Session Initiation Protocol (SIP)
Internet Draft
Document: draft-stucker-sip-guid-00.txt
Expires: July 2004

B. Stucker
Nortel Networks

February 2004

Client Globally Unique ID for SIP

Status of this Memo

Abstract

A number of applications using the Session Initiation Protocol (SIP) protocol require or can be enhanced by being able to uniquely identify a particular user agent (UA) instance in the network. This document describes an extension to SIP that allows clients to generate globally unique identifiers (GUID) for use within SIP based applications, providing an example of their use.

Table of Contents

1. Introduction

Within SIP, there arise situations where it is necessary to ensure that an action is applied to a particular user agent (UA) instance, but the existing mechanisms within SIP are not always reliable. For example, although registrations identify a routable address and port of a particular UA, in an environment that uses dynamically assigned IP addresses (NATs, VPNs, short-lease DHCP networks) there is no ready way of always tying registrations together across time for a particular UA instance. In these environments, the usual IP/port combination that defines a particular routing location of a UA is unreliable over time as an identifier of that UA.

As a result, an identifier that UAs can use as a "finger-printing" mechanism to identify themselves is useful. Whereas the Globally Routable UA URIs (GRUU) draft [4] seeks to address a server-generated identifier for the UA, this draft seeks to define a client-generated approach to a similar problem.

The mechanism defined in this document allows a particular UA instance to construct a globally unique identifier which can be used by SIP services to process requests that require, or are enhanced by,

the ability to identify a particular UA instance in the network over a long period of time.

2. Terminology

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 RFC-2119 [ii].

3. Creating a GUID

This section covers the details of creating a GUID on the client UA.

3.1 Characteristics of a GUID

The idea of a globally unique ID is hardly a new concept. Designers and developers of all sorts of applications in the physical world and the Internet have required the ability to uniquely identify a particular entity from a larger set. This is especially true when every other property of the entity is subject to substantial changes over time that would render it difficult or impossible to uniquely identify over time.

For example, governments frequently assign us a number (or other identifying string) when we are born because they have a need to identify us as taxpayers throughout our lives. There are several other examples of unique IDs, such as vehicle identification numbers and serial numbers on items we buy from large manufacturers.

A common characteristic of these identification numbers is that they have two basic properties:

- They are unique to the entity they are associated with.

- A central authority coordinates the assignment of IDs to ensure that no two entities are given the same identifier.

Note, that there is no requirement that there be any sort of registry

that knows which entity has what identifier. This would be needed if the identifier were to be used for non-repudiation purposes, but that is not always a goal that needs to be fulfilled depending on the application.

Sometimes entities need to be able to be identified uniquely, but to have a central authority assign an identifier would be difficult or impossible. In these situations, it is still possible for the entity to assign itself a unique identifier. This can be achieved by using a mechanism that ensures that the odds of any two entities having the same identifier are statistically insignificant.

An example of this mechanism would be human fingerprints. Fingerprints can be used as a globally unique identifier of who you

are, and the odds of two people having the same fingerprints are statistically insignificant (even twins have a different set). No central authority coordinates the assignment of who gets what fingerprints, and yet they can be used to uniquely identify a particular person. If they are registered with a central authority, they can be used for purposes of non-repudiation. In either case, they are very useful, as other characteristics of people may change wildly over time.

3.2 Construction of a SIP GUID

Constructing an identifier that describes a UA is trivialquite straightforward. SIP TAGs are frequently generated to identify a particular UA session within SIP. Ensuring that the identifier is unique within a small, controlled set of UAs is more difficult, but still manageable by simply assigning them directly to the UA upon creation (like assigning a static IP address to a machine on a LAN). However, making that identifier unique across very large sets could be very difficult by simple assignment through sheer logistics (think about your experiences trying to get a driver's license).

Because a straightforward assignment of a GUID is problematic at best (and impossible at worst) this approach is ruled out in favor of using a standard mechanism: use time and space to your advantage. All SIP GUIDs MUST be generated based off the time that they were generated, and the "space" in which they were generated.

Obviously, generating a SIP GUID that is composed of a three-digit

number would not satisfy most reasonable definitions of "unique" within a SIP network. Therefore, SIP GUIDs MUST be at least 128-bits in length.

3.2.1 Time Component

Time can be used to create uniqueness because each instant in time only occurs once. This can be used to constrain the set of all UAs that wish to create a GUID at that instant from the set of all UAs that will ever exist (ie. all of the UAs that wish to create a GUID on February 6th, 2004 at 10:45pm as opposed to all UAs that will ever exist from now to eternity). This means that a component of a GUID should be based on the current local time. It is not necessary that every UA generating a GUID need to have synchronized clocks with every other UA. This is because we're not interested in being able to tell the exact moment a GUID is created. It's used simply as a component of the GUID in order to constrain the larger set.

Many computers and development platforms vary in the scale at which time can be measured. Because we are using time to constrain the set of all UAs that may ever wish to generate a GUID, it is important

that the smallest available unit be used by the UA generating the GUID. Additionally, a large random number from a cryptographically-strong random number generator can be appended to the current time to create a pseudo-timestamp with very fine resolution.

