```
Internet Engineering Task Force                 Tsunemasa Hayashi, NTT
Internet Draft                                      Daisuke Andou, NTT
October, 2002                             Haixiang He, Nortel Networks
Expires: April, 2003                     Wassim Tawbi, Nortel Networks
                             Teruki Niki, Matsushita Electric Industrial
```

# IGMP for user Authentication Protocol (IGAP)

<draft-hayashi-igap-00.txt>

## Status of this Memo

## Abstract

IP Multicast applications are becoming more common. Two key concerns raised by the providers of such applications are the lack of control on what users can get multicast traffic and a method for tracking user usage (such as how long these users are joined to a multicast group).

This document introduces the IGMP for user Authentication Protocol (IGAP). IGAP extends the existing IGMPv2 protocol to add user authentication functionality. IGAP enables an IP multicast service provider to authenticate requests to join a specific multicast group based on user information.

## Table of Contents

## 1. Introduction

IP Multicast applications are becoming more common. One issue is the lack of access control and the ability to effectively collect per-user usage information. This concern discourages the deployment of IP multicast services.

The current IP multicast model provides by nature a non-secure non-controlled way for hosts attached to a network to access multicast traffic. The wide deployment of non-shared access networks (such as dial, DSL or switched Ethernet) facilitates providing access control for multicast traffic.

This document introduces IGMP for user Authentication Protocol (IGAP). 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 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. Authenticated and authorized requests enable a provider to effectively collect the usage information for a particular multicast group.

## 2. Motivations

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.

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. This model can make it difficult for a service provider to generate enough revenue to sustain multicast services such as IP multicast based Internet TV.

The wide deployment of non-shared access networks (such as dial, DSL or switched Ethernet) enables a provider to protect its revenue sources by means of controlling user's access to multicast traffic.

A provider can enforce such access control through static configuration on the last hop network devices such as 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 check. This leads to the need for a comprehensive way to authenticate and authorize end systems before they are granted access to some multicast groups.

Another issue that prevents the wide deployment of IP multicast service is the lack of multicast network management functions such as effective multicast accounting. For example, when deploying IP multicast based Internet TV, a service provider wants to collect some accounting information for a specific TV program such as how many viewers for this TV program and how long they watch this TV program. This accounting information is very important for content providers who own the TV programs.

IGAP introduced in this document enables the dynamic multicast receiver access control for non-shared access networks as well as effective multicast accounting. Hence it encourages the deployment of new commercial IP multicast services.

IGAP uses a user-based authentication model whereby the authentication procedures added to IGMP simply bind the user to a specific host for the duration of group membership. The benefits of a user-based model are well known. The decoupling of service identity from host addressing offers operational simplicity, in particular with respect to adds, moves, and changes.

IGAP can also be used in non-commercial environments such as enterprises. For example, in an enterprise where switched Ethernet is widely deployed in the last hop, IGAP can be used for closed videoconference. IGAP provides a mechanism to allow the access to the videoconference, only if the user is an authenticated user who is all owed to join the videoconference.

## 3. Applicability Statement

IGAP is designed to add authentication capability to IGMP transactions controlling multicast group membership. The transactions flow between an IGAP client (host or host proxy) and an authorizing gateway (authGW). The authGW is assumed to be 1 hop from the IGAP client, such that the client does not have a route that bypasses the authGW (or set of authGWs). An IGAP client MUST authenticate itself to an authGW in order to join a multicast group.

The IGAP model further assumes that the authGW that gates access to the multicast group is a trusted entity by the hosts and that the hosts desiring multicast access are untrusted by the authGW. The authGW is assumed to have access to authoritative information as to IGAP client privileges and is able to log usage statistics.

## 4. IGAP Protocol Overview

IGAP is based on IGMPv2 [IGMPv2] and simply extends the IGMPv2 message format. Details not clearly specified in this document default to those specified in IGMPv2. For example, all IGAP messages described in this document are sent with IP time-to-live (TTL) set to 1, and use the IP Router Alert option [IPRA] in their IP header as per the IGMPv2 requirements.

Like IGMP, IGAP messages are encapsulated in IP datagrams and the IP protocol number in the IP header is 2. The value in the IGMPv2/IGAP Type field in the header permits IGAP messages to be distinguished from IGMP messages.

