

**TTC標準**  
Standard

JT-H248.1

メディアゲートウェイ制御  
プロトコル

MEDIA GATEWAY CONTROL PROTOCOL

第2版

2006年6月1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

1. 規定範囲 .....	9
2. 参照規格 .....	9
2.1 規範的参照 .....	9
2.2 情報供給参照 .....	11
3. 定義 .....	11
4. 略語 .....	12
5. 規則 .....	13
6. コネクションモデル .....	13
6.1 CONTEXT .....	16
6.1.1 Contextの属性とディスクリプタ .....	16
6.1.2 Contextの生成、削除、変更 .....	16
6.2 TERMINATION .....	16
6.2.1 Termination変更(Dynamics) .....	19
6.2.2 TerminationID .....	19
6.2.3 パッケージ .....	20
6.2.4 Terminationのプロパティとディスクリプタ .....	20
6.2.5 Root Termination .....	22
6.3 ワイルドカード指定原則 (WILDCARDING PRINCIPLES) .....	22
6.3.1 特定のContextIDにおけるTerminationIDのワイルドカード指定 .....	23
6.3.2 ワイルドカード指定(ALL)されたContextID における特定のTerminationID .....	23
6.3.3 ワイルドカード指定(ALL)されたContextIDにおけるTerminationIDのワイルドカード指定 .....	24
6.3.4 ワイルドカードの応答 .....	24
7. コマンド .....	25
7.1 ディスクリプタ .....	26
7.1.1 パラメータの指定法 .....	26
7.1.2 Modemディスクリプタ .....	26
7.1.3 Multiplexディスクリプタ .....	26
7.1.4 Mediaディスクリプタ .....	27
7.1.5 Termination Stateディスクリプタ .....	27
7.1.6 Streamディスクリプタ .....	28
7.1.7 LocalControlディスクリプタ .....	29
7.1.8 LocalディスクリプタとRemoteディスクリプタ .....	31
7.1.9 Eventsディスクリプタ .....	33
7.1.10 EventBuffer ディスクリプタ .....	36
7.1.11 Signals ディスクリプタ .....	36
7.1.12 Auditディスクリプタ .....	37
7.1.13 ServiceChangeディスクリプタ .....	38
7.1.14 DigitMapディスクリプタ .....	38
7.1.15 Statistics ディスクリプタ .....	43
7.1.16 Packagesディスクリプタ .....	43
7.1.17 ObservedEventsディスクリプタ .....	43
7.1.18 Topologyディスクリプタ .....	44
7.1.19 ContextAttributeディスクリプタ .....	46
7.1.20 Errorディスクリプタ .....	47
7.2 コマンドAPI (APPLICATION PROGRAMMING INTERFACE) .....	47
7.2.1 Addコマンド .....	47
7.2.2 Modifyコマンド .....	49
7.2.3 Subtractコマンド .....	50
7.2.4 Moveコマンド .....	51
7.2.5 AuditValueコマンド .....	52
7.2.6 AuditCapabilitiesコマンド .....	55
7.2.7 Notifyコマンド .....	57
7.2.8 ServiceChangeコマンド .....	57

7.2.9 Context属性の操作と監査.....	61
7.2.10 汎用的なコマンド構文.....	62
8. TRANSACTIONS.....	63
8.1 共通パラメータ.....	64
8.1.1 Transaction識別子.....	64
8.1.2 Context識別子.....	64
8.2 TRANSACTION API (APPLICATION PROGRAMMING INTERFACE).....	64
8.2.1 TransactionRequest.....	64
8.2.2 TransactionReply.....	65
8.2.3 TransactionPending.....	66
8.3 メッセージ.....	67
9. トランスポート.....	67
9.1 コマンド順序.....	68
9.2 リスタート殺到の防止.....	69
9.3 NOTIFY (通知) 殺到の防止.....	69
10. セキュリティ.....	69
10.1 プロトコルコネクションの保護.....	69
10.2 暫定AHスキーム.....	70
10.3 メディアコネクションの保護.....	70
11. MG-MGC制御インタフェース.....	71
11.1 複数仮想MG.....	71
11.2 コールドスタート.....	72
11.3 プロトコルバージョンの交渉.....	72
11.4 MG故障.....	73
11.5 MGC故障.....	73
11.6 MGC-MG 間制御リンク監視.....	74
12. パッケージ定義.....	74
12.1 パッケージ定義のガイドライン.....	75
12.1.1 Package.....	75
12.1.2 Properties.....	75
12.1.3 Events.....	76
12.1.4 Signals.....	77
12.1.5 Statistics.....	77
12.1.6 Error Codes.....	78
12.1.7 Procedures.....	78
12.2 EVENTSとSIGNALSに対するパラメータ定義のガイドライン.....	78
12.3 識別子.....	79
12.4 パッケージ登録.....	79
13. プロファイルの定義.....	79
14. IANA.....	80
14.1 パッケージ.....	80
14.2 エラーコード.....	80
14.3 SERVICECHANGE理由.....	80
14.4 プロファイル.....	80
付属資料A バイナリ符号化 (BINARY ENCODING OF THE PROTOCOL).....	82
A.1 ワイルドカードの符号化.....	82
A.2 ASN.1構文仕様.....	83
A.3 DIGITMAPとPATH NAME.....	108
付属資料B テキスト符号化 (TEXT ENCODING OF THE PROTOCOL).....	110
B.1 ワイルドカードの符号化.....	110
B.2 ABNF仕様.....	110
B.3 16進オクテット符号化.....	130
B.4 16進オクテット列.....	130

付属資料C	メディアストリーム・プロパティのタグ (TAGS FOR MEDIA STREAM PROPERTIES) .....	131
C.1	一般メディア属性 (GENERAL MEDIA ATTRIBUTES).....	132
C.2	MUXプロパティ (MUX PROPERTIES) .....	133
C.3	一般ベアラプロパティ (GENERAL BEARER PROPERTIES).....	133
C.4	一般ATMプロパティ (GENERAL ATM PROPERTIES).....	133
C.5	フレームリレー .....	135
C.6	IP .....	135
C.7	ATM AAL2.....	136
C.8	ATM AAL1.....	137
C.9	ベアラ能力 (BEARER CAPABILITIES) .....	138
C.10	AAL5プロパティ .....	145
C.11	SDP EQUIVALENTS.....	145
C.12	H.245 .....	146
付属資料D	IPによるトランスポート (TRANSPORT OVER IP).....	147
D.1	アプリケーションレベルフレーミング(ALF)によるIP/UDPでのトランスポート .....	147
D.1.1	At-Most-Once機能の提供 .....	147
D.1.2	Transaction識別子と3-wayハンドシェーク .....	147
D.1.3	再送タイマの計算.....	148
D.1.4	暫定応答 .....	149
D.1.5	要求、応答、確認の繰り返し .....	149
D.2	TCPの使用.....	150
D.2.1	At-Most-Once機能の提供 .....	151
D.2.2	Transaction識別子と3-wayハンドシェーク .....	151
D.2.3	再送タイマの計算.....	151
D.2.4	暫定応答 .....	151
D.2.5	コマンド順序.....	151
付属資料E	BASICパッケージ (BASIC PACKAGES).....	151
E.1	GENERIC.....	152
E.1.1	Properties .....	152
E.1.2	Events.....	152
E.1.3	Signals.....	154
E.1.4	Statistics .....	154
E.2	BASE ROOT PACKAGE .....	154
E.2.1	Properties .....	154
E.2.2	Events.....	156
E.2.3	Signals.....	157
E.2.4	Statistics .....	157
E.2.5	Procedures .....	157
E.3	TONE GENERATOR パッケージ.....	157
E.3.1	Properties .....	157
E.3.2	Events.....	157
E.3.3	Signals.....	157
E.3.4	Statistics .....	158
E.3.5	Procedures .....	158
E.4	TONE DETECTION パッケージ .....	158
E.4.1	Properties .....	159
E.4.2	Events.....	159
E.4.3	Signals.....	161
E.4.4	Statistics .....	161
E.4.5	Procedures .....	161
E.5	BASIC DTMF GENERATOR パッケージ .....	161
E.5.1	Properties .....	161
E.5.2	Events.....	162
E.5.3	Signals.....	162
E.5.4	Statistics .....	163
E.5.5	Procedures .....	163
E.6	DTMF DETECTION パッケージ .....	163
E.6.1	Properties .....	164
E.6.2	Events.....	164
E.6.3	Signals.....	165
E.6.4	Statistics .....	165
E.6.5	Procedures .....	165
E.7	CALL PROGRESS TONES GENERATOR パッケージ .....	165

E.7.1	Properties .....	165
E.7.3	Signals .....	165
E.7.4	Statistics .....	166
E.7.5	Procedures .....	166
E.8	CALL PROGRESS TONES DETECTION パッケージ .....	166
E.8.1	Properties .....	167
E.8.2	Events.....	167
E.8.3	Signals.....	167
E.8.4	Statistics .....	167
E.8.5	Procedures .....	167
E.9	ANALOG LINE SUPERVISION パッケージ .....	167
E.9.1	Properties .....	167
E.9.2	Events.....	167
E.9.3	Signals.....	170
E.9.4	Statistics .....	170
E.9.5	Error codes .....	170
E.9.6	Procedures .....	171
E.10	BASIC CONTINUITY パッケージ.....	171
E.10.1	Properties .....	171
E.10.2	Events.....	171
E.10.3	Signals.....	171
E.10.4	Statistics .....	172
E.10.5	Procedures .....	172
E.11	NETWORK パッケージ.....	172
E.11.1	Properties .....	173
E.11.2	Events.....	173
E.11.3	Signals.....	174
E.11.4	Statistics .....	174
E.11.5	Procedures .....	175
E.12	RTP パッケージ.....	175
E.12.1	Properties .....	175
E.12.2	Events.....	175
E.12.3	Signals.....	176
E.12.4	Statistics .....	176
E.12.5	Procedures .....	177
E.13	TDM CIRCUIT パッケージ .....	177
E.13.1	Properties .....	177
E.13.2	Events.....	178
E.13.3	Signals.....	178
E.13.4	Statistics .....	178
E.13.5	Procedures .....	178
E.14	SEGMENTATION パッケージ.....	178
E.14.1	Properties .....	179
E.14.2	Events .....	180
E.14.3	Signals .....	180
E.14.4	Statistics.....	180
E.14.5	Error Codes.....	180
E.14.6	Procedures.....	180
E.15	NOTIFICATION BEHAVIOUR .....	181
E.15.1	Properties.....	182
E.15.2	Events .....	182
E.15.3	Signals .....	182
E.15.4	Statistics.....	182
E.15.5	Procedures.....	182
付属資料F	サービスチェンジ手順 (SERVICECHANGE PROCEDURES).....	188
F.1	導入.....	188
F.2	制御リンク(CONTROL ASSOCIATION)の定義 .....	189
F.3	SERVICECHANGE手順を起こすイベント.....	190
F.3.1	MG Registration .....	190
F.3.2	MG Re-Registration.....	190
F.3.3	MG Service Cancellation.....	191
F.3.4	MG Service Restoration .....	191
F.3.5	MG Redundant Takeover .....	191
F.3.6	MG Lost Communication.....	192
F.3.7	MG Capability Change .....	192
F.3.8	MGC Initiated MG Re-Registration.....	193
F.3.9	MGC Initiated Service Restoration .....	193

F.3.10	MGC Initiated Service Cancellation .....	193
F.3.11	MGC Redundant Failover .....	194
F.4	SERVICECHANGE要素の記述 .....	194
F.4.1	ServiceChangeMethod .....	194
F.5	SERVICECHANGEパラメータの使用 .....	197
F.5.1	ServiceChangeMethod .....	197
F.5.2	ServiceChangeReason .....	197
F.5.3	ServiceChangeDelay .....	198
F.5.4	ServiceChangeAddress .....	198
F.5.5	ServiceChangeProfile .....	199
F.5.6	ServiceChangeVersion .....	199
F.5.7	ServiceChangeMgclD .....	199
F.5.8	TimeStamp .....	199
F.6	SERVICECHANGE対TERMINATIONSTATE .....	199
付録I	コールフローの例 .....	201
I.1	レジデンシャルゲートウェイ間の呼 .....	201
I.1.1	アイドル中状態におけるレジデンシャルゲートウェイのアナログ回線terminationの設定 .....	201
I.1.2	発信先番号の収集とterminationの初期化 .....	203
付録II	H.248パッケージテンプレート .....	214

## <参考>

### 1. 国際勧告などとの関連

本標準は、メディアゲートウェイ制御プロトコルについて規定しており、2005年7月に開催されたITU-T SG16会合においてAAPにConsentされ2005年9月にApproveされたITU-T勧告H.248.1 Version3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

#### 2.4 原勧告との章立て構成比較表

上記国際勧告などとの章立て構成の相違はない。

### 3. 改版の履歴

版数	発行日	改版内容
第1版	2002年11月28日	制定
第2版	2006年6月1日	第2版制定。第1版からの改版内容はITU-T勧告H.248.1 Version2からVersion3の改版内容に準ずる。主な改版内容は以下の通り。 <ul style="list-style-type: none"><li>• Packageで実装されていないPropertyを指定した場合のエラーについての記述の追記</li><li>• IEPS(International Emergency Preference Scheme)に関する記述の追記</li><li>• OutOfService状態のTerminationを示すFlagについての記述の追記</li><li>• Segmentation Packageの追加</li><li>• Packageの要求条件定義の追加</li><li>• ストリームレベルの統計情報の追加</li><li>• Signalに関するRequestIDの追加</li><li>• 方向をあらわす基本Signalパラメータの追加</li><li>• 新トポロジータイプの追加 (OnewayexternalとOnewayboth)</li><li>• SignalList上の内部Signal遅延タイムに関する記述を追記</li><li>• ContextIDListの追加</li><li>• TerminationIDListに関する記述の追記</li><li>• ServiceChange手順に関する記述の明確化</li><li>• MGCによるNotify受信規制機能に関する記述の追記</li><li>• AuditValueコマンドの戻り値にフィルタ条件を設定する機能の追記</li></ul>

