IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 June 2022, accepted 23 June 2022, date of publication 30 June 2022, date of current version 8 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187421

METHODS

Migrating From IP to NDN Using Dual-Channel Translation Gateway

FERI FAHRIANTO[1,2], (Graduate Student Member, IEEE),
AND NORIAKI KAMIYAMA[3], (Member, IEEE)
[1]Graduate School of Engineering, Fukuoka University, Fukuoka 819-0395, Japan
[2]Faculty of Science and Technology, Syarif Hidayatullah State Islamic University Jakarta, South Tangerang 15412, Indonesia
[3]College of Information Science and Engineering, Ritsumeikan University, Shiga 525-0058, Japan

Corresponding author: Feri Fahrianto (feri.fahrianto@uinjkt.ac.id)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 18K11283 and Grant 21H03437, and in part by the Ministry of Religious Affairs of the Republic of Indonesia (MORA 5000 Doktor).

ABSTRACT The emergence of communication networks has resulted in a system of content-based networks. Therefore, a novel network design called Information-Centric Networking (ICN) promotes the efficient transfer of content objects, whereas Named-Data Networking (NDN) is one of the most promising candidates. While migration from Internet Protocol (IP) to NDN is inevitable, it is prohibitively expensive to replace all routers instantaneously with NDN routers during the transition period. Hence, we propose the dual-channel IP-to-NDN translation gateway to address this problem. The gateway promotes the semantics of the IP protocol to be equal to those of NDN at the network layer by utilizing two unique IP addresses as channels so that it can distinguish an IP packet as either an interest or a data packet. In addition, the proposed gateway uses a name binding mechanism to transmit packets between IP and NDN hosts seamlessly by conducting non-data-payload reading to preserve the privacy of users. Finally, the throughput performance of the dual-channel gateway is examined for two distinct binding strategies, namely static and dynamic prefix-name binding. We analyze the throughput by comparing a throughput-estimation model with an emulation testbed. Especially, we emphasize the relationship between the hit ratio, processing delay, and throughput. According to numerical evaluation, we show that the throughput-estimation model successfully predicts the throughput with an accuracy of about 90% and 85% in static and dynamic prefix-name binding schemes, respectively. Moreover, the results show that an increase in content popularity skewness, α, from 0.7 to 1.5 improved the throughput of the proposed gateway by about 70% when the cache size was 1% of the prefix name count in the static prefix-name binding. Alternatively, the throughput increased ten times when the content popularity skewness increased from 0.7 to 1.5 in the dynamic prefix-name binding.

INDEX TERMS Co-existence, ICN, migration, NDN, throughput analysis, translation gateway.

I. INTRODUCTION

Information-Centric Networking (ICN) transforms the paradigm of data exchange by renaming data objects on a network. Two distinct packets are required for an ICN data transaction: interest and data packets. To get content from a producer, the consumer first sends an interest packet with a particular prefix name, and the producer responds with a data packet. The ICN communication primitive works in a pull-driven manner, in contrast with IP, which may work in either a push-based or pull-based manner between hosts.

As an emerging ICN technology, Named-Data Networking (NDN) is gaining the attention of major communication network vendors. NDN focuses on material by naming and routing it. A content prefix name is made of one or more variable-length name segments and encoded using a Type Length Value (TLV) encoder in a manner similar to that of a Uniform Resource Identifier (URI). The NDN network layer communicates using application data names. An NDN router is composed of three components: the Forwarding

The associate editor coordinating the review of this manuscript and approving it for publication was Zubair Md. Fadlullah.

Information Base (FIB), which is used to forward interest to other routers or producers based on the Longest-Prefix-Matching (LPM) of their prefix names; the Pending Interest Table (PIT), which is used to forward content messages along reverse paths to consumers; and the Content Store (CS), which is used to cache the content. When a router receives an interest, it first examines its cache to see whether it can satisfy the interest locally. If the CS does not have the necessary material, it contacts its PIT to locate a pending version of the same interest. The new incoming interface is added to the PIT entry if there is a PIT match. Otherwise, according to FIB, it produces a new PIT entry and transmits the interest to the next hop. If an NDN router gets a content object that does not match the prefix name specified in the PIT entry, the content is deleted as unrequested.

The implementation of NDN technology will eventually dominate worldwide communication networks. However, cohabitation of IP and NDN will be unavoidable throughout the transition phase. As a result, a translation gateway that enables cross-platform communication between IP and NDN protocols is required to guarantee a flawless migration. We previously reported in [13] that there are several alternative approaches for such a migration: a dual-stack approach that serves both protocols; a hybrid approach that employs address translation; and a translation approach that employs gateways capable of communicating between the two stacks. These gateways operate at various layers of the communication stack. We are particularly interested in translating gateways that execute network-level translation in order to allow any IP traffic to traverse an NDN domain.

A network protocol migration needs a substantial upfront capital expense in the form of hardware purchases. Due to the coexistence of the IP and NDN protocols, network operators are required to convert all current routers in their service area to dual-protocol routers, resulting in an excessive increase in the investment budget. On the other hand, during the transition time, the number of users of the new protocol, such as NDN, is considerably fewer than the number of IP users. In this situation, minimizing hardware replacements is more realistic and cost-effective. As pointed out previously in [13], the translation gateway approach needs fewer hardware upgrades than others. This is because the required number of translation gateways deployed at the border of the IP and NDN networks seems to be one. As a result, the dual-channel translation gateway can significantly minimize the cost of network hardware upgrades.

The most general protocol migration solutions, such as the encapsulation gateway, operate exclusively at the application layer, resulting in a considerable packet size overhead. Additionally, the unencrypted prefix name encapsulated inside the IP payload may pose a security risk to a network, such as a man-in-the-middle attack. Another attractive approach, such as partial migration and hybrid-ICN (hICN), can convert ICN behavior into an IP packet header by concatenating the IP address and port number. However, the length of the hICN prefix name is constrained by the length of the IP address and port number, imposing a significant challenge when attempting to give longer prefix names. Thus, in this paper, we propose a dual-channel IP-to-NDN translation gateway that provides a flexible length for prefix-name binding to overcome this issue.[1]

The dual-channel IP-to-NDN translation gateway partitions channels into two packet types: interest and data. These channels can be used by the gateway to identify IP packets designated as interest packets or data packets at the network layer. There is an asymmetric name resolution table that includes a naming service for associating an IP address with a flexible-length prefix name. The table is composed of two different tables: a producer table and a consumer table. As a result, the dual-channel gateway enables IP semantics to be upgraded to NDN at the network layer so that cross-platform networking can be done effectively.

Deployment of the proposed dual-channel gateway should consider various network schemes, such as static prefix-name binding for small-scale networks and dynamic prefix-name binding for large-scale networks. Additionally, the relationship between the translation gateway component and the throughput must be thoroughly explored to comprehend the essential translation parameters. Thus, this paper explores the throughput of a dual-channel IP-to-NDN translation gateway using two different prefix-name binding schemes, namely static and dynamic. We evaluate a throughput-estimation model to determine the relationship between cache size, delay, and throughput. Moreover, we compare the model to an emulator to verify the accuracy and consistency of the model.

The contribution of the paper can be summarized as follows.

1) We propose a dual-channel IP-to-NDN translation gateway for a cross-platform communication protocol between IP and NDN that preserves the privacy of users.
2) We emulate the practical deployment of the dual-channel IP-to-NDN translation gateway using static and dynamic prefix-name binding schemes.
3) A throughput-estimation model based on deployment of the dual-channel IP-to-NDN translation gateway is introduced.
4) An analysis of an experimental throughput comparison between the estimation model and an emulation testbed is discussed.

The rest of the paper is organized as follows. The next section describes previous related works. Section III describes the dual-channel gateway principles, components, packet flow translations, binding schemes, security, and migration cost. Section IV discusses the throughput-estimation model. Section V reveals the numerical evaluation results. Section VI summarizes our findings.

---

[1] A shorter version of this manuscript was presented in [14].

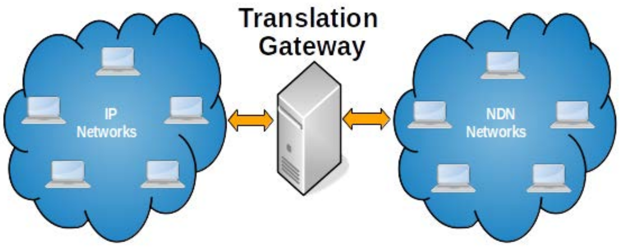
II. RELATED WORK

Many researchers have written about and recommended IP and ICN/NDN co-existence. An extreme suggestion is to redevelop all IP applications with NDN-friendly protocols from zero, which requires an enormous amount of cost and time. VoCCN [36], ACT [43], and DASC [40] are examples of such applications. While redesigning all TCP/IP-based applications from scratch is possible, it is impractical from a cost-time perspective. Encapsulation technology enables simple migration between network protocols. This deployment is offered in [9] as an immediate experimental solution for enabling ICN/NDN via IP protocol. However, this approach suffers from a packet size overhead, which is especially noticeable for longer prefix names. Additionally, it is considered an application layer migration rather than a network layer migration which is not comparable. A more realistic and comparable alternative is to use translation technology to transfer packets from the IP protocol to the NDN protocol.

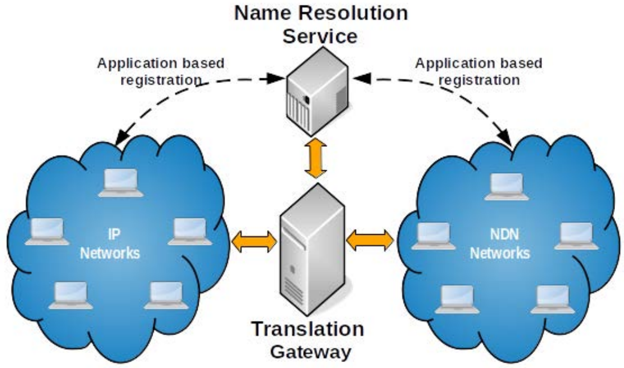
A translation strategy has already been adopted by Carofiglio *et al.* in hICN [15] that integrates an ICN/NDN packet inside the IP stack. hICN reuses packet formats deriving the ICN/NDN prefix name from a combination of IP and port headers. It requires minimal software updates and ensures backward compatibility with IP networks. However, it demands a pair of hICN routers to extract the hidden NDN packet from the IP packet as a minimum deployment. Moreover, the hICN limits the length of prefix names, which potentially causes a problem for longer and more varied prefix names. In addition, it also uses the packet flow direction to distinguish IP packets as interest or data packets, which is inefficient for recognizing whether the packet type is interest or data.

Moisennko *et al.* proposed a TCP/ICN proxy pair that can carry TCP traffic over an ICN infrastructure, namely forward proxy and reverse proxy [21]. The proxy approach managed to work correctly, but this approach focuses on the TCP handshaking mechanism, not semantic protocol translation. Similar to Moiseenko, Rafaei *et al.* in [34] did translation by using a gateway that consists of four components: a gateway daemon for accepting packets, gateway configuration file for naming and configuring packets, traffic handler (TH) for forwarding packets, and command line interface (CLI) for monitoring. However, each IP packet translation process in [34] is based on a full matched-packet filter that requires extra regex computation as overhead. Moreover, this strategy shows potential security breaches such as data privacy violations because it reads the data payload to construct a prefix name and bind a packet.

