A Hybrid Software Defined Networking Architecture for Next-Generation IoTs

Ahyoung Lee[a*], Xuan Wang[a], Hieu Nguyen[b] and Ilkyeun Ra[b]
[a]Department of Computer Science, University of South Dakota
Vermillion, SD 57069, USA
[b]Department of Computer Science and Engineering, University of Colorado Denver
Denver, CO 80217, USA
[e-mail : ahyoung.lee@usd.edu, xuan.wang@usd.edu, hieu.nguyen@cudenver.edu, ilkyeun.ra@ucdenver.edu
*Corresponding author: Ahyoung Lee

Received October 5, 2017; accepted December 28, 2017; published February 28, 2018

Abstract

Everything in the world is becoming connected and interactive due to the Internet. The future of interactive smart environments such as smart cities, smart industries, or smart farms demand high network bandwidth, high network flexibility, and self-organization systems without costly hardware upgrades, and they provide a sustainable, scalable, and replicable smart environment backbone infrastructure. This paper presents a new Hybrid Software-Defined architecture for integrating Internet-of-Things technologies that are essential technologies for smart environments. It combines a software-defined networking infrastructure and a real-time distributed network framework with an advanced optimization to enable self-configuration, self-management, and self-adaption for providing seamless communication and efficiently managing a vast number of smart heterogeneous devices.

Keywords: Internet of Things, Software Defined Networking, Hybrid-SDN architecture

A preliminary version of this paper was presented at APIC-IST 2017, and was selected as an outstanding paper.

http://doi.org/10.3837/tiis.2018.02.024 **ISSN : 1976-7277**

1. Introduction

Over the last 15 years, electronic devices have become cheaper, smaller and more available enabling automation in every aspect of life and creating a network of interconnected smart objects which is commonly referred as the Internet-of-Things (IoT). In the era of IoT, large number of things (objects) are efficiently interconnected through the Internet. The IoT represents the next evolution of the Internet as a Network of Networks that is made up of a loose collection of disparate, purpose-built networks. To obtain information by sensing, collecting, transmitting, analyzing and distributing data on a massive scale objects, the next evolution of the IoT will require control systems for managing multiple networks. Indeed, IoT researchers point out that "things or objects" (such as 50 billion devices) will be significantly more connected to the Internet than people by 2020 [1] shown in **Fig. 1**. The evolution of the IoT must consider such issues as how to integrate and interoperate disparate networks and a multitude of heterogeneous sensor devices. Moreover, the communication requirements to support IoTs (or Smart environments) are uniquely different from current networks in terms of various types and sizes of packet data, a variety of bandwidth requirements from many applications and vastly different levels of time and error tolerance.

Software Defined Networking (SDN) is an emerging networking paradigm that separates the network control plane from the data forwarding plane, which provides user applications with a centralized view of the distributed network states. This SDN promises to improve network resource utilization, simplify network management, and ease the interoperability of the heterogeneous devices from different manufactures. SDN was recently introduced primarily for data center networks [2] and was later extended to support a variety of wireless networks such as wireless sensor or 5G systems [3] for the next-generation Internet. The main ideas are to separate the data plane from the control plane and introduce novel network control functionalities based on an abstract representation of the network.

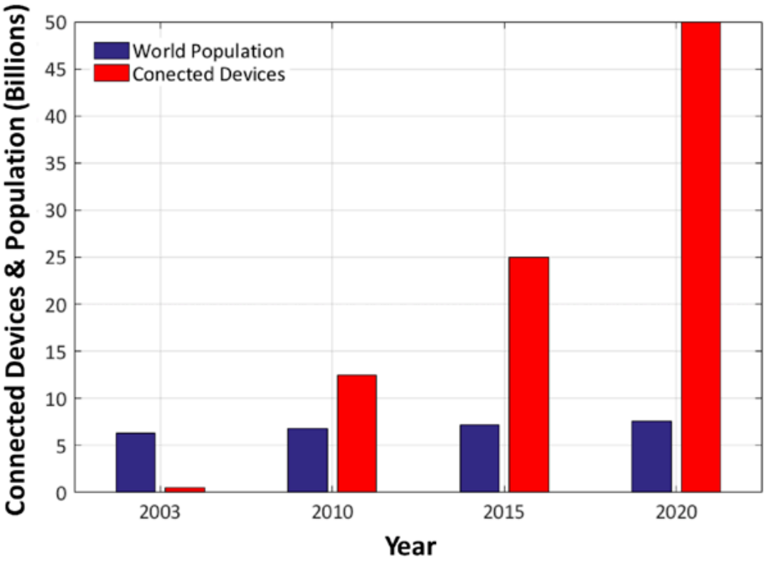


Fig. 1. The number of IoT connected devices compared with world population [1].

SDN makes it easier to introduce and deploy new applications and services rather than the classical hardware-dependent standards. This SDN promises to improve network resource utilization, simplify network management, and ease the interoperability of the heterogeneous devices from different manufactures [11]. We believe an SDN paradigm can meet the needs of current and next generation IoT requirements for heterogeneity and flexibility. In SDN, the network intelligence and state are handled by either one centralized controller for a small-scale network or many centralized controllers for a large-scale or wide area networks. However, the centralized controller architecture in SDN faces such challenges and limitations as scalability, availability, reliability, and deplorability [12]. These fundamental issues make the deployment of SDN limited to the small-scale networks, allowing less flexibility for data communication from outside of SDNs.

