Unmasking Your IP Addresses

Harald van Breederode
Senior Principal DBA Instructor
Founder Member OU EMEA SME Team
Oracle University NL

In the January edition of the "OU EMEA Newsletter", Joel Goodman and Harald van Breederode shared their insights on upgrading from the traditional DBA 1.0 skill set to the more modern DBA 2.0 skill set and the benefit this provided to any DBA job role.

In this edition of the "OU Expert's Corner" we will focus on those network skills required to maintain IP based Oracle Clusterware interconnects.

The IP protocol

The Internet Protocol (IP) was developed in the early 1980's to enable computers to communicate with each other over a network. Despite its name, it was not designed for the "Internet" as we know it today. It was designed for "DARPAnet" which was a network for the US military. Later, the IP protocol was used to build the ARPA network for research institutions, universities and other non-commercial organisations. Eventually, the ARPA network became "The Internet" and became available to the rest of the world as the "Internet" we know today. When we speak about an "Internet" sometimes referred to as an "intranet", it means an IP based network; "The Internet" is what your mother knows as "Internet". The IP protocol used today is known as IP version 4 or Ipv4 and required that each system have a unique address, called an IP address in order to communicate on the network.

The IP address

An IP address consists of four bytes or octets "X1.X2.X3.X4", and is usually written with decimal numbers, e.g. "101.102.103.104". It consists of a network number and a host number within that network. A network packet or datagram can be directly sent from one IP address to another, if both systems are on the same network, but if the network number differs, then the packets must be sent to a router which knows the route from one network to another. Thus, a router sends packets from one network to another network.

For a router to successfully route a packet, it must quickly determine the network number and host number within that network at a known boundary within the four bytes of the IP Address. Therefore, IP addresses are divided into network classes.

Network Classes

To support routing in different sized networks, the IP protocol caters for large, medium and small networks using network classes which divide the total IP address space into Class A, B, C, D and E networks as follows:

Class A Networks

Class A networks use IP addresses where the first bit is always 0. Therefore, class A IP addresses have an X1 value between 1 and 127 inclusive, denoting the network number, whilst X2, X3 and X4 indicate the host number within that network. There are thus 127 possible class A networks of roughly 16 million hosts each.

Note: Network 127.0.0.0 is reserved for loopback purposes.

Class B Networks

Class B networks use IP addresses where the first 2 bits are always 1 and 0 (10). This means that class B IP addresses have an X1 value between 128 and 191 inclusive, which together with X2 octet, contain the network number, whilst X3 and X4 encode the host number within that network. In this case, there are roughly 16000 possible Class B networks of roughly 64000 hosts each.

Class C Networks

Class C networks use IP addresses where the first 3 bits are always 1, 1 and 0 (110). As a result, class C IP addresses have an X1 value between 192 and 223 inclusive, which together with X2 and X3 octets, designates the network number whilst X4 defines the host number within that network. There are roughly 2 million possible class C networks of 256 hosts each.

Class D and E Networks

IP addresses above “224.x.x.x”, are reserved for multicasting purposes amongst other things and are known as Class D and Class E addresses.

The Netmask

A system may send an IP packet to another system directly over the network if both systems share the same network number. If not, then the packet needs to be sent to a router first which knows where to route the packet on another network. The network and a host numbers in an IP address are determined by the use of the "netmask" which specifies which bits contain the network number and which bits contain the host number within that network. Each address class has its own natural netmask. For Class A addresses the natural netmask is "255.0.0.0" meaning that all bits of X1 are used for the network number. For Class B addresses the natural netmask is "255.255.0.0" meaning that all bits of X1 and X2 are used for the network number and for Class C addresses the natural netmask is "255.255.255.0", meaning that all bits of X1, X2 and X3 are used for the network number. Performing a logical "*AND"* operation on the binary values of the IP address and the netmask results in the network number. All bits that are not part of the network number based on the masking bits form the host number within that network.

