QoS-Aware Channel Scanning for IEEE 802.11 Wireless LAN

Sang-Jo Yoo*, Nada Golmie, Haolang Xu
National Institute of Standards and Technology, Gaithersburg, MD
{sangjo.yoo, nada.golmie, haolangx}@nist.gov

Abstract – Channel scanning is an important aspect of seamless handovers since it is required in order to find a target point of attachment (PoA). In the IEEE 802.11 WLAN, scanning of other channels causes service disruptions with the current AP so that the provided quality of service (QoS) will be degraded seriously during the handoff. In this paper, we propose a QoS supported dynamic channel scanning algorithm. The scanning period is scheduled to guarantee the user's QoS requirements while the scan progresses. The simulation results show that the proposed mechanism reduces service disruptions and provides the desired quality of service to users during the scanning period.*

I. INTRODUCTION

During WLAN [1] handoffs, a certain level of service disruption in terms of delay and packet loss is simply unavoidable due to the channel scanning and link switching procedures that need to be performed in order to find a target AP and transition the connection to it. Since channel scanning can be a relatively time consuming procedure, the quality of service (QoS) degradation during a scan is a critical issue. Therefore, to support user QoS requirements and seamless handovers, service disruptions during the scanning procedure should be controlled and scheduled effectively.

Most previous scanning algorithms focused on reducing the scanning time [2-4]. Using either the previous channel selection history or information regarding the neighboring network topology, they aim at scanning fewer channels in an effort to reduce the handover latency. Few other scanning algorithms [5][6] were proposed for VoIP applications, in which scanning and VoIP traffic transmission are simply interleaved with VoIP delay requirements.

In this paper, we propose a new QoS-aware channel scanning mechanism for WLAN networks. To provide the desired QoS level to user applications while minimizing MAC layer delays and the packet loss ratio, the entire scanning period is scheduled based on the scanning policy and measured QoS level. The proposed scheduling method divides the entire required scanning period into several smaller scan times termed service interruption times. After each short scan during a SIT (service interruption time), the MS reverts back to the normal data transmission mode. The remainder of this paper is organized as follows: in Section II, we propose a new QoS-aware channel scanning mechanism. In Section III, simulation results are shown. We conclude this paper in Section IV.

II. QOS-AWARE CHANNEL SCANNING

A. QoS Supported Handoff Procedure

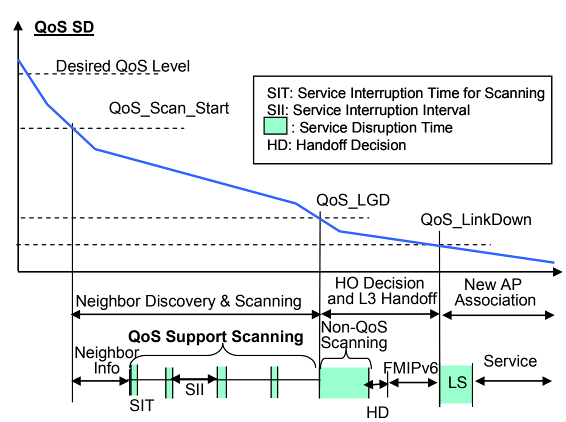
For QoS-supported scanning and handoff, a 'QoS satisfaction degree' is defined as a link quality metric in this paper. It is a function of QoS metrics as defined in (1). Each QoS component is a ratio between required and measured. The QoS satisfaction degree for class *c* can be defined as a minimum value from two QoS components or a weighted average depending on the user requirements. $QoS_c(t) \geq 1$ is desired.

$$QoS_c(t) = \mathrm{F}\left(\frac{\mathrm{R\_delay}}{\mathrm{M\_delay}(t)}, \frac{\mathrm{R\_loss}}{\mathrm{M\_loss}(t)}\right) \quad (1)$$

where, R_delay and R_loss are the required delay and loss ratio, respectively; M_delay(t) and M_loss(t) are the measured delay and loss ratio at time *t*.

In our approach, we separate the actual handoff execution that is most time critical behavior with QoS supported scanning. To start the QoS supported scanning, we have defined the QoS_Scan_Start link layer trigger. Fig. 1 shows the time sequence of the proposed approach. If the current measured QoS satisfaction degree is less than the QoS_Scan_Start threshold, then the MS sends query messages to the IEEE 802.21 information server (IS) [7] to obtain neighbor network information.

