TP – Mini-projet Supervision

1 Consignes

Durant ces TP, vous travaillerez de préférence en binôme. Il n'y a pas de contrôle de TP pour ce module, les manipulations et le compte-rendu seront notés au fur et à mesure.

Des séances encadrées et des séances en autonomie sont programmées sur l'emploi du temps, vous devez traiter le sujet dans l'ordre.

1.1 Compte-rendu et dépôt Github

Dès la première séance de TP, vous devez créer un dépôt Github privé et envoyer le lien à l'enseignant. Votre compte-rendu sera constitué de plusieurs fichiers au format Markdown et sera stocké sur ce dépôt. Vous devez avoir un fichier Markdown par partie. Le fichier README.md sera le point d'entrée de votre dépôt, il contiendra un lien vers chaque autre fichiers Markdown créer. Il décrira toutes les ressources disponibles dans le dépôt. Le dépôt contiendra également tous les fichiers importants que vous aurez créés et/ou modifiés, avec par exemple des fichier de configurations des routeurs, des fichiers de scripts, des fichiers Docker Image, des fichiers Docker Compose, des fichiers de configuration de service…

Les fichiers de configuration des routeurs ainsi que le script bash devront obligatoirement être versionnés. Par exemple pour les fichiers de configuration des routeurs on doit pouvoir retrouver facilement les différentes parties du TP (choisissez des messages de commit appropriés). Lors des validations et corrections je ne consulterai que les fichiers présents dans la branche **main**.

La convention de nommage pour le dépôt est la suivante : *identifiant_année-num_module-nom1-nom2*. Par exemple pour les TP du module ETRS813_TRI de l'année 2023 fait par le binôme composé de Thierry DUPONT et Alexandre MARTIN, le dépôt doit s'appeler **23-813-DUPONT-MARTIN**.

Vous aurez le droit d'apporter des modifications à votre dépôt au plus tard, une semaine après la dernière séance de TP ou de projet du module.

Il est fortement déconseillé de mettre des captures d'écran au format image dans votre compte-rendu. Bien évidemment, il est autorisé de mettre des résultats de commandes, mais le format texte doit être privilégié. Votre compte-rendu devra être à la fois **précis et concis**. Il devra pouvoir être lu et compris sans avoir besoin du sujet du TP, mais ne doit pas contenir de copier-coller des questions présentes dans le sujet.

Pour les questions « **Validation** », il faut que vous ayez rédigé et relu votre rédaction à toutes les questions précédents une validation avant de demander à l'enseignant de valider. Vos réponses doivent être **accompagnées d'un minimum d'explications et de justifications**. Une fois que vous êtes sûr de vos réponses vous envoyez un mail à l'enseignant pour lui demander de valider. Votre mail devra obligatoirement avoir pour objet/sujet **[ETRS813] – Validation X** (avec X le numéro de la validation). Votre mail devra comporter obligatoirement le lien vers votre dépôt Github.

Je corrigerai vos réponses par « lot ». Une fois que j'ai toutes les réponses pour une validation, je corrige et je fais soit un retour général, soit un retour particulier. Bien évidemment, vous devez continuer votre travail avant de recevoir une réponse de ma part.

Lors de l'examen écrit vous pourrez avoir des questions concernant le TP.

2 Introduction

Ce document contient toutes les parties qui doivent être traitées durant les 6 séances de TP et les séances de projet réalisées en autonomie. Le sujet est découpé en quatre parties :

