Performance Analysis of SDN-based network management in Content Centric Networks for WSN*

Shunmuga Priyan Selvaraju[1], Ali Balador[1,2], Hossein Fotouhi[1], Maryam Vahabi[1] and Mats Björkman[1]

[1]Mälardalen University, Sweden, {shunmuga.selvaraju, hossein.fotouhi, maryam.vahabi, mats.bjorkman}@mdh.se
[2]RISE Research Institute of Sweden, ali.balador@ri.se

Abstract. This technical document analyses a model of Content Centric Networking (CCN) in Wireless Sensor Network (WSN) when using Software Defined Networking (SDN) for network management. The CCN architecture reduces networking constraints in WSNs; addressing routing, node addressing and maintenance of network routes. However, the need for broadcast in CCN is inefficient when considering the dynamic nature of WSNs. This is improved by using SDN-based network management. This document explains the steps involved with the development of SDN-based network management in CCN. It also tracks experiments and implementation details of SDN in CCN for WSNs. Empirical results of the experimental evaluation affirms the conviction of SDN-based network management has improved the network performance in CCN in the context of WSNs. In networks up to 81 nodes, throughput has improved by 2 KBps, reliability through packet delivery ratio has been improved by 25% and fluctuations in the routing have been reduced with centralised SDN-based management over normal broadcast-based distributed management.

1 Introduction

In the Internet, there are primarily two methods for data forwarding or propagation between data sender and data receiver – host Centric Networking and Information Centric Networking (ICN) [20]. As the name suggests, the data transportation between two entities occur based on the logical location of the node in the network and only based on the data availability in the host centric networking and ICN respectively. Conventional Internet Protocol (IP) networking is based on host centric networking. Similarly, novel Content Centric Networking (CCN) networking is a type of CCN architecture where the information are referred as content and addressed by topics. Figure 1 illustrates the concepts of both networking types. One could sequentially observe and track the involved steps with each type of networking. First, in host-centric networking the soliciting receiver initiates the communication by requesting the location of data 'temp'. Upon receiving location information, the receiver initiates a new 'end-to-end' communication to retrieve the desired data. In CCN, only two steps are involved, where the soliciting subscriber requests the data with nearby forwarder and retrieves it. In host centric networking, there is a significant additional activity of finding the correct data producer to get the desired data. In complex networks, the step involved with finding the data produced involved computation-intensive routing algorithms. It adds additional burden in resource-constrained sensor nodes in Wireless Sensor Networking (WSN). Conversely, the needed step of locating the data producer is eliminated with CCN.

---

* This work was supported by the Swedish Foundation for Strategic Research via the MobiFog starting grant, and DAIS – Distributed Artificial Intelligent Systems research project.

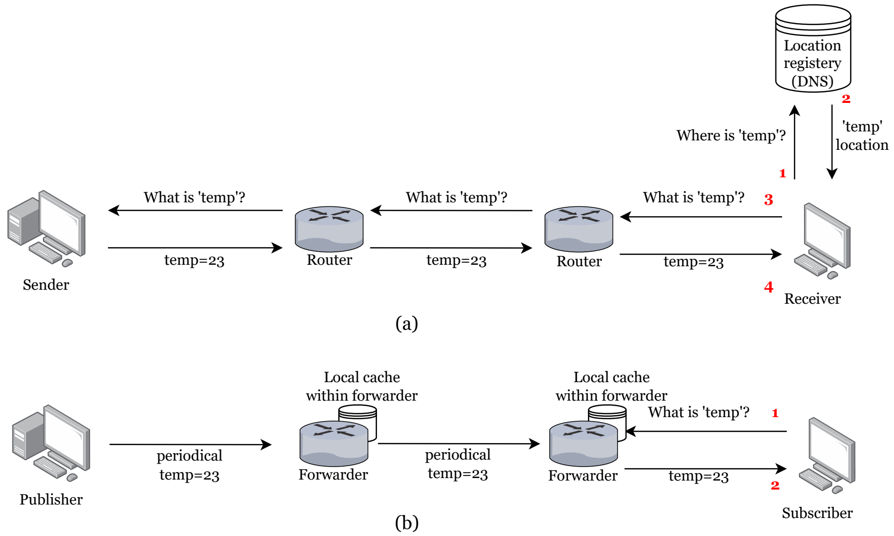

Fig. 1. An illustration of types of networking – (a) Host-centric Networking and (b) Content-centric Networking.

In IP networks, data is routed using network addresses of the nodes involved in the communication, if the nodes are directly connected. It may also involve intermediate nodes or *routers* when the nodes are not directly connected. Additionally, the nodes must establish end-to-end connection with one another to perform communication in IP networks. In CCN networks, data is routed using topics, and the *sender* and *receiver* are replaced by *publisher* and *subscriber*. The publisher nodes produce data under particular topic, and the subscriber requests data under specific topic. The intermediate nodes are called *forwarders*, cache the publishers' data to be accessed by subscribers at will. The subscriber can also set interest for particular topic with the forwarders. Once the data is available, the forwarders automatically push the interested data towards subscriber. The forwarders periodically update the data cached by them with newly generated data by the publishers.

Critical processes typically involved in IP networking include routing, host identification and defining networking parameters, which could be Classless Inter-Domain Routing (CIDR) subnet, MAC address tabling and Time-To-Live (TTL) values. From the system resources' viewpoint of a node, IP stack has heavy usage of system resources such as memory access, kernel context switching, CPU processing and kernel time. Thus, the nodes are also heavily concerned with networking and routing in addition to data acquisition or production. This becomes a burden in low-power networks especially resource-constrained sensor nodes. In contrast, the data-driven networking concepts of CCN has reduced the routing, addressing and networking needs of CCN and so, the nodes can be concerned with data production more. The produced data is broadcasted over link-layer medium and the subscribers or forwarders in the vicinity of the publisher receives the data and caches it. However, there is apparent scalability and inefficient broadcasting issues without intelligent caching and prioritisation techniques.

Coexistence between IP and CCN networking in WSN is a research concern that could be possibly alleviated with proper network management by Software Defined Networking (SDN) technology.

SDN is the method of abstraction and partition of control plane to ease the statefulness, congruity and supervision requirements in networking. SDN embraces NFV to distribute and arrange between isolated network functions, and they can offer network management either as a centralised or distributed management. Centralised management benefits from solidified organised, monitoring and devoted features but requests a strong a control channel, while distributed management is composed over a set of nodes through separate channels. These nodes are grouped and single failure don't influence the management operation, however there's an clear delay in engendering of data to all the hubs. SDN, by definition, provides platform for coexistence and interplay between different network services. SDN for WSN could use localized framework for reactive control with low complexity and overhead [10]. Utilization of a common platform to host multiple SDN controllers could enable seamless coordination and multi-tenancy for WNV in WSNs [19]. Holistic SDN framework can be used to perform end-to-end QoS negotiation and cooperate with external network to optimize the entire service from core to edge. The challenge is in extending the regular protocols of SDN interfaces into the sensing layer, which has a comparatively lowly capability than regular data-plane in any other SDN framework in traditional networks.

This technical document is the continuation of the experimental analysis and incorporation of new networking architectures along with their evaluation in our testbed proposed in [24]. The methodology and design principle for the experimentation is similar to our previous work, where we have considered two strategies for network management – distributed and centralised management, in CCN for WSN. This documentation is primarily concerned with the network management process of CCN in WSN. Below, we present the state-of-the-art for CCN in WSN and SDN in WSN in section 2. In section 3, we discuss the background for research motivations for coexistence of IP and CCN networks and design of SDN in WSN in detail. In section 4, we describe the implemented experimental setup with emphasis on the systemization technology and challenges. Results and discussion are detailed in the section 5. Finally, we summarise the technical document in Conclusion.

2 Related works

We explore related works in two research directions namely – CCN in WSNs and SDN in WSNs.

2.1 CCN in WSNs

In [22], the authors present single-channel cluster-based information-centric WSN (CCIC-WSN) and developed the key processes involved with network management in CCN networking. The processes include attachment of lower nodes with the head nodes in a cluster networking topology, data propagation algorithm between the lower and head nodes and service discovery at the head nodes by the lower nodes. The NDN style ICN based framework has been numerically analysed and simulated with clustering and lower node mobility use cases, and showcased improvement in resource usage and shortened the retrieval process at the nodes. The proposed framework must be implemented to consider the adaptability challenges factors in WSN.

In [7], authors have primarily focused the design and development of caching strategy from the data centric and pervasive WSN in IoT paradigm. The design of the cache strategy involves the free space of the cache, prioritisation in cache placement and decision-making in cache replacement strategy. The proposed algorithm has been implemented using TinyOS on Telosb sensor nodes with average message sizes of 14 Bytes. They have evaluated their caching strategy in comparison with conventional strategies such as random and FIFO. The algorithm shows an improvement in successful hit ratio while content caching, mean delay, mean hops between publisher and subscriber, and efficiency in resource utilisation.

In [21], the primary usage of CCN in content delivery networks and big-data scenarios have been replicated in IoT networks with the motive to handle the deployment challenges of ubiquitous smart things. The design strategy has also considered the information freshness in CCN networking in critical IoT. To reduce the impact of Quality-of-Service (QoS) demands in the communication level, the data freshness is checked by QoS requirement stored at the forwarders which is pushed to them by the subscriber in their interest information. Proposed work has been simulated using NS-3 simulator to evaluate cache hit ratio, mean number of hops and cache delay.

Challenges such as reachability, data-access mechanism, mobility support and lean caching in CCN for IoT networks have been analysed in [28] with Mobile Edge Computing (MEC) technology. Likewise, the article suggests a protocol for cluster-based routing in data centric networking of WSN, called Cluster Based Routing protocol for Information Centric WSN (CBR-ICWSN). It utilizes the continuous non-linear black widow optimization technique to select the optimal set of cluster heads effectively with oppositional artificial bee colony (OABC) routing optimisation. Proposed technique has been analytically evaluated and simulated for energy, sensor node or data availability and end-to-end delay.

[8] argues for the combination of widely researched geographic forwarding routing technique and CCN. In this work, they have proposed an CCN-like ICN compliant and robust implementation of geographic forwarding for ICN to combine both paradigms of node-based and data-based forwarding using Greedy Perimeter Stateless Routing (GPSR) method. The proposed algorithm includes complete methods of neighbour discovery, secure beaconing, and geographic forwarding based naming. Proposal has been implemented in openmote based IEEE 802.15.4 nodes and RIOT OS and tested for system overload, network overhead, energy consumption and cost of security.

