A Novel Forwarding and Caching Scheme for Information-Centric Software-Defined Networks

Khuhawar Arif Raza
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
Email: khuhawar@mail.sustech.edu.cn

Alia Asheralieva
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
Email: aasheralieva@gmail.com

Md Monjurul Karim
Department of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
Email: mkarim@bit.edu.cn

Kashif Sharif
Department of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
Email: kashif@bit.edu.cn

Mehdi Gheisari
Department of Computer Science and Engineering
Southern University of Science and Technology, Shenzhen, China

Young Researchers and Elite Club, Parand Branch
Islamic Azad University, Parand, Iran
Email: mehdi.gheisari61@gmail.com

Salabat Khan
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
Email: salabatwazir@gmail.com

2021 International Symposium on Networks, Computers and Communications (ISNCC) | 978-1-6654-0304-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISNCC52172.2021.9615667

***Abstract*—** This paper integrates Software-Defined Networking (SDN) and Information-Centric Networking (ICN) framework to enable low latency-based stateful routing and caching management by leveraging a novel forwarding and caching strategy. The framework is implemented in a clean-slate environment that does not rely on the TCP/IP principle. It utilizes Pending Interest Tables (PIT) instead of Forwarding Information Base (FIB) to perform data dissemination among peers in the proposed IC-SDN framework. As a result, all data exchanged and cached in the system are organized in chunks with the same interest resulting in reduced packet overhead costs. Additionally, we propose an efficient caching strategy that leverages in-network caching and naming of contents through an IC-SDN controller to support off-path caching. The testbed evaluation shows that the proposed IC-SDN implementation achieves an increased throughput and reduced latency compared to the traditional information-centric environment, especially in the high load scenarios.

***Index Terms*—** Information-Centric Networking, Software-Defined Networking, Controller, PIT, Forwarding, Off-path caching.

I. Introduction

Originally, the objective of the Internet was to exchange information via emails and accommodates communication between end-users differentiated by their Internet Protocol (IP) addresses and connecting to the Internet in the client-server mode. However, over the last decades, the number of Internet users has expanded exponentially and the Internet usage has grown drastically [1]. Furthermore, as more and more users utilize Internet to access various content, e.g., multimedia, the internetworking is becoming increasingly content-centric. As a result, ICN emerges as a paradigm shift from the host-to-host IP-based Internet architecture to the information-centric architecture [2], [3]. In the ICN approach, the content is discovered and delivered based on the content name in the FIB table instead of the traditional source-destination host pair used in the Internet. ICN also supports in-network caching and multicasting, which enables efficient and timely information delivery to the users. The main drawback of the ICN architecture relates to the size of the FIB table which is much larger than the size of the IP forwarding table. As a result, the ICN-based routing requires larger memory and processing power than the traditional IP-based routing [4], [5]. The caching is also not utilized efficiently because of on-path caching support [6]. To address this problem, the SDN approach has been proposed [7]. The concept of Software-Defined Networking utilizes virtualization techniques in server applications [8], where the abstraction layer is placed above the server hardware to support multiple virtual devices to

This work was supported in part by the National Natural Science Foundation of China (NSFC) Project No. 61950410603. Corresponding author: Alia Asheralieva.

distribute the server's available resources. The contemporary implementation of the SDN paradigm is based on the OpenFlow protocol [9], [10]. The data and control planes are detached by abstracting the infrastructure resources. The data plane contains the physical or virtual instances of wired or wireless units following intelligent instructions for automated operation through a control unit managed by the network administrators from a remote location.

The integration of SDN and ICN overcomes the limitations and magnify the benefits of both technologies. However, it presents many challenges due to incompatibility in their implementation. For example, traditional SDN is based on IP address, whereas ICN routes information using the named object instead of an IP address. Using the ICN's content centric principle, several new possibilities can be introduced in SDN by shifting the traffic forwarding principle from IP to content-based paradigm and leveraging in-network forwarding and caching [11]. SDN also manages large routing tables and support off-path caching due to a global network overview.

This paper proposes a novel solution that combines the ICN with SDN leveraging a flow-based approach for efficient data forwarding and cache utilization. The main contributions of the paper are as follows:

- We propose a novel clean-slate IC-SDN that functions similar to the traditional SDN controller but utilizes the named based forwarding principle and supports both on-path and off-path caching.
- We design the content forwarding strategy in which a single request packet is imposed on the SDN controller. The controller establishes a flow in which data is forwarded from the producer to the consumer in the form of multiple chunk-based response packet.
- We use PITs instead of FIBs, which leads to enhanced network performance and reduces round-trip-time and bandwidth utilization, mainly for data flow.
- To show the superior performance of our novel communication model, we evaluate and compare it with state of the art through throughput, cache-hit, off-path cache, latency, and interest overhead.

The rest of the paper is organized as follows. In Section II, we briefly review the related work, followed by the technical overview and working principle of the proposed system model in Section III. Section IV presents the proposed framework's performance evaluation, while Section V concludes the paper with the final remark.

