Les VLANs

Dans ce premier sous chapitre, nous allons voir ce que c'est qu'un VLAN, comment il est possible de configurer un SWITCH pour qu'il gère plusieurs VLANS :

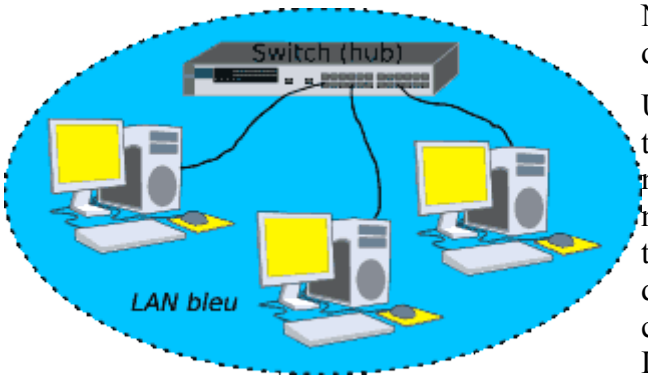
- de façon statique (niveau 1), façon de faire ne nécessitant rien d'autre qu'un SWITCH administrable et ne fait pas appel à un système d'authentification, mais qui ne répondra pas à notre cahier des charges initial,
- de façon dynamique à partir des adresses MAC des clients (niveau 2), il faudra ici non seulement un SWITCH administrable sachant dialoguer avec un serveur RADIUS pour la consultation des adresses MAC autorisées, mais aussi la mise en place de ce serveur RADIUS (que nous verrons dans le sous chapitre suivant).

Plan du chapitre

Théorie

Les bases

Un LAN



Nous sommes ici, c'est sous-entendu tout au long de cet exposé, sur un réseau Ethernet.

Un LAN est un réseau local dans lequel toutes les trames Ethernet sont visibles depuis tous les noeuds si le LAN est construit avec un HUB. Si nous avons affaire à un SWITCH, seules les trames de diffusions (broadcast) seront visibles depuis tous les noeuds, le SWITCH agissant comme un pont Ethernet entre chaque noeud du LAN.

Aujourd'hui, les HUBS ont quasiment disparu des catalogues des constructeurs. Compte tenu de l'usage grandissant des réseaux, il est clair que le HUB ne peut être considéré comme une bonne solution que sur les touts petits réseaux.

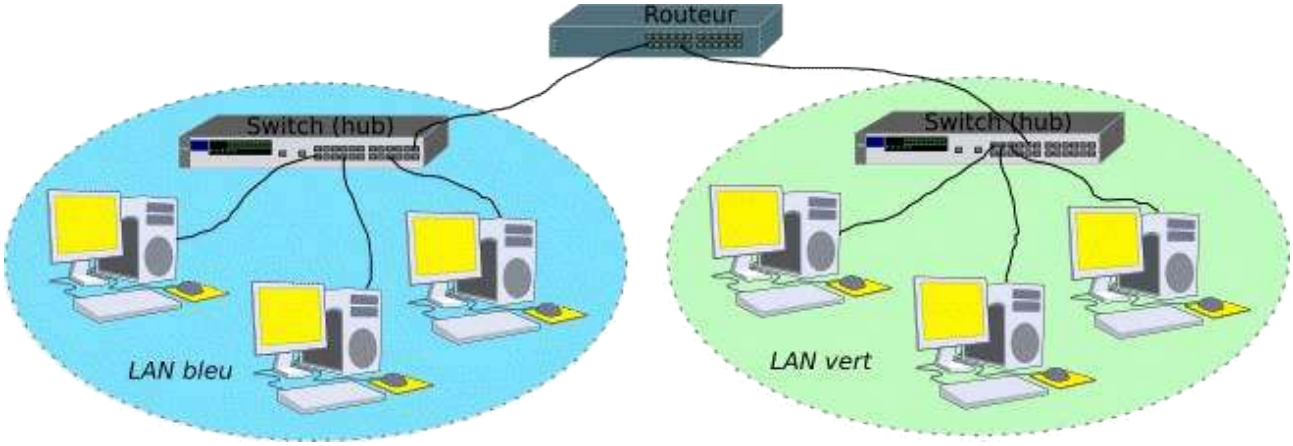
Au dessus de tout ceci, nous construirons un réseau IP, mais c'est à la limite assez peu notre problème ici. Que le réseau de niveau 3 soit IP, NetBEUI ou AppleTalk n'a aucune importance. Bien entendu, dans la suite, nous n'utiliserons qu'IP.

Ce qu'il est fondamental de comprendre, c'est que nous raisonnons au niveau Ethernet, que nous ne parlons que d'adresses MAC.

Un SWITCH, c'est le composant que nous utiliserons par la suite, à l'exclusion des HUBs, est capable d'apprendre et de retenir la ou les adresses MAC qui se présentent sur chacun de ses ports. Hormis les trames de diffusion qui seront systématiquement répercutées sur tous les ports, le SWITCH ne laissera communiquer entre eux que les ports concernés par un dialogue entre deux noeuds. C'est sa fonction principale de pont Ethernet.

