Network Working Group
Internet Draft
Intended status: Standards Track
Expires: July 30, 2012

P. Jones
C. Pearce
J. Polk
G. Salgueiro
Cisco Systems

January 30, 2012

End-to-End Session Identification in IP-Based Multimedia Communication Networks

draft-jones-ipmc-session-id-03.txt

Abstract

This document describes an end-to-end Session Identifier for use in IP-based Multimedia Communication systems that enables endpoints, intermediate devices, and management systems to identify a session end-to-end, associate multiple endpoints with a given multipoint conference, track communication sessions when they are redirected, and associate one or more media flows with a given communication session.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

IP-based multimedia communication systems like SIP [1] and H.323 [2] have the concept of a "call identifier" that is globally unique. The identifier is intended to represents an end-to-end communication session from the originating device to the terminating device. Such an identifier is useful for troubleshooting, billing, session tracking, and so forth.

Unfortunately, there are a number of factors that contribute to the fact that the current call identifiers defined in SIP and H.323 are not suitable for end-to-end session identification. Perhaps most significant is the fact that the syntax for the call identifier in SIP and H.323 is different between the two protocols. This important fact makes it impossible for call identifiers to be exchanged end-to-end when a network utilizes one or more session protocols.

Another reason why the current call identifiers are not suitable to identify the session end-to-end is that in real-world deployments devices like session border controllers often change the values as the session signaling passes through. This is true even when a single session protocol is employed and not a byproduct of protocol interworking.

This draft presents a new identifier, referred to as the Session Identifier or “Session ID”, and associated syntax intended to overcome the issues that exist with the currently defined call identifiers. The proposal in this document attempts to comply with the requirements specified in [5].

2. Conventions used in this document

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 [3].

3. Session Identifier Requirements

Requirements for the end-to-end Session Identifier can be found in a separate memo titled “Requirements for an End-to-End Session Identification in IP-Based Multimedia Communication Networks” [5].

4. Session Identifier Usage

The Session Identifier is intended to uniquely identify a communication session end-to-end. This document does not specify how the Session Identifier is to be used, but merely defines the identifier in such a way as to enable it to be used for situations encountered in real-world deployments of IP-based multimedia communication systems, including:

* End-to-end identification of a communication session

* Association of session signaling and media flows, made possible by including the session identifier in media-related messages (e.g., RSVP [7] or RTCP [8])

* Identification of devices taking part in the same multipoint conference

* Tracking sessions transferred from one endpoint to another

* Identification of recorded sessions

* Logging for the purposes of accounting, billing, debugging, etc.

5. Constructing the Session Identifier

The Session Identifier is comprised of two UUIDs [4] that are concatenated together, with each UUID created by the endpoints participating in the session. The first endpoint in the session will create a UUID and transmit that to the second endpoint. Likewise, the second endpoint will create a UUID and transmit that to the first endpoint. Each endpoint will then concatenate the two UUIDs to form the Session Identifier.

Note that it does not matter which endpoint might be considered the originating or terminating endpoint. For the purposes of session identification, it is only important that each endpoint create a UUID and transmit that value to the remote endpoint. Intermediaries such as session border controllers MUST NOT change any Session Identifier component received from an endpoint in a session.

What is also important is the order in which the UUIDs are concatenated together. To ensure that concatenation is performed consistently, a binary comparison is performed on the two UUIDs starting with the most significant byte. The UUID with the higher binary value is placed after the UUID with the lower binary value. Consider the following example.

Endpoint 1 produces this UUID: 0xaeffa652b22911dfa81f12313a006823

Endpoint 2 produces this UUID: 0xbe11afc8b22911df86c412313a006823

The resulting Session Identifier would be:
0xaeffa652b22911dfa81f12313a006823be11afc8b22911df86c412313a006823

In the above example, the UUIDs are presented as a string of hexadecimal characters that correspond to the binary values comprising the UUID as shown in the table at the end of Section 4.1.2 of RFC 4122 [4].

6. Transmitting the Session Identifier in SIP

Each session initiated or accepted MUST have a locally generated UUID associated with the session. This value MUST remain unchanged throughout the duration of the session and MUST persist even when the session is redirected (e.g., via a 3xx response) or transferred (e.g, via REFER [6]).

