

## JT-H501

# マルチメディアシステムにおける ドメイン内/ドメイン間通信及び モビリティ管理プロトコル

Protocol for mobility management and  
INTRA/inter-domain communication in multimedia  
systems

第1版

2004年4月20日制定

社団法人  
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、(社)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(社)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 > .....	4
要約 .....	5
1 適用範囲 .....	6
2 参照 .....	6
3 定義 .....	7
4 記号と略語 .....	7
5 要求条件 .....	7
5.1 トランスポート要求 .....	7
5.2 セキュリティに関する考察 .....	8
5.3 アドレッシング規則 .....	8
5.4 アドレステンプレートと記述子 .....	8
6 メッセージの定義 .....	9
6.1 記述子 .....	10
6.2 記述子情報 .....	10
6.3 アドレステンプレート .....	10
6.3.1 パターン .....	11
6.3.2 ルーティング情報 .....	11
6.3.3 課金情報 .....	11
6.3.4 接続情報 .....	12
6.4 共通構造体 .....	13
6.4.1 <i>AlternateBE</i> .....	13
6.4.2 <i>PartyInformation</i> .....	13
6.4.3 <i>CallInformation</i> .....	13
6.4.4 <i>UserInformation</i> .....	13
6.4.5 <i>Usage Specification</i> .....	13
6.4.6 <i>Security Mode</i> .....	14
6.5 サービス関係 .....	15
6.5.1 <i>Service Request</i> .....	15
6.5.2 <i>Service Confirmation</i> .....	15
6.5.3 <i>Service Rejection</i> .....	15
6.5.4 <i>Service Release</i> .....	16
6.6 記述子の配布 .....	16
6.6.1 <i>Descriptor Request</i> .....	17
6.6.2 <i>Descriptor Confirmation</i> .....	17

6.6.3 Descriptor Rejection.....	17
6.6.4 Descriptor ID Request.....	17
6.6.5 Descriptor ID Confirmation.....	18
6.6.6 Descriptor ID Rejection.....	18
6.6.7 Descriptor Update.....	18
6.6.8 Descriptor Update Acknowledgement.....	18
6.7 アクセスのための解決.....	19
6.7.1 Access Request.....	19
6.7.2 Access Confirmation.....	19
6.7.3 Access Rejection.....	19
6.8 REQUEST IN PROGRESS.....	20
6.9 非標準メッセージ.....	20
6.9.1 Non-Standard Request.....	20
6.9.2 Non-Standard Confirmation.....	20
6.9.3 Non-Standard Rejection.....	20
6.10 UNKNOWN MESSAGE RESPONSE.....	21
6.11 利用状況報告.....	21
6.11.1 Usage Request.....	21
6.11.2 Usage Confirmation.....	21
6.11.3 UsageRejection.....	21
6.11.4 Usage Indication.....	22
6.11.5 Usage Indication Confirmation.....	23
6.11.6 Usage Indication Rejection.....	23
6.12 検証.....	23
6.12.1 Validation Request.....	23
6.12.2 Validation Confirmation.....	24
6.12.3 Validation Rejection.....	24
6.13 認証.....	24
6.13.1 Authentication Request.....	24
6.13.2 Authentication Confirmation.....	25
6.13.3 Authentication Rejection.....	25
<b>付属資料 A : メッセージシンタックス ( ANNEX A: MESSAGE SYNTAX ) .....</b>	<b>26</b>

マルチメディアシステムにおけるドメイン内/ドメイン間通信及びモビリティ管理プロトコル

< 参考 >

1. 国際勧告などとの関連

本標準は、2002年xx月のITU-T SG16 会合において承認されたITU-T 勧告 H.501 に準拠している。

2. 上記国際勧告などに対する追加項目など

2.1 オプション選択項目

なし

2.2 ナショナルマター決定項目

なし

2.3 その他

なし

3. 改版の履歴

版数	発行日	改版内容
第1版	2004年4月20日	制定

4. 工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になれます。

5. その他

(1) 参照している勧告、標準など

TTC標準 : JT-H323、JT-H225.0

ITU-T勧告 : H.235、X.680、X.691

IETF標準 : RFC2401、RFC2402、RFC2406

## マルチメディアシステムにおけるドメイン内/ドメイン間通信及びモビリティ管理プロトコル

### PROTOCOL FOR MOBILITY MANAGEMENT AND INTRA/INTER-DOMAIN COMMUNICATION IN MULTIMEDIA SYSTEMS

#### 要約

本標準の目的は、モビリティ管理と、モバイルまたは非モバイル環境のドメイン内及びドメイン間通信のための、アドレス解決、ユーザ認証、サービスデータ交換、アクセス認証、呼の承認および利用状況報告を目的としたメッセージと手順を規定することである。

## 1 適用範囲

本標準は、ある種の論理的エレメントにより管理されるユーザへの及び（ユーザ）からの呼の終端を許可するためのマルチメディアパケット網の論理的エレメント間通信用プロトコルを記述している。このプロトコルは、アドレス解決、ユーザ認証、サービスデータ交換、アクセス認証、呼の承認及び利用状況報告のために、モバイル用及び非モバイル環境で使用することができる。これらの能力は、このプロトコルを、モバイル環境でのモビリティ管理で使用することを可能にする。

基本的な手順は、各管理ドメインが解決できるアドレスの形式中にあるユーザまたはエンドポイントの位置（ロケーション）に関して、論理的エレメントが情報交換することである。アドレスは、一般的な方法またはより具体的な方法によって明示されることができる。追加情報は、通話における宛先として動作するために最適な管理ドメインを決定することを、管理ドメイン内のエレメントに可能にする。論理的エレメントは、自身の公開アドレスへのアクセス制御を行ってもよく、また、これらのアドレスを介した通話の継続中に利用状況に関する報告を要求しても良い。

本標中で定義されるプロトコルを特定のアプリケーションで使用方法は、その他の標準（または勧告）で規定される。一つのアプリケーションが、全てのプロトコルを実装する必要はない。そのアプリケーションが、その要求条件に適したものを、メッセージと手順から選択することができる。

## 2 参照

以下の TTC 標準、ITU-T 勧告及びその他の参照が、この文書中の参照を通し、それぞれの規定を含んで、本標準の規定を構成している。版番号は、（本標準の）制定時に有効だったバージョンを示している。全ての標準、勧告及びその他の参照は、改版されることがある。そのため、本標準の全てのユーザは、以下にリストされた標準、勧告及びその他の参照の最新版の適用可能性について調査するよう努めるのがよい。現在有効な TTC 標準及び ITU-T 勧告のリストは、定期的に出版される。

- [1] TTC 標準 JT-H225.0 第 5 版（2003）, “パケットに基づくマルチメディア通信システムのためのシグナリングプロトコルとメディア信号のパケット化”
- [2] ITU-T Recommendation H.235 Version 2 (2000), “Security and Encryption for H Series (H.323 and Other H.245 Based) Multimedia Terminals”
- [3] TTC 標準 JT-H323 第 5 版（2003）, “パケットに基づくマルチメディア通信システム”
- [4] ITU-T Recommendation X.680 (1997), Information Technology – Abstract Syntax Notation One (ASN.1) Specification of Basic Notation”
- [5] ITU-T Recommendation X.691 (1997), “Information Technology – ASN.1 Encoding Rules – Specification of Packed Encoding Rules (PER)”
- [6] TTC 標準 JT-H323 付属資料 K “JT-H323 における”HTTP サービス制御伝送チャネル”（ TTC 標準 JT-H323 第 5 版（2003））
- [7] IETF RFC 2401, “Security Architecture for the Internet Protocol ” (November 1998)
- [8] IETF RFC2402, “ IP Authentication Header ” (November 1998)
- [9] IETF RFC2406; “ IP Encapsulating Security Payload ” (November 1998)

### 3 定義

管理ドメイン (Administrative domain)	管理ドメインは、1つの管理エンティティによって管理された論理的または物理的エンティティの集合である。管理ドメインは、1つまたは複数のゾーンから構成できる。
ゲートキーパー (Gatekeeper)	管理ドメイン内にて、他のエンティティに対し、特定のサービス(番号変換、アクセス制御等)を提供する論理的エレメント(JT-H323 参照)。
論理的エレメント (Logical element)	ネットワーク内で、機能性を定義されたエンティティ。論理的エレメントは、その規定として、いかなる要求条件も強制しない。それらの機能は、ハードウェアまたはソフトウェアによる適切な方法で実装することができる。
同位エレメント (Peer element)	本標準にて定義されるシグナリングメッセージを生成または終端する論理的エレメント。例としては、JT-H225.0 付属資料 G の同位エレメントや、JT-H323 のゲートキーパーなど。
ゾーン (Zone)	ただ一つのゲートキーパーの制御下にある管理ドメインのエンティティのサブセット。

### 4 記号と略語

AD	管理ドメイン (Administrative domain)
DNS	ドメインネームシステム (Domain Name System)
IP	インターネットプロトコル (Internet Protocol)
PDU	プロトコルデータユニット (Protocol Data Unit)
RAS	登録、承認、状態表示 (Registration, Admission and Status protocol)
SCN	回線交換網 (Switched Circuit Network)
TCP	伝送制御プロトコル (Transmission Control Protocol)
TPKT	トランスポートパケット (Transport Packet)
UDP	ユーザデータグラムプロトコル (User Datagram Protocol)
UTC	世界標準時 (Coordinated Universal Time)

### 5 要求条件

#### 5.1 トランスポート要求

メッセージは、信頼性のない転送サービス(例:UDP)または信頼性のある転送サービス(例:TCP)上で、公知のアドレスに対して送ってもよい。IPネットワークの場合、別のポートが送信側から通知されない限り、公知のポート(2099)をTCPとUDPの両方に対して使うべきである。各要素(エレメント)は、TCPとUDPの両方のポートを受信監視(listen)しなくてはならない。

メッセージが信頼性のある転送サービスで送られる場合、全てのメッセージは、信頼性のある転送プロトコルデータユニット(PDU)により定められた境界内で送られなければならない。(IPにおける実装では、例えば、JT-H225.0 付録 で概略が示されているように、このPDUはTKPTで境界を定められる(図1/JT-H501を参照)。各JT-H501 PDUは、本標準で定義されるメッセージを、ただ一つだけ含む。)

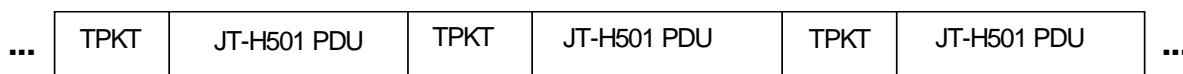


図 1/JT-H501 - TCP トランスポート  
(ITU-T H.501)

