Effects of the EAPOL Timers in IEEE 802.1X Authentication

Ya-Chin Sung and Yi-Bing Lin, *Fellow, IEEE*

***Abstract*— This paper studies IEEE 802.1X authentication for WLAN and cellular integration. In the IEEE 802.1X standard, several timeout timers are defined for message exchanges in the EAPOL protocol, where the same fixed value is suggested for these timeout timers. We observe that the delays for the EAPOL message exchanges may significantly vary. A modeling study is performed to tune the values of individual timers to yield better performance than that for the identical timeout period setting. Our study provides guidelines to select appropriate timeout values for IEEE 802.1X operation.**

***Index Terms*— EAPOL, IEEE 802.1X.**

I. Introduction

THE 802.1X standard specifies authentication and authorization for IEEE 802 LAN [1], which has also been widely adopted for mobile devices to access *Wireless Local Area Network* (WLAN). Furthermore, if WLAN is integrated with cellular network (such as GSM or UMTS [2]), the SIM module (in the mobile device) and the *Authentication Center* (AuC) are utilized together with IEEE 802.1X for authentication. An example of WLAN and cellular integration (in terms of authentication) is illustrated in Fig. 1.

In this figure, the *Home Location Register* (HLR) is a mobility database that stores and manages all mobile subscriptions of a specific operator. The AuC provides seccurity data management for mobile subscribers. The AuC is typically collocated with the HLR. The *Access Point* (AP) provides radio access to a mobile device. Before a mobile device is authenticated, the AP only allows this mobile device to send the IEEE 802.1X authentication messages. When the *Remote Authentication Dial In User Service* (RADIUS) server receives an authentication request from the mobile device, it retrieves the authentication information of the mobile device from the HLR/AuC. After the HLR/AuC returns the authentication information, the RADIUS server authenticates the mobile device following the standard GSM/UMTS authentication procedure [3].

In IEEE 802.1X, the mobile device to be authenticated is called a *supplicant*. The server (typically a RADIUS server) performing authentication is called the *authentication server*. The *authenticator* (e.g., a wireless access point) facilitates authentication between the IEEE 802.1X supplicant and the authentication server. The integrated system utilizes the *Extensible Authentication Protocol* (EAP) to support multiple authentication mechanisms based on the challenge-response paradigm [4]. The IEEE 802.1X supplicant encapsulates the EAP packets in *EAP over LAN* (EAPOL) frames before they are transmitted to the authenticator. Upon receipt of an EAPOL frame, the authenticator decapsulates the EAP packet from the EAPOL frame. Then the EAP packet is sent to the authentication server using the RADIUS protocol [5]. Implemented on top of UDP, RADIUS provides mechanisms for per-packet authenticity and integrity verification between the authenticator and the authentication server.

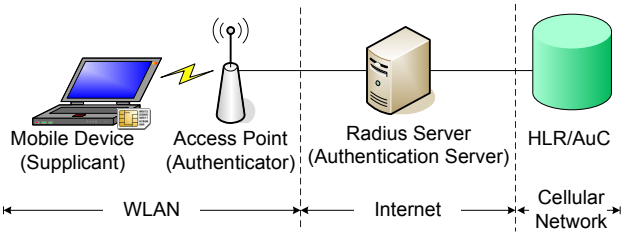


Fig. 1. A WLAN and cellular integration environment

IEEE 802.1X authentication for the WLAN and cellular integration network has been investigated in [6], [7] and [8]. These studies focused on the design of the network integration architectures, and proposed IEEE 802.1X authentication procedures for the integration network. This paper describes IEEE 802.1X authentication that enhances the WLAN security by allowing a mobile device to be authenticated before it is assigned an IP address.

In our solution, the WLAN and cellular integration network in Fig. 1 employs EAP-SIM authentication, which is an EAP-based authentication protocol utilizing the GSM *Subscriber Identity Module* (SIM) [9]. In GSM, a secret key Ki is stored in the HLR/AuC as well as in the SIM. The authentication server communicates with the HLR/AuC to obtain the GSM authentication information through the *Mobile Application Part* (MAP) implemented on top of the *Signaling System Number 7* (SS7) [10]. In the EAP-SIM authentication, the MAP is responsible for retrieving the GSM authentication information in the HLR/AuC. In the implementation of IEEE 802.1X authentication for WLAN, we observe that the elapsed times for authentication message pairs exchanged between the mobile device and the network are different. In IEEE 802.1X specification, the message pairs are associated with fixed timeout timers. We analyze the timeout timers used in IEEE 802.1X authentication and improve the performance of

Manuscript received October 13, 2005; accepted May 20, 2006. The associate editor coordinating the review of this paper and approving it for publication was Z. Haas.

