Network Working Group
Request for Comments: 4678
Category: Informational

A. Bivens
IBM Research

September 2006

Server/Application State Protocol v1

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control, or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this document should exercise caution in evaluating its value for implementation and deployment. See RFC 3932 for more information.

Abstract

Entities responsible for distributing work across a group of systems traditionally do not know a great deal about the ability of the applications on those systems to complete the work in a satisfactory fashion. Workload management systems traditionally know a great deal about the health of applications, but have little control over the rate in which these applications receive work. The Server/Application State Protocol (SASP) provides a mechanism for load balancers and workload management systems to communicate better ways of distributing the existing workload to the group members.

Table of Contents

1. Introduction

1.1. Overview

The Server/Application State Protocol is designed to enable load balancers or schedulers (1) to receive traffic weight recommendations from Workload Managers, (2) to register with Workload Managers members of load balancing/scheduling groups, and (3) to enable Workload Managers to suggest new load balancing group members to load balancers and schedulers

The figure below shows where the SASP entities are in typical load balancing topology.

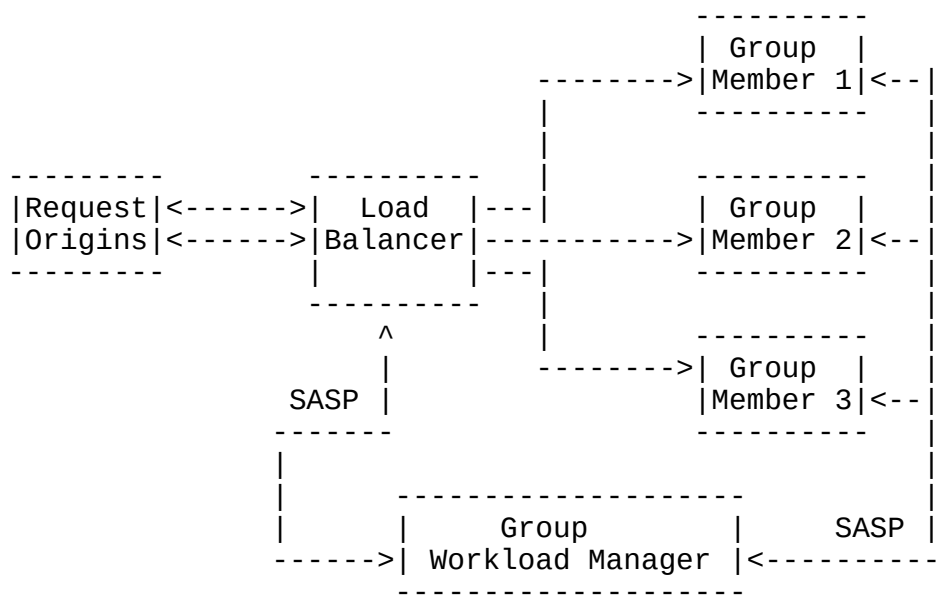


Figure 1

SASP is a binary protocol that facilitates communication from load balancers/schedulers to Workload Managers. The connection between the Group Workload Manager (GWM) and the load balancer/scheduler is expected to be a long-running TCP connection. In SASP interactions, the GWM acts as a SASP server waiting to receive connections from the other SASP components. Server port 3860 has been registered with the IANA for SASP communications. It is expected that all SASP components are configured with the DNS name of the GWM to develop

this connection. Security in SASP is handled by transporting binary messages over Secure Socket Layer/Transport Layer Security (SSL/TLS). This document only describes the message format and protocol behavior above the connection and security layers. Connection and security aspects including SSL's authentication and encryption will be implementation specific.

1.2. Identities

SASP identifies a load balancer by a UTF-8 string called a "LB UID". A group of "equivalent" servers providing a service is identified by a UTF-8 string called a "Group Name", which is interpreted in the context of the LB UID. A server is identified by its IP address and (optional) port and protocol numbers. A GWM is only identified implicitly as the entity on the other end of the TCP connection from a load balancer or group member. All of these identifiers are local; there are no globally unique identifiers. The LB UID and GroupName fields are unstructured so that components could assign values to these fields that are meaningful to an administrator. For example, in many cases, a load balancer would use the name an administrator provided for the serverfarm group as the groupname in a SASP-specified group. Since the naming options in industry load balancers do not carry explicit naming restrictions, SASP naming options also carry no naming restrictions.

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Conventions Used in This Document

- Load Balancer - Entity responsible for distributing requests amongst the available members.
- Member - Machine, process, or application used to service requests.
- Group Workload Manager (GWM) - Entity responsible for reporting or managing a group of members on multiple machines.

4. General Message Structure

Any string interpreted by the group workload manager is assumed to use UTF8. Components implementing SASP MUST support the printable ASCII subrepertoire of UTF8 (0x20-0x7E). Components MAY also choose to provide support for additional UTF8 character encodings. It is

recommended that customers using SASP-enabled products configure the string-generating components (load balancers and group members) to use the same character repertoire.

Many of the SASP structures involve the transfer of multi-byte integer values. In all cases where multi-byte integer values are used, they are considered to be in network-byte order (big-endian).

SASP is organized into several message components. For extendibility and ease of processing, each message component is described in a TLV (Type, Length, Value) format. An illustration of the SASP structure can be found in the example below. The first section is the header followed by the message component type. As mentioned, the header, message component, and all other components have a TLV format. Each component value contains a variable number of fields, some of which refer to upcoming components (explained component descriptions are in upcoming sections). After the first message component, any number of additional components may be included (as stipulated in the fields of the message type).

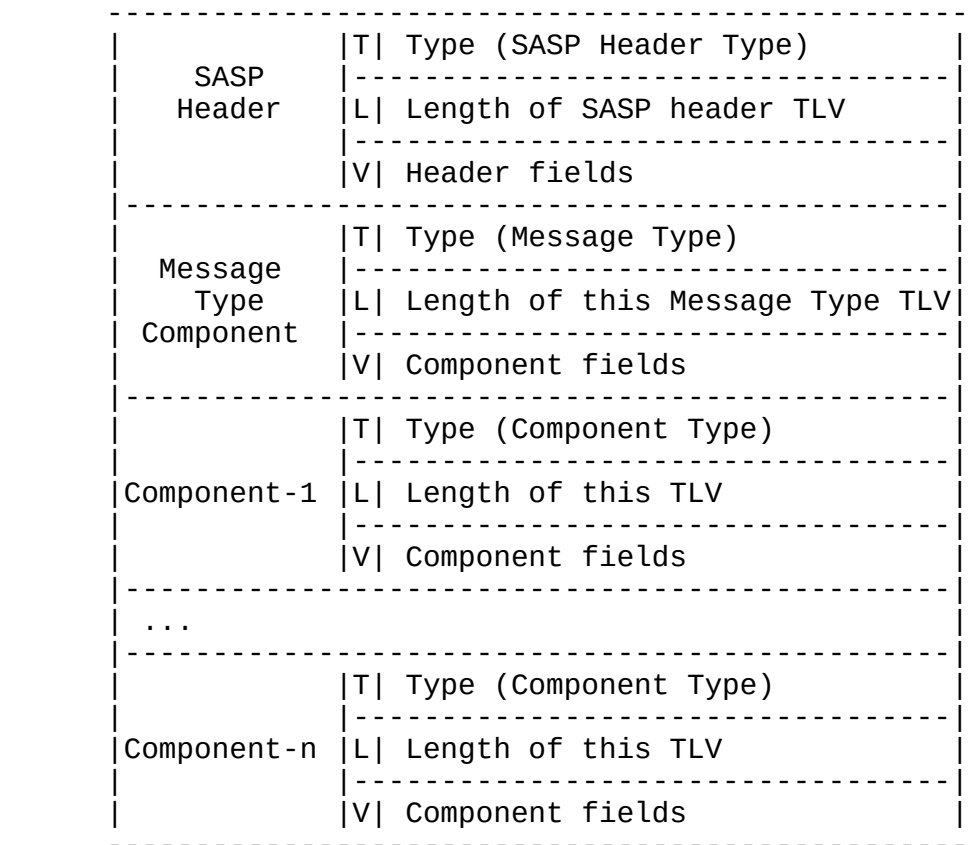


Figure 2

4.1. TLV Structure

An illustration of the TLV format is shown below. The Type is a two-byte field containing a binary value for the component type. The Length is a two-byte field containing the size of the TLV in bytes (including the Type and Length fields). The Value field is a variable-length field that actually contains the data of the component.

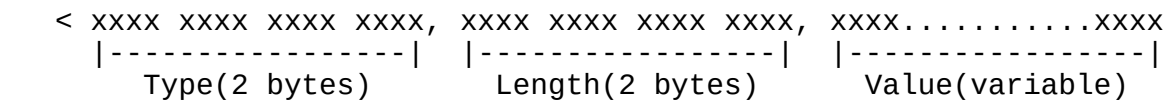


Figure 3

4.2. Component Types

The TLV structure requires a type value for each protocol component. All SASP types are listed in this section.

Reserved 0x0000-0x1000

Message Types

Registration Request 0x1010

Registration Reply 0x1015

DeRegistration Request 0x1020

DeRegistration Reply 0x1025

Get Weights Request 0x1030

Get Weights Reply 0x1035

Send Weights 0x1040

Set Load Balancer State Request 0x1050

Set Load Balancer State Reply 0x1055

Set Member State Request 0x1060

Set Member State Reply 0x1065

Utility Component Types

SASP Header 0x2010

Singular Component Types

Member Data 0x3010

Group Data 0x3011

Weight Entry Data 0x3012

Member State Instance 0x3013

Group Component Types

Group of Member Data 0x4010

Group of Weight Entry Data 0x4011

Group of Member State Data 0x4012

Reserved 0xF000-0xFFFF

4.3. SASP Protocol Header

An illustration of the SASP Header is found in the table below. It is expected that every message will start with the SASP Protocol Header component.

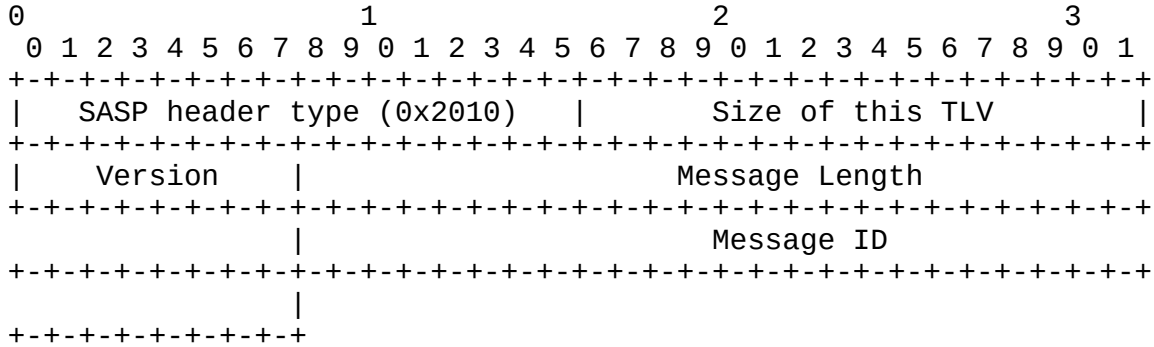


Figure 4

- Version: The version of the protocol used in this message.

- Message Length: A 4-byte signed integer value representing the total length of the SASP message. It is said to be a signed 4-byte value to make any Java implementations easier (or any other implementations without unsigned values); however, no negative lengths are valid.

- Message ID: Each request message is given a 4-byte Message ID by the message originator, which is simply returned in the Message ID field of the reply. This field is meant to assist the requester in correlating replies to the appropriate request when many requests have been sent. In the Send Weights message (the only message transaction that has no reply), this field serves no purpose.

