Ranged Name Retrieval: Design and Evaluation of a Flexible Data Retrieval Approach for ICN

Konstantinos Trichias, Lucia D'acunto, Floris Drijver, Bastiaan Wissingh

HAL Id: hal-02269735
https://inria.hal.science/hal-02269735

Submitted on 23 Aug 2019

Ranged Name Retrieval: Design and Evaluation of a Flexible Data Retrieval Approach for ICN

Konstantinos Trichias, Lucia D'Acunto, Floris Drijver, and Bastiaan Wissingh

TNO, The Hague, The Netherlands,
{kostas.trichias,lucia.dacunto,floris.drijver,bastiaan.wissingh}@tno.nl

Abstract. In Information-Centric Networking (ICN), applications request data based on the (exact) name of the data of interest. This poses a challenge for a growing number of applications, which do not know a priori the full name of a data object they seek, or do not necessarily need a specific data object, but are interested in a certain scope (e.g. the temperature in a particular geographical area). To address this challenge, we propose the more flexible and application-agnostic *Ranged-Name Retrieval* (RNR), which allows applications to define a range within each hierarchical component of the name they are requesting, obtaining in return one of the objects within that defined range. The performance evaluation of RNR indicates that, compared to the state of the art, it improves the bandwidth utilization efficiency 16-fold when retrieving data with unknown names, and even decreases the average end-to-end (E2E) delay for data delivery, while only adding 2% overhead for the processing of ranged-names.

1 Introduction

Information-Centric Networking (ICN) aims at evolving the Internet infrastructure from a connection-oriented towards a data-oriented approach, by directly addressing data objects themselves. The name of a data object is arguably a central concept in ICN. Popular ICN architectures, such as CCN [1] and NDN [2], have hierarchically designed names. The fact that any data object can be described by a hierarchical name has many benefits, from the ability to cache the data at intermediate nodes, to security and efficient routing and forwarding [2]. However, using the same hierarchical naming mechanism to also retrieve these data objects presents some limitations, primary among which is the fact that an application is required to know the exact name of the object it wants to obtain. In order to mitigate this issue, CCN supports the concept of "manifests", while NDN supports matching of Interests to Data packets with "partially known names", but both techniques suffer from limited flexibility and scalability, and can be unpractical for a number of emerging applications, where the space of potential named objects is very large and/or where a (ultra) low latency response is required. One example is that of an Internet-of-Things (IoT) application interested in sensor data related to a particular geographical area, e.g. a temperature

sensor between certain longitude/latitude boundaries. In this case, the application is not interested in the readings of a specific sensor within that area, and the space of potential data objects may possibly be much larger than the actual data objects available.

Easing the process of applications obtaining the data they need, through a potentially vast (and/or partially unknown) naming space within acceptable latency, becomes crucial for the success of ICN as future Internet architecture. We aim to address this challenge by increasing the flexibility of data retrieval for hierarchical names. In particular, we introduce a *Ranged-Name Retrieval* (RNR) approach, which allows applications to define a range within each hierarchical component of the name they are requesting, and obtain in return one of the objects within that defined range. An early version of RNR was demonstrated at the ACM ICN 2016 conference for a surveillance use case [3], as a proof of concept. In this paper, we expand on our previous work, and evaluate RNR's performance against legacy NDN operation via extensive large scale simulations in NS-3, as well as via small scale experiments with real nodes running a version of NDN with the RNR extension. Our experiments show that a 16-fold bandwidth utilization improvement and a decrease of the end-to-end (E2E) delay can be achieved with RNR, while only imposing a 2% increase in the computational load for realistic scenarios.

