

# Distributed Rendezvous placement for producer mobility support in NDN-IoT

Mohammed Bellaj<sup>1\*</sup>, Najib Naja<sup>1</sup>, and Abdellah Jamali<sup>2</sup>

<sup>1</sup>National Institute of Posts and Telecommunications Rabat, Morocco

<sup>2</sup>Ecole Nationale des Sciences Appliquées de Berrechid, University of Hassan I Settat, Morocco

**Abstract.** Named Data Networking is a new networking architecture proposed as a potential design for the next generation of networks. However, supporting the mobility of producers in NDN-based IoT systems remains an open challenge for researchers to address. To tackle this problem, the NDN research group introduced a solution called KITE [1]. Since this solution is an initial design, some issues to be resolved remain, including the rendezvous placement selection. The location of the rendezvous servers can significantly reduce the path stretch efficiently. This paper focuses on improving the KITE solution by addressing two specific issues. First, we aim to determine the optimal number of rendezvous servers to use in the network. Second, we seek to identify the best locations for the rendezvous servers to minimize the path stretch efficiently. We propose an evaluation metric called average hop count based on network centrality measures to select the number and the location for the rendezvous. We evaluate the Average hop count under different network centrality measures using random scale-free network topology. Our results show that the betweenness centrality measure performs better than other measures, and we, therefore, suggest using it as a metric for selecting the number of rendezvous servers and their locations.

## 1 Introduction

As the number of connected devices, known as "things," grows and becomes more diverse, the IoT is expected to involve billions of devices capable of collecting and sharing data. These devices have very different capabilities, requirements, and operating environments and may be resource-constrained, leading to the exchange of massive volumes of small data packets and a diverse range of traffic types. Given these challenges, there is concern that the current Internet architecture, based on the Internet Protocol (IP), may not be able to support the evolving needs of the IoT. To address these challenges, researchers are exploring the use of NDN, a cutting-edge solution that aims to improve data dissemination efficiency in the IoT ecosystem.

Named Data Networks (NDN) [2] is a recently developed networking architecture being proposed as a potential design for the next generation of networks. Unlike traditional networking architectures, which rely on location-based addresses (such as IP addresses),

---

\* Corresponding author: [bellaj.mohammed@inpt.ac.ma](mailto:bellaj.mohammed@inpt.ac.ma)

NDN utilizes a content-based approach in which data is requested by its name. This receiver-driven mechanism allows for more flexible and efficient data retrieval, as well as enhanced security and privacy compared to traditional approaches. NDN has been designed to be scalable, secure, and efficient for a wide range of applications, thanks to features such as in-network caching, name-based routing, and multicast. These features allow NDN to handle large amounts of data, making it a promising choice for the next generation of networks.

The NDN philosophy has the potential to offer many benefits in IoT environments, such as improved data distribution and transmission. However, some challenges have emerged that could hinder the adoption and deployment of NDN in IoT systems. One of these challenges is mobility, which refers to the ability of devices to move around and change their locations within a network. In NDN, data objects are requested by name rather than by location, which means that devices in the network need to be able to find the data objects that they are looking for based on their names, rather than their locations in the network. This can be a challenge in IoT environments, where the devices may be dispersed over a wide area and may not have a fixed location. When devices change their location or network connectivity, they may become unreachable, which can be a serious issue in IoT scenarios where mobility frequency may be high (e.g., in eHealth applications or intelligent transport systems) and where continuity of data is important for real-time sessions.

To address the challenges of supporting mobility in NDN, researchers have proposed various approaches. One such approach is a Tracing-Based mobility support approach [3], such as the KITE solution proposed by the NDN research group. This initial design uses a distributed rendezvous system with multiple rendezvous servers that work together to allow mobile producers to maintain connectivity as they move and improve the system's scalability concerning the number of mobile producers. KITE allows mobile producers efficiently distribute and transmit data by creating a path between the mobile producer and a reachable rendezvous server using Interest-Data trace packet exchanges. In this paper, we focus on improving the KITE solution by addressing two specific issues. First, we aim to determine the optimal number of rendezvous servers to use in the network. Second, we seek to identify the best locations for the rendezvous servers to minimize the path stretch efficiently. We propose an evaluation metric called average hop count based on network centrality measures to select the location for the rendezvous. We evaluate the Average hop count under different network centrality measures in a random scale-free network topology

The rest of this paper is organized as follows. In Section II, we provide a brief background on NDN and the mobility issue in NDN-IoT, with more details about KITE solution. Section III describes the related work on network centrality. Section IV presents the simulations we conducted and the main results. Finally, in Section V, we conclude the paper.

