一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CQ2021-95(2022-01)

# 動的 Prefix Name Binding と DualChannel を用いた IP-NDN ゲートウェイのスループット解析

フェリファリアント†　上山　憲昭††

† 福岡大学 大学院 工学研究科 電子情報工学専攻　〒814–0180 福岡市城南区七隈 8–19–1
†† 立命館大学 情報理工学部　〒525–8577 滋賀県草津市野路東 1–1–1
E-mail: †td196502@cis.fukuoka.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

**あらまし**　筆者らがこれまでに提案した Dual Channel ゲートウェイを用いることで，IP と NDN との間でセキュアで効率的なプロトコル変換が可能となる．IP-NDN ゲートウェイでは，IP パケットの IP address と NDN パケットの Prefix name とを対応づける必要があるが，外部の Name Resolution Server (NRS) を利用して，動的に対応づけを行う．本稿では，動的 Prefix 解決がスループットに与える影響を分析する．ゲートウェイのスループットに影響を与えるヒット率やパケット処理時間を解析し，これらを用いたスループットの簡易な解析モデルを提案する．平均すると，提案モデルはエミュレータによる実機測定値と比較して，IP producer - NDN consumer の場合には約 85%の，NDN producer - NDN consumer の場合には約 90%の近似精度が得られることを確認する．
**キーワード**　NDN，マイグレーション，スループット解析

# Throughput Analysis of Translation Gateway Between IP and NDN Using Dual Channel with Dynamic Prefix Name Binding

Feri FAHRIANTO† and Noriaki KAMIYAMA††

† Graduate School of Engineering, Fukuoka University
8–19–1, Nanakuma, Jounan, Fukuoka 814–0180
†† College of Information Science and Engineering, Ritsumeikan University
1–1–1 Nojihigashi, Kusatsu, Shiga 525–0058
E-mail: †td196502@cis.fukuoka.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

**Abstract**　A translation gateway can be used to migrate network protocols effectively. The dual-channel IP–to-NDN translation gateway enables the IP protocol to take advantage of the semantics of the NDN protocol. The IP addresses of IP packets are linked to prefix names of NDN packets within the gateway. The network's binding prefix-name technique can be dynamically implemented by exploiting the Name Resolution Server (NRS). This paper investigates the impact of dynamic prefix-name binding on overall throughput. The affecting components of the gateway, such as hit ratio and packet-processing time, are analyzed. The throughput estimation model is introduced in the study. On average, the model can accurately predict the gateway throughput to within around 85% in the case of an IP producer and an NDN consumer, and 90% in the case of an NDN producer and IP consumer of the emulation testbed.
**Key words**　NDN, migration, throughput analysis

## 1. Introduction

By naming the data items in the network, Information-Centric Networking (ICN) transforms the information transmission paradigm. The ICN data transaction involves the transmission of two distinct packets, the interest and data packets. In order to obtain data from a certain producer, the consumer sends an interest packet with a prefix name in advance, and the producer responds with a data packet.

Named Data Networking (NDN), an active and expanding ICN technology, has a rapid trial scale deployment. The installation of NDN technology will surely take over the global communication network in the future. During the network protocol transition, however, compatibility between IP and NDN will be unavoidable [1] [2]. Therefore, in [3], we proposed a dual-channel IP-to-NDN translation gateway that allows the IP protocol to interact on semantic levels at a


This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2022 by IEICE

pace with the NDN protocol. On the other hand, the prefix name binding between IP packets is essential for NDN and IP compatibility. As a result, we explore the effect of dynamic prefix name binding on overall throughput in order to better understand the key components of the dual-channel translation gateway.

