This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

IoT-RTP and IoT-RTCP: Adaptive Protocols for Multimedia Transmission over Internet of Things Environments

Omar Said [1], Yasser Albagory [2], Mostafa Nofal [3], Fahad Al Raddady [4]

***Abstract*—Recently, the Internet of Things (IoT) has attracted the interest of network researchers all over the world. Multimedia transmission through IoT presents an important challenge owing to nodes diversity. In this paper, adaptive versions of the Real-time Transport Protocol (RTP) and Real-time Control Protocol (RTCP), i.e., IoT-RTP and IoT-RTCP, are proposed. In these versions, the nature of IoT environments such as transmission channels heterogeneity, sudden change in session size, and different multimedia sources are considered. The basic idea of the proposed adaptive versions is to divide the large multimedia sessions into simple sessions with awareness of network status. To achieve this target, additional fields are added to the RTP and RTCP headers. These fields work under certain conditions to decrease the network overload. Finally, to test the performance of the proposed IoT-RTP and IoT-RTCP, a simulation environment is constructed using the network simulation package (NS2).The results of intensive simulations proved that the proposed adaptive versions of the multimedia protocols outperform the basic ones in terms of end-to-end delay, delay jitter, number of receiver reports, packet loss, throughput, and energy consumption.**

***Index Terms*— IoT, IoT Simulation, Multimedia Communication, RTP/RTCP, Internetworking**

I. INTRODUCTION

Nowadays, the Internet has become available everywhere and has spread significantly faster than any other technology. It facilitates the transmission of information in fast and safe manner through many advanced applications. In addition, it includes tools, machines, and software within standard infrastructure. Everyday devices, such as appliances, lamps, cars, and sensors, can communicate over Internet through a unique Internet Protocol (IP) address. This concept is called Internet of Things (IoT) [1, 2]. Development of old applications and creation of new ones can be achieved using IoT technology. The IoT concept has applications in various fields, and marketing is an important field that uses IoT [3]. In addition, intelligent traffic control also employs IoT for dynamic adaption of signal time and distribution of cars on the roads [4]. Garbage containers with ultrasonic sensors that relay descriptive information without human intervention also function on the IoT concept [5]. Moreover, in the near future, IoT technology will be applied to many other fields, such as healthcare, smart cities, security, e-learning, and military [6-8]. IoT technology functions on the basis of two main features, namely wireless sensor networks (WSNs) and radio-frequency identification (RFID). Most IoT operations such as sensing and data collection are accomplished using WSNs. WSNs are therefore considered to be the core of IoT. Hence, the problems of WSNs, such as limited processing unite, limited storage capacity, limited memory, and short range of communication, may also represent a challenge in IoT. In contrast, RFID is mostly used for passive functions such as tracking and identification [9].

For multimedia communication, the “real time” concept suggests that the media stream should be played out as it is received, instead of being stored in files for playback later. In practice, the network may impose an unavoidable delay, and the play out process should be done at the receiver site instantaneously and synchronously. Therefore, the primary requirement is prediction of the network transit time variation. For example, in IP telephony, the encoded voice can be transmitted in 20 milliseconds. This means that the voice source will transmit one packet/frame. These source packets should arrive at the receiver with the same spacing, and the play out process should be run immediately. Hence, transit time variations may be accommodated by a minor increase in the buffering delay at the receiver site. However, the receiver should be adapted to match the new variation. Naturally, reliable delivery of multimedia streams is desirable, but unfortunately, many multimedia applications cannot tolerate delay due to lost packets. Accordingly, multimedia protocols will alleviate the effect of delay and loss problems. It is well known that application type, encoding method, and loss pattern are three important factors affecting multimedia transmission over networks. However, for IoT, there are other factors affecting the multimedia streams, such as diversity in bandwidth, energy consumption ratio, passive feature, and large number of users that may join and leave multimedia session within a short time period. Therefore, multimedia communication protocols should be proposed to regulate multimedia streams transmission through IoT environments [10].

Recently, the demand for multimedia application over wireless networks has increased. However, wireless communication, which represents the infrastructure of IoT, has limitations in bandwidth, limit processing units, energy consumption, etc. [10]. It is also well known that the load added by multimedia streams is more than the capability of IoT owing to the requirements of higher bandwidth, memory space, and energy. Accordingly, direct implementation of the current versions of RTP and RTCP over an IoT environment

This paper was done as part of a research project number LGP – 35 – 207 which is funded by King Abdulaziz City for Science and Technology (KACST). The authors wish to thank KACST and our co-researchers for their support.

will not be feasible. This is due to lack of sufficient awareness about multimedia session properties such as node power, session size, multimedia streams size, and change in transmitted multimedia type. The main target of this paper is to propose adaptive versions of RTP and RTCP to transmit multimedia streams through IoT while preserving the accepted QoS and taking into consideration the special IoT features.

This paper is organized as follows; In Section II the problem definition of the paper is demonstrated. In Section III, the related works are introduced. The contribution of the paper is showed in Section IV. In Section V, IoT-RTP and IoT-RTCP are proposed. A simple mathematical analysis is introduced in Section VI. Simulation and evaluation of the proposed protocols are demonstrated in Section VII. Finally, conclusion and future work are introduced in Section VIII and Section IX respectively.

II. Problem Definition

Since multimedia communication protocols play a vital role in satisfying the demands of IoT applications, there is a pressing need to upgrade the traditional real time communication protocols. Multimedia protocols that should be extended to be suitable for IoT environment are RTP and RTCP. One of the main challenges in the upgradation of the multimedia protocols is the limitations of IoT systems, which include bottlenecks, heterogeneous environments, high rates of data growth, high frequency of exchanges, no human intervention, dynamic exchange of things positions, and energy restriction of many IoT nodes.

III. Related Work

Several theoretical and practical research studies have achieved multimedia transmission over different networks. Yang Q. introduced a routing scheme to enhance the throughput and decrease the delay for large scale networks. This scheme was proved theoretically, but has not been verified [11]. Delgalvis I. et al. proposed a method for real-time transmission of data through a special network. The buffer requirements for real-time data transfer were studied. Results indicated that the buffer size should be configured based on the length of multimedia packet and the status of network traffic. The drawback of this method is that it has strict requirements for transmitting the multimedia streams [12]. D. Francesco et al. demonstrated a storage infrastructure data model for IoT that considered IoT properties such as node diversity as well as notable increased number of participating devices [13]. Moreover, those results provided adaptive models for uploading multimedia files over IoT devices. Zhang H. et al. proposed a meta-model for real-time exchange using peer-to-peer connections. The warehouse meta model and Tuxedo middleware technology were used in this model, and it showed good performance in terms of scalability and robustness. The main drawback of this model is its dependency on the peer-to- peer concept [14]. Chilingarya S. et al. proposed a middleware to be applied for real-time communication as well as different data rates. This middleware is compatible with different platforms and exhibits heterogeneous environment interoperability and high performance. However, this middleware is not adapted with IoT environment [15]. Guan H. et al. demonstrated a real-time data model using the web service concept, and the model components were determined and analyzed. The disadvantage of this model was its low efficiency [16]. Zhou L. et al. introduced a model to maximize the mean opinion score (MOS) by power adjustment for each application. In addition, multimedia communication over IoT environment was also considered [17]. There are many studies related to the security of multimedia transmission through IoT systems [18, 19]. Danilo S. et al. presented a system of personal health devices (PHDs) based on IoT. This system adapted the Constrained Application Protocol (CoAP) and used it as the main model for system data exchange. Moreover, the communication method of the proposed system with other healthcare systems was demonstrated, and the system performance was evaluated and discussed [20]. Eleonora B. provided an IoT survey that included various technologies, features, applications, and scenarios of the IoT. Furthermore, it focused on problems occurring in such IoT environments. Open issues and research challenges in this area have also been introduced [21]. Ibrahim M. et al. introduced a more extended survey about IoT, Web of Things (WoT), and Social Web of Things (SWoT). This survey comprised architecture design, middleware, platform, system implementation, and applications of IoT, WoT, and SWoT. The challenges in these types of networks were also discussed [22]. Aijaz A. et al. studied machine-to-machine (M2M) communication from the perspective of protocol stacks. Their study covered the latest efforts and developments in the M2M communication protocols. Moreover, a routing protocol, medium access control (MAC) protocol, and distributed cognitive MAC protocol were introduced for M2M networks. Their work was evaluated in another study [23]. Pan J. et al. used the cloud computing concept and smart-phone paradigm to construct an IoT-based framework for energy control. The proposed framework was also analyzed, designed, and simulated [24]. Vasileios K. et al. compared the protocols involved in communication over Internet. In addition, application layer protocols for IoT were discussed. Some protocols such as IETFs CoAP, IBMs MQTT, and HTML have been highlighted. The suitability of these protocols for IoT in terms of energy consumption, security, and reliability has also been studied [25].

Jiang W. et al. introduced the most closely related work to this paper. They demonstrated a platform for IoT multimedia. This study adapted UDP to work within an IoT environment. In addition, they introduced a model for real-time exchange over the IoT environment. The main drawback of this platform is its inaccurate representation of the IoT environment. In addition, the results did not include many important performance parameters such as delay jitter. The platform only introduced an adaptive version of UDP, and did not consider the functions of the application layer in the multimedia communication [26, 27]. Jingwu C. et al. proposed a system for scheduling multimedia transmission over an IoT environment under an energy constraint [28, 29]. The main drawback of this system is absence of IoT properties such as a heterogeneous environment. Sheeraz A. et al. [30] only focused on communication between multimedia objects and presented a new paradigm called Internet of Multimedia Things (IoMT), in which different multimedia things can interact and cooperate with each other over the Internet. Their research discussed challenges and requirements for this paradigm architecture, but did not study the method for multimedia data transmission in the IoT system. Furthermore, it did not include multimedia transmission protocols. Pereira R. et al. tested suitability of applying the H.264 standard of video encoding in the IoT environment [31]. Scalable video encoders were discussed based on the special properties of IoT transmission channels and devices that have limited capabilities and many restrictions. Their research was concerned only with video encoding did not include adaptation of multimedia packets' structure and QoS parameters. The method for transmission of these packets in the IoT environment was not discussed.

