EAI Endorsed Transactions
on Wireless Spectrum

Research Article EAI.EU

Towards to Inter-domain Network Operations for Dynamic Networks with Software Defined Networking (SDN)

James Nguyen[1] and Wei Yu[2,*]

[1]US Army Communications-Electronics Research, Development, and Engineering Center (CERDEC), Aberdeen Proving Ground, MD, USA.

[2]Dept. of Computer and Information Sciences, Towson University, Towson, MD, USA.

Abstract

In this paper, we introduce an Software Defined Networking (SDN)-based approach to support the network operations of heterogeneous hierarchical multi-domain Mobile Ad hoc Networks (MANETs). Our approach offers several capabilities, including seamlessly interconnecting heterogeneous MANETs, reducing routing decision loads of gateway routers, and managing end-devices at the network edge from the Network Operation Center (NOC) with our SDN Proxy Protocol. To assess the feasibility of our proposed approach, we setup an emulation environment, and implement our designed system via the Constrained Application Protocol (CoAP), additionally integrating OLSRv2, OSPF-MDR, Babel, and others. Based on a set of designed test scenarios, we then carry out an extensive performance evaluation of our approach. Our experimental results show significant network performance improvements at the gateway routers. The SDN Proxy Protocol also outperforms the traditional CoAP-to-CoAP communication when network instability exists.

Received on 31 May 2018; accepted on 02 December 2018; published on 07 February 2019

Keywords: SDN, MANET, IoT, Multi-Domain Heterogeneous MANETs, CoAP, Dynamic Network Operations, Protocol Design, OLSR, Babel, OSPF-MDR, Quagga

doi:10.4108/eai.12-12-2017.156588

1. Introduction

Dynamic networks such as MANETs have been applied in a number of areas, including public safety and disaster recovery, law enforcement, military, emergency search and rescue, and others [2, 19, 28, 43]. MANETs have come to be considered critically important because of their beneficial characteristics including: (i) quick network deployment that does not require a network infrastructure backbone, (ii) self-configuration, (iii) self-organization in dynamic topologies, (iv) self-healing in establishing a resilient mesh network, and (v) transitivity through the use of multi-hop protocols to communicate between nodes [41, 44]. Nonetheless, as the nodes in a MANET may move around in an arbitrary fashion, it is not possible to predict the topology of these nodes or how they would form. As nodes can move in and out of the mesh network, the MANET often experiences network disruption, intermittent service, and low-bandwidth. This also requires the MANET nodes to update and share topology changes regularly to maintain the network connectivity.

A typical deployment of multi-domain heterogeneous MANETs is a natural solution for when the multiple organizations conduct joint forces and respond to the same event. Typically, each organization has a totally different communication system distinct from the others. Thus, the inter-connection of these communication systems is often carried out manually, which can be time consuming and counterproductive in a scenario where the joint organizations are required to react quickly to a disaster. It is also potentially error prone, possibly inducing routing loop problems, incorrect configurations, and others. The key solution, then, must quickly and automatically inter-connect these networks so that the joined forces can complete their mission. To date, however, there is little research into how to interconnect these MANETs. Although the Internet Engineering Task Force (IETF) has actively been developing

*Corresponding author. Email: wyu@towson.edu

MANET protocols, these efforts have been focused on a single domain MANET with a limited number of mobile nodes. What's more, there is a need to address how to intercontinentals well as the more challenging issues of multi-domain heterogeneous MANETs and hierarchical MANETs, as seen in tactical networks.

Generally speaking, the concept of SDN is the separation of the control plan and the data plan on the network node in order to have more control of how to drive the network node to perform specific tasks via in-house developed applications. The concept offers the potential to increase the efficiency of network management, including network deployment, network configuration management, network monitoring, and others. The deployment of networks with SDN also helps to reduce the costs of deployment, integration, troubleshooting, and evaluation of the networks, as the control plane solutions drive the network virtual functions [22].

In this paper, we propose an SDN-based approach to support multi-domain heterogeneous MANET operations using the emerging Constrained Application Protocol (CoAP) (RFC 7252) as the Northbound and Southbound APIs [37]. Our approach focuses on a three-tier hierarchical network architecture: Upper-Tier Network Level, Mid-Tier Network Level, and Lower-Tier Network Level, to seamlessly interconnect these heterogenous MANETs that are deployed with different ad hoc routing protocols (Optimized Link State Routing Protocol version 2 (OLSRv2), Babel, and Open Shortest Path First MANET Designated Routers (OSPF-MDR)). Our proposed approach is to decentralize the critical routing decisions to the Mid-Tier Network Level. In this approach, the overall network can still function when the centralized network backbone is not available. In addition, global network connectivity can be maintained by merging and splitting networks when the nodes are mobile.

We also design the basic deployment of the MANET and propose two protocols. One is the discovery protocol to automatically discover and migrate the lower-tier networks to the rest of the MANETs. The other is a proxy protocol to offload the network operation task to the SDN Controller. Our SDN-based approach is implemented based on the Eclipse Foundation's Californium CoAP [21]. We also implement an API to Quagga's Zebra to automatically allow the SDN Controller to redistribute network routes to OLSRv2 and OSPF-MDR protocols, as well as an API to access the Linux Kernel routing table. An extensive performance evaluation is conducted in a Common Open Research Emulation (CORE) environment of 48 nodes [3] with respect to overhead, packet loss ratio, and Round-Trip Time (RTT).

This paper is an extension of our prior work in [29]. Substantial new materials have been added in this journal extended version, including additional deployments, new SDN proxy protocol, analysis, and the extensions of SDN-based network operations and others. The remainder of this paper is organized as follows. In Section 2, we give an overview of SDN, CoAP, the relevant existing routing protocols (e.g., OLSRv2, Babel, OSPF-MDR) and the Quagga Routing Software Suite. In Section 3, we introduce the architecture of the multi-domain heterogenous MANETs. In Section 4, we present the implementation of the SDN Controller and the SDN Agents, and introduce a set of scenarios of multi-domain heterogeneous MANETs that can be used in real-world practice. In Section 5, we demonstrate and evaluate the feasibility of our proposed approach. In Section 6, we present the extension of architecture of the SDN-based network operations and implementations, and discuss some future research directions. In Section 7, we conduct a literature review. In Section 8, we summarize the paper.

2. Background

In this section, we present the basic concept of SDN, CoAP, and the existing routing protocols (e.g., OLSRv2, OSPF-MDR, Babel) that are used for the network in our study.

2.1. Software Defined Networking (SDN)

Generally speaking, aiming to provide flexibility to the design of network architectures, the objective of SDN is to provide control to drive the networks directly from applications by decoupling the control plane from the data traffic processing and forwarding plane [22]. More importantly, SDN provides mechanisms that can simplify the network architecture and network management through the automation of operational tasks and reduce costs of procurement.

The concept of SDN can be modeled as having the controlling processing software function (i.e., SDN Controller) close to the service management core. The model consists of the three basic layers: (i) Infrastructure Layer, (ii) Control Layer, and (iii) Application Layer. As shown in Figure 1, the Application Layer has an abstract view of the network, and provides business requirements to the SDN Controller Layer. The Control Plane provides the abstract view of the network to the Application Layer for making the business decisions, and translates the business requirements into more detail-oriented requirements for the infrastructure to comply with and execute as requests. The Infrastructure Layer, which consists of logical network devices, forwards or processes the data, or both. The Infrastructure Layer may also represent a set of physical devices. The communications between layers are denoted as Northbound application program interface (API) and

Southbound API. The Northbound API is the communication between the Application Layer and the Control Layer, while the Southbound API is the communication between the Control Layer and the Infrastructure Layer [22]. There are a variety of protocols that can be used for the Northbound and Southbound API, including OpenFlow, NETCONF, RESTCONF, CoAP, and others [22].

2.2. Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP), documented in RFC 7252 developed by the IETF Constrained RESTful Environments (core) Working Group (WG) to support the Internet of Things (IoT), was designed to simplify the web interface, portability, modifiability, visibility, and reliability [13]. CoAP adopts both HTTP and REST (Representational State Transfer) functionality of GET, PUT, POST, and DELETE methods, but its message mechanism is designed on top of User Datagram Protocol) instead of Transmission Control Protocol (TCP). It is worth noting that the purpose of using UDP is to reduce overhead and improve energy consumption.

2.3. Optimized Link State Routing Protocol version 2 (OLSRv2)

The Optimized Link State Routing Protocol version 2 (OLSRv2), which is documented in RFC 7181 and developed by the IETF, is a proactive and routing table-driven routing protocol designed especially for MANET. The protocol regularly shares topology changes (TCs) by using reduced flooding mechanisms to control the traffic throughout the network. The traffic control is carried out by selecting a set of nodes to forward the TC to the rest of the network. This mechanism is called Multipoint Relay (MPR) [9].

2.4. OSPF MANET Designated Routers (OSPF-MDR)

The Open Shortest Path First MANET Designated Routers, shortened to OSPF-MDR, is an extension of the OSPFv3 routing protocol designed to support MANETs. The routing protocol maintains the entire network topology by regularly sending out discovery packets to identify its neighbors and establish adjacent links. Every OSPF-MDR node keeps the information about its discovered neighbors in a link-state database. Similar to OLSRv2, OSPF-MDR also uses a flooding mechanism to reduce the transmissions of topology information by selecting a subnet of OSPF nodes to flood the TCs. This set of nodes forms a Connected Dominating Set (CDS). The ultimate goal of CDS is to reduce overhead and improve the reliable link adjacency [32].

2.5. Babel Routing Protocol

The Babel routing protocol is a distance-vector routing protocol designed for both wired and wireless networks. Babel, which is based on Destination-Sequenced Distance Vector (DSDV) Routing, Ad hoc On-Demand Distance Vector (AODV), and Cisco Systems' Enhanced Interior Gateway Routing Protocol (EIGRP), offers loop-free routing functionality by using the Bellman-Ford algorithm. The Babel protocol is considered robust and flexible for prefix-based wired networks and wireless mesh networks. Note that the Babel routing protocol is published as an experimental standard RFC 6126 by the IETF [8].

