Internet Redundancy and Balancing with ZeroShell

Leandro Ferrari Crocomo – leandro.crocomo at gmail.com

06 May 2011

1. The Scenario

A small office with a network containing about 20 nodes, ranging from printers, workstations, network storage devices, and wireless access points.

The office has 2 independent Internet connections; the first is a regular 10Mega ISDN local provider and the second one is a dedicate link with 512k bandwidth.

VoIP communication, without video, and normal web navigation are the main applications.

2. The Challenge

Adopt a system able to:

- Handle the two internet connections simultaneously,
- Implement redundancy in situations where one of the links is offline. (Fail-Over),
- Distribute traffic according to each link best capability, prioritizing VoIP traffic over the dedicated internet link,
- Generate usage statistics and monitor link availability,

Additionally, the solution should be able to provide other capabilities like DHCP server, Firewall, etc.

3. The Solution

After some web research we came across with ZeroShell and after reading the available documentation and check the product capabilities we decide to give it a try.

The network topology, including the ZeroShell Server and the 2 incoming links, are depicted below.

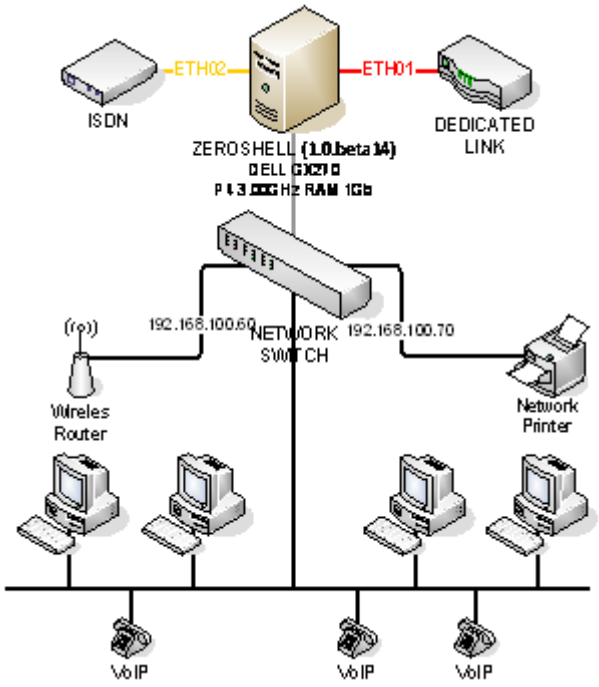


Figure 1 - Office Network Topology

4. Hardware

An old DELL Optiplex GX 270, Pentium IV, 3GHz, with 1G RAM, equipped 1 onboard Gigabit Network Interface Controller and 02 additional NIC's from Realtek RTL-8139/8139C/8139C+ (rev 10), 100Mb/s Full Duplex, was used as the base hardware for the deployment.

Figure 2 - DELL OptiPlex GX270 - P4 3GHz 1G RAM

The hardware is pretty much above the minimum requirements needed by Zeroshell: x86, Pentium 233MHz with 96MB RAM.

5. DOWNLOADING AND INSTALLING

After installing the two NIC's the next step was to download and burn the CD image (1.0.beta14) at:

http://na.mirror.garr.it/mirrors/zeroshell/ZeroShell-1.0.beta14.iso

Booting the system from the CD was an important step to make sure that the hardware was functional, and that all the NIC's in the system were properly recognized by ZeroShell.

Next step, was to download the image for installation in a Hard Drive or a CF card from:

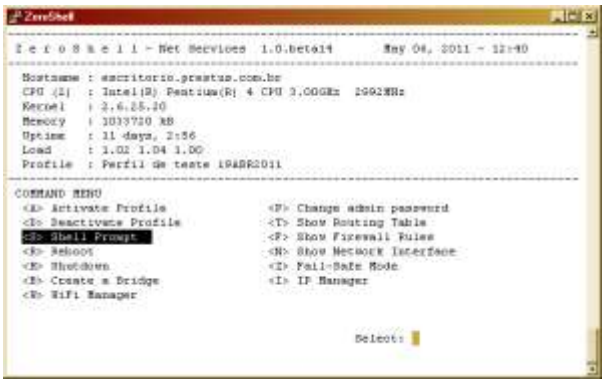
http://na.mirror.garr.it/mirrors/zeroshell/ZeroShell-1.0.beta14-CompactFlash-IDE-USB-SATA-1GB.img.gz

