Providing Producer Mobility Support in NDN Through Proactive Data Replication

Matheus B. Lehmann, Marinho P. Barcellos
Institute of Informatics
Federal University of Rio Grande do Sul, Brazil
Email: {mblehmann, marinho}@inf.ufrgs.br

Andreas Mauthe
School of Computing and Communications
Lancaster University, UK
&
KOM, TU Darmstadt, Germany
Email: a.mauthe@lancaster.ac.uk

***Abstract*—Named Data Networking (NDN) is a novel architecture expected to overcome limitations of the current Internet. User mobility is one of the most relevant limitations to be addressed. NDN supports *consumer* mobility by design but fails to offer the same level of support for *producer* mobility. Existing approaches to extend NDN are host-centric, which conflicts with NDN principles, and provide limited support for producer mobility. This paper proposes a content-centric strategy that replicates and pushes objects proactively, and unlike previous approaches, takes full advantage of NDN routing and caching features. We compare the proposed strategy with default NDN mechanisms regarding content availability, consumer performance, and network overhead. The evaluation results indicate that our strategy can increase the hit rate of objects by at least 46% and reduce their retrieval time by over 60%, while not adding significant overhead.**

I. Introduction

Mobile traffic has been increasing yearly and has now become a central part of computer networks. In 2014, 7.4 billion mobile devices increased their monthly traffic by 69% compared to 2013, generating 2.5 exabytes per month [1]. Despite this growing trend, today's Internet architecture fails to provide an adequate user mobility support with mechanisms such as Mobile IPv4 [2] and Mobile IPv6 [3].

Named Data Networking (NDN) [4], a network architecture proposal, aims to address some of the Internet current shortcomings, including support for user mobility. There are three key elements in NDN: content, users, and routers. Content is comprised of objects, which are single pieces of information. Users are entities that can produce, provide or consume objects. Routers are (static) devices which route and forward packets. They do not consume nor produce content, but can cache objects and provide them to interested consumers.

User mobility represents a challenge because users are associated with a device, which has limited resources (e.g. storage, bandwidth, battery), and which can be unavailable for a period of time before reappearing somewhere else. NDN natively supports *consumer* mobility: when mobile consumers move on an NDN network, they do not need to restore connections, sessions or update their locations. Instead, it is enough to re-send interest requests for non-retrieved data to fetch the desired content objects, which is possible because objects are idempotent, and NDN does not rely on end-to-end communication. In contrast, NDN does not support *producer* mobility. The difference compared to the consumer mobility is that the producer has to maintain the content objects available and provide them to requesters in spite of its possible movement or unavailability [5].

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

Producer mobility can be divided into two distinct periods: unavailability and re-attachment. While the former is characterized by the producer's lack of network connectivity during movement, the latter refers to the process of rejoining the network and restoring producer connectivity. The unavailability period is a more relevant research challenge, for it lacks proper support in current NDN. The latter period, in contrast, can be reasonably addressed through announcement messages.

In this paper, we focus on supporting producer mobility in NDN by keeping the content available. We propose a strategy based on a straightforward principle: proactively and efficiently replicating content by pushing it according to a placement policy. Its objective is to reduce the negative effects of producer mobility into content availability, without incurring significant performance loss for consumers nor overhead to the network. Unlike previous proposals, our strategy follows NDN principles and leverages its key features to overcome the loss of content availability induced by producer mobility.

The contributions of this paper are twofold. First, we propose a strategy to handle producer mobility in NDN based on data replication and aligned with NDN core principles. The strategy is able to keep content available despite producer mobility, with limited overhead. Second, we perform a detailed evaluation to study in which conditions data replication improves producer mobility support in NDN. The analysis focuses on understanding the impact of three key parameters of the strategy: vicinity size (producers knowledge about the network), replication degree, and placement policy. We measure their effects in two scenarios: a simpler case with a single producer, to highlight the effect of each strategy parameter; and one with multiple producers, to measure the effects of widely using the strategy, looking at the impact of the producers' local decision on the global results achieved.

The rest of the paper is organized as follows. Section II presents the state-of-the-art in NDN producer mobility. Section III describes the proposed strategy to handle producer mobility via data replication. Section IV presents the methodology used in our work to evaluate the proposed strategy. Section V discusses the results and main findings. Section VI concludes this paper and discusses future work.

II. State-of-the-art

In this section, we present and discuss the most recent advances in supporting the producer unavailability period in NDN. First, we define more precisely what is the unavailability period and discuss ways a network architecture can address it. Then, we organize existing proposals into four categories and discuss each one of them.

A. Unavailability Period

The unavailability period occurs when the producer is unable to provide its content. A network architecture can address this in three ways: provide no support, reduce the producer unavailability through host-centric mechanisms, or keep the content available through content-centric ones. These three alternatives are discussed next.

The simplest approach is not to provide any extra producer unavailability support. In other words, the network architecture does not have any feature to aid the producer in keeping itself or its content available during this period. This lack of producer mobility support is detrimental to applications that require the producer (or its content) to be accessible most of the time. Nevertheless, this approach can be acceptable in scenarios where the producers move but remain connected.

