공학석사학위논문

스마트 무선공유기를 이용한 와이파이 성능 향상

Improving Wi-Fi Performance using Smart Access Point

2017년 2월

서울대학교 대학원
컴퓨터공학부

김 기 석

스마트 무선공유기를 이용한 와이파이 성능 향상

Improving Wi-Fi Performance using Smart Access Point

지도교수 김 종 권

이 논문을 공학석사학위논문으로 제출함

2017년 2월

서울대학교 대학원

컴퓨터공학부

김 기 석

김기석의 석사학위논문을 인준함

2017년 2월

위 원 장	전 화 숙	(인)
부위원장	김 종 권	(인)
위 원	권 태 경	(인)

ABSTRACT

Improving Wi-Fi Performance using Smart Access Point

Kiseog Kim
Dept. of Computer Science and Engineering
The Graduate School
Seoul National University

Wireless network is constantly expanding due to their superior accessibility and convenience when compared to wired networks. In the meanwhile, as physically limited bandwidth of wireless networks becomes increasingly congested, it is suffering from performance degradation. Although TCP is widely adopted regardless of wired or wireless networks, it is fundamentally designed and optimized for wired networks. Furthermore, recent advances of combo-modules that enable two or more network interfaces to share an antenna introduce a new challenge of Cross Technology Interference (CTI). In this paper, we identified redundant acknowledgement process across multiple layers during

TCP transmission, and investigated root causes of interference pattern in combo-modules which induce transmission failures. Then we suggest two novel AP-side solutions called I-ACK (Implicit Acknowledgement) and Smart AP. I-ACK eliminates overhead caused by redundancy in TCP transmission procedure, and Smart AP stands strong against malfunction of combo-modules. We implemented these schemes on off-the-shelf AP and demonstrated the effectiveness in an anechoic chamber where external noise is blocked.

Keywords: Wi-Fi, Access Point, TCP, Cross-Layer Approach, Combo-modules, Power Saving Mode, Coexistence

Student Number: 2015-21228

CONTENTS

LIST OF FIGURES

Chapter Ⅰ

Introduction

1.1 Background

Recent technical innovations in the fields of IT, such as VR (Virtual Reality) and IoT (Internet of Things), are greatly affecting and improving our daily lives and changing the way people interact with things around them. Underneath such innovations, wireless communication stands as a strong foundation. And prominence of mobile devices inevitably brought about the congestion issue in wireless networks and caused transmission performance to fall. The congestion problem is especially severe on 2.4 GHz ISM (Industrial, Scientific and Medical) band, which bears Wi-Fi, Bluetooth, Zigbee and other heterogeneous interfaces. Therefore, mobile devices in public places compete with each other to occupy a medium, but fail to achieve enough bandwidth due to high congestion. Apple i-Phone's Wi-Fi connection failure during its demo in WWDC 2010 is a representative case which shows problems of the crowded ISM band [2].

To overcome the vulnerability on occurrences of transmission failure, wireless networks adopted Link Layer Acknowledgement (LL ACK). It shares some redundant processes with reliable transport protocols like TCP (Transmission Control Protocol). Main difference between LL ACK and TCP ACK lies on response time and medium acquisition overhead. While LL ACKs are usually sent right after 10 microseconds, TCP ACKs take at least few milliseconds to be sent, since they are transmitted after the packets are successfully processed by a receiver. It is mainly because LL ACKs are treated as control frames and sent right after SIFS (Short InterFrame Space), thus inducing no medium acquisition overheads. On the other hand, TCP ACKs are considered as data frames and sent at least after DIFS (Distributed coordination function InterFrame Space). Computing power of mobile devices has improved and the case where packets that arrived at the link layer fail to reach the transport layer has been obscured. In other words, although reliability of the LL ACK is now as high as that of TCP ACK, time spent on the medium is still as it is and not improved at all.

Efforts to improve TCP performance in wireless networks were steadily made. HACK [9] notably simplified transmission process, by attaching TCP ACK in the following block ACK to reduce medium acquisition overhead. SNOOP [11] improved the performance by making TCP retransmissions play on APs, which were originally managed by servers. But these methods require modifications on mobile clients and still have leeway to simplify transmission process and achieve performance gain.

Research interests also lie on coexistence problem of combo-modules, where two or more wireless interfaces share a single antenna. Especially, combo-modules that support Wi-Fi and Bluetooth, the two most popular technologies in mobile devices, are widely spread. These days, there are many use cases where these two interfaces are used simultaneously (e.g. A user enjoys movies streamed over Wi-Fi and listens to the sound through Bluetooth earphones). To make it work, Wi-Fi and Bluetooth in a combo-module shares a medium in TDM (Time Devision Multiplexing) manner. In use cases, Wi-Fi enters to Sleep Mode when a combo-module serves Bluetooth, and Wi-Fi transits back to Wakeup Mode when it should take the medium. To be specific, Wi-Fi notifies an AP of its Sleep Mode by borrowing Wi-Fi's existing Power Saving Mode (PSM) mechanism in IEEE 802.11s [10] and of its Wakeup Mode by sending any data. Because the audio streaming in Bluetooth requires real-time transmission, Bluetooth should be ensured to have enough time to send data, constantly. We have experimentally confirmed that combo-modules schedule Bluetooth service time every 30~50ms and Sleep/Wakeup transition happens along with it. Mobile devices notifies an AP of PSM information by transmitting null function data frames with a power management bit, but they are often dropped due to well-known hidden terminal problem and other heterogeneous interference issues. Then, the AP fails to catch up with the status of a mobile device in sleep mode and experiences consecutive transmission failures. These unnecessary data transmission degrades not only the performance of the target combo-module device but also that of the whole network it belongs to.

