=6.1.21-v8+ duplex=full ip=192.168.33.214 link=yes m
multicast=yes port=twisted pair speed=100Mbit/s
*-network:1
  description: Interface réseau sans fil
  identifiant matériel: 2
  nom logique: wlan0
  numéro de série: dc:a6:32:2b:2b:8b
  fonctionnalités: ethernet physical wireless
  configuration: broadcast=yes driver=brcmfmac driverversion=7.45.241 firmware=01-703fd60 multicast=yes wireless=IEEE 802.11

```

- **Taille** : 100 Mbit/s
- **Capacité** : Peut fonctionner à 1 Gbit/s, mais est configurée actuellement à 100 Mbit/s.

### USB:

```

pi@SAE301-14rpi:~$ sudo lshw -C bus
*-core
  description: Motherboard
  identifiant matériel: 0
*-usb
  description: USB controller
  produit: VL805 USB 3.0 Host Controller
  fabricant: VIA Technologies, Inc.
  identifiant matériel: 0
  information bus: pci@0000:01:00.0
  version: 01
  bits: 64 bits
  horloge: 33MHz
  fonctionnalités: pm msi pciexpress xhci bus_master cap_list
  configuration: driver=xhci_hcd latency=0
  ressources: irq:34 mémoire:600000000-600000fff
*-usbhost:0
  produit: xHCI Host Controller
  fabricant: Linux 6.1.21-v8+ xhci-hcd
  identifiant matériel: 0
  information bus: usb@1
  nom logique: usb1
  version: 6.01
  fonctionnalités: usb-2.00
  configuration: driver=hub slots=1 speed=480Mbit/s
*-usb
  description: USB hub
  produit: USB2.0 Hub
  fabricant: VIA Labs, Inc.
  identifiant matériel: 1
  information bus: usb@1:1
  version: 4.21
  fonctionnalités: usb-2.10
  configuration: driver=hub maxpower=100mA slots=4 speed=480Mbit/s
*-usbhost:1
  produit: xHCI Host Controller
  fabricant: Linux 6.1.21-v8+ xhci-hcd
  identifiant matériel: 1
  information bus: usb@2
  nom logique: usb2
  version: 6.01
  fonctionnalités: usb-3.00
  configuration: driver=hub slots=4 speed=5000Mbit/s

```

- **Contrôleur USB (USB Controller) :**
  - Produit : VIA Technologies, Inc. VL805 USB 3.0 Host Controller.
  - Version : 01.
  - Horloge : 33 MHz.
  - Ressources : IRQ : 34, plage mémoire : 0x600000000-0x600000fff.
- **Concentrateur USB (USB Hub 1) :**
  - Fonctionnalités : USB 2.0.

- Configuration : Pilote "hub", 1 emplacement, vitesse de 480 Mbit/s (USB 2.0).
- Concentrateur USB (USB Hub 2) :
  - Produit : USB2.6 Hub.
  - Fabricant : VIA Labs, Inc.
  - Version : 4.21.
  - Fonctionnalités : USB 2.0.
  - Configuration : Pilote "hub", puissance maximale de 100 mA, 4 emplacements, vitesse de 480 Mbit/s (USB 2.0).
- Contrôleur hôte xHCI (xHCI Host Controller) :
  - Fonctionnalités : USB 3.0.
  - Configuration : Pilote "hub", 4 emplacements, vitesse de 5000 Mbit/s (USB 3.0).

```

pi@SAE301-14rpi:~ $ lsusb -t
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 5000M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 480M
   |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
      |__ Port 1: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
      |__ Port 1: Dev 3, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
      |__ Port 2: Dev 4, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
pi@SAE301-14rpi:~ $ █

```

Nombre de bus : 2 bus

Le premier bus USB à un débit de 5000 M soit 5 Go, par déduction son type est donc USB3.

Le second bus USB à un débit de 480 M, donc par déduction un type USB2.

## Mémoire:

```

pi@SAE301-14rpi:~ $ sudo lshw -C memory
*-cache
  description: L1 Cache
  identifiant matériel: 0
  taille: 32KiB
*-cache
  description: L1 Cache
  identifiant matériel: 0
  taille: 32KiB
*-cache
  description: L1 Cache
  identifiant matériel: 0
  taille: 32KiB
*-cache
  description: L1 Cache
  identifiant matériel: 0
  taille: 32KiB
*-memory
  description: Mémoire système
  identifiant matériel: 6
  taille: 1849MiB
pi@SAE301-14rpi:~ $ █

```

La commande `sudo lshw -C memory` affiche des informations sur la mémoire de notre Raspberry Pi tels que:

- **Cache L1** : Il y a quatre caches L1, chacune d'une taille de 32 KiB.
- **Mémoire système** : La mémoire système totale est de 1849 MiB (comme vu précédemment avec la commande `free -m`).

### Processeur:

```

pi@SAE301-14rpi:~ $ cat /proc/cpuinfo
processor       : 0
BogoMIPS      : 108.00
Features       : fp asimd evtstrm crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant    : 0x0
CPU part      : 0xd08
CPU revision   : 3

processor       : 1
BogoMIPS      : 108.00
Features       : fp asimd evtstrm crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant    : 0x0
CPU part      : 0xd08
CPU revision   : 3

processor       : 2
BogoMIPS      : 108.00
Features       : fp asimd evtstrm crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant    : 0x0
CPU part      : 0xd08
CPU revision   : 3

processor       : 3
BogoMIPS      : 108.00
Features       : fp asimd evtstrm crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant    : 0x0
CPU part      : 0xd08
CPU revision   : 3

Hardware       : BCM2835
Revision      : b03111
Serial        : 100000006090029a
Model         : Raspberry Pi 4 Model B Rev 1.1
pi@SAE301-14rpi:~ $

```

```

pi@SAE301-14rpi:~ $ sudo lshw -C cpu
*-cpu:0
  description: CPU
  produit: cpu
  identifiant matériel: 1
  information bus: cpu@0
  taille: 1500MHz
  capacité: 1500MHz
  fonctionnalités: fp asimd evtstrm crc32 cpuid cpufreq
*-cpu:1
  description: CPU
  produit: cpu
  identifiant matériel: 2
  information bus: cpu@1
  taille: 1500MHz
  capacité: 1500MHz
  fonctionnalités: fp asimd evtstrm crc32 cpuid cpufreq
*-cpu:2
  description: CPU
  produit: cpu
  identifiant matériel: 3
  information bus: cpu@2
  taille: 1500MHz
  capacité: 1500MHz
  fonctionnalités: fp asimd evtstrm crc32 cpuid cpufreq
*-cpu:3
  description: CPU
  produit: cpu
  identifiant matériel: 4
  information bus: cpu@3
  taille: 1500MHz
  capacité: 1500MHz
  fonctionnalités: fp asimd evtstrm crc32 cpuid cpufreq
*-cpu:4 DÉSACTIVÉ
  description: CPU
  produit: l2-cache0
  identifiant matériel: 5
  information bus: cpu@4
pi@SAE301-14rpi:~ $

```

La commande `sudo lshw -C cpu` affiche des informations sur les unités de traitement central (CPU) de notre Raspberry Pi.

La commande `cat /proc/cpuinfo`, nous obtenir des informations supplémentaires sur le CPU (Central Processing Unit = Processeur) de notre Raspberry Pi. tel que le nom du modèle :ARMv7 Processor rev 3 (v711)

Nous disposons de quatre cœurs de processeur (CPU) sur notre système, tous fonctionnant à une fréquence de 1500 MHz. Ces CPU prennent en charge des instructions telles que:

- **fp** (point flottant),
- **asimd** (Advanced SIMD),
- **evtstrm** (événements temporisés),
- **crc32** (calcul de somme de contrôle CRC32),
- **cpuid** (instruction d'identification du CPU)
- **cpufreq** (gestion de la fréquence CPU).

Le cinquième CPU (CPU 4 est désactivé ou non présent dans notre configuration.)

Afin de finaliser le Jalon 2, nous avons dû répondre à quelques questions :

### ***1. Comment une carte réseau Ethernet détermine son mode de fonctionnement ?***

Une carte réseau Ethernet détermine son mode de fonctionnement (Ethernet, Fast Ethernet ou Gigabit Ethernet) en utilisant un processus appelé "négociation automatique". Ce processus permet à deux dispositifs connectés sur un réseau de choisir le mode de transmission le plus rapide qu'ils supportent tous les deux.

Lors de la négociation automatique, les dispositifs échangent des impulsions électriques pour indiquer les modes qu'ils supportent. Une fois que les deux dispositifs ont identifié un mode commun le plus rapide, ils se configurent pour utiliser ce mode.

### ***2. Quels sont les débits dans les 3 cas ?***

Ethernet ou 10BASE-T, offre un débit de 10 Mbps ou 1,25 Mo/s

FastEthernet ou 100BASE-TX, offre un débit de 100 Mbps ou 12,5 Mo/s

GigabitEthernet ou 1000BASE-T, offre un débit de 1 Gbps ou 125 Mo/s

### 3. Quelle est la vitesse actuelle du port ethernet en utilisant l'outil ethtool

```

pi@SAE301-14rpi:~ $ sudo ethtool eth0
Settings for eth0:
  Supported ports: [ TP      MII ]
  Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full

  Supported pause frame use: Symmetric Receive-only
  Supports auto-negotiation: Yes
  Supported FEC modes: Not reported
  Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full
  Advertised pause frame use: Symmetric Receive-only
  Advertised auto-negotiation: Yes
  Advertised FEC modes: Not reported
  Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                       100baseT/Half 100baseT/Full
  Link partner advertised pause frame use: No
  Link partner advertised auto-negotiation: Yes
  Link partner advertised FEC modes: Not reported
  Speed: 100Mb/s
  Duplex: Full
  Auto-negotiation: on
  master-slave cfg: preferred slave
  master-slave status: slave
  Port: Twisted Pair
  PHYAD: 1
  Transceiver: external
  MDI-X: Unknown
  Supports Wake-on: d
  Wake-on: d
  Current message level: 0x00000007 (7)
                               drv probe link