2.2 SDN in WSNs

In research works related to SDN in WSNs, we summarise the works that implicitly enable network management between varying network architecture, coexisting radio, routing logic and network functions. This gives us an overall scenario of network management with varying networking methods in WSN.

IPv6 is touted as de-facto protocol for networking in IoT. Accompanied by adaptive methods of 6LoWPAN – IPv6 over Low-Power Wireless Personal Area Networks, IPv6 is enabled in constrained WSNs [11]. Though 6LoWPAN was initially conceived to be built over IEEE 802.15.4, it is also adapted for other wireless technologies including BLE, ZigBee and power line control. With IPv6, 6LoWPAN enables routing in network layer between heterogeneous radios. Nevertheless, standard 6LoWPAN protocols are plagued with reduced packet delivery ratio, as it involves cumbersome header encoding and fragmentation [5].

Standard RPL provides the concept of instances of Destination Oriented Directed Acyclic Graph (DODAG), which offers a dedicated light-weight mechanism for individual routing patterns [9]. Multiple DODAG instances enable the interoperability between multiple network services in single node. Fair distribution of network resources between multiple RPL instances has been achieved through Cooperative - RPL (C-RPL) [4]. Coexistence of source routing and destination addressed modes in downward routing has been achieved using Dual Mode of Operation - RPL (DualMOP-RPL), even in high-density RPL network [14]. Resource optimization and coexistence among RPL modes prosper interoperability between different network structures in WSN.

Given the low-computational capabilities in WSN, routing additionally serves the task of enabling and delegating the network services. As they do, different network services are multiplexed within the routing domain. For instance, Energy-Efficient cooperative Routing mechanism for Heterogeneous WSNs (EERH) improves energy efficiency and PDR with harmonious sharing of forwarding switches between concurrent network services [12]. And Suitable Election Protocol - Traffic (SEP-T) has addressed and optimized energy-aware routing by proper segmentation and classification of traffic heterogeneity among network services [25].

Multi-path routing is one of the features of network interoperability [29]. Traditionally, WSN has disregarded multi-path routing to avoid routing loops. Reliability focused routing uses multi-path routing through heterogeneous networks to achieve path-redundancy. With proper mechanism for selecting same-ranked parents in acyclic RPL tree, multi-path routing can be enabled to mitigate unreliability and faster convergences [15].

Combined routing techniques targeting energy, path and traffic heterogeneities can bring forth the interoperability between diversified applications focusing varied QoS requirements. In WSN, SDN aims to foster multi-tenancy, programmable network control, and robust interface with external networks. Large Scale - Software Defined Virtualized networking (LS-SDV) has been envisioned for IoT to enable differentiation of services [16]. Abstraction in LS-SDV, interfaces different services enable their multi-tenancy in WSN with logical separation.

Soft-WSN is a SDN based softwarized network management, which can offer policy driven device or topology management in WSN [6]. Programmable network control can fine tune the edge or border router, as they interconnect heterogeneous links. Synchronous combined management framework of local WSN, external SDN, and cloud architecture has been developed [2]. These frameworks ease the interface between constrained WSN and regular internet.

In conclusion, a unified approach must be undertaken to solve many aspects concurrently, as unified approach reduces the overhead in WSN. Additionally, combined approach gives a holistic decision-making technique to enable network management based on SDN efficiently. Based on recent works, we proceed to model and develop a unified approach in network management in other networking architectures such as CCN based on novel and versatile methods.

3 Background

In this section, we look into CCN networks and SDN in WSNs. The design factors of SDN in WSN has been critically analysed for future development of robust SDN management in WSNs.

3.1 Content Centric Networking

Content Centric Networking (CCN), is a networking paradigm based on simplistic subscribe-publish model without need of node localization and addressing in the network. CCN offers several advantages over standard IP network. They can add the benefit of faster data retrieval and cache mechanism, multi-point data access, network management through distributed framework, and masqueraded data access methods for node privacy. Given these advantages, CCN networks can foster support for mobility, proxy-caching, congestion-free network, and flexible resource orchestration. Currently, IP networks have majorly occupied the ecosystem of Internet. Interoperability between CCN and IP networks benefits the development of CCN networks.

In [18], an overlay of CCN over IP network has been presented, where it uses proxy services and inter-proxy protocols to translate TCP/IP packets into CCN message, and vice versa. Usage of protocol translation in TCP layer allows the management of CCN by network manager as an application over IP network. Inherent data aggregation and transient nature in CCN network can readily benefit IoT networks. eXtensible Messaging and Presence Protocol (XMPP) [23] and Message Queue Telemetry Transport protocol (MQTT) [1] are some of publish-subscribe model giving a CCN adaption in application protocol for IoT networks. Interoperability between these CCN protocols and REST protocols such as HTTP has been achieved using descriptive Language for OS architecture in IoT [13]. Programmable platform allows the node to self-identify its address, drivers, and programs for formulating custom service APIs.

3.2 SDN in WSNs

Diversified application and deployment caused development of many vertical silos in WSN solutions which has resulted in extremely fragmented ecosystem [10]. To overcome this problem, SD WISE proposes a programmable WSN; application developers can concentrate on high-level application logic rather than intensive low-level details. Compared to stateless OpenFlow for wired SDN, stateful SD WISE aims to reduce the signalling between sensor nodes and controllers; make the sensor nodes programmable as Finite State Machines; provide the sensor node with duty cycles and support the in-network data aggregation. SD WISE also introduces logical and relational operators ($<, >, ==$ ..) for operating WISE flow table's matching rules on data packets. In this paper, the authors have also proposed the idea of multi-tenancy of software abstracted sensor resources viz. virtual network hypervisor - WISE-Visor. SDN-WISE sensor nodes are programmed with 3 data structures - WISE states array, Accepted ID Array (Accepted nodes' messages) and WISE flow Table. Like SDN flow tables, WISE flow table consists of Matching Rules, Action and Statistics section. Critical Topology Discovery (TD) layer of the SD WISE architecture can access and control the node's behaviour. Topology Management (TM) layer at the controller consolidates the TD information from all the nodes and acts upon it. Controllers can be remotely located, and their message are relayed through Adaption layer which is used for formatting device understandable messages. The implemented SD-WISE using physical sensor nodes, EMB-Z2530PA which provides IEEE 802.15.4 connectivity. Controller and Adaption layer are hosted in a desktop computer. The measurements of RTT and efficiency under different number of hops and payload sizes have been provided.

[3] presents research challenges with respect to SDN in IoT network and proposes μSDN framework for Contiki OS. SDN provides network optimization with global view and guaranteed throughput on per flow basis. With global view, inherent uncertainties in WSN network can be minimized. WSN poses a challenge in implementing SDN. Thus μSDN, having inherent low control overhead, can be supported on WSN. In any instant, traditional WSN networks can be categorized into data collection, data dissemination and alerts and actuation. With SDN, WSN can combine any of these two functionalities at same time. μSDN provides protocol optimization by eliminating fragmentation, reducing re-transmission; architectural optimization by source routing, controlling frequency of control message, resettable flowable lifetime; memory optimization by reusing flowtable matches or actions, for instance, two identical rules for same destination can be replaced by one; and controller optimization by having local controller inside WSN to process simple requests. μSDN has modular architecture with API provisions. μSDN introduces three-layer in 6LoWPAN stack: μSDN, μSDN-UDP and μSDN controller. μSDN is a protocol above IPv6 layer, which is interoperable with regular RPL. RPL is also used as fall-back routing mechanism, in case, node fail to support μSDN. μSDN-UDP is UDP transport layer with DTLS and μSDN controller operates above μSDN-UDP. μSDN protocol uses three modules: Controller adapter - abstract controller interface; SDN

engine - core communication logic handler; and SDN driver - flowtable actions, data aggregation and firewall. μSDN processes include controller discovery, controller join using RPL DAOs, node configuration and metrics, flowtable actions and overhead optimizations. μSDN is evaluated inside Cooja with EXP5438 devices under scenarios: network performance evaluation, overhead reduction and re-routing under interference scenarios. μSDN is built upon RPL DAO, so inherently RPL has lower network latency than μSDN and has lower control plane overhead. However, μSDN displays minimal convergence delay in case of interference. The authors have intentionally present limitation and shortcoming of their work as advantages. For instance, as use of source routing has been argued in many works, has bottleneck issues. And μSDN doesn't do perform fragmentation, since it sits on top IPv6 and fragmentation occur only at 6LoWPAN; there is a less credibility in their argument.

The paper [27] presents the implementation and evaluation of SD-6LoWPAN, SDN solution for WSN (Wireless Sensor Network) in Contiki devices. Machine-to-Machine (M2M) communications faces unique network constraints in WSNs, forcing network developers to implement one-time solution for the specific-case. Thus, standardization of the network is required for interoperability in IoT ecosystem. However, standardizing the whole IoT ecosystem is infeasible due to the immense diversity of the devices and the communication standards. SDN can be used for network-level interoperability without the need for standardization. For introducing SDN in WSNs, we need to significantly reduce the control plane overhead incurred in SDN operations. To this purpose, many works have been proposed such as μSDN and SDN-WISE. However, μSDN introduces a new layer to the 6LoWPAN communication stack, and SDN-WISE redefines the complete communication protocol stack. To avoid the use of non-standard protocols, SD-6LoWPAN uses only standard protocol with minimal modification and tools as compared to other solutions. The authors have mainly contributed to developing flow tables; mesh-under routing; SDN and local controller; and integration with 6LoWPAN. There are two routing schemes in 6LoWPAN - network-layer route-over and MAC-layer mesh-under routing, where latter is used here. Popularly, route-over routing has been used rather than mesh-under routing, despite mesh-under routing being more delay efficient. This is due to the lack of efficient routing algorithms in MAC layer. As mentioned in the above work, SDN can handle routing and enable mesh-under routing. SD-6LoWPAN architecture consists of (i) SDN controllers, (ii) SDN nodes, (iii) RPL border routers and (iv) local controllers, routing component in SDN node. Control-plane communication occurs in two directions: northward from SDN nodes and southward from SDN controller. Northward communication towards SDN controller uses RPL (Routing Protocol for Low-power and Lossy Network) messages from local controller. Southward communicates towards SDN nodes uses CoAP (Constrained Application Protocol) messages from southbound interface in SDN controller. Both SDN controller and SDN node exposes RESTful resources containing control information. Data-plane communication occurs as on-link M2M exchange, if the nodes are in the same access network and through border router, if they aren't in the same access network. SD-6LoWPAN is implemented as interceptor layer between network and LLSEC (Link-Layer Security) layers in 6LoWPAN protocol stack of standard Contiki OS. Primarily, SD-6LoWPAN has been evaluated inside Cooja simulator for feasibility. On completion of design verification, SD-6LoWPAN has been evaluated against standard RPL implementation, which uses route-over mechanism. Both have been compared under processing latency and control overhead in terms of network usage. In a physical testbed, 20 Zolertia RE-Motes have been used, where one node acts border router. The results showed that RPL has lower processing latency which is due to the topology updates of SDN. These updates increase overall network usage when compared with RPL. To showcase the heterogeneity support in the network, it could have been better if several WSN radios had been used. As 6LoWPAN has been intrinsically built upon IEEE 802.15.4, this solution may have difficulty in including other radios. Other uncovered limitation is that data-plane communication node-to-node occurs through border router instead of using mesh-networking. This may cause throughput bottleneck at the nodes which are on link with border router.