IGAP specifies different behaviors for IGAP clients and for authGWs. If a router also wants to join some multicast groups, it can perform both parts of the protocol.

IGAP must be implemented on all candidate IGAP clients desiring use

of access controlled multicast services. Similarly IGAP-enabled authGWs must be interposed between IGAP clients and the network to provide suitable access control.

IGAP clients send IGAP Report messages when they want to join an access controlled multicast group or in response to an IGAP Query message. Besides the normal group membership information, the IGAP clients also attach appropriate user authentication information based on the security/authentication mechanism employed/requested by the authGW and requested in the IGAP Query messages received. IGAP clients send IGAP Leave messages to leave a multicast group. AuthGWs periodically verify membership via sending IGAP Query messages.

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 function or mechanism. These functions or mechanisms can be located in different servers, within on server, or even within the authGWs. In this document, we use AAA servers as an example for these functions or mechanisms.

This document describes only the operations between IGAP clients and authGWs, and omits those about between authGWs and back-end Authentication, Authorization, and Accounting functions or mechanisms.

When an authGW receives an IGAP Report message, it first checks whether authentication and authorization are needed or not. If authentication and authorization are needed, the authGW sends the IGAP client's group join request information as well as its user authentication information to an authentication and authorization server. Based on the server's results of authentication and authorization processes, the authGW grants or denies the IGAP client's access request. In addition, the authGW sends an Authentication message to the IGAP client to inform the client of the results.

When the IGAP client's access request is granted, the authGW informs the accounting server to start accounting when it starts forwarding related multicast traffic into the client's network. When the IGAP client leaves the multicast group (either via silent departure or an explicit leave), the authGW informs the accounting server to stop accounting. Once it receives the response from the accounting server, it notifies the IGAP client with an Accounting message.

IGAP can support the use of any authentication mechanism such as, simple and basic password mechanism, more advanced challenge-response mechanism, access token, etc. This document only specifies the protocol supports for both the password mechanism and the

challenge-response mechanism. IGAP is designed to be extensible and the supports for other authentication mechanisms can be easily specified.

When authentication, authorization and accounting are not needed for a specific multicast group, an authGW sends a Notification message to the IGAP client when it receives its IGAP Report or Leave messages. When an authGW gets no response from the AAA servers or it does not receive the requested multicast traffic from upstream multicast routers, it sends an Error message to the affected IGAP clients.

## 5. IGAP Message Formats

IGAP messages have the following 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    |  Report Type  | # of Aux (N)  |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Auxiliary Data Records
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...
```

where each Auxiliary Data Record has the following format:

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
|   Aux Type    | Aux Data Len  |    Aux Data (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-...
```

Upper 8 bytes of the header correspond to those of IGMPv2. This part is called "IGMPv2 compatible part".  The next 3 bytes, called the "IGAP standard part", are used by all IGAP packet types and carry Version, Report Type, Number of Auxiliary Data Record fields. The Auxiliary Data Records are used to carry the information needed for authentication and accounting. This part is called "IGAP optional part".

### 5.1. Type

Currently, there are three types of IGAP messages.

0x40 = IGAP Membership Report (IGAP Join)

A membership report sent from  the IGAP client to authGW, is used to join a multicast group, and to reply to an IGAP Membership Query sent from authGW. Source address of IP header is IP address of the IGAP client sending the packet, and destination address of IP header is desired Group Address. Other details follow those of the IGMPv2 Membership report.

0x41 = IGAP Membership Query

This type, sent from authGW to IGAP clients, asks for the current status of multicast packet reception and Group Address. As in IGMPv2, there are two sub-types: General Query and Group Specific Query. The destination address of the former is the all-system multicast group (224.0.0.1). For the latter, it' is the Group Address that IGAP clients are currently receiving. The features of these sub-types are same as per IGMPv2.

0x42 = IGAP Leave Group

This type, sent from  IGAP client to authGW, is used to leave a multicast group. The IP Header includes source address (IP address of the IGAP client sending the packet), and destination address (224.0.0.2). Other details  follow those of IGMPv2 Leave message.

## 5.2. Max Response Time

The Max Response Time field is meaningful only for Membership Query messages (Type 0x41), and specifies the maximum allowed time before sending a response; the units are 0.1 seconds. In all other messages, it is set to zero by the sender and ignored by receivers.

