Wireless Network Synchronization for Multichannel Multimedia Services

Yonghwan Bang[1], Jongpil Han[1], Kyusang Lee[1], Jongwon Yoon[2], Jinoo Joung[2], Sungbo Yang[3], and June-Koo Kevin Rhee[1]

[1]*School of Engineering., Information and Communications University, Daejeon, S.Korea.*

[2]*Computer Science., Sangmyung University, Seoul, S.Korea.*

[3]*Advanced Information Research Team., Hyundai Motor Co., Seoul, S.Korea.*

{peha, cbr600f4, sangi}@icu.ac.kr, {navigater, jjoung}@smu.ac.kr, sbyang530@hyundai-motor.com, rhee.jk@icu.ac.kr

***Abstract* — When a single source of multimedia contents is distributed to multiple reproduction devices, the audio and video contents require synchronous play for multi-channel stereo sound and lip-synchronization. This paper investigates capability of IEEE 802.11n wireless LANs for providing synchronized services of real-time multimedia traffic. We implement IEEE 1588 Precision Time Protocol in wireless LAN devices to characterize delay jitter and clock synchronization accuracy. The results indicate a strong potential to provide a high fidelity real-time multi-channel multimedia networking service within 430-microsecond synchronization accuracy at an approximately 80 Mbps streaming rate.**

***Keywords* — carrier sense multi-access, multimedia communication, synchronization, wireless LAN.**

1. Introduction

Broad band networking has become more widely supported by wireless local area network (WLAN) technologies, especially by the IEEE 802.11 family [1]. Hence, WLAN is expected to develop their applications for multimedia contents distribution for multi-device audio and video playback. Digital contents distribution over a network can reproduce high quality audio and video contents [2]. For example, the available bandwidth of IEEE802.11n to the UDP (user datagram protocol) client is as high as 90 Mbps, which can transmit several channels of high definition television (HDTV) contents by use of MPEG2 and MEPG4 formats. Some time-critical multimedia applications adopt JPEG2000 for real-time playback.

A critical issue of high-fidelity digital contents playback is uncertainty in the play timing, for applications where audio and video devices are separated in space and thus digital reproduction of stereo sound and moving pictures are provided at different devices with respect to different time references. In order to achieve such high fidelity digital multimedia playback, each device has to provide an accurate clock reference to the playback applications. Such time reference can be used for a two-fold benefit: to control packet transmissions at a constant bit rate (CBR) in plesiochronous manner and to control playback time of the audio and video applications.

Time and rate-guaranteed packet transmission over wireless LAN (WLAN) is a challenging technology because the air channel is very unreliable transmission medium especially when CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) is used. Network synchronization is implemented in IEEE802.11 inherently. In a typical WLAN configuration, an access point (AP) broadcast time information in its beacon frames to its subordinate stations (STA). In this way, STAs can be tuned to the AP within an accuracy of a few microseconds. However, current IEEE 802.11 standard STA devices do not provide such time information to its client applications. There are several prior works that report timing synchronization performance of IEEE 802.11 and IEEE 1588 [3]. However, performance analysis under high traffic volume, such as for multimedia streaming, has not been reported.

In this paper, we focus how accurate network synchronization can be attained from commercially available standard IEEE802.11n STA devices for multimedia services, when the synchronization is achieved by clients of the network. Commercially available wireless routers are used to implement time synchronization by the use of IEEE 1588 precision time protocol (PTP) [4][5][6]. Implementing PTP functions in Linux-based routing processes of WLAN devices, we achieve clock synchronization based on measurements of clock difference (offset) and delay time between a pair of an AP (as a clock master) and an STA (as a clock slave). Statistic filtering is applied to increase accuracy of offset estimations at STAs. All implementation is accomplished without requiring a new hardware [7]; the timer for synchronization is borrowed from the Linux processor clock. Background traffic up to 72 Mbps is applied to make sure *PTP measurements are accurate and acceptable under multimedia streaming with high traffic volume.* Our results suggest that IEEE 802.11n is a very strong candidate for high-fidelity wireless multimedia service with a typical maximum service throughput of 80 Mbps and a delay jitter less 430μsec, which will be possible to support stereo sound reproduction.