2.6. Quagga Routing Software Suite

Quagga Routing Software Suite was originated by Kunihiro Ishiguro under the name GNU Zebra, and provided TCP/IP-based routing services. Quagga, which is an open source routing package, has been extended to provide additional routing protocols, including OSPF, Routing Information Protocol (RIP), Border Gateway Protocol (BGP), Babel, and Intermediate System to Intermediate System (IS-IS) that communicates with the Linux Operating System via Zebra API services [18]. The Quagga Routing Software Suite has continuingly been contributed by the research community. The integration of OSPF-MDR with Quagga was accomplished by the US Naval Research Laboratory (NRL) [32, 33].

3. SDN Network Operational Architectural Model

In this section, we present the SDN-based Network Operational Model for multi-domain heterogeneous MANETs. In our proposed approach, we decentralize the SDN controllers to the Mid-Tier Network Layer, in which auto-configuration management can be conducted. In the following, we first give an overview of hierarchical network operational model and then introduce the architecture of SDN-based hierarchical MANETs. Finally, we present SDN network model and communication protocols.

3.1. Overview of Hierarchical Network Operational Model

Our network model is based on typical hierarchical MANETs, and is built on two basic requirements: (i) The current network model is an IPv4-based model, and (ii) Every MANET node is deployed with a unique IP address. As illustrated in Figure 2, the network model consists of three levels: Upper-Tier Network Level, Mid-Tier Network Level, and Lower-Tier Network Level. Every level consists of one or more network domains.

- ***Upper-Tier Network Level***: This is where the Network Operation Center (NOC) typically is

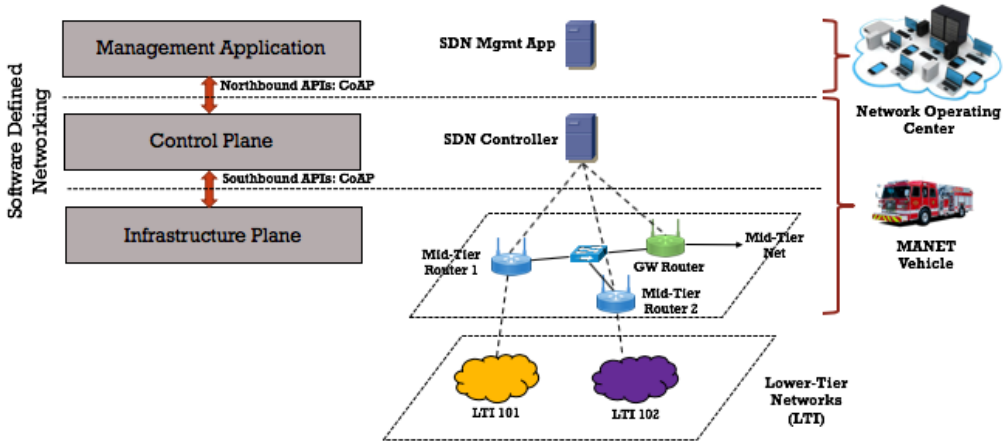


Figure 1. SDN Architecture of Heterogeneous MANET

located. Network Operations (NetOps) tasks are primarily executed at this level or carried out by a ranking Special Agent in charge (e.g., Network Administrative Supervisor). The Network Operations servers, hosts, tools, and others, which are often deployed in this layer, connect to the Mid-Tier networks through a gateway.

- ***Mid-Tier Network Level***: In this level, there is one or more Mid-Tier networks (e.g., MANET 1, MANET 2, MANET 3, and MANET 4 as shown in Figure 2). In our model, a Special Field Agent (e.g., Network Operator) of a Mid-Tier Network, who resides on the MANET Vehicle, coordinates communications between a ranking Special Agent in charge of the Upper-Tier Network and the Edge Units. The MANET Vehicle carries multiple routers and hosts to provide mission communications. The MANET Vehicle also serves at the gateway between its Lower-Tier networks, and furthermore interconnects to other Mid-Tier networks. A Mid-Tier network is configured connecting to its Upper-Tier Network through its Gateway Router (GW Router). Notice that the routing protocol on the GW Router that we use is the Babel routing protocol due to its support of prefix-based wired networks and wireless mesh networks. In addition to the GW Router, the MANET Vehicle also has one or more MANET routers (Mid-Tier Routers), which are used to connect to a Lower-Tier network. In our hierarchical network model, we use various routing protocols (e.g., OLSRv2, OSPF-MDR, Babel) to build heterogeneous networks. The GW Router and the Mid-Tier Routers are connected through a network switch that defines *local* Vehicle's Local Area Network (LAN) on the MANET Vehicle.
- ***Lower-Tier Network Level***: This is basically the network of Edge Units, in which every Edge Unit has a radio to communicate with each other. The radio, which is a single-channel radio, is a single network interface MANET router that runs one of the routing protocols: OLSRv2, OSPF-MDR, and Babel.

3.2. Architecture of SDN-based Hierarchical MANETs

Figure 1 illustrates our SDN Architecture in supporting multi-domain heterogeneous MANETs. The architecture consists of the three levels outlined in Section 3.1. The CoAP protocol is used in this model for Northbound APIs and Southbound APIs. As shown in Figure 1, the SND architecture consists of the following three layers.

- ***Application Layer***: The Application Layer is located at the Upper-Tier Network Level. The application in our model is simply a CoAP client used to perform basic monitoring management tasks on one of the Edge Units, such as obtaining the number of inbound packets or octets, obtaining the number of outbound packets or octets, etc.
- ***Control Layer***: The SDN Controller that we propose is deployed at the MANET Vehicle as shown in Figure 1 connected to the local Vehicle's LAN. The designed approach is to decentralize the control management to the lower level of the networks for the following two reasons. First, it removes workload from the centralized management systems. Second, it allows the lower-level networks to function independently without the need of the centralized management systems, allowing communications to remain intact in the event that the central unit is down or out of

communication range. The SDN Controllers also remove network routing decision loads from the GW Router by managing the topology changes reported by the GW Router and the Mid-Tier Routers. From another perspective, as the SDN Controllers are decentralized in the Mid-Tier Network Level, this allows the Mid-Tier network to continue to function with its Lower-Tier Networks in the case that it is isolated from the Upper-Tier Network and other Mid-Tier Networks.

- ***Infrastructure Layer***: The Infrastructure Layer is located at the Lower-Tier Network Level. At this layer, the Edge Routers communicate with Mid-Tier Routers on topology changes. As mentioned in Section 3.1, there are two types of Edge Routers: (i) a single-interface MANET router used by Edge Units, and (ii) a two-interface MANET router deployed at the MANET Vehicle, in which one interface is connected to the Vehicle's local area network and one is connected to the Lower-Tier Network via wireless communication.

3.3. SDN Network Model and Communication Protocols

In the following, we present the SDN network model and the communication protocols between them.

SDN Network Model. Our SDN Model consists of Edge Agent, Mid-Tier Agent, Gateway Agent, SDN Controller, and a Client Management Application, which are described below.

- ***Edge Agent***: The Edge Agent is simply an SDN Agent deployed on one of the Edge Routers at the Lower-Tier Network Level. This Edge Router, which is a member of a Lower-Tier Network, is a single-interface network node. This router is connected to the Lower-Tier Network via wireless communication.

- ***Mid-Tier Agent***: The Mid-Tier Agent, which is also an SDN Agent, is deployed at the MANET Vehicle at the Mid-Tier Network Level as illustrated in Figure 1 and Figure 2. Unlike the Edge Router at the Lower-Tier Network Level, this Mid-Tier Router has two interfaces: one connected to the Vehicle's local area network, and the other connected to the Lower-Tier Network via wireless communication. This Mid-Tier Agent is used to obtain the latest topology changes of its Lower-Tier Network. Additionally, the Mid-Tier Agent is a member of its own Mid-Tier Network's multicast communication group.

- ***Gateway Agent***: The Gateway Agent, which is an SDN agent, also has two interfaces: one connected to other MANET Vehicles' GW Routers and the other connected to its own Vehicle's local area network. The Gateway Agent gives the full topological view of its own Mid-Tier Network and other Mid-Tier Networks. This Gateway Agent is also a member of its own Mid-Tier Network's multicast communication group.

- ***SDN Controller***: The SDN Controller is considered the brain of the Mid-Tier Network. The Controller obtains the topology changes from all Lower-Tier Networks that are connected to the Vehicle and other Mid-Tier Networks, and intelligently redistributes the network routes so that all the nodes of the entire network can communicate with each other. The SDN Controller is also a member of its own Mid-Tier Network's multicast communication group.

- ***Client Management App***: As mentioned in the previous Section 3.2, the Client Application is just a simple CoAP Client in this paper. It is introduced to demonstrate some simple NetOps tasks at the NOC.

SDN Communication Protocols. We now present the SDN Communication Protocols. One is the SDN Discovery Protocol that is designed to obtain the topology changes of Lower-Tier Networks and the Mid-Tier Networks. The other is the SDN Proxy Protocol that is designed to support communications between the Client Management Application and the Edge Agent.

SDN Discovery Protocol: The SDN Discovery Protocol only takes place at the MANET Vehicle. The agents that are involved are the SDN Controller, the Mid-Tier Agent(s), and the Gateway Agent. These agents are required to join the Mid-Tier Network's multicast communication group, so that they can provide the topology changes to the SDN Controller and obtain topology updates from the SDN Controller. Periodically, the SDN Controller sends out a HELLO message, denoted *hello-msg*, to the Mid-Tier Network's multicast group to discover the Gateway Agent and the Mid-Tier Agent(s). A configurable HELLO interval, denoted *hello-interval*, measured in seconds, with a default value of 5 seconds, is used. If the agents receive the *hello-msg*, they will directly send an Acknowledgement (ACK) message back to the SDN Controller that includes its information (e.g., Identity, Router Information, Network Routes). The SDN Controller then compares the information with the previous records, updates the information for each agent, and saves the information in a local cache.

