WiFi Network Virtualization to Control the Connectivity of a Target Service

Kiyohide NAKAUCHI and Yozo SHOJI

This paper proposes a WiFi network virtualization technique to control the connectivity of a target service. While WiFi is a promising wireless access system that can accommodate a large quantity of mobile data traffic offloaded from cellular networks, congestions in WiFi networks and concurrent significant connectivity degradation in applications are still open issues. This paper addresses this issue by provisioning dedicated base station (BS) resources to the target service and allowing only the corresponding terminals to associate with the BSs. Simulation results show that the proposed technique can control the delay violation ratio of a target VoIP service.

1 Introduction

With the recent rapid increase in mobile data traffic, WiFi is becoming increasingly important as a system that can offload traffic from cellular networks. In the future, WiFi is also expected to be used in applications with severe delay requirements, such as VoIP or other audio services and cyber physical systems (CPS). However, network congestions frequently occur in WiFi areas crowded with users, causing significant connectivity degradation in applications. In order to use WiFi in applications with severe delay requirements, quality of service (QoS) control in WiFi will be essential.

IEEE 802.11e[1] has been standardized as a QoS control method in WiFi, but it is not widely used today since there is a need to add a mechanism to write class-of-service (CoS) information in the packet header. As another solution, network virtualization, which has been discussed for wired networks[2]–[5], is drawing attention and its application to WiFi is being considered[6]–[7]. Nevertheless, the need to consider the distinct properties of the wireless media such as time-varying channel capacity, interference and broadcast makes it more challenging to apply network virtualization directly to WiFi[8]–[10]. Traditional work has proposed a WiFi base station (BS) virtualization technique[11]–[16] for making shared use of a WiFi BS among multiple operators, but because it targets individual BSs and lacks inter-BS cooperation mechanisms, some issues still remain including the reduced efficiency of utilization of BS resources in the entire WiFi network and service disruption during handover between BSs.

This paper proposes a WiFi network virtualization technique to control the connectivity of a target service. The packet-level delay violation ratio of the target service to be prioritized can be reduced even when the WiFi network is in a congested situation by provisioning dedicated BS resources (a set of dedicated BSs) to the target service and allowing only the corresponding terminals to associate with the BSs. The proposed technique configures multiple physical BSs to use the same MAC address, so that the control functions for BS selection and handover are separated from the BSs and terminals and put together into a centralized controller in the network. Further, by cooperatively configuring consistent layer-2 data paths for the terminals in a BS backhaul network at the time of terminal connection, the technique can fully automate the necessary network configuration.

This paper is organized as follows. In Section **2**, the concepts of WiFi network virtualization and a virtual BS (vBS) are described. In Section **3**, as details of the proposed WiFi network virtualization architecture, a function model, the principle of configuring a service-specific vBS, and the principle of association with the service-specific vBS are presented. In Section **4**, a simulation-based evaluation of the effects of introducing the proposed technique is presented, and in Section **5**, a proof-of-concept prototype developed using off-the-shelf WiFi interface modules and commercial OpenFlow[17] switch hardware is described. Finally, the paper is concluded in Section **6**.

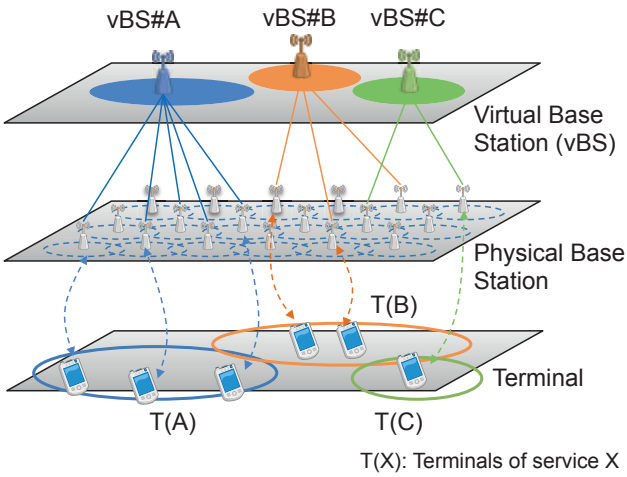


Fig. 1 A Generalized model of WiFi network virtualization

2 WiFi network virtualization

Figure 1 shows a generalized model of WiFi network virtualization. WiFi network virtualization is a technique in which physical WiFi network infrastructure resources and physical radio resources can be abstracted and isolated so as to enable multiple independent and customizable logical (virtual) WiFi networks on common WiFi network equipment[6][8]–[10][18]. In this paper, we call a set of physical WiFi network infrastructure resources and physical radio resources a BS resource for simplicity. In addition, as shown in Fig. 1, we define a set of BS resources provided by multiple physical BSs as a virtual BS (vBS). In other words, a vBS is equivalent to a multi-channel BS. Moreover, we define a vBS that dedicates all its own BS resources to a target service as a service-specific vBS[19][20].

In general, BS selection and handover decisions in WiFi are terminal-initiated based on the vendor-specific algorithms implemented in terminals using the signal strength and other information detected by the terminal. Therefore, it has been difficult to identify terminals of a target service, and guide only those terminals to a specific BS constituting a service-specific vBS. The distinct advantage of the proposed WiFi network virtualization technique is that the controller can fully manage the decisions on BS selection and handover of all terminals without depending on vender-specific algorithms that differ by terminal. Another advantage is that it can ensure equivalent communication quality for communication performed through terminals that are not IEEE 802.11e-compliant.

The BS association and handover procedures naturally go together with layer-2 routing (re)configurations in the BS backhaul. Cooperative and fast layer-2 path reconfiguration is essential for reducing handover latency and avoiding packet drops during handover. Thus, in this study, OpenFlow[17] is exploited for this purpose.

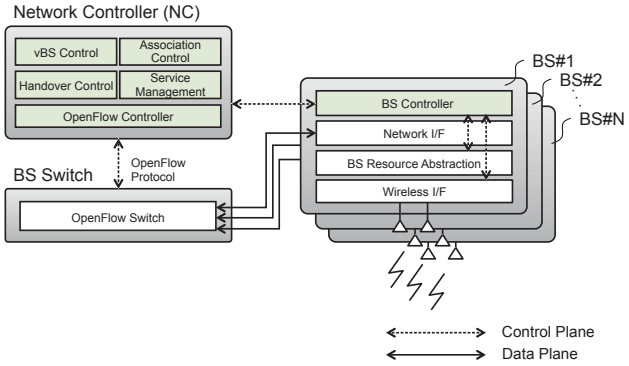


Fig. 2 Function model

Note that the impact of adjacent channel interference (ACI) should be considered when configuring a multi-channel vBS by using multiple adjacent physical BSs. For example, it has been reported that when Channel 36, 40, and 44 in the 5 GHz band are used simultaneously at three BSs, the throughput of the BSs is degraded by more than 50% at maximum[21]. However, a technique to decide the transmission timing in coordination between BSs and to perform pseudo time-sharing of the BS resources can mitigate the impact of ACI.

3 Principle

3.1 Function model

Figure 2 shows a function model of the proposed WiFi network virtualization architecture[22]. The proposed architecture has three main elements: the Network Controller (NC); the BS Switch (BS-SW); and a set of WiFi base stations (BSs). The NC is a centralized controller in a WiFi network and responsible for making decisions on association and handover for all the BSs and terminals under its control. The BS-SW is a programmable switch that has the capability of configuring the forwarding table for the specific flows defined by the 5-tuple of the packet header based on the commands from the NC, which also plays the role of an OpenFlow controller. The BS resource abstraction function is the characteristic function in a BS. By concealing the physical configuration of BSs against terminals, the BS resource abstraction function enables a logically integrated configuration where multiple BSs organize a service-specific vBS. The proposed architecture does not depend on a specific transmission mode of IEEE 802.11, and should also work with the recent IEEE 802.11ac and 802.11ad

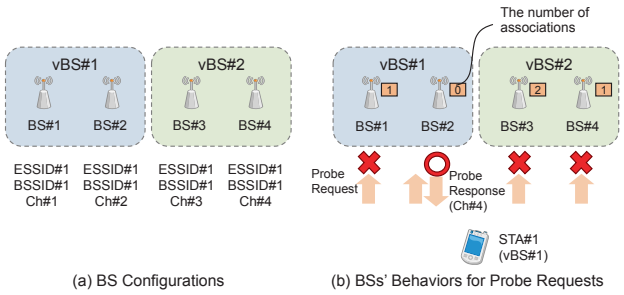


Fig. 3 A Principle of configuring a vBS

modes in principle.

The proposed WiFi network virtualization is realized by the following three principles: the principle of configuring a vBS; the principle of association with a vBS; and the principle of seamless handover between vBSs. In this paper, we explain the principle of configuring a vBS and the principle of association with a vBS. The principle of seamless handover between vBSs is needed when switching the vBS in line with the registration or switching of the service which a terminal uses, but for details of this principle, see Reference [22].

3.2 Principle of configuring a vBS

Figure 3 shows the principle of configuring a vBS. Figure 3(a) shows how BSs are configured to organize two service-specific vBSs with different target services. vBS#1 and vBS#2 are organized by BS#1 and BS#2 and by BS#3 and BS#4, respectively. These four physical BSs are configured with different channels. A characteristic point is that all four physical BSs are configured with the same MAC address, and hence with the same BSSID (Basic Service Set Identifier). The purpose for this is to separate the control logics for BS selection and handover from BSs and terminals and to enable the NC to perform centralized control and decisions instead. In the same way, the same ESSID (Extended Service Set IDentifier) is configured at all of the BSs. As a result, beacon frames transmitted by these BSs contain the same source MAC address and BSSID, but only with a different channel number in the BS parameter set field.

Meanwhile, Figure 3(b) shows how Probe Request frames for active scan are handled at BSs with the above configuration. In this figure, it is assumed that STA#1 is already bound to vBS#1. The basic behavior of the terminal to discover available BSs with the target ESSID is the same as that in a traditional WiFi network. Specifically, at the beginning, STA#1 broadcasts a Probe Request containing

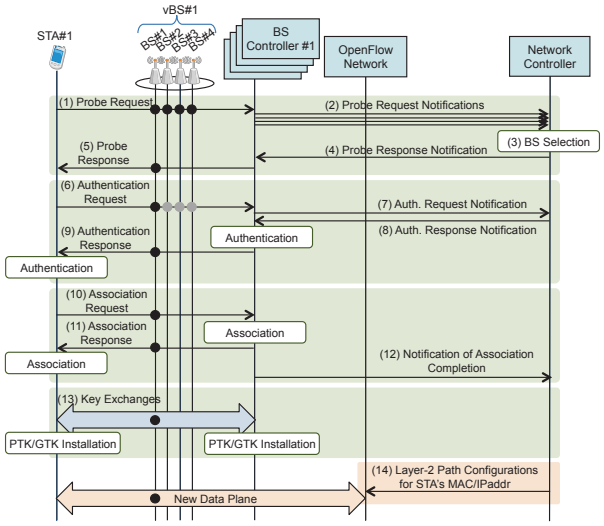


Fig. 4 A Principle of association with a vBS

the target ESSID in the frame body on a particular channel. If STA#1 does not receive any Probe Response frame in a timeout interval (MaxChannelTime) on that channel, STA#1 retries to discover available BSs on another channel in the same way. Because each BS that receives a Probe Request from STA#1 does not return a Probe Response without receiving an instruction from the NC, all the BSs in the proximity of STA#1 eventually receive Probe Requests from the STA on different channels.

Different from traditional WiFi BSs which immediately return a Probe Response by their own decision when they receive a Probe Request, in the proposed architecture the NC decides whether or not the BSs should respond to a Probe Request. In other words, the NC decides on which BS should be associated with the terminal that has sent the Probe Request. Since only the BS that receives an instruction to respond from the NC returns a Probe Response, STA#1 eventually finds one available BS with the target ESSID. Figure 3(b) shows how BS#2 is selected based on a BS selection algorithm from among the BSs organizing vBS#1 and returns a Probe Response. In this figure, an algorithm to select the BS with the least number of associations is adopted in order to balance the associations among the BSs. This BS selection algorithm can be replaced with other algorithms such as those based on the traffic amount or the RSSI (Received Signal Strength Indicator) of each BS.

3.3 Principle of association with a vBS

Figure 4 describes the principle of association with a vBS. Specifically, the figure explains the series of procedures from active scan (transmission of a Probe Request) by a

terminal to authentication, association, key exchange, and path configuration in the BS backhaul linked with these procedures. In this section, vBS#1 is assumed to consist of four BSs (BS#{1,2,3,4}) and these BSs are configured with different channels. The BS controller in Fig. 4 explicitly indicates a software module that performs processes such as authentication, association, and radio interface configuration in a BS. We redefine a set of a BS and a corresponding BS controller as a BS in this section for simplicity. An OpenFlow network in Fig. 4 is an abstraction of the BS-SW. In the proposed architecture, the BS-SW does not need to be a single switch, and it can be replaced by an OpenFlow-based network organized by multiple OpenFlow switches.

In the active scan procedure, each BS that receives a Probe Request from the terminal transmits a Probe Request Notification, including the MAC address of the terminal acquired from the Probe Request, to the NC, instead of returning a Probe Response to the terminal (Steps (1) and (2)). If the NC receives Probe Request Notifications for the terminal from multiple BSs within a specific time period, it chooses one BS according to the BS selection algorithm mentioned in the previous section (Step (3)). We call the selected BS the target BS. The target BS, which receives a Probe Response Notification for the terminal from the NC immediately returns a Probe Response when it receives a Probe Request from the terminal the next time (Steps (4) and (5)).

Similarly, in the authentication procedure, only the target BS that receives an Authentication Response Notification from the NC can return an Authentication Response to the terminal (Steps (6)–(9)). The reason that the BS sends a notification to the NC again in the authentication procedure is to be able to accommodate a terminal that does not perform an active scan in order to restrain power consumption. However, the association procedure that follows is only implemented by the terminal that has completed the authentication procedure. Therefore, the target BS that receives the Association Request immediately returns an Association Response to the terminal (Steps (10)–(12)).

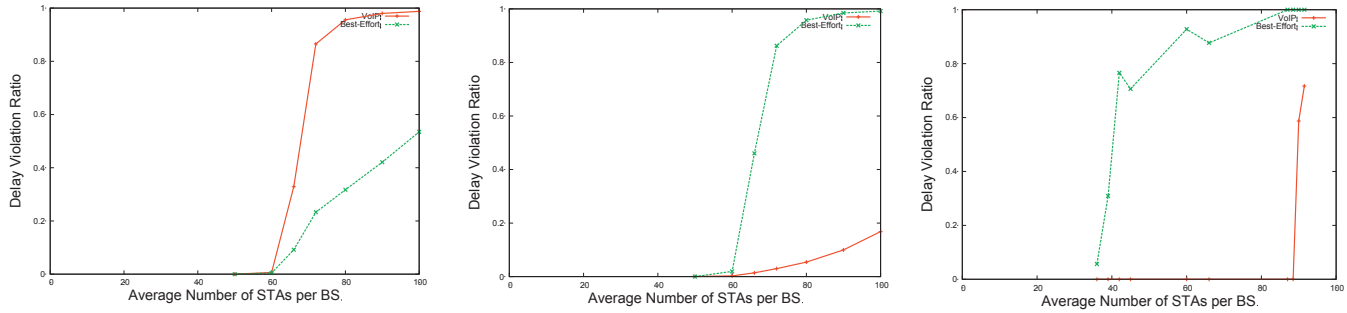
The key exchange procedure is implemented based on IEEE 802.11i[23], similar to the procedure in traditional WiFi (Step (13)). Specifically, key exchange and frame encryption are performed based on WPA2 (WiFi Protected Access 2) CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol). A pairwise transient key (PTK) is shared between a target BS and a terminal by a four-way handshake. A group temporal key (GTK) for encrypting multicast and broadcast messages is also shared between the target BS and the terminal by a periodical two-way handshake.

Finally, the NC configures layer-2 paths for the target terminal in the OpenFlow network in order to establish a data plane from the terminal to the gateway (GW) located upstream of the BS-SW (Step (14)). The OpenFlow controller in the NC registers the flow entries for the terminal in the flow table in the BS-SW by using a Flow_Mod message based on the OpenFlow protocol. For example, in the configuration shown in Fig. 3(b), the two flow entries shown below are registered in the flow table in the BS-SW. Here, MAC_STA1 and MAC_GW are the MAC addresses of STA#1 and GW, respectively. Meanwhile, Port_to_BS2 and Port_to_GW denote the physical ports to which BS#2 and GW are associated, respectively, in the OpenFlow switch (BS-SW). In order to also support communications between terminals belonging to the same vBS, MAC learning and ARP (Address Resolution Protocol) are also enabled at the OpenFlow switch.

```
Flow entry 1:
    Match:  SrcMAC = MAC_STA1
            && DstMAC = MAC_GW
    Action: SendOutPort(Port_to_GW)
Flow entry 2:
    Match:  DstMAC = MAC_STA1
    Action: SendOutPort(Port_to_BS2)
```

4 Reduction of delay violation ratio

The effectiveness of the proposed WiFi network virtualization architecture is assessed by evaluating the packet-level delay violation ratio (DVR) of a Voice over IP (VoIP) service when the WiFi network is congested because of the mix of traffic from VoIP and best-effort (BE) services. The DVR is a particularly effective indicator for evaluating the performance of delay-sensitive applications such as VoIP. The DVR of the proposed architecture is compared with that of the standard access mechanism, IEEE 802.11 DCF, and that of the standard QoS mechanism, IEEE 802.11e EDCA. By using an event-driven simulator, QualNet[24], a network is configured with four BSs and 400 still terminals, and an environment is created where terminals (stations; STAs) of the VoIP service and STAs of the BE service are mixed and congested at a WiFi hotspot. The G.729 codec is assumed to be used for the VoIP service, and traffic is generated based on it with a 60-byte packet size (consisting



(a) IEEE 802.11 DCF (b) IEEE 802.11e EDCA (c) WiFi Network Virtualization

Fig. 5 Delay violation ratio of VoIP services

of a 20-byte payload, an 8-byte RTP header, a 12-byte UDP header and a 20-byte IP header) and a fixed 20-ms packet interval. On the other hand, UDP-based traffic with a 1,500-byte packet size and a 100-ms packet interval with exponential distribution is generated from BE STAs.

In the case of a VoIP service, the maximum acceptable mouth-to-mouth delay for voice is of the order of 250 ms. Assuming that the delay budget for the core networks is approximately 100 ms, the tolerable delay for MAC buffering and scheduling in the wireless link is 150 ms. Hence, assuming that both of the end STAs conducting VoIP communication use WiFi, the tolerable delay in the wireless link would be 75 ms or less. There is a report that, to improve the voice quality further, the desirable delay in the wireless link should be 50 ms or less[25]. Therefore, we fix the threshold for delay violation at 50 ms in the simulations. For comparison, the same threshold is used for the BE service.

In the proposed WiFi network virtualization architecture, a VoIP-specific vBS that only accommodates VoIP STAs is created, and three among the four BSs are exclusively allocated to this vBS. The NC adopts a BS selection algorithm whereby VoIP STAs are equally distributed to those three BSs. The BE STAs are concentrated to the remaining BS. On the other hand, in IEEE 802.11 and IEEE 802.11e, all of the VoIP and BE STAs are equally distributed to the four BSs based on the normal RSSI-based BS selection. In IEEE 802.11e, VoIP traffic and BE traffic are accommodated by AC_VI and AC_BE classes, respectively. The ratio of the number of VoIP STAs and the number of BE STAs is 1:1. The four BSs are configured with different channels in order to avoid interference.

Figure 5 shows the DVRs of the VoIP service and the BE service in the IEEE 802.11 DCF, the IEEE 802.11e EDCA, and the proposed WiFi network virtualization architecture. The X-axis shows the average number of STAs per BS. In Figure 5(a) and (b), after the number of STAs exceeds 60, the DVR increases. At this time, the increase is slower for IEEE 802.11e than for IEEE 802.11e EDCA as a result of performing the preferential transmission process for VoIP packets.

On the other hand, in Fig. 5(c), the DVR of the VoIP service is almost zero until the number of STAs per BS exceeds 88.5 (network load = 0.53). In other words, until the number of VoIP STAs reaches the capacity of the VoIP-specific vBS (in this case, 177 STAs in total in the three BSs), the DVR can be controlled. Based on the simulation results above, we can draw the following conclusions. First, if we define the network load for which the DVR of the VoIP service can be controlled as the vBS capacity, the proposed WiFi network virtualization architecture can improve the vBS capacity by 47%. Second, the proposed architecture can achieve lower DVR for the VoIP service than IEEE 802.11e by 9% at maximum, and can provide a comparable level of connectivity also for STAs that are not IEEE 802.11e-compliant.

5 Proof-of-concept prototype

To prove the concept, we develop a WiFi network virtualization prototype system configured with two virtualization-capable BSs (vcBSs) which are capable of constituting a vBS, one virtualization-capable BS switch (vcBS-SW) which is equivalent to a BS-SW, and an NC. Figures 6 and 7 show the appearance of the prototype system and the vcBS software configuration, respectively. The vcBS hardware consists of a pre-processing PC, a main PC, and four commercial WiFi modules so that a single vcBS can form a vBS configured with four BSs at maximum. Considering the scalability of the number of WiFi modules, the OpenFlow function, which is responsible for distributing traffic among BSs, is isolated to the pre-processing PC.

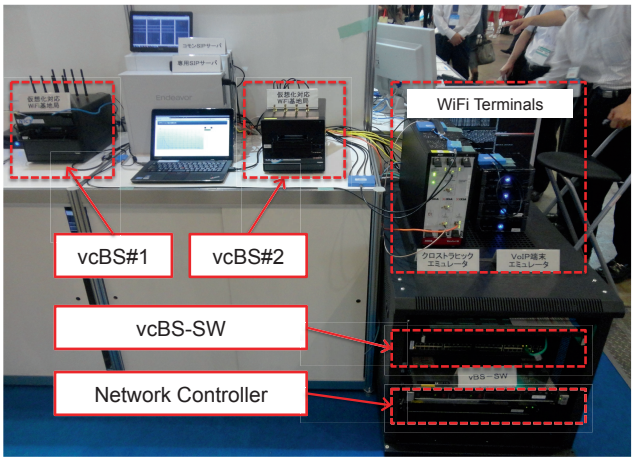


Fig. 6 External appearance of the prototype system

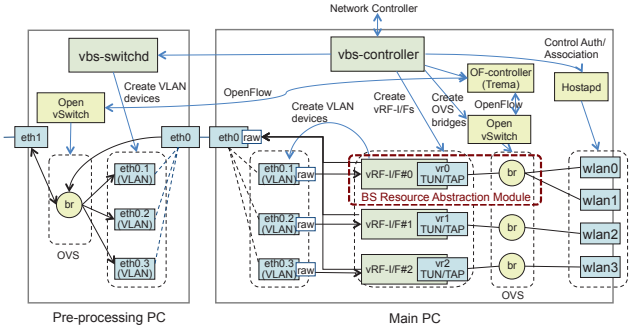


Fig. 7 Software architecture of the vcBS

The PCs are connected by Gigabit Ethernet (GbE), and four WiFi modules are connected to mini PCIe slots in the main PC. The pre-processing PC is equipped with a Dual-Core 1.33 GHz ARM processor and 1 GB RAM, while the main PC is equipped with a Quad-Core 2.8 GHz Intel Core i5 processor and 8 GB RAM. To achieve seamless handover among the BSs within a vcBS and to instantly establish a data plane at the time of association of terminals as described in Reference [22], OpenFlow software is also implemented in the main PC. The OpenFlow function of the main PC and the pre-processing PC use the open-source Open vSwitch (OVS). The WiFi modules support IEEE 802.11a/b/g/n, 2.4 GHz/5 GHz dual band, and 2 × 2 MIMO. Modules with an Atheros AR9280 chipset are selected since the device driver is readily available for Linux OS.

We adopt a commercial OpenFlow switch as the vcBS-SW that has 48 GbE ports. Because all traffic is concentrated to this point, a hardware-type switch is selected instead of an OVS, in order to secure transfer performance. The OpenFlow switch supports OpenFlow 1.0 and can hold a maximum of 2,048 flow entries at a time. Finally, the NC is implemented on an x86 IA server (Linux OS) as application software.

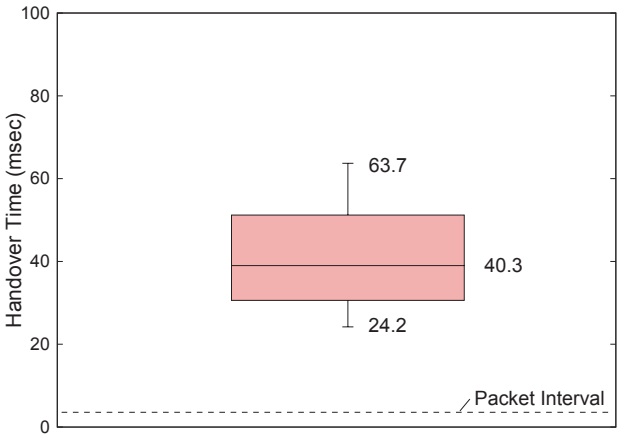


Fig. 8 The Delay of seamless handover to a service-specific vBS

The OpenFlow controller function for managing the vcBS-SW is implemented using the open-source Trema[26]. Figure 8 shows the boxplot of the measurement of the time required for seamless handover from a common vBS to a service-specific vBS. We conducted the handover and measurement ten times. The median value was 40.3 ms and the maximum value was 63.7 ms. We also confirmed that no packet drop occurred during the series of handover operations through packet capture in the wireless link. The results imply that the impact of handover operations on delay-sensitive applications such as VoIP was limited.

6 Summary

This paper proposed a WiFi network virtualization technique to control the connectivity of a target service. Specifically, this study introduced the concept of a virtual base station (vBS), a logical multi-channel base station organized through logical integration of multiple BSs on a physical WiFi network. The proposed architecture can achieve a lower packet-level delay violation ratio in a congested WiFi environment by allocating dedicated BS resources to the target service and allowing only the terminals using the target service to use those BS resources. Since traditional WiFi network virtualization architecture logically divided the wireless resources of a single physical BS, improving the efficiency of utilization of BS resources in the entire WiFi network and addressing the temporary suspension of service during handover between BSs remained as issues. In contrast, the proposed architecture logically integrates the wireless resources of multiple physical BSs, and addresses those issues by centralizing the decisions on BS selection and handover to the NC, and cooperatively configuring consistent layer-2 data paths in

the BS backhaul. The delay violation ratio of the target VoIP service when VoIP traffic competes with UDP-based best-effort traffic was evaluated by simulation. Simulation results showed that the proposed technique could achieve a delay violation ratio lowering effect also for a terminal that is not IEEE 802.11e-compliant to a level comparable to an IEEE 802.11e-compliant terminal. To prove the principle above, a WiFi network virtualization prototype system was developed. The results of an experiment using the prototype system showed that the handover from a common vBS to a service-specific vBS can be completed in less than 65 ms without any packet drop, and confirmed that the proposed architecture can be practically used also in delay-sensitive applications such as VoIP. As future work, we intend to thoroughly evaluate the proposed architecture in a more realistic environment and demonstrate its effectiveness in an environment where terminals nomadically move between base stations.

Acknowledgments

The authors would like to thank the members of the New Generation Network (NWGN) Laboratory in NICT for constructive discussions. Special thanks go to Dr. Makoto IMASE, Prof. Masayuki MURATA, and Dr. Nozomu NISHINAGA for their insightful comments and suggestions.

References

1 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications; Amendment: Medium Access Control (MAC) Quality of Service Enhancements. IEEE Std 802.11e-2005, Nov. 2005.

2 Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner, "Overcoming the Internet Impasse through Virtualization," IEEE Computer, 38(4), April 2005.

3 Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, and Jennifer Rexford, "In Vini Veritas: Realistic and Controlled Network Experimentation," Proc. ACM SIGCOMM '06, Sept. 2006.

4 Akihiro Nakao, "Network Virtualization as Foundation for Enabling New Network Architectures and Applications," IEICE Trans. Communications, E93-B(3):454–457, 2010.

5 N.M. Mosharaf, Kabir Chowdhury, and Raouf Boutaba, "A Survey of Network Virtualization. Computer Networks," 54(5), April 2010.

6 Sanjoy Paul and Srini Seshan, "Virtualization and Slicing of Wireless Networks," GENI Design Document 06-17, GENI Wireless Working Group, Sept. 2006.

7 Gregor Schaffrath, Christoph Werle, Panagiotis Papadimitriou, Anja Feldmann, Roland Bless, Adam Greenhalgh, Andreas Wundsam, Mario Kind, Olaf Maennel, and Laurent Mathy, "Network Virtualization Architecture: Proposal and Initial Prototype," Proc. ACM VISA '09, Aug. 2009.

8 Xin Wang, Prashant Krishnamurthy, and David Tipper, "Wireless Net- work Virtualization," Proc. ICNC '13, Jan. 2013.

9 Chengchao Liang and Fei Richard Yu, "Wireless Network Virtualization: A Survey," Some Research Issues and Challenges. IEEE Comm. Surveys and Tutorials, 16(3), July 2014.

10 Mao Yang, Yong Li, Depeng Jin, Lieguang Zeng, Xin Wu, and Athanasios V. Vasilakos, "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey," Mobile Networks and Applications, Sept. 2014.

11 Bernard Aboba, "Virtual Access Points," IEEE document, IEEE 802.11- 03/154r1, May 2003.

12 Gregory Smith, Anmol Chaturvedi, Arunesh Mishra, and Suman Banerjee, "Wireless Virtualization on Commodity 802.11 Hardware," Proc. WiNTECH '07, Sept, 2007.

13 Rajesh Mahindra, Gautam Bhanage, George Hadjichristofi, Ivan Seskar, Dipankar Raychaudhuri, and Yanyong Zhang, "Space Versus Time Separation For Wireless Virtualization On An Indoor Grid," Proc. NGI '08, April 2008.

14 Gautam Bhanage, Dipti Vete, Ivan Seskar, and Dipankar Raychaudhuri, "SplitAP: Leveraging Wireless Network Virtualization For Flexible Sharing Of WLANs," Proc. IEEE GLOBECOM '10, Dec. 2010.

15 Eiji Miyagaki and Akihiro Nakao, "Cache Sharing Method Using IEEE 802.11 Wireless Access Points for Mobile Environment," Proc. IEEE ICC '11, June 2011.

16 Kiyohide Nakauchi, Yozo Shoji, and Nozomu Nishinaga, "Airtime-based Resource Control in Wireless LANs for Wireless Network Virtualization," Proc. ICUFN '12, July 2012.

17 Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communications Review, 38(2), April 2008.

18 Fangwen Fu and Ulas C. Kozat, "Stochastic Game for Wireless Network Virtualization," IEEE/ACM Transactions on Networking, 21(1), Feb. 2013.

19 Yozo Shoji, Manabu Ito, Kiyohide Nakauchi, Zhong Lei, Yoshinori Kitatsuji, and Hidetoshi Yokota, "Bring Your Own Network —A Network Management Technique to Mitigate the Impact of Signaling Traffic on Network Resource Utilization—," Proc. MobiWorld 2014, Jan. 2014.

20 Kiyohide Nakauchi, Yozo Shoji, Manabu Ito, Zhong Lei, Yoshinori Kitatsuji, and Hidetoshi Yokota, "Bring Your Own Network —Design and Implementation of a Virtualized WiFi Network—," Proc. IEEE CCNC'14, Jan. 2014.

21 Yozo Shoji and Takeshi Hiraguri, "Virtualization-capable Multichannel WiFi System with a Coordinated Downlink Transmission Technique," Proc. 9th International Symposium on Communication Systems, Net- works and Digital Signal Processing (CSNDSP), July 2014.

22 Kiyohide Nakauchi and Yozo Shoji, "WiFi Network Virtualization to Control the Connectivity of a Target Service," IEEE Transactions on Network and Service Management, 12(2), pp.308–319, June 2015.

23 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Std 802.11-2012, March 2012.

24 Qualnet. http://web.scalable-networks.com/content/qualnet/.

25 Syed A. Ahson and Mohammad Ilyas, "Voip Handbook: Applications, Technologies, Reliability, and Security," CRC Press, Dec. 2008.

26 Trema. http://trema.github.io/trema/.

Kiyohide NAKAUCHI, Ph.D.

Senior Researcher, New Generation Network Laboratory, Network Research Headquarters
Network Virtualization, Mobile Network

Yozo SHOJI, Ph.D.

Director of Social ICT Laboratory, Social ICT Research Center/Research Manager, New Generation Network Laboratory, Network Research Headquarters
Optical, Wireless, and Microwave-photonics Communications Systems