Subnetting and Supernetting

Occasionally class A, B or C networks are too large for requirements. Sometimes class B or C networks are too small, or a class B or C network is too small and a class A or B is too large. "Supernetting" and "Subnetting" permit the combination of smaller networks into larger ones or the subdivision of larger ones into smaller ones respectively, by extending or shortening the natural netmask for the class. For example by using a class B network with address "140.85.0.0" and a netmask of "255.255.255.0" instead of the natural "255.255.0.0", the class B network is broken or subnetted into 256 networks of 256 hosts instead of 16,000 networks of 65,536 hosts. This is done by adding extra bits to the network number portion of the address and removing bits from the host number portion. Alternatively, to combine class C networks "200.200.200.0" and "200.200.201.0" into one larger network of 512 hosts instead of 2 networks of 256 hosts each, we would use a netmask of "255.255.254.0" rather than "255.255.255.0". This removes 1 bit from the network number portion of the IP address and adds 1 bit to the host number portion, thereby doubling the number of possible hosts in the network at the cost of reducing the number of networks.

Calculating the Network Number

Calculating the network number for networks that use a natural netmask is easy, but quite difficult when either subnetting or supernetting is used. Most people are not very good at performing logical "AND" operations in their heads, so a little script to help us do this calculation this is useful. The following *Perl* script will do the trick.

```
oracle$ cat apply_netmask
#!/usr/bin/perl -l

#
# Written by: Harald.van.Breederode@Oracle.com
#

use Socket;

print inet_ntoa( inet_aton($ARGV[0]) & inet_aton($ARGV[1]) );
```

This script takes an IP address and a netmask as arguments and returns the network number. For example:

```
oracle$ apply_netmask 10.161.125.60 255.255.254.0
10.161.124.0
```

CIDR Notation

In the mid 90's, routers began to use "Classless Inter-Domain Routing" (CIDR) and the concept of IP address classes began to fade away. CIDR uses a different notation for the netmask. Instead using four decimal numbers, CIDR only encodes the numbers of 1's in the netmask. So a netmask of 255.0.0.0 is written as /8. Usually you see something like "140.85.240.0/20" specifying a network number "140.85.240.0" with a netmask of "255.255.240.0" thus a subnetted class B network.

Coping with the shortage of IP network numbers

When the Internet was new, there was no shortage of IP network numbers available but the Internet Engineering Task Force (IETF) realized that IP network numbers would soon be exhausted due to the growth in Internet usage.

The first step in the solution was restricting the allocation of registered network numbers to computers directly connected to the Internet and the assignment of special ranges of numbers in each network class for systems not directly connected to the Internet. These freely available network numbers were declared non–routable, meaning they are not allowed for communication on the Internet but can be routed within and intranet. Transmission of IP packets with a non-routable IP address over The Internet would get you fired by the IP Police Department (IETF). These reserved network number ranges are: "10.0.0.0" for Class A, "172.16.0.0" to "172.31.0.0" for Class B and "192.168.0.0" to "192.168.255.0" for Class C and are available for any purpose providing they don't get transmitted on the Internet.

The second step was the introduction of the Network Address Translation Protocol also known as NAT, allowing the router to translate reserved network numbers inside a private network into a registered IP address when transmitting packets to the Internet and reversing the process for incoming packets. With NAT, you could use a reserved network number inside your network and still be connected to the Internet using only one registered IP address thereby greatly reducing the need for registered IP addresses. Today NAT is widely used, enabling the Internet to grow more than would have been possible without it. Eventually however, even NAT will not suffice, but it has given the IETF time to design the definitive solution called Ipv6, which is beyond the scope of this article.

IP and Oracle Clusterware

Now that we have a basic understanding on IP addresses, netmasks and subnetting it is time to focus on the need for this knowledge as a DBA 2.0 administering Oracle Clusterware.

During the installation of Oracle Clusterware, we specified hostnames for the public and private network interfaces as well as the required VIP address. This network configuration is stored inside the Oracle Clusterware Registry (OCR) while running the "root.sh" script, which invokes the Virtual IP Configuration Assistant (VIPCA) in silent mode. Sometimes, the Oracle10*g* VIPCA utility selects the wrong netmask and it is therefore advisable to verify that the VIP was correctly configured by using "srvctl" as shown below:

```
oracle$ srvctl config nodeapps -n nleduc1 -a
VIP exists.: /nleduc1vip/10.161.125.62/255.255.254.0/eth0
.
```