II. Related Works

Existing solutions on integrating ICN and SDN are divided into two categories: 1) clean-slate approach and 2) IP-dependent approach.

A. The Clean-Slate Solutions

The ICN clean-slate architecture offers a named-based host-to-content communication model. It relies on innovative features such as mobility support, multipath forwarding, in-network caching, and data encryption. It also has extensively attracted academia and industries to contribute innovative and novel solutions for forwarding [12], caching [13], congestion control [14], and other deployments such as IoT [15], 5G [16] and edge computing [17]. The clean-slate approach focuses on designing an ICN capable switches to accommodate information-centric, content-centric, or any other named-based packets. CRoS-ND [18] proposes a controller-centric single request-based proactive on-demand routing scheme. The router announces its neighbor on specific prefixes' availability, reducing the routing and controlling overhead induced by the IC-SDN controller. This work is implemented in the ndnSim 1.0 simulator that lacks extensive ICN features compared to the recent versions. Additionally, it does not install flows like traditional IP-based SDN, which leads to service and operation inflexibility in the network. The controller also lacks the awareness to cache and refresh the most desirable contents [7]. SDAR [11] presents a non OpenFlow-based intra-domain routing scheme that allows named-based data distribution in an NDN-based system. The control logic is similar to NFD (Named Data Network) [19], a forwarding scheme proposed by the NDN (Named Data Networking) community. Therefore, it is similar to a pure ICN approach that lacks the generic IC-SDN principle. The data and control planes are not logically separated, allowing all traffic to traverse through the centralized controller. It results in a bottleneck on the controller that leads to reduced overall system performance. The work in [20] proposes a community division IC-SDN solution that lessens ICN-based routing scalability and enhances content retrieval time. However, the rigid design introduces higher latency compared to the traditional ICN solution.

B. IP-dependent Solutions

The IP-dependent solution focus on specific Internet Service Provider (ISP) that seeks advantages from ICN without remodeling their entire service structure or pushing the users to adjust or replace their existing hardware and applications. In these circumstances, the existing protocols (e.g., HTTP, TCP, IP, and UDP) are utilized to support the ICN packet from one point to another.

The works in [21]–[23] proposed IP-overlay approaches to integrate OpenFlow with ICN. It requires a wrapper or plugin to implements ICN functionalities over OpenFlow. These studies have several drawbacks, such as processing delays due to additional plugins, semantic changes to IP fields due to overlay approaches and IP address dependency instead of content names.

III. Proposed System Model

The proposed IC-SDN architecture separates the control plane from the data plane and provides a programmable data plane interface. The control plane comprises a controller that functions as a control logic of the entire network [24]. The controller registers content generated by the producer or intermediate node and stores this content in the corresponding table. When the producer or ICN node obtain the content, they send it to the controller to register as an interest packet.

In the case when the consumer requests for content arriving at the controller, the content will be delivered through the best route by adopting the Dijkstra's shortest path [25] algorithm that minimizes the transmission time for a packet and installs flows information. The corresponding flows are installed by the controller with the help of the interest packet. Furthermore, the data plane is the forwarding plane that consists of the ICN node, content generator, and consumer. The ICN node is the data forwarding node, which routes data between producer and consumer, understands flow information, and sets the PIT on the ICN node.

In the following, we design a clean-slate SDN controller which is fully compatible with the ICN forwarding principle and utilizes in-network caching. The application for ICN node (router) is deployed on all switches that recognizes the flow information and assign node identifiers like node_id. Besides, we develop different application modules such as IC-SDN controller, a custom consumer, and a custom producer. Figure 1 shows an overall architecture of the proposed system.

We also modify the traditional ICN packet structure to contain the proposed flow-based information. Moreover, the proposed IC-SDN solution features multipath content forwarding, content management, caching strategy, and intelligent content discovery. The proposed IC-SDN leverages the followings:

1) The IC-SDN solution works on clean-slate ICN forwarding principle resulting in the limited overhead of mapping IP to name objects.
2) The data is accessed in chunks. Therefore, a lesser amount of interest packet is required to allocate many contents to the desired locations. The controller acknowledges the content and topology simultaneously, allowing less interest looping and control overhead.
3) The proposed framework requires half of the total round-trip-time (RTT) due to the single flow initializes multiple PIT entries resulting in faster data start delivery from producer to consumer. On the contrary, the traditional ICN only delivers a single packet based on a single request packet.
4) The IC-SDN controller has a global network view that provides the best forwarding or routing path to the requested consumer, improving the system's overall latency.

Accordingly, the proposed framework enhances the performance by utilizing the efficient forwarding strategy and off-path caching. In particular, our proposed IC-SDN architecture comprises of 1) Packet Structure 2) Content Management 3) Content Forwarding Strategy 4) Caching Strategy 5) Content Lookup. In the following, we describe the details of each architecture component.

A. Packet Structure

The ICN-based interest and data packet are modified in the proposed system to carry flow and other information. We categorize packets into four types: Flow, Interest, Registration, and Data.