Y.-C. Sung is with National Chiao Tung University, Room 117, Engineering Building III, Hsinchu, Taiwan (e-mail: ycsung@csie.nctu.edu.tw).

Y.-B. Lin is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan. He is also with the Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan (e-mail: liny@csie.nctu.edu.tw).

Digital Object Identifier 10.1109/TWC.2007.05814.

IEEE 802.1X authentication by selecting appropriate timer values.

II. SIM-Based IEEE 802.1X Authentication

This section describes the SIM-based IEEE 802.1X authentication procedure. The authentication message flow is illustrated in Fig. 2, which consists of the following steps:

Stap 1. The mobile device (the supplicant) sends the EAPOL-Start packet to the AP to initiate the IEEE 802.1X authentication.

Stap 2. The AP requests the identity of the mobile device through the EAP-Request message with type Identity. When the AP receives the EAP-Response/Identity message from the mobile device, it encapsulates this message in the Access-Request packet. Then the packet is sent to the RADIUS server.

Stap 3. The RADIUS server conducts the EAP-SIM authentication with the mobile device (the supplicant). Specifically, the RADIUS server generates an Access-Challenge packet that encapsulates the EAP-Request/SIM/Start in the EAP-Message attribute and sends it to the AP. This message requests the mobile device to initiate the EAP-SIM authentication. The AP decapsulates the EAP-Request from the Access-Challenge packet, and delivers it to the mobile device by an EAPOL packet.

Stap 4. The mobile device responds the RADIUS server with the EAP-Response/SIM/Start message containing a random nonce AT_NONCE_MT. The random nonce will be used to generate the encryption key for data transmission between the mobile device and the RADIUS server after the IEEE 802.1X authentication.

Stap 5. To obtain the authentication information of the mobile device, the RADIUS server sends the SS7 MAP_Send_Authentication_Info_Request message (with the argument IMSI) to the HLR/AuC. The HLR returns the authentication vector (*RAND*, *SRES*, *Kc*) through the SS7 MAP_Send_Authentication_Info_Response message, where *RAND* is a random number generated by the HLR/AuC. The authentication vector will be used at **Step 7** and **8** to exercise the GPRS authentication [10].

Stap 6. The RADIUS server sends the EAP-Request/SIM/Challenge (with the parameter *RAND*, which is encapsulated as AT_RAND) to the mobile device. To ensure the integrity of the challenge message, the message contains a *Message Authentication Code* (MAC) AT_MAC.

Stap 7. After verifying AT_MAC received from the RADIUS server, the mobile device passes the random number *RAND* to the SIM module to perform GSM authentication. The SIM module computes its signed result *SRES** and the encryption key *Kc** based on the received *RAND* and the authentication key *Ki* stored in the SIM card. Then the mobile device sends *SRES** (encapsulated in AT_MAC) to the RADIUS server through the EAP-Response/SIM/Challenge message.

