A Distributed Approach towards Improved Dissemination Protocol for Smooth Handover in MediaSense IoT Platform

Authors:

Shabir Ahmad, Ishfaq Hussain, Muhammad Fayaz, Do-Hyeun Kim

Date Submitted: 2018-07-31

Keywords: handover management, wireless sensor networks, Internet of Things

Abstract:

Recently, the Internet has been utilized by many applications to convey time-sensitive messages. The persistently expanding Internet coverage and its easy accessibility have offered to ascend to a problem which was once regarded as not essential to contemplate. Nowadays, the Internet has been utilized by many applications to convey time-sensitive messages. Wireless access points have widely been used but these access points have limitations regarding area coverage. So for covering a wider space, various access points need to be introduced. Therefore, when the user moves to some other place, the devices expected to switch between access points. Packet loss amid the handovers is a trivial issue. MediaSense is an Internet of Things distributed architecture enabling the development of the IoT application faster. It deals with this trivial handover issue by utilizing a protocol called Distributed Context eXchange Protocol. However, this protocol is centralized in nature and also suffers in a scenario when both sender and receiver address change simultaneously. This paper presents a mechanism to deal with this scenario and presents a distributed solution to deal with this issue within the MediaSense platform. The proposed protocol improves dissemination using retransmission mechanism to diminish packet loss. The proposed protocol has been delineated with a proof of concept chat application and the outcomes have indicated a significant improvement in terms of packet loss.

Record Type: Published Article

Submitted To: LAPSE (Living Archive for Process Systems Engineering)

Citation (overall record, always the latest version):	LAPSE:2018.0336
Citation (this specific file, latest version):	LAPSE:2018.0336-1
Citation (this specific file, this version):	LAPSE:2018.0336-1v1

DOI of Published Version: https://doi.org/10.3390/pr6050046

processes

MDPI

Article

A Distributed Approach towards Improved Dissemination Protocol for Smooth Handover in MediaSense IoT Platform

Shabir Ahmad [1], Ishfaq Hussain [2], Muhammad Fayaz [1] and Do-Hyeun Kim [1,*]

[1] Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; shabir@jejunu.ac.kr (S.A.); fayaz@jejunu.ac.kr (M.F.)

[2] Department of Information Technology and Media, Mid Sweden University, Sundsvall 85229, Sweden; engr.ishfaqhussain@gmail.com

* Correspondence: kimdh@jejunu.ac.kr

Received: 21 March 2018; Accepted: 14 April 2018; Published: 1 May 2018

Abstract: Recently, the Internet has been utilized by many applications to convey time-sensitive messages. The persistently expanding Internet coverage and its easy accessibility have offered to ascend to a problem which was once regarded as not essential to contemplate. Nowadays, the Internet has been utilized by many applications to convey time-sensitive messages. Wireless access points have widely been used but these access points have limitations regarding area coverage. So for covering a wider space, various access points need to be introduced. Therefore, when the user moves to some other place, the devices expected to switch between access points. Packet loss amid the handovers is a trivial issue. MediaSense is an Internet of Things distributed architecture enabling the development of the IoT application faster. It deals with this trivial handover issue by utilizing a protocol called Distributed Context eXchange Protocol. However, this protocol is centralized in nature and also suffers in a scenario when both sender and receiver address change simultaneously. This paper presents a mechanism to deal with this scenario and presents a distributed solution to deal with this issue within the MediaSense platform. The proposed protocol improves dissemination using retransmission mechanism to diminish packet loss. The proposed protocol has been delineated with a proof of concept chat application and the outcomes have indicated a significant improvement in terms of packet loss.

Keywords: Internet of Things; wireless sensor networks; handover management

1. Introduction

The number of smart electronic devices, such as smartphones, different wearables, and connected appliances, has increased significantly. A network of electronic devices like these, capable of communicating with each other to reach common goals, can be referred to as the Internet of Things (IoT) [1]. The devices are able to observe and interact with the physical environment, which allows the IoT to influence our lives significantly via applications in home automation, security, automated devices, health monitoring, and management of daily tasks. Current estimations claim that there will be over 50 billion connected devices as soon as 2020 [2], of which many will be typical IoT devices, such as small embedded computers (e.g., Raspberry Pi devices) or different wireless sensor networks. It is expected that in near future all things will be communicating. This is referred to as the Internet of everything.

In order to speed up the development of these IoT-based applications, many efforts are made to create middleware platforms. MediaSense is one of such attempts to act as a middleware between sensors and actuators and eases the development of IoT application on top of the platform by providing

application programming interface for end users without the need to develop everything from scratch. The major requirements of these middleware platforms are to exhibit real-time requirements, distributed and seamless in nature and to degrade gracefully with scaling up the number of connected things.

Numerous projects on the MediaSense have been done. The MediaSense project has created modules for the accumulation of contextual information from sensing devices connected to WSNs. The contextual information is originated from various diverse sources, for example, smart doors, smart cars, and smartphones. The communication between these smart devices is performed with Internet Protocol (IP) addresses. In order to deal these IP addresses and things' identity over network layer Distributed Context eXchange Protocol (DCXP) has been utilized by the MediaSense. It is a distributed protocol based on Distributed Hash Table (DHT) which deals with the communication among physical things. With the movement of one of these physical devices, the IPs assigned to them might change amid a session. This issue turns worst into a scenario when both the sender and receiver alter from their connection point at the same time. That being said, the solution does not serve context awareness which is considered among preliminary requirements in any IoT platform. DCXP deals with this problem by leveraging a centralized server which acts as a shield for packet loss but this idea to have a single node to avoid packet loss is not distributed in nature and hence fails to meet the "Distributed" requirement of the MediaSense. To tackle this challenge, we present a robust solution based on DCXP to enhance the scalability and mobility of the MediaSense platform and every node has given the autonomy to deal with packet loss, thus, gives the aided benefit of fulfilling the "Distributed" Requirement of the MediaSense Platform. That being said, the main technical merits of the proposed protocol are threefold; first it replaces the centralized nature of the approach used in DCXP and second it improves the overall architecture of the DCXP so that it scales well with increasing number of peers, finally, it introduces context-awareness in the proposed protocol by detecting simultaneous disconnection of nodes.

The rest of the paper is organized as follows. Section 2 exhibits the relevant research work on IoT platforms and describes MediaSense architecture in conjunction with DCXP. Section 3 presents the proposed protocol and discusses the design and architecture in detail. Section 4 shows the results by considering different factors like packet loss, number of peers and Jitter. Section 5, finally, conclude the paper and gives the future direction of this research.

2. Related Work

The related work presented in this section will provide an understanding of the technologies used for dealing with packet loss during handover.

2.1. Related Protocols

The increasing use of wireless access points for the establishment of communication has some impediments [3]. These access points cover a very limited space and in order to cover wider space, multiple access points need to be instated. During mobility, a node can switch from one access point to another access point called handover. In this section of the paper, an investigation of effective handover has been discussed in a scenario when devices move from one access point to the next. In case the IP addresses are used handover performs successfully but the session won't be retained and hence packet loss can be experienced during re-establishing of the session. Some squeezing issues in the current packet-based protocols include Loss-of-Trust, undesirable traffic, and poor support of multihoming and mobility [4].