2 Related Work

The need for data retrieval mechanisms better suited to the requirements of applications has been tackled in previous work in ICN. In particular, some of the advantages of using a range in the name of the Interest to retrieve data object(s) in ICN has been advocated by both industry and academia proposals. At a first instance, the authors in [4] propose the use of ranges in Interest names for a vehicular network application, as a way for vehicles to request information about an area around specific geographical coordinates and/or between certain streets, and within certain time periods. While this is a first attempt at defining efficient name ranges, the proposed scheme is limited by its application-specific nature (i.e. ranges can only be interpreted by the nodes running the specific vehicular networking application) and by the assumption of a known name-space (known street names). Another application-specific proposal, [5], describes a method for efficiently requesting content that is segmented (e.g. a video file comprised by multiple temporal "chunks") in ICN, by means of so-called "block queries". Block queries are inserted in the header of the Interest packet and the ICN routers interpret a range as being the last part of the name, hence limiting the applicability of this solution to names with only one range and always positioned at the end of the name.

The work in [6] also addresses the issue of obtaining all the data within certain segment ranges, with multiple components of the name possibly carrying the range. To enable their solution, the authors of [6] introduce the concept of "pointers", network nodes associated to specific name components which keep
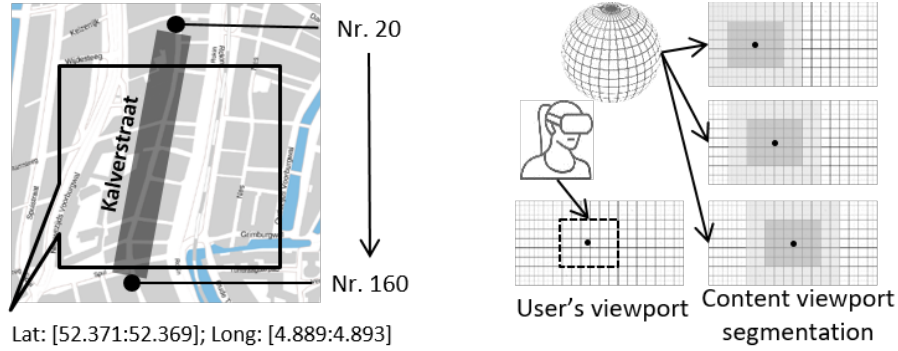
Fig. 1: Some use cases for RNR. IoT and V2X (left), VR/AR (right)

track of where all content within those specific names resides. However, the knowledge base of "pointers" needs to continuously be updated (which is difficult in highly dynamic scenarios), and pointers become single points of failure, decreasing the network's robustness.

What most of these solutions seem to have in common is that a) they propose application-specific naming schemes (only ICN nodes equipped with specific applications are able to understand / process the proposed naming scheme) and/or b) assume the existence of a finite and well defined space of named objects for which the request will be made. Our solution builds on top of the generic lines proposed in the literature and we expand the functionality of such a ranged-name retrieval mechanism to operate in an application-agnostic manner and to account for vast (unknown) name spaces, while still relying on the distributed routing mechanisms of ICN architectures.

3   RNR: the Concept

3.1   Use Cases

In the era of Big Data and Internet of Things, an exploding number of devices is producing and requesting data and in a plurality of scenarios the exact name of an object is not known in advance: either because of the mobility of data producers in the network, or because of the dynamicity of certain applications (e.g. content on-demand, or within specific time frames), or simply because the exact number, type and name of all nodes (IoT devices, sensors, vehicles, etc.) deployed within a certain area, is not always known. In many of these cases, though, applications may know an approximation or part of the name, and seek to retrieve a data object within one request - either because of latency requirements, or because the specific data object to be retrieved is not important, as long as it fits within a certain "scope". The design of RNR stems from these needs. A few use cases where RNR is beneficial are outlined below.

IoT/monitoring: Data is typically produced by (static) sensors placed in the area of interest (e.g. urban/environmental monitoring, use during festivals). In such cases, the name space of possible data objects may greatly exceed the

number of actual data objects and many applications may not even be interested in the output of a specific sensor, as long as they obtain one sensor reading within a certain range. Ranges may appear in several components of the name (e.g. one indicating latitude and the other longitude), see for example Figure 1 (left).