Luo *et al.* [32] proposed methods for translating TCP/IP-based packets into NDN packets at three different layers: internet, TCP, and application. The main idea is to name IP packets by using a unique hierarchical naming scheme. By simulating methods using Content-Centric Networking (CCNx) for FTP servers and clients, the authors showed the feasibility of multi-level IP/NDN translation and migration. However, the translation is not free from drawbacks. One is the usage of an interest packet to report the causes of the overhead mechanism to the NDN protocol. Another is the enormous size of CS for pre-storing the data.



(a) Static prefix-name binding.



(b) Dynamic prefix-name binding.

FIGURE 1. Dual-channel IP-to-NDN translation gateway in minimal set-up.

Compared to the mentioned-previous works, the dual-channel IP-to-NDN translation gateway improves the translation strategy by translating the packet semantics in the network layer. Hence, the gateway can save more time than rewriting IP to NDN application as proposed in [36], [40], [43]. The gateway can promote co-existence between IP and NDN protocol in the network instead of overlay communication that is emerged in the tunneling approach [9] or partial migration [15]. Furthermore, the gateway can preserve the data privacy by avoiding reading IP data packets to generate a prefix name as in [21] and [34]. Finally, the gateway eliminates an extra invocation such as the special interest for translation as mentioned in [32]. The dual-channel translation gateway focuses on building a translation gateway that obeys the driven-pull semantics in the NDN family protocol with a reusable infrastructure and minimal upgrading overhead.

III. DUAL-CHANNEL TRANSLATION GATEWAY

This section describes the principles, crucial components, packet flows of the proposed dual-channel IP-to-NDN translation gateway, the security improvement that can be achieved, and the estimation of migration cost. We propose two possible deployment schemes for prefix-name binding: static prefix-name binding and dynamic prefix-name binding.

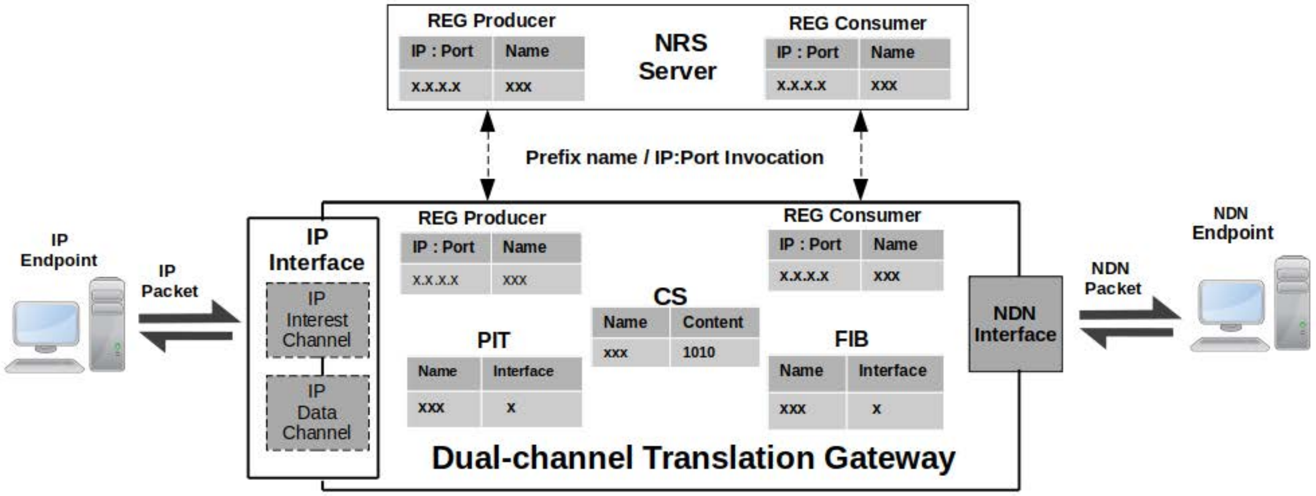


FIGURE 2. Components of dual-channel IP-to-NDN translation gateway.

A. PRINCIPLES

The fundamental concept of the dual-channel IP-to-NDN translation gateway is based on two distinct channels and a flexible-length packet naming strategy. The gateway assigns each incoming or outgoing IP packet to an allocated unique IP address. One unique IP address is associated with a corresponding interest packet type, while the other is associated with a corresponding data packet type. When an IP packet is received via the unique IP address attached to the interest type, the gateway immediately recognizes it as an interest packet from the IP consumer. If, however, it is received over the IP address attached to the data type, the gateway will immediately recognize it as a data packet from the IP producer.

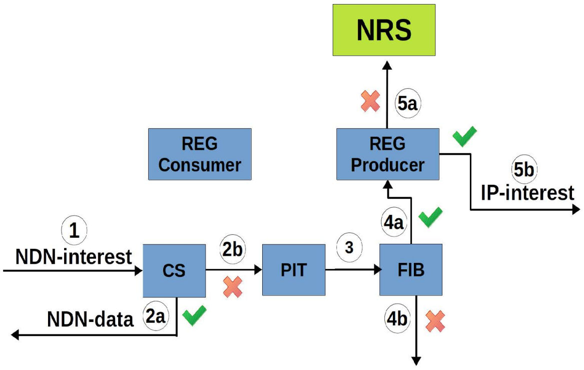
Regarding the variable length of the packet naming strategy, a table called a register table (REG) is required in the dual-channel gateway so that each IP packet in the gateway is entitled to a prefix name. Since two distinct situations of producer and consumer exist between IP and NDN, the gateway must have two asymmetric REG tables, namely REG producer and REG consumer. When an IP endpoint acts as a consumer, the REG consumer table is utilized to obtain a specific prefix name. However, when an IP endpoint acts as a producer, the REG producer table is used instead. Furthermore, we consider the independence of the REG producer and REG consumer tables because an IP endpoint may attach to a different prefix name when it is regarded as a producer or consumer in the IP networks.

The gateway registers all the prefix names in the static prefix-name binding in advance into the REG table. Therefore, when the gateway cannot resolve the prefix name from a given IP address, it drops immediately without allowing the client to modify the REG database on the fly. We assume that this is suitable for small-scale network deployment as seen in Figure 1(a). On the other hand, in dynamic prefix-name binding, the gateway utilizes an outer name server that feeds the gateway REG table with a set of associations of prefix names and IP addresses when requested. We use the name resolution server (NRS), proposed in [19] and [22], to provide the packet-to-name binding service for the dual-channel translation gateway. Due to limited storage capacity, the gateway partially stores a set of associations. However, the NRS must preserve all prefix names in the set of associations in the network. It can alter the set of associations when a client in IP or NDN networks registers a new prefix name to bind with a particular IP address through an application-based service. This is called dynamic prefix-name binding, as seen in Figure 1(b).

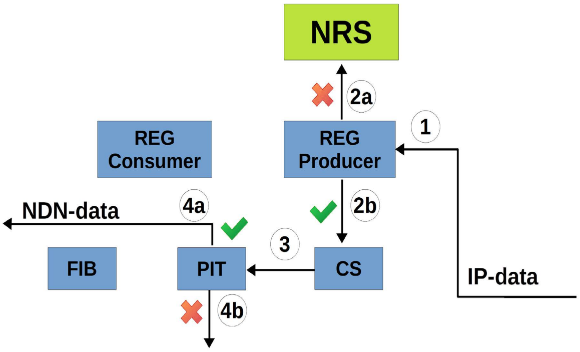
B. COMPONENTS OF DUAL-CHANNEL GATEWAY

The important components of the dual-channel IP-to-NDN translation gateway illustrated in Figure 2 can be explained as follows.

- **IP interest channel**
  The gateway utilizes a static IP address to recognize IP interest-like packets between the IP endpoint and the gateway. The static IP address used as the interest channel is known in advance by all IP endpoints in the network.
- **IP data channel**
  The gateway has a dedicated static IP address for sending and receiving IP data packets. All IP endpoints in the network recognize the static IP address chosen as the data channel in advance as well.
- **NDN interface**
  The NDN interface is constructed using layer two-socket interfaces. The NDN packet format follows the NDN packet v.3 specifications.

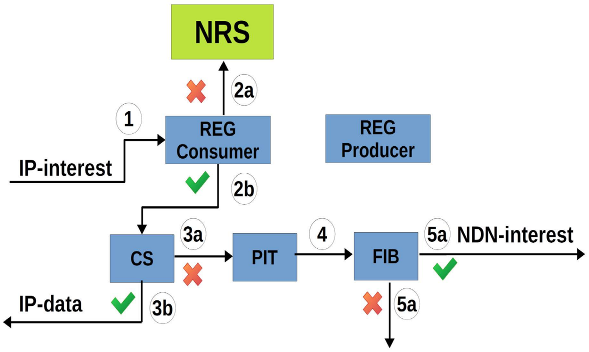


(a) NDN interest to IP interest packet flow.

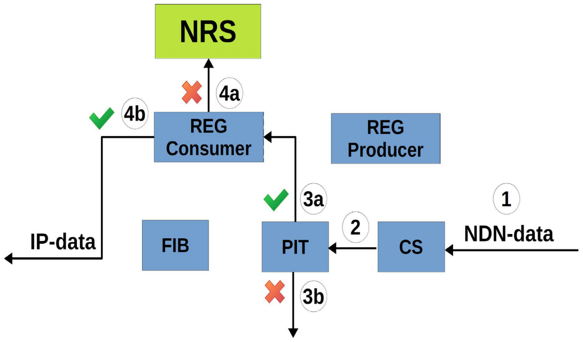


(b) IP data to NDN data packet flow.

FIGURE 3. Packet flow translation between IP and NDN protocol in the case of scenario 1.



(a) IP interest to NDN interest packet flow.



(b) NDN data to IP data packet flow.

FIGURE 4. Packet flow translation between IP and NDN protocol in the case of scenario 2.

- **REG**
  REG is used to provide information about the prefix name associated with the combination of IP address and port number. It consists of two different independent tables, namely a producer table and consumer table.
- **PIT**
  PIT is a native NDN router component that is used as a pool for collecting information regarding the incoming interfaces of interest and its corresponding prefix name. If the requested prefix name content is replied to by the producer, the PIT table will eliminate the prefix name and its associated interfaces.
- **FIB**
  FIB is used for forwarding an interest packet to its destination by retrieving interfaces that are associated with its particular prefix name. Afterward, the router forwards the interest packet through those interfaces.
- **CS**
  CS has a function to store temporally the received data content that is passed through the NDN router from a producer. The cached data content is forwarded later on to the consumer that requested the same prefix name.

C. PACKET FLOWS AND TRANSLATION PROCEDURE

This section describes the IP-to-NDN packet translation flow. The packet flow of the gateway differs in both the producer and consumer configurations. The packet flow in the case of the IP endpoint becoming a producer and NDN endpoint becoming a consumer, denoted as *scenario 1*, is shown in Figure 3. Alternatively, Figure 4 shows the packet translation flow in the case of a scenario where an NDN endpoint becomes a producer, and an IP endpoint becomes a consumer, denoted as *scenario 2*.

In the case of scenario 1, when a gateway receives an interest packet (1), it checks its CS for that packet availability, whether the content exists or not. The gateway responds with an NDN data packet if it is available (2a). If the requested content does not exist in the CS, the interest packet is passed to PIT, which registers the incoming interface (2b), and the gateway checks the forwarding interface at FIB (3). FIB drops the packet if there is no information about the prefix name being searched for (4b). When the outgoing interface is discovered, it is passed to the REG producer table to determine the corresponding IP address (4a). The gateway sends a resolving packet to NRS and waits for a response because the REG producer table is likewise restricted (5a). The IP interest packet is then transmitted across the IP channel (5b) as seen in Figure 3(a). Figure 3(b) shows the processing flow at the gateway when the data packet arrives at the gateway from an IP provider. When an IP data packet sent from an IP producer arrives at the gateway, the gateway checks the IP address and port number combination in the REG producer table (1). If it is not found, the gateway sends a resolving packet to NRS (2a). It then forwards the data packet through the interface (4a) before it is found in the PIT table (3) and stored in CS (2b).

