Multiparty Multimedia Session Control Working Group

68th IETF – Prague

19 March 2007

Please join the Jabber transcription experiment:
mmusic@jabber.ietf.org

Intellectual Property

- When starting a presentation you MUST say if:
  - There is IPR associated with your draft
  - The restrictions listed in section 5 of RFC 3978/4748 apply to your draft
- When asking questions or commenting on a draft:
  - You MUST disclose any IPR you know of relating to the technology under discussion
- References:
  - RFC 3978 (updated by RFC 4748), and RFC 3979
  - “Note Well” text

Agenda

17:40	Introduction and progress report	(Chairs)
17:45	ICE	(Rosenberg)
18:30	ICE-lite	(Rescorla)
18:40	SDP Capability Negotiation	(Andreasen)
19:10	SDP Media Capability Negotiation	(Even)
19:20	Media decoding dependency	(Schierl)
19:25	Source-Specific Media Attributes	(Lennox)
19:35	The MSR Bandwidth modifier in SDP	(Desineni)
19:40	Media Description for IKE in SDP	(Saito)
19:45	Configuring DiffServ using SDP	(Polk)
19:50	End	

Introduction and Progress Report

Ott/Perkins/Mulé

Working Group Status

- Published
  - draft-ietf-mmusic-sdp-bfcp-03.txt → RFC 4583
  - draft-ietf-mmusic-fec-grouping-05.txt → RFC 4756
  - draft-ietf-mmusic-sdp-media-content-06.txt → RFC 4796
- With RFC Editor:
  - draft-ietf-mmusic-sdpng-trans-04.txt
    (Revised ID Needed?)
- With IESG:
  - draft-ietf-mmusic-securityprecondition-03.txt
    (Revised ID Needed per ID Tracker)

Working Group Status

- Done:
  - draft-ietf-mmusic-connectivity-precon-02.txt
    - Waiting for ICE
  - draft-ietf-mmusic-ice-14.txt?
  - draft-ietf-mmusic-sdp-capability-negotiation-reqts-01.txt?
  - draft-ietf-mmusic-sdp-capability-negotiation-05.txt?
- Dead:
  - IMG related drafts
  - RTSP/2.0?
    - Need help to finish this!
    - Three individuals expressed interest in reviewing and helping resolve open issues; review team will be formed in April 2007 with regular calls

Future Directions

- **Finish ICE, ICE-lite, and ICE-TCP!**

- Finish media capability negotiation
  - Requirements considered done
  - Base specification close to done
    - Tie in to best effort SRTP and RTPSEC BOF
  - Media negotiation draft proposes significant extensions
    - How much flexibility/complexity do we want?

- Advance other documents in wg charter
- Re-charter to consider future directions

ICE

Rosenberg

draft-ietf-mmusic-ice-14.txt

draft-ietf-mmusic-ice-tcp-03.txt

Changes since -13

- Countless organizational and wording changes to improve readability (thanks Ekr, Dan)
- Spec was inconsistent on when you include candidates
  - All but port=0?
  - All but a=inactive?
  - Now clear that its all but port=0
- Included note saying the first m-line should be most important – its checked first
- For STUN servers – if you learn them through DNS, use the same one for all queries for this session
  - Makes sure the foundations are identical – otherwise frozen algorithm is suboptimal

Changes since -13

- Added error case: if TURN query fails during gathering, revert to regular STUN
- Changed a=inactive vs. 0.0.0.0 from SHOULD to MUST
  - ICE really doesn’t work with 0.0.0.0
- Not requiring subsequent offers to contain peer-reflexive candidates that you have learned
  - Will slow down ICE if you include them and don’t otherwise want to
  - Only reason to include is some really corner topology cases identified by Philip

