


# Design and performance evaluation of mixed multicast architecture for internet of things environment

Omar Said<sup>1,3</sup> · Amr Tolba<sup>2,3</sup> 

© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** Internet of things (IoT) has become one of the most important fields in computing arena. The environments of IoT require highly efficient, immediate and worldwide communication services. Accordingly, efficient multicast routing architecture is a fundamental premise for IoT. This paper proposes a mixed multicast architecture for IoT environments that employs the centric, hierarchical, and distributed traditional multicast architectures. The aim is to determine the most suitable traditional multicast architecture, relative to the current state of the IoT system. First, an algorithm to manage the proposed multicast architecture is introduced. Then, an IoT case study for each traditional multicast architecture is demonstrated. Finally, a simulation environment is established, using the network simulator package NS2, to measure the performance of the proposed architecture. The considered performance metrics are end-to-end delay, packet loss, throughput, energy consumption, and transformation rate between traditional multicast architectures. The results demonstrate the superiority of the proposed architecture relative to individual traditional multicast architectures.

**Keywords** Multicast architectures · Internet of things · Simulation of internet of things · Multicast tree

---

✉ Amr Tolba  
atolba@ksu.edu.sa

<sup>1</sup> College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

<sup>2</sup> Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

<sup>3</sup> Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin El-Kom 32511, Egypt

## 1 Introduction

Unicast, multicast, and broadcast are three basic methods used to transmit data over a network. Implementation of unicast and broadcast is considered a straightforward process because packets are sent to a unique receiver, or will be propagated to all network nodes. A multicast implementation is considered more complex because users must be distributed into groups and their locations re-identified. For security, traffic must also be managed and controlled to prevent distribution to unnecessary destinations. This also saves bandwidth for other data transmissions [1–4]. Multicast traffic on a network should be managed by the Internet Service Provider (ISP). However, the volume of traffic sent using the multicast method is increasing [5–9]. Websites using video and audio depend on multicast technology, which leads to a requirement to deliver a huge number of identical packets to a high number of users simultaneously. The replication of these packets also occurs at an exponential rate. The required bandwidth and routing overhead for these types of websites may be extremely high [10–14].

The IoT is generally defined as the interconnection between objects, passive or active, to support different types of applications. Heterogeneous devices and applications are supplied by vendors globally, all of which should be accommodated. As a result, researchers try to adapt modern IoT methodologies, protocols, and technologies using the standard TCP/IP protocol developed for the traditional Internet [15–18]. IoT has different specifications to the traditional Internet however, and IoT systems comprise very large numbers of resource-constrained nodes. IoT environments may, therefore, have limited power nodes that are required to work for long periods of time. By its nature, IoT also creates many challenges with security, addressing, and routing. The construction of multicast architectures is one of the most important challenges in an IoT environment [19–23]. Power limitations also make multicasting in IoT networks expensive; this is because a single multicast will contain a sequence of multi-hop forwarding stages, which wakes up many nodes and consumes additional power. Heterogeneity of devices in IoT systems also means that the flooding of multicast packets may lead to unpredicted situations [24–28]. There is, therefore, a requirement for a new multicast architecture to be designed, which considers the special characteristics of IoT systems.

Use of IP multicasting by upper layer protocols to either send notifications to multicast group members, or to create a query for users in a group, is an intensive task. Delivery of packets using the multicast method is a considerable challenge for IoT environments [29] for four main reasons: First, disabling of the multicast ACK by MAC protocols in wireless networks means lost packets cannot be recovered at the link layer level. Second, heterogeneity of devices means that different MAC protocols exist, which leads to different rates of data transmission; senders must, therefore, transmit their multicast packets at the lowest speed of the group receivers. Third, most IoT nodes are energy based so may be switched on or off depending on battery status; this means that many multimedia packets may be lost due to inactive nodes. Finally, the multicast method requires many paths, leading to awakening of all network nodes in readiness for packet forwarding; this consumes additional energy and further utilizes scarce resources.

## 1.1 Contributions of this paper

1. We analyze centric, hierarchical, and fully distributed multicast architectures with regard to IoT environment.
2. We propose a mixed IoT multicast architecture using the three traditional multicast architectures.
3. We construct a simulation environment for IoT using NS2 to measure the performance of the proposed mixed IoT multicast architecture.
4. We compare the performance of the proposed mixed IoT multicast architecture with the three traditional multicast architectures; the simulation results show the effectiveness of our proposed mixed IoT multicast architecture in IoT environment.

The performance metrics, which used to test the proposed mixed multicast architecture, were end-to-end delay, packet loss, throughput, and average energy consumption; to ensure that the proposed multicast architecture was working smoothly, a usage percentage for each architecture, and a transformation rate was measured. The results showed that the mixed multicast architecture improves upon traditional multicast architectures with regard to the performance metrics.

The rest of this paper proceeds as follows: Related work is discussed in Sect. 2. Section 3 introduced the proposed mixed multicast architecture, while Sect. 4 demonstrates a case study for each traditional multicast architecture in an IoT environment. A mathematical analysis of the mixed architecture is introduced in Sect. 5. Simulation of the proposed multicast architectures is demonstrated in Sect. 6 and finally the conclusion is presented in Sect. 7.

## 2 Related work

A large volume of recent research exists in the field of multicast routing. The two classes of traditional multicast routing protocol are Shared Tree (ST) and Shortest Path Tree (SPT). The issues of wasting bandwidth and scalability are handled by ST-based protocols, under which three further protocols can be classified: Simple Multicast (SM) [30, 31], Core-Based Tree (CBT) [32], and Protocol-Independent Multicast Sparse Mode (PIM-SM) [33]. The main advantage of ST-based protocols is their handling of the bandwidth consumption problem. However, these protocols introduce further problems. The core routers elected by these protocols may have the same specifications as other routers, which lead to limited computing efficiency. In addition, the communication mechanism between the source and multicast group is less efficient, with high costs and delays. SPT-based protocols construct a multicast tree between each source and its group. These protocols are Multicast Extensions to Open Shortest Path First (MOSPF) [34, 35], and Distance-Vector Multicast Routing Protocol (DVMRP) [36]. The basic advantage of SPT-based protocols is minimization of the end-to-end delay. However, these protocols have three main problems: Network scalability, bandwidth consumption—which is used to adapt both MOSPF and DVMRP—and the fact that MOSPF or DVMRP generates multicast trees using the shortest path, which may not be the lowest cost [37].

Many multicast protocols for wireless networks have also been introduced. The most common protocols are Destination-Sequenced Distance Vector Routing (DSDV), On-demand Distance Vector (AODV), and Dynamic Source Routing (DSR). All are evaluated in [38, 39]. The Load-Based Energy Aware Multimedia Routing (LEAR) protocol is also a multicast protocol and is used to construct multicast trees for multimedia Wireless Sensor Networks (WSN) [40]. Moreover, many special-purpose protocols, techniques, and methods have been proposed for construction of multicast routing trees. Hachisuka et al. proposed an algorithm to construct multicast routing trees in networks based on a hierarchical optical path [41]. This algorithm identifies the relationship between multicast groups by determining adjacent destination groups for each node. Also, it introduces a waveband tree with minimum weight. A shared tree-mesh framework has been proposed by Wang et al. [42]. This framework comprises both mesh and tree topologies. The basic ideas of this framework are the definition of special nodes and the construction of a tree-based backbone. The backbone is called the Treebone and is used to transmit most of the multicast data. The defined nodes with other network are organized in mesh overlay. This makes the constructed Treebone dynamic and allows for efficient use of the available bandwidth.

Polishchuk et al. [43] introduced multicast architecture to apply the Adaptive Hybrid Error Correction (AHEC) scheme to large overlay networks. This multicast architecture is scalable and decreases required control information. Silva et al. [44] introduced the Entity Title Architecture (ETA) implementation, which corroborates the Open Flow method and used the future Internet in the multicast purpose. Entity Title Model (ETM) and Software Defined Networking (SDN) were considered in the design of ETA. The Multi-User Aggregated Resource Allocation (MARA) extension was introduced by Jardim et al. [45] and named Multi-User Aggregated Resource Allocation Multi Ingress (MARA-MI). This handles the QoS violation that may periodically occur in a network and allows processing of a multicast aggregated path. MARA-MI also provides an acceptable level of service quality in multi-user network sessions. Atlas et al. [46] demonstrated internal multicast protection techniques including fast re-route and also introduced external multicast protection techniques such as live-to-live.

The service centric multicast model was introduced by [37] and comprised a dominant router called the m-router. This router is used to gather multicast information and handles most multicast processes; other processes are handled by other routers in the network. The m-router is designed to handle many multicast communication events simultaneously. At each multicast group, a dynamic multicast tree is constructed in the m-router using the Service-Centric Multicast Protocol (SCMP). To save a bandwidth, a special type of packet is used to construct the multicast tree [37]. Nevertheless, this architecture is based on the idea of centralization, which has many problems including high hardware complexity, high cost of dynamic multicast tree construction, and low fault tolerance. Furthermore, the m-router requires high bandwidth, which may not be available.

There is a large amount of research closely related to this paper. Rahmani et al. [47] introduced a group of solutions which allow the establishment of multiple-services sensor networks from one sensor network. This provides resource sharing, in addition to creation of virtual links. The designed layers are application oriented, physical object representation, and context-aware. The solutions presented in this research together

comprise construction of a multicast tree in IoT. However, their main drawback is a weakness in the test environment. This considers a WSN environment to be the same as IoT environment which is inaccurate. Martynov [48] proposed a security model for multicast routing in IoT systems. But, the security aspect of this research is unrelated to multicast tree construction methodology. Akkermans et al. [49] proposed a mapping between subscriber groups in the application layer and multicast groups in the network layer. The multicast address can be acquired from each multicast group manager. In addition, the bandwidth and the energy consumption metrics are enhanced. The results of this research are inaccurate due to an implementation fault, which considered an IoT network to be the same as a WSN.

Antonini et al. [50] introduced an algorithm to support multicasting in Low-power and Lossy Networks (LLNs), which was also used to discover the services for smart objects in those networks. This algorithm was simulated using the Cooja package, which is a weak simulation and does not accurately reflect an IoT environment. Mahmud et al. [51] introduced an approach to study the communication models in WSNs, which cannot also be applied to IoT systems. It was shown that when this approach was applied to a large number of multicast groups, it provided better results than when applied to a small number of groups. Wang et al. [52] proposed an algorithm to optimize inter-layer flow and to satisfy the heterogeneous QoS requirements for transmitted data. This algorithm is considered a special-purpose solution because it studied the fixed flow rate problem for multimedia only. Moreover, the implementation environment was inaccurate due to lack of IoT representation. Huang et al. [53] deal with the multimedia data which require special QoS. Furthermore, the IoT environment may comprise many different types of data. So, the multimedia is considered as a special case. Shiuan et al. [54] proposed a lightweight protocol for IoT multicast routing. This protocol is based on the shortest path in selection of the multicast link. This is against the IoT environment nature which may comprises more long links but has efficiency more than shortest path ones.

