Linux en Réseau

Introduction à Linux

E. Leblond[1]

[1]EdenWall Technologies SA

Outline

1. Introduction
2. Étude des protocoles et périphériques
   - Protocole de niveau 1 et 2
   - Protocole TCP/IP
3. Fonctionnalités additionnelles
   - Routage avancé et qualité de service
   - Sécurité
   - 2 cas pratiques

E. Leblond
Autobiographie

- Concepteur et développeur principal de NuFW
- Contributeur Netfilter
- Fondateur et directeur technique de EdenWall Technologies
- Né à Calais

EdenWall Technologies

- Éditeur de solutions de sécurités
  - Pare-feu EdenWall
  - Prelude
- Pare-feu EdenWall
  - Basé sur Netfilter
  - Identificatin des flux par NuFW
  - Solution clé en main
- Membre de l'APRIL

Un monde de réseaux.
Trivialités en tout genre.

- Les réseaux sont omni-présents
- à des échelles diverses
  - Téléphones portables
  - CPE (modem ADSL, freebox, pare-feu)
  - PE (routeur backbone)

Linux, une réponse globale
De l'opérateur au particulier.

- Avantage économique direct
- Offre logicielle étendue
- Maitrise et indépendance
- Compétences disponibles sur le marché

Esprit du cours

Heureux qui comme Ulysse . . .

- Découvrir Linux dans le cadre des réseaux
- Comprendre les possibilités de manipulations offertes
- Approcher les contraintes de réalisations

Plan

- Protocoles de niveau 1 et 2.
- TCP/IP et les protocoles courants
- Routage avancé
- Sécurité

- Résumé
  - Encapsulation
  - Imagination

Modèle OSI
L'effet oignon

C'est le schéma classique de décomposition du réseau :

- Une décomposition en couche
- Du physique à l'application
- Basée sur l'encapsulation et fragmentation

Intérêt pratique notamment au niveau du développement.

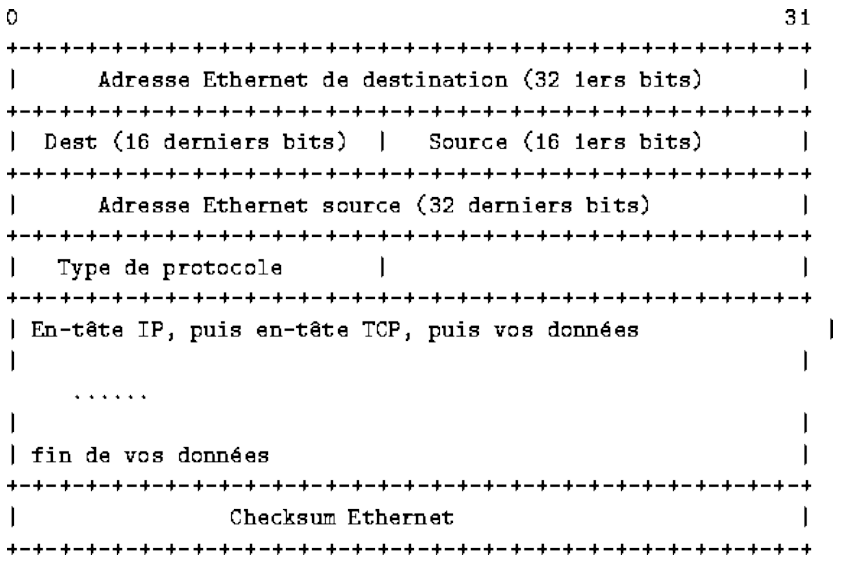
Décomposition du modèle OSI

1. Couche physique : 100base-TX, Wireless
2. Couche de liaison : Ethernet, ATM, TokenRing, Wi-Fi
3. Couche de réseau : ARP, IPv4, IPv6
4. Couche de transport : TCP, UDP, ICMP, SCTP
5. Couche de session : L2TP, PPTP, RPC
6. Couche de présentation : Unicode, MIME, HTML, XML
7. Couche application : SSH, NNTP, DNS, HTTP

Ethernet
Le protocole de tous les jours

- Solution à bas coût et haute performance
  - Connecteur et concentrateur bon marché
  - Gamme très variée
- Atteint des bandes passantes élevées (de 10M à 10G)
- Filaire ou WiFi
- Switch de paquets

Ethernet
Décomposition d'un datagramme ethernet



Ethernet

- La destination d'abord pour l'optimisation
- Le datagramme contient toutes les informations nécessaires au routage
- La taille du datagramme est variable et est limitée par le médium physique:
  - 1500 bytes : la norme et le chiffre à retenir
  - 9000 bytes : jumbo frame

Ethernet
Principe

- Adresse MAC des cartes, identifiant unique
- Communication par adresse MAC
- Mécanisme d'annonce

Recherche correspondance IP<->Adresse Mac :

```
arp who-has 192.168.1.128 tell 192.168.1.2
arp reply 192.168.1.128 is-at 00:0c:f1:5c:47:91
```

Ethernet
ARP

Basé sur la confiance et donc soumis à de nombreuses attaques. Cette couche n'offre aucune sécurité.