4.4. Version Negotiation

To negotiate the version of the protocol used by the entities involved in the connection, the GWM views the version included in the load balancer request as the load balancer's proposed version.

If the GWM supports the version proposed by the load balancer, it will respond to the connection with the appropriate response code and the load balancer's proposed version in the response header. This proposed version should be the version used for all messages in this connection.

If the GWM does not support the version proposed by the load balancer, the GWM will respond with a "message not understood" response code and the GWM's highest supported SASP version in the version field of the response header. This is an indication for the load balancer to come down to GWM's SASP version level.

5. Singular Protocol Components

The most basic of SASP components are singular components because they describe a single instance of a member, member resource, member weight, or group. Some of the SASP components reuse other SASP components. When this is the case, any component being reused by a base component will simply be given immediately following the base component. Some examples of this technique are seen and explained in the Weight Entry and Member State Instance components.

5.1. Member Data Component

The member data component describes a particular member and is referred to by other components.

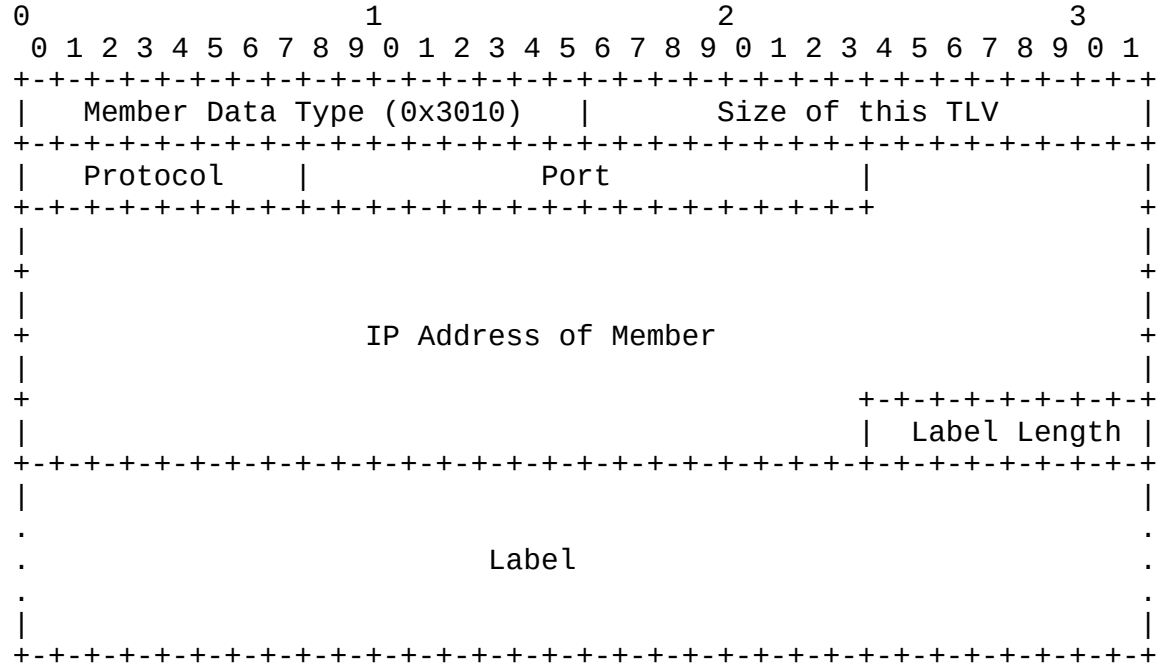


Figure 5

- Protocol: The assigned number of the IP transport layer used in the Protocol Field of the IP header. These are defined in [RFC1700]; however, a current list is maintained at http://www.iana.org.
  for example: TCP = 0x06, UDP = 0x11, etc.

- Port: The port number used for communication to the member. *** A value of 0 can be given for the Protocol and Port to signify a system level member. However, 0 shouldn't be perceived as a wildcard for either Port or Protocol fields (i.e., a deregistration request that includes a MemberData component with a 0 for the port doesn't mean deregister all applications listening on any port of that IP and protocol).

- IP Address: The current format is described by the following 16 bytes, where IPv4 addresses are represented as "IPv4-compatible IPv6 addresses" [RFC4291]. In the following example, the x's and zeros represent 4-bit hex values. The x's describe arbitrary hex values.

  IPv4 Address: 00 00 00 00 00 00 00 00 00 00 00 00 xx xx xx xx

  IPv6 Address: xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx

- Label length: The length, in bytes, of the label string to follow.

- Label: A UTF8 string that may be set while registering a member. This string is opaque to the GWM and is simply included with any correspondence containing the member data component. Note that the size of this label is <= 255 bytes. Because UTF8 character encodings may be up to 6 bytes, care must be exercised by the load balancer or member to make sure the UTF8 string it sends the GWM is in fact <= 255 bytes.

5.2. Group Data Component

The group data component simply describes a group with which to associate other singular components.

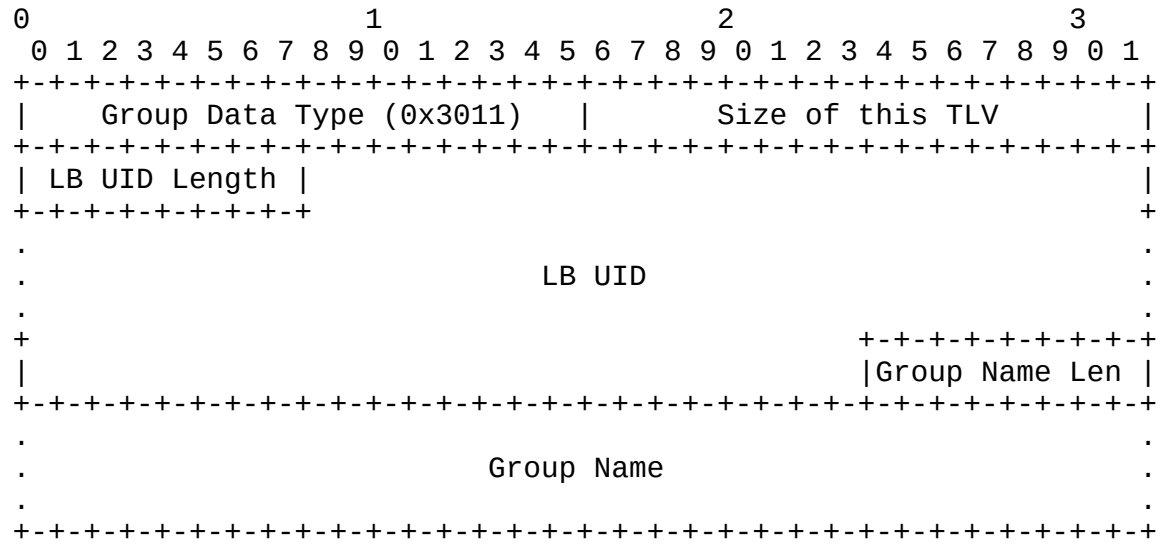


Figure 6

- LB UID Length: Length of the LB UID to follow (in bytes).
- LB UID: A UTF8 string used as a unique identifier and a context for the Group Name (e.g., a UTF8 representation of the MAC address of the load balancer or some type of Universally Unique Identifier (UUID)). This string is used by the Group Workload Manager to associate application registration and deregistration, and to set state messages with the correct load balancer. This unique identifier should not be any longer than 64 bytes.
- Group Name Len: Length of the Group Name field to follow (in bytes).
- Group Name: A UTF8 string the load balancer has chosen to tell the Group Workload Manager that members being registered with this Group Name are equivalent in function. In Get Weight and DeRegistration messages, the Group Name may be omitted (Group Name Length = 0) to indicate all groups from the associated load balancer.

5.3. Weight Entry Data Component

The Weight Entry Component is used by the get and send weight messages to associate a weight with a particular member (or Member Data). It also uses an opaque member state field and a general member flags field to denote extra information about a member (described below). When the Weight Entry component is used, the Member Data TLV it refers to is listed first, immediately followed by the Weight Entry TLV.

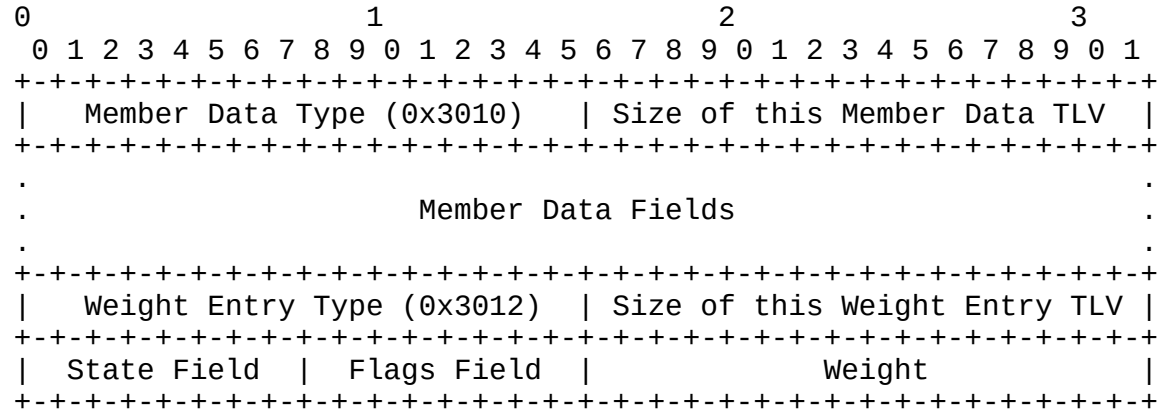


Figure 7

- State Field: This field is used by the member to communicate state information to the scheduler. The information placed in this field is opaque to the GWM and will simply be forwarded to the scheduler with the member weights. There are no defined values for this field.

- Flags Field: This field has several flag values that describe several attributes of the member.

  A. Contact Success Flag (set by the GWM): describes whether the member is currently running. If the contact success flag is off, this member should be avoided by the load balancer.

     + xxxx xxx1 The GWM has located this running system or application.

     + xxxx xxx0 The GWM has not located this running system or application.

  B. Quiesce Flag (set by the load balancer or Member): used when an administrator would like to temporarily remove a member from the weight calculation, but not deregister it from the

group. When quiesced, the member will still show up in the weights, but the quiesce flag will be set, and its weight will be zero. When the administrator returns this member to active, the quiesce flag will be 0, and a weight will be provided. If the quiesce flag is on, this member should be avoided by the load balancer.

+ xxxx xx1x The member is quiesced.

+ xxxx xx0x The member is active (not quiesced).

C. Registration Flag (set by the GWM): stores how the member was registered.

+ xxxx x1xx This member has been registered by the load balancer/scheduler.

+ xxxx x0xx This member has registered itself.

D. Confident Flag (set by the GWM): describes whether the GWM has knowledge of this member’s state. If this flag is off for only some of the members in the group while the remaining members have valid weights, the load balancer should avoid sending work to those members with the confident flag off. If the confident flag is off for all valid group members, the load balancer should disregard any recommendation from the GWM until the confident flag comes back on for at least one member. In this case where all confident flags are off, the load balancer should determine the correct distribution of work by other means (perhaps a different advisor, previously configured static weights, etc.).

The goal of the confident flag is to convey to the load balancer that it should look to other methods of distribution recommendations if the GWM cannot give recommendations for any of the valid group members. If some members of the group have the confident flag on but the contact flag off or the quiesced flag on (meaning these members should always be avoided) while the remaining members of the group have their confident flag off, the load balancer should determine the appropriate distribution of work for those members with the confident flag off by other means.

+ xxxx 1xxx GWM has determined it has knowledge of the state of this member.

+ xxxx 0xxx GWM has no knowledge of the state of this member.