## **2 Background**

### **2.1 Overview of named data networking**

The NDN architecture adopts a receiver-driven communication [4] model in which the consumer initiates the communication by sending an interest packet to request data by name, while the producer responds by sending the requested data. In addition, in NDN architecture, each node in the network maintains three main data structures to support efficient and reliable data delivery. These data structures include: 1) the Pending Interest Table (PIT), used to track

interest packets that have been sent by the node; when a data packet is received by a node, this latter checks its PIT to see if there is an outstanding matching interest for the data. If a match is found, the node sends the data back to the consumer along the reverse path taken by the interest packet; 2) the Forwarding Information Base (FIB), The FIB stores the next hop for each prefix and is used to determine the path that an interest packet should take through the network to reach the data it is requesting based on the routing strategy being used by the node; 3) the Content Store (CS) for storing locally cached data. When a node receives a data packet, it is stored in the CS for future retrieval.

Figure 1 outlines the Interest and Data forwarding pipeline in a typical NDN node.

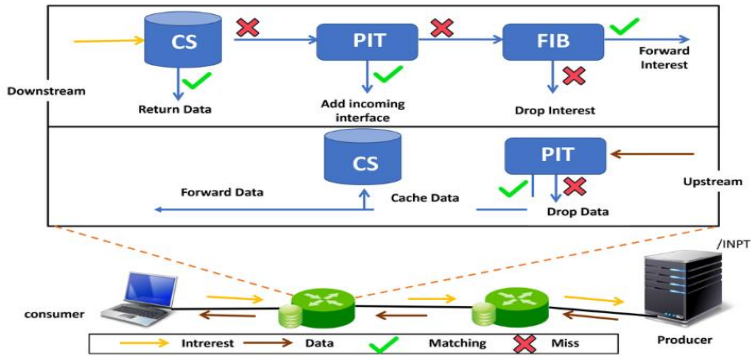


Fig.1. NDN forwarding pipeline.

## 2.2 Mobility in NDN-IoT networks

In the NDN network, mobility refers to the ability of devices to move within the network while maintaining connectivity and communicating with other devices. Mobility is critical in IoT, where devices may be mobile and need to maintain connectivity as they move. There are two main types of mobility in NDN: consumer mobility and producer mobility. In the NDN architecture, which adopts a receiver-driven communication model, consumer mobility is supported by design, as the consumer needs to resend unsatisfied requests from the new location. Producer mobility refers to the movement of devices that produce and transmit data packets in response to interest packets. In NDN, data objects are requested by name rather than by location, which means that mobile devices may become unreachable during and after a handoff when they change their location or network connectivity. To ensure the efficient and reliable operation of mobile devices within the Named Data Networking (NDN) architecture, researchers have proposed various solutions, including the Kite solution that this work is based on.

## 2.3 Mobility in NDN-IoT networks

To address the issue of producer mobility in Named Data Networking networks, the NDN research group has proposed a solution called KITE with Distributed rendezvous. This tracing-based solution involves using multiple rendezvous servers that cooperate to maintain connectivity for mobile producers as they move within the network. These rendezvous act as intermediaries between mobile producers and consumers. A mobile producer (MP) in KITE establishes soft-state paths to one or more rendezvous (RV) servers through an Interest-Data packet exchange with a special tag. Indeed, when the mobile producer connects to the

network, it sends a trace Interest to the closest RV servers. Upon receiving the trace Interest, the RV servers verify it and send a signed tracing data packet (TD) back to the mobile producer if the verification is successful. This TD packet establishes a trace FIB entry between the RV servers and mobile producers at the forwarders. Hence, allowing the MP to move within the network without updating its locator information. When a consumer requests data, it sends an Interest under the data name with prefix using the same rendezvous servers prefix announced in routing announcements, and the Interest packet will be forwarded to RV servers in the network toward the producer. Using multiple rendezvous servers in KITE solution offers several benefits. For instance, in terms of Reliability, it ensures less risk of a single point of failure. If one rendezvous server goes offline, the others can still handle the traffic. Furthermore, the system can handle a larger number of mobile producers without becoming overwhelmed. However, since this solution is an initial design, some issues to be resolved remain, including the rendezvous placement selection problem, which has not been discussed in the current design. The location of the rendezvous servers can significantly reduce the path stretch efficiently. However, simply selecting the closest one may not always be the best option, as the diverted path stretch length between the consumer and the producer may be longer than if the rendezvous were not used. Thus, selecting the placement rendezvous requires further investigation.

