Requirements and Analysis

Analyzing the requirements from the client for the project Internet Server IPv6 readiness

Client Organization: Murdoch University
Group Name: IT08 Ltd.
Submission Date: October 23, 2015
Version: 3.5

Table of Content

Table of Figures

Executive Summary

The purpose of this report is to record the requirements from the clients including functional and non-functional requirements. The software that will be developed will be based on these requirements. It includes introduction about the opportunities and problems as well as objectives and goals of the system, scope of the solution including functionalities, user interface, system output, constraints and the hardware as well as software requirement. After that, a list of functional requirements sorted by importance, list of non-functional requirements including documentation, hardware consideration etc. and the diagrams that represent the requirements. After reading this document, one will be clear of the client's needs for the product. It outlines the solutions, makes clear of the objectives, clarifies the requirements including both functional parts and non-functional parts, and describes the preliminary design of the product by means of different diagrams from different aspects of the product. The methods we used in developing this software include online research, software development tools and reviewing documents.

1 Introduction

1.1 The Authorization

In week 3 our client Dr. Sebastian Zander from Murdoch University authorized us to develop a software that can measure the performance of servers on the Internet from different parts of the world. The focus of the measurement is comparing the performance of the servers in IPv6 versus IPv4. As of 2011, the last usable public IPv4 address has been assigned. IPv6 was designed to expand the network addresses to an almost unlimited pool. Many servers have been capable of using IPv6 address to communicate while maintaining network communication using IPv4. At this stage we will develop a prototype software to measure the level of IPv6 readiness of servers from different parts of the world to obtain a statistical concept of this using graphics.

1.2 The Purpose

The software is to measure the level of IPv6 readiness of servers comparing to IPv4 referring to performance. Its main purpose is to present the level of IPv6 readiness and the performance of servers using graphical charts. It also provides the functions of searching individual records.

1.3 Problems and Opportunities

1.3.1 Problems

Networks need to be upgraded to use IPv6 addresses in all of the Internet but many servers and networks are still not IPv6 available. Many IPv6 available servers encounter a lot of problems when dealing with visitors using transition technologies to access their IPv6 addresses. Performance using IPv6 is sometimes not as well as that using IPv4.

1.3.2 Opportunities

By using this software, the network pictures of current IPv6 availabilities are clear.

Performance of IPv6 comparing to that of IPv4 can be clearly recognized. Servers that encounter IPv6 problems can be diagnosed to find where the problem lies.

1.4 Objectives and goals

The objectives include:

Probing servers: A list of Internet servers will be probed using different kinds of methods via IPv4 address and IPv6 address to measure their performance

Storing data: The data of the servers and the results of the measurements will be stored into the database automatically and can be searched by keywords

Graphically representing data: The level of IPv6 readiness and performance of servers can be represented by calculating the records in the database

Network diagnosis: Servers that can’t be reached using IPv6 can be diagnosed to find out where the problem lies

1.5 Methods

1.5.1 Online Research

The author will go through several webpages about the mechanism of IPv6 transitional technologies and the existing methods of realizing the functions.

1.5.2 Software Development Tools

The team will use Microsoft Visual Studio 2015 C# to develop the software and XAMPP MySQL as a database server. A program called MySQL connector is needed to connect the database to the programming environment.

1.5.3 Reviewing Documents

By reviewing the documents from previous similar projects and the document from the client, the team members try to depict the requirements of the client accurately.

2 Solution Outline

2.1 Scope of Solution

The software can gather information from servers and display them graphically. It also provides the function of searching related information about individual servers in the database. It cannot affect the configuration or performance of the servers. The methods for probing are from existing legal methods so there is no legal issue incurred from the probing action. The software can only test the servers and cannot be used as remote controlling software. It is only a prototype software so error is expected though largely it is bug free. As a result users are advised to follow the instructions when using the software.

2.2 System Functionality

The system will be able to measure performances of the servers using both IPv4 and IPv6 and graphically display the results in a clear and concise format. The probing will use different kinds of ports and protocols. It will also provide the function of searching server information inside the database by keywords and diagnosing problems in case where the server can’t be reached via IPv6 connection.

2.3 User Interface

The User Interface (UI) will provide buttons and textbox to let users choose what to display. It can accept individual server or a list of servers for probing. The textbox can also be used for inputting search keywords and the search result will be displayed in the same window of the software. It will also provide graphical output to statistically display the results. A designated field will be used for the graphical area. The result of diagnosis will also be provided in the same window.

2.4 System Output

The system will output statistical graphics based on data stored in the database during the process of sampling performance of the servers. Diagnosis results will be output in the text field. Search results of the servers inside the database will also be displayed in the text field.

2.5 Constraints

The program is only a prototype so it can only provide required functions proposed by the client. Users are required to input by following the instructions. It is anticipated that the software will only run on Windows platform. It requires the testing machine to have IPv6 connectivity using either native address or transitional address for probing servers using IPv6. Users are strongly recommended to install the software other than using only the executable inside the installation folder to execute the program.

2.6 Hardware & Software Environment

The recommended hardware requirement for the proposed system is:

- CPU - Intel core i5 2.3G Hz
- RAM - 8 GB
- Disk volume - 512 GB

The operating system must be Microsoft Windows 32 bit or 64 bit. The platform can be Windows XP SP3, Windows Vista, Windows 7, Windows 8 and Windows 8.1. It is tested that the Windows tablet operating system Windows 8.1 with Bing also supports it. .Net framework and database server are necessary to run the program.

3 Functional Requirements

The list of functional requirements will sort them out by criticality and include: name, description, criticality, technical issues, cost & schedules, risks and dependencies. This section contains the requirements that were gathered from the client. Each of these Requirement can subdivided into different values (such as Low, Medium & High if required) when used to measure certain metrics such as criticality, technical issues, cost, risks, dependencies, etc. Since this is an unpaid project, cost will be measured in terms of time spent on it (resource-time, hours spent by us as a team to build the prototype and final version) and not monetary terms.

3.1 Probing servers across the Internet

- **Description:** Probe and analyze servers being able to respond to both IPv4 and IPv6 requests. Different kinds of services will be tested in the process of probing in different regions in Australia and other continents as well. The probing methods include Internet Control Message Protocol (ICMP) Ping test, Hyper Text Transfer Protocol (HTTP) GET request, Simple Mail Transfer Protocol (SMTP) connection test and Domain Name Service (DNS) queries. The team members will choose a sample subset of servers for testing first from different continents. These servers will be tested by sending requests using both IPv4 addresses and IPv6 addresses. Their performance is determined by the latencies we measure during the tests. These tests will be undertaken in regular time interval. The probing time will start from 12:00 am or can be specified by the client and the test will be conducted every day in a week. At any of these times we will test the sample servers using the above methods. The probing of servers will be conducted sequentially one by one by loading a list of servers.

  Individual server can be probed by clicking on a button and the results of the probing can be displayed on screen. A list of servers can also be probed by clicking another button with the results displayed on screen too. Automatic probing using command line is also available where the user can specify the list of servers to probe, the location of the log file and the timeout for probing. All the probing results will be saved into a log file. The location of the log file can be chosen by the user in list probing or in command line when starting the program. When using the Graphic User Interface (GUI) to probe, a checkbox is provided for users to perform fast probing that has the same effect of specifying the timeout value in the command line.

