



Étude du filtrage de paquets IP dans un routeur d'accès à un réseau MPLS

*Study of IP packet filtering in a MPLS
label edge router*

Alexandre DAGAN

`alexandre.dagan@wanadoo.fr`

ENSSAT - Lannion - FRANCE

Plan



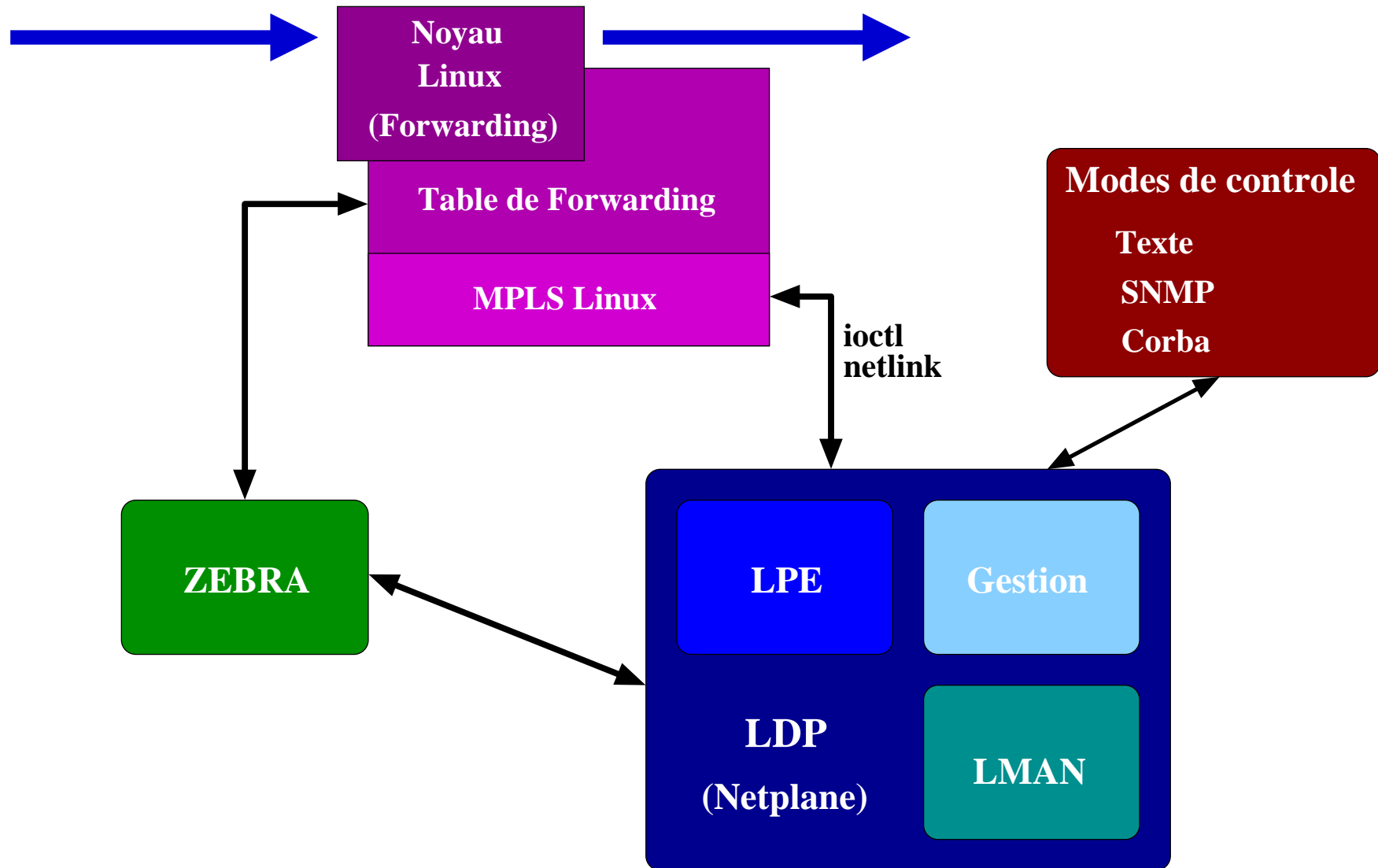
- Objectifs
- France Télécom et le haut débit
- MPLS en bref
- La plateforme de développement
- Le filtrage
- La partie commande