Numerous protocols have been sanctioned in order to perform smooth and packet loss-less handover. Though all approached the problem with a different frame of mind, however, their end goal was similar. For instance, Host Identity Protocol (HIP), Location Identifier Separation Protocol (LISP) and Mobile IPv6 offer their own particular intends to effectively handle the problem. In contrast, Hierarchical Mobile IPv6 and PMIPv6 have an attention to enhancing the aforementioned protocols [5]. As it is known that host and location are identified by IP addresses but it can last for a short time in case

the location of the device changes. HIP was proposed in order to tackle this issue. HIP uses public key encryption [6] and provides secure end-to-end connectivity. It uses Network Address Translation (NAT) to identify hosts [5]. Locator Identifier Separation Protocol (LISP), in contrast to HIP, is a network-based standard. It focuses on improving the scalability of the network routing system [7,8].

LISP has been proposed in view of perceptions produced using an alternate angle. The current IP routing and addressing architecture provides a single address for both device identification and the topology of the network. The LISP architecture separates device identity from its location identity [9,10]. In addition, because of the way that LISP uses map/encap techniques, so there is no compelling reason to alter the host stack [11] like in HIP. Issues like routing scalability are settled in such a way that for every device's IP two different numbers are assigned: one of them is called Routing Locators (RLOCs) and the other is called Endpoint Identifiers (EIDs). Routing Locators (RLOCs) assignment is topology-based and is utilized to forward data and route data in the network. On the other hand, Endpoint Identifiers (EIDs) are topology independent and are utilized for numbering [12]. It enhanced routing system scalability by utilizing topologically dependent Routing Locators (RLOCs) [12].

Mobile IPv6 focuses on the offering of a smooth handover of mobile nodes amid switching between access points. It gives unbroken connectivity to mobile nodes while roaming between wireless access points in an alternate subnet. (L3 handover) [13]. When both the wireless access points are on the same subnet the handover is performed on L2. In contrast, if the handover is performed on different subnets then it is called L3 handover. The mobile nodes are distinguished by their locations in the whole procedure. When a mobile nodes originates from its source location it has one address called home address. However, when it moves to the next location which resides on different subnet the address is changed and a new address is assigned to it known as the Care-of-Address(CoA). The mapping is registered in a table and this operation is known as binding update. So, in the meanwhile, the binding update is performed some packets are lost which is one major issue in this protocol.

Other protocol, for example, The Hierarchical Mobile IPv6 (aka known as HMIPv6) proposed which gave the idea of Mobile Anchor Points (MAP) [14]. In this protocol, if the location of the source node and destination resides on the same subnet and same access point the handover is performed locally and no binding update is registered [15,16]. Fast handover main goal is to optimize the handover latency of Mobile IPv6 [17] which are caused by binding registry and updates. In this protocol packet loss is diminished by combining packet tunneling and buffering amid handover [18].

2.2. MediaSense

Currently, there is a vast number of different systems used to connect IoT applications to sensors and actuators. Most are typical Cloud-based systems with one or more centralized servers on the Internet, such as Nimbits, Azure IoT, Serviocity, Evrythng, Dweet, and Thingsquare [19]. These Cloud-based systems are far from optimal when it comes to creating a future-proof and ubiquitous IoT system [20], especially when it comes to large-scale communication, and adding more nodes. The platform we consider for this paper is a fully distributed and peer-to-peer approach.

Most fully distributed IoT systems create an overlay using a Distributed Hash Table (DHT) [21–23] to enable logarithmic or better scaling when the participants increase in magnitude. There is some communication overhead related to the maintenance of the DHT itself since it needs to maintain references between the participants of the DHT. MediaSense [24] is a DHT-based distributed peer-to-peer platform proposed by Mid Sweden University. The main motive behind the idea of MediaSense is to enable seamless, scalable data sharing on heterogeneous network overlay. It is available under GNU public license and can be downloaded and used for development of IoT application on top of it [25]. The main characteristics which plays a pivotal role are that the platform guards against central failure. Moreover, the platform is scalable for sensor data sharing and with the amount of sensors increasing in the system it still gives legitimate response withing the deadline. MediaSense can be implemented using Raspberry PI with each Raspberry PI acting as a node in the peer-to-peer network where different sensors and actuators are connected to it and thus communicating

back and forth using MediaSense. Each physical sensors and actuators are represented using a unique identifier called Universal Context ID (UCI). When an application requests contextual data from sensors requests to the UCI of the sensor. The MediaSense Platform resolve the UCI and provide the IP address of the host [25]. MediaSense has 5 main primitives as described in Figure 1.

The Interface Layer is the main entry point to interact with the MediaSense platform. It provides application programming interface for end users. Additionally, this layer provides hook to interact with the lower layer of the platform. Beneath Interface Layer is Add-in Layer which is where new extension can be added to the platform. These extension can either be optimization of the platform or adding new functionality to the platform. Dissemination Layer, which resides next to Add-in Layer, holds communication between entities, discover entities and helps in sharing sensor data over a peer-to-peer networks. Distribution Layer is a mediator between MediaSense implementation and TomP2P . This is the place where the UCI from MediaSense and peers from TomP2P are mapped. Messages and UCIs are paired and communicated on the overlay using TomP2P . TomP2P is the advance implementation of DHT and has extended DHT operation. For instance, in DHT each entry is stored in exactly one key-value pair form but TomP2P extends the methods to support multiple values against a single key [26]. Utility are some additional classes which help in implementing functionality of the system and message are the data that is communicated between hosts.

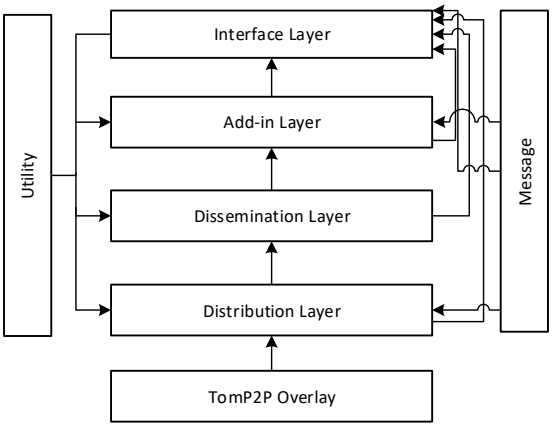


Figure 1. MediaSense Architecture.

2.3. DCXP

The Distributed Context eXchange Protocol (DCXP) [27] is an XML-based application layer peer-to-peer protocol which offers reliable communication among nodes in peer-to-peer network. End devices are connected to the Internet to register with the peer-to-peer network and may share context information (CI). DCXP uses UCIs to represent contextual information originating from sensors and other nodes. It transmits context information in the form of a DCXP Message. A typical DCXP Message has five main primitive mainly for *Registering UCI*, *getting UCI data*,*Resolving UCI*, *Notifing* and *Subscribing* as described in Table 1.

The nodes in the network form a ring structure as discussed earlier and every node has a service running all the time called Context User Agent (CUA). The ring structure is very vital for real-time peer-to-peer application due to its algorithmic access which is much more efficient than its counterparts typologies. The DCXP ring architecture is described in Figure 2. DB is a database repository which persists the Context Information. Each DCXP node has some services running, a Database Agent(A) which listens to the request in the form of message and forward them to database, a Database Dataminer (D) responsible for gathering context information and sending it to the database for persistence, and a Database Client (C) which sends request to A and receives the response from A. Additionally, a node which has bootstrap service running is the first node and must be called first. The Remote CUA and Local CUA are similar to server and client respectively. DCXP shields the

packet loss by employing a centralized server called Mobile DCXP Proxy (MDP). In Figure 2, the node having RCUA and bootstrap service is acting as a centralized server and each and every node needs to register here and it also responsible for preventing packet loss and radio disruption issues. That being said, this centralized approach is not scalable and packets are lost if the point of connection changes simultaneously. The paper focuses on the need to have a distributed approach that scales well with adding more nodes and can diminish packet loss in the above mentioned scenarios.