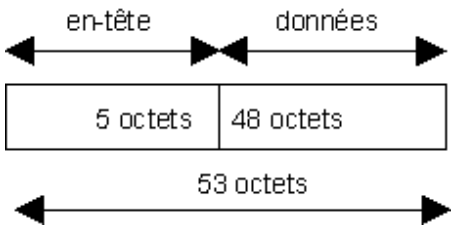
- arp-spoofing
  - Rien n'empêche d'annoncer une adresse illégale
  - Rien n'empêche de changer d'adresse MAC
- arp-poisonning
  - Rien n'empêche d'annoncer n'importe quoi

ATM
Un protocole créé pour les opérateurs (téléphoniques)

Équipement de réseau à grance échelle (taille et distance).

- Relais de cellules (de taille fixe)
- Point à point
- Qualité de service intégrée

ATM
Décomposition d'un datagramme ATM



- L'étiquette contient VP,VC, le mode de communication (AAL5, AAL3) et d'autres informations
- Le routage s'effectue grâce à l'étiquette.

ATM
Qualité de service

- CBR - Constant bit rate
- VBR - Variable bit rate: bande passante moyenne avec burst
- ABR - Available bit rate: minimum garanti
- UBR - Unspecified bit rate: meilleur effort

ATM
Court c'est court

- Taille de cellule fixe: 53 octets
- Découpage massif des paquets IP classiques
- Problème de qualité de service avec TCP

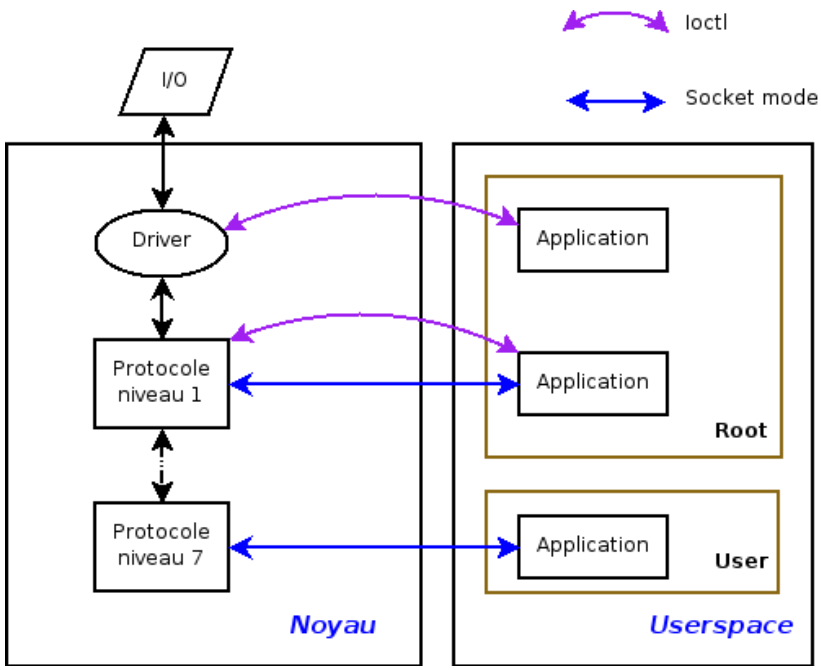
Le cas de l'ADSL en France
l'ADSL non dégroupé France Telecom

- Ethernet (ou WiFi) : chez le particulier
- PPPOE : entre ordi et modem, l'ordi est le début du tunnel PPP
- ATM : modem vers concentrateur
- L2TP : concentrateur vers backbone
- ATM : backbone FT vers terminaison
- Ethernet : Terminaison ATM vers terminateur PPP, fin du tunnel PPP

Les réseaux vus du noyau
Implémentation des réseaux dans linux

- Un support étendu
- Une architecture respectant le découpage OSI
- Fournir des couches d'intéractions avec l'espace utilisateur
  - ioctl : configuration
  - socket : communication

Les réseaux vus du noyau
Schéma fonctionnel



Problématiques
Support du matériel

- Accès aux spécificités :
  - Collaboration avec les fabricants
  - Problème des ports depuis Windows
- Factorisation : Avoir une implémentation de chaque protocole
  - Limiter les efforts
  - Centraliser le code pour les audits

Problématiques
Difficultés globales

- Segmentation
  - Migration entre les différentes couches
  - Réassemblage
- Complexité
  - Complexité élevé (liée au protocole)
  - Mécanisme annexe de contrôle

Positionnement dans le noyau

- Niveau 1 : géré par le "matériel"
  - Interface physique
  - Interface virtuelle
- Niveau 2 : géré par le noyau
  - Mise en forme des skb
  - Utilisation de l'API de la carte pour communiquer

Interface virtuelle
Présentation

- But
  - Offrir une présentation abstraite du matériel, en se limitant à la couche OSI concernée
  - Réutilisation des outils développés pour les versions matérielles
- Méthode
  - Ne pas sortir de la couche OSI
  - Présenter les fonctions disponibles et génériques du noyau