The second way focuses on reducing the producer unavailability via host-based mechanisms. The network architecture can use seamless mobility or connection restore. The former aims at minimizing the unavailability during a hand-off. The latter stores the communication when the producer is unavailable and restores it once the producer rejoins the network.

The third and last way, content-centric, keeps the content available despite the producer mobility. A network architecture usually employs data replication and caching to fulfill this goal. This kind of approach is seen in Content Delivery Networks (CDN), Distributed Hash Tables (DHTs), and NDN. This approach enables other network elements besides the producer to provide the requested data.

B. NDN Proposals

The NDN architecture does not address producer unavailability by design. Recent proposals to extend it can be separated into the following categories: (*i*) proactive content push, (*ii*) store and forward requests, (*iii*) use of NDN default or extended support, and (*iv*) use of non-NDN techniques.

Proactive push. [6] aims at maintaining the content available through proactive replication, instead of keeping the producer available at all times. It focuses on the case that the producer moves during a data transfer. Prior to moving, the producer pushes data proactively towards the requester. The router that receives the data stores it, enabling future requests to be satisfied on behalf of the producer.

Store and forward requests. This category focuses on avoiding the loss of requests and the need for consumers to re-issue them. [7]–[11] propose the addition of a network element responsible for storing requests when the producer is unavailable and forward them once the producer returns. The difference between them is how requests are forwarded when the producer returns: updating the FIB tables [8], using an indirection point [7], existing NDN features or a combination of them [9]–[11].

Using NDN default communication or extending its messaging protocol. [12], [13] map persistent and temporary data names of mobile producers. The producer updates its binding information through the existing NDN messages. [14] uses the NDN messaging protocol to notify the network when it detects a degradation in the current link signal caused by movement. The notification allows the routers to react and maintain the reachability toward the producer. [15] proposes Kite, a scheme that uses routable anchors to track the producer movement. It extends the NDN protocol, using PIT entries to create breadcrumbs from the anchor to the producer through the use of traceable interest packets. [16] proposes a solution based on name resolution that extends the interest packets to contain a hint of where the content might be located.

Non-NDN techniques for mobility support. [17] uses greedy routing, which can coexist with NDN default routing protocol, combined with indirection points. [18] proposes Auspice, a global name service to provide a low lookup latency, small update cost, and high availability. Despite not focusing on NDN, the solution can be applied to name-based communication such as NDN. [19] combines Software Defined Networking (SDN) with NDN to perform global and local FIB updates. It reduces the cost of routing information updates by limiting its scope when handling mobility.

Unlike previous work, host-centric, the proposed strategy supports producer mobility in a content-centric fashion by increasing the availability of its content in the network. Although [6] refers to a similar idea, it is a very limited study. Related proposals provide limited support for producer mobility to date and their designs conflict with NDN principles [4].

III. Data Replication Strategy

In this section, we present our proposed strategy to address producer mobility in NDN through data replication. The overall objective of the proposal is to increase the content availability and minimize the impact of the producer unavailability. It is based on a straightforward principle, i.e. proactive replication of content by the producer. When a producer creates a content object, it may push one or more replicas to other users. The strategy has five aspects, to be discussed: (*i*) vicinity, (*ii*) content push operation, (*iii*) data replication degree, (*iv*) content placement policy, and (*v*) producer re-attachment.

A. Vicinity

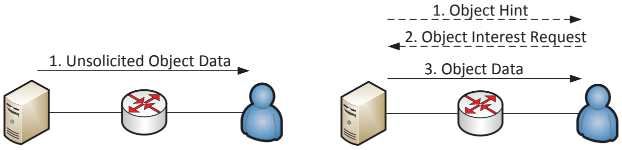
The vicinity of a device is defined as the set of nodes whose distances from the device are less than or equal to a threshold.

It can vary from the direct neighbors to the complete network. In the current NDN, a user device knows only about the router to which it is connected, which is the same as having threshold one.

We argue that a producer can benefit from a larger view and information about other devices. The proposed strategy extends NDN by expanding the device view to a vicinity, whose size varies with the topology and the device threshold. Useful information can be the availability of other devices, the interest on the content object, or the number of existing replicas in the vicinity to infer their popularity or rarity. Presently, we focus on the availability of devices. The producer attempts to obtain such information about devices in its vicinity, keeping it as soft state, and influencing the placement of content objects.

B. Content Push Operation

The concept of pushing data does not exist in the current NDN architecture, as the content dissemination in NDN is reactive rather than proactive (receiver-driven). There are two alternatives we could use to provide pushing in the NDN architecture: through unsolicited data or hints. They are exemplified in Figure 1.



(a) Unsolicited Data

(b) Hints

Fig. 1. Pushing operation using unsolicited data or hints

In the first alternative (Figure 1(a)), the producer has to 'force' the content to some other devices. This operation could be achieved by changing the router protocol to add new packet types or semantics, allowing producers to send their content to any other device on the network at any given moment. However, this alternative goes against the NDN principle of receiver-driven communication and is less secure.