### 3 Proposed IoT multicast architecture

The three traditional multicast architecture topologies are centric, hierarchical, and fully distributed [15, 37]. These architectures present several obstacles to their application in an IoT environment. The centric architecture means that all multicast functions must be achieved by a single powerful router; IoT environments contain billions of things (nodes) communicating in a single system, which generates heavy traffic and results in multicast functions that are too complex to be applied by one management router. In the hierarchical architecture, a powerful router can achieve most multicast functions, but two or more routers will assist the basic router with functions such as building the multicast tree; these routers can also replace the basic router if it fails. The hierarchical architecture is unsuitable for IoT environments due to very large multicast group sizes, which mean that management functions must be distributed on more hierarchical levels; the architecture also requires that the multicast functionality be distributed among the system routers. This architecture cannot be applied to an IoT environment due to their rapid extendibility, which may generate large volumes

of traffic and can cause the system routers to lose part of their multicast management function.

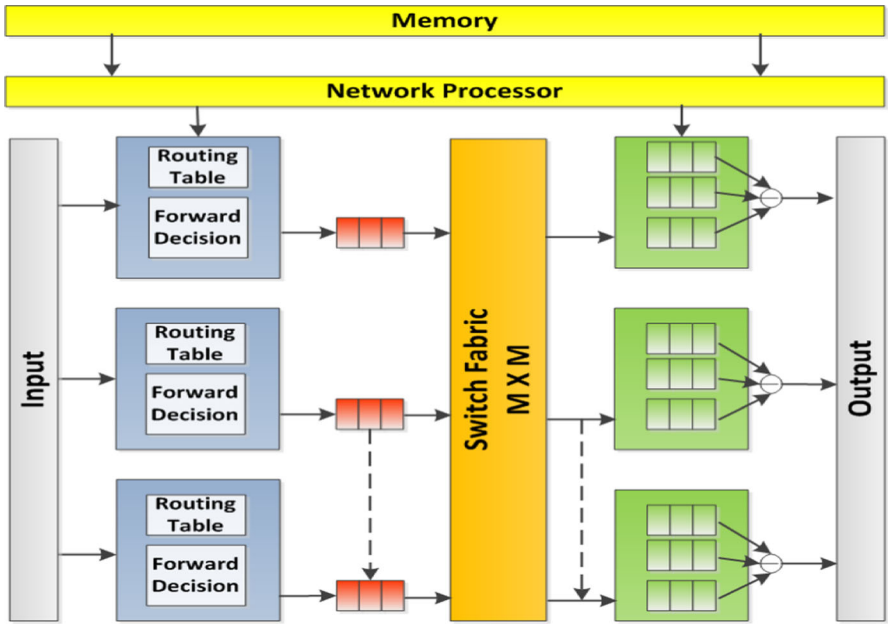
The proposed mixed architecture idea can be formulated based on the above short discussion and may be stated as follows: Each traditional multicast architecture may face problems when applied to IoT systems; therefore, the proposed architecture comprises a mixture of ideas from centric, hierarchical, and fully distributed architectures.

### 3.1 Components of the mixed IoT multicast architecture

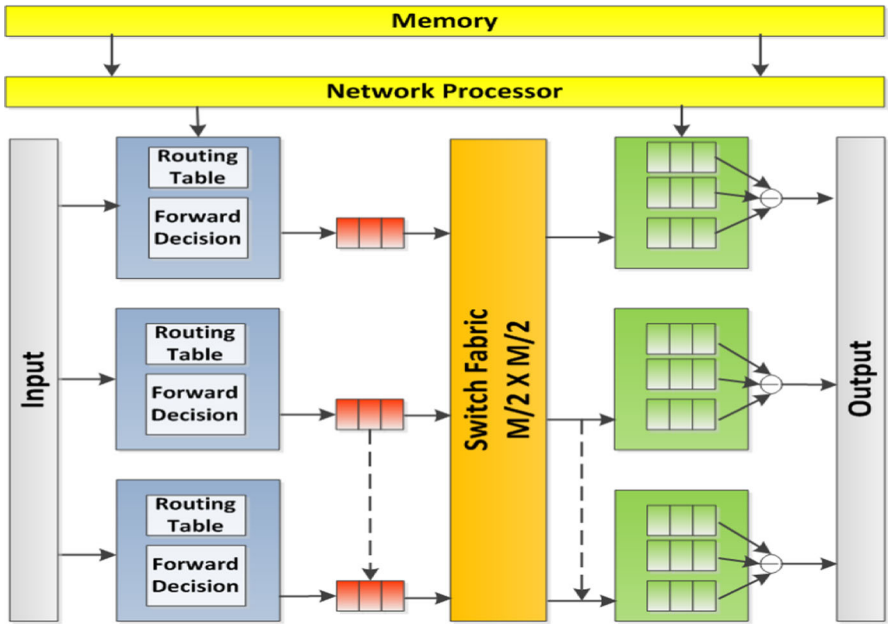
The proposed architecture comprises three types of router. The first are called Big-routers (B-router) and these can handle most multicast functions; they should also have the ability to avoid traffic jams and should forward large volumes of multicast data efficiently. B-routers are responsible for multicast group management, multicast tree generation, routing, transmission, and scheduling. These tasks may be run independently in parallel and the B-router should have efficient hardware support in the underlying switching fabric to allow them to achieve most multicast functions [37]. For each multicast group, many-to-many communication—or multicast connections from many sources—may be raised. The B-router hardware should support many-to-many communication by adopting the concepts of conference switching networks [37]. The second router type is called the Middle-router (M-router). These routers should have a level of hardware between traditional routers and B-routers, and may be used to recover B-routers; depending on the IoT system status, they may also fulfill a number of other multicast functions. The switch fabric size should be half that of the big routers, and the M-router network processor speed, database specifications, and buffer size should be less than that of the B-routers. The third router type is called a Traditional-router (T-router) and this can run simple multicast functions. Simple views of the B-router and the M-router are shown in Fig. 1a, b.

### 3.2 Mixed multicast architecture operation

A mixed multicast architecture includes ideas from the three traditional multicast architectures, such as Centric, hierarchical, and distributed. Each router in an IoT system should be able to accomplish multicast functions, which may be periodically required. The proposed architecture assumes that at the beginning of a multicast session, a centric multicast architecture will be used to manage the session. First, the B-routers are distributed in the IoT system, with the number of B-routers determined by the size of the IoT environment and the size of each multicast group; in mixed multicast architecture, it is assumed that a single B-router is used and is responsible for the multicast functions. The network state is periodically determined by network parameters such as delay, loss, bandwidth consumption, and energy. Passive things should also be determined because they have special specifications; for example, they can only receive data in the multicast group. If the B-routers experience traffic overload, the M-routers should then assist the B-routers with some multicast functions, providing that the multicast tree construction process remains unaffected. Because the B-router still has more multicast functions than the M-routers, at this stage the



(a)



(b)

Fig. 1 a Simple view of a B-router, b simple view of an M-router

Message Type	Length	Flag
M <sub>1</sub> -router Address		Reactivated Function Codes
M <sub>2</sub> -router Address		Reactivated Function Codes
-----		Reactivated Function Codes
M <sub>n</sub> -router Address		Reactivated Function Codes
T <sub>1</sub> -router Address		Reactivated Function Codes
T <sub>2</sub> -router Address		Reactivated Function Codes
-----		Reactivated Function Codes
T <sub>n</sub> -router Address		Reactivated Function Codes

**Fig. 2** Simple view of a transformation message

proposed multicast architecture will be transformed from centric to hierarchical. If further traffic overload occurs, the T-routers should then assist the M-routers, at which point the multicast architecture will be transformed from hierarchical to distributed; this transformation is achieved when M-routers cannot fulfill functions required to construct the multicast tree. Additional routers must be used to help the M-routers, so distributed architecture must be used. The new routers are selected by the T-routers and will reactivate the multicast functions; the T-router selection process depends on many factors, including the distance between the selected T-routers and multicast groups, the energy, and the routing path.

The multicast functions should be accurately identified. Those used in this paper are multicast tree generation, multicast group management, transmission, and scheduling. The execution methodology of the multicast functions among the routers should be achieved using the activate/deactivate/reactivate strategy; this strategy means that the multicast functions are already settled on each router system by the time of activation and deactivation should be determined depending on network status.

The transformation from one architecture to another should be completed using a special type of message, called a transform-message; this message should be sent to the M-routers informing them that their multicast functions should be reactivated and can be used to inform the T-routers that the M-routers share the B-router for the multicast tree construction. The structure of this message is found in Fig. 2 and consists of six fields: Message type, length, flag, M-router addresses, T-router addresses, and reactivated function codes. The message type field is 2 bits in size and is used to determine the type of message—informing either the M-router or the T-routers; the length field is 8 bits in size and is used to determine the total length of the transform message; the transformation from one multicast architecture to another is determined by the flag field, which is 2 bits in size; the size of the M-routers address field is 128 bits, and this determines the addresses of M-routers that require their multicast function to be reactivated; the size of the T-router address field is 128 bits, and this determines the T-routers that require their multicast functions to be reactivated; finally, the size of the reactivated function codes field is 8 bits and this determines the codes of the multicast functions that should be reactivated. The code of each multicast function is determined using a binary number; for example, tree management procedure 1 has code 0000, and the tree management procedure 2 has code 001. See Algorithm 1 for more clarification.



**Algorithm 1: Management algorithm for mixed multicast architecture****Inputs:**

D: Current IoT system delay  
 L: Current IoT system loss  
 O: Current IoT system load  
 $D_T$ : Threshold IoT system delay  
 $L_T$ : Threshold IoT system loss  
 $O_T$ : Threshold load IoT system load

**Outputs:**

Architecture Type  
 Member Join  
 Member Leave  
 Member Addition

**For** all IoT systems

**Begin**

**If**  $((D < D_T) \ \&\& \ (L < L_T) \ \&\& \ (O < O_T))$

{  
 B-router starts to construct the multicast tree using the centric architecture.  
 Deactivate all the multicast functions in M-router and T-routers.

**If** a join request is received

{  
 Call B-router\_F1  
 Call MA\_F  
 }

**Else If** a leave request is received

Call B-router\_F2

}

**Else If**  $((D == D_T) \ \&\& \ (L == L_T) \ \&\& \ (O == O_T))$

{  
 B-routers and M-routers start to construct the multicast tree using hierarchical architecture.  
 Reactivate all multicast functions in M-routers.  
 Use temporary buffering packet.

**If** a join request is received

{  
 Call B-M-router\_F1  
 Call MA\_F  
 }

**Else If** a leave request is received

Call B-M-router\_F2

}

**Else**

{  
 B-routers, M-routers, and T-routers start to construct the multicast tree using distributed architecture.  
 Reactivate all multicast functions in M-routers and T-routers.

