HIGH AVAILABILITY (HA) WITH OPENSIPS

Setting up the HA Environment

Norm Brandinger

textPlus

SIP Transport

- SIP is able to be transmitted using Multiple Protocols such as: UDP, TCP, or TCP with TLS (SSL) encryption
- Standard DNS “A” (Address) record lookups only return an IP address, not a protocol.
- This represents a problem when using DNS to perform lookup functions on SIP addresses.

DNS NAPTR

- NAPTR – Name Authority Pointer
  - Used to map servers (domain names) to user addresses
  - Used to find which services are available for a domain
    - SIP over UDP
    - SIP over TCP
    - SIP over TLS (SSL)
  - Used to determine the name used for a SRV lookup

DNS SRV

- SRV – Service Record
- Used to define a hostname and port number for servers (domain names) associated with a specific service.

For Example

- A call is made to 2125551212@example.com
- Don’t know the IP Address
- Don’t know the Port
- Don’t know the Protocol
- To find out this information we need to perform NAPTR, SRV and A record lookups.

DNS NAPTR Lookup

```
# host -t NAPTR example.com

example.com has NAPTR record 10 100 "S" "SIP+D2U" "" _sip._udp.example.com.
example.com has NAPTR record 20 100 "S" "SIP+D2T" "" _sip._tcp.example.com.
example.com has NAPTR record 30 100 "S" "SIPS+D2T" "" _sips._tcp.example.com.
```

NAPTR shows three ways to contact example.com.

- 10, 20, and 30 are Order with the lower value taking precedence.
- 100 is the Preference and is used to arrange entries with the same Order.
- "S" means that an SRV lookup should be performed against the the NAPTR record.
- "SIP+D2U" is SIP over UDP at _sip._udp.example.com
- "SIP+D2T" is SIP over TCP at _sip._tcp.example.com
- "SIPS+D2T" is SIP over TCP with TLS (SSL) at _sips._tcp_example.com.
- "" is a Regular Expression for replacement
- The last field is the replacement for example.com

DNS SRV Lookup

```
# host -t SRV _sip._udp.example.com
```

```
_sip._udp.example.com has SRV record 10 100 5060 osips01.example.com.
_sip._udp.example.com has SRV record 20 100 5060 osips11.example.com.
```

The SRV result shows two hosts that supply SIP over UDP services for example.com

- 10 and 20 are Priorities with the lower Priority taking precedence.
- 100 is the Weight that is used to order entries with the same Priority.
- 5060 is the Port that should be used on the target host.
- osipsxx.example.com is the Target host.

DNS A Record Lookup

The A (or Address) Record lookup provides us with the IP address of the host.

```
# host -t a osips01.example.com
```

```
osips01.example.com has IP address 192.168.1.213
```

We now have all of the information to route the call 2125551212@example.com to 192.168.1.213:5060 using UDP

In a HA environment it is important to note that 192.168.1.213 is NOT tied to a single machine, but is a floating IP often called a Virtual IP or a VIP.

High Availability (HA) Servers

- Two or more physical servers are required.
- Each of servers is configured with an administrative IP address that is not used by end-users.
- There is a floating IP address that is used by end-users.
- Usually the configuration, activation and deactivation of the floating IP address is performed automatically.
- The floating IP is active on only one of the servers at any time, this server is called the master or primary.

High Availability (HA) Servers

- When the master (or primary) server fails, the backup (or secondary hot-standby) server needs to take over the floating IP address.
- The backup server does this by bringing up an interface for the floating IP address.
- Most of the time, the floating IP is defined to the operating system as an IP alias.
- IP aliasing is the process of associating more than one IP address to an interface.

textPlus

IP Aliasing

Manually start / stop / display an interface with an IP Alias

```
# ifconfig eth0:0 192.168.1.213 up
# ifconfig eth0:0 192.168.1.213 down
# ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:6a:4a:45
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6a:4a45/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5917563 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51837 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:409866657 (390.8 MiB)  TX bytes:3550371 (3.3 MiB)

eth0:0    Link encap:Ethernet  HWaddr 00:0c:29:6a:4a:45
          inet addr:192.168.1.213  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

IP Alias on Debian

