

Résumé

L'objectif de ce support de travaux pratiques est de mettre en évidence les caractéristiques de fonctionnement du protocole de routage OSPF. Cette illustration s'appuie sur des liens de type Ethernet et sur l'utilisation des VLANs. Les questions sont présentées comme une introduction pas à pas au protocole de routage OSPF. On débute avec la mise en place d'une topologie réseau type basée sur le routage inter-VLAN, puis on implante les instances de démons de routage.

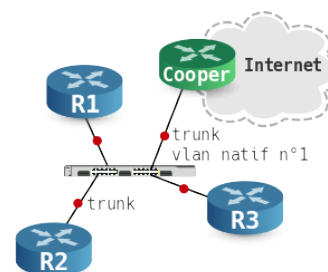


Table des matières

1. Copyright et Licence	2
1.1. Méta-information	2
1.2. Conventions typographiques	2
2. Architecture réseau étudiée	3
2.1. Topologie type	3
2.2. Plan d'adressage	4
3. Préparation des routeurs	6
4. Communications entre routeurs	8
5. Configuration OSPF de base	9
6. Publication d'une route par défaut via OSPF	10
7. Ajout de routes fictives	11
8. Adaptation de la métrique de lien au débit	12
9. Manipulations sur machines virtuelles	13
9.1. Préparation du commutateur Open vSwitch	13
9.2. Préparation des routeurs	15
9.3. Choix des identifiants de routeur OSPF	15
9.4. Table de routage du système hôte	16
10. Documents de référence	17

1. Copyright et Licence

Copyright (c) 2000,2015 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2015 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.3 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

1.1. Méta-information

Ce document est écrit avec **DocBook XML** sur un système **Debian GNU/Linux**. Il est disponible en version imprimable au format PDF : interco.ospf.q.pdf.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution Debian GNU/Linux qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes :

- procps - Utilitaires pour le système de fichiers /proc
- iproute2 - Outils de contrôle du trafic et du réseau
- ifupdown - Outils de haut niveau pour configurer les interfaces réseau
- iputils-ping - Outils pour tester l'accessibilité de noeuds réseaux
- quagga - BGP/OSPF/RIP routing daemon

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite \$ ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite # nécessite les privilèges du super-utilisateur.

2. Architecture réseau étudiée

2.1. Topologie type

La topologie réseau étudiée peut être présentée sous deux formes distinctes : logique et physique.

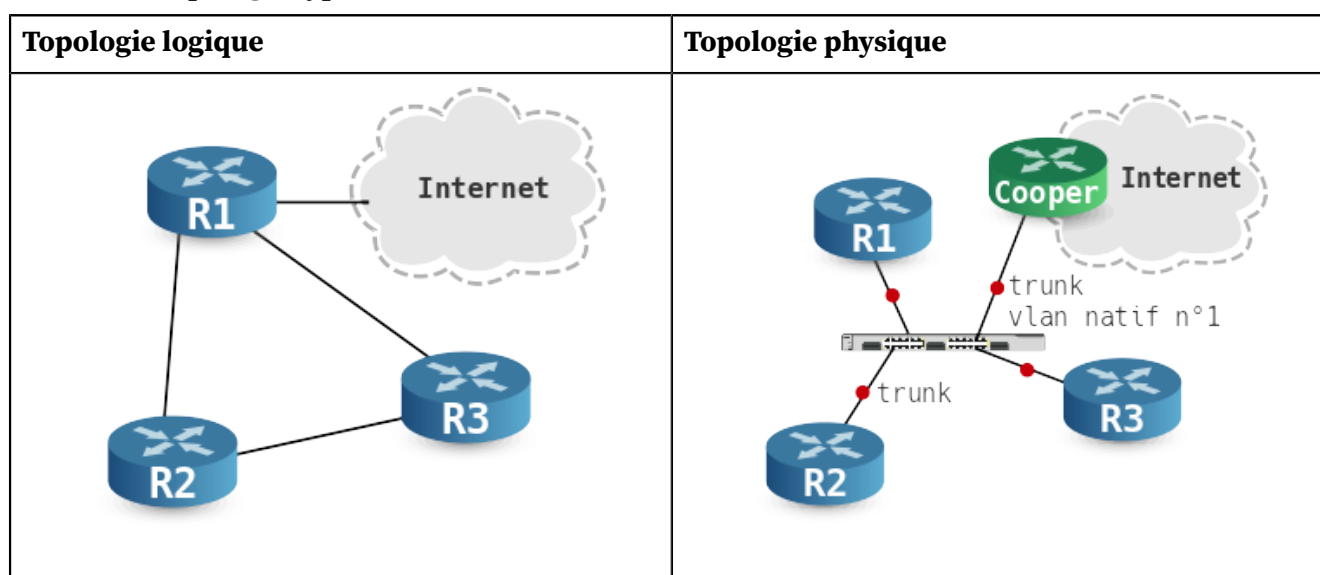
Topologie logique

On retrouve un grand classique dans l'introduction aux protocoles de routage dynamiques : le triangle. Tous les liens sont de type LAN.

