Mark Handley
ACIRI
Colin Perkins
UCL
Edmund Whelan
UCL

Session Announcement Protocol
draft-ietf-mmusic-sap-v2-05.txt

Status of this memo

This document is a product of the Multiparty Multimedia Session Control working group of the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at confctrl@isi.edu and/or the authors.

Abstract

This document describes version 2 of the multicast session directory announcement protocol, SAP, and the related issues affecting security and scalability that should be taken into account by implementors.

1 Introduction

In order to assist the advertisement of multicast multimedia conferences and other multicast sessions, and to communicate the relevant session setup information to prospective participants, a distributed session

directory may be used. An instance of such a session directory periodically multicasts packets containing a description of the session, and these advertisements are received by potential participants who can use the session description to start the tools required to participate in the session.

This memo describes the issues involved in the multicast announcement of session description information and defines an announcement protocol to be used. Sessions are described using the session description protocol which is described in a companion memo [4].

2 Terminology

A SAP announcer periodically multicasts an announcement packet to a well known multicast address and port. The announcement is multicast with the same scope as the session it is announcing, ensuring that the recipients of the announcement can also be potential recipients of the session the announcement describes (bandwidth and other such constraints permitting). This is also important for the scalability of the protocol, as it keeps local session announcements local.

A SAP listener learns of the multicast scopes it is within (for example, using the Multicast-Scope Zone Announcement Protocol [5]) and listens on the well known SAP address and port for those scopes. In this manner, it will eventually learn of all the sessions being announced, allowing those sessions to be joined.

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 [1].

3 Session Announcement

As noted previously, a SAP announcer periodically sends an announcement packet to a well known multicast address and port. There is no rendezvous mechanism - the SAP announcer is not aware of the presence or absence of any SAP listeners - and no additional reliability is provided over the standard best-effort UDP/IP semantics.

That announcement contains a session description and SHOULD contain an authentication header. The session description MAY be encrypted although this is NOT RECOMMENDED (see section 7).

A SAP announcement is multicast with the same scope as the session it is announcing, ensuring that the recipients of the announcement can also be potential recipients of the session being advertised. There a number of possiblities:

IPv4 global scope sessions use multicast addresses in the range 224.2.128.0 - 224.2.255.255 with SAP announcements being sent to 224.2.127.254 (note that 224.2.127.255 is used by the obsolete SAPv0 and MUST NOT be used).

IPv4 administrative scope sessions using administratively scoped IP multicast as defined in [7]. The multicast address to be used for announcements is the highest multicast address in the relevant administrative scope zone. For example, if the scope range is 239.16.32.0 - 239.16.33.255, then 239.16.33.255 is used for SAP announcements.

IPv6 sessions are announced on the address FF0X:0:0:0:0:0:2:7FFE where X is the 4-bit scope value. For example, an announcement for a link-local session assigned the address FF02:0:0:0:0:0:1234:5678, should be advertised on SAP address FF02:0:0:0:0:0:2:7FFE.

SAP announcements MUST be sent on port 9875 and SHOULD be sent with an IP time-to-live of 255 (the use of TTL scoping for multicast is discouraged [7]).

If a session uses addresses in multiple administrative scope ranges, it is necessary for the announcer to send identical copies of the announcement to each administrative scope range. It is up to the listeners to parse such multiple announcements as the same session (as identified by the SDP origin field, for example). The announcement rate for each administrative scope range MUST be calculated separately, as if the multiple announcements were separate.

Multiple announcers may announce a single session, as an aid to robustness in the face of packet loss and failure of one or more announcers. The rate at which each announcer repeats its announcement MUST be scaled back such that the total announcement rate is equal to that which a single server would choose. Announcements made in this manner MUST be identical.

If multiple announcements are being made for a session, then each announcement MUST carry an authentication header signed by the same key, or be treated as a completely separate announcement by listeners.