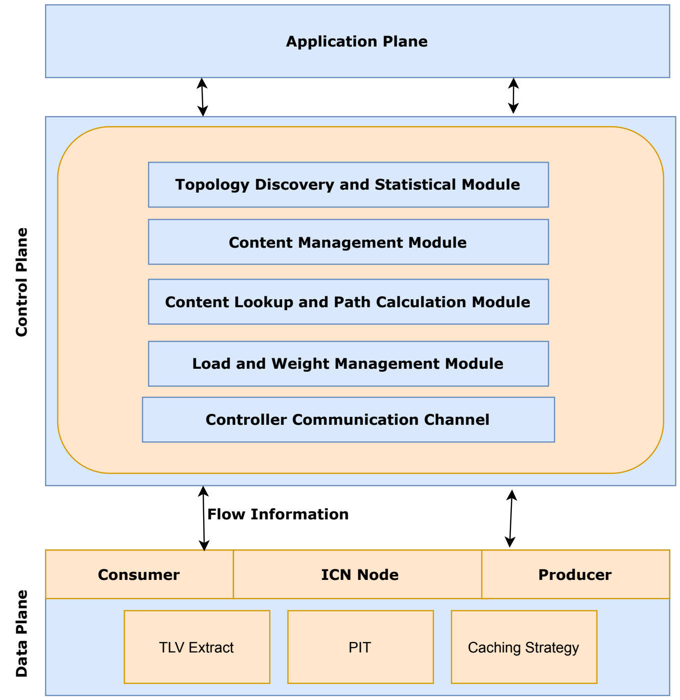
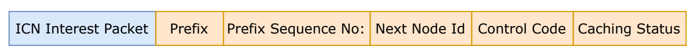
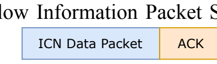


Figure 1: Components of IC-SDN Controller.

Firstly, we define Type-Length-Values (TLVs) for prefix, prefix sequence number, node id, control code, and cache status. Then we integrate them into the default ICN interest packet to correspond to a specific action for an individual task. The newly proposed interest packet structure is shown in Figure 2a.



(a) Flow Information Packet Structure.



(b) Data Packet Structure.

Figure 2: Packet Structure.

The data packet in our system has a few modifications. In particular, it adds the acknowledgment(ACK) field to the TLV of the packet, which tells the content is available or not and the size of the data chunk to access that chunk. The packet structure of the proposed data packet is shown in Figure 2b.

B. Content Management

The controller registers content from the original content generator or intermediate nodes and save them into the respective table (producer or caching table). The controller also manages the content status table to check whether the content is cacheable or not guided by the content generator. If the content is from the intermediate node, then the controller registers the name of the content, node-id, sequence of content, and the timer into its Cache-Table. If the content is not requested at a certain time, then the table's entry is deleted, but if the content is requested again, the timer is reset as shown in Algorithm 1 presented in Figure 3.

```
procedure Insert–Cache(Prefix,Node_Id,Timer)
    Input:Prefix,sequence,Node_Id,Control_Code
    if Node(x) =Producer Node then
        Content_Table←Insert(Prefix/seq,Node_id)
        Content_Status_Table←Insert(Prefix/seq,Caching_Status)
    end if
    if Node(x) =Intermediate Node then
        if Prefix is in Cache_Table then
            for each node n has Prefix do
                if n=Node_id then
                    Cache_Enrty_Exist_at_Node←True
                    Cache_Table_Timer←Update(Prefix/seq,Node_id,Time(10sec))
                    Schedule_Now←refresh(Prefix/seq,Node_id,Time(0sec))
                end if
                if Cache_Enrty_Exist_at_Node=False then
                    Cache_table←Insert(Prefix/seq,Node_Id,Timer)
                    Schedule(1sec)←refresh(Prefix/seq,Node_Id,Time(1 sec))
                end if
            end for
        else
            Cache_table←Insert(Prefix/seq,Node_Id,Timer)
            Schedule(1sec)←refresh(Prefix/seq,Node_Id,Time(1 sec))
        end if
    end if
end procedure
```

Figure 3: Algorithm 1 for Content Registration.

C. Content Forwarding Strategy

In the proposed architecture, we model a named flow-based forwarding strategy as part of the proposed framework as shown in Figure 4. We design a new communication pattern with a single interest for content from the consumer to the controller instead of sending interest for every data packet. The controller sends flow information to the ICN node by setting entries according to the number of chunks to be delivered into PITs, as shown in Figure 5. Each entry is created after a fixed time interval allowing an event on the producer to generate data and define the data rate accordingly. When content starts transferring from producer to consumer, the PIT entry for that content is removed. The single request delivers many chunks of data. As a result, the utilization of PIT instead of FIB enhances network performance, bandwidth utilization and reduces round-trip-time for data flow.

