

# Vicinity-based Replica Finding in Named Data Networking

Atthapol Suwannasa, Matthew Broadbent

School of Computing and Communications

Lancaster University, UK

Email: {a.suwannasa, m.broadbent}@lancaster.ac.uk

Andreas Mauthe

University of Koblenz-Landau, Germany

Email: mauthe@uni-koblenz.de

**Abstract**—In Named Data Networking (NDN) architectures, a content object is located according to the content’s identifier and can be retrieved from all nodes that hold a replica of the content. The default forwarding strategy of NDN is to forward an Interest packet along the default path from the requester to the server to find a content object according to its name prefix. However, the best path may not be the default path, since content might also be located nearby. Hence, the default strategy could result in a sub-optimal delivery efficiency. To address this issue we introduce a vicinity-based replica finding scheme. This is based on the observation that content objects might be requested several times. Therefore, replicas can be often cached within a particular neighbourhood and thus it might be efficient to specifically look for them in order to improve the content delivery performance. Within this paper, we evaluate the optimal size of the vicinity within which content should be located (i.e. the distance between the requester and its neighbours that are considered within the content search). We also compare the proposed scheme with the default NDN forwarding strategy with respect to replica finding efficiency and network overhead. Using the proposed scheme, we demonstrate that the replica finding mechanism reduces the delivery time effectively with acceptable overhead costs.

## I. INTRODUCTION

Named Data Networking (NDN [1]) offers a new way of accessing content by allowing users to locate content objects by their names (rather than by their locations). These objects can be replicated in several caches and they can be forwarded from different sources to the users. Thus, NDN can overcome the drawbacks of the existing point-to-point communication and the current problems in the TCP/IP Internet [2].

In NDN nodes (including consumers, routers and producers) are the key elements in the content delivery chain. The consumers send Interest packets to look for desired content objects originally provided by their producers. The routers are responsible for two main tasks: caching content, and forwarding packets. In each router a so called *Face* represents a network interface that is mapped with a content name. This name is exchanged among NDN nodes to announce its reachability and its location. An NDN forwarding strategy then selects an appropriate Face (for a single-path strategy) or several Faces (for a multi-path strategy) to forward an Interest packet.

Tyson et al. [3] showed that content caching can significantly reduce path lengths for retrieving content because Interest packets can be served closer to the requester. Interestingly,

pushing and moving content into a specific area helps to actively increase content availability [4], resulting in faster delivery, and reduced congestion on long links [5]. Hence, it can be seen that content caching can, indeed, bring content closer to end-users. This then results in several benefits especially in terms of delivery efficiency. However, few research papers directly investigate issues related to finding nearby objects. The default best route strategy of NDN selects the Face with the lowest cost to find a shortest path to a content object according to its name. Though, the best direction to find the object may not be in the default path [6]. Content replicas might be close to the requester, and in these cases it is opportune to look more actively for them. This is because the content could be located in the neighbourhood, away from the default path, but nevertheless may still be the best source for the consumer. These would not be located with the default NDN strategy. Hence, the default strategy may not take full advantage of locality, which can result in a reduced delivery efficiency.

To address this issue we propose a vicinity-based scheme, focusing on finding replicas within a neighbourhood<sup>1</sup>. The scheme is able to locate nearby replicas that serve as a better source for a consumer. In order to determine the benefits of the scheme we perform experiments to evaluate its performance and overhead costs. RTT values are used to measure the content/replica delivery efficiency. This finds the optimal vicinity size as well as provides a comparison of the scheme to the default NDN strategy. Message overhead and data volume are also considered to understand the impact of the scheme on additional overhead.

The rest of the paper is organized as follows: Section II discusses background and related work. Section III introduces the vicinity-based scheme. In section IV, we perform the detailed evaluation. Section V discusses the results and main findings. Finally, section VI concludes the paper.

## II. BACKGROUND AND RELATED WORK

### A. NDN Forwarding Strategy

To forward Interest/Data packets, there are three core elements in each NDN router: a Pending Interest Table (PIT),

<sup>1</sup>Note, we define a *neighbourhood* as the set of network nodes around the requester, whereas the *vicinity* is made up of neighbouring nodes that are within a specific, well-defined distance to the requester.

a Forwarding Information Base (FIB), and a Content Store (CS). The PIT stores Interest packets that have been forwarded and are waiting for their content objects. The FIB contains forwarding entries (Faces with name prefixes and costs) to be used in the forwarding strategy. This strategy determines Faces (e.g. by the lowest cost) to forward the Interest packets. The returning content objects are replicated in each router's CS along the path from their sources to their requesters.

