

## Research Article

# Proxy SDN Controller for Wireless Networks

**Won-Suk Kim and Sang-Hwa Chung**

*Pusan National University, Busan 46241, Republic of Korea*

Correspondence should be addressed to Sang-Hwa Chung; [shchung@pusan.ac.kr](mailto:shchung@pusan.ac.kr)

Received 4 March 2016; Revised 8 June 2016; Accepted 15 June 2016

Academic Editor: Claudio Agostino Ardagna

Copyright © 2016 W.-S. Kim and S.-H. Chung. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Management of wireless networks as well as wired networks by using software-defined networking (SDN) has been highlighted continually. However, control features of a wireless network differ from those of a wired network in several aspects. In this study, we identify the various inefficient points when controlling and managing wireless networks by using SDN and propose SDN-based control architecture called Proxcon to resolve these problems. Proxcon introduces the concept of a proxy SDN controller (PSC) for the wireless network control, and the PSC entrusted with the role of a main controller performs control operations and provides the latest network state for a network administrator. To address the control inefficiency, Proxcon supports offloaded SDN operations for controlling wireless networks by utilizing the PSC, such as local control by each PSC, hybrid control utilizing the PSC and the main controller, and locally cooperative control utilizing the PSCs. The proposed architecture and the newly supported control operations can enhance scalability and response time when the logically centralized control plane responds to the various wireless network events. Through actual experiments, we verified that the proposed architecture could address the various control issues such as scalability, response time, and control overhead.

## 1. Introduction

In recent years, software-defined networking (SDN) technology has gained attention because it can support network programmability [1]. SDN enables network devices, such as switches, to perform only data plane processing by separating the control plane and data plane of the network. At the same time, SDN controller performs only network control. The controller also maintains a global network view by periodically collecting the network state from the network devices. In addition, a network administrator can easily configure network policies by implementing a network application that utilizes the global network view through a northbound application programme interface (NBAPI) provided by the controller.

SDN has utilized the concept of the logically centralized control plane, and it naturally constitutes the control plane with physically distributed controllers to achieve a variety of control objectives. Whereas multiple physical controllers may address the control issues such as control bottlenecks and a single point of failure, management complexity becomes

quite high. Examples of increasing management complexity are synchronization between the controllers to share the network view, the operations of the network application such as policy distribution, and load balancing among the controllers. Unfortunately, although there are quite a large number of controller vendors, a standard of the east-bound/westbound protocol for multiple physical controllers does not yet exist.

Because cases of SDN application to actual networks are increasing, requests to control wireless networks and wired networks over a single logical control plane have been growing. Consequently, advanced SDN architectures for various wireless networks have been proposed continually [2–7]. In addition, such architectures can be utilized for controlling relatively complex control plane of a reliable wireless infrastructure, such as an advanced routing protocol, information centric networking (ICN), and a mobile network [8–13]. If a control point of wired and wireless networks is unified, the controller can perform management of the various resources of the wireless networks as well as flow control. Furthermore, it is possible to support a high level

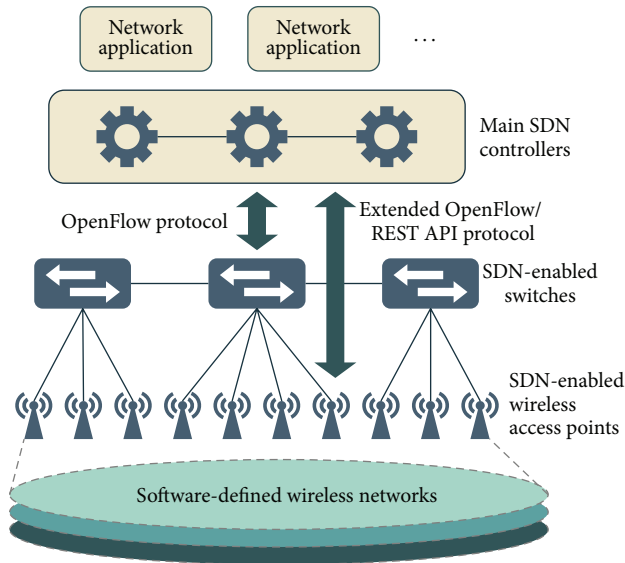


FIGURE 1: A concept of software-defined wired and wireless networks.

of control policies that reflect different types of networks at the same time. In other words, if the scope of SDN is expanded to wireless networks, the SDN architecture will be able to support a variety of network applications for wired and wireless networks.

The SDN-based control method of wireless networks is different from that of wired networks. Enterprise networks or campus networks that are aimed at users who often use mobile devices, such as smartphones, laptops, and handheld devices, generally consist of many switches and even more wireless network devices such as a wireless access point (WAP) or a home node B (HNB). Such networks are configured as seen in Figure 1, which shows several WAPs physically connected to a single switch. If remote controllers manage a large number of WAPs in addition to the switches, the control overhead at the controllers obviously becomes even greater than that when controlling only the wired networks.

In addition, most control for wireless network components such as WAPs is conducted by the event-driven scheme. A station in a wireless network generally has large mobility, so wireless network events, such as station handoff, received signal strength indicator (RSSI) change, signal-to-noise ratio (SNR) change, channel state change, and WAP error, occur more often than events in the wired networks such as link failure and switch error. In other words, network state updates by a topology change and variable link states occur more frequently. Therefore, if the control target of SDN is extended to wireless networks, the additional overhead in the controller will be inevitable by the additionally given roles, such as network join/leave of station, authentication, security, handoff, and WAP load balancing. When the role of the controller increases, control issues such as scalability and reliability will intensify, and new issues such as timeliness may arise.

When an existing SDN architecture controls the wireless networks, expected challenges are as follows:

- (i) Scalability and the control bottleneck due to a large number of WAPs.
- (ii) Controller overload by event-driven control.
- (iii) Complexity of the controller operations due to various event types of the wireless networks.
- (iv) Possibility of supporting events that require rapid processing.

We have previously studied SDN architecture to solve these issues [14]. The previous work was a concept study for local control and cooperation utilizing the proxy controller. The study however lacked the details of the event categorization and did not address the control type to respond to network events that require a global view. In this paper, we propose Proxcon, which is SDN architecture for wireless network control, to advance the proxy controller technology, and propose the operations and roles of the Proxcon agents, which are operated on each WAP, in detail. In particular, the agent is designed to support fine-grained control that can respond in a more adaptive way to the event types and network state. Based on the features related to the agent, in addition, we introduce a hybrid control type as well as the local control and cooperation based on the proxy controller.

Proxcon embeds a proxy SDN controller (PSC) and a Proxcon agent in each WAP. The PSC takes part of the role of a main SDN controller (MSC, the existing central controller). The PSC obtains the changed network policies by synchronizing with the MSC. With these policies, the PSC can directly perform wireless network control that requires only a local network view. In addition, the Proxcon agent maintains information on the controllers in charge of various network events and utilizes the information to forward control messages. The Proxcon agent also delivers information on the various network events occurring in the wireless networks to the MSC or PSC.

The challenging issues mentioned above arise when the existing SDN architecture controls wireless networks. We contribute to resolving these issues through Proxcon, as proposed in this paper. Proxcon supports three offloaded SDN control operations based on the PSC and the Proxcon agent mentioned above. The offloaded control operations are the local control by each PSC, the hybrid control utilizing both the PSC and the MSC, and the locally cooperative control utilizing the PSCs. These operations can be selectively used according to the event characteristic and address the challenging issues by reducing the loads of the MSC and the network efficiently. How these operations address the challenges is summarized as follows:

- (i) The local control by each PSC can resolve scalability and the control bottleneck due to a large number of WAPs. The control scalability is efficiently handled through the local control based on the synchronization between the MSC and PSCs.

- (ii) The controller overload can also be resolved by the local control. By this operation, the MSC will offload most of the control load onto the PSCs.
- (iii) The complexity of physically distributed controller affects the operation of network applications, such as policy distribution, and control load balancing among the controllers. These issues can be addressed appropriately through synchronization between the single MSC and PSCs in Proxcon. The network application instances on the MSC may be configured to perform the same role in all or part of the PSCs. The control load balancing need not be considered because the number of events that need to be processed in each PSC is only a small portion compared with the entire network.
- (iv) Supporting the events that require rapid processing can be achieved through the local control and cooperative control.

