# Named Data Object Organization in Distributed Name Resolution System for Information Centric Network Environment

Suhaidi Hassan, Walid Elbreiki, Shivaleela Arlimatti and Adib Habbal
*InterNetWorks Research Lab, School of Computing, Universiti Utara Malaysia,*
*06010 UUM Sintok, Kedah, Malaysia.*
*walid@internetworks.my*

***Abstract*—The Information-Centric Networking (ICN) is an emerging network communication model that focuses on what is being exchanged rather than who is exchanging information within a network. The named hosts make use of Named Data Objects (NDO) for data registration and name resolution. The Name Resolution System (NRS) is an element of the ICN that translates the object identifiers into network addresses. Distributing the NRS is an important and challenging issue for increased NDO registering, acquiring and storage. This study proposes a new NRS mechanism called the Distributed Name Resolution Mechanism (DNRM) to address the most significant issue of segregating the network and Balanced Binary Tree (BBT) structure to manage storage on the ever-increasing number of NDOs. The study formulates the proposed DNRM through the nearest neighbor algorithm by adding phases, means and methods, and probable outcomes. The stored NDOs are balanced with a balance factor to increase the scalability of NRS. The result shows that the overall NDO searching time is reduced by half for each iteration, and the proposed mechanisms are faster and more stable than the existing solutions in terms of better grouping NDOs. Both the mechanisms are simulated in the OMNeT++ simulation environment, a discrete event based simulator. The experimental results ensure that both the mechanisms have advantages in improving the network performance by minimizing the end-to-end delay and improving the network throughput.**

***Index Terms*—Information-Centric Networking; Name Resolution System; Named Data Objects; Balanced Binary Tree.**

## I. INTRODUCTION

Information Centric Networking (ICN) is a new network paradigm that reflects current usage of Internet based on information or content. The focus is based on information exchange (information dissemination and retrieval) rather than the location of information based on communication hosts. ICN has many advantages, which focuses on information exchange such as the multicast, anycast, cashing, mobility support, security and privacy [1].

Name Resolution System (NRS) is an overlay and the heart of the ICN, where names are used to match the IP address of the packets. The information or content organized in this system is in the form of Named Data Objects (NDO), where the names identify the content and guide forwarding the information within the network. Unfortunately, the size of the NDO storage is huge.

There are trillion of devices [2], such as phones, home appliances and sensors, which will offer unique additional content in the future [3]. The NRS should be able to handle all these unique NDOs, which may be in order to $101^6$ [4], in the future.

To mitigate the problems of NRS such as the interest query bottleneck, flooding, content congestion and NDO organization, this paper propose two mechanisms for distribution of NRS and organization of NDOs. In the first mechanism, NRS is distributed using the nearest neighbor algorithm. In the second mechanism, a balanced binary tree data structure is used to organize the NDOs in systematic order.

Both the mechanisms were simulated in the OMNeT++ simulation environment, a discrete event based simulator [5,6]. The main contributions of the paper are as follows:

1. Distributing NRS using the nearest neighbor algorithm, a well-known classification algorithm in data mining.
2. Organizing NDOs based on balance binary tree data structure.

The overall paper is organized as follows. Section II presents the related work discussed in the paper. Section III discusses the distributed name resolution system and Section IV presents data objects organization. In Section 5 performance evaluation is discussed. Finally, Section 6 concludes the paper.

## II. RELATED WORKS

The basic task of NRS is to map object names of its locators, which enables to reach the information about corresponding object, commonly known as IP addresses. The NRS till now mentioned in the literature of ICN, is non distributed and either based on DHT and based on BF. The distribution of NRS is proposed by Network of Information (NetInf). However, the implementation is not yet carried out and the proposed work is on random distribution [7,8].

