Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-mmusic-sdp-00.ps

MMUSIC WG
Mark Handley/Van Jacobson
UCL/LBL

26th March 1995
Expires: 26th Sept 1995

SDP: Session Description Protocol

Status of this Memo

Abstract

The *sd* session directory tool has been in use for some time on the Mbone for announcing multicast sessions. This document describes an exhanced version of the sd protocol (SDP v2), and explains the extensions to the protocol that have become desirable.

This document is a product of the Multiparty Multimedia Session Control (MMUSIC) 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.

1. Introduction

The *sd* session directory tool from Van Jacobson at LBL has been in use on the Mbone for some time to advertise multimedia conferences and communicate the conference addresses and conference tool specific information necessary for participation. This document is an attempt to describe this protocol and some extensions to the protocol that have become desirable. In the body of the paper, we describe a proposed SDP v2. In Appendix A, we describe how this differs from the protocol currently used by sd (SDP v1).

2. Background

The Multicast Backbone (Mbone) is an experimental overlay network on the internet that permits efficient many to many communication. For the past few years it has been used extensively for multimedia conferencing. Such multimedia conferences usually have the property that tight coordination of conference membership is not necessary; in order to receive a conference, a user at an Mbone site only has to know the correct multicast group address for the conference and the UDP ports the conferencing applications will use to receive the conference data streams.

In order to assist the advertisement of conference sessions and to communicate the relevant conference setup information to prospective participants, the *session directory* (sd) tool was written. Sd has now been in wide scale use for close to 2 years, during which time Mbone usage has greatly increased and diversified. The Mbone has now reached the stage where assistance with coordination of resource usage is required, and where compatible session announcement tools are starting to emerge.

This document is an attempt to prevent diversification of the sd protocol as tool writers each add their own modifications. It is also an attempt to provide guidelines to the writers of such announcement tools in order to protect the Mbone from misuse and to preserve the inherent scalability of the original sd program whilst enhancing its functionality.

3. The sd protocol

SDP is an announcement protocol for multicast conference sessions. An sd client that announces a conference session periodically multicasts an announcement packet on a well known multicast address and port. The announcement is multicast with the same TTL as the session it is announcing, which ensures that the recipients of the announcement can also be potential recipients of the session the announcement describes (bandwidth permitting). This is also important for the scalability of the protocol, as it keeps local session announcements local.

The time period between one announcement and its repetition is dependent on two factors - the scope (TTL) of the session, and the number of other sessions currently being announced by other sd clients. The goal is to keep the total bandwidth being used to below a predefined level for each scope.

Sd packets are of the following format:

```
0                   31
+--------------------+
| reserved (set to 0)|
+--------------------+
|original src IP addr|
+--------------------+
| text payload       |
|/\/\/\/\/\/\/\/\/\/\|
```

The first four bytes are reserved for authentication data for proxy announcements are should otherwise be set to zeros.

The second four bytes contain the IP address of the source host from which this announcement was originally made. When announcements are made by proxy, this allows the original host to be identified.

The text payload is an SDP session description, as described below.

4. Proposed Additional Functionality (SDP v2)

SDP version 1 is currently in use in the sd session directory tool from LBL. It allows the announcement of conference sessions some time in advance, and the starting of conference tools from sd with the correct parameters for participating in a conference. Use of sd over an extended period has shown it to be scalable, reliable and robust.

However, sd has failed to be used as a conference *booking* tool (as opposed to a conference startup and address allocation tool) , and currently the main way to avoid conflicts for bandwidth usage on the Mbone is to use a mailing list (send email to `rem-conf-request@es.net` to subscribe). A number of people have proposed using the World Wide Web for conference announcements, but this is not as scalable as sd, and does not have sd's scoping benefits (receivers of announcements are those in the correct scope to receive the conference).

One reason why sd has not been used as the main mechanism for conference booking is that its single presentation style (all sessions alphabetically listed by conference name) discourages bookings significantly in advance, and makes potential clashes difficult to spot. Session announcements also contain no bandwidth information, which makes it difficult to see if a potential clash is likely to cause a problem.

