Electronics and Computer Science

Faculty of Engineering and Physical Sciences

University of Southampton

Ziqian Qin

02/12/2023

Multi-protocol server-side program for IPv6 tunnel broker

Project supervisor: Graeme Bragg

A project progress report submitted for the award of

BSc Computer Science

Abstract

As Internet Service Providers roll out IPv6, the pace of development varies, leaving some networks isolated from IPv6 networks. In response to this challenge, the project proposes a tunnel broker server to facilitate the transition process from IPv4 to IPv6 protocol and provide a way for IPv4-only networks to access resources only available to IPv6. The proposed tunnel broker will support multiple protocols, ensuring compatibility with divergent network environments.

This report discusses the project goal and the literature reviewed during the first semester. Additionally, it includes sections showcasing the current progress of the project and outlines plans for the remaining work.

**Statement of Originality**

* I have read and understood the [ECS Academic Integrity](http://ecs.gg/ai) information and the University’s [Academic Integrity Guidance for Students](https://www.southampton.ac.uk/quality/assessment/academic_integrity.page).
* I am aware that failure to act in accordance with the [Regulations Governing Academic Integrity](http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-regs.html) may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
* I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

***You must change the statements in the boxes if you do not agree with them.***

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

|  |
| --- |
| **I have acknowledged all sources, and identified any content taken from elsewhere.** |

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

|  |
| --- |
| **I have not used any resources produced by anyone else.** |

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

|  |
| --- |
| **I did all the work myself, or with my allocated group, and have not helped anyone else.** |

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

|  |
| --- |
| **The material in the report is genuine, and I have included all my data/code/designs.** |

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

|  |
| --- |
| **I have not submitted any part of this work for another assessment.** |

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements*.* You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

|  |
| --- |
| **My work did not involve human participants, their cells or data, or animals.** |

*ECS Statement of Originality Template, updated August 2018, Alex Weddell* [*aiofficer@ecs.soton.ac.uk*](mailto:aiofficer@ecs.soton.ac.uk)

Contents

[1 Project Goals 4](#_Toc152534979)

[2 Background and report of literature search 5](#_Toc152534980)

[3 Report on Technical Progress 6](#_Toc152534981)

[4 Plan of remaining work 8](#_Toc152534982)

# Project Goals

While Internet providers are transitioning from IPv4 to IPv6, their pace differs. According to statistics from the IPv6 test [2023], approximately half of Internet users in the United Kingdom are still on IPv4-only networks as of 2023. And classic 6in4 tunnel faces issues with NAT as it cannot properly forward proto-41 packets [4]. This project strives to facilitate the transition process by the development of a tunnel broker server with multi-protocol support to address the issue.

IPv6 transitioning technologies fall into categories of dual stack, encapsulation, and translation [2]. Since tunnelling operates on the existing IPv4 network and is independent of the network infrastructures, this project shall be deployable by Internet service providers with minimal changes to existing infrastructure. Furthermore, third-party vendors with sufficient IPv6 addresses may also deploy the program to provide IPv6 access.

The project strives to construct a program capable of creating and managing (modifying, deleting) tunnels while handling connecting requests from clients concurrently. Tunnel protocols that are planned to be supported by the program include basic 6in4(proto-41) protocol and 6in4 with heartbeat protocol. In addition, the program shall also support AYIYA (Anything in Anything), which allows clients sitting behind NAT to use the service. The program will also implement TIC (tunnel information and control) protocol as the one used to manage network tunnels. Such a program will be built with security in mind. One integral part will be filtering packets from invalid addresses and spoofed addresses.

# Background and report of literature search

When working on this project, various literature works were reviewed. RFC documents make up a large portion of the literature and have a significant influence on the project's design decision. The literature reviewed covers the discussion around the Tunnel Broker, configured tunnels and necessary security measurements.

Tunnel broker is designed for automatically configuring tunnels and allows to spread of the load across multiple tunnel servers [3]. However, this project simplifies the architecture so that message passing is not needed between the tunnel broker and tunnel server. The drawback is that both the tunnel broker and tunnel server will be running on the same node. Thus, load balancing across tunnel servers becomes unachievable. Due to the nature of the service, the node running the tunnel broker will be required to have access to both IPv4 and IPv6 addresses, which may be present on two distinguished interfaces.