```
# vi /etc/network/interfaces

auto eth0:0
iface eth0:0 inet static
name Ethernet alias LAN card
address 192.168.1.213
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0

# /etc/init.d/networking restart
```

IP Alias on CentOS

```
# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0:0
```

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0:0
```

Replace DEVICE=eth0 with DEVICE=eth0:0
Replace IPADDR=xxx.xxx.xxx.xxx with IPADDR=192.168.1.213

Resulting file should look similar to this:

```
DEVICE=eth0:0
IPADDR=192.168.1.213
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
NAME=eth0:0
```

textPlus

Floating IP and ARP

- Once the interface with the IP Alias (of the floating IP) is up, the hot-standby is able to accept traffic on it.
- Address Resolution Protocol (ARP) is used by the hot-standby to answer requests for the floating IP address.
- Note that this doesn't ensure that all traffic will be sent to to the hot-standby machine.

textPlus

Address Resolution Protocol (ARP)

- Typically a host on the LAN broadcasts an ARP request for the hardware or Ethernet MAC address of an IP address.
- When another host on the LAN determines that it's responsible for the IP address it will send an ARP reply with the hardware address of the interface where the IP address is.
- The original host stores the hardware address in its ARP cache so that it doesn't have to send an ARP request for each packet destined to the IP address.
- Entries in the ARP cache typically expire after about two minutes.

ARP and the Master

- Although a master may be inaccessible, it may still be capable of answering ARP requests for the Ethernet MAC of the floating IP address.

- If this happens, then each time a host on the LAN sends out an ARP request it may receive a response from the master that has failed in some way.

- Even if the master doesn't respond to an ARP request, traffic will continue to be sent to it until the ARP cache entries of the other hosts and routers on the LAN expire.

Gratuitous ARP

- Is an ARP reply where there is no ARP request.
- If the ARP reply is sent to the broadcast address then all hosts on the LAN will receive the ARP reply and refresh their ARP cache.
- If Gratuitous ARPs are sent often enough then no host ARP cache entry for the IP address should expire and thus no ARP requests will be sent out.
- This means that there will be no chance that an ARP reply for the failed master will be sent out.

Enable Gratuitous ARP

- Note that Gratuitous ARP can be used to maliciously take over the IP address of a machine. Because of this, some routers and switches ignore or can be configured to ignore Gratuitous ARP.
- Linux machines have Gratuitous ARP disabled by default.

Accept Gratuitous ARP:

```
# echo 1 > /proc/sys/net/ipv4/conf/all/arp_accept
# echo 'net.ipv4.conf.all.arp_accept=1' >> /etc/sysctl.conf
# echo 'net.ipv4.conf.eth0.arp_accept=1' >> /etc/sysctl.conf
```

Ignore Gratuitous ARP:

```
# echo 0 > /proc/sys/net/ipv4/conf/all/arp_accept
# echo 'net.ipv4.conf.all.arp_accept=0' >> /etc/sysctl.conf
# echo 'net.ipv4.conf.eth0.arp_accept=0' >> /etc/sysctl.conf
```

Administrative ARP Tools

Note: ARP traffic is layer2 and is non-routable so it can only be used on a local network.

ARPING - A tool that can be used to discover hosts on a local network (it can be used to send ARP requests or replies)

ARPWATCH - A tool for monitoring ARP traffic on a network.

Moving the Virtual IP Address

There are various tools used to automate the movement of the floating or Virtual IP (or VIP) address from a master to a hot-standby server.

