International Journal of Cloud Applications and Computing
Volume 13 • Issue 1

Comparative Evaluation of Host-Based Translator Mechanisms for IPv4-IPv6 Communication Performance Analysis With Different Routing Protocols

Ala Hamarsheh, Computer Systems Engineering, Faculty of Engineering, Arab American University, Jenin, Palestine
https://orcid.org/0000-0002-4736-7331

Ahmad Alqeerm, Computer Science Department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

Iman Akour, Information Systems Department, College of Computing and Informatics, University of Sharjah, UAE

Mohammad Alauthman, Department of Information Security, Faculty of Information Technology, University of Petra, Amman, Jordan

Amjad Aldweesh, College of Computing and IT, Shaqra University, Shaqra, Saudi Arabia*
https://orcid.org/0000-0003-0319-1968

Ali Mohd Ali, Communications and Computer Engineering Department, Faculty of Engineering, Al-Ahliyya Amman University, Jordan

Ammar Almomani, School of Computing, Skyline University College, University City of Sharjah, UAE

Someah Alangari, College of Science and Humanities Dawadmi, Shaqra University, Saudi Arabia
https://orcid.org/0000-0003-1308-1762

ABSTRACT

The impending exhaustion of internet protocol (IP) version four (IPv4) addresses necessitates a transition to the more expansive IP version six (IPv6) protocol. However, this shift faces challenges due to the widespread legacy of IPv4 infrastructure and resistance among organizations to overhaul networks. Host-based translators offer a critical bridging solution by enabling IPv6-only devices to communicate with IPv4-only devices through software-level protocol translation. This paper comprehensively evaluates four pivotal host-based translator mechanisms—bump-in-the-stack (BIS), bump-in-the-application programming interface (API) (BIA), BIA version 2 (BIAv2), and bump-in-the-host (BIH). Using simulated networks with diverse configurations of IPv4/IPv6 applications, hosts, and routing protocols, the authors assessed performance through metrics including packet loss, convergence time, traffic throughput, and overhead. The results reveal variability in effectiveness across both translators and scenarios. BIAv2 demonstrated advantages in throughput and overhead due to stateless mapping. The research underscores the importance of selecting the optimal translation approach for specific network environments and goals. It guides smoother IPv6 adoption by demonstrating how host-based translators can facilitate coexistence during transition. Further exploration of performance tradeoffs can continue guiding effective deployment strategies.

KEYWORDS

Host-Based Translators, Ipv6, Ipv6 Transition Mechanisms, Performance Analysis, Socket API Mapping

DOI: 10.4018/IJCAC.332765

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

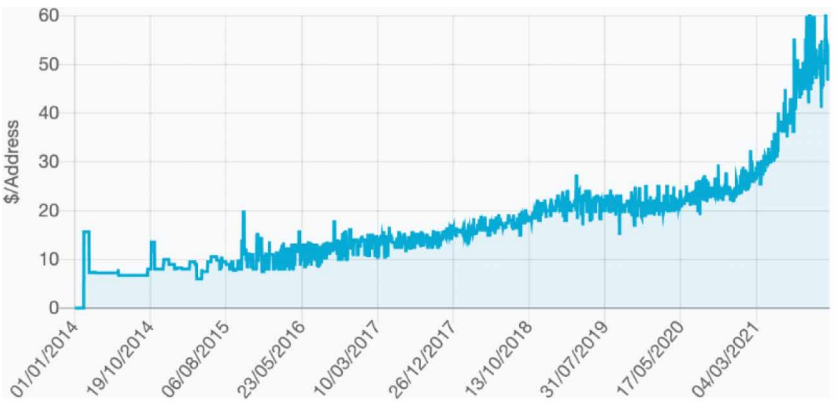
INTRODUCTION

For more than 30 years, the internet protocol (IP) version four (IPv4) has been the cornerstone of the internet. However, the limited number of accessible IPv4 addresses has become a serious concern as more devices connect to the internet. This issue created the IP version six (IPv6) protocol, which provides a significantly bigger address space. Despite its benefits, IPv6 adoption has been delayed due to several problems. These include the need for IPv4 compatibility, the high expense of updating existing infrastructure, and the lack of IPv6 functionality in specific network devices. The internet's fast expansion and the rising number of connected devices have depleted IPv4 addresses. Each device needs a unique IP address to interact with other devices, but the present IPv4 address space can only allow around 4.3 billion distinct addresses, which is inadequate to meet the increasing demand. Furthermore, some corporations have been collecting IPv4 addresses, exacerbating the scarcity. Figure 1 shows the current status of available IPv4 addresses (Hamarsheh & Eleyat, 2018; Hamarsheh & Goossens, 2014; Hamarsheh et al., 2012).

It was first expected that everyone participating in the internet would happily shift to IPv6. However, this assumption proved fairly foolish. It is now commonly acknowledged that human and business considerations were undervalued, hindering a spontaneous move to IPv6. Two key stakeholders are involved in the transition: network providers and end customers. IPv6 primarily helps network providers, whereas end users may gain only indirectly from enhanced network functioning. As a result, it is doubtful that most end users will be strongly motivated to migrate to IPv6. While network providers may profit, their desire to move still depends on their end users. This results in a deadlock: commercial network providers are unlikely to compel users to move against their will. As a result, the key to a successful IPv6 transition is to make it smooth and invisible to end users. Most end users are unconcerned with the network layer and do not care if their programs utilize IPv4 or IPv6. Although end users may be unmotivated to transfer, most would not object to the shift as long as they could continue utilizing their existing apps (Hamarsheh, 2019; Hamarsheh et al., 2011a).

Changing from IPv4 to IPv6 on the IP layer may first appear to not affect programs. However, this is not the case. Applications need IP addresses to communicate; therefore, when using IPv6, they must be able to handle longer IPv6 addresses. It is unrealistic to anticipate that all apps will be upgraded to be IPv6 compliant. While many common internet apps, such as web browsers and email clients, now support IPv6, hundreds of others may not be ready for the switch. Since many of these programs are created by tiny businesses or even one person, IPv6 compatibility upgrades may not always be a top concern, particularly for programs that solely utilize the internet to perform operations like registering or checking for updates. For this reason, it is likely that many apps will

Figure 1. The current state of available IPv4 addresses



not be changed to be IPv6 compatible until IPv6 use is more commonplace. However, it cannot be expected that all end users would upgrade their software following new versions of these apps with IPv6 functionality. Not all users may be able or willing to update software since it might be difficult and time-consuming. To lessen the effect on end users' experiences and promote a more seamless adoption of the new protocol, it is crucial to make the switch to IPv6 as transparent as possible (Hamarsheh & AbdAlaziz, 2019).

It is unrealistic to anticipate that all apps will be upgraded to be IPv6 compatible, particularly because many small businesses and custom applications may not prioritize this update. End customers can be open to the switch to IPv6 as long as their existing apps continue to function. For IPv4-only programs to continue interacting as before, standard provisions must be installed and activated on any general-purpose system capable of connecting via IPv6. This strategy is more practical than updating hundreds of apps to be IPv6 compliant. Even if it is less critical, it is crucial to ensure that IPv6-only applications may interact on devices with dual IPv4/IPv6 connection or IPv4-only connectivity and distant sites with IPv4-only access. Due to the small number of computers having IPv6 connections, minimal effort is being put into creating IPv6-compatible apps. This feature, known as bump-in-the-application programming interface (API) (BIA), is necessary to incentivize the development of these applications. Despite the pressing need for IPv6 adoption, the protocol's adoption has been sluggish, in part due to the difficulty of integrating with the IPv4 infrastructure already in. Switching over to IPv6 for all hardware and network infrastructure is impossible. Therefore, a method of progressive migration that encourages the coexistence of IPv4 and IPv6 is required. The coexistence of IPv4 and IPv6 protocols may be supported through IPv6 transition methods. These techniques provide IPv4 and IPv6 compatibility for devices and networks, enabling a smooth transition to IPv6. Dual stack, tunnelling, and translation are the primary IPv6 transitional techniques. Devices and networks may handle IPv4 and IPv6 concurrently thanks to the dual-stack technology. Dual-stack devices can connect with IPv4 and IPv6 since they have IPv4 and IPv6 addresses. Dual-stack networks may route and forward traffic using both IPv4 and IPv6. Organizations that have previously installed IPv4 and want to move progressively to IPv6 may use this mechanism (Abdalaziz & Hamarsheh, 2020).

Another system that enables the coexistence of IPv4 and IPv6 is tunnelling. IPv6 packets are enclosed in IPv4 packets and sent through an IPv4 network using tunnelling. The IPv6 packets are subsequently decapsulated at the receiving end to enable communication with IPv6 devices. An organization might utilize this technique to connect with IPv6 devices via an IPv4 network but has not yet switched to IPv6. A dual-stack method enables devices to utilize IPv4 and IPv6 protocols concurrently. When both IPv4 and IPv6 networks are accessible, this strategy works well and enables a smooth switch between the two protocols. Devices may connect using either IPv4 or IPv6, thanks to dual stack.

The third IPv6 transition method is translation. Using this approach, IPv6 packets are converted into IPv4 packets and vice versa. Network layers such as the network, transport, and application layers can support translation. The network's edge or the network itself may be used for translation. This technique is appropriate for businesses that can connect with IPv6 devices across an IPv4 network but cannot update all devices and network equipment to IPv6 (Hamarsheh & Goossens, 2012).

We will concentrate on the translation method in this study, particularly the bump-in-the-stack (BIS) (Tsuchiya et al., 2000), BIA (Lee et al., 2002), BIA version 2 (BIAv2) (Hamarsheh et al., 2011b), and bump-in-the-host (BIH) (Huang et al., 2012) approaches. Many IPv6-to-IPv4 translation methods take place at various network stack levels.

At the network layer, IPv6-to-IPv4 translation of the BIS kind takes place. A middle device is translated into BIS. This component converts IPv6 packets into IPv4 packets and vice versa, inserting itself between the source and destination networks. BIS is especially helpful when an IPv6 connection is unavailable since it enables devices to interact with IPv6-only devices utilizing their current IPv4 infrastructure.

At the application layer, an IPv6-to-IPv4 translation called BIA takes place. The BIA device handles the translation. This component converts IPv6 packets into IPv4 packets and vice versa and is put between the host's application and network stack. BIA is beneficial when an IPv6 connection is unavailable since it enables devices to interact with IPv6-only devices utilizing their current IPv4 infrastructure.