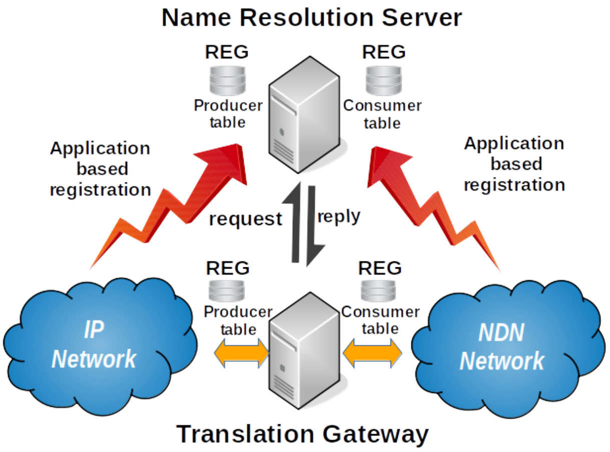



Fig. 1: Dual channel translation gateway architecture

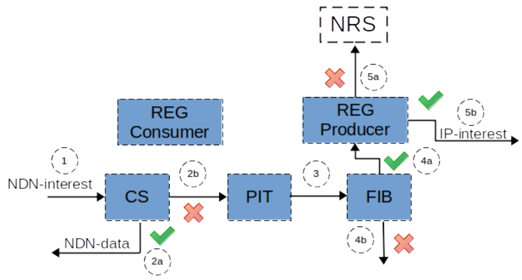
Name Resolution Server (NRS) interacts with the translation gateway to offer a naming service for binding an IP address to the register table (REG) that has two asymetrical table, namely REG producer table and REG consumer table, as seen in Figure 1. The NRS contains all prefix name binding in the network while the translation gateway partially stores the prefix name biding as a cache due to its limited memory size. Thus, a cache miss in translation gateway will request for a target prefix name to NRS. Furthermore, we look at the impact of dynamic prefix name binding on total throughput in the dual-channel gateway. A throughput model is used to show the relationship between Content Store (CS), REG, and throughput. In addition, we compare the estimation model to the emulator testbed to ensure the correctness of the throughput model. The study is divided into several sections, including an introduction, related work, dual-channel translation gateway and NRS, throughput analysis model, numerical evaluation, conclusion and future work.

## 2. Related work

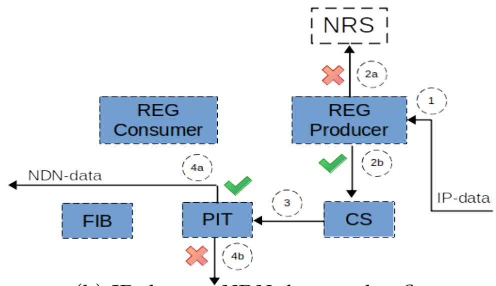
Many scholars have actively argued a concept for IP and NDN compatibility. G. Carofiglio et al. demonstrated hICN partial migration techniques, which placed an ICN/NDN packet within an IP packet [5]. Although the NDN packet can be transmitted within the IP network using an encoded prefix name composed of the encoded IP and port number, this requires the use of a pair of hICN routers to extract the hidden NDN packet from the IP packet. Moiseenko et al. suggested a TCP/ICN proxy pair comprised of a forward proxy and a reverse proxy capable of routing TCP traffic through an ICN infrastructure [8]. S. Luo et al. show a novel hierarchical naming strategy in the IP to ICN packet translation name [12]. Similar to S. Luo, Rafaei et al. in [14] translated via a gateway that employed XML to bind the prefix-name. None of them, however, address the impact of prefix name binding on overall throughput. The impact of dynamic prefix name binding on throughput in a dual-channel IP-to-NDN translation gateway is investigated in this paper. Moreover, the concept of binding the packet with prefix name that utilizing external server has been introduced in [10] and [11].

## 3. The dual-channel translation gateway and NRS

The dual-channel translation gateway equips with ordinary ICN router component such as CS, Pending Interest Table (PIT), Forwarding Information Base (FIB) plus an extra Register Table (REG). Moreover, the packet flow of the dual-channel translation gateway is differ in each of producer and consumer scenarios. The packet flow in the case of IP producer and NDN consumer, denoted as *scenario 1*, is shown in Figure 2. Alternatively, Figure 3 shows the packet flow in the case of NDN producer and IP consumer, denoted as *scenario 2*.