Table 1. DCXP Message's Premitives.

Premitive	Description
REGISTER_UCI	A node must invoke REGISTER for registering the UCI of Context Information with Context Storage
RESOLVE_UCI	The UCI is resolved to find the source address of the Context Information.
GET	Once the node receives the resolved address using *RESOLVE_UCI*, it GETs the Context Information from the source node.
SUBSCRIBE	This primitive allows nodes to subscribe to a Context Information, so, whenever the Context Information gets available it will receive it.
NOTIFY	The updates about new data is been communicated to subscribing nodes using this primitive.

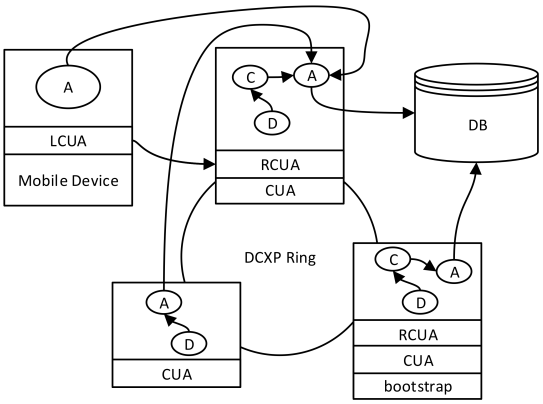


Figure 2. DCXP Architecture.

3. Proposed Protocol

The main goal of this research is to enhance the existing DCXP in such a way that the packet loss ratio decreased and smooth handover is performed even if the point of connection changes simultaneously. DCXP protocol works on Dissemination Layer of MediaSense so our focus is to workaround on the same layer to offer a better solution.

The approach utilized in this research is to go around data loss amid handover utilizing the DCXP protocol. Therefore, in case a node gives acknowledgment upon receiving the data, at that point it can be conceivable to keep on updating the node to the extent that the device remains associated with the network. Consequently, the concern is not to keep the node associated amid handover rather it is worried about the updating of data and accordingly anticipates data loss. The objective has just been to discover an answer for the smooth handover when dissemination of messages occurs in wireless network and the conviction is that the main wellspring of issues is DCXP. Since the dissemination of data happens in the real-time, it can be safely conveyed that only inside a specific time frame that data is viewed as meaningful for dissemination. In light of this reality, emerge the question "Is it unrealistic

to retry dissemination of packets that have been lost because of handover if the context information is still legitimate?" If that re-transmission is conceivable it implies that quite possibly in the retrial the peer appears in another wireless access-point inside limit of the legitimacy of the data and ready to affirm its validity.

3.1. Design and Architecture

In this paper, the main goal of the DCXP is modified to carry out re-transmission of lost packets within their time of relevancy. Real-time data have very short deadline and after that the data is irrelevant even if it is logically correct. Therefore, in a scenario when the the packets relevance of real-time data is more than the handover latency, the packets can be re-transmitted to the instant the packet is regarded as relevant. The proposed idea can be illustrated with Figure 3. Four nodes are roaming between two access points AP1 and AP2. The dotted circles represent the space these access points cover and the intersection of the circles represent the area where handover is performed and thus the packet drop could occur.

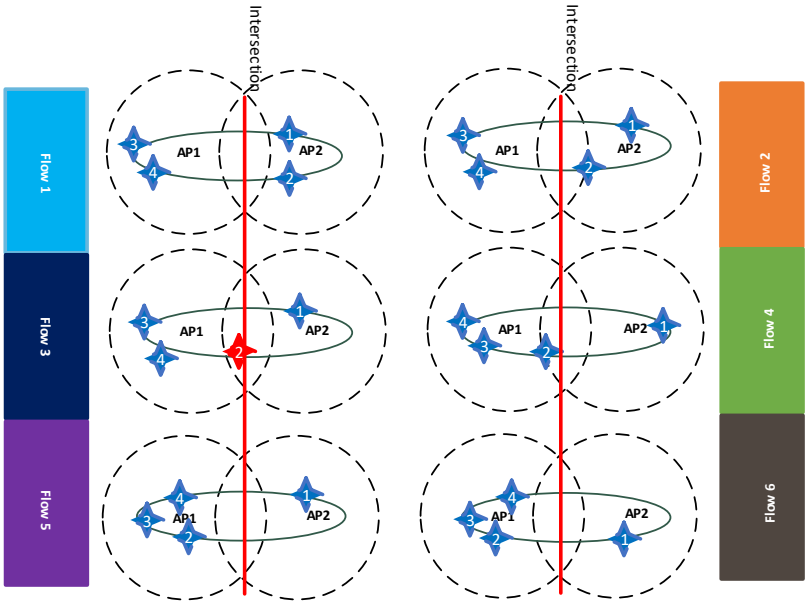


Figure 3. Handover.

It can be seen that node 2 is mobile node and moving towards access point AP1 from access point AP2. When it reaches the intersection area in Figure 3 flow 2, it gets disconnected from both access points as shown in Figure 3 flow 3. Lets assume the time it reaches the intersection flow 2 is t_1 and the time it reaches to the other end of the intersection i.e., flow 4 is t_2. The total handover time T_h is

$$T_h = t_2 - t_1$$

Suppose the deadline of the message or the relevancy time of the message is M_d then the proposed protocol works best in case

$$M_d \geq T_h$$

The solution proposed in this paper is a distributed algorithm to deal with the issue of packet loss emerging because of the detachment of mobile devices from one access points to the another access point. It is distributed in a sense that now instead of a centralized node, every node is responsible for ensuring the mobility and avoiding packet loss.

The motivation of this solution is to acquaint additional primitives that assist to trace out the destiny of a message disseminated. Accordingly, if a packet drops because of handover, at that

point re-transmission need to take after or if the packet successfully ends up in destination then re-transmission will not going to happen as shown in Algorithm 1. There are four global variables; node is of type Node and message1, message2, and ack is of type Message. In technical words, these variables are the objects of class Node and Message respectively. DISSEMINATE method takes node and msg variables as arguments and performs sending messages to member nodes. Similarly, REDISSEMINATE function have the same signature but recursively called DISSEMINATE function for all unacknowledged node that are still relevant and this goes on unless the message become obsolete i.e., the real-time relevancy of the message is expired.

Algorithm 1 Proposed Protocol

```
1:  global variables
2:      Node global node
3:      Message global message1
4:      Message global message2
5:      Message global ack
6:  end global variables
7:  function DISSEMINATE(node, msg)
8:      //Perform sending of messages to all the member nodes.
9:      return
10: end function
11: function REDISSEMINATE (node, msg)
12:     node ← acknowledged nodes
13:     //For all unacknowledged messages.
14:     repeat
15:         DISSEMINATE(node, msg)
16:     until msg ≠ Obsolete
17:     //A message is obsolete if its relevancy time is less than handover.
        ▷ If the handover latency is more than the message relevancy then simply discard the message
18:     return
19: end function
```

The real objective here is to perform re-transmission of packets (messages) in a legitimate time to those nodes experiencing handover and thus encountering packet loss. The real issue is that the messages get out of date in the span of brief time. This would mean if the relevancy of message is short enough to be re-transmitted within latency of handover, then our proposed protocol is a very great step to prevent packet loss and make it available until the relevancy of message expired. As the time taken by a node to switch from one access point then onto the next is exceptionally negligible so this approach could be a good work around solution.