1.2 Goal and Contribution

This dissertation proposes two AP-side mechanisms to enhance TCP performance and relieve congestion issue. First, we devised the Implicit ACK (I-ACK). When an AP receives a LL ACK from wireless devices, it creates a TCP ACK and send it to the server. Time taken by wireless devices to process packets and generate TCP ACKs is saved, and hence, it reduces the transmission process and RTT (Round-trip Time), and improve throughput. Second, we made an AP to recognize devices equipped with combo-modules that support two interfaces with a single chip. Then, the AP observes PSM cycle and refrain sending data to the mobile device at the expected starting time of the sleep mode. This prevents unnecessary transmission errors that occur when an AP fails to receive control frames and has no idea that the mobile device is in sleep mode, and moreover reduces the occurrence of Bluetooth retransmission.

We implemented the proposed schemes on Linux with the OpenWrt platforms and used TP-Link AC1750 AP. Target Linux kernel version was 3.18.23 and we modified Ath9k driver on Chaos Calmer 15.05 OpenWrt.

The main contributions of this dissertation are as follows.

- We analyzed rigorously the performance degradation problems occurred in the real world use cases and discovered root causes.
- We proposed several novel schemes that remedy the complex Wi-Fi congestion problem and enable efficient and fair sharing of the spectrum.

- A rigorous performance study under controlled environments showed the effectiveness of the proposed schemes.

1.3 Thesis Organization

The rest of the dissertation is organized as follows. In Section 2, prior studies on Wi-Fi congestion and combo-module problems are summarized. Then, we introduce TCP data transmission process and examine root causes of performance degradation in Section 3. AP-side solutions and actual implementation methodology are detailed in Section 4. The performance of the solutions with actual testbed are given in Section 5 followed by concluding remarks in Section 6.

Chapter Ⅱ

Related Work

2.1 Wi-Fi Performance Issue

TCP is the de facto standard for Internet-based commercial communication networks [15]. And it is designed solely for wired communication environments, where transmission errors are not frequent. Hence, TCP assumes implicitly that any packet loss is due to network congestion. But wireless networks are a little different, so several methods to improve performance of TCP in presence of transmission errors were proposed [11, 12, 13, 14]. The first approach isolates different types of networks. Wireless communications have greater chance of facing transmission failures compared to wired communications. Under this circumstance, wired communications follows original TCP and wireless communications exploits intermediate routers to solve problems. This type of approach includes Indirect TCP (I-TCP) [12], SNOOP [11]. In addition to that, TCP SACK [13], one of reactive congestion control methods, and TCP VEGAS [14], one of proactive congestion control, can be applied in a wireless network.

Efforts to improve the performance of Wi-Fi has been made in link layer. FICA [8], which divides a channel into small subchannels to exploit and attempt efficient data transmission, is representative. Approaches to combine link layer and transport layer solutions were taken as in HACK [9]. Although HACK takes a similar approach with I-ACK in the way it uses LL ACK as TCP ACK, they are fundamentally different in that HACK operates on client-side and it generates complicated packets composed of both LL ACK and TCP ACK.

2.2 Wi-Fi/Bluetooth Coexistence Issue

Prior work on Wi-Fi and Bluetooth coexistence assumed different use cases and they can be classified into three categories. The first in network-level coexistence [5, 6], where Wi-Fi and Bluetooth interfaces are located in separate devices. Specifically, Cabral and Lins [1] showed that a Wi-Fi devices suffer from collisions due to low-powered Bluetooth signal via a simple analysis. In [5, 6], authors proposed a packet OverLap Avoidance (OLA) scheme assuming Wi-Fi and Bluetooth can sense each other's transmissions. The assumption of mutual equal detection cannot be applied to a typical use scenario because Bluetooth signal is too week and the lack of CSMA in Bluetooth.

The second category is collocated coexistence. In this scenario, a Wi-Fi interface and a Bluetooth module are positioned in the same device. This use case can be commonly observed in many of laptops and tablet PCs of these days. Even if two interfaces are collocated in one device, Wi-Fi and Bluetooth can transmit and receive simultaneously. The challenge here is how to deal with self-talk interface that occurs when Bluetooth receives while Wi-Fi is transmitting or vice versa. Since the two interfaces are located in proximity, interference from one while the other is receiving overwhelms the desired signal. On collocated devices, an adverse phenomenon called “Avalanche Effect” was reported [7]. When a Wi-Fi packet collide with a Bluetooth transmission, the sender of the Wi-Fi packet will attempt to retransmit with lower rate, which leads to longer transmission time. The lengthened packet will have higher chance of being interrupted by Bluetooth transmissions. This loop-like effect is called Avalanche Effect and a remedy exploiting RTS/CTS (Request-To-Send/Clear-To-Send) was proposed in the paper.