(a) NDN interest to IP interest packet flow

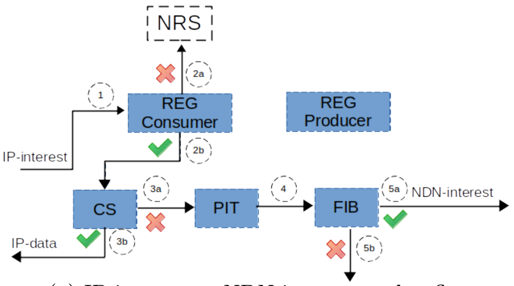



(b) IP data to NDN data packet flow

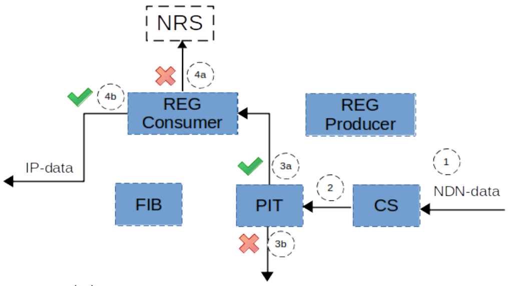
Fig. 2: IP and NDN protocol translation packet flow in case of scenario 1

In 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 searching prefix name (4b). When the outgoing interface is discovered, it is passed to the REG table to determine the corresponding IP address (4a). The gateway sends a resolving packet to NRS and waits for a response because the REG table is likewise restricted (5a). The IP interest packet is then transmitted across the IP channel (5b) as seen in Figure 2(a). When 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 table (1). If it is not found, the gateway sends a resolving packet to NRS (2a). And then forward the data packet through interface (4a) before that is found in PIT table (3) and stored in CS (2b) as shown in Figure 2(b).

In case of scenario 2, the packet flow is similar with scenario 1 except the interest packets are first checked in the REG 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




(a) IP interest to NDN interest packet flow




(b) NDN data to IP data packet flow

Fig. 3: IP and NDN protocol translation packet flow in case of scenario 2

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 3(a). When the data packet arrived, the gateway stores the content in CS (1) and verifies the outgoing interface in PIT (2) when NDN producers deliver NDN data packets. The prefix name retrieved the related IP address in the REG database because the outgoing interface is an IP interface. In the event that it is not found, the gateway sends a REG resolving packet (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) as shown in Figure 3(b).

## 4. Throughput analysis model

The throughput is defined as the average size of packet transmitted in a second. Thus, we define throughput as the average packet size divided by $\tau$ where $\tau$ is the average time consumed in a gateway. Based on Figures 2 and 3, $\tau$ depends on the scenarios and cases of packet arrival at the gateway. In scenario 1, there are 5 different cases of packet flow, whereas we have 6 different cases in scenario 2. The number of cases is effected by the combination of CS and REG. Furthermore, we also define $\beta$ as the hit ratio in CS, $\gamma_1$ as the hit ratio in REG when the prefix name is input key, and $\gamma_2$ as the hit ratio when IP address is input key, i.e., Reverse REG (RREG). Table 1 shows the descriptions and symbol of processing time in the gateway that used in the throughput model.

In the case of scenario 1, the value of $\tau$ can be described by

$$\tau = \begin{cases} t_{1a} & \text{if CS hit} \\ t_{1b} & \text{if CS miss, REG hit, and RREG hit} \\ t_{1c} & \text{if CS miss, REG hit, and RREG miss} \\ t_{1d} & \text{if CS miss, REG miss, and RREG hit} \\ t_{1e} & \text{if CS miss, REG miss, and RREG miss} \end{cases}$$

where

$$t_{1a} = t_{i_p} + t_c + t_{soh},$$

Tab. 1: The symbol description

| 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 REG |
| $t_{reg1'}$ | Negative look-up time in REG |
| $t_{reg2}$ | Positive look-up time in RREG |
| $t_{reg2'}$ | Negative look-up time in RREG |
| $t_{i_p}$ | 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 desconstruction 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 |

$$t_{1b} = t_{i_p} + t_{cn} + t_{reg1} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2} + 2t_{soh},$$

$$t_{1c} = t_{i_p} + t_{cn} + t_{reg1} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2'} + t_{nrs} + 3t_{soh},$$

$$t_{1d} = t_{i_p} + t_{cn} + t_{reg1'} + t_{nrs} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2} + 3t_{soh},$$

$$t_{1e} = t_{i_p} + t_{cn} + t_{reg1'} + t_{nrs} + t_{ip_c} + t_{pr1} + t_{ip_d} + t_{d_s} + t_{reg2'} + t_{nrs} + 4t_{soh}.$$

