



MSDN-IoT multicast group communication in IoT based on software defined networking

Youssef Baddi¹ · Anass Sebbar² · Karim Zkik⁴ · Yassin Maleh³ · Faysal Bensalah⁵ · Mohammed Boulmal²

Received: 26 November 2022 / Accepted: 3 April 2023 / Published online: 26 April 2023
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

The Internet of Things (IoT) is a fast-growing paradigm in on-going research fields and industrial domains that includes wireless sensor networks, cloud computing, big data, smart cities, large-scale industrial IoT services, etc. In the Internet of Things architecture, wireless sensor networks can be an important factor in optimizing IoT solutions. This architecture includes various gateways, controllers, application servers, and IoT clouds. In many Internet of Things (IoT) applications, messages may need to be distributed to a specific set of objects or nodes using the multicast communication mode. Existing IoT multicast routing algorithms are not performant or efficient enough to support multimedia group-based applications in an IoT context since they are primarily focused on ad hoc sensor networking scenarios. In this paper, we propose MSDN-IoT, a novel multicast software-defined network based on a hierarchical shared multicast tree, and a flexible set of SDN controller modules, like group management for dynamic multicast services. Our results show the effectiveness of our protocol over state-of-the-art protocols in terms of end-to-end delay, end-to-end delay-variation, scalability, and other metrics.

Keywords Software-defined networking (SDN) · IoT · Multicast IP · MSDN-IoT · Multi-controller

1 Introduction

Today's world knows a continuous development of Internet of Things (IoT) technology. IoT applications such as smart homes, smart factories, and smart cities are becoming more complex and larger in scale. Most of these IoT applications contain a large number of distributed sensors, which have multiple relationships and are dynamically connected to each other. In many IoT applications, messages may need to be sent to a certain group of objects or nodes. This can be done with the multicast communication mode. However, IoT technology and applications provide many advantages for the group's communication technology, which allows Internet customers to profit from a wide range of Internet services.

We usually notice the distribution of newsletters, database synchronization, and audio or video conferencing [1]. Multicasting is an important operation in the Internet of Things environment because it allows for effective group communication. In the past few years, the provider's researchers and developers have come up with and put into place a number of multicast protocols that help with routing, access control, and scalability in IP networks.

Existing multicast routing technologies used in wired networks and IoT environments are mainly concentrated in IP networks or ad hoc sensor networking scenarios. Hence, it does not have sufficient responsiveness and robustness to support multicast group communication applications in the IoT environment, such as multimedia applications. For such an intermittently connected IoT environment with high heterogeneity domains, traditional multicast protocols and algorithms designed for wired networks and wireless ad hoc networks cannot be applied.

To address this problem, we propose MSDN-IoT, the multicast software-defined networking solution for IoT environments, which is a novel multicast software-defined network architectural solution based on hierarchical shared multicast trees and a flexible set of SDN controller modules, including the Multicast Tree Computing module, which is responsible

✉ Youssef Baddi
baddi.y@ucd.ac.ma

¹ STIC Lab, UCD, El Jadida, Morocco

² TICLAB, UIR, Rabat, Morocco

³ LaSTI laboratory, ENSAK, SMSU, Beni Mellal, Morocco

⁴ Esaip école d'ingénieur, CERADE, Angers, France

⁵ LERSEM Lab, National School of Business and Management, Chouaib Doukkali University, El Jadida, Morocco

for building an optimal multicast tree; the RP Management Module, which is responsible for managing RP events; and the Group Management Module, which is responsible for handling all multicast group membership events.

The remainder of the paper is organized as follows: Sect. 2 of this article discusses SDN topologies, IP multicast protocols, and some background information and terminologies. We provide some related, state-of-the-art works in Sect. 3. We discuss our suggested architectural approach in Sect. 4 before conducting a performance study to demonstrate the effectiveness of our model in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Background and terminology

2.1 Multicast IP

There are two approaches to implementing traditional IP communication: one is based on unicast communication, and the other is based on broadcast communication [2]. The issue of group communication cannot be resolved using conventional unicast and broadcast communication techniques. In an IP network, multicasting is the process of distributing data packets using best-effort transmission to a specific group of nodes. The fundamental concept is that source hosts, like multicast sources, only deliver one piece of data at a time. Only hosts in the multicast group are able to receive the data; other hosts are still unable to receive the data. The destination address is the address of the multicast group; all recipients in the multicast group can receive the same copy of the data. The multicast router in this scenario is in charge of duplicating the data to however many recipients are included in the multicast group. Streaming media and some other network services and applications typically employ IP multicast. Figure 1 illustrates the two categories of intra-domain multicast routing protocols based on the topological distribution of receivers: dense mode and sparse mode.

Traditional IP multicast has faced various challenges from the beginning; the first is Deering [3]. Many limitations have been discovered in the literature by researchers from various perspectives [4, 5]. Although these limitations have been known for a long time, most of them still exist and are very relevant to inspections.

2.2 SDN-IoT

To present the Internet of Things environment, it can usually be defined as a set of scalable sensor networks that connect a set of smart objects connected to each other (i.e., machine-to-machine) and connected to the Internet. The smart object consists of a large number of sensor nodes. Its task is to sense the physical environment, and there are a small number of

receiving nodes (or base stations) whose task is to store and process sensory readings [6, 7].

The need to implement multicast IP in an IoT environment is caused by the existence of multiple IoT use cases that rely on or have multiple copies of the same packet when using multicast group communications instead of multiple unicast communications. So, the basic idea is to make multicast communication more efficient by keeping similar unicast packets from being sent to a group of receivers more than once.

Software-defined networking (SDN) [8–12] is a new network paradigm in which a central server called a controller treats the network as a whole and determines its behavior. In SDN, and as presented in the Fig. 2, different devices on the network become "simple devices" for routing messages. This is called the data plane; on the contrary, the control and logic of the network are only implemented in the controller, which is called the control plane.

Using this new method, network behavior can be modified dynamically and quickly, providing a significant advantage in programmability. In addition, IT professionals can control and manage all network devices from a centralized point in an automated manner, thereby providing flexibility for the entire system. These orchestration and management automations reduce operating costs, minimize human intervention, and save technical operation time, thereby prioritizing innovation time. With all the advantages brought by SDN, researchers hope to integrate this new network design with certain technologies that have emerged in recent years (such as IoT, 5G, VANet, etc.).

This integration solves many problems, such as the complexity of traffic engineering, high cost, and security issues. In fact, with the exponential growth of connectivity and scalability requirements, SDN provides powerful network solutions for emerging technologies to enhance robustness. Their communication system has powerful functions for automatic and dynamic network management. Therefore, many professionals are committed to deploying these solutions: IoT-based SDN, 5G-based SDN, VANet-based SDN, etc., to get a better user experience in all applications. Indeed, SDN is a powerful tool that can dynamically simplify the policy implementation and network reconfiguration of IoT and 5G infrastructure. However, because these technologies are coupled together, it brings some challenges, such as scalability and mobility, security and privacy, complete upgrade and conversion of the old version of the network to SDN, data volume, etc.