Paper [19] presents the usage of SDN as Network Management System (NMS) for WSNs. Related works such as MANNA which is an abstracted NMS but lacks adaptability, scalability, and robustness; BOSS proposed especially as a bridge between WSN and universal Plug and Play devices; and TinySDN, Sensor OpenFlow, and SDN-WISE are specific-case SDN implementation without any sophistication in network management. Similar proposals such as Smart and Soft-WSN uses monolithic architecture without modularity. To foster open-access development, SDNMM proposes a distributed modular management framework for rapid provisioning and prototyping of SDN solutions in WSN. SDNMM uses three-layered architecture: application plane, control plane and data plane. Application plane, which is the user-interface of the management framework, handles network monitoring, issuing network policies and providing application logic for WSN devices. Control plane hosts all SDN related network modules which are managed by global controller. Network modules through cluster manager manages network topology, node activities and network-level QoS. Inside global controller, Management Service Interface (MSI) provides Application Programming Interfaces (APIs) which supports flexible, loose coupling between network modules. MSI performs context-based policy adaption to control the modules and classification of module states. Context includes information such as node system resources, module states and application requirements. Data plane includes communication among WSN nodes and internet. Frequency of context collection can been adaptively set based on the available resource, number of running tasks at node and number of neighbours for each node. In MSI state classification, MSI assigns a network module to handle a network task based on requirements of the task and parameters of the network module. SDNMM system consists of four components: SDN core executing sensor activities, SDN node executing cluster manager, controller node executing global controller and Controller-PC executing application plane. SDNMM has been implemented as an adaption of open-source IT-SDN. Controller-PC is a Linux machine. Controller node, SDN node and SDN core are implemented in Contiki for sky motes. SDNMM has been evaluated with one controller and 16,25,36 nodes. Four evaluation metrics have been used: Packet Delivery Ratio (PDR), latency, energy consumption and computational overhead. SDNMM has been compared with SDN-WISE and ITSDN. SDNMM has reduced power consumption, computational overhead and latency, compared to SDN-WISE and ITSDN. This is because the management of multiple SDN-WSN networks and NFV orchestration were oversimplified. For instance, SDN node and controller node are implemented in Contiki which could rise scalability and compromise performance.

3.3 Motivation

We require a low-overhead approach in offering unified network management solution in resource-constrained WSN of heterogeneous and dynamic IoT network. WSN has limited capability in terms of networking. With the help of low-overhead SDN framework, we offload the networking activities to a external SDN controller and utilise the nodes mostly for sensing and actuation activities. Developed SDN framework must enable interoperability between neighbouring domain, robust network management and virtualisation of network functions to open the framework for external collaborations with application providers. Future IoT network will certainly experience a degree of heterogeneity in terms of networking architecture and communication technologies which could be offset through centralised management offered by a SDN framework.

4 Simulation

The SDN architecture for CCN in WSN has been presented in Figure 2. The inter-domain Gateway *GW* is the terminal data sink for the data produced at the data producer. Each node is a IEEE 802.15.4 sensor node having single channel communication capability. They are equipped single threaded processor with dedicated memory reserve for the main process. In the simulation environment, each node is isolated and delegated a share of the system resource. This effectively allows them independently and in isolation using the *namespace* technology of Linux based system virtualisation. We use network namespace to simulate our nodes with virtual network device namely virtual Wireless Personal Area Network (WPAN) interface, individual instance of the protocol stack, sockets, and ports. In detail, a namespace is an abstraction that encapsulates a global system resource in a way that makes it appear to virtual node processes inside the namespace that they have their own isolated instance of the global resource [26]. Other processes that are members of the namespace can see changes to the global resource, but they are invisible to other processes. Containers are one application of namespaces. A namespace is automatically pulled down if the final process in the namespace ends or exits the namespace without any additional causes. The code for the experimental setup is available online [1].

We considered two scenarios for our analysis, including distributed management simulated through normal broadcast algorithm without priority or selectivity in caching, and the second being centralised management simulated through SDN based virtual network management. The packets carry both the control plane and data plane packets simultaneously with upward route information preceding the sensory data. We consider a scenario where the publisher constantly publishes instant timestamps with microsecond precision. Figure 2 also depicts the propagation of the data in the CCN network with nodes closer to the data producer having more recent data than the farther nodes. In distributed management, all the nodes in the network are made to subscribe to the default topic published by the publisher. In the setup, the node ID is extracted from the MAC address of virtual node, as the topic and all the subscribers are hard coded to be interested in the particular topic at the beginning of the simulation. In normal broadcast scenario, all the subscriber also acts as forwarder, meaning they forward the sensory data once they receive them. Without priority or selective caching, all the nodes receive and relay the information simultaneously, effectively creating a *broadcast storm*.

A broadcast storm occurs when a large number of broadcast packets are sent in a short amount of time. As switches and endpoints attempt to keep up with the deluge of packets, a broadcast storm can overwhelm the network infrastructure and the network's performance suffers as a result [17]. Thus, *to counter the effects of broadcast storm, all the nodes are simulated with unrestrained resource utilisation such as always-on behaviour and inexhaustible elastic packet buffer to have thorough analysis of the network performance and CCN networking behaviour in WSN.* With virtually unlimited resources, the nodes are equipped to withstand any deluge of messages. With centralised management through SDN based virtual management, we effectively mitigate the broadcast storm and enabling only specific nodes tactfully. This creates a prioritised and selective caching strategy in the CCN networking with resource constrained WSN nodes. The effects of broadcast storm is shown in figure 2 in distributed management and aversion of it is shown in below figure. Sensor data is observed outside the local network domain where the network performance is of prime importance. The root node is connected with the inter-domain gateway over a separate IEEE 802.15.4 link.

---

[1] https://github.com/SELVARAJUshunmugaPriyan/IdeaValidation-IP-CCN

The message processing algorithm is given in the program 1.1. With single subscribed topic, the order of complexity followed in the packet processing for updating the contents at the nodes can be given by $O(N * M)$ where N is the number of incoming messages and M is the contents in the cache. This creates a significant processing delay, if there are more content in the cache, to avoid significant processing delay, we limit the cache storage to 20 messages at a time for a topic, implemented in code line 18. Limit 20 has been chosen based on the depth or number of hops the messages can take before reaching the root node. The SDN management code with virtual network management is given in the program 1.2 and node operations code in 1.3. The SDN calculates the route based on the diagonal forwarder and only those nodes are activated, while all nodes activated by default in distributed networking.

Listing 1.1. Python code of the node process

```python
import socket

layer2_socket = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.ntohs(0x0003)) # IEEE 802.15.4 layer 2 socket created
layer2_socket.bind((’wpan{}’.format(cache[’NODE_ID’]), 0, socket.PACKET_BROADCAST)) # IEEE 802.15.4 layer 2 interface locked with the node function
layer2_socket.setblocking(0) # Concurrent reception and transmission
print(’l2_socket established’)

def NodeProcess():
   # Receive operation
   if not randomPacketDrop() :
      recievedPacket = layer2_socket.receive(123)
      emptySocket(layer2_socket) # clear socket after use
   # Transmit operation
   if publisherNode and not randomPacketDrop(): # generating New content
         sendBuffer = bytes(datetime.now())
         totalSentBytes = layer2_socket.send(sendBuffer)
         print("Total sent bytes $totalSentBytes")
   elif recievedPacket :
      topic = recievedPacket[16:18].decode(’utf-8’)
      routeMap = recievedPacket[18:].decode(’utf-8’).split(’>’)
      content = recievedPacket[-1:]
      if topic in localCache[’interest’] and content not in localCache[’interest’][topic]:
         print(recievedPacket)
         localCache[’interest’][topic].append(_val)
         localCache[’interest’][topic] = cache[’interest’][topic][-20:]  # Clearing Old values
         if sendAllowed and not randomPacketDrop() :
            sendBuffer = recievedPacket + nodeID # Forwarding the latest received content
            totalSentBytes = layer2_socket.send(sendBuffer)
            print("Total sent bytes $totalSentBytes")
```

Listing 1.2. Python code of the SDN process

```
def SDNProcess() :
   if mgmtMode == ’sdn’ : # sdn or mesh
      ForwardersList = [0,1]
      # We select the nodes involved for communication
      for i in range(2, localCache[’XvalueInXxXMAtrix’]):
         # We’re calculating the forwarders based on geographical
            ↪ diagonality
         ForwardersList.append(ForwardersList[-1] + localCache[’
            ↪ XvalueInXxXMAtrix’] + 1)
      ForwardersList.append(localCache[’lastNodeID’]) # Also adding the
         ↪  publisher as the last forwarder
```

Listing 1.3. Python code of the node powering process

```

def nodeOnorOffProcess() :
   if mgmtMode == ’sdn’ : # sdn or mesh
      if localCache[’NodeID’] in ForwardersList :
         localCache[’pwr’] = True
      else:
         localCache[’pwr’] = False
   else :
      # If mesh networking all the nodes are communicating by default
      localCache[’pwr’] = True
```