Bloom Filter in Name Resolution System (BF-NRS) maintains and resolves the bind between the locators and names. It takes input as names and generates locator sets as output. The BF-NRS is based on a flat naming system, which is locator independent. The Flat naming system is the simplest name allocation system with high flexibility, most scalable and most advantageous in terms of privacy and persistency [9]. The main drawback of bloom filter is its quality of false positive and deletion of member is not possible. At the time of a deletion operation, reconstruction, operation is followed up in bloom filters which is expensive. This reconstruction is known as updating the bloom filter because of its inability to handle deletion operation itself.

The DHT based NRS is hierarchical, distributed and topologically embedded by the underlying network. Like bloom filters, DHT stores binding between object IDs and locators of object copies. The main design issue of DHT is its low latency, scalability, locality, agility, scoping and network utilization [10]. The main drawback of DHT is that all servers are linked in the form of circular linked list and the connections are stored in appropriate server other than any other servers. This causes serious trust problem with respect to authority issue and lookup messages are propagated through long paths.

Pastry [11] is decentralized object location, scalable and routing mechanism for large scale peer-to-peer system8. This is the overlay network where each node router interacts with local instances of applications. The protocol has the same drawback as the Chord [12].

## III. DISTRIBUTED NAME RESOLUTION SYSTEM

This section proposes a system model for distributed NRS. As per our convenience, we are dividing the network into two parts. i.e. two NRS. The NDO's in NRS distribution problem is considered as a network model which reduces a great deal of both practical and theoretical concepts. This model is built on the bases of Euclidean Distance (ED) for the nearest neighbor algorithm and divide the network based on the general metrics like time, distance and so on. In this research, time of publishing the NDO in the NRS is given as a priority, which is the time taken by a message to register in NRS.

K-nearest neighbor algorithm is a simple classification algorithm in Data Mining. In this research, we used this algorithm to classify the router among pre-defined classes known as NRS. i.e classification into two classes. Similarity measure used for classification is the Euclidean Distance Function [13]. A case classified based on the majority of its neighbors, with the assigned class amongst its K-nearest neighbor measured by a distance function. In this research, we are considering $K = 2$ i.e two class problem. Let us consider a number of publishing routers in the network are $N$. Assume the number of NRS are 2.

Let $T_1$ be a time taken by a publisher to publish NDO in NRS1.

Let $T_2$ is a time taken by publisher to publish in NRS2. Then $T_1$ is calculated by $T_1 = T_a - T_b$ where $T_a$ is a time to leave from the publisher and $T_b$ is a time it reaches NRS1. Similarly $T_2 = T_c - T_d$ , where $T_c$ is a time to leave from the publisher and $T_d$ is a time it reaches NRS2 each $T_1$ and $T_2$ are calculated by $T_a, T_b, T_c$ and $T_d$. The algorithm for the concept is as follows:

Algorithm 1 K-Nearest neighbor

```
BEGIN
  Input: Given N number of  publishers
  Output: network elements into two parts
    Methods
      T_1 = \sqrt{\sum_{i=1}^{n}(T_a - T_b)^2}
      T_2 = \sqrt{\sum_{i=1}^{n}(T_c - T_d)^2}
        If (T_1 < T_2)
        Then
      NRS1= Publisher Address
    Else
  NRS2= Publisher Address
END if
```

According to [14], the most accurate technique that achieves good classification is the k-neighbor neighbor algorithm. To find the nearest neighbors, this algorithm makes use of distance matrix D. The distance matrix is constructed using Euclidean Distance.

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{k}(T_i^k - T_{2i}^k)^2} \quad (1)$$

where $X_i$ and $X_j$ are two data vectors as explained earlier,

$$T_1 = \{T_{1i}, T_{1i+1}. \cdots T_m\}$$
$$T_1 = \{T_{2i}, T_{2j}. \cdots T_m\}$$

$T_1$ be a vector created using time, which takes to publish in NRS1, $T_2$ be a vector created using time, which takes to publish in NRS2. $D$ is a distance matrix which combines these two vectors.

$$D = T_1 T_2$$

$$\begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ X_3 & Y_3 \\ \vdots & \vdots \\ X_m & Y_m \end{bmatrix}$$

Based on this distance matrix, the classification of routes is carried out. The algorithm checks the condition of the node classification algorithm.

