

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

# MyvSwitch

## Présentation du projet

CRI

lundi 03 décembre 2018

# Switch

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Un *switch* (ou *commutateur*) est un équipement fondamental de tout réseau Ethernet moderne.
- Son rôle est de recevoir des *trames* depuis des *ports* et de les acheminer vers leurs destinataires respectifs, via le port qui permet de les joindre : on parle de *commutation de trames*.
- Un switch est caractérisé par deux éléments :
  - 1 Des *files d'attente* dans lesquelles sont placées les trames.
  - 2 Une capacité d'auto-apprentissage de la topologie du réseau permettant d'acheminer les trames vers le bon port de destination.

# Switch (exemple)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion



FIGURE – Un switch HP Aruba 2930F 48G 4SFP+

# Virtual switch

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Un *virtual switch* est un programme simulant le comportement d'un switch physique. Il opère sur des *interfaces virtuelles* qui lui servent de ports.
- Le noyau Linux dispose d'une implémentation d'un switch virtuel, configurable à l'aide de la commande `ip`.
- Des implémentations *userland* existent également, comme *VDE switch*, par exemple.

# Objectifs

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Le but de ce projet est de vous familiariser avec les concepts de base d'un réseau Ethernet et de certaines techniques de virtualisation.
- La compréhension des protocoles et de la façon dont les switchs fonctionnent vous permettront de mieux appréhender ces outils ainsi que la façon dont ils peuvent être utilisés.
- Il n'est pas attendu de vous l'implémentation de toutes les fonctionnalités proposées par le sujet.
  - Mais si vous vous en sentez capable, *be our guests*.

# Ressources

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires  
Attendues  
Autres

Notions utiles

L'API socket  
Le polling  
Outils

Conclusion

- Vous pouvez utiliser les bibliothèques présentes sur le PIE, à l'exception de *libpcap* (vous devez manipuler les sockets et analyser les trames vous-même).
- Une copie des standards IEEE 802.1 pertinents sont à votre disposition sur l'AFS :  
`/afs/cri/resources/doc/802.1/`
- Des membres du CRI assureront des permanences en SM pendant certaines soirées. Ces permanences seront indiquées sur Chronos.
- Un newsgroup `cri.projets` a été mis en place pour que vous puissiez y poser vos questions.

# Évaluation

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires  
Attendues  
Autres

Notions utiles

L'API socket  
Le polling  
Outils

Conclusion

- Votre code sera lu par les membres du CRI. Une soutenance aura lieu pour vous permettre d'expliquer vos choix techniques.
- Il n'est pas indispensable de respecter la *coding style*, même si c'est vivement recommandé.
- La triche sera, comme toujours, très sévèrement sanctionnée.
- Votre note sera fonction des fonctionnalités implémentées, de la pertinence de votre architecture et de la propreté du code rendu.

# Palier 1 - *Unidirectional forwarding*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

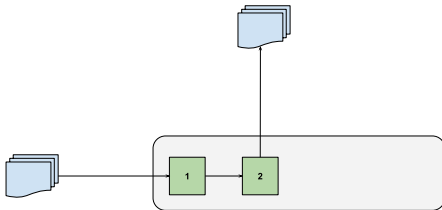
L'API socket

Le polling

Outils

Conclusion

- Votre switch accepte deux noms d'interface en paramètres.
- Toute donnée lue sur la première est immédiatement envoyée sur la seconde.





## Palier 2 - *Frame flooding*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

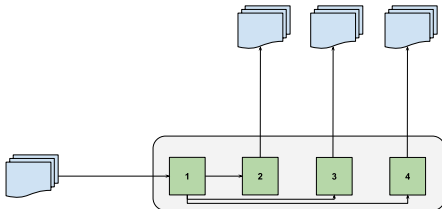
L'API socket

Le polling

Outils

Conclusion

- Votre switch accepte un nombre arbitraire de noms d'interface en paramètres.
- Toute donnée lue sur l'une des interfaces doit être envoyée sur toutes les autres.
  - Ça implique de faire du *polling* afin de pouvoir attendre que des données arrivent sur plusieurs interfaces en même temps.



# Palier 3 - *Accounting*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Votre switch doit afficher sur sa sortie standard, pour chaque trame reçue :
  - le port d'arrivée ;
  - les adresses MAC source et destination ;
  - la valeur du champ EtherType, ainsi que son nom pour certaines valeurs courantes.
- Vous devrez également maintenir des compteurs pour comptabiliser le nombre de trames et d'octets envoyés et reçus sur chaque port.
- Attention à l'ordre des octets lors de l'analyse des trames !

