VRRPd: overview, implementation and usage

Jerome Etienne
jme@off.net, http://www.off.net/~jme

Abstract

This paper is about high availability and more especially about vrrpd (http://w3.arobas.net/ jetienne/vrrpd), an implementation of VRRPv2 as specified in rfc2338. It run in userspace for linux. In short, VRRP is a protocol which elects a master server on a LAN and the master answers to a 'virtual ip address'. If it fails, a backup server takes over the ip address. This memo describes the vrrp protocol and its application, the implementation of vrrpd, examples of its usage and the security which can be expected.

1 Overview

VRRPD is an implementation of VRRPv2 as specified in [KWD+98]. It run in userspace for linux. It intends to be very easy to install and configure. The protocol is very simple so vrrpd has been designed to be alike. An important design criteria has been the occam razor 'keep it as simple as possible but no more' so a single deamon per virtual router.

1.1 Protocol overview

This section gives a short overview of VRRPv2 protocol which is fully described in [KWD+98]. VRRP provides high availability at the IP address level: A box, called the master, is in charge an IP address and other boxes, called the backups monitor the master. If the master dies, the backups elects the one will become master.

Several routers participe to a virtual router: a virtual box which has a virtual ID and virtual IP addresses. It is virtual as any of the physical router may become the instanciation of the virtual router. At any moment, only one participant, the master, is in charge of the virtual router. It periodically sends VRRP advertissements to indicate its. If a backup doesn't receive any advertissement during a delay, Master-Down-Timer, it considers itself as MASTER. Each participant has a given priority used as a tie breaker during the election of the master. The election process relies in the way Master-Down-Timer is computed: The higher is the priority, the shorter is the timer. Thus the highest priority will send its advertissement before the others which will never become master.

1.2 Possible applications

VRRP has been designed to provide redundancy to routers at the IP address level. It doesn't exchange any context for the layers above IP (e.g. TCP, HTTP).

The intended purpose is to provide high availibility to default routers. [KWD+98] is only about router but this limitation is historical. VRRP can be used to other servers such as web, dns. Nevertheless, vrrp participants exchanged only IP addresses and no higher layer context e.g. tcp, http or ftp. Consequently any fail over is done at the IP level and any higher level context is lost. The usefullness of vrrp between servers is limited to the ones not requiring long lived context. For example, DNS doesn't require any context and vrrp should provide unnoticeable fail over. HTTP requires rather short lived context and vrrp may provide an acceptable fail over mechanism. On the other hand, FTP requires long live context because of the tcp connection for commands [**?**, sec 2.3] and a vrrp fail over will likely be noticed by the user.

2 Implementation problems

The hardest part of implementing VRRP is the requirement to handle the virtual router MAC addresses [KWD+98, sec 7.3] i.e. each virtual router got a dedicated MAC address and the MASTER must handle it as a virtual network interface i.e. each packet sent from a virtual ip address must have the virtual MAC address too, the ARP request must be answered with the proper MAC.

So the network interface, except if it is fully dedicated to vrrp, needs to receive packets for several MAC addresses e.g. one per virtual router plus one per owned IP address. It can be done in user space by using the multicast address list of the network interface. This feature is originaly designed to receive multiple addresses matching multiple multicast groups but it may be use for our case.

When transmiting packet, the MAC address must be set according to the source virtual IP. It is hard to overcome without kernel modification because most OS assumes a network interface has a single MAC. VRRPD chooses not to modify the kernel to match its simplicity requirement. VRRPD sets the MAC address of the network interface to the virtual one. Each packet sent througth this interface will have the virtual MAC address as source address and the protocol will fully be respected. Nevertheless this prevent from having more than one virtual router per physical interface.

As far as we know, the only motivation behind this unusual requirement is to work around hosts with bogus ARP layers. VRRPD proposes an option (-n) to ignore this requirement. VRRPD sends gratuitous ARP each time it becomes MASTER to notify hosts on the LAN of the state change. Even with the -n option, VRRPD stays compatible with other VRRP participants, interacts perfectly with non-bogus ARP hosts and remove the limitation of no more than one virtual router per virtual interface. As hosts with bogus ARP layer are rather rare and as we considere it is better to fix a bug than to work around it (so to spread it over other hosts), we considere this option acceptable.

2.1 Portability

VRRP is developped under linux and uses various low level features, such as modifying the hardware addresses of the network interface [KWD+98, sec 7.3] or handling packets directly at the IP level [KWD+98, sec 5.1]. To access low level functions from user space is usually not well standardize and this make vrrpd non-trivial to port to an operating system different than linux. Nevertheless it has been ported on FreeBSD and OpenBSD.