An IPv4 SAP listener SHOULD listen on the IPv4 global scope SAP address and on the SAP addresses for each IPv4 administrative scope zone it is within. The discovery of administrative scope zones is outside the scope of this memo, but it is assumed that each SAP listener within a particular scope zone is aware of that scope zone. A SAP listener which supports IPv6 SHOULD also listen to the IPv6 SAP addresses.

3.1 Announcement Interval

The time period between repetitions of an announcement is chosen such that the total bandwidth used by all announcements on a single

SAP group remains below a preconfigured limit. If not otherwise specified, the bandwidth limit SHOULD be assumed to be 4000 bits per second.

Each announcer is expected to listen to other announcements in order to determine the total number of sessions being announced on a particular group. Sessions are uniquely identified by the combination of the message identifier hash and originating source fields of the SAP header (note that SAP v0 announcers always set the message identifier hash to zero, and if such an announcement is received the entire message MUST be compared to determine uniqueness).

Announcements are made by periodic multicast to the group. The base interval between announcements is derived from the number of announcements being made in that group, the size of the announcement and the configured bandwidth limit. The actual transmission time is derived from this base interval as follows:

1. The announcer initialises the variable tp to be the last time a particular announcement was transmitted (or the current time if this is the first time this announcement is to be made).

2. Given a configured bandwidth limit in bits/second and an announcement of ad_size bytes, the base announcement interval in seconds is

   interval = max(300; (8*no_of_ads*ad_size)/limit)

3. An offset is calculated based on the base announcement interval

   offset = rand(interval * 2/3)-(interval/3)

4. The next transmission time for an announcement derived as

   tn = tp + interval + offset

The announcer then sets a timer to expire at tn and waits. At time tn the announcer SHOULD recalculate the next transmission time. If the new value of tn is before the current time, the announcement is sent immediately. Otherwise the transmission is rescheduled for the new tn. This reconsideration prevents transient packet bursts on startup and when a network partition heals.

4 Session Deletion

Sessions may be deleted in one of several ways:

Explicit Timeout The session description payload may contain timestamp information specifying the start- and end-times of the session. If the current time is later than the end-time of the session, then the session SHOULD be deleted from the receiver's session cache.

Implicit Timeout A session announcement message should be received periodically for each session description in a receiver's session cache. The announcement period can be predicted by the receiver from the set of sessions currently being announced. If a session announcement message has not been received for ten times the announcement period, or one hour, whichever is the greater, then the session is deleted from the receiver's session cache. The one hour minimum is to allow for transient network partitionings.

Explicit Deletion A session deletion packet is received specifying the session to be deleted. Session deletion packets SHOULD have a valid authentication header, matching that used to authenticate previous announcement packets. If this authentication is missing, the deletion message SHOULD be ignored.

5 Session Modification

A pre-announced session can be modified by simply announcing the modified session description. In this case, the version hash in the SAP header MUST be changed to indicate to receivers that the packet contents should be parsed (or decrypted and parsed if it is encrypted). The session itself, as distinct from the session announcement, is uniquely identified by the payload and not by the message identifier hash in the header.

The same rules apply for session modification as for session deletion:

- Either the modified announcement must contain an authentication header signed by the same key as the cached session announcement it is modifying, or:

- The cached session announcement must not contain an authentication header, and the session modification announcement must originate from the same host as the session it is modifying.

If an announcement is received containing an authentication header and the cached announcement did not contain an authentication header, or it contained a different authentication header, then the modified announcement MUST be treated as a new and different announcement, and displayed in addition to the un-authenticated announcement. The same should happen if a modified packet without an authentication header is received from a different source than the original announcement. These rules prevent an announcement having an authentication header added by a malicious user and then being deleted using that header, and it also prevents a denial-of-service attack by someone putting out a spoof announcement which, due to packet loss, reaches some participants before the original announcement. Note that under such

