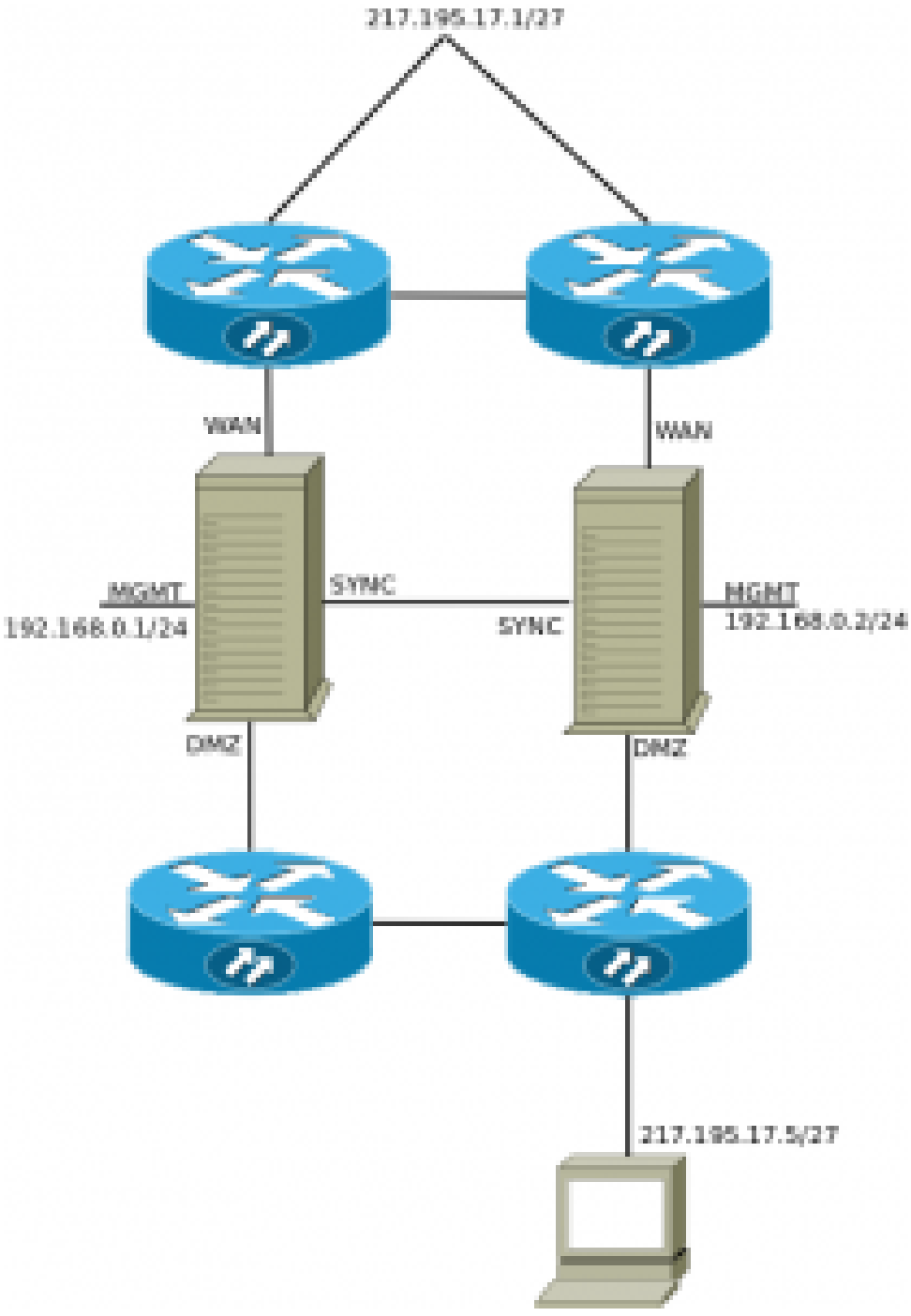
pfSense est une distribution basée sur FreeBSD, qui permet de transformer n'importe quel PC x86 en firewall.
Je me suis mis en tête d'installer et de configurer deux serveurs Dell PowerEdge R200 avec chacun 8Go de RAM et 4 cartes réseaux en firewall transparent redondant.
Ca doit ressembler à ça.