### 4. 工業所有権

本標準に関わる「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ホームページでご覧に



なれます。

## 5. その他

参照している勧告、標準など

ITU-T勧告: E.180/Q.35, G.711, H.221, H.223, H.225.0, H.226, H.235, H.246, H.248.4, H.248.5, H.248.8, H.323, I.363.1, I.363.2, I.363.5, I.366.1, I.366.2, I.371, Q.724, Q.763, Q.764, Q.765.5, Q.931, Q.1902.4 Q.2630.1, Q.2931, Q.2941.1, Q.2961, Q.2961.2, Q.2965.1, Q.2965.2, V.76, X.213, H.680, X.690

RFC: 768, 791, 793, 1006, 1661, 1889, 1890, 2234, 2327, 2401, 2402, 2406, 2460, 2543, 2805

## 6. 標準作成部門

信号制御専門委員会

### 概要

本標準は、物理的に分離されたマルチメディアゲートウェイの構成要素間、つまり、メディアゲートウェイおよびメディアゲートウェイコントローラの間で使用されるプロトコルについて規定する。本標準で規定されるプロトコルは、RFC2805で規定されるメディアゲートウェイ制御プロトコルの要求条件を満たしている。

ITU-T勧告H.248.1第1版準拠を主張する製品は、ITU-T勧告H.248.1(06/2000オリジナル版及び03/2002再発行版)の全ての必須要求項目に適合しなければならない。

TTC標準JT-H248.1第1版(ITU-T勧告H.248.1第2版)準拠を主張する製品は、TTC標準JT-H248.1第1版の全ての必須要求項目に適合しなければならない。

TTC標準JT-H248.1第2版(ITU-T勧告H.248.1第3版)準拠を主張する製品は、本標準の全ての必須要求項目に適合しなければならない。

H.248.1の製品は、使用するプロトコルの版数をServiceChangeVersionにより示さなければならない。

H.248.1第1版: '1'

JT-H248.1第1版(H.248.1第2版): '2'

JT-H248.1第2版(H.248.1第3版): '3'

## 1. 規定範囲

本標準は、物理的に分離されたマルチメディアゲートウェイの構成要素間にて使用されるプロトコルについて規定する。システムという観点で見ると、二つ以上の物理的な装置に潜在的に分配されたサブコンポーネントを持つ分離されたゲートウェイと、H.246で述べられるような単一のゲートウェイとでは、機能的な違いはない。本標準は、ゲートウェイ、多地点会議制御装置(multipoint control unit)もしくは自動音声応答装置(interactive voice response unit : IVR)がどのように動作するのかは規定しない。その代わりに、これらのアプリケーションに適した一般的な構造を述べる。

パケット網インタフェースは、IP、ATMもしくはそれ以外のものを含んでもよい。このインタフェースは、トーンシグナリング、ISDN、ISUP、QSIGおよびGSMを含む、多様なSCNシグナリングシステムをサポートする。これらのシグナリングシステムの地域による違いは、適用可能な範囲でサポートされる。

## 2. 参照規格

以下のITU-T勧告および他の参照規格は、このテキストでの参照を通じて、本標準の規定を構成している規定が記述されているものである。出版時においては、下記の版は有効であった。すべての標準および他の参照規格は改訂版に従う。そのため、本標準のすべてのユーザには、以下に挙げた標準と参照規格の最新版の適用を推奨する。現在有効なITU-T勧告のリストは定期的に出版されている。

### 2.1 規範的参照

- ITU-T Recommendation E.106 (2003), “International Emergency Preference Scheme (IEPS) for disaster relief operations.”
- ITU-T Recommendation H.225.0 (2003), “Call signalling protocols and media stream packetization for packet-based multimedia communication systems” .
- ITU-T Recommendation H.235 (2003), Corr.1 (01/2005) “Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals” .
- ITU-T Recommendation H.245 (2005), “Control protocol for multimedia communication”.
- ITU-T Recommendation H.246 (1998), “Interworking of H-series multimedia terminals with H-series multimedia terminals and voice/voiceband terminals on GSTN and ISDN”.
- ITU-T Recommendation H.248.4 (2000) Corr.1 (2004), “Gateway control protocol: Transport over Stream Control Transmission Protocol (SCTP)”.
- ITU-T Recommendation H.248.5 (2000), “Gateway control protocol: Transport over ATM”.
- ITU-T Recommendation H.248.8 (2005), “Gateway control protocol: Error code and service change reason description”.
- ITU-T Recommendation H.248.14 (2002), “Gateway control protocol: Inactivity Timer Package”.
- ITU-T Recommendation H.323 (2003), “Packet-based multimedia communications systems”.
- ITU-T Recommendation I.363.1 (1996),” B-ISDN ATM Adaptation Layer specification: Type 1 AAL”.
- ITU-T Recommendation I.363.2 (2000), “B-ISDN ATM Adaptation Layer specification: Type 2 AAL”.
- ITU-T Recommendation I.363.5 (1996), “B-ISDN ATM Adaptation Layer specification: Type 5 AAL”.
- ITU-T Recommendation I.366.1 (1998), “Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2”.

- ITU-T Recommendation I.366.2 (2000) Corr.1 (2002) “AAL type 2 service specific convergence sublayer for narrow-band services”.
- ITU-T Recommendation I.371 (2004), “Traffic control and congestion control in B-ISDN”.
- ITU-T Recommendation Q.763 (1999) Amd.3 (2004), “Signalling System No. 7 – ISDN user part formats and codes”.
- ITU-T Recommendation Q.765.5 (2004), “Signalling System No. 7 – Application transport mechanism: Bearer Independent Call Control (BICC)”.
- ITU-T Recommendation Q.931 (1998) Amd.1 (2002), “ISDN user-network interface layer 3 specification for basic call control”.
- ITU-T Recommendation Q.2630.1 (1999), “AAL type 2 signalling protocol – Capability Set 1”.
- ITU-T Recommendation Q.2931 (1995) Amd.4 (1999), “Digital Subscriber Signalling System No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control”.
- ITU-T Recommendation Q.2941.1 (1997), “Digital Subscriber Signalling System No. 2 – Generic identifier transport”.
- ITU-T Recommendation Q.2961.1 (1995), “Digital Subscriber Signalling System No. 2 – Additional traffic parameters: Additional signalling capabilities to support traffic parameters for the tagging option and the sustainable call rate parameter set”.
- ITU-T Recommendation Q.2961.2 (1997) Corr.1 (1999), “Digital Subscriber Signalling System No. 2 – Additional traffic parameters: Support of ATM transfer capability in the broadband bearer capability information element”.
- ITU-T Recommendation Q.2965.1 (1999) Amd.1 (2000), “Digital subscriber signalling system No. 2 – Support of Quality of Service classes”.
- ITU-T Recommendation Q.2965.2 (1999), “Digital subscriber signalling system No. 2 – Signalling of individual Quality of Service parameters”.
- ITU-T Recommendation V.76 (1996) Corr.1 (2005), “Generic multiplexer using V.42 LAPM-based procedures”.
- ITU-T Recommendation X.213 (2001) | ISO/IEC 8348:2002, “Information technology – Open Systems Interconnection – Network service definition plus Amendment 1 (1997), Addition of the Internet protocol address format identifier”.
- ITU-T Recommendation X.680 (2002) Amd.2 (2004) | ISO/IEC 8824-1:2004, “Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation”.
- ITU-T Recommendation X.690 (2002) Amd.1 (2003) | ISO/IEC 8825-1:2002 FDAM2:2004, “Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)”.
- ATM Forum (1996), “ATM User-Network Interface (UNI) Signalling Specification – Version 4.0”.
- IETF RFC 1006 (1987), “ISO Transport Service on top of the TCP, Version 3”.
- IETF RFC 2234 (1997), “Augmented BNF for Syntax Specifications: ABNF”.

- IETF RFC 2327 (1998), “SDP: Session Description Protocol”.
- IETF RFC 2402 (1998), “IP Authentication Header”.
- IETF RFC 2406 (1998), “IP Encapsulating Security Payload (ESP)”.

## 2.2 情報供給参照

- ITU-T Recommendation E.180/Q.35 (1998), “Technical characteristics of tones for the telephone service”.
- ITU-T Recommendation G.711 (1988), “Pulse code modulation (PCM) of voice frequencies”.
- ITU-T Recommendation H.221 (2004), “Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices”.
- ITU-T Recommendation H.223 (2001), “Multiplexing protocol for low bit rate multimedia communication”.
- ITU-T Recommendation H.226 (1998), “Channel aggregation protocol for multilink operation on circuit-switched networks”.
- ITU-T Recommendation Q.724 (1998) Amd.1 (1993), “Telephone user part signalling procedures”.
- ITU-T Recommendation Q.764 (1999) Amd.3 (2004), Signalling system No. 7 – ISDN user part signalling procedures.
- ITU-T Recommendation Q.1902.4 (2001) Amd.2 (2004), “Bearer Independent Call Control protocol – (Capability Set 2): Basic call procedures”.
- IETF RFC 768 (1980), “User Datagram Protocol”.
- IETF RFC 791 (1981), “Internet protocol”.
- IETF RFC 793 (1981), “Transmission control protocol”.
- IETF RFC 1661 (1994), “The Point-to-Point Protocol (PPP)”.
- IETF RFC 2401 (1998), “Security Architecture for the Internet Protocol”.
- IETF RFC 2460 (1998), “Internet Protocol, Version 6 (IPv6) Specification”.
- IETF RFC 2805 (2000), “Media Gateway Control Protocol Architecture and Requirements”.
- IETF RFC 3261 (2002), “SIP: Session Initiation Protocol”.
- IETF RFC 3550 (2003), “RTP: A Transport Protocol for Real-Time Applications”.
- IETF RFC 3551 (2003), “RTP Profile for Audio and Video Conferences with Minimal Control”.

## 3. 定義

本標準のために、以下の定義を適用する。

**アクセスゲートウェイ(Access Gateway)** : ISDNのようなユーザ-ネットワークインタフェース(UNI)を提供するゲートウェイ。

**ディスクリプタ(Descriptor)** : 関連するプロパティをグループ化した、プロトコルの構文上の要素。例えば、MG上のあるメディアフローのプロパティは、適切なディスクリプタを含むコマンドにより、MGCから制御され得る。

メディアゲートウェイ(MG) : MGは、あるネットワークから提供されたメディアを他のネットワークから要求されるフォーマットに変換する。例えば、MGは、回線交換網からのベアラチャネル(DS0等)やパケット網からのメディアストリーム(IP網でのRTPストリーム等)を終端する。このゲートウェイは音声、映像およびT.120の単独またはいかなる組合せでも処理する能力を持ってよく、さらに全二重メディア翻訳の能力を持つことになる。MGはまた音声/映像メッセージを再生し、他のIVR機能を実行してもよいし、もしくはメディア会議を実行してもよい。

メディアゲートウェイコントローラ(MGC) : MGにおけるメディアチャネルのコネクション制御に関わる呼状態を制御する。

多地点会議制御装置(Multipoint Control Unit : MCU) : 通常は、音声、映像およびデータの処理を含む複数ユーザ会議の設定と調整を制御する装置。

レジデンシャルゲートウェイ(Residential Gateway) : アナログ回線とパケット網を相互接続するゲートウェイ。レジデンシャルゲートウェイは通常、一つか二つのアナログ回線を有し、顧客宅内に設置される。

SCN FASシグナリングゲートウェイ(SCN FAS Signalling Gateway) : この機能は、SS7、ISDNもしくはその他のシグナリングリンクを終端するSCNシグナリングインタフェースを持ち、呼制御チャネルとベアラチャネルとが物理的に同じ媒体を共有する。

SCN NFASシグナリングゲートウェイ(SCN NFAS Signalling Gateway) : この機能は、SS7もしくはその他のシグナリングリンクを終端するSCNシグナリングインタフェースを持ち、呼制御チャネルはベアラチャネルと分けられている。

ストリーム(Stream) : 呼や会議の一部としてMGによって送受信される両方向のメディアもしくは制御フロー。

トランク(Trunk) : T1もしくはE1回線上のDS0のような、二つの交換システム間における通信チャネル。

トランクゲートウェイ(Trunking Gateway) : 通常は多くのデジタル回線を終端する、SCN網とパケット網の間のゲートウェイ。

#### 4. 略語

本標準では、以下の略語を使う :

AAD	Average Acknowledgement Delay :
AAL	ATM Adaptation Layer :
ADEV	Average Deviation :
ALF	Application Layer Framing : アプリケーションレイヤフレーミング
ATM	Asynchronous Transfer Mode : 非同期転送モード
C	Context : コンテキスト
CAS	Channel Associated Signalling : 個別線信号方式
DNS	Domain Name System : ドメインネームシステム
DTMF	Dual Tone Multi-Frequency : 二周波変調方式
FAS	Facility Associated Signalling : 対応信号方式
GSM	Global System for Mobile communications : ヨーロッパにおけるデジタル携帯電話システムの標準規格
GW	GateWay : ゲートウェイ
IANA	Internet Assigned Numbers Authority : インターネット番号割当機関
IEPS	International Emergency Preference Scheme :
IP	Internet Protocol : インターネットプロトコル