Moreover, Proxcon can ensure control reliability by performing the local control regardless of the network state. The efficiency of Proxcon was evaluated through actual experiments in the testbed.

The remainder of this paper is organized as follows. Section 2 presents related work, and Section 3 presents Proxcon details. In Section 4, the experimental results are analyzed, and we conclude the paper in Section 5.

## 2. Related Work

Research on wireless network control based on WAP control has been actively conducted. In addition, some researchers have configured a WLAN control system using their own protocol without applying the concept of SDN.

Shrivastava et al. [15] proposed Centaur which is a WAP-controller collaboration system to address a hidden/exposed terminal problem, which is an old and well-known issue. Murty et al. [16, 17] proposed DenseAP which is a centralized structure for wireless network control and introduced Dyson which is a network control scheme based on Python. Dyson can support various network applications for wireless network control with only the implementation of a simple code using Dyson APIs, but it incurs station-side modifications. These studies concentrated explicitly upon control functions of a central controller, but they did not consider the issues of scalability and reliability owing to the bottlenecks in the controller itself and the control overhead between WAPs and the controller.

SDN was first studied as a control system for only wired networks, and thus communication protocols such as OpenFlow have been designed for this purpose. However, as wireless networks have been rapidly diffused, SDN-based wireless network control technologies have also been steadily studied. Suresh et al. [18] and Schulz-Zander et al. [19] presented Odin which is a structure that concentrates most of the control functions for the WAPs on the central controller, and this system can support network-initiated handoff by configuring virtual basic service sets (BSSs). Zhao et al. [20] designed the

extended OpenFlow protocol for the WAP control, and the system based on this protocol supports seamless handoff by setting exclusive rules in the WAPs. However, these studies focused on SDN-based wireless network control only and did not consider scalability issues caused by deploying a large number of WAPs.

In addition, the SDN technologies for various types of wireless networks have constantly been studied [21]. The studies for software-defined wireless network (SDWN) have been designed to consider many features of wireless networks, such as error-prone communication medium, channel interference, user mobility, and unpredictable changes in link quality and network topology. Yao et al. [22] proposed a control traffic load balancing scheme between multiple controllers. The study adopted the method of distribution and centralization management between multiple network clusters and proposed a double threshold approach to evenly allocate the control traffic load. Proxcon has been designed to support the proxy controller based offloaded control to handle these challenges without high complexity caused by multiple controllers.

Riggio et al. [23] proposed programming abstractions for SDWN. The authors state that the flow abstraction of OpenFlow for software-defined wired network did not consider the characteristics of the wireless network. Also, they argued that the programming abstractions should consider the nature of radio link, resource allocation granularity, and link/radio heterogeneity. The proposed abstractions addressed the four control aspects of wireless networks such as state management, resource allocation, network monitoring, and network reconfiguration. This study increased the efficiency of network application implementation by providing the programming abstractions of SDWN but did not address the increase of control overhead in SDWN. Proxcon proposes the control system that eliminates some inefficiency control points, so Proxcon architecture and the programming abstractions may be compatible.

Research on the effective configuration of a physically distributed control plane of SDN has been performed recently. Tootoonchian and Ganjali [24] and Yeganeh et al. [25] proposed HyperFlow which is an event-based distributed control plane for OpenFlow. This architecture adopted the logically centralized, physically distributed control plane. It introduced the utilization scheme of multiple controllers holding their own local network views. The system assumed that the events explicitly affecting the global network view are very few. However, in wireless networks, the events affecting the network state occupy a significant percentage because of the greater mobility of stations.

Dixit et al. [26] proposed ElastiCon to reduce the control load by expanding or shrinking the controller pool in response to a current control overhead. Riggio et al. [27] performed a taxonomical analysis to assess the admissibility of a variety of wireless network management functionalities in the remote management plane. In addition, they also specified the control category for wireless networks by classifying well-known FCAPS (fault, configuration, accounting, performance, and security) functionalities. These studies focused on the operation between the controller domains;

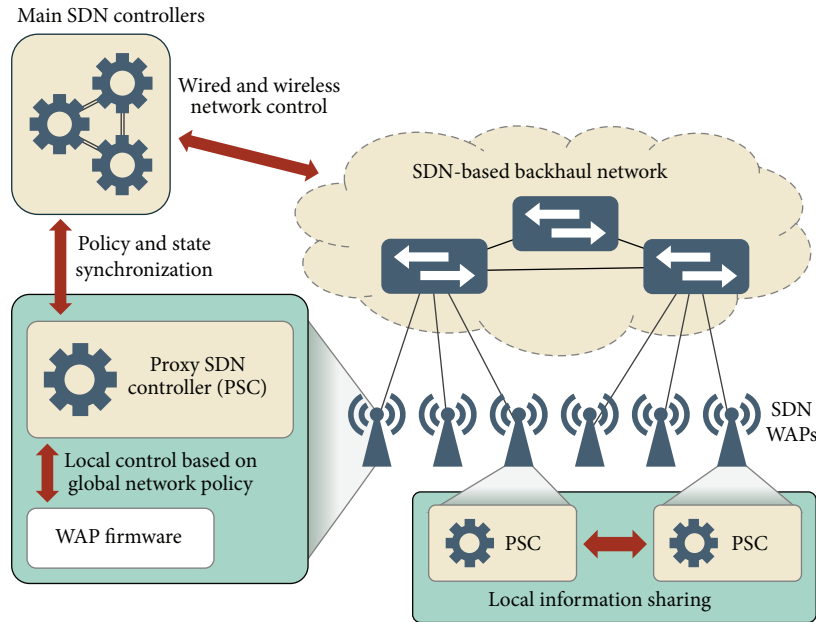


FIGURE 2: Overall architecture and functionalities of Proxcon.

in contrast, a method that maximizes operation efficiency within a single controller domain is proposed in this paper.

The scalability issue occurs inevitably in SDN based on the logically centralized control plane. Some studies have utilized network devices such as a switch to address this problem. Curtis et al. [28] proposed DevoFlow to support scalability through an operation in which the switches decide local routing by using a wildcard in a flow table while the controller directly controls elephant flows. Yu et al. [29] introduced DIFANE to take advantage of the concept of an authority switch to support scalability. In this system, the controller generates authority rules based on network policies and sends the authority rules to the authority switch. The authority switch sets the flow table by the authority rules and forwards rules to neighboring switches. These studies attempted to resolve the scalability problem by moving part of the control plane to the network devices. However, these techniques are unsuitable for response in a variety of network events that occur in wireless networks.

### 3. Proxcon Architecture

**3.1. Proxcon Architecture Overview.** In SDN-based control system for wireless networks, if the MSC additionally handles event detection and periodic reports from a larger number of WAPs compared to the number of switches, a significant load will be added onto the network as well as the controller itself. Because the overload causes degradation of the timeliness and availability of network control, the control reliability will be eventually reduced.

Figure 2 presents an overview of the overall architecture of Proxcon and its functionalities. Proxcon maintains the wireless network control plane in a logically centralized and physically distributed form. At first glance, it seems to be

a logically distributed control plane; but, actually, the MSC is solely in charge of controlling the wireless networks. The PSC is merely a proxy controller that has been delegated to control the WAP in which the PSC is operating.

Proxcon, which is proposed to improve the efficiency of the SDN-based control for wireless networks, assigns the PSC within the WAP to handle part of roles instead of the MSC. Proxcon supports three offloaded SDN control operations for control of the wireless networks by utilizing the concept of the proxy controller. The offloaded control operations are the local control by each PSC, the hybrid control utilizing both the PSC and the MSC, and the locally cooperative control utilizing the PSCs. By supporting these control operations, Proxcon provides an efficient control mechanism for a network administrator.

If the network application utilizes the features of Proxcon, control communication between the MSC and the WAPs will be conducted to exchange messages on the global control, the synchronization, and the report. The global control message is based on the global network view, the synchronization message sets the global network policy of each PSC, and the report message is required to maintain the recent global network view. In addition, when Proxcon is applied, the reliability of the wireless network control will be considerably increased because the PSC can be responsible for most of the wireless network control without the reference being affected by the network state.