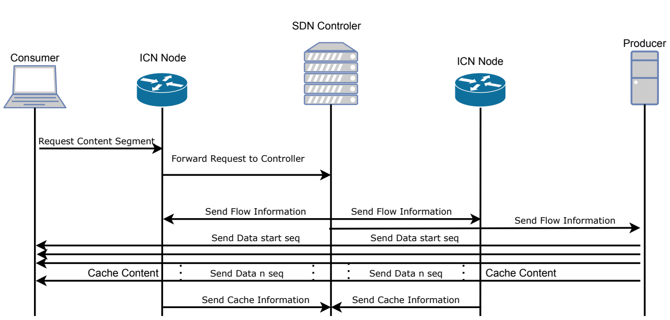


Figure 4: Forwarding Strategy

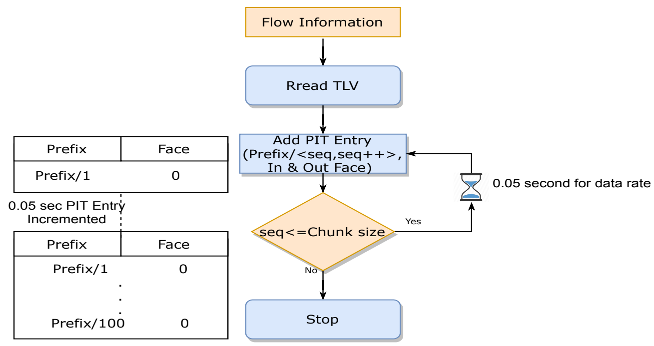


Figure 5: PIT Entry in ICN Node

D. Caching Strategy

We propose a caching strategy for IC-SDN where content is transferred from the producer to the consumer. The intermediate nodes cache the content and maintain the content table with a timer. The cached content information is also sent to the controller to register. We show the controller synchronization with the ICN node content table in Figures 6 and 7. The time continuously reduces per second, and the cached content from the table is automatically removed once the timer reaches zero. On the other hand, the controller and the ICN node update their content caching time while the content is accessed before expiry (shown in Figure 6 Algorithm 2). The content placement and replacement are in the number of the data chunk size. So it provides more locality of content with less computation required. It also supports off-path caching as the controller is aware of caching and selecting the path with the least distance to reduce content transmission delays. The content is also accessed from the node near the consumer, mainly from the edge, so popular content remains in edge nodes. Therefore, less popular content is deleted because of the timer making this Algorithm favoring edge and popular content caching.

```
procedure Refresh(Prefix,Node_Id,Time)
    for each node n is in Prefix do
        if n=Node_id & Timer! = 0 then
            Decrement Timer 1 sec
            Schedule(1 sec)←refresh(Prefix,Node_Id,Time(1 sec))
            break
        end if
        if n=Node_id & Timer= 0 then
            Delete That Entry From Cache_Table
        end if
    end for
end procedure
```

Figure 6: Algorithm 2 for Cache Refreshment Procedure.

E. Content Lookup

The controller initiates its decision-making module on receiving a request packet from the consumer for the requested content. It extracts the TLV components from the request packet and looks upon the Content-Table and Cache-Table. There are two possible events in this scenario. First, the

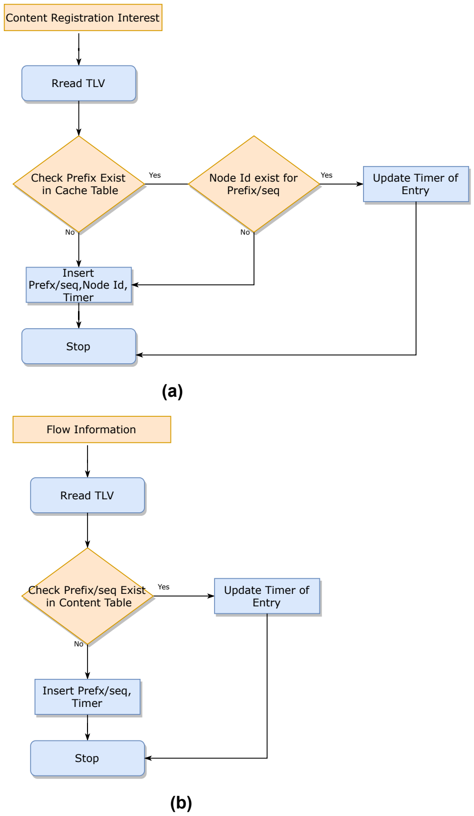


Figure 7: Flowchart of: (a) Controller Cache Information Management; (b) ICN Node Cache Information Management.

