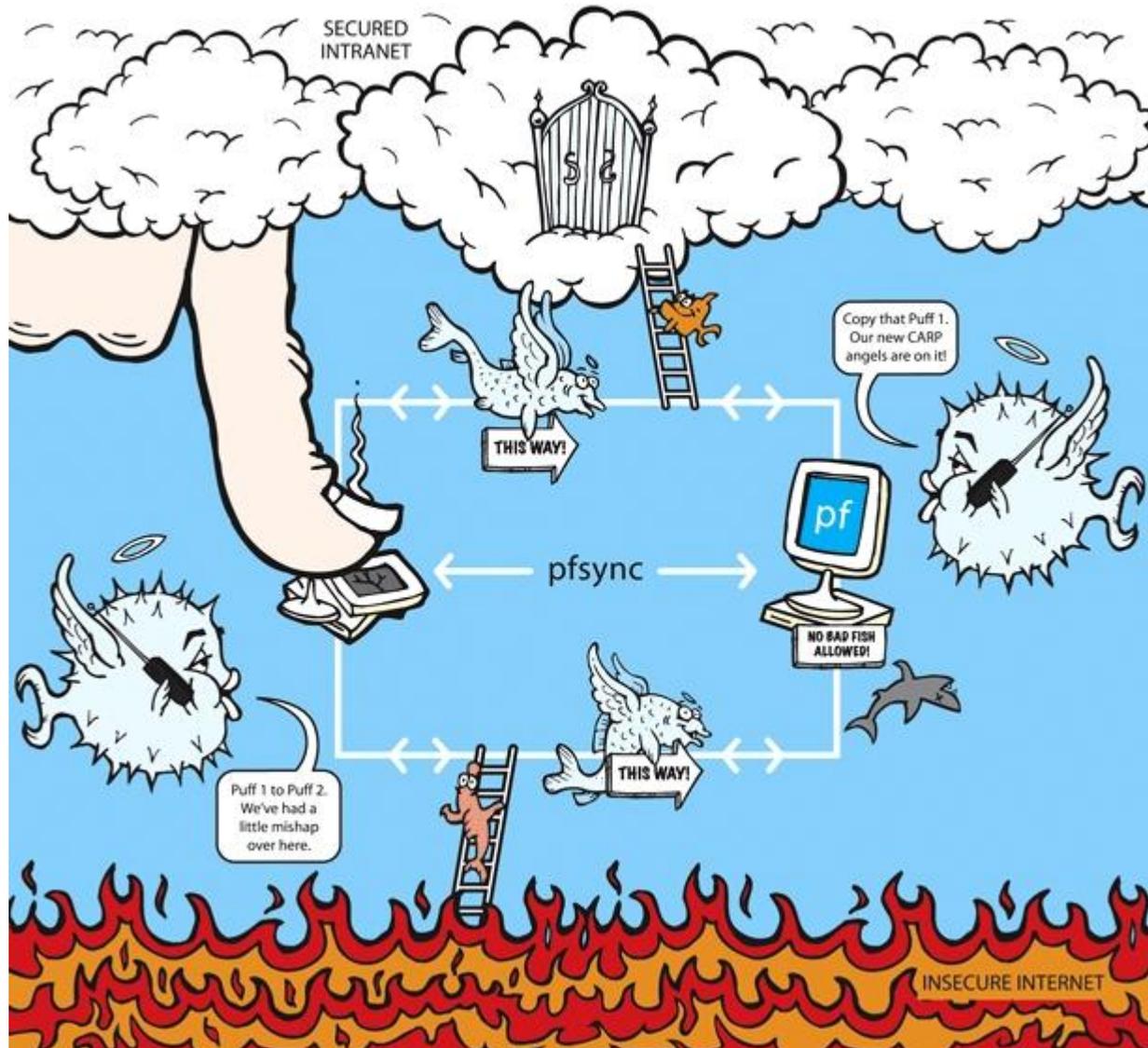


# Firewall failover e Active-Active in OpenBSD



# Firewall ridondati

Nel mercato, non sono molti i vendor che prevedono la cosiddetta funzionalità di HA (High Availability) nei loro sistemi.

