DOI: 10.3745/KIPSTC.2011.18C.3.187

IEEE 802.11 무선랜 환경에서의 스니핑 기반 전송률 측정 기법(VirtFrame)에 관한 연구

서성훈[†] · 백재종[††] · 김동건[†††] · 송주석[††††]

요약

국문요약-IEEE 802.11 무선랜을 기반으로 한 무선 통신 기술은 최근 비약적으로 발전하고 있다. 무선랜이 조밀하게 구축된 환경에서 이동국(Mobile Station)은 다중으로 접속점(Access Point)의 중복된 전송 범위에 위치할 때 일반적으로 그 중 최선의 접속점을 선택하여 연결을 시도한다. 대부분 최선의 접속점 선택 기준은 이웃한 접속점의 수신 신호 세기가 가장 높은 것을 선택하여 결정한다. 하지만 이러한 신호 세기만을 기반으로한 접속점 선택 방법은 각 접속점들에 걸리는 트래픽 부하가 고려되지 않아 트래픽 분산 측면에서 네트워크 성능에 악영향을 미칠 수 있는 문제점이 있다. 따라서, 본 논문에서는 802.11 기술의 스니핑 기술을 이용하여 이웃한 접속점들이 사용하는 전송률을 측정하는 기법(VirtFrame)을 제안한다. 제안된 기법은 스니핑된 프레임을 가상으로 조합하고, 각각 채널 접근 시간을 계산하여 이웃한 접속점 각각의 가용 전송률을 측정한다. 제안된 기법의 성능에 대한 분석 및 평가로써 측정된 전송률과 계산된 전송률 사이의 연관 관계를 분석한 모의실험 결과를 제공한다.

키워드 : IEEE 802.11, 무선랜, 전송률 측정

VirtFrame: A Sniffing-based Throughput Estimation Scheme in IEEE 802.11 Wireless LANs

SungHoon Seo[†] · JaeJong Baek[††] · Donggun Kim[†††] · JooSeok Song[††††]

ABSTRACT

IEEE 802.11 wireless LAN has become the center of attention for one of the most dominant wireless networking technologies nowadays. In densely deployed wireless LANs, mobile stations are exposed to a number of AP, thus it is needed to select the best AP to associate with. The most common approach is to select the AP with the highest received signal strength. However it does not consider traffic load imposed to each AP so that it may cause the poor network performance. Therefore, in this paper, we propose a throughput estimation scheme for neighboring APs by sniffing the traffic within 802.11 networks. We devise a tool, named "VirtFrame", which is to estimate the station's capable throughput from neighbor APs based on the channel access time by virtually combining the sniffed frames. Simulation results show that our proposed scheme well matches that there exists correlation between the channel access time and the actual throughput of the APs.

Keywords : IEEE 802.11, WLAN, Throughput estimation

1. INTRODUCTION

As various kinds of mobile devices are getting popular, wireless networking technologies play more important role nowadays. Most of all, IEEE 802.11 is one of the most dominant technologies for WLAN (Wireless Local Area Network). This technology is license free, so 802.11 based WLANs are deployed in many hotspot areas including homes, schools and offices. In addition, IEEE 802.11 technology makes people use public Internet access services with cheaper price compared to cellular technology.

In a densely deployed WLAN hotspot area, 802.11 stations (STAs) are exposed with a number of Access

---

※ This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government[NRF-2009-352-D00257].

† 정 회 원 : Columbia University 컴퓨터과학과 박사후연구원
†† 준 회 원 : 연세대학교 컴퓨터과학과 박사과정
††† 준 회 원 : 연세대학교 컴퓨터과학과 석사
†††† 종신회원 : 연세대학교 컴퓨터과학과 정교수

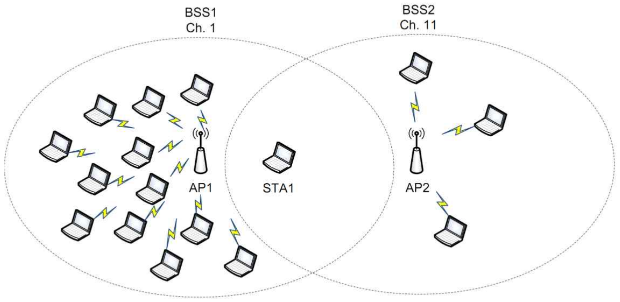
논문접수 : 2011년 1월 28일
수 정 일 : 1차 2011년 3월 22일
심사완료 : 2011년 3월 28일