This paper in the following sections discusses, first, the general requirement for distributed high fidelity multimedia playback system, second, a general IEEE802.11n system

This work at ICU was supported in part by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-313-D00685).

implementation model using a commercially available product model, third, IEEE802.11n channel characteristics investigation, and last, the clock synchronization performance using IEEE1588 PTP and statistical filtering of offset determination. Ref. [8] been similar study report. However, most previous reports overlook the impact of high traffic volumes and the productization boundaries of currently available standard IEEE 802.11 chipsets. In this paper, we address such practical issues for multimedia applications.

2. Wireless Multimedia Application Model

A. Wireless multimedia playback system

Multimedia contents become more complicated in order to reproduce more vivid video and sound to deliver a high quality of experience (QoE). High fidelity multimedia contents now can produce high definition television (HDTV) motion pictures and multichannel stereo sound. The bandwidth limit against the contents quality has been lifted off greatly by the use of digital technologies such as compression and coding. Recently the major progresses are made with MPEG4, H.264, and JPEG2000. These technologies provide a great means of storing and transporting the high fidelity digital multimedia contents. However, the performance of decoding, decompression, and playback process is somewhat not deterministic, so the contents in multiple channels may not be played in a harmonious way in time, limiting the performance of QoE. A typical wireless multichannel multimedia playback system is illustrates in Figure 1.

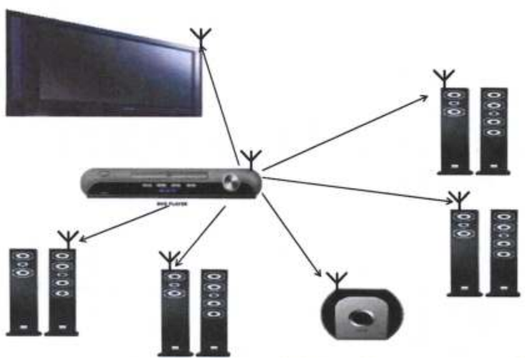

Figure 1. A wireless multichannel multimedia playback system model

In the multimedia system model, each audio or video contents is streamed in a separate channel, thus requiring streaming synchronization. Both the network streaming and playback application need synchronized timer at every device. Such synchronization can be attained by the use of IEEE 1588 in the case the time delay uncertainty with respect to the time stamp position is not negligible [9].

B. Contents synchronization requirements

When multimedia contents are played in a distributed multi-channel multi-device system, audio and video contents must be reproduced synchronously, to perform lip synchronization and stereo sound effect [10][11]. Early studies have reported on multimedia multichannel streaming synchronization requirements as shown in Table 1.

Table 1. Multimedia synchronization requirements

Contents	Inter-Stream Synchronization Requirement	Reference
Tightly coupled audio	± 10 μs	[12]
Real-time audio	2 ms	IEEE802.1AVB [13]
Lip synchronization	Sound should not lead by 15ms Sound should not lag by 45ms	ATSC [14]
Video animation	± 80 ms	IEEE802.1AVB [13]

3. Wireless LAN Synchronization Implementation

In general, MPU clocks and hence the kernel timers are free running independently causing timer skews between different device units. Time synchronization compensates such timer skew by measuring offset and compensates timers for the offset.

To implement a time-synchronization module in IEEE 802.11n, there are following possible considerations that are wireless communications characteristics such as CDMA/CA, automatic retransmission with a random back-off delay, packet loss and so on. We analysis these characteristics and adopt a filter algorithm. We find out the value of skewed rate at the independent device by monitoring two devices' timer information without synchronization. As result, at no traffic area, the timer skew rate is about 1.6 ppm.

We adopt IEEE 1588 precise time protocol (PTP) as the base line of our research. PTP exchanges time stamping information that denotes a PTP control message's arrival or transmission time. In order to know these time stamp information, it needs a time-stamping mechanism. Application layer time stamping (ATS) approach stamps a time mark at application layer. In ATS approach, it includes uncertainties of processing delay time or process scheduling delay when measures a timer offset between two devices.

To achieve more precise time synchronization in PTP, we need to know the actual timer value when a packet is transmitted or arrived. But typically, commercial wireless AP devices or MAC chips do not support these functions and moreover, we cannot modify them. For these reasons, we implement a time-stamping module at the device driver layer as close as possible to MAC or PHY level time-stamping. We called it device driver layer time stamping (DTS). By this approach, we can minimize the processing delay to measure the timer offset value between two devices at existing commercial AP devices without any PHY or MAC modifications. We modify device's kernel to implement DTS. Figure 2 shows concept of ATS and our approach, DTS.