Questa, di solito, si basa su di un set di macchine che sono in grado di compensare failure di tipo hardware/software.

Nel caso di firewall, i prodotti che hanno questa funzionalità, sono ancor meno.

# Firewall ridondati

Il problema principale nei firewall è che questi si basano sullo stato delle connessioni.

Vi sarà quindi la necessità di trasferire le connessioni dal firewall attivo a quello/i in standby.

Se così non fosse, quando il firewall attivo passasse allo stato di fault, ci si troverebbe nella situazione in cui il firewall apparirebbe 'appena acceso', ovvero non avrebbe conoscenza di nessuna comunicazione in atto.

# La storia

Il team di OpenBSD si è ispirato ad un prodotto esistente:

VRRP

VRRP (Virtual Router Redundancy Protocol) è un protocollo proprietario di cisco, IBM e NOKIA. Ha come numero IANA 112.

OpenBSD ha implementato CARP (Common Address Redundancy Protocol).

# La storia

CARP ricorre all'utilizzo di un indirizzo IP, o di un MAC address, condiviso tra più macchine.

Questo indirizzo verrà mappato sull'interfaccia *carp* ed essa indicherà verso quale firewall il traffico dovrà passare.

CARP fa la sua comparsa con OpenBSD 3.5.

CARP, al contrario di VRRP, non ha un protocol number assegnato. Il team ha deciso di utilizzare il numero che è di VRRP (112).

# La storia

CARP permette a pf di ricorrere ad un indirizzo 'virtuale' condiviso tra più macchine.

Tutto ciò, non è sufficiente a garantire un passaggio tra un firewall attivo ed uno in stand-by, in maniera semplice ed efficace.

Per tale scopo ci viene in aiuto *pfsync*, la cui funzione è quella di notificare i firewall non attivi del cambiamento di stato delle sessioni (vi ricordo che pf è uno stateful firewall).

# CARP

Vediamo di capire più in dettaglio.

CARP è uno strumento che di basa sul livello OSI 2 e 3.

L'interfaccia virtuale ha un IP assegnato ed un relativo MAC address (anch'esso virtuale) che può esser preso da una macchina fisica in qualsivoglia momento.

La macchina, che è in stato di master, invierà delle notifiche in rete per mezzo del protocollo 112 , in modo che gli altri host avvertano la sua presenza e rimangano quindi in uno stato di standby.

# CARP

La frequenza di questi messaggi è creata dalla seguente formula:

$$\text{advbase} + (\text{advskew} / 255)$$

*advbase* e *advskew* sono parametri di configurazione dell'interfaccia CARP.

Pertanto avremo che, nel caso di failure del nodo master, l'host con la frequenza più elevata prenderà il controllo dell'interfaccia carp e si annuncerà come master.

Di norma, un host si muta in master allorquando esso non abbia più ricevuto 3 notifiche successive dal master.

# CARP

Di norma, CARP invia i pacchetti di update in crittografati con SHA1-MAC per mezzo di una password. Questo fa sì che gli aggiornamenti siano protetti contro attacchi.

Per configurare una interfaccia carp dovremmo dare i seguenti comandi:

```
ifconfig carp0 inet 192.168.1.1 255.255.255.0 vhid 2  
[advskew 0 advbase 1] carpdev iface pass p@ssw0rd
```

# pfsync

*pfsync* fa la sua comparsa con OpenBSD 3.3.

Nasce per notificare, via multicast, altre macchine pf-based sui cambi dello stato. Questo protocollo, assieme a BGP e OSPF, vuole rappresentare un primo approccio alla ridondanza di OpenBSD come router.

pfsync cerca di non mandare notifiche per ogni cambio di sessione, ma li colleziona e li invia come set di tipo omogeneo (ovvero invia set di pacchetti contenenti variazioni dello stesso tipo).

# pfsync

pfsync, al contrario di CARP, non genera pacchetti protetti. Se è necessaria una connessione protetta tra le macchine coinvolte negli update di pfsync, potremmo utilizzare un tunnel IPsec e veicolare il traffico pfsync su questo tunnel.

