

JJ-22.14

企業 SIP 網に接続する SIP 端末⇔サーバ間 SDP とオファーアンサーモデル 技術仕様

SDP(session description protocol) and offer-answer model
technical specifications between SIP terminal <=> Servers
linked to a Private SIP network

第 1.0 版

2017 年 5 月 25 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、一般社団法人情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を一般社団法人情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

目次

<参考>	5
1. 概説	7
1.1. 本標準の適用範囲	7
1.1.1. 基本接続形態	7
1.2. 本標準の目的と規定	7
1.3. 本標準の内容	8
1.3.1. 本標準の策定の背景と位置づけについて	8
1.3.2. 本標準の位置づけ	8
2. 各種定義	9
2.1. 表における定義	9
3. オファーアンサーモデル	10
3.1. プロトコル制御	10
3.2. イニシャルオファーの生成	10
3.3. ユニキャストストリーム	11
3.4. マルチキャストストリーム	14
3.5. アンサーの生成	15
3.6. ユニキャストストリーム	15
3.7. マルチキャストストリーム	18
3.8. オファー送信側のアンサーの処理	18
3.9. セッション変更	18
3.10. メディアストリーム追加	19
3.11. メディアストリームの無効化	20
3.12. メディアストリームの変更	20
3.12.1. アドレス、ポート、または送信ポートの変更	20
3.12.2. メディアフォーマットのセットの変更	21
3.12.3. メディアタイプの変更	22
3.12.4. 属性変更	23
3.13. ユニキャストストリームの保留	23
3.14. 能力表示	24
3.15. オファーアンサーの交換の例	25
3.15.1. 基本的な交換	25
3.15.2. N個のコーデックから1つを選択	26
4. SDP仕様	29
4.1. プロトコルバージョン	31
4.2. オリジン	31
4.3. セッションネーム	32
4.4. セッションインフォメーション	32
4.5. URI	33
4.6. Eメールアドレスと電話番号	33
4.7. コネクションデータ	34

4.8.	帯域幅	35
4.9.	タイミング	36
4.10.	繰り返し回数	37
4.11.	タイムゾーン	37
4.12.	暗号化キー	38
4.13.	属性	39
4.14.	メディア記述(「m=」)	40
5.	SDP 属性	43
5.1.	a=cat:<category>	43
5.2.	a=keywds:<keywords>	43
5.3.	a=tool:<name and version of tool>	43
5.4.	a=ptime:<packet time>	43
5.5.	a=maxptime:<maximum packet time>	43
5.6.	a=rtptime:<payload type> <encoding name>/<clock rate> [/<encoding parameters>]	43
5.7.	a=recvonly	45
5.8.	a=sendrecv	45
5.9.	a=sendonly	45
5.10.	a=inactive	45
5.11.	a=orient:<orientation>	46
5.12.	a=type:<conference type>	46
5.13.	a=charset:<character set>	46
5.14.	a=sdplang:<language tag>	46
5.15.	a=lang:<language tag>	47
5.16.	a=framerate:<frame rate>	47
5.17.	a=quality:<quality>	47
5.18.	a=fmtp:<format> <format specific parameters>	48

<参考>

1. 国際勧告等の関連

本標準に関する国際勧告はない。

2. 改版の履歴

版数	制定日	改版内容
第 1.0 版	2017 年 5 月 25 日	初版制定

3. 参照文書

3.1 規準参照文書

- [1] RFC3264 An Offer/Answer Model with the Session Description Protocol (SDP)
- [2] RFC2327 SDP: Session Description Protocol
- [3] RFC3266 Support for IPv6 in Session Description Protocol (SDP)
- [4] "SIP:セッション開始プロトコル (SIP: Session Initiation Protocol)", TTC 標準 JF-IETF-RFC3261 第 1 版, 情報通信技術委員会 (The Telecommunication Technology Committee), 2005 年 6 月.
- [5] "SDP:セッション記述プロトコル (SDP: Session Description Protocol)", TTC 標準 JF-IETF-RFC4566 第 1 版, 情報通信技術委員会 (The Telecommunication Technology Committee), 2007 年 3 月.
- [6] RFC 3108 "Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections"
- [7] RFC 1889 "RTP:A Transport Protocol for Real-Time Applications"
- [8] RFC 1890 "RTP Profile for Audio and Video Conferences with Minimal control"
- [9] RFC1034 DOMAIN NAMES - CONCEPTS AND FACILITIE
- [10] RFC1035 DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION
- [11] RFC3490 Internationalizing Domain Names in Applications (IDNA)
- [12] RFC3550 RTP: A Transport Protocol for Real-Time Applications
- [13] RFC1305 Network Time Protocol (Version 3) Specification, Implementation and Analysis
- [14] RFC 4567 Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)
- [15] RFC4568 Session Description Protocol (SDP)Security Descriptions for Media Streams
- [16] RFC3548 The Base16, Base32, and Base64 Data Encodings
- [17] RFC2543 SIP: Session Initiation Protocol
- [18] “企業 SIP 網間における相互接続インタフェース” (Technical Specifications on Inter-connection Interface between Private SIP Networks)、JJ-22.01 情報通信技術委員会 (The Telecommunication Technology Committee), 2007 年 12 月.

4. 工業所有権

TTC の「工業所有権等の実施の権利に係る確認書」の提出状況は、TTC ホームページで公開されている。

5. 標準策定部門

企業ネットワーク専門委員会

1. 概説

1.1. 本標準の適用範囲

本標準は、JJ-22.01 [18] に規定されるフレームワーク標準の網接続アーキテクチャにおいて、私設総合サービス網交換機 (PINX) と SIP 端末間 (インターフェース E) の SDP を用いたオファアanswerモデルによるセッション確立のための推奨仕様を規定するものである。

また、インタフェース B を経由して接続される端末が、本標準の範囲を超えた能力を保持することを妨げるものではない。但し、その場合においても本標準に準拠する端末との接続性について考慮することが望ましい。

1.1.1. 基本接続形態

本標準は、図 1.1 で示す企業 SIP 網相互接続モデルに規定されるインタフェース E に適用可能な管理された企業 SIP 網との接続インタフェースの条件を示す。本インタフェースの規定を遵守できるインタフェースを有する企業 SIP 網に関して、本標準では“管理された企業 SIP 網”と呼ぶ。以下企業 SIP 網と表記する場合は、“管理された企業 SIP 網”であることを前提とする。

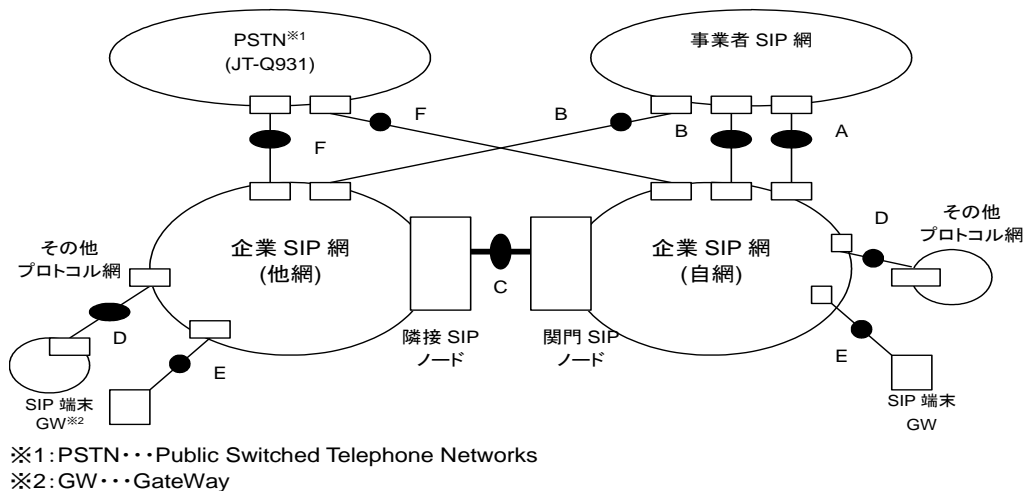


図 1.1 企業 SIP 網相互接続モデル

1.2. 本標準の目的と規定

本標準では、私設総合サービス網交換機 (PINX) 及び SIP 端末の実装に際して、

- 接続条件に関わる規定の解釈を一意とすることで、実装可能な標準とする。
- SIP 網を介した私設総合サービス網交換機 (PINX) と SIP 端末との接続において、共通的に適用することが可能な標準とする。
- 本規定の範囲を超えるまたは、厳密に本規定を遵守していない SIP UA との接続性にも最大限配慮した標準とする。

ことを目的に以下の規定を行う。

- オファアanswerモデルが定義されている RFC3264[1]に関する事項

- 古い端末がサポートする SDP 記法である RFC2327[2]と RFC4566 で規定される SDP 記法及びその拡張規定の用法に関する事項
- RFC3266[3]から取り込まれている IPv6 網で使う SDP 記法の用法に関する事項

1.3. 本標準の内容

本仕様の本文では、主として以下の事項について規定を行う。

- 私設総合サービス網交換機 (PINX) と SIP 端末は、B2BUA で接続されることを前提とし、SIP 網を介した私設総合サービス網交換機 (PINX) に接続するための動作規定として、オファーアンサーモデルの動作 (3 章) について規定する。
- オファーアンサーモデルで用いられる SDP メッセージのフォーマットとその属性に関する規定 (4 章、5 章) を行う。

1.3.1. 本標準の策定の背景と位置づけについて

本標準策定時点において、日本国内では私設総合サービス網交換機 (PINX) と SIP 端末間の SDP 伝送に関する規定が無かった。このため、本標準策定においては、本標準に準拠する端末同士が、それらが SDP による能力交換を行う際、私設総合サービス網交換機 (PINX) を介して SDP による能力交換を行うことだけを想定した規定ではなく、本標準に準拠する端末同士が、策定時点で既にサービス提供されている私設総合サービス網交換機 (PINX) を介して SDP による能力交換を行う場合や、さらには SDP による能力交換を行う私設総合サービス網交換機 (PINX) だけでなく相手端末が策定時点で既にサービス提供されている場合においても、それら私設総合サービス網交換機 (PINX) 及び端末との間で、基本的な SDP による能力交換が可能となるような共通の技術標準を策定することを目的として策定を行った。

1.3.2. 本標準の位置づけ

本標準は、既に存在する私設総合サービス網交換機 (PINX) や多様な音声端末との SDP による能力交換が可能となるような技術規準として定められる。なお、本標準の規定については、国内における SIP を利用し音声接続を行うことを目的とした端末における最低限の条件を満たす基準仕様として参照されることが期待される。

2. 各種定義

2.1. 表における定義

本規定に記載される表中において共通で使用する規定タイプの定義をエラー! 参照元が見つかりません。にて規定する。

表 2-1/JJ-22.14 表における規定タイプの定義

コード	コード名	意味
m	mandatory	この機能がサポートされている必要がある。 ここで“必須 (mandatory)”という言葉の意味することは、参照される規定を機能として有しているということである。必須とされているからといって必ず常に発現するというのではなく、参照する文書で規定される動作に従い、必要とされる場合に発現される必要があることを示している。 例えば、送信パラメータとして、このコードが指定されていた場合においては、常にそのパラメータが含まれていることを意味せず、参照文書の規定に従って必要な場合に含められることを意味している(動的な適用要求条件)。
o	optional	能力のサポートは、実装に依存しており、サポートされるかもしれないし、サポートされないかもしれない。
n/a	not applicable	この機能のサポート条件は、意味を持たない。(対応する機能を規定する標準を示す) サポート列は省略してよい。
x	prohibited (excluded)	この機能は、使用が許されない。
c <integer>	conditional	機能のサポート条件 (m,o,n/a,x) は、他の選択可能な条件 (c<数字>で示される) に依存している。
o.<integer>	qualified optional	同じ <数字>の中から排他的に選択可能なオプションであることを示す。
i	irrelevant	本標準の範囲外であることを示す。ベースとなる仕様が別の標準に立脚している場合などでは別の場合でも有用なケースもある。

3. オファーアンサーモデル

3.1. プロトコル制御

オファーアンサーの交換は、エージェント間のセッション確立を目的とする SDP 交換が可能な (SIP のような) 上位層プロトコルが存在することを前提とする。

プロトコル操作は、あるエージェントが他のエージェントにイニシャルオファーを送信する時に始まる。上位層プロトコルを通して、既に確立されたオファーが「どのコンテキストでもない場合」、そのオファーはイニシャルオファーである。

上位層プロトコルは、関連付けた様々な SDP 交換を許容した数種類のコンテキスト管理を提供することを前提とする。

オファーを受信するエージェントは、アンサーを生成してもよいし [MAY]、オファーを拒否してもよい [MAY]。

オファーを拒否する手段は、上位層プロトコルに依存する。

オファーアンサーの交換は、自動的に行われる。アンサーが拒否される場合、セッションはオファーの前の状態に戻る (セッションがなくなるかもしれない)。

双方のエージェントは、セッションを更新する新しいオファーをいつでも生成してもよい [MAY]。

しかし、アンサーか拒否をまだ行っていないオファーを受信した場合は、新しいオファーを生成してはいけない [MUST NOT]。

さらに、アンサーか拒否をまだ受信せずに前のオファーを生成する場合、新しいオファーは生成してはいけない [MUST NOT]。

エージェントがオファーを送信した後に、そのオファーに対するアンサーを受信する前に別のオファーを受信する場合、チャネル衝突な状態であると考えられる。

チャネル衝突という用語は、もともと回線交換ネットワークにおいて、2つの交換機が同じトランク上で有効な同じ回線を同時に使用しようとした状態を説明するために使用された。ここでは、両方のエージェントが更新オファーの送信を同時に試みたことを意味する。