Topologie physique

On s'appuie sur le support **Routage Inter-VLAN** pour constituer une topologie physique à base de réseaux locaux virtuels ou VLANs. On fait correspondre à chaque lien de la topologie logique en triangle un numéro de VLAN défini.

Tableau 1. Topologie type étudiée



Après avoir mis en œuvre la topologie physique en s'appuyant sur le support de la séance de travaux pratiques précédente : **Routage Inter-VLAN**, on implante les démons de routage OSPF sur les trois routeurs R1, R2 et R3.

Cette séance se limite à l'étude du routage dynamique à l'intérieur d'une aire unique. La seule «frontière» de communication inter-aire visible est constituée par le lien vers l'Internet. Cette route par défaut sera redistribuée via OSPF par le routeur R1 aux autres routeurs. On verra alors un exemple de route externe dans les bases de données OSPF.

On profite aussi de cette introduction pour employer une technique très répandue pour ajouter «artificiellement» des entrées de tables de routage en s'appuyant sur des interfaces virtuelles de type dummy équivalentes à des interfaces de boucle locale.

Pour les besoins de rédaction des questions et réponses de ce support, la topologie a été mise en œuvre sur machines virtuelles KVM avec le commutateur Open vSwitch fourni avec le paquet openvswitch-switch. Les éléments de réponse aux questions dépendent donc de cette mise en œuvre. Pour la séance de travaux pratiques «réelle», il convient donc de se conformer strictement au plan d'adressage fourni ci-après.

2.2. Plan d'adressage

Comme dans le support sur l'introduction au routage inter-VLAN, le seul point de configuration imposé est le raccordement au réseau d'interconnexion avec les routeurs de l'infrastructure de travaux pratiques.

- swd2.infra.stri en salle 211
- swd1.infra.stri en salle 213

Le raccordement au commutateur de la couche distribution utilise le port «de numéro le plus élevé» de chaque commutateur de couche accès ; gi0/2 normalement. Ces liens montants doivent être configurés en mode trunk en utilisant le VLAN natif numéro 1. Le réseau IP correspondant au VLAN numéro 1 a pour adresse : 172.16.0.0/20

Point important, la lecture de la section «Plan d'adressage» du document [Architecture réseau des travaux pratiques](#) donne l'adresse des deux routeurs connectés à l'Internet.

Tableau 2. Affectation des rôles, des numéros de VLANs et des adresses IP

Groupe	Commutateur	Poste	Rôle	router-id	VLAN	Interface	Réseau
1	sw5.infra.stri	alderaan	R10	0.0.0.10	1 (natif)	eth0	172.16.1.1/20
					221	eth0.221	10.1.21.1/26
					231	eth0.231	10.1.31.1/26
		bespin	R20	0.0.0.20	221	eth0.221	10.1.21.2/26
					232	eth0.232	10.1.32.2/26
					231	eth0.231	10.1.31.3/26
centares	R30	0.0.0.30	232	eth0.232	10.1.32.3/26		
			232	eth0.232	10.1.32.3/26		
2	sw6.infra.stri	coruscant	R40	0.0.0.40	1 (natif)	eth0	172.16.2.1/20
					241	eth0.241	10.2.41.1/26
					251	eth0.251	10.2.51.1/26
		dagobah	R50	0.0.0.50	241	eth0.241	10.2.41.4/26
					254	eth0.254	10.2.54.4/26
		endor	R60	0.0.0.60	251	eth0.251	10.2.51.5/26
254	eth0.254				10.2.54.5/26		
1	sw7.infra.stri	felucia	R70	0.0.0.70	1 (natif)	eth0	172.16.3.1/20
					261	eth0.261	10.3.61.1/26
					271	eth0.271	10.3.71.1/26
		geonosis	R80	0.0.0.80	261	eth0.261	10.3.61.6/26
					276	eth0.276	10.3.76.6/26
		hoth	R90	0.0.0.90	271	eth0.271	10.3.71.7/26
276	eth0.276				10.3.76.7/26		
4	sw8.infra.stri	mustafar	R100	0.0.0.100	1 (natif)	eth0	172.16.4.1/20

Groupe	Commutateur	Poste	Rôle	router-id	VLAN	Interface	Réseau
					281	eth0.281	10.4.81.1/26
					291	eth0.291	10.4.91.1/26
		naboo	R110	0.0.0.110	281	eth0.281	10.4.81.8/26
					298	eth0.298	10.4.98.8/26
		tatooine	R120	0.0.0.120	291	eth0.291	10.4.91.9/26
					298	eth0.298	10.4.98.9/26

Le positionnement des 4 commutateurs est référencé dans le support [Architecture réseau des travaux pratiques](#).

