Design of a Dual Stack DNS Server with Dynamic Updates

Olaleye Oludare, Olaniyan Ayodele, Famuyiwa Kolawole

Abstract- The need for dynamic reconfiguration of IP addresses increases mostly due to the increased mobility among internet connected devices. A previous research work was able to develop a solution for dynamic DNS with updates initiated by the client based on already available components. The solution works but only supports IPv4. The adoption of IPv6 is on the increase due to IPv4 depletion. The aim of this study is to adapt the provided solution to dual stack mode that supports both IPv4 and IPv6. The conclusion of the study is that the concept works and it was possible with some effort to solve the problem in a relatively simple way.

Index Terms- Client, DNS, Dynamic, IP, IP address, IPv4, IPv6, Mobility, Server.

———————————— ◆ ————————————

1 INTRODUCTION

Nowadays many client computers often change their Internet Protocol (IP) addresses. Owing to the shortage of IP addresses many users have dynamic IP addresses instead of a static one. Also, a lot of people own a laptop or other portable devices connected to the Internet. This furthermore increases the need for mobility since they will often join different networks and will obtain a different IP address each time. A problem arises if you want to host a service like a File Transfer Protocol (FTP) server or a web server. Users want to connect to a service using the same information every time they connect. The way to accomplish this is either to have a static IP address or a static Fully Qualified Domain Name (FQDN) e.g. "mywebserver.ddns.com". Since a static IP address is not always possible with different network, the solution has to be a static FQDN. Usually the FQDN is mapped to a static IP address in the DNS server, but for a laptop, this normally does not work. The ideal solution is called DDNS (Albitz & Liu, 2001).

DDNS permits automatic changes of DNS entries in the name server. In that way a new IP address can be mapped to the FQDN, allowing the server administrator to make use of dynamic IP addresses. In this way the user who wants to connect to a server only needs the FQDN. DDNS would also allow clients to keep their hostnames even if the IP configuration is done dynamically especially when they are moved in smaller environments, e.g. a computer lab in a university. In this case the DDNS would allow an administrator to move computers without any reconfiguration at all.

1.1 Similar Solution

DynDNS is a DNS service provided by Dynamic Network Services Inc. (DynDNS.com, 1997). Its primary concept is to allow dynamic updates of DNS entries. This is done by client software that periodically checks whether the IP address of the host has changed. If a change is detected, the changed IP address is sent to the DynDNS server and the DNS entry is updated. In this way the computer hosting the service can always be accessed although its IP address changes. A similar implementation is provided by no- ip.org (No-IP, 2000). The above stated DDNS solution does not support IPv6 and this make the solution not suitable for lots of devices connected to the internet using IPv6.

1.2 Motivation

Every device on the internet must be assigned an IP address in order to communicate with other devices. With the ever-increasing number of new devices being connected to the internet, the need arose for more addresses than IPv4 is able to accommodate.

IPv6 is the next generation Internet Protocol address standard intended to supplement, and eventually replace the IPv4 protocol most Internet services use to transact on the Internet today .IPv6 uses a 128-bit address, allowing 2^{128}, or approximately 3.4×10^{38} addresses, or more than 7.9×10^{28} times as many as IPv4, which uses 32-bit addresses. IPv4 allows only approximately 4.3 billion addresses. The two protocols are not designed to be interoperable, complicating the transition to IPv6. The

- *Olaleye Oludare is currently pursuing masters degree program in computer science in ladoke Akintola University of Technology,Nideria. E-mail: oludarejohnson@yahoo.com*
- *Olaniyan Ayodele is currently pursuing masters degree program in computer science in ladoke Akintola University of Technology,Nideria. E-mail: ezbkayo4live@yahoo.com*
- *Famuyiwa Kolawole is currently pursuing masters degree program in computer science in ladoke Akintola University of Technology,Nideria. E-mail: foluwa926@gmail.com*

goal of this study is to add IPv6 capability to the stated DDNS solution.