Applications may utilize numerous network interfaces on a host thanks to BIAv2 without modifying their source code. No matter how many or what kind of network interfaces are present on the host, the framework consists of a set of rules and an API that offers a single interface for communication. The BIAv2 API, BIAv2 module, and BIAv2 control plane are the three primary parts of the BIAv2 architecture. The framework employs signalling messages to transmit rule information between hosts and a set of rules to decide which network interface to use for outgoing packets. BIAv2 is also included in the document's security issues, such as the risk of denial-of-service attacks and the need for secure host-to-host signalling.

At the host level, an IPv6-to-IPv4 translation called BIH takes place. In BIH, a third party BIH device does the translation. This component converts IPv6 packets into IPv4 packets and vice versa. It is a part of the host operating system. As it enables devices to connect with IPv6-only devices using their current IPv4 infrastructure, BIH is especially helpful when there is little control over the network infrastructure.

The impending exhaustion of IPv4 addresses necessitates transitioning to the more expansive addressing scheme offered by IPv6. However, this shift is hampered by the entrenched IPv4 infrastructure and organizational inertia. Host-based translators have emerged as a critical stopgap solution, allowing IPv6-only devices to communicate with the remaining IPv4-only devices during this migration period. However, the adoption of host-based translators also introduces possible performance concerns and complexities that must be weighed carefully. Motivated by the need for more comprehensive guidance to inform the selection and deployment of host-based translation solutions, this paper undertook an extensive performance evaluation of prominent mechanisms under diverse network conditions. The goal was to benchmark key performance differentiators to empower administrators to choose the optimal approach tailored to their specific transition needs and environment. The following sections detail the study methodology, results, and implications of this rigorous comparative analysis of host-based IPv4-IPv6 translators.

This study makes several notable contributions to knowledge on host-based translator performance for IPv4-IPv6 communication by:

- Providing the first comprehensive comparative analysis of performance across four widely-used translator mechanisms (BIS, BIA, BIAv2, BIH) under diverse network configurations.
- Benchmarking critical performance differentiators such as packet loss, convergence time, traffic overhead, and throughput through exhaustive simulation-based evaluation under varied scenarios.
- Demonstrating the advantages of BIAv2 in terms of lower packet loss, faster convergence, and higher throughput attributed to its efficient stateless mapping methodology.
- Revealing variability in performance based on specific translator mechanisms and network conditions, underscoring the need for solutions tailored to the environment.
- Offering guidance to network administrators and organizations in selecting optimal host-based translators aligned with their transition needs and infrastructure.
- Establishing a framework and baseline results to inform future research directions in this domain, such as integration with network monitoring tools and AI-based optimization.

By addressing the lack of IPv4 addresses available, IPv6 transition approaches play a critical role in assuring the continuous expansion and development of the internet. We will assess these techniques' effectiveness using network measures, including latency, throughput, and packet loss. Since they permit the coexistence of both protocols and provide communication between sites on

various networks, IPv6 transition mechanisms are crucial for the effective transition from IPv4 to IPv6. Network managers may choose the appropriate strategy for the unique needs of their network by being aware of the benefits and limits of each mechanism.

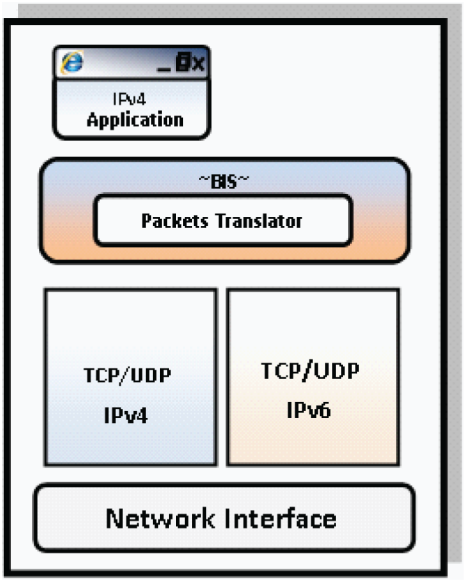
HOST-BASED TRANSLATORS

This section will delve deeper into each of the mechanisms discussed previously, providing a more comprehensive understanding of their functions and details.

BIS

The BIS is a network layer IPv6 transition mechanism per the Open Systems Interconnection model. Implementing BIS allows for concurrently supporting IPv4 and IPv6 protocols by devices without requiring substantial modifications to the underlying network infrastructure. This methodology involves utilizing a device equipped with dual IP stacks, wherein one stack is designated for IPv4 and the other for IPv6. The BIS mechanism functions within the operating system's kernel and intercepts the egressing IPv4 packets, subsequently encapsulating them with IPv6 headers to enable transmission over an IPv6 network. Comparably, the interception of incoming IPv6 packets takes place, whereby the IPv6 headers are extracted and the initial IPv4 packets are subsequently conveyed to the IPv4 stack. The operational capabilities of BIS resemble those of dual stack, albeit with the added benefit of obviating the requirement for distinct IPv4 and IPv6 addresses for every interface. The BIS system facilitates the transmission and reception of IPv4 and IPv6 packets through a singular interface. This mechanism proves advantageous in network environments where a considerable proportion of outdated devices and applications continue to operate on the IPv4 protocol, and a comprehensive migration to IPv6 is not a viable option. Implementing BIS facilitates the simultaneous operation of IPv4 and IPv6 protocols within a given network. The structure of the host based on BIS is illustrated in Figure 2.

Figure 2. The structure of BIS-based host



BIS confers various benefits to network operators, including:

1. Implementing the Border Gateway Protocol (BGP) for IPv6 is comparatively uncomplicated in contrast to alternative approaches for transitioning to IPv6. The implementation of this solution necessitates only slight modifications to the current network infrastructure and equipment.
2. Adopting BIS is a cost-effective alternative for network operators who cannot wholly upgrade to IPv6, as it does not necessitate a substantial overhaul of the network infrastructure.
3. The implementation of BIS results in a reduction of network management complexity as it facilitates the operation of both IPv4 and IPv6 protocols through a singular interface.
4. The compatibility aspect of BIS pertains to its ability to operate alongside IPv4 devices and applications without any disruption, thereby facilitating the coexistence of IPv6 devices and applications.

BIS exhibits certain limitations, including:

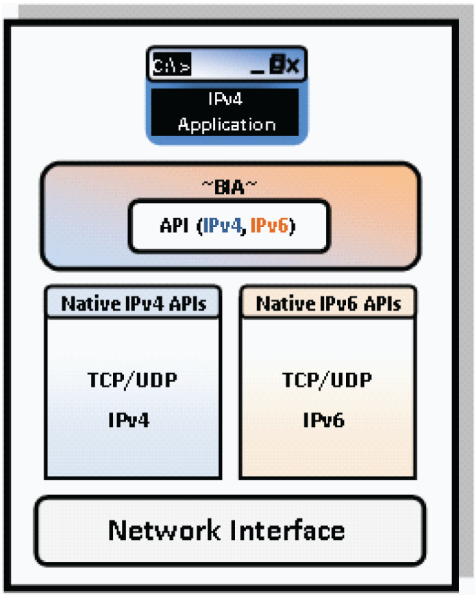
1. The performance of the network may be affected by BIS due to the supplementary overhead incurred by the process of encapsulating and decapsulating IPv4 packets within IPv6 headers.
2. Implementing BIS may introduce security vulnerabilities due to its involvement in intercepting and modifying packets within the operating system's kernel.
3. The scalability of BIS may be limited when implemented in extensive networks that involve a considerable number of legacy devices and applications.

The mechanism known as BIS is efficacious in facilitating the transition to IPv6, as it allows for the concurrent support of IPv4 and IPv6 protocols on devices. Adopting BIS presents a cost-effective alternative for network operators who cannot undertake a comprehensive upgrade to IPv6, as it necessitates only minimal modifications to the existing network infrastructure and equipment. The network performance may be affected by BIS due to the supplementary overhead incurred by the encapsulation and decapsulation of IPv4 packets in IPv6 headers. Before integrating BIS into their network infrastructure, network operators must thoroughly evaluate the benefits and drawbacks of this technology.

BIA

BIA is a mechanism for transitioning to IPv6 that functions at the application layer, offering a seamless approach to facilitating communication between networks that use IPv4 and those that use IPv6. The BIA mechanism operates through the interception of calls at the application layer, facilitating the translation of said calls between the IPv4 and IPv6 protocols. Notably, this process does not necessitate any modifications to the network or transport layers that underlie the system. The BIA mechanism facilitates communication between applications by establishing a virtual interface at the application layer that supports IPv4 and IPv6 protocols. The structure of the host based on BIA is depicted in Figure 2. Upon receiving a communication request from an application, the BIA layer intervenes and ascertains the target host's IP version (IPv4 or IPv6). If the target host utilizes IPv6, BIA will encapsulate the solicitation within an IPv6 packet and transmit it to the intended recipient. If the target host utilizes IPv4, BIA will envelop the solicitation within an IPv4 packet and transmit it to the intended recipient. BIA offers a transparent mechanism to facilitate communication between IPv4 and IPv6 networks, which is considered advantageous. This implies that there is no requirement for altering or adjusting applications to facilitate IPv6 and they can establish communication with other hosts utilizing IPv4 or IPv6 addresses. Moreover, the employment of BIA facilitates the interaction between IPv4-limited legacy applications and IPv6-based networks. The structure of the host based on BIA is illustrated in Figure 3.

Figure 3. The structure of BIA-based host



Another benefit of utilizing BIA is that its implementation does not necessitate modifications to the fundamental network infrastructure or transport layers. This renders BIA a comparatively uncomplicated and economical approach for facilitating communication between IPv4 and IPv6 networks. Nevertheless, the use of BIA is not without its constraints. A primary constraint is the applicability of the technology solely to applications compatible with IP-based protocols.

Applications that utilize non-IP-based protocols may be unable to communicate across IPv4 and IPv6 networks via BIA. Furthermore, BIA may contribute some extra delay and cost, especially in high-speed networks. To utilize BIA, apps must be configured to communicate over the BIA layer. This may be accomplished by altering the application code or using BIA-supporting libraries or middleware. The network must also be configured to route data between IPv4 and IPv6 networks. Overall, BIA offers a framework for facilitating application-layer communication across IPv4 and IPv6 networks without needing modifications to the underlying network or transport layers. While BIA has certain limitations regarding protocol compatibility and performance, it may be a viable option in some cases for permitting communication across IPv4 and IPv6 networks. Figure 4 depicts the BIA-treated situation.