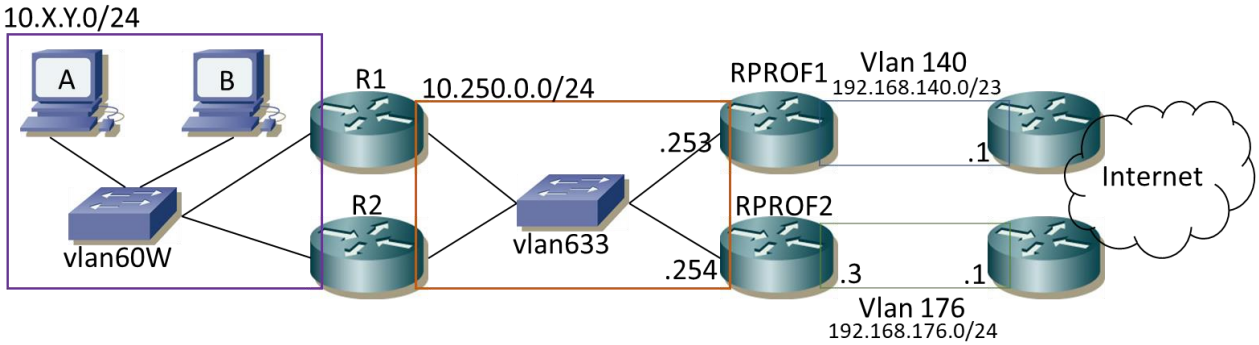
- Mise en place d'une maquette de réseau local avec haute disponibilité.
- Supervision et métrologie avec SNMP.
- Développement d'un script Bash de mesure de débit.
- Mini-projet Prometheus / Grafana / Netflow.

Vous devez consacrer environ 4 à 5 séances de 3.5H pour les 3 premières parties. Vous serez moins guidé pour le mini-projet et vous devez y consacrer également environ 4 à 5 séances de 3.5H.

3 Partie I : mise en place d'une maquette de réseau local avec haute disponibilité

La première séance a pour objectif de mettre en place une maquette de réseau sur laquelle, vous déploierez et testerez par la suite des outils de supervision. Cette maquette a pour particularité d'offrir de la haute disponibilité, ainsi l'accès à l'extérieur du réseau local se fera via deux routeurs. L'ensemble de la maquette sera déployé sur les serveurs que vous avez l'habitude d'utiliser.

3.1 Schéma du réseau



Les routeurs R1 et R2 sont des routeurs du réseau local d'une entreprise. Dans la réalité, il s'agirait plutôt de commutateurs de niveau 3. Les routeurs RPROF1 et RPROF2 sont les routeurs d'accès à Internet de l'entreprise, ici on a deux accès à Internet via deux FAI différents.

Chaque binôme devra mettre en place les machines A et B et les routeurs R1 et R2. Le protocole OSPF devra être configuré sur les routeurs ainsi tous les binômes pourront communiquer entre eux et avec l'extérieur. Les machines A et B sont des machines Linux (Alma Linux 9.2). Sur le schéma le réseau de gestion (vbox-tap0) n'apparaît pas, mais vous savez que chaque machine virtuelle disposera d'une interface sur ce réseau.

Le tableau suivant contient les paramètres que vous devrez utiliser :

Groupe	Binôme	Réseau interne	Switch interne	R1 externe	R2 externe	R1 lo: (/32)	R2 lo (/32)
1	1	10.100.1.0/24	vlan601	10.250.0.1	10.250.0.2	10.10.1.1	10.10.1.2
	2	10.100.2.0/24	vlan602	10.250.0.3	10.250.0.4	10.10.2.1	10.10.2.2
	3	10.100.3.0/24	vlan603	10.250.0.5	10.250.0.6	10.10.3.1	10.10.3.2
	4	10.100.4.0/24	vlan604	10.250.0.7	10.250.0.8	10.10.4.1	10.10.4.2
	5	10.100.5.0/24	vlan605	10.250.0.9	10.250.0.10	10.10.5.1	10.10.5.2
	6	10.100.6.0/24	vlan606	10.250.0.11	10.250.0.12	10.10.6.1	10.10.6.2
	7	10.100.7.0/24	vlan607	10.250.0.13	10.250.0.14	10.10.7.1	10.10.7.2
	8	10.100.8.0/24	vlan608	10.250.0.15	10.250.0.16	10.10.8.1	10.10.8.2
	9	10.200.1.0/24	vlan611	10.250.0.101	10.250.0.102	10.20.1.1	10.20.1.2
	10	10.200.2.0/24	vlan612	10.250.0.103	10.250.0.104	10.20.2.1	10.20.2.2
	Prof	10.100.9.0/24	Vlan609	10.250.0.17	10.250.0.18	10.10.9.1	10.10.9.2