Questo è possibile grazie anche alla natura di pseudo-device di pfsync che richiede una interfaccia su cui inviare il messaggio in multicast.

pfsync, di default, invia messaggi al gruppo multicast 224.0.0.240, ma può essere configurato per mandare notifiche ad uno specifico host.

# pfsync

Per configurare pfsync usremo la seguente sintassi:

```
ifconfig pfsync0 syncif <interfaccia>
```

```
ifconfig pfsync0 up
```

# pfsync

## Una nota sulle versioni di pfsync

*pfsync è stato sviluppato essenzialmente da David Gwynne che, nel tempo, ha cercato e di ottimizzare il codice e le performance. Questo ha prodotto, fino ad oggi, 5 versioni di pfsync.*

*Fino alla 4 queste erano essenzialmente compatibili con le precedenti. In OpenBSD 4.6 è stata introdotta la versione 5 che risulta essere incompatibile con le precedenti, essendo quest'ultima basata su di un nuovo approccio al problema.*

*Da notare, che la versione 5 non è stata testata in maniera completa: non è garantita la comunicazione tra piattaforme diverse.*

# pfsync

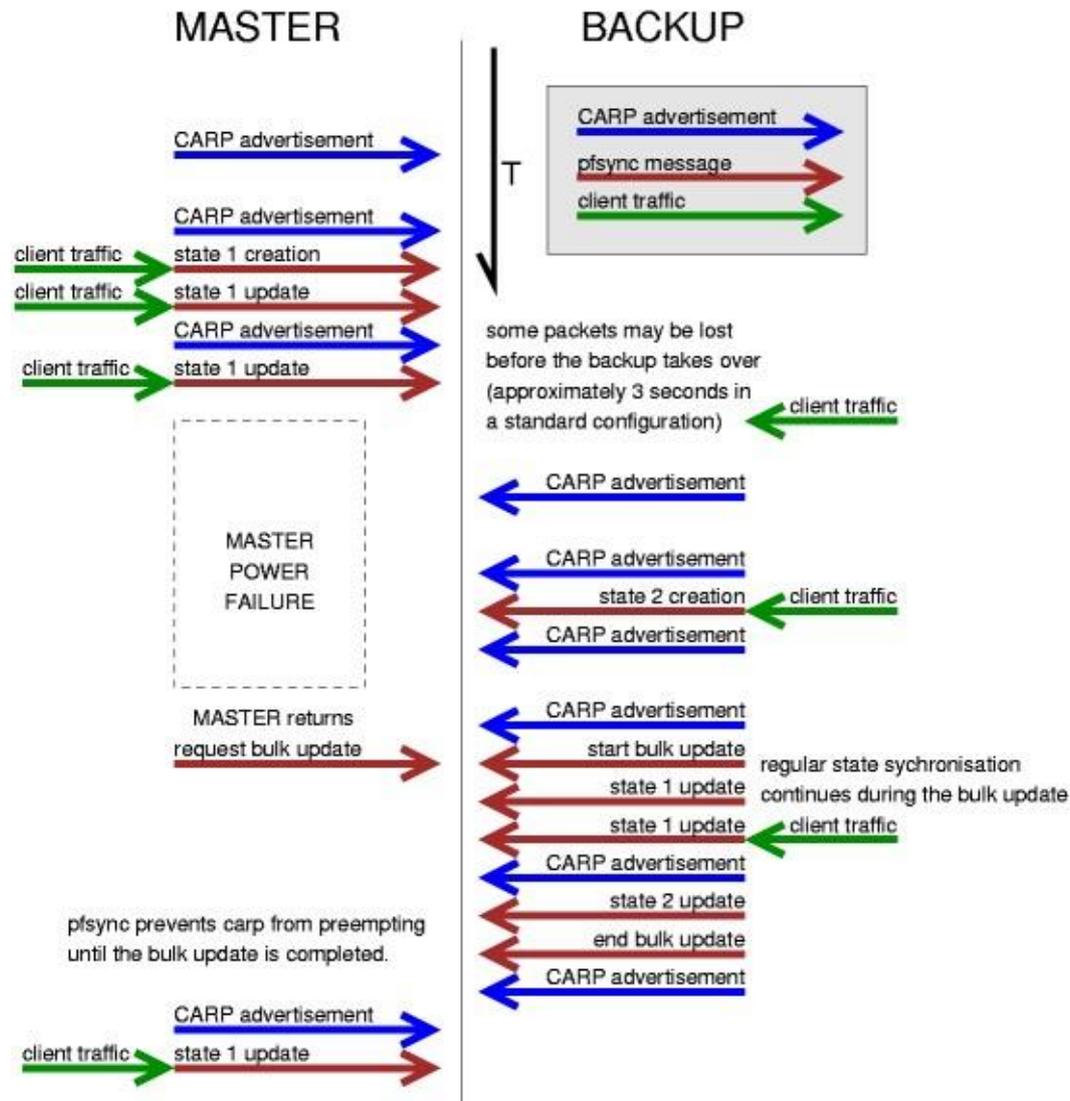