Deux LANS (ou plus)

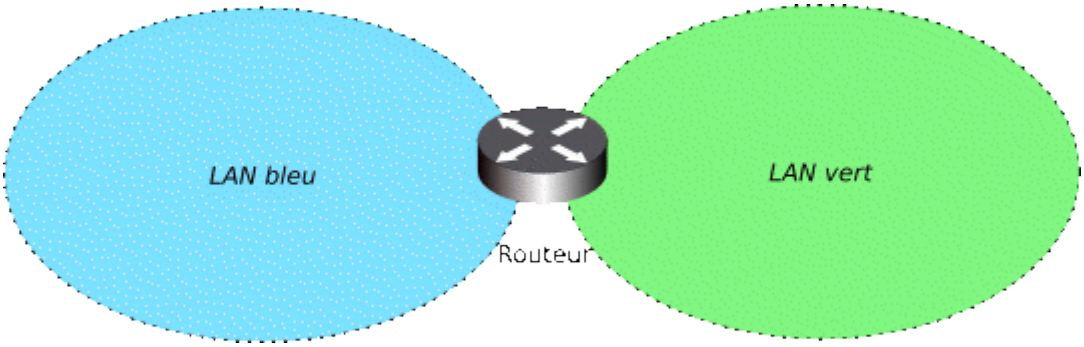
Lorsque nous avons deux LANs et que nous souhaitons les interconnecter, tout en conservant dans chaque LAN les mêmes propriétés au niveau Ethernet, nous devons faire appel à la couche 3 (IP) pour assurer l'interconnexion. Il nous faut donc un routeur.



Le routeur agit au niveau 3 (IP). Ce qu'il est absolument fondamental de comprendre, c'est qu'au niveau Ethernet, le LAN bleu ignore complètement l'existence du LAN vert, et réciproquement. Les trames Ethernet, qu'elles soient de la diffusion ou non, n'iront jamais dans l'autre LAN. Il y a isolation complète des deux LANs au niveau Ethernet et la présence du routeur n'y change rien. Les trames Ethernet qui transportent des données depuis le LAN vert dans le LAN bleu ne seront rien d'autre que des trames Ethernet issues du routeur côté LAN bleu (et réciproquement). Comme dans un roman de science-fiction de bonne facture, les mondes Ethernet bleu et vert sont des mondes parallèles, avec de temps en temps, une porte mystérieuse qui s'ouvre pour laisser passer des choses d'un monde à l'autre, mais en leur faisant perdre la mémoire de leur origine réelle (nous sommes au niveau 2, n'oublions pas).

Il faut monter au niveau de conscience supérieur (niveau IP), pour commencer à démythifier le fonctionnement de ces portes.

Mais à cette hauteur, les détails lointains s'estompent. Ce qu'il y a exactement dans chaque LAN au niveau Ethernet importe finalement assez peu. A première vue, le panorama serait plutôt le suivant :



Parce que, finalement, lors de l'interconnexion de réseaux, peu importe ce qu'il y a dans chaque réseau, ce sont les routes qui importent le plus.

Entendons par là que les équipements utilisés pour construire chaque LAN, qu'il s'agisse de HUBs ou de SWITCHs ou d'un mélange des deux n'a aucune importance.

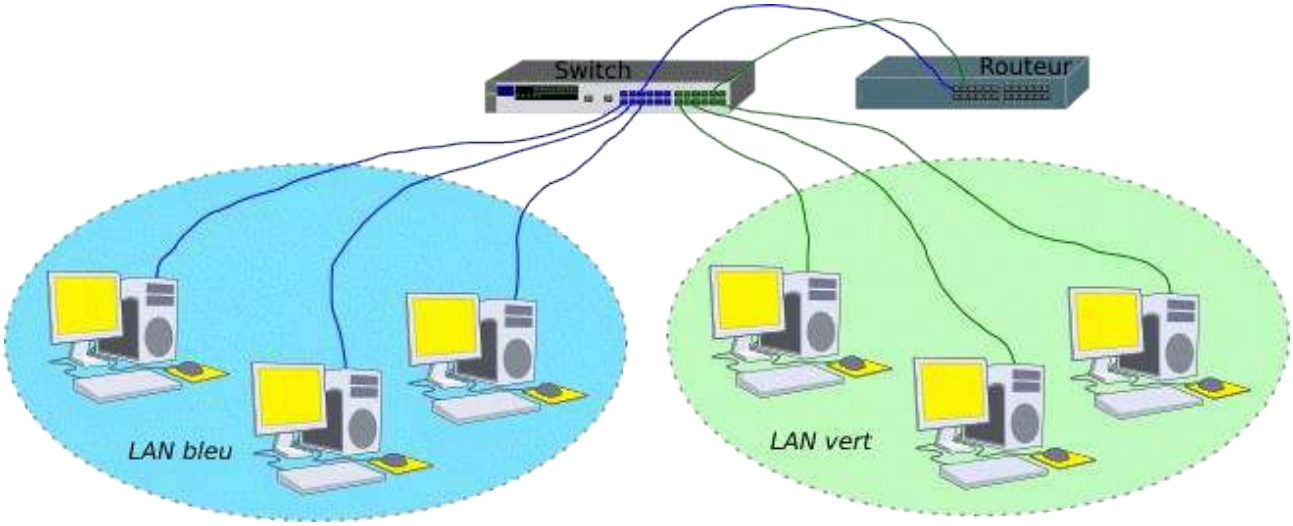
Où intervient le virtuel

Jusqu'ici, un SWITCH appartenait à un et un seul LAN. L'idée de base est de pouvoir assigner certains ports du SWITCH à un LAN, certains autres ports à un autre LAN etc :

Sur un même SWITCH physique, nous allons pouvoir créer plusieurs LANS et assigner certains de ses ports aux divers LANs créés. Ici, nous avons un LAN bleu et un LAN vert. Laissons pour le moment les ports gris de côté.

