Understanding Name-based Forwarding Rules in Software-Defined Named Data Networking

Marica Amadeo*, Claudia Campolo*, Giuseppe Ruggeri*, Antonella Molinaro*, Antonio Iera†
*University "Mediterranea" of Reggio Calabria, Italy, Email: name.surname@unirc.it
†Università della Calabria, Italy, Email: antonio.iera@dimes.unical.it

***Abstract*—Software Defined Networking (SDN) and Named Data Networking (NDN) have been recently advocated as complementary paradigms to improve content distribution in the next-generation Internet. On the one hand, SDN offers a centralized control plane that can optimize routing decisions; on the other, the distinctive features at the NDN data plane, such as name-based delivery, in-network caching, and stateful forwarding, simplify data dissemination. In the integrated design, when a request cannot be handled locally at the NDN data plane in the Forwarding Information Base (FIB), the SDN Controller is contacted to inject the forwarding rule. Decisions such as *which rules* need to be stored in the node and *for how long* deeply affect the packet forwarding performance. This paper debates about the issues related to forwarding rules in the FIBs of SDN-controlled NDN nodes, by specifically accounting for their name-based nature, representing a key novelty compared to legacy SDN implementations. Quantitative results are reported to showcase the impact of crucial parameters, like the content popularity, the content requests rate, the table size, on the FIB performance in terms of valuable metrics (e.g., hit ratio, rejected requests, incurred signaling with the Controller).**

***Index Terms*—Named Data Networking, Software Defined Networking, Information Centric Networking, Future Internet**

I. Introduction

Named Data Networking (NDN) [1] is an Information Centric Networking (ICN) architecture that shifts the Internet model from host-centric to information-centric, with application-level content names directly used at the network layer for data search and retrieval. NDN relies on the exchange of two packet types: the *Interest*, used by the consumer to request the content, and the *Data*, leveraged by the provider to answer the request. Compared to the IP design, the NDN data plane is stateful, adaptive, and implements in-network caching. When an Interest packet is received by an NDN node, it first searches for a cached Data result in the local Content Store (CS). In case of failure, it looks for a name matching in the Pending Interest Table (PIT), which records transmitted Interests that are not satisfied by the Data yet. If a matching is found, the request is discarded since an equal Interest is already pending. If both CS and PIT checks fail, the node can further forward the packet, according to the rules in the Forwarding Information Base (FIB), populated either by routing protocols, or by network administrators.

This work has been partially supported by the COGITO (A COGnItive dynamic sysTem to allOw buildings to learn and adapt) Project, funded by the Italian Government (PON ARS01_00836).

Recently, Software Defined Networking (SDN) has been considered as a potential solution to be integrated with NDN [2], in order to improve control plane features and enforce traffic engineering policies. SDN decouples the control plane from the data plane: as a result, the routing decision is implemented by a logically centralized entity, the Controller. In the integrated SDN-NDN design [3], network nodes are provided with the NDN data plane tables, with FIBs populated by the SDN Controller. When an Interest packet cannot be handled locally at the NDN data plane, the SDN Controller is contacted which enforces the routing decision and injects name-based forwarding rules in the FIB accordingly.

In a network with a large number of content requests, FIBs can be easily overflowed due to the limited table size, with inherent scalability issues. In fact, the FIB size must be kept small for fast look-up and good forwarding performance. A FIB overflow can cause blocking of new requests or eviction of existing entries to accommodate newly injected rules by the Controller. Hence, the decisions about *which entries* need to be stored in the FIB and *for how long* these shall be maintained highly affect the packet forwarding performance.

Multiple works in the SDN literature [4]–[6] investigated the management of *per-flow* rules in traditional SDN switches, but a departure from them is required in an integrated SDN-NDN architecture due to the presence of *name-based forwarding rules*. Indeed, NDN nodes can recognize (and aggregate) different requests for the same content or directly provide data from their CS, all without interacting with the Controller. Therefore, content-related attributes, such as the request frequency and the popularity, can influence the FIB occupancy status and its management, including the number of interactions with the Controller. So far, related work in the field of the integration of SDN and NDN has mainly focused on the architectural design and evaluation of content delivery performance [3], [7] without considering the impact of name-based forwarding on the FIB performance.