Vehicle-to-Everything (V2X): In this use case the sensors themselves are mobile (on the vehicles), and the data they produce may change name over time. For example, an application that may want to sample the speed of a random vehicle between street numbers 100 and 160 of the Kalverstraat in Amsterdam (Figure 1, left), may receive a response by any vehicle situated in the area, and at different times that information may be provided by different vehicles fitting the criteria of the request.

Immersive media (VR/AR): In Virtual or Augmented Reality (VR / AR) applications, content is commonly spatially segmented, with each spatial segment catering for a different, wide, viewport [7,8] (Figure 1, right). Viewports for immersive media are usually centered a few degrees apart (e.g., with steps of 30-60 degrees [7,9]), and are often partially overlapping [10]. As users move their head in the virtual world, the content matching their new orientation needs to be streamed to them. This adaptation needs to be very quick because of the extremely low latency requirements of immersive media use cases. If an application does not know the viewport segmentation used by the video source (e.g. when the content is produced *live*), it might be convenient for the application to specify a range of viewports that will cover the user's orientation (e.g. 30° clockwise from the north, within a 30° range: */WaltDisney/JungleBook/orientation/[15:45]*) and let the network fetch the closest viewport matching this orientation, in order to minimize latency.

3.2 A More Flexible data Retrieval Approach

To cater for the different needs and use cases as described in the previous section, we have designed and built *Ranged-Name Retrieval* (RNR) as an extension to NDN. Embedding RNR at the network layer (i.e. extending the core of NDN) also makes our approach application-agnostic, since nodes on the request path can process ranged-names without needing to invoke (and have installed) a specific application to deal with it. Our approach remains compliant to the "1 Interest packet, 1 Data packet" principle: even if an Interest with a ranged-name may be potentially satisfied by multiple data objects, only one will be returned to the requester. RNR comes with two inherent advantages. On one hand, it facilitates the retrieval of data with unknown names by applications, while on the other hand its application-agnostic design also has the potential of decreasing the network traffic, since it may be used by all applications, independently of their nature, and since ICN nodes in the network understand the range and will be able to fetch the closest data matching the range. Even in cases where the application knows the exact name of a data packet, the use of an Interest with a range may result in reduced end-to-end latency for data delivery and traffic per link, since a data object that also satisfies the application and is different than the originally intended one, may be fetched by an ICN node in closer vicinity.

4   Implementation and Experimental Design

4.1   RNR extensions to NDN

The extensions for supporting the RNR concept, implemented on NDN version 0.4.1, are described below.

A New Name Component in Interest Names: We have introduced a new type of Name Component, a Name-Range Component (NRC) [3], that includes special syntax to indicate ranges within Interest packets. The NRC is implemented as a new Type-Length-Value (TLV) type, so that NDN routers can easily distinguish them from regular name components. Only names in Interest packets are allowed to have a NRC in them.

Processing incoming Interests: The NDN network forwarder has been extended to be able to process incoming Interests bearing NRCs name components. When a NDN node receives a new Interest, it adds it to the Pending Interest Table (PIT) and checks whether this Interest is already pending. Upon insertion in the PIT, if the Interest contains a NRC, it is flagged and its position in the PIT is saved in an additional "NRC index table" for quicker lookups on the incoming data path. If the Interest is not pending, the node performs a lookup in the Content Store (CS). When the prefix of the Interest contains a NRC, the node searches the CS for possible entries within the range of the prefix. If a match is found, the matching entry is returned, and the corresponding PIT entry is removed. If no match is found, the node looks in the Forwarding Interest Base (FIB) for possible entries within the range of the prefix. If a matching entry within range is found, it is returned and used to forward the Interest containing the NRC.

Processing incoming Data packets: The NDN network forwarder has also been extended in order to match incoming Data packets with PIT entries containing NRCs name components. Specifically, upon receiving an incoming Data packet, a RNR node checks the PIT for entries that match the Name of the Data packet. When searching the PIT, the node first checks for standard entries, and then proceeds to check for NRC entries, using the NRC index table.