上位層プロトコルは、そのような状態を解決する手段を提供する必要がある。

上位層プロトコルは、各方向でメッセージ順序の手段を提供する必要があるでしょう。

SIP は、これらの必要条件を満たす [4]。

3.2. イニシャルオファーの生成

オファー (アンサー) は、1つの例外を除いて、RFC4566 [5] で定義されるような有効な SDP メッセージでなければならない [MUST]。

RCC2327 は、SDP メッセージの中に e 行か p 行のどちらかが存在することを必須としている。

この仕様は制約を緩める。オファーアンサーアプリケーションの為に形成された SDP は、e 行と p 行の両方を省略しても良い [MAY]。

o 行のセッション ID とバージョンの値は、64 ビット符号付整数で表さなければならない [MUST]。バージョンの初期値は、循環を避ける為に $(2^{*62})-1$ 未満にしなければならない [MUST]。

SDP 仕様は、大きな SDP メッセージの中に連結する複数のセッション記述を許容するが、オファアアンサーモデルで使用される SDP メッセージは、1 つのセッション記述のみを含まなければならない [MUST]。

SDP の「s=」行は、セッションのサブジェクトを伝える。マルチキャストには有効な定義であるが、ユニキャストには不適切な定義である。

ユニキャストセッションに対しては、1 つのスペース文字 (0x20) やダッシュ文字 (-) で構成することが推奨される [RECOMMENDED]。

残念ながら、SDP では「s=」行を空にすることは許容しない。

SDP の「t=」行は、セッション時間を伝える。

一般的に、ユニキャストセッションストリームは、SIP のような外部シグナリング手段を通して生成・破棄される。その場合、「t=」行は「00」の値にすべきである [SHOULD]。

オファアは、0 以上のメディアストリームを含む (各メディアストリームは、「m=」行とその関連属性で記述される)。0 のメディアストリームは、オファア側が通信を望んでいる事を示すが、セッションストリームは、編集されたオファアによって後で追加される。

ストリームは、ユニキャストとマルチキャストを組み合わせても良い [MAY]。後者は該当する「c=」行で明確にマルチキャストアドレスを示す。

各々にオファアされたストリームの構築は、ストリームがマルチキャストまたはユニキャストであるかに依存する。

3.3. ユニキャストストリーム

オファア側が、ピアに向けてストリーム上のメディア送信のみを望む場合、「a=sendonly」属性を sendonly ストリームに付けなければならない [MUST]。

ディレクション属性が、メディアストリーム属性かセッション属性のどちらかに存在する場合、特定のディレクションを付けるストリームについて言及する。

オファア側がピアからのメディア受信のみを望む場合、recvonly ストリームに付けなければならない [MUST]。

オファア側が通信したいが、同時にメディア送受信を望まない場合、「a=inactive」属性をストリームに付けなければならない [MUST]。

inactive のディレクション属性は、RFC3108 [6] で規定される。

リアルタイムトランスポートプロトコル (RTP) [7] の場合、RTCP は sendonly と recvonly と inactive ストリームで送受信されることに注意して下さい。

メディアストリームの方向性は、RTCP 使用においてインパクトがない。

オファア側が、ピアとメディア送受信を望む場合、sendrecv はデフォルトなので、「a=sendrecv」属性を含んでよいし省略してもよい [MAY]。

recvonly と sendrecv ストリームに対して、オファア内のポート番号とアドレスは、オファア側がどこでメディアストリームを受信したいかを示す。

sendonly RTP ストリームに対して、ポート番号とアドレスは、オファー側がどこで RTCP レポートを受信したいかを間接的に示す。

明確な指示がない場合は、レポートは指定されたポート番号よりも 1 つ大きなポート番号に送信される。オファー側に存在する IP アドレスとポート番号は、オファー側によって送信される RTP と RTCP パケットのソース IP アドレスとソースポートについて何も示さない。

オファーのポート番号 0 は、ストリームがオファーされたということを示すが使用してはいけない [MUST NOT]。

イニシャルオファーには役に立つセマンティクスは無いが、アンサーが拒否ストリーム (セクション 3.5) を示しながら 0 ポートを含むことができるので、完全性の理由により許容されている。

その上、存在するストリームは、ポートを 0 に設定することで停止することができる (セクション 3.9)。一般的にポート番号 0 は、メディアストリームを希望しないことを示す。

各メディアストリームに対するメディアフォーマットのリストは、2 つの情報を伝える。

具体的には、オファー側が送信およびまたは受信する能力 (方向属性に依存) があるフォーマット (RTP の場合におけるコーデックとコーデックに関するパラメータ) の設定と、RTP の場合にそれらのフォーマットを識別する為に使用する RTP ペイロードタイプ番号である。

複数のフォーマットが列挙されている場合、オファー側はセッション中にそれらのフォーマットのいずれも利用できることを意味する。

言い換えると、アンサー側は、新しいオファー送信をせずに列挙されているフォーマットのいずれかを使用してセッション中にフォーマットを変更してもよい [MAY]。

sendonly ストリームについては、オファーは、オファー側がストリーム送信を望んでいるフォーマットであることを示すべきである [SHOULD]。

recvonly ストリームについては、オファーは、オファー側がストリーム受信を望んでいるフォーマットであることを示すべきである [SHOULD]。

sendrecv ストリームについては、オファーは、オファー側が送受信を望んでいるコーデックであることを示すべきである [SHOULD]。

recvonly ストリームについては、ペイロードタイプ番号は、オファー側がコーデック受信を予測している RTP パケット内のペイロードフィールドの値を示す。

sendonly ストリームについては、ペイロードタイプ番号は、オファー側がコーデック送信を予測している RTP パケット内のペイロードフィールドの値を示す。

sendrecv ストリームについては、ペイロードタイプ番号は、オファー側が受信を予測して、送信しようとするペイロードフィールドの値を示す。

しかしながら、sendonly と sendrecv については、アンサーは、同じコーデックで異なるペイロードタイプ番号を示してもよいし、その場合は、オファー側はアンサーから示されたペイロードタイプ番号を付けて送信しなければならない [MUST]。

異なるペイロードタイプ番号は、H.323 に関わる相互互換の為に各方向で必要かもしれない。

RFC4566 に従って、fmp パラメータは、メディア形式の追加パラメータを提供するために存在してもよい [MAY]。

RTP ストリームの場合、全てのメディア記述は、RTP ペイロードからエンコードへのマッピングを行う「a=rtptime」を含むべきである [SHOULD]。

もし「a=rtptime」が無ければ、デフォルトペイロードタイプのマッピングにおいて、現在使用中のプロファイル (例えば RFC1890 [8]) で定義されるように使用される。

これは、静的ペイロードからの、より簡単な移行を許容する。

全ての場合において、「m=」行のフォーマットは、優先的に列挙された最初のフォーマットで、優先順に列挙されなければならない [MUST]。

この場合、オファーの受信者が許容する最高優先のフォーマットを使用すべきである [SHOULD] 事を意味する。

ptime 属性がストリームに存在する場合は、オファー側が受信したいパケット化間隔を示す。

ptime 属性は、ゼロよりも大きくしなければならない [MUST]。

帯域幅属性がストリームに存在する場合は、オファー側が受信したい希望の帯域幅を示す。

ゼロ値は許容されるが、推奨されない。メディアが全く送信されるべきでは無いことを示す。

RTP の場合は、全ての RTCP が無効になるでしょう。

異なるタイプのマルチメディアストリームが存在する場合は、オファー側が同時にストリームを使用したい事を表す。典型的な例は、ビデオ会議の一部としての音声とビデオのストリームです。

同じタイプの複数のメディアストリームがオファーに存在する場合は、オファー側が同時にそのタイプの複数のストリームを送信および/または受信したいことを表す。

同じタイプの複数のストリームを送信する時、そのタイプ (例えば、ビデオカメラ、ビデオの場合の VCR) の各メディアソースが、どのようにして各ストリームに配置されるかに関するローカルポリシーの問題です。

ユーザが特定のメディアタイプに対する単独のソースを持っている時、1つのポリシーだけが理解できる：そのソースは同じタイプの各ストリームに送信される。各ストリームは異なるエンコードを使用してもよい [MAY]。

同じタイプの複数のストリームを受信する時、各ストリームが様々な特定のタイプのメディアシンク (例えば、音声の場合のスピーカや録音デバイス) にどのように配置されるかは、ローカルポリシーの問題です。

しかしながら、ポリシーにはいくつかの規制がある。

初めに、同じタイプのマルチストリームを受信する時、ユーザへのプレゼンテーションの目的の為に、各ストリームは少なくとも1つのシンクに配置されなければならない [MUST]。

言い換えると、同じタイプのマルチストリームを受信する意図は、1つのストリームを選択するのではなく、全てが並列に存在すべきである。

もう一つの規制は、マルチストリームを受信して同じシンクに送信する時で、あるメディアの指定方法で結合しなければならない [MUST]。

例えば、2つの音声ストリームの場合は、お互いが受信したメディアはスピーカに配置されるかもしれない。

その場合は、結合操作でそれらは混合される。

シンクが画面となる複数のインスタントメッセージストリームの場合は、結合操作はユーザインターフェースにそれら全てを表示提示する。

3番目の規制は、複数ソースが同じストリームに配置されるなら、それらのソースはストリームを送信する前に、あるメディアの指定方法で結合されなければならない [MUST]。

これらの規制を超えたポリシーは可変だが、会議サーバでなければ、シンクからソースへメディアをコピーするポリシーをエージェントは一般的に要求したくない(すなわち、1つのストリーム上に受信するメディアを別のストリームにコピーしない)。

同じタイプのマルチメディアストリームに関する典型的な使用例は、プリペイド式テレホンカードアプリケーションです。ユーザは、同じカードで呼切断と発呼を行うために、~~開~~に通話中いつでも#キーを押して保留にすることができる。

ユーザから2つの宛先へのメディアが必要になる(リモートゲートウェイと#ボタン押下を検索するDTMF処理アプリケーションである)。

2つのメディアストリームにより達成が可能で、1つは、ゲートウェイへの `sendrecv` で、もう1つは、DTMFアプリケーションへの(ユーザから見た) `sendonly` である。

オファー側がオファーを送信すると、そのオファーで説明される全ての `recvonly` ストリームに対してメディアを受信する準備をしなければならない [MUST]。

オファーの全ての `sendrecv` ストリームに対するメディア送受信と、オファーの全ての `sendonly` ストリームに対するメディア送信を準備をしなければならない [MUST] (もちろん、必要なアドレスとポートを含めたアンサーを対向者が提供するまでは実際には送信できない)。

RTPの場合は、アンサーが届く前にメディアを受信するかもしれないが、アンサーが届くまではRTCP受信者レポートを送信できない。

3.4. マルチキャストストリーム

セッション記述が、受信(送信)専用リストのマルチキャストメディアストリームを含む場合は、オファー側とアンサー側を含む参加者が、そのストリームで受信(送信)のみが可能であることを意味する。

これは、方向性がオファー側とアンサー側の間のメディアフローについて示すユニキャスト視点とは異なっている。

その明確化を超えて、提供されたマルチキャストストリームのセマンティックは、RFC4566 [5] で説明される。

3.5. アンサーの生成

オファー側のセッション記述に対するアンサーは、オファー側のセッション記述に基づく。

アンサーが、何らかの方法でオファー側と異なる場合は (異なる IP アドレス、ポートなど)、アンサーは異なるエンティティによって生成されるので、基本の行はアンサーとは異ならなければならない [MUST]。

その場合は、アンサーの「o=」行のバージョンは、オファーの o 行のバージョンとは関係しない。

オファーの各「m=」行については、アンサーの「m=」行に対応しなければならない [MUST]。

アンサーは、オファーの「m=」行と同じ値を含まなければならない [MUST]。

これは、オーダーに基づいて照合されるストリームを許容する。

オファーが「m=」行を含まないなら、アンサーは「m=」行を含めてはならない [MUST]。

アンサーの「t=」行は、オファーと同じでなければならない [MUST]。

セッション時間は、ネゴシエートできない。

オファーされたストリームは、アンサーで拒否されるかもしれない [MAY]。

ストリームが拒否される場合、オファー側とアンサー側はストリームの為にメディア (または RTCP パケット) を生成してはいけない [MUST NOT]。

オファー側のストリームを拒否する為に、アンサーに対応するストリームのポート番号は、ゼロに設定しなければならない [MUST]。

どのようなメディアフォーマット形式も無視される。少なくとも、SDP で指定されるように存在しなければならない [MUST]。

各オファー側のストリームに対するアンサーの構成は、ユニキャストとマルチキャストで異なる。

3.6. ユニキャストストリーム

ストリームがユニキャストアドレスと共にオファーされる場合は、そのストリームに対するアンサーはユニキャストアドレスを含まなければならない [MUST]。

アンサーストリームのメディアタイプは、オファーのメディアタイプと一致しなければならない [MUST]。

ストリームが `sendonly` としてオファーされる場合は、対応するストリームはアンサー内に `recvonly` か `inactive` を付けなければならない [MUST]。