In this paper, we aim to fill this gap, by providing the following main contributions:

- We introduce a reference SDN-NDN integrated framework, by focusing on the modifications affecting the FIB design and resulting in the SDN-aided named forwarding fabric.
- We conduct a thorough analysis of the FIB management issues, and of the main parameters affecting them, with reference to the integrated framework.

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

• We provide quantitative results to showcase the performance of name-based forwarding rules, when considering a realistic network topology, under different simulation settings (i.e., content requests arrival rate, FIB size, content popularity).

The rest of this paper is organized as follows. The related literature is scanned in Section II, where a reference integration framework is also presented. The problem statement is described in Section III, along with the analysis of parameters affecting FIB management and performance. Results are reported in Section IV, before concluding in Section V.

II. SDN AND NDN INTEGRATION

A. A literature overview

Recently, the research community has started to recognize SDN and NDN as complementary technologies, which can benefit from each other [2]. SDN is characterized by the separation of control and data planes: routing decisions are taken in a centralized fashion by the SDN Controller, while the data plane consists of programmable network devices only in charge of data forwarding. Integrating the stateful NDN data plane in the SDN model can largely improve content delivery performance. Indeed, NDN communication directly based on content names well matches the current Internet usage, where clients are interested in the information itself, not in the location (or IP address) of the data producer. Moreover, by natively implementing in-network caching, NDN speeds up the content retrieval and reduces the traffic congestion. On the other hand, thanks to its programmable control, SDN can facilitate the commercial deployment of NDN, improve network management and make more judicious routing decisions.

Related literature on SDN-NDN has mainly focused either on implementation aspects [3], [8] and their impact on data delivery [7], or on orchestration of edge services [9]. The work in [3] presents the implementation of an SDN switch augmented with the NDN stateful data plane and focuses on caching procedures. In [7] an SDN-based routing solution for NDN is proposed, where novel control messages are defined to allow NDN nodes to communicate with the Controller. Periodically, the nodes report the status of their CS: when a novel content request arrives, the Controller is in charge of locating the closest content copy and setting up the path towards it. Evaluation accounts for the number of packets exchanged in the network and the time for taking the routing decision, however there is no clue about the load on the nodes in terms of amount of occupied FIB space. The work in [10] proposes an SDN-driven multipath forwarding strategy for NDN that efficiently distributes Interests by using the global network knowledge at the Controller. Results show that, thanks to SDN, data forwarding is not restricted only to the on-path routers: this reduces the latency for data retrieval and avoids single-path congestion. As for the previous case, however, FIB management issues are not considered. In this paper, we advance the state-of-the-art by investigating the impact of an SDN-driven control plane on the FIB management. We analyse multiple factors influencing it, ranging from request arrival rate to content popularity, from network topology to FIB size.

B. A reference framework

In this section, we shortly describe the reference SDN-NDN framework for our analysis, building upon the existing literature, with focus on routing and forwarding operations.

Basics. The framework consists of a centralized SDN control plane and a NDN-based data plane, with OpenFlow (OF) [11] chosen as Southbound Interface, Fig. 1, as common in the literature [3], [12]. Network nodes implement the NDN Data plane tables, i.e., CS, PIT and a modified FIB. The latter one is practically deployed as an SDN flow table augmented with the notion of name-based forwarding. For routing purposes, the Controller maintains two main data structures: the *Network Information Base* (NIB), which tracks the graph of the network topology as in legacy SDN, and the *Content Information Base* (CIB), which tracks the contents stored in the domain.

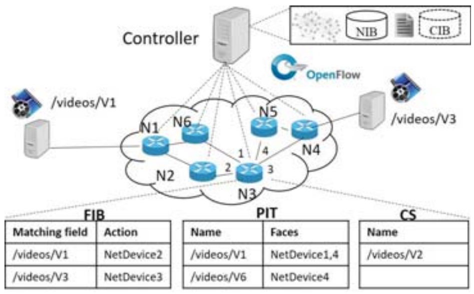


Fig. 1. Reference SDN-NDN architecture.