2 Technical Background

2.1 DNS

The DNS is used to translate host names and service names (e. g. www.chalmers.se) to numeric IP addresses (e.g. 129.16.221.8), which is a more suitable format for computers. In short this is done by a lookup in the DNS server's database, and if not found the server will contact other DNS servers to get the correct IP address for the requested lookup (Fig 1). IP addresses to other DNS servers and hosts will be cached in the local DNS server performing a lookup. How long an address is valid in the cache is decided by the TTL value of the IP address. TTL is set when the address is added in its authoritative DNS server's database. This cache function results in that commonly used addresses and domains are often found in the cache. With cached information the time of the lookup decreases and less data traffic is produced.

In a full DNS lookup without any cached information, the local DNS server will request a root name server for an IP address to the correct top level domain DNS server (e.g. .se,.com). The top level server, which knows what lower level DNS servers' IP addresses are, redirects the local DNS server further down the hierarchy of DNS servers. This proceeds between the local DNS server and other DNS servers until the IP of the requested hostname is known or results in an error. The local DNS server also sends the correct address or an error to the client that requested the DNS lookup. The DNS server contacted last in a lookup which is responsible for a portion of the name space delegated to its organization, which is called the DNS zone. This authoritative DNS server has the address saved in its database along with the TTL value.

An example of a DNS lookup where the top level domain server is known by the local DNS server is illustrated in Fig. 1.

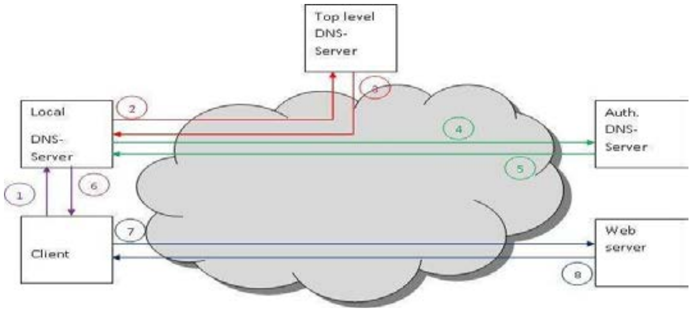


Fig. 1. Standard DNS lookup

1. Client asks for an IP address to a certain name of for example a web server.
2. Local DNS server asks a top level domain server for an address to a lower level DNS server.
3. The top level domain server answers with an IP-address to an authoritative DNS server.
4. Local DNS server asks the new DNS server for the address of the web server.
5. The server was an authoritative DNS server thus it answers with the correct IP address.
6. The IP address is forwarded to the client.
7. Now the client can ask for the web page from the web server since it got the exact address.
8. Data exchange between client and web server starts.

2.1.1 DNS Security Extensions (DNSSEC)

DNSSEC is a software solution that adds security to the DNS protocol (Arends, 2005). It provides origin authentication, data integrity and authenticated denial of existence for the DNS protocol. In regular DNS lookups the client can ask for digital signed DNS data sent from all DNS servers used in the lookup. A server of one level higher position in the hierarchy can verify the origin of the DNS server below. Signing can be done with a symmetric key (e.g. TSIG) or an asymmetric key (e.g. SIG) (Albitz& Liu, 2001).

DNSSEC is also used when updating the DNS database which is useful over an open network (Albitz& Liu, 2001). If redundancy is needed, i.e. the DNS implementation is based on two or more computers; in that case it is vital to use DNSSEC for sending the DNS database between the servers to maintain full security. In this case it is not only encryption that is needed, other security solutions are useful e.g. Message Authentication Code (MAC).

2.2 RADIUS

The RADIUS is an Authentication, Authorization and Accounting (AAA) network authentication protocol (Rigney, C. et al, 2000), (Cisco, 2006). RADIUS is used for authentication and can easily be combined with other software. The client sends a request to the Network Access Server (NAS) in order to use a specific resource. The NAS forwards the request to the RADIUS server, which replies with a challenge for the request (e.g. password). If the client's password is found in the RADIUS server's database and several other credentials (e.g. IP address) are met access is granted, else access is denied.

The RADIUS protocol is in widely spread usage which means there are a large number of client implementations.

This allows most platforms to make use of the solution. FreeRADIUS is a server implementation of the RADIUS protocol. It comes with a PHP web interface which can be modified at users' discretion.

