

Internet Engineering Task Force  
Internet Draft  
March, 2003  
Expires: September, 2003

Tsunemasa Hayashi, NTT  
Daisuke Andou, NTT  
Haixiang He, Nortel Networks  
Wassim Tawbi, Nortel Networks  
Teruki Niki, Matsushita Electric Industrial

IGMP for user Authentication Protocol (IGAP)  
<[draft-hayashi-igap-01.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) except that the right to produce derivative works is not granted.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

Abstract

This memo documents the IGMP for user Authentication Protocol (IGAP), a protocol developed by NTT, Nortel Networks and Panasonic. IGAP extends the existing IGMPv2 protocol to add user authentication functionality, accounting functionality and status notification functionality. IGAP is only used in IPv4 environment. In a controlled or managed multicast environment, IGAP can be used to replace IGMP for group membership communication between hosts and their associated first hop routers. The user authentication information in IGAP can enable a provider to control the distribution of the multicast traffic as well as collecting real time user accounting information in an environment where the last hop access networks are not shared.

**1. Introduction**

IP multicast provides an efficient mechanism for delivering packets to multiple destinations. Unfortunately, IP multicast services, especially commercial IP multicast services, are not widely deployed. One of the important reasons that discourage the deployment

is related to the current IP multicast model.

The current IP multicast model provides by nature a non-secure non-controlled way for end systems attached to a network to access multicast traffic. Lack of access control in this model makes it difficult for a service provider to generate enough revenue to sustain multicast services such as IP multicast based Internet TV.

A provider can enforce such access control through static configuration on the last hop network devices including Ethernet switches or routers. However, the rules to control the access to multicast data may change dynamically or the rules may be very specific such as user-based rules instead of end system based rules that a network device is not always able to enforce. This leads to the need for a comprehensive way to authenticate and authorize end systems before they are granted access to some multicast groups.

The IGMP for user Authentication Protocol (IGAP) is designed to facilitate the last hop network devices to enforce the dynamic multicast receiver access control in a non-shared access networks environment. IGAP extends the existing IGMPv2 [[IGMPv2](#)] protocol to add authentication functionality via permitting known authentication mechanisms such as password mechanism and challenge-response mechanism to be incorporated into IGMP protocol sequences. IGAP is used only in IPv4 environment. IGAP enables an IP multicast service provider to authenticate and authorize a host's requests to join a specific multicast group based on its user's authentication information and then to control the user's access to the multicast traffic accordingly.

IGAP uses a user-based authentication model versus IP or MAC address based authentication model. The benefits of a user-based model are well known. It offers operational simplicity and flexibility, in particular with respect to adds, moves, and changes.

Another issue that discourages the wide deployment of IP multicast services is the lack of multicast network management functions especially an effective multicast accounting function. Effective user-based accounting information is critical in two aspects. On one hand network providers who provide commercial multicast services need to accurately identify the users and collect their usage information to generate correct billing information. On the other hand some content providers need to learn the content usage information. For example, in IP multicast based Internet TV services, network providers need to know which TV program and how long a user watches so that they can charge the user differently based on the values of the TV programs. In such services, content providers, TV programs owners, need to know how many viewers for a TV program and how long they watch the TV program so they can generate appropriate advertisement revenue.

IGAP combines the user information including user ID with the multicast group addresses that reflect the different contents.

Authenticated and authorized group join requests enable providers to effectively collect the user usage information for different content.

IGAP not only encourages the wide deployment of new commercial IP multicast services, but also can be used in non-commercial environments such as enterprises. For example, IGAP can be used for closed video broadcasting. IGAP provides a mechanism to allow the access to the video broadcasting, only if the user is an authenticated user who is allowed to join the video broadcasting.

IGAP is designed to add authentication capability to IGMP transactions controlling multicast group membership. The transactions flow between an IGAP host client and an IGAP router. The IGAP router is assumed to be 1 hop from the IGAP host, such that the host does not have a route that bypasses the IGAP router. An IGAP host **MUST** authenticate itself to an IGAP router in order to join a multicast group.

## 2. IGAP Message Format

Similar to IGMP, IGAP messages are encapsulated in IP datagrams, with an IP protocol number of 2. All IGAP messages described in this document are sent with IP TTL 1, and contain the IP Router Alert option [[RFC 2113](#)] in their IP header.