Points (APs). Since an STA can associate with an AP at the same time, selecting the best AP is a very important research issue in the WLAN technology. Existing approaches have used the highest received signal strength as a common criterion to select the best AP. (Figure 1) depicts a scenario where a station (STA1) is within the hotspot coverage of two APs (AP1 and AP2). AP1 and AP2 are configured with Basic Service Set 1 (BSS1) in channel number 1 and BSS2 in channel number 11, respectively, in order to avoid inter-channel interference. STA1 is located closer to AP1, other than AP2, so the received signal strength which measured from AP1 is relatively higher level than that from AP2. On the other hand, many STAs are already connected to AP1 and highly contend the channel medium, while few STA is associated with AP2. The received signal strength based AP selection may lead to AP1, however, STA1 may achieve higher throughput than if STA1 associates with AP2. Therefore, in this paper, we propose a throughput estimation scheme by using traffic sniffing in 802.11 WLANs. We devise a tool named “VirtFrame” which is to estimate throughput by analyzing the channel access time of sniffed frames. More specifically, the VirtFrame regards some types of sniffed frames as virtually one frame to simplify the analysis of the throughput estimation. We also conduct simulation studies and results show that our proposed scheme is able to find correlation between the channel access time and the actual throughput that STAs can achieve from APs.

2. RELATED WORK

2.1 AP Selection Mechanisms

IEEE 802.11 standard does not specify an AP selection mechanism. Currently wide used mechanism to select an AP is based on received signal strength from an AP. But this mechanism does not consider traffic load imposed to each AP, so it frequently leads poor performance [1]. In order to enhance the AP selection, several approaches have been appeared in literature. Vasudevan et al. proposed a potential bandwidth estimation scheme to estimate downstream and upstream bandwidth by using beacon delays and transmitted data frames to an AP with their delay, respectively [2]. Chen et al. proposed estimating traffic load of an AP by sending a probe request frame and measure the delay until probe response frame is received [3]. Some existing AP selection mechanisms require a modification protocol stack in an AP to provide some information for its STAs [4] [5]. But the modification is not encouraged because it is not compatible with previously deployed APs. We thus focused on designing our proposed scheme without any modification in APs. Therefore, STAs sniff traffic on each candidate AP’s wireless channel and perform the AP selection based on the analyzed sniffed frames.



(Fig. 1) A scenario of AP selection

2.2 Analysis of IEEE 802.11 MAC Protocol

Analysis of IEEE 802.11 MAC protocol has been extensively studied. Giuseppe Bianchi proposed two-dimensional Markov chain model to evaluate the performance of IEEE 802.11 DCF (Distributed Coordination Function) [6]. His model is very accurate to estimate throughput and most widely used. In addition, this model was extended to analyze traffic model in IEEE 802.11e EDCA (Enhanced Distributed Channel Access) [7]. However, solutions for this model are very complex since it consists of nonlinear equations.

As another approach, Tay and Chua introduced a methodology for the throughput estimation called mean value analysis [8]. Their approach is much simple than Bianchi’s Markov model, while still maintaining accuracy of throughput estimation. It was also extended for analyzing IEEE 802.11e EDCA [9]. In this paper, the proposed scheme is based on the approach that Tay and Chua [8] proposed.

3. SNIFFING-BASED THROUGHPUT ESTIMATION SCHEME

3.1 General Description

In IEEE 802.11 DCF, frames cannot be transmitted immediately and there is some delay between the time sending request to MAC layer from upper layer and the time when frames are actually transmitted. A channel access time is defined as the delay between the time that an STA starts to request a new frame sending through its MAC layer and the time the frame successfully transmitted.

The channel access time includes the time waiting for the STA to be idle, time for backoff process, and time related to the error recovery when error occurs. Let T_{ca} denote the average channel access time which is an expected value of the channel access time. For T_{ca}, we only consider cases that transmission eventually succeeds, which mean that the transmission succeeds within retry limit.

In our scheme, an STA calculates the T_{ca} for every candidate AP by associating to the AP and trying to transmit a single DATA frame. After finishing the calculation, the STA selects one of the candidate APs as a target AP which has the minimum T_{ca}. This is rationale because small access time means a small traffic load, so we can expect better quality of service with smaller access time. The channel access time reflects the key features of IEEE 802.11 DCF including carrier sensing, backoff process, collision, and retransmission with increased contention window size.