3.2 Etude théorique préparatoire

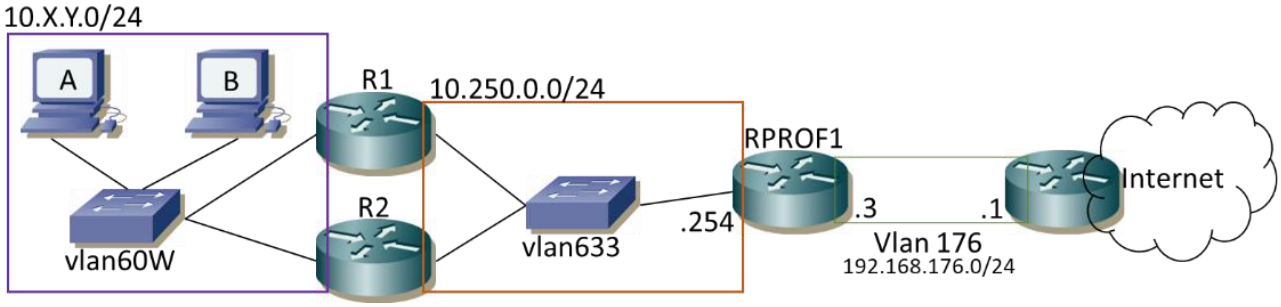
Avant de commencer le déploiement, il est important de déterminer quels résultats on obtiendra lorsque le réseau sera déployé.

Question 1 : Combien de lignes doivent être, théoriquement, présentes dans votre table de routage lorsque la topologie sera complète et lorsque les protocoles auront convergé ? Donner pour chacun de vos routeurs, une table de routage partielle qui contiendra 7 routes choisies de manière pertinente.

Question 2 : Expliquer en 4 lignes le rôle du protocole VRRP qui sera mis en place dans les routeurs.

Question 3 : Expliquer précisément le fonctionnement de VRRP qui permet aux machines A et B d'utiliser le « bon » routeur. Comment fonctionne ce mécanisme lorsque le routeur utilisé devient défaillant ?

On considère le réseau suivant :



Il s'agit de la même topologie que celle à mettre en place mais avec un seul routeur de sortie.

Question 4 : Expliquer en quelques lignes le rôle du protocole OSPF dans le réseau ci-dessus. Justifier entre autres que l'utilisation du routage statique n'aurait pas été pertinente.

I. Faire valider.

3.3 Mise en place et configuration des machines virtuelles

Chaque binôme doit donc configurer 4 machines virtuelles :

- A et B : Alma Linux 9.2
  - /mnt/public_nas_maurienne/virtualmachines/VirtualBox/Alma9.ova
- R1 et R2 : Routeur virtuel Cisco CSR1000v (voir commandes ci-après).

La mise en place et la configuration va se dérouler en quatre étapes :

- Etape 1 : Création des machines virtuelles A et R1, configuration IP de ces VMs
- Etape 2 : Configuration du protocole de routage OSPF dans R1
- Etape 3 : Création des machines virtuelles B et R2, configuration IP et OSPF
- Etape 4 : Mise en place de VRRP

Si c'est le membre 1 du binôme qui déploie A et R1, il faudra que ce soit l'autre membre du binôme qui déploiera B et R2.

Les commandes Cisco pour la configuration des routeurs ne sont pas données, vous devez les rechercher sur Internet si vous ne les connaissez pas.

3.3.1 Etape 1 : Création des machines virtuelles A et R1, configuration IP de ces VMs

A la fin de cette étape, les machines A et R1 doivent être démarrées et leur paramètres IP configurés. Dans cette première étape, le routeur R1 n'est pas complètement configuré : il n'y a pas de protocole de routage et il n'y a pas de routes statiques.

Question 5 : Rédiger les tests que vous mettrez en œuvre à la fin de cette étape pour valider le fonctionnement du réseau. L'objectif est d'écrire le moins de tests possibles pour tester le plus de fonctionnalités possibles.