Here's an example:

- A computer's clock can be resolved down to 1 millisecond.
- The computer's random number generator can produce a random integer up to (2^31)-1.
- From this a "pseudo-clock" can then be constructed that resolves time down to the order of a pico-second (10^-12 seconds, or trillionths of a second).

Friday, February 6th, 2004 at 21:30:54 CST can be expressed as 1076124654957 milliseconds since January 1, 1970, 00:00:00 GMT.

A possible random number generated by a cryptographically-strong random number generator: 190182543.

Taken together, it is possible to create a "pseudo-time" of 1076124654957190182543 pico-seconds since January 1, 1970 00:00:00 GMT.

This is a very powerful notion, and if further resolution is required, successive random numbers can be appended to further resolve the "pseudo-clock" to fantastically small instants of time. It is critical, however, that an actual clock source be used as the most-significant digits of the "pseudo-clock".

In the example given, even if 1 billion SIP UAs decided to generate a new GUID at the same time, it is still a 1 in a trillion chance that they come up with the same "pseudo-clock" time.

SIP GUIDs MUST use a "psuedo-clock" that resolves to a minimum of 10^-12 seconds.

3.2.2 Space Component

The other component to a well-formed globally unique identifier that is not assigned by a central authority is to use space (or an approximation of it) as a component. It can obviously be the same time in multiple places, but no two UAs can ultimately occupy the same position in "space".

Because we are dealing with the electronic world, the notion of space is used somewhat conceptually; depending on the application, what

constitutes "space" may vary. The MAC address of the device that the UA instance runs upon would be a good way to denote its position in space, where space is given as the network. However, there are security implications involved with handing out a MAC address at the application level. For one, it can be used to discover the manufacturer of the device, which may help an attacker determine a method of attack.

Therefore, MAC addresses SHOULD NOT be used as an identifier of space for the purposes of a SIP GUID.

Additionally, there may be multiple UA instances executing on the same CPU. For this reason, it is RECOMMENDED that the space component of a SIP GUID be a location in memory that is uniquely held by that

UA instance, as well as the IP address of the UA. Taken together with the time component, this still provides a high level of uniqueness within the network. It is extremely unlikely that two UA instances would be stored in the same location in memory, on two computers with the exact same IP address, at the exact same "pseudo-clock" time.

SIP GUIDs MUST contain a space component that provides no fewer than 64 bits of uniqueness.

3.3 Comparing SIP GUIDs

When comparing two SIP GUIDs, their values SHOULD be considered a unique identifier for the UA instance associated with the party that sent the SIP request, including any aliases of the user or entity identified by the sending party.

3.4 The GUID Header

The GUID header identifies a UA GUID. This header denotes the GUID for that UA instance. The GUID header MUST NOT appear in a SIP response, and if present MUST be ignored by the recipient. The GUID header MAY appear in any SIP request type. It is RECOMMENDED that user agents include their GUID in any REGISTER request sent.

3.4.1 Syntax

This document adds the following entry to Table 2 of [1]. Additions to this table are also provided for extension methods defined at the time of publication of this document. This is provided as a courtesy to the reader and is not normative in any way. MESSAGE, SUBSCRIBE and NOTIFY, REFER, INFO, UPDATE, PRACK, and PUBLISH are defined respectively in [6], [7], [5], [8], [9], [10], and [11].

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG	MSG
GUID	R		o	o	o	o	o	o	o

	SUB	NOT	REF	INF	UPD	PRA	PUB

			---	---	---	---	---	---	---
GUID	R	o	o	o	o	o	o	o

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [3].

GUID = "GUID" HCOLON token

A SIP request MUST contain no more than one GUID.

Examples:

GUID: f7ca930e2412f1bf016eb4940441672d3c26b17

GUID: 10761246549571901825 43+47bfc83e+10.33.15.8

3.5 Proxy Behavior

Proxies MUST NOT modify the contents of the GUID header during processing. It MAY be stripped according to the privacy policies of the system should header privacy have been requested by the UA sending the request in accordance with RFC-3323.

4. Security Considerations

The extension defined in this document may impact the security of a particular SIP application. Depending on the use of the GUID in a given application, considerations may need to be made to use a secure transport mechanism such as TLS for sending SIP requests containing a GUID.

5. IANA Considerations

5.1 Registration of the "GUID" SIP header

Name of Header: GUID

Short form: none

Normative description: section 3.4 of this document.

6. References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R. Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

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

[3] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[4] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", draft-ietf-sip-gruu-00 (work in progress), January 2004.

[5] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[6] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C. and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[7] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[8] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.

[9] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[10] Rosenberg, J. and Schulzrinne, H., "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 2362, June 2002.

[11] Niemi, A., "SIMPLE Presence Publication Mechanism", draft-ietf-sip-publish-02 (work in progress), January 2004.

7. Acknowledgements

Thanks to Jennifer Beckman, Mary Barnes, Alberto Villarica, Trip Ingle, and William Janning for comments and helping to create the foundation for this document. Additionally, thanks is given to Jonathan Rosenberg for introduction of the GRUU draft which describes the server-side generation of unique identifiers within SIP.

Author's Addresses

Brian Stucker
Nortel Networks
2380 Performance Drive
Richardson, Texas
Email: bstucker@nortelnetworks.com