Several methodologies have been designed real-time transmission of streams over wireless communications. Most of these methods include coder modification through either multi description approaches [32] or adaptation techniques [33] or a combination of both [34]. Other methods used RTP and routing protocol [35]. Cross layer approaches have also been used for real-time transmission of streams [36]. Furthermore, the Adaptive Multiple Description Coding Protocol (AMDCP) has been designed for transmitting video streams over the Internet by using an MDC coder [37]. A combination of previous methodologies has been used by many protocols such as the Multi-flow Real-time Transport Protocol (MRTP) [38], which was designed for ad hoc networks. MRTP uses a combination of MDC and transport protocol techniques, and is considered a session-oriented protocol that divides multimedia streams into flows such that each flow is sent on a standalone path. Several parameters are required to be determined in the connection setup process, such as the number of flows that should be used. Further, it generates a control report to describe the transmission state in a statistical view. It also reorders packets received from different flows. However, the main drawback of this protocol is that is constructed only for ad hoc networks and is not suitable IoT systems owing to issues with scalability, diversity, and no human intervention. In addition, it is a session-oriented protocol, and this does not represent a multimedia transmission over IoT. The energy level for network nodes, which is a main parameter in the IoT, is not included in this protocol. The Adaptive Multi-flow Real-time Multimedia Transport Protocol (AdamRTP) for WSNs was introduced [39] to modify RTP and RTCP for WSN systems. AdamRTP alleviates the congestion and guarantees good QoS, in addition to providing acceptable energy consumption level, for multimedia transmission over WSN system.

IV. PAPER CONTRIBUTION

The main target of this paper is to develop two adaptive versions of RTP and RTCP protocols to transmit multimedia streams over an IoT environment, while maintaining acceptable QoS. It is well known that IoT may have limited resources such as bandwidth and power. Hence, the proposed versions should divide the multimedia session into small parts such that each part can manage itself. This will help to avoid the multimedia stream flooding problem. In addition, each multimedia stream should be divided into number of flows such that each flow will be defined by its small session. Then, transmission of these multimedia flows through IoT channels should be determined such that each flow will be sent over a separate path. Accordingly, the phenomenon of overwhelming of intermediate nodes would be greatly reduced. Moreover, the rate of transmission in each small session should be adapted dynamically according to the IoT system state. To achieve all of these requirements, the proposed adaptive versions, namely IoT-RTP and IoT-RTCP, should be aware of the IoT system state.

V. PROPOSED IOT-RTP AND IOT-RTCP

In this section, adaptive versions of RTP and RTCP, which are referred to as IoT-RTP and IoT-RTCP, are presented. These versions consider the current state of the multimedia session in the IoT system, such as the number of participants, type of participants, and coding of multimedia streams. The adaptive versions of RTP and RTCP are considered as application layer protocols with some transport layer functions. Therefore, the header of these protocols should be added to the multimedia payload after the transport and network layer headers.

A. IoT-RTP

The main challenge in multimedia transmission through IoT systems is the component diversity, which may lead to creation of bottlenecks. These bottlenecks may affect the transmission of multimedia flow owing to their sensitivity to delay and loss. Furthermore, changing of multimedia coding during transmission process is another challenge. The size of multimedia session in the IoT systems may also be suddenly increased or decreased as a result of fast joining or leaving of participants. Therefore, the adaptive version of RTP should accommodate many changes in the traditional RTP version in order to function irrespective of the IoT environment challenges. These changes are related to multimedia session, energy consumption ratio for each routing path nodes, thing type (passive or active), thing state, and multimedia stream prioritization; see Fig. 1. The required changes are described in the following sub-sections.

- *Multimedia Session*

The multimedia sessions in the IoT systems differ from the multimedia sessions in other systems, even the Internet. The main difference is scalability, which refers to the property that a large number of things may join the multimedia session within short period. This means that a large number of multimedia flows may be sent through the session. This leads to large bandwidth requirement, which represents a challenge owing to bandwidth limitations in many IoT sectors. The IoT-RTP deals with the IoT multimedia session as small parts. Hence, the multimedia session should be divided into simple sessions. Then, these simple sessions may be divided again to simpler sessions (if required). The multimedia streams should be divided into flows that are further divided into small flows (sub-flow) in case of bandwidth limitation. Each sub-flow should be identified by integer number followed by its original flow identifier.

- *Energy consumption*

Energy is the main challenge for multimedia transmission over IoT systems. The IoT systems may comprise passive and active nodes. This means that the multimedia data may be transmitted through energy-based nodes or normal nodes. With regards to normal nodes, the energy is not an important factor. For the energy-based nodes, IoT- RTP should be aware of the state of these types of nodes. Therefore, a field should be added in the IoT-RTP to record the energy level for each node in the multimedia session. However, keeping the energy level for each node consumes a large number of bytes in the header, and so, two new fields should be added in the header to handle the state of each node. The first field is used to record the energy level for the nodes that has multimedia stream. The second field is used to record the nodes that have critical energy levels. First, the energy level for all nodes is read, and then, only the critical values are saved. The critical level of energy is determined depending on many parameters such as node state, type, and multimedia coding. These two fields will guarantee awareness of energy level in the multimedia session.

- *Thing type*

The main difference between IoT systems and other traditional systems is that the IoT systems may have passive things. The passive things should be determined to decrease the load on the system. This is because passive things may not have the ability to pass or receive multimedia streams. The IoT-RTP should have a header field to determine whether the type of node in the multimedia session is passive or active.

- *Things state*

There are three thing states: active, sleep or failed. The IoT-RTP header should comprise a field to determine the current state of things provided that this thing is found in the multimedia session. The rate of multimedia transmission and reception may differ from an active state to a sleep state. This will be useful to alleviate the overload problem that may result from a large number of participants in the multimedia session. The active state differs from the active type. Active state means that the IoT node is in work, but the active type means that the node infrastructure has a processing unit.

- *Priority*

The multimedia session in IoT systems suffers from dynamic scalability, which leads to multiple problems such as delay and loss. Further, an increase in the number of multimedia participants leads to an increase in multimedia transmission, which in turn leads to congestion and many unstable states in the multimedia session. Therefore, in the case of a large multimedia session, the multimedia streams should be prioritized. The prioritization process should be achieved based on many parameters, such as size, importance, and coding. The technique reported in [40] can be used to achieve the prioritization process.

- *V* is the version of IoT-RTP. The size of this field is 3 bits.
- *PT* is the payload type of IoT-RTP. The size of this field is 7 bits.
- *Session ID* is the identifier of multimedia session in IoT-RTP. The size of this field is 10 bits.
- *Sub-ID* is the sub-session identifier. The size of this field is 3 bits.
- *Sub- Sub-ID* is the sub-sub-session identifier (the second level of session division). This field size is 3 bits.
- *Flow ID* is the flow number in the transmitted multimedia. This field size is 3 bits.
- *Sub-FID* is the *Sub*-flow number in the transmitted flow. The size of this field is 2 bits.
- *Sub-Sub-FID* is the *Sub-Sub*-flow number in the transmitted *Sub*-flow. This field size is 2 bits.
- *Node Counter* is the number of nodes that are used to receive and transmit the multimedia flows from source to destination. The size of this field is 30 bits.
- *Thing Type* determines if the IoT node is active or passive. The size of this field is 2 bits.
- *Normal Energy Level* contains the energy level for normal nodes. The size of this field is 16 bits.
- *Critical Energy Level* contains the energy levels for the critical energy nodes. The size of this field is 16 bits.
- *Thing State* represents the state of each IoT node: active, sleep, or failed. The size of this field is 2 bits.
- *Priority* determines the importance of each multimedia stream. The size of this field is 2 bits.
- The time stamp and extended sequence number are described in the literature [39].

It should be noted that there are many additional fields in the IoT-RTP. These additional bits may represent an overload of the proposed IoT-RTP version and may affect the transmission of multimedia streams through the IoT environment. To solve this problem, the additional IoT-RTP fields should be sensitive to the IoT system status. In case of stability in the IoT system, the fields will be not activated, but in case of IoT system starvation, these additional fields should be reactivated gradually to describe the actual states of the IoT system, multimedia session, and things. In addition, the size of additional fields is taken from old files bits. So, there are additional fields are added in the IoT-RTP.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

<table>
<tr><td>0</td><td>1</td><td>2</td><td>4</td><td>--</td><td>8</td><td>9</td><td>--</td><td>18</td><td>19</td><td>--</td><td>21</td><td>22</td><td>--</td><td>24</td><td>25</td><td>--</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="2">V</td><td colspan="4">PT</td><td colspan="3">Session ID</td><td colspan="3">Sub-SID</td><td colspan="3">Sub-Sub-SID</td><td colspan="3">Flow ID</td><td colspan="2">Sub-FID</td><td colspan="2">Sub-Sub-FID</td></tr>
<tr><td colspan="20">Node Counter</td><td colspan="2">Thing Type</td></tr>
<tr><td colspan="14">Normal Energy Level</td><td colspan="8">Critical Energy Level</td></tr>
<tr><td colspan="18">Extended Sequence Number</td><td colspan="2">Thing State</td><td colspan="2">Priority</td></tr>
<tr><td colspan="22">Time Stamp</td></tr>
</table>

Fig. 1: IoT-RTP header.

- *Adaptation strategy*

RTP requires several changes in order to be compatible with the IoT environment. As stated above, the multimedia session should be divided into simple sessions. If any simple multimedia session is still too large to service, it should be further divided into more simple sessions. The division process continues until the network resources are sufficient for the resulted simpler session. The session id, simple session id, and simpler session id are saved in the IoT-RTP. The IoT-RTP permits for only three division levels in the multimedia session. In the future, additional division levels may be considered. By using the same idea, the multimedia streams are divided into simpler streams. Similarly, the IoT-RTP also permits for multimedia flows to be divided two times and saves the resulting sub-flow ID. In case of problems with the size of multimedia sessions or a large number of flows, the simpler sessions (sub-sessions) should decrease sending multimedia streams until the IoT system returns to the normal state. The node counter is used to determine the number of nodes that the multimedia data will visit through its trip. A large number of nodes indicates that the multimedia streams may face more diversity in the nodes, which means that the IoT-RTP additional fields may be reactivated. The thing type field is used to determine if the node is active or passive. This is useful for accurate determination of multimedia size that may be transmitted in a specific session. This is because the passive nodes send restricted small sizes of multimedia streams and mostly do not receive multimedia streams. The energy level should be saved for each node that may be involved in the multimedia trip. There are two types of energy levels, namely critical and normal. The normal energy level determines the energy value for nodes that have sufficient power to achieve their functions within a specific period, which is determined by IoT administrative application. The critical energy level field determines the energy value for the nodes that have low energy and may be excluded from the routing path. The thing state field is used to determine if the node is in active, sleep, or failed state. The priority field is used to arrange the importance of each multimedia flow. When the IoT system is in the stable state, all multimedia flows will have the same priority. However, when the IoT system is in starvation, the priority field will be activated to decrease the number of flows that can be transmitted through the IoT system, see Fig. 2. Also, see IoT-RTP descriptive algorithm for more clarification.

