Secure SDN-based In-network Caching Scheme for CCN

Amna Fekih,
Isitcom,
GP1 H-Sousse, Tunisia
emna.fekih@isetso.rnu.tn

Sonia Gaied,
ESST,
Abassi H-Sousse, Tunisia
soniagaied3@gmail.com

Habib Youssef,
CCK,
Univerity campus of Manouba,
Tunisia habib.youssef@fsm.rnu.tn

Abstract—**The different technologies deployment for in-formation exchange on the network and the terminal diver-sity that support them, produce users unwittingly attracted to the Internet that is everywhere in their daily lives. Users require a high Quality of Service (QoS) especially for video streaming applications that generate the main part of the internet traffic. For service providers, it is important to reduce this traffic and increase the Quality of user Experience (QoE). To meet these needs, two additional keys must be exploited: routing and caching. Content Centric Networks (CCN), which is one of the most developed Information Centric Networks (ICN) approaches, offers efficient management of caches at CCN nodes, but the lack of scalable routing is one of the obstacles that slow down its deployment at the internet level. Then, Software Defined Networks (SDN) architecture has been proposed to facilitate programming of the network and to automate the management of the complex architecture but without taking advantage of the capacity of the intermediate caches of the connected nodes. To accomplish tasks, a CCN node uses three tables, Content Store, Forwarding Information Base, and Pending Interest Table, which present attacks to privacy of content requesters and producers. In this paper we propose a Secure In-network Caching Scheme (SICS) for CCN networks based on SDN architecture taking advantage of its global vision. SICS does not only improve cache hit rate and reduce response time, but it also allows to monitor access to CCN routers and detect attacks in order to block them. The results of our model implemented in NS-3 simulator will be presented and evaluated.**

Keywords- CCN; SDN; in-network caching; popularity; Security.

I. INTRODUCTION

The Internet architecture has long been based on the concept of a stack of independent protocols, which requires that all data exchanges via this network must be carried out by establishing communication channels between the network equipment. They are end-to-end, host-centric communications. Today, the Internet is no longer a small network connecting a small community of researchers but it is a large global network connecting virtually all people and organizations around the world. Its use itself is changed too. The evolution of software technologies, improved storage media capabilities, the heterogeneity of terminals and connected equipment, and the nature of frequently used applications such as video streaming [1]. All these factors push researchers to revise, modify and even radically change the current architecture in order to meet the needs of users and service providers without offering temporary solutions to attach them to the Internet architecture.

In recent decades, a simple analysis of network traffic allows us to see that streaming video applications either live or on demand are increasing explosively with the increase in the number of users connected to the Internet. These applications are very greedy in terms of spatial and temporal resources. CCN [2]-[4] is an innovative paradigm that attracts the atten-tion of several researchers. Its principle is to provide content to users instead of establishing communication channels as-sociated with its requests. It aligns with the trend of Internet usage. Users do not look for where and how the requested content will be delivered but they check and compare when at what quality will be received. CCN architecture saves energy consumption and achieves green communication as it dramatically reduces distances by pushing content to users. The important issue in CCN is how to store and manage data packets in local caches of CCN nodes in order to optimize network performance and improve the quality perceived by users. Therefore, this problem is discussed in this article to minimize total network traffic. CCN that is based in-network caching using caches operated at the nodes (Content Store) is able to respond adequately to users' expectations. But the deployment of CCN equipment in existing networks is a critical issue as their design strategies are totally different. In this context, the SDN [5][6] architecture presents a real opportunity, via network programming, to introduce the CCN functionality such as caching and to improve it. The rest of this paper is structured as follows: Related work will be detailed in Section II. The next section describes original CCN and SDN architectures. Section IV provides an overview of our model and details its specifications. The results of our model will be presented and evaluated in Section V. Finally, this work will be closed by a brief conclusion introducing our future work in Section VI.

II. RELATED WORK

In the last years, several works are focused on in-network caching. Although all are aimed at the same objectives, the techniques used are different. [9] exploited the cooperation of the neighbor. The principle was to select cluster headers. Then, they perform cooperative mechanisms of redundancy and elimination avoidance to optimize caching performance. In [10], the authors proposed an admission policy based on a

 ISBN: 978-1-61208-669-9

content discovery protocol that allows the router to decide whether the transported content should be cached or not. WAVE [11] is also a system that integrates router collaboration to suggest caching of content blocks to the downstream CCN router. The latter makes caching decision by examining the content polarity and inter-chunk distance. [12] proposed an SDN-based forwarding strategy in CCN as an alternative to the basic strategy that causes additional network traffic due to the flooding of interest packets. For this purpose, the authors used two types of routers, intermediate and gateway routers. The latter are responsible for interconnecting clusters and redirecting inter-cluster interest packets thus minimizing the traffic on the network. [13] split the network into autonomous systems (AS). Each AS selects a control node based on betweenness centrality and cache replacement rate. Their idea is to map SDN features in a CCN network to improve cache performance by exploiting a cache table managed by control nodes.

