Received 26 December 2022, accepted 4 February 2023, date of publication 13 February 2023, date of current version 17 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244247

RESEARCH ARTICLE

E-DRAFT: An Efficient Data Retrieval and Forwarding Technique for Named Data Network Based Wireless Multimedia Sensor Networks

AHMAD ARSALAN[1], MUHAMMAD BURHAN[2], RANA ASIF REHMAN[3], (Senior Member, IEEE), TARIQ UMER[4], (Senior Member, IEEE), AND BYUNG-SEO KIM[5], (Senior Member, IEEE)

[1]Faculty of Information Technology and Computer Science, University of Central Punjab, Lahore 54000, Pakistan
[2]Department of Information Technology, University of the Punjab, Gujranwala 54590, Pakistan
[3]Department of Computer Science, National University of Computer and Emerging Sciences, Lahore Campus, Lahore 54700, Pakistan
[4]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
[5]Department of Computer and Information Communication Engineering, Hongik University, Sejong City 30016, South Korea

Corresponding author: Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant by the Korea Government through Ministry of Science and ICT (MSIT) under Grant 2022R1A2C1003549.

ABSTRACT Wireless Multimedia Sensor Network (WMSN) is a wirelessly connected sensor network featuring multimedia devices such as cameras and microphones that can retrieve video and audio streams, photographs, and scalar sensor data. Because of the low cost of cameras and microphones, in addition to significant advances in distributed signal processing and multimedia data coding approaches, WMSNs can offer multimedia content. Moreover, the most critical difficulty in the implementation of WMSN is to improve the efficiency and lifetime of the network. The Named-Data Network (NDN) technique is used in WMSNs to increase communication and transmission mechanisms efficiency. NDN is thought to be one of the most demanding Internet architectures in the future. Unlike host-centric techniques that focus on the location of the content, NDN focuses on providing the content. In this paper, we propose a novel accumulative Interest-based content-store architecture for NDN-based WMSNs to improve the efficiency of individual nodes as well as the overall network. Furthermore, the proposed solution also controls packet flooding. The proposed scheme is named ''Efficient Data Retrieval and Forwarding (E-DRAFT)''. Moreover, the ndnSIM simulator is used to run comprehensive simulations. Compared to existing methodologies, the simulation results reveal that the proposed approach stops additional packet flooding and effectively stores and retrieves Data packets with enhanced processing efficiency.

INDEX TERMS NDN, wireless, sensor, network, multimedia, Internet.

I. INTRODUCTION

The Internet has seen massive development since its first inception. At first, the Internet was used by a relatively small group of people but with time the usage of the Internet increased exponentially. Mainly Internet is used for social networking sites to communicate with others by transferring photos, videos, documents, etc. [1]. Besides providing basic communication facilities Internet is also used for the distribution of large content from one service provider to another. Wireless sensor networks (WSNs) are a type of ad-hoc network in which nodes are composed of tiny sensors. These sensors can transfer a limited byte of data. Examples of these sensors are a temperature sensor, pressure sensor, alarm sensor, humidity sensor, and water level sensor. Due to the availability of CMOS-based hardware and tiny microphones, the WSNs paved the path for Wireless Multimedia Sensor Networks (WMSNs). In WMSNs, multimedia content can be sent across a network and this data can be in form of pictures, videos, or audio. Sensors that are used in WMSNs are small cameras or microphones. WMSN sensor nodes consist of many types of sensors that can transfer multimedia

The associate editor coordinating the review of this manuscript and approving it for publication was Zihuai Lin.

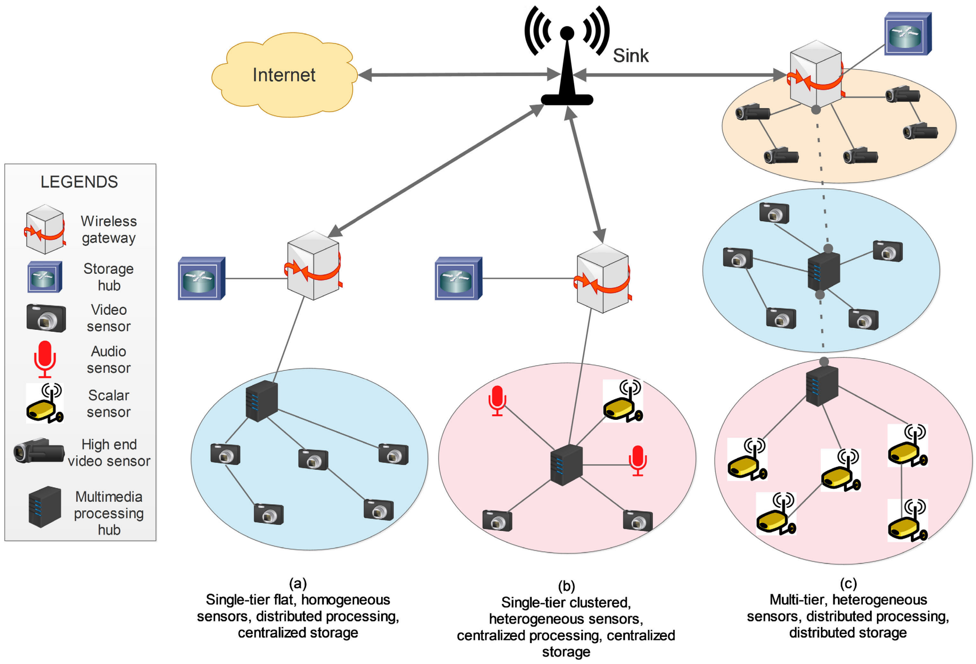


FIGURE 1. WMSN architecture.

TABLE 1. Abbreviations.

WMSN	Wireless Multimedia Sensor Network
ICN	Information Centric Network
NDN	Named Data Network
WSN	Wireless Sensor Network
CMOS	Complementary Metal Oxide Semiconductor
IoT	Internet of Thing
ITS	Intelligent Transportation System
EIPS	Enhanced Interest Packet Size
E-DRAFT	Efficient Data Retrieval & Forwarding Technique
CS	Content Store
NL	Network Lifetime
CST	Cache Search Time
CIT	Cache Insertion Time
TNI	Total Number of Interest

content. These sensor nodes are dispersed throughout a network and work in parallel to transfer content with the centralized node. This centralized node is often called a controller node. The gathered data is subsequently transferred to a storage hub, where it is used locally or over the Internet. Sensor devices have been upgraded to enable both audio and video modules as technology has advanced [2]. Some of the application areas of WMSN are surveillance networks [3], traffic avoidance [4], health care [5], smart homes [6], and environmental and agriculture monitoring [7]. Same as WSN the main challenge in WMSNs is to extend their network lifetime and improve network performance [8], [9].

Figure 1 depicts a typical WMSN architecture with various components such as sensor nodes, gateways, storage hubs, and Internet clouds. Moreover, WMSN is divided into three categories.

- Single-tier flat architecture: It's an architecture that is uniform and flat, and sensors with physical properties are utilized. Nodes with the same data processing functionalities are called multimedia processing hubs.
- Single-tier clustered architecture: There are nodes with varying physical capabilities in single-tier grouped architecture, such as multimedia nodes and scalar nodes [10]. Clustered heads (CH) are nodes with additional capacity, processing power, batteries, and more. The remaining nodes interact with these CHs, which process and transmit the data to the receiving node. The information is kept in central storage which is called a hub.
- Multi-tier architecture: In this architecture there exist several layers of nodes with various kinds of processing duties. The data is dispersed for processing and storage.

Thanks to WMSNs, mobile data traffic has grown dramatically [11]. People nowadays commonly utilize these multiple sensor devices to publish their content [12]. Furthermore, regardless of their destination location, individuals are increasingly interested in obtaining the needed content. Traditional Internet communication is based on a push-based mechanism in which each device requires a unique IP address to be recognized. Hence, it is unsuitable for future Internet applications. The current Internet, which is built on TCP/IP, has several flaws, including location-dependent mobility support, scalability, and security concerns. Unlike a traditional network, future internet architecture works on a pull-based mechanism. In such a architecture nodes are identifiable using their content names rather than their IP addresses. Many researchers have proposed multiple new Internet architectures to facilitate users' requirements. Information-centric network (ICN) has become one of the most famous architectures both in academics as well as in industry [13]. All networks, including delay-tolerant networks, mesh networks, LANs, autonomous networks, sensor networks, and vehicular and mobile ad-hoc networks, can benefit from ICN's

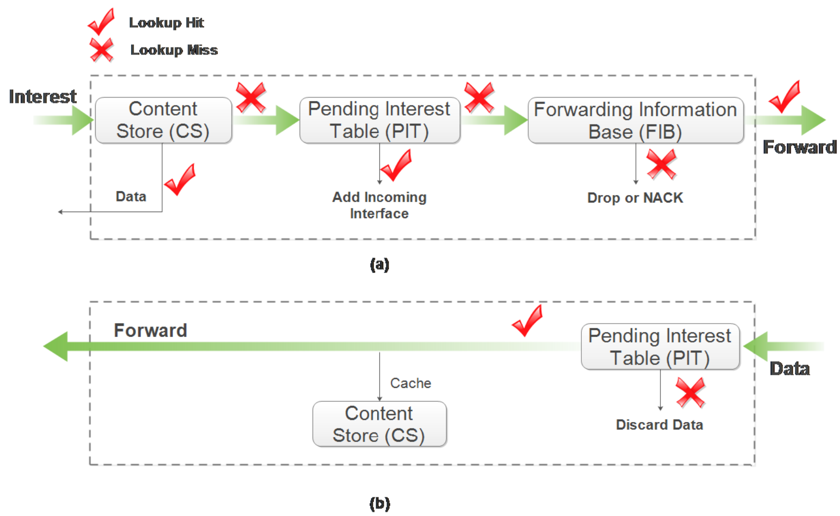


FIGURE 2. NDN packets forwarding flow.