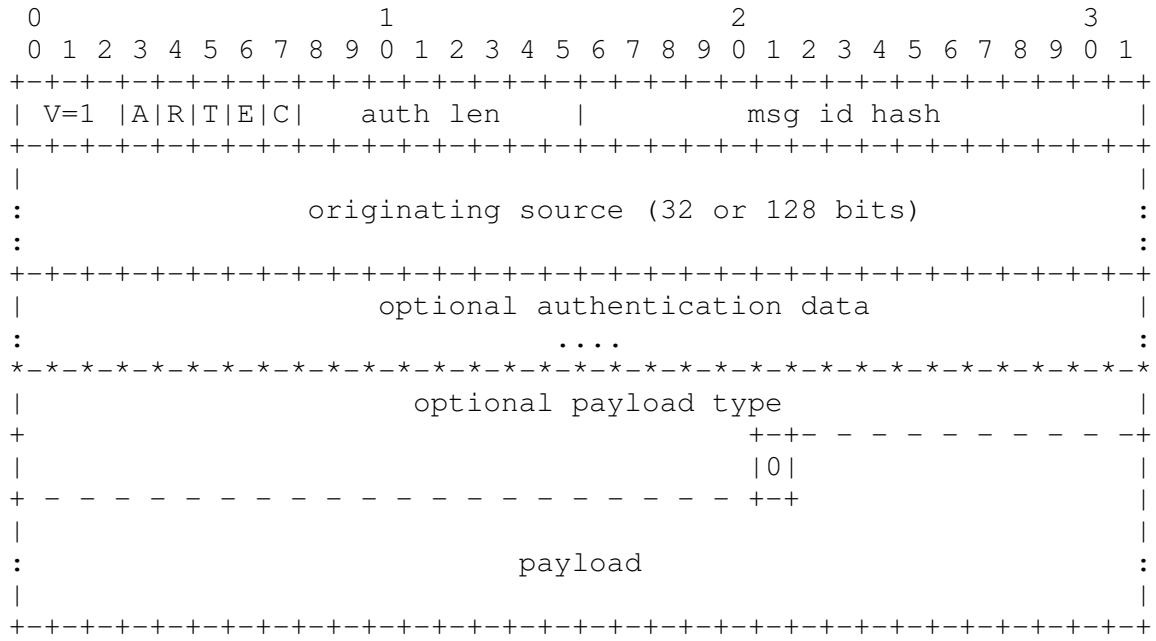

Figure 1: Packet format

circumstances, being able to authenticate the message originator is the only way to discover which session is the correct session.

6 Packet Format

SAP data packets have the format described in figure 1.

V: Version Number. The version number field MUST be set to 1 (SAPv2 announcements which use only SAPv1 features are backwards compatible, those which use new features can be detected by other means, so the SAP version number doesn't need to change).

A: Address type. If the A bit is 0, the originating source field contains a 32-bit IPv4 address. If the A bit is 1, the originating source contains a 128-bit IPv6 address.

R: Reserved. SAP announcers MUST set this to 0, SAP listeners MUST ignore the contents of this field.

T: Message Type. If the T field is set to 0 this is a session announcement packet, if 1 this is a session deletion packet.

E: Encryption Bit. If the encryption bit is set to 1, the payload of the SAP packet is encrypted. If this bit is 0 the packet is not encrypted. See section 7 for details of the encryption process.

C: Compressed bit. If the compressed bit is set to 1, the payload is compressed using the zlib compression algorithm [3]. If the payload is to be compressed and encrypted, the compression MUST be performed first.

Authentication Length. An 8 bit unsigned quantity giving the number of 32 bit words following the main SAP header that contain authentication data. If it is zero, no authentication header is present.

Authentication data containing a digital signature of the packet, with length as specified by the authentication length header field. See section 8 for details of the authentication process.

Message Identifier Hash. A 16 bit quantity that, used in combination with the originating source, provides a globally unique identifier indicating the precise version of this announcement. The choice of value for this field is not specified here, except that it MUST be unique for each session announced by a particular SAP announcer and it MUST be changed if the session description is modified.