Objectifs



- ▶ Mise en place d'une plateforme de développement
 - ⇒ PC sous Linux transformé en routeur MPLS
- ▶ Classification de paquets
 - ⇒ Étude et implémentation des algorithmes de filtrage
 - ⇒ Adaptation de la classification de paquets à MPLS

La plateforme idéale



▶ France Télécom R&D :

- ⇒ 3900 personnes dont 1500 sur le site de Lannion
- ⇒ Recherche : nouveaux réseaux et nouvelles architectures
- ⇒ Développement : nouveaux services

▶ Le haut débit

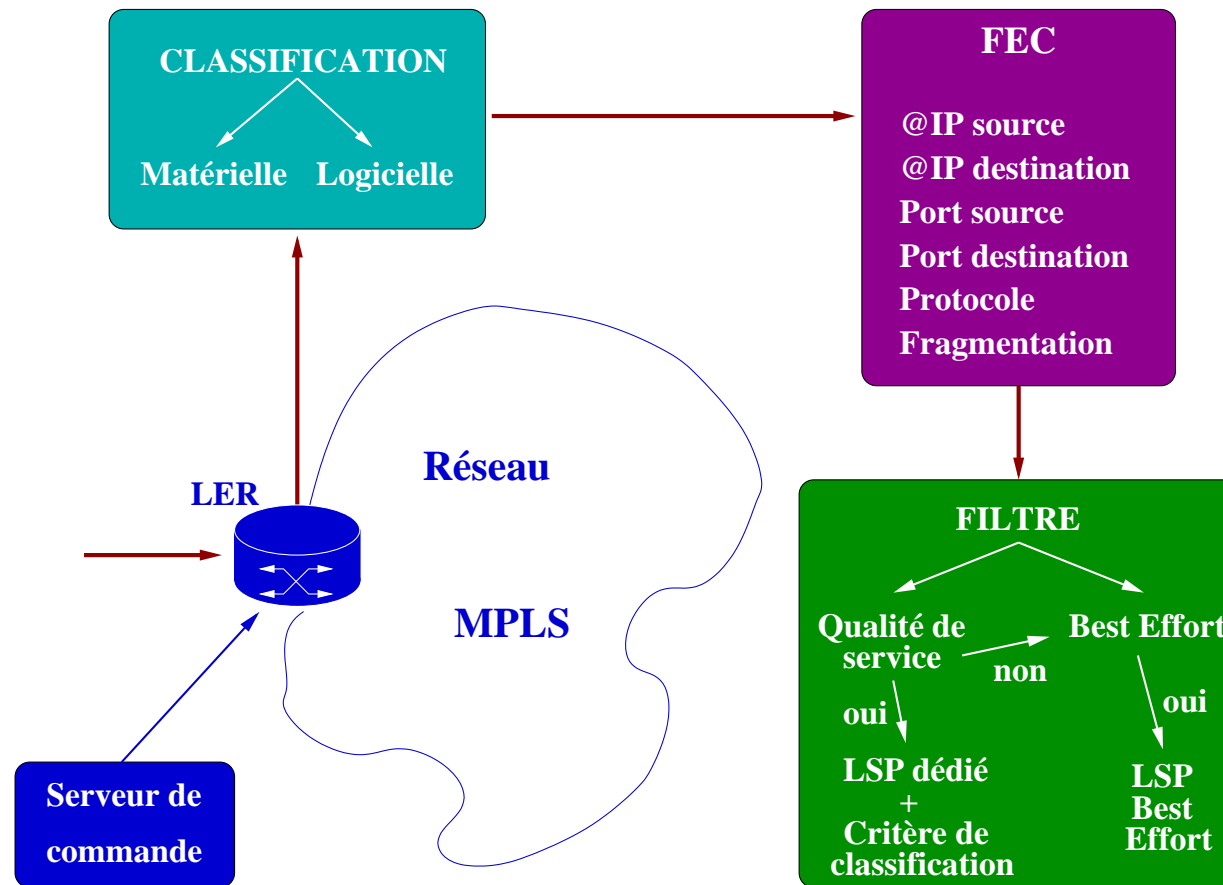
- ⇒ Nécessité d'une qualité de service garantie
- ⇒ Garantir débits et délais de manière transparente pour l'utilisateur final
- ⇒ QoS sur IP, le support de l'Internet