**If** a join request is received

{  
 Call B-M-T-router\_F1  
 Call MA\_F  
 }

**Else If** a leave request is received

Call B-M-T-router\_F2

}

**End.**

<b>B-router F1: For Member Join</b>	<b>B-router-F2: For Member Leave</b>
<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Join multicast group.</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">N: Number of multicast groups  M: Number of router interfaces  <b>For</b> I = 1 to N  {      <b>If</b> GID exists      {          <b>For</b> J = 1 to M          <b>If</b> FID exists          Send join message to B-router.          <b>Else</b>          Create FID.      }      <b>Else</b>      Create multicast routing entry using (GID and FID)  }</p> <p><b>End</b></p>	<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Leave multicast group and Delete message</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">Delete FID from the multicast routing entry.  <b>If</b> no downstream      Send Remove message      Send Leave message  <b>Else if</b> all downstream are routers      Send leave message to B-router.</p> <p><b>End</b></p>

<b>B-M-router F1: For Member Join</b>	<b>B-M-router F2: For Member Leave</b>
<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Join multicast group.</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">N: Number of multicast groups  M: Number of router interfaces  <b>For</b> I = 1 to N  {      <b>If</b> GID exists      {          <b>For</b> J = 1 to M          <b>If</b> FID exists          Send join message to M-router.          <b>Else</b>          Create FID.      }      <b>Else</b>      Create multicast routing entry using (GID and FID) at M-router      M-router sends update message to B-router  }</p> <p><b>End</b></p>	<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Leave multicast group and Delete message.</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">Delete FID from the multicast routing entry at M-router.  <b>If</b> no downstream      Send remove message      Send Leave message  <b>Else if</b> all downstream are routers      Send leave message to M-router.      M-router sends leave message to B-router.</p> <p><b>End</b></p>

<b>B-M-T-router-F1: For Member Join</b>	<b>B-M-T-router-F2: For Member Leave</b>
<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Join multicast group.</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">N: Number of multicast groups  M: Number of router interfaces  <b>For</b> I = 1 to N  {      <b>If</b> GID exists      {          <b>For</b> J = 1 to M          <b>If</b> FID exists          Send join message to T-router (the most recent one).          <b>Else</b>          Create FID.      }      <b>Else</b>      Create multicast routing entry using (GID and FID) at T-router.      Update the multicast tree at T-router.  }</p> <p><b>End</b></p>	<p><b>Input:</b> Multicast group identification (GID) and Interface identification (FID)  <b>Output:</b> Leave multicast group and Delete message.</p> <p><b>Begin</b></p> <p style="padding-left: 20px;">Delete FID from the multicast routing entry at M-router.  <b>If</b> no downstream      Send remove message      Send Leave message  <b>Else if</b> all downstream are routers      Send leave message to T-router.      Upgrade the multicast tree at T-router.</p> <p><b>End</b></p>

**Member Addition (MA F)**

**Input:** R, x, and  $N_h$   
**Output:**  $N_{h+1}$   
 R: Multicast group.  
 x: Node which should join the multicast group.  
 $N_{h+1}$ : Multicast tree after joining the new node.  
 A: Group members that receive the data.  
 $M_c$ : Minimum cost.  
 $M_{ax}$ : Maximum cost.  
 $P_{LC}(i, x)$ : Least cost link between node i and node x.  
 $P_{SD}(i, x)$ : Shortest delay link between node i and node x.  
 $Ev(i, x)$ : Energy between node i and node x.  
 $\Delta$ : Minimum energy which required for data transmission.  
 D: Delay.  
 Tgp: Graft path which connects the best node to node x.  
 Tgn: Best graft node to node x.

```

Begin
  If  $x \in N_h^x$ 
     $A_{F+1} = A_i + \{x\};$ 
  Else
     $M_c = M_{ax}$ 
    For  $i = 1$  to  $t$ 
      If  $i \neq$  Passive
        If  $((D(P_{LC}(i, x)) + D(P(i)) < D_T) \&\&Ev(i, x) > \Delta)$ 
          If  $C(P_{LC}(i, x)) < M_c$ 
            {
               $M_c = C(P_{LC}(i, x))$ 
               $Tgp = P_{LC}(i, x)$ 
               $Tgn = i$ 
            }
          Else If  $((D(P_{SD}(i, x)) + D(P(i)) < D_T) \&\&Ev(i, x) > \Delta)$ 
            If  $C(P_{SD}(i, x)) < M_c$ 
              {
                 $M_c = C(P_{SD}(i, x))$ 
                 $Tgp = P_{SD}(i, x)$ 
                 $Tgn = i$ 
              }
            }
        }
    }
   $N_{h+1} = N_h \cup gp$ 
End
    
```

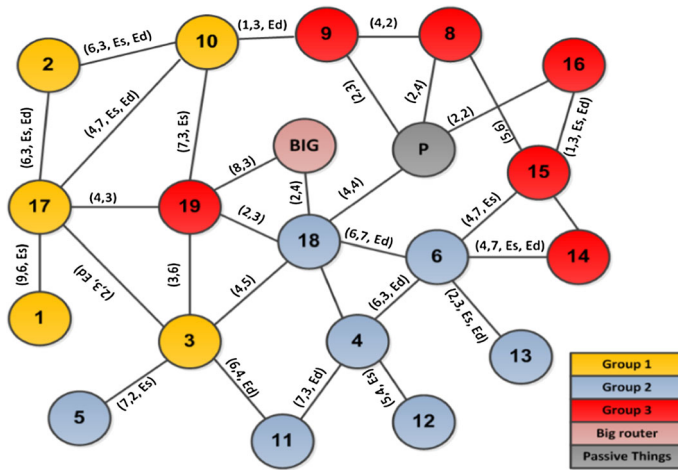
**Temporary buffer of multicast packets**

**Input:** Multicast packet  
**Output:** Unicast packet

**Begin**  
 A: Number of bottlenecks in the IoT system.  
 R<sub>p</sub>: Router which will buffer the multicast packets.  
 P<sub>i</sub>: Bottleneck i.  
 B<sub>c</sub>: Number of routers around R<sub>p</sub>.  
**For**  $I = 1$  to  $a$   
 {  
 Select the router before the bottleneck point (R<sub>p</sub>)  
**If** R<sub>p</sub> has a small buffer size  
**For**  $1$  to  $B_c$   
 {  
 Select the biggest router buffer  
 R<sub>p</sub> = Biggest router buffer  
 }  
 R<sub>p</sub> sends unicast packet for the P<sub>i</sub> node.  
**If** P<sub>i</sub> is deleted  
 Multicast packets will be processed  
 }  
**End**

**4 Case study**

The case study describes how the multicast tree will be constructed in the three multicast architectures using part of the IoT domain. The transformation from one multicast architecture to another is illustrated.



**Fig. 3** Part of an IoT topology

#### 4.1 Centric IoT multicast architecture

The proposed centric IoT multicast architecture has one management router, called the B-router. This router should have powerful specifications, to allow management of the whole multicast tree and is responsible for contact with the Internet Service Provider (ISP). The rest of the IoT system routers are also managed by the B-router. Distribution of these routers in an IoT system depends on the following factors: load balance, link cost, link delay, type of thing, and network state. Other routers should be added by logging into the ISP domain. To complete the login process, each new router should communicate with the B-router. Communication is achieved by sending a unicast message to the B-router; on receiving this message, the B-router upgrades the multicast tree.

Figure 3 shows a simple IoT topology, comprising three multicast groups. The nodes in the first group are 8, 9, 14, 15, 16, and 19. In the second group the nodes are 4, 5, 6, 11, 12, 13, and 18, and in the third group 1, 2, 3, 10, and 7. A passive object is represented in this topology by one node. Each link has minimum of two parameters, namely link cost and link delay. Because many nodes are energy based in an IoT environment, energy nodes should be represented for accuracy. The links should, therefore, identify not only cost and delay, but also source and/or destination energy. Many links contain source or destination energy because one of the two nodes is energy based. Figure 4 shows the routing loop deletion; the links with dotted lines represent those deleted and are determined using an existing algorithm [15, 37]. Figure 5 shows the output multicast tree if centric multicast architecture is used. It is notable that the B-router should control and manage multicast functions in the multicast architecture. The IP address of this B-router should be known to other routers or nodes in the system, which can be accomplished by inserting this into the configuration file of each router; each router can then join or leave the multicast group using a join or leave message [15] sent to the B-router. The B-router can then update the multicast tree in accordance with the current state of domain. Membership messages, and collection of information

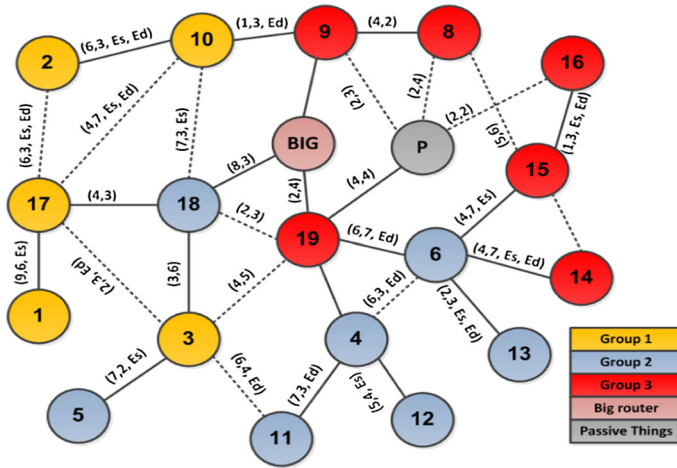


Fig. 4 Routing loop deletion

by the B-router, are two main factors used in multicast tree construction. After the multicast tree is constructed, each router should upgrade its routing table based on the new multicast tree. A special packet type is used to complete the information used to construct the multicast tree and generation of this type of packet is the responsibility of the B-router. The packet type can also be used by other routers in the system to update their routing table. The special packet is sent from one router to another in the lower levels recursively, so the packets will be transmitted in bidirectional mode. Energy is an important factor in multicast tree construction. If critical energy levels exist during the multicast tree construction, even for source or destination nodes, an alternative path should be found; alternatively, the load of information transmitted by a special packet may be decreased, by neglecting some of the packets containing less important information. Passive objects can receive packets but cannot transmit new packets; in the multicast tree, therefore, the nearest router can accomplish the functions required from the passive object.

### 4.2 Hierarchical IoT multicast architecture

In a hierarchical multicast architecture, multicast functions will be distributed among more than one router. In this case study, two M-routers, M1-router and M2-router, are used to help the B-router with the multicast tree construction process. The IoT topology will be divided into two parts, with the M1-router used to act as a B-router. The M1-router can therefore send and receive specific packets, update multicast information, construct the multicast tree, and determine the passive objects, so determining if the link is active or dead. The multicast tree construction uses unicast messages between the system routers and the M1-router, which are sent in bidirectional mode from higher level routers to lower level routers, and vice versa. The multicast tree for the M2-router can be constructed using the same methodology as the M1-router. After the two parts

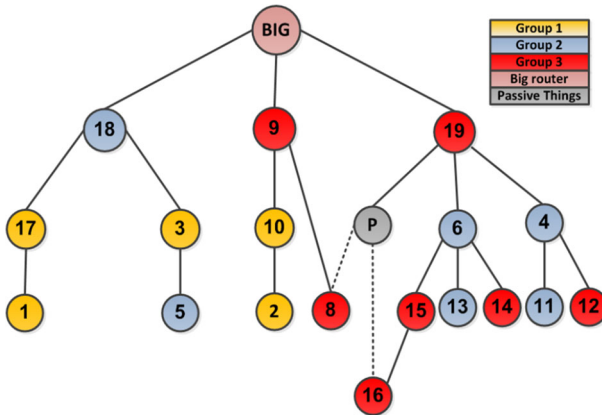


Fig. 5 Multicast tree construction using centric multicast architecture

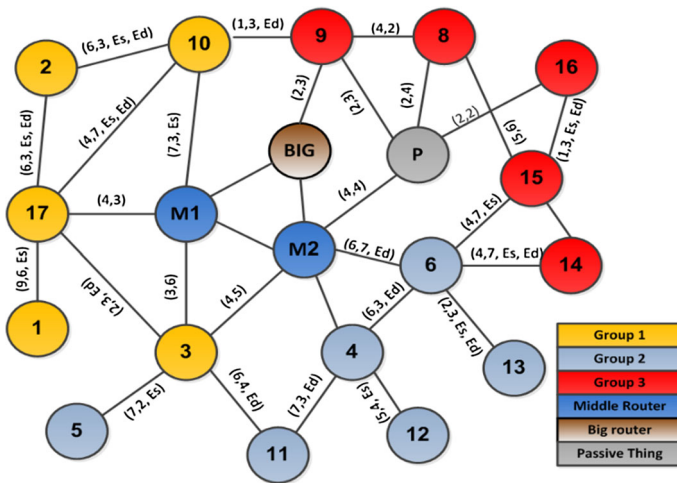


Fig. 6 Part of an IoT topology

of the multicast tree have been constructed, they should be merged using the algorithm described in [55].