As mentioned in the introduction, there are many combo-modules having both Wi-Fi and Bluetooth on a single SoC (System on Chip) circuit and sharing a single antenna. Most of the previous work addressing the coexistence among Wi-Fi and Bluetooth does not consider this third cases of Wi-Fi and Bluetooth coexistence. The challenge here is to ensure proper protocol operations while efficiently sharing a common antenna. Since the two protocols need to share the antenna and frequency bandwidth in TDM manner, many conventional medium access schemes assuming separate interfaces may not be directly applicable. There are a few papers [3, 16] mentioning the possibility of the combo scenario and they only suggest a naïve packet prioritization scheme based on the scheduling specification of Bluetooth.

Chapter Ⅲ

Understanding The Performance of Mobile Devices

3.1 TCP Data Transmission in Wireless Networks

TCP is designed as a reliable transport protocol. For this reason, TCP ACK is an essential element for successful transmissions. A TCP server, which is responsible for a transmission, sends a data packet and keeps it until the server receives a corresponding TCP ACK. Only after right reception of the TCP ACK, the server can delete the packet and send the next one. Meanwhile, a transmission over a wireless link is error-prone due to the physical characteristics of radio frequency. So the link layer retransmission is also an essential factor for successful transmissions. Under downlink scenarios, an AP receives data packets from the server and delivers them to the clients, keeping data packets safely until it receives corresponding LL ACKs. Once a LL ACK arrives to the AP, it deletes corresponding data packets. Otherwise link layer retransmission is conducted. Figure 3.1 illustrates the whole process of TCP data transmission.

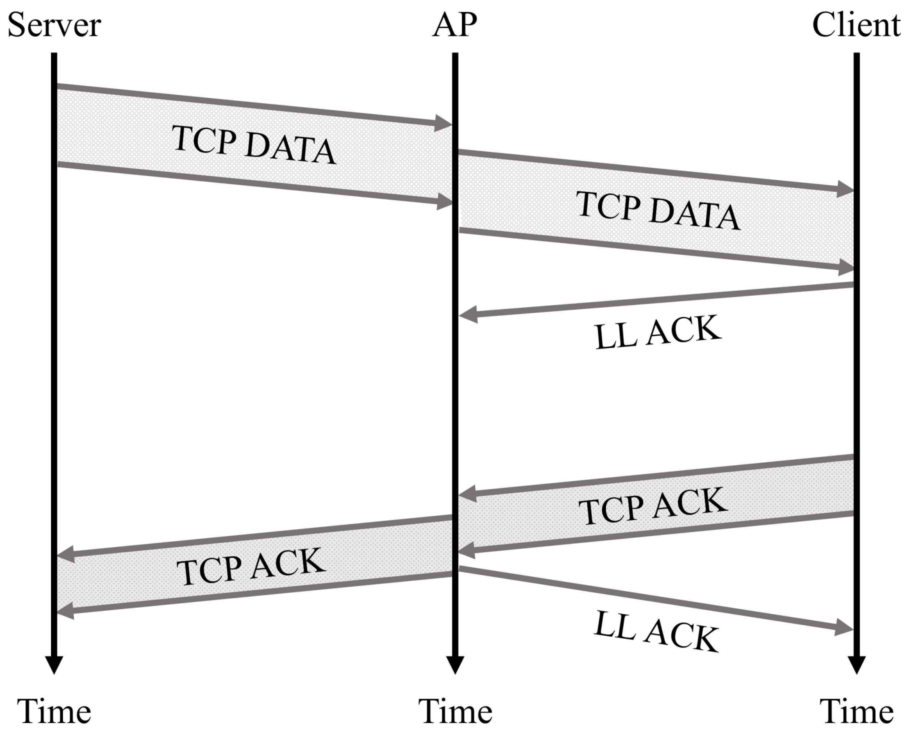

Figure 3.1: TCP Data Transmission

From the moment of birth of wireless networks, many researchers examined whether TCP is an appropriate protocol model for wireless networks [11]. We went further and detailed the question. Is TCP an appropriate protocol model for wireless devices whose computing power overwhelm that of an old mainframe server? The point which claims our attention is that TCP ACK and LL ACK hold similar functionalities. Although they differ in the point where retransmission is performed and other small details, they fundamentally serves same main purpose to assure reliable data transmission. If the TCP data received at the link layer is not dropped out in the path of getting to the transport layer of a wireless device, an AP can assume that the data is successfully delivered with LL ACK from the client. As shown in Figure 3.1, the process took after the reception of a LL ACK on the AP side can be considered unnecessary to confirm successful transmission of a packet.

TCP clients in wireless networks are mostly mobile devices. The reason why TCP data is not delivered to the transport layer can be inferred as follows; TCP Receiver Window size (RWND) is too small or the process terminates abnormally. However, the current mobile devices have higher computing capability and more memory compared to aged server computers. And the operating systems running on mobile devices maintain RWND large enough by auto-tuning. Therefore, the case where TCP data is dropped in the middle of link layer and transport layer is extremely rare.

