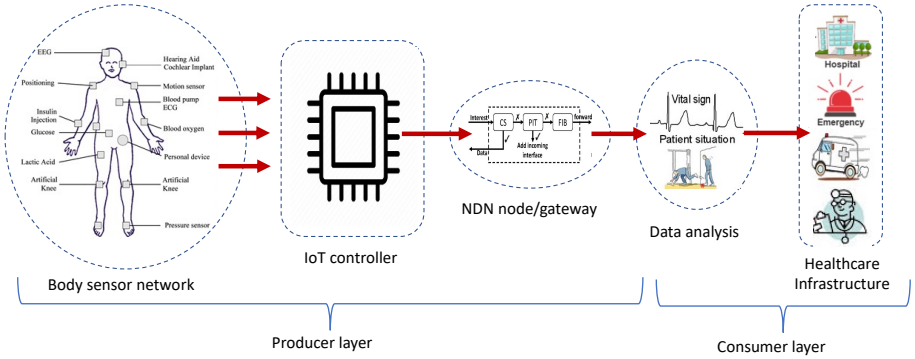
Graphical Abstract

Secure and Pervasive Communication Framework using Named Data Networking for Connected Healthcare

Rajan Kumar Dudeja, Rasmeet Singh Bali, Gagangeet Singh Aujla



Highlights

Secure and Pervasive Communication Framework using Named Data Networking for Connected Healthcare

Rajan Kumar Dudeja, Rasmeet Singh Bali, Gagangeet Singh Aujla

- A content name-based and in-network caching mechanisms of NDN are designed for suitability in effective forwarding of healthcare information generated through different IoT sensors.
- A framework for implementing pervasive healthcare communication is designed by utilizing NDN as a backbone network.
- A sensor registration and authentication scheme is proposed followed by a data monitoring system.
- An adaptive producer-consumer mechanism is proposed to handle data based on the severity level of sensed healthcare parameters.

Secure and Pervasive Communication Framework using Named Data Networking for Connected Healthcare

Rajan Kumar Dudeja[a], Rasmeet Singh Bali[a], Gagangeet Singh Aujla[b,*]

[a]*Computer Science & Engineering Department, Chandigarh University, Mohali, India*
[b]*Department of Computer Science, Durham University, Durham, United Kingdom*

Abstract

Connected healthcare system is one of the most critical Internet of Things (IoT) applications offering numerous healthcare services. But it requires the next-generation technologies to collude with the IoT standards to provide prompt end-user services. Thus, a data-centric network architecture like, Named Data Network (NDN) can be effective to tackle the concerns related to communication in the connected healthcare. In this article, we propose a Secure and Pervasive Health Care Framework (SecPHCF) that attempts to process healthcare data securely and transmit it using NDN thereby aiding faster healthcare support. The data dissemination model in SecPHCF contains two operating modes, a) a consumer-based model where the end-user initiates the data communication process and b) a producer-based model where alert messages indicating critical conditions trigger the transmission. SecPHCF is validated through extensive simulations on different network scenarios by considering evaluation parameters such as throughput, network delay, packet delivery ratio, and computation overheads.

Keywords: Internet of Things, Named Data Network, Healthcare, Body Sensor Network, Data Dissemination.

1. Introduction

Internet of Things (IoT) encompasses smart objects and devices (such as sensors, smartphones, and wearable) that are connected to the Internet and can eventually talk to each other. The IoT devices are connected to automated systems (like connected healthcare) wherein they gather (or sense) data, analyse (or process) this data and trigger (or create) an action (or decision) concerning a specific task or a process. The proliferation of increasingly small and affordable embedded devices has resulted in large scale applicability of IoT in numerous domains. However, enabling these devices with IoT functionality for efficient local and global communication has been a key challenge in the development of IoT-based computational models. Most of the currently

*Corresponding Author
Email addresses: cucse.rajan@gmail.com (Rajan Kumar Dudeja), rasmeetsbali@gmail.com (Rasmeet Singh Bali), gagi_aujla82@yahoo.com (Gagangeet Singh Aujla)

deployed IoT models are primarily focused on equipping the things with capability and capacity to share information and act autonomously to meet demands of different applications like agriculture, environmental monitoring, smart city, healthcare, and industrial applications [1]. Healthcare systems are one of the most sensitive IoT applications that can realize several aspects of healthcare spectrum such as real-time patient monitoring, indoor quality monitoring, ubiquitous and pervasive information access, eventually benefiting healthcare professionals and patients at the large. There are several IoT-based healthcare products ranging from a small smartwatch to big monitoring devices in hospitals that can be leveraged to realize the healthcare services. A small chip (or sensor) can fit into an IoT devices to collect patients' vital heath parameters (known as heath data). These devices can monitor and examine human health diagnoses based on various parameters such as blood pressure, pulse rate, leg movements, body temperature, cholesterol, uric acid, and urea. This vital data when combined with analytics can help to reveal meaningful patterns that can be useful for deciding long term (and short term) patient care and treatment [2].

However, the implementation of any IoT-based healthcare solution requires a secure and reliable communication backbone. Most of the existing frameworks fail to provide a satisfactory (or effective) solution for developing a pervasive healthcare system [3]. The key bottlenecks that have constrained the development in the field of connected healthcare include issues such as accessibility, portability, interoperability, information security, and privacy. Additionally, technical issues such as interconnection among heterogeneous devices, and computational (and energy) constraints must be handled efficiently to pave the road for the adoption of IoT for healthcare [4]. One of the most critical issues in the adoption of any connected healthcare system is the management of healthcare data. Any IoT-based healthcare system will generate a large amount of data (including sensitive data) related to human health and disease. This critical data needs to be filtered, processed, and stored, in a secure manner in the real-time. However, the low computational and storage capacity of IoT devices has limited the adoption of IoT-based healthcare solutions. Thus, this has necessitated the need of integrating new age computational and communication technologies to work alongside the current IoT standards and protocols [5].

Named Data Network (NDN) has emerged as a viable alternative to overcome the above-mentioned challenges and offer a promising solution to data-intensive Internet-based applications. Unlike traditional TCP/IP, NDN achieves data delivery based on producer and consumer. In the NDN scenario, the producer is a node that generates the data chunks, and a consumer is a node that requires the generated data by the producer. It is a pull-based model in which the consumer will demand data as per their requirement and it operates on the basis of '*what is required*' rather than '*where it is generated*' [6]. Every router of NDN maintains three data structures; i) Content Store (CS) is used for in-network caching to increasing the probability of sharing the content, ii) Pending Interest Table (PIT) stores all the pending interest packets forwarded by the router, and iii) Forwarding Information Base (FIB) stores the information of the next-hop relay nodes for which the reachable destination exists [7]. NDN supports in-networking caching used to store the data generated by the producer in the buffer space of each node termed as CS space. Some of the prominent in-networking caching techniques are leave copy everywhere (LCE), leave copy down (LCD), edge caching,

consumer caching, and probcache caching [8]. Apart from cache storage techniques, cache replacement strategy also plays a vital role in the overall performance of these networks. The major role of these caching policies is to store information that is required by the consumer [9]. In NDN, the applications can also define their naming schemes according to their requirement independent of the network. This resolves the problem of huge address space requirements of traditional IPV4 and IPV6-based Internet. It assigns Content-Name-Prefix to data chunks produced by the producer [10].

Looking into the above discussed benefits, NDN can a viable technology that can be integrated with the IoT-based pervasive healthcare systems for communication purposes. The schematic diagram of a generic NDN-based healthcare system is shown in Figure 1. In this system, the sensors or wearable attached to the human body (to monitor their vital parameters or regular activities) establish a network called Body Sensor Network (BSN). These sensors connected to an IoT controller that further establishes a connection with a gateway node. IoT controller is equipped with suitable processing and computational capabilities also. Now, the data is further passed for analysis and thereafter to heath monitoring center and finally reaches the health expert. In the defined scenario, the producer consists of sensors, IoT controllers, and gateway nodes to connect with NDN networks. There is basic decision-based computation performed on generated data for further action. The action performed at the producer end is based on the type of data generated by the sensors. The healthcare monitoring center, data analytics servers & tools, and healthcare experts (doctors) are considered as consumers.

