University of Padova

Department of Information Engineering

ICT for Internet and Multimedia

Master Thesis on

Enhancing IPv6 Adoption in Linux Clients within IPv4 Infrastructures through Teredo Tunneling: A Comprehensive Study and Implementation Approach

Supervisor
Prof. Andrea Zanella
University of Padova

Doctor Luca Finotti
Alessio Celin

Master Candidate
Nasim Bahari

Student ID
2050798

Academic Year
2023-2024
27/02/2024

“I went to the woods because I wished to live deliberately, to front only the essential facts of life, and see if I could not learn what it had to teach, and not, when I came to die, discover that I had not lived.”
—Henry David Thoreau

Abstract

For the Internet to sustain enduring growth and sustainability, shifting from IPv4 to IPv6 is imperative. However, the widespread adoption of IPv6 faces numerous challenges, including the need for infrastructure upgrades and service provider support. In this thesis, we propose an innovative approach to promoting IPv6 connectivity by leveraging IPv4 tunneling to bring IPv6 capabilities to end users without requiring a complete transition to IPv6. By allowing users to test IPv6 connectivity through tunneling, we aim to familiarize them with the benefits of IPv6 and encourage a gradual migration. This thesis presents an in-depth analysis of the advantages and challenges of this approach, along with practical implementation and evaluation.

This approach was practically implemented in collaboration with InfoCamere, the IT Society of the Italian Chambers of Commerce, and VSIX, Established by the University of Padua, which operates the Neutral Access Point of the North East, aiming to promote Internet use in Veneto (the north-eastern part of Italy) through cooperation with local, national, and international Internet Service Providers. The partnership with InfoCamere and VSIX allowed real-world deployment of the IPv4 tunneling strategy within a dynamic network environment, providing valuable insights into its feasibility and effectiveness.

The results of this collaboration showcase the potential of IPv4 tunneling as an effective strategy for promoting IPv6 adoption while minimizing disruption to existing IPv4 networks. Furthermore, this practical application emphasizes the importance of collaborative efforts in advancing internet technology for a sustainable future in addition to the technical aspects.

Contents

Listing of figures

1 Introduction

With billions of devices connected, the Internet has become an essential component of our everyday life in the rapidly changing digital age, facilitating smooth communication and information sharing. This vast digital network serves as a conduit for information exchange, collaboration, and innovation, fostering a borderless environment where geographical distances are bridged by clicking a button.

At the heart of this interconnected realm lies the intricate framework of Internet Protocol (IP), the backbone of global communication. IP protocols, such as IPv4 and IPv6, play a pivotal role in facilitating the seamless flow of data across networks, ensuring that information can traverse continents and reach every corner of the world. These protocols provide the numerical addresses that identify and distinguish devices on the Internet, acting as the linchpin for the connectivity that underpins our digital interactions.

The evolution of the Internet Protocol is intricately linked to the concept of networks, where two or more computers are interconnected to exchange data through cable or wireless connections. A network facilitates the exchange of various resources, such as data, applications, and hardware. Essential hardware components for network functionality include cables, switches, routers, and Network Interface Cards (NICs). In a network, there are server computers, which share resources like scanners and printers, and client computers, which access these resources. The connection to the network is established through NICs physically connected via Ethernet cables to switches, which, in turn, connect to the broader network.

This synthesis of hardware and protocols forms the backbone of the digital interconnectedness that underlies the Internet's evolution. The Internet's ubiquity and its critical role in shaping modern society bring forth new challenges, particularly concerning the depletion of available IPv4 addresses, raising concerns about the future growth and scalability of the Internet. IPv6 was introduced as the next-generation internet protocol to address this issue, offering significantly larger address space, enhanced security features, and improved network eficiency.

Despite the clear advantages of IPv6, its adoption could have been faster. The transition from IPv4 to IPv6 requires substantial investment in infrastructure upgrades and coordination among service providers, making it a complex and time-consuming process. Additionally, end users often need more awareness of IPv6 and the benefits it can bring.

This master's thesis proposes an approach to promoting IPv6 connectivity without requiring a complete transition from IPv4. The idea is to leverage IPv4 tunneling techniques to bring IPv6 capabilities to end users, allowing them to experience IPv6 connectivity while still operating within the IPv4 network infrastructure.

Overall, this master's thesis seeks to bridge the gap between IPv4 and IPv6 by providing a practical and incremental approach to IPv6 promotion, enabling end users to experience the benefits of IPv6 connectivity while minimizing disruption and maximizing compatibility with existing infrastructure.

1.1 Background of IPv4

The dominant version of the Internet Protocol, IPv4, emerged in the early 1980s, operating on a 32-bit address space and allowing for approximately 232, equal to 4.3 billion unique addresses [1]. However, the surge in Internet-connected devices globally led to a rapid depletion of available IPv4 addresses.

1.1.1 Address Format and Representation

As demonstrated in figure below 1.1, IPv4 addresses are 32-bit numerical labels, usually expressed in dotted-decimal format, consisting of four octets separated by dots. Each octet represents 8 bits, and the entire address is divided into network and host portions. The first part (usually the first one, two, or three octets) denotes the network, while the remaining part designates the specific host within that network. A subnet mask, also a 32-bit value, determines the division between the network and the host. This addressing scheme allows approximately 2^{32} equal to 4.3 billion unique addresses.

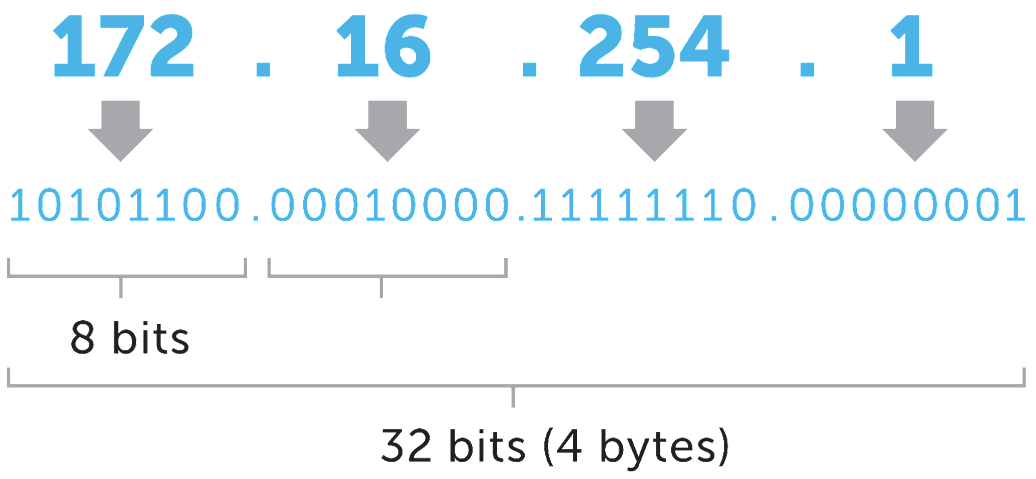


Figure 1.1: Sample IPv4 address tin dotttted-dectimal Nottatttion and IPv4 Address tin Btinary Nottatttion

The Internet Protocol is one of the major protocols in the Transmission Control Protocol/Internet Protocol (TCP/IP) model, which is the global standard for Internet communications, and it makes it possible for devices connected to the Internet to communicate with one another across the network. In the OSI model, which is also a conceptual framework that describes how different networking protocols interact within a network, the IPv4 protocol works on the Network layer. The protocol's primary function is to identify hosts based on their logical addresses to route data between them over the network. The logical address of a host in a network is the IP address, and the IPv4 addressing scheme has been used for a while now to identify hosts in a network. This system is based on a 32-bit logical address [2]. The exponential growth of the Internet, however, was not predicted. Over roughly 25 years, the Internet's user base escalated from a handful to billions of users, a growth that swiftly depleted the IPv4 address space despite a significant portion remaining unused. The inherent structure of IPv4 addresses, where only 126 networks were allocated half of the address space, played a substantial role in this depletion.

1.1.2 Classful Addressing and Drawbacks

The early internet was significantly shaped by classful addressing, a fundamental IPv4 addressing architecture that was first published in 1981 with RFC 791. A, B, C, D, and E are the five unique classes into which the IPv4 address space was split. While classes D and E were set aside for specific uses like multicasting and future use, respectively, classes A, B, and C were mostly utilized for network hosts. A default network mask was assigned to each class; Class A used an 8-bit mask, Class B a 16-bit mask, and Class C a 24-bit mask. Each class of this fixed-length subnet masking mechanism catered to networks of different sizes and offered an organized method of allocating addresses.

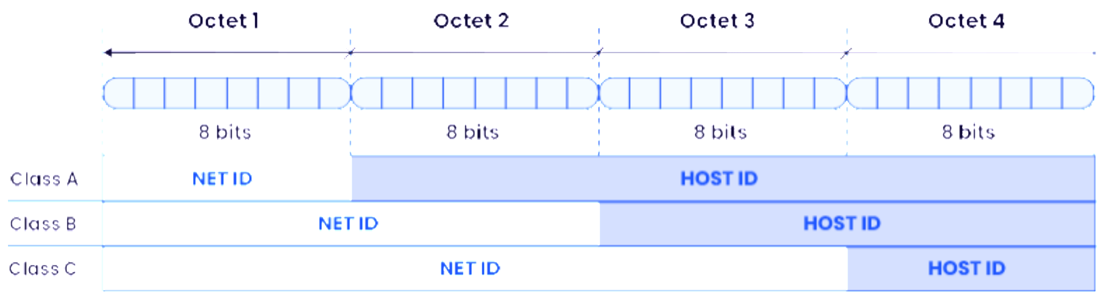


Figure 1.2: Class A allocattes tthe tintitttial 8 btitts for tthe nettwork tidentttiftier, whtile Class B and Class C asstign 16 and 24 btitts, respecttttively.

The inherent ineficiency of classful addressing in address allocation was one of its main drawbacks, especially as the internet grew quickly. Due to the fixed-length subnet masks, networks were allocated in blocks that frequently surpassed the real needs of the businesses, resulting in a large amount of address waste. For instance, a Class B network with over 65,000 usable addresses would be issued to a tiny firm that only needed a few hundred IP addresses, leaving large areas of unutilized addresses. As the demand for IP addresses increased, this ineficiency became more and more of a concern, and by April 2017, it was one of the main causes of the IPv4 address shortage. Despite its limitations, classful addressing was a pivotal phase in the evolution of internet protocols, establishing the foundation for later developments like Classless Inter-Domain Routing (CIDR).

1.1.3 Transition from Classful Addressing to CIDR

Recognizing the looming issue, the Internet Engineering Task Force (IETF) began to address the problem in 1991 by looking ahead to the future of Internet architecture. The emergence of CIDR aimed to enhance eficiency and alleviate the burden on routers, though predictions still indicated exhaustion between 2005 and 2011 [3].

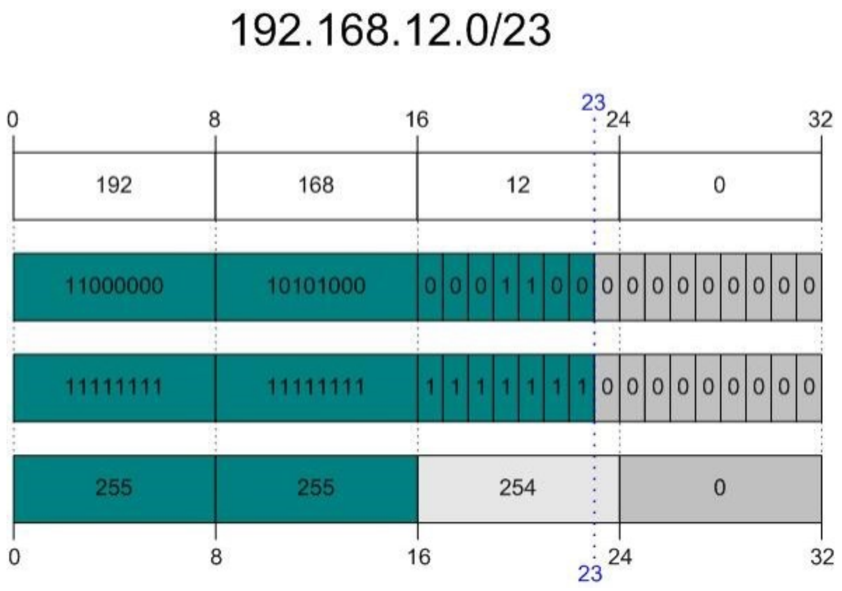


Figure 1.3: CIDR example: usting 23 of tthe 32 btitts for tthe nettwork ID. Thtis leaves you wtitth 9 btitts for tthe hostt ID. So, tthere are 192.168.12.0 - 192.168.13.255 as avatilable addresses tin tthe subnett.

CIDR is a major development in IP addressing. With the help of CIDR, several Class C networks can be combined to form bigger supernet blocks, like a /23 or /22. CIDR introduces a dynamic allocation mechanism based on specified criteria, in contrast to classful addressing, which assigns addresses strictly based on predefined classes. IP address blocks are dynamically assigned in a classless addressing system, allowing for more flexible and effective address allocation depending on the real needs of networks. By offering a more scalable and flexible method of managing IP addresses, this dynamic allocation mechanism overcomes the drawbacks of classful addressing and is crucial for meeting the internet's always-growing needs.

1.1.4 Introduction of IPv6 and Addressing Future Challenges

In this context, the introduction of IPv6 becomes particularly relevant. IPv6, with its 128-bit address space, was conceived to address the limitations of IPv4. The enormous address space, allowing for over 340 undecillion unique addresses, was designed to accommodate the increasing number of devices and users in the digital age. IPv6's adoption aims to prevent the depletion issues experienced by IPv4. However, despite the clear advantages of IPv6, its widespread implementation has been gradual due to the complexities of transitioning from the established IPv4 infrastructure.

In summary, the evolution of IPv4's address space scarcity and the introduction of IPv6's vast address range reflects the dynamic landscape of internet technology, addressing the ever-growing need for unique identifiers in the digital realm.

1.2 Background of IPv6

The limitations of IPv4's address space led to the introduction of IPv6, the next iteration of the Internet Protocol, to solve these issues.

1.2.1 IPv6 Address Structure

As seen in Figure 1.4, IPv6 addresses are stated in hexadecimal format and comprise eight sets of four hexadecimal digits separated by colons. There are a total of 128 bits because each group represents 16 bits. The Internet can accommodate almost an infinite number of unique addresses because to this expanded address space, which guarantees its continuous expansion [4].

Compared to the decimal format used for IPv4, the hexadecimal format for IPv6 addresses has a number of advantages. Large numbers can be represented more succinctly using hexadecimal notation, which makes IPv6 addresses simpler to understand and work with. Hexadecimal notation is also shorter than decimal notation, which shortens IPv6 addresses and makes address administration and allocation easier. Each set of four hexadecimal digits in an IPv6 address corresponds to a 16-bit address segment. Colon marks, which act as group delimiters, are used to divide these sections apart. Colonies are a useful delimiter because they improve readability and simplify IPv6 address parsing.

Moreover, IPv6 addresses allow successive groups of zeros within an address to be represented by double colons (::). Address notation can be made simpler and address lengths can be further reduced thanks to this capability, which is called zero compression. When there are significant blocks of zeros in an address, as in link-local or site-local addresses, zero compression is especially helpful. Overall, the IPv6 addressing scheme's scalability and eficiency are facilitated by the concise notation and hierarchical structure of IPv6 addresses, which allow the Internet to handle the increasing number of connected devices and services.

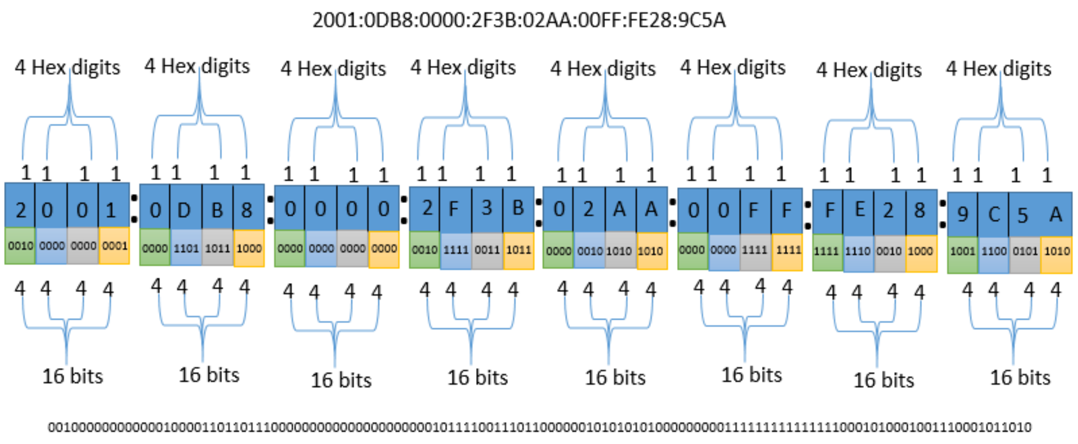


Figure 1.4: IPv6 Address tin Hexadectimal Nottatttion and IPv6 Address tin Btinary Nottatttion

1.2.2 Challenges of NAT and Loss of Transparency

One of the most contentious adjustments made to IPv4 involves using private network address space and network address translators (NATs). This alteration disrupts the core principle of IP computing known as end-to-endness, where communication between source and destination nodes occurs without intermediary interference. Although many applications can find workarounds, complications arise in terms of security. The Internet Security Protocol (IPsec) relies on the uniqueness of nodes' IP addresses to prevent packet spoofing. Additionally, not all applications can effectively navigate NATs. IPv6 aims to alleviate these challenges by providing a wealth of new addresses, reducing the need for new NATs, and ensuring end-to-end interoperability.

One of the key advantages of IPv6 lies in its ability to eliminate the need for Network Address Translation (NAT), a common workaround used in IPv4 to mitigate address scarcity. NAT introduces complexities and limitations, hindering the seamless end-to-end connectivity envisioned for the Internet. With IPv6, each device can have a globally unique and routable IP address, facilitating direct communication and reducing the reliance on address translation mechanisms.

End-to-endness, or transparency, characterizes a network environment where endpoints can collaborate without accounting for the network's middle layer or facing interference from intermediate systems. This transparency streamlines network application development, requiring developers to design applications to interface with the network cloud alone. The absence of openness entails grappling with intermediate components such as firewalls, NATs, and caching proxies, with security susceptible to this factor.

1.2.3 Impact of NAT on End-to-Endness

The function of NATs impedes end-to-endness as they modify inbound and outbound packet headers. When headers are altered, security protocols at the network layer lose their eficacy. The distinction between a NAT changing packet header on a secured packet and a malicious actor manipulating headers to execute packet spoofing becomes blurred. Both instances appear as potential threats. The rise of the NAT-dependent Internet has led to a slowdown in applications reliant on transparency.

NATs modify packet headers because the original headers within the NAT-ed network employ non-unique private (NET-10) addresses across the global Internet. Using standard unique addresses ensures an unmistakable approach to addressing every network node, irrespective of location. In cases of ambiguous node numbering, NATs and other intermediaries must intervene to prevent confusion and enable communication among nodes bearing identical addresses—transparency guarantees direct communication between nodes unaffected by events within the network cloud. In the event of a NAT or intermediate system failure, the mediated sessions also collapse, lacking a straightforward way for communicating hosts to circumvent the issue.

The original Internet was designed with end-to-end interoperability as a fundamental feature. However, as IPv4 addresses dwindle and more networks resort to NATs, proxies, and gateway devices, the Internet's nature is evolving. Rather than maintaining its robustness to circumvent failures, the modern Internet is becoming less dependable and experiencing performance degradation. Instead of retaining scalability to accommodate new applications without extensive network upgrades, the current Internet confines applications to relying on Web services. Simultaneously, changes in network infrastructure necessitate adjustments to node software and configurations.

RFC 3424, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," delves into the challenges associated with network trafic traversing multiple NAT domains and their implications for network transparency.

As we delve further, IPv6 eliminates NATs, introducing link-local and site-local network address concepts. These local addresses can be likened to telephone extensions, where external callers might dial a country code, area code, exchange, and extension. In contrast, internal callers from a specific ofice (link-local) or organization (site-local) can reach the destination by dialing an extension.

It is worth noting that the loss of transparency is merely a symptom of a broader issue. Had the Internet not expanded rapidly and dramatically, it would operate as a transparent network with end-to-end interoperability as the norm. The menace to transparency arises from the same growth that has caused the IPv4 address shortage and the escalating reliance on NATs for conservation purposes.

1.2.4 Address Space Management

Beyond the address space, IPv6 incorporates several other improvements over its predecessor. It offers built-in support for features like auto-configuration, simplifying the process of connecting devices to the network. IPv6 also introduces a more eficient and streamlined packet header format, reducing the processing overhead on networking devices and enhancing network performance. Additionally, IPv6 enhances the security of IP communications by integrating IPsec (IP security), a mandatory part of the protocol, providing encryption, authentication, and integrity features at the network layer.

In summary, IPv6 represents a transformative step forward, addressing the limitations of IPv4 and providing a foundation for a more scalable, eficient, and secure internet. The following sections of this master's thesis will delve into specific aspects of IPv6 adoption and propose strategies to promote its connectivity alongside existing IPv4 infrastructure [5].