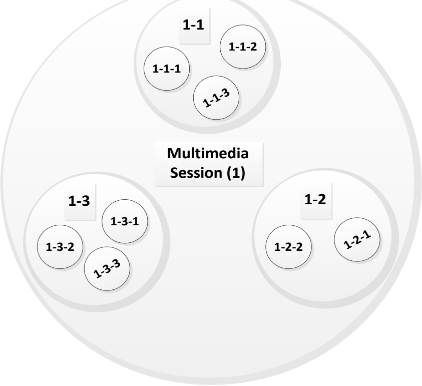

Fig. 2: General view of the multimedia session in IoT-RTP.

IoT-RTP Descriptive Algorithm

```
N is the number of users in the multimedia session
M is the number of sub-sessions
L is the number of sub-sub sessions
D is the network capacity
$D_M$ is the network capacity after session division.
V is the number of sub-flows.
IF ($\sum_{i=1}^{N} s_i <= D$)
        $normal\ state$
Else
  For I = 1 to M
    Begin
      IF ($\sum_{i=1}^{N/M} s_i <= D_M$)
        $normal\ state$
      Else
          $F_i = s_i/V$
      IF ($\sum_{i=1}^{N/M} F_i <= D_M$)
        $normal\ state$
      Else
        For j = 1 to 4
          $F_{ij}\ should\ be\ tranmitted$
    End
End of Algorithm
```

B. IoT-RTCP

The adaptive version of RTCP also considers the state of the IoT system with regards to the transmission of multimedia streams. In addition, its reports collect information about things that may be found in the IoT systems and that may differ from other systems. This version also considers the type of nodes (active or passive). The specs of each thing in the IoT system, such as processing, memory and energy, should be considered. Moreover, different multimedia coding should be considered. Minimization of RTCP reports without affecting the multimedia transmission through IoT systems is an important target in IoT-RTCP, especially in case of network starvation. To achieve this target, prioritization of control reports should be applied. In the traditional RTCP, there are two main types of reports, namely sender report (SR) and receiver report (RR). SR comprises many variables such as the number of transmitted packets within a period, Network Time Protocol (NTP) timestamps (NTPTs), and synchronization sources (SSRCs) [41]. RR comprises many fields such as fraction lost (FL), estimated number of packets expected (NPE), and inter-arrival jitter. [41]. In IoT-RTCP, SR and RR should be upgraded by adding many fields to gather specific information about the IoT system. In addition, these reports should be transmitted under restricted conditions so as to minimize the overloading of the additional fields. As stated above in the basic idea of IoT-RTP, IoT-RTCP also divides the multimedia session into a group of simple sessions. Each session has a manager that is selected using the mechanism reported in [42]. The division of multimedia session processes should be restricted with session size. If the multimedia session size is larger than a predetermined threshold, the division should be executed until the simple session reaches a normal size (i.e., the size at which the IoT resources can service the session nodes in a safe manner). However, in the case of a normal multimedia session, the division process will be neglected. The threshold and normal sizes are described in the simulation section. The changes that should be applied in the SR and RR are described below.

- *SR upgrades*

There are five fields that should be added to the header of SR reports. These fields are sender type, sender mode counter, sender energy level, sender state, and sender transmission rate. These fields are stated in Fig. 3 and described below.

- *Sender type field* determines if the thing is active or passive. Active thing means that it contains a processing unit and memory. Active things may send multimedia streams without restriction, but the passive things need interaction from an active thing to send multimedia streams. For example, suppose RFID is attached to a passive thing, such as trashcan. The trashcan may send its information to a predetermined user after interaction with a sensor that receives an order autonomously from the IoT application manager. The size of this field is 3 bits.
- *Sender mode counter field* determines how many times the sender was in the active mode; in addition to how many times the sender was in the sleep mode. The size of this field is 3 bits.
- *Sender energy level field* determines the energy value of the sender within a specified interval. Energy is an important metric in the IoT system. Therefore, retaining of the energy value for each node helps in computing the consumption of energy required to send a specific type of multimedia coding. The size of this field is 8 bits.
- *Sender state field* determines if the sender can receive multimedia streams (only sender or sender/receiver). Each node in the IoT system may be sender, receiver, or both. Determining each node's status gives an opportunity to decrease the number of transmitted multimedia streams in the case of network starvation. This is because the prioritization process will consider this field to arrange multimedia streams from the most to least importance. The size of this field is 3 bits.
- *Sender transmission rate field* determines the number of times that the sender uses the multimedia session. This helps in determining the importance of that sender in the multimedia session (i.e., more sending means more importance and vice versa). The size of this field is 6 bits.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>….</th><th>7</th><th>8</th><th>….</th><th>11</th><th>12</th><th>….</th><th>15</th><th>16</th><th>….</th><th>19</th><th>20</th><th>….</th><th>27</th><th>28</th><th>….</th><th>31</th><th></th></tr>
<tr><td colspan="2">V</td><td></td><td colspan="2">RC</td><td colspan="6">PT = SR = 200</td><td colspan="9">Length</td><td></td></tr>
<tr><td colspan="20">SSRC of Sender</td><td></td></tr>
<tr><td colspan="20">NTP timestamp, most significant word</td><td></td></tr>
<tr><td colspan="20">NTP timestamp, least significant word</td><td></td></tr>
<tr><td colspan="20">RTP timestamp</td><td></td></tr>
<tr><td colspan="20">Sender's packet count</td><td></td></tr>
<tr><td colspan="20">Sender's octet count</td><td></td></tr>
<tr><td colspan="20">SSRC_1 (SSRC of first source)</td><td></td></tr>
<tr><td colspan="7">Fraction lost</td><td colspan="10">Cumulative number of packets lost</td><td colspan="3">Sender Type</td><td rowspan="4">Block 1</td></tr>
<tr><td colspan="17">Extended highest sequence number received</td><td colspan="3">Energy level</td></tr>
<tr><td colspan="14">inter-arrival time jitter</td><td colspan="3">Sender mode counter</td><td colspan="3">Sender state</td></tr>
<tr><td colspan="17">Last SR (LSR)</td><td colspan="3">Sender transmission rate</td></tr>
<tr><td colspan="20">Delay since last SR (DLSR)</td><td></td></tr>
<tr><td colspan="20">SSRC_2 (SSRC of first source)</td><td></td></tr>
<tr><td colspan="17">Cumulative number of packets lost</td><td colspan="3">Sender Type</td><td rowspan="4">Block 2</td></tr>
<tr><td colspan="17">Extended highest sequence number received</td><td colspan="3">Energy level</td></tr>
<tr><td colspan="14">inter-arrival time jitter</td><td colspan="3">Sender mode counter</td><td colspan="3">Sender state</td></tr>
<tr><td colspan="17">Last SR (LSR)</td><td colspan="3">Sender transmission rate</td></tr>
<tr><td colspan="20">Delay since last SR (DLSR)</td><td></td></tr>
</table>

Fig. 3: SR header in IoT-RCTP.

- *RR upgrades*

Six fields should be added to the header of RR reports. These fields are session ID, number of sub-sessions, flow ID, number of sub-flows, number of active things, and number of passive things. These fields are stated in Fig. 4 and described below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

- *Session ID field* determines the session identifier in the case of original and divided sessions. This field should therefore be divided into three levels: session ID, sub-session ID, and sub-sub-session ID. For decreasing the used header bits, the last two sub fields should remain in passive state until the division process activated. The size of this field is 10 bits.
- *Number of sub-sessions field* determines the final number of small sessions resulting from the division process. The size of this field is 6 bits.
- *Flow ID field* determines the flow identifier in the case of a normal multimedia stream. This field should therefore be divided into three levels: flow ID, sub-flow ID, and sub-sub-flow ID. For decreasing the used header bits, the last two sub fields should be activated after the division process occurs. The size of this field is 4 bits. This filed may be used only for cumulative RR report.
- *Number of sub-flow field* determines the number of flows after the flow division process. The size of this field is 3 bits.
- *Number of active things field* determines the number of active things in the multimedia session. Using this number may provide an accurate expectation of the number of multimedia flows that will be transmitted in the near future. The size of this field is 6 bits.
- *Number of passive things field* determines the number of passive things in the multimedia session. Decreasing this number may help in acceleration of the transmission process. The size of this field is 6 bits.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>…</th><th>7</th><th>8</th><th>…</th><th>11</th><th>12</th><th>….</th><th>15</th><th>16</th><th>….</th><th>19</th><th>20</th><th>….</th><th>27</th><th>28</th><th>….</th><th>31</th><th></th></tr>
<tr><td colspan="2">V</td><td colspan="5">RR Count</td><td colspan="7">Packet Type</td><td colspan="2">Seq. No.</td><td colspan="4">No. of Flows</td><td></td></tr>
<tr><td colspan="7"><i>Session ID</i></td><td colspan="4"><i>No. of sub-Sessions</i></td><td colspan="5"><i>Flow ID</i></td><td colspan="4"><i>No. of sub-Flows</i></td><td rowspan="5">Block 1</td></tr>
<tr><td colspan="4"><i>Fraction Lost</i></td><td colspan="7"><i>Cumulative number of lost packets</i></td><td colspan="5"><i>No. of Passive Things</i></td><td colspan="4"><i>No. of Active Things</i></td></tr>
<tr><td colspan="20"><i>Average Energy</i></td></tr>
<tr><td colspan="20"><i>Extended highest sequence number</i></td></tr>
<tr><td colspan="20"><i>Estimate inter-arrival time jitter</i></td></tr>
<tr><td colspan="7"><i>Session ID</i></td><td colspan="4"><i>No. of sub-Sessions</i></td><td colspan="5"><i>Flow ID</i></td><td colspan="4"><i>No. of sub-Flows</i></td><td rowspan="5">Block 2</td></tr>
<tr><td colspan="4"><i>Fraction Lost</i></td><td colspan="7"><i>Cumulative number of lost packets</i></td><td colspan="5"><i>No. of Passive Things</i></td><td colspan="4"><i>No. of Active Things</i></td></tr>
<tr><td colspan="20"><i>Average Energy</i></td></tr>
<tr><td colspan="20"><i>Extended highest sequence number</i></td></tr>
<tr><td colspan="20"><i>Estimate inter-arrival time jitter</i></td></tr>
<tr><td colspan="20">-----</td><td></td></tr>
<tr><td colspan="20">-----</td><td></td></tr>
<tr><td colspan="20">Block n for sub-flow n</td><td></td></tr>
</table>