- ▶ Indépendant du protocole de niveau 2
 - ⇒ Adaptation de l'existant facilité
 - ⇒ Économiquement avantageux
- ▶ Le principe de routage contraint amélioré
 - ⇒ La maîtrise des connexions
 - ⇒ La commutation d'étiquette remplace le routage
- ▶ MPLS permet :
 - ⇒ Qualité de service sur des protocoles sans QoS
 - ⇒ Routage niveau 3 vers commutation niveau 2
 - ⇒ Accélération du routage dans les équipements du réseau

Principe de fonctionnement



Un PC sous Linux fonctionnant comme un routeur d'accès à un réseau MPLS.



La plateforme de développement &

Deux choix logiciels :

- ▶ Le noyau Linux
- ▶ Une solution commerciale pour la pile de signalisation MPLS

La plateforme de développement &

Deux choix logiciels :

- ▶ Le noyau Linux
 - ⇒ Configuration en routeur
 - ⇒ Patch *MPLS for Linux* + compilation
 - ⇒ Rapidement réalisé
- ▶ Une solution commerciale pour la pile de signalisation MPLS

La plateforme de développement &

Deux choix logiciels :

- ▶ Le noyau Linux

- ⇒ Configuration en routeur
- ⇒ Patch *MPLS for Linux* + compilation
- ⇒ Rapidement réalisé

- ▶ Une solution commerciale pour la pile de signalisation MPLS

- ⇒ Sur-couche logicielle, portable
- ⇒ Travail à partir d'un fichier de démo
- ⇒ Importantes évolutions et modifications
- ⇒ Beaucoup de contraintes, réalisation difficile

Le noyau Linux



Travail réalisé :

- ▶ Travail sur deux machines en parallèle
- ▶ Patch et compilation du noyau Linux
- ▶ Test de fonctionnement avec *mplsadm*
 - ⇒ Établissement d'une connexion ATM
 - ⇒ Mise en place d'un LSP
 - ⇒ Transfert de données
- ▶ Tests concluants !

Le démon LDP (1)



Adaptation de la démo de base

- Familiarisation et étude approfondie du code
- Fichier de configuration + *parser*
- Masquage des interruptions

Résultats

- 5000 lignes de code analysées
- 1000 pages de documentation étudiées
- Une centaine de lignes de code écrites
- Le masquage des interruptions n'est pas au point.

Le démon LDP (2)



La partie LPE

- ▶ Interconnexion LDP / noyau Linux
- ▶ Répercution des actions sur notre structure
- ▶ Transmission des modifications au noyau

Bilan

- ▶ Évolution perpétuelle du code
- ▶ Plusieurs mois de travail
- ▶ 6000 lignes de codes étudiées
- ▶ Une centaine de lignes écrites

Le démon LDP (3)



Intégration de *Zebra*

- MPLS nécessite un protocole de routage
- Centralisation des informations de routage
- Module *Zebra* dans LDP

Bilan

- Corrections d'erreurs dans leur code
- Documentation inexistante
- Dialogue impossible via *Zebra*
- Abandon \Rightarrow connexion directe par *netlink*
- Utilisation indirecte de *Zebra* via la table de routage du noyau

Le voyage d'un paquet



Étude approfondie des mécanismes réseau du noyau Linux et des particularités dues à MPLS.

