

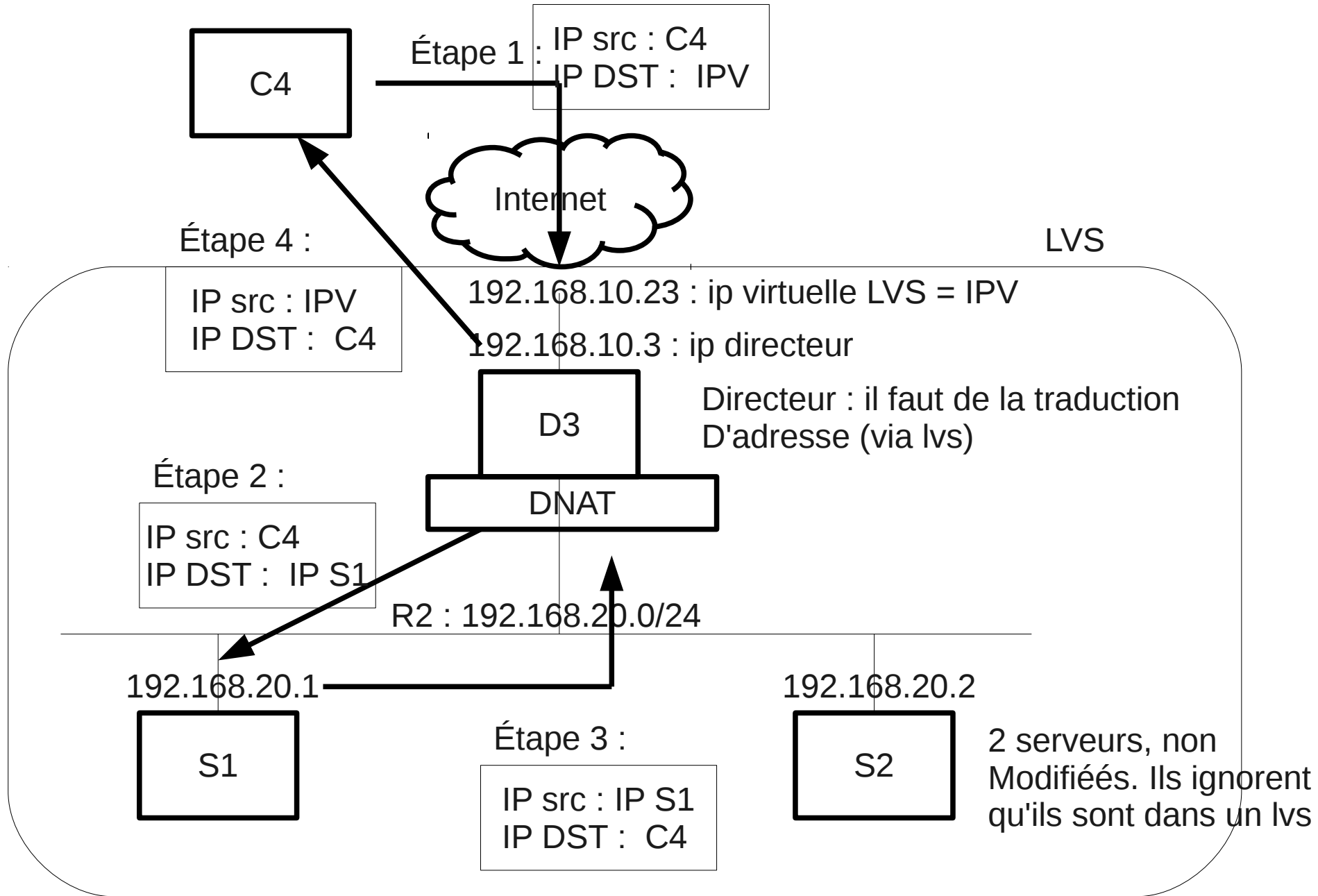
# Équilibrage de charge

- Linux virtual server
  - Un directeur répartit les requêtes
  - Les serveurs ne le savent pas forcément
  - Agit comme un répartiteur de couche 4
- Cluster beowulf
  - La charge est répartie de façon coopérative entre les serveurs du cluster