In CCN, no data packet will be transported on the network if it did not have an explicit request initiated by an Interest packet. These explicit requests slow down DoS attacks [14]-

Although CCN relies heavily on cryptography to authenticate content, the dynamic nature of CCN routers produces vulnerabilities related to these interior tables. The Pending Interest Table (PIT) is critical since it exploits a stateful routing of CCN network. PIT attack [17] is both easy to perform and difficult to detect. A hacker can flood the PIT table with requests for any content that results in the denial of service of pending communications. In contrast to the limited IPv4 / IPv6 address space, the CCN space is delimited opening the door to FIB attacks by generating and publishing content belonging to non-routable domains on CCN network. An adversary looking to decrease caching system performance uses a content store attack DoS. Fortunately, this attack is difficult to achieve since it is greedy in bandwidth.

Our approach is a hybrid solution inspired by the works already mentioned. It is characterized by: i)The control is provided by local SDN controllers whereas CCN routers are simple forwarding devices. ii)A distributed content popularity base that is filled and operated by local SDN controllers. iii)A secure policy-based forwarding strategy defined in [12] where the FIB tables are managed only by the local SDN controllers. iv)A collaboration between CCN routers and their local SDN controller defined in [13] is adopted to ensure the visibility of content stored for CCN routers in the same cluster. v)A new popularity-based caching strategy that associates each content with local popularity according to interest packets processed by transited CCN router. Content may be marked by global popularity. vi)Caching is supplemented by replication and replacement policies based on content popularity, number of CCN router claims (probe packet), and number of hops to retrieve the data.

III. BACKGROUND

In this section, we describe the CCN and SDN architectures. Table 1 presents the different aspects that motivate our work [7][8].

TABLE I. TRADITIONAL IP NETWORKS VS CCN - SDN

	Traditional IP Networks	**Content Centric Networks**	**Software Defined Networks**
Control	Tightly coupled control and data planes	Forwarding strategy is separate from the routing strategy	Decoupling network control and forwarding functions
Native cache	Not supported	Supported (key to success)	Not supported
Routing	Hybrid	Distributed between CCN routers based on prefixes	Centralized, managed by the SDN controllers

A. CCN Architecture

Content Centric Networks is the most popular ICN architecture that motivates many researchers with its in-network caching feature.

Naming: In CCN, names are hierarchical. This architecture uses name prefix aggregation to ensure routing scalability. Each content publisher uses a unique prefix to name each content before publishing it. CCN defines two types of packets. Clients use interest packets (INTEREST) to express their requests by specifying the name and the data packet (DATA) is the response provided by any cache router with a copy of the requested content or by the data source.

Routing and Transport: In CCN, each router contains three tables. Each is characterized by one or more well-defined functionalities. The Content Store (CS) is used to cache Named Data Object (NDO). The Pending Interest Table (PIT) that contains a list of pending interests ensures data delivery and the Forwarding Information Base (FIB), which ensures requests forwarding. Subscribers send INTEREST packets to express their requests by specifying the requested NDO that arrive as DATA packets. The data is sent only in response to an interest.

Caching: Unlike other ICN architectures that enable or disable caching, CCN natively supports on-path caching. The Content Store (CS) is a fundamental component in CCN that can be compared to the buffer in IP routers but with a persistent data storage capacity. When an Interest packet is received, each content router verifies its CS first to deliver the requested content directly from its local CS. And when it receives a DATA packet, it stores the transported data in its CS according to the defined caching policy. If there is no match in CS, content router uses the FIB table to forward the request and the PIT table to keep all received interest packets

 ISBN: 978-1-61208-669-9

requesting the same NDO. This removes unnecessary transfers on the network.

Security: CCN uses a human-readable hierarchical namespace to improve routing scalability. Security is provided by the content publisher by signing each DATA packet with its secret key. This packet contains a signature over the name, the information included in the message and information about the key used to produce the signature. So the naming in CCN does not contain the publisher key which makes self-certification impossible. The trust in the signing key must be established by external means.

B. SDN Architecture

The technological revolution has made available a diversity of services and a range of equipment and devices that exceed the capacity of the current Internet. This is obvious since the internet was not designed to support these technologies. SDN is a new emerging network paradigm that presents the results of research into virtualization and automation solutions for hardware and software resource management. Open Networking Foundation describes SDN architecture that consists of three layers and different types of APIs allowing communication between them.