Tout va (presque) se passer comme si l'on avait découpé notre SWITCH en deux morceaux (sans pour autant le détruire).

Dans une première approche, notre maquette deviendrait ceci :



Le SWITCH a été virtuellement coupé en deux. Les deux VLANs sont complètement étanches au niveau Ethernet (Un SWITCH est en principe un outil qui ne va pas au delà du niveau 2). Pour interconnecter ces deux LANs, un routeur est toujours nécessaire.

Pourquoi faire ?

Il y a bien entendu quelques avantages à pratiquer de la sorte. Nous pouvons au moins en citer deux :

- optimisation du matériel. En effet, c'est évident sur l'illustration, nous n'avons plus besoin que d'un seul SWITCH, là où il nous en fallait deux au départ,
- passer un poste de travail d'un LAN à l'autre devrait pouvoir se faire de façon "soft". Plutôt que de débrancher puis de rebrancher ailleurs le lien du poste, nous pourrons le faire par l'outil de configuration du SWITCH.

Voilà pour le principe de base. Vous devinez que si je prends la peine de rédiger un chapitre sur les VLANs, c'est qu'il y a d'autres choses encore derrière ce concept. Jusqu'ici, c'est assez simple. Les choses vont maintenant se compliquer progressivement pour arriver à des solutions qui peuvent vite devenir un casse tête. Il faudra alors résister à la tentation de réaliser des "usines à gaz" là où ce

n'est pas nécessaire. Les solutions les plus simples, pourvu qu'elles répondent au cahier des charges, sont toujours les meilleures.

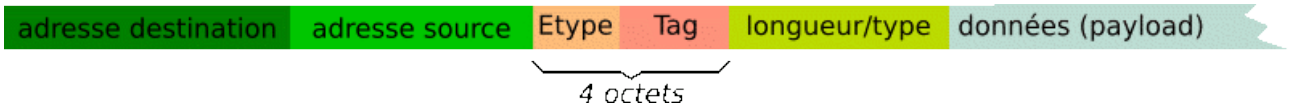
802.1q (ou l'art du tag)

Ici, l'idée serait d'arriver à ce que certains ports du swith puissent être assignés à plusieurs VLANs, ça fera économiser du câble (et aussi des ports sur le SWITCH).

Le principe consiste à ajouter dans l'en-tête de la trame Ethernet un marqueur qui va identifier le VLAN. Il existe quelques solutions propriétaires pour réaliser ceci, mais le système s'est avéré tellement intéressant qu'une norme a été définie, il s'agit de la norme 802.1q.

Alors qu'une trame Ethernet "normale" est constituée comme ceci :

Une trame modifiée par la norme 802.1q se trouve allongée de 4 octets :

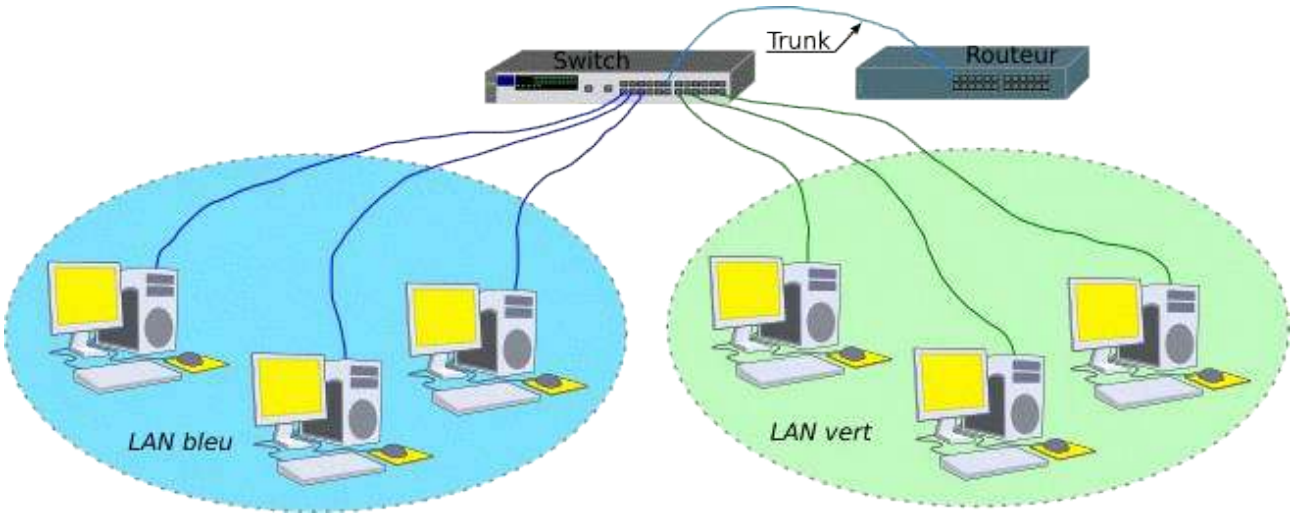


Il n'est peut-être pas nécessaire de détailler le contenu de ces deux nouveaux champs. Pour l'instant, retenons que le VID (Identifiant du VLAN) est codé sur 12 bits, ce qui laisse une latitude confortable.

Il est aussi nécessaire de rappeler qu'une trame Ethernet ne doit pas dépasser 1518 octets et que donc, quatre octets de plus dans l'en-tête risquent d'aboutir à une fragmentation des trames, ce qui n'est jamais bien bon. Si l'on doit avoir recours à des VLANS "taggués", il sera sans doute nécessaire de prévoir ce détail.