Therefore, we consider a second alternative (Figure 1(b)), in which producers suggest to other devices that they should request a given content object. A device that receives the indication may follow it or not. In the case of using the tip, the user requests the content and retrieves it from the producer. This alternative is implemented at the application layer, which makes it compatible with the NDN architecture design. Its downside is the number of exchanged messages (two interest and one data packets) and the time to complete the operation, which decreases the content dissemination performance.

C. Data Replication Degree

For each content object created, the producer decides how many copies to push, to improve availability. Considering the potentially high cost of this operation for both the producer and the network (e.g. a single replica will double the object transmission cost for the producer), we expect only small replication degrees (1-5) to be affordable and, even so, not applicable to all objects/producers.

Thus, to use network resources efficiently, the number of copies pushed is inversely proportional to the producer availability. In other words, a less available producer pushes more copies than a highly available one. A maximum replication degree parameter is used to define how aggressive the producer can push its data; it can take into consideration content popularity, rarity or available device resources.

D. Content Placement Policy

A producer uses its vicinity and replication degree to build a ranking of devices to receive a copy of an object. The ranking is built according to a set of metrics collected from the vicinity, such as device availability, interest in objects, movement patterns, and stability. For the moment, we consider two placement policies: *random* and *availability-based*. The former policy only requires knowledge about the vicinity, and dispenses with any specific information about the devices themselves, while the latter takes advantage of device availability information.

E. Producer Re-attachment

When a producer rejoins the network at a different location, its content has to be made available to other users. Specifically, the routers must update their routing information to route interest requests to the correct producer location. Currently, the strategy relies on the NDN's default mechanism to perform this operation, i.e. each provider re-announces itself at the new location, and routers propagate this information to converge their routing tables. This decision adds an announcement cost and a routing convergence time to the strategy performance.

IV. Methodology

The evaluation is based on a scenario with the following characteristics: a network in which every user is potentially mobile and a producer, users are not always available, content is user generated, and interest for content objects is location biased. This section describes the evaluation model, scenarios, parameters, and metrics.

A. Evaluation Model

The model describes an NDN network with mobility and serves as the basis to study the proposed strategy to support producer movement. Its fundamental elements are the devices and the content objects in the network. A device is characterized by a mobility pattern, interest in content objects, and available resources. Devices can be either users or routers. User devices are mobile elements that produce, provide, and consume content. Routers, static components that form the network infrastructure, route requests, forward data, and cache content objects to serve future requests.

A content object is a piece of information with some popularity that is produced and consumed by users. We assume that every object has the same size and is composed by one chunk. The life cycle of a content object consists of its creation, pushing, request, and caching, as discussed next.

A content object is **created** by a user (its producer), which can execute three actions: satisfy incoming interests for the object, push it to other devices or announce its availability. We assume that each user has a home network, which is pre-configured in the routers. As a result, the producer only announces an object creation when away from its home network.

After creating a content object, the producer can **push** it to other devices. The producer decides how many replicas are created and where they are placed in the network based on the strategy description in Section III-D. Each device receiving a copy of the object becomes a provider of this object, announces its possession, and can satisfy future interests for it. The routers in the path between the producer and a new provider store a copy in their cache and update their routing information for this content object with the new provider.

Whether an object was pushed or not, it may be sought by a consumer **request**. In the evaluation, we consider static consumers only, which means they do not move while receiving an object. When users request objects, they retrieve data from the closest provider, according to the routing information in routers. If a router in the path has the object in its cache, it will provide the data for the consumer instead of the provider. During the retrieval, the content object is cached in the routers in the path between the provider and the consumer. If not a single provider is available during the request, the retrieval fails, and the consumer re-issues a new interest request in the subsequent interval.

Throughout the lifetime of a content object, its copies may be **cached** by both users and routers. Users can retrieve any content object they desire but only become providers of the subset of objects that they decide to announce. Routers cache every data that goes through them. The cache of every device has a maximum size, which forces the device to replace old entries using LRU when caching an object in an already full cache. Further, a cache entry in a router has a maximum lifetime to emulate other traffic that goes through it.

B. Scenarios

We employ two scenarios in the evaluation, according to the number of objects/producers: one or multiple. In the first scenario, we analyze the benefits of the proposed strategy comparing it with default NDN. Further, a sensitivity analysis is performed to understand the impact of the primary parameters of the proposed strategy: vicinity size, replication degree, and placement policy.

The second analysis studies how our strategy works on a larger scale, with multiple content objects and producers. The strategy parameters used in this scenario are based on the results obtained from the previous analysis. The proposed strategy executes locally in each producer without a global view of the network regarding the available resources and object placement. Throughout this scenario evaluation, we investigate whether the strategy can achieve good global results solely based on local decisions of each producer.

C. Topology & Workload

The model is implemented as a discrete simulation[1]. The topology is based on a random geometric graph, which covers a square area. The topology is composed of 33 routers and 56 links between them with a homogeneous latency of 10ms. The topology has an average shortest path of around 3.83 hops, its eccentricity (the largest shortest path between any two vertices) is 10 hops, and the clustering coefficient (how grouped are the nodes) is 48%. The network has 750 user devices uniformly distributed, averaging around 23 users per router [20]. The simulation is executed over 240 time steps, each representing around 1 minute.