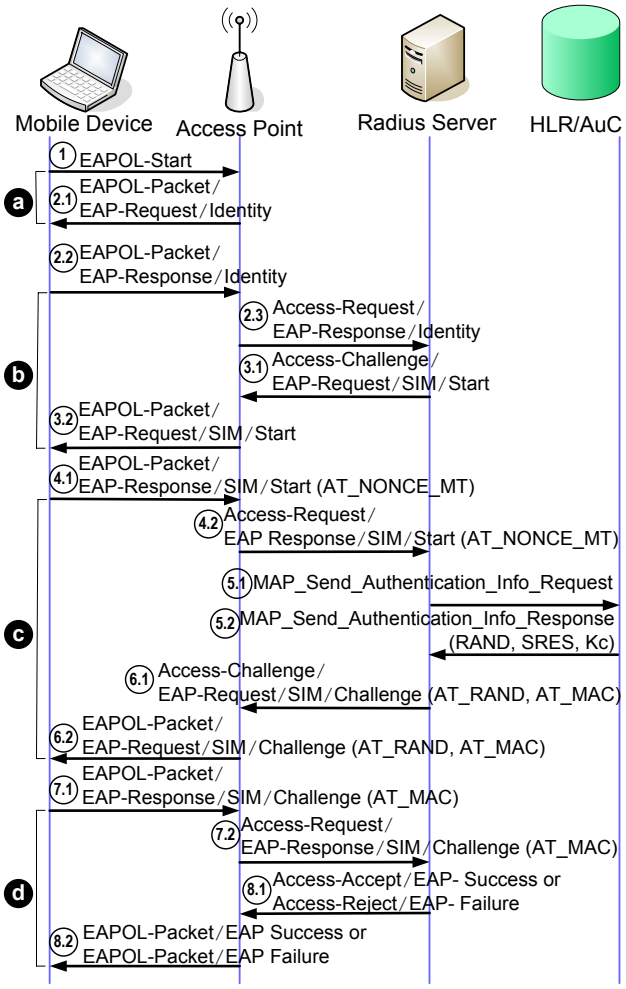


Fig. 2. SIM-based IEEE 802.1X Authentication Message Flow

Stap 8. The RADIUS server verifies AT_MAC and compares *SRES** calculated by the mobile device with the *SRES* received from the HLR/AuC. If the values are identical, the RADIUS server notifies the AP that authentication is successful through the EAP-Success message (encapsulated in the RADIUS Access-Accept packet). The AP passes the EAP-Success message to the mobile device. At this point, the mobile device is allowed to access the network through the AP. If the signed results are not the same, the RADIUS server notifies the AP that the authentication fails through the EAP-Failure message (encapsulated in the RADIUS Access-Reject packet).

III. EAPOL Timers

In the IEEE 802.1X supplicant (mobile device), three EAPOL timers are defined:

1) *startWhen* (associated with message pair (a) in Fig. 2): When the IEEE 802.1X supplicant initiates the authentication, it sends EAPOL-Start to the authenticator and

starts the *startWhen* timer. If the supplicant has not received any response from the authenticator after this timer expires, it resends EAPOL-Start. The supplicant gives up when it sends EAPOL-Start for n_1 times. In the IEEE 802.1X specification [1], the default n_1 value is 3. The default value of the *startWhen* timer is 30 seconds.

2) *authWhile* (associated with message pairs (b), (c), and (d) in Fig. 2): Every time the supplicant sends an authentication message (Steps 2.2, 4.1, and 7.1 in Fig. 2), it starts the *authWhile* timer. If the supplicant does not receive any response from the authenticator after this timer has expired, the supplicant sends an EAPOL-Start message to re-start the authentication procedure. The supplicant gives up after it has consecutively sent EAPOL-Start for n_2 times. The default n_2 value is 3. The default value of the *authWhile* timer is 30 seconds.
3) *heldWhile* (associated with Step 8.2 message in Fig. 2 if the client fails the authentication): If the IEEE 802.1X authentication fails, the supplicant has to wait for a period *heldWhile* before it re-starts the authentication procedure. The default value of the *heldWhile* timer is 60 seconds.

Selection of the EAPOL timer values is not trivial. If the timer value is too large, it will take long time before the mobile device detects the failure of the network (e.g., RADIUS server failure). If the timer value is too small, the timer may expire before the mobile device receives the response message. In this case, the mobile device needs to re-start the authentication process due to *false failure detection*.

Table I shows the expected *Round-Trip Times* (RTTs) of message exchanges that measured from the implementation in National Chiao Tung University. These measurements do not experience waiting delays due to queuing at the network nodes (i.e., AP, RADIUS server and HLR/AuC).

In our measurement, the mobile device and the AP are located in one subnet. The RADIUS server and the HLR are located in another subnet. The transmission rate of the wired network is 100Mbps. It is observed that the RTT of a message exchange between the mobile device and the RADIUS server are much shorter than that of a message exchange between the mobile device and the HLR/AuC. This significant RTT discrepancy is due to the fact that accessing the HLR/AuC is much more time-consuming than accessing the RADIUS server. This phenomenon is especially true when the HLR/AuC is fully loaded by cellular user accesses and when the RADIUS server and the HLR/AuC are located at different cities or different countries. To reduce the false failure detection probability without non-necessary timer timeout delay, the values of the *startWhen* timer and *authWhile* timers should not be identical for all message exchanges in the IEEE 802.1X authentication. For example, the *authWhile* timer for (c) in Table I should be different from that for (b) and (d).