Dans ma configuration, et pour plus de clareté les interfaces pfSense sont renommées:

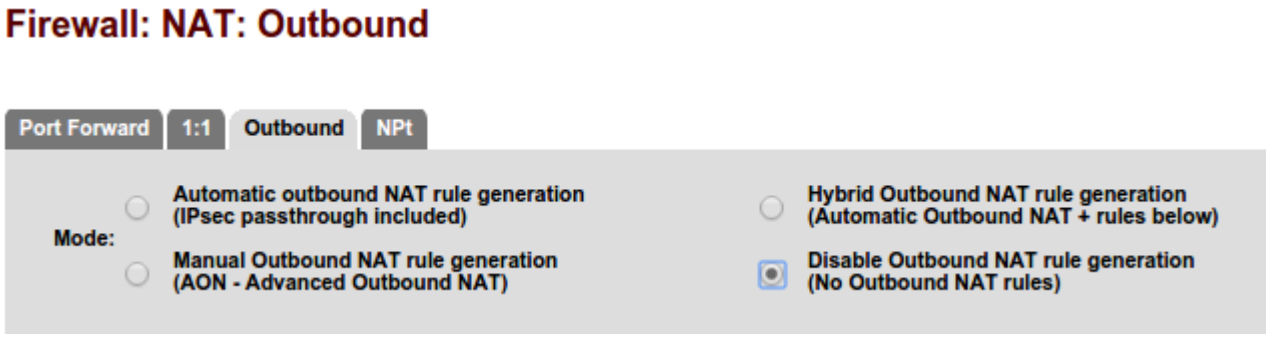
wan = WAN : connectée sur des switchs « backbone »
lan = MGMT : connectée sur un switch d'administration
opt1 = SYNC : connectée à l'autre firewall par un câble croisé
opt2 = DMZ : connectée aux switchs sur lesquels se trouvent les serveurs de vos clients
opt3 = BRIDGE (non représentée sur le schéma)
L'avantage d'un firewall en mode transparent c'est qu'il permet de fournir un adressage public à vos clients, mais il peut aussi s'intercaler dans un réseau existant sans avoir à renuméroter toutes les machines du client comme dans un NAT.
L'installation de pfSense est des plus simples à partir de l'ISO (download here).
Ensuite il faut procéder par étapes :

1. Définitions de l'IP de l'interface « lan » de pfSense

Opération réalisée à travers la console une fois l'installation terminée.
Option 1 : attribution des cartes réseaux
Option 2 : définition des IP des cartes réseaux
Je ne configure que la patte « lan » pour accéder à l'interface d'administration.

2. Modification du comportement par défaut

pfSense est une firewall qui NAT. La configuration d'origine est faite en ce sens.
Il faut donc procéder à quelques modifications pour avoir un fonctionnement optimal en mode transparent.
Aller dans **Firewall -> NAT -> Outbound**, et cocher : **Disable Outbound NAT rule generation (No Outbound NAT rules)**