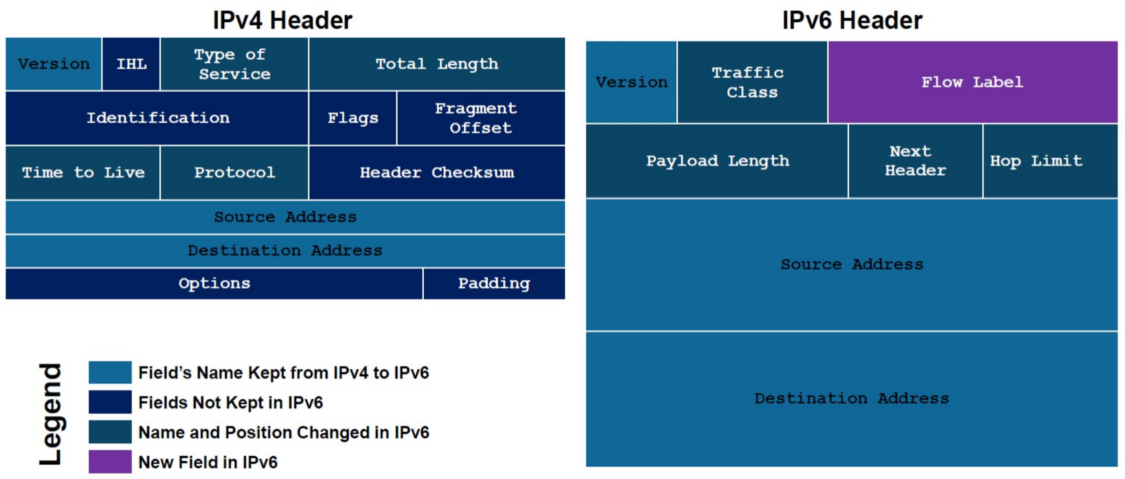


Figure 1.5: Compartison of IPv4 and IPv6 Headers: A vtisual representtatttion htighltighttting crtitttical dtistttinctttions tin tthe header sttructttures of IPv4 (leftt) and IPv6 (rtightt) prottocols

1.3 Motivation and Problem Statement

One of the primary motivations behind the development of IPv6 is the exhaustion of IPv4 addresses. IPv4, with its 32-bit address space, can only accommodate approximately 4.3 billion unique addresses allocated to network devices such as computers, smartphones, servers, and IoT devices. With the explosive growth of Internet-connected devices worldwide, the available pool of IPv4 addresses has been rapidly depleting.

IPv6 introduces a significant expansion of the address space by utilizing a 128-bit format, providing an astronomical number of unique addresses—roughly 340 undecillion addresses, about 3.4 billion. This vast address space ensures every device can have an individual and globally routable IP address, eliminating the need for address conservation techniques like Network Address Translation (NAT) used in IPv4.

Beyond the address space, IPv6 incorporates several other improvements over its predecessor. It offers built-in support for features like auto-configuration, simplifying the process of connecting devices to the network. IPv6 also introduces a more eficient and streamlined packet header format, reducing the processing overhead on networking devices and enhancing network performance. Additionally, IPv6 enhances the security of IP communications by integrating IPsec (IP security) as a mandatory part of the protocol, providing encryption, authentication, and integrity features at the network layer.

While IPv6 brings numerous advantages and is designed to replace IPv4, its deployment has been gradual due to various challenges. These challenges include the need for infrastructure upgrades, interoperability with existing IPv4 networks, and the complexity of transitioning from IPv4 to IPv6 without disrupting existing services. However, the continued growth of Internet-connected devices and the increasing awareness of IPv4 address exhaustion have spurred greater adoption and deployment efforts in recent years.

Encouraging IPv6 adoption is essential to maintaining the Internet's scalability and future expansion. It requires a concerted effort from network operators, service providers, software developers, and policymakers to facilitate a smooth transition from IPv4 to IPv6. By embracing IPv6, the Internet can support an ever-expanding array of devices and services, enabling innovation, connectivity, and a sustainable digital future [6].

1.4 Objective of Study

In this master's thesis, we will delve into the details of IPv4 tunneling and its potential for promoting IPv6 adoption. We will explore the Teredo tunneling protocol, analyzing its advantages, limitations, and compatibility with existing network architectures. Furthermore, we will investigate the performance, reliability, and user experience aspects of IPv6 connectivity over IPv4 tunnels.

By evaluating the proposed approach through practical implementation and experimentation, we aim to provide insights into the viability and effectiveness of using IPv4 tunneling as a strategy for IPv6 promotion. Our findings will contribute to the existing knowledge on IPv6 adoption and offer recommendations to overcome potential challenges and improve the overall experience.

2 Literature Review

In this section, we provide a literature review and background knowledge to understand the context and the main concepts of this master's thesis. This information serves as a foundation for the subsequent chapters and helps the reader to follow the arguments, methods, and results presented in this work.

2.1 Evolution of Internet Protocol (IPv4 to IPv6)

The evolution of the IP is intricately tied to the fundamental nature of networking technologies, shaping the digital era's interconnected landscape. At the core of this evolution is the concept of IP addresses, unique identifiers assigned to every computer on the Internet. The IP defines the format of data packets and establishes an addressing system with dual functions: identifying hosts and providing a logical location service [7].

To overcome the limitations of IPv4 and introduce advancements in Quality of Service for streaming services, IPv5 was conceptualized. Despite being envisioned as a connection-oriented complement to IPv4, IPv5 was never introduced for public use, paving the way for developing the next generation—Internet Protocol Version 6 (IPv6). [8].

IPv6, also known as IP next generation (IPng), represents a monumental leap in addressing the challenges posed by IPv4. With its 128-bit address format, IPv6 offers an astronomical number of unique addresses. This expansive address space not only resolves the issue of address exhaustion but also accommodates the diverse array of devices constituting the contemporary Internet landscape.

The journey from IPv4 to IPv6 is not merely an expansion of address space; it is a comprehensive response to the intricate demands of the digital age, encompassing connectivity, performance, and security. The ongoing evolution continues to shape the dynamic landscape of technology, ensuring the Internet's resilience and adaptability in the face of ever-expanding digital frontiers.

2.2 IPv6 Adoption Challenges and Solutions

As a new protocol, IPv6 introduces complexities requiring careful consideration, particularly in security, monitoring, compatibility, interoperability, and stability. These challenges underscore the need for comprehensive solutions addressing the diverse issues organizations and network professionals face as they transition to IPv6. This section aims to provide valuable insights and guidance for stakeholders involved in IPv6 adoption efforts by delving into each challenge and proposing practical solutions.

2.2.1 Security

The emergence of IPv6 necessitates a nuanced approach to security and privacy within computer networking, demanding the attention of seasoned professionals. Unlike its predecessor, IPv4, IPv6 introduces an augmented attack surface owing to its expansive address space and advanced features. The security landscape is further transformed by implementing distinct mechanisms, such as Internet Protocol Security (IPsec) and Stateless Address Auto-Configuration (SLAAC), which mandate meticulous configuration and adept management. IPv6 introduces privacy concerns, with certain IPv6 addresses potentially exposing user identities and locations. Consequently, proficiency in IPv6 becomes imperative for networking professionals, demanding a comprehensive understanding of potential threats and vulnerabilities.

A major security challenge in IPv6 adoption stems from the vulnerabilities linked to IPv6 extension headers. Despite IPv6's advancements, these extensions introduce security risks, notably in terms of Denial of Service (DoS) attacks. Extension headers carry supplementary data for network device processing of IPv6 packets, yet their mishandling presents a significant threat. Particularly concerning are DoS attacks facilitated by these headers, an issue that remains largely unresolved according to RFC 8200.

The security vulnerabilities associated with IPv6 extension headers include covert channel threats in Hop-by-Hop options header and destination options headers, fragmentation attacks, and threats related to router header source routing and router alert. These vulnerabilities have the potential to evade security controls, impose processing requirements leading to DoS, and result in DoS due to implementation errors.

Moreover, the negative performance impact of IPv6 extension headers on handling devices, such as routers, firewalls, and Network Intrusion Detection Systems (NIDS), raises concerns. Without proper rules or controls, attackers can exploit this weakness by inundating the network with a large volume of IPv6 trafic using extension headers, thereby executing a DoS attack. The repercussions extend to affecting the performance of critical network infrastructure.

Despite efforts to address these vulnerabilities, recent research has demonstrated that even well-known security devices lack inherent capabilities to fully thwart DoS attacks exploiting extension header vulnerabilities. The study raises questions about the eficacy of current intermediary devices, emphasizing that they may still be susceptible to exploitation by attackers.

In summary, the challenge lies in understanding and mitigating the risks associated with IPv6 extension headers, particularly in the context of DoS attacks. As organizations transition to IPv6, a comprehensive approach to securing extension headers is imperative to ensure the resilience of network infrastructure.

Solutions for Security Challenges of IPv6 Adoption:

Awareness and Training: The study underscores the importance of proper knowledge and training for network administrators before transitioning to IPv6. Acknowledging that security challenges are not vendor-specific but rooted in protocol design, the conclusion advocates equipping administrators with the expertise needed to understand and address IPv6-related security issues effectively.

Protocol Design Improvement: Recognizing security challenges as a protocol design issue, addressing the vulnerabilities associated with IPv6 extension headers requires improvements in protocol design. This implies collaborative efforts by the Internet Engineering Task Force (IETF) and other relevant bodies to enhance the security features of IPv6.

Defendable Architecture: The ultimate goal of finding a solution to the security challenges of IPv6 adoption is to propose a defendable architecture. This architecture should work within the existing infrastructure and precisely target and mitigate the identified attack vectors related to IPv6 extension headers. The emphasis is on developing a robust framework that can withstand evolving security threats.

Transition Planning: Given that transitioning to IPv6 is a complex process, there is a necessary need for a well-thought-out transition plan. This involves considering security implications and preparing for potential threats, especially in the context of newly introduced features like extension headers.

In summary, the proposed solutions involve a combination of education, collaborative efforts for protocol design improvements, and developing a defendable architecture tailored to IPv6 security challenges. By addressing these aspects, the conclusion aims to pave the way for a secure and resilient IPv6 adoption, ensuring the effective coexistence of the new protocol in the ever-evolving landscape of network security [9].

2.2.2 Monitoring

The deployment of IPv6 alongside IPv4 introduces several challenges in network monitoring. One significant challenge arises from the auto-configuration of IPv6 addresses, which complicates the identification of hosts within a Local Area Network (LAN) due to multiple temporary IPv6 addresses. This lack of clear host identification exposes network users and organizations to vulnerabilities, as users may unknowingly violate security policies when bypassing standard IPv4 firewall rules.

Traditional monitoring approaches face limitations when applied to IPv6 trafic. Issues such as temporary addresses, different encapsulation types of IPv6 over IPv4, and non-unique mappings between data link addresses and IP addresses make it challenging to employ conventional monitoring techniques effectively. Furthermore, tunneling IPv6 over IPv4 introduces complexities, allowing packets to bypass firewall rules and demanding specialized monitoring to detect potential sources of uncontrolled user trafic.

Stateful and stateless IPv6 configurations pose unique challenges for user identification. Stateless configurations, particularly, create non-unique IPv6 addresses, making it dificult to establish a standardized identification system. Even with stateful configurations, limitations in DHCPv6 (Dynamic Host Configuration Protocol version 6), such as the inability to obtain a default gateway, add to the complexities of uniquely identifying users.

Practical solutions are imperative to address the challenges associated with IPv6 monitoring. An innovative approach involves developing new monitoring techniques capable of handling

IPv6 trafic nuances. This includes devising methods to overcome the limitations of temporary addresses, encapsulation, and non-unique mappings. Additionally, specific solutions are required for monitoring tunneled IPv6 trafic during the transition period, ensuring that potential security threats are identified.

In response to the challenges of IPv6 address configurations, a proposed solution revolves around the creation of a comprehensive data structure for unique host identification. This involves collecting various pieces of information, such as IPv6 addresses, MAC addresses, login names, timestamps, and switch ports. By combining this information into a structured format, the system aims to facilitate effective user identification in IPv6 networks, overcoming the limitations associated with stateful and stateless configurations.

Regarding monitoring data collection, there is a solution that emphasizes using the Simple Network Management Protocol (SNMP) to gather information from switches, routers, and other network devices. This data, including IPv6-to-MAC address mappings and trafic statistics obtained through Netflow, are then stored in a central database. By integrating this information with additional data from Router Advertisement messages, Netflow records are extended to include a unique identifier, allowing for the identification and tracking of user activities, even with temporary IPv6 addresses.

While these solutions address current challenges in IPv6 monitoring, there remains a need for further research to enhance the reliability of data transmission during network attacks and to optimize monitoring systems for scalability in larger networks. The proposed solutions aim to bridge gaps in practical IPv6 monitoring and contribute to the development of robust and scalable monitoring architectures for both IPv4 and IPv6 environments [10].

2.2.3 Compatibility, interoperability, and stability

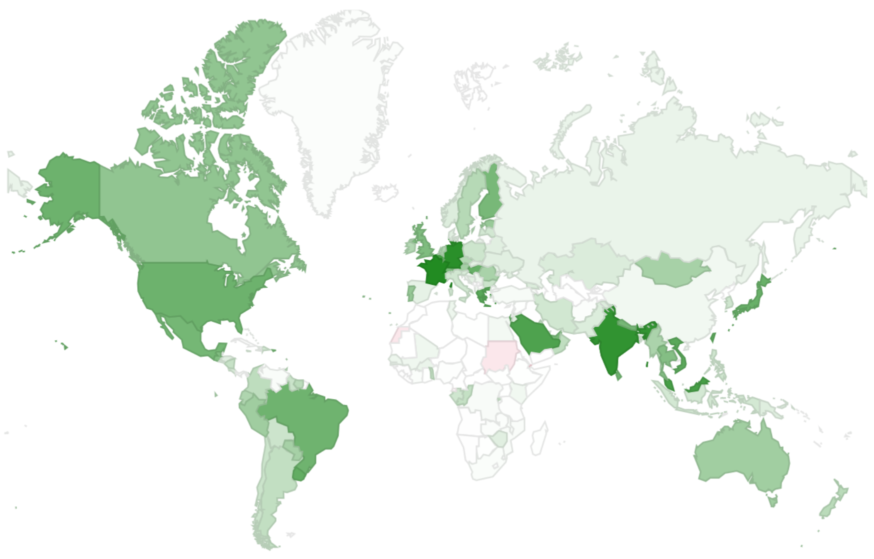
One of the challenges in the transition to IPv6 is the lack of backward compatibility between the two versions. This incompatibility means that devices and networks using different versions cannot communicate directly, necessitating transition mechanisms such as dual-stack, tunneling, and translation to enable coexistence and interoperability [11]. However, these mechanisms can introduce complexity, overhead, and security risks, highlighting the need for IPv6 professionals to be well-versed in the different options and their trade-offs and to be capable of implementing and troubleshooting them effectively [12].

The incompatibility between IPv4 and IPv6 has been recognized as a significant obstacle to the smooth adoption of IPv6, leading to the development of numerous transition technologies by the IETF to facilitate the gradual and seamless transition to IPv6 [12]. Additionally, the lack of backward compatibility has been identified as a major challenge, necessitating the coexistence of IPv4 and IPv6 in various forms during the transition period [13]. Furthermore, the incompatibility between the two protocols has been acknowledged as a long process, requiring the use of transition mechanisms to bridge the gap and ensure interoperability [14].

Challenges of Compatibility, Interoperability, and Stability Aspect of IPv6 Adoption

IPv4 Address Exhaustion and Migration Complexity: The advent of the digital age has led to an unprecedented demand for Internet Protocol addresses, resulting in the imminent exhaustion of IPv4 addresses. The transition to IPv6, designed to address this scarcity, poses a formidable challenge. Legacy systems and infrastructure reliant on IPv4 face complexity in migrating to the new protocol. Coordinating a smooth transition while maintaining uninterrupted network operations become a significant hurdle [15].

Slow Adoption and Knowledge Gap: IPv6 has not been widely adopted and deployed yet, despite being available for more than two decades. According to Google, only about 45% of the global Internet users access its services over IPv6 as of Jan 2024. Many factors affect the adoption and deployment of IPv6, such as cost, inertia, lack of awareness, and regulatory barriers. Therefore, IPv6 professionals need to advocate for the benefits and urgency of IPv6, and be able to plan and execute migration strategies that minimize disruption and maximize eficiency.

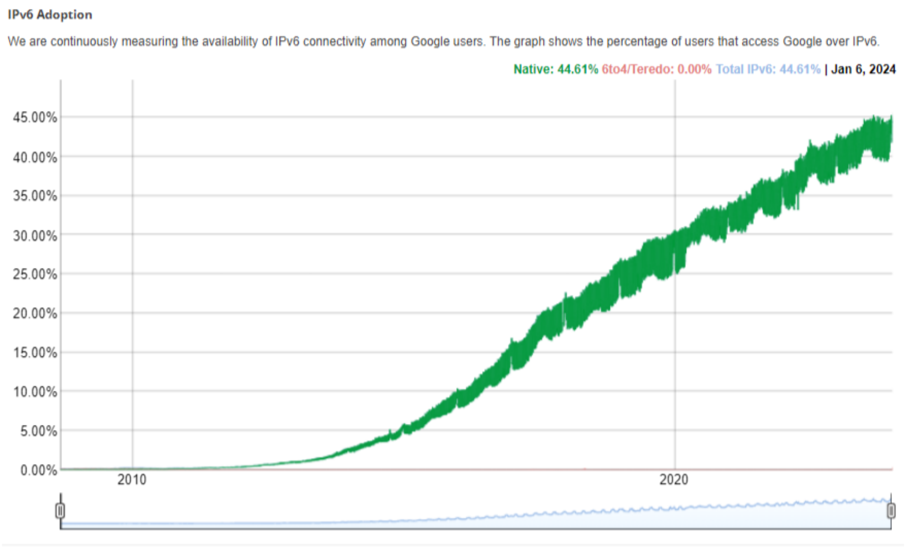


🟩 Regions where IPv6 is more widely deployed (the darker the green, the greater the deployment) and users experience infrequent issues connecting to IPv6-enabled websites.

🟧 Regions where IPv6 is more widely deployed but users still experience significant reliability or latency issues connecting to IPv6-enabled websites.

🟥 Regions where IPv6 is not widely deployed and users experience significant reliability or latency issues connecting to IPv6-enabled websites.

Figure 2.1: Google collectts sttatttistttics aboutt IPv6 adopttion per counttry tin tthe Intternett tin tthe world



IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

Native: 44.61% 6to4/Teredo: 0.00% Total IPv6: 44.61% | Jan 6, 2024

Figure 2.2: Google collectts sttatttistttics aboutt IPv6 adopttion on tthe Intternett on an ongoting bastis.

Performance Concerns and Header Size IPv6 introduces improvements in mobility support and a vastly expanded address space. However, the larger header size inherent in IPv6 raises concerns about potential impacts on network performance. The increased size of IPv6 headers, compared to the more concise IPv4 headers, necessitates thorough research and optimization to ensure eficient data transmission without compromising network speed.

Therefore, despite the clear advantages of IPv6, its adoption has been slow, creating a digital divide between IPv4 and IPv6 networks. One of the primary challenges lies in the lack of awareness and understanding among organizations, businesses, and individuals regarding the benefits and necessities of IPv6. Bridging this knowledge gap becomes imperative for fostering widespread adoption.

Solutions for Compatibility Challenges of IPv6 Adoption:

Strategic IPv6 Transition Planning To address the challenge of IPv4 address exhaustion and migration complexity, organizations need to develop strategic transition plans. Implementing dual-stack configurations, where both IPv4 and IPv6 coexist, allows for a gradual migration without disrupting existing services. Robust planning and execution are essential to ensure a seamless transition without compromising network stability [16].

Educational Initiatives and Awareness Campaigns Overcoming the slow adoption of IPv6 requires concerted efforts in education and awareness. Initiatives that educate IT professionals, network administrators, and the general public about the advantages of IPv6 can significantly contribute to its widespread adoption. Awareness campaigns should emphasize the urgency of migration and dispel myths or misconceptions surrounding IPv6.

Performance Optimization and Research Addressing concerns related to IPv6 header size and performance necessitates ongoing research and optimization efforts. Network administrators and researchers should collaborate to develop and implement eficient algorithms and protocols that minimize the impact of large headers. Optimization strategies can ensure that IPv6 networks operate at optimal speeds comparable to IPv4.

In conclusion, the challenges associated with IPv6 compatibility, interoperability, and stability demand a multifaceted approach. Strategic planning, educational initiatives, and continuous research are key components of effective solutions to propel the seamless integration of IPv6 into the fabric of the global network infrastructure [17].

2.3 The Role of Transition Techniques in IPv6 Implementations

In the landscape of networking, the coexistence of IPv4 and IPv6 has become a prominent feature, necessitating the development of transition techniques to facilitate seamless communication between these two protocols. Among these techniques, tunneling emerges as a pivotal approach, offering a method to encapsulate IPv6 packets within IPv4 headers, thus enabling their traversal across IPv4 networks. This section delves into the intricacies of ISATAP Tunneling, 6to4 tunneling, and Teredo Tunneling.

2.3.1 ISATAP Tunneling

ISATAP, also known as the Intra-Site Automatic Tunnel Addressing Protocol, serves as a pivotal facilitator for communication between IPv4 hosts and ISATAP routers through the utilization of tunneling mechanisms. This process fundamentally involves encapsulating IPv6 packets within IPv4 packets to navigate IPv4 networks seamlessly. Imagine a scenario where an IPv4-only computer endeavors to communicate with a computer residing on an IPv6-only network. Initially, the IPv4 computer generates an IPv6 packet. However, since direct traversal across the IPv6 network is not feasible, the IPv6 packet undergoes encapsulation within an IPv4 packet [18].