3 State of the art

In recent years, due to the basic role played by SDN and the many problems that SDN solves in many systems, SDN has aroused increasing interest. In this way, the idea behind

Fig. 1 Multicast routing protocols

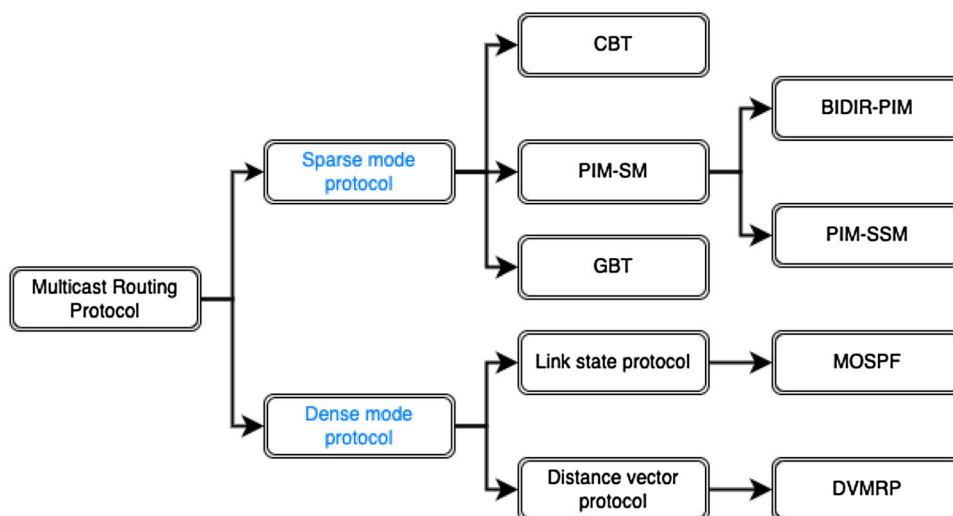
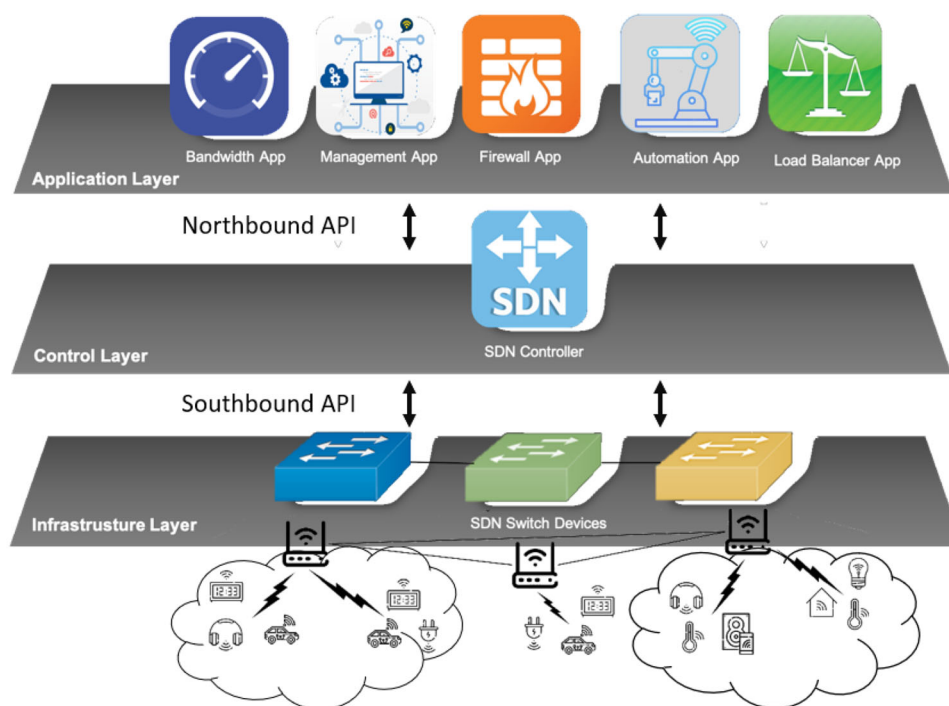


Fig. 2 Software defined network based on IoT model



SDN is based on many previous efforts, which have pointed out the importance of SDN through several studies and research work [3–5, 13–17]. SDN introduces logical centralized controls with open interfaces and provides APIs for the abstraction of network elements to program their forwarding behavior. It is based on the idea of decoupling the control plane from the data plane, as shown in Fig. 2. In this section, we illustrate the summary study of proposed multicast routing protocols in IoT based on SDN in the literature. We structure our survey study under classical solutions, IoT-based solutions, and SDN-based solutions. Many survey works categorize classical multicast routing protocols as tree-based multicast routing (TBM) protocols or

geographically based multicast (GMT) routing protocols. In the tree-based multicast routing protocols, there is a large set of research contributions looking to compute an optimized SPT or shared tree.

Baddi et al. [18–23] presented an RP-selection protocol to select an optimal RP node and then used this note to build the multicast shared tree. This protocol uses VNS-RP [18–20], VND-RP [21], and PGRASP-RP [22, 23] algorithms to select the optimal node.

Su et al. [24] proposed an energy-optimal multicast routing protocol for wireless sensor networks called OCast. The two proposed versions of the algorithms, centralized and distributed, are provably optimal when the number of desti-

nations is small. The proposed model is inspired by scenarios in which sensors report to a limited number of base stations or data must be replicated on a limited number of additional nodes.

The multicast trees identified and built based on the oCast [24] algorithm may not satisfy end-to-end delay requirements. Xie et al. [25] proposed DB-oCast [25], an extended version of oCast [24], to discover an optimal multicast trees under a predefined delay bound.

Park et al. [26] proposed a multicast routing protocol for real-time data dissemination to a multicast group, which is defined as data delivery to each member of the multicast group within the desired time deadline defined initially before the multicast tree is created. Park et al. [26] proposed constructing a virtual multicast tree first, then computing a multicast tree based on paths that can be close to the virtual multicast tree paths.

Sanchez et al. [27] propose BRUMA as a geographic multicast routing (GMR) in a wireless sensor network. The proposed solution is designed for a topology with very limited control overhead and overall bandwidth consumption. The proposed solution is a beacon-less geographic multicast routing protocol; unlike other GMT solutions, BRUMA [27] nodes do not need to exchange beacon messages between themselves to gather information about the position of their neighbors.

Partitioned Multicast Tree (PAMTree) was proposed by Santamaria and colleagues [28] as a centralized multicast routing protocol for vehicular networks. The critical distance is introduced as a new metric for comparing proposed solutions by Santamaria et al. [28]. The critical distance is seen as the real-time constraint in the communication between a source and the furthest member since an end-to-end delay metric is proportional to a physical distance in wireless sensor networks.