3. Préparation des routeurs

La première étape consiste à mettre en place la topologie physique.

- Vérifier l'installation du paquet quagga avant de brasser les postes sur les commutateurs non programmés.

```
$ aptitude search ~iquagga
i   quagga                - BGP/OSPF/RIP routing daemon
```

- Vérifier que la fonction de routage des paquets IPv4 est active au niveau noyau.

```
$ cat /proc/sys/net/ipv4/ip_forward
1
```

Si ce n'est pas le cas, il est possible d'éditer le fichier `/etc/sysctl.conf` pour fixer les valeurs des paramètres de configuration des protocoles de la pile TCP/IP dans le noyau Linux. Voir la section [Fonctions réseau d'une interface du support Configuration d'une interface de réseau local](#).

```
# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.proxy_arp = 0
```

- Créer les sous-interfaces associées aux VLANs sur chacun des routeurs R1, R2 et R3 à l'aide du script suivant :

```
#!/bin/bash

for vlan in $*
do
  ip link add link eth0 name eth0.$vlan type vlan id $vlan
  ip link set dev eth0.$vlan txqueuelen 10000
  tc qdisc add dev eth0.$vlan root pfifo_fast
  ip link set dev eth0.$vlan up
done
```

Sur le routeur R1, on utilise le script avec les numéros de VLANs 12 et 13 par exemple.

```
r1:~# sh ./subinterfaces.sh 12 13
```

On adapte l'utilisation du même script aux routeurs R2 et R3 avec les numéros de VLANs concernés.

- Activer les démons zebra et ospfd sur chaque routeur en éditant le fichier `/etc/quagga/daemons` et en remplaçant `no` par `yes`.

```
r1:~# grep -v '^#' /etc/quagga/daemons
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

- Créer les fichiers de configuration de base pour les deux démons zebra et ospfd sur chaque routeur en utilisant les patrons livrés avec le paquet quagga.

```
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/zebra.conf.sample zebra.conf
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/ospfd.conf.sample ospfd.conf
r1:/etc/quagga# chown quagga zebra.conf ospfd.conf
```

- Éditer le patron du fichier de configuration du démon zebra en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat zebra.conf
! -*- zebra -*-
!
hostname R1-zebra
password zebra
enable password zebra
!
log file /var/log/quagga/zebra.log
```

- Éditer le patron du fichier de configuration du démon ospfd en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat ospfd.conf
! -*- ospf -*-
!
hostname R1-ospfd
password zebra
enable password zebra
!
log file /var/log/quagga/ospfd.log
```

- Compléter la configuration des interfaces dans le démon zebra de façon à fixer la bande passante de chaque interface active.