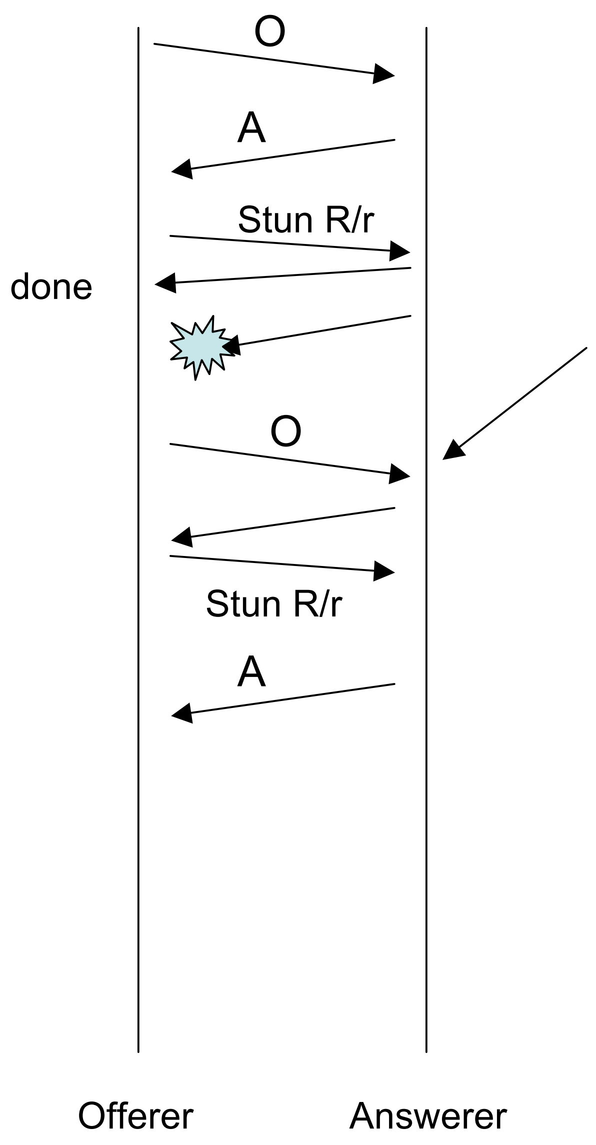
Changes since -13

- When you send an updated offer in Completed state, now you MUST only include your selected candidate pairs
  - Used to be SHOULD – but there is never a reason to send anything but
- Fixed race condition:
  - Controller sends USE-CANDIDATE, and is using the aggressive mode
  - Controlled party stops retransmitting all checks
  - Packet loss causes inconsistent view on whether a higher priority check (that had USE-CANDIDATE) succeeded
  - Fix is: only cease retransmits on lower priority pairs

Changes since -13

- If a controller uses an aggressive algorithm, once one is selected, it waits 1 second before updated offer
  - Might be transient selected pairs, gives it time to stabilize
  - Updated offer is not needed to send media, just fixup for intermediaries, so waiting as long as 1 second is fine
- Biggest change: delayed answer in remote-candidates race case

Race Flow



This arrives with an a=remote-candidates. If there is a symmetric NAT between offerer and answerer, the value in there may not yet be known to the answerer, so it cannot include a candidate attribute in the answer. This used to not matter – omit it – but now we require a match of candidates and m/c-line to deal with SBC. So answerer needs to delay answer till check completes. Note this has no impact on media delays.

Changes since -13

- Added text hinting how RTP/RTCP mux would work
- Documented how ICE and ANAT work together

Recent Bug not Addressed in -14

- ICE assumes you always have one agent as controller, one as controlled
- This is true for ‘normal’ cases and for the four 3pcc flows in RFC 3725
- However, there are 3pcc and application flows that end up with:
  - Both ends thinking they are controller
  - Both ends thinking they are controlled
- ICE will currently fail badly in both cases
  - In the former, disagreement on selected candidates and glared invites
  - In the latter, non-conclusion on ICE

Fortunately, fix is easy

- For both agents as controller
  - If a controlling agent gets a USE-CANDIDATE in a STUN request it receives, we’ve detected this
  - Solution is: reject request and make controller/controlled decision based on tie-breaker (largest ufrag wins controller role). Proceed.
- For both agents as controlled
  - Checks will complete and there is never a USE-CANDIDATE
  - Solution is: 500ms after completion of all checks, if you don’t get a use-candidate, make a tie-breaker decision as above. Proceed.

ANAT

- There is definite overlap between ICE and ANAT, and also SDP capability negotiation
- ICE Alone
  - For each media stream include v4 and v6 candidates
  - Prioritize them as needed (v6 first, then v4)
  - Can include multiple v6 candidates
  - Selection will be based on priority and connectivity
  - In cases where v6 path is broken, will fall back to v4
  - Requires a re-invite to ‘fix-up’ SDP if v6 is selected and v4 was in m/c-line
  - Can use custom selection algorithms to take delay into consideration also
- ANAT Alone
  - Include two m-lines and a Require header in INVITE
  - Select v6 if other side is v6-capable, else v4
  - Reinvite needed if remote side doesn’t support ANAT

ANAT

- ICE+ANAT
  - Include two m-lines, one for v4, one for v6
  - V4 m-line includes v4 candidates
  - V6 m-line includes v6 candidates
  - Choose highest priority v6 candidate if one works, else highest priority v4 candidate
  - Re-invite required if remote side doesn’t support ANAT, and also if selected candidate doesn’t match m/c lines
- SDP Cap negotiation
  - Include v6 IP address as a high priority capability
  - Will be similar to ANAT but better in that it allows for graceful fallback without a Require header field
- SDP Cap Negotiation + ICE?
  - Could use ICE to drive connectivity-based selection of potential configs
  - But this is really complicated – no clear need

ANAT

- Question:
  - What is our recommendation for dual-stack hosts to do v4/v6 selection?
    - ANAT alone
    - ICE alone, deprecate ANAT
    - ICE+ANAT
    - SDP Caps, deprecate ANAT
  - Where is this recommendation documented?
- Proposal
  - Transport address selection is always better dynamically than statically
  - Have ICE obsolete ANAT, remove section on ANAT interaction
    - Note that no other changes are required – ICE always allowed for multi-address candidates
  - Document v4/v6 application in the SIPPING transition document

ICE-lite

- Need to make sure we are happy with the way ICE and ICE-lite relate
- My view is:
  - All normative text for supporting ICE-lite is in ICE, it cannot be separate
  - ICE-lite spec serves as an informational document to assist ICE-lite implementors find what parts of ICE they need to read by explaining things and pointing into ICE