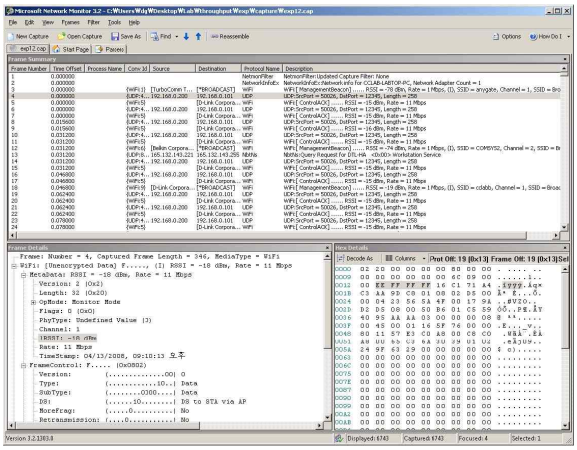
In order to calculate T_{ca}, in our scheme, an STA should firstly sniff frames from the channel of candidate APs during a predefined time. By using the sniffed frames, the STA gathers information related to the channel i.e., the average channel access time. We will show details how to calculate T_{ca} in Sec. III-B

The frame (or packet) sniffing in IEEE 802.11 can be implemented in software using an ordinary NIC (Network Interface Card). Some operating systems including Windows, Linux, and BSDs provide API (Application Programming Interface) to enable the 802.11 NIC on monitor mode [10], sometimes called promiscuous mode. There are already several frame/packet sniffing tools which support the monitor mode in IEEE 802.11, including Microsoft's Network Monitor, Wire-shark, OmniPeek, etc. (Figure 2) shows the output of sniffed 802.11 MAC frames using the monitor mode of Microsoft's Network Monitor [11].

When the sniffing on a specific channel is performed in the monitor mode, an STA can obtain every frame transmitted within its communication range regardless of neither frame type nor BSS. By using the sniffing tools, the STA can gather all MAC layer information for each frame in addition to several extras such as RSSI and data rate. On the other hand, the snipping tools cannot gather PHY (physical) layer information but it is negligible because our proposed scheme does not require any information from the PHY layer.

3.2 Calculation of Channel Access Time using VirtFrame

To estimate the throughput performance, it is necessary to calculate the channel access time by STAs. Let T denote the period while an STA performs the sniffing with its monitor mode. N is the number of MAC frames obtained during the sniffing period, T, and each frame is numbered in the range from 0 to N-1. Li is the length of the i-th MAC frame in bit(s).