TABLE I
EXPECTED ROUND-TRIP TIMES FOR EAP-SIM AUTHENTICATION MESSAGES (WITHOUT QUEUING DELAYS)

Events occurring at the mobile device	Associated timer	RTT (sec.) (no queueing)
(a) in Fig. 2	*startWhen*	0.005
(b) in Fig. 2	*authWhile*	0.013
(c) in Fig. 2	*authWhile*	1.087
(d) in Fig. 2	*authWhile*	0.013

TABLE II
INPUT PARAMETERS AND OUTPUT MEASURES

Input Parameters				
message pair	associated timeout period	service time of message exchange	response time of message exchange	timeout probability
(a)	T_s	t_s	τ_s	$p_s = Pr[\tau_s \geq T_s]$
(b)	T_{a_1}	t_{a_1}	τ_{a_1}	$p_{a_1} = Pr[\tau_{a_1} \geq T_{a_1}]$
(c)	T_{a_2}	t_{a_2}	τ_{a_2}	$p_{a_2} = Pr[\tau_{a_2} \geq T_{a_2}]$
(d)	T_{a_3}	t_{a_3}	τ_{a_3}	$p_{a_3} = Pr[\tau_{a_3} \geq T_{a_3}]$
Output Measures				
p_f	the false failure detection probability of the IEEE 802.1X authentication procedure; p_f = Pr[the mobile device has consecutively sent the EAPOL-Start frame for three times]			
E_τ	the expected response time of the IEEE 802.1X authentication procedure			

IV. PERFORMANCE MODELING

We investigate the *false failure detection* probability p_f of the IEEE 802.1X authentication procedure and the expected elapsed (response) time E[τ] for the IEEE 802.1X authentication procedure. Input parameters and output measures are listed in Table II. Let $F_X^*(s)$ be the Laplace Transform for the response time τ_X distribution, where $X = s, a_1, a_2, a_3$.

$$F_X^*(s) = B_X^*(s) \sum_{k=0}^{\infty} (1 - \lambda E[t_X])(\lambda E[t_X])^k \left(\frac{1 - B_X^*(s)}{sE[t_X]} \right)^k, \quad (1)$$

The density function $f_X(t_X)$ can be obtained by inverting the Laplace Transform $F_X^*(s)$ in (1). Therefore, we have

$$f_X(t_X) = b_X(t_X) \sum_{k=0}^{\infty} (1 - \lambda E[t_X])(\lambda E[t_X])^k \hat{b}_k(t_X), \quad (2)$$

where $\hat{b}_k(t_X)$ is the k-fold convolution of the function $\frac{1 - B_X(t_X)}{E[t_X]}$.

Let T_X be the timeout period associated with the timer for the message pair X and p_X be the timeout probability of the message exchange.

$$p_X = Pr[\tau_X \geq T_X] = \int_{T_X}^{\infty} f_X(t)dt \quad (3)$$

From (3), the expected response time $E[\tau_X]$ of the message exchange can be expressed as

$$E[\tau_X] = p_X T_X + (1 - p_X) \int_0^{T_X} t f_X(t)dt \quad (4)$$

The probability transition diagram of the mobile device is illustrated in Fig. 3. In IEEE 802.1X, the AP can also control the number of retransmissions for EAPOL-Start sent from the mobile device to the AP ((1) in Fig. 2). To simplify our discussion, we assume that the number of retransmissions is

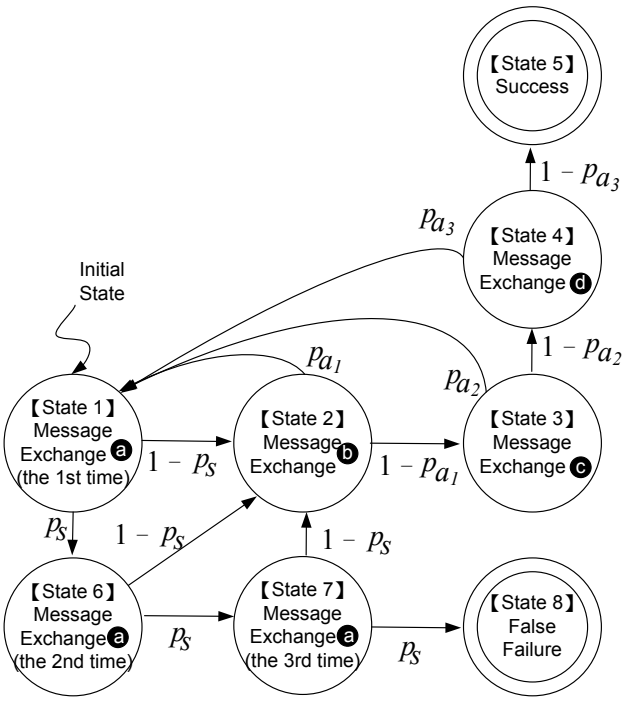


Fig. 3. The probability transition diagram of the IEEE 802.1X authentication message exchange.

sufficiently large, so that the state diagram in Fig. 3 is not affected.