In Proxcon architecture, the purpose of the communication between the MSC and the WAPs is to manage connection, synchronization, events, and devices. In order to achieve SDN, OpenFlow protocol has been used as de facto standard for southbound protocol between the MSC and switches. Although we have implemented Proxcon by utilizing vendor type messages of OpenFlow, there is no

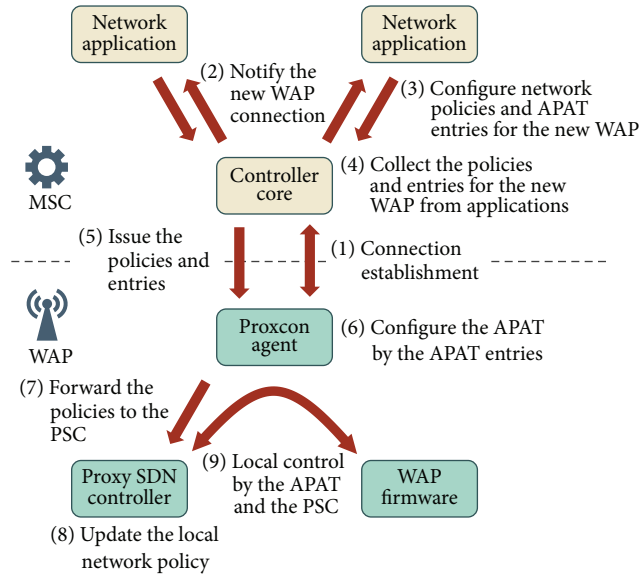


FIGURE 3: First control sequence for the WAP newly connected to the MSC in Proxcon.

standard of the communication protocol between the MSC and the WAPs. The latest SDN controller, such as open network operating system (ONOS) and OpenDayLight, can support several southbound protocols through the provider [30, 31]. Therefore, the communication protocol between the MSC and the WAPs does not need to be configured as the extended OpenFlow but may be a parallel signaling channel using REST API.

**3.2. Policy and State Synchronization.** The MSC and the PSCs in each WAP conduct information synchronization frequently. There are two types of synchronization: synchronization for policy establishment and synchronization for reporting. The synchronization for policy establishment goes from the MSC to the PSC to set the network policy changed by a network application. The synchronization for reporting goes from the PSC to the MSC to update the global network view. When a WAP is connected to the MSC at the beginning or when the network policy is changed, the synchronization for policy establishment is conducted. In contrast, when the network is operating normally, the network state information collected by the PSC is periodically reported to the MSC through the synchronization for reporting.

The modification of the controller in charge of a certain event is performed by the network application, and the information is held in the access point action table (APAT) of a Proxcon agent within each WAP. The event is then processed by the MSC or the PSC in accordance with APAT entry.

Figure 3 shows the synchronization for policy establishment during the initial connection of a WAP. After OpenFlow connection is established between the MSC and the Proxcon agent within the WAP, a network application on the MSC determines whether there is a need to forward their policies to the WAP. If there is an event that can be processed by local control, the network application forwards the MSC

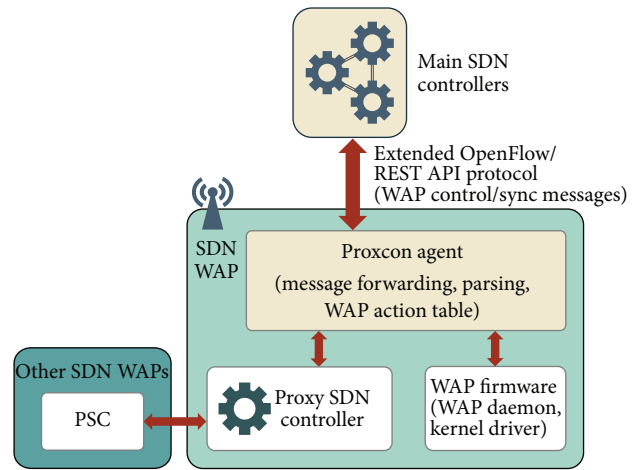


FIGURE 4: Internal architecture of wireless access point and control message flows of Proxcon.

the event type and the network policy related to the event. The MSC collects the network policies and the event types from the network applications and then forwards the WAP the collected data and the APAT entries. After the Proxcon agent receives the message, the agent updates its APAT according to the APAT entries, including the information of the controllers in charge of the specific event, and forwards the network policies to the PSC. The PSC sets an event processing policy with the received message. When the network event occurs after this process is performed, the WAP firmware will notify the Proxcon agent. The agent then converts the received event information into an OpenFlow message and forwards this message to the MSC or the PSC according to the APAT entry.

A network application sets APAT entry for a PSC to collect unimportant monitoring data, such as link quality above the threshold and the congestion level below a certain value. Therefore, the unimportant, ordinary monitoring data collected by WAPs are stacked in the PSC within the WAP. In addition, when a certain link quality decreases excessively or the congestion level increases sharply, critical monitoring data may be collected by the WAP. To update the global network view, the APAT may be configured to forward the critical monitoring data to both the MSC and the PSC.

If the network application that requires ordinary monitoring data is utilized, the APAT may be configured to forward all monitoring data to both the MSC and the PSC. Unfortunately, this configuration incurs considerable control traffic in the network. Therefore, to reduce the control traffic, the network application can modify the APAT for the PSC to directly collect noncritical monitoring data and periodically report processed data to the MSC.

**3.3. Design of the Wireless Access Point in Proxcon.** The WAPs in Proxcon consist of a PSC, a Proxcon agent, and WAP firmware. Figure 4 shows the internal structure of the WAP and control message flows. The Proxcon agent, which is a core module of the WAP, is responsible for communication with the MSC and processes messages based on OpenFlow

TABLE 1: Example of the access point action table in the Proxcon agent.

Event source	Event type	Condition	Event target
WAP firmware	Station join	$RSSI > \alpha$	PSC, MSC
WAP firmware	Station join	$RSSI \leq \alpha,$ $RSSI > \beta$	MSC
WAP firmware	Station join	$RSSI \leq \beta$	DROP
WAP firmware	Station leave		PSC, MSC
WAP firmware	Station monitoring	...	PSC
WAP firmware	Network monitoring	...	PSC
WAP firmware	WAP failure		PSC, MSC
MSC	Policy issue		PSC
MSC	Global control message		WAP firmware
PSC	Event report		MSC
PSC	Local control message		WAP firmware

or REST API, which is a southbound protocol. In addition, it maintains the information of the controllers in charge of the specific event by using the APAT. The PSC performs local control for the WAP by utilizing an event response policy, which is set by synchronization with the MSC, and supports local cooperation with neighboring PSCs. Proxcon is based on the extended OpenFlow protocol or REST API that includes synchronization messages and WAP control messages.

Table 1 shows the structure of the APAT in the Proxcon agent. The APAT is composed of an event source, an event type, response conditions, and an event target. The event source and target include the WAP firmware, the MSC, and the PSC. The event type field contains the event type defined in Proxcon. The response condition, which is an optional field, divides the destination of a particular event type according to its own entry.

As mentioned above, the Proxcon agent parses predefined protocol messages received from the MSC and forwards the messages to the PSC or the WAP firmware. The message forwarded to the PSC is the synchronization message, and the message delivered to the WAP firmware is the direct control message for the WAP. The PSC and the WAP firmware also send and receive messages through the Proxcon agent. The PSC sends the local control message for the WAP and the report message to the WAP firmware and the MSC through the Proxcon agent, respectively. The WAP firmware forwards messages, such as a response to the WAP control message, an event in the wireless networks, or a periodic report about channel or network state, to the Proxcon agent. The messages from the WAP firmware are forwarded to the MSC or the PSC according to the APAT.

**3.4. Proxcon Operation Details.** As mentioned previously, the changes in network state and topology are frequent in wireless networks compared to wired networks owing to the greater

mobility of stations and the link characteristic. Therefore, control of the wireless networks is based on the network events, such as station joining, channel resource control, and station handoff. To provide a reliable network service for users in such an environment, a control structure is required to respond appropriately to various network events that occur periodically. This section describes Proxcon operations divided into three categories with respective examples.

Figure 5 shows three offloaded SDN operations supported by Proxcon. The operations based on the concept of the proxy controller are intended to respond to wireless network events efficiently. The first operation is the local control by each PSC, which enhances the response time and scalability. The second operation is the hybrid control utilizing both the PSC and the MSC, which is used to respond to an event that may not be handled by the PSC alone. The controller in charge of the event is selected by the APAT according to the preset condition. The last operation is the locally cooperative control utilizing the PSCs. This operation reduces the control traffic in the network and enhances the scalability.