### 3 Related work

To the best of our knowledge, the problem of selecting the placement of rendezvous servers in an NDN network for producer mobility support has not been previously addressed in the literature. We propose network centrality as a method for selecting the locations of rendezvous servers in the NDN network to address the issue of path stretch in the KITE solution. Network centrality [5] is a measure of the importance of a node within a network, and it can be used to identify the most important or influential nodes or those with the highest degree of connectivity. By considering network centrality, we aim to minimize the path stretches between mobile producers and consumers to reduce data retrieval latency. There are several ways to measure the centrality of a node in a network to determine its relative importance. In this paper, we will use four measures of centrality commonly used in network analysis [6]. For a given network topology, we define  $G = (V, E)$  as an undirected graph, where  $V$  is the set of nodes (also known as routers) and  $E$  is the set of edges (also known as links). The following are the related centrality definitions:

#### 3.1 Betweenness Centrality BC

The Betweenness Centrality [7] measure reflects the number of times a node located on the shortest path between two other nodes in the network. The formula for betweenness centrality is:

$$C_{BC}(v) = \sum_{s \neq t \neq v} \frac{\omega(s, t|v)}{\omega(s, t)} \quad (1)$$

where  $v$  is the node,  $s$  and  $t$  are nodes in the network,  $\omega(s, t)$  is the number of shortest paths between  $s$  and  $t$ , and  $\omega(s, t|v)$  is the number of shortest paths between  $s$  and  $t$  that pass through  $v$

### 3.2 Closeness Centrality CC

The Closeness Centrality [8] measure represents how close a node is to all other nodes in the network, based on the shortest path between them. Mathematically, it is expressed as:

$$C_{cc}(v) = \sum_{t \in V|v} \frac{1}{d(v, t)} \quad (2)$$

Where  $d(v, t)$  is the shortest distance between nodes  $v$  and  $u$ . Closeness centrality

### 3.3 Degree centrality DC

The degree centrality [8] of a node is the number of edges connected to it. This can be represented mathematically as:

$$C_{DC}(v) = \deg(v) \quad (3)$$

where  $v$  is the node, and  $\deg(v)$  is the degree of node  $v$ .

### 3.4 Eigenvector Centrality EC

The eigenvector centrality [9] of a node is the relative influence of a node within the network based on the centrality of its neighbours. It is defined as follows:

$$C_{EC}(v) = \sum_{t \in Adj(v)} w(t, v) C_{EC}(t) \quad (4)$$

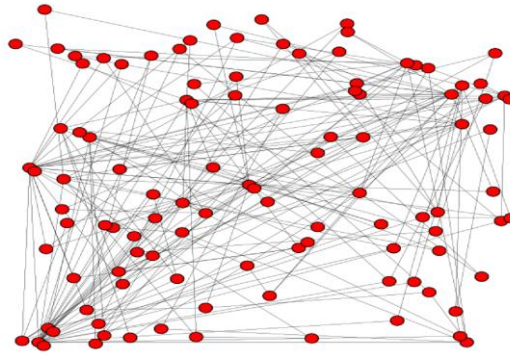
Where  $u$  is the node,  $v$  is a node in the network,  $Adj(v)$  is the set of neighbors of  $u$ , and  $w(t, v)$  is the weight of the edge between  $t$  and  $v$ .

## 4 Performance evaluation

In this section, we will first provide details about the simulation environment setup and system parameters. Then, we will present and discuss the results of our experiments.

### 4.1 Simulation setup

In our simulations, we used the Python package "networkx" [10] to compute network centralities including the degree centrality, closeness centrality, betweenness centrality and eigenvector centrality. Based on these measures, we identified potential locations for rendezvous servers and inputted them into the ndnSIM [11] simulator, where we implemented the KITE solution. To evaluate the effectiveness of our proposed solution, we conducted experiments using a random scale-free network generated using the Barabási–Albert (BA) model [12]. Figure 2. The scale-free network had 100 routers with 100 participants (99 consumers and 1 producer a special node to model IoT devices (e.g., eHealth applications)). We assumed that all network nodes act as routers and access points and forward data without using caching feature. The consumer continuously retrieved data from a MP for 100 seconds. The simulation was run for a total of 100 seconds, with each simulation repeated 10 times for each different MP position to obtain reliable results.