Exemples de commandes pour la mise en place de R1 :

```
vboxmanage import
/mnt/public_nas_maurienne/virtualmachines/VirtualBox/CSR1000v_Amsterdam.ova
vboxmanage modifyvm " CSR100v" --name 813-R1
vboxmanage modifyvm 813-R1 --nic1 bridged --bridgeadapter1 vbox-tap0
vboxmanage modifyvm 813-R1 --nic2 bridged --bridgeadapter2 vlan609 ##A adapter
vboxmanage modifyvm 813-R1 --nic3 bridged --bridgeadapter3 vlan633
vboxmanage modifyvm 813-R1 --nicpromisc2 allow-all
```

La dernière commande active le « promiscuous mode » de la carte 2. Si vous avez bien réalisé l'étude théorique, vous avez dû voir que VRRP crée une nouvelle adresse MAC pour les routeurs. Par défaut Virtualbox émule un commutateur qui ne transmet des trames qu'aux adresses MAC des machines virtuelles. Le mode « promiscuous » indique au commutateur de VirtualBox de retransmettre à une machine virtuelle les trames même si l'adresse MAC n'est pas celle de la carte de la machine virtuelle.

Pour vérifier la configuration de la VM puis la démarrer :

```
vboxmanage showvminfo 813-R1
vboxmanage startvm 813-R1 --type headless
```

Une fois le routeur démarré, ce qui prend environ 3 minutes, déterminer son adresse IP sur le réseau de gestion (**get-ip-vm.sh**), puis s'y connecter en telnet **password** puis passer en mode privilégié **password**.

Après avoir créé les machines virtuelles, n'oubliez pas de changer leur nom ainsi que le mot de passe root pour s'y connecter ou, dans le cas des routeurs.

Mettre en place la machine A et configurer ces paramètres IP.

- Tester la configuration et vérifier que vous obtenez les résultats attendus.

3.3.2 Etape 2 : Configuration du routage OSPF dans R1

L'objectif de cette partie est de configurer le protocole OSPF dans R1.

Question 6 : Rédiger les tests que vous mettrez en œuvre à la fin de cette étape pour valider le fonctionnement du réseau.

- Mettre en place le protocole OSPF dans R1.
- Tester le fonctionnement du réseau actuellement déployé.

3.3.3 Etape 3 : Création des machines virtuelles B et R2, configuration IP et OSPF

Dans cette partie l'objectif est de mettre en place la machine B et le routeur R2. Il faut également que vous configuriez les paramètres IP de B et du routeur. Pour finir, vous configurez OSPF dans R2 et vous testez l'ensemble du réseau ainsi déployé.

3.3.4 Etape 4 : Mise en place de VRRP

Durant cette étape, vous devez mettre en place le protocole vrrp dans les routeurs. Il vous appartient de trouver les commandes adéquates pour cette mise en place. Une fois le protocole vrrp configuré, vous devrez le tester de manière exhaustive et pertinente.

- La commande : vboxmanage controlvm nom_vm setlinkstateX on | off permet d'activer/désactiver facilement l'interface X de la machine appelée nom_vm. Sur certaines versions de VirtualBox, j'ai pu constaté que la commande ne fonctionnait pas. Vous pouvez alors faire un shut de l'interface.

- Configurer VRRP dans les routeurs. En principe une seule commande permet de mettre en place VRRP.

Question 7 : Rédiger les tests mis en œuvre pour valider le fonctionnement du réseau avec haute disponibilité. Ici les tests demandés doivent permettre de vérifier de manière exhaustive le fonctionnement de VRRP et aussi de l'ensemble des mécanismes assurant de la haute disponibilité. Ainsi, les tests sont un moyen de mieux comprendre le fonctionnement du réseau et de valider les résultats de l'étude théorique.

II. Faire valider

4 Partie II : Supervision et métrologie avec SNMP

4.1 Configuration de SNMPv3 dans les routeurs

Vous devez mettre en place un agent SNMP sur les deux routeurs. Dans un premier temps, on choisit de mettre en place la version 3 de SNMP. L'agent SNMPv3 ne répondra qu'à des requêtes SNMP émises par un utilisateur authentifié. Les requêtes et réponses SNMP seront chiffrées. On utilisera une authentification **sha** avec comme mot de passe **auth_pass** pour l'utilisateur **snmpuser**. Pour la confidentialité, l'algorithme utilisé sera **aes 128** bits avec comme mot de passe **crypt_pass**. Les machines A et B seront utilisées comme client SNMP.