Ethernet
Manipulations basiques

- ifconfig : manipulation de la plupart des paramètres de configuration d'un réseau ethernet
  - mtu
  - mode
  - promiscuous
- Manipulation bas niveau :
  - mii-tool
  - eth-tool

Ethernet
ARP

- La configuration est automatique
- Utilisation de l'utilitaire arp
  - liste des adresses
  - enregistrement en dur de certaines adresses
- arpwatch : Outil de diagnostic et de surveillance
  - Mail d'annonce des machines du réseau
  - Détection des conflits
  - Alerte sur les flip-flop

Ethernet
Virtualisation sur ethernet

- Tuntap : Communication avec des entités virtuelles
  - VirtualBox
  - KVM
  - OpenVPN
- Bonding : accroissement des performances
  - Bande passante
  - Sécurité
- Bridge : ajoût des fonctionnalités de switch à Linux

Ethernet
Tuntap : plus d'interfaces pour le même prix

- Le périphérique est remplacé par un programme utilisateur
- Déclaration par accès à `/dev/net/tun`
- Utilisation d'`uml_switch` pour communiquer à plusieurs programmes utilisateurs sur une interface

```
tunctl -u someuser
ifconfig tap0 192.168.0.254 up
```

Ethernet
Bonding : N fois plus pour prix de N

- Aggrégations d'interface avec encapsulation Ethernet
  - Le noyau répartit l'entrée de l'interface virtuelle présentée entre les interfaces physiques
  - Augmentation de la bande passante
  - Collaboration de l'équipement réseau nécessaire
- Détection des interfaces défaillantes
  - Détection bas-niveau (miimon)
  - Envoi de requêtes arp ciblées

```
On host A :
 # modprobe bonding miimon=100
 # ifconfig bond0 addr
 # ifenslave bond0 eth0 eth1
```

```
On the switch :
 # set up a trunk on port1
   and port2
```


Ethernet
Bridge : quand linux se prend pour un switch

- Réunir des interfaces ethernet pour former une entité logique plus importantes
- Relayage des requêtes arp entre les interfaces physiques du bridge
- Switch logiciel, pare-feu transparent

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
```

ATM
Principes

- Principes :
  - Accès au couple VP,VC par une socket
  - Maintien de la connexion par un programme en espace utilisateur
- Deux modes :
  - Démon de signalling
  - Utilisation par un binaire lié à un protocole spécifique

ATMTCP
L'ATM du pauvre

- Interface virtuelle :
  - Fait le lien entre une socket TCP et une interface ATM virtuelle
  - Mode client serveur (ils doivent fonctionner perpetuellement)
- Une mise en oeuvre limitée :
  - Pas de qualité de service
  - Le mélange des AALs n'est pas possible

```
# session A
atmtcp virtual listen
# session B
atmtcp virtual connect Host_B
```

Protocole RFC2684

Pour l'amour de l'encapsulation

- Principes et apports
  - Avoir une interface de type ethernet transportée sur ATM
  - Qualité de service d'ATM sur ethernet
  - Ethernet longue distance
- Méthode
  - Avoir une interface virtuelle
  - La lier à une socket ATM
  - Rediriger de l'un vers l'autre

```
br2684ctl -b -e 0 -c 0 -a 0.35
ifconfig nas0 192.168.0.1 netmask 255.255.255.0 up
```


tcpdump

Le meilleur ami de l'Homme

- Principe
- Copie du flux juste avant la sortie du driver
- Un système de filtre puissant

```
tcpdump -i eth1 -X -nv host mail.inl.fr and not port 22
tcpdump 'icmp[icmptype] != icmp-echo and \
icmp[icmptype] != icmp-echoreply'
```

tcpdump
Le meilleur ami de l'Homme

```
00:28:37.218739 IP 192.168.1.128.1154 > 195.101.59.116.80: . 1:1449(1448) ack 1 win 730
        0x0000:  4500 05dc 0682 4000 4006 6d98 c0a8 0180  E.....@.@.m.....
        0x0010:  c365 3b74 0482 0050 1880 5cf1 f4fb 186b  .e;t...P..\....k
        0x0020:  8010 02da 105d 0000 0101 080a 0030 5a92  .....].......0Z.
        0x0030:  6de4 15a5 4745 5420 2f20 4854 5450 2f31  m...GET./.HTTP/1
        0x0040:  2e30 0d0a 486f 7374 3a20 6d61 696c 2e69  .0..Host:.mail.i
        0x0050:  6e6c                                     nl
```


Introduction

- Famille de protocoles incontournables
- Beaucoup plus complexe qu'il n'y parait
- Coexistence avec les autres piles et les RFCs
  - Contournement des bugs
  - Violation des RFC

Principe

- Architecture du datagramme semblable à Ethernet
- Espace d'adresses de 32bit
- Construit pour l'encapsulation

Le paquet IPv4
Propriétés notables

- Taille variable
- Mécanisme de fragmentation
- Protocole de contrôle icmp
  - Vérification de connectivité : ping
  - Mécanisme d'information : reject
  - Indication de routage : redirect

IP
Décomposition du datagramme

+	Bits 0 - 3	4 - 7	8 - 15	16 - 18	19 - 31
0	Version	Header length	Type of Service (now DiffServ and ECN)	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options				
160/192+	Data				

IP / Exemple de fragmentation
Ping de 1500 sur un lien à MTU 400