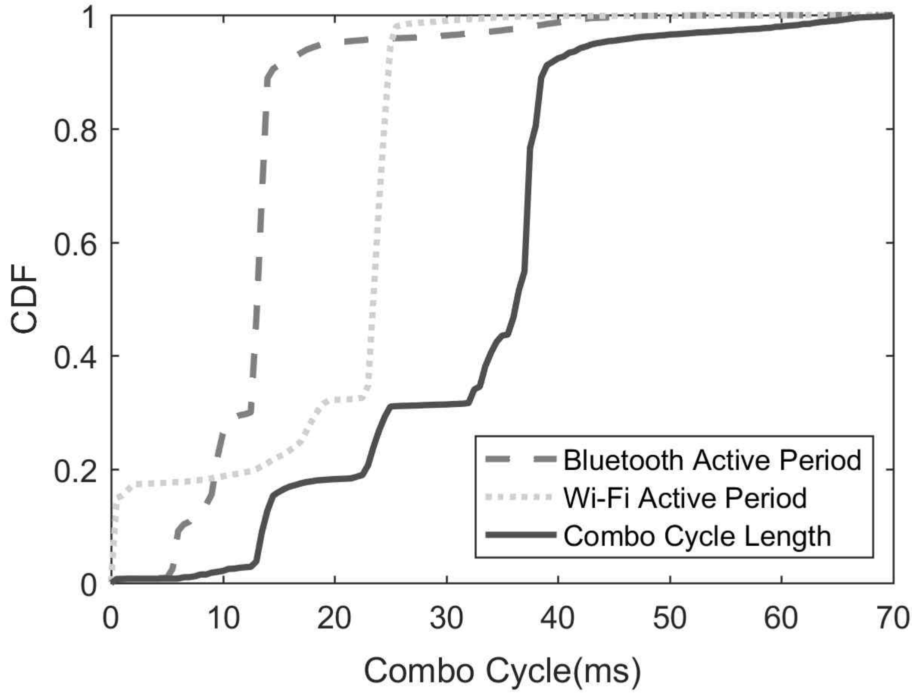

Figure 3.2: CDF measurement of Combo Cycle

3.2 Operation of Combo-modules

While the operation of Wi-Fi interfaces is transparent to the types of traffic, Bluetooth operates differently depending on traffic types. For example, Bluetooth interface classifies audio devices as A2DP (Advance Audio Distribution Profile) and tries to provide real time transfer of audio traffic using ACL (Asynchronous Connection-Less) links [4]. Because most audio traffic is less than 200 Kbps, audio traffic uses up to 20 % of airtime of 1Mbps Bluetooth interfaces in ideal case. In this paper, we focus on audio transmission. Combo-modules support generic Wi-Fi

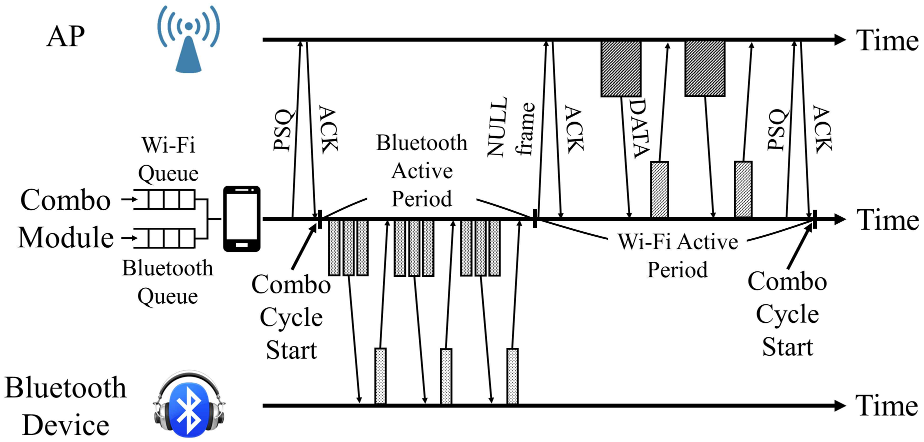

Figure 3.3: Bluetooth and Wi-Fi coexistence in TDM manner on a combo-module

traffic and real time Bluetooth audio traffic by sharing the medium in a TDM manner. Time is partitioned into repeated cycles (Combo Cycle) each of which consists of a Bluetooth active period and a Wi-Fi active period. The cycle length is fixed at about to 40 ms in Figure 3.2 and the boundary between Bluetooth and Wi-Fi periods are variable allowing sufficient transmission opportunities to time critical Bluetooth traffic.

Figure 3.3 shows the typical pattern of the TDM based sharing mechanism. At the start of each cycle, to provide uninterrupted airtime to the Bluetooth interface, a combo-module silences the AP as well as itself by issuing a frame that triggers Unscheduled Automatic Power Saving Delivery (U-APSD). We call the control frame as PSQ (Power Saving reQuest). Combo-modules transmit a PSQ frame several times until it is ACKed. After PSQ transmissions, combo-modules switch the Wi-Fi interface into the sleep mode and transmits Bluetooth

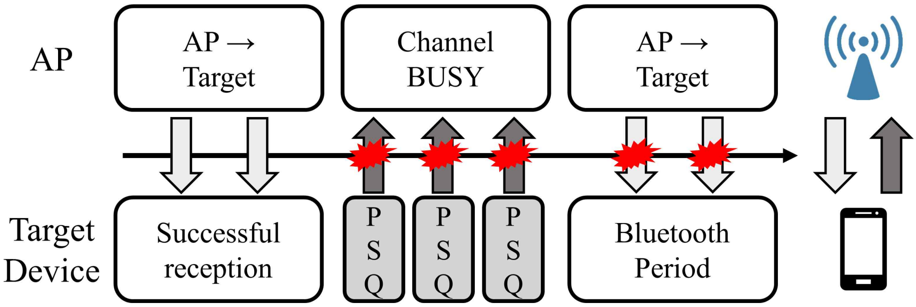

Figure 3.4: The failure of PSQ delivery fails to prevent the associated AP from sending packets to the combo client device. The AP continues to send packets to the client even when the client is not able to receive since it is on the Bluetooth active period.

frames. Wi-Fi frames generated during Bluetooth active periods are queued at the device driver. After finishing Bluetooth transmission, combo-modules awake the AP by sending either a queued data frame or a null frame. Again, Bluetooth frames generated during Wi-Fi active periods are queued and served during the next Bluetooth active period.