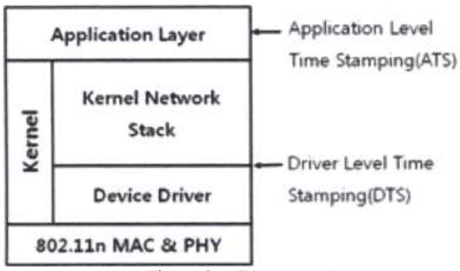

Figure 2. ATS and DTS

ATS stamps the time information of PTP messages at the application layer and the DTS stamps at the device driver layer. For example, master sync device saves a SYNC message's transmission time stamp information when a SYNC message is sent to the device driver from the network layer. At that time, the DTS module saves current time information as actual transmission time of SYNC message, and later, kernel informs to application layer when an application sync module requests. At the slave side, when the SYNC message is received by slave's wireless interface, it is transferred from device driver to network layer. At that time, DTS module detects the PTP message and saves current time information as SYNC arrival time at the kernel. And also this information is used at the application sync module by a system call request.

We use a Linksys wrt350n AP network device as a mobile device. It supports full capabilities of IEEE 802.11n. We measure actual bandwidth of this AP device. We find out the average performance of throughput of this device is about 80 Mbps.

We modify a DD-WRT firmware which is a third party developed firmware released under the terms of the GPL for IEEE 802.11a/b/g/h/n wireless routers mainly based on a Linksys devices [16].

Using Linksys wrt350n device and DD-WRT firmware, we implement synchronization module at the commercial network device which is available for typical purpose.

4. Timer synchronization performance

Using IEEE 1588 PTP messaging, we measure the timer skew between an AP and STA device pair. Since the time stamps are based on micro processor unit (MPU) clock, the time stamping references are limited to the MPU clock accuracy and the embedded Linux kernel performance. The MPU clock stability is estimated to be approximately 1.6ppm. This translates approximately 100μs per minute.

Time stamping accuracy is first investigated in order to understand the IEEE 802.11n device characteristics. The aforementioned two time-stamping functions in the application layer and in the kernel device driver layer are used to compare to see if the application layer implementation is acceptable. IEEE 1588 PTP gives a means of measuring time difference of the timers at two devices. At the slave device implemented at the STA, the measured time offsets are used to correct the local timer value. However, a typical transmission delay performance of a IEEE 802.11n channel is very nondeterministic due to the nature of CSMA/CA, and automatic retransmission on frame losses. In addition, the MPU kernel and application process uncertainty adds up the delay uncertainty in the frame transmission. Such time stamping uncertainty and delay jitter affects the timer offset measurement between two devices, resulting in measurement errors. In order to minimize the impact of such measurement error, we introduce statistical filtering before applying timer compensation at the slave STA device. The statistical filtering is accomplished by the following steps:

1. Conduct n number of IEEE 1588 PTP offset measurements. Number n is widely varied between 1 and 100.
2. Calculate the mean μ_0 and standard deviation σ_0 of the n offset samples.
3. Delete offset samples outside an interval defined by the boundaries at $\mu_0 \pm \beta\ \sigma_0$.
4. Calculate a new mean μ with filtered samples.

We discuss the choice of n for estimation of synchronization accuracy, we use the following metric:

Synchronization accuracy estimation: calculate the offset standard deviation δ of m's after a large number of repetitions of Steps 1 to 4 in the above.

The filter parameters n and β is researched in a wide range of possible choice. We find empirically optimized value in the range of $1 \leq \beta \leq 3$, and we choose $\beta = 1$ as it is not a sensitive value against the offset standard deviation performance metric. The average sample size n is chosen carefully according to the following experimental results.

In our characterization experiments, background traffic is added to the air medium in order to investigate the impact of extensive traffic. The experimental set-up is shown in Figure 3.

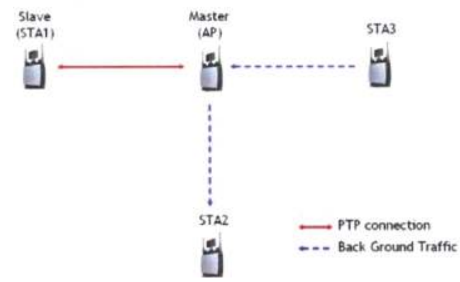

Figure 3. Experimental set up to characterize IEEE 802.11n device synchronization performance under various background traffics

A. Application layer time stamping

As discussed in the previous section, application layer time stamping is the simplest form of implementing IEEE 1588 PTP. In this case, the propagation delay consists of (Figure 2).