```
IP (tos 0x0, ttl  64, id 5503, offset 0, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: ICMP echo request, id 49167, seq 1, length 376
IP (tos 0x0, ttl  64, id 5503, offset 376, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: icmp
IP (tos 0x0, ttl  64, id 5503, offset 752, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: icmp
IP (tos 0x0, ttl  64, id 5503, offset 1128, flags [none], proto: ICMP (1), length: 400) 127.0.0.1
> 127.0.0.1: icmp
IP (tos 0x0, ttl  64, id 5504, offset 0, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: ICMP echo reply, id 49167, seq 1, length 376
IP (tos 0x0, ttl  64, id 5504, offset 376, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: icmp
IP (tos 0x0, ttl  64, id 5504, offset 752, flags [+], proto: ICMP (1), length: 396) 127.0.0.1
> 127.0.0.1: icmp
IP (tos 0x0, ttl  64, id 5504, offset 1128, flags [none], proto: ICMP (1), length: 400) 127.0.0.1
> 127.0.0.1: icmp
```

Configuration d'IP
/proc est ton ami

- Regarder `/proc/sys/net/ipv4/`
- Lire `filesystems/proc.txt`
- Lire `networking/ip-sysctl.txt`

IP
/proc

- `ip_forward` : Activation du routage
- `ip_local_port_range` : Defaut à 1024-4999. À étendre à 32768-61000 sur les machines stressées.
- `ipfrag_high_tresh` et `ipfrag_low_tresh` : Seuils de l'utilisation mémoire réservée à la défragmentation des paquets.
- `ipfrag_time` : Durée de conservation d'un fragment IP en mémoire

IP

- `ipv4/conf/all/accept_redirect` : Accepte les indications de routage venant d'ICMP
- `ipv4/conf/all/secure_redirects` : Limite l'acceptation à la passerelle par défaut
- `ipv4/conf/all/rp_filter` : Filtrage des routes en entrées, le paquet est refusé si le noyau n'a pas une route vers ce réseau sur cette interface.
- `ipv4/conf/IF/proxy_arp` : On répond sur une interface pour les machines se trouvant derrière une autre interface

UDP

- Multiplexage des connexions entre deux IPs
  - Port source
  - Port destination
  - Espace des ports de 16bit
- Notion "propre" de socket au niveau du développement

TCP

- Surcouche de UDP
- Orienté connexion
  - TCP Handshake
  - SYN, SYN ACK, ACK
- Transport fiable
  - Contrôle et réémission en cas de pertes de paquets
  - Négotiation du HandShake
  - Mécanisme avancé permettant d'accélerer les transferts malgré le contrôle

TCP
Décomposition du datagramme

+	Bits 0 - 3	4 - 9	10 - 15	16 - 31
0	Source Port			Destination Port
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Reserved	Flags	Window
128	Checksum			Urgent Pointer
160	Options (optional)			
160/192+	Data			

TCP
Petite liste de fonctions et options

- Réémission :
  - À l'initialisation
  - Durant l'existence d'une connexion
- TCP window :
  - Permet d'accéler le trafic en limitant les ACK
  - Facteur de sécurité dans les réseaux

TCP
/proc (1/2)

Mécanismes de protection :

- `tcp_syncookies` : protection contre les attaques SYN
- `tcp_window_scaling` : vérification des fenêtres
- `tcp_fin_timeout` : Délai pendant lequel on attend FIN

TCP
/proc (2/2)

- Gestion de la performance :
  - `tcp_max_syn_backlog` : longueur de la file d'attente pour une socket serveur
  - `tcp_mem` : seuils globaux d'utilisation de la mémoire par TCP
  - `tcp_rmem` : paramétrage de l'utilisation mémoire en réception des sockets
  - `tcp_wmem` : paramétrage de l'utilisation mémoire en émission des sockets
- Gestion des problèmes de congestion
  - `tcp_congestion_control` : choix de l'algorithme de contrôle de congestion
  - `tcp_ecn` : Explicit congestion notification mal implémenté par certains concurrents

TCP
Conclusion

- Le support des protocoles TCP/IP est complet.
- Les options de configurations sont nombreuses et très fines.
- Heureusement, le défaut est souvent suffisant.
- Par delà, cette implémentation des RFCs et des standards, une surcouche avancée existe . . .

Routage
Principe générique

- Directives de direction
  - Réseau segmenté
  - Comment atteindre une destination ?
  - Passerelle pour le réseau
- Routage : du plus spécifique au plus général

Introduction
Étude des protocoles et périphériques
Fonctionnalités additionnelles
Routage avancé et qualité de service
Sécurité
2 cas pratiques

Routage
Exemple de table de routage

```
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.0.0     192.168.1.2     255.255.255.0   UG    0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1