Algorithm 1 Dual-Channel Packet Translation on IP Interface

```
Input: IP packet
Output: IP packet or NDN packet
LOOP Process
if IP interest channel receives packet then
    Retrieve prefix name in REG Consumer
    while Prefix name is not in REG Consumer do
        Request update and wait response to NRS server
    end while
    if Content is available in CS then
        Retrieve content and send via IP data channel
    else
        Name-based routing (Algorithm 3) via NDN channel
    end if
end if
if IP data channel receives packet then
    Retrieve prefix name in REG Producer
    while Prefix name is not in REG Producer do
        Request update and wait response to NRS server
    end while
    Content caching (Algorithm 4) via NDN channel
end if
```

Algorithm 2 Dual-Channel Packet Translation on NDN Interface

```
Input: NDN packet
Output: NDN packet or IP packet
LOOP Process
if NDN channel receives NDN interest packet then
    if Content is available in CS then
        Retrieve content and send via NDN channel
    else
        Retrieve IP address and port in REG Producer
        while IP address and port is not in REG Producer do
            Request update and wait response to NRS server
        end while
        Name-based routing (Algorithm 3) via IP interest channel
    end if
end if
if  NDN channel receives NDN data packet then
    Retrieve IP address and port in REG Consumer
    while IP address and port is not in REG Consumer do
        Request update and wait response to NRS server
    end while
    Content caching (Algorithm 4) via IP data channel
end if
```

Algorithm 3 Name-Based Routing

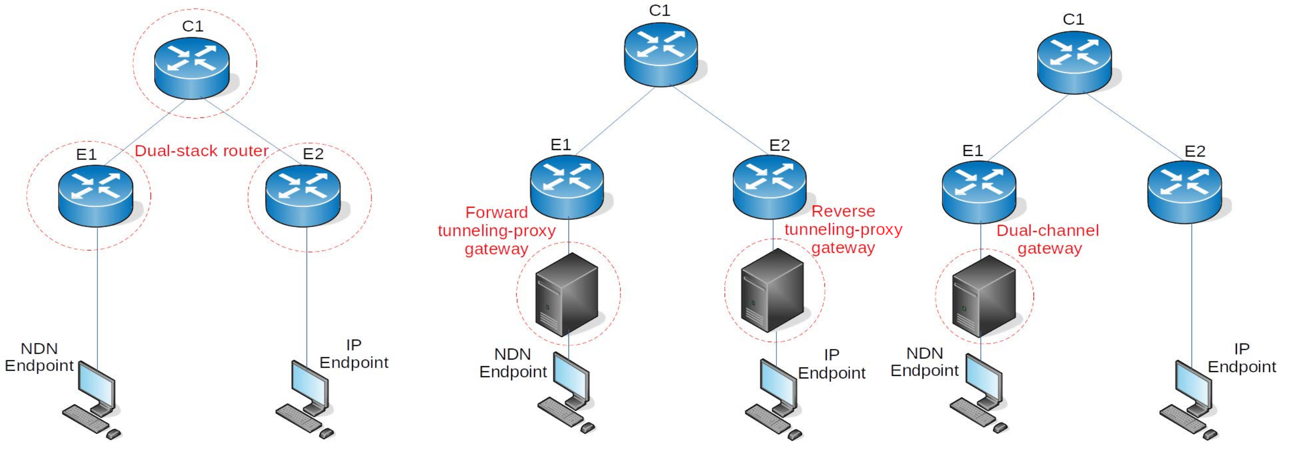
```
Input: Interest packet
Output: Interest packet
Register incoming interface in PIT
Retrieve outgoing interface in FIB
if Prefix name is not in FIB then
    Drop interest packet
else
    Forward interest packet
end if
```

Algorithm 4 Content Caching

```
Input: Data packet
Output: Data packet
Store data content to CS
Retrieve outgoing interface in PIT
if Prefix name is not in PIT then
    Drop data packet
else
    Delete PIT entry
    Forward data packet
end if
```

The packet flow translation in the case of scenario 2 is quite similar to scenario 1, except that the interest packets are first checked in the REG consumer table to determine the associated prefix name (1). When an IP packet is received from NRS, it is converted to an NDN packet and then checked for availability in the CS (2b). If it is not found, the packet is sent to the NDN interface (5a) that was previously found in the FIB (4) and registered in the PIT incoming interface (3a) as seen in Figure 4(a). When the data packet has arrived at the gateway, as shown in Figure 4(b), the gateway stores the content in CS (1) and verifies the outgoing interface in PIT (2) when the NDN producer delivers the NDN data packet. The prefix name is used to retrieve the related IP address in the REG consumer table because the outgoing interface is an IP interface. In the event that it is not found, the gateway sends a REG resolving packet to NRS server (4a). The data content is transformed into an IP packet after receiving a response from the NRS server. The data packet is subsequently forwarded across the IP data channel (4b).

The dual-channel translation gateway operates on two network interfaces that continuously listen for incoming packets. Algorithm 1 illustrates the pseudo-code for the dual-channel gateway translation procedure on the IP interface. Alternatively, Algorithm 2 describes the pseudo-code for the translation procedure on the NDN interface. We define two native procedures of ICN/NDN routers, namely named-based routing and content caching, to simplify the instructions of Algorithm 1 and 2. Name-based routing forwards the interest packet to the producer as shown in Algorithm 3. On the other hand, content caching stores the data packet to CS and forwards it to the consumer as described in Algorithm 4. In the



(a) Dual-stack router. **(b)** Tunneling-proxy gateway. **(c)** Dual-channel gateway.

FIGURE 5. **IP and NDN co-existance technology.**

initialization setup, two unique IP addresses are configured to the IP interface: IP interest and data channel, while the NDN interface is attached to a specific MAC address. When an IP interface receives an incoming packet, the gateway examines if it is of interest or a particular data type. The NDN interface is likewise subjected to the exact mechanism. Assume that the NDN interface receives packets. The gateway verifies the specific type of packet and applies infinite looping to keep the process still operating on both network interfaces.

D. USER PRIVACY AND SECURITY

The NDN technology uses a prefix name in its interest or data packets. A prefix name is written out in plaintext so that a consumer can easily identify the name of the content. The content in the data packet can be signed and encrypted by exchanging keys between producer and consumer. On the other hand, IP protocol has source and destination addresses in its packets that propagate through routers until they reach their final destination. Unless the data is encrypted at the application layer, IP data packets are usually sent in plain text form. It is challenging to maintain security concerns across those protocols due to the distinct behavior of exposed packet components.

The gateway relies on data packet load to transport other protocols from node to node in the encapsulation approach. The NDN packet is placed inside the IP data load in the case of NDN over IP. The unencrypted data payload may disclose both the NDN prefix name and the content within the IP network. When the packet propagates from router to router, the interceptor can easily determine the prefix name and the content. The attacker may also modify the content resulting in a data integrity breach called a Content Poisoning Attack (CPA). The CPA has been reported as the main cause of the Denial of Service (DoS) attack in the NDN network [30].

The dual-channel gateway can lower the risk of a CPA attack by separating the prefix name and the data content. The prefix name is located in the REG table in the gateway linked to one of the IP headers, either the destination or source address. The propagated IP packet does not expose the prefix name inside the IP data payload. Therefore, the CPA can be reduced in the dual-channel gateway. At the same time, it also guarantees user privacy in the translation process at the gateway. This is because the gateway does not require reading the IP data payload content bit by bit. The translation mechanism in the dual-channel only necessitates the IP address header reading.

TABLE 1. **Symbols related to cost metrics.**

Symbol	Description
C_s	Software upgrade cost
C_h	Hardware upgrade cost
C_r	Hardware replacement cost
C_v	Vendor support cost
C_d	Development cost
C_m	Miscellaneous cost
ξ_a	Decision coefficient of software upgrade
ξ_b	Decision coefficient of hardware upgrade
$\neg\xi_a$	Negation of software upgrade decision coefficient

E. MIGRATION COST

The expense of new hardware purchases should be minimal to avoid up-front capital investment due to migration. On the other hand, the total cost of migration depends on the cost of the individual components, as reported in [8]. The authors of [8] described that the total estimation of migration cost, C_t, for L, the number of the routers, was formulated by

$$C_t = \sum_{i=1}^{L} \{\xi_a(C_s+\xi_b C_h)+(\neg\xi_a)C_r+C_v+C_d+C_m\}, \tag{1}$$

where the symbols of cost metrics are described in Table 1.

The symbol ξ_a and ξ_b have Boolean values of either 1 (True) or 0 (False). If the value ξ_b equals 1, then a router

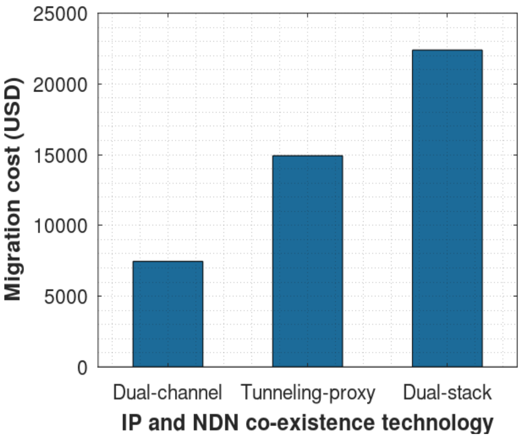


FIGURE 6. Migration cost in IP and NDN co-existence technology.

is upgraded. While ξ_b equals 0, then it means otherwise. However, the necessity of a router upgrade is influenced by hardware peripherals such as memory, network interface, and processor speed. Under certain conditions, it is impossible to upgrade the hardware without a software upgrade. Thus, if ξ_b equals 1, then ξ_a is likewise. On the other hand, the hardware upgrade is non-compulsory following the software upgrade. Hence, ξ_b can be either 0 or 1, although ξ_a is equal to 1. Furthermore, the router replacement means the investment of new hardware with the latest pre-installed software. Therefore, the equation 1 has symbol $\neg\xi_a$ multiplied by C_r which is the negation from ξ_a multiplied by C_s and C_h. This is because the router replacement will automatically nullify the hardware or software upgrade and vice versa.

TABLE 2. Average migration cost of individual components.

Description	Cost
Software upgrade cost	$ 275
Hardware upgrade cost	$ 700
Hardware replacement cost	$ 7000
Vendor support cost	$ 140
Development cost	$ 250
Miscellaneous cost	$ 75

To compare migration costs on different IP and NDN co-existence technology, we consider a simple network topology consisting of three routers, namely $C1$, $E1$, and $E2$, as shown in Figure 5. Afterward, we evaluate the total migration cost using the average migration cost of individual components in [8], as described in Table 2. The result shows that the migration cost of the dual-stack router is the most expensive, followed by the tunneling-proxy and dual-channel gateway, as seen in Figure 6. This is because the dual-stack router deployment demands total router replacements. On the other hand, the tunneling-proxy gateway necessitates two additional hardware installations on each different network protocol, while the dual-channel gateway requires only one additional hardware on the border of the NDN network.