ICE-TCP

- Major changes in this draft
- Removed TLS entirely
  - ICE is about transport connections
  - Whether a TCP connection is used for TLS or not is signaled in m-line
  - If you want to negotiate TCP or TLS use our new capability negotiation draft, not ICE
- ICE-TCP requires a full implementation of ICE, not lite
- If default is UDP, and a TCP candidate is meant as a TLS alternate, include fingerprint attribute
- If a TCP candidate is for TLS, the fingerprint attribute is present in the initial offer
  - This allows you to verify fingerprints immediately
  - Previous spec only did this in re-invite – introducing clipping for TLS. No longer

ICE-TCP

- Previously, from each simultaneous-open and passive local candidate, you'd obtain a relayed candidate
  - Unclear whether relayed candidates were s-o or passive
  - And uneccesary to obtain relayed from BOTH s-o and passive local – both have same bases
  - So, fixed this
- Relationship to comedia now clear
  - Comedia attributes included only when default candidate is tcp, to allow backwards compatibility with non-ICE peers
  - Comedia is not used in any way with ICE itself
- Regular selection (not aggressive) is required for TCP candidates

ICE-TCP

- ICE restart interaction with connection management is clarified
  - Keep selected connection open
  - Redo checks
  - If previous connection is re-selected, continue using, else close
- Connection management overall clarified
  - Can close a connection at any time – just reopen and reuse without ICE checks
    - Bug in the spec, I think – what if connection creates new IP/ports. Perhaps we need STUN?
  - If connection cannot be reopened, do a manual restart of ICE

ICE-TCP Issues

- Issue #1: If S-O TCP is used with TLS, who sends ClientHello?
- Choices
  - Define new tls-act/pass/actpass attribute like comedia’s
    - But applies only to TLS
  - Controlling agent always sends
    - Won’t work in cases where one end is a gateway or other device that has a cert, and other side doesn’t
  - Reuse comedias attributes
    - Semantics are subtely different though
- Proposal
  - Define new attributes and add to ice-tcp

ICE-tcp Issues

- Issue #2: Can we use session resumption with TLS when connections close, and if so, how to signal?
- Answer:
  - Apparently yes, without any documentation
  - Proposal: just remove open issue text, add a comment that resumption will get used

Plan of Action

- Issue ICE-15 with
  - List comments on -14 (all minor)
  - Bug fix for controller/controlling mismatches
  - ANAT resolution
- ICE-15 gets issued this week or some dire punishment gets laid on me by the group
  - Removal as editor
  - Banned from submitting new drafts
  - Owe Colin/Joerg/J-F an expensive bottle of wine
  - Etc.
- ICE-tcp update shortly. Are we ready for WGLC?
- Issue WGLC on ICE-15 immediately upon issuance
  - And a request to group: Please do not nitpick this draft to death with WGLC comments
  - It has had an amazing amount of review and revision already, it needs to be finished more than anything else

ICE-lite

Rescorla

draft-rescorla-mmusic-ice-lite-00.txt

draft-rescorla-mmusic-ice-lite-00

- In San Diego we agreed we would do an “ice for gateways”
- Assumptions:
  - You have a public IP address
  - You don’t want to bother with timers, check lists, etc.
  - But you do want to do ICE
- You also don’t want to bother reading the entire ICE spec
- This document attempts to fill that need

Overall strategy

- Provide an overview of how ICE-lite works
- For each feature:
  - Provide an overview of the ICE-lite behaviors
  - Reference the specific ICE sections you need to implement
    - This means the document does not stand alone
- Open issues
  - Should we provide complete descriptions so you don’t have to read ICE at all?
  - Should this be the normative description of ICE-lite or is that in ICE?

SDP Capability Negotiation

Andreasen

draft-ietf-mmusic-sdp-capability-negotiation-05.txt
draft-ietf-mmusic-sdp-capability-negotiation-reqts-01

IPR Statement

- Cisco is the owner of one or more pending unpublished patent applications relating to the subject matter of "SDP Capability Negotiation" <draft-ietf-mmusic-sdp-capability-negotiation-05.txt>.
  - See https://datatracker.ietf.org/public/ipr_detail_show.cgi?&ipr_id=761
- If technology in this document is included in a standard adopted by IETF and any claims of any Cisco patents are necessary for practicing the standard, any party will have the right to use any such patent claims under reasonable, non-discriminatory terms, with reciprocity, to implement and fully comply with the standard.
- **The reasonable non-discriminatory terms are:**
  - If this standard is adopted, **Cisco will not assert any patents** owned or controlled by Cisco against any party for making, using, selling, importing or offering for sale a product that implements the standard, **provided, however that Cisco retains the right to assert its patents (including the right to claim past royalties) against any party that asserts a patent it owns or controls (either directly or indirectly) against Cisco** or any of Cisco's affiliates or successors in title or against any products of Cisco or any products of any of Cisco's affiliates either alone or in combination with other product. Cisco retains the right to assert its patents against any product or portion thereof that is not necessary for compliance with the standard.
- **Royalty-bearing licenses will be available to anyone who prefers that option.**