/sbin/route del -net 192.168.0.0/24 gw 192.168.1.2
```

Routage
Problèmes de performance

- Algorithme de routage :
  - Recherche de la destination dans la liste des réseaux routés
  - Déterminer les inclusions de réseaux
  - Tenir le Giga ou plus
  - Avec 15000 routes (routeur backbone BGP)
- Système de cache :
  - Point de départ : Si on a routé un paquet vers une machine on risque d'en envoyer d'autres
  - Stockage des décisions unitaires
  - Recherche dans une table de Hash (algo en O(1))

Routage
Exemple de cache

Extrait du cache après une simple requête web vers google :

```
route -C
cache de routage IP du noyau
Source          Destination     Passerelle      Indic Metric Ref    Use Iface
192.168.1.128   66.102.9.104    192.168.1.254           0      0        1 eth1
72.14.221.104   192.168.1.128   192.168.1.128   l       0      0        9 lo
192.168.1.128   72.14.221.104   192.168.1.254           0      0        3 eth1
192.168.1.128   66.249.93.104   192.168.1.254           0      0        1 eth1
```

Routage IP
Communication et paramétrage

- RTnetlink
  - Communication avec noyau<->utilisateur au moyen d'une socket de type netlink
  - Transport des messages de routages
  - Configuration
  - montée d'information
  - Mécanisme utilisé par les démons de routage (Quagga, bird)
- Contrôle avancés des décisions de routage
  - Routage par la source
  - Interdiction de routage
  - Par interface réseau

Paramétrage avancé
iproute

- Configuration

```
ip route add 192.168.2.0/24 via 192.168.1.42 dev eth1
```

- Interrogation

```
ip route list
```

- Vidage du cache

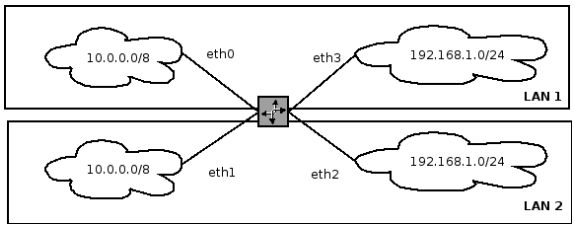
```
ip route flush cache
```

Attention au mot clé cache

- Spécification des règles de routage

Routage : Un cas réel

L'amour du vice



Solution ? Gestion de plusieurs tables de routage

Routage IP multitables

- Gestion des réseaux en doublons
- Différencier les systèmes de routage suivant des critères arbitraires :
  - Filtrage par interface
  - Utilisation de la marque

attention à rp_filter

Routage IP multitables
Mise en oeuvre

- Mise en place des tables dans `/etc/iproutre2/rt_tables`

```
255     local
254     main
```

- Spécification des routes

```
ip route add 192.168.3.0/24 via 192.168.1.1 dev eth0 table ADSL
```

- Listing des routes

```
ip route list table all
ip route list table ADSL
```

- Création des règles d'aiguillage

```
ip rules add dev ppp0 lookup table ADSL
ip rule list
```


Introduction
Étude des protocoles et périphériques
Fonctionnalités additionnelles
Routage avancé et qualité de service
Sécurité
2 cas pratiques

Routage IP multitables
Mise en oeuvre

- Déclaration des tables

```
2 LAN1
3 LAN2
```

- Remplissage des tables

```
ip route add 192.168.1.0/24 dev eth3 table LAN1
ip route add 10.0.0.0/8 dev eth0 table LAN1

ip route add 192.168.1.0/24 dev eth2 table LAN2
ip route add 10.0.0.0/8 dev eth1 table LAN2
```

- Mise en place des règles de routage

```
ip rule add dev eth0 lookup LAN1
ip rule add dev eth3 lookup LAN1

ip rule add dev eth1 lookup LAN2
ip rule add dev eth2 lookup LAN2
```

Leblond
Linux en Réseau

Qualité de service
Implémentation sur IP

- ATM
- MPLS
  - Théorie
    - Semblable à ATM
    - Routage par étiquette
    - Réseau Privé Virtuel par routage
  - Pratique (opérateur)
    - Marketing : résolution de vos problèmes
    - Réalité : MPLS au niveau du backbone
    - Flexibilité faible pour le client

Qualité de service
DiffServ

- Politique d'ordonnancement des paquets en un point du réseau en utilisant comme critère les paramètres propres au paquet.
- Ne nécessite pas de modification du trafic
- Placement aux points de congestion pour être efficace
- Cas du RPV : Mise en place sur tous les sites (devant les liens internet)

Qualité de service
DiffServ

- Egress : Emission Gress
  - Couche glissée juste en amont du driver
  - Contrôle de la file d'émission du driver
  - Stockage dans un buffer interne au noyau
- Ingress : Input Gress
  - Ordonancement des paquets impossibles
  - Vitesse et ordre de réception sont liés au medium physique
  - Seule intéraction est le DROP de paquet
  - IMQ : solution alternative

Qualité de service
Discipline et class

- Queue
  - Tampon entre le noyau et le périphérique
  - Contrôle les flux en appliquant une heuristique
  - Gère les paquets en attente en fonction de cette heuristique
  - `pfifo_fast` : queue par défaut

    ```
    tc qdisc ls
    ```
- Classes
  - Sous éléments de certaines disciplines
  - Permettent de scinder les flux dans des canaux dont les caractéristiques sont paramétrables
  - Des disciplines sont applicables aux feuilles des classes
  - Construction d'une arborescence

Qualité de service
Politique basique sans classe : prio

- Classement par rapport au bit TOS
- Possibilité de forcer l'entrée de certains flux dans certaines classes