By using the Linux "**ifconfig**" command we can display all configured IP interfaces along with their settings (output shown on the next page) and verify the settings we retrieved from the OCR.

```
oracle$ ifconfig

 eth0      Link encap:Ethernet  HWaddr 00:0A:5E:5E:C3:BB
           inet addr:10.161.125.60  Bcast:10.161.125.255  Mask:255.255.254.0
           inet6 addr: fe80::20a:5eff:fe5e:c3bb/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:879587874 errors:0 dropped:0 overruns:1 frame:0
           TX packets:693624383 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:958829897 (914.4 MiB)  TX bytes:1421349176 (1.3 GiB)
           Interrupt:169 Base address:0xdc80

 eth0:1    Link encap:Ethernet  HWaddr 00:0A:5E:5E:C3:BB
           inet addr:10.161.125.62  Bcast:10.161.125.255 Mask:255.255.254.0
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           Interrupt:169 Base address:0xdc80

 eth1      Link encap:Ethernet  HWaddr 00:12:3F:AB:32:A7
           inet addr:192.168.150.1  Bcast:192.168.150.255 Mask:255.255.255.0
           inet6 addr: fe80::212:3fff:feab:32a7/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:242291440 errors:0 dropped:0 overruns:0 frame:0
           TX packets:202285444 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1350466250 (1.2 GiB)  TX bytes:3621593015 (3.3 GiB)
           Interrupt:169

 lo        Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING  MTU:16436  Metric:1
           RX packets:63339930 errors:0 dropped:0 overruns:0 frame:0
           TX packets:63339930 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:166846994 (159.1 MiB)  TX bytes:166846994 (159.1 MiB)
```

The output shows the following interfaces:

- **eth0** with IP address 10.161.125.60
- **eth0:1** with IP address 10.161.125.62
- **eth1** with IP address 192.168.150.1
- **lo** with IP address 127.0.0.1

Now that we know which interfaces are configured, we must know which one is used for which purpose. Based on IP address "127.0.0.1", we can conclude that interface "**lo**" is used as the loopback device and we already know that "**eth0:1**" is used for the VIP, but what are the other two interfaces used for?

In your Clusterware ORACLE_HOME, you will find the "Oracle Interface Configuration" tool "**oifcfg**". This utility displays all available and configured interfaces together with their intended purposes. The default output for "**oifcfg**" when invoked with no arguments is shown below

.

```
oracle$ oifcfg

Name:
        oifcfg - Oracle Interface Configuration Tool.

Usage:  oifcfg iflist [-p [-n]]
        oifcfg setif {-node <nodename> | -global} {<if_name>/<subnet>:<if_type>}...
        oifcfg getif [-node <nodename> | -global] [ -if <if_name>[/<subnet>] [-type
<if_type>] ]
        oifcfg delif [-node <nodename> | -global] [<if_name>[/<subnet>]]
        oifcfg [-help]

        <nodename> - name of the host, as known to a communications network
        <if_name>  - name by which the interface is configured in the system
        <subnet>   - subnet address of the interface
        <if_type>  - type of the interface { cluster_interconnect | public |
storage }
```

Using the "**iflist**" argument, we can display the interfaces available to the Clusterware for its usage as shown below.

```
oracle$ oifcfg iflist -p -n
eth0  10.161.124.0  PRIVATE  255.255.254.0
eth1  192.168.150.0  PRIVATE  255.255.255.0
```

This output shows that there are two interfaces available to the Clusterware together with the configured network numbers and associated netmasks. Using the "**getif**" argument we can display the interface information as configured in the Oracle Cluster Registry (OCR) with the intended usage, public or interconnect.

```
oracle$ oifcfg getif
eth0  10.161.124.0  global  public
eth1  192.168.150.0  global  cluster_interconnect
```

Looking at the help and output from the "**`oifcfg`**" utility we clearly see that it uses network numbers instead of IP address, so the skill to extract the network number from an IP address is needed if we want to understand our cluster network configuration or we want to change our setup. Changing the VIP address can be done using the "srvctl" command and by using the "oifcfg" utility, we can change the public and private addresses.

Conclusion

Knowledge of IP addresses and netmasks, and the ability to extract the network number from an IP address, are required to successfully administer and maintain the Oracle Clusterware configuration.