IV. THROUGHPUT ANALYSIS

Throughput is defined as the average number of data bits transmitted in a second. Thus, we define throughput as the average packet size divided by τ, where τ is the average time of a request that has elapsed in the gateway. As shown in Figure 3 and 4, the value of τ depends on the schemes, scenarios, and cases of packet flow at the gateway. In the case of dynamic prefix-name binding scheme, scenario 1 has five different cases of packet flow, whereas scenario 2 has six different cases. This is because the combination of CS and REG affects the number of packet flow cases. However, in the case of static prefix-name binding, the number of packet flow cases is reduced to 2 for both scenarios 1 and 2. This is because only the CS component affects the packet flow translation.

The dynamic prefix-name binding scheme allows clients to dynamically register prefix names through a name server. Therefore, the NRS contains all prefix-name bindings in the network, while the translation gateway partially stores the prefix-name bindings in a cache due to its limited memory capacity. A cache miss in the translation gateway will cause a request to be made to the NRS for a target prefix name. As a result, in addition to β, there is a supplemental parameter for a prefix-name binding cache called γ. We introduced $\gamma_{p,1}$ as the hit ratio of the REG producer table (REGp) and $\gamma_{c,1}$ as the hit ratio of the REG consumer table (REGc) in the case of requesting IP address by invoking a prefix name. Alternatively, $\gamma_{p,2}$ is the hit ratio of the Reverse REG producer table (RREGp), and $\gamma_{c,2}$ is the hit ratio of the Reverse REG consumer table (RREGc) in the case of requesting a prefix name by invoking an IP address. Table 3 shows a description of delay components in the gateway used in the throughput model.

TABLE 3. Definition of delay components.

Symbol	Description
t_c	Positive look-up time in CS
t_{cn}	Negative look-up time in CS
t_{reg1}	Positive look-up time in REGp or REGc
$t_{reg1'}$	Negative look-up time in REGp or REGc
t_{reg2}	Positive look-up time in RREGp or RREGc
$t_{reg2'}$	Negative look-up time in RREGp or RREGc
t_{i_p}	NDN interest packet parsing time
t_{i_s}	NDN interest packet serialization time
t_{d_p}	NDN data packet parsing time
t_{d_s}	NDN data packet serialization time
t_{ip_c}	IP packet construction time
t_{ip_d}	IP packet deconstruction time
t_{pr1}	Response time from IP producer
t_{pr2}	Response time from NDN producer
t_{soh}	Socket overhead time
t_{nrs}	NRS resolving round trip time

A. STATIC PREFIX-NAME BINDING

When static prefix-name binding is implemented in the network, the hit ratio of CS is the dominant parameter that affects the overall throughput. As for scenario 1, the value of τ can be described by

$$\tau = \begin{cases} ts_{1a} & \text{if CS hit} \\ ts_{1b} & \text{if CS miss} \end{cases}$$

where

- $\mathrm{ts}_{1a} = t_{i_p} + t_c + t_{soh}$,
- $\mathrm{ts}_{1b} = t_{cn} + t_{i_p} + t_{ip_c} + t_{ip_d} + t_{d_s} + t_{pr1} + 2t_{soh}$.

However, in the case of scenario 2, the value of τ can be defined by

$$\tau = \begin{cases} ts_{2a} & \text{if CS hit} \\ ts_{2b} & \text{if CS miss} \end{cases}$$

where

- $\mathrm{ts}_{2a} = t_c + t_r + t_{i_p} + t_{ip_c} + t_{ip_d} + t_{d_p} + t_{soh}$,
- $\mathrm{ts}_{2b} = t_{cn} + t_r + t_{i_s} + t_{ip_c} + t_{ip_d} + t_{d_p} + t_{pr2} + 2t_{soh}$.

Thus, the average processing time, τ, in the case of scenario 1 can be expressed by

$$\tau = \beta \mathrm{ts}_{1a} + (1 - \beta)\mathrm{ts}_{1b}, \tag{2}$$

and for scenario 2, τ can be given by

$$\tau = \beta \mathrm{ts}_{2a} + (1 - \beta)\mathrm{ts}_{2b}, \tag{3}$$

where β is the CS hit ratio.

The value of β can be approximated by calculating the hit ratio of each prefix name i, $\beta(i)$, with its distribution, $q(i)$, for the total number of unique prefix names, M. The β can be expressed by

$$\beta = \sum_{i=1}^{M} q(i)\beta(i). \tag{4}$$

The approximation of the hit ratio can be calculated by using Che's approximation, which was originally proposed by Che *et al.* [17]. Then, the hit ratio for a particular prefix name $\beta(i)$ can be estimated by the following

$$\beta(i) = 1 - e^{-q(i)t_c}. \tag{5}$$

$q(i)$ is the ratio of requests for content i, and t_c is the unique root of the equation

$$C = \sum_{i=1}^{M} (1 - e^{-q(i)t_c}), \tag{6}$$

where C is the storage capacity of CS since we use LRU as a replacement algorithm in our translation gateway.

B. DYNAMIC PREFIX-NAME BINDING

When dynamic prefix-name binding is implemented in the network, the overall throughput are collaboration from the hit ratio of CS, REGp, RREGp in the scenario 1 and CS, REGc, RREGc in the scenario 2. Therefore, in the case of scenario 1, the value of τ can be described by

$$\tau = \begin{cases} td_{1a} & \text{if CS hit} \\ td_{1b} & \text{if CS miss, REGp hit, and RREGp hit} \\ td_{1c} & \text{if CS miss, REGp hit, and RREGp miss} \\ td_{1d} & \text{if CS miss, REGp miss, and RREGp hit} \\ td_{1e} & \text{if CS miss, REGp miss, and RREGp miss} \end{cases}$$

where

- $\mathrm{td}_{1a} = t_{i_p} + t_c + t_{soh}$,
- $\mathrm{td}_{1b} = t_{i_p} + t_{cn} + t_{reg1} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2} + 2t_{soh}$,
- $\mathrm{td}_{1c} = t_{i_p} + t_{cn} + t_{reg1} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2'} + t_{nrs} + 3t_{soh}$,
- $\mathrm{td}_{1d} = t_{i_p} + t_{cn} + t_{reg1'} + t_{nrs} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2} + 3t_{soh}$,
- $\mathrm{td}_{1e} = t_{i_p} + t_{cn} + t_{reg1'} + t_{nrs} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2'} + t_{nrs} + 4t_{soh}$.

The average time for processing a packet, τ, can be obtained by

$$\begin{aligned} \tau = {} & \beta \mathrm{td}_{1a} + (1 - \beta)\gamma_{p,1}\gamma_{p,2}\mathrm{td}_{1b} \\ & + (1-\beta)\gamma_{p,1}(1-\gamma_{p,2})\mathrm{td}_{1c} + (1-\beta)(1 - \gamma_{p,1})\gamma_{p,2}\mathrm{td}_{1d} \\ & + (1 - \beta)(1 - \gamma_{p,1})(1 - \gamma_{p,2})\mathrm{td}_{1e}. \end{aligned} \tag{7}$$

However, in the case of scenario 2, the value of τ can be defined by

$$\tau = \begin{cases} td_{2a} & \text{if CS hit, RREGc hit} \\ td_{2b} & \text{if CS hit, RREGc miss} \\ td_{2c} & \text{if CS miss, REGc hit, and RREGc hit} \\ td_{2d} & \text{if CS miss, REGc hit, and RREGc miss} \\ td_{2e} & \text{if CS miss, REGc miss, and RREGc hit} \\ td_{2f} & \text{if CS miss, REGc miss, and RREGc miss} \end{cases}$$

where

- $\mathrm{td}_{2a} = t_{ip_d} + t_{reg2} + t_c + t_{d_p} + t_{ip_c} + t_{soh}$,
- $\mathrm{td}_{2b} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_c + t_{d_p} + t_{ip_c} + 2t_{soh}$,
- $\mathrm{td}_{2c} = t_{ip_d} + t_{reg2} + t_{cn} + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 2t_{soh}$,
- $\mathrm{td}_{2d} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_{cn} + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 3t_{soh}$,
- $\mathrm{td}_{2e} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_{cn} + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 3t_{soh}$,
- $\mathrm{td}_{2f} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_{cn} + t_{i_s} + t_{d_p} + t_{nrs} + t_{pr2} + t_{reg1'} + t_{ip_c} + 4t_{soh}$.

Therefore, the average packet processing time, τ, for scenario 2 can be expressed by the following

$$\begin{aligned} \tau = {} & \beta\gamma_{c,2}\mathrm{td}_{2a} + \beta(1 - \gamma_{c,2})\mathrm{td}_{2b} + (1 - \beta)\gamma_{c,1}\gamma_{c,2}\mathrm{td}_{2c} \\ & + (1-\beta)\gamma_{c,1}(1-\gamma_{c,2})\mathrm{td}_{2d} + (1-\beta)(1 - \gamma_{c,1})\gamma_{c,2}\mathrm{td}_{2e} \\ & + (1 - \beta)(1 - \gamma_{c,1})(1 - \gamma_{c,2})\mathrm{td}_{2f}. \end{aligned} \tag{8}$$

The value of the hit ratio in CS, β, can be calculated obeying the same equation (4) for static prefix-name binding with its associated parameters such as the storage capacity of CS, C, the ratio of requests for content i sent by the consumer, $q(i)$, and the unique root of an equation, t_c. However, the value of the hit ratio in the REG producer or consumer table may differ due to the packet flow behavior in the translation process. As for scenario 1, the values of $\gamma_{p,1}$ and $\gamma_{p,2}$ are highly correlated with the CS miss ratio, (1-β). Therefore, an identical formula can be used to estimate both $\gamma_{p,1}$ and $\gamma_{p,2}$. First, we introduce the ratio of uncached content i, $r(i)$ as the product from the miss ratio of CS, (1-β), which can be approximated by the following

$$r(i) = q(i)(1 - \beta(i)). \tag{9}$$

By using equation (6), the unique root, t_r, for the REGp table can be obtained by replacing the size of CS, C, with the size of REG, R. Finally, the hit ratio of a particular prefix name i in REGp and RREGp, $\gamma_{p,1}(i)$ and $\gamma_{p,2}(i)$, must be adjusted by a normalization factor as described by

$$\gamma_{p,1}(i) = \gamma_{p,2}(i) = \frac{1}{1-\beta}(1 - e^{-r(i)t_r}). \tag{10}$$

Finally, the cumulative hit ratio of $\gamma_{p,1}(i)$ and $\gamma_{p,2}(i)$ from M total unique prefix names can be defined by the following

$$\gamma_{p,1} = \gamma_{p,2} = \sum_{i=1}^{M} r(i)\gamma_{p,1}(i). \tag{11}$$

As for scenario 2, the value of $\gamma_{c,1}$ is correlated with the CS miss ratio, (1-β), so it follows equations (9), (10), (11) consecutively. However, the value of $\gamma_{p,2}$ is not affected by the CS miss ratio, (1-β), so it follows the steps of equation (4), (5), (6) used similarly to find β. This is because the distribution of a prefix name requested in RREGc and CS is identical to $q(i)$.

V. NUMERICAL RESULTS

The emulation testbed was constructed following the topology as shown in Figure 1(a) in the case of static prefix-name binding and Figure 1(b) in the case of dynamic prefix-name binding. Virtual box and python 3.7 were used in the emulation testbed by utilizing the external library of NDN packet v.3 in [38].