The mobility model parameters describe how users move through the network. The user behavior is described by active-inactive cycles [21], and its movement is modeled using the Graph-based Random Waypoint Model [22]. User behavior is a sequence of successive sessions, characterized by periods of connectivity (i.e. active session) and inactivity (i.e. movement). The session duration follows the distribution measured in [21], in which 75% of the sessions are up to 7 minutes and 92% are shorter than an hour. It is roughly represented by a Pareto distribution with a shape of 0.38 and scale of 0.18. We assume that the longest session lasts at most 80 time steps (or minutes). Users move in a subset of between 2 and 9 routers (possible locations) uniformly distributed in the network [23]. They move between their possible locations following the graph paths at a random speed between 1 and 5 time steps per hop [21].

The workload of the simulation is based on User-Generated Content (UGC) [24]. Either one or multiple content objects are produced according to the evaluation scenario. In the case of multiple objects, 250 content objects are created, which results in 20% of producers in the network. All objects of the catalog have the same size of 10MB and are formed by a single piece of data. Their popularity distribution follows a Zipf distribution with $\alpha = 0.44$. This results in a long-tail of unpopular objects, characteristic of User-Generated Content [25]. In the context of UGC, consumers are concentrated based on their geo-location. In the evaluation, 75% of consumers of an object are grouped in an area whereas the rest is randomly distributed in the network [18]. The cache size of each device stores 1% (default value for NDN) of the maximum catalog size, and in-network caches have a lifespan of 3 time steps to simulate other traffic on the network. Besides the data objects, there is also non-cacheable control objects for the vicinity learning and content announcement, whose size are 10KB each.

The vicinity parameters are varied between 1 (default NDN and without the strategy) and 11 hops (complete view of the network). The maximum replication degree varies between 1 and 6 replicas. The strategy selects either a random or the best available device to push content to.

[1] The source code, input, and output files are available in https://github.com/mblehmann/noms-2015

D. Metrics

We evaluate three aspects of the strategy: content availability, consumer performance, and overhead. The following metrics are used to evaluate them:

- Content Hit Rate: the percentage of successful retrievals of content by the consumers. A higher hit rate indicates that the content has a higher availability.
- Content Retrieval Time: the average time for consumers to retrieve a content object from the closest provider. If no copies are found, the time of a failure request attempt (i.e. the time to send a request to the producer) is added to the overall retrieval time. This metric also includes provider announcement time.
- Number of Packets: the network overhead. A higher number of packets results in more traffic in the network, which in turn can cause congestions and delays. We measure the number of packets required for vicinity learning, pushing, and announcement separately.
- Data Volume: how much bandwidth is consumed. This metric complements previous one to describe the overhead caused by our strategy. A higher data volume can saturate the network bandwidth usage, and hence decreases its overall performance. The data volume takes into consideration both the packet size and the distance traveled.

V. Results

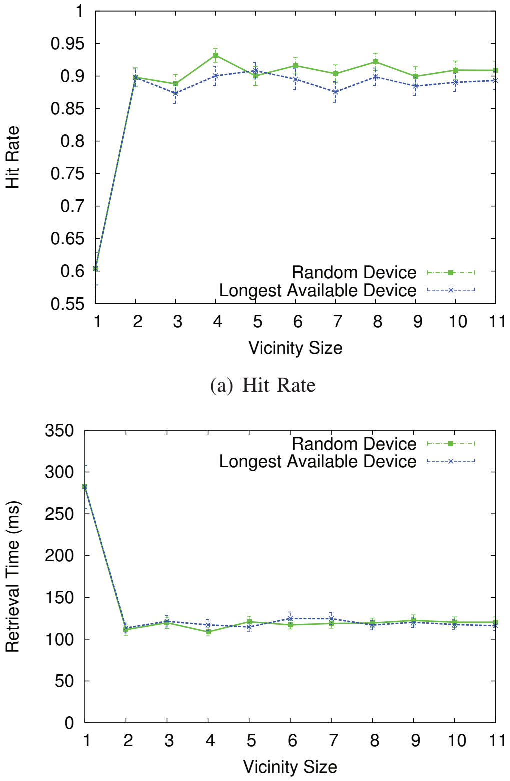
In this section, we first present and discuss the results for the single producer scenario, followed by the one with multiple mobile producers.

A. Single Producer Scenario

The goal of this analysis is first to quantify how well NDN supports producer mobility and compare it with the proposed strategy. In addition, it aims to assess the impact of the parameters vicinity size, placement policy, and replication degree on the strategy performance.

1) Vicinity Size: Figure 2 shows how the vicinity size and placement policies influence on content hit rate and retrieval time. When the vicinity size is set to one, the strategy is not executed because the producer knows only the router to which it is connected and does not know any device to push data to. Because of the topology properties, a vicinity of size eleven guarantees that the producer will have a full view of the network[2]. Regarding the placement policies, we evaluate two policies: random, and longest available device.