IGAP adopts and extends the IGMPv2 packet format. All IGAP messages of concern have the following packet format:

0										1										2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	(bit)								
Type										Max Resp Time										Checksum																				
Group Address																																								
Version										Subtype										Reserved-1										Challenge ID										
Account Size										Message Size										Reserved-2																				
User Account (16 bytes)																																								

. Message (64 bytes)  
.  
|

.  
.  
|

Hayashi, Andou, He, Tawbi, Niki

[Page 3]

+--+

## **[2.1](#) Type**

There are three types of IGAP messages of concern.

0x40 = IGAP Membership Report (IGAP Join)

0x41 = IGAP Membership Query (IGAP Query)

0x42 = IGAP Leave Group (IGAP Leave)

Unrecognized message types should be silently ignored.

## **[2.2](#) Max Response Time**

The meaning and the usage of Max Response Time are the same as those of the IGMP messages as described in [RFC 2236](#) [[IGMPv2](#)].

## **[2.3](#) Checksum**

Checksum covers the IGAP message (the entire IPv4 payload). The algorithm is the same as described in [RFC2236](#) [[IGMPv2](#)].

## **[2.4](#) Group Address**

In a Basic Query message described in [section 2.6](#), the group address field is set zero. In both Authentication Message and Accounting Message described in [section 2.6](#), the group address field holds the IP multicast address of an IGAP Join. In a Membership Report or Leave Group message, the group address field holds the IP multicast group address of interest or the group being left.

## **[2.5](#) Version**

This field indicates the version of IGAP. It is set to 0x10 to indicate the IGAP version 1.

## **[2.6](#) Subtype**

This field indicates the subtype of message transferred within the IGAP packet. Usage of this field is described later.

The following 3 Subtypes are only used in IGAP join (Type 0x40).

Hayashi, Andou, He, Tawbi, Niki

[Page 4]



- 0x02 : Password Mechanism Join  
(Password-Join)
- 0x03 : Challenge-Response Mechanism Join Challenge Request  
(Challenge-Request-Join)
- 0x04 : Challenge-Response Mechanism Join Response  
(Challenge-Response-Join)

The following 4 Subtypes are only used in IGAP Query (Type 0x41).

- 0x21 : Basic Query
- 0x23 : Challenge-Response Mechanism Challenge  
(Challenge)
- 0x24 : Authentication Message
- 0x25 : Accounting Message

The following 4 Subtypes are used in IGAP Leave (Type 0x42).

- 0x41 : Basic Leave
- 0x42 : Password Mechanism Leave  
(Password-Leave)
- 0x43 : Challenge-Response Mechanism Leave Challenge Request  
(Challenge-Request-Leave)
- 0x44 : Challenge-Response Mechanism Leave Response  
(Challenge-Response-Leave)

## **[2.7](#) Reserved-1**

This field should be set to 0x00. It is ignored when received.

## **[2.8](#) Challenge ID**

This field is meaningful only when Challenge-Response authentication mechanism is used. The value is set according to the Challenge-Response protocol. If this field is not used, it is set to the default value of 0x00.

## **[2.9](#) Account Size**

This field indicates the valid length in units of bytes of the User Account field described in [section 2.12](#). The value must be less than or equal to 16. If this field is not used, it is set to the default value of 0x00.

## **[2.10](#) Message Size**

This field indicates the valid length in units of bytes of the Message field described in [section 2.13](#). The value must be less than

or equal to 64. If this field is not used, it is set to the default value of 0x00.

### **[2.11](#) Reserved-2**

This field should be set to 0x00. It is ignored when received.

### **[2.12](#) User Account**

This field contains the user account information. The size of this field is 16 bytes. The length of the valid information is decided by the Account Size field described in [section 2.9](#). If the user account occupies less than 16 bytes, the field is padded with 0x00.

### **[2.13](#) Message**

This field contains certain information such as password. Different IGAP messages contain different Message information. The size of this field is 64 bytes. The length of the valid information is decided by the Message Size field described in [section 2.10](#). If message information occupies less than 64 bytes, the field is padded with 0x00.

## **[3.](#) Protocol Description**

IGAP is used for controlled or managed multicast services. A user must use IGAP to access such multicast services. IGMP is not needed when IGAP is used. An IGAP router will ignore all IGMP messages.

Different from IGMPv2, IGAP tracks individual host's group membership information. This feature allows an IGAP router to implement fast leave feature. In another word, IGAP does NOT implement Group-Specific Query feature that IGMPv2 has. When an IGAP router receives an IGAP Leave message, it will not send Group-Specific Query. Instead it will just delete corresponding state information. To facilitate tracking individual host's group membership, Host Suppression feature is NOT allowed in IGAP.

IGAP specifies different behaviors for IGAP hosts and for IGAP routers. If an IGAP router attempts to join some multicast groups, it can perform both parts of the protocol.