信頼性のない転送サービスを使用する場合、要求メッセージは、再送してもよい。再送タイマーのデフォ



ルト値は、適応的に遅延を検知する方法（例えば、TCP プロトコルを使う方法など）により決められるべきである。再送間隔は、指数関数的に間隔をあけるようにしなくてはならない。再送回数は、5 回を超えないようにしなくてはならない。応答メッセージ類は、再送してはならない。

UDP/IP における実装では、メッセージは、複数のメッセージを 1 個のパケットに格納するため、TKPT によりプリフィックスされなくてはならない（図 2/JT-H501 を参照）。各 JT-H501 PDU は、本標準で定義されるメッセージを、ただ一つだけ含む。UDP パケット長フィールドは、全てのメッセージとその TPKT ヘッダを含む、ペイロードのトータルレングス長を持っていないなくてはならない。

UDP Header	TPKT	JT-H501 PDU	TPKT	JT-H501 PDU	TPKT	JT-H501 PDU
------------	------	-------------	------	-------------	------	-------------

図 2/JT-H501 - UDP データグラムフォーマット  
(ITU-T H.501)

## 5.2 セキュリティに関する考察

同位エレメント間で交換されるメッセージに対して、認証、完全性、トランスポートレイヤでの暗号化が要望されるとき、IETF RFC2402(“IP Authentication Header”)と RFC 2406 (“IP Encapsulating Security Payload”)のいずれかまたは両方を含む RFC2401 (“Security Architecture for the Internet Protocol”)によって記述される IP セキュリティの動作をしなくてはならない。

アプリケーションレイヤでの防御が要求される場合は、H.235 の手順と構成がアプリケーションレベルのセキュリティをサポートするために使用されなくてはならない。特に、トークンフォーマットと認証の交換が用いられなくてはならない。応答メッセージにおいて受信されたトークンと暗号化トークンは、後に続く関連する要求メッセージで用いられるべきである。

## 5.3 アドレッシング規則

ドメイン間に互換性を提供するために、シグナリングメッセージで送信されたアドレッシングフォーマットが受信側システムによって理解されることは重要である。同位エレメントは、アプリケーションがグローバルに使用してもよい別名アドレス (*AliasAddress*) の全タイプをサポートしなくてはならない。

最低限、電子メール ID (*email-id*) と *partyNumber* (*internationalNumber* を要素に持つ *PublicTypeOfNumber* からなる *PublicNumber* を使用) フォーマットの両方をサポートしなくてはならない。他の同位エレメントと通信する時、関連する管理ドメイン間で、他のフォーマットの使用について事前の合意がなければ、電子メール ID (*email-id*) と別名アドレス (*AliasAddress*) の *partyNumber* タイプだけが使われるべきである。例えば、もし管理ドメインのグループが私設番号計画の解釈について合意したなら、これらの番号をメッセージで使ってもよい。

モバイル環境では、関連する管理ドメイン間の合意の上で、グローバルなユーザ識別モジュールとして、*mobileUIM* フォーマットが使われてもよい。このフォーマットは、TTC 標準 JT-H225.0 に詳細が記述されている。

## 5.4 アドレステンプレートと記述子

アドレステンプレート（略して「テンプレート」）は、別名アドレス (*AliasAddress*) 識別子のセット、それらのアドレスへの呼を完了するための料金情報、そしてセットの到達アドレスで使われるプロトコルを定義する。管理ドメインは、解決することができる呼を示すためのテンプレートを公表する。テンプレートは「記述子」として既知の識別子によって集められる。いったんテンプレートが記述子によって分類されると記述子のテンプレートに対するいかなる変更でも記述子「グループ」に対する変更を意味する。テンプレート情報は、もしアドレッシング体系が階層型構造か、もしくは配送可能な方法で整えられるなら、アドレッシング情報の結合を許してもよい。（例えば、所定のゾーンが、1303538 から始まるすべての電話番号を意味して、1303538\*を処理するかもしれない）。（注：「\*」が意味のある文字であるため、実際にはテンプレートは特定のアドレスまたはワイルドカードで示されるアドレスであるかどうかを示すための明示的なフィールドを含む。これらの例はワイルドカードを示すために「\*」を使うが、テンプレートでの実際の表記は明示的なフィールドを通してである。「\*」がワイルドカードを示す場合、それは回線には送信されない。）

テンプレート例：

「1 555 123 4567 は、AccessRequest メッセージを同位エレメント A に送る」

- 「1 555 987\*は、 AccessRequest メッセージを同位エレメント B に送る」
- 「1 555 987 6543 は、呼設定メッセージをゲートウェイ X に送る」
- 「 \*@example.org は、 AccessRequest メッセージを同位エレメント A に送る」
- 「1 \*は、 AccessRequest メッセージを同位エレメント B に送る」
- 「私設 31\*は、 AccessRequest メッセージを同位エレメント C に送る」
- 「44 171 112\*は、存在しない」

## 6 メッセージの定義

本節では、本標準で定義されるプロトコルのメッセージと要素（エレメント）を規定している。付属資料 A は、勧告 X.680 に従う、対応する ASN.1 定義を含んでいる。各メッセージは、勧告 X.691 に従う、パケットエンコード規則の基本整列変形（basic aligned variant）を用いてエンコードされるべきである。

各々のメッセージは、共通フィールドとメッセージ固有の情報のフィールドから成る。共通フィールドは以下の通りである。

フィールド	説明
sequenceNumber	要求または更新メッセージは、ユニークなシーケンス番号を持つ。要求メッセージへの応答として送信されたメッセージ(確認または拒否メッセージ)は要求メッセージに含まれていたシーケンス番号を用いる。再送信されたメッセージは同じシーケンス番号を用いなければならない。
replyAddress	これは要求メッセージに対する応答を返す時のアドレスである。アドレスがトランスポートレイヤから得られる場合を除いて、全ての要求メッセージは、replyAddress を含まなければならない。IP ネットワークでは、要求メッセージの送信側がデフォルトポート(2099)を受信監視(listen)している場合には、replyAddress を含む必要はない。そのような場合には、受信側は、受信した要求パケットの IP ヘッダから送信側の IP アドレスが分かり、それにデフォルトポート(2099)を合わせて、送信側のトランスポートアドレスを得ることができる。 <sup>1</sup>
version	このメッセージの送信側で用いられているプロトコルのバージョン
annexGversion	JT-H225.0 付属資料 G のプロトコルバージョン(後方互換性のためにのみ表示される。「Annex G V2」を示すべき)。
hopCount	このメッセージが伝播する時に通過してもよい同位エレメントの個数を定義する。同位エレメントがこのメッセージを受信し、別の同位エレメントへ転送する時に、hopCount を 1 減ずる。hopCount が 1 以上であるならば、同位エレメントは新しいホップカウント値をメッセージ内にセットする。もし hopCount が 0 になった場合は、同位エレメントはメッセージを転送してはならない。この時、メッセージが要求であったならば、同位エレメントは適用可能な情報とともに確認メッセージで応答するべきである。情報が全く得られない場合は、同位エレメントは拒否メッセージで応答すべきである。
integrityCheckValue	より改善されたメッセージ完全性/メッセージ認証を提供する。暗号に基づく完全性チェック値が、交渉された完全性アルゴリズムと秘密鍵を用いてメッセージ全体について送信側で計算される。integrityCheckValue の計算に先立って、このフィールドの各バイトはゼロに設定されなければならない。計算後は、送信側は、計算された完全性チェック値を integrityCheckValue フィールドに入れてメッセージを送信する。
tokens	操作を許可するために要求されるかもしれないデータである。利用可能な場合は、データはメッセージに挿入されなければならない。
cryptoTokens	暗号化トークン。
nonStandard	非標準情報。
serviceID	この識別子は 2 つの同位エレメント間の特定のサービス関係セッションを

<sup>1</sup> 同位エレメントは、ネットワークアドレス変換 (NAT) 装置の後に隠されることはないとは仮定される。それゆえ、RAS メッセージの場合に、replyAddress 上のトランスポートアドレスを優先させることは要求されない。

示す。同位エレメントが、新しいサービス関係（それは、ServiceRequest メッセージの中に、サービス ID（serviceID）フィールドが無いことにより示される）の確立を要求する ServiceRequest メッセージを受信する時は常に、**グローバルにユニークなサービス ID を割り当て、ServiceRequest メッセージの送信側に、ServiceConfirmation メッセージで返さなくてはならない。**

一度、サービス関係が確立すると、サービス ID は、同位エレメントによる全ての後続のメッセージ（例えば、UsageIndication、DescriptorIDRequest、DescriptorRequest、AccessRequest）に含まれなければならない。これは、メッセージを受信した同位エレメントにより、メッセージの送信側との間にサービス関係があるか否かを確認するために使用される。

genericData

メッセージに関連した全ての一般的なデータを搬送する。GenericData は、JT-H323/JT-H225.0 で記述されている。

featureSet

一般的な機能セットのネゴシエーションのために使われる。ネゴシエーションスキームは、JT-H323/JT-H225.0 の RAS シグナリング用に記述されている。

### 6.1 記述子

記述子は、メッセージではなく、テンプレートの集合にラベルをつけるためのメッセージ要素である。記述子は以下の情報を含む。

フィールド	説明
descriptorInfo	記述子のためのユニークな識別子と最後に変更された時刻を持つ。（以下の記述子情報を参照のこと。）
templates	この記述子が解決できるアドレスを定義しているテンプレートの集合である。
gatekeeperID	記述子の所有者（すなわち、このメッセージを生成したゲートキーパー）を示すテキスト識別子である。

### 6.2 記述子情報

記述子情報は、記述子をユニークに指し示し、記述子が最後に変更された時刻を示す。

フィールド	説明
descriptorID	この記述子を、他にあり得る多数の記述子から識別するためのグローバルでユニークな識別子である。
lastChanged	この記述子が最後に変更された UTC 日時である。

### 6.3 アドレステンプレート

アドレステンプレートは、一つあるいはそれ以上の別名アドレスを示す。テンプレートはメッセージでないが、他の要素のビルディングブロックとして使われる要素である。テンプレートは、他の構造体で成り立ち、以下の節で説明される。

フィールド	説明
pattern	パターンのリストである。（以下の、パターン（Pattern）の節を参照）
routeInfo	このテンプレートのルーティング情報のリスト（以下の、ルーティング情報（Route Information）の節を参照）。
timeToLive	このテンプレートが有効である時間を秒単位で示す。
supportedProtocols	このテンプレートによりサポートされるプロトコルのタイプを示す（例：音声（voice）、FAX（fax））。
featureSet	この addressTemplate がサポートする一般的な機能セットと、それがリモートのエンドポイントにおいて必要とし、要望する一般的な機能を指定する。このレベルで指定された一般的な機能情報は、この AddressTemplate に対して適用可能な全ての routeInformation に適用する。汎用拡張フレームワークは、JT-H323/JT-H225.0 に記述されている。