3. Assignation et activation des autres cartes réseaux
Interfaces -> (assign)

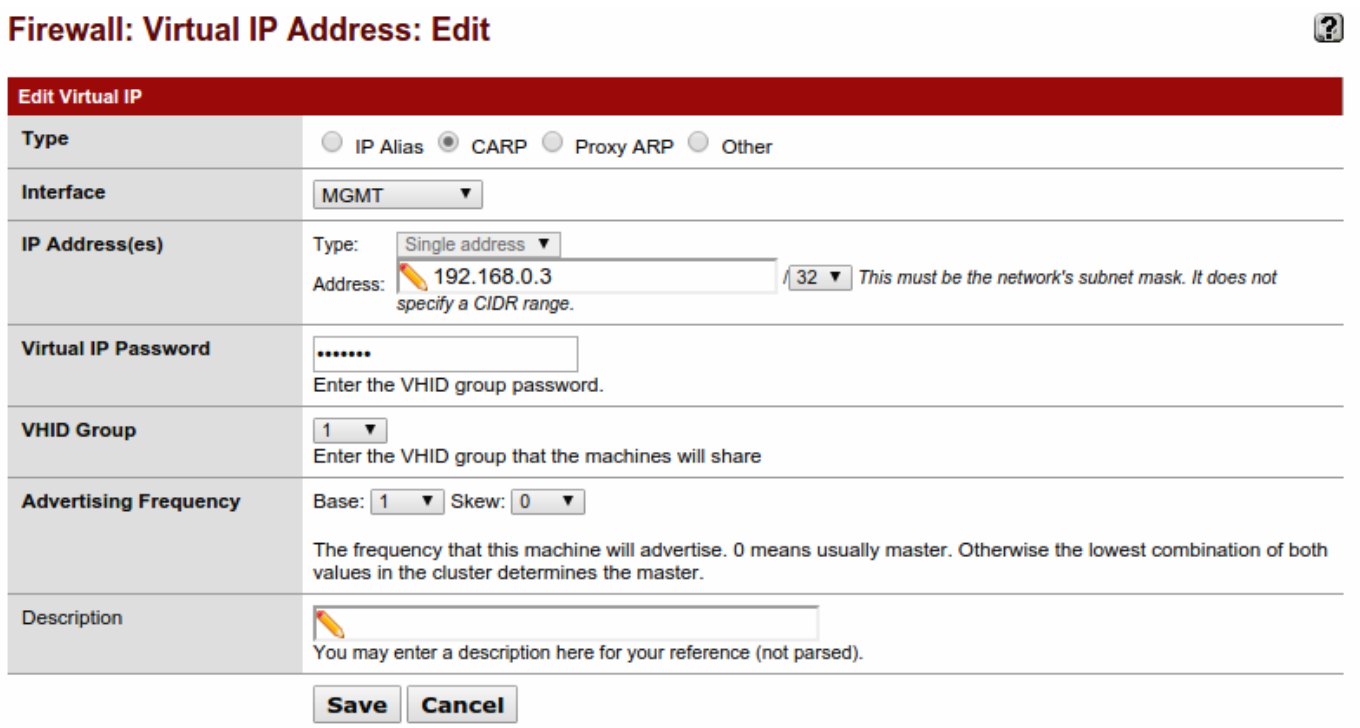
Toutes les interfaces sont activées.
Seules les interfaces lan(MGMT) et opt1(SYNC) auront une adresse IP.
L'interface SYNC servira à synchroniser les 2 firewalls.

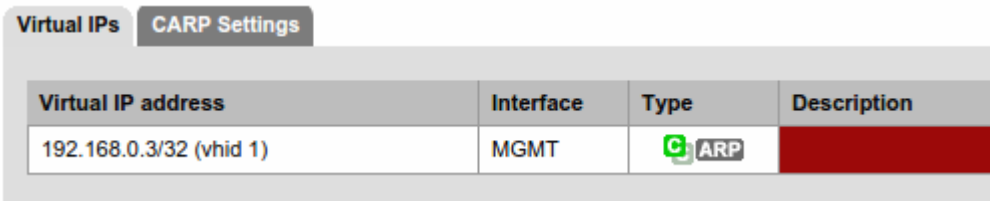
4. Mise en place de la redondance

Nous allons configurer CARP. Pour que CARP est un rôle à jouer dans notre plateforme cible, il faudra assigner une IP (VIP) à l'interface CARP. Ce sera le seul objectif car nous ne nous servirons pas de cette IP. Dans une configuration en mode bridge ce seront les interfaces MGMT qui porteront cette VIP.

4.1. Création de l'adresse IP virtuelle
Firewall -> Virtual IP Addresses

Créer une Virtual IP de type CARP, portée par l'interface MGMT et qui aura une IP en /32.