**3.4.1. Proxcon Operation 1: Local Control by Each Proxy SDN Controller.** The network application of Proxcon can be configured to set the PSC for direct processing of particular events that may not require the global network view. The type of these events may include a station join/leave, monitoring data collection, WAP resource control, traffic shaping, and deep packet inspection (DPI). In Proxcon, the MSC should provide the appropriate NBAPI to allow the network applications to transfer part of their functions to the PSC.

For example, consider the WAP resource control. In IEEE 802.11 wireless local area network (WLAN), one WAP can provide multiple virtual basic service sets (VBSS) through WAP virtualization or assign different resources for each VBSS. There is a scheme that sets the channel share differently for each VBSS by the uplink/downlink traffic control. In the existing SDN architecture, the network application for VBSS management is in charge of VBSS-specific resource allocation [32]. After each WAP reports its current state to the MSC, the WAP is configured to apply the changed resource values, such as queueing, enhanced distributed channel access (EDCA), and transmit opportunity (TXOP) values. In this case, the network application must respond to rapidly changing traffic conditions by station mobility or traffic patterns. Thus, this scheme must be implemented considering the load of the MSC, control traffic in the network, and a relatively slow response time. In addition, if the number of WAPs or stations is increased, this results in a scalability issue. On the other hand, the control reliability can be ensured in Proxcon because the PSC within each WAP can directly respond to the traffic condition after the initial policy synchronization.

**3.4.2. Proxcon Operation 2: Hybrid Control Utilizing Both the PSC and the MSC.** The APAT can be implemented to support a conditional operation. The condition may include the event attributes, event frequency, response overhead, and current load of the MSC or the WAP. It is possible to select the controller in charge in accordance with the conditions.

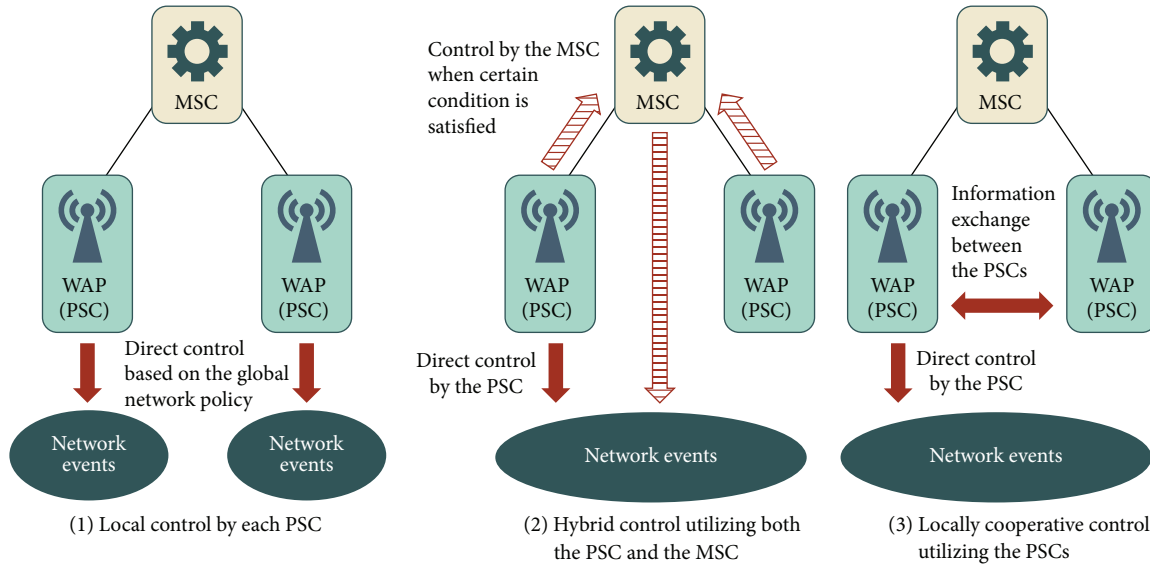


FIGURE 5: Offloaded SDN operations supported in Proxcon.

Naturally, the policy to select the controller in charge is also determined at the network application.

A station sends a probe request frame containing the service set identifier (SSID) value to request a connection to the IEEE 802.11 WAP. Because the probe request frame is transmitted as a broadcast, the request frame can be received by the adjacent WAPs operating on the same channel at the same time. The WAP near the station may receive the frame with a higher RSSI value, whereas the WAP far from the station will receive the frame with low signal strength. In this case, if the policy is set to perform a direct control by the PSC, it may lead to inefficient results because one PSC cannot be aware of the situation of several WAPs. For example, every WAP may respond to the request frame regardless of the RSSI value, and thus the link quality of the connection may indicate a poor quality. For these events, the network application in Proxcon can be implemented to support policies to take advantage of both the MSC and the PSC.

Consider the station join event in the APAT shown in Table 1 again. The APAT entries indicate that the controllers in charge of the station join event are configured differently according to the condition, which is set by the network application. It is assumed that the network application has set the conditions for the station join event at three according to the RSSI value.

Table 2 illustrates the result of conditional action for the station join event in Table 1. It is assumed that there are two WAPs (WAP A and WAP B), and one station sends the probe request frame. The two leftmost columns in the table show the range of the RSSI values of the probe request frame received in each WAP as classes A, B, and C. Class A ( $RSSI > \alpha$ ) specifies an even higher RSSI range that can be directly processed by the PSC, class B ( $\beta < RSSI \leq \alpha$ ) indicates the range to directly choose the appropriate WAP by the network application on the MSC, and class C ( $RSSI \leq \beta$ ) denotes the range of low

RSSI values that cannot communicate normally even if the connection is established.

If both WAPs receive the probe request frame with RSSI belonging to class A, the PSCs in both WAPs will directly process the station join event separately. Therefore, the station will directly select one of two WAPs for the connection. If one WAP receives the frame in class A and the other WAP receives class B frame, only the PSC in the WAP that receives the frame with class A will perform the response. Note that the Proxcon agent in the WAP that receives the frame belonging to class A forwards the frame to the MSC. This is a report for processing the requests with class B in the network application. The network application collects the request frames that belong to classes A and B, and then it does not respond to the station join event because it knows that the PSC in the WAP receiving class A frame already responded. In addition, the probe request frame that belongs to class C is dropped without any process.

If both WAPs receive the frame belonging to class B, the frames will be reported to the network application on the MSC, and then the network application will issue a command to respond by selecting one of the WAPs. If both WAPs receive the frame belonging to class C, any process for the station join event will not proceed.

Through the methods described above, the network application can be implemented to take advantage of both the MSC and the PSC for a particular event.

**3.4.3. Proxcon Operation 3: Locally Cooperative Control Utilizing the Proxy SDN Controllers.** Another operation of Proxcon is locally cooperative control utilizing the PSCs. This operation can be used for an event that cannot be handled by the PSC alone. A case in which the PSC cannot solely respond to the event may indicate that direct control of the

TABLE 2: Example of conditional operations for the station join event.

WAP A	WAP B	WAP A PSC	WAP B PSC	Network application on MSC	Result
A	A	Response to the station	Response to the station	Ignore	The station selects the WAP by itself
A	B	Response to the station	—	Ignore	WAP A by the PSC of WAP A
A	C	Response to the station	—	Ignore	WAP A by the PSC of WAP A
B	B	—	—	Select the WAP reporting the higher RSSI value (A or B)	One of WAPs by the MSC
B	C	—	—	Select the WAP reporting the higher RSSI value (A)	WAP A by the MSC
C	C	—	—	—	The station cannot join the network

MSC is required to process the event. However, if the event target is limited to a specific area, it may be processed by information sharing between the PSCs without intervention by the MSC. A representative example is a network-initiated station handoff. In the case of the handoff, a network global view is not required to move the station to the other WAP, and only a small amount of information on the adjacent WAP is required. Proxcon supports the PSC cooperation for processing the event that has local characteristics.