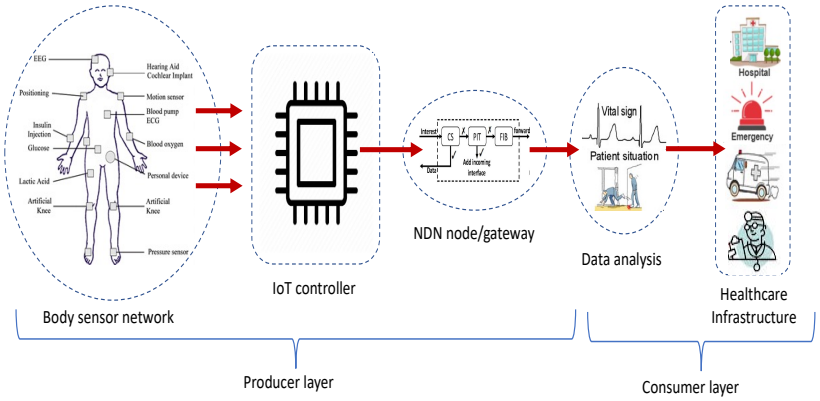


Figure 1: NDN-based connected healthcare system

1.1. Motivation and Research Questions

One of the most critical requirements for a healthcare monitoring system is to perform real-time processing and transmission of healthcare data generated from IoT devices and wearable. Although the processing capabilities of hardware devices have

exhibited an exponential increase, the communication systems still rely on legacy protocols that are not fully capable to provide the required scalability in their operations. This can act as a bottleneck for the applicability of the IoT-based communication framework in vital domains like healthcare. Named Data Networking of Things has emerged as a viable alternative to overcome these challenges due to its efficient data delivery mechanism as compared to TCP/IP-based communication framework [11]. Thus, it is utmost need of the hour to investigate the NDN-IoT-Healthcare coalesce to understand the applicability and various challenges that come across in a real system. Based upon this discussion, the following research questions (RQ) are very essential and must be answered.

- RQ1. How to design a pervasive data dissemination scheme for healthcare system supported by NDN?
- RQ 2. How to design a delay-sensitive mechanism to handle the real time healthcare data efficiently (in terms of performance) and securely (to avoid unauthorized access)?
- RQ 3. How to design an adaptive segregation and prioritization technique for critical healthcare data?

1.2. Research Contributions

The NDN-IoT-based healthcare framework is a promising solution for efficient data dissemination to answer the above defined research questions. This framework collects data from various deployed sensors on the human body, apply lightweight processes and caching at the producer plane, and achieve on-demand data delivery for consumers. To answer the above research questions, the research contributions of the paper are provided as below.

- A content name-based and in-network caching mechanisms of NDN are designed for suitability in effective forwarding of healthcare information generated through different IoT sensors.
- A framework for implementing pervasive healthcare communication is designed by utilizing NDN as a backbone network.
- A sensor registration and authentication scheme is proposed followed by a data monitoring system.
- An adaptive producer-consumer mechanism is proposed to handle data based on the severity level of sensed healthcare parameters.
- The effectiveness of the proposed data dissemination framework is validated based on extensive simulation results implemented on the ndnSIM simulator concerning packet rate, average delay, and throughput.

1.3. Organization of the article

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 presents the system model. The proposed scheme is outlined in Section 4. The evaluations and results are illustrated in Section 5 and the paper is concluded in Section 6.

2. Related Work

The researchers introduced numerous solutions to handle to aforementioned challenges. For example, Huo *et al.* [12] published a survey manuscript on the scope of Wireless Sensor Networks (WSNs)-based advancement for the medical assistance and healthcare fields. The ability to replace wired equipment with wireless ones in the hospital has a significant impact on the healthcare system. In similar manner, Zhang *et al.* [13] discussed a WSN-based multitier architecture for the healthcare system. The analysis of the IEEE 802 series standards in the access layer and their capacity to set up WSNs for healthcare has been performed. Tyagi *et al*. [14] discussed a cloud-based conceptual framework for the healthcare industry that implements standard IoT protocols for smart healthcare solutions. Selvaraj *et al.* [15] discussed challenges involved in IoT-based healthcare systems and identified the limitations in the existing systems use an increased number of devices such as power consumption, low availability of resources, and security issues. Saxena *et al.* [16] have discussed NDN for Internet paradigm to improve and simplify existing IoT communication issues. This study highlighted the benefits of content-centric approach that make NDN more suitable for an IoT-based machine-to-machine communication. The study proposed an NDN-based Smart Health IoT (NHealthIoT) system that integrates the features of NDN for IoT. Wang *et al.* [17] discussed the integration of IoT along with its edge computing capability to the NDN. Based on the idea, a framework called NDN-based IoTE (NIoTE) is proposed that allows IoT device to retrieve data efficiently from its nearest provider using request aggregation and in-networking cache.

Li *et al.* [18] proposed a three factor authentication scheme based on fuzzy commitment mechanism to handle user bio metric data. The results indicate suitability of this scheme for an IoT based environment. Li *et al.* [19] proposed a three-tier mutual authentication scheme among different mobile users and sensors with their controller nodes. The security of the proposed framework is analyzed by validating appropriate internet security protocols. Mick *et al.* [20] proposed a scalable lightweight authentication framework through hierarchical routing for NDN-IoT based scenario. The framework is validated based on extensive simulations that shows its effectiveness with respect to secure boarding and routing. Huang *et al.* [21] proposed data packet level authentication and security scheme using mobile edge computing scenario for NDN-IoT. The edge computing architecture is used to tackle the resource level issues. This scheme uses a certificate-less group authentication and batch verification that improves its performance in IoT based scenario. Based on the above review, a comparative analysis is provided in Table 1.

Table 1: Comparison of Existing Schemes

Proposed Framework	**1**	**2**	**3**	**4**
Survey on WSN based Healthcare [12]	✓	–	–	–
Multi-tier Healthcare Architecture [13]	✓	–	–	✓
Cloud-based Healthcare Conceptual framework [14]	–	✓	–	–
IoT-Healthcare challenges [15]	–	✓	–	–
NDN-based Smart Healthcare [16]	–	✓	✓	–
NDN-based edge integrated Smart Healthcare [17]	–	✓	✓	–
Three-factor authentication mechanism [18]	✓	✓	–	✓
Three-factor authentication for Mobile-IoT [19]	–	✓	–	✓
Scalable Framework for Lightweight Authentication for NDN-IoT [20]	–	✓	✓	✓
Certificate-less Group Authentication [21]	–	✓	✓	✓

1. WSN-BSN Integration 2. IoT Integration 3. NDN Integration 4. Security Concerns

3. Scheme Model

The heightened health risk for humans has contributed to an exponential increase in the number of healthcare monitoring applications based on IoT. This has led to availability of large amount of healthcare data that needs to be processed efficiently. The proposed Secure and Pervasive Healthcare Communication Framework for Named Data Networking of Things (SecPHCF) attempts to efficiently process the healthcare data and transfers it by using NDN thereby aiding in faster healthcare support. SecPHCF uses NDN as the basic communication framework and divides the system into three different planes named: producer plane, consumer plane and core NDN network plane. SecPHCF executes the data dissemination model in two different operating modes. The first mode is named as consumer-based data dissemination model where end user initiates the data communication process followed by forwarding of (Im) towards the producer and producer replies with the required data. In the second mode named as producer-based data dissemination model, alert messages indicating critical conditions will trigger the transmission and critical packet will be broadcast towards the consumer that further leads the end user to trigger the emergency actions accordingly. Figure 2 provides a schematic description of these planes and hybrid data dissemination model that are discussed below.