メディアストリームがオファー内に `recvonly` として列挙される場合は、アンサーはアンサー内に `sendonly` か `inactive` を付けなければならない [MUST]。

オファー側のメディアストリームが `sendrecv` として列挙される場合は (または、メディアかセッションレベルに方向属性が無い場合)、アンサーに対応するストリームは、`sendonly`, `recvonly`, `sendrecv`, `inactive` を付けてもよい [MAY]。

アンサーに `recvonly` を付けたストリームについては、「m=」行には、アンサー側がオファーで列挙されたものの中から受信する意思のあるメディアフォーマットを少なくとも 1 つ含まなければならない [MUST]。

ストリームは、オファーでは関連するストリームに列挙されていない、アンサー側が受信する意思のある追加メディアフォーマットを示してもよい [MAY]。

アンサーに `sendonly` を付けたストリームについては、「`m=`」行には、アンサー側がオファーで列挙されたものの中から送信する意思のあるメディアフォーマットを少なくとも1つ含まなければならない [MUST]。

アンサーに `sendrecv` を付けたストリームについては、「`m=`」行にはアンサー側がオファーで列挙されたものの中から送信および受信する意思のあるコーデックを少なくとも1つ含まなければならない [MUST]。

ストリームは、オファーでは関連するストリームに列挙されていない、アンサー側が送信または受信する意思のある追加メディアフォーマットを示してもよい [MAY] (もちろん、オファーで列挙されないの、この時にそれらを送信することはできない)。

アンサーに `inactive` を付けたストリームについては、メディアフォーマットのリストはオファーに基づいて構成される。

オファーが `sendonly` の場合は、アンサーがまるで `recvonly` であるかのようにリストが構成される。

同様に、オファーが `recvonly` の場合は、アンサーがまるで `sendonly` であるかのようにリストが構成される。そして、オファーが `sendrecv` の場合は、アンサーがまるで `sendrecv` であるかのようにリストが構成される。

オファーが `inactive` の場合は、オファーがまるで `sendrecv` であり、アンサーがまるで `sendrecv` であるかのようにリストが構成される。

アンサー内のコネクションアドレスとポートは、アンサー側がメディアを受信する意思があるアドレスを示す (RTP の場合に、明確な指示が他に無い場合、RTCP は1つ上のポート番号で受信される)。

このアドレスとポートは、`sendonly` ストリームの為に存在しなければならない [MUST]。

RTP の場合は、1つ上のポート番号は、RTCP を受信するために使用される。

RTP の場合に、特定のコーデックがオファー中の特定のペイロードタイプ番号と共に参照される場合は、同じペイロードタイプ番号がアンサーのコーデックに使用されるべきである [SHOULD]。

同じペイロードタイプ番号が使用されていても、アンサーは、動的なペイロードタイプの為にペイロードタイプのマッピングを定義するのに `rtpmap` 属性を含まなければならない [MUST]、静的ペイロードタイプの為にマッピングを含むべきである [SHOULD]。

「`m=`」行のメディアフォーマットは、優先して列挙された最初のフォーマットで優先的に列挙されなければならない [MUST]。

この場合の優先とは、オファー側がアンサーから最優先でフォーマットを使用すべき [SHOULD] であることを意味する。

アンサーは優先的に決定されたフォーマットを列挙してもよい [MAY] が、特定の理由が無ければ、アンサー側はオファーで提示された同じ順序でフォーマットを列挙することが推奨される [RECOMMENDED]。

言い換えれば、オファーのストリームが音声コーデックを 8,22,48 の順序で列挙する場合は、アンサー側は、コーデック 8,48 だけをサポートされる場合、アンサー側にそれを変更する理由が無い場合は、アンサーのコーデック順序は 8,48 であり、48,8 ではないことが推奨される。

これは、同じコーデックが両方向で使用されることの保証を助ける。

オファ어의 `fmtp` パラメータの解釈は、パラメータに依存する。多くの場合、それらのパラメータはメディアフォーマットの独自設定を説明していて、メディアフォーマット値のように処理されるべきである。

それらが説明するメディアフォーマットがアンサーに存在するなら、同じ値の同じ `fmtp` パラメータがアンサーに存在しなければならない [MUST] ということの意味する。

この場合は、アンサーは `fmtp` パラメータを含んでも良いし [MAY]、オファーの中に同じ値を持っても良いし [MAY]、異なっても良い [MAY]。

新しいパラメータを定義する SDP 拡張子は、オファーアンサーに適切な解釈を指定すべきである [SHOULD]。

アンサー側は、どのメディアストリームに対しても 0 でない `ptime` 属性を含んでもよい [MAY]。これはアンサー側が受信したいパケット化の間隔を示す。

パケット化の間隔は、特定のストリームに対して各方向で同じである必要は無い。

アンサー側は、どのメディアストリームに対しても帯域幅属性を含んでもよい [MAY]。これは、アンサー側がオファー側にメディア送信時に使用したい帯域幅を示す。0 の値は許容され、セクション 3.2 で説明されるように解釈される。

アンサー側が、特定のオファーのストリームに対して、共通のメディアフォーマットを持たない場合は、アンサー側はポートを 0 に設定することでメディアストリームを拒否しなければならない [MUST]。

全てのストリームに対して共通のメディアフォーマットが無い場合は、全体のオファーのセッションは拒否される。

アンサー側がアンサーを送信すると、アンサーで説明されるどの `recvonly` ストリームに対してもメディアを受信する準備をしなければならない [MUST]。

アンサーの中の全ての `sendrecv` ストリームに対して送受信する準備をしなければならない [MUST]、直ぐにメディアを送信してもよい [MAY]。

アンサー側は、`recvonly` や `sendrecv` に対して、アンサーの中のストリームで列挙されている全てのメディアフォーマットを使用して、メディアを受信する準備をしなければならない [MUST]、直ぐにメディアを送信してもよい [MAY]。

メディアを送信する時は、オファーに `ptime` 属性が存在する場合は、オファーの中の `ptime` 属性値と同じパケット化の間隔を使用すべきである [SHOULD] 。

オファーに帯域幅属性が存在する場合は、オファーの中の帯域幅属性値よりも高くない帯域幅を使用してメディアを送信すべきである [SHOULD] 。

アンサー側は、アンサーで列挙されたオファー内のメディアフォーマットを使用して送信しなければならない [MUST]、アンサーで列挙されたオファー内の最優先のメディアフォーマットを使用して送信すべきである [SHOULD] 。

RTP の場合は、アンサーのペイロードタイプ番号と異なっても、オファーからペイロードタイプ番号を使用しなければならない [MUST] 。

3.7. マルチキャストストリーム

ストリームの観点が2つあるユニキャストとは異なり、マルチキャストにはストリームの観点は1つしかない。

そのように、マルチキャストオファーへのアンサー生成は、一般的に、ストリーム側面の限定的な設定変更を伴う。

マルチキャストストリームが許容される場合は、アンサー内のアドレスとポート情報は、オファー内の情報と一致しなければならない [MUST]。

同様に、アンサー内の方向情報 (`sendonly`, `recvonly`, `sendrecv`) はオファー内の情報と一致しなければならない [MUST]。

これは、RFC4566 のマルチキャストに関して暗黙に了解されているマルチキャストセッション内の全ての参加者がセッションパラメータの同一観点を必要とする為である。

アンサー内のメディアフォーマットのセットは、オファー内のそれらと同等か、サブセットでなければならない [MUST]。

フォーマットの削除は、フォーマットをサポートしていない事を示すアンサー側の手段である。

アンサー内の `ptime` と帯域幅属性は、オファーが存在する場合は、オファー内のそれと同等でなければならない [MUST]。

オファーが存在しない場合は、0でない `ptime` はアンサーに追加してもよい [MAY]。

3.8. オファー送信側のアンサーの処理

オファー側がアンサーを受信したとき、受け入れられたストリームでメディアを送信してもよい [MAY] (アンサーで `sendrecv` または `recvonly` として記載されているならば)。

アンサーで記載されたメディアフォーマットを使用して送信しなければならず [MUST]、送信時にはアンサーに記載された最初のメディアフォーマットを使用すべきである [SHOULD]。

これが MUST ではなく、SHOULD である理由 (アンサー側もまた MUST では無く SHOULD である) は、多くの場合、臨機応変にコーデックを変える必要があるためである。

例えば、無音期間の間にエージェントはコンフォートノイズコーデックに切り替えたいかもしれない。またはユーザがキーパッドの数字を押した場合、エージェントは RFC2833 を使用して送信したいかもしれない。輻輳制御はフィードバックに基づいて、より低レートのコーデックに変更する必要があるかもしれない。

オファー側はアンサー側の `ptime` の値と `bandwidth` 属性に従ってメディアを送信すべきである [SHOULD]。

オファー側は最初のオファーで記載したが、アンサーには存在していないメディアフォーマットについては聴取するのをすぐに停止してもよい [MAY]。

3.9. セッション変更

セッションの間のいつでも、どちらの参加者でも、セッションの特性を変更するための新規のオファーを出してもよい [MAY]。

これは、上記で定義された正確な同じオファーアンサー手順が、既存のセッションのパラメータを変更するために使用されるオファーアンサーモデルの動作の基本である。

オファーは (オファー、またはアンサーで提示されたかもしれない) 相手方に提示された最後の SDP と同一であってもよい [MAY]、または異なってもよい [MAY]。

本章では“前の SDP”という条件で最後の SDP を参照する。

オファーが同じであれば、アンサー側から前の SDP と同じアンサーであってもよい [MAY] し、または異なってもよい [MAY]。

もし、オファーした SDP が前の SDP と異なっていた場合、後述されるように、構築に若干の制約がある。

セッションのほとんど全ての場面で変更できる。

新規ストリームを追加することができ、既存のストリームを削除することができ、既存のストリームのパラメータは変更することができる。

セッションを変更するオファーを発行する場合、新規 SDP の「o=」行は、前の SDP と同一でなければならない [MUST]。

これには origin フィールドのバージョンは、前の SDP から 1 つインクリメントしなければならない [MUST] という例外がある。

origin 行のバージョンがインクリメントしていないならば、SDP はそのバージョン番号での SDP と同一でなければならない [MUST]。

アンサーは、バージョンが変更されていない SDP を含むオファーを受信する準備ができていなくてはならない [MUST]。

これは事実上ノーオペレーションである。

しかし、第 3.5 章で定められた手順に従い、アンサー側は正しいアンサーを生成しなければならない [MUST]。(アンサーから前の SDP と同じである場合もあり [MAY]、異なる場合もある [MAY]。)

前の SDP と異なった SDP をオファーする場合、新規の SDP は前の SDP での各メディアストリームに一致させたメディアストリームを持たなければならない [MUST]。

言い換えると、前の SDP が N 個の“m=”行を持っているならば、新しい SDP は少なくとも N 個の“m=”行を持たなければならない [MUST]。

前の SDP の先頭から数えて i 番目のメディアストリームは、新規の SDP の先頭から数えて i 番目のメディアストリームに一致させる。

この一致させることは、アンサーが新規の SDP のどのストリームが前の SDP のストリームと対応するかを決定するために必要である。

これらの必要条件のため、ストリームの“m=”行の数は決して減らしてはならず、同数のままか、増加する。

前の SDP から削除したメディアストリームは新規の SDP で無効にしてはならない [MUST NOT]、しなしながらこれらのストリームの属性 (attribute) は提示される必要が無い。

3.10. メディアストリーム追加

新規のメディアストリームは、既存の下に新規に追加するメディアを記述、またはポートを 0 に設定することで無効にされた古いメディアストリームで使用していた“スロット”を再利用することにより作成される。

スロットの再利用は、新規のメディア記述が古い方を置き換えるが、SDP 内の他のメディア記述との相対位置は保持することを意味する。

新規のメディア記述は既存のメディアセクションの下に表示しなければならない [MUST]。

これらのメディア記述のフォーマットのルールはセクション 3.5 で説明されたものと同じである。

アンサー側が前の SDP より多くのメディア記述を持つ SDP をオファー側から受信した時、またはポートが以前は 0 であったスロットにメディアストリームを持つ SDP を受信した時、アンサー側は新規のメディアストリームの追加があったことを知る。

アンサーに適切に構成されたメディア記述を配置することで、拒否したり、受け入れたりすることができる。アンサーでの新規メディア記述を構成するための手順はセクション 3.5 で説明されている。

3.11. メディアストリームの無効化

既存のメディアストリームは、ストリームのポート番号を 0 にセットした新規の SDP を作成することで無効化される。

ストリーム記述は以前に存在した全ての属性 (attribute) を省略してもよい [MAY]、ただ一つのメディアフォーマットのみを記載してもよい [MAY]。

ポートを 0 としてオファーされたストリームはアンサーでもポート 0 を指定しなくてはならない [MUST]。オファーのようにアンサーも以前に存在した全ての属性 (attribute) を省略してもよい [MAY]、オファーの中からただ一つのメディアフォーマットを記載してもよい [MAY]。