Subsequently, this encapsulated packet embarks on a journey through an ISATAP tunnel to reach the designated ISATAP router. While this example depicts a straightforward scenario, it's worth noting that the packet may traverse multiple IPv4 routers en route to its destination. Upon reaching the destination, the IPv4 packet is discarded, leaving behind the encapsulated IPv6 packet, which is then forwarded to its intended destination within the IPv6 network. This elementary example sheds light on the fundamental process underpinning ISATAP tunneling, laying the groundwork for a more comprehensive exploration of its intricacies [19].

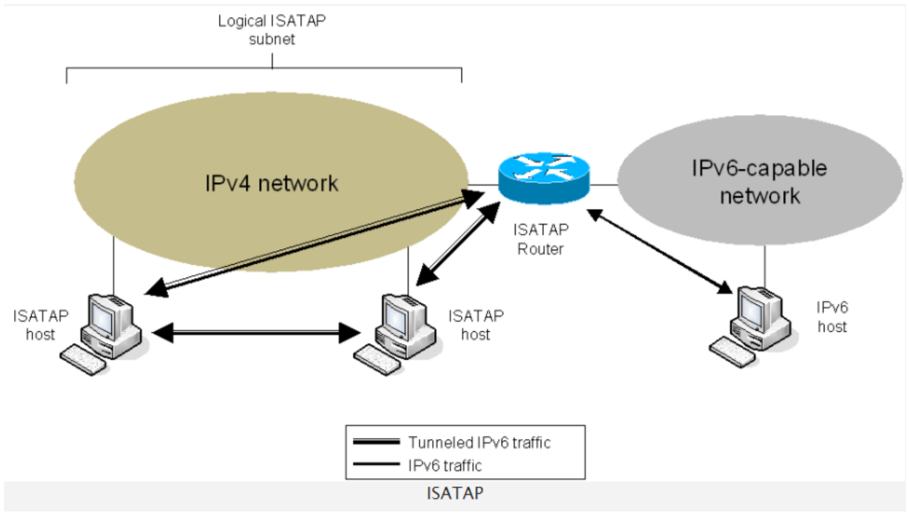


Figure 2.3: ISATAP Tunnelting schema

Delving deeper into the inner workings of ISATAP, let's consider a scenario where a computer is tethered to an IPv4 network and aims to communicate with a computer situated on an IPv6-only network. Initially, the IPv4 computer solicits the ISATAP server to procure an IPv6 address. This IPv6 address comprises a network prefix, typically configured by the administrator or provided by the ISATAP server, along with an interface ID derived from the IPv4 address of the computer.

Despite acquiring an IPv6 address, the computer remains entrenched within an IPv4 network. At this juncture, the transition protocol assumes significance. The computer endeavors to locate an ISATAP router, a process that can be achieved through manual configuration or by querying the DNS server. Subsequently, the computer formulates an IPv6 packet, with its allocated ISATAP address serving as the source address and the destination address being that of the IPv6 computer. Given the computer's residence within an IPv4-only network, the IPv6 packet is encased within an IPv4 packet. Upon transmission, the IPv4 packet traverses the network until it reaches the ISATAP router. Here, the IPv4 encapsulation is stripped away, and the embedded IPv6 packet is forwarded to the destination as a conventional IPv6 packet. This reciprocal process underscores the pivotal role of the ISATAP router as a bridge between IPv4-only and IPv6-only networks, albeit within internal, non-publicly routable network environments [20].

2.3.2 6to4 tunelling

6to4 tunneling, also known as automatic 6to4 tunneling, stands out as a versatile solution for bridging the gap between IPv4 and IPv6 networks. Unlike manual tunneling, which requires meticulous configuration, 6to4 tunneling offers scalability and simplicity, making it a preferred choice for transitioning to IPv6 in various network environments. At the heart of 6to4 tunneling lies the encapsulation of IPv6 packets within IPv4 headers, allowing for seamless communication across IPv4-only networks.

6to4 tunneling is intended to make IPv6 connection between IPv6 sites and hosts possible across the current IPv4 Internet infrastructure. Through the use of this technique, IPv6 packets are encapsulated within IPv4 packets and may easily navigate IPv4 networks. It uses the IPv4 header Protocol Number 41 (Protocol-41) to identify IPv6 communication from ordinary IPv4 trafic. 6to4 makes it possible to communicate between IPv6 networks via the IPv4 Internet by encapsulating IPv6 packets within IPv4.

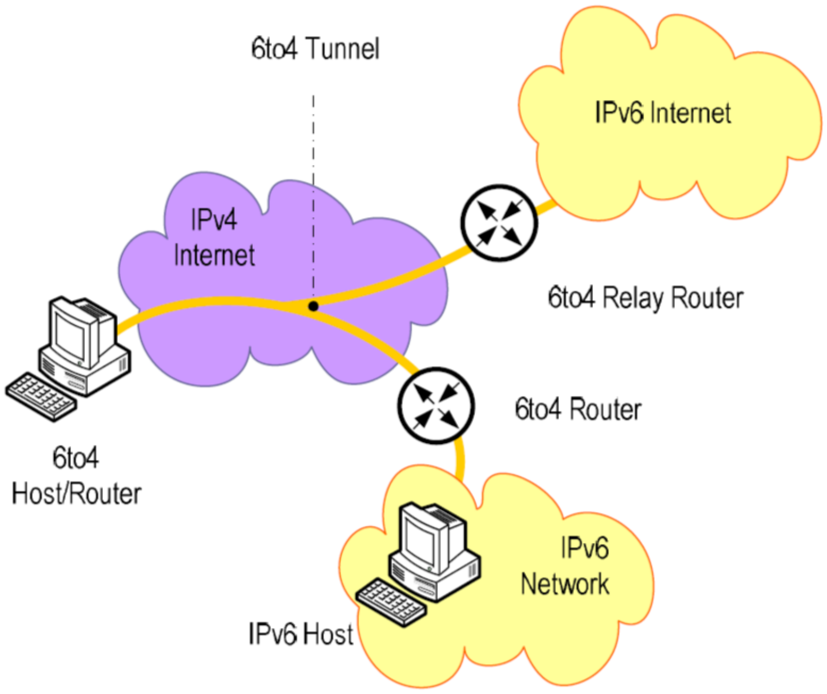


Figure 2.4: (a) Entttirely IPv4 (b) Entttirely IPv6 (c) IPv6 wtitth 6tto4 ttunnelting

The ability of 6to4 tunneling to deliver unicast IPv6 connectivity over IPv4 networks without requiring manual tunnel endpoint setting is one of its primary benefits. This is accomplished by automatically deriving IPv6 addresses using the unique IPv6 prefix ”2002::/16” and the IPv4 address of the tunnel endpoint. Furthermore, 6to4 views the IPv4 Internet as a single virtual link, enabling connectivity across various networks over IPv4 Internet connections. As a result, it may be scaled to establish IPv6 connectivity over a variety of network topologies [21].

In conclusion, 6to4 tunneling emerges as a critical component in the transition towards IPv6 adoption, offering a seamless bridge between IPv4 and IPv6 networks. Its automatic connectivity features, combined with scalability and versatility, make it a valuable tool for network administrators seeking to embrace IPv6 while maintaining compatibility with existing IPv4 infrastructure. As organizations continue to navigate the complexities of IPv6 migration, 6to4 tunneling remains a reliable transition mechanism, paving the way for a future where IPv4 and IPv6 networks coexist harmoniously.

2.3.3 Teredo Tunneling

The Teredo protocol plays a crucial role as an IPv6 transition mechanism, particularly in scenarios where other mechanisms like ISATAP and 6to4 may encounter challenges. Unlike its counterparts, Teredo is designed as a last resort transition strategy due to its unique approach. Instead of directly encapsulating IPv6 packets within IPv4, Teredo employs a method involving UDP encapsulation within IPv4 packets. While this introduces additional overhead, it offers the advantage of compatibility with Network Address Translation (NAT), making it more likely to function in diverse network environments.

Teredo is tailored explicitly for internet use and operates effectively even when hosts have private IPv4 addresses behind a NAT. The protocol tackles the challenge of assigning global IPv6 addresses to hosts with private IPv4 addresses, a task that 6to4 accomplishes by leveraging public IPv4 addresses.

In the Teredo framework, a Teredo Server is pivotal in assigning global unicast IPv6 addresses to clients. These addresses share a common network prefix, typically 2001:0:/, with the first 32 bits derived from the fixed sequence `2001:0000` and the next 32 bits representing the IPv4 address of the Teredo Server in hexadecimal. This approach ensures a consistent network prefix among Teredo clients connected to the same server.

Teredo clients require a Teredo Relay to communicate with the IPv6 Internet. The relay, often set up by an ISP or organization, advertises its capability to route to the entire Teredo network prefix. Teredo clients send encapsulated IPv6 packets to the Teredo Relay, which then forwards them as pure IPv6 packets to the IPv6 Internet, with the source address set to the client's global unicast Teredo address.

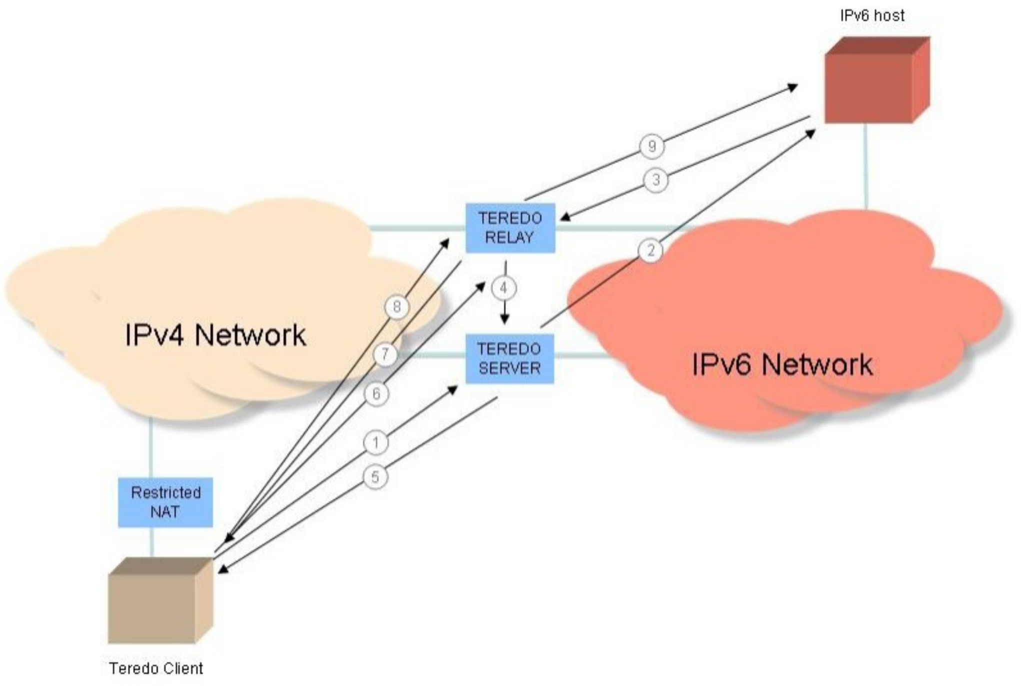


Figure 2.5: Teredo schema sttep by sttep for a resttrtictted NAT

As demonstrated in Fig.2.5, the process initiates with A Teredo 'Internet Control Message Protocol version 6 (ICMPv6) Request is sent from the Teredo client to the Teredo server to start the procedure. The client's intention to create an IPv6 connection is indicated by this request. The Teredo server receives the request, processes it, and gets ready to help the client and IPv6 hosts communicate.

The Teredo server routes the ICMPv6 Request to the appropriate IPv6 host after receiving the original request. By doing this step, the host may be certain that they are aware of the client's request and can react appropriately. The IPv6 host confirms that it is prepared to interact with the client by creating an ICMPv6 Replay in response to the forwarded request. Next, the host sends this ICMPv6 Replay to the Teredo relay, completing a crucial stage in the tunneling connection's setup.

The ICMPv6 Replay is encapsulated within a Teredo bubble by the Teredo relay and sent back to the Teredo server as the process proceeds. The ICMPv6 Replay uses this Teredo bubble as a carrier to go over IPv4 networks and get to the server. The Teredo server then sends the ICMPv6 Replay back to the client by enclosing it in another Teredo bubble. By taking this step, you can make sure that the client gets an email from the server confirming that the tunnel connection was successfully established.

A fascinating aspect of Teredo is the ability of IPv6 hosts with both IPv6 and IPv4 addresses to communicate directly with Teredo clients, bypassing the Teredo Relay. This functionality, known as Teredo Host Specific Relay, allows hosts to send IPv4 packets containing IPv6 packets directly to Teredo clients. While Teredo demonstrates its effectiveness in providing IPv6 connectivity for hosts behind NATs, the protocol also introduces challenges. The selection of Teredo-Relays, essential for communication with remote IPv6 hosts, involves a process where Teredo clients interact with Teredo servers to establish connections with specific relays [22].

In conclusion, the Teredo protocol serves as a versatile and valuable tool for IPv6 transition, offering a unique approach to addressing the complexities of network scenarios involving NATs. Its role extends beyond providing connectivity, encompassing the intricate process of selecting relays and facilitating direct communication between IPv6 hosts and Teredo clients. As a standardized protocol developed by Microsoft and outlined in RFC 4380, Teredo plays a significant role in the ever-evolving landscape of IPv6 implementations [23].

Furthermore, Teredo's practical applications extend to supporting roaming and handoff of User Equipment (UE) in wireless communications, demonstrating its adaptability in diverse networking environments [24]. However, performance evaluations and comparisons with other transition mechanisms, such as ISATAP, have been conducted, focusing on parameters like throughput, end-to-end delay, round-trip time, and jitter [25]. These evaluations contribute to a comprehensive understanding of Teredo's strengths and limitations in real-world scenarios.

2.4 Case Studies on IPv6 Implementation

This section offers actual case studies of businesses implementing IPv6. These case studies provide valuable examples of IPv6 adoption initiatives and serve as a roadmap for upcoming deployment projects by providing insights into the tactics, obstacles, and achievements faced.

2.4.1 Deploying IPv6 in the Google Enterprise Network

Google has been a pioneer in adopting IPv6 across its services. The company started enabling IPv6 on its main domains, including Google Search, Gmail, and YouTube. Their case study covers the challenges of implementing IPv6 at a large scale and the benefits derived from the transition. Google's experience implementing IPv6 within its corporate network. Motivated by the exhaustion of IPv4 addresses and a desire to break the inertia surrounding IPv6 adoption, Google initiated a grassroots effort driven by enthusiastic volunteers.

The methodology emphasized a global and iterative approach, ensuring IPv6 implementation across all aspects of the corporate network. Challenges were encountered in networking, including the lack of enterprise IPv6 features in major vendors and in application and client software, where whitelists and OS (Operating System) support posed hurdles.

The lessons learned underscore the complexity of IPv6 migration, touching every facet of the organization. Unexpected challenges, such as MTU (Maximum transmission unit) issues and immature OS support, prolonged the project beyond its initial timeline. Google emphasizes the need for a holistic approach, testing every IPv6 feature due to the prevalence of new and sometimes buggy code. Despite the challenges, the current status reveals substantial IPv6 access for engineers, and ongoing efforts focus on enabling IPv6 support for internal tools, highlighting the evolving nature of IPv6 integration within a large enterprise network [26].

2.4.2 Comcast IPv6 Rollout

In early 2010, Comcast initiated IPv6 technology trials focused on high-speed data and Internet services. The document covers various technologies, including 6to4, 6RD, Native Dual Stack, and Dual Stack Lite, and explores Comcast's experiences and observations.

Comcast's 6to4 deployment involved using on-network relays to improve performance, emphasizing the reduction of latency by over 50% compared to open relays operated by third parties. The document acknowledges the challenges of 6to4, advocating for its diminishing use over time.

Comcast's deployment of 6RD technology, addressed its advantages over 6to4 but noting challenges such as limited Border Relay (BR) implementations. The document stresses the importance of the quantity and location of 6RD BRs in impacting end-user experience and IPv6 geo-location.

Native Dual Stack emerged as a central component of Comcast's IPv6 program, supporting trial and production deployment. The document details the upgrade and enablement of Cable

Modem Termination Systems (CMTS) for IPv6 support and the controlled introduction of native dual stack in specific network areas.

Dual Stack Lite is mentioned as part of Comcast's trial plans, although there are no immediate deployment plans beyond a limited technology trial. Content and services, back-ofice considerations, World IPv6 Day observations, and the conclusion emphasizing the preference for native dual stack are also covered.

The document concludes with acknowledgments and references, providing valuable information for the community involved in IPv6 transition efforts.

The document "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion" provides insights into Comcast's deployment of Dual Stack Lite (DS-Lite) technology, which is mentioned as part of Comcast's trial plans. Although there are no immediate deployment plans beyond a limited technology trial, the document covers the implications of DS-Lite in the context of IPv4 exhaustion, which aligns with the focus on high-speed data and Internet services in the given sentence [27].

Additionally, the article "A Comprehensive Survey of the Most Important IPv4aaS and IPv6 Transition Technologies, Their Implementations and Performance Analysis" by D'Yab discusses benchmarking methodologies for network interconnect devices, including IPv6 transition technologies. This is relevant as it provides a comprehensive overview of IPv6 transition technologies, which aligns with the various technologies covered in the given sentence [28].

Furthermore, the paper "Gateway-Initiated Dual-Stack Lite Deployment" introduces Gateway-Initiated Dual-Stack Lite (GI-DS-Lite) as a variant of DS-Lite, which applies to network architectures using point-to-point tunnels. This reference is relevant as it aligns with the discussion of DS-Lite in the given sentence, emphasizing the preference for a native dual-stack over DS-Lite [29].

Akamai IPv6 Adoption

Akamai, a prominent content delivery network (CDN) provider, has been at the forefront of incorporating IPv6 into its global infrastructure to enhance content delivery eficiency [30]. The incorporation of IPv6 into Akamai's infrastructure has been a subject of evaluation, with studies focusing on measuring IPv6 adoption from the perspective of a website operator and assessing the impact of adding IPv6 to a website on its users [31]. Additionally, research has delved into the impact of Domain Name System (DNS) resolvers on content delivery network (CDN) performance, providing insights into the causal model that captures the influence of the DNS service on the throughput performance experienced by clients accessing resources hosted by the Akamai CDN [32].

2.4.3 Facebook's IPv6 Implementation

In the realm of IPv6 implementation, Facebook has undertaken a notable stride towards establishing an IPv6-only internal network, as outlined by Paul Saab during the 2014 v6 World Congress in Paris. The fundamental motivation behind this initiative is to alleviate the com-plexities associated with maintaining a dual-stack (IPv4/IPv6) internal network. Saab suggests a pragmatic approach: transition the internal network to IPv6-only and retain dual-stack capa-bilities at the network periphery for interactions with the legacy IPv4 Internet. The challenges encountered by Facebook in this venture encompassed issues with vendor equipment, software applications, and the prevalence of IPv4-only code in development. An innovative solution involved removing IPv4 from developers' machines to enforce IPv6-centric coding practices.

The statistics presented in Saab's presentation underscore the success of Facebook's IPv6 migration strategy. Notably, 100% of the hosts considered crucial have transitioned to IPv6, signaling the phased retirement of IPv6-incapable hosts. Internal trafic now predominantly operates over IPv6, constituting 75% of the total trafic, with a targeted goal to achieve 100% by Q3 2014 or earlier. Moreover, a substantial proportion of trafic in and out of HHVM (HipHop Virtual Machine) and memcache exclusively utilizes IPv6. Facebook envisions the culmination of this transition with a bold objective of becoming 100% IPv6-only within the next 2-3 years. These achievements exemplify Facebook's commitment to embracing IPv6 and offer valuable insights into the challenges and triumphs of such a large-scale migration. The successful execution of Facebook's IPv6 initiative is an illuminating case study for organizations contemplating similar transitions in the face of IPv4 exhaustion.

Facebook, as a major social media platform, has encountered the challenge of transitioning to IPv6 while accommodating a large user base. The strategies employed for a gradual IPv6 deployment without disrupting user experience can be better understood through quantitative analysis of the extent of IPv6 deployment [33]. Additionally, insights into transition strategies for enterprise networks, which may apply to a large-scale platform like Facebook, can be gained from studies focusing on IPv4 to IPv6 transition strategies for enterprise networks [34]. Furthermore, the optimization of migration costs for legacy network migration to software-defined IPv6 networks is crucial for a seamless transition, and this can provide valuable insights into the strategies that Facebook may have employed [35].

2.4.4 ARIN's IPv6 Case Studies

The American Registry for Internet Numbers (ARIN) has emerged as a pivotal player in IPv6 adoption, actively contributing to the transition within the North American region. ARIN's engagement in IPv6 deployment is underscored by its integral role in addressing the challenges posed by the exhaustion of IPv4 addresses [36]. As IPv4 resources dwindle, ARIN has been at the forefront of facilitating the migration to IPv6, recognizing the imperative to embrace the larger address space offered by IPv6 [37].

ARIN's involvement in IPv6 adoption is notably reflected in its address allocation practices. With the exhaustion of its IPv4 address pool, ARIN has adjusted its allocation policies, emphasizing the necessity for organizations to incorporate IPv6 into their network infrastructure. ARIN's exhaustion of IPv4 addresses signifies a critical milestone, positioning it as the fourth Regional Internet Registry (RIR) to deplete its IPv4 resources [38]. In response to this scarcity, ARIN has been compelled to adopt a strategic approach, reserving its remaining IPv4 addresses for cases requiring only a small block, thus prioritizing and expediting the IPv6 transition [39].