```
tc qdisc add dev eth0 prio
```


Qualité de service
Discipline htb

- Principe :
  - Découpage de la bande passante disponible en canaux
  - Paramétrage des canaux en terme de bande passante
  - Notion de priorité des canaux
  - Les canaux sont chainables
- Attribution de la classe au périphérique

```
tc qdisc add dev eth0 root handle 1: htb default 12
```

- Déclaration de la classe racine :

```
tc class add dev eth0 parent 1: classid 1:1 htb \
rate 10mbit ceil 10mbit burst 20k
```

Qualité de service
Classes htb

- Déclaration des classes :
  - 1:6 : Classe pour TCP, limité à 2mbit/s
  - 1:17 : Classe pour UDP, limité à 6mbit/s.
  - 1:12 : Classe pour les autres, limité à 1mbit/s

```
tc class add dev eth0 parent 1:1 classid 1:6 htb \
rate 2mbit ceil 10mbit burst 20k
tc class add dev eth0 parent 1:1 classid 1:17 htb \
rate 6mbit ceil 10mbit burst 20k
tc class add dev eth0 parent 1:1 classid 1:12 htb \
rate 1mbit ceil 1mbit burst 20k
```

- Paramètres :
  - rate : Débit à l'équilibre
  - ceil : Débit maximal autorisé pour la classe
  - burst : Taille du seau, elle est égale à la quantité maximale de données pouvant être envoyée d'une classe avant de servir une autre classe.

Qualité de service
Aiguillage

- Classifieurs : aiguilleur de paquets
- Classifieur fw : par marque

```
tcfilter add dev eth0 protocol ip handle 1 fw flowid 1:6
tc filter add dev eth0 protocol ip handle 2 fw flowid 1:17
```

- Classifieur u32 : par décomposition du paquet ip

```
tc filter add dev eth0 parent 1:0 \
protocol ip prio 1 u32 match ip src 192.168.1.0/24 flowid 1:1
```

Qualité de service
But recherché et principes

- But :
  - Garantir le service rendu aux utilisateurs
  - Les applications métiers doivent être stables au niveau du réseau
  - Les performances doivent être optimales pour l'ensemble des flux
- Principes :
  - Ne pas perdre de vue le but poursuivi
  - Simple is beautiful

Bibliographie

- http://lartc.org/
- http://www.regit.org/article.php3?id_article=13
- http://www-rp.lip6.fr/Ĩochin/qos/qoshtml/qos.html
- man page de ip

Problématique

- Confidentialité
  - Conservation de l'information
  - Dissimulation des échanges
- Protection des ressources
  - Limitation des accès
  - Filtre IP
  - Authentification

VPN et ipsec
Confidentialité en milieu hostile

- Objectifs
  - Connecter différentes entités entre elles
  - Permettre au réseau privé de passer à travers internet
  - Assurer la confidentialité des échanges
- Contraintes
  - L'ensemble des échanges est public
  - Le réseau n'est pas fiable

VPN et ipsec
Principes

- Négotiation d'une clé de chiffrement :
  - En espace utilisateur
  - Négotiation sur UDP port 500
- Établissement d'un canal chiffré
  - ESP : Encapsulating Security Payload
  - protocole IP : 50
- Dialogue
  - Les flux réseau à réseau sont routés par ce canal
  - Attention aux communications pare-feu à pare-feu
- Rafraichissement périodique des éléments cryptographiques

VPN et ipsec
IPsec

- racoon
  - Infrastructure des *BSD
  - Directive de configuration de type routage

```
#!/sbin/setkey -f
add 10.0.0.216 10.0.0.11 ah 24500 -A hmac-md5 "1234567890123456";
add 10.0.0.216 10.0.0.11 esp 24501 -E 3des-cbc "123456789012123456789012";

spdadd 10.0.0.216 10.0.0.11 any -P out ipsec
   esp/transport//require
   ah/transport//require;
```

- openswan
  - Successeur de freewan
  - Fonctionnement par fichier de configuration