First, we checked the performance of combo-modules' TDM based sharing mechanism. The baseline performance of a combo-module is measured when Wi-Fi is operating alone without Bluetooth. We ran iperf in a free channel to have downlink traffic and found the throughput to be 23.32 Mbps. Then we streamed a music to a Bluetooth speaker, and the throughput while operating Wi-Fi and Bluetooth at the same time was measured to be 13.25 Mbps, which is 56.82 % of the baseline throughput. The average of combo cycle in the experiment was 31.68 ms, and Wi-Fi active period took 59.29%

out of whole portion recording 18.79 ms. As a result, we confirmed that there occurs 2.47% of throughput degradation as combo-modules operate in TDM manner.

Then, to determine the cause of performance degradation, we turned our focus onto the process of the combo-module switching from Bluetooth active period to Wi-Fi active period. If a combo-module enters to Bluetooth active period, without having succeeded on delivering PSQ to an AP due to busy channel environment, the AP cannot tell which state the associated mobile device is in. Then the AP blindly sends data packets to the client, without knowing that it is in Bluetooth active period, and faces transmission failures. Figure 3.4 illustrates the scenario of missing PSQ. This phenomenon confuses the link layer and forces the AP to adjust transmission rate of the following packets unnecessarily low. This unpredictable problem is known as Avalanche Effect.

Chapter Ⅳ

Proposed Scheme

4.1 Implicit ACK (I-ACK)

In Section I, we mentioned that LL ACK is as highly credible as TCP ACK. From this insight, we devised an AP side solution where an AP receives an LL ACK and transmits a TCP ACK back to the server. An AP keeps data packets destined to clients until it receives LL ACK as provisions for link layer retransmission. In other words, it implies that an AP is capable of generating a corresponding TCP ACK using the stored data packet, which is originally a charge of the clients, at the moment it receives a LL ACK. When the TCP ACK generated by an AP is delivered to the server, it can process and send more data to the destined clients as it gains time advantage in simplified procedure. We named the scheme sending a TCP ACK generated by an AP for fast response Implicit ACK (I-ACK).

Figure 4.1 illustrates the operation of an I-ACK supported AP. The scheme takes one of two different procedures depending on whether the client is aware of I-ACK support or not. (1) First, as depicted in the Figure 4.1, the clients who are ignorant of I-ACK support will operate as usual. That is, they will normally transmit TCP ACKs back to an AP. Then, the I-ACK supported

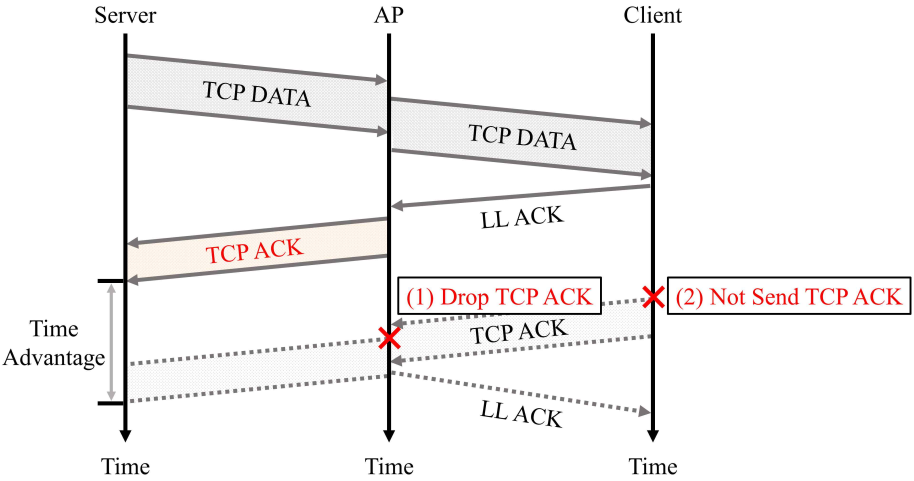

Figure 4.1: TCP data transmission by an I-ACK supported AP

AP, which has already sent I-ACKs to the server, simply drops the received TCP ACKs and prevents occurrence of unnecessary traffics. (2) And in case where the clients are aware of I-ACK support, they suppress transmission of TCP ACKs, and thereby avoid needless medium access. In this case, TCP ACKs, which used to follow LL ACK of TCP data, and LL ACKs, which used to follow those TCP ACK, no longer exist and noticeably clear the medium. Here the clients can selectively choose preferable operation like it does in 802.11e' NoACK policy, and APs can provide customized services to each client.

In addition, we also identified TCP ACKs having abnormal ACK number after transmission failure. To remedy this issue, we managed an AP to maintain dropped ACK numbers similarly with TCP SACK.

