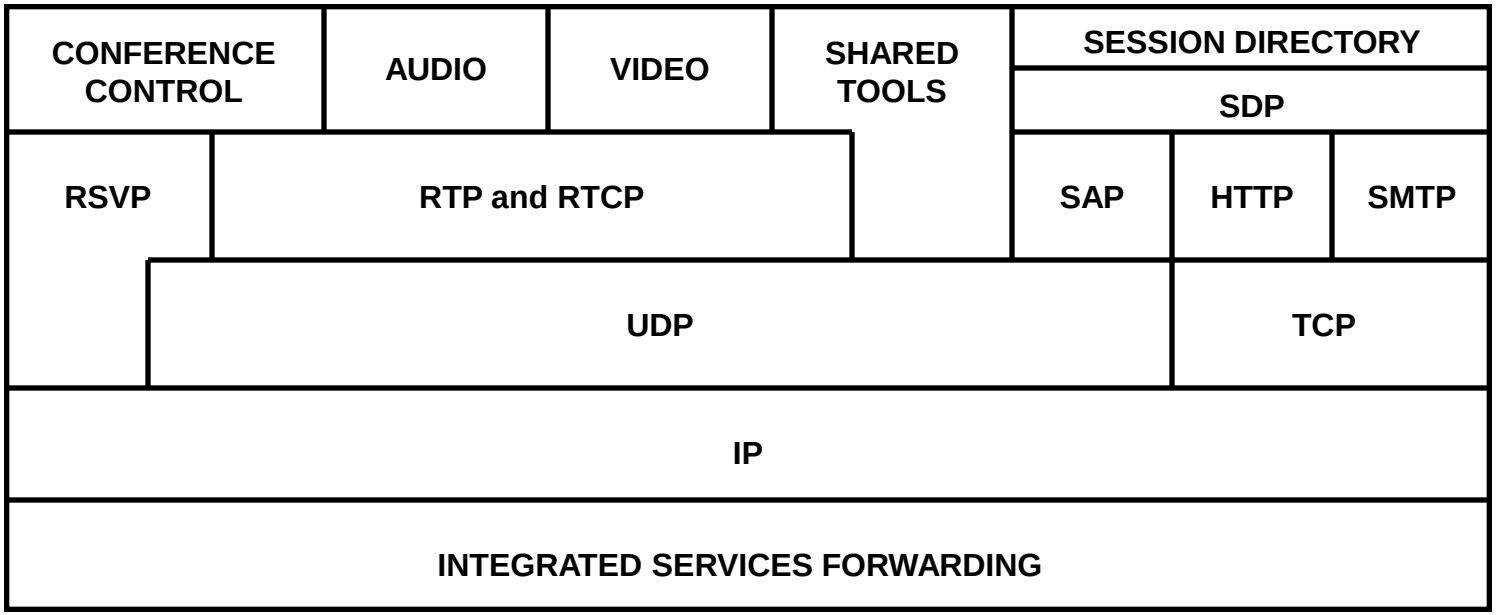
# The Internet Multimedia Conferencing Architecture

# The Internet Multimedia Conferencing Architecture

- Product of the Multiparty Multimedia Session Control working group (MMUSIC)
- For synchronous "real-time" conferencing, including audio, video and shared whiteboards
- Architecture is general and scalable to very large groups
  - permits the open introduction of new media and new applications as they are devised

## Internet Multimedia Conferencing

# Protocol Stacks



# Multicast Traffic Description (1)

- ***IP multicast service model***
  - sources send datagrams to the address of a multicast group (class D addresses)
  - receivers express an interest in (join) certain multicast groups (using IGMP)
  - multicast routers work together to deliver datagrams with multicast group addresses from the senders to the receivers (e.g., DVMRP, MOSPF, CBT, PIM)