```
r1:/etc/quagga# grep -1 bandwidth zebra.conf
interface eth0
  bandwidth 100000
  ipv6 nd suppress-ra
--
interface eth0.12
  bandwidth 100000
  ipv6 nd suppress-ra
--
interface eth0.13
  bandwidth 100000
  ipv6 nd suppress-ra
```



Note

Contrairement à un routeur «intégré» avec un système d'exploitation dédié, le démon de routage statique n'a pas directement accès aux interfaces matérielles. Or, sur un système GNU/Linux, le débit d'une interface nommée `eth0` peut aller de 10Mbps à 10Gbps. Sans information spécifique du noyau, l'application «service de routage» n'a aucun moyen de connaître le débit exact de l'interface `eth0`. C'est la raison pour laquelle il est nécessaire de paramétrer manuellement les débits de chaque interface dans la configuration du démon `zebra`.



Avertissement

Ce dernier paramétrage est essentiel dans le **calcul des métriques** et le fonctionnement du protocole de routage OSPF. Si les calculs de métriques pour les liens actifs sont erronés, le choix des routes à emprunter pour faire transiter le trafic utilisateur entre deux routeurs peut lui aussi être erroné.

4. Communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider les communications IP entre chaque routeur et de visualiser les tables de routage déjà connues.

Q1. Quelles sont les opérations à effectuer pour implanter les adresses IP des interfaces correspondant à chacun des VLANs routés ?

Au niveau liaison, les sous-interfaces ont déjà été configurées avec le script `subinterfaces.sh`. Il reste à paramétrer les adresses IP de ces sous-interfaces.

Q2. Quelles sont les opérations à effectuer pour valider les communications IP entre routeurs ?

Lancer les tests ICMP usuels entre chaque routeur sur chaque lien actif.

Q3. Quelles sont les opérations à effectuer pour visualiser la table de routage existante d'un routeur au niveau système et au niveau du démon de routage statique zebra ?

Utiliser la commande `ip` pour visualiser la table de routage puis afficher la même table de routage à partir de la connexion au démon zebra avec la commande usuelle du système Cisco™ IOS.

Q4. Quelle est l'opération à effectuer pour activer la fonction routage du noyau Linux ?

Reprendre l'instruction présentée dans le document `Configuration d'une interface de réseau local : activation du routage`.

L'opération doit être répétée sur chacun des trois routeurs pour que le protocole de routage dynamique puisse fonctionner normalement.

5. Configuration OSPF de base

Dans cette section, on introduit les premières commandes de configuration du protocole de routage dynamique OSPF qui permettent d'activer le protocole puis d'introduire des entrées de réseau dans la base de données de ce protocole.

- Q5.** Comment peut-on contrôler si le protocole OSPF est actif ou non sur le routeur ?
Une fois connecté au démon `ospfd`, lancer la commande de visualisation globale du protocole. Cette commande est utilisable sur chacun des trois routeurs.
- Q6.** Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF et fixer manuellement l'identifiant du routeur ?



Avertissement

Les identifiants à utiliser lors de la séance de travaux pratiques sont donnés dans le [Tableau 2](#), « Affectation des rôles, des numéros de VLANs et des adresses IP ».

- Q7.** Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF pour les réseaux IPv4 connus de chaque routeur ?
Dans la configuration du démon `ospfd`, ajouter une entrée de réseau pour chaque lien connu du routeur. La liste des liens connus correspond aux entrées marquées c de la table de routage visualisée à partir du démon `zebra`.
- Q8.** Quelle est la commande qui permet de visualiser l'état des interfaces actives du routeur vis-à-vis du protocole de routage OSPF ?
Les interfaces sont automatiquement activées dès qu'une entrée de réseau est saisie au niveau du démon `ospfd` et que l'adresse IP de l'interface correspond à ce réseau.
- Q9.** À partir des informations affichées dans la question précédente, retrouver l'identifiant de routeur et le type de réseau, repérer et identifier la présence d'un autre routeur sur le même réseau.
Pour chaque interface vue du démon de routage OSPF, repérer les informations relatives au type de réseau et au décomptage des routeurs voisins.
- Q10.** Comment peut-on vérifier que l'algorithme SPF du protocole OSPF a été correctement exécuté, que le protocole a convergé et que les entrées de table de routage ont été publiées ?
Visualiser les listes des routeurs voisins puis la liste des routes présentes dans la base de données du démon `ospfd`. Faire la correspondance entre les métriques affichées et les bandes passantes de chaque lien.
- Q11.** Quel est le mode d'affichage de la table de routage du système qui offre le plus d'informations ? Identifier le « lieu de la synthèse » de tous les canaux d'information sur la table de routage d'un routeur.
- Q12.** Comment visualiser la table de routage au niveau système ?
Utiliser une commande usuelle de visualisation de la table de routage.

6. Publication d'une route par défaut via OSPF

Dans la topologie logique étudiée, le routeur R1 dispose d'un lien montant vers l'Internet. On peut donc considérer que ce lien est la route par défaut vers tous les réseaux non connus de l'aire OSPF contenant les trois routeurs.

Il est possible de publier une route par défaut via le protocole OSPF depuis le routeur R1 vers les routeurs R2 et R3.

Voici, pour mémoire, une copie de la base de données OSPF avant la mise en place de la publication de route par défaut. On reconnaît les LSAs (Link State Advertisement) de type 1 et 2 qui correspondent respectivement aux annonces de routeurs et de réseaux.