### 6.3.1 パターン

パターン (*pattern*) 構造体は、アドレステンプレートに現れる。パターンは、単一の別名アドレス、ワイルドカード付き別名アドレス、または、ある範囲の値の別名アドレスで指定してよい。ワイルドカード指定のアドレスは、階層的に構造化されたアドレスを仮定している。一方、アドレスの範囲指定は、指示された構造体を表す、意味のあるアドレスタイプのみを仮定している。

フィールド	説明
specific	特定の別名アドレス
wildcard	文字列の可能な拡張を表現する階層的な定義。例えば、E.164 番号では、番号の最後、電子メールアドレスでは先頭で、拡張できる。例えば <i>wildcard</i> が、「+1 303」ならば、その <i>pattern</i> は Denver の地域コードの全ての数値を意味する。
range	アドレスの範囲。範囲の最初と最後を示す値を含む。

### 6.3.2 ルーチング情報

テンプレート (*routeInfo* フィールド) 内のルーティング情報構造体は、以下を含む。

フィールド	説明
messageType	このテンプレート内の特定のアドレスの解決を試みる際に送るメッセージのタイプを示す。 <i>SendAccessRequest</i> , <i>sendSetup</i> , <i>nonExistent</i> (アドレスは存在しないことを示す) のいずれかの可能性がある。
callSpecific	TRUE にセットされた場合、このルーティングに対するそれぞれの呼に対して認証が要求され、 <i>AccessRequest</i> メッセージが呼情報( <i>call information</i> )を含まなければならないことを意味する。 <i>MessageType</i> が <i>sendAccessRequest</i> の場合にのみ、このブール値は有意であり、さもなければ <i>callSpecific</i> は FALSE にセットされなければならない。
usageSpec	もし現れる場合、このルートに対する呼に関して送信しなければならない <i>UsageIndication</i> メッセージを指定する。
priceInfo	この特定のルーティングのための課金情報のリストである(以下の、課金情報の節を参照)。異なる課金構造をもつ複数のゲートウェイは、複数の <i>RouteInformation</i> 構造体に記述されることに注意すること。
contacts	<i>routeInfo</i> の <i>messageType</i> フィールドで指定したように、これは、メッセージを受け入れる要素の接続情報である。接続情報は、可能な接続のリストとして提供してもよい。(以下の、接続情報の記述を参照)
type	呼を受け入れることができるエンドポイントのタイプを示す。ゲートキーパー経由型の場合には、ゲートキーパー自身ではなく、ゲートキーパーによって接続されるエンドポイントのタイプを示す。
featureSet	このルートがサポートする一般的な機能セットと、それがリモートのエンドポイントで必要とし、要望する一般的な機能を指定する。このレベルで指定された一般的な機能情報は、この <i>RouteInformation</i> 要素に適用可能な全ての <i>ContactInformation</i> に適用する。一般的な拡張フレームワーク (Generic Extensible Framework) は、JT-H323/JT-H225.0 に記述されている。
circuitID	もし現れる場合、これは、特定の呼に適用する SCN 回線情報を持つ。このレベルで指定された回線情報は、この <i>RouteInformation</i> 要素に適用可能な全ての <i>ContactInformation</i> に適用する。
supportedCircuits	もし現れる場合、これは、宛先のパターンに対して、ドメインまたはゾーン内でサポートされる SCN 回線用の回線識別値を持つ。これは、同位エレメントに、リモートの同位エレメントに対して宛先の回線情報のサポートを公表することを許可する。このレベルで指定された回線情報は、この <i>RouteInformation</i> 要素に適用可能な全ての <i>ContactInformation</i> に適用する。

### 6.3.3 課金情報

課金情報は、ルーティング情報 (*Route Information*) 構造体 (*PriceInfo* フィールド) の要素として現れる。課金情報は、*PriceInfoSpec* と *PriceElement* 構造体を通して定義される。

フィールド	説明
currency	ISO-4217 の通貨表示。
currencyScale	左シフトさせる基数の位置を示す数値である。例えば、通貨 ( <i>currency</i> ) として USD が指定された場合、 <i>currencyScale</i> の 2 は <i>priceElement</i> が米国のセント

validFrom	(cent) で額が表示されていることを示す。
validUntil	この情報が有効となる UTC 日時の起源を示す。
hoursFrom	この情報が期限切れとなる UTC 日時を示す。
hoursUntil	このレートが開始される時刻を示す。
priceElement	このルーチングが終了する時刻を示す。hoursFrom より小さな値は 00:00 を跨いでいることを示す。
priceFormula	課金の合計を示す PriceElement のオプションのリストであり、これらの合計が課金情報となる。
	課金の式を含むオプションの文字列で、構造化された priceElement の代替として使われる。

PriceElement 構造体は、下記のフィールドを含む。

フィールド	説明
amount	計量の増分。quantum(特定量)ごと、あるいは quantum(特定量)の一部ごとに計量は増加する。
quantum	amount が適用する単位の数。例えば、秒単位 (units) の値 60 は、呼は 1 分ごと、または分の一部ごとに課金されることを示す。units フィールドが initial, minimum, maximum のいずれかの値にセットされた場合には、quantum フィールドは有効でなく、受信側では、その値は無視されなければならない。
units	Quantum を表現する単位 (unit) のタイプ。 <ul style="list-style-type: none"> <li>• Seconds - 呼の接続時間の秒数</li> <li>• packets - 送信または受信パケット数</li> <li>• bytes - 送信または受信バイト数</li> <li>• initial - 初期の接続料金</li> <li>• minimum - 呼の最低料金</li> <li>• maximum - 呼の最高料金</li> </ul>

#### 6.3.4 接続情報

接続情報構造体は、ルーチング情報 (Route Information) 構造体 (contact フィールド) の一要素である。

フィールド	説明
transportAddress	Route Information 構造体の messageType フィールドで指定されたメッセージをいずれに送るべきかを示す別名アドレス(例えばトランスポートアドレスまたは URL)。可能な場合は必ず、トランスポートアドレスを使わなければならない。
priority	複数の宛先がリストされた場合、priority フィールドは、その宛先が試みられるべき順番を指定する。例えば、接続が試みられるべき順序に優先順位がなければ、リスト中の接続は、優先順位 (priority) を共有できる。Priority が 0 は、一番高い priority (最初の選択) を示す。
transportQoS	この接続に関して行われた全てのリソースの予約への責任のある場所を示す。
security accessTokens	接続を伴う通信の際に使われる、優先順位をさ下げていくセキュリティ機構。この接続に対するメッセージ (呼設定または AccessRequest) の中で通過させなければならないトークンのセット。また、トークンは、後に続くこのテンプレートを使った呼に属する UsageIndication メッセージの中で送られなければならない。
multipleCalls	このフィールドは、RouteInformation 構造体の messageType フィールド中の値が sendSetup であった場合のみ意味を持つ。multipleCalls が TRUE の場合、これは、接続がただ一つの呼シグナリングコネクション上で、複数呼のシグナリングができることを示す。FALSE の場合は、この接続はこの能力を持たない。
featureSet	この ContactInformation 要素と関連するエンティティがサポートする一般的な機能セットと、それがリモートのエンドポイントで必要とし、要望する一般的な機能を指定する。一般的な拡張フレームワーク (Generic Extensible Framework) は、JT-H323/JT-H225.0 に記述されている。
circuitID	もし現れる場合、これは、特定の呼に適用する SCN 回線情報を持つ。このレベルで指定された回線情報は、この ContactInformation 要素に関連した接続に適用する。
supportedCircuits	もし現れる場合、これは、宛先のパターンに対して、ドメインまたはゾーン内でサポートされる SCN 回線用の回線識別値を持つ。これは、同位エレメント

に、リモートの同位エレメントに対して宛先の回線情報のサポートを公表することを許可する。このレベルで指定された回線情報は、この *ContactInfo* 要素に関連した接続に適用する。

#### 6.4 共通構造体

この節で定義される構造体は、多くのメッセージで出現する。

##### 6.4.1 AlternateBE

フィールド	説明
contactAddress	これは、代替同位エレメントのトランスポートアドレスである。 (このプロトコルのメッセージの送信先となるアドレス)
priority	複数の代替がリストされた場合、 <i>priority</i> フィールドは、その代替が試みられるべき順番を指定する。例えば、代替が試みられるべき順序に優先順位がなければ、リストの中の代替は優先順位 ( <i>priority</i> ) を共有できる。Priority が 0 は、一番高い <i>priority</i> (最初の選択) を示す。
elementIdentifier	この代替同位エレメントは、この Unicode 文字列を識別子として使う。

##### 6.4.2 PartyInformation

この構造体は、呼のパーティ (発信元もしくは宛先) に関する情報を持つ。

フィールド	説明
logicalAddress	パーティを識別する、E-mail または E.164 フォーマットのアドレス。
domainIdentifier	呼を発生または終端させる AD (管理ドメイン) を識別する別名アドレス。複数のドメインが一つの呼に含まれている場合には、送信者の見込みによる、呼の送信元または終端先として扱われるドメインを示すべきである。
transportAddress	エンドポイントのトランスポートアドレスである。
endpointType	エンドポイントのタイプと能力についての詳細を示す。
userInfo	呼の背後にいるユーザに関する情報。以下の <i>UserInformation</i> の節を参照。
timeZone	課金目的のための、パーティのタイムゾーン。発信元のパーティがゲートウェイの場合、そのゲートウェイのタイムゾーンが伝送されなければならない。UTC (世界標準時) との差を秒で記述する。

##### 6.4.3 CallInformation

個々の呼を識別するための情報。

フィールド	説明
callIdentifier	これは、呼のユニークな識別子を提供する。これは、RAS 及び呼シグナリングメッセージ中で同一の呼と関連付けられた <i>callIdentifier</i> でなければならない。
conferenceID	これは、呼の属する会議についてのユニークな識別子を提供する。これは、RAS 及び呼シグナリングメッセージ中で同一の会議と関連付けられた <i>conferenceID</i> でなければならない。
circuitID	もし現れる場合、これは、呼に適用する SCN 交換網情報を持つ。

##### 6.4.4 UserInformation

呼のパーティによって表現されたユーザを識別する情報。

フィールド	説明
userIdentifier	ユーザをユニークに識別する別名アドレス。
userAuthenticator	セキュアな認証のための暗号化されたトークン。

##### 6.4.5 Usage Specification

この要素は、UsageIndication メッセージで報告されるべき要求されたパラメータを記述する。ここでの記述が適用される呼は、UsageSpecification 要素を含むメッセージの前後関係から決定される。