It has been the tradition that sessions that are going to be intermittently active over a fixed period are given a single announcement for that fixed period. However, this does not help a user decide whether there will be a clash at a particular time. There is also a need for a repeat booking mechanism to allow for bookings of the form ''every Wednesday at 10am GMT for 1 hour'', or ''every weekday from 1-2pm PST in the week beginning 19th July''.

Another problem is that of private sessions. It has been observed that the ability to announce private sessions in sd (sessions that are only visible to authorised receivers) greatly increases the number of sessions. However, the Mbone does not have enough bandwidth available to cope with many such private sessions if they have international, or even national, scope. The authors do not wish the Mbone to be viewed as a private free telephone service under the current internet funding process. Instead there has been an unwritten goal that those sessions that are of the most interest to the most people (such as IETF conferences) should be those to have priority. However, there have been a number of cases where sessions scheduled a long time in advance have been disrupted by sessions unannounced either in sd or on the rem-conf mailing list. The information necessary to participate in these sessions was usually communicated by electronic mail.

It is inevitable that if people cannot announce their private sessions in sd, they will use another mechanism to set up private sessions. It is also likely that setting up private sessions in sd will increase unreasonable Mbone usage. There is clearly a need for semi-private announcements which convey that there is a desire to have a conference at a particular time which will require a certain amount of resources, but which does not convey enough information to join the conference. These semi-private announcements should also contain enough contact information (email address and telephone number) to alert the person responsible for the semi-private conference of any potential clashes, and to negotiate a suitable usage of the network resources. It should be noted that advance discovery of potential clashes on bandwidth limited links is required irrespective of the deployment of resource reservation mechanisms in the network.

It has become normal to provide additional descriptive information about a conference on the world wide web. This has been added in the descriptive information field in the sd protocol, but we believe there should now be a specific place for a URL in the sd protocol.

Current sd announcements are in ASCII. The specification recommends the use of 8 bit ISO 8859 character sets to allow the extended ASCII characters used by many western and northern European languages to be represented. However, there are many languages that cannot be represented in an ISO 8859 character set. SDP needs an extension to allow other font types to be used when required.

4.1. SDP v2 Session Descriptions

Sd session descriptions are entirely ASCII text. The textual form, as opposed to a binary encoding such as ASN/1 or XDR, was chosen to enhance portability, to enable a variety of transports to be used (e.g, session description in an email message) and to allow flexible, text-based toolkits (e.g., tk/tcl) to be used to generate and process session descriptions. However, since the total bandwidth allocated to all sd announcements is strictly limited, the encoding is deliberately compact. Also, since announcements may be transported via very unreliable means (e.g., email) or damaged by an intermediate caching server, the encoding was designed with strict order and formatting rules so that likely errors would result in malformed announcements and malformed announcements could be easily detected and discarded.

An sd session description takes the form of a number of lines of ASCII text of the form <type>=<value>. <type> is always exactly one character and case is significant. <value> is a structured text string whose format depends on <type>. In general <value> is either a number of fields delimited by a single space character or free format string.

Each announcement consists of a session description section followed by zero or more ‘media’ description sections. The session description starts with an ‘s=’ line and continues to the first media description or the next session description. The media description starts with an ‘m=’ line and continues to the next media description or session description. Some lines in each description are required and some are optional but all must appear in exactly the order given here (the fixed order greatly enhances error dectection and allows for a simple parser). (Optional items are marked with a ‘*’.)

Session description
s= (session name)
i=* (description)
u=* (URL of description)
o= (owner/creator)
e=* (email address)
p=* (phone number)
h=* (originating host)
c= (connection info)
b=* (bandwidth info)
t=* (zero or more repeat times)
k=* (encryption key)
a=* (zero or more session attribute lines)

Media description
m= (media name and transport addr)
c=* (connection info)
k=* (encryption key)
a=* (zero or more media attribute lines)

The set of 'type' letters is deliberately small and not intended to be extensible -- sd must completely ignore any announcement that contains a 'type' letter that it does not understand. The 'attribute' mechanism (described below) is the primary means for extend sd and tailoring it to particular applications or media. Some attributes (the ones listed in this document) have a defined meaning but others may be added on an application, media or session specific basis. Sd must ignore any attribute it doesn't understand.

