LETTER

A Novel Solution to Minimize the Interest Flooding and to Improve the Content-Store Performance for NDN-Based Wireless Sensor Networks

Muhammad MUDASIR QAZI^{†a)}, Rana ASIF REHMAN^{†b)}, Asadullah TARIQ^{†c)}, *Nonmembers*,
and Byung-Seo KIM^{††d)}, *Member*

SUMMARY Information-centric networking (ICN) provides an alternative to the traditional end-to-end communication model of the current Internet architecture by focusing on information dissemination and information retrieval. Named Data Networking (NDN) is one of the candidates that implements the idea of ICN on a practical level. Implementing NDN in wireless sensor networks (WSNs) will bring all the benefits of NDN to WSNs, making them more efficient. By applying the NDN paradigm directly to wireless multi-hop ad-hoc networks, various drawbacks are observed, such as packet flooding due to the broadcast nature of the wireless channel. To cope with these problems, in this paper, we propose an Interest called the accumulation-based forwarding scheme, as well as a novel content store architecture to increase its efficiency in terms of storing and searching data packets. We have performed extensive simulations using the ndnSIM simulator. Experimental results showed that the proposed scheme performs better when compared to another scheme in terms of the total number of Interests, the content store search time, and the network lifetime.

key words: content, interest, Named Data Networking, sensors, flooding

1. Introduction

Since the start of the Internet, there have been fundamental changes to the way the Internet is used because of the increased use of social networking services to view and share digital content, such as videos, pictures, documents, etc. Instead of providing basic connectivity, the Internet has become largely a distribution network with massive amounts of video and web page content flowing from content providers to viewers [1]. To meet the requirements of users, many researchers have proposed new Internet architectures following the paradigm of information-centric networking (ICN), but Named Data Networking (NDN) [1] has become one of the most gregarious proposals to achieve ICN. A wireless sensor network (WSN) is composed of a large number of low-cost, low-power sensor nodes that are

equipped with sensing, data processing, and communication components. WSNs are used to collect such data as temperature, humidity, pressure, pollutant levels, rare animals, and vital body functions [2]. A major research challenge in the area of WSNs is to improve their network lifetime [3], [4].

Due to the robust and simple communication methodology of NDN, it has rapidly gained huge popularity and consensus in the research community. It supplies built-in characteristics, such as multicast data delivery, data authenticity, in-network caching, etc. The authors in [5] introduced two kinds of forwarding strategies named blind forwarding (BF) and provider-aware forwarding (PAF) for NDN-based wireless ad-hoc networks. BF represents a type of controlled flooding scheme that mitigates packet flooding over the network. On the other hand, the PAF scheme also considers the location of the provider node by using the distance approach based on hop count. In [6], the authors proposed dual modes of Interest flooding called the directive mode and the flooding mode. The flooding mode is the same as in standard NDN, and the directive mode involves attempting to minimize Interest flooding in the network. In the directive mode, the Interest is flooded by giving it a destination node, and it is forwarded only to the destination. The authors of [7] suggested a locality-aware skip-list to improve performance. In their design, since the package will likely share a prefix with previous packages, the search can start in the bypass list directly near the previously accessed node for that prefix. The authors in [8] proposed a name search mechanism (named BF-PDT) that combines the Bloom filter, the popularity chart, and the degraded tire.

In this study, we propose a novel solution that will take care of the total number of Interests, as well as the performance of the content store, i.e., its searching and storing time. The key contributions of this study are as follows:

- The proposed scheme sends accumulative Interest requests for multiple data packets; it decreases network traffic, so bandwidth can be used more efficiently. A WSN node will be able to retrieve more data by sending fewer Interest requests. This leads to an improved content download time, lowered data redundancy, reduced Interest retransmission, and increased network lifetime.
- We have used the B-tree algorithm in an innovative way

Manuscript received March 30, 2020.

Manuscript revised September 29, 2020.

Manuscript publicized November 30, 2020.

[†]The authors are with the Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus 35400, Pakistan.