# Palier 4 - *Frame switching*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- À l'issue de ce palier, vous aurez un « vrai switch » : il s'agit en effet d'ajouter une *FIB* (pour *Forwarding Information Base*).
- Toute trame à destination d'une adresse unicast doit être commutée vers le bon port de destination, s'il est connu par la FIB.
  - Dans les autres cas, la trame est envoyée sur tous les ports à l'exception du port d'arrivée.
- À la réception d'une trame, son adresse source est associée, dans la FIB, au port depuis laquelle elle provient.

# Palier 4 - *Frame switching*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

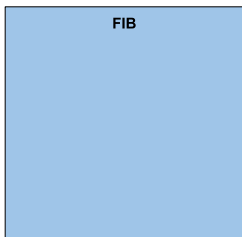
Notions utiles

L'API socket

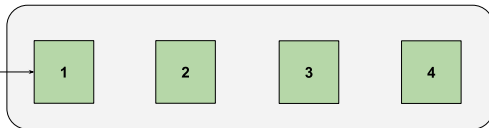
Le polling

Outils

Conclusion



SRC=94:c6:91:a2:c0:da  
DST=14:4f:8a:b4:0d:ed



# Palier 4 - *Frame switching*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

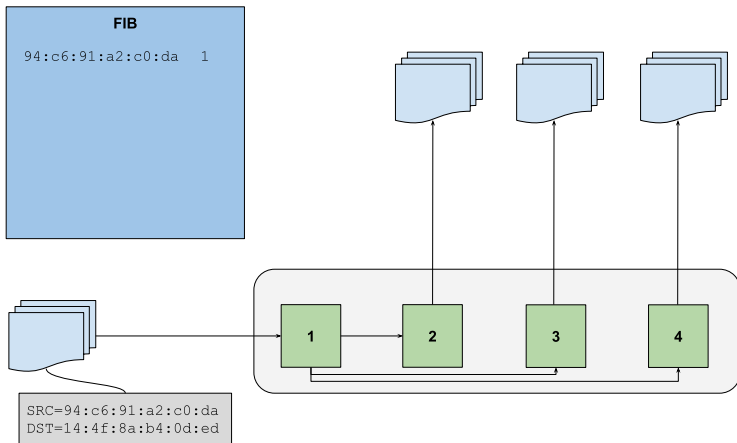
Notions utiles

L'API socket

Le polling

Outils

Conclusion



# Palier 4 - *Frame switching*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

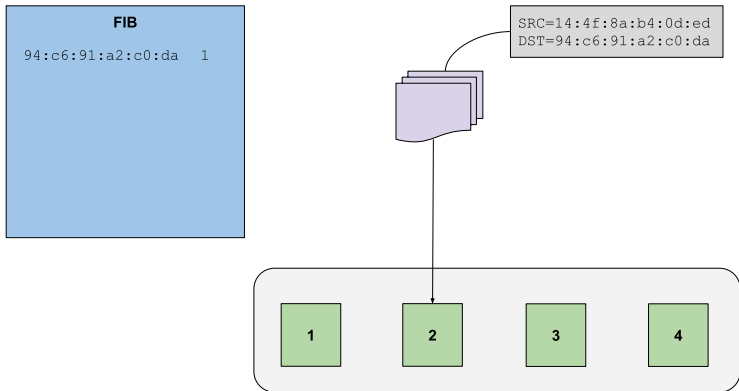
Notions utiles

L'API socket

Le polling

Outils

Conclusion



# Palier 4 - *Frame switching*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

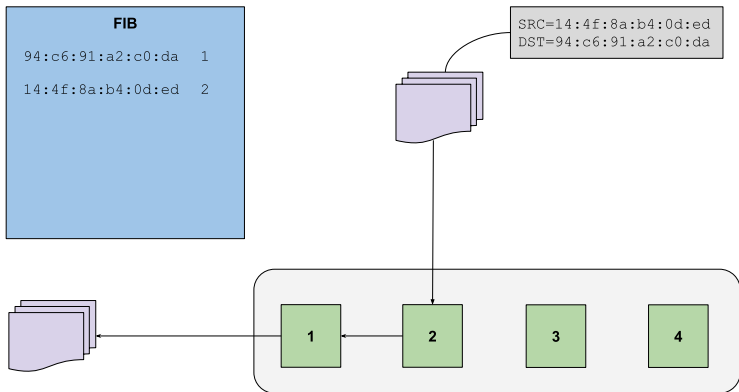
Notions utiles

L'API socket

Le polling

Outils

Conclusion



# Palier 5 - *CLI and configuration*

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Pour ce palier, vous devez faire en sorte qu'une fois lancé, votre switch propose à l'utilisateur une CLI rudimentaire.
- Vous devez aussi disposer d'un moyen de sauvegarder et restaurer la configuration du switch depuis la CLI.
- Il peut être judicieux de ne pas attendre d'arriver à ce palier pour commencer la CLI, puisqu'elle vous permettra de récupérer facilement des informations sur l'état interne de votre switch.
- Vous avez le droit d'utiliser *libreadline*.



# Port mirroring

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

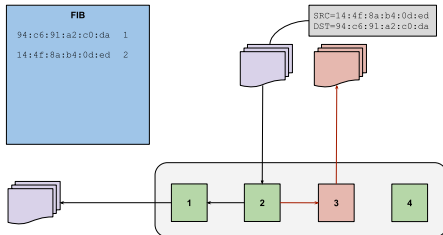
L'API socket

Le polling

Outils

Conclusion

- Le *port mirroring* permet de dupliquer les trames qui transitent par un ou plusieurs ports et de faire sortir les copies par un port désigné.