```
R1-ospfd# sh ip ospf database
```

```
    OSPF Router with ID (0.0.0.1)
```

```
        Router Link States (Area 0.0.0.0)
```

Link ID	ADV Router	Age	Seq#	CkSum	Link count
0.0.0.1	0.0.0.1	210	0x80000021	0xebbe	2
0.0.0.2	0.0.0.2	1345	0x8000001e	0xe7ae	2
0.0.0.3	0.0.0.3	1654	0x8000001f	0xcf44	3

```
        Net Link States (Area 0.0.0.0)
```

Link ID	ADV Router	Age	Seq#	CkSum
10.1.12.1	0.0.0.1	1670	0x80000018	0xda8b
10.1.13.1	0.0.0.1	1120	0x80000018	0xdd86
10.1.23.2	0.0.0.2	1405	0x80000018	0x69ed

Q13. Quelle est la condition préalable à respecter pour que le routeur R1 soit en mesure de publier une route par défaut via son démon OSPF ?

Parmi toutes les méthodes de redistribution de routes dans la protocole OSPF, il en existe une dédiée à l'injection de route par défaut dans une aire normale : Voir **Open Shortest Path First (OSPF)**.

Q14. Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPF ?

Rechercher le mot clé `default` dans la liste des commandes relatives au démon `ospfd`.

Q15. Comment la publication de route par défaut apparaît-elle sur les différents routeurs OSPF ?

Relevez la métrique de la route par défaut sur les routeurs qui n'ont pas une connexion directe vers l'Internet.

7. Ajout de routes fictives

L'introduction de nouvelles entrées fictives dans les tables de routage est une pratique très répandue. Elle permet de qualifier le bon fonctionnement du filtrage réseau ou d'un service Internet sans ajouter de matériel. Cette section décrit justement la mise en place d'un service Web de test.

Q16. Quelles sont les opérations à effectuer pour pouvoir utiliser des interfaces réseau virtuelles de type boucle locale sur un système GNU/Linux ?

Avec un noyau Linux, il est conseillé d'utiliser les interfaces baptisées `dummy` pour ce genre d'usage. Les opérations à effectuer consistent à charger le module du même nom en mémoire et à appliquer une nouvelle configuration IP.

Q17. Quelles sont les opérations à effectuer pour installer un service Web en écoute exclusivement sur l'adresse IP de l'interface `dummy0` ?

Pour aller au plus court, on installe le paquet `apache2` et on édite la configuration du service de façon à limiter l'accès à l'adresse IP voulue.

Q18. Comment ajouter la route correspondant au nouveau réseau `10.1.3.0/29` dans le domaine de routage OSPF ?

Comme dans le cas de la mise en place des autres routes dans la configuration du démon `ospfd`, on ajoute une entrée `network` dans l'instance OSPF du routeur.

Q19. Comment valider l'accès à ce service Web depuis les autres réseaux ?

En respectant l'ordre des protocoles de la pile TCP/IP, on commence par valider la connectivité au niveau réseau avant de passer à la couche application.

8. Adaptation de la métrique de lien au débit

Par défaut, le calcul de métrique du le protocole OSPF se fait à partir de l'expression : $10^8 / \text{Bande_Passante_du_lien}$.

Cette règle a été établie à une époque où l'utilisation d'un lien à 100Mbps devait être considéré comme une situation d'exploitation futuriste. Aujourd'hui, les liens à 100Mbps sont monnaie courante et les 100Gbps vont bientôt le devenir.

Cette section traite donc de la configuration des instances de protocole de routage OSPF utilisant des liens avec une capacité supérieure à 100Mbps.

Q20. Quelle est l'instruction à utiliser pour que le calcul de métrique OSPF se fasse sur la base d'un débit de lien à 10Gbps ?

Rechercher le mot clé bandwidth dans la liste des instructions de configuration du démon ospfd.

Q21. Comment modifier le débit d'un lien à 1Gbps ?

Normalement, le débit d'un lien est directement extrait des paramètres de l'interface connectée au lien. Dans le cas d'interface qui n'ont «aucune réalité physique», ce débit peut être attribué arbitrairement par configuration. On doit rechercher dans les options des démons zebra et ospfd le moyen d'attribuer un débit aux sous-interfaces de VLANs.

Q22. Comment peut-on identifier le débit d'un lien dans la configuration OSPF ?

Visualiser les paramètres des interfaces réseau depuis la console du démon ospfd.

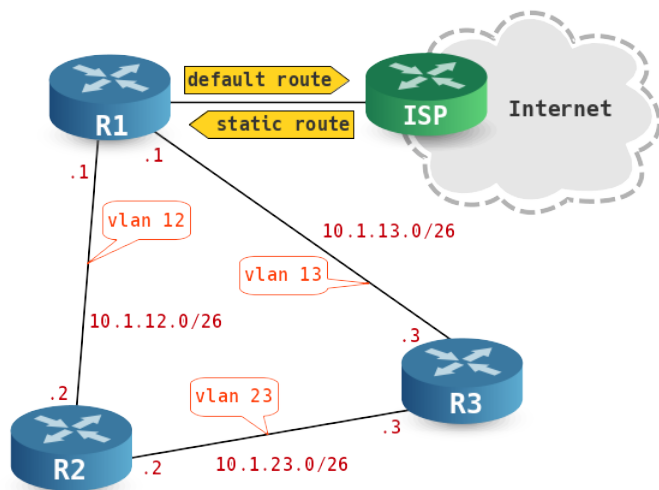
Q23. Quel est le coût d'accès au pseudo service Internet (réseau 10.1.3.0/29) après modification de la référence de calcul de métrique ? Justifier la valeur de métrique obtenue.

À partir des informations de la base de données du démon ospfd, faire la somme des métriques de chaque lien entre les deux extrémités en communication.

9. Manipulations sur machines virtuelles

Il est possible de réaliser l'ensemble des manipulations de ce support à l'aide de trois instances de machines virtuelles et du commutateur **Open vSwitch** présenté dans le guide **Virtualisation système et enseignement**.

Voici quelques éléments sur la mise en œuvre de cette «infrastructure de travaux pratiques». Dans la figure ci-dessous, le routeur baptisé ISP correspond au système hôte sur lequel les systèmes virtuels sont exécutés.



9.1. Préparation du commutateur Open vSwitch

On se réfère au guide **Virtualisation système et enseignement** dans lequel le lancement du commutateur virtuel **Open vSwitch** est intégré à la configuration du système hôte. Voici un extrait du fichier `/etc/network/interfaces` relatif à la configuration du commutateur `swd-host`.