content-oriented characteristics [14]. Several projects are presently happening in ICN. But in all of them named data network (NDN) [15] is the most notable due to its simple and resilient design. The basic difference between NDN and traditional networks is that NDN replaces IP architecture with named contents. NDN addresses conventional IP-based Internet problems while also introducing new features like built-in caching, data authentication, mobility, flow balance methods, and multi-cast data delivery. Named Data Network (NDN) is a new emerging future Internet paradigm that promises to answer all the limitations of traditional Internet Protocol (IP) based networks. NDN creates a communication paradigm that is simple and dependable on the consumer. Interests and Data are two sorts of packets that are exchanged in NDN communication. NDN maintains the reliability and originality of content by combining the content publisher's signature with various types of validation in each chunk of content. Every node in NDN has a content cache that provides speedy data/content retrieval and low network load time. Every NDN node contains three entities (I) Content-Store (CS) for holding frequently used Data packets (II) Pending Interest Table (PIT) for tracking records of Interests that have yet to be satisfied (III) Forwarding Information Base (FIB) for guiding Data and Interest packets [1].

In Figure 2, (a) depicts the decision flow of the Interest packet and (b)) depicts the Data packet flow. For WMSNs, NDN can be a useful option. NDN capabilities are linked to the use cases and applications of WMSNs which allow any possible constraints to be addressed. WMSN can enjoy the benefit of NDN in terms of the cache [16], scalability [17], transparency, and aggregation [18]. Table 1 shows the abbreviations that are used in this paper.

A. CONTRIBUTIONS AND ORGANIZATION

The major contributions of our proposed work are given below:

1) **NDN Based WMSN:** We propose a novel NDN-based WMSN in which communication between sensor nodes is done through the content name.
2) **Efficient Data Forwarding:** We implement a lightweight yet effective forwarding mechanism for our Data packet forwarding in NDN-based WMSN. We control the Data packet response whenever an accumulative Interest packet arrives on the sensor node.
3) **Efficient Content Store:** We change the default NDN CS structure using B+ Tree as it is faster compared to other data structures. Through this cache hit ratio and network throughput increase.
4) **Modified NDN Interest and Data packet:** We extend the default NDN Interest and Data packets and add a few more fields related to our scheme. These extra fields help us to control flooding in the network in response to accumulative Interest packets.

The goal of E-DRAFT is to provide an efficient method for obtaining data from a node cache and subsequently managing data packet flooding while in existing schemes, the authors don't focus on accumulative data packet management.

The remainder of the paper is structured as follows: In Section II, we discussed previously suggested NDN-based WMSN studies. A detailed description of the problem statement is given in Section III. Section IV goes into great depth on the proposed E-DRAFT protocol. Section V is dedicated to extensive performance assessments and results. In Section VI, we discussed some limitations regarding NDN network. The conclusion and future work are presented in Section VI.

II. RELATED WORK

Because of its obvious advantages and straightforward architecture, wireless NDN has recently attracted the attention of the research community.

In [19] and [34] the authors offer a technique to reduce network packet flooding. Relay nodes make judgments depending on their distance from the provider node as well as the time it takes to receive data.

The authors in [20] describe Enhance content-centric (CCN) multi-hop wireless networks (E-CHANET), a novel protocol for CCN-based wireless networks. E-CHANET technique also contains a distance table, which determines packet forwarding based on the distance between the consumer and producer nodes. In [21] authors present a solution for multi-store based wireless CCN. Two new packages, EFS-ACK and EFS, are modified in addition to the Interest and Data packets.

The authors in [22] and [23] present a novel protocol that forwards packets depending on the proper address. Each participating node employs GPS technology as well as two extra packages (i.e. CMD, ACK) to reduce network overhead.

The authors in [24] and [35] discussed an energy-based forwarding method for multi-hop wireless ad hoc networks in which packets are routed based on the node remaining energy.

In [25] authors proposed two types of strategies to forward the packets in the wireless network. The first strategy is known as blind forwarding and the second strategy is known as provider-aware forwarding. The first strategy controls overhearing of Interest packet redundancy. For this purpose, some additional information and defer timers are used. Whereas, the second strategy, attaches additional information regarding the provider in the Interest packet as well as the Data packet.

In [26] authors introduced AIRDrop, a new method that uses unicast communication. The suggested AIRDrop method additionally includes tables and buffers which are used during communication operations. In [17] authors proposed a scheme, named the dual-mode Interest forwarding scheme. It is a dual-mode Interest forwarding scheme where two types of mechanisms are used; (I) flooding and (II) directive. The flooding mode is the same as the conventional way of NDN, whereas the directive mode controls the broadcast storm of redundant packets in the network. In the directive mode, the Interest packet is forwarded to only the destination node. However, the proposed scheme reduces the problems of the broadcast storm as well as ensures the issue of energy consumption.

In [27] authors proposed the idea of a skip list, to save time for information retrieval. In the proposed scheme, a packet shares a prefix with its previous packets and then searches in the skip list by starting from the closest node previously obtained from the prefix. Therefore, the correspondence proposed scheme saves significant time.

In [28] authors combined a bloom filter and popularity chart to propose a mechanism named BF-PDT. The bloom filter is used to confirm whether either element exists or not. Moreover, the popularity chart and degraded tier are based on access characteristics and are used to improve the overall performance of the tier.

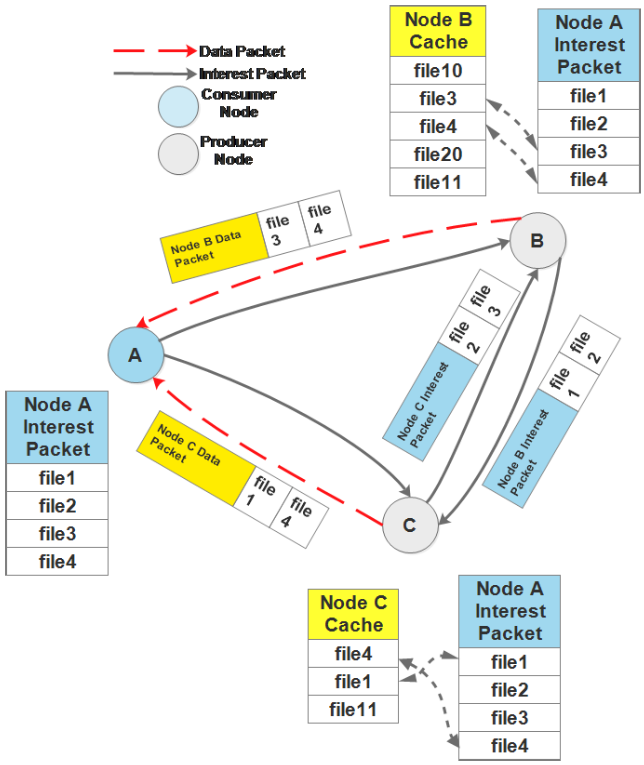


FIGURE 3. Accumulative interest/data packet forwarding problem.

III. PROBLEM STATEMENT

Our problem statement is divided into two parts. In the first, we discussed Interest/Data packet flooding whereas in the second part we discussed the limitation of CS in WMSN-based NDN nodes.

A. INTEREST/DATA PACKET FLOODING

In [30], the authors proposed accumulative Interest names forwarding scheme in which NDN-based WSNs send Interest packets accumulatively rather than sending Interest separately. To better understand Interest/Data packet flooding let's take an example. In Figure 3, there are three WMSN nodes. Node A is the consumer node while nodes B and C are producer nodes. These three nodes are at the distance of a single hop. Node A requires 4 Data packets (file1, file2, file3, and file4) which it sends through an accumulative (single) Interest packet. Node A Interest packet will be received on both node B and node C. After receiving the Interest packet from node A, node B and node C both will check the node A Interest packet with their cache. Node B cache contains file10, file3, file4, file20 and file11. While node C cache contains file4, file1, and file11. After checking node B, we find two entries (file3, file4) in its cache, and in node C we find two entries (file4, file1) as well. Then node B will forward the matching entries (file3, file4) and node C will forward entries (file1, file4) back to node A. After this, node B will broadcast the remaining unmatched entries (file1, file2) into the network. As seen in Figure 3, node C lies in the range of node B, so node C will receive a broadcast message. At the same time, node C will also broadcast the

Data Structures	Time Complexity (worse case)				Space Complexity
	Search	Insertion	Deletion	Access	
Linked List	O(n)	O(1)	O(1)	O(n)	O(n)
Skip-List	O(n)	O(n)	O(n)	O(n)	O(log(n))
B-Tree [3]	O(log(n)) + O(log(n))	O(log(n) + O(log(n))	O(log(n)) + O(log(n))	O(log(n)) + O(log(n))	O(n) * O(n)
B+ Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)

FIGURE 4. Comparison of different data structures.

remaining unmatched entries (file2, file3), and these entries will be received at node B. Now during this transmission, node C Interest packet contains a file3 entry which is already been satisfied, and node B Interest packet contains a file1 entry which is also satisfied so through the accumulative Interest packet forwarding scheme duplicate Data packets will be generated by nodes. This duplication of Data packets is directly proportional to the number of nodes in the network. Moreover, searching in the CS of NDN node is slow and time-consuming. Moreover, searching in the CS is done in linear time which increases network complexity.