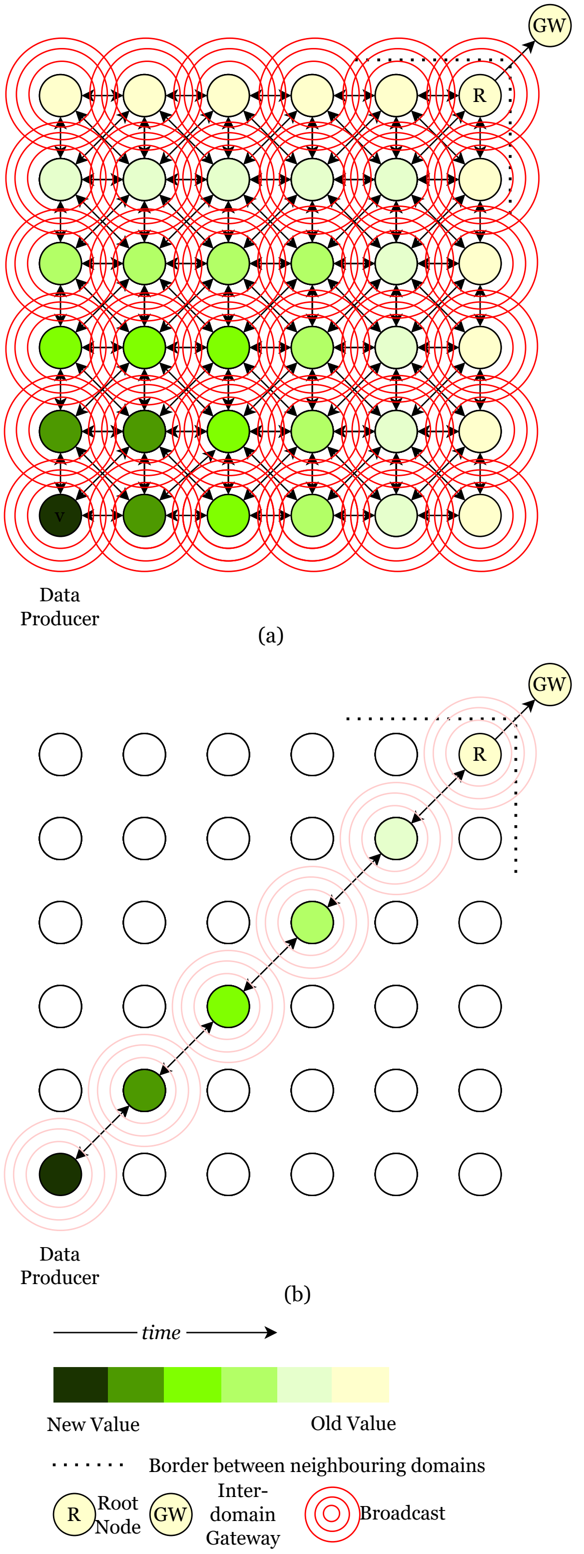

Fig. 2. Implementation of SDN in CCN for WSN. The color shading shows the propagation of the message over time. (a) Distributed Management and (b) Centralised Management.

5 Results and Discussion

We compare the network performance of different network management strategies in line with extension of the evaluation performed in prior work [24]. This document integrates the novel networking architecture of CCN in WSN in our previously proposed and developed simulation environment. In addition to previously observed network performance metrics, we are evaluating the number of hops to signify the behaviour of networking or data routing in presence and absence of network management.

5.1 Evaluation Settings

To analyse the network performance, the GW gateway also acts as an active observer, which means that GW engages in unfiltered reception from the root node at the network-edge. The performance could have been measured in promiscuous mode at the root node, but this has not been preferred to avoid performance implication at the root node which is also involved in the data path. Moreover, measurements must be observed at the egress of the domain to have accurate performance analysis. Additionally, the GW is befitted with required system resources to continuously track and monitor the performance, as scenarios involve with upward of 20000 data transmissions and generated log files weighing more than $43MB$ [2]. The evaluation has been carried out to analyse the networking or routing aspect through number of hops and standard metrics such as throughput (in $KBps$), latency (in milliseconds) and Packet Deliver Rate (PDR) (in $packets/second$). CCN networking performance can be empirically tracked with the number of hops between the publisher and subscriber, as analysed in CCN networking proposals [7], [21]. The latency has been calculated as the end-to-end propagation delay between the data producer and subscriber at the edge of the network domain. Observed PDR is multiplied with the mean packet size (in Bytes) to derive the throughput values. In our experiment, we have improved the message sizes up to $107Bytes$ out of available $123B$ while leaving only $16B$ for MAC and PHY headers in comparison with $28Bytes$ of experimentation in BBR-CVR algorithm [7] in IEEE 802.15.4 wireless medium for experimental setup.

Table 1 summaries the performance evaluation for all the use-cases considered. Comparison between the management strategies has been performed with two scenarios – small networks up to 36 nodes and medium-sized networks up to 81 nodes. The distributed management has been performed through normal broadcasting technique and centralised management through virtual SDN based network management. 81 nodes is the upper-bound of the capability to be handled in current testbed, given the operational limitations of a stand-alone computer. Network density ranges between 4 and 81 nodes, and the simulated packet drop (in %) ranges between 0 and 80 %, to simulate best-effort lossy nature of the WSN. Small network scenarios were conducted for a total of 10 simulation seconds and mid networks were simulated for 180 seconds, given the required time for network convergence in distributed management. Also, the networks beyond 9, 25 and 36 nodes cannot be evaluated for packet losses more than 80, 40 and 30 % respectively as shown in table 1 given the multi-hop dependencies for the communication in larger networks.

5.2 Discussion

Following, we will discuss the observed network performance summary:

---

[2] Online repository of test logs: https://github.com/SELVARAJUshunmugaPriyan/IdeaValidation-IP-CCN/tree/master/Idea-validation-ip-ccn/TestLogs_01032108

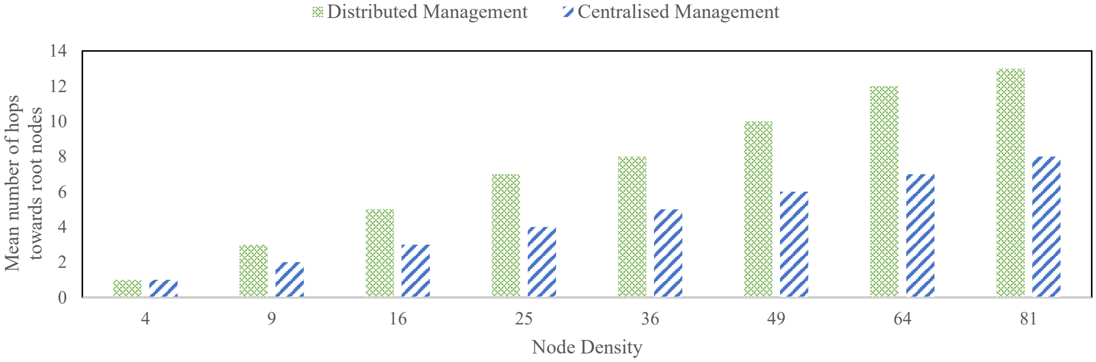

Fig. 3. Mean Number of Hops towards Root Node against varying network density (in total number of simulated nodes).

Number of Hops Towards Root Node. The number of hops quantitatively shows the path taken by the data packets with larger number indicates indirect and cyclic routes taken by the data messages. Ideally SDN provides direct path with minimum hops between the inner data publisher and the network edge. Figure 3 gives the comparison in the mean number of hops taken by the data messages under each of the management strategy. Centralised management clearly has the advantage of whole network view to activate only the necessary nodes involved in the communication. Figure 4 give the mean against the varying latency. Herr, the main takeaway being the fluctuation in the routes as indicated by the mean which is indicated in the standard deviation ($S.D.$) graph 5. The fluctuation in the route is mainly due to the availability of multi-path routing in CCN networking with random variations in loss% selective routes are enabled. In the graph 5, we could observe that the centralised management does not experience in the data-path.

Mean End-to-End Latency Figures 6 and 7 gives the mean end-to-end latency between the core data-publisher and network-edge root-node subscriber. We could observe that the latency values are out of bound ranging between 10s to 100s of milliseconds in distributed management and consistently within 50 milliseconds in centralised management against varying network density shown in 6. When considering mid-sized networks, the network has mostly converged in distributed management strategy resulting in lower latencies given the longer observation period occurred based on the larger network size. We could also observe a declining trend with varying loss in graph 7 due to limited communication as more nodes are unable to communicate and clearing the wireless channel. This phenomenon is similar to the latency observed in heterogeneous wireless networks with varying radio co-existence [24].

Mean Packet Deliver Rate Figures 8 and 10 gives the mean Packet Deliver Rate (PDR) between the core data-publisher and network-edge root-node subscriber. Packet delivery rate has been calculated instead of the Packet delivery ratio, due to the technically unavailable of original transmission count. We could observe that the PDR values are out of bound ranging between 90 and 10 packets per second in distributed management and greater PDR between 140 and 20 in

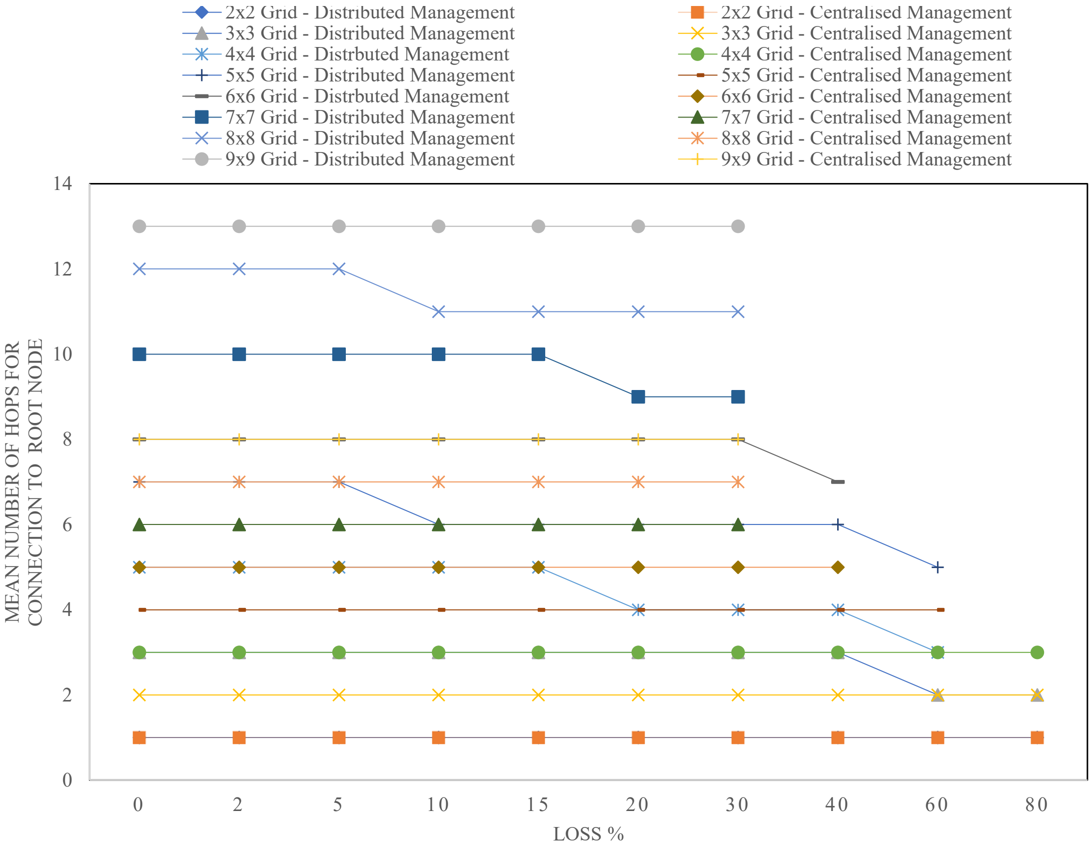

Fig. 4. Mean Number of Hops towards Root Node against varying simulated Packet Drop (in %).