With the NDN's default settings, producer mobility causes the content to have a low availability. The content achieves a hit rate just over 60%, as shown in Figure 2(a). The consequence of the content unavailability also reflects on the consumer performance. When consumers are unable to retrieve the content, their request times out and they need to issue a new interest request. The results presented in Figure 2(b) show that consumers take about 282ms to retrieve an object due to this behavior.



(a) Hit Rate

(b) Retrieval Time

Fig. 2. Hit rate and retrieval time results varying the vicinity size

Expanding the vicinity to two generally allows the producer to discover other devices and push data to them. The results improve significantly compared to those with default NDN, as shown in Figure 2. By pushing one replica of the content object, the producer increases the content hit rate from 60% to around 89%. It also reflects on the consumer performance because it reduces the chance of not being able to find the desired object. As a matter of fact, the copy reduces the consumers retrieval time by 60% from 282ms down to 111ms.

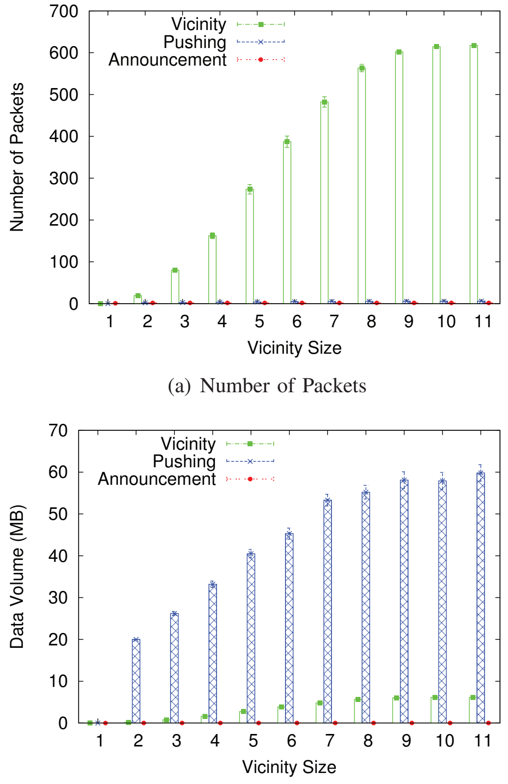
Figure 2 also shows the effects of the placement policy variation. The placement policy based on availability takes into consideration the remaining time of the devices' current session to find those that will stay the longest in the short term. Regardless of the vicinity size, the impact of the availability-based placement policy was negligible in comparison to the random. Analyzing the results with vicinities whose threshold lies between two and eleven, we see that the results vary between 88% and 93% for hit rate, and 108ms and 122ms for retrieval time without a clear trend.

The producer's extra knowledge did not improve the results due to the combination of poor information quality and the network dynamics. For simplicity, the producer learns only about the current session of another device. Besides, the fact that session durations are independent, it is not possible to predict

[2] Note, a producer with a vicinity of size nine or larger may have complete knowledge about the network due to its topological position.

the duration of future sessions of a given device. Hence, for the sake of the evaluation, the availability information of a device is not useful in the medium and long-term. So, in the remaining analysis, we present only the results for the random device policy. Note, however, that the proposed strategy allows a producer to leverage network properties and availability profile of devices to improve content placement.

The results of the network overhead caused by our strategy in terms of processing and bandwidth are shown in Figure 3. Figure 3(a) shows that a larger vicinity requires more packets to learn about other devices due to the higher number of known devices. The more packets created, the more processing a router requires to route and forward them, which may overload the network. Because of the network topology and its clustering coefficient of 48%, each expansion of the vicinity slightly increments the number of known devices. In comparison, a highly clustered topology would exponentially increase the number of vicinity packets sent in the first steps and then quickly converge to the maximum value. From a vicinity of size nine onwards a producer can have a complete view of the network, which explains why the number of packets for the vicinity learning converges at a vicinity of size nine. We also see that the number of packets for the pushing and announcement operations does not vary with different vicinity sizes.



(a) Number of Packets

(b) Data Volume

Fig. 3. Network overhead results varying the vicinity size

To have a better understanding of the actual network overhead, the data volume generated by the producer is also analyzed. The results are shown in Figure 3(b). The dominant aspect in the data volume is the pushing operation. The minimum value obtained for the data volume overhead is 20MB, which is achieved by pushing the 10MB-sized object to a device two hops away. As the vicinity size expands, the data overhead of pushing just one replica increases up to 60MB (six times the original object) because the producer may push a replica farther away, which consumes more resources. The data volume depends directly on the object size and the travel distance. Since the producer pushes the replica to a random device, the results reflect the average distance of the known devices in the vicinity. In the evaluation, the data volume growth is almost linear due to the network topology and uniform device distribution. If the producer uses a different placement policy (e.g. one that favors closer devices), the network has a different topology or devices have another distribution pattern the growth might not be linear. Learning about other devices in a small vicinity does not add significant data to the network, but as it is expanded the data volume overhead becomes a relevant factor. With a complete view of the network, the producer may generate an extra 6MB to learn about other devices.