B. CONTENT STORE (CS) COMPLEXITY

The Skip-List data structure is a more advanced form of Linked-List data structure. The Skip-List data structure is used in NDN-based architecture to store and search data from CS [29]. CS functions took O(n) time which is considered the worst-case time complexity and O(n log (n)) is considered its worst spatial complexity. WMSN nodes have restricted resources in terms of poor processing power and battery backup so in that case, the worst-case time complexity of the Skip-List data structure in WMSN might be difficult to handle. Moreover, in the article [30] they used B-Tree to improve CS architecture in WSNs. They extract files and chunk names from the Data packet and store them in their respective trees. Both these trees work separately. Every time a different file is received chunks tree saved the file. This leads to the problem of memory shortage in the CS and needs more processing time. Also, the time complexity of the B-Tree increases significantly. As WMSNs have low memory space and low processing power so this is a fatal problem that needs to handle. With such limited resources, the time complexity of a WMSN node should avoid worst-case scenarios and high-performance circumstances. This is a significant flaw that exists both in the traditional NDN and in B-Tree [30]. These techniques are compared with our proposed approach in the result section. On the other hand, B+ trees can be able to accomplish this task better than Skip-List and B-Tree due to their better time complexity and storage. Talking the worst-case time complexity of the B+ tree is O(log(n)), while the worst-case spatial complexity is O(n) [31] that's why the B+ Tree data structure is superior to the Skip-List and B- Tree data structure. Figure 4 compares Linked-List, Skip-List, and B+ Tree data structures in terms of time and space complexity. Figure 4 demonstrates that B+ Tree outperforms Skip-Lists and B-Tree for big data sets. The B+ Tree is a well-kept data structure. We just uniquely utilized the B+ tree to increase the overall efficiency of the content store in this paper.

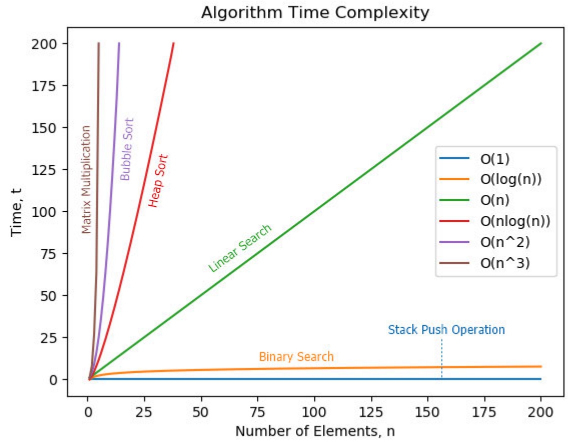


FIGURE 5. Algorithm time complexity graph [31].

Figure 5 depicts a visual representation of the above-mentioned data structures. Figure 5 clearly shows that O(log(n)) which is represented by the orange line, has a lower time complexity than O(n), as seen in the green line. Furthermore, algorithms with O (1) complexity have the best performance, as shown by the blue line, but algorithms $O(n^3)$ complexity has the lowest performance, as shown by the dark brown line. Meanwhile, performance deteriorates when the line changes from blue to dark brown. Figures 4 and 5 show that the B+ Tree data structure performs better than the Skip-List and B Tree data structure in terms of complexity.

IV. PROPOSED FRAMEWORK

To cope with the above problem statement, we divided our framework into two parts. In the first part, we address the issue of Data packet forwarding while in the second part we address the issue of CS.

A. RESEARCH METHODOLOGY

In this paper, we investigate our proposed model in the context of WMSN using a qualitative research methodology named design science research methodology (DSRM) [32]. In order to do that, we first identified the issue and then established our goals. In the second step, we created our architecture, and in the third, we put our plan into action. In the final step, we ran extensive simulations to check the efficacy and efficiency of our proposed method. Figure 6 depicts the entire DSRM cycle.

B. PART I: EFFICIENT DATA RETRIEVAL AND FORWARDING TECHNIQUE

As described in the problem statement, accumulative Interest packet forwarding generates duplication of Data packets in a network. To solve this problem, we proposed a novel E-DRAFT (Efficient Data Retrieval and Forwarding Technique) scheme. The working of the E-DRAFT scheme is shown in Algorithm 1 and it is explained below.

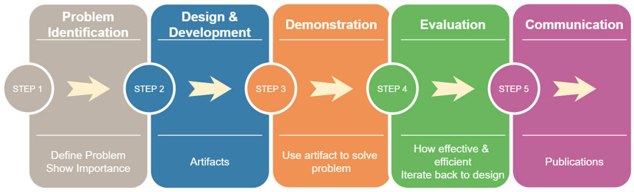


FIGURE 6. **Design science research methodology (DSRM) cycle [32].**

- Lines 2-4: Whenever a node receives an Interest packet, it performs the default NDN mechanism, which is it first checks the CS, and PIT and decides whether to forward the packet in the network.
- Lines 5-7: After receiving the Interest packet, a node will start its timer to minimize the broadcast storm problem for Interest and Data packets. A node will check its timer continuously until it expires. If the timer expires then the flow will jump to line 28 otherwise line 7 will continue.
- Line 8: At this line, a node will check whether a similar packet has arrived again or not. If a similar packet arrived again then line 9 executes and if no packet arrived, then the flow will continue from line 10 onward.
- Lines 10-15, 26-28: In line 12, Satisfied_Node_ID column entries are checked with NULL values and Interest packet entries will be matched with the node's CS one by one. If this check returns a true value then it means that there exists some content that needs to be answered so lines 13, 14, and 15 will execute. At the above lines, a node will generate a Data packet according to matched entries and the flow will transfer to lines 26 and 7 where the node will wait to broadcast packets until its timer expired.
- Lines 16-25: Otherwise, if the above check at line 12 returns false then this means that the Satisfied_Node_ID column is already filled. In this case, a node will discard that incoming Interest packet and will stop the broadcast at lines 17, 18, and 19.
- Lines 29-33: These lines will execute whenever the timer of a node expires. At line 29, a boolean variable packetUpdated is checked to see whether the Satisfied_Node_ID column has been updated or not. If the column has been updated, then it means that new content has been found that matches the Interest packet entry and CS of a node. In this case, a node will broadcast two packets in a network: one will be a Data packet that is created due to new matched data and the second packet will be the modified Interest packet. The modified Interest packet will be generated to let other nodes in a network know that entries of the accumulative Interest packet have been already full filled. So that whenever that Interest packet arrived on another node it didn't forward the content which is already delivered to the consumer node.
- Lines 34: This line will execute when there exists some content in node CS which is required by another node.
- Lines 37-41: These lines will execute when a node receives the same PIT entry again for some content. In this case, a node will update its already stored PIT entry and discard the new PIT request.

Algorithm 1 OnIncomingInterestPacket

```
     Data: Name, Satisfied Node ID (S_NID),
           ContentStore (CS), Nonce
(1)  bool packetUpdated = FALSE
(2)  if (this → Name != PITEntry) then
(3)      if (this → Content != CS) then
(4)          PIT.Insert(this → InterestPacket)
(5)          d(p, q) ← CalculateDistance(receiverNode(x,y),
               senderNode(x,y))
(6)          T_n ← (EIPS / d(p,q))
(7)          while (timer != FALSE) do
(8)              if
                   (NewInterestPacketArrived()
                   == TRUE) then
(9)                  DiscardInterestPacket(this →
                       InterestPacket)
(10)             else
(11)                 for (int i = 1; i <
                       this → InterestPacket.length;
                       i++) do
(12)                     if
                           ((InterestPacket.S_NID[i]
                           == NULL) &&
                           (InterestPacket.Name[i]
                           == this → CS)) then
(13)                         DataPacket[i] =
                               CreateDPEn-
                               try(InterestPacket.Name[i])
(14)                         InterestPacket.S_NID[i] =
                               this → nodeID
(15)                         packetUpdated=TRUE
(16)                     else
(17)                         DiscardInterestPacket()
(18)                         Stop broadcasting
(19)                         return
(20)                     end
(21)                 end
(22)             end
(23)             timer++
(24)         end
(25)         if (packetUpdated == TRUE) then
(26)             ForwardUpdatedInterestPacket()
                   SendDataPacket(this → DataPacket)
                   packetUpdated = FALSE
(27)         end
(28)     else
(29)         Forward Data;
(30)     end
(31) else
(32)     UpdatedPITEntry(this → InterestPacket)
(33)     DiscardInterestPacket(this → InterestPacket)
(34) end
```

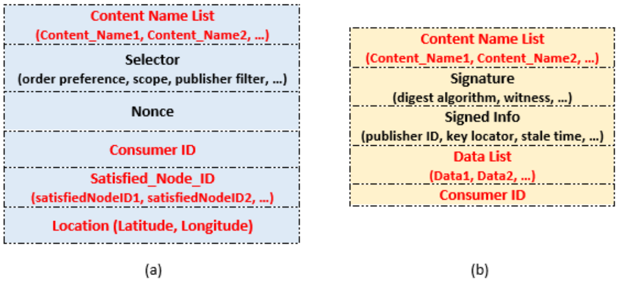


FIGURE 7. Packets format for the proposed protocol: (a) Interest packet and (b) Data packet.