Algorithm 2 Classification of Nodes

```
For (i = 1 to m) do
  If (x_i, y_i)
  Then
    Classify the node to NRS1
      Else
        Classify the node to NRS1
End if
End for
```

## IV. S NAME DATA OBJECTS ORGANIZATION

The previous section of this paper discussed the classification of publishers/subscribers based on the time of publishing that is distributing them among the NRSs. The continuation of this work, which is the organization of NDO inside the NRS is developed in this section with the mechanism BBT-NDO storage. The concept of balanced binary tree data structure is used in this mechanism to store the NDOs. The tree accommodates up to 30 levels and each insertion is followed by balancing the tree using balance factor.

We divided the network into several domains as a prerequisite for this mechanism. Each domain has its own NRS to store the NDOs. All the resolution domains are interconnected. In our work, we consider one NRS for better understanding and set of operations on it.

Balanced Binary Tree data structure conceived to efficiently perform all data structure, basic operations on large data sets [15].

Let the storage of NRS, in which NDOs are organized in the form of balanced binary search tree, for its every interval node $v$ of $T$, the height of the children of $v$ can differ by at most 1.

Assuming these NDOs are numbered, the storage NRS is as shown in Figure 1. For easy understanding, throughout this paper, we assume the NDO number that represents the actual naming is not a number but a string. The end leaf node is the links kept for the next NDO storage.

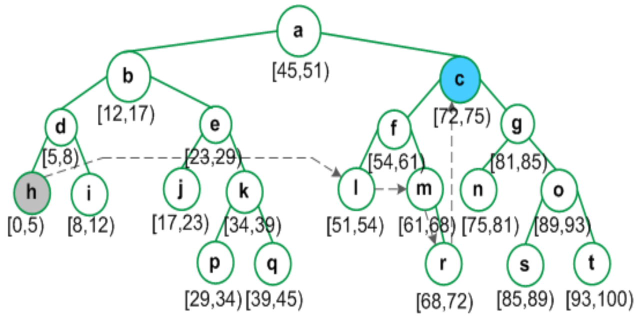


Figure 1: Balanced Binary Tree Data Structure

When a new content enters the publisher router, the publisher register this new content in the nearest NRS. Choosing the nearest NRS is already implemented in the DNRM mechanism of this chapter. The newly entered NDO is inserted in the NRS list according to the concept of BBT. And each insertion is followed by tree balance concept.

Algorithm 3 to Search an element in a balanced binary tree

```
BST_SEARCH (x, k)
  If (x == null || k(x) == k)
  Return x
     If (x < key(x))
   Return (BST_SEARCH(left(x), k));
     Else
      Return (BST_SEARCH(right(x), k));
 Where x = node and k = key
```

Let $T(n)$ is a number of comparisons made to search a key element in the tree. To search an element on the right side of the tree, the number of comparisons is $T(n/2)$ and the left side is $T(n/2)$. Each time, we remove half of the elements from the search space. If an $i^{th}$ iteration searching elements is n then in $(i+1)^{th}$iteration searching elements is $n/2$. Assume $n = 2k$

$$T(n) = T(n/2) + C$$

where $C$ is a constant.

For example:

$$T(n) = T(n/2) + 2$$
$$T(n/2) = T(n/4) + (2+2)$$
$$= T(n/8) + 2 + 2 + 2$$
$$= T\left(\frac{n}{24}\right) + 4.2$$

Therefore,

$$T(n/2k) + k.2$$
$$\text{As } n = 2^k$$
$$T(2k/2k) + k.2 = T(1) + 2k$$
$$T(n) = T(1) + 2k$$
$$\text{If } n = 2^k$$
$$\text{Then } k = log_2 n$$
$$\text{If } n = 1$$
$$T(n) = C1 + 2k$$
$$= C1 + log_2 n$$
$$= O(logn)$$

## V. Performance Evaluation

In this paper, OMNeT++ simulation environment with INET framework and ICNSim have been used to build simulation models. The simulation tool is well-designed and widely-used, modular system; OMNeT++ is an open source network simulator that has been made available for teaching, research and development purposes [4,15]. The simulator has been developed using Python and C++ along with scripting capability in a modular fashion as a set of libraries. These libraries can be combined together with other external software libraries for data analysis for better presentations of the outcomes [6].