In the NDN routing plane, a routing protocol is crucial to compute and insert forwarding entries into a FIB. Named-data Link State Routing protocol (NLSR) [7], for instance, is a routing protocol that advertises the reachability of name prefixes to every router's FIB. By design, the NLSR's overhead grows dramatically [8] (especially the FIB's entries). To mitigate this problem, a hyperbolic routing protocol [8] uses the coordinates of NDN nodes to calculate a possible shortest path to a producer. Several forwarding strategies (e.g. [9], [10]) aim to find the shortest paths to desired objects. However, nearby replicas in a vicinity of consumers, which are often located off those paths, might be opportunistically fetched to achieve a higher delivery efficiency.

### B. Replica Finding

Consumers can gain advantages on fetching nearby replicas (e.g. lower RTT, and higher hit-rate). Several proposals (e.g. [4], [11], [12]) try to pro-actively push more replicas to a specific area to increase content availability. However, there are few proposals on investigating finding these replicas, which are already cached close to consumers.

Cao et al. [6] have proposed a design, named Fetching the Nearest Replica (FNR). A tracker server calculates a nearest path to forward an Interest packet to fetch a popular replica. However, the centralized design may not take full advantage of the decentralized concept of NDN [13], [14] especially in dynamic topologies. Similarly, in [15], a topology is divided into different domains and a centralized controller in each domain is responsible for forwarding decisions.

Scoped-flooding [16], [17], [18] is usually applied to find off-path content with a certain probability. A hop limit is usually used to restrict a scope. However, flooding Interest packets might create several data chunks in return, resulting in an increased cache replacement rate [16]. In [19], [20], several attributes related to real time network conditions are considered to select a Face with a high probability. Nevertheless, by considering many metrics, the scheme needs a high calculation time to make a forwarding decision [21]. In addition, the scheme may not locate replicas that are already cached nearby.

A multi-path Interest forwarding strategy might increase the opportunity to locate nearby replicas. Each NDN router forwards every Interest to all upstreams, according to the supplied FIB entries. However, the increasing of overhead costs might impact the entire network.

## III. THE SCHEME

Several nearby replicas are often cached in a vicinity. However, the concept of proactively finding replicas in a

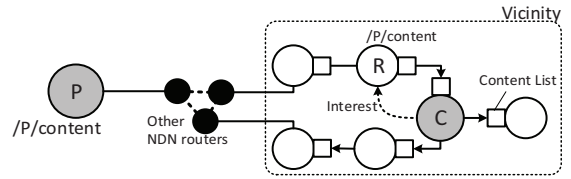


Fig. 1. Core Scheme

vicinity does not exist in the current NDN architecture. Hence, the main objective of this proposal is to locate these nearby replicas in an effective manner. Consider, for instance, browsing to the first page of Google (a content object), where the source of the content is located far away or it is unreachable (e.g. in a dynamic situation, where connectivity is not fixed). The first node requests this content object and it is then replicated in the vicinity. The next consumer who needs the same content can benefit by instead looking in the vicinity for the object, rather than going to the original producer.

### A. Vicinity

By centering a consumer node, a vicinity contains the set of NDN nodes who are connected to the consumer in different distances [4]. A distance defines the number of hops to the center node and is less than or equal to a threshold. A vicinity size is a threshold number that defines the scope of its vicinity. The number of nodes in the vicinity depends on their topology and the number of links around the consumer.

This proposal expands the consumer's view into a vicinity. Unlike the narrow view of the current NDN paradigm, the consumer can gain more advantages in a larger view, especially in cases of content finding. However, the scope can be too wide or too narrow. Thus, our experiments also investigate the influence of the vicinity size.

### B. Content List

In the default NDN best route strategy, a router selects a Face with the lowest cost to fetch a content object indicated by its name. It does not consider nearby replicas. Hence, to find these replicas, consumers must know the availability of content objects or replicas in their vicinity.