The neighbor information also can be obtained from the current serving AP using neighbor advertisement messages in WLAN. Neighbor report frame from the current AP includes the list of the neighbor APs. This neighbor AP information is stored in MS's scanning list. After obtaining neighbor AP information, the MS performs QoS supported scanning. To provide the desired QoS in terms of delay and loss ratio during the entire QoS supported scanning period, short time scanning and normal data transmission are interleaved.



---

* sjyoo@inha.ac.kr , Associate professor of Inha University, Korea.
"This research was supported by the NIST/Office of Law Enforcement Standards (OLES)"

978-1-4244-2644-7/08/$25.00 ©2008 IEEE

Figure 1. QoS supported scanning and handoff procedure.

When the measured QoS crosses the QoS Link_ Going_Down (QoS_LGD) trigger threshold and if a target PoA that can serve the QoS requirements of the user has not been found via QoS supported scanning, the MS performs non-QoS supported horizontal scanning. With non-QoS supported scanning, to find the target PoA, the MS consecutively scans all remained channels, as in this case the link down event is imminent and the scanning is a time-critical process. QoS_LGD trigger also activates layer 3 fast handoff protocol such as Fast Mobile IPv6 if the target AP is not on the same subnet of the current serving AP.

B. *QoS Supported Scanning Mechanism*

In the conventional horizontal channel scanning, candidate neighboring channels are generally scanned consecutively. A service interruption of several hundred milliseconds occurs, during which wireless stations cannot send or receive data packets. The main objective of the proposed QoS supported scanning is to minimize this type of disruptive scanning effect on application traffic and to guarantee the user QoS demands during scanning procedures. Instead of consecutively scanning all channels, the length of every scanning time and interval between scans are dynamically determined in the proposed scanning mechanism based on the QoS requirements and current QoS measurements.

Two scanning parameters are defined for each service class *c*, including the service interruption time (SIT) and service interruption interval (SII), as shown in Fig. 2. SIT and SII indicate the actual scanning interval (during which the current service is interrupted) and the time between two scanning intervals, respectively.

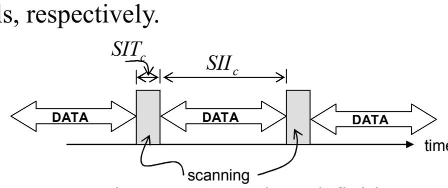


Figure 2. SIT and SII definition.

The desired QoS level in terms of the delay and loss ratio during the scanning period ($R\_delay_c^S$ and $R\_loss_c^S$) can be set by the user to be slightly lower than the QoS level under normal operation conditions, as in (2). In (3), R_d and R_l represent tolerable delay and loss degradation ratios, respectively.

$$R\_delay_c^S \geq R\_delay_c, \quad R\_loss_c^S \geq R\_loss_c \qquad (2)$$

$$\begin{cases} R\_delay_c^S = R_d \times R\_delay_c \\ 10\log_{10}\left(R\_loss_c^S\right) = \dfrac{10\log_{10}\left(R\_loss_c\right)}{R_l}, R_d, R_l \geq 1 \end{cases} \qquad (3)$$

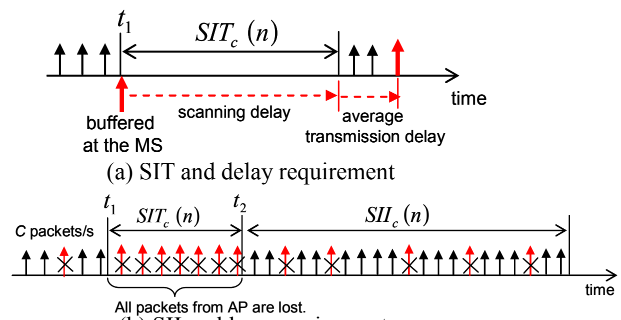
To adaptively derive the scanning parameters over time *t*, constant bit rate (CBR) applications are assumed. As shown in Fig. 3-a, during the SIT, the MS cannot send data to the currently serving PoA. The data packets are stored in the buffer of the MS and they can be sent after the SIT has finished. Therefore, the SIT generates additional delay. The delay during the scanning period should be kept lower than the desired delay $R\_delay_c^S$, as in (4).