# Principe de fonctionnement

- Les clients voient l'IP virtuelle (VIP) du LVS
- La VIP est routable et accessible depuis les clients
- Les serveurs ont des IP réelles (RIP)
- Le directeur est un répartiteur de charge (load balancer)
- 3 modes de fonctionnement :
  - LVS-NAT
  - LVS-DR
  - LVS-TUN

# LVS-NAT



# LVS-NAT

- 1) Le client fait une requête sur l'ip virtuelle du LVS (IPV) ;
- 2)DNAT : Le directeur change l'ip destination (et parfois le port destination) qui devient celle du serveur cible (IP S1 dans l'exemple) ;
- 3)Le serveur répond au client. Sa réponse transite par le directeur qui est aussi son routeur par défaut ;
- 4)La règle DNAT de l'étape2 est appliquée dans l'autre sens et le paquet est transmis au client avec ipv comme ip source.

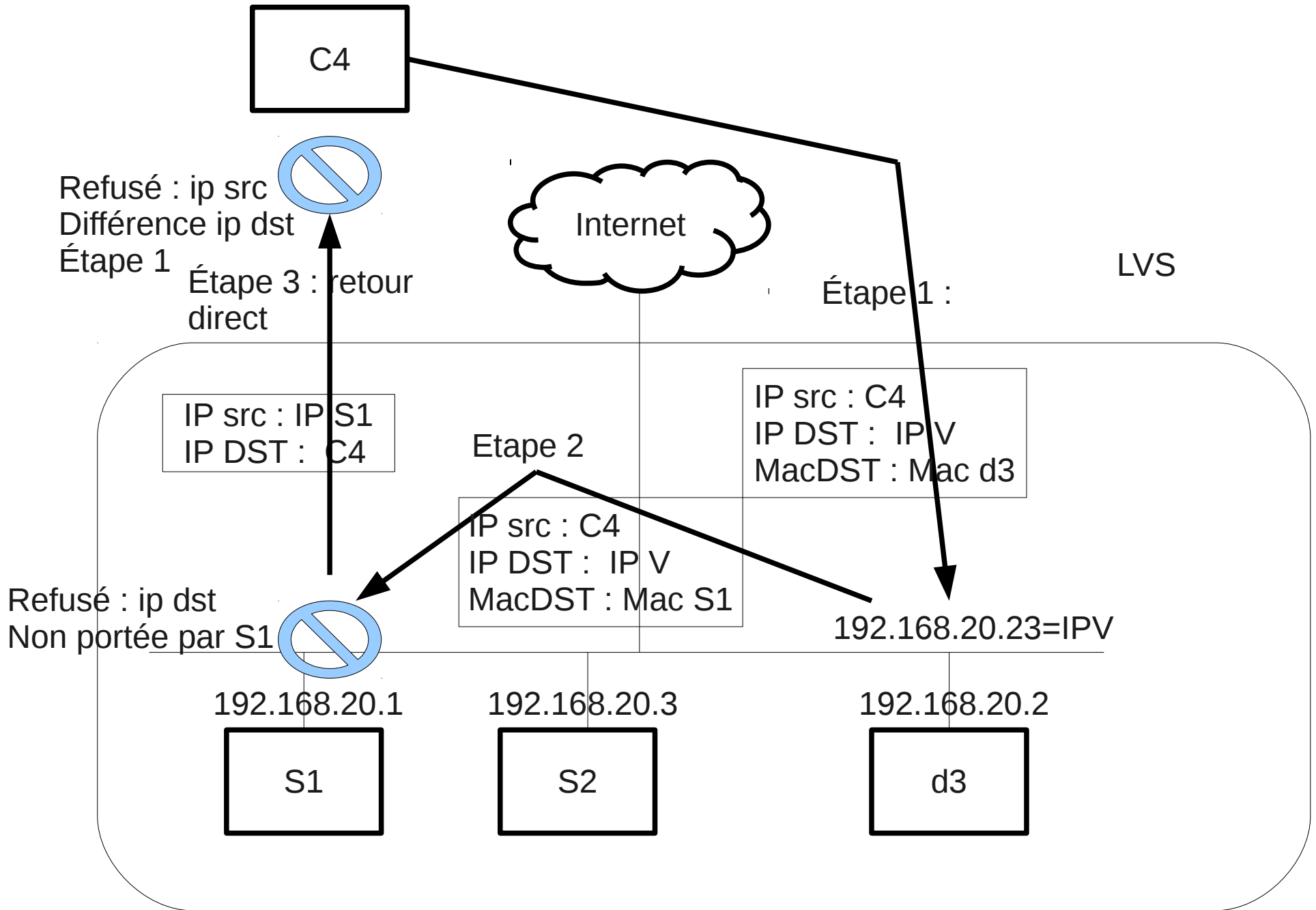
# LVS-NAT

- Les clients interagissent avec le directeur
- Le directeur fait de la traduction d'adresse
- Le directeur est le routeur de sortie des serveurs
- Le DNAT est géré par le module ip\_vs
- Avantage :
  - les serveurs ignorent qu'ils sont dans un LVS
  - OS des serveurs sans importance
  - Les serveurs WeB peuvent écouter sur un port différent de celui sur lequel le client envoie sa requête (permet par le DNAT)
- Défauts :
  - Tout passe par le directeur en entrée (petit) et en sortie (gros)