2.3 LAMP

LAMP is a solution stack of software (ONLamp.com, 2001). That means it is a package of software which together works as a full solution. LAMP is an acronym for Linux, Apache, MySQL and PHP (also Perl or Python). These components work together as a web server. The LAMP concept is used in this study. Apache HTTP Server is open source software which is used for web page hosting (Laurie, B & Laurie, P, 1999). It is the most common web hosting application used (Netcraft, 2009). There are many modules for apache which extends the possibilities of usage.

MySQL is an open source implementation of a relational database (MySQL, 1998). The distribution also contains the tools necessary for editing and managing databases. As the name implies it uses the database computer language SQL.

PHP, which is the chosen scripting language in this research, is a server scripting language designed for dynamic web pages i.e. the content of the web site is changing.

2.4 Certificates

Certificates are used for secure connection between a client and a server. When the client tries to connect to a web page the server will send a certificate containing information that the client can use for encrypting further traffic exchanges with the server. If the client accepts the certificates the server grants access to the requested web page and all data traffic goes under encryption.

2.4.1 TLS/SSL

If a client wants to connect to a secured web site, it first sends a "client hello" message to the server (Figure 2). This message includes information about the TLS/SSL protocol version, what ciphers and compression method the client supports, and some initial random numbers. The server send the same information about itself to the client in response, it also adds its certificate ("server hello"). The server might request a client certificate to verify the client, but this is optional.

In any case client calculates a pre-master secret out of the two random numbers and sends it to the server ("client_key_exchange"). The pre-master secret is encrypted with the public key of the server. Now both the server and the client can calculate a master secret out of the pre-master secret, by applying a combination of the MD5 and SHA algorithms on the two random numbers and the pre-master secret. This master secret is used to calculate six further keys, three for each side. One key for writing, one key for signing and the last one as initialization vector for block encryption. All these new keys are symmetric, to limit the time needed for decryption. Finally a control message is sent containing all messages send previously but encrypted. The server returns the same information. From now on every further message is encrypted.

This function is included in Apache as a module called mod_ssl (Engelschall, 2001). If the signing process is done by the server itself the resulting certificate is called a self-signed certificate and the user is informed about this by its web browser. On the other hand if it is signed by a registered certificate provider no extra information is provided.

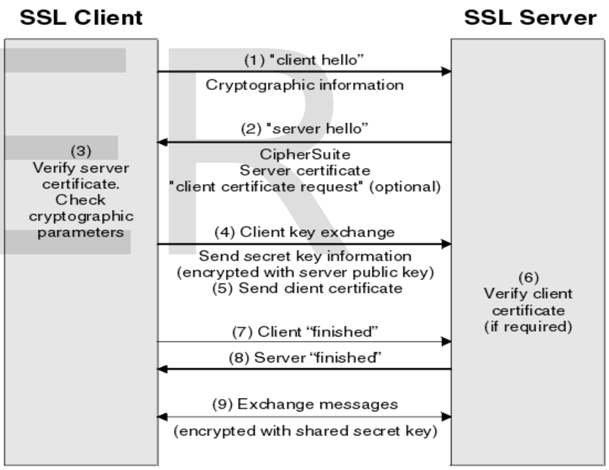


Fig. 2. TLS/SSL handshake

2.5 Iptables / Ip6tables

Iptables and Ip6tables are open source software, often preinstalled with many linux distributions (netfilter.org, 2002). They are used to decide what should be done with incoming or outgoing packets, they can e. g. be dropped, accepted or logged. Decisions can be made based on several cases. For example which interface or what port that is used, whether the connection was already established or not. It is also possible to filter by protocol. They are often used as firewall software but can also be extended to other purposes e.g. routing and forwarding of data traffic. Iptables work with IPv4 while Ip6tables work with IPV6.

2.6 STUN

STUN (**Session Traversal Utilities for NAT**) is a standardized set of methods and a network protocol to allow an end host to discover its public IP address if it is located behind a network address translator (NAT). STUN is necessary when a client is located behind a NAT device that has been configured to allow "PORT FORWARDING" to the client.

