A study of performance improvement in EAP

Eun-Chul Cha, Hyoung-Kee Choi

School of Information and Communication
Sungkyunkwan University, Suwon, South Korea
Email : {iris1212, hkchoi}@ece.skku.ac.kr

Abstract. Followed by the popularity of the Internet, a number of access technologies to the Internet have been developed. EAP is an authentication framework. It is designed to provide the authentication functionality in the access network. Because of its flexibility and extensibility EAP poses a global solution for the authentication supported by many access networks. However, EAP has critical weaknesses in the protocol which may, in turn, decrease the EAP performance. Some of the weaknesses are caused by the "lock-step" flow control which only supports a single packet in flight. Considering the weaknesses, we propose the solution for the flow control. Using simulation we prove that our solutions improve the EAP performance.

Keyword: Extensible Authentication Protocol (EAP), Network Access Authentication, Network Security

1 Introduction

The popularity of the Internet makes it possible for people to access the Internet anywhere and anytime. Followed by the popularity of the Internet, a number of access technologies to the Internet have been developed. Those users who want to access the Internet via the access networks such as 802.11 and 802.16 by IEEE must get permissions from a service provider. This transaction happens at the first time of entering networks between a user's terminal and a base station (or access point). The base station authenticates the terminal if the user is a valid identity and vice versa if needed.

A variety of mechanisms are available for an authentication such as Authentication and Key Agreement (AKA), Transport Layer Security (TLS) and Tunneled Transport Layer Security (TTLS). These mechanisms only explain methodologies for the authentication. They cannot be used in the access network without fitting them to collaborate with network protocols. Extensible Authentication Protocol (EAP) is designed for such situation. EAP provides a framework to overlay diverse authentication mechanisms on the access network. EAP defines only the message

"This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2006-C1090-0603-0028)

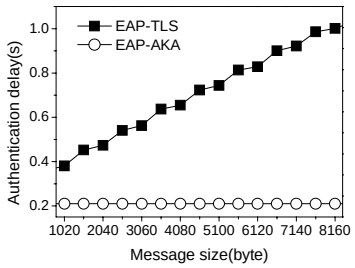

Fig. 1. Authentication delays of EAP-TLS and EAP-AKA with respect to the message size. EAP-AKA has a shorter delay because all messages in EAP-AKA fits to a single EAP message

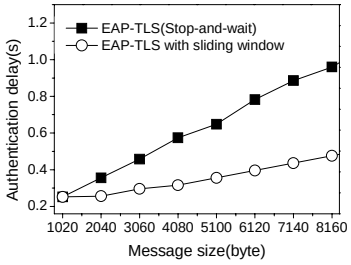

Fig. 2. Comparison of authentication delay between lock-step and sliding window protocols

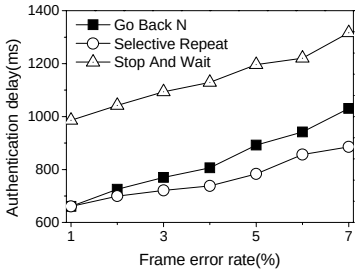

Fig. 3. Comparison of authentication delay between error recovery mechanisms with EAP message of 8160 bytes

format and the message exchange. The actual authentication is done by the authentication mechanism encapsulated in the EAP message.

EAP has critical weaknesses in the protocol which may, in turn, decrease the EAP performance. The weaknesses may be caused by the "lock-step" flow control. This flow control allows an only single packet in flight. If the EAP has a number of messages these messages should be sent in back-to-back instead of in pipeline. The delay to complete the delivery becomes relatively long. We propose to adopt the sliding window protocol in place of the lock-step protocol. This protocol allows a sender to transmit multiple messages in pipeline. Certainly this would lessen the overall delay to complete the authentication. The change of the flow control makes us review the incumbent error recovery scheme, that is "Stop-and-Wait" ARQ. We suggest replacing the "Stop-and-Wait" ARQ by Select Repeat ARQ. In conjunction with the sliding window protocol Selective Repeat ARQ is able to retransmit the only missing message.

This paper is organized as follows. In Section 2, we present the performance issue in EAP and propose the improvement in EAP. We conclude in Section 3.

2 Proposed Improvement in EAP

We present performance issues in EAP and propose possible solutions for those issues in this section. We use a metric, so-called authentication delay, to compare the performance. The authentication delay is the elapse time to complete the authentication using EAP.