^{††}The author is with the Department of Software and Communication Eng., Hongik University, Sejong City, South Korea.

a) E-mail: mudasirqazi00@gmail.com

b) E-mail: r.asif@nu.edu.pk

c) E-mail: realasadullahtariq@gmail.com

d) E-mail: jsnbs@hongik.ac.kr

DOI: 10.1587/transinf.2020EDL8049

to improve the insertion and search time of the content store from a linear asymptotic function to a logarithmic function. The proposed architecture has a worst-case time complexity of $O(\log(n)) + O(\log(m))$ units for both insertion and searching.

2. Proposed Solution

The following are two major parts of our proposed solution, which are explained below:

(1) Accumulative Interest Names Forwarding Scheme

In the accumulative Interest names forwarding scheme, we propose that an NDN node should not send separate Interest requests for each piece of data it requires; instead, it should send a request for multiple data packets in a single Interest request. For example, consider a scenario where we have a consumer and producer in a network at the distance of a single hop. There are no other nodes in the network for the sake of this example. Further, let's suppose the consumer requires six data packets. In a traditional NDN scenario, six separate Interest requests will be sent and six data packets will be received in a reply one by one. This way, each node has to process 12 packets and transmissions, as well as receive six packets. In the accumulative Interest forwarding scheme, we propose that the consumer node should send only one or two Interest requests (depending on the accumulation size) to receive all six data packets. This will decrease the number of Interest transmissions. An Interest transmission uses much power in the context of WSNs, and it costs much to a node. However, if we decrease the number of transmissions, we can save node energy and, hence, a node can live longer, which will increase the network lifetime. Therefore, the question is how a node decides how many data packets it should require in a request. To answer this, we use a dynamic calculation method that depends on the content download time and data redundancy of the node. If a node (say: N_1) is receiving the required data quicker (i.e. from neighbor node) than normal cases, it means there is another node nearby (say: N_2) that has the required file and is replying with it. In such a case, the node will increase the accumulation size. Therefore, if in a previous Interest request two names have been accumulated, with a faster reply, three names will be accumulated for the next Interest request. This way, it will receive more data by sending fewer Interests. On the other hand, if the data reply for any Interest is slow or late, the node will decrease the accumulation size by one.

To achieve this in implementation, we have converted the *Name* field of the Interest packet to a list, so it can carry multiple names. Before flooding an Interest (say: I_i), the consumer node will check how many packets it requires and what its accumulation size (N) is. It then selects the names of the first N packets, assigns them to the *Name* field of I_i , and sends the Interest. This is how a single Interest packet can carry multiple names for the required data. When the Interest I_i is received at any intermediate node, the node will

retrieve the *Name* field and check if it has the required data for any of the names in the *Name* field by applying a loop on it. If the node has the required data, it remove its name from the I_i names list and further broadcast it to its neighbors, so that unsatisfied Interests can be fulfilled. Similarly, when the I_i is received by any producer node, it will check the required names by applying a loop in the *Name* field of the received packet. In the loop, it will reply with the corresponding data for each name. When the consumer node receives the data as a reply to its Interest request, it checks for the total delay and data redundancy. If the delay is lesser than the delay of the previously sent request, the node will increase the accumulation size by one; if the delay is increased, then the node will decrease the accumulation size by one. A node can achieve a maximum accumulation size of 10 and a minimum of one. We have conducted extensive experiments to know the minimum and maximum limits. If we keep increasing the accumulation size, it will end up decreasing the efficiency because of the loops added in PIT and CS tables. Similarly, when the data redundancy has decreased, a node will increase the accumulation size by one, and if the data redundancy is increased, then the node will decrease the accumulation size by one.

Interest handling in the Pending Interest Table (PIT) is also updated as the *Name* field is updated and is now an array of names. Therefore, when an Interest packet is received at a node, a loop is applied on its *Name* field to check for all the names if any of the name is present in PIT. All names from the *Name* array, which are already present in PIT, are removed from Interest and it is then moved further to check in Content-Store.