Au final, notre SWITCH a donc la possibilité d'ajouter ces marqueurs aux trames Ethernet. Si c'est le cas, il sera alors possible théoriquement d'assigner un même port à 2^{12} VLANS différents. Grâce au VID de chaque VLAN, les données seront acheminées correctement.

Si nous appliquons cette technique à notre maquette, nous obtenons ceci :



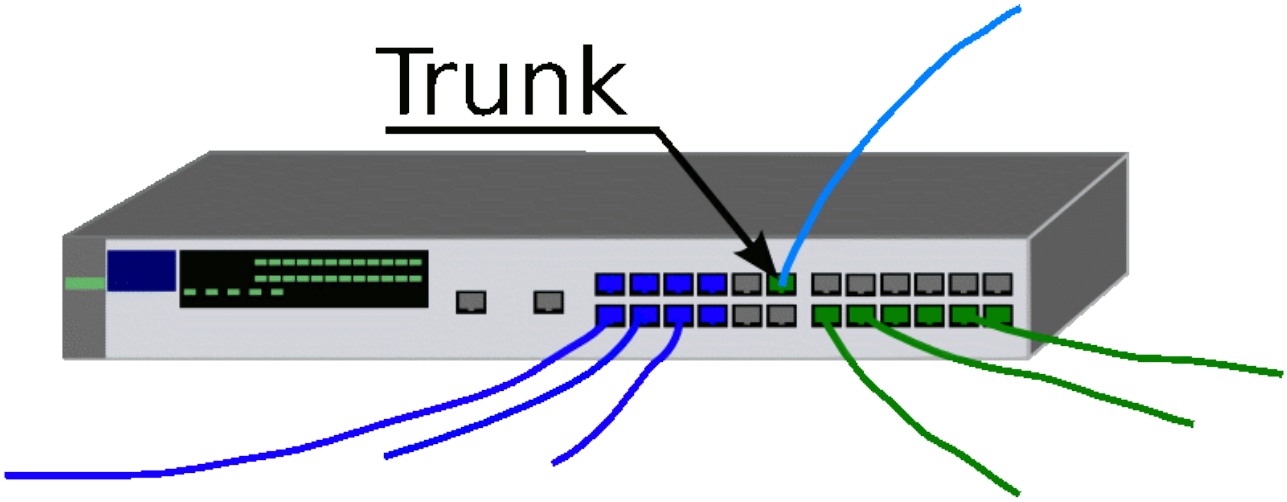
Que les esprits sensibles gardent leur sérénité. Il n'y a effectivement qu'un seul câble qui relie l'unique swith au routeur, et pourtant, nous allons effectivement router les données entre les deux

LANs. Il y a tout de même une condition à respecter : le routeur doit être "802.1q compliant", c'est-à-dire qu'il doit savoir lire les tags que le SWITCH a posé sur au moins l'un des deux VLANs.

Explications

Au niveau 2

Si nous faisons un gros plan sur le SWITCH, nous observons ceci :



Le port marqué "trunk" appartient à la fois aux deux VLANs bleu et vert. Sur ce port, il faut bien sûr qu'au moins l'un des deux VLANs soit "taggué".

Sur le câble relié à ce port, il circulera donc à la fois les trames du VLAN bleu et celles du VLAN vert. Il n'y aura pas de problèmes tant qu'à chaque bout du câble, l'interface Ethernet sera capable de trier les trames en fonction du tag. Ceci impose donc naturellement que le routeur soit compatible avec la norme 802.1q, c'est-à-dire que son interface soit capable d'exploiter ces tags.

Sous Linux, c'est tout à fait possible, il existe sur les distributions modernes un module spécialisé : le module 8021q (testé sur Debian Sarge et Etch).

Au niveau 3

Le SWITCH n'a (en principe) rien à faire du niveau 3. Chacun des VLANs se trouvera avec un plan d'adressage IP qui lui est propre, mais le SWITCH n'est pas concerné, si ce n'est par le fait que pour l'administrer, il faudra bien y accéder par IP. Pour ce faire, le SWITCH disposera d'une adresse IP sur au moins l'un des VLANs, et la machine d'administration devra pouvoir accéder à ce VLAN. Il y aura quelques problèmes de sécurité à envisager à ce niveau, mais nous n'y sommes pas encore.

Au niveau du routeur, en revanche, il faudra que l'interface Ethernet physique puisse présenter autant d'interfaces virtuelles qu'il y a de VLANs sur le "trunk", chacune avec une adresse IP dans le VLAN concerné.

Attribution d'un port à un VLAN

Il y a plusieurs façons de s'y prendre. Vous trouverez sans doute de nombreuses pages qui parlent

de ce sujet, en vous parlant des VLANs de niveau 1, 2, voire 3. Nous allons essayer de voir ceci de façon plus pragmatique.

Attribution statique (niveau 1)

C'est la méthode la plus simple et aussi la moins souple, qui consiste, comme nous l'avons sous entendu jusqu'ici, à attribuer un port du SWITCH à un VLAN donné, en configurant statiquement le SWITCH. Nous n'avons besoin de rien d'autre que d'un SWITCH administrable.