3.1. Producer Plane

The producer plane consists of sensor devices connected to IoT controller followed by a NDN gateway node. After successful registration and authentication of sensors, the sensor starts collecting the data on periodic basis and transmit the data to the IoT controller. The IoT controller executes a decision model that is used for categorization of the various healthcare packets. This categorization is done on basis of the severity level of the transmitted healthcare packets by various BSNs. In SecPHCF, the severity level for each type of packet has been categorized into three levels that have been

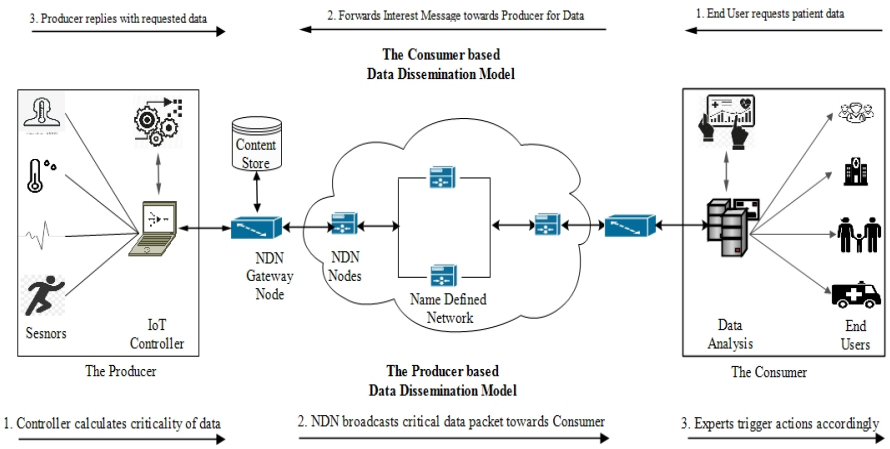


Figure 2: System Architecture for Hybrid Data Dissemination in SecPHCF

designated as *Normal*, *Mild* and *Critical*. These severity levels decide the priority of a packet and the storage and transmission of the packets is done according to the decided priority . The evaluation of these levels is based on the range falling between constant values $\alpha, \beta, \gamma, \delta$ that indicate the threshold values for a particular healthcare measurement [22]. The value of these threshold constants is assigned based on standard statistical inferences for each parameter being sensed. Equation 1 is used for segregating the severity level following the above constraints.

$$f(x) = \begin{cases} L_1, & (x < \beta) \cap (x > \alpha) \\ L_2, & ((x \leq \gamma) \cap (x \geq \beta)) \cup ((x \leq \alpha) \cap (x \geq \delta)) \\ L_3, & (x > \gamma) \cup (x < \delta) \end{cases} \tag{1}$$

where, function $f(x)$ contains the computed severity level of healthcare data produced by sensors based on standard constants. As shown in Table 2, L_1 indicates *Normal Condition* to signify that a particular user has normal readings. The severity condition L_2 indicates *Mild Condition*, that will leads to precautionary measures being recommended from healthcare experts. The level L_3 indicates *Critical Condition*, which may need immediate medical care.

Table 2: Symbols indicating Severity Level

Severity level	**Action performed**
L_1 (Normal)	Data Stored for further clinical purpose
L_2 (Mild)	Precautionary Measures will be advised
L_3(Critical)	Immediate Action is triggered

The IoT controller is further connected to NDN gateway node that is third component of the producer plane. The gateway node is the interface that provides connectivity

between the core IoT to NDN network. The gateway node acts as an interface as well as considered as a part of producer plane in SecPHCF to buffer the data packets received from the controller. Although, it is an interface, but gateway node has the capacity to buffer the data packets received from the controller. Another important functionality of gateway node is to assign *Content-Name-Prefix* to the data chunks produced by the sensors using conventional NDN naming scheme.

The packet storage eligibility depends on the value of $f(x)$ and accordingly the priority of the packet is decided. If value of $f(x)$ is L_1 or L_2, then storage eligibility will be finalized. Afterwards, the SecPHCF treats L_1 and L_2 as least significant packets. The L_1 and L_2 packets are buffered only if additional storage space is available at gateway node. However, if $f(x)$ declared packet as level L_3, it is considered as a critical level health packet and forwarded on priority by gateway node. Thus, the transmission mechanism followed in above message transmissions is a producer-based data dissemination model that is triggered based on critical value of healthcare information received by gateway node of producer plane.

3.2. Consumer Plane

Consumer plane consists of data analysis units, monitoring applications and end users. Data analysis units are used to store and analyze the healthcare packets received from the producer plane. The analysis units have enough computational power to store and analyze large amount of data that can be used for further clinical purpose. Monitoring applications are deployed on data analysis units to prioritize the received healthcare packets. The analyzed information is forwarded to end user that is healthcare experts or medical professionals for further decision making. The end user then recommends requisite actions based on the received information. Whenever the healthcare expert wants to monitor the patient in producer plane, the consumer plane triggers a NDN event using the consumer-based data dissemination model. This is done by generating (Im) that is then forwarded to producer plane through NDN network. When this (Im) reaches the producer plane, a data packet is sent back containing the required Data Packet (Dp). Another important function performed by the consumer plane is to trigger the action to be performed based on the severity level of the packet received through both data dissemination model.

3.3. NDN Network Plane

NDN Network plane consists of core NDN nodes that will transmit the packets between the producer and consumer plane. Both the data packet types, i.e., Im and Dp are used for transmitting the healthcare information in the NDN network. Im is used for generating the requirements of the specific healthcare data item from the consumer plane. Whereas Dp carries the actual payload generated by the producer. Figure 3 depicts the packet formats for Im and Dp used in SecPHCF. The main fields in Im are content name, that defines the content name prefix for which the interest has been generated. Other than this, it contains information related to selector which defines the order preferences, publisher filter and scope of Dp. It consists of lifetime indicator that works to find the remaining period of Im in network. The Dp consists of content name that defines its producer information and meta information related to the content type

and content freshness. It also consists of security information generated by signature mechanism such as publisher ID, key locator, etc. It consists of an actual payload field which consists of generated healthcare packet by the sensors. There is also customization to basic Dp, by adding a special sequence number 0 to it, that turns it into the alert message having priority data. This type of Dp is generated during producer-based data dissemination model [23].

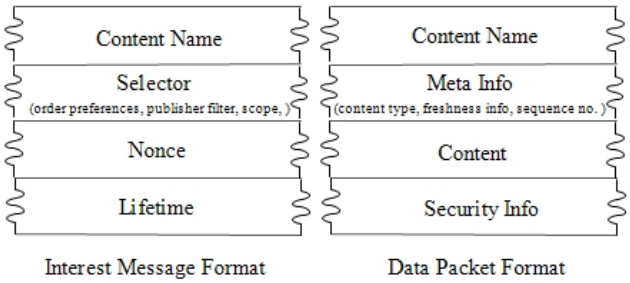


Figure 3: Packet Format in SecPHCF

An NDN node consists of PIT and FIB where PIT table stores the information about the pending requests of each Dp on that node and FIB stores the information about the neighbour's and immediate nodes falling in path towards destination. When consumer generates Im, it reaches to NDN Node, then it checks for data availability and act accordingly. Now, if the data is available then it responded with Dp, otherwise forwards Im towards producer based on information stored in FIB and creates entry in PIT. As Im reaches the producer, it responds with Dp and travels towards consumer based on entry created in PIT by intermediate nodes. The nodes in network that forwards the Dp will clear the entry for that packet in PIT. In SecPHCF, LCD approach is used to store the data generated by producers on the Gateway Node ($Node_g$) only. The Dp required for any Im generated by consumer is only available at $Node_g$ itself. The above discussed scenario is applicable during the consumer-based data dissemination model. In producer-based data dissemination model, $Node_g$ receive the Dp of critical severity level. In response, it suspends the PIT and broadcast the Dp with special sequence number to all its neighbors. All NDN nodes receiving such Dp will follow the same procedure. This high priority Dp will reache the healthcare experts to initiate immediate action.