BIAv2

The BIAv2 technology enables applications to utilize numerous network interfaces on a given host without modifying the application's underlying code. The BIAv2 framework comprises a collection of regulations and an API that furnishes a unified communication interface, irrespective of quality

Figure 4. BIA-based host scenario

BIA-based Host			Peer
Application Type	Connectivity	Network	Connectivity
IPv4	IPv4/IPv6	←IPv6→	IPv6

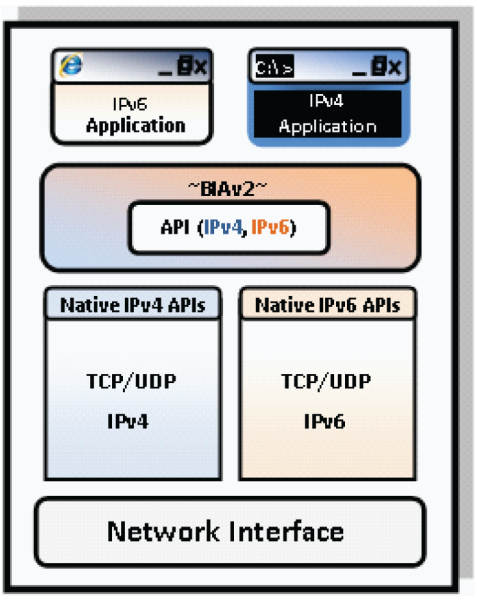
or category of the host's network interfaces. The utilization of the BIAv2 framework can enhance performance, scalability, and security by enabling applications to exploit numerous network interfaces without any modifications to the application code. The BIAv2 framework comprises three primary constituents: the API, module, and control plane. The BIA API serves as a mediator between the application and the BIA module, enabling the application to utilize a unified API for communication, irrespective of the host's network interfaces' quantity or category. The BIAv2 module is tasked with intercepting egress packets from the application and subsequently effecting any necessary modifications to guarantee their transmission via the suitable network interface. The module retains a predetermined set of regulations that ascertain the appropriate interface to employ contingent on the packet's destination address. The host-based architecture of BIAv2 is illustrated in Figure 5.

The BIAv2 control plane manages the rules used by the BIA module. The BIAv2 framework uses a set of rules to determine which network interface to use for outgoing packets. Configuring the rules used by the BIAv2 module involves three steps: rule discovery, rule selection, and rule installation. Rule discovery involves discovering the available network interfaces and their associated addresses. Figure 6 shows all types of applications running on BIAv2-based hosts with all possible types of network connectivity. Rule selection involves selecting the appropriate rule for a given packet based

Figure 5. BIAv2-based host scenarios

BIAv2-based Host			Peers
Application Type	Connectivity	Network	Connectivity
IPv4	IPv6	←IPv6→	IPv6
IPv4	IPv6	←IPv6→	IPv4/IPv6
IPv6	IPv4	←IPv4→	IPv4
IPv6	IPv4	←IPv4→	IPv4/IPv6
IPv4	IPv4/IPv6	←IPv6→	IPv6
IPv6	IPv4/IPv6	←IPv4→	IPv4

Figure 6. The structure of the BIAv2-based host



on the destination address and other factors such as bandwidth or latency. Rule installation involves installing the selected rule in the BIA module so that it can be used to modify outgoing packets.

BIAv2 uses signalling messages to exchange rule information between hosts. The signalling messages can be sent using a separate control channel, such as the BGP or OSPF protocols, or through a dedicated signalling protocol, such as the BIA Signalling Protocol. The signalling process involves three steps: discovery, negotiation, and synchronization. Discovery involves discovering the available BIAv2-capable hosts and their associated network interfaces. Negotiation involves exchanging information about network interfaces and selecting the appropriate rules. Synchronization ensures hosts have the same rules and use the same network interface for a given destination address. The authors discuss the security considerations associated with using BIAv2, including the potential for denial-of-service attacks and the need for secure signalling between hosts.

BIH

BIH allows IPv4 and IPv6 hosts to communicate during IPv6 transition. It creates a virtual dual-stack host to communicate with IPv4 and IPv6 hosts. IPv6-enabled hosts may communicate with IPv4-only hosts and vice versa. BIH intercepts packets at the network layer and translates IPv4 and IPv6 protocols. BIH-enabled hosts check packets for IPv4 or IPv6 format. BIH transforms IPv4 packets to IPv6 and sends them to IPv6 hosts. BIH translates IPv6 packets to IPv4 and sends them to IPv4 hosts. Figure 7 shows the BIH-based host structure.

BIH enables IPv4 and IPv6 networks to communicate transparently without host reconfiguration. IPv4 or IPv6 addresses may connect hosts. BIH also lets older IPv4 sites connect with IPv6 networks. Address translation and access control from BIH safeguard networks against assaults and illegal access. BIH has drawbacks. In extensive networks, its setup may be complicated. In high-speed networks, BIH's packet translation between IPv4 and IPv6 demands a lot of processing power, increasing latency and performance.

Figure 8 shows that BIH is only used in the following scenarios:

- When an IPv4-only application communicates with an IPv6-only server via a dual-protocol network.
- When an IPv4-only application communicates with an IPv6-only server via an IPv6-only network.
- When an IPv4-only application communicates with an IPv4/IPv6 server across an IPv6-only network.

Figure 7. The structure of BIH-based host

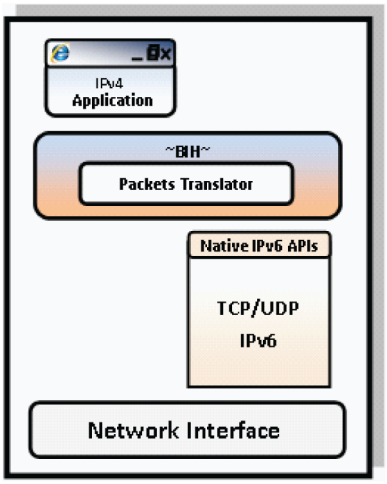


Figure 8. BIH-based host scenarios

BIH			peer
Application Type	Connectivity	Network	Connectivity
IPv4	IPv4/IPv6	←IPv6→	IPv6
IPv4	IPv6	←IPv6→	IPv6
IPv4	IPv6	←IPv6→	IPv4/IPv6

To put it another way, BIH is a technology that allows communication between devices that utilize different IP versions (IPv4 or IPv6) by inserting an intermediate host between them. The BIH is used in the cases indicated above to guarantee that communication can occur despite a mismatch in the IP versions supported by the devices involved and the network to which they are connected.

The host must be set to utilize the BIH translation layer in order to use BIH. This may be accomplished via human configuration or automated procedures like DHCPv6 (Farrer, 2022). The network must also be configured to route data between IPv4 and IPv6 networks. Although it has certain complexity and performance constraints, BIH offers a framework for facilitating communication across IPv4 and IPv6 networks at the host level. Consequently, alternative IPv6 transition mechanisms like tunnelling and translation are often employed in place of BIH.

NETWORK CONFIGURATIONS

Simulating IPv4 and IPv6 networks is complicated by the need for different addressing skills and routing protocols and the varying sizes of end and intermediate networks. Choosing an appropriate host-based translator is essential to allow communication between applications and different IP networks, ensuring a smooth transition. Therefore, evaluating the performance of different host-based translators based on various parameters is critical. The following sections provide network configurations that are intended to offer solutions for various scenarios involving IPv6-only and IPv4-only hosts and applications, enabling seamless communication between devices regardless of the IP protocol used.

IPv6-Only Host That Is Connected to an IPv6-Only Network and Running an IPv4-Only Application

This section discusses the different host-based translator configurations for connecting an IPv6-only host running an IPv4-only application on Site A with a server on Site B through the IPv6 backbone network. The network's design requires various devices or objects, as described in Table 1. The proposed network provides a means to evaluate the performance of various host-based translators based on different parameters, providing valuable insights into their effectiveness. By analyzing these performance parameters, network administrators and researchers can determine which host-based translator mechanism best suits their network's requirements.

We have used numerous host-based translators on networks to study them in this context. Figure 9 depicts the configuration of these host-based translators on an IPv6-only network. We utilized the simulation, profile, link failure recovery, and ping configuration object modules in OPNET simulator (Zhuo et al., 2023) to create the networks to replicate online traffic and video streaming applications. We have independently set up four separate host-based translators, namely BIS, BIA, BIAv2, and BIH, utilizing RIP (Zhou, 2023) on the IPv4 network and RIPng (Lemeshko et al., 2023) on the IPv6 network, as well as OSPF (Hasan et al., 2023) on the IPv4 network and OSPFv3 (Castillo-Velázquez et al., 2023) on the IPv6 network. Four distinct scenarios have been developed based on these combinations: Scenario 1, Scenario 2, Scenario 3, and Scenario 4. Table 2 describes the features of

Table 1. Components used in network configuration

Component	Description	Qty
Nodes	Ethernet_wkstn_adv ppp_wkstn_adv	4 3
Routers	Cisco 7000	2
Switches	Ethernet4_switch_adv	1
Servers	ethernet_server_adv	1
IPv6 backbone	IPv6 cloud	1
Links	PPP_DS3, 100BaseT	8,3
Modules	Application	1
	Profile	1
	Ping Parameter	1
	Link Failure Recovery	1

Figure 9. BIS, BIA, and BIH host-based translators are configured on an IPv6-only network

