Failover Firewalls with OpenBSD and CARP

Jason Dixon
DixonGroup Consulting

August 4, 2005

Introduction

- Firewalls are a mandatory network component

Introduction

- Firewalls are a mandatory network component
- Should be both a guardian and a guide

Introduction

- Firewalls are a mandatory network component
- Should be both a guardian and a guide
- Often a single point of failure

Introduction

- Firewalls are a mandatory network component
- Should be both a guardian and a guide
- Often a single point of failure
- Failover Firewalls are as vital as HA Application clusters

Introduction

- Firewalls are a mandatory network component
- Should be both a guardian and a guide
- Often a single point of failure
- Failover Firewalls are as vital as HA Application clusters
- Chicks dig redundancy

History of OpenBSD

- A leading secure UNIX-like operating system
- Emphasize code robustness and security
- Open licensing is crucial
- OpenBSD Packet Filter (PF) born out of IPFilter license change

PFSYNC Protocol

- OpenBSD team acknowledged need for failover

PFSYNC Protocol

- OpenBSD team acknowledged need for failover
- PFSYNC (IP Protocol 240)

PFSYNC Protocol

- OpenBSD team acknowledged need for failover
- PFSYNC (IP Protocol 240)
- Pfsyncd sends state updates via multicast

PFSYNC Protocol

- OpenBSD team acknowledged need for failover
- PFSYNC (IP Protocol 240)
- Pfsyncd sends state updates via multicast
- Other firewalls will update their own state tables

PFSYNC Protocol

- OpenBSD team acknowledged need for failover
- PFSYNC (IP Protocol 240)
- Pfsyncd sends state updates via multicast
- Other firewalls will update their own state tables
- Synchronized state == graceful failover

Before CARP

- OpenBSD lacked failover mechanism
- Virtual Router Redundancy Protocol (VRRP) assigns virtual gateway between physical routers
- Operates at OSI Layers 2 and 3
- Master/Backup relationship
- VRRP encumbered by Cisco patent

CARP Protocol

- Common Address Redundancy Protocol (IP Protocol 112)

CARP Protocol

- Common Address Redundancy Protocol (IP Protocol 112)
- Addressed the need for a patent-free failover mechanism

CARP Protocol

- Common Address Redundancy Protocol (IP Protocol 112)
- Addressed the need for a patent-free failover mechanism
- Virtual MAC and IP addresses