Due to the limited ipv4 address space, most computers obtain Internet access through modem connected to a NAT. The NAT device's external interface is configured with a public IP address. The computers behind the NAT, on the other hand, are invisible to hosts on the Internet as they each communicate only with a private IP address. Port forwarding allows remote computers (for example, computers on the Internet) to connect to a specific computer or service within a private local-area network (LAN)

3 Design and Implementation

3.1 Design

The technical design is kept as simple as possible (fig. 3). All necessary client information (user name, password, host name, domain name, last IP) is stored in a MySQL database. The client is able to update its IP in the database through a java program that will be installed on the client system. The java program will transmit the login username, password and the new IP address to the PHP web application hosted on the apache web server. The transmission is protected by SSL/TLS (https). The account data is evaluated through a RADIUS server (FreeRADIUS) that verifies the login data against the one stored in the database. If the username and password provided are correct, the current IP of the user is updated in the MySQL database and a script is evoked. The script is provided with host name, domain name and IP address of the user that just logged in. Then the script communicate with the DNS service, deletes the old entry for the combination of host name, domain name and adds a new entry with the new IP address.

The reason a RADIUS server was used is that it is easy to add other ways of authentication at a later time. For example, RADIUS has built in support for authentication through cryptographic certificates and there are RADIUS clients available for most platforms.

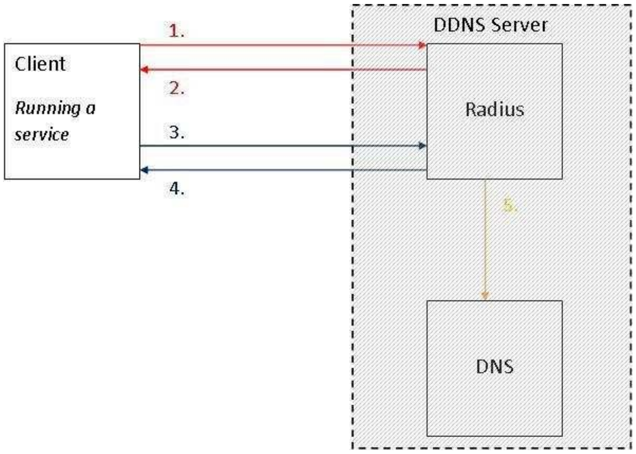


Fig. 3. Implementation of the solution

1. Client asks for connection
2. RADIUS and client set up a secure connection
3. Client provides user name and password
4. Authentication OK
5. Update of clients new IP in the DNS database

3.2 Implementation

The implementation was started with basic services like DNS. Later on FreeRADIUS with MySQL, and Apache were set up. Lastly, a firewall was added. The network was setup in dual stack mode so as to be able to process both IPv4 and IPv6 client request. This means the Apache web server and the BIND DNS is configured to be working in dual stack mode. Finally, the firewall software (iptables and ip6tables) are configured to allow necessary network connections to be made and prevent unnecessary connections that may cause security breach on the network.

3.2.1 IPv6 Support

IPv6 support in a Dynamic DNS server needs to be defined in two ways.

1. DNS server software needs to support IPv6 related resource records

2. DNS server should support IPv6 transport (IPv6 packet processing). These two factors are independent from each other but server is said to be IPv6 supported only if both the requirements are met.

Operating system and underlying hardware support for IPv6 characterizes the IPv6 transport ability of the Server. BIND version or any other DNS software being used defines the ability of processing resource records. IPv6 support in Ubuntu Linux 12.04 is quite satisfactory. From kernel version 2.4 IPv6 support is adapted to the Linux distributions. (Linux kernel 2.3 does not support IPv6). The current BIND version supports IPv6.

3.2.2 DNS

The DNS server was implemented at a basic level. The ISC BIND which is widely used for the implementation of DNS was used for the DNS part of the server. BIND already made package for Ubuntu 12.04. Also the DNS servers were implemented redundant, i.e. there is a master and a slave server (Albitz& Liu, 2001). The default configuration only contains standard data. There was need to create a new zone that we gave the name warning. In order to allow for updates the zone property "allow-update" was set accordingly. The two servers synchronize each other periodically and on changes in the DNS table (semi-complete, 2008), i.e. when the master gets updated through the script it immediately sends the new entry to its peer. For this synchronization a TSIG key is used to increase security.