4.2   Simulator Integration and Evaluation Scenarios

The network level performance of RNR is evaluated using version 2.3 of the `ndnSIM` module [11], available for the NS-3 simulator. For all nodes in NS-3 simulations (whether a consumer or a producer) caching was disabled for reasons of simplicity. In fact, for this initial evaluation, our goal is to provide a good understanding of how RNR helps with fetching content with unknown name components from the original, spatially distributed producers themselves. The introduction of caching, which is part of our future work, is not expected to alter the outcome significantly, since any node that would cache a data object could also be seen as a new producer of that same data object. The consumer nodes transmit NRC Interests following a realistic Zipf-Mandelbrot distribution.

Network level evaluation To get a full picture of RNR's capabilities, we evaluated its performance in a large scale network, and performed a sensitivity study of the various network parameters that can affect its performance. For this purpose, we have used the NS-3 simulator with the `ndnSIM` extension and the RNR upgrades.

The performance of RNR under varying network conditions is bench marked against baseline scenarios where legacy NDN is used, applying three major Key Performance Indicators (KPIs):

- *average end-to-end delay (E2E delay)*, i.e. from the generation of the first Interest to the delivery of the corresponding Data packet;
- *bandwidth utilization*, i.e. traffic on the wire in MB (differentiating between Interest and Data traffic);
- *Interest success ratio* i.e. percentage of Interests that returned a Data packet.

Since there is no official approach in NDN on requesting data with unknown names, an assumption has been made regarding the functionality of the baseline scenarios using legacy NDN. A traditional approach would be having the consumer sending out sequential legacy NDN Interests for the names in the specific range under consideration, until a name is used that corresponds to actual existing data in the network, at which case the correct data object will be returned. As an example, in the case of a temperature reading from an unknown street number in Amsterdam in the range [20:40], the RNR scenario would use an Interest with a name such as */NL/Amsterdam/Kalverstraat/streetnr/[20:40]/temperature*, while a baseline scenario would start transmitting sequential Interests with names such as */NL/Amsterdam/Kalverstraat/streetnr/20/temperature*, */NL/Amsterdam/Kalverstraat/streetnr/21/temperature*,..., etc. until it transmitted an Interest with an existing name, and the corresponding Data packet is returned. Since transmitting these Interests in a sequential fashion, and waiting for a timeout of the previous one before sending the next one, would result in unrealistic delays that no application could tolerate, we have designed the legacy NDN baseline operation to flood all individual Interests at the same time, thus keeping delays at a reasonable level. In the aforementioned example, that would mean that in the legacy NDN case the consumer would flood the network with 21 Interests at the same time, and hope that at least one of them finds a match and returns a Data packet. The *Traffic Mix (% ranged requests)* parameter (Table 1) indicates the percentage of requests where a range is used, which are implemented by means of one NRC interest for the case of the RNR implementation, and by means of the flooding approach for the case of the legacy NDN implementation. The *Range Size* parameter indicates, for the legacy NDN implementation, the amount of parallel Interests that are flooded in the network.

For both the legacy and RNR versions of NDN, the *"best route"* forwarding strategy is employed. Furthermore, a Zipf distribution is used for content generation and a realistic *Barabási–Albert* topology is used for the network layout, with a varying number of nodes to cover the entire range from small-scale to large-scale NDN networks, while the simulation duration was network-size

Table 1: RNR simulation parameter settings

Topology	Barabási–Albert
Data Availability	1%, **<u>10%</u>**, 25%, 50%, 75%, 100%
Consumers / Producers	90%/10%, **75%/25%**, 50%/50%, 25%/75%
Traffic Mix (% ranged requests)	0%, **<u>10%</u>**, 20%, ..., 100%
Range Size	5, 10, **<u>25</u>**, 50, 100
Number of Nodes	25, 100, **<u>500</u>**, 1000, 2000
Avg. Data Packet Size	10 B, 100 B, 500 B, **<u>1 KB</u>**, 1.5 KB

dependent with a minimum requirement for 10^5 transmitted Interests per scenario. All nodes in the network were either a consumer generating Interests or a producer generating data (content randomly distributed over the available producers), while at the same time all of them could act as forwarders. Different combinations of consumer / producer ratios were evaluated as can be seen in Table 1. Finally, different Data Packet Sizes were implemented in order to cater for applications with different needs, with a maximum packet size of 1.5 KB to simulate the maximum MAC PDU size over Ethernet.