# Multicast Traffic Description (2)

- senders do not have to know who the receivers are:
  - no single point in the network needs to know who all the receivers are; hence, IP Multicast is scalable to very large groups
- receivers do not need to know who the senders are:
  - no need for conference set up and resource location mechanism
- need a mechanism for allocating group addresses dynamically, and a directory service which users can look up
  - distributed mechanism based on a hash function for forming initial random values for the address
  - directory service advertises conferences through multicast messaged on a well-known multicast address

# Transport Protocols (1)

## 1. Separate flows for each media stream

- simpler for receivers
- different media can be given different quality of service (e.g., in case of congestion, preferentially drop video packets over audio packets)
- some sites may not wish to receive all the media flow (e.g., a site with slow access link may participate using only audio and a whiteboard)

## 2. Receiver adaptation

- buffering to remove the jitter added by the network and recover the original timing relationships between media data
- each packet must carry a timestamp

# Transport Protocols (2)

## *3. Synchronization*

- requires that the time base for different flows from the same source can be related at the receivers
- making available the absolute times at which each flow was captured

## *4. RTP*

- provides a standard format packet header, including
  - media specific timestamp data
  - payload format information
  - sequence numbering
  - source ID (as well as a list of source Ids of all contributing sources in case of mixing)

# Transport Protocols (3)

- RTCP packets
  - relationship between the real-time clock at a sender and the RTP media timestamps
  - textual information to identify a sender in a conference from the source ID

## *5. Conference membership and reception feedback*

- RTCP session messages provide approximate membership information
- RTCP reception quality reports
  - useful for rate adaptation at sender

# Conference Control

- Two models
  - light weight sessions:
    - multicast based multimedia conferences that lack explicit session membership and explicit conference control mechanisms
    - RTCP session information provides approximate membership list
  - tightly coupled sessions
    - explicit conference membership and explicit conference control mechanisms (flow control protocol)
    - protocol based on ITU T.124 may be developed

# Conference Discovery (1)

- Two basic forms of the conference discovery mechanism:
  - session advertisement (session directories)
  - session invitation
- Session directories:
  - rendez-vous mechanism for light-weight sessions
    - multicast based session directory
    - distributes session descriptions to all potential session participants
      - advertisement that session will exist
      - sufficient information (multicast addresses, ports, media formats, session times)
      - session description protocol (SDP)

# Conference Discovery (2)

- appropriate agent to perform multicast address allocation
  - session announcement protocol (SAP)
- may be also be used to advertise tightly coupled sessions
  - requires additional information about the mechanism to use in order to join session

- Session invitation:
  - not all sessions are advertised
  - even those sessions that are advertised may require a mechanism to explicitly invite a user to join a session
  - session invitation protocol (SIP)

# SDP

*Session Description Protocol*
*RFC 2327*

# SDP

- In the context of a multimedia multicast Internet, a session directory tool is used to advertise multimedia conferences and communicate the conference addresses and conference tool-specific information necessary for participation.

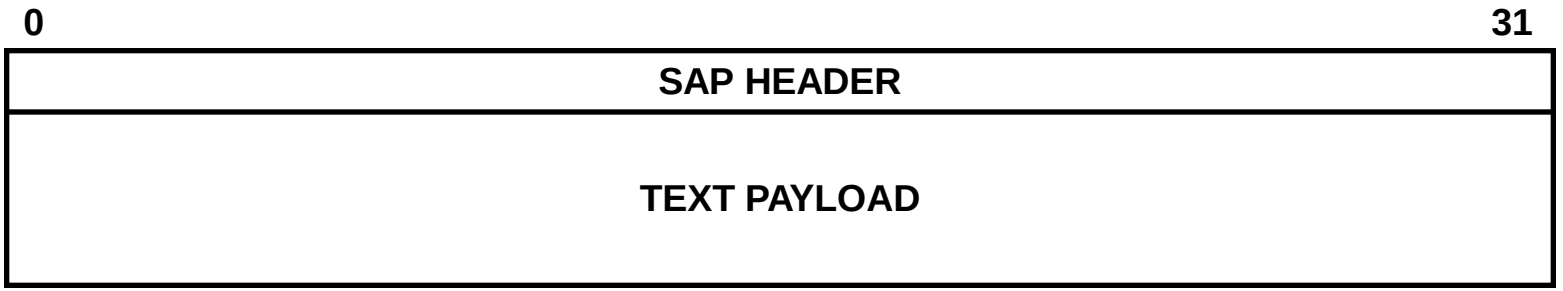
- SDP is designed to convey such information. It does not incorporate a transport protocol but uses different transport protocols as appropriate including:
  - SAP (Session Announcement Protocol)
  - Email using MIME extensions
  - WWW (HTTP - HyperText Transport Protocol)