Attribution dynamique (niveaux > 1)

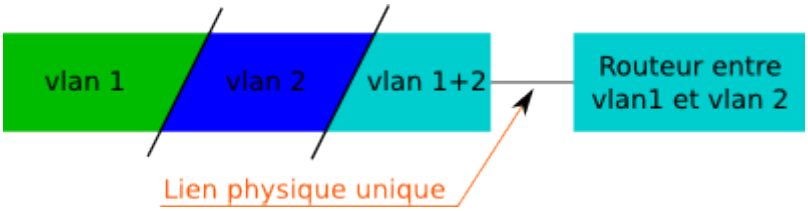
Ici, nous ferons appel à 802.1x et à un procédé d'authentification. Dans ce qui suit, nous disposerons d'un SWITCH capable d'envoyer à un serveur d'authentification (typiquement RADIUS) l'adresse MAC de la station connectée à un port, en guise de "login/password". Si l'adresse MAC est connue (authentification réussie), le serveur pourra envoyer au SWITCH le numéro de VLAN attaché à la station. Attention, tous les SWITCHs 802.1q ne savent pas forcément réaliser cette opération.

Cette méthode est plus souple, puisqu'une station donnée pourra se connecter sur n'importe quel port, elle se retrouvera toujours sur le VLAN qui lui convient.

Il est possible d'utiliser cette méthode avec autre chose que l'adresse MAC (login/password, certificat x509, smartcard...), il faudra alors mettre en oeuvre un "supplicant" sur la station. Nous ne verrons pas cette possibilité dans l'étude du réseau filaire, nous nous contenterons des adresses MAC.

Manip VLANs

Oui, mais au niveau 3 ?



Dans notre exemple, le SWITCH est configuré pour supporter deux VLANS, respectivement d'ID 1 et 2. Les ports verts appartiennent au VLAN d'ID 1 et les ports bleus au VLAN d'ID 2. Aucun de ces ports n'a besoin d'être "taggué" puisqu'ils n'appartiennent qu'à un seul VLAN.

En revanche, sur les ports qui vont véhiculer les trames des deux VLANs, au moins l'un des deux devra être "taggué" au passage de ces ports. Encore une fois, c'est l'interface d'administration du SWITCH qui permettra de réaliser cette configuration. Disons pour fixer les idées que le VLAN vert, d'ID 1 ne sera pas marqué et que le VLAN bleu, d'ID 2 le sera, sur les ports du "trunk" VLAN 1 + VLAN 2.

Pratiquement, admettons que le VLAN 1 supporte un réseau IP 192.168.10.0/24 et que le VLAN 2 soit adressé en 192.168.11.0/24.

Sur le routeur (une machine Debian Etch), nous devons configurer l'unique interface Ethernet physique de manière à ce qu'elle présente deux interfaces IP, chacune pour un VLAN. Il nous faut d'abord installer le paquetage "vlan".

```
~# apt-get install vlan
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
....
```

Nous devons également monter le module noyau qui permet de gérer les tags :

```
~# modprobe 8021q
```

Nous venons d'installer les outils nécessaires à la gestion des VLANs (le paquetage vlan et le montage du module 8021q).

Nous devons maintenant créer une interface réseau virtuelle, qui sera chargée de traiter le VLAN bleu d'ID 2 (celui qui est "taggué"). La commande "vconfig" va le permettre :

```
~# vconfig add eth0 2
Added VLAN with VID == 2 to IF -:eth0:-
```

Vérifions avec la commande ifconfig :

```
~# ifconfig -a
eth0      Lien encap:Ethernet  HWaddr 00:20:18:54:99:F9
          inet adr:192.168.10.1  Bcast:192.168.10.255  Masque:255.255.255.0
          adr inet6: fe80::220:18ff:fe54:99f9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1631 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1233 errors:0 dropped:0 overruns:0 carrier:0
          collisions:2 lg file transmission:1000
          RX bytes:993172 (969.8 KiB)  TX bytes:118423 (115.6 KiB)
          Interruption:11 Adresse de base:0xa800

eth0.2    Lien encap:Ethernet  HWaddr 00:20:18:54:99:F9
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:332 (332.0 b)  TX bytes:0 (0.0 b)
...
```

L'argument "-a" est nécessaire pour visualiser les interfaces "down". eth0.2 existe maintenant, mais n'est pas encore montée.

Nous disposons maintenant sur notre Debian d'un unique adaptateur Ethernet (eth0) qui pourra recevoir nativement le VLAN bleu, puisqu'il n'a pas de tag, et un adaptateur virtuel (eth0.2) qui traitera les trames Ethernet du VLAN vert, d'ID 2. Reste à fixer une adresse IP à eth0.2 et à la monter :

```
~# ip addr add 192.168.11.1/24 broadcast 192.168.11.255 dev eth0.2
~# ifconfig eth0.2 up


~# ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:20:18:54:99:F9
          inet adr:192.168.10.1  Bcast:192.168.10.255  Masque:255.255.255.0
          adr inet6: fe80::220:18ff:fe54:99f9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1631 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1233 errors:0 dropped:0 overruns:0 carrier:0
          collisions:2 lg file transmission:1000
          RX bytes:993172 (969.8 KiB)  TX bytes:118423 (115.6 KiB)
          Interruption:11 Adresse de base:0xa800

eth0.2    Lien encap:Ethernet  HWaddr 00:20:18:54:99:F9
          inet adr:192.168.11.1  Bcast:192.168.11.255  Masque:255.255.255.0
          adr inet6: fe80::220:18ff:fe54:99f9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:0 (0.0 b)  TX bytes:344 (344.0 b)
```

La dernière étape consistera à :

- vérifier que le kernel autorise le routage,
- écrire éventuellement des règles iptables pour le filtrage de paquets, en fonction des besoins.