Layers and Open API: From the bottom up, the infrastructure layer that consists of network devices that are simple forwarding devices. The control layer is the heart of the SDN architecture. It includes SDN controllers. The latter exploit Open-API to control and manage the forwarding behavior on the network. They communicate via interfaces: southbound, northbound and east/westbound interfaces. The last, application layer where we find user applications such as network virtualization, monitoring and network application.

Features of SDN:

Data/Control planes are decoupled: The decoupling of two planes facilitates the fast automated management and reconfiguration of forwarding devices according to the state of the network.

Logical centralized control: Via its interfaces, the SDN controller can collect information and build a global view of the network.

Network programmability: It is the key feature of the SDN architecture. It ensures scalability and encourages innovation.

Dynamic updating of forwarding rules: This feature aligns with the objectives of managing network resources. enhancing configuration and improving performance thanks to the global knowledge on the network.

OpenFlow Protocol: is the most popular protocol for communication between the infrastructure/control layers. It standardizes information exchange between controller and forwarding devices. Each forwarding device or OpenFlow switch communicates with a controller via secure channel. Each contains one or more flow tables themselves consist of flow entries. These determine how packets will be processed and transmitted.

IV. SECURE SDN-BASED IN-NETWORK CACHING SCHEME

In this section, we describe our secure caching scheme and detail its different features.

A. Overall Architecture

We propose a Secure In-network Caching Scheme for CCN networks based on SDN architecture (SICS) in order to reduce response time, minimize network traffic and improve user perception. SICS is depicted in Figure 1. CCN routers and gateways are located in the data plane with a simple role of forwarding without any control. The control layer consists of SDN controllers.

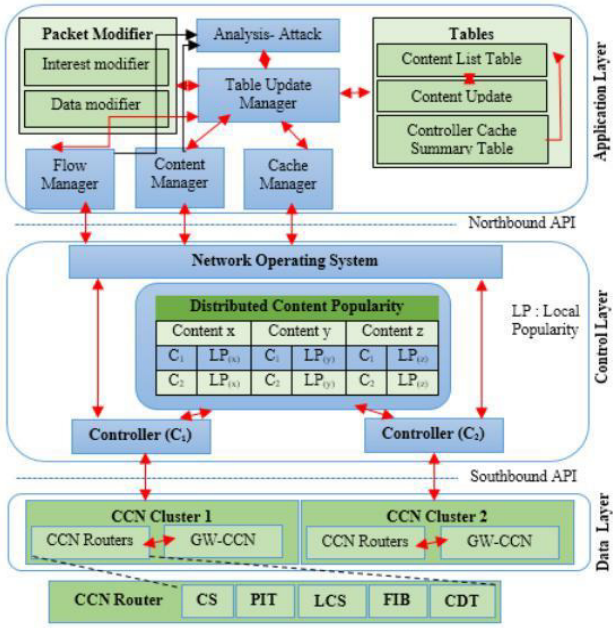

Figure 1. Secure SDN-based In-network Caching Scheme

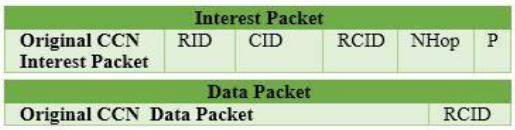

Figure 2. Interest and Data packets

Each local controller SDN forms a network global view on its controlled area and manages it by communicating with the forwarding devices by OpenFlow protocol. Automated management and dynamic reconfiguration of network resources are assigned to managers and programmable modules that are specified in the application layer. The module analysis-attack Detection allows the controller to supervise the contents circulating on the network and to block attacks if detected. Flow manager and Update Manager Table are responsible for the flows and routes forwarding of interest packet (respectively data packet) from original producer or any CCN cache to requester. Content Manager manages the transferred content. On the other hand, Cache

 ISBN: 978-1-61208-669-9

Manager and Update Manager Table are responsible for managing and updating the list of contents cached on CCN routers. These are the managers who update Content List Table, Content Update Table, and Controller Cache Summary Table accordingly when they detect network changes.

B. *Proposed Packet Types*

Our system organizes the packet types into two categories. The first presents the forwarding packets exchanged between clients, CCN routers and SDN controllers. The second includes management packets exchanged between CCN routers and SDN controllers to retrieve, manage, and update CCN caches.