Since Last IETF

- Formed Design Team consisting of
  - Roni Even, Robert Gilman, Matt Lepinski, Joerg Ott, Colin Perkins, Thomas Stach, and Flemming Andreasen
- Additional input and feedback from
  - Francois Audet, John Elwell, Cullen Jennings, and Dan Wing
- Weekly conference calls from 12/12/06 to 2/27/07
  - Adopted <draft-andreasen-mmusic-sdp-capability-negotiation-01.txt> as starting point
  - Split document in two
    - Requirements: Initial design team focus
    - Solution document: Based on requirements document

Documents

- Requirements
  - draft-ietf-mmusic-sdp-capability-negotiation-reqts-01.txt
  - Main guiding principles
    - Provide a general purpose capability negotiation mechanism
    - Avoid undue complexity or scope creep for people that need only negotiation of transport protocols and associated attributes
  - Stabilized pretty quickly; divided requirements into
    - Core: All implementations must support
    - Extensions: Optional to implement
  - Ended up moving media requirements to "extensions" in order to have a small and focused "core"
    - More rapid progress towards mature solution
- Ended up with two solution documents per the above
  - SDP Capability Negotiation
    - draft-ietf-mmusic-sdp-capability-negotiation-05.txt
    - This is the "core" document
  - SDP Media Capabilities Negotiation:
    - draft-ietf-mmusic-sdp-media-capabilities-01.txt
    - An "extension" document

SDP Capability Negotiation (Core)

- Provides capabilities for the following
  - Attributes
  - Transport protocols
- Allows for those capabilities to be used in SDP Capability Negotiation
  - Potential Configurations provided in offer SDP
    - Each potential configuration references one or more capabilities
  - Answerer may pick one of the potential configurations from the offer and negotiate use of it as the actual configuration
    - Actual configuration chosen indicated in answer SDP
- Extensions supported via option-tags that can be required or just listed as supported

Open Issues in Core

- There aren’t any known open issues in the spec per-se, but the following is worth discussing
  - Add, replace, delete semantics for attributes
  - Inconsistent media-level selection of session-level capabilities

Background Information

- Conceptually, potential configurations are formed by purely syntactical operations on an offer SDP:
  - For each potential configuration, create a "potential configuration SDP" consisting of
    1. The original SDP
    2. For transport protocol capabilities included, <u>replace</u> the transport protocol (e.g. "RTP/SAVP") in the "m=" lines
    3. For attribute capabilities included, <u>add</u> the attributes to the original SDP
  - Use this potential configuration SDP as the potential offer

<u>Actual Configuration SDP</u>

```
v=0
o=- 25678 753849 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 53456 RTP/AVP 0 18
a=tcap:1 RTP/SAVP RTP/AVP
a=acap:1 a=crypto:1 AES_CM_...
a=pcfg:1 t=1 a=1
```

<u>Potential Configuration SDP</u>

```
v=0
o=- 25678 753849 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 53456 RTP/SAVP 0 18
a=crypto:1 AES_CM_...
```

To Add, Replace or Delete Attributes

- Problem with just adding attributes
  - There may be attributes in the potential configuration SDP that redefine an existing attribute value
    - Example: An “a=rtpmap” that assigns a different codec to a given payload type
  - There may be attributes in the potential configuration SDP that contradict the added ones
    - Example: Direction attributes (“sendonly” and “sendrecv”)
  - There may be attributes in the potential configuration SDP that overlap the added ones
    - Example: key-mgmt and crypto attributes (use only one)
  - There may be attributes in the actual configuration SDP that you don’t want in a particular potential configuration SDP
    - Example: ICE candidates when you select a transport protocol that you do not want to run over UDP or TCP (e.g. T.38 over TCP versus PCMU over RTP)
- We do not want the SDP Capability Negotiation layer to try and sort these out
  - SDP Capability Negotiation would need to be aware of all SDP attribute, etc. semantics
  - Extensibility (e.g. new attributes) would be problematic

To Add, Replace, or Delete Attributes

```
v=0
o=- 25678 753849 IN IP4 192.0.2.1
s=
t=0 0
c=IN IP4 192.0.2.1
a=key-mgmt:mikey AQAFgM0XflA...
a=acap:1 a=key-mgmt
a=tcap:1 RTP/SAVP RTP/AVP RTP/AVPF
m=audio 59000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
a=candidate:1 1 UDP 2130706178
a=candidate:2 1 UDP 1694498562
a=candidate:3 1 tcp-act 1706178213
a=acap:2 a=crypto:1 AES_CM_128_...
a=pcfg:1 t=1 a=-1,2
a=pcfg:2 t=2 a=-1
a=...
```