If an NDN node is a producer, in its CS, it has its own content objects, but if it acts as a forwarder, the CS also stores replicas of other producers. To provide the availability of the content objects/replicas, we design a *Content List* that contains the name prefixes from the CS. The list is used to advertise the availability of the content objects and replicas from a requester to other requesters in a vicinity. To reduce the list's size, we ignore the name prefixes of NDN default (e.g. NLSR messages) and local name prefixes.

### C. Content List Pushing Operation

Our core idea is conceptually simple that each NDN node has a *Content List* to be pushed to announce the availability of content objects and replicas in its CS to other nodes in its

vicinity. A node who needs a content object can try to find nearby replicas by looking at the pushed list first.

As presented in Fig. 1, we assume that the producer P produces a content object with the name prefix “/P/content” and this object is already requested and replicated in the R’s CS. When the router R receives the object, the *Content List* is then pushed to other downstream nodes in the vicinity except the upstream node who has forwarded the object. When a new list passes through a node, the entries (content/replica names) from the list are then mapped with the Face that has received the list. Now, the consumer C knows what the current content objects/replicas are in the list owner and which a proper Face is to be used to find these content objects/replicas.

#### D. Replica Finding Operation

In the replica finding operation, as shown in Fig. 1, when the consumer C starts to request the content object, it then checks the existing of the content name in the pushed *Content List* by the router R. At the current stage, this name can be found in the list and the consumer C knows where to find the replica. If the name does not exist, the NDN default strategy is activated to find the content object.

#### E. The Forwarding Strategy

To fetch the content “/P/content”, the consumer C selects the Face to the owner of the list that contains this name. The Face is already mapped with the name of the replica as described in Section III-C. There could be other Faces mapped with the similar name but we focus the Face to the nearest replica (the router R). Then, an Interest packet is originally created at the consumer C and it will be reversely forwarded to each Face of each router until it reaches the node that is the owner of the list (the router R). The replica will then be fetched without sending any requests to the further producer.

### IV. EVALUATION

To evaluate the performance of our scheme in realistic-like conditions and to support our scheme operations, we modify Mini-NDN [22]. The networking emulation tool that integrates with the NDN libraries, NFD, NLSR, and NDN-tools released by the NDN project. The emulation tool can perform a fully running system and can experiment the scheme as close as possible to reality [23].

#### A. Experimental Model

This model serves as the basis to study the proposed scheme to support vicinity-based replica finding in NDN. The NDN nodes and the content objects are designed to emulate a replica finding situation. The nodes can be either producers, routers, and consumers. A producer creates content objects to be requested by the consumers (requesters). After caching a replica at a requester, the *Content List* is then updated and pushed to other nodes in the requester’s vicinity. The requester can be considered as a previous requester, if there is a node in the vicinity that tries to find the same content. The node, as a consumer, can check the pushed list from the previous

requester. If the name of the replica is in the list, it means that the content object can be fetched nearby. Notably, the replica might be cached in many nodes in the vicinity, but we focus on the nearest first.

#### B. Topology and Scenarios

**Topology:** We inspect a router-level network with a limited scale to better understand the flows of packets. The main topology is composed of 22 nodes and 50 links as shown in Fig. 2. This topology is based upon a snapshot of the actual NDN testbed, introduced in [8]. To consider only the number of hops, we assume that each link has a homogeneous latency of 10ms (the same cost).

**Scenarios:** We employ two main scenarios in the evaluation. The first scenario analyzes the delivery efficiency of the proposed scheme by investigating how much can the average round trip time (RTT) be reduced as compared to the default NDN mechanism. We fill in each cache with the replicas depending on the replica densities. Replica density is defined by the percentage of the number of distinct replicas in the network compared to the potential number of distinct locations that they could be cached. A higher percentage means a higher number of replicas in the topology. Each cache is then fixed to prevent that successive request effect the further outcome of the test, i.e. an increasing of the replica density is prevented. In each density we then execute 30 requests to get an average RTT by ignoring any RTT from local caches.