During IEEE 802.1X authentication, the mobile device restarts the procedure (i.e., come back to state <1> again) whenever the *authWhile* timer (associated with message exchanges (b), (c), and (d)) expires. The authentication exits and is considered failed if the *startWhen* timer (associated with message exchange (a)) has consecutively expired for three times (i.e., the *finite state machine* (FSM) moves to state <8>).

Let x be the probability that the FSM starts from state <1> and eventually comes back to state <1> (i.e., state <1> may be revisited zero or more times). All possible scenarios for the probability transitions in Fig. 3 are described as follows:

- **Scenario I:** From state <1> (i.e., state <1> may have been visited zero or more times), the *startWhen* timer consecutively expires for three times (i.e., the last transitions are <1>→<6>→<7>→<8>. The probability for Scenario I is xp_s^3.
- **Scenario II:** From state <1>, the *startWhen* timer consecutively expires for two times, and the procedure successes at the third try (i.e., the last transitions are <1>→<6>→<7>→<2>→<3>→<4>→<5>. The probability for Scenario II is $xp_s^2(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})$.
- **Scenario III:** From state <1>, the *startWhen* timer expires once, and the procedure successes at the second try (i.e., the last transitions are <1>→<6>→<2>→<3>→<4>→<5>. The probability for Scenario III is $xp_s(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})$.
- **Scenario IV:** From state <1>, the procedure successes without incurring any timer expiration (i.e., the last transitions are <1>→<2>→<3>→<4>→<5>. The probability for Scenario IV is $x(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})$.

TABLE III
THE p_X VALUES: ANALYSIS VERSUS SIMULATION ($T_X = 10 \times E[t_X]$, $var[t_X] = E[t_X]^2$, AND X = s, a_1, a_2, OR a_3).

λ (Unit: $\frac{1}{E[t_X]}$)	0.2	0.4	0.6	0.8
Simulation	0.0003	0.0025	0.0183	0.1353
Analytic	0.0004	0.0027	0.0196	0.1271
Error	0.0001	0.0002	0.0013	0.0082

It is apparent that the false failure probability p_f is the probability that Scenario I occurs. The success probability $(1-p_f)$ is the probability of either Scenarios II, III, or IV. That is,

$$p_f = Pr[Scenario\ I] = xp_s^3 \tag{5}$$

and

$$\begin{aligned} &1-p_f \\ &= Pr[Scenarios\ II,\ III,\ or\ IV] \\ &= x(p_s^2+p_s^1+1)(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3}) \\ &= x(1-p_s^3)(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3}) \end{aligned} \tag{6}$$

From (5) and 6), we have

$$p_f+(1-p_f) = x[p_s^3+(1-p_s^3)(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})] \tag{7}$$

By rearranging (7), we have

$$x = \frac{1}{p_s^3+(1-p_s^3)(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})} \tag{8}$$

From (8) and (5),

$$p_f = \frac{p_s^3}{p_s^3+(1-p_s^3)(1-p_s)(1-p_{a_1})(1-p_{a_2})(1-p_{a_3})} \tag{9}$$

By using (3) and (9), the value of p_f can be computed from λ, f_s, f_{a_1}, f_{a_2}, and f_{a_3}.

The above analytic model is validated against simulation experiments. The simulation model follows the discrete event approach [11], and the details are omitted. Table III indicates that the analytic and the simulation results are consistent (the errors are within 1%). Therefore, both the analytic model and the simulation implementation are validated.

V. NUMERICAL EXAMPLES

This section uses numerical examples to investigate the impact of timers on the performance of IEEE 802.1X authentication where the expected service times of the EAPOL message exchanges $E[t_s]$, $E[t_{a_1}]$, $E[t_{a_2}]$, and $E[t_{a_3}]$ are obtained from the measurements as listed in Table I. Other parameters include the EAPOL message arrival rate λ , and the variances of the EAPOL service times (i.e., $var[t_X]$, where X = s, a_1, a_2, or a_3). We have the following observations.

- **Observation 1.** The probability p_f is mainly affected by p_s. From the transitions of the FSM in Fig. 3, it is clear

that if p_{a_1}, p_{a_2}, $p_{a_3} > 0$, we have $\lim_{p_s \to 1} p_f = 1$ and $\lim_{p_s \to 0} p_f = 0$.

