Efficient caching in Content-Centric Networks using OpenFlow

Xuan Nam Nguyen, Damien Saucez, Thierry Turletti
INRIA Sophia Antipolis Méditerranée
Sophia Antipolis, France
{xuan-nam.nguyen, damien.saucez, thierry.turletti}@inria.fr

***Abstract*—Content-Centric Networking (CCN) is designed for efficient content dissemination and supports caching contents on the path from content providers to content consumers to improve user experience and reduce costs. However, this strategy is not optimal inside a domain. In this paper, we propose a solution to improve caching in CCN using a Software-Defined Networking approach.**

Keywords—Content-Centric Networking, Software-Defined Networking, OpenFlow

I. INTRODUCTION

Content distribution and retrieval are dominant Internet applications today. However, the Internet was designed for communication between exactly two machines and this communication model reduces the efficiency of content delivery services. To fit today's trend, *Content-Centric Networking* (CCN [1]) has been proposed. In CCN, contents are retrieved directly by their names, instead of locations. In addition, contents may be cached at intermediate nodes on the path from content providers to consumers. This "on-path caching" strategy makes contents closer to consumers, and hence reduces bandwidth consumption and content retrieval time. However, it has been demonstrated that this strategy is not optimal as it may imply a high content replication that limits the maximum number of contents that can be cached inside a domain [2], [3]. Instead, "off-path caching" is an alternative strategy that can avoid duplications and can significantly increase the overall hit rate [3]. This strategy consists of three policies: (1) caching only most popular contents, (2) deciding where to cache those contents by solving an optimization problem, and (3) deflecting requests to these optimally selected caches.

We argue that *Software-Defined Networking* (SDN) [4] is the perfect candidate to implement off-path caching. In SDN, the control plane is completely decoupled from the data plane, which allows to use efficiently routes computed using high level information such as traffic statistics. Among SDN implementations, OpenFlow (OF) [5] gets all our attention in this paper as it relies on centralized controllers and offers per-flow forwarding decisions. The advantages of a centralized controller are twofold in our case. On the one hand, the centralized controller has a global view of network state, therefore it can detect the most popular contents at the global level. On the other hand, the controller can perform intensive computations (e.g., solving an optimization problem) and populate quickly new routing tables to nodes (e.g., to deflect requests to appropriate caches).

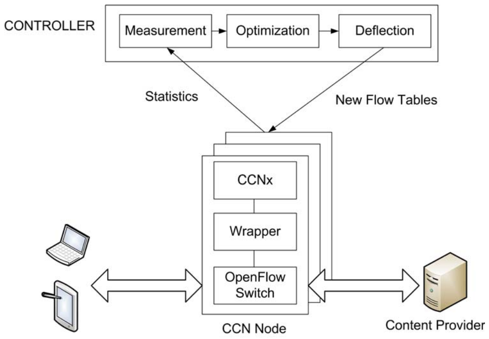

Fig. 1: Architecture supporting off-path caching in CCN

There are three challenges to solve when using OF to implement off-path caching. First, how to provide the CCN functionality using current OF switches that support neither name-forwarding nor content caching. Second, how to measure the popularity of contents accurately and without a large overhead. Third, how to build and optimize routing tables to perform deflection.

The rest of this paper is organized as follows. Section II describes our OpenFlow-based architecture that aim to solve the above challenges to enable off-path caching in CCN. Section III gives a preliminary evaluation of our solution. Finally, section IV concludes this work.

II. ARCHITECTURE

Current OF switches do not support deep packet inspection [6], and in particular they are not able to forward CCN packets using names. To provide the CCN functionality over OF networks, Melazzi et al. [7] propose to add a new IP option field and some modifications in the OF protocol to handle this field. We prefer to follow a different approach which requires modifications in neither current OF nor IP and that can use CCNx [8], the CCN reference implementation. To be compatible with the existing architecture, CCNx builds an IP overlay to transport Interest and Data packets.

To link a CCNx deamon with an OF switch, we propose to use a wrapper (see Fig. 1). The wrapper pairs a switch interface to a CCNx interface, decodes and hashes content names in CCN messages into fields that an OF switch can process (e.g., IP addresses, port numbers). The large naming space offered by these fields limits the probability of having collisions between two different content names. Thanks to the wrapper,

978-1-4673-1017-8/13/$31.00 ©2013 IEEE

an OF switch can forward and monitor Interest packets using content names. The number of contents that an OF switch can process at one given time without the controller's support is bounded by the maximum number of flow entries it can support. The advantage of this approach is that we benefit from hardware capacity in forwarding and monitoring. In addition, the wrapper can be easily adapted to new versions of the CCNx and the OF protocol. The main drawbacks are that the fields used for hashing lose their original meaning and that the forwarding performance is reduced.