To solve the problems, we propose a new hybrid-SDN architecture that combines Software Defined Network and Distributed Network for supporting next-generation Internet-of-Things. In our proposed hybrid-SDN architecture, there are OpenFlow (OF)-enable switches that enable to support different traffic services in order to incorporate with flow specifications between legacy networks and SDN networks. We studied the network performance difference between three different network architectures: distributed network (DN), software defined network (SDN), and hybrid-software defined network (hybrid-SDN), by network simulation using Mininet [13] and Floodlight SDN controller [14] with OpenvSwitch. We presented our experimental results that were collected by our network simulation for three different network architectures with two different network topologies as COST266 and FRANCE introduced in [15]. Our simulation results suggested a hybrid-SDN can maintain very similar performance to SDN. Thus, our proposed hybrid-SDN integrates a software-defined networking infrastructure and a real-time distributed network framework with advanced optimization to enable self-configuration, self-management, and self-adaption for efficiently managing a vast number of heterogeneous connected things. The remainder of this paper is organized as follows. Section 2 presents briefly review related other research works. Section 3 describes three different network models: DN, SDN and hybrid-SDN. Section 4 gives the performance evaluation of three network models. Section 5 gives the summary of our contributions and the discussion of our future works.

2. Related Work

In SDN models, many researches have focused on the scalability problem of SDN models. Kandoo [4] has a two-level hierarchy for controllers such as local controllers and a logically centralized root control. ElastiCon [5] is an elastic approach to change the number of controllers dynamically under different conditions. Orion [6] is a hybrid hierarchical control plane architecture to solve the path stretch problem. These approaches try to solve a large traffic load problem in SDN networks using hierarchical controller deployment or hybrid distributed hierarchical controller deployment. However, they only consider a flow management of intra-area or/and inter-area links in networks based on pure SDN paradigm.

In this paper, we consider hybrid-SDN models that combine SDN based on centralized control approaches with traditional networks and distributed control approaches. Modern Internet consists of various types of network devices and protocols through traditional and SDN networks. Yet, Internet should be able to provide seamless communication between distributed networks and SDN enabled networks. Most of majority of SDN hardware and software vendors have been trying to support gradual migration from traditional network to SDN without rapidly ripping off current existing non-confirming SDN devices and protocols.

To support smooth and efficient migration from traditional computer networks to software defined network, several research efforts have been proposed to facilitate from traditional networks to SDN networks. HybNET [7] is a network management framework for hybrid OpenFlow-legacy networks. It provides a centralized configuration interface and visualization across from OpenFlow switches to legacy devices. Cardigan [8] is a distributed router using OpenFlow and deployed it at a public Internet exchange. It provides a distributed router based on RouterFlow [9] to reduce the operational complexity of maintaining IP routers through OpenFlow switches in the network. OSHI [10] is an open source hybrid IP/SDN network using Quagga routing software suite. It provides a hybrid IP/SDN device as a router/switch node that can operate both at IP level and SDN level within the same domain.

However, those approaches do not fully support different traffic services in their hybrid-SDN models. According to the research [12], in the service-based hybrid SDN model, there are existing both DN and SDN, and they provide different traffic services. The two paradigms can span an overlapping set of nodes and control a different portion of the forwarding information base (FIB) of each node. In this case, hybrid-SDN models may be willing to keep proved the traditional IP routing protocols and technologies for some traffic services, such as VPNs and IPSec encrypted connections, instead of relying on new software to be integrated in their SDN controllers. Thus, our proposed hybrid-SDN architecture provides OF-enable switches for guaranteeing both IP-based traffic services from traditional distributed networks and OF-based services from SDN networks.

3. Proposed Hybrid-SDN Architecture

In this section, we discuss the overview of network models with research challenging issues, and we present the design and operation of the proposed hybrid-SDN architecture.

3.1 Overview of Network Models

Early network devices where network control and data planes are integrated into each device and the integrated network planes are distributed in a given domain to perform an autonomous and independent manner to the greatest extent possible. The data plan is a network functionality including QoS, access control list (ACL), routing, forwarding decisions, etc.

However, this integrated and distributed network model became increasingly dysfunctional due to the scale and complexity of networks growth. Moreover, there are main limitations: (1) closed architecture as vendor dependency that proprietary hardware and software architecture limit innovation and interoperability, (2) myriad of protocols that may different protocols, devices and configuration increase maintenance overhead, and (3) distributed control plane that the limited view of the network leads a limited set of decisions to optimize the performance. We refer to this network design and management model as Distributed Traditional Networking (DN), and thus, the DN model is not appropriate for the next evolution of IoTs.

Software Defined Network (SDN) model can meet the need of current and future IoT requirements of heterogeneity and flexibility. Because SDN has a centralized traffic engineering system that can handle massive scale of IoT objects from multiple networks much more efficiently and intelligently compared to the conventional approaches as IP-based traffic engineering (such as the sortest path routing). More specifically, the benefities of SDN are (1) centralized visibility including global network information (e.g., dynamically changing the network status) and global application information (e.g., QoS requirements), (2) the programmability without having to handle individual infrastructure elements, (3) openness, where data plane elements have a unified interface open to the controller for data plane programming and network status collection, and (4) multiple flow table pipelines in OF switches can make flow management more flexible and efficient [11].