IS	In-Service: サービス中
ISUP	ISDN User Part : ISDNユーザパート
IVR	Interactive Voice Response : 自動音声応答装置
MG	Media Gateway : メディアゲートウェイ
MGC	Media Gateway Controller : メディアゲートウェイコントローラ
MWD	Maximum Waiting Delay :
NFAS	Non-Facility Associated Signalling : 非対応信号方式 (Dチャンネル共用)
OoS	Out-of-Service : 非サービス中
PRI	Primary Rate Interface : 一次群速度インタフェース
PSTN	Public Switched Telephone Network : 公衆回線電話網
QoS	Quality of Service : サービス品質
RTP	Real-time Transport Protocol : リアルタイムトランスポートプロトコル
SC	ServiceChange :
SCN	Switched Circuit Network : 回線交換網
SG	Signalling Gateway : シグナリングゲートウェイ
SS7	Signalling System No.7 : 共通線信号システムNo.7
T, Term	Termination: ターミネーション

## 5. 規則

本標準は以下の規則を使用する :

“しなければならない”は、必須要求項目を示す。

“すべきである”は、推奨するが、オプションであることを示す。

“してもよい”または“できる”は、動作上のオプションであることを示す。

## 6. コネクションモデル

コネクションモデルは、MGCにより制御されるMGの論理エンティティ、またはオブジェクトを表す。コネクションモデルは主にContextとTerminationを用いて記述される。

Terminationは一つまたは複数のストリームの送信または受信、あるいは送信と受信の両方を行う。マルチメディア会議においては、Terminationはマルチメディアの処理を行うことが可能であり、複数メディアストリームの送信または受信を行う。メディアストリームパラメータは、ベアラパラメータと同様にTermination内にカプセル化される。

Contextは複数Termination間のアソシエーションを示す。他のどのcontextにも存在しない、それ故、他のTerminationとアソシエーションを持たないTerminationを含むNull Contextと呼ばれる特殊なContextがある。例えば、接続されていないアクセスゲートウェイにおいて、全ての空き回線はNull Contextの中のTerminationとして表現される。

以下はこれらの概念を図示したものである。図6-1/JT-H248.1は、いくつかの例を示しているが、全ての例を示しているわけではない。各Contextに示されたアスタリスクは、Contextに含まれる論理的なTerminationのアソシエーションを示す。

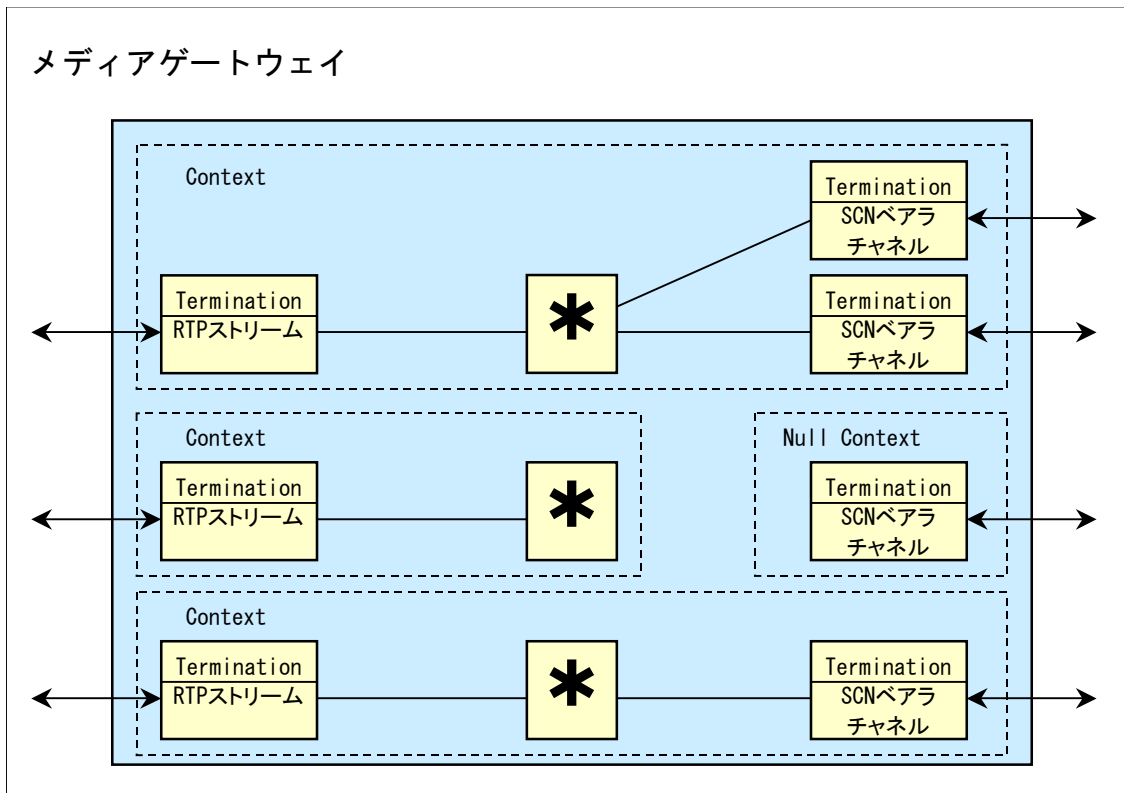


図6-1/JT-H248.1 JT-H248.1接続モデルの例  
(ITU-T H.248.1)

図6-2/JT-H248.1および図6-3/JT-H248.1は、分離されたアクセスゲートウェイにおいてコールウェイティン  
グシナリオを完成させるための例を示している。これらの図は、Context間のTerminationの配置を示している。  
両方向の音声呼におけるTermination T1とTermination T2が、Context C1に属しており、Termination T3からの2  
番目の音声呼がT1を待っているところである。T3のみがContext C2に存在している。T1はT3からの呼を受け  
付け、T2を保留状態に置く。この動作によって、図6-3/JT-H248.1に示すように、T1がContext C2に移動する  
ことになる。

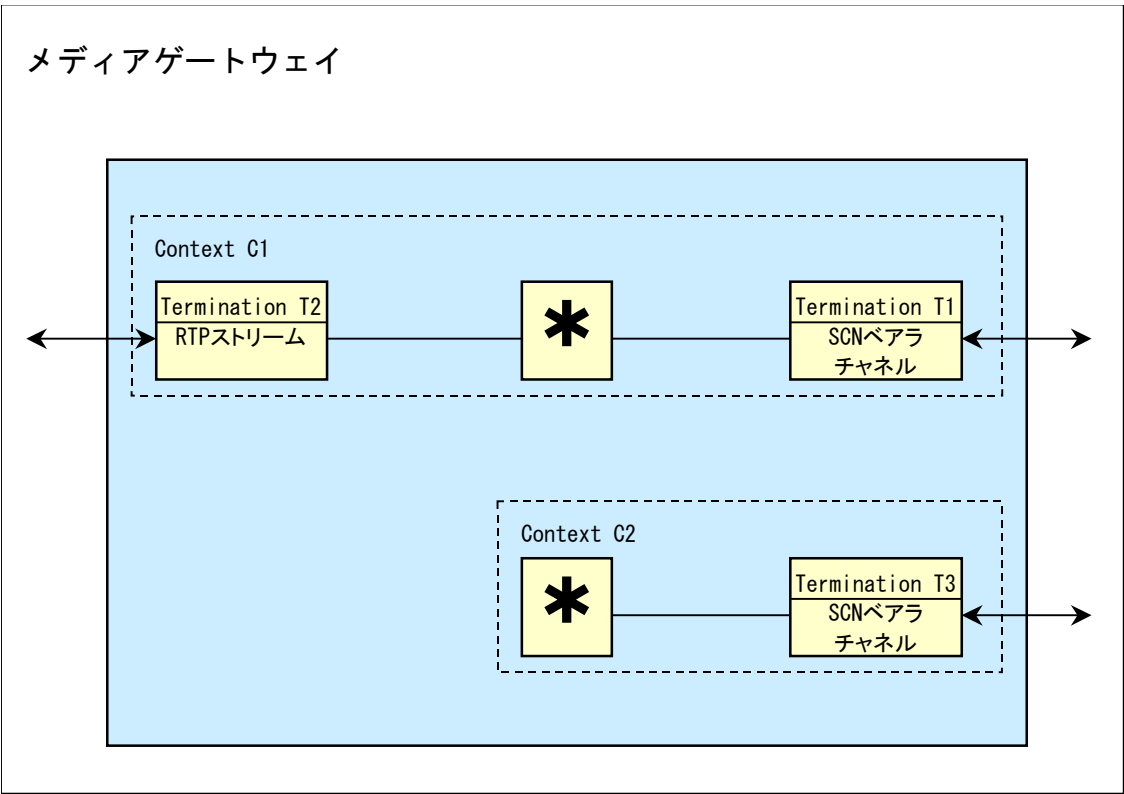


図6-2/JT-H248.1 コールウェイティングシナリオの例/T1呼出  
(ITU-T H.248.1)

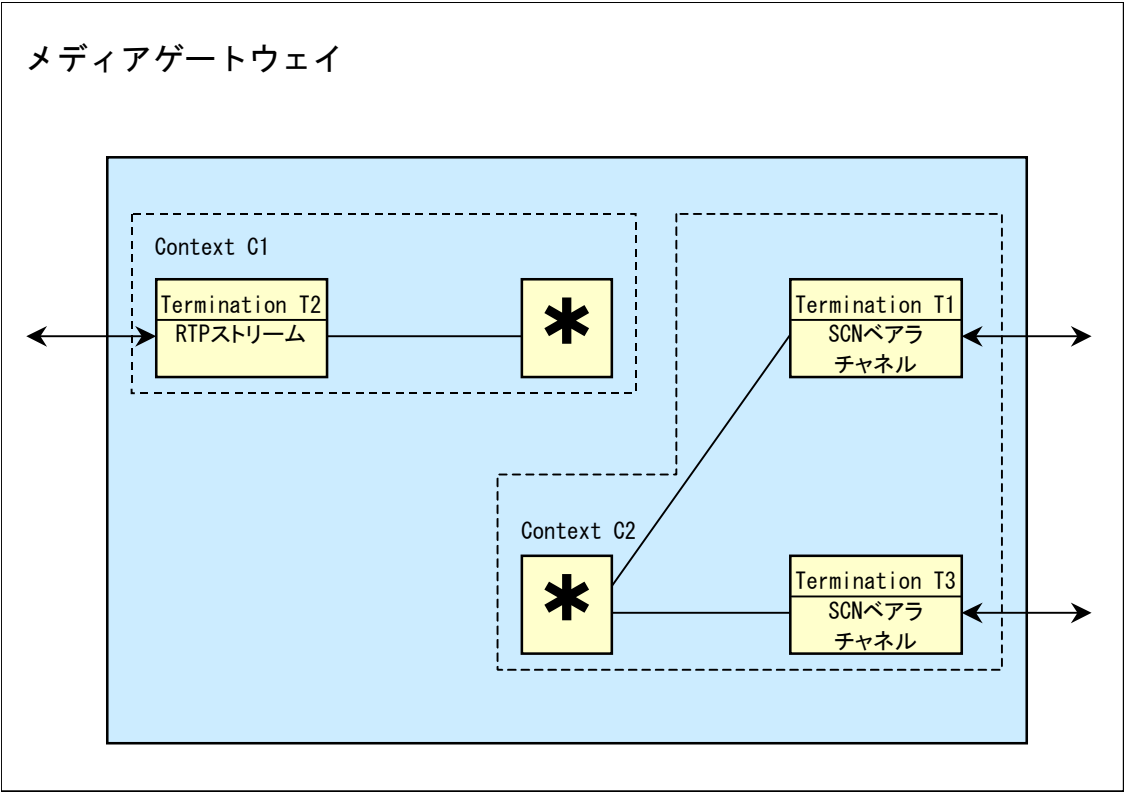


図6-3/JT-H248.1 コールウェイティングシナリオの例/T1応答  
(ITU-T H.248.1)



## 6.1 Context

Contextは複数Termination間のアソシエーションを示す。Contextは(誰が誰を見る/聞くのかの)トポロジを示す。また、アソシエーションに三つ以上のTerminationが含まれるなら、Contextは、メディア合成または交換、あるいは両方のパラメータを示す。

Null Contextと呼ばれる特殊なContextがある。これは、他のどのContextにも存在しない、それ故、他のTerminationと関連していないTerminationを含む。Null Context内のTerminationはそれらのパラメータを変更されたり、また検査されたりすることができる。また、イベントを検出されてもよい。

一般的に、AddコマンドはTerminationをContextに追加するために使用される。MGCがTerminationを追加される既存Contextを指定していないのなら、MGは新しいContextを生成する。Terminationは、Subtractコマンドによって、Contextから削除されてもよい。また、Terminationは、Moveコマンドで、あるContextから他のContextへ移動されてもよい。あるTerminationは、ある時点においては、ただ一つのContextのみに存在していなければならない。

Context内における最大Termination数は、MGプロパティによる。一対一接続のみを提供するMGは、Context内に最大二つのTerminationが存在することを許容する。多地点会議をサポートするMGは、Context内に三つ以上のTerminationが存在することを許容する。

### 6.1.1 Contextの属性とディスクリプタ

Contextの属性は、以下のものがある。

- Context ID
- (誰が誰を見る/聞くのかの)トポロジ ディスクリプタ  
Contextのトポロジは、Context内のTermination間のメディアフローを表す。一方、Terminationのモードプロパティ(“SendOnly”、“RecvOnly”、他)は、MGの出口、若しくは入口でのメディアフローを表す。
- PriorityはContext処理の優先順位をMGに通知するために使用される。同時にたくさんのContextが処理されなければならないといった状況(例えば、リスタート)において、MGCがMGにおける自律的なトラフィックの優先順位を制御するために、MGCがPriorityを使用することができる。Priorityは、0が最低、15が最高である。
- 緊急呼を表す指標が、MGにおける優先処理を許容するために提供される。
- IEPS呼を表す指標は、E.106の機能と技術の達成を可能とするために提供される。
- Context属性記述子(ディスクリプタ)は、パッケージ拡張機構(メカニズム)を使用することにより付加的なcontext記述子の定義を可能とする。(7.1.19節参照)