Fig. 4: RR header in IoT-RCTP

- *Adaptation strategy*

The adaptive strategy of IoT-RTCP is based on two factors. The first factor is the decrease in the number of bits that may be used in SR and RR headers. The second factor is the collection of the most required information about the IoT multimedia session that enables fast and accurate reaction against the transmission problems. Decreasing the RR bits comprises two mechanisms. The first one is a decrease in the number of bits in the RR header. The second mechanism is a decrease in the number of RRs that may be sent through the IoT system within a predetermined period; see Fig. 5. To decrease the RR header bits, the state of the network should be considered. To determine the network state, many metrics such as congestion, delay, and loss should be periodically evaluated. If the IoT metrics are in a normal state, the additional SR and RR fields will not be activated. However, in the case of a decrease in the value of the performance metrics, the fields will be transformed to the active state to generate real-time feedback about the IoT system. With regards to the second mechanism, the number of RRs transmitted can be decreased by increasing the interval in which the RRs should be sent. This interval is determined depending on the network state and the importance of RR that should be sent in this interval. The RTCP header should be changed in order to gather the most important data about the multimedia session. For simplicity, the additional fields in the header should not consume additional bits. To achieve this target, many of the old fields in the RTCP header are further divided into several fields to acquire the traditional required information in addition to the urgent information required from the IoT multimedia session.

Another aspect of the IoT-RTCP is the abstraction of RR reports. This is achieved by collecting multiple RRs in one RR; see Fig. 6. As stated above, the multimedia session is divided into simple sessions. Therefore, there are three types of RTCP agents: simple, middle, and master. The simple agent is responsible for feedback extraction from a small multimedia session. The middle agent is responsible for gathering the feedback from simple agents. The master agent is responsible for gathering feedback from middle agents. Each RTCP agent filters the feedback information to decrease the RR overhead. In the filtering process, each RTCP agent saves urgent data and neglects trivial data. The urgent data is determined by the IoT system administrator and can be changed periodically. The filtering process can be achieved using the technique reported in [43]. Number of levels in the IoT-RTCP can be changed depending on the multimedia session size. Therefore, if the number of users in the session decreases, the levels of IoT-RTCP agents may be decreased and vice versa. Collecting and merging of RR reports into one report can be achieved using the technique reported in [44]. See IoT-RTCP Descriptive Algorithm for more clarification

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

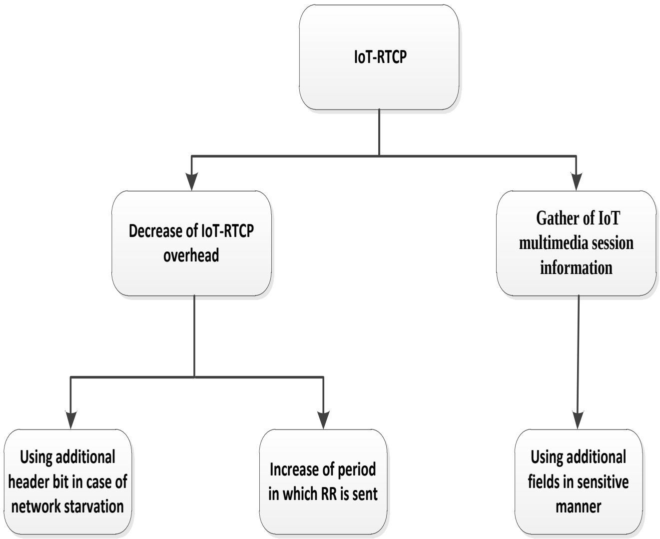

Fig. 5: IoT-RTCP idea

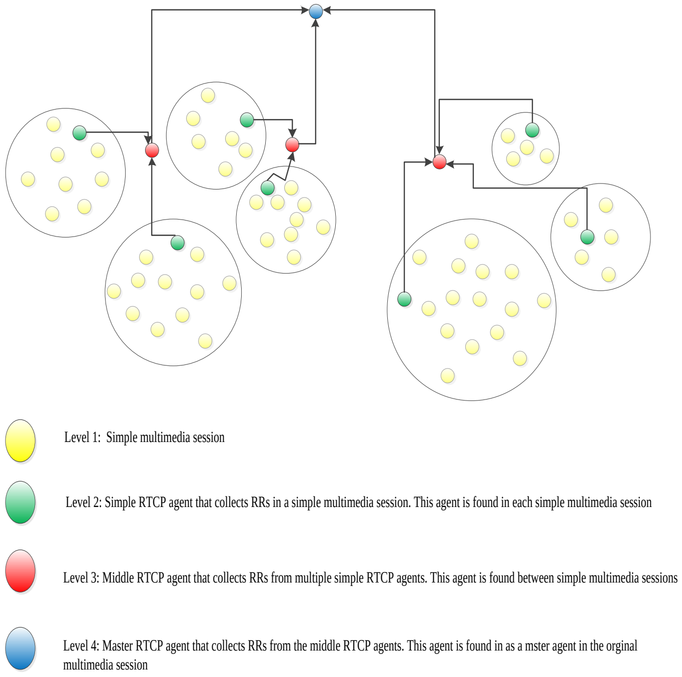

Fig. 6: General view of IoT-RTCP.

IoT-RTCP Descriptive Algorithm

```
PT: Predetermined time period in which RR should be sent.
Z: Upper level of multimedia session division.
M: Middle level of multimedia session division.
N: Lower level of multimedia session division.
ED: End-to-End Delay.
PL: Packet Loss.
JD: Delay Jitter.
RR: Receiver Report.
If T< PT
  Begin
    For H = 1 to Z
      Begin
        For J = 1 to M
          Begin
          For I = 1 to N
            Begin
              IF (ED, PL, JD, are normal values)
                Begin
                  Additional fields of IoT-RTCP are stopped
                  Send RR
                  $RR_J = RR_J + RR_I$
                  End
              Else
                Begin
                  Increase PT value
                  Reactivate the additional fields of IoT-RTCP
                  Send $RR_I$
                  $RR_J = RR_J + RR_I$
                  End
            End
          $RR_H = RR_H + RR_J$
          End
      $RR_Z = RR_Z + RR_H$
      End
  End of Algorithm
```

Important note: It’s notable that there is a redundancy in some fields between IoT-RTP and IoT-RTCP. To clarify this point, the redundant SR fields, sender type and sender energy level, should be neglected in IoT-RTP for the senders of multimedia streams. In addition, the redundant RR fields, which found in the IoT-RTP header, are required in the merging of RRs in one general RR report. But, if the IoT system is starved, the fields in IoT-RTP, which cause redundancy, should not be activated until the network status is changed. This is due to sending of the IoT-RTCP reports can be controlled more than the IoT-RTP message which positively affect the control bits (i.e. the interval between RR transmissions can be increased in case of IoT system starvation which decreases the load of IoT control bits by decreasing of transmitted RRs number).

VI. MATHEMATICAL ANALYSIS

As stated above that the basic idea of the two adaptive protocols, IoT-RTP and IoT-RTCP, is division of big size multimedia session into small size sessions. In addition, this division process be continues until the available services in the IoT network can cover the need of multimedia streams for safe transmission. Many factors such as load balance, distances between nodes, and similarity of nodes are considered in the division process. Similarity between nodes is defined as follows; the nodes, which send the same type of multimedia streams at the same rate form adjacent places through analogous channels, are called similar. Also, the division process comprises the multimedia stream. But, the idea of multimedia streams’ division is travail and can be accomplished using the same method of the fragmentation process. So, the division of multimedia session is a real challenge and must be mathematically analyzed as an optimized problem.

In this simple mathematical analysis, three main topics are covered. These topics are stated as follows: update the centroid of each output cluster, the probability of node(s) to be outside of any cluster, and the calculation of distance between nodes as it is considered the basic parameter of the division process.

A. *Updating of cluster considered*

The start point in the division process (cluster formulation) is to find the centroid of each cluster. Suppose that B clusters C_1, C_2, C_3, …, C_B have centroids r_1, r_2, r_3, …, r_B. The k-means [45] is used to reassign the centroids for the new clusters. The k-means objective function is stated in equation 1

$$F(r_j, C_j) = \sum_{j=1}^{B} \sum_{y_i \in C_j} \left\| y_i - r_j \right\|^2 \tag{1}$$

The centroid r_j can be updated for C_j cluster using equation 2.

$$r_j = \frac{1}{|C_j|} \sum_{yi \in C_j} y_i \tag{2}$$

The distances between nodes and their centroid are calculated using Euclidean distance [46] in equation 3.

$$\begin{aligned} D(y_i, y_j)^2 &= \left\| y_i - y_j \right\|_2^2 \\ &= \sum_{i=1, j=i+1}^{z} (y_i, y_j)^2 \\ &= (y_i - y_j) \,.\, (y_i - y_j) \end{aligned} \tag{3}$$

Where z is the number of nodes in the cluster and $(y_i - y_j) \,.\, (y_i - y_j)$ is a dot product of vectors y_i and y_j. So, y_t node becomes inside cluster C_t if $\left\| y_t - r_j \right\|_2^2 \leq \| y_t - r_l \|_2^2 \ \forall$ t ≠ l.

As stated above that continuous updating of centroid node of each cluster should be considered especially for new generated ones. Suppose that $\acute{r}_t$ is the new centroid of a new cluster C_t. After the division process, the distance between nodes and the new centroid will be decreased. Hence, the objective function should be minimized, see equation 4.

$$\begin{gathered} \Delta F = \sum_{yi \in C_t} \| y_i - \acute{r}_t \|_2^2 - \sum_{yi \in C_t} \| y_t - r_t \|_2^2 = \\ \sum_{yi \in C_t} [\| y_i - \acute{r}_t \|_2^2 - \| y_t - r_t \|_2^2 \ ] = \sum_{yi \in C_t} - \| \acute{r}_t - \\ r_t \|_2^2 - \sum_{yi \in C_t} 2(y_i - \acute{r}_t). (\acute{r}_t - r_t) = -|C_t| \| \acute{r}_t - \\ r_t \|_2^2 - 2(\acute{r}_t - r_t). \sum_{yi \in C_t} 2(y_i - \acute{r}_t) = -|C_t| \| \acute{r}_t - \\ r_t \|_2^2 \leq 0 \end{gathered} \tag{4}$$

B. *Probability of multimedia node to be outside of any cluster*

In this mathematical analysis hierarchical clustering [46] is used to build a tree in which each partition is defined by its predecessor. Suppose that a number of multimedia nodes are distributed in the IoT environment and construct multimedia session. The division process will divide this session into two sessions. The division process starts when find a dissimilarity of nodes after test the distance metric. The size of each output cluster should be re-calculated and the division process will be continued until the multimedia streams find their requirements or each cluster comprises only one node. Join of new nodes to multimedia clusters are achieved based on similarity factor.

