Towards Exploring the Benefits of Scope-Flooding in Information-Centric Networks

Andriana Ioannou and Stefan Weber
School of Computer Science and Statistics
Trinity College
Dublin 2, Ireland
Email: ioannoa, sweber@scss.tcd.ie

***Abstract*—Approaches to *Information-Centric Networking (ICN)* employ caches within a network to reduce transfers of information from a source to consumers. Copies of content within a network may be exploited through the use of a forwarding mechanism. *Scope-flooding* is a forwarding strategy based on the distribution of requests through flooding. The scope of the flooding is defined through a *scope-value* which prevents network-wide flooding.**

In this paper, we investigate the impact of scope-flooding forwarding against Shortest-Path Routing (SPR) using a number of on-path caching policies. Our results show that scope-flooding does not always benefit the performance of a caching policy, as stated so far. We conclude by discussing a number of features that may be adopted to enhance the performance of scope-flooding.

***Index Terms*—Distributed Networks; Information-Centric Networking; On-Path Caching; Forwarding Strategy; Scope-Flooding**

I. INTRODUCTION

Information-Centric Networking (ICN), an alternative to the host-centric communication model of the current Internet infrastructure, decouples the bind between content and hosts using *content identifiers*. Content identifiers define content resources and should involve no topological information [7]. If this constraint is met, the content can be freely replicated.

ICN introduces the capacity to cache content within the infrastructure; routers are equipped with cache memory and enabled with a caching capability. *On-path caching* is a feature that distributes copies of content at routers along the delivery paths to reduce delivery times. On-path caching and forwarding are integrated as follows: upon the arrival of a content request, a node performs a forwarding decision based on its forwarding table and *forwarding strategy*. A content request triggers a content reply issued by either a source or any node that has a copy. Upon a content reply, a node on the delivery path decides to cache content according to a *caching policy*.

In contrast to the coupled nature of caching policies and forwarding strategies, the majority of caching policies have been investigated regardless of the forwarding strategy and vice versa [9]. According to the results of recent efforts [3], [9], considering both caching policies and forwarding strategies, Shortest-Path Routing (SPR), that has been commonly used as the default forwarding strategy, may considerably decrease the performance of a caching policy. The reason to this outcome is the opportunistic nature of SPR, according to which only copies that lay on the shortest routing paths may be exploited. To this end, *scope-flooding* is a flooding-based forwarding strategy that targets the exploitation of copies on all paths. To prevent infinite flooding of content requests in a network, a *scope-value* is set. Content requests of scope-value zero are discarded.

This paper focuses on the investigation of scope-flooding as a forwarding mechanism for the exploitation of cached copies and its effect on the performance of a caching policy. To this end, a number of on-path caching policies are evaluated via simulations. Based on the results, scope-flooding forwarding may not always conclude to a better performance of a caching policy, as stated so far [9]. Towards explaining this outcome, we discuss potential features that may improve performance.

The remainder of this paper is structured as follows. Section II presents the existing work on caching policies and flooding mechanisms. Section III describes the scope-flooding strategy. Section IV presents the simulation model. Section V discusses the evaluation results and suggestions to further improvements. Section VI is dedicated to the conslusions.

II. RELATED WORK

Depending on the caching criteria used, existing on-path caching policies can be divided into *probabilistic policies*, *fixed* or *dynamic*, *graph-based policies* and *content-based policies*.

Fixed policies, *FIX(p)*, base the probability to create a replica at a node on a priori defined value [2], [7]; thus, neglect the variable nature of content requests and network topologies. In order to address this, *ProbCache* is based on the number of replicas to be cached on a delivery path, decided based on the cache capacity of the delivery path and a factor to counterbalance the tendency to cache far from the source.

Graph-based policies such as *Betweenness Centrality (BC)* and *Degree Centrality (DC)* [10] take into account the topology of the network. BC takes into account the number of times a node is included in the shortest paths between sources and consumers while DC is based on the number of connections of a node. These approaches neglect the frequency and distribution of content requests. To this end, *Leave Copy Down (LCD)* [8] caches a copy of the requested content one hop closer to a client each time a content request arrives.