FIB design. In the legacy NDN implementation, the FIB is a collection of entries containing a name prefix and a non-empty collection of NextHop records, which include the outgoing interfaces toward a potential content source and the associated routing cost. FIB entries are filled by the Routing Information Base (RIB), which in turn may receive *static routes* configured manually, and *dynamic routes* determined from routing protocols. In the integrated SDN-NDN framework, instead, routing decisions are centrally managed by the SDN Controller, which uses OF to inject forwarding rules in the FIB of the nodes it oversees. In the FIB entry, the flow identifiers, consisting of some IP header fields in the traditional SDN deployment, are replaced by the content name[1]. A set of actions can be specified by the Controller for each FIB entry, including: *(i)* forwarding the Interest to a given output port (e.g., *NetDeviceFace 1*); *(ii)* contacting the Controller, whenever there is no entry that matches the content name of the incoming packet.

Forwarding fabric. Requests for named contents are carried in legacy Interest packets issued by consumers and processed according to the NDN forwarding fabric. Only in case of failure of the local CS/PIT/FIB matching, the SDN control plane is called upon. In such a case, the NDN node encapsulates the packet header into a PACKET_IN, the legacy OF message used to request a flow rule from the

[1]Matching-by-name can be enabled via the Experimenter extension, and, in particular, through the OF eXtensible Match (OXM) [11], similarly to [8].

Controller [11]. Upon receiving it, the Controller looks in the CIB to check the availability of the content, identifies the forwarding path towards it by accessing the NIB, and injects forwarding rules in the FIB of on-path nodes through the OF FLOW_MOD message [11]. The Controller sets up only the one-way routing paths towards the content; indeed, once Interests are transmitted according to the action in the FIB entry, they remain recorded in the PIT to allow the forwarding of the content back and also the aggregation of incoming requests for the same content from different consumers.

III. Management of Forwarding Rules

A. *Flow table management in legacy SDN*

In SDN networks, traffic is forwarded according to rules stored in the flow table of network nodes. A flow table cannot store an unlimited number of flow entries. The typical size of an SDN flow table is in the order of 1k-10k entries, due to hardware limitations related to the usage of Ternary Content Addressable Memory (TCAM), quite expensive in terms of cost and energy [5]. Therefore, in a network with a large number of flows, flow tables at switches can be easily overflowed. When a flow table-miss occurs, the corresponding switch must interact with the Controller to insert new flow entries, but this results in additional processing time and communication overhead, increasing the latency.

According to OF specifications [11], two timeouts are associated to each entry in order to maintain flow-table efficiency, i.e., the *hard-timeout* and the *idle-timeout*. An entry gets deleted after the hard timeout expires *even if there are still packets* arriving for this entry. An entry is deleted *if no packets arrive* for an entry within the idle-timeout period.

The timeout values should be set by the Controller to reflect the available usage time of the forwarding action and not to keep stale information. Notwithstanding, the usage of a *fixed timeout*, after which the switch automatically removes the rules from its flow table, is practically considered, which has shown several drawbacks. In some cases, a timeout longer than the shortest flows' lifetime can lead to inefficient flow table utilization: entries for inactive flows remain stored in the flow tables until their timeouts expire and waste flow table space. Conversely, active flows may be removed due to flow table overflow. This, as well, is especially detrimental in terms of additional load to the Controller. Several solutions have been proposed in the literature, which *dynamically* set the timeout to different flows according to their characteristics [5], [6].

When a flow table is full, OF v1.3 enables a naïve admission control: a network node rejects newly inserted flow entries and sends error messages to the Controller. In addition to OF eviction mechanisms, a further workaround to keep useful flow entries in the node as long as possible is applying a Least Recently Used (LRU) algorithm to evict the least recently used entries, as suggested in [4], [13].