E. Leftmost four bits are reserved (0000 xxxx - 1111 xxxx).

- Weight: This field represents the GWM's recommendation for the relative amount of work that should be sent to this member. This is a 16-bit field with a possible range of 0 to 65536. Load balancers should be prepared to receive a wide range of weight values. Load balancers with limited maximum weight values may restrict the granularity of management by the GWM and in turn cause less than optimal performance. Many existing implementations have supported a minimum raw weight range from 0 to 100.

5.4. Member State Instance Component

The Member State Instance Component is used by the set member state message to indicate the sender's perceived state of the member mentioned. This component is used to set values that will ultimately end up in the WeightEntry component. When the Member State Instance component is used, the Member Data TLV it refers to is listed first, immediately followed by the Member State Instance TLV.

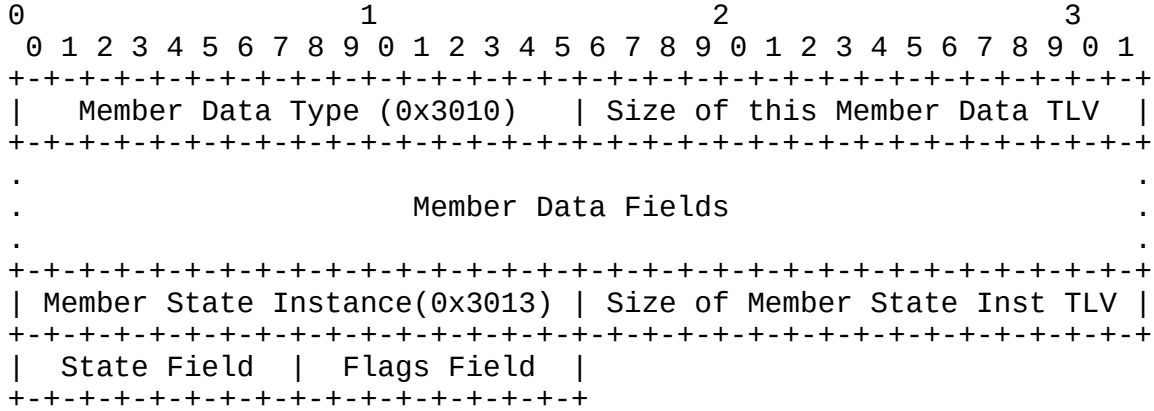


Figure 8

- State Field: This field is used by the member to communicate state information to the load balancer or scheduler. There are no defined values for this field.

- Flags Field: This field describes attributes of the member. Currently the only flag value defined is that of the quiesce flag. The quiesce flag is used when an administrator would like to temporarily remove a member from the weight calculation, but not deregister it from the group. When quiesced, the member will still show up in the weights, but the quiesce flag will be set,

and its weight will be zero. When the administrator returns this member to active, the quiesce flag will be 0, and a weight will be provided.

A. Quiesce Flag

+ xxxx xxx1 The member or load balancer setting this state is quiescing this member.

+ xxxx xxx0 The member or load balancer setting this state is placing the member in a non-quiesced state.

B. Leftmost seven bits are reserved (0000 000x - 1111 111x).

6. Group Protocol Components

Group protocol components each contain a collection of related singular components. In particular, they associate Member Data, Weight Entry, or Member State Instance components to a particular Group Data component. In these cases, the particular "Group of x" component will be immediately followed by the Group Data component. The Group Data component will be immediately followed by any number of singular components the group contains. In figures listed in this document, a component type with an asterisk denotes a component that is repeated a number of times.

6.1. Group of Member Data Component

The "group of member data" component describes a particular group of members and is used in the registration message components.

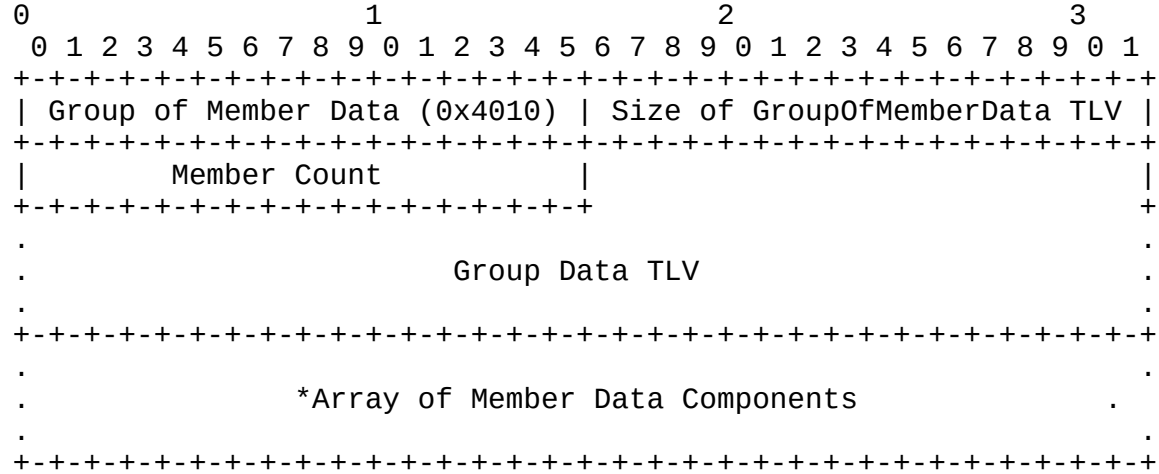


Figure 9

- Member Count: The number of Member Data Components immediately following the Group Data structure.

- Array of Member Data Components: There will be as many Member Data TLVs as Member Count has specified. A load balancer/scheduler would use these components to pass information that would enable the Group Workload Manager to identify the members to associate with this Group Name. The Member Data Component was described in Section 5.1. In DeRegistration messages, the Member Count may be set to 0 to indicate all members of a particular group.

6.2. Group of Weight Data Component

The "Group of Weight Data" Component is used by the get and send weight messages to create a list of Weight Entry Components for a particular group.

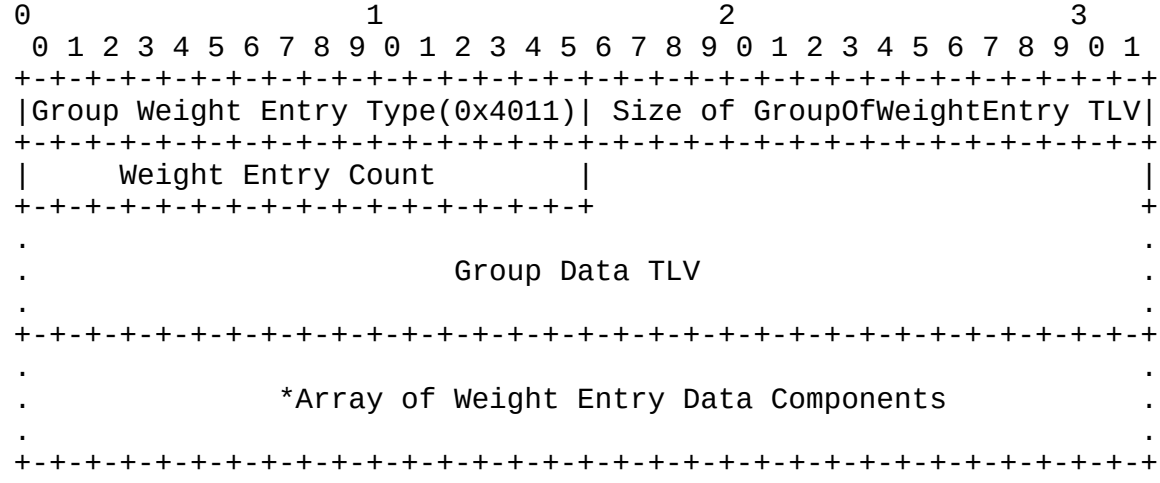


Figure 10

- Weight Entry Count: The number of Member Data / Weight Entry combinations to follow the Group Data TLV.

- Array of Weight Entry Data TLVs: There will be as many [Member Data / Weight Entry] TLVs as Weight Entry Count has specified. Each Weight Entry component is preceded by its corresponding Member Data component as explained in Section 5.3. This Member Data / Weight Entry data combination will repeat to form as many Weight Entry items as the Weight Entry Count specifies.

6.3. Group of Member State Data Components

The "group of member state data" component describes a particular set of members and their corresponding state fields used in the Set Member State messages.

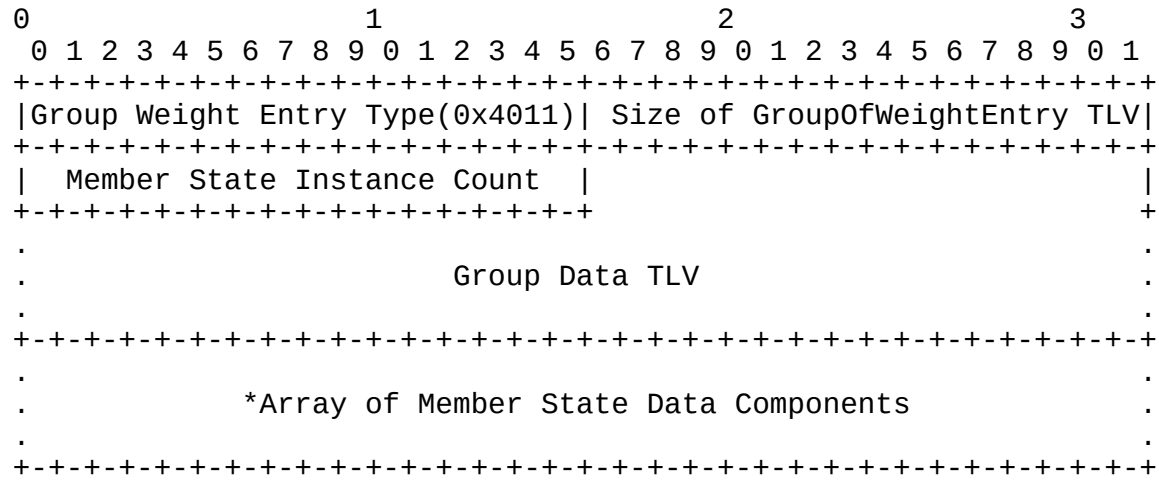


Figure 11

- Member State Instance Count: The number of Member Data / Member State Instance combinations following the Group Data component.

- Array of Member State Data Components: Each Member State Instance component is immediately preceded by its corresponding Member Data component as explained in Section 5.4. This Member Data / Member State Instance combination will repeat to form as many Member State items as the Member State Instance Count specifies.

7. Protocol Messages

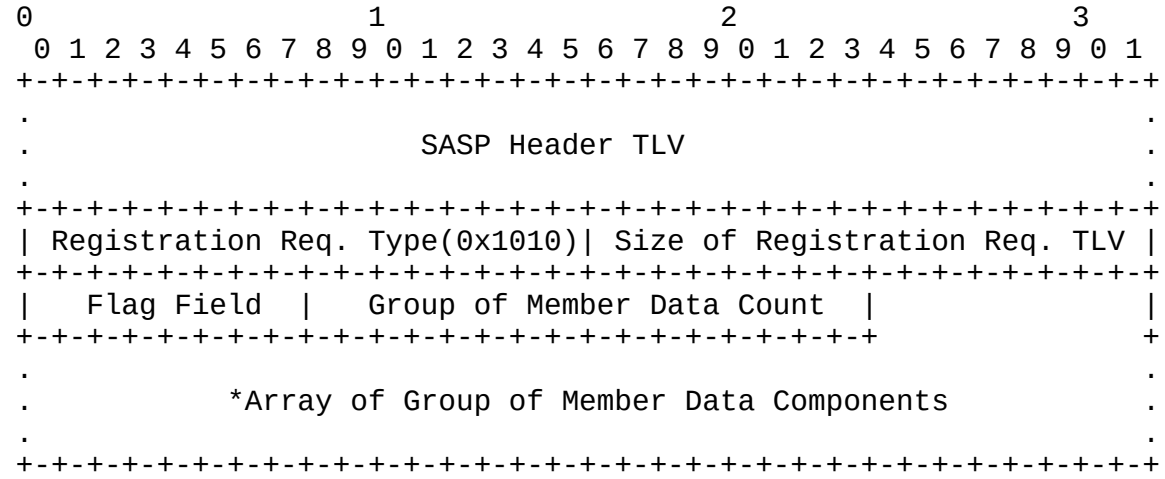
SASP messages are a collection of TLVs (Type, Length, and Value components). The header has no information as to what type of message it is part of; the purpose-specific information is in the message component. This format could facilitate placing more than one message component in a single message; however, this use of multiple message components is not supported in every GWM and could produce indeterminate behavior. Similar to the other protocol components, when a message component needs to involve other components, the additional components immediately follow the message component.