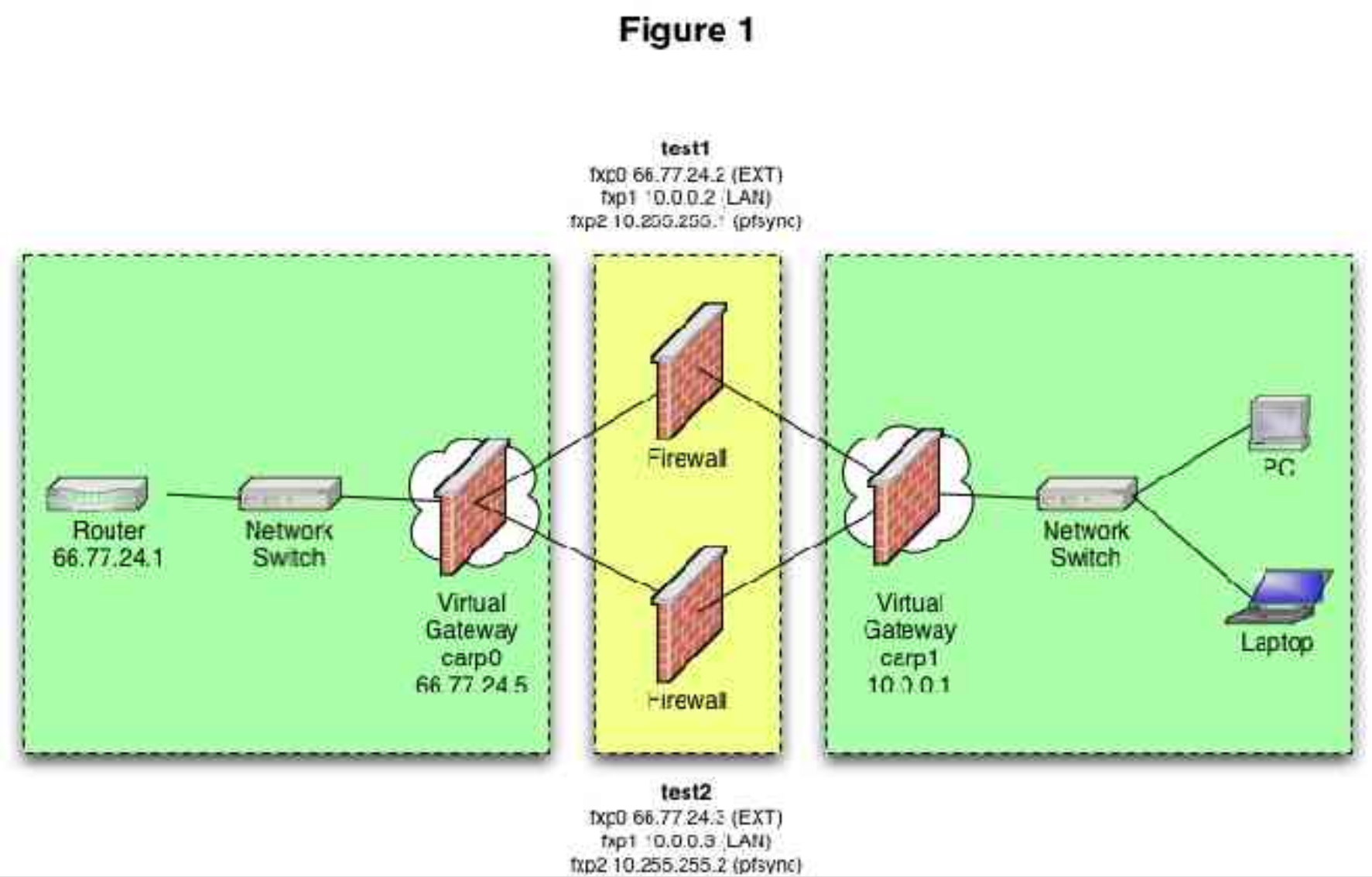
CARP Protocol

- Common Address Redundancy Protocol (IP Protocol 112)
- Addressed the need for a patent-free failover mechanism
- Virtual MAC and IP addresses
- Supports IPv4 and IPv6

CARP Protocol

- Common Address Redundancy Protocol (IP Protocol 112)
- Addressed the need for a patent-free failover mechanism
- Virtual MAC and IP addresses
- Supports IPv4 and IPv6
- Also provides load-balancing, preemption, and crypto hashed announcements

Basic CARP Failover



Basic CARP Example

- Single CARP virtual host on each gateway

Basic CARP Example

- Single CARP virtual host on each gateway
- Virtual host ID (vhid)

Basic CARP Example

- Single CARP virtual host on each gateway
- Virtual host ID (vhid)
- Control frequency of CARP advertisements (advskew)

Basic CARP Example

- Single CARP virtual host on each gateway
- Virtual host ID (vhid)
- Control frequency of CARP advertisements (advskew)
- Authenticate your advertisements (pass foo)

Basic CARP Example (cont'd)

- Auto-recovery (net.inet.carp.preempt)

Basic CARP Example (cont'd)

- Auto-recovery (net.inet.carp.preempt)
- Secure pfsync interface OR peer address (syncpeer) + IPSec

Basic CARP Example (cont'd)

- Auto-recovery (net.inet.carp.preempt)
- Secure pfsync interface OR peer address (syncpeer) + IPSec
- Filter and translate on the physical interface

Basic CARP Example (cont'd)

- Auto-recovery (net.inet.carp.preempt)
- Secure pfsync interface OR peer address (syncpeer) + IPSec
- Filter and translate on the physical interface
- Must allow PFSYNC and CARP protocols

Basic Filtering - pf.conf

```
# Macros
ext_if="em0"
int_if="em1"
pfsync_if="em1"
carp0="192.168.0.5"

# Options
set block-policy return

# Normalization
scrub in no-df

# Translation
nat on $ext_if from $int_if:network \
    to any -> $carp0

# Filters
block in log on $ext_if
pass quick on lo
pass quick on $pfsync_if proto pfsync
pass quick on { $ext_if $int_if } \
    proto carp keep state
pass in on $int_if keep state
pass out on $ext_if keep state
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server1# ifconfig pfsync0 syncdev em1
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server1# ifconfig pfsync0 syncdev em1

server1# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server1# ifconfig pfsync0 syncdev em1

server1# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server1# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server1# ifconfig pfsync0 syncdev em1

server1# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server1# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1

server1# pfctl -nf /etc/pf.conf
```

Basic Setup – server1.oreilly.com

```
server1# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server1# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server1# ifconfig pfsync0 syncdev em1

server1# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server1# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1

server1# pfctl -nf /etc/pf.conf

server1# pfctl -f /etc/pf.conf
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
server2# ifconfig carp1 advskew 100
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
server2# ifconfig carp1 advskew 100

server2# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
server2# ifconfig carp1 advskew 100

server2# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server2# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1
```

Basic Setup – server2.oreilly.com

```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
server2# ifconfig carp1 advskew 100

server2# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server2# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1

server2# pfctl -nf /etc/pf.conf
```