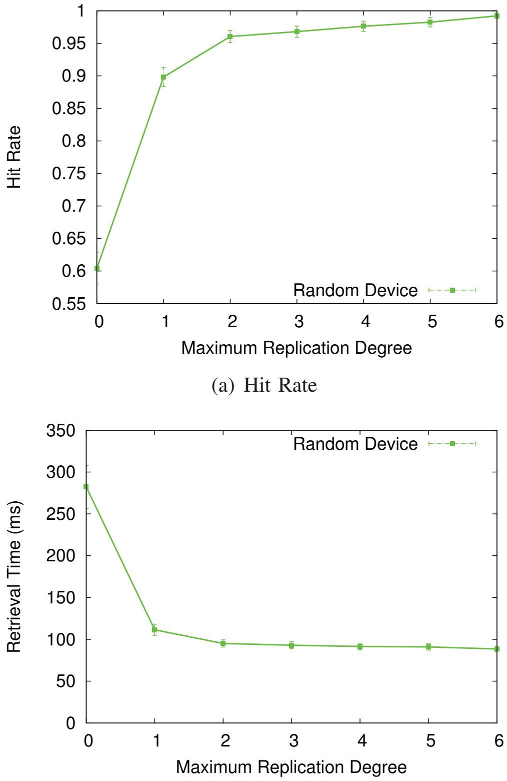
The results of combining the cost and benefits analysis show that a vicinity of size two is the best choice. The producer increases significantly the hit rate and reduces the consumer retrieval time by knowing at least one device to push a replica to. The small vicinity does not add significant overhead because it restricts the communication close to the producer, hence reducing the overall resource consumption of the network. In the remainder analysis, a vicinity size of two is assumed.

2) Replication Degree: Figure 4 shows the results for hit rate and retrieval time with a different number of maximum replicas pushed by the producer. As we can see in the results shown in Figure 4(a) when the producer pushes at least one copy it increases the content hit rate from 60% with default NDN to 89%. It reflects directly on the consumer performance presented in Figure 4(b), where the retrieval time is reduced by 60% from 282ms to 111ms.

When the producer pushes more than one copy to other devices, it improves further the results. With two copies, the content hit rate increases from 88% to around 96%. With three or more copies, the hit rate gain is marginal until it converges to 99% with six copies. This behavior is also seen in the consumer performance. Pushing two copies reduces the retrieval time by 16ms down to 95ms while six copies only reduce it to 88ms.

It is important to keep in mind that these results are affected by the availability of devices. In a network with lower availability, a producer would need to push more replicas to achieve similar results. On the other extreme, a producer in a highly available network would require fewer replicas to obtain the same level of results.

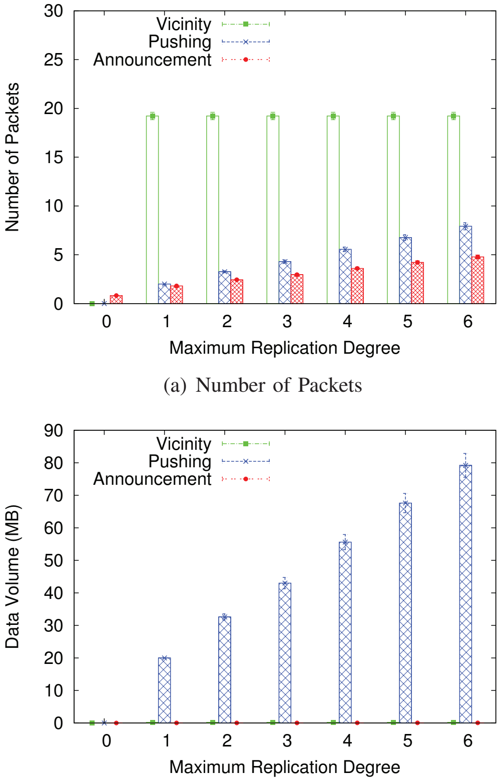
The results for the network overhead when a different number of replicas are being pushed is presented in Figure 5, for the number of packets and volume of data exchanged.



(a) Hit Rate

(b) Retrieval Time

Fig. 4. Hit rate and retrieval time results varying the replication degree



(a) Number of Packets

(b) Data Volume

Fig. 5. Network overhead results varying the replication degree

As expected, the more replicas are pushed, the higher the number of packets and volume of data exchanged. The vicinity learning is the dominant cost in terms of packets, as seen in Figure 5(a), despite not varying with different maximum replication degrees (because it is associated with vicinity). Also, when the producer pushes more replicas, it sends more data and creates more providers for the content object, hence increasing the number of announcements. Keep in mind that each announcement forces the routers to update their FIB tables and converge their routing information. As the number of providers increases due to more replicas, this overhead could become significantly detrimental to the network.