3 usage

This section describes the command line options and 3 examples of vrrpd's usage.

3.1 Command line options

This section explains the meaning of the command line options. The syntax is 'vrrpd -i ifname -v vrid [-f piddir] [-s] [-a auth] [-p prio] [-nh] [-m prog] [-b prog] ipaddr'.

- **-i ifname**: the interface name to run on (for example eth0, ppp0)
- **-v vrid** : the id of the virtual server [1-255]. Several virtual servers can run on a single link. Due to implementation's limit, to run several servers on a single interface require to use the -n option.
- **-p prio** : Set the priority of this host in the virtual server. The higher is the value, the higher is the priority. The value should be between 1 and 254 and has a default of 100.
- **-d delay** : Set the advertisement interval (in sec) (dfl: 1). The delay is the amount of time between 2 advertisements sent by the master. The Master-down-interval is 3 times this delay plus a skew time depending on the priority [KWD+98, sec 6.1.2]. This delay should be tuned according to the tradeoff between the bandwith used by the advertisement and the detection time of a problem.

- **-s** : Switch the preemption mode (Enabled by default). vrrp handles a preemption mode to decide wheter a higher priority backup preempts a lower priority master. It may be disabled to avoid state changes [KWD+98, sec 6.1.2].

- **-a auth** : (not yet implemented) set the authentification type auth= (none— pw/hexkey— ah/hexkey) hexkey=0x[0-9a-fA-F]+

- **-n** : dont handle the virtual mac address. (WORK: ref)

- **-f piddir**: specify the directory where the pid file is stored (dfl: /var/run) (see section 3.2)

- **-m prog** : Set the program to run when it becomes master (see section 3.2)

- **-b prog** : Set the program to run when it becomes backup (see section 3.2)

- **ipaddr** : the ip address(es) of the virtual server

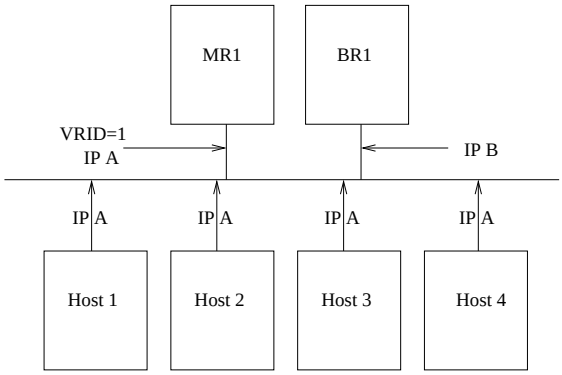
3.2 Link with external applications

In some cases, it may be interesting to link the state of vrrp to external applications e.g. port tracking (see section 3.5). VRRPd provides this feature with the -f, -m and -b command-line options and with the possibility to modify the current state by signals. If vrrpd receives SIGUSR1, it forces its change to MASTER. If it receives SIGUSR2, it forces its to BACKUP.

For example, a web server listens on the virtual IP address and the administrator want not to run it when the server is not MASTER. He can write a -m script to launch his web server and a -b one to shut it down.

3.3 Single backup

The single backup is the most basic case [KWD+98, sec 4.1].



In this case, all the hosts use IP A as the default router. The router B backs up the router A. If the router B doesn't receive any VRRP advertisement from the router A during MASTER_DOWN_INTERVAL (default around 3 sec [KWD+98, sec 6.1.2]), the router B takes over the MASTER role and starts to respond to the Virtual Router IP address i.e. IP A. But if the router B fails, the router A doesn't take over.

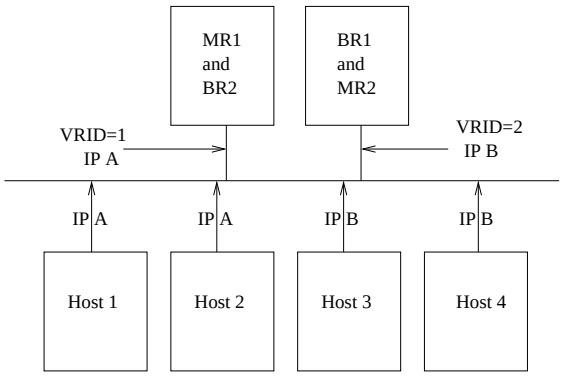
3.3.1 Command line

Assuming eth0 is the name of the interface and 10.0.0.1 the IP A, the command lines are:

- router A : vrrpd -i eth0 -v 1 -p 150 10.0.0.1
- router B : vrrpd -i eth0 -v 1 -p 100 10.0.0.1

3.4 Mutual backup

The mutual backup is likely more pratical [KWD+98, sec 4.2]. It allows load balancing and high availability.