Let Y= $\{y_1, \ y_2, \ y_3, \ …, \ y_A\}$ are the set of nodes in the multimedia sessions. To define how the partition process will be accomplished, the clustering methodology will be applied. Hence, the nodes set will be divided into B clusters such that each cluster contains similar nodes. The similarity of nodes based on nature, specs, multimedia streams types, transmission rate, and node type (active or passive). As stated above, these parameters are considered a second category and come after the distance between nodes which is considered as a first category. A multimedia session, M (B), will be divided into a set of clusters equals B $\{C_1, C_2, C_3, …, C_B\}$, 0 < B < A, see equations 5 and 6.

$$\cup_{j=1}^{B} C_j = \mathrm{Y},\ C_j \neq 0 \tag{5}$$

$$C_i \cap C_j = 0,\ \mathrm{i} \neq \mathrm{j} \tag{6}$$

The objective function, which determines if a set of users can be out of clusters after division process, is stated in equation 7.

$$\left| \emptyset(Y, B) \right| = S_A^B = \frac{1}{B!} \sum_{j=0}^{B} (-1)^{B-j} \binom{B}{j} j^A \tag{7}$$

Where $\emptyset(X, B)$ defines the probability of one node is out of the set of all possible output clusters.

C. *Different estimations of distance between multimedia nodes*

As stated above that the distance is the basic factor in the division of multimedia session. So, the distance should be tested internally between nodes. Generally, the nodes $y_h, y_k, y_e, …$ and y_l are supposed to be in a cluster. Hence, to divide these nodes into many clusters, the distances between these nodes are calculated using equation 7.

$$\begin{gathered} \mathrm{D}(y_h, y_l) = \alpha_{\mathrm{k}}\, \mathrm{D}(y_k, y_l) + \alpha_{\mathrm{e}}\, \mathrm{D}(y_e, y_l) + \beta_{\mathrm{k}}\, \mathrm{D}(y_k, y_e) + \\ \gamma \ |\mathrm{D}(y_k, y_l) - \mathrm{D}(y_e, y_l)| \end{gathered} \tag{8}$$

Where α_{k}, α_{e}, β, and γ should take many values to determine all of the distances between nodes in a small cluster. So, if $\alpha_{\mathrm{k}} = \alpha_{\mathrm{e}} = 1/2$, $\beta = 0$, and $\gamma = -1/2$, the distance in one cluster is calculated using equations 9 and 10. Also there are many methods to measure the distance between the nodes in the IoT system. These methods are stated in table 1.

$$\begin{gathered} \mathrm{D}(x_h, x_l) = 1/2\ (\mathrm{D}(x_k, x_l) + \alpha_{\mathrm{e}}\, \mathrm{D}(x_e, x_l))\ \text{-}\ 1/2 \\ |\mathrm{D}(x_k, x_l) - \mathrm{D}(x_e, x_l)| \end{gathered} \tag{9}$$

$$\mathrm{D}(x_h, x_l) = min\ (\mathrm{D}(x_k, x_l) + \alpha_{\mathrm{e}}\, \mathrm{D}(x_e, x_l)) \tag{10}$$

TABLE 1
SAMPLE OF METHODOLOGIES TO MEASURE THE DISTANCES BETWEEN NODES INTERNALLY IN ONE CLUSTER

Methods	Single-Linkage	Un-Weighted average	Median	Group Average	Centroid
α_{k}	0.5	0.5	0.5	$e_i/(e_i + e_j)$	$e_i/(e_i + e_j)$
α_{e}	- 0.5	0.5	0.5	$e_j/(e_i + e_j)$	$e_j/(e_i + e_j)$
β	0	0	-0.25	0	$e_i e_j/(e_i + e_j)^2$
γ	0.5	0	0	0	0

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

VII. SIMULATION AND EVALUATION

In this section, the simulation setup that defines the IoT environment and that is used to test IoT-RTP and IoT-RTCP is proposed. In addition, the simulation results are shown and discussed.

A. Simulation setup

Simulation of the homogeneous environment of IoT is a challenge, as it comprises wireless things, wired things, and other passive things such as every day tools and devices. Therefore, in this paper, an efficient media-aware simulation framework is proposed to facilitate various multimedia streams to be transmitted over IoT environment. In the framework, diversity of multimedia traffic and things are considered. Moreover, in the designed simulation framework, a tradeoff between the IoT system flexibility, efficiency, and scalability are considered. In addition, increasing the number of nodes in the IoT system increases the interaction between these nodes, which leads to an increase in the size of multimedia transmission. Furthermore, an increase in the multimedia size should affect the services introduced by the IoT system resources. To create the simulation framework, the network simulation package NS2 is used to construct experiments that measure the performance of the proposed IoT-RTP and IoT-RTCP. NS2 is a widely used simulation tool for different networks such as WSN and RFID [47]. The simulation framework comprises three different networks, namely WSN, RFID, and mobile ad hoc networks (MAN). Internet is used as a communication medium between the three simulated networks. These types of networks are selected to provide flexibility, scalability, dynamism, and diversity that represent the main concept of an IoT environment. In addition, the behavior of these networks nodes changes periodically, which makes the simulation framework a mirror to the IoT system. Furthermore, these networks contain active nodes such as PCs, mobiles, routers, and sensors. Active nodes are found in WSN and MAN. In addition, passive nodes such as doors, chairs, and widows are found in RFID network. Moreover, these networks contain a large number of nodes that require the simulation framework to be extended rapidly, which is also most an important feature of an IoT system. The simulation starts with middle number of nodes, and the size of the IoT system is dynamically increased and decreased. To our knowledge, there are no previous researches or trials that aim to develop protocols for transmission of multimedia streams through IoT environments. So, the simulation results are compared with those from traditional RTP [48]. In this simulation, the methodology reported in [49] is used as a routing algorithm for multimedia flows. The topology of the IoT system is dynamically generated by NS2. In this simulation, the created nodes are uniformly distributed. The distance between nodes in one simulated network, in addition to the distance between each network and its neighbors, is determined randomly. The coverage area of the simulated networks is determined between the lower and upper limits by using a seed value that is changed dynamically during the simulation time period. The gaps between networks are changed depending on each network size. Hence, the intersections between networks are related to the gaps between networks. IoT systems should function without human intervention. To simulation this technical point, a table that contains a group of instructions is constructed and saved in each network server. These instructions are concerned with a group of events such that one or more instruction is mapped to one event. In the case of event occurrence, the instruction(s) will be executed automatically. The simulation events are stored in another table and also stored in the network server. The instructions and events are dynamically changed [50].

The simulation of WSN includes a number of sensors that are distributed randomly in a square area. For diversity, there are different transmission ranges of sensors. Each sensor has horizontal and vertical coordinates with values between zero and the maximum of its network coverage area. The sensors are arranged in a hierarchal view such that there is a sink for each group of sensors and there is a sink for each group of sinks. The simulation starts with three hierarchal levels. Each sink node is located at the center of its sensors. The energy consumption for each sensor is an important feature in WSN. Hence, our simulation framework should determine the energy level for each sensor and scale the consumption percentage depending on three factors, namely sensing, processing, and transmission of environmental data [51]. The simulation of RFID networks involves readers, tags, sites, and applications. In this simulation, sites and applications are represented by a strategy that is used to access IoT things with tags and readers. The management of simulated RFID is supposed to be central and performed remotely. In addition, the RFID network is extended in every simulation by a predetermined time to recover more geographical areas. The RFID function can be described briefly as follows: First, the RFID reader gathers information about its nodes and then sends it to the controller of network reader using one or more wireless access points. Then, these collected data are sent to the storage center that is related to a specific enterprise in the IoT system. Moreover, the reader controllers of RFID manage the whole network. Large data storage and node dynamic locations are considered to be the most important factor in RFID simulation. MAN networks are simulated in a square area. These network nodes are classified into two types: clients and servers. The sources of anycast requests are clients. Clients also act as intermediate nodes that transmit requests. The servers generate the replies through the unicast technique. The number of clients and servers is determined randomly and changed within the simulation time. Thus, one network server can receive and handle multiple anycast requests. The position of each node in the network is determined using uniform probability distribution. The mobility model reported in [52] is applied in the proposed simulation model to determine each node's speed and direction. As stated above, the transmission medium for the three simulated networks is the Internet. To complete the proposed simulation framework, Internet simulation should be performed. The Internet is represented as a collection of nodes. Each node has its own variables, structures and protocols. In addition, messages are transmitted between nodes using unicast, multicast, or broadcast techniques.

Simulated events in the Internet are created and synchronized using a timer. The creation, transmission, and processing of Internet messages are handled using built-in functions that are stored in one library in the simulation package. The proposed Internet simulation consists of three main properties, namely model decomposition or partitioning, efficient synchronization, and efficient process-to-processor mappings. These properties reduce communication overheads and achieve load balance between the nodes. The bandwidth of the Internet is a random value between 1 Mbps and 10 Mbps. The queuing system used in the Internet simulation is first in, first out (FIFO). The transport layer protocol is TCP and is changed to UDP in the case of network starvation and an increase in the transmitted multimedia streams over the predetermined threshold. The Internet simulation contains 100 routers and 20 servers with a random buffer size, between 50 kb and 2000 kb. General architecture of the proposed simulation model is shown in Fig. 7.