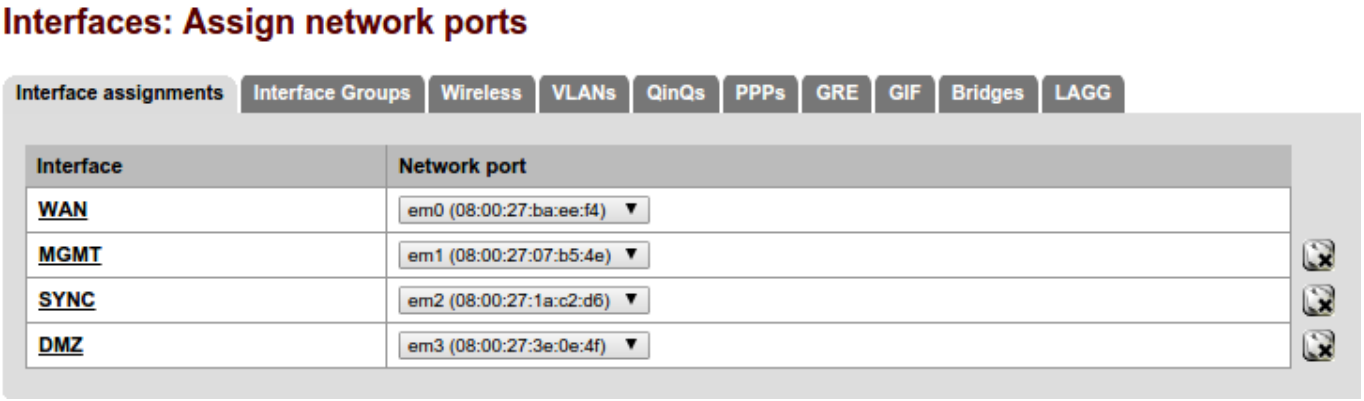
4.2. Configurer la synchronisation.
System: High Availability Sync ou cliquer sur **« CARP Settings »**
State Synchronization Settings (pfsync)
Synchronize States activé
Synchronize Interface SYNC
pfsync Synchronize Peer IP10.10.10.2
Configuration Synchronization Settings (XMLRPC Sync)
Synchronize Config to IP 10.10.10.2
Remote System Username admin
Remote System Password *******
Synchronize Users and Groups coché
Synchronize rules coché
Synchronize Firewall Schedulescoché
Synchronize aliases coché
Synchronize Static Routes coché
Synchronize Virtual IPs coché

Pour le firewall SLAVE, ne configurer que la section du haut en mettant l'IP du MASTER dans le champ « pfsync Synchronize Peer IP ».

5. Création du bridge

5.1 Création des interfaces WAN et DMZ

Nous allons activé ces interfaces sans leur donner d'adresse IP.