All SASP requests sent to the GWM will be acknowledged with a reply. The reply contains information requested as well as a single-byte response code describing the success of the request. SASP defines some general response codes in the range of 0x00 - 0x3F that may be used regardless of the response message type. However, some request types may cause specific error conditions not covered by the general response codes. The response code range of 0x40 - 0xFF is used for these message-specific response codes. Any given SASP response will only contain one response code (depending on the error type). This section explains the format and purpose of specific SASP messages.

7.1. Registration Request and Reply

This exchange happens between the load balancer/scheduler and the Group Workload Manager as well as between the Group Workload Manager and the member to register the members in a group specified by Group Name. Applications are identified with an IP address, Protocol, and Port. Systems are identified only with an IP Address (Port = 0x0000 and Protocol = 0x00). All members in a group have equivalent functionality, so the Group Workload Manager can direct routers, load balancers, and schedulers to any member in the group. Even though registrations can come from either the load balancer/scheduler or the actual member, member-initiated registrations will only be considered if the Trust flag is set while the state of the load balancer/scheduler is set.

7.1.1. Registration Request



*There will be as many Group of Member Data Components as "Group of Member Data Count" has specified.

Figure 12

- Flag Field

  A. Load Balancer Flag

     + xxxx xxx1 The entity sending this message is the load balancer.

     + xxxx xxx0 The entity sending this message is an Application.

  B. Leftmost seven bits are reserved (0000 000x - 1111 111x).

- Group of Member Data Count: The number of "Group of Member Data" components immediately following the Registration Request component.

- Array of Group of Member Data Components: Each "Group of Member Data" component is immediately followed by Group Data Components and its Member Data components (as described in Section 6.1). In the case where several of these "Group of Member Data" components may be present, the second "Group of Member Data" component only appears after all of the internal components that are referred to by the first "Group of Member Data" component are listed. The format is the same for all subsequent "Group of Member Data" components in the message.

7.1.2. Registration Reply

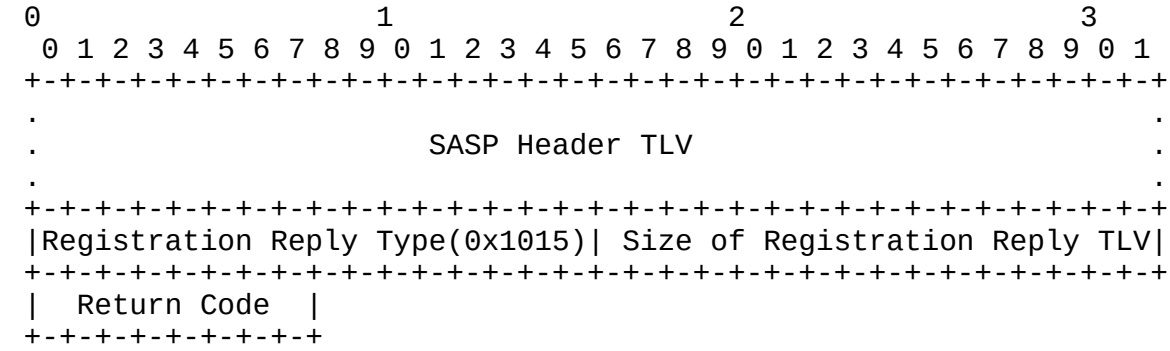


Figure 13

- General SASP return codes (0x00 - 0x3F)

  * 0x00 Successful

  * 0x10 Message not understood

    * 0x11 GWM will not accept this message from the sender. Reasons for this include the following:
      a. The message was not sent by a LB and trust flag is off
      b. LB attempted to address members of a different LB in the message
      c. Vendor specific criteria for this message type were not met.

- Message-Specific return codes (0x40 - 0xFF)

    * 0x40 Member already registered

    * 0x44 Duplicate Member in Request

    * 0x45 Invalid Group (determined by the GWM)

    * 0x50 Invalid Group Name Size (size == 0)

    * 0x51 Invalid LB UID Size (size == 0 or > max)

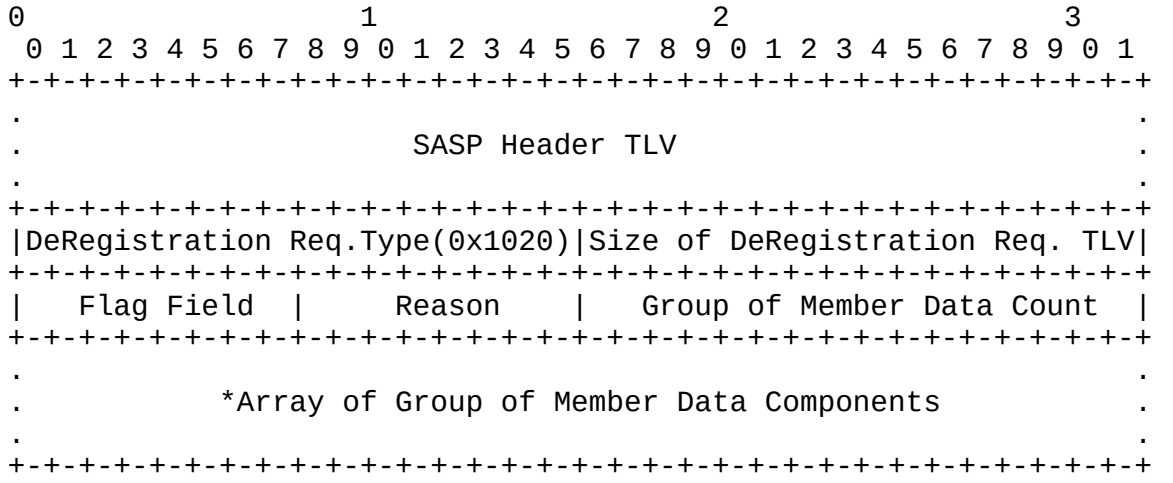
    * 0x61 Member is registering itself, but LB hasn't yet contacted the GWM. This registration will not be processed.

**The Invalid Group error return code refers to the LB or member attempting to form a group that the GWM considers invalid. For example, some GWM vendors may not support the registration of both System and Application members in the same group. To determine what can cause a GWM to return this error code, the vendor's documentation must be consulted.

7.2. DeRegistration Request and Reply

This exchange happens between the load balancer/scheduler and the Group Workload Manager as well as between the Group Workload Manager and the Member to deregister members from a group specified by Group Name with the Group Workload Manager. Even though deregistrations can come from either the load balancer/scheduler or the actual member, member-initiated deregistrations will only be considered if the Trust flag is set with a Set LB State message.

7.2.1. DeRegistration Request



*There will be as many Group of Member Data Components as "Group of Member Data Count" has specified.

Figure 14

- o Flag Field
  - A. Load Balancer Flag
    - + xxxx xxx1 The entity sending this message is the load balancer.
    - + xxxx xxx0 The entity sending this message is an Application.
  - B. Leftmost seven bits are reserved (0000 000x - 1111 111x).
- o Reason: Byte describing the reason for deregistering the group or instance.
  - A. SASP-defined Reason Codes (0x00-0x7F)
    - + 0x00 No reason given.
    - + 0x01 Learned and Purposeful, i.e., a human has deconfigured this member from the load balancer configuration.
    - + 0x80-0xFF Open for vendor specific deregistration reason codes.

- Group of Member Data Count: The number of "Group of Member Data" components immediately following the DeRegistration Request component.

- Array of Group of Member Data Components: Each "Group of Member Data" component is immediately followed by Group Data Components and its Member Data components (as described in Section 6.1). In this case, where several of these "Group of Member Data" components may be present, the second "Group of Member Data" component only appears after all of the internal components that are referred to by the first "Group of Member Data" component are listed. The format is the same for all subsequent "Group of Member Data" components in the message.

** If Member Count equals zero in the Group of Member Data component, the Group Workload Manager will deregister the entire group.

** Recall that the Group Data Component contains both a Unique LB Identifier field and a Group Name field. If the Group Data component has no Group Name (GroupData's Group Name Length==0), the Group Workload Manager will deregister all groups associated with this load balancer.

7.2.2. DeRegistration Reply

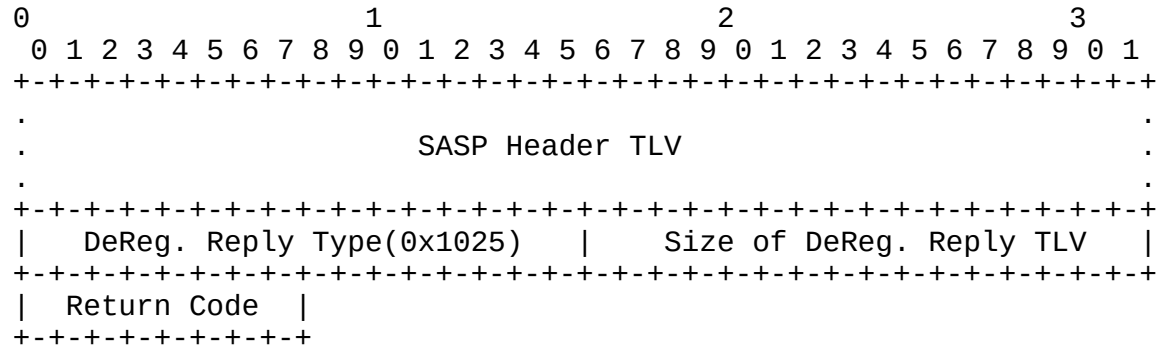


Figure 15

- Return Code: A byte return code indicating the status of action taken.

  A. General SASP return codes (0x00 - 0x3F)

     + 0x00 Successful

     + 0x10 Message not understood

+ 0x11 GWM will not accept this message from the sender. Reasons for this include the following:
  a. The message was not sent by a LB and trust flag is off
  b. LB attempted to address members of a different LB in the message
  c. Vendor specific criteria for this message type were not met.