If RTP, then don't want

If SRTP, then don't want both

If RTP/AVPF, then don't want ice-tcp

Or, don't use FEC or RTP retransmission if ice-tcp

To Add, Replace or Delete Attributes

- Solution 1
  - Keep existing attributes and establish rules for how to deal with attributes added. For example:
    - Add all attributes at the beginning
    - For existing well-known attributes, specify rules that say to ignore subsequent attributes that redefine or contradict an existing value (e.g. rtpmap, fmtp, sendonly/recvonly)
    - Keep your fingers crossed for the rest
  - Pros:
    - Little text in spec (i.e. it looks simple)
    - Appealing if all you care about is a limited well-known problem
  - Cons:
    - Fuzzy semantics once we get into extensions (interoperability problem)
    - Does not address all well-known problems (e.g. key-mgmt versus crypto)

To Add, Replace or Delete Attributes

- Solution 2
  - Delete all attributes from the actual configuration SDP
  - Require the potential configuration SDP to add all necessary attributes
  - Pros
    - Clean semantics
    - Easy to understand and implement
  - Cons
    - All actual configuration attributes of relevance to a potential configuration will have to be duplicated as attribute capabilities
      - Message size concern, e.g. consider keying material, ICE candidates, etc.
      - Not clear we can always reconstruct the SDP we actually want
        - » Consider grouping framework which spans both the session and media-level
      - Magnus' 10k SDP ("How big can SDP get", MMUSIC, 5/17/06)

To Add, Replace or Delete Attributes

- Solution 3
  - Keep existing attributes
  - Provide mechanisms to remove one or more of the existing attributes
    - This is what is specified in the current solution
      - Allows all attributes at session and/or media level to be removed
      - Allows attributes with a particular name to be removed
  - Pros
    - Clean semantics
    - Provides decent message size efficiency
    - Addresses well-known problems and provides for extensibility
  - Cons
    - Probably not that simple to understand and use
      - Implementer has to know what he is doing
      - Currently specified mechanism effectively allows fallback to solution 2 though

To Add, Replace or Delete Attributes

- Recommendation
  - Solution 3
    - Is current specification for this adequate ?
    - Should we extend DELETE/REPLACE to
      - Have specific rules for rtpmap and fmtp parameters (include payload type or media format ?)
        - » Would allow for more efficient replacement of those values
      - Allow for string-based matching of attribute name and value as opposed to just attribute name
        - » Would allow for more targeted deletion and/or replacement of attributes

Inconsistent media-level selection of session-level capabilities

- Attribute capabilities can be present for the session-level and media-level
- Potential configurations are present at the media-level
  - The potential configuration applies to that media description
  - Can select media-level attribute capabilities
  - Can select session-level attribute capabilities
    - In case of two different media descriptions, each media description may select different session-level attributes.
    - It is possible, that inconsistent (or conflicting) session-level attributes could be selected as a result of that

Inconsistent media-level selection of session-level capabilities

- Possible solutions
  1. Disallow use of session-level attribute capabilities
     - Significant restriction
     - Would not enable session-level key-mgmt, diffie-hellman, etc.
  2. Add warning text in draft and encourage use of media-level attribute capabilities only, whenever possible
     - Clearly, there will still be room for errors here
  3. Add a mechanism to be able to specify attribute capabilities that are mutually exclusive (constraints)
     - More complexity
     - Still seems to address only a subset of the general problem
       - Constraints on combinations of multiple attributes and/or possibly related to other capabilities (e.g. transport)
       - Constraints between media descriptions

SDP Capability Negotiation

- Next Steps
  - We have a solution that satisfies the requirements at this point
  - Issue update addressing the comments raised on the list
    - Are we then ready for WGLC ?

SDP Media Capability Negotiation

Even

draft-ietf-mmusic-sdp-media-capabilities-01.txt

Media Capabilities

- Extends the base capabilities negotiation for RTP based media
- Codec definition can be very complicated
  - Simple media codec
  - Redundancy codecs (like FEC)
  - Composite like layered codecs
- Work is trying not to cover every combination (not H.245)

Solution model

- Unique codec numbers at session level
- Configuration parameters for rtpmap and fmtp associated with one or more codec numbers
- Map on the media level the codec numbers to RTP payload types assigning parameters from the configuration parameters.

Extends capability negotiation - example

- Adds media negotiation attributes

```
a=creq:v1
a=cmed:1 audio AMR AMR PCMU AMR
a=cmed:5 video H263
a=cenc:1,2 8000/1
a=cenc:4 16000/1
a=cfmt:1,4 max-red=220
a=cfmt:2,4 octet-align=1
a=lcfg:1 m=5
m=audio 3456 RTP/AVP 0 97
a=rtpmap:97 AMR/16000/1
a=fmtp:97 mode-change-capability=2; max-red=220
a=pcfg:1 m=1|2 pt=1:100,2:101
a=pcfg:2 m=3|4 pt=3:102,4:103
```

Open issues

- The work extends the basic capability negotiation whose full scope is not finalized.
- Address RTP based media – is this enough
- Final format is still open to suggestion – please review

Media Decoding Dependency

Schierl

draft-schierl-mmusic-layered-codec-03.txt

draft version 01, 02: Changes

- Extended and more precise definition and use-case sect.
- Extended section on Offer/Answer, declarative usage
- Removed SSRC mux in -02, also removed from (related) draft on RTP Payload for SVC – draft-ietf-avt-rtp-svc-01
- Original SSRC mux use-case for SVC:
  Adaptation of encrypted and authenticated content without being in the security context.
- Found out: Is not possible, since RTCP feedback is also authenticated within SRTP
- That means: Not possible to cover full SRTP functionality
- Remaining use-case:
  Adaptation of media stream without parsing Payload header in strongly restricted use-cases