DCXP protocol throws "DestinationNotReachable" exception for the situation when the destination node is not reachable as a result of different factors which include a short disruption because of handover. Thus, in this situation, the packets are being dropped and the destination node is not ready to get the message in spite of the fact that a very mundane fraction of time are taken by the handover and the message is still relevant. The flowchart of the proposed protocol is demonstrated in Figure 4. It can be seen from the flowchart that the context information (CI) in form of messages are sent and the condition is evaluated first to check if the destination node is reachable and second if the destination node is active. A node is reachable if it is attached to an access point, however, a node is still considered active if the Context Information (Message) it is supposed to receive is still relevant and within the real-time deadline. So, the source nodes keep sending the message until the message losses their relevancy and get outdated or until the node gets the message. Once the message arrives at the destination node, acknowledgment is sent to the source node. In case of DCXP, if the condition

is evaluated to false it simply throws "DestinationNotReachable" exception and discard the message in spite of the fact that the message is still relevant and can be re-transmitted.

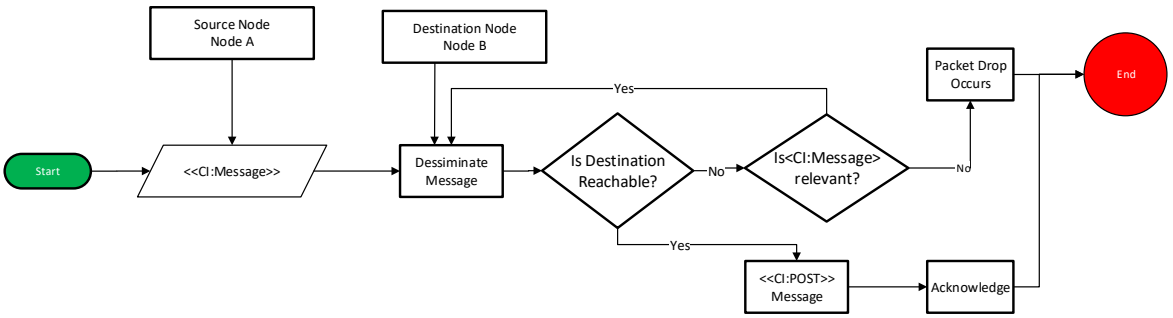


Figure 4. Proposed Protocol Flowchart.

Figure 5 shows the interaction model of peer nodes within MediaSense platform utilizing proposed protocol. The Interface Layer deals with interfacing with peer nodes and the Dissemination Layer is bridging between the MediaSense networking layer. The corresponding layers are shown in the figure. The request is parsed by MediaSense and resolved to the correct UCI of the sensor. The Sensor posts the Context Information in real-time which is accumulated by the MediaSense platform which in turns disseminates the data using proposed protocol. The proposed protocol add a loop construct which continuously compares the message relevancy time and the handover latency and tries re-dissemination of the messages till the point the message deems valid. The messages are acknowledged all the way to Node A and the action are performed on the correct destination i.e., Node B.

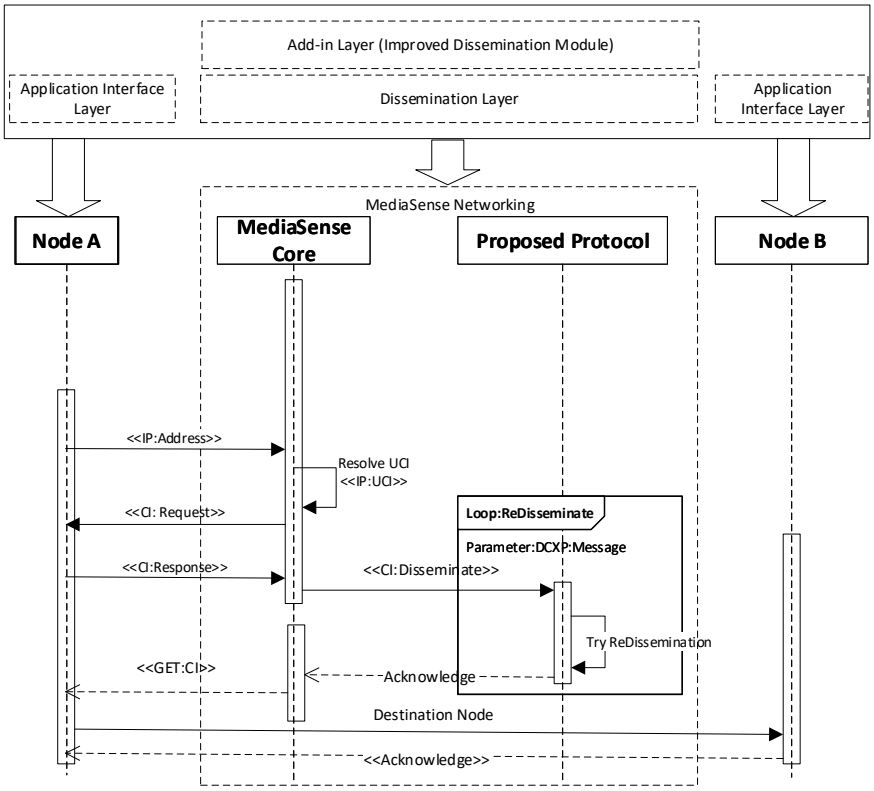


Figure 5. Interaction Model of Peer Nodes within MediaSense.

3.2. Implementation Detail

Although DCXP is based on the Dissemination Layer of the MediaSense, but MediaSense developers strongly discourage to do any changes to the core of the system in order to keep the platform secure and unaltered. For this reason, the implementation of the proposed protocol is added on Add-in Layer since this is where users defined extensions and optimization modules need to be added. Figure 6 shows the enhanced MediaSense architecture utilizing proposed protocol. The proposed protocol is added to Add-in Layer and in order to assess and compare the performance of both DCXP and proposed protocol, an android chat application has been developed. Thus, the application keeps running over MediaSense platform and the extension included in the Add-in Layer of the MediaSense have been utilized as a part of the application. In order to assess how much optimization have been achieved, the application is run on the two systems; the one which have MediaSense with DCXP version and the other proposed protocol version. The access points are turned on and off to emulate handover. The amount of packet loss are recorded for both the systems.

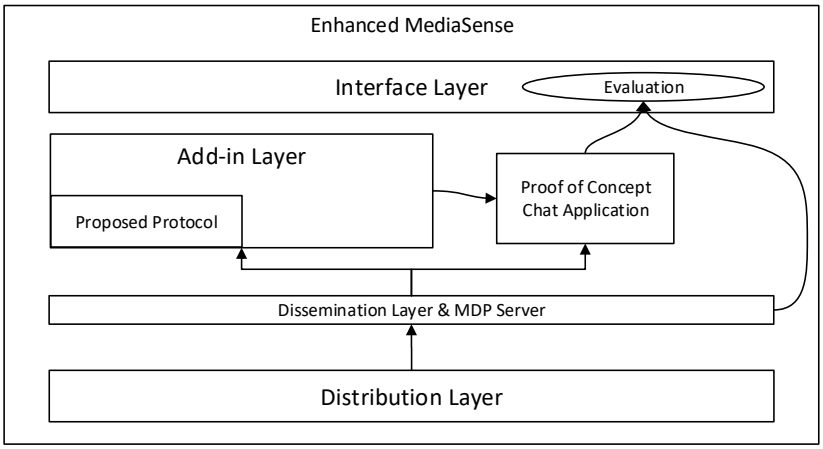


Figure 6. Enhanced MediaSense Architecture.

The implementation details and technologies used in the experiments are described in Table 2.

Table 2. Technology Stack of IoT Server.