Earlier versions of SAP used a value of zero to mean that the hash should be ignored and the payload should always be parsed. This had the unfortunate side-effect that SAP announcers had to study the payload data to determine how many unique sessions were being advertised, making the calculation of the announcement interval more complex that necessary. In order to decouple the session announcement process from the contents of those announcements, SAP announcers SHOULD NOT set the message identifier hash to zero.

SAP listeners MAY silently discard messages if the message identifier hash is set to zero.

Originating Source. This gives the IP address of the original source of the message. This is an IPv4 address if the A field is set to zero, else it is an IPv6 address. The address is stored in network byte order.

SAPv0 permitted the originating source to be zero if the message identifier hash was also zero. This practise is no longer legal, and SAP announcers SHOULD NOT set the originating source to zero. SAP listeners MAY silently discard packets with the originating source set to zero.

The header is followed by an optional payload type field and the payload data itself. If the E or C bits are set in the header both the payload type and payload are encrypted and/or compressed.

The payload type field is a MIME content type specifier, describing the format of the payload. This is a variable length ASCII text string, followed by a single zero byte (ASCII NUL). The payload type SHOULD be included in all packets. If the payload type is ‘application/sdp’

both the payload type and its terminating zero byte MAY be omitted, although this is intended for backwards compatibility with SAP v1 listeners only.

The absence of a payload type field may be noted since the payload section of such a packet will start with an SDP 'v=0' field, which is not a legal MIME content type specifier.

All implementations MUST support payloads of type 'application/sdp' [4]. Other formats MAY be supported although since there is no negotiation in SAP an announcer which chooses to use a session description format other than SDP cannot know that the listeners are able to understand the announcement. A proliferation of payload types in announcements has the potential to lead to severe interoperability problems, and for this reason, the use of non-SDP payloads is NOT RECOMMENDED.

If the packet is an announcement packet, the payload contains a session description.

If the packet is a session deletion packet, the payload contains a session deletion message. If the payload format is 'application/sdp' the deletion message is a single SDP line consisting of the origin field of the announcement to be deleted.

It is desirable for the payload to be sufficiently small that SAP packets do not get fragmented by the underlying network. Fragmentation has a loss multiplier effect, which is known to significantly affect the reliability of announcements. It is RECOMMENDED that SAP packets are smaller than 1kByte in length, although if it is known that announcements will use a network with a smaller MTU than this, then that SHOULD be used as the maximum recommended packet size.

7 Encrypted Announcements

An announcement is received by all listeners in the scope to which it is sent. If an announcement is encrypted, and many of the receivers do not have the encryption key, there is a considerable waste of bandwidth since those receivers cannot use the announcement they have received. For this reason, the use of encrypted SAP announcements is NOT RECOMMENDED on the global scope SAP group or on administrative scope groups which may have many receivers which cannot decrypt those announcements.

The opinion of the authors is that encrypted SAP is useful in special cases only, and that the vast majority of scenarios where encrypted SAP has been proposed may be better served by distributing session details using another mechanism. There are, however, certain scenarios where encrypted announcements may be useful. For this reason, the encryption bit is included in the SAP header to allow experimentation with encrypted announcements.

This memo does not specify details of the encryption algorithm to be used or the means by which keys are generated and distributed. An additional specification should define these, if it is desired to use encrypted SAP.

Note that if an encrypted announcement is being announced via a proxy, then there may be no way for the proxy to discover that the announcement has been superseded, and so it may continue to relay the old announcement in addition to the new announcement. SAP provides no mechanism to chain modified encrypted announcements, so it is advisable to announce the unmodified session as deleted for a short time after the modification has occurred. This does not guarantee that all proxies have deleted the session, and so receivers of encrypted sessions should be prepared to discard old versions of session announcements that they may receive. In most cases however, the only stateful proxy will be local to (and known to) the sender, and an additional (local-area) protocol involving a handshake for such session modifications can be used to avoid this problem.