The average time of a packet processed, $\overline{\tau}$, can be obtained by

$$\overline{\tau} = \beta t_{1a} + (1-\beta)\gamma_1\gamma_2 t_{1b} + (1-\beta)\gamma_1(1-\gamma_2)t_{1c} + (1-\beta)(1-\gamma_1)\gamma_2 t_{1d} + (1-\beta)(1-\gamma_1)(1-\gamma_2)t_{1e} \quad (1)$$

However, in the case of scenario 2, the value of $\tau$ can be defined by

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

where

$$t_{2a} = t_{ip_d} + t_{reg2} + t_c + t_{d_p} + t_{ip_c} + t_{soh},$$

$$t_{2b} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_c + t_{d_p} + t_{ip_c} + 2t_{soh},$$

$$t_{2c} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_c n + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 3t_{soh},$$

$$t_{2d} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_c n + t_{i_s} + t_{d_p} + t_{nrs} + t_{pr2} + t_{reg1'} + t_{ip_c} + 4t_{soh},$$

$$t_{2e} = t_{ip_d} + t_{reg2} + t_c n + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 2t_{soh},$$

$$t_{2f} = t_{ip_d} + t_{reg2'} + t_{nrs} + t_c n + t_{i_s} + t_{d_p} + t_{pr2} + t_{reg1} + t_{ip_c} + 3t_{soh}.$$

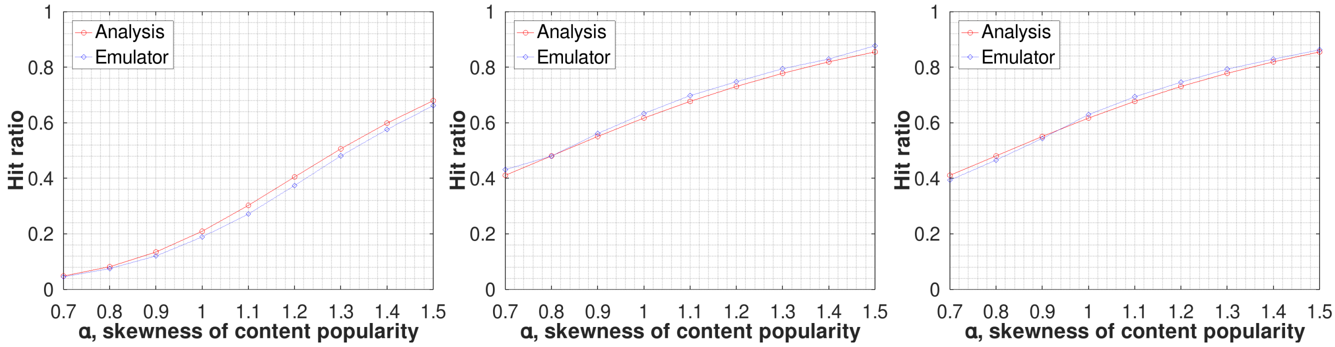
The average processing time for scenario 2 is expressed by

$$\overline{\tau} = \beta\gamma_2 t_{2a} + \beta(1-\gamma_2)t_{2b} + (1-\beta)\gamma_1\gamma_2 t_{2c} + (1-\beta)\gamma_1(1-\gamma_2)t_{2d} + (1-\beta)(1-\gamma_1)\gamma_2 t_{2e} + (1-\beta)(1-\gamma_1)(1-\gamma_2)t_{2f} \quad (2)$$

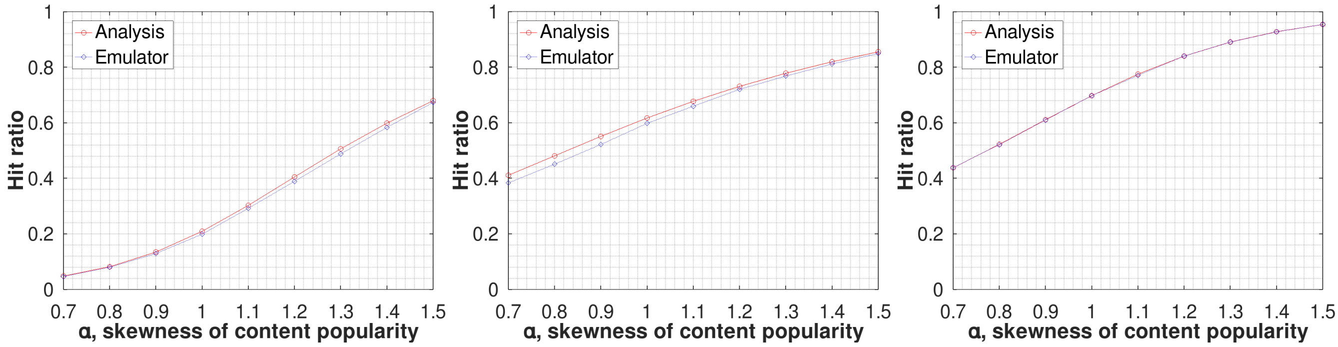
The value of $\beta$ can be approximated by calculating the hit ratio of each prefix name, $\beta(i)$, with its distribution, $q(i)$, for the total unique prefix names, $M$. The formula can be expressed by