However, SDN based on the centralized manner has research challenges and limitations such as scalability, availability, reliability and deployability. Thus, the critical circumstance is currently contributing to limit the deployment of SDN within a small-scale network [12]. So that, we introduce a new hybrid-SDN architecture that integrates a software-defined networking infrastructure and a real-time distributed network computing framework with advanced optimization to enable self-configuration, self-management and self-adaption for efficiently managing a vast number of IoT heterogeneous devices. At this time, we only provide a simple hybrid-SDN model presented in **Fig. 2**.

Some possible hybrid SDN models are presented in [12]: (1) the topology-based hybrid SDN model is a traditional hybrid concept architecture, such as using BGP between SDN and DN network zones at edge nodes, (2) the service-based hybrid SDN model mainly consists of hybrid SDN nodes, such as traffic flow services maintained by hybrid SDN nodes, (3) the class-based hybrid SDN model mainly consists of hybrid-SDN controllers, and (4) the integrated hybrid SDN model has all interfaces over DN and SDN nodes, such as the FIB of any node controlled by SDN controllers. The basic design of our hybrid-SDN model is based on the service-based hybrid SDN, because it is much less complex model than the other modes especially compared to the class-based hybrid SDN and it is appropriate to support a service-based transition between DN and SDN network zones.

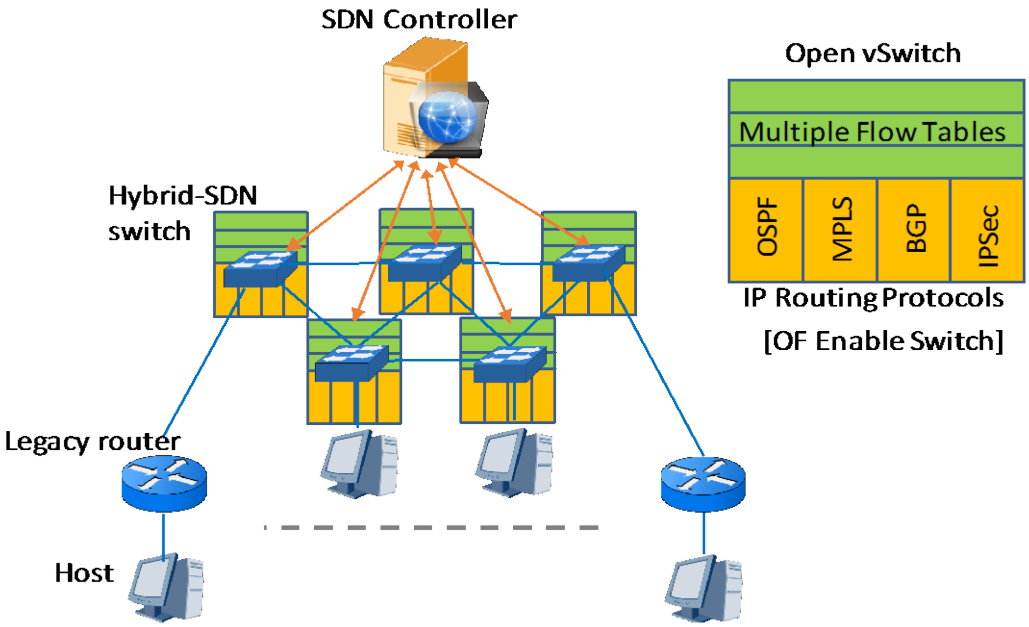


Fig. 2. A hybrid-SDN architecture model

Hence, our proposed hybrid-SDN model is willing to keep proved the traditional IP routing protocols and technologies for traffic flow QoS requirements, instead of relying on new software to be integrated in their SDN controllers. Therefore, our hybrid-SDN architecture provides OF-enable switches for guaranteeing both IP-based traffic services from traditional networks and OF-based services from SDN networks.

3.2 Design and Operation of our Hybrid-SDN Model

In our proposed hybrid-SDN model, there are three kinds of nodes - SDN controller, hybrid-SDN switch, and legacy router/switch. Legacy routers/switches run traditional network protocol stack. Hybrid-SDN switches directly connected to the SDN controller run the both IP routing-based protocol stack in DN and OF forwarding-based stack in SDN. Thus, our proposed hybrid-SDN switch nodes are different than other SDN switch nodes for the service-based hybrid SDN. Because usually SDN switch nodes are used for a legacy network when the nodes are idle in hybrid-SDN models.

All nodes in DN support a shortest path routing as Open Shortest Path First (OSPF) protocol while all nodes in SDN forward data-packet flows based on their multiple flow tables that are updated by the SDN controller. If a flow path between DN and SDN nodes, packets of the flow are forwarded through the shortest path in DN, and if the packets of the flow are in SDN they are managed by either the multiple flow tables built on the Ternary Content Addressable Memory (TCAM) or the routing tables built on the Static Random-Access Memory (SRAM) at each OF-enable switch for depending on their flow specification requirements (e.g., OSPF, MPLS, BGP, IPSec). The purpose of TCAM is a very high speed search for data-plane forwarding entries and it has very limited scalability with small-scale flows in SDN networks. The purpose of SRAM is high speed search for data-plane routing entries for large-scale flows in legacy networks [11]. In order to make flow management more flexible and efficient, therefore, the hybrid-SDN architecture will provide better solutions to the next evolution of IoTs. The more detailed operations are shown in **Fig. 3**.