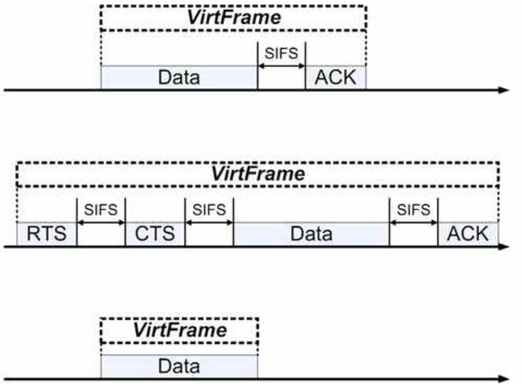


```
Frame Number  Time Offset  Process Name  Conv Id  Source            Destination        Protocol Name  Description
1             0.000000                                              NetmonFilter       NetmonFilter:Updated Capture Filter: None
2             0.000000                                              NetworkInfoEx      NetworkInfoEx:Network info for CCLAB-LABTOP-PC, Network Adapter Count = 1
3             0.000000     {WiFi:1}  [TurboComm T...   [*BROADCAST]       WiFi           WiFi:[ ManagementBeacon] ...... RSSI = -78 dBm, Rate = 1 Mbps, (I), SSID = anygate, Channel = 1, SSID = Bro
4             0.000000     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
5             0.000000     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
6             0.000000     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
7             0.000000     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
8             0.015600     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
9             0.015600     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -16 dBm, Rate = 11 Mbps
10            0.031200     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
11            0.031200     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
12            0.031200     {WiFi:6}  [Belkin Corpora... [*BROADCAST]      WiFi           WiFi:[ ManagementBeacon] ...... RSSI = -74 dBm, Rate = 1 Mbps, (I), SSID = COMSYS2, Channel = 2, SSID = Br
13            0.031200     {UDP:8... 165.132.143.221   165.132.143.255    NbtNs          NbtNs:Query Request for DTL-HA   <0x00> Workstation Service
14            0.031200     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
15            0.031200     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
16            0.046800     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
17            0.046800     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
18            0.046800     {WiFi:9}  [D-Link Corpora... [*BROADCAST]      WiFi           WiFi:[ ManagementBeacon] ...... RSSI = -19 dBm, Rate = 1 Mbps, (I), SSID = cclabb, Channel = 1, SSID = Broac
19            0.062400     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
20            0.062400     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
21            0.062400     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
22            0.062400     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
23            0.078000     {UDP:4... 192.168.0.200     192.168.0.101      UDP            UDP:SrcPort = 50026, DstPort = 12345, Length = 258
24            0.078000     {WiFi:5}                    [D-Link Corpora... WiFi           WiFi:[ ControlACK] ...... RSSI = -15 dBm, Rate = 11 Mbps
```

```
Frame: Number = 4, Captured Frame Length = 346, MediaType = WiFi
WiFi: [Unencrypted Data] F....., (I) RSSI = -18 dBm, Rate = 11 Mbps
  MetaData: RSSI = -18 dBm, Rate = 11 Mbps
    Version: 2 (0x2)
    Length: 32 (0x20)
    OpMode: Monitor Mode
    Flags: 0 (0x0)
    PhyType: Undefined Value (3)
    Channel: 1
    lRSSI: -18 dBm
    Rate: 11 Mbps
    TimeStamp: 04/13/2008, 09:10:13 오후
  FrameControl: F..... (0x0802)
    Version:         (..............00) 0
    Type:            (............10..) Data
    SubType:         (........0000....) Data
    DS:              (......10........) DS to STA via AP
    MoreFrag:        (.....0..........) No
    Retransmission:  (....0...........) No
```

```
0000  02 20 00 00 00 00 80 00 00  . ..... ..
0009  00 00 00 00 00 00 6C 09 00  ......l..
0012  00 EE FF FF FF 16 C1 71 A4  .îÿÿÿ.Áq¤
001B  C3 AA 9D C8 01 08 02 D5 00  Ãª È...Õ.
0024  00 04 23 56 5A 4F 00 17 9A  ..#VZO..
002D  D2 D5 08 00 50 B6 01 C5 59  ÒÕ..P¶.ÅY
0036  40 95 AA AA 03 00 00 00 08  @ ªª.....
003F  00 45 00 01 16 5F 76 00 00  .E..._v..
0048  80 11 57 E3 C0 A8 00 C8 C0   .WãÀ¨.ÈÀ
0051  A8 00 65 C3 6A 30 39 01 02   .eÃj09..
005A  24 9F 63 29 00 00 00 00 00  $ c).....
0063  00 00 00 00 00 00 00 00 00  .........
006C  00 00 00 00 00 00 00 00 00  .........
0075  00 00 00 00 00 00 00 00 00  .........
007E  00 00 00 00 00 00 00 00 00  .........
0087  00 00 00 00 00 00 00 00 00  .........
0090  00 00 00 00 00 00 00 00 00  .........
0099  00 00 00 00 00 00 00 00 00  .........
00A2  00 00 00 00 00 00 00 00 00  .........
00AB  00 00 00 00 00 00 00 00 00  .........
```

(Fig. 2) An example of sniffed frames/packets by Microsoft's Network Monitor



(Fig. 3) Examples of virtual frame, VirtFrame

When the i-th frame is a DATA frame and the i+1th frame is an ACK frame corresponding to the former (i-th) frame, no other frame is transmitted between transmitting these two frames because they are happened with a gap of SIFS (Short Inter Frame Space). To simplify the analysis, we regard the pair of the i-th and i+1th frames as one virtual frame (VirtFrame). The VirtFrame can be categorized as follows: (1) any two adjacent frames in a VirtFrame separated less than DIFS (Distributed Inter Frame Space) time and (2) any frame within or out of a VirtFrame separated with greater than or equal to DIFS time. As an example, we can regard contiguous frame(s) that are transmitted without contention as a VirtFrame. For the RTS/CTS (Ready To Send/Clear To Send) enabled 802.11 WLANs, the VirtFrame may consists of a set of frames in the sequence of RTS + CTS + DATA +

ACK. For the case of broadcast or multicast DATA frame, no ACK frame follows thus the VirtFrame consists of single DATA frame. (Figure 3) depicts the above mentioned examples of the VirtFrame.

When a sending request is arrived at an STA's MAC layer from upper layer and the wireless medium is idle for DIFS time (T_{DIFS}), the STA immediately transmits the frame where T_{ca} = T_{DIFS}. On the other hand, if the wireless medium is busy during the DIFS, the STA enter into a backoff process by starting the backoff timer. Then the average channel access time, T_{ca}, is given by

$$T_{ca} = (1 - p_{\text{backoff}})T_{\text{DIFS}} + p_{\text{backoff}}(T_{\text{w}} + T_{\text{backoff}}(1)) \quad (1)$$

where $p_{backoff}$ is probability to perform backoff, T_W is average waiting time for the wireless medium to be idle for T_{DIFS}, $T_{backoff}$ (i) is average time between start of backoff timer of i-th transmit trial and start of successful transmission.