- Remarque : il est également possible de mettre en place un agent SNMP sur les machines Linux. Pour gagner du temps vous ne le ferez pas.

Vous choisirez des valeurs adéquates pour syslocation, syscontact.

- Mettre en place le protocole SNMPv3 dans les routeurs.
- Tester.

Question 8 : Relever la commande snmpget saisie pour récupérer l'objet syslocation et le résultat obtenu.

III. Faire valider.

4.2 Configuration de SNMPv2 dans les routeurs

Dans cette partie, vous allez mettre en place SNMPv2 dans les routeurs. Cette mise en place est uniquement à but pédagogique, vu que SNMPv3 est configuré on n'a pas besoin de mettre en place snmpv2. Le protocole SNMPv2 n'étant pas chiffré, on pourra faire des analyses de trames facilement.

Pour mettre en place snmpv2, il suffit de définir une communauté, vous utiliserez pour cette communauté : **123test123**. Une fois SNMPv2 mis en place, le tester brièvement.

4.2.1 Capture et analyse de trames SNMP

Dans cette partie, vous allez vérifier que les formats des données transmises correspondent bien à ceux qui ont été vus en cours.

Question 9 : Rappeler l'encodage utilisé lors de l'émission de données en SNMP.

Question 10 : Capturer (inclure la transcription au format octet) la trame et sa réponse lorsque vous faites un SNMP-GET sur le MTU de la 2ème interface d'un de vos routeurs. Vérifier de manière détaillée que cette trame correspond bien à la théorie vue en cours.

- Pour capturer des trames, le logiciel tshark est installé sur les machines Linux. L'option -x permet d'afficher la trame capturée sous la forme d'une suite d'octets.

4.2.2 MIB et VRRP

On s'intéresse ici à l'affichage des informations SNMP correspondant au fonctionnement de **VRRP**.

- Rechercher sur Internet le fichier contenant la description ASN de la MIB VRRP.

Question 11 : Copier-coller la ligne du fichier de la MIB VRRP qui indique l'OID relatif de la branche VRRP par rapport à mib-2.

- Saisir la commande :

```
snmpwalk -v2c -c 123test123 10.100.X.Y  vrrpMIB
```

La commande devrait échouer.

- Saisir la commande :

```
snmpwalk -v2c -c 123test123 10.100.X.Y  mib-2.68
```

La commande affiche le contenu des objets associés à la MIB VRRP.

Question 12 : Pourquoi la première commande échoue alors que la deuxième réussie ?

Question 13 : On s'intéresse à la table vrrpOperTable. Donner l'OID par rapport à mib-2 de cette table. Relever dans la vrrpOperTable de R1 et expliquer les 8 premières colonnes et comment est constitué l'index.

IV. Faire valider.

4.3 Métrologie

On s'intéresse dans cette partie aux possibilités offertes par SNMP pour faire de la métrologie. On se focalisera sur la mesure du débit et notamment sur la précision de cette mesure.

4.3.1 Génération de trafic avec iperf

Pour générer du trafic nous utiliserons le programme iperf. Ce programme va nous permettre de générer un flux de données régulier et que l'on maîtrise.

- Installer iperf (version 3) sur vos machines Linux.
- Iperf fonctionne sur le mode client-serveur : une machine cliente se connecte à une machine serveur. Un flux de données est alors émis entre ces deux machines.
- Lancer iperf en mode serveur sur la machine B et iperf en mode client sur la machine A.

```
Sur B : iperf3 -s
Sur A : iperf3 -c @IP-B
```

Question 14 : Quel est le protocole de transport utilisé pour le test de débit entre les deux machines ? Quelle est la durée pendant laquelle la mesure est effectuée ?

L'option **-b** (bandwidth) permet de limiter le débit.