The basic concept of the hybrid forwarding table suggested in [16] was adapted into our OF-enable switch for managing different traffic follow services in the hybrid-SDN network. **Fig. 3** shows the operation of flow management in the hybrid forwarding table of an OF-enable switch that is a hybrid-SDN switch in our hybrid-SDN network. The OF-enable switch comprises of a small TCAM table and a large SRAM table. The TCAM table is used for OF flows, thus it has OF forwarding entries. The SRAM table is used to store all the source-destination routing forwarding entries.

The basic operation of flow management is as follows when a flow arrives at an OF-enable switch: (1) the OF-enable switch checks that the flow has a specific requirement under IP routing protocols came from DN nodes, called the legacy flow, (2) if the flow is legacy flow, then selects a routing entry from the SRAM table, (3) otherwise the flow is the SDN flow, then selects a forwarding entry from the TCAM table, and (4) all subsequent packets in the flow are forwarded in the data plane along the path.

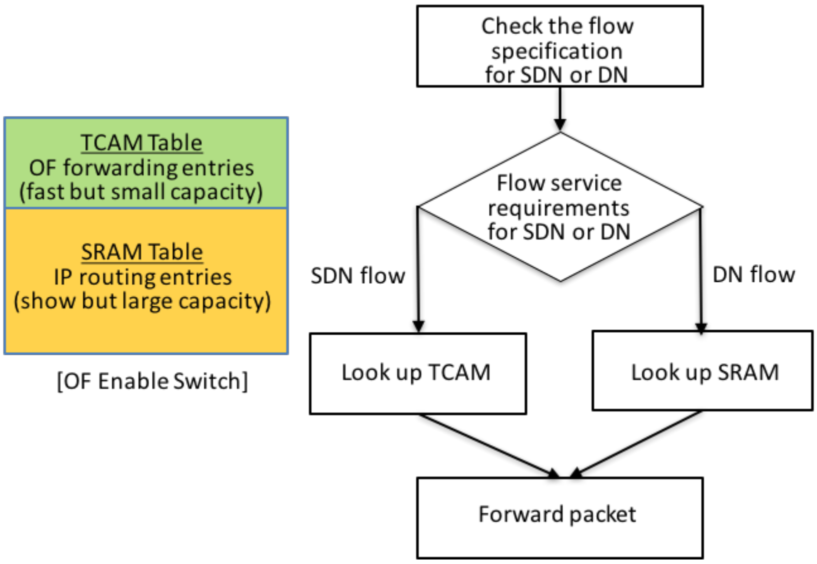

TCAM Table
OF forwarding entries
(fast but small capacity)
SRAM Table
IP routing entries
(show but large capacity)
[OF Enable Switch]
Check the flow specification for SDN or DN
Flow service requirements for SDN or DN
SDN flow
DN flow
Look up TCAM
Look up SRAM
Forward packet

Fig. 3. Operation of flow management in the hybrid forwarding tables

4. Experimental Evaluation

In this section, we analyze some performance of the OF-enable switch implementation over the hybrid-SDN testbeds.

4.1 Experimental Setups

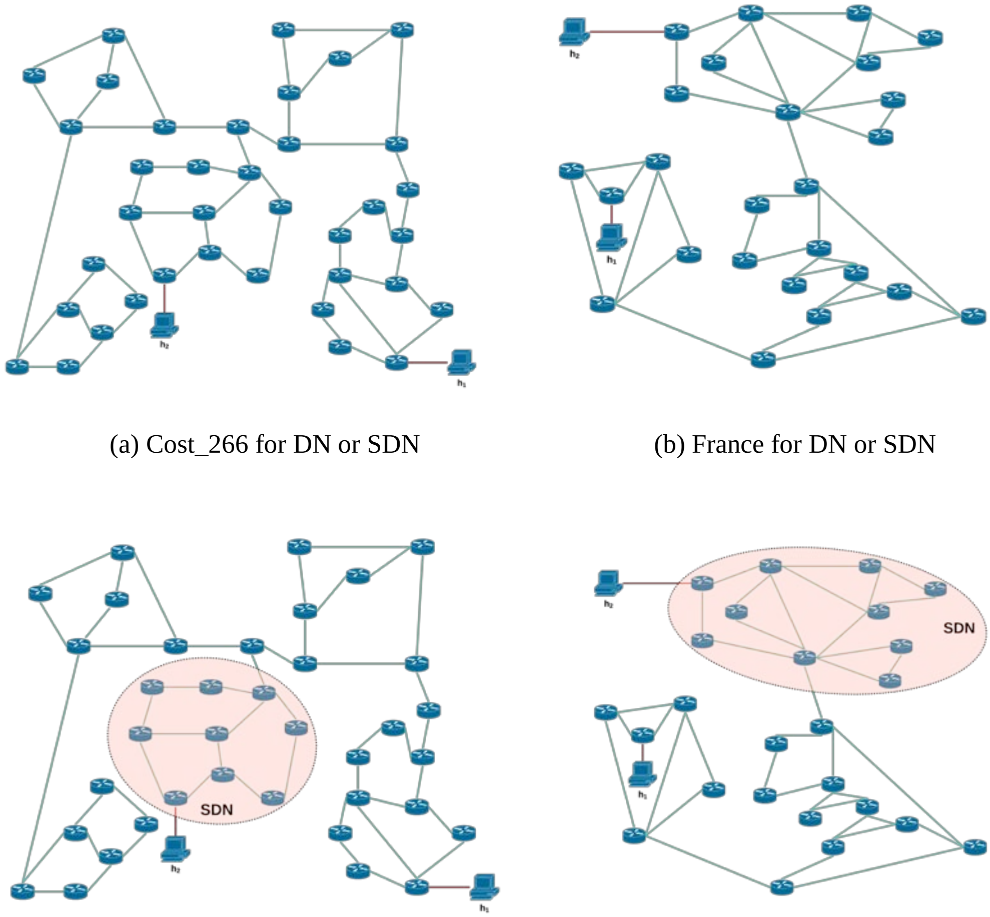
We present experimental performance results with the comparison of DN, SDN and hybrid-SDN network models. We used two network topologies – Cost_266 and France based on [15] to evaluate each network model. Cost_266 topology which is the advanced infrastructure for Photonic networks [17] comprises 37 nodes and 57 links, and France network topology which is WDM planning data provided by a network operator comprises 25 nodes and 45 links. These two topologies with each network model (DN, SDN, hybrid-SDN) were emulated using Mininet [13], and we used Floodlight controller [14] to control SDN-enabled switches (OpenvSwitch) for both SDN and hybrid-SDN.