The interest packets were generated asynchronously, about ten thousand packets per second, to measure the throughput. The distribution of requested content followed a Zipf distribution with skewed parameter α due to some studies reporting the request distribution of various types of digital content. Websites and user-generated videos followed the Zipf distribution as reported in [25] and [27]. Breslau *et al.* said that the request count of webpages obeyed the Zipf distribution with a parameter α between 0.64 and 0.83 in [25]. Moreover, Mahanti *et al.* showed that it was between 0.74 and 0.84 in [2]. The request count of YouTube videos obeyed the Zipf distribution with a parameter α about 0.8 in [27]. Therefore, we assumed that α was between 0.7 and 1.5 in our testbed to cover all possible request distribution regarding content popularity on the internet.

We captured the data log from our emulator testbed about fifty thousand packets as a data sample for time evaluation. We repeated the data collection process at random about ten times to ensure that the data was consistent and statistically compliant. Additionally, we took the data sample every second within five minutes from the data log to evaluate the throughput.

A. STATIC PREFIX-NAME BINDING

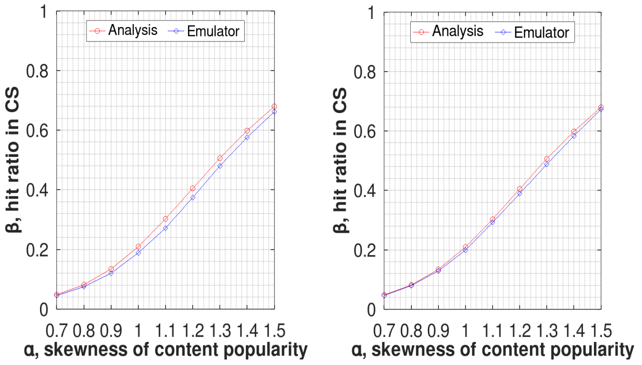
The major parameter set-up in the case of static prefix-name binding follows the parameter values as shown in Table 4.

TABLE 4. Setting values of main parameters in static prefix-name binding scheme.

Paramater	Value
Number of content items (M)	1000
Cache-size (C)	10, 100
Skewness of content popularity (α)	0.7 - 1.5
Interest rate	10000 interests/s

1) HIT RATIO

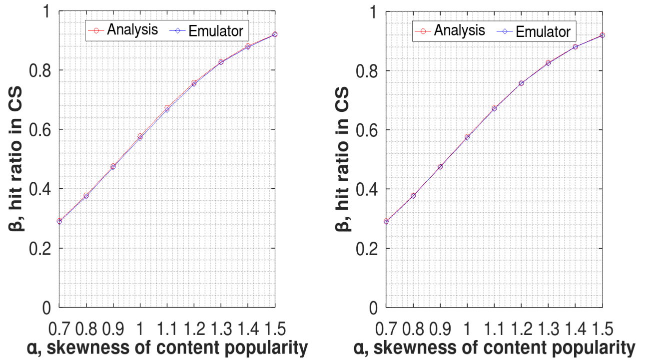
The emulation testbed values of the CS hit ratio, β, were measured and compared against the analysis model in both scenarios, and there were also different CS sizes. Figure 7 shows a comparison of CS hit ratios between the analysis and emulation testbeds when the CS size was equal to 10 content items. Alternatively, Figure 8 illustrates a comparison of CS hit ratios between the analysis and emulation testbeds when the CS size is equal to 10 content items. The analysis hit ratio, β, is obtained from equations (5) and (6).



(a) Scenario 1. (b) Scenario 2.

FIGURE 7. Comparison of hit ratios of CS when CS capacity is equal to 1% from total number of prefix names in the case of static prefix-name binding scheme.

The hit ratio of CS, β, was constantly increased from 5% to 68% when the content request skewness, α increased from 0.7 to 1.5 for CS size equal to 10 out of 1000 content items in both scenarios 1 and 2. The gap between the analytical and emulation CS hit ratios, $\Delta\beta$, was about 5% on average, where the gap monotonously increased when α increased for scenario 1. In the case of scenario 2, the gap of β also followed the trend of scenario 1 with a gap about 2% smaller than scenario 1 as shown in Figure 7.



(a) Scenario 1. **(b)** Scenario 2.

FIGURE 8. Comparison of hit ratios of CS when CS capacity is equal to 10% from total number of prefix names in the case of static prefix-name binding scheme.

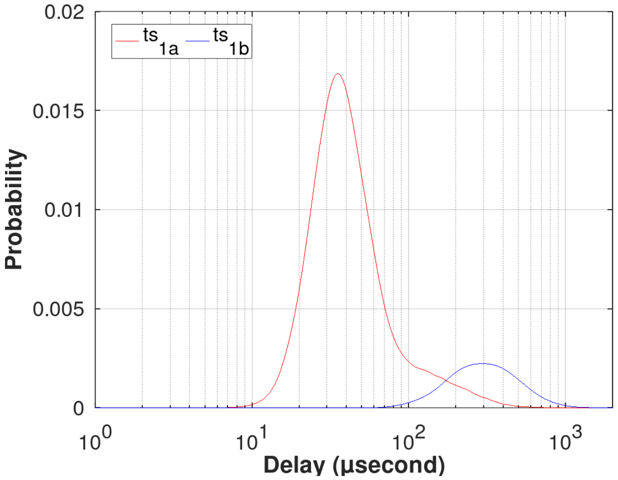


FIGURE 9. Distribution of processing time components in the case of scenario 1 and static prefix-name binding scheme.

Whenever the size of CS increased, i.e., CS equaled 100 content items, a significant rise occurred in the CS hit ratio, β, from 5% to 29% at $\alpha = 0.7$ and 68% to 92% at $\alpha = 1.5$. Moreover, the gap of β decreased impressively to about 0.1% for all axes of α in both scenarios 1 and 2 as seen in Figure 8. We assume that this phenomenon occurred because of delayed caching, which was revealed in [29].

2) PROCESSING TIME DISTRIBUTION

Processing time is essentially required to calculate the throughput. It is constructed from an accumulation of delay components in each case of packet flow in the translation process. Therefore, the processing time distribution can determine the range and deviation of each case of translation. We used two values, median and mean, taken from five hundred thousand packets in our emulation testbed experiment. Table 5 shows the median and mean value of each delay component.

Figure 9 shows the distribution of processing times ts_{1a} and ts_{1b} in the case of scenario 1. In general, the distribution of ts_{1b} is wider than ts_{1a}. This is because the number of delay components in ts_{1b} was more than the number for ts_{1a}. The accumulation of several delay components resulted in a longer processing time in ts_{1b}. The distribution of ts_{1b} was shifted to the right from ts_{1a} at a scale of 100 μs. The processing time interval of ts_{1a} reached about 1000 times between the min and max values. Moreover, the distribution skewness of ts_{1a} and ts_{1b} appeared similar, that is, skewed to the right, since the mean was slightly longer than the median of the processing time.

TABLE 5. Statistical results of delay components.

Symbol	Median	Mean	Symbol	Median	Mean
t_c	1 μs	1.1 μs	t_{d_p}	45 μs	47.4 μs
t_{cn}	1 μs	0.9 μs	t_{d_s}	130 μs	141.8 μs
t_{reg1}	1 μs	1.2 μs	t_{ip_c}	4 μs	5.5 μs
$t_{reg1'}$	1 μs	1.1 μs	t_{ip_d}	5 μs	6.5 μs
t_{reg2}	1 μs	1.2 μs	t_{pr1}	95 μs	98.5 μs
$t_{reg2'}$	1 μs	1.1 μs	t_{pr2}	111 μs	115 μs
t_{ip}	44 μs	47 μs	t_{soh}	24 μs	24.7 μs
t_{i_s}	88 μs	90.5 μs	t_{nrs}	10000 μs	10000 μs

Figure 10 shows the processing time distribution of ts_{2a} and ts_{2b} in the case of scenario 2. The distribution of ts_{2a} was centered at about 80 μs with a minimum processing time of around 10 μs and a maximum of about 1000 μs. With the smaller range, the interval of ts_{2b} was also similar to ts_{2a} with a starting point from 50 μs to 1000 μs. Moreover, its center value was shifted to the right around 170 μs from the central value of ts_{2a}. In addition, the number of delay components for ts_{2b} was more than the number for ts_{2a}. The distribution of ts_{2b} was also affected by the producer response time, which contributed to a longer accumulation of processing time.

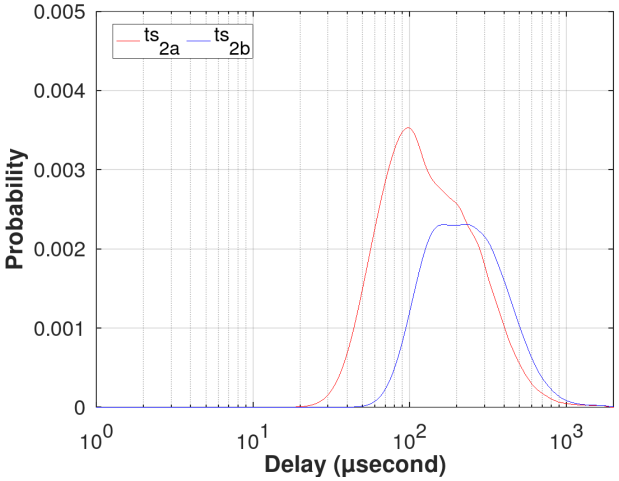
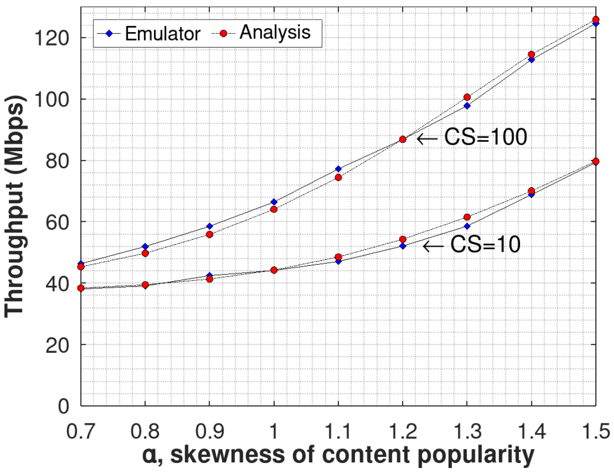


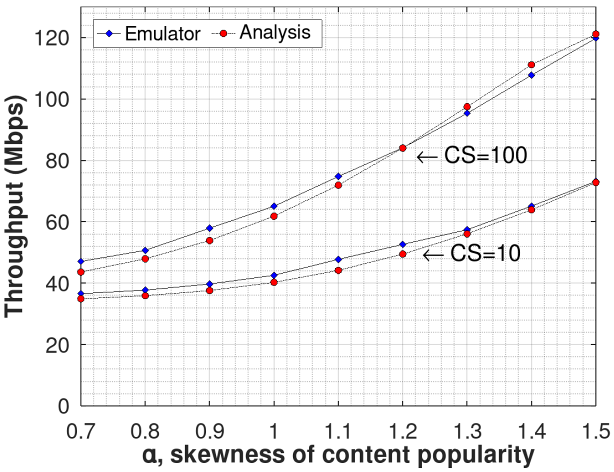
FIGURE 10. Distribution of processing time components in the case of scenario 2 and static prefix-name binding scheme.

3) THROUGHPUT

The throughput of a gateway can be determined by the number of packets sent by the gateway to a customer in a second.