メディアストリームの無効化は、メディアがそのストリーム上で送信することが無く、受信された全てのメディアが廃棄されることを意味する。

RTP の場合、RTCP パケットの受信処理と同様に、RTCP 送信も停止する。

それに関連したリソースも解放される。

例えば、PC で関連するウィンドウを閉じることによって、ユーザインタフェースはストリームが終了したことを示すかもしれない。

3.12. メディアストリームの変更

メディアストリームのほとんどすべての特性を変更できる。

3.12.1. アドレス、ポート、または送信ポートの変更

ストリームのポート番号は変更してもよい [MAY]。

これを実施するため、オファー側は前の SDP で対応しているストリームとは異なった m 行のポート番号を持つ新規のメディア記述を作成する。

もし、ポート番号のみを変更する場合、メディアストリーム記述の残りについては変更無しとすべきである [SHOULD]。

オファー側はオファーが送信されたらすぐに新旧のポート両方でメディアを受信する準備を整えなければならない [MUST]。

オファー側はアンサーが受信され、メディアが新ポートに到着するまで、旧ポートでのメディア聴取を停止すべきではない [SHOULD NOT]。

そう (停止) することは、(ポート番号の) 移行中にメディアをロスする結果となることがあり得る。

この場合の「受信した」とは、メディアがメディア受信装置を通ったことを意味する。

この意味は再生バッファがある場合、エージェントは新規ポートでのメディアが再生バッファの先頭に達するまで、旧ポートでの聴取を続けることである。

その時には旧ポートでのメディアの聴取を停止してもよい [MAY]。

アンサーの中的一致しているメディアストリームは、アンサー側からの前の SDP のストリームと同じであってもよい [MAY] し、異なってもよい [MAY]。

もし更新したストリームがアンサー側で受け入れられるなら、アンサー側はすぐに新規ポートへトラフィックを送信し始めるべきである [SHOULD]。

もしアンサー側が前の SDP からポートを変えるなら、アンサーを送信してすぐに旧ポートと新ポートの両方でメディアを受信する準備をすべきである [MUST]。

アンサー側が新ポートでメディアが到着するまで旧ポートでのメディア聴取を停止してはならない [MUST NOT]。

新ポートでメディア到着時、旧ポートでのメディア聴取を停止しても良い [MAY]。

同じことは新規ポートを持つ、更新オファーを送信するオファー側についても当てはまる。メディアが新規ポートに到着するまで、旧ポートでのメディア聴取を停止してはならない [MUST NOT]。

もちろん、オファーされたストリームが拒否された場合、オファー側は拒否を受信するとすぐに新ポートでの受信する準備を停止する。

メディアを送信される IP アドレスを変更するために、ポート番号を変更するのと同じ手順を取る。唯一の違いは、コネクション行 (注: "c="行) がアップデートされることである。ポート番号では更新されない。

ストリームについての送信ポートは変更されても良い [MAY]。

実行するための処理は送信ポートが更新される以外はポートを変更することと同一である。ポートでは更新されない。

3.12.2. メディアフォーマットのセットの変更

セッションの時に使用されるメディアフォーマットのリストは、変更してもよい [MAY]。

変更するために、オファー側は前の SDP に対応するメディアストリームとは異なった "m =" 行のメディアフォーマットのリストを持つ新規のメディア記述を作成する。

このリストは新規のフォーマットを含んでも良く [MAY]、前の SDP から存在するフォーマットを無効にしてもよい [MAY]。

しかしながら、RTP の場合、メディアストリーム内で特定のダイナミックペイロードタイプ番号から特定のコーデックにマッピングされたものは、セッションの間は変更してはならない [MUST NOT]。

例として、A が動的ペイロードタイプ番号 46 に G711 を割り当ててオファーを生成した場合、この時点から以降は、セッション中のメディアストリームについてのオファー、アンサーではペイロードタイプ番号 46 は G711 を参照しなければならない [MUST]。

しかしながら、同じコーデックに複数のペイロードタイプ番号を割り当てることは許容されているので、アップデートのオファーとして、G711 のためにペイロードタイプ番号 72 を使用することもできる。

SDP の信号交換とメディアストリームの間は同期していないため、セッションの間はマッピングが固定されたままの必要がある。

アンサーで一致したメディアストリームはセクション 6 で説明されるように定式化されており、同様にメディアフォーマットの変更という結果になる場合がある。

同様にセクション 6 で説明されているように、アンサー側はアンサーを送信するとすぐに、アンサーにも存

在し、オファーにあるフォーマットのどれかでメディアを送り始めなければならない [MUST]、そして、アンサーでもリストされたオファーの中で最も推奨するフォーマットを使用すべき [SHOULD] (ストリームの送信が可能と仮定するならば) で、たとえ相手からの前の SDP で存在したとしても、オファーには無いフォーマットを使用して送ってはならない [MUST NOT]。

同様に、オファー側がアンサーを受信した時に、アンサーにあるどれかのフォーマットを使用してメディアを送信し始めなければならない [MUST]、最も推奨する一つを使用すべきであり [SHOULD] (ストリームの送信が可能と仮定するならば)、たとえ相手からの前の SDP で存在したとしても、アンサーにはないフォーマットを使用して送ってはならない [MUST NOT]。

エージェントがメディアフォーマットの使用を停止する時(前の SDP にはあったが、オファー/アンサーではフォーマットが記載されていないことによる)、エージェントはそのまま短時間、メディアを受信するための用意ができていなければならない必要がある。

そのフォーマットで受信を停止する準備ができた時をどのように知るのか？

もし、知る必要があるのならば、適用できる技術が3つある。

第1は、エージェントはフォーマットを変更することに加え、ポートを変えることができる。

メディアが新規ポート上で到着した時、対向側が古いフォーマットでの送信を停止したことを知り、それで受信の準備を停止することができる。

このアプローチは、独立しているメディアフォーマットである利点がある。しかしながら、ポートの変更はリソース予約の変更や、セキュリティプロトコルの再キーイングが必要になるかもしれない。

第2のアプローチは、1つが廃棄されるとき、全てのコーデックに対して動的なペイロードタイプの全く新しいセットを使用することである。

メディアが新しいペイロードタイプの1つを受信したとき、エージェントは対向側が古いフォーマットでの送信を停止したことを知る。このアプローチは、予約、またはセキュリティ関連には影響しないが、RTP固有のものであり、非常に小さいペイロードタイプスペースの無駄がある。

第3のアプローチは、タイマーを使用する事である。

対向側から SDP を受信した時、タイマーがセットされる。

タイマ満了した時、エージェントは古いフォーマットを受信するための準備を停止することができる。

1 分の値で通常は十分である。場合によっては、エージェントは考慮しないかもしれず、従って、古いフォーマットで受信するための準備を継続している。この場合、何もする必要が無い。

もちろん、オファーしたストリームが拒否された場合、オファーは拒否が受信されてすぐに新規のフォーマットを使用した受信準備を停止することができる。

3.12.3. メディアタイプの変更

ストリームに対するメディアタイプ (オーディオ、ビデオ等) は変更されてもよい [MAY]。

同じ論理データが異なったメディアフォーマットで伝達される場合、メディアタイプを変更する (新規のストリームを追加することとは対照的に) ことを推奨する [RECOMMENDED]。

これは両方が別個のメディアタイプである、音声帯域ファックスとファックスが1つのストリームの中で取り換えるために、大いに有効である。

これを実施するために、オファー側は、変更することになる、前の SDP の記述に代わって、新規メディアタイプを持つ新規のメディア記述を作成する。

アンサーで一致したメディアストリームはセクション 3.5 で説明されているように定式化される。ストリームが受け入れられると仮定して、アンサー側はオファーを受信するとすぐに、新規のメディアタイプとフォーマットで送り始めるべきである [SHOULD]。オファー側は、アンサーが受信されて、新規のタイプのメディアを受信し再生バッファの先頭に到達するまで、古いタイプと新しいタイプの両方でメディアを受信する準備をしなければならない [MUST]。

3.12.4. 属性変更

メディア記述のその他属性もオファーまたはアンサーで更新されてもよい [MAY]。一般に、エージェントは変更がある SDP が受信されると、新しいパラメータを使用してメディアを送信しなければならない [MUST] (ストリームの方向性が許されているならば)。

3.13. ユニキャストストリームの保留

通話中の一方がもう一方の相手を「保留」にした場合、言い換えれば、1 つまたはそれ以上のユニキャストメディアストリームの送信を一時的に止めることを要求するならば、更新された SDP をもう一方に提案する。

保留するストリームが、前は `sendrecv` メディアストリームであった場合、`sendonly` を指定することによって保留にされる。

保留するストリームが、前は `recvonly` メディアストリームであった場合、`inactive` を指定することによって保留にされる。

これは、ストリームは各方向で別々に「保留」にされることを意味する。

各ストリームは独立して「保留」される。

保留のストリームのオファーの受信者は保留に対応するストリームのアンサーを自動的に返すべきではない [SHOULD NOT]。

「保留」のすべてのストリームの SDP は、保留 SDP と呼ばれる。

アンサー側が保留 SDP に対して、保留 SDP で答えるとき、いくつかのサードパーティコールコントロール、のシナリオは機能しない。

一般に、ユーザが保留ボタンを「押す」とき、エージェントは SDP で全てのストリームが `sendonly` の方向性を示すオファーを生成し、ローカル側でミュートするだろう。そのため、メディアが末端には送られず、メディアは再生されない。

RFC2543 は、ユーザを保留にするには、接続アドレスを 0.0.0.0 に設定することで実現すると規定している。ただし、その通話を保留する用法には注意が必要である。それは RTCP が保留されたストリームとともに使用されることを許容せず、IPv6 では機能せず、接続方向のメディアを切断するためである。

しかしながら、オファー側がメディアストリームとフォーマットの特定のセットを使用したいことを知っているが、オファーの時点ではアドレスとポートを知らない時の最初のオファーでは有益である。

もちろん、使用する時、ポート番号は 0 であってはならない [MUST NOT]、それはストリームが無効にされたことを示す。

エージェントは 0.0.0.0 の接続アドレスで SDP を受けることができなければならない [MUST] し、それは RTP も RTCP も対向に送信してはならないことを意味する。

3.14. 能力表示

エージェントがオファーを送信する前に、オファーにあるメディアフォーマットがアンサー側で受け入れられるかを知ることは有効である。

SIPのような、いくつかのプロトコルはこのような能力を問い合わせる手段を提供する。

このセクションではそのような SDP メッセージがどのようにフォーマットされるのかを述べる。

SDP は、このような問合せに応答することで能力を表示するために使用できる。

このセクションは、SDP メッセージがどのようにフォーマットされるかについて述べる。

SDP にはメッセージが能力表示の目的であることを示す方法がないので、これは上位レイヤプロトコルの文脈から決定される。

能力を示す基本的な SDP の機能は、非常に限られている。

それは許容パラメータ範囲や値を表すことができなくて、オファーアンサーを平行に実施することもできない。

拡張機能として、将来はこのような制限に対処することがある。

メディア機能を示すために作られる SDP は、以下の通りに構築される。

それは、「e=」と「p=」行の両方を省略してもよい [MAY] ことを除いて、有効な SDP でなければならない [MUST]。

そして、「t=」行は、「0 0」と等しくなければならない [MUST]。

各メディアタイプはエージェントでサポートされ、そのタイプの対応するメディア記述が無ければならない [MUST]。

origin フィールドのセッション ID は、メディア機能を表示するために構築された各 SDP は特殊でなければならない [MUST]。

ポートは 0 にセットされなければならない [MUST] が、接続アドレスは任意である。

ポート 0 の使用は、能力のためにフォーマットされた SDP をオファーまたはアンサーとして解釈された場合に、メディアストリームを確立する結果とならないようにする。

「m=」行の transport 構成要素は、メディアタイプの transport を示す。

エージェントでサポートされたタイプの各メディアフォーマットは、「m=」行でリストされたメディアフォーマットがあるべきである [SHOULD]。

RTP の場合、動的ペイロードタイプが使用された場合、rtpmap 属性は特定のフォーマットにタイプを結びつけるために存在しなければならない [MUST]。

特定のコーデックについて、多くの同時ストリームをサポートできるかといった制限を表示する方法は無い。

```
v=0
o=carol 28908764872 28908764872 IN IP4 100.3.6.6
s=-
t=0 0
c=IN IP4 192.0.2.4
m=audio 0 RTP/AVP 0 1 3
a=rtpmap:0 PCMU/8000
a=rtpmap:1 1016/8000
a=rtpmap:3 GSM/8000
```



```
m=video 0 RTP/AVP 31 34
a=rtpmap:31 H261/90000
a=rtpmap:34 H263/90000
```

図 1: 能力を示す SDP

図 1 の SDP は、エージェントが 3 つのオーディオコーデック (PCMU、1016、および GSM) と 2 つのビデオコーデック (H.261 と H.263) をサポートできることを示す。