Meng-Shiuan Pan et al. [29] proposed a novel lightweight and distributed geo-geographic multicast routing protocol. The proposed scheme contains three phases. First, the first phase selects intermediate nodes to reach multicast destinations. Then, the second phase removes loops and trims routes constructed in the first phase. Finally, the last phase checks if the selected multicast links can be further merged.

Mauro Conti et al. [30] proposed a reliable and secure multicast routing protocol (REMI) for IoT networks to enable efficient communication in low-power and lossy networks such as the IoT. REMI [30] uses a cluster-based routing approach that triggers a faster multicast dissemination of messages within the network.

Jun Huang et al. [31] proposed two algorithms for setting up a multicast routing tree for multimedia data transmissions. The suggested techniques use a perturbation theory method to combine all weights into a comprehensive measure, which is then used to search a multicast tree using the spanning tree and shortest path tree algorithms.

Table 1 Summary of proposed solution in the literature 1= Classical solution, 2= SDN-based, 3= IoT based, 4 =TBM, 5= GMT

Solution	1	2	3	4	5
Baddi et al. [18–23]	Yes	No	No	No	No
OCast [24], DB-oCast [25]	No	No	No	Yes	No
H. Park et al. [26]	No	Yes	No	Yes	No
BRUMA [27]	No	No	Yes	No	Yes
PAMTree [28]	No	No	Yes	Yes	No
Meng-Shiuan Pan et al. [29]	No	No	Yes	No	Yes
REMI [30]	No	No	Yes	Yes	No
Jun Huang et al. [31]	No	No	Yes	Yes	No
EBSPT [32]	No	Yes	No	Yes	No
LAMA [33]	No	Yes	No	Yes	No
DCN [34]	No	Yes	No	Yes	No

Using multicast techniques in an SDN environment is a real challenge for network experts. Some research has tried to propose applying the SDN technology to the legacy multicast mechanism; others have proposed a native SDN-based solution.

Tim et al. [32] proposed an early branching shortest path tree (EBSPT) as an SDN-based SBT multicast SBT, called [32]. The proposed solution uses the unicast mode to forward packets between nodes that constitute the multicast path. The multicast packet sends itself to branches through the IP tunnel. The packet that arrives at the brunch is converted to a unicast packet and goes to the receivers.

Lin et al. [33] proposed the Locality-aware multicast approach (LAMA). The solution separates multicast groups by the bandwidth threshold of sources when multicast groups form. LAMA [33] is an RP based multicast Tree, which need an RP selection procedure, the proposed algorithm selects an RP by the smallest hop counts between sources and candidate RP switches.

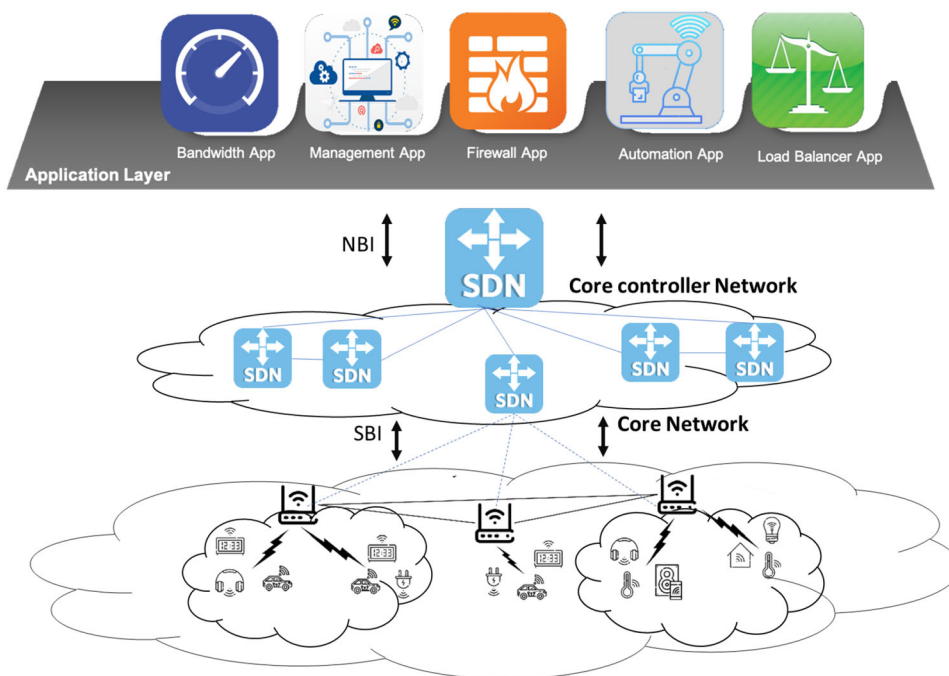
Cui et al. [34] proposed another RP-based multicast tree solution. called Data Center Network (DCN). The proposed solution selects the RP node as a set of core switches and constructs the shared trees. In this solution, the multicast packets are fragmented and sent to receivers through each shared tree.

A summary of the different multicast-based IoT architectural frameworks presented in this survey is given in Table 1.

4 MSDN-IoT: multicast group communication in IoT based on SDN

In this section, we present the details of the working methodology of our proposed MSDN-IoT architecture. In particular,

Fig. 3 MSDN-IoT network design



we will discuss the details of its design considerations, characteristics, and routing functionalities.

The design of our solution is based on multiple sides, including heterogeneity, big data, high scalability, security, and privacy, which also concern the main SDN and IoT challenges.

4.1 A multicast SDN in IoT

Our architectural proposal in spite of being SDN-controlled, the network and modules are not centralized designed. We use a hierarchical design with multiple SDN controllers to overcome the complexity of a single SDN controller in a highly scalable network, such as an IoT environment.

The proposed architecture is distributed in five levels, with each level constituting an architectural layer, with the main objective of building an architecture characterized by a hierarchical control design, as shown in Fig. 3. In this case, most of the network intelligence is deployed in the core SDN controller in the Core Network Layer (CNL), and then a small part of this intelligence is distributed in a set of cluster SDN controllers. As shown in Fig. 3, we have the following layers: (1) The Application Layer (AL), (2) The Core Network Layer (CNL) with the Core SDN Controller (C-SDN-C), (3) The Cluster Network Layer (CINL) with the Cluster SDN Control (CI-SDN-C), (4) The Device Layer (DL), and (5) The Multicast Group Members Layer (MGML).

(1) The Application Layer: This layer acts as the standard application layer in any SDN solution, the layer process as a communication interface between the Core SDN Control C-SDN-C and applications using northbound APIs. Appli-

cations in this layer can be deployed over a virtualized core network; some of these applications can be open to, or deployed by, third parties to offer Software as a Service (SaaS).

(2) The Core Network Layer (CNL): this layer acts as the kernel of the proposed architecture; it represents the intelligent part, and it contains a set of equipment that connects the clusters of the Cluster Network Layer (CINL). This layer contains a set of SDN-enabled equipment managed by the Core SDN Controller (C-SDN-C).