First, consider the existing method in SDN for the network-initiated handoff. In the central control scheme for the handoff, the MSC should maintain the signal strength values or other handoff metrics of all stations in the network that are received from all WAPs. For example, if there are twenty WAPs and twenty stations in the network, the MSC always maintains up to four hundred RSSI values in real time. If the network scale increases, this approach may cause large overhead of the MSC and quite high control traffic in the network. Another scheme for the handoff control is a completely distributed one. It performs the handoff by exchanging information with the adjacent WAP when it is necessary for each WAP. However, this approach can make it difficult to change the handoff policy. On the other hand, Proxcon supports locally cooperative control utilizing the PSCs for the station handoff, and the handoff policy can be easily changed through synchronization for policy establishment between the MSC and the PSCs.

The information exchange process between neighboring WAPs designed in this study is as follows. The WAP obtains the information on the neighboring WAPs by scanning after boot and reports it to the MSC after the OpenFlow connection is established. The MSC then informs the WAP of the detailed information on the neighboring WAPs for connection establishment, such as an IP address and a port number. This process is performed periodically during network operation.

Figure 6 shows the control flow between WAPs during the station handoff. It is assumed that the network application has specified the handoff policy to each PSC before the handoff

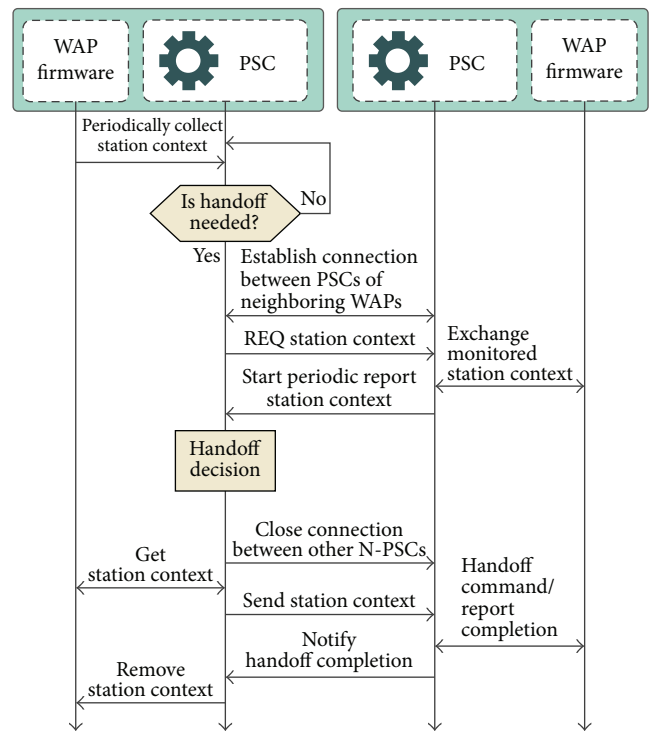


FIGURE 6: Handoff process by locally cooperative control utilizing the PSCs.

process, such as the station RSSI collection cycle, the start of signal overhearing at adjacent stations, the RSSI level to start the information exchange with the PSCs in the neighboring WAPs (N-PSCs), and the RSSI level to perform the handoff. The PSC periodically receives the RSSI value of the stations from the WAP firmware. When the exchange of the RSSI value of a certain station with the N-PSCs is required because the RSSI value of the station is lower than the threshold preset by the network application, the PSC will establish a connection with the N-PSCs. After the connection with



the N-PSCs is established, the PSC periodically exchanges the RSSI values and compares them. If it is determined that the station handoff is required, the PSC will terminate the connection with all other N-PSCs except for the N-PSC that is in the handoff target WAP. The PSC then performs the station handoff by exchanging the information on the station with the target N-PSC. Each PSC reports the handoff result to the MSC after the handoff is completed. By this process, Proxcon can significantly reduce the control overhead of the MSC and the control traffic in the network.

*3.4.4. Time Cost Analysis of Each Proxcon Operation.* In this section, we analyze the offloaded control operations mentioned above in terms of the time cost. For the analysis of the time cost, we use the following notations:

- (i)  $T_{WAP}$ : processing time of the WAP that includes intermodule communication.
- (ii)  $T_{MSC}$ : processing time of the MSC.
- (iii)  $T_{M-W}$ : transmission delay between the MSC and the WAP.
- (iv)  $T_{W-W}$ : transmission delay between the WAP and a neighboring WAP.
- (v)  $T_{collect}$ : collection time at the MSC.
- (vi)  $N_{WAP_n}$ : the number of WAPs near the WAP in which an event occurs.
- (vii)  $T_{O_x}^E$ : consumed time for performing the  $X$ -type operation in the existing SDN architecture.
- (viii)  $T_{O_x}^P$ : consumed time for performing the  $X$ -type operation in Proxcon architecture.

This analysis does not consider unpredictable latency between the WAP and the station, because it intends to evaluate the network control efficiency.

When the event that can be handled through local control occurs, the consumed time can be expressed as follows:

$$\begin{aligned}
 T_{O_L}^E &= 2T_{M-W} + T_{MSC}, \\
 T_{O_L}^P &= T_{WAP}, \\
 T_{O_{L-1}}^E &= T_{M-W} + T_{MSC}, \\
 T_{O_{L-1}}^P &= T_{WAP},
 \end{aligned} \tag{1}$$

where  $T_{O_{L-1}}$  means the time cost of the control operation for handling the event that does not require the response, such as the monitoring information collection.

The time cost of the hybrid control is as follows:

$$\begin{aligned}
 T_{O_H}^E &= 2T_{M-W} + T_{collect} + T_{MSC}, \\
 T_{O_H}^P &= (\beta - 1)(T_{WAP}) + \beta(2T_{M-W} + T_{collect} + T_{MSC}),
 \end{aligned} \tag{2}$$

where  $\beta$  means the probability of occurrence of the event requiring the support of the MSC. The hybrid control operation of Proxcon is directly processed in the WAP or is carried out with the assistance of the MSC.

The time cost of the cooperative control is as follows:

$$\begin{aligned}
 T_{O_C}^E &= T_{M-W} + N_{WAP_n}(T_{M-W} + T_{MSC}) + 2T_{M-W} \\
 &\quad + T_{WAP} + 2T_{M-W} + T_{WAP} + T_{M-W}, \\
 T_{O_C}^P &= T_{WAP} + N_{WAP_n}(T_{W-W} + T_{WAP}) + 2T_{W-W} \\
 &\quad + T_{WAP}.
 \end{aligned} \tag{3}$$

Note that the transmission delay is generally greater than the processing time, so  $T_{WAP}$  and  $T_{MSC}$  are negligible. In addition, we can assume that  $N_{WAP_n}$  is 1, and then the above equations can be simplified as follows:

$$\begin{aligned}
 T_{O_L}^E &= 2T_{M-W}, \\
 T_{O_L}^P &= 0, \\
 T_{O_{L-1}}^E &= T_{M-W}, \\
 T_{O_{L-1}}^P &= 0, \\
 T_{O_H}^E &= 2T_{M-W} + T_{collect}, \\
 T_{O_H}^P &= \beta(2T_{M-W} + T_{collect}), \\
 T_{O_C}^E &= 7T_{M-W}, \\
 T_{O_C}^P &= 3T_{W-W}.
 \end{aligned} \tag{4}$$

Note that  $T_{M-W} > T_{W-W}$  and  $\beta < 1$  are always valid, so  $T_{O_{L,H,C}}^E > T_{O_{L,H,C}}^P$  will be satisfied in all cases.

## 4. Experimental Results

In this study, we evaluated the performance and efficiency of Proxcon in various aspects. Experiments were carried out on the wireless network testbed based on the proof-of-concept (PoC) implementation of Proxcon. The proposed architecture was evaluated compared to the existing SDN architecture, which consists of one central controller. In the experiment, the scalability, control efficiency, and control overhead were evaluated, including the response time according to the increase in the number of WAPs, the communication overhead of the central controller, and the control overhead when the actual network application is operated on each architecture.

*4.1. Experimental Scenario and Testbed.* The MSC was based on ONOS as the SDN controller and was operated on a desktop based on Ubuntu 14.04.2 LTS [30]. The WAP was based on an APU1D4 embedded board by Netgate, and a WLE200NX by Compex Systems based on the Atheros AR9280 chipset was used as the WLAN module [33, 34]. The WAP firmware was implemented by modifying the Linux kernel driver and Hostapd. The stations used in the experiment were based on Raspberry Pi 1 Model B+, and a USB wireless adapter and a separate battery were added [35].