フィールド	説明
sendTo	UsageIndication メッセージの送り先の同位エレメント。

when	<p>送信元が、その同位エレメントとサービス関係を持つ場合、これは ServiceConfirmation メッセージで返される要素識別子である。</p> <p>指示が送られるべき、呼の段階と頻度を指定する。</p> <ul style="list-style-type: none"> <li>・ Never - メッセージの送信停止。</li> <li>・ Start - 呼の開始時。</li> <li>・ End - 呼の終了時か、その後。</li> <li>・ Period- - 呼の生存中、定期的に（送出）。時間は秒単位で計測される。</li> <li>・ Failure - 呼設定の試みが失敗した時に報告。</li> </ul>
required	UsageIndication メッセージ中のフィールドになければならない ( <i>must</i> ) 識別子のリスト。Usage information の送信元は、これらのフィールドを提供できない時は、このメッセージを含むメッセージを拒否もしくは無視しなくてはならない。
preferred	UsageIndication メッセージ中のフィールドにあるべき ( <i>should</i> ) 識別子のリスト。
sendToBEAddress	<p>（アドレスが）解決された場合、これは、UsageIndication メッセージが送信されなくてはならない同位エレメントのアドレスを指定する妥当なアドレスである。このフィールドによるアドレス解決が、2つ以上のアドレスであった場合（例えば、DNS クエリーが、アドレスのリストを返信した場合）には、同位エレメントは UsageIndication メッセージをリスト中の同位エレメントのただ一つだけに対して送信しなくてはならない。</p> <p>同位エレメントが一つのアドレスへの送信に失敗した場合には、同位エレメントはリストから別のアドレスを選択し、その新しいアドレスに対して UsageIndication メッセージを送信することを試みてよい。同位エレメントは、UsageIndicationConfirmation、UsageIndicationRejection を受信するか、または試みるアドレスがなくなるまでは、リスト中の追加アドレスそれぞれに対して試みることを続けてよい。</p> <p>「sendToBEAddress」フィールドは、UsageSpecification 中の「sendTo」フィールドとは異なることに注意すること。「sendTo」フィールドは識別子である。「sendTo」フィールドは、特定の境同位エレメントの識別子（例えば、「border_element1」）となり得るし、同位エレメントのセットを論理的に表現する識別子（例えば、「border elements of my company」）にもなり得る。</p> <p>「sendToBEAddress」フィールドは、一つまたはそれ以上のアドレスを解決するものである。</p>

#### 6.4.6 Security Mode

この要素は、ピア・ツー・ピア通信で使われる特定のセキュリティプロファイルについて記述する。

フィールド	説明
authentication	これは、使用される認証機構を示す。この認証機構は ServiceRequest メッセージで提供されたセットの中から選ばなければならない。
integrity	これは、使用される完全性機構を示す。もし現れた場合、後続の全てのメッセージは integrityCheckValue フィールドを持たなくてはならない。この場合、AuthenticationMode は秘密鍵がどのように生成されるかを記述する。(DH <sup>(*)</sup> 鍵交換、あるいは a-priori (あらかじめ鍵を共有する) など)
algorithmOID	これは、セキュリティ機構のための暗号化アルゴリズムを示す。

(TTC 注)

DH : Diffie-Hellman

公開鍵暗号方式を利用した共通鍵共有方式。IPSEC では IKE(Internet Key Exchange:RFC2409)のデフォルトの鍵交換アルゴリズムとして採用されている。

(TTC 注おわり)

## 6.5 サービス関係

### 6.5.1 Service Request

同位エレメントは、サービス関係を確立するために、ServiceRequest メッセージを他の同位エレメントに送信してもよい。この関係により、同位エレメントの間で使用されるセキュリティ機構が指定され、代替もしくはバックアップの同位エレメントの識別が可能となる。この関係は一方向の関係であることに注意。2つの同位エレメントの間で交渉されるセキュリティは、ServiceRequest の送信側の同位エレメントにより送信される要求、および ServiceRequest の受信側により送信される応答に使用される。セッション鍵は、サービス関係の確立の過程で生成されてもよい。これらの鍵は、サービス関係の継続時間の間は有効である。H.235 に定義されているように、この目的のためにトークンを使用してもよい。

ServiceRequest の受信側は、ServiceRequest の送信側がバックアップサービスを要求することができる代替の同位エレメントを指定してもよい。サービス関係の確立は、同位エレメントのポリシーがそのような関係を要求するかもしれないが、オプションの手続きである

同位エレメントは、既に関係をもっている同位エレメントに対して、その関係の期間を終了させ、新しい期間を開始するために、ServiceRequest メッセージを送信してもよい。サービス関係は有効期間を設けてもよい。同位エレメントは、新たな ServiceRequest を送信することにより、関係をリフレッシュしてもよい。

フィールド	説明
elementIdentifier	要求送信元の同位エレメントを識別する文字列。
domainIdentifier	サービス関係の要求元の管理ドメイン
securityCapability	この同位エレメントがサポート可能なセキュリティ機構のセット
timeToLive	提案された、サービス関係の有効期間（単位：秒）。 もし存在しなければ、有効期間は無限であるものと仮定される。
usageSpec	これは、発信側同位エレメントと受信側同位エレメント間の全ての呼に関して、発信側同位エレメントが、受信側同位エレメントに送信を要求する利用状況情報を指定する。

### 6.5.2 Service Confirmation

ServiceRequest メッセージを受信した同位エレメントは、サービス関係の確立に同意することを示すために、ServiceConfirmation メッセージで応答する。全ての新たなサービス関係が、サービス識別子によって識別される。同位エレメントは、サービス ID の無い ServiceRequest メッセージを受信する度に、同位エレメントはあるユニークなサービス ID を割り当て、それを ServiceConfirmation メッセージで ServiceRequest メッセージ送信側へ返す。もしその同位エレメントが、ServiceRequest メッセージの送信元の同位エレメントと既にサービス関係を持っていた場合には、ServiceConfirmation の送信は最初の関係の期間が終了し、新しい期間が開始されることを示す。ServiceConfirmation メッセージは、ServiceRequest メッセージにより送られたサービス ID と同じ値を含まなくてはならない。承認しないサービス ID を含む ServiceRequest メッセージを受信する同位エレメントは、ServiceRejection メッセージを返さなければならない。

フィールド	説明
elementIdentifier	これは、同位エレメントを識別するための文字列である。
domainIdentifier	要求に対して応答する管理ドメイン。
alternates	これは、この同位エレメントが応答に失敗した場合にアクセス可能な代替の同位エレメントのリストである。
securityMode	これは、このサービス関係において使用すべきセキュリティ機構を示す。セキュリティ機構は ServiceRequest メッセージで提供されたセットから選択されなければならない。
timeToLive	サービス提供側の同位エレメントにより決定されるサービス関係の有効期間（単位：秒）
usageSpec	これは、発信側同位エレメントと受信側同位エレメント間の全ての呼に関して、受信側同位エレメントが、サポート可能な利用状況情報を指定する。

### 6.5.3 Service Rejection

ServiceRequest メッセージを受信した同位エレメントは、サービス関係の確立を拒否することを示すために、ServiceRejection メッセージで応答する。もしその同位エレメントが、ServiceRequest メッセージの送信元の同位エレメントと既にサービス関係を持っている場合には、ServiceRejection の送信は提案された新しい期間の開始が拒否され、元の関係の期間が継続されることを示す。



フィールド	説明
reason	<p>これは、同位エレメントが ServiceRequest を拒否した理由である。選択肢は以下の通り：</p> <ul style="list-style-type: none"> <li>• serviceUnavailable – この同位エレメントは現在サービス提供不可。</li> <li>• serviceRedirected – 試行されるべき代替の同位エレメントのリスト同位エレメント。</li> <li>• security – この同位エレメントは、ServiceRequest メッセージで提案されている全てのセキュリティ機構のサポートが不可能である。</li> <li>• continue – 多段階の鍵交換処理を継続するために、引き続き ServiceRequest メッセージを送信すべきであることを示す。</li> <li>• undefined – ServiceRequest を拒否する理由がどの選択肢とも一致しないことを示す。</li> <li>• unknownServiceID – ServiceRequest メッセージに含まれる serviceID フィールドが、同位エレメントに承認されていない。</li> <li>• cannotSupportUsageSpec – 同位エレメントが、提案された UsageSpecification を承認できない。</li> <li>• neededFeature – 要求したエンティティが、1 つまたはそれ以上の必要な一般的機能をサポートしていないという理由で、要求が失敗したことを示す。必要とされた機能は、応答の MessageCommonInfo フィールド中の neededFeatures フィールドの中で示される。</li> <li>• genericDataReason – 本当の理由 (reason) コードが、メッセージと共に送られた genericData フィールドに含まれていることを示す。genericData は、2 つ以上の理由 (reason) コードを含んでいるかもしれない点に注意。</li> <li>• usageUnavailable – 同位エレメントは、利用状況報告をサポートしていない。</li> <li>• unknownUsageSendTo – 提案された UsageSpecification で指定された sendTo または sendToPEAddress が、解決できない。</li> </ul>
alternates	<p>これは、ServiceRequest に応えることのできる可能性がある代替の同位エレメントのリストである。もし、reason が ServiceRedirected である場合には、少なくとも 1 つの代替 (の同位エレメント) が提供されるべきである。</p>

#### 6.5.4 Service Release

サービス関係中の同位エレメントのいずれかが、ServiceRelease メッセージを送信することにより関係を終了させてもよい。

フィールド	説明
reason	<p>これは、同位エレメントがサービス関係を終了させた理由である。選択肢は以下の通り：</p> <ul style="list-style-type: none"> <li>• outOfService – 同位エレメントがサービス提供を終了しつつある。</li> <li>• maintenance – 同位エレメントがメンテナンスのためにサービス停止中。</li> <li>• terminated – 同位エレメントが関係の終了を決定。</li> <li>• expired – サービス関係の有効期間が終了。</li> </ul>
alternates	<p>これは、サービス関係を確立できる可能性のある代替の同位エレメントのリストである</p>

#### 6.6 記述子の配布

### 6.6.1 Descriptor Request

DescriptorRequest メッセージは、エンティティに特定の記述子に対する同位エレメントを問い合わせることを許可する。

フィールド	説明
descriptorID	このメッセージの送信者により要求された 1 つ又はそれ以上の特定の記述子を識別する。

### 6.6.2 Descriptor Confirmation

DescriptorConfirmation メッセージは、同位エレメントが要求を解釈でき、実装において情報交換を許可している場合の、DescriptorRequest への同位エレメントの肯定応答である。