# Definitions

- ***Conference***: a two or more communicating users along with the software they are using to communicate
- ***Session***: a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session.
- ***Session Announcement or Advertisement***: a mechanism by which a session description is conveyed to users in a pro-active fashion, i.e. the session description is sent without being explicitly requested by users
- ***Session Description***: a well defined format for conveying sufficient information too users for them to discover and participate in a multimedia session.

# SDP Usage (1)
# Multicast Announcements

- A user announces a conference by periodically sending an SDP packet to a well-known *multicast* address and port using the Session Announcement Protocol.
- SAP packets are UDP packets containing a header and a text payload.
- The header is a SAP header as described in RFC 2974.
- The text payload is an SDP session description:
  - should not exceed 1 kbyte in length
  - only one session description is allowed per SAP packet



# SDP Usage (2)
# Email and WWW Announcements

- MIME content type should be set to "application/sdp" in order for the WWW client or mail reader to automatically launch the appropriate application for participation in the session.
- Announcements by Email or WWW do not guarantee that the user is able to receive the session as the scope of the session may be more limited than Email reception or WWW access.

# Session Information

- The session information carried in SDP messages includes:
  - session name and purpose
  - time(s) the session is active. Note that a session is defined as a set of multimedia streams that exist for some duration of time, the times these streams are active need not be continuous
  - the media comprised in the session
  - information to receive these media (addresses, ports, formats...etc)
  - additional information that might be useful such as:
    - information about the bandwidth to be used by the conference
    - contact information for the person responsible for the conference

# Timing Information

- Sessions may be bounded or unbounded in their time duration, however they may only be active at specific times.
- SDP conveys:
  - an arbitrary list of start and stop times bounding the session active time
  - for each bound, repeat times are indicated such as “every Tuesday at 2:45 PM”
- The timing information is globally consistent irrespective of time zone or daylight saving time

# Media Information (1)

- The SDP includes the following information for each type of media:
  - type (audio, video, …)
  - transport protocol (RTP, UDP, IP, H.320, …)
  - format (H.261, MPEG1, G.711, …)
  - for IP multicast sessions, the following are specified:
    - destination multicast address of stream
    - destination transport port of stream

# Media Information (2)

- for IP unicast sessions, the following are specified:
  - remote address
  - transport port for contact address
  - semantics of the address and port depend on the type of media and transport protocol defined, default is remote port and remote address to send data to, and remote address and local port for receiving data. Some media types may choose to use the address and port to establish a control channel for the actual media transfer.

# Other Issues

- ***Private sessions***
  - encrypt the session description before distributing it
  - private session announcements may convey encryption keys for the decoding of the different media types that constitute the session including indication of the encryption scheme used for each media type
- ***More information about the session*** can be conveyed through additional pointers in the form of Universal Resources Identifiers (URIs)
- ***Categorization*** of sessions is supported by SDP . This allows filtering of session announcements to be performed according to user interests
- ***Internationalization*** is supported by allowing the usage of character sets other than that used for coding extended ASCII characters.