- Corosync
- Heartbeat
- Pacemaker
- KeepaliveD
- VRRPd

Corosync

Corosync manages clusters of servers. It is typically used with Pacemaker to set up clusters.
Requires authkey, corosync.conf

```
# corosync-keygen
```

/etc/corosync/authkey

/etc/corosync/corosync.conf

```
totem {
        version: 2
        secauth: off
        threads: 0
        interface {
                ringnumber: 0
                bindnetaddr: 192.168.1.211 This is the Administrative IP address, NOT the virtual IP address
                mcastaddr: 226.94.1.1
                mcastport: 5405
                ttl: 1
        }
}
```

Pacemaker

Must set up a cluster

```
# ccs -f /etc/cluster/cluster.conf --createcluster osips01
# ccs -f /etc/cluster/cluster.conf --addnode osips01-a
# ccs -f /etc/cluster/cluster.conf --addnode osips01-b
```

```
# crm_mon -1 | grep Online
```

```
Online: [ osips01-a osips01-b ]
```

To Pacemaker, an IP address is a resource (and it can handle many different types of resources. We must now configure the Virtual IP as a resource

Pacemaker on CentOS

CentOS has switched from using crm to pcs (Pacemaker Configuration System) so if you don't have crm, it can be obtained as follows:

```
# wget -P /etc/yum.repos.d/ http://download.opensuse.org/repositories/
network:/ha-clustering/CentOS_CentOS-6/network:ha-clustering.repo

# yum install crmsh.x86_64

# crm configure

# crm(live)configure# primitive osips01 IPaddr params
ip=192.168.1.213 cidr_netmask="255.255.255.0" nic="eth0"
# crm(live)configure# commit
# crm(live)configure# exit
```

textPlus

Pacemaker

The next thing we want to do is tell Pacemaker on which node we'd prefer the Virtual IP to live when both nodes are up.

```
# crm configure
crm(live)configure# location osips01_pref osips01 100: osips01-a.example.com
crm(live)configure# commit
crm(live)configure# exit
```

Set up Pacemaker monitoring

```
# crm configure
crm(live)configure# monitor osips01 40s:20s
crm(live)configure# commit
crm(live)configure# exit
```

textPlus

Heartbeat

Heartbeat requires three files: authkeys, ha.conf, haresources

/etc/ha.d/ha.cf Main configuration file

```
#	What interfaces to broadcast heartbeats over?
bcast   eth0

#	Tell what machines are in the cluster. Must match uname -n
node osips01-a
node osips01-b
```

/etc/ha.d/haresources Resource configuration file

/etc/ha.d/authkeys Authentication information

```
auth 1
1 sha1 5c493d15b6fd61c064be7792a01ce9c7
```

KeepaliveD

Routing software written in C. The goal of the project is to provide simple and rob oust facilities for load balancing and high-availability to Linux systems. Load balancing relies on the Linux IP Virtual Server (IPVS) kernel module while high-availability is achieved by VRRP.

/etc/keepalived/keepalived.conf

```
vrrp_instance VI_1 {
   virtual_router_id 100
   state BACKUP
   interface eth0              # interface for inside_network, bound by vrrp
   garp_master_delay 1         # delay for gratuitous ARP after transition to MASTER in sec
   smtp_alert                  # send an alert when this instance changes state from MASTER to BACKUP
   priority 101                # for electing MASTER, highest priority wins
   advert_int 1                # how often should we vote, in seconds?
   nopreempt                   # VRRP will normally preempt a lower priority
                               # "nopreempt" allows the lower priority # machine to maintain the master role,
                               # even when  a higher priority machine comes back online.
                               # NOTE: For this to work, the initial state of this # entry must be BACKUP.
   authentication {
      auth_type PASS
      auth_pass Secret
   }
   virtual_ipaddress {
      192.168.1.213            # IP addresses that Keepalived will setup and takedown on this machine.
   }
}
```

textPlus

Virtual Router Redundancy Protocol Daemon (VRRPd)

VRRPd is very easy to configure. All configuration is provided by parameters on the command line.
http://sourceforge.net/projects/vrrpd/