The connection ('c=') and attribute ('a=') information in the session section applies to all the media of that session unless overridden by connection info or an attribute of the same name in the media description. E.g., in the example below each media behaves as if it were given a 'recvonly' attribute.

An example sd description is:

```
s=Sd Seminar
i=A Seminar on the sd protocol used by LBL's sd tool
o=mhandley@maus.cs.ucl.ac.uk
c=224.5.17.11 127 2873397496 2873404696
a=recvonly
m=audio 3456 0
m=video 2232 0
a=fmt:h261
m=whiteboard
a=orient:portrait
```

Text records such as the session name and information may contain any 8 bit ISO 8859 character with the exceptions of 0x0a (newline) and 0x0d (carriage return). Carriage Return is prohibited, and Newline is used to end a record.

SDP version 1 is a subset of the session description described below. The differences between

SDP v1 and SDP v2 are described in Appendix A.

Session Name

`s=`*<session name>*

The ‘‘s’’ field is the session name. This must be one and only one ‘‘s’’ field per announcement, and it must contain printable ISO 8859 characters.

Information

`i=`*<session description>*

The ‘‘i’’ field is information about the session. There must be no more than one ‘‘i’’ field per session announcement. Although it can be omitted, this is discouraged, and user interfaces for composing sessions should require text to be entered. If it is present it must contain printable ISO 8859 characters (but see also the ‘`charset`’ sttribute below). Descriptions may contain embedded newlines (indicated via a ‘\n’ escape sequence) but it is strongly suggested that they do not - user interface agents format this text to fit in some arbtrarily sized window and explicit line breaks often defeat this formatting and produce an ugly result.

Originator

`o=`*username@hostname*

The ‘‘o’’ field gives the originator of the session. This is defined as ‘‘*username@hostname*’’ where *username* is the user’s login on the originating host (or it is blank if the originating host does not support the concept of user ids) and the *hostname* is the fully qualified host name of the host the session was announced from. This is not necessarily the email address of the originator, though in many cases it may work as such.

Conference Data

`c=`*<conference address> <ttl> <start time> <stop time>*

The ‘‘c’’ field is conference data. It contains a number of sub-fields:

- The first sub-field is the conference multicast address. If the conference is not multicast, then this contains the unicast IP address of the expected data source or data relay. It is not expected that unicast addresses will be given in a session description that is communicated by a multicast announcement.

  The RSVP WG of the IETF has defined that hierarchical encoding schemes should be transmitted in multiple multicast groups to allow multicast pruning to keep unwanted traffic from sites only requiring some levels of the hierarchy. For applications which require multiple multicast groups, we allow the following notation to be used for the conference address:

  <base multicast address>`/`*<number of addresses>*

  If the number of addresses is not given it is assumed to be one. Multicast addresses so assigned are contiguously allocated above the base address, so that, for example:

  ```
  c=224.2.1.1/3 127 0 0
  ```

  would state that addresses 224.2.1.1, 224.2.1.2 and 224.2.1.3 are to be used.

- The second sub-field contains the TTL (time to live) with which multicast packets sent in this conference should be sent. This bounds the scope of the conference. If, however, the conference address given is a unicast address then this field should be zero, and should be ignored by receivers. TTL values must be in the range 0-255. The MBone usage guidelines (currently available at ftp://ftp.isi.edu/mbone/faq.txt) define several standard settings for ttl:

```
local net:     1
site:         15
region:       63
world:       127
```

Other settings may have local meaning (e.g., 47 for all sites within an organization).

- The third and fourth sub-fields give the start and stop times for the conference respectively. These values are the decimal representation of Network Time Protocol (NTP, [1]) time values in seconds. To convert these values to UNIX time, subtract decimal 2208988800. If these values are both set to zero, then the conference is not bounded. User interfaces should prohibit or strongly discourage the creation of unbounded conferences as they give no information about when the session is actually going to be active. It is prohibited for the start time to be after the stop time.

A session announcement must contain at least one ‘‘c’’ field. It may contain one additional ‘‘c’’ field per media field (see below), in which case the per media values override the conference wide settings for the relevant media. The start and stop times should normally be the same as for the conference as a whole, though they can be for a shorter range of time. It is prohibited for a media start time to be before the conference start time, or for a media stop time to be after the conference stop time.