In the second scenario, we aim to understand the message overhead costs of our scheme against the default NDN. We execute 30 runs in every experiment. To evaluate the effects of the number of links, we set only one requester in each time step. We increase the number of requesters from 5 to 35 to study the impact of the number of requesters for requesting random content objects. Furthermore, the scheme might create additional traffic due to the *Content List* pushing mechanism. Hence, we also analyse the average data volume in each node. We inspect from the small number of requests to the higher number of requests (the beginning to the end of the experiment (time  $t_1$  to  $t_{30}$ ))

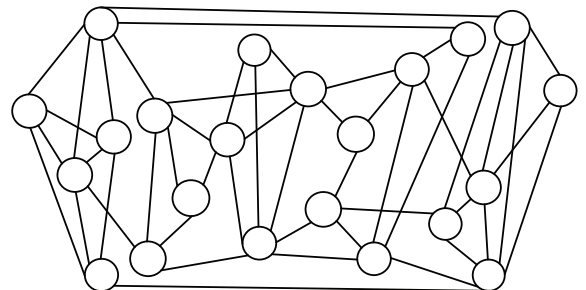


Fig. 2. Topology

#### C. Vicinity and Content Placement

In our hypothesis, the vicinity size should not be too wide. Hence, to evaluate a narrow vicinity, the size is set to 1 hop.

We expand the vicinity size to 2 hops and 3 hops to evaluate the larger view of replica finding.

In each experiment, a producer is randomized and then random consumers request a content from the producer to create replicas depending on the percentage of their density.

A content object is a piece of data that is provided by a producer and consumed by a consumer. Each object is created in one chunk and has the same size (1024 bytes).

By comparing our scheme in different vicinity sizes with the default NDN operation, we study three main experiments to study the overhead costs. The first experiment investigates the overhead costs against the number of links. We analyze the average number of Interest packets (message overhead) in each request that are produced by our scheme. The second experiment considers the average message overhead costs, varying the number of requesters (5 to 35). In the third experiment, we aim to understand the data volume that might affect the traffic in each NDN node. The position of producers, replicas, and consumers are randomly selected.

#### D. Metrics

In our evaluation, we focus on three main aspects: delivery efficiency, message overhead, and data volume. We analyze these aspects by comparing our scheme with different vicinity sizes against the default NDN. The itemized metrics below are used to evaluate them.

- **RTT:** the average RTT per request by a consumer to retrieve a desired content object. It is used to measure the delivery efficiency. We try to evaluate the replica finding performance of the consumer. We assume that there is a replica that is already cached (the filled in replicas). A lower RTT indicates that a consumer can fetch a desired content object faster.
- **Message Overhead:** After finishing a request, the Interest messages are counted from every node. The average message overhead is calculated from each request. We focus only messages that are involved in content finding by ignoring some system messages such as NLSR messages or localhost messages. A higher average number of messages might increase more traffic. However, the number should be considered that it is acceptable or not to perform a better performance in terms of content delivery.
- **Data Volume:** The average data volume in each node is considered to understand how much the pushing *Content List* scheme can affect the traffic of each NDN node. To understand the traffic in each node, the data volume results include every packets that pass through each single node. The higher data volume means the higher traffic. It might impact the entire network.

## V. RESULTS

This section presents and discusses the results of the experiments by considering the aforementioned metrics.

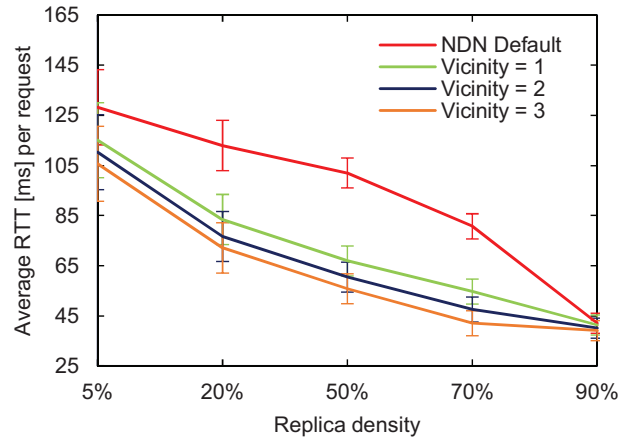


Fig. 3. Average RTT per request performing by each consumer varying vicinity sizes with different replica densities

#### A. RTT

This analysis is to indicate how well the default NDN deliveries content and to compare the results with the proposed scheme. The results of average RTT per request with error bars that represent 95% confident interval are shown in Fig. 3.