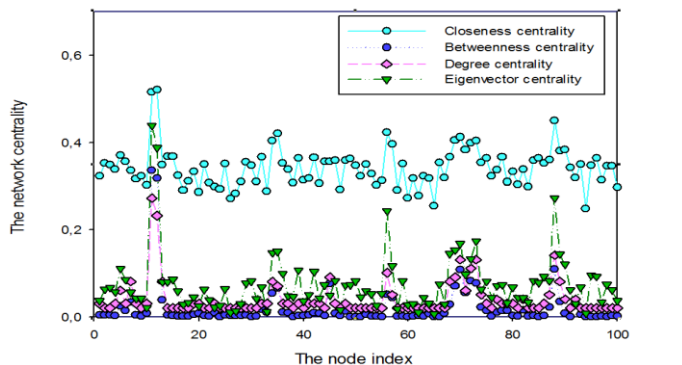


**Fig.2.** The Barabási–Albert model network topology.

## 4.2 Simulation Results

In the following, we present our study's results in three steps. First, we start by compute the network centralities of each node in the BA network topology. Secondly, we conducted a quantitative analysis to determine the optimal number of rendezvous servers that minimize the average hop count. This involved running simulations with different numbers of rendezvous servers under different network centralities. and measuring the average hop count for each configuration. By comparing the results of these simulations, we could identify the number of rendezvous servers that resulted in the lowest average hop count for the BA network topology. Finally, we also analysed the impact of mobile producer speed on the average hop count using the optimal number of rendezvous under different network centralities.

### 4.2.1 The network centrality of each node



**Fig.3.** The network centralities of each node in the BA Model.

To determine the placement of rendezvous servers in the network, we compute the network centralities of each node in the BA network topology. As shown in Figure 3, The results, demonstrate that each node has different centralities, which means that they have different levels of importance within the network. Based on these results, we can identify the candidate

locations for connecting the rendezvous servers. in this study, we select the top 15 most central nodes as rendezvous servers.

#### 4.2.2 The Average hop count

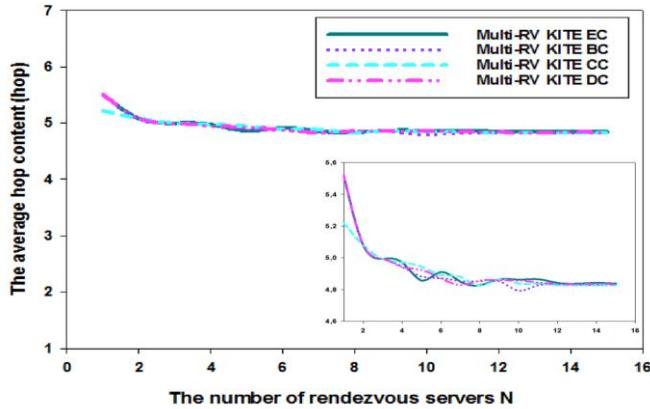


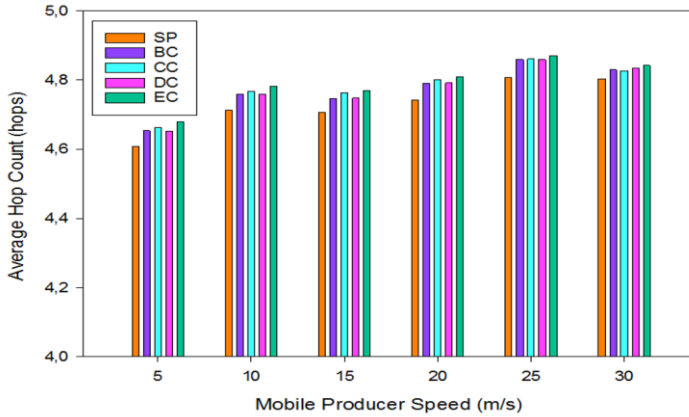
Fig.4. The average hop count vs the number of RV