1. application scheduling delay by the kernel at the master,
2. kernel process scheduling delay at the master,
3. MAC access delay at the master,
4. PHY and air propagation delay,
5. MAC access delay at the slave,
6. kernel process scheduling delay at the slave, and
7. application scheduling delay by the kernel at the slave.

In this regard, the nondeterministic process of automatic retransmission by IEEE 802.11 is not included, as most of such cases are eliminated by the aforementioned statistical filtering. The synchronization accuracy is measured under various traffic load conditions. The corresponding result is presented in Figure 4. The solid curve labeled as SS&C is a result from the synchronization compensation scheme without statistical filtering. The rest data are taken with different *n* values for averaging defined by the statistical filtering. There are two important features to understand from this data: First, statistical filtering is highly effective, which can improve the performance by an order of magnitude. Second, the impact of background traffic causes a large penalty, so the worst synchronization accuracy is 210ms.

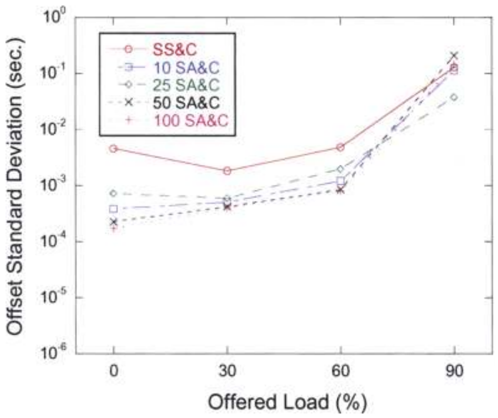

Figure 4. Offset standard deviation measure after 100 IEEE 1588 messages and compensations with application layer time stamping implementation. SS&C: single sample and compensation and *n* SA&C: *n*-sample averaging and compensation

B. Device driver layer time stamping

The large penalty observed in the previous section can be mitigated if time stamping is moved closer to the transmission and reception points. As discussed in the previous sections, device driver layer stamping can eliminate time delay jitter in the kernel's application scheduling. We observe that this is a critical improvement when the traffic is highly offered. In this case, the propagation delay consists of

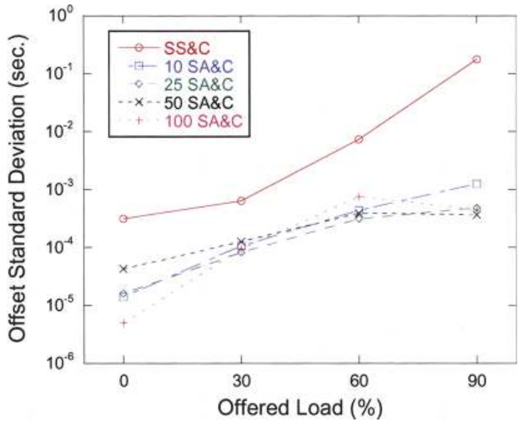

Figure 5. Offset standard deviation measure after 100 IEEE 1588 messages and compensations with device driver layer time stamping implementation

1. kernel process scheduling delay at the master,
2. MAC access delay at the master,
3. PHY and air propagation delay,
4. MAC access delay at the slave, and
5. kernel process scheduling delay at the slave.

The corresponding result under various traffic load conditions is presented in Figure 5. There are two important observations in this data. First, overall synchronization accuracy is improved by at least an order of magnitude. This implies that the application scheduling delay is the main bottleneck in attaining time stamping function in a commercially available IEEE 802.11n device on an embedded Linux platform. Second, statistical filtering is highly effective, and even more efficient algorithm is anticipated to enhance the synchronization accuracy under a large traffic volume. Third, at light volume this system can support tightly coupled audio applications with about 50µs accuracy as well as all other services. In our system the DST scheme performs the worst-case synchronization accuracy of 0.36ms.

The optimal number for average sample size *n* is empirically selected by rearranging Figure 5 into Figure 6. As the average sample size increases the overall trend show reduction of offset jitter δ. However, too large value of *n* requires a long intervals between compensations, thus the performance can get worse. For example in our system, the IEEE 1588 messaging interval is chosen to be 10ms. An average sample size *n* =100 results in a synchronization update interval of 1 s, which may be too long in a fast varying traffic conditions. In Figure 6, the value *n* between 25 and 100 shows nearly constant performance. In our filtering scheme *n*=25 or 50 can be chosen as an optimum.