  Link detected: yes
pi@SAE301-14rpi:~ $ █

```

- *Speed (Vitesse)* : La vitesse actuelle de l'interface est de 100 Mb/s.
- *Duplex* : L'interface est configurée en mode Full Duplex (duplex complet).

## Analyse carte SD performances:

Nous avons installé et utilisé "agnostics", mais nous avons commis une erreur. Par conséquent, nous avons utilisé l'interface graphique en passant par "Menu RPI > Accessoires > Diagnostic".

Nous avons attendu un moment car cela s'est bloqué, puis nous avons cliqué sur "Annuler" => "Afficher les journaux" => "Vitesse", ce qui nous a donné le résultat suivant :

```

rpdiaigs.txt x
1  Raspberry Pi Diagnostics - version (unknown)
2  Mon Oct 16 12:35:24 2023
3
4  Test : SD Card Speed Test
5  Run 1
6  prepare-file;0;0;21059;41
7  seq-write;0;0;21650;42
8  rand-4k-write;0;0;1105;276
9  rand-4k-read;5561;1390;0;0
10 Sequential write speed 21650 KB/sec (target 10000) - PASS
11 Random write speed 276 IOPS (target 500) - FAIL
12 Random read speed 1390 IOPS (target 1500) - FAIL
13 Run 2
14 prepare-file;0;0;24326;47
15 seq-write;0;0;24183;47
16 rand-4k-write;0;0;968;242
17 rand-4k-read;5672;1418;0;0
18 Sequential write speed 24183 KB/sec (target 10000) - PASS
19 Random write speed 242 IOPS (target 500) - FAIL
20 Random read speed 1418 IOPS (target 1500) - FAIL
21 Run 3
22 prepare-file;0;0;24353;47
23 seq-write;0;0;21005;41
24 rand-4k-write;0;0;1019;254
25 rand-4k-read;5592;1398;0;0
26 Sequential write speed 21005 KB/sec (target 10000) - PASS
27 Random write speed 254 IOPS (target 500) - FAIL
28 Random read speed 1398 IOPS (target 1500) - FAIL
29 Test FAIL
30
31

```

La carte SD a une bonne vitesse d'écriture séquentielle, mais une vitesse d'écriture et de lecture aléatoires inférieure à la cible.

Les résultats des commandes **sudo hdparm -Tt /dev/mmcblk0** et **dd if=/dev/zero of=/tmp/sample.bin bs=8k count=128k** fournissent des informations sur les performances de notre carte SD tels que:

- Vitesse lecture:

```
pi@SAE301-14rpi:~ $ sudo hdparm -Tt /dev/mmcblk0
/dev/mmcblk0:
Timing cached reads: 1898 MB in 2.00 seconds = 950.98 MB/sec
Timing buffered disk reads: 132 MB in 3.04 seconds = 43.40 MB/sec
pi@SAE301-14rpi:~ $
```

Lecture mise en cache (Cached Reads) : 950.98 MB/s

Cela représente la vitesse à laquelle les données sont lues à partir de la mémoire cache.

Lecture tamponnée depuis le disque (Buffered Disk Reads) : 43.40 MB/s

Cela indique la vitesse de lecture depuis le disque.

- Vitesse d'écriture:

```
pi@SAE301-14rpi:~ $ dd if=/dev/zero of=/tmp/sample.bin bs=8k count=128k
131072+0 enregistrements lus
131072+0 enregistrements écrits
1073741824 octets (1,1 GB, 1,0 GiB) copiés, 44,488 s, 24,1 MB/s
```

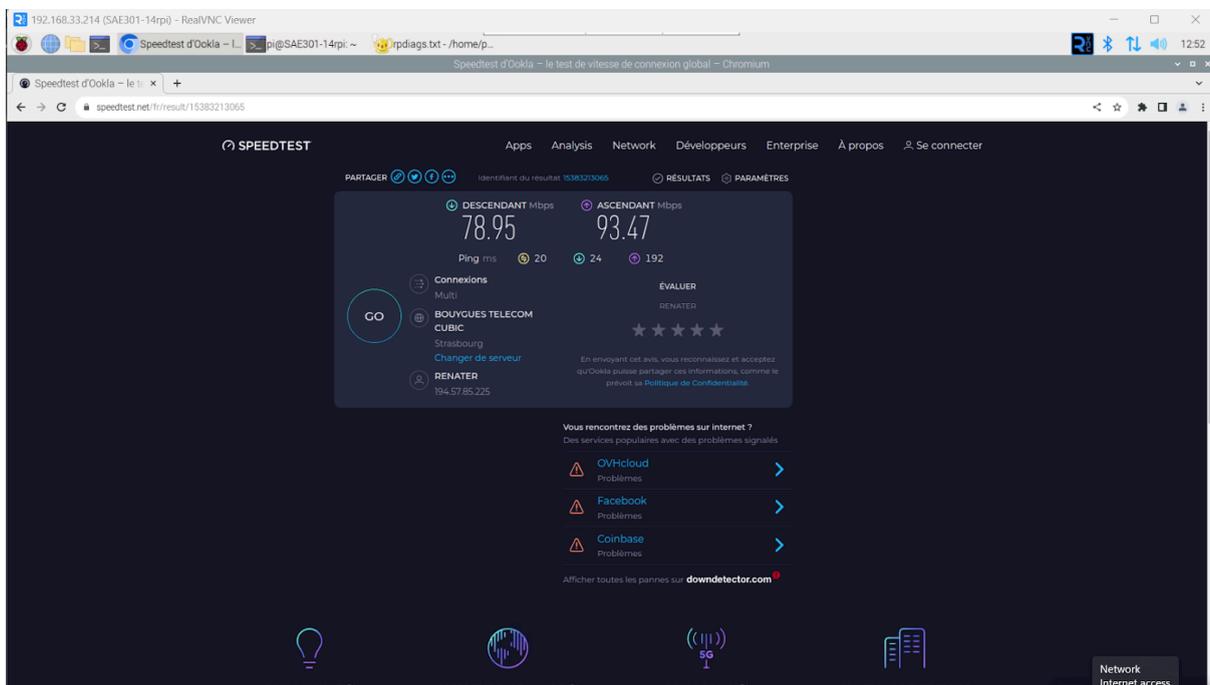
**dd if=/dev/zero of=/tmp/sample.bin bs=8k count=128k** : Cette commande effectue un test d'écriture en écrivant des données dans un fichier. La vitesse d'écriture est de 24.1 MB/s.

	Ecriture	Lecture
Carte micro SD	24,1Mb/s	950,98Mb/s
Disque dur HDD 7200 tours	42,1Mo/s	43,7Mo/s
Disque dur SSD sata	600Mo/s	600Mo/s
Disque dur pcie/nvme	8500Mo/s	10500Mo/s
Disque dur M2 sata	5000Mo/s	5500Mo/s
Clé USB3 servant de boot	5 Gbits/s	5 Gbits/s

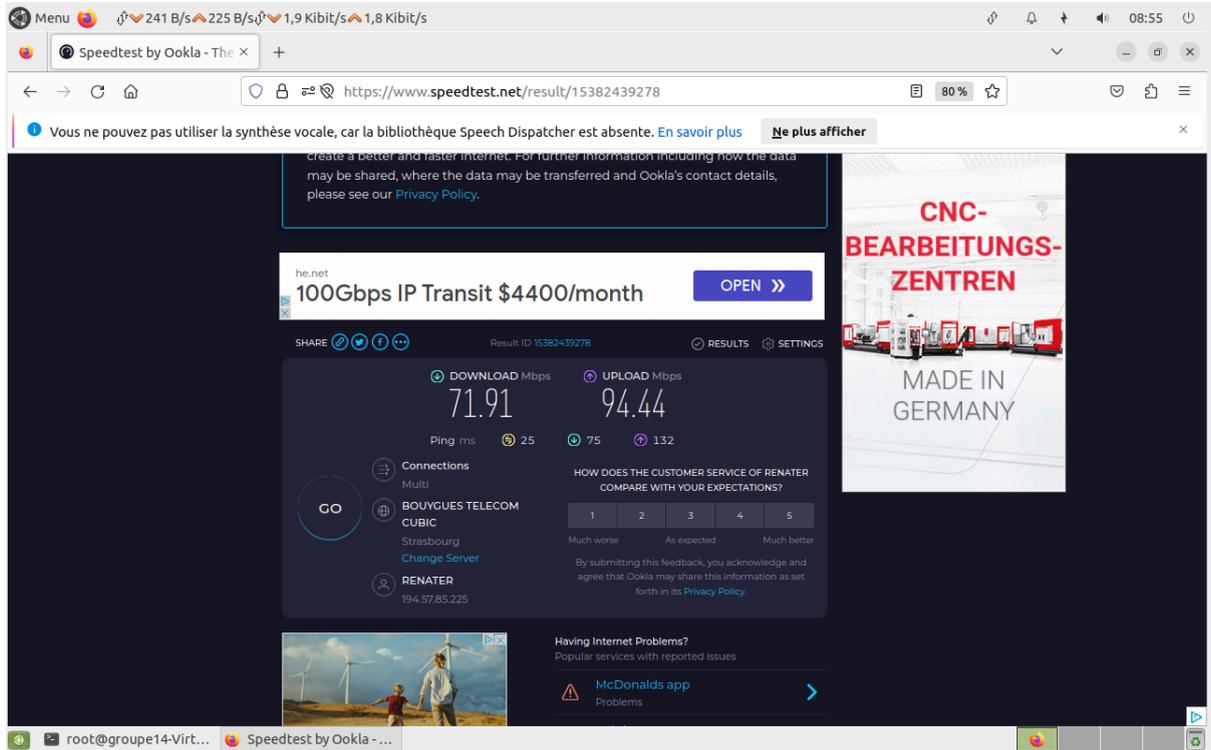
Speedtest :

Un speed test est un outil en ligne qui mesure la vitesse de votre connexion internet. Il évalue la vitesse de téléchargement (download), la vitesse d'envoi (upload) et la latence. Cela permet de savoir si votre connexion est rapide et stable, et de vérifier si vous obtenez la vitesse promise par votre fournisseur d'accès à internet.

depuis le RPI 4 :



## Speedtest depuis le VM :



## Analyse du matériel (VM) :

<b>Général</b> Nom : SAE_GRP14 Système d'exploitation : Ubuntu (64-bit)			
<b>System</b> Mémoire vive : 4096 Mo Processeurs : 4 Ordre d'amorçage : Disquette, Optique, Disque dur, Réseau Accélération : VT-x/AMD-V, Pagination imbriquée, PAE/NX, Paravirtualisation KVM			
<b>Affichage</b> Mémoire vidéo : 16 Mo Contrôleur graphique : VMSVGA Accélération : 3D Serveur de bureau à distance : Désactivé Enregistrement : Désactivé			
<b>Stockage</b> Contrôleur : IDE Maître secondaire IDE : [Lecteur optique] Vide Contrôleur : SATA Port SATA 0 : SAE_GRP14.vdi (Normal, 40,00 Gio)			
<b>Audio</b> Pilote hôte : PulseAudio Contrôleur : ICH AC97			
<b>Réseau</b> Interface 1: Intel PRO/1000 MT Desktop (Interface pont eth0)			
<b>USB</b> Contrôleur USB : OHCI Filtres de périphérique : 0 (0 actif)			
<b>Dossiers partagés</b> Aucun			
<b>Description</b> Aucune			

## Général:

- Nom du système : SAE\_GRP14
- Système d'exploitation : Ubuntu (64 bits)
- Mémoire vive : 4096 Mo (4 Go)
- Processeurs : 4
- Accélération : VT-x/AMD-V, Pagination imbriquée, PAE/NX, Paravirtualisation KVM

## Stockage:

- Contrôleur : SATA
- Port SATA 0 : SAE\_GRP14.vdi (40,00 Gio)

## Performances de la machine virtuelle Virtualbox :

*iperf3 est un outil utilisé pour mesurer la bande passante réseau. Les options -u et -R ont des significations spécifiques dans iperf3:*

-u : Cette option indique à iperf3 d'utiliser le protocole UDP plutôt que le protocole TCP par défaut. L'utilisation du protocole UDP peut être utile pour tester la perte de paquets sur un réseau.

-R : Cette option signifie "reverse". Lorsqu'elle est utilisée, la direction du test est inversée. Par défaut, le client iperf3 envoie des données au serveur, mais avec l'option -R, le serveur enverra des données au client. Cela peut être utile pour tester la bande passante dans la direction opposée ou pour contourner certains types de restrictions ou de configurations réseau.

depuis le Client VM:

- **iperf3 -c 192.168.33.214**