Component	Description
Hardware	Raspberry PI 3 Model B
Operating System	Raspbian
RAM	1 Gega Byte
IoT Server	MediaSense
Resources	LED, Temperature Sensor, Breadboard, Connecting wires
Libraries	TomP2P, General Purpose Input/Output GPIO
Communication Protocol	DCXP and Proposed Protocol
IDE	Android Studio and Eclipse (Remote Access)
Programming Language	Android and Java

The peer-to-peer network configuration model is depicted in Figure 7, which highlights that various peer nodes form a ring and communicate with each other using proposed protocol deployed on top of MediaSense. Each node is equipped with the MediaSense, proposed protocol, and GPIO libraries for interaction with physical sensors and actuators. A typical of an IoT peer is a Raspberry PI based node with the specifications outlined in Table 2.

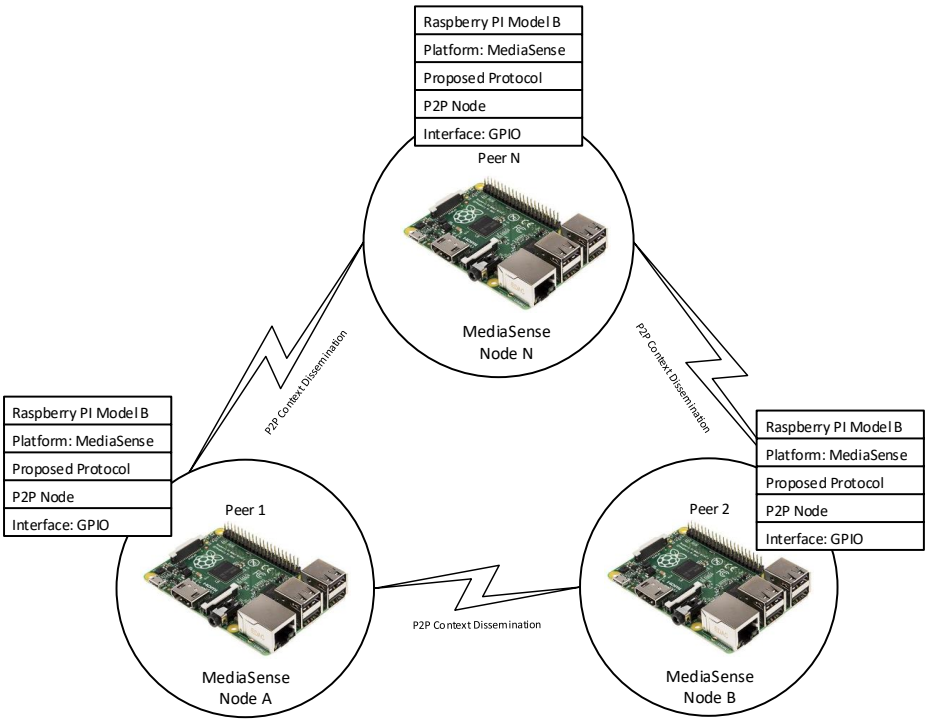


Figure 7. Peer-to-peer Network Configuration Model.

4. Results

This section presents correlation between DCXP with the proposed protocol. The improvements accomplished with the proposed protocol have been shown in Table 3. Moreover, the extension developed to accomplish the implementation of the proposed protocol has been clarified in detail.

Table 3. Comparison of DCXP with Proposed Protocol.

DCXP	Proposed Protocol
Packet loss occurs during switching of Wi-Fi	There is slight possibility of packet loss during switching of Wi-Fi
Uses Proxy DCXP, a centralized server, to handle problems related to packet loss	Uses a distributed approach, to deal with packet loss that might occur during handover.
When packet loss occur the *DestinationNotReachable* exception is thrown.	During the time the node disappears when Wi-Fi switching occurs, the re-transmission of lost packets are carried out.

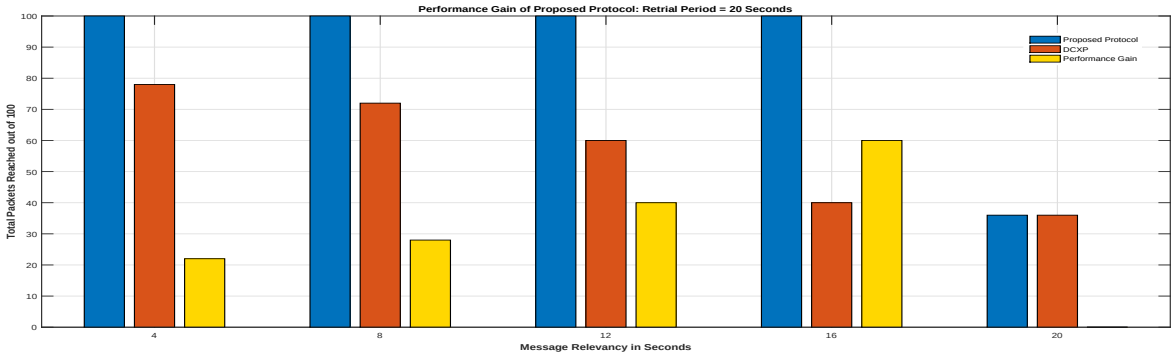
To inspect the gain in performance of the proposed protocol in contrast to the DCXP with respect to packet loss, an experiment has been carried out in which unique numbers are assigned for every message transmitted. These unique IDs could cluster the approaching messages and additionally reveal to us which packet is missing. Hence, switching on/off the Wi-Fi has tried for the two cases. For this situation, there is one factor, which is the relevancy time of the message. As it is known the context information originates from wireless sensor networks, the experiment ought to consider the relevancy factor of context information. To tackle this situation, after a few trials we have to quit sending the message. So, in worst case if a node is experiencing longer delay than its maximum time limit for real time communication then some packets will be lost which is unavoidable but this scenario is highly unlikely. Based on the type of applications, the re-transmission of the packets, which are lost, for additionally expanded time frame could be valuable than leave the lost packet and simply swing to the new packets.

4.1. Packet Loss

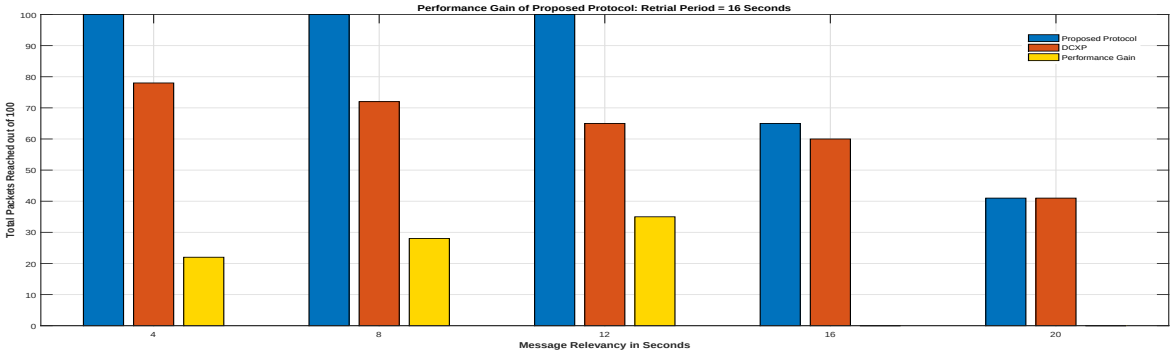
To find the quantity of lost packets in DCXP and proposed protocol, the access point has been turned on and off to reenact the handover. This strategy has been favored due to convenience and because of the way that it is conceivable to oversee delay that could happen amid handover.