4. Proposed Scheme

The proposed scheme consists of three subsystems named as Sensor Registration and Authentication System (SRAS), Sensor based Data Monitoring System (SDMS) and NDN-based Network Delivery System (NNDS). The SRAS elaborates the registration and authentication process of sensors with controller. SDMS contains description about complete data monitoring mechanism for producer and consumer planes. On the other hand, in NNDS, the forwarding strategy used for transmitting data packets on NDN Nodes is proposed. SecPHCF assumes that there are *n* users that are registered and each user can be provided up to *m* sensors. It assigns ID_s and ID_c to individual

sensors and its IoT controller. Initially, the sensors must register with controller and completes authentication process. The time stamps $T_1, T_1^{'}, T_2, T_2^{'}$ and random numbers N_1, N_2, N_3, N_x, N_y are used during the registration and authentication process. After the registration and authentication process, each sensor $S_{i,j}$ will monitor a specific health condition. These sensors will periodically transmit the sensed data where $t_{i,j}$ indicates the time duration between two successive data transmission of the j^{th} sensor on the i^{th} user. IoT controller ($Ctrlr_i$) then process the data. Table 3 contains the definitions of various procedures for implementing the elementary functionality of the network. These procedures (named as Transmit(), Forward(), Broadcast(), Create() and Receive()) act as the basic networking functions. Additionally, the table also contains some additional procedures such Insert(), Suspend(), Store(), Update(), Assign(), Find() and Initialize() that implement the functions that are specifically related to NDN. All these procedures have been assumed to be overloaded to enable the system to invoke them with different parameters. Table 3 also provides a description about the different symbols used in proposed scheme. The section below contains description about the implementation of the subsystems defined in SecPHCF.

Table 3: Symbols and Functions used

Symbol	**Description**
ID_s	Sensor ID
ID_c	IoT Controller ID
SID_s	Shadow ID for Sensor
K_s	Key for Sensors
$T_1, T_1^{'}, T_2, T_2^{'}$	Time stamps
N_1, N_2, N_3, N_x, N_y	Random numbers
n	Number of users
m	Number of sensors for individual user
Im	Interest message
Dp	Data packet
$S_{i,j}$	j^{th} Sensor installed at i^{th} user
$t_{i,j}$	Access time interval of j^{th} sensor of i^{th} user
α, β, γ, δ	Standard healthcare measurement constant
Sl_1, Sl_2, Sl_3	Severity level's of Dp
Dp^{l-1}	Level-1 Dp
Dp^{l-2}	Level-2 Dp
Dp^{l-3}	Level-3 Dp
$Node_c$	Consumer node
$Node_p$	Producer node
$Node_g$	Gateway node
$Node_n$	Intermediate NDN node
$Ctrlr_i$	IoT controller node also represented by ID_c
$Node_n^{PIT}$	Pending Interest Table at $Node_n$
EL	Eligibility Level of Dp to store at cache
θ	Eligibility threshold value

FIB	Forward Information Database table
OC	Cache occupation Level of node
Transmit()	Transmission of packet from source
Forward()	Forwards the received packet
Broadcast()	Broadcast the packet to its neighbors
Create()	Creation of packet at source
Recieve()	Packet received at destination
Insert()	Insertion of entry in table
Suspend()	Suspension of PIT table at $Node_n$
Store()	Store the Dp in cache at $Node_g$
Update()	Update the stored Dp in cache at $Node_g$
Assign()	Mapping of $Content - Name - Prefix$ to $S_{i,j}^{th}$ stream
Find()	Find $S_{i,j}^{th}$ stream's $Dp_{i,j}$ for $Im_{i,j}$
Initilize()	Initialization of constant's and timing interval related to $S_{i,j}$

4.1. Sensor Registration and Authentication System (SRAS)

SecPHCF initialization process begins when sensors are turned ON. Thereafter, the sensor registration and mutual authentication with the respective IoT controller is performed. The medium channel used to communicate between sensor device and its controller could be assumed as a secure. The registration process among sensor and controller is depicted in Figure 4.

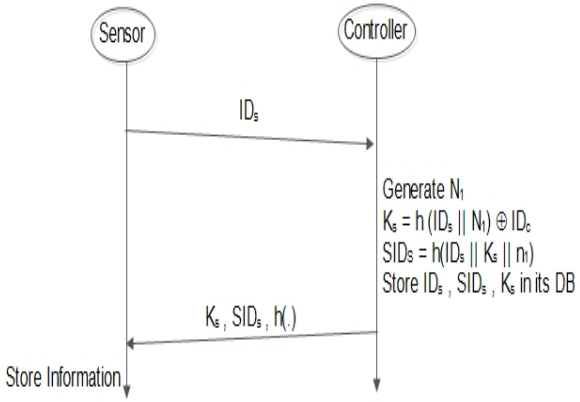


Figure 4: Sensor Registration for SecPHCF

Each sensor device is identified through a unique $48\ bits$ identification called ID_s which is its MAC address. The process is initiated when sensors transfer their ID_s to its IoT controller represented by ID_c of $48\ bits$. On receiving the ID_s, the controller generates a random number N_1 of $32\ bits$ and computes $K_s = h(ID_s \mid\mid N_1) \oplus ID_c$ where K_s is key for a ID_s, and the symbol $\mid\mid$ represents concatenation operator. The $h(.)$ is the one-way hash function used for generating hashes. Subsequently, the hash function is used by the controller to generate the shadow-id value denoted as SID_s

for a sensor ID_s using the equation $SID_s = h(ID_s \mid\mid K_s \mid\mid N_1)$. The controller stores the ID_s, SID_s *and* K_s in its database and transfers the message containing K_s, SID_s, *and* $h(.)$ to the sensor. The sensor stores this information into its database and registration is complete.

Figure 5 describes the detailed steps used in authentication mechanism between sensor device ID_s and its controller ID_c. It starts with sensor device fetching the SID_s and K_s from its database and generating a time stamp T_1. Then it generates the random number N_2 and computes the $S_1 = h(SID_s \mid\mid K_s \mid\mid N_2 \mid\mid T_1)$ and N_x performing $\oplus$ operation on K_s & N_2. After this, the value of S_2 is computed by performing $\oplus$ on BSN_id and $h(\ k_s \mid\mid N_2 \mid\mid T_1)$. The BSN_id represents the id of BSN that consists of various sensors. Now the message comprising S_1, S_2, N_x and T_1 for authentication request is transmitted to controller.