### **[3.1](#) User Authentication Mechanisms**

Currently IGAP supports two user authentication mechanisms for Join

operation: simple and basic password authentication mechanism [[PAP](#)],  
and more advanced challenge-response authentication mechanism

[[CHAP](#)]. These mechanisms are not used at the same time. Only one mechanism is configured to be used in a network. An IGAP implementation **MUST** support password authentication mechanism. Challenge-response authentication mechanism is optional.

IGAP is intended for use with standard AAA servers such as RADIUS [[RADIUS](#)] servers that with necessary extensions can be used to achieve the authentication, authorization and accounting functions described in this document. However, IGAP is not limited for use with only standard AAA servers. It can be used with any back-end Authentication, Authorization, and Accounting functions or mechanisms. These functions or mechanisms can be located in different servers, within one server, or even within the IGAP routers. In this document, we use AAA servers as an example for these functions or mechanisms.

### **[3.2](#) IGAP Host side Protocol Description**

This section describes the IGAP host behavior. Based on the configured authentication mechanism, an IGAP host behaves differently.

#### **[3.2.1](#) Password Authentication Mechanism**

When an IGAP host joins a multicast group, it should immediately transmit an unsolicited IGAP Membership Report that has a Subtype field of 0x02 (Password Mechanism Join) to the corresponding group. The User Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account. The Message field is filled with the user password while the Message Size field is set to the length of the password.

When a host receives an IGAP Query, it sets delay timers as described in [RFC2236](#) [[IGMPv2](#)]. If a timer for the group is already running, it is reset to the random value only if the requested Max Response Time is less than the remaining value of the running timer. When a group's timer expires, the host sends a Membership Report that has a Subtype field of 0x02. In this message, the User Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account.

When an IGAP host leaves from a multicast group, it sends an IGAP Leave Group message to the all-routers multicast group (224.0.0.2). Normally an IGAP host sends a Leave message that has a Subtype field of 0x41 (Basic Leave). In Basic Leave, the User Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account. In scenarios

where Leave message authentication is required, an IGAP host can send a Leave message that has a Subtype field of 0x42 (Password Mechanism

Leave). In Password Mechanism Leave, the User Account and Account Size fields are set to the values as in Basic Leave. The Message field is filled with the user password while the Message Size field is set to the length of the password. An IGAP implementation MUST support Basic Leave. Password Mechanism Leave is optional.

### **3.2.2 Challenge-Response Authentication Mechanism**

When an IGAP host joins a multicast group, it sends a Challenge-Request-Join that has a Subtype field of 0x03 (Challenge-Response Mechanism Join Challenge Request) to the corresponding group. The user Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account. The message field is not used.

When the IGAP host receives a Challenge that has a Subtype of 0x23 (Challenge-Response Mechanism Challenge) as a response to the Challenge-Request-Join, the IGAP host sends a Challenge-Response-Join that has a Subtype of 0x04 (Challenge-Response Mechanism Join Response). The Challenge ID field is set to the same value of Challenge ID on the Challenge packet. The user Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account. The Message field is set the results of MD5 calculation. The Message Size field is set to 0x10.

When a host receives an IGAP Query, it follows the behavior described above to set the delay timer. When a group's timer expires, the host sends a Membership Report that has a Subtype field of 0x03. In this message, the User Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account.

When an IGAP host leaves from a multicast group, it sends an IGAP Leave Group message to the all-routers multicast group (224.0.0.2). Normally an IGAP host sends a Basic Leave message as described above. In scenarios where Leave message authentication is required, an IGAP host can send a Leave message that has a Subtype field of 0x43 (Challenge-Response Mechanism Leave Challenge Request). The User Account field is filled with the user account (user ID) while the Account Size field is set to the length of the user account. The other fields are not used. When the IGAP host receives a Challenge that has a Subtype of 0x23 (Challenge-Response Mechanism Challenge) as a response to the Challenge-Response Leave, it sends a Leave message that has a Subtype field of 0x44 (Challenge-Response Mechanism Leave Response). The User Account field and Account Size field are the same. The Message field is set to the

results of MD5 calculation. The Message Size field is set to 0x10.  
An IGAP implementation MUST support Basic Leave. Challenge-Response  
Authentication Mechanism Leave is optional.