- **Criticality:** This is the essential function of the software and should be placed at the front of the requirements. All other requirements are based on this.
- **Technical Issues:** The testing machine must possess a native IPv6 address or obtain an IPv6 address through transition technologies. These technologies including 6to4 (6to4, n. d.), Teredo (Teredo tunneling, n. d.), Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) (ISATAP, n. d.) and 6over4 (6over4, n. d.). We will use native IPv6 addresses realized by using tunnel broker. If the native IPv6 address can't be realized, we will use 6to4 among these technologies as it has the least security concerns and should represent the average level of measurement.
- **Cost & Schedules:** 3 weeks for module development,1 week for module testing and 2 weeks for server probing
- **Risks:** 1.The programmer is sick and will leave for a period of time.

  2. The programmer's computer is unable to be used anymore.

  3. The network environment has changed, thus the IPv6 capability of the test machine is lost.

- **Risk mitigation methods**:

  1. Find someone else who is capable of programming for help.

  2. Backup data frequently on tertiary storage devices to mitigate possible threat.

  3. The same as above.

- **Dependencies:** This function is the utmost important one so every other task is dependent on this. It doesn't depend on any other requirement. This module can be developed simultaneously with the creation of database listed in function 4.2.

3.2 Processing the raw data and storing them into database

- **Description:** After measuring these servers, the test data will be pre-processed by removing any kinds of measurement errors and stored into the database. The test data include: latencies for all kinds of tests, server's Fully Qualified Domain Name (FQDN), server’s IPv4 and IPv6 addresses, server's country, server's continent, time when test is conducted and the types of services the server is providing. The network latencies will be processed so that the measurement errors would be removed. At the same time, a database will be built to store the data automatically described above.
- **Criticality:** This function is also crucial and mandatory. It will process the data from function 4.1 and store the data which can be used in the next function.
- **Technical Issues:** When the request is timed out, the result of the latency will become zero. This is one kind of measurement error. The other kind of error is the server doesn't respond at all and there is no connection timeout native in the operating system. A maximum response timer will be used to solve this which leads to the result of the measurement becomes the maximum response time. This also needs to be solved.
- **Cost & Schedules:** 2 weeks for database creation,1 week for module testing and 2 weeks for processing and storing data
- **Risks:** 1.The programmer is sick and will leave for a period of time.

  2. The programmer’s computer is unable to be used anymore.

- **Risk mitigation methods:**

  1. Find someone else who is capable of programming for help.

  2. Backup data frequently on tertiary storage devices to mitigate possible threat.

- **Dependencies:** Part of it should be finished first which is the creation of the database. After this function 4.1 can start to probe simultaneously with processing and storing data.

3.3 Searching the database and graphically representing the data

- **Description:** After the data is stored into the database, the end users will be able to search the database using keywords. By clicking on the "search" button, the related table records will be displayed in the same window. The range of search can be narrowed down by choosing the continent and the country in two dropdown lists. When the search results are displayed, additional keywords can be input to filter the displayed results. The fields in the search result can be selected and copied by clicking and dragging the mouse using hotkey combination “Ctrl+C”. The results can also be sorted by clicking on the title of the column to display the results in ascending or descending order.

The user will also be able to view the data represented by statistical charts. There will be:

1. A button that can display the percentages of IPv6 available servers in each continent using column chart where the abscissa is the continent and the ordinate is the percentage;
2. A button that can compare the percentages of IPv6 available servers in different countries using column chart where up to 4 countries can be selected by dropdown lists for comparison. The abscissa will be the country and the ordinate is the percentage;
3. A button that can display the percentages of IPv6 available servers for different server types using column chart. The abscissa will be the server type and the ordinate is the percentage;
4. A dropdown list which lists the service types that can be used to reach the server. When certain service is chosen, only servers that have that type of service will be taken into calculation when clicking the above three buttons.
5. A button that can sort out all the data of one server to display the performance of the server on different weekdays using multi-series column chart. The abscissa will be the weekdays and the ordinate will be the average latency of the server on that day. Two series will be provided for both IPv4 and IPv6.

- **Criticality:** This function is the last of all mandatory functions. It is also an essential function to be realized so that users can have an impression of the IPv6 readiness level of today’s network which is the main purpose of building this software.

- **Technical Issues:** The Structured Query Language (SQL) statements must be aligned with the output of the C# methods. It is up to the programmer to deal with the possible problems that can arise among this process of liaison.

- **Cost & Schedules:** 2 Weeks for module development and 1 week for testing
- **Risks:** 1.The programmer is sick and will leave for a period of time.

  2. The programmer’s computer is unable to be used anymore.

  3. The database software becomes a paid version which is quite expensive.

- **Risk mitigation methods:**

  1. Find someone else who is capable of programming for help.

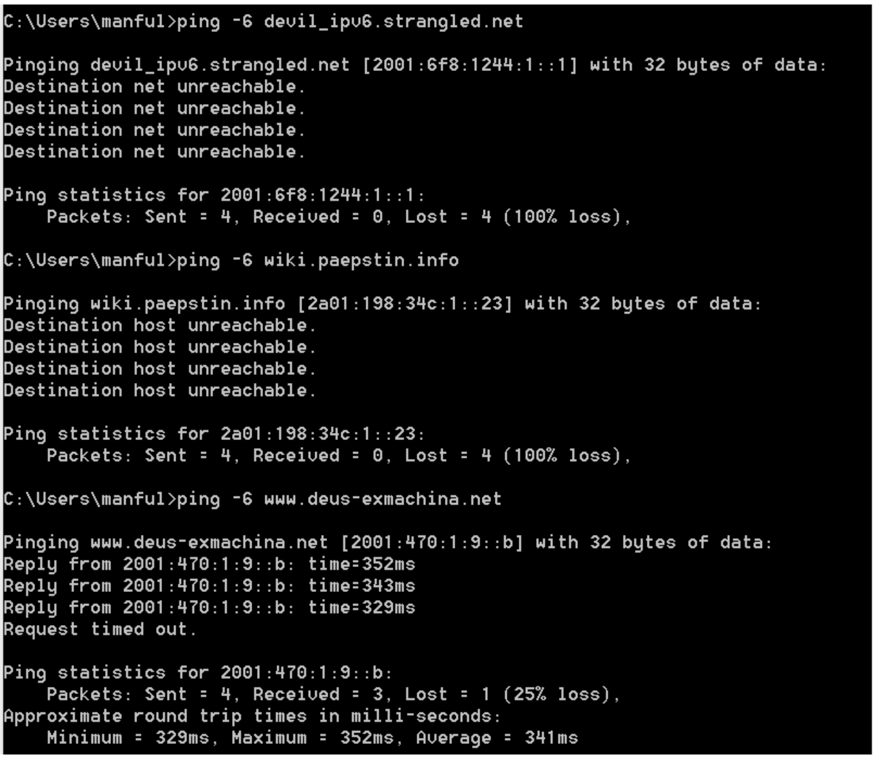
  2. Backup data frequently on tertiary storage devices to mitigate possible threat.