In addition, periodically, the SDN Controller will query the routing information from the local cache and

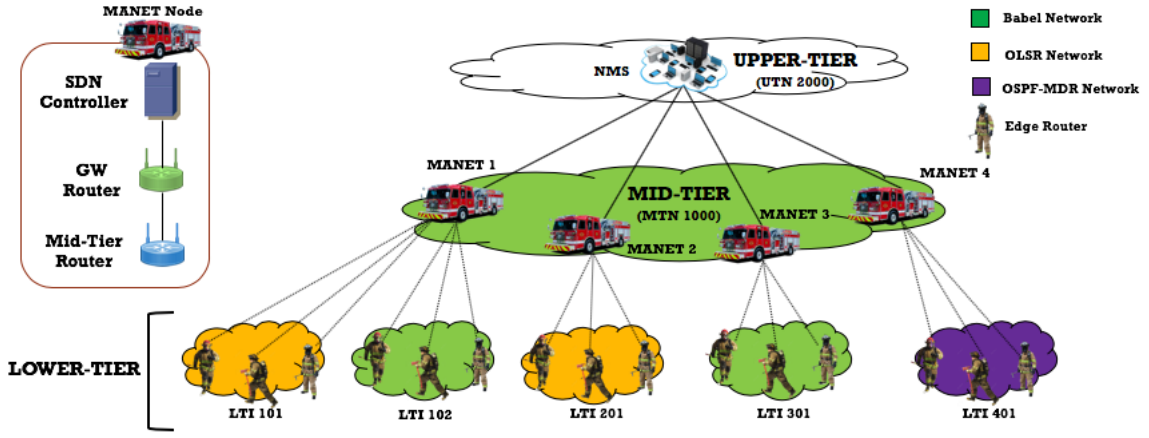


Figure 2. SDN Operational Overview (OV)

distribute the information to the agents. As the general rule of route redistribution (RR), the SDN Controller must not redistribute routes of a network into that same network. Otherwise, a routing loop will occur. The periodic time variable, denoted the *update-interval*, is a configurable variable measured in seconds, with a default value of 5 seconds. Figure 3 illustrates the SDN Discovery Protocol. The variables of the protocol are described in Table 1.

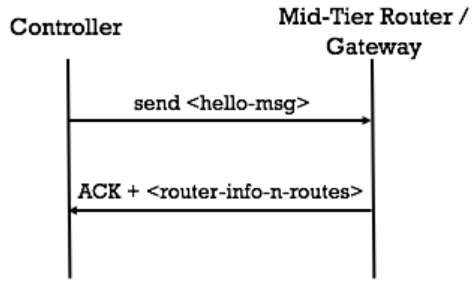


Figure 3. Illustration of SDN Discovery Protocol

SDN Proxy Protocol: As motivated by the proxy functionality of CoAP [37], we design the SDN Proxy Protocol to offload the management task to the SDN Controller by the Client Management App. This protocol is useful in case the Client Application is not reachable to the lowest Edge Agent in the hierarchical MANET network tree. The protocol also helps to reduce the workload at the NOC and allows the centralized management systems to shift the resources to other tasks while waiting for the response from the Edge Agent. The Client Application first sends the management message, denoted *conf-msg*, to the SDN Controller. The SDN Controller then reconstructs the message to make it look like the message originated from the SDN Controller and sends it to the Edge Agent. If the Edge Agent receives the message, it will send a response message, denoted *response-msg*, back to the SDN Controller. The SDN Controller again reconstructs the message and sends it back to the Client App. Figure 4 illustrates the SDN Proxy Protocol.

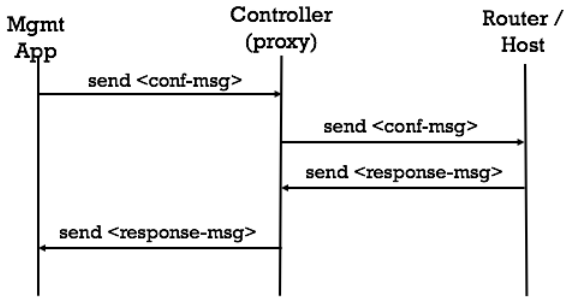


Figure 4. Illustration of SDN Proxy Protocol

Table 1. SDN Discovery Protocol Variables

Variables	Description	Values
hello-interval	periodic time to send out the *hello-msg*	5 s
update-interval	periodic time to send out the *update-msg*	5 s

We now analyze the SDN Discovery Protocol. As described earlier, the Discovery Protocol has two processes: (i) the discovery of the Mid-Tier Agents and (ii) the route redistribution (RR). Although these two processes can be implemented to run independently from each other, the overall Discovery Protocol works best if there is a correlation between them. The RR should only run if there are new routing updates found by the discovery process. Moreover, the RR should not run while the discovery process is in progress. Thus, the performance of the SDN Discovery Protocol is largely affected by the two variables: (i) *hello-interval* and (ii) *update-interval*. The *hello-interval* indicates how often the discovery of the Mid-Tier Agents runs while the *update-interval* indicates how often the RR is executed. Our goal is to define the most proper values of the *hello-interval* and the *update-interval* that can provide the best performance of the SDN Discovery Protocol.

Assume that the set of the Mid-Tier Agents is $\{r_1, r_2, r_3, \ldots, r_n\}$, where n is the total number of the Mid-Tier Agents. Supposedly, the average time for the Mid-Tier Agents to receive the *hello-msg* from the SDN Controller, denoted as $T_{avg}^{(tx)}$, and the average time for the SDN Controller to receive the routing information from the Mid-Tier Agents, denoted as $T_{avg}^{(rx)}$, can be derived as follows:

$$T_{avg}^{(tx)} = \frac{\sum_{r=1}^{n} t_r^{(tx)}}{n}, \tag{1}$$

$$T_{avg}^{(rx)} = \frac{\sum_{r=1}^{n} t_r^{(rx)}}{n}. \tag{2}$$

We define the average time to complete the discovery of the Mid-Tier Agent as follows:

$$T_{avg}^{(discovery)} = T_{avg}^{(tx)} + T_{avg}^{(rx)}, \tag{3}$$

$$T_{avg}^{(discovery)} = \frac{\sum_{r=1}^{n} (t_r^{(tx)} + t_r^{(rx)})}{n}. \tag{4}$$

The optimal solution is to ensure the discovery process completed before the start of RR process.

$$T^{(update-interval)} \geq T_{avg}^{(tx)} + T_{avg}^{(rx)}. \tag{5}$$

Furthermore, the average time for the SDN Controller to perform the RR for all the Mid-Tier Agents is as follows:

$$T_{avg}^{(rr)} = \frac{\sum_{r=1}^{n-1} t_r^{(rr)}}{n-1}. \tag{6}$$

Notice that, *n-1* means that the SDN Controller will not redistribute the routes of a network *A* into that same network A to avoid routing loop. In order to obtain the optimal performance of the RR, the RR process must be completed before the next discovery cycle. Thus, we define

$$T^{(hello-interval)} \geq T_{avg}^{(rr)}. \tag{7}$$

Overall, the average time that will be taken to complete the discovery of Mid-Tier Agents and execution of RR, denoted as $T_{avg}^{(elapsed)}$, which is derived as follows:

$$T_{avg}^{(elapsed)} = T_{avg}^{(discovery)} + T_{avg}^{(rr)}, \tag{8}$$

and

$$T_{avg}^{(elapsed)} = \frac{\sum_{r=1}^{n} (t_r^{(tx)} + t_r^{(rx)})}{n} + \frac{\sum_{r=1}^{n-1} t_r^{(rr)}}{n-1}. \tag{9}$$

4. SDN Implementation and Scenarios

We now present the SDN implementation and scenarios.

4.1. Overview

From our prior research [15, 30], we know that CoAP performs well in the constrained and mobile networks. Recall that in Section 1, the protocols in Section 3.3 are implemented based on the Californium CoAP (Cf CoAP) [21] and the message exchanges are based on the CoAP standard RFC 7252 [37]. The SDN Agent, which is a subclass of Cf CoAP's CoAPServer, is a base class Agent. The SDN Controller and the Edge Agent are subclasses of the SDN Agent. Furthermore, the Mid-Tier Agent and the Gateway Agent are subclasses of the Edge Agent with an additional network interface.

In addition to the SDN implementation, we integrate the OLSR Network Framework (OONF)'s OLSRv2 [4] into the Quagga Routing Software Suite [18] in order to provide route redistribution to OLSRv2. With this mechanism, both OSPF-MDR and OLSRv2 can automatically be route-redistributed through Zebra when there are topology changes, as illustrated in Figure 5. The Babel routing protocol that we use in this study is implemented by Institut de Recherche en Informatique Fondamentale (IRIF). The message format that we use in the SDN protocols is Google's Protocol Buffers [36]. With the use of Protocol Buffers, our SDN implementation is simpler and more maintainable in terms of constructing and parsing the messages.

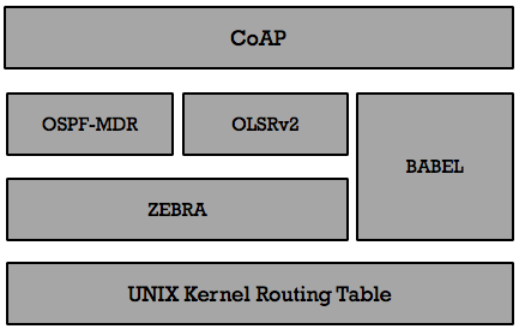


Figure 5. SDN Protocol Suite

4.2. Scenarios of Multi-Domain Operations

We now introduce the possible scenarios of the heterogeneous MANET deployment, including the deployment of a Mid-Tier Network, deployment of a Lower-Tier Network, Unit Assignments, and tests of network connectivity, which will be detailed below.