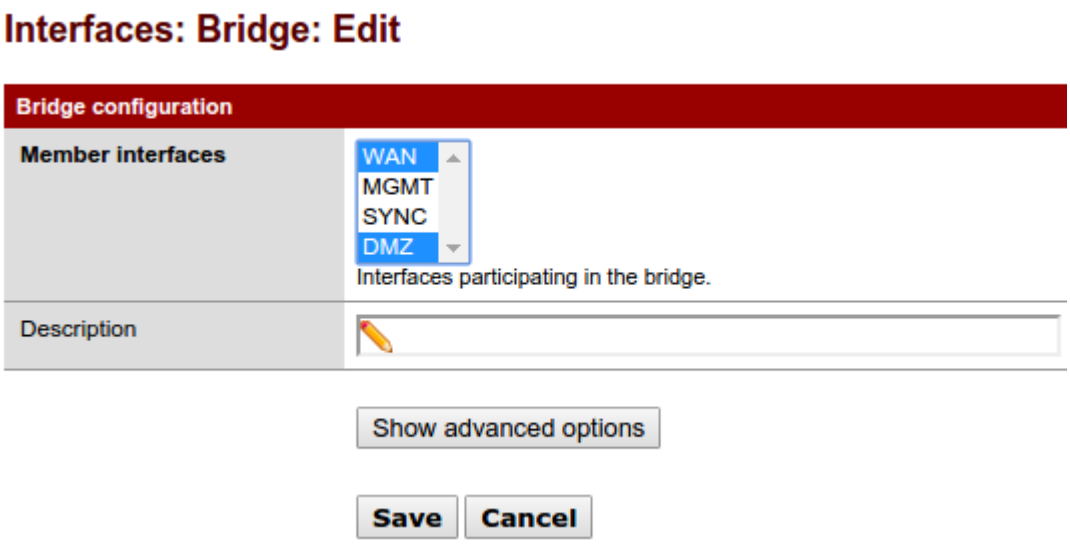
5.2. Création du bridge

Le bridge va assembler 2 interfaces pour n'en faire qu'une seule.
Ainsi tout ce qui arrive sur l'interface WAN sera « reproduit » sur l'interface DMZ.

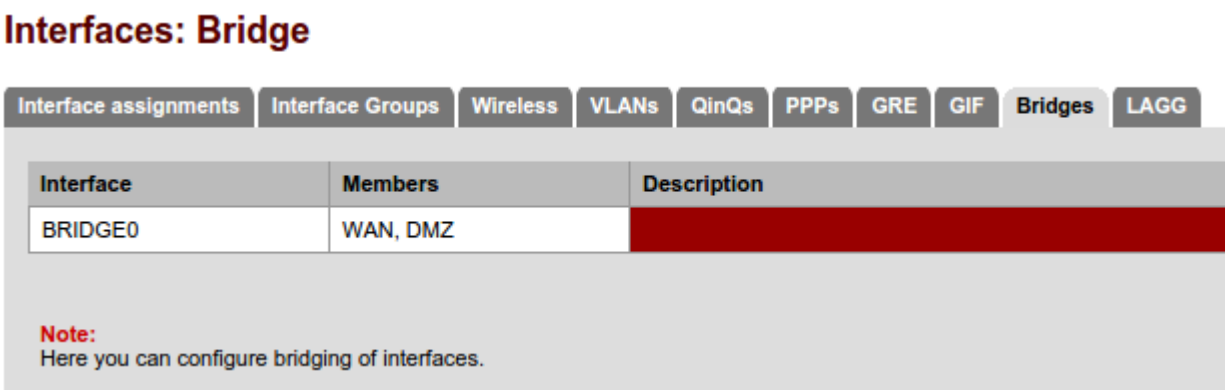
Dans notre installation il est très important de ne pas connecter physiquement l'interface WAN du deuxième firewall (SLAVE) sur le switch « backbone ».
Si vous connecter les 2 interfaces WAN sur le même switch, alors que les 2 interfaces DMZ sont connectées sur une autre switch vous aller créer une boucle de switching et déclencher une tempête de brodacast qui mettra votre réseau à terre.
Le plus vicieux dans les effets d'une boucle de switching peuvent mettre quelques secondes, voir minutes à se manifester après que vous ayez brancher le câble de trop.

Interfaces -> (assign) -> Bridges

Le bridge sera créé avec les interfaces WAN et DMZ.