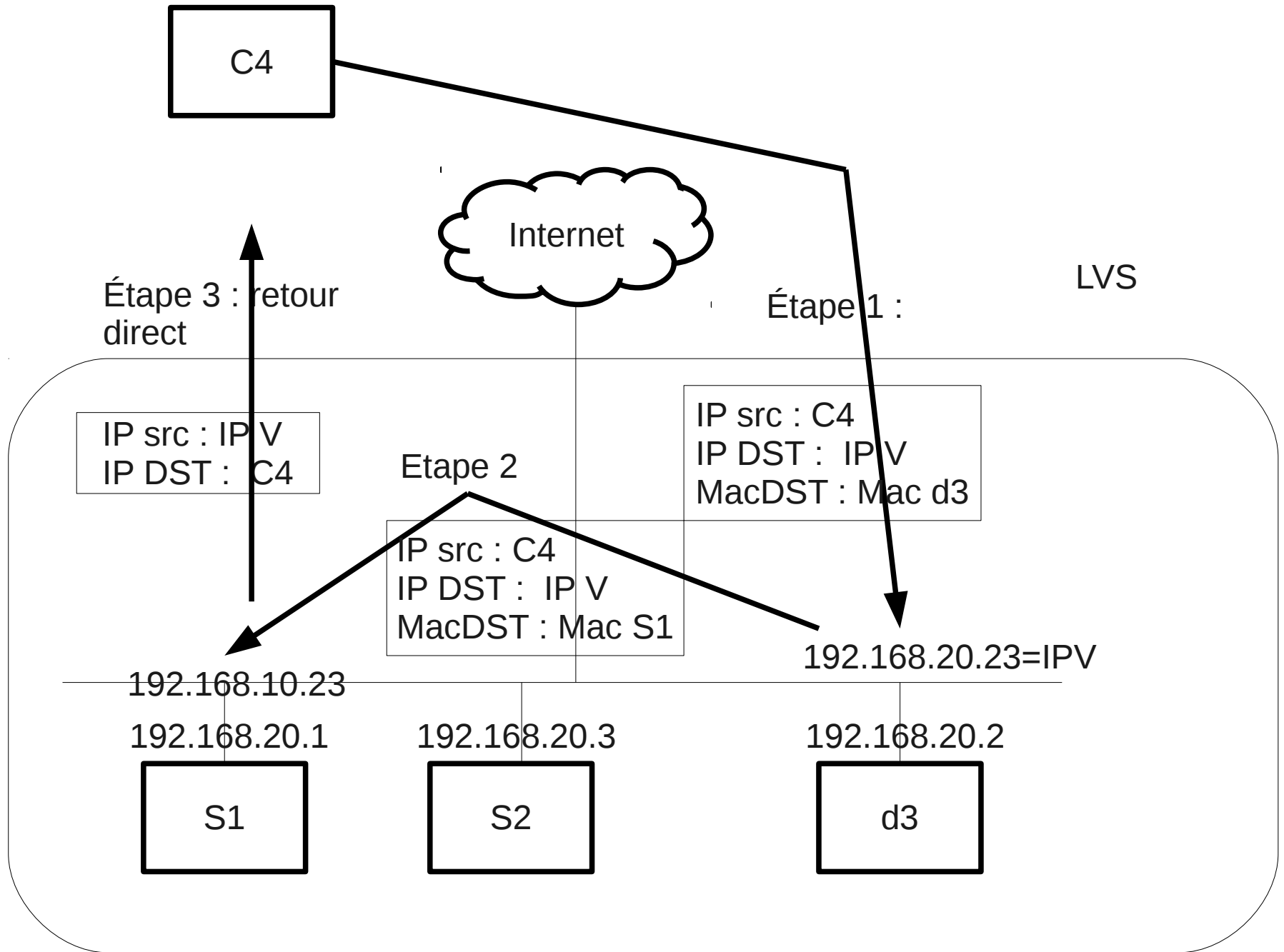
# LVS-DR

- But : éviter de passer par le directeur pour les réponses des serveurs
- Le retour se fait directement depuis les serveurs vers les clients
- Depuis une ip virtuelle commune aux serveurs et au directeur
- Seul le directeur annonce cette ip via arp (car lui seul doit avoir les paquets en entrée)

# LVS-DR



# LVS-DR : solution





# LVS-DR : arp sur les serveurs

- Seul le directeur doit répondre aux requêtes ARP sur l'ip virtuelle du LVS
- Solutions :
  - Affecter l'IPV à chaque serveur et :
    - Paramètres arp\_announce et arp\_ignore du noyau des server
    - Filtrage par arptables :
      - Filtrer sur les serveur tout paquet ARP entrant destiné à l'ip virtuelle
      - Remplacer l'ip source virtuelle des annonces arp des serveurs par leur ip réelle
  - En faisant de la redirection d'adresse (éventuellement de port) en entrée sur les serveurs avec netfilter/iptables
    - Tout ce qui est destiné à l'ip virtuelle est réécrit comme destiné à l'ip réelle du serveur
    - Dans ce cas, le serveur n'a pas besoin d'avoir l'ip virtuelle comme ip

# LVS-TUN

- Idem lvs-DR mais le lien directeur-serveurs est un tunnel
- N'impose pas aux serveurs d'être localisés au même endroit