3.15. オファー/アンサーの交換の例

このセクションはオファー/アンサー交換例を提供する。

3.15.1. 基本的な交換

発呼側のアリスが、彼女のオファーとして以下の記述を入れると仮定する。

それは双方向のオーディオストリームと 2 つの双方向のビデオストリームを含み、H.261 (ペイロードタイプ 31) と MPEG (ペイロードタイプ 32) を使用する。

オファー SDP は以下の通り。

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
s=-
c=IN IP4 host.anywhere.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

被呼側のボブは最初のビデオストリームを受信、または送信したくないので、彼はアンサーとして以下の SDP を返す:

```
v=0
o=bob 2890844730 2890844730 IN IP4 host.example.com
s=-
c=IN IP4 host.example.com
t=0 0
m=audio 49920 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

後のどこかの時点で、ボブはオーディオストリームを受信するポート番号を変更する (49920 から 65422 へ) ことを決め、同時にイベント (telephone-event) のために RTP ペイロードフォーマットを使用して、受信専用の追加オーディオストリームを加える。

ボブはオファーで以下の SDP を提供する:

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example.com
s=-
c=IN IP4 host.example.com
t=0 0
m=audio 65422 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 51434 RTP/AVP 110
a=rtpmap:110 telephone-events/8000
a=recvonly
```

アリスは、追加のメディアストリームを受け入れ、以下のアンサーを作成する。:

```
v=0
o=alice 2890844526 2890844527 IN IP4 host.anywhere.com
s=-
c=IN IP4 host.anywhere.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 53122 RTP/AVP 110
a=rtpmap:110 telephone-events/8000
a=sendonly
```

3.15.2. N 個のコーデックから 1 つを選択

一般的な音声圧縮に使用される DSP が埋め込まれた電話でよくあることは、同時に複数のコーデックをサポートすることができるが、一度コーデックが選択されると、運用中に容易に変更することができないということである。

この例は、セッションが最初のオファー-アンサー交換を使用してセットアップし、直後の第 2 によってコーデックのセットを固定する方法を示している。

アリスからボブへの最初のオファーは、DSPで利用可能な3つのオーディオコーデックを持つ一つのオーディオストリームを示す。

ストリームは、コーデックが固定化されるまでメディアを受け取ることができないので、inactiveとして指示する。

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
s=-
c=IN IP4 host.anywhere.com
t=0 0
m=audio 62986 RTP/AVP 0 4 18
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=inactive
```

ボブはPCMUとG.723の間の動的切り替えをサポートすることができる。

それで、彼は以下のアンサーを送る:

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example.com
s=-
c=IN IP4 host.example.com
t=0 0
m=audio 54344 RTP/AVP 0 4
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=inactive
```

アリスはこれらの2つのコーデックのどちらも選択できる。

それで、彼女はsendrecvストリームとして更新したオファーを送る:

```
v=0
o=alice 2890844526 2890844527 IN IP4 host.anywhere.com
s=-
c=IN IP4 host.anywhere.com
t=0 0
m=audio 62986 RTP/AVP 4
a=rtpmap:4 G723/8000
a=sendrecv
```

ボブは単一のコーデックを受け入れます:

```
v=0
o=bob 2890844730 2890844732 IN IP4 host.example.com
s=-
c=IN IP4 host.example.com
t=0 0
m=audio 54344 RTP/AVP 4
a=rtpmap:4 G723/8000
a=sendrecv
```

アンサー側 (ボブ) が N 個うち 1 つのコーデックしかサポートできないなら、ボブは、オファーからコーデックの 1 つを選択して、アンサーにそれをセットするでしょう。
この場合、アリスは、そのコーデックでそのストリームを起動するために **re-INVITE** を実施するでしょう。

最初の交換で "a = inactive" を使用する代わりとして、アリスは全てのコーデックをリストし、ボブからメディアを受信するとすぐに、受信した一つのコーデックに固定して更新したオファーを生成することができる。もちろんボブが N 個のコーデックのうち 1 つのみをサポートする場合、彼のアンサーには 1 つのコーデックしかないだろう。そしてこの場合、一つのコーデックに固定するための **re-INVITE** は不要である。

4. SDP 仕様

SDPセッション記述は、メディアタイプで“application/sdp”と示される。

SDPセッション記述は、全体的にUTF-8エンコーディングされたISO 10646キャラクタセットを使用したテキストからなる。SDPのフィールド名と属性名では、UTF-8のUS-ASCIIサブセットのみを使用する。ただし、テキストフィールドと属性値には、ISO 10646のフル文字セットを使用してもよい [MAY]。完全なUTF-8キャラクタセットを使用するフィールドと属性値は直接比較されることはない。それゆえUTF-8正規化に要求はない。ASN.1やXDRなどのバイナリエンコーディングに対して、テキスト方式が選択された理由はポータビリティを強化するためである、多彩なトランスポート層の使用を可能とし、また柔軟なテキストベースのツールキットを使用してセッション記述の生成と処理を行うことができる。

しかしながら、SDPは、セッション記述の最大許容サイズが制限されている環境で使用されるかもしれないため、エンコーディングは意図的にコンパクトになっている。アナウンスは、信頼性が非常に低いトランスポートであったり、仲介するキャッシュサーバーで壊される可能性があるため、エンコーディングは厳密な順序規則と書式規則で設計されている。これにより不正な形式のセッションアナウンスになったものをエラーとして容易に検出して破棄するためである。これにより受信側が正しいキーを持っていない暗号化済みセッションアナウンスを速やかに破棄することができる。

SDPセッション記述は、複数行のテキストによって構成される

<type>=<value>

<type>は、大文字と小文字を区別できる1文字でなければならない [MUST]。また、<value>は構造化されたテキストであり、<type>によってフォーマットが変わる。一般的に、<value>は、スペース1文字で区切られた複数のフィールド、もしくはフリーフォーマットの文字列である。さらに、特に定義されたものでなければ、大文字と小文字は区別される。ホワイトスペースは、“=”記号の両側に使用してはならない [MUST NOT]。

SDPセッション記述は、セッションレベルのセクションと、それに続く0個以上のメディアレベルのセクションからなる。セッションレベルのパートは“v=”行で始まり、最初のメディアレベルのセクションへと続く。各メディアレベルセクションは“m=”行で始まり、次のメディアレベルのセクションまたはセッション記述全体の最後まで続く。一般的に、セッションレベルの値は、同等のメディアレベルの値で上書きされない場合、すべてのメディアのデフォルト値になる。

各記述の行には必須 (REQUIRED) のものとオプション (OPTIONAL) のものがある。ただし、いずれも本文書で指定した順序どおりに現れなければならない [MUST] (順序を固定化によってエラーの検出が大幅に強化され、パーサをシンプルにできる)。

OPTIONAL の項目は、“*”という印が付ける

Session description

v= (protocol version)

o= (originator and session identifier)
s= (session name)
i=* (session information)
u=* (URI of description)
e=* (email address)
p=* (phone number)
c=* (connection information -- not required if included in
all media)
b=* (zero or more bandwidth information lines)
One or more time descriptions ("t=" and "r=" lines; see below)
z=* (time zone adjustments)
k=* (encryption key)
a=* (zero or more session attribute lines)
Zero or more media descriptions

Time description

t= (time the session is active)
r=* (zero or more repeat times)

Media description, if present

m= (media name and transport address)
i=* (media title)
c=* (connection information -- optional if included at
session level)
b=* (zero or more bandwidth information lines)
k=* (encryption key)
a=* (zero or more media attribute lines)

文字のセットは意図的に少なくされており、拡張する意図はない。SDPパーサは、理解できない文字を含むセッション記述は完全に無視しなければならない [MUST]。属性メカニズム (後述する“a=”) は、SDPを拡張し、特定のアプリケーションやメディアに合わせて対応するための主な手段である。いくつかの属性 (これらは本メモの5章でリストとしてあげる) には意味が定義されているが、アプリケーション、メディア、またはセッション特有ベースの属性を追加してもよい。SDPパーサは、理解できない属性はすべて無視しなければならない [MUST]。

SDPセッション記述には、“u=”行、“k=”行および“a=”行の外部コンテンツを参照するURIを含めることができる。場合によってはこのURIを逆参照し、セッション記述自体を含まないようにすることもできる。

セッションレベルセクションの接続(「c=」)と属性(「a=」)の情報は、メディア記述で同じ名前の接続情報または属性で上書きされない限り、すべてのセッションのメディアに適用される。たとえば、次の例に示

そのような場合は、各メディアは”recvonly”属性を与えられたようにふるまう。

SDPの記述例はつぎのとおり：

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

セッション名や情報などのテキストフィールドは、0x00 (Nul)、0x0a (ASCIIの改行)、および0x0d (ASCIIのキャリッジリターン)を除く任意のオクテットを含めてよいオクテット文字列である。CRLF(0x0d0a)の並びでレコードを終了するが、パーサは1つの改行で終了するレコードを受けつけるべきである [SHOULD]。“a = charset”属性が存在しない場合、このようなオクテット文字列は、UTF-8エンコードされたISO-10646文字を含むものとして解釈しなければならない [MUST] (“a = charset”属性が存在する場合、一部のフィールドに対して異なる翻訳を強制してもよい)。

セッション記述の”o=”行、“u=”行、“e=”行、“c=”行、“a=”行にはドメイン名を含めることができる。SDPで使用するドメイン名は、[9]と[10]に準拠しなければならない [MUST]。Internationalised domain name (IDNs)は、[11]で定義されるASCII Compatible Encoding (ACE)形式で表記しなければならない [MUST]。そして直接UTF-8またはその他のエンコーディングで表記してはならない [MUST NOT] (この要件は、国際対応ドメイン名の開発より前に発行されたRFC2327や他のSDP関連の規格と互換性を保つためである)。

4.1. プロトコルバージョン

```
v=0
```

”v=”フィールドはSession Description Protocolのバージョンを示す。本メモはバージョン0について定義する。マイナーバージョンは存在しない。

4.2. オリジン

```
o=<username> <sess-id> <sess-version> <nettype> <addrtype> <unicast-address>
```

”o=”フィールドはセッションの由来(ユーザの名前とホストアドレス)を示す。また、セッションの識別子とバージョン番号も示す。

<username> はユーザの発信したホストでのログインもしくは発信したホストがユーザIDの概念がない場合

は ”-“ である。<username> はスペースを含んではならない [MUST NOT]。

<sess-id> は <username> のタプルである数字列であり、<sess-id>, <nettype>, <addrtype>, <unicast-address> は世界的に唯一なセッションの ID を作る。<sess-id> のつかう値は生成するツールによっているが、唯一性を保証するために Network Time Protocol (NTP) フォーマットのタイムスタンプを使うことが提案されている。

<sess-version> はセッション記述のバージョンである。その使い方は生成するツールにより、セッションデータが生成される都度、増加し続ける。これも NTP タイムスタンプの仕様が推奨される [RECOMMENDED]。

<nettype> はネットワークタイプを与えるテキストストリングである。 ”IN” が ”Internet” の意味で定義されており、他の値は将来的に登録してよい [MAY]。

<addrtype> は続くアドレスのタイプを示すテキストである。 ”IP4” と ”IP6” が定義されており、他の値は将来登録してよい [MAY]。

<unicast-address> はセッション生成されたマシンのアドレスである。IP4 のアドレスタイプの場合、FQDN もしくはドットで分けられた IPv4 アドレスで表現される。IP6 のアドレスタイプの場合、FQDN もしくは圧縮テキストで表現された IPv6 のアドレスで表現される。IP4 も IP6 も不可能でなければ FQDN が使われるべきである [SHOULD]。その場合、グローバルに唯一のアドレスが置かれる。ローカル IP アドレスはアドレスが意味をもつ範囲を残さなければならないコンテキストで使うべきではない [MUST NOT]。(例として、ローカルアドレスはアプリケーションレベルの参照に含めるべきではない [MUST NOT]。)

一般的に ”o=” フィールドはセッション記述のバージョンに対して、グローバルで唯一な ID であり、version を除いたサブフィールドは合わせてセッションは変更を考慮せずに特定する。

プライバシーの観点から、username と発信者の IP アドレスはわかりにくくすることが望ましい。任意の <username> やプライベートの <unicast-address> が一般的な ”o=” フィールドに使われ、グローバルに唯一のフィールドであることに影響しないマナーで選ばれる。

4.3. セッションネーム

“s=” フィールドはセッションネームである。一つかセッション記述に一つの ”s=” フィールドでなければならない [MUST]。 ”s=” フィールドは空ではならず [MUST NOT]、ISO 10646 characters (さもなければ ”a=charset” attribute を参照) を含むべきである。もしセッションが意味のある名前を持たない場合、 ”s=” が使われるべきである [SHOULD]。(すなわち、セッションネームとして一つのスペース)

4.4. セッションインフォメーション

i=<session description>

“i=” フィールドはセッションに関する情報を提供する。多くとも一つのセッション毎にセッションレベルの ”i=” フィールド、そしてメディアごとに多くともひとつの ”i=” フィールドがなければならない [MUST]。もし ”a=charset” 属性がある場合、それは ”i=” フィールドで使うキャラクタセットが指定される。もし ”a=charset” 属性がない場合、 ”i=” フィールドは UTF-8 エンコードされた ISO 10646 キャラクタを含まなければならない。