Let T_i denote the time required for i-th frame, then $p_{backoff}$ is given by

$$p_{\text{backoff}} = \frac{\sum_{i=0}^{N-1}(T_i + T_{\text{DIFS}})}{T} \quad (2)$$

and T_W is expressed by

$$T_{\text{w}} = \frac{T_{\text{v}} + T_{\text{DIFS}}}{2} + T_{\text{DIFS}} \quad (3)$$

where T_V is average time length of the VirtFrame.

Let CW_i be a contention window size for a frame of i-th trial and it is given by

$$CW_i = \begin{cases} aCWmin & \text{if } i = 1 \\ \min\left(2(CW_{i-1} + 1) - 1,\ aCWmax\right) & \text{if } i > 1 \end{cases} \quad (4)$$

where $aCWmin$ and $aCWmax$ is the minimum and the maximum contention windows size defined in the 802.11 standard. If the STA is currently in i-th trial, possible value of random integer for backoff timer is 0 to CW_i and each value is chosen in probability $1/(CW_i + 1)$. For each time slot in backoff, if some other STAs start to transmit, backoff timer is not decremented and waits for the medium being idle for T_{DIFS} and then the backoff is resumed. When the backoff timer is expired, the STA starts to transmit the frame but if any other STAs are in the transmission during that time, collision occurs and the STA should retransmit the frame. Let T_s and T_a are a slot time and ACK timeout, respectively. Then the backoff time for the i-th trial, $T_{backoff}$ (i), can be expressed as in Eq. 5 where T_V is the average time length of the first frame of each VirtFrame, and p is the probability that other STAs transmit in a specific time slot. It is worth noting that that p is assumed constant over time, and the probability of a collision occurrence is also p for each trial. In Eq. 5, transmission is assumed to be successful until retry limit (R) increases to i since we only consider the case that the transmission succeeds within the retry limit. Finally, we simplify $T_{backoff}$ (i) as in Eq. 6.

$$T_{\text{backoff}}(i) = \begin{cases} \sum_{j=0}^{CW_i} \frac{1}{CW_i + 1} \sum_{k=0}^{j-1} \left(T_{\text{s}} + \sum_{l=1}^{\infty} p^l (T_{\text{v}} + T_{\text{DIFS}})\right) + p\left(\overline{T_v} + T_{\text{a}} + T_{\text{backoff}}(i+1)\right) & \text{if } i < R \\ \sum_{j=0}^{CW_i} \frac{1}{CW_i + 1} \sum_{k=0}^{j-1} \left(T_{\text{s}} + \sum_{l=1}^{\infty} p^l (T_{\text{v}} + T_{\text{DIFS}})\right) & \text{if } i = R \end{cases} \quad (5)$$

$$= \begin{cases} \frac{CW_i}{2}\left(T_{\text{s}} + \frac{p(T_{\text{v}} + T_{\text{DIFS}})}{1-p}\right) + p\left(\overline{T_v} + T_{\text{a}} + T_{\text{backoff}}(i+1)\right) & \text{if } i < R \\ \frac{CW_i}{2}\left(T_{\text{s}} + \frac{p(T_{\text{v}} + T_{\text{DIFS}})}{1-p}\right) & \text{if } i = R \end{cases} \quad (6)$$

Let n is the number of STAs which perform backoff process after the VirtFrame ends. pn denotes the probability that some station starts transmission in a backoff slot. By assuming that all these STAs are in the first trial, we can approximate p_n as follows:

$$p_n \approx 1 - \left(1 - \frac{1}{CW_1 + 1}\right)^n \quad (7)$$

The recurrence relation of p_n is also given by

$$p_n \approx \begin{cases} 0 & \text{if } n = 0 \\ 1 - (1 - p_{n-1})\left(1 - \frac{1}{CW_1 + 1}\right) & \text{if } n \geq 1 \end{cases} \quad (8)$$

In order to calculate p, we should first determine whether the medium is currently saturated or not. The medium saturation means that there exists at least one STA performing backoff process after any VirtFrame ends. We can determine the medium is saturated if and only if inequality holds the condition which is given by

$$\frac{N(T_{\text{v}} + T_{\text{DIFS}} + \frac{CW_1}{2} \cdot T_{\text{s}})}{T} > \tau \quad (9)$$

where we set the threshold τ =0.9.