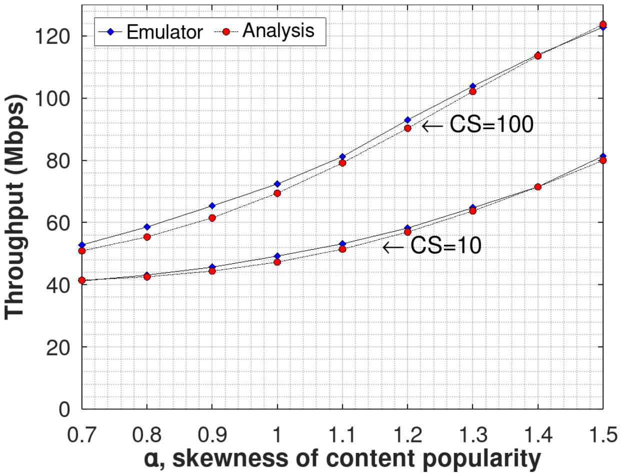


(a) Median approach.

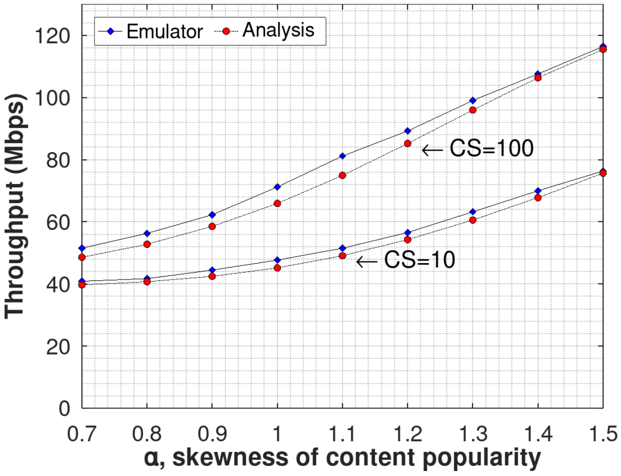


(b) Mean approach.

FIGURE 11. Throughput comparison between estimation analysis and emulation testbed when CS size was equal to 1% and 10% from total number of prefix names in the case of scenario 1 using static prefix-name binding scheme.



(a) Median approach.



(b) Mean approach.

FIGURE 12. Throughput comparison between estimation analysis and emulation testbed when CS size was equal to 1% and 10% from total number of prefix names in the case of scenario 2 using static prefix-name binding scheme.

The emulation throughput was determined by capturing each second within five minutes in the emulation testbed. Two statistical variables, namely the median and mean, were then utilized to compare the emulation and analysis throughputs. A throughput analysis was calculated by using the analytical hit ratio and the statistic of processing time taken from emulation testbed. As a result, a median and mean throughput analysis could be done with this calculation.

Figure 11 depicts a throughput comparison between the emulation and analysis in the case of scenario 1 by using the median and mean with a CS capacity equal to 1% and 10% of the total number of prefix names, respectively. The increase in α caused a constant increase in the throughput in different cases of CS sizes. This is because the consumer requested more popular content, which lead to an increase in the CS hit ratio, so the parameter β made a dominant contribution to the increase in throughput. For example, when the CS size was 10 content items, the throughput increased from about 40 Mbps at $\alpha = 0.7$ to 80 Mbps at $\alpha = 1.5$ for the median approach. However, in the mean approach, the increase in throughput was 5% less from the median. Another fact shows that an increase in CS size of about 10 times resulted in an increase in the average throughput along the α axis of about 30% on average for both the median and the mean. This is also because of the increase in the hit-ratio property β.

In terms of a comparison of emulation and analysis throughput, the difference between analysis and emulation for the median was generally lower than the mean. In the case of a CS size equal to 10 items, the gap between emulation and analysis throughput was about less than 5% for the median compared with 8% for the mean with a gap that was higher in the middle of the α axis between 0.9 and 1.4. However, it was smaller at the edge of the α axis. This is because the property of the hit ratio and processing time changed over the

α axis. When α was at the minimum value, the hit ratio gap was lower, but it widened as α increased. When α was at the minimum value, the processing time was at the maximum, but it shortened as α increased. Therefore, the contribution of the longer processing time and higher gap in hit ratio at the middle of the α axis caused a decrease in accuracy performance. However, in the case of a CS size equal to 100 items, the gap between analysis and emulation was generally higher at a lower α of about 5% and 7% in the median and mean approaches, respectively. This is because the increase in CS size affected the difference in the hit ratio between emulation and analysis, making it close to zero. Therefore, the cause of differences between the analysis and emulation was only affected by the property of processing time, where the processing time decreased when alpha increased.

Figure 12 illustrates a throughput comparison between emulation and analysis in the case of scenario 2 by using the median and mean with a CS capacity equal to 1% and 10% of the total number of prefix names, respectively. Similar to scenario 1, the increase in α caused a constant increase in the throughput for both cases of CS sizes equal to 1% and 10%. The throughput increased from about 41 Mbps at $\alpha = 0.7$ to 81 Mbps at $\alpha = 1.5$ for the median approach. However, for the mean approach, the throughput increased from about 38 Mbps at $\alpha = 0.7$ to 74 Mbps at $\alpha = 1.5$. The increase in CS size from 10 to 100 resulted in a significant increase of about 35% of average throughput along the α axis for both the median and mean. The difference between analysis and emulation for the median was also lower than the mean approach. The average deviation was about 3% for the median compared with 5% for the mean for CS sizes equal to 10 out of 1000 content items and 7% for the median compared with 10% for the mean for CS size equals to 100 out of 1000 items.

Having a similar behavior, the difference between scenarios 1 and 2 was highlighted at the cross point between analysis and emulation throughput, especially when the CS capacity was equal to 10% of the total number of prefix names. In the beginning, the analysis throughput was located below the emulation graph at an α of less than 1.2, and it exceeded the emulation graph afterward in the case of scenario 1. However, in the case of scenario 2, the analysis throughput crossed the emulation graph at an α of higher than 1.4. This is because scenario 1 had a shorter processing time at a higher alpha than scenario 2. As a result, it improved the overall analysis throughput estimation.

B. DYNAMIC PREFIX-NAME BINDING

The major parameter set-up in the case of dynamic prefix-name binding followed the parameter values shown in Table 6. In this case, the dual-channel gateway only caches a limited number of prefix names due to its limited memory capacity. The NRS that holds all associations of prefix name and IP address will immediately send a response when a request is sent by the gateway. The number of prefix names cached in the REG table should be far bigger than the number of unique content items in CS. This is because the REG table only stores a prefix name and its associated IP address, which requires less memory compared with the data content.

TABLE 6. Setting value of main parameters in dynamic prefix-name binding scheme.

Paramater	Value
Number of content items (M)	1000
Cache size (C)	10
Reg size (R)	200
Skewness of content popularity (α)	0.7 - 1.5
Interest rate	10000 interests/s

1) HIT RATIO

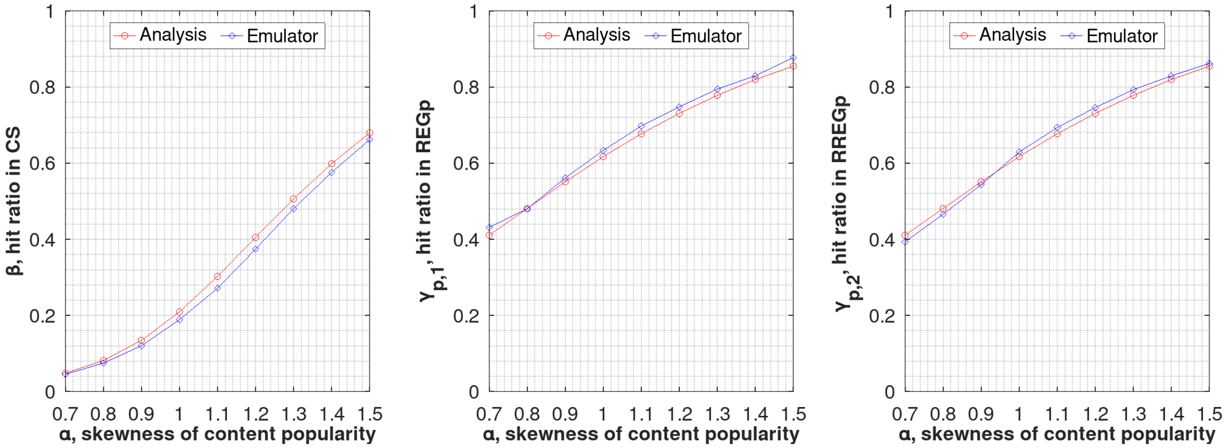
The emulation values of β, $\gamma_{p,1}$, $\gamma_{p,2}$, $\gamma_{c,1}$, and $\gamma_{c,2}$ were measured and compared against the analysis model seen in Figure 13. In the case of scenario 1, the hit ratio of CS, β, constantly increased from 5% to 68% when the content request skewness, α, increased from 0.7 to 1.5. Moreover, the gap of β between the analytical model and emulation increased as α increased by up to 5% as seen in Figure 13(a). As for $\gamma_{p,1}$ and $\gamma_{p,2}$ with an REG size of 200 out of 1000 items, the hit ratios, $\gamma_{p,1}$ and $\gamma_{p,2}$, gradually increased from 41% to 85% when the content request skewness, α, increased from 0.7 to 1.5. Furthermore, the gap hit ratios of $\gamma_{p,1}$ and $\gamma_{p,2}$ were quite similar to each other with an average of 2% oscillation along with the α axis as shown in Figure 13(b) and (c).

In the case of scenario 2, the hit ratio of CS, β, was also followed a similar trend as scenario 1 with a rise from 5% to 68% when the content request skewness, α, increased from 0.7 to 1.5. The gap of β between the analysis and emulation was 2% smaller than in scenario 1 as shown in Figure 13(d). The $\gamma_{c,1}$ and $\gamma_{c,2}$ behaved differently due to different ratios of content request. The $\gamma_{c,1}$ request ratio came from $q(i)$, while that of $\gamma_{c,2}$ came from $r(i)$. As a result, $\gamma_{c,1}$ increased from 41% to 85% as shown in Figure 13(e), while $\gamma_{c,2}$ increased from 43% to 95% as seen in Figure 13(f) when the content request skewness, α, developed from 0.7 to 1.5 with an REG size equal to 20% of the total number of prefix names. Furthermore, the $\gamma_{c,1}$ gap began at around 3% and decreased as α increased. However, the gap between $\gamma_{c,2}$ was nearly zero for all values of α.

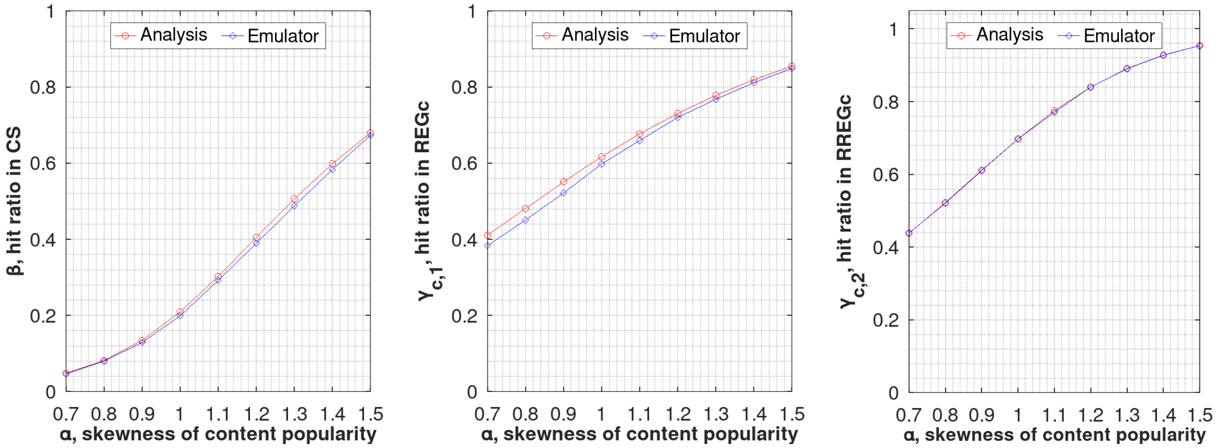
2) PROCESSING TIME DISTRIBUTION

In the case of dynamic prefix-name binding deployment, scenario 1 had five different processing time distributions, whereas scenario 2 had six. The introductions of the hit ratio of the REGp, REGc, RREGp, and RREGc were the main cause for the increase in processing time. Moreover, the NRS resolving round trip time also dominantly affected some of the distributions, causing them to shift to the far right in processing time distribution.