### **3.3 IGAP Router side Protocol Description**

IGAP routers use IGAP to learn which groups have members on each of their interfaces. They can be physical interfaces or virtual interfaces such as VLANs. Same as IGMP, an IGAP router keeps a list of multicast group memberships for each attached network, and a timer for each membership. Each group membership state has the conceptual following format:

(group address, user-id, host IP, timer)

IGAP routers periodically [Query Interval] send an IGAP Membership Query on each attached network to solicit membership information. On startup, a router SHOULD send [Startup Query Count] IGAP Membership Queries spaced closely together [Startup Query Interval] in order to quickly and reliably determine membership information. [Query Interval], [Startup Query Count] and [Startup Query Interval] are same as [RFC 2236](#) [IGMPv2].

An IGAP Membership Query is addressed to the all-systems multicast group (224.0.0.1), has a Group Address field of 0, has a Max Response Time of [Query Response Interval], and has a IGAP Type field of 0x21 (Basic Query). Other fields are not used. [Query Response Interval] is same as [RFC 2236](#) [IGMPv2].

When an IGAP router receives an IGAP Membership Report or an IGAP Group Leave, it takes different actions based on the configured authentication mechanism.

#### **3.3.1 Password Authentication Mechanism**

When an IGAP router receives a Password Mechanism Join (an IGAP join that has a Subtype field of 0x02), if the router already has the corresponding group membership state, it refreshes the associate timer.

If the router does not have the group membership state, it forwards the user's group join request information as well as its user authentication information including its user account and password to the back-end AAA server. Based on the AAA server's results of authentication and authorization processes, the IGAP router grants or denies the user's access request. When the IGAP router grants the request, it adds the group being reported to the list of multicast group memberships on the interface on which it received the Report and sets the timer for the membership to the [User Membership Interval].

When an IGAP router receives an IGAP Leave message for a group that

has group members on the reception interface, it deletes the corresponding group membership state.

If Leave message authentication is required, an IGAP Leave (Password-Leave) MUST have a Subtype field of 0x42, and includes a user authentication information which is same to a user

authentication on Password-Join, and the router forwards the user's group leave information as well as the user authentication information to the back-end AAA server. If the group leave request is authenticated and authorized, the router deletes the corresponding group membership state. Otherwise, the leave request is ignored.

### **3.3.2 Challenge-Response Authentication Mechanism**

When an IGAP router receives a Challenge-Response Mechanism Join Challenge Request Mechanism Join (a Challenge-Request-Join that has a Subtype field of 0x03), the router tries to establish Challenge-Response communication for a Join process, then the router sends a Challenge-Response Mechanism Challenge (a Challenge that has a Type field of 0x41, a Subtype field of 0x23, a Challenge ID field of an ID [[CHAP](#)], a User Account set to the same user ID in the Challenge-Response-Join, and a Message set to a Challenge value [[CHAP](#)]).

When the IGAP router receives a Challenge-Response Mechanism Join Response (a Challenge-Response-Join that has a Subtype field of 0x04), if the router already has the corresponding group membership state, it refreshes the associate timer.

If the router does not have the group membership state, it forwards the user's group join request information as well as its user authentication information including its user account and password to the back-end AAA server. Based on the AAA server's results of authentication and authorization processes, the IGAP router grants or denies the user's access request. When the IGAP router grants the request, it adds the group being reported to the list of multicast group memberships on the interface on which it received the Report and sets the timer for the membership to the [User Membership Interval].

When an IGAP router receives an IGAP Leave message for a group that has group members on the reception interface, it deletes the corresponding group membership state.

If Leave message authentication is required, a host oriented Challenge-Response communication is establish between a host and the IGAP router. When a IGAP router receives an Challenge-Response Mechanism Leave (Challenge-Request-LLeave that has a Subtype field of 0x43), the router sends a Challenge-Response Mechanism Challenge (a Challenge that has a Type field of 0x41, a Subtype field of 0x23, a

Challenge ID field of an ID [[CHAP](#)], a User Account set to the same user ID in the Challenge-Request-Leave, and a Message set to a

Challenge value [[CHAP](#)]).

When the IGAP router receives a Challenge-Response Mechanism Leave Response (a Challenge-Response-Leave that has a Subtype field of 0x44, the User Account field and Account Size field are the same. The Message field is set to the results of MD5 calculation. The Message Size field is set to 0x10), and the router forwards the user's group leave information as well as the user authentication information to the back-end AAA server. If the group leave request is authenticated and authorized, the router deletes the corresponding group membership state. Otherwise, the leave request is ignored.

An IGAP implementation **MUST** support Basic Leave. Challenge-Response Authentication Mechanism Leave is optional.

### **[3.4](#) Status Notifications**

In controlled or managed multicast environments, it is very important to notify a user of its service statuses. IGAP supports the following status notifications.