When the medium is not saturated, the probability of contention occurrence among STAs is lower than when the medium is saturated. In other word, we can expect that the probability that a backoff process is interrupted by other station is also low. By excluding this case, we can approximate the average number of STAs which performs backoff process, n as follows:

$$\overline{n} \approx \sum_{k=0}^{N_{\mathrm{STA}}-1} \left((T_{\mathrm{v}} + T_{\mathrm{DIFS}})\frac{s_k}{T}\right) \tag{10}$$

where N_{STA} is the number of stations appeared in the address field of the sniffed frames and sk is the number of initiating VirtFrame by k-th STA. Since the summation of s_k is the total number of VirtFrame, n can be simplified by

$$\overline{n} \approx (T_{\mathrm{v}} + T_{\mathrm{DIFS}})\frac{N}{T} \tag{11}$$

Therefore, in case that the medium is not saturated, p is given by

$$p \approx 1 - \left(1 - \frac{1}{CW_1 + 1}\right)^{\overline{n}} \quad \text{if not saturated} \tag{12}$$

On the other hand, if the medium is saturated, contention occurs only when any VirtFrame ends, but it is difficult to find how many STAs are in backoff process using timing information. We thus address the information of retransmission flag that is in IEEE 802.11 MAC header. Let p_r denote the ratio of the number of frames setting the retransmission flag over the total number of sniffed frames. By assuming that all frames are successfully transmitted within the second trial, p_r equals the probability of collision occurred when one frame is transmitted. Therefore, p in the case that the medium is saturated is given by

$$p \approx 1 - (1 - p_r)\left(1 - \frac{1}{CW_1 + 1}\right) \quad \text{if saturated} \tag{13}$$

4. PERFORMANCE EVALUATION

We evaluate our proposed scheme with simulation study. We developed a discrete event simulator for IEEE 802.11 MAC protocol written in Python programming language. This simulator was implemented with the most of functionalities in 802.11 DCF mechanism including carrier sense, random backoff, MAC layer acknowledgement, ACK timeout, retransmission, and retransmission limit. During simulation, it is assumed that all STAs are located within small area and channel condition is good, so there are no hidden terminal problems or capturing effects, and all transmission would be successfully received by all other stations without error unless collision occurs. In addition, propagation and turnaround time via wireless medium are assumed to be zero. We consider two types of applications for the simulation, *i.e.*, FTP and CBR (Constant Bit Rate) applications. TCP with 1460 octet payload size is used for the FTP applications, while UDP with constant length of payload in every constant interval is used for the CBR applications. In one STA, each application has same chance to send a packet when there exists a packet to send. According to requirements in [12], TCP receiver should send acknowledgement for every second data packets received. In FTP application, there is infinite data to send and every STA tries to maximize its throughput. <Table 1> represents parameters used for the simulation. These parameters are basis of IEEE 802.11a PHY layer in 54 Mbps data rate.

<Table 1> SIMULATION PARAMETERS

Parameter	Value
aSlotTime	9 µs
aSIFSTime	16 µs
aPHY-RX-START-Delay	25 µs
aPreambleLength	16 µs
aPLCPHeaderLength	4 µs
aCWMin	15
aCWMax	1023
NDBPS (Number of data bit per symbol)	216 bits/symbol
TSYM (Symbol time)	4 µs
dot11ShortRetryLimit	7
MAC header length of Data frame	28 octets
Length of ACK frame	14 octets
LLC header length	8 octets
IP header length	20 octets
TCP header length	20 octets
UDP header length	8 octets
TCP payload length	1460 octets

4.1 Simulation Scenario

Five STAs (labeled from STA1 to STA5) are connected to an AP. Each STA is running some application or is idle. Each application is started in random time in first 1 second of simulation time. There is another STA which performs sniffing for 2 seconds to calculate T_{ca}. And it is then associated with the AP and measures FTP upload throughput for 10 seconds and measure FTP download throughput for 10 seconds. All non-AP stations are communicating with only AP stations. If an application is running upload, data is transferred from the station to an AP, and in download case, data is transferred from AP to the STA.

<Table 2> SIMULATION TEST CASES

Case no.	description
1	Idle
2 - 11	FTP
12 - 21	CBR 250 octet payload, 10 ms interval
22 - 31	CBR 500 octet payload, 10 ms interval
32 - 41	CBR 1000 octet payload, 10 ms interval
42 - 51	CBR 250 octet payload, 1 ms interval
52 - 61	CBR 500 octet payload, 1 ms interval
62 - 71	CBR 1000 octet payload, 1 ms interval

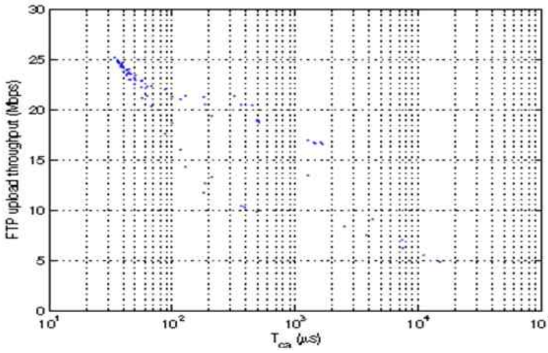
The applications performed in these 5 STAs are determined by test case which is categorized in <Table 2>. The first test case is for the idle where all 5 STAs are idle. Seventy different test cases (from the second to 71st) exist where each test case involved in one type of applications. For one application, we performed 10 test cases by varying the number of STAs (*i.e.*, from 1 to 5) and the direction of traffic (*i.e.*, upload and download).