3.2.3 NTP

Both DNS and DHCP require the time difference on the different servers to be within a certain limit, especially the TSIG includes a time stamp which has to be accurate. In order to ensure DHCP and DNS to work correctly it is necessary to synchronize the time on the servers, therefore they were set up as peers using NTPD version 4.2.2 (ntp.org, 2000). The local clock on Server2 acts as reference clock. Since ordinary PCs are used as test computers the Local Clock in this case is the bios clock, which is the built-in clock in the hardware of the computer. This allows Server1 and Server2 to have the same time even though the real time will be inaccurate. As the test network is completely isolated from the Internet and there is no access to a better time source, this is sufficient. The accuracy of the bios clock is very poor, but the goal is to keep a global clock in synchronization and not to have accurate time.

3.3 RADIUS

Since it is aimed to have a perfect match between security and simplicity of implementation, it was decided to make use of the RADIUS protocol (Rigney, C. et al, 2000). It provides security features like encrypted passwords and can make use of certificates. Furthermore its accounting abilities can be used to limit access to the service and track the use of the service. This gives enough security for most small to medium sized networks.

3.3.1 FreeRADIUS

FreeRADIUS was used as RADIUS server (FreeRADIUS, 2006). First an older Ubuntu 12.04 package (version 1.3) was installed, but it did not support certain password features, e.g. plain text passwords that are good for testing. Consequently it was updated to version 2.1, which created some inconsistency in the system, as example some language warnings turned up when installing new packages. Still there were some features regarding certificates missing. The source code was consulted and needed files were copied.

MySQL was used to store user data (e. g. passwords, user names). FreeRADIUS is also able to interact with Oracle and a variety of other databases, but they were not that easily available as MySQL (Elmasri&Navathe, 2007). FreeRADIUS provided a script which made it easy to add all necessary tables to the database.

3.4 The Server Agent

This is made up of a single PHP script (dologin.php). It enables the clients to update their IP address on the DNS server. It receives the account login details and the new IP address from the java program installed on the client system. It authenticates the account against FreeRADIUS server. After the user is authenticated, its IP address is obtained, and the data containing host name and domain name is extracted from a table in the MySQL database.

Finally, it determines the version of the IP address (IPv4 or IPV6) and sends the information to the appropriate script. The file dologin.php now becomes responsible also for updating the IP address in the MySQL database.

3.5 The DNS Update Shell Script

The script is responsible for updating DNS server. It uses nsupdate and a TSIG key. It only updates the actual zone. First the DNS entry is cleared then the new DNS entry is added. The information needed is provided by the dologin.php script below:

```
#!/bin/bash USER="$1" IP="$2" HOSTNAME="$3"
DOMAIN="$4"
# update DNS entry
nsupdate<< EOF
server <server-ip>
keyupdatekey<some-secret-key>
zone warning
update delete $HOSTNAME.$DOMAIN
update add $HOSTNAME.$DOMAIN. 600 IN A $IP
send
quit
```

The script files “DNS-update IPv4”

```
#!/bin/bash USER="$1" IP="$2" HOSTNAME="$3"
DOMAIN="$4"
# update DNS entry
nsupdate<< EOF
server <server-ip>
keyupdatekey<some-secret-key>
zone warning
```

```
update delete $HOSTNAME.$DOMAIN
update add $HOSTNAME.$DOMAIN. 600 IN AAAA $IP
send
quit
```

The script files “DNS-update IPv6”

3.6 Apache

The java program on the client system will connect to the PHP application on the Linux server to update the IP address of the client system. PHP application engine needs a web server to run. Apache is an open-source HTTP web server for modern operating systems including Linux, UNIX and Windows. Nowadays Apache is the most widely available web server for Linux. The **Apache2** web server package in Ubuntu Linux was installed. The web server was also configured for https connection and IPv4 / IPv6 Dual Stack Mode. The dual stack mode enables the web server to process request from both IPv4 and IPv6 clients.

3.7 Certificates

The login interface uses a certificate. Since there is no Internet access available for the servers, it is impossible to sign them by any trusted agency. If the client decides to accept the certificate the normal key negotiation process takes place.