Therefore, the actual time access point takes to reboot is around 4 s which implies peer-to-peer communicating devices will experience DestinationNotReachable error. In DCXP, packet loss happens and we see no real way to overcome and recover those lost packets while in the proposed protocol every peer retries the transmission of lost packets for differed retrial periods. For this situation to gauge the likelihood of packet loss as it happens in the extended delay has been depicted in Figure 8. It can be clearly seen from figure that we have gathered the amount of packet loss in percent for delay of 20 s, 16 s, 12 s and 60 s. An aggregate number of 100 messages have been used as a part of the experiment and each message is doled out a unique serial number.

The bar graph in Figure 8 demonstrates the performance gain of the proposed protocol with the existing DCXP version of the MediaSense platform. Results clearly suggests on increasing the delay of Wi-Fi handover, the number of packet dropped in the existing DCXP increases while in the proposed protocol the packets are not dropped till the point the messages are deemed relevant. The performance gain of the proposed protocol with respect to existing DCXP are increasing as the delay in the handover increases given that it still falls within the relevancy time of the message. In that case both the protocol perform in a similar way. In spite of the fact that it could have been a conceivable answer for considering additionally expanded retrial period however that does not satisfy the real-time requirement of the MediaSense. Though the likelihood of packet loss has decreased yet at the same time there exists a breach to lose packets in the situation of extended time grasped by handover. Notwithstanding, under normal conditions, it does not take a longer period than mentioned in this paper.

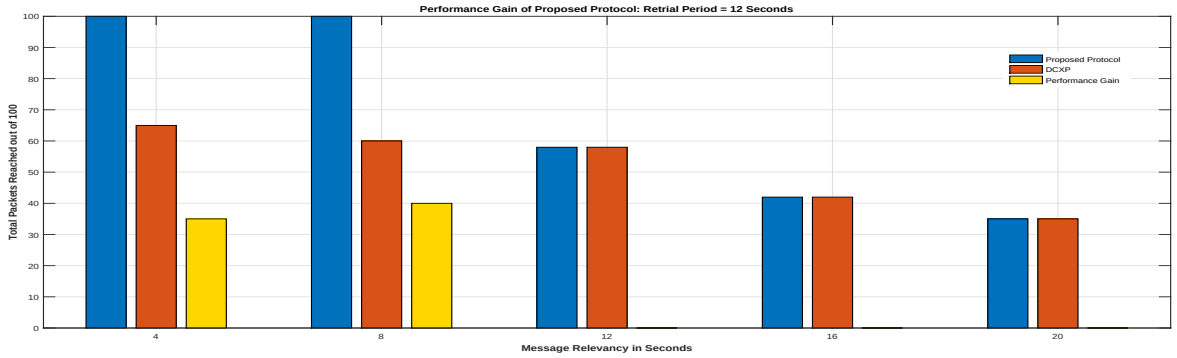


(**a**) Retrial Period = 20 s

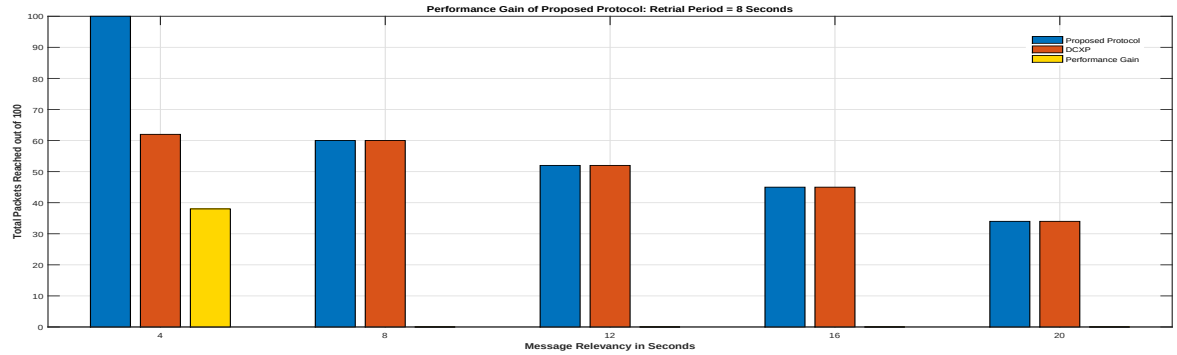


(**b**) Retrial Period = 16 s

Figure 8. *Cont.*



(**c**) Retrial Period = 12 s



(**d**) Retrial Period = 8 s

Figure 8. Bar Chart illustrating Packet Loss in the Proposed Protocol and DCXP. (**a**) Retrial Period = 20 s; (**b**) Retrial Period = 16 s; (**c**) Retrial Period = 12 s; (**d**) Retrial Period = 8 s.

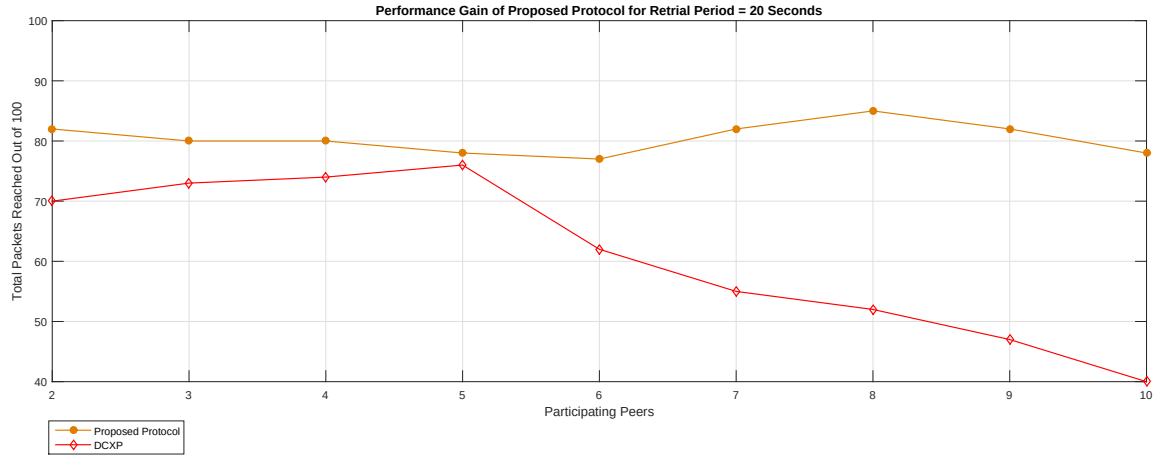
4.2. Scalability Measures

In order to measure whether the proposed protocol is scalable when more peers are added, an experiment has been conducted. The chat application has been installed on two systems initially and the performance are measured then more peers are added and the performance are recorded. With the existing DCXP the performance of the protocol degraded whereas the proposed protocol is steady and not degraded much.