B. *FIB management in an integrated SDN-NDN framework*

The management of FIBs in an integrated SDN-NDN framework has not been fully addressed in the literature. As the FIB is deployed as an SDN flow table, its management gets particularly critical, being exposed to the same limitations. Although FIB management in SDN-controlled NDN nodes can build upon the routines described above for legacy SDN implementations, some modifications are entailed to specifically account for name-based forwarding.

Unlike per-flow rules (typically based on the source/destination IP addresses and transport ports) in the legacy SDN implementations, per-name rules are enabled in the integrated SDN-NDN framework. This feature allows aggregation of requests for the same named content. In other words, the same FIB entry could match different requests for the same named content issued by different consumers. Such a capability, which could save the number of stored entries, cannot prevent the FIB to get saturated in some cases [14].

In our study, we assume that an LRU policy is applied whenever the FIB is full, and we use the idle timeout mechanism for identifying inactive rules. If the FIB is full and there are no inactive entries to be deleted, the new content request is rejected to prevent ongoing content delivery to be harmed. Fig. 2 summarizes the Interest processing in the FIB of a network node, in case of a local failed CS and PIT matching.

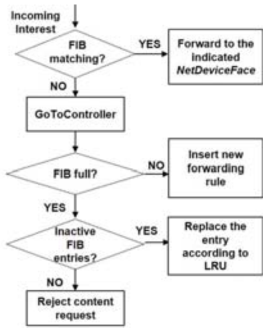


Fig. 2. Interest processing in the FIB.

Different factors affect the FIB management and its occupancy. Some of them play the same role also in the legacy SDN implementation and are quite intuitive, whereas others are specific for the integrated framework. The ones we deemed more relevant are listed below:

- *Interest rate.* Intuitively, the higher the rate of content requests the larger (potentially) the number of FIB entries to be maintained.
- *Content size.* It affects the FIB size and its dynamics. Larger contents entail longer retrieval time and, consequently, entries to be stored for a longer time.
- *Content popularity.* Not all contents in the catalog are equally requested, and some of them may not be requested at all. If more requests tend to concentrate on a few contents (i.e., the popular ones), then the number of FIB entries required to be injected by the Controller would be reduced.
- *Content-based betweenness centrality.* Not all nodes have the same FIB behaviour: this may depend on the node *centrality*. A widely used topological centrality metric is the betweenness, which counts the fraction of shortest paths going through a

given node. A node hardly belongs to the paths set by the Controller to retrieve *all the contents*; this depends on its topological position in the domain. In an NDN network, the node centrality should be evaluated not only from a topological perspective, but also with respect to the content popularity. A node n could be traversed by multiple paths towards several data producers and, therefore, have a high betweenness centrality. However, if such producers do not generate popular traffic, n has not a central role in the content delivery. We, therefore, consider here the notion of content-based centrality (CBC), originally conceived in [15] and defined as:

$$CBC = \sum_{u,x} \frac{\sigma_n(x)}{\sigma(x)} p_x, \forall x \in X, \quad (1)$$

where $\sigma_n(x)$ is the number of shortest paths passing from node n to reach content x belonging to the catalog X, $\sigma(x)$ is the total number of shortest paths in the network to reach content x, p_x is the popularity of content x. The larger the CBC of a node the larger its FIB occupancy status.

• *FIB size.* The larger the FIB size the higher the number of requests that can be locally satisfied. However, we remark that the FIB size cannot be overly large due to hardware limitations and to ensure efficient lookup performance.

IV. Performance Evaluation

The conducted study aims at capturing the impact of (most of) the aforementioned parameters on the FIB performance in terms of locally satisfied and rejected content requests and signaling exchange with the Controller. The influence of each of the considered parameters on the performance is not valid *per se*, but their effect should be jointly evaluated. Evaluation is performed with a co-simulation framework integrating Matlab® and ndnSIM, implementing the SDN control plane and the NDN data plane, respectively. Results are averaged over 100 runs and reported with 95% confidence intervals.