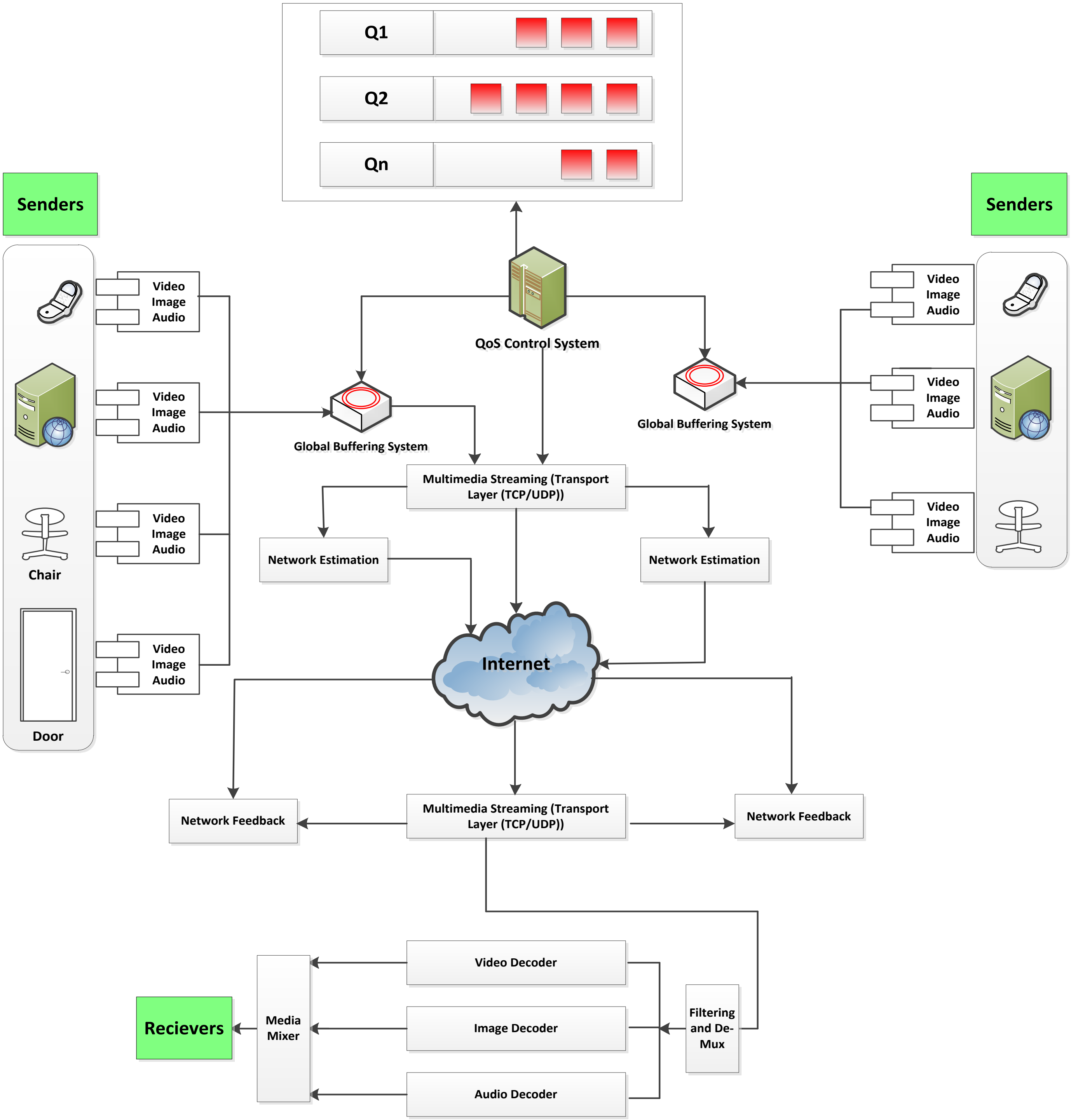

Fig. 7: General architecture of the proposed simulation model.

The simulation parameters for WSN are stated as follows; frequency equals 2400 MHz, transmission data rate equals 200 kb/s, RF power equals -10 dBm, receive sensitivity equals -94 dBm, current drain in transmission mode equals 11 mA, current drain in receiving mode equals 19.7 mA, battery 1250 mAH, number of sensors equals 1300, and coverage area equals 2000 m X 2000 m.

The simulation parameters for MAN are stated as follow; size of packet equals 1Mb, network area equals 2500 m x 2500 m, number of nodes equals 500, total number of requests 20000, interval between requests transmission equals 500 ms, TTL is random value between 4 ms and 7 ms, link availability is between 0 and 1, maximum transmission distance is from 30 to 210 m, maximum node speed is from 30 Km/h to 60 Km/h, and changing direction probability equals zero.

The simulation parameters for RFID are stated as follows; data channel frequency equals 915 mhz, control channel frequency equals 930 mhz, no inter-channel interference, no fading, SNR based signal reception equals 10, data rate equals 2 Mbps, radio RX sensitivity equals -91 dBm, RX threshold equals -81 dBm, transmission power of the RFID equals -45 dBm, reading range equals 1.62 m, sensing range equals 5.4 m, Interference range equals 7.1 m, and number of nodes equals 1000.

Within the IoT concept, this simulation contains two types of things, passive and active. The passive things are found in the RFID networks connected to the Internet using RFID tags. The active things may be found in one or more simulated networks, even in the RFID network. These active things send or receive multimedia streams internally through their networks or externally through the Internet. The multimedia streams that are sent or received in this simulation are video, audio, and images. MPEG-4 is the compression coding standard for video. For audio, PCM is used as a compression coding standard. The images are transmitted in JPG format. The traditional RTP and RTCP are compared to IoT-RTP and IoT-RTCP. The traditional RTP and RTCP simulation models are found in [45]. Packet loss, delay jitter, system throughput, and the number of multimedia packets sent and received are used as the performance metrics. The simulation is run for 1000 min. In addition, on average, ten simulation attempts are extracted to obtain accurate results.

For multimedia session properties in the simulation framework, the number of created sessions is dynamically changed within a range of 5 to 10. In the case of a small number of participants in a multimedia session, this session may be neglected and its participants will be distributed among other sessions depending on the distances between each participant and its near session. The number of participants in each is also dynamically changed within a range of 10 to 500. The division process occurred on two levels. At the start of the simulation, the participants of the original session are equally distributed among simple sessions. After that, for accuracy, the number of participants may be dynamically changed from the simple session to others.

B. Results and discussion

The performance metrics measured in the simulation experiments are end-to-end delay, delay jitter, number of RRs, packet loss, throughput, and energy consumption ratio. The results of these performance metrics are extracted from five simulation trails. Then, the average of the extracted results is calculated for each metric. Below, the results of the performance metrics are presented and discussed.

The end-to-end delay is one of the most important metrics to determine the effectiveness of IoT-RTP and IoT-RTCP for multimedia transmission through an IoT environment. The end-to-end delay is defined as the consumed time for transmission of multimedia packets from the source to the destination. Fig. 8 illustrates the results of a comparison between the proposed adaptive versions IoT-RTP and IoT-RTCP and traditional ones RTP and RTCP, in terms of their end-to-end delay. The x-axis represents the simulation time and the y-axis represents the average end-to-end delay. In the end-to-end delay measurements, the multimedia streams are generated from many hosts in the IoT environment and transmitted through intermediate nodes using wired and wireless channels. Owing to heterogeneous nodes, there are irregularities in the plots of end-to-end delay. In addition, the bandwidth may be limited, shared, and unpredictable in many parts of IoT system, which leads to instability of the end-to-end delay values. Nevertheless, most of end-to-end delay values for the proposed IoT protocols are lower than those for the traditional RTP and RTCP protocols. This is attributable to the awareness of each IoT system state and nodes. IoT-RTP and IoT-RTCP comprise fields that have information about the state of network, which enables multimedia streams to change their routing in the case a problem occurs. Further, in the proposed adaptive versions, the decrease or increase in the number of header bits is achieved dynamically depending on the network state. As seen in Fig. 8, there are high end-to-end delay values even for adaptive versions at simulation time points 4, 24, and 52 or for the traditional versions at simulation time points 5, 50, and 61. This is explained by the sudden increase in the transmitted multimedia streams resulting from the large number of multimedia session participants. This affects the transmission process and causes a notable increase in the end-to-end delay. Otherwise, the end-to-end delay for most (approximately 80%) of the simulation time points is normal for IoT-RTP and IoT-RTCP. Thus, the adaptive versions exhibit a better end-to-end delay than the traditional ones.

The delay jitter is also an important metric to test the performance of IoT-RTP and IoT-RTCP. The delay jitter is defined as the variations in delay. This metric is scaled from point to point over time. Wide variations in the transmission delay may affect the quality of video multimedia streams. There is a relationship between the delay jitter and the jitter buffer; more jitter buffer indicates a reduction in the delay's jitter effect on the network. Measurement of the delay jitter is important because of variations in the buffer size in the IoT system infrastructure. Fig. 9 shows the results of delay jitter for the adaptive versions and traditional ones. The x-axis represents the simulation time and y-axis represents the average delay jitter. IoT-RTP/RTCP has less delay jitter compared to traditional RTP/RTCP. This is explained by existence of passive nodes in the multimedia stream's routing path, which require recalculations during the transmission process. Moreover, the active nodes that have insufficient buffer are flagged, and in most cases, they are neglected in the case they have sufficient buffer. The results of the delay jitter include two averages. The first average is calculated every 10 simulation min. The second average is calculated every 5 simulation trails. There are notable irregularities in the curve plots, which can be explained by the sudden increase or decrease in multimedia transmission. Furthermore, construction and redefinition of multimedia session clusters may affect the transmission process. For IoT-RTP and IoT-RTCP, the average delay jitter values are in range of 0.005 to 0.024, which are acceptable values for transmission of multimedia streams through an IoT environment. For the traditional versions of RTP and RTCP, the average delay jitter values are in the range of 0.011 to 0.039.

To determine the effect of the newly added fields on the multimedia transmission through the IoT environment, the number of RRs within a time period should be measured. The number of RRs describes the overhead that may be added from IoT-RTP and IoT-RTCP. Fig. 10 shows the number of RRs for adaptive versions and traditional versions. The x-axis represents the simulation time and the y-axis represents the number of RRs. The results prove that the number of RRs for IoT-RTP and IoT-RTCP is lower than that for the traditional RTP and RTCP. This is attributable to the IoT-RTP/RTCP flexibility. The adaptive versions decreased their reports dynamically in the case of bandwidth starvation to permit for the multimedia streams to be transmitted without problems. The interruptions in the traditional version plot result from unpredictable events that may occur within the IoT multimedia session, such as bottlenecks, passive things, and nodes diversity. For the adaptive versions, the number of RRs is in the range of 88 to 190, whereas for traditional versions, the number of RRs is in the range of 88 to 280. In addition, the adaptive version plot is almost stable.