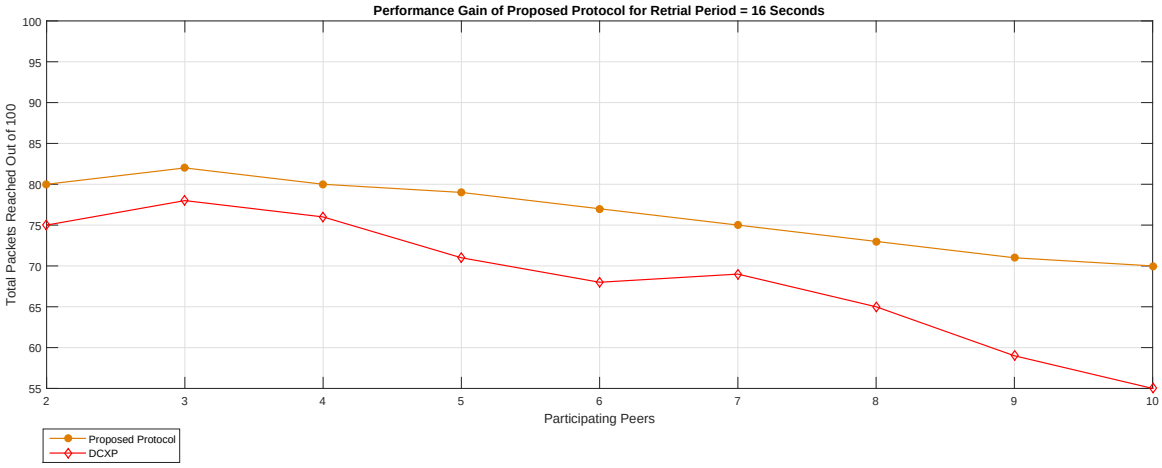
Figure 9 shows the the experimental results for 2, 4, 6, 8, 10 peers and the results are recorded for different handover latency which is manually emulated by turning on and off the access points. For this scenario we kept the message relevancy time constant for both cases.

It can be seen from the figure that the overall effect of both the systems are same with respect to the number of messages but the proposed protocol is more robust and degrade gracefully.

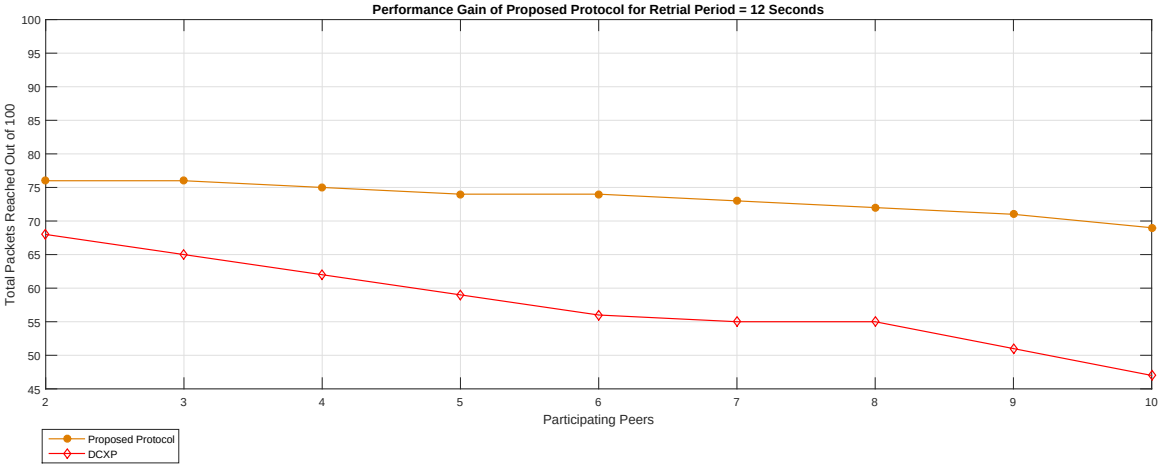
The other critical issue is the way that IP address could change and the correct node would be hard to follow out. In this scenario, while turning the wireless access points on and off, the IP addresses do not change so it has not been a challenge here. However, in a situation where there are two access points and where there are numerous connection and disconnection, this could be an issue to deal with. One of the alternatives to deal with this could be to discover the node by their UCI. The resolve function could just result in either currently assigned IP address/port or it just throws an error.



(**a**) Retrial Period = 20 s

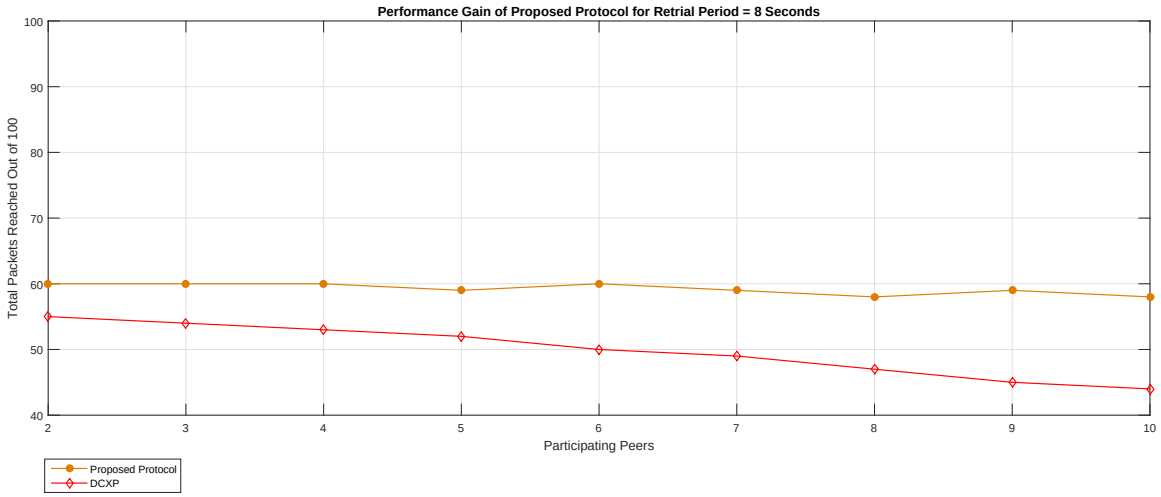


(**b**) Retrial Period = 16 s



(**c**) Retrial Period = 12 s

Figure 9. *Cont.*

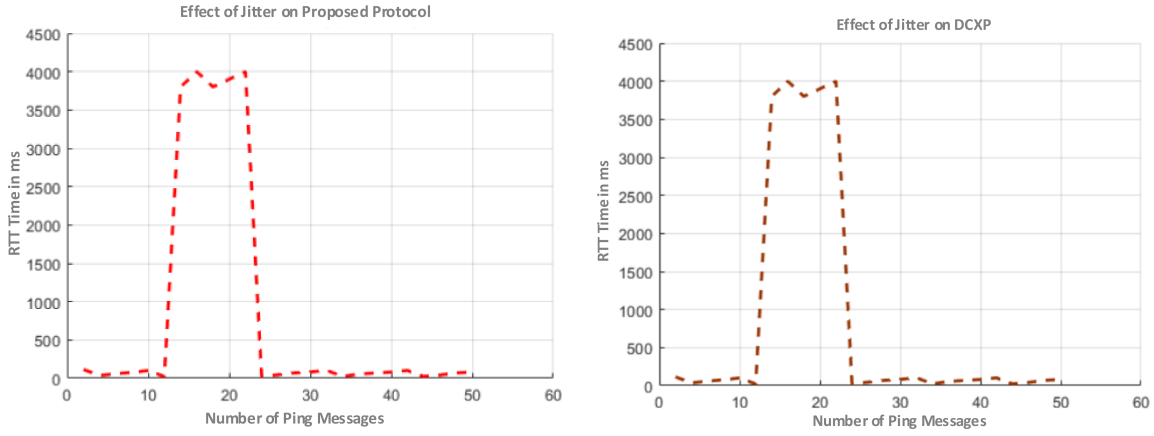


(**d**) Retrial Period = 8 s

Figure 9. Effect of Peers on Proposed Protocol and DCXP.

4.3. Jitter

In order to assess the Jitter induced in MediaSense an experiment has been performed. Jitter is an attribute which measures the quality of network. However, our modification has been done on Add-in Layer which is application level layer so we witness no change for both the solutions as depicted in Figure 10.



(**a**) (**b**)

Figure 10. Effect Of Jitter on Proposed Protocol and DCXP. (**a**) Proposed Protocol; (**b**) DCXP.