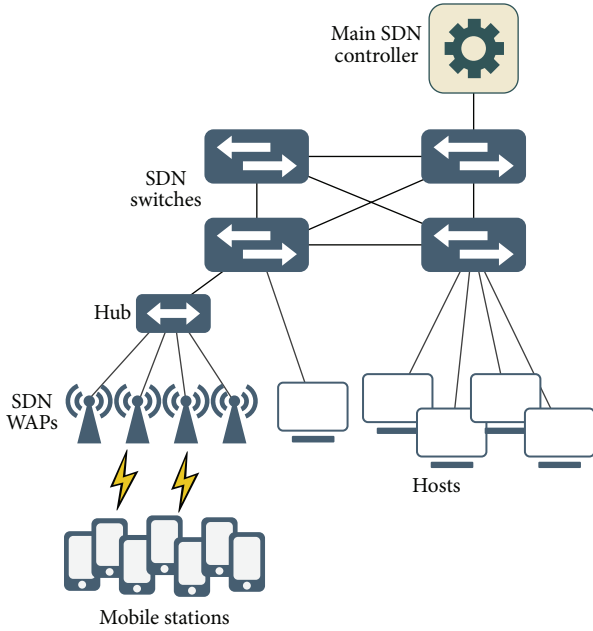


FIGURE 7: Testbed topology for experiments.

To evaluate Proxcon, as shown in Figure 7, we have implemented the SDN-based wireless network testbed composed of a single MSC, seven mobile stations, and four WAPs consisting of the PSC and the Proxcon agent. In all experiments, approximately 100 to 200 wireless network events occur per second from each WAP. Most of these events are to report the RSSI values of the stations; in other words, each WAP reports the RSSI values of the neighboring stations every 20 ms. The generated events also include the event that requires a response, such as packet-IN and network join/leave of the client, and the event initiated by the control plane, such as the WAP configuration change and client handoff.

The frequency of the synchronization between the MSC and the PSCs may be variably set depending on the purpose of the network application, but it was set to 1 s in the experiments. All experiments were compared to the control architecture using a single MSC. Most of the SDN-based wireless network researches have focused on the client handoff, WAP configuration, and the abstraction scheme, and researches on the control plane architecture have concentrated on deployment of the MSC. Therefore, the experiment compared to the single MSC-based control architecture will show quite similar results with direct comparison with existing SDN-based wireless network researches.

**4.2. Scalability and Control Latency Issue.** Figure 8 depicts a cumulative distribution function (CDF) graph showing the average response time of all transactions in accordance with the architecture type and the number of WAPs. The result was quite intuitive, and the response time in Proxcon was significantly faster for most of the event processing than that in the existing SDN architecture regardless of the number of WAPs. In particular, the response time in the existing architecture is even longer because the transaction time of

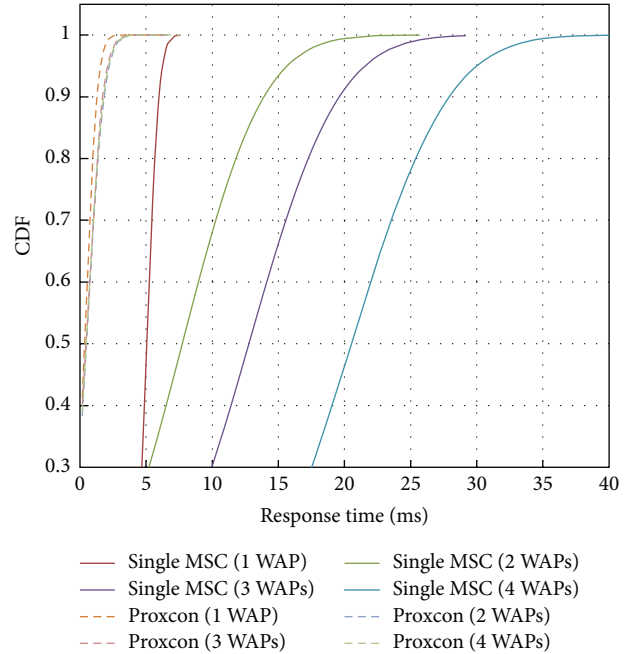


FIGURE 8: Experimental results related to scalability issues according to the control architecture and the number of WAPs.

an event such as station join, which should be processed in a way determined after collection at the MSC, includes a report collection time as well as a network delay. On the other hand, the response time in Proxcon is considerably fast because the PSC solely determines the station connection in most cases by using metrics such as RSSI.

Naturally, the distribution of the response time in Proxcon depends on the proportion of the certain operation type such as the hybrid operation utilizing both the MSC and the PSC and the local cooperation between the PSCs. However, compared to the existing SDN architecture in which the remote MSC is in charge of responses to all events, the distribution in Proxcon will remain low.

Figure 9 shows the response time distributions in a boxplot graph. As described above, the response time in the existing architecture is higher than that in Proxcon owing to the network delay and the data collection latency at the MSC for the particular events. The outliers of the Proxcon graph express the response time of the MSC when the synchronization or hybrid operation is performed.

As shown in both Figures 8 and 9, the response time of the existing architecture is increased according to the number of WAPs, whereas the response time of Proxcon does not increase. As mentioned above, one WAP was set to generate many events owing to the limitations of the testbed. In the real environment, because a large number of WAPs will generate a number of events, the average response time in the real environment will be similar to the foregoing results.

Figure 10 shows the results of a single WAP control response time measured on the existing SDN architecture with background traffic. The background traffic was set to 0, 20, 50, 100, 200, 300, 500, 750, and 1000 Mbps on

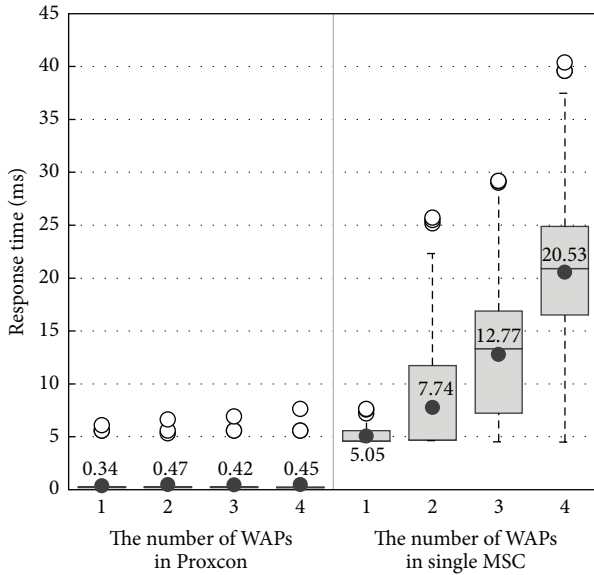


FIGURE 9: Responsiveness comparison between the existing architecture and Proxcon.

a wired link whose physical bandwidth was 1 Gbps. The high background traffic caused the irregular, high control response time. In particular, more than 70 percent of the event response time exceeded 50 ms when the wired link was saturated. These results show that the SDN architecture using a single controller is quite vulnerable to network congestion. In addition, the control reliability may decline significantly in a network with significant background traffic.

**4.3. Effects on Actual Network Application.** The main purpose of Proxcon introducing the concept of the PSC within a WAP is to resolve the issues of scalability and slow response time. To resolve these issues is to respond to the events rapidly and accurately. In this section, we evaluate the control efficiency of the Proxcon operations, such as the PSC-based control, the hybrid control, and the locally cooperative control, when Proxcon actually supports the wireless network applications. Two experiments were conducted: one handles the station join event to evaluate the efficiency of the hybrid control, and the other controls the station handoff to assess the PSC cooperation.

Figure 11 indicates the average RSSI values and the average response time of the station join and station handoff. The stations used in the experiment are stationary or mobile within the wireless network coverage, and the station handoff occurs based on the RSSI value during movement. In both experiments, the average RSSI value is the arithmetic mean of the RSSI values measured in all WAPs every 100 ms. The average latency is a measured value from the moment when the significant value is inputted to the WAP until the moment when the proper handling for the event is performed. As seen in the graph, the average RSSI values of the existing architecture and Proxcon in both experiments remained at a similar level. However, a significant difference is seen in the time required to respond to the event.

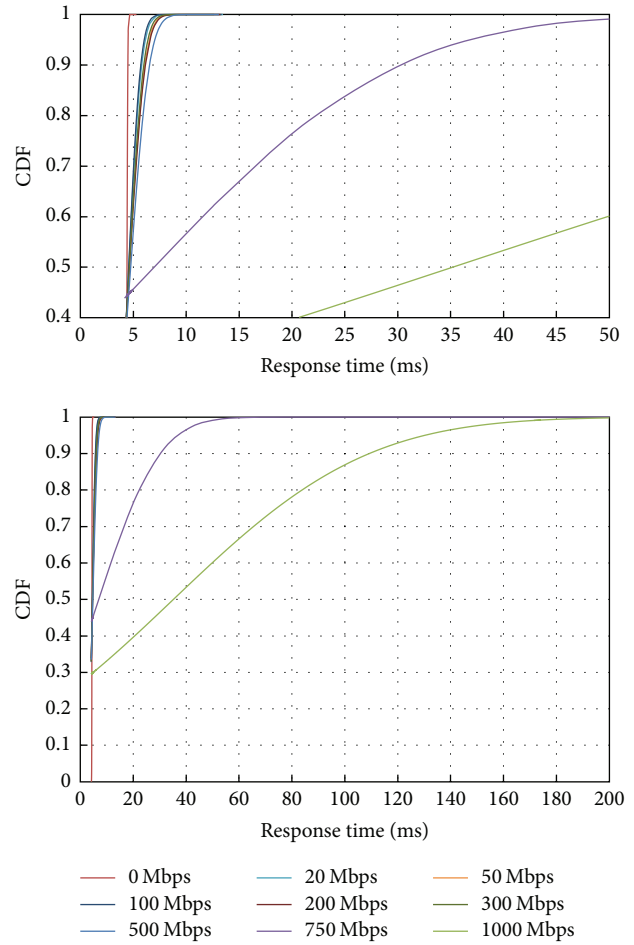


FIGURE 10: Response time in the network congestion state.

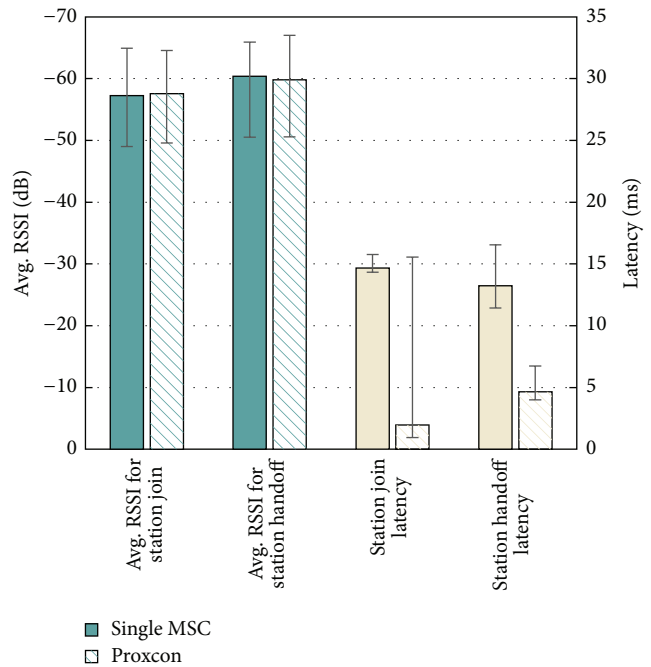


FIGURE 11: Experiments on the actual applications. The yellow bars refer to “single MSC” and the yellow striped bars refer to “Proxcon.”

The latency for a single MSC-based station join process is the value consisting of the time to transmit the event information to the MSC from the WAP, the waiting time for data collection at the MSC, and the time to send the response to a certain WAP. On the other hand, in the case of Proxcon, if the probe request frame with RSSI value exceeding the threshold is received, the PSC immediately processes the event, so only the interprocess communication (IPC) delay is incurred. In a low probability, if a probe request frame with a lower RSSI value than the threshold is received by all WAPs, the hybrid control is required to handle the event, the delay of which would be similar to the single MSC-based control.

The latency for the single MSC-based handoff control is the value comprising the time to transmit the RSSI data from the WAP to the MSC, the time to collect all RSSI reports at the MSC, the time to request the station information on the old WAP and to receive it, the time to forward the information to the new WAP and receive the completion report, and the time to issue the command for removing the station context. On the other hand, the latency in Proxcon consists of the time to transmit the RSSI value of the station to the N-PSCs, the response time from the N-PSCs that overheard the station communication with a higher RSSI value, the transmission time of the station information, and the response time of the handoff completion. Because the communication with the neighboring WAPs is carried out at a relatively low latency, it was confirmed that the handoff control in Proxcon is performed with lower latency compared to communication with the MSC in the existing SDN architecture.

As seen in the experiments above, although Proxcon distributes part of roles of the central controller to each WAP, the wireless network controls based on Proxcon, such as the station join processing and the station handoff control, were performed with almost the same efficiency as the existing architecture. However, the average control time for event processing is decreased significantly. In addition, the difference in the response time will increase with increasing network size or number of WAPs. In other words, when target networks become larger, the control result is expected to be significantly different depending on the control architecture because a high control delay of the existing SDN architecture will significantly reduce the control reliability.

In particular, in terms of the mobile network, it is expected that this control overhead reduction scheme becomes even more important. Looking briefly, the network obviously needs control, and collecting information at the controller is especially required for decision-making in SDN. The mobile network is pushing forward the progress to 5G, and one of the main features of 5G is small cell that collaborates with Wi-Fi. In other words, the range of the base station of 5G mobile network may be extended to the Wi-Fi WAPs, and it means that the density of the base station is also increased.

*4.4. Effects on Main SDN Controller in terms of Traffic.* If the concept of a proxy controller is introduced to control the wireless networks, the control load of the central controller will be naturally reduced. Figure 12 indicates the traffic load

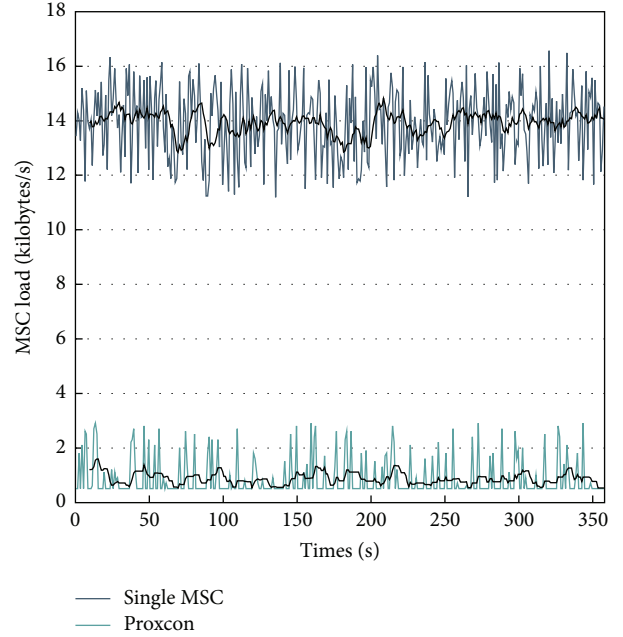


FIGURE 12: Average traffic load of the main SDN controller.

of the MSC according to the architecture type. As mentioned earlier, the synchronization cycle was set to 1 s.

The average control traffic at the MSC was 13.93 kB/s in the existing SDN architecture and 0.87 kB/s in Proxcon. Understandably, Proxcon can reduce the control traffic at the MSC because most of the event responses are performed by the communication of the internal WAP or between the WAPs. This means that an increase in the number of WAPs under Proxcon does not significantly affect the control overhead of the MSC, so the MSC can concentrate on the control of the wired networks as in the existing SDN architecture. In addition, Proxcon can reduce the control traffic on the network and thus perform the network operation more efficiently.

As seen in the experiments above, Proxcon distributes part of roles of the central controller pertaining to network control to each WAP, and thus the control traffic on the network and the response time are reduced. In addition, the network control is not significantly affected by network congestion. This means that it is possible to increase the reliability of the network control and support the scalability. In addition, although the role of the central controller is distributed to each WAP, the control operations for actual network applications, such as station join and station handoff, demonstrated approximately the same efficiency as the existing architecture, whereas it was confirmed that the response time is significantly reduced.

## 5. Conclusion

In this paper, we proposed Proxcon, a control architecture based on SDN for the control of wireless networks. Because control of wireless networks mostly comprises event-based management, Proxcon provides scalability regardless of

the MSC performance and distance on the MSC by supporting three offloaded control operations based on the PSC and the Proxcon agent in every WAP. It also ensures reliable, immediate control based on this scheme.

The proposed architecture was evaluated through experiments on the testbed based on the PoC implementation. According to the experimental results, compared to the existing SDN architecture, Proxcon efficiently resolves the scalability issue that results from the number of WAPs in the network, and it was confirmed that the control performance is not significantly affected by backhaul network conditions such as congestion. From the experiments on handling actual network applications, although the proposed architecture does not manage all network information in the central controller, the response delay was reduced, whereas the control performance demonstrates the same efficiency as the MSC-based control. In addition, Proxcon reduced the operation load of the MSC by reducing the traffic to the MSC, which prevented degradation of the MSC control performance due to the impact of additional functions such as wireless network control.

In the future work, we will focus further on the SDN control architecture to expand the concept of the proxy control plane to the wired networks. More specifically, a study will be performed to address the issues of efficiency and consistency of the synchronization between the central controllers and the proxy controllers. In particular, we will seek to combine the concept of SDN and fog computing that moves some controller instances to WAPs or switches.

## Competing Interests

The authors declare that they have no competing interests.


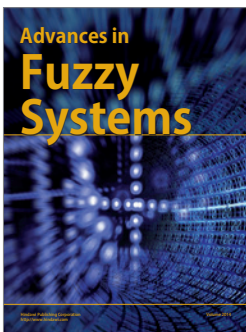
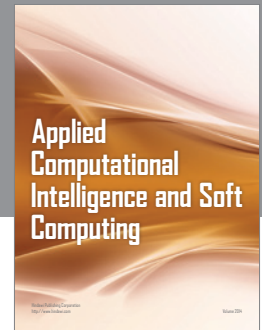
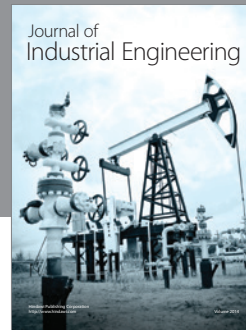
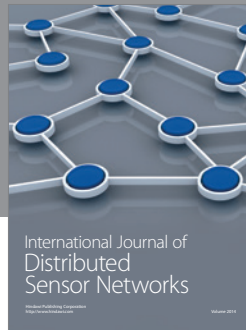
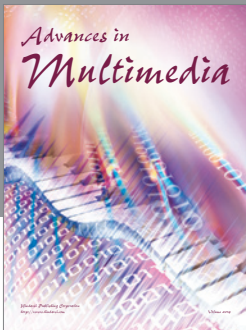
## Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (no. B0126-16-1051, A Study on Hyper Connected Self-Organizing Network Infrastructure Technologies for IoT Service).

## References

- [1] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [2] X. F. Xiao and X. Kui, "The characterizes of communication contacts between vehicles and intersections for software-defined vehicular networks," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 98–104, 2015.
- [3] C.-F. Lai, R.-H. Hwang, H.-C. Chao, M. M. Hassan, and A. Alamri, "A buffer-aware HTTP live streaming approach for SDN-enabled 5G wireless networks," *IEEE Network*, vol. 29, no. 1, pp. 49–55, 2015.
- [4] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: centralized, distributed, or hybrid?" *IEEE Network*, vol. 29, no. 4, pp. 32–38, 2015.
- [5] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 70–75, 2014.
- [6] S. Sezer, S. Scott-Hayward, P. Chouhan et al., "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.
- [7] S. Gringeri, N. Bitar, and T. Xia, "Extending software defined network principles to include optical transport," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 32–40, 2013.
- [8] W.-S. Kim and S.-H. Chung, "Interface assignment-based AODV routing protocol to improve reliability in multi-interface multichannel wireless mesh networks," *Mobile Information Systems*, vol. 2015, Article ID 768596, 16 pages, 2015.
- [9] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, "Information centric networking over SDN and OpenFlow: architectural aspects and experiments on the OFELIA testbed," *Computer Networks*, vol. 57, no. 16, pp. 3207–3221, 2013.
- [10] T.-W. Um, G. M. Lee, and J. Kim, "SDN-based active content networking," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 1603239, 6 pages, 2016.
- [11] W.-S. Kim, S.-H. Chung, and J.-W. Moon, "Improved content management for information-centric networking in SDN-based wireless mesh network," *Computer Networks*, vol. 92, pp. 316–329, 2015.
- [12] C. J. Bernardos, A. De La Oliva, P. Serrano et al., "An architecture for software defined wireless networking," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 52–61, 2014.
- [13] N. Chen, S. Sun, M. Kadoch, and B. Rong, "SDN controlled mmwave massive MIMO hybrid precoding for 5G heterogeneous mobile systems," *Mobile Information Systems*, vol. 2016, Article ID 9767065, 10 pages, 2016.
- [14] W. Kim, S. Chung, and J. Shi, "WiPCon: a proxied control plane for wireless access points in software defined networks," in *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE '14)*, pp. 923–929, Chengdu, China, December 2014.
- [15] V. Shrivastava, N. Ahmed, S. Rayanchu et al., "CENTAUR: realizing the full potential of centralized WLANs through a hybrid data path," in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 297–308, ACM, Beijing, China, September 2009.
- [16] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance enterprise Wi-Fi networks," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI '08)*, pp. 73–88, San Francisco, Calif, USA, April 2008.
- [17] R. Murty, J. Padhye, A. Wolman, and M. Welsh, "Dyson: an architecture for extensible wireless LANs," in *Proceedings of the USENIX Annual Technical Conference (ATC '10)*, pp. 1–14, Boston, Mass, USA, June 2010.
- [18] L. Suresh, J. Schulz-Zander, R. Merz, and A. Feldmann, "Demo: programming enterprise WLANs with odin," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 279–280, 2012.
- [19] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Huhn, and R. Merz, "Programmatic orchestration of WiFi networks," in *Proceedings of the USENIX Annual Technical Conference (ATC '14)*, pp. 347–358, Philadelphia, Pa, USA, June 2014.

- [20] D. Zhao, M. Zhu, and M. Xu, "Supporting 'One Big AP' illusion in enterprise WLAN: an SDN-based solution," in *Proceedings of the 6th International Conference on Wireless Communications and Signal Processing (WCSP '14)*, pp. 1–6, Hefei, China, October 2014.
- [21] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: a survey and taxonomy," *IEEE Communications Surveys & Tutorials*, 2016.
- [22] H. Yao, C. Qiu, C. Zhao, and L. Shi, "A multicontroller load balancing approach in software-defined wireless networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 454159, 8 pages, 2015.
- [23] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.
- [24] A. Tootoonchian and Y. Ganjali, "HyperFlow: a distributed control plane for OpenFlow," in *Proceedings of the Internet Network Management Workshop/Workshop on Research on Enterprise Networking (INM/WREN '10)*, pp. 1–6, San Jose, Calif, USA, April 2010.
- [25] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 136–141, 2013.
- [26] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 7–12, 2013.
- [27] R. Riggio, S. Kuklinski, T. Rasheed, D. Andreatta, D. Tacconi, and F. Antonelli, "Up in the clouds: a taxonomical analysis of network management functionalities from a network as a service perspective," in *Proceedings of the 18th European Wireless Conference (EW '12)*, pp. 1–8, Poznan, Poland, April 2012.
- [28] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: scaling flow management for high-performance networks," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 254–265, 2011.
- [29] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 351–362, 2010.
- [30] Open Network Operating System, <http://onosproject.org/>.
- [31] OpenDaylight, <https://www.opendaylight.org/>.
- [32] K. Guo, S. Sanadhya, and T. Woo, "ViFi: virtualizing WLAN using commodity hardware," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 3, pp. 41–48, 2014.
- [33] PC Engines APU1D4 System Board, <http://www.pceingines.ch/apu1d4.htm>.
- [34] PC Engines wle200nx 802.11a/b/g/n miniPCI express radio, <http://www.pceingines.ch/wle200nx.htm>.
- [35] Raspberry Pi 1 Model B+, <https://www.raspberrypi.org/>.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