# Session Description

- SDP session descriptions are entirely textual. This allows the use of a variety of transport methods and of flexible text-based toolkits.
- A compact encoding scheme is used to reduce the bandwidth usage
- SDP has a strict order and formatting rules that allow detection of errors which result in the malformating of announcements
- A session description consists of a session-level section (that apply to the whole session and to all media streams) followed by zero or more media-level descriptions (details that apply to the particular media stream)

# Session Description Syntax (1)

- The session description syntax consists of number of text lines of the form:

***type=<value>***

  - no space between the = sign and the other fields is allowed
  - *type* field is one character and case significant
  - *value* is a free-format text string or a series of text strings separated by single space characters.
  - the session-level description starts with a "***v=***" line and ends at the first media level section encountered.
  - the media-level description starts with a "***m=***" line and ends at the next media level section encountered or at the end of the whole session description. A value at media-level overrides that at the equivalent session-level value for the particular media type.

# Session Description Syntax (2)

- When SDP is carried by SAP, only one session description is allowed per packet.
- When Email or WWW is used to transport SDP information, many SDP session descriptions may be concatenated together (the "***v=***" line that starts a session description ends the previous one).
- Some lines in the description are required but other are optional, but *all must appear* in the *exact order given* (an '*' next to an item indicates that it is optional).

# Session Description Syntax (3)

- Session description
  - *v=* protocol version
  - *o=* owner/creator and session number
  - *s=* session name
  - *i=** session information
  - *u=** URI of description
  - *e=** email address
  - *p=** phone number
  - *c=** connection information - optional if included in all media
  - *b=** bandwidth information
  - *>> one more time descriptions*
  - *z=** time zone adjustments
  - *k=** encryption key
  - *a=** zero or more session attribute lines
  - *>> zero more media descriptions*

# Session Description Syntax (4)

- Time description
  - *t*= time the session is active
  - *r*=* zero or more repeat times
- Media description
  - *m*= media name and transport address
  - *i*=* media title
  - *c*=* connection information - optional if included at session level
  - *b*=* zero or more repeat times
  - *k*=* encryption key
  - *a*=* zero or more session attribute lines
- The *type* set is small and not intended to be extensible, SDP parser must ignore any announcement type that they do not recognize

# Session Description Syntax (5)

- The attribute mechanism is used to tailor SDP to particular applications or media.
  - some attributes are specified in the RFC and have a defined meaning, but others may be added on an application-, media- or session-specific basis. A session directory must ignore any attribute value it does not understand.
- Text records (such as the session name and information) may contain any printable ISO 8859-1 (the character set supporting extended ASCII) character except 0x0A (newline) and 0x0D (carriage return). The first is used as end of record and the second is forbidden.

# Session Description Syntax (6)

- Example session description:

```
v=0
o=noronha 2890844526 2890842807 IN IPv4 171.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.stanford.edu/home/noronha/sdp.03.ps
e=noronha@stanford.edu (Ciro Noronha)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 3456 RTP/AVP 0
m=video 2232 RTP/AVP 31
m=whiteboard 32416 udp wb
a=orient:portrait
```

# Session Description Syntax (7)

- The different line syntaxes are:
  - v: SDP version number
  - o: originator information

    **<username> <sessionID> <version> <network type> address type> <address>**
  - s: must have one and only per announcement

    **<session name>**
  - i: labeling information, no more than one for session-level. a single I field can also be used for every media-level description

    **<session description>**
  - u: Universal Resource Identifier as used by WWW clients, pointer to additional information about the conference. No more than one per session description

    **<URI>**

# Session Description Syntax (8)

- e: email address of person responsible for the conference

  **<email address>**
- p: phone number of person responsible for the conference

  **<phone number>**
- c: connection data

  **<network type> <address type> <connection address>**
- b: bandwidth, modifier value is CT (conference total) or AS (application specific maximum) or a experimental modifiers (should start with **X-**)