Furthermore, ARIN's influence extends to documenting case studies illuminating successful IPv6 implementations within the North American region. These case studies offer valuable insights into the strategies and best practices adopted by diverse organizations, providing a rich knowledge repository for entities navigating the complexities of IPv6 deployment. As ARIN continues to navigate the evolving landscape of Internet address resources, its commitment to fostering IPv6 adoption remains a cornerstone in addressing the evolving needs of a networked world.

2.4.5 Government and Defense IPv6

The early embrace of IPv6 by governments and defense organizations worldwide stands out as a noteworthy trend, reflecting a strategic shift in the critical infrastructure landscape. Case studies from these entities offer invaluable insights into the intricate aspects of IPv6 adoption, shedding light on security considerations, meticulous planning, and the execution strategies employed in the transition process.

The imperative to transition to IPv6 is underscored by the depletion of IPv4 addresses, necessitating a larger and more sustainable address space provided by IPv6 [40]. The strategic importance of this transition is particularly evident in the defense sector, where entities such as the US Department of Defense have articulated plans to integrate IPv6 capabilities across all their IP networks [41]. This strategic vision emphasizes the role of IPv6 in fortifying the technological infrastructure of defense organizations, aligning with the evolving demands of modern warfare and secure communications.

In addition to the defense sector, the readiness of government organizations to embrace cloud computing serves as a barometer of internal maturity, and the feasibility of adopting transformative technologies like IPv6 [40]. As governments worldwide recognize the intrinsic value of IPv6, case studies become instrumental in elucidating the nuances of this technological shift, from risk assessments to seamless execution.

The Canadian Institute of Chartered Accountants has contributed to this narrative by developing guidelines that assist organizations in navigating information technology governance issues, a crucial aspect of transitioning to IPv6 [42]. This underscores the multidimensional challenges organizations face during the IPv6 adoption process, extending beyond technical considerations to encompass governance and risk management.

Furthermore, on the international stage, the European Commission has integrated IPv6 adoption into its e-Europe Action Plan, emphasizing its significance in government and organizational contexts [43]. The recognition of IPv6 as a key enabler aligns with the broader digital strategies of nations and organizations, positioning IPv6 as a cornerstone for future-proofing and enhancing digital capabilities.

These case studies offer a comprehensive panorama of the global IPv6 landscape, illustrating the diverse strategies, challenges, and benefits associated with its adoption across the government and defense sectors.

2.5 Summery: Open Challanges

In conclusion, the evolution from IPv4 to IPv6 is not merely an expansion of address space but a holistic response to the intricate demands of the digital age, encompassing connectivity, performance, and security considerations. The challenges in IPv6 adoption are multifaceted, ranging from security concerns and monitoring complexities to compatibility issues between IPv4 and IPv6. Security challenges, particularly those related to IPv6 extension headers and potential Denial of Service (DoS) attacks, underscore the need for a nuanced and proficient approach to IPv6 implementation. Proposed solutions advocate education, collaborative efforts in protocol design improvement, and the development of a defendable architecture tailored to IPv6 security challenges. Moreover, challenges in monitoring IPv6 networks necessitate innovative techniques and comprehensive data structures for effective host identification.

The Teredo protocol's role in IPv6 implementation is highlighted, emphasizing its significance in enabling IPv6 connectivity by encapsulating IPv6 packets within IPv4 User Datagram Protocol (UDP) datagrams. Real-world case studies from organizations such as Google, Comcast, Akamai, Facebook, ARIN, and government/defense entities provide valuable insights into the challenges and strategies employed during IPv6 adoption. These case studies underscore the importance of addressing technical, security, and strategic considerations for successful implementation.

In summary, the literature review paints a nuanced picture of the IPv6 landscape, emphasizing the complexity and multidimensionality of its adoption. Successful implementation requires a comprehensive approach that includes education, collaboration, strategic planning, and ongoing research to overcome the challenges posed by security, monitoring, and compatibility issues. The real-world case studies serve as valuable sources of practical insights, highlighting the diverse approaches taken by organizations to embrace IPv6 and providing a foundation for further research and development in this crucial area of network technology.

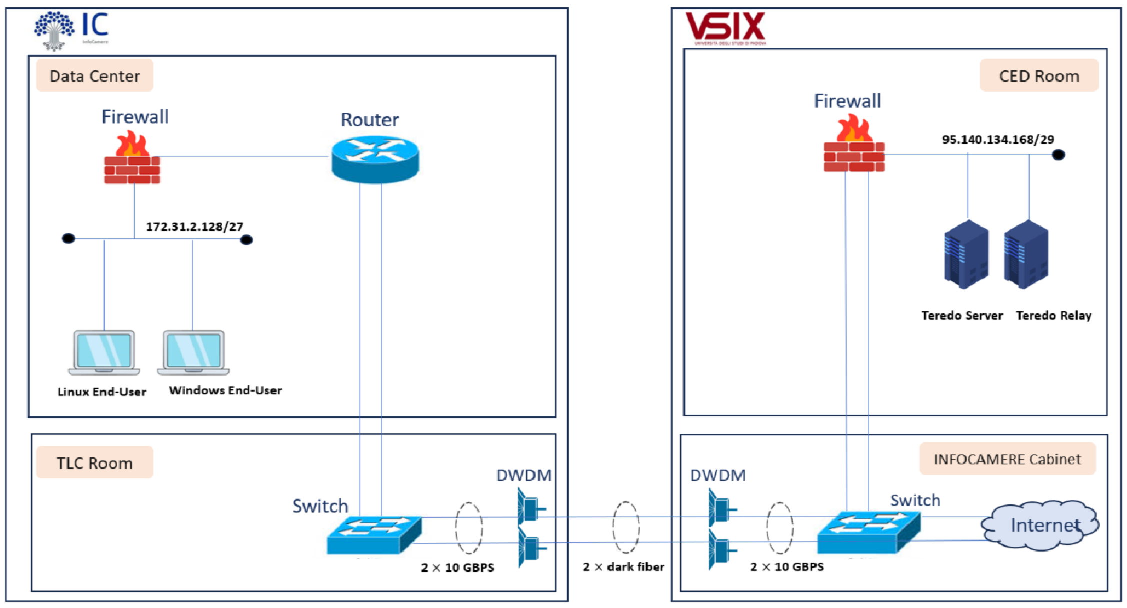
3 models

The primary objective of this section is to provide a comprehensive and detailed description of the methods employed in this master's thesis project. By utilizing the Teredo tunneling system, this thesis addresses the configuration challenges organizations encounter in adapting to evolving Internet Protocols.

3.1 Experimental Facilities and Setup

In pursuing IPv6 promotion, this master's thesis focuses on utilizing the Teredo tunneling system within the operational network infrastructure of organizations that desire to provide IPv6 connectivity to their end users. InfoCamere, which is a pertinent use case study in this master's thesis project, embodies the challenges organizations face in adapting to the evolving landscape of Internet Protocols. The primary objective is to empower end-users within the client-side network with a seamless IPv6 experience encapsulated within a network instruction set fully aligned with IPv4 protocols. Importantly, this isolation from IPv6 connectivity is intentional, mirroring the prevalent operational model—subsequently, this experiment endeavors to bridge this divide by introducing IPv6 connectivity to end users. A pivotal component in this transformation is the Teredo server providers, which in this project case is an Internet Exchange Point, VSIX company, strategically employed to facilitate the integration of IPv6 capabilities, thereby enriching the overall network infrastructure within organizational confines.

The infrastructure for the IPv6 promotion project encompasses server machines operating on a Linux platform, strategically positioned behind NAT and Firewall configurations within the client-side network. These machines are interconnected through an Internet Router (RT), which interfaces with the IPv4 public Internet. Noteworthy is the exclusive reliance on IPv4 protocols throughout the entire client-side network infrastructure, a characteristic that underlines the challenges associated with transitioning to IPv6.



Policy Firewall Infocamere

Source	Destination	Protocol/Category
172.31.2.128/27	95.140.134.168/29	any
172.31.2.128/27	Internet	Computer/Internet Software/Downloads
172.31.2.128/27	Internet	ICMP

NAT Firewall Infocamere

Original Source	Original Destinatio	Protocol	Translated Source	Translated Destination
172.31.2.128/27	Any	Any	80.82.10.180	Original

Figure 3.1: Nettwork Archtittectture of Teredo Servers and End Users wtitth tthetir Ftirewall Poltictties

The successful implementation of Teredo within the Teredo Server and Teredo Relay (within the Internet Exchanger environment in this project case) necessitates specific requisites. A minimum of three public IPv4 addresses is mandated for the Teredo Server, a number reduced to two in cases where the Teredo Server and Teredo Relay components coexist within the same machines. Similarly, the Teredo Relay demands at least two public IPv6 addresses, a count halved in scenarios mirroring the consolidation of server and relay functionalities. The most critical requirement involves allocating a dedicated public IPv6/32 network exclusively earmarked for Teredo Clients.

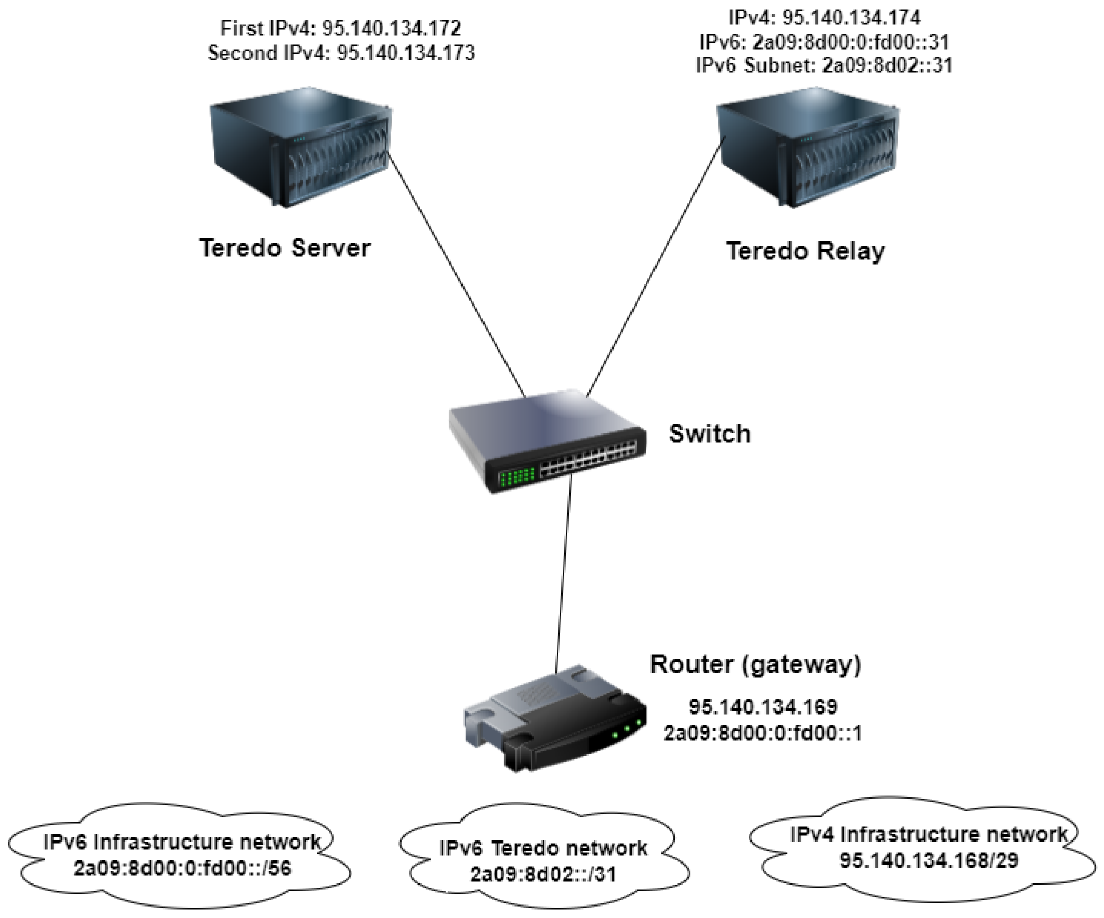


Figure 3.2: Teredo Server and Teredo Relay IP conftiguratttions

As demonstrated in Fig. 3.2, strategically placed within the same subnet, the Teredo Server and Relay aim to simplify communication and address uncertainties regarding port configurations. Firewall settings on the router for all involved IP addresses have been temporarily disabled within the client-side network, ensuring unimpeded trafic flow. Considerations for enhancing security involve selectively opening only the essential ports, prompting the need for subsequent testing to ascertain the optimal configuration.

Crucial to the success of the Teredo Relay server is enabling IPv6 routing, accomplished through the inclusion of the `net.ipv6.conf.all.forwarding=1` directive within the

`/etc/sysctl.conf` configuration file. This directive empowers Teredo Relay to route IPv6 trafic eficiently.

3.1.1 Teredo Server and Teredo Relay Setup with Dual-Stack Ipv4 and Ipv6

Miredo is an open-source Teredo client for Unix-like operating systems. The Teredo Server and Teredo Relay machines in this project are running on Linux OS, therefore Miredo emerged as the preferred Teredo client for this project due to its open-source nature, compatibility with Unix-like systems, and customizable features, making it a suitable choice for enabling IPv6 connectivity within the designated network environment. This decision stemmed from the project's overarching objective of furnishing end users with IPv6 connectivity within a designated IPv6 network. Given that Teredo facilitates IPv6 access for devices behind NAT, utilizing a versatile and customizable Teredo client to tailor the configuration settings according to the project's specific requirements was imperative.

Miredo, provided the ideal solution for this endeavor. Its compatibility with Unix-like systems like Linux ensured seamless project infrastructure integration. By leveraging Miredo, the Teredo Server and Teredo Relay machines could effectively establish Teredo tunnels and enable IPv6 connectivity for end users, circumventing the limitations imposed by NAT environments.

Furthermore, Miredo's flexibility and extensibility allowed for fine-tuning Teredo settings and configuration parameters to align with the project's objectives. This capability was crucial for ensuring optimal performance and reliability of the Teredo tunnels, thereby facilitating uninterrupted IPv6 access for end users.

The installation and configuration of the Miredo and Miredo-Server packages on the server machines with Unix-like operating systems represent pivotal milestones in the project. The correct execution of these steps ensures the seamless integration of Miredo into the network, facilitating the realization of the Teredo tunneling system.

The Miredo-Relay server boasts a distinct network configuration featuring a public IPv4 address. In this master's thesis project, the IPv4 of Teredo Relay is 95.140.134.174, and a corresponding IPv6 address is 2a09:8d00:0:fd00::31. The significance of these addresses lies in their role in establishing connectivity and relaying IPv6 trafic, bridging the gap between IPv4 and IPv6 networks within the project' client-side domain.

One notable architectural decision in the Teredo setup is the coexistence of the Teredo Server and Teredo Relay within the same subnet. This simplification was adopted for expediency, mitigating potential complexities associated with cross-subnet communication and minimizing the need for intricate firewall configurations. This approach aligns with the project's emphasis on eficiency and clarity in network communication.

Furthermore, as part of the initial configuration process, meticulous tests are conducted to determine the essential application ports required for the optimal functioning of Teredo services. These tests will guide establishing a more secure network environment by selectively opening only the necessary ports.

In conclusion, the Teredo Server setup represents a meticulous integration of components and configurations within the VSIX infrastructure, the server-side infrastructure in this master's thesis project. The choice of network parameters, subnet coexistence, and routing configurations underscores a commitment to eficiency and clarity, laying the groundwork for successful IPv6 promotion within the corporate network.

3.1.2 End User Setup

As demonstrated in Fig. 3.1 a meticulous firewall policy has been devised to ensure the secure and controlled flow of network trafic within the client-side network infrastructure, which in this project is InfoCamere company. For communications originating from the source IP range of the client network, which in this master's thesis project is 172.31.2.128/27 destined for the IP range 95.140.134.168/29, which is the network of Teredo Server and Relay, the protocol/-category parameter is set to 'any.' This rule facilitates unrestricted connectivity between specific internal hosts and designated external servers, providing flexibility for various communication types.

Additionally, a more specific firewall rule has been implemented for trafic originating from the source IP range of the end-user network 172.31.2.128/27 bound for the broader destination of the Internet. The protocol is categorized as 'Computer/Internet Software/Downloads.' This configuration allows hosts within the specified source range to access the Internet specifi-cally for downloading and installing software relevant to the Teredo and Miredo components. This targeted access ensures the fulfillment of software-related requirements without compro-mising security.

Moreover, considering the same source IP range 172.31.2.128/27 and the destination as the Internet, a specialized firewall rule has been established. The protocol for this scenario is set to 'ICMP,' allowing for the controlled transmission of Internet Control Message Protocol pack-ets. This facilitates communication, such as ping requests and responses, enabling network diagnostics and troubleshooting functionalities for the specified source hosts.

Nat Firewall Configuration Inside End User Network Infrastructure

Within the confines of the client-side network infrastructure, an additional layer of security is implemented by deploying a network address translation (NAT) firewall. This firewall policy governs the translation of internal IP addresses to a designated external address, enhancing security and privacy.

In a specific configuration, originating from the source IP range **172.31.2.128/27** with an original destination of 'any,' the NAT firewall facilitates translation. The translated source is assigned IP address 80.82.10.180, while the translated destination remains unrestricted ('any'). This policy ensures that internal hosts within the specified source range are shielded by the NAT firewall when communicating externally, further fortifying the security posture of the Infocamere network.

3.2 Methodological Procedures

In this section, we elucidate the detailed procedures and methodologies employed in configuring the experimental setup for our specific project. A key aspect of our study involves the modification of default Teredo settings to enhance the performance and adaptability of the network infrastructure. We provide a step-by-step account of the changes made to the Teredo configuration, outlining the specific parameters adjusted and the rationale behind each modification. This section serves as a comprehensive guide to the experimental design, detailing the intricacies of our methodological procedures, with a particular emphasis on the alterations made to the default Teredo settings to suit the unique requirements of our research.

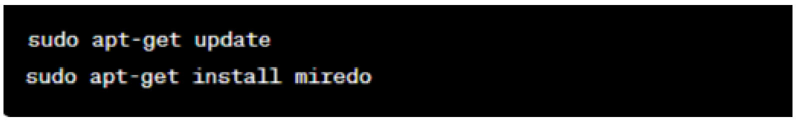
3.2.1 Teredo Installation

The initial step in deploying Teredo necessitates accessing the appropriate software repositories or downloading the requisite packages.as described in 3.1.1 section For Unix-like systems, Miredo is essentially an implementation of the Teredo client, allowing to leverage Teredo technology for IPv6 communication.

Miredo Setup on Server and Client machine

In this master's thesis project for the client-side setup On Linux distributions like Ubuntu, the Advanced Package Tool (APT) is a reliable software installation mechanism. The following commands exemplify a streamlined installation process:

Figure 3.3: Insttallting Mtiredo on Ltinux

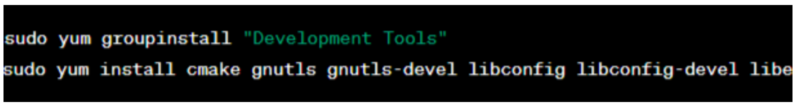


```
sudo apt-get update
sudo apt-get install miredo
```

Upon execution, the APT utility fetches the Miredo package along with its dependencies, ensuring a coherent installation environment conducive to subsequent configuration steps.

Given the absence of Miredo in the default repositories of some Linux distributions like CentOS, the installation process involves building it from the source. This necessitates the installation of essential dependencies and the compilation of Miredo using the following steps:

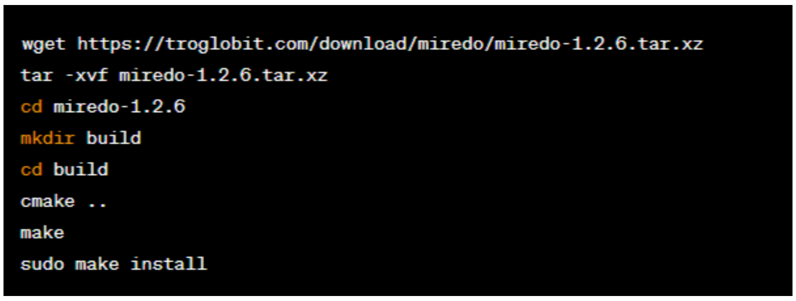
Figure 3.4: Insttallatttion of Developmentt Tools and Dependencties tin Bash Envtironmentt



```
sudo yum groupinstall "Development Tools"
sudo yum install cmake gnutls gnutls-devel libconfig libconfig-devel libe
```

Following the installation of dependencies, downloading and compiling Miredo from source is the subsequent step:

Figure 3.5: Comptilting Mtiredo From Source

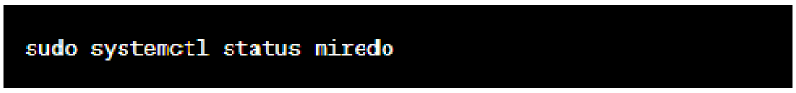


```
wget https://troglobit.com/download/miredo/miredo-1.2.6.tar.xz
tar -xvf miredo-1.2.6.tar.xz
cd miredo-1.2.6
mkdir build
cd build
cmake ..
make
sudo make install
```

This intricate process ensures the availability of Miredo on CentOS, providing a solid foundation for integrating Teredo tunneling.

Post-installation, Miredo is designed to operate seamlessly, leveraging the Teredo tunneling protocol to facilitate IPv6 connectivity. Verification of the Miredo service status offers insights into its operational state.