- **Observation 2.** The probabilities p_{a_1}, p_{a_2}, and p_{a_3} have indirect impact on p_f . Increasing p_{a_1}, p_{a_2}, and p_{a_3} will increase the probability that the FSM moves toward state <1> (i.e., loops among states <1>, <2>, <3>, and <4>) and thus decrease the probability that the FSM enters state <5>. However, the effect in Observation 2 is not as significant as that in Observation 1.

Based on simulation experiments, Table IV shows how timeout period T_X (X = s, a_1, a_2, or a_3) affects p_f when the variance of t_X is large (i.e., $var[t_X] = 100 \times E[t_X]^2$). These results in Table IV are consistent with Observations 1 and 2. That is, T_S has more significant effect on p_f than T_{a_1}, T_{a_2}, and T_{a_3} do, especially when the EAPOL message arrival rate λ is high. Specifically, when $\lambda = 0.975 \times E[t_{a_2}]^{-1}$ and if T_{a_1}, T_{a_2}, and T_{a_3} are fixed to 10 seconds, changing the value of T_S from 5 seconds to 15 seconds will reduce p_f from 95.69% to 50.00% (about 50% improvement). Conversely, changing any of the values for T_{a_1}, T_{a_2}, and T_{a_3} from 5 to 15 seconds only insignificantly affects p_f (less than 13% improvement).

When $var[t_X]$ is small (e.g., $var[t_X]$ is less than $E[t_X]^2$), the service time t_X does not significantly vary, and the IEEE 802.1X authentication message exchange is more likely to complete if the T_X value is set larger than the expected service time $E[t_X]$ of the corresponding EAPOL message pair exchange. Therefore, we have the following observation.

- **Observation 3.** When $var[t_X]$ is small and T_X is sufficiently large, changing T_X only insignificantly affects p_f.

Table IV shows how T_X and $var[t_X]$ affect $E[\tau]$ when the EAPOL message arrival rate λ is set to $0.5 \times E[t_{a_2}]^{-1}$. From this table, we have the following observation.

- **Observation 4.** When $var[t_X]$ is small (i.e., $var[t_X] = E[t_X]^2$), $E[\tau]$ is insignificantly affected by the change of T_X if T_X is larger than the expected response time of the EAPOL message exchanges. This result is similar to that in Observation 3.

We also observe two effects on T_X .

- **Effect 1.** When T_X is increased, the probability p_X is decreased, and the number of loopings among states <1>, <2>, <3>, and <4> is reduced.

When the variance of service time $var[t_X]$ is increased, more large t_X periods are expected. Therefore, even if T_X is increased, these long EAPOL message delays still result in timeouts, and p_f cannot be significantly reduced. Therefore, we have

- **Observation 5.** Effect 1 is more significant when $var[t_X]$ is small than when $var[t_X]$ is large.
- **Effect 2.** When T_X is increased, if timeout does occur, the non-necessarily waiting for timeout is increased and $E[\tau]$ is increased.

Based on the above discussion, we examine the results in Table V as follows.

- **Observation 6.** If $var[t_X]$ is large and T_X is not much larger than $E[t_X]$ (e.g., 5 seconds $< T_X <$ 10 seconds), then increases T_X only insignificantly decreases p_f (indicated in Observation 5), but significantly increases extra waiting period for timeout, as described in Effect 2. Therefore $E[\tau]$ is significantly increased. This phenomenon occurs when changing T_{a_2} in Table V, where $E[T_{a_2}]$ = 1.087 seconds.

TABLE IV
EFFECTS OF T_X ON p_f ($var[t_X] = 100 \times E[t_X]^2$; X = s, a_1, a_2, OR a_3).

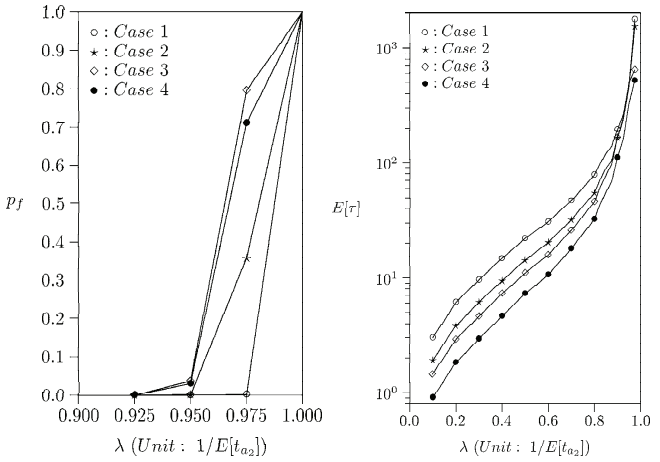
Timeout Timers (Unit:second)				Arrival Rate λ (Unit: $\frac{1}{E[t_{a_2}]}$)		
T_s	T_{a_1}	T_{a_2}	T_{a_3}	0.900	0.950	0.975
5	10	10	10	0.0199	0.3473	0.9569
15	10	10	10	0.0000	0.0023	0.5000
10	5	10	10	0.0000	0.0574	0.8676
10	15	10	10	0.0000	0.0314	0.7510
10	10	5	10	0.0000	0.0416	0.8140
10	10	15	10	0.0000	0.0344	0.7801
10	10	10	5	0.0000	0.0582	0.8679
10	10	10	15	0.0000	0.0317	0.7527