3.8 Firewall

In order to enhance the level of security for the servers a firewall was implemented on each server. It was chosen to use iptables for IPV4 and ip6tables for IPV6, as both came already installed with the system. The configuration was kept simple, all incoming ports that were used are allowed and all others are blocked (LinuxHelp, 2002). There is no filter on outgoing ports. Allowed incoming ports include:

Table 2: Firewall

Port	Service
80	http (website)
443	SSL (certificate)
53	DNS
22	SSH (administration)

3.9 The User Agent

A java based agent program was developed as part of the DDNS solution. This automated agent software will be installed on the client. The agent would automatically detect a change of the client public IP address and immediately send a request to update the address record in the DNS server through the RADIUS server. The application makes use of the open source JSTUN library ("JSTUN" is a Java-based STUN implementation) to detect the client public IP address change.

4 Results and Discussion

4.1 Test and Verification

The solution was implemented in a simple network connected to the internet and tested with clients in different IPv4 and IPv6 networks. The result was as expected. Clients were able to successfully update their DNS entry in DDNS server when their IP address changes.

User Agent Interface:

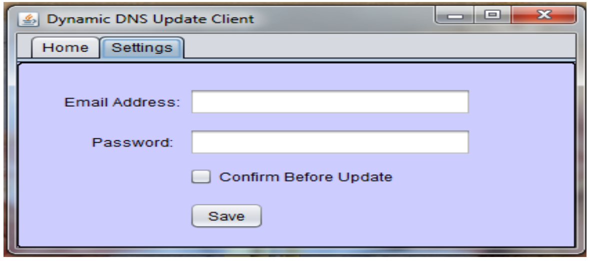
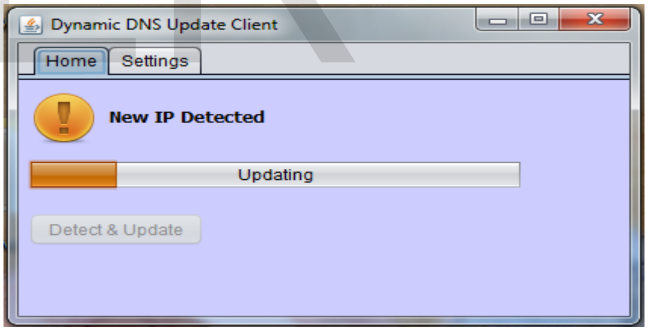


Fig. 5. Interface to accept email address and the password to the account attached to the users domain name.

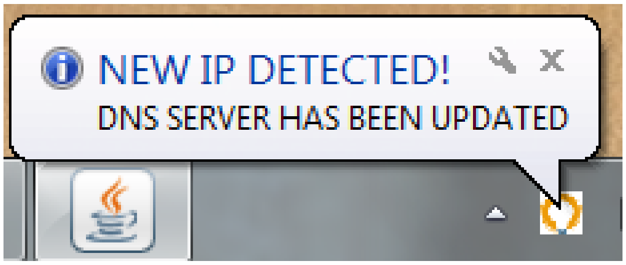
User can also indicate if the agent should confirm before updating the domain name with the new IP address detected.



New IP Detected

Updating

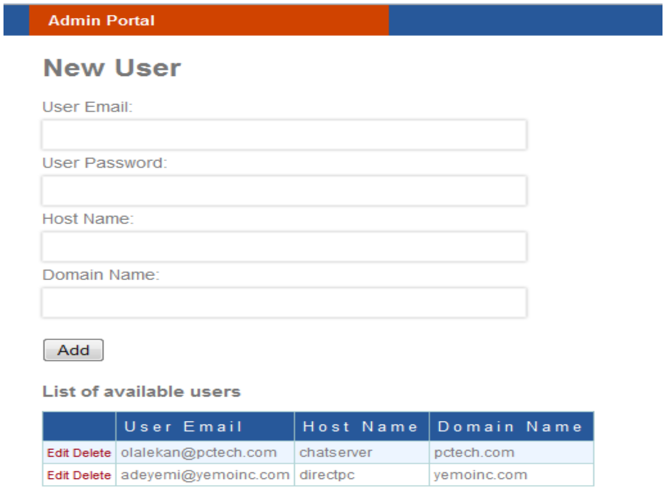
Fig. 6. Interface for manual detection and update of new IP address on the server.



NEW IP DETECTED!
DNS SERVER HAS BEEN UPDATED

Fig. 7. Interface that indicates the automatic detection and update of a new IP address to the user.