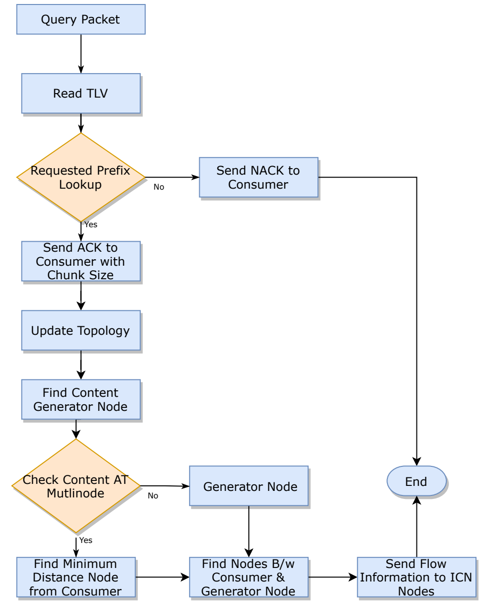


Figure 8: Flowchart of the Content Search Procedure.

```
procedure Discovery (Prefix,Node_Id)
    Input:Prefix,sequence,Consumer_ Node_Id,Control_Code
    Read TLV Prefix,Node_Id
    for each Node(x) = Consumer do
        Content_tabe\Cashe_table←Search(Prefix,Producer)
        if Search=Successfull then
            Find_Shortest_Path ← Dijistra(g,x)
            Multi_Node ← Compare_Distace_Between_Node
            if Node=Intermediate_Node then
                Cache_Table_Timer←Update(Prefix,Node_id,Time(10sec))
                Schedule_Now←refresh(Prefix,Node_id,Time(0sec))
            end if
            Consumer ← Send_Response(ACK)
            Generate_Request_Packet(Prefix,node,next_node,size)
            Intermediate_Node & Producer← Forward_Flow_Information
        else if Search! = Succesfull then
            Consumer ← Send Response(NACK)
        end if
    end for
end procedure
```

Figure 9: Algorithm 3 Content Lookup.

controller looks up the content-name on the Cache-Table. In that case, it would detect the ICN router's ID in charge of caching the content. It then calculates the most optimal path using Dijkstra algorithm to define the consumer's best optimal path to the ICN router with the cached data. Second, the content name is located on the Content-Table, but not the Cache-Table. In this situation, the controller finds the producer node and detects the shortest route from the consumer to the producer, as shown in Figure 8. The content and cache tables are defined as hash entries to achieve faster searching of content. Figure 9, Algorithm 3 shows the algorithm for content registration and discovery.

IV. Testbed Evaluation

In this section, we present the evaluation of our framework through an extensive simulation testbed environment. Our assessment is multi-folded. Firstly, we categorize the scope of our implementation to showcase meaningful experiments to ensure our implementation's feasibility. Secondly, all results in this paper are calculated with 95% confidence interval over ten runs. Finally, we explain why our solution is better than the traditional approach.

A. Experimental Setup

To validate our data-centric framework proposal, we initialize simulated testbed scenarios using ndnSIM [26], which is an open-source ICN simulator based on the NS-3 simulation framework [27]. We initialize simulated testbed using ndnSIM to carry on identical experiments to verify our solution's feasibility and effectiveness. We start by presenting our simulation environment's configuration, and later we elaborate on each experiment's details. The technical specifications for the necessary parameters are provided in Table I.

We use BRITE [28] tool to pull Deutsche Telekom (DT) topology from [29] into our modified version of the ndnSIM-based testbed. The topology consists of two layers: the core and access layer.

The core layer has 14 ICN nodes inter-connected with multiple neighboring ICN nodes. On the other hand, the access layer has 10 ICN nodes connected with numerous consumers and producers. We increase the number of consumers than the producers to achieve efficient results where each consumer

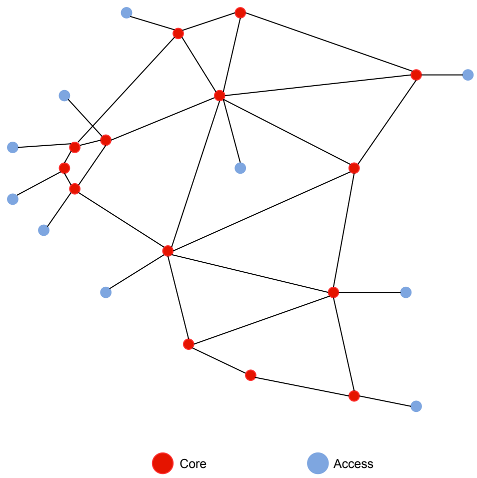


Figure 10: Generic DT Topology.

requests 20 packets per second. Finally, the controller is connected with the ICN nodes in the core and access layer. The partial view of the DT topology is depicted in Figure 10. We also use the NCC routing strategy and LRU (Least Recent Used) for the traditional ICN network because NCC [30] and LRU [31] are the ICN's default strategies.

Table I: Simulation Parameters.

Parameter	Value
No. of Nodes	134
No. of Access Nodes	10
No. of Core Node	14
No. of Consumers	100
No. of Producers	10
Link Delay	10 ms
Link Capacity	1 Gbps
Content Popularity	Zipf
Content Access	Random
No: of Unique Content	100
Each Consumer run time	20 seconds
Interest Rate	20 per second
Simulation Time	40 seconds

B. Throughput

In this section, we execute two different experiments to showcase the level of successful content delivery for 10 to 100 consumers and producers, both in traditional ICN and proposed IC-SDN implementation. Also, each of those ten producers has two types of unique content.

1. Experiment 1: Each consumer requests random content, with an interest rate of 20, and the content size is 400 packets.
2. Experiment 2: Each consumer requests 400 different content, and some content may not be available.