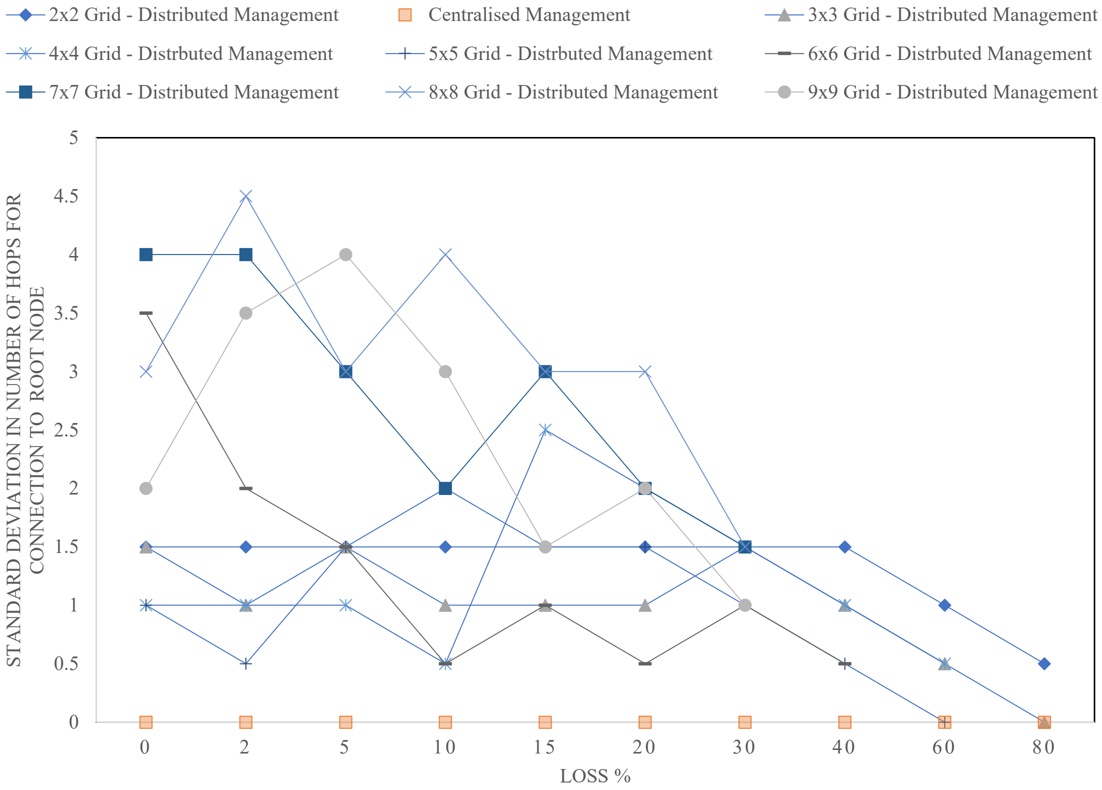

Fig. 5. Standard Deviation (S.D.) of Hops towards Root Node against varying simulated Packet Drop (in %).

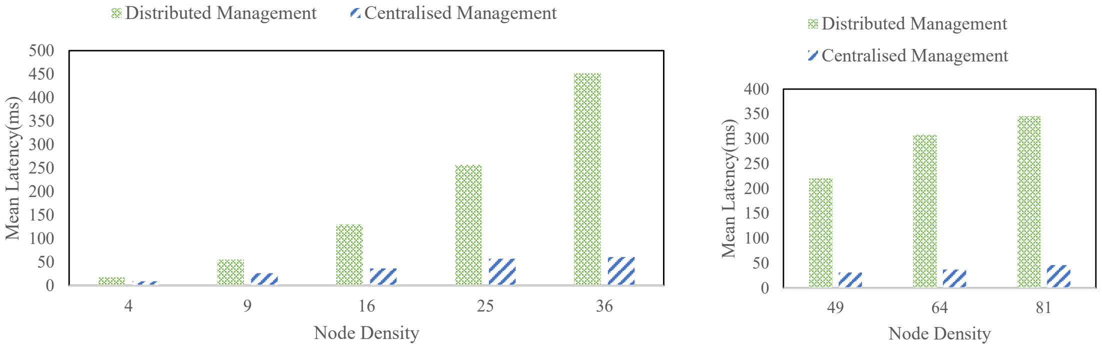

Fig. 6. Mean end-to-end Latency (in milliseconds) against varying network density (in total number of simulated nodes)

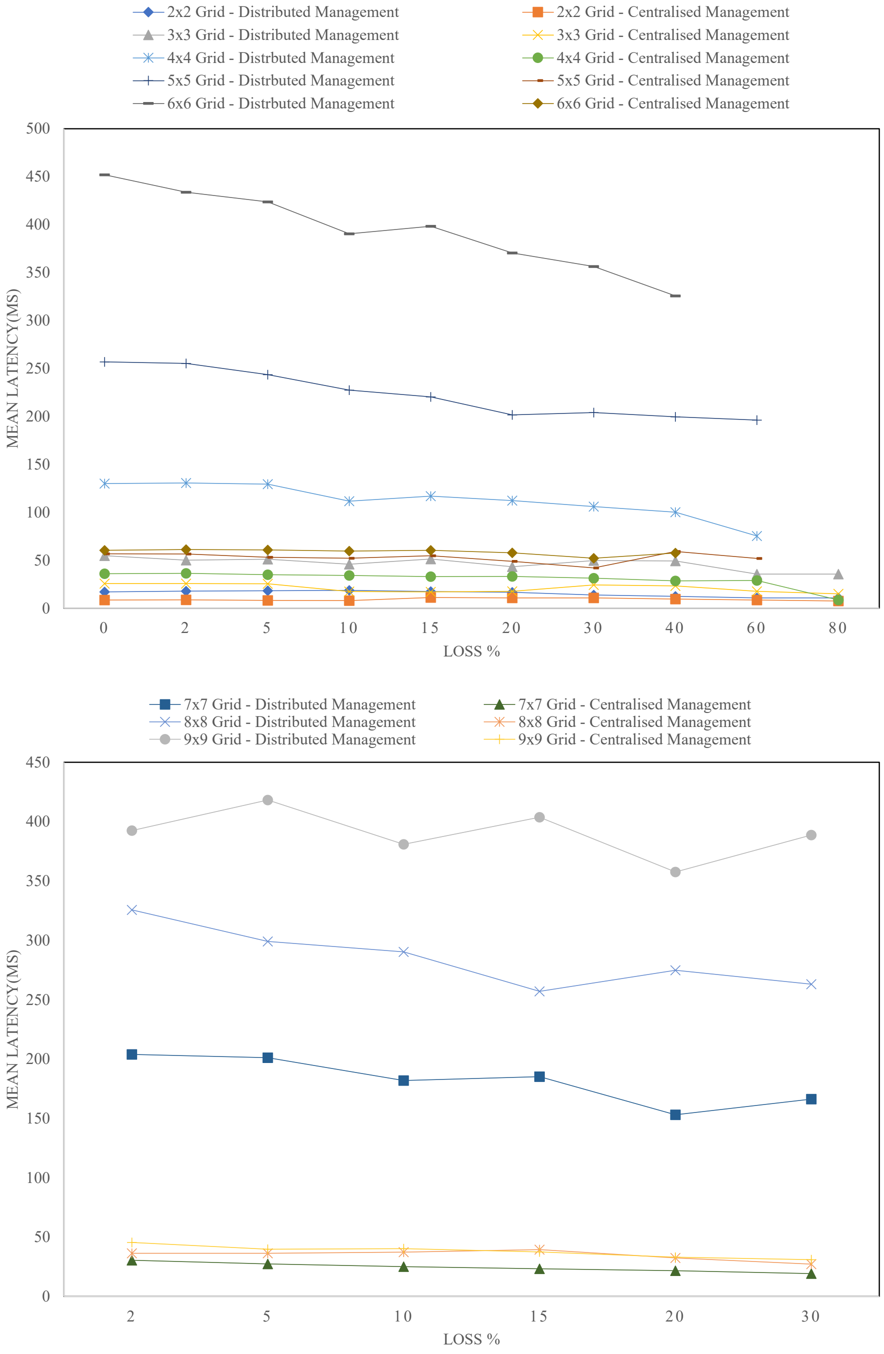

Fig. 7. Mean end-to-end Latency (in milliseconds) against varying simulated Packet Drop (in %)

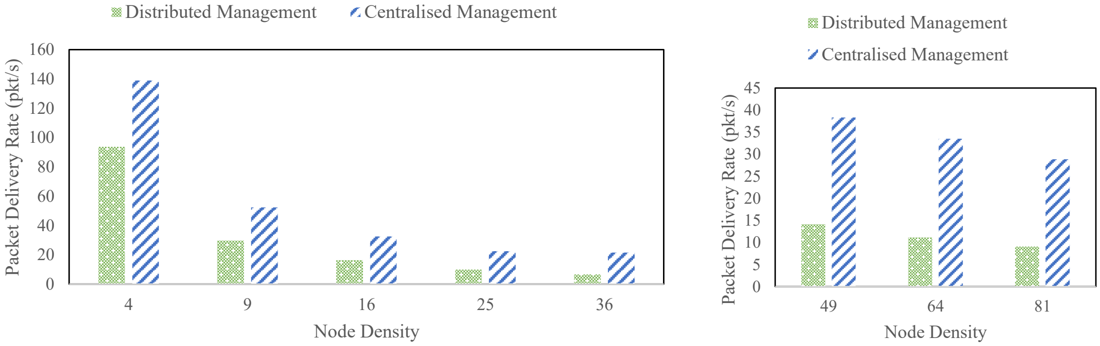

Fig. 8. Mean Packet Delivery Rate (pkt/s) varying network density (in total number of simulated nodes)

centralised management against varying network density shown in 6. When considering mid-sized networks along network density graph 8, longer observation period has counter-balanced the initial network convergence period where zero PDR could have been observed, leading to significantly higher PDR than $6x6$ networks. In PDR against loss graph 10, PDR has significant fluctuations under centralised management in small network when compared against mid-sized networks. This correlates to our conclusion in previous work [24] that the centralised management is more efficient and cost-effective, in terms of system resources, only in larger networks.