♦ Méthode

- ⇒ Base : document sur le trajet effectif d'un paquet IP
- ⇒ Traque du trajet dans le code Linux
- ⇒ Détermination des variations dues à MPLS

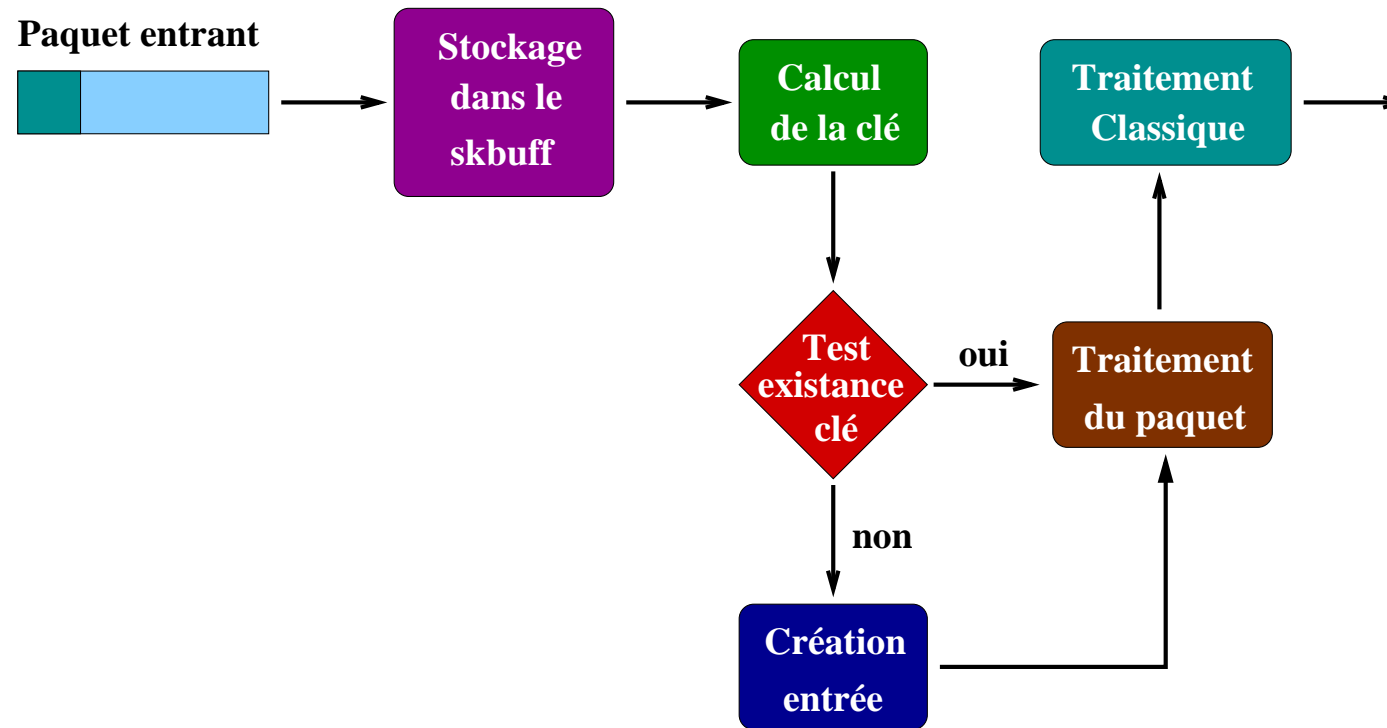
♦ Résultats

- ⇒ Changements importants avec MPLS
- ⇒ Nouvelle table de routage
- ⇒ Filtrage des paquets entrant ⇒ traitement
- ⇒ Traduction et augmentation du document initial

Clé et table de hashage



Le routage classique :