Figure 6 shows the domain IoT topology which will be transformed into the multicast tree. As stated above, there are three multicast groups. The nodes in the first group are 8, 9, 14, 15, and 16, the nodes in the second group 4, 5, 6, 11, 12, and 13, and the nodes in the third group 1, 2, 3, 10, and 17. This IoT topology also includes one passive object as an example. The links in the IoT topology are defined using the link delay, link cost, energy of source and/or destination, and link state (active or dead). Figure 7 shows the routing loop deletion process using the algorithm described in [37], with the loops denoted by dotted lines. Figure 8 shows the multicast tree using the hierarchical multicast architecture. Fault tolerance, load balancing, and some multicast message structures are proposed in [15].

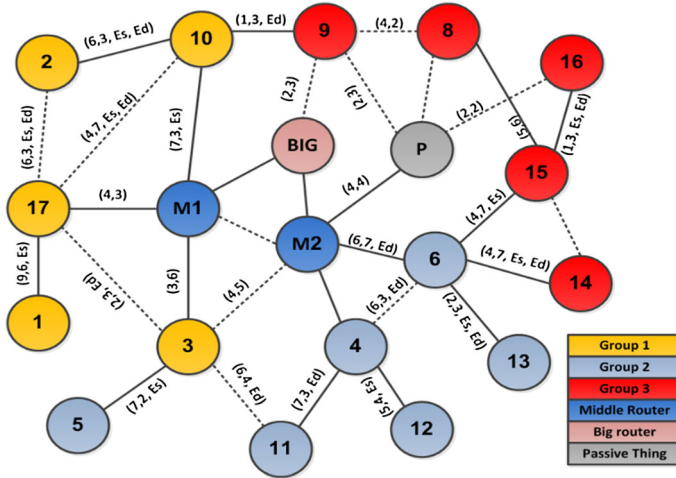


Fig. 7 Deletion process of a routing loop

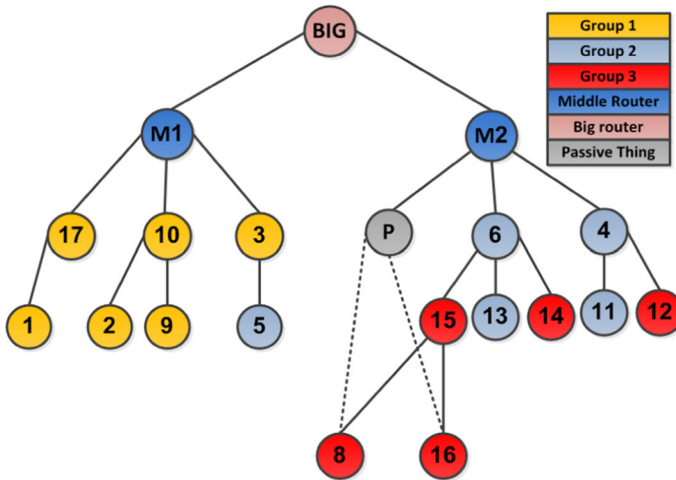
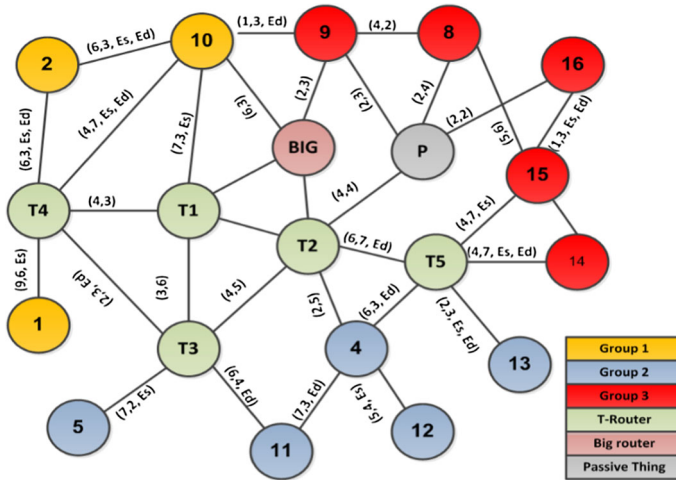


Fig. 8 A multicast tree using hierarchical multicast architecture

When building the multicast tree using the hierarchical multicast architecture, routers are organized in levels. The first level is for the B-router, the second level for the M-routers, and the third level is the T-routers. Each router can construct its multicast tree and send it to the M1-router or the M2-router, depending on which region the T-router belongs to. Many terminologies may be used with this architecture, including border router, global distance, local distance, and zero distance. A border router is a router which has a direct connection with an M1-router or M2-router; the global distance is defined as the distance between the T-router and the M-router; the local distance is the distance between two T-routers in the same area; lastly, the zero distance is the distance between any router and itself. When a new router needs to



**Fig. 9** Part of an IoT topology

connect to the multicast tree, it should send a unicast message to either the M1-router or M2-router, depending on its region.

### 4.3 Distributed IoT multicast architecture

In distributed multicast architecture, multicast functions are distributed among the B-router, M-routers, and T-routers, and each router in the domain can complete these functions. Each router can construct its multicast tree, which is then considered as a part of the global multicast tree. This will be sent to the higher level router, which then receives, collects, and merges these sub-trees into one large sub-tree and again sends it to higher level routers. The process continues until the global multicast tree has been constructed. When a new host needs to join a multicast group, it should send a unicast message to the higher level router; the message is transferred from one T-router to another, until it reaches the target router responsible for the target multicast group.

Figure 9 shows the IoT domain, which comprises three groups. The nodes in the first group are 8, 9, 14, 15, and 16, the nodes in the second group 4, 5, 11, 12, and 13, and the nodes in the third group 1, 2, and 10. To clarify the distributed IoT multicast architecture, the T-routers have been determined and marked as T1, T2, T3, and T4. The passive objects are represented with one node. The B-router has a connection to multiple hosts which means that it can construct part of multicast tree. As with the two previous architectures, the links in a distributed IoT multicast architecture have cost, delay, source energy, and/or destination energy properties. Figure 10 shows the deletion of routing loops, denoted by dotted lines. Figure 11 shows the multicast tree constructed using distributed IoT multicast architecture.



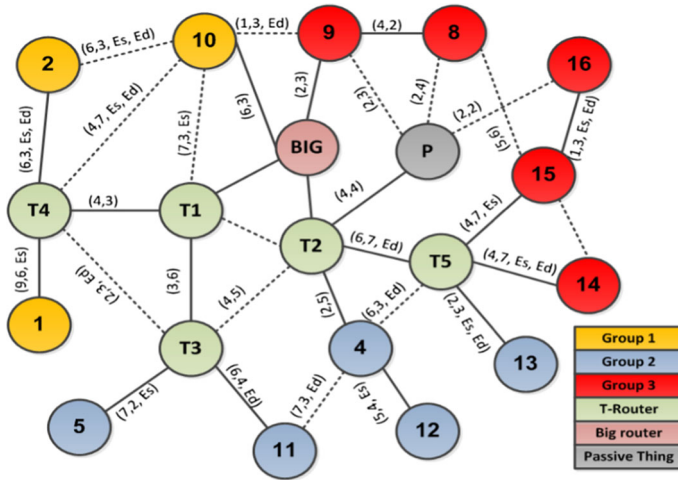


Fig. 10 Deletion process of a routing loop

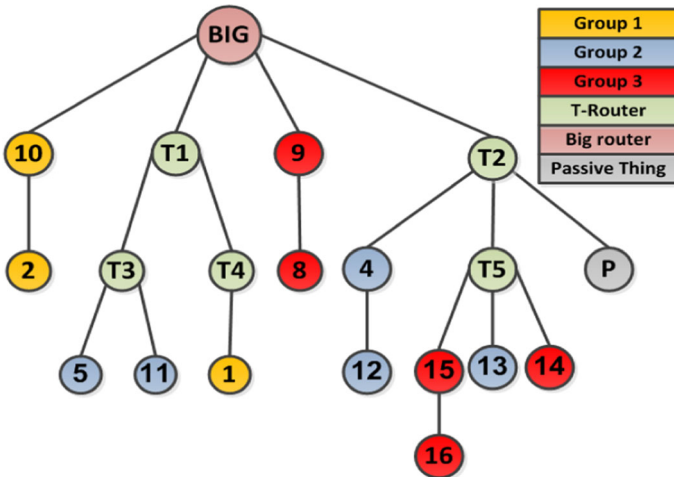


Fig. 11 A multicast tree using distributed multicast architecture

### 5 Mathematical analysis of the IoT multicast problem

Our proposed mixed IoT multicast architecture can be mathematically formalized based on [37] as follows: The undirected graph  $G(V, E)$  is used to represent the point-to-point communication, where  $V$  represents a set of nodes and  $E$  represents a set of edges. The nodes represent routers. Three positive functions are link cost, link delay, and energy cost. The cost of a link is determined by the utilization of that link; lower utilization means lower cost and vice versa, and the link cost function is denoted by  $LC : E \rightarrow \mathbb{R}^+$ . The delay of the link is calculated by a summation of buffering delay, transmission delay, and propagation delay; the link delay is denoted by  $LD : E \rightarrow \mathbb{R}^+$ .

The link energy is determined by the energy values at one or both of its ends. The link has high energy if the energy at its end(s) is greater than a predetermined value. This value varies depending on the type of the data that may be transmitted through this link and is denoted by  $LE : E \rightarrow \mathbb{R}^+$ . The link state is determined by the type of its ends; if one or two ends have passive objects, the state will be dead.