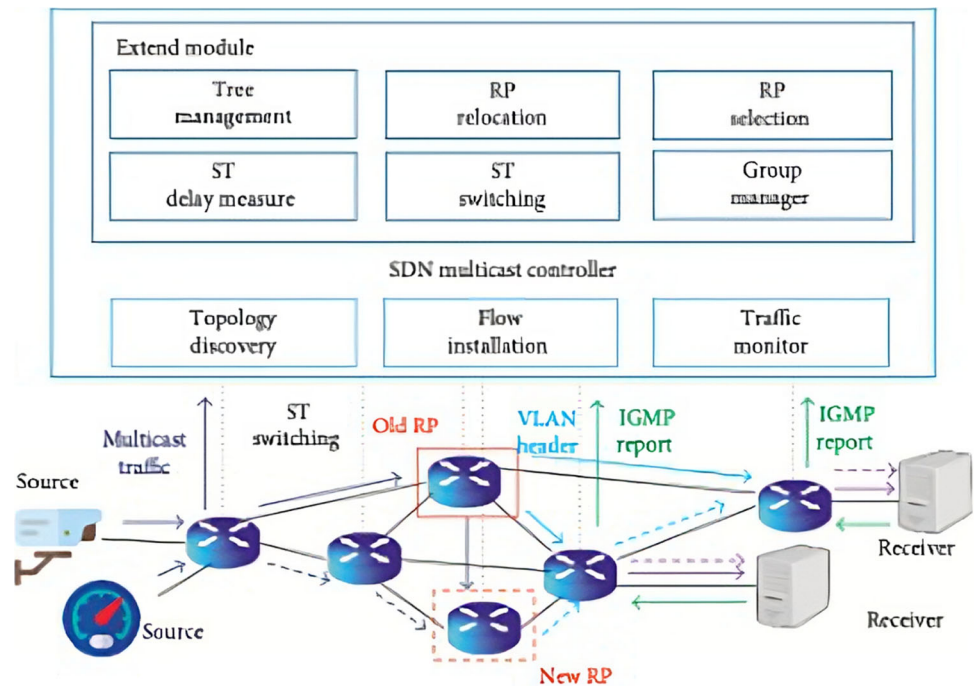
The Core SDN Controller (C-SDN-C): The Core SDN Controller is an essential component in our architecture, which provides functions of global network monitoring, network service definition, and configuration. It represents the intelligent part of the Core Network Layer (CNL), and it is responsible for managing SDN and non-SDN-based devices in the Core Network Layer (CNL).

(3) The Cluster Network Layer (CINL): This layer can be a multicast domain, i.e., a PIM domain, and all the devices included are equipped with SDN-enabled networks and are managed by the Cluster SDN Control (CI-SDN-C) and connected to the Core Network Layer (CNL).

The Cluster SDN Control (CI-SDN-C): This node constitutes the intelligent equipment in each cluster; it's an SDN controller that manages all the multicast membership messages related to each multicast domain represented by the cluster. Each cluster SDN controller is connected to and managed by the Core SDN Controller (C-SDN-C).

(4) The Device Layer: This layer is the densest, with all end devices and objects connected by different supported network access technologies, such as wired medium, wire-

Fig. 4 MSDN-IoT controller modules



less connections, Bluetooth, and others. This heterogeneous layer contains devices with diverse performances, for example, CPU processing, RAM memory, energy power, etc.

(5) The multicast group members layer: This layer consists of a set of all heterogeneous multicast group members (sources or members) consuming or forwarding multicast packets.

4.2 SDN controller modules

To complete IP multicast functionality in an IoT context, we present the design and main modules of the MSDN-IoT controller in this section. The main element of a multicast network is the multicast controller, which manages multicast groups, handles joining and leaving events, computes multicast trees, and handles joining and departing events. Figure 4 provides a high-level overview of our developed controller, which is made up of nine modules: group management, topology discovery, RP relocation, RP selection, shared-tree delay measurement, shared-tree switching, multicast tree management, flow installation, and traffic monitoring.

In the following sections, we will present some of these modules. We start with the mathematical modeling used in the multicast tree management, RP relocation, and RP selection.

4.3 Mathematical modeling

In this section, we formulate and present Mathematical modeling of the problems and define the notions that will be used

throughout the paper. A network topology mathematically can be formulated as a graph structure, $G(N, E)$, this graph is generally directed and fully connected. The graph structure $G(N, E)$ is composed by two elements N and E , E is a collection of links that connect the nodes in the finite set of nodes N .

Assume that there are $|N|$ network nodes and $|E|$ network links. Given that the graph is directed and that an edge $e(u, v)$ linking two neighboring nodes $u \in N$, and $v \in N$, will be marked, it is assumed that there is an edge $e(v, u)$ connecting v and u . Each edge is associated with many positive real values as a weight W , this weight may be either monetary cost or any measure of resource utilization, we specify two possible weights W_0 as a cost function and W_1 as a delay function: $W_0 = C(e) = C(e(u, v))$ which can represent link utilization, and $W_1 = D(e) = D(e(u, v))$ represents the delay that the packet experiences through passing that link including switching, queuing, transmission and propagation delays, therefore, the $W(v_0, v_n) = \sum W_i(v_0, v_n)$. We associate for each path $P(v_0, v_n)$ defined by the vector $(P(v_0, v_1), P(v_1, v_2), \dots, P(v_{n-1}, v_n))$, in the network a weight which is the sum of all edges weight, for example,

$$C(P(v_0, v_n)) = \sum_{n-1}^0 C(e(v_i, v_{i+1})), \quad (1)$$

and

$$D(P(v_0, v_n)) = \sum_{n-1}^0 D(e(v_i, v_{i+1})). \quad (2)$$

Definition 1 Multicast Tree (MT), Given a set of source node $S \subseteq V$ and a set of destination nodes $D \subseteq V$, multicast tree $MT(S, C, D)$ is a subgraph of G whose root node is selected core node C and whose leaf nodes form the subsets of D .

Definition 2 Multicast Trees (MTs). Given a set of Multicast Trees (MTs) in a specific network topology and a set of destination nodes $D \subseteq V$, multicast tree MT is a minimal subgraph of G covering all multicast trees.

Definition 3 Minimum Steiner Tree (MST) [35]. An MST is a multicast tree in MT whose weight is minimum among all multicast trees $MT(S, C, D)$.

An SPT or a Shared-T can be calculated in a polynomial time, while many studies demonstrate that MST cannot be computed in polynomial time [31, 35]. This paper mostly focuses on finding an optimal multicast tree that covers all multicast trees in the network topology with multiple constraints.