$$\beta = \sum_{i=1}^{M} q(i)\beta(i) \quad (3)$$

The approximation of hit ratio could be calculated by using




(a) The value of $\beta$ in scenario 1 (b) The value of $\gamma_1$ in scenario 1 (c) The value of $\gamma_2$ in scenario 1




(d) The value of $\beta$ in scenario 2 (e) The value of $\gamma_1$ in scenario 2 (f) The value of $\gamma_2$ in scenario 2

Fig. 4: Comparison of $\beta$, $\gamma_1$, $\gamma_2$ between analytical estimation model and emulation testbed in the case of scenario 1 and scenario 2

Che's approximation that was introduced by Che et al. [6]. Then, the hit ratio for a particular prefix name, $\beta(i)$, could be estimated by

$$\beta(i) = 1 - e^{-q(i)t_c}. \tag{4}$$

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

$$C = \sum_{i=1}^{M} (1 - e^{-q(i)t_c}) \tag{5}$$

where $C$ is the storage capacity of CS/REG that uses LRU as a replacement algorithm.

The value of hit ratio in REG table $\gamma_1$ or $\gamma_2$ may differ due to packet flow in the translation process. As for scenario 1, the value of $\gamma_1$ and $\gamma_2$ are highly corelated with the CS miss ratio so the same formula can be used to estimate both $\gamma_1$ and $\gamma_2$. The distribution of content $i$ requested at the REG table, $r(i)$, can be aproximated by

$$r(i) = q(i)(1 - \beta(i)) \tag{6}$$

By using the equation (5), the unique root, $t_r$, for REG table can be obtained. And finally, the hit ratio of particular prefix name in REG table must be adjusted by normalization factor as described in equation (7).

$$\gamma(i) = \frac{1}{(1-\beta)}(1 - e^{-r(i)t_r}) \tag{7}$$

As for scenario 2, the value of $\gamma_1$ is corelated with the CS miss ratio so it follows the equation (7). However, the value of $\gamma_2$ is not affected by the CS miss ratio so it follows the step of equation (3), (4), (5) that is used similarly to find $\beta$. This is because the distribution of prefix name requested in REG and CS are identical.

## 5. Numerical evaluation

The emulation testbed was constructed with following the topology shown in Figure 1. Virtual box and python 3.7 were used in the testbed.

Tab. 2: Setting values of main parameters

| Paramater | Value |
|---|---|
| Number of content items | 1000 |
| Cache-size | 10 |
| REG-size | 200 |
| Zipf parameter ($\alpha$) | 0.7 - 1.5 |
| Interest rate | 10000 interests/s |

The major parameters set up of the emulator are shown in Table 2. The interest packets were generated asynchronously about ten thousand packets in second to measure the throughput. The request distribution of content items followed the Zipf distribution with skewed parameter $\alpha$. Furthermore, the emulation testbed was assumed that REG table size is limited to 200 prefix names.

Tab. 3: The values of processing time

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

Median and mean processing time were taken from five hundred thousand packets in the emulation testbed. Table 3 shows the both processing time outcome.

### 5.1 Hit ratio

The emulation values of $\beta$, $\gamma_1$, and $\gamma_2$ were measured and compared against the analysis model as seen in Figure 4. In the case of scenario 1, the gap of $\beta$ between the analytical model and emulation increased as $\alpha$ increased by up to 5%. As for $\gamma_1$ and $\gamma_2$ the gap was quite similar with an average of 2% oscillation along the value of $\alpha$. In the case of scenario 2, the gap of $\beta$ was 2% smaller than in scenario 1, even though it was following the same trend. The $\gamma_1$ gap began at around 3% and decreased as $\alpha$ increased. However, the gap between $\gamma_2$ was nearly zero for all value of $\alpha$.