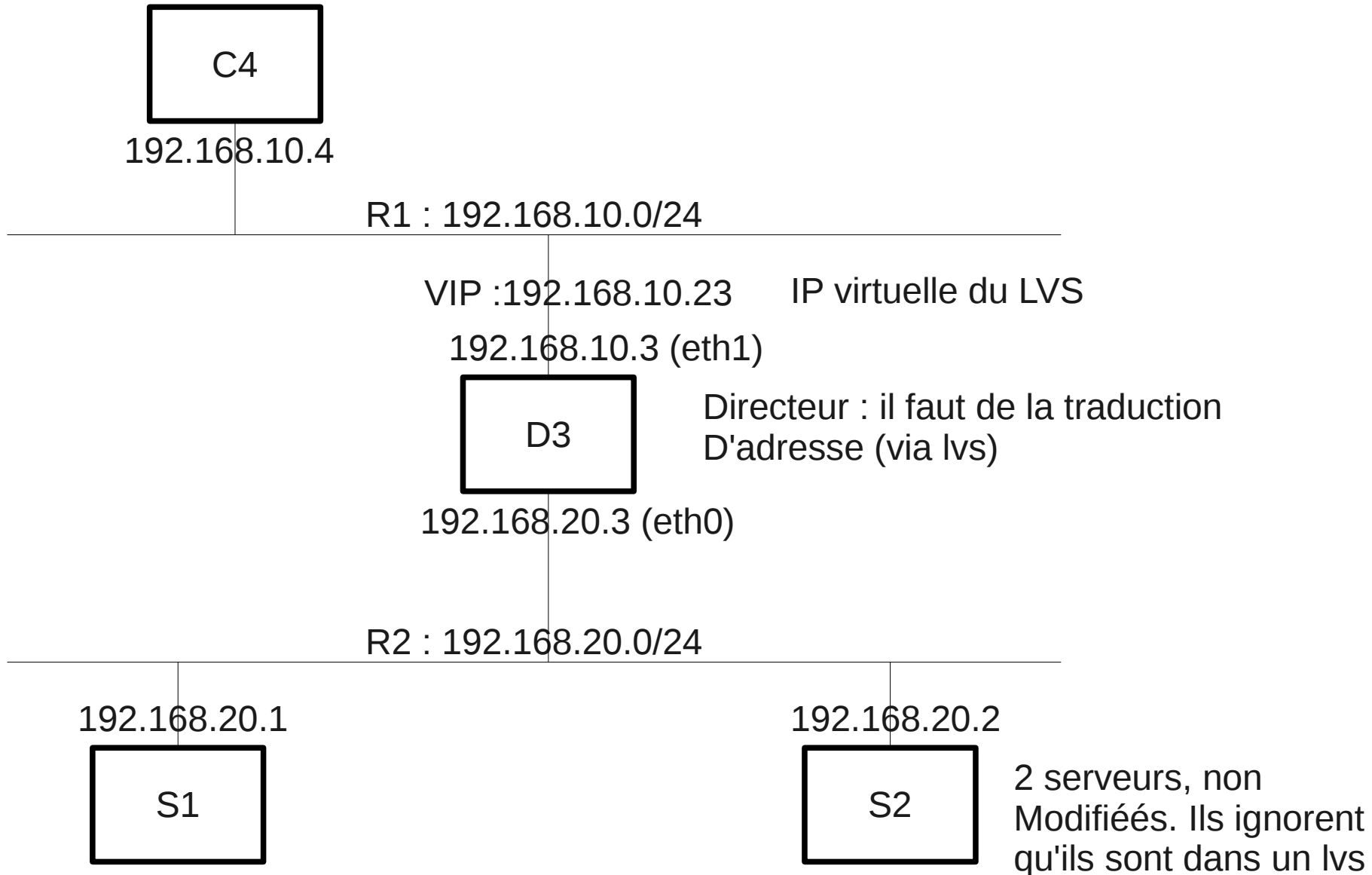
# Haute disponibilité/répartition de charge

- LVS : répartit la charge sur les serveur
- MAIS pas de tolérance de panne :
  - Le directeur est un SPOF
  - Le directeur assume toute la charge de son travail de directeur
  - LVS ne palie pas l'indisponibilité d'un serveur
- Solution : (cf keepalived plus loin)
  - Haute disponibilité sur le directeur (directeur backup, ...)
  - ne plus utiliser les serveur HS : les détecter, les isoler

# LVS : Pré-requis logiciels

- Noyau supportant lvs
  - Dans le noyau : `grep ip_vs System.map`
  - En module :
    - Chargé à la première utilisation de `ip_vs`
    - La disponibilité peut-être vérifié avec `modinfo ip_vs`
- Commande `ipvsadm` (package debian `ipvsadm`)