- ***Deployment of a Mid-Tier Network***: In this scenario, we discuss how our proposed SDN approach to seamlessly interconnects the two Lower-Tier Networks (e.g., LTI 101, LTI 102) into the Mid-Tier Network MANET 1 as illustrated in Figure 6 (a). The Lower-Tier Network, LTI 101, operates on the OLSRv2 routing protocol, while the Lower-Tier Network, LTI 102, uses

the Babel routing protocol. The SDN Controller runs the Discovery Protocol and redistributes the routes from one Mid-Tier router to the other. The expectation of this scenario is that the routers in LTI 101 Network routers should be able to communicate with the LTI 102 Network routers, and LTI 102 routers should be able to communicate with LTI 101 routers.

- ***Deployment of Full Hierarchical MANET***: This scenario is similar to the previous scenario, Deployment of Mid-Tier Network, but considers the interconnection of four heterogeneous Mid-Tier Networks: MANET 1, MANET 2, MANET 3, and MANET 4. The deployment of this scenario is illustrated in Figure 6 (b). The expectation is that all the MANETs should be able to reach from one to another.

- ***Scenarios of Unit Assignments***: In these scenarios, we discuss how to assign an Edge Node to a Lower-Tier Network and how to assign a Lower-Tier Network of Edge Nodes to a Mid-Tier Network. Figure 7 illustrates the scenarios. In the Scenario of assigning an Edge Node to a Lower-Tier Network, as shown in Figure 7 (a), the isolated Edge Node at first is set out of range with the whole network, and then slowly moves to the Lower-Tier Network. The Lower-Tier Network LTI 101 automatically discovers the Edge Node and updates its topology. From this point, every node in the whole network should obtain the network route to reach this newly joined Edge Node. Similarly, in the scenario of assigning a group Edge Nodes to the Mid-Tier Network, as shown in Figure 7 (b), once the isolated group of Edge Nodes joins the Lower-Tier Network LTI 102, every node in the Network should be able to connect to the newly joined Edge Nodes.

- ***Test Connectivity***: There are two different tests of connectivity. For the first test, from one Edge Node of a Lower-Tier Network, we perform ping tests to every other node of all Lower-Tier Networks and measure the Round Trip Time (RTT). Figure 8 (a) shows that the ping tests start from an Edge Node in Lower-Tier Network LTI 101 to all other nodes in all Lower-Tier Networks. In the second test, we perform a NetOps task from the NOC. The NetOps task utilizes the SDN Proxy Protocol that was discussed in Section 3.3. Instead of directly reaching the Edge Node, we utilize the SDN Proxy Protocol to offload the work to the SDN Controller, as in Figure 8 (b). The objective is to compare the overhead of the communications between the Management App to the Edge Node and the Client App to the Edge Node through the SDN Proxy.

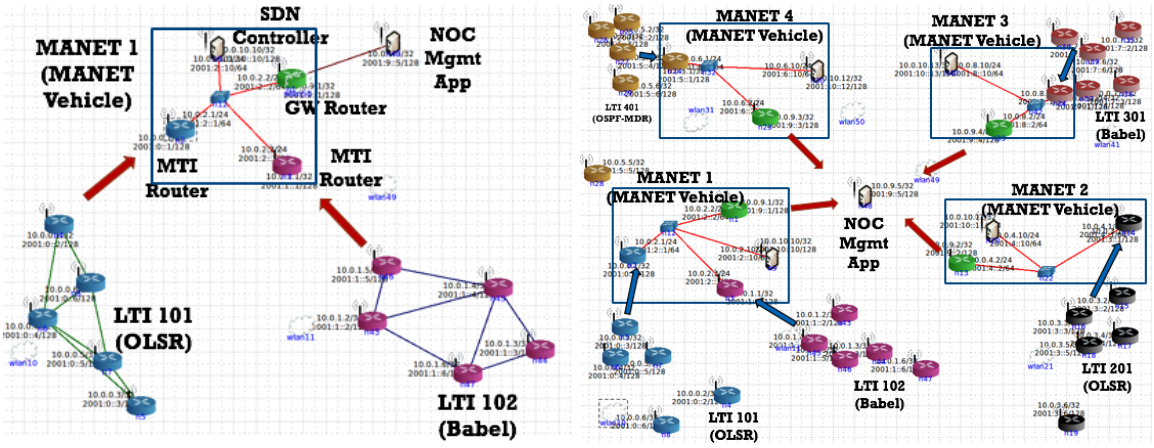
5. Performance Evaluation

We conduct extensive evaluation to validate the effectiveness of our approach. Since the scope of this paper is to introduce a network operational model using SDN, we focus on the functions of the whole network.

Our performance evaluations are carried out in an emulation environment using CORE [3]. In our experiments, we consider the following performance metrics: (i) ***Overhead***: We measure both the total number of packets transmitted per second and the total number of packets received per second. (ii) ***Packet Loss Ratio***: This is the ratio of packet loss when we perform the ping from one Edge Node of a local MANET to other local MANETs. (iii) ***Round-Trip Time (RTT)***: This is the RTT of a CoAP request from the operation center to one Edge Node of a local MANET. We measure both direct communication and through the SDN Proxy.

In the scenario of the Deployment of Mid-Tier Network, as shown in Figure 9, the total number of packets transmitted per second (*txpck/s*) or the total number of packets received per second (*rxpck/s*) at the Gateway Router is not significantly different before and after the interconnection of the two Lower-Tier Networks (LTI 101 and LTI 102) occurs (around 350th second of the experiment). The rates of transmission and the reception of the SDN Controller are also remain generally unchanged. Nonetheless, the SDN Controller noticeably helps to reduce the workload of discovering Mid-Tier Routers and route-redistributing for the Gateway Router. Nonetheless, the rate of transmission of two Lower-Tier Networks increases to nearly double the rate from before the interconnection took place, the most dramatic change being in Lower-Tier Network LTI 101, as they advertise the new routes from other networks in their own networks.

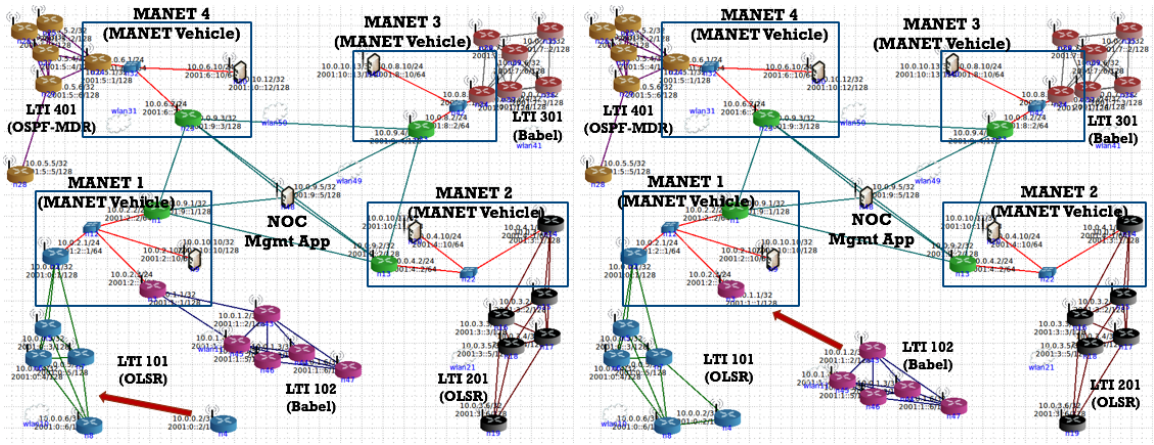
Similar to the scenario in the Deployment of the Mid-Tier Network, the Deployment of the three-level hierarchical heterogeneous MANETs is almost four times in size. In this scenario, we collect the critical data from the SDN Controllers (e.g., MANET 1 Controller, MANET 2 Controller, MANET 3 Controller, and MANET 4 Controller) and the Mid-Tier Gateway Routers. As shown in Figure 10, the interconnection of the networks takes place just before the 300th second of the experiment. The rates of transmission and reception of the controllers remain largely unchanged. The Gateway Routers show some spikes of about eight times greater than normal after the 300th second, but return to the typical rates quickly. During this experiment, the Gateways are performing route redistributions amongst themselves. The Mid-Tier Routers are also redistributing the routes that they learn from the



(a) Deployment of a Mid-Tier Network: LTI 101 and LTI 102 networks are moving close to the MANET 1 Vehicle to form a MANET.

(b) Deployment of a Three-Level Hierarchical MANET: LTI 101, LTI 102, LTI 201, LTI 301, and LTI 401 networks are moving close to the NOC to form a complete network of 4 MANETs.

Figure 6. Scenarios of Hierarchical Network Deployment



(a) Assignment of an Edge Router to a Mid-Tier Network: a new MANET router is added to the LTI 101 network.

