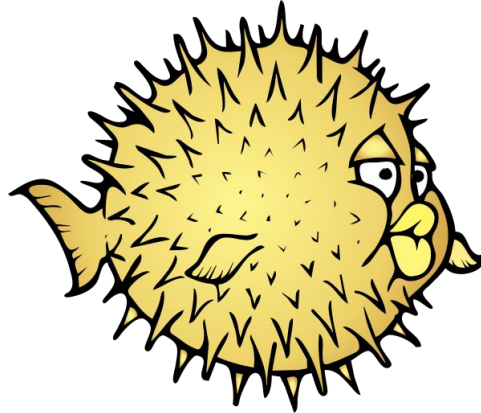


# Failover and NLB

Kuznetsov Sergey

cARP



*Open* **BSD**

sARP

**Немного  
из  
истории**



<https://tools.ietf.org/html/rfc2281>

- *Активный маршрутизатор (Active Router)*
- *Резервный маршрутизатор (Standby Router)*
- *Группа резервирования (Standby Group)*
- *Таймер приветствия (Hello Time)*
- *Таймер удержания (Hold Time)*

# HSRP

## Структура пакета

0		7	15	23
Version	Type	Virtual Rtr ID	Priority	Count IP Adrs
Auth Type		Advet Int	Checksum	
IP Address (1)				
...				
IP Address (n)				
Authentication Data (1)				
Authentication Data (2)				

VRRP (Virtual Router Redundancy Protocol)



# VRRP

- **VRRP-маршрутизатор (VRRP Router)**
- **Виртуальный маршрутизатор (Virtual Router, VR)**
- **Владелец IP-адреса (IP Address Owner)**
- **VRRP-объявление (ADVERTISEMENT)**
- **Виртуальный IP-адрес (Virtual IP address)**
- **Virtual Router Master или VRRP Master router**
- **Virtual Router Backup или VRRP Backup router**
- **Виртуальный MAC-адрес (Virtual MAC) - (0000.5E00.01xx)**

Примеры использования



**КАК ПРАВИЛЬНО  
ЕСТЬ МОЗГ**

CISCO



НЕХОРОШИЙ ЧЕЛОВЕК



# Таймеры VRRP

**Adver Timer** — таймер, который срабатывает для инициации отправки VRRP-объявлений на основании Advertisement Interval

**Master Down Timer** — таймер, который срабатывает когда VRRP-объявление не приходило в течение Master Down Interval

# Описание состояний маршрутизаторов

**Initialize**

**Backup**

**Master**

# Отказоустойчивые службы при помощи CARP



# Введение

## Начало. Инсталяция

1)

```
apt-get install ucarp
```

2)

```
wget download.pureftpd.org/pub/ucarp/ucarp-1.2.tar.bz2  
tar jxf ucarp-1.2.tar.bz2  
cd ucarp-1.2  
./configure  
make  
make install
```

3)

```
apt-get install apache2
```

## Конфигурация UCARP

```
mkdir /etc/ucarp
```

```
touch /etc/ucarp/ucarp.conf
```

```
# Интерфейс на котором будет запущен ucarp
```

```
UCARP_INTERFACE=eth0
```

```
# Виртуальный интерфейс которому будет присвоен виртуальный ip кластера
```

```
UCARP_IF_ALIAS=eth0:0
```

```
# Действительный ip интерфейса. Естественно для каждого узла он будет своим.
```

```
UCARP_SRCIP=172.16.0.11
```

# CARP ID виртуального сервера

UCARP\_VHID=1

# Интервал для сравнения узлов, в секундах.

# Чем он меньше, тем выше приоритет слейва

UCARP\_ADVBASE=1

# Пароль для hmac шифрования соединения (посредством sha1).

UCARP\_PASS=geheim

```
# Виртуальный ip по которому будет доступен кластер
```

```
UCARP_ADDR=172.16.0.1
```

```
# Маска нашего сегмента сети
```

```
UCARP_MASK=255.255.0.0
```

```
# Путь к скриптам которые будут запущены при активации/деактивации узла
```

```
UCARP_UPSCRIPT=/etc/ucarp/ucarp-up.sh
```

```
UCARP_DOWNSCRIPT=/etc/ucarp/ucarp-down.sh
```

**`/etc/ucarp/ucarp.conf:`**

# Скрипты настройки

```
cd /etc/ucarp
```

```
touch ./ucarp-up.sh
```

```
touch ./ucarp-down.sh
```

/etc/ucarp/ucarp-up.sh

```
#!/bin/bash
```

```
source /etc/ucarp/ucarp.conf
```

```
ifconfig $UCARP_IF_ALIAS $UCARP_ADDR netmask $UCARP_NETMASK
```

/etc/ucarp/ucarp-down.sh

```
#!/bin/bash
```

```
source /etc/ucarp/ucarp.conf
```

```
ifconfig $UCARP_IF_ALIAS down
```

```
source /etc/ucarp/ucarp.conf
```

```
ucarp /  
--interface=$UCARP_INTERFACE /  
--srcip=$UCARP_SRCIP /  
--vhid=$UCARP_VHID /  
--pass=$UCARP_PASS /  
--advbase=$UCARP_ADVBASE /  
--preempt /  
--addr=$UCARP_ADDR /  
--daemonize /  
--upscript=$UCARP_UPSCRIPT /  
--downscript=$UCARP_DOWNSCRIPT
```