Primary:	**vrrpd -i eth0 -p 100 -n -v 1 192.168.1.213**
Backup:	**vrrpd -i eth0 -p 50 -n -v 1 192.168.1.213**

```
Usage: vrrpd -i ifname -v vrid [ -M monitor ] [-s] [-a auth] [-p prio] [-nh] ipaddr

  -h              : display this short inlined help
  -n              : Dont handle the virtual mac address
  -i ifname       : the interface name to run on
  -v vrid         : the id of the virtual server [1-255]
  -s              : Switch the preemption mode (Enabled by default)
  -a auth         : auth=(none|pw/hexkey|ah/hexkey) hexkey=0x[0-9a-fA-F]+
  -p prio         : Set the priority of this host in the virtual server (dfl: 100)
  -d delay        : Set the advertisement interval (in sec) (dfl: 1)
  ipaddr/length   : Should be at the end - IP address(es) of the virtual server and the length of the subnet mask -
  -V              : display version

Supplementary Options:
  -U              : (-U <file>): run <file> to become a master)
  -D              : (-D <file>): run <file> to become a backup)
  -M              : (-M x) Monitoring process and Network (Max 9)
```

Locally run Services - Non Local Bind

- In certain situations, you may want to send packets as if they're coming from a different computer.
- Linux prevents such IP address spoofing by default, because the most well known use is as a malicious spoofing attack.
- In order to be able to bind on a IP which is not yet defined on the system, we need to enable non local binding at the kernel level.
- This setting is required by both OpenSIPS and FreeSWITCH.

```
# echo 1 > /proc/sys/net/ipv4/ip_nonlocal_bind
# echo 'net.ipv4.ip_nonlocal_bind=1' >> /etc/sysctl.conf
```

textPlus

OpenSIPS Listen

OpenSIPS should listen on the floating or Virtual IP Address first

```
# Floating or Virtual IP
listen=udp:192.168.1.213:5060
# Administrative IP
listen=udp:192.168.1.211:5060
```

textPlus

OpenSIPS Registrations

- There are two situations for handling SIP registrations
- When this instance of OpenSIPS is the master all registrations must be stored and then replicated to the backup(s)
- When this instance of OpenSIPS is a backup the registration information from the master must be stored

OpenSIPS Registrations Backup

```
modparam("registrar", "sock_hdr_name", "Sock-Info")

if (is_method("REGISTER")) {

        # We are a BACKUP
        if ($si == "192.168.1.212") {
                if (!save("location", "msE30e15")) {
                        xlog("L_ERR", "ERROR saving replicated reg $ru")
                        sl_reply_error();
                } else {
                        xlog("L_INFO", "Saved replicated registration $ru")
                }
                exit;
        }
…
```

textPlus

OpenSIPS Registrations Primary

```
# We are a PRIMARY
        if (!save("location", "E30e15")) {
                xlog("L_ERR", "ERROR saving registration $ru")
                sl_reply_error()
                exit;
        } else {
                xlog("L_INFO", "Saved registration $ru")
        }

        # Replicate the registration to the BACKUP
        add_sock_hdr("Sock-Info");
        force_send_socket("192.168.1.211");
        t_on_failure("REPLICATE_FAIL")
        t_on_reply("REPLICATE_REPLY")
        t_replicate("192.168.1.212");
        exit;
}
```

OpenSIPS Registrations Replies and Failures

```
onreply_route[REPLICATE_REPLY] {
		xlog("L_INFO", "Replication of registration succeeded $fu $ru");
}

failure_route[REPLICATE_FAIL] {
		xlog("L_INFO", "Replication of registration failed $fu $ru");
}
```

textPlus

OpenSIPS – Load Balancer

- As the name implies, it routes requests to different destinations based on load.
- OpenSIPS keeps track of the load at each destination and routes new requests to the destination with the least load.
- The load balancer can probe the destinations to determine their health.

OpenSIPS Load Balancer Destination Set

A destination is defined to the load balancer by its address (SIP URI) and a description and capacity of its resources.