Fig. 4 shows four cases of network testbeds. The (a) Cost_266 and (b) France topologies are used for DN and SDN. All nodes are OF switches in (a) Cost_266 and (b) France when we test the SDN network, and to test the DN network all nodes are just legacy IP-routers or switches in (a) Cost_266 and (b) France topologies. The (c) Hybrid-Cost_266 and (d) Hybrid_France topologies are used for the hybrid-SDN network – each network is partitioned into SDN zone (i.e., the pink cloud) and non-SDN zone (we can say that is CN zone). In order to provide specific features of different protocols when exchange information between SDN zone and CN zone, all nodes in SDN zone are OF-enable switch that can run the both protocol stacks in SDN and DN.

The source-destination pairs are connected randomly over the network. There are maximum connections of 24 nodes with sending rate of 1 packet per second and with different packet sizes - 512 bytes and 1KB using TCP communication links with 100 Mbps bandwidth. The summary of the topologies is listed in **Table 1.**

Table 1. The network topologies used in evaluaion

Topology	Number of Nodes	Number of Directed Links	Packet Size / Link bandwidths	Number of Connected Nodes
Cost_266	37	57	512 Bytes and 1KB / 100 Mbps	24
France	25	45		



(a) Cost_266 for DN or SDN

(b) France for DN or SDN

(c) Hybrid-Cost_266

(d) Hybrid_France

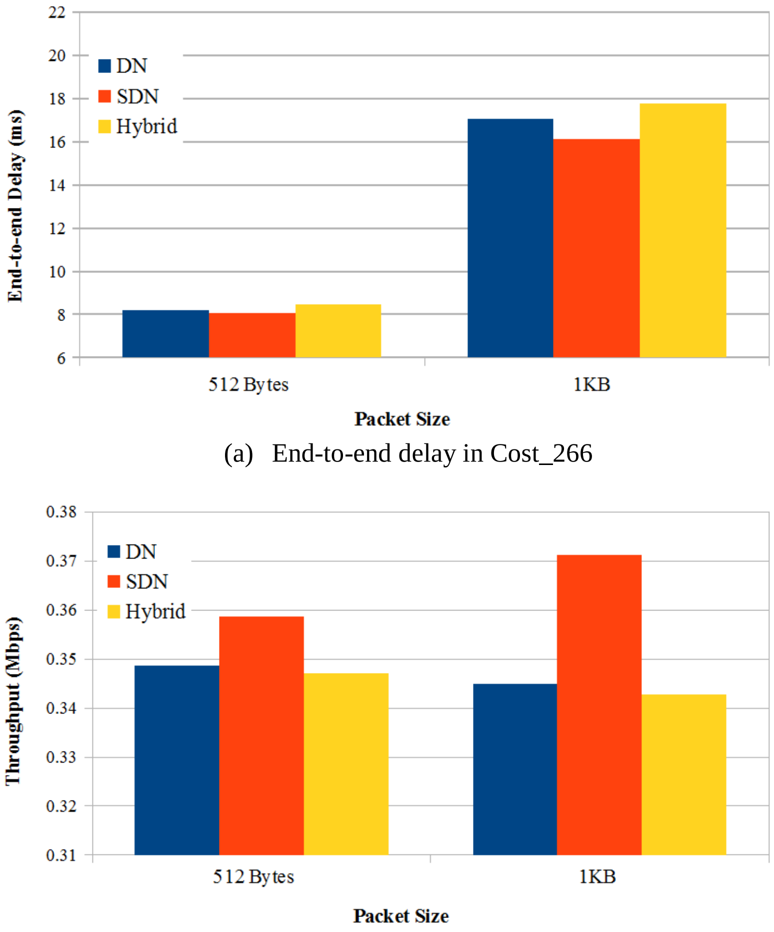
Fig. 4. The four network topologies for the experimental tests.