#### **[3.4.1](#) Authentication Result Notification**

When an IGAP router receives the authentication result from the back-end AAA server, it notifies the user of the result by unicasting an Authentication message to the host.

The Authentication message has a Type field of 0x41 (IGAP Query) and a Subtype field of 0x24. The Group Address field contains the corresponding group address for authentication. The Max Resp Time field is not used and is ignored by IGAP hosts. It can be set to any value or set to the default value 0x64. The User Account contains the user account (user ID) for authentication and the Account Size field is set the length of the user account.

The Message Size field is set to 0x01. The Message field has the following values:

- 0x11: Authentication success.
- 0x21: Authentication failure.

An IGAP implementation **MUST** support the above mandatory values. It supports the any other vendor specific values. Appropriate value is chosen to reflect the result from the AAA server as well as other vendor specific processes. The process adopted by the IGAP hosts upon receiving this packet type is up to implementation. However, it must not affect other IGAP process.



### **3.4.2 Accounting Status Notification**

An IGAP router informs the accounting server to start accounting when it starts forwarding related multicast traffic into the host's network. When the IGAP host leaves the multicast group (either via silent departure or an explicit leave), the router informs the accounting server to stop accounting. Once it receives the response from the accounting server, it notifies the IGAP host by unicasting an Accounting message.

The Accounting message has a Type field of 0x41 (IGAP Query) and a Subtype field of 0x25. The Group Address field, the Max Resp Time field, the User Account field, and the Account Size field are the same as those in the Authentication message described in [section 3.4.1](#).

The Message Size field is set to 0x01. The Message field has the following values:

0x11: Accounting start  
0x12: Accounting stop

An IGAP implementation MUST support the above mandatory values. It supports the any other vendor specific values. The process adopted by the IGAP host upon receiving this packet type is up to implementation. However, it must not affect other IGAP process.

### **3.5 Validity Period**

For each group membership state, an IGAP router MAY maintain another timer: Validity Period timer. This timer indicates the valid period of an accounting to a group membership. When the timer is expired, an IGAP router needs to re-authenticate the group membership. The value of the "Validity Period" can be statically configured or dynamically set based on the results from the AAA server.

When "Validity Period" is enforced, an IGAP router checks this timer when it receives an IGAP Join. If the timer does not expire, the IGAP router does not ask the AAA server a user authentication by a IGAP Join response. If the timer expires, it follows the procedures for initial authentication described above to re-authenticate the join request. During the re-authentication period, an IGAP router continues forwarding the multicast traffic and does not stop accounting. If the re-authentication succeeds, an IGAP router resets the group timer and the Validity Period timer. If the re-authentication fails, an IGAP router stops accounting and deletes the group membership state.

#### **4. Security Considerations**

Hayashi, Andou, He, Tawbi, Niki

[Page 12]



IGAP is based around an asymmetrical trust model in which the IGAP router does not trust the IGAP host, but the IGAP host trusts the IGAP router therefore may not be suitable for use in isolation where mutual authentication is required.

IGAP supports password and challenge-response authentication mechanisms and inherits the security concerns of each. For multicast content encryption related technology, please refer to other IETF work. IGAP does not obstruct snooping of multicast traffic by unauthorized host that have access to media shared with multicast traffic.

Some of the security issues discussed in IGMPv2 document also apply here. Please refer to IGMPv2 document [[IGMPv2](#)] for details.

## 5. IANA Considerations

This document introduces the following new Types of IGMP that require allocation by IANA:

0x40:	IGAP Membership Report (IGAP Join)
0x41:	IGAP Membership Query (IGAP Query)
0x42:	IGAP Leave Group (IGAP Leave)

## Acknowledgments

Portions of this document are copied from [RFC 2236](#) [[IGMPv2](#)]. The authors would like to thank Daphne Tong, Dave Allen, Abbie Barbir, Ghassem Koleyni, Paul Knight, Kaori Izutsu, Akihiro Tanabe, Takashi Shimizu, and Atsushi Takahara for their kindness, patience, and time to review the document and to provide their valuable suggestions.

## Normative References

### [IGMPv2]

W. Fenner, "Internet Group Management Protocol, Version 2", [RFC 2236](#), Xerox PARC, November 1997.

### [IPRA]

D. Katz, "IP Router Alert Option", [RFC 2113](#), Cisco Systems, February 1997.

### [MD5]

R. Rivest, S. Dusse, "The MD5 Message-Digest Algorithm", [RFC 1321](#), April 1992.

[RADIUS]

C. Rigney, S. Willens, A. Rubens, W. Simpson, "Remote Authentication

Hayashi, Andou, He, Tawbi, Niki

[Page 13]

Dial In User Service (RADIUS)", [RFC 2865](#), June 2000.