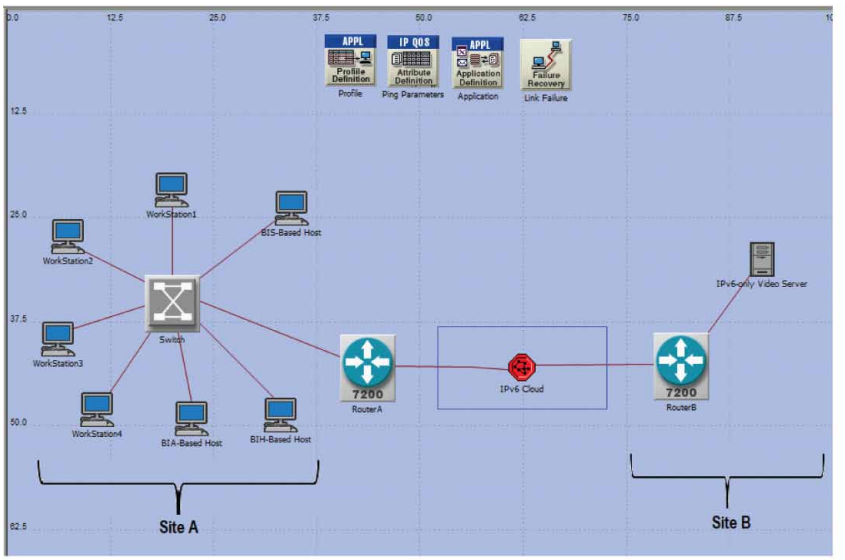


Table 2. Simulation scenarios with routing protocols

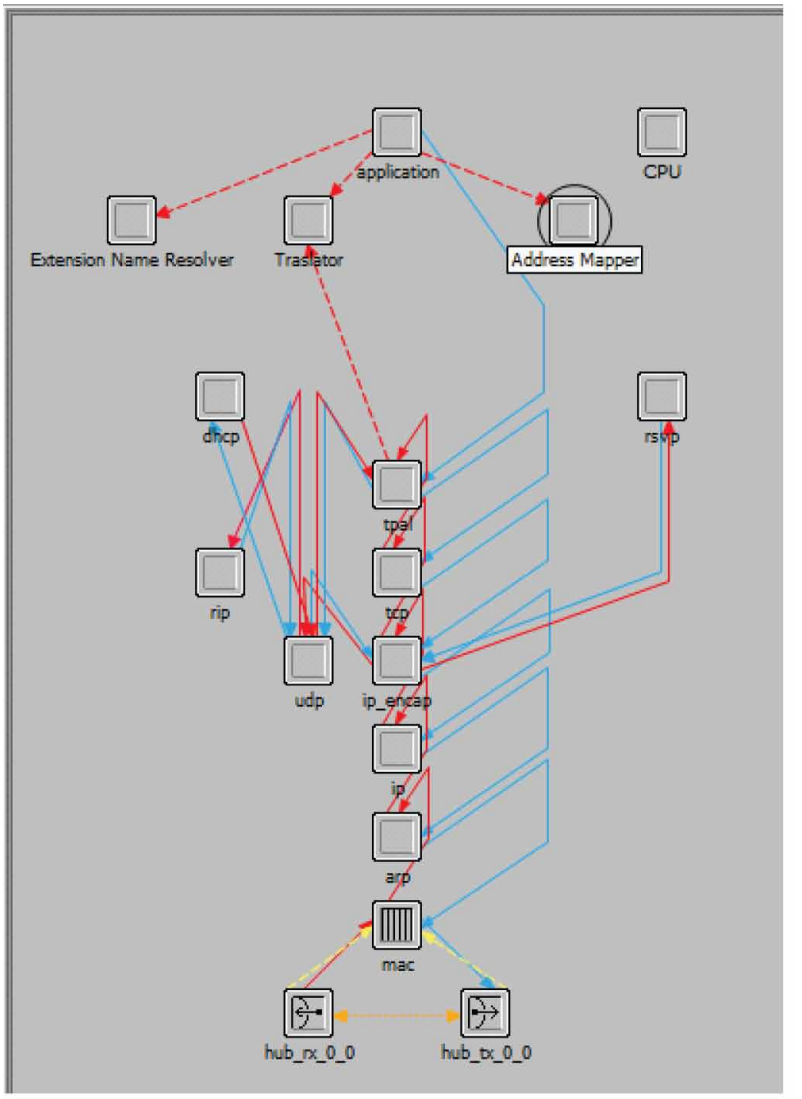
Scenario	Host-Based Translator	Routing Protocol
1	BIS	RIPng
2	BIA	RIPng
3	BIAv2	RIP
4	BIH	OSPFv3

these scenarios, including the BIS RIPng Network, BIA RIPng Network, BIAv2 RIP Network, and BIH OSPFv3 Network. This section goes through the network settings for Scenario 1, Scenario 2, and Scenario 4. These scenarios feature multiple IPv6 and IPv4 host and application combinations, with the network setup intended to promote communication between them. Hosts and applications may interact smoothly regardless of the IP version by establishing the necessary network setup.

This network design comprises two locations, Site A and Site B. Site A consists of three kinds of hosts: BIS, BIA, and BIH-based hosts. BIS hosts are critical in allowing communication between IPv4-only applications operating on IPv6-only hosts and their IPv6 counterparts.

The BIS-based hosts comprise three main components: address mapper, extension name resolver, and translator. The address mapper is responsible for mapping the IPv4 addresses of the application servers to the IPv6 addresses of the BIS hosts. The extension name resolver resolves any name

Figure 10. BIS-based host configuration using OPNET simulator

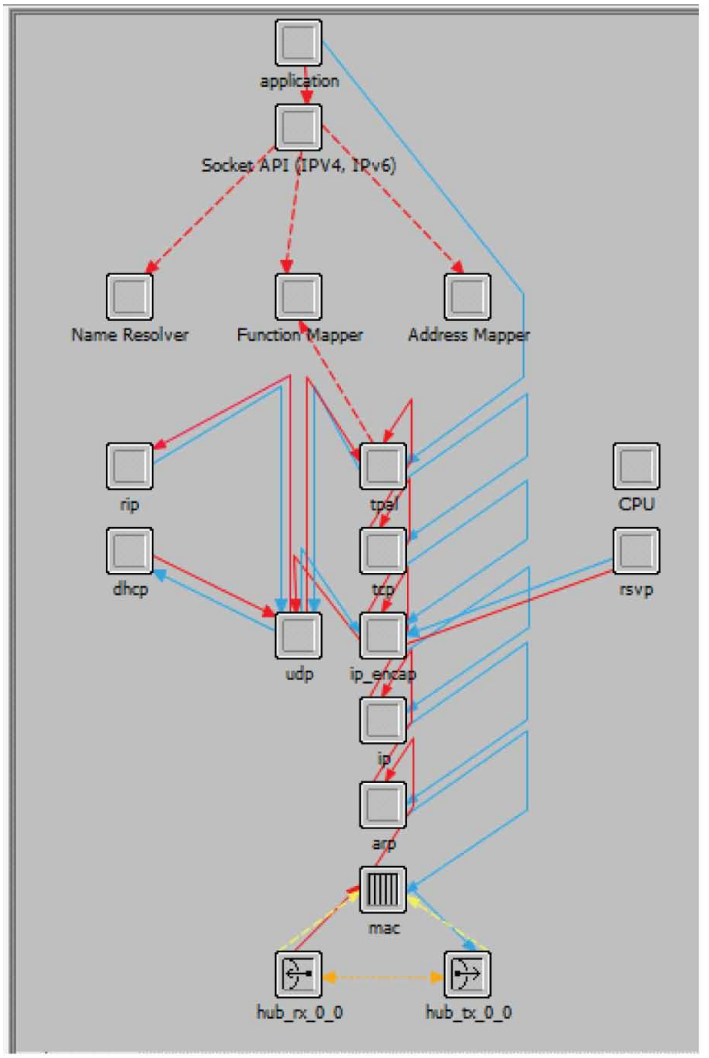


extensions required to access the application servers, and the translator translates the IPv4 packets from the application servers to IPv6 packets that can be transmitted over the network.

Figure 10 represents the components of BIS-based hosts used in the simulator. By implementing this network configuration, IPv6-only hosts can communicate with IPv4-only applications by relying on the translation mechanisms of the BIS-based hosts. This configuration ensures seamless communication between hosts and applications despite differences in their IP protocols.

Another type of host used in this network configuration is the BIA-based host. This host facilitates communication between hosts and applications that use different socket APIs. Figure 11 provides a detailed overview of BIA-based hosts' components, including the name resolver, address mapper, and function mapper. The name resolver component is responsible for resolving the names of the application servers with which the BIA-based hosts need to communicate. The address mapper maps the IPv4 addresses of the application servers to the IPv6 addresses of the BIA-based hosts. Finally, the function mapper maps the socket API functions of the IPv4-only application to their corresponding

Figure 11. BIA-based host configuration using OPNET simulator



functions in the IPv6 socket API. Unlike the BIS-based hosts, which translate the IP packet header, the BIA-based hosts map between the IPv4 and IPv6 socket APIs. This approach ensures seamless communication between hosts and applications, regardless of the socket API used. Overall, the BIA-based hosts provide an efficient way of handling communication between hosts and applications that use different socket APIs, and their components ensure that this communication is efficient and secure.

The network architecture we are discussing also employs a third form of host, a BIH-based host. Depending on the needs of the hosts and applications participating in the connection, this host may use both socket API translation and IP packet translation. Like the BIA-based hosts we covered previously, the BIH-based host may employ socket API translation to facilitate communication between hosts and apps using various socket APIs. When hosts and applications utilize various IP protocols, such as IPv4 and IPv6, the BIH-based host may employ IP packet translation to ease communication. This includes altering the IP packet header and any required modifications to the packet content to facilitate transmission across the multiple protocols. Both translation algorithms are employed in the simulation for this network setup, and the BIH-based host chooses which one to use randomly each time an IPv4-only application begins communication with an IPv6 server.

The structure of the BIH-based host is shown in Figure 12, which contains components such as the protocol translator, name resolver, function mapper, and address mapper. Incoming packets are intercepted by the protocol translator and sent to the appropriate protocol mapper or address translator, which then conducts the required translation and modification before passing the packet to its destination. Overall, the BIH-based host is a flexible component in this network architecture, capable of responding to various communication conditions and guaranteeing smooth connection between hosts and applications using various IP protocols and socket APIs.