4.2 Simulation Results

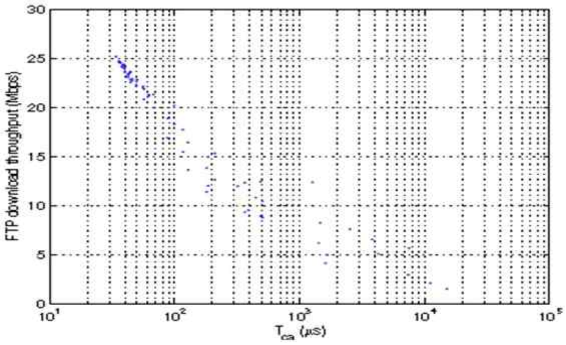
To show the effectiveness that our proposed traffic sniffing scheme can be used for better AP selection, we calculated average channel access time (T_{ca}) for each simulation test case, and compared it with FTP throughput measured by simulation studies.

(Figs. 4 and 5) show the correlation between the channel access time and the throughput of FTP application in upload and download, respectively. We plot simulated FTP upload and download throughput as a function of calculated average channel access time, T_{ca}, in log-scale. In these figures, each dot corresponds to the results from each test case denoted in <Table 2>.

As shown in these figures, there is correlation between T_{ca} and the FTP throughput. When T_{ca} is small we can expect better FTP throughput, and when T_{ca} is large we can expect poor FTP throughput.

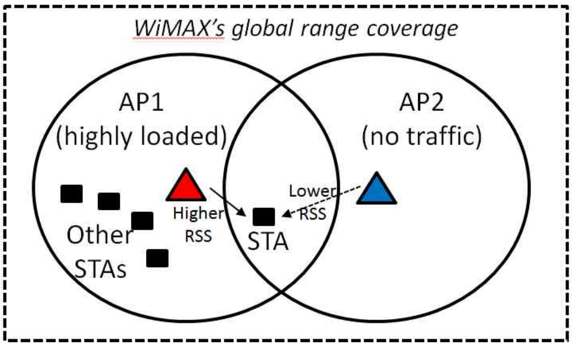


(Fig. 4) Correlation between Tca and FTP upload throughput



(Fig. 5) Correlation between Tca and FTP download throughput

The proposed scheme can be applied into various applications. For example, the estimated throughput with VirtFrame can be used as a metric for an AP selection when there is an overlapped coverage by a number of APs so that an STA should determine the best capable AP in regard to obtaining the highest throughput. We thus performed another simulation study to verify the proposed throughput estimation tool is suitable for increasing the throughput as well as fairness to the entire WLAN infrastructure. As a simulation environment, we assume all the area is covered by cellular technology such as longer range of WiMAX, and two APs' coverage are overlapped one another within the cellular coverage. (Fig. 6) shows the network environment we are using in this simulation study.



(Fig. 6) Simulation environment

As shown in (Fig. 6), AP1's resource is almost saturated thus an STA obtains less throughput if it is associated to the AP1, while an STA can achieve higher throughput if it associates to the AP2 where no ongoing radio resource is allocated at the AP2. Because an STA is located within the overlapped area between AP1 and AP2's coverage, selecting AP2 as a new point of attachment may give more benefits in terms of throughput even the RSSI from AP2 is somehow lower than that from AP1. Conventional AP selection scheme determines the AP1 as a new point of attachment, but by using our proposed throughput estimation tool can guide the AP2 as a new AP without actually connected to the AP2. <Table 3> shows the simulation result of STA's achieved throughput when AP1 is saturated and AP2 has no loaded traffic. It means that AP2 can give the STA more throughput than the AP1. To see the achieved throughput at STA, we generate constant rate of UDP traffic from an STA as soon as the STA associated to any AP. Both APs are configured with 1 Mbps fixed channel rate by turning off an automatic fallback feature. As shown in the result, by using the proposed scheme, an STA can achieve in average 407 Kbps more than when using the conventional RSSI based AP selection scheme.