Session announcements that are encrypted with a symmetric algorithm may allow a degree of privacy in the announcement of a session, but it should be recognised that a user in possession of such a key can pass it on to other users who should not be in possession of such a key. Thus announcements to such a group of key holders cannot be assumed to have come from an authorised key holder unless there is an appropriate authentication header signed by an authorised key holder. In addition the recipients of such encrypted announcements cannot be assumed to only be authorised key holders. Such encrypted announcements do not provide any real security unless all of the authorised key holders are trusted to maintain security of such session directory keys. This property is shared by the multicast session tools themselves, where it is possible for an un-trustworthy member of the session to pass on encryption keys to un-authorised users. However it is likely that keys used for the session tools will be more short lived than those used for session directories.

Similar considerations should apply when session announcements are encrypted with an asymmetric algorithm, but then it is possible to restrict the possessor(s) of the private key, so that announcements to a key-holder group can not be made, even if one of the untrusted members of the group proves to be un-trustworthy.

8 Authenticated Announcements

The authentication header can be used for two purposes:

- Verification that changes to a session description or deletion of a session are permitted.
- Authentication of the identity of the session creator.

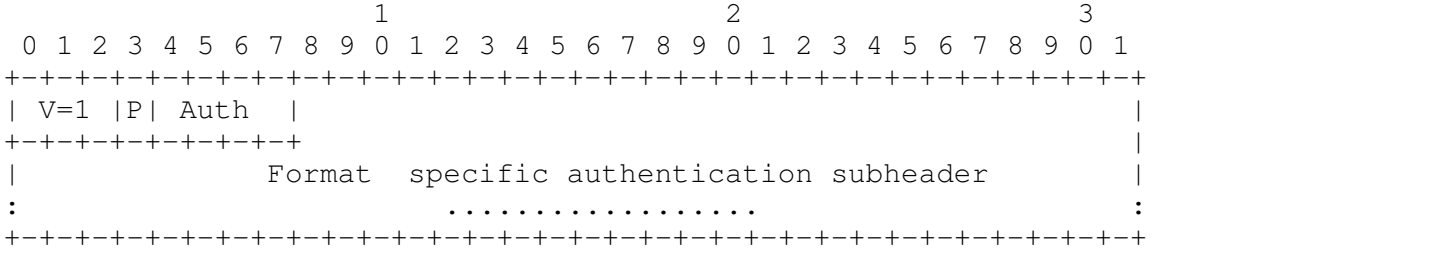

Figure 2: Format of the authentication data in the SAP header

In some circumstances only verification is possible because a certificate signed by a mutually trusted person or authority is not available. However, under such circumstances, the session originator may still be authenticated to be the same as the session originator of previous sessions claiming to be from the same person. This may or may not be sufficient depending on the purpose of the session and the people involved.

Clearly the key used for the authentication should not be trusted to belong to the session originator unless it has been separately authenticated by some other means, such as being certified by a trusted third party. Such certificates are not normally included in an SAP header because they take more space than can normally be afforded in an SAP packet, and such verification must therefore take place by some other mechanism. However, as certified public keys are normally locally cached, authentication of a particular key only has to take place once, rather than every time the session directory retransmits the announcement.

SAP is not tied to any single authentication mechanism. Authentication data in the header is self-describing, but the precise format depends on the authentication mechanism in use. The generic format of the authentication data is given in figure 2. The structure of the format specific authentication subheader, using both the PGP and the CMS formats, is discussed in sections 8.1 and 8.2 respectively.

Version Number, V: The version number of the authentication format specified by this memo is 1.

Padding Bit, P: If necessary the authentication data is padded to be a multiple of 32 bits and the padding bit is set. In this case the last byte of the authentication data contains the number of padding bytes (including the last byte) that must be discarded.

Authentication Type, Auth: The authentication type is a 4 bit encoded field that denotes the authentication infrastructure the sender expects the recipients to use to check the authenticity and integrity of the information. This defines the format of the authentication

subheader and can take the values: 0 = PGP format, 1 = CMS format. All other values are undefined and SHOULD be ignored.