```
groupe_14@groupe14-VirtualBox:~$ iperf3 -c 192.168.33.214
Connecting to host 192.168.33.214, port 5201
[ 5] local 192.168.33.114 port 53382 connected to 192.168.33.214 port 5201
[ ID] Interval            Transfer          Bitrate          Retr  Cwnd
[ 5]  0.00-1.00    sec   14.3 MBytes    120 Mb/s         0    601 KBytes
[ 5]  1.00-2.00    sec   11.2 MBytes    94.3 Mb/s         0    1.15 MBytes
[ 5]  2.00-3.00    sec   11.2 MBytes    94.4 Mb/s         50   1.04 MBytes
[ 5]  3.00-4.00    sec   11.2 MBytes    94.4 Mb/s         0    1.16 MBytes
[ 5]  4.00-5.00    sec   11.2 MBytes    94.3 Mb/s         0    1.25 MBytes
[ 5]  5.00-6.00    sec   11.2 MBytes    94.4 Mb/s         0    1.33 MBytes
[ 5]  6.00-7.00    sec   11.2 MBytes    94.4 Mb/s         0    1.38 MBytes
[ 5]  7.00-8.00    sec   11.2 MBytes    94.3 Mb/s         5    1.02 MBytes
[ 5]  8.00-9.00    sec   11.2 MBytes    94.4 Mb/s         0    1.08 MBytes
[ 5]  9.00-10.00   sec   11.2 MBytes    94.4 Mb/s         0    1.12 MBytes
-----
[ ID] Interval            Transfer          Bitrate          Retr
[ 5]  0.00-10.00   sec   116 MBytes    96.9 Mb/s         55
[ 5]  0.00-10.14   sec   113 MBytes    93.6 Mb/s
sender
receiver
iperf Done.
```

- **iperf3 -c 192.168.33.214 -R**

```
groupe_14@groupe14-VirtualBox:~$ iperf3 -c 192.168.33.214 -R
Connecting to host 192.168.33.214, port 5201
Reverse mode, remote host 192.168.33.214 is sending
[ 5] local 192.168.33.114 port 53776 connected to 192.168.33.214 port 5201
[ ID] Interval            Transfer          Bitrate
[ 5]  0.00-1.00    sec   11.3 MBytes    94.6 Mb/s
[ 5]  1.00-2.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  2.00-3.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  3.00-4.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  4.00-5.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  5.00-6.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  6.00-7.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  7.00-8.00    sec   11.2 MBytes    94.1 Mb/s
[ 5]  8.00-9.00    sec   11.2 MBytes    93.6 Mb/s
[ 5]  9.00-10.00   sec   11.2 MBytes    93.6 Mb/s
-----
[ ID] Interval            Transfer          Bitrate          Retr
[ 5]  0.00-10.04   sec   113 MBytes    94.2 Mb/s         0
[ 5]  0.00-10.00   sec   112 MBytes    94.1 Mb/s
sender
receiver
iperf Done.
```

- iperf3 -c 192.168.33.214 -u

```
groupe_14@groupe14-VirtualBox:~$ iperf3 -c 192.168.33.214 -u
Connecting to host 192.168.33.214, port 5201
[ 5] local 192.168.33.114 port 48506 connected to 192.168.33.214 port 5201
[ ID] Interval           Transfer     Bitrate      Total Datagrams
[ 5]  0.00-1.00   sec      129 KBytes   1.05 Mbits/sec  91
[ 5]  1.00-2.00   sec      127 KBytes   1.04 Mbits/sec  90
[ 5]  2.00-3.02   sec      126 KBytes   1.01 Mbits/sec  89
[ 5]  3.02-4.01   sec      129 KBytes   1.07 Mbits/sec  91
[ 5]  4.01-5.04   sec      127 KBytes   1.02 Mbits/sec  90
[ 5]  5.04-6.00   sec      132 KBytes   1.12 Mbits/sec  93
[ 5]  6.00-7.00   sec      127 KBytes   1.04 Mbits/sec  90
[ 5]  7.00-8.01   sec      126 KBytes   1.02 Mbits/sec  89
[ 5]  8.01-9.13   sec      130 KBytes    954 Kbits/sec  92
[ 5]  9.13-10.00  sec      129 KBytes   1.21 Mbits/sec  91
-----
[ ID] Interval           Transfer     Bitrate      Jitter        Lost/Total Datagrams
[ 5]  0.00-10.00  sec      1.25 MBytes   1.05 Mbits/sec  0.000 ms    0/906 (0%)  sender
[ 5]  0.00-10.05  sec      1.25 MBytes   1.04 Mbits/sec  0.129 ms    0/906 (0%)  receiver

iperf Done.
```

- iperf3 -c 192.168.33.214 -u -R

```
groupe_14@groupe14-VirtualBox:~$ iperf3 -c 192.168.33.214 -u -R
Connecting to host 192.168.33.214, port 5201
Reverse mode, remote host 192.168.33.214 is sending
[ 5] local 192.168.33.114 port 37079 connected to 192.168.33.214 port 5201
[ ID] Interval           Transfer     Bitrate      Jitter        Lost/Total Datagrams
[ 5]  0.00-1.00   sec      134 KBytes   1.10 Mbits/sec  0.086 ms    0/95 (0%)
[ 5]  1.00-2.00   sec      129 KBytes   1.06 Mbits/sec  0.073 ms    0/91 (0%)
[ 5]  2.00-3.00   sec      127 KBytes   1.04 Mbits/sec  0.095 ms    0/90 (0%)
[ 5]  3.00-4.00   sec      129 KBytes   1.06 Mbits/sec  0.082 ms    0/91 (0%)
[ 5]  4.00-5.00   sec      127 KBytes   1.04 Mbits/sec  0.061 ms    0/90 (0%)
[ 5]  5.00-6.00   sec      129 KBytes   1.05 Mbits/sec  0.061 ms    0/91 (0%)
[ 5]  6.00-7.00   sec      127 KBytes   1.04 Mbits/sec  0.051 ms    0/90 (0%)
[ 5]  7.00-8.00   sec      129 KBytes   1.05 Mbits/sec  0.053 ms    0/91 (0%)
[ 5]  8.00-9.00   sec      127 KBytes   1.04 Mbits/sec  0.089 ms    0/90 (0%)
[ 5]  9.00-10.00  sec      129 KBytes   1.06 Mbits/sec  0.073 ms    0/91 (0%)
-----
[ ID] Interval           Transfer     Bitrate      Jitter        Lost/Total Datagrams
[ 5]  0.00-10.05  sec      1.26 MBytes   1.05 Mbits/sec  0.000 ms    0/910 (0%)  sender
[ 5]  0.00-10.00  sec      1.26 MBytes   1.05 Mbits/sec  0.073 ms    0/910 (0%)  receiver

iperf Done.
```

## depuis le Serveur RPI

- iperf3 -c 192.168.33.214

```
pi@SAE301-14rpi:~ $ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 192.168.33.114, port 53378
[ 5] local 192.168.33.214 port 5201 connected to 192.168.33.114 port 53382
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-1.00    sec    10.8 MBytes    90.4 Mbits/sec
[ 5]  1.00-2.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  2.00-3.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  3.00-4.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  4.00-5.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  5.00-6.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  6.00-7.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  7.00-8.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  8.00-9.00    sec    11.2 MBytes    93.9 Mbits/sec
[ 5]  9.00-10.00   sec    11.2 MBytes    93.9 Mbits/sec
[ 5] 10.00-10.14   sec     1.52 MBytes    94.0 Mbits/sec
-----
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-10.14   sec    113 MBytes    93.6 Mbits/sec
-----
receiver
```

- iperf3 -c 192.168.33.214 -R