1. *Interest Packet*: Upon receiving client request, the CCN router looks for the content name specified in the packet interest in CS and then in Local Cache Summary Table. In case of cache hit, the response will be transferred to the client. Otherwise, CCN router sends this interest packet to its local controller (as packet_IN in original OpenFlow protocol). To guarantee the transfer of the request from local area to another in case of cache miss, the original CCN interest packet is extended by five fields (Figure 2) to globally identify a request on the network.
2. *Data Packet*: To deliver a requested content, two cases will be possible. the content exists in the same controlled area or another. Then, we adopt the proposal of [12] to add a new field, RCID, to the original CCN data packet to specify the controller ID that initiated the request since the local areas are interconnected by gateways. The RID field (Figure 2) specifies the identifier of CCN router that received the request. The local controller of the area where this CCN router belongs will be recorded in the CID field. The RCID field contains the identifier of the controller where the first CCN router belongs, which will begin the retrieval of requested content. In addition, we add another field, NHop, which presents the number of routers crossed during the delivery of data packet in order to require caching or not. NHop is initialized to 0. While the data packet is transferred to a router belonging to the same controller, NHop will be incremented by 1. If it is forwarded to a router belonging to another controller, NHop will be incremented by 1+ SUM (CCN routers in local controller). The last P field specifying the perimeter of content search. P is local if it is a request to retrieve content from content store of a router belonging to the same local controller. Otherwise, P is extended.
3. *Report Packet*: To inform their local controller of the state of the local caches, the attached CCN routers send periodically the state of their CSs. However, when the CS is changed, the CCN router reports the CS update directly to its controller. CCN router also uses this packet to claim that the CS is full so that the local controller executes the proper content replacement strategy.
4. *Update Packet*: Collecting CS summaries allows the local controller to form a complete view of cached content in con-trolled area, calculate popularity, and replicate when needed. Each local controller advertises its CCST table by sending update packet periodically to its CCN routers. Otherwise, local controller sends an update packet to a specified CCN router requesting replication or delete of a particular content based on its popularity, number of probe packets and the number of hops to deliver it.

Content Store			
Content name	Created_date	Last_update	Data

Forwarding Information Base				
Content name	RID	CID	Requesting router	Requesting controller

Pending Interest Table			
Content name	Created_date	ports	nports

Cache Decision Table			
Content name	RID	CID	CS-MGT

Controller Cache Summary Table / Local CST				
Content name	RID	CID	Priority	Face

Figure 3. CCN router Tables

Algorithm 1 : Caching process

Input: Interest Packet IP, Requested Router RR, Content C
local popularity LP, threshold
Output: Caching decision

receive Interest packet
compute local popularity of this content in Requested Router $LP(C)_R$
update the local popularity of this content at Controller level $LP(C)_C$
if ($LP(C)_C$ > MAX ($LP(C)_{Ci}$) **then**
mark interest packet with global popularity
endif
if (interest packet has global popularity) **then**
$R_{selected}$ ← betweeness Centrality cache (local controller)
$CS\text{-}MGT_{(Rselected)} \leftarrow 1$ -- $R_{selected}$: Selected Router
endif
if ($LP(C)_{(RR)}$ == threshold) **then**
  if (C in CCST) **then**
    if (priority == G) **then** -- Global popularity
    $CS\text{-}MGT_{(RR)} \leftarrow 0$
    else
    $CS\text{-}MGT_{(RR)} \leftarrow 1$
    $CS\text{-}MGT_{(CR)} \leftarrow 0$ --cache router avoid cache redundancy
    endif
  else
    $CS\text{-}MGT_{(RR)} \leftarrow 1$
  endif
endif

Figure 4. Caching process

5. *Probe Packet*: If CCN router sends a packet interest to local controller, a new entry is added to its PIT table specifying the requested content name and its creation date. Then, a timeout is executed. Once it reaches the value 0 and the router does not receive a data packet. The later sends a probe packet to controller to signal response delay.
6. *Attack_Avoidance Packet*: when SDN controller detects an attack during the content flow analysis, it blocks the source of this attack and forward results to all its CCN routers by an attack_avoidance packet to mark this source.

C. *CCN Router Structure*

The basic CCN router includes three types of tables: Content Store, Pending Interest table, and Forwarding