Open issues/TBD:

- Should we re-think the SSRC discussion? See next presentation: draft-lennox-mmusic-sdp-source-attributes-00
- TBD: Capability Negotiation interaction/issues
- Planned: Integration of mechanism proposed in draft-schmidt-mmusic-media-dependency-00 indicating relations between medias of different types, e.g. subtitling text stream for a video stream

- I like to ask for working group status, since basic mechanism (SDP dependency grouping + dependency attrib.) seems to be stable

Source-Specific Media Attributes

Lennox

draft-lennox-mmusic-sdp-source-attributes-00.txt

Source-Specific Attributes: Review

- RTP allows multiple sources in an RTP session, but SDP has no way to signal this.
- Solution: define an SDP attribute for characteristics of a source.

```
m=video 49170 RTP/AVP 96
a=rtpmap:96 H264/90000
a=ssrc:12345 cname:user@example.com
a=ssrc:12345 information:Main camera
a=ssrc:67890 cname:user@example.com
a=ssrc:67890 information:Alternate camera
```

- Map SDP “source-specific” attributes into the `ssrc` attribute.
- This generalizes material that was previously in the RTP Single-Source Multicast draft.

Motivation

- Avoid clashes with the SSRC id of a single media sender.
  - This is needed for Single-Source Multicast.
- Make SSRC multiplexing explicit.
  - Describe, and differentiate between, multiple SSRCs from the same participant in the same RTP session.
  - Examples:
    - Multiple cameras
    - FEC
    - Retransmission
    - Layered codecs

Open Issues

- What does this mean for RTP collision detection?
  - Discussed in AVT.
- What source-specific attributes should be defined?
  - Define as needed: don’t inherit blindly from media level
- Should it be possible to select individual sources with offer/answer?
- IANA registration issues
  - Some media-level parameters (group semantics, bwtypes) are re-used at the source level.
  - Do they need a separate IANA registry?

Open Issues 2

- MIKEY (RFC 3830, signaled in SDP with RFC 4567) also specifies SSRCs in SDP.
  - Proposal: if you use both, they MUST be consistent.
  - Does more need to be said?
- Needs to work with capability negotiation.
  - Negotiate capabilities of a source, or negotiate sources?
- Backward compatibility.
- Need to clarify this isn’t for combining different media types.
  - Don’t multiplex audio and video.
  - One payload type number space.

Next Steps

- How much of this do we need?
  - declaration/identification of an SSRC vs. further qualification/negotiation.
- Need to move quickly on the base spec (RTCP-SSM dependency).
- Should this be a working group item of MMUSIC?