Cette méthode offre l'avantage de pouvoir réaliser facilement des manipulations pour la mise en oeuvre de la solution, mais offre en revanche l'inconvénient d'être totalement manuelle. Il est bien sûr possible d'arranger ça avec un script bien placé, mais il existe sur Debian (et probablement aussi sur d'autres distributions) une autre solution, qui passe par le fichier de configuration des interfaces (/etc/network/interfaces), dont voici un exemple :

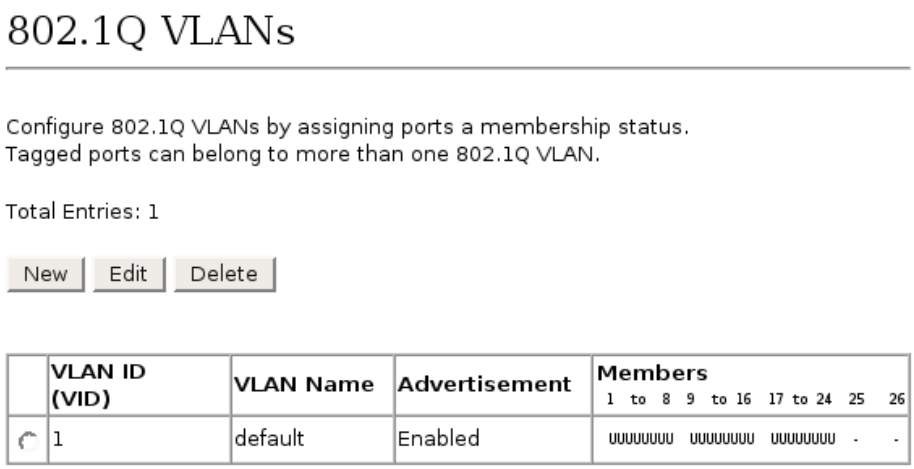
```
...
auto eth0
iface eth0 inet static
    address 192.168.10.1
    netmask 255.255.255.0
    network 192.168.10.0
    broadcast 192.168.10.255

auto vlan2
iface vlan2 inet static
        address 192.168.11.1
        netmask 255.255.255.0
        broadcast 192.168.11.255
        vlan_raw_device eth0
...
```

Nous aurons alors une interface vlan2 en lieu et place de eth0.2, mais qui remplira exactement le même rôle. Pour éviter d'éventuels problèmes de montage du module 8021q, autant l'ajouter dans le fichier /etc/modules.

Mise en oeuvre

La manipulation qui suit est faite avec un SWITCH de type D-Link DES-3326s. Lorsque nous configurons notre SWITCH, nous créons plusieurs VLANs. Chacun de ces VLANs dispose d'un VID.

Ensuite, nous attribuons chaque port à un VLAN particulier. Encore une fois, un port (ou plus) peut appartenir à plusieurs VLANs. Voyons ceci sur l'interface d'administration du D-Link DES-3326s :



Nous constatons qu'ici, il n'existe pour l'instant qu'un seul VLAN, de VID 1. Tous les ports de 1 à 24 lui sont assignés de façon "untagged".

802.1Q VLANs

Configure 802.1Q VLANs by assigning ports a membership status.
Tagged ports can belong to more than one 802.1Q VLAN.

Total Entries: 2

New | Edit | Delete

	VLAN ID (VID)	VLAN Name	Advertisement	Members 1 to 8	9 to 16	17 to 24	25	26
○	1	default	Enabled	UUUUUUUU	UUUUUUUU	UUUUUUUU	-	-
○	2	Demo	Enabled	--------	--------	--------	-	-

Nous allons commencer par créer un second VLAN de démonstration, puis nous nous occuperons des ports laissés libres (25 et 26).

Comme nous le voyons, le second VLAN est créé, mais pour l'instant, aucun port ne lui est assigné.

802.1Q VLANs

Configure 802.1Q VLANs by assigning ports a membership status.
Tagged ports can belong to more than one 802.1Q VLAN.

Total Entries: 2

New | Edit | Delete

	VLAN ID (VID)	VLAN Name	Advertisement	Members 1 to 8	9 to 16	17 to 24	25	26
○	1	default	Enabled	UUUUUUUU	UUUUUUUU	UUUUUUUU	-	U
○	2	Demo	Enabled	--------	--------	--------	U	T

Enfin, nous assignons le port 25 au VLAN 2 de façon "untagged", puis le port 26 aux deux VLANs. Ici, il faudra utiliser les tags :

- "untagged" dur le VLAN 1,
- "tagged" sur le VLAN 2

Au final, si nous relions le port 26 à notre Debian Sarge sur son interface physique eth0, nous aurons la possibilité de router les paquets entre les VLANs 1 et 2.

VLANs niveau 2

Position du problème

Le principe des VLANs étant compris, la dernière étape va consister à mettre en oeuvre une commutation automatique des ports du SWITCH sur l'un ou l'autre VLAN, suivant que la machine qui s'y connecte sera authentifiée ou non.

Conformément au cahier des charges, nous utilisons simplement l'adresse MAC de la machine, ce qui évitera d'avoir à installer sur chaque client un système d'authentification plus sophistiqué (un certificat, par exemple, comme nous le verrons avec WPA2-TLS). Cette méthode n'est pas parfaite, loin de là, dans la mesure où une adresse MAC peut être falsifiée, mais elle a le mérite d'être simple à mettre en oeuvre.

Il nous faudra tout de même disposer de SWITCHs capables d'interroger un serveur RADIUS, en lui envoyant l'adresse MAC du client en guise de nom d'utilisateur et de mot de passe. Nous utilisons ici un SWITCH HP Procurve 2650.