## Una nota sulle versioni di pfsync

*Ciò che essenzialmente è cambiato è il modo in cui pfsync invia i messaggi.*

*Fino alla versione 4 questi erano solo di tipo omogeneo. Si avevano quindi pacchetti che notificavano di nuovi stati, o di chiusura di stati attivi, o di cambio di stato.*

*A partire dalla versione 5, per poter avere insieme di firewall in stato active-active, ne è stata cambiata la struttura. Ora il pacchetto contiene le specifiche su quali tipi di aggiornamento saranno comunicati. Si avrà quindi un set di insert, di remove, update e quant'altro.*

# How it works



L'immagine ci dà una idea di quello che succede.

Sul master transita del traffico che crea degli stati.

Essi sono notificati al backup per mezzo degli update via pfsync.

Quando il master fallisce, la macchina di backup (dopo non aver ricevuto 3 notifiche dal master) prende il controllo ed inizia a notificare i nuovi cambiamenti di stato.

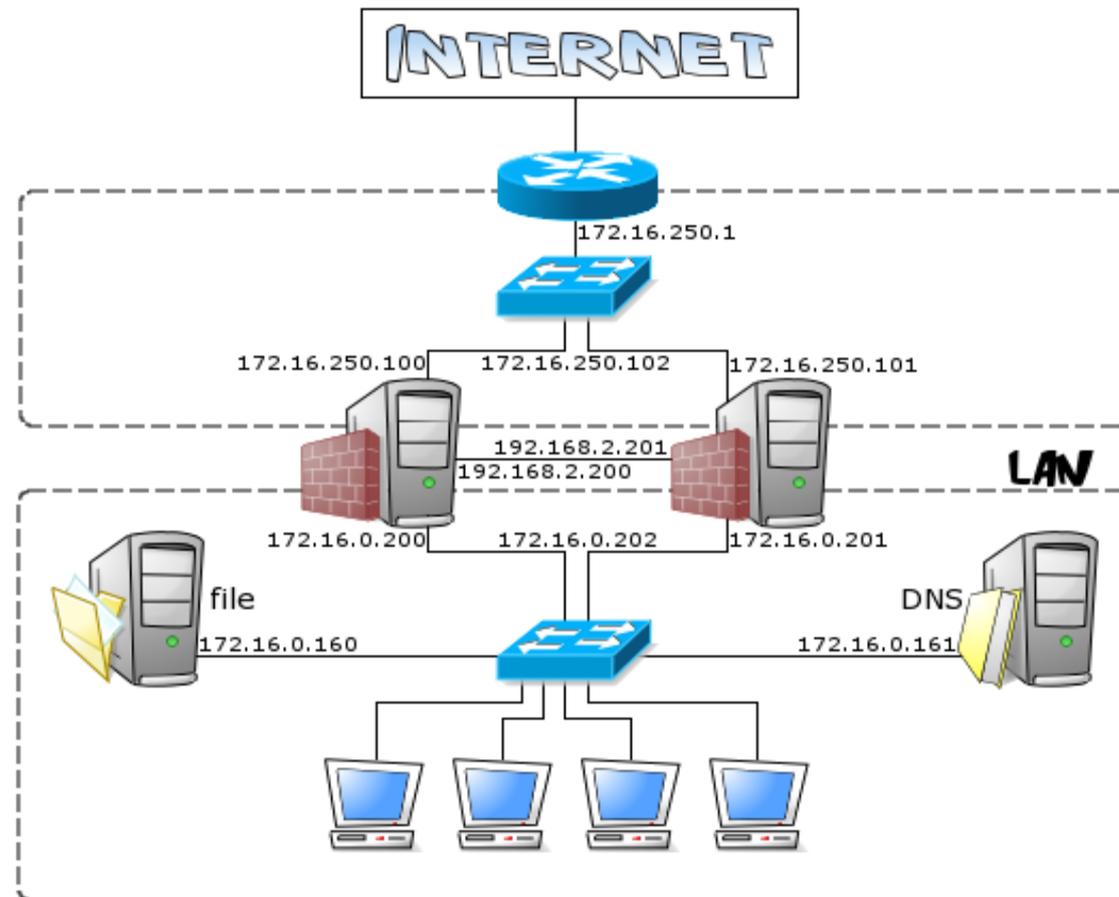
Una volta che il master sia tornato attivo, esso chiederà un bulk update, fino a che il suo stato non sia allineato.