Media Announcements

`m=`*<media> <port> <id>*

A session announcement may also contain a number of media announcements. Each media announcement starts with an ‘‘m’’ field, and is terminated by either the next ‘‘m’’ field, or by the end of the session announcement. A media field also has several sub-fields:

- The first sub-field is the media type. Currently defined media are ‘‘audio’’, ‘‘video’’, and ‘‘whiteboard’’, though this list may be extended as new communication modalities emerge (e.g., telepresense or conference control).

- The second sub-field is the UDP port to which the media stream will be sent. Other ports used by the media application (such as the RTCP port, see [2]) should be derived algorithmically from the base media port. The value should be between 1025 and 65535. For RTPv2 complience it should be an even number.

- The third sub-field is the ID. This is currently used by vat and by RTPv1 media tools, but should become obsolete as media tools move to RTPv2 (see [2]) compliance. This field is here for compatibility reasons only and should be set to zero.

Attributes

`a=`*<flag>*
`a=`*<attribute>*`:`*<value>*

A media field may also have any number of attributes (‘‘a’’ fields) which are media specific. Attribute fields may be of two forms:

- flag attributes. A flag attribute is simply of the form ‘‘`a=`*flag*’’. These are binary attributes, and the presence of the attribute conveys that the attribute is ‘‘true’’.
- value attributes. A value attribute is of the form ‘‘`a=`*attribute*`:`*value*’’. An example might be that a whiteboard could have the value attribute ‘‘`a=orient:landscape`’’

Attribute interpretation depends on the media tool being invoked. Thus receivers of sd session descriptions should be configurable in their interpretation of announcements in general and of attributes in particular.

Attribute fields (‘‘a’’ fields) can also be added before the first media field. These attributes would convey additional information that applies to the conference as a whole rather than to individual media. An example might be the conference’s floor control policy.

Bandwidth

`b=`*bandwidth [modifier]*

- This specifies the proposed bandwidth to be used by the conference
- *bandwidth* is in kilobits per second
- *modifier* is an optional single ASCII word giving the meaning of the bandwidth figure. Currently no modifiers are defined.
- If no modifier is given, bandwidth is interpreted to be application specific, i.e., will be the application’s concept of maximum bandwidth. Normally this will coincide with what is set on the applications ‘‘max. bandwidth’’ control if applicable.
- An implicit max. bandwidth is associated with each ttl on the MBone or within a particular multicast administrative scope region (the MBone bandwidth vs. ttl limits are given in the mbone faq). If the bandwidth of a session or media in a session is different from the bandwidth implicit in the ttl, ‘b=’ lines should be supplied for the session and/or media.

Normally, the bandwidth setting is taken as an upper limit. However, there is potential for sd clients or media applications to interpret bandwidth data and to determine sensible rate limits or compression schemes as appropriate. There is also potential for floor control agents to dynamically allocate a conference’s bandwidth budget between the media applications of the conference, depending on the activity in the conference, by performing such actions as switching a proportion of video bandwidth to sending sites who are also active verbally.

Email Address and Phone Number

`e=`<*email address*>
`p=`<*phone number*>

- These specify contact information for the person responsible for the conference. This is not necessarily the same person that created the conference announcement.
- One or both of these *must* be specified for a semi-private conference (i.e., one that possesses no media data).
- If these are present, they should be specified before the first media field.
- More than one email or phone field can be given for a session description.

URL

`u=`<*URL*>

- A URL is a Uniform Resource Locator as used by WWW clients.
- The URL should be a pointer to additional information about the conference.
- This field is optional, but if it is present it should be specified before the first media field.
- No more than one URL field is allowed per session description.

Originating Host

`h=`<*originating host*>

- This is the IP addresses in dotted decimal notation of the host from which the announcement was first made.
- It is not required if the announcement is made using a multicast sd announcement, as the data is already present in binary form in the sd announcement packet. However, it should be added if the announcement is being conveyed by other means such as email or the World Wide Web.