Remarques à propos du ProCurve 2650

Lorsqu'il sort de sa boîte, ce SWITCH est configuré avec un seul VLAN, nommé "DEFAULT_VLAN" (et qui est aussi le "PRIMARY_VLAN"). Tous les ports du SWITCH sont affectés à ce VLAN, si bien que sans aucune configuration particulière, ce SWITCH fonctionnera comme un SWITCH de base.

Pour le configurer, plusieurs solutions sont proposées, à commencer par une liaison série RS232 (gardez au moins un vieux PC), qui est initialement le seul moyen possible pour accéder à la configuration (*In the factory default configuration, the SWITCH has no IP (Internet Protocol) address and subnet mask, and no passwords. In this state, it can be managed only through a direct console connection*).

Par la suite, nous pourrons accéder au SWITCH par le réseau, via telnet, un mini serveur web embarqué (mais vraiment minimaliste), ou même ssh. En effet les SWITCHs administrables peuvent recevoir une adresse IP pour accéder à l'administration par le réseau. Sur ce modèle de SWITCH, nous pourrons même assigner une adresse IP par VLAN, ce qui n'est absolument pas nécessaire, voire dangereux. Comme notre propos est plutôt de parler des VLANs, nous passerons ces détails sordides.

Nous supposons donc que la configuration de base du SWITCH est faite, et principalement la configuration IP du DEFAULT_VLAN :

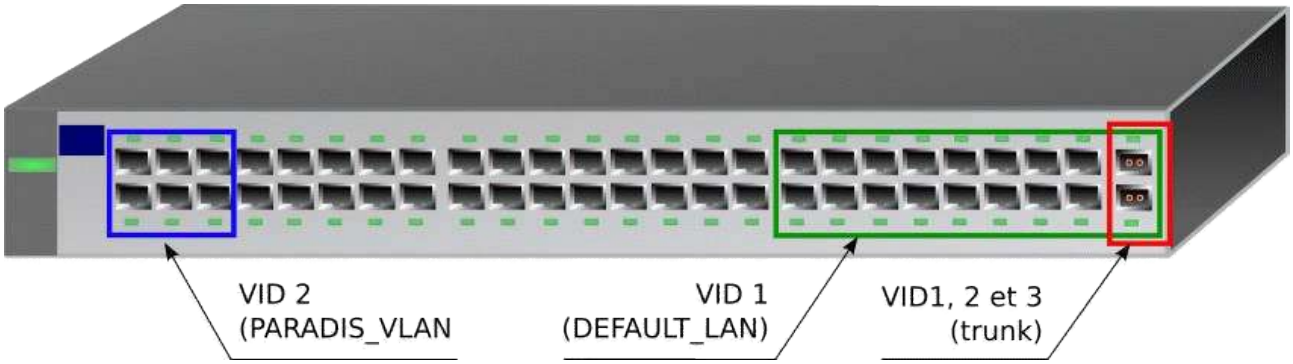
```
Internet (IP) Service

 IP Routing : Disabled

 Default Gateway : 192.168.10.1
 Default TTL     : 64
 Arp Age         : 20

 VLAN         | IP Config  IP Address      Subnet Mask
 ------------ + ---------- --------------- ---------------
 DEFAULT_VLAN | Manual     192.168.10.11   255.255.255.0
```

Donc, nous avons un "DEFAULT_VLAN" qui est aussi le "PRIMARY_VLAN", et qui contient tous les ports du SWITCH. Il y a quelques contraintes à connaître à propos de ces deux VLANs :

- DEFAULT_VLAN ne peut pas être supprimé et a forcément le VID 1, en revanche, rien n'interdit de ne lui assigner aucun port,
- PRIMARY_VLAN est nécessaire à certaines fonctions d'administration que nous n'utiliserons pas forcément ici, comme le pseudo empilage de SWITCHs. Cette fonction de PRIMARY_VLAN peut être assignée à n'importe quel VLAN existant, pas forcément au DEFAULT_VLAN, mais il doit exister. Nous laissons la configuration par défaut.

Nous n'allons conserver que quelques ports, dont les deux ports 1GB/s sur DEFAULT_VLAN, laisser PRIMARY_VLAN dessus, tous les autres ports étant réservés à deux autres VLANs qu'il nous reste à créer :

- PARADIS_VLAN, de VID 2, qui accueillera le LAN des hôtes connus,
- ENFER_VLAN de VID 3, qui accueillera toutes les machines que l'on ne sait pas identifier.



Nous allons restreindre le DEFAULT_VLAN aux ports 33-50 (sans tag). Ce VLAN nous servira à administrer le SWITCH, à accueillir les DNS, DHCP, RADIUS et autres services "administratifs".

Les ports 1 à 6 seront assignés au PARADIS_VLAN de façon statique (sans tag). Nous y placerons par la suite les ressources du réseau à offrir aux stations "connues".

Comme ce SWITCH ne supporte pas d'avoir des ports assignés à aucun VLAN, nous allons mettre les ports 7 à 32 dans le ENFER_VLAN. Nous reviendrons éventuellement sur ce choix plus tard.

Les ports 49 et 50 seront quant à eux assignés aux trois VLANs. Bien entendu, ici, il faudra utiliser les tags. L'un des deux ports servira à connecter le routeur et l'autre sera réservé aux extensions futures (un second SWITCH, par exemple).