 ISBN: 978-1-61208-669-9

Information Base. Content Store (CS) where the content chunks are stored to minimize latency. Pending Interest table (PIT) stores pend-ing requests grouped by name content to reduce network traffic and avoid gateway-level congestion. Forwarding Information Base (FIB) is the forwarding table that plays the same role as FIB in IP protocol. When receiving an interest packet, CCN router first checks its CS. If the content name exists, it provides this content directly to the requester. Otherwise, it checks in PIT. If it exists, it adds an entry specifying the port and deletes this interest packet. If there is no corresponding entry in PIT, then it creates a new entry in PIT and FIB table to transfer to other routers. To achieve our goal, we enrich CCN router with two tables to manage the content store properly and reduce the latency of users. These tables are issued by the local controller. The FIB table is also entrusted to SDN controllers. Upon receiving a data packet, the CCN router checks the CS_MGT field. If it is set to 1 in Cache Decision Table (CDT), this content is saved in CS. Otherwise, the CCN router checks the PIT table and sends the content to the ports that requested. Our system assumes that the network is divided into clusters where each is administered by a local controller. Only the local controller that collects the caching information by the exploit of report packets received from CCN routers when CS is updated. For that we propose Local Cache Summary Table (LCST) in order to publish the CS information between CCN routers present in the same cluster. Local controller groups this information into Controller Cache Summary Table (CCST). In other words, having like a caching cooperation managed by local controller to improve use of cache resources in the same cluster. Interest and data packets structures are presented in Figure 2. Whereas, Figure 3 details tables format in CCN router already described.

D. Forwarding Process

The packet forwarding process in SICS is summarized by the activity diagram in Figure 5. A user requests desired content by sending an interest packet to its home CCN router. When this packet is received, our CCN router works in the same way as a basic CCN node for CS and PIT tables. If there is no corresponding entry in PIT, it creates a new entry in PIT. Thereafter, it verifies LCST. If there is an entry for the re-quested content. Then, the P field of the packet interest is set to 0 (default value for external perimeter initialized to 1). Finally, the router checks if there is a FIB entry then it transfers interest packet. Otherwise, it transfers the request to its local controller. The latter exploits two forwarding mechanisms according to the location of data packet. When a request is transferred to the local controller. It first checks P field. Two cases are possible. If P equal 0, then the forwarding mechanism is limited in the cluster managed by local controller. After that, the local controller looks in its CCST table for the corresponding entry. If it exists, it modifies the interest packet and calculates the requested content popularity. Interest packet will forward to a router belonging to the same controller, the number of hops (NHop) will be incremented by 1. Finally, install the rules in FIB and CDT of CCN router while the RID router that stores the requested content is not found. If P is not 0, the forwarding mechanism is beyond the scope of the local controller. Local controller checks CLT to retrieve the default values and then modify the interest packet to retrieve the data packet stored elsewhere. Interest packet will forward to a router belonging to another controller, NHop will be incremented by 1+ SUM (CCN routers in local controller). Finally, it installs the rules in FIB and CDT tables. When receiving a data packet, CCN router checks if there is a match in its PIT table. If it exists, it delivers the packet to the client. Otherwise, data packet is forwarded to its local controller to process it. The latter specifies the requesting router and injects the corresponding route into FIB table.

E. Caching Process

In SICS, a CCN Router stores content in its CS if and only if its local controller has validated caching by setting the CS-MGT field to 1 in the CDT table. Whenever the local controller receives a packet interest, it calculates and updates the local popularity of requested content. For effective use of CCN caches, it then compares this probability with the local popularities of this interest packet in other controllers through distributed content popularity base. If local popularity is greater than the maximum of local popularities in other clusters, then local SDN controller marks this interest packet with a global popularity. Our caching strategy is detailed in Figure 4. Two cases are possible. If interest packet is marked with global popularity, then local SDN controller executes betweenness centrality cache to select the best location (selected router) and validates caching by setting the CS-MGT field to 1 in the CDT table of the selected router. If interest packet reaches the local popularity threshold, then local controller verifies its existence in CCST table. If it is present and the associated priority is set to G (Global), then it sets the CS-MGT field to 0. Otherwise, local controller validates caching by setting the CS-MGT field to 1 for requesting router and asks cache router to remove this content and avoid cache redundancy.

F. Replication Process

Our caching strategy helps to store the most popular content and exclude unpopular content. Our replication policy takes into account the stored content and seeks to improve the quality perceived by the users. It allows caching of non-popular content based on the number of probe packets that are sent from CCN router to the local controller to warn it of past latency. Each local SDN controller periodically analyzes the received probe packets. Two cases are possible. The requested content is popular so SDN controller checks its CCST table and calculates the time needed to deliver it according to NHop field. It decides content replication if the timeout exceeds the average wait time in the cluster. The content is not popular so SDN controller calculates the number of probe packets received for this content. If it reaches the critical threshold of satisfaction, it selects the CCN router having the maximum popularity for this content where it decides its caching.