The packet loss metric is used to test whether the additional fields of the adapted versions represent an overhead in the IoT system (i.e., higher packet loss ratio means higher IoT-RTP/RTCP overhead and vice versa). The packet loss ratio is calculated by dividing the number of packets that reach destinations by the number of sent packets within a time period. Fig. 11 illustrates the packet loss ratio for the adaptive versions and traditional ones. The x-axis represents the simulation time, and the y-axis represents the average packet loss. It should be noted that the packet loss ratio of IoT-RTP/RTCP is less than that of traditional RTP/RTCP. This is attributable to the awareness of the IoT events as well as the reduced overhead of RTCP reports. The infrastructure of traditional RTP/RTCP suffers from slow discovery of sudden events that may occur dynamically and periodically in the IoT system. Further, traditional RTP/RTCP consumes high bandwidth consumption for control messages, which affects the multimedia transmission, and the packet loss ratio for traditional RTP/RTCP is high, especially at simulation points 9, 29, 39, 68, 80, and 98.

The throughput is defined as the number of bits that are transmitted through network and reach the destinations correctly within a time period. Fig. 12 shows the throughput for the adaptive and traditional versions. The x-axis represents the simulation time, and the y-axis represents the throughput values. The throughput of IoT-RTP/RTCP is higher than that of traditional RTP/RTCP. This is attributable to the readiness of IoT-RTP/RTCP to deal with a sudden increase in nodes in the IoT multimedia session. The IoT-RTP/RTCP divides the multimedia session into small sessions with normal distribution of nodes. In contrast, the traditional RTP/RTCP does not consider a sudden increase in multimedia session participants, which increases the transmitted multimedia streams and results in network starvation and affects the throughput value. The average throughput of IoT-RTP/RTCP reaches 200 Mb and that of traditional RTP/RTCP reaches 123 Mb.

The energy metric is considered one of the most important metrics in the IoT environment. IoT systems comprise high number of nodes, which are based on energy in its work. Hence, this metric of the proposed IoT-RTP/RTCP should be tested to ensure that the upgrades to the traditional versions do not affect the energy consumption rate for the IoT nodes. Fig. 13 illustrates the energy consumption for the nodes in the WSN, MAN, and RFID networks under IoT-RTP/RTCP and traditional RTP/RTCP. For all network types, the results prove the superiority of the proposed adaptive versions over the traditional ones. A high number of transmitted and processed packets indicates high energy consumption ratio. In the traditional RTP/RTCP, a large number of RRs may be sent within a small time period, regardless the network status. Hence, the traditional RTP/RTCP nodes are affected by transmission or processing of these RRs.

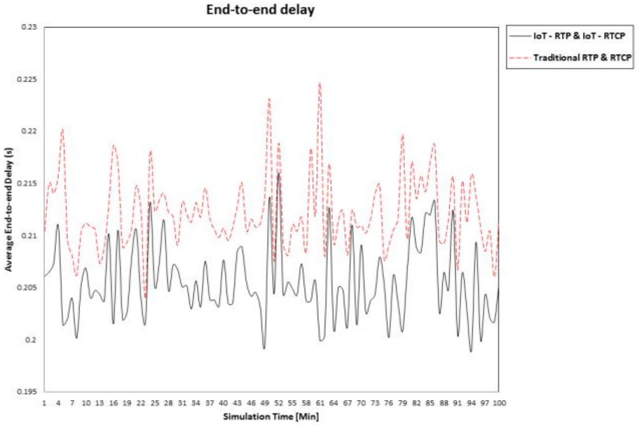

Fig. 8: End-to-end delay for adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

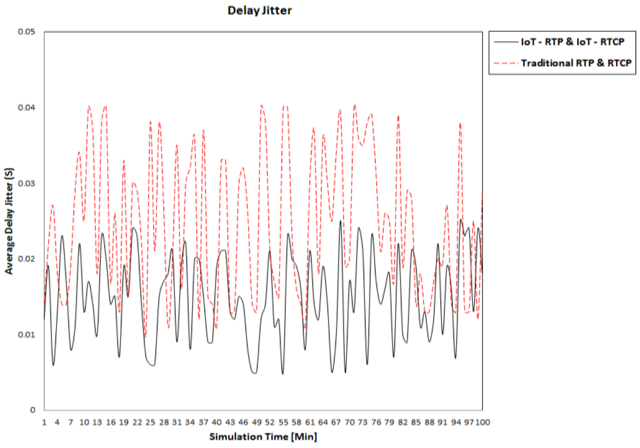

Fig. 9: Delay jitter for adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2726902, IEEE Access

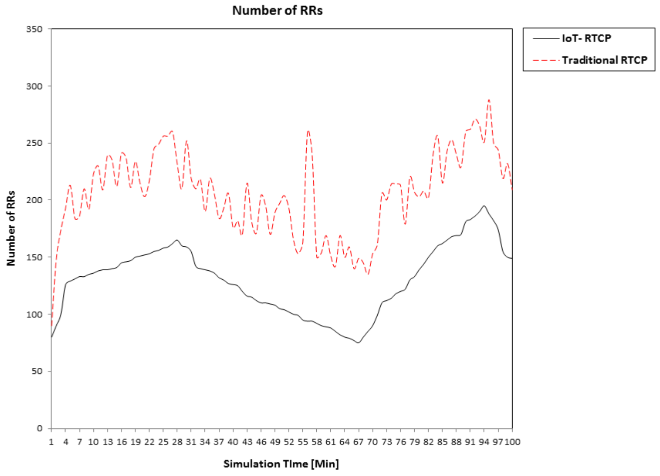

Fig. 10: Number of RRs for adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

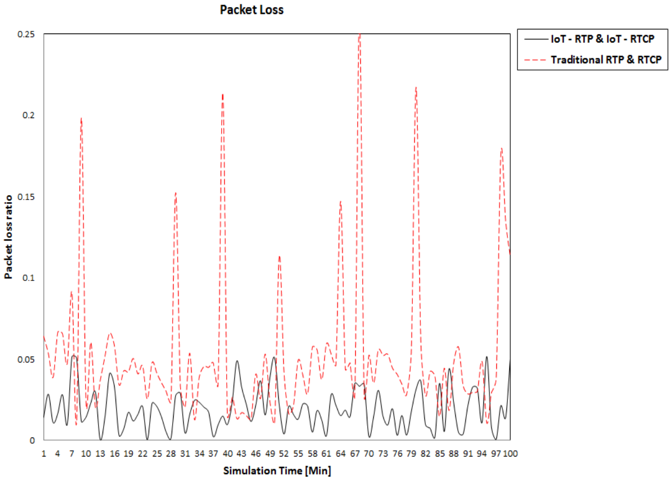

Fig. 11: Packet loss for adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

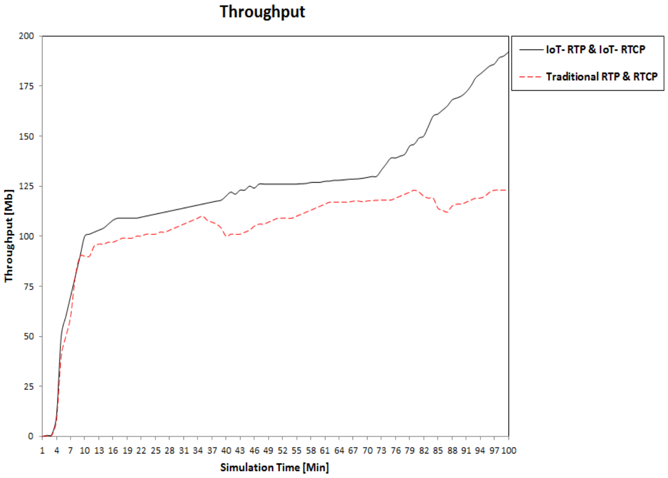

Fig. 12: Throughput of adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

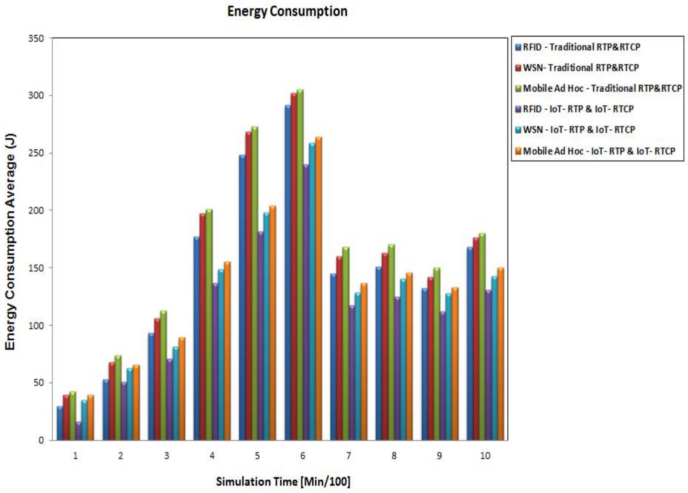

Fig. 13: Energy consumption for adaptive versions, IoT-RTP and IoT-RTCP, and traditional versions, RTP and RTCP.

VIII. CONCLUSION

In this paper, adaptive versions of RTP and RTCP are proposed. These versions transmit multimedia streams through an IoT environment. These versions are called IoT-RTP and IoT-RTCP. The basic concept of these versions is the division of a large multimedia session into smaller ones. Moreover, these versions consider special properties of IoT environment, such as coding, diversity, channels, and bottlenecks. New fields are added in the header of the adapted versions to determine the status of network periodically. NS2 is used to construct a simulation IoT environment for testing the proposed adapted versions. The measured performance metrics are end-to-end delay, delay jitter, packet loss, number of RRs, throughput, and energy consumption. The results prove that the proposed IoT-RTP/RTCP outperforms the traditional versions of RTP and RTCP. The percentages of improvement in the end-to-end delay, delay jitter, number of RRs, packet loss, and throughput are 2.05%, 39.14%, 36.61%, 37.42%, and 16.51%, respectively. The percentages of improvement in energy consumption for RFID, WSN, and MAN networks are 20.72%, 18.35%, and 17.54%, respectively. The use of IoT-RTP/RTCP to transmit multimedia streams thorough IoT environments is therefore recommended.