A. Evaluation settings and metrics

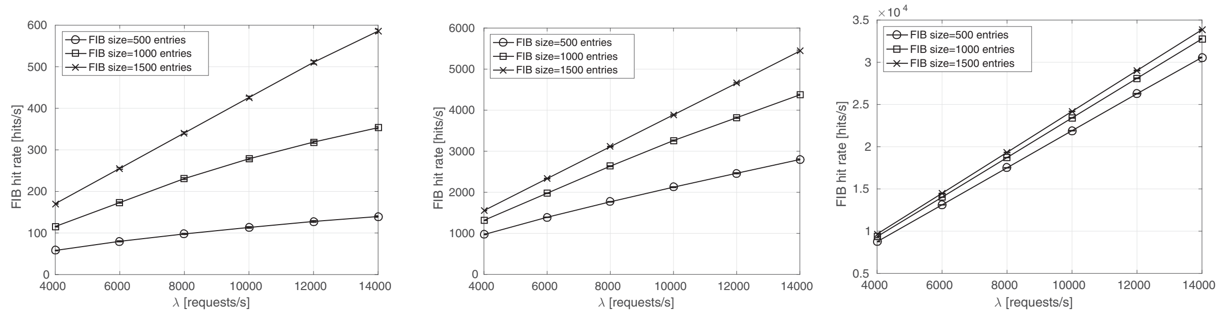
We consider the Abilene network topology [16], with 11 nodes, which consumers and providers are attached to. Consumers can request contents belonging to a catalog of size $|X|$ set to 10^6. We model both *mice* and *elephant* contents. More in detail, similarly to [4], each elephant content size is a random value between 10 MB and 20 MB; each mice content's size is a random value between 100 KB and 200 KB; the average ratio of mice contents to elephant contents is 9:1. Granted such settings, we evaluate the performance when varying the following parameters. We model the content popularity through the Zipf distribution with skewness parameter α, varying in the range $0.4 - 1.2$. We assume that the Interest arrivals at the simulated nodes follow the Poisson distribution [17]. The Poisson parameter λ is varied from 4000 to 14000 Interests/s (in the whole topology) to resemble very heavy load conditions. These settings are intended to saturate the FIBs and remove artifacts due to cold start effects in presence of a few requests. The FIB size is varied, from 500 to 1500 entries.

The following metrics have been considered. The *FIB hit rate* is derived as the success rate of finding a matching entry in the FIB for an incoming request at each node along the consumer-provider path. The *rejected requests* metric accounts for the percentage of incoming requests which cannot be satisfied due to FIB overflow. *OF signaling* is derived as the rate of OF packets exchanged between the nodes and the Controller for each content request. It captures the overhead incurred whenever a content request cannot be satisfied by entries stored in the FIB. In particular, it accounts for the OF PACKET_IN and FLOW_MOD messages exchanged between the network nodes, along the path towards the requested content, and the Controller. The *Active entries* metric measures the percentage of FIB entries for which the timeout has not yet expired.

B. Results

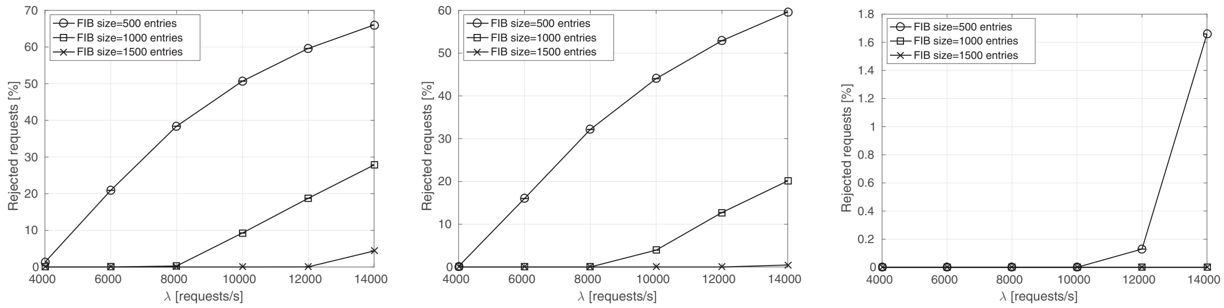
Fig. 3 reports the FIB hit rate under different request arrival rates (λ) and Zipf parameter (α) settings for different FIB sizes. Intuitively, the metric increases with λ because of the higher number of requests to be served. Moreover, it reasonably decreases for small FIB size values: there is a low chance for an incoming request to find a matching entry in the FIB. The impact of the FIB size is more remarkable for low α values. In the latter cases, indeed, several entries need to be stored for several distinct content requests, which cannot be removed if active (Fig. 6). The higher the parameter α the larger the metric. In such a case, requests concentrate on a few popular contents for which a lower number of entries need to be stored in the FIB, hence leading to a higher FIB hit rate. Such a trend is confirmed by results in Figs. 4, 5, and 6, showing the rejected requests, the OF signalling rate, and the active entries, respectively. For α equal to 1.2, findings are as follows: *(i)* the majority of content requests are locally satisfied with a FIB matching (Fig. 3(c)); *(ii)* only a few nodes need to contact the Controller to get instructed with forwarding rules (Fig. 5(c)); *(iii)* these rules can be injected in the FIB, indeed there is enough space (under most settings active entries account for less than half of the FIB size, Fig. 6(c)); *(iv)* the percentage of rejected requests is negligible (Fig. 4(c)). For FIB size set to 500 entries and α to 0.4, 14000 requests/s, more than 60% of content request are rejected, Fig. 4(a).