Several security measurements was proposed by the standard of 6in4 tunnel.

# Report on Technical Progress

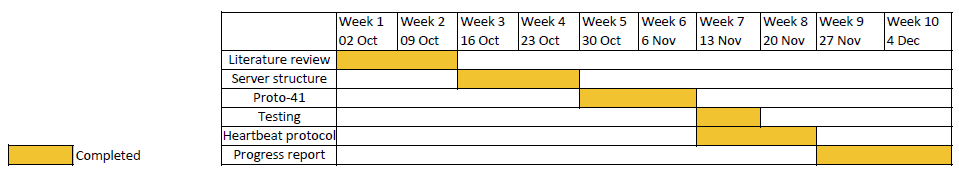


Figure 1

Figure 1 shows the Gantt chart upon finishing this report. A working server that supports the 6in4 protocol was developed over the 2-month period. Currently, the program is capable of capturing and identifying proto-41 packets along with the ability to encapsulate IPv6 packets to proto-41 packets. Once receiving a proto-41 packet, the program will decapsulate it and validate it against a set of rules to minimise security risks before sending out the IPv6 packet. When receiving IPv6 packets that are destined for one of the tunnel clients, the program will encapsulate the packet and then forward it into the tunnel.

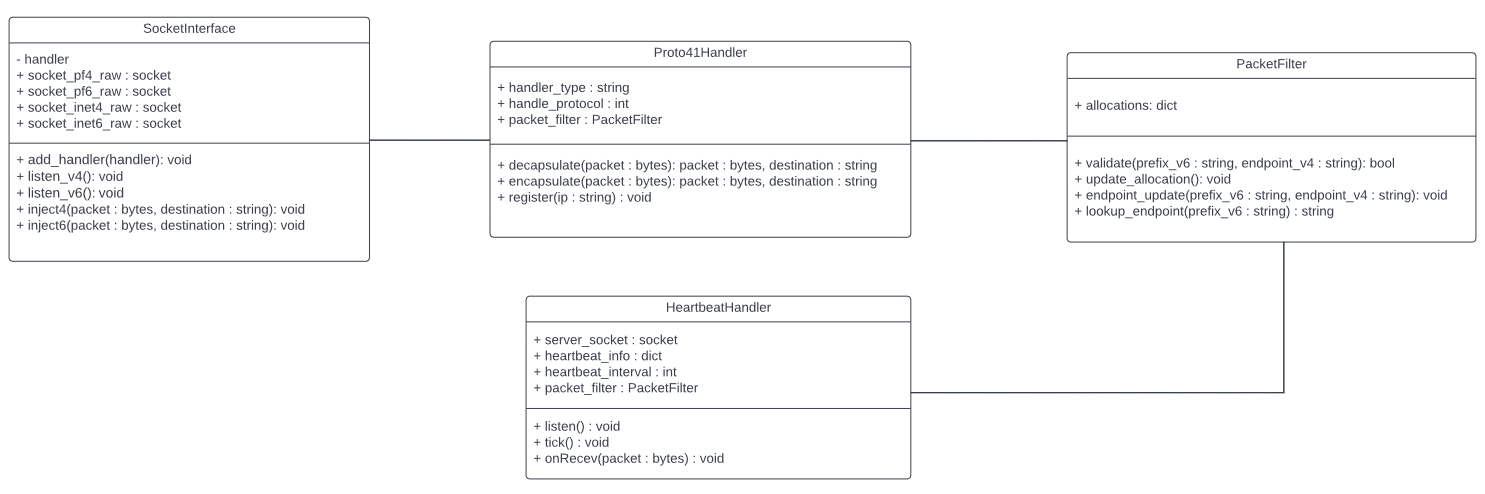


Figure 2

Figure 2 shows the current UML class diagram of the design. The program is written in Python. And it uses raw sockets for sending and receiving proto-41 packets. The project sticks with an object-oriented approach and makes functionalities modular to be used across different protocols.