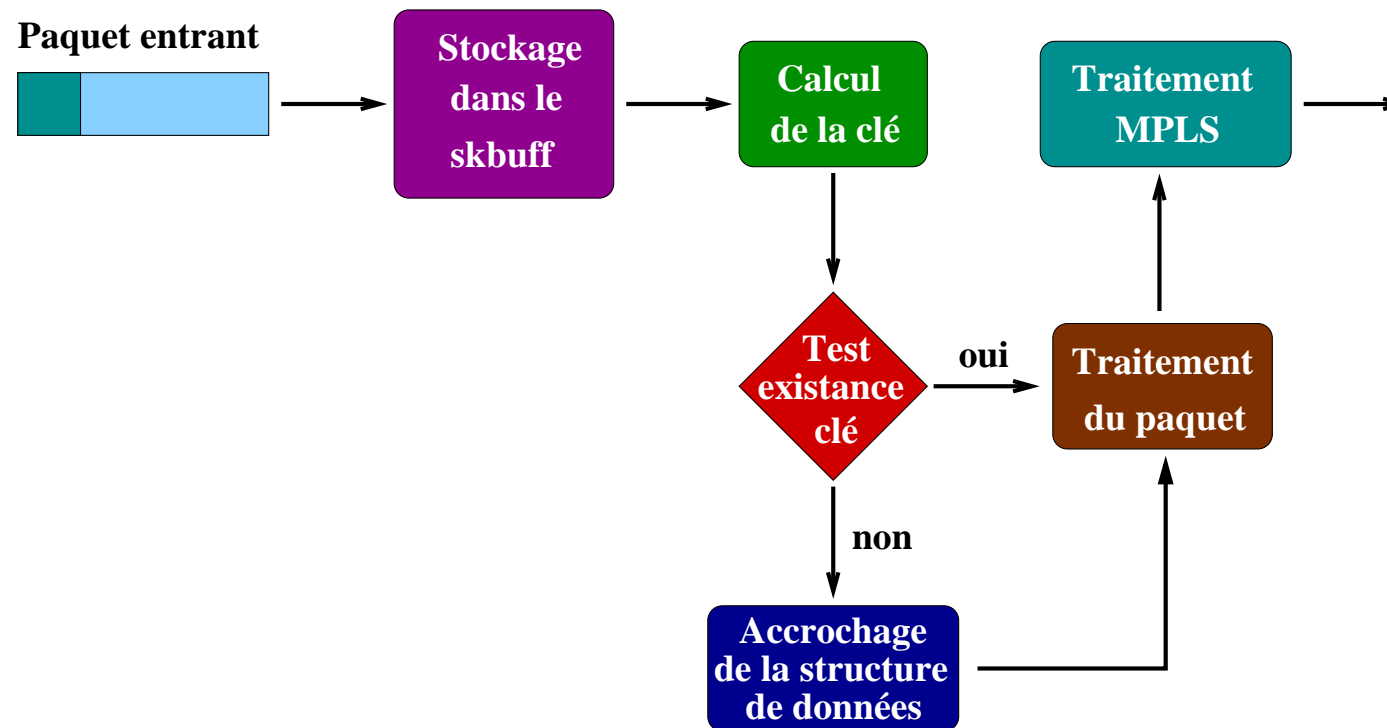


- ▶ Accélération du routage avec la clé de hash
- ▶ Algorithme du *longest match prefix* déroulé pour le premier paquet seulement

Clé et table de hashage



Le routage MPLS :