### 6.1.2 Contextの生成、削除、変更

本プロトコルは、(暗黙的に)Contextを生成し、あるいは、既存Contextのパラメータ値を変更するために使用することができる。本プロトコルは、TerminationをContextに追加するためのコマンド、ContextからTerminationを削除するためのコマンド、Context間でTerminationを移動させるためのコマンドを持つ。Contextは、最後に残されたTerminationが除去されたとき、あるいは移動されたときに、暗黙的に削除される。

## 6.2 Termination

Terminationは、メディアおよび制御ストリームまたはそのいずれかを、送信または受信または送受信するための、MG上の論理エンティティである。Terminationは複数のプロパティによって表される。プロパティは、コマンドに含まれるディスクリプタの組に分類される。Terminationはユニークな識別子(TerminationID)を持ち、TerminationIDは、Termination生成時、MGによって割り当てられる。

物理エンティティを表すTerminationは、半永久的に存在する。例えば、TDMチャンネルを表すTerminationは、

ゲートウェイでそれが提供される限り存在する。RTPフローのようなある短期間だけ存在する(ephemeral)情報フローを表すTerminationは、それが使用されている間だけ存在する。

Ephemeral TerminationはAddコマンドによって生成され、Subtractコマンドによって削除される。対照的に、物理Terminationが追加されたり、Contextから削除されたりする時は、Terminationはそれぞれ、Null Contextから持ってこられ、Null Contextに戻される。

Terminationはそれらに適用されるシグナルを持ってもよい(節7.1.11参照)。Terminationは、MGCに対する通知メッセージのきっかけ、またはMG(自身)の動作のきっかけとなるイベントを検出するようにプログラムされてもよい。統計情報をTermination上で蓄積してもよい。統計情報は要求(AuditValueコマンドによる。節7.2.5参照)に従ってMGCに報告され、また、Terminationが終了するとき、若しくはSubtractコマンドの目的によりNULL Contextに返されるときに報告される。

マルチメディアゲートウェイは多重化されたメディアストリームを処理してもよい。例えば、ITU-T勧告H.221は複数のデジタル64kbit/sチャンネル上で多重化された複数のメディアストリームのためのフレーム構造を記述している。コネクションモデルにおいて、このようなケースは以下のように処理される。多重化されたストリームの一部を運ぶ全てのベアラチャンネル毎に、物理、もしくは短期間だけ存在する(ephemeral)「ベアラTermination」が存在する。デジタルチャンネルの送受信を行うベアラTerminationは、“Multiplexing Termination”と呼ばれる分離されたTerminationに接続される。Multiplexing Terminationは、構造が決められたセッションを示す、Ephemeral Terminationである。このTerminationのMultiplexDescriptorが、使用される多重化法を記述し(例えば、H.320セッションのためのH.221)、収容されたデジタルチャンネルがフレームの中で集められるように指示する。

Multiplexing Terminationはカスケードでもよい(例えば、デジタルチャンネルのH.226 多重化が、H.324 セッションをサポートするH.223 多重化に流れ込むような)。

セッションの中で運ばれる個々のメディアストリームは、Multiplexing Termination上でStreamDescriptorによって記述される。これらのメディアストリームは、Multiplexing TerminationをサポートするベアラTerminationよりも、むしろContext中のTerminationにより、送受信されるストリームと関連付けられる。各ベアラTerminationは単独のデータストリームのみをサポートする。これらのデータストリームは、Multiplexing Termination上のストリームとしては明確には現れず、Contextのその他の部分からは隠される。

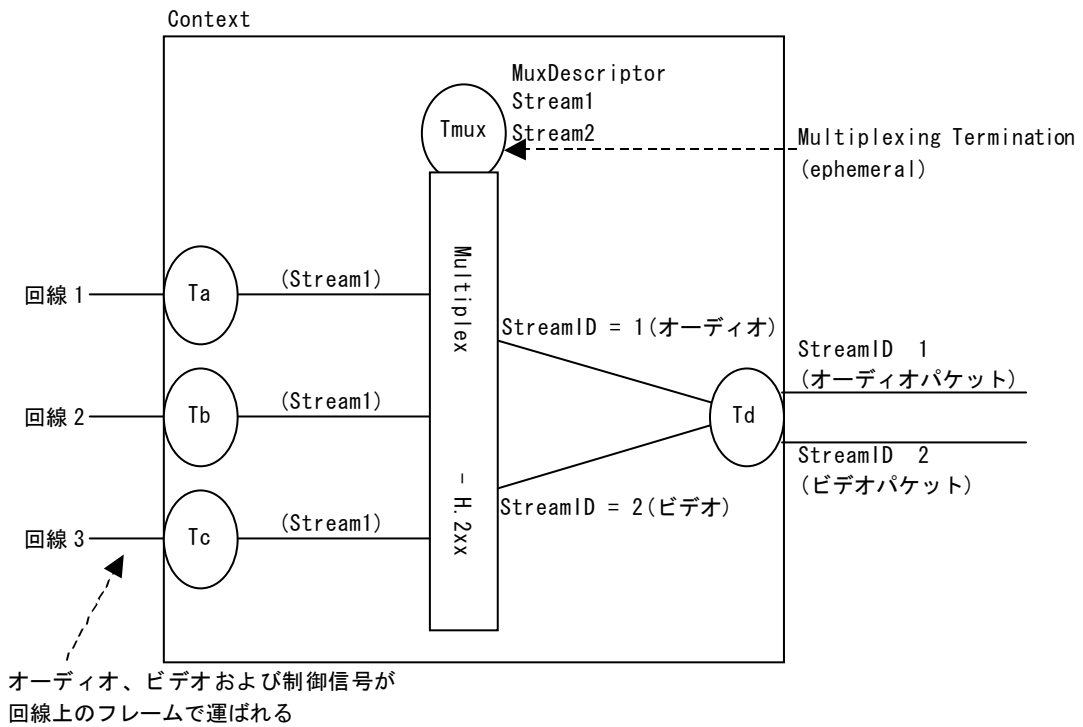


図6-4/JT-H248.1 多重化されたTermination – 回線網からパケット網へ (ITU-T H.248.1)

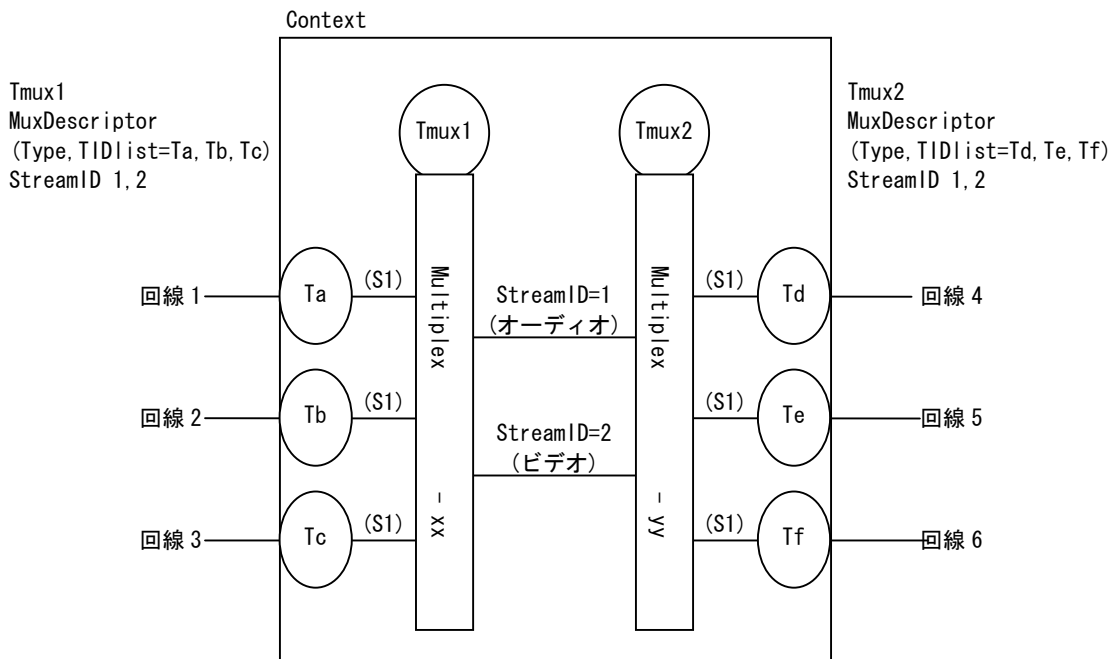


図6-5/JT-H248.1 多重化されたTermination – 回線網から回線網へ (ITU-T H.248.1)

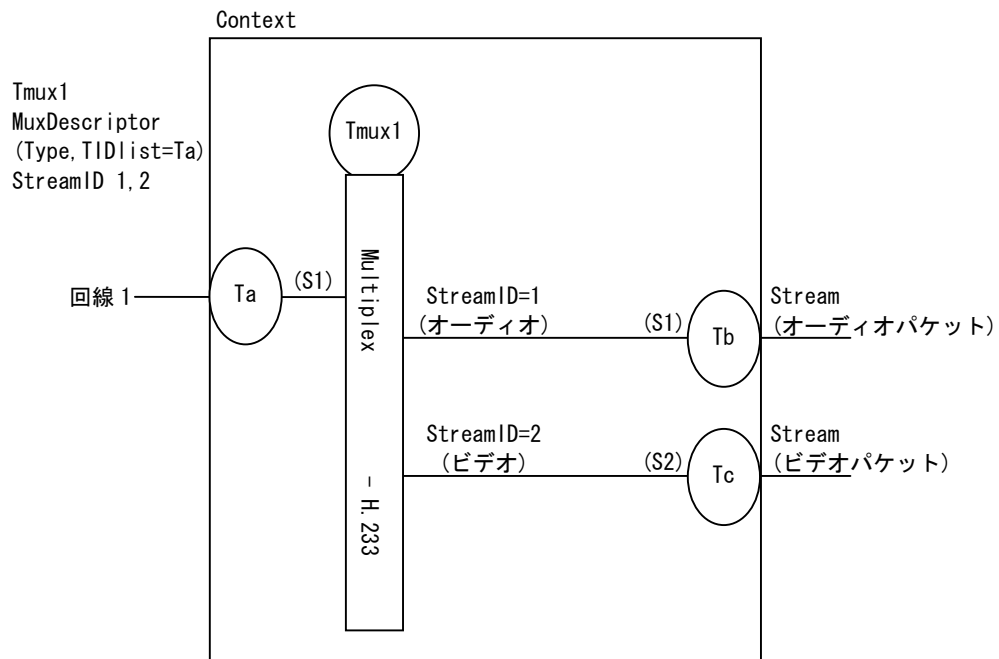


図6-6/JT-H248.1 多重化されたTermination – 一对多  
(ITU-T H.248.1)

前の段落で記述されているMultiplexing Terminationと異なり、多重化されたベアラTermination(ATM AAL TYPE2ベアラのような多重化されたベアラ)は、メディアストリームを運ばない。それらは、実際のベアラの生成と削除のモデルとしての目的のために厳密に表される。新たな多重化されたベアラが生成されるとき、Ephemeral Terminationがこの目的のために作られたContext中に生成される。このTerminationが削除されるとき、多重化されたベアラは削除される。

### 6.2.1 Termination変更(Dynamics)

このプロトコルは、新しいTerminationを生成したり、既存Terminationのプロパティ値を変更したりするために使用することができる。これらの変更は、イベントやシグナルの追加や削除の可能性を含む。Terminationのプロパティ、イベントやシグナルは、次の章で記述される。MGCは、Terminationおよびリソースを開放もしくは変更することができる。NULL Context内に存在するリソース、若しくはAddコマンドなどを通して既に占有されているリソースがterminationにより表される。

### 6.2.2 TerminationID

TerminationはMGによって選択された任意のスキーマであるTerminationIDによって識別される。

物理TerminationのTerminationIDはMGで提供される。TerminationIDは構造を持つように選択されてもよい。例えば、TerminationIDは、トランク群およびトランク群の中のトランクで構成されてもよい。

二種類のワイルドカードを使用するワイルドカードメカニズムが、TerminationIDに対して使用することができる。その二種類のワイルドカードとは、"ALL"と"CHOOSE"である。前者は、複数のTerminationを一度に指すために使用される。後者は、部分的に指定されたTerminationIDを満足する一つのTerminationを選択しなければならないことをMGに指示するために使用される。これは、例えば、MGCがMGにトランク群の中のある回線を選択するように指示することを可能にする。

また、TerminationIDはリストで指定することが可能である。TerminationIDListは階層的TerminationID構造が不可能なケースや各TerminationIDのために個々のコマンドを送ることが望まれていないケースにおいて、その使用が推奨される。

### 6.2.3 パッケージ

異なる種類のゲートウェイは、異なる特性を持つTerminationを実装してよい。TerminationがMGに実装されたオプションのプロパティ、イベント、シグナル、統計情報(Statistics)を持つことを許容することで、様々なTerminationがこのプロトコルに適合する。

MG/MGCの相互接続性を達成するために、このようなオプションはパッケージに分類される。そして、一般的にTerminationはこのようなパッケージのセットで実現される。パッケージに関する詳細情報は、12章に記述される。MGCは、どのパッケージを実現しているかを知るためにTerminationを監査(Audit)することができる。