TABLE V
EFFECTS OF T_X AND $var[t_X]$ ON $E[\tau]$ (X = s, a_1, a_2, OR a_3; $\lambda = 0.5 \times E[t_{a_2}]^{-1}$).

Timeout Timers (Unit:second)				$var[t_X] = E[t_X]^2$		$var[t_X] = 100 \times E[t_X]^2$	
T_s	T_{a_1}	T_{a_2}	T_{a_3}	$E[\tau]$	effects	$E[\tau]$	effects
5	10	10	10	2.24	decreasing 0.45%	11.00	decreasing 0.09%
15	10	10	10	2.23		0.99	
10	5	10	10	2.24	decreasing 0.45%	10.96	decreasing 0.36%
10	15	10	10	2.23		11.00	
10	10	5	10	2.25	decreasing 0.89%	7.26	increasing 96.01%
10	10	15	10	2.23		14.23	
10	10	10	5	2.24	decreasing 0.45%	11.17	decreasing 1.70%
10	10	10	15	2.23		10.98	

When $E[t_X] \ll T_X$, then even if $var[t_X]$ is large, it is still likely that $t_X < T_X$, and both Effects 1 and 2 are not significant. Instead, the situation is similar to that in Observation 4. That is, $E[\tau]$ is insignificantly affected by $var[t_X]$ and T_X when $E[t_X] \ll T_X$. The phenomena occur when changing T_s, T_{a_1}, and T_{a_2} in Table V, where these timeout periods T_X are larger than $500 \times E[t_X]$.

A comparison of different timers settings is shown in Fig. 4. In Cases 1, 2, and 3, the timers in different message pairs are set to identical values, which means that a fixed *authWile* timer is used in message pairs (b), (c), and (d) (as suggested in IEEE 802.1X standard). In Case 4, we adjust the values of the timers according to the previous discussions to obtain better performance. For the illustration purpose, It is assumed that the variance of service time $var[t_X] = 100 \times E[t_X]^2$. Similar results are observed for different variances. Fig. 4 indicates that the false failure detection probability p_f is zero if the timeout period T_s is larger than 10 seconds and the EAPOL message arrival rate λ is below $0.925 \times E[t_{a_2}]^{-1}$. Theses figures indicate that when $\lambda < 0.925 \times E[t_{a_2}]^{-1}$, Case 4 has the same p_f performance as Cases 1 - 3, but



(a) (b)

Fig. 4. Effects of different timeout period settings when $var[t_X] = 100 \times E[t_X]^2$ (X = s, a_1, a_2, or a_3), where Case 1 ($\circ$): $T_s = T_{a_1} = T_{a_2} = T_{a_3} = 30$, Case 2 ($\star$): $T_s = T_{a_1} = T_{a_2} = T_{a_3} = 15$, Case 3 ($\diamond$): $T_s = T_{a_1} = T_{a_2} = T_{a_3} = 10$, and Case 4 ($\bullet$): $T_s = 10$, $T_{a_1} = 10$, $T_{a_2} = 5$, $T_{a_3} = 30$.

has much better $E[\tau]$ performance than these three Cases. When $\lambda > 0.925 \times E[t_{a_2}]^{-1}$, Case 4 improves $E[\tau]$ at the cost of degrading p_f as compared with Cases 1 and 2. For all λ values, Case 1 outperforms Case 2, and Case 2 outperforms Case 3 in terms of the p_f measure. For the $E[\tau]$ performance, the result reverses. In Case 3, the total timeout value $T_s + T_{a_1} + T_{a_2} + T_{a_3} = 40$ seconds. For Case 4, the total timeout value is 55 seconds. It is interesting to note that for all λ values, Case 4 outperforms Case 3 for both p_f and $E[\tau]$ performances. Also note that when $\lambda > 0.925 \times E[t_{a_2}]^{-1}$, the system is saturated, and will not be allowed in most commercial operations.