The results indicate that our scheme benefits in almost every replica density. When the replica density is low (e.g. 5%), it means that there are few replicas within the network. Hence, the RTT is comparatively high for both, default NDN and the Vicinity based schemes. For example, it is 128.26 ms for the default NDN and 115.2 ms for a vicinity size of 1. When the replica density is medium (e.g. 50%), the vicinity based scheme can gain most advantages compared to the default NDN since there is a higher number of replicas and the larger radius of our scheme increase the opportunity that replicas are located in the neighbourhood. When the vicinity size is set to 1, the average RTT is approximately 67.86 ms while the default NDN is around 101.98 ms. When the replica density is high (e.g. 90%), the replicas are located at almost every node. Hence, the replicas can be easily located, not just in the neighbourhood but also in the NDN default path. This is why in this specific case the differences between our scheme and the default NDN are marginal.

#### B. Message Overhead

In this analysis, we consider two factors that can affect the results. First, the number of links are studied. In Fig. 4, the results of the average message overhead are quite stable when using NDN default and the scheme with 1-hop vicinity. It may be slightly higher than the default NDN because the scheme has to push a *Content List* to update the content availability in the vicinity. The are slight increments compared to the default NDN in case of 1-hop vicinity. The overhead results are slightly increased from 30 links to 50 links, when the vicinity size is set to 2. There are around 14 packets that are added from the default NDN in the 50 link case. When the vicinity size is set to 3, the results seem to be too high.

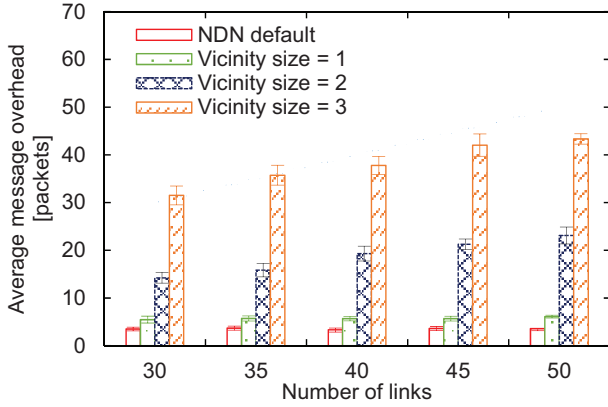


Fig. 4. Average message overhead varying the number of links

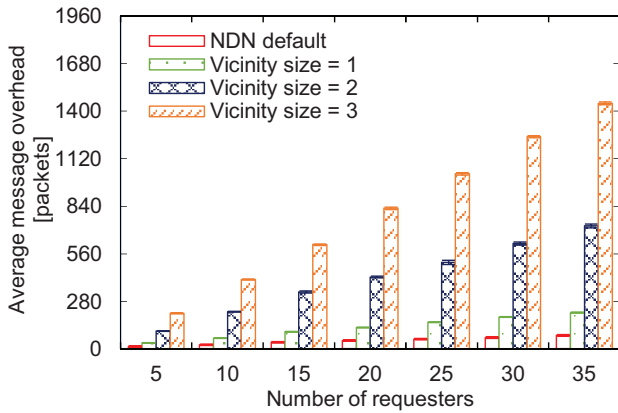


Fig. 5. Average message overhead varying the number of requesters

Second, the effects that the number of requesters has on the message overhead are investigated, as shown in Fig. 5. When we increase the number of requesters from 5 to 35, the results are slightly increased by using the default NDN. Likewise, there are slight increments, when the vicinity size is set to 1. There are additional 21 packets, when the vicinity size is set to 1 with 5 requesters. The overhead costs seem to be high when the vicinity size is set to 2 or 3. The overhead packets are increased to approximate 93.7%, when the vicinity is set to 3 with 5 requesters. It can be seen that a larger vicinity means a higher amount of nodes in the vicinity that the scheme must push the *Content List* to and the more number of requesters creates a greater number of packets to be pushed. The message overhead growth is almost linear because when the vicinity size expands and the number of requesters increases, each requester has to push the more number of *Content Lists* into a larger vicinity. If we increase the number of requesters (more than 35), the graph would grow in the same pattern.