$$\left(SIT_c\,(n) + M\_delay_c(t_1)\right) \leq R\_delay_c^S \qquad (4)$$

where the *n*-th SIT starts at time t_1 and $M\_delay_c(t_1)$ is the measured delay at time t_1 . Therefore, the maximum bound of the SIT is given as in (6).

$$SIT_c\,(n) \leq \left(R\_delay_c^S - M\_delay_c(t_1)\right) \qquad (5)$$

$$MaxSIT_c\,(n) = \left(R\_delay_c^S - M\_delay_c(t_1)\right) \qquad (6)$$



(a) SIT and delay requirement

(b) SII and loss requirement

Figure 3. SIT and SII derivation.

In WLAN, there is no management frame to inform the current serving AP of a pending scan by the MS; thus, all packets from the AP can be lost, as shown in the case in Fig. 3-b. When the SIT starts at t_1 and finishes at t_2 , as in Fig. 3-b, in order to guarantee that the packet loss ratio during $(t_1, t_1 + SIT_c(n) + SII_c(n)]$ is less than or equal to the desired loss level $\left(R\_loss_c^S\right)$, $SII(n)$ should be determined using (7) and the minimum value used in this case is given in (9).

$$\frac{\text{Average lost packets during } (t_1, t_1 + SIT_c(n) + SII_c(n)]}{\text{Transmitted packets during } (t_1, t_1 + SIT_c(n) + SII_c(n)]} \leq R\_loss_c^S$$
$$\frac{C \times SIT_c\,(n) + C \times SII_c\,(n) \times M\_loss_c(t_2)}{C \times \left(SIT_c\,(n) + SII_c\,(n)\right)} \leq R\_loss_c^S \qquad (7)$$

$$SII_c\,(n) \geq \frac{\left(1 - R\_loss_c^S\right) \times SIT_c(n)}{R\_loss_c^S - M\_loss_c(t_2)} \qquad (8)$$

$$MinSII_c\,(n) = \frac{\left(1 - R\_loss_c^S\right) \times SIT_c(n)}{R\_loss_c^S - M\_loss_c(t_2)} \qquad (9)$$

where, $M\_loss_c(t_2)$ is the measured loss ratio at time t_2 and C is the packet transmission rate. In the proposed mechanism, the measurement periods for $M\_delay(t)$ and $M\_loss(t)$ do not include SITs. Therefore, $M\_loss_c(t_2) = M\_loss_c(t_1)$.

In this paper, the scanning actions that can be performed during a SIT time are considered. A SIT can be either too short to handle even a single channel scan, or large enough for multiple channel scans. The new concept of '**independent scanning pieces**' is defined to address this. If two procedures (message exchanges) are not correlated (controlled by the same timer) or if the correlated timer values for the next message transmissions at the MS side are relatively long compared with the SII, they are considered as independent scanning pieces. For example, for active scanning in a WLAN, probe request and probe response messages are not independent and should be handled as a set. Fig. 4 shows an example of this. In order to scan one channel, six message exchanges are required. The message exchange M_1 and M_2 are controlled by timer T_1 and messages from M_3 to M_6 are controlled by correlated timers T_2, T_3, and T_4. It is assumed that the required time (T_2) to receive M_3 message after sending M_2 at the PoA is much larger than the SII value. Therefore, in this example we have two independent scanning pieces.

Let define sp_i be the i-th independent scanning piece, t_sp_i be the required time for independent scanning piece sp_i, T_i^j be the j-th timer of the i-th independent scanning piece, and $\overline{T_i} = \{T_i^1, T_i^2, \cdots T_i^R\}$ be a set of correlated timers for sp_i on the MS side. In Fig. 4, $\overline{T_2} = \{\mathrm{T}_2, \mathrm{T}_4\}$. The required time for the i-th independent scanning piece (t_sp_i) is derived as (10).

$$t\_sp_i = \sum_{k=1}^{R} T_i^k \tag{10}$$

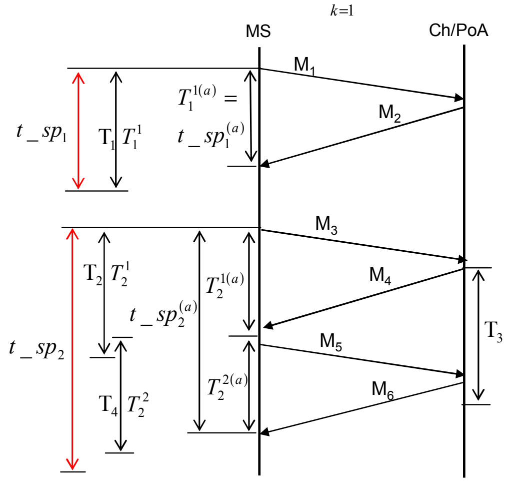


Figure 4. Independent scanning pieces.