In Fig. 5, the signalling reasonably increases with λ, since more requests need to be served and with the help of the Controller. The metric achieves lower values when the FIB size is small, because content requests are rejected, hence the Controller has no rules to inject. This holds for α equal to 0.4 and 0.8. Conversely, when considering α equal to 1.2, an opposite trend can be observed: the larger the FIB size the lower the signaling. Such a behaviour has to be ascribed to the fact that it is more likely for a content request to match a stored entry. The impact of the FIB size is less remarkable as λ increases, compared to the cases with lower α values. To better understand the incurred signaling footprint, let us refer to the number of OF signaling packets exchanged per node and per request which is reported in Table I. In the worst case (FIB size=1500 entries, α=0.4), almost each node reached by



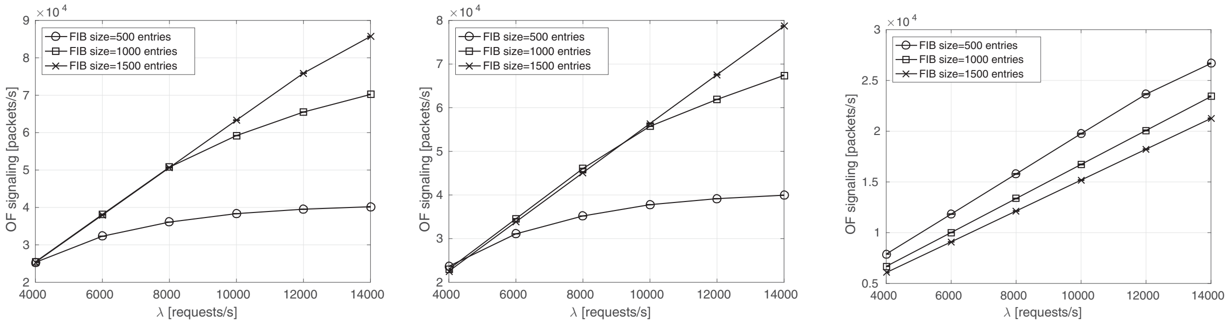
(a) $\alpha = 0.4$. (b) $\alpha = 0.8$. (c) $\alpha = 1.2$.

Fig. 3. FIB hit rate when varying the request arrival rate (λ), for different Zipf parameter (α) and FIB size settings.



(a) $\alpha = 0.4$. (b) $\alpha = 0.8$. (c) $\alpha = 1.2$.

Fig. 4. Rejected requests when varying the request arrival rate (λ), for different Zipf parameter (α) and FIB size settings.



(a) $\alpha = 0.4$. (b) $\alpha = 0.8$. (c) $\alpha = 1.2$.