IPv4-Only Host That Is Connected to an IPv4-Only Network and Running an IPv6-Only Application

This section will discuss the BIAv2 host-based translator configurations that connect an IPv4-only host running an IPv6-only application on Site A with a server on Site B through the IPv4 backbone network. To achieve this, the network design requires various devices and objects, described in detail in Table 3. The proposed network provides a means to evaluate the performance of the BIAv2 host-based translator based on different parameters. These parameters include factors such as packet loss, delay, and throughput. By analyzing these performance parameters, network administrators and researchers can determine which host-based translator mechanism best suits their network's requirements. The BIAv2 host-based translator is a crucial component in this network configuration, as it enables communication between IPv4-only hosts and IPv6-only applications. This translation is achieved by mapping between the IPv4 and IPv6 socket APIs. By evaluating the performance of the BIAv2 host-based translator, network administrators can gain insights into its effectiveness and identify any areas for improvement. For example, if packet loss is consistently high, adjusting the network configuration or using a different translator mechanism may be necessary to improve performance.

The suggested network configuration offers an excellent testing environment for BIAv2 host-based translators and other translation methods. Network administrators and researchers may make educated judgments about how to enhance their network's performance and guarantee flawless communication between hosts and applications utilizing various IP protocols by examining the results of these tests.

Figure 13 depicts the configuration of a BIAv2 host-based translator on an IPv4-only network. We utilized the simulation, profile, link failure recovery, and ping configuration object modules in the OPNET simulator to create the networks to replicate online traffic and video streaming applications.

The BIAv2-based host is intended to ease communication between hosts and apps that utilize various socket APIs. BIAv2 provides broader scenarios than its predecessor, allowing IPv6-only programs to connect with IPv6-only peers. Figure 14 depicts a thorough overview of the components of BIAv2-based hosts, including the name resolver, address mapper, and function mapper. The name resolver is critical in resolving the names of the application servers with which the BIAv2-based

Figure 12. BIA-based host configuration using OPNET simulator

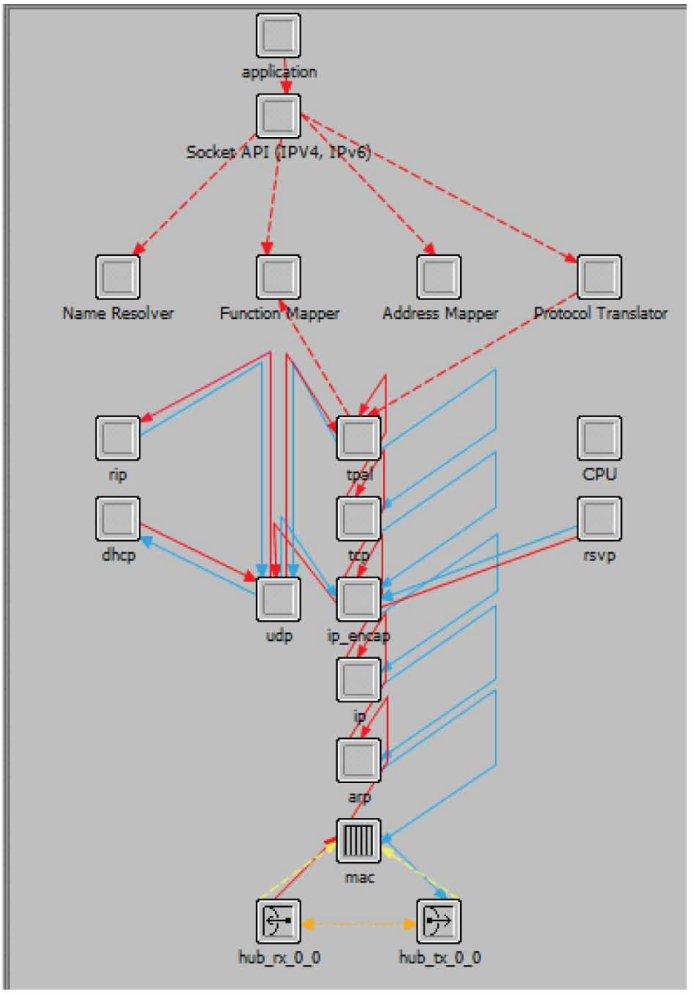


Table 3. Network configuration for Scenario 3

Component	Description	Qty
Nodes	Ethernet_wkstn_adv ppp_wkstn_adv	6 1
Routers	Cisco 7000	2
Switches	Ethernet4_switch_adv	1
Servers	ethernet_server_adv	1
IPv6 backbone	IP32_cloud	1
Links	PPP_DS3, 100BaseT	8,3
Modules	Application	1
	Profile	1
	Ping Parameter	1
	Link Failure Recovery	1

Figure 13. BIAv2 host-based translator is configured on an IPv4-only network

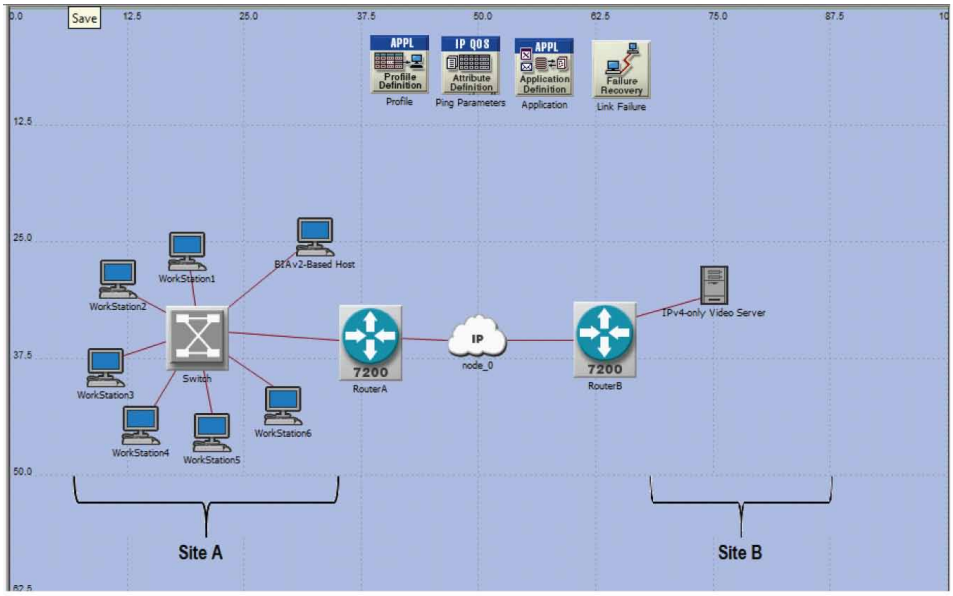
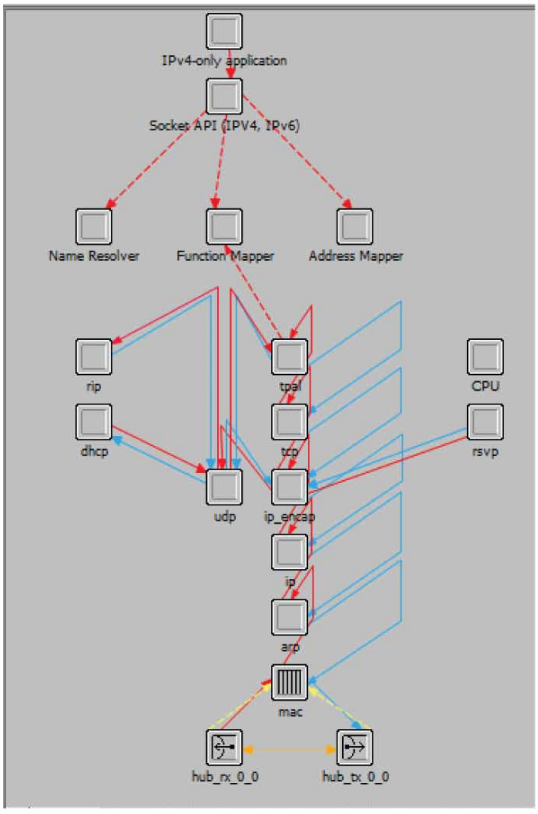


Figure 14. BIAv2-based host configuration using OPNET simulator



hosts must connect. Meanwhile, the address mapper oversees translating the application servers' IPv4 addresses to the IPv6 addresses of the BIAv2-based hosts. Like the original BIA, BIAv2 enables IPv6-only programs to interact across IPv4. This requires establishing a connection between external IPv4 addresses and internal IPv6 addresses. BIAv2 is sometimes referred to as "IPv4-in-IPv6" address embedding. The address resolver is set up to generate an IPv4-embedded IPv6 address, which consists of a 32-bit network-specific prefix (NSP), a 32-bit IPv4 destination address, and a 64-bit suffix. Finally, the function mapper works differently depending on the host's connection. The function mapper is used in dual connectivity hosts (IPv4 and IPv6 connection) to decide which API functions to call in the present communication. In the event of an IPv4-only application connecting via IPv6, the relevant IPv6 socket API methods will be called. The program will use the IPv4 socket API to connect with other hosts. Because the application requires IPv6, the function mapper intercepts IPv4 socket API functions and invokes IPv6 socket API functions instead.

RESULTS AND ANALYSIS

Global parameters are a collection of measurements used in computer networking to gauge a network's overall performance and efficiency. These characteristics are derived by examining different network aspects such as traffic delivered and received (Soto et al., 2022), packet loss rate (Ding et al., 2022), network convergence time (Rybowski & Bonaventure, 2022), and network throughput (Sentala et al., 2022).

IP traffic dropped is an essential global parameter that measures the number of packets not successfully delivered to their intended destination. Various circumstances, including network congestion, routing mistakes, or device issues, may cause dropped IP traffic. Monitoring this metric is crucial for identifying possible performance problems and taking remedial steps to guarantee that network services remain available and dependable. Another critical global metric is network convergence duration, which is the time it takes to recover after a breakdown or outage. This value is crucial for restoring network services immediately following an interruption. The time it takes for the network to re-establish its routing tables and protocols after a failure is used to calculate network convergence duration. Another critical global indicator is throughput, which quantifies the volume of data transferred through a network in a particular time. This number, usually in bits or bytes per second, measures the network's ability to handle traffic. Throughput monitoring may assist network administrators in identifying possible bottlenecks and optimizing network efficiency. Finally, the global parameter of traffic sent quantifies the volume of data transferred across the network. This parameter is critical for determining the amount of data provided and received by network devices. Network administrators may spot possible performance problems or unexpected patterns of network utilization by monitoring traffic transmitted.