E-DRAFT uses a unique timer function to forward Data and Interest packets as shown in line 6 in Algorithm 1. By using the timer function, we limit the Data and Interest packet flooding exponentially. Our timer function is based on two entities one is "Enhanced Interest Packet Size (EIPS)" and the other is the "Euclidean Distance ($d(\mathbf{p}, \mathbf{q})$)" value. The euclidean distance formula is shown in Equation 1 where P and Q are two WMSN nodes in Euclidean N-space. and q_i and p_i are the euclidean vectors.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{1}$$

Equation 2 shows the formula to calculate the timer value.

$$T_{node} = \frac{EIPS}{d(\mathbf{p}, \mathbf{q})} \tag{2}$$

EIPS is the exact size of the native NDN Interest packet along with some added fields which are required for E-DRAFT. Figure 7 shows the modified Interest and Data packet's format according to E-DRAFT.

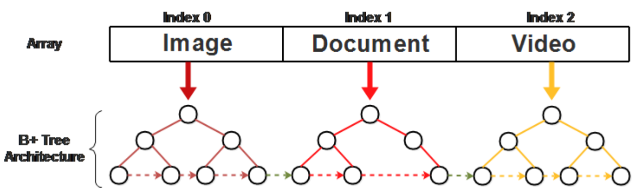


FIGURE 8. Proposed B+ Tree architecture.

C. PART II: EFFICIENT CONTENT-STORE ARCHITECTURE

WMSN node has short processing and battery life therefore efficient CS plays a critical part in a node. The CS mostly performs two functions (I) It looks for content for an existing packet (II) Or it stores new data in the cache. In NDN architecture CS algorithm is known as Skip-List which has the worst-case complexity of O(n). Due to its complexity Skip-List affect the efficiency of the CS and cause a delay in the network. To tackle such issues, we used an array-based B+ Tree architecture in a novel way to reduce the time complexity of content storage in worst-case scenarios. We assumed that Data packets that are received on nodes will be of three types named "Image", "Document", and "Video". These three types will be stored in a fixed-size array of 3. After that, each index of the array will have a specific tree and these trees will contain chunks of three files. The proposed B+ Tree architecture is shown in Figure 8. As seen in Figure 9, index 0 of the array contains an Image tag that's why chunks with the image tag will be present in the corresponding B+ Tree. Moreover, for other indexes, the same process will be followed. In B+ Tree architecture, there will be three types of nodes (root, inner, and leaf). A leaf or a node with two or more children can be the root. The B+ Tree is a B-Tree with an additional level at the bottom with connected leaves, with each node containing only keys (rather than key-value pairs as in the B-Tree). Data will only be stored at leaf nodes of the B+ Tree. In a block-oriented storage environment, the major utility of a B+ tree is to store data for efficient retrieval. B + Tree's core implementation hasn't changed. NDN names the Interest and Data packets using hierarchical variable-length nomenclature, which implies that Data packets for the same array index will have an identical array index prefix with the chunk name appended at the end. Image/chunk1 and Image/chunk2 are, for example, the names of two parts from the same array index, where Image/ is the array index and chunk1, and chunk2 are the chunk names. Keeping this in mind, when a Data packet is received, the array index and the chunk name are extracted. Let's assume a scenario where we have three multimedia file types named Image, Video, and Document. Suppose that the Image file has 5, the Video file has 6, and the Document file has 5 chunks, and we are using our proposed array-based B+ Tree for this scenario. Further, assume that Data packets are received at a node according to the order which is shown in Figure 9. Figure 10 shows the complete process of image chunks stored in the content store, as explained below:

Packet 1	Packet 2	Packet 3	Packet 4
Image/C1	Document/C2	Document/C5	Image/C5
Packet 5	**Packet 6**	**Packet 7**	**Packet 8**
Video/C4	Video/C1	Video/C5	Video/C6
Packet 9	**Packet 10**	**Packet 11**	**Packet 12**
Video/C2	Image/C3	Image/C2	Document/C1
Packet 13	**Packet 14**	**Packet 15**	**Packet 16**
Document/C4	Video/C3	Document/C3	Image/C4

FIGURE 9. Packet chunks receiving order.

- Step 1: When the first chunk C1 of the image file is received, a B+ Tree node will be created. If no other B+ Tree exists before for the image index in the array, then this will be the root node of the B+ Tree. If a B+ Tree already existed, then this chunk will be stored accordingly. After that, another chunk C5 will also be stored in the root node as a maximum of 2 chunks can be stored at one time.
- Step 2: When the chunk C3 of the image file is received on WMSN, step 2 will be executed. In step 2, the root node will be divided because of three chunks C1, C5, and C3. In B+ Tree every node can have a maximum capacity of 2 chunks. Upon this middle chunk which will be chunk C5 becomes the root node. Now values of C1

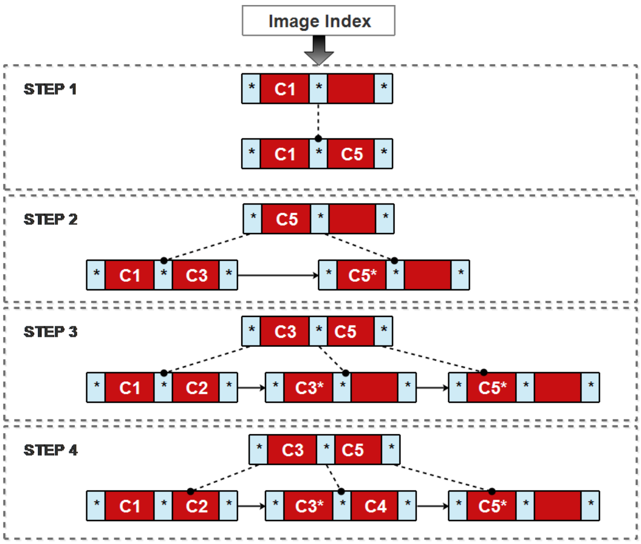


FIGURE 10. Image chunks receiving order.

and C3 are smaller than the root node so they become the left child of the root node. In the right child, the C5 pointer will be stored as in B+ Tree data will only exist in leaf nodes. Moreover, after this, both left and right children will be linked together via a pointer for faster access to data.

- Step 3: When the chunk C2 is received on WMSN, step 3 will be executed. As C2 is smaller than the root node so it will become the left node of the B+ Tree. On the other hand, a left node is already full. Now again left node will be divided and the middle chunk C3 will move to the root node as there exist a space for data. After this root node will contain C3 and C5. Again, the right child of C3 will contain the pointer of C3 which is the data of chunk 3.
- Step 4: When the last chunk of an image which is C4 received, step 4 will be executed. In step 4, C4 will be matched first with root node entries. C4 is bigger than C3 and smaller than C5 therefore C4 pointer will be added to the right child node of C3. This process will be followed every time a new chunk arrived for the image index. In last, every child node will be linked with another child node through a pointer for faster access.

Figure 11 shows the complete process of document chunks stored in the content store, as explained below:

- Step 1: When the first chunk C2 of the document file is received, a B+ Tree node will be created. If no other B+ Tree exists before for the document index in the array, then this will be the root node of the B+ Tree. If a B+ Tree already existed, then this chunk will be stored accordingly. After that, another chunk C5 will also be stored in the root node as a maximum of 2 chunks can be stored at one time.
- Step 2: When chunk C1 of the document file is received on WMSN, step 2 will be executed. In step 2, the root node will be divided because of three chunks C2, C5, and C1. As in B+ Tree, every node can have a maximum capacity of 2 chunks so the middle chunk which will be

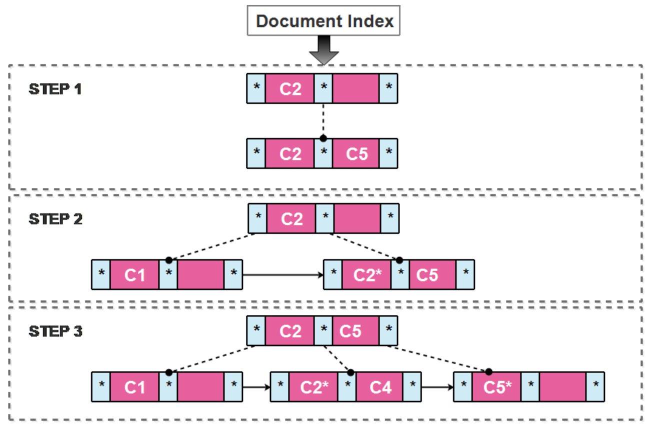


FIGURE 11. Document chunks receiving order.

chunk C2 becomes the root node. Now the value of C1 is smaller than the root node so it becomes the left child of the root node. While a right child of the root node will have a pointer of C2 and C5. In the right child, the C2 pointer will be stored in the leaf node as in B+ Tree data will only exist in leaf nodes. After this, both left and right children will be linked together via a pointer for faster access to data.