```
-----
Server listening on 5201
-----
Accepted connection from 192.168.33.114, port 53764
[ 5] local 192.168.33.214 port 5201 connected to 192.168.33.114 port 53776
[ ID] Interval            Transfer           Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec    11.3 MBytes    94.9 Mbits/sec     0   106 KBytes
[ 5]  1.00-2.00    sec    11.2 MBytes    93.8 Mbits/sec     0   115 KBytes
[ 5]  2.00-3.00    sec    11.2 MBytes    94.4 Mbits/sec     0   115 KBytes
[ 5]  3.00-4.00    sec    11.2 MBytes    93.8 Mbits/sec     0   115 KBytes
[ 5]  4.00-5.00    sec    11.4 MBytes    95.9 Mbits/sec     0   122 KBytes
[ 5]  5.00-6.00    sec    11.2 MBytes    93.8 Mbits/sec     0   122 KBytes
[ 5]  6.00-7.00    sec    11.2 MBytes    94.4 Mbits/sec     0   122 KBytes
[ 5]  7.00-8.00    sec    11.2 MBytes    94.4 Mbits/sec     0   122 KBytes
[ 5]  8.00-9.00    sec    11.0 MBytes    92.3 Mbits/sec     0   122 KBytes
[ 5]  9.00-10.00   sec    11.2 MBytes    93.8 Mbits/sec     0   122 KBytes
[ 5] 10.00-10.04   sec     573 KBytes     111 Mbits/sec     0   122 KBytes
-----
[ ID] Interval            Transfer           Bitrate      Retr
[ 5]  0.00-10.04   sec    113 MBytes    94.2 Mbits/sec     0
-----
sender
```

- iperf3 -c 192.168.33.214 -u

```
-----  
Server listening on 5201  
-----  
Accepted connection from 192.168.33.114, port 53822  
[ 5] local 192.168.33.214 port 5201 connected to 192.168.33.114 port 48506  
[ ID] Interval      Transfer      Bitrate      Jitter      Lost/Total Datagrams  
[ 5] 0.00-1.00    sec  56.6 KBytes  463 Kbits/sec  0.056 ms  0/40 (0%)  
[ 5] 1.00-2.00    sec  188 KBytes  1.54 Mbites/sec  0.081 ms  0/133 (0%)  
[ 5] 2.00-3.00    sec  129 KBytes  1.05 Mbites/sec  0.203 ms  0/91 (0%)  
[ 5] 3.00-4.00    sec  116 KBytes  950 Kbits/sec  0.098 ms  0/82 (0%)  
[ 5] 4.00-5.00    sec  139 KBytes  1.14 Mbites/sec  0.135 ms  0/98 (0%)  
[ 5] 5.00-6.00    sec  134 KBytes  1.10 Mbites/sec  0.025 ms  0/95 (0%)  
[ 5] 6.00-7.00    sec  129 KBytes  1.05 Mbites/sec  0.039 ms  0/91 (0%)  
[ 5] 7.00-8.00    sec  122 KBytes  996 Kbits/sec  0.100 ms  0/86 (0%)  
[ 5] 8.00-9.00    sec  129 KBytes  1.05 Mbites/sec  0.165 ms  0/91 (0%)  
[ 5] 9.00-10.00   sec  127 KBytes  1.04 Mbites/sec  0.152 ms  0/90 (0%)  
[ 5] 10.00-10.05  sec  12.7 KBytes  2.30 Mbites/sec  0.129 ms  0/9 (0%)  
-----  
[ ID] Interval      Transfer      Bitrate      Jitter      Lost/Total Datagrams  
[ 5] 0.00-10.05   sec  1.25 MBytes  1.04 Mbites/sec  0.129 ms  0/906 (0%) receiver
```

- iperf3 -c 192.168.33.214 -u -R

```
-----  
Server listening on 5201  
-----  
Accepted connection from 192.168.33.114, port 41570  
[ 5] local 192.168.33.214 port 5201 connected to 192.168.33.114 port 37079  
[ ID] Interval      Transfer      Bitrate      Total Datagrams  
[ 5] 0.00-1.00    sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 1.00-2.00    sec  127 KBytes  1.04 Mbites/sec  90  
[ 5] 2.00-3.00    sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 3.00-4.00    sec  127 KBytes  1.04 Mbites/sec  90  
[ 5] 4.00-5.00    sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 5.00-6.00    sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 6.00-7.00    sec  127 KBytes  1.04 Mbites/sec  90  
[ 5] 7.00-8.00    sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 8.00-9.00    sec  127 KBytes  1.04 Mbites/sec  90  
[ 5] 9.00-10.00   sec  129 KBytes  1.05 Mbites/sec  91  
[ 5] 10.00-10.05  sec  5.66 KBytes  1.03 Mbites/sec  4  
-----  
[ ID] Interval      Transfer      Bitrate      Jitter      Lost/Total Datagrams  
[ 5] 0.00-10.05   sec  1.26 MBytes  1.05 Mbites/sec  0.000 ms  0/910 (0%) sender  
-----  
Server listening on 5201  
-----
```

```

root@groupe14-VirtualBox:/home/groupe_14# traceroute paris.testdebit.info
traceroute to paris.testdebit.info (89.84.1.194), 30 hops max, 60 byte packets
 1 crim-portail.portail (192.168.33.1) 1.769 ms 1.700 ms 1.657 ms
 2 gwv5src.pu-pm.univ-fcomte.fr (192.168.100.253) 1.548 ms 1.454 ms 1.369 ms
 3 cerbere1.pu-pm.univ-fcomte.fr (194.57.85.2) 1.284 ms 1.159 ms 1.040 ms
 4 * * *
 5 172.22.0.42 (172.22.0.42) 2.547 ms 2.455 ms 2.298 ms
 6 172.22.0.22 (172.22.0.22) 5.010 ms 4.216 ms 5.722 ms
 7 * * *
 8 172.20.240.206 (172.20.240.206) 3.595 ms 2.946 ms 3.192 ms
 9 * * *
10 194.57.79.198 (194.57.79.198) 4.738 ms 4.545 ms 5.113 ms
11 rr-sequane-ren-nr-besancon-rtr-091.noc.renater.fr (193.55.202.81) 4.803 ms 4.687 ms 4.655 ms
12 vlan1934-be2-ren-nr-besancon-rtr-091.noc.renater.fr (193.55.202.80) 3.757 ms 3.369 ms 3.339 ms
13 te0-0-0-8-ren-nr-dijon-rtr-091.noc.renater.fr (193.51.177.215) 13.785 ms 13.582 ms 13.553 ms
14 xe-0-1-0-ren-nr-lyon1-rtr-131.noc.renater.fr (193.55.204.1) 13.356 ms 13.394 ms te0-3-2-0-lyon1-rtr-001.noc.renater.fr (193.51.177.72)
    13.216 ms
15 et-3-1-7-ren-nr-paris1-rtr-131.noc.renater.fr (193.51.180.166) 12.897 ms 13.367 ms 12.935 ms
16 et-3-1-2-ren-nr-paris2-rtr-131.noc.renater.fr (193.55.204.195) 13.381 ms 13.890 ms et-2-0-0-ren-nr-paris2-rtr-131.noc.renater.fr (193.
    55.204.193) 12.701 ms
17 equinix-paris.as5410.net (195.42.144.217) 12.460 ms 16.064 ms 14.606 ms
18 62.34.2.87 (62.34.2.87) 15.890 ms 15.789 ms 15.569 ms
19 212.194.171.76 (212.194.171.76) 15.429 ms 15.389 ms 15.314 ms
20 * * *
21 89.89.101.141 (89.89.101.141) 13.858 ms 13.987 ms 13.940 ms
22 89.84.1.194 (89.84.1.194) 13.452 ms 13.405 ms 13.777 ms