フィールド	説明
descriptors	上記の記述子である。

### 6.6.3 Descriptor Rejection

同位エレメントは、様々な理由により、記述子の要求を拒否することが可能である。

フィールド	説明
reason	<p>これは DescriptorRequest が拒否された理由である。選択肢は以下の通り:</p> <ul style="list-style-type: none"><li>• packetSizeExceeded - 応答は、最大パケットサイズを超過するため、要求側は他の伝送方法を用いて送信すべきである (例えば UDPの代わりにTCPを使用)。</li><li>• illegalID - DescriptorRequestの受信側には、要求された記述子の記録がない。</li><li>• security - DescriptorRequestは、受信側のセキュリティ要求に合わなかった。</li><li>• hopCountExceeded - ホップカウントがゼロになり、情報が利用できない。</li><li>• unavailable - 受信側が記述子を提供できない。静的な情報又は本プロトコルの枠外での情報獲得手段 (out-of-band provisioning)が使用されるべきである。</li><li>• noServiceRelationship - 受信側は、サービス関係の確立後にのみ、この情報を交換する。</li><li>• undefined - サービス要求 (DescriptorRequest) を拒否する理由がどの選択肢とも一致しないことを示す。</li><li>• neededFeature - リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答のMessageCommonInfo フィールド中のneededFeatureフィールドで通知される。</li><li>• ggenericDataReason - 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は1つ以上の理由コードを含むことが可能であることに注意すること。</li><li>• unknownServiceID - DescriptorRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li></ul>
descriptorID	この応答に対する特定の記述子を識別する。

### 6.6.4 Descriptor ID Request

DescriptorIDRequest を用いて、エンティティは同位エレメントに対し、その同位エレメントが所属する管理ドメイン内の記述子識別子のリストを問い合わせることができる。

### 6.6.5 Descriptor ID Confirmation

DescriptorIDConfirmation メッセージは、DescriptorIDRequest メッセージへの同位エレメントの肯定応答である。DescriptorIDConfirmation メッセージの受信における同位エレメントは、記述子の送信要求へのDescriptorRequest メッセージを送信してよい。

フィールド	説明
descriptorInfo	記述子情報のリストであり、ここでリスト内における各々の登録は、記述子及びそれが最後に変更された時刻をユニークに識別する。

### 6.6.6 Descriptor ID Rejection

同位エレメントは、様々な理由により DescriptorIDRequest を拒否することが可能である。

フィールド	説明
reason	<p>これは、要求拒否に対する理由を示す。選択肢は以下の通り:</p> <ul style="list-style-type: none"><li>• noDescriptors – 同位エレメントに、報告する記述子が存在しないことを示す。</li><li>• security – DescriptorIDRequestは、受信側のセキュリティ要求に合わなかった。</li><li>• hopCountExceeded – ホップカウントがゼロになり、情報が利用できない。</li><li>• unavailable - 受信側が記述子を提供できない。静的な情報又は本プロトコルの枠外での情報獲得手段(out-of-band provisioning)が使用されるべきである。</li><li>• NoServiceRelationship - 受信側は、サービス関係の確立後にのみ、この情報を交換する。</li><li>• undefined – サービス要求 (DescriptorIDRequest) を拒否する理由がどの選択肢とも一致しないことを示す。</li><li>• neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答のMessageCommonInfo フィールド中のneededFeatureフィールドで通知される。</li><li>• genericDataReason - 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は1つ以上の理由コードを含むことが可能であることに注意すること。</li><li>• unknownServiceID – DescriptorIDRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li></ul>

### 6.6.7 Descriptor Update

DescriptorUpdate メッセージは、アドレス情報が変化した場合の同位エレメントの通知である。同位エレメントはまた、初期化の間に、DescriptorUpdate メッセージを送信してよい。DescriptorUpdate の受信における同位エレメントは、DescriptorUpdate 内で識別される要素からの情報を要求してよい。

フィールド	説明
sender	DescriptorUpdate の受信における要素は、このアドレスへの要求を送信してよい(例えばトランスポートアドレス又はURL)。
updateInfo	更新のリストである。リスト内の各々の登録は、更新された記述子又は記述子の識別子のどちらかを提供する。リスト内の各々の登録はまた、記述子が増えられたのか、追加されたのか、削除されたのかどうかを示す。

### 6.6.8 Descriptor Update Acknowledgement

同位エレメントは、DescriptorUpdateAck メッセージの送信により、DescriptorUpdate メッセージの受信の応答をすべきである。応答内で使用されるシーケンス番号は、DescriptorUpdate メッセージにて受信され

るシーケンス番号と同一であるべきである。同位エレメントは、マルチキャストで到着する DescriptorUpdate メッセージに応答すべきでない。

## 6.7 アクセスのための解決

### 6.7.1 Access Request

同位エレメントは、特定の別名アドレスの解決を要求するために、もう1つの同位エレメントに対して AccessRequest メッセージを送信することが可能である。

フィールド	説明
destinationInfo	解決されるべきアドレスである。
sourceInfo	アクセスが要求される呼の発呼中のパーティに関する情報である。
callInfo	アクセスの認証が要求される特定の呼の識別を提供する。もし提供されない場合、要求は指定された宛先への、不特定の呼に対するものとなる。
usageSpec	発呼中のパーティが応答パーティに対し、このメッセージで要求される、関係する呼の送信を要求する、使用法メッセージを示す。callInfo が存在する場合だけ適用する。
desiredProtocols	エンドポイントがその呼に対して望んでいるプロトコル種別を優先順に識別する（例えば音声やファックス）。アドレス解決側は優先順位を考慮して、プロトコルをサポートしているエンドポイントの位置を探すために、このフィールドを使用することが可能である。着呼側はこの構造中の supportedPrefixes フィールドを無視するべきである。

### 6.7.2 Access Confirmation

同位エレメントは、AccessRequest メッセージで要求された情報を AccessConfirmation メッセージで返す。

フィールド	説明
templates	AccessRequest の特性と条件の合うテンプレートのリストである。
partialResponse	TRUE であるならば、このメッセージは有効な情報の一部を含んでいる。完全な情報はパケットサイズを超過しているため、送信されなかった。完全な情報は他のトランスポートタイプを使用して復旧するべきである。（例えば TCP）
supportedProtocols	サポートされているプロトコル種別を示す。（例えば音声、ファックス）
serviceControl	JT-H225.0 に定義されている様に使用。例えば JT-H323 の付属資料 K に記述されている通り。

### 6.7.3 Access Rejection

同位エレメントは、様々な理由で AccessRequest を拒否できる。

フィールド	説明
reason	要求を拒否する理由である。選択肢は以下の通り： <ul style="list-style-type: none"><li>noMatch – AccessRequest で指定された宛先は解決されない。</li><li>packetSizeExceeded - 返信が最大のパケットサイズを超過するため、要求側は他の伝送方法を用いて要求を送信するべきである。（例えば UDP の代わりに TCP を使用）</li><li>security – AccessRequest は受信者のセキュリティ要求に合わなかった。</li><li>hopCountExceeded – ホップカウントがゼロになり、情報が利用できなかった。</li><li>noServiceRelationship – 受信側はサービス関係の確立後にのみ、この情報を交換する。</li><li>needCallInformation – 明確な呼情報は要求の中に出てこなかった。</li></ul>

- undefined – AccessRequest を拒否する理由が、どの選択肢とも一致しない。
- neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。
- genericDataReason - 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は 1 つ以上の理由コードを含むことが可能であることに注意すること。
- destinationUnavailable - 宛先は解決されたが利用不可である。
- aliasesInconsistent - 複数の別名が複数の宛先と結びついている。
- resourceUnavailable - 一つ以上の要求されたリソースが使用不可である。
- incompleteAddress - 宛先を明確に識別できない。
- unknownServiceID – AccessRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。
- usageUnavailable – 同位エレメントが利用状況報告をサポートしていない。
- cannotSupportUsageSpec – 同位エレメントが提案されている UsageSpecification に従うことができない。
- unknownUsageSendTo – 提案されている UsageSpecification 中に規定されている sendTo や sendToPEAddress が解決できない。

serviceControl

JT-H225.0 に定義されている様に使用。例えば JT-H323 の付属資料 K に記述されている通り。

## 6.8 Request In Progress

同位エレメントは、同位エレメントによって要求された要求応答時間が、標準的に期待される応答間隔を超過することを示す RequestInProgress メッセージを返信してもよい。要求に対して送信しようとしているメッセージのシーケンス番号は、要求の中にあるシーケンス番号と同一でなければならない。

フィールド

説明

delay

同位エレメントが最初の要求に対して応答するのに要する期待される時間長。msec で表現される。

serviceControl

JT-H225.0 に定義されている様に使用。例えば JT-H323 の付属資料 K に記述されている通り。

## 6.9 非標準メッセージ

### 6.9.1 Non-Standard Request

Non-Standard Request は、付属資料 G に定義されていない要求メッセージを表すよう、同位エレメントから送信されてもよい。非標準の情報は MessageCommonInfo の nonStandard 要素で伝えられる。

### 6.9.2 Non-Standard Confirmation

NonStandardConfirmation は、NonStandardRequest メッセージへの応答として、同位エレメントから送信されてもよい。非標準の情報は MessageCommonInfo の nonStandard 要素で伝えられる。

### 6.9.3 Non-Standard Rejection

NonStandardRejection は、NonStandardRequest メッセージへの応答として、同位エレメントから送信されてもよい。非標準の情報は MessageCommonInfo の nonStandard 要素で伝えられる。

フィールド reason	説明 要求を拒否する理由である。選択肢は以下の通り：
	<ul style="list-style-type: none"> <li>• notSupported – 受信側は、これは NonStandardRequest であると理解できるが、非標準のデータは理解またはサポートしない。</li> <li>• noServiceRelationship – 受信側はサービス関係の確立後にのみ、この情報を交換する。</li> <li>• undefined – NonStandardRequest を拒否する理由が、どの選択肢とも一致しない。</li> <li>• neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。</li> <li>• genericDataReason - 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は 1 つ以上の理由コードを含むことが可能であることに注意すること。</li> <li>• unknownServiceID – NonStandardRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li> </ul>

#### 6.10 Unknown Message Response

理解できないメッセージを受信した同位エレメントは、UnknownMessageResponse メッセージで送信側に応答するべきである。いくつか他の付属資料 G のメッセージが適切な応答を供給する場合は、同位エレメントはこのメッセージを使用してはならない。（例えば DescriptorRequest が不正な記述子の識別子を持つ場合には、DescriptorRejection が適切な応答である。）

フィールド unknownMessage reason	説明 不明メッセージの内容である。 UnknownMessageResponse が使用される理由である。選択肢は以下の通り：
	<ul style="list-style-type: none"> <li>• notUnderstood – メッセージが理解できない。</li> <li>• undefined – UnknownMessageResponse を送信する理由が、どの選択肢とも一致しない。</li> </ul>