The *Data Availability* parameter indicates the actual percentage of existing data objects in relevance to the possible name space for a name component. As can be understood, Data Availability is far less than 100% in most scenarios, resulting in a lot of failed (timed-out) Interests in both legacy and RNR implementations of NDN, when looking for data with unknown names. For both implementations a "penalty" of 1 second (equal to the timeout timer) was imposed to the E2E delay for every failed Interest.

For the purposes of this evaluation, we use the "immersive media" use case described in section 3 where the video content is spatially segmented into specific viewport orientations. For simplicity, we assume that the content only offers a horizontal degree of freedom (i.e. users can look around 360°, but cannot look up or down). In this case, the possible name space for the component *"orientation"* is [0-359], corresponding to degrees starting from the North. In an exemplary case where the entire 360° video would be covered by 24 viewports, equally spaced (each 15° apart), the Data Availability for such a case would be 24/360 = 6.6%. An overview of all the settings is presented in Table 1.

5   Performance Evaluation

In this section, we present the outcome of the large-scale NS-3 simulations and elaborate on the gained insights. During the evaluation while the effect of one parameter on the performance of RNR was under investigation, some default values have been selected for the rest of the parameters in Table 1, for the experiments on the simulator. The selected default values per parameter represent a realistic scenario and are indicated with bold and underlined text in Table 1.

5.1   Network wide evaluation

We have evaluated the network-wide performance of RNR NDN based on our NS-3 implementation (see Section 4.2), using the parameter values from Table 1.
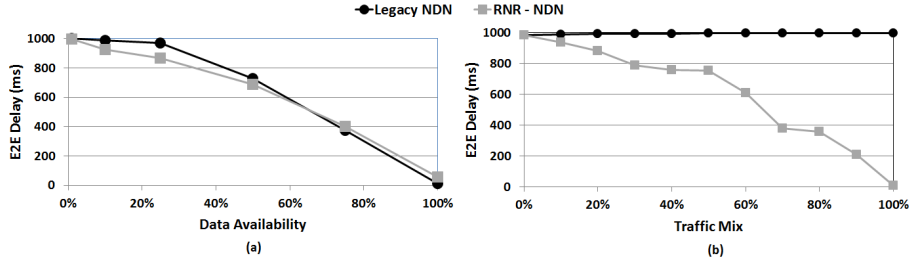
Fig. 2: Average End-to-End (E2E) Delay versus (a) Data Availability and (b) Traffic Mix (% ranged names)

Figure 2 depicts the average E2E delay per Interest-Data packet pair, for various Data Availability values (a), and for different ranged request percentages within the Traffic Mix (b). In both cases there is an upper limit of the E2E delay at 1 second (1000 ms) which is caused by the high "miss" ratio of Interests and the resulting enforcement of the time-out penalty (as described in Section 4.2). When Data Availability is very small (Figure 2 (a), left side), or when only legacy NDN Interests are used to request data with unknown names (Figure 2 (b)), most Interests do not find an appropriate Data packet to return and hence the time-out penalty dominates the E2E delay.

From Figure 2 (a) we can observe that the average E2E delay drops significantly for higher Data Availability values (down to about 11 ms for 100% Data Availability). In these cases most Interests return a Data packet and there are almost no time-out penalties enforced, depicting the pure latency of the network. The fact that legacy NDN and RNR NDN (using the default values for RNR settings) seem to provide similar delays is attributed to the implementation of flooding for the legacy NDN as a way to request data with unknown names (see Section 4.2). Since legacy NDN 'floods' the network with multiple Interests at the same time, it is able to attain a similar latency as RNR NDN does by using one Interest to achieve the same 'search'. This results in a significant difference in used bandwidth, as depicted in Figure 3.