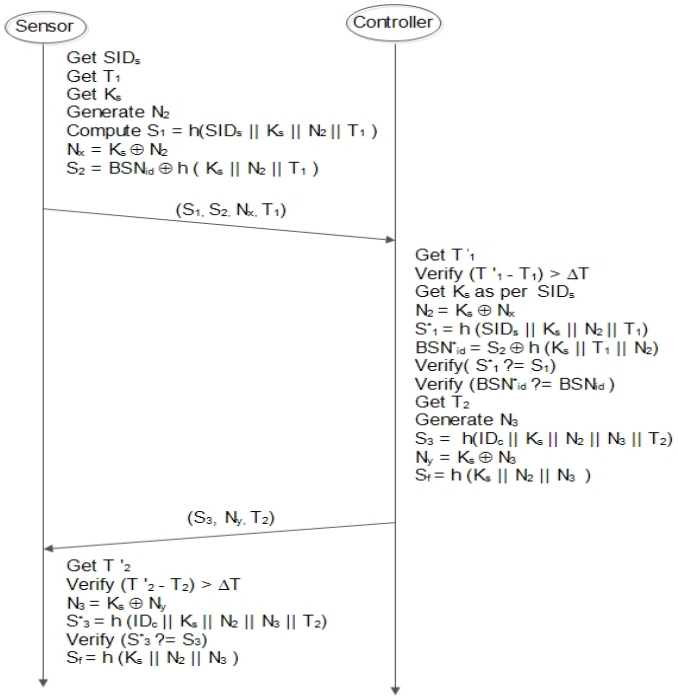


Figure 5: Sensor authentication for SecPHCF

When the authentication request is received at time T_1', the controller first validates the T_1 by computing the difference between T_1' and T_1. If the difference is higher than allowed communication delay represented by ΔT, the controller aborts the authentication process. After validation of T_1, controller retrieves the K_s and SID_s from its database and calculates the N_2 by performing $\oplus$ operation on K_s and N_x. After this, the values of $S_1^* = h(SID_s \mid\mid K_s \mid\mid N_2 \mid\mid T_1)$ and $BSN_i^*d = S_2 \oplus h(K_s \mid\mid T_1 \mid\mid N_2)$ are computed. The controller compares the value of S_1 with S_1^* and value of S_2 with

S_2^* to verify the authenticity of sensor device. After verifying the authenticity of the sensor device, the controller gets its current time stamp T_2 and then generates another random number N_3. These parameters are used for computing S_3 using the equation $S_3 = h(ID_c \mid\mid K_s \mid\mid N_2 \mid\mid \mid\mid N_3 \mid\mid T_2)$. Subsequently, N_y is computed using $\oplus$ operation on K_s and N_3. The message containing S_3, N_y and T_2 is then transmitted to sensor device. Finally, the controller computes the session key $S_f = h(K_s \mid\mid N_2 \mid\mid N_3)$.

Upon receiving the response message at times T_2', the sensor device validates the T_2. If $(T_2' - T_2 > \Delta T)$ then the sensor device dismisses the response message whereas ΔT represents the allowed communication delay. After validation of T_2, the sensor device calculates the value of N_3 by performing $\oplus$ operation on K_s and N_y. The value of $S_3^* = h(ID_c \mid\mid K_s \mid\mid N_2 \mid\mid \mid\mid N_3 \mid\mid T_2)$ is calculated and compared with S_3 to verify the authenticity of controller. Finally, the sensor device computes the session key $S_f = h(K_s \mid\mid N_2 \mid\mid N_3)$, that can be used as encryption key by sensor device to send the confidential data apart from ensuring message authenticity.

Algorithm 1 describes the process of registration and initialization of all the sensors $S_{i,j}$ represented by their respective ID_s linked to its respective IoT controller $Ctrlr_i$ represented by the ID_c. The process is initiated whenever a sensor $S_{i,j}$ is turned on and it transmits a hello packet consisting of its unique ID_s to indicate the activation. This is followed by the registration and authentication process elaborated above. After successful authentication of $S_{i,j}$, the $Ctrlr_n$ insert the details about sensor connectivity interface in its table. It initiates the values for different parameters for sensor $S_{i,j}$ like α, β, γ and δ that are specific to the healthcare condition being monitored. It initiates the $t_{i,j}$ which is data generation interval for $S_{i,j}$. The $Ctrlr_i$ also forward the credentials of the $S_{i,j}$ to the $Node_g$, that will assign a $Content - Name - Prefix$ to the data stream generated by the $S_{i,j}$.

Algorithm 1 Sensor Initialization

INPUT: $Node_g$, $Ctrlr_n$, $Sensor_{i,j}$, n, m
OUTPUT: $Sensor - Activation$

```
Begin:
for (i = 0; i ≤ n; i++) do
    for (j = 0; j ≤ m; j++) do
        if (Sensor_{i,j} == True ) then
            Transmit(Hello_{i,j}; (Sensor_{i,j} , ID_s) ; ( Ctrlr_n , ID_c))
            Initialize Registration Process
            Initialize Authentication Process
            Insert(Ctrlr_n, Interface-Table)
            Initialize (t_{i,j})
            Initialize ((α, β, γ, δ), Sensor_{i,j})
            Forward (Credentials, Ctrlr_n, Node_g)
            Assign ((Content-Name-Prefix)_{i,j}, Sensor_{i,j}, Node_g)
        else
            return FALSE
        end if
        Initialize (θ, Node_g)
    end for
end for
```

4.2. Sensor Data Monitoring System (SDMS)

SDMS subsystem is primarily responsible for management of the data generated by different components of SecPHCF. It is divided into two sub modules where the first sub module performs monitoring of data generated by sensors on the producer plane and the other module handles decision making concerning the action to be performed on the healthcare data at the consumer plane. After completing the sensor registration and authentication, the Algorithm 2 is triggered for the monitoring of the data generated by medical sensors.

Algorithm 2 Sensor Data Monitoring System (SDMS)

Input: $Vital\_Signs, n, m, t_{i,j}$
Output: $Alert\_Messages$
Initialization: $Current\_time = 0$

Begin:
while (1) **do**
For Producer's:
  for $(i = 1; i \leq n; i++)$ **do**
    for $(j = 1; j \leq m; j++)$ **do**
      if ($Current\_time$ *mod* $t_{i,j}$)== *0* **then**
        Create $(Dp_{i,j})$
        $Flag_{i,j} == True$
      end if
      if $(Flag_{i,j} == True)$ **then**
        if $((Dp_{i,j} < \beta_{i,j}) \cap (Dp_{i,j} > \alpha_{i,j}))$ **then**
          Set $Dp_{i,j} \rightarrow Dp_{i,j}^{l-1}$
        else if $((Dp_{i,j} > \gamma_{i,j}) \cup (Dp_{i,j} < \delta_{i,j}))$ **then**
          Set $Dp_{i,j} \rightarrow Dp_{i,j}^{l-3}$
        else
          Set $Dp_{i,j} \rightarrow Dp_{i,j}^{l-2}$.
        end if
        Call Algo. 3
        $Flag_{i,j} == False$
      end if
    end for
  end for
For Consumer's:
  for $(i = 1; i \leq n; i++)$ **do**
    for $(j = 1; j \leq m; j++)$ **do**
      if $(Current\_time \,\%\, t_{i,j} == 0)$ **then**
        Transmit($Im_{i,j}, Node_c, Node_p$)
        Call Algo. 3
        Receive($Dp_{i,j}, Node_p, Node_c$)
        if $(Dp_{i,j} == Dp_{i,j}^{l-3})$ **then**
          Call Emergency Services.
        else if $(Dp_{i,j} == Dp_{i,j}^{l-2})$ **then**
          Send Precautionary Measures to $User_i$
        else
          Data Stored for Further Clinical Process.
        end if
      end if
    end for
  end for
  Update $Current\_time$
end while

In Algorithm 2, at producer end, $S_{i,j}$ creates $Dp_{i,j}$ after a time interval ($t_{i,j}$) and forwards it to $Ctrlr_i$ for further processing. The severity level for every $Dp_{i,j}$ is computed based on process defined in previous sections. The processed $Dp_{i,j}$ along

with its $Sl_{i,j}$ value is forwarded to $Node_g$ for filtering based on the priority of healthcare packet $DP^l_{i,j}$. At the consumer end, *Data-mgt-sys* analyses the severity level of received $Dp^l_{i,j}$ and recommends further action based on Sl. It also generates $Im_{i,j}$ when any consumer plane entity requires $Dp_{i,j}$ for particular user. The most recent $Dp_{i,j}$ value present in the content store at producer end will be transmitted back as a response to $Im_{i,j}$.