# Maquette de test LVS-NAT



# Configuration de base

- Sur S1 : un serveur WeB écoutant sur le port 80 avec une page d'accueil contenant le texte « Je suis le serveur 1 et j'ai faim »
- Sur S2 : un serveur WeB écoutant sur le port 80 avec une page d'accueil contenant le texte « Je suis le serveur 2 et j'ai soif»
- Le directeur doit être le routeur par défaut des deux serveurs
- Activation du routage sur le directeur :
  - Décommenter la ligne `net.ipv4.ip_forward` dans `/etc/sysctl.conf`
  - Puis recharger le fichier par « `sysctl -p` »

# Configuration de LVS-NAT

- ajout d'une ip-aliases sur eth1
  - ajouter les lignes suivantes dans /etc/network/interfaces :

```
auto eth1:0
```

```
iface eth1:0 inet static
```

```
address 192.168.10.23
```

```
netmask 255.255.255.0
```

- puis : ifup eth1:0
- vérifier via 'ping 192.168.0.23'

# Configuration de LVS-NAT

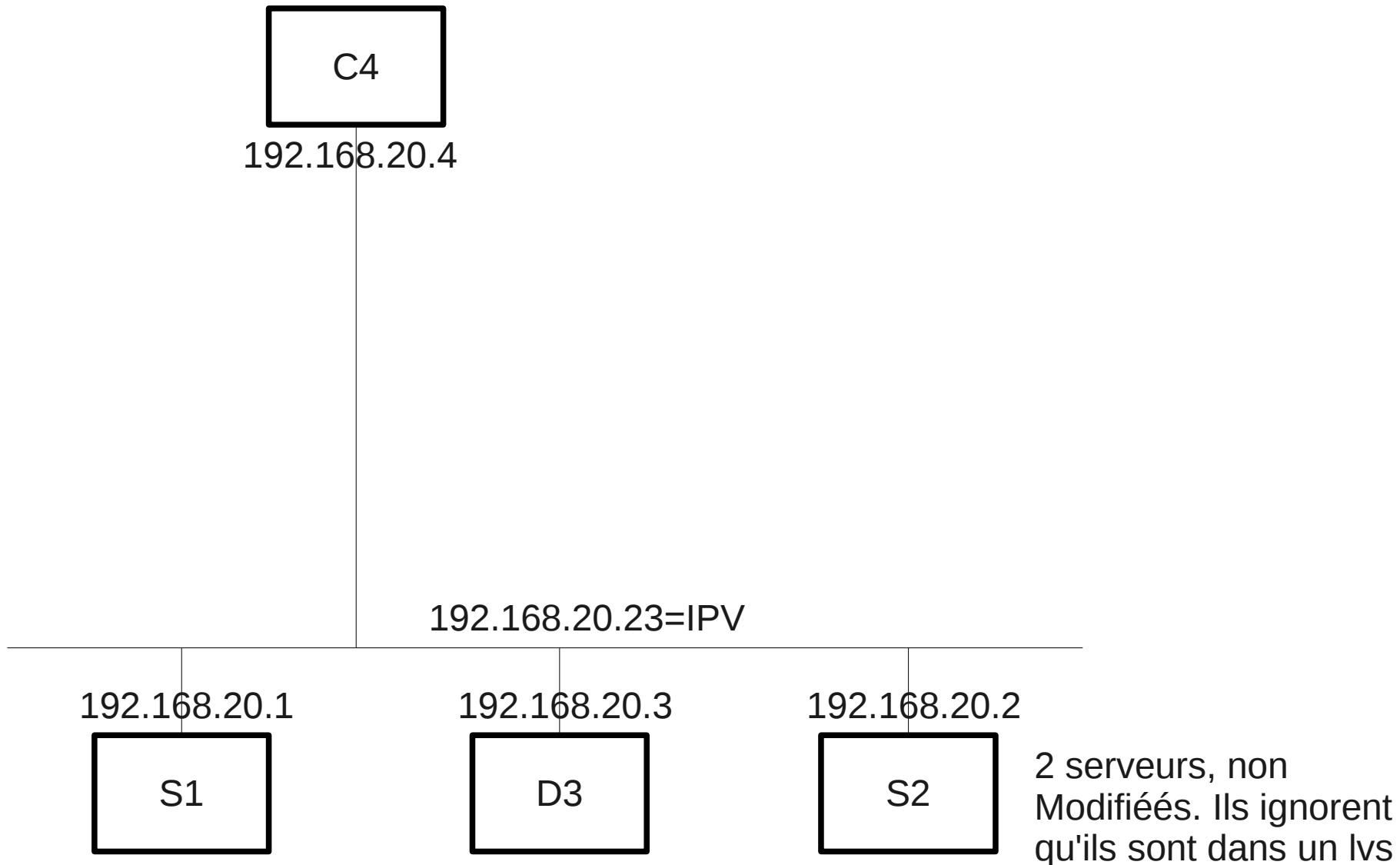
- Déclaration du LVS :
  - `ipvsadm -A -t 192.168.10.23:80 -s rr`
    - `-A` : ajout d'un LVS
    - `-t` : tcp
    - `-s` : méthode de répartition de charge
- Ajout des serveurs
  - `ipvsadm -a -t 192.168.10.23:80 -r 192.168.20.1:80 -m -w 1`
    - `-a` : ajout d'un serveur réel à un LVS
    - `-r` : ip:port du serveur réel à ajouter
    - `-m` : active la traduction d'adresse (masquerading)
    - `-w` : poids (inutile ici car on est en round-robin)