Destinations are grouped into a destination set by a group id.

group_id	dst_uri	resources
10	sip:freeswitch.ip1.com	pstn=50;international=10
10	sip:freeswitch.ip2.com	pstn=50;international=10
20	sip:freeswitch.ip3.com	vm=25;conf=10
20	sip:freeswitch.ip4.com	vm=25;conf=5

OpenSIPS Load Balancer - Example

```
t_on_failure("PSTN_FAILURE");
if (!load_balance("10","pstn")) {
        xlog("L_INFO", "PSTN servers are at capacity");
        sl_send_reply("500", "Service full");
        exit;
}

failure_route[PSTN_FAILURE] {
        if (t_check_status("(408)|(5[09][09])")) {
                lb_disable(); # Disable the last destination used for the current call
        }

        if (load_balance("10", "pstn")) {
                t_on_failure("PSTN_FAILURE");
                xlog("L_INFO", "New destination is $du");
                t_relay();
        } else {
                t_reply("500", "Server Error");
        }
}
```

OpenSIPS Dispatcher

Computes a hash over parts of the request and selects an address from from the destination set.

Algorithms used to select a destination (not inclusive)

0 - Hash over CallId
1 - Hash over From URI
2 - Hash over To URI
3 - Hash over Request URI
4 - Round Robin
5 - Hash over Authorization Username
6 - Random

Destinations

setid	destination
10	sip:freeswitch.ip1.com
10	sip:freeswitch.ip2.com
20	sip:freeswitch.ip3.com
20	sip:freeswitch.ip4.com

OpenSIPS Dispatcher - Example

```
t_on_failure("DISPATCHER_FAILURE");
ds_select_domain("15", "0");
t_relay();
exit;

failure_route[DISPATCHER_FAILURE] {
        # If the failure was caused by a timeout put the destination into a probing state
        if (t_check_status("408")) {
                ds_mark_dst("p");
        }

        if (ds_next_domain()) {
                t_on_failure("DISPATCHER_FAILURE");
                t_relay();
        } else {
                xlog("L_INFO", "No destinations are available");
        }
}
```

OpenSIPS – Dynamic Routing

Based on multiple criteria the best gateway/destination to be used for delivering a call.

Carriers table

carrierid	gwlist
level3	l3_01,l3_02
verizon	vz_01,vz_02

Gateways table

gwid	address
l3_01	l3.ip1.com
l3_02	l3.ip2.com
vz_01	vz.ip1.com
vz_02	vz.ip2.com

OpenSIPS – Dynamic Routing Example

```
if (route_to_carrier("level3")) {
        t_on_failure("CARRIER_FAILURE");
        t_relay();
        exit;
}

failure_route[CARRIER_FAILURE] {
        # If the failure was caused by a timeout put the destination into a probing state
        if (t_check_status("(408)|(5[09][09])")) {
                dr_disable(); # Disable the last destination used for the current call
        }

        if (use_next_gw()) {
                t_on_failure("CARRIER_FAILURE");
                t_relay();
        } else {
                xlog("L_INFO", "No carriers are available");
        }
}
```

References

http://anders.com/cms/264
http://tools.ietf.org/html/rfc2915
http://tools.ietf.org/html/rfc2782
http://horms.net/projects/has/html/node8.html
http://linux-ip.net/html/ether-arp.html
http://www.linuxfoundation.org/collaborate/workgroups/networking/ip-sysctl
http://www.habets.pp.se/synscan/programs.php?prog=arping
http://beginlinux.com/blog/2009/06/defend-against-arp-spoofing/
http://www.keepalived.org/
http://www.linuxvirtualserver.org/software/ipvs.html
http://en.wikipedia.org/wiki/Virtual_Router_Redundancy_Protocol
http://clusterlabs.org/quickstart.html
http://www.zivtech.com/blog/setting-ip-failover-heartbeat-and-pacemaker-ubuntu-lucid
http://clusterlabs.org/wiki/Initial_Configuration
http://www.fwbuilder.org/4.0/docs/users_guide5/vrrpd_cluster.shtml
https://www.kernel.org/doc/ols/2001/vrrpd.pdf
http://lunar-linux.org/~nestu/vrrpd.txt
http://www.opensips.org/Documentation/Tutorials-LoadBalancing
http://images.tmcnet.com/expo/astricon/presentations/cc-10-iancu-opensip.pdf

textPlus