4.3. NDN based Network Delivery System (NNDS)

NNDS is used for providing efficient and pervasive data delivery of healthcare packets by utilizes the standard NDN data delivery mechanism for data forwarding. NNDS also exploits the cache management functionality of NDN which helps in providing the consumer plane entities with appropriate and required healthcare data packets. Algorithm 3 explains the process followed for forwarding Im and Dp between consumer and producer planes using NNDS. It works in two independent phases. In the first phase, cache management of Dp is performed at $Node_g$. SecPHCF uses SL of Dp to determine the cache storage policy to be used for concerned Dp. In NNDS, the function to determine the cache storage eligibility is defined below.

$$EL_{i,j} = (C - L_{i,j}) + W_{i,j} \tag{2}$$

where, $El_{i,j}$ indicates the eligibility level for a $Dp_{i,j}$, C is constant that depends on number of severity levels of healthcare packets (3 in SecPHCF) and $L_{i,j}$ has assigned constant values of 1, 2 and 3 for L_1, L_2 and L_3 packets, respectively. $W_{i,j}$ is weight of $DP_{i,j}$ that is calculated based on $t_{i,j}$. The value of $W_{i,j}$ is inversely proportional to $t_{i,j}$ and packet with higher generation rate have lower value of $W_{i,j}$ and vice versa.

The value of $El_{i,j}$ indicates the priority of a particular $Dp_{i,j}$ for storage. Lower the value of $El_{i,j}$, higher will be possibility of that data packet being cached. If any $Dp_{i,j}$ has severity level of L_3, it will be directly cached as value of EL for such packet is $\leq \Theta$. Once any $Dp_{i,j}$ is found eligible for storage, the algorithm also validates whether an older version of $Dp_{i,j}$ is present in the cache or not. If it is available, then the older $Dp_{i,j}$ is replaced with its updated copy. However, if older version is not available in CS, the occupancy level of CS is checked. If space is available then $Dp_{i,j}$ is directly cached otherwise it replaces some other data packet using LRU replacement policy.

The second part of Algorithm 3 focus on the packet forwarding strategy at $Node_n$. This part of mechanism is called when $Node_c$ generates $Im_{i,j}$ for requirement of $Dp_{i,j}$ from $Node_p$. The standard NDN procedure is used in this case which allows to maintain PIT for pending interests. The $Im_{i,j}$ forwards to next node ($Node_{n+1}$) on the path of $Node_p$ based on data stored in FIB. As $Im_{i,j}$ reaches the $Node_p$, the required $Dp_{i,j}$ is transmitted towards the $Node_c$ using reverse process based on created log entries on $Node_n$. Now, when $Node_p$ generates $Dp^{L-3}_{i,j}$, it suspends the PIT and broadcasts customized $Dp_{i,j}$ with special sequence number 0 to all its neighbors. The $Node_{n+1}$ which receives the $Dp_{i,j}$ with special sequence number follows the same process till $Dp_{i,j}$ reaches its destination.

Algorithm 3 NDN based Network Delivery System (NNDS)

INPUT: $Im_{i,j}$, $Dp_{i,j}$, $L_{i,j}$

Begin:
while (**do**1)
Cache Management at NDN Gateway Node
  if ($Dp_{i,j} == True$) **then**
    if ($L_{i,j} == L_1 || L_2$) **then**
      Calculate $EL_{i,j} = (C - L_{i,j}) + W_{i,j}$.
      if ($EL_{i,j} \leq \Theta$) **then**
        Eligible for Cache Storage
        Set $Flag_{i,j} = 1$.
      else if ($L_{i,j} == L_3$) **then**
        Eligible fo Cache Storage
        Transmit ($Dp_{i,j}$, N_c)
        Set $Flag_{i,j} = 1$.
      end if
    end if
    if ($Flag_{i,j} = True$) **then**
      if ($Dp_{i,j}$(*old*) $\rightarrow Available$) **then**
        Update ($Dp_{i,j}$(*old*), $Cache$)
      else if ($OC_n \leq 1$) **then**
        Store ($Dp_{i,j}$, $Cache$)
      else
        Call LRU Replacement Strategy
        Store ($Dp_{i,j}$, $Cache$)
      end if
    end if
    Set ($Flag_{i,j} = False$)
  end if
Forwarding Strategy at NDN Intermediate Node
  if ($Node_c$ Generates $Im_{i,j}$) **then**
    while ($Im_{i,j}$! $\rightarrow Node_p$) **do**
      while ($Im_{i,j} \rightarrow Node_n$) **do**
        Create Log: Insert($Im_{i,j}$, $Node_n^{PIT}$)
        As per FIB: Forward($Im_{i,j}$, $Node_{n+1}$, $Node_p$)
      end while
    end while
    Find ($Node_p$. $Dp_{i,j}$, $Im_{i,j}$)
    As per PIT Log, Transmit ($Dp_{i,j}$, $Node_p$, $Node_c$)
  end if
  if ($Node_p$ Generates $Dp_{i,j}^{l-3}$) **then**
    while ($Dp_{i,j}^{l-3}$! $\rightarrow Node_c$) **do**
      while ($Dp_{i,j}^{l-3} \rightarrow Node_n$) **do**
        Suspend($Node_n^{PIT}$)
        Broadcast ($Dp_{i,j}^{l-3}$, $Node_{n+1}$)
      end while
    end while
  end if
end while

5. Performance Evaluation

The proposed scheme is evaluated using ndnSIM simulator by integrating it with raw healthcare data packets generated through standard IoT sensors. The performance evaluation is based on standard evaluation parameters. SecPHCF functions as a two-phase process in which first phase is used to develop an IoT-based application and then an NDN-based scenario is developed using ndnSIM simulator. The IoT application is used to monitor and transmit the health condition of patients. The system is used for real time health detection and alert system based on smart wearable monitoring and tracking. SecPHCF consists of various sensors used to continuously track and monitor

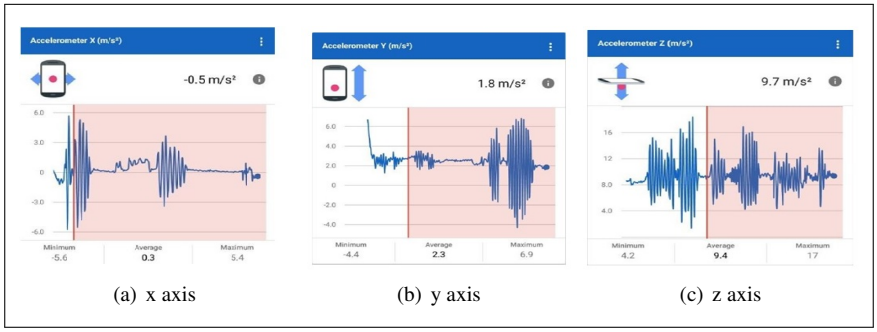
the activity of patient. The model is focused on transmission and effective analysis of data packets transmitted from the sensors. The module embedded with our scheme utilizes sensors such as, GY 61 Accelerator, HW 596 Gyroscope, GY NEO6MV2 GPS Module, Ambience Sensor, and IR Sensor. These sensors have been incorporated in Android-based smartphone devices that are available with patients. The accelerometer sensor is used to detect the change in motion of the peripheral in all axes, i.e., either x, y, z and also the change in momentum. HW 596 Gyroscope module can extract the location or at what position the object is with respect to the ground. NEO6MV2 is a low powered GPS module giving approximated location of the device in real time over the network depending on the number of satellite caches by the module. There are a number of other sensors that have been attached to monitor the health of patient. These sensors are combined to form a BSN. These sensors are further connected to IoT controller device in a wireless mode. SecPHCF also employees $Node\_MCU$ ESP8266 module that is used to connect all BSN sensors. $Node\_MCU$ ESP8266 is like an arduino microcontroller with added WiFi capability in our IoT-based application. The controller implements the source code as a *.ino* file used for initial filtering of raw sensor data. This code becomes the decision system based on threshold levels of individual sensors. These threshold levels are used to detect the health condition of patient which has been categorized into three generic levels in SecPHCF. As a case study, we have considered an epilepsy monitoring module in this work. Table 4 depicts the different sensor positions in terms of x, y and z values along with the facial positions. These values are used by $Node\_MCU$ ESP8266 Module to develop the decision system for epilepsy patients.