# Test de LVS-NAT

- Avec un navigateur, faites des requêtes sur <http://192.168.10.23/> (ip virtuelle du LVS)
- Commentez l'affichage
- Sur le LVS, que donne la commande `ipvsadm ? ipvsadm -L ? ipvsadm -Lc ?`
- Faites une requête et consultez encore `ipvsadm` (`ipvsadm -L, -Lc, -Ln, ...`)
- Faites des analyses de trames sur les deux réseaux et comparez les trames avant et après
- Déconnectez la carte réseau vmware d'un des serveurs et répétez les étapes ci-dessus

# Maquette de test LVS-DR



# Configuration de base

- Sur S1 : un serveur WeB écoutant sur le port 80 avec une page d'accueil contenant le texte « Je suis le serveur 1 et j'ai faim »
- Sur S2 : un serveur WeB écoutant sur le port 80 avec une page d'accueil contenant le texte « Je suis le serveur 2 et j'ai soif»
- Le directeur a l'IP virtuelle du LVS comme seconde IP sur sa carte réseau
- Vous appliquerez l'une des trois solutions (ou les trois tour à tour en fonction de votre temps) pour que S1 et S2 acceptent les paquets destinés à IPV sans émettre de requêtes ARP indiquant l'IP IPV.

# Configuration de LVS-DR

- Déclaration du LVS :
  - `ipvsadm -A -t 192.168.20.23:80 -s rr`
    - `-A` : ajout d'un LVS
    - `-t` : tcp
    - `-s` : méthode de répartition de charge
- Ajout des serveurs
  - `ipvsadm -a -t 192.168.20.23:80 -r 192.168.20.1:80 -g -w 1`
    - `-a` : ajout d'un serveur réel à un LVS
    - `-r` : ip:port du serveur réel à ajouter
    - `-g` : active le 'direct routing'
    - `-w` : poids (inutile ici car on est en round-robin)

# Test de LVS-DR

- Avec un navigateur, faites des requêtes sur `http://192.168.20.23/` (ip virtuelle du LVS)
- Commentez l'affichage
- Sur le LVS, que donne la commande `ipvsadm` ?  
`ipvsadm -L` ? `ipvsadm -Lc` ?
- Faites une requête et consultez encore `ipvsadm`  
(`ipvsadm -L`, `-Lc`, `-Ln`, ...)
- Faites des analyses de trames sur les deux réseaux et comparez les trames avant et après
- Déconnectez la carte réseau vmware d'un des serveurs et répétez les étapes ci-dessus