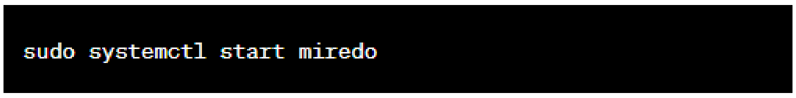
Figure 3.6: Insttallting Mtiredo on machtines wtitth Ltinux operattting systtem



```
sudo systemctl status miredo
```

Should the service not be active, manual initiation ensures its commencement:

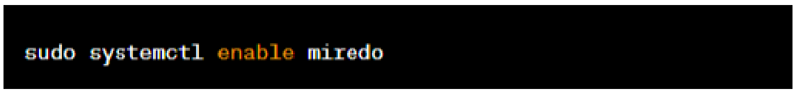
Figure 3.7: Sttarttting Mtiredo on Ltinux



```
sudo systemctl start miredo
```

To fortify Miredo's integral role within the Linux system architecture, configuring it for automatic startup upon system boot is imperative.

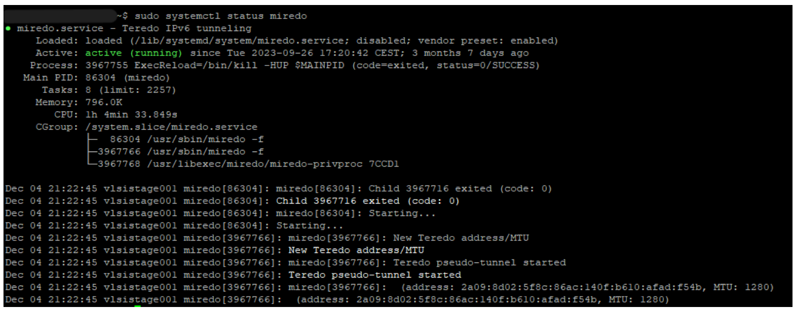
Figure 3.8: Enablting Mtiredo on Ltinux



```
sudo systemctl enable miredo
```

The status of Miredo after enabling it:

Figure 3.9: Sttattus of enabled Mtiredo on Ltinux



```
[redacted]~$ sudo systemctl status miredo
● miredo.service - Teredo IPv6 tunneling
     Loaded: loaded (/lib/systemd/system/miredo.service; disabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-09-26 17:20:42 CEST; 3 months 7 days ago
    Process: 3967755 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
   Main PID: 86304 (miredo)
      Tasks: 8 (limit: 2257)
     Memory: 796.0K
        CPU: 1h 4min 33.849s
     CGroup: /system.slice/miredo.service
             ├─  86304 /usr/sbin/miredo -f
             ├─3967766 /usr/sbin/miredo -f
             └─3967768 /usr/libexec/miredo/miredo-privproc 7CCD1

Dec 04 21:22:45 vlsistage001 miredo[86304]: miredo[86304]: Child 3967716 exited (code: 0)
Dec 04 21:22:45 vlsistage001 miredo[86304]: Child 3967716 exited (code: 0)
Dec 04 21:22:45 vlsistage001 miredo[86304]: miredo[86304]: Starting...
Dec 04 21:22:45 vlsistage001 miredo[86304]: Starting...
Dec 04 21:22:45 vlsistage001 miredo[3967766]: miredo[3967766]: New Teredo address/MTU
Dec 04 21:22:45 vlsistage001 miredo[3967766]: New Teredo address/MTU
Dec 04 21:22:45 vlsistage001 miredo[3967766]: miredo[3967766]: Teredo pseudo-tunnel started
Dec 04 21:22:45 vlsistage001 miredo[3967766]: Teredo pseudo-tunnel started
Dec 04 21:22:45 vlsistage001 miredo[3967766]: miredo[3967766]:  (address: 2a09:8d02:5f8c:86ac:140f:b610:afad:f54b, MTU: 1280)
Dec 04 21:22:45 vlsistage001 miredo[3967766]:  (address: 2a09:8d02:5f8c:86ac:140f:b610:afad:f54b, MTU: 1280)
```

To fortify Miredo's integral role within the Linux system architecture, configuring it for automatic startup upon system boot is imperative.This configuration directive guarantees the persistence of Miredo's functionality, reafirming its commitment to promoting IPv6 connectivity in tandem with IPv4 networks.

The meticulous installation and configuration of Miredo on Linux systems epitomize a strategic endeavor to bolster IPv6 promotion in contemporary networking paradigms. Through methodical procedures, from installation via package managers to configuration via system utilities, Miredo emerges as a pivotal tool, bridging the IPv4-IPv6 connectivity chasm. As subsequent chapters delve deeper into performance evaluations, user experiences, and overarching benefits, the foundational significance of Miredo's deployment remains unequivocally paramount.

3.2.2 Teredo Configurations

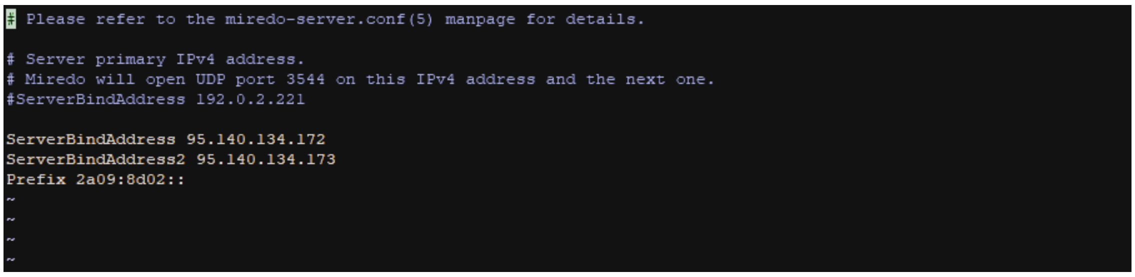
Teredo Server

Continuing with methodological procedures, in this project, the focus is on the dual-stack environment at the VSIX side, where both IPv4 and IPv6 coexist. After enabling Miredo on the server side, specific configurations were applied to the Teredo server and Teredo relay. The server's operating system supports dual-stack capabilities, allowing seamless integration of IPv4 and IPv6.

Similar to the end-user configuration, adjustments were made to the `/etc/miredo/` file, located in the directory `/etc/miredo/`. The Vim text editor was employed for this task, following the commands:

```
sudo -i
cd /etc/miredo/
sudo vim miredo-server.conf
```

Opening the 'miredo-server.conf' file in insert mode within Vim facilitated the modification of default settings to align with the requirements of our dual-stack environment. The specific configurations implemented to optimize the Teredo server and relay functionalities are depicted in the following figure.



```
# Please refer to the miredo-server.conf(5) manpage for details.

# Server primary IPv4 address.
# Miredo will open UDP port 3544 on this IPv4 address and the next one.
#ServerBindAddress 192.0.2.221

ServerBindAddress 95.140.134.172
ServerBindAddress2 95.140.134.173
Prefix 2a09:8d02::
~
~
~
~
```

Figure 3.10: Conftiguratttion setttings wtitthtin tthe 'mtiredo-server.conf' ftile on tthe VSIX dual-sttack server stide.

Figure 3.10 visually presents the content within the 'miredo-server.conf' file, showcasing the tailored configurations applied to enhance the Teredo server and relay operations in the dual-stack environment on the server side of our project which is VSIX.

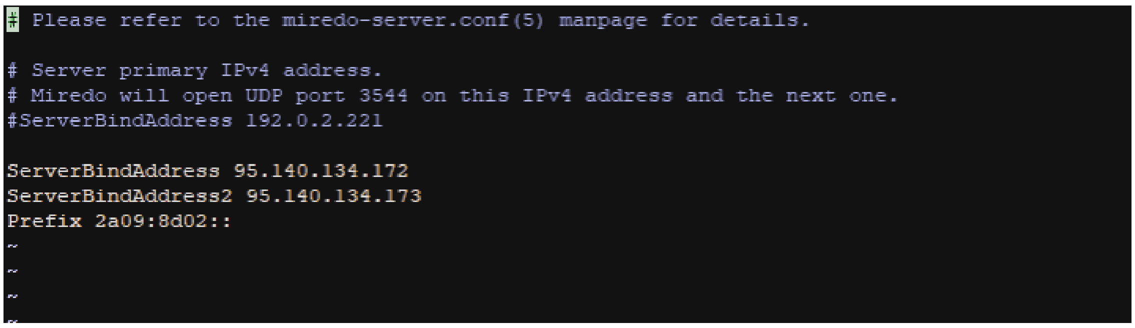
Teredo Relay

Extending our methodological procedures to include the Teredo relay in the dual-stack environment at the VSIX side, additional configurations were applied to the 'miredo-relay.conf' file. Following the successful enabling of Miredo on the server side, specific adjustments were made to the Teredo relay settings to ensure optimal performance within the dual-stack infrastructure.

The Vim text editor was utilized to initiate the modification process. The sequence of commands executed is as follows:

```
sudo -i
cd /etc/miredo/
sudo vim miredo.conf
```

By opening the 'miredo.conf' file in insert mode within Vim, default settings were modified to suit the requirements of our dual-stack environment. The screenshot below visually represents the content within the 'miredo-relay.conf' file, showcasing the tailored configurations implemented to enhance Teredo relay functionality.



```
# Please refer to the miredo-server.conf(5) manpage for details.

# Server primary IPv4 address.
# Miredo will open UDP port 3544 on this IPv4 address and the next one.
#ServerBindAddress 192.0.2.221

ServerBindAddress 95.140.134.172
ServerBindAddress2 95.140.134.173
Prefix 2a09:8d02::
~
~
~
~
```

Figure 3.11: Conftiguratttion setttings wtitthtin tthe 'mtiredo.conf' ftile on tthe VSIX dual-sttack relay stide.

Figure 3.11 illustrates the specific parameters adjusted in the 'miredo-relay.conf' file, providing insights into the configurations made to optimize Teredo relay operations within the dual-stack environment at the VSIX side.

Teredo in End-user with Unix-like Operating Systems

This section details the methodological procedures undertaken to configure the Teredo client within the IPv4 infrastructure at Infocamere. The client-side operating system, Ubuntu 22.04.2 LTS, was selected for its compatibility with our experimental requirements. As the preceding section (Sec. 3.2.1) outlined, the Miredo tool enabled Teredo functionality.

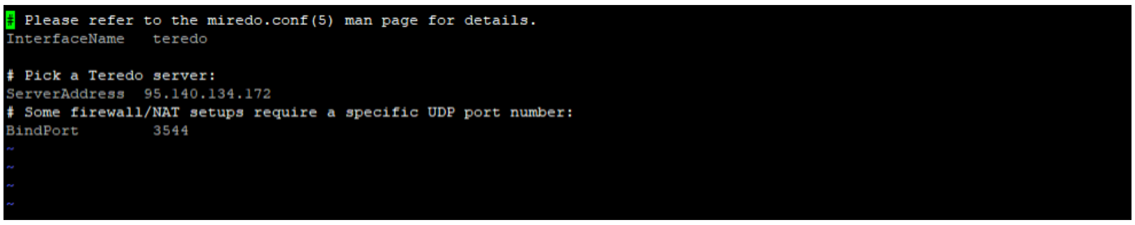
Upon enabling Miredo, the configuration directory can be accessed by executing the following commands. Overall, the configuration demonstrated in this chapter showcases the successful integration of

```
sudo -i
cd /etc/miredo/
```

With the directory changed, the configuration file, 'miredo.conf', was edited using a text editor for example in the present project Vim tex editor has been used.

```
sudo vim miredo.conf
```

Opening the 'miredo.conf' file in insert mode allowed for the incorporation of specific adjustments to tailor the Teredo client to the IPv4 infrastructure at Infocamere.



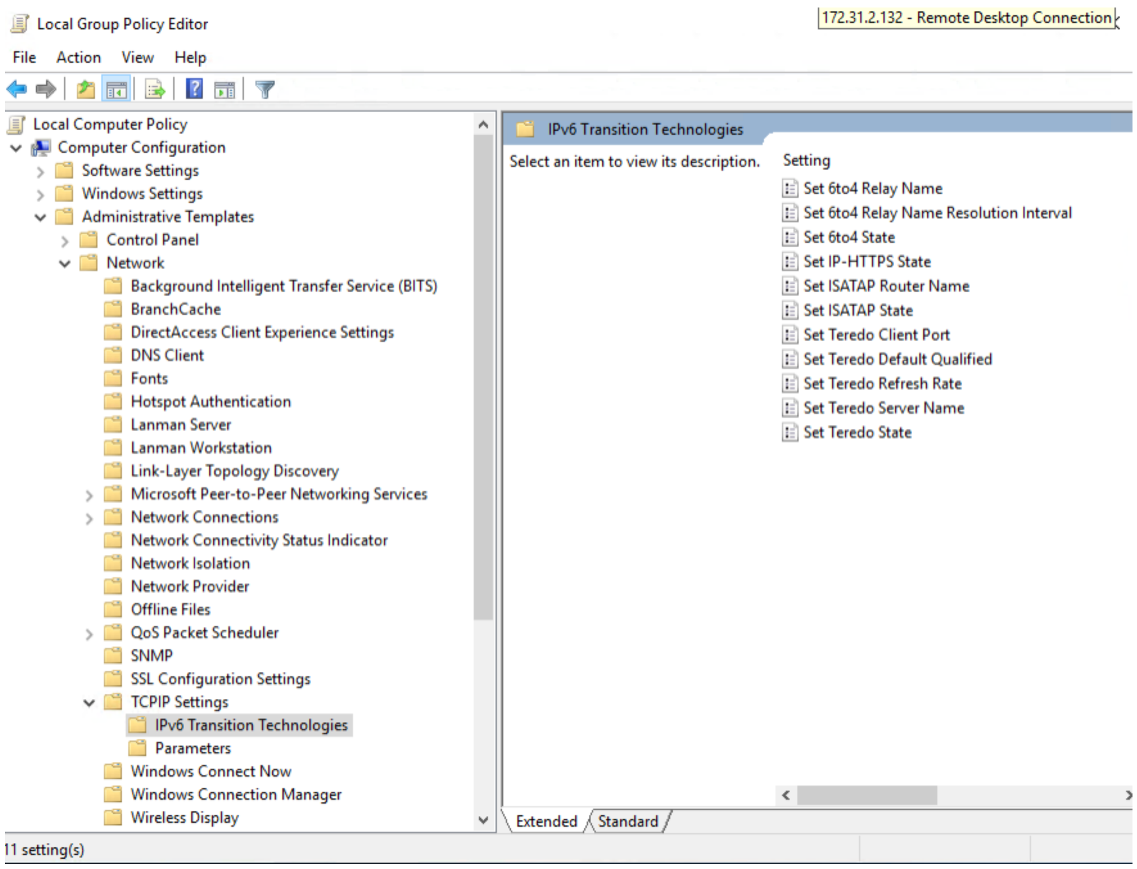
```
# Please refer to the miredo.conf(5) man page for details.
InterfaceName   teredo

# Pick a Teredo server:
ServerAddress  95.140.134.172
# Some firewall/NAT setups require a specific UDP port number:
BindPort        3544
~
~
~
~
```

Figure 3.12: Conftiguratttion setttings wtitthtin tthe 'mtiredo.conf' ftile on tthe Ubunttu cltientt stide.

The Figure 3.12 provides a visual representation of the actual content within the 'miredo.conf' file on the client side. These configurations encapsulate the intricacies of our adjustments, demonstrating the specific parameters modified to optimize Teredo functionality within the IPv4 infrastructure of client-side network infrastructure.

Teredo in End-user with Windows Operating Systems

In the pursuit of configuring Teredo on a Windows 10 client, a systematic step-by-step approach was undertaken within the Local Group Policy Editor. This configuration process involved navigating through specific hierarchies within the Local Computer Policy. Specifically, attention was directed to TCP/IP Settings in the Administrative Templates within the Network section.


Local Group Policy Editor

172.31.2.132 - Remote Desktop Connection

File Action View Help

- Local Computer Policy
  - Computer Configuration
    - Software Settings
    - Windows Settings
    - Administrative Templates
      - Control Panel
      - Network
        - Background Intelligent Transfer Service (BITS)
        - BranchCache
        - DirectAccess Client Experience Settings
        - DNS Client
        - Fonts
        - Hotspot Authentication
        - Lanman Server
        - Lanman Workstation
        - Link-Layer Topology Discovery
        - Microsoft Peer-to-Peer Networking Services
        - Network Connections
        - Network Connectivity Status Indicator
        - Network Isolation
        - Network Provider
        - Offline Files
        - QoS Packet Scheduler
        - SNMP
        - SSL Configuration Settings
        - TCPIP Settings
          - IPv6 Transition Technologies
          - Parameters
        - Windows Connect Now
        - Windows Connection Manager
        - Wireless Display

IPv6 Transition Technologies

Select an item to view its description.

Setting
Set 6to4 Relay Name
Set 6to4 Relay Name Resolution Interval
Set 6to4 State
Set IP-HTTPS State
Set ISATAP Router Name
Set ISATAP State
Set Teredo Client Port
Set Teredo Default Qualified
Set Teredo Refresh Rate
Set Teredo Server Name
Set Teredo State

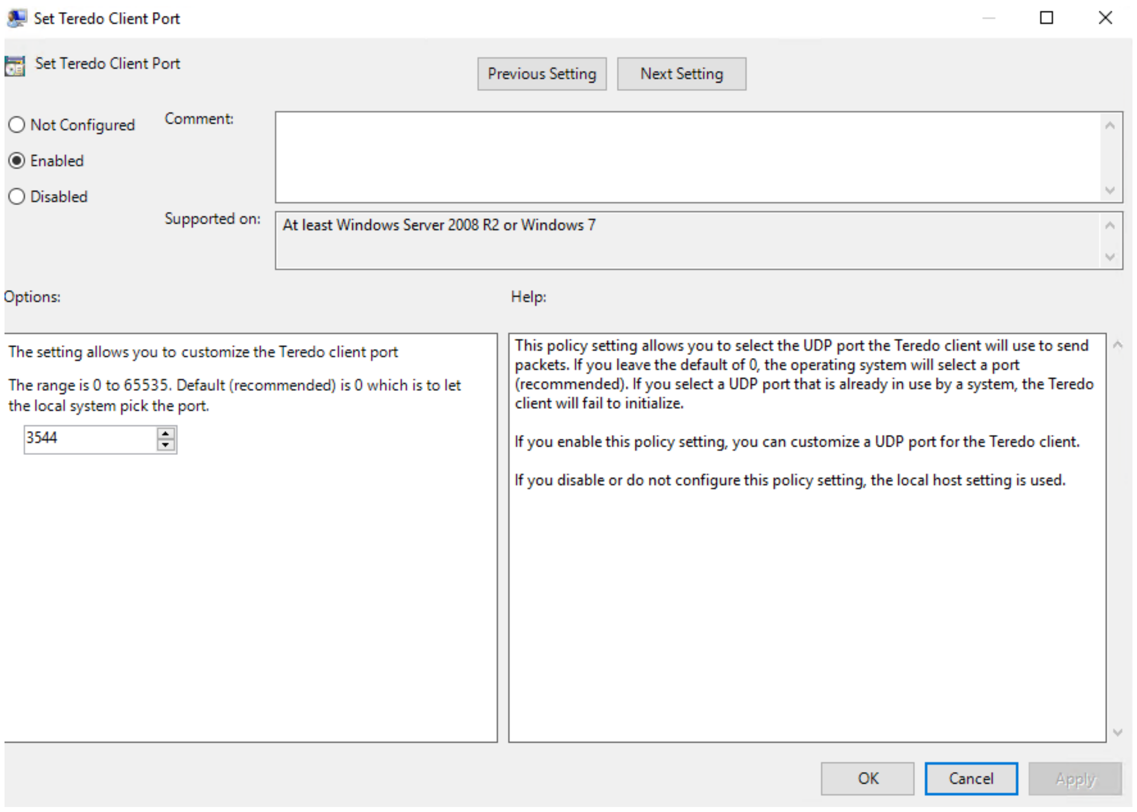
Extended Standard

11 setting(s)

Figure 3.13: Teredo Setttings Locatttion on Wtindows 10 Cltientt

Set Teredo Client port This policy parameter provides the flexibility to designate the UDP port used by the Teredo client for packet transmission. Opting for the default value of 0 allows the operating system to select a port, which is recommended automatically. However, specifying a UDP port already in use by the system will result in the Teredo client failing to initialize.

By activating this policy parameter, customization of a UDP port for the Teredo client becomes possible. Conversely, if this policy is disabled or remains unconfigured, the local host setting will be applied.


Set Teredo Client Port

Set Teredo Client Port

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Windows Server 2008 R2 or Windows 7

Options:

The setting allows you to customize the Teredo client port

The range is 0 to 65535. Default (recommended) is 0 which is to let the local system pick the port.

3544

Help:

This policy setting allows you to select the UDP port the Teredo client will use to send packets. If you leave the default of 0, the operating system will select a port (recommended). If you select a UDP port that is already in use by a system, the Teredo client will fail to initialize.

If you enable this policy setting, you can customize a UDP port for the Teredo client.

