DOCTOR THESIS

SHIBAURA INSTITUTE OF TECHNOLOGY

Energy-Efficient Heterogeneous Wireless Communication with Extended SDN-Controller

SEPTEMBER 2017

NGUYEN DUC TOAN

Acknowledgements

First of all, I would like to thank my supervisor Prof. Eiji KAMIOKA, who gave me the opportunity to join the "Hybrid Twinning Program" at Shibaura Institute of Technology and supervised me through my thesis. He has always been of great help, encouragement and supported me with everything I needed. Working at his lab was a pleasure since he gave me the freedom to find my own research interests and advised me in every possible way. This work would have never reached completion without his helpful discussions.

I am graceful to Prof. Shuji Kubota at Shibaura Institute of Technology (SIT), Prof. Shigeki Yamada at National Institute of Informatics (NIII), Prof. Hiroaki Morino and Prof. Takumi Miyoshi at SIT for serving on my oral committee. Prof. Kubota, who is my co-supervisor, has advised me both ideas and conducive comments. Prof. Yamada has provided me not only a systematic view of my study, but also instructive comments on every chapter of my dissertation. Prof. Miyoshi gave me useful comments from the SDN viewpoint. Prof. Morino gave me conducive comments on VHO.

I would also like to extend my thanks to all my close colleagues and friends from HUST I have made over the years: Nguyen Manh Linh, Nguyen Kien Trung, Pham Ngoc Hung, Nguyen Duy Dinh, Tran Van Thuc, Nguyen Van Tinh, Tao Ngoc Linh, Tran Thanh Son, Nguyen Tri Cuong, Phan Xuan Tan, and Nguyen Kien. They were often a source of new ideas and their discussions and feedback helped me very much. Particularly, I would like to thank Nguyen Kien, who has always been generous with his time and advised me in many ways. His feedback was always of great help, and it gave me new ideas to think about. I also very much enjoyed the collaboration with Nguyen Manh Linh, our research discussions on electronic circuits.

Besides my research colleagues, I would also like to thank the faculty and staff members of Shibaura Institute of Technology for their help and support. Thanks to all the Mobile Multimedia Communication Laboratory members, especially to Keishi Usa, Manami Kanamaru, Hikari Itani, and Mohamad Sabri, who have been always willing to help and give their best suggestions.

Above all, I am deeply grateful to my family. I would like to thank my brother Nguyen Binh An, and in particular my parents Nguyen Van Son and Nghiem Thi Hoang Mai. They have been always supporting and encouraging me with their best wishes. I am forever grateful and indebted to them for their love and care. I would also like to express my gratitude to my parents-in-law, Nguyen Trong Thuan and Le Thi Thanh, who have always given me a second home. I would like to express my love to my dear sons Nguyen Duc Huy and Nguyen Bao Chau. They become the glowing sun of my life, motivate me to achieve higher goals. Finally and most importantly, my deepest and most sincere thanks belong to my beloved wife Nguyen Thanh Huong. Without her love, support, and smile, I would not have given up before I started. I am, and will always be, grateful to you for your love, your encouragement, and for being the joy of my life. The past year apart from each other has been the toughest and most trying things we could both endure. All I can say is that I am profoundly happy that this time is coming to an end and that we can be together again.

Abstract

In the next generation of wireless networks such as the fifth generation (5G), different radio access technologies (RATs) will be integrated into each mobile device as a fundamental feature, aiming to connect any person using any device at anytime, anywhere. To allow the users to enjoy the ubiquitous connectivity, RATs are generally activated simultaneously. Therefore, the mobile device has to consume higher energy for the simultaneous activation of multiple wireless interfaces as well as the continuous connectivity. Although RATs co-exist in the same environment, they are designed heterogeneously. The technologies that offer high data rate are generally energy-consuming ones while low-energy technologies commonly provide low data rate. When the services run on the device do not always require high data rate, it is not an energy efficient way to keep using high-speed technology. If a low energy technology, i.e., Bluetooth, can be used in replacement of the energy-consuming one, i.e., Wi-Fi, the energy consumption of the wireless communication can be saved. It is obvious that the consumed energy will be saved most when the unused technology is turned off. To migrate between different technologies without disrupting any ongoing services, the network-layer connectivity must be maintained, or a vertical handover (VHO) is required.

The VHO process is generally controlled by a centralized control. In many occasions, the centralized control center does not function properly affecting the mobility management's execution. For instance, in catastrophic disasters like the East Japan Earthquake, crucial network infrastructures were destroyed, causing the centralized control to be isolated. Thereby, the VHO should be controlled by the mobile device itself to be aware of the environment. The mobile device typically navigates traffic through the firmware of the wireless network interface cards (WNIC) using their drivers, which are typically dependent on the vendors. To be aware of the vendors, the control of the traffic navigation between WNICs should not be relied on any modification of the WNICs' drivers.

Not to modify WNICs' driver, network traffic should be under control of software as introduced by Software-Defined Networking (SDN). Unfortunately, the control in existing SDN architecture ends at the network switches or routers. When applying the SDN architecture

in a mobile device, a traditional SDN-controller (traditional-SDNC) become a local controller. The traditional-SDNC still directly controls a virtual OpenFlow switch, which turns WNICs into its ports. However, traditional-SDNC is deployed locally, it lacks the global view of the network topology and cannot navigate the traffic correctly. To support the traditional-SDNC in controlling traffic, an extended SDN-controller (extSDNC) is proposed.

To control the switch, a traditional-SDNC needs at least information of the network such as the network topology. To learn the network topology, the network information including IP and MAC addresses must be exchanged among the devices. Therefore, the first feature that needs to be enhanced for the traditional-SDNC is a controller-to-controller (C2C) communication. Since the traditional-SDNC's design is to talk only with the networking devices, i.e., OpenFlow switches, the messages exchanged in a C2C communication must go through the switches. Unfortunately, in the OpenFlow specification, there is no option to allow a traditional-SDNC to send any messages through switches to another traditional-SDNC. The traditional-SDNC needs a network application (nwApp) to instruct the switches to forward and receive a message to another traditional-SDNC. Note that nwApp in SDN architecture is mainly to steer the switches through a traditional-SDNC. Therefore, it is recommended to use a non-SDN software, or an extended SDN controller (extSDNC), to play the role of exchanging C2C's messages.

To perform a VHO smoothly, the VHO must be triggered properly. For this purpose, different VHO trigger algorithms are introduced for a different context. For instance, in disaster scenarios, VHO can be triggered based on speech pattern recommended by ITU-T to prolong the communication time for victims. In wireless multi-hop networks, VHO between an energy-consuming technology, i.e., Wi-Fi, and a low-energy one, i.e., Bluetooth, can be used to reduce unnecessary energy intermediate nodes spent to relay messages. In this case, VHO is triggered based on bandwidth utilization, i.e., Bluetooth's capacity. In any case, experimental results have confirmed that data traffic was migrated smoothly from any direction between different wireless access technologies.

When using the extSDNC, mobile devices can control data traffic between wireless network interfaces (WNICs) with ease and flexibility. When all WNICs are activated, the traffic will be navigated simply by modifying OpenFlow rules. However, when there is only one WNIC is activated and the others are disabled, to migrate traffic from the activated WNIC to a disabled WNIC, the disabled WNIC firstly is enabled. After that, the enabled WNIC is configured. Although the WNIC is configured, new OpenFlow rules must be installed to navigate traffic to the newly configured WNIC. To reduce the time, the WNIC configuration and OpenFlow installation can be executed in parallel. However, it is not easy to complete two processes at the same time, thus, the difference between the end time of configuring network and that of installing OpenFlow rules is one of the sources of handover delay. To eliminate this delay, the VHO process should be executed after both processes finish. Besides, the VHO execution processes on a device utilizing the link layer event to reduce the handover delay as well as the number of packet loss. However, when both communicating devices have to change the access technology, the different of when they finish the VHO process is another source of handover delay. To reduce this delay, a VHO timing adjustment mechanism is introduced to insure the VHO processes to complete at the same time. Two mentioned solutions are integrated in a proposed framework named esVHO. The esVHO framework executes VHO after the destination WNIC is configured and assigned an IP address, hence, the handover delay as well as the number of packet loss is reduced. Experimental and simulation results have confirmed that esVHO saves energy for the wireless communication with a small number of packet loss.

Contents

List of Figures

List of Table

List of Abbreviations

ICT	Information and Communication Technology
MIMO	Multiple Input Multiple Output
AP	Access Point
WLAN	Wireless Local Area Network
PSM	Power Saving Mode
IEEE	Institute of Electrical and Electronics Engineers
LR-WPANs	Low-Rate Wireless Personal Area Networks
VHO	Vertical Handover
IP	Internet Protocol
MIH	Media Independent Handover
SIP	Session Initiation Protocol
OSI	Open Systems Interconnection
WNIC	Wireless Network Interface Card
RSSI	Received Signal Strength Indicator
SDN	Software-Defined Networking
extSDNC	extended SDN-controller
traditional-SDNC	traditional SDN-controller
SBI	Soundbound Interface
NBI	Northbound Interface
nwApp	Network Application
lt-SDNC	local traditional-SDNC

C2C	Controller-to-Controller
MAC	Media Access Control
ARP	Address Resolution Protocol
RTT	Round-Trip Time
OVS	Open vSwitch
XML	eXtensible Markup Language
SSID	Service Set IDentifier
RAM	Random Access Memory
CPU	Central Processing Unit
ICMP	Internet Control Message Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
FTP	File Transfer Protocol
ADC	Analogue-to-Digital Converter
D2DHWC	Device-to-Device Heterogeneous Wireless Communications
US	United States
DHS	Department of Homeland Security
ITU-T	International Telecommunication Union – Telecommunication
VoIP	Voice over IP
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
RTP	Real Time Protocol
QoS	Quality of Service
PC	Personal Computer
USB	Universal Serial Bus

WPAN	Wireless Personal Area Network
BU	Bandwidth Utilization
DB	Database
LTE	Long-Term Evolution
WiMAX	Worldwide Interoperability for Microwave Access

Chapter 1 Introduction

1.1 Motivation

With the development of Information and Communication Technology (ICT), wireless access technologies and systems have been developed and deployed throughout the world, aiming to provide reliable high-speed wireless accesses from anywhere at any time. On that networks, different types of applications and services to deliver various media types including text, audio, videos, and images to end users have been introducing. To access such diverse multimedia contents, many types of affordable wireless devices including smartphones, tablets, embedded control systems and entertainment devices are emerging. As a result, the number of users that uses mobile devices to access applications and services wirelessly increases exponentially. In 2017, over sixty-five percent of the world's population is mobile users as reported in [1]. The network traffic generated by wireless networks has also grown significantly. For example, web traffic comes from mobile devices is more than a half of the global web traffic. The growing data traffic and the continuity of the ubiquitous accesses have escalated the demand for energy consumption.

The escalation of energy consumption raises the cost of operation and maintenance. For example, when the data rate increases 1000 times [2], an equivalent increase of transmitting/receiving power would lead to an exponential growth of electricity bills and maintenance costs. To avert the escalation of energy consumption due to the data rate increased 1000 times, the energy efficiency should be improved 1000 times.

Besides, the high demand of energy consumption for the rapid expansion of wireless networks would threat environmental conservation. As reported in [3], the global information and communication technology (ICT) infrastructures are accounted for two percent of the worldwide Carbon dioxide (CO2) emissions in 2007. The given ratio will escalate with the aggressive growth of the energy demand. Therefore, the Groupe Speciale Mobile Association (GSMA) demands a reduction of CO2 emissions per connection by 40% by 2020.

In general, energy efficiency can be improved on all technology layers including hardware design improvement, the network planning and deployment strategies, network operation, and management. Many efforts have attempted to give energy efficient hardware solution such as power amplifiers [4, 5], hybrid analog and digital beamforming structures [6, 7 and 8]. Other works consider the planning, deployment, and operation of the wireless networks. For example, the work in [9] analyzed the trade-off between energy efficiency and node densification. The work in [10] densified the number of deployed antennas in a MIMO system while considering the energy efficiency. Since the cost of improving the hardware design or changing the deployment of the network is high and mobile devices can communicate directly without relying on any existing infrastructure, i.e., access point (AP) in wireless local area network (WLAN), many researchers focus on increasing the energy efficiency of the wireless communication system by allocating the communication radio resources [11-15]. The resources are no longer be allocated for maximizing the amount of transmitting information. The considered factor is the amount of energy consumed to reliably deliver the maximized amount of information.

With a fixed infrastructure wireless network, the energy efficiency can increase by improving the power saving mechanism defined in each wireless technology. For example, Wi-Fi, which offers high-speed data rate, has a power saving mode (PSM) [16] that allows the clients to sleep more by buffering packets at the access point (AP). The performance of PSM can be improved as introduced in [17, 18 and 19] by extending the sleeping duration of Wi-Fi. However, Wi-Fi in sleep state still consumes more energy than low energy consumption technologies, i.e., Bluetooth, in the active state. Such an energy-consuming technology should be used less when it is unnecessary.

An alternative approach to saving energy is to reduce the energy consumption. To this end, IEEE 802.15.4 standard [20] defines the operation for Low-Rate Wireless Personal Area Networks (LR-WPANs). The performance of 802.15.4-based approaches can be further improved if an ON/OFF scheduling for the network interface is applied as introduced in [21, 22]. However, 802.15.4 has not been designed to support various types of data traffic with a high data rate. To support such kind of data traffic with low energy consumption, Bluetooth

[23] is commonly used. Although Bluetooth is also optimized for low power consumption, it does not support high data rates, i.e., up to 1Mbps.

Therefore, when the application and services on the mobile devices do not require high bandwidth, high-speed technology, i.e., Wi-Fi, becomes an energy inefficient communication means. Just for example, bandwidth-hungry services such as video streaming do not always require high bandwidth utilization [24]. In this cases, a low-energy technology, i.e., cellular or Bluetooth, should be used in replacement of the energy-consuming one, i.e., Wi-Fi. When data traffic is migrated from one network to another, a vertical handover (VHO) is required to insure the quality, reliability, and continuity of all ongoing services.

The main focus of this dissertation is to save energy for the wireless communication when different types of data are continuously transferred with dynamic change of data rate by performing VHO. The energy saving mechanism must not affect the quality of the current connection as well as all running application and services. The mechanism should not rely on any existing infrastructures or any external supports. Also, it should not require any customization of existing hardware as well as a device driver, or not be depended on any specific hardware vendor.

1.2 Existing technologies constraints and challenges

To maintain all ongoing sessions when migrating data traffic between different networks, VHO is used. Vertical handover techniques have been largely studied in a variety of domains including IP-based wireless networks [25]. Several techniques have been standardized such as Media Independent Handover (MIH) [26], Mobile IP [27], Mobile IPv6 [28] and Session Initiation Protocol (SIP) [29]. However, these approaches operate at the vertical handover at the Network Layer or Transport Layer in the OSI model [30], hence, meet a long delay, i.e., 1500ms [31], in the handover process due to the network overhead. To reduce the handover delay, Astorga [32] attempted to perform the handover at Data Link Layer. In Astorga's work, to support MIH for handover management, the device drivers of wireless interfaces have been modified. All the mentioned works above, however, require an additional network element, i.e., the decision engine in [32], to generate a handover trigger and gather the network topology information.

VHO can also be used for saving energy. For instance, the authors in [33] proposed to save energy consumption for a mobile device when the wireless connection is in an idle state by disabling the Wi-Fi and migrating the data session to the cellular network. When Wi-Fi is activated, the scan and listen intervals are adjusted based on the measured RSSI to save energy consumption and to reduce the network overhead. The proposal, however, does not work well when the wireless interface is awake for continuously transferring data. To this end, the work in [24, 34] activates all wireless technologies simultaneously. The work in [24] considers the scenario in which data traffic is unidirectional one. A VHO in the downlink will occur based on network throughput to save energy. However, the energy consumed by the unused wireless technology has not considered. In addition, the impact of the VHO to the performance of the system has not been well investigated. The work in [34] also saves energy consumption by switching active sessions from Wi-Fi to Bluetooth based on network throughput. The unused technology is left in an idle state so that it can be used immediately without re-associating to its network. Although Bluesaver [24] requires a customized AP to provide both Wi-Fi and Bluetooth connections at the same time as well as a modification in Wi-Fi device driver, the energy consumption of the wireless communication can be saved up to 25%. Although Wi-Fi is in the sleep state and does not take part in the communication, the device still has to consume energy for maintaining the Wi-Fi connection. To this end, several works turn off the Wi-Fi connection when it is not used for saving energy [35, 66]. The work in [35] allows the WLAN interface to be completely turned off when it is in the idle state. The WLAN interface does not need to wake up regularly for checking beacon frame, hence, the energy is saved. Similarly, the proposal in [36] turn off the WLAN interface after a certain time of inactivity. The WLAN interface will be woken up only when the sending/receiving data reaches a certain threshold level. However, existing works [35, 36] assumed that the two different networks are interconnected. Although mentioned works show the possibility of saving energy by performing a VHO, the energy consumed by the unused technologies should be considered. Almost all of above works focus on performing VHO using an infrastructure network, it is necessary to consider the scenario in which there is no infrastructure.

In a non-interconnected environment, the wireless access networks are not connected, thus, the mobile devices have to perform a VHO fully by themselves. For this purpose, they have to control their data traffic navigation. Data traffic is typically navigated through the firmware of the wireless network interface cards (WNICs) using their drivers, which are dependent on the vendors. Thereby, the control of traffic navigation between WNICs commonly requires modification of the WNICs' drivers [24, 37, and 38]. After having full control of data traffic navigation, to perform a VHO, mobile devices must collect enough information for the VHO decision and the VHO execution process. Not to rely on any external support, mobile devices should be able to exchange information to gather all necessary data. In addition, when two communicating devices have to perform a VHO to the same designated network, the VHO processes should be completed at almost the same time to avoid losing packets or interrupting ongoing application and services.

Without any modification of the WNIC's driver, Software-Defined Networking (SDN) [39] is a potential candidate for controlling the WNIC using software, i.e., an SDN-controller. Unfortunately, the control in existing SDN architecture ends at the network switches or routers [70, 71]. To control the incoming and outgoing data traffic on WNIC of mobile devices, the SDN concept can be applied locally [24, 40, and 41]. However, in these approaches, the embedded SDN-controller, which becomes a local SDN-controller, operates passively behind a global SDN-controller. This is because local SDN-controller needs network-wide information such as network topology to correctly control the data traffic inside the devices. This information can be collected easily by the global SDN-controller as it has the global view of the network.

In a series of our previous works [42-47] an extended SDN-controller (extSDNC) and a controller-to-controller (C2C) communication means have been developed to enable the mobile devices to manage many types of data traffic over wireless network interface cards (WNICs), to perform VHO without any external support or modification of WNIC driver and to save energy. The aim of extSDNC [42] is to manage data traffic on a mobile device while the C2C communication means in [45] is to exchange information between extSDNCs on communicating devices. The esVHO framework [47] is to reduce the number packet loss generated by the VHO process and to save energy.

The key element in the previous works is the extSDNC. The next section, hence, will give an overview of extSDNC, highlighting the difference between traditional SDN-controller and extSDNC.

1.3 The extended SDN-controller (extSDNC)

1.3.1 Traditional SDN architecture

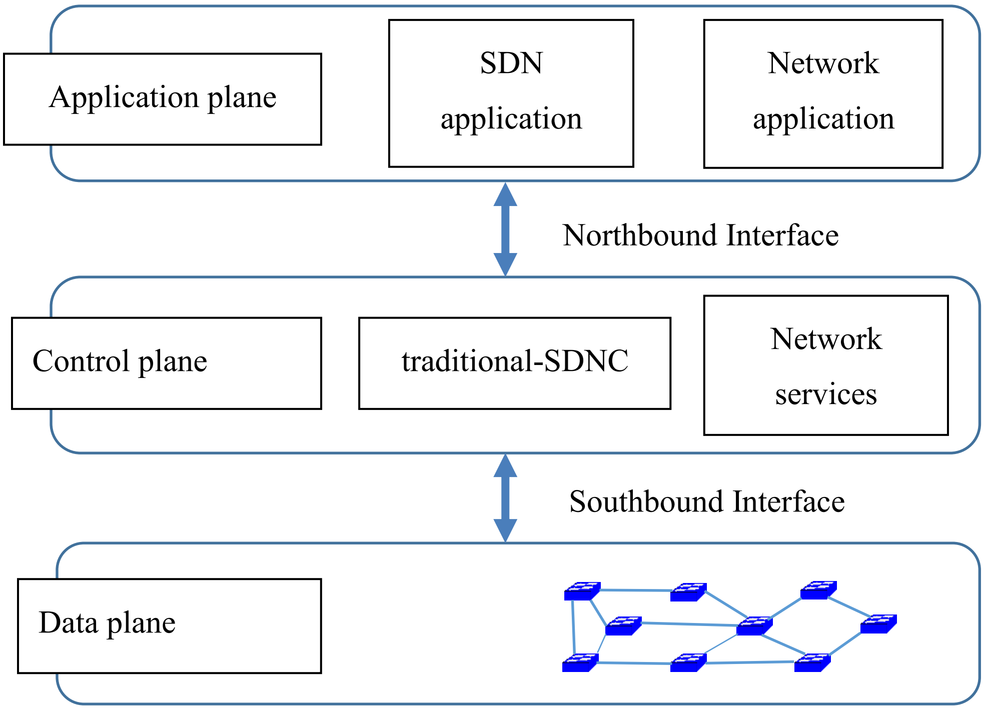


Fig. 1.1: Traditional SDN Architecture

Software-Defined Networking (SDN) has emerged as a flexible way to control the network in a programmatic way. The core idea of SDN is to separate control plane and data plane. In the data plane, networking devices, i.e., switches, simply perform packet forwarding. The control function of network devices is executed by a software program called SDN-controller. The SDN architecture is initially designed in a centralized manner mainly for wired network, especially for data center deployments. The centralized approach allows the SDN-controller to have a global view of the network, hence, it can manage and operate the network with ease. For instance, in a wired network, the connection between any two end devices is a point-to-point full duplex link that can be controlled without modifying the devices.

In SDN architecture, the control and data planes are decoupled as shown in Fig. 1.1. In the data plane, networking devices, i.e., switches, work as simple packet forwarding elements, thus sometimes the data plan is called forwarding plane. The control function of network devices is moved to an external element, the so-called SDN-controller. The traditional SDN-controller (traditional-SDNC) has two key functions. First, it is responsible for controlling or forwarding packets in the data plane. The packets are forwarded in a flow-based mechanism. In the mechanism, all the packets that match the criteria defined in one flow rule will be processed in the same way, forming a flow. To check the matching, when packets reach the networking devices, their header field values are taken out and are compared to the stored flow rules. The flow rules are typically defined by the traditional-SDNC and then are sent to the networking devices via the soundbound interface (SBI). Second, the traditional-SDNC collects status information from networking devices to offer a global view of the network to applications in the application plane. The traditional-SDNC communicates with the application via the northbound interface (NBI). The interface also allows the traditional-SDNC to listen to requirements from each network application (nwApp). After receiving the requirements, the traditional-SDNC interprets them into the instructions, which will be used to control the data plane.

1.3.2 The need of an extended SDN-controller developed separately from traditional-SDNC

As mentioned in the previous section, a traditional-SDNC typically controls the packets that go through the networking devices, i.e., OpenFlow switches. The traditional-SDNC is commonly located separately from the switches and has a global network view including the network topology, link status, etc. Therefore, to control the packets, the controller only needs to talk with the switches. However, when a traditional-SDNC and an OpenFlow switch are embedded in one node of the network, the local traditional-SDNC (lt-SDNC) may not be able to have the global network view due to the network disconnection and it is unable to answer any requests from the switch. Therefore, it is necessary to enrich the features of the lt-SDNC so that it can control the OpenFlow switch.

To control the switch, a lt-SDNC needs at least information of the network such as the network topology. To learn the network topology, the network information including IP and MAC addresses must be exchanged among the devices. Therefore, the first feature that needs to be enhanced for the lt-SDNC is a controller-to-controller (C2C) communication. Since the lt-SDNC's design is to talk only with the networking devices, i.e., OpenFlow switches, the messages exchanged in a C2C communication must go through the switches. Note that the lt-SDNC mainly uses OpenFlow protocol to talk with the switches. In the OpenFlow specification [48], there is no option to allow a lt-SDNC to send any messages through switches to another lt-SDNC. However, it is possible that a lt-SDNC instructs the switches to forward and receive a message, which is not generated by the lt-SDNC, to another lt-SDNC. Since network applications (nwApps) in SDN architecture is to steer the switches through a lt-SDNC. Therefore, it is recommended to use a non-SDN software, or an extended SDN-controller (extSDNC), to play the role of exchanging C2C's messages.

In case a link failure occurs, the proposed system must perform a vertical handover to maintain the end-to-end connection. When performing the vertical handover, all the ongoing traffic are switched from one physical interface to another. This action requires the reconfiguration of the network to detach an access technology and attach to the other one. Since the lt-SDNC and nwApps are targeted at controlling the switches, the extSDNC should be involved in managing the physical network configuration. Besides, the change of physical interface will change the IP and MAC addresses in the packet's header, hence, the system must ensure that any running application is not interrupted during and after the vertical handover process. Not to let the applications to detect the change of physical interface, the system needs to provide a virtual interface with unchanged IP and MAC addresses for them. To this end, the packets, which are exchanged between two end devices, must have those values in their header rewritten. SDN technology already offers the packet's header rewriting feature. However, to handle ARP packets, the extSDNC is again required.

Since a lt-SDNC is a computer software, it can be hung or killed while the other computer software keeps running. When the traditional-SDNC dies, nwApps will lose their interpreter to translate their abstract requirements into the commands, which will be executed on the switches. Different from nwApps, the extSDNC is designed as a separated program from lt-

SDNC, thus, it can be in charge of controlling the switch while waiting for the lt-SDNC to be recovered.

1.4 Contribution and dissertation organization

In this dissertation, the main contributions can be summarized as follows:

(1) An extended SDN-controller (extSDNC) was proposed to embed the SDN architecture into mobile devices, allowing them to well manage various types of incoming and outgoing traffic over different WNICs without relying on any external support or requiring modification of WNIC driver;

(2) The local SDN architecture was implemented using a real testbed;

(3) A controller to controller (C2C) communication means was proposed to allow extSDNCs on communicating devices to exchange information. The C2C communication means is also used for making decision whether or not to trigger a VHO in different scenarios;

(4) A VHO timing adjustment mechanism was proposed to enable the VHO processes on two communicating devices to be finished at almost the same time, reducing the VHO delay as well as the number of packet loss;

(5) An esVHO framework was proposed to improve the performance of extSDNC in term of reducing VHO delay and number of packet loss as well as saving energy.

(6) An energy consumption model was proposed based on direct measurement and was validated under realistic traffic.

The organization of the dissertation is illustrated in Fig. 1.2 and is described as follows:

Chapter 1: Introduction. The motivation and background of this study are represented in this chapter. Additionally, a literature review on existing related technologies was conducted to clearly show the necessity of the proposal in this dissertation. Also, why the extSDNC is separately from traditional-SDNC was also concretely discussed. Finally, the summary of the key contributions in this research was also described in this chapter.

From chapter 2 to chapter 6, the design of extSDNC, the C2C communication means, the energy efficient VHO algorithm and the esVHO framework contributed in this study followed three phases of a VHO process are presented.

Chapter 2: Design and feasibility of extSDNC. This chapter describes the design of the extSDNC and shows how its feasibility is confirmed. Specifically, the performance of extSDNC was evaluated under various protocols in a real testbed.