#### 6.11 利用状況報告

##### 6.11.1 Usage Request

受信側に、特定の呼に関係している UsageIndication メッセージを送るように要求する。

フィールド CallInfo UsageSpec	説明 UsageIndication を送るべき呼 UsageIndication がいつ到着するべきであるか、そして UsageIndication が何を 含んでいるべきであるか明記する。
--------------------------------	---

##### 6.11.2 Usage Confirmation

UsageConfirmation メッセージは、受信側が要求を受理したこと、そして UsageIndication を後で送信することを示すために UsageRequest メッセージの応答として送られる。

##### 6.11.3 UsageRejection

UsageRejection メッセージは、受信側が要求を拒否したこと、そして UsageIndication を後で送信ないことを示すために UsageRequest メッセージの応答として送られる。

フィールド	説明
reason	<p>これは、同位エレメントが UsageRequest メッセージ を拒否した理由である。選択肢は以下の通り：</p> <ul style="list-style-type: none"> <li>• invalidCall – その UsageRequest により規定された呼が、承認されていない呼である。</li> <li>• security – UsageRequest が、受信側のセキュリティ要求に合わなかった。</li> <li>• unavailable – 受信側が、要求された呼についての usages 情報を持っていない。</li> <li>• noServiceRelationship – 受信側は、サービス関係の確立後にのみ、この情報を交換する。</li> <li>• undefined – UsageRequest を拒否する理由がどの選択肢とも一致しないことを示す。</li> <li>• neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。</li> <li>• genericDataReason – 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は 1 つ以上の理由コードを含むことが可能であることを注意すること。</li> <li>• unknownServiceID – UsageRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li> </ul>

#### 6.11.4 Usage Indication

呼の詳細情報と利用情報を伝える。このメッセージは、呼に関連している同位エレメントによって受信された最後の UsageSpecification 要素に関して送られる。

フィールド	説明
callInfo	UsageIndication が適応されるべき呼。
accessTokens	呼のためのアクセストークン。呼で使われているアドレスプレートで受信されて、同じ呼の中の AccessRequest /呼設定メッセージでも使われる。
senderRole	UsageIndication の送信側の役割： <ul style="list-style-type: none"> <li>• <b>originator</b> – 発信元パーティ</li> <li>• <b>destination</b> – 着信先パーティ</li> <li>• <b>nonStandard</b> – その他</li> </ul>
usageCallStatus	現在の呼の状態： <ul style="list-style-type: none"> <li>• <b>preConnect</b></li> <li>• <b>callInProgress</b></li> <li>• <b>callEnded</b></li> <li>• <b>preConnect</b></li> </ul>
srcInfo	発呼側パーティの E.164 アドレスまたは電子メールアドレス。E.164 アドレスの場合、ANI/CLI（発信電話番号通知/発信番号識別）を指定する。
destAddress	着呼側パーティの E.164 アドレスまたは電子メールアドレス。
startTime	呼が開始された時の UTC フォーマットによる時刻。呼設定の段階を通過した呼のためだけに関連している。呼で使用される複数のメディアタイプのために、おのおののメディアタイプは、メディアストリームが開始された時刻と一致している異なった startTime を報告すべきである。周期的なメッセージであるため、startTime は、前のメッセージの endTime と一致しているべきである。
endTime	呼が終了した時の UTC フォーマットによる時刻。終了した呼のためだけに

関連している。呼で使用される複数のメディアタイプのために、おのこのメディアタイプは、メディアストリームが終了した時刻と一致している異なった endTime を報告すべきである。周期的なメッセージであるため、endTime は、報告期間が終了する時刻である。

terminationCause  
usageFields

呼が終了した理由。終了した呼のためだけに関連している。  
情報フィールドの集合。それぞれのフィールドが標準あるいは非標準になり得る UsageField によって表される。標準 UsageFields は将来の課題である。

#### 6.11.5 Usage Indication Confirmation

UsageIndicationConfirmation メッセージは、UsageIndication メッセージに応答して送られ、受信側が、その UsageIndication を受理したことを示している。

#### 6.11.6 Usage Indication Rejection

UsageIndicationRejection メッセージは、 UsageIndication メッセージに応答して送られ、受信側が、 UsageIndication を拒否したこと、そして UsageIndication を無視するであろうことを示している。

フィールド  
reason

説明  
これは、同位エレメントが UsageIndication メッセージを拒否した理由である。選択肢は以下の通り：

- unknownCall – UsageIndication で明記された呼が、承認された呼ではない。
- incomplete – UsageIndication が、この UsageIndication に適用した。
- noServiceRelationship –受信側は、サービス関係の確立後にのみ、この情報を交換する。
- undefined – UsageIndication を拒否する理由がどの選択肢とも一致しないことを示す。
- neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。
- genericDataReason - 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は1つ以上の理由コードを含むことが可能であることを注意すること。
- unknownServiceID – UsageIndication メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。

### 6.12 検証

#### 6.12.1 Validation Request

呼の着呼側の同位エレメントは、その呼の発呼側を検証するために、別の同位エレメントに ValidationRequest メッセージを送ることができる。

フィールド	説明
accessTokens	呼のためにアクセス権限を証明する発呼側から受信したトークン
destinationInfo	呼の着呼側の詳細
sourceInfo	発呼したエンドポイントのタイプ情報
callInfo	アクセス権限が要求される特定の呼の識別を提供する。
usageSpec	もし存在しているなら、検証された呼に関して、 UsageIndication が送られることをリクエストするメッセージを送っている同位エレメントを示す。



## 6.12.2 Validation Confirmation

呼が検証されている事を示す。リクエストしている同位エレメントは、呼を終端してもよい。検証を行う同位エレメントは呼を終端するために別名をしめす。

フィールド	説明
destinationInfo	受信側同位エレメントで使われるための、相手先用代替パラメータ。
usageSpec	これが存在するなら、検証された呼に関する usageIndication が送られる事を、確認要求を送っている同位エレメントに示す。

## 6.12.3 Validation Rejection

呼が検証されていない事を示す。リクエストしている同位エレメントは、呼を確立しないかもしれない。

フィールド	説明
reason	<p>これは、リクエストを拒否した理由である。選択肢は以下の通り：</p> <ul style="list-style-type: none"><li>• tokenNotValid – 与えられたアクセストークンがその呼に対して検証されない。</li><li>• security – validationRequest は、受信側のセキュリティ要求に合わなかった。</li><li>• hopCountExceeded – ホップカウントがゼロになり、情報が利用出来ない。</li><li>• missingDestInfo – 与えられた着信側情報が呼を検証するのに十分でなかった。</li><li>• noServiceRelationship – 受信側はサービス関連の設定後のみ、この情報を交換する。</li><li>• undefined – validationRequest を拒否する理由がどの選択肢とも一致しないことを示す。</li><li>• neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。</li><li>• genericDataReason – 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は 1 つ以上の理由コードを含むことが可能であることに注意すること。</li><li>• unknownServiceID – UsageIndication メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li></ul>

## 6.13 認証

### 6.13.1 Authentication Request

同位エレメントはユーザ認証のために異なる同位エレメントに対して AuthenticationRequest メッセージを送信することができる。

フィールド	説明
applicationMessage	認証のためのアプリケーションプロトコルメッセージを含む。

### 6.13.2 Authentication Confirmation

成功した認証を示す。このメッセージに対して定義されたフィールド情報は存在しない。関連する情報(トークンや暗号化トークン)はメッセージの共通部分に含まれている。

### 6.13.3 Authentication Rejection

認証が失敗したことを示す。

フィールド	説明
reason	<p>リクエストが拒否された理由である。下記に理由一覧を示す。</p> <ul style="list-style-type: none"><li>• security – AuthenticationRequest が受信者のセキュリティの必要条件に合致しなかった。</li><li>• hopCountExceeded – hopCount が 0 に到達し、情報が無効になった。</li><li>• noServiceRelationship – 受信側はサービス関係を確立した後のみ、この情報を交換する</li><li>• undefined – AuthenticationRequest が拒否された理由はこの理由一覧の範囲外である。</li><li>• neededFeature – リクエスト側が一つ以上の必要な一般的な機能をサポートしていなかったため、リクエストが失敗したことを示す。必要とされる機能は、応答の MessageCommonInfo フィールド中の neededFeature フィールドで通知される。</li><li>• genericDataReason – 実際の理由コードが、このメッセージと共に送信される genericData フィールド内に含まれていることを示す。genericData は 1 つ以上の理由コードを含むことが可能であることに注意すること。</li><li>• unknownServiceID – AuthenticationRequest メッセージに含まれる serviceID フィールドが同位エレメントにより認識されていない。</li><li>• securityWrongSyncTime – 送信側がセキュリティ的に不適当なタイムスタンプを受け入れられなかった。これはタイムサーバの問題、同期はずれもしくは大きなネットワーク遅延が原因と考えられる。</li><li>• securityReplay – 繰り返しの攻撃が生じた。これは割り当てられたタイムスタンプに対して複数回にわたり同一のシーケンス番号が生じた場合である。</li><li>• securityWrongGeneralID – メッセージ中の generalID が一致しないことを示す。これは誤ったアドレッシングにより生じることがある。</li><li>• securityWrongSendersID – メッセージ中の送信側 ID が一致しないことを示す。これは誤ったユーザのエントリにより生じることがある。</li><li>• securityMessageIntegrityFailed – 完全性や署名チェックが失敗した。これは初期リクエストにおけるパスワードの間違いやタイプミス、間違った秘密鍵 / 公開鍵の使用や頻繁な攻撃により生じることがある。</li><li>• securityWrongOID – トークンの OID (暗号化の有無に関わらず) もしくは暗号化 algorithm OID に不一致があったことを示す。異なったセキュリティアルゴリズム / プロファイルが使用されている。</li></ul>

付属资料 A : メッセージシンタックス ( Annex A: Message Syntax )

< do not translate or modify this section >

H501-MESSAGES DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS

AuthenticationMechanism,  
TimeStamp,  
ClearToken  
FROM H235-SECURITY-MESSAGES

AliasAddress,  
TransportAddress,  
ReleaseCompleteReason,  
ConferenceIdentifier,  
CallIdentifier,  
CryptoH323Token,  
CryptoToken,  
EndpointType,  
GatekeeperIdentifier,  
GloballyUniqueID,  
NonStandardParameter,  
NumberDigits,  
PartyNumber,  
SupportedProtocols,  
TransportQOS,  
VendorIdentifier,  
IntegrityMechanism,  
ICV,  
FeatureSet,  
GenericData,  
ServiceControlSession,  
CircuitInfo,  
CircuitIdentifier  
FROM H323-MESSAGES;