B. Message-Specific return codes (0x40 - 0xFF)

+ 0x41 Application or System not registered

+ 0x42 Unknown Group Name

+ 0x43 Unknown LB UID

+ 0x44 Duplicate Member in Request

+ 0x46 Duplicate Group in Request (for remove all members/groups requests)

+ 0x51 Invalid LB UID Size (size == 0 or > max)

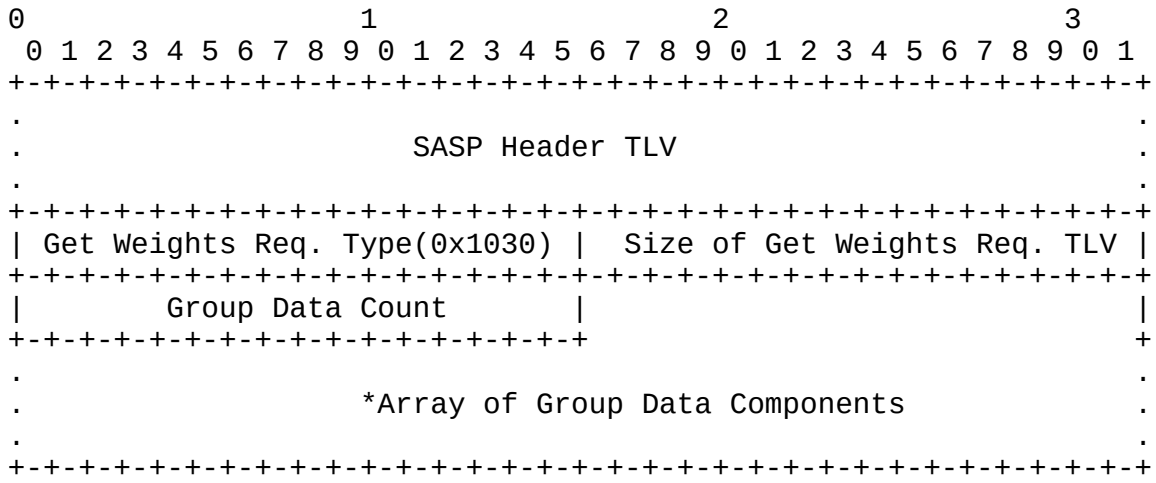
+ 0x61 Member is deregistering itself, but LB hasn't yet contacted the GWM. This deregistration will not be processed.

7.3. Get Weights Request and Reply

This exchange happens between the load balancer/scheduler and the Group Workload Manager to get weights for the groups specified in the list of GroupData objects. In the case of application load balancing (balancing workloads between applications with the same functionality), the load balancer would call the Group Workload Manager every Interval (parameter returned by the Group Workload Manager below) to get an array of weights and associated members (e.g., Application1 20, SecondCopyOfApplication 30, ThirdCopyOfApplication 5). The load balancer then uses these weights to determine the fashion in which work will be sent to each of the members. For example, in the case of weighted round robin, the load balancer/scheduler would then send a request to Application1, the next to SecondCopyOfApplication, and the next to ThirdCopyOfApplication. After 15 requests, the load balancer/scheduler would only send work to Application1 and SecondCopyOfApplication. After an additional 30 requests, the load balancer/scheduler would only send requests to SecondCopyofApplication. After another 10 requests, the load balancer/scheduler product would start over using the weights of 20,

30, and 5 again; or if the Interval number of seconds have passed, the load balancer/scheduler would get a new set of weights.

7.3.1. Get Weights Request



*There will be as many Group Data Components as "Group Data Count" has specified.

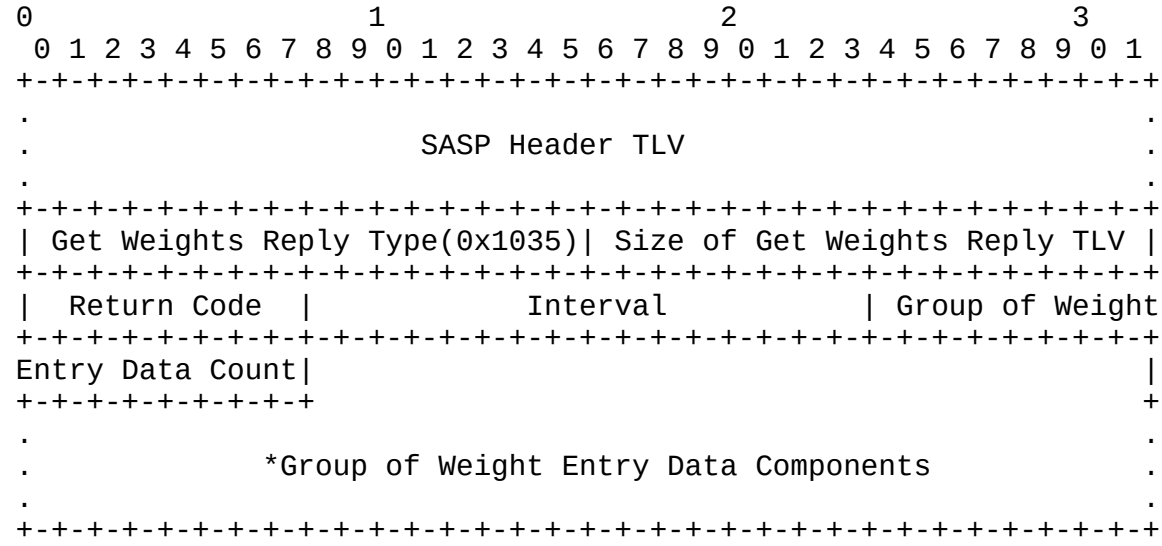
Figure 16

- Group Data Count: The number of "Group Data" components immediately following the Get Weights Request TLV.

- Array of Group Data Components: This array of Group Data Components lists the groups for which the load balancer wants to get weights.

** If there is no group name in the Group Data structure of the Get Weights Request, the load balancer is requesting weights for all groups registered for the load balancer.

7.3.2. Get Weights Reply



* There will be as many Group of Weight Entry Data Components as "Group of Weight Entry Data Count" has specified.

Figure 17

- Return Code: A byte return code indicating the status of action taken.

  A. General SASP return codes (0x00 - 0x3F)

     + 0x00 Successful

     + 0x10 Message not understood

     + 0x11 GWM will not accept this message from the sender. Reasons for this include the following:
       a. LB attempted to address members of a different LB in the message
       b. Vendor specific criteria for this message type were not met.

  B. Message-Specific return codes (0x40 - 0xFF)

     + 0x42 Unknown Group Name

     + 0x43 Unknown LB UID

     + 0x46 Duplicate Group in Request

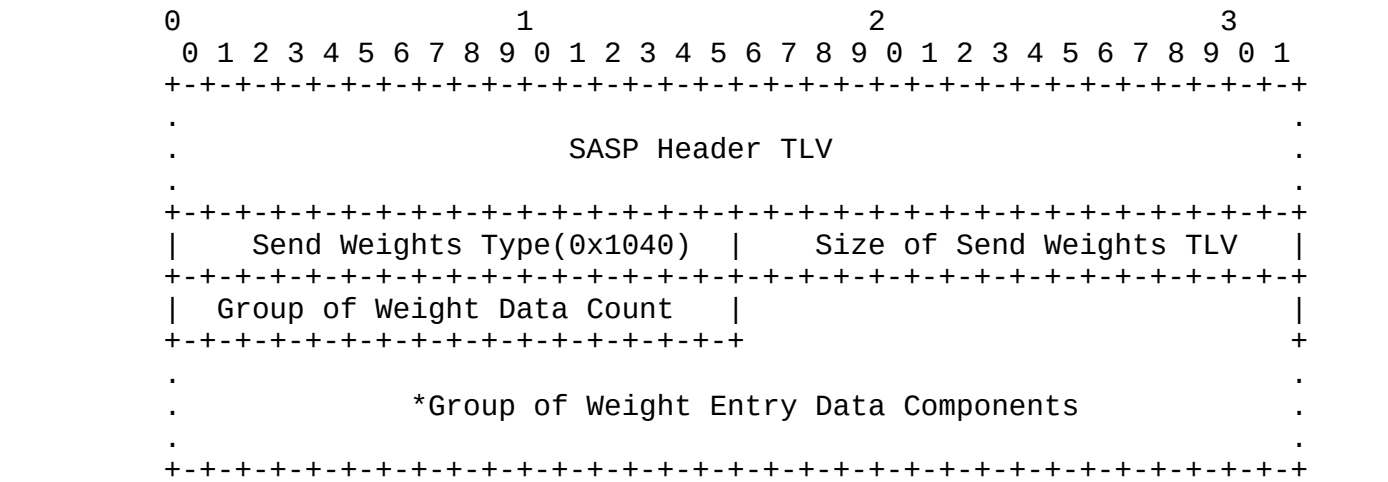
+ 0x51 Invalid LB uid Size (size == 0 or > max)

- Interval: These two bytes indicate a recommended polling interval for the load balancer to use. The Group Workload Manager is stating that any polling interval smaller than the suggested interval would probably retrieve values before they have had a chance to change.

- Group of Weight Entry Data Components: Each "Group of Weight Data" component is immediately followed by Group Data Components and its Weight Entry Data components (as described in Section 6.2). In this case, where several "Group of Weight Data" components may be present, the second "Group of Weight Data" component only appears after all of the internal components that are referred to by the first "Group of Weight Data" component are listed. The format is the same for all subsequent "Group of Weight Data" components in the message.

7.4. Send Weights

This exchange happens between the Group Workload Manager and the load balancer/scheduler to send the new weights for the group specified in Group Name. This message is unique in that it is the only message exchange initiated by the Group Workload Manager and the only message that has no reply. In the case of application load balancing (balancing workloads between applications with the same functionality), the Group Workload Manager would message the load balancer at a possibly dynamic interval (chosen by the Group Workload Manager) to send an array of weights and associated members (e.g., Application1 20, SecondCopyOfApplication 30, ThirdCopyOfApplication 5). The load balancer then uses these weights to determine the fashion in which work will be sent to each of the members. For example, in the case of weighted round robin, the load balancer/scheduler would then send a request to Application1, the next to SecondCopyOfApplication, and the next to ThirdCopyOfApplication. After 15 requests, the load balancer/scheduler would only send work to Application1 and SecondCopyOfApplication. After another 30 requests, the load balancer/scheduler would only send requests to SecondCopyofApplication. After an additional 10 requests, the load balancer/scheduler product would start over using the weights of 20, 30, and 5 again, if it has not yet received a new set of weights. The Group Workload Manager only sends this message if the Push flag has been enabled using a Set Load Balancer State message.



* There will be as many Group of Weight Entry Data Components as "Group of Weight Data Count" has specified.

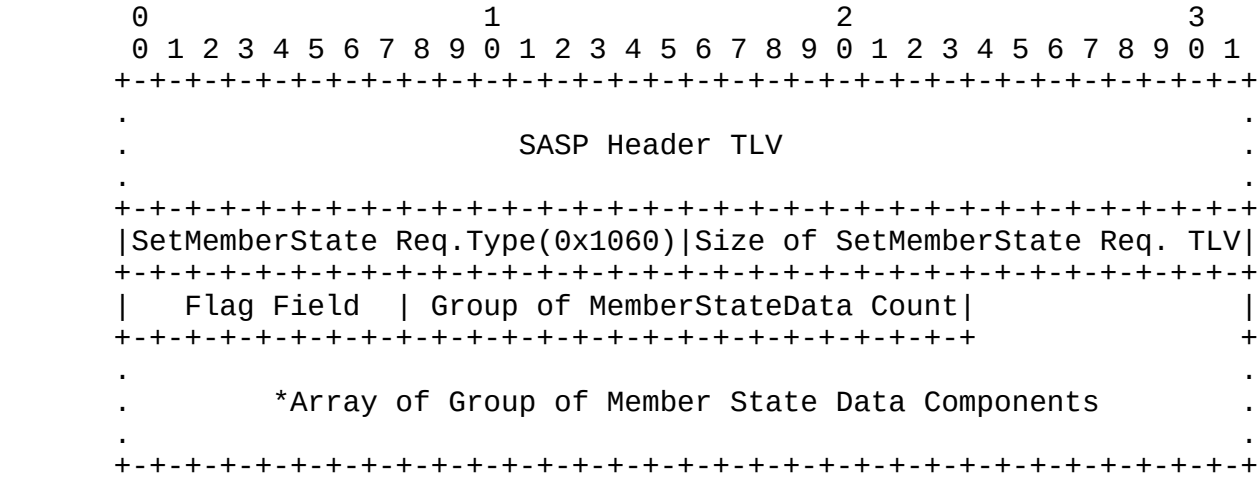
Figure 18

- o Group of Weight Entry Data Components: Each "Group of Weight Data" component is immediately followed by Group Data Components and its Weight Entry Data components (as described in Section 6.2). In this case, where several "Group of Weight Data" components may be present, the second "Group of Weight Data" component only appears after all of the internal components that are referred to by the first "Group of Weight Data" component are listed. The format is the same for all subsequent "Group of Weight Data" components in the message.