```
conn host-to-host
    left=192.0.2.2                # Local vitals
    leftid=@xy.example.com        #
    leftrsasigkey=0s1LgR7/oUM...  #
    leftnexthop=%defaultroute     # correct in many situations
    right=192.0.2.9               # Remote vitals
    rightid=@ab.example.com       #
    rightrsasigkey=0sAQOqH55O...  #
    rightnexthop=%defaultroute    # correct in many situations
    auto=add                      # authorizes but doesn't start this
```

VPN et ipsec
OpenVPN

- Caractéristiques :
  - VPN basé sur une encapsulation SSL
  - Basé sur TunTAP
  - Multi plateforme
  - Authentification par clés
- Avantages :
  - Configuration simple
  - Passage sur TCP donc plus facilement accessible depuis un réseau ouvert
  - Passe le NAT
  - Support des proxys
- Désavantages :
  - Moins standard que Ipsec

Netfilter
Une infrastructure de filtrage complète

- Couche pare-feu de Linux 2.4 et 2.6
  - Successeur de ipchains
  - Pare-feu avec inspection d'état
- Avantages :
  - Des fonctionnalités avancées
  - Des extensions multiples
- Désavantages :
  - Des extensions multiples

Filtrage

- Filtres disponibles :
  - Champs des protocoles
  - État des paquets
  - IPSEC : vérifications avancées
- Décisions :
  - DROP : Bloquage et oubli du paquet

```
iptables -A OUTPUT -p tcp --dport 25 -j DROP
```

  - REJECT : Envoi d'un message de contrôle
  - LOG : Envoi des détails du paquet dans syslog
  - ULOG, NFLOG : Envoi des détails du paquet sur une socket netlink
    - ulogd
    - specter
    - ulogd2
  - TARPIT

Suivi de connexions

- Prendre en compte la notion de connexion
  - Remonter une couche du modèle OSI
  - Prendre en compte la notion de flux
  - Un paquet réponse doit "forcément" passer
- Mise en place d'un suivi de connexions
  - Table de hash contenant les connexions, accès à une entrée en O(1)
  - Prise en compte des protocoles non linéaires
  - Utilisable pour le NAT
- Des filtres adaptés :
  - NEW : le paquet ne fait pas partie d'un flux connu des services de police

```
iptables -A OUTPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT
```

  - ESTABLISHED : le paquet fait partie d'un flux connu
  - RELATED : le paquet est relatif à un flux connu (ftp et ftp-data par ex)

Translation d'adresses

- Des plages d'adresses non routables sont allouées
- Il faut donc les dissimuler en sortie de réseau d'entreprise
  - SNAT
  - Problème des adresses non fixes : MASQUERADE
  - Il peut être pratique de rediriger un flux vers une adresse publique sur une adresse privée : DNAT

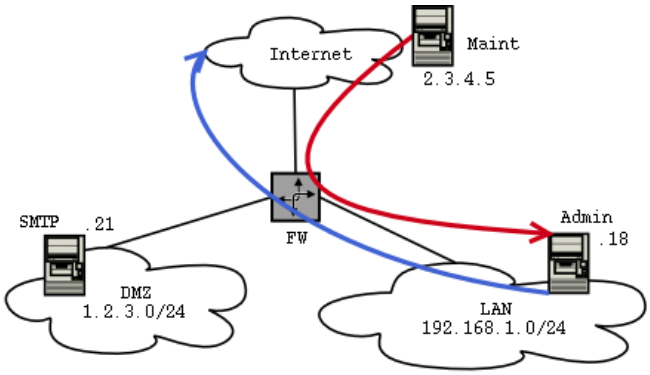
Packet mangling

Le contenu des paquets peut être modifié pour ajuster certains paramètres ou permettre des traitements.

- MARK : marque La marque n'est pas visible sur le réseau
- CONNMARK : marque sur la connexion est pas sur le paquet
- SECMARK : marque de sécurité
- TCPMSS : spécification du Maximum Segment Size
- IPMARK
- TOS
- TTL
- ...

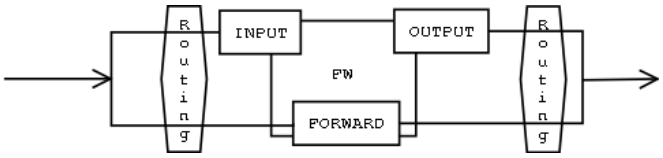
Base de travail

L'architecture de Netfilter sera décrite avec pour exemple le schéma suivant :



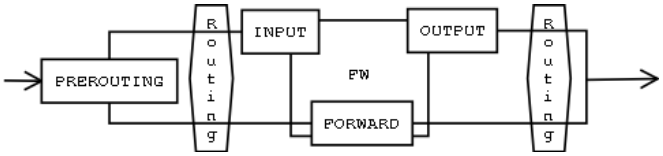
Aiguillage des flux

- Paquet pour le pare-feu INPUT
- Paquet sortant du pare-feu OUTPUT
- Paquet traversant le pare-feu FORWARD



Traduction d'adresse de destination

On veut rediriger les paquets à destination du port 80 du pare-feu vers le serveur en DMZ.

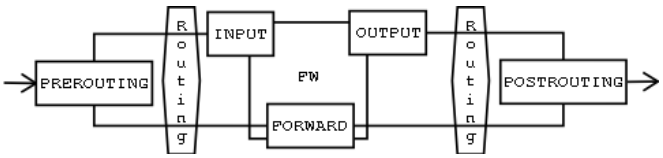


- Le résultat final est celui qui compte au niveau du filtrage
- Le plus tôt est donc le mieux pour traverser
- La modification est faite en entrée.