Table 4: Decision Table for epilepsy case study

Axis Value	State
if(z $>$ 8)	Face UP
if(z $\leq$ 8)	Face Down
if(y $>$ 8)	Top UP
if(y $\leq$ 8)	Tail UP
if(x $>$ 8)	Right UP
if(x $\leq$ 8)	Left UP

Thereafter, the controller processes the data received from accelerometer and gyro sensors that are attached with it [24]. Figure 6 shows the axis movement along X, Y, and Z axes for the data received from the sensors and represents the value of X, Y, and Z graphically. It shows the minimum, maximum and average value during selected interval of time. This axis movement represents the movement of a person under observation. Figure 7 represents the value of X, Y, and Z axes along with computed value of state on output console based on decision system. SecPHCF performs continuous monitoring of patients in a similar manner based on the states and severity level of received data packet derived for further initiating the NDN process.

For validation of the proposed scheme, an NDN scenario is created using ndnSIM



(a) x axis (b) y axis (c) z axis

Figure 6: Axis movement of sensors

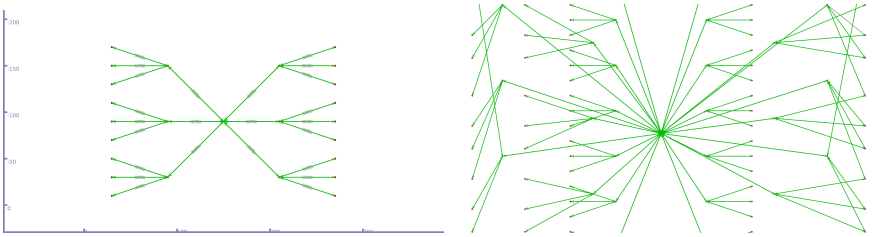
Figure 7: Snapshot of decision system

simulator [25]. Table 5 shows the various parameters used during the simulation. The number of active nodes at a given time in our scenario has been assumed to be approximately 100 and the simulation has been carried out for about 500 nodes. The number of producers, consumers and NDN router nodes have been distributed in the ratio of 4:4:2, respectively. The table also defines the value of link bandwidth and interest packet rate. The simulations was performed for different number of nodes for their specific link bandwidths. There after, the data generated from $cs\ trace$, $rate\ trace$, $app\ delay\ trace$, $and\ drop\ trace$ trace files was analyzed to monitor the performance of SecPHCF. Figure 8(a) and Figure 8(b) shows the simulation scenarios based on star topology generated during the evaluation of SecPHCF. These scenarios represent a partial representation of nodes acting as producers, con-

sumers and NDN routers.

Table 5: Simulation Parameters

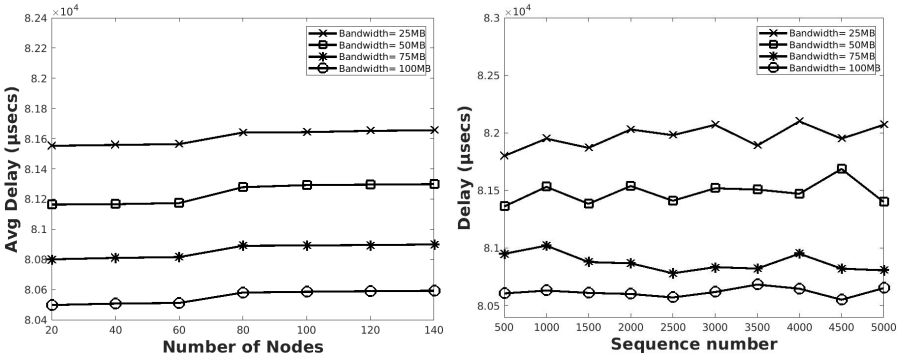
Parameter	**Description**
Number of Nodes	500
Number of Active Nodes	100 (Approx.)
Number of Consumers	39
Number of Producers	39
Number of Routers	13
Topology Used	Tree
Routing Method	Best-Route
No. of Interest Packet Generated	1000/sec
Cache Size	100
Virtual Payload for IoH Data	1024$Bytes$
Link Bandwidth	25 MB, 50 MB, 75 MB, 100 MB
Simulation Time	1$hour$



(a) Simulation scenario for sparse network scenario (b) Simulation scenario for dense network scenario

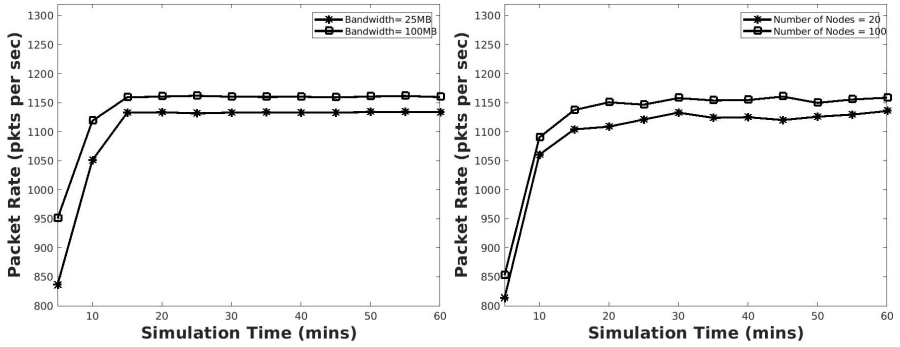
Figure 8: Simulation Scenario

Figure 9(a) shows the variation of average packet delay as a function of number of nodes for various bandwidth links. The value of delay varies from 0.080 sec to 0.081 sec for different bandwidths. The values of packet delay remain almost constant after the specific number of nodes. It shows that SecPHCF provides efficient delivery of healthcare packets. The constant delay can be attributed to the underlying NDN that efficiently utilizes the available network resources. In figure 9(b), the variation of packet delay with respect to variation in sequence number of data packets generated during simulation is depicted for various bandwidth links. Although the figure indicate the variation in value of delay for each bandwidth as the sequence number progresses, the variation is of lower than 0.02 sec. The reason for the same is due to asynchronous generation of data packets that are generated in response to interest packets. Figure 10(a) shows the variation in packet rate as a function of simulation time with respect to various link bandwidths. The value of packet rate first increases in a linear manner



(a) Average Delay v/s Number of Nodes (b) Delay v/s Packet sequence number

Figure 9: Dealy



(a) Packet Rate v/s Simulation time (b) Average Packet Rate v/s Simulation time

Figure 10: Packet Rate

and then stabilizes at a maximum value of 1150 packets per sec. This is on account of lesser number of packets being transmitted during the initial phase of simulation. This value increases as number of packets being transmitted increases and then stabilizes to network saturation value and remains constant thereafter. It is also observed from the figure that packet rate variation between various scenarios for different bandwidth link is less than 50 packets per sec. Figure 10(b) shows the variation in average packet rate for network as a function of simulation time for various node densities. The value of average packet rate exhibits similar behavior to that of packet rate indicating the stable packet carrying capability in SecPHCF. Figures 10(a) and 10(b) show that the underlying NDN network in SecPHCF can utilize network effectively by delivering customized packet size data as required in healthcare systems. Figure 11 shows the average data rate in KB per sec as a function of number of nodes. The variation in data rate for number of nodes is varies from minimum value of 1005 KBps to maximum value of 1030 KBps. However, the data rate becomes almost constant for larger number

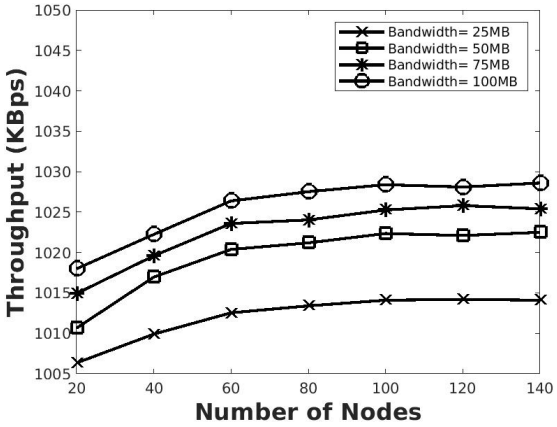


Figure 11: Throughput v/s Number of nodes

of nodes. This behavior signifies the efficiency of SecPHCF. This improved data rate behavior is achieved due to efficient cache handling mechanism used at various NDN nodes.