Mean Throughput Figures 9 and 11 gives the mean throughput (in KBps) between the core data-publisher and network-edge root-node subscriber. Throughput is a derived value by the product of mean packet size (in Bytes) and the PDR. The throughput experiences direct correlation to the observed PDR, after all throughput can be considered only with successful message transmissions. When viewing mid-sized networks along network density chart 9, elongated experimentation period has producing higher throughput negating the network convergence period in the beginning which has zero PDR. In mid-sized networks throughput graph 11 against the loss, the centralised management has lower throughput value than distributed management at loss $> 15\%$ given the multi-path efficiency or availability of redundant routes in distributed management

In conclusion, the performance summary can be read that the resource constrained WSN can benefit from versatile SDN based network management in larger networks. The observed improvements within single network domain can be translated into multi-domain networking through usage of SDN networking. The SDN networking eliminates the penalties of heterogeneous networking architecture, heterogeneous communication technologies and inconsistencies in networking.

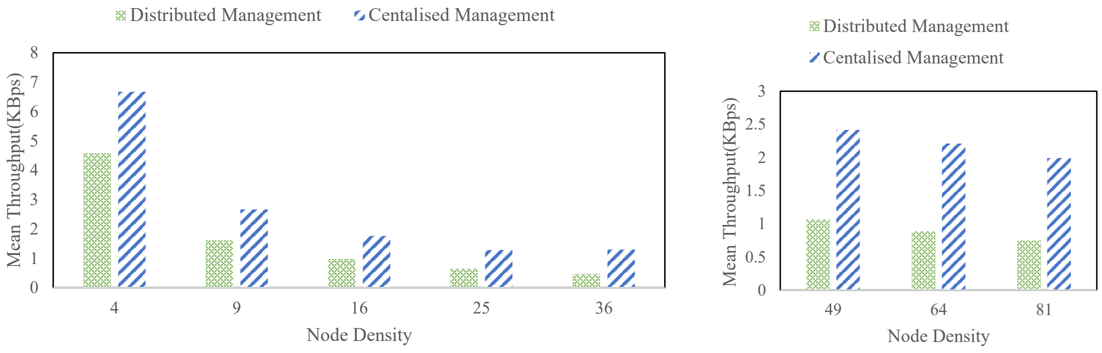

Fig. 9. Mean end-to-end Throughput in (KBps) against varying network density (in total number of simulated nodes)

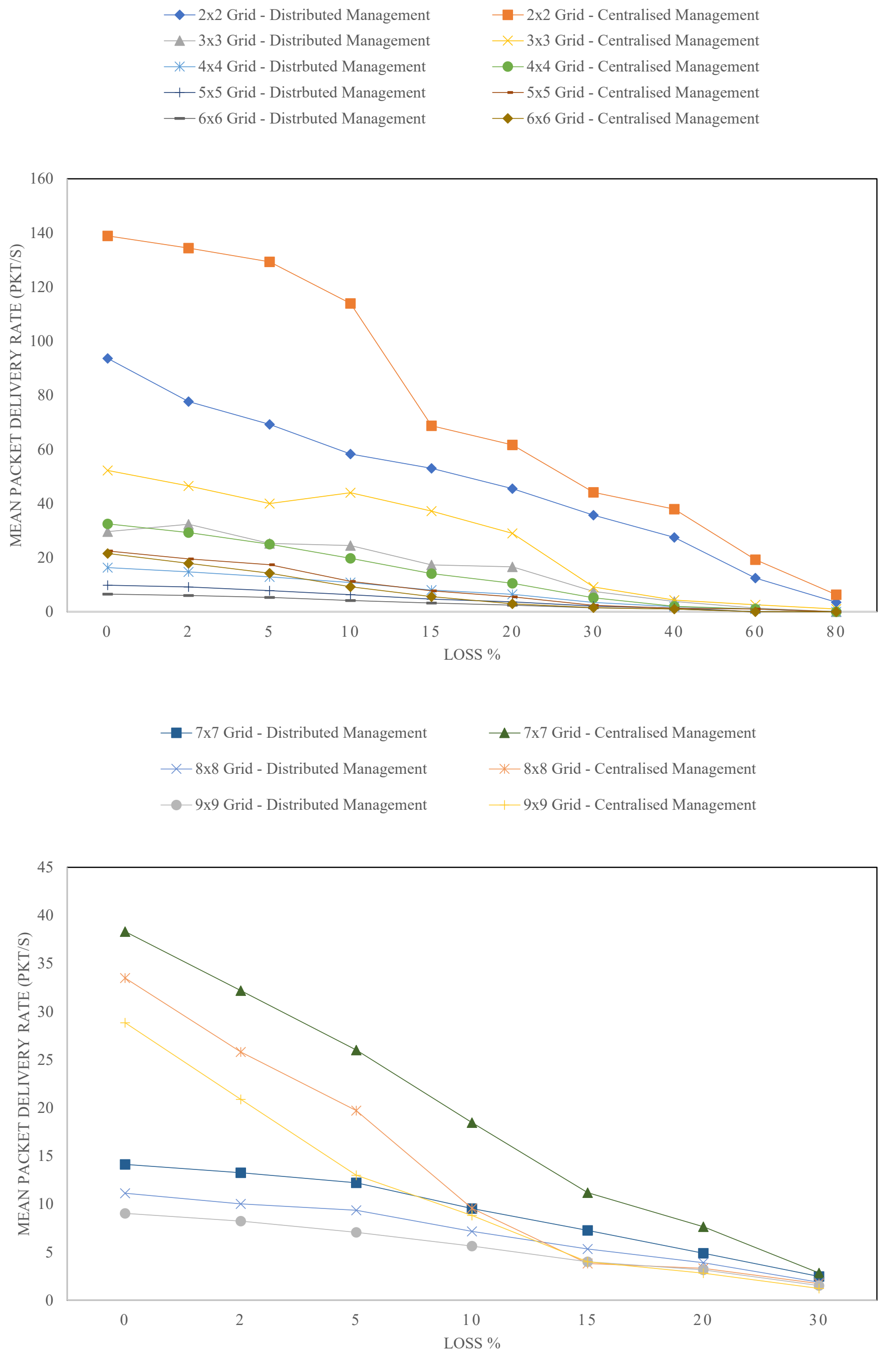

Fig. 10. Mean Packet Delivery Rate (pkt/s) against varying simulated Packet Drop (in %)

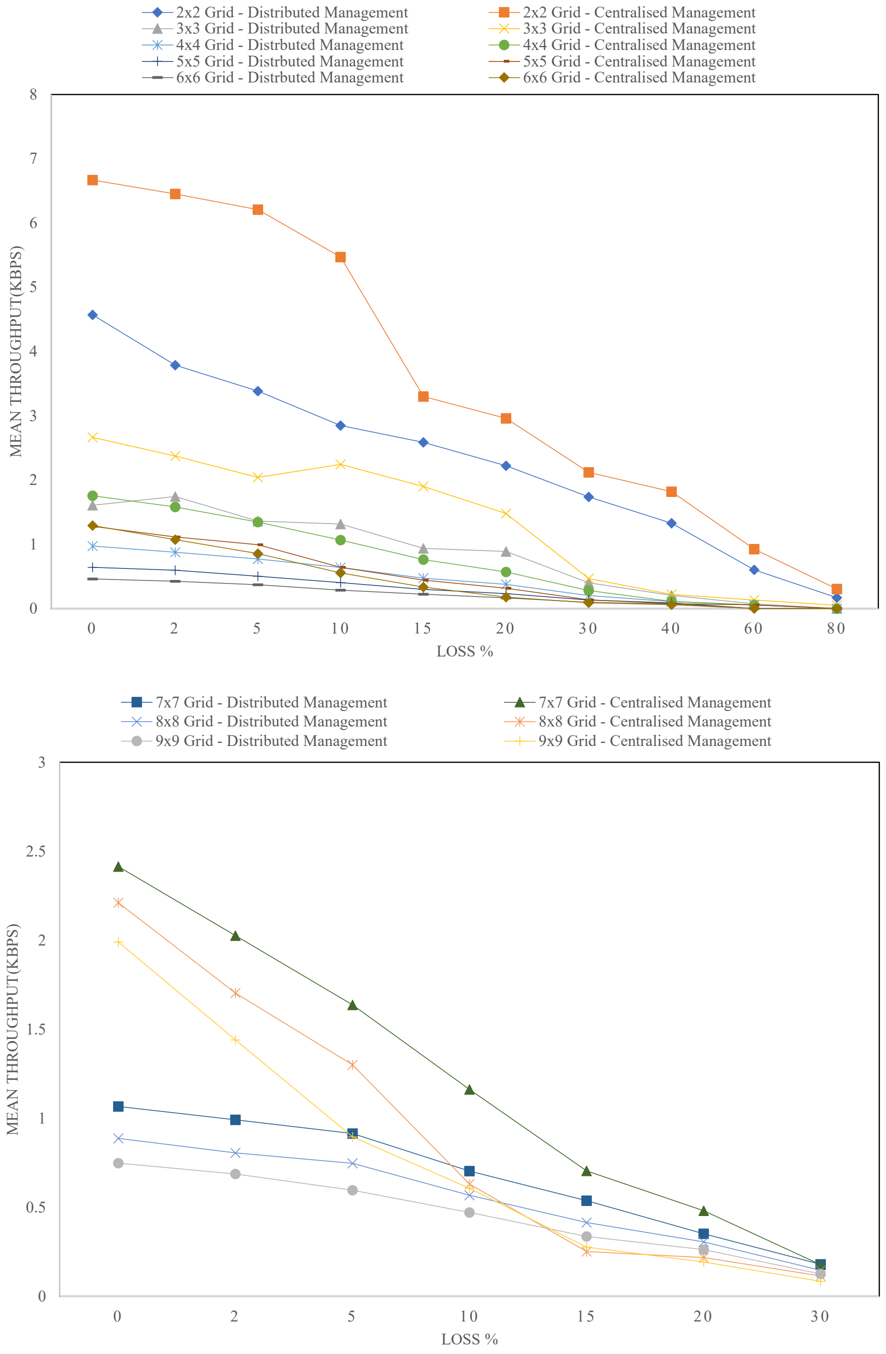

Fig. 11. Mean end-to-end Throughput in (KBps) against varying simulated Packet Drop (in %)