# Tolérance de panne

- Pas de tolérance de panne sur notre maquette LVS
  - Les serveur HS sont toujours utilisés par LVS
  - Si un serveur est HS sur un lot de 10, une fois sur 10, les requêtes échouent
  - Si c'est 9 sur 10 qui sont HS, ...
- Si le directeur tombe, le LVS est hors service et ne répond pas

# Tolérance de panne : les serveurs WeB

- Nécessité de réagir en cas de de serveurs WeB HS :
  - Tester l'état de serveur
  - Retirer du LVS tout serveur HS
  - Y remettre les serveurs revenus à la vie
- Solutions :
  - À la main avec des scripts : facile mais fastidieux
  - Avec keepalived

# Tolérance de panne : keepalived

- Permet la mise en place d'un LVS (/etc/keepalived/keepalived.conf en général)
- Tolérance de panne des serveurs WeB
- Tolérance de panne du directeur via vrrp



# Travail :

- Migrer une des maquettes existantes vers keepalived
- Sans tolérance de panne pour l'instant
- La tester

# Keepalived : tolérance de panne des serveurs WeB

- Quorum : le poids total des serveurs actifs doit être supérieur au quorum
- Sinon :
  - On peut paramétrer le lancement de scripts : pour désactiver le LVS par ex.
  - Si un sorry server existe, toutes les requêtes lui sont transmises
- Sorry server :
  - Serveur vers lequel basculent toutes les requêtes en cas de poids total inférieur au quorum
  - En général, répond avec une page du style « service interrompu pour cause de charge trop élevée »

# Keepalived : tolérance de panne des serveurs WeB

- Paramètre `délay_loop` : temps en secondes entre deux tests de serveurs WeB
- Healthchecker : tests des serveurs
  - Se spécifie dans la définition d'un serveur réel
    - Des serveurs réels appartenant au même LVS peuvent avoir des tests différents
  - Plusieurs tests peuvent être indiqués
    - Si un seul échoue, le serveur est considéré comme HS

# Healthchecker classiques

- TCP\_CHECK :
  - teste si un serveur écoute sur le port spécifié
- HTTP\_GET et SSL\_GET :
  - Teste une page WeB donnée
    - À partir du statut http retournée
    - En validant le contenu de la page via md5
  - SSL\_GET : via SSL
  - On peut préciser un virtualhost dans la définition du LVS (donc pas au niveau du healthchecker)
  - Genhash : outil pour générer le code md5 d'une page WeB distante
- SMTP\_CHECK :
  - Pour tester des serveurs SMTP
- MISC\_CHECK
  - Pour utiliser des scripts écrites par l'administrateur pour tester les serveurs

# Votre travail :

- Mettre en place la partie tolérance de panne des serveurs dans votre maquette keepalived
- Tester plusieurs healthchecker
- Tester un sorry server (à installer d'abord)

# Tolérance de panne des directeurs

- Directeur HS => lvs inaccessible
- Solution : tolérance de panne sur les directeurs
- Keepalived la gère via vrrp (rfc 3768)
  - Sans Mac virtuelle (limitation du noyau linux)
  - Remplacée par un paquet ARP gratuit
- Keepalived gère la synchronisation des tables (NAT & Co ) en démarrant les services de synchronisation noyau :
  - ipvs\_syncmaster
  - ipvs\_syncbackup

# Votre travail

- Mettre en place une maquette haute disponibilité avec deux directeurs
- Tester divers cas de figure
- Mettre én évidence les comportements associés via des captures de trames commentées

# Bibliographie

- « Linux : Solutions haute disponibilité » de S. Rohaut, éditions ENI : un bon livre qui présente les concepts et qui est très détaillé sur la partie pratique
- <http://www.keepalived.org> : une section documentation utile quoique un peu ancienne
- Keepalived userguide :  
<http://www.keepalived.org/pdf/UserGuide.pdf>