- C'est un outil de diagnostic très utile pour les administrateurs réseaux.



# VLAN

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Vous n'êtes pas prêts.

- Ça va mal se passer.

« *Les VLAN ? Ils n'y arriveront jamais !* »

- Gaby, à propos des VLAN.

« *Même moi j'y bite que dalle !* »

- Chewie, à propos d'un autre truc  
random qui n'a rien à voir.

# VLAN

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Un *VLAN* (pour *virtual LAN*) est une subdivision artificielle d'un réseau au sein d'un équipement.
- L'idée est de multiplexer les flux de façon à minimiser le câblage et le nombre d'équipements requis.
- Du point de vue des utilisateurs, tout se passait comme si les trames envoyées sur un VLAN passaient par leurs propres câbles et leurs propres switchs : les VLAN permettent donc d'isoler des réseaux.
- Cette isolation se fait via l'injection d'un « tag » spécifique dans l'en-tête Ethernet afin de différencier les flux. Chaque tag contient un identifiant de VLAN.

# VLAN (suite)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- On va donc se retrouver à manipuler deux types de trames, des trames « taguées » et des trames « non-taguées ».
- Chaque port devra donc être configuré pour déterminer à quel VLAN doivent être rattachées les trames « non-taguées » et quels sont les identifiants des VLAN autorisés pour les autres.
- Il y a deux terminologies répandues : *tag/untag* et *access/trunk*. Elles correspondent chacune à une façon différente de conceptualiser les VLAN.

# VLAN (schéma)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

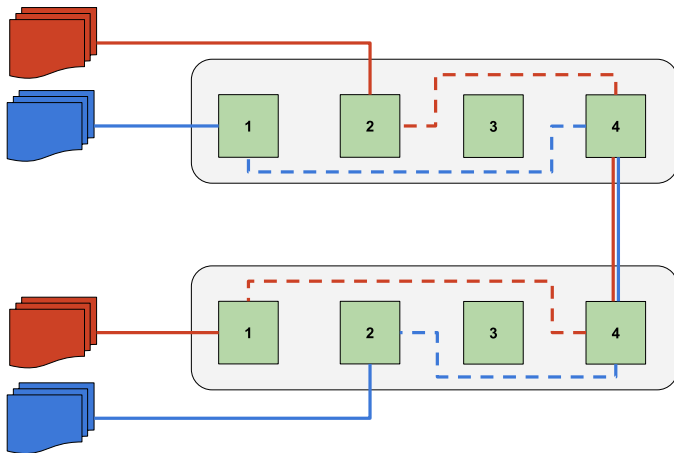
Notions utiles

L'API socket

Le polling

Outils

Conclusion



# Autres fonctionnalités

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- D'autres fonctionnalités sont décrites dans le sujet.
- Elles se répartissent en deux catégories, les *fonctionnalités supplémentaires* et les *fonctionnalités avancées*.
- Les fonctionnalités avancées sont **difficiles**, il n'est pas attendu que vous les fassiez nécessairement.
  - Si vous voulez tenter l'aventure, il est possible d'ignorer complètement les fonctionnalités supplémentaires pour se concentrer sur l'une d'entre elles.