Content-based policies such as *Congestion-Aware Caching (CAC)* [3] and *Prob-PD* [11] base their caching decision on

978-1-4673-6770-7/15/$31.00 ©2015 IEEE

content characteristics, i.e. content popularity, calculated by the number of requests for a content to the total requests.

Eventhough, caching policies and forwarding strategies are strongly coupled, their performance has not been examined as a combined problem until recently [8], [9]. Based on the results, SPR that has been commonly used as the default forwarding strategy for the evaluation of caching policies may significantly decrease the performance of a caching policy. SPR lies on the category of *opportunistic forwarding* where no information with regard to content copies is maintained. A copy is exploited if it lies on the shortest path of a subsequent request. *Caching-aware forwarding* is based on the collection of information with regard to content copies. Caching-aware forwarding is more likely to benefit the performance of a caching policy as a better utilization of memory resources may be achieved.

Scope-flooding [3], [4], [9] is a caching-aware forwarding mechanism based on flooding. Forwarding is accomplished in two phases, the *discovery phase* and the *exploitation phase*. In the former, a router is using its already known forwarding path(s) and exploring alternative paths via flooding. This is because the discovery of a new forwarding path does not guarantee the discovery of a copy. In the later, a router is using the information obtained during the exploration phase. If a copy has been discovered, a decision on which path is preferable is made based on latency. To prevent the infinite propagation of content requests, a *scope-value* is defined. Content requests with scope-value zero are discarded.

Scope-flooding has been recently proposed [9] as a forwarding strategy able to increase the performance of a caching policy and exhibit performance rates close to the optimal [5]. In this paper we define the caching policies where a scope-flooding forwarding mechanism may substantially decrease performance as well as the reasons that lead to this outcome.

III. SCOPE FLOODING ALGORITHM

In this section, a description of the scope-flooding forwarding mechanism is provided based on the *Content-Centric Networking (CCN)* [7] communication model. To relate the results with those presented in [9], the same assumptions are followed where possible; as the operation of the algorithm has not been fully specified, a number of assumptions had to be made to finalize the implementation, e.g. the update of temporary routes.

Consumers issue object requests. Each object request corresponds to a sequence of *Interest packets*, i.e. content requests issued on a packet level. Prior to the propagation of the first Interest, a *search Interest*, used for the exploration of replicas in a network, is flooded. Upon the arrival of a search Interest on a node, a longest-match lookup is performed on its *Content Store (CS)* table, i.e. a table that stores objects. If an object that matches the Interest exists, a *search Data packet*, i.e. a content reply packet that contains no data, is forwarded via the face where the Interest arrived. Empty content replies ensure cache eviction avoidance while minimize the network overhead caused by flooding. If no object is found, the scope value of the search Interest is reduced by one. If the scope value is positive, the search Interest is flooded and discarded otherwise. A *Pending Interest Table (PIT)* keeps track of the incoming and outgoing faces with regard to Interest packets; a PIT entry is created upon the arrival of a search Interest.

Due to flooding, one search Interest packet may trigger multiple search Data packets. Upon the arrival of the first search Data packet, a node will search its PIT table for a matching Interest. If an Interest exists, the node will check if the obtained hop-count in the search Data packet, is less than the hop-count towards a source. If objects can be found at a source with equal or less cost compared to a replica, the source is preferred as objects are permanently available. If this condition holds, the node will search its *Temporary Forwarding Interest Base (TFIB)* table for a corresponding entry that matches the object's name. If no entry exists, a new one will be created. If otherwise, the entry will be updated to the currently obtained metric. The search Data packet will be further propagated based on the incoming list of the PIT entry. The entry is then deleted and subsequent search Data packets are discarded. Upon the arrival of a search Data packet or the expiration of a search Interest packet, a consumer will start to retrieve the object.

Upon the arrival of an actual Interest packet, a node will search its CS table. If the requested object is not found, the node will search its TFIB table to forward the Interest. If a TFIB entry is not found, the Interest is propagated based on the information of the *Forwarding Interest Base (FIB)* table. A FIB table is structured based on a routing protocol's output. As before, Interest requests are hold as entries in the PIT table while subsequent Interests that refer to the same content are suppressed. *Interest suppression* is not applied on search Interests as different scope values may be contained. Merging these values would result in flooding search Interests to the wrong scopes. Upon the delivery of the last Data packet, i.e. upon the retrieval of an object, a caching decision is performed.