### 5.2 Distribution of processing time

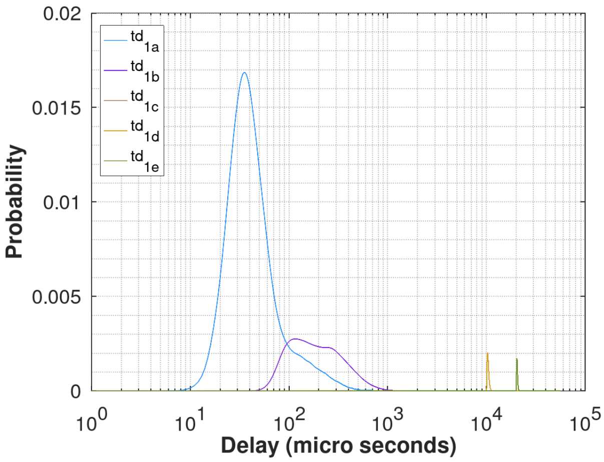



Fig. 5: Processing time distribution in the case of scenario 1

Figure 5 illustrates the distribution of processing time for various types of packet flows in the scenario 1, there were five distinct instances of packet processing time from $t_{1a}$ to $t_{1e}$. The value of $t_{1a}$ had the shortest processing time in the range of 10 to 900 $\mu$s, whereas $t_{1b}$ had the second shortest processing time between 80 and 1000 $\mu$s. The distributions of $t_{1c}$ and $t_{1d}$ that had a central value of around 10 milliseconds overlaped reciprocally. Finally, $t_{1e}$ was the longest with a median of around 20 milliseconds.

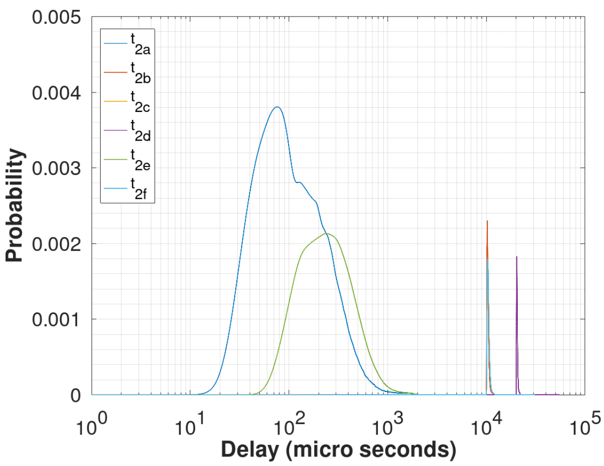



Fig. 6: Processing time distribution in the case of scenario 2

As seen in Figure 6, scenario 2 had six distinct cases of packet-processing time, with $t_{2a}$ being the smallest and followed by $t_{2c}$, $t_{2b}$, $t_{2d}$, $t_{2e}$, and $t_{2f}$. Approximately in the range of 10 to 900 $\mu$s was the value of $t_{2a}$, whereas 80 to 1000 $\mu$s was the value of $t_{2c}$. Interestingly, the distribution of $t_{2b}$, $t_{2d}$, and $t_{2a}$ mainly overlaped with each other, with a median of roughly 10 ms. Finally, $t_{2f}$ had the longest processing time, with a median of roughly 20 milliseconds.

### 5.3 Throughput

Figure 7 shows the throughput comparison between emulation and analysis by using the median and mean obtained in the scenario 1. Alternatively, Figure 8 depicts the comparison in case of scenario 2. Two statistic values from processing time, median and mean, were used to estimate the throughput due to wide range of processing time distribution.