```
# The primary network interface
allow-ovs swd-host
iface swd-host inet manual
    ovs_type OVSBridge
    ovs_ports eth0 vlan1

allow-swd-host vlan1
iface vlan1 inet static
    address 192.0.2.29/26
    gateway 192.0.2.1
    ovs_type OVSPort
    ovs_bridge swd-host
    dns-nameservers 192.0.2.1
    up ip link set txqueuelen 10000 dev $IFACE
    up tc qdisc add dev $IFACE root pfifo_fast

iface vlan1 inet6 static
    address 2001:db8:fe00:8175::1d/64
    gateway 2001:db8:fe00:8175::1
    dns-nameservers 2001:db8:fe00:8175::1

allow-swd-host eth0
iface eth0 inet manual
    pre-up ip link set txqueuelen 10000 dev $IFACE
    up ip link set dev $IFACE up
    down ip link set dev $IFACE down
    ovs_bridge swd-host
    ovs_type OVSPort
```

Dans cette configuration, `vlan1` est l'interface principale du système hôte. Elle sert à la fois au routage et à l'accès aux instances de machines virtuelles depuis les autres réseaux.

La configuration de départ du commutateur sur le système hôte est la suivante :

```
$ sudo ovs-vsctl show
0fb1e80b-37cf-4dce-9a19-17b9fb989610
  Bridge swd-host
    Port "eth0"
      Interface "eth0"
    Port "vlan1"
      Interface "vlan1"
        type: internal
    Port swd-host
      Interface swd-host
        type: internal
  ovs_version: "2.3.0"
```

Partant de cette configuration de départ, on crée 3 cordons de brassage ; un par routeur. On dispose d'un script pour répéter les opérations de base.

```
#!/bin/bash

for patchtap in $*
do
    if [ -z "$(ip addr ls | grep $patchtap)" ]; then
        echo setting $patchtap up
        sudo ip tuntap add mode tap dev $patchtap group kvm
        sudo ip link set dev $patchtap up
    else
        echo $patchtap is already there!
    fi
done
```

Avec cet outil, la création des cordons s'effectue comme suit :

```
$ sh ./patchcables.sh tap1 tap2 tap3
```

Maintenant que les cordons sont disponibles, il faut les brasser sur le commutateur swd-host. L'opération ci-dessous crée trois nouveaux ports sur le commutateur.