The MSR Bandwidth Modifier

Dondeti

draft-hdesinen-mmusic-oa-send-bw-attr-02.txt

Offer/answer for optimized asymmetric reservation

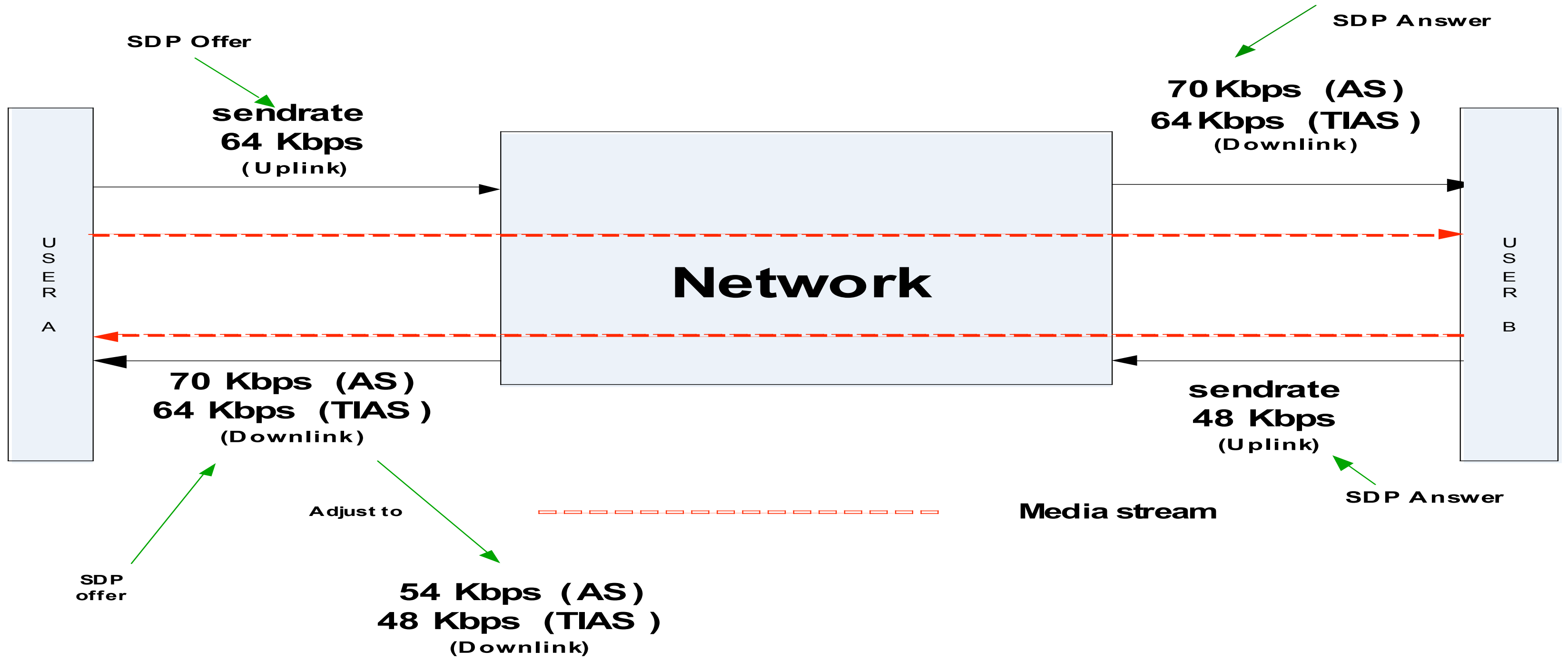
- Offer from A

m=video 34234 RTP/AVP 96	
a=AS:70 (Kbps)	(A ← B)
a=TIAS:64000 (bps)	(A←B)
a=maxprate:20	(A ←B)
b= MSR:64000	(A→B)

- Answer from B

m=video <>	
a=AS:70	(A → B)
a=TIAS:64000	(A → B)
a=maxprate:20	(A ←B)
b= MSR:48000	(B→A)

[Note: Text in the parenthesis is for information purpose only. It is not part of the actual signaling]



We need a bandwidth modifier in ‘send’ direction to signal the asymmetric bitrates of a bi-directional (‘sendrecv’) stream. This is also useful to signal the max bitrate of a ‘sendonly’ stream.

Is it sensible to use MSR bandwidth modifier for the above?