(2) Content-Store Architecture

The content store has a major role in the performance of a node, specifically a WSN node, which has a limited battery life and many packets to process. Most of the time, the content store searches the data from an already existing packet or stores new data in it. The algorithm that conventional NDN uses for a content store is a skip list [7]. A skip list has a worst-case time complexity of $O(N)$, which slows it down and causes delays. We can improve this by using our proposed content store architecture, which uses B-tree algorithms in an innovative way to improve the content stores worst-case time complexity, as in Eq. (1), where n represents the number of nodes in a file tree and m represents the number of nodes in a chunk tree. This means the proposed architecture will bring the time complexity of the content store from a linear function to a logarithmic function, leading to a much better performance.

$$O(\log(n)) + O(\log(m)) \quad (1)$$

NDN packet names are in hierarchical order, containing the file name and its chunk name, e.g., book1/chunk1, book1/chunk2, and so on, where book1 is the name of the file and chunk1 and chunk2 are its chunks. Therefore, these two chunks belong to the same file as they have the same file name prefix. Note that we use the words chunk and data

packet herein interchangeably. Using this naming scheme, we have divided the content store into two parts, called the file tree and the chunk tree. Both of these parts are made up using the conventional B-tree data structure. When a data packet (or chunk) is received, we separate its file name and chunk name. The file name is stored in the file tree, while the chunk is stored in the chunk tree, along with the data payload (actual data). For each distinct file in the file tree, there is a separate chunk tree for its chunks. Therefore, in our content store, there will be only one file tree and many chunk trees. The number of chunk trees depends on the number of nodes in the file tree. Each file tree node has a pointer to its chunk tree to retrieve its data. An example scenario is as follows: to understand our proposed architecture, let us look at the example where a node is receiving data packets in the following order, as shown in Fig. 1. First, it receives the c1 for File1, then the c2 for File2, and so on up to c7 for File3. In total, it receives 21 packets from three different files. Figure 2 shows the insertion of all received data packets step by step. The final content store will contain only a file tree with three nodes for each distinct file, and each node in the file tree has a pointer to its respective chunk tree, which holds the actual chunks. Insertion in both trees is done according to the rules of the standard B-tree algorithm.

1	2	3	4	5	6	7	8	9	10	11
file1/c1	file2/c2	file2/c5	file1/c5	file3/c4	file3/c1	file3/c5	file3/c9	file3/c2	file1/c3	file1/c2

12	13	14	15	16	17	18	19	20	21
file2/c1	file2/c4	file3/c3	file3/c8	file1/c4	file3/c6	file2/c7	file2/c3	file2/c6	file3/c7

Fig. 1 Order of reception of data packets at the node

Fig. 2 Data presentation in proposed architecture

3. Simulation and Results

For the implementation of the proposed solution, we have used the ndnSIM simulator, which is specifically developed for use with NDN. We have used the approach from the paper Energy Efficient Interest Forwarding in NDN-based Wireless Sensor Networks [6], called dual mode Interest forwarding (DMIF), as a baseline for comparison, as it is the most suitable. A grid topology of 10×10 nodes is used for experiments by utilizing the constant position mobility model. Table 1 represents all the parameters of the experimentation environment for both proposed solution and DMIF. For the short length of this paper, we will discuss only the total number of Interests, the packet search time, and network lifetime performance metrics.

(1) Total Number of Interests

Figure 3 shows a comparison of the total number of Interests flooded in the network using both the proposed scheme and DMIF. It shows that the proposed solution is leading for a simple reason. Instead of sending each packet separately, multiple Interest names are accumulated in one request to

Table 1 Simulation parameters

Parameter	Value
Interest size	50 bytes
Data payload size	1200 bytes
Wireless interface	IEEE 802.15.4
Radio coverage range	125 m
Number of nodes	20 to 100
Number of content producers	(1 to 5) out of 100
Number of content consumers	(1 to 5) out of 100
Initial energy	50 Joule
Simulation time	1000 sec
Mobility model	ConstantPositionMobilityModel
Placement policy	LCE
Replacement policy	LRU
Content-Store size	1000 packets
RTT computation	Jacobsons estimation
Energy model	EnergyModel
Data rate	24 Mbps
Power consumption (T_x)	1.58 W
Power consumption (R_x)	1.31 W
Power consumption (idle)	1.22 W