```
$ for port in tap1 tap2 tap3; do sudo ovs-vsctl add-port swd-host $port; done
```

Pour respecter la topologie physique proposée dans l'énoncé, ces trois ports doivent être configurés en mode trunk de façon à ce que chaque routeur puisse router les VLANs qui le concernent. Avec **Open vSwitch**, les paramètres de configuration sont les suivants :

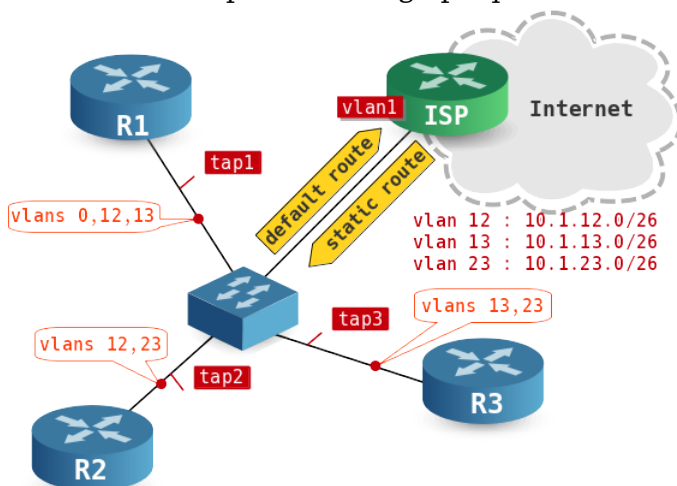
```
$ sudo ovs-vsctl set port tap1 vlan_mode=trunk trunks=0,12,13
$ sudo ovs-vsctl set port tap2 vlan_mode=trunk trunks=12,23
$ sudo ovs-vsctl set port tap3 vlan_mode=trunk trunks=13,23
```



Note

La particularité du commutateur **Open vSwitch** est d'utiliser le numéro 0 comme VLAN par défaut. Dans notre cas, ce VLAN 0 devient le VLAN natif dans lequel circulent les trames sans étiquettes IEEE802.1Q.

On aboutit à la représentation graphique suivante :



La table de correspondance entre VLAN et liaison devient :

VLAN numéro 0	Liaison R1 <-> ISP
VLAN numéro 12	Liaison R1 <-> R2
VLAN numéro 13	Liaison R1 <-> R3
VLAN numéro 23	Liaison R2 <-> R3

Une fois que les opérations des sections suivantes auront été effectuées, la consultation de la table CAM du commutateur donnera les informations suivantes :

```
$ sudo ovs-appctl fdb/show swd-host
[sudo] password for latu:
port  VLAN  MAC                Age
 11   12   ba:ad:ca:fe:00:02    10
 12   23   ba:ad:ca:fe:00:03     9
 12   13   ba:ad:ca:fe:00:03     9
 10   13   ba:ad:ca:fe:0d:01     9
 10   12   ba:ad:ca:fe:0c:01     4
 11   23   ba:ad:ca:fe:00:02     1
 1    0    de:56:80:40:1d:ed     0
 2    0    42:d9:5d:6c:89:d8     0
```

9.2. Préparation des routeurs

Sachant que l'on dispose d'une image master sur laquelle est installée un système Debian GNU/Linux de base, on crée 3 images différentielles qui vont correspondre aux instances de routeurs. Voici un exemple de suite de commandes pour la mise en place des images.

```
:~$ mkdir -p ~/vm/ospf
:~$ cd vm
~/vm$ wget http://www.stri/vm/vm0-debian-testing-i386-base.qed
<snipped/>
~/vm$ cd ospf
~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r1.qed
~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r2.qed
~/vm/ospf$ ../scripts/diff-img.sh ../vm0-debian-testing-i386-base.qed r3.qed
```

Le code du [script de création de fichier image différentiel](#) est fourni dans le guide [Virtualisation système et enseignement](#).

Ensuite, on crée un nouveau script shell de lancement des instances de «routeurs» dans lequel on fixe les paramètres d'initialisation de ces mêmes «routeurs».



Avertissement

Ce script ne doit être lancé qu'après l'initialisation du commutateur virtuel pour que le brassage des routeurs sur les ports du commutateur puisse se faire correctement.