パッケージの中のプロパティ、イベント、シグナル、統計情報は、それらのパラメータと同様、識別子(ID)で識別され、識別子は検索される。各パッケージに関して、PropertyID、EventID、SignalID、StatisticsIDおよびParameterIDは、固有のネームスペースを持ち、そして、同じ識別子がそれぞれに使用されてもよい。また、異なるパッケージの二つのPropertyIDは同じ識別子を持ってよい。

特殊なパッケージをサポートするために、MGは、パッケージの中で定義される全てのプロパティ、イベント、シグナル、統計情報を認識しなければならない。MGは、全てのシグナルとイベントのパラメータもサポートしなければならない。もしこれらプロパティ、シグナル、イベント、及び統計情報の背後にある機能が実装されない場合は、MGはシンタックスエラーや未定義IDエラーを返すべきではなく、501 “Not implemented” エラーを返すべきである。MGは、ある特定のプロパティやパラメータに対するパッケージの中に挙げられる値の一部をサポートしてもよい。もしサポートされていない値がMGCで定義されるのであれば、MGは501 “Not implemented” エラーを返すべきである。

パッケージを拡張するとき、元のパッケージ中で定義されたプロパティ、イベント、シグナル、統計情報は、拡張されたパッケージの名前か、もしくは元のパッケージの名前のどちらかを用いて記述することができる。例えば、もしパッケージAがイベントe1を定義し、パッケージBがパッケージAを拡張すると、B/e1は、パッケージBを実装したTerminationに関するイベントである。定義によれば、MGは元のパッケージを実装しなければならないが、許容されるインタフェースとして元のパッケージを公表(publish)することは任意である。もしパッケージAが公表されると、パッケージAはパッケージBと同様にAuditValue内のパッケージディスクリプタ上で報告され、A/e1をTermination上で利用してもよくなる。もしMGがパッケージAを公表しない場合、B/e1のみが利用可能となる。AuditValueにて公表されると、A/e1とB/e1は同じイベントとなる。

相互接続性の改善とバックワード互換の目的で、MGはTerminationによってサポートされる全てのパッケージ(拡張されたパッケージの元となったパッケージを含む)を公表してもよい。ただし、元のパッケージが“拡張のためだけにデザインされた”として特別に定義される場合だけは例外である。

### 6.2.4 Terminationのプロパティとディスクリプタ

Terminationはプロパティを持ち、プロパティは固有のPropertyIDを持つ。ほとんどのプロパティはデフォルト値を持ち、それらは、本標準ないしパッケージ(12章参照)で明示的に定義されているか、あるいは供給時に設定される。もし、値が提供されていないのなら、TerminationStateとLocalControlを除く全てのディスクリプタのプロパティは、Terminationが最初に生成されたときとNull Contextに戻されるときに、emptyないし“no value”に初期化される。Terminationが最初に生成される時、若しくはNULL Contextにもどされる時に、この状態は“idle”ライン、トランク、またはエンティティを表す。二つの例外のデフォルト値は、節7.1.5と節7.1.7に記述される。

MGにおけるプロパティ値の設定は、本標準もしくはパッケージで供給されるいかなるデフォルト値より優先される。したがって、もしMGCがTerminationのプロパティ値を完全に制御することが重要であれば、TerminationをContextに「加える」場合には明確な値を示すべきである。あるいはまた、物理Terminationに対してMGCは、Null Context中にあるTerminationを監査することで、設定されたプロパティ値を決定すること

ができる。

たくさんのTermination用共通プロパティ(common properties)や、メディアストリーム固有プロパティが存在する。共通プロパティは、Termination Stateプロパティとも呼ばれる。それぞれのメディアストリームに関して、Localプロパティや送受信フローのプロパティが存在する。

ベースプロトコルに含まれないプロパティは、パッケージで定義される。これらのプロパティは、PackageNameとPropertyIDで構成される名前によって識別される。ほとんどのプロパティはPackageDescriptionに記述されているデフォルト値をもつ。プロパティはread-onlyないしはread/writeでもよい。プロパティのpossible valueは、これらの現在の値にできるとして、監査されてもよい。read/writeの能力のあるプロパティのために、MGCはプロパティの値を設定することができる。プロパティは、パッケージを実現しているすべてのTerminationによって共有される単一の値をもつ"Global"として宣言されてもよい。便宜上、関連するプロパティはディスクリプタに分類される。

TerminationがContextに追加されるとき、そのTerminationのread/write プロパティの値はAddコマンドのパラメータとして適切なディスクリプタを含むことで設定できる。同様に、Context中のTerminationのプロパティは、Modifyコマンドによってその値が変更されてもよい。また、Moveコマンドの結果として、あるContextから他のContextにTerminationが移動したときにも、プロパティはその値を変更させてもよい。ときには、ディスクリプタがコマンドの出力値として返される。

同じcontext内で異なるtermination上にプロパティを設定することは、暗黙的にMGに対して確実な機能を実行することを指示する。例：もしG.711コーデックがTermination Aに設定され、G.729コーデックがTermination Bに設定されているとした場合には、MGはその2つのTermination間においてメディアフローが可能となると（すなわち、Mode Propertyが各terminationにて”Inactive”以外の状態に設定されることにより）、直ちにトランスコーディング機能が起動される。

注記：MGリソースの不必要な起動を避けるために、そのプロパティが決定しているストリームに使用されるまでの間、MGCは与えられたtermination及びストリーム用としてMode Propertyを”Inactive”状態に設定すべきである。

一般的に、ディスクリプタが前述のコマンドのうちの一つから完全に省略される場合、そのディスクリプタ中のプロパティはコマンドが作用するTerminationに対するそれらの以前の値を保持する。他方では、いくつかのread/writeプロパティがコマンド中のディスクリプタから省略される場合（つまり、ディスクリプタが、単に部分的に指定される場合）、パッケージが他の振る舞いを規定していなければ、それらのプロパティは、コマンドが作用するTerminationについてそのデフォルト値にリセットされることになる。個々のディスクリプタの詳細は節7.1を参照。

上記振る舞いはシグナル、イベント、及びそれらそれぞれのパラメータに等しく適応される。イベント デスクリプタの供給(プロビジョニング)は使用するRequestID及び供給（プロビジョニング）に含まれるイベントのどんなパラメータでも含む。イベントデスクリプタに関してのMGにおける供給(プロビジョニング)は全て、MGからのNotify コマンドに対するエラーレスポンスを避けるためにMGCにコピーされるべきである。

以下の表は、考えられているすべてのディスクリプタとその使用法を一覧に示している。なお、全てのディスクリプタがすべてのコマンドのインプットあるいはアウトプットとして使用されるわけではない。

表6-1/JT-H248.1 ディスクリプター一覧

(ITU-T H.248.1)

ディスクリプタ名	説明
Modem	適用できるときにModem typeやプロパティを識別する(*)
Mux	マルチメディアTerminationのための多重化タイプ(H. 221, H. 223, H. 225. 0など)を表す。また、input muxを形作るTerminationを表す。

Media	メディアストリームのリスト(節7.1.4を参照)
TerminationState	ストリームに特化しない(パッケージの中で定義することのできる)Terminationのプロパティ
Stream	単一のストリームのためのRemote/Local/LocalControl ディスクリプタのリスト
Local	MGがremoteエンティティから受信するメディアフローを示すプロパティを含む
Remote	MGがremoteエンティティへ送信するメディアフローを示すプロパティを含む
LocalControl	MGとMGCの間に関連するプロパティ(パッケージの中で定義される)を含む
Events	MGによって検知されるイベントおよびイベントが検知されたときにすることを記述
EventBuffer	Event Bufferingがアクティブなときに、MGによって検知されるイベントを表す
Signals	Terminationに適用されるシグナル(7.1.1参照)を表す
Audit	Auditコマンドにおいて、どの情報が要求されるかを示す
Package	AuditValueにおいて、Terminationによって実現されているパッケージのリストを返す
DigitMap	イベントが個々に通知されるよりもグループとして通知されるように、マッチしていなければならないイベントのセットの順番のパターンを記述する。
ServiceChange	ServiceChange コマンドにおいて、サービス変更の目的・理由など
ObservedEvents	Notifyないし、AuditValueにおいて、検知されたイベントの報告
Statistics	SubtractおよびAuditにおいて、Termination、又はストリームに保持されている統計情報(Statistics)を報告
Topology	Contextの中でTermination間のフローの方向を示す
ContextAttribute	全体でcontextに影響する(パッケージ内で定義可能な)プロパティを含む
Error	エラーコードとオプションとしてエラーテキストを含む; コマンド 応答や Notify 要求に現れてもよい

(\*)ModemDescriptorはH.248.1 version 2(2002/05)では賛成を得られなかった。

### 6.2.5 Root Termination

まれに、ゲートウェイ内のTerminationに対してではなく、ゲートウェイ自体にコマンドを適用しなければならないことがある。この目的のために特殊なTerminationID "Root"が予約されている。パッケージはRoot上で定義されてもよい。このため、Rootは、プロパティ、イベント、シグナルや統計情報を持ってよい。したがって、RootのTerminationIDは以下に現れてもよい。

- Modify コマンド プロパティの変更、シグナルの送信、イベントのセットのため
- Notify コマンド イベント報告のため
- AuditValue の返信 プロパティの値やRoot上に実装された統計情報を検査するため
- AuditCapability コマンド Rootのどのプロパティが実装されるかを決定するため
- ServiceChange 全体のゲートウェイがサービス中であるか否かを申告するため

RootのTerminationIDの他での使用はエラーとなる。その場合、エラーコード410(不正識別子: Incorrect identifier)が返送されなければならない。

### 6.3 ワイルドカード指定原則 (Wildcarding Principles)

この節では全てのコマンドに適用されるContextIDとTerminationIDのワイルドカード指定における振る舞いを規定する。これらのコマンドを処理する際に2つのワイルドカード処理の形式を考えなければならない。

#### 1. Context ワイルドカード処理

## 2. Termination ワイルドカード処理

ワイルドカード指定手順の目的として、1つ以上のTerminationIDからなるTerminationIDListはワイルドカード指定されたTerminationIDと考えられる。ワイルドカード指定されたcontextと任意にワイルドカード指定されたterminationを含むトランザクションを実行する際には、トランザクション内の全てのコマンドがContextIDの特定のインスタンスにおいて、次ContextIDインスタンスに移行する前に、順番に実行される。トランザクション内において多数のコマンドが存在するケースにおいては、最初のコマンドで指定されるTerminationID (ワイルドカード指定、若しくは特定の) が特定のContextIDのインスタンスにマッチするときのみ、そのトランザクション内における次コマンドが実行される。もしそのトランザクションにおいて次コマンドのTerminationID (ワイルドカード指定、若しくは特定の) が特定のContextIDインスタンスにマッチしない場合は、431 (“No TerminationID matched a wildcard”) エラーコードが返信され、エラーが出力されたコマンドがオプションとして示されない限り、ワイルドカードContextIDの次インスタンス処理は止められる。特定のワイルドカードの組み合わせにおける実行について以下で議論する。

### 6.3.1 特定のContextIDにおけるTerminationIDのワイルドカード指定

ある特定のContextIDにおいて、あるコマンドのTerminationIDにALLを使用したときの効果は、マッチするTerminationIDの各々に対してそのコマンドを繰り返すことと等しい。ALLの使用はRoot terminationをアドレス指定しない。これらコマンドの各々がレスポンスを生成かもしれないため、全体のレスポンスのサイズは大きくなるかもしれない。このように、そのContextでワイルドカードが1つ以上のTerminationIDにマッチする場合には、すべての可能なマッチ処理が試みられ、それぞれから個々に結果が返送される。ワイルドカード指定されたTerminationIDによって参照されるTerminationが特定のcontextに存在しない場合は、431 (“No TerminationID matched a wildcard”(ワイルドカードに相当するTerminationIDなし)) エラーコードが返される。指定されたcontextにないワイルドカード指定されたTerminationIDによって特定された個々のterminationに対してはエラーは返されない。

例：ある1つのゲートウェイに4つのterminationがあると仮定する。t1/1、t1/2、t2/1、およびt2/2。またそのゲートウェイのContext 1にはt1/1とt2/1があり、Context 2にはt1/2とt2/2があると仮定する。

コマンド：

```
Context=1 {Command=t1/* {Descriptor/s}}
```

返り：

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

### 6.3.2 ワイルドカード指定(ALL)されたContextID における特定のTerminationID

ContextIDがワイルドカード指定される場合(すなわち、ContextID=ALL)で、且つTerminationIDが完全に指定される場合には、その効果は、1つ以上のマッチするTerminationIDを含む各non-NULL contextに対して、ワイルドカード指定にマッチするそれぞれのTerminationIDでコマンドを繰り返すことと等しい。したがって、検索によりcontextが見つけれられる必要があり、唯一のコマンドインスタンスだけが実行される。その特別なterminationを含まないcontextに対してエラーは通知されない。そのterminationがどのような(non-NULL)contextにも含まれないならばエラーコード 431 (“No TerminationID matched a wildcard”) (ワイルドカード指定にマッチするTerminationIDなし) が返される。NULL以外のcontextが存在しない場合にはエラーコード411 (“The transaction refers to an unknown ContextID”) (トランザクションは未定義ContextIDに関連する) が返される。例えばMGとMGC間の状態不一致を修正する場合においてはContextIDを定義する形式よりむしろこのような動作を使うことの方が反対ではあるが役立つかもしれない。