Repeat Times

`t=`<*min*> <*hour*> <*day of mon*> <*mon*> <*day of week*> <*duration in seconds*>

- These allow the specification of when a session will be active and when it will be inactive between it's start and stop times.
- Each takes the form of a extended UNIX crontab entry, with six subfields. These subfields are integer patterns that specify the following:
  - minute (0-59)
  - hour (0-23)
  - day of the month (1-31)
  - month of the year (1-12)
  - day of the week (0-6, with 0=Sunday)
  - duration of session in seconds
- Each of the first five patterns may be either an asterisk (meaning all legal values) or a list of elements separated by commas. Duration is not allowed to be an asterisk, but must be a single positive integer. For example, to specify ''every Monday and Tuesday from 0815hrs UTC to 0900hrs UTC'', the field would have the value:

  ```
  t=15 8 * * 1,2 2700
  ```
- Note that times and dates must be in UTC.
- The user is not normally expected to enter these fields manually - a suitable user interface should be provided.
- ''t'' fields are optional. If present they should be before the first media field. More than one ''t'' field may be present per announcement - in which case each field specifies additional time when the session is active. The active times specified by multiple ''t'' fields should not

overlap.

Encryption Keys

`k=`<*encryption key*>

- In countries where encrypted conferences are legally permitted, the session description protocol may be use to convey encryption keys.
- A key field is permitted before the first media entry, or for each media entry as required.
- The format of keys and their usage is outside the scope of this document, but see [3]

Suggested Attributes

The following attributes are suggested. As application writers may add new attributes as they are required, this list is not exhaustive.

`a=fmt:`<*media format*>

This specifies the media format being used. The following media format strings are currently suggested, as used by vat [4] and vic [5]:

Audio:

`pcm`: 78Kb/s 8-bit mu-law encoded 8KHz PCM (20ms frames)
`pcm2`: 71Kb/s 8-bit mu-law encoded 8KHz PCM (40ms frames)
`pcm4`: 68Kb/s 8-bit mu-law encoded 8KHz PCM (80ms frames)
`idvi`: 46Kb/s Intel DVI ADPCM (20ms frames)
`dvi2`: 39Kb/s Intel DVI ADPCM (40ms frames)
`dvi4`: 36Kb/s Intel DVI ADPCM (80ms frames)
`gsm`: 17Kb/s GSM (80ms frames)
`lpc4`: 9Kb/s Linear Predictive Coder (80ms frames)

Video:

`jpeg`: Motion JPEG
`jpg`: Still image JPEG (as used by imm [6])
`mpeg`: MPEG
`h261`: CCITT H.261
`nv`: Xerox PARC Network Video
`nvm`: uncompressed (with nv's thresholding algorithm)
`cellb`: Sun CellB

Whiteboard:

`wb`: the *wb* whiteboard from LBL.

`a=recvonly`

This specifies that the tools should be started in receive only mode where applicable.

`a=sendrecv`

This specifies that the tools should be started in send and receive mode. This is necessary for interactive conferences with tools such as *wb* which defaults to receive only mode.

`a=orient:`<*whiteboard orientation*>

Normally only used in a whiteboard media specification, this specifies the orientation of a the whiteboard on the screen. Permitted values are '`portrait`', '`landscape`' and '`seascape`'.

`a=type:`<*conference type*>

This specifies the type of the conference. Suggested values are '`broadcast`', '`meeting`', and '`moderated`'. 'recvonly' should be the default for '`type:broadcast`' sessions, '`type:meeting`' should imply 'sendrecv' and '`type:moderated`' should imply the use of a floor control tool that the media tools are started to mute new sites.

`a=charset:`<*character set*>

This specifies the character set to be used to display the session name and information data. By default, an ISO8859 character set is used. If an ISO8859 character set is not suitable, the use of unicode (ISO 10646, [7],[8]), as specified in rfc1641 [9] is suggested. In particular, the UTF-7 (rfc1642, [10]) encoding is suggested with the following sdp:

```
a=charset:unicode-1-1-utf-7
```

If this field is present, it must be before the first media field.

4.2. Unresolved Issues

A few issues have been raised that this first draft leaves unresolved.