[PAP]

B. Lloyd and W. Simpson, "PPP Authentication Protocols", [RFC1334](#), October 1992.

[CHAP]

W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)", [RFC 1994](#), August 1996.

Author's Addresses

Tsunemasa Hayashi  
NTT Network Innovation Laboratories  
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan  
Phone : +81 468 59 8790  
Email : hayashi@exa.onlab.ntt.co.jp

Daisuke Andou  
NTT Access Network Service Systems Laboratories  
1-6 Nakase Mihiama-ku, Chiba-shi, Chiba, 261-0023 Japan  
Phone : +81 43 211 2115  
Email : dandou@ansl.ntt.co.jp

Haixiang He  
Nortel Networks  
600 Technology Park Drive  
Billerica, MA 01801, USA  
Phone : 1 978 288 7482  
Email : haixiang@nortelnetworks.com

Wassim Tawbi  
Nortel Networks  
4655 Great America Parkway  
Santa Clara, CA 95054, USA  
Email : wtawbi@nortelnetworks.com

Teruki Niki  
Matsushita Electric Industrial Co.,Ltd  
Multimedia Systems Research-Laboratory  
4-5-15 Higashi-Shinagawa Shinagawa-ku, Tokyo, 140-8632 Japan  
Phone : +81 3 5460 2741  
Email : niki.teruki@jp.panasonic.com

## Appendix 1. IGAP State Machines on Password authentication

Hayashi, Andou, He, Tawbi, Niki

[Page 14]

This Section describes an example of IGAP State Machines. The example of FSM on Password authentication mechanism with a infinity value of Validity-Period is shown about each pattern as follows.

Authentication and Accounting packets are operated between IGAP routers and AAA servers. In this example, Basic-Leave is used, although there is another alternative for Leave operation (Password Mechanism Leave). The choice is depends on the strategy of a network service provider.

Here, we define a vendor specific Message "Error-Message" in Authentication Message. The Message field has 0x25.

#### **A.1.1. FSM for Host**

```
PH1[Non Member]:
  if join group{
    send Password-Join;
    start Authentication-Timer;
    transition PC2;
  }

PH2[Waiting Authentication Message Member]:
  if Authentication-Message(Reject) received
    or Error-Message(Response time out) received
    or Authenticated-Timer expired{
    stop Authenticated-Timer;
    transition PC1;
  }
  else if Authentication-Message(Success) received{
    stop Authenticated-Timer;
    transition PC3;
  }

PH3[Idle Member]:
  if Basic-Query received{
    start Delaying-Timer;
    transition PC4;
  }
  else if leave group{
    send Basic-Leave;
    transition PC5;
  }

PH4[Delaying Member]:
  if leave group{
    send Basic-Leave;
    stop Delaying-Timer;
```

```
    start Accounted-Timer;  
    transition PC5;  
}
```

```
    else(Delaying-Timer expired){
        send Password-Join;
        stop Delaying-Timer;
        transition PC2;
    }

PH5[Waiting Accounting Message Member]:
    if Accounting-Message(Stop) received{
        stop Accounted-Timer;
        transition PC1;
    }
    else if Basic-Query received{
        send Basic-Leave;
        continue(no transition);
    }
    else(Accounted-Timer expired){
        send Basic-Leave;
        restart Accounted-Timer;
        continue(no transition);
    }
}
```

#### A1.2. FSM for IGAP router

```
PR1[No Transfer Present]:
    if Password-Join received{
        send Authentication Request;
        start Authentication-Timer;
        transition PR2;
    }
    else if Basic-Leave received{
        send Accounting-Request(Stop);
        transition PR5;
    }
}

PR2[Waiting Authentication-Response]:
    if Access-Reject received{
        send Authentication-Message(Reject);
        stop Authentication-Timer;
        transition PR1;
    }
    else if Access-Accept received{
        send Accounting-Request(Start);
        send Authentication-Message(Success);
        stop Authentication-Timer;
        start Accounting-Timer;
        transition PR3;
    }
}
```

```
else(Authentication-Timer expired){  
    send Error-Message(Response time out);  
    stop Authentication-Timer;
```



```
        start User-Membership-Interval-Timer;
        transition PR4;
    }

PR3[Waiting Accounting-Response(Start)]:
    if Accounting-Response received{
        send Accounting-Message(Start);
        stop Accounting-Timer;
        start User-Membership-Interval-Timer;
        transition PR4;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        start User-Membership-Interval-Timer;
        transition PR4;
    }

PR4[Transfer Present]:
    if Password-Join received{
        restart User-Membership-Interval-Timer;
        continue(no transition);
    }
    else if Basic-Leave received{
        send Accounting-Request(Stop);
        stop User-Membership-Interval-Timer;
        start Accounting-Timer;
        transition PR5;
    }
    else(User-Membership-Interval-Timer expired){
        send Accounting-Request(Stop);
        start Accounting-Timer;
        transition PR5;
    }

PR5[Waiting Accounting-Response(Stop) for Leave]:
    if Accounting-Response received{
        send Accounting-Message(stop);
        stop Accounting-Timer;
        transition PR1;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        transition PR1;
    }
}
```