Basic Setup – server2.oreilly.com

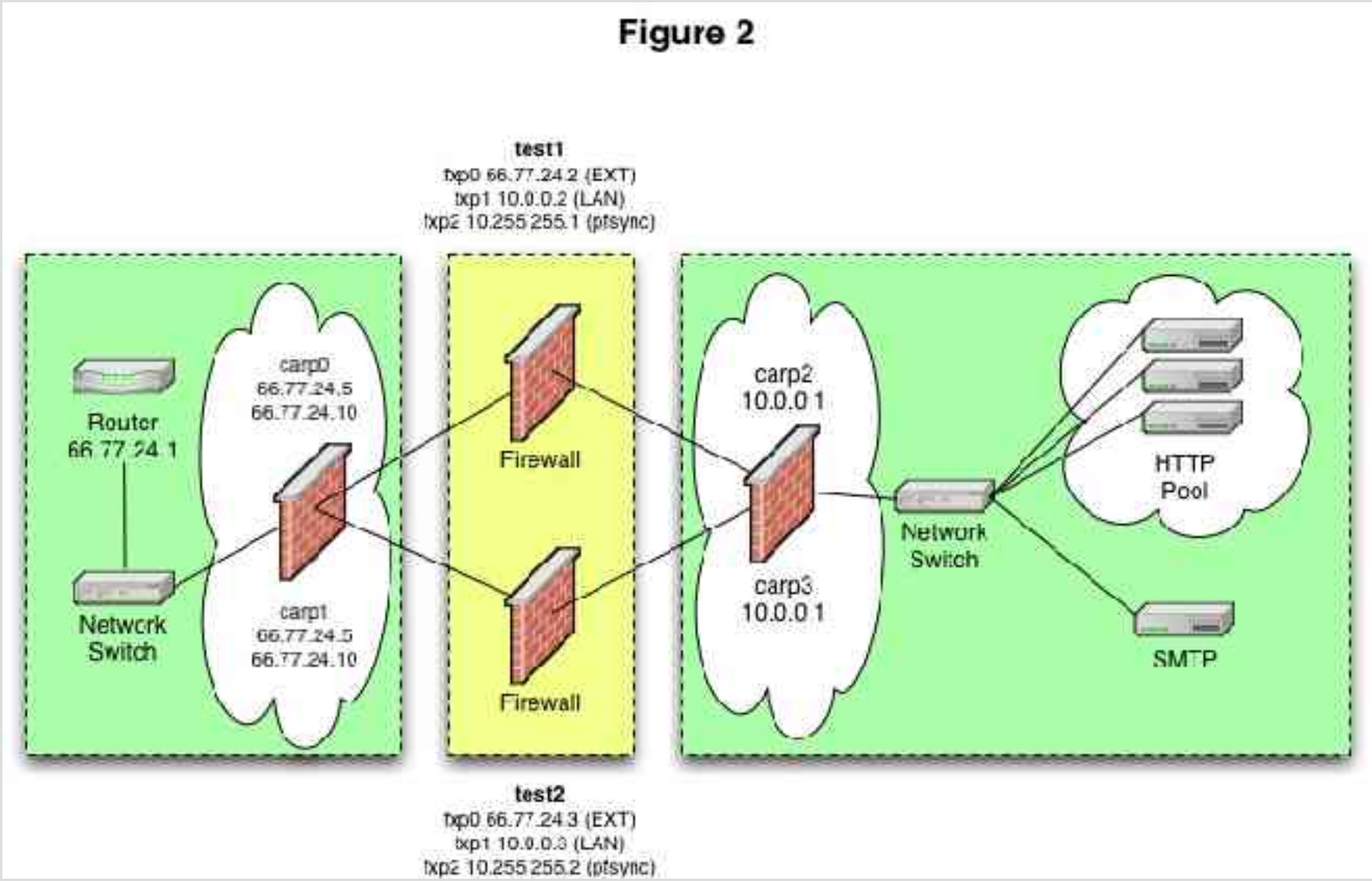
```
server2# ifconfig carp0 192.168.0.5 \
> netmask 255.255.255.0 vhid 1 pass foo

server2# ifconfig carp1 10.0.0.1 \
> netmask 255.255.255.0 vhid 1 pass bar

server2# ifconfig pfsync0 syncdev em1

server2# ifconfig carp0 advskew 100
server2# ifconfig carp1 advskew 100

server2# sysctl -w net.inet.carp.preempt=1
net.inet.carp.preempt 0 -> 1

server2# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding 0 -> 1

server2# pfctl -nf /etc/pf.conf

server2# pfctl -f /etc/pf.conf
```