  3. Do not upgrade the software before the end of the project.

- **Dependencies:** This function depends on function 4.2 to finish creating the database and storing the data. The development of this module can start as long as the database is created.

3.4 Diagnosis toward IPv6 unavailable networks

- **Description:** An additional requirement from the client states when there is server that can't be reached using IPv6, there should be diagnosis which indicates where the problem lies. Possible problems are: server is not IPv6 available, server's network is not IPv6 available and server's configuration for IPv6 is wrong. If the server is not IPv6 available, there should be no AAAA record on the DNS server when resolving the server's FQDN. Apart from this, the output of the command "ping" can indicate whether the server's network is unreachable or the server is unreachable which represent the latter two situations. The result of the diagnosis will be stored into the database during the process of probing. The status of the server reflects the result of diagnoses.
- **Criticality:** This function is optional.
- **Technical Issues:** Client is worried if the result of this diagnosis is correct by relying on the output of the command “ping”. The following picture states the possible outputs of “ping” using IPv6 addresses:



```
C:\Users\manful>ping -6 devil_ipv6.strangled.net

Pinging devil_ipv6.strangled.net [2001:6f8:1244:1::1] with 32 bytes of data:
Destination net unreachable.
Destination net unreachable.
Destination net unreachable.
Destination net unreachable.

Ping statistics for 2001:6f8:1244:1::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\manful>ping -6 wiki.paepstin.info

Pinging wiki.paepstin.info [2a01:198:34c:1::23] with 32 bytes of data:
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.

Ping statistics for 2a01:198:34c:1::23:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\manful>ping -6 www.deus-exmachina.net

Pinging www.deus-exmachina.net [2001:470:1:9::b] with 32 bytes of data:
Reply from 2001:470:1:9::b: time=352ms
Reply from 2001:470:1:9::b: time=343ms
Reply from 2001:470:1:9::b: time=329ms
Request timed out.

Ping statistics for 2001:470:1:9::b:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 329ms, Maximum = 352ms, Average = 341ms
```

Figure 1 - Diagnose using "Ping"

In the figure above, one can see the “ping” command can provide required diagnoses indicating different kinds of problems.

- **Cost & Schedules:** 1 week for module development and 1 week for testing
- **Risks:** 1.The programmer is overburdened and he ignored this function.

  2. The programmer’s computer is unable to be used anymore.

- **Risk mitigation methods:**

  1. Find someone else who is capable of programming for help.

  2. Backup data frequently on tertiary storage devices to mitigate possible threat

- **Dependencies:** As this is optional, it should be implemented after all the above mandatory functions have been implemented and tested completely.

3.5 Import & Export Database

- **Description:** An additional requirement from the client states when installing the software and its requirements for the first time the initial database which is necessary for the software to function normally should be imported automatically. The project team also decides to add a function that can connect to the database if the database server software is not already running. The user will also be able to import the database from a database file and export the database to a file for backup.
- **Criticality:** This function is optional.
- **Technical Issues:** The software needs to know the path of the database software to perform database import and export. Users may install the database software into any places. To manage these situations a configuration file is used to record the location of the database software.
- **Cost & Schedules:** 1 week for module development and 1 week for testing
- **Risks:** 1.The programmer is overburdened and he ignored this function.

  2. The programmer's computer is unable to be used anymore.

- **Risk mitigation methods:**

  1. Find someone else who is capable of programming for help.

  2. Backup data frequently on tertiary storage devices to mitigate possible threat.

- **Dependencies:** As this is optional, it should be implemented after all the above mandatory functions have been implemented and tested completely.

3.6 Tables & Charts Export

- **Description:** Another additional requirement from the client states it is ideal if the tables and charts displayed by the software can be exported in correct formats. Tables can be exported to Comma Separated Value (CSV) format and charts can be exported to Portable Network Graphics (PNG) format or Enhanced Meta File (EMF) format. When charts are saved as EMF format the resolution won’t change in case the picture is scaled to another level.
- **Criticality:** This function is optional.
- **Technical Issues:** The client’s initial requirement was exporting the chart to Scalable Vector Graphics (SVG) format with the same quality as that of the EMF format file. The project team couldn’t find any way to realize that. After communication the client agreed to use EMF format as a substitution.
- **Cost & Schedules:** 1 week for module development and 1 week for testing
- **Risks:** 1.The programmer is overburdened and he ignored this function.

  2. The programmer’s computer is unable to be used anymore.

- **Risk mitigation methods:**

  1. Find someone else who is capable of programming for help.

  2. Backup data frequently on tertiary storage devices to mitigate possible threat

- **Dependencies:** As this is optional, it should be implemented after all the above mandatory functions have been implemented and tested completely.

4 Non-functional Requirements

4.1 Documentation

The documentation includes:

- **Requirement and analysis document:** the audience of this is the client and the team members. Both should go through all the requirements regularly.
- **Project management plan:** the audience of this is the project team members and the supervisor. Both should go through the plan regularly and if there is change it should be approved before being made available into the plan.
- **Design document:** the audience is the client and the programmer. Both should inspect the design diagrams and if there is modification the client should be notified immediately.
- **Deliverable task breakdown statements**: The audience is the supervisor and the unit coordinator. These will demonstrate the acknowledgement of the team members to their percentages of contribution to individual deliverables.
- **User document**: The audience is the user. These will guide the user to install and use the software without the help of the developer.
- **Test plan**: The audience is the supervisor and the unit coordinator. These will document the tests towards the software by the developer, project team members and the user in detail.

4.2 Hardware Consideration

The recommended hardware requirement for the proposed system is:

- CPU - Intel core i5
- RAM - 8 GB
- Disk volume - 512 GB

The minimum hardware requirement is:

- CPU - Intel Pentium 4
- RAM - 512 MB
- Disk volume - 20 GB

It can be used on any 32 bit or 64 bit CPU.

4.3 Performance Characteristics

Following are the performance characteristics under normal circumstances:

- Installation time: < 5 seconds
- Startup time: < 3 seconds
- Response time: < 1 second for common use; < 10 seconds in average for probing a server

Under extreme conditions (CPU & RAM are highly consumed and network is congested):

- Installation time: <15 seconds
- Startup time: < 13 seconds
- Response time: < 11 second for common use; < 20 seconds in average for probing a server

As the software can pre-process the data, there is no constraint on the data's size or capacity. Because the probing of servers will be done sequentially, the total time for probing the list of servers can be significant. The probing is thus recommended to be conducted at the beginning of the day. It is estimated that the time for probing 100 servers will be 20 minutes.

4.4 Error Handling and Extreme Conditions

The system returns errors should there be invalid input on fields controlled by users. The error message will specify where the problem is. Operations not affected by the error will continue. In extreme conditions, such as extremely long input strings, the system will catch the error and return it. The system is unlikely to crash in any case.