4.2.1. Conference Control Policy

There is some debate over the way conference control policy should be communicated. In general, the authors believe that an implicit declarative style of specifying conference control is desirable where possible.

A simple declarative style uses a single conference attribute field before the first media field, possibly supplimented by flags such as '`recvonly`' for some of the media tools. This conference attribute conveys the conference control policy. An example might be:

```
a=type:moderated
```

In some cases, however, it is possible that this may be insufficient to communicate the details of an unusual conference control policy. If this is the case, then a conference attribute specifying external control might be set, and then one or more ''media'' fields might be used to specify the conference control tools and configuration data for those tools. A fictional example might be:

```
...
a=type:external-control
m=audio 12345 0
m=video 12347 0
m=whiteboard 12349 0
m=control 12341 0
a=fmt:chaired-type-3
a=chair:128.16.64.2
a=video:follows-audio
a=audio:on-demand
a=audio:chair-mutes-mike
a=whiteboard:chaired
```

In this fictional example (i.e., this is not implemented anywhere), a general conference attribute is specified stating that conference control will be provided by an external tool, and specific attributes are given to specify the control policy that tool should use.

In this document, we only specify the former style of conference control declaration, though we recognise that some variant on the latter may also be used eventually.

4.2.2. Multiple Multicast Groups for Hierarchical Encoding Schemes

A number of hierarchical encoding schemes have been suggested for video and audio compression, although at the time of writing, none have been deployed on the Mbone. The RSVP (Resource reSerVation Protocol) working group has made the recommendation that the layers of a hierarchy are carried on multiple multicast groups to allow multicast distribution only of required traffic to each receiver.

SDP version 1 only allows each media to have one multicast address. A preliminary suggestion is for the different multicast groups of a hierarchy to be algorithmically derived from the base group, rather than being explicitly specified. This means session directory tools do not have to know about different encoding schemes and the number of levels of hierarchy present. A similar approach is taken for RTCP port numbers relative to RTP ports.

However, sd also uses a smart address allocation algorithm known as ‘‘informed, partitioned, random allocation’’. This relies on listening to the addresses being announced by other sites to avoid simultaneous duplicate address allocations. However, if only one address from a block of addresses is announced, sd cannot know which (if any) other addresses are in use, and thus cannot avoid duplicate address allocations. This document makes the strawman proposal that announcements contain the first address of a block and the size of the block (where the size defaults to one if not supplied). This allows address allocation to be done based solely on the announcements but does imply that the number of addresses used by a media will be fixed at the time the session is created. It does not seem like this is a significant restriction but this are clearly needs more experimentation and input from other working groups.

5. The Use of SDP

5.1. SDP announcement by periodic multicast

The sd tool announces multicast sessions by periodic multicast of SDP session descriptions on a well known multicast address and port (224.2.127.255, port 9876). The session announcements should be multicast with the same TTL with which the conference session will be multicast. If the different media in an announcement are given different TTLs, then multiple announcements are necessary to ensure that anyone joining the conference can in fact receive data for each media started. For example, if we have an announcement to make containing audio at ttl 127 and video at ttl 63, then we make an announcement at ttl 63 containing both media, and a separate announcement at ttl 127 containing only the audio. It is up to the receiving sd to parse both announcements as the same announcement if it is within the appropriate scope to get both announcements. (If there are multiple announcements for the same session, the receiving sd should simply take the largest.)

The time period between one announcement and its repetition is dependent on two factors - the scope (TTL) or the session, and the number of other sessions currently being announced by other sd clients.

The recommended bandwidth limits for each TTL are:

TTL	bandwidth
<16	1Kbps
16-47	500bps
48-63	200bps
64-127	80bps
128-191	30bps
192-255	10bps

The figures above represent target total bandwidths for all announcements at the specified TTL and above. Thus, if there were no announcements of a session with a TTL above 191, the announcements of sessions with TTLs in the range 128-191 could take 30 bps. If announcements of sessions with TTLs over 192 were already using 10bps, then those in the range 128-191 are only permitted an additional 20bps.