The documentation at the link below was an excellent reference on how to get ZeroShell running from your HD:

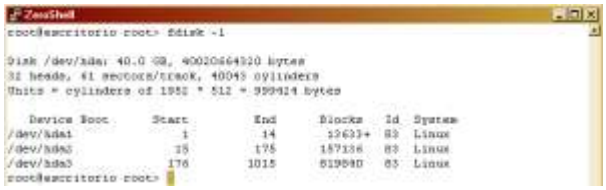
http://digilander.libero.it/smasherdevourer/schede/linux/Zeroshell%20su%20HD-EN.pdf

To make it short, after booting up from the CD, and assuming the you have a monitor an keyboard attached to the PC where ZeroShell is running you should follow the steps:

1. Press 'S' to get the Shell Access,



2. Check if your hard drives and that the USB Pen Drive where you have previously saved the ZeroShell image were properly recognized:



3. Create the directory to where your PenDrive is going to be mounted:

```
root@escritorio root> cd /mnt/
root@escritorio mnt> mkdir pendrive
```

4. Mount your Pen Drive to the directory created above:

```
root@escritorio mnt> cd /
root@zeroshell /> mount /dev/sda1 /mnt/pendrive
```

5. Unzip the image file to your Hard Drive:

```
root@zeroshell /> cd /mnt/pendrive
root@zeroshell pendrive> gunzip -c zeroshell.img.gz>/dev/had
```

Note: The option - c used with the gunzip command, writes output on standard output; keeping the original file unchanged. The stdout is pointed directly to /dev/had, and that is the reason why you do not need to modify partitions on the HD prior to the installation.

6. Restart your system, and remove the CD from your drive.

```
root@zeroshell pendrive> reboot
```

Your system should now boot up from the installation in your Hard Drive.

6. GENERAL CONFIGURATIONS

After logging into the web interface from ZeroShell, the first step is to check in the Network tab, under *System* → *Setup*, if all the NIC's were recognized and if the physical connection to them is OK.

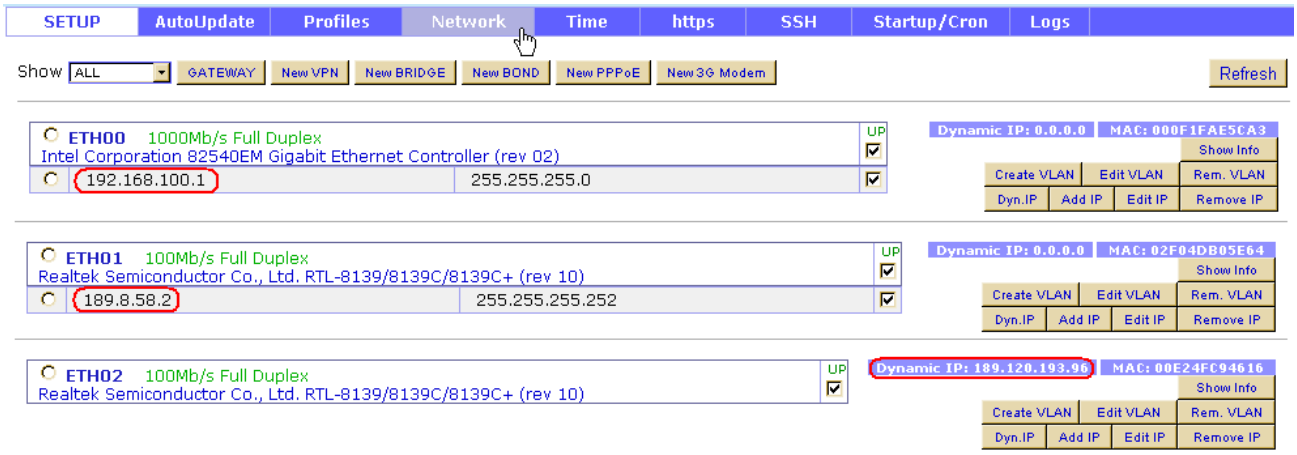


Figure 3 - Checking Status and Configuring IPs for all the NIC's.