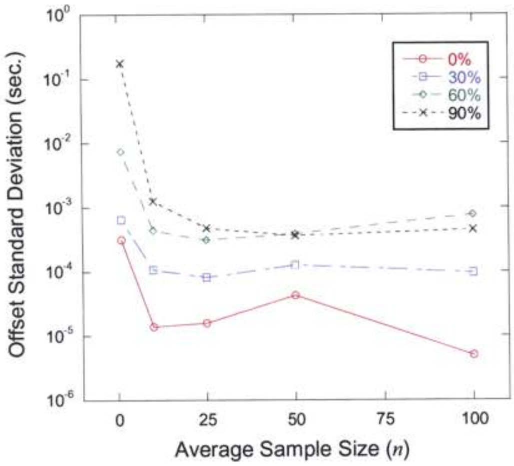

Figure 6. Offset standard deviation measure after 100 IEEE 1588 messages and compensations with device driver layer time stamping implementation as a function of average sample size *n*

5. Conclusions

Multichannel multimedia wireless service feasibility is studied based on IEEE 802.11n and IEEE 1588 technologies. We investigate the time synchronization accuracy in the proposed system to report a potential of multichannel wireless service. When the network is lightly load the system can provide tightly couple audio reproduction, such as stereo sound, with about 50μs timing accuracy in the network layer. At the high traffic limit up to 72 Mbps application throughput, the timing accuracy of 430μs can be provided when the time stamping is achieve at the interface of MAC/PHY hardware by a device driver level implementation.

REFERENCES

[1] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std. 802.11,* June 2007.

[2] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Draft 3: *Enhancements for Higher Throughput* 2007.

[3] T. Cooklev, J. C. Eidson, and A. Pakdaman, "An Implementation of IEEE 1588 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes," IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 5, October 2007.

[4] IEEE, Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2002,* 2002.

[5] A. Vallat and D. Schneuwly, "Clock Synchronization in Telecommunications via PTP (IEEE 1588), " *Frequency Control Symposium, 2007 Joint with the 21st European Frequency and Time Forum. IEEE International,* May 29 2007-June 1 2007, pp.334-341.

[6] J. Edison, J. Mackay, G.M. Garner and V. Skendizicm, "Provision of Precise Timing via IEEE 1588 Application Interfaces", *IEEE symposium on Precision Clock Synchronization (ISPCS) for Measurement, Austria Oct.* 1-3, 2007.

[7] K. Correll, N. Barendt, and M. Branicky, "Design considerations for software-only implementations of the IEEE 1588 Precision Time Protocol, *Porc. Conference on IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,* NIST and IEEE. Winterthur WSITZERLAND, October 10-12, 2005.

[8] K. Arvind, "Probabilistic Clock Synchronization in Distributed Systems," *IEEE Transactions on Parallel and Distributed Systems,* May 1994, pp.474-487.

[9] R. Steinmentz and T. Meyer, "Multimedia Synchronization Techniques: Experiences used on Different System Structures", Multimedia Communications, 1992. *MULTIMEDIA '92 4th IEEE ComSoc International Workshop on,* 1-4 Apr 1992.

[10] K. Wang, M. Faulkner, and Tolochko, "Timing Synchronization for 802.11a WLANs under Multipath Channels, "*Proceedings on Australian Telecommunications, 2003.*

[11] F. Cristian and C. Fetzer, "The Timed Asynchronous Distributed System Model," *IEEE Trans. On Paralleland Distributed Systems,* 10(6):642-657, June 1999.

[12] R. Steinmetz, "Human perception of jitter and media synchronization", *IEEE Journal on Selected Areas in Communications, VOL. 14, NO. 1,* JANUARY 1996.

[13] G. M. Garner, *et al.,* "IEEE 802.1 AVB and Its Application in Carrier-Grade Ethernet", IEEE Communications Magazine, December 2007.

[14] ATSC Implementation Subcommittee Finding: Relative Timing of Sound and Vision for Broadcast Operations, ATSC, Doc. IS-191, 26 June, 2003.

[15] S. Vuong, K. Cooper, and M. Ito, "Petri Net Models for Describing Multimedia Synchronization Requirements," *International Conference Proceedings on Network Protocols,* 1995 7-10 Nov. 1995

[16] http://dd-wrt.com/