一つの ”i=” フィールドはそれぞれのメディア定義のために使われるかもしれない [MAY]。メディア定義では ”i=” フィールドはメディアストリームをラベルする意図で主に存在する。一つのセッションが同じメディアタイプで、一つ以上の区別すべきメディアストリームを持つ場合、有用であろう。例として2つの異なるホワイトボードで一つがスライド、もう一つが質問のフィードバックの場合があげられる。

“i=” フィールドは可読性のあるフリーフォームでセッションかメディアストリームを表現するためにある。自動分析することには向かない。

4.5. URI

u=<uri>

URI は WWW クライアントで使われる Uniform Resource Identifier である。URI はセッションの追加情報を指すべきである。本フィールドはオプション扱いであるが、使う場合メディアフィールドより先に指定しなければならない。セッション記述に対して 1つの URI フィールドだけ許容される。

4.6. E メールアドレスと電話番号

e=<email-address>

p=<phone-number>

“e=” と ”p=” 行は会議の責任者への連絡情報を指定する。email アドレスと電話番号を含めるかどうかは OPTION である。古いバージョンの SDP はこれらのフィールドは必須であったが、多くの場合無視された。この変更は一般的な使用法に合わせたものである。

email アドレスや電話番号が存在する場合、始めのメディアフィールドの前で指定しなければならない。複数の email や電話番号フィールドがセッション記述で指定できる

電話番号は ”+” で始まる国際公衆番号 (ITU-T 勧告 E.164 参照) であるべきで、スペースとハイフンは可読性を高めるために電話番号を分割するために使うことができる。例を以下にあげる。

p=+1 617 555-6011

email アドレスと電話番号は関連する自由なテキストを OPTION で持つことができる。通常、連絡先の人物の名前などである。使うときは括弧で閉じられなければならない。例を以下に示す。

e=j.doe@example.com (Jane Doe)

email アドレスも電話番号も RFC2822 の名前引用も代替手段として許可される。例を以下に示す。

e=Jane Doe <j.doe@example.com>

自由なテキストは UTF-8 エンコードされた ISO10646 キャラクタセットとすべきである。そうでない場合、ISO-8859-1 やほかのエンコーディングが適切なセッションレベルで ”a = charset” 属性として設定される。

4.7. コネクションデータ

c=<nettype> <addrtype> <connection-address>

“c=”フィールドはコネクションデータを含む。

セッション記述は少なくともメディア記述につき一つ以上、もしくはセッションレベルで一つの”c=”行を含まなければならない [MUST]。一つのセッションレベルの”c=”フィールドと、メディア記述についての”c=”フィールドを含み、それぞれのメディアのセッションレベル設定をメディアの値で上書きするかもしれない [MAY]。

一つ目のサブフィールドである”<nettype>“はネットワークタイプであり、ネットワーク種別を与えるテキストである。”IN”は”Internet”を意味し、他の値は将来的に登録されるかもしれない [MAY]。

二つ目のサブフィールドである”<addrtype>”はアドレスタイプである。これはSDPがIPベースではないセッションで使われることを許容する。本メモはIP4とIP6のみを定義し、他の値は将来的に登録されるかもしれない [MAY]

三つ目のサブフィールドである”<connection-address>”はコネクションアドレスである。OPTIONAL サブフィールドは<addrtype> フィールドによってコネクションアドレスのあとに追加されるかもしれない [MAY]。

<addrtype> が IP4 か IP6 の場合、コネクションアドレスは次のように定義される。

- ・セッションがマルチキャストである場合、コネクションアドレスは IP マルチキャストアドレスとなりうる。セッションがマルチキャストでない場合、コネクションアドレスは期待するデータソースかデータリレーかデータシンクのユニキャスト IP アドレスを含む。これらは追加属性フィールドで決められる。

マルチキャストアナウンスで通信されるセッション記述でユニキャストアドレスが与えられることを期待できないが、これは禁止されてはいない。

- ・IPv4 マルチキャストコネクションアドレスをつかったセッションは time to live (TTL) 値をマルチキャストアドレスに加えて持たなければならない。TTL とアドレスはセットでその会議のマルチキャストパケットが送られるスコープを定義する。TTL は 0-255 の範囲でなければならない [MUST]。TTL は指定されなければならない [MUST] が、マルチキャストトラヒックのスコープを定めるのに使うことは非難されている。アプリケーションはスコープが管理されたアドレスを代わりに用いるべきである [SHOULD]。

セッションの TTL はスラッシュをセパレータとして追加される。例を挙げる。

c=IN IP4 224.2.36.42/127

IPv6 マルチキャストは TTL をスコープとして用いない。それゆえ、TTL 値は IPv6 マルチキャストでは存在してはならない [MUST NOT]。スコープの定められた IPv6 アドレスが会議の範囲を制限するのにつかわれることが期待される。

階層的、もしくはレイヤー化されたエンコードスキームはデータストリームであり、一つのメディアソースが複数のレイヤーに分けられエンコードされる。受信者はこれらのレイヤーのサブセットに加入することに

よってのみ望まれる品質(帯域による)を選ぶことができる。レイヤー化されたエンコーディングは通常、複数のマルチキャストグループにマルチキャストの枝刈りを許容する。この技術はある階層レベルの必要とするグループのみ不要なトラヒックをわける。マルチキャストグループを必要とするアプリケーションには、コネクションアドレスとして次のような表記を使うことを許容する。

```
<base multicast address>[/<ttl>]/<number of addresses>
```

もし複数のアドレスが与えられない場合、1つであると想定される。マルチキャストアドレスはベースアドレスより上のアドレスが連続して割り当てられる、したがって次のようになる。

```
c=IN IP4 224.2.1.1/127/3
```

これは 224.2.1.1, 224.2.1.2, and 224.2.1.3 のアドレスが 127 の TTL を使うことを表す。これはセマンティクスとしてメディア記述の中の複数の "c=" 行を含むのと同じである。

```
c=IN IP4 224.2.1.1/127
```

```
c=IN IP4 224.2.1.2/127
```

```
c=IN IP4 224.2.1.3/127
```

同様に IPv6 の例は次のとおり

```
c=IN IP6 FF15::101/3
```

これはセマンティクスとして下記と同等である

```
c=IN IP6 FF15::101
```

```
c=IN IP6 FF15::102
```

```
c=IN IP6 FF15::103
```

(TTL が IPv6 マルチキャストでは存在しない)

複数のアドレスや "c=" 行はメディア毎に指定されるかもしれない [MAY]。そしてそれらは階層的もしくはレイヤー化されたエンコーディングスキームの異なるレイヤーのマルチキャストアドレスを与える。これらはセッションレベルの "c=" 行で指定してはならない [MUST NOT]。

前述した複数アドレスのためのスラッシュ表記は IP ユニキャストアドレスのために使ってはならない [MUST NOT]。

4.8. 帯域幅

```
b=<bwtype>:<bandwidth>
```

この OPTIONAL フィールドは、セッションもしくはメディアで使用される帯域の提案を指定する。<bwtype> は英数字であらわされ、<bandwidth> の意味を与える。2つの値が本仕様で定義されているが、その他の値を将来的に登録してもよい [MAY]。

CT: セッションまたはセッション内のメディアの帯域が、スコープからくる暗黙的な帯域と異なる場合、"b=CT" 行を指定することで、帯域幅の上限を与えるべきである [SHOULD]。この主な目的は、2つもしくは

それ以上のセッションが同時に共存できるかどうかについて、適切な考え方を提供することである。RTPと共にCTを使用し、複数のRTPセッションがカンファレンスの一部である場合、カンファレンス全体は、すべてのRTPセッションの総帯域を参照する。

AS: この帯域は、アプリケーション特有のものに変換される(つまり、アプリケーションの最大帯域の考え方である)。適用可能な場合、通常はアプリケーションの”最大帯域”制御の設定と一致する。RTPベースのアプリケーションでは、[12]の6.2章で定義されているように、ASがRTPの”セッション帯域”を指定する。

CTはすべてのサイトのすべてのメディアの総帯域の桁を指定するという点に注意すること。ASでは、同時に複数サイトの送信がある場合でも、1つのサイトにおける1つのメディアの帯域幅の桁を指定する。

プレフィクス”X-“が<bwtype>名のために定義されている。これは、実験目的にのみ使用することができる。例:

b=X-YZ:128

”X-“プレフィクスの使用は推奨されていない(NOT RECOMMENDED)。代わりに、標準の名前空間でIANAに新しい修飾語を登録すべきである[SHOULD]。SDPパーサは未知の修飾語が含まれる帯域フィールドを無視しなければならない[MUST]。修飾語は、英数字にしなければならない[MUST]。また、長さの制限がない場合でも、短いことが推奨される。

<bandwidth>は、デフォルトでkbpsと解釈される。新しい<bwtype>修飾語の定義で、その他の単位で帯域を翻訳されるように指定してもよい[MAY] (本メモで定義されている”CT”と”AS”修飾語はデフォルトの単位を使用する)。

4.9. タイミング

t=<start-time> <stop-time>

”t=”行は、セッションの開始時間と終了時間を指定する。不定期に複数回の間隔を置いてセッションがアクティブになる場合、複数の”t=”行を使用してもよい[MAY]。追加の各”t=”行で、セッションがアクティブになる各期間を指定する。セッションが規則正しい時間にアクティブになる場合、”t=”行とは別に、続けて”r=”行(後述)を使用すべきである。この場合、”t=”行では、繰り返しシーケンスの開始時間と停止時間を指定する。

最初と2つ目のサブフィールドで、セッションの開始時間と終了時間を指定する。この値は、1900年以降の秒であらわしたNetwork Time Protocol (NTP)の値を10進数で表記したものである[13]。この値をUNIX時間に変換するには、10進の2208988800を減算すればよい。

それ以外の場合、NTPタイムスタンプは64ビット値で表記され、2036年のある時間でラップする。SDPでは任意長の10進表記を使用するため、これで問題が発生することはない(SDPタイムスタンプは、1900年以降の秒数をカウントし続けなければならない[MUST]。NTPは64ビット制限でまるめた値を使用する)。

<stop-time>をゼロに設定した場合、セッションに境界がなくなるが、<start-time>後になるまでアクティブにならない。<start-time>もゼロに設定した場合、セッションは恒久的であるとみなす。

境界がなく永続的なセッションを作成するユーザーインターフェースは、避けるべきである [SHOULD]。これは、セッションが実際に終了する時間に関する情報が与えられず、スケジューリングが困難になるためである。

ユーザーに対して、タイムアウトが指定されていない境界なしのセッションを表示する場合、一般的な想定はなされる。制限なしのセッションは、現在の時間またはセッションの開始時間のうちのどちらか遅いほうから 30 分までアクティブであるという想定である。これ以外の動作が必要な場合、セッションが実際に終了する時間に関する新しい情報が入手できたときに、必要に応じて終了時間を与え、修正すべきである [SHOULD]。

恒久的なセッションは、ユーザーに提示してもよい。これは、関連付けられた繰り返し回数があり、セッションをアクティブになる正確な状態がわからなければ、アクティブにならないためである。

4.10. 繰り返し回数

r=<repeat interval> <active duration> <offsets from start-time>

“r=”フィールドはセッションの繰り返し回数を指定する。たとえば、セッションが、3 か月の間、毎週月曜日と火曜日の午前 10 時に、1 時間アクティブになる場合、関連する “t=”フィールドの <start-time> は最初の月曜日の午前 10 時を表す NTP、<repeat interval> は 1 週間、<active duration> は 1 時間、オフセットはゼロと 25 時間になる。対応する “t=”フィールドの終了時間は、最終セッションである 3 か月後の終了時間の NTP 表記になる。デフォルトで、すべてのフィールドは秒単位である。そのため、“r=”と “t=”フィールドは次のようになる。

t=3034423619 3042462419

r=604800 3600 0 90000

記述をよりコンパクトにするために、時間の単位を日、時間、分で与えてもよい。これらの文法は、数字のすぐ後に大文字小文字を区別するな 1 文字をつづける。

分数の単位は許容されない。代わりに小さな単位を使用すべきである。次の単位を規定する文字が許容されている。

d - days (86400 seconds)

h - hours (3600 seconds)

m - minutes (60 seconds)

s - seconds (allowed for completeness)

したがって前述のセッションアナウンスは、次のように記述することができる

r=7d 1h 0 25h

月単位および年単位の繰り返しは、1 つの SDP 繰り返し回数で直接指定することはできない。代わりに、“t=”フィールドとは別のフィールドをセッション回数を明示的なリストとして使用するべきである。

4.11. タイムゾーン

z=<adjustment time> <offset> <adjustment time> <offset>

繰り返しのセッションをスケジュールする際、サマータイムから標準時間への切り替え期間またはその逆の場合に、ベースの時刻からのオフセットを指定する必要がある。これは、タイムゾーンによって日時が変わり、国によってサマータイムかどうかが変わり、サマータイム制度がない国もあるためである。