4.2 Experimental Results

To compare the performance of three network models with respect to two performance metrics such as average end-to-end delay and throughput. Each performance result in **Fig. 5** and **Figl. 6** shows both the small packet size (512 Bytes) and the large packet size (1KB) of data flows.

The average end-to-end delay in Cost_266 is show in **Fig. 5. (a)**, the small packet size and the large packet size, respectively. The end-to-end delay performance of DN and hybrid-SDN is close to each other. We found that SDN has low end-to-end delay compared to DN and hybrid-SDN for both the small and large packet sizes. The throughput performance is shown in **Fig. 5 (b)**. SDN has high throughput for both the small and large packet sizes compared with DN and Hybrid-SDN.

We notice that SDN performs very well in the large and complex network as the Cost_266 network that comprises 37 nodes and 57 links, although it has only one centralized SDN controller. Because the SDN controller has a global view of the network, thus it can manage all traffic flows as like in real-time manner. Therefore, SDN outperforms both DN and hybrid-SDN in the Cost_266 network.



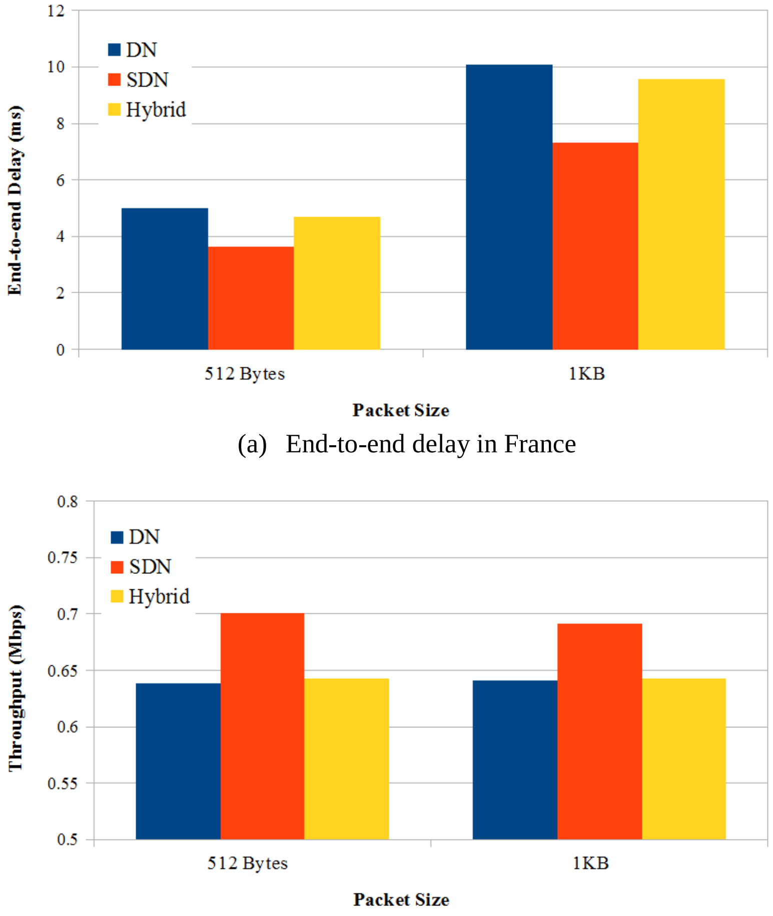
(a) End-to-end delay in Cost_266

(b) Throughput in Cost_266

Fig. 5. The experimental results under Cost_266 network topology with small to large packet sizes.

Fig. 6 presents the performance results of average end-to-end delay and throughput in the France network. The network testbed of France is less complex than the Cost_266 network. There are 25 nodes and 45 links in the France network. The average end-to-end delay in France are shown in **Fig. 6 (a)**, the small packet size and the large packet size, respectively. SDN has low end-to-end delay compared to DN and hybrid-SDN for both the small and large packet sizes. Also, SDN has high throughput for both the small and large packet sizes compared with DN and hybrid-SDN shown in **Fig. 6 (b)**.