 ISBN: 978-1-61208-669-9

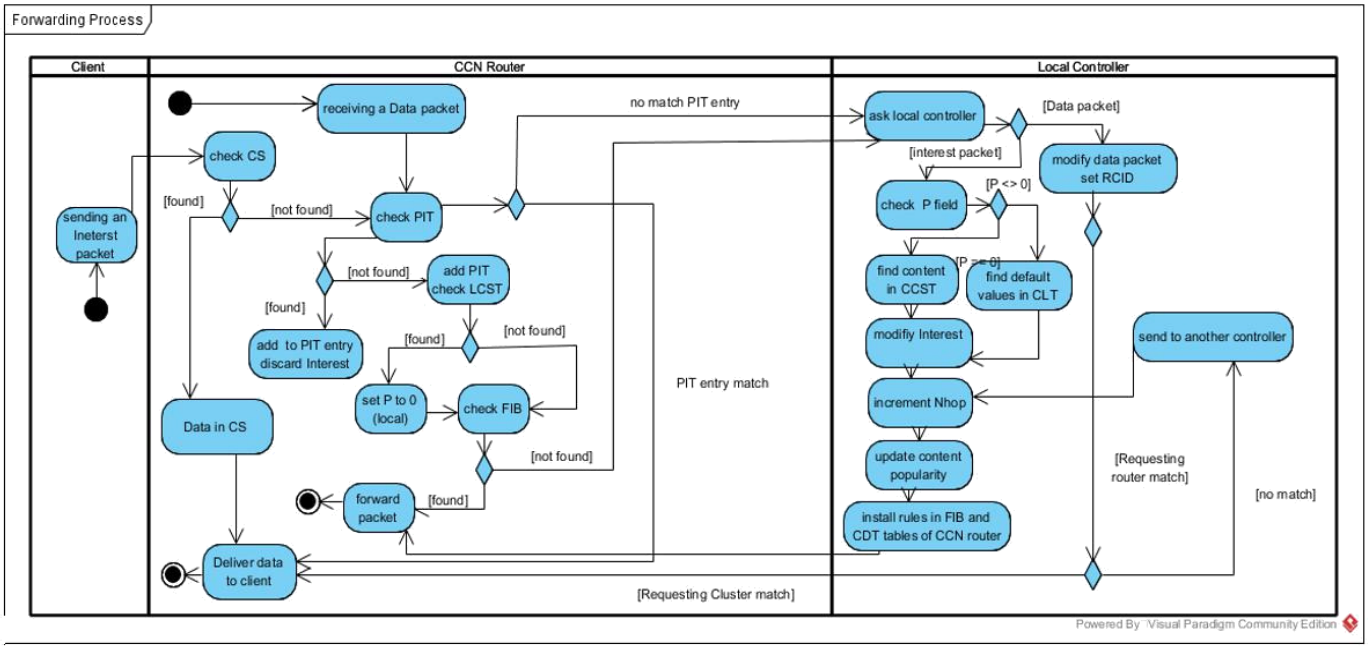

Figure 5. Activity diagram for forwarding process

G. *Replacement Process*

If CS is full, CCN Router sends a Report packet to its local controller that runs the appropriate content replacement policy. First, local controller classifies the contents stored in CS according to their popularities, local or global. Then, they will be put in order. Content with global popularity will be at the top of CS table. While the others will be added at the end of the table. Subsequently, the exclusion will always occur from the end of CS table.

V. SIMULATION AND EVALUATION

In this section, we evaluate the forwarding and caching strategies proposed in CCN based on SDN. We focus on evaluating the efficiency of content delivery, the network traffic rate, the average number of hops during content delivery, and the usage rate of caches. Performance is evaluated using NS3 simulator [21] by integrating OpenFlow and ndnsim modules.

A. *Parameters used for simulation*

The parameters used in the simulation are shown in Table 2. Using the important parameters, we evaluate the average time for content delivery, network traffic, cache hit rate and average hop count compared to original CCN paradigm (BasicCCN), Autonomous System Collaboration Caching Strategy (AS-CCS) [13] and Forwarding Strategy on SDN-based Content Centric Network (FS) [12].

B. *Simulation Results*

With simulation results part, we demonstrate four main comparisons. One is the comparison of the time taken to get data for requests between the proposed forwarding strategy SICS and others. The other is the comparison of total number of interest packets in the network according to different number of requests. Figure 6-a shows the comparison of the time taken for data delivery. It is proved that the proposed forwarding mechanism takes less time for content delivery, which means that it is more efficient than other forwarding strategies such as ASCCS, FS and BasicCCN. In Figure 6-b, as the number of requests gradually increases from 100 to 500 the amount of traffic caused by the packets of interest increases considerably for the BasicCCN strategy since it broadcasts interest packets to all other neighboring routers. On the other hand, ASCCS which exploits the neighbors' cooperation and FS which is characterized by the global vision on the network have almost the same increase for interest packet. But our proposed routing strategy that combines the benefits of other strategies does not have as much impact on the total amount of traffic as BasicCCN does. Figure 6-c shows that SICS can dramatically improve the success rate of the cache, which is shown in the graph clearly compared to the basic CCN architecture and architectures adopted. By increasing the size of the caches the redundancy of the contents will be more possible. Figure 6-d shows that our SICS policy works better than the others by avoiding redundant content in each cluster.