Table 1: Summary of Network Performance of CCN networking in WSN under presence and absence of SDN based virtual Network Management

Toplogy	**Total Number of nodes**	**Mode**	**Loss (%)**	**Mean Throughput (KBps)**	**Mean Latency (ms)**	**Mean Packet size (B)**	**Max Packet size (B)**	**Mean PDR (pkt/s)**	**Mean Hops to Root node**	**Min Hops to Root node**	**Max Hops to Root node**	**S.D. Hops to Root node**
2x2	4	mesh	0	3.071	17.026	32.803	41	93.627	1	1	4	1.5
2x2	4	mesh	2	2.545	17.854	32.776	41	77.633	1	1	4	1.5
2x2	4	mesh	5	2.275	18.182	32.869	41	69.2	1	1	4	1.5
2x2	4	mesh	10	1.914	18.514	32.834	41	58.283	1	1	4	1.5
2x2	4	mesh	15	1.739	17.358	32.83	41	52.967	1	1	4	1.5
2x2	4	mesh	20	1.492	16.552	32.809	41	45.483	1	1	4	1.5
2x2	4	mesh	30	1.165	13.782	32.648	41	35.683	1	1	4	1.5
2x2	4	mesh	40	0.889	12.256	32.372	41	27.45	1	1	4	1.5
2x2	4	mesh	60	0.4	10.754	32.119	38	12.467	1	1	3	1
2x2	4	mesh	80	0.112	10.653	32.015	35	3.483	1	1	2	0.5
2x2	4	sdn	0	4.445	8.582	32	32	138.917	1	1	1	0
2x2	4	sdn	2	4.301	8.649	32	32	134.407	1	1	1	0
2x2	4	sdn	5	4.138	8.082	32	32	129.322	1	1	1	0
2x2	4	sdn	10	3.646	7.896	32	32	113.933	1	1	1	0
2x2	4	sdn	15	2.2	11.066	32	32	68.75	1	1	1	0
2x2	4	sdn	20	1.972	10.635	32	32	61.633	1	1	1	0
2x2	4	sdn	30	1.412	10.802	32	32	44.117	1	1	1	0
2x2	4	sdn	40	1.212	9.503	32	32	37.867	1	1	1	0
2x2	4	sdn	60	0.616	8.472	32	32	19.237	1	1	1	0
2x2	4	sdn	80	0.202	7.564	32	32	6.317	1	1	1	0
3x3	9	mesh	0	1.133	54.674	38.226	50	29.633	3	2	7	1.5
3x3	9	mesh	2	1.225	50.052	37.92	47	32.317	3	2	6	1
3x3	9	mesh	5	0.958	51.085	37.951	50	25.233	3	2	7	1.5
3x3	9	mesh	10	0.924	45.865	37.754	47	24.483	3	2	6	1
3x3	9	mesh	15	0.657	51.302	37.971	47	17.305	3	2	6	1
3x3	9	mesh	20	0.622	43.262	37.489	47	16.583	3	2	6	1
3x3	9	mesh	30	0.279	49.578	37.263	50	7.492	3	2	7	1.5
3x3	9	mesh	40	0.142	49.276	37.296	47	3.81	3	2	6	1
3x3	9	mesh	60	0.048	35.538	35.682	38	1.355	2	2	3	0.5
3x3	9	mesh	80	0	35.496	35	35	0	2	2	2	0
3x3	9	sdn	0	1.828	25.685	35	35	52.233	2	2	2	0
3x3	9	sdn	2	1.628	25.658	35	35	46.517	2	2	2	0
3x3	9	sdn	5	1.4	25.465	35	35	40	2	2	2	0
3x3	9	sdn	10	1.539	17.299	35	35	43.983	2	2	2	0
3x3	9	sdn	15	1.303	17.076	35	35	37.217	2	2	2	0
3x3	9	sdn	20	1.014	17.661	35	35	28.967	2	2	2	0
3x3	9	sdn	30	0.32	24.19	35	35	9.133	2	2	2	0
3x3	9	sdn	40	0.15	23.321	35	35	4.283	2	2	2	0
3x3	9	sdn	60	0.089	17.593	35	35	2.549	2	2	2	0
3x3	9	sdn	80	0.035	15.019	35	35	1	2	2	2	0
4x4	16	mesh	0	0.711	130.007	43.589	56	16.317	5	3	9	1
4x4	16	mesh	2	0.64	130.571	43.413	56	14.75	5	3	9	1
4x4	16	mesh	5	0.563	129.309	43.655	56	12.898	5	3	9	1

Table 1 continued from previous page

Topology	**Total Number of nodes**	**Mode**	**Loss (%)**	**Mean Throughput (KBps)**	**Mean Latency (ms)**	**Mean Packet size (B)**	**Max Packet size (B)**	**Mean PDR (pkt/s)**	**Mean Hops to Root node**	**Min Hops to Root node**	**Max Hops to Root node**	**S.D. Hops to Root node**
4x4	16	mesh	10	0.463	111.536	42.872	53	10.8	5	3	8	0.5
4x4	16	mesh	15	0.344	116.774	43.094	65	7.983	5	3	12	2.5
4x4	16	mesh	20	0.272	112.193	42.422	56	6.417	4	3	9	2
4x4	16	mesh	30	0.146	105.946	42.17	53	3.456	4	3	8	1.5
4x4	16	mesh	40	0.077	100.151	42.5	50	1.82	4	3	7	1
4x4	16	mesh	60	0.039	75.373	39	41	1	3	3	4	0.5
4x4	16	mesh	80									
4x4	16	sdn	0	1.235	36.04	38	38	32.5	3	3	3	0
4x4	16	sdn	2	1.111	36.328	38	38	29.25	3	3	3	0
4x4	16	sdn	5	0.947	34.967	38	38	24.933	3	3	3	0
4x4	16	sdn	10	0.75	34.172	38	38	19.733	3	3	3	0
4x4	16	sdn	15	0.536	33.019	38	38	14.117	3	3	3	0
4x4	16	sdn	20	0.4	33.055	38	38	10.517	3	3	3	0
4x4	16	sdn	30	0.197	31.234	38	38	5.172	3	3	3	0
4x4	16	sdn	40	0.077	28.399	38	38	2.02	3	3	3	0
4x4	16	sdn	60	0.038	28.87	38	38	1	3	3	3	0
4x4	16	sdn	80	0	8.51	38	38	0	3	3	3	0
5x5	25	mesh	0	0.483	256.883	49.273	65	9.8	7	4	12	1
5x5	25	mesh	2	0.448	255.208	49.104	62	9.117	7	4	11	0.5
5x5	25	mesh	5	0.377	243.506	48.605	68	7.767	7	4	13	1.5
5x5	25	mesh	10	0.302	227.354	48.141	65	6.279	6	4	12	2
5x5	25	mesh	15	0.222	220.388	47.713	62	4.65	6	4	11	1.5
5x5	25	mesh	20	0.173	201.513	47.586	62	3.644	6	4	11	1.5
5x5	25	mesh	30	0.097	204.031	47.12	59	2.064	6	4	10	1
5x5	25	mesh	40	0.063	199.535	46.615	56	1.357	6	4	9	0.5
5x5	25	mesh	60	0	196.218	44	44	0	5	5	5	0
5x5	25	mesh	80									
5x5	25	sdn	0	0.92	56.723	41	41	22.443	4	4	4	0
5x5	25	sdn	2	0.8	56.448	41	41	19.517	4	4	4	0
5x5	25	sdn	5	0.713	53.03	41	41	17.383	4	4	4	0
5x5	25	sdn	10	0.461	52.065	41	41	11.233	4	4	4	0
5x5	25	sdn	15	0.318	54.756	41	41	7.75	4	4	4	0
5x5	25	sdn	20	0.226	48.948	41	41	5.508	4	4	4	0
5x5	25	sdn	30	0.097	42.039	41	41	2.37	4	4	4	0
5x5	25	sdn	40	0.053	59.228	41	41	1.286	4	4	4	0
5x5	25	sdn	60	0.041	51.855	41	41	1	4	4	4	0
5x5	25	sdn	80									
6x6	36	mesh	0	0.354	451.873	54.236	83	6.533	8	5	18	3.5
6x6	36	mesh	2	0.326	433.704	54.071	74	6.033	8	5	15	2
6x6	36	mesh	5	0.284	423.721	53.675	71	5.29	8	5	14	1.5
6x6	36	mesh	10	0.218	390.521	52.787	65	4.131	8	5	12	0.5
6x6	36	mesh	15	0.172	398.257	53.106	68	3.23	8	5	13	1
6x6	36	mesh	20	0.126	370.458	52.297	65	2.4	8	5	12	0.5
6x6	36	mesh	30	0.074	356.258	52.35	68	1.405	8	5	13	1
6x6	36	mesh	40	0.051	325.759	50.643	59	1	7	5	10	0.5
6x6	36	mesh	60									