The actual scanning time for sp_i can be less than t_sp_i because the MS can finish the required message exchanges before the related timer expirations. Let $T_i^{k(a)}$ be the actual time used by operations controlled by timer T_i^k. The actual time used by sp_i, $t\_sp_i^{(a)}$, is given as (11).

$$t\_sp_i^{(a)} = \sum_{k=1}^{R} T_i^{k(a)} \tag{11}$$

As in (6) and (9), $MaxSIT_c(n)$ and $MinSII_c(n)$ are determined at the n-th SIT and SII start time, respectively. However, $SIT_c(n)$ and $SII_c(n)$ are dynamically determined based on the actually used time for each independent scanning piece as shown in Fig. 5. The set of independent scanning pieces to scan the all required channels $\overline{S}$ is given as (12),

$$\overline{S} = \{sp_1, sp_2, \cdots, sp_{N\cdot S}\} \tag{12}$$

where N is the number of channels to scan and S is the number of independent scanning pieces to scan a single channel. The durations for $SIT_c(n)$ and $SII_c(n)$, at time t_1 and t_2, respectively, are derived as in Algorithm 1, where $\overline{V_n}$ is the set of scanning pieces that have been performed during $SIT_c(n)$.

ALGORITHM 1(Dynamic SIT and SII determination):

For the set $\overline{S}$, if during $SIT_c(n-1)$, independent scanning pieces up to sp_{l-1} have been performed, then

$\overline{V_n} = \{\phi\}$, $next = l$,

$MaxSIT_c(n) = (R\_delay_c^S - M\_delay_c(t_1))$

while $\left(\mathrm{sum}_{\forall sp_i \in \overline{V_n}} \{t\_sp_i^{(a)}\} + t\_sp_{next} \le MaxSIT(n)\right)$ {

Perform sp_{next};

$sp_{next} \rightarrow \overline{V_n}$; $next = next + 1$; }

Finish n-th SIT scanning,

$$SIT_c(n) = \mathrm{sum}_{\forall sp_i \in \overline{V_n}} \{t\_sp_i^{(a)}\} \tag{13}$$

The n-th SII time is

$$SII_c(n) = MinSII_c(n) = \frac{(1 - R\_loss_c^S) \times SIT_c(n)}{R\_loss_c^S - M\_loss_c(t_2)} \tag{14}$$

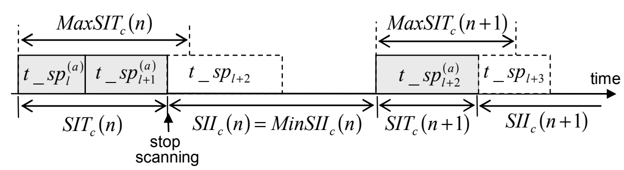


Figure 5. Dynamic SIT and SII determination.

To finish each independent scanning piece successfully with QoS guarantee, the required time of each independent piece scanning should be less than or equal to the SIT maximum bound, as in (15).

$$t\_sp_i \le MaxSIT_c(n), \quad \forall i \in \overline{S} \tag{15}$$

If t_sp_i is greater than $MaxSIT_c(n)$, then the MN must adjust some of timer values of $\overline{T_i}$ to fit $t\_sp_i \le MaxSIT_c(n)$. If timer adjustment is not available, then a vertical handoff procedure begins.

C. WLAN Case Study

During the actual scan, by receiving probe response frames from neighbor APs during an active scan, the MS can ensure that genuine connectivity to the neighbor APs exists, can estimate the possible transmission rate, and can recognize the QoS conditions of the neighbor APs. The probe response frame includes the basic service set (BSS) channel utilization, the available admission capacity, the average access delay of the BSS, and other parameters. After the MS finishes the proposed QoS supported scanning, it can select a target AP that currently provides highest level of QoS.