したがって、冬と夏の同じ時刻にセッションの予定を立てるには、セッションの予定を立てたタイムゾーンを明確に指定できるなければならない。受信側のこのタスクを単純にするために、本文書では、送信側がタイムゾーンの調整が発生する NTP 時間と、最初にセッションを予定した時点からのオフセットを指定できるようにした。"z=" フィールドを使用して、送信側はこの調整時間のリストと、ベース時間からのオフセットを指定することが許容される。

次に例を挙げる

```
z=2882844526 -1h 2898848070 0
```

これは 2882844526 の時点指定し、セッション繰り返し時間の計算のベースタイムが 1 時間バックするようにシフトする。そして 2898848070 の時点指定して、セッションのオリジナル時間がリストアされる。調整は指定した開始時刻から相対的なものとなり、累積ではない。調整はセッション表現におけるすべての "t=" と "r=" 行に適用される。

セッションが数年間続く場合、1 つのセッションアナウンスは数年に値する送信するのではなく、セッションアナウンスを定期的に修正することが期待される。

4.12. 暗号化キー

```
k=<method>
```

```
k=<method>:<encryption key>
```

安全で信頼できるチャンネルで伝送する場合、セッション記述プロトコルを使用して暗号化キーを伝達してもよい[MAY]。キー交換のシンプルなメカニズムとして、キーフィールド("k=")が提供されているが、主にこれは古い実装との互換性のためにサポートされているため、使用は推奨されない [NOT RECOMMENDED]。SDP と併用するための新しいキー交換メカニズムの定義が作業されており [14] [15]、新しいアプリケーションはこれらのメカニズムを使用することが期待される。

キーフィールドは、必要に応じて、最初のメディアエントリの前 (この場合はセッション内の全てのメディアに適用される)、または各メディアエントリに許容されている。キーの形式とキーの用法は、本文書の範囲外である。また、キーフィールドに、使用する暗号化アルゴリズム、キーの種類、またはキーに関するその他の情報を示す機能はない。このような情報は、SDP を使用するより高レベルのプロトコルで提供されると想定されている。このような情報を SDP 内で伝達する必要がある場合、前述した拡張を使用すべきである [SHOULD]。多くのセキュリティプロトコルでは 2 つのキーを必要とする。1 つは機密保護のためのキー、もう 1 つは整合性のためのキーである。本仕様では、2 つのキーの転送をサポートしない。

```
k=clear:<encryption key>
```

暗号化キーは、このキーフィールドに変換しないで含める。

この方法は、安全なチャネルで SDP が伝達されると保証可能な場合を除き、使用してはならない [MUST NOT]。暗号化キーは、charset 属性に従ったテキストとして翻訳されるため、"k=base64:" メソッドを使用して文字を伝達すること。それ以外の文字列は SDP では禁止されている。

k=base64:<encoded encryption key>

暗号化キーは、このキーフィールドに含まれるが、SDP で禁止されている文字を含むため、base64 でエンコードされている [16]。この方法は、SDP が安全なチャネルで伝達されると保証可能な場合を除き、使用してはならない [MUST NOT]。

k=uri:<URI to obtain key>

Uniform Resource Identifier がキーフィールドに含まれる。URI はキーを含むデータを参照し、キーが返ってくる前に追加で認証が要求されるかもしれない。与えられた URI に対してリクエストが送信された場合、応答ではキーのエンコーディングを指定すべきである。この URI はしばしば Secure Socket Layer/Transport Layer Security (SSL/TLS) で保護された HTTP URI ("https:") であるが、必須ではない。

k=prompt

この SDP 記述にはキーが含まない。このキーフィールドが参照するセッションまたはメディアストリームは暗号化される。ユーザーがセッションに参加しようとする時、キーの入力が要求されるべきである。また、このユーザーが入力したキーは、メディアストリームの復号に使用されるべきである。ユーザー指定したキーの使用は推奨されていない [NOT RECOMMENDED]。そのようなキーはセキュリティが弱いという傾向があるためである。

このキーフィールドは、SDP が安全で信頼できるチャネルで伝達されると保証可能な場合を除き、使用してはならない [MUST NOT]。このようなチャネルの一例としては、S/MIME メッセージまたは TLS で保護された HTTP セッション内に SDP を組み込む方法がある。安全なチャネルがセッションに参加する権限を持つパーティにあるように保証することが重要である。仲介であってはならない。つまりキャッシュするプロキシサーバーを使用する場合、プロキシが信頼されているか、プロキシが SDP にアクセスできないように保証することが重要である。

4.13. 属性

a=<attribute>

a=<attribute>:<value>

属性は、SDP を拡張するための主な手段である。"セッションレベル"属性、"メディアレベル"属性、またはその両方に使用する属性を定義できる。

メディア記述には、メディア特有の属性 ("a=" フィールド) を複数指定してもよい。これは、"メディアレベル"属性と呼ばれ、メディアストリームに関する情報が追加される。属性フィールドは、最初のメディアフィールドの前に追加することもできる。この"セッションレベル"属性は、個々のメディアではなく、カン

ファレンス全体に適用する追加情報を伝達する。

属性フィールドは2つの形式をとりうる

o プロパティの属性は、単純に“a = <flag>”という形式を持つ。これはバイナリの属性であり、この属性があると、この属性がセッションのプロパティであることが伝達される。一例を挙げると、“a = recvonly”のようなものである。

o 値の属性は、“a = <attribute> : <value>”という形式を持つ。一例を挙げると、ホワイトボードが“a = orient:landscape”という値属性を持つ場合などである

属性の翻訳は、実行されるメディアツールによって変わる。したがって受信側のセッション記述は一般的なセッション記述の翻訳と特定の属性の翻訳とが設定できるようにすべきである。

属性名は、ISO-10646/UTF-8のサブセットである US-ASCII の文字を使用しなければならない [MUST]。

属性値はオクテット文字列であり、0x00 (Nul)、0x0A (LF)、および0x0D (CR)を除いた任意のオクテットを使用してもよい (MAY)。デフォルトでは、属性値は、UTF-8 エンコーディングの ISO-10646 文字セットとして翻訳される。他のテキストフィールドとは異なり、属性値は“charset”属性には通常、影響を受けない (NOT)。これは、既知の値に対する比較で問題を起すためである。しかしながら、属性が定義されたとき、charset 依存に定義することができる。このケースにおいて、その値は ISO-10646 ではなくセッションの文字セットで解釈すべきである。

属性は、IANA で登録しなければならない [MUST]。理解できない属性を受信した場合、受信側は無視しなければならない [MUST]。

4.14. メディア記述(「m=」)

m=<media> <port> <proto> <fmt> ...

セッション記述には、複数のメディア記述を含めることができる。

各メディア記述は「m=」フィールドで始まり、次の「m=」フィールドまたはセッション記述の末尾で終了する。メディアフィールドには、複数のサブフィールドが含まれる。

<media> はメディアタイプである。現在定義されているメディアは、「audio」、「video」、「text」、「application」、「message」だが、このリストは将来的に拡張される可能性がある。

<port> は、メディアストリームを送信する送信ポートである。

送信ポートの意味は、関連する「c=」フィールドで指定されているように、使用するネットワークによって変わる。また、メディアフィールドの <proto> サブフィールドで定義される送信プロトコルによっても変わる。メディアアプリケーションで使用するその他のポートは、基本のメディアポートからアルゴリズム的に取得してもよい [MAY]。または、別の属性(「a = rtcp:」など)で指定してもよい [MAY]。

隣接していないポートが使用される場合、または偶数の RTP ポートと奇数の RTCP ポートというパリティ規則に従わない場合、「a=rtp」属性を使用しなければならない [MUST]。アプリケーションが、奇数の <port> にメディアを送信するようにリクエストされ、「a=rtp:」が存在する場合、RTP ポートから 1 を差し引いてはならない [MUST NOT]。つまり、RTP は <port> で指定されたポートに送信し、RTCP は「a=rtp」属性で指定されたポートに送信しなければならない [MUST]。

階層的にエンコードしたストリームがユニキャストアドレスに送信される場合、アプリケーションは複数の送信ポート指定が必要な場合もある。

この場合、「c=」フィールドの IP マルチキャストアドレスと同様の表記方法を使用する。

```
m=<media> <port>/<number of ports> <proto> <fmt> ...
```

このような場合、使用ポートは送信プロトコルによって変わる。

RTP の場合、デフォルトは、データには偶数のポートのみが使用され、その RTP セッションに所属する RTCP には、対応する 1 つ上の奇数ポートが使用される。<number of ports> は、RTP セッションの数を示す。例:

```
m=video 49170/2 RTP/AVP 31
```

これは、ポート 49170 と 49171 が 1 つ目の RTP/RTCP ペアを構成し、49172 と 49173 が 2 つ目の RTP/RTCP ペアを構成することを示す。

RTP/AVP は送信プロトコルであり、31 は形式である (以下を参照)。隣接していないポートが必要な場合、異なる属性を使用してシグナルする必要がある (たとえば、「a=rtp:」など)。

複数のアドレスを「c=」フィールドで指定し、複数のポートを「m=」フィールドで指定する場合、ポートと対応するアドレスに関する 1 対 1 のマッピングされることを表す。例:

```
c=IN IP4 224.2.1.1/127/2  
m=video 49170/2 RTP/AVP 31
```

これは、224.2.1.1 はポート 49170 と 49171 と共に使用され、アドレス 224.2.1.2 はポート 49172 と 49173 と共に使用されることを表す。

同じ送信アドレスを使用した複数の「m=」行の表現は定義されていない。

つまり、制約された過去の慣例とは異なり、このような手段で定義された暗黙的なグルーピング方法はないため、明示的なグルーピングの枠組みを代わりに使用して目的の表現をすべきである。

<proto> は送信プロトコルである。送信プロトコルの意味は、関連する「c=」フィールドのアドレスタイプフィールドによって変わる。

したがって、IP4 の「c=」フィールドは、送信プロトコルが IP4 上で実行されていることを示す。次の送信プロトコルが定義されている。

ただし、IANA で新しいプロトコルが登録されて拡張される可能性がある。

* udp: UDP 上で実行されている、特定されていないプロトコルを示す。

* RTP/AVP: UDP 上で使用されている RTP を示す。

* RTP/SAVP: UDP 上で実行されている、送信プロトコルを示す。

メディア形式とは別に送信プロトコルを指定する主な理由は、ネットワークプロトコルが同じ場合でも、異なる送信プロトコル上で同じ標準のメディア形式が伝達される可能性があるためである。

従来の例を挙げると、vat Pulse Code Modulation (PCM) 音声や RTP PCM 音声がある。その他に、TCP/RTP PCM 音声も考えられる。さらに、トランスポートプロトコル固有だが、形式には非依存のリレーツールと監視ツールが考えられる。

<fmt> はメディア形式の説明である。第 4 以降のサブフィールドは、メディアの形式を記述する。メディア形式の解釈は、<proto> サブフィールドの値によって変わる。

<proto> サブフィールドが「RTP/AVP」または「RTP/SAVP」の場合、<fmt> サブフィールドには RTP ペイロードタイプ番号が含まれる。ペイロードタイプ番号の一覧が指定される場合、そのペイロード形式すべてをそのセッションで使用してもよい [MAY] ことを意味するが、セッションのデフォルト形式としてリストの最初にある形式を使用すべきである [SHOULD]。動的なペイロードタイプ割り当ての場合、「a=rtpmap:」属性を使用して、RTP ペイロードタイプ番号から、ペイロード形式を特定するメディアエンコード名へとマップすべきである [SHOULD]。「a=fmtp:」属性を使用して、特定のパラメータを指定してもよい [MAY]。

<proto> サブフィールドが「udp」の場合、<fmt> サブフィールドは、トップレベルのメディアタイプ「audio」、「video」、「text」、「application」、「message」の下で形式を記述するメディアタイプを参照しなければならない [MUST]。メディアタイプの登録は、UDP トランスポートと共に使用するパケット形式を定義すべきである [SHOULD]。

他の送信プロトコルを使用するメディアの場合、<fmt> フィールドはプロトコル固有である。新しいプロトコルを登録する場合、<fmt> サブフィールドを解釈する規則を定義しなければならない [MUST]。

5. SDP 属性

次の属性が定義される。アプリケーションの作成者は必要に応じて新しい属性を追加してもよいが、この一覧がすべてではない。

5.1. a=cat:<category>

この属性は、ドットで区切った、セッションの階層的なカテゴリである。用途は、受信側がカテゴリを使用して不要なセッションをフィルタするためである。カテゴリを一元管理しているところはない。

セッションレベル属性であり、文字セットに依存しない。

5.2. a=keywds:<keywords>

cat 属性と同様に、これは受信側が必要なセッションを特定するときの補助として使用する属性である。これによって、受信側は、セッションの目的を記述したキーワードに基づいて、関心のあるセッションを選択することができる。キーワードを一元管理しているところはない。

セッションレベルの属性である。この属性は文字セットに依存する。

つまり、この属性値は、セッション記述で文字セットが指定されている場合はその文字セットで解釈し、それ以外の場合はデフォルトの ISO10646/UTF-8 で解釈すべきである。