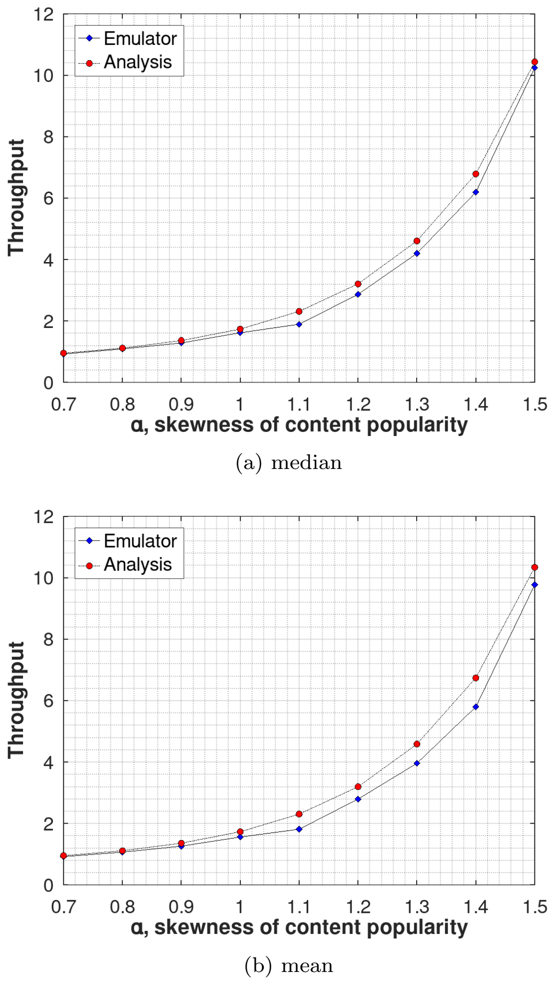
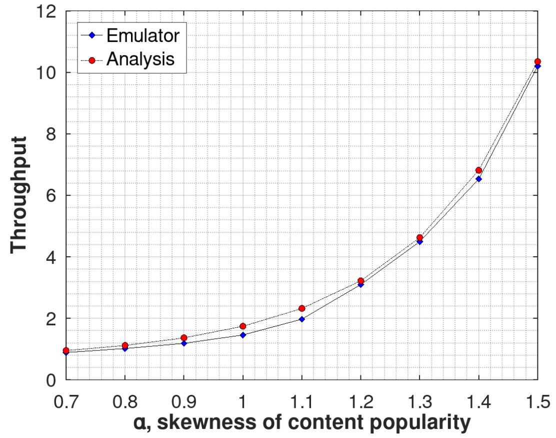



Fig. 7: Throughput comparison between analysis model and emulation in the case of scenario 1

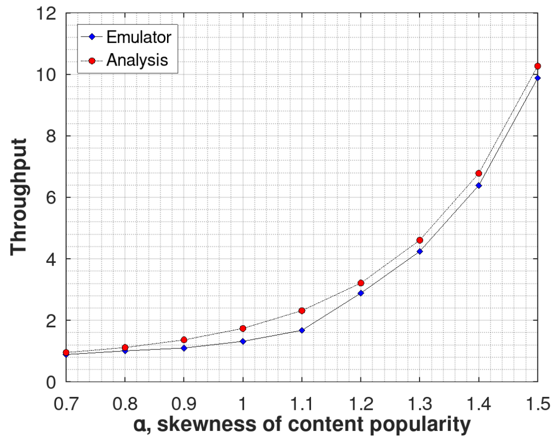
In both scenarios, the difference between analysis and emulation was smaller in the median than in the mean. In the case of scenario 1, the analysis model could estimate the throughput at about 90% with median that was slightly higher than the mean with only 80% as seen in Figure 7. Moreover, scenario 2 had a higher accuracy compared to scenario 1. The analysis model could estimate throughput with 95% accuracy when using the median, but only with 85% accuracy when using the mean, as shown in Figure 8. This was because scenario 2 had a smaller gap in $\beta$ than scenario 1. As a result, the estimation accuracy had slightly improved.

The estimation throughput was close to emulation when the value of $\alpha$ was extremely low and high. As for a low $\alpha$, the estimation could accurately estimate the throughput because the gap of $\beta$ between analysis and simulation was near to zero and it increased constantly as the $\alpha$ increased.

However, in the high $\alpha$, the contribution of shorter processing times, such as $t_{1a}$ and $t_{2a}$, had improved the accuracy of estimation since the values of $t_{1a}$ and $t_{2a}$ had fewer time components than longer processing times. More time components in packet processing means more errors. Therefore, when the value of $\alpha$ was located in the middle of the graph, the accuracy of throughput estimation was underperformed. This was because of the collaboration of hit-ratio inaccuracy and longer packet processing time that created poor throughput estimation. This reason explains why the higher deviation was located in the middle of $\alpha$ axis.