In Shared-T Tree protocols, all source nodes must forward in unicast mode the multicast data to selected core node C , after that, the core node transmits this multicast data to all recipients via the Shared Tree. The following equation, which the F function can be either a cost function or a delay function, is used to represent the transmission of multicast data via these two components divided by the core node C :

$$F(C, MTs) = \begin{cases} \min(MTs(S, C, D)) \\ \text{delay} < \alpha \\ \text{delay_variation} < \beta \end{cases}, \quad (3)$$

where α and β are two positive weights values associated respectively to the end-to-end delay and delay variation thresholds, this two value can be set initially by the administrator, and can be dynamically adjusted, by the SDN controllers, based on application exigence.

4.4 Clusters and areas presentation

Most IoT applications include a large number of distributed sensors that are interesting in a specific geographical area. Our approach is logically hierarchical organization into clusters and areas, with multiple SDN controllers. This conception, with a multi-controller design compared with a single-controller approach, can effectively improve the performance of multicast communication in SDN and IoT networks.

An area represents an autonomous system, which can be a corporate network or a multicast domain (e.g., the PIM domain) with its own multicast group address. An area can also be any wireless LAN with an access router and many access points. This separation allows you to manage wired

and wireless areas separately. Areas belonging to the same domain or corporate network are organized into clusters, where each cluster is controlled by a Cluster SDN Control CI-SDN-C.

4.5 RP management module

In this present work, we will focus on the RP Selection sub-module as a part of the global RP Management module. which uses a search algorithm that determines the RP location when any multicast group member's membership changes and based on other SDN modules' events.

Algorithm 1 displays the pseudocode for the RP selection algorithm.

Algorithm 1 RP Selection Pseudo code

```

maxItWithoutImprovement ← 0
while events do
  totalIt ← 0
  initialSolution ← random()
  Solutionsheking
  Solutionlocal
  while  $i < \text{maxItWithoutImprovement}$  &  $\text{totalIt} < \text{maxit}$ 
  do
     $k \leftarrow 0$ 
    while  $\text{currentit} < \text{maxit}$  &  $k < \text{kmax}$  do
      if  $\text{totalIt} > \text{maxit}$  then
        Break
      end if getNK(s)
      if ( $\text{opt\_Function}(\text{Solution\_local}) > \text{opt\_Functions}(s)$ ) then
         $k \leftarrow k + 1$ 
         $s \leftarrow \text{Solutionlocal}$ 
         $k \leftarrow 0$ 
         $\text{totalIt} \leftarrow \text{totalIt} + 1$ 
         $\text{currentit} \leftarrow \text{currentit} + 1$ 
        if ( $\text{lastSCost} > \text{optFunction}(s)$ ) then
           $i \leftarrow 0$ 
        else if
           $i \leftarrow i + 1$  then
        end if
      end if
    end while
  end while
end while
return  $s$ 

```

4.6 Multicast tree computing module

Most traditional multicast routing techniques and protocols implement routing procedures principally using a cost function based on the minimum-hop constraint or the shortest path tree (SPT) as the measurement.

We based our proposal architecture on a multicast routing protocol based on Shared Multicast Tree SMT (or Core-Based Tree CBT), such as the Protocol Independent Multicast-Sparse Mode Protocol (PIM-SM) [36] or the Core-

Table 2 Summary of simulation parameters and values

Parameters	Values
Simulator	Mininet emulator [38, 39]
Simulation time	10min
SDN Controller	NOX [40]
Simulation Dimension	100–100 to 1000–1000
Machine performance	Xeon processors 2.00GHz; 8-CPU 4 MB; 32 GB of RAM
Network size	160 nodes
Transport Layer	UDP
Sources set size	5%
Routing Protocol	OSPF
receivers set size	5%–40%
Application layer	CBR
Traffic rate	0.5–0.7 pk

Based Tree (CBT) [37] protocol. Our choice is motivated by the scalability and dynamism of SMT multicast trees; globally, SMT-based protocols are designed for the larger and sparser groups encountered on the Internet. SMT-based protocols develop a routing table without relying on unicast protocols or infrastructure topology. Instead, they use soft state mechanisms to adapt to the underlying topology-gathering protocol.

After selecting the RP by calling the RP Selection submodule, the Multicast Tree Computing Module (MTCM) tries to compute the optimal shared multicast tree by combining the shortest path between the source and the RP and the shortest path between the RP and all multicast receivers. Initially, the density of the multicast routing topology is widely sparse; each group member is involved in a multicast tree with one element.

In our proposed scheme, we select the best first multicast path, and we distinguish two selection approaches to choose an optimal multicast path as an initial solution.

After a cluster SDN controller obtains all group members' location information, it computes the distance to each group member and the density of group members.

4.7 Multicast membership management module

Any multicast protocol, as proposed by Deering [2], uses two important messages to build and manage group membership. These messages are known as "multicast join" and "member leave."

When the cluster controller receives a joining group event, it updates the group membership. If a multicast member wishes to leave a specific multicast group, it sends a multicast leave message, and the SDN controller deletes the corre-

sponding switch flow entry and updates memberships in the group management database.

5 Testbed and simulations

5.1 Testbed implementation

We implement a testbed environment with a distributed SDN environment based in Mininet emulator [38, 39] using a quad-core machine with Xeon processors at 2.00 GHz, an 8-core 4 MB, and 32 GB of memory to demonstrate the efficiency and good performances of our proposed architectural solution MSND-IoT. We design all SDN controllers based on open-source software, NOX [40]. In our topology, we distinguish between two types of networks: wired networks and wireless networks.

In the case of the wired network, we use a customized network topology generator [41] based on the BRITE [42] module with the Waxman model [43] as the graph pattern. Our performance studies were performed for each simulation on a set of 100 random network topologies. The values of $\alpha = 0.1$ and $\beta = 0.3$ were adopted to generate network topologies with nodes' average degree between 3 and 4 in the mathematical model of Waxman. The edge probability is given by $P(u, v) = ke^{-(u; v)/LD}$, where $d(u, v)$ is the distance between nodes u and v , D is the maximum distance between two nodes, and K and L are parameters in the range $[0; 1]$. It is noted that increasing L increases the number of connections between far-off nodes, and increasing K increases the degree of each node.

In the case of the wireless network We use the GENSEN [44], which is a wireless network generator that is more realistic, to determine pseudo-randomly all network elements, such as node locations, battery-powered devices, communicating nodes, etc.

Table 2 provides the details of various parameters along with their values that are used to configure the testbed.

5.2 Simulation results

To demonstrate the performance of our architectural solution (MSDN-IoT), we compare it with the following algorithms: branching shortest path tree (EBSPT) [32], locality-aware multicast approach (LAMA) [33], data center network [34], and a simulation with a native PIM-SM protocol [36] deployment.

The main objective of any multicast solution is to reduce end-to-end delay and end-to-end delay variation during any multicast session; therefore, we start this simulation results study by comparing these two metrics, and then we will review other metrics.