5.3. a=tool:<name and version of tool>

この属性は、セッション記述を作成するときを使用したツールの名前とバージョン番号を指定する。セッションレベル属性であり、文字セットに依存しない。

5.4. a=ptime:<packet time>

この属性は、パケット内のメディアが表す時間の長さをミリ秒単位で指定する。この属性は、音声データの場合にのみ意味があるが、有用であれば他のメディアタイプで使用してもよい。RTP または vat 音声をデコードするときに ptime を知る必要はない。この属性は、音声のエンコーディング/パケット化に関する推奨として使用するためのものである。メディアレベル属性であり、文字セットに依存しない。

5.5. a=maxptime:<maximum packet time>

各パケットでカプセル化することができるメディアの総量。ミリ秒単位の時間として表される。この時間は、パケット内に存在するメディアの合計時間数として計算すべきである [SHALL]。フレームベースのコーデックの場合、この時間は、フレームサイズの倍数(整数)にすべきである [SHOULD]。この属性は、音声データの場合にのみ意味があるが、有用であれば他のメディアタイプで使用してもよい。メディアレベル属性であり、文字セットに依存しない。この属性は RFC 2327 以降に導入されたということに注意すべきである。それ以前の実装の場合、この属性は無視される。

5.6. a=rtpmap:<payload type> <encoding name>/<clock rate> [<encoding parameters>]

この属性は、RTP ペイロードタイプ番号から、使用するペイロード形式を示すエンコーディング名へとマップする。また、クロックレートとエンコーディングパラメータに関する情報も指定する。メディアレベル属性であり、文字セットに依存しない。

RTP プロファイルで、ペイロードタイプ番号の静的な割り当てをペイロード形式にしてもよいが、「a=rtpmap:」属性を使用して動的に割り当てを実行する方が一般的である。静的ペイロードタイプの一例を挙げ

ると、8 kHz でサンプリングされた u-law PCM コーディングの単一チャンネル音声がある。これは、RTP Audio/Video プロファイルでペイロードタイプ 0 であると完全に定義される。そのため、「a=rtpmap:」属性を使用する必要はない。また、UDP ポート 49232 にこのようなストリームが送信される場合、メディアは次のように指定することができる。

```
m=audio 49232 RTP/AVP 0
```

動的なペイロードタイプの一例を挙げると、16kHz でサンプリングされた、16 ビットリニアエンコーディングのステレオ音声がある。このストリームに動的な RTP/AVP ペイロードタイプ 98 を使用する場合、それを示すために追加の情報が必要である。

```
m=audio 49232 RTP/AVP 98
a=rtpmap:98 L16/16000/2
```

指定するメディア形式ごとに、最大でも 1 つの rtpmap 属性しか定義できない。したがって、次のように指定する。

```
m=audio 49230 RTP/AVP 96 97 98
a=rtpmap:96 L8/8000
a=rtpmap:97 L16/8000
a=rtpmap:98 L16/11025/2
```

動的ペイロードタイプの使用を指定する RTP プロファイルで、そのプロファイルを SDP と共に使用する場
合、有効なエンコーディング名とエンコーディング名を登録する手段の両方または片方を定義しなければ
ならない [MUST]。「RTP/AVP」と「RTP/SAVP」の各プロファイルは、「m=」行で示す最優先のメディアタ
イプにより、エンコーディング名のメディアサブタイプを使用すべきである。

上記の例では、このメディアタイプが「audio/18」と「audio/116」になる。

音声ストリームの場合、<encoding parameters> は、音声チャンネルの数を示す。このパラメータはオプション
[OPTIONAL] であり、チャンネル数が 1 で、追加のパラメータが必要ない場合は省略してもよい。

ビデオストリームの場合、エンコーディングパラメータは現在のところ指定されていない。

将来的に、追加のエンコーディングパラメータを定義してもよい [MAY]。ただし、コーデック固有のパラ
メータは追加すべきではない [SHOULD NOT]。「a=rtpmap:」属性にパラメータを追加する場合、セッシ
ョンに参加する適切なメディアを選択するときに、セッションごとに必要なものを定義すべきである
[SHOULD]。

コーデック固有のパラメータは、他の属性（「a=fmtp:」など）で追加すべきである。

注意: RTP 音声形式には、一般的に、パケットごとのサンプル数に関する情報は含まれない。デフォルト以
外 (RTP Audio/Video Profile の定義による) のパケット化が必要な場合、「ptime」属性を前述のように使用
する。

5.7. a=recvonly

受信のみのモードであることを指定する。セッションレベル属性とメディアレベル属性のどちらも可能であり、文字セットに依存しない。

recvonly はメディアにのみ適用され、関連する制御プロトコルには適用されないということに注意すべきである (たとえば、recvonly モードの RTP ベースシステムでも、RTCP パケットを送信すべきである [SHOULD])。

5.8. a=sendrecv

送受信モードであることを指定する。

この属性が必要なのは、受信のみのモードがデフォルトの場合と、対話式的カンファレンスを行う場合である。セッションレベル属性とメディアレベル属性のどちらも可能であり、文字セットに依存しない。

「sendonly」、「recvonly」、「inactive」、「sendrecv」のどの属性も存在しない場合、カンファレンスタイプが「broadcast」または「H332」でないセッションには、「sendrecv」をデフォルトと仮定すべきである [SHOULD] (以下を参照)。

5.9. a=sendonly

送信のみのモードであることを指定する。

例としては、異なるユニキャストアドレスを使用した発信先へ接続する場合は考えられる。

このような場合、2つのメディア記述を使用し、1つは sendonly にし、もう1つは recvonly にする。セッションレベル属性とメディアレベル属性のどちらも可能だが、通常はメディア属性としてのみ使用される。文字セットには依存しない。sendonly はメディアのみに適用されるため、関連する制御プロトコル (RTCP など) は通常どおり受信され、処理されるべきである [SHOULD] ということに注意すべきである。

5.10. a=inactive

inactive モードであることを指定する。

これが必要なのは、対話式カンファレンスで、ユーザーが別のユーザーを保留する場合である。inactive のメディアストリームでは、メディアが送信されない。RTP ベースのシステムは、inactive で起動した場合でも、RTCP を送信すべきである [SHOULD] ということに注意すべきである。セッションレベル属性とメディアレベル属性のどちらも可能であり、文字セットに依存しない。

一般に、ユーザーが保留を「押す」場合、エージェントは sendonly の方向を示す SDP 内ですべてのストリームとともにオファーを生成し、さらに、ローカルをミュートするだろう。なぜなら、メディアは向こう側に送信されないし、メディアは再生されないからである。

RFC2543 [17] は、ユーザーを保留中にすることは、接続アドレスを 0.0.0.0 に設定することで実現すると規定している。通話を保留中にするその用法はもう推奨されない。なぜなら、それは、RTCP が保留されたストリームとともに使用されることを許容せず、IPv6 では機能せず、基点のメディアとの接続を切るためである。だがこれは、オファー側が特定のメディアストリームとフォーマットのセットを使用したいことがわかっているが、オファー時にアドレスとポートを知らない場合、最初のオファーでは有益である。もちろん、使用されるときにポート番号は 0 ではない [MUST NOT]。それは、ストリームが使用不可であることを示すためである。エージェントは、RTP も RTCP もピアに送信されないような場合であっても、0.0.0.0 の接続アドレスを持つ SDP を受信できなければならない [MUST]。

5.11. a=orient:<orientation>

通常、ホワイトボードツールまたはプレゼンテーションツールにのみ使用される。画面上の作業スペースの向きを指定する。メディアレベルの属性である。使用できる値は、「portrait」、「landscape」、「seascape」(landscape (横長) の上下反対)。文字セットには依存しない。

5.12. a=type:<conference type>

カンファレンスの種類を指定する。提案されている値は「broadcast」、「meeting」、「moderated」、「test」、「H332」である。「type:broadcast」セッションの場合、「recvonly」をデフォルトにすべきである。「type:meeting」は「sendrecv」として動作すべきである。「type:moderated」は、フロア制御ツールの使用を示し、さらにカンファレンスに参加する新しいサイトをミュートするようにメディアツールを起動することを示すべきである。

「type:H332」属性を示す場合、ITU H.332 仕様の定義どおり、この疎結合のセッションが H.332 セッションの一部であることを示す。

メディアツールは「recvonly」で起動すべきである。

「type:test」属性を指定する場合、明示的にリクエストされていない限り、受信側がこのセッション記述をユーザーに表示するのを防ぐことができることを示す。

type 属性はセッションレベル属性であり、文字セットに依存しない。

5.13. a=charset:<character set>

セッション名と情報を表示するときに使用する文字セットを指定する。デフォルトで、UTF-8 エンコーディングで設定された ISO-10646 文字が使用される。よりコンパクトな表記が必要な場合、別の文字セットを使用してもよい。

たとえば、次の SDP 属性では ISO 8859-1 が指定されている。

```
a=charset:ISO-8859-1
```

これはセッションレベル属性であり、文字セットに依存しない。指定する文字セットは、ISO-8859-1 など、IANA に登録されているものでなければならない [MUST]。文字セットの識別子は US-ASCII 文字列であり、大文字と小文字を区別する比較方法で、IANA の識別子と比較しなければならない [MUST]。識別子を認識できない場合、またはサポートしていない場合、その識別子の影響を受ける文字列はすべてオクテッ文字列として扱うべきである [SHOULD]。

指定する文字セットに、0x00 (Nul)、0x0A (LF)、および 0x0d (CR) を使用することは禁止しなければならない [MUST] ということに注意すべきである。文字セットでこれらの文字を使用する必要がある場合、これらのバイトがテキストフィールドに表示されないように引用メカニズムを定義しなければならない [MUST]。

5.14. a=sdplang:<language tag>

セッションレベル属性とメディアレベル属性のどちらも可能である。

セッションレベル属性としては、セッション記述の言語を指定する。

メディアレベル属性としては、このメディアと関連するメディアレベルの任意の SDP 情報フィールドについて、言語を指定する。セッション記述で複数の言語を使用する場合、またはメディアで複数の言語を使用する場合、セッションレベルまたはメディアレベルで、複数の `sdplang` を指定することができる。この場合、属性の順序は、セッションまたはメディア内の言語の重要度を示す。最重要の言語から重要度が低い言語の順である。

一般的に、複数言語で構成されたセッション記述を送信することは推奨されていない。そうではなく、1つのセッションで1つの言語を記述して、複数の記述を送信すべきである [SHOULD]。

ただし、この方法はどの送信手順でも可能ではない。

そのため、複数の `sdplang` 属性は許容されているが、推奨されていない [NOT RECOMMENDED]。

「`sdplang`」属性値は、US-ASCII で記載した、単一の RFC 3066 言語タグにしなければならない。文字セット属性には依存しない。受信者の言語を想定できない場合、またはセッションがローカルで想定した標準とは異なる言語で行われる場合、セッションの範囲がそのように地理的境界を越えるときには、「`sdplang`」属性を指定すべきである [SHOULD]。

5.15. `a=lang:<language tag>`

セッションレベル属性とメディアレベル属性のどちらも可能である。

セッションレベル属性としては、記述するセッションのデフォルト言語を指定する。メディアレベル属性としては、そのメディアの言語を指定する。セッションレベル言語が指定されている場合、メディアレ

ベル属性の方が優先される。セッション記述またはメディアで複数言語を使用する場合、セッション記述またはメディアで複数の言語を使用することができる。この場合、属性の順序は、セッションまたはメディア内の言語の重要度を示す。最重要の言語から重要度が低い言語の順である。

「`lang`」属性値は、US-ASCII で記載した、単一の RFC 3066 言語タグにしなければならない。文字セット属性には依存しない。受信者の言語を想定できない場合、またはセッションがローカルで想定した標準とは異なる言語で行われる場合、セッションの範囲が地理的境界を越えるときには、「`lang`」属性を指定すべきである [SHOULD]。

5.16. `a=framerate:<frame rate>`

フレーム/秒単位で最大のビデオフレームレートを指定する。ビデオデータのエンコーディングに関する推奨として使用するためのものである。「<整数部>.<小数部>」という表記方法を使用して、十進の小数値を表記することができる。メディアレベル属性であり、ビデオメディア用にのみ定義されている。文字セットには依存しない。

5.17. `a=quality:<quality>`

整数値で、エンコーディングの品質に関する推奨を示す。ビデオの場合、`quality` 属性の用途は、フレームレートと静止画の品質に関して、デフォルトではないトレードオフを指定することである。ビデオの場合、0～10の範囲の値があり、次の意味が提案される。

10 - 圧縮スキームで指定できる最高の静止画像品質。

5 - 品質に関して何も提案しないデフォルトの動作。

0 - コーデック設計者が想定する最低の静止画像品質も使用することができる。

メディアレベル属性であり、文字セットに依存しない。

5.18. a=fmtp:<format> <format specific parameters>

この属性は、SDP が理解する必要のない方法で伝達される、特定の形式に固有のパラメータを許容する。この形式は、メディアに指定された形式の 1 つでなければならない。形式固有のパラメータは、この形式を使用するメディアツールに変更がなければ、SDP が伝達する必要がある任意のパラメータでもよい。1 つの形式ごとに、この属性の 1 つの形式のみが許容されている。

メディアレベル属性であり、文字セットに依存しない。