Message ::= SEQUENCE

```
{  
    body          MessageBody,  
    commonMessageCommonInfo,  
    ...  
}
```

MessageBody ::= CHOICE

```
{  
    serviceRequest          ServiceRequest,  
    serviceConfirmation    ServiceConfirmation,  
    serviceRejection       ServiceRejection,  
    serviceRelease         ServiceRelease,  
    descriptorRequest      DescriptorRequest,  
    descriptorConfirmation DescriptorConfirmation,  
    descriptorRejection    DescriptorRejection,  
    descriptorIDRequest    DescriptorIDRequest,  
    descriptorIDConfirmation DescriptorIDConfirmation,  
    descriptorIDRejection  DescriptorIDRejection,  
    descriptorUpdate       DescriptorUpdate,  
    descriptorUpdateAck    DescriptorUpdateAck,  
    accessRequest          AccessRequest,  
    accessConfirmation     AccessConfirmation,  
}
```

```

accessRejection      AccessRejection,
requestInProgress    RequestInProgress,
nonStandardRequest   NonStandardRequest,
nonStandardConfirmation NonStandardConfirmation,
nonStandardRejection NonStandardRejection,
unknownMessageResponse UnknownMessageResponse,
usageRequest          UsageRequest,
usageConfirmation     UsageConfirmation,
usageIndication       UsageIndication,
usageIndicationConfirmation UsageIndicationConfirmation,
usageIndicationRejection UsageIndicationRejection,
usageRejection        UsageRejection,
validationRequest     ValidationRequest,
validationConfirmation ValidationConfirmation,
validationRejection   ValidationRejection,
...,
authenticationRequest AuthenticationRequest,
authenticationConfirmation AuthenticationConfirmation,
authenticationRejection AuthenticationRejection
}

```

MessageCommonInfo ::= SEQUENCE

```

{
sequenceNumber      INTEGER (0..65535),
annexGversion        ProtocolVersion, -- set to "H.225.0 Annex G V2"
hopCount             INTEGER (1..255),
replyAddress         SEQUENCE OF TransportAddress OPTIONAL,
                    -- Must be present in request
integrityCheckValue ICV OPTIONAL,
tokens               SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens         SEQUENCE OF CryptoH323Token OPTIONAL,
nonStandard          SEQUENCE OF NonStandardParameter OPTIONAL,
...,
serviceID            ServiceID OPTIONAL,
genericData          SEQUENCE OF GenericData OPTIONAL,
featureSet           FeatureSet OPTIONAL,
version              ProtocolVersion -- current H.501 protocol version
}

```

ServiceID ::= GloballyUniqueID

```

--
-- H.501 messages
--

```

ServiceRequest ::= SEQUENCE

```

{
elementIdentifier ElementIdentifier OPTIONAL,
domainIdentifier  AliasAddress OPTIONAL,
securityMode      SEQUENCE OF SecurityMode OPTIONAL,
timeToLive        INTEGER (1..4294967295) OPTIONAL,
...,
usageSpec         UsageSpecification OPTIONAL
}

```

SecurityMode ::= SEQUENCE

```

{
authentication  AuthenticationMechanism OPTIONAL,
integrity        IntegrityMechanism OPTIONAL,
algorithmOIDs   SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
...
}

```

}

ServiceConfirmation ::= SEQUENCE

```
{
  elementIdentifier ElementIdentifier,
  domainIdentifier AliasAddress,
  alternates          AlternatePEInfo OPTIONAL,
  securityMode       SecurityMode OPTIONAL,
  timeToLive         INTEGER (1..4294967295) OPTIONAL,
  ...,
  usageSpec          UsageSpecification OPTIONAL
}
```

ServiceRejection ::= SEQUENCE

```
{
  reason             ServiceRejectionReason,
  alternates          AlternatePEInfo OPTIONAL,
  ...
}
```

ServiceRejectionReason ::= CHOICE

```
{
  serviceUnavailable      NULL,
  serviceRedirected       NULL,
  security                 NULL,
  continue                 NULL,
  undefined                NULL,
  ...,
  unknownServiceID        NULL,
  cannotSupportUsageSpec  NULL, -- Cannot comply with proposed spec
  neededFeature           NULL,
  genericDataReason        NULL,
  usageUnavailable        NULL, -- Usage reporting not supported
  unknownUsageSendTo      NULL  -- Usage sendTo could not be resolved
}
```

ServiceRelease ::= SEQUENCE

```
{
  reason             ServiceReleaseReason,
  alternates          AlternatePEInfo OPTIONAL,
  ...
}
```

ServiceReleaseReason ::= CHOICE

```
{
  outOfService      NULL,
  maintenance        NULL,
  terminated         NULL,
  expired            NULL,
  ...
}
```

DescriptorRequest ::= SEQUENCE

```
{
  descriptorID  SEQUENCE OF DescriptorID,
  ...
}
```

```

DescriptorConfirmation ::= SEQUENCE
{
    descriptor          SEQUENCE OF Descriptor,
    ...
}

DescriptorRejection ::= SEQUENCE
{
    reason              DescriptorRejectionReason,
    descriptorID       DescriptorID OPTIONAL,
    ...
}

DescriptorRejectionReason ::= CHOICE
{
    packetSizeExceeded  NULL, -- use other transport type
    illegalID          NULL, -- no descriptor for provided descriptorID
    security            NULL, -- request did not meet security requirements
    hopCountExceeded   NULL,
    noServiceRelationship  NULL,
    undefined          NULL,
    ...,
    neededFeature      NULL,
    genericDataReason  NULL,
    unknownServiceID   NULL -- The serviceID is not recognized by
                        -- the peer element
}

DescriptorIDRequest ::= SEQUENCE
{
    ...
}

DescriptorIDConfirmation ::= SEQUENCE
{
    descriptorInfo      SEQUENCE OF DescriptorInfo,
    ...
}

DescriptorIDRejection ::= SEQUENCE
{
    reason              DescriptorIDRejectionReason,
    ...
}

DescriptorIDRejectionReason ::= CHOICE
{
    noDescriptors      NULL, -- no descriptors to report
    security           NULL, -- request did not meet security requirements
    hopCountExceeded   NULL,
    noServiceRelationship  NULL,
    undefined          NULL,
    ...,
    neededFeature      NULL,
    genericDataReason  NULL,
}

```

```

        unknownServiceID      NULL -- The serviceID is not recognized by
                                -- the peer element
    }

```

```

DescriptorUpdate ::= SEQUENCE
{
    sender      AliasAddress,
    updateInfo  SEQUENCE OF UpdateInformation,
    ...
}

```

```

UpdateInformation ::= SEQUENCE
{
    descriptorInfo CHOICE {
        descriptorID  DescriptorID,
        descriptor    Descriptor,
        ...
    },
    updateType      CHOICE
    {
        added        NULL,
        deleted      NULL,
        changed      NULL,
        ...
    },
    ...
}

```

```

DescriptorUpdateAck ::= SEQUENCE
{
    ...
}

```

```

AccessRequest ::= SEQUENCE
{
    destinationInfo  PartyInformation,
    sourceInfo       PartyInformation OPTIONAL,
    callInfo         CallInformation OPTIONAL,
    usageSpec        UsageSpecification OPTIONAL,
    ...,
    desiredProtocols SEQUENCE OF SupportedProtocols OPTIONAL,
}

```

```

AccessConfirmation ::= SEQUENCE
{
    templates          SEQUENCE OF AddressTemplate,
    partialResponse    BOOLEAN,
    ...,
    supportedProtocols SEQUENCE OF SupportedProtocols OPTIONAL,
    serviceControl     SEQUENCE OF ServiceControlSession OPTIONAL
}

```

```

AccessRejection ::= SEQUENCE
{
    reason            AccessRejectionReason,
    ...,
    serviceControl    SEQUENCE OF ServiceControlSession OPTIONAL
}

```

```

}

AccessRejectionReason ::= CHOICE
{
    noMatch                NULL, -- no template matched the destinationInfo
    packetSizeExceeded     NULL, -- use other transport type
    security                NULL, -- request did not meet security requirements
    hopCountExceeded       NULL,
    needCallInformation     NULL, -- Call Information must be specified
    noServiceRelationship  NULL,
    undefined              NULL,
    ...,
    neededFeature          NULL,
    genericDataReason      NULL,
    destinationUnavailable NULL, -- Destination was resolved but is
                                -- unavailable
    aliasesInconsistent   NULL, -- Multiple aliases identify distinct
                                -- destinations
    resourceUnavailable    NULL, -- One or more required resources are
                                -- unavailable
    incompleteAddress      NULL, -- Destination cannot be distinctly
                                -- identified
    unknownServiceID      NULL, -- The serviceID is not recognized by
                                -- the peer element
    usageUnavailable       NULL, -- Usage reporting not supported
    cannotSupportUsageSpec NULL, -- Cannot comply with proposed spec
    unknownUsageSendTo    NULL -- Usage sendTo could not be resolved
}

```

```

UsageRequest ::= SEQUENCE
{
    callInfo CallInformation,
    usageSpec UsageSpecification,
    ...
}

```

```

UsageConfirmation ::= SEQUENCE
{
    ...
}

```

```

UsageRejection ::= SEQUENCE
{
    reason UsageRejectReason,
    ...
}

```

```

UsageIndication ::= SEQUENCE
{
    callInfo          CallInformation,
    accessTokens      SEQUENCE OF AccessToken OPTIONAL,
    senderRole        Role,
    usageCallStatus   UsageCallStatus,
    srcInfo           PartyInformation OPTIONAL,
    destAddress        PartyInformation,
    startTime         TimeStamp OPTIONAL,
    endTime           TimeStamp OPTIONAL,
}

```



```

        terminationCause      TerminationCause OPTIONAL,
        usageFields           SEQUENCE OF UsageField,
        ...
    }

UsageField ::= SEQUENCE
{
    id                        OBJECT IDENTIFIER,
    value                     OCTET STRING,
    ...
}

UsageRejectReason ::= CHOICE
{
    invalidCall              NULL,
    unavailable              NULL,
    security                 NULL,
    noServiceRelationship   NULL,
    undefined                NULL,
    ...,
    neededFeature           NULL,
    genericDataReason       NULL,
    unknownServiceID       NULL -- The serviceID is not recognized by
                            -- the peer element
}

UsageIndicationConfirmation ::= SEQUENCE
{
    ...
}

UsageIndicationRejection ::= SEQUENCE
{
    reason                   UsageIndicationRejectionReason,
    ...
}

UsageIndicationRejectionReason ::= CHOICE
{
    unknownCall             NULL,
    incomplete              NULL,
    security                NULL,
    noServiceRelationship   NULL,
    undefined                NULL,
    ...,
    neededFeature          NULL,
    genericDataReason       NULL,
    unknownServiceID       NULL -- The serviceID is not recognized by
                            -- the peer element
}

ValidationRequest ::= SEQUENCE
{
    accessToken             SEQUENCE OF AccessToken OPTIONAL,
    destinationInfo        PartyInformation OPTIONAL,
    sourceInfo              PartyInformation OPTIONAL,
    callInfo                CallInformation,
    usageSpec               UsageSpecification OPTIONAL,
    ...
}