Solo a tal punto egli tornerà ad essere master ed attivo.

# How it works

Fino ad ora si è parlato e di CARP e di pfsync.

Vediamo ora di capire come funzionano le cose:



Questa è l'architettura che vogliamo implementare.

Ci serviranno quindi:

2 pc con (almeno) 3 schede di rete

2 switch (uno lato LAN ed uno WAN/Internet)

1 cavo cross (per collegare le 2 interfacce pfsync)

Iniziamo con le configurazioni:

*Master (firewall\_1)*

```
carp0: inet 172.16.250.1/24 vhid 2 carpdev fxp0 \  
      pass interna
```

```
carp1: inet 172.16.0.202/24 vhid 1 carpdev fxp1 \  
      pass esterna
```

```
rl0: inet 192.168.2.200/24
```

```
fxp0: inet 172.16.250.100/24
```

```
fxp1: inet 172.16.2.200/24
```

```
pfsync0: up syncif rl0
```

## *Backup (firewall\_2)*

carp0: inet 172.16.250.1/24 vhid 2 carpdev fxp0 \

advskew 100 pass interna

carp1: inet 172.16.0.202/24 vhid 1 carpdev fxp1 \

advskew 100 pass esterna

rl0: inet 192.168.2.201/24

fxp0: inet 172.16.250.101/24

fxp1: inet 172.16.2.201/24

pfsync0: up syncif rl0

pf.conf (per ambo le macchine)

*ext\_if="fxp1"*

*int\_if="fxp0"*

*pfsync\_if="rl0"*

*pass quick on lo*

*pass quick on \$pfsync\_if proto pfsync keep state (no-sync)*

*pass quick on { \$ext\_if \$int\_if } proto carp keep state (no-sync)*

*pass #Lo utilizziamo per verificare il semplice transito. Non andiamo, in tal caso, ad applicare regole di filtering*

## Variabili di sistema

*net.inet.ip.forwarding=1*

*net.inet.carp.preempt=1*

*net.inet.carp.allow=1*

***TESTING***

---

# Active-Active

Per poter passare ad un configurazione active-active stabile, dobbiamo tenere in considerazione che gli switch a monte ed a valle del nostro set di firewall DOVRANNO supportare il protocollo STP (o si rischia di bloccare lo switch).

# Sulle interfacce carp dovremmo variare la configurazione

*Per un bilanciamento a livello2*

*ifconfig carpN **balancing arp carpnodes** vhidA:advsekwA, vhidB:advsekwB*

*Per un bilanciamento a livello3*

*ifconfig carpN **balancing ip carpnodes** 1:0,2:100*

# Credits

Il materiale utilizzato per queste slides è nato dai talk di Florin Iamandi e Fabio Cazzin nonché dalla consultazione di:

*OpenBSD.org*

*www.youtube.com/watch?v=XvJAV9qI14Q*

*calomel.org*

*www.countersiege.com/doc/pfsync-carp/*

*www.kernel-panic.it/openbsd/carp/carp1.htm*

*L'immagine del setup di rete è presa (adattandola) da*

*www.kernel-panic.it*