- Vérifier que l'option -b est plus précise avec un flux udp qu'avec un flux tcp.
- Générer un flux de 500 kbit/s entre vos deux machines. Capturer simultanément les trames échangées et les enregistrer dans un fichier de capture :

```
tshark -i enp0s8 udp port 5201 -w /tmp/capture-500k.pcap
```

- Visualiser les statistiques sur le fichier de capture :

```
capinfos /tmp/capture-500k.pcap
```

Question 15 : Pourquoi selon vous le débit calculé par capinfos est légèrement plus élevé que le débit généré par iperf ?

4.3.2 Relever de compteurs de données SNMP et mesure de débit

L'objectif de cette partie est de vérifier que les compteurs d'octets associés aux interfaces permettent d'avoir une vision précise du nombre de bits entrant/sortant d'une machine. Ces compteurs d'octets accessibles en SNMP permettront donc de calculer les débits entrant/sortant d'une machine. Pour générer du trafic à travers le routeur, il faut mettre la machine iperf serveur à l'extérieur du réseau. Pour cela vous pouvez, temporairement, placer sa carte réseau sur le vlan140 et configurer la connexion réseau associée en mode dhcp.

```
vboxmanage controlvm 813-B nic2 bridged vlan140
Puis dans la machine B
nmcli> set ipv4.method auto
nmcli> remove ipv4.gateway
nmcli> remove ipv4.dns
nmcli> save
nmcli> activate
```

Question 16 : Les compteurs d'octets sont disponibles en version 32 bits (ifInOctets, ifOutOctets) ou en version 64 bits (ifHCInOctets, ifHCOutOctets). Justifier précisément quels OID il faut utiliser.

Question 17 : Décrire une manipulation « simple » permettant de trouver le débit entrant ou sortant du réseau en utilisant les compteurs d'octets snmp. Comparer les débits obtenus par SNMP avec les débits générés.

Remarques :
Pour faciliter la manipulation vous avez la possibilité de générer un flux avec iperf pendant une durée plus grande que 10s (option -t).
Toujours pour faciliter la mesure vous pouvez faire un script de 3 lignes et utiliser la commande sleep entre vos deux commandes snmpget.

V. *Faire valider.*

5 Partie III : Script bash de mesure de débit en SNMP (1 séance)

Durant cette partie vous devez développer un script Bash qui permet de mesurer les débits des flux entrant ou sortant d'un équipement.

Par exemple votre script doit permettre de connaître le débit sortant d'une interface d'un routeur, avec un relevé effectué toutes les minutes sur une très longue période. Les mesures de débit sont stockées dans un fichier.

On fixe comme périodicité minimale de la mesure 1 minute et comme périodicité maximale 30 minutes.

Pour arriver à ce résultat vous devez écrire un script qui fonctionnera sur la machine Linux. Ce script devra être écrit en Bash. Votre script ne devra pas surcharger inutilement le système il devra donc fonctionner conjointement avec l'ordonnanceur Cron ou via la fonctionnalité timer disponible avec Systemd. En effet, il n'est pas pertinent de créer un processus qui serait toujours actif en tâche de fond. La version finale de votre script devra être la plus générique possible.

L'objectif du script est d'obtenir la valeur du débit moyen entrant ou sortant d'une interface du routeur.

Le script sera donc lancé toutes les X minutes par Cron ou systemctl, il :

- Récupère en SNMP la valeur du compteur d'octets en SNMP
- Calcule le débit moyen depuis la mesure précédente
- Enregistre ces informations dans un fichier qui contiendra donc, la date et l'heure de la mesure, la valeur du compteur d'octets, la valeur de débit moyen depuis la dernière mesure (s'il y en a une).

Le fichier pourra contenir d'autres informations si nécessaire.

Dans sa version finale, le script recevra en paramètres :

- le nom du fichier dans lequel il devra stocker les résultats
- un OID à demander
- l'adresse IP de l'agent SNMP interrogé
- la communauté SNMP

Le développement du script et la mise en place finale vont se faire progressivement en plusieurs étapes que je vous impose. Il faut bien évidemment que votre script soit commenté. Vous devez également mettre à profit les fonctionnalités de « versionning » de Git et Github lors du développement de votre script. Je dois pouvoir retrouver l'évolution de votre développement via les commits que vous aurez réalisés.