This Section describes an example of IGAP State Machines. The example of FSM on Challenge-Response with a finite value of Validity-Period

is shown about each pattern as follows. Authentication and Accounting packets are operated between IGAP routers and AAA servers. In this example, Basic-Leave is used, although there is another alternative for Leave operation (Challenge-Response Mechanism Leave). The choice is depends on the strategy of a network service provider.

Here, we define a vendor specific Message "Error-Message" in Authentication Message. The Message field has 0x25.

#### A2.1. FSM for Host

```
CH1[Non Member]:
  if join group{
    send Challenge-Request-Join;
    start Challenge-Timer;
    transition CC2;
  }

CH2[Waiting Challenge Member]:
  if Challenge received{
    send Challenge-Response-Join;
    stop Challenge-Timer;
    start Authenticated-Timer;
    transition CC3;
  }
  else(Challenge-Timer expired){
    stop Challenge-Timer;
    transition CC1;
  }

CH3[Waiting Authentication Message Member]:
  if Authentication-Message(Reject) received
    or Error-Message(Response time out) received
    or Authenticated-Timer expired{
    stop Authenticated-Timer;
    transition CC1;
  }
  else if Authentication-Message(Success) received
    stop Authenticated-Timer;
    transition CC4;
  }

CH4[Idle Member]:
  if Basic-Query received{
    start Delaying-Timer;
    transition CC5;
  }
```

```
else if leave group{  
    send Basic-Leave;  
    start Accounted-Timer;
```

```
        transition CC6;
    }

CH5[Delaying Member]:
    if leave group{
        send Basic-Leave;
        stop Delaying-Timer;
        start Accounted-Timer;
        transition CC6;
    }
    else(Delaying-Timer expired){
        send Challenge-Request-Join;
        stop Delaying-Timer;
        start Challenge-Timer;
        transition CC2;
    }

CH6[Waiting Accounting Message Member]:
    if Accounting-Message(Stop) received{
        stop Accounted-Timer;
        transition CC1;
    }
    else if Basic-Query received{
        send Basic-Leave;
        continue(no transition);
    }
    else(Accounted-Timer expired){
        send Basic-Leave;
        restart Accounted-Timer;
        continue(no transition);
    }
}
```

#### A2.2. FSM for IGAP router

```
CR1[No Transfer Present]:
    if Challenge-Request-Join received{
        send Challenge;
        start Response-Timer;
        transition CR2;
    }
    else if Basic-Leave received{
        send Accounting-Request(Stop);
        transition CR7;
    }

CR2[Waiting Challenge-Response]:
    if Challenge-Response-Join received{
```

```
send Authentication Request;  
stop Response-Timer;  
start Authentication-Timer;
```

```
        transition CR3;
    }
    else(Response-Timer expired){
        stop Response-Timer;
        transition CR1;
    }

CR3[Waiting Authentication-Response]:
    if Access-Reject received{
        send Authentication-Message(Reject);
        stop Authentication-Timer;
        transition CR1;
    }
    else if Access-Accept received{
        send Accounting-Request(Start);
        send Authentication-Message(Success);
        stop Authentication-Timer;
        start Accounting-Timer;
        transition CR4;
    }
    else(Authentication-Timer expired){
        send Error-Message(Response time out);
        stop Authentication-Timer;
        start User-Membership-Interval-Timer;
        start Validity-Timer;
        transition CR5;
    }
}

CR4[Waiting Accounting-Response(Start)]:
    if Accounting-Response received{
        send Accounting-Message(Start);
        stop Accounting-Timer;
        start User-Membership-Interval-Timer;
        start Validity-Timer;
        transition CR5;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        start User-Membership-Interval-Timer;
        start Validity-Timer;
        transition CR5;
    }
}

CR5[Transfer Present]:
    if Challenge-Request-Join received{
        if Validity-Timer < Validity-Period{
            restart User-Membership-Interval-Timer;
```

```
        continue(no transition);  
    }  
    else(Validity-Timer expired){
```



```
        send Accounting-Request(Stop);
        stop Validity-Timer;
        stop User-Membership-Interval-Timer;
        start Accounting-Timer;
        transition CR6;
    }
}
else if Basic-Leave received{
    send Accounting-Request(Stop);
    stop User-Membership-Interval-Timer;
    stop Validity-Timer;
    start Accounting-Timer;
    transition CR7;
}
else(User-Membership-Interval-Timer expired){
    send Accounting-Request(Stop);
    stop Validity-Timer;
    start Accounting-Timer;
    transition CR7;
}

CR6[Waiting Accounting-Response(Stop)]:
    if Accounting-Response received{
        send Accounting-Message(Stop);
        send Challenge;
        stop Accounting-Timer;
        start Response-Timer;
        transition CR2;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        start Validity-Timer;
        transition CR5;
    }
}

CR7[Waiting Accounting-Response(Stop) for Leave]:
    if Accounting-Response received{
        send Accounting-Message(stop);
        stop Accounting-Timer;
        transition CR1;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        transition CR1;
    }
}
```