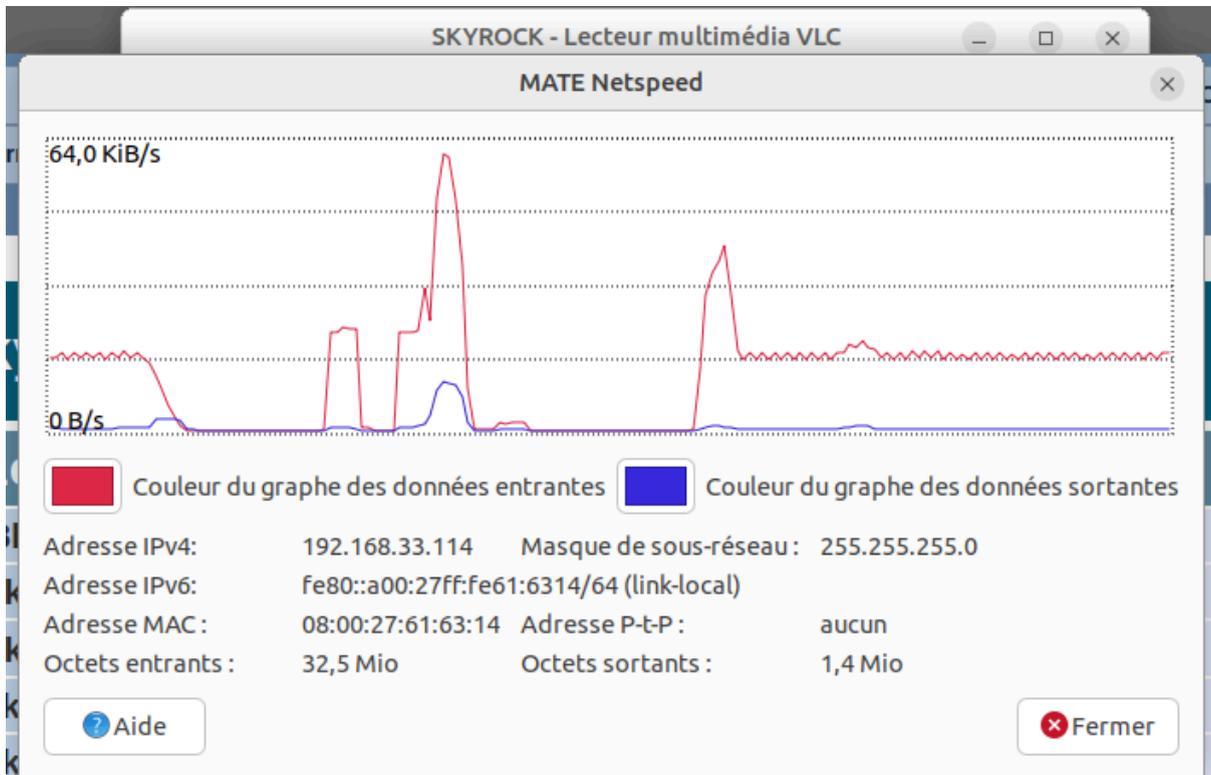
```

Ces lignes représentent la sortie de la commande "traceroute" vers le serveur externe choisi: **paris.testdebit.info** sur <https://iperf.fr/iperf-servers.php>

### Audio :

Pour faire ces mesures, nous avons choisi la radio Skyrock

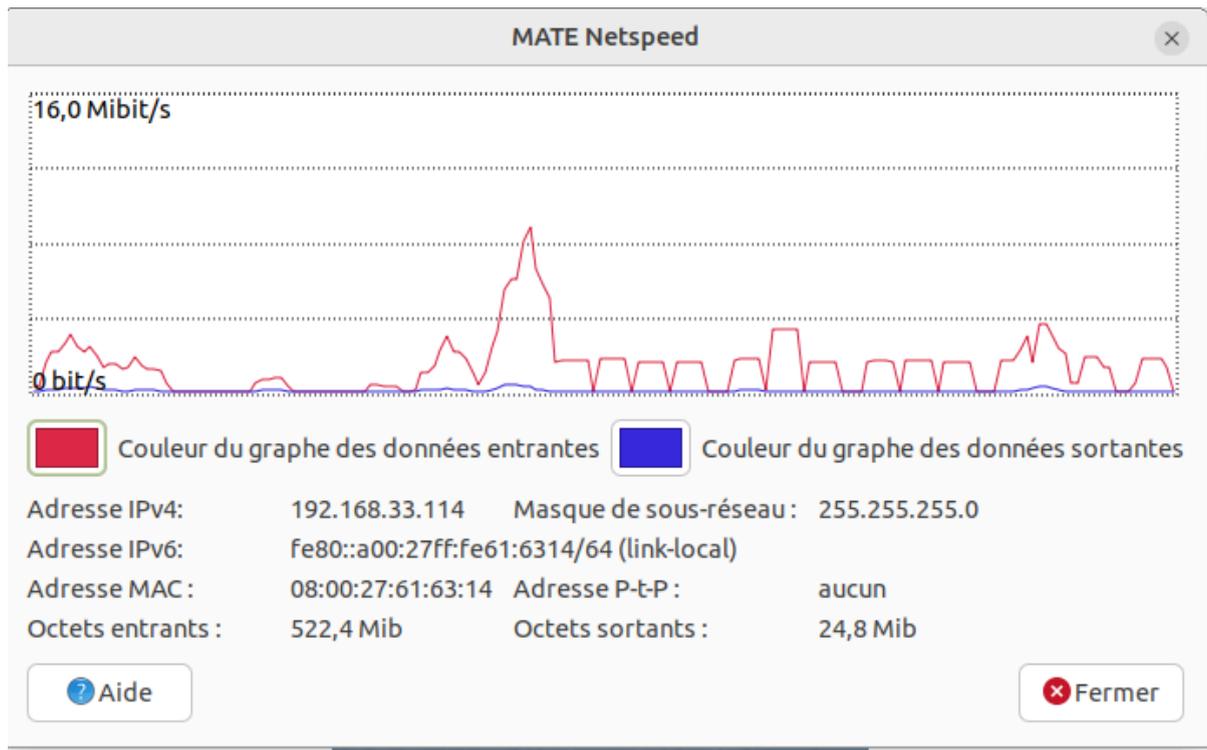
- Capture netspeed lors de l'utilisation de la Radio



### Vidéo :

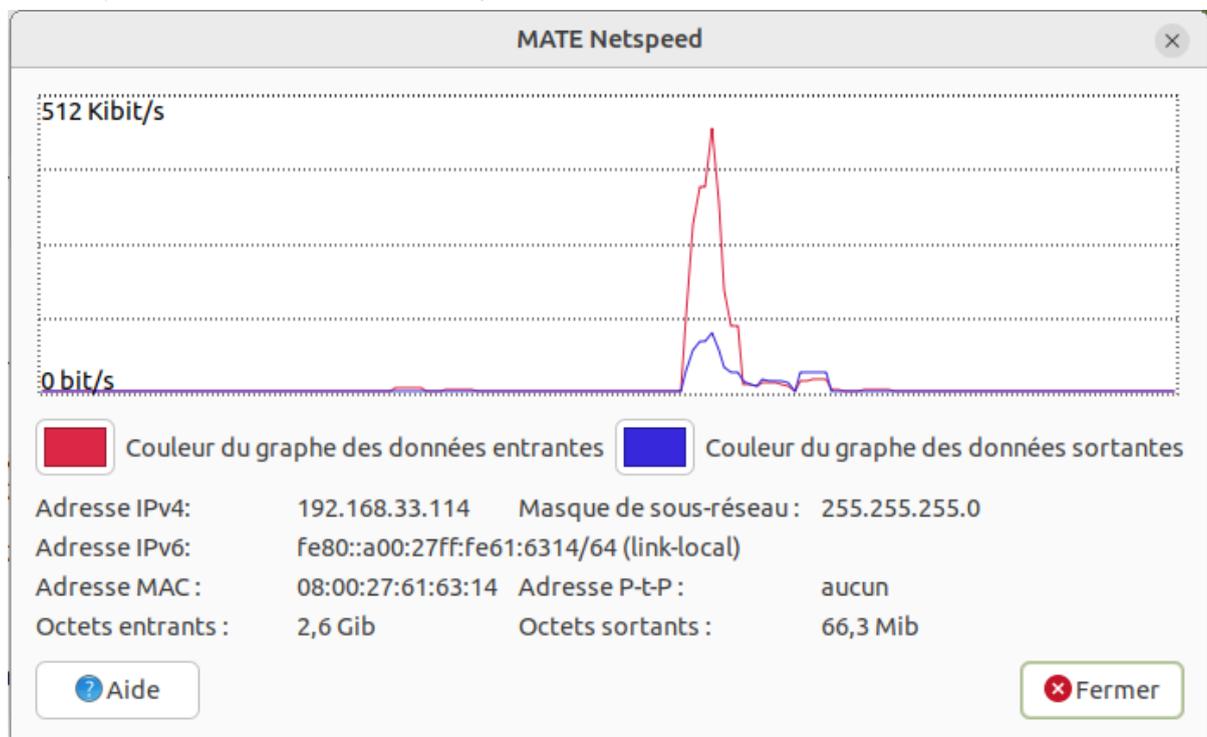
Afin de réaliser ces mesures, nous avons utilisé la TV en direct de BFMTV qui possède une bonne qualité vidéo.

- Capture Netspeed de la vidéo en HD :

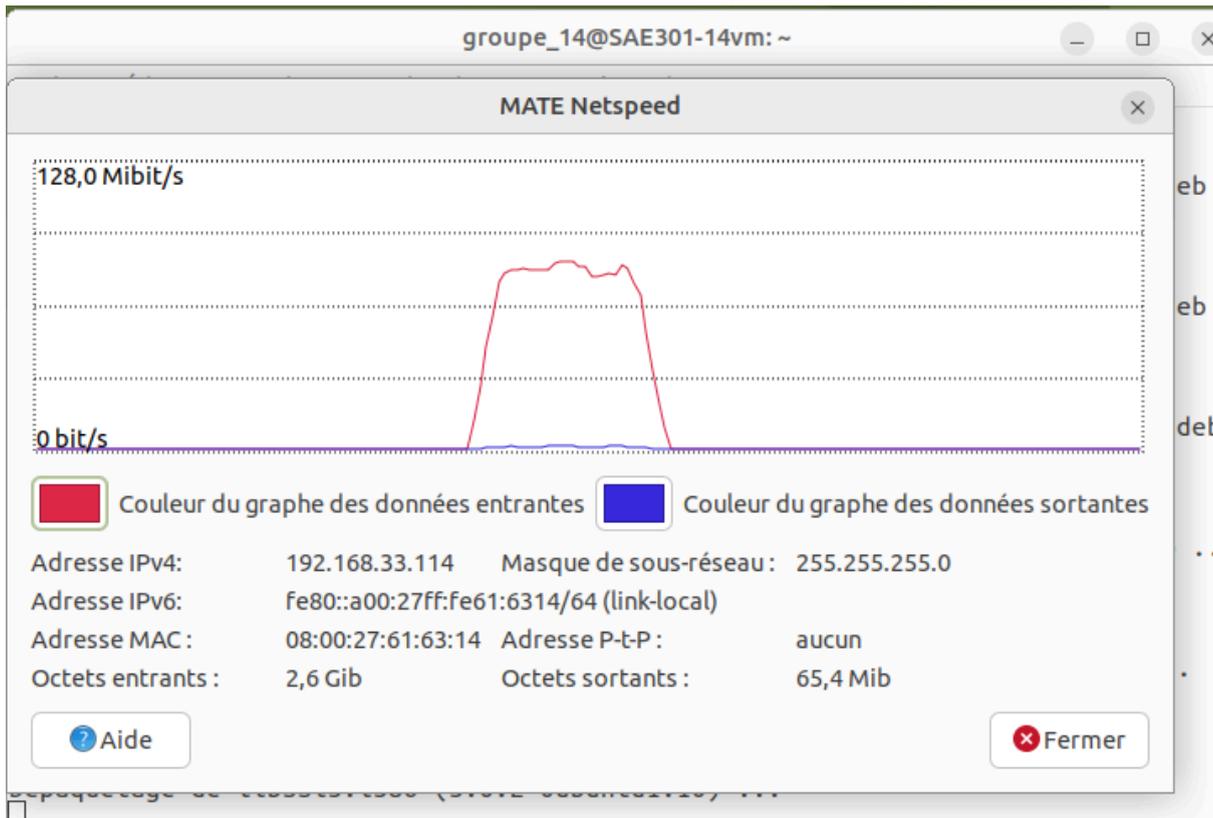


### Données :

L'analyse Netspeed d'un ping :



Lors de la mise à jour, le graphique obtenu sur Netspeed est :

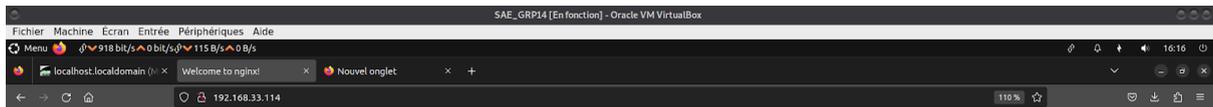


# Partie 3

## Systèmes d'analyse de débit

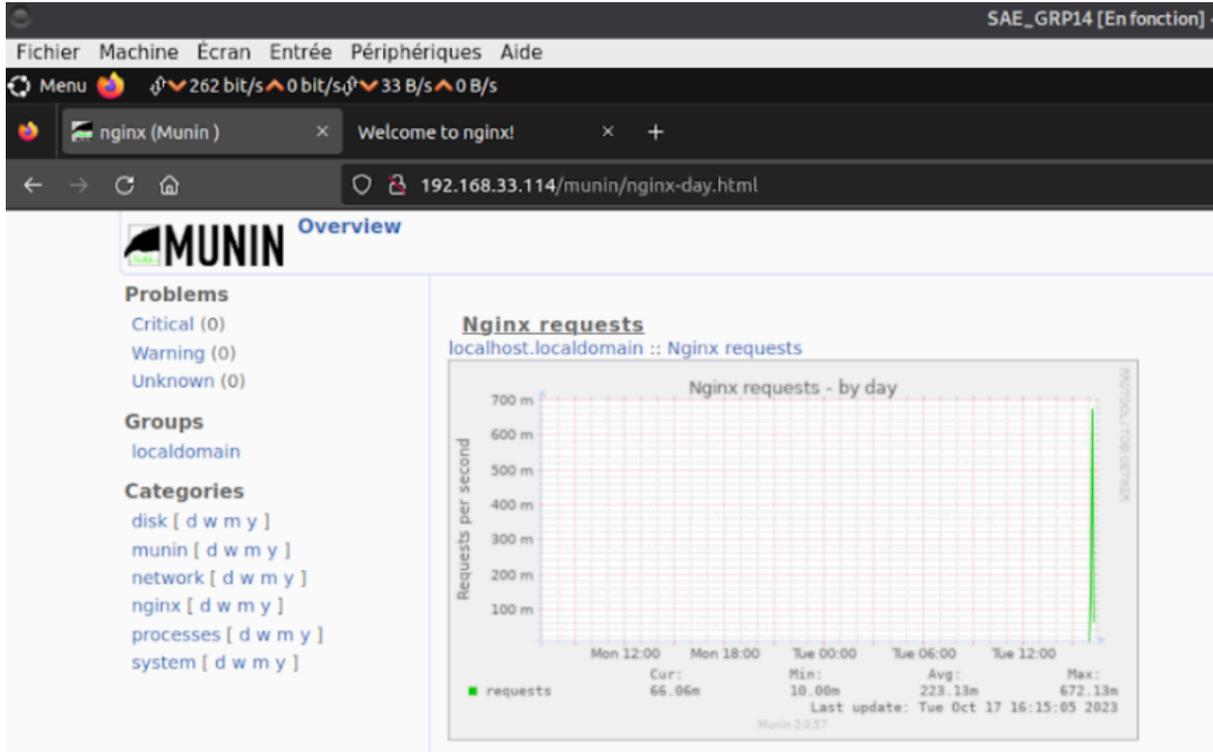
### *Munin*

Dans cette partie, nous avons réalisé l'installation d'un serveur web Nginx. L'installation se fait par l'exécution de commande d'installation de paquet **nginx-full**. Nous avons dû configurer ce serveur afin de préparer l'installation de Munin. Les paquets à télécharger étaient **munin** , **munin-node** et **munin-plugins-extra**.