Table 1 continued from previous page

Toplogy	Total Number of nodes	Mode	Loss (%)	Mean Throughput (KBps)	Mean Latency (ms)	Mean Packet size (B)	Max Packet size (B)	Mean PDR (pkt/s)	Mean Hops to Root node	Min Hops to Root node	Max Hops to Root node	S.D. Hops to Root node
6x6	36	mesh	80									
6x6	36	sdn	0	0.947	60.443	44	44	21.533	5	5	5	0
6x6	36	sdn	2	0.786	61.075	44	44	17.869	5	5	5	0
6x6	36	sdn	5	0.626	60.804	44	44	14.233	5	5	5	0
6x6	36	sdn	10	0.406	59.557	44	44	9.23	5	5	5	0
6x6	36	sdn	15	0.244	60.416	44	44	5.55	5	5	5	0
6x6	36	sdn	20	0.128	57.684	44	44	2.914	5	5	5	0
6x6	36	sdn	30	0.065	52.139	44	44	1.467	5	5	5	0
6x6	36	sdn	40	0.044	57.472	44	44	1	5	5	5	0
6x6	36	sdn	60									
6x6	36	sdn	80									
7x7	49	mesh	0	0.842	220.534	59.632	95	14.113	10	6	22	4
7x7	49	mesh	2	0.78	203.947	58.795	95	13.262	10	6	22	4
7x7	49	mesh	5	0.719	201.179	58.95	89	12.203	10	6	20	3
7x7	49	mesh	10	0.551	181.864	57.849	83	9.528	10	6	18	2
7x7	49	mesh	15	0.422	185.228	58.022	89	7.265	10	6	20	3
7x7	49	mesh	20	0.274	153.116	56.07	77	4.883	9	6	16	2
7x7	49	mesh	30	0.142	166.194	57.359	71	2.467	9	7	14	1.5
7x7	49	sdn	0	1.801	30.88	47	47	38.326	6	6	6	0
7x7	49	sdn	2	1.512	30.522	47	47	32.177	6	6	6	0
7x7	49	sdn	5	1.222	27.372	47	47	26.007	6	6	6	0
7x7	49	sdn	10	0.868	25.076	47	47	18.468	6	6	6	0
7x7	49	sdn	15	0.526	23.273	47	47	11.183	6	6	6	0
7x7	49	sdn	20	0.359	21.715	47	47	7.645	6	6	6	0
7x7	49	sdn	30	0.133	19.244	47	47	2.825	6	6	6	0
8x8	64	mesh	0	0.71	308.014	63.933	98	11.113	12	7	23	3
8x8	64	mesh	2	0.645	325.585	64.373	107	10.023	12	7	26	4.5
8x8	64	mesh	5	0.598	299.126	63.967	98	9.354	12	7	23	3
8x8	64	mesh	10	0.453	290.228	63.25	98	7.166	11	7	23	4
8x8	64	mesh	15	0.33	256.985	62.024	92	5.321	11	7	21	3
8x8	64	mesh	20	0.244	274.882	62.688	92	3.899	11	7	21	3
8x8	64	mesh	30	0.117	263.019	62.37	83	1.879	11	7	18	1.5
8x8	64	sdn	0	1.675	36.986	50	50	33.498	7	7	7	0
8x8	64	sdn	2	1.291	36.415	50	50	25.811	7	7	7	0
8x8	64	sdn	5	0.986	36.311	50	50	19.717	7	7	7	0
8x8	64	sdn	10	0.478	37.339	50	50	9.563	7	7	7	0
8x8	64	sdn	15	0.191	39.455	50	50	3.816	7	7	7	0
8x8	64	sdn	20	0.166	32.468	50	50	3.317	7	7	7	0
8x8	64	sdn	30	0.087	27.195	50	50	1.735	7	7	7	0
9x9	81	mesh	0	0.605	345.467	66.966	95	9.03	13	8	22	2
9x9	81	mesh	2	0.557	392.436	67.774	104	8.218	13	8	25	3.5
9x9	81	mesh	5	0.484	418.32	68.507	104	7.066	13	9	25	4
9x9	81	mesh	10	0.382	380.849	67.793	101	5.63	13	8	24	3
9x9	81	mesh	15	0.273	403.681	68.081	92	4.007	13	8	21	1.5
9x9	81	mesh	20	0.212	357.621	67.441	95	3.147	13	8	22	2
9x9	81	mesh	30	0.102	388.776	66.886	89	1.518	13	8	20	1

Table 1 continued from previous page

Topology	**Total Number of nodes**	**Mode**	**Loss (%)**	**Mean Throughput (KBps)**	**Mean Latency (ms)**	**Mean Packet size (B)**	**Max Packet size (B)**	**Mean PDR (pkt/s)**	**Mean Hops to Root node**	**Min Hops to Root node**	**Max Hops to Root node**	**S.D. Hops to Root node**
9x9	81	sdn	0	1.529	45.951	53	53	28.849	8	8	8	0
9x9	81	sdn	2	1.107	45.488	53	53	20.882	8	8	8	0
9x9	81	sdn	5	0.689	39.846	53	53	13	8	8	8	0
9x9	81	sdn	10	0.467	40.305	53	53	8.806	8	8	8	0
9x9	81	sdn	15	0.212	37.563	53	53	4.008	8	8	8	0
9x9	81	sdn	20	0.149	33.076	53	53	2.81	8	8	8	0
9x9	81	sdn	30	0.065	30.991	53	53	1.226	8	8	8	0

6 Conclusion

This technical report summaries the network performance of SDN-based virtual network management in CCN networking for WSN and analyses the networking aspects of the CCN networking in WSN has been analysed in the technical document. It details the background on the issues of interoperable coexistence between IP and CCN networking in future IoT networks. The design factors and motivation for SDN in WSN has been thoroughly analysed with emphasis for unified approach in network management for heterogeneous wireless networks in future IoT networks, in continuation of our prior works. Further, the link-level communication aspects including radio coexistence, link-layer heterogeneity and type of MAC – TDMA or CSMA, must be included to add better scoping of the evaluation results. Our prior work deals with these primary aspects in heterogeneous wireless networks in IoT. Through the evaluation, we could observe the significant improvement in performance of throughput, latency and routing under SDN based network management.

References

1. MQTT. home page. https://mqtt.org, [Online; accessed 28-April-2020]
2. Anadiotis, A.G., Milardo, S., Morabito, G., Palazzo, S.: Toward Unified Control of Networks of Switches and Sensors Through a Network Operating System. IEEE Internet of Things Journal 5(2), 895–904 (2018)
3. Baddeley, M., Nejabati, R., Oikonomou, G., Sooriyabandara, M., Simeonidou, D.: Evolving SDN for low-power IoT networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). pp. 71–79. IEEE (2018)
4. Barcelo, M., Correa, A., Vicario], J.L., Morell, A.: Cooperative interaction among multiple RPL instances in Wireless Sensor Networks. Computer Communications 81, 61 – 71 (2016)
5. Bello, O., Zeadally, S., Badra, M.: Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). Ad Hoc Networks 57, 52–62 (2017)
6. Bera, S., Misra, S., Roy, S.K., Obaidat, M.S.: Soft-WSN: Software-Defined WSN Management System for IoT Applications. IEEE Systems Journal 12(3), 2074–2081 (2018)
7. Chen, B., Liu, L., Zhang, Z., Yang, W., Ma, H.: Brr-cvr: A collaborative caching strategy for information-centric wireless sensor networks. In: 2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). pp. 31–38 (2016)
8. Enguehard, M., Droms, R., Rossi, D.: Slict: Secure localized information centric things. In: Proceedings of the 3rd ACM Conference on Information-Centric Networking. p. 255–260. ACM-ICN '16, Association for Computing Machinery, New York, NY, USA (2016), https://doi.org/10.1145/2984356.2988519
9. Gaddour, O., Koubâa, A.: RPL in a nutshell: A survey. Computer Networks 56(14), 3163–3178 (2012)
10. Galluccio, L., Milardo, S., Morabito, G., Palazzo, S.: SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM). pp. 513–521. IEEE (2015)
11. Hui, J., Thubert, P.: RFC 6282: Compression format for IPv6 datagrams over IEEE 802.15.4-based networks. Internet Engineering Task Force (2011)
12. Hung, L.., Leu, F.., Tsai, K.., Ko, C..: Energy-Efficient Cooperative Routing Scheme for Heterogeneous Wireless Sensor Networks. IEEE Access 8, 56321–56332 (2020)
13. Khaled, A.E., Helal, S.: Interoperable communication framework for bridging restful and topic-based communication in iot. Future Generation Computer Systems 92, 628 – 643 (2019)
14. Ko, J., Jeong, J., Park, J., Jun, J.A., Gnawali, O., Paek, J.: DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network. ACM Trans. Sen. Netw. 11(2) (Mar 2015)
15. Le, Q., Ngo-Quynh, T., Magedanz, T.: RPL-based multipath Routing Protocols for Internet of Things on Wireless Sensor Networks. In: 2014 International Conference on ATC 2014. pp. 424–429 (2014)

16. Li, H., Ota, K., Dong, M.: LS-SDV: Virtual Network Management in Large-scale Software-Defined IoT. IEEE Journal on Selected Areas in Communications 37(8), 1783–1793 (2019)
17. Li, Y., Su, X., Lindgren, A., Shi, X., Cai, X., Riekki, J., Que, X.: Distance assisted information dissemination with broadcast suppression for icn-based vanet. In: International Conference on Internet of Vehicles. pp. 179–193. Springer (2016)
18. Moiseenko, I., Oran, D.: Tcp/icn: Carrying tcp over content centric and named data networks. In: Proceedings of the 3rd ACM Conference on Information-Centric Networking. pp. 112–121. ACM (2016)
19. Ndiaye, M., Abu-Mahfouz, A.M., Hancke, G.P.: SDNMM—A Generic SDN-Based Modular Management System for Wireless Sensor Networks. IEEE Systems Journal (2019)
20. Nour, B., Sharif, K., Li, F., Biswas, S., Moungla, H., Guizani, M., Wang, Y.: A survey of internet of things communication using icn: A use case perspective. Computer Communications 142, 95–123 (2019)
21. Quevedo, J., Corujo, D., Aguiar, R.: Consumer driven information freshness approach for content centric networking. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 482–487 (2014)
22. Rehman, M.A.U., Ullah, R., Kim, B.S., Nour, B., Mastorakis, S.: Ccic-wsn: An architecture for single-channel cluster-based information-centric wireless sensor networks. IEEE Internet of Things Journal 8(9), 7661–7675 (2021)
23. Saint-Andre, P., et al.: Extensible messaging and presence protocol (xmpp): Core (2004)
24. Selvaraju, S.P., Balador, A., Fotouhi, H., Vahabi, M., Björkman, M.: Network management in heterogeneous iot networks. In: IWCMC'21-17th International Wireless Communications & Mobile Computing Conference (2021)
25. Sharma, D., Goap, A., Shukla, A.K., Priyanka, Bhondekar, A.P.: Traffic Heterogeneity Analysis in an Energy Heterogeneous WSN Routing Algorithm. In: Proceedings of 2nd International Conference on Communication, Computing and Networking. pp. 335–343. Springer Singapore, Singapore (2019)
26. Siever, E., Weber, A., Figgins, S., Love, R., Robbins, A.: Linux in a Nutshell. " O'Reilly Media, Inc." (2005)
27. Tanganelli, G., Virdis, A., Mingozzi, E.: Implementation of Software-Defined 6LoWPANs in Contiki OS. In: 2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). pp. 1–6. IEEE (2019)
28. Vaiyapuri, T., Parvathy, V.S., Manikandan, V., Krishnaraj, N., Gupta, D., Shankar, K.: A novel hybrid optimization for cluster-based routing protocol in information-centric wireless sensor networks for iot based mobile edge computing. Wireless Personal Communications pp. 1–24 (2021)
29. Wibowo, F.X., Gregory, M.A., Ahmed, K., Gomez, K.M.: Multi-domain software defined networking: research status and challenges. Journal of Network and Computer Applications 87, 32–45 (2017)