Demonstration

Basic Example - Troubleshooting

- Tools:
  - Tcpdump, ifconfig, arp, pfctl, vmstat, pftop
- Items to check:
  - CARP, PFSYNC announcements
  - Interface state (MASTER, BACKUP, INIT)
  - LAN client settings (CARP gateway)
- Application tests:
  - Ping
  - Large file transfer (scp/sftp)
  - Multimedia over NFS
- Failover mechanisms:
  - Down CARP interface
  - Unplug ethernet cable
  - Power down CARP member

Advanced Failover with Load-Balancing



Advanced Example

- Multiple CARP virtual hosts on each gateway

Advanced Example

- Multiple CARP virtual hosts on each gateway
- Virtual host ID (vhid) must be unique for each CARP interface on each segment

Advanced Example

- Multiple CARP virtual hosts on each gateway
- Virtual host ID (vhid) must be unique for each CARP interface on each segment
- Each CARP member will serve as MASTER for one CARP interface, and BACKUP as the other

Advanced Example

- Multiple CARP virtual hosts on each gateway
- Virtual host ID (vhid) must be unique for each CARP interface on each segment
- Each CARP member will serve as MASTER for one CARP interface, and BACKUP as the other
- Load-balance at the packet level (net.inet.carp.arpbalance)

Advanced Filtering - pf.conf

```
# Macros
ext_if="em0"
int_if="em1"
pfsync_if="em1"
smtp_ext="192.168.0.10"
smtp_int="10.0.0.110"
http_ext="192.168.0.5"
http_int="10.0.0.105"

# Options
set block-policy drop

# Normalization
scrub in no-df

# Translation
binat on $ext_if from any to $smtp_ext \
    -> $smtp_int
rdr on $ext_if from any to $http_ext \
    port { http, https } \
    -> $http_int source-hash

# ...
```

Advanced Filtering - pf.conf

```
# ...

# Filters
block in log on $ext_if
pass quick on lo
pass quick on $pfsync_if proto pfsync
pass quick on { $ext_if $int_if } \
    proto carp keep state
pass in on $ext_if inet proto tcp \
    from any to $smtp_int port smtp \
    flags S/SA keep state
pass in on $ext_if inet proto tcp \
    from any to $http_int \
    port { http, https } \
    flags S/SA keep state
pass in on $int_if keep state
pass out on $ext_if keep state

# EOF
```

Advanced Setup - server1

```
test1# ifconfig carp0 192.168.0.5 netmask 255.255.255.0 \
> vhid 1 pass foo

test1# ifconfig carp0 alias 192.168.0.10 netmask 255.255.255.0 \
> vhid 1 pass foo

test1# ifconfig carp1 192.168.0.5 netmask 255.255.255.0 \
> vhid 2 advskew 100 pass foo

test1# ifconfig carp1 alias 192.168.0.10 netmask 255.255.255.0 \
> vhid 2 advskew 100 pass foo

test1# ifconfig carp2 10.0.0.1 netmask 255.255.255.0 \
> vhid 1 pass bar

test1# ifconfig carp3 10.0.0.1 netmask 255.255.255.0 \
> vhid 2 advskew 100 pass bar

test1# sysctl -w net.inet.carp.arpbalance=1
net.inet.carp.arpbalance: 0 -> 1
```

Advanced Setup - server2

```
test1# ifconfig carp0 192.168.0.5 netmask 255.255.255.0 \
> vhid 1 advskew 100 pass foo

test1# ifconfig carp0 alias 192.168.0.10 netmask 255.255.255.0 \
> vhid 1 advskew 100 pass foo

test1# ifconfig carp1 192.168.0.5 netmask 255.255.255.0 \
> vhid 2 pass foo

test1# ifconfig carp1 alias 192.168.0.10 netmask 255.255.255.0 \
> vhid 2 pass foo

test1# ifconfig carp2 10.0.0.1 netmask 255.255.255.0 \
> vhid 1 advskew 100 pass bar

test1# ifconfig carp3 10.0.0.1 netmask 255.255.255.0 \
> vhid 2 pass bar

test1# sysctl -w net.inet.carp.arpbalance=1
net.inet.carp.arpbalance: 0 -> 1
```

Summary

- HA Firewall solution
- Commodity hardware
- Available for all BSDs and Linux (uCARP)
- Competes with (or exceeds) functionality of commercial offerings
- Coming soon... sasyncd!!!

References

- OpenBSD -- http://www.openbsd.org/
- PF -- http://www.benzedrine.cx/pf.html
- VRRP RFC 3768 -- http://www.faqs.org/rfcs/rfc3768.html
- OpenBSD FAQ -- http://www.openbsd.org/faq/index.html
- PF User's Guide -- http://www.openbsd.org/faq/pf/index.html
- Userland CARP -- http://www.ucarp.org/

Stateful Failover