To prevent excessive burstiness of IGAP traffic on a subnet, the IGAP clients on the subnet should choose a random delay time less than the Max Response Time, and send their Membership Report after this time.

## 5.3. Checksum

The checksum is the 16-bit one's complement of the one's complement sum of the whole IGAP message (the entire IP payload). This algorithm is as described in IGMPv2.

## 5.4. Group Address

In a Membership Query message, the group address field is set to the group address being queried. 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.

## 5.5. Version

Version field is set to 0x10 as the value to identify IGAP packets.

## 5.6. Report Type

Report Type field indicates the type of message transferred within the IGAP packet. Usage of this field is described later.

In Type 0x40 (IGAP Join), there are four Report Types as follows.

```
0x01 : Basic Join
0x02 : Password Mechanism Join
0x03 : Challenge-Response Mechanism Join Challenge Request
0x04 : Challenge-Response Mechanism Join Response
```

In Type 0x41 (IGAP Query), there are seven Report Types as follows.

```
0x21 : Basic Query
0x22 : Challenge-Response Mechanism Challenge
0x23 : Authentication Message
0x24 : Accounting Message
0x25 : Notification Message
0x26 : Error Message
```

In Type 0x42 (IGAP Leave), there are four Report Types as follows.

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

## 5.7. # of Aux (N)

This field indicates the number of Auxiliary Data Record in the packet.

## 5.8. Reserved

This field is set to 0xff.

## 5.9. Auxiliary Data Records

An IGAP packet can contain 0 or many Auxiliary Data Records. The Aux Type field is 1 byte long and specifies the type of the Auxiliary Data Record. The Aux Data Len field is 1 byte long and specifies the length of the auxiliary data in bytes. The Aux Data field contains

the appropriate data. This document only specifies the following Auxiliary Data Records.

### 5.9.1. User Account (Aux Type 0x01) used in all Type

The User Account record contains the account information of a user. This record is only used in the Join or Leave messages.

### 5.9.2. Password (Aux Type 0x11): used in Type 0x40 and 0x42

The Password record contains the password of a user account. This record is only used in the Join or Leave messages and in the environment where password authentication mechanism is used.

### 5.9.3. Challenge Value (Aux Type 0x12): used in Type 0x41

The Challenge Value record contains the challenge information. This record is only used in the Query messages and in the environment where challenge-response authentication mechanism is used. The first byte of the Aux Data field indicates the challenge ID.

### 5.9.4. Response Value (Aux Type 0x13): used in Type 0x40 and 0x42

The Response Value record contains the response information. This record is only used in the Join or Leave messages and in the environment where challenge-response authentication mechanism is used. The first byte of the Aux Data field indicates the challenge ID.

### 5.9.5. Authentication Result (Aux Type 0x21): used in Type 0x41

The Authentication Result record contains the result of user authentication. This record is only used in the Authentication messages.

The Aux Data field is 1 byte long. This document specifies the following values.

```
0x11 : Success
0x12 : Reject
```

### 5.9.6. Accounting Action Result (Aux Type 0x22): used in Type 0x41

The Accounting Action Result record contains the result of the accounting actions that an authGW takes. This record is only used in

the Accounting messages.

The Aux Data field is 1 byte long. This document specifies the following values.

```
0x11 : Start
0x12 : Stop
```

### 5.9.7. Notification (Aux Type 0x23): used in Type 0x41

The Notification record contains the information about the status in IGAP process. This record is only used in the Notification messages.

The Aux Data field is 1 byte long. This document specifies the following values.

```
0x11 : Multicast packets delivery start
0x12 : Multicast packets delivery stop
```

### 5.9.8. Error (Aux Type 0x24): used in Type 0x41

The Error record contains the error information in IGAP process. This record is only used in the Error messages.

The Aux Data field is 1 byte long. This document specifies the following values.

```
0x11 : Response time out of authentication server
```

## 6. IGAP Packet Type

IGAP Packet type is determined by the Report Type field.
Some types of packets take an auxiliary data. When other auxiliary data which is not specified this document added to IGAP packet, it should not be discarded it. We show a auxiliary data by each IGAP packet we need at least here.

### 6.1. Basic Join

Type : 0x40
Group Address : connected group address
Report Type : 0x01
Auxiliary data record : User Account

This packet type is used in the case which the join process does not require the authentication process.