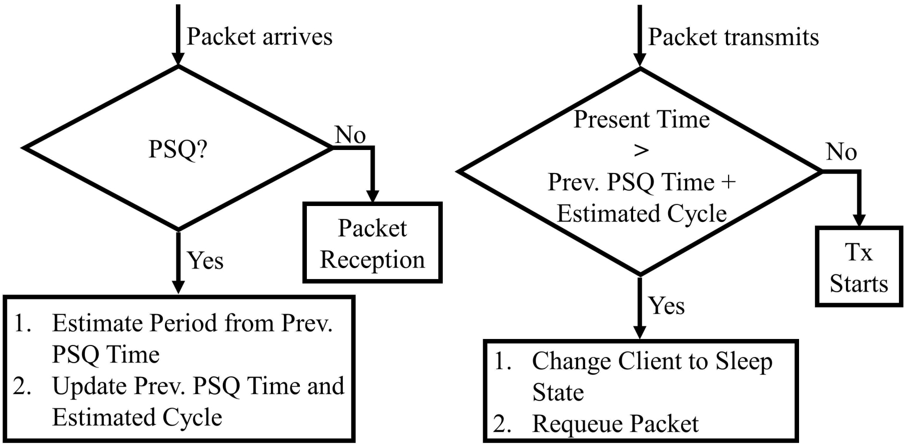

Figure 4.2: Smart AP Operation

4.2 Smart AP

We confirmed an overall degradation throughout a single AP and its connected mobile devices, due to the existence of a combo-module which performs combo operations. This is caused by a transmission failure of the PSQ and following Avalanche Effect, during the transition process into a Bluetooth active period. So we made that an AP could estimate combo cycle. Even in the cases where PSQ is not delivered to the AP, it may assume that the combo cycle is constant in some degree and thus restrain data transmissions towards mobile devices. We designate this operation as "Smart AP."

Figure 4.2 shows the process of a Smart AP. As soon as a PSQ packet is received, the AP alters the client's state, which internally performs combo operation. From that stage, the packets destined to the client are stored in a queue, until the client transmits a null data frame. Serving Bluetooth traffics in real-time manner, cycle length of combo-modules are maintained fairly consistent. Therefore an AP can estimate the cycle length based on several successful receptions of PSQ packets. Then, it suppresses sending data packets to a combo-module device, once elapsed time reaches the point where the target device is assumed to be in Bluetooth active period. Through this process, even in the situation of PSQ packet transmission failure, an AP avoids sending data packet to combo-modules in Bluetooth active period and prevents successive transmission failures.

Chapter V

Evaluation

5.1 Experiment Setup

We implemented I-ACK and Smart AP schemes by modifying ath9k driver in OpenWrt (Chaos calmer 15.05) running on off-the-shelf APs (TP-Link Archer C7). I-ACK and Smart AP are applied with roughly 0.7K lines of code. Nexus 5, equipped with Broadcom BCM4339 wireless chipset, was used as the target mobile device. Two servers exploited in experiments were located in a place where it shows RTT of ~6ms. In the meanwhile, target mobile device and competing device were managed to receive traffics from different servers. We ran experiments on 802.11g channel 11 (2.462 GHz) in an anechoic chamber. We used iperf to generate TCP data streams with a 1500 byte MTU and 54Mbps rate which is of the highest rate 802.11g supports. Performance of I-ACK and Smart AP was measured and analyzed both separately and combinatorially. During experiments, overall CPU usage of AP remained under 10%.

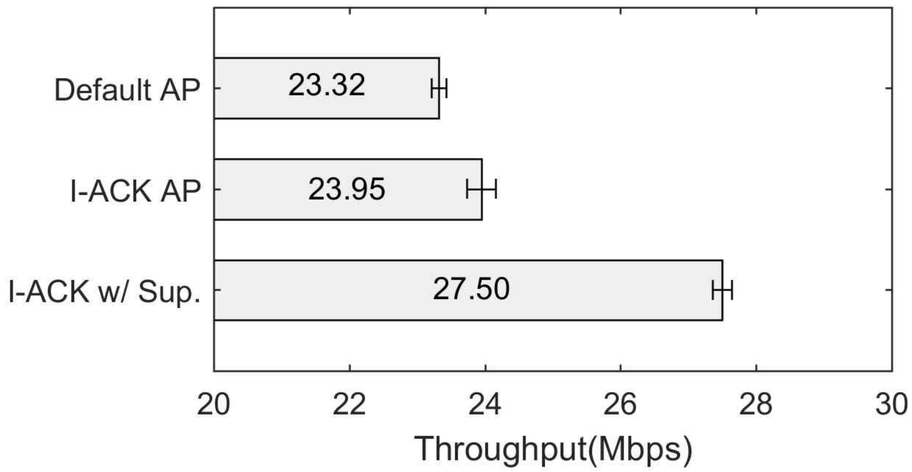

Figure 5.1: Throughput of default AP (Defaulkt AP), with I-ACK AP (I-ACK AP), and with I-ACK AP and a mobile device suppressed TCP ACK (I-ACK w/ Sup.)

5.2 Implicit ACK (I-ACK)

I-ACK is a scheme designed to reduce the redundancy in TCP data transmission procedure in a wireless environment. To verify the effectiveness of the scheme, we measured performance of a default AP without any revision and an I-ACK supported AP. And for the target device, we differentiated the cases where it operates with no modifications and where it suppresses TCP ACK.