Half of the hosts uses the router A as default router, and the other half uses the router B. Thus the load is coarsely balanced between A and B.

The routers A and B mutually monitor each others and any of them can takes over if the other fails. Thus the service is no more available only if both fail at the same time. If the administrator wishes more availbility, he can increase the number of routers which monitor each others.

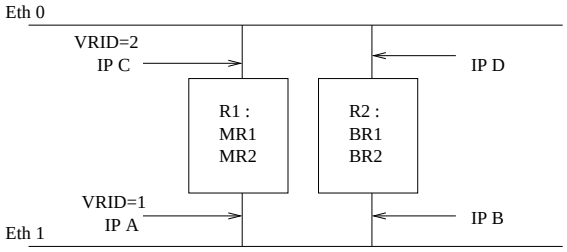
3.4.1 Command line

Assuming eth0 is the name of the interface, 10.0.0.1 is the IP A, and 10.0.0.2 is the IP B, the command lines are:

- 'vrrpd -n -i eth0 -v 1 -p 150 10.0.0.1' and 'vrrpd -n -i eth0 -v 2 -p 100 10.0.0.2' on the router A
- 'vrrpd -n -i eth0 -v 1 -p 100 10.0.0.1' and 'vrrpd -n -i eth0 -v 2 -p 150 10.0.0.2' on the router B

3.5 Port tracking

In some cases, it may be usefull to link the behaviour of multiple virtual groups of routers. For example, two firewalls are forwarding packets from a link 0 to a link 1 and vice versa. For simplicity, the example shows a single backup but it can be generalized to the mutual backup case.



The fates of eth0 and eth1 are linked: if eth1 of R1 fails, it should stop to accept packet from eth0, and vice versa. This solution can be fixed with 'port tracking'. VRRPD implements it through two mechanisms: It can launch a script after a state transition (option -m and -b) and it forces state transitions on signal reception (SIGUSR1 to be MASTER and SIGUSR2 to be BACKUP).

For example, if eth1 of R1 fails, its vrrpd with VRID=1 will change its state to BACKUP and may launch a script. The script sends a SIGUSR2 to the vrrpd with VRID=3 and forces it to be in a BACKUP state.

4 Security

The security provided by vrrp is to authentify the packet but not to encrypt them. Anybody directly connected to the LAN is able to read the packets. VRRPv2 has 3 authentications methods [KWD+98, sec 10]:

- **no authentication**: the packets aren't authenticated in anyway [KWD+98, sec 10.1]. An attacker directly connected to the link can send any packet without need to eavedrop or modify legitimate packets.
- **simple text password**: A 64bit text is included in clear inside each packet [KWD+98, sec 10.2]. An attacker directly connected to the link needs to eavedrop legitimate packets to learn the password and then he can freely forge any kind of packet. This authentication should not be considered secure because this attack is trivial and efficient. It has been designed to protect against accidental misconfiguration.
- **IP authentication header**: AH[KA98a] is the authentication protocol of IPSec[KA98b]. This protocol is efficient to prevent an attacker from modifying or creating packets. Nevertheless it has been mainly designed for unicast communication but vrrp is multicast [KWD+98, sec 3]. In multicast, AH doesn't provide an anti-replay protection thus an attacker is eavedrop packets and replay them latter.

The absence of anti-replay in the strongest method AH allows an attacker to perform multiple DoS which prevent legitimate participants from becoming MASTER and so make them useless for vrrpd sake. I am currently adapting a authentication method to solve this security flaw. It is called *anti-replay authentication* and has been designed to fix flaws i found in OSPFv2 and RIPv2 (www.off.net/ ietf/jme).

5 Summary

Vrrp is suitable for servers whom the clients are used to retry automatically in case of faillure: e.g. IP router as ip packets may be lost by the network. The security problem (see section 4) makes it unsuitable if the LAN isn't fully trusted. Once adapted, the *anti-replay authentication* will hopefully fix this problem and be proposed to the IETF as an extension to the vrrp protocol.

References

[KA98a] S. Kent and R. Atkinson. Ip authentication header. *Request For Comment (Proposed standard) RFC2402*, November 1998.

[KA98b] S. Kent and R. Atkinson. Security architecture for the internet protocol. *Request For Comment (Proposed standard) RFC2401*, November 1998.

[KWD+98] S. Knight, D. Weaver, D.Whipple, R. Hinden, D. Mitzel, P. Hunt, and P. Higginson. Virtual router redundancy protocol. *Request For Comment (Proposed Standard) RFC2338*, April 1998.