VI. CONCLUSION

In this paper, we proposed a novel Secure In-network Caching Scheme for CCN networks based on SDN architecture (SICS) in order to reduce response time, minimize network traffic and improve user perception. The proposed strategy enables to overcome the problems that the original CCN faces with the help of centralized table management for packet forwarding. Furthermore, simulation result was shown to prove that the efficiency of the average

 ISBN: 978-1-61208-669-9

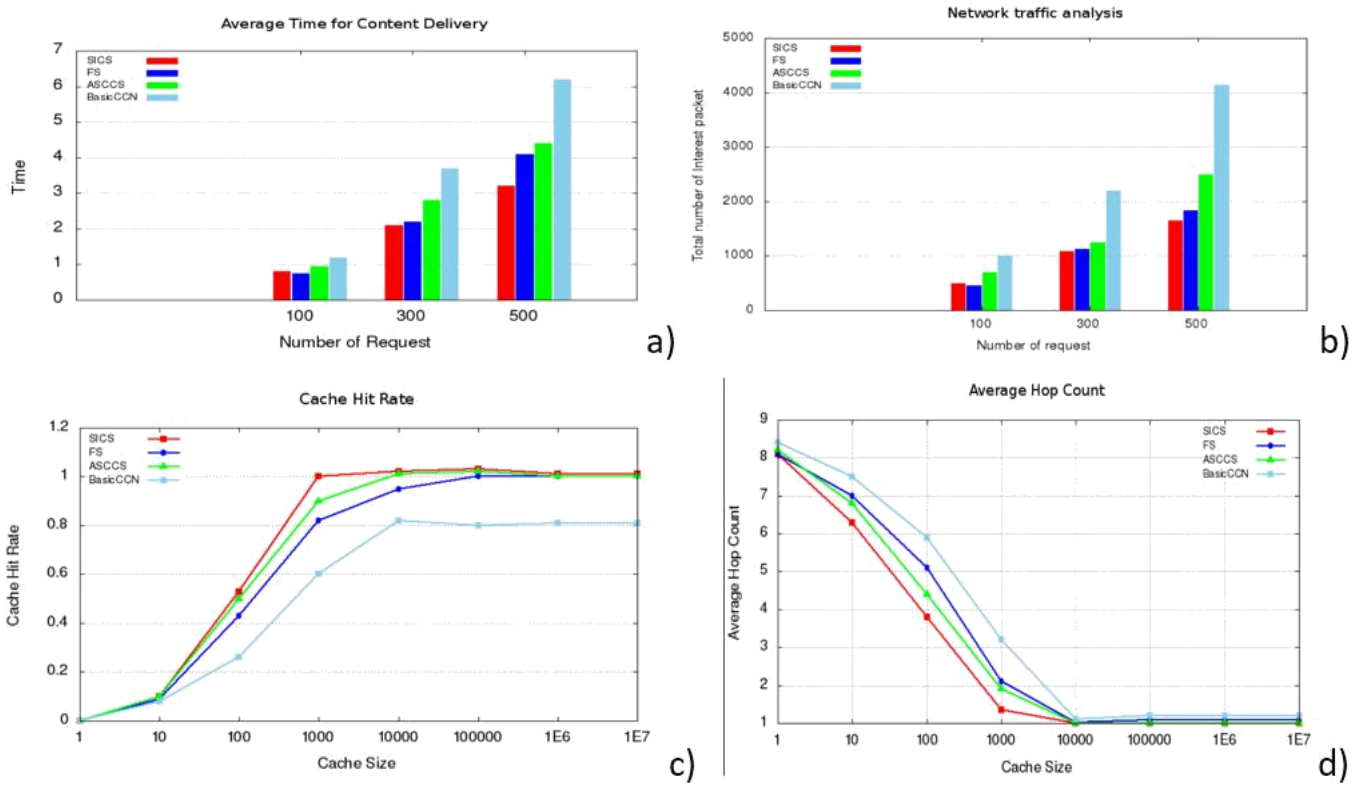

Figure 6. Simulation Results

time for content delivery, network traffic, cache hit rate and average hop count were reasonably decreased. In the near future, these results will be brought together with those of our analytical model presenting our strategy to prove the observed performance improvements. In addition, our current work will be followed by a detailed and deeper study of the compatible protocols between SDN and CCN, guaranteeing the reliability of the proposed architecture.