The performances of each case are shown in Figure 5.1. Default AP and I-ACK AP is tested with a mobile client without any modifications. As seen from the figure, I-ACK AP achieved 2.7% higher performance compared to the default AP. This performance gain occurs because TCP ACKs arrives to the server quickly by saving the time spent on the client to process data packets. When I-ACK is complemented with the client-side TCP ACK suppress, it showed remarkable increase in performance about 17.92% compared to that of the default AP. The wireless medium, once suffered from crowded traffic and acted as a bottleneck, became relatively free with absence of unnecessary TCP ACKs by clients. For every TCP data packets sent to the clients, we save medium acquisition time and transmission time, and hence improve the throughput.

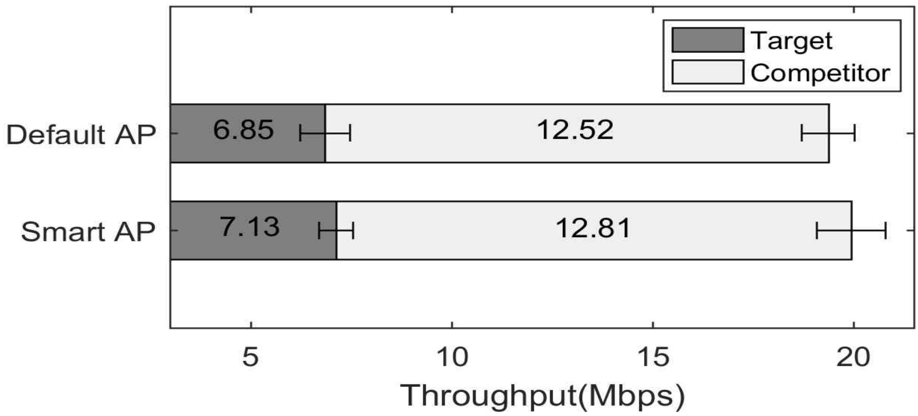

Figure 5.2: Throughput of target and competitor devices using default AP and Smart AP

5.3 Smart AP

Smart AP implements a mechanism that handles the case where PSQ frames from the clients to enter Bluetooth active period are not delivered to the AP. In such scenario, throughput not only of combo modules but of all devices in the same channels degrades significantly. To set up an environment crowded by multiple devices, we placed two interferer APs on channel 1 and 4, respectively, and generated interfering traffic with iperf.

Figure 5.2 shows the throughput of target and competitor devices when using default AP and Smart AP respectively. Using the Smart AP, the throughput of the target device, which performs combo operation, increased by 4.1% and the throughput of the overall network increased by 2.94%. This implies that the Smart AP can effectively solves the problem occurred when control frames are missed in the air.

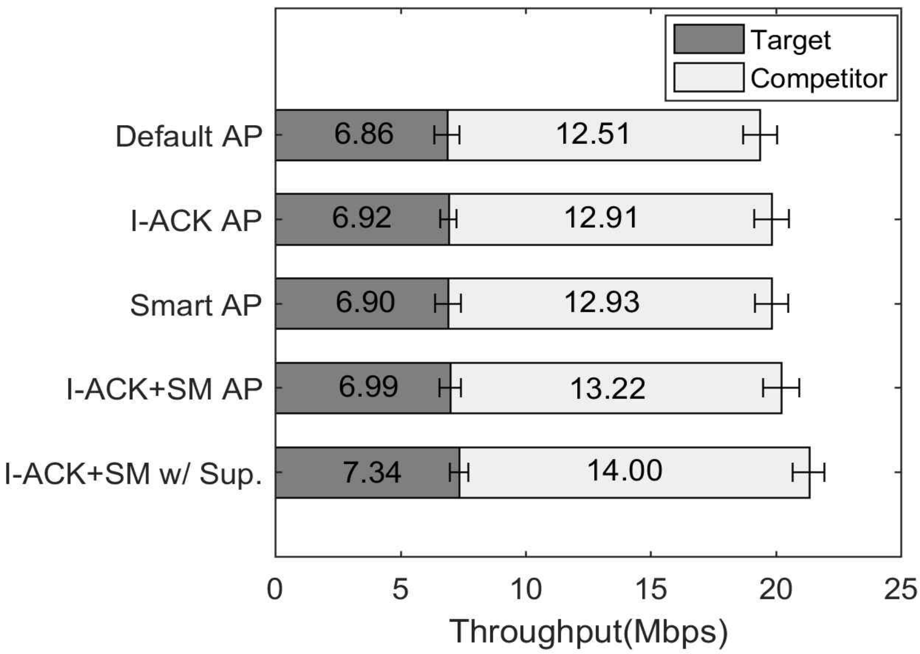

Figure 5.3: Comprehensive throughput measurements

5.4 Mixed Scheme

The performance gain by combining suggested schemes is shown in figure 9. The experiment was conducted without interferences to explicitly show the benefits of each scheme. Wi-Fi channel 11 was occupied by a target mobile device and its competitor. I-ACK and Smart AP both ensured higher performance to target and competitor devices. Finally, when all suggested schemes combined together with a target device that suppresses TCP ACKs, it achieved 7.0% and 5.01% of throughput gain at the target mobile device and at overall network respectively.