## 6.2. Password Mechanism Join

Type : 0x40
Group Address : connected group address
Report Type : 0x02
Auxiliary data record : User Account and Password

This packet type is used in the case which the join process require password authentication process.

## 6.3. Challenge-Response Mechanism Join Challenge Request

Type : 0x40
Group Address : connected group address
Report Type : 0x03
Auxiliary data record : User Account

This packet type is used to initiate the challenge process for Challenge-response authentication mechanism. AuthGW received this packet issues  Challenge packet.

## 6.4. Challenge-Response Mechanism Join Response

Type : 0x40
Group Address : connected group address
Report Type : 0x04
Auxiliary data record : User Account and Response Value

This packet type is used to respond to the Challenge issued by the authGW. The Response Value is determined from two parameters. One is the Challenge Value, which is in the Challenge packet. The other is the value calculated from the user password by MD5 [MD5].

## 6.5. Basic Query

This packet type is used in the case which authGW checks whether the IGAP client(s) are receiving multicast traffic at regular intervals, and authGW confirms the IGAP client's request to leave a multicast group. There are two sub-types of Basic Query, as described in section 3.1. In the case of General Query, i.e. destination address of IP header is the all-systems multicast group (224.0.0.1), it's called the General-and-Basic Query. In this sub-type, the value of Group Address field is set to zero. In the case of Group Specific Query, i.e. destination address of IP header is the desired group address, it's called the Group-Specific-and-Basic Query. In this sub-type, the value of Group Address field is equal to the value of the desired destination address. This packet type doesn't have to have IGAP optional part.

### 6.5.1. General-and-Basic Query

Type : 0x41
Group Address : (set to zero)
MaxRespTime : given value (default : 0x64)
Report Type : 0x21

### 6.5.2. Group-Specific-and-Basic Query

Type : 0x41
Group Address : connected group address
MaxRespTime : given value (default : 0x64)
Report Type : 0x21

### 6.6. Challenge-Response Mechanism Challenge

Type : 0x41
MaxRespTime : given value (default : 0x64)
Group Address : connected group address
Report Type : 0x22
Auxiliary data record : User Account and Challenge Value

This packet type is used in the case which authGW responds to CHAP Join Challenge Request from an IGAP client. The AuthGW sends this packet to notify Challenge Value. The IGAP client that received this packet replies with a Join Response packet.

### 6.7. Authentication Message

Type : 0x41
MaxRespTime : given value (default : 0x64)
Group Address : connected group address
Report Type : 0x23
Auxiliary data record : User Account and Authentication Result

This packet type is used in the case which authGW provides information about the result of authentication. The Message field holds one of two values: either authentication succeed or authentication reject. The process adopted by the IGAP client upon receiving this packet type is up to implementation, however, neither value must impact other IGAP process.

### 6.8. Accounting Message

Type : 0x41
MaxRespTime : given value (default : 0x64)
Group Address : connected group address

Report Type : 0x24
Auxiliary data record : User Account and Accounting Action Result

This packet type is used in the case which authGW provides information about accounting status. The Message field holds one of two values: one indicates the start of accounting, and the other indicates the termination of accounting. The process adopted by the IGAP client upon receiving this packet type is up to implementation, however, neither value must impact other IGAP process.

### 6.9. Notification Message

Type : 0x41
MaxRespTime : given value (default : 0x64)
Group Address : connected group address
Report Type : 0x25
Auxiliary data record : User Account and Notification

This packet type is used in the case which authGW provides information about a correct status in IGAP process, except authentication and accounting. The process adopted by the IGAP client upon receiving this packet type is up to implementation, however, neither value must impact other IGAP process.

### 6.10. Error Message

Type : 0x41
MaxRespTime : given value (default : 0x64)
Group Address : connected group address
Report Type : 0x26
Auxiliary data record : User Account and Error

This packet type is used in the case which authGW provides information about an error status in IGAP process, except authentication and accounting. The process adopted by the IGAP client upon receiving this packet type is up to implementation, however, neither value must impact other IGAP process.

### 6.11. Basic Leave

Type : 0x42
Group Address : connected group address
Report Type : 0x41
Auxiliary data record : User Account

This packet type is used in the case which the leave process does not require the authentication process.