IX. FUTURE WORK

To create a long-term solution for the problems of multimedia transmission over IoT environments, a new transport layer protocol should be designed. Further, an adaptive version of QoS protocols such as RSVP should be proposed. Hence, the IoT-RTP/RTCP should be tested under the new adaptive version of the transport layer protocol and with a QoS protocol.

REFERENCES

[1] Yunchuan, S.; Song, H., and Jara, A. (2016). Internet of Things and Big Data Analytics for Smart and Connected Communities, IEEE Access. 4: 766 - 773.

[2] Chen, Z.; Wang, H. and Liu, Y., Bu, F., and Wei, Z. (2012). A Context-Aware Routing Protocol on Internet of Things Based on Sea Computing Model, Journal of Computer. 7(1): 96-105.

[3] Al-Fuqaha, A.; Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Communication Surveys & Tutorials. 17(4): 2347 - 2376.

[4] Chen, S.; Xu, H., Liu, D., Hu, B., and Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. IEEE Journal of Internet Things, 1(4): 349 - 359.

[5] Diana, M.; et al. (2016). Towards industrial internet of things: Crankshaft monitoring, traceability and tracking using RFID. Elsevier Journal of Robotics and Computer-Integrated Manufacturing. 41: 66 - 77.

[6] Stojkoska, B.,Trivodaliev, K. (2017). A review of Internet of Things for smart home: Challenges and solutions. Elsiver Journal of Cleaner Production. 140(3) 1454 – 1464.

[7] Laplante, P.; et al. (2016). The Internet of Things in Healthcare: Potential Applications and Challenges. IT Professional, 18(3): 2 – 4.

[8] Ejaz, W., Naeem, M., Shahid, A. (2017). Efficient Energy Management for the Internet of Things in Smart Cities. IEEE Communications Magazine. 55(1) 84 – 91.

[9] Mainetti, L.; et al. (2015). A Software Architecture Enabling the Web of Things, IEEE Journal of Internet of Things. 2(6): 445 - 454.

[10] Wang, J.; et al. (2016). A distributed algorithm for inter-layer network coding-based multimedia multicast in Internet of Things. Elsevier Computers & Electrical Engineering. 52: 125–137.

[11] Yang, Q. (2005). Improved performance using shortcut path routing within data vortex switch network. Electronics Letter, 41: 1253 - 1254.

[12] Delgalvis, I.; and Davison, G. (1964). Storage requirements for a data exchange, IBM System Journal. 3: 2-13.

[13] Francesco, D.; et al. (2012). A Storage Infrastructure for Heterogeneous and Multimedia Data in the Internet of Things. IEEE International Conference on Green Computing and Communications (GreenCom), Besancon, France 20-23 Nov. 2012. PP 26 – 33,

[14] Zhang, H.; and Dong, Y., (2007). The study on the exchange of metadata based on CWM and TUXEDO. Modern Electronics Technique. 6: 147 – 150.

[15] Chilingaryan, S.; et.al. (2011). A GPU-based Architecture for Real-Time Data Assessment at Synchrotron Experiments. IEEE Transactions on Nuclear Science. 58: 1447-1455.

[16] Guan, H.; et.al. (2013). Cache isolation for virtualization of mixed general-purpose and real-time systems. Journal of Systems Architecture - Embedded Systems Design. 59 (10): 1405 – 1413.

[17] Zhou, L., BaoyuZheng, M., and Jingwu, C. (2012). Green multimedia communications over Internet of Things. IEEE International Conference on Communications (ICC), Ottawa, Canada, 2012. PP 1948 – 1952.

[18] Sicari, S.; et al. (2016). A secure and quality-aware prototypical architecture for the Internet of Things. Elsevier Journal of Information Systems. 58: 43–55.

[19] Suryadevara, J., Sunil, B., and Kumar, N. (2013). Secured multimedia authentication system for wireless sensor network data related to Internet of Things. Seventh International Conference on Sensing Technology (ICST), Wellington, New Zealand, 2013. PP 109-115.

[20] Danilo, S.; Hyggo, A., and Angelo, P. (2015). A personal connected health system for the Internet of Things based on the Constrained Application Protocol. Elsevier Computers & Electrical Engineering. 44: 122–136.

[21] Eleonora, B. (2014). The Internet of Things vision: Key features, applications and open issues. Elsevier Computer Communications. 54 (1): 1 - 31.

[22] Ibrahim, M.; et al. (2015). Choices for interaction with things on Internet and underlying issues. Elsevier Ad Hoc Networks. 28: 68-90.

[23] Aijaz, A.; and Aghvami, H. (2015). Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective. IEEE Internet of Things Journal. 2(2): 103 – 112.

[24] J. Pan, J.; et al. (2015). An Internet of Things Framework for Smart Energy in Buildings: designs, Prototype, and Experiments. IEEE Internet of Things Journal, 2(6): 527 – 537.

[25] Vasileios, K.; Periklis, C., Francisco, V., and Jesus, A. (2015). A Survey on Application Layer Protocols for the Internet of Things. Transaction on IoT and Cloud Computing. 3(1): 1-10.

[26] Jiang, W.; and Meng, L. (2012). Design of Real Time Multimedia Platform and Protocol to the Internet of Things. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012. PP 1805–1810.

[27] Jiang, W.; and Meng, L. (2013). IOT Real Time Multimedia Transmission over CoUDP, International Journal of Digital Content Technology and its Applications (JDCTA) 7(6), 19-28.

[28] Jianxin, J.; and BaoyuZheng, C. (2013). Energy-aware distributed scheduling for multimedia streaming over Internet of Things. Int. J. of Ad Hoc and Ubiquitous Computing, 13(4): 176 - 186.

[29] Chen, J., et al. (2013). Application-dependent frame design for the Internet of Things. IEEE International Conference on Communications (ICC), Budapest, Hungary, 9-13 June 2013. PP 3694 - 3698,

[30] Sheeraz, A.; et al., (2015) Internet of multimedia things: Vision and challenges. Elsevier Ad Hoc Networks. 33: 87–111.

[31] Pereira, R., Pereira, E. (2015). Video Streaming: H.264 and the Internet of Things. IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Gwangiu, South Korea, 24-27 March 2015. PP 711 - 714.

[32] Atiquzzaman, M.; and Hassan, (2001). M. Adaptive Real-Time Multimedia Transmission over Packet Switching Networks. Real-Time Imaging, 7(1): 219-220.

[33] Wang, Y.; Reibman, A., and Lin, S. (2005). Multiple description coding for video delivery. Proceedings of the IEEE. 93(1): 57-70.

[34] Durresi, A.; Paruchuri, V., Durresi, M., and Barolli, L. (2008). Adaptive Layered Multimedia Transmissions over Wireless Networks. International Conference on Distributed Computing Systems Workshops, Beijing, China, 17-20 June 2008. PP 48 - 53.

[35] He, T., Stankovic, J., Lu, C., and Abdelzaher, T. (2003). SPEED: a stateless protocol for real-time communication in sensor networks. 23rd International Conference on Distributed Computing Systems, Providence, RI, USA, 19-22 May 2003. PP 46- 55.

[36] Mangharam, R.; Rowe, A., and Rajkumar, R. (2007). FireFly: a cross-layer platform for real-time embedded wireless networks. Real-Time Systems, 37(3): 183 - 231.

[37] Lotfallah, O., and Panchanuthan, S. (2003). Adaptive Multiple Description Coding for Internet Video. IEEE International Conference on Acoustics, Speech, and Signal Processing, 6-10 April 2003. PP 732-735.

[38] Mao, S.; Bushmitch, D., Narayanan, S., and Panwar, S. (2006). MRTP: a multiflow real-time transport protocol for ad hoc networks. Transactions on Multimedia. 8(2): 356 - 369.

[39] Zakaria, A., and El-Marakby, R. (2009). AdamRTP: Adaptive Multi-flows Real-time Multimedia delivery over WSNs. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, UAE, 14-17 Dec. 2009. PP 440 – 445.

[40] Iera, A., Molinaro, A., Ruggeri, G., and Tripodi, D. (2005). Dynamic prioritization of multimedia flows for improving QoS and throughput in IEEE 802.11e WLANs, IEEE International Conference on Communications (ICC), Seoul, Korea,16-20 May 2005. PP 1184 – 1189.

[41] Ott, J.; et al. (2006). Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF). RFC 4585.

[42] Baek, J.; and Salem, W. (2011). Dynamic cluster header selection and conditional re-clustering for wireless sensor networks, IEEE Transactions on Consumer Electronics. 56(4) 2249 - 2257.

[43] Hong, S., et al. (2011). A New Data Filtering Scheme Based on Statistical Data Analysis for Monitoring Systems in Wireless Sensor Networks, IEEE 13th International Conference on High Performance Computing and Communications (HPCC), Banff, AB. 2-4 Sept. 2011. PP 635 - 640

[44] Tsitsipis, D. (2012). Segmentation and reassembly data merge (SaRDaM) technique for Wireless Sensor Networks, International

Conference on Industrial Technology (ICIT), Athens, Greece. 19-21 March 2012. PP 1014 - 1019.

[45] MacQueen, B.; (1967). Some methods for classification and analysis of multivariate observations, in Proceedings of the 5th Berkeley Symposium on Math, Statistics and Probability, (eds J. Neyman and L. Le Cam) volume 1, Berkely, 1967. University of California Press, 281–297.

[46] Arendt W.; Schleich, W., (2009). Mathematical Analysis of Evolution, Information, and Complexity. WILEY-VCH Verlag GmbH & Co.KGaA,Weinheim, New York, USA.

[47] The Network Simulator – ns-2. 2008. [Accessed 30 Aug. 2016] (http://www.isi.edu/nsnam/ns/).

[48] El-Ramly, N.; et al. (2005). Analysis, Design, and Performance Evaluation of MS-RTCP: More Scalable Scheme for the Real-Time Control Protocol. J. UCS, 11(6): 874-897.

[49] Alvi, A., Shah, A., and Mahmood, W. (2015). Energy efficient green routing protocol for Internet of Multimedia Things, IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 7-9 April 2015. PP 1 – 6.

[50] Said O.; (2016). Analysis, design and simulation of Internet of things routing algorithm based on ant colony optimization. Wiley Int. J. Commun. Syst., DOI:10.1002/dac.3174.

[51] Zhou, Z.; et al. (2015). E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things. IEEE Sensors Journal, 16(11): 4072 - 4082.

[52] Jambli, M., (2012). Simulation Tools for Mobile Ad-hoc Sensor Networks: A State-of-the-Art Survey, International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, 26-28 Nov. 2012. PP 1-6.