Suppose that a path  $T$  in the IoT system comprises a set of links  $S_i$ , where  $i = 1, 2, 3, \dots, n$ . The link cost, link delay, link energy, and link state are determined using the threshold of Eqs. (1)–(4):

$$\text{Cost}(T) = \sum_{i=1}^n \text{LC}(S_i) \quad (1)$$

$$\text{Delay}(T) = \sum_{i=1}^n \text{LD}(S_i) \quad (2)$$

$$\text{Energy}(T) = \sum_{i=1}^n E_i(S_i) \quad \forall E_i > E_{\text{threshold}} \quad (3)$$

$$\text{State}(T) = \sum_{i=1}^n \text{LS}(S_i) \quad (4)$$

The multicast session contains many groups. The node that generates the data is  $N \in V$  and the set of group members is denoted by  $A \subseteq V$ . The group members can receive the data generated by  $N$  that may or may not be found in set  $M$ . The multicast tree is denoted by  $L$ ;  $L \subset G$  where  $G$  is a global graph.  $L^v$  and  $L^e$  represent the set of nodes and the set of edges, respectively. The tree cost, tree delay, tree energy, and tree state equations can thus be re-formalized using tree edges, as seen in Eqs. (5)–(8). Nodes used to transmit the data generated by  $N$  that do not belong to the multicast group are called relay nodes.

$$\text{Cost}(T) = \sum_{e \in L^e} \text{LC}(e) \quad (5)$$

$$\text{Delay}(T) = \sum_{e \in L^e} \text{LD}(e) \quad (6)$$

$$\text{Energy}(T) = \sum_{e \in L^e} E_i(S_i) \quad \forall E_i > E_{\text{threshold}} \quad (7)$$

$$\text{State}(T) = \sum_{e \in L^e} \text{LS}(e) \quad (8)$$

If  $Y$  is a node in  $L^v$ , there is at least one path connecting  $N$  to  $Y$  which satisfies the required cost, delay, and energy requirements; the state of this path is active (not dead) and the path is denoted by  $P^L(Y)$ . The two types of multicast requests are added or removed. The addition request is denoted by a pair  $(v_i, p_i)$ , where  $v_i \in V$ ,  $p_i \in \{add\}$ ; the removal request is denoted by a pair  $(v_i, p_i)$ , where  $v_i \in V$ ,  $p_i \in \{remove\}$ . The sequence of requests is denoted by  $S$ , where  $S = \{v_1, v_2, v_3, v_4, \dots, v_n\}$ . There are

also many sub-trees for which the number increases or decreases periodically, given by  $L^n = \{l_1, l_2, l_3, l_4, \dots, l_n\}$ . The problem of a mixed IoT multicast tree can thus be formalized as follows: Suppose that we have a global graph  $G$ , a source node  $N$ , link cost function (LC), link delay function (LD), link energy function (LE), link state function (LS), and  $\Delta$ , which represents a threshold value for each transmission requirement. A group of multicast trees must be constructed such that  $\text{Delay} (P^L (Y)) \langle \Delta, \text{Cost} (P^L (Y)) \langle \Delta, \text{and Energy} (P^L (Y)) \rangle \Delta \forall N \in l_i$ . The link state should also be considered.

The host can join the multicast tree by sending a multicast request  $S_i$ . The initial multicast tree starts with  $N$  as a source node:  $L_0^v = \{N\}$ ,  $L_0^e = \emptyset$ . Let us suppose that the construction process of multicast tree,  $L_F$ , is completed after processing  $F$  multicast requests, where  $L_F^v$  is a set of nodes and  $L_F^e$  is the set of edges. After joining of node  $j$  to the multicast tree  $L_F$ , it becomes  $L_{F+1}$ . There are two possibilities when node  $V$  joins the multicast tree: The first is  $j \in L_F^v$ , which means that node  $j$  is already in the multicast tree, so no changes are made:  $L_F = L_{F+1}$ . The second is where  $F$  is a new node not already on the tree:  $j \notin L_F^v$ . There are many paths to connect this new node to a node in the multicast tree, providing that the cost and delay of these paths can be arranged in ascending order and the minimum is selected. The energy of these paths should also be determined and the maximum selected; the link states of these paths should also be determined. Accordingly, the minimum cost, minimum delay, maximum energy, and minimum number of dead links are required in the path, to allow connection of the new node to the multicast tree.

There are also two possibilities when deleting a node from the multicast tree. The first case considers the deleted node as a leaf node, and the deletion process starts from the lowest level up, until it reaches the target router; this is the B-router for centric architecture, the M-router for hierarchical architecture, and the T-router for distributed architecture. The second case does not consider the delete node as a leaf node. In this case, the deletion process is achieved from the lowest level up and from to the highest level down. When moving up, the higher level routers are informed that the node has become null; when moving down, the lower level routers are informed that this node is null, and depending on the current architecture, the management router should find an alternative node; this is in case the node is used in a routing path. In cases of deletion, therefore, the multicast tree will be transformed from  $L_{F+1}$  to  $L_F$ .

## 6 Simulation and evaluation

### 6.1 Simulation environment

To test the proposed mixed IoT multicast architecture, a simulation environment was constructed using the simulation tool NS-2 [56]. The definition of IoT states that it may comprise different networks and the networks represented in the simulation environment are WSN, Mobile Ad-hoc Network (MANET), Satellite, radio-frequency identification (RFID), and High Altitude Platform (HAP). The configurations of these networks are shown in Table 1.

**Table 1** Simulations parameters of WSN, MANET, Satellite, RFID, and HAP

WSN		MANET		Satellite	
FRQ	2400 mHz	Size of packet	1 Mb	Type of satellite	LEO
TX	250 kb/s	Size of coverage area	2000 (m) × 2000 (m)	Inter-satellite distance	60 km
Power of RF	10 dBm	Number of nodes	100 node	Altitude	800 km
RX	94 dBm	Number of transmitted requests	5000 request	Delay of inter-satellite link	7.8 ms
Transmission mode (current drain)	11 (mA)	Interval between transmitted requests	500 ms	Degree of inclination	86 degree
Receiving mode (current drain)	19.7 mA	Time to live	Random range (4: 7) (ms)	Cross-seam inter-satellite link	Neglected
Battery	1250 (mAh), 1.5 (V)	Availability of link	Range (0: 1) (time unit)	Elevation mask	8.2 degree
Number of sensors	1500 (sensor)	Distance of transmission	Range (30: 210) (m)	Inter-satellite links per satellite	6
Size of coverage area	1000 (m) × 1000 (m)	Speed of node	Range (30: 60) Km/h	Uplink and downlink speed	1.5 mbps
		Probability of direction change	0	Bandwidth of inter-satellite	25 Mbps
RFID		HAP		Power	1 W.
FRQ of data channel	915 MHz	Maximum power of TX		Number of satellites	7
FRQ of control channel	930 MHz	Height of HAP's antenna			
Interference of inter-channel	Null	Maximum power of TX (per link)			
Fading	Null	Power of TX (common pilot channel)			
SNR	Range (7: 10) (dB)	Limitation of uplink load			

Table 1 continued

	RFID	HAP
Data rate	2 (Mbps)	Base station noise figure
Sensitivity of RX	-91 (dBm)	Transmit power of user equipment
Threshold of RX	-81 (dBm)	Add window of SHO
RFID transmission power	-45 dBm	Bit rate
Range of reading	1.62 m	Eb/N0 for uplink
		Eb/N0 for downlink
		STDEV of slow fading
		Orthogonality factor of downlink
Range of sensing	5.4 m	SHO gain for uplink and downlink
		Number of HAPs
Range of interference	7.1 m	Altitude
		Forward channel
Total number of RFID nodes	1200 nodes	Cell size
		Return channel
		5 dB
		21 dBm
		3 dB
		12.2 kbps
		5 dB
		9.5 dB
		4 dB
		0.9
		1 dB
		140 HAP
		20–50 km
		60 mb/s
		Between 0.5 and 10 km
		30 mb/s

To simulate the Internet, the environment comprises 100 routers distributed over six countries. Between 50 and 100 nodes—or things—are connected to each router, and a sink node is used to collect the information from the IoT nodes. The management of routers and other networks is achieved using six servers, which also execute interactive and intelligent application processes. Communication between the satellite and HAP can briefly be stated as follows: There are seven satellites of type LEO-Iridium, which communicate with 140 HAPs; each satellite can, therefore, communicate with 20 HAPs, and the satellites redirect data between HAPs. The bandwidth in the IoT simulation environment is between 1 and 2 Mb/s and is determined between routers, nodes, and sinks. The distance between IoT nodes determines each links' propagation delay; in the simulation, the propagation delay is between 25 and 30 ms, and the edge propagation delay is 5 ms. Selection of routing paths is achieved using the technique described in [51]. To reflect realistic IoT specifications, passive objects should be represented in the same way as active objects; RFID technology is, therefore, used to allow the passive objects to communicate with the IoT system. The active objects are equipped with devices and have technologies such that they can send and receive data easily. The percentage between passive things to active things equals 3:10. For accurate representation of data, video, audio, image and text data types are transmitted over the simulation environment. MPEG-2 is used for video data, PCM for audio data, and JPG for image data. Traffic is created using Poisson distribution and the objects, or things, are distributed randomly. The buffer size for each active object in the IoT system is between 100 and 1000 kb. The buffer size for satellites and HAP components was determined in [51]. For transport layer connections, either TCP or UDP is used, depending on the data type and IoT state. The metrics, which are used to determine the performance of the mixed IoT multicast architecture, are End-to-end delay at tight and moderate levels, packet loss, energy consumption, rate of architecture transformation, and throughput. The simulation interval is 6 min. To obtain accurate results, 10 simulation trails were completed and the result averages used.

There are two files, which are used to clarify how the functions in the IoT environment will be executed without human intervention. The first file is used to save the events which may be occurred in the IoT environment and the second file is used to save the actions. Mapping between these two files is achieved such as for each event in the first file there is an action in the second file. This action should be executed when the event is occurred.

## 6.2 Results and discussion

In this sub-section, the performance metric results are shown and discussed. The mixed IoT multicast architecture is compared to the centric, hierarchical and distributed IoT multicast architectures.

### 6.2.1 End-to-end delay

The large number of nodes found in IoT environments results in a large number of transmitted packets. To regulate the effect of the multicast architectures on the quality

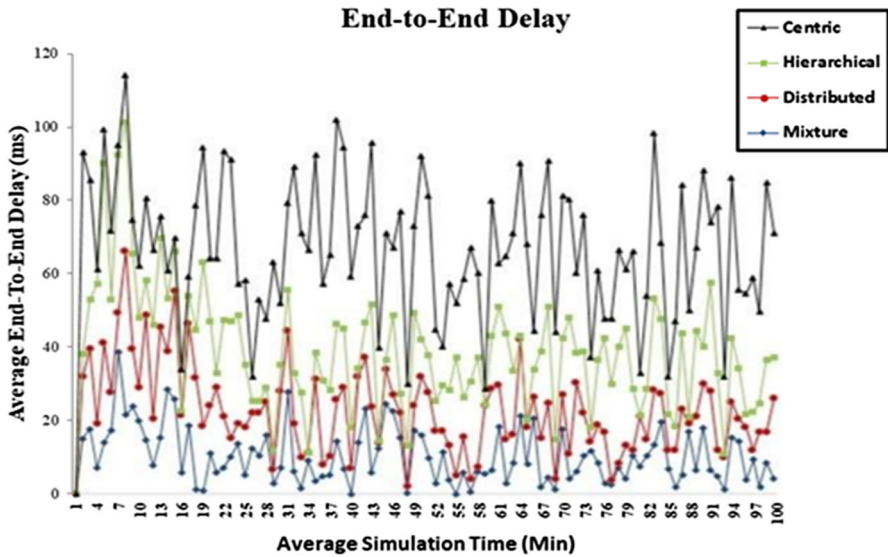


Fig. 12 Average end-to-end delay

of data transmission, the end-to-end delay should be measured; this is defined as the time taken to transmit a packet from source to destination, including the queuing delay, where the destination node belongs to the constructed multicast tree. Figure 12 shows the end-to-end delay results. The X-axis represents average simulation time and the Y-axis shows the average values of the end-to-end delay in ms; the graph shows that the best end-to-end delay is for the proposed mixed architecture. This could be explained by the transformation from one architecture to another, depending on the state of the IoT system. If a sudden event occurred in the IoT—such as a very large number of multicast requests that a B-router could not individually handle—the current multicast architecture may not be able to provide the best end-to-end delay. It could then be replaced by another multicast architecture, which may be more suited to handling the sudden event. The centric multicast architecture has a large end-to-end delay because it has a central router that handles all multicast functions, regardless of the number of join and leave requests; this represents a load that may cause a delay in the central B-router. The hierarchical and distributed multicast architectures show middling values for the end-to-end metric; however, the distributed multicast architecture has the best end-to-end delay. This could be explained by the distribution of multicast functions among multiple routers instead of one router, which decreases the processing load and in turn positively affects the end-to-end delay. It is notable that there are hesitations in the four multicast architecture plots. This could be explained by variations in the available bandwidth, which is an important specification in an IoT environment. The passive objects may affect results in the routing paths by changing their states to dead; this event forces the proposed multicast architecture to find alternative paths, which may not provide the same QoS to the transmitted data.