Session announcers in the same scope band can normally be expected to hear your announcements, and reduce their data rates accordingly. Thus you should calculate the available bandwidth for your session's scope band by subtracting from the figure above the bandwidth being used by scope bands with a higher TTL, and then divide the resulting figure by the number of other announcers in your scope band. This gives you your bandwidth allocation, which, given the size of your data packets, can be used to derive the base interval for announcements. For each announcement, you must add a random value (+/- 1/3 of the base interval) to the value used for the inter-announcement period to prevent announcement synchronisation. It is also important to keep monitoring other announcements and adjust the base interval accordingly.

For TTL 63 or above session announcements with 200 bytes in each announcement, these figures give a mean inter-arrival time of 20 seconds between announcement arrivals. Obviously the inter-arrival time *per session* is dependent on the number of other sessions being announced.

5.2. SDP announcement by email and WWW

It should be stressed that announcements made only via email or the World Wide Web do not allow the scalable booking query mechanism as discussed above, and so should normally be used to supplement periodic multicast announcements.

For both email and WWW distribution, the use of the MIME content type ‘‘`application/x-sd`’’ is suggested. This enables the automatic launching of applications from the WWW client or mail reader in a standard manner.

Appendix A: Summary of differences between SDPv1 and SDPv2

For this purpose SDPv1 is defined as the protocol in use by version 1.14 of the sd session description tool. SDPv2 is the proposed protocol described in the document.

SDPv1 allows only the use of the following SDP fields:

`s=` **session name -** no change

`i=` **session information** - no change

`o=` **originator -** no change

`c=` **conference data** - only one conference data field is allowed before the first media field in SDPv1. SDPv1 incorrectly uses NTP timestamps. To convert from SDPv1 timestamps to UNIX time, subtract decimal 2085978496. To convert from SDPv2 timestamps to UNIX time, subtract decimal 2208988800.

`m=` **media** - no change

`a=` **attributes** - attributes are only allowed after the first media field in SDPv1.

SDPv2 additionally defines the following fields which are defined above:

`b=` **Bandwidth**

`e=` **Email Address**

`p=` **Phone Number**

`u=` **URL**

`h=` **Originating Host Address**

`t=` **Repeat Time**

`k=` **Encryption Key**

Appendix B: Authors Addresses

Mark Handley
Department of Computer Science
University College London
London WC1E 6BT
United Kingdom
electronic mail: M.Handley@cs.ucl.ac.uk

Van Jacobson
MS 46a-1121
Lawrence Berkeley Laboratory
Berkeley, CA 94720
United States
electronic mail: van@ee.lbl.gov

References

[1] D. Mills, ‘‘Network Time Protocol version 2 specification and implementation", RFC1119, 1st Sept 1989.

[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, ‘‘RTP: A Transport Protocol for Real-Time Applications’’, INTERNET-DRAFT, draft-ietf-avt-rtp-07.txt, 21st March 1995.

[3] M. Handley (editor), ‘‘The Use of Plain Text Keys for Multimedia Conferences’’, Research Note RN-95-19, Department of Computer Science, University College London, Feb 1995. http://www.cs.ucl.ac.uk/rns/RN9519.ps

[4] V. Jacobson, S. McCanne, ‘‘vat - X11-based audio teleconferencing tool’’ vat manual page, Lawrence Berkeley Laboratory, 1994.

[5] S. McCanne, ‘‘vic - video conference’’, vic manual page, Lawrence Berkeley Laboratory,

1994.

[6] W. Dang, ‘‘imm - Image Multicaster Client’’, imm manual page, University of Hawaii, 1993.

[7] ‘‘The Unicode Standard, Version 1.1’’: Version 1.0, Volume 1 (ISBN 0-201-56788-1), Version 1.0, Volume 2 (ISBN 0-201-60845-6), and "Unicode Technical Report #4, The Unicode Standard, Version 1.1" (available from The Unicode Consortium, and soon to be published by Addison- Wesley).

[8] ISO/IEC 10646-1:1993(E) Information Technology--Universal Multiple-octet Coded Character Set (UCS).

[9] D. Goldsmith, M. Davis, ‘‘Using Unicode with MIME’’, RFC1641, July 1994

[10] D. Goldsmith, M. Davis, ‘‘UTF-7 - A Mail-Safe Transformation Format of Unicode’’, RFC1642, July 1994