- Step 3: When the chunk C4 is received on WMSN, step 3 will be executed. As C4 is greater than the root node so it will go to the right node of the B+ Tree. But the right node is already full. So, the right node will be divided and C5 will move to the root node. Now, C4 will be added to the right node of C2, and C5's right child will contain a pointer of C5. In last, every child node will be linked with another child node through a pointer for faster access.

Figure 12 shows the complete process of video chunks stored in the content store, as explained below:

- Step 1: When the first chunk C4 of the video file is received, a B+ Tree node will be created. If no other B+ Tree exists before for the video index in the array, then this will be the root node of the B+ Tree. If a B+ Tree already existed, then this chunk will be stored accordingly. After that, another chunk C1 will also be stored in the root node as a maximum of 2 chunks can be stored at one time.
- Step 2: When the chunk C5 of the video file is received on WMSN, step 2 will be executed. In step 2, the root node will be divided because of three chunks C4, C1, and C5. As in B+ Tree, every node can have a maximum capacity of 2 chunks so the middle chunk which will be chunk C4 becomes the root node. Now the value of C1 is smaller than the root node so it becomes the left child of the root node. While the right child of the root node will have a pointer of C4 and C5. In the right child, the C4 pointer will be stored as in B+ Tree data will only exist in leaf nodes. Moreover, after this, both left and right children will be linked together via a pointer for faster access to data.

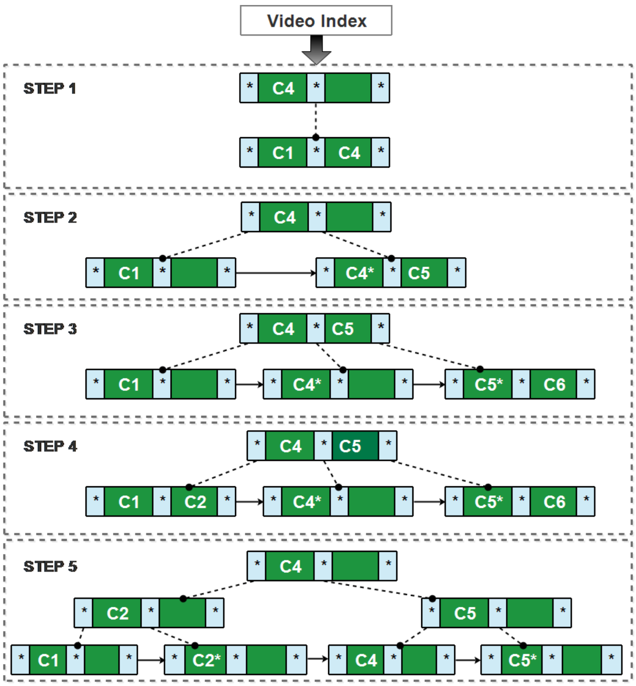


FIGURE 12. Video chunks receiving order.

- Step 3: When chunk C6 is received, step 4 will be executed. As C6 is greater than the root node then C6 will go to the right node of C5 the B+ Tree. But the right node is already full again. So, the right node will be divided and C5 will move to the root node. Now, C6 will be added to the right node of C5 which will now contain two chunks first the pointer of C5 and C6. While C4 right child will have a pointer of C4. In last, every child node will be linked with another child node through a pointer for faster access.
- Step 4: Now when chunk C2 is received, step 4 of B+ Tree will be executed. Chunk C2 is smaller than the root value then C2 will go to the left child of the root node.
- Step 5: Step 5 will execute when chunk C3 will be received. Now C3 has a smaller value than the root node so it will go to the left child of the root node. But the left child is already full so again we must divide the left child. When we divide the left child chunk C2 will move upward toward the root node. Here, the root node is already full. Now again root node will be divided and chunk C4 will become the new root node and its left child will have chunk C2. In the C2 node, a left child will contain C1, and a right child will contain a pointer to C2. While a right child of the root node will contain chunk C5. The C5 node has now again two children left and right. C5 left will have chunk C4 and the right will have a pointer to chunk C5. Again, every child node will be linked with another child node through a pointer for faster access.

In [30], two separate trees are created named as File-Tree and Chunk-Tree. If File-Tree has N nodes, then there will be M Chunk-Tree. Moreover, it is mandatory that for every node

TABLE 2. Simulation parameters.

Parameters	Value
Number of Sensor Nodes	102
Interest Packet Size	50 bytes
Data Packet Size	110 bytes
Node Initial Energy	0.5J
MAC Layer IEEE	802.15.4
Simulation Time	1000 secs
Network Field	100 m x 100 m
Number of Consumer Nodes	(1 to 5) out of 102
Number of Producer Nodes	(1 to 5) out of 102
Content-Store Size	1000 Packets
Tx Energy Consumption	50nJ/bit
Rx Energy Consumption	50nJ/bit

of the File Tree there must be a Chunk-Tree. So average and worse time complexity of [30] is shown in Equation 3.

$$O(\log(F)) + O(\log(C)) \tag{3}$$

In native NDN worse case time complexity is O(N). Our proposed array-based B+ Tree outperformed B-Tree-based architecture (3T architecture) [30], TEZEM [33], and native NDN (Skip-List) in terms of time and space complexity. According to the proposed scheme the time complexity of our scheme is shown in Equation 4.

$$O(1) + O(\log(N)) \tag{4}$$

Here, O (1) represents the array index's complexity which has a negligible impact on time complexity while O(log(N)) represents B+ Tree time complexity. In terms of space complexity, our proposed array-based B+ Tree architecture has O(n) complexity while B- Tree-based architecture [30] has O(F)*O(C).

V. PERFORMANCE EVALUATIONS

In this section, we performed several simulations to see how well our proposed system worked. TEZEM and 3T architecture are the two latest approaches that are closely related to our proposed scheme hence they are compared with E-DRAFT along with native NDN which is used for its simple structure and flooding. For simulation, network simulator 3 (NS3) and ndnSIM simulator are used. The simulation parameters used in this approach are listed in Table 2. We considered a grid topology of 6 * 17 as shown in Figure 13. A green node represents a consumer node while blue nodes represent producer nodes. The distance between each node is constant which is 85m while the energy level of each node is set to 50 joules. For mobility, we used the constant mobility model. Moreover, each node can cache up to 1000 packets and an LRU replacement strategy is used.

A. PERFORMANCE METRICS

For our approach, we consider those performance metrics which are related to cache. Detail of these metrics are given below:

- **Network Lifetime (NL)**: It is described as the time during which a network remains active. For WMSN, NL is very important. WMSN nodes have very limited energy. Therefore, it is necessary to use such nodes

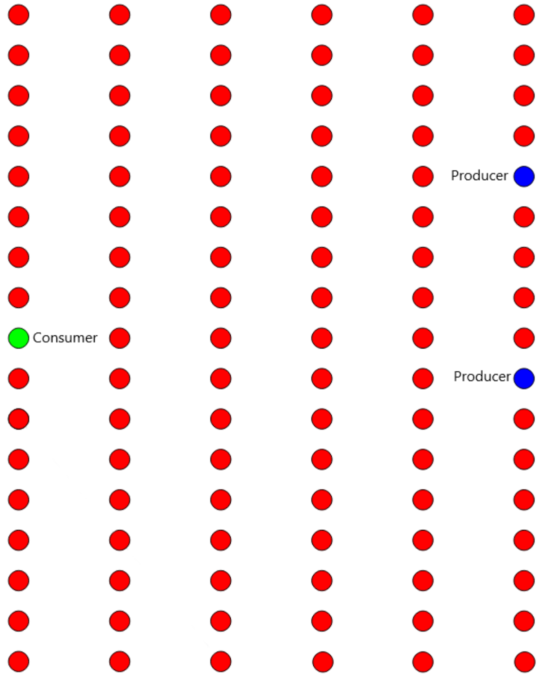


FIGURE 13. Network topology for the simulations.

which can provide better throughput. In WMSN, Interest transmission or receiving consumes more energy than packet processing. NL can be calculated using Equations 5 and 6.

$$T_i = \frac{E_0}{E_i} \tag{5}$$

$$NL = minT_i \tag{6}$$

Here, i shows any number of nodes i= $i_1, i_2, i_3, \ldots i_n$. Whereas E_0 represents initial energy of node and E_i represents energy consumption per unit time.