Figure 14 illustrates the distribution in processing time for various types of packet flows in scenario 1. There were five distinct instances of packet processing time from td_{1a}



(a) Hit ratio of CS in scenario 1. **(b)** Hit ratio of REGp in scenario 1. **(c)** Hit ratio of RREGp in scenario 1.



(d) Hit ratio of CS in scenario 2. **(e)** Hit ratio of REGc in scenario 2. **(f)** Hit ratio of RREGc in scenario 2.

FIGURE 13. Comparison of hit ratio for CS (β), REG producer ($\gamma_{p,1}$), RREG producer ($\gamma_{p,2}$), REG consumer ($\gamma_{c,1}$), and RREG consumer ($\gamma_{c,2}$) when CS and REG size equal 1 % and 20 %, respectively, of total number of prefix names in the case of dynamic prefix-name binding.

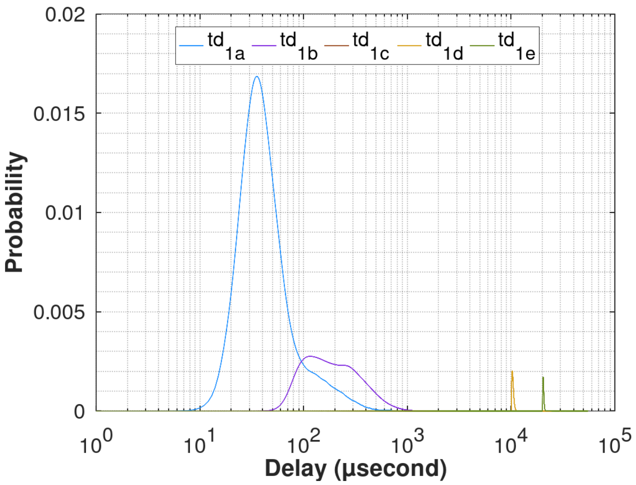


FIGURE 14. Distribution of processing time components in the case of scenario 1 and dynamic prefix-name binding scheme.

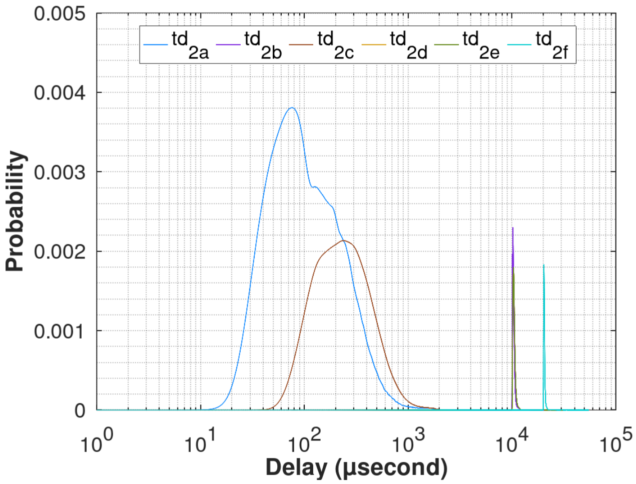
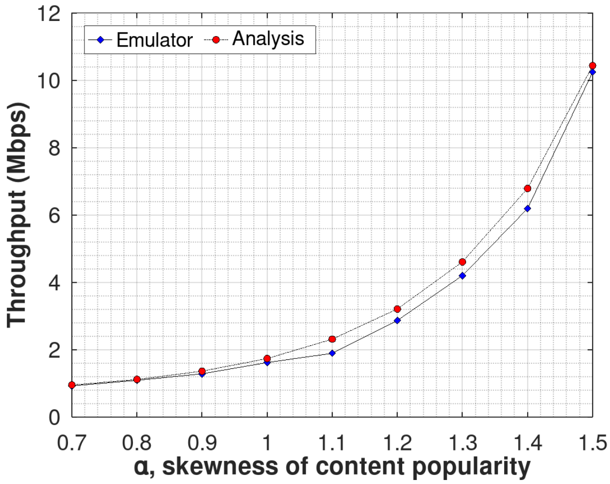
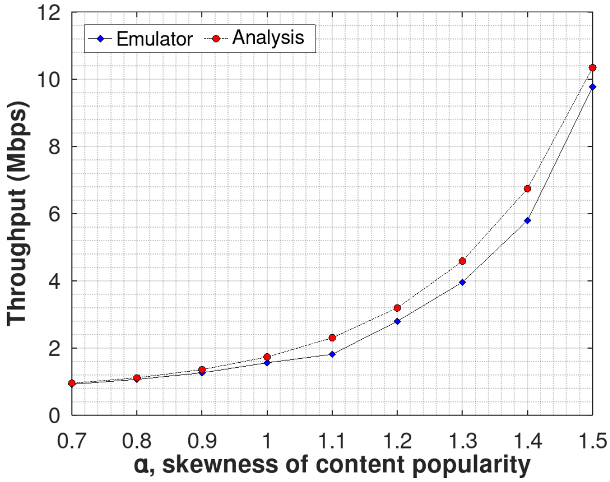


FIGURE 15. Distribution of processing time components in the case of scenario 2 and dynamic prefix-name binding scheme.

to td_{1e}. The td_{1a} distribution had the shortest processing time with an interval between 10 and 900 μs, whereas the td_{1b} distribution had the second shortest processing time between 80 and 1000 μs, followed by the td_{1c} and td_{1d} distributions, which overlapped each other with a central value of around 10 milliseconds and an interval between maximum



(a) Median approach.

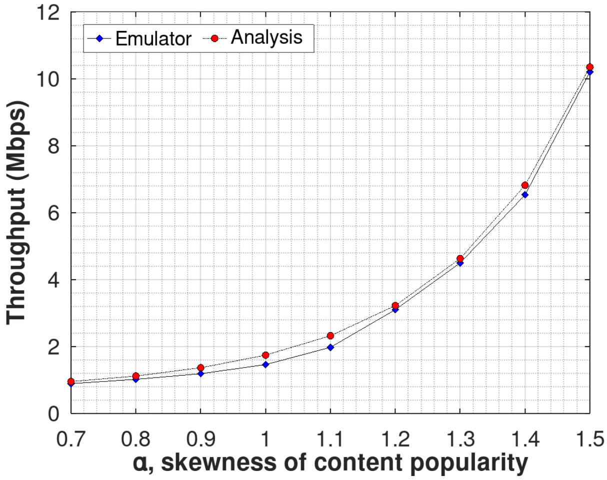


(b) Mean approach.

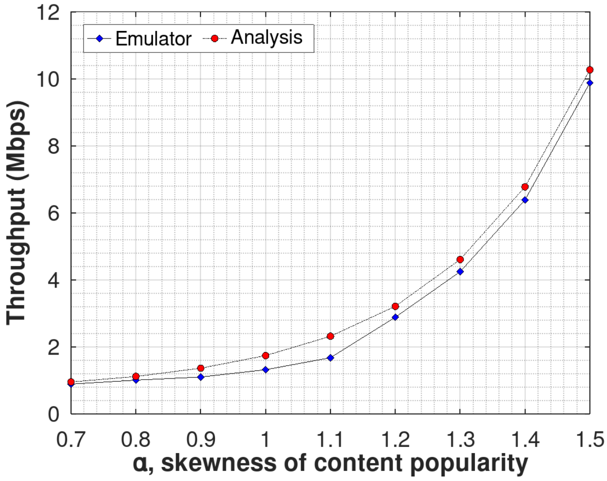
FIGURE 16. **Throughput comparison between analysis and emulation when CS size was 1% of total number of prefix names in the case of scenario 1 and dynamic prefix-name binding scheme.**

and minimum of about 1000 μs. This overlap was caused by these distributions accessing the NRS server once. Finally, td_{1e} was the longest with a central value of about 20 milliseconds since td_{1e} accessed the NRS server twice.

In the case of scenario 2, there were six distinct cases of packet-processing time distribution, with td_{2a} being the smallest followed by td_{2c}, td_{2b}, td_{2d}, td_{2e}, and td_{2f} as seen in Figure 15. The distribution of td_{2a} was in the approximate range of 10 to 900 μs, whereas the value of td_{2c} was in the range of 80 to 1000 μs. Interestingly, the distributions of td_{2b}, td_{2d}, and td_{2e} mainly overlapped with each other, with a median of roughly 10 ms. As expected, this is because the distribution of td_{2b}, td_{2d}, and td_{2e} experienced the NRS resolving round trip time once. As the longest, the distribution of td_{2f} had a center processing time of roughly 20 milliseconds because it experienced the NRS resolving round trip time twice.



(a) Median approach.



(b) Mean approach.

FIGURE 17. **Throughput comparison between analysis and emulation when CS size was 1% of total number of prefix names in the case of scenario 2 and dynamic prefix-name binding scheme.**

3) THROUGHPUT

Figure 16 shows the throughput comparison between emulation and analysis with the two different statistic approaches, median and mean, in the case of scenario 1. Alternatively, Figure 17 depicts the comparison of the throughput in the case of scenario 2.

With the capacity, only 20% of the total number of prefix names were stored in the REG table. The throughput of the dual-channel gateway in dynamic prefix-name binding was between 2.5% and 25% from the static prefix-name binding along the α axis when the CS capacity was equal to 1% of the total number of prefix names. Additional requests to the NRS server caused a significant drop in the overall throughput of the dual-channel gateway because they led to a longer processing time.

In dynamic prefix-name binding, similar to static prefix-name binding, the median approach had a higher accuracy than the mean for both scenarios. In the case of scenario 1, the throughput gap between analysis and emulation was less than 10%, with the median being slightly better than the mean at less than 15% as seen in Figure 16. Moreover, scenario 2 had a higher accuracy compared with scenario 1. The analysis model could estimate throughput with a 95% accuracy when using the median approach but with only an 85% accuracy when using the mean approach, as shown in Figure 17. This is because scenario 2 had a smaller difference in the hit ratio comparison, particularly for β, than scenario 1. In addition to that, the hit ratio gap between analysis and emulation in the REG consumer table, especially $\gamma_{c,2}$, was close to zero for any value of α. As a result, the estimation accuracy in the case of scenario 2 slightly improved.

The estimation throughput was close to that for emulation when the value of α was either extremely low or high. As for a low α, the throughput could be accurately estimated because the gap of the CS hit ratio, β, between the analysis and simulation, was near zero; then it increased constantly as α increased. However, for a high α, the contribution of shorter processing times, such as td_{1a} and td_{2a}, improved the accuracy of estimation since the values of td_{1a} and td_{2a} had a lower number of delay components. A number of delay components in packet processing results in a longer processing time. Therefore, when the value of α was located in the middle of the graph, the accuracy of throughput estimation underperformed. This was because of the contribution of hit-ratio inaccuracy and longer packet processing time, which had created poor throughput estimation. This explains why a higher deviation was located in the middle of the α axis.