## 6.12. Password mechanism Leave

Type : 0x42
Group Address : connected group address
Report Type : 0x42
Auxiliary data record : User Account and Password

This packet type is used in the case which the leave process require password authentication process.

## 6.13. Challenge-Response Mechanism Leave Challenge Request

Type : 0x42
Group Address : connected group address
Report Type : 0x43
Auxiliary data record : User Account

This packet type is used to initiate the challenge process for challenge-response authentication. AuthGW received this packet issues Challenge packet.

## 6.14. Challenge-Response Mechanism Leave Response

Type : 0x42
Group Address : connected group address
Report Type : 0x44
Auxiliary data record : User Account and Response Value

This packet type is used to respond to the Challenge issued by the authGW. The algorithm used to calculate the Response value is the same method of Challenge-response Join Response.

## 7. IGAP State Machines

This Section describes examples of IGAP State Machines. The example of FSM is shown about each pattern as follows.

Note:

-There are two ways of leaving a multicast group: General Leave Process and Fast Leave Process. General Leave Process basically equals the leave process of IGMPv2.
Fast Leave Process dispenses with IGAP Query packets. When AuthGW receives IGAP Leave message from the user client, it stops sending multicast packets without sending IGAP Query packet. Fast Leave Process is recommended when fast response for IGAP Leave is needed.

-About operation of General Query, it is the same as FSM of IGMPv2.

### 7.1. FSM for Client

```
   C1[Non Member]:
    if join group{
        send Challenge-Request-Join;
        start Challenge-Timer;
        transition C2;
    }

   C2[Waiting Challenge Member]:
    if Challenge received{
        send Challenge-Response-Join;
        stop Challenge-Timer;
        start Authenticated-Timer;
        transition C3;
    }
    else(Challenge-Timer expired){
        stop Challenge-Timer;
        transition C1;
    }

   C3[Waiting Authentication Message Member]:
    if Authentication-Message(Reject) received
      or Error-Message(Response time out) received
      or Authenticated-Timer expired{
        stop Authenticated-Timer;
        transition C1;
    }
    else if Authentication-Message(Success) received
        stop Authenticated-Timer;
        transition C4;
    }

   C4[Idle Member]:
    if General-Query received
      or Group-Specific-Query received{
        start Delaying-Timer;
        transition C5;
    }
    else if leave group{
        send Basic-Leave;
        transition C6;
    }

   C5[Delaying Member]:
    if leave group{
        send Basic-Leave;
        stop Delaying-Timer;
        transition C6;
    }
```

```
    else(Delaying-Timer expired){
        send Challenge-Request-Join;
        stop Delaying-Timer;
        start Challenge-Timer;
        transition C2;
    }

   C6[Waiting Accounting Message Member]:
    if Accounting-Message(Stop) received{
        transition C1;
    }
    else if General-Query received{
        send Basic-Leave;
        continue(no transition);
    }
```