Nous avons réalisé la configuration du serveur nginx pour la préparation de munin avec la modification du fichier `/etc/nginx/sites-enabled/default`.

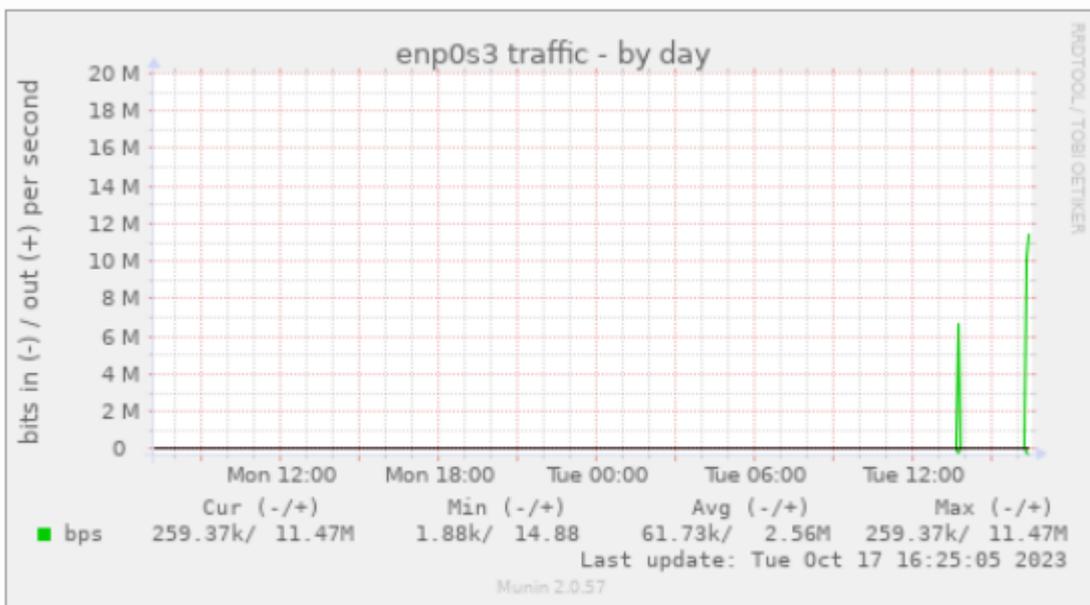
Par la suite, nous pouvons accéder à la page web de munin:



Lorsque l'on crée 50 connexions avec la commande :

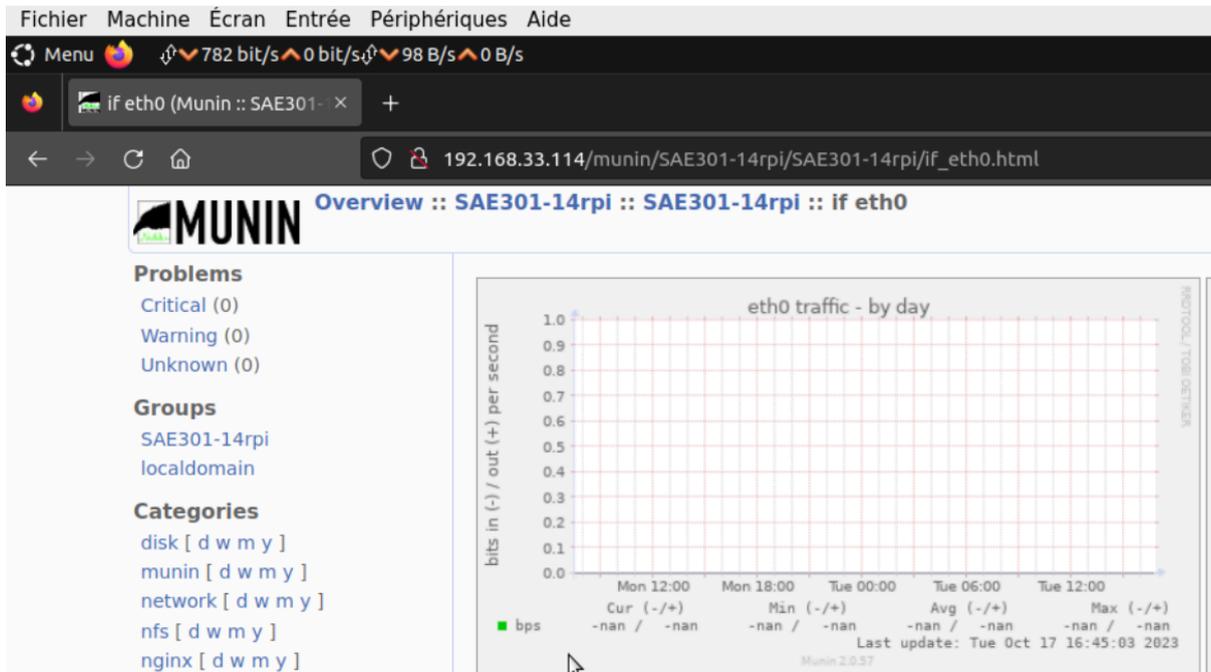
```
root@SAE301-14vm:/etc/munin# curl -s http://192.168.33.114?[1-50] http://192.168.33.114/Test.php?[1-50]
```

Les graphiques évoluent



Nous avons configuré le RPI et la VM pour que l'on ait accès aux mesures sur RPI dans le serveur de la VM

- Page d'accueil du munin où l'on voit le noeud RPi :

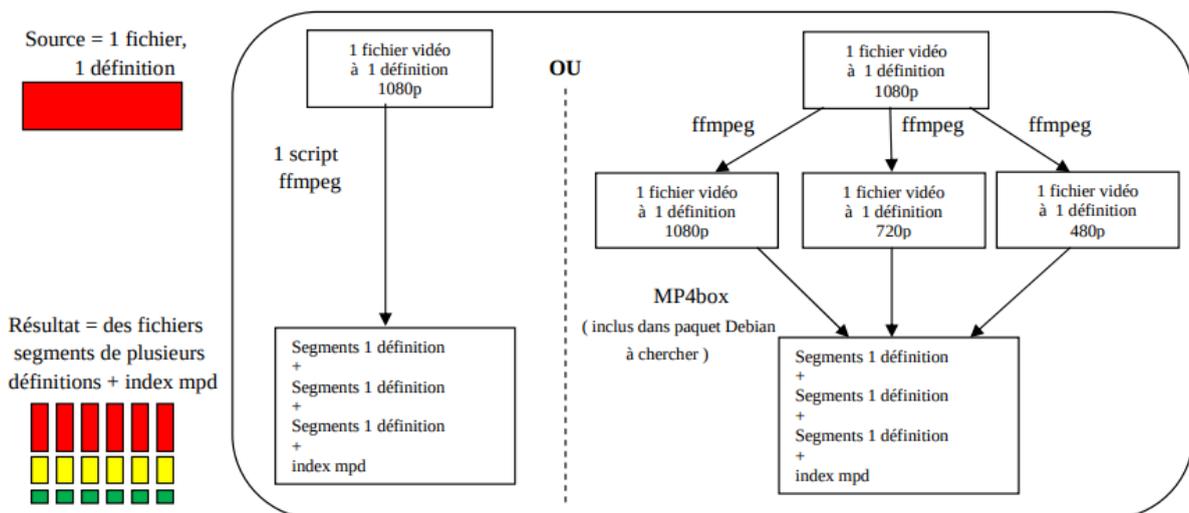


# Partie 4

## Application à une transmission vidéo : Serveur vidéo à débit adaptatif

### Préparation des fichiers vidéo :

DASH est une norme de vidéo à débit adaptatif, les segments m4s sont donc sous la norme DASH. Afin de mettre la vidéo sur le serveur en lecture adaptative, il nous faut mettre la vidéo dans le format mpd. Ce format crée donc des segments m4s et un fichier mpd qui contient les informations de fichier vidéo selon la qualité et l'ordre de lecture.



### Segmentation de vidéo en 1080p