VI. CONCLUSION

Migration technology for cross-platform communication between two different network protocols is very important, especially during transition. During this period, a cost-effective solution is expected. In this paper, we proposed a dual-channel translation gateway enabling a minimal upgrade cost between IP and NDN protocols.

Using two different channels for recognizing two types of NDN packets in the IP network, the proposed method can leverage IP protocol semantics, especially in the network layer, without violating user privacy. A packet naming mechanism was solved by providing two asymmetric REG tables, namely REG producer and REG consumer tables. Depending on the IP endpoint configuration, these tables are utilized by either the IP as a producer or as a consumer. We also provided simple analytical models of the throughput of the proposed gateway, and we showed that the proposed model can accurately estimate the throughput of the gateway.

REFERENCES

[1] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, ''NDNS: A DNS-like name service for NDN,'' in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul./Aug. 2017, pp. 1–9.

[2] A. Mahanti, C. Williamson, and D. Eager, ''Traffic analysis of a web proxy caching hierarchy,'' *IEEE Netw.*, vol. 14, no. 3, pp. 16–23, May 2000.

[3] A. Padmanabhan, L. Wang, and L. Zhang, ''Automated tunneling over IP land: Run NDN anywhere,'' in *Proc. 5th ACM Conf. Inf.-Centric Netw.*, Boston, MA, USA, Sep. 2018, pp. 188–189.

[4] B. Nour, H. Ibn-Khedher, H. Moungla, H. Afifi, F. Li, K. Sharif, H. Khelifi, and M. Guizani, ''Internet of Things mobility over information-centric/named-data networking,'' *IEEE Internet Comput.*, vol. 24, no. 1, pp. 14–24, Jan. 2020.

[5] B. Nour, F. Li, H. Khelifi, H. Moungla, and A. Ksentini, ''Coexistence of ICN and IP networks: An NFV as a service approach,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[6] B. Nour, H. Khelifi, R. Hussain, H. Moungla, and S. H. Bouk, ''A collaborative multi-metric interface ranking scheme for named data networks,'' in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Limassol, Cyprus, Jun. 2020, pp. 2088–2093.

[7] B. Nour, S. Mastorakis, R. Ullah, and N. Stergiou, ''Information-centric networking in wireless environments: Security risks and challenges,'' *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 121–127, Apr. 2021.

[8] B. R. Dawadi, D. B. Rawat, S. R. Joshi, P. Manzoni, and M. M. Keitsch, ''Migration cost optimization for service provider legacy network migration to software-defined IPv6 network,'' *Int. J. Netw. Manage.*, vol. 31, no. 4, p. e2145, Jul./Aug. 2021.

[9] B. Wissingh, C. Wood, A. Afanasyev, L. Zhang, D. Oran, and C. Tschudin, *Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology*, document RFC 8793, Jun. 2020. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8793.pdf

[10] C.-I. Fan, I.-T. Chen, C.-K. Cheng, J.-J. Huang, and W.-T. Chen, ''FTP-NDN: File transfer protocol based on re-encryption for named data network supporting nondesignated receivers,'' *IEEE Syst. J.*, vol. 12, no. 1, pp. 473–484, Mar. 2018.

[11] C. Guimaraes, J. Quevedo, R. Ferreira, D. Corujo, and R. L. Aguiar, ''Content retrieval while moving across IP and NDN network architectures,'' in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Barcelona, Spain, Jun./Jul. 2019, pp. 1–6.

[12] C. Gundogan, C. Amsuss, T. C. Schmidt, and M. Wählisch, ''IoT content object security with OSCORE and NDN: A first experimental comparison,'' in *Proc. IFIP Netw. Conf. (Networking)*, Paris, France, Jul. 2020, pp. 19–27.

[13] F. Fahrianto and N. Kamiyama, ''Comparison of migration approaches of ICN/NDN on IP networks,'' in *Proc. 5th Int. Conf. Informat. Comput. (ICIC)*, Nov. 2020, pp. 1–7.

[14] F. Fahrianto and N. Kamiyama, ''The dual-channel IP-to-NDN translation gateway,'' in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Boston, MA, USA, Jul. 2021, pp. 1–2.

[15] G. Carofiglio, L. Muscariello, J. Augé, M. Papalini, M. Sardara, and A. Compagno, ''Enabling ICN in the internet protocol: Analysis and evaluation of the hybrid-ICN architecture,'' in *Proc. 6th ACM Conf. Inf.-Centric Netw.*, Macao, China, Sep. 2019, pp. 55–66.

[16] G. Xylomenos, M. J. Reed, M. Al-Naday, G. Petropoulos, K. Katsaros, M.-E. Xezonaki, J. Riihijarvi, Y. Thomas, X. Vasilakos, M. Georgiades, A. Phinikarides, I. Doumanis, S. Porter, D. Trossen, and S. Robitzsch, ''IP over ICN Goes live,'' in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2018, pp. 319–323.

[17] H. Che, Y. Tung, and Z. Wang, ''Hierarchical web caching systems: Modeling, design and experimental results,'' *IEEE J. Sel. Areas Commun.*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.

[18] H. Kim and N. Ko, ''A scalable pub/sub system for NDN,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, Oct. 2020, pp. 1067–1069.

[19] H. Liu, K. Azhandeh, X. de Foy, and R. Gazda, ''A comparative study of name resolution and routing mechanisms in information-centric networks,'' *Digit. Commun. Netw.*, vol. 5, no. 2, pp. 69–75, May 2019.

[20] H. Wu, J. Shi, Y. Wang, Y. Wang, G. Zhang, Y. Wang, B. Liu, and B. Zhang, ''On incremental deployment of named data networking in local area networks,'' in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, Beijing, China, May 2017, pp. 82–94.

[21] I. Moisennko and D. Oran, ''TCP/ICN: Carrying TCP over content centric and name data networks,'' *Proc. ICN*, Kyoto, Japan, Sep. 2016, pp. 112–121.

[22] J. Hong, T.-W. You, and Y.-G. Hong, ''Name resolution service for CCN,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, Oct. 2017, pp. 1276–1279.

[23] J. Li, Y. Sheng, and H. Deng, "Two optimization algorithms for name-resolution server placement in information-centric networking," *Appl. Sci.*, vol. 10, no. 10, p. 3588, May 2020.

[24] J. Thompson, P. Gusev, and J. Burke, "NDN-CNL: A hierarchical namespace API for named data networking," in *Proc. 6th ACM Conf. Inf.-Centric Netw.*, New York, NY, USA, Sep. 2019, pp. 30–36.

[25] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Conf. Comput. Commun., 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. Future Now (INFOCOM)*, vol. 1, Mar. 1999, pp. 126–134.

[26] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. C. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 4, pp. 66–73, Jul. 2014.

[27] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[28] M. N. D. Satria, F. H. Ilma, and N. R. Syambas, "Performance comparison of named data networking and IP-based networking in palapa ring network," in *Proc. 3rd Int. Conf. Wireless Telematics (ICWT)*, Palembang, Indonesia, Jul. 2017, pp. 43–48.

[29] N. Atre, J. Sherry, W. Wang, and D. S. Berger, "Caching with delayed hits," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Architectures, Protocols Comput. Commun. (SIGCOMM)*, New York, NY, USA, Aug. 2020, pp. 495–513.

[30] P. Yue, R. Li, and B. Pang, "The random content poisoning attack in NDN," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. With Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, New York City, NY, USA, Sep./Oct. 2021, pp. 853–860.

[31] S. Gao, H. Zhang, and B. Zhang, "Supporting multi-dimensional naming for NDN applications," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[32] S. Luo, S. Zhong, and K. Lei, "IP/NDN: A multi-level translation and migration mechanism," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Taipei, Taiwan, Apr. 2018, pp. 1–5.

[33] T. Liang, J. Pan, and B. Zhang, "NDNizing existing applications: Research issues and experiences," in *Proc. 5th ACM Conf. Inf.-Centric Netw.*, Boston, MA, USA, Sep. 2018, pp. 172–183.

[34] T. Refaei, J. Ma, S. Ha, and S. Liu, "Integrating IP and NDN through an extensible IP-NDN gateway," in *Proc. ACM Inf. Centric Netw.*, Berlin Germany, Sep. 2017, pp. 224–225.

[35] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. ACM CoNEXT*, Rome, Italy, Dec. 2009, pp. 1–12.

[36] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-over content-centric networks," in *Proc. ReArch ACM CoNEXT*, Rome, Italy, Dec. 2009, pp. 1–6.

[37] X. Tan, W. Feng, J. Lv, Y. Jin, Z. Zhao, and J. Yang, "F-NDN: An extended architecture of NDN supporting flow transmission mode," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6359–6373, Oct. 2020.

[38] X. Ma. (2019). Python-NDN, UCLA, CA, USA. [Online]. Available: https://zjkmxy.github.io/projects/python-ndn

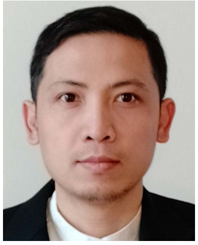
[39] Y. Cui, J. Dong, P. Wu, J. Wu, C. Metz, Y. L. Lee, and A. Durand, "Tunnel-based IPv6 transition," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 62–68, Mar./Apr. 2013.

[40] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Müller, C. Timmerer, and H. Hellwagner, "Dynamic adaptive streaming over CCN: A caching and overhead analysis," in *Proc. IEEE Int. Conf. Commun.*, Budapest, Hungary, Jun. 2013, pp. 3629–3633.

[41] Y. N. Rohmah, D. W. Sudiharto, and A. Herutomo, "The performance comparison of forwarding mechanism between IPv4 and named data networking (NDN). Case study: A node compromised by the prefix hijack," in *Proc. 3rd Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2017, pp. 302–306.

[42] Z. Yan, G. Geng, S. Zeadally, and Y.-J. Park, "Distributed all-IP mobility management architecture supported by the NDN overlay," *IEEE Access*, vol. 5, pp. 243–251, 2017.

[43] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "ACT: Audio conference tool over named data networking," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw. (ICN)*, New York, NY, USA, Aug. 2011, pp. 68–73.



FERI FAHRIANTO (Graduate Student Member, IEEE) received the bachelor's degree in electronics engineering from the National Institute of Technology, Bandung, Indonesia, in 2004, and the master's degree in computer science and communication engineering from the University of Duisburg-Essen, North Rhine-Westphalia, Germany, in 2009. He is currently pursuing the Ph.D. degree with the Graduate School of Engineering, Fukuoka University, Kyushu, Japan. He has also been a Lecturer at the Syarif Hidayatullah State Islamic University Jakarta, Indonesia, since 2010. His research interest includes migration of future internet architectures. Besides the future internet architecture area, he is also interested in embedded systems, mobile technology, and the IoT. He is currently a Student Member of IEICE.

NORIAKI KAMIYAMA (Member, IEEE) received the M.E. and Ph.D. degrees in communications engineering from Osaka University, in 1994 and 1996, respectively. From 1996 to 1997, he was with the University of Southern California as a Visiting Researcher. He joined NTT Multimedia Network Laboratories, in 1997, and he has been at NTT Network Technology Laboratories, since 2016. He was also with Osaka University as an invited Associate Professor, from 2013 to 2014, and an invited Professor, in 2015. Since 2017, he has been a Professor at Fukuoka University. Since 2021, he has been a Professor at Ritsumeikan University. His research interests include distribution systems, network design, network economics, traffic measurement and analysis, and traffic engineering. He received the Best Paper Award at IFIP/IEEE IM 2013. He is a member of ACM and IEICE.

• • •