In a passive scan, the MS switches to a channel and simply waits for beacon frames from APs. As the default beacon interval is 100 ms, $100\text{ms} \times 11 = 1.1$ s is required to scan all

APs in 11 channels. Most current WLAN cards use active scanning to probe available APs. In an active scan, as shown in Fig. 6, the MS switches to each channel of the scanning list and waits for the indication of an incoming frame or for the ProbeDelay time (T_1) to expire. It then broadcasts a probe request frame on one channel after contending to the medium and starts a probe timer. If no activity is detected in the wireless media when the probe timer reaches MinChannelTime (T_2), the station determines that no AP is working in that channel and scans another channel. If the station detects that the channel is not idle, it will wait for probe response frames from working APs until the probe timer reaches MaxChannelTime (T_3). An empirical measurement shows that the ProbeDelay time is a few μ s, MinChannelTime is approximately 20ms, and that MaxChannelTime ranges from 30 to 40ms [2][12]. Therefore, in a WLAN, there exists only one independent piece to exchange probe request and probe response frames. The required time for the independent piece is given by (16).

$$t\_sp_1 = ProbeDelay\ \ Time + MaxChannel\ \ Time = \mathrm{T}_1 + \mathrm{T}_3 \tag{16}$$

As there may exist no AP that responds in certain channels, the actual used time for the sp_1 can be

$$t\_sp_1^{(a)} = \begin{cases} \mathrm{T}_1 + \mathrm{T}_2, \text{if there is no Probe Response} \\ \mathrm{T}_1 + \mathrm{T}_3, \text{if there is at least one Probe Response} \end{cases} \tag{17}$$

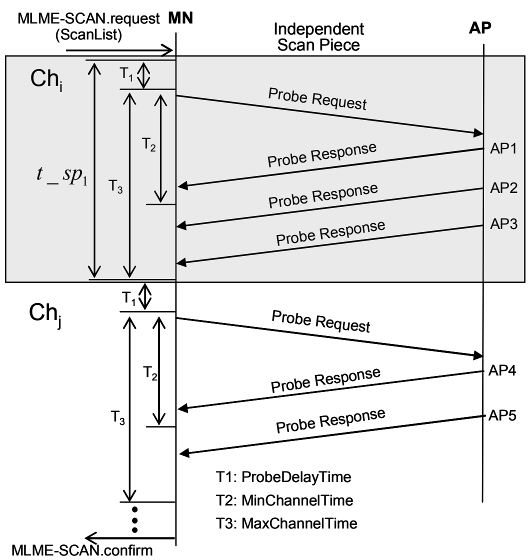


Figure 6. Scanning operations in a WLAN.

When $MaxSIT_c(n)$ is given, the minimum number of channels that the MS can scan during the *n*-th SIT is derived using (18).

$$N_{\min}^{(n)} = \left\lfloor \frac{MaxSIT_c(n)}{\mathrm{T}_1 + \mathrm{T}_3} \right\rfloor \tag{18}$$

In WLAN networks, if the current $MaxSIT_c(n)$ is too small to capture t_sp_1 , the following rule for the Layer 2 parameter adjustment is applied in the proposed scanning mechanism. Δ is the smallest marginal time to receive a probe response.

ALGORITHM 2(Layer 2 parameter adjustment in WLAN):

When

$(ProbeDelayTime + MaxChannelTime) > MaxSIT_c(n)$

if $MaxSIT_c(n) \le (ProbeDelayTime + \Delta)$

Start vertical scanning and perform a vertical handoff;

else if $MaxSIT_c(n) < (ProbeDelayTime + MinChannelTime)$

$MinChannelTime = MaxSIT_c(n) - ProbeDelayTime;$

$MaxChannelTime = MinChannelTime;$

else if $MaxSIT_c(n) < (ProbeDelayTime + MaxChannelTime)$

$MaxChannelTime = MaxSIT_c(n) - ProbeDelayTime;$

IV. Simulation Results

A. Numerical Results

In this numerical experiment, first the SIT and SII variations are evaluated based on various network conditions in terms of different measured QoS levels and different QoS classes. Second, the total scanning times are compared for different QoS classes. Table 1 shows the numerical analysis parameter values used. We assumed that there are 10 neighbor channels to scan.