When considering host-based translators, these global factors become more critical. They allow devices that utilize various network standards or protocols to communicate with one another. Host-based translators may considerably influence network performance. While utilizing these devices, it is critical to check global parameters to ensure that network services remain dependable. Calculating global parameters is crucial for monitoring network performance and health, especially when utilizing host-based translators. Network administrators may detect possible faults and take remedial action to ensure network services remain accessible and dependable by examining data such as IP traffic lost, network convergence length, throughput, and traffic transmitted.

At Site A, numerous hosts run programs that originate network traffic, including BIS, BIA, BIAv2, and BIH. This network traffic is destined for a video server at Site B. The simulation lasts two hours, during which the network is assessed for different simulation settings. The simulation progress table in Table 4 shows data acquired for various simulation settings.

Table 4. Simulation parameters

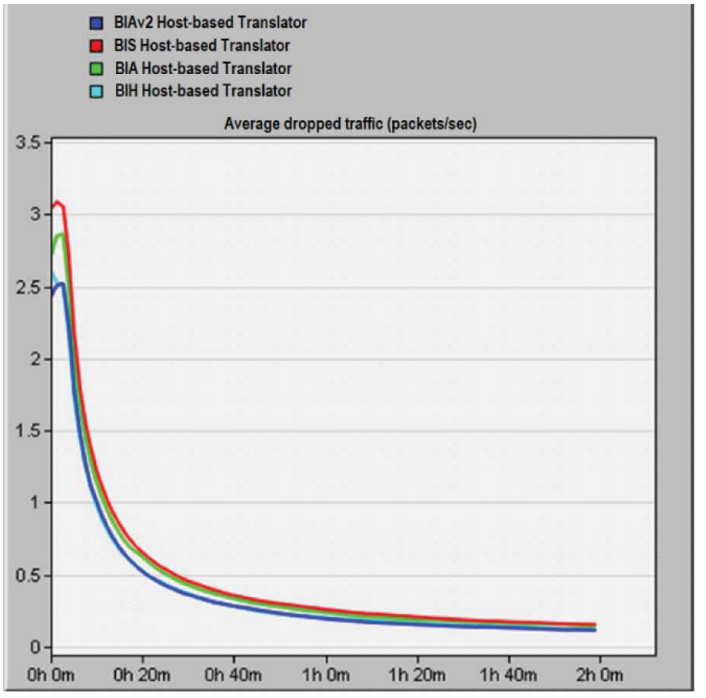
Parameter	Value
Simulation time	2 hours
Speed	132,668 events/sec
Elapsed time	17 sec
Total event	61,254

IP Traffic Dropped

IP traffic dropped refers to all IP packets lost across all IP interfaces in a network and is measured in packets per second. The study examines four scenarios and evaluates IP traffic dropped over a two-hour simulation period when host-based translators initiate traffic from Site A to the video server on Site B. The results indicate that all four scenarios demonstrate a similar trend in dropping IP traffic, with packet drops initially ranging from 2.4 to 3.2 per second and decreasing over time. There are two possible reasons for the dropped packets. First, host-based translators at the sender add extra processing overhead to the network, leading to a buffer overflow, which can result in packet loss. Second, packets may be lost at the router buffer if the rate of receiving packets is lower than the rate of incoming packets. Figure 15 shows the IP traffic dropped for all host-based translators.

Further analysis reveals that the BIS and BIA host-based translators with RIPng have a higher IP traffic dropped rate of 0.24 and 0.23 packets per second, respectively. In contrast, the BIH host-based translator with OSPFv3 has a lower dropped packet rate of 0.22 packets per second, and the BIAv2 host-based translator with RIP has the lowest dropped packet rate of 0.20 packets per

Figure 15. Average dropped traffic for BIS, BIA, BIH, and BIAv2 host-based translators (from Site A to Site B)



second. Thus, the study suggests that BIS, BIA, and BIH host-based translators result in a higher IP traffic drop rate than the BIAv2 host-based translator. The initial packet drop rate observed in the simulation could be attributed to the initial network configuration, which requires time to stabilize. However, the packet drop rate decreases as the simulation progresses, indicating that the network configuration stabilizes. Host-based translators add processing overhead, leading to increased buffer utilization. This could explain the higher packet drop rate observed in the BIS, BIA, and BIH host-based translators with RIPng.

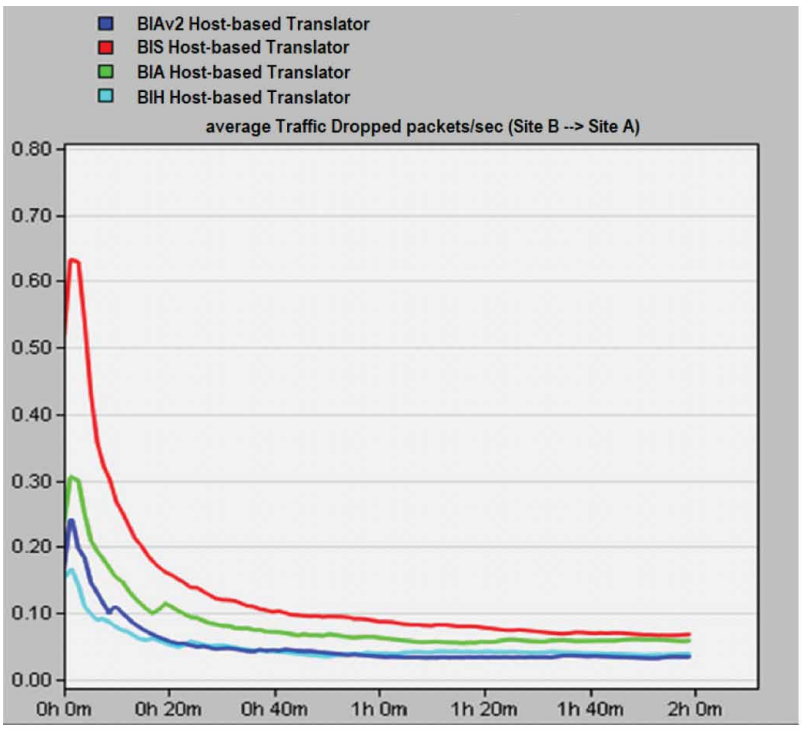
Conversely, the BIAv2 host-based translator with RIP has a lower packet drop rate, which could be attributed to using a different routing protocol or lower processing overhead. The results suggest that selecting the appropriate host-based translator and routing protocol is crucial to minimizing IP traffic dropped in a network. Table 5 shows the average dropped packets within 2 hours.

The study measures the dropped traffic while the host-based translators download video traffic from a video server located in Site B. Figure 16 shows that all scenarios have higher traffic dropped

Table 5. Average dropped packets/sec within 2 HOURS

	1 sec	1200 sec	2400 sec	3600 sec	4800 sec	6000 sec	7200 sec
BIS	3.2	0.65	0.4	0.34	0.32	0.25	0.24
BIA	2.8	0.55	0.38	0.33	0.31	0.25	0.23
BIAv2	2.4	0.52	0.35	0.30	0.29	0.24	0.20
BIH	2.6	0.53	0.37	0.32	0.30	0.24	0.22

Figure 16. Dropped traffic from Site B to Site A



initially, which gradually decreases as time increases. The loss of packets can occur due to several reasons, such as the buffer being overwhelmed, packets taking too much time to reach the destination, congested networks, or delays at the receiving host-based translator.

Among all the cases, the network with BIS-based hosts has the highest drop rate of 0.132547 packets per second, while the network with BIA-based hosts dropped 0.079365 packets per second. However, the network with BIH-based hosts has a lower drop rate of 0.049214 packets per second. Interestingly, the network with BIAv2-based hosts has the minimum drop rate of 0.042258 packets per second compared to all other cases. One possible reason is that, unlike other host-based translators, the networks with BIAv2-based hosts use socket API mapping and stateless address translation, which reduces the overhead caused by translating IP headers between protocols.

Dropped traffic may be a severe problem for video streaming apps, resulting in a bad user experience and reduced video quality. As a result, decreasing missed packets is critical for improving network performance. Based on the study's findings, it is possible to conclude that deploying host-based translators, particularly BIAv2-based hosts, might minimize missed packets and enhance network performance. However, the efficiency of these host-based translators might vary depending on the network environment. Implementing them requires careful consideration of their advantages and disadvantages.

Network Convergence

The duration required for the IP forwarding tables to converge is known as network convergence duration, measured in seconds. It refers to the time needed to calculate the best path, update routing tables, and share the correct information among routers in a network. A faster network convergence rate leads to a better network. Four scenarios were simulated, and the results in Figure 17 showed that a network with BIH-based hosts had a convergence duration of 0.565223 seconds, while a network with BIAv2-based hosts had a convergence duration of 0.535254 seconds. However, a network with BIA-bae hosts took 1.87885 seconds, and a network with BIS-based hosts required 1.912514 seconds to converge. The results indicate that a network with BIS-based hosts converges slower than BIAv2-based and BIH-based hosts.

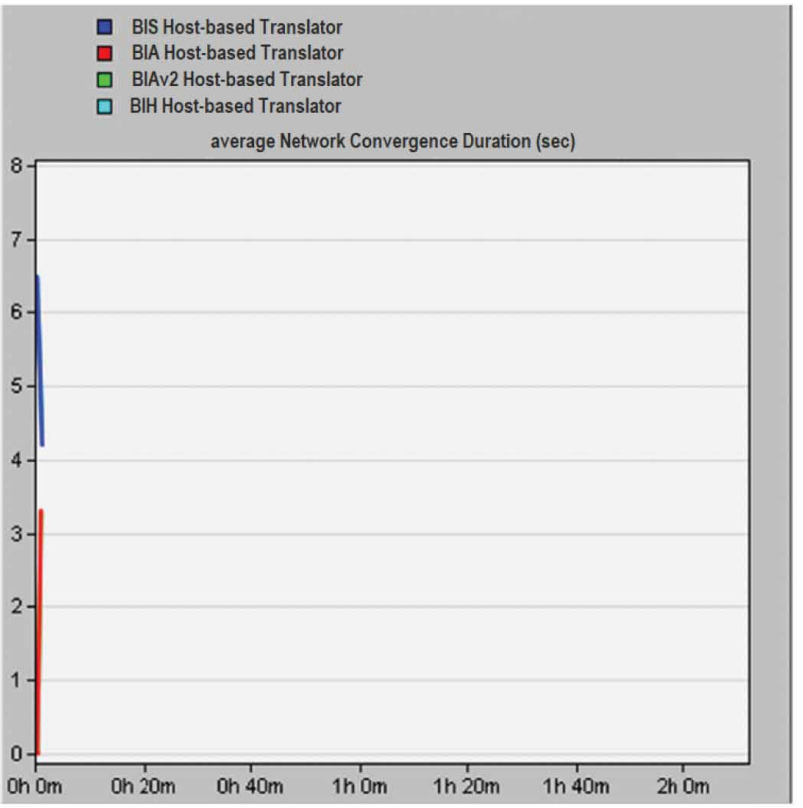
Host-based translators such as BIH and BIAv2 can help improve the network convergence rate by reducing packet loss and enhancing the stability of TCP connections. This can be particularly useful in networks that have NAT (Bhattacharjya et al., 2019) devices and firewalls, which can disrupt or slow down TCP connections. BIAv2 and BIH modify certain aspects of the TCP connection in transit, such as sequence and acknowledgement numbers, to ensure that packets are delivered reliably and in the correct order, even during network failures or routing changes. This can reduce the time needed for the network to converge after a failure or routing change since TCP connections can recover faster and resume regular operation.