Par souci d'efficacité nous nous limiterons à l'utilisation de snmp V2 (pas d'authentification, pas de chiffrement).

Question 18 : Il aurait été possible de ne pas utiliser le cron ou les timers systemd pour ce script, par exemple en utilisant la fonction sleep. Pourquoi est-il plus pertinent d'utiliser le cron ou les timers systemd plutôt que la fonction sleep ?

5.1 Récupération du compteur d'octets

Durant cette première étape, vous devez écrire les premières lignes de code pour récupérer la valeur du compteur d'octets de l'interface 3.
Pour l'instant, toutes les informations seront codées « en dur » dans le programme.

Le squelette du script en bash est le suivant :

```
#??

oid="??"
agent_ip="??"
community="??"

value=$(snmpget -v2c …)

echo" ${value}"
```

Voilà le résultat du script lorsqu'il est exécuté :

```
[root@813-A ~]# ./snmp-1.sh
34499394
```

On peut noter que le résultat d'une commande snmpget « classique » est le suivant :

```
IF-MIB::ifHCOutOctets.X = Counter64: 34501313
```

Ici on a besoin de récupérer que la valeur, dans un premier temps, il est possible de simplifier l'affichage du résultat de la commande snmpget avec une option -O adéquate. En ce qui me concerne j'ai par la suite utilisé la commande **cut** pour sélectionner le champ qui me convenait.

5.2 Gestion de la date et enregistrement des résultats dans un fichier.

On part du script précédent, on va rajouter maintenant les actions pour stoker les données dans un fichier ainsi que pour gérer la date.

Un débit s'exprime en bit/s. Il faut donc pour obtenir le débit savoir combien de secondes se sont écoulées entre deux mesures. La commande date permet d'afficher la date.

- Chercher l'option de la commande date qui permet d'afficher la date au format nombre de secondes écoulées depuis 01/01/1970.

Pour stocker les données dans un fichier nous utiliserons la redirection de commande « >> ».

Le squelette du script est le suivant :

```
#??

oid=" "
agent_ip="??"
community="??"
filename="??"
```

```
value=$(snmpget -v2c -Oq -c ${community} ${agent_ip} ${oid} | cut -d " " -f 2)
date=$(date ??)
echo "???" >> "${filename}"
```

Lorsqu'on exécute le script on obtient le résultat suivant :

```
[root@813-A ~]# ./snmp-2.sh
[root@813-A ~]# cat throughput_int1.txt
1613912716;34512886
```

1613912716 est la date en nombre de secondes depuis le 01/01/1970.
34512886 est la valeur du compteur d'octets.

5.3 Lecture de la dernière ligne du fichier, calcul et enregistrement du débit

On rajoute au script les actions pour lire la dernière ligne du fichier où sont stockées les mesures et pour calculer le débit et bien évidemment stocker les résultats dans un fichier.

```
[root@813-A ~]# cat throughput_int1.txt
1613914655;56620183;735
1613914666;58548447;1402
1613914675;59192127;572
1613914726;66267349;1109
```

- Tester le fonctionnement de votre script.

Question 19 : Consigner la procédure de tests et les résultats des tests dans votre compte-rendu.

VI. Faire valider par l'enseignant

5.4 Gestion du fichier vide et gestion du rebouclage du compteur d'octets.

Lorsque le fichier est créé pour la première fois, il ne contient pas de donnée. Il n'est alors pas possible de calculer le débit.

- Mettre en place dans votre script, cette gestion de la première exécution.

Le compteur d'octets peut reboucler.

Question 20 : Expliquer le problème posé par le rebouclage du compteur. Expliquer la solution à mettre en place.

- Mettre en place dans votre script, cette prise en compte du rebouclage.

5.5 Utilisation du cron pour que le script s'exécute toutes les minutes

Configurer le système pour que le script s'exécute automatiquement toutes les minutes. Attention à bien stocker les fichiers dans des emplacements adéquats.