### Appendix 3. IGAP State Machines of Query Process

Hayashi, Andou, He, Tawbi, Niki

[Page 21]

This Section describes an example of IGAP State Machines on Query Process.

```
QR1[Initial]:
  start IGAP{
    send Basic-Query;
    start Startup-Query-Interval-Timer;
    start Startup-Query-Counter;
    transition QR2;
  }

QR2[Startup]
  if Startup-Query-Interval-Timer expired{
    if Startup-Query-Counter < Startup-Query-Count{
      send Basic-Query;
      restart Startup-Query-Interval-Timer;
      continue(no transition)
    }
    else{
      send Basic-Query;
      stop Startup-Query-Counter;
      start Query-Interval-Timer;
      transition QR3;
    }
  }

QR3[Affirmed Connection]:
  if Query-Interval-Timer expired{
    send Basic-Query;
    restart Query-Interval-Timer;
    continue(no transition);
  }
```

#### Appendix 4. List of Timers, Counters

This section describes the parameters set in IGAP router and Host when supporting IGAP processes.

##### A4.1. Robustness Variable

It is the same meaning as IGMPv2.

##### A4.2. Timers for Host

###### A4.2.1. Challenge-Timer

It controls waiting time from sending Join message to receiving

Challenge Message.

Hayashi, Andou, He, Tawbi, Niki

[Page 22]

#### A4.2.2. Authenticated-Timer

It controls waiting time from sending Response Message to receiving Authentication Message (accept or reject) from IGAP router.

#### A4.2.3. Accounted-Timer

It controls waiting time from sending Response Message to receiving Accounting Message (start or stop) from IGAP router.

#### A4.2.4. Delaying-Timer

It controls waiting time from receiving Query to sending Join Message to IGAP router. It is calculated from Max Resp Time.

### A4.3. Timers and Counters for IGAP router

#### A4.3.1. Response-Timer

It controls waiting time from sending Challenge Message to receiving Response Message.

#### A4.3.2. Authentication-Timer

It controls waiting time from sending Authentication request to receiving Authentication Response.

#### A4.3.3. Accounting-Timer

It controls waiting time from sending Accounting request to receiving Accounting Response.

#### A4.3.4. Validity-Timer

This is an integer multiple of Basic-Query Interval in units of second, and used by IGAP router to determine whether user authentication is necessary or not.

#### A4.3.5. Query-Interval-Timer

It is the same meaning as IGMPv2. The Query Interval is the interval between Basic Queries.

#### A4.3.6. Query-Response-Interval-Timer

It is the same meaning as IGMPv2. The Max Response Time inserted into the periodic Basic Queries.

#### A4.3.7. User-Membership-Interval-Timer



The User Membership Interval is the amount of time that must pass before a IGAP router decides there are no more users of a group on a network. This value MUST be ((the Robustness Variable) times (the Query Interval)) plus (one Query Response Interval).

#### A4.3.8. Startup-Query-Interval-Timer

It is the same meaning as IGMPv2.

The Startup Query Interval is the interval between General Queries sent by a Querier on startup.

#### A4.3.9. Startup-Query-Counter

It is the same meaning as IGMPv2.

The Startup Query Count is the number of Queries sent out on startup, separated by the Startup Query Interval.