Fig. 5 Network size vs Average end-to-end delay

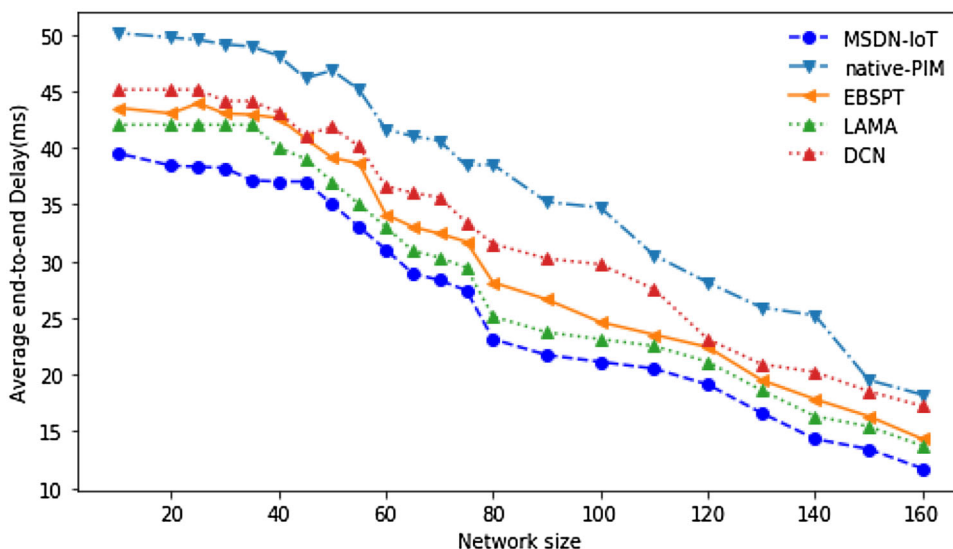
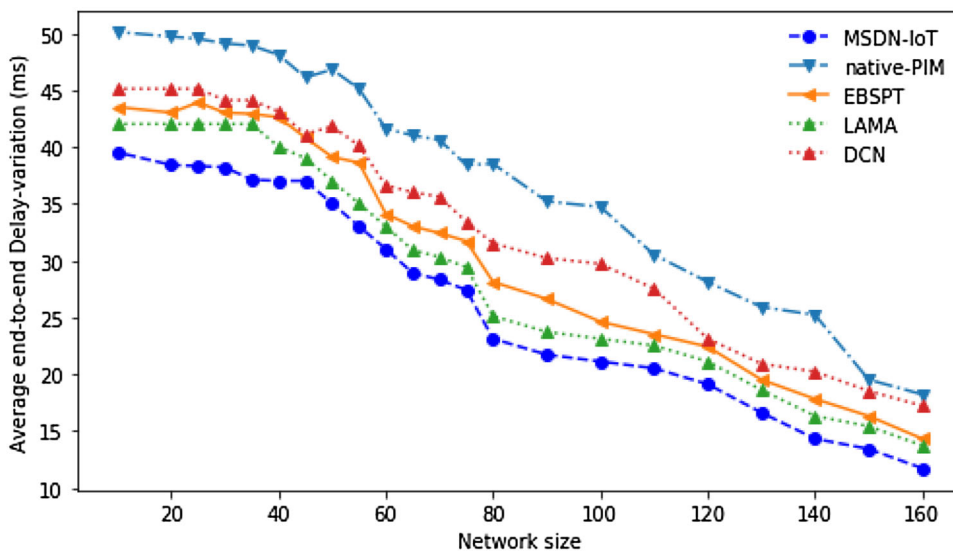


Fig. 6 Network size vs Average end-to-end delay-variation]



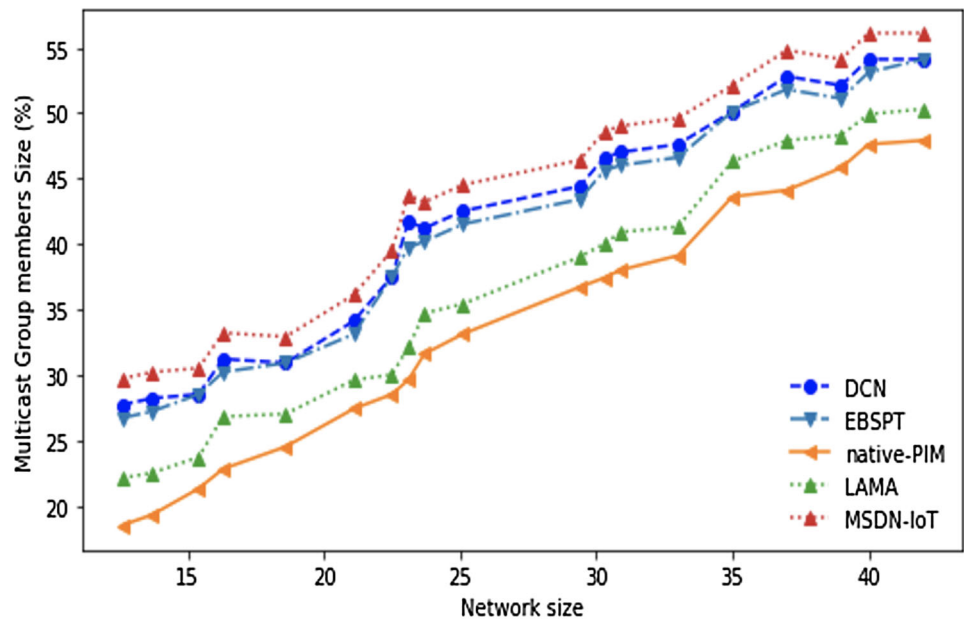
End-to-end Delay is defined as the time spent to forward multicast data from the farthest source node to the selected RP to the farthest receiver node in the multicast group tree. In this simulation, we use a network topology with multicast group members that are 10% of the total network nodes, and we vary the number of network topology nodes from 10 to 160 nodes. Figure 5 shows the simulation results of multicast end-to-end delay versus network topology size in terms of the number of nodes.

To evaluate real-time application support by the multicast solution and if the used multicast tree algorithm selects and builds an optimal multicast tree, we use the end-to-end delay variation metrics because the multicast nature requires that all multicast members receive the information at the same time. End-to-end delay variation is the difference between the first time a multicast packet is received by a receiver in a multicast group and the last time the same multicast packet

is received by another receiver. Also, in this simulation, we use a network topology with a multicast group member’s size equal to 10% of the overall network nodes, and we vary the number of network topology nodes in the range of 10–160 nodes. Figure 6 shows the simulation results of multicast end-to-end delay variation versus network topology size in terms of the number of nodes.

In this section, we evaluate also the scalability of the proposed system in terms of supported multicast group number, supported multicast group number throughout the network topology, and supported multicast group member size. Figure 7 shows an examination of scalability for each solution with regard to the supported multicast group members in the multicast session. We see that in this network scenario, the percentage of nodes that will participate in the multicast group session varies from 20% to 80%.