# Introduction

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- L'API socket offre un moyen de communication inter-processus très flexible, tout en respectant la sémantique d'un fichier.
  - Elle permet même de faire communiquer ensemble des processus qui ne sont pas situées sur la même machine. D'où son utilisation pour de la programmation réseau.
- Une caractéristique essentielle d'une socket est la *famille d'adresse* qui est utilisée pour la désigner. Cette famille d'adresse va conditionner le mécanisme ou le protocole qui sera utilisé pour communiquer.

# socket(2)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- La famille `AF_PACKET` est utilisée pour créer des sockets permettant de recevoir les trames transitant par une ou plusieurs interfaces réseau.
- Puisqu'on ne demande pas au noyau d'utiliser un protocole en particulier, on indique `SOCK_RAW` en second paramètre de l'appel système `socket(2)`.

Exemple de création d'une socket de type `AF_PACKET`

```
int sockfd = socket(AF_PACKET, SOCK_RAW, 0);
```



# bind(2)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Afin de commencer à recevoir des trames sur la socket, il faut utiliser `bind(2)` pour indiquer au noyau depuis quelle interface on veut recevoir des données.
- Pour ça il faut trouver *l'index de l'interface*, on peut utiliser `if_nametoindex(3)`.
- On peut alors remplir une structure de type `sockaddr_ll`, comme indiqué dans la manpage `packet(7)`, puis appeler `bind(2)` :

## Utilisation de `bind(2)`

```
bind(sockfd, (struct sockaddr *) &sockaddr, sizeof  
(sockaddr));
```

# Lire et écrire sur une socket

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- On peut lire et écrire sur une socket à l'aide des appels systèmes `recv(2)` et `send(2)`.

## Lire sur une socket

```
ssize_t size;  
char buffer[ETH_FRAME_LEN];  
size = recv(socket, buffer, sizeof (buffer), 0);
```

## Écrire sur une socket

```
send(socket, buffer, size, 0);
```

# Polling

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Comment faire pour attendre des données depuis plusieurs sockets en même temps ?
  - Si on utilise des sockets bloquantes et que l'une d'entre elle ne reçoit jamais rien, on ne pourra jamais lire ce qui arrive sur les autres.
  - Si on utilise des sockets non-bloquantes, on va utiliser du temps CPU pour rien.
- Il nous faudrait un appel système qui endort le processus et qui le réveille dès qu'il y a des données à traiter en lui indiquant où elles sont.
  - Il y en a en réalité plusieurs, plus ou moins anciens, performants et portables, à vous de choisir : `select(2)`, `poll(2)`, `epoll(7)`, ...

# Les wrappers du CRI

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Le CRI met à disposition un ensemble de scripts pour que vous puissiez configurer votre environnement de test et exécuter votre switch dedans.

**Attention !**

- Ces scripts doivent-être exécutés à l'aide de `sudo`.

**Lancer son switch dans un environnement de test**

```
$ sudo ns-init sw host1 host2
$ sudo ns-add-if p1 sw host1
$ sudo ns-add-if p2 sw host2
$ sudo ns-exec sw ./myvswitch p1-1-sw p2-1-sw
```

# wireshark

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

- Vous pouvez utiliser wireshark pour inspecter le contenu des trames qui entrent et sortent de votre switch.
- Si le namespace réseau de votre switch se nomme `sw`, vous pouvez lancer wireshark de cette façon :

Lancer wireshark dans le namespace `sw`

```
$ sudo ns-exec sw HOME=$HOME wireshark-gtk
```

# Inscription (rappel)

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

## Procédure d'inscription

- Envoyer un ticket à `tickets@cri.epita.fr` :
  - 1 *Conforme à la nétiquette.*
  - 2 Ayant pour tag : « [RECRUTEMENT] [CANDIDATURE] ».
  - 3 À partir du lundi 03 décembre à 20h00 et avant le lundi 10 décembre à 12h00.

## Dates importantes

- Sujet disponible le mercredi 05 décembre à 19h00.
- Clôture des inscriptions lundi 10 décembre à 12h00.
- Rendu du projet le dimanche 21 décembre à 23h42.

# Questions

MyvSwitch

CRI

Introduction

Déroulement  
du projet

Fonctionnalités

Obligatoires

Attendues

Autres

Notions utiles

L'API socket

Le polling

Outils

Conclusion