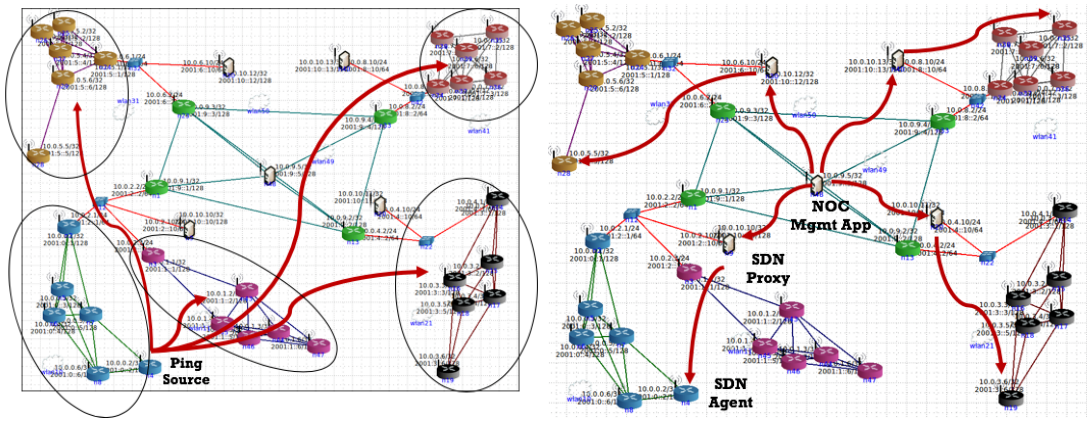
(b) Assignment of a Lower-Tier Network to a Mid-Tier Network: a new LTI 102 network is added to the MANET 1 network.

Figure 7. Scenarios of Unit Assignments

Gateway Routers into their network domains. This means that the SDN Controllers help to reduce the workload of the Gateways. Moreover, the use of Google Protocol Buffers seems to be efficient in end-to-end communication due to the low rates of transmitting and receiving of packets.

In the scenarios of Unit Assignments, in the event of assigning an Edge Node to the Lower-Tier Network LTI 101, the volume of traffic seems to have little effect, as demonstrated in Figure 11. On the other hand, in the event of assigning a Lower-Tier Network LTI 102 to the Mid-Tier Network MANET 1, the traffic is significantly different around 170th second, as seen in Figure 12. The rate of transceiving of the controllers do not observe noticeable effects, but the GW Router sees some loads more than four times that of the peak prior to the event.

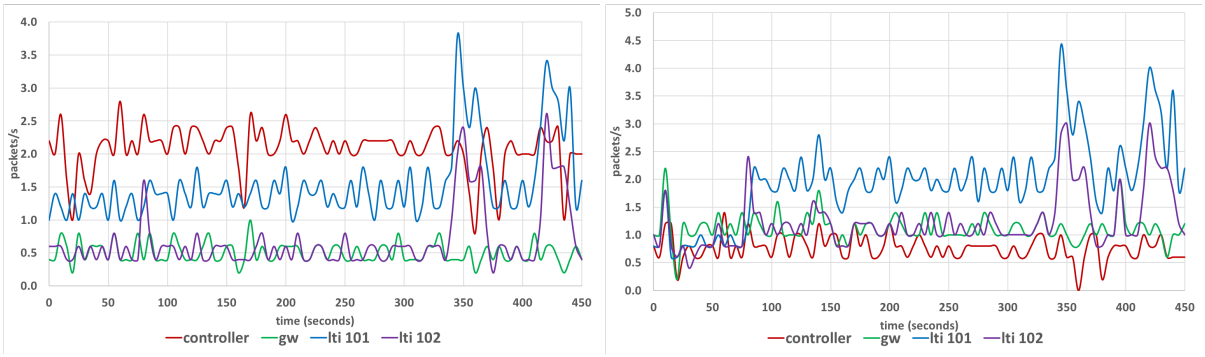
As explained in Section 4.2 concerning the testing of network connectivity, we perform 10 test pings from an Edge Node of the Lower-Tier Network LTI 101 to every other node of all the Lower-Tier Networks. We measure the RTT performance in emulation scenarios of 0 % and 5 % packet loss ratios. As predicted, the average RTTs of the Lower-Tier Network LTI 101 are significantly lower than the average RTTs of the other Lower-Tier Networks, because the ping source is in the same network. When we perform the ping tests at the NOC, the average RTTs are smaller compared to the previous tests, regardless of whether the network is randomly emulated with 0 % or 5 % packet loss ratio.



(a) Testing Connectivity: from one of the MANET routers to all other MANET routers in the complete network of 4 MANETs.

(b) Performing a NetOps Task from the NOC: SDN Client delegates NetOps tasks to the SDN Proxy.

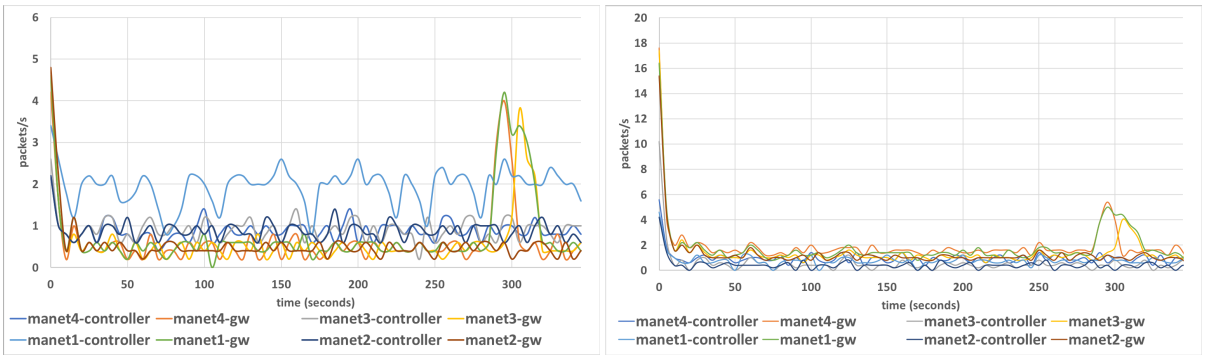
Figure 8. Testing Connectivity



(a) Total Number of Transmitted Packets per Second

(b) Total Number of Received Packets per Second

Figure 9. Deployment of a Mid-Tier Network



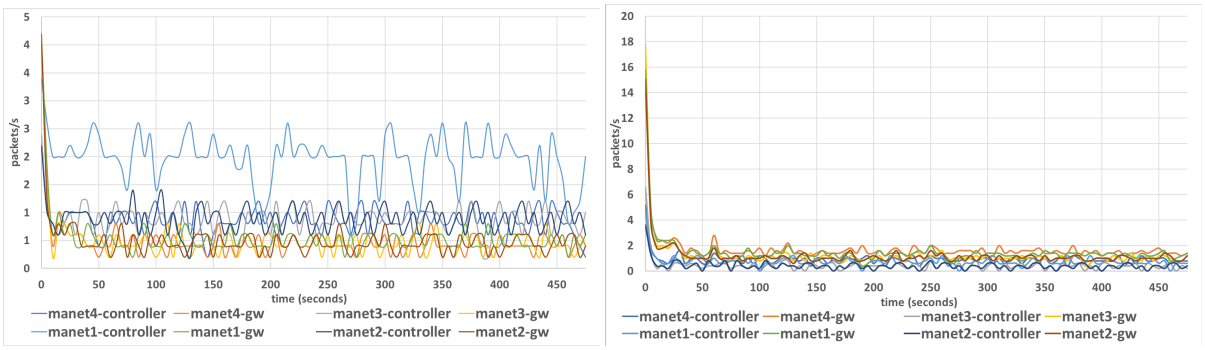
(a) Total Number of Transmitted Packets per Second

(b) Total Number of Received Packets per Second

Figure 10. Deployment of a Hierarchical Heterogeneous MANET

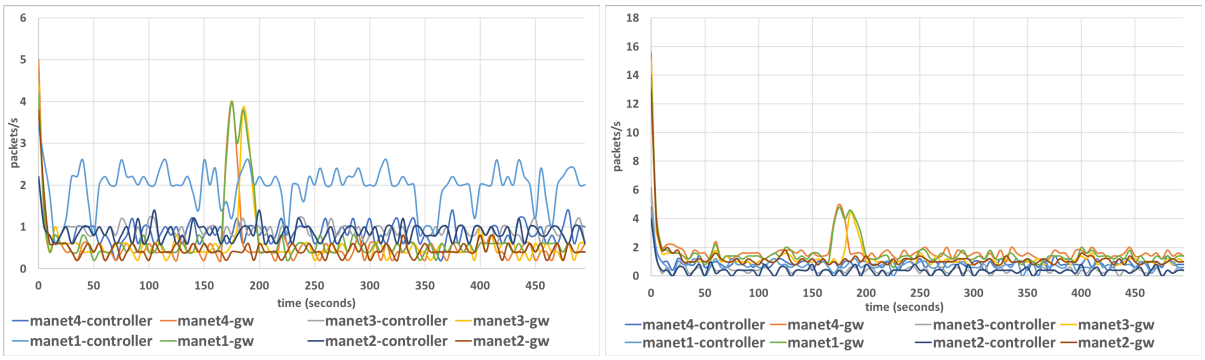
The packet loss ratio of the test ping from the NOC to an Edge Node is smaller than the packet loss ratio of the test ping from an Edge Node to an Edge Node. This is attributable to the path from the NOC to one of the Edge Nodes being shorter than the path from an Edge Node to an Edge Node.