- **Average Total Number of Interest Packets (TNI)**: It is the average total number of Interest packets plus re-transmitted Interest packets that are generated to retrieve some Data packets by the consumer and intermediate node. It is calculated using Equation 7.

$$Avg.TNI = Interestpackets + Retransmittedpackets \tag{7}$$

- **Cache Search Time (CST)**: It is the time required to search a particular content in a CS.
- **Cache Insertion Time (CIT)**: It is the time required to save a data packet in CS. It is calculated as the difference between the time when the CS lookup function starts to insert data content and when the CS lookup function ends. Moreover, Equation 8 shows CIT.

$$CIT = CS\_StartLookup - CS\_EndLookup \tag{8}$$

B. RESULTS

This section presents results that are calculated by running each simulation exactly 20 times. Following are some

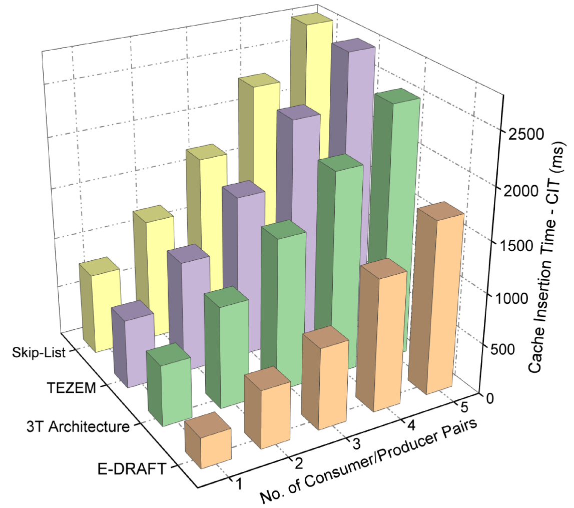


FIGURE 14. Cache insertion time concerning the number of consumer/producer pairs.

simulation results that are derived through the number of consumer/producer pairs and the number of packets.

- **Number of Consumer/Producer Pairs** Figure 14 depicts the CIT concerning the number of consumer/producer pairs. The graph shows that the proposed solution, E-DRAFT delivers better results as compared to the existing solutions, Skip-List, 3T Architecture, and TEZEM. This is all because, Skip-List uses the function O(n), which gives the same output as provided input with some tolerance. Therefore, the insertion process in the Skip-List will be getting slow with the increase in the number of consumer/producer pairs. Furthermore, 3T Architecture uses the B-Tree data structure, which gives the complexity for the insertion as O(log(N)) + O(log(N)). In addition, TEZEM uses the default architecture of the CS and does not deal with improving the CS architecture. In contrast, the E-DRAFT uses a B+ Tree data structure, which performs task logarithmically. Due to its logarithmic asymptotic time function, which is O(log(N)), it faces less time as compared to the Skip-List, 3T Architecture, and TEZEM.
  Figure 15 depicts the CST concerning the number of consumer/producer pairs. Furthermore, the graph shows that the proposed solution, E-DRAFT delivers better results as compared to the Skip-List, 3T Architecture, and TEZEM. The Skip-List performs the searching process under the function O(n), however, time complexity matters in the number of consumer/producer pairs and the time will be increased with the increase in the number of consumer/producer pairs, whereas the 3T Architecture uses a B-Tree data structure, where searching is performed under the O(log(N)) + O(log(N)). While TEZEM has not proposed any model to improve the architecture of the CS, because it uses default architecture. Furthermore, the proposed solution, E-DRAFT provides O(log(N)) complexity to search the

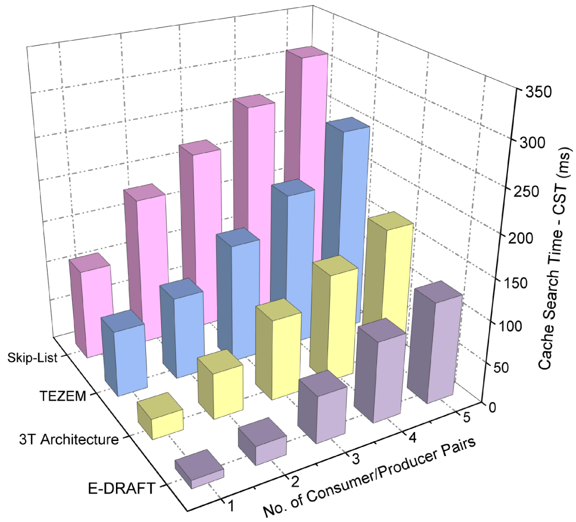


FIGURE 15. Cache search time concerning the number of consumer/producer pairs.

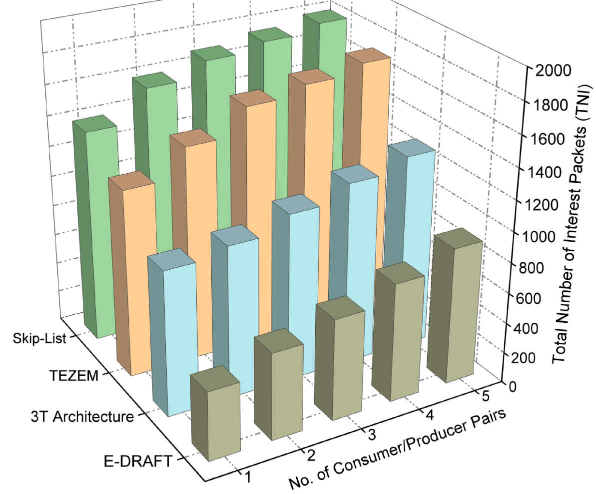


FIGURE 17. Total number of interest packets concerning the number of consumer/producer pairs.

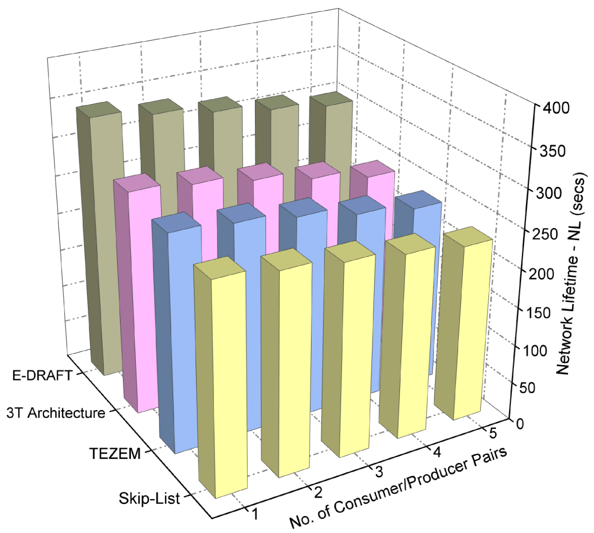


FIGURE 16. Network lifetime concerning the number of consumer/producer pairs.

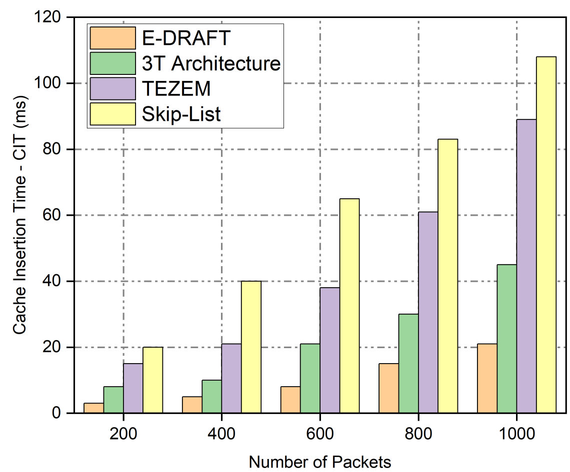


FIGURE 18. Cache insertion time concerning the number of packets.

content objects from the CS, as it uses the B+ Tree data structure. Therefore, it can be concluded that E-DRAFT has better performance results as compared to the Skip-List, 3T Architecture, and TEZEM as the number of packets increase in the network.

Figure 16 depicts the NL concerning the number of consumer/producer pairs. As E-DRAFT transmits a lower number of packets in the network, therefore it saves the energy of the nodes as well as increases the NL as compared to the all-existing compared solutions, such as Skip-List, 3T Architecture, and TEZEM. Moreover, it can be concluded that whenever consumers or producers increase in the network our proposed scheme deals with extra packets transmission better than other solutions which improves the network lifetime.

Figure 17 depicts the comparison of TNI flooded in the network concerning the number of consumer/producer pairs. The graph shows that the proposed solution, E-DRAFT delivers better results as compared to the Skip-List, 3T Architecture, and TEZEM. In E-DRAFT, multiple content requests are accumulated in one Interest packet, instead of separately. Furthermore, it also uses a timer to control the broadcast storm of Interest packets and Data packets. Besides, existing protocols do not control the flooding and broadcast storm of Interest packets and Data packets in this way. Hence, it can be concluded that E-DRAFT has better performance results as compared to Skip-List, 3T Architecture, and TEZEM.

- **Number of Packets**

  Figure 18 depicts the CIT concerning the number of packets. The graph shows that the proposed solution, E-DRAFT delivers better results as compared to the existing, Skip-List, 3T Architecture, and TEZEM. This is all because, Skip-List uses the function O(n), which gives the same output as provided input with some tolerance. Furthermore, 3T Architecture uses the

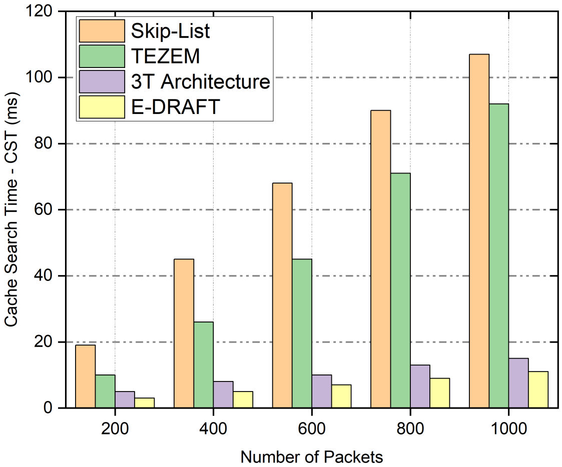


FIGURE 19. Cache search time concerning the number of packets.