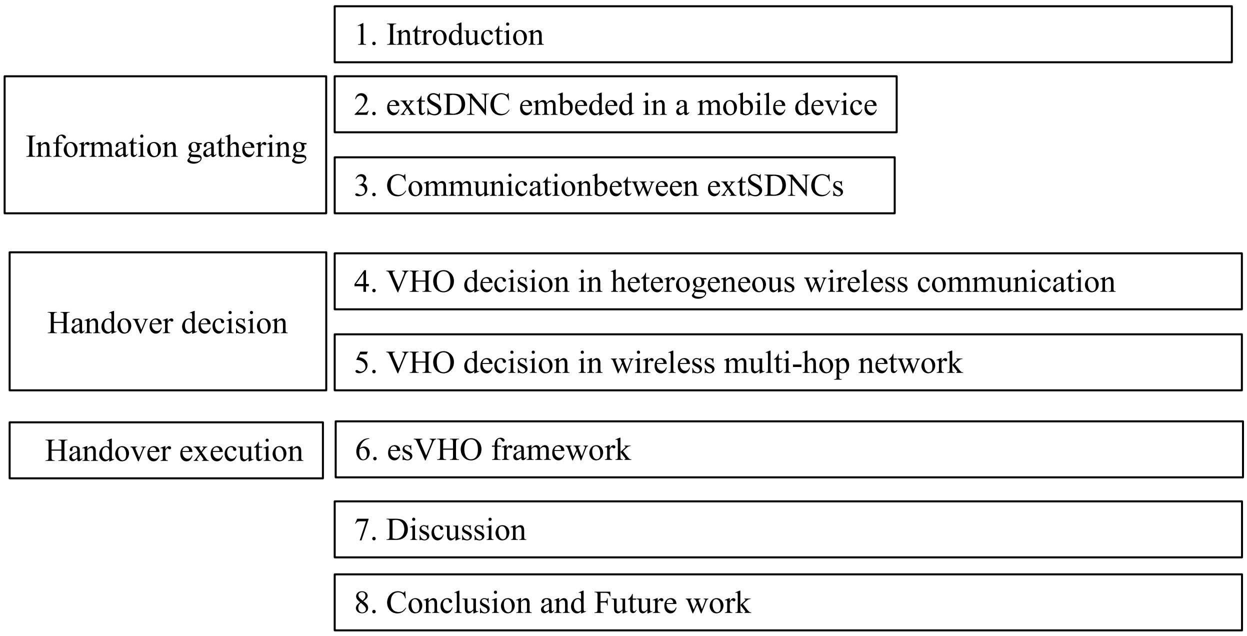


Fig. 1.2: Organization of this dissertation

Chapter 3: Communication between extSDNCs. This chapter presents the C2C communication between extSDNCs embedded on communicating mobile devices. The detail of the message sequences confirms that the VHO processes on two devices can complete at almost the same time, reducing packet loss as well as service interruption. In addition, the energy efficient model shows that performing VHO using extSDNC can save energy for the wireless communication.

Chapter 4 and Chapter 5 discuss the VHO decision in different scenarios. Specifically, in Chapter 4, VHO is triggered based on network throughput, allowing victims in a disaster situation to have a device-to-device heterogeneous wireless communication. The performance of the wireless communication is evaluated using a testbed and the energy efficiency of the system is validated by direct measurement. Similarly, in Chapter 5, a triggering algorithm for energy efficient VHO in wireless multi-hop network is introduced to reduce the energy the intermediate nodes spend to relay messages. The communicating nodes also consume less energy by choosing the least energy-consuming interface for transfer.

Chapter 6: esVHO: energy saving solution for wireless communication. This chapter describes how the number of packet loss generated by the VHO process is reduced by using the esVHO framework. Besides, the energy consumption model based on direct measurement results confirmed the effectiveness of the proposal in term of energy saving. Experimental as well as simulated results have confirmed that esVHO outperforms the others.

Chapter 7: Discussion. This chapter discusses the advantages as well as the opened issues of the proposals in this dissertation.

Chapter 8: Conclusion and Future work. This chapter concludes the dissertation and outlines future work.

Chapter 2 Design and feasibility of extSDNC

This chapter describes the requirement and the design of extSDNC in order to allow mobile devices to perform VHO fully by themselves.

2.1 Design requirements

2.1.1 extSDNC must work with any existing SDN controllers

As mentioned in section 1.3.1, a traditional-SDNC typically controls the packets that go through the networking devices, i.e., OpenFlow switches. The traditional-SDNC is commonly located separately from the switches and has a global network view including the network topology, link status, etc. Therefore, to control the packets, the controller only needs to talk with the switches. However, when a traditional-SDNC and an OpenFlow switch are embedded in one node of the network as shown in Fig. 2.1, the traditional-SDNC may not be able to have the global network view due to the network disconnection and it is unable to answer any requests from the switch. Therefore, the extSDNC must be able to support all existing traditional-SDNCs, which generally have their own design with different APIs, to operate locally on a mobile device.

2.1.1 extSDNC must be to monitor and control network interfaces

In order to provide sufficient information for the traditional-SDNC, the extSDNC must be able to collect network information and analyze them. For example, it collects RSSI values and estimates the distance between nodes. The collected information must include not only the network configuration information such as MAC and IP addresses, access point properties, but also the network traffic information like throughput, round-trip time (RTT). In addition, it needs to have permission to change the state of the physical interfaces. For instance, it can assign a new IP address to any interface, or provide fundamental parameters to a wireless interface so that it will associate with an access point (AP) without user interaction.

2.2 The design of extSDNC

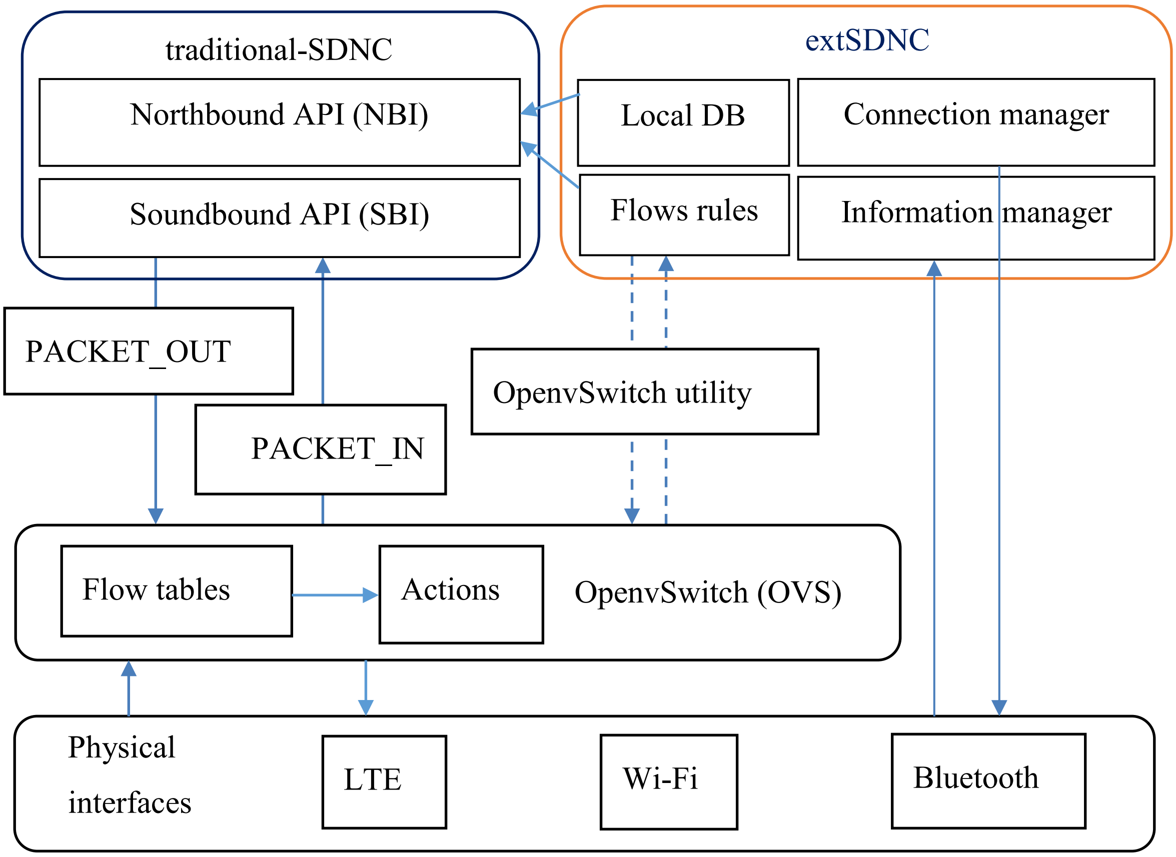


Fig. 2.1. SDN-controllers embedded in a mobile device

Figure 2.1 shows the control logic of a mobile device applied the SDN architecture. All physical interfaces of the device are configured to be ports of a software Open Flow switch named OpenvSwitch (OVS). The OVS navigates network traffic based on OpenFlow rules stored in the Flow tables. When a packet reaches the OVS, its header information is matched with the OpenFlow rules. If the switch finds that there is no available flow rule for the request, the OVS will ask the traditional-SDNC using OpenFlow protocol. For example, it sends a PACKET_IN message to the traditional-SDNC and waits for the answer. The traditional-SDNC will reply with a PACKET_OUT message to instruct the OVS to install or modify rules in the Flow tables. Based on the installed rules, corresponding actions are applied to all packets, which have the same header information.

To support the traditional-SDNC to handle the switch requests as well as local network behavior, the extSDNC is designed with four components named “Local DB”, “Flow rules”, “Connection manager” and “Information manager”. The “Information manager” component is to collect network state and stores them in a local database called “Local DB”. For flexibility in the deployment, an XML format is picked as a storage. After that, the traditional-SDNC, which have no communication channel except the one with a switch, can read the network information at any time. If the collected information is big enough, the traditional-SDNC might have a sufficient knowledge of the network topology. It is possible because the extSDNC can exchange and update this database to any devices to which it connects. Table 2.1 shows the main content of the file, in which, all network configurations are described. For example, the tag Connection is to present the node name and the connection status from the local one to the other nodes. It has two attributes named Name and Status which are marked with X in Table 2.1. The second tag, Media, shows the method which the local node is using to connect to the outside. It has three attributes Name, ID and Key. The attribute Name shows the name of the connection, i.e., Wi-Fi or Bluetooth, while ID shows the SSID of the AP or the MAC address of the Bluetooth master and Key contains security string for authentication. On the third row, the tag Bridge describes information of the bridge which is a virtual interface and it has the same number of the attribute with the Interface tag which presents physical interfaces. The MAC and IP address for virtual and physical interfaces are recorded, however, these values are defaulted ones, they can be changed when the node talks with another node. The invited node normally needs to change its IP address to join the talk. The status of the interface can also be changed to ON or OFF depending on the interface which the node actives and establishes a connection.

TABLE 2.1: NETWORK STATE DATABASE

Tag name	Attributes						
	Name	Switch port	IP	MAC	Status	ID	Key
Connection	X				X		
Media	X					X	X
Bridge	X	X	X	X	X		
Interface	X	X	X	X	X		

Based on the collected information, the “Flows rules” component of the extSDNC prepares OpenFlow rules to actively ask the traditional-SDNC to instruct the OVS with new rules. The “Flows rules” component can play a role of a temporary controller when the traditional-SDNC has problems in controlling the switch. For instance, when the traditional-SDNC is frozen or being killed, the Open vSwitch [66] will have no controller to ask. In this situation, the extSDNC needs to be in charge of controlling the switch at least until the traditional-SDNC comes back. The “Flows rules” component can answer the switch with the prepared rules by utilizing a built-in utility of the Open vSwitch called ovs-vsctl.

The “connection manager” component is developed to be able to control any wireless interface. For instance, it can enable or disable an interface. It can also associate an interface with its network without user interaction. Thereby, it brings a simpler and easier way to maintain an ongoing session between two devices when the current physical connection is going to be lost. In theory, to maintain an ongoing session, the same local end-point IP and MAC addresses must be kept through the session. The applications here only see the middle interface created by the OVS. As a result, if the IP and MAC addresses of the middle interface are not changed, the change in the physical interface does not affect the upper applications. Note that wireless base stations generally only allow packets with the source MAC address of WNIC that has completed the initial handshake [49]. Consequently, on establishing a new connection, the destination sends out an ARP packet asking for MAC address of the new IP address. The extSDNC will automatically send the replies to these requests, however, only selective ARP requests are carefully answered to avoid security risks.

2.3 Evaluation

In this section, the performance of the proposed system is evaluated using a real Testbed. In the testbed, several Linux computers were connected via Wi-Fi or Bluetooth. Each computer was equipped with Core 2 Duo @2.26 GHz and 2GB RAM.

2.3.1 Handover between Wi-Fi and Wi-Fi interfaces

The goal of this experiment is to show the feasibility of the extended SDN controller in a handover in wireless networks and to provide sample flow entry configurations under different handover scenarios.

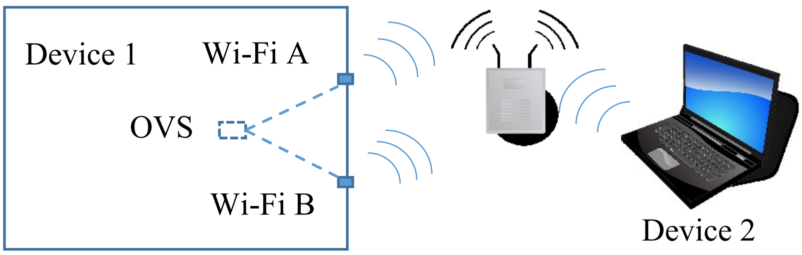


Fig. 2.2. Handover between Wi-Fi and Wi-Fi interfaces Testbed

In this experiment, device 1, which is a mobile device equipped with SDN-controllers, is talking with a conventional mobile device via an AP as depicted in Fig. 2.2. The device 1 is equipped with two wireless interfaces in which one interface is active and the other is for backup purpose. Note that when a device has more than two Wi-Fi interface, any packets sent from the node are easily be looped among the interfaces, hence, they must be controlled by the OVS and the default network-manager service must be disabled. Switch port numbers port_wifi_A and port_wifi_B are assigned to the two interfaces, respectively while the switch port number of the OVS is “LOCAL”. The MAC address of the OVS is borrowed from one interface. It is assumed that the OVS and the Wi-Fi B interface have the same MAC address. On the other hand, the extSDNC collects this information and fills into the local database. It also gathers the information about the device 2 to which the SDN-based node connects. Based on the collected information, OpenFlow rules are described as shown in Table 2.2. When the active interface is Wi-Fi B, flow rules are simple as the OVS simply forwards the packet to corresponding ports. However, when the active interface is Wi-Fi A, the MAC address of the bridge and the active interface are different, thus, the IP and MAC address in the packet header must be rewritten. The mod-dl-dst and mod-nw-dst fields in the flow actions are used to rewrite the destination MAC and IP address in the packet header. For example, when device 1 sends a packet to device 2, the packet will go through the OVS and be encapsulated with the IP and MAC addresses of the OVS. To detect ICMP packets sent from device 1, the OVS checks incoming packets to see if the packet type is ICMP, the input port in_put is “local” and the source MAC address dl-src is equal to the bridge MAC address bridge-mac. The switch then changes the source and destination MAC and IP addresses to the values of two physical interfaces as if the packets were exchanged between them. This is because a wireless client must associate with the AP before exchanging data. The association time sometimes lasts several seconds, thus, the experiment must be long enough to let the handover event occur during transferring data.

In the Testbed, device 1 sent 4000 ICMP packets to device 2 at the rate of 1ms and handovers took place randomly during the experiment. When a handover took place, the extSDNC activated the backup interface and installed new flow entries into the flow table. After the current interface was shut down, the flow of ICMP began running over the new interface. The number of packet loss was also observed in the inverted direction. Each experiment was repeated 30 times and the number of packet loss was shown in Figure 2.3. The figure shows that the highest packet loss rate was 0.3% when switching the ICMP flow from the interface B to the interface A. In an inverted way, there was less packet loss and the number of packet loss was more stable. This is because the interface B has the same MAC address with the bridge, hence, flow rules were simple and the system spent less time to switch the flow.

TABLE 2.2: FLOW RULES

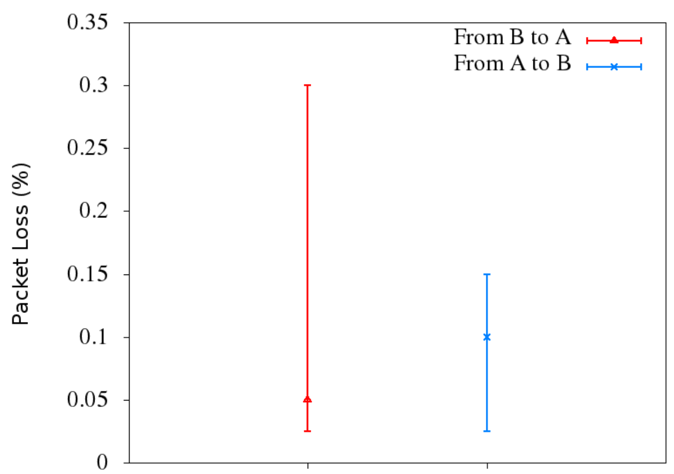


Fig. 2.3. Packet Loss in Handover between Wi-Fi and Wi-Fi interfaces

Direction	*Flow rules*	*Flow actions*
From interface A to B	in_port=local in_port= port_wifi_B	Output: port_wifi_B Output: local
From interface B to A	in_port = local, dl-src = bridge-mac, type = icmp in_port = port_wifi_A, dl-dst = wifi-A-mac, type = icmp	mod-dl-dst=wifi-dst-mac, mod-nw-dst=wifi-dst-IP, mod-nw-src=wifi-A-IP, mod-dl-src=wifi-A-mac, output: port_wifi_A mod-dl-dst=bridge-mac, mod-nw-dst= bridge -IP, output: LOCAL

2.3.2 Handover between Wi-Fi and Bluetooth interfaces

The goal of this section is to evaluate the performance of the system in the vertical handover between Wi-Fi and Bluetooth interfaces. Since two physical interfaces use different ways to connect, it is assumed that two devices are able to coordinate to switch traffic between them.

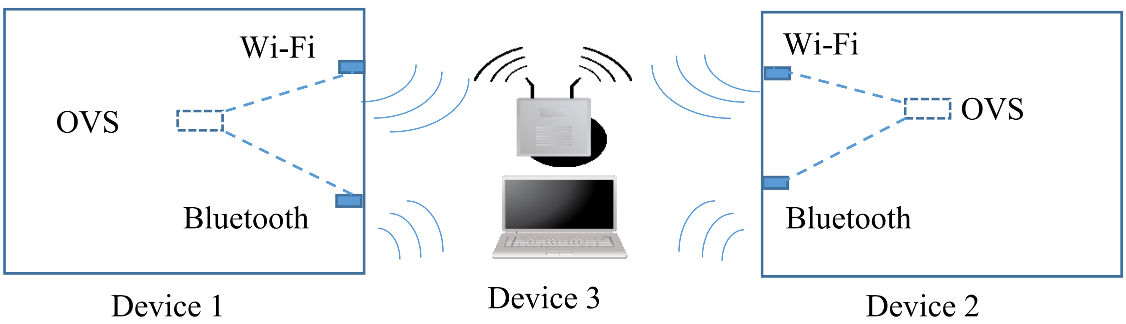


Fig. 2.4: Handover between Wi-Fi and Bluetooth interfaces Testbed

Figure 2.4 shows the scenario for the vertical handover experiment. In the figure, two devices equipped with one Wi-Fi and one Bluetooth interface are communicating. It is assumed that the nodes can be either member in the same Bluetooth piconet formed by another conventional mobile node or two wireless clients which connect to the same AP. All physical interfaces are controlled by the OVS and switch port numbers are assigned to them. The OVS chooses the

MAC address for the bridge, assumed that it takes the value of the Bluetooth interface. This information is collected and recorded in the local database which has the same structure as the one in the previous experiment. The system also uses the same way to define flow rules to rewrite IP and MAC address in the packet header. Note that the Wi-Fi interface has a higher speed than the Bluetooth one, hence, it is difficult to separate the number of packet loss because of reducing the connection bandwidth and the packet loss during handover. Consequently, in this experiment, the handover only happens in the direction from the Bluetooth interface to Wi-Fi interface. When the Bluetooth connection is going to be lost, i.e., two nodes are out of the Bluetooth communication range, the ongoing traffic is automatically shifted to the Wi-Fi connection.

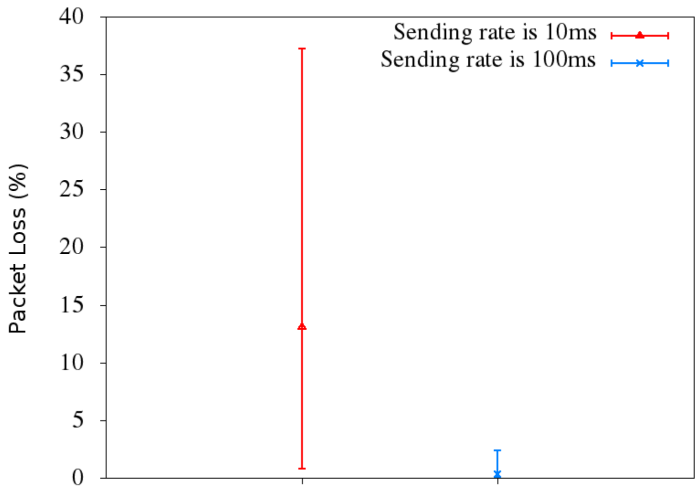


Fig. 2.5. Packet Loss Counts in Handover between

In the first experiment, 1000 ICMP packets were exchanged between two nodes at the rate of 10ms and handovers took place randomly during the experiment. The experiment was repeated 30 times with random handover and the number of packet loss are presented in Fig. 2.5. Figure 2.5 shows that the packet loss is up to 37.2%, or the maximum number of packet loss is 372 packets over 1000 sent ones. This rate is reduced when packets are sent with a slower speed. The highest rate of packet loss now is only 2.4% when the packet interval is 100ms. The results confirmed that the extSDNC enhanced the system resilience since it can automatically switch the traffic.

2.3.3 Handover when TCP and UDP traffic are transferred

This section is to validate the proposed system with TCP and UDP protocol, which are basic protocols for almost all network applications, in a real Testbed.

2.3.3.1 Experiment settings

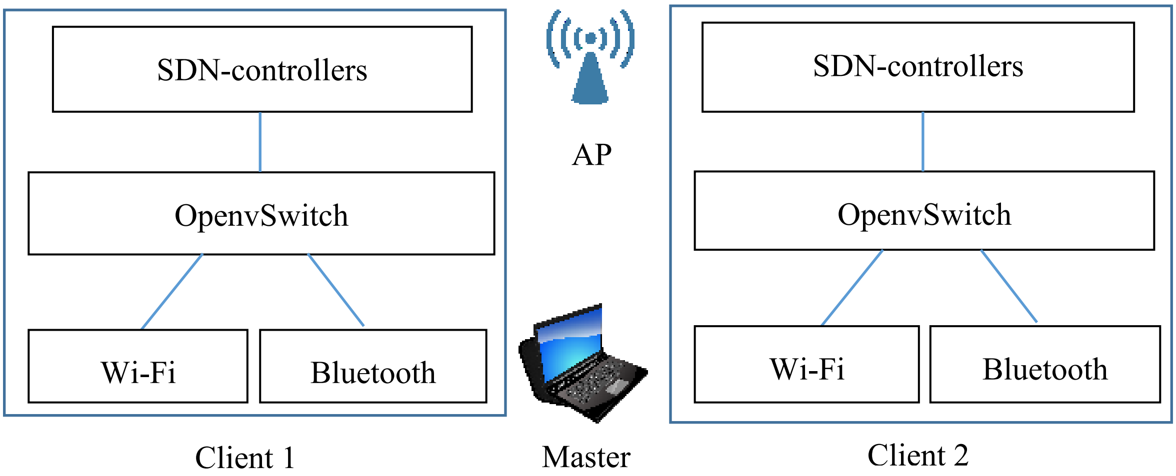


Fig. 2.6: Experiment topology

The system has been evaluated using the network topology as illustrated in Figure 2.6. In the figure, two SDN-based mobile nodes equipped with one Wi-Fi and one Bluetooth interface are communicating in a Bluetooth piconet. For the Bluetooth connection, we disabled the built-in Bluetooth interface, as it only supports Bluetooth version 1.2. An external class 2 Bluetooth dongle is added to every node in order to test the system with the latest version of Bluetooth standard – Bluetooth v4.0. Regarding the Wi-Fi connection, any nodes already have a built-in Wi-Fi interface which is able to connect to 802.11n Wi-Fi networks. These physical interfaces on each node are under control of the Open vSwitch which is managed by the original SDN controller. As also shown in Fig. 2.6, two SDN-based nodes are communicating in a piconet via a master node. The master node in the piconet can be either an SDN-based mobile node or a conventional mobility device such as a tablet, a mobile phone or a laptop. Note that the master node provides IPs in a different subnet than the Wi-Fi interface. This is feasible as the packets, which are passed to an interface, are navigated by the flow rules in the flow table on the switch.

Besides, the Bluetooth and Wi-Fi connectivity parameters such as Bluetooth MAC address, Bluetooth authentication key, Wi-Fi SSID and Wi-Fi authentication key, IP address of all physical interfaces are assumed to be already in the local database. Therefore, all association processes are automated. Based on the information in the database, OpenFlow rules are also defined.

2.3.3.2 TCP traffic load

The goal of this experiment is to estimate the maximum execution time for handover detection procedure so that the system can perform a soft handover between Bluetooth and Wi-Fi while TCP-based applications are running. To this end, a vertical handover was triggered during an FTP file exchange session. The session duration was limited to less than 10 seconds as a person with a speed of 1m/s can go out of a Bluetooth communication coverage (10m) within that period of time. Therefore the file size was selected to be small enough, i.e., 900 KB, so that the file transfer time over the Bluetooth connection is less than 10 seconds.

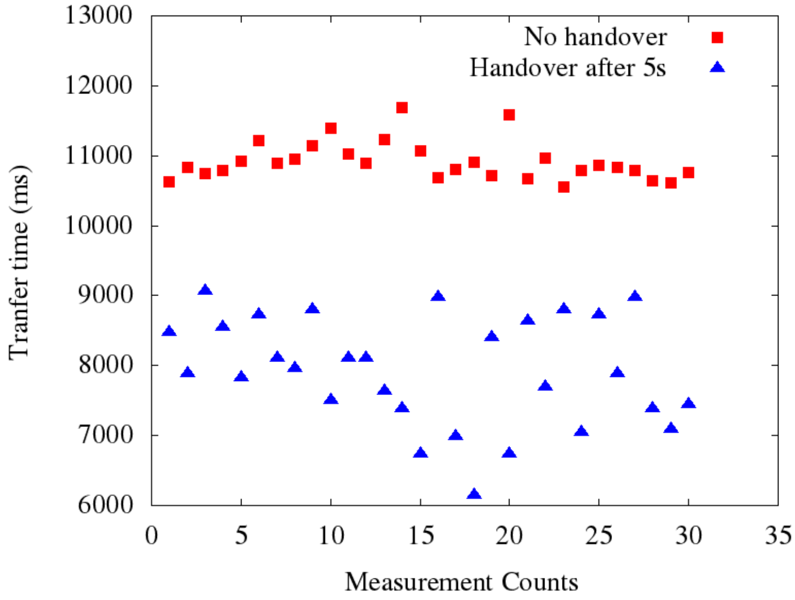


Fig. 2.7. FTP file transfer time

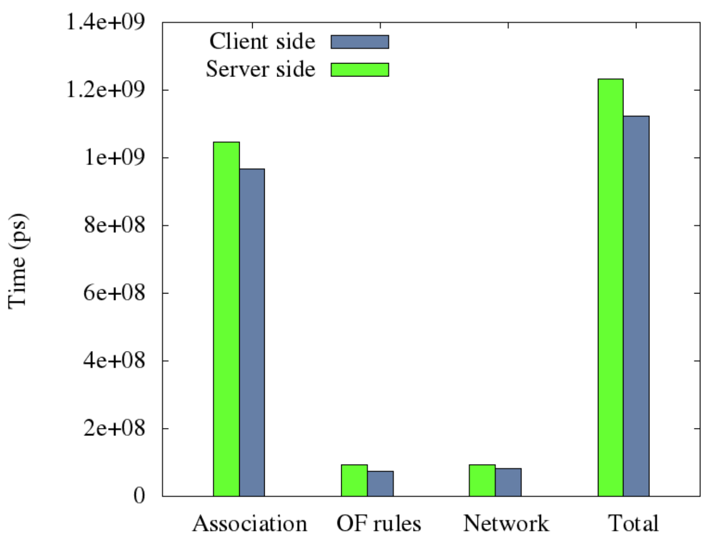


Fig. 2.8. Average execution time

The measurement procedure with TCP traffic load is as follows: Initially, node 1 downloads a file from node 2 in an FTP client-server manner (i.e., using WGET program to download the file through FTP protocol) via Bluetooth connection. After 5 seconds, a vertical handover procedure is invoked on each node. The point of time that the procedures are called can be slightly different as the handover occurs only when the synchronization procedure verified that both nodes have already finished the association process. The experiment was repeated 30 times and all the transferred files were not corrupted. Note that the WGET program basically shows transfer times in seconds, thus, the "time" command is used to call the WGET program in order to allow a measurement with millisecond accuracy. The file transfer durations with the vertical handover are also compared with that duration when there is no handover as shown in Fig. 2.7. Figure 2.8 shows that the maximum transfer duration when a vertical handover occurs is 9.064 seconds while the minimum duration when there is no handover is 10.559 seconds. Note that the handover was triggered 5 seconds after the receiver began to download the file. This means if the handover detection lasts less than 6.5 seconds, the vertical handover still occurs in a soft handover manner or make-before-break.