Au final, la (magnifique) interface web nous montrera ceci :

VLAN ID	VLAN Name	VLAN Type	Tagged Ports	Untagged Ports	Forbid Ports	Auto	
1	DEFAULT_VLAN (Primary)	STATIC	(STATIC) None (GVRP) None	33-50	None	None	Modify
2	PARADIS_VLAN	STATIC	(STATIC) 49-50 (GVRP) None	1-6	None	None	Modify
3	ENFER_VLAN	STATIC	(STATIC) 49-50 (GVRP) None	7-32	None	None	Modify

Mais qu'importe la beauté lorsque le travail est bien fait, n'est-ce pas ?

Notez que tout le travail fait jusqu'ici a pu l'être par l'entremise de cette interface. Pour la suite, ce sera quelque peu différent. Il nous reste maintenant à expliquer au SWITCH que les ports 7 à 32 doivent être attribués au VLAN 2 ou au VLAN 3, suivant que l'authentification par adresse MAC sera réussie ou non. Pour réaliser cette opération, l'interface web n'est pas exploitable, le menu en mode texte via telnet, ssh ou rs232 non plus. Il nous faut passer par le moyen le plus rustique, la ligne de commande. C'est la documentation qui va venir à notre secours :

- Access Security Guide : SWITCH 2600 Series SWITCH 2600-PWR Series SWITCH 2800 Series
  SWITCH 4100 Series SWITCH 6108
  - Web and MAC Authentication for the Series 2600/2600-PWR and 2800 SWITCHes
    - Configuring MAC Authentication on the SWITCH
      - Configure the SWITCH for MAC-Based Authentication

Voici la procédure. Dans la console :

```
configure
```

(Pour entrer dans le mode configuration)

```
aaa port-access mac-based addr-format multi-colon
aaa port-access mac-based 7-32
```

Vérifiez que vous obtenez un message indiquant que LACP a été désactivé (LACP est un protocole de gestion d'agrégations de liens, qui sort du cadre de cette étude, mais qui est incompatible avec les VLANs de niveau 2).

- La première ligne indique que les adresses MAC doivent être envoyées sous la forme xx:yy:zz:aa:bb:cc. D'autres formats sont possibles, tout dépendra de la façon qui nous est la plus commode pour collecter et enregistrer ces adresses MAC dans notre futur RADIUS,
- la seconde ligne indique que les ports 7 à 32 seront assignés à un VLAN en fonction de l'adresse MAC du client connecté.

```
aaa port-access mac-based 7-32 auth-vid 2
aaa port-access mac-based 7-32 unauth-vid 3
```

- La première ligne indique que les ports 7 à 32 devront être assignés au VLAN de VID 2 si l'authentification est réussie,
- la seconde ligne indique que les ports 7 à 32 devront être assignés au VLAN de VID 3 si l'authentification est ratée.

```
show port-access 7-32 mac-based config
```

Cette ligne permet de vérifier que notre configuration est bien enregistrée :

```
Port Access MAC-Based Configuration

  MAC Address Format : multi-colon

                Client Client Logoff    Re-Auth   Unauth   Auth
  Port  Enabled Limit  Moves  Period    Period    VLAN ID  VLAN ID
  ----- ------- ------ ------ --------- --------- -------- --------
  7     No      1      No     300       0         3        2
  8     No      1      No     300       0         3        2
  9     No      1      No     300       0         3        2
  10    No      1      No     300       0         3        2
  11    No      1      No     300       0         3        2
  12    No      1      No     300       0         3        2
  13    No      1      No     300       0         3        2
  14    No      1      No     300       0         3        2
  15    No      1      No     300       0         3        2
  16    No      1      No     300       0         3        2
  17    No      1      No     300       0         3        2
  18    No      1      No     300       0         3        2
  19    No      1      No     300       0         3        2
  20    No      1      No     300       0         3        2
  21    No      1      No     300       0         3        2
  22    No      1      No     300       0         3        2
  23    No      1      No     300       0         3        2
  24    No      1      No     300       0         3        2
  25    No      1      No     300       0         3        2
  26    No      1      No     300       0         3        2
  27    No      1      No     300       0         3        2
  28    No      1      No     300       0         3        2
  29    No      1      No     300       0         3        2
  30    No      1      No     300       0         3        2
  31    No      1      No     300       0         3        2
  32    No      1      No     300       0         3        2
```

C'est correct. Nous pouvons écrire le tout en mémoire :

```
write memory
```

En ce qui concerne "l'auth-vid", ce paramètre pourra éventuellement être écrasé par une valeur renvoyée, en cas d'authentification réussie, par le serveur RADIUS, ce qui apporte plus de souplesse si l'on doit disposer de plus de deux VLANs suivant les clients.

C'est presque fini, mais pas tout à fait. Notre SWITCH ne devinera pas tout seul qui est notre serveur RADIUS, il faut le lui indiquer.

Notre futur serveur RADIUS aura l'adresse 192.168.10.2. Comme nous le verrons plus loin, il faudra définir une clé de chiffrement partagée entre le serveur et ses "clients". Nous allons choisir quelque chose de difficile à trouver : chutt

```
radius-server host 192.168.10.2 key chutt
```

Vérifions :

```
show radius

Status and Counters - General RADIUS Information

  Deadtime(min) : 0
  Timeout(secs) : 5
  Retransmit Attempts : 3
  Global Encryption Key :

                  Auth  Acct
  Server IP Addr  Port  Port  Encryption Key
  --------------- ----- ----- --------------------------------
  192.168.10.2    1812  1813  chutt
```

C'est ok

```
write memory
```

Il ne reste plus qu'à mettre en place le serveur RADIUS, que nous connecterons sur le DEFAULT_VLAN (que nous avons configuré pour un réseau IP 192.168.10.0/24).