

The AirJaldi Bandwidth Maximizer (BwM)

Proving Concept, Demonstrating potential and Viability

Annex – installation

Dates covered by this report: March 2009 – June 2010

Country: India

Project leader : Michael Ginguld

Team members : Michael Ginguld, Fredric Renet,

Aurelien Personnaz Yahel Ben David, Arti Sinha

Table of Contents

I.	Annex: Installation Procedure for the AirJaldi BwM.....	3
1.1.	Requirements	3
1.2.	Installing Ubuntu Server 10.4.....	3
1.3.	Installing Required Packages.....	4
1.4.	PDNS – Recursor Configuration	4
1.5.	Mpath Installation.....	5
	<i>Configuration.....</i>	6
	<i>Gateways and Rules’ Configuration</i>	7
	<i>Test.....</i>	9
	<i>Use of Mpath Web Interface</i>	9
1.6.	SQUID Installation	10
1.7.	SNMPD Installation.....	11
1.8.	Firewall Script	12



I. Installation Procedure for the AirJaldi BwM

1.1. Requirements

This document details the installation procedure for BwM server version 1.0. The attached configuration files and specific software packages are necessary to complete a successful installation.

In order to do the installation you need the following:

- A PC with at least 80 GB, Processor speed >1 GHz, RAM >1 GB
- CDROM Ubuntu Server 10.4
- Working Internet connection to download updates
- This document and the installation file bwm-1.1-conf.tgz (available on the AirJaldi website)

The basic requirements for the server will not allow you to use the server in a production environment, unless you operate with a less than 2 Mbps upstream speed.

For a production environment with a 10Mbps upstream link the server should be:

- Processor >1.6GHz Core 2 Duo with 64 bits support
- RAM 4 GB
- 2 HDD SATA 7200tr/mn 300 GB or more

1.2. Installing Ubuntu Server 10.4

The 10.4 version of Ubuntu server is tagged LTS (Long Term Support), it means the version will be updated and security fixes will be supported for the next 5 years. More information can be found at:

<https://help.ubuntu.com/10.04/serverguide/>

A very detailed documentation for Ubuntu server installation can be found at:

<https://help.ubuntu.com/10.04/installation-guide/amd64/index.html>

Recommended reading:

<https://help.ubuntu.com/10.04/serverguide/C/advanced-installation.html#software-raid>

In order to configure RAID partitions.

Hard drive partitioning:

With two 300 GB HD we do the following partitioning:

- /raid1 5 GB
- /usr raid1 20 GB
- /var raid1 120 GB
- /tmp raid1 1 GB



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

AirJaldi BwM - Final Technical report

- /squid1 on SDA 120 GB
- /squid2 on SDB 120 GB
- /swap1 on SDA 8 GB
- /swap2 on SDB 8 GB

The raid partitions will allow the system to withstand the failure of one hard drive. We recommend having two separate partitions for squid cache, one on each hard drive, and to not use raid on these partitions. The two partitions will be mounted as /squid1 and /squid2.

In the event of a disk failure, the SQUID configuration file will need to be modified by removing the cache pointing the failed drive.

All partitions will be formatted with the ext4 file system and the remaining drive space (if any) is left not used.

You do not need to install any package in addition to the base system, so when you are prompted to install other packages or choose “tasks” do not select any item and complete the installation.

1.3. Installing Required Packages

In order to install the required packages type the following command:

```
sudo aptitude install openssh-server snmpd squid pdns-recursor apache2 ntop
```

This will install the main components of the server and the associated dependencies automatically.

When the installation is finished, each component must be configured. The files with custom configuration options are in an archive named bwm-1.1-conf.tgz (available in the AirJaldi website). Transfer this to your home directory using scp from another computer:

```
scp bwm-1.1-conf.tgz IpAddressOfTheNewlyInstalledServer:/home/NameOfUser
```

You must replace IpAddressOfTheNewlyInstalledServer by the real IP of the server, and NameOfUser by the username given during the installation.

To know the IP address of the server if not set statically type the command:

```
ifconfig
```

1.4. PDNS – Recursor Configuration

First, allow PDNS-Recursor to start, edit the file with :

```
sudo vi /etc/default/pdns-recursor
```

and replace the line START=no by START=yes and then save the file.

Next, copy the configuration file :

```
sudo cp ~/bwm-1.1-conf/etc/powerdns/recursor.conf /etc/powerdns/recursor.conf
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

Next, you should start the server :

```
sudo /etc/init.d/pdns-recursor start
```

You must then configure the system to use the newly installed DNS :

```
sudo vi /etc/resolv.conf
```

This file must contain only one line : `nameserver 127.0.0.1`

You can now test the DNS by typing the commands :

```
sudo aptitude install dnsutils
```

```
dig A www.google.com @127.0.0.1
```

1.5. Mpath Installation

Install PHP:

```
sudo aptitude install php5 php5-cli libphp-adodb
```

Copy the two files necessary for MPath in your account:

```
scp mpath-tools-airjaldi.patch mpath-tools-1.0.0.tgz \  
IpAddressOfTheNewlyInstalledServer:/home/NameOfUser
```

Of course you must replace `IpAddressOfTheNewlyInstalledServer` by the real IP of the server, and `NameOfUser` by the username given during the installation.

Untar the sources:

```
tar -zxvf mpath-tools-1.0.0.tgz
```

Move inside:

```
cd mpath-tools-1.0.0
```

Patch the sources:

```
patch -p1 < ../mpath-tools-airjaldi.patch
```

Install:

```
sudo php install.php  
sudo cp -r www /var/www/mpath
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

Configuration

IP route configuration

Some specific IP route configuration is needed to run Mpath.

Edit the /etc/iproute2/rt_protos file and add

```
#  
# mpathd  
#  
200 mpathd
```

Edit the /etc/iproute2/rt_realms file and add

```
#  
# local  
#  
1 gateway1  
2 gateway2
```

Replace the gateway1 and 2 by the names of your gateways. You can add more if needed.

Edit the /etc/iproute2/rt_tables file and add

```
#  
# mpathd  
#  
1 gateway1  
2 gateway2  
10 mpath_dyn_0  
11 mpath_dyn_1  
12 mpath_dyn_2  
13 mpath_dyn_3  
14 mpath_dyn_4  
15 mpath_dyn_5  
16 mpath_dyn_6  
17 mpath_dyn_7  
18 mpath_dyn_8  
19 mpath_dyn_9  
20 mpath_dyn_10  
21 mpath_dyn_11  
22 mpath_dyn_12
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

23 mpath_dyn_13
24 mpath_dyn_14
25 mpath_dyn_15
26 mpath_dyn_16
27 mpath_dyn_17
28 mpath_dyn_18
29 mpath_dyn_19

Replace the gateway1 and 2 by the names of your gateways. You can add more if needed.

Mpath Configuration

Gateways and Rules' Configuration

- Gateway definition files

The gateways are defined in the directory `/etc/mpath-tools/gateways`.

Remove all the gateways except `adsl.conf` in `/etc/mpath-tools/gateways`. Copy it once for each of the gateways and rename them with the names of your gateways. Edit the `ip_addr`, `network`, `nextthop` and `interface` values inside to fit with your settings.

`ip_addr` is the address of your server connected to this gateway.

`network` is the address of the network that you use to connect to this gateway

`nextthop` is the address of this gateway

`interface` is the name of the network interface you use to connect to this gateway.

- `gateways.conf`

Then edit the file `/etc/mpath-tools/gateways.conf` to have one block like the following for each of your gateways and remove everything else:

```
gateway "GW" {  
include "gateways/GW.conf";  
}
```

You have to replace `GW` by the name of your gateways.

- `mpathd.conf`

Edit the file `/etc/mpath-tools/mpathd.conf`.

Then edit the events – shutdown parameter. It decides which gateway should be used when `mpath` is shut down. So replace “`ppp0`” by the name of the device of this gateway and “`adsl`” by the name of this gateway.



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

AirJaldi BwM - Final Technical report

- probe_common.conf

Edit the file /etc/mpath-tools/probe_common.conf

Add the following line at the end:

```
ip_addr "IP ADDRESS";
```

This defines which address the program will try to reach with a ping to check if the gateways are alive. Replace IP ADDRESS with an IP of your choice (we recommend 8.8.8.8 which is the DNS of Google)

- rules.conf

Edit the file /etc/mpath-tools/rules.conf

In the section "default", remove the second "via" line and edit the first one. This line defines which gateway is used and what is its weight. You have to enter the gateways like this :

via "GW1:WEIGHT" "GW2:WEIGHT" "GW3:WEIGHT"

In our case it gives :

```
via "airtel:4" "reliance:1";
```

Finally, edit the "fwmark" sections. It defines the marking of the connections on each gateway.

You have to set a "fwmark" section for each of your gateways like this:

```
fwmark "0xID/0x0f" {  
via "GATEWAY";  
}
```

You have to replace ID with a hexadecimal value between 00 and FF, which should be different for each gateway. And replace GATEWAY by the name of your gateway.

In our case it gives:

```
fwmark "0x01/0x0f" {  
via "airtel";  
}
```

```
fwmark "0x02/0x0f" {  
via "reliance";  
}
```

- Automatic startup

Use the startup script manager of your distribution to start /etc/init.d/mpathd automatically.

With ubuntu/debian the command is:

```
update-rc.d mpathd defaults
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

AirJaldi BwM - Final Technical report

Test

Run the mpath daemon with the command:

```
mpathd
```

Check that it is working with the command:

```
mpath show gateways
```

Mpath should return a list of the gateways. If not, check your configuration.

You should now be able to connect to the web interface at the URL:

```
http://IP_OF_YOUR_SERVER/mpath
```

Use of Mpath Web Interface

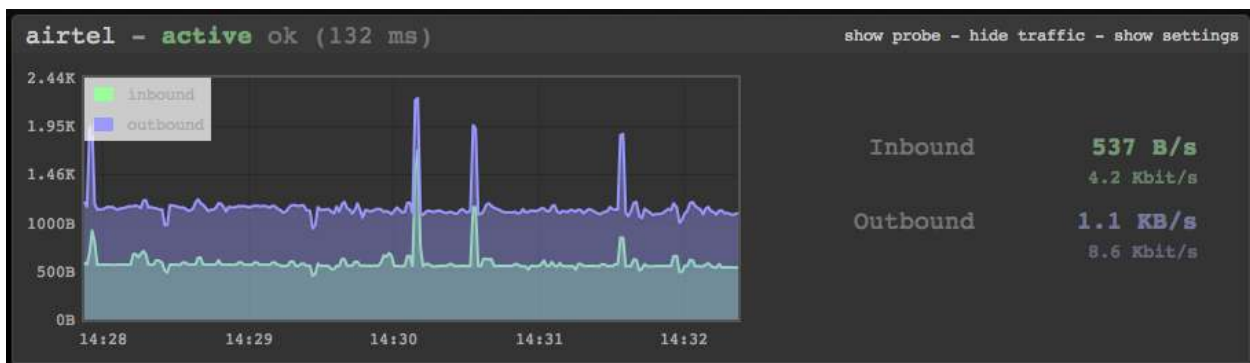
The web interface is separated in blocks, one for each of the gateways and one for the routing rules.

For each gateway you can display:

- A live monitor of the probe (ping details to the configured IP address)



- A live monitor of the traffic on this interface

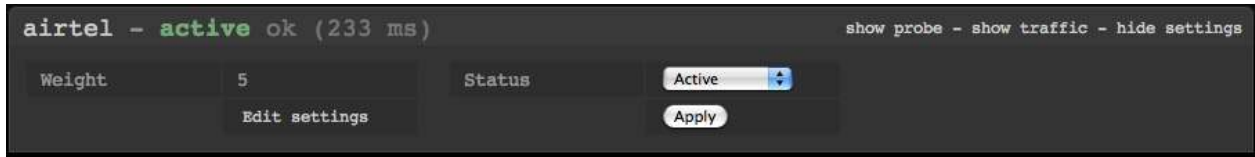


- A form to edit the settings of this gateway



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

AirJaldi BwM - Final Technical report



There you can edit the weight of the gateway and change the status of the gateway by hand.

The available status options are:

- Active, the gateway is part of the load balancing.
- Disengaged, some traffic can go on this gateway but it is no longer part of the load balancing.
- Disabled, the gateway is down.

Finally you can add or remove some routes managed by mpath in the routing rules block



To add a rule you have to give the address of a machine or a network, to choose if it is a source or a destination address, and to choose to which gateway the traffic will be sent.

Source means that all the traffic coming from the given address will be sent to the given gateway.

Destination means that all the traffic going to the given address will be sent to the given gateway.

1.6. SQUID Installation

First, copy the configuration files to their location :

```
sudo cp ~/bwm-1.1-conf/etc/squid/* /etc/squid/
```

Next, copy the storeurl helper script and the windows caching script to its location

```
sudo cp ~/bwm-1.1-conf/usr/bin/storeurl.py /usr/bin/
```

```
sudo cp ~/bwm-1.1-conf/usr/bin/windows_update-squid.sh /usr/bin/
```

Next the cronjob for the windows update script

```
sudo cp ~/bwm-1.1-conf/etc/cron.d/win-update-squid /etc/cron.d/
```

Finally, fix the different ownership of relevant directories :

```
sudo chown proxy /squid?
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

```
sudo touch /var/log/storeurl.log
sudo chown proxy /var/log/storeurl.log
and create the required directories for squid :
sudo squid -z
```

SQUID is now ready to run:

```
sudo /etc/init.d/squid restart
```

To check if squid is working do :

```
curl -x 127.0.0.1:3128 http://www.google.com
```

You should see the HTML code for the page displayed in your console. If it is not working you should check SQUID's log files :

```
sudo tail -n 100 /var/log/squid/access.log
```

```
sudo tail -n 100 /var/log/squid/error.log
```

1.7. SNMPD Installation

In the basic configuration, you will need to add a script to get Power DNS data.

Add the line :

```
extend pdns-rec /usr/local/bin/pdns_stats.sh
```

at the end of the file : /etc/snmp/snmpd.conf

And copy the script :

```
sudo cp ~/bwm-1.1-conf/usr/local/bin/pdns_stats.sh /usr/local/bin/
```

Finally, reload the snmpd configuration :

```
sudo /etc/init.d/snmpd reload
```



This work is licensed under the Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License.

1.8. Firewall Script

The firewall script opens only the necessary ports on the server and does the transparent proxy redirection for all TCP traffic on port 80. First copy the script in its location :

```
sudo cp ~/bwm-1.1-conf/etc/init.d/firewall /etc/init.d/
```

To start it at each boot, put a link in the RC2.d directory :

```
sudo ln -s /etc/init.d/firewall /etc/rc2.d
```

And finally run the script :

```
sudo /etc/init.d/firewall start
```