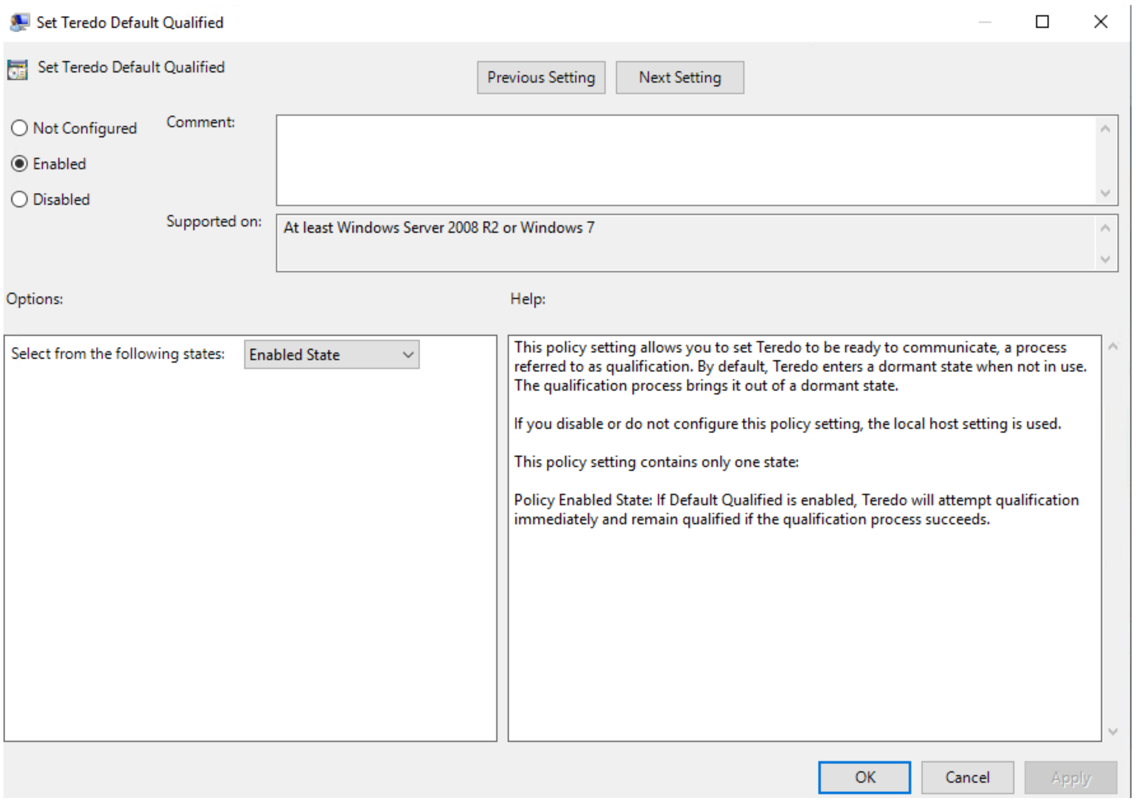
If you disable or do not configure this policy setting, the local host setting is used.

Figure 3.14: Teredo Cltientt Portt conftiguratttion on Wtindows 10 cltientt

Set Teredo Default Qualified This policy configuration enables the initiation of the Teredo communication readiness process, known as qualification. By default, Teredo assumes a dormant state during periods of inactivity. The qualification process is designed to transition it out of this dormant state.

The local host setting will be applied if it has been turned off or this policy setting has been left unconfigured. The policy setting itself encompasses a single state:t

Policy Enabled State: If Default Qualified is activated, Teredo will promptly undertake the qualification process and maintain the qualified status if the qualification process is successful.



Set Teredo Default Qualified

Set Teredo Default Qualified

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Windows Server 2008 R2 or Windows 7

Options:

Select from the following states: Enabled State

Help:

This policy setting allows you to set Teredo to be ready to communicate, a process referred to as qualification. By default, Teredo enters a dormant state when not in use. The qualification process brings it out of a dormant state.

If you disable or do not configure this policy setting, the local host setting is used.

This policy setting contains only one state:

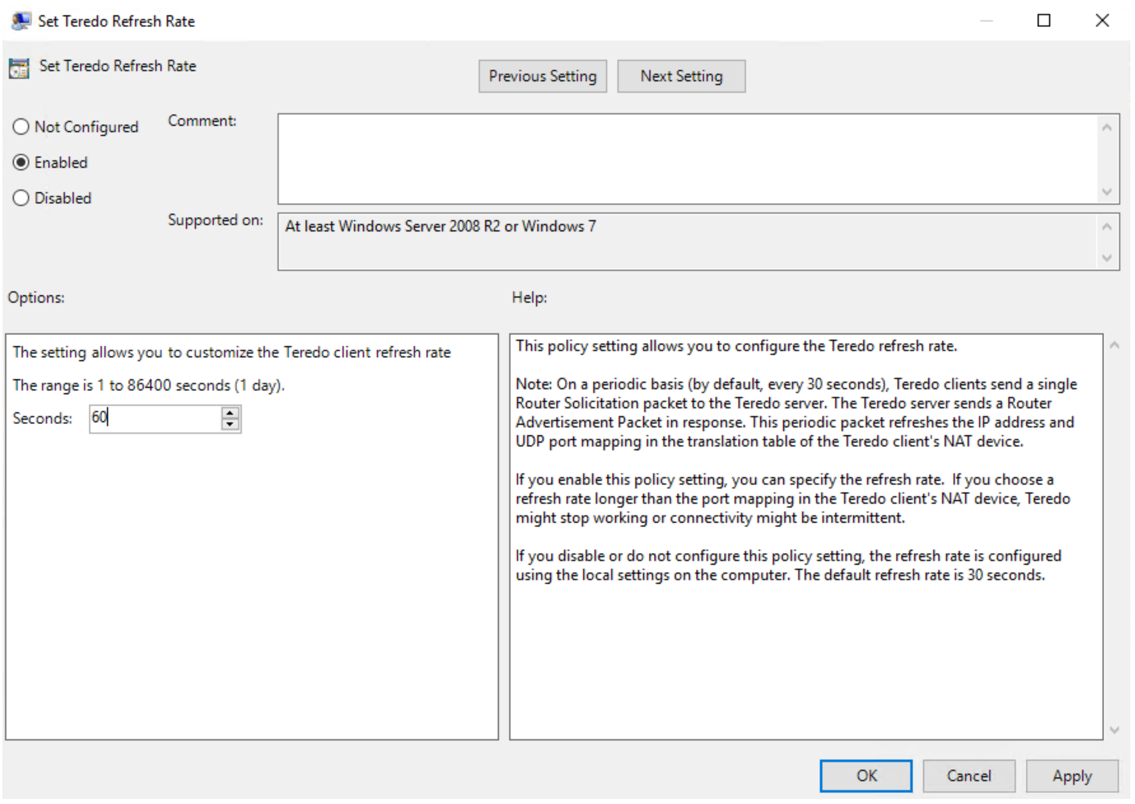
Policy Enabled State: If Default Qualified is enabled, Teredo will attempt qualification immediately and remain qualified if the qualification process succeeds.

Figure 3.15: Teredo Defaultt Qualtiftied conftiguratttion on Wtindows 10 cltientt

Set Teredo Refresh Rate This policy configuration provides the ability to adjust the refresh rate for Teredo. It is important to note that Teredo clients dispatch a Router Solicitation packet to the Teredo server at regular intervals (typically every 30 seconds by default). In response, the Teredo server sends a Router Advertisement Packet, refreshing the IP address and UDP port mapping in the translation table of the Teredo client's NAT device.

Enabling this policy setting allows customization of the refresh rate. However, it is crucial to select a rate that is at most the port mapping duration in the Teredo client's NAT device; otherwise, Teredo functionality may be compromised, leading to intermittent connectivity issues.

If it is chosen to disable or not configure this policy setting, the refresh rate will be determined by the local settings on the computer, with the default rate set at 30 seconds.



Set Teredo Refresh Rate

( ) Not Configured
(●) Enabled
( ) Disabled

Comment:

Supported on: At least Windows Server 2008 R2 or Windows 7

Options:

The setting allows you to customize the Teredo client refresh rate

The range is 1 to 86400 seconds (1 day).

Seconds: 60

Help:

This policy setting allows you to configure the Teredo refresh rate.

Note: On a periodic basis (by default, every 30 seconds), Teredo clients send a single Router Solicitation packet to the Teredo server. The Teredo server sends a Router Advertisement Packet in response. This periodic packet refreshes the IP address and UDP port mapping in the translation table of the Teredo client's NAT device.

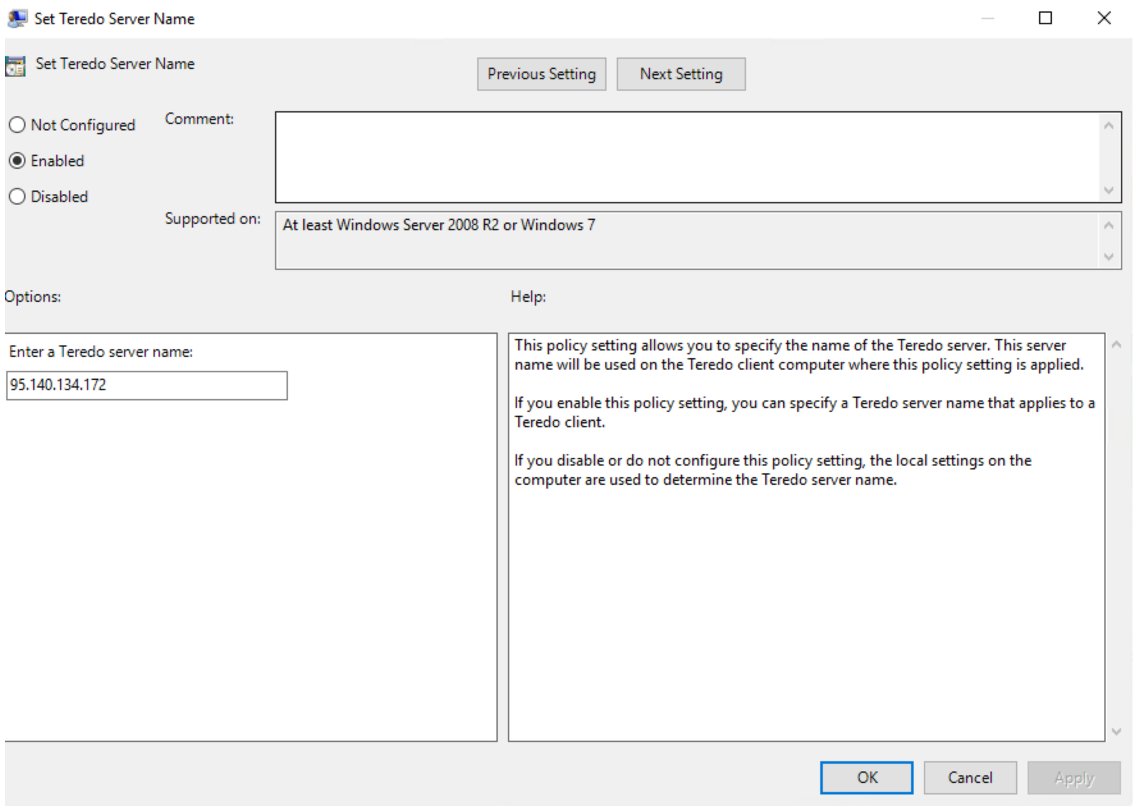
If you enable this policy setting, you can specify the refresh rate. If you choose a refresh rate longer than the port mapping in the Teredo client's NAT device, Teredo might stop working or connectivity might be intermittent.

If you disable or do not configure this policy setting, the refresh rate is configured using the local settings on the computer. The default refresh rate is 30 seconds.

Figure 3.16: Teredo Refresh Ratte conftigurattion on Wtindows 10 cltientt

Set Teredo Server Name This policy parameter enables the definition of the Teredo server name, which will be utilized on the Teredo client computer where this policy setting is implemented. Activating this policy setting provides the option to designate a specific Teredo server name for a given Teredo client.

Conversely, if it is chosen to disable or leave this policy setting unconfigured, the Teredo server name will be determined by the local settings on the computer. In this project, the Teredo Server Name should be the IPv4 address of the Teredo Server, which has been explained in section 3.1.1.


Set Teredo Server Name

Set Teredo Server Name

( ) Not Configured [✓ Enabled] ( ) Disabled

Comment:

Supported on: At least Windows Server 2008 R2 or Windows 7

Options:

Enter a Teredo server name: 95.140.134.172

Help:

This policy setting allows you to specify the name of the Teredo server. This server name will be used on the Teredo client computer where this policy setting is applied.

If you enable this policy setting, you can specify a Teredo server name that applies to a Teredo client.

If you disable or do not configure this policy setting, the local settings on the computer are used to determine the Teredo server name.

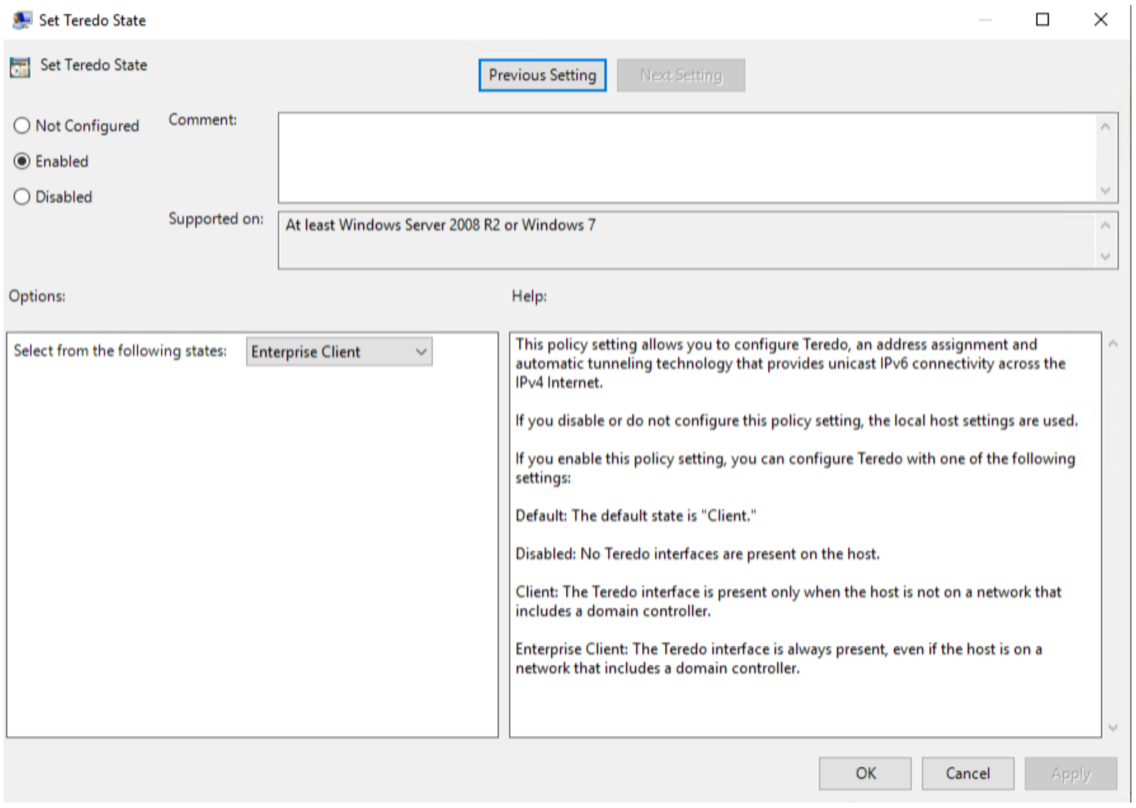
Figure 3.17: Teredo Server Name conftiguratttion on Wtindows 10 cltientt

Set Teredo State This policy setting configures Teredo, a technology designed for address assignment and automatic tunneling, providing unicast IPv6 connectivity across the IPv4 Internet.

If this policy setting is disabled or left unconfigured, the local host settings take precedence, governing the behavior of Teredo.

Upon enabling this policy setting, customization options become available, allowing for the selection of one of the following settings:

Default: This designates the default state as ”Client.” Disabled: Under this option, no Teredo interfaces are active on the host. Client: The Teredo interface becomes active only when the host is not connected to a network with a domain controller. Enterprise Client: Opting for this setting ensures that the Teredo interface always remains active, even when the host is on a network that includes a domain controller. In conculusion, in earlier versions of Windows, such as Windows 7 and Windows 8, Teredo was implemented as a default feature, providing a seamless way for devices behind NAT to access the IPv6 Internet. However, with the transition



( ) Not Configured
(●) Enabled
( ) Disabled

Supported on: At least Windows Server 2008 R2 or Windows 7

Select from the following states: Enterprise Client

This policy setting allows you to configure Teredo, an address assignment and automatic tunneling technology that provides unicast IPv6 connectivity across the IPv4 Internet.

If you disable or do not configure this policy setting, the local host settings are used.

If you enable this policy setting, you can configure Teredo with one of the following settings:

Default: The default state is "Client."

Disabled: No Teredo interfaces are present on the host.

Client: The Teredo interface is present only when the host is not on a network that includes a domain controller.

Enterprise Client: The Teredo interface is always present, even if the host is on a network that includes a domain controller.

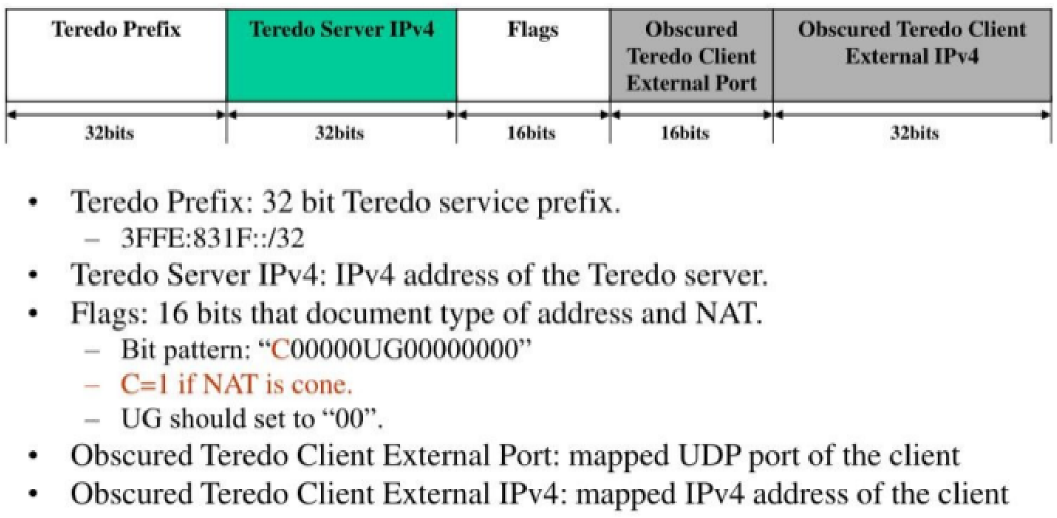
Figure 3.18: Teredo Sttatte conftiguratttion on Wtindows 10 cltientt

to Windows 10, there have been notable changes in Teredo functionality. While still present, Teredo may encounter issues such as remaining stuck in probe mode, as observed in our project tests. These changes reflect the evolving landscape of networking technologies and the need for adaptability in addressing IPv6 connectivity challenges across different Windows versions.

3.3 Analytical Framework and Interpretation

This network setup and configuration have enabled Teredo on the end-user with a Unix-like OS, Within an IPv4 infrastructure. The Teredo IPv6 address assigned to the end-user with Linux operating system is **2a09:8d02:5f8c:86ac:140f:b610:afad:f54b**. The Teredo Server responsible for managing the Teredo tunneling is configured with the prefix **2a09:8d02::** as shown in 3.11 and has IPv4 addresses 95.140.134.172 and 95.140.134.173.

The Teredo IPv6 address **2a09:8d02:5f8c:86ac:140f:b610:afad:f54b** combines various components and correctly identifies parts as demonstrated in 3.18. Breaking down the structure: ”2a09:8d02:” is the Teredo prefix. 5f8c:86ac represents the hexadecimal version of the Miredo relay IPv4 address, 95.140.134.172. The remaining part, 140f:b610:afad:f54b, is generated based on the Teredo client’s information. The Teredo address format is constructed below figure.


Teredo Prefix
Teredo Server IPv4
Flags
Obscured Teredo Client External Port
Obscured Teredo Client External IPv4
32bits
32bits
16bits
16bits
32bits

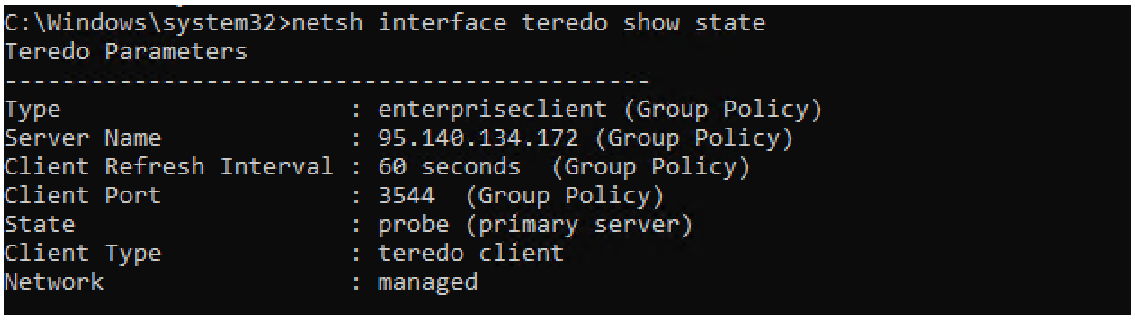
- Teredo Prefix: 32 bit Teredo service prefix.
  - 3FFE:831F::/32
- Teredo Server IPv4: IPv4 address of the Teredo server.
- Flags: 16 bits that document type of address and NAT.
  - Bit pattern: “C00000UG00000000”
  - C=1 if NAT is cone.
  - UG should set to “00”.
- Obscured Teredo Client External Port: mapped UDP port of the client
- Obscured Teredo Client External IPv4: mapped IPv4 address of the client

Figure 3.19: Teredo Address Formatt Encodting

Teredo relay server, identified as ”ens160,” has the IPv4 address 95.140.134.174. This server is an intermediary for Teredo trafic, facilitating communication between Teredo clients and the IPv6 Internet. The relay server is actively relaying Teredo trafic, as indicated by the non-zero packet counts in the RX (received) and TX (transmitted) directions.