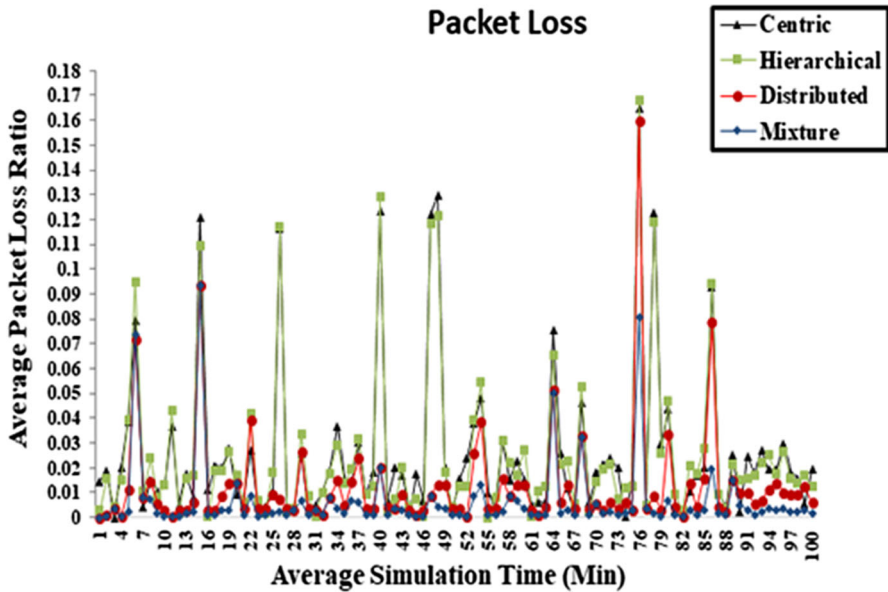


Fig. 13 Average packet loss ratio

### 6.2.2 Packet loss

To measure the effectiveness of our proposed mixed IoT multicast architecture, packet loss should also be measured. The packet loss metric determines whether the proposed algorithm represents an overload while a packet is processed. To measure packet loss, the total sent packets should be known; this metric is, therefore, defined as the percentage of the number of lost packets relative to the number of sent packets, using a predetermined window and buffer size. Figure 13 shows the average packet loss results; the X-axis represents the average simulation time and the Y-axis represents the average packet loss values. The proposed mixed multicast architecture has the lowest packet loss value, because the mixed multicast algorithm considers the buffer size as a link cost parameter. The energy and delay are also considered. Each packet should, therefore, be routed using the best path in the multicast tree considering the energy, cost, and delay, as well as link state, which should be active. Notably, the centric architecture has the biggest packet loss values because there is no option for the hosts to connect with the B-router in the case of a join or leave request. If there are minor or major upgrades, the B-router should be informed so that it can remain up-to-date with the current state of the multicast group. The centric multicast architecture does not consider the buffer size in addition to the energy parameters; thus, if a congestion occurs in a multicast tree route, the packet loss percentage will be negatively affected. For hierarchical and distributed multicast architectures, the packet loss values are middling, but the distributed multicast architecture has lower packet loss values than the hierarchical multicast architecture. Figure 13 shows simulation time points, such as 40, 47, 48, and 76, at which the packet loss percentage has extremely high values. This



could be explained by a sequence of bottlenecks, such as lower energy nodes, limited bandwidth, or many passive objects. The bottleneck sequence occurs in a narrow time period, which affects the selection of alternative paths, and transformation from the current multicast architecture to a more suitable architecture. The packet loss average at these simulation time points is, therefore, notably larger.

### 6.2.3 Throughput

Throughput is one of the most important performance measurement parameters of the proposed multicast architecture. The throughput is measured by the total number of packets sent from sources to destinations within the multicast groups, and which are generated in the simulation environment under IoT specifications and requirements. Figure 14 shows the throughput results. The proposed mixed multicast architecture has the highest throughput values and the centric multicast architecture has the lowest throughput values. This is because the proposed multicast architecture can adapt itself to the current state and selects the multicast architecture that can tolerate problems relating to IoT system overload. At the start of the simulation throughput values are high, after which the plots became stable. The throughput values for the distributed multicast architecture are higher than those for the hierarchical multicast architecture. Moreover, the overall throughput increases in parts of the simulation environment and decreases in other parts; this is because accurate representation of an IoT environment requires the simulation environment to comprise both satellite and HAP networks. This means that the Internet signal within these networks will have a high bit error rate, which affects the packet loss and delay metrics; this in turn means that throughput

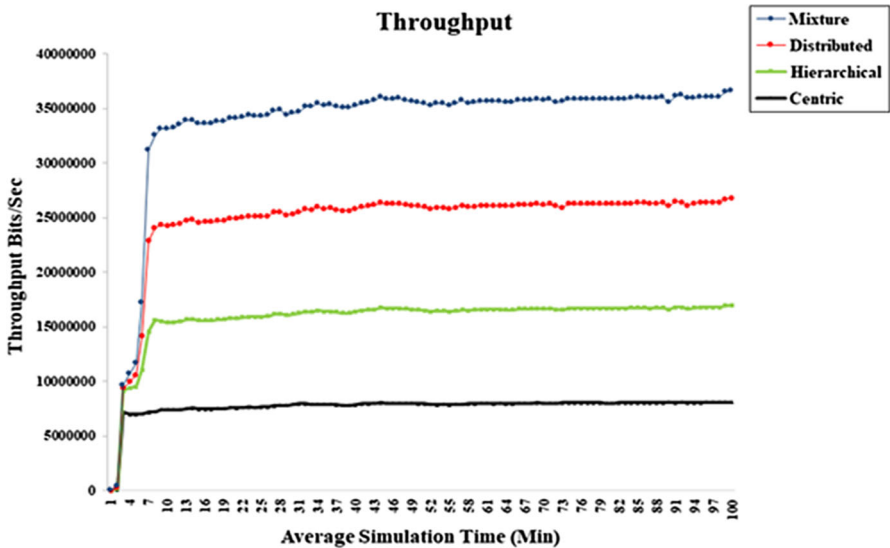


Fig. 14 Average throughput

values decrease in parts of the environment that use the space Internet signal, compared to those in parts of the environment using the ground Internet signal.

#### 6.2.4 Energy consumption

Energy consumption is another important metric, in that constructed multicast groups in IoT systems may comprise energy-based nodes. Less energy consumption is a target of mixed multicast architecture, because an increase in the node active life time will directly increase the total number of packets transmitted. The energy consumed in the proposed mixed multicast architecture should, therefore, be compared to the energy consumed in the centric, hierarchical, and distributed multicast architectures. The simulated IoT environment comprises energy-based nodes for network types WSN, RFID, and MANET. The energy consumption metric is measured in these networks using techniques stated in [57–59]. Figures 15, 16 and 17 show the energy consumption results in the WSN, RFID, and MANETs, respectively. The X-axis represents the average simulation time (10 points), whilst the Y-axis represents the average energy consumption values. The values over most simulation times for the mixed multicast architecture are the lowest. The distributed and the hierarchical multicast architectures come second and third after the proposed one, respectively, while the centric multicast architecture has the highest energy consumption values. This may be explained by high packet loss and delay that affect throughput of the architecture. High rates of packet loss may result in high retransmission rates, which in turn lead to a high rate of energy consumption. In hierarchical multicast architecture, the multicast functions are distributed over more than one router, which alleviates the centralization problems; thus, the hierarchical multicast architecture has energy consumption values less than those