From Figure 2 (b) we can conclude that RNR NDN is a much more efficient way of looking for data with unknown names, since the more ranged requests are used, the lower the average E2E delay. The entire range of traffic mix for the legacy NDN is dominated by the 1 second penalty (due to the default 10% Data Availability), since with higher traffic mix, more Interests are 'flooded' into the network and even though more Data packets are returned, the ratio of successful-to-failed Interests remains very low. This is not the case for the RNR NDN implementation which can achieve a search over the same range of names with only one Interest. As a result when the Traffic Mix is close to 100% (all Interests are ranged requests) most Interests are successful and there are very few time-out penalties imposed which greatly improves the performance. In a realistic scenario we expect the Traffic Mix to match the percentage of data requests with unknown names, i.e. if 10% of the requested data have unknown names then approximately 10% of NRC Interests should be used.
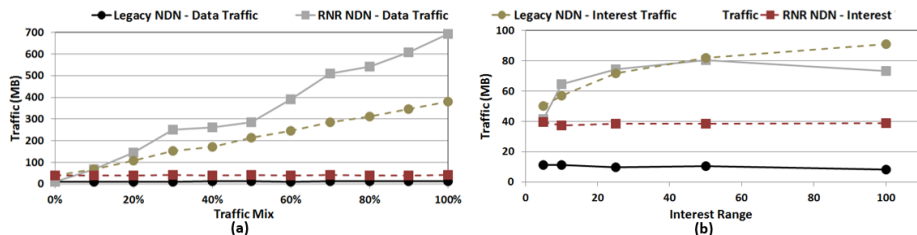
Fig. 3: Interest and Data packet bandwidth utilization vs (a) Traffic Mix, (b) Interest Range

Next to the average E2E delay, we have looked at the bandwidth used, in terms of MB on the wire, to achieve the same result (i.e. obtain a data object in the range) for both legacy and RNR NDN. This KPI enables us to evaluate the overall efficiency of the two solutions.

Figure 3 (a) depicts the bandwidth utilization per solution versus the Traffic Mix. This figure provides many interesting insights, foremost of which is that the bulk of the traffic in the legacy NDN case originates from the Interests while in the RNR NDN case it originates from the Data packets returned. Although the Data packets are much larger in size (default value of 1KB) compared to the 58 Bytes of the Interest packets, in the case of legacy NDN the Interest traffic dominates the bandwidth utilization, as a result of the fact that a large number of Interest packets is transmitted while only a low number of Data packets is returned. The RNR NDN on the other hand uses a fixed amount of bandwidth for Interest traffic, while the more ranged requests that are used the better the success ratio (transmitted Interests vs Delivered Data packets) and hence more data is delivered. It is also interesting to note that the legacy NDN Interest traffic load increases with increasing range utilization since more and more Interests need to be flooded, while at the same time the increase in returned Data packets is minimal (from about 10 MB for 0% ranged requests to about 12 MB for 100% ranged requests) and remains even under the RNR NDN Interest traffic. During our experiments we also observed that for 100% ranged requests the success ratio is also 100% which serves as a good sanity check for the scenario under consideration. With the default values used here (Data Availability = 10%, NRC range = 25, name space = [0-359]) it is expected that there are about 360*10% = 36 individual data objects (VR viewports). With an approximate equal spacing among them it is expected that each viewport will be about 360 / 36 = 10° apart. Since the range used has a size of 25°, it is then expected that when 100% of the Interests are ranged requests there will be at least one (likely two) data object within this 25° range that satisfies each Interest and hence all Interests are successful, bringing the success ratio to 100%. The above presented analysis indicates that RNR clearly outperforms legacy NDN and operates much more efficiently in scenarios where data objects with unknown names need to be requested.