Meanwhile, the Teredo client, represented by the ”ens160” and ”teredo” interfaces, has the Teredo IPv6 address **2a09:8d02:5f8c:86ac:140f:b610:afad:f54b**. The client is communicating over IPv4 through the Teredo relay server. The ”Miredo” client, represented by the ”ens160” and ”teredo” interfaces, has the IPv4 address 172.31.2.131 and is actively sending and receiving Teredo trafic.

On the other hand, the interpratation of the configuration of end user with windows operating system is unseccessful since the status of Teredo remained stuck in probe mode 4.4 . This state indicates active attempts to establish a Teredo tunnel to the Teredo server but transitions quickly to ofline status after a short duration.



```
C:\Windows\system32>netsh interface teredo show state
Teredo Parameters
---------------------------------------------
Type                    : enterpriseclient (Group Policy)
Server Name             : 95.140.134.172 (Group Policy)
Client Refresh Interval : 60 seconds  (Group Policy)
Client Port             : 3544  (Group Policy)
State                   : probe (primary server)
Client Type             : teredo client
Network                 : managed
```

Figure 3.20: Wtindows 10 Cltientt demonsttrattting Teredo Intterface Sttatte

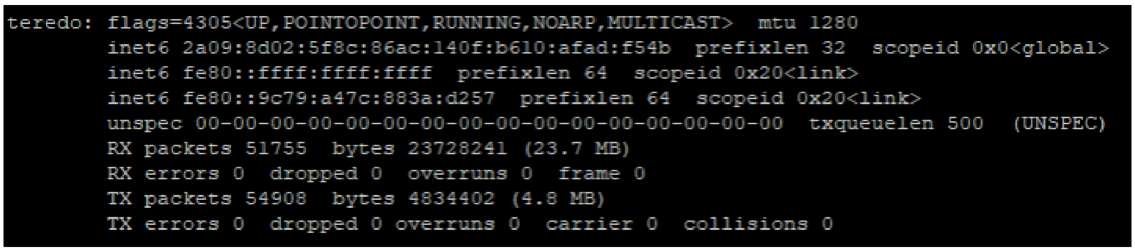
Overall, the configuration demonstrated in this chapter showcases the successful integration of Teredo tunneling technology within IPv4 infrastructure for the end user with a Unix-like OS, providing IPv6 connectivity for the end user and facilitating communication over the IPv6 Internet through the designated Teredo relay server. The Teredo relay server bridges the gap between IPv4 and IPv6, allowing seamless communication between devices behind NAT.

4
Results and Discussion

In this section, we discuss the outcomes of the present project. The primary objective was to develop an architecture that effectively enables end-user clients to have a powerful connection with IPv6 networks without having any IPv6 infrastructure.

4.1 Analysis of Findings

In analyzing our findings, we delved into the configuration and deployment of Miredo on both the Teredo Server and the Teredo Relay, in addition to our end users. A notable outcome was observed upon successful implementation: a new network interface named ”teredo,” as previously defined in Sec.3.3. This interface was automatically allocated a unique IPv6 address, reflecting the successful integration of Teredo tunneling technology into our IPv4 infrastructure.

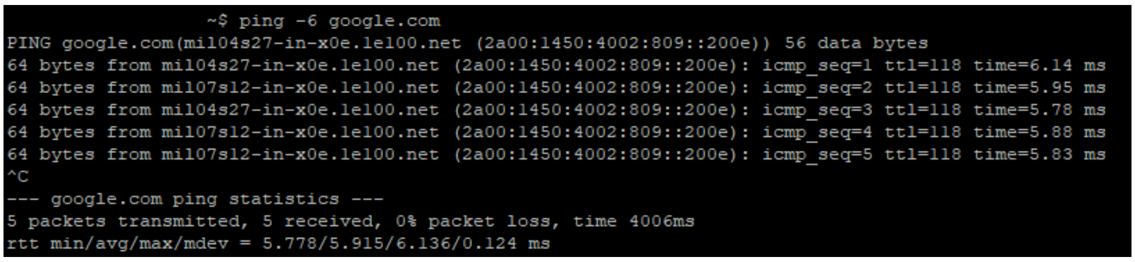


```
teredo: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1280
        inet6 2a09:8d02:5f8c:86ac:140f:b610:afad:f54b  prefixlen 32  scopeid 0x0<global>
        inet6 fe80::ffff:ffff:ffff  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::9c79:a47c:883a:d257  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 51755  bytes 23728241 (23.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 54908  bytes 4834402 (4.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 4.1: Teredo Intterface asstigned tto Cltientt based on tthe conftiguratttions of Teredo Server and Teredo Relay

Introducing the Teredo interface allowed the end user (in this project case, the end-user has a Linux operating system) to establish connectivity to the IPv6 Internet. Upon executing the command `ifconfig`, the dynamically assigned IPv6 address is associated with the ”Teredo” interface, which is observed in Fig. 4.1. This address facilitated communication between our Client and the broader IPv6 network.

To empirically demonstrate the successful connection to the IPv6 Internet, a ping test to Google’s IPv6 address `ping6 google.com` was performed in the end-user’s terminal. The selected target, google.com, serves as a representative example of the broader IPv6 landscape. While the Google ping results are specifically presented here, our Client can ping various other IPv6 networks. Remarkably, the ping requests yielded responsive replies from Google’s IPv6 servers, afirming the effective integration of Teredo for IPv6 connectivity. This result underscores the significance of Teredo as a tunneling mechanism, seamlessly bridging the gap between IPv4 and IPv6 and enabling our Linux client to interact with IPv6-enabled services on the Internet.



```
                  ~$ ping -6 google.com
PING google.com(mil04s27-in-x0e.1e100.net (2a00:1450:4002:809::200e)) 56 data bytes
64 bytes from mil04s27-in-x0e.1e100.net (2a00:1450:4002:809::200e): icmp_seq=1 ttl=118 time=6.14 ms
64 bytes from mil07s12-in-x0e.1e100.net (2a00:1450:4002:809::200e): icmp_seq=2 ttl=118 time=5.95 ms
64 bytes from mil04s27-in-x0e.1e100.net (2a00:1450:4002:809::200e): icmp_seq=3 ttl=118 time=5.78 ms
64 bytes from mil07s12-in-x0e.1e100.net (2a00:1450:4002:809::200e): icmp_seq=4 ttl=118 time=5.88 ms
64 bytes from mil07s12-in-x0e.1e100.net (2a00:1450:4002:809::200e): icmp_seq=5 ttl=118 time=5.83 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 5.778/5.915/6.136/0.124 ms
```

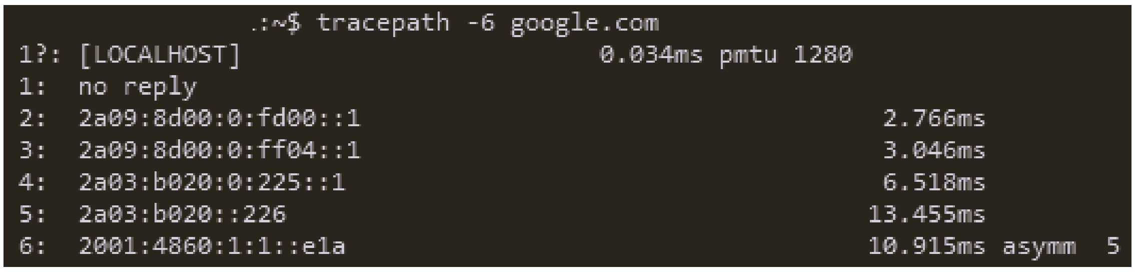
Figure 4.2: User Cltientt Ltinux can pting google.com tinstide tthe IPv6 nettwork

The representation of ping replies, as illustrated in Fig. 4.2, is a tangible confirmation of established connectivity. These findings highlight the practical implications of deploying Teredo tunneling protocol, demonstrating their eficacy in providing IPv6 connectivity for end-user within an IPv4-centric environment.

The ping results demonstrated a commendable level of performance, with an average round-trip time (RTT) of approximately 5.92 milliseconds between the Client and Google’s IPv6 address **2a00:1450:4002:809::200e**. RTT is a metric that quantifies the time a packet travels from the source to the destination and back again. This real-time assessment is crucial for applications requiring prompt data exchange, such as video streaming or online gaming.

Furthermore, the Time-To-Live (TTL) values are consistently registered at 118 across multiple ICMP sequences. TTL represents the maximum number of hops or routers a packet can traverse before being discarded. In the context of our analysis, a TTL of 118 indicates that the packets traversed a moderate number of network hops to reach their destination. It is an essential metric in network diagnostics, helping to identify potential routing issues or loops.

In the figure below, the tracepath results for Google's IPv6 network are showcased, offering a visual representation of the route packets take to reach the destination. By examining the sequence of hops and their corresponding TTL values, insights into the eficiency and stability of the network path are revealed. This depiction enhances the understanding of TTL behavior within IPv6 networks, particularly in the context of promoting IPv6 adoption and leveraging Teredo Tunneling. Through this graphical representation, the intricacies of packet traversal and network performance are elucidated, contributing to a comprehensive analysis within the realm of IPv6 promotion and deployment strategies.



```
              .:~$ tracepath -6 google.com
1?: [LOCALHOST]                        0.034ms pmtu 1280
1:  no reply
2:  2a09:8d00:0:fd00::1                                   2.766ms
3:  2a09:8d00:0:ff04::1                                   3.046ms
4:  2a03:b020:0:225::1                                    6.518ms
5:  2a03:b020::226                                       13.455ms
6:  2001:4860:1:1::e1a                                   10.915ms asymm  5
```

Figure 4.3: Tracepatth Resultts for Google's IPv6 Nettwork, Illusttrattting TTL Behavtior and Nettwork Path Analystis

In summary, the ping results signify robust IPv6 connectivity between the end user and Google, supported by the eficiency of the Teredo tunneling system. The low RTT values and consistent TTL values underscore the reliability and responsiveness of the IPv6 infrastructure, afirming its suitability for diverse online activities.

4.2 Linux vs. Windows Clients

Unix-like OSes and Windows exhibit distinctive networking models and configuration approaches. In the Linux ecosystem, networking is often managed through configuration files and command-line tools, such as `/etc/network/interfaces`, `ifconfig`, and `IP`. The flexibility of Unix-like operating systems allows for extensive customization and control over network settings, making it a preferred choice for users with a penchant for fine-tuning. On the other hand, Windows employs a user-friendly graphical interface accessible through the Control Panel, making network configuration more approachable for users who might not be familiar with command-line operations. Both operating systems adhere to a client-server networking model, but the means of achieving and managing network connectivity differ significantly.

Concerning network security, Linux stands out for its robustness. Tools like iptables and ufw provide granular control over incoming and outgoing trafic, contributing to a secure networking environment. With its built-in Windows Defender Firewall, Windows caters to users who appreciate a more straightforward approach to security management. However, the prevalence of Windows in desktop environments has historically made it a target for a higher number of malware and viruses compared to Unix-like operating systems.

4.2.1 IPv6 Differences

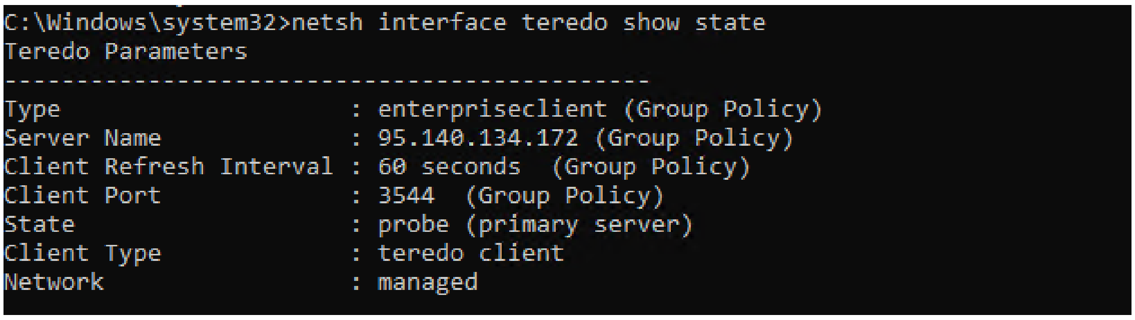
In the realm of IPv6, both Linux and Windows demonstrate their commitment to modern networking standards. Linux distributions, known for their adaptability, offer strong support for IPv6, integrating tools like IP and `sysctl` for IPv6 configuration. It ensures that Linux systems seamlessly transition to the next-generation IP protocol. Similarly, Windows operating systems have embraced IPv6 in several versions, with IPv6 being enabled by default. Windows users can easily configure IPv6 settings through the intuitive network settings interface, making it accessible even to those less versed in networking intricacies.

IPv6 configuration reveals further nuances in the operating system. Linux administrators can fine-tune IPv6 settings using command-line tools, allowing customized address assignment and routing. Conversely, Windows provides a user-friendly interface that simplifies IPv6 configuration, making it accessible to a broader user base. This emphasis on user experience aligns with Windows' broader strategy of catering to a diverse audience with varying levels of technical expertise.

The Teredo mechanism, designed to facilitate IPv6 connectivity in networks utilizing IPv4 Network Address Translation (NAT), illustrates another point of contrast between Linux and Windows. In the Linux landscape, Teredo is not natively supported. However, third-party implementations like Miredo bridge this gap, enabling Linux systems to leverage Teredo tunneling when interfacing with networks relying on this mechanism. On Windows, Teredo is seamlessly integrated and serves as a default feature. It automatically engages when necessary, ensuring IPv6 connectivity for devices situated behind IPv4 NAT devices.

The adoption and usage of Teredo also differ between the two operating systems. While Teredo may find limited application in Linux due to the prevalence of native IPv6 support, Windows systems widely employ it to navigate environments where IPv4 NAT persists. This divergence in approach highlights the adaptability and customization capabilities of Linux, contrasted with Windows' emphasis on seamless integration and user-friendliness in addressing networking challenges.

The comparison between Unix-like operating systems and Windows clients in the context of Teredo tunneling for IPv6 promotion highlights a significant issue with the Teredo state on Windows clients as discussed in Sec. 3.3. While the end-user with the Unix operating system, which in this project is the Linux operating system, successfully receives a reply from the IPv6 internet, the Windows end-user encounters a challenge where the Teredo state remains stuck in probe mode. This state indicates that the Client is actively attempting to establish a Teredo tunnel to the Teredo server. It should transition to a qualified state once the tunnel is successfully established. However, in this case, the Teredo state quickly shifts to ofline after being in probe mode for a short duration.



```
C:\Windows\system32>netsh interface teredo show state
Teredo Parameters
---------------------------------------------
Type                    : enterpriseclient (Group Policy)
Server Name             : 95.140.134.172 (Group Policy)
Client Refresh Interval : 60 seconds  (Group Policy)
Client Port             : 3544  (Group Policy)
State                   : probe (primary server)
Client Type             : teredo client
Network                 : managed
```

Figure 4.4: Wtindows 10 Cltientt demonsttrattting Teredo Intterface Sttatte

To investigate and compare the behavior of the two clients, trafic capture was performed during the initiation of the Teredo setup. The analysis revealed that both clients could connect to the Miredo server initially. However, a notable distinction emerged in the IPv6 address assignment. In the Windows client trafic, there was no evidence of an IPv6 address with the expected prefix "2a09:8d02::," which was the configured prefix for Miredo clients. In contrast, the Linux client successfully received an IPv6 address as configured.

The investigation on the issue of Windows clients using Teredo tunneling with the proposed way of configuration in this master's thesis revealed that the Miredo was historically tested against Windows XP, and certain restrictions were identified, such as a fixed prefix (2001::/32) and subsequent server IPs. However, given the evolving landscape and the absence of recent testing on Windows 10, Rémi suggested consulting Microsoft for more information. Additionally, Rémi mentioned that the Miredo server had not been personally tested on Windows 10 due to the time lapse since his involvement.

In light of the historical context, a pertinent query arises regarding the compatibility of the Miredo server with Windows 10. An avenue worth exploring for potential solutions involves contemplating the utilization of a Windows Server as the Teredo server in the configuration of the Windows client. This strategic approach holds promise for compatibility and effectively resolving the challenges encountered with the Miredo server.

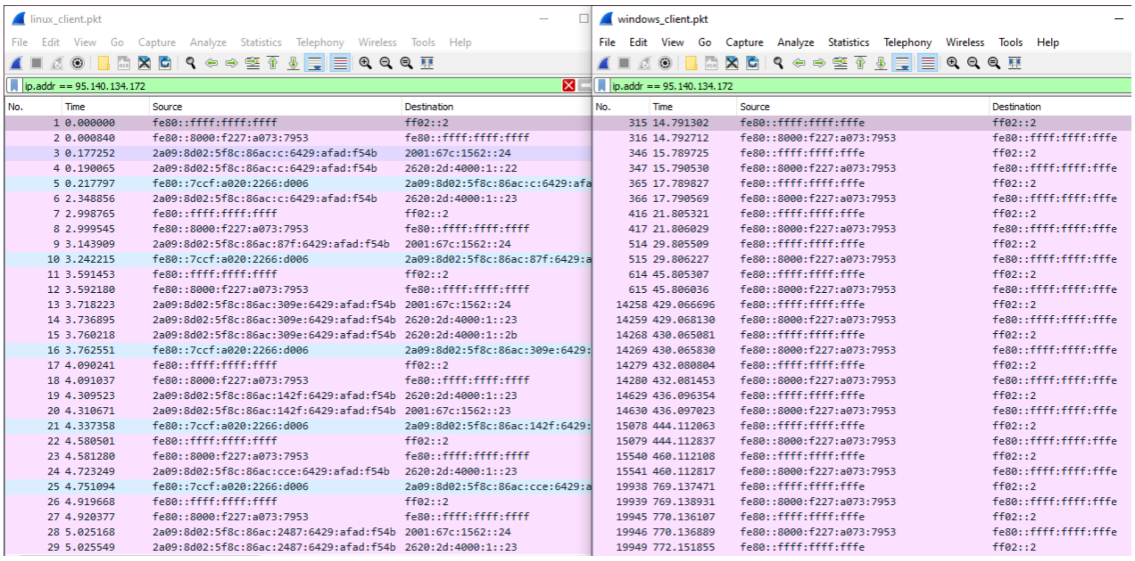
In conclusion, the investigation into the Teredo tunneling issue reveals a discrepancy in the IPv6 address assignment between Linux and Windows clients. The feedback from Rémi underscores the need for exploration into Windows 10 compatibility and suggests considering alternative Teredo server configurations, such as using a Windows Server. Further collaboration with Microsoft and testing different server configurations could potentially lead to a resolution for the Teredo tunneling challenges on Windows clients.

4.3 Risk Management and Mitigation Strategies

Troubleshooting has been done to understand the issues of the Teredo state on the end-user with the Windows operating system side. The trafic capture has been conducted on both end-user systems, which in the present project operating systems are Windows 10 and the Linux, specifically Ubuntu 22.04.2 LTS, during their initiation of the Teredo setup. Subsequently, an analysis was undertaken to discern the disparities between the two. Applying a Wireshark filter (ip.addr == 95.140.134.172, representing the Teredo Server IP) showed that both clients' initial two lines exhibited uniformity. This observation suggests that the Windows Teredo client can initially establish a connection with the Miredo server. However, an incongruity was identified in the IPv6 assignment process.

Within the trafic originating from the Windows client, there was a notable absence of IPv6 addresses bearing the specified prefix "2a09:8d02::." Despite configuring both the Miredo server and Miredo relay to allocate this precise prefix to Miredo clients, the Windows client's IPv6 assignment needed to align with the anticipated configuration. In contrast, the Linux client successfully received the configured IPv6 address, adhering to the expected parameters.

Figure 4.5: The compartison between tthe capttured ttraftic bettween Ltinux and Wtindows Cltientt


linux_client.pkt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
windows_client.pkt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 95.140.134.172
No.  Time      Source                                   Destination
   1 0.000000  fe80::ffff:ffff:ffff                     ff02::2
   2 0.000840  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
   3 0.177252  2a09:8d02:5f8c:86ac:c:6429:afad:f54b     2001:67c:1562::24
   4 0.190065  2a09:8d02:5f8c:86ac:c:6429:afad:f54b     2620:2d:4000:1::22
   5 0.217797  fe80::7ccf:a020:2266:d006                2a09:8d02:5f8c:86ac:c:6429:afa
   6 2.348856  2a09:8d02:5f8c:86ac:c:6429:afad:f54b     2620:2d:4000:1::23
   7 2.998765  fe80::ffff:ffff:ffff                     ff02::2
   8 2.999545  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
   9 3.143909  2a09:8d02:5f8c:86ac:87f:6429:afad:f54b   2001:67c:1562::24
  10 3.242215  fe80::7ccf:a020:2266:d006                2a09:8d02:5f8c:86ac:87f:6429:a
  11 3.591453  fe80::ffff:ffff:ffff                     ff02::2
  12 3.592180  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
  13 3.718223  2a09:8d02:5f8c:86ac:309e:6429:afad:f54b  2001:67c:1562::24
  14 3.736895  2a09:8d02:5f8c:86ac:309e:6429:afad:f54b  2620:2d:4000:1::23
  15 3.760218  2a09:8d02:5f8c:86ac:309e:6429:afad:f54b  2620:2d:4000:1::2b
  16 3.762551  fe80::7ccf:a020:2266:d006                2a09:8d02:5f8c:86ac:309e:6429:
  17 4.090241  fe80::ffff:ffff:ffff                     ff02::2
  18 4.091037  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
  19 4.309523  2a09:8d02:5f8c:86ac:142f:6429:afad:f54b  2620:2d:4000:1::23
  20 4.310671  2a09:8d02:5f8c:86ac:142f:6429:afad:f54b  2001:67c:1562::23
  21 4.337358  fe80::7ccf:a020:2266:d006                2a09:8d02:5f8c:86ac:142f:6429:
  22 4.580501  fe80::ffff:ffff:ffff                     ff02::2
  23 4.581280  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
  24 4.723249  2a09:8d02:5f8c:86ac:cce:6429:afad:f54b   2620:2d:4000:1::23
  25 4.751094  fe80::7ccf:a020:2266:d006                2a09:8d02:5f8c:86ac:cce:6429:a
  26 4.919668  fe80::ffff:ffff:ffff                     ff02::2
  27 4.920377  fe80::8000:f227:a073:7953                fe80::ffff:ffff:ffff
  28 5.025168  2a09:8d02:5f8c:86ac:2487:6429:afad:f54b  2001:67c:1562::24
  29 5.025549  2a09:8d02:5f8c:86ac:2487:6429:afad:f54b  2620:2d:4000:1::23
```

```
ip.addr == 95.140.134.172
No.     Time        Source                     Destination
  315  14.791302  fe80::ffff:ffff:fffe       ff02::2
  316  14.792712  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
  346  15.789725  fe80::ffff:ffff:fffe       ff02::2
  347  15.790530  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
  365  17.789827  fe80::ffff:ffff:fffe       ff02::2
  366  17.790569  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
  416  21.805321  fe80::ffff:ffff:fffe       ff02::2
  417  21.806029  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
  514  29.805509  fe80::ffff:ffff:fffe       ff02::2
  515  29.806227  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
  614  45.805307  fe80::ffff:ffff:fffe       ff02::2
  615  45.806036  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
14258 429.066696  fe80::ffff:ffff:fffe       ff02::2
14259 429.068130  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
14268 430.065081  fe80::ffff:ffff:fffe       ff02::2
14269 430.065830  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
14279 432.080804  fe80::ffff:ffff:fffe       ff02::2
14280 432.081453  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
14629 436.096354  fe80::ffff:ffff:fffe       ff02::2
14630 436.097023  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
15078 444.112063  fe80::ffff:ffff:fffe       ff02::2
15079 444.112837  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
15540 460.112108  fe80::ffff:ffff:fffe       ff02::2
15541 460.112817  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
19938 769.137471  fe80::ffff:ffff:fffe       ff02::2
19939 769.138931  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
19945 770.136107  fe80::ffff:ffff:fffe       ff02::2
19946 770.136889  fe80::8000:f227:a073:7953  fe80::ffff:ffff:fffe
19949 772.151855  fe80::ffff:ffff:fffe       ff02::2
```