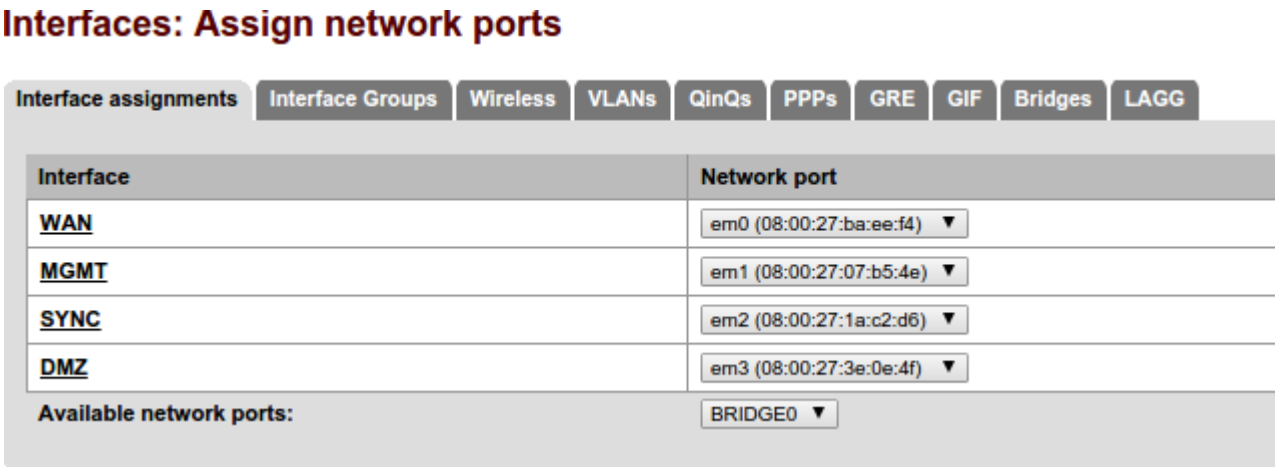


Cela donne ceci dans l'interface WEB.

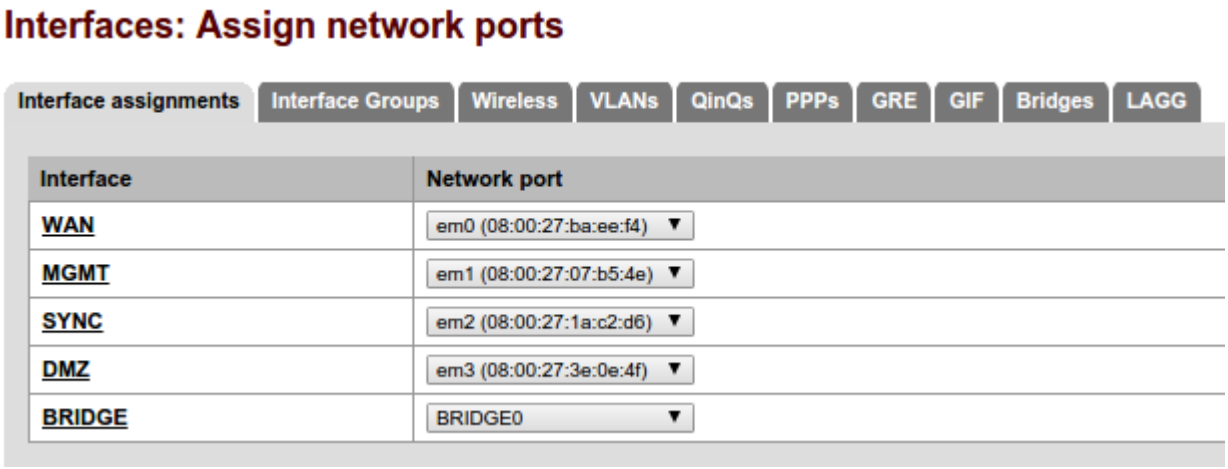


5.3. Activation de l'interface BRIDGE

Cette étape est important car elle nous permettra de configurer UP ou DOWN l'interface BRIDGE, et lutter contre les problèmes de « spanning tree » évoqués plus haut.



On clique sur le « plus » pour créer puis activer l'interface.



5.4. Activer le filtrage au niveau du bridge
System -> Advanced -> System Tunables
Par défaut le filtrage s'opérera sur les interfaces WAN et DMZ, et pas sur le bridge à proprement parlé.
Nous allons donc modifier les propriétés suivantes :
net.link.bridge.pfil_onlyip = 0
net.link.bridge.pfil_member = 1
net.link.bridge.pfil_bridge = 0
6. Quelques paramètres supplémentaires
System -> Advanced -> Firewall and NAT -> Firewall Advanced
Cocher ou choisir les options suivantes :
Disable reply-to on WAN rules
Clear invalid DF bits instead of dropping the packets
Firewall Optimization Options : conservative
Disables the PF scrubbing option which can sometimes interfere with NFS and PPTP traffic.
La première option s'entend car nous avons activé le en mode bridge.
Les suivantes nous préviennent des problèmes de fragmentation de paquets.