Figure 3 (b) depicts the bandwidth utilization of Interest and Data traffic for various values of Range Sizes. As the Range Size increases, the Interest traffic for legacy NDN also increases while the Interest traffic for RNR NDN remains
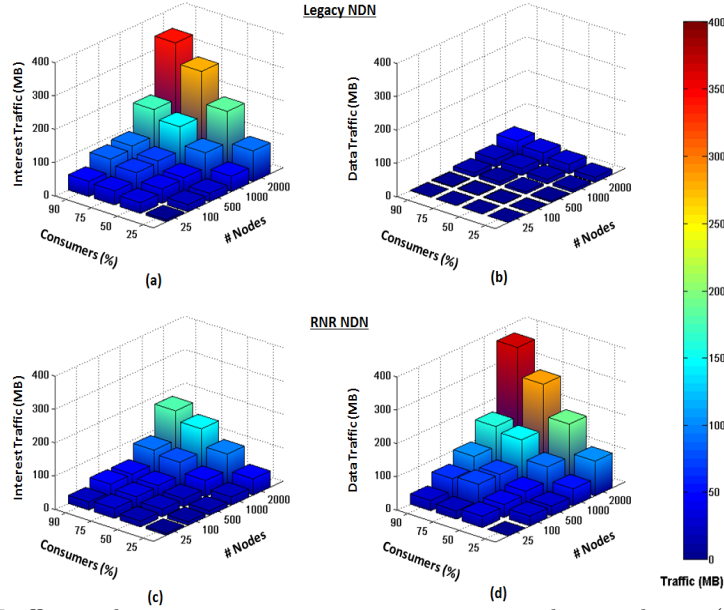
Fig. 4: Traffic on the wire versus consumer percentage and network size (number of nodes) originating from (a) Legacy NDN Interests, (b) Legacy NDN Data packets, (c) RNR NDN Interests and (d) RNR NDN Data packets

the same. This can be explained as follows: for the legacy case an increase in the Range Size from 25° to 50° means a doubling of Interest traffic since for every data object search a consumer would have to 'flood' 50 sequential legacy NDN Interests instead of 25 resulting in 50*58 = 2900 Bytes on the wire per search instead of 25*58 = 1450 Bytes per search. This phenomenon can be clearly seen in Figure 3 (b). In the case of RNR, however, an increase in the Range Size from 25° to 50° would simply mean an update of the relevant name component of the Interest from e.g. /[50-75] to e.g. /[50-100], thus keeping the Interest traffic to a minimum of 58-60 Bytes transmitted per search. As a result, the Interest traffic of RNR remains independent of the range used which is a great advantage when the possible name-space of an unknown data object is very large. It is also interesting to note that even though the data traffic of RNR increases at first with increasing Range Size, it seems to reach a maximum (truncation point) after some value of the range (around a Range Size of 25°) and after that an increase in the range has no effect on the number of returned packets. This effect is attributed to the fact that with a Data Availability of 10%, a Range Size of 25° is enough to guarantee that every NRC Interest will find at least one data object. A further increase of the range beyond that point will not result in any extra Data packets delivered.

As a final step, we have investigated the effect of a varying network size and a varying mix of consumers and producers in the network on the RNR implementation, in order to establish the trade-offs of our solution against different types of networks. Figure 4 is a 3D bar graph depicting the bandwidth utilization for a variety of network sizes (number of nodes) and consumer / producer mixes

showcasing (a) the legacy NDN Interest traffic, (b) the legacy NDN data traffic, (c) the RNR Interest traffic and (d) the RNR data traffic. It is evident from Figure 4 that both Interest and Data traffics grow with increasing number of nodes and increasing number of consumers since more consumers means more Interest packets and more Data packets as a response. The corresponding decrease of producers (result of the increase of consumers) does not negatively affect the data traffic in this case, since the number of contents actually remains the same. The only difference is that the same content is now more scarcely placed within the network, evenly distributed among the decreased number of producers. This scarce distribution of content may affect other KPIs such as E2E delay (content available at a producer further away) but this effect can also be counter-balanced by the inherent NDN caching mechanism. We observe that both the legacy and RNR NDN average E2E delays are unaffected by the change in Consumer / Producer ratio for these experiments due to the fact that in both cases the E2E delay is dominated by the time-out penalties enforced. The delay is 10% shorter for the RNR case due to the higher success ratio but in both cases there are a lot of non-ranged requests (since the default value of 10% Traffic Mix is used) which result in several time-out penalties.