```
root@SAE301-14vm:/etc/nginx# ffmpeg -y -i /home/groupe_14/Téléchargements/X2Download.app-Muslim\ Style\ T
echno\ Dance-\(1080p\).mp4 -c:v libx264 -r 24 -x264opts 'keyint=48:min-keyint=48:no-scenecut' -vf scale=
2:1080 -b:v 8000k -maxrate 8000k -movflags faststart -bufsize 8600k -profile:v main -preset fast -an "out
put2_1080.mp4"
```

### Segmentation de vidéo en 720p

```
root@SAE301-14vm:/etc/nginx# ffmpeg -y -i /home/groupe_14/Téléchargements/X2Download.app-Muslim\ Style\ T
echno\ Dance-\(1080p\).mp4 -c:v libx264 -r 24 -x264opts 'keyint=48:min-keyint=48:no-scenecut' -vf scale=
2:720 -b:v 2M -maxrate 2M -movflags faststart -bufsize 2M -profile:v main -preset fast -an "output2_720.m
p4"
```

### Segmentation de vidéo en 480p

```
root@SAE301-14vm:/etc/nginx# ffmpeg -y -i /home/groupe_14/Téléchargements/X2Download.app-Muslim\ Style\ T
echno\ Dance-\(1080p\).mp4 -c:v libx264 -r 24 -x264opts 'keyint=48:min-keyint=48:no-scenecut' -vf scale=
2:480 -b:v 500k -maxrate 500k -movflags faststart -bufsize 500k -profile:v main -preset fast -an "output2
_480.mp4"
```

Commande MP4Box qui permet de rassembler nos trois vidéos de qualités différentes. Le "package" permet de mettre sous la norme DASH les segments créés :

```
root@SAE301-14vm:/etc/nginx# MP4Box -dash 4000 -rap -frag 4000 -rap -segment-name 'segment_' -fps 24 output2_1080.mp4#video:id=1080p output2_720.mp4#video:id=720p output2_480.mp4#video:id=480p -out /var/www/html/dash/dash3/output2.mpd
```

### Vidéo sans aucune limitation de la bande passante :

Etat	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	Temps
200	GET	192.168.33.114	/	document	Html	434 o	245 o	0 ms
200	GET	192.168.33.114	index.js	script	js	453 o	194 o	0 ms
200	GET	192.168.33.114	shaka-player.compiled.min.js	script	js	134,96 Ko	411,86 Ko	0 ms
200	GET	192.168.33.114	output2.mpd	shaka-player.compo...	octet-st...	1,83 Ko	3,50 Ko	0 ms
200	GET	192.168.33.114	favicon.ico	favicon.ico	image/x-ico...	318 o	162 o	0 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	1,24 Ko	989 o	0 ms
200	GET	192.168.33.114	segment_r2_1.m4s	shaka-player.compo...	octet-st...	205,77 Ko	205,44 Ko	0 ms
200	GET	192.168.33.114	segment_r1_2.m4s	shaka-player.compo...	octet-st...	959,21 Ko	958,89 Ko	1 ms
200	GET	192.168.33.114	segment_r1_3.m4s	shaka-player.compo...	octet-st...	906,07 Ko	905,75 Ko	1 ms
200	GET	192.168.33.114	segment_r1_4.m4s	shaka-player.compo...	octet-st...	965,55 Ko	965,23 Ko	1 ms
200	GET	192.168.33.114	segment_4.m4s	shaka-player.compo...	octet-st...	4,04 Mo	4,04 Mo	30 ms
200	GET	192.168.33.114	segment_5.m4s	shaka-player.compo...	octet-st...	3,83 Mo	3,83 Mo	36 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	1,24 Ko	989 o	0 ms
200	GET	192.168.33.114	segment_r14.m4s	shaka-player.compo...	octet-st...	3,81 Mo	3,81 Mo	38 ms
200	GET	192.168.33.114	segment_r15.m4s	shaka-player.compo...	octet-st...	3,37 Mo	3,37 Mo	42 ms
200	GET	192.168.33.114	segment_r16.m4s	shaka-player.compo...	octet-st...	4,10 Mo	4,10 Mo	102 ms
200	GET	192.168.33.114	segment_r17.m4s	shaka-player.compo...	octet-st...	3,59 Mo	3,59 Mo	84 ms

### Vidéo en "Good 3G" :

Etat	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	Temps
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	989 o	989 o	1079 ms
200	GET	192.168.33.114	segment_r2_13.m4s	shaka-player.compo...	octet-st...	258,11 Ko	258,11 Ko	1304 ms
200	GET	192.168.33.114	segment_r2_14.m4s	shaka-player.compo...	octet-st...	229,67 Ko	229,67 Ko	1091 ms
200	GET	192.168.33.114	segment_r2_15.m4s	shaka-player.compo...	octet-st...	212,40 Ko	212,40 Ko	1087 ms
200	GET	192.168.33.114	segment_r2_16.m4s	shaka-player.compo...	octet-st...	253,17 Ko	253,17 Ko	1088 ms
200	GET	192.168.33.114	segment_r2_17.m4s	shaka-player.compo...	octet-st...	252,59 Ko	252,59 Ko	1084 ms
200	GET	192.168.33.114	segment_r2_18.m4s	shaka-player.compo...	octet-st...	255,64 Ko	255,64 Ko	1084 ms
200	GET	192.168.33.114	segment_r2_19.m4s	shaka-player.compo...	octet-st...	225,74 Ko	225,74 Ko	1082 ms
200	GET	192.168.33.114	segment_r2_20.m4s	shaka-player.compo...	octet-st...	269,35 Ko	269,35 Ko	1084 ms
200	GET	192.168.33.114	segment_r2_21.m4s	shaka-player.compo...	octet-st...	226,51 Ko	226,51 Ko	1084 ms
200	GET	192.168.33.114	segment_r2_22.m4s	shaka-player.compo...	octet-st...	226,92 Ko	226,92 Ko	1086 ms
200	GET	192.168.33.114	segment_r2_23.m4s	shaka-player.compo...	octet-st...	219,73 Ko	219,73 Ko	1086 ms
200	GET	192.168.33.114	segment_r2_24.m4s	shaka-player.compo...	octet-st...	232,86 Ko	232,86 Ko	1086 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	989 o	989 o	108 ms
200	GET	192.168.33.114	segment_r2_24.m4s	shaka-player.compo...	octet-st...	232,86 Ko	232,86 Ko	1089 ms
200	GET	192.168.33.114	segment_r2_25.m4s	shaka-player.compo...	octet-st...	244,51 Ko	244,51 Ko	1079 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	989 o	989 o	108 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	989 o	989 o	108 ms
200	GET	192.168.33.114	segment_init.mpd	shaka-player.compo...	mp4	989 o	989 o	112 ms
200	GET	192.168.33.114	segment_r2_20.m4s	shaka-player.compo...	octet-st...	269,35 Ko	269,35 Ko	1087 ms
200	GET	192.168.33.114	segment_r2_21.m4s	shaka-player.compo...	octet-st...	226,51 Ko	226,51 Ko	1087 ms
200	GET	192.168.33.114	segment_r2_22.m4s	shaka-player.compo...	octet-st...	226,92 Ko	226,92 Ko	1089 ms
200	GET	192.168.33.114	segment_r2_23.m4s	shaka-player.compo...	octet-st...	219,73 Ko	219,73 Ko	1088 ms
200	GET	192.168.33.114	segment_r2_24.m4s	shaka-player.compo...	octet-st...	232,86 Ko	232,86 Ko	1086 ms

Lors de la lecture de la vidéo à débit adaptatif, nous pouvons voir le débit vidéo selon le type de connexion et se rendre compte que lors du changement de sans limite vers la 3G, le débit vidéo diminue et que la latence de chargement de vidéo augmente.

# Partie 5

## Application à une transmission multimédia :

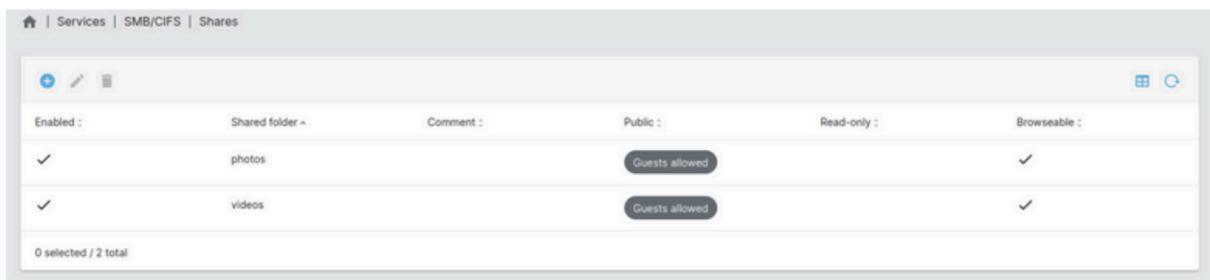
### *Serveur NAS*

Un serveur **NAS** est un **serveur de stockage** permettant à tous les utilisateurs du réseau de déposer et de récupérer des fichiers de tout type sur celui-ci. Dans notre cas, ce serveur était une machine virtuelle sur laquelle nous avons installé l'ISO Openmediavault (OMV). Celle-ci était hébergée sur nos PC personnels.

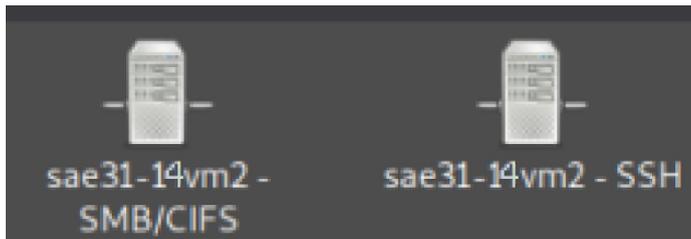
Nous avons modifié le nom d'hôte, créé un disque dur en format ext4 que nous avons monté. Ensuite, nous avons configuré un RAID linéaire en utilisant les deux disques durs de 20 Go. Nous avons activé le service SMB (Server Message Block) et créé deux répertoires nommés "photos" et "vidéos".

Nous n'avons pas réussi à mettre en place le FTP en raison de problèmes de permissions.

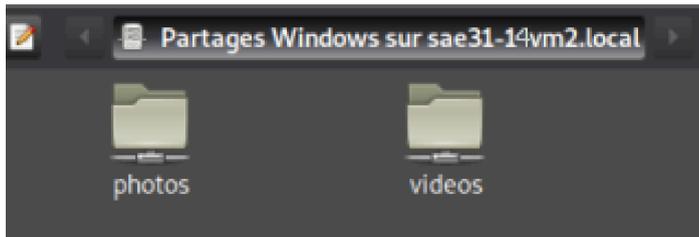
```
Device      Boot      Start          End      Sectors  Size Id Type
/dev/sda1   *          2048 18970623 18968576    9G 83 Linux
/dev/sda2             18972670 20969471 1996802    975M  5 Extended
/dev/sda5             18972672 20969471 1996800    975M 82 Linux swap / Solaris
```



Nous avons observé les différents serveurs sur le réseau public



Et lorsque nous cliquons sur nos serveur, nous pouvons voir les dossiers partagés que nous avons créés via l'interface graphique de OMW :



# Partie 6

## Application à une transmission sécurisée :

### Serveur VPN

#### Partie Wireguard :

On a commencé l'installation des paquets requis pour faire fonctionner Wire Guard, à la fois sur le serveur et le client. Ensuite, nous avons créé une paire de clés, comprenant une clé publique et une clé privée, en utilisant une série de commandes reliées entre elles grâce à l'opérateur "|". Cette génération de clés a été effectuée depuis notre ordinateur personnel via une connexion SSH vers le serveur.