IV. SYSTEM MODEL

The evaluation is based on the ndnSIM simulator [1], an ns-3 module that adopts the CCN communication model [7]. A real network topology of 94 nodes, i.e. 39 backbone nodes and 58 gateway nodes is used. Nodes are equipped with a CCN stack. Consumer and producer applications can only be installed on gateway nodes. One producer, chosen based on the metric of degree-centrality is assumed; a node with degree-centrality 5 is chosen, where 1 is the minimum and 14 is the maximum. A mean value of 200 consumers is installed on each gateway, following a uniform distribution. Consumers issue object requests, i.e. a sequence of chunk requests of size equal to the object size divided by the chunk size, 10KB [6]. Request arrivals follow an exponential distribution of $\lambda = 1.0$.

To ease readers relate the results with those presented in [9], a catalog size of $|O| = 10^4$ and CS size of $cs_i = 10$ are chosen; due to memory restrictions, both parameters have been reduced by a magnitude of 10. Similar to [9], object sizes follow a normal distribution of mean 10MB and standard deviation 9.8MB [6] while object popularity follows a Zipf distribution of $\alpha = 1$. Objects in CS are replaced using the LRU policy [7].

TABLE I: Parameters of the system model used for evaluation.

Parameter	Symbol	Value	Definition
No. of backbones	$\lvert B\rvert$	39	Total No. of backbone nodes
No. of gateways	$\lvert G\rvert$	58	Total No. of gateway nodes
Capacity of links	BW	40GB	Available bandwidth
Catalog Size	$\lvert O\rvert$	10000	Total No. of objects
Object Size	o_i	$\forall o_i, i\epsilon\lvert O\rvert \sim N(10000KB, 9800KB)$	Size of object o_i in KB
Chunk Size	Ch	$10KB$	Chunk size in KB
Contents Size	$\lvert C\rvert$	$\sum_{i=1}^{\lvert O\rvert} o_i/Ch$	Total No. of chunks
Cache Size	cs_i	$\forall cs_i, i\epsilon\lvert N\rvert, cs_i\epsilon\{10\}$	Cache capacity of node i in cunks
Consumers Size	u_i	$\forall u_i, i\epsilon\lvert G\rvert \sim U(100, 300)$	No. of users on gateway i
Zipf Exponent	α	1.0	Exponent of the Zipf distribution
Arrival rate	λ	1.0	Exponential request arrival rate

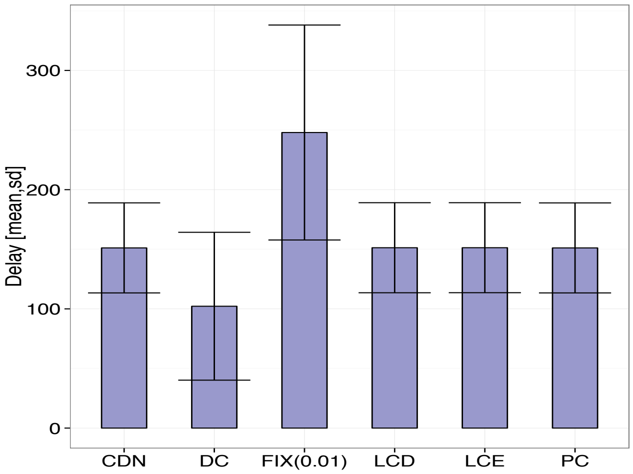

Fig. 1: Mean delay for the SPR strategy.