# Master status

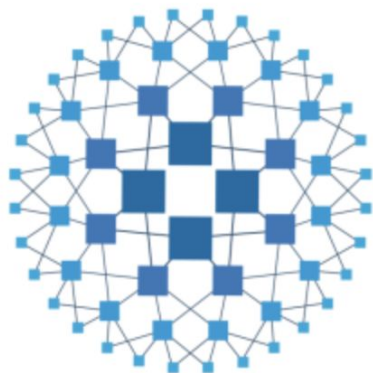
```
Nov 26 03:49:17 vmdebian01 ucarp[2327]: [WARNING] Switching to state: MASTER
```

```
Nov 26 03:49:17 vmdebian01 ucarp[2327]: [WARNING] Spawning [/etc/ucarp/ucarp-up.sh eth0]
```

# Slave status

```
Nov 26 03:50:24 vmdebian02 ucarp[3802]: [WARNING] Switching to state: BACKUP
```

```
Nov 26 03:50:24 vmdebian02 ucarp[3802]: [WARNING] Spawning [/etc/ucarp/ucarp-down.sh eth0]
```



**HAPROXY**

# Список контроля доступа (ACL)

```
acl url home pat beg /home
```

# Backend

```
backend web-backend
    balance roundrobin
    server web1 web1.yourdomain.com:80 check
    server web2 web2.yourdomain.com:80 check
```

```
backend blog-backend
    balance roundrobin
    mode http
    server site1 site1.yourdomain.com:80 check
    server site1 site1.yourdomain.com:80 check
```

# Frontend

Набор ip адресов и портов

ACL

Правила `use_backend`

# Типы распределения нагрузки

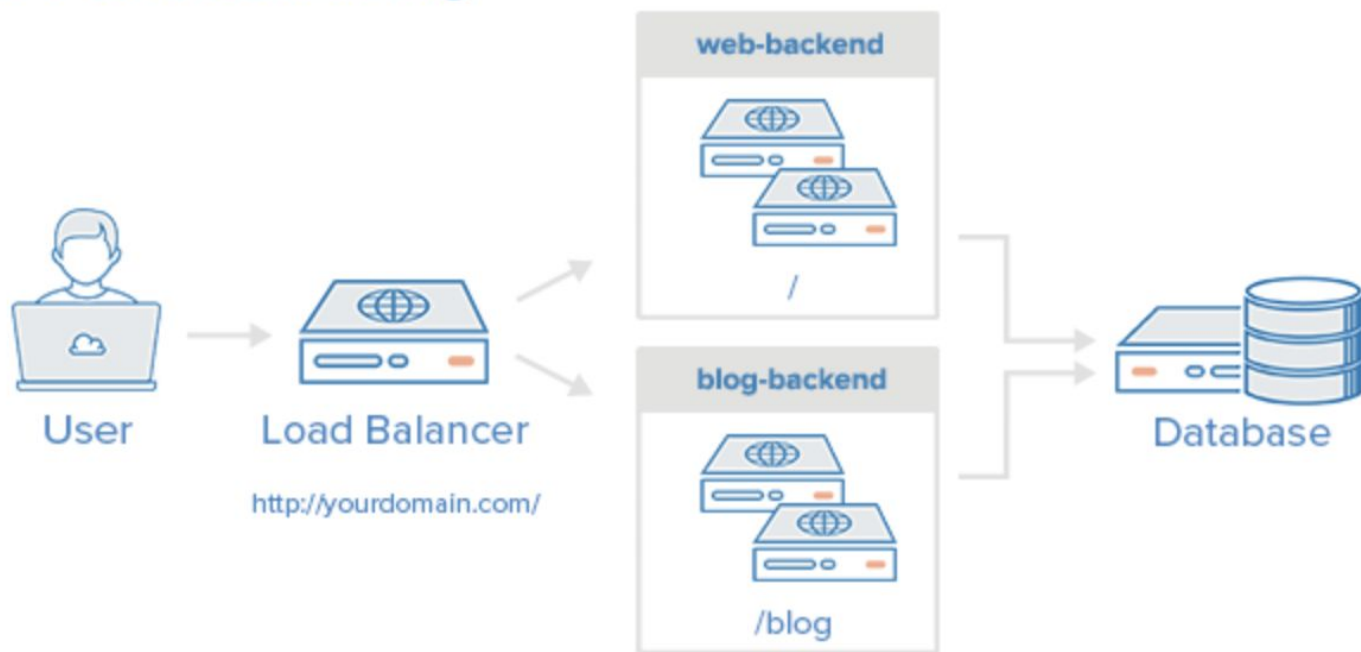
## No Load Balancing



## Layer 4 Load Balancing



## Layer 7 Load Balancing



# Front-end config

```
frontend http
```

```
bind *:80
```

```
mode http
```

```
acl url site path beg /site
```

```
use backend site-backend if url site
```

```
default backend web-backend
```

# Алгоритмы распределения нагрузки

- *roundrobin*
- *leastconn*
- *source*

# Продолжительные сессии

Иногда приложению требуется, чтобы пользователь был подсоединен к одному и тому же серверу.

Это может быть достигнуто путем применения параметра `appsession` для необходимого backend.

# Проверка работоспособности HAProxy

# Orchestrator

Orchestrator — это решение по управлению рабочими процессами в центре обработки данных. Orchestrator позволяет автоматизировать создание, мониторинг и развертывание ресурсов в вашей среде.