7.5. Set Member State Request and Reply

This is a special exchange that can take place between the load balancer and the Group Workload Manager or between the Member and the Group Workload Manager to pass information about the state of the member including placing the member in quiesced or non-quiesced states. In particular, the load balancer/scheduler can use this message to quiesce a set of members. Members can also use this message to quiesce themselves as well as to pass certain state information to the load balancer/scheduler that is opaque to the Group Workload Manager. This opaque state information is passed to the load balancer/scheduler with the weights during get and send weight messages.

7.5.1. Set Member State Request



*There will be as many Group of Member State Data Components as "Group of Member State Data Count" has specified.

Figure 19

- Flag Field

  A. Load Balancer Flag

     + xxxx xxx1 The entity sending this message is the load balancer.

     + xxxx xxx0 The entity sending this message is an Application.

  B. Leftmost seven bits are reserved (0000 000x - 1111 111x).

- Group of Member State Data Count: The number of "Group of Member State Data" components immediately following the Set Member State Request TLV.

- Array of Group of Member Data Components: Each "Group of Member State Data" component is immediately followed by Group Data Components and its Member State Instance components (as described in Section 6.3). In the case where several "Group of Member State Data" components may be present, the second "Group of Member State Data" component only appears after all of the internal components that are referred to by the first "Group of Member State Data" component are listed. The format is the same for all subsequent "Group of Member State Data" components in the message.

7.5.2. Set Member State Reply

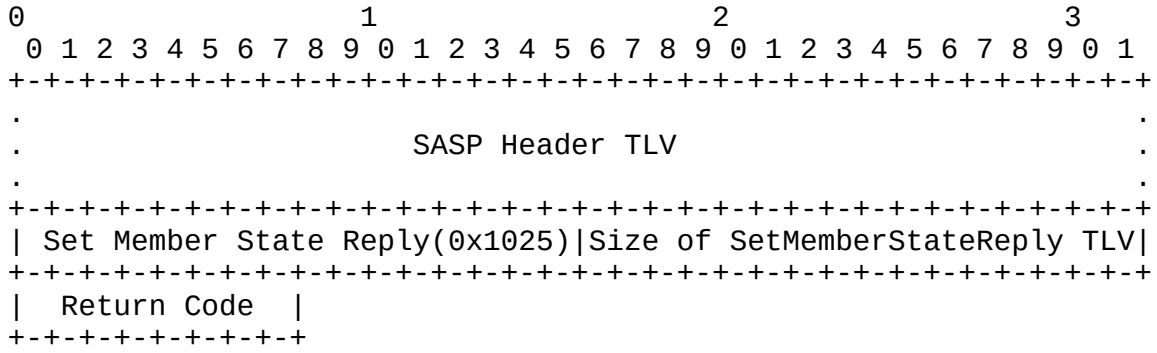


Figure 20

- o Return Code: A byte return code indicating the status of action taken.

  A. General SASP return codes (0x00 - 0x3F)

  + 0x00 Successful

  + 0x10 Message not understood

  + 0x11 GWM will not accept this message from the sender. Reasons for this include the following:
    a. The message was not sent by a LB and trust flag is off
    b. LB attempted to address members of a different LB in the message
    c. Vendor specific criteria for this message type were not met.

  B. Message-Specific return codes (0x40 - 0xFF)

  + 0x41 Application or System not registered

  + 0x42 Unknown Group Name

  + 0x43 Unknown LB UID

  + 0x44 Duplicate Member in Request

  + 0x46 Duplicate Group in Request

  + 0x50 Invalid Group Name Size (size == 0)

  + 0x51 Invalid LB UID Size (size == 0 or > than max)

+ 0x61 Member is setting state for itself, but LB hasn't yet contacted the GWM. This request will not be processed.

7.6. Set Load Balancer State Request and Reply

This is an exchange that can take place between the load balancer and the Group Workload Manager to pass information about the state (and partial configuration) of the load balancer.

7.6.1. Set LB State Request

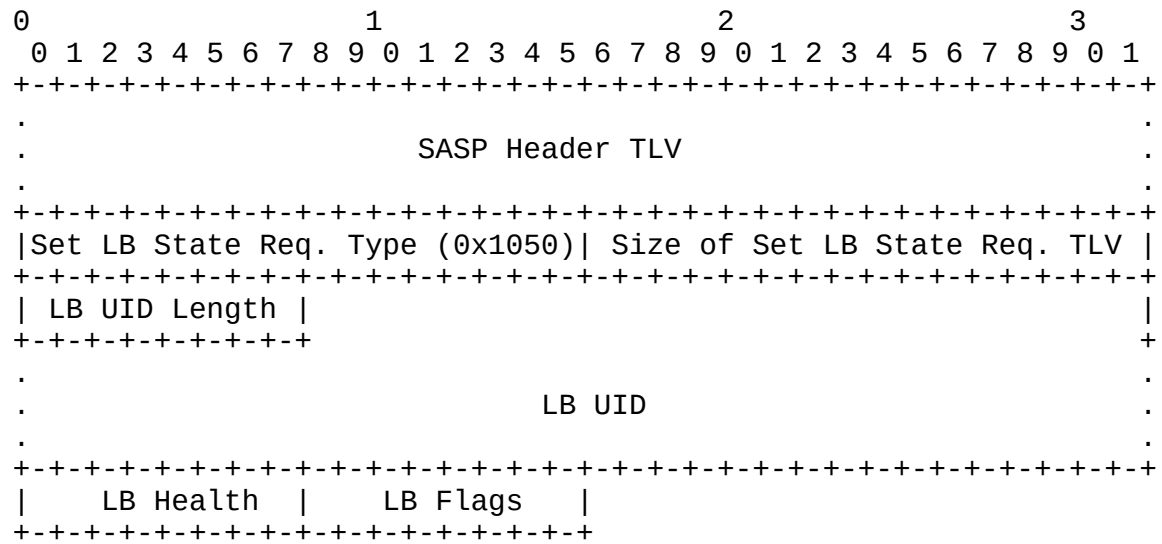


Figure 21

- LB UID Length: one-byte length field describing the size of the following LB UID.
- LB UID: This should be the same unique identifier given when registering group members for this particular load balancer.
- LB Health: This field gives the load balancer a chance to pass in a metric describing its own health or state.

  0x00 - 0x7F Least Healthy - Most Healthy

  0x80 - 0xFF Reserved

- LB Flags:

  A. Push Flag

+ xxxx xxx1 The load balancer should receive weights through the Send Weights message (GWM pushes weights to load balancer). Even if this flag is set, the GWM must still respond accordingly to any Get Weights messages from the load balancer.

+ xxxx xxx0 The load balancer will send a Get Weights message to get the new weights. This is the default behavior. (load balancer pulls weights from GWM).

B. Trust Flag

+ xxxx xx1x Trust any member-initiated registration, deregistration, or set state message. Immediately reflect the registration, deregistration, or new state in the weights sent.

+ xxxx xx0x Do not trust any member-initiated registration, deregistration, or set state message. Registration, Deregistration, and State Setting of members can only occur from the load balancer. Discard any member-initiated registration, deregistration, or set state message. This is the default behavior.

C. No Change / No Send Flag

+ xxxx x1xx The GWM must not include members whose weights and state (i.e., contact and quiesce flags) have not changed since they were last sent.

+ xxxx x0xx The GWM must include the weights of all group members when sending the weights to this load balancer (including members whose weights and state have not changed). This is the default behavior.

D. Leftmost five bits are reserved (0000 0xxx - 1111 1xxx).

7.6.2. Set LB State Reply

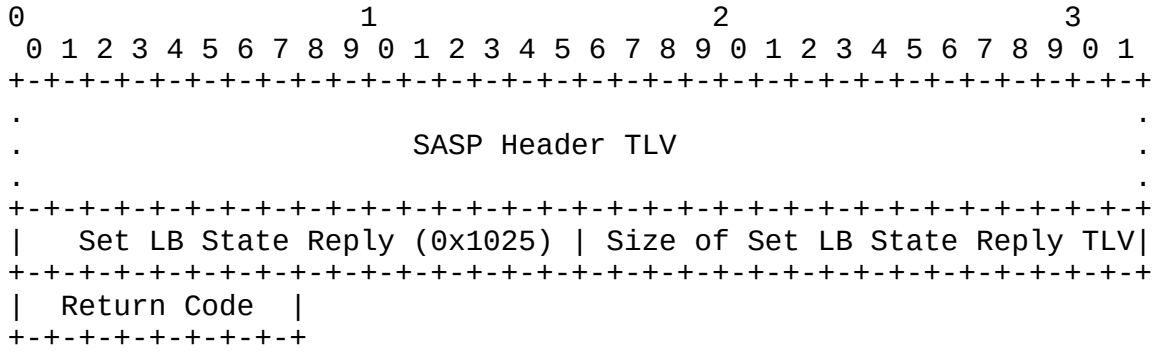


Figure 22

- o Return Code: A byte return code indicating the status of action taken.

  A. General SASP return codes (0x00 - 0x3F)

  + 0x00 Successful

  + 0x10 Message not understood

  + 0x11 GWM will not accept this message from the sender. Reasons for this include the following:
    a. LB attempted to address the state of a different LB
    b. Vendor specific criteria for this message type were not met.

  B. Message-Specific return codes (0x40 - 0xFF)

  + 0x51 Invalid LB UID Size (size == 0 or > max)

8. Example of SASP Message Encoding

This section provides an example of the actual SASP message encoding. For this example, we will look at a sample GetWeights Reply in which two webservers are registered to a serverfarm called FARM1. The IP addresses of the two webservers are 10.10.10.1 and 10.10.10.2. Currently the GWM has a weight of 40 for 10.10.10.1 and 20 for 10.10.10.2. The load balancer has a unique Identifier of "LB1" and the message example was sent by the GWM in response to a request (MessageID: 0x32000000) for FARM1’s weights.

The TLVs necessary for this message are shown in the following list.