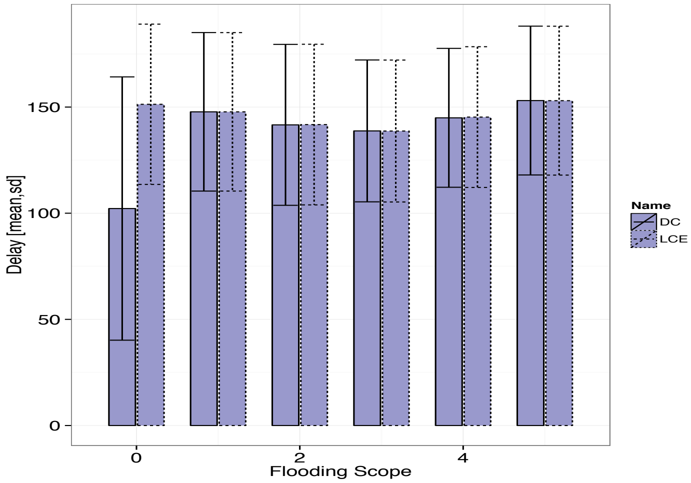

Fig. 2: Mean delay for the scope-flooding strategy.

V. EVALUATION

In this section, a report of the evaluation results of a number of caching policies with regard to SPR and scope-flooding forwarding is provided. In order to compare the results with those presented in [9], the same caching policies are taken into account while based on our previous work that concludes DC caching policy to perform the best among the alternatives [11], DC is also taken into account. To clarify the gains of ICNs against Content Delivery Networks (CDNs) [5], a CDN network is included as a benchmark. CDNs cache objects on the last node of a delivery path i.e. gateway nodes. To fairly compare the two technologies, the same total caching capacity is assumed, i.e. the caching capacity of a gateway is equal to the capacity of the network divided by the number of gateways.

Simulation results are based on the average and standard deviation of the delay metric over ten simulation runs. This metric represents the waiting time experienced by a client with regard to an object retrieval. Eventhough, the hop-count metric has also been considered for the evaluation of the caching policies, the results are omitted due to space limitations and due to the fact that they are quite similar to the results concluded for the delay metric. Caching policies are represented using abbreviations, i.e. *DC, FIX, LCD, LCE and PC* for the Degree-Centrality, Fixed probabilistic, Leave-Copy Down, Leave-Copy Everywhere and ProbCache caching policies. The *CDN* abbreviation corresponds to the CDN network scenario.

Fig. 1 illustrates the average delay and standard deviation of caching policies against the CDN scenario with regard to SPR. According to Fig. 1, the majority of caching policies provide similar gains to CDNs, with the exception of FIX(0.01) and DC. FIX(0.01) corresponds to an increase of 65% to the delay observed by the majority of policies, i.e. approximated at 151 seconds, while DC corresponds to a decrease of about 33% against the same reference. However, the standard deviation of the delay of DC is almost double of the delay experienced by most of the policies, i.e. about 62 seconds and 38 seconds, respectively. FIX(0.01) exhibits an even higher diversity of delay, with its standard deviation being close to 90 seconds.

Fig. 2 plots the average delay and standard deviation of DC and LCE with regard to scope-flooding. DC has been chosen based on the aforementioned results, as the optimal caching policy when SPR is applied while LCE has been used as a benchmark. To study the effect of scope parameter on the performance of the forwarding strategy, a variety of scope-values, $sv\epsilon\,[0,5]$, have been considered. A scope-flooding forwarding strategy of $sv = 0$ results in a SPR forwarding strategy while a scope-flooding forwarding strategy of $sv = 5$ is the widest illustration of the algorithm, equal to the maximum distance between consumers and the source.

According to Fig. 2, the performance of both LCE and DC may be considerably affected by parameter sv; the average delay decreases as sv increases from 1 to 3 and increases again for $sv \geq 4$. In contrast to the average delay, the standard deviation decreases for $1 \geq sv \geq 2$ and $sv = 4$ and increases again for $sv = 2$ and $sv = 5$. Due to the similarity of the average delay rates of DC to LCE that correspond to an increase of 1%-5%, only the delay rates of LCE are described. LCE average delay decreases from 148 to 142 seconds for $sv = 2$ and from 142 to 139 seconds for $sv = 3$ while increases from 139 to 145 seconds for $sv = 4$ and from 145 to 153 seconds for $sv = 5$. The standard deviation of both caching policies lies in the range of $\{33, 38\}$. Based on the results, one can conclude that scope-flooding forwarding may not always be beneficial.