On the other hand, the success of host-based translators depends on the individual network environment and the network operator's aims. Host-based translators may contribute more complexity and delay into the network in certain circumstances, impeding convergence or creating other difficulties. Furthermore, host-based translators may fail in all network circumstances, especially if the underlying network architecture is unsuitable for these approaches. Consequently, before installing host-based translators, it is necessary to carefully weigh the advantages and disadvantages in the context of each given network environment.

Traffic Delivered and Received

Traffic delivered and received are two critical performance measures in contemporary computer networks that network managers and engineers closely monitor. The traffic sent metric measures the number of packets sent by a network device or node, while the traffic received metric measures the number of packets received by the same device or node. These measurements are crucial for

Figure 17. The duration of network convergence when using host-based translators



guaranteeing the network's efficiency and effectiveness since they may give insight into possible bottlenecks, congestion spots, and other difficulties that may develop.

Traffic delivered and received take on added relevance in the context of host-based translators. Host-based translators are devices that help connect various networks or devices that employ different communication protocols. A host-based translator, for example, may simplify communication between networks that use the IPv4 protocol and devices that utilize the IPv6 protocol. In this case, traffic delivered refers to packets sent by the host-based translator's CPU to all IP interfaces on the network. This traffic enables connectivity across various devices or networks since it allows for translating communication protocols. In contrast, traffic received refers to packets received by the host-based translator from all IP interfaces on the network. This traffic is also essential since it allows the host-based translator to ease communication across various networks or devices.

Network engineers often utilize tools such as network analyzers or packet sniffers to measure traffic delivered and received. These technologies collect and analyze network traffic in real-time, giving extensive information on how much traffic is delivered and received by various devices or nodes. Aside from these tools, simulation is a popular way of measuring traffic delivered and received.

Different network scenarios are generated and executed in a simulation, with traffic delivered and received monitored for each scenario. This allows network engineers to evaluate the performance of various devices or nodes under a range of scenarios without having to do costly and time-consuming real-world testing.

The BIS-based host, BIA-based host, BIH-based host, and BIAv2-based host were the four host-based translators evaluated, as shown in Figure 18. Two hours were spent testing each of these

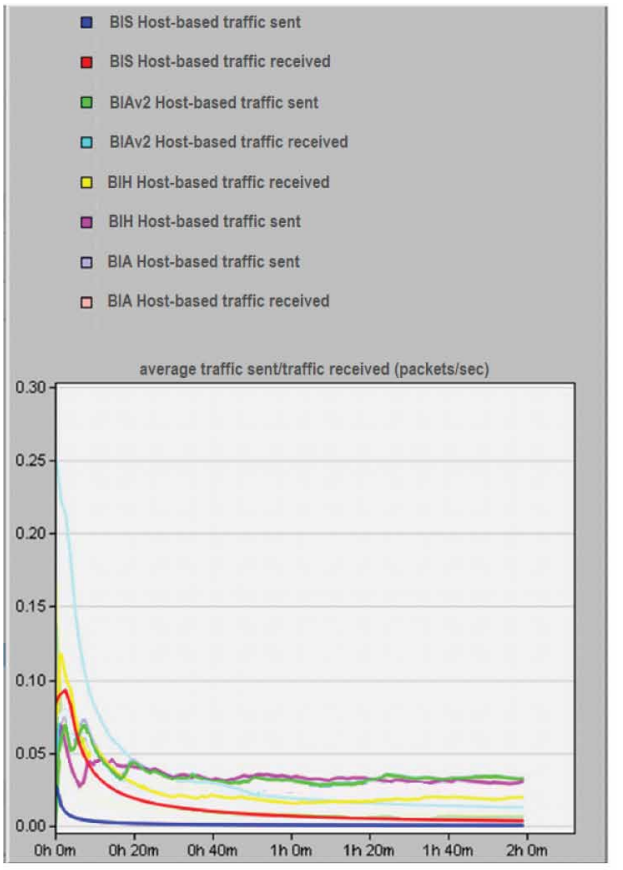
translators, with traffic delivered and received being monitored the whole time. The simulation's findings demonstrated that BIAv2 and BIH-based hosts transmitted more traffic than BIS and BIA-based hosts. BIAv2 transmitted 0.032543 packets per second of transmission, and BIH sent 0.057554 packets per second. BIS and BIA, on the other hand, sent 0.030215 packets per second and 0.026521 packets per second, respectively.

Figure 18 depicts the results of four separate host-based translator tests: BIS-based host, BIA-based host, BIH-based host, and BIAv2-based host. Each of these translators was tested for two hours, with traffic delivered and received being measured at all times. The simulation results revealed that BIAv2 and BIH-based hosts delivered more traffic than BIS and BIA-based hosts. BIAv2 sent 0.032543 packets per second and BIH sent 0.057554 packets per second, whereas BIA sent 0.030215 packets per second and BIS sent 0.026521 packets per second.

Traffic sent figures are not always indicative of overall network performance. While larger figures for traffic transmitted may indicate that a device or node is doing well, this is not necessarily the case. High traffic sent numbers may indicate that a device or node is creating excessive overhead or suffering network congestion, which may have a detrimental influence on overall network performance in certain instances.

The simulation measured both traffic delivered and traffic received. The findings indicated that both BIAv2 and BIH-based hosts got more traffic from the video server at Site B, with computed values of 0.25 and 0.18 packets per second, respectively. BIS and BIA-based hosts got less traffic,

Figure 18. Average traffic sent and received



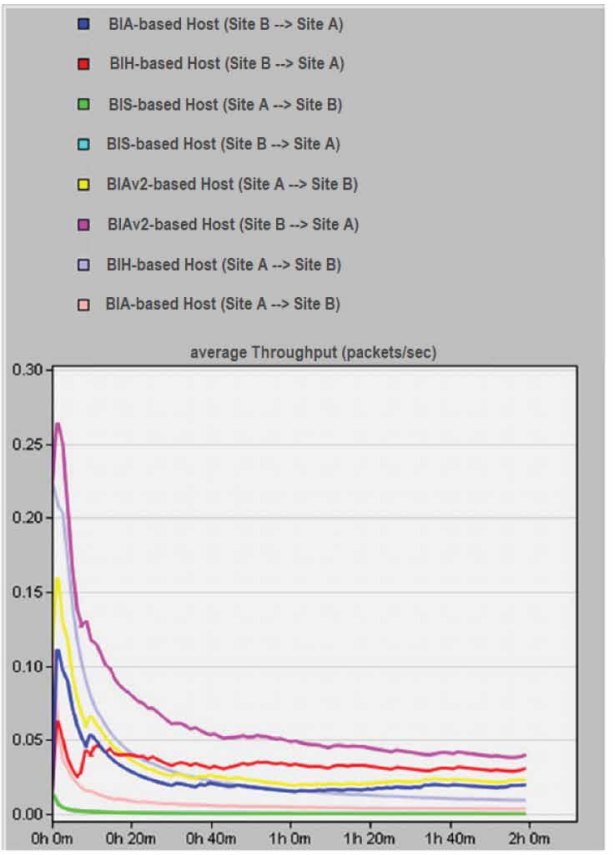
with computed values of 0.028745 and 0.036521 packets per second, respectively. As with traffic transmitted, traffic received figures do not necessarily indicate overall network performance. While larger figures for traffic received may indicate that a device or node is working well, this is not necessarily the case. Even if a device or node is working properly, factors like packet loss, network congestion, and connection failures may all influence traffic received. Overall, the simulation findings give useful information on the performance of several host-based translators under various scenarios. By analyzing both sent and received traffic, network engineers may better understand how these devices function and how they affect overall network performance.

Network engineers may use various ways to assess traffic transmitted and received in addition to simulations. They may, for example, use network monitoring systems that offer real-time data on network traffic. These technologies may give vital insight into network performance, allowing engineers to detect and resolve problems swiftly. Finally, measuring traffic transmitted and received is vital to ensure that contemporary computer networks operate efficiently and effectively. Network engineers can discover possible faults, enhance network performance, and guarantee that communication between various devices and networks is accessible and dependable by monitoring these data.

Throughput

The study measures the data throughput between a host-based translator at Site A and a video server at Site B while considering various host-based translation strategies. Figure 19 shows the data throughput between Site A's host networks and Site B's video server using BIS, BIA, BIH, and BIAv2.

Figure 19. Average throughput host-based translator ←→ video server



The average data throughput from Site A to Site B was calculated to be 0.042587 packets per second for the BIAv2-based host network, 0.038478 packets per second for the BIH-based host network, and 0.034242 packets per second for the BIA-based host network. Similarly, the rate at which data was sent from the host-based translator at Site A to the video server at Site B was measured. The average data throughput for the host networks based on BIS, BIA, BIH, and BIAv2 was determined to be 0.033254 packets per second, 0.038878 packets per second, 0.052145 packets per second, and 0.062147 packets per second, respectively.

The research found that the BIAv2-based host network had the maximum data throughput compared to the other host-based translators. Two factors are responsible for this result. First, BIAv2 reduced the translation function's processing load by using straightforward socket IPv4/IPv6 mapping techniques. Second, BIAv2 used a stateless translation mode that did not need a mapping database to be kept up to date to keep track of translated packets. Instead, to determine the IPv4 addresses used in the translation process, BIAv2 used a straightforward method. These features significantly decreased the processing overhead, which sped up packet-forwarding.