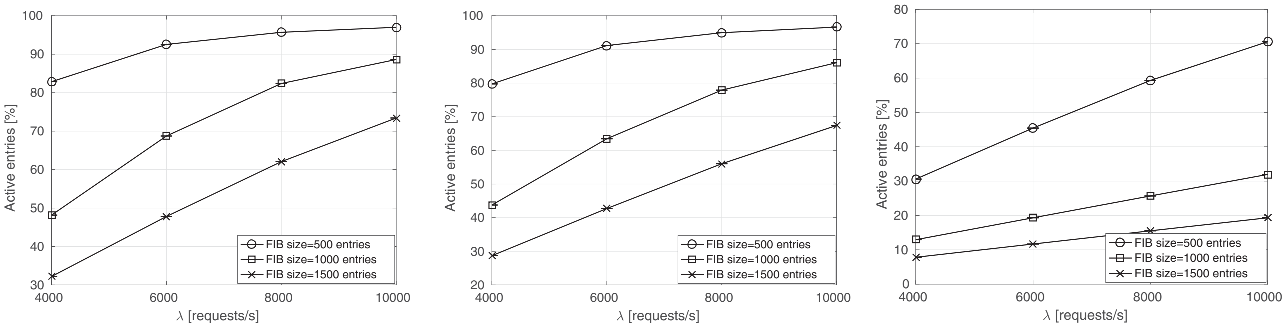
Fig. 5. OF signaling when varying the request arrival rate (λ), for different Zipf parameter (α) and FIB size settings.

a content request interacts with the Controller to get instructed about the forwarding action (2 OF packets are exchanged on average), the FIB hit ratio is very low correspondingly (slightly higher than 0.01). The metric decreases to nearly 0.5 for α=1.2 and, correspondingly, the FIB hit ratio gets over 0.76. The aforementioned metrics have been also measured at the most central node derived both according to the in-betweenness centrality (we considered the Kansas City node in the Abilene topology) and according to the CBC in Eq. (1). Results reported in Table II clearly show that the FIB performance for the central nodes highly differs from the average value measured for the topology[2]. Central nodes exhibit a higher FIB hit ratio, more heavily interact with the Controller and they have more active entries. Moreover, the node with the highest CBC is more loaded than the Kansas City one, confirming the intuition discussed in Section III.

[2]Please notice that average values are normalized over the number of nodes.

V. Conclusion

In this paper we have analyzed the FIB design and management in an integrated SDN-NDN framework. We have scrutinized how the FIB design is re-engineered in the integrated framework, and we have analyzed the FIB management procedures required to counteract the table overflow and to ensure the effective handling of incoming content requests. The conducted study unveils the decisive impact of the content popularity on the FIB behaviour, as well as of



(a) $\alpha = 0.4$. (b) $\alpha = 0.8$. (c) $\alpha = 1.2$.

Fig. 6. Active entries in the FIB when varying the request arrival rate (λ), for different Zipf parameter (α) and FIB size settings.

TABLE I
METRICS WHEN VARYING THE FIB SIZE, FOR DIFFERENT ZIPF PARAMETER (α) SETTINGS, λ=10000 REQUESTS/S.

α	**OF signaling/per node/per request**			**FIB hit ratio/per node/per request**		
	FIB size=500 entries	FIB size=1k entries	FIB size=1.5k entries	FIB size=500 entries	FIB size=1k entries	FIB size=1.5k entries
0.4	1.363	1.784	1.945	0.0048	0.009	0.0132
0.8	1.283	1.648	1.754	0.0899	0.1068	0.1211
1.2	0.604	0.526	0.477	0.6045	0.7368	0.761

TABLE II
METRICS FOR DIFFERENT ZIPF PARAMETER (α) SETTINGS, λ=10000 REQUESTS/S, FIB SIZE=1000 ENTRIES, DERIVED AS AVERAGED FOR THE NODES OF THE TOPOLOGY, FOR THE NODE WITH THE HIGHEST IN-BETWEENNESS, AND FOR THE NODE PROVIDING THE HIGHEST CBC.