1. SASP Header TLV

	Field	Size	Value
T	Type	2 bytes	0x2010
L	Length	2 bytes	0x000D
	Version	1 byte	0x01
V	Mesg Len	4 bytes	0x0000 006A
	Mesg ID	4 bytes	0x3200 0000

Figure 23

2. Get Weights Reply TLV

	Field	Size	Value
T	Type	2 bytes	0x1035
L	Length	2 bytes	0x0009
	RetCode	1 byte	0x00
V	Interval	2 bytes	0x0040
	GWD Count	2 bytes	0x0001

*GWD Count = Group of Weight Data Count

Figure 24

3. Group of Weight Data TLV

	Field	Size	Value
T	Type	2 bytes	0x4011
L	Length	2 bytes	0x0006
V	WE Count	2 bytes	0x0002

*WE Count = Weight Entry Count

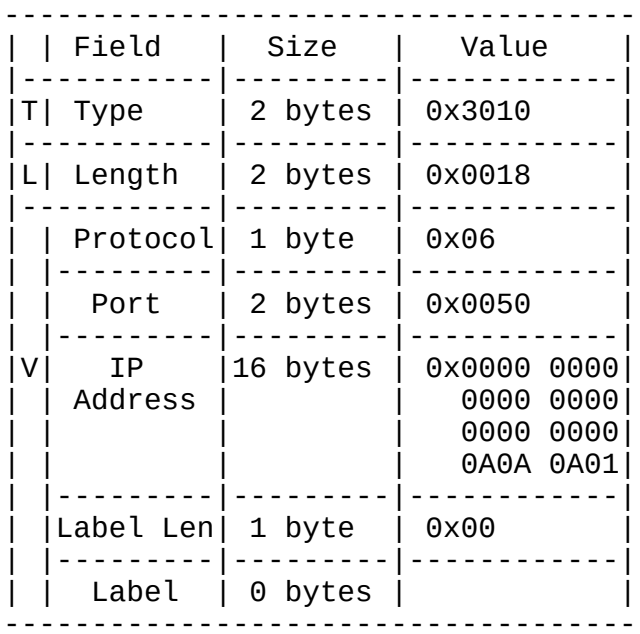
Figure 25

4. Group Data TLV

	Field	Size	Value
T	Type	2 bytes	0x3011
L	Length	2 bytes	0x000E
	LBUID len	1 byte	0x03
	LBUID	3 bytes	"LB1" or 0x4C 42 31
V	GroupName Length	1 byte	0x05
	Group Name	5 bytes	"FARM1" or 0x46 41 52 4D 31

Figure 26

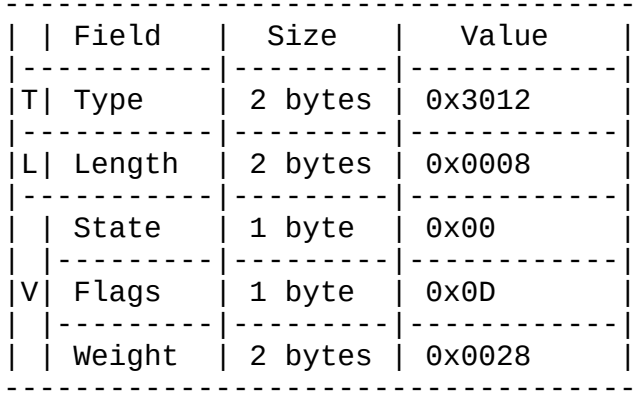
5. Member Data TLV



	Field	Size	Value
T	Type	2 bytes	0x3010
L	Length	2 bytes	0x0018
	Protocol	1 byte	0x06
	Port	2 bytes	0x0050
V	IP Address	16 bytes	0x0000 0000 0000 0000 0000 0000 0A0A 0A01
	Label Len	1 byte	0x00
	Label	0 bytes	

Figure 27

6. Weight Entry Data TLV



	Field	Size	Value
T	Type	2 bytes	0x3012
L	Length	2 bytes	0x0008
	State	1 byte	0x00
V	Flags	1 byte	0x0D
	Weight	2 bytes	0x0028

Figure 28

7. Member Data TLV

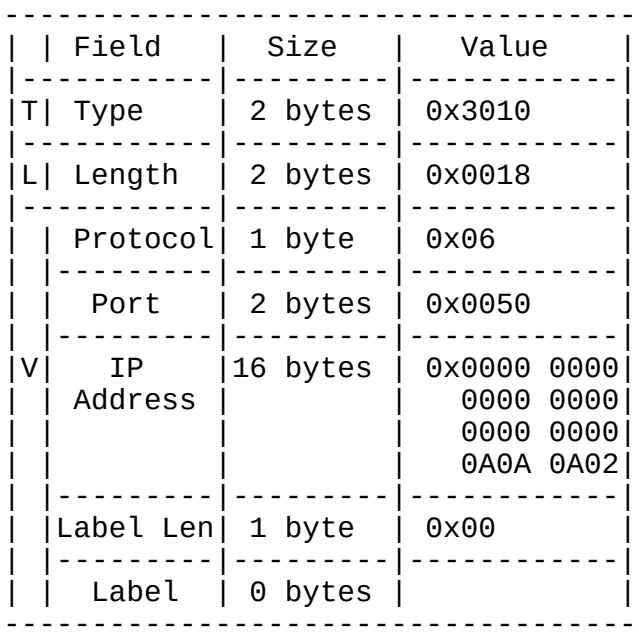


Figure 29

8. Weight Entry Data TLV

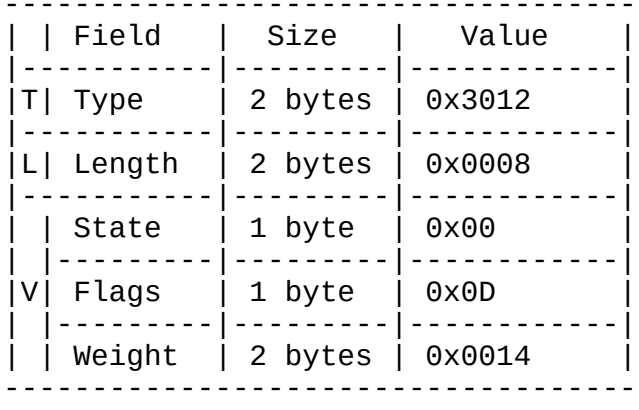


Figure 30

A hex stream representing this same message is below:

20 10 00 0D 01 00 00 00 6A 32 00 00 00 10 35 00 09 00 00 40

00 01 40 11 00 06 00 02 30 11 00 0E 03 4C 42 31 05 46 41 52

```
4D 31 30 10 00 18 06 00 50 00 00 00 00 00 00 00 00 00 00 00

00 0A 0A 0A 01 00 30 12 00 08 00 0D 00 28 30 10 00 18 06 00

50 00 00 00 00 00 00 00 00 00 00 00 00 0A 0A 0A 02 00 30 12

00 08 00 0D 00 14
```

(106 bytes)

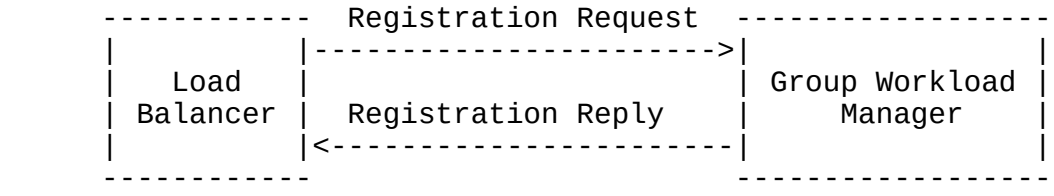
9. Protocol Flow

This section describes the expected general flow of the SASP messages.

9.1. Normal Protocol Flow

SASP first starts with a connection from an LB to the GWM. This is expected to be a long-running connection and will be used for many messages. After establishing the connection, the LB either registers a group of members or sets a Trust flag to allow the members to register themselves. The Trust flag is set using a Set LB State Request (both message flows are shown below).

Registration from load balancer



Set LB State from load balancer

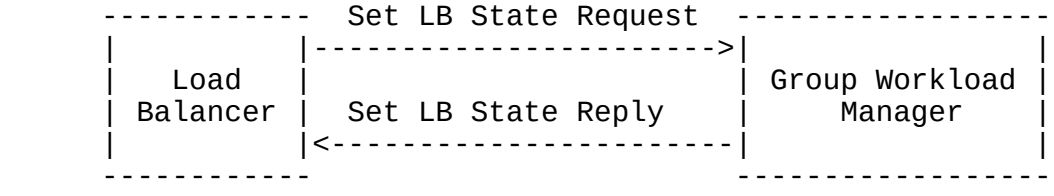


Figure 31

The connection can start with other requests, but any other request would likely result in an error (unless this connection is a reconnection that has happened a short period of time after the original connection). For example, if the load balancer issues a deregistration request as its first message, it will receive an error because it has not registered any groups.

The load balancer always drops all state information after a loss of connection and can recover it using a GetWeights message. The establishment of a new connection causes the GWM to assume that the old one is broken. In this case, the GWM will keep all state for the load balancer for a limited time after a detected break. After the limited time has expired, all state for the broken connection will be discarded by the GWM.

Registration of group members may be done at any time. A load balancer can register anywhere from one group with one member to many groups of many members. The member may also register itself if the Trust flag has been set and it knows the appropriate load balancer information. Registrations will add to groups that already exist, but return errors if any of the registered members already exist.

In the case of system load balancing, the representation of a member is only the member's IP address with a 0 used as the value for the port and protocol. In the case of application load balancing, the representation of a member is the member's IP address and the Application's port and protocol.

Deregistration of group members may be done at any time. A load balancer can deregister anywhere from one group with one member to many groups of many members. The LB may also deregister entire groups or deregister all of its groups at once. The member may also deregister itself if the Trust flag has been set and it knows the appropriate load balancer information.

Once members are registered, the GWM will start the monitoring and weight computation processes to determine weights to be sent back to the load balancer. At any time the load balancer may issue a GetWeights message and ask for the weights for members in a particular group. The LB may also set a flag telling the GWM to send the weights without waiting for the GetWeights message. If this flag is set, the GWM will send the weights at an interval it feels is appropriate (the interval could change depending on the algorithm used and variance of the weights generated).

At any time the LB or a particular member may quiesce the member through the use of a SetMemberState message. In this case, the member's weight will always be zero, and the quiesce flag will be

turned on when sending its weight. Members may also use this message to send an opaque state value that will also be presented when sending weights.

At any time, the load balancer may choose to send the GWM a SetLBState request to configure its interaction. The message allows the load balancer to set the Push, Trust, and NoChange_NoSend flags. It also allows the load balancer to pass a health value to the GWM to be displayed.

9.2. Behavior in Error Cases

While behaviors in many error conditions will be product specific, the following error cases should have the following expected behavior.

Case: The protocol is violated in an unrecoverable manner by either end of the connection.

Behavior: Either end of the connection may choose to disconnect to avoid future message synchronization problems. The state kept when disconnected is vendor specific.

Case: LB or application attempts to connect to the GWM before the GWM is fully up and running.

Behavior: The LB or application should wait at least 20 seconds to retry the connection.

Case: Members attempt to register or deregister themselves before the LB develops the connection with the GWM.

Behavior: In this case, the members would receive a reply with an error code signifying that there is no LB registered with that LB UID.

Case: Member registers or deregisters for an LB who has not set the Trust flag.

Behavior: GWM will send Member a reply containing an error code.

Case: LB asks for weights for a group that doesn’t exist.

Behavior: GWM will send LB a reply containing an error code.

Case: LB or Member attempts to register a member that is already registered in that group.

Behavior: GWM will send sender a reply containing an error code.

Case: LB or Member attempts to deregister a member or group that doesn’t exist.

Behavior: GWM will send sender a reply containing an error code.

Case: LB or Member tries to set state for a non-registered server.

Behavior: GWM will send sender a reply containing an error code.

Case: LB tries to Get Weights for an unregistered group.

Behavior: GWM will send LB a reply containing an error code.