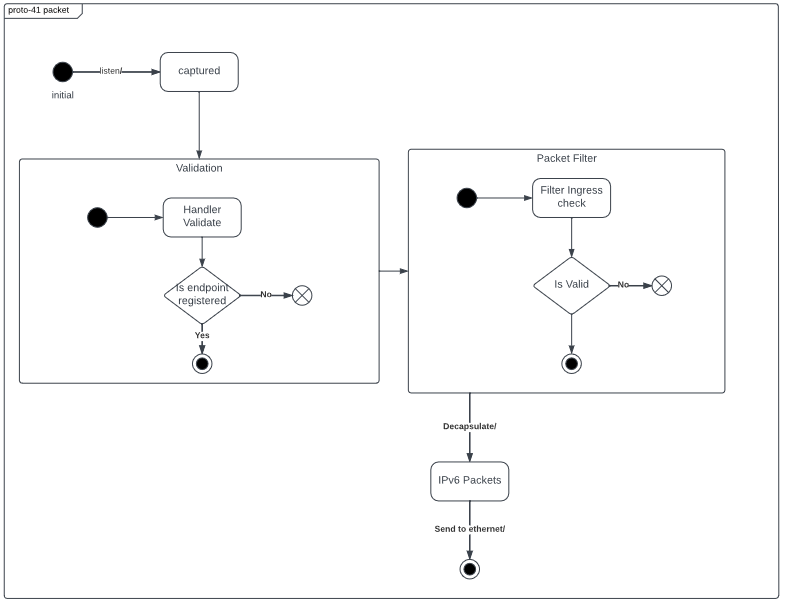


Figure 3

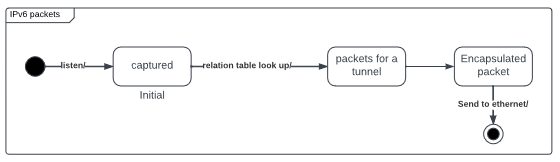


Figure 4

Figure 3 show the state diagram once the program receives a proto-41 packet. And figure 4 shows the state diagram once the program receives a proto-41 packet for a tunnel that is configured to use basic 6in4 protocol.

The implementation of heartbeat protocol was finished but it can only integrate into the system once other components are complete. Further details will be discussed in the next section.

# Plan of remaining work

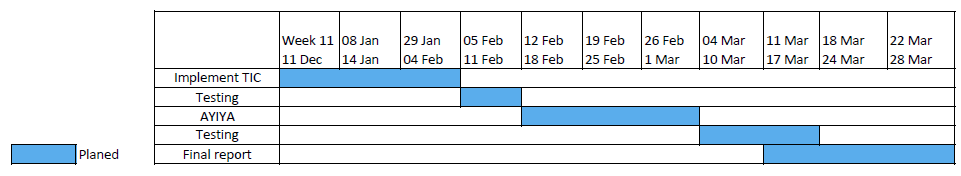


Figure 5

Figure 5 outlines the roadmap for completing the remaining tasks. The schedule prioritises the implementation of the TIC protocol, given its pivotal role in testing the heartbeat protocol. And once it is finished, the program is almost complete except for the AYIYA protocol. As the program’s functionality is being modulated, integrating a new protocol into the system should be relatively straightforward.