### C. Data Volume

According to Fig. 4 and Fig. 5, the message overhead results seem to be higher when the vicinity size is set to 2 or 3 compared to the default NDN and the scheme with 1-hop

vicinity. The results are quite high especially with a 3-hop vicinity. Hence, the goal of this analysis is to quantify how much the data volume overhead can be occurred in each node. The data volume can indicate the traffic that a single node has to process and it can lead to understand the entire network traffic. In the default NDN and the scheme with a vicinity size of 1, the data volume results are approximately 0.57MB. They are quite stable from time  $t_1$  to  $t_{30}$  because there is a small number of packets that has to be sent. These have to be sent and pushed by using the scheme with 1-hop vicinity. Although, the results include all messages (e.g., routing messages, and messages created by our scheme), the data volume results are still low (less than 0.8MB in 30 requests) because of the small size of the pushing packets.

When the vicinity size is set to 2, the results are slightly increased from time  $t_1$  to  $t_{30}$  because when time passes, there are more packets to be processed. Similarly, the increments of the data volume are higher, when the vicinity is 3. Interestingly, although the message overhead costs seem to be high according to the results in Section V-B, the differences of the data volume results are negligible (only 12% of increment in the time  $t_{30}$  (the worst case)). It can indicate that a higher number of messages overhead does not mean that the data volume will be high. The pushing packets are quite small and do not introduce a huge impact to the network traffic. However, it is important to keep in mind that the more packets created, the more processing an NDN node requires to process them, which may overload the network. So, the vicinity size should not be too wide to keep the overhead as low as possible, while introducing impressive results.

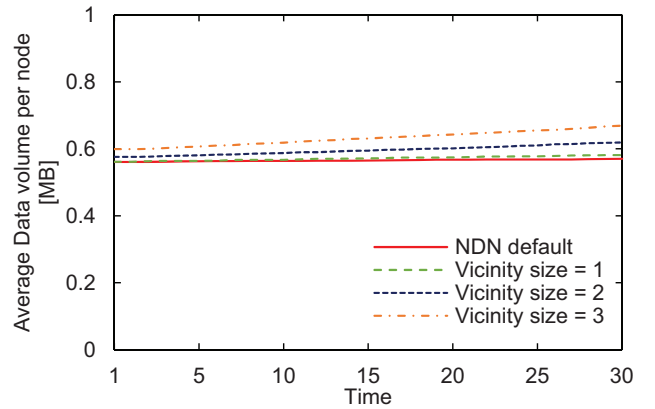


Fig. 6. Average data volume per node from time  $t_1$  to  $t_{30}$

### D. Discussion

In summary, considering the additional benefits and costs, the experiment results show that a 1-hop vicinity tends to be the optimal vicinity size since there is considerable benefits while the overhead costs are still low compared to the default NDN case. In some particular cases, a 2-hop vicinity might be still worthwhile due to the further decreased RTT. For example, where a producer is quite far from a requester and

the neighbourhood within which there is interest in the content is larger than 1 hop.

The experiments have demonstrated the effects of content location on the proposed scheme, mainly within a fixed network environment. Though, there are other challenges where the proposed scheme can have benefits. In the context of mobility, the scheme might help to find nearby replicas, even though the original producer of the content is going offline. For instance, content distribution in mobile environments is a challenge, e.g. in scenarios such as platooning and dynamically changing topology. Locating replicas with different popularity distributions can help to understand the effects of content distributions. For example, if we have a large topology with a few consumers, it means the network is less populated, it might have some effects in the replica finding results. These challenges would be explored in our future work.

## VI. CONCLUSION

Most current NDN content selection schemes focus on finding a content object indicated by its name within the path from the requester to the source. We argue that it might be opportune to be more efficient looking for nearby replicas within a vicinity. It is shown that the proposed vicinity-based replica finding scheme can achieve a higher delivery efficiency especially in the case of the medium number of replicas (50% density). The paper also discusses the additional costs from the proposed scheme. As a result we suggest that the optimal vicinity size should be 1, since overhead costs remain low in this case. However, it can be expanded to 2 in particular circumstances to increase the larger view of content finding. For example, where a producer is quite far from a requester and content may be of interest within a wider neighbourhood this could be of advantage. In contrast, when a producer distance and a replica distance are further away, the vicinity size of 3 can still slightly reduce the RTT but the overhead costs increase much more, outweighing the benefits. Although in general the benefits of the scheme are not for free, the introduced overhead costs are still limited for small size vicinities. This is also due to the fact that the packets for updating *Content Lists* are quite small and do not create a huge impact on the overall network traffic.