If a SAP packet is to be compressed or encrypted, this MUST be done before the authentication is added.

The digital signature in the authentication data MUST be calculated over the entire packet, including the header. The authentication length MUST be set to zero and the authentication data excluded when calculating the digital signature.

It is to be expected that sessions may be announced by a number of different mechanisms, not only SAP. For example, a session description may placed on a web page, sent by email or conveyed in a session initiation protocol. To ease interoperability with these other mechanisms, application level security is employed, rather than using IPsec authentication headers.

8.1 PGP Authentication

Implementations which support authentication MUST support this format. A full description of the PGP protocol can be found in [2]. When using PGP for SAP authentication the basic format specific authentication subheader comprises a digital signature packet as described in [2]. The signature type MUST be 0x01 which means the signature is that of a canonical text document.

8.2 CMS Authentication

Support for this format is OPTIONAL.

A full description of the Cryptographic Message Syntax can be found in [6]. The format specific authentication subheader will, in the CMS case, have an ASN.1 ContentInfo type with the ContentType being signedData.

Use is made of the option available in PKCS#7 to leave the content itself blank as the content which is signed is already present in the packet. Inclusion of it within the SignedData type would duplicate this data and increase the packet length unnecessarily. In addition this allows recipients with either no interest in the authentication, or with no mechanism for checking it, to more easily skip the authentication information.

There SHOULD be only one signerInfo and related fields corresponding to the originator of the SAP announcement. The signingTime SHOULD be present as a signedAttribute. However, due to the strict size limitations on the size of SAP packets, certificates and CRLs SHOULD

NOT be included in the signedData structure. It is expected that users of the protocol will have other methods for certificate and CRL distribution.

9 Scalability and caching

SAP is intended to announce the existence of long-lived wide-area multicast sessions. It is not an especially timely protocol: sessions are announced by periodic multicast with a repeat rate on the order of tens of minutes, and no enhanced reliability over UDP. This leads to a long startup delay before a complete set of announcements is heard by a listener. This delay is clearly undesirable for interactive browsing of announced sessions.

In order to reduce the delays inherent in SAP, it is recommended that proxy caches are deployed. A SAP proxy cache is expected to listen to all SAP groups in its scope, and to maintain an up-to-date list of all announced sessions along with the time each announcement was last received. When a new SAP listeners starts, it should contact its local proxy to download this information, which is then sufficient for it to process future announcements directly, as if it has been continually listening.

The protocol by which a SAP listener contacts its local proxy cache is not specified here.

10 Security Considerations

SAP contains mechanisms for ensuring integrity of session announcements, for authenticating the origin of an announcement and for encrypting such announcements (sections 7 and 8).

As stated in section 5, if a session modification announcement is received that contains a valid authentication header, but which is not signed by the original creator of the session, then the session must be treated as a new session in addition to the original session with the same SDP origin information unless the originator of one of the session descriptions can be authenticated using a certificate signed by a trusted third party. If this were not done, there would be a possible denial of service attack whereby a party listens for new announcements, strips off the original authentication header, modifies the session description, adds a new authentication header and re-announces the session. If a rule was imposed that such spoof announcements were ignored, then if packet loss or late starting of a session directory instance caused the original announcement to fail to arrive at a site, but the spoof announcement did so, this would then prevent the original announcement from being accepted at that site.

A similar denial-of-service attack is possible if a session announcement receiver relies completely on the originating source and hash fields to indicate change, and fails to parse the remainder of announcements for which it has seen the origin/hash combination before.

A denial of service attack is possible from a malicious site close to a legitimate site which is making a session announcement. This can happen if the malicious site floods the legitimate site with huge numbers of (illegal) low TTL announcements describing high TTL sessions. This may reduce the session announcement rate of the legitimate announcement to below a tenth of the rate expected at remote sites and therefore cause the session to time out. Such an attack is likely to be easily detectable, and we do not provide any mechanism here to prevent it.