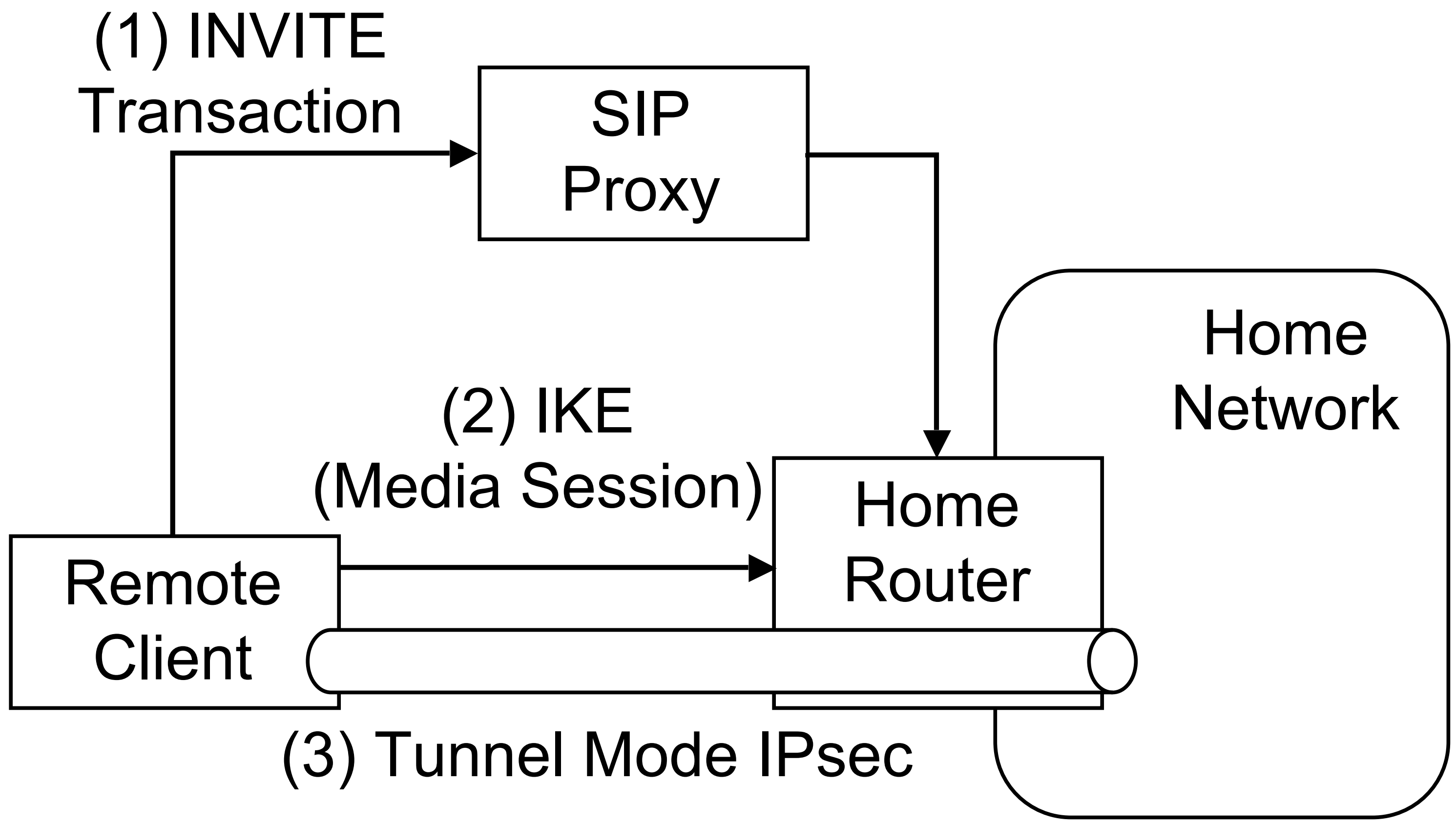
Media Description for IKE in SDP

Saito

draft-saito-mmusic-sdp-ike-00.txt

Purpose

- Setting up IPsec (IKE) Using SIP
  - VPN to a home router (or NAT device), etc.



Proposals

- Exchanging Fingerprint of Self-Signed Certificate for IKE Authentication
  - Same Concept as Comedia-tls (RFC4572)
- New Media Format Description “IKE”
  - m=application 500 UDP IKE
- New Attribute “udp-setup”
  - Similar to Comedia (RFC4145)

a=udp-setup:active	(IKE Initiator)
a=udp-setup:passive	(IKE Responder)

Configuring DiffServ using SDP

Polk

draft-polk-mmusic-dscp-attribute-01.txt

Summary

- Media Level Attribute
- For granularity of each stream within an offer/answer
- To be set by offerer or server in signaling path (in offer and in answer)
- Example:

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.com
c=IN IP4 10.1.3.33
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=dscp 46
m=video 51172 RTP/AVP 31 34
a=rtpmap:31 H.261/90000
a=rtpmap:34 H.263/90000
a=dscp 41
```

Changes in -01

1. Allow offerer to include “a=“ line per stream without a value to indicate support for extension
2. Allow offerer to include dscp value to indicate which dscp of a multi-dscp plan a user happens to want for each stream (a gold/silver/bronze service)
3. Added a “/dscp-value” to the attribute per stream to be able to set RTCP dscp value too

    ‘a=dscp 46/11’ or ‘a=dscp 46 11’

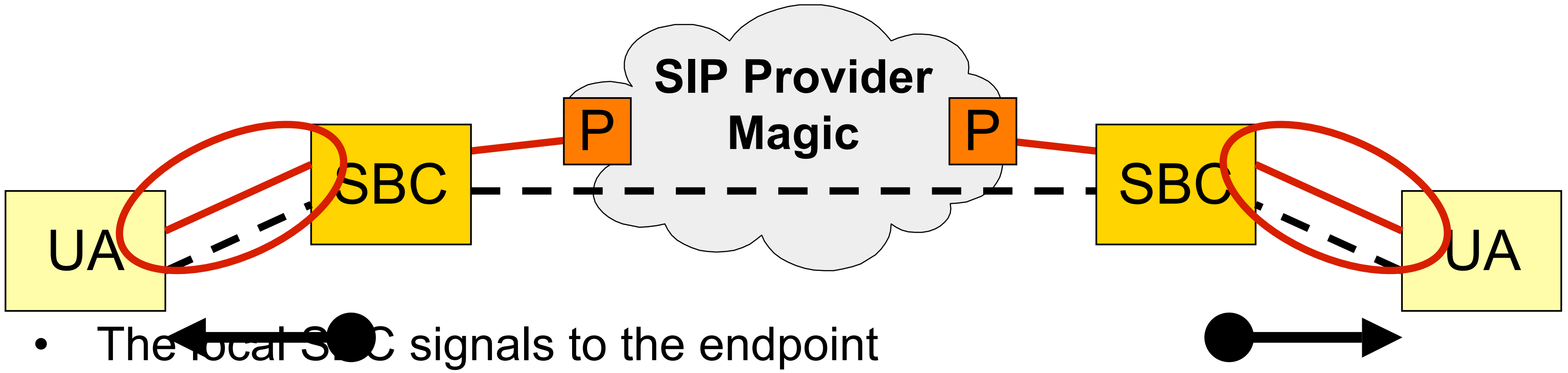
4. Added TSVWG item ‘EF DSCP for Capacity Admitted Traffic’ ID as requirement here
5. Cleaned up some text

Remaining Open Issue

- Should there be a direction indication (send, recv, sendrecv)?
  - Will need use cases if wanted
- Is there sufficient interest in this to make it a WG item?

General observations…

- This is some kind of middle-to-end signaling



- The local SBC signals to the endpoint
  - The rest of the SDP is more of less e2e though
  - Breaks end-to-end security
  - If we want to get something e2e at some point, will every have to implement both?
  - Feature interaction between multiple SBCs (who is the last hop)?
- Is this the first out of many things to come?
- What is the appropriate mechanism?
  - SDP signaling path? Media path? Some other signaling?