4.5 System Interfacing and Compatibility

The input of the software comes solely from the end user or the tester. The output of the software will be displayed only inside the window of the software. Generally there is no restriction on the input format. However, users are discouraged to input invalid characters on purpose. The software can only be run in Microsoft Windows platforms including: Windows XP SP3, Windows Vista, Windows 7, Windows 8 and Windows 8.1. It can also be run on Windows tablet. Windows .Net framework is not necessary to run the program. For testing IPv6 activities, the testing machine must be configured with transitioned IPv6 address in case the network provider doesn't support IPv6 routing. The program doesn’t need to be executed with administrator’s account.

4.6 Quality Issues

The user is advised not to run the program during high CPU consumption period or when the remaining RAM is too low. In both situations, the program will respond extremely slowly. The system will trap faults and display the error message should there be errors including situations where Internet is not connected, user input is invalid and various kinds of situation. If there were unlikely situation where the program has to restart, the restart time will be less than 1 second. The system is not portable and the installation file is necessary in order to keep the program complete without malfunctioning. The only operating system it can run on is the Windows workstation series 32 bits and 64 bits.

4.7 System Modifications

The system can be improved in the future on the following parts:

- **User interface:** the user interface will be more user friendly with better options in the future.
- **Functions for representing data:** More functions representing the data using different types of charts can be added.
- **Error handling:** More smart error handling functions can be added.

4.8 Physical Environment

There is no special requirement for the location where the computer will run the software. The computer can run the software no matter it is inside a quiet room or outside in the street. The network environment can be either wireless or wired. In extremely high temperature or high frequency of vibration, the software can still endure it until the computer can't. The same condition applies in high magnetic fields.

4.9 Security Issues

As the software stores data that are retrieved from the public servers using common methods, the data themselves are not for profit and not copyright protected. It is not necessary to implement security to the database regarding to confidentiality. The software is only a prototype so it is not prepared for the malicious users to tamper with the data intentionally. However it can be a target software for penetration tests. No physical security issue arises for this software.

4.10 Resource Issues

The database will be backed up every two days during the probing stage. The programmer will be responsible for the backup, installation and system maintenance. Once probing the servers is accomplished the data can then be manually accessed by using select queries in the database.

5 Diagrams

5.1 Use Case model

Use case diagram represents the relationships among the system and the actors. The asterisk represents “many” relationship in the association. It roughly represents the possible interactions among the external factors and the internal system functions. “<<Include>>” means the database is included in that function. “<<Use>>” means the database is used for storing data. The following is the use case diagram:

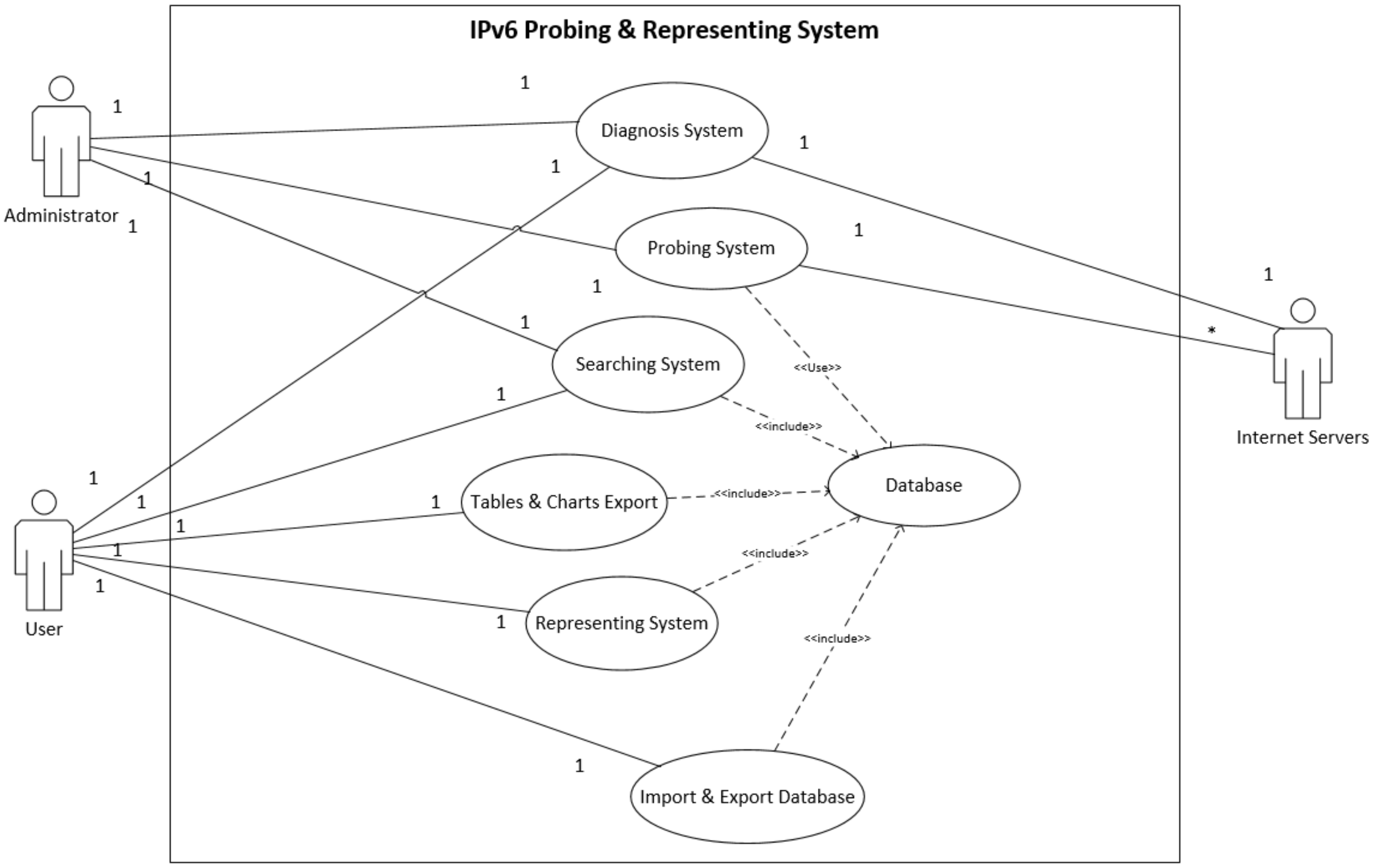


Figure 2 - Use case diagram

5.2 Context Diagram

Similar to use case diagram but is drawn in a different way. It represents the dataflow between the whole system and the other sources and sinks. No data store is there. The context diagram is like following:

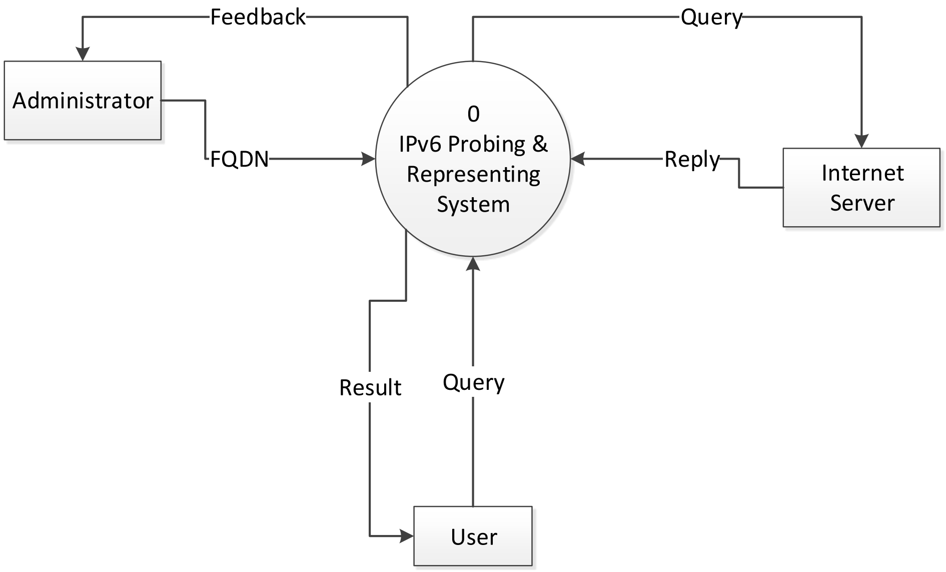


Figure 3 - Context diagram

5.3 Level-0 Data Flow Diagram (DFD)

Level-0 DFD decomposes the system in context diagram to be different functions of the system. No new source or sink will be there. Data stores will be added. Data among sources/sinks and the data store must go through the processes. The Level-0 DFD is like following:

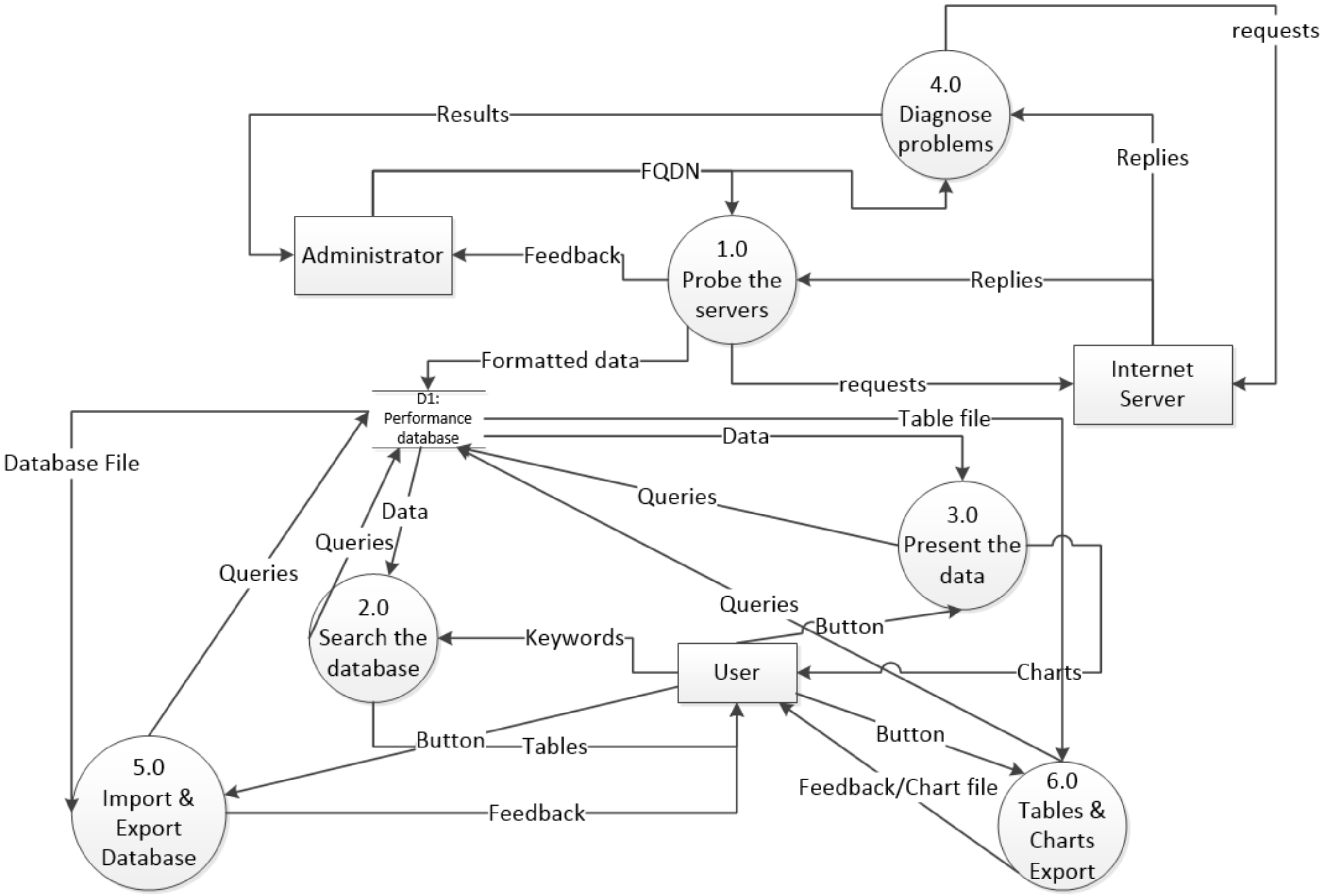


Figure 4 - Level-0 data flow diagram

5.4 Conceptual Entity-Relationship Diagram (ERD)

E-R diagram represents the relationships of different kinds of data in database. The following is the conceptual E-R diagram which has been normalized in the first degree with multiple values for one record removed and is in the second degree too because no table has composite primary key:

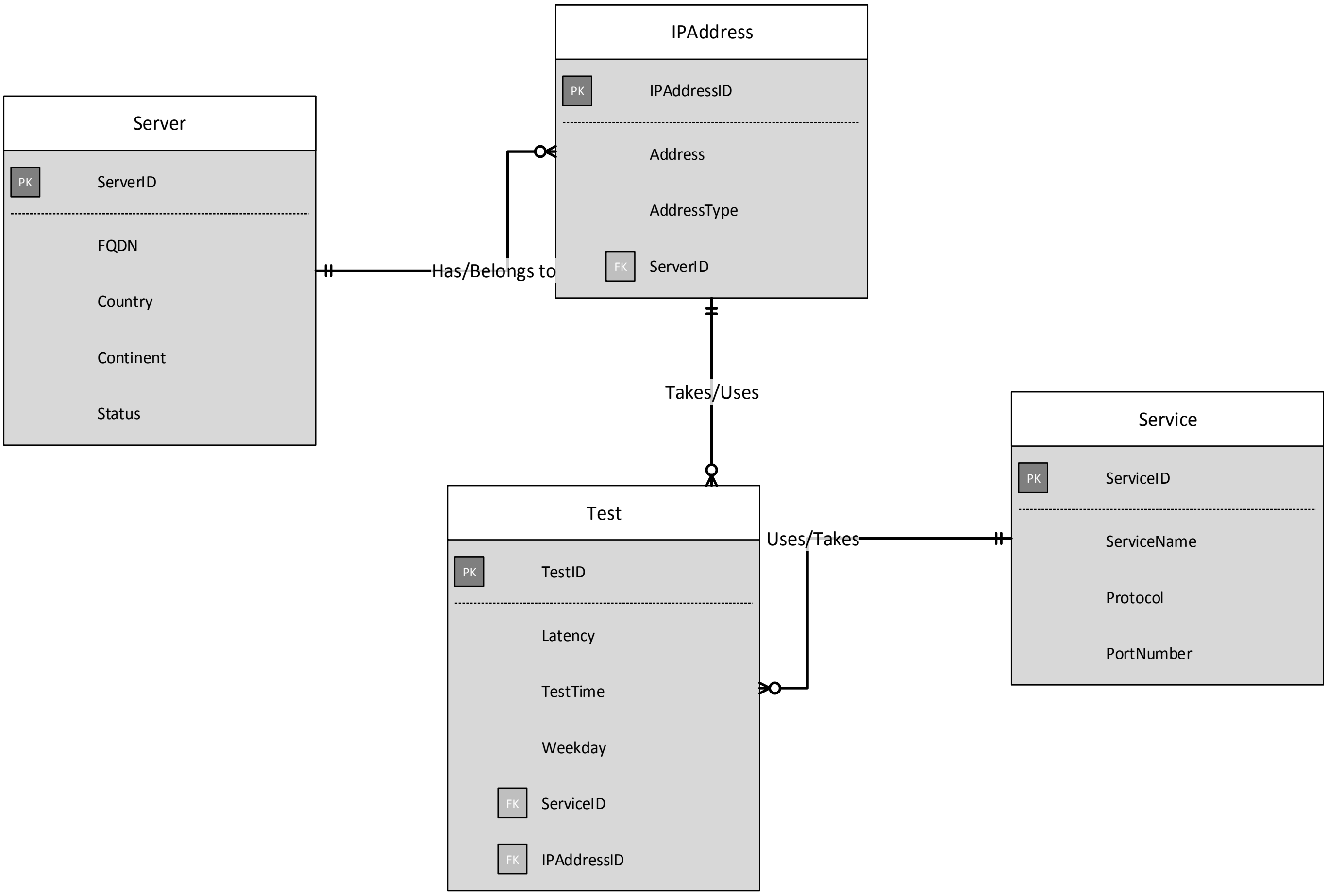


Figure 5 - Conceptual E-R diagram

5.5 Network Topology Diagram

Network topology represents the physical locations of the servers and the system. It also shows how the IPv6 transition technologies are used in the Internet. In the scenario of using tunnel broker to set up a native IPv6 address, the local endpoint of the tunnel is local machine with IPv4 address and remote endpoint is the tunnel broker's server that routes the packets using IPv6 addresses in the IPv6 network to the IPv6 available server. Packets from the server will be routed via the broker's server and get to local machine via the tunnel. Local machine must have public IPv4 address to set up the native IPv6 address. If local machine uses private addresses using Network Address Translation (NAT) Teredo must be used. In our development environment the machine will use public IPv4 address so the first option is to set up a tunnel broker which will allow the use of IPv6 native address. The advantage of using IPv6 native address is ensuring that the reply from remote server can be routed to our local machine avoiding the situation where the reply can't come back leading to the judgment that it is not reachable using IPv6. The following is the topology:

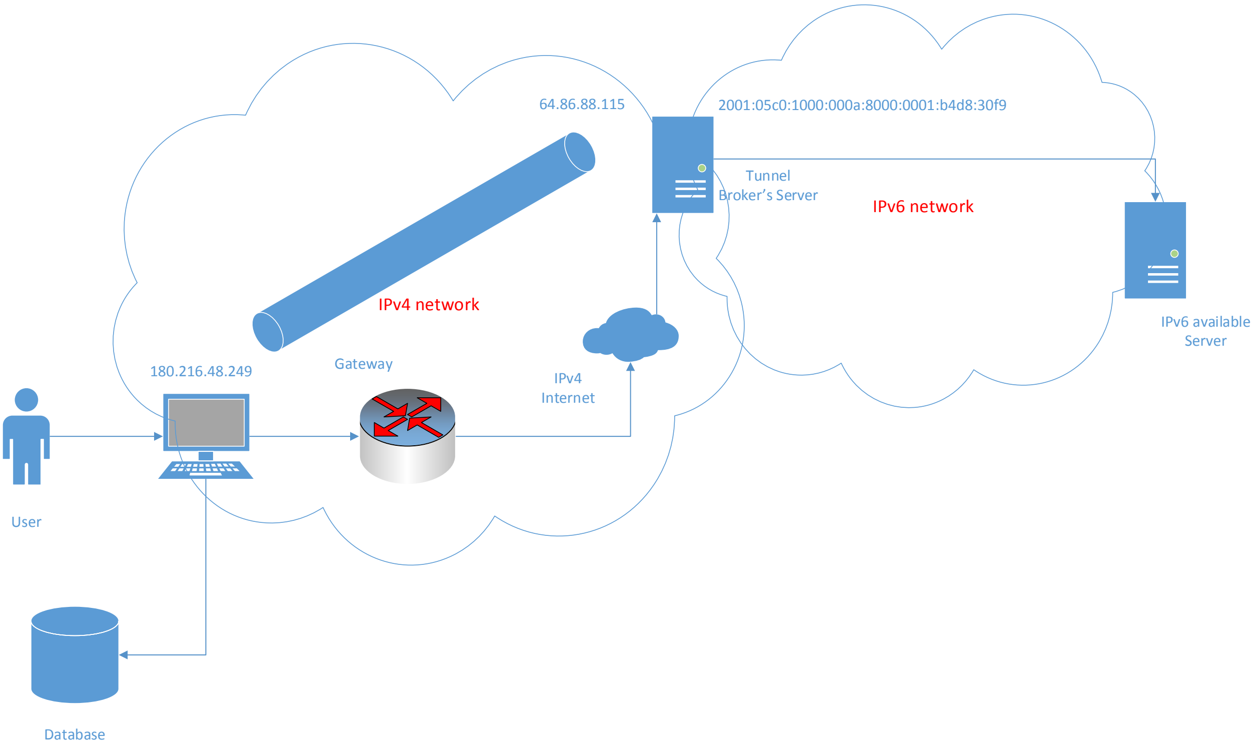


Figure 6 - Native IPv6 address

In case the tunnel broker's server is not reachable, the fallback method is to use 6to4. In the topology, the testing machine has a 6to4 translated IPv6 address. If the server also uses a translated IPv6 address, the packets will go through the IPv4 Internet and arrive at each end points using derived IPv4 addresses. If the server uses a native IPv6 address, the packets will go through a 6to4 relay router which uses an IPv6 anycast address 2002:c058:6301::c058:6301 whose route is advertised by the Internet Service Provider (ISP). (6to4, n. d.). The 6to4 technology uses the nearest relay router so it is not reliable to measure the performance of the servers. The following is the network topology:

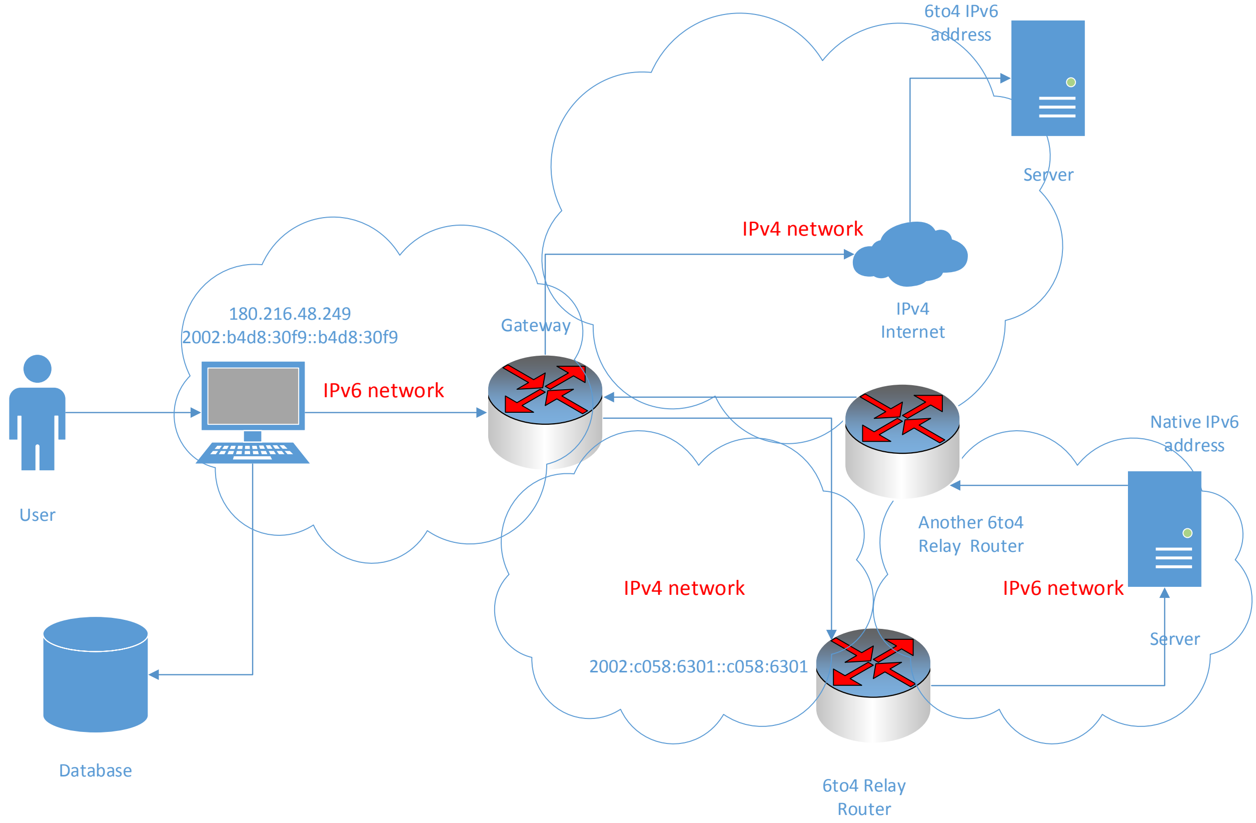


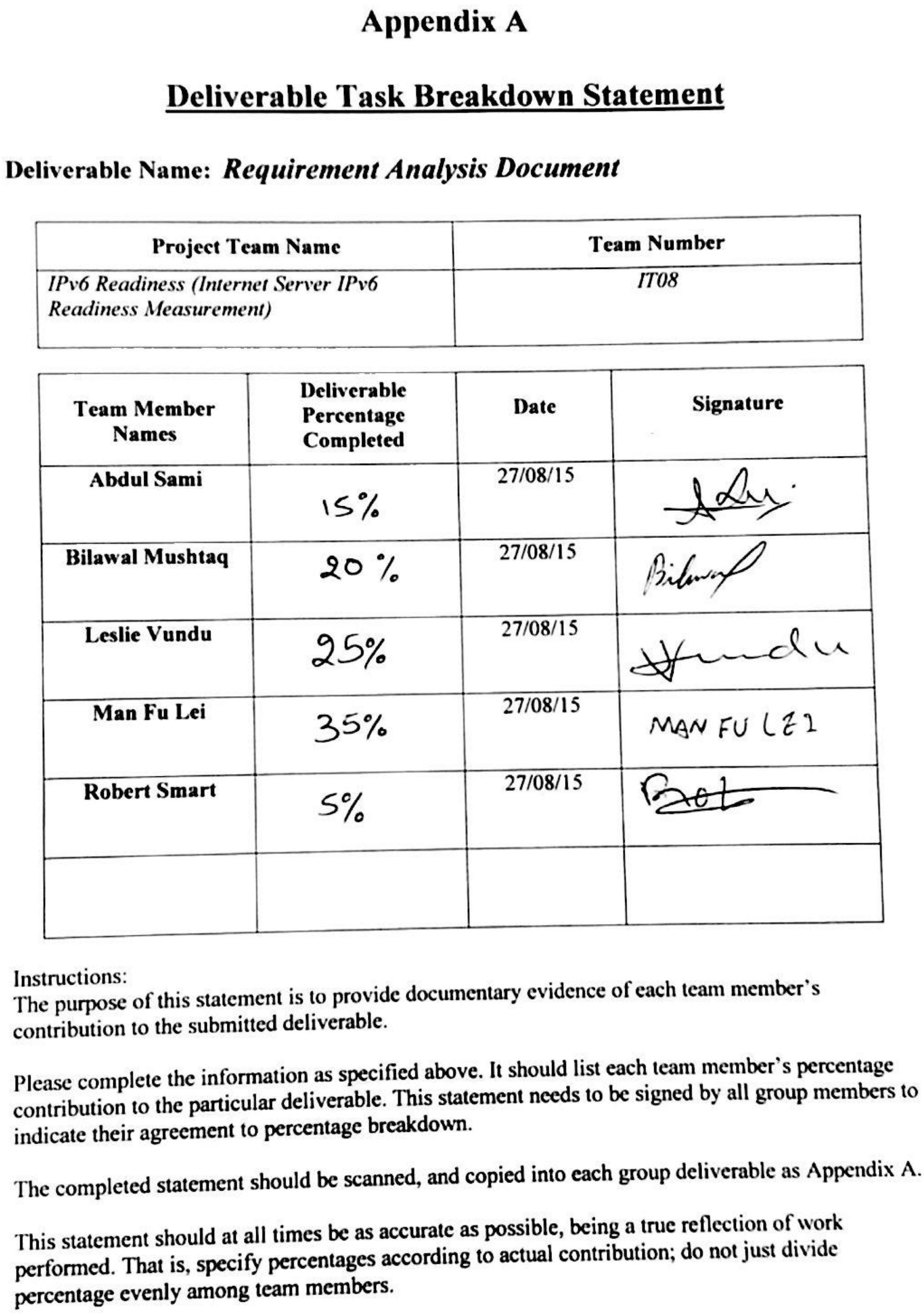
Figure 7 – 6to4 Network topology

6 Conclusions

This document is used to record the requirements. It discusses different areas involved in requirement analysis including methods we used in developing the software, the outline of the solution, functional requirement, non-functional requirement and preliminary design diagrams. The software will be able to provide a user interface that can probe a list of Internet servers, store the information of the servers and the measurement results, search the database for result and provide charts representing the requirements from the client. It will also provide the diagnosis function when there is a server that can’t be reached using IPv6. The main constraint of the software is it can only be used on Microsoft Windows platforms. The user is also required to input information by following the instructions. Preliminary design diagrams representing the data flow and the data structure in the database are also presented. From the network diagram, one can see the mechanism of the software.