(a) median




(b) mean

Fig. 8: Throughput comparison between analysis model and emulation in the case of scenario 2

## 6. Conclusion and Future work

The study demonstrated that collaboration between hit ratio and packet processing time has an effect on the translation gateway throughput. The dynamic-prefix-name binding results in increased packet processing time and a decrease in the hit ratio of the CS and REG tables. However, the throughput estimation model can predict gateway throughput with an accuracy level of roughly 85% on average by using the mean, where the median estimation surpasses the mean estimation by about 5% due to the broad dispersion of packet processing time. The reasons of hit-ratio decline in $\beta$, $\gamma_1$, and $\gamma_2$ are being investigated for future work.


### Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 18K11283 and 21H03437, as well as the Ministry of Religious Affairs of the Republic of Indonesia (MORA 5000 Doktor).


### References


[1] B. Nour, F. Li, H. Khelifi, H. Moungla, and A. Ksentini," Coexistence of ICN and IP Networks: An NFV as a Service Approach," IEEE Global Communications Conference (Globecom), Waikoloa, USA, December 2019.

[2] F. Fahrianto and N. Kamiyama, "Comparison of Migration Approaches of ICN/NDN on IP Networks," ICIC-APTIKOM 2020, Online, November 2020.

[3] F. Fahrianto and N. Kamiyama, "The Dual-Channel IP-to-NDN Translation Gateway," 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2021.

[4] G. Xylomenos, Y. Thomas, X. Vasilakos et al., " IP Over ICN Goes Live, " in European Conference on Networks and Communications, June 2018.

[5] G. Carofiglio, L. Muscariello, J. Auge, M. Papalini, M. Sardara, and A. Compagno," Enabling ICN in the Internet Protocol: Analysis and Evaluation of the Hybrid-ICN Architecture," Proceedings of the 6th ACM Conference on Information-Centric Networking, Macao, China, September 2019.

[6] H. Che, Y. Tung, and Z. Wang, " Hierarchical Web Caching Systems:Modeling, Design and Experimental Results, " IEEE J. Selected Areas of Commun., vol.20, no.7, pp.1305-1314, September. 2002.

[7] 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 Network," ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Beijing, China, May 2017.

[8] I. Moiseenko and D. Oran, "TCP/ICN: Carrying TCP Over Content Centric and Name Data Networks," in ICN'16, Kyoto, Japan, September 2016.

[9] J. Thompson, P. Gusev, and J. Burke., "NDN-CNL: A Hierarchical Namespace API for Named Data Networking," In Proceedings of the 6th ACM Conference on Information-Centric Networking. Association for Computing Machinery, New York, NY, USA, 2019.

[10] J. Hong, T. -W. You and Y. -G. Hong, "Name resolution service for CCN," International Conference on Information and Communication Technology Convergence (ICTC), October 2017.

[11] Suyong E., Masahiro J., Masayuki M., Hitoshi A., Nozomu N., " Self-Organizing Name Resolution System in a Small World," IEICE Transactions on Communications, 2014, Volume E97.B, Issue 12, Pages 2606-2615, Released December 01, 2014.

[12] S. Luo, S. Zhong, K. Lei, "IP/NDN: A Multi-Level Translation and Migration Mechanism," IEEE/IFIP Network Operations and Management Symposium, Taiwan, April 2018.

[13] T. Liang, et al., NDNizing Existing Applications: Research Issues and Experiences, ACM ICN 2018.

[14] T. Refaei, J. Ma, S. Ha, and S. Liu, "Integrating IP and NDN through an Extensible IP-NDN Gateway," In Proceeding of ACM Information Centric Networking, Berlin Germany, September 2017 (ICN 2017).

[15] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, "Networking Named Content," CoNEXT 2009, Rome, December 2009.

[16] Xinyu Ma, UCLA, CA, USA. python-ndn(2019) [Online]. Available: https://zjkmxy.github.io/projects/python-ndn.

[17] Z. Zhu, S. Wang, X. Yang, V. Jacobson, L. Zhang," ACT: Audio Conference Tool Over Named Data Networking," ICN 2011, Ontario, Canada, August 2011.