例：上記ゲートウェイ構成を取っている場合には

コマンド：

```
Context=* {Command=t1/1 {Descriptor/s}}
```



返り :

```
Context=1 {Commnad=t1/1 {Descriptor/s}}
```

### 6.3.3 ワイルドカード指定(ALL)されたContextIDにおけるTerminationIDのワイルドカード指定

ContextIDがワイルドカードの場合(すなわち、ContextID=ALL)で、且つTerminationIDがワイルドカード指定の場合には、その効果は、1つ以上のマッチするTerminationIDを含む各non-NULL contextに対して、ワイルドカードにマッチするそれぞれのTerminationIDでコマンドを繰り返すことと等しい。したがって、そのワイルドカードが特定のワイルドカードContextIDのインスタンスで1つ以上のTerminationIDにマッチするならば、結果がそれぞれのために報告されている状態で、すべての可能なマッチ処理が試みられる。ワイルドカードTerminationIDにマッチしないterminationを含まないcontextに対してはエラーが報告されることはない。特定のワイルドカードContextIDのインスタンス内には存在しないワイルドカードTerminationIDによって定義される個々のterminationに対してはエラーが報告されることはない。もしワイルドカードContextID、及びTerminationIDのどちらにもマッチしない場合は、431 (“No TerminationID matched a wildcard”) エラーコードが返される。

例：上記ゲートウェイ構成を取っている場合には  
コマンド：

```
Context=* {Commnad=t1/* {Descriptor/s}}
```

返り :

```
Context=1 {Commnad=t1/1 {Descriptor/s}}
```

```
Context=2 {Commnad=t1/2 {Descriptor/s}}
```

複数のコマンドがワイルドカードTerminationID、且つ/又はワイルドカードContextIDに含まれるケースにおいては、最初のコマンドが最初のContextIDおよびTerminationIDインスタンスにマッチしない場合には、そのリクエストにおける次コマンドはそのインスタンスに対しては実行されない。

### 6.3.4 ワイルドカードの応答

個々のレスポンスは必要でない場合は、ワイルドカード応答が要求されるかもしれない。このような場合には、単一の応答が生成される。その応答は全ての個々の応答のUNIONで構成されるか、さもなければ、その応答は値のコピーを抑止しされながら生成される。例えば、プロパティ p1=a、p2=bでTermination Taが与えられ、プロパティ p2=c、p3=dでTermination Tbが与えられ、UNIONレスポンスはワイルドカードTerminationIDとプロパティ p1=a、p2=b、c、およびp3=dの順序から成る。ワイルドカードレスポンスはAuditコマンドで特に役に立つかもしれない。ワイルドカードcontextに関連して、ワイルドカードUNIONレスポンスを使用するならば、TerminationIDによって参照をつけられているすべての個々のtermination/sのUNIONにて単一のレスポンスが送られる。そのレスポンスはContext=ALL、ワイルドカードTerminationID、およびプロパティの順序を含んでいる。

あるエラーがワイルドカードレスポンスを定義するワイルドカードリクエストの実行中に発生した場合には、まだあまり大きくないエラーサイズが維持されている間は、そのエラーに関する役立つ情報を提供する為に特別な操作が要求される。ワイルドカードレスポンスが要求される時、たとえ1つ以上のエラー結果となったとしても、そのコマンドの全てのインスタンス(以下に定義される)が実行されるものとするが、そのトランザクションにおけるその後のコマンドは(オプションとして定義されていない限り)実行されない。多数(マルチプル)コマンドレスポンスはエラーとなったコマンドに対して返されることとする。初期コマンドレスポンスはそれらの成功したコマンドに対してのレスポンスから成るUNIONを含むノーマルワイルドカードレスポンスであることとする。それらのいずれも成功しない場合にはUNIONは空とする。失敗となった各TransactionIDに対する付加コマンドレスポンスは適切なエラーデスク립タで返されることとする。

る。

例：

コマンド：

```
Context={Commnad=t1/*{Descriptor/s}}
```

エラーレスポンス：

```
Context={Commnad=t1/*{Union response descriptors}},
```

本ワイルドカードメカニズムのコード化については付属資料AおよびBで詳細に規定する。

## 7. コマンド

本プロトコルは、コネクションモデルの論理エンティティ、すなわち、ContextとTerminationを操作するためのコマンドを提供する。コマンドは、このプロトコルによってサポートされている最も細かいレベルでの制御を提供する。例えば、コマンドはTerminationをContextに追加するため、Terminationを変更するため、ContextからTerminationを削除するため、ContextやTerminationのプロパティを監査するために存在している。コマンドはContextやTerminationのプロパティの完全な制御を供給する。これは、Terminationが通知するべきイベント、Terminationに適用されるべきシグナル/アクションを指定すること、およびContextの(誰が誰を見る/聞くのかの)トポロジを特定することを含む。

ほとんどのコマンドは、コマンド開始者であるMGCがコマンド応答者であるMGを制御するために使用される。例外はServiceChangeコマンドとNotifyコマンドである。NotifyはMGからMGCに送信され、ServiceChangeはどちらのエンティティから送信されてもよい。以下にコマンドの概要を示す。詳細については、節7.2に記述してある。

1. **Add** AddコマンドはTerminationをContextに追加する。Contextに最初のTerminationを追加するとき、AddコマンドはContext生成も行う。
2. **Modify** Modifyコマンドは、Terminationのプロパティ、イベントやシグナルを変更する。
3. **Subtract** Subtractコマンドは、ContextからTerminationを切断する。また、ContextのTerminationの統計情報を返す。Contextにおける最後のTerminationに関するSubtractコマンドは、Contextを削除する。
4. **Move** Moveコマンドは、Terminationを他のContextに移動させる。
5. **AuditValue** AuditValueコマンドは、Terminationの現在のプロパティ、イベント、シグナルや統計情報の状態を返す。
6. **AuditCapabilities** AuditCapabilitiesコマンドは、MGによって許容されるTerminationのプロパティ、イベント、シグナルに関するすべての使用可能な値を返す。
7. **Notify** Notifyコマンドは、MGがMGにおけるイベントの発生をMGCに通知することを可能にする。
8. **ServiceChange** ServiceChangeコマンドは、MGがMGCに、TerminationあるいはTerminationのグループが、サービス中であるかどうかを通知することを可能にする。また、MGCにMGの再開の開始および終了を通知する。また、ServiceChangeは、MGがMGCに、MGが利用できることを通知(登録)するために使用される。MGCはServiceChange コマンドを送信することで、MGにハンドオーバーを通知してもよい。また、MGCはMGにTerminationないし、Terminationのグループのサービス開始もしくは停止を指示するために、ServiceChangeコマンドを使用してもよい。

これらのコマンドの詳細は、節7.2.1から節7.2.8に記述されている。

## 7.1 ディスクリプタ

コマンドのパラメータは、ディスクリプタと呼ばれる。ディスクリプタは名前とアイテムのリストで構成される。なお、アイテムは値を持ってよい。複数のコマンドが、共通のディスクリプタを共有している。この節は、これらのディスクリプタを列挙している。ディスクリプタは、コマンドの出力として使用されてもよい。任意のこのようなディスクリプタ内容の戻りにおいて、空のディスクリプタはリストを伴わない名前によって表される。パラメータおよびコマンドタイプ特有のパラメータの使用方法については、コマンドの節で扱う。

### 7.1.1 パラメータの指定法

コマンドパラメータは複数のディスクリプタによって表現される。一般的なディスクリプタのテキストフォーマットは、以下のものである。

```
DescriptorName=<someID>{parm=value, parm=value...}
```

パラメータは、完全指定、過多指定、従属指定のいずれのタイプでもよい。

- (1) 完全指定パラメータは、単一で厳密な値であり、コマンド開始者がコマンド応答者に指定したパラメータの使用を指示していることを示す。
- (2) CHOOSE値を使用する従属指定パラメータは、コマンド応答者がサポートしているいずれかの値を選択することを可能にする。
- (3) 過多指定パラメータは、潜在的な値(potential value)のリストをもつ。リストの順序は、コマンド開始者の選択優先順である。コマンド応答者が提供されたリストから1つの値を選択し、コマンド開始者にその値を返信する。

Auditディスクリプタ以外の要求されるディスクリプタがコマンドで特定されていないとき、Terminationのディスクリプタに設定された以前の値が、もしあるなら、維持される。Subtract以外のコマンドでは、Auditディスクリプタの欠落は、空のAuditディスクリプタと等価である。未指定パラメータに関するMGの動作は、後述の節で示すように、関連するディスクリプタとともに変化する。パラメータが従属指定ないし過多指定のときはいつでも、応答者によって選択された値を含むディスクリプタがコマンドの出力値として含まれる。

各コマンドは、そのコマンドが影響するTerminationのTerminationIDを含む。このTerminationIDはワイルドカードでもよい。コマンドのTerminationIDがワイルドカードのときは、その効果は、一致するTerminationIDにそれぞれコマンドを送信したのと同様にならなければいけない。

### 7.1.2 Modemディスクリプタ

Modemディスクリプタは、H.324やtext conversation等での使用が要求されるのなら、Modemタイプやパラメータを明示する。このディスクリプタはModemタイプ(V.18、V.22、V.22bis、V.32、V.32bis、V.34、V.90、V.91、Synchronous ISDN)を含み、拡張を考慮している。デフォルトでは、TerminationにはModemディスクリプタは存在しない。

ModemDescriptorの使用は、JT-H248.1第2版および以降の版数においては受け入れられない。このことは、ModemDescriptorは送信された内容の一部として含まれるべきでなく、またもし受信された場合は、無視されるか、もしくはインプリメンテーションのオプションで処理されなくてはならないことを意味する。ModemタイプはLocalDescriptorおよびRemoteDescriptorにおけるデータストリームの属性として指定されることになっている。

### 7.1.3 Multiplexディスクリプタ

マルチメディアコールでは、複数のメディアストリームは(多分、別々の)複数のベアラで運ばれる。Multiplexディスクリプタは、メディアとベアラを関係付ける。このディスクリプタは以下の多重化タイプを

含む。

- H.221
- H.223
- H.226
- V.76
- Nx64K
- 適当な拡張

また、順番に並んだ、多重化されたベアラを表すTerminationIDのセットを含む。

例： Mux = H. 221 { MyT3/1/2, MyT3/2/13, MyT3/3/6, MyT3/21/22 }

Nx64K多重化タイプはNx64Kサービスを実装する(例えば、Q.931 情報転送速度(Information Transfer Rate)やQ.763 通信路要求表示(Transmission Medium Requirement)で定義されるように)。Context側において、それは広帯域データの単一ストリームをサポートする。ベアラTerminationがNx64K多重化Terminationの生成の結果としてContextに暗黙に加えられる時、そのベアラTermination用のstreamDescriptorは、そのベアラTerminationの帯域が64kbpsでなくてはならないこと以外は、そのstreamDescriptorがその多重化Termination用に定義したのと同じ値をとらなくてはならない。

#### 7.1.4 Mediaディスクリプタ

Mediaディスクリプタは、すべてのメディアストリームのパラメータを表す。これらのパラメータは2つのディスクリプタで表される。これらは、Termination Stateディスクリプタ (ストリームに関係しないTerminationのプロパティを表す) と、各々が単一のメディアストリームを記述している一つ以上のStreamディスクリプタである。

ストリームはStreamIDで識別される。StreamID は1から65535の範囲の値でなければならない。StreamIDは、Contextの中の同類のストリームを関連付けるために使用される。Terminationからの複数のストリームは互いに同期しなければならない。Streamディスクリプタの中には、四つまでの補助的なディスクリプタがある。それらは、LocalControl、Local、Remote、Statisticsである。これらのディスクリプタの関係は以下のようになる。

Mediaディスクリプタ

TerminationStateディスクリプタ

Stream ディスクリプタ

LocalControlディスクリプタ

Localディスクリプタ

Remoteディスクリプタ

Statisticsディスクリプタ

便宜上、LocalControl、Local、RemoteまたはStatisticsのディスクリプタは、それらをStreamディスクリプタで囲わないままMediaディスクリプタに含まれてもよい。この場合、StreamIDは1と見なされる。

#### 7.1.5 Termination Stateディスクリプタ

Termination StateディスクリプタはServiceStatesプロパティ、EventBufferControlプロパティおよびストリーム固有ではない(パッケージで定義される)Terminationのプロパティを含む。

ServiceStatesプロパティは(ストリーム固有ではない)Terminationの全体の状態を表す。Terminationは"test"、"out of service"、"in service"のいずれかの状態をとることができる。"test"状態はTerminationがテスト中であることを示す。"out of service"状態は、Terminationが通信に使用することができないことを示す。"in service"状態は、Terminationが通常の通信に使用することができる、あるいは使用されていることを示す。"in service"

がデフォルト状態である。

プロパティに割り当てられる値は単純値(simple value)(整数(integer)/文字列(string)/列挙(enumeration))でもよいし、従属指定(underspecified)値であってもよい。二つ以上の値が提供されるなら、MGは選択を行ってもよい。

- **Alternative 値:** リストの中に複数の値が含まれ、それらの一つを選択しなければならない。
- **Ranges:** 最小値と最大値が示され、最小値以上、最大値以下の任意の値を選択しなければならない。
- **Greater Than/Less Than:** 指定された値より大きい/小さい値を選択しなければならない。
- **CHOOSE ワイルドカード:** MG はプロパティに供される値から選択する。

EventBufferControlプロパティは、Eventディスクリプタのイベントの検知の後に続くイベントをバッファに保留するか、即座に処理するかを示す。詳細は、節7.1.9を参照。

#### 7.1.5.1 Termination State プロパティ

##### 7.1.5.1.1 ServiceStates

Property Name : ServiceStates

Description : プロパティの値はそのときのTerminationのサービス状態を示す。

Type : 列挙 (enumeration)

Possible Value :