In these numerical examples, we demonstrate that appropriate T_X values can be selected through our modeling study to yield better performance than the fixed T_X value setting.

VI. Conclusion

This paper described IEEE 802.1X authentication for WLAN and Cellular integration. In the IEEE 802.1X standard, a fixed-value timer is used in all authentication message exchanges, which does not reflect the real network operation. A modeling study was presented to tune the values of individual timers, which yields better performance than the fixed timeout period setting.

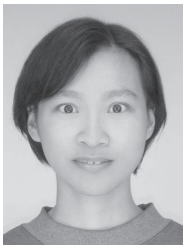
Our study provides guidelines to select appropriate timeout periods for corresponding authentication message exchanges. For example, comparing with the fixed timeout periods setting where T_X are set to 10 seconds, the suggested setting for the timeout periods (i.e., T_s = 10 seconds, T_{a_1} = 10 seconds, T_{a_2} = 5 seconds, and T_{a_3} = 30 seconds) decreases both the false failure detection probability p_f and the expected response time $E[\tau]$ of the IEEE 802.1X authentication procedure.

Acknowledgment

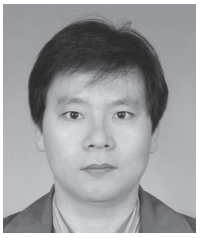
This work was sponsored in part by the NSC Excellence project NSC 95-2752-E-009-005-PAE, NSC 94-2219-E-009-001, NSC 94-2219-E-009-024, NTP SIP-based B3G project under grant number NSC 95-2219-E-009-010, NTP IMS Integration Project under grant number NSC 95-2219-E-009-019, Intel, Chung Hwa Telecom, IIS/Academia Sinica, the ITRI/NCTU Joint Research Center, and MoE ATU.

References

[1] *IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control*, LAN/MAN Standards Committee of the IEEE Computer Societ Std. 802.1X, 2001.
[2] Y.-B. Lin and A.-C. Pang, *Wireless and Mobile All-IP Networks*. New York: John Wiley & Sons, Inc., 2005.
[3] Y.-B. Lin and Y.-K. Chen, "Reducing authentication signaling traffic in third generation mobile network," *IEEE Trans. Wireless Commun.*, vol. 2, no. 3, pp. 493–151, May 2003.
[4] *Extensible Authentication Protocol (EAP)*, IETF Std. RFC 3748, 2004.
[5] *Remote Authentication Dial In User Service (RADIUS)*, IETF Std. RFC 2865, 2000.
[6] A. K. Salkintzis, C. Fors, and R. Pazhyannur, "WLAN-GPRS integration for next-generation mobile data networks," *IEEE Wireless Commun. Mag.*, vol. 9, no. 5, pp. 112–124, Oct. 2002.
[7] K. Ahmavaara, H. Haverinen, and R. Pichna, "Interworking architecture between 3GPP and WLAN systems," *IEEE Commun. Mag.*, vol. 41, no. 11, pp. 74–81, Nov. 2003.
[8] A. K. Salkintzis, "Interworking techniques and architectures for WLAN/3G integration toward 4G mobile data networks," *IEEE Wireless Commun. Mag.*, vol. 11, no. 3, pp. 50–61, June 2004.
[9] IETF, "Extensible authentication protocol method for gsm subscriber identity modules (eap-sim)," IETF Draft, June 2002.
[10] Y.-B. Lin and I. Chlamtac, *Wireless and Mobile Network Architectures*. New York: John Wiley & Sons, Inc., 2001.
[11] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*. Upper Saddle River, NJ: Prentice Hall, 2001.



Ya-Chin Sung received the BSCSIE and MSCSIE degrees from National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 2002 and 2003, respectively. She is currently working toward the Ph.D. degree at NCTU. Her current research interests include design and analysis of personal communications services networks, network security and performance modeling.



Yi-Bing Lin (M'96-SM'96-F'04) is Chair Professor and Dean of Computer Science College, National Chiao Tung University. His current research interests include wireless communications and mobile computing. Dr. Lin has published over 200 journal articles and more than 200 conference papers. He is the coauthor of the books *Wireless and Mobile Network Architecture* with Imrich Chlamtac (New York: Wiley, 2001) and *Wireless and Mobile All-IP Networks* with Ai-Chung Pang (New York: Wiley, 2005). Lin is an IEEE Fellow, an ACM Fellow, an AAAS Fellow, and an IET Fellow.