The successful implementation of the study on enhancing IPv6 adoption in Linux Clients through Teredo Tunneling within IPv4 infrastructures requires meticulously considering the potential risks and formulating effective mitigation strategies. This section identifies and addresses critical risks associated with technical issues, routing errors, and firewall policies.

4.3.1 Technical Issues in IPv6 Connectivity for Clients

A potential risk involves technical impediments preventing clients from establishing IPv6 network connectivity. Specifically, if clients cannot receive a ping reply from the IPv6 interface of a server, such as google.com, the root of the problem may be attributed to the Miredo server or Miredo relay machines. Network administrators must employ comprehensive troubleshooting and debugging techniques to address this risk.

A mitigation strategy is proposed to restart and reboot the Miredo/Teredo services on the servers. The following command accomplishes this task:

```
sudo systemctl reload-or-restart miredo
```

By executing this command, the Miredo services undergo a reloading or restarting process, potentially resolving connectivity issues and ensuring the smooth functioning of the Teredo tunnel.

4.3.2 Client Trouble Receiving Reply Packets

Another identified risk is clients experiencing dificulties receiving reply packets from the desired host server. This issue can be recognized by capturing trafic causing the `tcpdump` utility on the server. The command below exemplifies this approach:

```
sudo tcpdump -i ens160 host 2a00:1450:4002:405::200e or icmp6 -nn
```

This command captures network trafic associated with a specific IPv6 address, shedding light on potential reasons for reply packets not reaching the Teredo client. The analysis of captured packets can inform corrective actions.

4.3.3 Routing Errors Impacting ICMP Packets

Routing errors may pose a significant risk, particularly if ICMP packets encounter issues in their routing, leading to a failure in forwarding trafic from the Miredo Relay to the Client. In order to investigate and address routing errors, the IPv6 routing table can be examined using the command:

```
route -6
```

The results of this command reveal the routing configuration. If discrepancies are identified, corrective actions can be taken. For instance, removing an incorrect route using the following command can rectify routing errors:

```
ip -6 route del <incorrect_route>
```

The command mentioned above ensures that the correct route, such as the Teredo interface, is prioritized for packet forwarding, mitigating the risk of routing errors impacting ICMP packets.

4.3.4 Firewall Policies Configuration

Potential issues arising from firewall policies may present a risk to the study. If the network administrator restricts the ICMP protocol, clients may face dificulties connecting to the Miredo Relay and Miredo Server. To mitigate this risk, firewall policies permitting the necessary protocols are imperative.

Regular communication and coordination with network administrators help align firewall policies with study requirements. Comprehensive documentation of firewall configurations and ongoing monitoring are essential components of this mitigation strategy.

In conclusion, the risk management and mitigation strategies above are integral to anticipating and addressing potential challenges. By proactively managing and troubleshooting technical issues, routing errors, and firewall policies, we aim to ensure the resilience and success of our study on enhancing IPv6 adoption in end-users of an organization through Teredo Tunneling.

5 Conclusion and Recommendations

5.1 Summary of Findings

The culmination of this comprehensive study and implementation approach to enhance IPv6 adoption in Linux clients within IPv4 infrastructures through Teredo tunneling represents a significant milestone in addressing the challenges imposed by the pervasive limitations of IPv4. This chapter provides an overview of the key findings, the implications of the implemented solution, and future directions for further exploration.

The outcomes of our endeavor to integrate Teredo tunneling technology into the network infrastructure of the client and server sides have been promising. The analysis of our findings began in section 3.2.2 with a meticulous examination of the configuration and deployment of Teredo on the Teredo Server and Teredo Relay and the end-user with the Linux operating system. The successful implementation created a Teredo interface that dynamically assigned a unique IPv6 address. This interface became the conduit for establishing connectivity to the IPv6 Internet, effectively bridging the gap between IPv4 and IPv6.

Our empirical demonstrations included ping tests to Google's IPv6 address, explained in 4.1 section, showcasing responsive replies and afirming the successful integration of IPv6 connectivity through Teredo tunneling on end-user without any infrastructure of the IPv6 network. The ping results indicated commendable performance, with low round-trip times (RTT) and consistent Time-To-Live (TTL) values. The representation of ping replies in 4.2 served as tangible confirmation of established connectivity, highlighting the eficacy of Miredo and Teredo in providing IPv6 connectivity for Linux clients within an IPv4-centric environment.

The examination of IPv6 connectivity between the end-user and servers revealed robust performance. Moreover, eficiency emphasizes the reliability of the implemented Teredo tunneling system. The comparison between Linux and Windows clients elucidated distinctive networking models and configuration approaches, particularly in Teredo tunneling for IPv6 promotion. Windows clients presented challenges, notably with the Teredo state remaining stuck in probe mode, prompting further investigation into Windows 10 and Mac clients' compatibility.

Our achievements include successful IPv6 connectivity for clients with Unix-like operating systems. In this master's thesis project case, the operating system of the end-user is Linux, specifically Ubuntu 22.04.2 LTS, which contributes to advancing network capabilities in an environment constrained by IPv4 infrastructure. In this project case, InfoCamere provided the network environment with IPv4 infrastructure. As exemplified in our ping tests, the accomplishment of effective communication between IPv6 and IPv4 networks underscores the feasibility of Teredo tunneling as a viable solution.

However, challenges surfaced, particularly in the context of Windows clients. The inability to modify default settings and the observed Teredo state issues pose areas for further exploration and potential collaboration with Microsoft. Despite these challenges, the comprehensive documentation of our implementation and the proposed solutions pave the way for ongoing refinement and improvement.

This study's risk management and mitigation strategies provide a proactive approach to addressing potential challenges. Technical issues in IPv6 connectivity for clients are addressed through troubleshooting and restarting Miredo/Teredo services on servers, ensuring the smooth functioning of the Teredo tunnel. The risk of clients facing dificulties receiving reply packets is mitigated by capturing and analyzing trafic, allowing for informed corrective actions. Routing errors impacting ICMP packets are addressed by examining and managing the routing table, ensuring proper routing configurations. Firewall policies pose potential issues, and comprehensive coordination with network administrators is proposed to align policies with study requirements.

The journey from the initial exploration phase to the successful implementation of Teredo tunneling has been challenging and rewarding. The milestones achieved, such as practical IPv6 connectivity for Linux end-users, underscore the dedication and effort invested in this project.

5.2 Conclusion and Future Directions

In conclusion, the successful implementation of Teredo tunneling represents a noteworthy advancement in addressing the challenges associated with IPv4 limitations that organizations face. The study has significantly contributed to the broader landscape of IPv6 adoption within complex network environments. This section provides a comprehensive overview of the achievements, challenges encountered, and risk management strategies and outlines potential future directions.

The project's primary objective was to establish a robust architecture enabling end-users within an organization, which in the present project case was InfoCamere, to connect seamlessly to IPv6 networks despite the prevalent IPv4 infrastructure. The adoption of Teredo tunneling emerged as a strategic choice, encapsulating IPv6 packets within IPv4 packets and facilitating uninterrupted communication between devices on IPv6 and IPv4 networks with the help of an ISP or an Internet Exchange point, which in the present project case was VSIX.

The successful implementation yielded tangible results, particularly in Linux clients' ability to ping IPv6 addresses and receive responsive replies, bridging the gap between IPv4-centric environments and the evolving IPv6 landscape. The introduction of the Teredo interface, dynamic IPv6 address allocation, and the subsequent demonstration of connectivity to prominent IPv6 destinations such as Google underscores the eficacy of the Teredo solution.

However, our journey encountered challenges when extending the Teredo tunneling approach to Windows 10 and Mac clients. Notably, the default settings on Windows 10 posed restrictions, limiting our ability to tailor the configuration to our specific requirements. Additionally, unforeseen issues arose in the context of Mac operating systems, introducing complexities our existing solution needed to address seamlessly.

These challenges shed light on the intricate nature of achieving interoperability across diverse operating systems. It is essential to candidly acknowledge these limitations and recognize that a uniform solution may require tailored approaches for different platforms.

The risk management strategies implemented for technical issues, routing errors, and firewall policies effectively addressed potential roadblocks during the project. However, the unique characteristics of Windows 10 and Mac clients demand a nuanced and adaptive risk management approach. Collaborative efforts with Microsoft for Windows 10 compatibility and dedicated testing on Mac clients are critical components of our ongoing commitment to refining and expanding the reach of the Teredo tunneling solution.

5.3 Recommendations for Future Research and Implementations

Looking ahead, the research and development trajectory in network protocols calls for a nuanced focus on refining the Teredo tunneling solution. The identified challenges within the Windows 10 environment present opportunities for targeted enhancements. A pivotal avenue for progress involves close collaboration with key stakeholders, notably Microsoft, and the initiation of rigorous testing across diverse platforms. This cooperative effort is essential in pursuing a more inclusive and universally applicable solution.

Beyond the immediate challenges lies the imperative for continuous exploration and innovation, given the dynamic nature of network protocols and connectivity. As we delve into the complexities of IPv4 and IPv6 coexistence, the insights gleaned from this project serve as a cornerstone for future advancements. The experiences and lessons learned on this journey pave the way for a more comprehensive and effective solution. The commitment to progress, marked by exploration and overcoming challenges, sets the stage for seamless communication between diverse IP versions in the diverse landscapes of evolving networks.

Besides enhancing current systems, prospective research initiatives may encompass developing and deploying an innovative tunneling system. Inspired by Teredo tunneling, this novel technique attempts to close the IPv4–IPv6 divide. Investigating novel tunneling systems has the potential to meet the changing requirements of network infrastructures and provide durability and adaptability in the face of shifting technological environments. This line of inquiry advances the continuous development of network protocols and builds a solid basis for future networked communication in various IP contexts.

References

[1] J. Doshi, R. Chaoua, S. Kumar, and S. Mallya, "A comparative study of ipv4/ipv6 co-existence technologies," 5 2012.

[2] A. N. A. Ali, "Comparison study between ipv4 & ipv6," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, p. 314, 5 2012.

[3] P. Loshin, "Ipv6: Theory, protocol, and practice," 2004.

[4] D. Plonka and A. Berger, "Temporal and spatial classification of active ipv6 addresses," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 509–522.

[5] M. Rostański and T. Mushynskyy, "Security issues of ipv6 network autoconfiguration," in *IFIP International Conference on Information Systems and Industrial Management (CISIM)*. Krakow, Poland: Springer, 9 2013, pp. 218–228.

[6] R. Graziani, *IPv6 fundamentals: a straightforward approach to understanding IPv6*. Pearson Education, 2013.

[7] J. Czyz, K. Lady, S. G. Miller, M. Bailey, M. Kallitsis, and M. Karir, "Understanding IPv6 internet background radiation," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, 10 2013, pp. 105–118.

[8] M. Bakni and S. Hanbo, "A survey on internet protocol version 4 (ipv4)," *WikiJournal of Science*, vol. 5, 2022, ffhal-03914308f.

[9] M. A. Naagas and A. P. Gamilla, "Denial of service attack: An analysis to ipv6 extension headers security nightmares," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 2922–2930, 2022.

[10] M. Gregr, P. Matoušek, M. Sveda, and T. Podermanski, "Practical ipv6 monitoring-challenges and techniques," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. IEEE, 2011, pp. 650–653.

[11] V. Jain, D. Tiwari, S. Singh, and S. Sharma, “Impact of ipv4, ipv6 and dual stack interface over wireless networks,” *International Journal of Computer Network and Information Security*, vol. 10, no. 4, pp. 65–71, 2018.

[12] A. Al-hamadani and G. Lencse, “Survey on the performance analysis of ipv6 transition technologies,” *Acta Technica Jaurinensis*, vol. 14, no. 2, pp. 186–211, 2021.

[13] R. W. Kerbs, “Internet gaming in the era of ipv6,” *The Electronic Library*, vol. 22, no. 1, pp. 16–22, 2004.

[14] P. Grayeli, S. Sarkani, and T. A. Mazzuchi, “Performance analysis of ipv6 transition mechanisms over mpls,” *International Journal of Communication Networks and Information Security (Ijcnis)*, vol. 4, no. 2, p. 91, 8 2012.

[15] S. Ashraf, D. Muhammad, and Z. Aslam, “Analyzing challenging aspects of ipv6 over ipv4,” *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, vol. 6, no. 1, p. 54, 2020.

[16] M. M. Alhassoun and S. R. Alghunaim, “A survey of ipv6 deployment,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 9, pp. 1–5, 2016.

[17] D. R. Al-Ani and A. R. Al-Ani, “The performance of ipv4 and ipv6 in terms of routing protocols using gns 3 simulator,” *Procedia Computer Science*, vol. 130, pp. 1051–1056, 2018.

[18] S.-J. Yoon, J.-T. Park, D.-I. Choi, and H. K. Kahng, “Performance comparison of 6to4, 6rd, and isatap tunnelling methods on real testbeds.” *International Journal on Internet & Distributed Computing Systems*, vol. 2, no. 2, 2012.

[19] C. Zacker, *Installing and Configuring Windows Server 2012 R2: Exam 70-410*. Wiley, 2014.

[20] T. Gleeson, D. Thaler, and F. Templin, “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP),” RFC 5214, Mar. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5214

[21] M. K. Muchamad and M. T. Bingamawa, "A review on network performance evaluation of 6to4 tunneling," *Jurnal Komputer, Informasi Teknologi, dan Elektro*, vol. 6, no. 3, 2021.

[22] S. Zander, L. L. H. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Investigating the ipv6 teredo tunnelling capability and performance of internet clients," *Acm Sigcomm Computer Communication Review*, vol. 42, no. 5, pp. 13–20, 2012.

[23] D. V. B. Tien, M. Komu, M. Siekkinen, and A. Ylä-Jääski, "H-box: Interconnecting devices across local networks," pp. 244–252, 2014.

[24] S. M. Huang, Q. Wu, Y. Lin, and C. H. Yeh, "Sip mobility and ipv4/ipv6 dual-stack supports in 3g ip multimedia subsystem," *Wireless Communications and Mobile Computing*, vol. 6, no. 5, pp. 585–599, 2006.

[25] M. Elich, P. Velan, T. Jirsík, and P. Čeleda, "An investigation into teredo and 6to4 transition mechanisms: Trafic analysis," pp. 1018–1024, 2013.

[26] H. Babiker, I. Nikolova, and K. K. Chittimaneni, "Deploying ipv6 in the google enterprise network: Lessons learned (practice & experience report)," in *25th Large Installation System Administration Conference (LISA 11)*, vol. 10, 2011.

[27] A. Durand, j. woodyatt, R. Droms, and Y. L. Lee, "Dual-stack lite broadband deployments following ipv4 exhaustion," 2011.

[28] O. D'yab, "A comprehensive survey on the most important ipv4aas ipv6 transition technologies, their implementations and performance analysis," *Infocommunications Journal*, vol. 14, no. 3, pp. 35–44, 2022.

[29] F. Brockners, S. Speicher, D. Ward, and S. Gundavelli, "Gateway-initiated dual-stack lite deployment," 2012.

[30] A.-J. Su, D. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind akamai: Inferring network conditions based on cdn redirections," *Ieee/Acm Transactions on Networking*, vol. 17, no. 6, pp. 1752–1765, 2009.

[31] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice, 2010.

[32] H. Hours, E. W. Biersack, P. Loiseau, A. Finamore, and M. Mellia, "A study of the impact of dns resolvers on cdn performance using a causal approach," *Computer Networks*, vol. 109, pp. 200–210, 2016.

[33] E. Karpilovsky, A. Gerber, D. Pei, J. Rexford, and A. Shaikh, *Quantifying the Extent of IPv6 Deployment*, 2009.

[34] J. Sansa-Otim and A. Mile, *IPv4 to IPv6 Transition Strategies for Enterprise Networks in Developing Countries*, 2013.

[35] B. R. Dawadi, D. B. Rawat, S. R. Joshi, P. Manzoni, and M. Keitsch, "Migration cost optimization for service provider legacy network migration to software-defined ipv6 network," *INTERNAtional Journal of Network Management*, vol. 31, no. 4, p. e2145, 2021.

[36] V. Bajpai and J. Schönwälder, "Understanding the impact of network infrastructure changes using large-scale measurement platforms," 2016.

[37] F. Li, X. Wang, T. Pan, and J. Yang, "A case study of ipv6 network performance: Packet delay, loss, and reordering," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[38] J. Pickard and J. B. Southworth, "The state of ipv6: Measuring global adoption," 2016.

[39] Y. Vanaubel, P. Mérindol, J. Pansiot, and B. Donnet, *A Brief History of MPLS Usage in IPv6*, 2016.

[40] A. S. M. Al-rawahna, C.-W. Hung, and S. Chen, "Readiness of government organizations for cloud-computing age: An empirical evidence from jordan," *Journal of Business and Management Sciences*, vol. 6, no. 4, pp. 152–162, 2018.

[41] E. Jankiewicz, D. Green, and M. E. Fiuczynski, "Ipv6 translation for ipv4 embedded systems," pp. 1–4, 2006.

[42] S. Heroux and A. Fortin, "Exploring it dependency and it governance: A canadian survey," in *CAAA Annual Conference*, 2012.

[43] M. Mackay, C. Edwards, M. Dunmore, T. Chown, and G. S. d. Carvalho, "A scenario-based review of ipv6 transition tools," *IEEE Internet Computing*, vol. 7, no. 3, pp. 27–35, 2003.

Acknowledgments

I extend my sincere gratitude to Professor Andrea Zanella, my academic supervisor, for his invaluable guidance and unwavering support throughout the course of my Master's thesis. His expertise and encouragement have been instrumental in shaping the success of this research endeavor.

I would also like to express my appreciation to the VSIX company and InfoCamere for their significant contributions to this project. Dr. Luca Fintto and Alessio Celin from VSIX deserve special mention for their exceptional encouragement and support, which played a pivotal role in overcoming challenges and achieving milestones.

At InfoCamere, I am thankful to Enrico Perini and Enrico Ferro for their dedicated assistance and collaborative spirit. Their expertise and insights have enriched the project, and I am truly grateful for their contributions.

Beyond the academic and professional realm, my heartfelt thanks go to my family—Alireza Bahari, Fatemeh Fallahi, and Sevin Bahari—for their unwavering encouragement and support. Their belief in my abilities has been a constant source of inspiration.

I would also like to acknowledge the steadfast presence of my boyfriend, Vafa Solati, throughout this challenging process. His unwavering support, understanding, and encouragement have been my pillars of strength.

In conclusion, I am deeply grateful to everyone who has played a role, big or small, in the realization of this thesis. Your support has been invaluable, and I am truly fortunate to have had such a dedicated and inspiring network of individuals around me.