Fig. 7 Network size vs % of Multicast Group Members



5.3 Discussion

As we've already said in this paper, our architectural proposal's main goal is to reduce end-to-end delay and end-to-end delay-variation metrics, because To compute the multicast tree, we use Eq. 3 inside algorithm 1. Figures 5, 6, and 7 demonstrate how network size influences network end-to-end delay, delay-variation and scalability metrics for compared solutions. As it can be seen, our proposed MSDN-IoT has the best performance, with the lowest end-to-end delay and end-to-end delay variation, due to the fact that the combination of multicast tree computing and the RP management module uses the optimal function to select an optimal multicast tree. Figure 6 shows that our solution supports more multicast group members compared to other solutions when we increase the network size.

6 Conclusion

We propose MSDN-IoT, a novel architectural solution for integrating multicast communication mode on SDN-based IoT environments, in this paper. The proposed architecture network and modules are not centralized; we use a hierarchical design with multiple SDN controllers to overcome the complexity of a single SDN controller in a highly scalable network, such as an IoT environment. The proposed SDN solution is based on a flexible set of SDN controller modules, including the Multicast Tree Computing module, which is responsible for building an optimal multicast tree; the RP Management Module, which is responsible for managing RP events; and the Group Management Module, which

is responsible for handling all multicast group membership events. Our results show the effectiveness of our protocol over state-of-the-art protocols in terms of end-to-end delay, end-to-end delay variation, scalability, and other metrics. As future work, we are currently working on an enhanced version of our architectural solution to support node mobility.

Author Contributions All authors of this work participate equally to produce the paper.

Funding Not applicable.

Declarations

Conflict of interest Not applicable.

Ethical approval Not applicable.

References

- Salma B, Youssef B, Abderrahim H (2020) Software defined networking based for improved wireless sensor network vol. 92. https://doi.org/10.1007/978-3-030-33103-0_25
- Deering SE (1989) Host extensions for IP multicasting. Request for Comments RFC 1112, Internet Engineering Task Force . <https://doi.org/10.17487/RFC1112>. Num Pages: 17. <https://datatracker.ietf.org/doc/rfc1112>. Accessed 13 Nov 2022
- Khan FI, Hameed S (2016) Software defined security service provisioning framework for Internet of Things. Int J Adv Comput Sci Appl (IJACSA). <https://doi.org/10.14569/IJACSA.2016.071254>. (Number: 12 Publisher: The Science and Information (SAI) Organization Limited. Accessed 2022-11-13)
- Khan S, Ali M, Sher N, Asim Y, Naeem W, Kamran M (2016) Software-defined networks (SDNs) and Internet of Things (IoTs): a qualitative prediction for 2020. Int J Adv Com-

- put Sci Appl 7(11):10. <https://doi.org/10.14569/IJACSA.2016.071151>. (Accessed 2022-11-13)
5. Chen J, Zheng X, Rong C (2015) Survey on software-defined networking. In: Qiang W, Zheng X, Hsu C-H (eds) Cloud computing and big data. Lecture Notes in Computer Science. Springer, Cham, pp 115–124. https://doi.org/10.1007/978-3-319-28430-9_9
 6. Khanh QV, Hoai NV, Manh LD, Le AN, Jeon G (2022) Wireless communication technologies for IoT in 5G: vision, applications, and challenges. *Wirel Commun Mob Comput*. <https://doi.org/10.1155/2022/3229294>. (Publisher: Hindawi)
 7. Quy VK, Nam VH, Linh DM, Ngoc LA (2022) Routing algorithms for MANET-IoT networks: a comprehensive survey. *Wirel Pers Commun* 125(4):3501–3525. <https://doi.org/10.1007/s11277-022-09722-x>
 8. Salma B, Youssef B, Abderrahim H (2020) Software defined networking based for improved wireless sensor network. In: Ezziyiani M (ed) *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019)*. Springer, Cham, pp 246–258
 9. Jararweh Y, Al-Ayyoub M, Darabseh A, Benkhelifa E, Vouk M, Rindos A (2015) SDIoT: a software defined based internet of things framework. *J Ambient Intell Humaniz Comput* 6(4):453–461. <https://doi.org/10.1007/s12652-015-0290-y>. (Accessed 2022-11-25)
 10. Gu W, Zhang X, Gong B, Wang L (2015) A survey of multicast in software-defined networking. Atlantis Press. ISSN: 2352-5401. <https://doi.org/10.2991/icimm-15.2015.198>. <https://www.atlantis-press.com/proceedings/icimm-15/25690> Accessed 13 Nov 2022
 11. Morreale PA, Anderson JM (2014) *Software defined networking: design and deployment*. CRC Press, Boca Raton. <https://doi.org/10.1201/b17708>
 12. Mohammadi R, Javidan R (2022) Efsute: a novel efficient and survivable traffic engineering for software defined networks. *J Reliable Intell Environ* 8:1–14. <https://doi.org/10.1007/s40860-021-00139-0>
 13. Ahmad I, Namal S, Ylianttila M, Gurtov A (2015) Security in software defined networks: a survey. *IEEE Commun Surv Tutor* 17(4):2317–2346. <https://doi.org/10.1109/COMST.2015.2474118>. (Conference Name: **IEEE Communications Surveys & Tutorials**)
 14. Hu F, Hao Q, Bao K (2014) A survey on software-defined network and OpenFlow: from concept to implementation. *IEEE Commun Surv Tutor* 16(4):2181–2206. <https://doi.org/10.1109/COMST.2014.2326417>. (Conference Name: **IEEE Communications Surveys & Tutorials**)
 15. Rehmani MH, Davy A, Jennings B, Assi C (2019) Software defined networks-based smart grid communication: a comprehensive survey. *IEEE Commun Surv Tutor* 21(3):2637–2670. <https://doi.org/10.1109/COMST.2019.2908266>. (Conference Name: **IEEE Communications Surveys & Tutorials**)
 16. Benzekki K, El Fergougui A, Elbelrhiti Elalaoui A (2016) Software-defined networking (SDN): a survey. *Secur Commun Netw* 9(18): 5803–5833. <https://doi.org/10.1002/sec.1737>. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1737>. Accessed 13 Nov 2022
 17. Bizanis N, Kuipers FA (2016) SDN and virtualization solutions for the internet of things: a survey. *IEEE Access* 4:5591–5606. <https://doi.org/10.1109/ACCESS.2016.2607786>. (Conference Name: **IEEE Access**)
 18. Baddi Y, Ech-Chrif El Kettani MD (2012) VNS-RP algorithm for RP selection in multicast routing protocol PIM-SM. In: *Proceedings of 2012 International Conference on multimedia computing and systems, ICMCS 2012*, pp 595–600. <https://doi.org/10.1109/ICMCS.2012.6320123>
 19. Baddi Y, Kettani MDE (2012) VNS-RP algorithm for RP selection in multicast routing protocol PIM-SM. *Tangier*, pp 595–600
 20. Baddi Y, Ech-Chrif El Kettani M.D (2012) VNS-RP algorithm for RP selection in multicast routing protocol PIM-SM. In: *Proceedings of 2012 International Conference on multimedia computing and systems, ICMCS 2012*, pp 595–600. <https://doi.org/10.1109/ICMCS.2012.6320123>
 21. Baddi Y, El Kettani MDEC (2012) VND-CS: a variable neighborhood descent algorithm for core selection problem in Multicast Routing Protocol. In: Benlamri R (ed) *Networked digital technologies*. NDT 2012. Communications in computer and information science, vol 293. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30507-8_37
 22. Baddi Y, El Kettani MDE-C (2013) Parallel GRASP algorithm with delay and delay variation for rendezvous point selection in PIM-SM multicast routing. *J Theoret Appl Inf Technol* 57(2):235–243
 23. Baddi Y, Kettani MDE (2012) GRAS-RP: greedy randomized adaptive search algorithm for RP selection in PIM-SM multicast routing. In: *Seventh International Conference on intelligent systems : theories and applications (SITA12)*, Mohammedia, Morocco
 24. Su L, Ding B, Yang Y, Abdelzaher T.F, Cao G, Hou JC (2009) oCast: optimal multicast routing protocol for wireless sensor networks. In: *2009 17th IEEE International Conference on network protocols*, pp. 151–160. <https://doi.org/10.1109/ICNP.2009.5339689>. (ISSN: **1092-1648**)
 25. Xie L, Jia X, Zhou K (2012) QoS multicast routing in cognitive radio ad hoc networks. *Int J Commun Syst* 25(1):30–46. <https://doi.org/10.1002/dac.1285>. (Accessed 2022-11-13)
 26. Park H, Lee J, Park S, Oh S, Kim S-H (2011) Multicast protocol for real-time data dissemination in wireless sensor networks. *IEEE Commun Lett* 15:1291–1293. <https://doi.org/10.1109/LCOMM.2011.102611.110995>
 27. Sanchez JA, Marin-Perez R, Ruiz PM (2012) Beacon-less geographic multicast routing in a real-world wireless sensor network testbed. *Wirel Netw* 18(5):565–578. <https://doi.org/10.1007/s11276-012-0419-2>. (Accessed 2022-11-13)
 28. Santamaria AF, Sottile C, Fazio P (2015) PAMTree: partitioned multicast tree protocol for efficient data dissemination in a VANET environment. *Int J Distrib Sens Netw* 11(5):431492. <https://doi.org/10.1155/2015/431492>. (Publisher: **SAGE Publications**. Accessed 2022-11-13)
 29. Pan M-S, Yang S-W (2017) A lightweight and distributed geographic multicast routing protocol for IoT applications. *Comput Netw* 112:95–107. <https://doi.org/10.1016/j.comnet.2016.11.006>. (Accessed 2022-11-13)
 30. Conti M, Kaliyar P, Lal C (2017) Remi: a reliable and secure multicast routing protocol for iot networks. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security. ARES '17*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3098954.3106070>
 31. Huang J, Duan Q, Zhao Y, Zheng Z, Wang W (2017) Multicast routing for multimedia communications in the Internet of Things. *IEEE Internet of Things J* 4(1):215–224. <https://doi.org/10.1109/JIOT.2016.2642643>. (Conference Name: **IEEE Internet of Things Journal**)
 32. Humernbrum T, Hagedorn B, Gorlatch S (2016) Towards efficient multicast communication in software-defined networks. In: *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp 106–113. <https://doi.org/10.1109/ICDCSW.2016.15>. (ISSN: **2332-5666**)
 33. Lin Y-D, Lai Y-C, Teng H-Y, Liao C-C, Kao Y-C (2017) Scalable multicasting with multiple shared trees in software defined networking. *J Netw Comput Appl* 78:125–133. <https://doi.org/10.1016/j.jnca.2016.11.014>. (Accessed 2022-11-13)
 34. Cui W, Qian C (2014) Dual-structure data center multicast using software defined networking. <https://doi.org/10.48550/arXiv.1403.8065>. <http://arxiv.org/abs/1403.8065>, [arXiv:1403.8065](https://arxiv.org/abs/1403.8065) [cs]. Accessed 13 Nov 2022