For more detail of the vertical handover execution time, the time for Wi-Fi association, OpenFlow rules installation and Network configuration were measured separately as given in Fig. 2.8. Note that the handover procedure is written in Python [50] so it is not difficult to get the time of execution of a block of code by using existing Python's modules, i.e., "time" module can provide various functions for manipulating clock time. The figure shows that the average time of the whole execution process is 1.122 seconds at the client side and 1.232 seconds at the server side. However, the average time of the handover delays, which affect the running applications, including the time for installing OpenFlow rules and configuring the network at the client and server is 0.156 seconds and 0.184 seconds respectively. These values can be reduced by running processes in parallel. The difference between client and server is caused by the synchronization as the starting points of the handover procedure on two nodes are different.

The experiment results have confirmed the feasibility of the SDN-based vertical handover system with a TCP traffic load. Moreover, it shows that the performance of the system can be measured in detail with a picosecond deviation.

2.3.3.3 UDP traffic load

Although there are some packet losses in a TCP transmission, the protocol retransmits all of them. Therefore, to measure network performance parameters such as the number of packet loss and jitter, the system was evaluated with UDP traffic load. The experiment time was also set to 10 seconds.

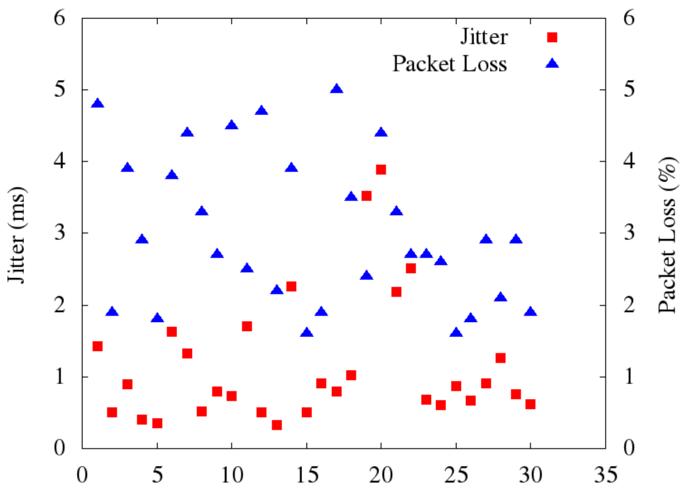


Fig. 2.9. UDP packet loss and jitter

The measurement was conducted by using the Iperf tool. Firstly, an Iperf server is started on node 2 listening for incoming requests. Then, an Iperf client, which is enabled on node 1, attempts to send UDP traffic to the Iperf server. On both nodes, a vertical handover is triggered randomly while UDP traffic is being sent. The experiment was repeated 30 times and the measured results are shown in Fig. 2.9. The figure shows that the highest packet loss rate was 5% with 0.8ms jitter delay. Although the packet loss rate varies from 1.6% to 5%, the jitter is mostly less than 1ms. The results have confirmed the feasibility of the system with UDP traffic load. Furthermore, the evaluation suggests that the SDN-based system is a promising solution to maintain ongoing session of UDP-based applications during such a vertical handover.

2.4 Conclusion

The goal of this work is to allow mobile devices to perform VHO without any external support and extSDNC has been proposed. In the system, the SDN-controllers were embedded in each mobile device to collect network state and gave the traditional-SDNC the knowledge about the network topology by storing information in a local database. The performance of the system was evaluated under different scenarios. The evaluation results confirmed that communicating devices that equipped extSDNC can maintain an ICMP flows when changing wireless technologies. However, the mechanism that allows two communicating devices to change the technology at the same time has not been discussed. The next chapter will address this issue.

Chapter 3 Communication between extSDNCs

In the previous chapter, the proposed extSDNC can support mobile devices to perform VHO based on their own recorded information, i.e., local measurement executed by themselves. However, it cannot guarantee that the VHO processes on both communicating devices finish at the same time. When they complete the VHO process at the different point of time, their connection will be interrupted. If the connection is interrupted in a long time, a lot of packets will be lost, hence, the quality of ongoing services cannot be guaranteed. To address this issue, a mobile-controlled VHO management is proposed to synchronize the VHO process on each device, minimizing the number packet loss.

3.1 Assumed network environment

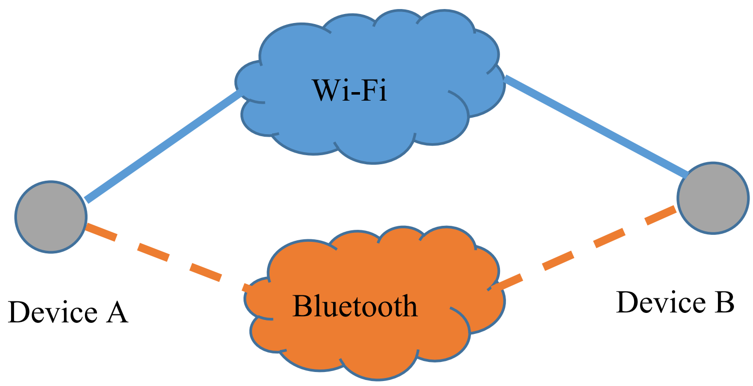


Fig. 3.1. Considered network topology

The considered scenario in this work is shown in Fig. 3.1. In the figure, two mobile devices, which are equipped with extSDNC described in the previous chapter, are communicating with each other. On each device, Wi-Fi and Bluetooth are available and the interference between them is assumed to be minimized. Basically, the devices use Wi-Fi technology to connect to the other via an access point (AP) or in an ad-hoc manner. Bluetooth technology is used as a backup one. It is also assumed that they are al-ways able to join a Bluetooth network. However, the Wi-Fi network and Bluetooth network are not interconnected.

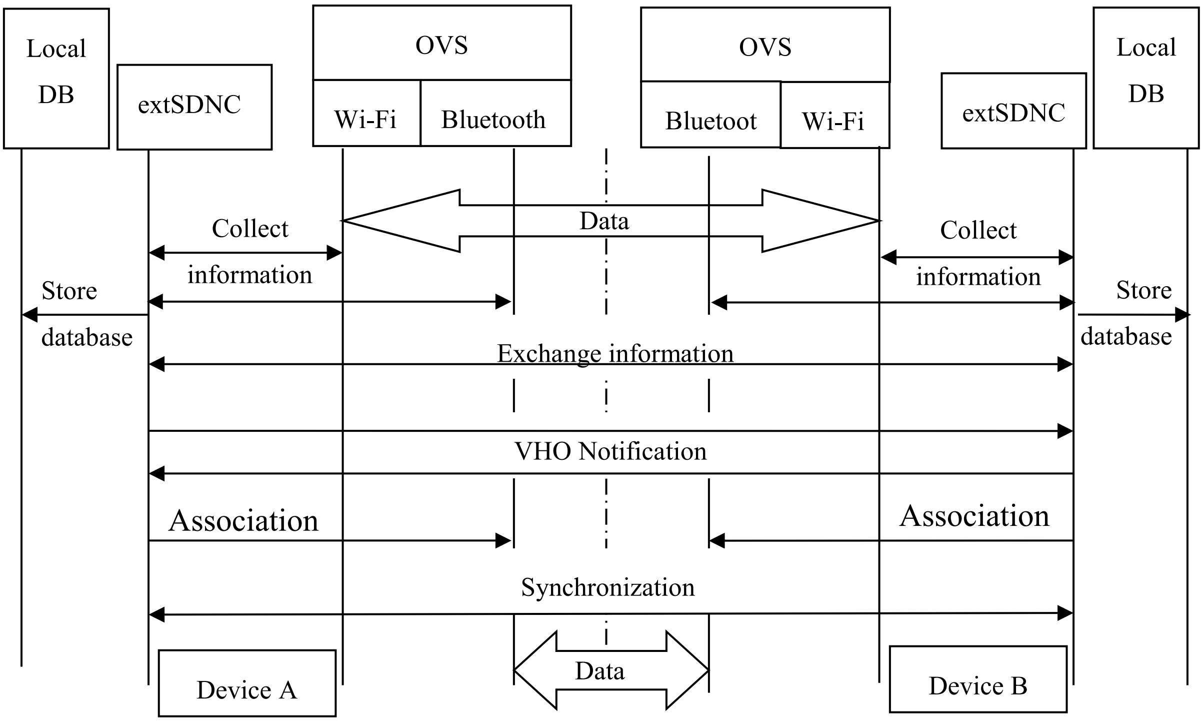


Fig. 3.2. Messages exchanged in a mobile-controlled VHO management

3.2 Mobile-controlled vertical handover management

The mobile-controlled VHO management process here follows a three-phase procedure as illustrated in Fig. 3.2. As can be seen from the figure, the messages are exchanged only between two communicating devices. There is no external element involves in the control of the VHO process. The information for the VHO is collected and exchanged when two devices start communicating. When the devices decide to perform VHO, a simple message exchange procedure is executed, aiming to shorten the VHO delay with the cost of a few packet loss.

The detail of the message sequence is as follow:

In the system discovery phase, the extSDNC, described in the previous chapter, on the mobile devices collects all the necessary information to make a decision whether or not to trigger VHO from the current access technology to the other. The collected information includes the local information of the device such as battery level, MAC and IP addresses as well as the information from the network side like access point properties, throughput, round-trip time (RTT). The information collected by each device is stored in a local database and is exchanged with other devices. If there is no updated information, they exchange the information only one time right after their communication starts. Both traditional-SDNC and extSDNC can retrieve information from the database. However, only the extSDNC can add or modify the data in the database. Based on the collected information, the extSDNC makes a decision whether or not to trigger a VHO in the second phase. In this chapter, the VHO decision is made simply based on the total throughput B_{total}. If the B_{total} is smaller than a threshold B_{Thres}, the extSDNC will decide to perform a VHO from Wi-Fi to Bluetooth. Otherwise, it connects two devices via a Wi-Fi connection. When the VHO decision is made, the extSDNC notifies the extSDNC on the other device of the decision and which technology is going to be used. Once both sides receive the VHO notification, the VHO procedure starts. The extSDNC activates and associates the backup wireless interface with its network. When the association process has finished, two devices must synchronize with the other in a similar way to the TCP 3-way handshake (SYN, SYN-ACK, and ACK). The synchronization process is to let the new wireless technology on two devices be available at almost the same time, hence, the packet loss will be reduced. After that, the extSDNC can tell the traditional-SDNC to instruct the switch to navigate the traffic to the new interface.

3.3 Energy consumption model

Let N_B and N_W denote the number of packets traveled over Bluetooth and Wi-Fi, respectively. Also let R_B and R_W denote the average data rate in the network of Bluetooth and Wi-Fi, respectively. The duration times used for Bluetooth and Wi-Fi are T_B and T_W, respectively, and the system takes T seconds to send the total N packets as illustrated in Fig. 3.3. The energy consumption of the wireless communication E is the total energy consumption when each wireless technology is being used and is expressed by equation (1):

$$E = P_W T_W + P_B T_B \quad (1)$$

The duration time T_B and T_W can be calculated by the following equations if there is no packet loss:

$$\begin{cases} N_B + N_W = N & (2) \\ N_B/R_B + N_W/R_W = T & (3) \end{cases}$$

From equation (2) and (3), we can calculate T_W and T_B as follows:

$$T_W = N_W / R_W = (R_B R_W T - R_W N) / (R_B - R_W) R_W \qquad (4)$$

$$T_B = N_B / R_B = (R_B R_W T - R_B N) / (R_W - R_B) R_B \qquad (5)$$

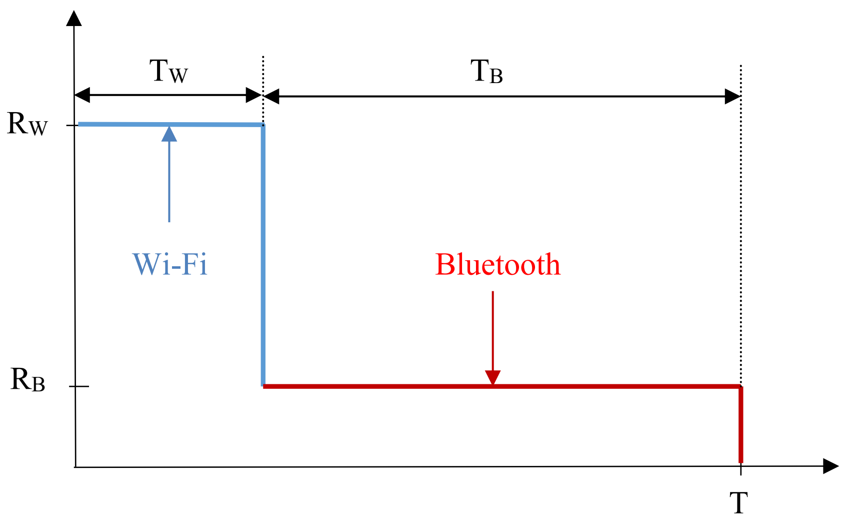


Fig. 3.3: Energy consumption model for the mobile-controlled VHO

The energy consumption of the wireless communication is minimal if the device uses Wi-Fi in the shortest duration. The shortest time T_{Wmin} is achieved when the VHO process from Wi-Fi to Bluetooth and the service start at the same time. After that, only Bluetooth is used (Fig. 3.3). The T_{Wmin} is also the total execution time needed to perform a VHO from Wi-Fi to Bluetooth called $T_{VHO\text{-}WB}$. The minimal energy consumption E_{min} when using the mobile-controlled VHO management is calculated as follow:

$$E_{min} = P_W T_{Wmin} + P_B T_B = P_W T_{VHO\text{-}WB} + P_B T_B \quad (6)$$

Although the device can successfully perform the VHO from Wi-Fi to Bluetooth to save energy, it has to switch back to Wi-Fi in some cases. For example, if the required data rate of the communication is higher than Bluetooth capability, i.e., 1Mbps, the device needs to switch the connection back to Wi-Fi to ensure the quality of running services. In the worst case, the device has to perform another VHO from Bluetooth to Wi-Fi right after finishing the handover from Wi-Fi to Bluetooth. In this case, the time period the device has been using Bluetooth corresponds to the total execution time to perform the VHO from Bluetooth to Wi-Fi ($T_{VHO\text{-}BW}$). The maximal energy consumption *Emax* when using the mobile-controlled VHO management is calculated as follow:

$$E_{max} = P_W T_W + P_B T_{Bmin} = P_W T_W + P_B T_{VHO\text{-}BW} \qquad (7)$$

Since it takes $T_{VHO\text{-}WB}$ to perform a handover from Wi-Fi to Bluetooth and $T_{VHO\text{-}BW}$ to perform a handover in the inverted direction, the number of successful VHO in the duration T is the quotient in the division given in the equation (8):

$$N_{VHO} = T \,/\, (T_{VHO\text{-}WB} + T_{VHO\text{-}BW}) \qquad (8)$$

The minimal energy consumption when there is N_{VHO} VHO occurred is:

$$E_N = P_W T_W + P_B N_{VHO} T_{VHO\text{-}BW} \qquad (9)$$

3.4 Evaluation

3.4.1 Mobile-controlled VHO management performance evaluation

This evaluation is to show the effectiveness of the mobile-controlled management when two mobile devices equipped extSDNC, at the same time, switch network traffic from Bluetooth to Wi-Fi and vice versa. The performance metrics are the packet loss rate and the handover delay.

3.4.1.1 UDP traffic load

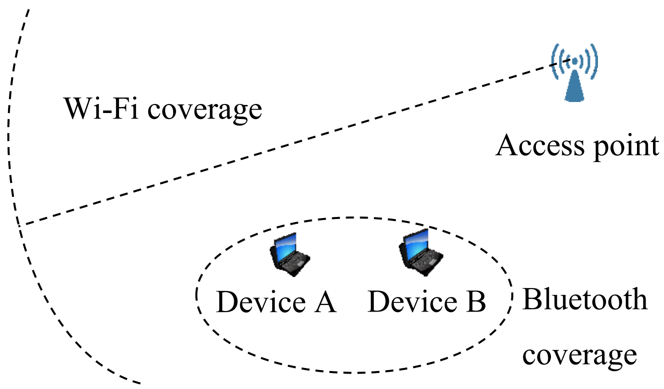


Fig. 3.4. Experiment topology

To observe the handover delay and the packet loss rate, UDP is selected as the transport protocol because it provides a unidirectional communication, thus, lost packets are not retransmitted.

The testbed in this evaluation is illustrated in Fig. 3.4. The figure shows that two mobile devices are placed within the communication coverage of a Wi-Fi and Bluetooth networks. In the evaluation, Iperf [51] is used to generate a UDP flow toward the direction from device A to device B.

As mentioned in the previous chapter, the OVS is used to provide a virtual interface between an application and physical interfaces. Since the application only sees the virtual interface, it does not notice the change of physical interfaces. On the other hand, physical interfaces play the role of switch ports. The OVS navigates traffic that comes in and goes out of the interfaces by OpenFlow rules, which are created by the SDN controller and are stored in OVS's flow table. The time taken to install OpenFlow rules is named $T_{OFrules}$. Note that the physical interface can be used after it is activated and associated with its network. This process takes $T_{NWconfig}$ seconds. The difference between the end time of configuring network and that of installing OpenFlow rules is defined as handover delay. The delay is the source of packet loss. For instance, if OVS finishes installing OpenFlow rules before the physical interface is configured, the traffic following the OpenFlow rules is sent to a physical interface, which has not been ready yet. Alternatively, if the new interface is ready, however, the OpenFlow rules is not installed, the traffic is not sent to it.

The measurement procedure is as follows: Firstly, device A with the role of a server starts Iperf in server mode and waits for incoming requests. On device B, an Iperf client then attempts to establish a connection with the Iperf server. While a 100 Kbps UDP traffic is being sent, a VHO occurs randomly. The delays were captured at the same time with measuring the packet loss rate in the previous experiments. The procedure was repeated 50 times and the measured results are shown in Fig. 3.5 and Fig. 3.6. The figure 3.5 shows that the network configuration duration T_{BNW} and the OpenFlow installation duration T_{BOF} of Bluetooth are 119.45ms and

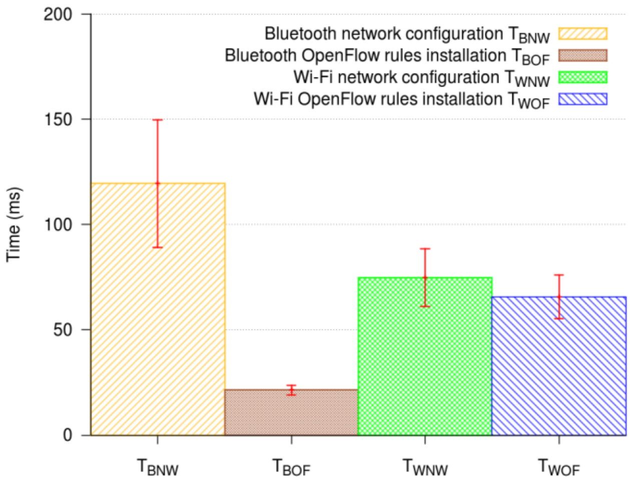


Fig. 3.5. Handover delay measurement

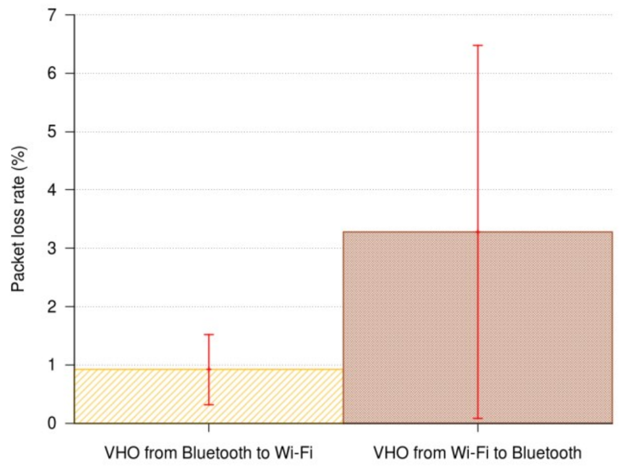


Fig. 3.6. Packet loss rate when handover between Bluetooth and Wi-Fi

21.34ms on average, respectively. The handover delay is as long as 98.11ms, causing 3.5% packets loss on average as shown in Fig. 3.6. The packet loss rate is relatively high, however, mainly below 7%, even if the traffic was switched from high data rate network to a lower one. In the inverted direction, a delay was inserted before executing install OpenFlow rules, reducing the handover delay to 18ms on average. As a result, the packet loss is only 1% as given in Fig. 3.6. The obtained results confirm that the shorter delay is, the fewer packets are lost. Moreover, the handover delay can be reduced by reducing the difference between T_{WOF} and T_{WNW}. The handover delay is as small as 18ms with 1% packet loss rate can support real-time services.

3.4.1.2 TCP traffic load

The goal of this experiment is to verify the feasibility of the proposed system in maintaining the continuity of the service. In addition, the obtained results will be used to calculate the total execution time to perform VHO from Wi-Fi to Bluetooth and vice versa. To this end, a vertical handover during an FTP file exchange session has been observed. The average date rate of Bluetooth and Wi-Fi are also measured when there is no handover.

The measurement procedure with TCP traffic load is as follows: Initially, a program named WGET on device A tries to download a file from device B, on which an FTP server is running, via Wi-Fi connection. When the program begins, a vertical handover procedure is also invoked on both devices. The total execution time has been captured. The experiment was repeated 30 times and all the transferred files were not corrupted.

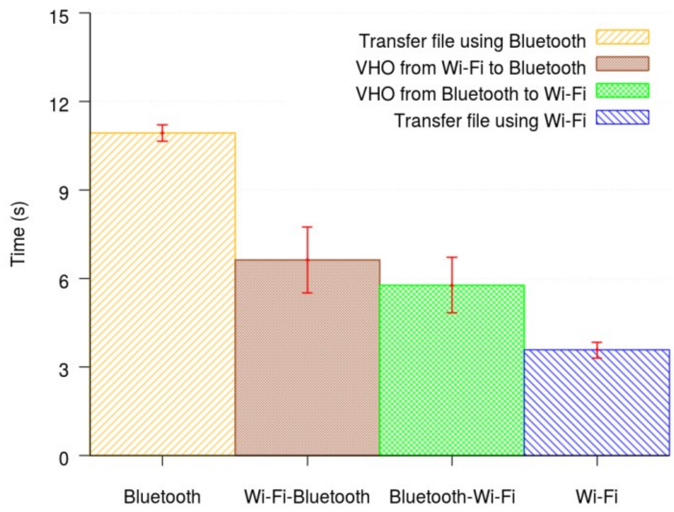


Fig. 3.7. File transfer time

The file transfer durations with the vertical handover are also compared with that duration when there is no handover as shown in Fig. 3.7. Figure 3.7 shows that it takes 10.93±0.28 seconds to transfer an 894KB file using Bluetooth. The data rate of the Bluetooth connection is 83.88±2.07 KBps. If Wi-Fi is used, that duration is only 3.57±0.27 seconds. The data rate of the Wi-Fi connection is 257.87±18.89KBps. When a VHO from Bluetooth to Wi-Fi occurs, the file transfer time is 5.77±0.94 seconds. In the inverted direction VHO, the file transfer time is 6.62±1.12 seconds. From the collected values, the average value of T_{Wmin} and T_{Bmin} can be calculated using equation (3) and (4), respectively:

$$T_{Wmin} = (R_B R_W T - R_W N) / (R_B - R_W)\, R_W$$

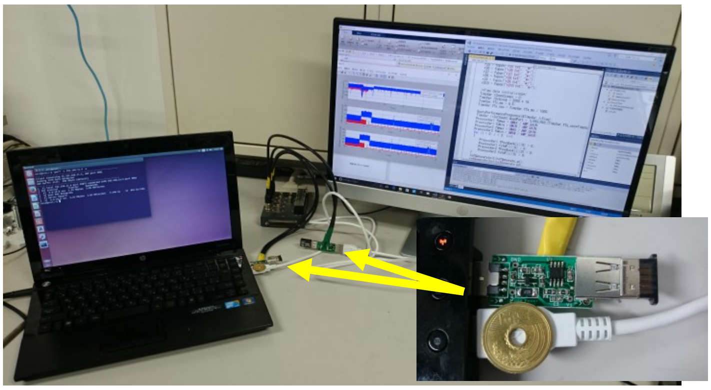
$$= (83.88*257.87*6.62 – 257.87*894) / (83.88 – 257.87)257.87 = 1.95s$$

$$T_{Bmin} = (R_B R_W T - R_B N) / (R_W - R_B)\, R_B$$

$$= (83.88*257.87*5.77 – 83.88*894) / (257.87 – 83.88)\ 83.88 = 3.41s$$

3.4.2 Energy efficiency evaluation

3.4.2.1 Experiment setup



Current monitoring circuit

Fig. 3.8: Energy consumption measurement setup

The energy consumption of wireless communication with and without a VHO has been measured by using a real testbed illustrated in Fig. 3.8. On the left of the figure, the mobile device is a Linux computer (Core 2 Duo @2.26 GHz processor and 2GB RAM, Ubuntu 14.04 64-bit). The device connects to Wi-Fi and Bluetooth networks using USB-based Wi-Fi and Bluetooth adapters, respectively. The adapters are connected to a current monitoring circuit before being attached to the mobile device. The output signal of the monitoring circuit is converted to a digital signal via an analogue-to-digital converter (ADC). The digital signal is then sent to the signal analysis device. The signal analysis device is a computer in which a small program is running to capture and save the digital signals.

3.4.2.2 Energy consumption measurement

In this evaluation, each device only uses one wireless technology at almost any given time, the other technologies are turned off. A technology is called “off” when it is not attached to any network. Therefore, the technology still consumes energy even it is turned off. The instantaneous power consumed in “off” state of Wi-Fi and Bluetooth has been measured and the results are given in Fig. 3.9. As shown in the figure, the average power of Wi-Fi $P_{Wi\text{-}Fi\text{-}off}$ and Bluetooth $P_{Bluetooth\text{-}off}$ in the “off” state are 400.75mW and 37.25mW, respectively.

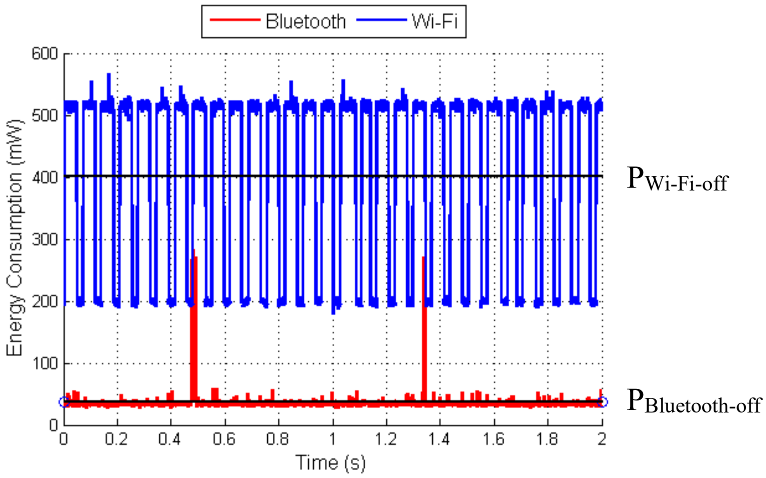


Fig. 3.9. Instantaneous power when wireless interfaces are off

To examine the power consumption by each technology, device A and device B exchange file using FTP protocol. When two devices use Wi-Fi, Bluetooth is turned off and vice versa. The measured results are shown in Fig. 3.10 and Fig. 3.11. Fig. 3.10 shows the instantaneous power P_B when the device uses Bluetooth. As shown in the figure, Bluetooth only consumes 82.24mW on average. The total average power of the wireless communication when using Bluetooth is:

$$P_{Wireless\text{-}B} = P_B + P_{Wi\text{-}Fi\text{-}off} = 82.24 + 400.75 = 482.99\text{mW}$$

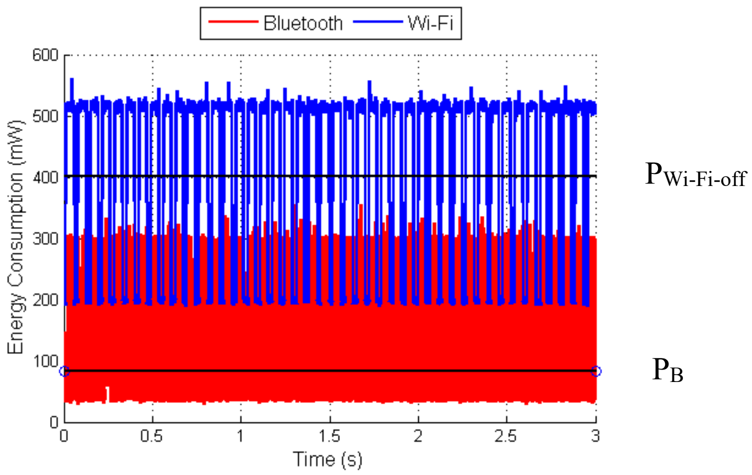


Fig. 3.10. Instantaneous power when using Bluetooth

Similarly, Fig. 3.11 shows the instantaneous power when the device uses Wi-Fi P_W. As shown in the figure, Wi-Fi consumes 605.05mW on average. The total average power of the wireless communication when using Wi-Fi is:

$$P_{Wireless\text{-}W} = P_W + P_{Bluetooth\text{-}off} = 37.25 + 605.05 = 642.30\text{mW}$$

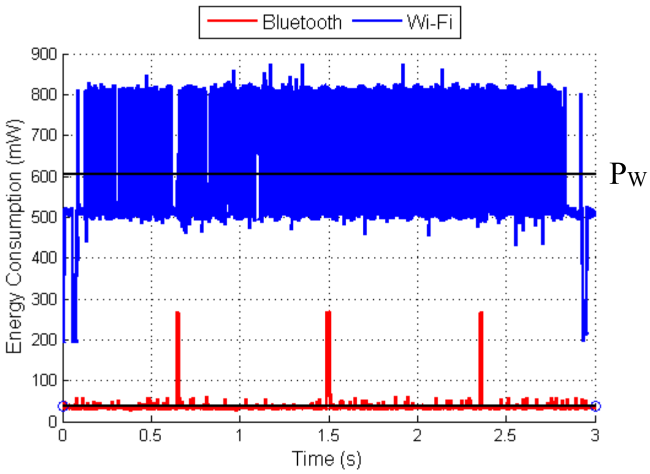


Fig. 3.11: Instantaneous power when using Wi-Fi

3.4.2.3 Energy saving estimation

Assume that a service S last T=15s, the average power of the wireless communication when using Wi-Fi is:

$$P_W = P_{Wireless\text{-}W}T = 642.30*15 = 9634.5\text{mJ}$$

The average power of the wireless communication when using Bluetooth is:

$$P_W = P_{Wireless\text{-}B}T = 482.99*15 = 7244.85\text{mJ}$$

During the duration T, the maximal and minimal average power by using the mobile-controlled VHO management are calculated by using equation (6) and (7), respectively:

$$P_{min} = P_{Wireless\text{-}W}T_{Wmin} + P_{Wireless\text{-}B}(T - T_B)$$

$$= 642.30*1.95 + 482.99\ (15\text{-}1.95) = 7555.50\text{mJ}$$

$$Pmax = P_{Wireless\text{-}B}T_{Bmin} + P_{Wireless\text{-}W}(T - T_{Bmin})$$

$$= 482.99*3.41 + 642.30\ (15\text{-}3.41) = 9091.25mJ$$

The number of VHO can be occurred is the quotient in the division given in the equation (8):

$$N_{VHO} = (T / (T_{Wmin} + T_{Bmin}))$$

$$= (15 / (1.95+3.41)) = 2$$

The minimal power consumed when there are two successful VHO occurred during the duration T is calculated using equation (9):

$$P_{N=2} = P_{Wireless\text{-}W}\ (T - N_{VHO}T_{Bmin}) \ + \ P_{Wireless\text{-}B}N_{VHO}T_{Bmin}$$

$$= 642.30*(15\text{-}2*3.41) + 482.99*2*3.41 = 8548.01mJ$$

From the estimated results, the power consumption by using the proposed mobile-controlled VHO management can save from 5.64% to 21.58% the power consumed by using Wi-Fi. The results also show that, even in the worst case, multiple successful VHO can save energy at least 11.28%.

3.5 Conclusion

In this chapter, an energy-efficient mobile-controlled VHO management has been introduced. The mobile-controlled VHO is an extension of the extSDNC, aiming to support mobile devices to reduce energy consumption as well as to maintain the continuity of all active services. The communication behavior of the system in term of message interchange has been presented and the performance of the system has been evaluated with TCP and UDP traffic load. The obtained results have confirmed that the proposed management system can maintain the continuity of real-time services by offering fast handover with low packet loss rate. The energy consumption of the system has also been measured by direct measurements. Based on the measured results, the minimal and maximal energy saving has been estimated. The results show that the system can save up to 21.58% in comparison to the energy consumed by using Wi-Fi. In the worst case, the system can save only 5.64%, however, performing multiple VHO can save more energy. The task remaining is to design an algorithm to trigger VHO at the right time and it is the topic of the next chapter in this dissertation.

Chapter 4 VHO decision in heterogeneous wireless communication

The chapters show that it is feasible that mobile devices equipped extSDNC can perform a VHO fully by themselves. However, when to trigger the VHO process has not been discussed yet. Therefore, this chapter proposes a VHO trigger algorithm and a device-to-device heterogeneous wireless communications (D2DHWC) network to allow mobile devices to communicate, without a doubt, by using different wireless access technologies, i.e., Bluetooth or Wi-Fi. The considered scenario is a disaster situation in which the essential network infrastructures could be destroyed interrupting the communication among the victims and isolating them with the outside. Fortunately, multiple wireless technologies built in a single mobile device enables the exploration of different network access, thus, it is highly possible to utilize these technologies to form a heterogeneous wireless communication. Such a heterogeneous communication means is useful, especially in disaster scenarios, because having an uncorrupted and timely information exchanging means is critical for affected victims to survive or to connect to the outside world.

4.1 Requirements for the proposed system

4.1.1 Assumptions in a disaster environment

In this work, disaster situations are considered; where public infrastructured networks such as cellular network and WiMAX network cannot work. However, one or more local infrastructured network elements like APs could practicably be alive. Regardless of whether or not these APs can provide the Internet access, they could at least bring local connections to nearby devices. In the case of no alive APs, any mobile node can become a soft-AP [52] to establish communication channels for the others. In addition, victims' devices such as smartphones and tablets can join a Bluetooth network since Bluetooth is available on those devices as a standard function. These tentative networks should effectively be used for local communications.

When evacuating, the victims tend to potentially go together to form a group and to assist others [53, 54, 55 and 56]. In this case, they often need to use a real-time application such as VoIP by their mobile devices as a critical communication means rather than passing messages for rapidly making right decisions [57]. Therefore, the voice stream must be sustained until exiting the affected area. In such a VoIP network, a call server is commonly required. However, the call server is just to process the call setup, and thus it does not switch the voice stream. Once the call server finishes setting up a call, it generally becomes dormant during the conversation. Therefore, the mobile devices should know the other's IP address to have a peer-to-peer call without working with a call server.

4.1.2 Energy-efficient communications in disaster scenarios

The US Department of Homeland Security (DHS) has a program named SAFECOM [58] in which technical requirements for communication services in disasters have been introduced. The key aspects of the requirements include Quality of Service (QoS), energy consumption, robustness, and reliability. Therefore, the proposed communication system needs to meet these requirements.

- Energy efficient communication:

Wi-Fi is designed for high-speed wireless data transmissions. Although the standard has already introduced a power-saving mode (PSM) for saving energy, the power consumption in Wi-Fi is mostly linear with the obtained throughput [59]. Besides, the WLAN interface consumes energy even if it is in the sleep state [36]. In contrast, the devices can operate for years if they use Bluetooth technology [60]. Therefore, turning off Wi-Fi and switching the on-going sessions to Bluetooth are reasonable for an energy-efficient communication. To realize this real-time session switching, a fast vertical handover algorithm is necessary.

- Heterogeneous wireless communication:

The proposed system must be able to maintain all ongoing sessions while changing the access technologies. For instance, if two devices connecting each other via Bluetooth are going to be disconnected due to their distance, the system must be able to switch the ongoing traffic to Wi-Fi with low packet loss rates. The proposed system also should be reliable adapting to the change of network environment including the device mobility, resulting in providing a robust heterogeneous communication. In the heterogeneous communication environment, all the devices are able to connect to the others without nervousness about what technology they are using. The connection in the network must meet the required QoS parameters for real-time services such as VoIP.

4.2 Energy-efficient wireless communications algorithm

This algorithm is for conserving battery life of the device. Besides the basic methods such as dimming the brightness of the display screen and disabling certain applications, turning off or disabling unconnected wireless interfaces will allow the mobile device to use much lower

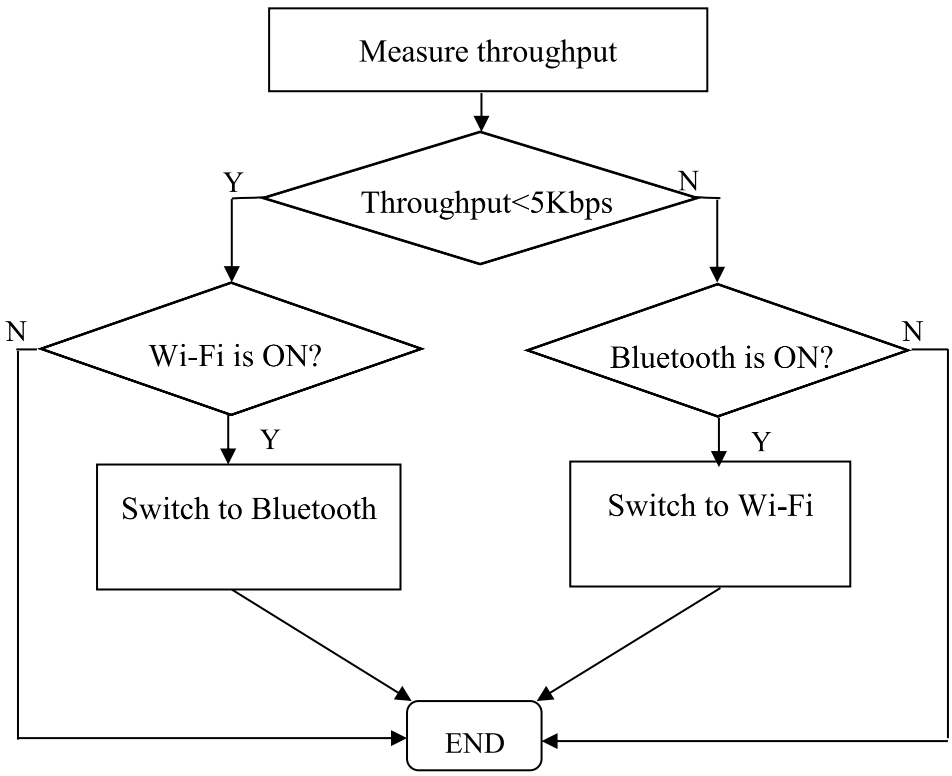


Fig. 4.1. A vertical handover triggering algorithm

power during periods of inactivity. The key reason for focusing on this is that wireless technologies constantly seek for a good connection by scanning, hence, they consume the energy continuously. Note that the feature of switching between the activated wireless interface and the backup one without disturbing the running application has been introduced in Chapter 2. In this Chapter, when and how the handover is triggered is discussed more concretely.

The algorithm for triggering the energy-efficient vertical handover is given in Fig. 4.1. Firstly, the network throughput is measured by utilizing the Python psutil library [67]. In this Chapter, if the data rate for communication is lower than 5Kbps, it is considered as no traffic. If there is no traffic for a certain period in second (hereafter this is called ThresholdWB), the extSDNC will check which wireless interface is currently activated. If the activated interface is Wi-Fi, the extSDNC will try to switch the connection to Bluetooth for saving energy. In case Bluetooth interface is used, it will be maintained. Note that if at least one of the communicating two devices is out of the Bluetooth communication coverage, the Wi-Fi connection will be maintained. The ThresholdWB is selected based on ITU-T P.59 [61] in which VoIP traffic is half duplex because Wi-Fi is also half duplex. In WLANs, radio channels are using 2.4 GHz or 5 GHz. However, APs commonly share one radio channel for both uplink and downlink because they apply an access control method called CSMA/CA (Carrier sense multiple access with collision avoidance). As a result, Wi-Fi enabled devices to communicate in a half-duplex manner where they either transmit or receive at any given time to avoid collisions. As shown in Fig. 4.2, the listener typically responses after a pause period. Thereby, it is assumed that if there is no response in two pause periods, it is possible to switch the connection to Bluetooth to save energy. Note that the Bluetooth connection still allows the users to continue the conversation, but the voice quality may be poor. Since the average of a pause period is 1.587s as given in [61], the ThresholdWB is selected as 3s. Similarly, if the data rate is greater than 5Kbps in ThresholdBW seconds, the extSDNC will also check which access technology is being used. If it is Bluetooth, it will be switched to Wi-Fi for better voice quality. The value of ThresholdBW is selected as the period of time to let both sides hear the other’s voice. As shown in Fig. 4.2, ThresholdBW is equal to twice the sum of the silence time (Mutual Silence) and the talking time (Talkspurt). In ITU-T P.59, the average Mutual Silence (M.S) and Talkspurt

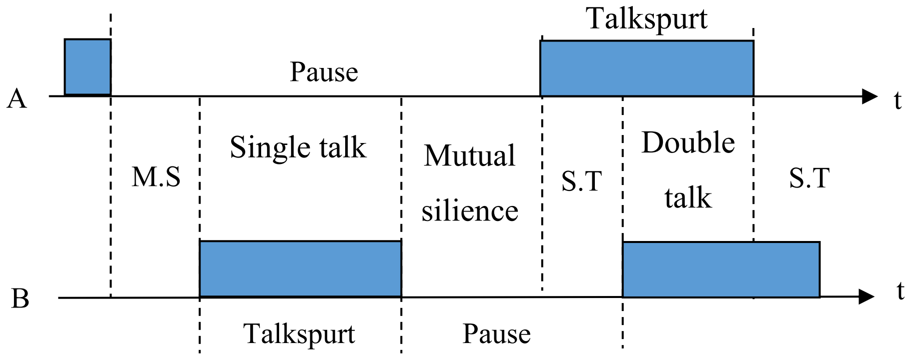


Fig. 4.2. Conversational speech model in ITU-T P.59

are 0.508s and 1.004s, respectively, thus, ThresholdBW is determined as 3s. Also, if the switching process fails, the Bluetooth connection is maintained.

On the next step, the extSDNC prepares for switching the connection. Based on the backup access network, a corresponding association procedure is generated and saved in a shell file waiting for the call from the extSDNC. OpenFlow rules are also defined and described so that they can be sent to the traditional-SDNC as well as directly to the switch if necessary. The information of MAC and IP addresses for the backup interface are also prepared. For the sake of simplicity, it is assumed that transferring packets, i.e., ARP, ICMP, TCP, UDP, and IP, in the network are already known. Traffic classification is out of scope in this work.

To execute a handover between two wireless access technologies, the extSDNC on each device makes the mobile device abandon the current connection and get attached to the other network, which is selected in the previous step. To ensure no packet is lost in this process, several protocols have been standardized [26, 27, 28, and 29]. The more complex in this procedure is, the more network overhead is generated, and the longer the handover delay is. Therefore, in this Chapter, a simple socket message exchange procedure, which allows both sides to understand the other's status, is used. Since the procedure is simple, it reduces the handover delay. The detail of this procedure is illustrated in Fig. 4.3. Firstly, the extSDNC on each device checks which interface is being used and which one is in the backup state. The extSDNC then wakes the backup interface up and associates it with its corresponding network. After finishing the association process, the device must enter the synchronizing step, which is shown in Fig. 4.4, to make sure the other side has also finished this association process and to allow both

```
Get current activated interface
Switch to another interface based on activated one
  Case "Bluetooth"
    Turn Wi-Fi interface on
    Associate Wi-Fi interface to an access point
    If association process finished:      Perform synchronization process
  Case "Wi-Fi"
    Turn Bluetooth interface on
    Join a Bluetooth piconet
    If pairing process finished:      Perform synchronization process
Open a thread to execute network configuration
Open a thread to install OpenFlow rules
Update the local database
```

Fig. 4.3. Handover execution

```
Timer is ON
If received the SYN message from the other device
// which already finished its association process
  Notify him with a SYN message
  Do {wait for partner response
Else
  Notify him with a SYN message
  Sleep in RTT/2
End if
```

Fig. 4.4. Synchronization process

sides to switch to the other access network at the same time. Before the association process begins, a timer is set in case there is an interference between Bluetooth and Wi-Fi. When the interference occurs, the exchanged packets may get lost or spend a long time to reach the destination. In that case, when the timer counts down to zero, the association process is over.

Otherwise, in a normal association process, the device first checks if it receives any message from the other one to verify whether that device finishes the association process on the backup interface yet. If there is no message, meaning that the other side has not finished associating the interface, the device sends an SYN message to him informing that this side has finished the association process. After that, the device waits for the other side device's response. Note that the traffic still goes through the active interface during this time. In case the other side device has already finished the association process, he will reply with an SYN message as the confirmation and then wait for a half of RTT to let the message reach the destination. After both sides have received the SYN message, each one performs network configuration and installs new OpenFlow rules. Note that these processes need to perform at the same time to minimize the handover delay, thus, multithreading or multiprocessing technique is used. As a result, the handover delay will be the largest in the period from when configuring the network to when installing the rules. When the network configuration process finishes, the extSDNC must send the ARP packets to the other side advertising the new interface with new MAC and IP addresses. After the ARP packets are exchanged, two devices can continue to communicate via the new interface.

4.3 D2DHWC: Device-to-device heterogeneous wireless communication

Although tethering, which is supported in both Wi-Fi and Bluetooth, allows sharing the Wi-Fi or Internet connection of the mobile device with other devices, it cannot maintain the connection when the device changes the access technology. To this end, D2DHWC is designed to enable two mobile devices to communicate without concerning which technology they are

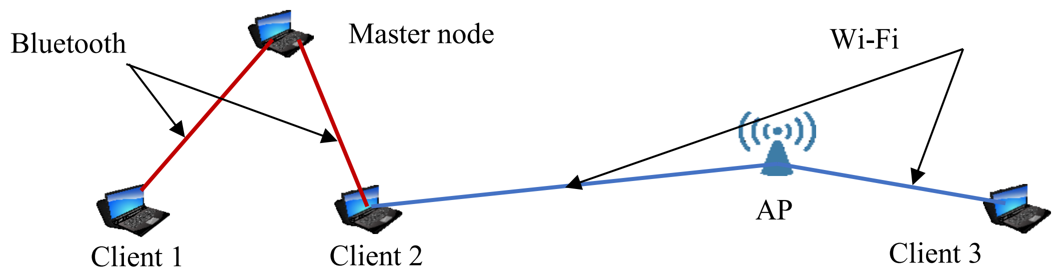


Fig. 4.5. Schematic of the sustainable heterogeneous communication network

using. D2DHWC allows low battery devices, which need Bluetooth technology to prolong the operating lifetime, to communicate with the others. One case study for this mode is depicted in Fig. 4.5. The figure shows that on the left side, there are three nodes named Client1, Client2 and the Master node which are communicating in a Bluetooth piconet. On the right side, Client2 connects to the Client3 via the Wi-Fi AP. To allow Client1 to talk with Client3 without changing the access network, Client2 volunteers as a repeater node. On Client2, both Bluetooth and Wi-Fi interfaces are turned on and all incoming packets from Client1 and Client3 are passed via Wi-Fi and Bluetooth interface, respectively. Note that while Client2 acts as a repeater, it is still able to communicate with other nodes in the piconet via the Bluetooth interface as well as with Client3 via Wi-Fi. This is feasible because all physical interfaces of Client2 are under control of the Open vSwitch. The switch navigates traffic by asking a traditional-SDNC. The traditional-SDNC refers to the collected network information in the "Local DB" provided by the extSDNC to correctly answer the switch. Once received the response from the traditional-SDNC, the switch saves the rule as an entry in its flow table. Thereafter, any packets that meet the stored flow rules are forwarded through the switch without referring to the traditional-SDNC. Moreover, the Open vSwitch has the ability to rewrite the packets' header, hence, the source and destination MAC and IP addresses of the incoming packets from an interface can be rewritten and sent to the other interface. Note that to enable the packets to reach their destination, ARP packets have already been handled by the extSDNC.

4.4 Performance evaluation

Experiments were conducted in a real testbed in which several Linux computers were connected via Wi-Fi or Bluetooth. Each computer was equipped with Core 2 Duo @2.26 GHz processor and 2GB RAM. On each computer, two USB-based wireless adapters including a wireless LAN adapter and a Bluetooth 4.0 one have been installed for accessing Wi-Fi and Bluetooth networks, respectively.

4.4.1 Evaluation of energy-efficient communication

4.4.1.1 Measurement of energy consumption

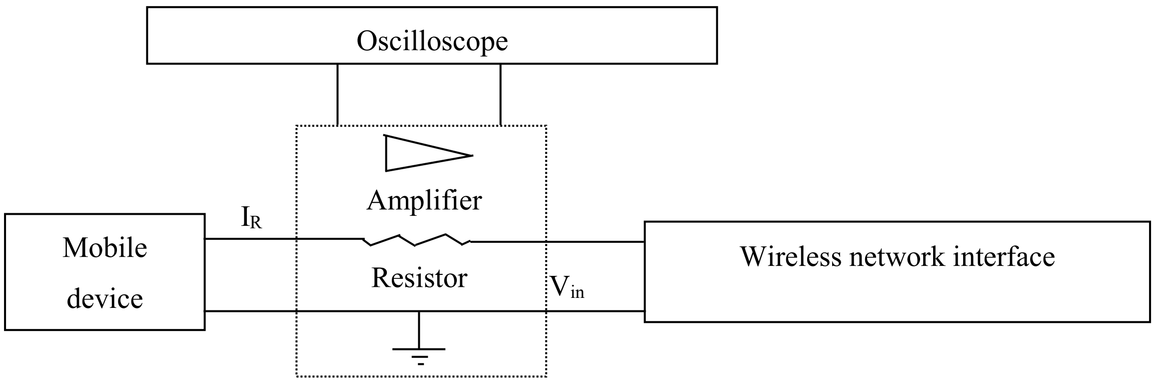


Fig. 4.6. Measurement setup

In this section, the energy consumption in different states of wireless network interfaces, i.e., Wi-Fi and Bluetooth is presented. The states in the experiment include “wake up”, sleep, active and off. The “wake up” state starts when the interface is activated and finishes when the interface has completed the association process. The sleep state in this work is defined as the state where the mobile device has connected to a wireless network, but no applications on the device are using the wireless connection. In the event of active, the interface can transmit or receive the network traffic which has the rate higher than 5Kbps. When the interface is off, it is still attached to the mobile device, but it is disabled.

The experiment setup to determine the power consumed in each event is illustrated in Fig. 4.6. In the figure, a wireless network interface is connected to a mobile device, which has been equipped with the proposed system, via a 0.5-ohm shunt resistor. The obtained current $I_R(t)$ needs to be amplified before being sent to the oscilloscope since the value of the current can be very small in the case of Bluetooth. The voltage $V_{In}(t)$ dropped on the network interface is measured directly. The instantaneous power consumption $P(t)$ of the wireless interface is computed based on the Ohm's law: $P(t)=I_R(t)V_{In}(t)$. After that, the average power consumption is calculated, similarly to the technique used in [36], within a period of time that approximately wraps a state. For example, the average time for Bluetooth interface to be woken up and be associated with the master node is T_{Wakeup}, thus, the average energy consumed in this state is the average value of $E(t)|_0^{T_{Wakeup}}$. The measured results of the average energy consumption of each wireless interface in different events are shown in Table 4.1. As seen from the table, for completing the "wake up" process, Bluetooth consumes 417.98 millijoule while Wi-Fi consumes more energy, which is 1.2 times of Bluetooth. The comparison unit is millijoule since Wi-Fi and Bluetooth complete the "wake-up" process in different T_{Wakeup} time. The average value of measured T_{Wakeup} of Wi-Fi and Bluetooth are 1.4s and 3.03s, respectively. In the sleep and active states, Wi-Fi consumes energy more than 6 times of Bluetooth. Besides, although the interface has been disabled in the off state, it still consumes energy.

TABLE 4.1: AVERAGE POWER CONSUMPTION OF WIRELESS NETWORK INTERFACES

Interface	Wake-up (mJ)	Sleep (mW)	Active (mW)	Off (mW)
Wi-Fi	536.77	495.05	660.09	213.75
Bluetooth	417.98	79.24	104.21	38.20

4.4.1.2 Evaluation of energy saving

This evaluation is to show how the proposed system saves the energy consumed by the wireless communication. In the system, the energy-efficient vertical handover algorithm allows the wireless connection to be switched from Wi-Fi to Bluetooth whenever there is no traffic. When there is no traffic, the power-saving mode (PSM) will turn Wi-Fi to sleep state to consume $P_{wi\text{-}Fi\text{-}sleep}$=495.05mW. Although the Bluetooth interface is disabled, it still consumes $P_{Blue\text{-}off}$=38.20mW. The average power consumed by the wireless communication is $P_{wireless} = P_{wi\text{-}Fi\text{-}sleep} + P_{Blue\text{-}off}$ = 533.25mW. This means the wireless communication consumes 533.25mJ

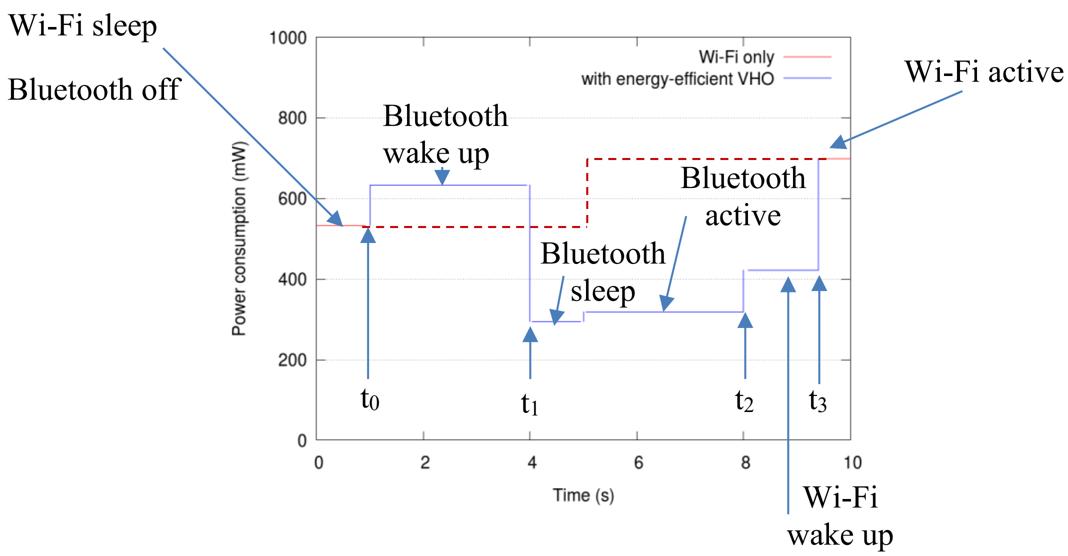


Fig. 4.7: Energy saved when performing a vertical handover

every second. Assumed that at t_0=1s the system starts the vertical handover process. The power consumed by the wireless communication at t=t_0 is 533.25mW as shown in Fig. 4.7. Before the device can use the Bluetooth connection, Bluetooth interface must be woken up and be associated with the master node. This process lasts $t_{Blue\text{-}wakeup}$=3.03s on average and consumes $P_{Blue\text{-}wakeup}$=417.98/3.03=137.95mW. In addition, during this time, Wi-Fi is still in sleep mode and consumes $P_{wi\text{-}Fi\text{-}sleep}$ (495.05mW). The total energy consumed at t_1=$t_{Blue\text{-}wakeup}$ is $E_{wireless}(t)|_{t_0}^{t_1}$ = (137.95+495.05)x3.03 = 1917.99mJ. The consumed energy in this period is higher than the energy consumed by the sleeping Wi-Fi interface and Bluetooth interface, which is in the off state, $E_{wifi-only}(t)|_{t_0}^{t_1}$ =(495.05+38.20)x3.03=1615.75.12mJ.

After Bluetooth is woken up, Wi-Fi is disabled, then the total energy consumption will be $E_{wireless}(t)|_{t_1}^{t_1+t_{Blue-sleep}}$=(79.24+213.75)$t_{Blue\text{-}sleep}$ (mJ) if Bluetooth interface sleeps in $t_{Blue\text{-}sleep}$ seconds. In the worst case, there is some traffic in the network right after the handover process has finished. In that case, $t_{Blue\text{-}sleep}$ = 0.

When there is traffic in the network, Bluetooth interface switches to active state and consumes the energy $E_{Blue-active}(t)|_{t_1}^{t_1+t_{Blue-active}}$=104.21$t_{Blue\text{-}active}$ (mJ). Due to the proposed algorithm, the time needed to decide for switching back to Wi-Fi is 3s, hence, $t_{Blue\text{-}active}$=t2=3 seconds. During this period, the disabled Wi-Fi interface consumes $P_{Wi\text{-}Fi\text{-}off}$=213.75mW. The total energy consumed by the wireless communication is $E_{wireless}(t)|_{t_1}^{t_2}$ =

(104.21+213.75)x3=953.88mJ. If the system does not use the proposed algorithm, Wi-Fi

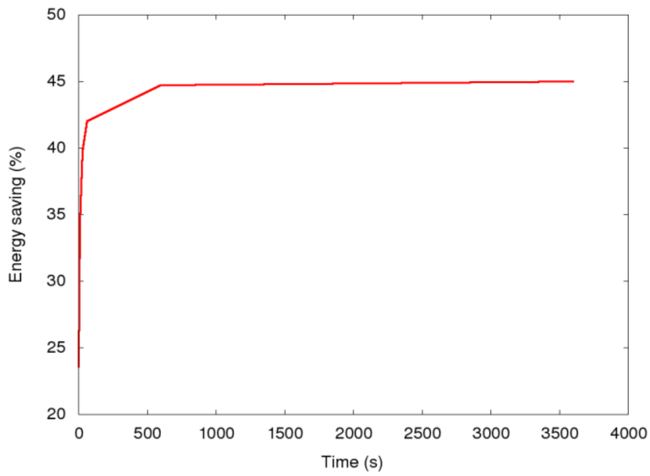


Fig. 4.8. Energy saved by using proposed system

connection will be used all the time. Then, the energy consumed when Wi-Fi is active is $E_{Wi-Fi-only}(t)|_{t_1}^{t_2}$= (660.09+38.2)x3=2094.87mJ.

To switch the connection back to Wi-Fi, Bluetooth interface is disabled, Wi-Fi interface is woken up and is associated with the AP. The average time for Wi-Fi to complete this process is $t_{Wi\text{-}Fi\text{-}wakeup}$=$t_3$=1.4 seconds. The system consumes $P_{Wi\text{-}Fi\text{-}wakeup}$=536.77/1.4=383.41mW to wake the Wi-Fi interface up and associate it with an AP. The total amount of consumed energy during this time is $E_{wireless}(t)|_{t_2}^{t_3}$=(383.41+38.2)x1.4=590.25mJ. The energy consumed by using only Wi-Fi is much higher $E_{Wi-Fi-only}(t)|_{t_2}^{t_3}$ =(660.09+38.2)x1.4=977.60mJ. The total amount of energy consumed by only Wi-Fi is $E_{Wi-Fi-only}(t)|_{t_0}^{t_3}$=4593.54mJ while the energy consumption by using the proposed algorithm in the worst case is $E_{wireless}(t)|_{t_0}^{t_3}$=3471.66mJ. This means, the proposed system has saved 24.42% of the total energy consumption by the wireless communication. The Fig. 4.8 shows the amount of energy saved when the $t_{Blue\text{-}sleep}$ increases. For example, the system saves 44.7% energy consumption when the $t_{Blue\text{-}sleep}$=600s. The obtained results have confirmed that in any cases, the proposed system can save the energy consumed by the wireless network interfaces.

4.4.2 Evaluation of heterogeneous communication

4.4.2.1 Evaluation of the communication means

In order to evaluate the performance of the communication means, the network quality has been observed when a vertical handover occurs. The evaluation metrics is the handover delay, which is defined as the largest in the time period from when configuring the network to when installing the OpenFlow rules.

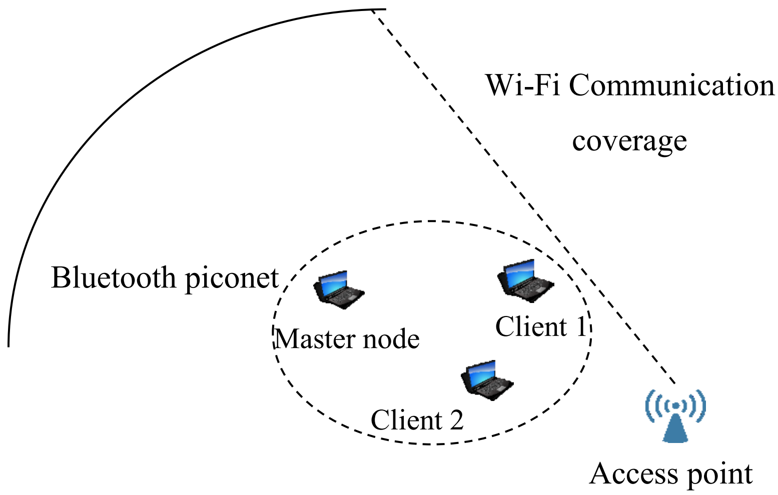


Fig. 4.9. Experiment topology

The testbed in this evaluation consists of three mobile devices as illustrated in Fig. 4.9. The figure shows that the devices are communicating in a Bluetooth piconet for saving energy. It is assumed that all the devices accept the participations from other devices. The devices are also able to join the 802.11g network in this testbed by utilizing the built-in Wi-Fi interface. The 802.11g protocol was selected for the test as the Wi-Fi connection simply gives a faster transmission rate and a larger communication coverage. In the evaluation, Iperf [51] is used to generate a UDP flow toward the direction from Client1 to Client2. The execution times on both Client1 and Client2 were captured by utilizing a Python module named "time" and the results are given in Fig. 4.10. As shown in the figure, for each process, the average and 95% confidence interval of the execution times were measured. The measured results with two nodes are similar. In case the connection is going to be switched from Bluetooth to Wi-Fi, both t_{WOF} and t_{WNW} for Wi-Fi network are around 80ms and the difference between them is small. In the invert direction, t_{BNW} varies from 80ms to 150ms and t_{BOF} is around one sixth of t_{BNW}. The

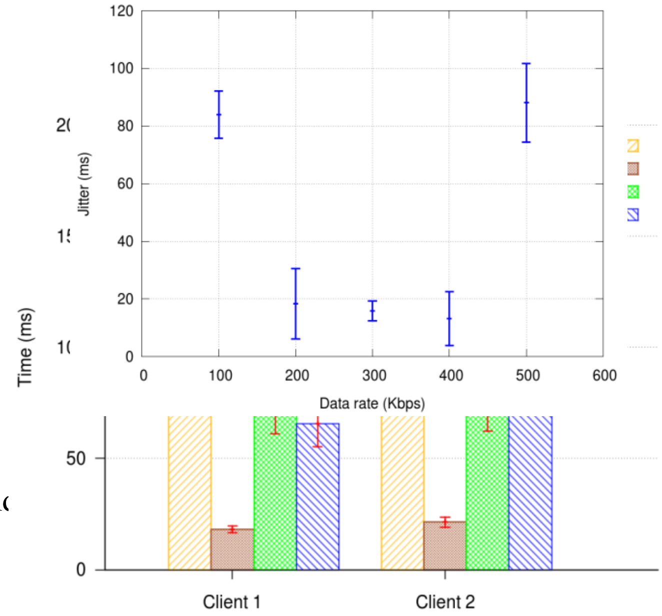


Fig. 4.11. Enc ation network

Fig. 4.10. Handover delay measurement

larger the difference between t_{BOF} and t_{BNW} is, the larger the number of packet loss is. The packets are lost because they are navigated to another interface due to the newly installed OpenFlow rules. However, the interface has not been ready yet. Ideally, the configuration and the installation should be processed at the same time, and thus the packet loss does not occur.

The experiment results have confirmed the flexibility of the proposed system in the bidirectional vertical handover between Bluetooth and Wi-Fi with different UDP traffic load. In addition, the handover disruption time, which causes the handover delay, was measured in the experiment. The result shows that the handover delay was still less than 150ms, which is acceptable for VoIP applications as recommended by ITU-T Recommendation G.114 [68].

4.4.2.2 Evaluation of the heterogeneous communication network

To evaluate the quality of the heterogeneous communication network in supporting real-time services, i.e., VoIP data, the QoS parameters including packet loss rate and jitter have been measured. These parameters were selected because packet loss in VoIP will typically reduce the quality of the voice communication. The variation in the delay of received packets, or jitter, is also a reason of discarded packets if it is so large.

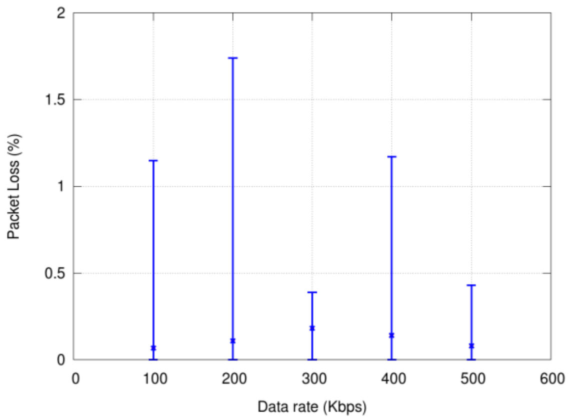


Fig. 4.12. Packet loss rate of the robust heterogeneous communication network

Figure 4.9 shows the real testbed in this evaluation. In the figure, on the right side of topology, the Client2 and Clinet3 are attached to the AP, through the 802.11g Wi-Fi interface. On the left side, the Client2 is also a member of a Bluetooth piconet which has a Master node and another client named Client1. The Client2`s duty is to pass the traffic between Client1 and Client3.

In the evaluation, UDP traffic from Client3 to Client1 was generated using Iperf. The UDP protocol was selected because the VoIP data is placed in the packets of Real Time Protocol (RTP) and RTP normally uses UDP as the default transmission protocol. The traffic throughput was varied from 100Kbps to 500 Kbps. In this experiment, the QoS parameters including jitter and packet loss rate were observed since they were selected as the evaluation metrics. Each experiment was repeated 30 times and the results are presented in Fig. 4.11 and Fig. 4.12.

Figure 4.11 shows that when the throughput was between 200 Kbps and 400 Kbps, the average end-to-end jitter was less than 20ms. When throughput was 500Kbps or very small, i.e., 100Kbps, the jitter was around 85ms. This is because there was congestion when the traffic throughput was high. In contrast, when the traffic throughput was low, the number of sending packets is small, hence, if a few packets are delayed in reaching the destination, the average delay still increases. Besides, when the data rate is 300Kbps the jitter is more stable and less than 20ms.

Figure 4.12 shows the end-to-end packet loss rate of the heterogeneous communication network. As shown in the figure, the minimum packet loss rate is zero with any data rate.

Although the rate is fluctuated, it is under 0.2% on average. When the data rate is 300 Kbps, the loss rate is more stable and the maximum packet loss rate is 0.39%.

The obtained results have indicated that the system can provide a heterogeneous communication network with the packet loss rate of less than 0.2% and the jitter of less than 20ms when the data rates are varied from 200 Kbps to 400 kbps. The results also pointed out that the proposed network meets the requirement of real-time IP-based services as recommended by ITU-T Y.1541 [69].

4.5 Conclusions

In this Chapter, a device-to-device heterogeneous wireless communications network for victims in disaster scenarios has been introduced. Also, the performance of the system has been evaluated on four major criteria: energy saving, pack loss rate, jitter, and handover delay. The system succeeded in dynamically performing handovers between Bluetooth and Wi-Fi in any direction when the traffic was traveling across the system. The handover delay was as small as 85ms when switching the connection from Bluetooth to Wi-Fi. In the invert direction, the delay was larger, however, it was still less than 150ms which is the maximum value of one-way latency recommended by ITU-T G.114 [68]. Thereby, the proposed system can save at least 24.42 percent the energy consumed by the wireless communication. Moreover, the system can allow a mobile device using Bluetooth to talk with another one using Wi-Fi over a heterogeneous connection. The measured end-to-end jitter and packet loss rate show that the proposed network is acceptable for real-time IP-based services referring to ITU-T Y.1541 [69].

Chapter 5 VHO decision in wireless multi-hop network

In chapter 4, a VHO trigger algorithm for heterogeneous wireless communication between communicating devices has been proposed. When the distance between the communicating devices are out of the communication range, they may need support of other devices to communicate in a multi-hop manner. In this wireless multi-hop network, the devices are to relay messages are called intermediate devices. The intermediate devices relay messages which are not sent to them, hence, it is necessary to reduce the energy for relaying such messages. In this chapter, an energy-efficient VHO trigger mechanism will be proposed to address that issue.

5.1 Assumed wireless multi-hop network

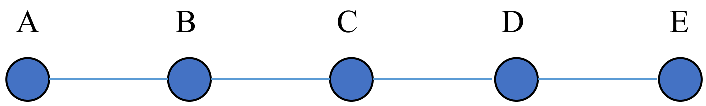


Fig. 5.1. Wireless multi-hop network topology

The devices communicate to the others by passing messages from device to device, so called a multi-hop communication. In such a wireless multi-hop network, the communication between two devices commonly requires intermediate devices' help to relay messages. The considered scenario in this work is illustrated in Fig. 5.1. In the figure, several mobile devices equipped with an extended SDN controller forms a wireless multi-hop network. On each device, Bluetooth and Wi-Fi are integrated as a standard feature, allowing devices to be able to join a Wi-Fi or Bluetooth network. In this work, it is assumed that the possibility of Bluetooth/Wi-Fi interference has been minimized. The devices basically use Wi-Fi for communication. When device A wants to communicate with device E, they need the cooperation of other intermediate devices. Trivially, the intermediate devices have to consume their battery power for the messages that are not sent to them.

5.2 Wireless multi-hop network management

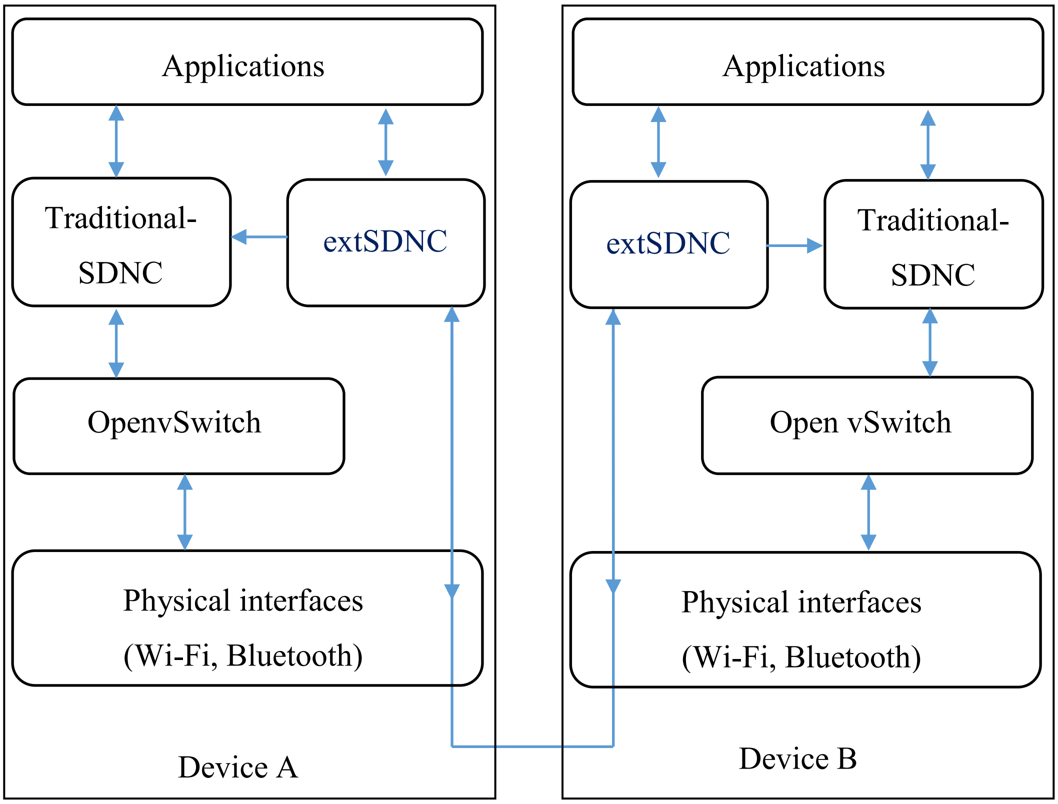


Fig. 5.2. Controller to controller communication

The control logic on mobile devices as well as the controller to controller communication, which is used in the management of the network, is given in Fig. 5.2. On each device, the traffic goes through physical interfaces managed by the virtual interface created by a software OpenFlow switch called OpenvSwitch. The switch controls the physical interfaces by asking the traditional-SDNC. Since the traditional-SDNC is embedded in a device and becomes a local controller, it needs support from the extSDNC, which has been introduced in Chapter 2. The main function of the extSDNC is to collect network state to provide necessary information to the traditional-SDNC. The traditional-SDNC then can instruct the switch to navigate traffic correctly. The extSDNC can also be used to create a controller to controller communication channel. The channel is useful for the management of the network as there is no centralized control. For example, the extSDNC can get the bandwidth utilization of each application and send the required bandwidth to other devices. Based on the exchanged information, the other devices can prepare a corresponding bandwidth for the connection. Since the devices are equipped with Wi-Fi and Bluetooth, if the required bandwidth is small enough, they can use Bluetooth instead of Wi-Fi for energy-saving. Note that to allow a device using Wi-Fi to send messages to another device using Bluetooth, the system must select two intermediate devices, which are the nearest ones to the communicating devices, to act as supporter devices. The nearest devices can be selected based on hop-count. The role of supporter devices is to turn on both Wi-Fi and Bluetooth and then to forward messages from Wi-Fi to Bluetooth, and vice versa.

5.3 Energy efficient wireless multi-hop network

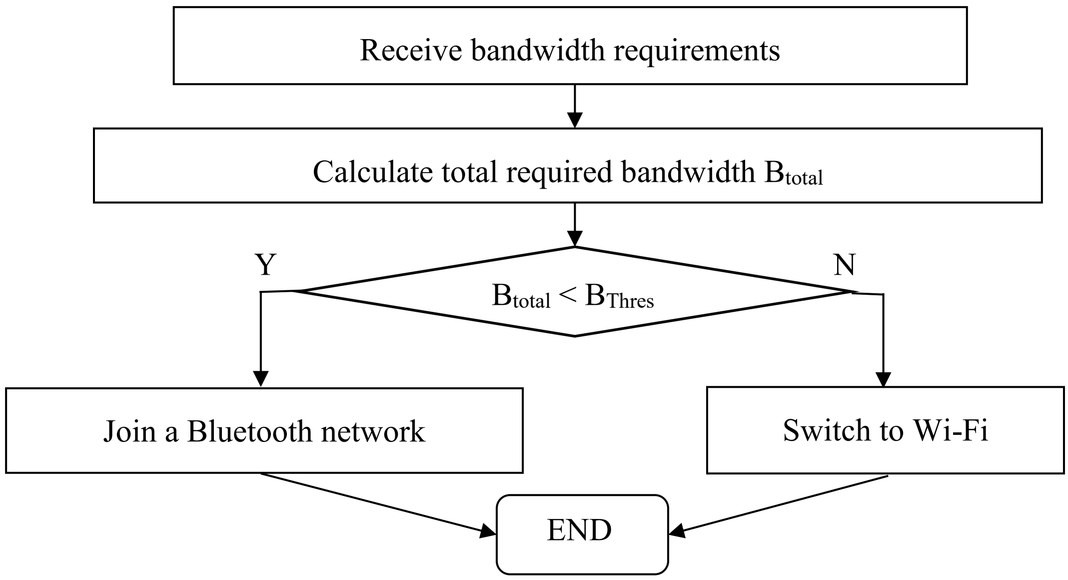


Fig. 5.3. Handover triggering algorithm

Since the intermediate devices in wireless multi-hop network have to spend their energy for relaying messages which are not sent to them, the least energy-consuming technology should be used. Specifically, if the required bandwidth for the communication is not high, i.e., less than Bluetooth capability, it is a waste of energy if they keep using an energy consuming protocol like Wi-Fi. In this case, turning off Wi- Fi and switching all the running sessions to Bluetooth not only guarantee the requirement but also save energy.

The algorithm for triggering the energy-efficient vertical handover on the intermediate device is given in Fig. 5.3. Firstly, all the devices form a wireless multi-hop network by using Wi-Fi network. On the device, which wants to communicate with another, the extSDNC sends a text file in which the bandwidth requirements are described to the other intermediate devices. Since the text file size is small and is sent only one time in the beginning of the communication, it is assumed that the energy consumption of this process is very small. On the other hand, the extSDNCon each intermediate device basically sleeps, it only wakes up to receive information from communicating devices. Based on the received information, the extSDNC calculates the total required bandwidth B_{total}. If B_{total} is smaller than a threshold B_{Thres}, the extSDNC will perform a vertical handover from Wi-Fi to Bluetooth. If the Bluetooth has already been used, it will maintain the connection. Similarly, if B_{total} is bigger than B_{Thres}, the extSDNC will switch the connection to Wi-Fi. If the Wi-Fi is already ON, the Wi-Fi connection will be retained. Since the extended controllers on the intermediate devices mainly sleep, it is assumed that they do not consume much energy.

5.4 Evaluation

5.4.1 Power consumption measurement

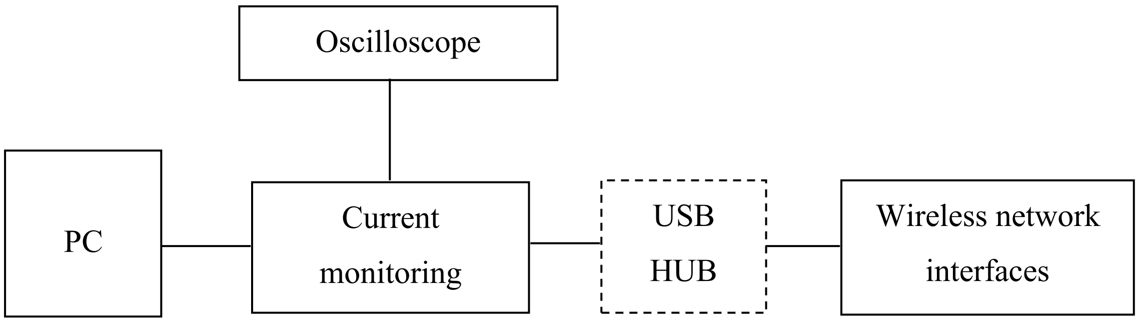


Fig. 5.4. Measurement setup

The power consumption of wireless technologies including Wi-Fi and Blueooth has been measured by using a real testbed illustrated in Fig. 5.4. The PC is a mobile device, which is a Linux computer (Core 2 Duo @2.26 GHz processor and 2GB RAM, Ubuntu 14.04 64-bit). The PC connects to Wi-Fi and Bluetooth networks using USB-based Wi-Fi and Bluetooth adapters, respectively. When measuring how much energy has been saved by using the proposed method, a USB hub is used to collect the total energy consumed by the wireless communication. Note that the hub itself consumes energy and adds noise to the measured result. However, the measurement results are acceptable since its target is to compare the energy consumed by using only Wi-Fi with PSM enabled with that by using the proposed method. When measuring the energy consumed in different states of the wireless interface, i.e., sending and receiving, the hub will be removed. The measured results are given in Fig. 5.5, 5.6, 5.7 and 5.8.

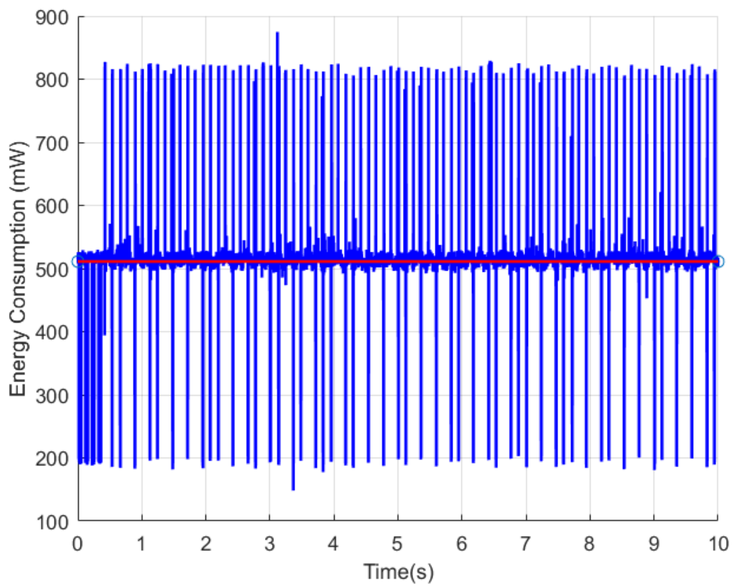


Fig. 5.5: Power consumption when Wi-Fi is used for transmitting traffic

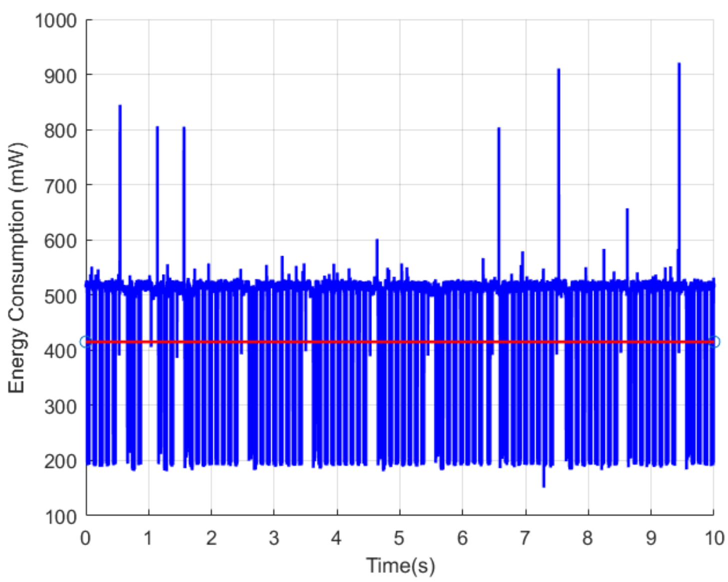


Fig. 5.6. Power consumption when Wi-Fi is used for receiving traffic

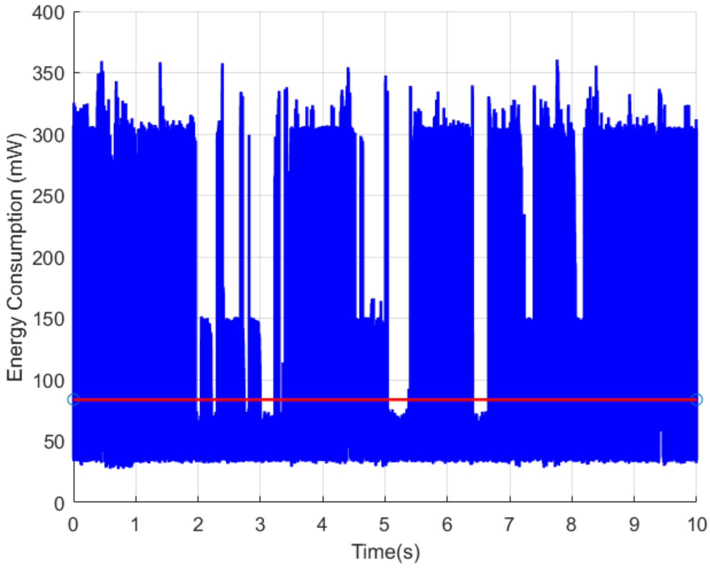


Fig. 5.7. Power consumption when Bluetooth is used for transmitting traffic

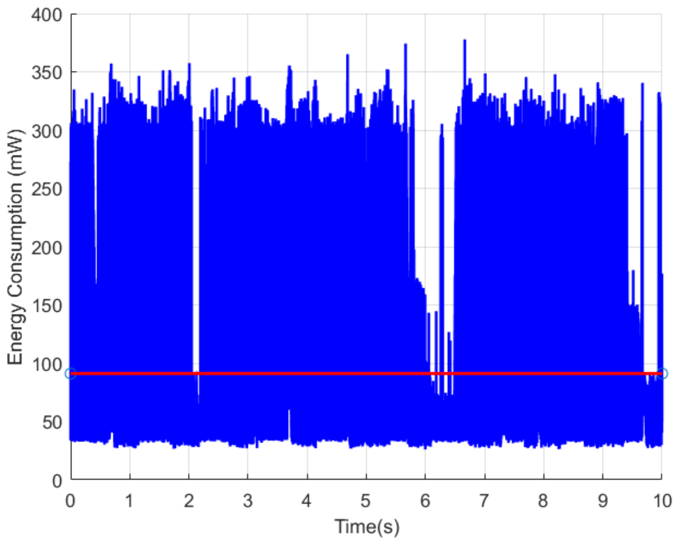


Fig. 5.8. Power consumption when Bluetooth is used for receiving traffic

5.4.2 Energy consumption measurement on intermediate devices

To examine the energy consumed on device C, Iperf [51] is used to send a 300 Kbps UDP flow from device A to device E as shown in Fig. 5.1. While traffic is being sent via Wi-Fi, device C receives information of the required bandwidth B_{req} from device A. Since B_{req} is smaller than

1Mbps, it performs a vertical handover from Wi-Fi to Bluetooth. If device C does not receive any invitation for communication or the bandwidth requirement keeps being less than 1Mbps, it will maintain the Bluetooth connection. Otherwise, it must switch the connection back to Wi-Fi. In the worst case, right after device C turns Wi-Fi off and successfully joins a Bluetooth network, it must turn Bluetooth off and then turn Wi-Fi on. The energy consumed by the wireless communication in the worst case is given in Fig. 5.9. From the figure, the average power consumption when switching the network interface to use Bluetooth connection is $P_{Bluetooth}$=670.92mW. Since this is the worst case, the Bluetooth connection has been used for $T_{worstcase}$=3.24 seconds. The average power consumed by using only Wi-Fi is $P_{Wi\text{-}Fi\text{-}only}$=939.47mW, thus, the energy saved in the worst case is E_0 (mJ):

$$E_0 = (P_{Wi\text{-}Fi\text{-}only} - P_{Bluetooth}) T_{worstcase} = 870.102\text{mJ}$$

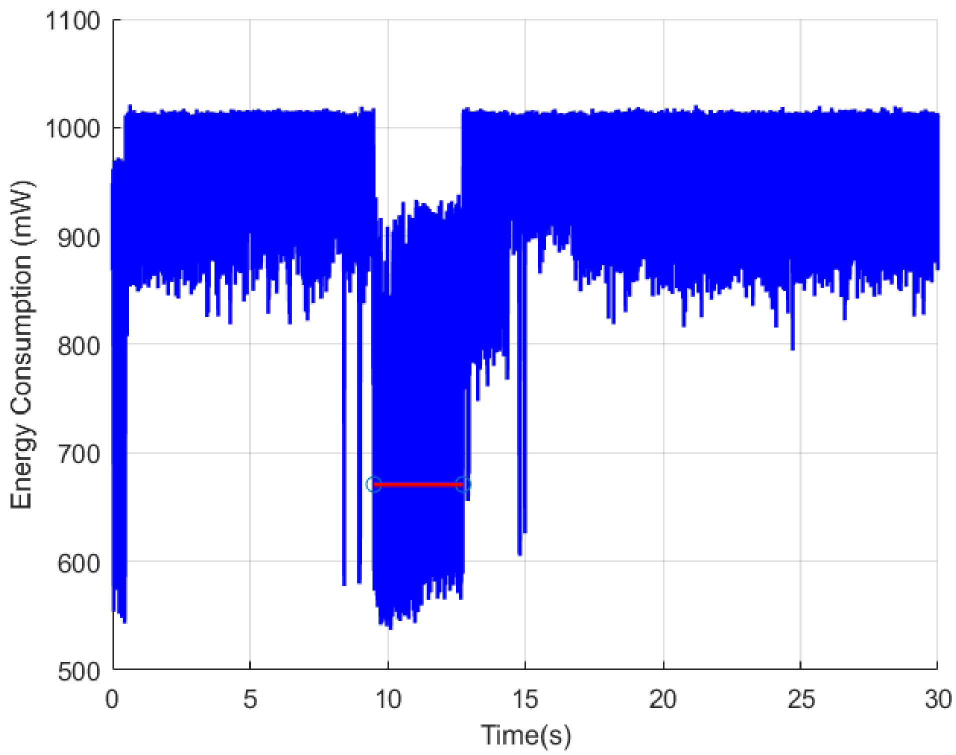


Fig. 5.9: Energy consumed by the wireless communication when using proposed method

If device C maintains the Bluetooth connection in T_M (seconds) it will save E_{saved} (mJ):

$$E_{saved} = E_0 + (P_{Wi\text{-}Fi\text{-}only} - P_{Bluetooth})\, T_M$$

$$= 870.102 + 268.55 x T_M \text{ (mJ)}$$

If there are N intermediate devices that can switch their connection from Wi-Fi to Bluetooth, the total energy saved in the whole network is $Total_{saved} = N \, x \, E_{saved}$ (mJ).

The measured results have confirmed that the proposed method can save the energy consumption on intermediate devices. The longer the middle devices can use Bluetooth, the more energy can be saved.

5.4.3 Energy consumption measurement on supporter devices

As discussed in section II, to allow intermediate devices like device C to perform the energy-efficient vertical handover, intermediate devices such as device B and device D must act as supporter devices by turning on both Wi-Fi and Bluetooth at the same time. A supporter device works by receiving traffic from Wi-Fi interface and forwarding it to Bluetooth interface, and vice versa. Let ($P_{Wi\text{-}Fi\text{-}transmit}$, $P_{Wi\text{-}Fi\text{-}receive}$) and ($P_{Bluetooth\text{-}transmit}$, $P_{Bluetooth\text{-}receive}$) denote the power consumed on the supporter device for Wi-Fi and Bluetooth to transmit and receive traffic, respectively. When device A sends a message to device E, the supporter device consumes $P_{Wi\text{-}Fi\text{-}receive}$ (mW) for receiving the message. To forward the message to another device, toward to device E, it consumes $P_{Bluetooth\text{-}transmit}$ (mW). Similarly, on the inverted direction, the supporter device consumes $P_{Bluetooth\text{-}receive} + P_{Wi\text{-}Fi\text{-}transmit}$ (mW) for relaying the message. The total power consumed on a supporter device for passing the exchanged messages is P_S (mW):

$$P_S = P_{Wi\text{-}Fi\text{-}receive} + P_{Bluetooth\text{-}transmit} + P_{Bluetooth\text{-}receive} + P_{Wi\text{-}Fi\text{-}transmit}$$

If the device uses only Wi-Fi for passing messages, it will consume $P_{S\text{-}Wi\text{-}Fi\text{-}only}$ (mW):

$$P_{S\text{-}Wi\text{-}Fi\text{-}only} = 2 \, x \, (P_{Wi\text{-}Fi\text{-}receive} + P_{Wi\text{-}Fi\text{-}transmit})$$

To compare P_S and $P_{S\text{-}Wi\text{-}Fi\text{-}only}$, the power consumption on a supporter device to transmit or to receive traffic via Wi-Fi or Bluetooth has been measured. The transmitting power consumption was measured when the device was sending out a 300Kbps UDP flow. The receiving power consumption was measured when the device was receiving a 300Kbps UDP flow.

The obtained results are given in Fig. 6, 7, 8 and 9. From the figures, the red horizontal line shows the average power consumption when the supporter device receives or transmit the traffic via Wi-Fi or Bluetooth respectively are $P_{Wi\text{-}Fi\text{-}receive}$=415.27mW, $P_{Wi\text{-}Fi\text{-}transmit}$=511.12mW, $P_{Bluetooth\text{-}receive}$=91.14mW, $P_{Bluetooth\text{-}transmit}$=83.80mW. As a result, the supporter device can save $P_{supporter\text{-}save} = P_{S\text{-}Wi\text{-}Fi\text{-}only} - P_S$ = 752.45mW even if it turns on both Wi-Fi and Bluetooth interfaces.

The measured results have confirmed that the proposed method can save the energy consumption on supporter devices. The amount of saved energy is depended on the volume of traffic, which the supporter devices relay.

5.5 Conclusion

The goal of this work is to reduce the energy consumption on the intermediate devices which are cooperating to maintain the connectivity for a communication between two end devices. In this paper, an energy efficient vertical handover algorithm has been proposed. The vertical handover was performed by the devices themselves based on the exchanged information between extSDNC, which has been proposed in Chapter 2. The effectiveness of the system has been evaluated by measuring the energy consumption of wireless network interfaces directly. The measured results have confirmed that all devices which are cooperating to help two other devices to communicate can save energy. Specifically, each intermediate device can save at least 870.102mJ and the device can save more energy if it stays longer in the Bluetooth network. The supporter devices can also save 752.45mW each time relaying despite turning on both Wi-Fi and Bluetooth interfaces.

Chapter 6 esVHO: energy-saving solution for wireless communication

Although the extSDNC proposed in chapter 2 can perform VHO based on a link layer trigger the number of packet loss is still significant. Therefore, in this Chapter, a framework named esVHO is proposed to reduce the number of packet loss while saving energy consumption for the wireless communication.

The esVHO framework is an improvement of extSDNC. It also operates by using a low power consumption technology, i.e., Bluetooth, as a replacement of the energy-consuming technology, i.e., Wi-Fi, whenever the running application and services do not require high data rate. The esVHO framework firstly reduces the number of packet loss generated by the VHO process on each mobile device. In addition, when two communicating devices perform VHO at almost the same time, the difference between when these VHO processes finish is also a source of packet lost. The esVHO framework addresses this issue by introducing a VHO timing adjustment mechanism.

6.1 esVHO design

6.1.1 Assumed non-interconnected environment

In this work, it is supposed that two mobile devices equipped with a local SDN architecture are communicating as illustrated in Fig. 6.1. The devices generally use Wi-Fi to connect to the other via a wireless local area network (WLAN). The Wi-Fi connection is provided by an AP or is established in an adhoc manner. The devices are assumed to be able to join a wireless personal area network (WPAN) using Bluetooth. To avoid the interference between Bluetooth and Wi-Fi, Wi-Fi uses the 5GHz band while Bluetooth uses 2.4 GHz band.

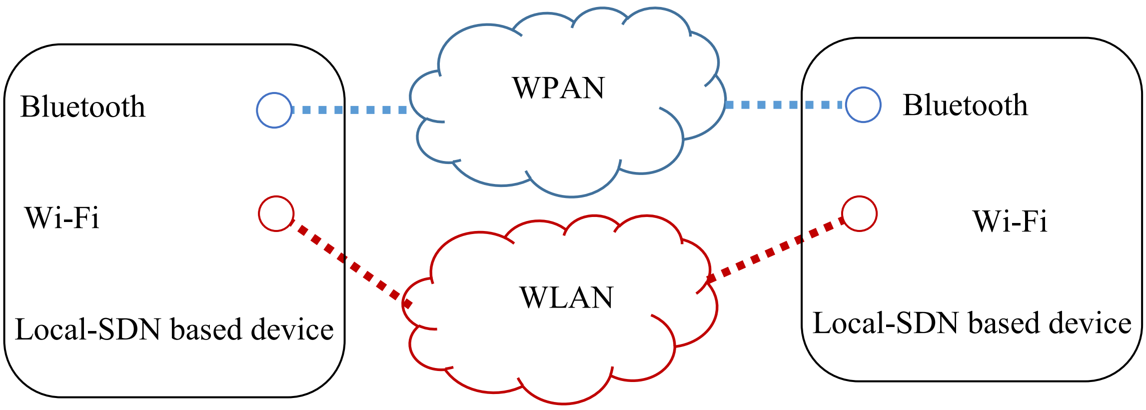


Fig. 6.1: Non-interconnected networks

In such a non-interconnected environment, different applications or services like low-bandwidth or bandwidth-hungry ones, i.e., chat and file exchange applications, can run simultaneously. Low-bandwidth applications or services are assumed to last for several minutes to a few hours. While the bandwidth-hungry ones are assumed to operate in a short duration, i.e., several seconds. This means the applications or services can start and finish at any time. Also, the data traffic generated by the application can flow in any direction.

6.1.2 Design requirement

6.1.2.1 Reduce the packet loss rate and VHO execution time.

The higher layer of OSI model [52] the VHO process occurs in, the fewer packets are lost, but the longer the VHO execution time is. If the packet loss rate is high, the quality of service cannot be maintained, while a long VHO execution time will disrupt the service. Both of them produce a negative effect to end users. Therefore, the esVHO must have an optimal mechanism that offers both low packet loss rate and short VHO execution duration. Besides, esVHO must be able to support various types of applications or services run on the network. The VHO executed by esVHO must be aware of the direction as well as the rate of the data traffic, which can be changed at any time.

6.1.2.2 Save energy for the wireless communication

Wi-Fi is an energy consuming technology, it consumes energy even in sleep state [36]. When it is in the active state, it consumes at least 15-20% of the total energy capacity of a laptop [62]. Therefore, when the IP-based services on the mobile device do not require high bandwidth, Wi-Fi becomes an energy inefficient communication means. Just for example, bandwidth-hungry services such as video streaming do not always require high bandwidth utilization [24]. In that case, a low-energy technology, i.e., Bluetooth, should be used instead of Wi-Fi. Additionally, Wi- Fi must be turned off and all the running sessions should be migrated to Bluetooth. By doing that, not only the energy is saved but also the continuity of the connection is maintained.

6.1.3 esVHO framework for non-interconnected environment

6.1.3.1 Vertical handover using local SDN architecture

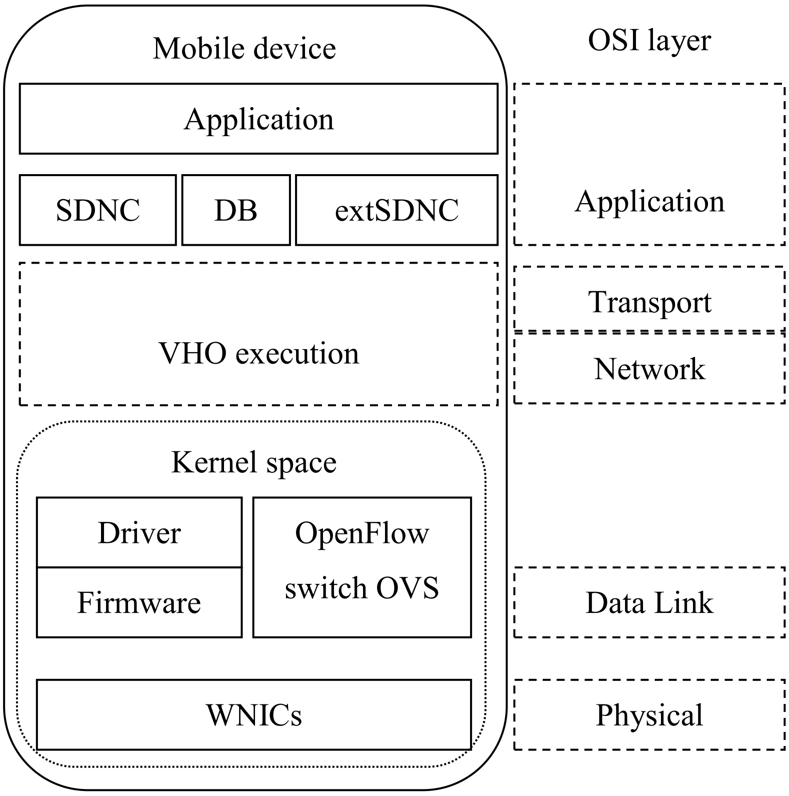


Fig. 6.2: esVHO framework on a mobile device

The esVHO framework is an improvement of extSDNC and is illustrated in Fig. 6.2. In esVHO, the local SDN architecture is utilized, however, the function of the extSDNC is enriched to address the requirements of an energy saving VHO described in section 3.2. The esVHO includes three phases of a traditional VHO [63], namely handover information gathering, handover decision and handover execution.

In the first phase, the esVHO gathers all necessary information, which is for making decisions as well as preparing for the VHO execution in the next phases. The gathered information includes, but not limited to, the data rate, the AP's properties, and the mobile device's state such as battery level, information of wireless interfaces. The collected information is stored in a local database named DB and is exchanged with the other devices. The devices exchange their local information right after their communication begins. After exchanging information, they do not either update the local database nor exchange collected information unless there is any new or modified information.

In the second phase, the decision when a handover should be triggered and which network should be selected will be made. In the esVHO framework, the VHO decision is simply based on the bandwidth utilization BU, which is defined as the total amount of traffic (from any directions) goes over the network interface in one second. The measured results show that the maximum BU of Bluetooth named $BU_{Bluetooth}$ is 1Mbps. Hence, if the measured BU is smaller than $BU_{Bluetooth}$ and Wi-Fi is being used, a VHO from Wi-Fi to Bluetooth will be executed. If Bluetooth is disabled, it is enabled and is associated with its WPAN network. This process is controlled by the Bluetooth firmware, which is located at the data link layer. When this process finishes, a link layer event is created. The mobile-controlled VHO introduced in Chapter 3 utilized this event to trigger the VHO process. Different from the mobile-controlled VHO, the VHO process is triggered after the Bluetooth's IP address assignment finishes. The Bluetooth connection is maintained as long as possible for saving energy. In this Chapter, we consider that Wi-Fi is used for high-speed services and Wi-Fi will be at full load when it is in the active state. The average energy consumed by Wi-Fi when it is at full load is a fixed value, thus, the higher energy consumed by Bluetooth, the smaller energy is saved. In other words, the energy saving is estimated at the minimum when Bluetooth is used with the highest bandwidth utilization. In that case, when BU exceeds $BU_{Bluetooth}$ and Bluetooth is being used, a VHO from Bluetooth to Wi-Fi will be executed. The VHO is also triggered after Wi-Fi is assigned an IP address.

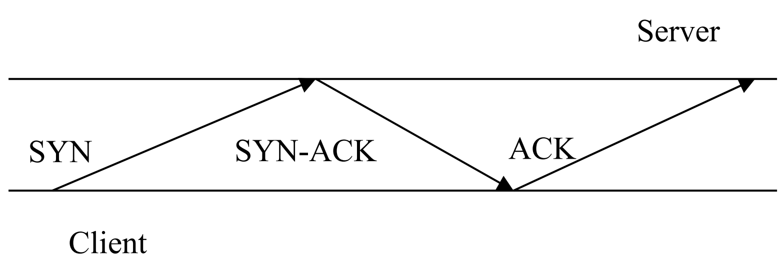


Fig. 6.3. VHO timing adjustment mechanism

In the last phase, the VHO between a source WNIC and a target WNIC is executed. Before the VHO process is initiated, both communicating devices first follow a VHO timing adjustment mechanism to exchange messages in order to tell the other and to confirm that they both are ready. The VHO timing adjustment mechanism is the second enhancement of esVHO in comparison to the extSDNC. The detail of the VHO timing adjustment mechanism is given in Fig. 6.3. As shown in the figure, one device plays a role of a client sending an SYN message to the other device, which acts as a server. When the server receives the SYN messages, it replies with an SYN-ACK message. When the client receives SYN-ACK it can start to instruct the OVS to install new OpenFlow rules. At the same time, the client sends back an ACK message to the server. The server performs OpenFlow rules installation after receiving the ACK message. At this moment, the network traffic still goes over the source WNIC. Once the new OpenFlow rules are installed, the network traffic will be directed to the target WNIC. Finally, the source WNIC is turned off for saving energy afterward.

6.1.3.2 Energy saving model

The energy saving mechanism of the esVHO framework is based on the reduction of the duration in which energy consuming technology, i.e., Wi-Fi, is being used. Specifically, all

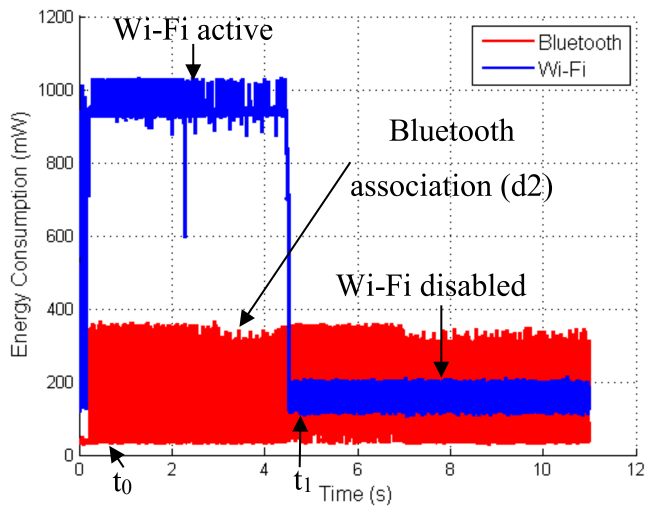


Fig. 6.4. Energy consumption transition
when a handover from Wi-Fi to Bluetooth is performed

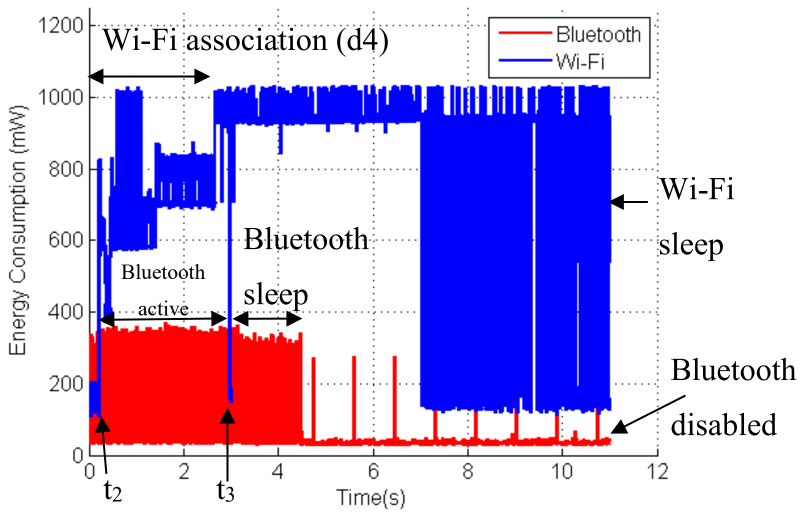


Fig. 6.5. Energy consumption transition
when a handover from Bluetooth to Wi-Fi is performed

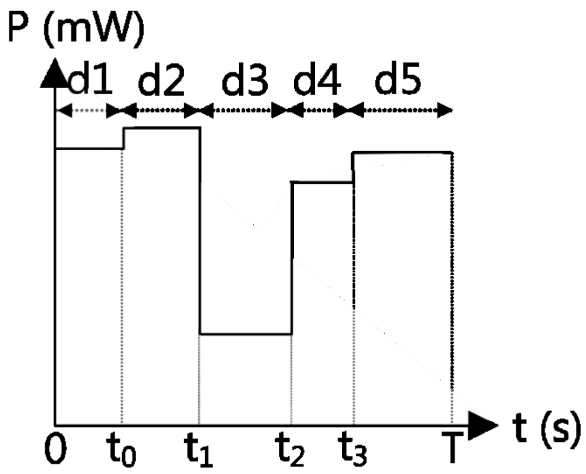


Fig. 6.6. Energy consumption model for esVHO

active sessions will be migrated from Wi-Fi to Bluetooth whenever the bandwidth utilization is lower than $BU_{Bluetooth}$. In addition, when Wi-Fi is used, its PSM mode is enabled and Bluetooth is turned off.

Figure 6.6 shows an energy saving model of esVHO. The model is formed based on the real measurement as given in Fig. 6.4 and Fig. 6.5. Figure 6.6 shows a scenario in which a service S initially uses Wi-Fi. At the time $t=t_0$, a VHO from Wi-Fi to Bluetooth for saving energy occurs. As shown in Fig. 6.4, Bluetooth is enabled at that time and is associated with its WPAN network. The Bluetooth association process takes $t_{VHO-WB}=(t_1-t_0)$ seconds to finish.

When this process finishes, Wi-Fi is disabled, and then, the service S uses only Bluetooth connection. At $t=t_2$, the system has to switch back to Wi-Fi, thus, Wi-Fi is woken up and is associated with its WLAN network. As shown in Fig. 6.5, the Wi-Fi association process takes place in duration d4. At $t=t_3$, Wi-Fi association process finishes, then, network traffic is navigated to Wi-Fi and Bluetooth is disabled.

Here, consumed power in each situation is expressed as shown in Table 6.1. The unit is mW.

TABLE 6.1 DEFINITION OF EACH PARAMETER WHICH SHOWS CONSUMED POWER

$\overline{P}_{WiFi-ac}$	Average power consumption by Wi-Fi when it is in active state
$\overline{P}_{WiFi-sl}$	Average power consumption by Wi-Fi when it is in sleep state
$\overline{P}_{WiFi-of}$	Average power consumption by Wi-Fi when it is in disabled state
$\overline{P}_{WiFi-as}$	Average power consumption by Wi-Fi when it is in association state
$\overline{P}_{Blue-ac}$	Average power consumption by Bluetooth when is in active state
$\overline{P}_{Blue-sl}$	Average power consumption by Bluetooth when it is in sleep state
$\overline{P}_{Blue-of}$	Average power consumption by Bluetooth when it is in disabled state
$\overline{P}_{Blue-as}$	Average power consumption by Bluetooth when it is in association state

Let t_{VHO-WB} and t_{VHO-BW} denote the time to complete a VHO from Wi-Fi to Bluetooth and vice versa.

In the duration d1, only Wi-Fi is used and Bluetooth is disabled. The total energy consumed by the wireless communication in duration d1, $\overline{E}_{wireless}(t)|_{0}^{t_0}$ (mJ), is expressed as follows:

$$\overline{E}_{wireless}(t)|_{0}^{t_0} = (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-of})t_0 \text{ (mJ)} \qquad (1)$$

The duration d2 lasts for $t_{VHO\text{-}WB}$=t_1-t_0 seconds and Bluetooth consumes $\overline{P}_{Blue-as}$ (mW) on average. In this duration, Wi-Fi is still being used and consumes $\overline{P}_{WiFi-ac}$ (mW). The total energy consumed by the wireless communication in the duration d2, $\overline{E}_{wireless}(t)|_{t_0}^{t_1}$ (mJ), is expressed as follows:

$$\overline{E}_{wireless}(t)|_{t_0}^{t_1} = (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-as})t_{VHO-WB} \qquad (2)$$

After Bluetooth association finishes, Wi-Fi is disabled. The total energy consumed by the wireless communication in the duration d3, $\overline{E}_{wireless}(t)|_{t_1}^{t_2}$, is expressed as follows:

$$\overline{E}_{wireless}(t)|_{t_1}^{t_2} = (\overline{P}_{WiFi-of} + \overline{P}_{Blue-ac})(t_2 - t_1) \qquad (3)$$

In the next duration d4, Wi-Fi is woken up and is associated with its WLAN network. This process lasts for $t_{VHO\text{-}BW}$=t_3-t_2 seconds and $\overline{P}_{WiFi-as}$ (mW) on average is consumed. The total energy consumed by the wireless communication in the duration d4, $\overline{E}_{wireless}(t)|_{t_2}^{t_3}$ (mJ), is expressed as follows:

$$\overline{E}_{wireless}(t)|_{t_2}^{t_3} = (\overline{P}_{WiFi-as} + \overline{P}_{Blue-ac})t_{VHO-BW} \qquad (4)$$

When Wi-Fi comes back to the active state, Bluetooth is turned off. The total energy consumed by the wireless communication in this duration d5, $\overline{E}_{wireless}(t)|_{t_3}^{T}$ is expressed as follows:

$$\overline{E}_{wireless}(t)|_{t_3}^{T} = (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-of})(T - t_3) \qquad (5)$$

Finally, the energy consumption model of esVHO, $\overline{E}_{esVHO}(t)|_{0}^{T}$, is calculated as follows:

$$\begin{aligned}\overline{E}_{esVHO}(t)\big|_0^T &= (\overline{P}_{WiFi-ac}+\overline{P}_{Blue-of})(T+t_0-t_3)\\ &+(\overline{P}_{WiFi-ac}+\overline{P}_{Blue-as})t_{VHO-WB}\\ &+(\overline{P}_{WiFi-of}+\overline{P}_{Blue-ac})(t_2-t_1)\\ &+(\overline{P}_{WiFi-as}+\overline{P}_{Blue-ac})t_{VHO-BW}\end{aligned} \quad (6)$$

The consumed energy with esVHO is different from the one with only Wi-Fi. The consumed energy with only Wi-Fi, $\overline{E}_{wireless}(t)\big|_0^T$, is calculated as follows:

$$\overline{E}_{wireless}(t)\big|_0^T = (\overline{P}_{WiFi-ac}+\overline{P}_{Blue-of})T \quad (7)$$

Let t_B denote the duration in which Bluetooth is used. By comparing equation (6) with equation (7), the amount of saved energy when using esVHO is obtained as follows:

$$\begin{aligned}\overline{E}_{esVHO-saved}(t)\big|_0^T &= (\overline{P}_{Blue-of}-\overline{P}_{Blue-as})t_{VHO-WB}\\ &+(\overline{P}_{WiFi-ac}-\overline{P}_{WiFi-as}+\overline{P}_{Blue-of}-\overline{P}_{Blue-ac})t_{VHO-BW}\\ &+(\overline{P}_{WiFi-ac}-\overline{P}_{WiFi-of}+\overline{P}_{Blue-of}-\overline{P}_{Blue-ac})t_B\end{aligned} \quad (8)$$

When the VHO happens N times, the amount of saved energy when using esVHO is calculated as follows:

$$\begin{aligned}\overline{E}^{n=N}{}_{esVHO-saved}(t)\big|_0^T &= N(\overline{P}_{Blue-of}-\overline{P}_{Blue-as})t_{VHO-WB}\\ &+N(\overline{P}_{WiFi-ac}-\overline{P}_{WiFi-as}+\overline{P}_{Blue-of}-\overline{P}_{Blue-ac})t_{VHO-BW}\\ &+(\overline{P}_{WiFi-ac}-\overline{P}_{WiFi-of}+\overline{P}_{Blue-of}-\overline{P}_{Blue-ac})t_B\end{aligned} \quad (9)$$

6.2 Performance evaluation

6.2.1 Experiment results

6.2.1.1 Experiment setup

The testbed used to measure the energy consumption of wireless communication is illustrated in Fig. 6.7. On the left of the figure, the mobile device (PC in Fig. 6.9) is a conventional computer (Core 2 Duo @2.26 GHz processor and 2GB RAM, Ubuntu 14.04 64-bit). The device connects to WLAN and WPAN networks using USB-based Wi-Fi and Bluetooth adapters, respectively. To reduce noise, each adapter is connected to a separated monitoring circuit

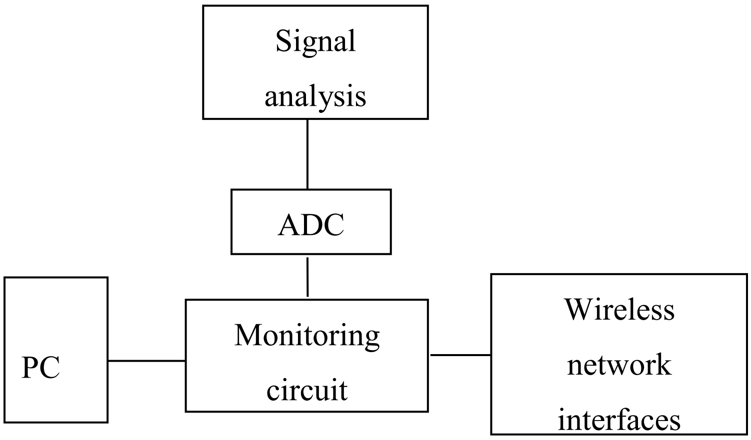


Fig. 6.7. Energy consumption measurement setup

before being attached to the mobile device. The monitoring circuit is to capture the instantaneous voltage $V(t)$ and current $I(t)$ observed on the adapters. The consumed instantaneous power $E(t)$ is calculated based on the Ohm's law: $E(t)=I(t)V(t)$. The analogue output signals of the monitoring circuits are sent to an analogue-to-digital converter (ADC) with a sampling rate of 2 kHz. The ADC is connected to a signal analysis device, which is also a conventional computer. In the signal analysis device, a small program is used to capture the digital signals from the ADC.

6.2.1.2 Measurement of energy consumption

In this section, the energy consumption in different states of wireless network interfaces, i.e., Wi-Fi and Bluetooth is presented. The states in the experiment include association, active, sleep, and disabled as shown in Fig. 6.4 and Fig. 6.5. Note that the state defined in our work is different from the state defined by 802.11 standards. Specifically, the association state is defined as the state in the duration, which starts when the interface is enabled and finishes when the interface has completed the association process. The active state is defined as the state when the interface is at full load and can transmit or receive the network traffic. The data rates when Wi-Fi and Bluetooth are at full load are measured as 65Mbps and 1Mbps, respectively. The sleep state in this work is defined as the state where the mobile device has connected to a wireless network, but no applications on the device are using the wireless connection. Finally, the disabled state is defined as the state when the network interface is disabled, but it is still attached to the mobile device.

The measured results of the average power consumption of each wireless interface in different states are shown in Table 6.2. As seen from the table, for completing the association process, Bluetooth consumes 161.71mW while Wi-Fi consumes more power, which is 4.5 times of Bluetooth. The largest difference between Wi-Fi and Bluetooth can be seen in the sleep state in which Wi-Fi consumes power more than eight times of Bluetooth. The power consumed by Wi-Fi in the sleep state is 3.6 times higher than the one consumed by Bluetooth in the active state. Besides, although Wi-Fi has been disabled, it still consumes 168.95mW, which is equal to 92% of the power consumed by Bluetooth in the active state.

TABLE 6.2 AVERAGE POWER CONSUMPTION ON WIRELESS NETWORK INTERFACES

Interface	Association (mW)	Active (mW)	Sleep (mW)	Disabled (mW)
Wi-Fi	726.85	949.77	673.91	168.95
Bluetooth	161.71	183.55	83.07	37.20

6.2.1.3 Measurement of total VHO execution time

Since unused WNICs are disabled for saving energy, it takes time to wake them up when a VHO occurs. This section is to measure that duration. Specifically, the total VHO execution time between a source WNIC and a target WNIC in this work is defined as the difference between when the target WNIC's association process finishes and when it is enabled as shown in Fig. 6.4 and Fig. 6.5. Figure 6.4 shows the duration in which Bluetooth is enabled and is associated with its WPAN network. While Fig. 6.5 shows the duration in which Wi-Fi is enabled and associated with its WLAN network. The total VHO execution times for esVHO, extSDNC VHO and Bluesaver VHO [21] were obtained from the measurements repeated 30 times using a real testbed. The results are given in Table 6.3. As shown in the table, the extSDNC VHO takes the longest time to execute the VHO in both directions. The Bluesaver VHO takes the shortest duration as all WNICs are always connected to their networks.

TABLE 6.3 AVERAGE VHO EXECUTION TIME

	esVHO (s)	extSDNC VHO (s)	Bluesaver VHO (s)
$t_{VHO\text{-}WB}$	3.47	4.57	0.38
$t_{VHO\text{-}BW}$	2.70	2.76	0.29

6.2.2 esVHO framework validation

In this section, experiments which use traffics generated by real applications and the results are discussed to validate the proposed energy saving model as well as the feasibility of the esVHO framework. Specifically, the total energy consumptions of the wireless communication when Wi-Fi is being used and when esVHO is being used are measured by using the testbed, which was described in section 4.1.1. The obtained results are then being compared with the estimated results using the energy saving model given in equation (7) and (9). The parameters used in the estimation are taken from Table 2 and Table 3, which were given in sections 4.1.2 and 4.1.3, respectively.

In the experiment, a 10-minute conversation traffic was recorded to a file, and then, the recorded file was streamed from one device (called “device A”) to another (called “device B”) by using the VLC tool [64]. The streamed audio was recorded at device B using Audio recorder tool [65]. Some different files were stored in the storage on the device B, which works as an FTP server as well, allowing the device A to download them anytime.

The experimental scenario was as follow: Two devices initially communicate using Bluetooth since the bandwidth of the streaming flow is as small as 34Kbps. During the streaming session, device A downloads different files from device B. Two applications (Audio streaming and FTP File transfer) generated UDP and TCP traffics, respectively. The generated data traffics are transferred in opposite directions, aiming to show that esVHO can support various protocols and the data traffic can be transferred in any direction. The total amount of traffic (from all directions) that goes over the network interface in one second is captured every second, which is called bandwidth utilization. When device A starts downloading the files, the bandwidth utilization increases. If the bandwidth utilization becomes larger than a threshold $BU_{Bluetooth}$, esVHO will invoke a VHO process to migrate the traffic from Bluetooth to Wi-Fi. When device A finishes the download process, the bandwidth utilization decreases. When the bandwidth utilization becomes lower than a threshold $BU_{Bluetooth}$, esVHO is triggered again to perform a handover from Wi-Fi to Bluetooth.

The recorded audio data used in the experiment were almost the same for both cases when two devices used only Wi-Fi and when they used the esVHO framework. The results of energy consumption in two cases are given in Fig. 6.8 and Fig. 6.9, respectively. The Fig. 6.9 shows that Wi-Fi was being used only when downloading the file. Also, the figure confirms that esVHO can perform a bidirectional VHO with various types of data traffic. The same procedure was repeated 20 times and the 95% confidence interval of the total energy consumption for the overall experiment is given in Fig. 6.10. The figure shows that the total energy consumption of the wireless communication when esVHO was used is 170.11J, which is less than 55% of the total energy consumed when only Wi-Fi was used. The figure also shows the total energy consumption of the wireless communication estimated by using the energy saving model. The

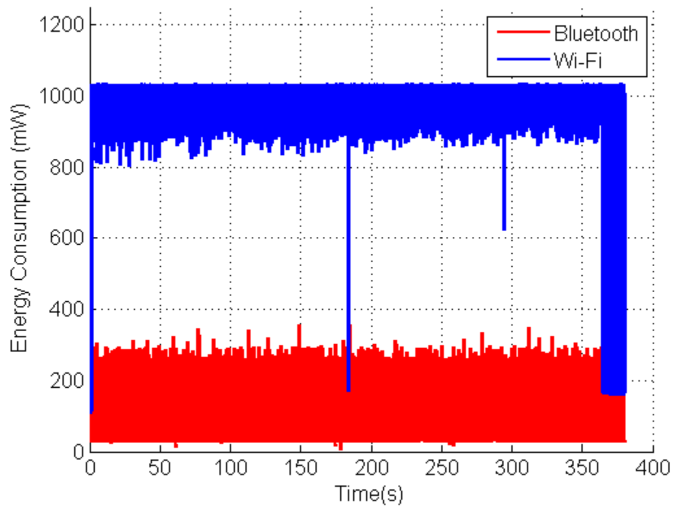


Fig. 6.8. Energy consumption when only Wi-Fi is used

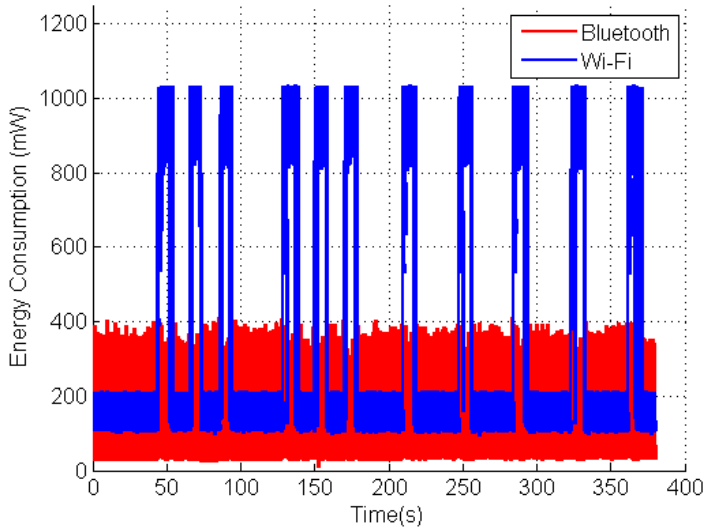


Fig. 6.9. Energy consumption when esVHO is being used

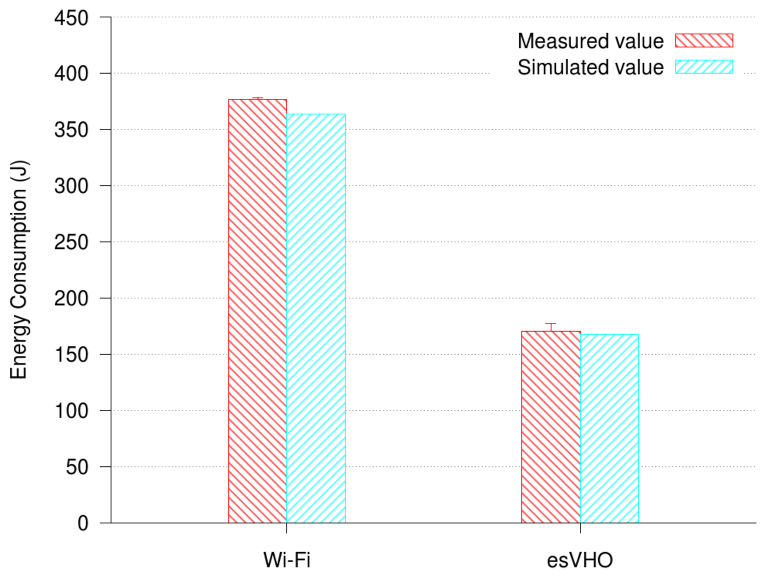


Fig. 6.10: Total energy consumption: measurement vs. simulation

values obtained by the simulation when only Wi-Fi is being used and when esVHO is used are respectively equal to 96.45% and 98.67% of the measured values. This means the energy consumption model can estimate the energy consumption of the system almost accurately.

6.2.3 Energy saving evaluation

By applying the measured values stated in the section 6.2.1.2 and 6.2.1.3 to the energy saving model of esVHO as given in the equation (8), the saved energy can be expressed in equation (10):

$$\overline{E}_{esVHO-saved}(t)\,|_0^T = (\overline{P}_{Blue-of} - \overline{P}_{Blue-as})t_{VHO-WB}$$
$$+ (\overline{P}_{WiFi-ac} - \overline{P}_{WiFi-as} + \overline{P}_{Blue-of} - \overline{P}_{Blue-ac})t_{VHO-BW}$$
$$+ (\overline{P}_{WiFi-ac} - \overline{P}_{WiFi-of} + \overline{P}_{Blue-of} - \overline{P}_{Blue-ac})t_B$$

$$= (37.2\text{-}161.71)3.47 + (949.77\text{-}726.85\text{+}37.2\text{-}183.55)2.7 + (949.77\text{-}168.95\text{+}37.2\text{-}183.55)\, t_B$$

$$= \text{-}225.31 + 634.47\, t_B \qquad (10)$$

Equation (10) shows that esVHO can save energy for the wireless communication if $\overline{E}_{Mobile-saved}(t)\,|_0^T \geq 0$ or $t_B \geq 0.36$s. When Bluetooth is used in the longest duration $t_B = T - t_{VHO-WB} - t_{VHO-BW}$, the highest amount of energy is saved. Similarly, when the VHO happens N times and Bluetooth is used in the longest duration $t_B = T - N(t_{VHO-WB} + t_{VHO-BW})$, the maximum amount of saved energy is

$$\overline{E}^{n=N}_{esVHO-saved-Max}(t)\,|_0^T = \text{-}4139.99N + 634.47T \quad (11)$$

In addition, the energy saving by using esVHO is compared to other methods. The first method uses Bluetooth for the communication in the whole duration T, or $t_B=T$, and Wi-Fi is disabled. This case can be considered as the ideal case for all methods which utilize Bluetooth for saving energy. The energy consumed by the wireless communication is

$$\overline{E}_{Blue-only}(t)\,|_0^T = (\overline{P}_{WiFi-of} + \overline{P}_{Blue-ac})T \qquad (12)$$

By comparing the equation (12) with the equation (7) and then applying the captured values in Table 6.2 and Table 6.3, the saved energy when only Bluetooth is used can be obtained as follows:

$$\overline{E}_{Blue-only-saved}(t)\,|_0^T = (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-of})T -$$
$$(\overline{P}_{WiFi-off} + \overline{P}_{Blue-ac})T$$

=(949.77+37.2-168.95-183.55)T=634.47T=634.47t_B (13)

In the second method, extSDNC VHO is used. The energy consumption model for extSDNC VHO is similar to esVHO's model and is calculated as follows:

$$\begin{aligned}\overline{E}_{extSDNCVHO}(t)\,|_0^T &= (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-of})(T + t_0 - t_3) \\ &+ (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-as})t'_{VHO-WB} \\ &+ (\overline{P}_{WiFi-of} + \overline{P}_{Blue-ac})(t_2 - t_1) \\ &+ (\overline{P}_{WiFi-as} + \overline{P}_{Blue-ac})t'_{VHO-BW}\end{aligned} \quad (14)$$

Where t'_{VHO-WB} and t'_{VHO-BW} are the total execution time of an extSDNC VHO from Wi-Fi to Bluetooth and vice versa. The saved energy when using extSDNC VHO is calculated as follows:

$$\begin{aligned}\overline{E}_{extSDNC-saved}(t)\,|_0^T &= (\overline{P}_{Blue-of} - \overline{P}_{Blue-as})t'_{VHO-WB} \\ &+ (\overline{P}_{WiFi-ac} - \overline{P}_{WiFi-as} + \overline{P}_{Blue-of} - \overline{P}_{Blue-ac})t'_{VHO-BW} \\ &+ (\overline{P}_{WiFi-ac} - \overline{P}_{WiFi-of} + \overline{P}_{Blue-of} - \overline{P}_{Blue-ac})t_B\end{aligned}$$

= (37.2-161.71)4.57 + (949.77-726.85+37.2-183.55)2.76 + (949.77-168.95+37.2-183.55) t_B

= -357.68 + 634.47 t_B (15)

Equation (15) shows that extSDNC VHO saves energy less than esVHO does. The different is 132.37mJ. Also, the extSDNC VHO can save energy for the mobile device when $\overline{P}_{extSDNC-saved}(t)\,|_0^T$, or $t_B \geq 0.56$s. When the VHO happens N times and $t_B=T-N(t'_{VHO-WB}+t'_{VHO-BW})$, the maximum amount of saved energy is

$$\overline{E}^{n=N}_{extSDNCVHO-saved-Max}(t)\,|_0^T = -4875.98N + 634.47T \quad (16)$$

The third method uses a mechanism proposed in [24], so-called Bluesaver VHO. To make a fair comparison, the same environment is used. Specifically, to take the control of the incoming and outgoing network traffic via the WNICs, the local SDN architecture is used instead of a modification in WNIC driver. The devices are able to communicate with both Wi-Fi and Bluetooth connected to their networks without any external support or modified AP. The WNIC selection is based on the OpenFlow rules installed on the OVS.

The energy consumption model of Bluesaver VHO is illustrated in Fig. 6.11. As shown in the figure, in the duration d"1 and d"3, Wi-Fi is used and Bluetooth is sleeping. In duration d"2, Bluetooth is used and Wi-Fi is sleeping. The total energy consumed for the wireless communication is expressed as follows:

$$\overline{E}_{BluesaverVHO}(t)\,|_0^T = (\overline{P}_{WiFi-ac} + \overline{P}_{Blue-sl})T + (\overline{P}_{Blue-ac} - \overline{P}_{Blue-sl} + \overline{P}_{WiFi-sl} - \overline{P}_{WiFi-ac`})(t''_1 - t''_0) \quad (17)$$

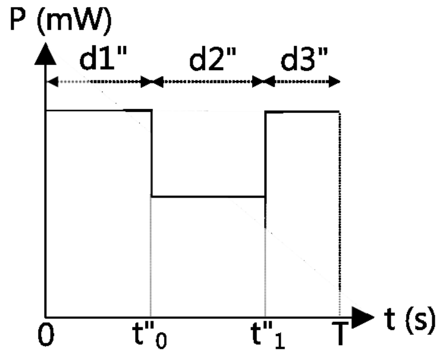


Fig. 6.11. Energy consumption model for Bluesaver VHO

Similarly, the saved energy when using Bluesaver VHO is:

$$\overline{E}_{BluesaverVHO-saved}(t)\,|_0^T = (\overline{P}_{Blue-of} - \overline{P}_{Blue-sl})T - (\overline{P}_{Blue-ac} - \overline{P}_{Blue-sl} + \overline{P}_{WiFi-sl} - \overline{P}_{WiFi-ac`})t_B$$

$$= (37.2-83.07)T - (183.55-83.07+673.91-949.77)t_B$$

$$= -45.87T+175.38\, t_B \quad (18)$$

Equation (18) shows that Bluesaver VHO can save energy when Bluetooth is used in $t_B \geq 0.26T$ (seconds). Let t'_{VHO-WB} and t'_{VHO-BW} denote the total execution time of an extSDNC VHO from

Wi-Fi to Bluetooth and vice versa. When the VHO happens N times and Bluetooth is used in the longest duration, or $t_B = T - N(t''_{VHO\text{-}WB} + t''_{VHO\text{-}BW})$, the maximum amount of saved energy is

$$\overline{E}^{n=N}_{BluesaverVHO-saved-Max}(t)\,|_0^T = -117.5N + 3885.3T \quad (19)$$

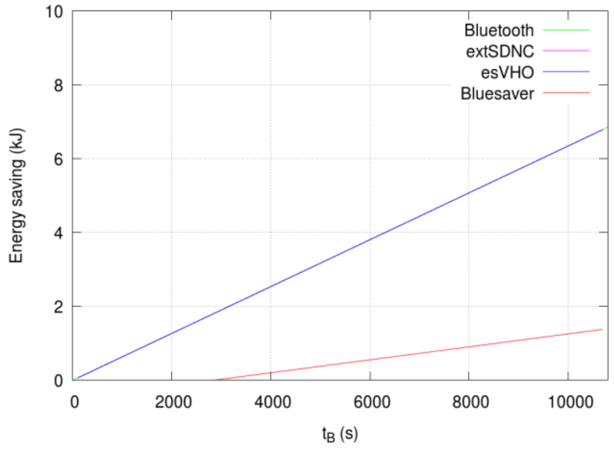


Fig. 6.12. Energy saving measurement

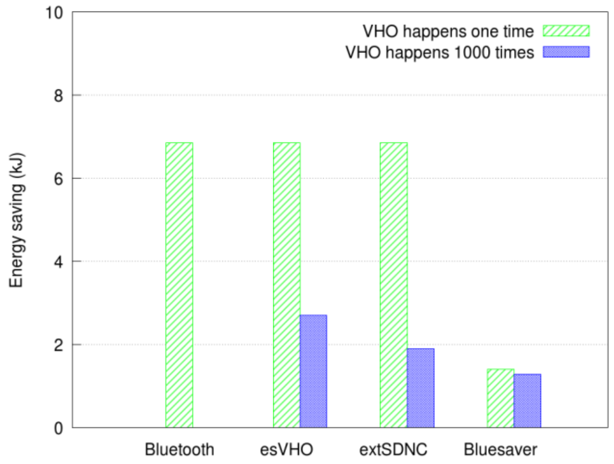


Fig. 6.13. Maximum energy saved when VHO happens frequently

Figure 6.12 summarizes the amount of energy saved by using four methods. The figure confirms that the longer Bluetooth is used, the more energy is saved. The figure also points out that esVHO and extSDNC VHO outperforms Bluesaver VHO in term of energy saving. The energy saved by using either esVHO or extSDNC VHO is close to the ideal case where Bluetooth is used instead of Wi-Fi.

Figure 6.13 shows the maximum saved energy when the VHO happens in a duration T=3h. As shown in the figure, when the VHO happens once, esVHO and extSDNC VHO save 6.85kJ while Bluesaver VHO saves 1.4kJ. When the VHO happens 1000 times, the amount of saved energy is reduced. However, esVHO still saves 2.7kJ, which is 1.42 times higher than extSDNC VHO does and 2.11 times higher than Bluesaver VHO does.

6.2.4 Vertical handover evaluation

In this section, the performance of the VHO in the esVHO framework is evaluated by comparing with other two VHO mechanisms, namely, extSDNC VHO and Bluesaver VHO. Since Bluesaver VHO does not have the VHO timing adjustment mechanism, the connection between two communicating devices will be dropped when a VHO occurs at any device. In the best case where both communicating devices perform a VHO to the same network at almost the same time, the connection can be maintained. The performance of Bluesaver VHO in the best case is assumed to be achieved by executing Bluesaver VHO manually on both devices at almost the same time. The evaluation metric is the number of packet loss since it is a critical factor that affects the quality of the services.

The testbed in this evaluation consists of two mobile devices as illustrated in Fig. 6.7. The devices use 802.11n to join the WLAN network. The standard 802.11n operates at 5GHz to avoid the interference with Bluetooth. The devices are also able to join a WPAN network using Bluetooth 4.0 technology.

In the evaluation, UDP traffic between two devices is generated using Iperf tool [41]. The measurement procedure is as follows: on one device, Iperf tool runs in server mode to receive traffic from Iperf clients. On the other device, Iperf runs in client mode and sends UDP traffic to the Iperf server. While UDP traffic is being sent, a VHO occurs randomly. The procedure was repeated with different data rate varied from 100Kbps to 500 Kbps. Each scenario was repeated 30 times.

The same procedure is applied to three types of VHOs from Wi-Fi to Bluetooth and vice versa. The obtained results are given in Fig. 6.14 and Fig. 6.15. As shown in the figures, for each process, the average and 95% confidence interval of the packet loss rate were measured. The packet loss rate in Fig. 6.15 is smaller than the one in Fig. 6.14 because the traffic was switched from a low data rate network to a higher one. As depicted in the figures, esVHO has the smaller number of packet loss than extSDNC VHO in any cases since the VHO timing adjustment mechanism reduced the time difference between the two devices to finish the VHO process. The number of packet loss of esVHO is smaller than Bluesaver VHO when a handover from Bluetooth to Wi-Fi was performed. To compare fairly the performance of Bluesaver VHO with the others, a probability coefficient α ($0 \leq \alpha \leq 1$) is added to the captured number of packet loss. Let $L_{measured}$ be the measured number of packet loss of Bluesaver VHO in the best case at a fixed data rate. In addition, let N_{Total} be the total number of sent packets, then the average number of packet loss of Bluesaver VHO will be estimated as follows:

$$L_{Bluesaver} = L_{measured} + (N_{Total} - L_{measured})\alpha \quad (20)$$

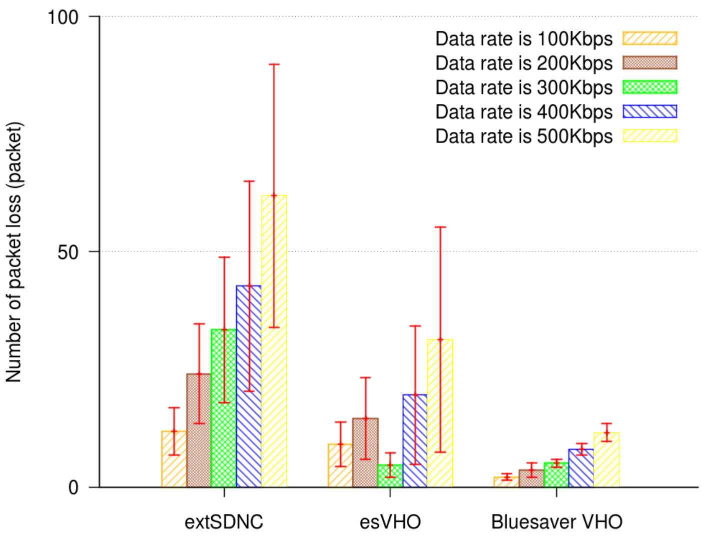


Fig. 6.14. Handover from Wi-Fi to Bluetooth

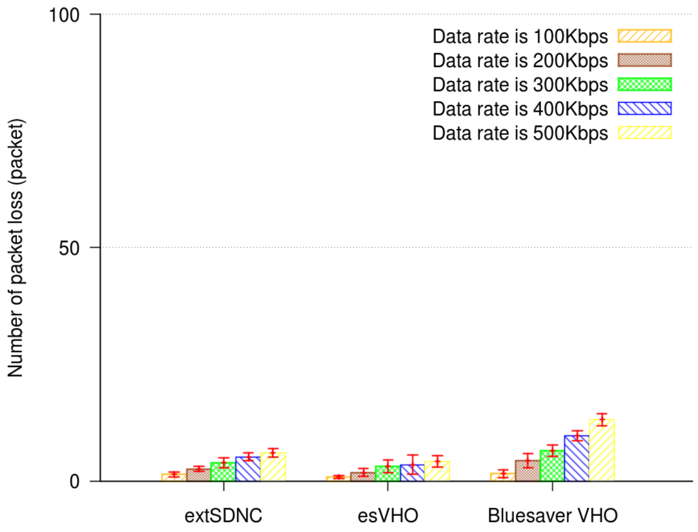


Fig. 6.15. Handover from Bluetooth to Wi-Fi

At the same data rate, let L_{esVHO} and $L_{extSDNC}$ denote the average number of packet loss of esVHO and extSDNC VHO, respectively. The average number of packet loss of Bluesaver VHO is smaller than that of esVHO when $L_{Bluesaver} < L_{esVHO}$. This means $\alpha < \frac{L_{esVHO} - L_{measured}}{N_{Total} - L_{measured}}$. Similarly, $\alpha < \frac{L_{extSDNC} - L_{measured}}{N_{Total} - L_{measured}}$ means Bluesaver VHO has the smaller number of packet loss compared to extSDNC VHO. Applying the measured number of packet loss at different data rates to above equations, we obtained the probability that Bluesaver VHO has the smaller number of packet loss than the others as shown in Fig. 6.16. The figure shows that the highest probabilities for Bluesaver VHO to have less packet loss than extSDNC VHO and esVHO are 12.16% and 6.19%, respectively. Consequently, esVHO has the best performance among three approaches.

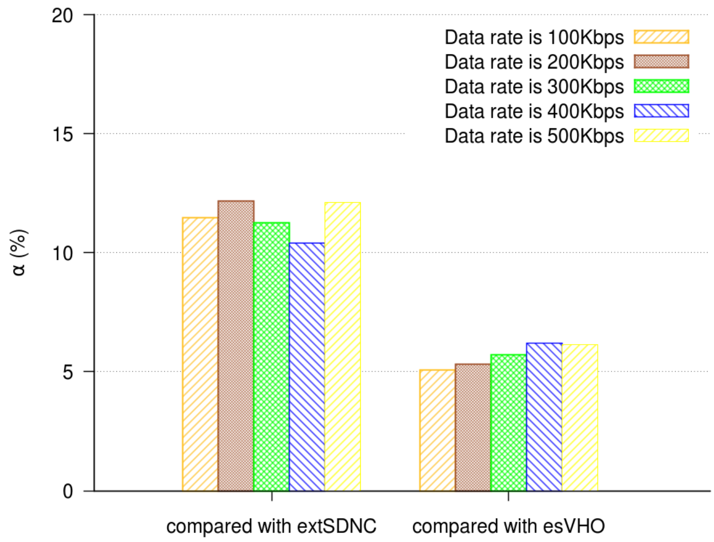


Fig. 6.16. Probability for Bluesaver VHO to have less packet loss than the others

As can be seen from Fig. 6.14 and Fig. 6.15, the number of packet loss when handover from Wi-Fi to Bluetooth is higher than when handover in the inverted direction. Figure 6.14 shows that the highest average packet loss of esVHO, when the data rate is 500Kbps, is 31.33 packets. The packet size used in the experiment was 1470 bytes, hence, the total amount of lost data is 46055 bytes (368440 bits). As shown in Table 6.1, Wi-Fi consumes 949.77mJ in the active state. As described in section 6.2.1.2, Wi-Fi is at full load, or at the highest data rate (65Mbits), when it is in the active state. This means each bit requires Wi-Fi to spend $\frac{949.77}{68157440}$ (mJ). The energy wasted on the packet loss is $368440\frac{949.77}{68157440} = 5.13$ mJ. Similarly, the highest average packet loss of esVHO, when the data rate is 500Kbps, is 4.17 packets. The amount of lost data is 33.36 bytes (49039 bits). Table 6.1 shows that Bluetooth consumes 183.55mJ in the active state. As described in section 6.2.1.2 in the revised manuscript, the data rate of Bluetooth in the active state is 1Mbits. Therefore, the energy wasted on the packet loss is $49039\frac{183.55}{1048576}$=8.58 mJ. When a bidirectional VHO occurs, the total energy wasted on the packet loss is $5.13+8.58=13.71$ mJ. This is only 1.44% of the energy that Wi-Fi consumes every second in the active state. Therefore, we have decided to ignore the influence of packet loss on the energy consumption in this work.

6.3 Conclusion

In this Chapter, an esVHO framework has been introduced to reduce the number of packet loss caused by VHO and to save energy. Experiments confirmed that when esVHO is being used, the number of packet loss is lower than extSDNC. The results of direct measurements of energy consumption proved that the proposed esVHO outperforms the other existing approaches even if the VHO occurs frequently. In the current work, the esVHO focuses on saving energy for the wireless communication of mobile devices. The energy saving for the mobile devices including the energy consumed by CPU, screen, and the application will be investigated as a future study.

Chapter 7 Discussion

This chapter discusses the advantages as well as the remaining issues of the proposed solutions through the results which this study has achieved.

As mentioned in this dissertation, saving energy for the wireless communication plays an important role in preventing the increase of CO2 emission, which has been identified as one of the major threats to the environmental conservation, and in reducing the cost of operation and maintenance. With a fixed infrastructure wireless network, the existing energy saving solutions are mainly based on the sleeping duration of the wireless technology. These approaches, however, do not work well when various types of data traffic are delivered continuously at different rates in the network. Therefore, it motivates this study to find a novel energy saving mechanism.

This work has explored the fact that application and services including bandwidth-hungry ones like video streaming do not always require high bandwidth. At the same time, each mobile device already has multiple built-in wireless technologies for providing ubiquitous access. Wireless technologies that offer high data rate are generally energy-consuming ones while low-energy technologies commonly provide low data rate. This dissertation utilizes the heterogeneous access of the mobile devices to save energy by using a low power consumption wireless access technology instead of the energy-consuming one whenever the bandwidth requirement is low enough. To achieve this goal, the continuity of the connectivity as well as the quality of all running application and services must be maintained, or a seamless VHO is required. Since the mobile devices may be composed of hardware provided by different vendors and they can communicate directly or via an infrastructure device, i.e., AP in WLAN, Master node in Bluetooth piconet, several issues need to be resolved to meet the above requirements:

- To control the traffic that comes in and goes out of a mobile device from the inside without depending on any particular vendor

- To perform seamless VHOs on two communicating devices without any external support
- To decide whether or not to trigger a VHO based on collected information
- To save energy for the wireless communication no matter how many VHO occur

The above issues stimulate the introduction of a novel framework called esVHO. The framework consists of an energy efficient mobile-controlled VHO and an energy saving model.

The contributions of this dissertation can be discussed concretely as follows:

7.1 Energy efficient mobile-controlled VHO

The proposed esVHO framework saves energy for the wireless communication by performing VHOs between energy-consuming and low power consumption wireless access technologies in non-interconnected environments. The key difficulties and the corresponding solutions in three phases of a VHO process are summarized as follows:

7.1.1 VHO Information gathering

The VHO information is commonly gathered by a centralized control like Media Independent Information Service in MIH, Home Agent in Mobile IP and Mobile Ipv6. However, in non-interconnected scenarios, the VHO information can be gathered and exchanged only by the mobile devices.

- **Controlling local incoming/ outgoing traffic:** To control local traffic from the inside, many attempts modified the WNIC driver. Not to rely on a particular vendor, SDN architecture is applied locally in a mobile device. With the embedded local SDN controller, the traffic is navigated to a corresponding WNIC via an OpenFlow switch following OpenFlow rules stored in an OpenFlow table. However, to correctly navigate local traffic, the local SDN-controller needs network-wide information such as network topology. Without a global SDN-controller, the local SDN-controllers are unable to instruct the OpenFlow switch to navigate in and out traffic. To this end, a controller-to-controller (C2C) communication means was proposed to allow extended controllers on mobile devices to exchange local information. The feasibility of the C2C communication means has been confirmed by experiment results on a real testbed. However, the scalability of this has not as yet been considered. Besides, the security for C2C has not been addressed yet. It is necessary to encode or encrypt the transferred data and secure the connection.

- **Making the heterogeneity on the access to be transparent to application and services:** Not to let the applications to detect the change of physical interface, the IP and MAC addresses must be unchanged from their viewpoints. Also, ARP messages generated whenever the physical interface changed must be handled properly. The initial version of OpenFlow specification does not allow the ARP payload to be rewritten, thus, existing approaches that deploy local SDN architecture inside a mobile device have to rely on a global SDN controller. To address this issue, an extended SDN-controller was proposed. This proposal is one of the key novelties of the proposed esVHO framework.

7.1.2 VHO decision

The VHO decision in this dissertation is to determine whether or not a VHO is triggered. Since this study is to save energy by conducting VHOs, the key criterion to make VHO decision is the bandwidth requirement. To this end, communicating devices must be able to measure the throughput by themselves. The solution of this issue was mentioned in Chapter 4. The obtained results show that if Bluetooth is used in 10mins, the system can save 44.7% of energy for the wireless communication. When the communication between two devices requires the support from intermediate devices, the energy consumed for relaying messages must be reduced. This issue was addressed in Chapter 5. Instead of monitoring the throughput, the intermediate devices receive the bandwidth requirement from the communicating devices, and then, decide to perform a VHO to save energy. In both cases, the energy saving solution for the wireless communication has been confirmed.

In this work, the destination network was defined when the devices start communicating. However, the best destination network to be selected can be changed due to many aspects such as device location, the distance between communicating devices. The future VHO should take into account this issue. Additionally, the mobility of the devices has not been considered. When the devices move, the distance between them changes and the connection can be dropped. The mobility pattern of the devices can generate a ping pong effect making the VHO decision confusing. In these cases, the extSDNC should be able to estimate when the connection will be lost to perform a make-before-break VHO. Otherwise, a break-before-make VHO will occur. In fact, the extSDNC is already able to perform a break-before-make VHO because the destination network has already been defined in the local database. Obviously, the destination network may be unreachable due to the mobility of the devices, hence, a list of potential destination networks should be made for backing up these cases.

7.1.3 VHO execution

The VHO affects the delay generated by the VHO process due to the layer of the OSI model where it is executed. A link layer VHO, which is controlled by the firmware placed in the link layer, occurs before any upper-layer VHOs. A network layer VHO operates using IP traffic. One of the network layer VHO solutions is Mobile IP, which forwards packets to mobile devices that are away from their home networks using tunnels. A proxy-based VHO and SIP are examples of transport layer VHO and application layer VHO, respectively. These VHOs do not require any changes to the TCP/IP protocol stack. SIP uses signaling messages to delay ongoing traffic, thus, there is no packet loss. VHOs at the network layer and above generally have serious handover delay due to the network overhead. Therefore, the link layer information is recently used to have a soft VHO, reducing the handover delay.

The link layer trigger was utilized in the VHO execution of extSDNC. However, the VHO delay, as well as the number of packet loss, was as high as 37.2% as given in Chapter 2. To reduce the number of packet loss, in the esVHO framework, VHO is executed after the assignment of IP address finished. It is clear that there is a duration in which both wireless access technologies are running. However, the duration is small and does not affect the energy saving purpose. Besides, the benefit of this approach is to let the C2C communication channel exchange messages on the high-speed connection, thus, the VHO delay can be reduced. Note that the communicating devices have to perform a VHO to the same destination network and at almost the same time to maintain the connection and reduce the packet loss. For this purpose, a VHO timing adjustment mechanism was introduced. Experiment results in Chapter 6 have confirmed that esVHO has the smallest number of packet loss compared to other works.

7.2 Energy saving model

Since this work evaluates all proposals using direct measurement on a real testbed, it is necessary to verify the effectiveness in various cases, thus, energy saving model for esVHO framework was proposed. The model was validated under realistic traffic as described in Chapter 6. The validation confirms that the estimated energy consumption is close to the measurement values. Also, it is used to compare the effectiveness of esVHO with the others. The simulated values show that esVHO outperforms the other approaches and the amount of energy saved by using esVHO is close to the ideal case where Bluetooth is utilized as a replacement of Wi-Fi for saving energy. Note that the model is formed by using values from direct measurement, thus, the obtained results are bounded by the specification of the devices used in the measurement. When applying this approach to another experiment setting, the amount of energy that can be saved may be different. However, the system is still able to save energy.

Chapter 8 Conclusion and Future work

The insufficient energy resource of mobile devices requires an energy efficient wireless communication. While many existing approaches tried to improve the energy efficiency of energy-consuming wireless technologies like Wi-Fi, this work focuses on saving energy in scenarios where different types of data are continuously transferred with dynamically changing data rate.

The dissertation firstly describes how to control incoming and outgoing traffic of any wireless network interface by using software, i.e., an SDN-controller. Although the traditional-SDNC was deployed locally and lacked the global view of the network, it has been supported by the extSDNC to operate properly. By using extSDNC, various types of data traffic from any direction can be handled with ease and flexibility. The deployment of extSDNC does not require any modification of the hardware or device driver. This means extSDNC is an implementation independent.

Besides, extSDNC can be used to provide a mobile-controlled vertical handover management scheme. The VHO on both communicating devices can be executed without any external support. With this mechanism, two applications can be seen are device-to-device heterogeneous wireless communication for victims in disaster scenarios and energy efficient wireless multi-hop network. In disaster scenarios, mobile devices can communicate without concerning which technology they are using. Additionally, a VHO trigger algorithm was introduced to prolong the communication time. In a wireless multi-hop network, the intermediate nodes equipped extSDNC consume less energy for relaying messages. While the energy consumption of wireless communication between two communicating nodes was reduced by selecting the appropriate interface for sending and receiving messages.

This dissertation also explains how the esVHO framework utilizes extSDNC to collect network information and exchanges collected information among mobile devices to obtain an energy efficient wireless communication while maintaining all active sessions. The framework has been validated under realistic traffic. Moreover, experimental and simulated results have confirmed that esVHO can save energy in any case even when multiple VHO incurs. The amount of saving energy is mainly depended on the duration in which Bluetooth is used. The number of packet loss caused by the VHO process is as small as the quality of the received speech is almost the same as the sent one.

In summary, this dissertation introduced an energy-efficient heterogeneous wireless communication with extended SDN-controller. The energy was saved by performing VHO between the Wi-Fi and Bluetooth interface. It may be also interesting to investigate opportunities of saving energy for other technologies such as LTE or WiMAX. This is feasible since VHO between Wi-Fi and LTE has already introduced in several existing works [72-74]. Similarly, VHO between WiMAX and Wi-Fi has also been studied [75-78]. Besides, the effectiveness of the proposed system in improving the operating time of smartphones such as mobile phones or tablet devices will be studied. In addition, the mobility of the devices will be considered.

References

[1] Kemp, S. “Digital In 2017: Global Overview”. Avaiable at: www.wearesocial.com/uk/blog/2017/digital-in-2017-global-overview. Viewed: Febreury 1, 2017. (2017)

[2] Qualcomm. “The 1000x data challenge,”. Tech. Rep. [Online]. Available: http://www.qualcomm.com/1000x

[3] Mingay, Simon. "Green IT: the new industry shock wave." Gartner RAS Research Note G 153703 (2007): 2007.

[4] Joung, Jingon, Chin Keong Ho, Koichi Adachi, and Sumei Sun. "A survey on power-amplifier-centric techniques for spectrum-and energy-efficient wireless communications." IEEE Communications Surveys & Tutorials 17, no. 1 (2015): 315-333.

[5] Mahmood, Farhad, Erik Perrins, and Lingjia Liu. "Modeling and analysis of energy consumption for RF transceivers in wireless cellular systems." In Global Communications Conference (GLOBECOM), 2015 IEEE, pp. 1-6. IEEE, 2015.

[6] Han, Shuangfeng, I. Chih-Lin, Zhikun Xu, and Corbett Rowell. "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G." IEEE Communications Magazine 53, no. 1 (2015): 186-194.

[7] Han, Shuangfeng, Chih Lin, Corbett Rowell, Zhikun Xu, Sen Wang, and Zhengang Pan. "Large scale antenna system with hybrid digital and analog beamforming structure." In Communications Workshops (ICC), 2014 IEEE International Conference on, pp. 842-847. IEEE, 2014.

[8] Dai, Linglong, Xinyu Gao, and Zhaocheng Wang. "Energy-efficient hybrid precoding based on successive interference cancelation for millimeter-wave massive MIMO systems." In Radio and Antenna Days of the Indian Ocean (RADIO), 2015, pp. 1-2. IEEE, 2015.

[9] Soh, Yong Sheng, Tony QS Quek, Marios Kountouris, and Hyundong Shin. "Energy efficient heterogeneous cellular networks." IEEE Journal on Selected Areas in

Communications 31, no. 5 (2013): 840-850.

[10] Björnson, Emil, Luca Sanguinetti, Jakob Hoydis, and Mérouane Debbah. "Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?." IEEE Transactions on Wireless Communications 14, no. 6 (2015): 3059-3075.

[11] Sinaie, Mahnaz, Alessio Zappone, Eduard A. Jorswieck, and Paeiz Azmi. "A novel power consumption model for effective energy efficiency in wireless networks." IEEE Wireless Communications Letters 5, no. 2 (2016): 152-155.

[12] Bacci, Giacomo, E. Veronica Belmega, Panayotis Mertikopoulos, and Luca Sanguinetti. "Energy-aware competitive power allocation for heterogeneous networks under QoS constraints." IEEE Transactions on Wireless Communications 14, no. 9 (2015): 4728-4742.

[13] She, Changyang, Chenyang Yang, and Lingjia Liu. "Energy-efficient resource allocation for MIMO-OFDM systems serving random sources with statistical QoS requirement." IEEE Transactions on Communications 63, no. 11 (2015): 4125-4141.

[14] Wang, Yan, Wenjun Xu, Kewen Yang, and Jiaru Lin. "Optimal energy-efficient power allocation for OFDM-based cognitive radio networks." IEEE Communications Letters 16, no. 9 (2012): 1420-1423.

[15] Lv, Jing, Alessio Zappone, and Eduard A. Jorswieck. "Energy-efficient MIMO underlay spectrum sharing with rate splitting." In Signal Processing Advances in Wireless Communications (SPAWC), 2014 IEEE 15th International Workshop on, pp. 174-178. IEEE, 2014.

[16] IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Aug. 1999.

[17] Takeuchi, Shojiro, Kaoru Sezaki, and Yasuhiko Yasuda, "An improved power saving mechanism for mac protocol in ad hoc networks," IEICE transactions on Communications, Vol.E88-B, No.7, 2005, pp.2985-2993.

[18] Qiao, Daji, and Kang G. Shin, "Smart power-saving mode for IEEE 802.11 wireless LANs," Proceedings of IEEE Conference on Computer Communications (INFOCOM2005), Vol.3, pp.1573-1583, 2005.

[19] Hiraguri, Takefumi, Masakatsu Ogawa, Makoto Umeuchi, and Tetsu Sakata, "Power saving scheme suitable for wireless LAN in multimedia communications," IEICE Transactions on

Fundamentals of Electronics, Communications and Computer Sciences, Vol.E92-A, No.9, 2009, pp.2184-2190.

[20] J. A. Gutierrez et al., “IEEE 802.15.4: A Developing Standard for Low-Power, Low-Cost Wireless Personal Area Networks,” IEEE Network, Vol.15, No.5, Sept./Oct. 2001, pp.12–19.

[21] Taeshik, S. H. O. N., Eui-Jik Kim, I. N. Jeongsik, and P. A. R. K. Yongsuk, "Design and implementation of hybrid MAC-based robust architecture for wireless sensor network," IEICE Transactions on Communications, Vol.E93-B, No.4, 2010, pp.1016-1019.

[22] SangKwon, M. O. O. N., Y. O. O. Jong-Woon, K. I. M. Jaesub, and P. A. R. K. Kyu-Ho, "MAC protocol for energy efficiency and service differentiation with high goodput in wireless sensor networks," IEICE Transactions on Communications, Vol.E96-B, No.6, 2013, pp.1444-1458.

[23] Bluetooth Specification, www.bluetooth.com/developer/specification

[24] Andrew Pyles, David T. Nguyen, Xin Qi, and Gang Zhou, ”Bluesaver: AMulti-PHY Approach to Smartphone Energy Savings”, IEEE Transactions on Wireless Communications, 14(6):3367–3377, 6 2015.

[25] G. Lampropoulos, A. Kaloxylos, N. Passas, and L. Merakos, “A Power Consumption Analysis of Tight-Coupled WLAN/UMTS Networks,” presented at the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007, 2007, pp. 1–5.

[26] Yoon Young An et al. Reduction of handover latency using MIH services in mipv6. In 20th International Conference on Advanced Information Networking and Applications, pp 229–234, Apr 2006.

[27] Perkins, C. Ip mobility support for IPv4. RFC Editor, 2002.

[28] Charles E. Perkins and David B. Johnson. Mobility support in IPv6. In Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking, pages 27–37, 1996.

[29] J. Rosenberg et al. Sip: Session Initiation Protocol. In RFC3344. IETF, 2002.

[30] H. Zimmermann. “OSI Reference Model - The IS0 Model of Architecture for Open Systems Interconnection.” In IEEE Transactions on Communications COM-28, No. 4: April 1980.

[31] Nilanjan Banerjee, WeiWu, Kalyan Basu, and Sajal K. Das, “Analysis of sip-based mobility manage-ment in 4g wireless networks”, Comput. Commun, 27(8):697–707, May 2004.

[32] Jasone Astorga, Marina Aguado, Nerea Toledo, and Marivi Higuero, “A high performance link layer mobility management strategy for professional private broadband networks”, J. Network and Computer Applications, 36(4):1152–1163, 2013.

[33] Seo, SungHoon, and JooSeok Song, "Energy-Efficient Vertical Handover Mechanism," IEICE Transactions on Communications, Vol.E92-B, No.9, 2009, pp.2964-2966.

[34] Deng, Shuo, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. "Wifi, lte, or both?: Measuring multi-homed wireless internet performance." In Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 181-194. ACM, 2014.

[35] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair, “Wake on wireless: an event driven energy saving strategy for battery operated devices”, In Proceedings of the 8th Annual International Conference on Mobile computing and Networking (MobiCom '02). ACM, New York, NY, USA, 160-171.

[36] Feeney, Laura Marie, and Martin Nilsson. "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment." In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, pp. 1548-1557. IEEE, 2001.

[37] Smith, Gregory, Anmol Chaturvedi, Arunesh Mishra, and Suman Banerjee. "Wireless virtualization on commodity 802.11 hardware." In Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization, pp. 75-82. ACM, 2007.

[38] Nicholson, Anthony J., Scott Wolchok, and Brian D. Noble. "Juggler: Virtual networks for fun and profit." IEEE Transactions on Mobile Computing 9, no. 1 (2010): 31-43.

[39] 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 Comput. Commun. Rev., Computer Communication Review 38, No. (2) March (2008): 69-74.

[40] J. Lee et al, Lee, Jeongkeun, Mostafa Uddin, Jean Tourrilhes, Souvik Sen, Sujata Banerjee, Manfred Arndt, Kyu-Han Kim, and Tamer Nadeem. "meSDN: mobile extension of SDN." In Proc. of the 5th Int. In Proceedings of the Fifth International Workshop on Mobile Cloud Computing and Services. MCS'14, New York, NY, USA, pp. 7-14, 2014.

[41] Yap, Kok-Kiong, Te-Yuan Huang, Masayoshi Kobayashi, Yiannis Yiakoumis, Nick McKeown, Sachin Katti, and Guru Parulkar. "Making use of all the networks around us: a case study in android." In Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, pp. 19-24. ACM, 2012.

[42] Nguyen-Duc, Toan, and Eiji Kamioka. "An extended SDN controller for handover in heterogeneous wireless network." In Communications (APCC), 2015 21st Asia-Pacific Conference on, pp. 332-337. IEEE, 2015.

[43] Nguyen-Duc, Toan, and Eiji Kamioka. "Feasibility of SDN-based vertical handover between bluetooth and Wi-Fi." In Communications, Management and Telecommunications (ComManTel), 2015 International Conference on, pp. 24-29. IEEE, 2015.

[44] Nguyen-Duc, Toan, and Eiji Kamioka. "An SDN Approach for an Energy Efficient Heterogeneous Communication Network in Disaster Scenarios." arXiv preprint arXiv:1701.02150 (2017).

[45] Nguyen-Duc, Toan, and Eiji Kamioka. "A self-adaptive and energy-efficient wireless multi-hop network." In Communications (APCC), 2016 22nd Asia-Pacific Conference on, pp. 296-301. IEEE, 2016.

[46] Nguyen-Duc, Toan, and Eiji Kamioka. "An Energy-Efficient Mobile-Controlled Vertical Handover Management for Real Time Services." Journal of Computer and Communications 4, no. 17 (2016): 59.

[47] Toan Nguyen Duc, and Eiji Kamioka, "esVHO: Energy Saving Vertical Handover Extension for Local SDN in non-Interconnected Environment," IEICE Special Section: Network Virtualization, Network Softwarization and Fusion Platform of Computing and Networking, Vol.E100-B, No.11, Nov.2017 (Accepted).

[48] Open Networking Foundation. https://www.opennetworking.org/.

[49] Rescorla, Eric, Marsh Ray, Steve Dispensa, and Nasko Oskov. Transport layer security (TLS) renegotiation indication extension. No. RFC 5746. 2010.

[50] Python. https://www.python.org/

[51] Iperf tool, http://iperf.sourceforge.net/

[52] Quang Tran Minh, Kien Nguyen, Cristian Borcea, and Shigeki Yamada, “On-the-fly establishment of multihop wireless access networks for disaster recovery”, IEEE Communications Magazine, 52:60–66, 2014.

[53] Clarke L, “Panic: myth or reality?” Contexts (American Sociological Association). Fall 2002:21–26. www.asanet.org/contexts.

[54] Quarantelli EL, Dynes RR, “When disaster strikes (It isn’t much like what you’ve heard or read about)”, Psychology Today. February 1972:67–70.

[55] Quarantelli EL, “The sociology of panic”, Newark, DE: University of Delaware, Disaster Research Center; 1999. www.udel.edu/DRC/preliminary/pp283.pdf.

[56] Scanlon TJ, “Disaster Preparedness: Some Myths and Misconceptions: The Easingwold Report”, Wallasey, UK: Eaton Press; 1992.

[57] G. Phillip and R. Hodge, “Disaster area architecture: telecommunications support to disaster response and recovery”, In Military Communications Conference, 1995. MILCOM ’95, Conference Record, IEEE, volume 2, pages 833–837, 1995.

[58] SAFECOM. Public safety statements of requirements for communications and interoperability vols. I and II. In internal report. US communications program of the Department of Homeland Security, 2006.

[59] Trevor Pering, Yuvraj Agarwal, Rajesh Gupta, and Roy Want, “Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces”, In Proceedings of the 4th International Conference on Mobile Systems, Applications and Services, pages 220–232, 2006.

[60] Carles Gomez, Joaquim Oller, and Josep Paradells, “Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology”, Sensors, 12(9):11734–11753, 2012.

[61] ITU-T Recommendation P.59, "Artificial conversational speech”, March, 1993.

[62] Aqeel Mahesri and Vibhore Vardhan, “Power consumption breakdown on a modern laptop”, In Proceedings of the 4th International Conference on Power-Aware Computer Systems, pages 165–180, 2005.

[63] Kassar, M., Kervella, B., and Pujolle, G., Kassar, Meriem, Brigitte Kervella, and Guy Pujolle. "Autonomic-oriented architecture for an intelligent handover management scheme." In 6th Annual Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual, pp. 139-146. IEEE, 2008.

[64] VLC tool, http://www.videolan.org/

[65] Audio recorder tool, https://launchpad.net/audio-rec

[66] Open vSwitch. http://openvswitch.org/.

[67] Python psutil library. https://pypi.python.org/pypi/psutil.

[68] ITU-T Recommendation G.114, "One-Way Transmission Time", 1996.

[69] ITU-T Recommendation Y.1541, "Network performance objectives for IP-based services", December, 2011

[70] Yang, Mao, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. "OpenRAN: a software-defined ran architecture via virtualization." In ACM SIGCOMM computer communication review, vol. 43, no. 4, pp. 549-550. ACM, 2013.

[71] Yap, Kok-Kiong, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. "OpenRoads: Empowering research in mobile networks." ACM SIGCOMM Computer Communication Review 40, no. 1 (2010): 125-126.

[72] Kim, Tae-sub, Ryong Oh, Sang-Joon Lee, Suk-Ho Yoon, Choong-Ho Cho, and Seng-Wan Ryu. "Vertical handover between LTE and wireless LAN systems based on common resource management (CRRM) and generic link layer (GLL)." In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1160-1166. ACM, 2009.

[73] Bukhari, Jumana, and Nadine Akkari. "QoS based approach for LTE-WiFi handover." In Computer Science and Information Technology (CSIT), 2016 7th International Conference on, pp. 1-6. IEEE, 2016.

[74] Coskun, Gurkan, Ibrahim Hokelek, and Hakan Ali Cirpan. "Energy efficient handover between heterogeneous networks using IEEE 802.21." In Signal Processing and Communications Applications Conference (SIU), 2014 22nd, pp. 674-677. IEEE, 2014.

[75] Gehlot, Abhishek, and Anand Rajavat. "Handoff between WiMAX and WiFi wireless networks." In Colossal Data Analysis and Networking (CDAN), Symposium on, pp. 1-5. IEEE, 2016.

[76] Hamaydeh, Nasim, Abedelaziz Khalil, Samer Bali, and Labib Arafeh. "The impact of mobile speed on vertical handover process between WiFi and WiMAX networks." In RF and Microwave Conference (RFM), 2013 IEEE International, pp. 138-143. IEEE, 2013.

[77] Dai, Zuoran, Roberta Fracchia, Jeremy Gosteau, Pietro Pellati, and Guillaume Vivier. "Vertical handover criteria and algorithm in IEEE802. 11 and 802.16 hybrid networks." In Communications, 2008. ICC'08. IEEE International Conference on, pp. 2480-2484. IEEE, 2008.

[78] Edward, E. Prince, and V. Sumathy. "A survey of seamless vertical handoff schemes for Wi-Fi/WiMAX heterogeneous networks." In Signal Processing and Communications (SPCOM), 2010 International Conference on, pp. 1-5. IEEE, 2010.