A Summary of differences between SAPv0 and SAPv1

For this purpose SAPv0 is defined as the protocol in use by version 2.2 of the session directory tool, sdr. SAPv1 is the protocol described in the 19 November 1996 version of this memo (draft-ietf-mmusic-sap-00.txt). The packet headers of SAP messages are the same in V0 and V1 in that a V1 tool can parse a V0 announcement header but not vice-versa.

In SAPv0, the fields have the following values:

- Version Number: 0
- Message Type: 0 (Announcement)
- Authentication Type: 0 (No Authentication)
- Encryption Bit: 0 (No Encryption)
- Compression Bit: 0 (No compression)
- Message Id Hash: 0 (No Hash Specified)
- Originating Source: 0 (No source specified, announcement has not been relayed)

B Summary of differences between SAPv1 and SAPv2

The packet headers of SAP messages are the same in V1 and V2 in that a V2 tool can parse a V1 announcement header but not necessarily vice-versa.

- The A bit has been added to the SAP header, replacing one of the bits of the SAPv1 message type field. If set to zero the announcement is of an IPv4 session, and the packet is backwards

compatible with SAPv1. If set to one the announcement is of an IPv6 session, and SAPv1 listeners (which do not support IPv6) will see this as an illegal message type (MT) field.

- The second bit of the message type field in SAPv1 has been replaced by a reserved, must-be-zero, bit. This bit was unused in SAPv1, so this change just codifies existing usage.

- SAPv1 specified encryption of the payload. SAPv2 includes the E bit in the SAP header to indicate that the payload is encrypted, but does not specify any details of the encryption.

- SAPv1 allowed the message identifier hash and originating source fields to be set to zero, for backwards compatibility. This is no longer legal.

- SAPv1 specified gzip compression. SAPv2 uses zlib (the only known implementation of SAP compression used zlib, and gzip compression was a mistake).

- SAPv2 provides a more complete specification for authentication.

- SAPv2 allows for non-SDP payloads to be transported. SAPv1 required that the payload was SDP.

- SAPv1 included a timeout field for encrypted announcement, SAPv2 does not (and relies of explicit deletion messages or implicit timeouts).

C Acknowledgments

SAP and SDP were originally based on the protocol used by the sd session directory from Van Jacobson at LBNL. Version 1 of SAP was designed by Mark Handley as part of the European Commission MICE (Esprit 7602) and MERCI (Telematics 1007) projects. Version 2 includes authentication features developed by Edmund Whelan, Goli Montasser-Kohsari and Peter Kirstein as part of the European Commission ICE-TEL project (Telematics 1005), and support for IPv6 developed by Maryann P. Maher and Colin Perkins.

D Authors' Addresses

Mark Handley <mjh@aciri.org>
AT&T Center for Internet Research at ICSI,
International Computer Science Institute,
1947 Center Street, Suite 600,
Berkeley, CA 94704, USA

Colin Perkins <c.perkins@cs.ucl.ac.uk>
Department of Computer Science,
University College London,
Gower Street,
London, WC1E 6BT, UK

Edmund Whelan <e.whelan@cs.ucl.ac.uk>
Department of Computer Science,
University College London,
Gower Street,
London, WC1E 6BT, UK

References

[1] S. Bradner. Key words for use in RFCs to indicate requirement levels, March 1997. RFC2119.

[2] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP message format, November 1998. RFC2440.

[3] P. Deutsch and J.-L. Gailly. Zlib compressed data format specification version 3.3, May 1996. RFC1950.

[4] M. Handley and V. Jacobson. SDP: Session Description Protocol, April 1998. RFC2327.

[5] M. Handley, D. Thaler, and R. Kermode. Multicast-scope zone announcement protocol (MZAP), February 2000, RFC2776.

[6] R. Housley. Cryptographic message syntax, June 1999. RFC2630.

[7] D. Mayer. Administratively scoped IP multicast, July 1998. RFC2365.