TABLE II. PARAMETERS USED FOR SIMULATION

Parameters	Symbol	Value
Number of users	U	10
Number of requests	λ	100, 300, 500
File size	F	1
chunk Size	S	10 KB
Forwarding	FS	SICS, ACCCS, FS, and BasicCCN
Cache Size	CS	1, 10, 100, 1000, 10000, 1E5, 1E6, 1E7

REFERENCES

[1] Visual Networking Index: Forecast and Methodology, 2016-2021, CISCO, Tech. Rep., September 2017.

[2] Z. Liu, M. Dong, and B. Gu, "Impact of item popularity and chunk popularity in CCN caching management", In Network Operations and Management Symposium (APNOMS), 18th Asia-Pacific, IEEE, 2016. pp. 1-6, 2016.

[3] C. Yufei, Z. Min , and W. Muqing, "A centralized control caching strategy based on popularity and betweenness centrality in CCN", In Wireless Communication Systems (ISWCS), International Symposium on, IEEE, 2016, pp. 286-291, 2016.

[4] G. Xylomenos, C. Ververidis, and V. Siris, "A survey of information-centric networking research", IEEE Communications Surveys & Tutorials, vol. 16, no 2, pp. 1024-1049, 2014.

[5] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey", IEEE communications surveys & tutorials, vol. 16, no 1, pp. 493-512, 2014.

[6] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN Control: Survey, Taxonomy and Challenges", IEEE Communications Surveys & Tutorials, vol. 20, no 1, pp. 333-354, 2017.

[7] R. Jmal and L.C. Fourati, "Content-Centric Networking Management Based on Software Defined Networks: Survey", IEEE Transactions on Network and Service Management, vol. 14, no 4, pp. 1128-1142, 2017.

[8] A. Fekih, S. Gaied Sonia, and H. Yousef, "A comparative study of content-centric and software defined networks in smart cities", In Smart, Monitored and Controlled Cities (SM2C), International Conference on. IEEE, 2017, pp. 147-151, 2017.

[9] Y. Zhang, H. Qu, and J. Zhao, "A distributed caching based on neighbor cooperation in ccn", In Wireless Communications, Networking and Mobile Computing (WiCOM), 10th International Conference on, Sept 2014, pp. 386–392, 2014.

[10] W. Wong, L. Wang, and J. Kangasharju,"Neighborhood search and admission control in cooperative caching networks", In Global Communications Conference (GLOBECOM), 2012 IEEE, Dec 2012, pp. 2852– 2858, 2012.

 ISBN: 978-1-61208-669-9

[11] K. Cho, M. Lee, and K. park, “Wave: Popularity-based and collaborative in-network caching for content-oriented networks”, In Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, pp. 316-3212012.

[12] J. Son, D. Kim, and H.S. Kang, “Forwarding strategy on SDN-based content centric network for efficient content delivery”, In Information Networking (ICOIN), 2016 International Conference on, IEEE, pp. 220-225, 2016.

[13] W. Kaili, W. Muqing, and Z. Min, “An autonomous system collaboration caching strategy based on content popularity in CCN”, In Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on, IEEE, pp. 1-6, 2016.

[14] K. Thakker, C.H. Lung, and P. Morde, “Secure and Optimal Content-centric Networking Caching Design”, In Trustworthy Systems and Their Applications (TSA), 2015 Second International Conference on, IEEE, 2015. pp. 36-43.

[15] C. Li, S. Gong, and X. Wang, “Secure and Efficient Content Distribution in Crowdsourced Vehicular Content-Centric Networking”, IEEE Access, 2018, vol. 6, pp. 5727-5739.

[16] J. Lv, X. Wang, and M. Huang, “RISC: ICN routing mechanism incorporating SDN and community division”, Computer Networks, 2017, vol. 123, pp. 88-103.

[17] D. Goergen, T. Cholez, and J. François, “Security monitoring for content-centric networking”, In Data privacy management and autonomous spontaneous security, Springer, Berlin, Heidelberg, 2013. pp. 274-286.

[18] “Software-defined networking: The new norm for networks”, Palo Alto, CA, USA, White Paper, Apr. 2012.

[19] O. Blial, M. Ben Mamoun, and R. Benaini, “An overview on SDN architectures with multiple controllers”, Journal of Computer Networks and Communications, 2016, vol. 2016.

[20] B.A. Nunes, M. Mendonca, and X.N. Nguyen, “A survey of software-defined networking: Past, present, and future of programmable networks”, IEEE Communications Surveys & Tutorials, 2014, vol. 16, no 3, pp. 1617-1634.

[21] https://www.nsnam.org/ [retrieved: November 2017].

 ISBN: 978-1-61208-669-9