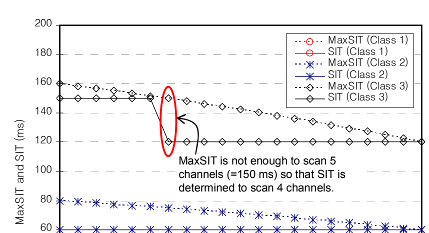
Table 1. Numerical analysis parameters

Parameter	Value
(R_d, R_l)	(2, 2)
R_delay	40ms(*c*=1),80ms(*c*=2),160ms(*c*=3)
R_loss	10^{-2}(*c*=1),$2*10^{-2}$(*c*=2),$4*10^{-2}$(*c*=3)
T_1+T_3	30 ms

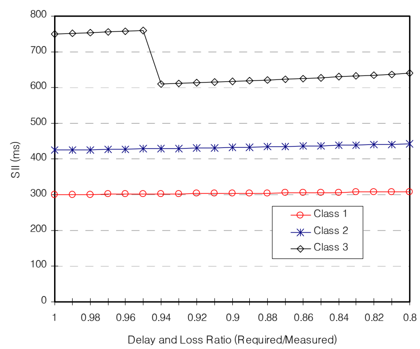
Fig. 7 shows the MaxSIT, SIT, and SII variations depending on different QoS conditions. As the measured QoS level decreases, the MaxSIT slowly decreases because a lower current QoS level requires a shorter scan. On the other hand, the SII slowly increases as the measured QoS level decreases. It is clear that, for a higher QoS class, a shorter MaxSIT is required. In Fig. 7-a, if the delay ratio is less than 0.95, the MaxSIT cannot include five independent scanning pieces. Hence, only four channels can be scanned.

Fig. 8 shows the SIT and SII variations for the different QoS requirements. The required delay and loss ratio vary from 20 ms to 120 ms and from 10^{-3} to 10^{-2}, respectively when the delay and loss ratio are fixed at 0.9. The smaller required delay makes the SIT smaller and results in a smaller SII for a given loss ratio. For a given delay requirement, a smaller loss ratio requires a larger SII. The SII does not increase proportionally to the required delay due to the restriction in the number of allowable independence scanning pieces.

Fig. 9 shows the proposed QoS support scanning time for the scanning of 10 neighbor channels. Here, is assumed that the measured QoS degradation follows exp(-0.06*t) function. A lower QoS class results in a shorter total scanning time and a longer SIT time (i.e., more channels scanned during a SIT).



(a) MaxSIT and SIT



(b) SII

Figure 7. MaxSIT, SIT, and SII variations.

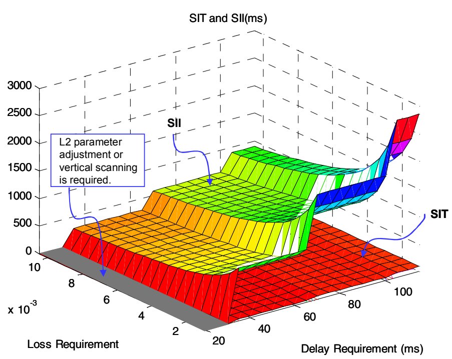


Figure 8. SIT and SII for different QoS requirements.

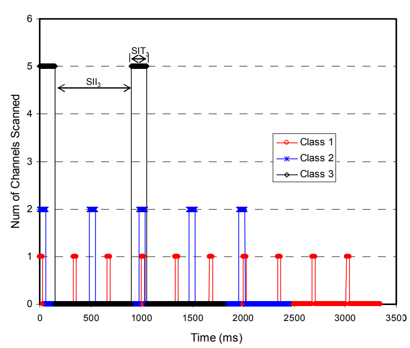


Figure 9. QoS support scanning.

B. NS-2 Simulation Results

In this simulation using NS-2, the network topology of Fig. 10 is used. In addition to the MS that is communicating with the corresponding node, there exist other 22 mobile stations generating background traffic. The aggregated background traffic increased until the total network traffic triggered the QoS scanning. Table 2 shows the simulation parameters. There exist 10 neighbor channels to scan.