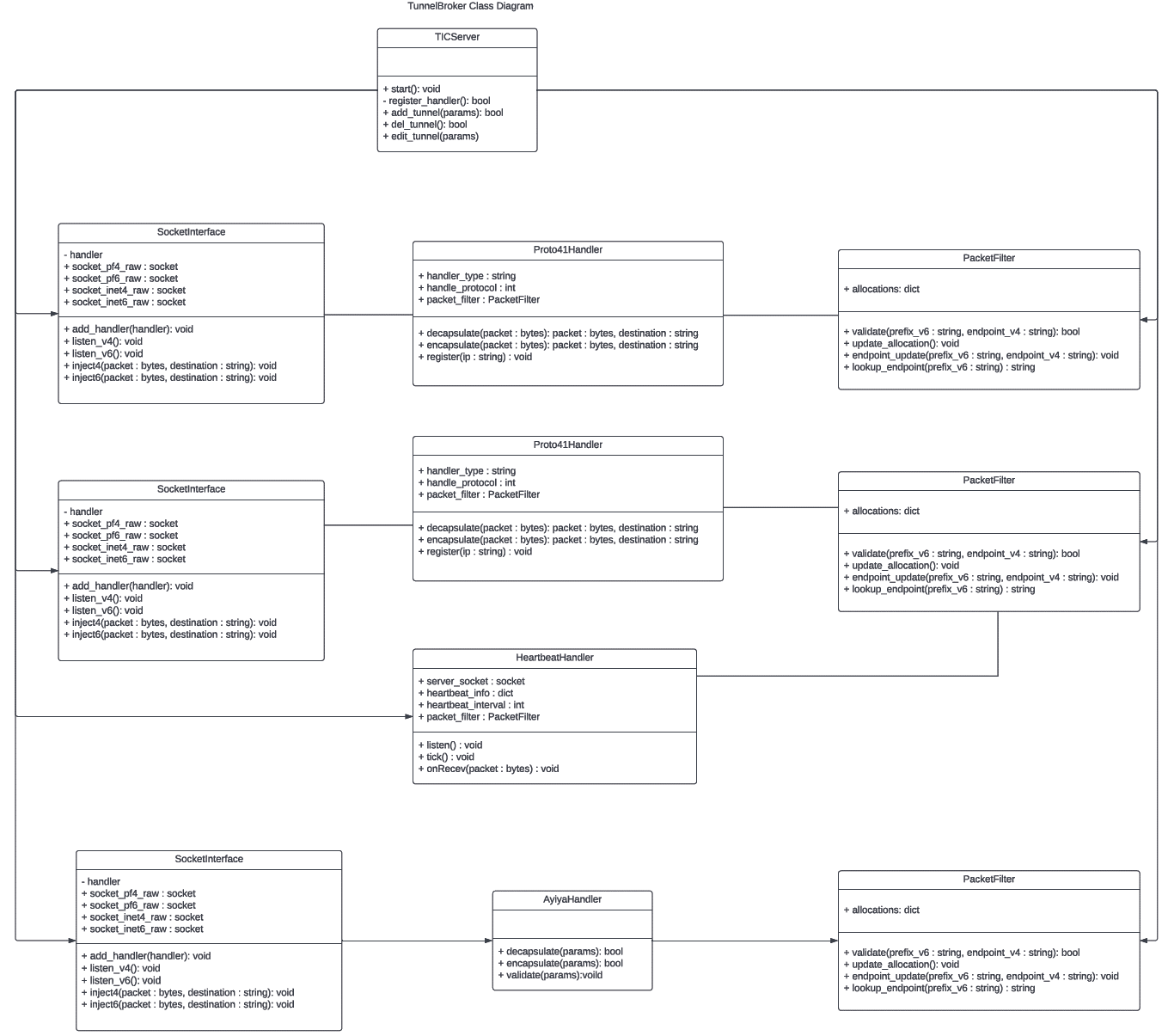


Figure 6

Additionally, Figure 6 illustrates the designed class relationship of the program.

References

[1] IPv6 test. 2023. IPv6 test - Statistics for United Kingdom. Retrieved from <https://ipv6-test.com/stats/country/GB>

[2] Marius Georgescu, Liviu Pislaru, and Gábor Lencse. 2017. Benchmarking Methodology for IPv6 Transition Technologies. Retrieved from <https://www.rfc-editor.org/info/rfc8219>

[3] Dr Paolo Fasano, Dr Ivano Guardini, Alain Durand, and Domenico Lento. 2001. IPv6 Tunnel Broker. Retrieved from <https://www.rfc-editor.org/info/rfc3053>

[4] S. J. M. Steffann, Iljitsch van Beijnum, and Rick van Rein. 2013. A Comparison of IPv6-over-IPv4 Tunnel Mechanisms. Retrieved from <https://www.rfc-editor.org/info/rfc7059>

[5] Robert E. Gilligan and Erik Nordmark. 2005. Basic Transition Mechanisms for IPv6 Hosts and Routers. Retrieved from <https://www.rfc-editor.org/info/rfc4213>