Admin Interface:



	User Email	Host Name	Domain Name
Edit Delete	olalekan@pctech.com	chatserver	pctech.com
Edit Delete	adeyemi@yemoinc.com	directpc	yemoinc.com

Fig. 8. Admin Interface

5 Conclusions

This paper presents a DNS solution that permits automatic changes of DNS entries in the name server. In that way a new IP address can be mapped to the FQDN, enabling the server administrator to make use of dynamic IP addresses. This solution would allow clients to keep their hostnames even if the IP configuration is done dynamically especially when they are moved in smaller environments. Also, this solution provide dual stack mode that supports both IPv4 and IPv6 which makes it suitable for lots of devices connected to the internet using IPv6.

5.1 FutureWork

There are several functions we have thought of during the course of this study that would add more functionality to our solution. However, due to time constrains this was limited to the functions we have implemented.

5.1.1 Redundancy

If the web server would fail, no client can connect to it and therefore it is not able to update its DNS entry in case its IP address changes. Resulting in no client service is reachable, as the IP address stored in DNS entries is outdated. The same problem arises if the RADIUS server would fail. This problem can be solved by adding a redundant solution for both the web service and the RADIUS service. Depending on the implementation of the redundancy there might be a need for some round robin DNS entries and a watchdog updating the entries if one service is down.

Another problem arises if the database fails, the user cannot be identified although the web interface is reachable and RADIUS is giving correct replies but probably a "user cannot be identified" error message. To enable redundancy for the database, one solution would be to have two database instances that are synchronized somehow. The synchronization could be implemented as a master/slave system or by having an external master for example.

References

[1] Arends R. et al., 2005. DNS Security Introduction and Requirements. [Online] The Internet Society. Available at: http://www.ietf.org/rfc/rfc4033.txt

[2] Bind9 DNS server howto https://help.ubuntu.com/community/BIND9ServerHowto

[3] Cisco, 2004, How Does RADIUS Work? [Online] (Updated 19 Jan 2006) Available at: http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a00800945cc.shtml

[4] FreeRADIUS, 2013, FreeRADIUS Wiki [Online] (Updated 21 January 2013) Available at: http://wiki.freeradius.org/Home

[5] Heinlein, P., 2005. Failover with ISC DHCP. [Online] (Updated 28 Mars 2009) Available at: http://www.madboa.com/geek/dhcp-failover/

[6] Internet Systems Consortium, 1997.Internet Systems Consortium. [Online] (Updated 18 May 2009) Available at: https://www.isc.org/

[7] LinuxHelp, 2013. Firewall Script. [Online] (Updated 18 May 2009)Available at: http://www.linuxhelp.net/guides/iptables/

[8] IPv6 Firewall For Linux. [Online]Available at: http://www.linuxhelp.net/guides/iptables/

[9] Linux Howto. Bind9 DNS Ipv6 support [Online Available at: http://tldp.org/HOWTO/Linux+IPv6-HOWTO/hints-daemons-bind.html

[10] MySQL, 2013. MySQL :: The world's most popular open source database. [Online] Available at: http://www.mysql.com/

[11] netfilter.org, 2010. netfilter/iptables project homepage - The netfilter.org project. [Online] Available at: http://www.netfilter.org/projects/iptables/index.html

[12] netfilter.org, 2013. netfilter/ip6tables project homepage - The netfilter.org project. [Online] Available at: http://ipset.netfilter.org/ip6tables.man.html

[13] ntp.org, 2000.Reference clocks. [Online] (Updated 18 May 2009) Available at: http://www.ntp.org/ntpfaq [Accessed 11 August 2013].

[14] What is LAMP? [Online] (Updated 16 November 2013) Available at: http://en.wikipedia.org/wiki/LAMP_(software_bundle)

[15] Rigney, C. et al., 2000. RFC2865 - Remote Authentication Dial In User Service (RADIUS). [Online] The Internet Society Available at: http://www.ietf.org/rfc/rfc2865.txt

[16] "STUN" - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translation (NAT) http://en.wikipedia.org/wiki/STUN

[17] "JSTUN" - Java Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translation (NAT) http://jstun.javawi.de

[18] Ubuntu 12.04 WIKI. Ipv6 support [Online] (Updated 14 June 2013) Available at: https://wiki.ubuntu.com/IPv6