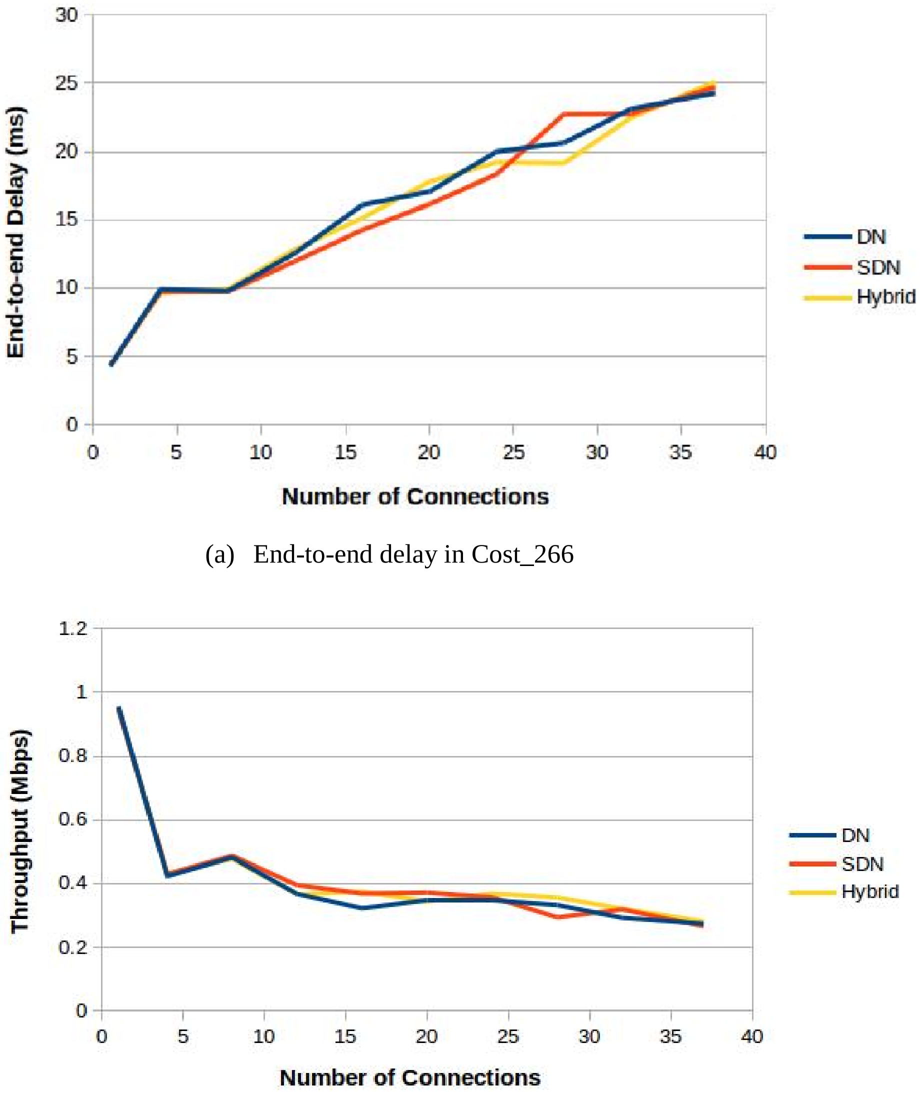
These performance results in the France network are similar to the performance results in the Cost_266 network in **Fig. 5.** However, we found that hybrid-SDN has low delay compared to DN for both the small and large packet sizes, and also it has high throughput at the small packet size and it has a similarly throughput with DN at the large packet sizes. Therefore, we can expect that our hybrid-SDN model may be the efficient network architecture for supporting dynamic traffic services under the specific features of different protocols when exchange information between SDN zone and CN zone in the IoT network. However, there are still many open research problems in hybrid-SDN networks, such as flow management, fault tolerance, topology update and traffic analysis issues. Thus, our next challenge is how to efficiently handle dynamic and different traffic flows in order to improve the network performance in our hybrid-SDN network.



(a) End-to-end delay in France

(b) Throughput in France

Fig. 6. The experimental results under France network topology with small to large packet sizes.



(a) End-to-end delay in Cost_266

(b) Throughput in Cost_266

Fig. 7. The experimental results under Cost_266 network topology at a variety of node connections.

Fig. 7 presents the performance results of average end-to-end delay and throughput when total number of node connections are increased from 1 to 37 with 1KB packet size in the Cost_266 network for comparing three network models. The average end-to-end delay is shown in **Fig. 7 (a)**, which SDN outperforms both DN and hybrid-SDN from 10 to 25 node connections. The delay performance of hybrid-SDN is between DN and SDN up to 25 node connections, and hybrid-SDN outperforms both DN and SDN from 25 to 35 node connections. DN has high delay compared to SDN and hybrid-SDN when node connections are increased. Also, DN has low throughput at 8 through 34 node connections shown in **Fig. 7 (b)**. However, SDN and hybrid-SDN have similar throughput performance and they outperform DN from 8 to 34 node connections.

Therefore, the hybrid-SDN model can be a promising and efficient network architecture when we apply a traffic engineering techniques such as load balancing or scheduling schemes into our hybrid-SDN architecture. Thus, our ongoing research work is to develop the traffic engineering schemes for flow management in hybrid SDN networks for IoT environments.

5. Conclusion

In this paper, we introduced a hybrid-SDN architecture that integrates Software Defined Network and Distributed Network for the next evolution of IoTs. The future IoT requires more efficient management systems for a vast number of smart heterogeneous devices. The performance evaluation validated that our hybrid-SDN network approach is able to apply to the next generation IoTs. However, the hybrid-SDN network needs more improvement in its delay and throughput in order to be widely accepted by many IoT applications. Thus, our immediate future improvement of this project would involve a study of flow management with load-balancing and scheduling solutions.

References

[1] Evans, Dave. "The Internet of Things How the Next Evolution of the Internet is Changing Everything (April 2011)," *White Paper by Cisco Internet Business Solutions Group* (IBSG), 2012.