In the traditional ICN, interests are sent simultaneously to the producer resulting in increased round-trip-time (RTT) and unnecessary bandwidth consumption. Therefore we design the IC-SDN framework to retrieve more content with less interest packet solves this issue by transmitting less interest. Therefore, we compare the throughput based on successful data delivery percentage against an increasing number of consumers. The comparison results are shown in Figure 11. In the proposed system, a single request returns multiple chunks of contents, decreasing the packet redundancy issue in the system, unlike the traditional ICN, where packet redundancy impacts performance degradation on the system. Similarly, the controller installs multiple flows for a single request. Also, we utilize the PIT to keep data moving from content producer to consumer after installing flows.

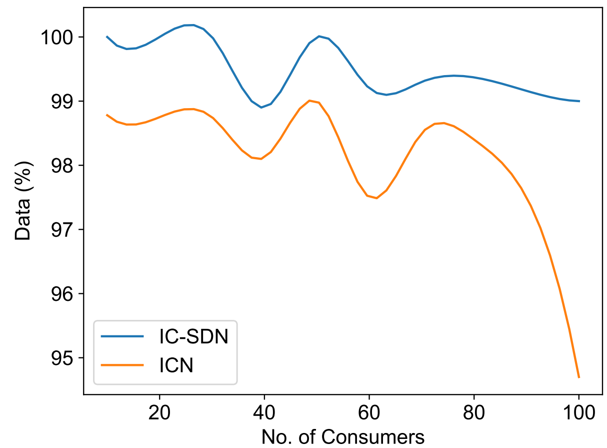


Figure 11: Throughput

In the second experiment, we evaluate the throughput performance based on the data packet processing between the proposed IC-SDN and pure ICN framework. The comparison result is shown in Figure 12, favoring the IC-SDN to achieve a higher rate of data processing than pure ICN. The pure ICN struggles as the number of consumers increase due to the constant sending of interest to ICN nodes that result in flooding on the routing table. Conversely, the proposed system checks the availability of data and then installs PIT entries that enable data movement in the network. The process reduces interest flooding and achieves better utilization of resources. The pure ICN performs better when the number of requests for each consumer is decreased to the minimum. The higher number of interests leads to performance degradation due to the higher RTT and on-path caching principle, which does not impact our proposed IC-SDN system.

C. Cache Hit

In the third experiment, we observe the cache hit ratio for 10 to 100 consumers while randomly requesting 10 producers. Each producer has two different prefixes containing 400 packets, and we assign 1 MB cache size to the intermediate ICN nodes. Figure13 shows the cache hit ratio between the proposed IC-SDN and the pure ICN strategy. The proposed strategy performs better than the conventional LRU strategy when the number of consumers increases. The pure ICN's cache hit ratio is reduced due to increasing requests requiring the contents to be cached to satisfy more requests. Again,

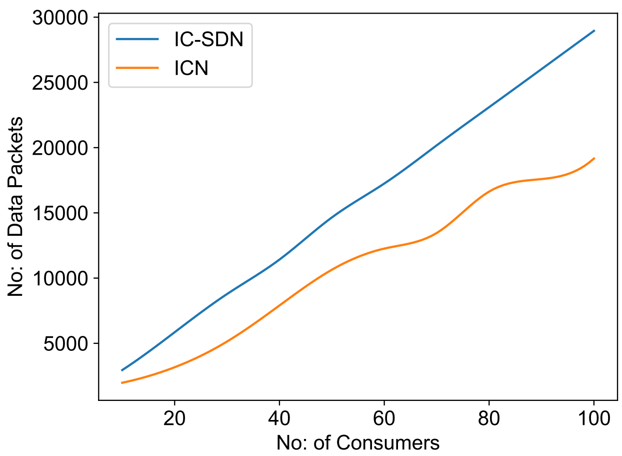


Figure 12: Throughput with Content Miss

ICN's on-path caching principle introduces a significant number of missed caches while multiple consumers keep sending requests to the ICN nodes. In contrast, the IC-SDN supports off-path caching, and the controller has the cached content's visibility. Therefore, the controller optimizes the caching hit ratio on the system as the number of requests increases.

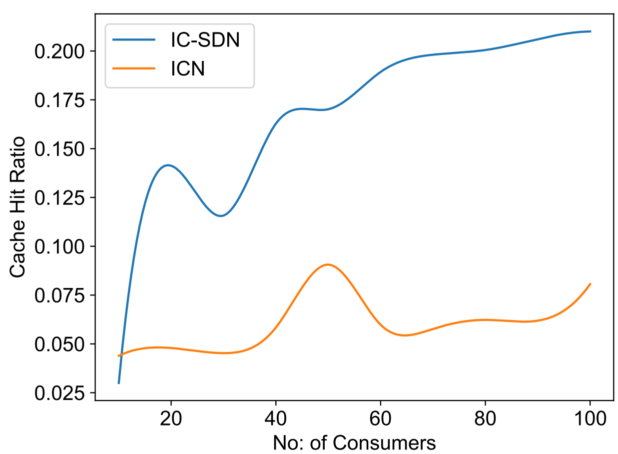


Figure 13: Cache Hit

D. Off-Path Cache

In addition to the previous experiment, we perform a simple test to demonstrate off-path caching support in the present section. The screenshots from the experiment are taken from the ndnSIM visualizer. Figures 14 and 15 shows the off-path caching support as compared to pure ICN's on-path caching.