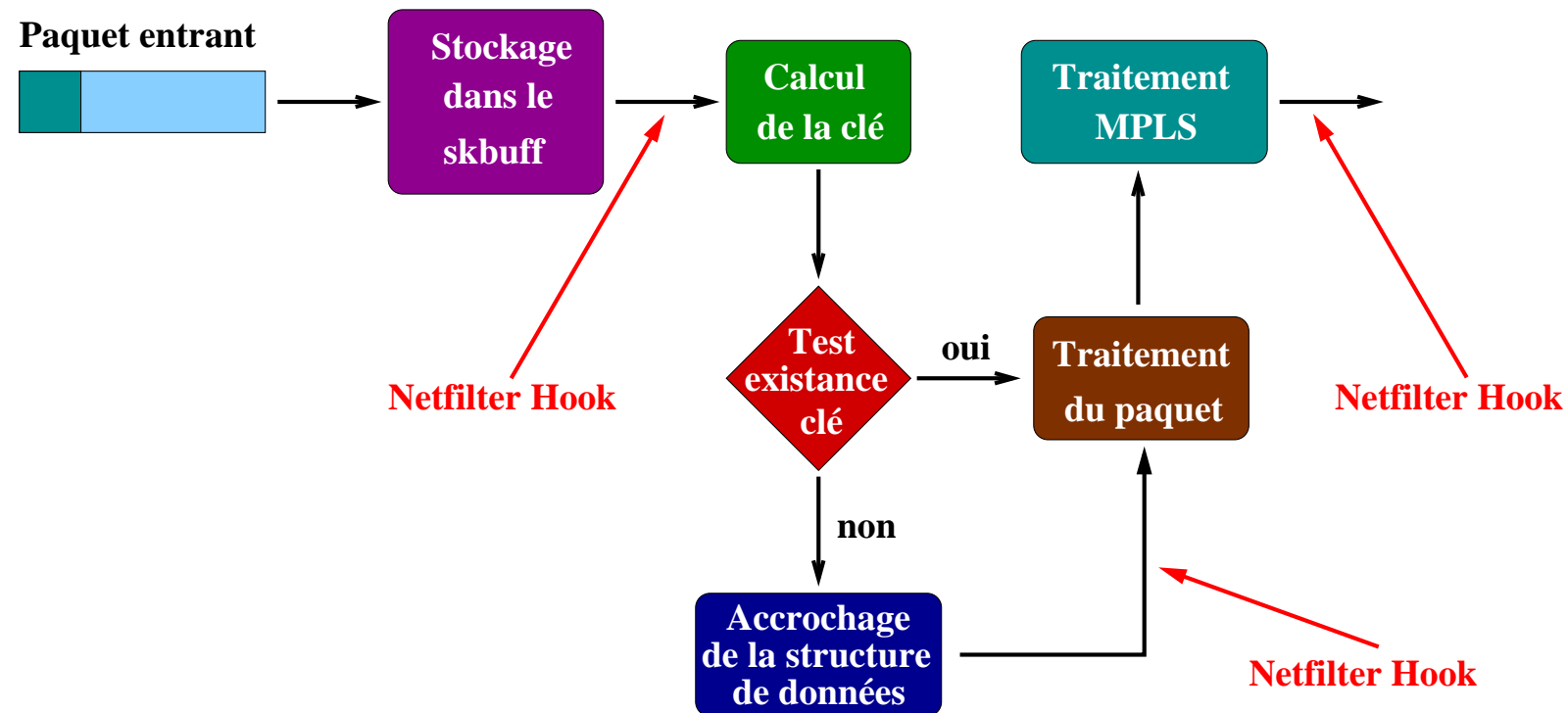


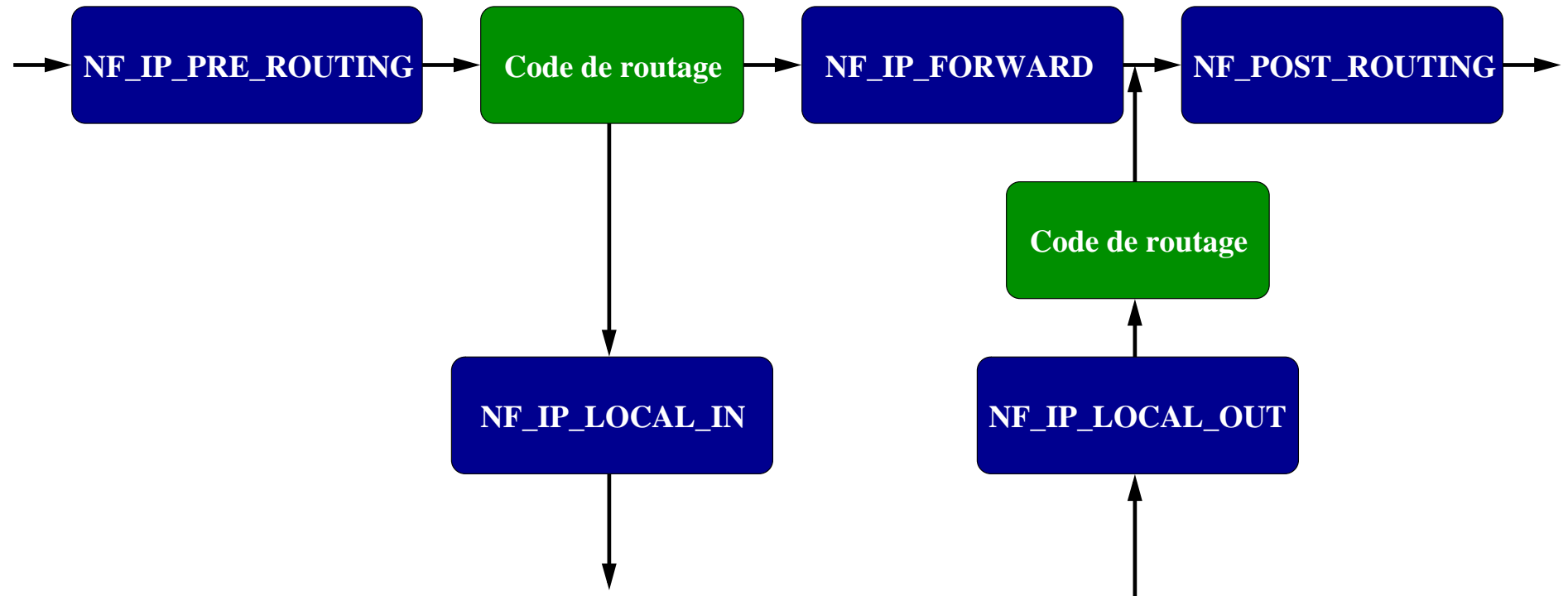
- ▶ Accélération du routage avec l'index mpls
- ▶ Algorithme de création de la clé déroulé pour le premier paquet seulement

Clé et table de hashage



Le routage MPLS et Netfilter :





- Mécanisme de gestion du filtrage
- Canevas pour modification des paquets IP
- Notion d'accroche ou *hook*
- Programmation modulaire