In the evaluation of the SDN Proxy Protocol, the connectivity tests are performed via the CoAP-to-CoAP communication and CoAP-to-Proxy communication from the NOC. Figure 14 indicates *Direct Communication* and *Proxy Communication*. When the packet



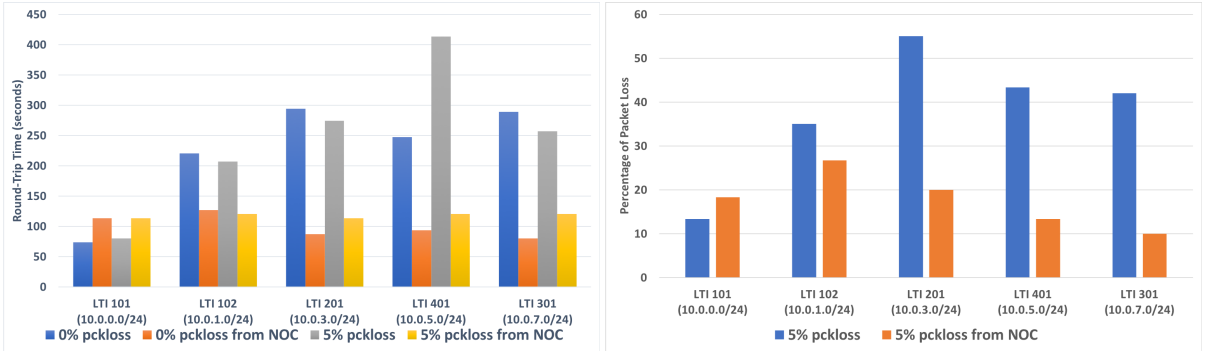
(a) Total Number of Transmitted Packets per Second (b) Total Number of Received Packets per Second

Figure 11. Unit Assignment - Assigning an Edge Node to the Networks



(a) Total Number of Transmitted Packets per Second (b) Total Number of Received Packets per Second

Figure 12. Unit Assignment - Assigning a Lower-Tier Network to the Networks



(a) RTT vs. LTI Networks (b) Packet Loss Ratio vs. LTI Networks

Figure 13. Tests of Connectivity

loss ratio is 0 % (meaning the networks are unrealistically stable), the performances of CoAP-to-CoAP communication and CoAP-to-Proxy communication are very similar. However, the CoAP-to-Proxy communication surprisingly outperforms CoAP-to-CoAP communication under the 5 % packet loss ratio in the network. The CoAP-to-Proxy communication path consists of two paths: one from the NOC to the SDN Controller (2 hops from each other), and one from the SDN Controller to the Edge Node. The performance deteriorates if the packet loss occurs on every link between two hops along the path. The RTT is also greatly affected by the hop count of the two paths. The shorter the two paths is, the shorter the RTT. In other words, the shorter path means the probability of better performance is higher. In this case, the RTT is most likely affected by the path from the SDN Controller to the Edge Node. However, it is not entirely clear why this outperforms CoAP-to-CoAP,

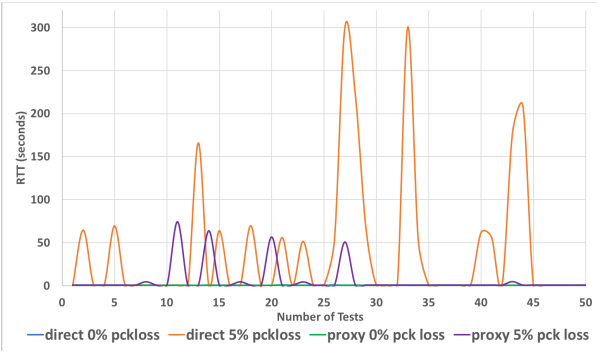


Figure 14. Direct Communication vs. Proxy Communication

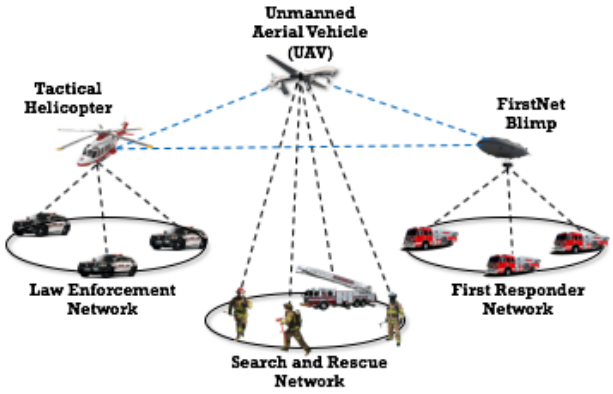


Figure 15. Distributed SDN Controllers

but the likelihood of a smaller RTT for CoAP-to-Proxy is feasible, and requires further investigation.

6. Extensions of SDN-based Network Operations

In this section, we discuss a number of different deployments along with architecture that show how SDN-based network operations (NetOps) can be applicable in the real-world applications. The extensions of SDN-based NetOps offers visions and future directions of SDN applications ranging from real-world implementations to the architecture of Internet of Things (IoT).

6.1. Distributed SDN Controllers in 3-Dimensional Dynamic Networks

The current SDN framework that we propose earlier can be more extensible and distributed when the SDN Controllers share the routing information and take the workload off the Gateway Router. In this architecture as illustrated in Figure 15, the SDN Controllers are deployed in the drone, the blimp, or the helicopter. In this situation, the SDN Controllers share the networking routes of its domain to other SDN Controllers. When an SDN Controller receives network routing information about other domains, it updates its routing table and distributes the reachability of other domains in its domains. Moreover, the SDN Controller shares its own network routing information and the reachability of other domains to the adjacent SDN Controllers. While the concept is related to [31], the concept that we propose is to extend the architecture to three-dimensional heterogenous multi-domain MANETs. The network performance can also be optimized by different metrics such as the hop count, network bandwidth, distance between SDN Controllers, and others.

6.2. Extension to the Real-World SDN Deployment with Open vSwitch (OVS)

Our SDN-based network operational approach can also be applied in the real-world networking deployment with the use of Open vSwitch (OVS) and its framework of virtualization. The concept of server virtualization, which is not new as the VMWare pioneered in the late 1990s, allows a server to host several virtual machines (VMs). The architecture of virtualization provides a hypervisor that separates the physical resources from the virtual environment and divides the physical resources precisely for the virtual environment. All these VMs can be dedicated to use an allocated resource of the server host and run different operating systems or software stack. For example, a server of virtualization can hosts a mail server, a web server, or any application [11].

The Open vSwitch (OVS), which is an open source, adopts the virtualization. However, OVS uses the a software stack to virtualize the networking layer. In OVS framework, the VMs can be connected to each other on the same server hosts or different server hosts via the virtual networking layer. The main difference between the legacy or traditional Layer 2 switch and OVS is that the OVS is designed to handle dynamic networking environment as the network state may change [11, 34]. Thus, OVS fits nicely in SDN architecture, in which rich and dynamic switching functionalities can be implemented. Our approach can be enhanced by replacing the physical switch and the Mid-Tier Routers as described in Figure 1 with the OVS as illustrated in Figure 16. Inside the OVS server host, the VMs, which can be allocated to host the routing protocols (e.g., OLSR, Babel, OSPF-MDR), are logically bridged together and communicate with other VMs on different hosts through the data port. The SDN Controller controls and manages the VMs through a management port. Similar to the three-dimensional operational overview in Section 6.1, we vision that the SDN Controllers are decentralized to each domain and also share network routing information among each other.

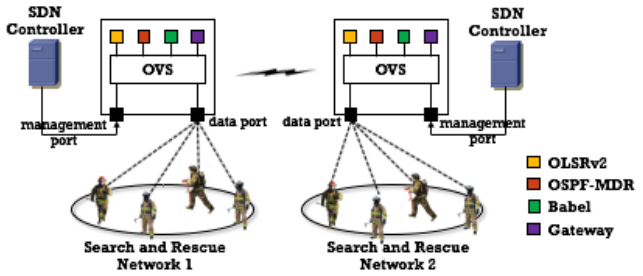


Figure 16. SDN-based Network Operations with Open vSwitch

6.3. Ad hoc Inter-domain Routing Protocol

Although our approach in this paper has already solved the inter-connecting problem of heterogeneous multi-domain networks in which there is an SDN Controller handling the exchange routing information between autonomous systems, the framework can be extended to support the exchange of routing information between the gateways within an autonomous system [14]. As the three-dimensional architecture that was described in Section 6.1, the network operations may be costly as it involves drones or helicopters to provide communication coverage to the network nodes on the ground [24]. Moreover, the three-dimensional architecture does not cover dynamic network topologies when the MANET nodes are on the move. Thus, the ad hoc inter-domain routing protocol is required to support the exchange of routing information within an autonomous system and between two or more autonomous systems as described in Figure 17. The ad hoc inter-domain routing protocol will also support splitting and merging of dynamic network topologies when MANET nodes move around in arbitrary fashion. Additionally, the ad hoc inter-domain routing protocol need to support load balancing as described in Figure 18.

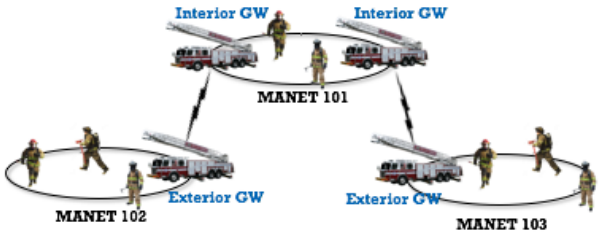


Figure 17. SDN-based Ad hoc Inter-domain Routing Protocol

6.4. Extension of SDN Proxy

In Section 3.3, we introduce an SDN Proxy protocol that can delegate a network operational task to the SDN Controller. The motivation is to offload the management task and allocate resources for different processes.

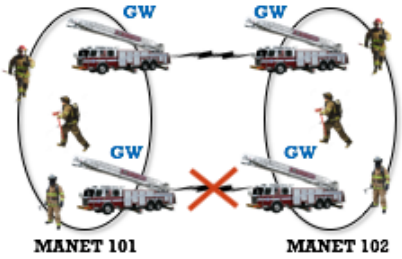


Figure 18. Alternative Gateway

Additionally, the proxy protocol is useful when the client management applications cannot reach the lowest network nodes in the tree-like network topology. Additionally, as the client management applications do not have to have the same transport protocol with the Lower-Tier network nodes, the delegation of executing the network operational task would make more sense. The notable concept that was described in Figure 4 can be extended to support various deployments as follows:

- ***Publisher-Subscriber Communication***: As the demand and the growth of the Today's Internet, Client-Server communication cannot support numerous applications. Round-trip of request and reply communication may be too costly and expensive as the end-to-end communication is not reliable. To address the problem, the Publisher-Subscriber Communication is a feasible way that can significantly reduce overhead in terms of energy consumption and average of number of transmitted and received packets [38]. The Publisher-Subscriber Communication can be deployed at the SDN Proxies. In this architecture shown in Figure 19, the Lower-Tier network nodes publish the changes of network states or application data to the SDN Proxies while the Management App at the NOC subscribes for the data of interests.