Eventhough, both DC and LCE correspond to similar average delays with regard to scope-flooding, their average delay with regard to SPR is different; LCE exhibits a lower average delay using scope-flooding forwarding while DC exhibits a lower average delay using SPR. LCE results in a delay reduction of 13 seconds calculated by the average delay obtained using SPR forwarding and the lowest average delay observed using scope-flooding forwarding while DC results in a delay increase of 36 seconds with regard to the same reference points.

Based on the evaluation results, the highest benefits of ICN can be obtained by coupling a DC caching policy and a SPR forwarding strategy. Consequently, the gain of ICNs over CDNs can be as high as a reduction of 33% of the average delay experienced by consumers for the retrieval of an object.

The behavior of DC and LCE caching policies with regard to the increase of the delay metric can be explained based on the operation of the scope-flooding forwarding mechanism. In more detail, once a TFIB entry is created, the entry may be updated but not deleted. Based on section III, the entry is updated upon the arrival of a search Data packet triggered by a new object request. However, replicas may be deleted at any time, leading to incorrect TFIB entries. Upon the arrival of a content request, a node that has no related TFIB entries will forward the request based on its FIB table. Hence, none node that participates in the aforementioned communication will be aware of the missed replica and no action will be taken to prevent the forwarding of the content request on that route.

Based on the caching policy used, the impact of this behavior may result in minor differences, e.g. LCE or significant differences, e.g. DC. Since the likelihood to find a replica on a forwarding path using LCE is equal or higher than DC, content requests propagated based on out of date TFIB entries are more likely to be satisfied by an other node on the same forwarding path when LCE is applied. This outcome verifies the conclusion that caching policies and forwarding strategies are complementary features [3], [9] but contradicts the conclusion that scope-flooding forwarding is more beneficial than SPR [9].

To address the drawbacks of scope-flooding, alternative options for the maintenance of TFIB entries may be considered such as the deletion of an entry upon the expiration of a lifetime value, calculated based on the average replacement time of the node that holds the replica or the deletion of an entry upon the arrival of a notification message that is triggered once a miss of a replica occurs. We attempt to fulfill this as a future work.

VI. CONCLUSION

In this paper, we have explored the benefits of scope-flooding against SPR using a number of caching policies that we evaluated via simulations. Based on the results, scope-flooding that has been proposed as a forwarding mechanism able to enhance performance of a caching policy to rates close to the optimal, may not always provide such benefits; an increase of 36 seconds of the average delay of DC policy has been concluded. The overall performance of a network depends on the combination of forwarding stategies and caching policies.

VII. ACKNOWLEDGMENTS

This work is funded by the Higher Education Authority (HEA) and the European Regional Development Fund (ERDF).

REFERENCES

[1] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim: Ndn simulator for ns-3," *Technical Report NDN-0005, Named-Data Networking Project*, October 2012.

[2] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *Proceedings of the Re-Architecting the Internet Workshop of the ACM CoNEXT Conference (ReArch '10), Philadelphia, USA, November 2010*.

[3] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda, "Congestion-aware caching and search in information-centric networks," in *Proceedings of the 1st ACM Conference on Information-Centric Networking (ICN'14), Paris, France, September 2014*, pp. 37–46.

[4] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino, "Exploit the known or explore the unknown?: hamlet-like doubts in icn," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 7–12.

[5] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable icn," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications (SIGCOMM '13), Hong Kong, China, August 2013*, pp. 147–158.

[6] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '11), San Diego, CA, USA, October 2007*, pp. 15–28.

[7] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09), Rome, Italy, December 2009*, pp. 1–12.

[8] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Telecom ParisTech Ecole, Tech. Rep., November 2011.

[9] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proceedings of the 1st ACM Conference on Information-Centric Networking (ICN'14), Paris, France, September 2014*, pp. 127–136.

[10] ——, "On sizing ccn content stores by exploiting topological information," in *Proceedings of the 1st Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN '12), Orlando, FL, USA, March 2012*, pp. 280–285.

[11] S. Weber and A. Ioannou, "Information-centric networking: A thorough evaluation of popularity-based probabilistic on-path caching," Technical Report TCD-CS-2014-04, Tech. Rep., November 2014.