```
$ cd ~/vm/ospf
$ cat ospf-startup.sh
#!/bin/bash

../scripts/ovs-startup.sh r1.qed 512 1

../scripts/ovs-startup.sh r2.qed 512 2

../scripts/ovs-startup.sh r3.qed 512 3
```

9.3. Choix des identifiants de routeur OSPF

Pour toute instance du protocole de routage OSPF, le choix de l'identifiant se fait dans l'ordre suivant :

1. Adresse IPv4 définie à l'aide de l'instruction **router-id**.
2. Adresse IPv4 la plus élevée parmi toutes les interfaces de boucle locale

3. Adresse IPv4 la plus élevée parmi toutes les interfaces actives

La liste de identifiants utilisés pour les trois routeurs qui servent à la rédaction des réponses types est la suivante.

R1	0.0.0.1
R2	0.0.0.2
R3	0.0.0.3

9.4. Table de routage du système hôte

Pour que les réseaux de l'aire OSPF puissent communiquer avec le système hôte et l'Internet, il est nécessaire de compléter la table de routage du système hôte. Dans ce contexte le système hôte joue le rôle d'un routeur central et la technique usuelle employée pour répondre au besoin d'interconnexion consiste à implanter une route statique avec un «super réseau» qui rassemble tous les réseaux de l'aire en une seule entrée.

L'aire OSPF étudiée contient quatre réseaux :

- 10.1.3.0/29 - réseau fictif ajouté sur R3,
- 10.1.12.0/26 - réseau correspondant au lien entre R1 et R2,
- 10.1.13.0/26 - réseau correspondant au lien entre R1 et R3,
- 10.1.23.0/26 - réseau correspondant au lien entre R2 et R3,

L'utilisation de l'outil ipcalc permet de vérifier qu'un masque de 19 bits permet d'englober ces quatre réseaux en une seule entrée.

```
$ ipcalc 10.1.0.0/19
Address: 10.1.0.0          00001010.00000001.000 00000.00000000
Netmask: 255.255.224.0 = 19 11111111.11111111.111 00000.00000000
Wildcard: 0.0.31.255      00000000.00000000.000 11111.11111111
=>
Network: 10.1.0.0/19      00001010.00000001.000 00000.00000000
HostMin: 10.1.0.1         00001010.00000001.000 00000.00000001
HostMax: 10.1.31.254      00001010.00000001.000 11111.11111110
Broadcast: 10.1.31.255    00001010.00000001.000 11111.11111111
Hosts/Net: 8190           Class A, Private Internet
```

On complète donc la table de routage du système hôte avec l'instruction suivante :

```
$ sudo ip route add 10.1.0.0/19 via 192.0.2.10
```

Une fois cette nouvelle entrée de la table de routage du système hôte en place, on peut valider l'accessibilité des réseaux de l'aire en testant le service Web factice implanté sur le routeur R3 depuis le système hôte.

```
$ wget -O /dev/null http://10.1.3.3
--2015-10-30 15:35:37-- http://10.1.3.3/
Connexion à 10.1.3.3:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 11104 (11K) [text/html]
Sauvegarde en : « /dev/null »
```

Le chargement de la page depuis le serveur web factice fonctionne bien !

10. Documents de référence

Architecture réseau des travaux pratiques

Le support [Architecture réseau des travaux pratiques](#) présente la topologie physique des salles de travaux pratiques avec la [Disposition des équipements dans l'armoire de brassage](#) ainsi que les configurations par défaut des équipements. On y trouve aussi le plan d'adressage IP utilisé avec les autres supports de travaux pratiques, le plan de numérotations des VLANs et les affectations des groupes de ports des commutateurs.

Configuration d'une interface réseau

Le support [Configuration d'une interface de réseau local](#) présente les opérations de configuration d'une interface réseau et propose quelques manipulations sur les protocoles de la pile TCP/IP

Initiation au routage, 1ère partie

L'article [Initiation au routage, 1ère partie](#) introduit l'utilisation de quagga et de son premier démon baptisé zebra. Ce démon permet de mettre en place un routage statique associé à la table de routage définie dans la configuration du système.

Initiation au routage, 3ème partie

L'article [Initiation au routage, 3ème partie](#) introduit l'utilisation du protocole OSPF sur plusieurs aires. Ce n'est pas l'objectif de ce support qui se limite au routage dynamique dans un système autonome unique ; donc une aire unique.

Introduction au routage inter-VLAN

Le support [Routage Inter-VLAN](#) introduit le principe du routage inter-VLAN ainsi que ses conditions d'utilisation. C'est aussi un support de travaux pratiques dans lequel on n'utilise que du routage statique et de la traduction d'adresses sources (S-NAT) pour acheminer le trafic utilisateur entre les différents réseaux.

Open Shortest Path First (OSPF)

La page consacrée au protocole sur le site Cisco™ : [Open Shortest Path First \(OSPF\)](#) regroupe des ressources importantes sur la conception d'architecture réseau utilisant ce protocole.

On peut aussi citer les supports de formation Cisco Networking Academy qui sont d'une grande qualité sur l'initiation à l'utilisation des protocoles de routage. Malheureusement, ce ne sont pas des documents libres d'utilisation.

Virtualisation système et enseignement

Le support [Virtualisation système et enseignement](#) présente la solution de virtualisation intégrée au noyau Linux : KVM. Associée au commutateur [Open vSwitch](#), cette solution permet de construire des maquettes de travaux pratiques très complètes en offrant de nombreuses fonctions réseau «réelles» dont une table [Content-addressable memory](#).