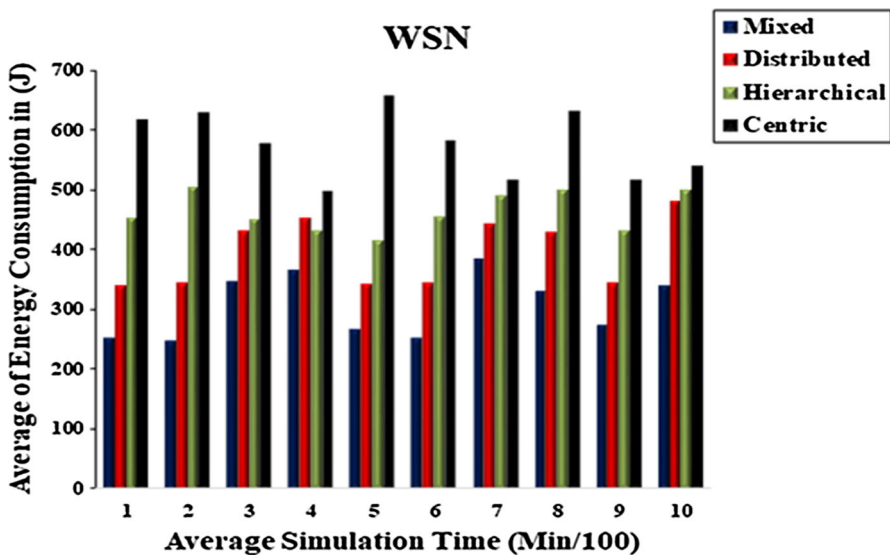


Fig. 15 Average energy consumption in the WSN

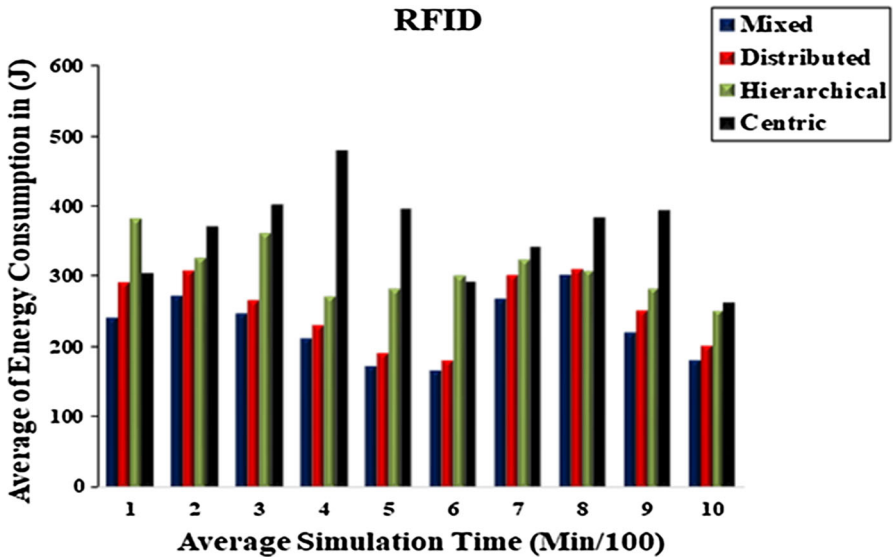


Fig. 16 Average energy consumption in the RFID network

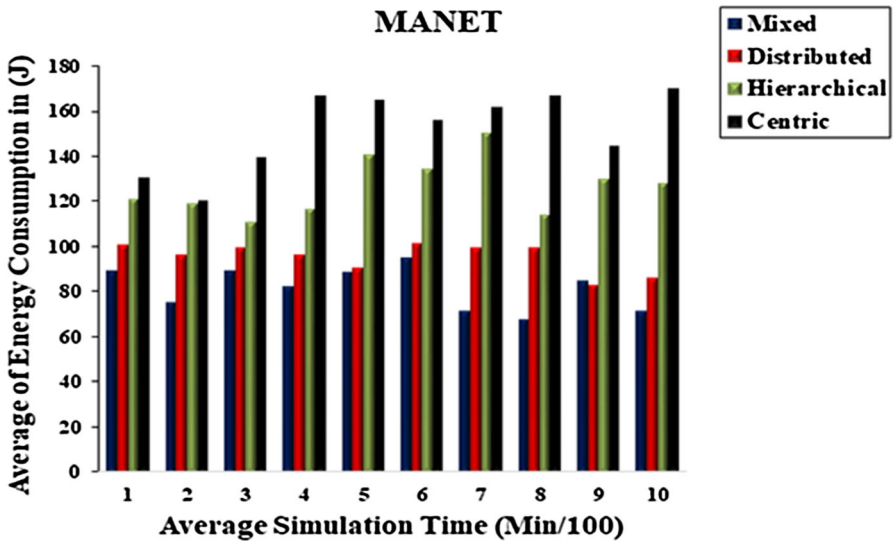


Fig. 17 Average energy consumption in the MANET

of the centric multicast architecture. In the distributed multicast architecture, multicast functions will be handled by a large number of T-routers that reduce the processing overload; thus, the distributed multicast architecture has energy consumption values less than those of the hierarchical multicast architecture. The main difference between the proposed multicast architecture and the distributed architecture is consideration of the IoT system status; in some IoT systems, the centric multicast architecture may

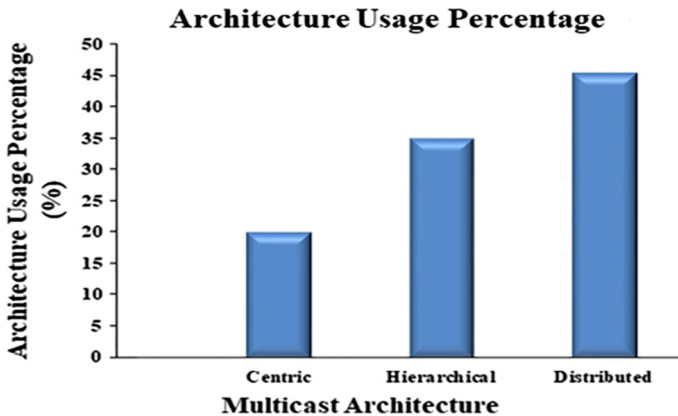


Fig. 18 Percentage use of traditional multicast architectures in mixed architecture

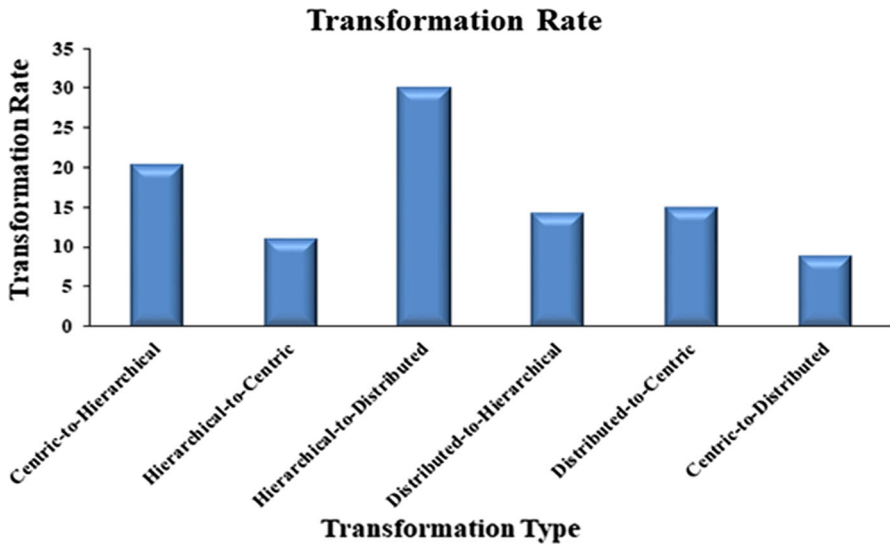


Fig. 19 Transformation rate between traditional multicast architectures in mixed architecture

provide better results than other multicast architectures, due to easy communication between the B-router and other system routers.

### 6.2.5 Transformation between multicast architectures

As stated above, the mixed multicast architecture uses the distributed, hierarchical, and centric multicast architectures, but also considers the current state of IoT network. It is important that the transformation process from one multicast architecture to another be accomplished accurately and smoothly. Figure 18 shows the usage percentage for each traditional multicast architecture. The X-axis represents the usage percentage and the Y-axis represents the architecture type and it can be seen that the

**Table 2** Sample results: (a) Delay, packet loss, and throughput, (b) Energy consumption, (c) Architecture usage percentage, (d) transformation rate

Delay	Packet loss						Throughput							
	C	H	D	M	C	M	C	H	D	M	C	H	D	M
(a)	93.254	18.352	20.361	20.101	0.019934	0.014934	0.000166	0.000151	7470448	8098363	9006351	9042966	9042966	
	78.212	28.817	21.951	5.982	0.038924	0.039124	0.011324	0.002265	7474260	8101094	9018321	9049097	9049097	
	62.945	52.911	28.125	24.918	0.079151	0.094951	0.071651	0.073884	7508681	8138485	9053763	9081570	9081570	
	105.921	58.101	67.112	23.975	0.004302	0.010302	0.008302	0.008302	7512388	8138101	9049053	9083493	9083493	
	64.201	102.715	76.992	25.901	0.014955	0.023755	0.014655	0.007328	7587616	8211276	9134193	9166501	9166501	
	99.219	104.247	79.924	52.842	0.010626	0.008326	0.005626	0.001875	7573104	8201877	9120341	9147414	9147414	
	116.381	111.245	78.121	30.106	0.013	0.013	0.0031	0.00062	7605922	8238441	9149104	9186500	9186500	
	45.916	97.325	58917	26.291	0.036924	0.043024	0.000476	0.000159	7652704	8272221	9197214	9226153	9226153	
	76.297	98.254	49.218	21.815	0.005	0.0018	0.003	0.0006	7630341	8263748	9169907	9208069	9208069	
	64.448	74.981	39.915	30.750	0.017848	0.015948	0.002848	0.001424	7635946	8264557	9175294	9215561	9215561	
WSN	MANET						RFID							
	C	H	D	M	C	H	D	M	C	H	D	M	C	M
(b)	618.3	452.5	340.8	250.3	130.1	120.6	100	88.9	303.3	380.6	290	240.4	240.4	
	629.1	505.3	345.9	245.1	120	118.9	95.7	75	370.4	324.8	307.1	270.3	270.3	
	578	451.2	432	345.6	139.2	110.2	99	89	400.2	359.5	265.4	245.9	245.9	
	499	432.9	454.2	364.9	166.4	115.9	96.1	81.9	477.7	271.7	230	210.7	210.7	
	657.8	415.5	342.5	265	164.5	140.1	90.4	88.2	395.5	282	191	171.5	171.5	
	583.1	456.4	344.5	251	155.9	133.9	100.8	94.4	291.2	300.8	180.9	166.3	166.3	
	517.3	490.2	443.9	383.3	161.2	149.6	99	70.8	340.8	323.7	299.8	267.2	267.2	
	631.5	501	429	329.5	166.1	113.9	98.8	67.3	381.9	305.9	308.7	300.5	300.5	
	517.9	432.9	344.4	271.8	144.3	129.4	82.6	81.1	393.5	281.1	249.7	219.2	219.2	
	539.4	499.5	480.9	337	169.8	127.9	85.8	71	263.7	251.2	200.9	180.3	180.3	

Table 2 continued

Centric	Hierarchical	Distributed
(c)		
19.97	34.8	45.23
(d)		
Centric-to-Hierarchical	20.5	
Hierarchical-to-centric	11.07	
Hierarchical-to-distributed	30.1	
Distributed-to-hierarchical	14.3	
Distributed-to-centric	15.13	
Centric-to-distributed	8.9	

distributed architecture has a highest percentage value. This may be explained by the data transmission overload, which is a result of the very large number of nodes in the multicast groups. This data transmission overload results in periodic generation of high numbers of multicast requests, which require large number of routers to handle them. Figure 19 shows the transformation rate between one multicast architecture and another; the X-axis represents the transformation rate and the Y-axis represents the transformation type.

A sample result with respect to end-to-end delay, packet loss, throughput, energy consumptions, architecture usage percentage and transformation rate is shown in Table 2.

## 7 Conclusion

In this paper, a mixed multicast architecture for IoT environments is proposed, which selects the traditional multicast architecture—centric, hierarchical, or distributed—most suited to the current state of an IoT system. The paper introduces an algorithm to construct the multicast tree using the proposed architecture and then presents a case study to describe each traditional multicast architecture in an IoT environment. To measure performance of the proposed architecture, a simulation environment was constructed using the network simulation package NS-2; a performance comparison between the proposed multicast architecture and the three other traditional other multicast architectures was then completed. The performance metrics used were end-to-end delay, packet loss, throughput, and average energy consumption; to ensure that the proposed multicast architecture was working smoothly, a usage percentage for each architecture, and a transformation rate was measured. The results showed that the mixed multicast architecture improves upon traditional multicast architectures as follows: End-to-end delay decreased by 25.16%; packet loss was reduced by 38.12%; throughput increased by 28.57%; and average energy was reduced by 23.11, 9.95, and 14.51% for WSN, RFID, and MANET respectively. Finally, transformation between multicast architectures showed that the distributed multicast architecture has the highest percentage utilization, which is consistent with the characteristics of IoT environments.