Fig. 3 Comparison of total number of interests

Fig. 4 Comparison of total search time (ms)

Fig. 5 Comparison of network lifetime (s)

minimize total Interest flooding. The proposed solution also has the edge of engaging in dynamic behavior to get the best results. On the other hand, DMIF minimizes flooding by sending the Interest in flooding or directive mode. After receiving the Interest in directive mode, a node then decides whether to flood it further. However, the flooding mode does not apply any flooding prevention mechanism. According to Fig. 3, when we vary the number of pairs from 1 to 5, the total number of Interests is increased. This is because, at the start, there is only a single consumer to transmit Interest requests. As you increase the number of consumers, it will also increase the total number of Interests in the network. In addition, each request may get re-flooded by each intermediate node again. Therefore, the total number of Interests will increase when the number of pairs is increased. This increase will also affect both schemes, but the proposed solution manages this better than DMIF.

(2) Search Time

DMIF has not proposed anything related to the content store architecture, and it uses the default architecture. Figure 4 shows a comparison of the time required for searching packets in the content store, varying from 200 to 1000. It is clearly shown that the proposed architecture takes much less time compared to DMIF. In addition, when the number of packets increases, the increase in DMIF is linear, while it is logarithmic in the proposed solution. This is because of

the nature of algorithms and their asymptotic behavior, as explained in the proposed solution section.

(3) Network Lifetime

A WSN node has very limited energy, and it requires much more energy to transmit or receive an Interest than to process it. Therefore, a lower number of Interest transmissions/receptions will save the energy of the node, and it will increase the network lifetime. Figure 5 shows a comparison of network lifetime between DMIF and the proposed solution. Clearly, our proposed solution has improved the network lifetime.

4. Conclusions

Conventional NDN uses the one-Interest/one-data pattern to get data from neighbors and a skip list as the underlying architecture of the content store. We have proposed a one-Interest/multiple-data pattern by accumulating multiple names in a single Interest request and dynamically managing the accumulation size. We have also proposed a novel content store architecture that uses a B-tree as its underlying structure and improves the performance. We have implemented and performed experiments of our solution using ndnSIM. The overall performance improvement is 8% when we compared our proposed solution with the DMIF results.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2018R1A2B6002399).

References

- [1] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," *Proc. 5th international conference on Emerging networking experiments and technologies*, pp.1–12, ACM, 2009.
- [2] H.M.A. Fahmy, "WSNs applications," *Wireless Sensor Networks*, pp.69–213, Springer, 2016.
- [3] M. Pawar and J. Agarwal, "A literature survey on security issues of wsn and different types of attacks in network," *Indian Journal of Computer Science and Engineering*, vol.8, no.2, pp.80–83, 2017.
- [4] M. Miettinen and N. Asokan, "Ad-hoc key agreement: A brief history and the challenges ahead," *Computer Communications*, 2018.
- [5] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *Journal of Network and Computer Applications*, vol.50, pp.148–158, 2015.
- [6] S. Gao, H. Zhang, and B. Zhang, "Energy efficient interest forwarding in ndn-based wireless sensor networks," *Mobile Information Systems*, vol.2016, 2016.
- [7] T. Pan, T. Huang, J. Liu, J. Zhang, F. Yang, S. Li, and Y. Liu, "Fast content store lookup using locality-aware skip list in content-centric networks," *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp.187–192, IEEE, 2016.
- [8] H. Hao, C. Xu, S. Yang, J. Guan, Y. Liu, and L. Zhong, "Bf-pdt: A new name lookup mechanism in content-centric networking," *2017 International Conference on Networking and Network Applications (NaNA)*, pp.69–74, IEEE, 2017.