To solve the two remaining challenges, we propose three new controller modules: **Measurement**, **Optimization**, and **Deflection** (see Fig. 1). OF switches always maintain a set of counters for active flows and allow to retrieve network statistics easily. Content popularity can thus be inferred directly from flow statistics using the **Measurement** module on the controller side. The **Measurement** module periodically queries and processes statistics from ingress OF switches to return the list of most popular contents. It is worth noting that the time between two queries should be chosen wisely to avoid large overhead while maintaining an updated view of the network.[1]

After that, the **Optimization** module uses the list of most popular contents as an input for the optimization problem mentioned in [3]. The objective function of this optimization problem consists in minimizing the sum of the delays over deflected contents under the following constraints: (1) each popular content is cached at exactly one node, (2) caching contents at a node does not exceed node's capacity, and (3) caching should not cause link congestion. This optimization problem can be solved by Linear Programming or approximated with heuristics [3]. Finally, the **Optimization** module provides the optimal set of *(content name, node caching this content)* pairs for the current traffic and topology.

The **Deflection** module uses the above set to build a mapping, for every content, between the content name (by the mean of addresses and ports computed from the content name hash) and an outgoing interface toward the node where the content is cached (e.g., *ip.destination = hash(content name), action = forward to interface 1*). Finally, mappings are installed on switches' flow tables using the OF protocol such that subsequent Interest packets can be forwarded to appropriate caches.

III. Evaluation

As a first evaluation of our solution, we implemented the wrapper, ran it along with a CCNx daemon on a node connected to a Pronto 3290 OpenFlow switch. Another node generates Interest packets with different packets-per-second (PPS) rates to the OpenFlow switch. We considered three scenarios when forwarding Interest packets: using the OF switch; using the OF switch and the CCNx deamon; using the OF switch, the wrapper and the CCNx deamon.

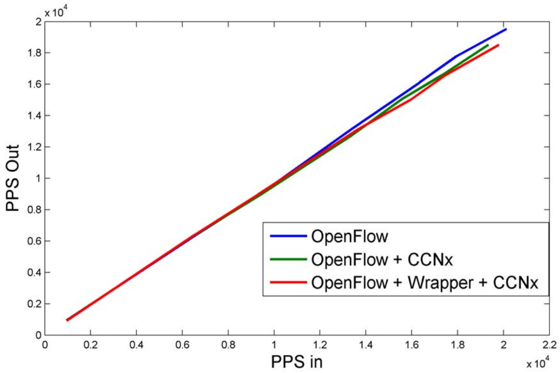

Fig. 2: Wrapper's impact on forwarding performance

Fig. 2 shows the outgoing rate (PPS out) for different incoming rates in three scenarios. At high incoming rate, using only the OF switch gives the best PPS out because CCN packets are forwarded at switch level. In contrast, using the wrapper slightly degrades forwarding performance but no more than 5%. This can be justified by the fact that the wrapper processes CCN packets coming from both the OF switch and the CCNx daemon. This preliminary evaluation shows that the wrapper does not significantly degrade forwarding performance.

IV. Conclusion

Caching strategy might dramatically influence performance and efficiency of content-centric networks. In this paper, we present an OpenFlow-based architecture that performs efficient caching for content-centric networks. Preliminary results show the feasibility and the low overhead of using a wrapper in our approach. We are currently evaluating the efficiency of the complete solution on a simulator and a testbed. As for further study, we plan to extend our solution to dynamic environments such as wireless mesh networks.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[2] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proceedings of the 11th international IFIP TC 6 conference on Networking - Volume Part I*, ser. IFIP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 27–40.

[3] D. Saucez, A. Kalla, C. Barakat, and T. Turletti, "Minimizing bandwidth on peering links with deflection in named data networking," INRIA Sophia Antipolis Méditerranée, Tech. Rep., 2012. [Online]. Available: http://hal.inria.fr/hal-00684453

[4] "Software-Defined Networking," https://www.opennetworking.org/.

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[6] "OpenFlow Switch Specification," [accessed 08-Jan-2013]. [Online]. Available: https://www.opennetworking.org/standards/openflow-switch

[7] N. Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, and L. Veltri, "An openflow-based testbed for information centric networking," in *Future Network Mobile Summit, 2012*, july 2012, pp. 1 –9.

[8] "CCNx," http://www.ccnx.org.

[1] The overhead caused by statistics retrieval could be reduced if OF switches would support a push mode for statistics.