- ***Inter-domain Multicast Routing***: As mentioned earlier, the client management application at the NOC may not have direct access to or may not be in the same group communication with the Lower-Tier network nodes. The client management application, in this case, either send a unicast message to a SDN Proxy or send a multicast message to all the SDN Proxies. The SDN Proxies, then, reconstruct the message and replay the messages to all the Lower-Tier network nodes in their domains. The SDN Proxies, then, wait for all responses from the Lower-Tier network nodes, aggregate the responses, and send them to the client management app.

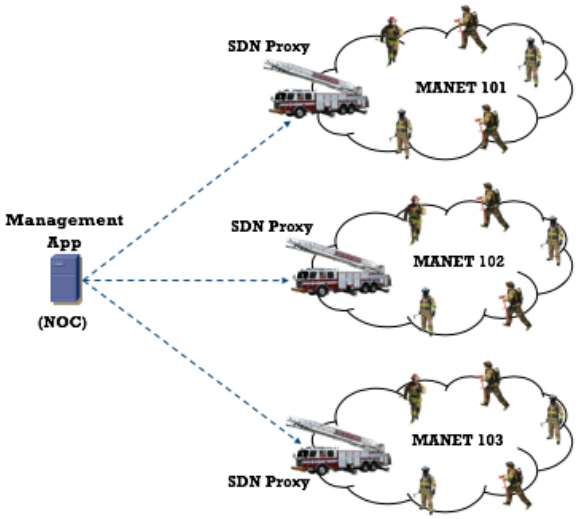


Figure 19. Extension of SDN Proxy Protocol

6.5. SDN in Network Operations of the Internet of Things (IoT)

According to [1, 10], The global IoT market will estimably be $457B by 2020. Additionally, by 2025, there will be over 75 billion IoT devices surpassed the total number of cellular devices and more than three billion of these IoT devices connected to the Global Internet. The domain applications supported by the IoT include smart cities, smart health, smart transportation, smart electricity grid, smart manufacturing, and others [10, 12, 25, 26, 39, 40]. The challenges still remain as how these humongous number of these IoT devices can be connected to the Global Internet and how to be managed. Our SDN approach can be extended to integrate the IoT devices by developing an SDN-based IoT Gateway to inter-connect the IoT devices and the Edge Computing networks as illustrated in Figure 20, and eventually to inter-connect to the rest of Global Internet. The SDN-based IoT Gateway exchange routing information with other SDN Controllers via East-West or West-East communication protocol. SDN Proxy protocol that we discussed in Section 6.4 can also be used to help data collecting tasks. The Edge Computing infrastructure, which is considered to be a Fog Computing Network, simultaneously collects and processes the data before upload the necessary data to the enterprise cloud in the Core Network [5].

6.6. SDN in Network Operations of Human–Smart Things Networks

According to Columbus, the Wearables will estimably hold the IoT market share about 3% of $457 billion dollars by the year 2020 [10]. Real-world applications would be deployed in health services or mission critical network operations. In health services, the patients may have a variety of wearables to monitor their heartbeats, vitals, temperature, blood pressure, etc. In mission critical network operations, the participants are equipped with smart sensors to assist them with their mission such as GPS location, body temperature, body camera to track the situational awareness, agent detection, motion, batteries, etc. These sensors are connected to a small smart router which is called Cluster Head (CH) to inter-connect these smart devices. The SDN Controller would help to drive and maintain the network connectivities of all the CHs to the rest of the network as described in Figure 20.

6.7. SDN in Studying Situational Understanding (SU)

In the following, we discuss a real-world application by putting all the extensions together as illustrated in Figure 20. The network is a hierarchical network model that consists of three tiers. The Lower-Tier is where all the constrained devices such as tactical radios, sensors, etc. As mentioned earlier in Section 6.6, each domain may have its own deployment network structure: random (mostly applied for MANET or VANET), structured (mostly applied for the IoT), and semi-structured (a combination of MANET and IoT). As the connectivities between the Lower-Tier and the Mid-Tier are not reliable and low bandwidth, time scheduling algorithm of uploading data to the Mid-Tier may be required to overcome latency and congestion issues. The Mid-Tier is where the Edge Computing infrastructure is. These systems consist of SDN Controllers, Fog/Edge Nodes, and high-power communication devices.

The high-power communication devices are deployed to provide network connectivity between the Lower-Tier nodes and the Edge Systems. The SDN Controllers help to inter-connect heterogeneous networks, maintain network connectivities and manage the network nodes at the Lower-Tier. The SDN Controllers can be configured to distributedly provide load balancing for the networks. The cluster of Fog/Edge Nodes, which is a small cloud-based system, collects, processes data collected from the Lower-Tier, and sends the processed data to the Upper-Tier. The presence of the Fog Nodes is to take off the workload of the enterprise cloud systems by simultaneously processing and filtering data before uploading the data to the Upper-Tier. The Upper-Tier is where the core network is. The enterprise cloud-based systems are deployed at the Upper-Tier along with network management systems and tools. The Fog/Edge and Cloud systems compliment each other to provide best-effort real-time data processing. Machine Learning or Deep Learning techniques, with the support of cloud/edge computing platform may also be applied at this tier to analyze network behaviors, including

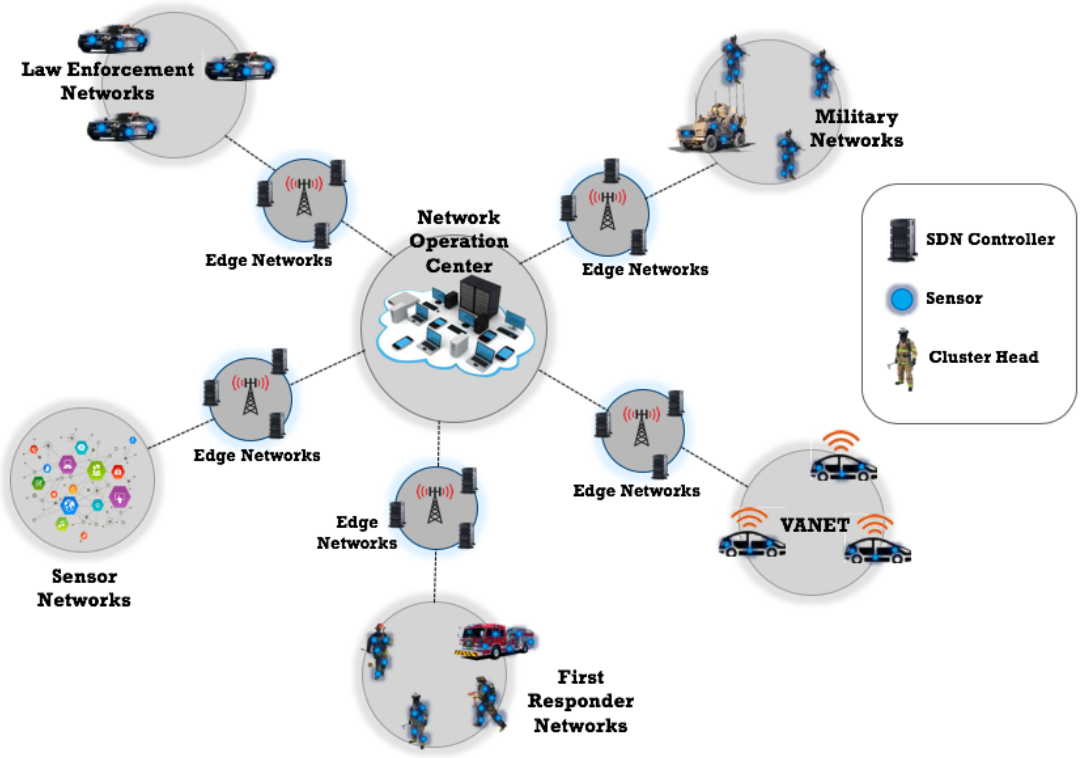


Figure 20. Heterogeneous Multi-domain Networks with the Internet of Things (IoT)

network malfunctions, threat detections, and others [6, 7, 17, 42].

7. Related Work

There has been a variety of research and development of multi-domain networks using SDN in the past. Nonetheless, the existing approaches have primarily focused on heterogeneous Wide Area Networks (WANs) and overlay networks [16, 20, 23, 27, 35]. For example, Phemius *et al.* [35] introduced an east-west communication between the SDN controllers to provide end-to-end network connectivity and other services. The approach was based on the Floodlight OpenFlow and Advanced Message Queuing Protocol (AMQP). The implementation and the performance evaluation were conducted on a small set of small network devices in a reliable wired networking environment. This effort was closest to what we have done in this paper, but the focus was not in the unreliable environment of MANET. One major difference is that the Floodlight framework used in their study, compared with the Californium CoAP as the baseline source code to build a generic SDN framework.

Miguel *et al.* [27] designed and implemented a CoAP-based Control Plane for SDN Wireless Sensor Networks (WSN) for the Contiki Operating System (OS). Their approach offered a comprehensive architectural model, including specifications of the control plane, infrastructure communications, control plane protocols, and basic network functions built in controllers. In addition, their approach focused on the deployment of centralized homogeneous WSN specifically on Contiki OS rather than a multi-domain heterogeneous MANETs. In addition, Yu *et al.* [16] introduced a centralized network management using SDN for MANET. The designed approach was to use a centralized management unit to make the routing decisions (adding routes, removing routes, updating topology, etc.) by communicating directly to every MANET node using Device-to-Device (D2D) communications. The overhead and the use of resources were minimal because all routing tasks were performed at the centralized unit. Such a deployment seems to be impractical in a mission critical network, however, because the entire network would be vulnerable if the centralized unit is interfered with or destroyed during the mission. As every deployed node is required to be configured and communicated to the centralized unit, their proposed approach also took away the MANET's characteristics of self-organized, self-healing, and dynamic topology.