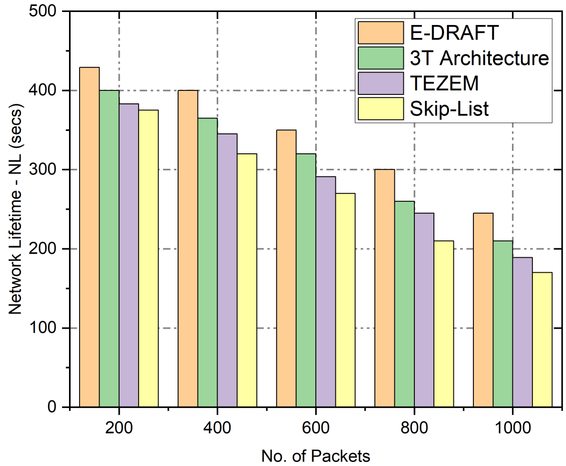


FIGURE 20. Network lifetime concerning the number of packets.

B-Tree data structure, which gives the complexity for the insertion as O(log(N)) + O(log(N)). In addition, TEZEM has not any mechanism related to the CS architecture, and further, it uses default architecture. In contrast, the E-DRAFT uses a B+ Tree data structure, which is considered logarithmically. Due to its logarithmic asymptotic time function, which is O(log(N)), it faces less time as compared to the Skip-List, 3T Architecture, and TEZEM. Figure 19 depicts the CST concerning the number of packets. Furthermore, the graph shows that the proposed solution, E-DRAFT delivers better results as compared to the Skip-List, 3T Architecture, and TEZEM. The Skip-List performs the searching process under the function O(n), whereas the 3T Architecture uses a B-Tree data structure, where searching is performed under the O(log(N)) + O(log(N)). While TEZEM has not proposed any model to improve the architecture of the CS, because it uses default architecture. Furthermore, the proposed solution, E-DRAFT provides O(log(N)) complexity to search the content objects from the CS, as it uses the B+ Tree data structure. Therefore, it can be concluded that E-DRAFT has better performance results as compared to the Skip-List, 3T Architecture, and TEZEM as the number of packets increase in the network. In the WMSNs, the nodes require a high level of energy to process the Interest packets and Data packets. Figure 20 depicts the NL concerning the number of packets. As E-DRAFT transmits a lower number of packets in the network, therefore it saves the energy of the nodes as well as increases the NL as compared to the 3T Architecture and TEZEM. Furthermore, Skip-List is just a data structure, and it does not have any mechanism related to improving the NL.

VI. DISCUSSION AND LIMITATIONS

In [30], authors introduced a novel approach for accumulative Interest packet forwarding in NDN-enabled WSN. Their approach solves the packet forwarding delay at the consumer by combining multiple Interest packets. Unfortunately, by doing this their scheme suffers when there exist multiple chunks of one accumulative Interest packet on multiple nodes. In that case, the accumulative Interest packet will be forwarded again even if some Interest packet chunks are satisfied. This packet re-transmission increases the number of duplicate packets both in form of Interest and Data packets. Also, it decreases network throughput. To solve these issues, we proposed a new lightweight scheme to tackle accumulative Interest packets along with modification of CS by using B+ Tree. Our scheme outperforms all other schemes in the context of accumulative Interest packet forwarding. We run simulations multiple times to get concrete results. But there exist some limitations and challenges in developing and implementing efficient Interest/Data packet flooding and content store management in an NDN network. These limitations are given below.

- Suitable Network Configuration: Network configuration or topology can impact the performance and accuracy of a packet flooding in the experiment. A small-scale network or a large-scale network with various topologies may be the focus of packet flooding. Among the topologies that researchers frequently employ to depict the situation in NDN architecture are general network topology, standard topology, ISP topology, tree topology, and customized topology. It is advised to set up a custom or ISP topology with enough nodes to cover the trial area.
- Standard Evaluation Framework: NDN is an emerging paradigm and is continuously developing. At this point, it is sufficient to identify the operations in the NDN architecture using host-specific simulators or testbeds. The challenge for researchers today, however, is the lack of a standardized and effective platform for evaluating NDN performance in terms of cache management. The simulator should be able to offer various configurations for various network environments of varying sizes, highlight faults for additional debugging analysis, and support event tracking and traffic tracing.

- Suitable Evaluation Metrics: Researchers have proposed a number of evaluation measures to evaluate the effectiveness of NDN packet flooding. The algorithm should determine which metrics to use. For instance, the simulator needs to be set up to capture particular parameters if the metric incorporates packet flooding. The effectiveness, performance, etc. of the experiments can all be improved with the use of these evaluation metrics. The network's time and resources will be used as little as possible by capturing the appropriate parameters.
- Compact System Design: In order to guarantee overall performance, a lightweight or compact mechanism is necessary. The design ought to be more straightforward, more cost-effective, and capable of delivering its best performance with complexity in O(n). In the default NDN routers, either at the prefix, interface, or router level, it must use little memory and have minimal leftover loads.

VII. CONCLUSION AND FUTURE WORK

The native NDN paradigm works on a single packet pattern and it uses a Skip-List data structure for caching the content. In this paper, we proposed a new architecture named E-DRAFT where we modified a one-Interest-one Data pattern by accumulating multiple Interest entries in a single request. E-DRAFT uses B+ Tree architecture to improve the efficiency of CS, especially packet insertion time, packet search time, and network lifetime. Moreover, NDN works on a broadcast nature and by implementing NDN in WMSN we also provide a mechanism to deal with the issue of flooding, where each node has multiple copies of the same Interest packet. Thus, extensive simulations are performed in ndnSIM to evaluate the proposed scheme. Our results show that the proposed scheme efficiently stores and retrieves packets with improved processing efficiency as compared to the various existing methods.

In the future, we will update CS replacement policies by involving machine learning models to better predict and improve network throughput.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2009, pp. 1–12.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless multimedia sensor networks: Applications and testbed," *Proc. IEEE*, vol. 96, no. 10, pp. 1588–1605, Oct. 2008.

[3] A. Amelia, M. Zarlis, S. Suherman, and S. Efendi, "Review on target tracking method in multimedia wireless sensor networks," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Aug. 2021, pp. 320–324, doi: 10.1109/ICSECS52883.2021.00065.

[4] (Oct. 20, 2008). *Kansas City Scout Intelligent Transportation System*, Kansas, MO, USA. [Online]. Available: http://www.roadtraffictechnology.com/projects/kansas/

[5] S. Chao, W. Ru-Chuan, H. Hai-Ping, and S. Li-Juan, "A type of healthcare system based on intelligent wireless sensor networks," *J. China Universities Posts Telecommun.*, vol. 17, pp. 30–39, Jul. 2010.

[6] J. Taysheng, C. Chien-Hsu, W. Jhing-Fa, W. Chung-Hsien, K. Jar-Ferr, C. Sheng-Tzong, F. Jing, and C. Pau-Choo, "House of the future relies on multimedia and wireless sensors," *Proc. SPIE*, p. 2, 2008.

[7] T. He, S. Krishnamurthy, and L. Luo, "VigilNet: An integrated sensor network system for energy-efficient surveillance," in *Proc. ACM Trans. Sensor Netw.*, 2006, pp. 1–35.

[8] *Cisco Visual Networking Index: Forecast and Methodology: 2013–2018*, CISCO, San Jose, CA, USA, 2018. [Online]. Available: https://www.anatel.org.mx/docs/interes/Cisco_VNI_Forecast_and_Methodology.pdf

[9] J. Lee and D. Kim, "Proxy-assisted content sharing using content-centric networking (CCN) for resource-limited mobile consumer devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 477–483, May 2011.

[10] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[11] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[12] A. Detti, D. Tassetto, N. B. Melazzi, and F. Fedi, "Exploiting content centric networking to develop topic-based, publish–subscribe MANET systems," *Ad Hoc Netw.*, vol. 24, pp. 115–133, Jan. 2015.

[13] W. M. H. Azamuddin, A. H. M. Aman, R. Hassan, and T.-A. N. Abdali, "Named data networking mobility: A survey," in *Emerging Technology Trends in Internet of Things and Computing* (Communications in Computer and Information Science), vol. 1548, P. Liatsis, A. Hussain, S. A. Mostafa, and D. Al-Jumeily, Eds. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-030-97255-4_20.

[14] M. Pawar and J. Agarwal, "A literature survey on security issues of wsn and different types of attacks in network," *Indian J. Comput. Sci. Eng.*, vol. 8, pp. 80–83, Jan. 2017.

[15] M. Miettinen and N. Asokan, "Ad-hoc key agreement: A brief history and the challenges ahead," *Comput. Commun.*, vol. 131, pp. 32–34, Oct. 2018.