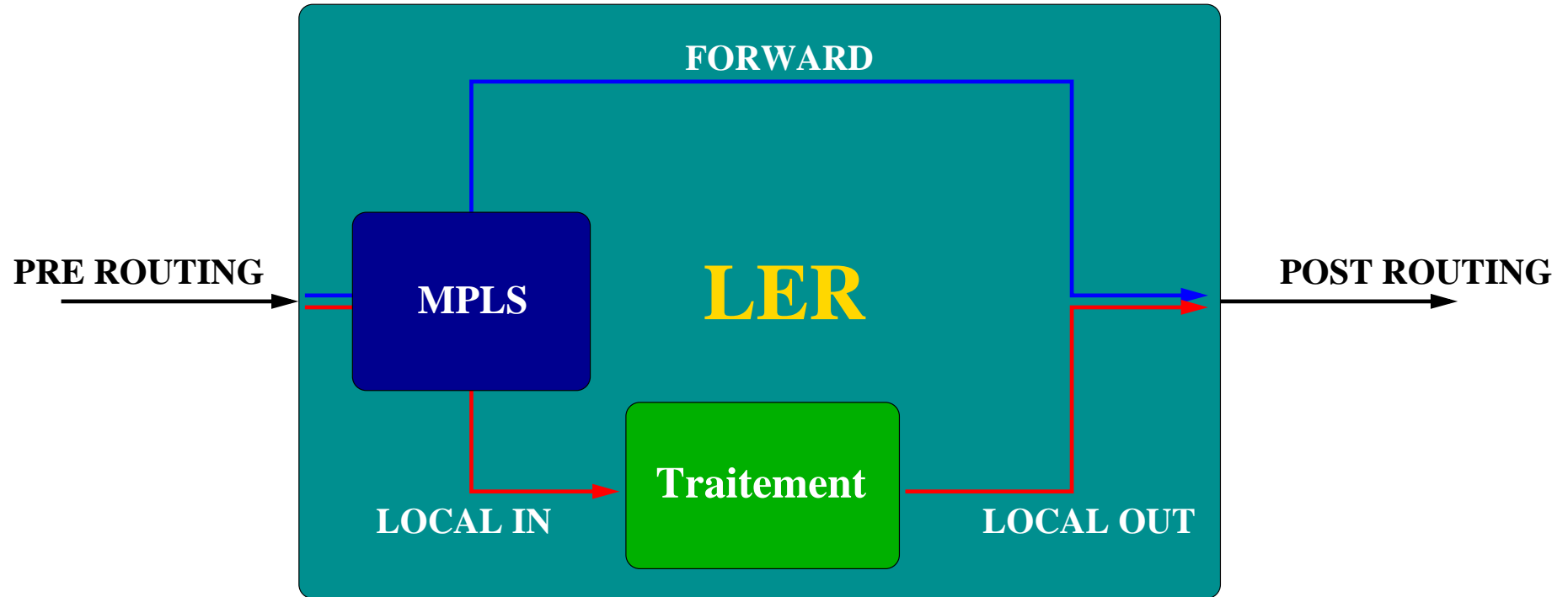
Iptables : l'utilitaire de *Netfilter*

- ▶ Programme de commande et de configuration de *netfilter*
- ▶ Gestion d'un tableau de règles en mémoire
- ▶ Insertion/suppression de règles dans la table de filtrage
- ▶ Envoi d'informations vers le noyau

Extension du filtrage (1)



Un nouveau module noyau pour netfilter :



- ▶ Principe similaire au FWMARK
- ▶ Traitement en PRE_ROUTING
- ▶ Marquage des paquets en fonction de leur FEC

Extension du filtrage (2)



Une nouvelle librairie pour *Iptables* :

- ▶ Version utilisateur du module noyau précédent
- ▶ Modèle semblable à celui employé pour NFMARK
- ▶ Nouvelles règles basées sur le marquage MPLS

Réalisation et proposition de patchs pour le noyau Linux à la communauté

La partie commande



Interface de commande

- ▶ Principe de client/serveur
- ▶ Communication par *sockets*
- ▶ Mode ligne de commande

Bilan

- ▶ 4 types d'actions : ajout, suppression, affichage et destruction
- ▶ Fonctionnel...
- ▶ ... mais, rudimentaire

Conclusion



▶ Point de vue sur projet

▶ Point de vue sur le tutorat

Conclusion



- ▶ Point de vue sur projet
 - ⇒ Fortes contraintes dues aux choix techniques
 - ⇒ Découverte approfondie des mécanismes réseau du noyau Linux
 - ⇒ Adéquation connaissances acquises / connaissances personnelles (Réseau / Linux)
- ▶ Point de vue sur le tutorat

Conclusion



♦ Point de vue sur projet

- ⇒ Fortes contraintes dues aux choix techniques
- ⇒ Découverte approfondie des mécanismes réseau du noyau Linux
- ⇒ Adéquation connaissances acquises / connaissances personnelles (Réseau / Linux)

♦ Point de vue sur le tutorat

- ⇒ Projet d'envergure
- ⇒ Confrontation au monde de la R&D
- ⇒ Longue durée, permet d'observer toutes les phases d'un projet



Des questions ?