### *A. Simulation setup*

An experiment environment has been created in the latest OMNeT++ version 4.6, with INET 2.6 framework along with ICNSim, and the GCC compiler 4.9 running on Ubuntu 14.04 LTS.

### *B. Results and discussion*

In order to model a more accurate environment, our distribution mechanism was evaluated over random topology. A number of publishers assumed here are 10, 20, 30, 40 and 50. The simulation was executed in 200 seconds, two different performance metrics, throughput and delay were analyzed to study the performance of our mechanism.

Figure 2 shows the average throughput of our mechanism, traditional distribution of NRS and single NRS network over random topology. The horizontal axis represents the number of publishers from 10 to 50 numbers. The vertical axis represents the total throughput in Mbps. Initially, for 10 numbers of publishers, the traditional distribution of NRS throughput was more as compared to our mechanism. Interestingly, our mechanism throughput increases gradually as the number of publishers is increasing. Single NRS and traditional distribution throughput over the same speeds followed a similar trend, but with lower values. However, our mechanism considerably increased in the later stages. The increase in the total throughput is because of more NRSs reduces the searching time and processing time, leading to an increase in the total throughput.

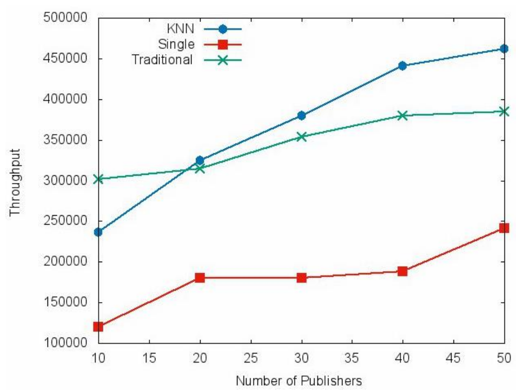


Figure 2: Throughput in Distributed NRS Network

Figure 3 presents an average end-to-end delay of our mechanism, traditional distribution of NRS and single NRS with a number of publishers. As compared with traditional distribution and single NRS, our mechanism shows better performance. Even though, the delay is increasing gradually in the graph, it reduces comparatively as compared to the previous two mechanisms.

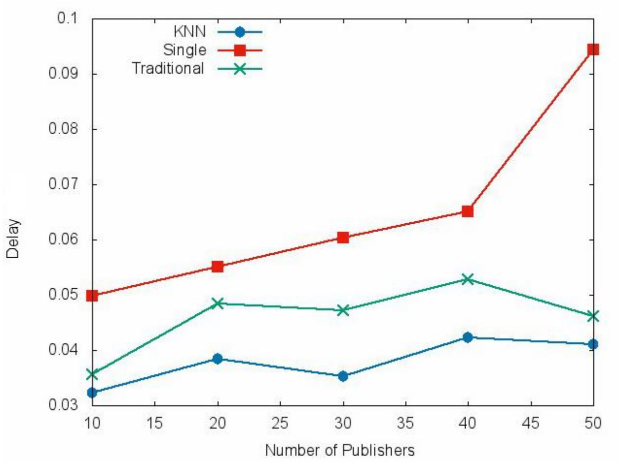


Figure 3: Delay in Distributed NRS Network

As a conclusion, the single NRS, traditional distribution of NRS and distribution with our mechanism were compared in random topology and the graph with the results are presented in the same section above. The analysis shows that all three mechanisms behave similarly with the number of NRS and network conditions. Our mechanism achieves slightly higher throughput compared to other two, and lesser average end-to-end delay.