A SIP user agent MUST convey its Session Identifier UUID in all transmitted messages. To do this, each transmitted message MUST include the following header:

```
Session-ID-UUID: aeffa652-b229-11df-a81f-12313a006823
```

In the above example, the UUID is presented in string form with hyphens inserted as shown in the UUID ABNF syntax shown in Section 3 of [4]. Note that the namespace-related syntax “urn:uuid:” is NOT present in the Session-ID-UUID header.

The formal Session-ID-UUID header syntax is:

```
Session-ID-UUID = "Session-ID-UUID" HCOLON UUID
```

The actual Session Identifier is derived, as described in the previous section, by concatenating the locally generated UUID value and the UUID value received from the remote endpoint.

Intermediaries that wish to utilize the Session Identifier must take note of the UUIDs transmitted in each direction between endpoints. Intermediaries MUST NOT alter the UUIDs. If performing interworking between SIP and another session protocol, the intermediary MUST convert the Session-ID-UUID header as necessary so that it preserves the value of the UUID.

Developers should understand that a session MAY be transferred at any point and without any explicit signaling. This is not uncommon for back-to-back user agents that provide various call control functions. When the session is transferred, joined, or merged, perhaps a new INVITE message might be received bearing a new Session-ID-UUID value. When a new UUID is received, the endpoint MUST compute a new Session Identifier value, as the session has in fact changed. The endpoint MUST NOT generate a new UUID in response, however.

7. Associating Endpoints in a Multipoint Conference

Multipoint Control Units (MCUs) group two or more sessions into a single multipoint conference. Each session that is grouped into a conference SHOULD utilize the same UUID from the MCU to each of the endpoints in the conference. In so doing, each individual session in the conference will have a unique Session Identifier (since each endpoint will create a unique UUID of its own), but will also have one UUID in common with all other participants in the conference.

Intermediary devices, such as proxies or session border controllers, or network diagnostics equipment might assume that when they see two or more sessions with different Session Identifiers, but with one UUID in common, that the sessions are part of the same conference.

Note, however, that this assumption is true only if the sessions are operating in parallel. If A tries to establish a session with B and B redirects the session to C, each of A, B, and C will share at least one UUID in common (i.e., the UUID created by A). Likewise, if B transfers the session between A and B to C, A will retain its UUID and it will appear that A, B, and C are in a single conference. This is a desirable behavior, even though it may not be precise. It is assumed that any device that might wish to utilize this information would also recognize that a session is redirected or transferred.

8. Correlating Media Flows with Sessions

As mentioned previously, it may be desirable to insert the Session Identifier into media-related packets, such as RSVP messages or RTCP packets. In so doing, it is possible for network elements to

1. correlate session signaling with media flows,
2. associate multiple media flows with a single session, and
3. associate multiple media flows from multiple devices that are part of a single conference

Notwithstanding the foregoing, the use of the Session Identifier for purposes other than end-to-end session identification is outside the scope of this document.

9. Security Considerations

TBD

10. IANA Considerations

There are no IANA considerations associated with this document.

11. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

12. References

12.1. Normative References

[1] Rosenberg, J., et al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] Recommendation ITU-T H.323, "Packet-based multimedia communications systems", December 2009.

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

[4] Leach, P., Mealling, M., Salz, R., "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[5] Jones, et al., "Requirements for an End-to-End Session Identification in IP-Based Multimedia Communication Networks", draft-jones-ipmc-session-id-reqts-01.txt, January 2012.

12.2. Informative References

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

[7] Braden, R., et al., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[8] Schulzrinne, H., et al., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

Author's Addresses

Paul E. Jones
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC 27709
USA

Phone: +1 919 476 2048
Email: paulej@packetizer.com
IM: xmpp:paulej@packetizer.com

Chris Pearce
Cisco Systems, Inc.
2300 East President George Bush Highway
Richardson, TX 75082
USA

Phone: +1 972 813 5123
Email: chrep@cisco.com
IM: xmpp:chrep@cisco.com

James Polk
Cisco Systems, Inc.
3913 Treemont Circle
Colleyville, Texas, USAUSA

Phone: +1 817 271 3552
Email: jmpolk@cisco.com
IM: xmpp:jmpolk@cisco.com

Gonzalo Salgueiro
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC 27709
USA

Phone: +1 919 392 3266
Email: gsalguei@cisco.com
IM: xmpp:gsalguei@cisco.com