## 7.2. FSM for AuthGW

```
   A1[No Transfer Present]:
    if Challenge-Request-Join received{
        send Challenge;
        start Response-Timer;
        transition A2;
    }
    else if Basic-Leave received{
        send Accounting-Request(Stop);
        transition A8;
    }

   A2[Waiting Challenge-Response]:
    if Challenge-Response-Join received{
        send Authentication Request;
        stop Response-Timer;
        start Authentication-Timer;
        transition A3;
    }
    else(Response-Timer expired){
        stop Response-Timer;
        transition A1;
    }

   A3[Waiting Authentication-Response]:
    if Access-Reject received{
        send Authentication-Message(Reject);
        stop Authentication-Timer;
        transition A1;
    }
```

```
    else if Access-Accept received{
        send Accounting-Request(Start);
        send Authentication-Message(Success);
        stop Authentication-Timer;
        start Accounting-Timer;
        transition A4;
    }
    else(Authentication-Timer expired){
        send Error-Message(Response time out);
        stop Authentication-Timer;
        start Validity-Timer;
        transition A5;
    }

   A4[Waiting Accounting-Response(Start)]:
    if Accounting-Response received{
        send Accounting-Message(Start);
        stop Accounting-Timer;
        start Validity-Timer;
        transition A5;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        start Validity-Timer;
        transition A5;
    }

   A5[Transfer Present]:
    if Challenge-Request-Join received{
        if Validity-Timer < Validity-Period{
            continue(no transition);
        }
        else(Validity-Timer expired){
            send Accounting-Request(Stop);
            stop Validity-Timer;
            start Accounting-Timer
            transition A6;
        }
    }
    else if Basic-Leave received{
        if General Leave Process{
            send Group-Specific-Query;
            stop Validity-Timer;
            start Last-Member-Query-Interval-Timer;
            transition A7;
        }
```

```
        else if Fast Leave Process{
            send Accounting-Request(Stop);
            stop Validity-Timer;
            transition A8;
        }
    }

   A6[Waiting Accounting-Response(Stop)]:
    if Accounting-Response received{
        send Accounting-Message(Stop);
        send Challenge;
        stop Accounting-Timer;
        start Response-Timer;
        transition A2;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        start Validity-Timer;
        transition A5;
    }

   A7[Waiting Membership Report]:
    if Challenge-Request-Join received{
        if Validity-Timer < Validity-Period{
            stop Last-Member-Query-Interval-Timer;
            clear Last-Member-Query-Counter;
            transition A5;
        }
        else(Validity-Timer expired){
            send Accounting-Request(Stop);
            stop Validity-Timer;
            stop Last-Member-Query-Interval-Timer;
            start Accounting-Timer;
            clear Last-Member-Query-Counter;
            transition A6;
        }
    }
    else(Last-Member-Query-Interval-Timer expired){
        if Last-Member-Query-Counter < Last-Member-Query-Count{
            send Group-Specific-Query;
            restart Last-Member-Query-Interval-Timer;
            increment Last-Member-Query-Counter;
            continue(no transition);
        }
```

```
        else(Last-Member-Query-Counter expired){
            send Accounting-Request(Stop);
            stop Validity-Timer;
            stop Last-Member-Query-Interval-Timer;
            start Accounting-Timer;
            clear Last-Member-Query-Counter;
            transition A8;
        }
    }

   A8[Waiting Accounting-Response(Stop) for Leave]:
    if Accounting-Response received{
        send Accounting-Message(stop);
        stop Accounting-Timer;
        transition A1;
    }
    else(Accounting-Timer expired){
        send Error-Message(Response time out);
        stop Accounting-Timer;
        transition A1;
    }
```

## 8. List of Timers, Counters

This section describes the parameters set in AuthGW and Client when supporting IGAP processes.

### 8.1. Timers and Counters for Client

#### 8.1.1. Challenge-Timer

It controls waiting time from sending Join message to receiving Challenge Message.

#### 8.1.2. Authenticated-Timer

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

#### 8.1.3. Delaying-Timer

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

### 8.2. Timers and Counters for AuthGW

### 8.2.1. Robustness Variable

It is the same meaning as IGMPv2.
Default: 2.

### 8.2.2. Response-Timer

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

### 8.2.3. Authentication-Timer

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

### 8.2.4. Accounting-Timer

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

### 8.2.5. Validity-Period (Validity-Timer)

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

### 8.2.6. Query-Response-Interval (Query-Response-Interval-Timer)

It is the same meaning as IGMPv2. The Max Response Time inserted into the periodic General Queries.
Default: 100 (10 seconds)

### 8.2.7. Last-Member-Query-Interval (Last-Member-Query-Interval-Timer)

It is the same meaning as IGMPv2. The Last-Member-Query-Interval is the Max Response Time inserted into Group-Specific Queries sent in response to Leave Group messages, and is also the amount of time between Group-Specific Query messages.
Default: 10 (1 second)

### 8.2.8. Last-Member-Query-Count

It is the same meaning as IGMPv2. The Last-Member-Query-Count is the number of Group-Specific Queries sent before the router assumes there

are no local members.
Default: the Robustness Variable.

## 9. Security Considerations

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

IGAP supports a number of authentication mechanisms and inherits the security concerns of each, especially when there is shared media between the IGAP client and the authGW.

IGAP does not encrypt multicast traffic. For multicast content encryption related technology, please refer to other IETF work. IGAP does not obstruct snooping of multicast traffic by unauthorized hosts 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.

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

## 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
Phone : 1 978 288 7482
Email : haixiang@nortelnetworks.com

Wassim Tawbi
Nortel Networks
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@trl.mei.co.jp