EAP makes use of a lock-step protocol for the flow control [1]: i.e., at any given time EAP is allowed to have a single request outstanding. This is okay with EAP as well as some authentication mechanisms running over EAP as the next message is always generated after the previous message returns. However, a few authentication mechanisms like EAP-TLS can have one message larger than one EAP MTU size due to the large certificate. If this is the case this large message needs to be fragmented at

the EAP layer and sent out one fragmentation at one RTT. The Peer may not have the response until the complete EAP message is delivered. In the meantime AUTH after sending the first message would wait for the response. To avoid a possible deadlock in this situation EAP is designed that the Peer sends a null EAP response message to AUTH if the message is fragmented. The delay to complete the delivery of the message equals to the number of fragmentations times RTT. At the worst case TLS can have a certificate which can as large as 16 Mbyte. With the 1020 Byte MTU size it would take 16,000 RTTs [2].

To validate the effect of the lock-step flow control we simulate EAP-TLS varying the size of one EAP message. For the simulation we use the ns-2 simulator. The result of the simulation is shown in Fig. 1. The authentication delay of EAP-AKA does not change with the message size. This is because EAP-AKA suggests not having the EAP message larger than one EAP MTU size [3]. The authentication delay of EAP-TLS increases linearly with the message size.

It is quite inefficient to use the lock-step flow control in EAP-TLS. Instead, we propose to adopt the sliding window flow control. In this scheme the sender may send multiple messages without having to wait for the response of the previous message.

It is necessary to modify the EAP protocol to apply the sliding window flow control. The two buffers in the AUTH and Peer sides are essential. In addition, the Peer must be able to inform AUTH of the available buffer size. This would change the EAP message format. We introduce the option field to allow the Peer to advertise its window size. This advertisement is included in the acknowledgement. The acknowledgment is carried in the EAP null message. The identifier field indicates the message ID being acknowledged. For the backward compatibility issue those who do not understand the new message format would generate the NAK response. Then the lock-step protocol should be used in this case.

For the EAP message smaller than or equal to the EAP MTU size (1020 bytes) no delay difference between the “lock-step” and “sliding window” as shown in Fig. 2. However, as the EAP message grows beyond the EAP MTU size the gap between the two schemes is apparent. For instance the authentication delay in the sliding window is decreased by 53 percents at the 8160 byte message.

Like many other protocols the error recovery in EAP works in conjunction with the flow control. Since the sliding window protocol is proposed for the flow control it is necessary to find an ARQ mechanism to work closely with the sliding window protocol. We examine the two mechanisms, Go-BACK-N ARQ and Selective Repeat ARQ. In Go-Back-N if the Nth message is detected to err, the sender must go back to the Nth message and transmit following messages again from the Nth message. The (N+1)th message may be sent without errors at the first attempt. However, Go-Back-N ends up with retransmit ting the (N+1)th message. At the same situation Selective Repeat ARQ allows to retransmit only the Nth message without having to retransmit the (N+1)th message.

To evaluate the effect of the two ARQ mechanisms we measure the authentication delay of EAP varying the frame error rate (FER) on the IEEE 802.16 link. EAP messages are delivered over the PKM protocol in 802.16. PKM provides its own ARQ scheme. However, the frame drop is still possible. After the collision the mobile station attempts to retransmit the frame within the limited number of times. If the collision continues beyond the certain limit the MS gives up transmitting that frame.

Fig. 3 shows the authentication delay against the FER. The two proposed schemes decrease the authentication delay significantly. At the three percent FER, Selective Repeat ARQ and Go-Back-N lessen the delay by 35.1 and 29.6 percents, respectively. The EAP protocol needs to be modified to adopt the two proposed schemes. Because most of the modifications for the flow control are shared with the error recovery we explain the additional changes unique to the error recovery. One major concern in deploying ARQ is that the messages can be out of sequence due to the selective retransmission. EAP is designed to drop out-of-sequence messages implicitly [1]. The great care must be taken to implement ARQ on EAP. The modification for Go-Back-N is minimal because the message, in this scheme, cannot be out of sequence at the receiver side. The problem is only valid for Selective Repeat ARQ. We propose to assign a sequence number to all messages to rearrange the order at the receiver side. To do this we take advantage of the identifier field in EAP for the sequence number. For the temporal storage for out-of-sequence messages we use the receiver buffer made it for the flow control.

3 Conclusion

EAP is a lock-step protocol which only supports a single message in flight. If the message is large the fragmentation in EAP is inevitable. The delay to complete the delivery of the message equal to the number of fragmentations times RTT. We proposed the sliding window protocol for the flow control in EAP. The simulation result indicates that the sliding window protocol decreases the authentication delay of the two fragmented message by 53 percent.

Reference

1. B. Aboba et al., “Extensible Authentication Protocol(EAP),” RFC 3748, Jun. 2004.
2. B. Aboba and D. Simon, “PPP EAP TLS Authentication Protocol,” RFC 2716, Oct. 1999.
3. J. Arkko and H. Haverinen, “Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement(EAP-AKA),” RFC 4187, Jan. 2006.