The results presented in Figure 4 also indicate that the RNR solution scales very well with network size as well as with content size and/or amount of content producers, which makes it highly adaptable and suitable for a wide range of networks and applications. The legacy NDN on the other hand shows scalability issues which would make it unsuitable, as is, to be implemented in large scale networks and/or networks with a high percentage of unknown names (e.g. massive IoT sensor networks).

6   Conclusions and Future Work

In this paper we have presented the design, implementation and performance evaluation of *Ranged-Name Retrieval*, an extension of NDN with support for names with range indications to enable the consumers to request data with partially unknown names. The direct implementation of RNR within the core of the NDN architecture enables it to operate at wire speed and on any NDN node, making it application-agnostic so that data from different applications can be cross-utilized.

Our extensive performance evaluation indicates that RNR manages to address a key issue where legacy NDN is highly inefficient (retrieval of data with unknown names), by significantly improving the network performance (up to 16-fold bandwidth utilization improvement), while at the same time exhibiting a high degree of scalability and imposing minimal overhead (2% additional computation load for realistic scenarios).

The fact that RNR can accommodate varying Range Sizes without increasing the overhead on the nodes, and support multiple ranges within the same Interest (for different name components), makes it highly scalable and adaptable to current of future networks and applications.

Our future work is planned along two main directions, (a) the further development of the RNR scheme and (b) the more detailed testing, evaluation and benchmarking of RNR. As a first step we will extend RNR to support multiple ranges within a single name, each range specifying a search space for a specific component of the name. In this way data objects with inherent tuples in their name, such as latitude and longitude, can be requested while specifying an individual search space for each attribute. New ways to increase the functionality of the range component will also be researched, e.g. combining ranges and data manipulation functions (i.e. to request data aggregates like average, minimum, etc.).

In terms of performance evaluation, a larger variety of scenarios will be tested in the simulator and more extensive KPIs will be used, such as hop count, cache hit ratio and latency, which will allow for a deeper understanding of RNR under different network conditions. Finally, a first version of RNR will be implemented in the TNO IoT testbed, currently under construction, which will consist of dozens of IoT devices / sensors spreading over multiple rooms and floors and running both IP and NDN protocol stack. Experimenting on such a real-life IoT testbed will give us new insights regarding the performance of RNR and the potential issues that may arise or be solved by it, in a true environment.

References

1. V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *ACM CoNEXT*, 2009.
2. L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, 2014.
3. B. Wissingh and L. D'Acunto, "Extending named-data-networking with support for names with range indications," *ACM ICN*, 2016.
4. L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, "Data naming in vehicle-to-vehicle communications," in *IEEE INFOCOM WKSHPS*, 2012.
5. L. Samsung Electronics Co., "Terminal device based on content name, and method for routing based on content name," *US 20140181140 A1*, 2014.
6. M. Varvello, J. Esteban, M. Smith, L. Greenwald, Y. Guo, and M. Schurgot, "Range queries in manet: Can information-centric networking help?" in *IEEE ICCCN*, 2013.
7. X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," *arXiv preprint arXiv:1609.08042*, 2016.
8. L. D'Acunto, J. van den Berg, E. Thomas, and O. Niamut, "Using mpeg dash srd for zoomable and navigable video," in *ACM MMSys*, 2016.
9. *Next Generation Video Encoding Techniques for 360 Video and VR*, https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/.
10. O. Niamut, G. Thomas, E. Thomas, R. van Brandenburg, L. DAcunto, and R. Gregory-Clarke, "Immersive live event experiences-interactive uhdtv on mobile devices," British Broadcasting Corporation, Tech. Rep., 2014.
11. S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim 2: An updated ndn simulator for ns-3," Internet Research Lab, Tech. Rep., 2016.