<Table 3> Comparison of throughput achievement between conventional RSSI based AP selection scheme and the proposed throughput estimation based scheme

Item	Conventional scheme	Proposed scheme
STA's achieved throughput	21 Kbps	428 Kbps

As an additional aspect, when there is a number of STAs want to change their point of attachment, the proposed scheme make traffic load be fairly distributed which is comparable to using the conventional AP selection scheme, i.e., RSSI based AP selection, that does not consider the traffic load distribution.

5. CONCLUSION

In this paper, we proposed a new throughput estimation scheme for IEEE 802.11 wireless networks, which calculates average channel access time (T_{ca}) using traffic data collected with sniffing, and selects AP with smallest channel access time. We used mean value analysis to calculate Tca from collected traffic data. Our simulation result shows that there exists correlation between Tca and FTP throughput, so we can expect better FTP throughput with smaller channel access time.

REFERENCE

[1] G. Judd and P. Steenkiste, "Fixing 802.11 access point selection," ACM SIGCOMM Computer Communications Review, Vol.32, No.3, pp.31, Jul., 2002.

[2] S. Vasudevan, K. Papagiannaki, C. Doit, J. Kurose, and D. Towsley, "Facilitating access point selection in IEEE 802.11 wireless networks," IMC, Oct., 2005.

[3] J.-C. Chen, T.-C. Chen, T. Zhang, and E. van den Berg, "Effective AP selection and load balancing in IEEE 802.11 wireless LANs," in GLOBECOM, Dec., 2006.

[4] Y. Fukuda and Y. Oie, "Decentralized access point selection architecture for wireless LANs," in VTC, Sep., 2004.

[5] Y. Morioka, T. Higashino, K. Tsukamoto, and S. Komaki, "AP selection algorithm for real-time communications through mixed WLAN environments," IEICE Transactions on Communications 2008, Vol. E91-B, No.10, pp.3077-3084, Oct., 2008.

[6] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," Selected Areas in Communications, IEEE Journal on, Vol.18, No.3, pp.535-547, Mar., 2000.

[7] Z. Kong, D. H. K. Tsang, B. Bensaou, and D. Gao, "Performance analysis of IEEE 802.11e contention-based channel access," Selected Areas in Communications, IEEE Journal on, Vol.22, No.10, pp.2095- 2106, Dec., 2004.

[8] Y. Tay and K. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," Wireless Networks, Vol.7, No.2, pp.159-171, Mar., 2001.

[9] Y. Lin and V. W. Wong, "Saturation throughput of IEEE 802.11e EDCA based on mean value analysis," in Wireless Communications and Networking Conference, 2006. in WCNC, Apr., 2006.

[10] "Monitor mode -wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Monitor mode

[11] "Network monitor." [Online]. Available: http://blogs.technet.com/ netmon

[12] R. Braden, "Requirements for internet hosts - communication layers," RFC 1122, October 1989. [Online]. Available: http://www.ietf.org/rfc/ rfc1122.txt?number=1122

서 성 훈

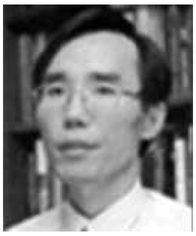
e-mail : hoon@cs.columbia.edu
2000년 한성대학교 정보시스템공학과
(공학사)
2002년 연세대학교 컴퓨터과학과
(공학석사)
2009년 연세대학교 컴퓨터과학과
(공학박사)
2009년~현 재 Columbia University 컴퓨터과학과
박사후연구원
관심분야 : 유/무선 통신 프로토콜 공학, 성능평가 등

백 재 종

e-mail : jjb27@emerald.yonsei.ac.kr
1996년 한밭대학교 전자계산학과(공학사)
2001년 연세대학교 대학원 컴퓨터과학과 (공학석사)
2007년~현 재 연세대학교 컴퓨터과학과 박사과정
관심분야 : 정보보호, 유/무선통신, 역공학, 사이버테러 등

김 동 건

e-mail : dg@emerald.yonsei.ac.kr
2006년 연세대학교 물리학과(학사)
2009년 연세대학교 컴퓨터과학과 공학석사
관심분야 : 정보보호, 이동통신, 네트워크 관리 등



송 주 석

e-mail : jssong@emerald.yonsei.ac.kr
1976년 서울대학교 전기공학과(학사)
1979년 한국과학기술원 전기전자공학 (석사)
1988년 Univ. of California at Berkeley, 컴퓨터과학(박사)
1988년~1989년 Assistant Professor in Naval Postgraduate School
1989년~현 재 연세대학교 컴퓨터과학과 정교수
2006년 한국정보보호학회 회장 역임
관심분야 : 유/무선 통신, 정보보호 등