Question 21 : Décrire la configuration mise en place. Indiquer dans votre compte-rendu l'adresse IP de la machine sur laquelle le script s'exécute et les commandes permettant de valider le fonctionnement de votre système de supervision du débit. L'enseignant doit pouvoir se connecter à votre machine pour valider votre travail.

VII. Faire valider par l'enseignant

5.6 Script générique

Si ce n'est déjà fait il ne reste plus qu'à rendre votre script générique. C'est-à-dire que toutes les données codées en dur dans le script (adresse IP, nom du fichier, identifiant d'interface interrogée) doivent pouvoir être fournies sous la forme d'arguments.

Question 22 : Rendre votre script le plus générique possible. Passer environ 1.5 heures sur cette partie. Le script développé lors du paragraphe précédent ne doit plus être modifié pour que l'enseignant puisse valider son fonctionnement. Il faut donc que vous utilisiez un nouveau fichier. Votre fichier devra dans son nom contenir la chaine ***_generic*** *pour pouvoir le différencier du script précédent.*

Il resterait sûrement beaucoup d'autres fonctionnalités à développer dans ce script, par exemple, passer un nom d'interface et une direction plutôt qu'un OID, mettre en place des mécanismes de gestion des erreurs … Bien qu'intéressantes, ces fonctionnalités prendraient trop de temps à développer et je préfère que vous vous consacriez à d'autres aspects de la supervision.

6 Partie IV : Projet Prometheus / Grafana / Netflow / Logs

Vous continuez de travailler sur la maquette de réseau créée lors des séances précédentes. Lors de ce projet vous allez déployer des outils de supervision que l'on pourrait qualifier de relativement modernes.

L'outil de supervision que vous devez déployer est Prometheus. Prometheus fait partie d'un vaste écosystème, vous devrez donc également mettre en place d'autres outils de cet écosystème. Parmi ces autres outils, vous utiliserez Grafana pour la création de tableaux de bord (dashboard) de supervision.

L'ensemble de la solution sera déployé sous la forme de conteneurs Docker. Il est recommandé d'instancier les services répondant aux mêmes fonctionnalités via la commande Docker Compose. Tous les fichiers importants devront être déposés sur votre dépôt Github. Les fonctionnalités mises en place devront être testées, si possible de manière automatisée.

Pour les validations, vous devez préciser dans le mail envoyé à l'enseignant l'adresse et les identifiants (login/mot de passe) permettant d'accéder à votre Grafana.

Les grandes étapes du déploiement sont les suivantes :

- Recherches et documentations sur Prometheus et Grafana.
- Mise en place de Prometheus et Grafana pour la supervision des routeurs.
- Récupération via SNMP des compteurs d'interface et création d'un dashboard Grafana pour l'affichage des débits des interfaces.
- Rédaction d'une procédure de tests et éventuellement de scripts de tests.

VIII. Faire valider par l'enseignant.

- Réalisation d'un dashboard faisant apparaître les noms des interfaces ayant le plus de débit durant la dernière heure.
- Ajout sur le dashboard du « **sysuptime** » des routeurs.
- Ajout des informations VRRP de routeur actif/passif.

- Rédaction d'une synthèse de 2 pages présentant le fonctionnement « technique » de Prometheus et des outils liés à Prometheus. Cette synthèse s'adresse à des ingénieurs réseaux et systèmes. Elle est entre autres le résultat des différentes recherches que vous avez effectuées mais aussi de l'expertise que vous avez acquise lors des mises en place précédentes.

IX. Faire valider par l'enseignant.

- Mise en place d'un serveur Web Docker sur A avec au moins 3 pages différentes situées à des URL différentes.
- Création de deux nouveaux dashboard :
  - Informations classiques de supervision pour la machine A
  - Informations classiques de supervision pour un serveur WEB
- Rédaction d'une procédure de tests et éventuellement de scripts de tests.

X. Faire valider par l'enseignant.

- Mise en place d'un collecteur Netflow dans les routeurs.
- Récupération des données Netflow si possible dans Prometheus, sinon en utilisant une autre interface.
- Rédaction d'une procédure de tests et éventuellement de scripts de tests.

XI. Faire valider par l'enseignant.