Chapter Ⅵ

Conclusion

In this paper, we have described the design and implementation of I-ACK and Smart AP. I-ACK scheme is a cross-layer design for TCP and 802.11 MAC, which alleviates performance degradation caused by redundant functionalities in TCP applied to wireless networks. And the Smart AP implements an inner scheduler that manages combo cycle inferred from PSQ frames. Both schemes can be applied with only a few revisions on current AP operation and guarantee great deal of improvement in Wi-Fi throughput. Efforts to improve performance in wireless networks have been continuously made by researchers [9, 11]. However, in many cases, the suggested solutions could not be applied only with modifications on APs but also on multiple components in networks. As main contributions, we proposed AP side solutions to improve Wi-Fi performance. In addition, with complement of TCP ACK suppress on mobile devices, the throughput gain could be further enhanced, presenting a new possibility to break through increasingly crowded wireless environment.

BIBLIOGRAPHY

[1] L. Cabral and R. Lins, "An Analysis of the QoS in the Transmission in 802.11g Networks in the Presence of Bluetooth Interference", In 2009 Fifth International Conference on Wireless and Mobile Communications, 2009, pp. 76-81.

[2] Jobs. S, Apple demo failure 2010 (iphone wi-fi fail). https://www.youtube.com/watch?v=zNlBLyf39Bk, 2010.

[3] C. Cordeiro, S. Abhyankar, R. Toshiwal, D. Agrawal, "BlueStar: Enabling Efficient Integration Between Bluetooth WPANs and IEEE 802.11 WLANs", In Mobile Networks and Applications 9, 4, 2004, pp. 409-422.

[4] SIG, B. Bluetooth specification, 2014.

[5] C. Chiasserini and R. Rao, "Coexistence Mechanisms for Interference Mitigation between IEEE 802.11 WLANsBluetooth.", In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2002, vol. 2, pp. 590-598.

[6] C. Chiasserini and R. Rao, "Coexistence Mechanisms for Interference Mitigation in the 2.4-GHz ISM Band", Wireless Communications, IEEE Transactions on 2, 5, 2003, 964-975.

[7] A. Xhafa, X. Lu, D. Shaver, "Coexistence of Collocated IEEE 802.11 and Bluetooth Technologies in 2.4 GHz ISM Band", In AccessNets, 2008, pp. 138-145.

[8] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang. "Fine-grained Channel Access in Wireless LAN", In SIGCOMM, 2010.

[9] L. Salameh, A. Zhushi, M. Handley, K. Jamieson, B. Karp, "HACK: Hierarchical ACKs for Efficient Wireless Medium Utilization", In Proc. USENIX ATC, 2014.
[10] IEEE Standards 802.11 specification, http://standards.ieee.org/about/get/802/802.11.html, 2012.
[11] H. Balakrishnan, S. Seshan, E. Amir, R.H. Latz, "Improving TCP/IP Performance over Wireless Networks", In Proc. Mobicom, 1995.
[12] A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", Proc. ICSCS 1995, pp. 136-143.
[13] M. Mathis et al., "TCP Selective Acknowledgment Options", RFC 2018, 1996.
[14] L. Brakmo and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," IEEE JSAC, vol. 13, no. 8, Oct. 1995, pp. 1465-80
[15] K. Leung and V. Li, "Transmission Control Protocol (TCP) in Wireless Networks: Issues, Approaches, and Challenges", IEEE Communications Surveys & Tutorials , 4th Quarter 2006, pp. 64-79.
[16] A. Palin, M. Honkanen, "VoIP call over WLAN with Bluetooth headset multiradio interoperability solutions", In Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on, 2005, vol. 3. pp. 1560-1564. In this pap

요 약

스마트 무선공유기를 이용한 와이파이 성능 향상

김기석
컴퓨터공학부
서울대학교 대학원

무선 네트워크는 유선 네트워크와 비교할 때 우수한 접근 용이성과 편의성으로 인해 활용 범위가 지속적으로 확장되고 있다. 하지만 물리적으로 제한된 주파수 환경은 점증하는 네트워크 혼잡으로 인한 성능 저하를 야기시키고 있다. 그리고 유선, 무선을 구분하지 않고 대다수의 네트워크에는 TCP 프로토콜이 사용되고 있는데, 사실 이 프로토콜은 태생적으로 유선 네트워크에 최적화되어 설계된 것이다. 또한 안테나를 공유하여 여러 네트워크 인터페이스를 사용하는 콤보 모듈의 등장은 기술간 간섭이라는 새로운 도전을 만들어 냈다. 본 논문에서는 먼저 TCP 전송 과정간 다중 계층에서 중복적으로 발생하는 확인 과정과 불필요한 전송 실패를 유발하는 콤보 모듈의 간섭 현상의 원인을 확인하였다. 그 후 무선공유기에서 이 문제점들을

개선하는 해결책인 I-ACK과 스마트 무선공유기를 제안한다. I-ACK은 TCP 전송 절차에서 중복 요소를 제거하고, 스마트 무선공유기는 콤보 모듈로 인해 야기되는 오작동에 대처할 수 있다. 끝으로 이 두가지 제안을 상용 무선공유기에서 구현하고 외부 간섭에서 자유로운 환경인 무반향실에서 와이파이 성능 개선이 발생함을 입증하였다.

주요어: 와이파이, 무선공유기, TCP, 교차계층 접근방법, 콤보 모듈, 절전모드, 이기종 네트워크 공존

학번: 2015-21228