9.3. Example Flow 1: Load Balancer Registration, Getting Weights, and Application-Side Quiescing

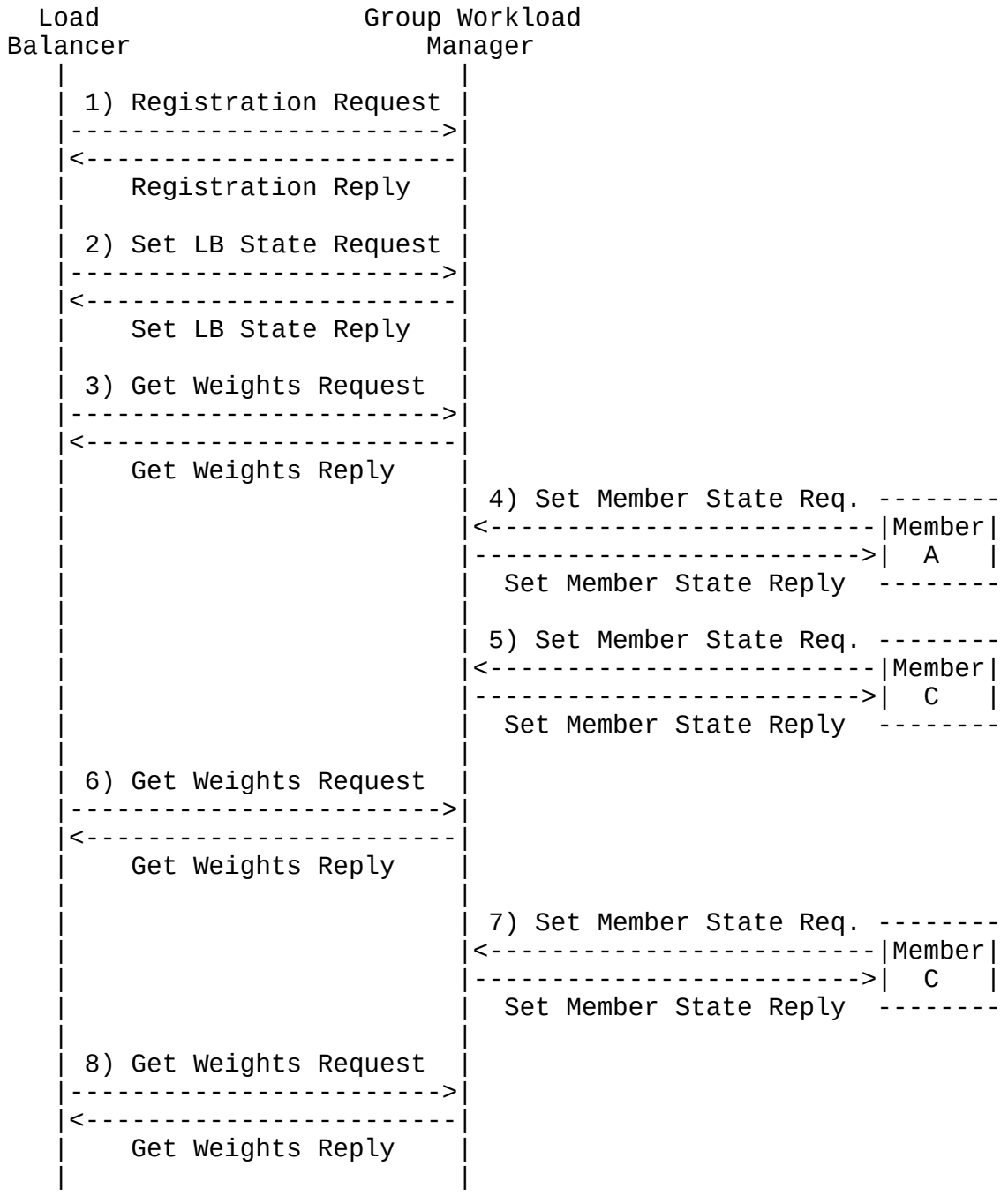


Figure 32

1. The LB registers Members A, B, and C in a group named GRP1. The GWM replies with no error.

2. The LB turns its trust flag on by issuing a Set LB State message:

   LB Health: 0x00 Flags: 0000 0010

3. The LB sends a Get Weights message for GRP1 and gets the reply:

Members	Opaque State	Flags	Weight
Member A	0x00	0000 1101	20
Member B	0x00	0000 1101	40
Member C	0x00	0000 1101	5

4. Member A sends a Set Member State message with flags:

Members	Opaque State	Flags
Member A	0x32	0000 0000

5. Member C sends a Set Member State message to quiesce itself with the following flags:

Members	Opaque State	Flags
Member C	0x0A	0000 0001

6. The LB sends the Get Weights message for GRP1 and receives the following:

Members	Opaque State	Flags	Weight
Member A	0x32	0000 1101	20
Member B	0x00	0000 1101	40
Member C	0x0A	0000 1111	5

7. Member C sends a Set Member State message to resume (un-quiesce itself) with the following flags:

Members	Opaque State	Flags
Member C	0x0A	0000 0000

8. The LB sends a Get Weights message for GRP1 and gets the reply:

Members	Opaque State	Flags	Weight
Member A	0x32	0000 1101	20
Member B	0x00	0000 1101	40
Member C	0x0A	0000 1101	5

9.4. Example Flow 2: Set Load Balancer State, Application Registration, and Load Balancer Group DeRegistration

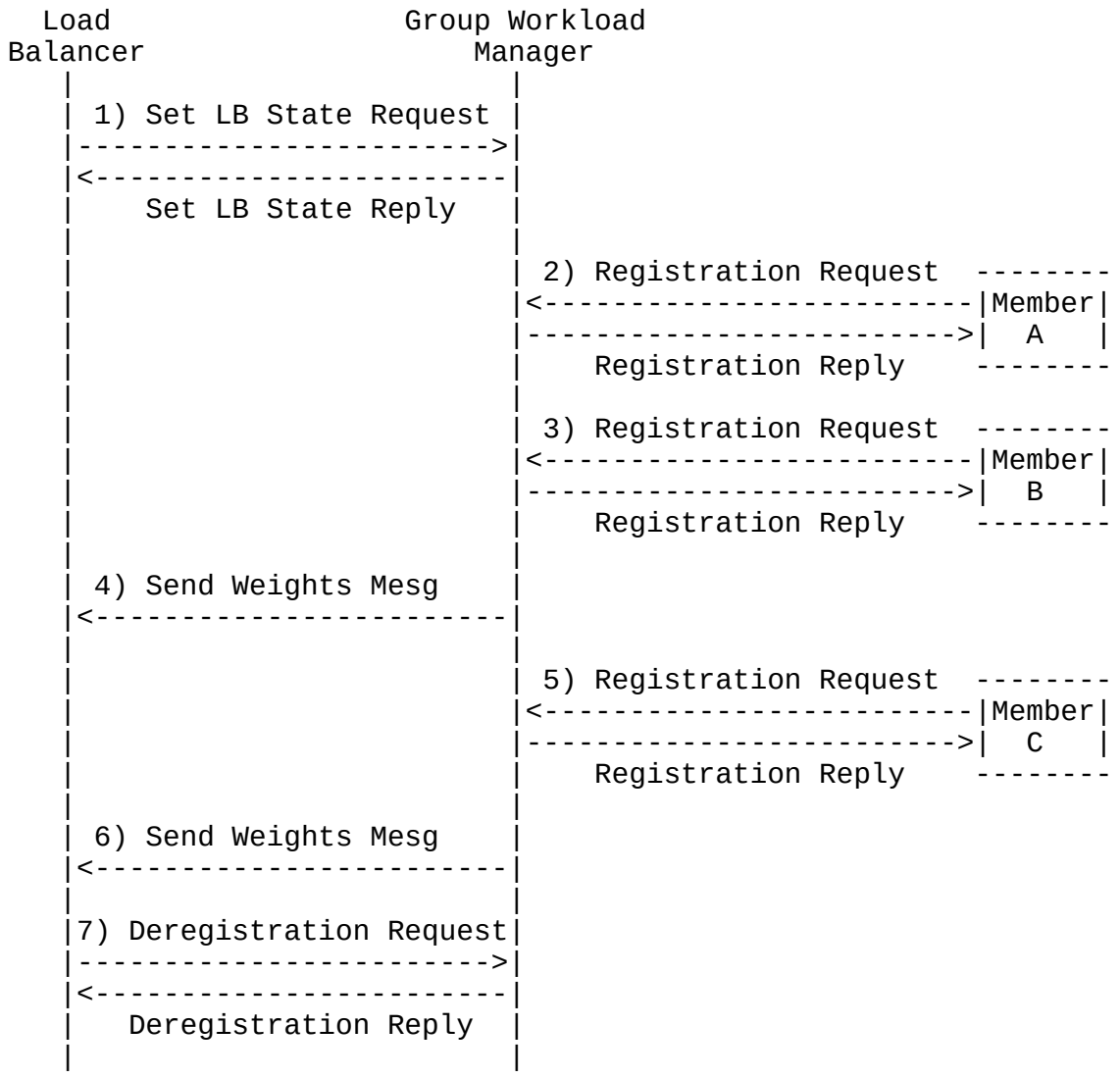


Figure 39

1. The LB sets its state with the Set LB State message and the following parameters.

   Health: 0x7F Flags: 0000 0011

2. Member A registers itself for work in GRP1 using the Register message.

3. Member B registers itself for work in GRP1 using the Register message.

4. The GWM issues a Send Weights message to the LB.

Members	Opaque State	Flags	Weight
Member A	0x00	0000 1001	20
Member B	0x00	0000 1001	40

5. Member C registers itself for work in GRP1 using the Register message.

6. The GWM issues a Send Weights message to the LB.

Members	Opaque State	Flags	Weight
Member A	0x00	0000 1001	20
Member B	0x00	0000 1001	40
Member C	0x00	0000 1001	5

7. LB deregisters GRP1 by using the DeRegister message with the Member Data Count = 0

9.5. Avoiding Single Points of Failure

- To avoid having a single point of failure at the load balancer, an administrator may choose to have multiple load balancers in his or her environment. SASP provides for the GWM to keep track of multiple load balancers through the use of load balancer unique identifiers (LB UIDs).

- To avoid having a single point of failure at the GWM or enhance the load balancing strategy by utilizing the strengths of several different GWMs, an administrator may choose to have multiple GWMs in his or her environment. In this case, the load balancer would

connect to multiple GWMs and register the same groups with corresponding members. The load balancer may choose to coordinate the recommendations of each GWM by any method it chooses (e.g., statistical combination such as averaging). The coordination of weights from multiple GWMs is product specific and not addressed in this protocol.

10. Security Considerations

SASP is a binary stream expected to be transported over a TCP connection. To secure this protocol, it is expected that implementers of the protocol use a secure mode of transport such as SSL/TLS. Discussions around security concerns have been listed below:

Security Issue: In insecure environments, if the LB UID becomes known by another system, the other system could initiate a connection and send messages to the GWM causing the GWM to replace the previous (possibly valid) connection for the new (potentially bad) connection.

Solution: This may not be a concern if the load balancer and GWM are in protected parts of the network. If the administrator is concerned about this vulnerability, she should use SSL or TLS to provide authentication for the connection. When using SSL or TLS to secure the connection, the administrator SHOULD use both server and client authentication through client and server certificates. The GWM will trust any certificate that is signed by an authority it’s been configured to trust.

Security Issue: In insecure environments, if the load balancer turns the Trust Flag on, any member or other system can send a Registration Message and be included in the serverfarm to receive work. A person with bad intentions and the correct information could exploit this feature and register his own application to receive work. His counterfeit application could capture valuable data from unsuspecting clients as their transactions are sent to his system.

Solution: This may not be a concern if the GWM and its members are in protected parts of the network. If the administrator is concerned about this vulnerability, she should use SSL or TLS to provide authentication for the member connections. When using SSL or TLS to authenticate the connection, the administrator would need to explicitly install valid certificates on each component

while at the same time establishing the trusted certificates of each component. This would make certain that only those trusted components would be permitted to connect to the GWM.

11. Normative References

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

Appendix A. Acknowledgements

The author gratefully acknowledges contributions by Mark Albert, David McCowan, John Fenton, Derek Huckaby, Dyan Collins, and Stefano Testa. Mark Albert, David McCowan, John Fenton, Derek Huckaby, Dyan Collins, and Stefano Testa were supported for this work by Cisco Systems Inc.

The author would also like to thank John Arwe, Dave Bostjancic, Brian Carpenter, Donna Dillenberger, Gus Kassimis, and Thomas Narten for their efforts in the creation and refining of this work.

Author's Address

Alan Bivens
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
US

EMail: jbivens@us.ibm.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).