6. Conclusion and future analysis

A pervasive healthcare system requires an effective data communication system between the users and healthcare facility. A data communication framework is used to provide real time healthcare to users based on NDN and IoT. As efficient data dissemination is a primary means to achieve this objective, we designed different protocols that have been tailored to the requirements of IoT environment. To eliminate redundant transmission of data packets that are generated by sensors, a data packet filtering mechanism has been designed for achieving an efficient data delivery mechanism. This has been done with the help of NDN-based communication module as a backbone network for transmitting data. The work was validated over a simulated scenario considering epilepsy monitoring module. The results obtained suggest the effectiveness of the work based on the findings related to average delay concerning the number of nodes and the packet sequence number. The outcomes related to packet rate and average packet rate also looks promising. Lastly, the throughput witnessed in the experiments varies from a minimum to maximum value and thereafter stabilizing even for large number of nodes, clearly depicting the role of the proposed work. Nonetheless, our results evaluating the proposed show that it is practical, efficient, and effective in delivery of healthcare data packets. Given the broad scope of the subject tackled in this paper, there is ample space for future work on each of the individual components of our framework.

Acknowledgement

This work is partially supported through Startup Fund provided by the Durham University, UK.

References

[1] G. Marques, R. Pitarma, N. M Garcia, N. Pombo, Internet of things architectures, technologies, applications, challenges, and future directions for enhanced living environments and healthcare systems: a review, Electronics 8 (2019) 1081.

[2] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, L. Zhang, Named data networking of things, in: 2016 IEEE first international conference on internet-of-things design and implementation (IoTDI), IEEE, 2016, pp. 117–128.

[3] P. Singh, R. S. Bali, N. Kumar, A. K. Das, A. Vinel, L. T. Yang, Secure healthcare data dissemination using vehicle relay networks, IEEE Internet of Things Journal 5 (2018) 3733–3746.

[4] V. Gazis, M. Goertz, M. Huber, A. Leonardi, K. Mathioudakis, A. Wiesmaier, F. Zeiger, Short paper: Iot: Challenges, projects, architectures, in: 2015 18th International Conference on Intelligence in Next Generation Networks, IEEE, 2015, pp. 145–147.

[5] M. Al-Khafajiy, L. Webster, T. Baker, A. Waraich, Towards fog driven iot healthcare: challenges and framework of fog computing in healthcare, in: Proceedings of the 2nd international conference on future networks and distributed systems, 2018, pp. 1–7.

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Communications Magazine 50 (2012) 26–36.

[7] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, M. Wählisch, Information centric networking in the iot: Experiments with ndn in the wild, in: Proceedings of the 1st ACM Conference on Information-Centric Networking, 2014, pp. 77–86.

[8] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, How to cache in icn-based iot environments?, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), IEEE, 2017, pp. 1117–1124.

[9] M. A. Naeem, R. Ali, B.-S. Kim, S. A. Nor, S. Hassan, A periodic caching strategy solution for the smart city in information-centric internet of things, Sustainability 10 (2018) 2576.

[10] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, S. Gannouni, Afirm: Adaptive forwarding based link recovery for mobility support in ndn/iot networks, Future Generation Computer Systems 87 (2018) 351–363.

[11] N. Kumar, R. Iqbal, S. Misra, J. J. Rodrigues, An intelligent approach for building a secure decentralized public key infrastructure in vanet, Journal of Computer and System Sciences 81 (2015) 1042–1058.

[12] H. Huo, Y. Xu, H. Zhang, Y. H. Chuang, T.-C. Wu, Wireless-sensor-networks-based healthcare system: a survey on the view of communication paradigms, International Journal of Ad Hoc and Ubiquitous Computing 8 (2011) 135–154.

[13] Y. Zhang, L. Sun, H. Song, X. Cao, Ubiquitous wsn for healthcare: Recent advances and future prospects, IEEE Internet of Things Journal 1 (2014) 311–318.

[14] S. Tyagi, A. Agarwal, P. Maheshwari, A conceptual framework for iot-based healthcare system using cloud computing, in: 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), IEEE, 2016, pp. 503–507.

[15] S. Selvaraj, S. Sundaravaradhan, Challenges and opportunities in iot healthcare systems: a systematic review, SN Applied Sciences 2 (2020) 1–8.

[16] D. Saxena, V. Raychoudhury, Design and verification of an ndn-based safety-critical application: A case study with smart healthcare, ieee transactions on systems, man, and cybernetics: systems 49 (2017) 991–1005.

[17] X. Wang, X. Wang, Y. Li, Ndn-based iot with edge computing, Future Generation Computer Systems 115 (2021) 397–405.

[18] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, K.-K. R. Choo, A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments, Journal of Network and Computer Applications 103 (2018) 194–204.

[19] X. Li, M. H. Ibrahim, S. Kumari, R. Kumar, Secure and efficient anonymous authentication scheme for three-tier mobile healthcare systems with wearable sensors, Telecommunication Systems 67 (2018) 323–348.

[20] T. Mick, R. Tourani, S. Misra, Laser: Lightweight authentication and secured routing for ndn iot in smart cities, IEEE Internet of Things Journal 5 (2017) 755–764.

[21] H. Huang, Y. Wu, F. Xiao, R. Malekian, An efficient signature scheme based on mobile edge computing in the ndn-iot environment, IEEE Transactions on Computational Social Systems (2021).

[22] P. Gope, T. Hwang, Bsn-care: A secure iot-based modern healthcare system using body sensor network, IEEE sensors journal 16 (2015) 1368–1376.

[23] I. Moiseenko, L. Wang, L. Zhang, Consumer/producer communication with application level framing in named data networking, in: Proceedings of the 2nd ACM Conference on Information-Centric Networking, 2015, pp. 99–108.

[24] S. Sharma, R. K. Dudeja, G. S. Aujla, R. S. Bali, N. Kumar, Detras: deep learning-based healthcare framework for iot-based assistance of alzheimer patients, Neural Computing and Applications (2020) 1–13.

[25] A. Afanasyev, I. Moiseenko, L. Zhang, et al., ndnsim: Ndn simulator for ns-3, University of California, Los Angeles, Tech. Rep 4 (2012) 1–7.

Rajan Kumar Dudeja is working towards his Ph.D in Computer Science and Engineering Department from Chandigarh University, India. He is also an assistant professor of with the Computer Science and Engineering Department, Chandigarh University.

Rasmeet Singh Bali received the M.E. degree from the National Institute of Technical Teachers Training and Research, Chandigarh University, Punjab, India, in 2013, and the Ph.D. degree from the Thapar Institute of Engineering and Technology, Punjab, India, in 2018. He is currently a Professor with the Computer Science and Engineering Department, Chandigarh University.

Gagangeet Singh Aujla is an assistant professor of computer science at Durham University, UK. He was awarded the 2018 IEEE TCSC Outstanding Ph.D. Dissertation Award and the 2021 IEEE Systems Journal Best Paper Award. He is an Area Editor of Adhoc Networks (Elsevier) and Associate Editor of IET Smart Grid.