α	**OF signaling [packets/s]/per node**			**FIB hit rate [hits/s]/per node**			**Active entries [%]**		
	Average	In-betweenness	CBC	Average	In-betweenness	CBC	Average	In-betweenness	CBC
0.4	5387.45	7629.5	7718.4	25.34	34.299	37.272	88.644	98.421	98.635
0.8	5072.09	7436.7	7653.1	293.63	327.7	463.27	86.09	98.106	98.504
1.2	1519.36	2270.3	2483.7	2127.18	2363.1	3881.2	31.928	42.692	46.678

other workload and FIB parameters. In the presence of highly popular contents, the peculiar name-based forwarding of NDN allows to largely improve the FIB management and the packet forwarding: almost all content requests are locally served, without harming active content retrievals, and independently of the FIB size. For lower content popularity settings, the number of rejected content requests is still practically acceptable for common FIB size values (1k entries), even when the number of content requests is very high. This finding, which derives from the name-based forwarding rules, suggests a better FIB scalability, compared to legacy SDN implementations. As a further study, we plan to improve the FIB management by letting routing decisions at the Controller be taken also according to the FIB occupancy status, similarly to the study in [4], by inspecting different routing paths.

REFERENCES

[1] L. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.
[2] Q.-Y. Zhang *et al.*, "Software Defined Networking Meets Information Centric Networking: a Survey," *IEEE Access*, vol. 6, 2018.
[3] A. Mahmood *et al.*, "Efficient Caching through Stateful SDN in Named Data Networking," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 1, p. e3271, 2018.
[4] Z. Guo *et al.*, "STAR: Preventing Flow-table Overflow in Software-defined Networks," *Computer Networks*, vol. 125, pp. 15–25, 2017.
[5] H. Zhu, H. Fan, X. Luo, and Y. Jin, "Intelligent Timeout Master: Dynamic Timeout for SDN-based Data Centers," in *IFIP/IEEE IM*, 2015.
[6] K. Kannan and S. Banerjee, "FlowMaster: Early Eviction of Dead Flow on SDN Switches," in *Int. Conf. on Distributed Computing and Networking*. Springer, 2014, pp. 484–498.
[7] E. Aubry *et al.*, "Implementation and Evaluation of a Controller-based Forwarding Scheme for NDN," in *IEEE AINA*, 2017.
[8] P. Zuraniewski *et al.*, "Facilitating ICN Deployment with an Extended OpenFlow Protocol," in *ACM ICN Conference*, 2017, pp. 123–133.
[9] M. Amadeo, C. Campolo, G. Ruggeri, A. Molinaro, and A. Iera, "SDN-Managed Provisioning of Named Computing Services in Edge Infrastructures," *IEEE TNSM*, vol. 16, no. 4, pp. 1464–1478, 2019.
[10] M. Alhowaidi, D. Nadig, B. Ramamurthy, B. Bockelman, and D. Swanson, "Multipath Forwarding Strategies and SDN Control for Named Data Networking," in *IEEE ANTS*, 2018, pp. 1–6.
[11] "Openflow Switch Specification - version 1.3.1 Open Networking Foundation (ONF)," September 2012.
[12] R. Jmal and L. C. Fourati, "An OpenFlow Architecture for Managing Content-Centric-Network (OFAM-CCN) based on Popularity Caching Strategy," *Computer Standards & Interfaces*, vol. 51, pp. 22–29, 2017.
[13] E.-D. Kim, Y. Choi, S.-I. Lee, and H. J. Kim, "Enhanced Flow Table Management Scheme with an LRU-based Caching Algorithm for SDN," *IEEE Access*, vol. 5, pp. 25 555–25 564, 2017.
[14] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable Name-based Packet Forwarding: from Millions to Billions," in *ACM ICN 2015*.
[15] J. A. Khan, C. Westphal, and Y. Ghamri-Doudane, "A Content-based Centrality Metric for Collaborative Caching in Information-Centric Fogs," in *IFIP Networking*, 2017, pp. 1–6.
[16] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE JSAC*, vol. 29, no. 9, pp. 1765–1775, 2011.
[17] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-scale Video-on-Demand Systems," in *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4, 2006, pp. 333–344.