Table 2. Simulation parameters

Parameter	Value
Delay and loss QoS requirements	40 ms, 10^{-2}
QoS_Scan_Start threshold	0.9
Wireless channel capacity	2 Mbps
Wired network data rate	100 Mbps
Data packet size	200 bytes
Data transmission rate	28 kbps (Poisson)

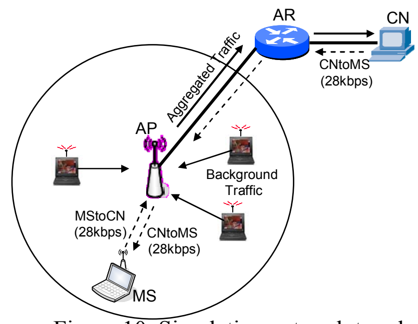


Figure 10. Simulation network topology.

Since we used RTS/CTS mechanism for data transmission in this WLAN simulation, the packet loss ratio was not dominant component compared with packet transmission delay in the QoS satisfaction degree computation. Fig. 11 shows the observed QoS satisfaction degree variations in time when we increase the aggregated network traffic.

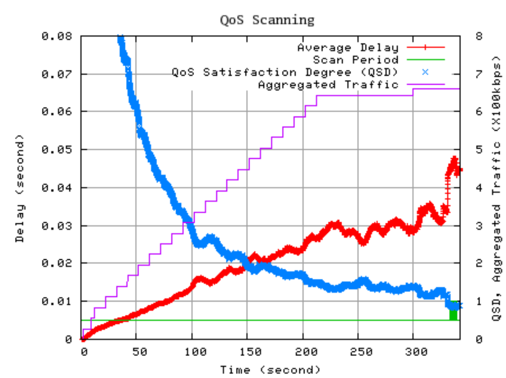
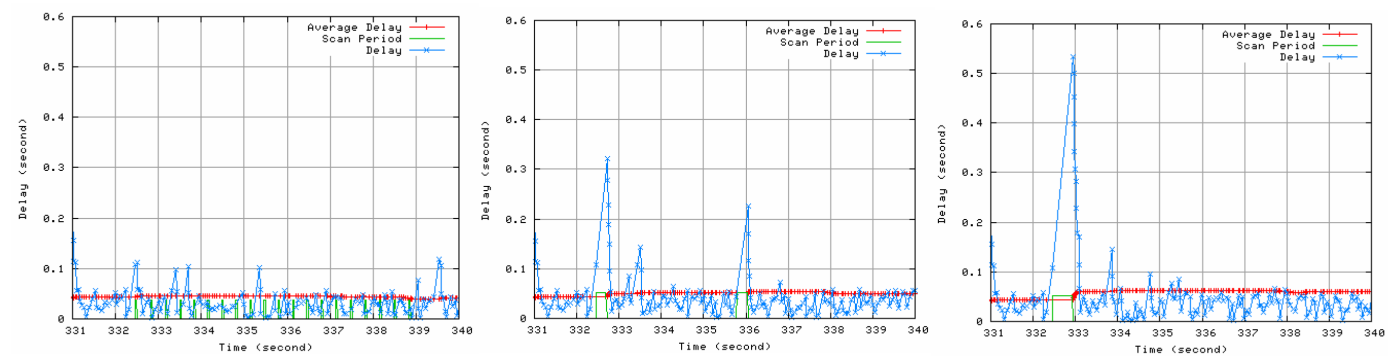


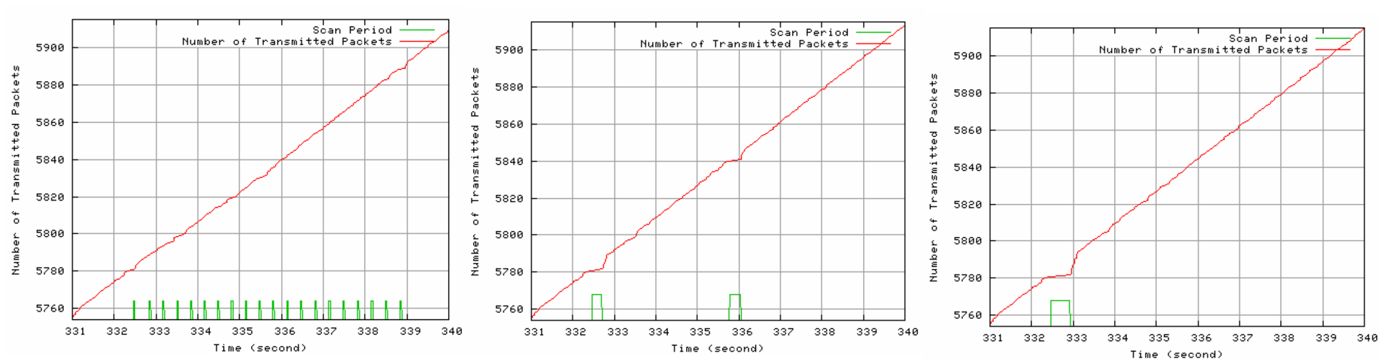
Figure 11. QoS satisfaction degree variations.