Similar to our proposed solution, Kim *et al.* [20] proposed to use CoAP to develop an SDN framework for Advanced Metering Infrastructure (AMI) of the smart grid. In this approach, the communications between the SDN controllers and the AMI devices were introduced. The AODV routing protocol was used on AMI nodes, and the routing performance between AODV and the SDN framework was compared.

The performance and analysis were based on a testbed of 25 to 125 constrained meter devices in a combination of virtualization environment with VMWare and simulation with NS-3. The key difference between Kim's proposed solution and our approach is that our approach is designed to solve the network management problem of interconnecting of multi-domain heterogeneous MANETs.

8. Final Remarks

In this paper, we proposed an SDN-based approach to support dynamic networks, i.e., multi-domain heterogeneous MANETs. We demonstrated how to leverage the concept of SDN to seamlessly interconnect hierarchical heterogenous MANETs. Our approach is to decentralize the SDN Controllers in the Mid-Tier Network Level to reduce the workload of making routing decisions at the centralized management station (NOC), to give the Mid-Tier Networks capabilities to fully function in case MANET nodes are out of range from the rest of the networks, and to support automatic splitting and merging of the networks without requiring manual configuration management. In our SDN framework, we adopted the CoAP protocol as the Northbound API between the Client Management Application and the SDN Controllers, and as the Southbound API between the SDN Controllers and the Lower-Tier Networks. In addition, at the Mid-Tier Network Level, we designed the SDN Discovery Protocol to automatically discover the Lower-Tier Networks and seamlessly connect them to the rest of the network.

Our experimental results show that the SDN Controllers help to reduce the routing workload of the GW Routers and significantly improves network performance. We also designed and implemented the SDN Proxy Protocol to offload the NetOps tasks from the Upper-Tier Network Level to the lower tiers in the hierarchical multi-domain MANETs. Our experimental results show that the SDN Proxy Protocol is promising, as CoAP-to-Proxy communication outperforms CoAP-to-CoAP communication when the network is unstable. We also discussed a number of extensions of SDN-based network operations, including real-world SDN development and real-world network deployment.

References

[1] 13 IoT statistics defining the future of Internet of Things. https://www.newgenapps.com/blog/iot-statistics-internet-of-things-future-research-data, Retrieved January 1, 2018.

[2] L. Abusalah, A. Khokhar, and M. Guizani. A survey of secure mobile ad hoc routing protocols. *IEEE Communications Surveys Tutorials*, 10(4):78–93, Fourth 2008.

[3] J. Ahrenholz. Comparison of core network emulation platforms. In *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, pages 166–171, Oct 2010.

[4] C. Barz, J. Niewiejska, and H. Rogge. Nhdp and olsrv2 for community networks. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 96–102, Oct 2013.

[5] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson. Flow based security for IoT devices using an SDN gateway. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 157–163, Aug 2016.

[6] Z. Chen, S. Wei, W. Yu, J. Nguyen, and W. G. Hatcher. A cloud/edge computing streaming system for network traffic monitoring and threat detection. *International Journal of Security and Networks (IJSN)*, 13(3), 2018.

[7] Z. Chen, G. Xu, V. Mahalingam, L. Ge, J. Nguyen, W. Yu, and C. Lu. A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3:10–23, 2016.

[8] J. Chroboczek. The babel routing protocol. (6126), 2011.

[9] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg. The optimized link state routing protocol version 2. RFC 7181, 2014.

[10] L. Columbus. 2017 roundup of Internet Of Things forecasts. https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#62532b71480e, Dec 2017.

[11] Q. Duan, Y. Yan, and A. V. Vasilakos. A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *IEEE Transactions on Network and Service Management*, 9(4):373–392, December 2012.

[12] N. Ekedebe, C. Lu, and W. Yu. Towards experimental evaluation of intelligent transportation system safety and traffic efficiency. In *2015 IEEE International Conference on Communications (ICC)*, pages 3757–3762, June 2015.

[13] R. T. Fielding and R. N. Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Doctoral dissertation, 2000.

[14] W. Gao, J. Nguyen, Y. Wu, W. G. Hatcher, and W. Yu. A bloom filter-based dual-layer routing scheme in large-scale mobile networks. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, July 2017.

[15] W. Gao, J. Nguyen, W. Yu, C. Lu, and D. Ku. Assessing performance of constrained application protocol (CoAP) in MANET using emulation. In *Proc. of ACM International Conference on Research in Adaptive and Convergent Systems*, 2016.

[16] C. Y. Hans, G. Quer, and R. R. Rao. Wireless SDN mobile ad hoc network: From theory to practice. In *Proc. of IEEE International Conference on Communications (ICC)*, 2017.

[17] W. G. Hatcher and W. Yu. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access*, 6:24411–24432, 2018.

[18] P. Jakma and D. Lamparter. Introduction to the quagga routing suite. *IEEE Network*, 28(2):42–48, March 2014.

[19] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu. A survey of multicast routing protocols for mobile ad-hoc networks. *IEEE Communications Surveys Tutorials*, 11(1):78–91, First 2009.

[20] J. Kim, F. Filali, and Y.-B. Ko. A lightweight CoAP-based software defined networking for resource constrained AMI devices. In *Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015.

[21] M. Kovatsch, M. Lanter, and Z. Shelby. Californium: Scalable cloud services for the internet of things with coap. In *2014 International Conference on the Internet of Things (IOT)*, pages 1–6, Oct 2014.

[22] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[23] M. Labraoui, M. Boc, and A. Fladenmuller. Self-configuration mechanisms for SDN deployment in wireless mesh networks. In *Proc. of IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2017.

[24] F. Liang, J. Nguyen, W. Gao, W. G. Hatcher, and W. Yu. Towards UAV assisted multi-path data streaming in mobile ad-hoc networks. In *Proc. of International Conference on Computing, Networking and Communications (ICNC)*, 2018.

[25] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on Internet of Things: architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, Oct 2017.

[26] S. Mallapuram, N. Ngwum, F. Yuan, C. Lu, and W. Yu. Smart city: The state of the art, datasets, and evaluation platforms. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 447–452, May 2017.

[27] M. L. Miguel, M. C. Penna, E. Jamhour, and M. E. Pellenz. A CoAP based control plane for software defined wireless sensor networks. *Journal of Communications and Networks*, 19(6):555–562, 2017.

[28] Z. Movahedi, Z. Hosseini, F. Bayan, and G. Pujolle. Trust-distortion resistant trust management frameworks on mobile ad hoc networks: A survey. *IEEE Communications Surveys Tutorials*, 18(2):1287–1309, Secondquarter 2016.

[29] J. Nguyen and W. Yu. An SDN-based approach to support dynamic operations of multi-domain heterogeneous manets. In *Proc. of the 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2018.

[30] J. Nguyen, W. Yu, and D. Ku. A reliable transport for mobile ad hoc network (MANET) with constrained application protocol (coap) over negative-acknowledgment oriented reliable multicast (norm). In *Proc. of International Conference on Computing, Networking and Communications (ICNC)*, 2018.

[31] M. Obadia, M. Bouet, J. Leguay, K. Phemius, and L. Iannone. Failover mechanisms for distributed sdn controllers. In *2014 International Conference and Workshop on the Network of the Future (NOF)*, volume Workshop, pages 1–6, Dec 2014.

[32] R. Ogier and P. Spagnolo. Mobile ad hoc network (manet) extension of ospf using connected dominating set (cds) flooding. RFC 5614, 2009.

[33] R. G. Ogier. Use of OSPF-MDR in single-hop broadcast networks. (7038), 2013.

[34] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado. The design and implementation of open vswitch. In *12th USENIX Symposium on Networked Systems Design and Implementation, NSDI 15, Oakland, CA, USA, May 4-6, 2015*, pages 117–130, 2015.

[35] K. Phemius, M. Bouet, and J. Leguay. Disco: Distributed multi-domain sdn controllers. In *Proc. of IEEE Network Operations and Management Symposium (NOMS)*, 2014.

[36] S. Popi?, D. Pezer, B. Mrazovac, and N. Tesli? Performance evaluation of using protocol buffers in the internet of things communication. In *2016 International Conference on Smart Systems and Technologies (SST)*, pages 261–265, Oct 2016.

[37] Z. Shelby, K. Hartke, and C. Bormann. The constrained application protocol (CoAP). (7252), 2014.

[38] Y. Tekin and O. K. Sahingoz. A publish/subscribe messaging system for wireless sensor networks. In *2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pages 171–176, July 2016.

[39] G. Xu, W. Yu, D. Griffith, N. Golmie, and P. Moulema. Toward integrating distributed energy resources and storage devices in smart grid. *IEEE Internet of Things Journal*, 4(1):192–204, Feb 2017.

[40] H. Xu, W. Yu, D. Griffith, and N. Golmie. A survey on industrial internet of things: A cyber-physical systems perspective. *IEEE Access*, pages 1–1, 2018.

[41] W. Yu and J. Lee. DSR-based energy-aware routing protocols in ad hoc networks. In *Proc. of IEEE International Conference on Wireless Network (ICWN)*, 2002.

[42] W. Yu, G. Xu, Z. Chen, and P. Moulema. A cloud computing based architecture for cyber security situation awareness. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 488–492, Oct 2013.

[43] W. Yu, H. Xu, J. Nguyen, E. Blasch, A. Hematian, and W. Gao. Survey of public safety communications: User-side and network-side solutions and future directions. *IEEE Access*, pages 1–1, 2018.

[44] D. Zhang, L. Ge, R. Hardy, W. Yu, H. Zhang, and R. Reschly. On effective data aggregation techniques in host-based intrusion detection in manet. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 85–90, Jan 2013.