```
iptables -A PREROUTING -t nat -p tcp --dport 80 -d IPFW -j DNAT --to IPDMZ
```


Traduction d'adresse de destination

Les utilisateurs doivent pouvoir aller sur internet. La réécriture des adresses sources est nécessaire.



- Le monde extérieur doit voir une autre adresse source
- Pas de raison de se mentir
- La modification est faite en sortie.

```
iptables -A POSTROUTING -t nat -o eth0 -j SNAT --to IPFW
```

NfNetlink

- La révolution 2.6.14
- Une nouvelle infrastructure :
  - Nouveau système de passage de messages entre le noyau et l'espace utilisateur
  - Nouvelles bibliothèques et nouveaux outils
- De nouvelles possiblités :
  - Manipulation du suivi de connexions
  - Décision en espace utilisateur (améliorations de l'existant)
  - Système d'accounting et de remonté de logs évolué

Modification et interrogation du suivi de connexions

- L'utilitaire conntrack et libnetfilter_conntrack:
  - Interrogation de la table
  - Modification et suppression d'entrées
  - Écoute des événements
- Conntrackd
  - Réplication des connexions entre deux pare-feu en temps réél
  - Démons en espace utilisateurs qui échangent les données
- pynetfilter_conntrack :
  - Bibliothèque de gestion en python
  - Framework de développement (fait par INL)

Décision en espace utilisateur

- Problème des traitements en mode noyau
  - Difficulté de gestion de la mémoire
  - Danger d'instabilité
  - Modification d'un traitement entraîne redémarrage de la machine
- Prendre la décision en espace utilisateur : libnetfilter_queue
  - Traitement complexe synchrone : snort-inline
  - Traitement complexe asynchrone : NuFW

Load balancing de liens
Principes

- Aggréger les bandes passantes
- Répartir les flux entre plusieurs liens
- Différents modes de découpage
  - Par protocoles
  - Par connexions

Load balancing
Par protocole

- Principe
  - On marque certains flux
  - On change leurs routages suivant la marque
- Mise en oeuvre
  - Création de X tables de routages (une par lien)
  - La route par défaut est le lien sur la table du lien
  - Utilisation de mangle pour marquer les paquets
  - Utilisation de tc pour spécifier la bonne table de routage

Load balancing
Par connexion

- Principe
  - On marque les paquets d'initialisation de manière en suivant la proportion des liens
  - On applique la marque à tous les paquets de la connexion
  - On change leurs routages suivant la marque
- Mise en oeuvre
  - Création de X tables de routages (une par lien)
  - La route par défaut est le lien sur la table du lien
  - Utilisation de mangle pour marquer les paquets
  - Utilisation de tc pour spécifier la bonne table de routage

Load balancing
Mise en oeuvre par protocole

- Utilisation de CONNMARK
  - Propagation de la marque sur l'ensemble des paquets d'une connexion
  - Le suivi de connexion stocke la marque
  - La target CONNMARK permet de restaurer ou de sauver la marque depuis ou dans le suivi de connexion
- Utilisation de nth
  - Création de compteurs
  - Match par rapport au numéro dans le compteur
  - Exemple de syntaxe :

```
iptables -A PREROUTING -t mangle -m nth --counter 1 -
```


Introduction
Étude des protocoles et périphériques
Fonctionnalités additionnelles
Routage avancé et qualité de service
Sécurité
2 cas pratiques

Load balancing
Mise en oeuvre par protocole

- 2 liens
- le premier est trois fois plus gros que le deuxième
- on attribue la marque 1 au lien 1
- on attribue la marque 2 au lien 2
- on crée un compteur de taille 4 :
  - 1 : mark 1
  - 2 : mark 2
  - 3 : mark 1
  - 4 : mark 1

Load balancing
Mise en oeuvre par protocole

dada

```
iptables -A PREROUTING -t mangle -j CONNMARK --restore-
iptables -A PREROUTING -t mangle -m mark --mark 0 -m nt
--every 4 --packet 1 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -m mark --mark 0 -m nt
--every 4 --packet 2 -j MARK --set-mark 2
iptables -A PREROUTING -t mangle -m mark --mark 0 -m nt
--every 4 --packet 3 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -m mark --mark 0 -m nt
--every 4 --packet 4 -j MARK --set-mark 1
iptables -A POSTROUTING -j CONNMARK --save-mark
```


Load balancing
Routage

```
ip route add default gw GW_LINK1 table LINK1
ip route add default gw GW_LINK2 table LINK2
ip rule add fwmark 1 lookup table LINK1
ip rule add fwmark 2 lookup table LINK2
```

NuFW
Pare-feu authentifiant

- L'utilisateur au coeur de l'entreprise
- Majorité des attaques vient de l'intérieur
- Besoin de traçabilité
- Facilité d'administration :
  - Association statique IP==Utilisateur
  - DHCP
  - Machines multi-utilisateurs

Introduction
Étude des protocoles et périphériques
Fonctionnalités additionnelles
Routage avancé et qualité de service
Sécurité
2 cas pratiques

NuFW
Pare-feu authentifiant

- Briser l'association IP==Utilisateur
- Authentification a posteriori
- TCP/IP n'est pas suffisant
- Nécessité de collaborer avec l'OS du poste client