```
interface: wg0
  public key: IcwrB0Ls70zG7Fpz0yE0pr8n6J0BYeXytcN/Y3sXoFk=
  private key: (hidden)

peer: SSQ0dH5RbVhyj4mjx0S057cSyxh9cX6eVbM0wts5YAc=
```

Afin de confirmer la configuration correcte de notre VPN, nous avons effectué une capture Wireshark sur l'interface enp0s3 du serveur tout en lançant simultanément une commande ping depuis le client vers l'adresse IP associée au VPN (wg0) du serveur. Cela nous permet d'observer :

23	4.434597598	192.168.33.114	192.168.33.214	TCP	66 S
24	4.438657995	192.168.33.214	192.168.33.114	SSH	150 S
25	4.438658276	192.168.33.214	192.168.33.114	WireGu...	170 T

```

Protocol: UDP (17)
Header Checksum: 0xa56a [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.33.214
Destination Address: 192.168.33.114
- User Datagram Protocol, Src Port: 51820, Dst Port: 51820
  Source Port: 51820
  Destination Port: 51820
  Length: 136
  Checksum: 0x4124 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 10]
  [Timestamps]
  UDP payload (128 bytes)
- WireGuard Protocol
  Type: Transport Data (4)
  Reserved: 000000
  
```

Dans ce paquet de type "wireguard", l'intégralité des informations n'est pas visible, cependant, plus bas, il était indiqué "paquet chiffré", ce qui indiquait que la configuration du VPN était correcte.

# Partie HTTPS

## Mise en place d'une sécurisation sur le serveur Vod

Par la suite, nous avons intégré une nouvelle section qui concerne notre page HTTPS. Contrairement à la partie HTTP de la même page, cette section inclut la configuration pour écouter en SSL sur le port 443 (port HTTPS), ainsi que l'intégration du certificat SSL et de la clé SSL que nous avons créés précédemment dans cette section.

```
root@SAE301-14vm:/etc/nginx/sites-enabled# cat vod-site
server {
    listen 443 ssl;
    server_name 192.168.33.114;

    ssl_certificate /var/www/html/dash/server.crt;
    ssl_certificate_key /var/www/html/dash/server.key;

    root /var/www/html/dash/;
    index index.html;

    location / {
        auth_basic "Zone sécurisée";
        auth_basic_user_file /etc/nginx/.htpasswd;

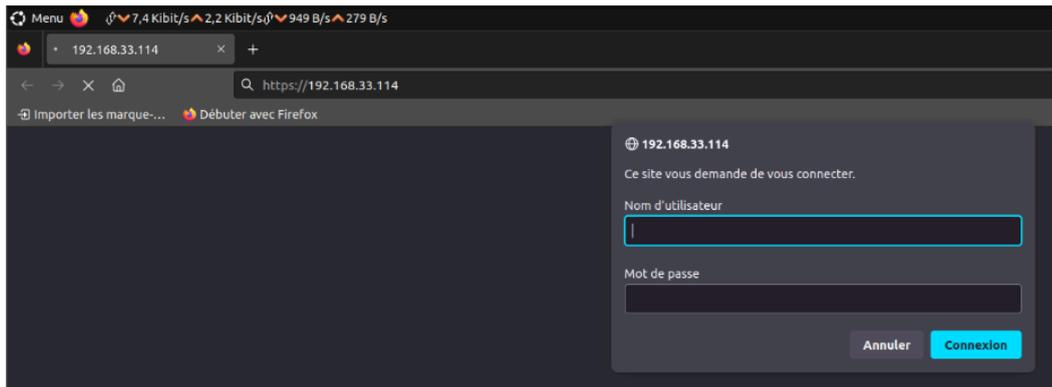
        try_files $uri $uri/ =404;
    }

#   location / {
#       try_files $uri $uri/ =404;
#   }

    location ~ /\.mpd$ {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Cache-Control' 'no-cache';
    }

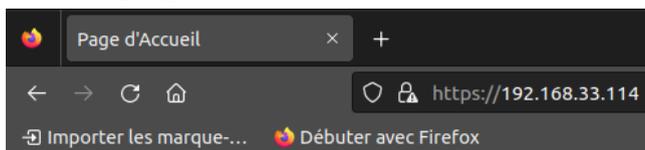
    location ~ /\.m4s$ {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Cache-Control' 'no-cache';
    }
}
```

Nous avons mis en place à un authentification avec un mots de passe sur notre site grace au fichier dans `/etc/nginx/.htpasswd`



Lorsque nous nous connectons avec l'authentification, nous pouvons accéder à nos deux images représentatives, qui mènent à nos deux vidéos que nous avons configurées pour pointer vers nos fichiers .mpd

Voici la page d'accueil de notre site



## **Bienvenue sur notre site de videos**

### **Video 1**



### **Video 2**



Voici les fichiers de nos serveurs web

```
root@SAE301-14vm:/var/www/html/dash# ls  
dash2 dash3 image2.png image.png index.html server.crt server.key video1.html video2.html
```

Nous pouvons voir clairement les deux dossiers "dash" qui contiennent les segments avec les deux fichiers .mpd. Nous apercevons également le fichier "index.html" qui contient la page d'accueil, ainsi que les fichiers "server.crt" et "server.key" qui renferment la clé de sécurité de nos serveurs. Enfin, nous trouvons les deux images représentatives et les deux pages HTML qui contiennent les vidéos.

## Conclusion :

Pendant cette SAE, nous avons mis en pratique nos compétences en réseau pour déployer des services, tout en utilisant nos compétences en télécommunications pour analyser les données relatives au débit. La maîtrise de l'anglais s'est révélée cruciale pour consulter des documentations techniques, tandis que la gestion de projet a été essentielle, car nous avons travaillé de manière autonome et en équipe. Cette expérience a renforcé notre esprit d'équipe, notre sens de l'organisation et notre rigueur.

Cependant, nous avons également été confrontés à divers problèmes, tels que des Raspberry Pi qui ne fonctionnaient pas, des tentatives d'accès non autorisées via SSH sur nos ordinateurs et des difficultés d'affichage des graphiques sur Munin, entre autres.

La résolution de ces problèmes a exigé de la collaboration et une grande détermination, car certains de ces obstacles ont pu nous faire perdre un temps précieux. Finalement, grâce à l'entraide et à notre persévérance, nous avons réussi à surmonter ces difficultés.