This research has important practical implications for organizations and network administrators planning and executing the IPv6 transition. Host-based translators are a crucial stopgap, allowing the legacy IPv4 infrastructure to communicate with the emerging IPv6 devices and networks during this migration period. However, mismatches in performance needs and capabilities can lead to suboptimal utilization of these translation mechanisms. By exhaustively profiling the performance of prominent translators under diverse conditions, this study empowers companies to select the optimal solution tailored to their specific industrial environments and connectivity requirements. For instance, enterprises seeking to maximize throughput for IPv4-IPv6 video streaming may opt for BIAv2 deployment based on the benchmark results. Alternatively, networks with paramount reliability and rapid failover would likely benefit from BIH integration. Equipped with these performance insights, industry practitioners can smoothly navigate the IPv6 adoption process, avoiding pitfalls and maximizing efficiency. This work furthers industrial progress by elucidating best practices for successful technological transition amid continued evolution.

CONCLUSION AND FUTURE WORK

In conclusion, this work provided an exhaustive simulation-based analysis of performance differences across four widely adopted host-based translator mechanisms—BIS, BIA, BIAv2, and BIH. The study benchmarked critical metrics, including packet loss, convergence time, traffic overhead, and throughput under diverse network configurations encompassing varied routing protocols, IPv4/IPv6 applications, and connections. Several illuminating findings emerged from this comprehensive performance profiling. All translators exhibited a high initial packet loss that declined over time, with BIAv2 demonstrating the lowest loss overall. BIAv2 and BIH enabled substantially faster convergence, highlighting benefits for reliability. BIAv2 also offered advantages in minimizing traffic overhead and maximizing throughput attributed to its efficient stateless mapping. The breadth of results underscored the performance variability tied to specific translator and network conditions. This highlights the need to select translation solutions tailored to individual environments and IPv6 transition goals. By exhaustively mapping this performance landscape, this research contributes to providing definitive guidance to network administrators seeking optimal deployment strategies. The benchmark results and framework established lay the groundwork for additional investigations into evolving real-world conditions, topologies, and emerging network paradigms. Moving forward, this knowledge can continue informing the effective utilization of host-based translators in navigating the nuanced and ongoing process of IPv6 adoption worldwide.

AUTHOR NOTE

The authors would like to thank the research deanship at Shaqra University for supporting the research.

REFERENCES

Abdalaziz, Y., & Hamarsheh, A. (2020). Analyzing the IPv6 deployment process in Palestine. *International Journal of Computer Network and Information Security*, *12*(5), 31–45. doi:10.5815/ijcnis.2020.05.03

Bhattacharjya, A., Zhong, X., Wang, J., & Li, X. (2019). On mapping of address and port using translation. *International Journal of Information and Computer Security*, *11*(3), 214–232. doi:10.1504/IJICS.2019.099419

Castillo-Velázquez, J. I., Varela-Sánchez, I., Buendia-Gomez, Y., & Huerta, M. K. (2023, January 29-31). *The Pacific wave advanced network backbone: An emulation approach under IPv6* [Paper presentation]. 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China. doi:10.1109/ICPECA56706.2023.10075846

Ding, W., Zhai, W., Liu, L., Gu, Y., & Gao, H. (2022). Detection of packet dropping attack based on evidence fusion in IoT networks. *Security and Communication Networks*, *2022*, 1–14. Advance online publication. doi:10.1155/2022/1028251

Farrer, I. (Ed.). (2022). RFC 9243: A YANG Data Model for DHCPv6 Configuration. Academic Press.

Hamarsheh, A. (2019). Deploying IPv4-only connectivity across local IPv6-only access networks. *IETE Technical Review*, *36*(4), 398–411. doi:10.1080/02564602.2018.1498031

Hamarsheh, A. & AbdAlaziz, Y. (2019, April 3-4). *Transition to IPv6 protocol, Where we are?* [Paper presentation]. International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia. doi:10.1109/ICCISci.2019.8716482

Hamarsheh, A., & Eleyat, M. (2018). Performance analysis of Ain-Pt, Ain-Slt And Siit network-based translators. In F. Xhafa, S. Caballé, & L. Barolli (Eds.), *Advances on P2P, parallel, grid, cloud and internet computing: Proceedings of the 12th international conference on P2P, parallel, grid, cloud and internet computing (3PGCIC-2017)* (pp. 367–378). Springer International Publishing. doi:10.1007/978-3-319-69835-9_35

Hamarsheh, A., & Goossens, M. (2012). Illustrating the impediments for widespread deployment of IPv6. *Proceedings of the 11th International Conference on Signal Processing*. https://www.researchgate.net/profile/Ala-Hamarsheh/publication/224729779_Illustrating_the_Impediments_for_Widespread_Deployment_of_IPv6/links/543769fa0cf2643ab9889e52/Illustrating-the-Impediments-for-Widespread-Deployment-of-IPv6.pdf

Hamarsheh, A., & Goossens, M. (2014). A review: Breaking the deadlocks for transition to IPv6. *IETE Technical Review*, *31*(6), 405–421. doi:10.1080/02564602.2014.950348

Hamarsheh, A., Goossens, M., & Al-Qerem, A. (2012). Assuring interoperability between heterogeneous (IPv4/IPv6) networks without using protocol translation. *IETE Technical Review*, *29*(2), 114–132. doi:10.4103/0256-4602.95384

Hamarsheh, A., Goossens, M., & Alasem, R. (2011a). Configuring hosts to auto-detect (IPv6, IPv6-in-IPv4, or IPv4) network connectivity. *KSII Transactions on Internet and Information Systems*, *5*(7), 1230–1251. doi:10.3837/tiis.2011.07.002

Hamarsheh, A., Goossens, M., & Alasem, R. (2011b). Decoupling application IPv4/IPv6 operation from the underlying Ipv4/Ipv6 communication (DAC). *American Journal of Scientific Research*, (14), 101–121. https://www.researchgate.net/publication/215595605_Decoupling_Application_IPv4IPv6_Operation_from_the_Underlying_IPv4IPv6_Communication_DAC

Hasan, H., Cosmas, J., Shanshool, B., & Khwandah, S. (2023). Employing the Sdn principle to reduce the link failure effect efficiently in conventional Ospf/Mpls routed network. *Telematique*, *22*(01), 1070–1088.

Huang, B., Deng, H., & Savolainen, T. (2012). *Dual-stack hosts using "bump-in-the-host" (BIH)*. Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc6535.html

Lee, S., Shin, M. K., Kim, Y. J., Nordmark, E., & Durand, A. (2002). *Dual stack hosts using "bump-in-the-API" (BIA)*. The Internet Society. https://www.rfc-editor.org/rfc/rfc3338

Lemeshko, O., Yeremenko, O., Mersni, A., Yevdokymenko, M., Persikov, M., & Kruhlova, A. (2023, February 22–25). *Analysis of Proactive Models of Fault-Tolerant Routing under Load Balancing and Border Routers Availability* [Paper presentation] 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Jaroslaw, Poland. doi:10.1109/CADSM58174.2023.10076525

Rybowski, N., & Bonaventure, O. (2022, June 22-23). *Evaluating OSPF Convergence with ns-3 DCE* [Paper presentation]. 2022 Workshop on ns-3, Worldwide. doi:10.1145/3532577.3532597

Sentala, B., Lubobya, C. S., & Zulu, A. (2022). Performance evaluation and compression of IP packets in a wireless local area network (WLAN). *Journal of Wireless Networking and Communications*, *11*(1), 1–10. doi:10.5923/j.jwnc.20221101.01

Soto, I., Calderon, M., Amador, O., & Urueña, M. (2022). A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies. *Vehicular Communications*, *33*, 100428. doi:10.1016/j.vehcom.2021.100428

Tsuchiya, K., Higuchi, H., & Atarashi, Y. (2000). *Dual stack hosts using the "bump-in-the-stack" technique (BIS)*. The Internet Society. https://www.rfc-editor.org/rfc/rfc2767

Zhou, Z. (2023). RIP analysis for the weighted ℓr-ℓ1 minimization method. *Signal Processing*, *202*, 108754. doi:10.1016/j.sigpro.2022.108754

Zhuo, Z., Huang, J., Lu, W., & Lu, X. (2023). Research on communication stability of inter-cannonball network ased on OPNET. *Applied Sciences (Basel, Switzerland)*, *13*(7), 4588. doi:10.3390/app13074588

Ala Hamarsheh is an associate professor at the Faculty of Engineering and Information Technology of the Arab American University of Jenin. Dr Hamarsheh has obtained a PhD in engineering sciences from Vrije Universiteit Brussel (VUB)/Brussels-Belgium in 2012. He graduated in computer science at the Faculty of Science, Birzeit University, Palestine, in 2000. He obtained an MSc degree in computer science at the Kind Abdullah II School for IT, The University of Jordan, Jordan, in 2003. Dr Hamarsheh has published numerous papers in international refereed journals and conferences.

Mohammad Alauthman Received PhD degree from Northumbria University Newcastle, UK in 2016. He received a B.Sc. degree in Computer Science from Hashemite University, Jordan, in 2002, and received M.Sc. degrees in Computer Science from Amman Arab University, Jordan, in 2004. Currently, he is Assistant Professor and senior lecturer at Department of Information Security, Petra University, Jordan. His main research areas cyber-security, Cyber Forensics, advanced machine learning and data science applications.

Amjad Aldweesh is a computer assistant professor interested in the Blockchain and Smart contracts technology as well as cyber security. Amjad has a Bachelor degree in computer science. He has a MSc degree in advanced computer science and security from the University of Manchester with distinction. Amjad is the second in the UK and the first in the middle east to have a PhD in the Blockchain and Smart contracts technology from Newcastle University.

Someah Alangari is an Assistant Professor, at College of Science and Humanities Dawadmi at Shaqra University, Saudi Arabia. Dr. Someah has got her Ph.D in Computer Science at University of Southampton in the UK. She received her master's degree in Software Engineering at University of Southampton in the UK, and BSc in Computer Science at King Saud University in Saudi Arabia. Her research interests include Blockchain, ML, software engineering, and information systems. Dr. Someah has over 10 years of working experience in the academic sector.