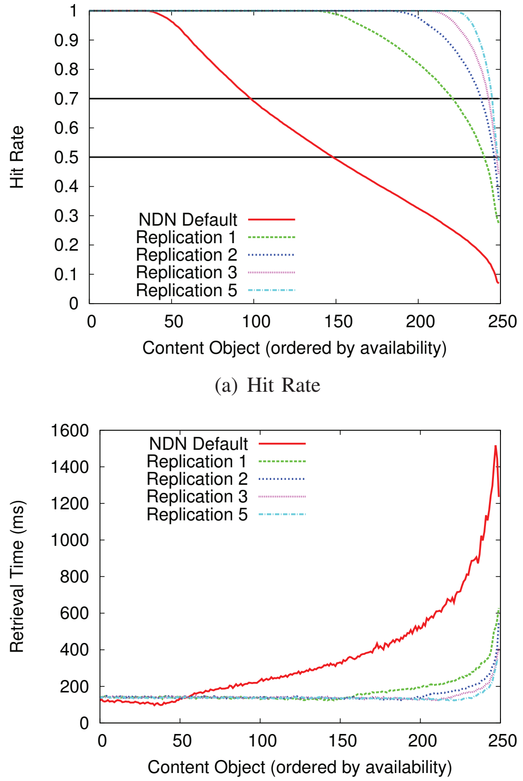
Regarding the data volume the result is straightforward: more replicas lead to more traffic in the network, as seen in Figure 5(b). The data volume added to the network varies from 20MB with one copy until almost 80MB with a maximum of six replicas. In this evaluation, the strategy may push a number of replicas up to the maximum replication degree according to the producer availability, as explained in Section III-C. Since the vicinity and the object size are fixed in 2 and 10MB respectively, one can infer the average number of replicas pushed. For instance, the 80MB data overhead, with the maximum replication degree of six, indicates that only four replicas were pushed in average because of the producer availability. The data volume generated by the vicinity or announcement operations are negligible in comparison to the pushing one.

The conclusion of the replication degree analysis is that pushing a single copy is enough to achieve most of the benefits. Pushing the second copy improves further the producer mobility support. However, pushing three or more copies only increases the results marginally. The network and producer overheads are proportional to the number of replicas pushed, which can be adjusted to find a balance between the benefits obtained and the overhead.

B. Multiple Producers Scenario

The second analysis focuses on evaluating the strategy in a scenario with multiple objects produced. Figure 6 shows the results for both hit rate and retrieval time. Throughout the analysis, the objects are classified in terms of availability according to their hit rate, compared to the average device availability of the network (60%). So, we define the following levels according to the hit rates: (a) **low**, for less than 50%; (b) **medium**, for hit rates between 50% and 70%, inclusive, and (c) **high**, for cases above 70%.

With default NDN support to mobility, the content objects have an average hit rate of 61%, as shown in Figure 6(a). The lack of producer mobility support by NDN causes a high variance on the hit rate of content objects, which is determined by the producer availability. Observe that the distribution between high, medium and low availability levels is approximately 40-20-40%, respectively. These results show that the NDN default mechanisms are not enough to provide at least a medium availability (network average availability) for nearly 40% of the objects in the network. Further, the poor availability reflects on the consumers' retrieval time, shown in Figure 6(b). The average time to retrieve a piece of content is 360ms. Despite around 24% of objects being quickly retrieved in between 100 and 150ms, more than 45% of the catalog takes more than 300ms (1.15 times the maximum RTT of the network).



(a) Hit Rate

(b) Retrieval Time

Fig. 6. Hit rate and retrieval time results with multiple content objects

When at least one replica is pushed for each produced content object, the overall hit rate improves significantly to up to 91%. The strategy provides effective support to producer mobility through the replicas. They allow more than 88% of objects (in the worst case) to have a high availability. Although the strategy fails to provide a high hit rate to every object in the system, it limits the low available objects to only 4% in the worst case. The higher hit rate causes consumers to retrieve the objects faster (evidenced by the average retrieval time of 170ms). The percentage of objects that are retrieved in under 150ms increases from 24% to 64%. The fastest retrieval time is around 125ms (25ms higher than with default NDN) due to the announcement convergence time caused by the extra providers in the network. Lastly, the strategy is able to reduce down to just under 6% the percentage of objects that takes more than 300ms to be retrieved.

Although an increase in the number of pushed replicas is beneficial to producers, the gains obtained by each new replica decreases exponentially. Despite pushing more copies marginally improves the results for the average producer as shown in Section V-A2, it allows more producers (especially those with low availability) to achieve better results and highly improve the availability of their content. If the network overhead caused by five replicas can be afforded, the average hit rate rises to 98%, and the number of objects with medium or low availability is reduced to only 2% (or 5 objects). The average retrieval time of objects decreases slightly to 140ms and only 3 objects take more than 300ms to be retrieved.

The network overhead has a similar trend as the one presented in the replication degree evaluation. The vicinity cost has a small variance between producers because of the device distribution in the network. In average, each producer knows about 18 and 21 devices in the vicinity. The pushing and announcement costs, on their turn, grow according to the number of replicas. The overhead is measured during four simulated hours over a network composed of 33 routers. Pushing one replica for each object adds 5860 packets (representing a volume of 5GB). When the strategy pushes up to five replicas, theses values increase to 7738 packets and 17.5GB. The highest overhead measured adds roughly 234 extra packets per router and 1.2MB/s extra traffic in the network.