35. Garey MR, Johnson DS (1990) Computers and intractability. In: Klee V (ed) A guide to the theory of NP-completeness. A series of books in the mathematical sciences. W. H. Freeman and Co., USA
36. Fenner B, Handley M, Holbrook H, Kouvelas I (2006) Protocol independent multicast-sparse mode (PIM-SM): Protocol Specification (Revised). Request for Comments, vol. 4601. IETF, ???
Published: RFC 4601 (Proposed Standard) Updated by RFC 5059. <http://www.ietf.org/rfc/rfc4601.txt>. Accessed 20 Nov 2022
37. Ballardie T, Francis P, Crowcroft J (1993) Core based trees (CBT). ACM SIGCOMM Comput Commun Rev 23(4):85–95. <https://doi.org/10.1145/167954.166246>. (Accessed 2022-11-15)
38. de Oliveira RLS, Schweitzer CM, Shinoda AA, Prete LR (2014) Using Mininet for emulation and prototyping Software-Defined Networks. In: 2014 IEEE Colombian Conference on communications and computing (COLCOM), pp. 1–6. <https://doi.org/10.1109/ColComCon.2014.6860404>
39. Mininet: an instant virtual network on your laptop (or Other PC)-Mininet. <http://mininet.org/>. Accessed 13 Nov 2022
40. Gude N, Koponen T, Pettit J, Pfaff B, Casado M, McKeown N, Shenker S (2008) NOX: towards an operating system for networks. ACM SIGCOMM Comput Commun Rev 38(3):105–110. <https://doi.org/10.1145/1384609.1384625>. (Accessed 2022-11-13)
41. Baddi Y, Sebbar A, Zkik K, Boulmalf M, Ech-Cherif El Kettani MD (2020) MSND: multicast software defined network based solution to multicast tree construction. In: Habachi O, Meghdadi V, Sabir E, Cances J-P (eds) Ubiquitous networking. Lecture Notes in Computer Science. Springer, Cham, pp 245–256. https://doi.org/10.1007/978-3-030-58008-7_20
42. Heckmann O, Piringer M, Schmitt J, Steinmetz R (2003) On realistic network topologies for simulation. In: Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research. MoMeTools '03, pp. 28–32. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/944773.944779>. Accessed 13 Nov 2022
43. Waxman BM (1988) Routing of multipoint connections. IEEE J Sel Areas Commun 6(9):1617–1622. <https://doi.org/10.1109/49.12889>. (Conference Name: IEEE Journal on Selected Areas in Communications)
44. Camilo T, Silva JS, Rodrigues A, Boavida F (2007) GENSEN: a topology generator for real wireless sensor networks deployment. In: Obermaisser R, Nah Y, Puschner P, Rammig FJ (eds) Software technologies for embedded and ubiquitous systems. Lecture Notes in Computer Science. Springer, Berlin, pp 436–445. https://doi.org/10.1007/978-3-540-75664-4_46

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.