  **<modifier>:<bandwidth-value>**
- t: start and stop time of conference session

  **<start time> <stop time>**
- r: repeat times for a session

**<repeat interval> <active duration> <list of offsets from start time>**

# Session Description Syntax (9)

- z: list of time zone and daylight savings adjustments

  **<adjustment time> <offset> <adjustment time> <offset>...**
- k: encryption keys

  **<method>:<encryption keys>**
- a: attributes

  **<attribute>:<value>**
- m: media announcements

  **<media> <port> <transport protocol> <media format list>**

# Session Announcement Protocol

RFC 2974

# SAP
# Session Announcement Protocol

- A session directory is used to advertise multimedia conferences and communicate:
  - session addresses (multicast or unicast)
  - conference-tool specific information
- An instance of such a session directory periodically multicasts packets containing a complete description of a multimedia session
- SAP is an announcement protocol for multicast conference sessions.
  - distribution mechanism
  - packet format

# SAP (1)

- SAP client periodically multicasts an announcement packet to a well-know multicast address and port
- Announcements should be sent with TTL=255.
- Time period between announcements depends on
  - number of sessions being announced by other session directory clients
  - goal: keep total bandwidth below predefined level

# Time Interval Between Announcements

- Base time interval = max(300, (8 * no_of_ads * ad_size)/limit)
- Add a random value +/- 1/3 of base interval
- Bandwidth limit defaults to 4000 bits/second for a given SAP group, but other values can be used as well.

# SAP (2)

- SAP functions
  - session announcement
  - session deletion
  - session modification
- Session announcement contains
  - session description (may be encrypted)
  - authentication header

# SAP (3)

- Session deletion
  - explicit timeout
    - session description contains start and end times for each session announcement
  - implicit timeout
    - SA message should arrive periodically. The period may be predicted by the receiver from the set of sessions currently being announced
    - lack of SA message after max(10 intervals, 30 minutes)
  - explicit deletion
    - session deletion packet specifying the version of the session to be deleted
    - same IP source address as that from which the session announcement was originally advertised
    - if authentication header is cached, the deletion packet must have a signature with the same key

# SAP (4)

- Session modification
  - by announcing modified description
  - version hash in SAP header could be changed to indicate packet must be parsed
  - session itself is uniquely identified by the SDP origin field in the payload and not by the version hash in the SAP header
  - same rules as for session deletion regarding authentication and source address information
    - if not satisfied, modified announcement is displayed in addition to the preexisting announcement

# SDP Announcement by Periodic Multicast

- Appropriate address determined by scope mechanisms in force at the sites of the intended participants
- Well-known address and port defined (address 224.2.127.254, port 9875)
- Administratively scoped
  - one well known address (within the corresponding scope zone) and port used for each administrative scope zone
  - The highest multicast address in the zone should be used

# SDP Announcement (2)

- An instance of the session directory should
  - listen to multiple multicast addresses
  - normally only need to send a particular announcement to the single multicast address corresponding to the scope of the session being described
- Discovery of administrative scope zones is outside the scope of the SAP RFC. However, it is assumed that each instance of the session directory within a particular scope zone is aware of that scope zone address, port, TTL, and session addresses allocated.

# Other Matters

- Encrypted announcements
  - encrypting announcements
  - decrypting announcements
- Security considerations

# Session Initiation Protocol (SIP)

RFC 2543

# Session Initiation Protocol

- SIP is a protocol designed to enable the invitation of users to participate in multimedia session
  - not tied to a specific conference control scheme
  - supports loosely or tightly controlled sessions
  - enables user mobility by relaying and redirecting invitations to a user's current location

# Call Setup

- Initial phase
  - requesting client tries to ascertain the address where it should contact the remote user
- Subsequent phases
  - implement request-response protocol
  - session description is sent with an invitation to join
- Status code responses
  - informational - request received, continuing process
  - success - action successfully received, understood, and accepted
  - redirection - further action must be taken in order to complete the request
  - client error - request contains bad syntax
  - server error - server failed to complete an apparently valid request