[16] Z. Ren, M. A. Hail, and H. Hellbruck, "CCN-WSN—A lightweight, flexible content-centric networking protocol for wireless sensor networks," in *Proc. IEEE 8th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, Apr. 2013, pp. 123–128.

[17] S. Gao, H. Zhang, and B. Zhang, "Energy efficient interest forwarding in NDN-based wireless sensor networks," *Mobile Inf. Syst.*, vol. 2016, pp. 1–15, 2016.

[18] D. Freet and R. Agrawal, "An overview of architectural and security considerations for named data networking (NDN)," in *Proc. 8th Int. Conf. Manage. Digit. EcoSystems*, Nov. 2016, pp. 52–57.

[19] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Jan. 2013, pp. 1118–1122.

[20] M. Amadeo, A. Molinaro, and G. Ruggeri, "E-CHANET: Routing, forwarding and transport in information-centric multihop wireless networks," *Comput. Commun.*, vol. 36, no. 7, pp. 792–803, 2013.

[21] D. Kim and Y.-B. Ko, "A novel message broadcasting strategy for reliable content retrieval in multi-hop wireless content centric networks," in *Proc. 9th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2015, p. 108.

[22] Y. Lu, B. Zhou, L.-C. Tung, M. Gerla, A. Ramesh, and L. Nagaraja, "Energy-efficient content retrieval in mobile cloud," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput.*, Aug. 2013, pp. 21–26.

[23] R. A. Rehman, T. D. Hieu, H.-M. Bae, S.-H. Mah, and B.-S. Kim, "Robust and efficient multipath interest forwarding for NDN-based MANETs," in *Proc. 9th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Jul. 2016, pp. 187–192.

[24] R. A. Rehman and B.-S. Kim, "Energy aware forwarding in content centric based multihop wireless ad hoc networks," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E98.A, no. 12, pp. 2738–2742, 2015.

[25] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *J. Netw. Comput. Appl.*, vol. 50, pp. 148–158, Apr. 2015.

[26] D. Kim, J.-H. Kim, C. Moon, J. Choi, and I. Yeom, "Efficient content delivery in mobile ad-hoc networks using CCN," *Ad Hoc Netw.*, vol. 36, pp. 81–99, Jan. 2016.

[27] 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," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 187–192.

[28] H. Hao, C. Xu, S. Yang, J. Guan, Y. Liu, and L. Zhong, "BF-PDT: A new name lookup mechanism in content-centric networking," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Oct. 2017, pp. 69–74.

[29] S. H. Ahmed, S. H. Bouk, and D. Kim, *Content-Centric Networks: An Overview, Applications and Research Challenges*, 1st ed. Berlin, Germany: Springer, 2016.

[30] M. M. Qazi, R. A. Rehman, and B.-S. Kim, "Minimizing content-store data access time using two-tier tree architecture for NDN-based WSNs," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Jun. 2020, pp. 1–6, doi: 10.1109/WCNCW48565.2020.9124794.

[31] (Aug. 21, 2019). *Big O Notation*. [Online]. Available: https://medium.com/@kaushalmistry2611/big-o-notation-103432369f63

[32] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, "Choosing a design science research methodology," in *Proc. ACIS*, 2017, p. 112. [Online]. Available: https://aisel.aisnet.org/acis2017/112

[33] Z. U. A. Jaffri, M. Asif, W. U. Khan, Z. Ahmad, Z. U. A. Akhtar, K. Ullah, and M. S. Ali, "TEZEM: A new energy-efficient routing protocol for next-generation wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 18, no. 6, pp. 1–18, Jun. 2022, doi: 10.1177/15501329221107246.

[34] M. Anwar, F. Masud, R. A. Butt, S. M. Idrus, M. N. Ahmad, and M. Y. Bajuri, "Traffic priority-aware medical data dissemination scheme for IoT based WBASN healthcare applications," *Comput., Mater. Continua*, vol. 71, no. 3, pp. 4443–4456, 2022.

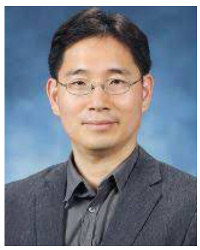
[35] M. Anwar, A. H. Abdullah, A. Altameem, K. N. Qureshi, F. Masud, M. Faheem, Y. Cao, and R. Kharel, "Green communication for wireless body area networks: Energy aware link efficient routing approach," *Sensors*, vol. 18, no. 10, p. 3237, 2018.

AHMAD ARSALAN received the M.S. degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree in computer science with COMSATS University, Lahore Campus, Pakistan. Besides this, he is working as a full-time Senior Lecturer with the Faculty of Information Technology and Computer Science, University of Central Punjab, Lahore, Pakistan. His research interests include the design and development of future internet architectures, cross-layer design for wireless networks, and efficient forwarding and routing protocols for named-data networking-based software-defined networks. He is also serving as a Reviewer for IEEE Access, *Future Generation Computer Systems* journal (Elsevier), and the *International Journal of Communication Systems* (Wiley).

MUHAMMAD BURHAN received the B.S. degree in information technology from University of the Punjab, Pakistan, in 2016, and the M.S. degree in computer science from the National University of Computer and Emerging Sciences, Pakistan, in 2019, under the supervision of Dr. Rana Asif Rehman. He is currently a Lecturer with the Department of Information Technology, University of the Punjab. His research interests include the design and development of efficient routing and forwarding protocols, wireless networks, named-data networks-based vehicular ad hoc networks, and software-defined networks.

RANA ASIF REHMAN (Senior Member, IEEE) received the M.Sc. degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 2010, the M.S. degree in computer science from International Islamic University, Islamabad, Pakistan, in 2012, and the Ph.D. degree in electronics and computer engineering from Hongik University, South Korea, under the supervision of Prof. Byung-Seo Kim, in 2016. In 2013, he was a Lecturer with the University of the Sargodha, Lahore Campus, Pakistan. From July 2016 to August 2021, he was an Assistant Professor with the Department of Computer Science, FAST-NUCES, Chiniot-Faisalabad Campus, Pakistan. After that, he joined FAST-NUCES, Lahore Campus, as an Assistant Professor, from September 2021 to June 2022. He is currently an Associate Professor with the FAST School of Computing, National University of Computer and Emerging Sciences, Lahore Campus. He is also a Cisco Certified Network Associate and a Microsoft Certified Professional. Moreover, he is also a HEC approved Ph.D. Supervisor in Pakistan. His research interests include the design and development of energy efficient routing protocols, cross-layer architectures, caching and forwarding for cognitive radio ad hoc networks, and named-data networking-based wireless networks. He is also a member of KSII, the IEEE Computer Society, the IEEE Communication Society, the IEEE Signal Processing Society, and the IEEE Young Professionals.

TARIQ UMER (Senior Member, IEEE) received the master's degree in computer science from Bahauudin Zakariya University, Multan, Pakistan, in 1997, and the Ph.D. degree in communication systems from the School of Computing and Communication, Lancaster University, U.K., in 2012. He worked in the IT Education Sector, Pakistan, for more than 20 years. In January 2007, he joined as an Assistant Professor with the Computer Science Department, COMSATS University Islamabad (CUI), Lahore Campus, Lahore. Currently, he is an Associate Professor (TTS) with CUI, Lahore Campus. His research interests include vehicular ad-hoc networks, the Internet of Things, digital twins, and telecommunication network design. He is an Active Member of the Pakistan Computer Society and the Internet Society Pakistan. He served as a TPC member for various international conferences. He is also an Editor of *Future Generation Computer Systems* (Elsevier) and an Associate Editor of IEEE Access. He is serving as a Reviewer for IEEE Communications Letters, *IEEE Wireless Communication Magazine*, *IEEE Network*, IEEE Access, *Computers and Electrical Engineering* (Elsevier), *Journal of Network and Computer Applications* (Elsevier), *Ad Hoc and Sensor Wireless Networks*, *Wireless Networks* (Springer), and MDPI journals.



BYUNG-SEO KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Y. Fang. From 1997 to 1999, he was with Motorola Korea Ltd., Paju-si, South Korea, as a Computer Integrated Manufacturing Engineer in advanced technology research and development. From January 2005 to August 2007, he was with Motorola Inc., Schaumburg, IL, USA, as a Senior Software Engineer in networks and enterprises, where his research focused on designing the protocol and network architecture of wireless broadband mission-critical communications. He is currently an Associate Professor with the Department of Computer and Information Communication Engineering, Hongik University, Seoul, South Korea. He was the Chairperson of the Department of Computer and Information Communication Engineering, from 2012 to 2014. His work has appeared in around 123 publications and he has 19 patents. His research interests include the design and development of efficient wireless/wired networks, including link-adaptable/cross-layer-based protocols, multiprotocol structures, wireless CCNs/NDNs, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He was a member of the Sejong-City Construction Review Committee and the Ansan-City Design Advisory Board. He was the Guest Editor of the *Journal of the Institute of Electronics and Information Engineers* and a Technical Program Committee Member for the IEEE VTC 2014-Spring Conference and the EAI FUTURE2016 Conference. He is also a Lead Guest Editor of the annual Special Issue of the *International Journal of Distributed Sensor Networks* (SAGE).