In the case of our office, the link connected to the ETH02 has a Dynamic IP, so DHCP was enabled for this NIC.

Figure 4 - Enabling DHCP for ETH02.

For the dedicated link, connected to ETH01, the configuration provided by the link provider was used, as depicted in the figure below.

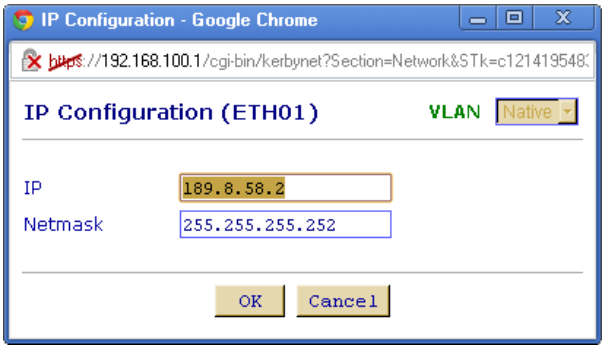


Figure 5 - Manual IP Configuration.

An important part is to configure "Default Gateway". In our case, we choose the gateway from the ETH02 ISP provider.

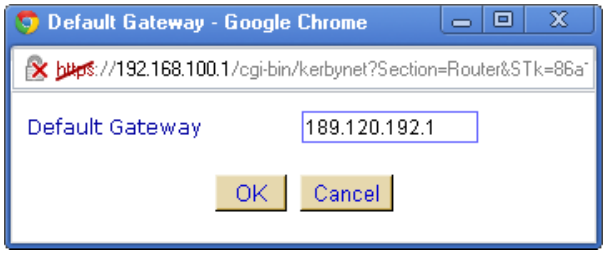


Figure 6 - Defining the Default Gateway.

The ETH00 interface, connected directly to the Network Switch was configured manually with the IP 192.168.100.1, which is the IP for the ZeroShell maintenance and web interface access, as well.

The DHCP configuration for the LAN side, was activated and configured in *Network* → *DHCP*.

The dynamic IP range starts with 192.168.100.20 and goes up to 192.168.100.50, and for some peripherals a static IP address was assigned.

HINT: We configured the IP address 192.168.100.1 as Primary DNS, which is the ZeroShell box itself, and as Secondary and Tertiary DNS we adopted the Google's Public DNS (8.8.8.8 and 8.8.4.4), as it can be found at:

http://code.google.com/intl/pt-BR/speed/public-dns/docs/using.html

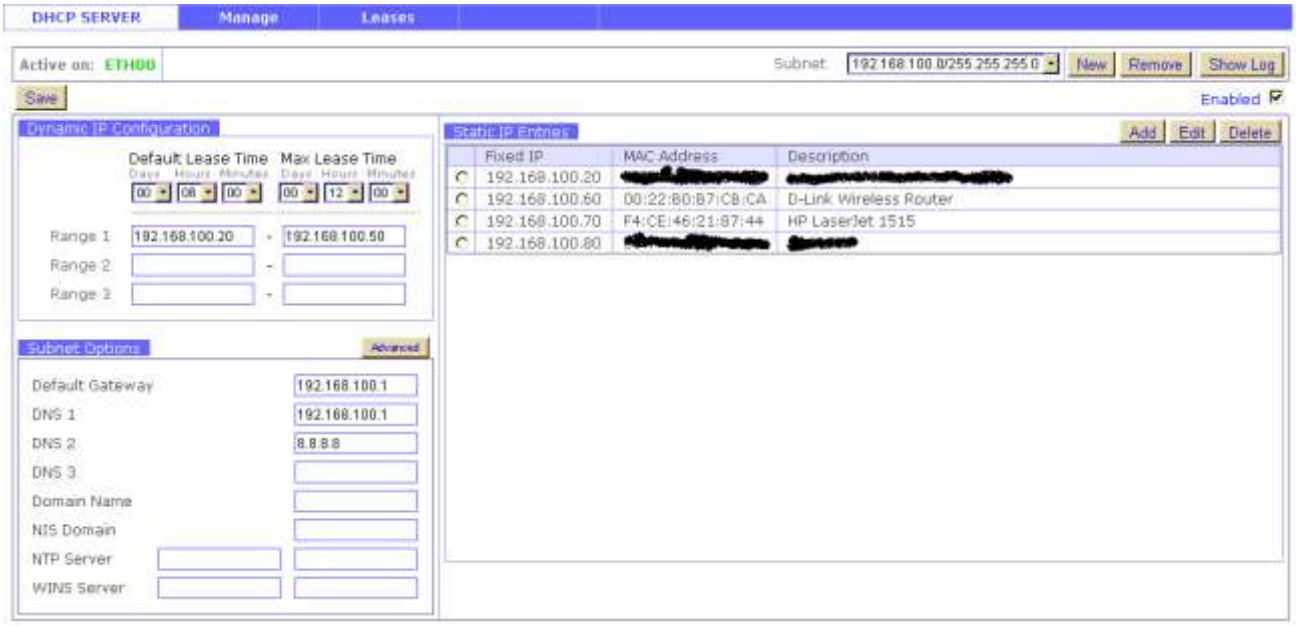


Figure 7 - Configuring DHCP and Static IP Clients at the LAN Side.

After checking that the local clients were able to get valid local IP addresses assigned to them, the next configuration step is under *Network* → *Router*. The figure below, shows the NAT configurations, as we are using:

Figure 8 - NAT Configuration.

A comprehensive tutorial about NAT, can be found at: http://en.wikipedia.org/wiki/Network_address_translation But in a few words, NAT can be used to enable computers on a network such as in small offices or home offices (SOHOs) to have a common Internet connection using a single public IP address.

After completing this step you should be already connected to the web. The next configurations will discuss the failover and redundancy aspects, and how to route VoIP traffic through the proper outgoing link.

7. Load Balacing and Failover

In this configuration step we aim at a scenario where if both links are UP, ALL VoIP traffic flows through the ETH01 interface and all other traffic goes through ETH02 interface.

In the situation where one of the links are down, we have no choice so all traffic will go through the available link.

Under the *Network* → *Net Balancer* tab we add the two Gateways and select the option "Load Balancing and Failover". It is important to mention at this point that the weight given to each gateway reflect the bandwidth relation between them, in our case, 20:1.

Additionally we define the FailOver criteria, which in our case relays in the IP addresses 200.160.4.20 and 200.147.67.142. To be considered "DOWN" the number of failures in the ICMP check was set to be 3, while to be considered "UP" again it must pass 5 times. This difference in the limits to be considered UP or DOWN is sometimes referred to as "*hysteresis*".

Figure 9 - Configuring Balancing and Failover.

At this point we could "play" by removing the Ethernet cables that connect to the ETH01 and ETH02, or by removing the coaxial cable that feeds the ISP cable modem. In either situation, physically or logically disconnection of one link should be no problem since as soon as ZeroShell detects it, the traffic will be routed through the available link. Try it out! ☺

8. BALANCING RULES

The second part of our requirements could be achieved by creating two balancing rules.

Go to *Network* → *Net Balancer*, and click on the Balancing Rules tab.

We will describe rules here that will make all the VoIP traffic to go through ETH01. In our setup, all the VoIP traffic that needs to be routed to ETH01 is always directed to our SIP server, and coming from ETH00.

Moreover, we use a VoIP Gateway, which has a static LAN IP, so one rule will define that all traffic coming from this device AND going to our SIP server should be routed through ETH01.

The second rule defines that all traffic going to our SIP provider flow through ETH01 as well. In reality, just one rule would do the job, but we decided to keep both of them.

Figure 10 - Configuring the Balancing Rules.