In this section, we compare the average hop count using different numbers of rendezvous servers under different network centrality measures to identify which configuration performs best in terms of minimizing the average hop count. To calculate the average hop count, we need to divide the total number of hops taken by all the data packets received by consumers by the total number of data packets. Figure 4 shows the average hop count under a different number of rendezvous servers for the BA network topology. As a first observation, we can notice that with the increase in the number of Rendezvous servers, the average hop count is decreased, which means that the end consumer can obtain the content faster due to the increase of the rendezvous. Moreover, we observe a rapid decrease in the average hop count with increasing the number of rendezvous from N = 1 to around N = 5. We further notice that at a certain value of N= 8, the average hop count remains nearly constant. This is not surprising as the number of rendezvous servers increases, the centrality of the rendezvous servers decreases, which means that they have less impact on the network. This can lead to a less significant improvement in the average hop count, as the network is not as influenced by the rendezvous servers as it was with fewer rendezvous servers

#### 4.2.3 The Impact of mobile producer speed in Rendezvous Placement

To verify the impact of MP speed on the placement of RV servers in the multi-rendezvous KITE solution, we conducted a simulation using 8 RV servers (the optimal number in a BA network topology), with RV placement selected based on network centralities. The MP moves at a varying speed of 5m/s to 30 m/s using a random walk mobility model. the results are presented in Figure 5. By analysing the results, we can see that the average hop count increases as the MP speed increases. This is because as the MP moves faster, it becomes more challenging for the RV servers to keep up with its movement and maintain a connection. We can notice that the difference between the average hop count using the network centralities and the short path (SP) is relatively constant across different MP speeds. This suggests that if the RV servers are placed using the network centralities, they can maintain a connection with the MP and reduce the path stretch, even as the MP moves at high speeds. However, we

can notice the average hop count is slightly short when the RV servers are placed using the betweenness centrality measure compared to when they are placed using the closeness centrality or the degree centrality measure. This may be due to the betweenness centrality measure considering the shortest path between all pairs of nodes, which may result in fewer hops compared to when the RV servers are placed based on their proximity to the consumer (as in the case of the closeness centrality measure).



**Fig.5.** The Average hop count vs MP Speed.

## 5 Conclusions

In this study, we explored the use of network centrality measures to determine the optimal number and placement of rendezvous servers in the multi-rendezvous KITE solution for NDN-IoT networks. Our simulation results demonstrate that considering the centrality of nodes in the network can significantly improve the performance of the KITE solution, resulting in a lower average hop count and more efficient data retrieval. In future work, we plan to investigate additional factors such as energy consumption and communication cost to determine optimal rendezvous placements.

## References

1. Zhang, Y., Xia, Z., Mastorakis, S., & Zhang, L "Kite: Producer mobility support in named data networking." Proceedings of the 5th ACM Conference on Information-Centric Networking. (2018)
2. Afanasyev, Alexander, Ilya Moiseenko, and Lixia Zhang. "ndnSIM: NDN simulator for NS-3." (2012).
3. Zhang, Y., Afanasyev, A., Burke, J., & Zhang, L. A survey of mobility support in named data networking. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 83-88). (2016, April). IEEE.
4. Meddeb, M., Dhraief, A., Belghith, A., Monteil, T., Drira, K., & Gannouni, S. AFIRM: Adaptive forwarding-based link recovery for mobility support in NDN/IoT networks. *Future Generation Computer Systems*, **87**, 351-363. (2018).



5. sahl, Tore; Agneessens, Filip; Skvoretz, John. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, Vol. **32**, No.3, pp: 245, (2010)
6. Jurgen Pfeffer, Kathleen M. Carley, K-Centralities: Local Approximations of Global Measures Based on Shortest Paths, WWW2012 LSNA'12 Workshop, pp: 1043-1050, (2012)
7. Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* 40, 35-41, 1977. Newman, Mark EJ. "A measure of betweenness centrality based on random walks." *Social networks* 27.1 (2005)
8. Freeman, L. C. 1979. Centrality in social networks: Conceptual clarification. *Social Networks*, pp: 215-239, (1979)
9. Bonacich, P., Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, (1972).
10. Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, (Aug 2008)
11. Afanasyev, Alexander, Ilya Moiseenko, and Lixia Zhang. "ndnSIM: NDN simulator for NS-3." (2012).
12. R. Albert; A.-L. Barabási, *Statistical mechanics of complex networks*. *Reviews of Modern Physics*, Vol. 74, pp: 47–97, (2002).