```

}

ValidationConfirmation ::= SEQUENCE

```
{
  destinationInfo  PartyInformation OPTIONAL,
  usageSpec        UsageSpecification OPTIONAL,
  ...
}
```

ValidationRejection ::= SEQUENCE

```
{
  reason           ValidationRejectionReason,
  ...
}
```

ValidationRejectionReason ::= CHOICE

```
{
  tokenNotValid  NULL,
  security       NULL, -- request did not meet security requirements
  hopCountExceeded  NULL,
  missingSourceInfo  NULL,
  missingDestInfo  NULL,
  noServiceRelationship  NULL,
  undefined       NULL,
  ...,
  neededFeature  NULL,
  genericDataReason  NULL,
  unknownServiceID  NULL -- The serviceID is not recognized by
                       -- the peer element
}
```

RequestInProgress ::= SEQUENCE

```
{
  delay          INTEGER (1..65535),
  ...,
  serviceControl  SEQUENCE OF ServiceControlSession OPTIONAL
}
```

NonStandardRequest ::= SEQUENCE

```
{
  ...
}
```

NonStandardConfirmation ::= SEQUENCE

```
{
  ...
}
```

NonStandardRejection ::= SEQUENCE

```
{
  reason         NonStandardRejectionReason,
  ...
}
```

NonStandardRejectionReason ::= CHOICE

```
{
  notSupported    NULL,
  noServiceRelationship  NULL,
  ...
}
```

```

    undefined          NULL,
    ...,
    neededFeature     NULL,
    genericDataReason NULL,
    unknownServiceID  NULL -- The serviceID is not recognized by
                        -- the peer element
}

```

```

UnknownMessageResponse ::= SEQUENCE
{
    unknownMessage     OCTET STRING,
    reason              UnknownMessageReason,
    ...
}

```

```

UnknownMessageReason ::= CHOICE
{
    notUnderstood     NULL,
    undefined          NULL,
    ...
}

```

```

AuthenticationRequest ::= SEQUENCE
{
    applicationMessage ApplicationMessage, -- e.g. RAS message in H.323
    ...
}

```

```

ApplicationMessage ::= OCTET STRING

```

```

AuthenticationConfirmation ::= SEQUENCE
{
    ...
}

```

```

AuthenticationRejection ::= SEQUENCE
{
    reason              AuthenticationRejectionReason,
    ...
}

```

```

AuthenticationRejectionReason ::= CHOICE
{
    security              NULL,
    hopCountExceeded     NULL,
    noServiceRelationship NULL,
    undefined             NULL,
    neededFeature        NULL,
    genericDataReason    NULL,
    unknownServiceID     NULL,
    securityWrongSyncTime NULL, -- time server problem or network delay
    securityReplay       NULL, -- replay attack encountered
    securityWrongGeneralID NULL, -- wrong general ID
    securityWrongSendersID NULL, -- wrong senders ID
    securityIntegrityFailed NULL, -- integrity check failed
}

```

```

    securityWrongOID          NULL, -- wrong token OIDs or crypto alg OIDs
    ...
}

```

```

--
-- structures common to multiple messages
--

```

```

AddressTemplate ::= SEQUENCE
{
    pattern          SEQUENCE OF Pattern,
    routeInfo       SEQUENCE OF RouteInformation,
    timeToLive      INTEGER (1..4294967295),
    ...,
    supportedProtocols SEQUENCE OF SupportedProtocols OPTIONAL,
    featureSet      FeatureSet OPTIONAL
}

```

```

Pattern ::= CHOICE
{
    specific        AliasAddress,
    wildcard        AliasAddress,
    range SEQUENCE {
        startOfRange PartyNumber,
        endOfRange   PartyNumber
    },
    ...
}

```

```

RouteInformation ::= SEQUENCE
{
    messageType CHOICE
    {
        sendAccessRequest  NULL,
        sendSetup          NULL,
        nonExistent        NULL,
        ...
    },
    callSpecific    BOOLEAN,
    usageSpec       UsageSpecification OPTIONAL,
    priceInfo       SEQUENCE OF PriceInfoSpec OPTIONAL,
    contacts        SEQUENCE OF ContactInformation,
    type            EndpointType OPTIONAL,
                    -- must be present if messageType = sendSetup
    ...,
    featureSet      FeatureSet OPTIONAL,
    circuitID       CircuitInfo OPTIONAL,
    supportedCircuits SEQUENCE OF CircuitIdentifier OPTIONAL
}

```

```

ContactInformation ::= SEQUENCE
{
    transportAddress AliasAddress,
    priority          INTEGER (0..127),
    transportQoS     TransportQOS OPTIONAL,
    security          SEQUENCE OF SecurityMode OPTIONAL,
    accessTokens     SEQUENCE OF AccessToken OPTIONAL,
    ...,
    multipleCalls    BOOLEAN OPTIONAL,
    featureSet       FeatureSet OPTIONAL,
}

```

```

    circuitID          CircuitInfo    OPTIONAL,
    supportedCircuits SEQUENCE OF CircuitIdentifier OPTIONAL
}

PriceInfoSpec ::= SEQUENCE
{
    currency           IA5String (SIZE(3)),           -- e.g., "USD"
    currencyScale     INTEGER(-127..127),
    validFrom         GlobalTimeStamp OPTIONAL,
    validUntil        GlobalTimeStamp OPTIONAL,
    hoursFrom         IA5String (SIZE(6)) OPTIONAL, -- "HHMMSS" UTC
    hoursUntil        IA5String (SIZE(6)) OPTIONAL, -- "HHMMSS" UTC
    priceElement      SEQUENCE OF PriceElement OPTIONAL,
    priceFormula      IA5String (SIZE(1..2048)) OPTIONAL,
    ...
}

PriceElement ::= SEQUENCE
{
    amount            INTEGER(0..4294967295), -- meter increment
    quantum           INTEGER(0..4294967295), -- each or part thereof
    units CHOICE
    {
        seconds       NULL,
        packets       NULL,
        bytes         NULL,
        initial       NULL,
        minimum       NULL,
        maximum       NULL,
        ...
    },
    ...
}

Descriptor ::= SEQUENCE
{
    descriptorInfo    DescriptorInfo,
    templates         SEQUENCE OF AddressTemplate,
    gatekeeperID     GatekeeperIdentifier OPTIONAL,
    ...
}

DescriptorInfo ::= SEQUENCE
{
    descriptorID     DescriptorID,
    lastChanged     GlobalTimeStamp,
    ...
}

AlternatePEInfo ::= SEQUENCE
{
    alternatePE      SEQUENCE OF AlternatePE,
    alternateIsPermanent BOOLEAN,
    ...
}

AlternatePE ::= SEQUENCE
{
    contactAddress   AliasAddress,
    priority         INTEGER (1..127),
    elementIdentifier ElementIdentifier OPTIONAL,
}

```

```

...
}

AccessToken ::= CHOICE
{
    token          ClearToken,
    cryptoToken    CryptoH323Token,
    ...,
    genericData    GenericData
}

CallInformation ::= SEQUENCE
{
    callIdentifier      CallIdentifier,
    conferenceID        ConferenceIdentifier,
    ...,
    circuitID           CircuitInfo    OPTIONAL
}

UsageCallStatus ::= CHOICE
{
    preConnect          NULL, -- Call has not started
    callInProgress      NULL, -- Call is in progress
    callEnded           NULL, -- Call ended
    ...,
    registrationLost    NULL -- Uncertain if call ended or not
}

UserInformation ::= SEQUENCE
{
    userIdentifier      AliasAddress,
    userAuthenticatorSEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}

UsageSpecification ::= SEQUENCE
{
    sendTo              ElementIdentifier,
    when SEQUENCE
    {
        never           NULL OPTIONAL,
        start           NULL OPTIONAL,
        end             NULL OPTIONAL,
        period          INTEGER(1..65535) OPTIONAL, -- in seconds
        failures        NULL OPTIONAL,
        ...
    },
    required            SEQUENCE OF OBJECT IDENTIFIER,
    preferred           SEQUENCE OF OBJECT IDENTIFIER,
    ...,
    sendToPEAddress    AliasAddress OPTIONAL
}

PartyInformation ::= SEQUENCE
{
    logicalAddresses SEQUENCE OF AliasAddress,
    domainIdentifier AliasAddress OPTIONAL,
    transportAddress AliasAddress OPTIONAL,
    endpointType     EndpointType OPTIONAL,

```

```

        userInfo      UserInformation OPTIONAL,
        timeZone      TimeZone OPTIONAL,
        ...
    }

```

```

Role ::= CHOICE
{
    originator      NULL,
    destination     NULL,
    nonStandardData NonStandardParameter,
    ...
}

```

```

TimeZone ::= INTEGER (-43200..43200)    -- number of seconds relative to UTC
                                           -- including DST if appropriate

```

```

TerminationCause ::= SEQUENCE
{
    releaseCompleteReason      ReleaseCompleteReason,
    causeIE                    INTEGER (1..65535) OPTIONAL,
    nonStandardData            NonStandardParameter OPTIONAL,
    ...
}

```

```

ProtocolVersion ::= OBJECT IDENTIFIER
-- shall be set to
-- {itu-t(0) recommendation(0) h(8) h-225-0(2250) annex(1) g(7) version(0) 2}
-- in field annexGversion;
-- {itu-t(0) recommendation(0) h(8) 501 version(0) 1}
-- in field version

```

```

DescriptorID ::= GloballyUniqueID

```

```

ElementIdentifier ::= BMPString (SIZE(1..128))

```

```

GlobalTimeStamp ::= IA5String (SIZE(14))
-- UTC, in the form YYYYMMDDHHmmSS
-- where YYYY = year, MM = month, DD = day,
-- HH = hour, mm = minute, SS = second
-- (for example, 19981219120000 for noon
-- 19 December 1998)

```

```
--  
-- REPOSITORY FOR APPLICATION SPECIFIC DATA --  
--  
-- Annex-G profile data  
--  
idAnnexGProfiles          INTEGER ::= 0  
idAnnexGProfileA         INTEGER ::= 1  
  
annexGProfileA    EnumeratedParameter ::=  
{  
  id          standard:idAnnexGProfileA  
}  
  
END -- of H501-MESSAGES
```