4.4. Subjective Testing

As part of the subjective testing, eighty students of university have been given the job to run MediaSense-based applications using existing DCXP and then using proposed protocol and inspect the difference between both. Experiments results are recorded in a questionnaire. Results propose that there is a genuine performance gain. From Figure 11 it has been affirmed that the progressions done on the MediaSense platform have for sure achieved gain in performance. As appeared in the figure, 96% of the considerable number of subjects has noticed the difference. Moreover, 80% of the considerable number of members highlighted the modification in the proposed protocol.

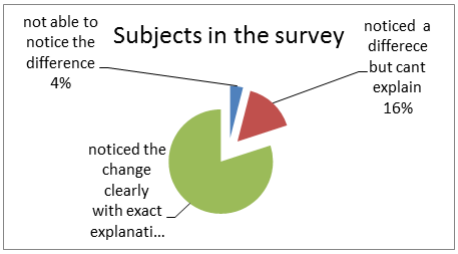


Figure 11. Subjective Testing Survey.

5. Conclusions

In this paper, a new protocol has been proposed which tends to optimize DCXP regarding packet loss. The experimental analysis alongside subjective testing demonstrates that the performance of the proposed protocol is unmistakably superior to the existing DCXP. Packet loss, scalability and Jitter have been considered for the experimentation and results shows that it clearly outperforms the existing DCXP. Moreover, the proposed protocol is distributed in a sense that every node is now responsible for the dissemination and re-transmission of the messages instead of the centralized DCXP proxy server. In future the work can be extended to tested for real scenarios where actual handover can take place.

Author Contributions: S.A. conceived the idea for this paper, designed the experiments and wrote the paper; I.H. and M.F. assisted in model designing and experiments. D.H.K. conceived the overall idea of proposed protocol, and proof-read the manuscript.

Acknowledgments: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00313) supervised by the IITP (Institute for Information and communications Technology Promotion), and this work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00756, Development of interoperability and management technology of IoT system with heterogeneous ID mechanism). Any correspondence related to this paper should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Ericsson, L. *More Than 50 Billion Connected Devices*; White Paper; Ericsson: Stockholm, Sweden, 2011.
3. Rappaport, T.S. *Wireless Communications: Principles and Practice*; Prentice Hall: Upper Saddle River, NJ, USA, 1996; Volume 2.
4. Nikander, P. Evolution of networking: Current problems and future directions. In Proceedings of the Third International Conference on IEEE Security and Privacy in Communications Networks and the Workshops, Nice, France, 17–21 September 2007; p. 518.
5. Moskowitz, R.; Nikander, P.; Jokela, P.; Henderson, T. *Host Identity Protocol*; Technical Report; IETF Network Working Group, 2008. Available online: https://tools.ietf.org/html/rfc5338 (accessed on 23 January 2018).
6. Nikander, P.; Moskowitz, R. Host Identity Protocol (Hip) Architecture, 2006 IETF Network Working Group. Available online: https://tools.ietf.org/html/rfc4423.html (accessed on 25 January 2018).
7. Kafle, V.P.; Inoue, M. Locator ID Separation for Mobility Management in the New Generation Network. *JoWUA* **2010**, *1*, 3–15.
8. Natal, A.R.; Jakab, L.; Portolés, M.; Ermagan, V.; Natarajan, P.; Maino, F.; Meyer, D.; Aparicio, A.C. LISP-MN: Mobile networking through LISP. *Wirel. Pers. Commun.* **2013**, *70*, 253–266. [CrossRef]
9. Farinacci, D.; Lewis, D.; Meyer, D.; Fuller, V. *The Locator/ID Separation Protocol (LISP)*; IETF, 2013. Available online: https://tools.ietf.org/html/rfc6830 (accessed on 2 February 2018).

10. Dong, P.; Zhang, H. MobileID: Universal-ID Based Mobility in Locator/ID Separation Networks. In Proceedings of the 2010 International Conference on IEEE Communications and Mobile Computing (CMC), Shenzhen, China, 12–14 April 2010; Volume 1, pp. 473–477.
11. Castro Casales, A.; Germán Duarte, M.; Yannuzzi, M.; Masip Bruin, X. Insights on the Internet routing scalability issues. In Proceedings of the 1st Workshop on Multilayer Networks, Spain, 2009.
12. Saucez, D.; Iannone, L.; Bonaventure, O.; Farinacci, D. Designing a Deployable Future Internet: The Locator/Identifier Separation Protocol (LISP) case. *IEEE Internet Comput.* **2012**, *16*, 14–21. [CrossRef]
13. Montavont, N.; Noel, T. Handover management for mobile nodes in IPv6 networks. *IEEE Commun. Mag.* **2002**, *40*, 38–43. [CrossRef]
14. Soliman, H.; Castelluccia, C.; Elmalki, K.; Bellier, L. *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*; Technical Report; IETF Network Working group, 2008. Available online: https://tools.ietf.org/html/rfc5380.html (accessed on 2 February 2018).
15. Jung, H.; Kim, E.; Yi, J.; Lee, H. A scheme for supporting fast handover in hierarchical mobile IPv6 networks. *ETRI J.* **2005**, *27*, 798–801. [CrossRef]
16. Pack, S.; Choi, Y. A study on performance of hierarchical mobile IPv6 in IP-based cellular networks. *IEICE Trans. Commun.* **2004**, *87*, 462–469.
17. Ryu, S.; Lee, K.; Mun, Y. Optimized fast handover scheme in Mobile IPv6 networks to support mobile users for cloud computing. *J. Supercomput.* **2012**, *59*, 658–675. [CrossRef]
18. Koodli, R. *Fast Handovers for Mobile IPv6*; The Internet Society: Reston, VA, USA, 2005.
19. Alamri, A.; Ansari, W.S.; Hassan, M.M.; Hossain, M.S.; Alelaiwi, A.; Hossain, M.A. A survey on sensor-cloud: Architecture, applications, and approaches. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 917923. [CrossRef]
20. Zhang, B.; Mor, N.; Kolb, J.; Chan, D.S.; Lutz, K.; Allman, E.; Wawrzynek, J.; Lee, E.A.; Kubiatowicz, J. The Cloud is Not Enough: Saving IoT from the Cloud. In Proceedings of the HotCloud'15—7th USENIX Workshop on Hot Topics in Cloud Computing, Santa Clara, CA, USA, 6–7 July 2015.
21. Galuba, W.; Girdzijauskas, S. Distributed hash table. In *Encyclopedia of Database Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 903–904.
22. Naor, M.; Wieder, U. A simple fault tolerant distributed hash table. In *Peer-to-Peer Systems II*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 88–97.
23. Zhang, H.; Goel, A.; Govindan, R. Incrementally improving lookup latency in distributed hash table systems. *ACM SIGMETRICS Perform. Eval. Rev.* **2003**, *31*, 114–125. [CrossRef]
24. Kanter, T.; Österberg, P.; Walters, J.; Kardeby, V.; Forsström, S.; Pettersson, S. The mediasense framework. In Proceedings of the Fourth International Conference on IEEE Digital Telecommunications, Athens, Greece, 13–19 June 2009; pp. 144–147.
25. MediaSense. Available online: http://mediasense.se/?page_id=45 (accessed on 3 April 2018).
26. P2P with TomP2P. Available online: https://tomp2p.net/doc/M05-1up_v2.pdf (accessed on 4 April 2018).
27. Walters, J.; Kanter, T.; Norling, R. Distributed Context Models in Support of Ubiquitous Mobile Awareness Services. In *Proceedings of the International Conference on Sensor Systems and Software*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 121–134.