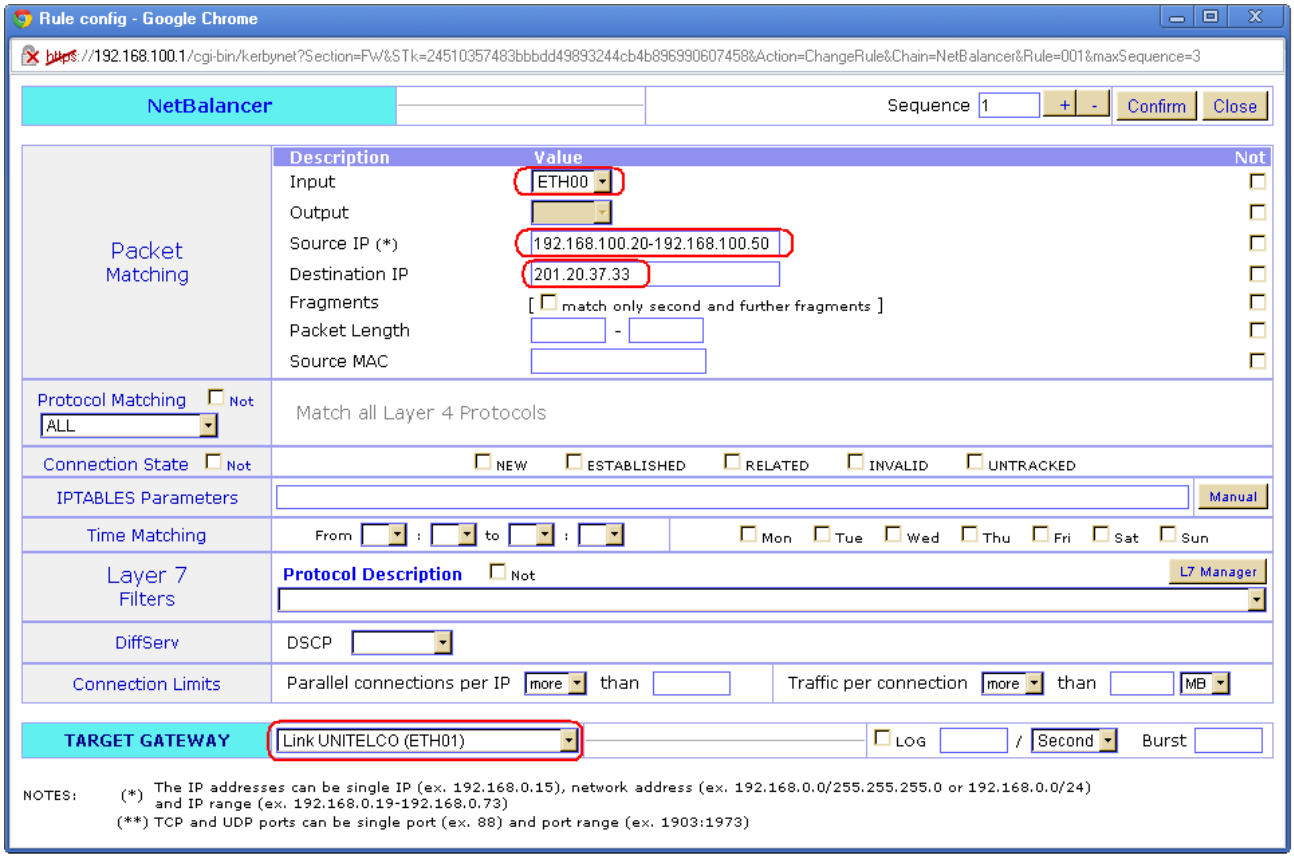


Figure 11 - Rule describing the traffic balancing.

9. Conclusion

ZeroShell is proving itself as an excellent stable solution for our needs.

The installation process was easy, and getting it properly configured and running took us just a couple of days for reading the documentation and experimenting with the configurations.

The capability of ZeroShell to deal with Configuration Profiles allows for experimentation without compromising the validated production setup, which is our case, is also an important aspect.

Adding static routes so that VoIP traffic flows through ETH01 was tried at the beginning. At a first glance, this approach seems to be ok but it only works fine as long as ETH01 is UP. Describing the same rule as a balancing rule allow VoIP traffic to flow through ETH02, but only in those situations where ETH01 is DOWN, which one of the requirements we were pursuing.

10. Next Steps

In the near future we consider:

- Checking how SAMBA works with ZeroShell (http://www.zeroshell.net/eng/forum/viewtopic.php?t=2209)
- Pluging in a USB Wireles Adapter and trying to setup a Wireless Access Point.
- Playing with Captive Portal.
- Figure out how reports from ZeroShell data could be reported per e-mail, automatically.