[2] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M. and Zolla, J. "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp.3-14, 2013. Article (CrossRef Link)

[3] Akyildiz, I.F., Wang, P. and Lin, S.C. "SoftWater: Software-defined networking for next-generation underwater communication systems," *Ad Hoc Networks*, 46, pp.1-11, 2016. Article (CrossRef Link)

[4] Hassas Yeganeh, S. and Ganjali, Y. "Kandoo: a framework for efficient and scalable offloading of control applications," in *Proc. of Proceedings of the first workshop on Hot topics in software defined networks*, pp. 19-24. ACM, August 2012. Article (CrossRef Link)

[5] Dixit, A., Hao, F., Mukherjee, S., Lakshman, T.V. and Kompella, R.. "Towards an elastic distributed SDN controller," in *Proc. of ACM SIGCOMM Computer Communication Review*, Vol. 43, No. 4, pp. 7-12. ACM, August 2013. Article (CrossRef Link)

[6] Fu, Y., Bi, J., Chen, Z., Gao, K., Zhang, B., Chen, G. and Wu, J. "A hybrid hierarchical control plane for flow-based large-scale software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp.117-131, 2015. Article (CrossRef Link)

[7] Lu, Hui, Nipun Arora, Hui Zhang, Cristian Lumezanu, Junghwan Rhee, and Guofei Jiang. "Hybnet: Network manager for a hybrid network infrastructure," in *Proc. of Proceedings of the Industrial Track of the 13th ACM/IFIP/USENIX International Middleware Conference*, p. 6. ACM, 2013. Article (CrossRef Link)

[8] Stringer, J., Pemberton, D., Fu, Q., Lorier, C., Nelson, R., Bailey, J., Corrêa, C.N. and Rothenberg, C.E. "Cardigan: Sdn distributed routing fabric going live at an internet exchange," in *Proc. of Computers and Communication (ISCC), 2014 IEEE Symposium on* (pp. 1-7), June 2014. Article (CrossRef Link)

[9] Rothenberg, C.E., Nascimento, M.R., Salvador, M.R., Corrêa, C.N.A., Cunha de Lucena, S. and Raszuk, R. "Revisiting routing control platforms with the eyes and muscles of software-defined networking," in *Proc. of Proceedings of the first workshop on Hot topics in software defined networks*, pp. 13-18, August 2012. Article (CrossRef Link)

[10] Salsano, S., Ventre, P.L., Lombardo, F., Siracusano, G., Gerola, M., Salvadori, E., Santuari, M., Campanella, M. and Prete, L. "Hybrid IP/SDN networking: open implementation and experiment

management tools," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp.138-153, 2016. Article (CrossRef Link)

[11] Akyildiz, I.F., Lee, A., Wang, P., Luo, M. and Chou, W. "A roadmap for traffic engineering in SDN-OpenFlow networks," *Computer Networks*, 71, pp.1-30, 2014. Article (CrossRef Link)

[12] Vissicchio, S., Vanbever, L. and Bonaventure, O. "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp.70-75, 2014. Article (CrossRef Link)

[13] De Oliveira, R.L.S., Shinoda, A.A., Schweitzer, C.M. and Prete, L.R. "Using mininet for emulation and prototyping software-defined networks," in *Proc. of Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, pp. 1-6, June 2014. Article (CrossRef Link)

[14] Floodlight, P. "Project floodlight open source software for building software defined networks," 2012. Article (CrossRef Link)

[15] Orlowski, S., Wessäly, R., Pióro, M. and Tomaszewski, A. "SNDlib 1.0—Survivable network design library," *Networks*, vol. 55, no. 3, pp.276-286, 2010.

[16] Zhang, J., Xi, K., Luo, M. and Chao, H.J. "Load balancing for multiple traffic matrices using SDN hybrid routing," in *Proc. of High Performance Switching and Routing (HPSR), 2014 IEEE 15th International Conference on*, pp. 44-49, July 2014. Article (CrossRef Link)

[17] De Maesschalck, S., Colle, D., Lievens, I., Pickavet, M., Demeester, P., Mauz, C., Jaeger, M., Inkret, R., Mikac, B. and Derkacz, J. "Pan-European optical transport networks: an availability-based comparison," *Photonic Network Communications*, vol. 5, no. 3, pp.203-225, 2003. Article (CrossRef Link)

Ahyoung Lee received the M.S., and Ph.D. degrees in Computer Science and Engineering from the University of Colorado, Denver, CO USA in 2006 and 2011, respectively. She was a Postdoctoral Fellow at the Georgia Institute of Technology, in the Broadband Wireless Networking Laboratory (BWN Lab) under the supervision of Prof. Dr. Ian F. Akyildiz with a research project focused on Software Defined Networking (SDN). Currently, she is an assistant professor with the Department of Computer Science at the University of South Dakota. Her main research interests include Ad Hoc Wireless Networks, Sensor Networks, Mobile Wireless Networks, Software Defined Networking, and future IoT/IoE Networks.

Xuan Wang, is a Master student of Computer Science. Interest include research, algorithm and network. He currently study at the University of South Dakota for Master Degree since 2015 and work with Dr. Lee with a research project focused on Hybrid Software Defined Networking (HSDN).

Hieu Nguyen received the B.S. degree in telecommunication engineering from Hanoi University of Science and Technology, Vietnam, in 2013. He is currently working toward the Ph.D. degree at the DECENT Lab, Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO, USA. His research interests include computer networks and high-speed communication system utilizing SDN and NFV technologies.

Ilkyeun Ra received a Ph.D. degree in Computer and Information Science from Syracuse University, USA, MS degree in Computer Science from University of Colorado Boulder, Colorado, USA, and BS degree and MS degree in Computer Science from Sogang University, Seoul, Korea. Currently, he is an associate professor in the Department of Computer Science and Engineering at the University of Colorado Denver. His main research interests include computer networks, IoT, cloud computing, large scale of distributed computing systems, and high performance computing.