E. Packet Delay

We calculate the round-trip-time between consumer and producer using the topology presented in Figure 10. The proposed scheme reduces RTT for interest sent for every data packet. We use a single request for a chunk of data and utilizes PIT instead of FIB. This modification achieves less amount of interest flows in the network. The controller sets the data rate and initializes PIT entries in the ICN nodes according to the consumer's request ratio. The data travels to the requested consumer from the producer based on the rate set by the controller. Again, this rule applies to the fresh content, not the cached ones. The proposed PIT-based forwarding performs intelligent data routing among the participant nodes, improving the network's latency and overhead. Figure 16 shows the RTT performance comparison between the IC-SDN and pure ICN solution, where we calculate the routing latency against the increasing number of hops.

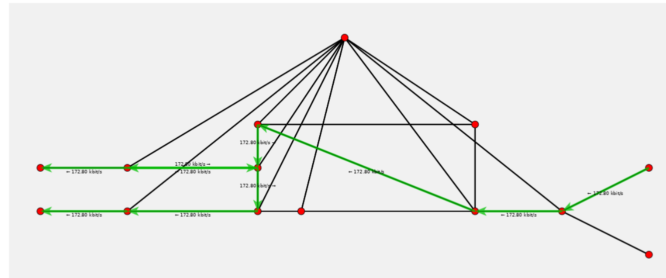


Figure 14: ndnSIM visualizer screenshot showing off-path Cache Access IC-SDN

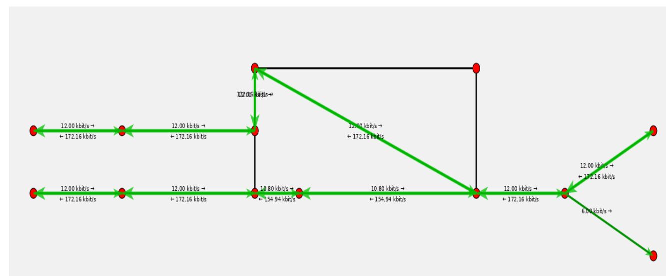


Figure 15: ndnSIM visualizer screenshot showing only in-path Cache Access ICN

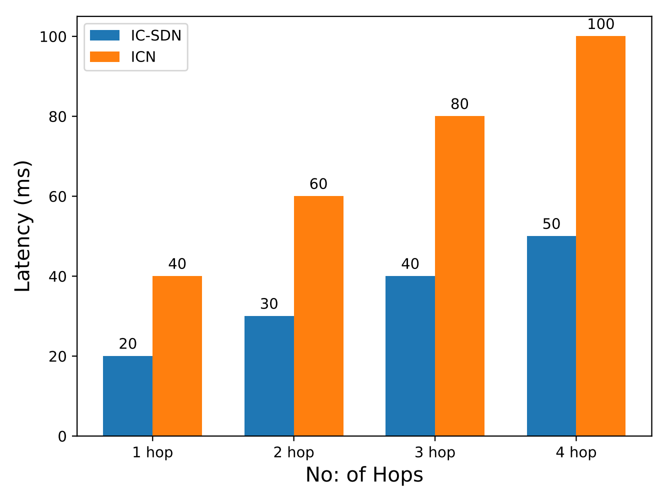


Figure 16: Round-Trip-Time

F. Interest Overhead

The proposed system has less interest overhead than pure ICN because the IC-SDN system access data in chunks. The ICN uses an interest packet for every data packet while we access the data in chucks. To access 100 data packets, IC-SDN uses only one interest packet while ICN uses 100 interest packets. We collect the interest overhead of proposed and pure ICN using simulation. We found that 10 consumers inquire 4000 data packets use 4000 interests in pure ICN. In comparison, the proposed IC-SDN solution requires 40 interests at 1% interest overhead compared to pure ICN to process the same amount of contents in the network.

V. Conclusion

The presented work explores some of the existing problems the generic SDN-ICN principles face and proposes a clean-slate IC-SDN controller-driven solution that provides efficient routing and caching of the contents using a PIT-based stateful forwarding. It reduces the overhead of both fresh and requested interest packets that originate from a single or multiple consumers. Besides, the novel in-network caching utilizes the producer's available content and brings them closer to the consumer. The controller installs a single flow per one unique content reducing the additional overhead from the intermediate ICN nodes. The ndnSIM-based proof-of-concept implementation and evaluation results exhibit the design flexibility and performance efficiency of the proposed IC-SDN solution highlighting higher throughput and lower latency than traditional NDN which is one of the mostly adopted ICN implementations.