System -> Advanced -> Networking -> Network Interfaces
Suppress ARP messages
7. Lutte contre le Spanning Tree
Dans notre configuration en mode bridge, si nous branchons les 4 interfaces WAN et DMZ, nous allons créer une boucle de switching et nous faire pourrir par notre admin réseau préféré.
En partant du principe que cet admin ne voudra par activer ni le Spanning Tree (SPT) ni le Rapid Spanning Tree (RSPT) sur les switchs, nous devons nous assurer que nous n'allond pas créer de boucle de switching en faisant tomber l'interface BRIDGE sur le firewall BACKUP, et en la remontant quand celui-ci devient MASTER suite à une défaillance du firewall MASTER.
Pour faire cela nous allons faire appel à devd
Ce daemon scrute ce qui se passe sur le système et en fonction d'événements va déclencher des actions.
En lançant devd en mode debug (/sbin/devd -d) sur le BACKUP, on observe les événements suivants quand le firewall MASTER est éteint.

```
setting system=CARP
setting subsystem=1@em2
setting type=MASTER
Processing notify event
Testing system=CARP against ^CARP$, invert=0
Testing type=MASTER against ^(MASTER|BACKUP)$, invert=0
Popping table
```

Nous allons donc créer le fichier /usr/local/etc/devd/carp.conf dans lequel nous allons décrire l'événement « l'interface CARP change d'état » et nous allons l'associer à un script.

```
#
# Processing event '!system=CARP subsystem=1@em2 type=BACKUP'
#
notify 200 {
    match "system" "CARP";
#   match "subsystem" "1@em2";
#   match "subsystem" "[0-9]+@[0-9a-z]+";
    match "type" "(MASTER|BACKUP)";
    action "/usr/local/bin/bridge-carp $type";
};
```

Le script « /usr/local/bin/bridge-carp »

```
#!/bin/sh

if [ $# -eq 0 ]
then
    echo "No arguments supplied"
    exit 0
fi

case "$1" in
    MASTER)
        /sbin/ifconfig bridge0 up
        ;;
    BACKUP)
```

```
        /sbin/ifconfig bridge0 down
        ;;
    *)
        /sbin/ifconfig bridge0 down
        ;;
esac
```

Si l'interface CARP est en état MASTER, je monte le bridge.
Si l'interface CARP est en état BACKUP, je descend le bridge.
Dans les logs on verra :

```
setting system=CARP
setting subsystem=1@em2
setting type=MASTER
Processing notify event
Testing system=CARP against ^CARP$, invert=0
Testing type=MASTER against ^(MASTER|BACKUP)$, invert=0
Executing '/usr/local/bin/bridge-carp MASTER'
Popping table
```

Ainsi on espère ne pas créer de boucle de switching.
A noter les options de carp dans le sysctl du système.
L'option « net.inet.carp.preempt » nous assure que toutes les cartes tombent en cas de défaillance d'une interface.
Mais aussi du retour au mode nominale lorsque que le MASTER revient.

```
# sysctl -a|grep net.inet.carp
net.inet.carp.allow: 1
net.inet.carp.preempt: 1
net.inet.carp.log: 1
net.inet.carp.demotion: 0
net.inet.carp.senderr_demotion_factor: 0
net.inet.carp.ifdown_demotion_factor: 240
```

7. Inspiration

Setting up pfSense as a Stateful Bridging Firewall with commodity hardware by Diggory Gray
Transparent Firewall/Filtering Bridge – pfSense 2.0.2 by William Tarrh
Getting pfsense to failover with a bridge using the CD-ROM platform
pfSense bridge gateway vmware ovh ip failover ripe
and many other

Save as PDF