### *C. Contributions*

The main contributions of DNRM BBT-NDO storage mechanism are as follows

- To the best of our knowledge, we are the only one among other researchers who are working on the distribution of the system NRS based on the time of registering the NDOs in the NRS by publishers. The main objective of this work is to classify the routers in to some clusters and each cluster will have one NRS to handle their NDOs.
- The classification of the routers has been conducted based on the time of publishing the NDOs. The minimum publishing time NRS is allotted with the router.
- The classification process has adopted the data mining with the nearest neighbor algorithm to classify the routers based on the NDO publishing time.
- The Balanced binary tree concept is the data structure which is used to store the data elements in systematic order. In our research, we used this tree data structure to store the NDOs which are the object identifiers.
- The BBT NDO storage minimizes the searching time in NRS.
- The BBT helps to organize the memory storage so that all of the operations, such as the new NDO insertion and NDO deletion are conducted properly. This helps in searching the NDO faster compared to other mechanisms because each iteration eliminates half of its space.

Simulations were conducted to evaluate the proposed mechanisms. The simulation result demonstrates that data mining algorithm at the nearest neighbor is the best classification algorithm, which can also be applied to classify network devices and BBT is the best storage for NDOs.

## VI. CONCLUSIONS

In this paper, we presented two mechanisms distributed name resolution system and named data organization in NRS. Distributed NRS aims to distribute the NRS based on the time of publishing the NDO in the system. The distribution mitigates the bottleneck, while sending the interest to NRS and flooding. On the other hand, the NDO storage uses balanced binary tree concepts. This mechanism overcomes the drawbacks of DHT and bloom filters as mentioned in the related work. To the best of our knowledge, both the mechanisms improve the performance of the ICN by increasing the scalability of the system, and same time reduces the end-to-end delay.

## REFERENCES


[1] G. Xylomenos, C. N. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos and others, "A survey of information-centric networking research," *IEEE, Communications Surveys, Tutorials,* vol. 16, pp. 1024--1049, 2014.

[2] J. Alpert and N. Hajaj, "Google we knew the web was big," 25 July 2008. [Online]. Available: https://googleblog.blogspot.my/2008/07/we-knew-web-was-big.html.

[3] D. Evans, "The internet of everything: How more relevant and valuable connections will change the world," *Cisco IBSG,* pp. 1--9, 2012.

[4] C. Dannewitz, M. DAmbrosio and V. Vercellone, "Hierarchical DHT based name resolution for information-centric networks," Vols. 36, no. 7, p. 736749, 2013.

[5] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," 2008.

[6] A. R. Khan, S. M. Bilal and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, 2012.

[7] S. Project, "Scalable and Adaptive Internet Solutions (SAIL)," 2010.

[8] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren and H. Karl, "Network of Information (NetInf) - An Information-centric Networking Architecture," *Computer Communications,* vol. 36, pp. 721-735, 2013.

[9] W. Elbreiki, S. Hassan, A. Habbal, M. Firdhous and M. Elshaikh, "A Comparative Study of Chord and Pastry for the Name Resolution System Implementation in Information Centric Networks," in *Conference: 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015),* Cyberjaya, Kuala Lumpur, Malaysia, 2015.

[10] P. Project, "Publish/Subscriber (PURSUIT)," 2010. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/.

[11] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," *IFIP/ACM International Conference on Distributed Systems Platforms,* pp. 329-350, 2001.

[12] I. Stoica, R. Morris, D. L. Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," *SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,* pp. 149-160, 2001.

[13] Barrett and Paul, "Euclidean distance: Raw, normalised, and double-scaled coefficients," *Unpublished paper retrieved from http://www.pbmetrix. com/techpapers/Euclidean_Distance. pdf,* 2006.

[14] Y. Li and L. E. Parker, "Nearest neighbor imputation using spatial--temporal correlations in wireless sensor networks," *Information Fusion,* vol. 15, pp. 64--79, 2014.

[15] G. S. Manku, "Balanced Binary Trees for ID Management and Load Balance in Distributed Hash Tables," in *Proceedings of the Twenty-third Annual ACM Symposium on Principles of Distributed Computing,* New York, NY, USA, 2004.