## References

1. Malik S, Srinivasan K, Khan S (2012) Convergence time analysis of open shortest path first routing protocol in internet scale networks. *IEEE Electron Lett* 48(19):1188–1190
2. Al-Fuqaha Ala, Khreishah Abdallah, Guizani Mohsen, Rayes Ammar, Mohammadi Mehdi (2015) Toward better horizontal integration among IoT services. *IEEE Commun Mag* 53(9):72–79
3. Arabnia HR (1990) A parallel algorithm for the arbitrary rotation of digitized images using process-and-data-decomposition approach. *J Parallel Distrib Comput* 10(2):188–193
4. Arabnia HR (1995) Distributed stereocorrelation algorithm. *Int J Comput Commun* (Elsevier Science) 19:707–712
5. Arabnia HR, Taha TR (1998) A parallel numerical algorithm on a reconfigurable multi-ring network. *J Telecommun Syst Special Issue Interconnect Netw* 10:185–203
6. He X, Arabnia HR (2006) Design of a uni-directional switch. *Int J Comput Sci Netw Secur (IICSNS)* 6(6):130–138

7. Arabnia HR, Smith JW (1993) A reconfigurable interconnection network for imaging operations and its implementation using a multi-stage switching box. In: Proceedings of the 7th Annual International High Performance Computing Conference. The 1993 High Performance Computing: New Horizons Supercomputing Symposium. Calgary, Alberta, Canada, June, pp 349–357
8. Bhandarkar SM, Arabnia HR (1993) The multi-ring reconfigurable multiprocessor network for computer vision. In: Proceedings of the IEEE Workshop on Computer Architectures for Machine Perception (CAMP'93), IEEE Computer Society Press (Eds.: Magdy A. Bayoumi, Larry S. Davis, and Kimon P. Valavanis), New Orleans, Louisiana, Dec. 15–17, pp 180–191
9. Arabnia HR (1994) Broadcasting mechanisms on the reconfigurable MultiRing network. In: Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, July 11–15, 1994, Georgia Institute of Technology, Atlanta, Georgia, vol 3, pp 1076–1080
10. Arabnia HR (1997) The stereo correspondence problem on a ring-based network. In: Proceedings of the 1997 Aizu International Symposium on Parallel Algorithms/Architectures Synthesis (PAs'97), IEEE/ACM, March 17–21, 1997, Aizu-Wakamatsu, Japan. Invited paper, pp 265–276
11. He X, Arabnia HR (2004) Scalable switch for bi-directional multiring network. In: Proceedings of The 4th IEEE International Symposium on Signal Processing and Information Technology, ISSPIT'04, (IEEE Signal Processing and IEEE Computer Society), December 18–21, 2004, Rome, Italy, pp 279–282
12. Ahmed AM, Kong X, Liu L, Xia F, Abolfazli S, Sanaei Z, Tolba A (2017) BoDMaS: bio-inspired selfishness detection and mitigation in data management for ad-hoc social networks. *Ad Hoc Netw* 55:119–131
13. Xia F, Liaqat HB, Ahmed AM, Liu L, Ma J, Huang R, Tolba A (2016) User popularity-based packet scheduling for congestion control in ad-hoc social networks. *J Comput Syst Sci* 82(1):93–112
14. Abualigah LMQ, Hanandeh ES (2015) Applying genetic algorithms to information retrieval using vector space model. *Int J Comput Sci Eng Appl* 5(1):19
15. Said O (2013) Accurate performance evaluation of internet multicast architectures. *KSII Trans Internet Inf Syst* 7(9):2194–2212
16. Ammar M, Russello G, Crispo B (2018) Internet of things: a survey on the security of IoT frameworks. *J Inf Secur Appl* 38:8–27
17. Campbell W (2018) The Impact of the Internet of Things (IoT) on the IT Security Infrastructure of Traditional Colleges and Universities in the State of Utah. In: *The Internet of People, Things and Services*, pp 132–153
18. Jeon Soobin, Jung Inbum (2018) Experimental evaluation of improved IoT middleware for flexible performance and efficient connectivity. *Ad Hoc Netw* 70:61–72
19. Zikria YB, Afzal MK, Ishmanov F, Kim SW, Yu H (2018) A survey on routing protocols supported by the Contiki Internet of things operating system. *Future Gener Comput Syst* 82(5):200–219
20. Nascimento N, De Lucena C (2017) FIoT: an agent-based framework for self-adaptive and self-organizing applications based on the Internet of Things. *Elsevier Inf Sci* 378(1):161–176
21. Yongrui Q et al (2016) When things matter: a survey on data-centric internet of things. *J Netw Comput Appl* 64(4):137–153
22. Stojkoska B, Trivodaliev K (2017) A review of Internet of Things for smart home: challenges and solutions. *Elsevier J Cleaner Product* 140(3):1454–1464
23. Said O, Masud M (2013) Towards internet of things: survey and future vision. *Int J Comput Netw (IJCN)* 5(1):1–17
24. Ouaddah A et al (2017) Access control in the Internet of Things: big challenges and new opportunities. *Elsevier Comput Netw* 112(15):237–262
25. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *J Supercomput* 73(11):4773–4795
26. Ejaz W, Naeem M, Shahid A (2017) Efficient energy management for the internet of things in smart cities. *IEEE Commun Mag* 55(1):84–91
27. Said O, Albagory Y (2017) Internet of things-based free learning system: performance evaluation and communication perspective. *IETE J Res* 63(1):31–44
28. Mahalaxmi G, Rajakumari KE (2017) Multi-agent technology to improve the internet of things routing algorithm using ant colony optimization. *Ind J Sci Technol* 10(31):1–8



29. Shang W, Yu Y, Droms R (2016) Challenges in IoT Networking via TCP/IP Architecture, NDN Technical Report NDN-0038. Revision 1: February 10, 2016. <https://named-data.net/wp-content/uploads/2016/02/ndn-0038-1-challenges-iot.pdf>. Accessed 21 Apr 2018
30. Perlman R et al. (1999) Simple multicast: a design for simple, low-overhead multicast. Internet draft. <http://tools.ietf.org/html/draft-perlman-simple-multicast-02>. Accessed 21 Apr 2018
31. Wg P, Adams A, Nicholas J, Siadak W (2004) Protocol independent multicast—dense mode (PIM-DM): protocol specification. Internet draft
32. Ballardie A, Cain B, Zhang Z (1998) Core based trees (CBT version 3) multicast routing. Internet draft. <http://www.ietf.org/proceedings/44/I-D/draft-ietf-idmr-cbt-spec-v3-01.txt>. Accessed 21 Apr 2018
33. Deering S et al. (1996) Protocol independent multicast-sparse mode (PIM-SM): motivation and architecture. [https://pdfs.semanticscholar.org/7bd4/6e6a9dbc3ede45fe8cc598c608fb0a8a1c72.pdf?\\_ga=2.259887306.1336837772.1524313230-747404841.1522487116](https://pdfs.semanticscholar.org/7bd4/6e6a9dbc3ede45fe8cc598c608fb0a8a1c72.pdf?_ga=2.259887306.1336837772.1524313230-747404841.1522487116). Accessed 21 Apr 2018
34. Moy J (1994) Multicast Extension to OSPF. RFC 1584. <https://tools.ietf.org/html/rfc1584.html>. Accessed 21 Apr 2018
35. Moy J (1998) OSPF version 2. RFC 2328. <http://www.ietf.org/rfc/rfc2328.txt>. Accessed 21 Apr 2018
36. Waitzman D, Partridge C (1988) Distance vector multicast routing protocol. RFC 1075. <http://www.ietf.org/rfc/rfc1075.txt>. Accessed 21 Apr 2018
37. Yang Y et al (2008) A service-centric multicast architecture and routing protocol. *IEEE Trans Parallel Distrib Syst* 19(1):35–51
38. Shrivastava L, Dauria S, Tomar G (2011) Performance evaluation of routing protocols in MANET with different traffic loads. In: *Proceedings of IEEE International Conference on Communication Systems and Network Technologies*, Jammu, India, pp 13–16
39. Santhi S, Sadasivam G (2011) Performance evaluation of different routing protocols to minimize congestion in heterogeneous network. In: *Proceedings of IEEE International Conference on Recent Trends in Information Technology*, India, pp 336–341
40. Ionela C, Croitoru V, Popescu A (2011) Comparative performance evaluation of IPSAG and HC-IPSAG cognitive radio routing protocols. In: *International Symposium on Signals, Circuits and Systems (ISSCS)*, Romania, pp 1–4
41. Hachisuka Y, Hasegawa H, Sato K (2011) Design algorithm of waveband multicast tree in hierarchical optical path networks that utilizes grouping of destination node sets. In: *SPIE Proceedings Hierarchical and Heterogeneous Optical Networks*, China, pp 1–6
42. Wang F, Xiong Y, Liu J (2010) mTreebone: a collaborative tree-mesh overlay network for multicast video streaming. *IEEE Trans Ransactions Parallel Distrib Syst* 21(3):379–392
43. Polishchuk T et al. (2012) Scalable architecture for multimedia multicast internet applications. In: *IEEE International Symposium, World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, San Francisco, Canada, pp 1–6, June 25–28
44. Silva F et al. (2012) On the analysis of multicast traffic over the entity title architecture. In: *Proceedings of IEEE International Conference on Networks (ICON)*, Singapore, pp 30–35, Dec. 12–14
45. Jardim S et al. (2012) Applying advanced network resource provisioning in future internet systems. In: *IEEE Latin-America Conference on Communications (LATINCOM)*, Cuenca, Ecuador, 7–9 Nov. <https://doi.org/10.1109/latincom.2012.6506000>
46. Atlas A et al. (2012) An architecture for multicast protection using maximally redundant trees. Internet Draft, July 12
47. Rahmani R, Kanter T (2015) Layering the internet-of-things with multicasting in flow-sensors for internet-of-services. *Int J Multimed Ubiquitous Eng* 10(12):37–52
48. Martynov N (2014) Secure multicast with source authentication for the Internet of things, technical university of Denmark, degree project- in- second level. Stockholm, Sweden
49. Akkermans S, Bachiller R, Matthys N (2016) Towards efficient publish-subscribe middleware in the IoT with IPv6 multicast. In: *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, 22–27, May, pp 1–6
50. Antonini M et al. (2014) Lightweight multicast forwarding for service discovery in low-power IoT networks. In: *IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, USA, 17–19 Sept, pp 133–138
51. Mahmud A, Kanter T, Rahmani R (2012) Flow-sensor mobility and multicast support in Internet of Things' virtualization. In: *IEEE International Conference on ICT Convergence (ICTC)*, Jeju, South Korea, 15–17 Oct, pp 16–22

52. Wang J et al (2016) A distributed algorithm for inter-layer network coding-based multimedia multicast in Internet of Things. Elsevier Comput Electr Eng J 52:125–137
53. Huang J, Duan Q, Zhao Y, Zheng Z, Wang W (2017) Multicast routing for multimedia communications in the Internet of Things. IEEE Internet Things J 4(1):215–224
54. Pan MS, Yang S-W (2017) A lightweight and distributed geographic multicast routing protocol for IoT applications. Comput Netw 112:95–107
55. Xiao-yong L, Xiao-lin G (2006) Merging source and shared trees multicast in MPLS networks. In: Proceedings of Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), Taiwan, pp 23–28
56. Said O (2016) Analysis, design and simulation of Internet of Things routing algorithm based on ant colony optimization. Wiley Int J Commun Syst. <https://doi.org/10.1002/dac.3174>
57. Zhou H et al (2011) Modeling of node energy consumption for wireless sensor networks. Wirel Sensor Netw 3:18–23. <https://doi.org/10.4236/wsn.2011>
58. Yan X, Liu X (2013) Evaluating the energy consumption of the RFID tag collision resolution protocols. Springer Telecommun Syst J 52(4):2561–2568
59. Xiao H, Ibrahim DM, Christianson B (2014) Energy consumption in mobile ad hoc networks. In: IEEE Conference on Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 6–9 April, pp 2599–2604