InService	Terminationがサービス中または通常に動作中である。
OutOfService	Terminationがサービス停止中または通信に利用できない。
Test	Terminationがテスト中である。

Default : InService

Defined in : Termination State

Characteristics : Read/Write

##### 7.1.5.1.2 EventBufferControl

Property Name : EventBufferControl

Description : Eventディスクリプタのイベントの検知の後に続くイベントをバッファに保留するか、即座に処理するかを示す。節7.1.9を参照。

Type : 列挙 (enumeration)

Possible Value : LockStep 節7.1.9に従って、イベントをバッファに保留して処理する。  
OFF イベントを即座に処理する。

Default : OFF

Defined in : Termination State

Characteristics : Read/Write

#### 7.1.6 Streamディスクリプタ

Streamディスクリプタは一つの両方向ストリームのパラメータを示す。これらのパラメータは、ストリームに特化したTerminationのプロパティを含むディスクリプタとLocalフローのディスクリプタおよびRemoteフローのディスクリプタの三つで表現される。Streamディスクリプタはストリームを識別するStreamIDを含む。ストリームは、ContextのTerminationにおいて新しいStreamIDを示すことによって生成される。ストリームは、以前そのストリームをサポートしていたContext内のすべてのTerminationにおいて"false"に設定されたLocalControlのReserveGroupとReserveValueをもつストリームに対して空のRemoteディスクリプタとLocalディスクリプタを設定することによって削除される。

StreamIDは、MGCとMGの間においてローカルな意味を持ち、MGCによって割り当てられる。StreamIDは、どのメディアフローが接続されているかを表すためのものである。すなわち、同じStreamIDをもつストリーム同士が接続されている。

Terminationが、あるContextから他のContextに移動するとした時、Terminationが移動してくるContextにおける結果は、移動してくるTerminationのStreamIDと同じStreamIDをもつ新しいTerminationが追加された場合と同じである。

### 7.1.7 LocalControlディスクリプタ

LocalControlディスクリプタは、Modeプロパティ、ReserveGroupプロパティ、ReserveValueプロパティおよび(パッケージで定義される)ストリーム固有のTerminationのプロパティを含む。また、MGCとMGの関係を表す。プロパティの値は、節7.1.1で記述したように指定されてもよい。

Modeプロパティが取り得る値は、send-only、receive-only、send/receive、inactive、loop-backである。" send-only "、" receive-only "、" loop-back "は、Contextの外部と関係する。例えば、Modeがsend-onlyに設定されたストリームは、受信したメディアをContextの中に送らない。Terminationでストリームが" loop-back "に設定された場合、Terminationで受け取ったメディア (Local ディスクリプタ) はterminationの送信者側 (Remote ディスクリプタ) までloop-backされることとなる。Context内の異なるTermination間で送信されるメディアはない。Loop-backされたメディアはRemoteディスクリプタ に従って送信されなくてはならない。Modeプロパティに対するデフォルト値は"Inactive"である。シグナルとイベントはModeに影響されない。LocalControlのModeプロパティはLocalディスクリプタとRemoteディスクリプタ内で指定されるいずれのモード上でも優先される。

Terminationのブール値を持つReserveプロパティ、ReserveValueとReserveGroupは、MGがLocalディスクリプタおよびRemoteディスクリプタまたはそのいずれかを受信したときに何をするかということを示す。

Reserveプロパティの値がTrueなら、MGはLocalディスクリプタおよびRemoteディスクリプタまたはそのいずれかで示されるすべての選択肢のためのリソースを予約しなければならない。MGは、リソースを予約した選択肢で応答する。もし、いずれの選択肢もサポートすることができないのなら、空のLocalディスクリプタおよび空のRemoteディスクリプタまたはそのいずれかを含む応答をMGCに返さなければならない。二つ以上の選択肢が予約されている時にメディアが流れ始めた場合、メディアパケットはいずれの選択肢で送受信されてもよく、処理されなくてはならないが、ある時点ではただ一つの選択肢だけが活性化される。

Reserveプロパティの値がFalseなら、MGはLocalディスクリプタ(存在するなら)で示される選択肢から一つを選択しなければならない。そして、MGはRemoteディスクリプタ(存在するなら)で示される選択肢から一つを選択しなければならない。もし、MGが選択した選択肢をサポートするためのリソースをまだ予約していないのなら、MGはリソースを予約しなければならない。一方、(Trueに等しいReserveValueやReserveGroupの事前の交換によって)すでにMGがリソースを予約しているのなら、すでに予約された余分なリソースを解放しなければならない。最終的に、MGは選択したRemoteディスクリプタやLocalディスクリプタの選択肢を含む応答をMGCに返さなければならない。MGが十分なリソースを予約できないのなら、MGはエラー510(リソース不足 : Insufficient resources)で応答しなければならない。

ReserveValueとReserveGroupのデフォルト値はFalseである。これら二つのReserveプロパティの詳細な説明は、節7.1.8に示される。

LocalControlディスクリプタの新しい設定は、MGにおける以前のディスクリプタの設定に完全に取って代わる。したがって、前の設定を維持したいのなら、MGCはその情報を新しい設定の中を含めなければならない。MGCが存在するディスクリプタからいくつかの情報を削除したいのなら、単にその情報が含まれていないディスクリプタを(Modifyコマンドで)再送信するだけである。

NOTE- Modeプロパティは付属資料 Aや付属資料Bのencoding (符号化) 定義にあるStreamモードとしても知られている。これらの項はH.248内で代替可能である。

#### 7.1.7.1 Local Controlプロパティ

##### 7.1.7.1.1 Mode

Property 名 : StreamMode

Description : プロパティの値はterminationのそのときの状態を示す。

Type : 列挙 (enumeration)

Possible Value (使用可能な値) :

- Inactive Terminationはストリームに関するいずれのメディアも送らない。
- SendOnly TerminationはContextの内部から外部へのストリームに関するメディアを送る。
- RecvOnly TerminationはContextの外部から内部へのストリームに関するメディアを送る。
- SendRecv TerminationはContextの中、外の両方にあるストリームに関するメディアを送る。
- Loopback Terminationは送信者にストリームに関する受け取ったメディアをループバックする。

Default : Inactive

Defined in : Local Control

Characteristics : Read/Write

##### 7.1.7.1.2 ReserveGroup

Property 名 : ReserveGroup

Description : MGが単独のメディアグループをサポートするためのリソースを予約すべきなのか、できる限り多くのメディアグループをサポートするためのリソースを予約すべきなのかをLocalディスクリプタとRemoteディスクリプタの定義に従って、指定する。

Type : Boolean

Possible Value (使用可能な値) :

- True MGはLocalディスクリプタとRemoteディスクリプタで示された可能なすべてのメディアを予約することとする。
- False MGは節7.1.8で述べられているポリシーに従って、LocalディスクリプタとRemoteディスクリプタの各々から単独のメディアグループを予約することとする。

Default : False

Defined in : Local Control

Characteristics : Read/Write

##### 7.1.7.1.3 Reserve Value

Property 名 : Reserve Value

Description : MGが単独セットのプロパティ値 (例 : 単独のコーデックやそれに関係した属性) をサポートするためのリソースを予約すべきなのか、できる限り多くのセットのプロパティ値をサポートするためのリソースを予約すべきなのかをLocalディスクリプタとRemoteディスクリプタの定義に従って、指定する。節7.1.8参照。

Type : Boolean

Possible Value (使用可能な値) :

- True MGはLocalディスクリプタとRemoteディスクリプタの中で選択されたメディアグル

ープ (ReserveGroupがFalseなら) もしくは、各々のメディアグループ (ReserveGroupがTrueなら) の中で示された、出きる限り多くのセットのプロパティ値を提供するためのリソースを予約することとする。

**False** MGはLocalディスクリプタとRemoteディスクリプタの中で選択されたメディアグループ (ReserveGroupがFalseなら) もしくは、各々のメディアグループ (ReserveGroupがTrueなら) の中で示されたものの中から、単独のセットのプロパティ値を予約することとする。

Default : Inactive

Defined in : Local Control

Characteristics : Read/Write

### 7.1.8 LocalディスクリプタとRemoteディスクリプタ

MGCは、与えられたストリームとTermination用にメディアを復号化および符号化するためのMGリソースを予約して確保するためにLocalディスクリプタとRemoteディスクリプタを使用する。MGは、実際に何が準備されたかを示すために、これらのディスクリプタを応答に含める。必須のプロパティがMGCによって生成された要求に存在していないのなら、MGは追加のプロパティやそれらの値を応答に含めなければならない (例えば、MGCがペイロードタイプしか指定しない場合に、詳細なビデオ符号化パラメータを含めることで示す)。

リソースを予約して確保するためにMGに要求する際に不明確になることを避けるため、MGCは不完全指定 (例えば、CHOOSE) を使用するときと同じ位の量の情報を与えるべきである。それにより、MGが明確な選択をできるようになる。例えば、要求されるアプリケーションのタイプ (例えば、SDP符号化の場合の“メディアネーム”) が指定されずにCHOOSEを使用する際には、追加の情報が必要とされてもよい。 (例えば、SDP符号化の場合の属性)

LocalディスクリプタはMGが受信するメディアを示し、RemoteディスクリプタはMGが送信するメディアを示す。

プロトコルがテキスト符号化のとき、ディスクリプタは、SDP(RFC2327)で定義されたセッション記述から成る。MGCからMGに送信されるセッション記述において、以下のRFC2327の構文の例外が許容される。

- "s=", "t=", "o="はオプションである。
- 単一パラメータ値(single parameter value)のかわりに、CHOOSEの使用が許容される。
- 単一パラメータ値の代わりに、選択肢の使用が許容される。

Streamディスクリプタは一つの両方向メディアストリームを示し、そのため一つのセッション記述は二つ以上のメディア記述("m=")を含んではならない。Streamディスクリプタは選択肢として追加のセッション記述を含んでもよい。Terminationのそれぞれのメディアストリームは全く別個のStreamディスクリプタに現れなくてはならない。複数のセッション記述が一つのディスクリプタ内に存在するとき、境界として"v="行が要求される。そうでなければ、"v="行はMGに送信されるセッション記述におけるオプションである。実装は、完全にRFC2327に一致するセッション記述を受け付けなくてはならない。プロトコルがバイナリ符号化のとき、ディスクリプタは、付属資料Cに記述されているプロパティのグループ(タグと値のペア)からなる。それぞれのグループが、セッション記述のパラメータを含んでもよい。

以下に、RemoteディスクリプタとLocalディスクリプタの意味を詳細に記述する。仕様は二つの部分からなる。最初の部分は、ディスクリプタの内容の解釈について記述する。二番目の部分は、LocalディスクリプタやRemoteディスクリプタを受信したときにMGが行わなければならないアクションを記述する。MGが行う



アクションは、LocalControlディスクリプタのReserveValueプロパティとReserveGroupプロパティの値に依存する。

LocalディスクリプタかRemoteディスクリプタまたは双方は以下の場合がある。

- ・ 未指定(すなわち、存在しない)
- ・ 空
- ・ 従属指定：プロパティの値に CHOOSE を使用する。
- ・ 完全指定
- ・ 過多指定：複数のプロパティのグループおよび多分、一つ以上のそのグループの中の複数のプロパティの値を表す。

ディスクリプタがMGCからMGに送信される時、ディスクリプタは、節7.1.1の規則に従って解釈される。なお、明確化のための以下のコメントを記述する。

- (a) 未指定 Local ディスクリプタ、未指定 Remote ディスクリプタは、必須パラメータがないと解釈される。MGはそのディスクリプタに最後に指定されたものを使用することが要求される。前に指定した値が存在しない可能性があり、この場合は、関連するディスクリプタは無視される。
- (b) MGCからのメッセージの中の空Local(Remote)ディスクリプタは、メディアフローのために予約されているあらゆるリソースを解放する要求の意味を持つ。
- (c) 複数のプロパティのグループがRemoteディスクリプタやLocalディスクリプタに存在するのなら、あるいはグループ内に複数の値が存在するのなら、優先度の順番は降順である。
- (d) MGCによって送信されたプロパティのグループの中の過多指定プロパティまたは従属指定プロパティは、MGがサポートできる一つ以上の値をMGが選択することを要求されることを示す。過多指定プロパティの場合、値のリストの優先度の順番は降順である。

上記の規則に従うため、後のアクションはLocalControlのReserveValueプロパティやReserveGroupプロパティの値に依存する。

もし、ReserveGroupがTrueであるなら、MGが今サポートすることのできる要求されたプロパティグループの選択肢のいずれかをサポートするために必要とされるリソースを予約する。ReserveValueがTrueであるなら、MGが今サポートすることのできる要求されたプロパティ値の選択肢のいずれかをサポートするために必要とされるリソースを予約する。

注記：LocalディスクリプタまたはRemoteディスクリプタが複数のプロパティのグループを含み、ReserveGroupがTrueであるなら、選択肢によってメディアストリームを符号化および復号化できるように、MGはリソースを予約することを要求される。例えば、Local ディスクリプタが二つのプロパティのグループを含むなら(ここでは一つはG.711 A-law audioを表し、もう一つがG.723.1を表す)、MGはG.711 A-lawとG.723.1のどちらで符号化された音声も復号化できるように、リソースを予約する。MGはG.711 A-lawで符号化された音声とG.723.1で符号化された音声の二つのAudioストリームを同時に復号化するためにリソースを予約する必要はない。ReserveValueの使用に関する意図と似ている。