References

[1] https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf

[2] Vasilakos, Athanasios V., et al. "Information centric network: Research challenges and opportunities." Journal of network and computer applications 52 (2015): 1-10.

[3] Nour, Boubakr, et al. "NNCP: A named data network control protocol for IoT applications." 2018 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2018.

[4] Mun, Ju Hyoung, and Hyesook Lim. "On Sharing an FIB Table in Named Data Networking." Applied Sciences 9.15 (2019): 3178.

[5] Nour, Boubakr, et al. "ICN publisher-subscriber models: Challenges and group-based communication." IEEE Network 33.6 (2019): 156-163.

[6] Ioannou, Andriana, and Stefan Weber. "A taxonomy of caching approaches in information-centric network architectures." Elsevier Journal (2013).

[7] Kreutz D, Ramos F M, Verissimo P E, et al. Software-defined networking: A comprehensive survey [J]. Proceedings of the IEEE, 2014, 103(1): 14–76.

[8] Zhu, Liehuang, et al. "SDN Controllers: A Comprehensive Analysis and Performance Evaluation Study." ACM Computing Surveys (CSUR) 53.6 (2020): 1-40.

[9] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[10] Latif, Zohaib, et al. "A comprehensive survey of interface protocols for software defined networks." Journal of Network and Computer Applications 156 (2020): 102563.

[11] Liu, Yaoqing, and Hitesh Wadekar. "SDAR: Software Defined Intra-Domain Routing in Named Data Networks." 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA). IEEE, 2016.

[12] Li Z, Xu Y, Zhang B, et al. Packet forwarding in named data networking requirements and survey of solutions[J]. IEEE Communications Surveys & Tutorials, 2018, 21(2): 1950–1987

[13] Din I U, Hassan S, Khan M K, et al. Caching in information-centric networking: Strategies, challenges, and future research directions[J]. IEEE Communications Surveys & Tutorials, 2017, 20(2):1443–1474

[14] Mangili M, Martignon F, Paris S, et al. Bandwidth and cache leasing in wireless information-centric networks: A game-theoretic study[J]. IEEE Transactions on Vehicular Technology, 2016, 66(1): 679–695.

[15] Gheisari, Mehdi, et al. "Mapp: A modular arithmetic algorithm for privacy preserving in iot." 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC). IEEE, 2017.

[16] Zhang X, Zhu Q. Information-centric virtualization for software-defined statistical qos provisioning over 5g multimedia big data wireless networks[J]. IEEE Journal on Selected Areas in Communica- tions, 2019, 37(8): 1721–1738.

[17] Gheisari, Mehdi, Guojun Wang, and Shuhong Chen. "An edge computing-enhanced internet of things framework for privacy-preserving in smart city." Computers & Electrical Engineering 81 (2020): 106504.

[18] Torres J V, Alvarenga I D, Boutaba R, et al. Evaluating cros-ndn: a comparative performance analysis of a controller-based routing scheme for named-data networking[J]. Journal of Internet Services and Applications, 2019, 10(1): 1–24.

[19] Afanasyev, Alexander, et al. "NFD developer's guide." Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021 (2014).

[20] Lv J, Wang X, Huang M, et al. Risc: Icn routing mechanism incorporating sdn and community division[J]. Computer Networks, 2017, 123: 88–103

[21] Jmal, Rihab, and Lamia Chaari Fourati. "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy." Computer Standards and Interfaces 51 (2017): 22-29.

[22] Xiulei W, Ming C, Chao H, et al. Sdicn: A software defined deployable framework of information centric networking[J]. China Communications, 2016, 13(3): 53–65.

[23] Detti A, Blefari Melazzi N, Salsano S, et al. Conet: a content centric inter-networking architecture [C]. Proceedings of the ACM SIGCOMM workshop on Information-centric networking. [S.l.: s.n.], 2011: 50–55.

[24] Arora, Pranav. "The Heart and Brain of SDN: SDN Controllers." Innovations in Software-Defined Networking and Network Functions Virtualization. IGI Global, 2018. 100-126.

[25] johnson D B. A note on dijkstra's shortest path algorithm[J]. Journal of the ACM (JACM), 1973, 20(3): 385–388.

[26] Mastorakis, Spyridon, Alexander Afanasyev, and Lixia Zhang. "On the evolution of ndnSIM: An open-source simulator for NDN experimentation." ACM SIGCOMM Computer Communication Review 47.3 (2017): 19-33.

[27] Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator." Modeling and tools for network simulation. Springer, Berlin, Heidelberg, 2010. 15-34.

[28] Medina, Alberto, et al. "BRITE: An approach to universal topology generation." MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. IEEE, 2001.

[29] Knight, Simon, et al. "The internet topology zoo." IEEE Journal on Selected Areas in Communications 29.9 (2011): 1765-1775.

[30] J. Shi, "ccnd 0.7.2 forwarding strategy," https://redmine.named-data.net/projects/nfd/wiki/CcndStrategy, University of Arizona, Tech. Rep.

[31] 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.