Fig. 12 shows the measured delay for different scanning methods. The MS performs two rounds of scanning, in which a round is for scanning all the neighboring channels (in this simulation, 10 channels). For performance comparison, we also implemented two types of consecutive scanning, in which the MS scans all channels of the neighbor networks consecutively. The consecutive scanning A starts the second round at the same time for the second round in the proposed QoS scanning. The consecutive scanning B executes the first and second rounds continuously without any time break. As shown in Fig. 12-(a), during the QoS scanning, the measured average and instance delay are maintained at the low level (less than the tolerable delay). In case of consecutive scanning, in Figs. 12(b)-(c), the MS can’t transmit packets during the long scanning time and the delay is significantly increased. Fig. 13 shows the total number of successfully transmitted packets by the MS. The MS is not able to send and receive data packets to and from the corresponding node during the



(a) QoS scanning (b) Consecutive scanning A (c) Consecutive scanning B

Figure 12. Measured delay.



(a) QoS scanning (b) Consecutive scanning A (c) Consecutive scanning B

Figure 13. Number of transmitted packets.

scanning periods. As we can see, since the proposed method interleaves short time scanning and normal data transmission in aware of possible delay and loss behaviors, the service disruption in terms of packet transmission is much smaller than the compared consecutive scanning methods. In the consecutive scanning, during the scanning period the packet transmissions from and to the MS are halted and the packets are buffered.

V. CONCLUSIONS

In this paper, we have proposed a new channel scanning mechanism in the IEEE 802.11 WLAN networks. To support the desired delay and loss ratio requirements during a scanning period, a short scanning time is interleaved after normal data transmissions. The actual scanning time (SIT) and its interval (SII) are adaptively determined by the QoS metric. All message exchange actions related to scanning are grouped as independent scanning pieces and only groups that can be handled during a single SIT interval are executed in the proposed scanning mechanism. The timing and manner of adjustments to the Layer 2 system parameters can be altered to meet the scanning requirements.

A simulation of the proposed technique showed the adaptive scanning parameter decision results for different QoS requirements and network QoS conditions. Higher QoS requirements need a shorter service disruption time and a longer scanning time interval. Compared to the conventional consecutive scanning method, the proposed QoS scanning provides low packet delay without discontinuity of packet transmissions.

REFERENCES

[1] IEEE, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE P802.11/D9.0, 2006.

[2] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, “Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs”, ACM International Workshop on Mobility Management & Wireless Access Protocols (MOBIWAC’04), pp. 19-26, 2004.

[3] Franco Tommasi, Simone Molendini, and Andrea Tricco, “Experience-driven Selective Scan for 802.11 Networks”, IEEE International Conference on Software in Telecommunications and Computer Networks (IEEE SoftCom’06), pp. 137-141, 2006.

[4] Yuh-Shyan Chen, Chung-Kai Chen, and Ming-Chin Chuang, “DeuceScan: Deuce-Based Fast Handoff Scheme in IEEE 802.11 Wireless Networks”, IEEE Vehicular Technology Conference (IEEE VTC’06 Fall), pp. 1-5, 2006.

[5] Haitao Wu, Kun Tan, Yongguang Zhang, and Qian Zhang, “Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN”, IEEE INFOCOM’07, pp. 749-757, 2007.

[6] Jeng-Ji Huang, Yi-Hsuan Chen, Sen-Ching Chang, and Huei-Wen Ferng, “An Efficient Channel Scan Scheduling Algorithm for VoIP Handoffs in WLANs”, IEEE Vehicular Technology Conference (IEEE VTC’07 Spring), pp. 1340-1344, 2007.

[7] IEEE, “Draft Standard for Local and Metropolitan Area Networks: Media Independent Handoff Services”, IEEE P802.21/D05.00, April 2007.