These results demonstrate that the presented strategy adds homogeneous and equal support to every content object. Even though each producer pushes its objects only using local information, the proposed strategy allows over 88% of objects to remain available under producer mobility. This is a very positive result, considering producers compete for limited network resources without a global coordination, and their local decisions may impact negatively on each other.

VI. Conclusion

In this paper, we presented a novel content-centric strategy to support producer mobility in NDN. It leverages key features of NDN to overcome the loss of content availability induced by producer mobility. The key insight is to replicate proactively content by pushing it according to a placement policy. A detailed evaluation of the strategy and its parameters was performed to understand better the trade-offs associated with replicating objects in support to mobility.

We showed that NDN does not support producer mobility adequately, as expected, presenting poor average hit rate (only 60% of requests to objects can be satisfied). Then, we showed the proposed strategy can improve the hit rate by 46% and reduce the retrieval time by 60%. When there are multiple producers, the strategy reduces the percentage of low available content objects from 40% (using only NDN) down to 4%. This benefit is not for free, but the overheads to the network and producers are limited because the strategy restricts the scope and number of replicas through the vicinity size and replication degree.

Regarding future work, we will implement the strategy in NDN and address the re-attachment process. The evaluation of the strategy in a real environment will allow us to gain even more understanding of the strategy.

References

[1] C. V. Mobile, "Cisco visual networking index: Global mobile data traffic forecast update, 2014–2019," *San Jose, CA*, 2014.

[2] C. Perkins, "Ip mobility support for ipv4, revised," 2010.

[3] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in ipv6," 2011.

[4] 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. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[5] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, "A survey of mobility in information-centric networks," *Commun. ACM*, vol. 56, no. 12, pp. 90–98, Dec. 2013. [Online]. Available: http://doi.acm.org/10.1145/2500501

[6] T. Woo, H. Park, S. Jung, and T. Kwon, "Proactive neighbor pushing for enhancing provider mobility support in content-centric networking," in *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*, July 2014, pp. 158–163.

[7] J. Lee, S. Cho, and D. Kim, "Device mobility management in content-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 28–34, December 2012.

[8] R. Ying, L. Hongbin, G. Deyun, Z. Huachun, and Z. Hongke, "Lbma: A novel locator based mobility support approach in named data networking," *Communications, China*, vol. 11, no. 4, pp. 111–120, April 2014.

[9] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi, "Publisher mobility support in content centric networks," in *Information Networking (ICOIN), 2014 International Conference on*, Feb 2014, pp. 214–219.

[10] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting seamless mobility in named data networking," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 5854–5869.

[11] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "End-to-end mobility support in content centric networks," *International Journal of Communication Systems*, vol. 28, no. 6, pp. 1151–1167, 2015.

[12] J. Tang, H. Zhou, Y. Liu, H. Zhang, and D. Gao, "A source mobility management scheme in content-centric networking," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, Jan 2014, pp. 176–181.

[13] F. Hermans, E. Ngai, and P. Gunningberg, "Global source mobility in the content-centric networking architecture," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, ser. NoM '12. New York, NY, USA: ACM, 2012, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/2248361.2248366

[14] Y. Luo, J. Eymann, and A. Timm-Giel, "Mobility support for content centric networking," *Telecommunication Systems*, vol. 59, no. 2, pp. 271–288, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11235-014-9960-x

[15] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for ndn," in *Proceedings of the 1st International Conference on Information-centric Networking*, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 179–180. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660159

[16] X. Jiang, J. Bi, and Y. Wang, "What benefits does ndn have in supporting mobility," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, June 2014, pp. 1–6.

[17] L. Wang, O. Waltari, and J. Kangasharju, "Mobiccn: Mobility support with greedy routing in content-centric networks," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, Dec 2013, pp. 2069–2075.

[18] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook, and A. Yadav, "A global name service for a highly mobile internetwork," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 247–258. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626331

[19] J. Ge, S. Wang, Y. Wu, H. Tang, and Y. E, "Performance improvement for source mobility in named data networking based on global–local fib updates," *Peer-to-Peer Networking and Applications*, pp. 1–11, 2015. [Online]. Available: http://dx.doi.org/10.1007/s12083-015-0353-z

[20] Y. Zhang, "User mobility from the view of cellular data networks," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1348–1356.

[21] C. Tuduce and T. Gross, "A mobility model based on wlan traces and its validation," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, March 2005, pp. 664–674 vol. 1.

[22] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, April 2002, pp. 337–344.

[23] Z. Gao, A. Venkataramani, J. F. Kurose, and S. Heimlicher, "Towards a quantitative comparison of location-independent network architectures," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 259–270. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626333

[24] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298309

[25] F. Zhou, L. Zhang, E. Franco, A. Mislove, R. Revis, and R. Sundaram, "Webcloud: Recruiting social network users to assist in content distribution," in *Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on*, Aug 2012, pp. 10–19.