7 Appendices

A. Deliverable Task Breakdown Statements



Appendix A

Deliverable Task Breakdown Statement

Deliverable Name: ***Requirement Analysis Document***

Project Team Name	Team Number
IPv6 Readiness (Internet Server IPv6 Readiness Measurement)	*IT08*

Team Member Names	Deliverable Percentage Completed	Date	Signature
Abdul Sami	15%	27/08/15	
Bilawal Mushtaq	20 %	27/08/15	
Leslie Vundu	25%	27/08/15	
Man Fu Lei	35%	27/08/15	MAN FU LEI
Robert Smart	5%	27/08/15	

Instructions:
The purpose of this statement is to provide documentary evidence of each team member's contribution to the submitted deliverable.

Please complete the information as specified above. It should list each team member's percentage contribution to the particular deliverable. This statement needs to be signed by all group members to indicate their agreement to percentage breakdown.

The completed statement should be scanned, and copied into each group deliverable as Appendix A.

This statement should at all times be as accurate as possible, being a true reflection of work performed. That is, specify percentages according to actual contribution; do not just divide percentage evenly among team members.

B. Copy of Client Documents

Internet Server IPv6 Readiness

As of August 2015 the pool of unallocated IP version 4 (IPv4) addresses is almost depleted. IP version 6 (IPv6), the next version of the IP protocol, has virtually unlimited addresses, but the roll-out of IPv6 has been very slow. One measure to identify how much of the Internet is IPv6-capable, is to test the IPv6 capability of servers.

Even if IPv6 is supported by a server, it may not work as well as IPv4, e.g. there may be higher network delay with IPv6. To compare the two, we need to measure the performance of accessing a server with IPv4 and IPv6.

The goal of this project is to develop a prototype that can measure and collect data for IPv6 readiness of servers (including measuring baseline performance with IPv4), and present the results in table/graphical form. From the presentation one should be able to identify what fraction of the Internet's server infrastructure is IPv6 ready, difference between different countries and server types, and performance of IPv6 relative to IPv4.

The project should develop prototype software than can do the following:

* Probe a chosen representative sample of servers across the whole Internet. This means servers in different countries as well as different types of servers (e.g. web servers, email servers, domain name servers).

* Probe servers with IPv4 and IPv6 so we measure and compare the performance of both.

* Probe servers in regular time intervals so we can measure the IPv6 readiness over time.

* Store the measurement data in a database so all measured data can be accessed and searched

easily.

* Pre-process the raw data as necessary, for example remove measurement errors.

* Graphically display the IPv6 readiness information in a nice easy to understand form.

Ideally, in cases where IPv6 does not work, some diagnostics should be performed to inform us where the problem lies, for example is a server not IPv6 capable, is the network not IPv6 capable, is there misconfiguration of IPv6. This information should also be presented in the GUI.

8 List of References

6over4. (n. d.). In *wikipedia*. Retrieved August 21, 2015 from https://en.wikipedia.org/wiki/6over4

6to4. (n. d.). In *Wikipedia*. Retrieved August 21, 2015 from https://en.wikipedia.org/wiki/6to4

ISATAP. (n. d.). In *wikipedia*. Retrieved August 21, 2015 from https://en.wikipedia.org/wiki/ISATAP

Teredo tunneling. (n. d.). In *wikipedia*. Retrieved August 21, 2015 from https://en.wikipedia.org/wiki/Teredo_tunneling#Exposure

9 Glossary

6over4: is an IPv6 transition mechanism meant to transmit IPv6 packets between dual-stack nodes on top of a multicast -enabled IPv4 network

6to4: is an Internet transition mechanism for migrating from IPv4 to IPv6, a system that allows IPv6 packets to be transmitted over an IPv4 network (generally the IPv4 Internet) without the need to configure explicit tunnels

AAAA record: also known as "IPv6 address record", maps a hostname to a 128-bit IPv6 address in the Domain Name System (DNS) Internet Authentication Service

Anycast: is a network addressing and routing methodology in which datagrams from a single sender are routed to the topologically nearest node in a group of potential receivers, though it may be sent to several nodes, all identified by the same destination address

Context Diagram: is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it

Data Flow Diagram (DFD): is a graphical representation of the "flow" of data through an information system, modelling its process aspects

Database normalization: is the process of organizing the attributes and tables of a relational database to minimize data redundancy

Database: is a collection of data that is organized so that its contents can easily be accessed, managed, and updated

Domain Name Service (DNS): is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network

Entity-Relationship Diagram (ERD): is a data model for describing the data or information aspects of a business domain or its process requirements

Fully Qualified Domain Name (FQDN): is a domain name that specifies its exact location in the tree hierarchy of the Domain Name System (DNS)

HTTP GET request: Requests data from a specified resource

Hyper Text Transfer Protocol (HTTP): is an application protocol for distributed, collaborative, hypermedia information systems

Integrated Development Environment (IDE): is a software application that provides comprehensive facilities to computer programmers for software development

Internet Control Message Protocol (ICMP): is an error-reporting protocol network devices like routers use to generate error messages

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP): is an IPv6 transition mechanism meant to transmit IPv6 packets between dual-stack nodes on top of an IPv4 network

IPv4: is the fourth version in the development of the Internet Protocol (IP)

IPv6 transition technology: is a technology that facilitates the transitioning of the Internet from the Internet Protocol version 4 (IPv4) infrastructure in use since 1981 to the successor addressing and routing system of Internet Protocol Version 6 (IPv6)

IPv6: was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion

Network port: serves as an endpoint in an operating system for many types of communication

Network protocol: serves as a language of communication among computing devices

Penetration test: is a software attack on a computer system that looks for security weaknesses, potentially gaining access to the computer's features and data

Ping: is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time

Probing: will show you which protocols are being used on your network, which hosts are sending and receiving data etc.

Simple Mail Transfer Protocol (SMTP): is an Internet standard for electronic mail (email) transmission

Structured Query Language (SQL): is a special-purpose programming language designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system

Teredo: is a transition technology that gives full IPv6 connectivity for IPv6-capable hosts that are on the IPv4 Internet but have no native connection to an IPv6 network

Tunnel Broker: is a service which provides a network tunnel that can provide encapsulated connectivity over existing infrastructure to another infrastructure

Use Case diagram: is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved