We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

End-to-End Handover Management for VoIP Communications in Ubiquitous Wireless Networks

Shigeru Kashihara[1], Muhammad Niswar[2], Yuzo Taenaka[3],
Kazuya Tsukamoto[4], Suguru Yamaguchi[1] and Yuji Oie[4]
[1]*Nara Institute of Science and Technology*
[2]*University of Hasanuddin*
[3]*The University of Tokyo*
[4]*Kyushu Institute of Technology*
[1,3,4]*Japan*
[2]*Indonesia*

1. Introduction

With the development and widespread of diverse wireless network technologies such as wireless local area networks (WLANs) and worldwide interoperability for microwave access (WiMAX), the number of mobile internet users keeps on growing. This rapid increase in mobile internet users accelerates the further spread of these wireless networks, and thus various wireless service providers (WSPs) and individuals will provide many different wireless networks. These networks then will be the underlying basis of ubiquitous wireless networks, as illustrated in Fig. 1. Thus, ubiquitous wireless networks will provide stable Internet connectivity at anytime and anywhere. At the same time, voice over IP (VoIP) is expected to become a killer application in the ubiquitous wireless networks, i.e., the next generation cell-phone. Recently, many users have easily used VoIP communication such as Skype (Skype, 2003) in wireless networks. However, users cannot seamlessly traverse wireless networks during VoIP communication due to various factors such as the inherent instability of wireless networks, a limited communication area and changes of IP addresses.

This chapter focuses on what is needed to maintain VoIP communication quality during movement in the ubiquitous wireless networks. If you are a subscriber of a WSP, the WSP will provide for your mobility inside the WSP's wireless network. Unfortunately, as described above, in ubiquitous wireless networks consisting of wireless networks provided by various WSPs and individuals, because each wireless network has a different network address, a mobile station (MS) needs handovers with changes of IP addresses. However, in the current Internet architecture, VoIP communication is broken when changing IP addresses. Furthermore, since ubiquitous wireless networks consist of wireless networks provided by various providers, it is next to impossible for a single provider to support mobile service for users in the ubiquitous wireless networks. Hence, an MS needs a method to traverse wireless networks managed independently by different providers without communication termination. Then, even if an MS can avoid communication termination at handover, the following problems must also be resolved to maintain VoIP communication quality during movement. First, when an MS executes handover to a wireless network with a different
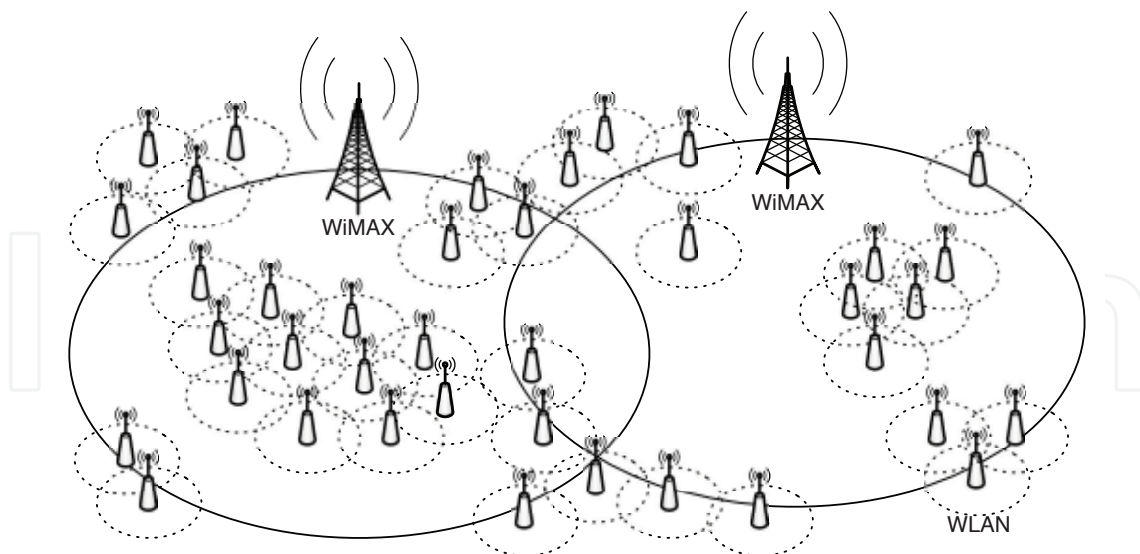
Fig. 1. Ubiquitous wireless networks

network address, layer 2 and 3 handover processes inevitably lead to interruption of VoIP communication. Second, the timing to initiate handover is also a critical issue. In fact, late handover initiation severely affects VoIP communication quality because the wireless link quality suddenly degrades. Third, how to recognize which network will be the best choice among available networks is an issue of concern. Thus, to maintain VoIP communication quality during movement, the following requirements must be satisfied.

1. Keep VoIP communication from communication termination by change of IP address

2. Eliminate communication interruption due to layer 2 and 3 handover processes

3. Initiate appropriate handover based on reliable handover triggers

4. Select a wireless network with good link quality during handover

This chapter introduces end-to-end handover management methods satisfying all of the above requirements, to maintain VoIP communication quality during movement. As illustrated in Fig. 1, since we assume that the ubiquitous wireless networks consist of a large number of WLANs and WiMAX, we focus on two mobility scenarios, i.e., WLAN-WLAN and WLAN-WiMAX scenarios. Note that the concept of our proposed methods also will be suitable for other new wireless networks such as 3GPP Long Term Evolution (LTE).

This chapter is organized as follows. Section 2 surveys related work. Section 3 presents an end-to-end handover management method in a WLAN-WLAN scenario and the implementation of the prototype system. In Section 4, to consider a more realistic environment, we extend our handover management method to apply for multi-rate and congested WLANs. Section 5 presents a handover management method among different wireless access technologies, i.e., WLAN and WiMAX. Finally, Section 6 presents concluding remarks and future work.

2. Related Work

There have been numerous discussions about supporting an MS's mobility among wireless networks with different network addresses. In this section, we focus especially on handover management needed to keep VoIP communication quality during such movement. As

described in Section 1, in the ubiquitous wireless networks, an MS may experience many handovers with changes of IP addresses. Mobile IPv4 (MIPv4) (Perkins, 2002) and Mobile IPv6 (MIPv6) (Johnson et al., 2004) have received significant interest as a network-based mobility management method to support mobility with changes of IP address. To avoid communication termination due to a change of IP address, MIPv4/v6 employs agent servers in the wireless networks, and the agent servers manage the location of MSs and control packet transmission between an MS and a corresponding station (CS). Although the agent servers do keep communication connections even when the IP address of the MS is changed, MIPv4/v6 is not enough to provide a seamless handover. To move to another wireless network, an MS has to perform layer 2 and 3 handover processes and it cannot send and receive any packets during that time. Furthermore, location registration with the agent servers also introduces an interruption delay. Thus, such interruptions lead to degradation of VoIP communication quality.

To support seamless handover with MIPv4/v6, many extension methods have been studied. In Hierarchical Mobile IPv6 (HMIP) (Soliman et al., 2008), an additional server reduces the registration period inside the same domain. However, when an MS moves between different domains, the HMIP eventually requires layer 2 and 3 handover processes and a location update like the original MIPv4/v6. In Fast handover for Mobile IPv6 (FMIP) (Koodli, 2005), additional functions are added to allow an MS to update the location before executing handover. However, FMIP also needs layer 2 and 3 handover processes after updating the location. Therefore, it does not completely eliminate communication interruption due to layer 2 and 3 handover processes (Kim et al., 2005) (Montavont & Noel, 2003). In addition, since MIP-based methods require special agent servers, they cannot easily be used in ubiquitous wireless networks because a different provider independently manages each wireless network. Thus, it is desirable to provide an end-to-end handover management method without extra network facilities.

As for end-to-end handover approaches, the mobile Stream Control Transmission Protocol (mSCTP) (Xing et al., 2002) and the Media Optimization Network Architecture (MONA) (Koga et al., 2005) have been proposed. The mSCTP is a mobile extension of the Stream Control Transmission Protocol (SCTP) (Stewart, 2007), and allows an MS to simultaneously use two or more wireless interfaces for communications, i.e., multi-homing architecture. Compared with the single-homing architecture, the multi-homing architecture can contribute to elimination of communication interruption due to layer 2 and 3 handovers because an MS can connect with another wireless network by using an idle interface before breaking off the current communication. However, the mSCTP supports only non-real-time communications such as a file transfer; real-time communications such as VoIP are not supported. On the other hand, MONA also has a multi-homing function and it can handle both real-time and non-real-time communications. However, MONA does not focus on handover management for maintaining VoIP communication quality.

3. End-to-end handover management in WLAN-WLAN scenario

This section focuses on a case where an MS traverses WLANs with different network addresses. As illustrated in Fig. 1, with the proliferation of free WLAN hotspots such as FON (FON, 2005), so that the overlapping WLANs provide wide coverage as ubiquitous WLANs, an MS will be able to access the Internet via the ubiquitous WLANs everywhere. However, the coverage of each access point (AP) is relatively small and each AP also independently provides wireless connectivity, i.e., they have different network addresses. Thus, in what

follows, we focus on end-to-end handover management to enable an MS to maintain VoIP communication while traversing WLANs with different network addresses. In Section 3.1, to maintain VoIP communication quality during movement, we first discuss a handover trigger needed to appropriately detect degradation of wireless link quality. We then introduce our handover management architecture and the implementation design in Sections 3.2 and 3.3, respectively. Section 3.4 shows the basic performance of our prototype system.

3.1 Handover trigger for WLAN

A handover trigger plays an important part in maintaining VoIP communication quality during movement. In fact, late handover initiation severely affects VoIP communication quality because the wireless link quality suddenly degrades. Prevention of such degradation requires a handover trigger that promptly and reliably detects degradation of the wireless link quality. The received signal strength indication (RSSI) is generally employed as a common index of wireless link quality. However, the RSSI fluctuates drastically due to various complicated effects such as distance to an AP, multi-path fading, and intervening objects. Moreover, the values obtained from each WLAN interface depend on a vendor, e.g., the RSSI range of Atheros's chipset is from 0 to 60 and that of Cisco's chipset is from 0 to 100 (Muthukrishnan et al., 2006). Therefore, since it is very difficult to set an optimal handover threshold for the RSSI, the RSSI cannot serve as a reliable handover trigger.

Because RSSI cannot serve as a reliable handover trigger, we focused on the number of data frame retries as a new handover trigger to promptly and reliably detect degradation of wireless link quality due to movement (Kashihara & Oie, 2007). In a WLAN, a sender can detect successful packet transmission by receiving an ACK frame in response to a transmitted data frame. If a data or an ACK frame is lost, the sender transmits the same data frame until the number of data frame retries reaches a predetermined retry limit. Note that when Request-to-Send (RTS)/Clear-to-Send (CTS) is applied, the retry limit is set to four, otherwise, the retry limit of seven is applied. If the number of data frame retries reaches the retry limit, the sender treats the data frame as a lost packet. Thus, since data frame retries mainly occur for the following two reasons: (i) reduction of RSSI and (ii) collision with other frames, we can suppose that the number of data frame retries indicates how much wireless link quality is degraded before packet loss actually occurs.

To show the effectiveness of data frame retries as a handover trigger, we investigated RSSI and data frame retries in a real environment (Tsukamoto et al., 2007). The paper discussed the characteristics of RSSI and data frame retries for FTP and VoIP applications in an open-space and an indoor environments. We here introduce only the results of VoIP communication in the indoor environment. In Fig. 2, the MS has VoIP communication with the CS via the AP, and then it goes away from the AP. In the experiment, we employed the ORiNOCO AP-4000 (Proxim, 2007) as an AP. The transmission speed of the WLAN (802.11b) is set to a fixed 11Mb/s, and RTS/CTS is activated. As a WLAN interface of the MS, the ORiNOCO 802.11a/b/g Combo Card Gold (Proxim, 2007) is used. Note that the RSSI ranges from 0 to 60 because the WLAN interface has the Atheros's chipset. An analyzing station (AS) captures transmitted frames over the WLAN by using Ethereal 0.10.13 (Ethereal, 1998).

The graph shows the results of packet loss ratio, RSSI, and the number of data frame retries for VoIP communication when the MS actually moves away from the AP at a walking speed. "Retry: n" indicates that a packet experiences frame retries "n" times, and its associated symbol marked in the graph shows when that occurs. From the graph, we can see that since the RSSI drastically fluctuates and decreases abruptly with the movement of the MS,
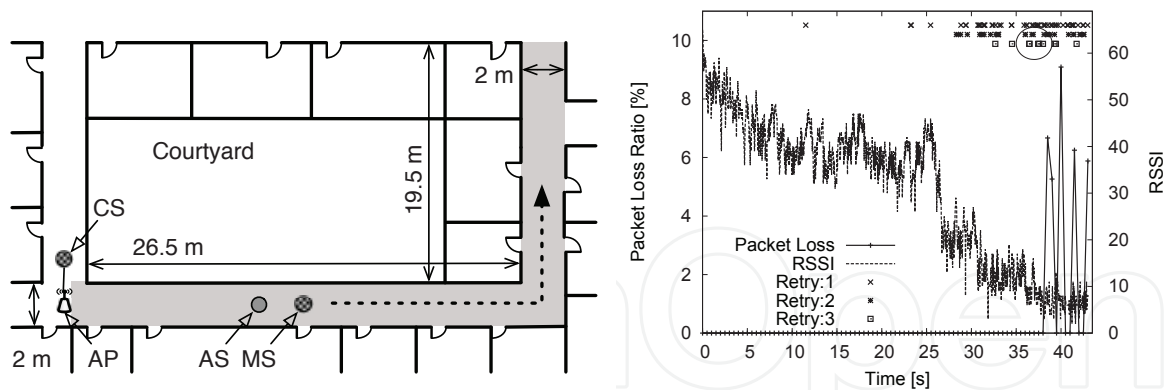
Fig. 2. Experimental environment and result

it is difficult to determine a threshold value to appropriately initiate handover. On the other hand, as for data frame retries, in particular, "Retry: 3" occurs just before appearance of lost packets. Thus, the number of data frame retries can be used to promptly and reliably detect deterioration of the wireless link quality.

3.2 Handover management architecture

As described in Section 1, in order to maintain VoIP communication quality during movement, we need to satisfy the following requirements.

1. Keep VoIP communication from communication termination by change of IP address

2. Eliminate communication interruption due to layer 2 and 3 handover processes

3. Initiate appropriate handover based on reliable handover triggers

4. Select a wireless network with good link quality during handover

Moreover, to freely traverse wireless networks without special network facilities, a handover management on an end-to-end basis is also required. We then proposed a handover management method on an end-to-end basis for ubiquitous WLANs (Kashihara & Oie, 2007) (Kashihara et al., 2007).

We outline here our handover management method. Figure 3 illustrates our architecture design for seamless handover. To satisfy requirements (1) and (2), we employed a multi-homing architecture and a handover manager (HM) on the transport layer. The multi-homing architecture enables an MS to handle two or more wireless interfaces simultaneously. If an MS with a single WLAN interface moves among WLANs with different network address, inherently it can never avoid communication termination and interruption because a single interface cannot access more than one AP at a time. On the other hand, since a multi-homing MS can execute layer 2 and 3 handover processes using an idle WLAN interface in advance before breaking off communications on the active WLAN interface, it can seamlessly switch to a candidate AP without communication termination and interruption. Moreover, to control handover without additional agent servers, the handover should be managed over the transport layer because the transport layer is the lowest layer that controls an end-to-end flow. Therefore, in our architecture, we implemented the HM, which controls handovers according to wireless link condition, on the transport layer.

To satisfy requirement (3), we employed the number of data frame retries as a new handover trigger because data frame retries inevitably occur before occurrence of packet loss in wireless networks. Thus, to control handover, the HM needs to obtain information from the MAC layer
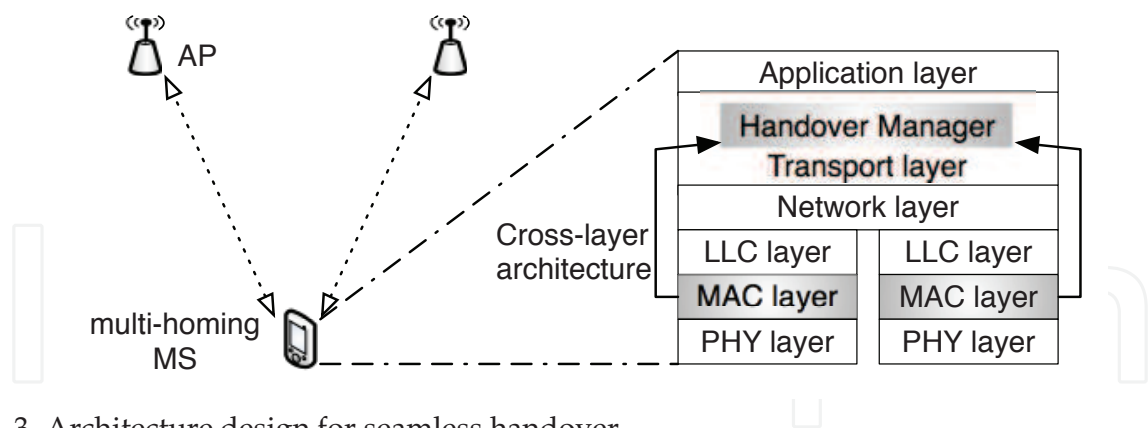
Fig. 3. Architecture design for seamless handover

based on the cross-layer architecture. As for requirement (4), the HM needs to select a WLAN with better link quality to avoid an inappropriate handover to an AP with poor link quality. In our proposed method, when performing handover in an overlap area, an MS starts to transmit duplicated packets via both APs; that is, the MS switches to multi-path transmission. During multi-path transmission, the HM investigates the wireless link quality of both APs based on the number of data frame retries and then selects the better one. After that, it reverts to single-path transmission via the selected AP. Therefore, the HM achieves a seamless handover by appropriately switching between single-path and multi-path transmissions.

3.3 Design and implementation

As described above, to achieve an end-to-end seamless handover, we need to implement a multi-homing architecture, cross-layer architecture, and multi-path transmission function. In this section, since we actually implemented our prototype system on a real system (Taenaka et al., 2007), we introduce its design and implementation.

In our implementation, we first employed MONA (Koga et al., 2005) as the base system enabling an MS to handle multiple wireless interfaces. That is, our multi-homing architecture basically depends on MONA. Next, we explain how to exploit the cross-layer architecture, which enables the HM to obtain the number of data frame retries from the MAC layer. In the previous simulation study (Kashihara & Oie, 2007), although the number of data frame retries is directly passed from the MAC layer to the HM at every packet through the cross-layer architecture, we found that it actually causes significant deterioration of kernel performance due to frequent interruptions. Therefore, in the paper (Taenaka et al., 2007), we proposed an asynchronous process between the HM and the MAC layer. In the design, as illustrated in Fig. 4, the MAC layer for each WLAN interface writes the number of data frame retries into its own shared memory, and the HM retrieves the information from the shared memory. The shared memory consists of (1) index and (2) retry count. The retry count region consists of an array containing 100 elements with a ring buffer. Actually, the MAC layer records the number of data frame retries for one data packet in the shared memory whenever each data packet is successfully transmitted or else discarded due to maximum frame retries. Then, the MAC layer also writes the latest array position of the retry count region into the index region.

To achieve a seamless handover, our proposed method also employed two transmission modes, i.e., single-path and multi-path transmission modes. Next, we describe the details of the switching procedures. An MS usually communicates by single-path transmission. When the number of data frame retries exceeds the Multi-Path Threshold (MP_TH) in the HM, the
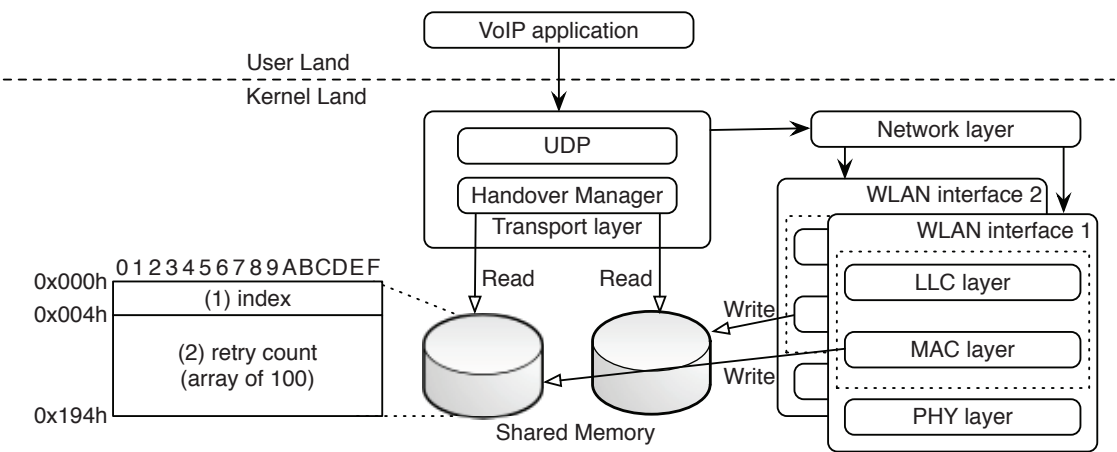
Fig. 4. Design of cross-layer architecture and shared memory

HM switches to multi-path transmission to prevent packet loss and to investigate the wireless link quality of both WLANs. Figure 5 illustrates the flowchart for switching to multi-path transmission. Note that the process is executed at every packet transmission. The flowchart is divided into two parts: (a) reading the information from the shared memory, and (b) switching to multi-path transmission according to the wireless link quality. In the flowchart, "italic letters" and "bold letters" indicate variable and system parameters, respectively. In our proposed method, since the HM and the MAC layer work asynchronously, some packets may already be sent at the MAC layer when the HM checks the number of data frame retries, i.e., the HM needs to check past packet transmissions (see Fig. 5(a)). Then, the HM first checks the range of elements (*get_cnt*) updated in the retry count region after the previous execution. This procedure is depicted in Fig. 6. In the procedure, two position indexes are employed: *start_pos* indicates an array position where the HM starts to obtain the number of data frame retries in the retry count region, and *end_pos* indicates the latest array position. To get *start_pos*, the process first checks whether this is a first-time execution or not. If so, *start_pos* is set to 0. Otherwise, *start_pos* has already been set to the latest array position in the previous execution. On the other hand, *end_pos* is set to the value of the index region in the shared memory. Therefore, updated elements (*get_cnt*) are calculated by *start_pos* and *end_pos*.

In Fig. 5(b), the HM compares the number of data frame retries (*retries*) with the MP_TH, which is a threshold to switch to the multi-path transmission, *get_cnt* times. In the comparison, if a value obtained from the retry count region exceeds the MP_TH, the HM immediately escapes
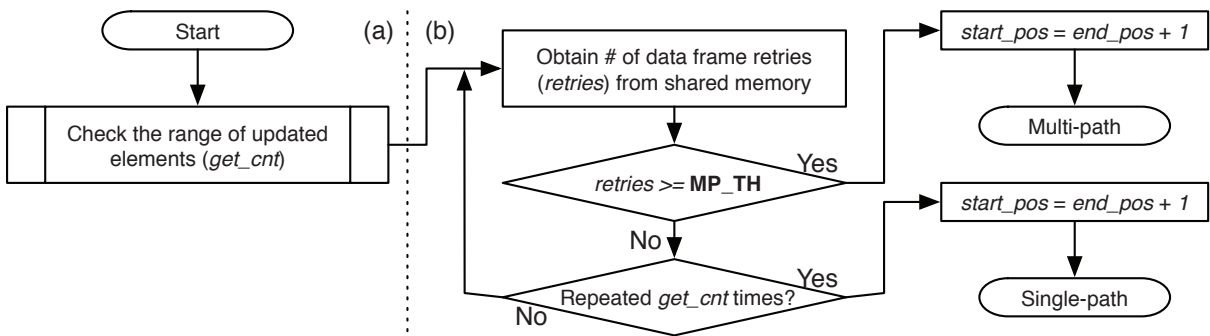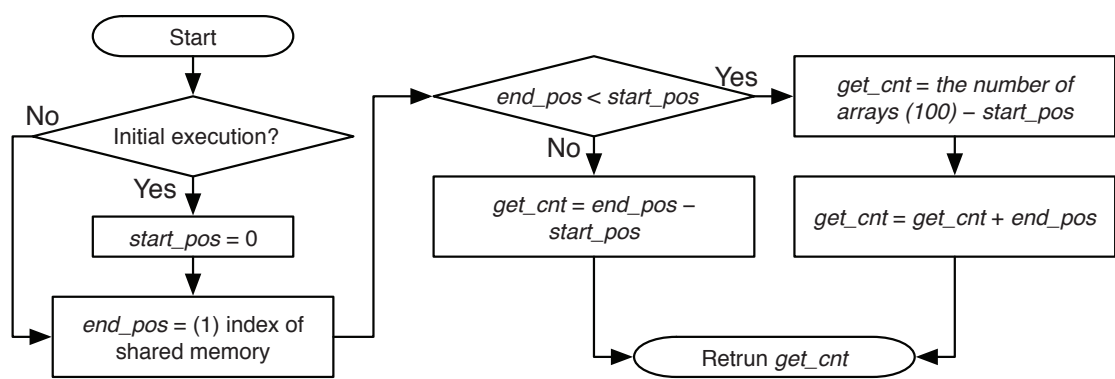
Fig. 5. Switching to multi-path transmission

Fig. 6. Calculation of the range of updated elements

from the loop and switches to the multi-path transmission. Then, *start_pos* is set to *end_pos* + 1 for the next execution. Otherwise, the HM continues to compare it with the MP_TH. If any element does not exceed the MP_TH, single-path transmission continues. Then, after the process, *start_pos* is set to *end_pos* + 1 for the next execution.
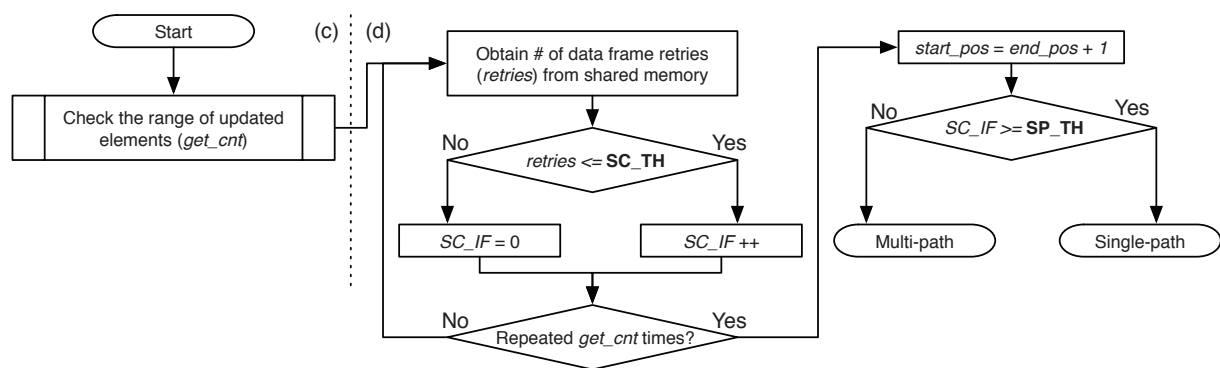

Fig. 7. Switching to single-path transmission

In multi-path transmission, since an MS sends duplicated data packets through the two WLANs, the network traffic load doubles. Thus, an MS needs to return to single-path transmission as soon as possible to reduce the extra network traffic. Figure 7 illustrates a switching operation to single-path transmission. In Fig. 7(c), the HM first calculates the range of updated elements (*get_cnt*) in the retry count region for each WLAN, as does the single-path transmission operation. In Fig. 7(d), the HM then obtains the updated *retries* for each WLAN interface from the shared memory. The HM continuously compares the obtained value with the SC_TH, which is a threshold to check the stability of wireless link condition, for each WLAN interface *get_cnt* times. If the value is smaller than the SC_TH, the *SC_IF*, which is a counter for the network stability, is incremented by one. Otherwise, *SC_IF* is reset to zero because the HM decides that the wireless link quality for the WLAN interface is still unstable. After the loop, the HM updates *start_pos* for the next execution and compares the *SC_IF* of each WLAN interface with the SP_TH, which is a threshold to return to single-path transmission. If the *SC_IF* exceeds the SP_TH, the HM switches back to single-path transmission. Otherwise, the HM continues multi-path transmission.

Next, we describe our implementation environment. Our handover management architecture is implemented in the Cent OS 4.3 (Linux kernel version 2.6.9) on Lenovo ThinkPad X60 (CPU: Core Duo 1.66 GHz, Memory: 512 MB). Since an MS has two WLAN interfaces for

a multi-homing architecture, it employs a built-in WLAN interface (P/N: 40Y7028) and a PC card WLAN interface (ORiNOCO 802.11 a/b/g Combo Card Gold). Then, to extract the number of data frame retries from the WLAN interfaces with Atheros chipset, the MadWifi driver (MadWifi, 2004) is employed. The cross-layer architecture is implemented only on the MadWifi driver.

3.4 Performance evaluation

This section demonstrates the basic performance of our prototype system described in Section 3.3 (Bang et al., 2009). Our proposed method employs the following three thresholds, MP_TH of three, SP_TH of two, and SC_TH of one, to control handover. The three values are determined based on the results in our paper (Kashihara & Oie, 2007).

Figure 8 shows the experimental topology and the result. In the topology, we employed five PCs for an MS, a CS, two APs, and a router. The router belongs to three networks with different network addresses and directly connects to the CS and the two APs by wired connection. Since we assume that ubiquitous WLANs consist of many WLANs provided by various providers, the paths to the CS have different delays to reproduce a realistic environment; that is, the path delay through AP1 is 10 ms and that through AP2 is 30 ms. The delays are intentionally added at the router using dummynet of FreeBSD (FreeBSD, 1995). The two APs are constructed on two identical laptops (HP nx6120) with a PC card WLAN interface (ORiNOCO 802.11 a/b/g combo Card Gold) in which Fedora Core 6 with MadWifi driver is installed. Then, both APs are configured as the master mode of the MadWifi driver to stand as an AP. In the wireless settings, the data rate is fixed at 11 Mb/s of IEEE 802.11b and the RTS/CTS function is disabled. The distance of APs is 45 meters.
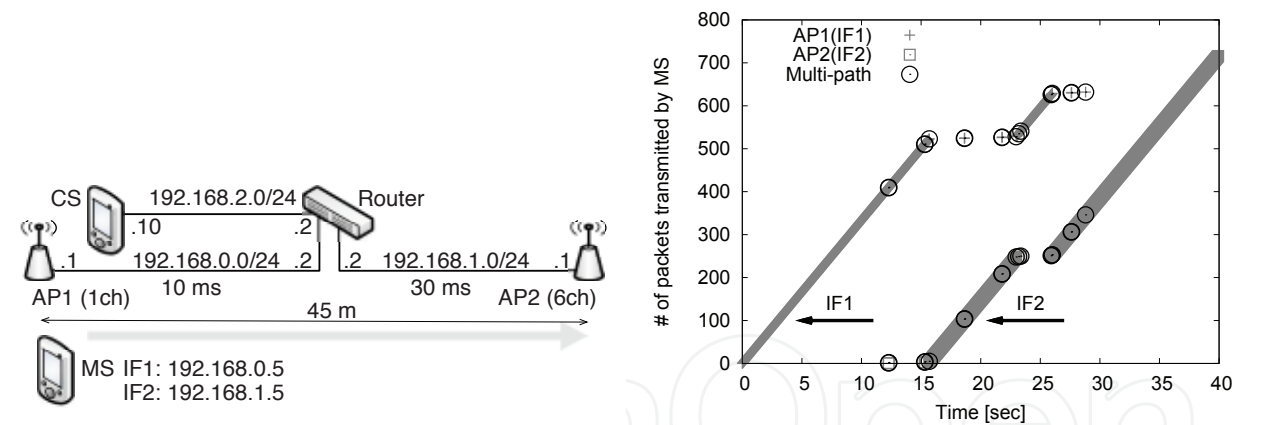

Fig. 8. Experimental topology and result

Now we describe our experimental scenario. Since we focus on the VoIP communication performance during handover, the two WLAN interfaces of the MS are assumed to associate with the two APs before starting an experiment. That is, WLAN interface 1 (IF1) associates with AP1, and, likewise, WLAN interface 2 (IF2) associates with AP2. This means that the MS is in an overlap area of two APs and has already connected to them. Then, after starting to capture traffic using tcpdump (TCPDUMP, 2000), the MS walks from AP1 to AP2 while communicating with the CS using VoIP (G.711). When it arrives at AP2, we stop capture of the traffic. The graph shows the number of packets transmitted by the MS over time. The AP1 and AP2 marks indicate when single-path transmission through AP1 and AP2 are executed, respectively, while that of multi-path shows when multi-path transmission is employed. The graph shows that the MS initiates handover at approximately 15 seconds according to the

Maximum	Minimum	Average	Median
11	0	4.7	5

Table 1. Packet loss for nine experiments

Maximum	Minimum	Average	Median
2.1 %	0.4 %	1.3 %	1.2 %

Table 2. Multi-path transmission ratio for nine experiments

change of the wireless link condition. After that, the MS repeats the handovers two more times: it returns back to AP1 at approximately 23 seconds and switches again to AP2 at approximately 26 seconds. Note that we here define switching to single-path transmission on the next WLAN as a handover. Table 1 shows the number of lost packets from the MS to the CS throughout the nine experiments. From the result, we can see that the prototype system can execute handover seamlessly. Table 2 lists the multi-path transmission ratios during the nine experiments. The result shows that our prototype system has extremely few redundant packets. Therefore, our prototype system can achieve seamless handover while appropriately switching between single-path and multi-path transmissions.

4. End-to-end handover management for multi-rate and congested WLANs

In the previous section, we introduced a prototype system of our handover management method to maintain VoIP communication quality during movement in ubiquitous WLANs. Although we focused on degradation of wireless link quality during movement in Section 3, we also need to consider the impact of multi-rate function and congestion at an AP to apply for a more realistic environment. Section 4.1 first discusses the handover triggers for multi-rate and congested WLANs. We next introduce our handover management method and show the basic performance in Section 4.2 and 4.3, respectively.

4.1 Handover triggers for multi-rate and congested WLANs

In Section 3, we introduced the number of data frame retries as a handover trigger for movement in WLANs with a fixed transmission rate (11 Mb/s). However, in a real environment, almost all WLANs employ a multi-rate function that can change the transmission rate according to the wireless link condition. If the transmission rate is dropped by the multi-rate function, a more robust modulation type is selected and thus data frame retries are decreased. As a result, since an MS cannot properly detect degradation of wireless link quality only from data frame retries in multi-rate WLANs, we need to consider more reliable handover triggers.

Next, we consider an RTS frame retry ratio as an alternative metric of data frame retries. Since an RTS frame is always transmitted at the lowest rate (e.g., 6 Mb/s in 802.11a/g and 1 Mb/s in 802.11b), an MS can appropriately detect changes in wireless link quality by utilizing it. Moreover, the RTS frame also prevents collisions due to hidden nodes in a wireless network. However, as the RTS threshold is set to 2,347 bytes in the IEEE802.11 standard, an RTS frame is not sent because of the VoIP packet size (e.g., 160 bytes). Therefore, in our proposal, all MSs must set the RTS threshold to 0 to enable MSs to send RTS frames. To show the effectiveness of the RTS frame retry, we investigated the behavior of RTS retry ratio when an MS moves

away from an AP using Qualnet 4.0.1 (Scalable Network Technologies, 2006) (see Fig. 9(a)). In the study, RTS frame retry ratio is employed instead of the frequency of data frame retries for reliable detection of degradation of wireless link quality. Actually, an instantaneous increase of RTS frame retries may lead to more misdetection of wireless link quality. The RTS retry ratio is calculated as follows:

$$RTS\ frame\ retry\ ratio = \frac{the\ number\ of\ RTS\ frame\ retries}{total\ transmitted\ RTS\ frames} \tag{1}$$

Fig. 9. Simulation model

Figure 10 shows the relationships between the MOS and the RTS frame retry ratio over distances between the AP and the MS. We here employ MOS (ITU-T G.107, 2000) to assess the VoIP quality. Note that MOS of more than 3.6 indicates an adequate VoIP call quality. The graph shows that the MOS tends to degrade with increase in the RTS frame retry ratio when the MS moves away from the AP. Since the RTS frame retry ratio is fluctuating due to the unstable wireless link quality, we employed a least-squares method to grasp their trend and estimate the best fit of the occurrences of RTS frame retry ratio over the distance, shown as a straight line in the graph. The line shows that the RTS frame retry ratio of 0.6 indicates the starting point of VoIP quality degradation. Hence, we employ the RTS frame retry ratio of 0.6 as one of the thresholds to initiate handover in this study.

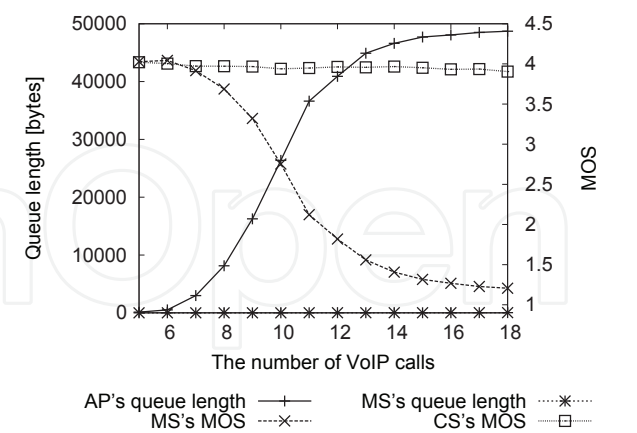Fig. 10. RTS frame retry ratio vs. MOS over distance

Fig. 11. Relationship among the number of VoIP calls, queue length, and MOS

We next consider a problem in a congested WLAN. In a congested WLAN, with the increase of VoIP calls, the AP queue length also increases. This is because the transmission opportunity of an AP is the same as that of each MS. That is, when the number of VoIP calls increases, the transmission probability from an AP to MSs decreases. As a result, packets routed to the MS

are queued in the AP buffer, and they may experience large queuing delays or packet losses due to an increase in the queue length or the buffer overflow. Consequently, the increase in an AP's queue length severely affects the VoIP quality at MSs. However, an AP based on the IEEE802.11 (a/b/g/n) standard unfortunately does not provide a mechanism that can inform MSs of the AP's queue length. Therefore, to maintain VoIP quality, each MS needs to autonomously detect the congestion of the AP.

Next we investigated the relationship between the number of VoIP calls and an AP's queue length through simulation experiments (see Fig. 9(b)). In the simulation scenario, we randomly locate from one to 18 MSs in a WLAN. Each MS communicates with a CS using VoIP. Figure 11 shows the relationships between the number of VoIP calls, AP's and MS's queue length, and MS's and CS's MOS. The graph shows that although the CS's MOS is kept adequate even if the AP's queue length increases with the increase of VoIP calls, the MS's MOS degrades significantly. Although the results indicate that the AP's queue length has a large impact on VoIP communication quality, how can MSs detect the increase in the AP's queue length without modifying an AP? We then proposed estimating the AP's queue length based on the RTT between an MS and an AP (Niswar et al., 2009a). Note that in this chapter the RTT between the MS and the AP is called Wireless RTT (WiRTT). As depicted in Fig. 12, to calculate the WiRTT, an MS periodically sends a probe request packet (ICMP message) to an AP and receives a probe reply packet from an AP. When there is an increase in the AP's queuing delay, the WiRTT also increases because the probe reply packet inevitably experiences queuing delay in the AP buffer. Therefore, the WiRTT can be used to derive information about the AP's queue length.
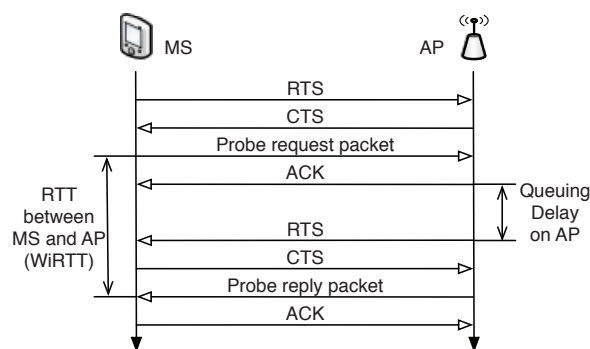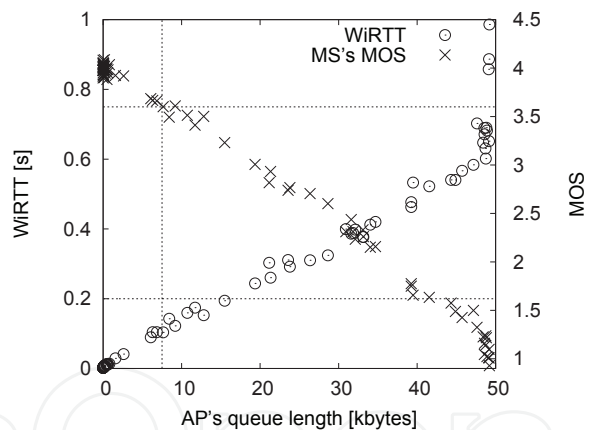

Fig. 12. RTT between MS and AP (WiRTT)

Fig. 13. Relationship among AP's queue length, WiRTT and MOS

Figure 13 shows the relationships between the AP's queue length, WiRTT and MOS in the simulation model of Fig. 9(b). The graph shows that to satisfy adequate VoIP quality (MOS of 3.6), the AP's queue length should be kept less than 7,500 bytes. That is, a WiRTT that is less than 200 ms can keep adequate VoIP quality. Therefore, in our proposed method, we employ WiRTT to estimate the AP's queue length and set the WiRTT threshold (WiRTT_th) at 200 ms to maintain an adequate VoIP quality.

The WLAN also supports a multi-rate function, which can automatically change the transmission rate based on the wireless link condition. In the case where the wireless link quality degrades, for example when the transmission rate is degraded by a change in the modulation type, packets sent at the lower transmission rate occupy more wireless resources,

i.e., longer transmission period. As a result, the lower transmission rate is likely to cause congestion of an AP. Therefore, to avoid congestion of an AP, the transmission rate should also be treated as a handover trigger.

4.2 Handover management for multi-rate and congested WLANs

As described above, to achieve seamless handover among multi-rate and congested WLANs, we employ RTS frame retry ratio, WiRTT, and transmission rate as handover triggers. We then proposed an extended handover management method based on these triggers (Niswar et al., 2009a). To support soft handover on an end-to-end basis, the handover management method also supports multi-homing, cross-layer architectures, a multi-path transmission function, and a handover manager (HM) similar to the previous method (Kashihara & Oie, 2007).

Figure 14 shows an algorithm for switching to single/multi-path transmission when an MS moves within the overlap area of two APs (AP1 and AP2). In the proposed method, an HM transmits a probe packet to the two associated APs at 500 ms intervals to estimate each AP's queue length. If *WiRTTs* of both WLAN interfaces (IF1 and IF2) are below the WiRTT_th, the HM detects that both APs are not congested. The HM then investigates the RTS frame retry ratio (*retry_ratio*) of the current active IF to detect degradation of the wireless link condition due to movement. If the *retry_ratio* reaches the threshold to switch to multi-path transmission (R_Mth), the HM switches to multi-path transmission to investigate both the wireless link conditions and avoid packet loss. On the other hand, if the *WiRTT* of IF1 reaches WiRTT_th, i.e., detection of congestion at the AP1, the HM switches to the AP2 directly without switching to multi-path transmission because the multi-path transmission leads to more serious congestion in WLANs, and vice versa. Finally, if both measured *WiRTTs* reach WiRTT_th, the HM then investigates the wireless link condition indicated by the *retry_ratio* of the active single WLAN interface.
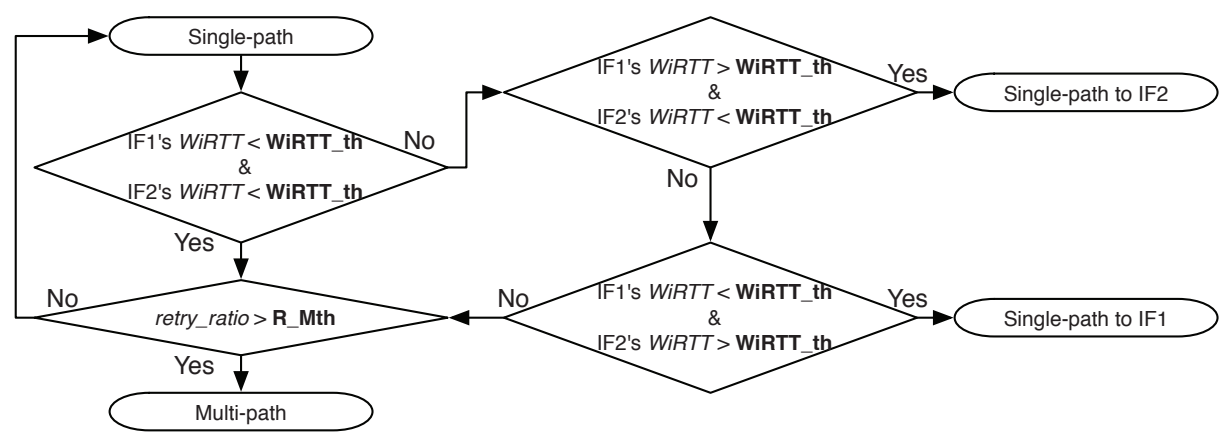

Fig. 14. Switching to single/multi-path transmissions

In multi-path transmission, to maintain VoIP quality and investigate both wireless link qualities, an HM sends the same data packets via both IFs. Hence, the HM needs to switch back to single-path transmission as soon as possible to prevent redundant network overload. Figure 15 illustrates an algorithm for switching back to single-path transmission. The HM measures *WiRTTs* of both IFs at all times. If either of the *WiRTTs* is below the WiRTT_th, the HM switches to the IF with the smaller *WiRTT*. If both *WiRTTs* are simultaneously below the WiRTT_th, the HM then compares the *retry_ratio* of both IFs. Figure 16 shows an algorithm to compare RTS retry ratios of both IFs. If both *retry_ratio* of the IFs are equal, the HM continues

multi-path mode. On the other hand, if either of the *retry_ratio* is below the threshold to switch back to single-path transmission (R_Sth), the HM switches back to single-path transmission through the IF with the smaller *retry_ratio*.
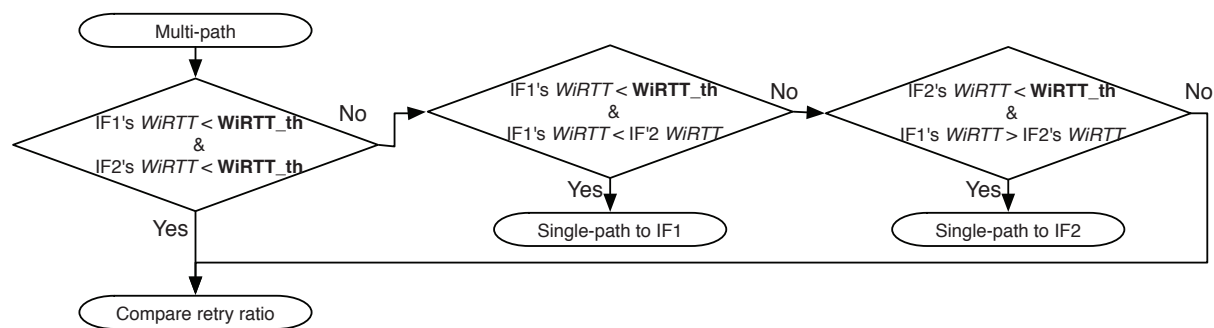

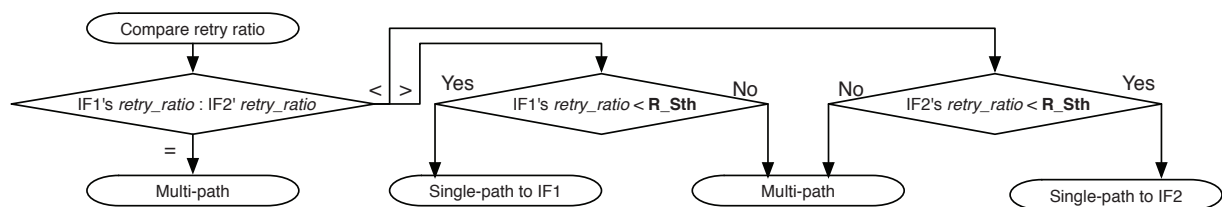Fig. 15. Switching from multi-path transmission to single-path transmission

Fig. 16. Comparing RTS retry ratios

If all MSs send probe packets to measure *WiRTT*, the probe packets may contribute to the congestion of the AP. As a result, they may unfortunately detect congestion of the serving AP (e.g., AP1) at nearly the same time. Then, they may switch the communication to a neighbor AP (e.g., AP2) and leave the AP1. As a result, AP2's queue length drastically increases, and then they switch back to the AP1 again. This phenomenon, typically known as the ping-pong effect, leads to degradation of all VoIP quality due to fluctuations in both APs' queue length. To avoid the ping-pong effect, an MS executes handover based on its own current transmission rate, as shown in Fig. 17. As mentioned earlier, since an MS with lower transmission rate occupies more wireless resources, it is more liable to lead to congestion in the AP. Moreover, as MSs with a lower transmission rate typically are farther away from the connected AP, they should execute handover as soon as possible to maintain their VoIP communication quality. Hence, in our proposed method, MSs start to execute handover in order based on transmission rates. The process actually works as follows. When the serving AP is congested, all MSs detect the congested AP through *WiRTT*. Then, MSs with the lowest transmission rate of 6Mb/s first execute the handover since transmission rate number (*TR_num*) is set to zero by default. Note that *TR_num* of zero indicates 6 Mb/s. After that, if the AP's queue length is still congested after Time_th seconds, the remaining MSs increase the *TR_num* by one, i.e., the *TR_num* of one indicates 9 Mb/s. Then, MSs with transmission rates under 9 Mb/s execute handover. This handover process is repeated until congestion of the AP is alleviated. If the congestion is alleviated, *TR_num* of all MSs are set back to the default value, i.e., zero. That is, each MS can autonomously execute handover without knowing whether another MS with lower transmission rate has executed the handover or not. Therefore, to execute handover, all MSs monitor only their own transmission rate and compare the rate with the current *TR_num*.

If every MS sends a probe packet to measure *WiRTT*, these probe packets may aggravate congestion in a WLAN. To eliminate redundant probe packets, we further extended the
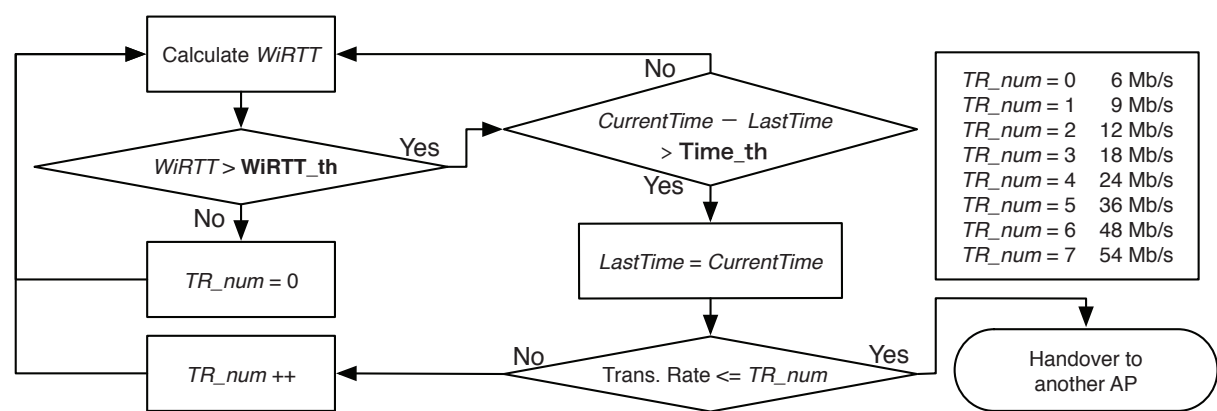
Fig. 17. Handover based on transmission rate

handover management method. In the extension, only one representative MS sends a probe request packet to an AP, and then other MSs measure *WiRTT* by capturing the probe packets that the representative MS sends and receives. Figure 18 shows how to calculate *WiRTT* from captured probe packets. Each MS first monitors all packets over a WLAN before sending a probe packet by itself. If it receives a probe request packet sent by another MS, it cancels the transmission of a probe request packet and tries to measure *WiRTT* by capturing the probe and probe reply packets that the representative MS exchanges with the AP. Each MS can identify whether a captured packet is a probe packet or not by checking the probe packet size (64 bytes). An MS can also identify whether a probe packet is a request (ICMP Request) or a reply (ICMP Response) by observing the MAC address of the captured probe packet because all MSs can identify the MAC address of the connecting AP. If the destination MAC address of the captured probe packet is that of the AP, each MS can judge the packet is a probe request packet. On the other hand, if the source MAC address is an AP's, then each MS judges the packet is a probe reply packet. In Fig. 18, *ProbeReq_Time* and *ProbeReply_Time* are the receiving time of a probe request packet transmitted by another MS and that of a probe reply packet transmitted by the AP, respectively. As every MS can identify whether a captured packet is a probe request or probe reply, it can calculate the *WiRTT* (= *ProbeReply_Time - ProbeReq_Time*) properly. In this way, our proposed method can eliminate redundant probe packets.
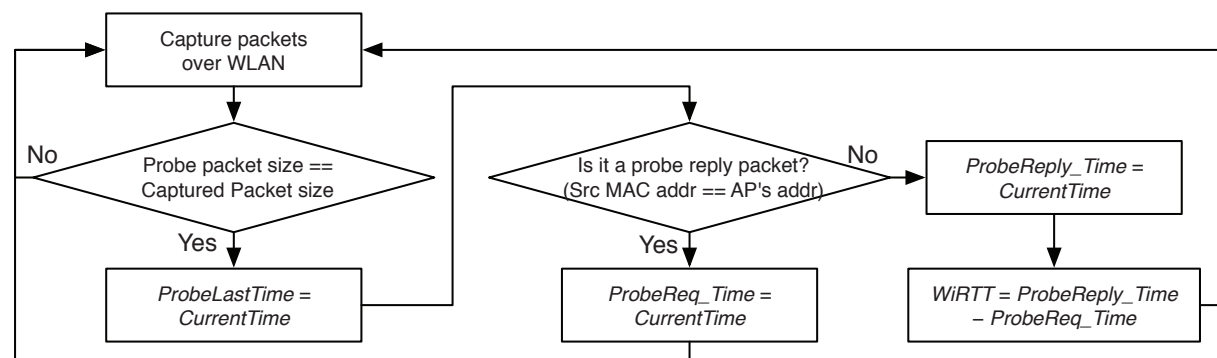

Fig. 18. Calculate *WiRTT* from captured probe packets

If the representative MS leaves a WLAN, one of the remaining MSs needs to start periodically sending a probe packet as a representative MS. Figure 19 shows how a representative MS is selected. First, all MSs always examine the difference between the last received time of a probe

packet (*ProbeLastTime*) and the current time (*CurrentTime*). If the difference is greater than probeAbsenceTime, that is, if an MS cannot capture a probe packet for probeAbsenceTime seconds, MSs with the lowest transmission rate in a WLAN try to send a probe packet. This is because almost all MSs in the WLAN can capture a probe packet transmitted at the lowest transmission rate. Thus, the timing to send a probe packet among MSs is based on *WaitingTime*. Basically, an MS with the smallest *WaitingTime* will be the representative MS because *WaitingTime* is calculated based on *TR_Weight*, which indicates the weight of the transmission rate. Thus, if *TR_Weight* is lower, then an MS gets a small *WaitingTime*. If there are several MSs with the same transmission rate, then a random value in *WaitingTime* helps to stagger the timing to send a probe packet.

Fig. 19. Selection of a representative MS

4.3 Performance evaluation

This section shows the basic performance of the proposed method. The proposed method was implemented on Qualnet 4.0.1. Figure 20 shows a simulation model and the system parameters. In the simulation, 15 multi-homing MSs randomly move within a coverage area between two APs at the speed of 1 m/s. We employed a G.711 VoIP codec that sends a 160-byte packet at 20-ms intervals.

Figure 21 shows the MOS and the AP1's queue length over time for the previous method using only the data frame retries and the extended method. In the left graphs (the previous method), the average of AP1's queue length is extremely high and MS's MOS does not satisfy adequate VoIP quality (3.6) at all. On the other hand, in the right figure (the extension method), the extension method almost always maintains adequate VoIP quality. Also, even though MOS did degrade at times, it recovered promptly because each MS investigated the wireless link quality, congestion state, and its own transmission rate. Therefore, MSs can promptly and reliably execute handover among multi-rate and congested WLANs based on the handover triggers.
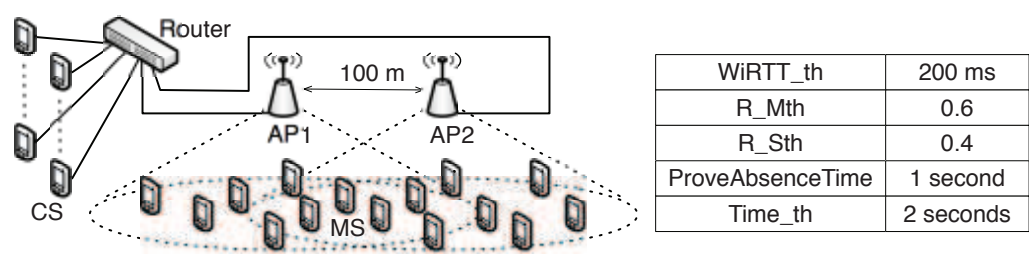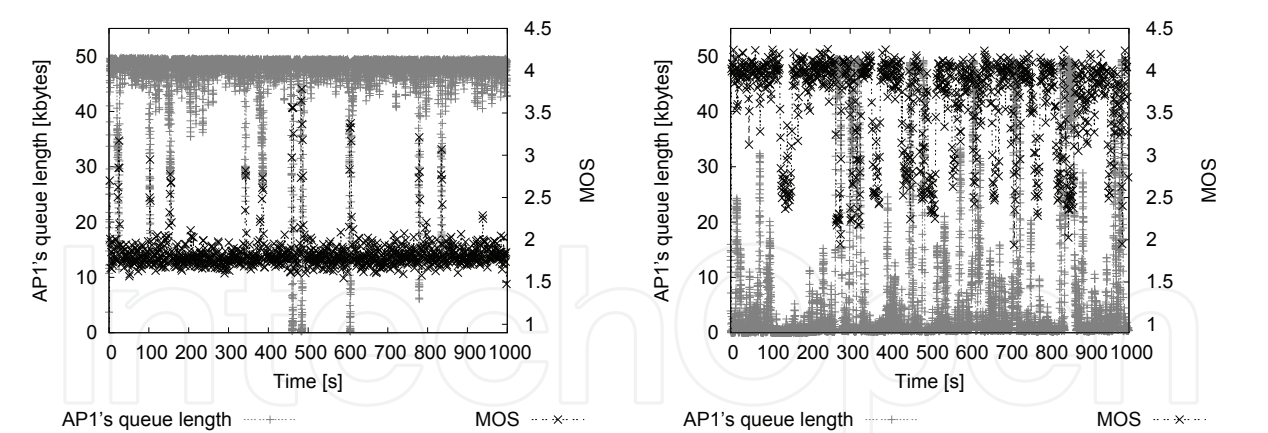

Fig. 20. Simulation model

Fig. 21. Relationships among AP1's queue length and MOS for the previous (left graph) and the extension (right graph) methods

5. End-to-end handover management in WLAN-WiMAX scenario

The previous sections introduced handover management methods in a WLAN-WLAN scenario. However, as shown in Fig. 1, mobile users also have opportunities to connect with different types of wireless networks such as WiMAX and LTE. This section focuses on an end-to-end handover management method in a WLAN-WiMAX scenario. This section contributes to selection of appropriate handover triggers and development of handover management methods among different wireless technologies. This concept will be applied to other future wireless networks. Section 5.1 first discusses the handover triggers for WiMAX. We then introduce our handover management method and the basic performance in Section 5.2 and 5.3, respectively.

5.1 Handover triggers for WiMAX

In a WLAN-WiMAX scenario, unlike a WLAN-WLAN scenario, an MS connects to wireless networks with different wireless technologies over time. However, since these wireless networks have different wireless characteristics, we cannot use the same handover triggers and handover management to maintain VoIP communication quality during movement. Thus this section discusses appropriate handover triggers for WiMAX. Note that for a WLAN we use RTS retry ratio, WiRTT, and transmission rate as the handover triggers described in Section 4.

Since we focus on handover management on an end-to-end basis, handover triggers should be basically obtained on the MS side. Furthermore, handover triggers also need to detect both wireless link conditions and congestion states in a WiMAX network. WiMAX supports high-data rates and multi-service types; hence, it is a strong contender for wireless broadband access technologies to support real-time applications such as VoIP over wireless networks. However, since WiMAX employs best effort (BE) service during the initial phase of deployment, VoIP applications must contend with various types of applications over WiMAX. To maintain bi-directional VoIP communication, then, we need to consider handover triggers indicating both downlink and uplink transmission conditions.

To maintain VoIP communication quality over WiMAX, we proposed a combined use of the following two handover triggers, Carrier to Interference plus Noise Ratio (CINR) and an MS's interface queue length (Niswar et al., 2009b). We first describe the CINR. As described before, RSSI is generally employed as a handover trigger. However, since RSSI provides only signal

strength at a receiver side, it cannot actually detect noise, interference, and other channel effects (e.g., multi-path fading and shadowing). Hence, a high RSSI does not always mean that the wireless link quality is good. On the other hand, as CINR indicates signal effectiveness including noise, it can provide an appropriate index of wireless link quality.

We then investigated the characteristics of CINR and MOS over WiMAX through a simulation experiment. Figure 22(a) depicts the simulation model. In the simulation model, we employ a Rayleigh fading model because we assume an urban environment. For an application, a VoIP application of G.711 that sends a 160-bytes packet every 20 ms is employed. In the simulation, the MS moves away from the base station (BS).

Fig. 22. Simulation model in WiMAX

Figure 23 shows the relationship between CINR and MOS. From the graph, when CINR goes below 50 dB, MOS starts to degrade with dynamic fluctuation. To estimate the CINR's threshold (CINR_th) for initiating handover from the trend of CINR, we employ the solid line by a loess method. The solid line indicates that MOS is 3.6 when CINR is 26 dB. Thus, in this study, we set a CINR of 26 dB as the CINR_th.
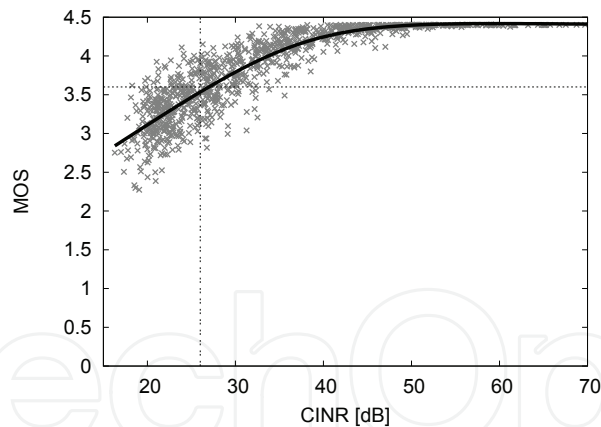

Fig. 23. Relationship between MOS and CINR

In WiMAX, although the bandwidth (BW) scheduler of a BS thoroughly manages downlink BW, uplink BW is allocated as described below. As illustrated in Fig. 24, a BS first tries to allocate BW to MSs through Downlink Map (DL MAP), Downlink Channel Descriptor (DCD), Uplink Channel Descriptor (UCD), and Uplink Map (UL MAP). Then, an MS transmits a BW request in response to the allocated time slot in UL_MAP. After the MS obtains an Uplink (UL) Grant from the BS, it can transmit data packets with the allocated uplink grants. Hence, the end-to-end uplink delay (T_{E2E}) from sending a data packet at an MS to reaching a CS is calculated as follows:

$$T_{E2E} = T_{queue} + T_{BWreq} + T_{Sch} + T_o \qquad (2)$$
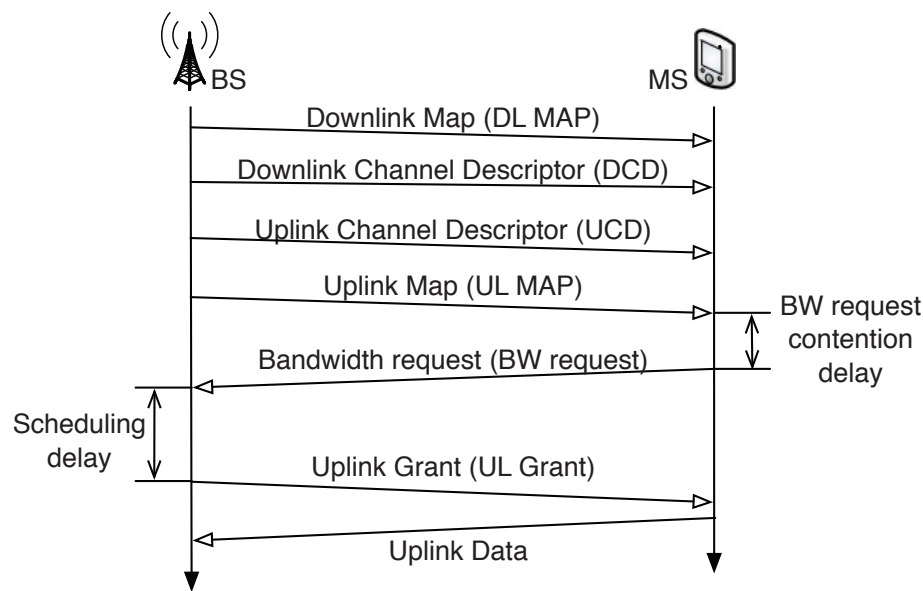
Fig. 24. Uplink transmission process over WiMAX

where T_{queue} represents queuing delay in the interface buffer at an MS. T_{BWreq} and T_{Sch} indicate BW request contention delay and BS's scheduling delay, respectively. T_o represents additional delays, e.g., transmission delays over wired and wireless link. Although the BW request contention and the BS scheduling delay may become longer with the increase in MSs, eventually they lead to the increase of queue length at an MS's interface. Therefore, the MS's interface queue length has a potential to be a handover trigger to detect wireless network congestion in a WiMAX.

We then investigated the relationship between the number of VoIP calls and MS's queue length. As illustrated in Fig. 22(b), up to 30 MSs are randomly located within the coverage area of a BS. Each MS randomly moves at a speed of 1 m/s during VoIP communication with a CS. Figure 25 shows the relationship among the number of VoIP calls, MOS, and MS's queue length. The graph shows that although uplink MOS decreases as the number of VoIP calls increases, downlink MOS is kept at an adequate value. Moreover, the MS's queue length increases as the number of VoIP calls. Therefore, we can see that uplink MOS decreases with the increase of the MS's queue length, which leads to the large queuing delay. Then, in terms of accommodation of VoIP calls in a single BS, Fig. 25 shows that up to 20 VoIP calls can be accepted to maintain appropriate VoIP communication quality. That is, not all VoIP communication quality can be maintained when the MS's queue lengths are more than 12,000 bytes. Thus, we employ 12,000 bytes as a threshold of MS's queue length (QL_th).

5.2 Handover management among different wireless technologies

This section introduces a handover management method in a WLAN-WiMAX scenario (Niswar et al., 2010). The proposed method also implements an HM on the transport layer of an MS like our previous method (Kashihara & Oie, 2007). A multi-homing MS has two wireless interfaces, WLAN and WiMAX. Note that since we assume that each wireless network has a different network address, each wireless interface has a different IP address. Then the proposed handover management method also employs multi-homing, cross-layer architectures, and multi-path transmission mode to maintain VoIP quality during handover.
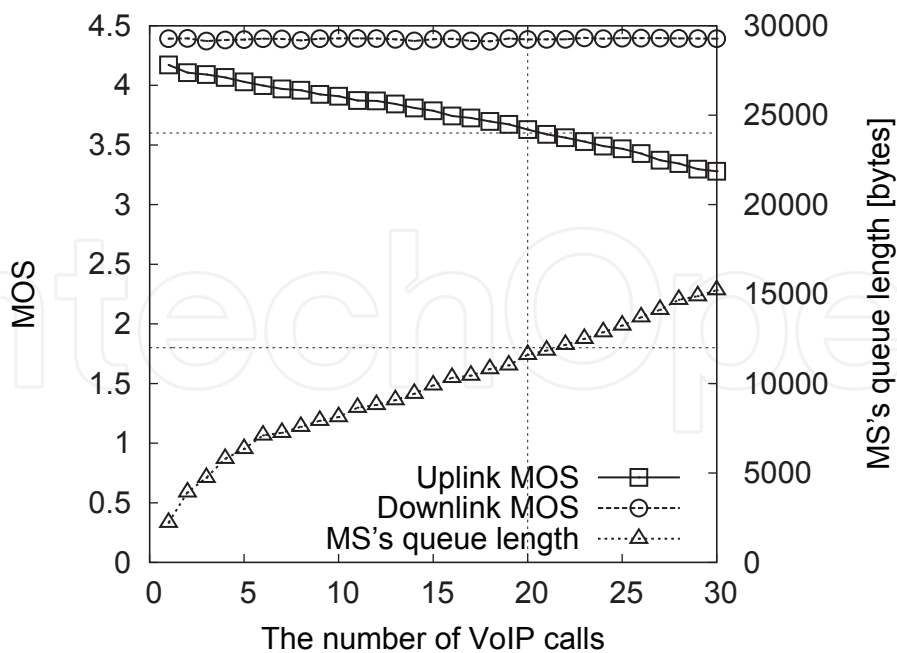
Fig. 25. Relationship among the number of VoIP calls, MOS, and MS's queue length

Figure 26 depicts an algorithm for switching from single-path transmission to multi-path transmission when an MS is in the overlap area of WLAN and WiMAX. Our handover management method first examines the wireless link conditions and then investigates wireless network congestion. For example, when an MS communicates through a WLAN, the HM monitors the RTS frame retry ratio (*retry_ratio*) to detect the wireless link quality at sending of every packet. If the *retry_ratio* exceeds R_Mth of 0.6, the HM switches to multi-path transmission; otherwise, it next examines *WiRTT* to detect the congestion state of the WLAN. If the *WiRTT* exceeds the WiRTT_th of 200 ms, multi-path transmission starts; otherwise, the HM continues single-path transmission via the WLAN. On the other hand, in WiMAX, the HM first monitors *CINR*. If *CINR* is less than the CINR_th of 26 dB, the HM switches to multi-path transmission to maintain the VoIP quality and investigates the condition of both wireless networks; otherwise, it next examines the MS's queue length (*MS's QL*) to detect the congestion state of WiMAX. If the *MS's QL* exceeds the QL_th of 12,000 bytes, the HM switches to multi-path transmission; otherwise it keeps using single-path transmission via the WiMAX.
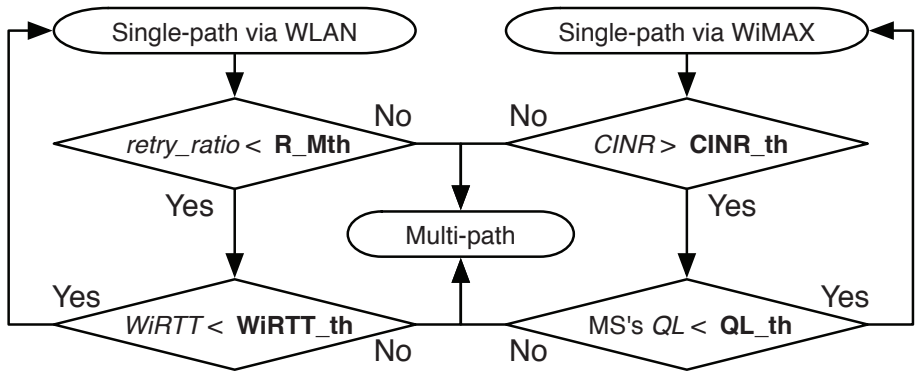

Fig. 26. Switching from single-path to multi-path

When multi-path transmission is applied, the HM monitors all handover triggers and compares them. Figure 27 illustrates a switching process from multi-path transmission to single-path transmission. First, an HM evaluates handover triggers for wireless link quality, i.e., RTS retry ratio (WLAN) and CINR (WiMAX). If the *retry_ratio* exceeds R_Sth as well as the *CINR* is below the CINR_th, the HM continues multi-path transmission because both wireless link qualities are unstable. If the *CINR* exceeds the CINR_th, the HM switches back to single-path transmission via WiMAX. On the other hand, if the *retry_ratio* is below the R_Sth, it switches back to single-path transmission via WLAN. If both the wireless link conditions indicate good conditions at the same time, the HM next examines handover triggers indicating the congestion states of the wireless networks, i.e., *WiRTT* (WLAN) and *MS's QL* (WiMAX). If both the handover triggers exceed the thresholds, the HM continues multi-path transmission. If the *MS's QL* is below the QL_th, the HM switches to single-path transmission via the WiMAX. On the other hand, if the *WiRTT* is below the WiRTT_th, the HM switches to single-path transmission via the WLAN. If both handover triggers are below the thresholds at the same time, the HM returns to single-path transmission via the previous wireless network.
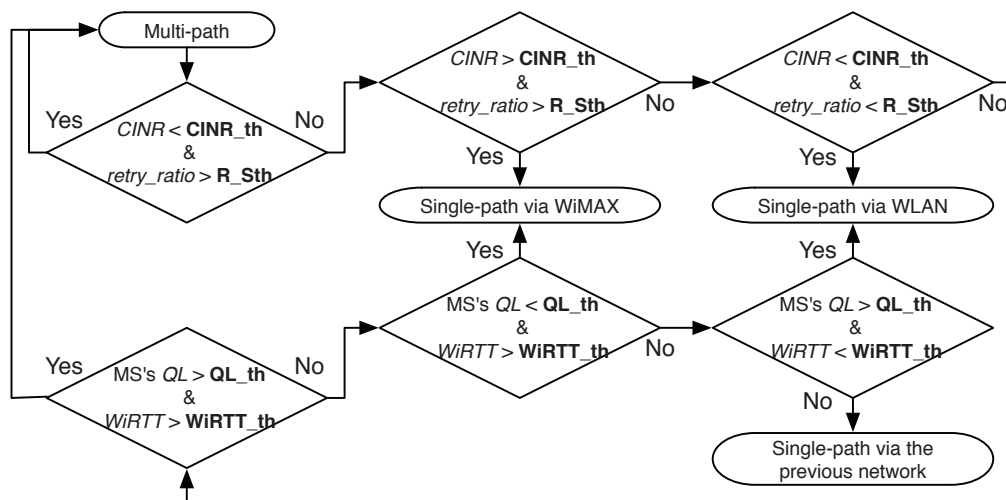

Fig. 27. Switching from multi-path to single-path

5.3 Performance evaluation

This section shows the basic VoIP communication performance of the proposed method through simulation experiments. We first investigated the VoIP communication quality when an MS moves between WLAN and WiMAX. We then evaluated the VoIP communication quality in congested wireless networks.

Figure 28(a) illustrates a simulation model in a movement environment. In the scenario, an MS conducting VoIP communication with a CS moves between WLAN and WiMAX at a speed of 1 m/s. Figure 29 shows that our proposed method can obtain an average uplink MOS of 4.29 and downlink MOS of 4.28 when an MS moves from WLAN to WiMAX. On the other hand, in movement from WiMAX to WLAN, Fig. 30 shows our proposed method can obtain an average uplink MOS of 4.06 and downlink MOS of 4.29.

Figure 28(b) illustrates a simulation model in a congested environment. The simulation experiments have the following two scenarios; one is a congested WLAN, the other is a congested WiMAX. In a congested WLAN scenario, 13 MSs are randomly distributed in the
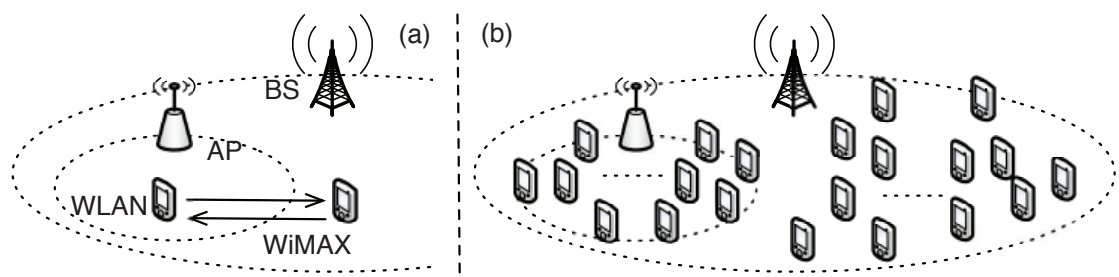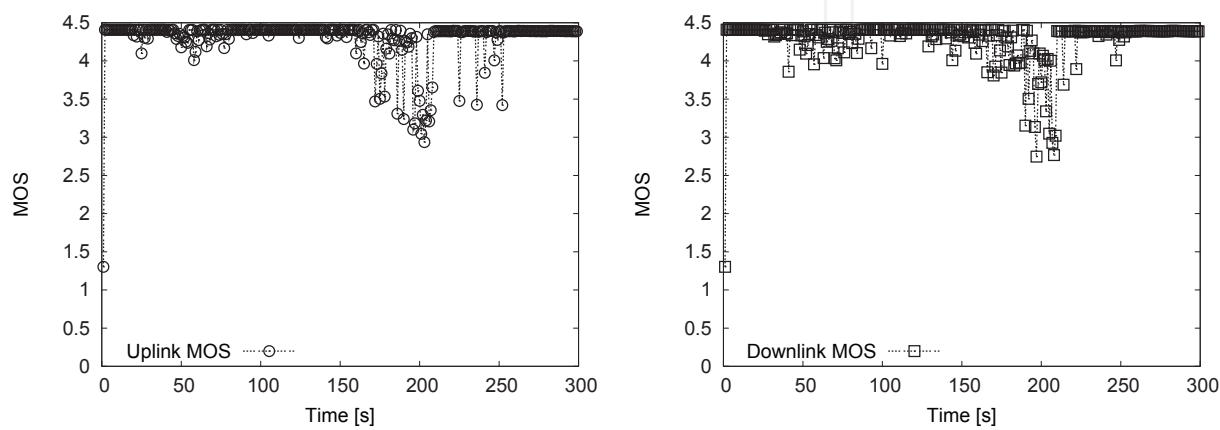
Fig. 28. Simulation model
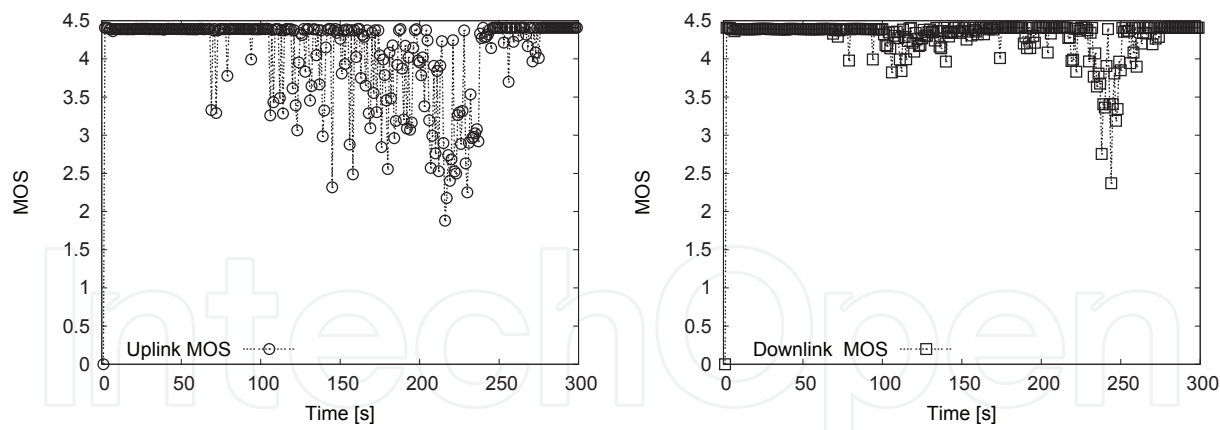

Fig. 29. MOS during movement (from WLAN to WiMAX)

Fig. 30. MOS during movement (from WiMAX to WLAN)

WLAN and no MS is in the WiMAX. Only one MS employs our proposed handover method, and it establishes a VoIP call via the WLAN at the start of the simulation. Then, the remaining MS, which does not employ the proposed method, establishes a VoIP call with a CS every five seconds. That is, the traffic in the WLAN gradually increases. From Fig. 31, the simulation results show that the MS which employs our proposed method obtains the average uplink MOS of 4.26 and downlink MOS of 4.25.

Furthermore, we also evaluated the basic performance of our proposed method in a congested WiMAX as depicted in Fig. 28(b). In the simulation scenario, 30 MSs are randomly distributed
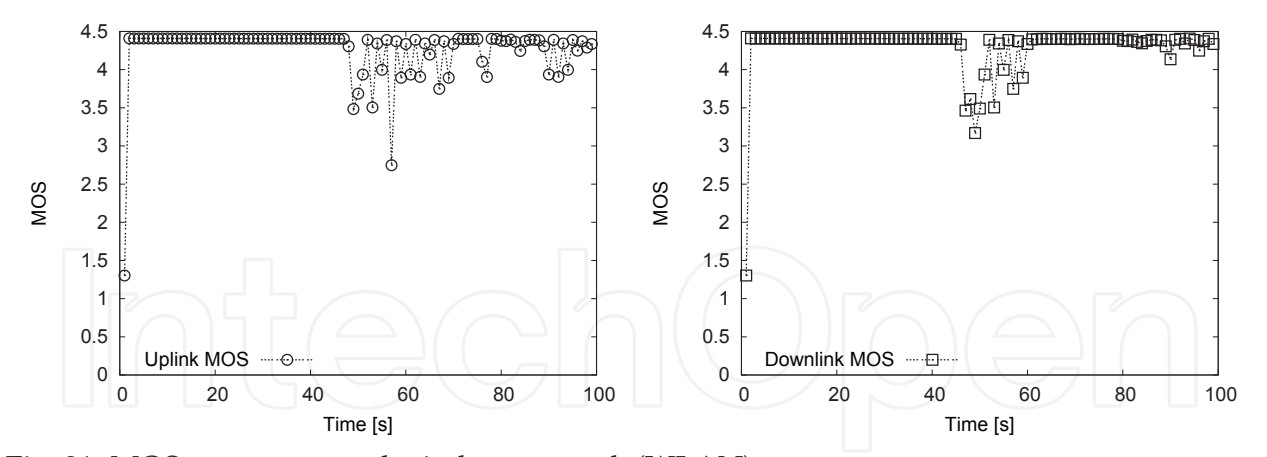
Fig. 31. MOS over congested wireless network (WLAN)

within WiMAX, but no MS is in the WLAN. In this study, since the acceptable number of VoIP calls in the WiMAX is 20 MSs, all VoIP quality is degraded if each MS does not autonomously execute appropriate handover according to the wireless network condition. Here also, only one MS employs our proposed method and it establishes a VoIP call through WiMAX at first. After that, a new VoIP call is established through WiMAX every three seconds. Figure 32 shows that the MS which employs our proposed method obtains the average uplink MOS of 3.88 and downlink MOS of 4.34. Therefore, our proposed method can maintain VoIP communication quality during movement among different types of wireless networks.
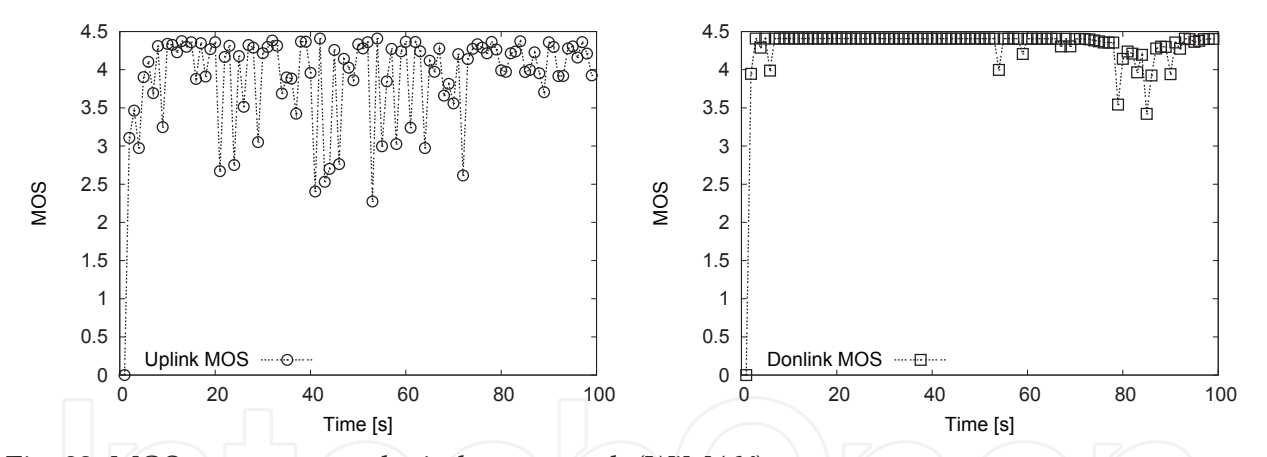

Fig. 32. MOS over congested wireless network (WiMAX)

6. Conclusion

In this chapter, we introduced end-to-end handover management methods for VoIP communication in ubiquitous wireless networks. As described in Section 1, since current and future wireless networks have different network addresses, an MS will need to move among wireless networks while maintaining VoIP communication. To achieve seamless handover among such wireless networks, the following requirements should be satisfied.

1. Keep VoIP communication from communication termination by change of IP address

2. Eliminate communication interruption due to layer 2 and 3 handover processes

3. Initiate appropriate handover based on reliable handover triggers

4. Select a wireless network with good link quality during handover

First, to satisfy requirements (1) and (2), we employed a multi-homing architecture and the HM on the transport layer. A multi-homing architecture is indispensable when moving among wireless networks with different network addresses to avoid communication termination and interruption. On the other hand, the HM can control handovers among the multiple IP addresses on an end-to-end basis, i.e., it needs no special network agent like MIP. Then, to satisfy requirement (3), we employed reliable handover triggers considering VoIP communication quality in WLAN and WiMAX. To maintain VoIP communication quality during movement in ubiquitous wireless networks, we need to consider wireless link quality and congestion states in a wireless network. For wireless link quality, we proposed handover triggers that quickly grasp characteristics of a wireless network, i.e., RTS frame retry ratio in WLAN and CINR in WiMAX. On the other hand, we also proposed handover triggers to detect congestion states in a wireless network, i.e., WiRTT and transmission rate in WLAN, and MS's queue length in WiMAX. The HM can promptly and reliably detect the wireless network condition by using the handover triggers. Finally, to satisfy requirement (4), the HM employed multi-path transmission. When the wireless network condition is degraded, the HM switches to multi-path transmission. Multi-path transmission avoids packet loss during handover while investigating the wireless network condition. Thus, multi-path transmission contributes to achieve seamless handover.

Although this chapter focused on end-to-end handover management, the following problems still must be solved to achieve seamless mobility. First, to execute handover to an AP with a good network condition, an MS needs to locate and connect with a candidate AP with a better network condition among many APs. Although RSSI is commonly employed to select a candidate AP, as described in Section 3.1, RSSI cannot appropriately detect wireless network condition. Actually, we also proposed and implemented an AP selection method to solve this problem (Taenaka et al., 2009), but due to the lack of space here, we cannot describe the details. Moreover, when the number of VoIP calls exceeds the acceptance limit of the wireless networks, all VoIP communication quality degrades. In this situation, the network should not accept a new VoIP call. Thus, to avoid such the degradation, APs and BSs should have an admission control method. Also, our proposed handover methods have no location management function. To manage MSs' location, our proposed method needs to cooperate with some location management functions. For example, we can utilize a dynamic DNS and an overlay network like Skype as network and application level approaches, respectively. Once a VoIP communication is established between an MS and a CS through a location management function, our proposed handover method can maintain VoIP communication during handovers.
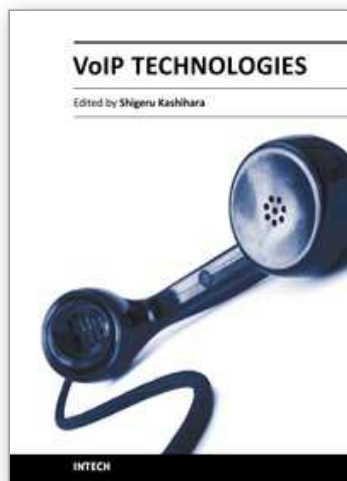
7. Acknowledgements

8. References

Skype Limited. (2003), http://www.skype.com
Perkins, C. (Ed.) (2002). IP Mobility Support for IPv4, IETF RFC 3344
Johnson, D.; Perkins, C. & Arkko, J. (2004). IP Mobility Support for IPv6, IETF RFC 3775
Soliman, H.; Castelluccia, C.; ElMalki, K. & Bellier, L. (2008). Hierarchical Mobile IPv6 (HMIPv6) Mobility Management, IETF RFC 5380

Koodli, R. (Ed.) (2005). Fast Handovers for Mobile IPv6, IETF RFC 4068

Kim, Y.; Kwon, D.; Bae. K. & Suh, Y. (2005). Performance Comparison of Mobile IPv6 and Fast Handovers for Mobile IPv6 over Wireless LANs, Proceedings of IEEE Vehicular Technology Conference 2005-fall (VTC2005-fall), pp. 807-811, September 2005

Montavont, N. & Noel, T. (2003). Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN, Mobile Networks and Applications, Vol. 8, No. 6, pp. 643-653, December 2003

Xing, W.; Karl, H.; Wolisz, A. & Muller, H. (2002). M-SCTP: Design and Prototypical Implementation of an End-to-End Mobility Concept, Proceedings of 5th International Workshop The Internet Challenge: Technology and Application, October 2002

Koga, H.; Haraguchi, H.; Iida, K. & Oie, Y. (2005). A Framework for Network Media Optimization in Multi-homed QoS Networks, Proceedings of ACM First International Workshop on Dynamic Interconnection of Networks (DIN2005), pp. 38-42, September 2005

Stewart, R. (Ed.) (2007). Stream Control Transmission Protocol, IETF RFC 4960

FON wireless Ltd. (2005), http://www.fon.com

Muthukrishnan, K.; Meratnia, N.; Lijding, M.; Koprinkov, G. & Havinga, P. (2006). WLAN location sharing through a privacy observant architecture, Proceedings of 1st International Conference on Communication System Software and Middleware (COMSWARE), pp. 1-10, January 2006

Kashihara, S. & Oie, Y. (2007). Handover management based on the number of data frame retransmissions for VoWLANs, Elsevier Computer Communications, Vol. 30, No. 17, pp. 3257-3269, November 2007

Tsukamoto, K.; Yamaguchi, T.; Kashihara, S. & Oie Y. (2007). Experimental evaluation of decision criteria for WLAN handover: signal strength and frame retransmissions, IEICE Transactions on Communications, Vol. E90-B, No. 12, pp. 3579-3590, December 2007

Proxim Wireless Corporation (2007), http://www.proxim.com

Ethereal (1998), http://www.ethereal.com

Kashihara, S.; Tsukamoto, K. & Oie. Y. (2007) Service-oriented mobility management architecture for seamless handover in ubiquitous networks, IEEE Wireless Communications, Vol. 14, No. 2, pp.28-34, April 2007

Taenaka, Y.; Kashihara, S.; Tsukamoto, K.; Kadobayashi, Y. & Oie, Y. (2007). Design and implementation of cross-layer architecture for seamless VoIP handover, Proceedings of The Third IEEE International Workshop on Heterogeneous Multi-Hop Wireless and Mobile Networks 2007 (IEEE MHWMN'07), October 2007

MadWifi (2004), http://madwifi.org

Bang, S.; Taenaka, Y.; Kashihara, S.; Tsukamoto, K.; Yamaguchi, S. & Oie, Y. (2009). Practical performance evaluation of VoWLAN handover based on frame retries, Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'09), CD-ROM, August 2009.

FreeBSD (1995), http://www.freebsd.org

TCPDUMP/LIBCAP public repository, http://www.tcpdump.org

Scalable Network Technologies (2006), http://www.scalable-networks.com

ITU-T G.107 (2000), The E-model, a computational model for use in transmission planning (ITU-T Recommendation G.107), Telecommunication Standardization Sector of ITU, Series G: Transmission systems and media, digital systems and networks, 2000

Niswar, M.; Kashihara, S.; Tsukamoto K.; Kadobayashi Y. & Yamaguchi S. (2009a). Handover management for VoWLAN based on estimation of AP queue length and frame retries, IEICE Transactions on Information and System, Vol. E92-D, No. 10, pp. 1847-1856, October 2009

Niswar, M.; Kashihara, S.; Taenaka, Y.; Tsukamoto, K.; Kadobayashi, Y. & Yamaguchi, S. (2009b). MS-initiated handover decision criteria for VoIP over IEEE 802.16e, Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'09), CD-ROM, August 2009

Niswar, M.; Kashihara, S.; Taenaka, Y.; Tsukamoto, K.; Kadobayashi, Y. & Yamaguchi, S. (2010). Seamless vertical handover management for VoIP over intermingled IEEE 802.11g and IEEE 802.16e, Proceeding of 8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2010), CD-ROM, June 2010

Taenaka, Y.; Kashihara, S.; Tsukamoto, K.; Yamaguchi, S. & Oie, Y. (2009). Proactive AP selection method considering the radio interference environment, IEICE Transactions on Information and System, Vol. E92-D, No. 10, pp. 1867-1876, October 2009

VoIP Technologies

Edited by Dr Shigeru Kashihara

This book provides a collection of 15 excellent studies of Voice over IP (VoIP) technologies. While VoIP is undoubtedly a powerful and innovative communication tool for everyone, voice communication over the Internet is inherently less reliable than the public switched telephone network, because the Internet functions as a best-effort network without Quality of Service guarantee and voice data cannot be retransmitted. This book introduces research strategies that address various issues with the aim of enhancing VoIP quality. We hope that you will enjoy reading these diverse studies, and that the book will provide you with a lot of useful information about current VoIP technology research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Shigeru Kashihara, Muhammad Niswar, Yuzo Taenaka, Kazuya Tsukamoto, Suguru Yamaguchi and Yuji Oie (2011). End-to-End Handover Management for VoIP Communications in Ubiquitous Wireless Networks, VoIP Technologies, Dr Shigeru Kashihara (Ed.), ISBN: 978-953-307-549-5, InTech, Available from: http://www.intechopen.com/books/voip-technologies/end-to-end-handover-management-for-voip-communications-in-ubiquitous-wireless-networks

INTECH
open science | open minds