In our future work we will also study the characteristics of content popularity and distribution. For instance, the benefit of the vicinity based scheme for highly popular content could be different compared to medium or less popular content. Interestingly, applying the scheme in the context of mobility would be more challenging.

## REFERENCES

- [1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.
- [2] L. Wang, V. Lehman, A. K. M. M. Hoque, B. Zhang, Y. Yu, and L. Zhang, "A secure link state routing protocol for ndn," *IEEE Access*, vol. 6, pp. 10470–10482, 2018.
- [3] G. Tyson, S. Kaune, S. Miles, Y. El-khatib, A. Mauthe, and A. Taweel, "A trace-driven analysis of caching in content-centric networks," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, July 2012, pp. 1–7.
- [4] M. B. Lehmann, M. P. Barcellos, and A. Mauthe, "Providing producer mobility support in ndn through proactive data replication," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 383–391.
- [5] N. Abani, G. Farhadi, A. Ito, and M. Gerla, "Popularity-based partial caching for information centric networks," in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, June 2016, pp. 1–8.
- [6] J. Cao, D. Pei, X. Zhang, B. Zhang, and Y. Zhao, "Fetching popular data from the nearest replica in ndn," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2016, pp. 1–9.
- [7] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "Nlsr: Named-data link state routing protocol," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 15–20.
- [8] V. Lehman, A. Gawande, B. Zhang, L. Zhang, R. Aldecoa, D. Krioukov, and L. Wang, "An experimental investigation of hyperbolic routing with a smart forwarding plane in ndn," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, June 2016, pp. 1–10.
- [9] H. Dai, J. Lu, Y. Wang, and B. Liu, "A two-layer intra-domain routing scheme for named data networking," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 2815–2820.
- [10] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, "Navigo: Interest forwarding by geolocations in vehicular named data networking," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2015, pp. 1–10.
- [11] H. Farahat, R. Atawia, and H. S. Hassanein, "Robust proactive mobility management in named data networking under erroneous content prediction," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.
- [12] H. Farahat and H. S. Hassanein, "Supporting consumer mobility using proactive caching in named data networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.
- [13] K. Shilton, J. Burke, C. Duan, and L. Zhang, "A world on ndn: Affordances & implications of the named data networking future internet architecture!" 2014.
- [14] V. Sivaraman and B. Sikdar, "Hop-count based forwarding for seamless producer mobility in ndn," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.
- [15] N. Aloulou, M. Ayari, M. F. Zhani, and L. Saidane, "A popularity-driven controller-based routing and cooperative caching for named data networks," in *2015 6th International Conference on the Network of the Future (NOF)*, Sept 2015, pp. 1–5.
- [16] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 127–136. [Online]. Available: <http://doi.acm.org/10.1145/2660129.2660153>
- [17] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, A. Sathiseelan, and J. Crowcroft, "Pro-diluvian: Understanding scoped-flooding for content discovery in information-centric networking," in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '15. New York, NY, USA: ACM, 2015, pp. 9–18. [Online]. Available: <http://doi.acm.org/10.1145/2810156.2810162>
- [18] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, June 2016, pp. 1–7.
- [19] K. Lei, J. Wang, and J. Yuan, "An entropy-based probabilistic forwarding strategy in named data networking," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5665–5671.
- [20] K. Lei, J. Yuan, and J. Wang, "Mdpf: An ndn probabilistic forwarding strategy based on maximizing deviation method," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.
- [21] N. Aloulou, M. Ayari, M. F. Zhani, L. Saidane, and G. Pujolle, "Taxonomy and comparative study of ndn forwarding strategies," in *2017 Sixth International Conference on Communications and Networking (ComNet)*, March 2017, pp. 1–8.
- [22] Mini-ndn: A mininet based ndn emulator. [Online]. Available: <https://github.com/named-data/mini-ndn>
- [23] C. M. Cabral, C. E. Rothenberg, and M. F. Magalhães, "Mini-ccnx: Fast prototyping for named data networking," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 33–34.