ReserveGroupがTrueであるなら、あるいは、ReserveValueがTrueであるなら、以下の規則が適用される。

- ・ MGはMGCによって要求された選択肢とMGCがLocalディスクリプタとRemoteディスクリプタの両方で要求するリソースのすべてをサポートするために十分なリソースを持っていないのなら、MGはLocalディスクリプタとRemoteディスクリプタの選択肢を少なくとも一つをサポートするためのリソースを予約すべきである。
- ・ MGがMGCから受信したLocal(Remote)ディスクリプタの選択肢のためのリソースを1つも予約で

きないのなら、MG は応答として空の Local(Remote)ディスクリプタを返さなければならない。

- MGC が Local ディスクリプタと Remote ディスクリプタを含んでいるとき、MG の MGC へのレスポンスに、MG が予約したすべてのプロパティのグループとプロパティ値を含む Local ディスクリプタと Remote ディスクリプタを含めなければならない。リソースをまったく予約できないのなら、MG は空の Local(Remote)ディスクリプタを返さなければならない。
- LocalControl ディスクリプタの Mode プロパティが、RecvOnly、SendRecv または Loopback であるなら、MG は MGC への応答に含められた選択肢にしたがって符号化されたメディアを受信する準備をしなければならない。

ReserveGroupがFalseであり、かつReserveValueがFalseであるなら、MGは一つの選択肢に対しLocalディスクリプタおよびRemoteディスクリプタを解決するために以下の規則を適用するべきである。

- MG は、Remote において少なくとも一つの選択肢をサポートすることができる最初のを Local において選択する。
- MG が全くリソースを予約することができないのなら、エラー510(リソース不足 : Insufficient resources)を返す。
- MG は、Local および Remote でそれぞれ選択したものを返す。

LocalディスクリプタまたはRemoteディスクリプタの新しい設定は、MGの前のディスクリプタの設定に完全に取って代わる。したがって、前の設定を維持したいのなら、MGCはその情報を新しい設定の中に含めなければならない。MGCが存在するディスクリプタからいくつかの情報を削除したいのなら、単にその情報が含まれていないディスクリプタを(Modifyコマンドで)再送信するだけである。

### 7.1.9 Eventsディスクリプタ

EventsDescriptorパラメータには、RequestIdentifierとMGが検出と報告を要求されるイベントリストが含まれている。RequestIdentifierは要求とこれがトリガとして生成されるかもしれない通知(notification)の対応付けのために用いられる。例えば、要求されるイベントにはFAXトーンや導通試験(continuity test)結果、オンフックオフフックの遷移が含まれる。RequestIdentifierはEventsDescriptorが空(すなわちイベントが指定されていない)であれば省略される。

ディスクリプタの各イベントには、イベント名とオプションのstreamID、オプションのKeepActiveフラグ、オプションのNotifyBehaviourフラグ、オプションのResetEventsDescriptor、オプションのパラメータ(複数)が含まれる。イベント名はパッケージ名(ここでイベントが定義される)とeventIDからなる。ワイルドカード"ALL"を、与えられたパッケージが検出すべき全てのイベントを示すイベントIDとして用いてよい。eventIDのデフォルト値は0であり、検出されるイベントが特定のメディアストリームに関係していないことを示す。イベントはパラメータを持つことができる。これは、一つのイベント記述が多くの個別のイベントを作ることなく意味の上でバリエーションを持つことができることを可能とする。これ以外のイベントパラメータはパッケージで定義される。

もしDigitMap完了イベントがEventsDescriptorに存在するか暗黙的に含まれているならば、EventDMパラメータが関連するDigitMapの名前または値のいずれかを運ぶために使用される。詳細は節7.1.14参照。

イベントが、有効なEventsディスクリプタの内容に逆らって処理され、そのディスクリプタ内の存在がわかった("認識された(recognized)")時、MGのデフォルトの動作は、MGCにNotifyコマンドを送付することである。もしイベントが有効なDigitMapの現在のダイヤル列に取り込まれるなら、通知は延期されてもよい(節7.1.14参照)。Notifyコマンドの送信はNotifyBehaviourフラグによって影響を受けてもよい。他の動作は今後の課題である。さらに、この節の終わりに示すように、イベントの認識は、一般に、現在有効なシグナルを停止させるか、または、現在のイベント/シグナルを置き換えさせるかしてもよい。もしEventsディスクリプタが別のEventsディスクリプタによって置換えられないならば、イベントが検出されたあとは有効のままであ

る。

EventBufferControlプロパティの値がLockStepならば、以後のイベントの検出時には、イベントの通常の処理は中断される。その後検出されるEventBufferディスクリプタにある任意のイベントは、検出時刻と一緒に、EventBufferの末尾に追加される(FIFOキュー)。MGは新しいEventsDescriptorがロードされるのを待たなければならない。新しいEventsDescriptorが、新しいEventsDescriptorを持つコマンド受信の結果として、もしくは、埋め込み型EventsDescriptorを有効にすることにより、ロードされることことができる。

EventBufferControlが0ffの場合、MGは現在有効なEventsDescriptorに基づく処理を継続する。

埋め込み型EventsDescriptorが有効とされた場合に、MGは新しく有効になったEventsDescriptorによるイベントの処理を継続する。(注意：EventBufferの処理の観点から見ると、埋め込み型EventsDescriptorの有効化というのは、新しいEventsDescriptorの受け取りということと等価である。)

MGが新しいEventsDescriptorを有したコマンドを受け取った時、一つ以上のイベントが、MGのEventBufferに保持されていた場合がある。このとき、EventBufferControlの値によって、MGにパufferリングされたこのようなイベントの取り扱い方が決まる。

#### ケース 1

EventBufferControlがLockStepで、かつMGが新しいEventsDescriptorを受け取っているとき、FIFOバッファ(EventBuffer)をチェックし、以下のアクションを取る。

1. EventBufferが空の場合は、MGは新しいEventsDescriptorに基づいてイベントの検出を待つ。
2. EventBufferが空でない場合は、MGはFIFOキューの最初のイベントから処理を始める。
  - (a) キュー内のイベントが新しいEventsDescriptorの中のリストにある場合は、MGはそのイベントに従い、EventBufferから当該イベントを削除する。Notifyのタイムスタンプは、イベントが実際に検出された時刻でなければならない。それからMGは新しいEventsDescriptorを待つ。新しいEventsDescriptorを待っている間、EventBufferDescriptor内にある任意のイベントが検出されるとEventBufferに置かれる。新たなEventsDescriptorを受け取ると、イベント処理がステップ1から繰り返される。
  - (b) 該当イベントがEventsDescriptorにない場合は、MGは当該イベントを破棄してステップ1から繰り返さなければならない。

#### ケース 2

EventBufferControlが0ffで、かつMGが新しいEventsDescriptorを受け取る場合、新たなEventsDescriptorを使って新しいイベントを処理する。

MGがEventBufferControlの値を0ffに設定するように指示するコマンドを受け取った時は、EventBuffer内の全てのイベントは破棄されなければならない。

MGは、個々のイベントのレポートを不必要に遅らせない限り、幾つかのイベントのレポートを一つのTransaction内で行ってもよい。

Notifyコマンドを転送することに関する手続きについては、個別の転送に関する検討に関わる付属資料またはH.248のサブシリーズを参照のこと。

EventBufferControlのデフォルト値は0ffである。

注記： EventBufferControlのプロパティはTerminationStateDescriptorに記載されているので、MGは、EventBufferControlプロパティを変更し、EventsDescriptorを含まないコマンドを受け取ってよい。

通常、イベントを検出するとその時点で有効なシグナルは停止される。KeepActiveがイベントに記載され

ている場合は、MGはイベントを検出したTermination上のいかなる動作中シグナルも中断してはならない。

NotifyBehaviourフラグはNotifyコマンドの以下の状態を示すために使われてもよい。

- ・ 即時に送信する。(NotifyImmediate: デフォルト)
- ・ 決して送信しない。(NeverNotify)
- ・ MGCの負荷に従って、規制される。(例: 送信または抑制) (RegulatedNotify)

Notify Behaviour使用についての更なる詳細についてはH.248.1付属資料E.15を参照。Digit Mapと共に使われるNotify Behaviourは有効なDigit Mapが完了した時に発生する。Regulated Notifyはそれに関係した代わりとなる規制された埋め込み型Eventsディスクリプタおよび埋め込み型Signalsディスクリプタを持ってもよい。もし通知が規制されるならば、(例えば、suppressed) そのときこれに代わる規制された埋め込み型のディスクリプタが有効とされるべきである。Notifyが規制されない場合、そのときオリジナルの埋め込み型ディスクリプタはトリガとなる。NorifyImmediateもしくはNeverNotifyが設定される場合は、いかなるオリジナルの埋め込み型ディスクリプタもイベントの検知においてトリガとなる。

一つのイベントに対して設定されたResetEventディスクリプタフラグの遭遇の結果はそのイベントに含まれるEventsディスクリプタが埋め込み型のEventsディスクリプタかどうかによって依存する。埋め込み型のEventsディスクリプタの場合には、ResetEventsディスクリプタフラグは、terminationに対するEventsディスクリプタを、埋め込み型のディスクリプタが有効であったときよりも前の状態にリセットさせる。(例: Modifyコマンドによって明示的に設定された最後のEventsディスクリプタ、もしくはNULL Contextの中に入れられることにより、最後にリセットされたtermination以来Modifyコマンドがなかった場合のMG-provisioned Eventsディスクリプタ) 非埋め込み型Eventsディスクリプタの場合にResetEventsディスクリプタフラグは、適合していて、動作を停止されていたDigitMap完了イベントを再び有効にすることによってterminationに対する有効なEventsディスクリプタをリセットさせる。

ResetEventsディスクリプタフラグは物理terminationだけにセットされるべきである。ResetEventsディスクリプタフラグはNULL文中の中にterminationが存在するときに実行されるべきである。もし、ResetEventsディスクリプタフラグが、terminationがNULL文中から出るときに遭遇するならば、ResetEvents ディスクリプタフラグは影響を持つべきではない。

イベントは、埋め込み型Signalsディスクリプタおよび埋め込み型Eventsディスクリプタまたはそのいずれかを包含し、もし存在するならば、イベントを検出したとき、現在のSignals/Eventsディスクリプタを置き換えることができる。例えば、オフフックイベントが検出されたときにダイヤルトーンシグナルを発生させるあるいはデジットを検出したときにダイヤルトーンシグナルを止めるというような記述が可能である。MGCは、KeepActiveのマークがされていて、かつ、埋め込み型SignalsDescriptorを包含しているイベントのEventsDescriptorを送付してはならない。

埋め込みは一階層のみが許される。埋め込み型EventsDescriptorは他の埋め込み型EventsDescriptorを持つてはいけぬ。埋め込み型EventsDescriptorは埋め込み型SignalsDescriptorを持ってもよい。

MGが受信したEventsDescriptorは現在のEventsディスクリプタを置きかえる。実行中のイベント通知が終了し、新しいEventsDescriptorを含むコマンドが実行された後に検出されたイベントが新しいEventsDescriptorに従って実行されなければならない。

空のEventsディスクリプタは全てのイベント認識と報告をしない。空のEventBufferディスクリプタはEventBufferをクリアし、LockStepモードにおいて全てのイベント蓄積を行わない。報告される唯一のイベントはEventsディスクリプタが有効な間に現れるイベントである。TerminationがLockStepモードにおいて動作している間に空のEventsディスクリプタが有効にされるならば、EventBufferは直ちにクリアされる。

### 7.1.10 EventBuffer ディスクリプタ

EventBufferディスクリプタはイベントのリストを持っている。EventBufferControlがLockStepのとき、MGに検知とバッファリングを要求するパラメータも同時に保有している(節7.1.9参照)。

### 7.1.11 Signals ディスクリプタ

シグナルは、フックスイッチのようなベアラに関するシグナルだけでなく、トーンやアナウンスようなMGが生成するメディアである。もっと複雑なシグナルはメディアやベアラに関するシグナルの受信と解析を含み、条件付けされた単純なシグナルが連続したものを含んでもよい。例えば、導通試験パッケージに規定されているような受信データの折り返しを含む。シグナルはまた、将来のシグナルのためのメディアの準備を要求してもよい。

SignalsDescriptorは、MGにTerminationへ適用するように指示するシグナルの集合を含むパラメータである。SignalsDescriptorは、多くのシグナル、および/もしくは、シーケンシャルシグナルリストを持っている。SignalsDescriptorはシグナルとシーケンシャルシグナルリストを持たなくてもよい。シーケンシャルシグナルリストのサポートはオプションである。

シグナルはパッケージの中で定義される。シグナルは(シグナルを定義する)パッケージ名とSignalIDによって名前を決められなくてはならない。SignalIDにワイルドカードを使用してはならない。SignalsDescriptorに現れるシグナルはオプションのStreamIDパラメータ(デフォルトは0で、これはそのシグナルが特定のメディアストリームとは関係ないことを示している)、オプションのシグナルの型(下記参照)、オプションの継続時間、および、シグナルを定義しているパッケージ内で定義されている取り得るパラメータを持つ。一つのシグナルが意味的に異なる複数のバリエーションを持つことができる。これで、多くの個別のシグナルを作らずにすむ。

最後に、オプションのパラメータ”notifyCompletion”によって、MGCはシグナルが送出(playout)を終了したときの通知依頼を明示することができる。シグナルが終了するケースとして考えられるのは、タイムアウトする(さもなければシグナル自身が完了する)、イベントで中断される、Signalsディスクリプタが置き換わって中止される、もしくは他の理由で停止するか開始できない、等である。notifyCompletionパラメータがSignalsディスクリプタに含まれないなら、他の理由でシグナルが停止するか開始できない場合に限り、通知が生成される。通知については、その時点で有効なEventsディスクリプタにおいてシグナル完了イベント(付属資料E.1.2参照)が有効でなくてはならない。

オプションのパラメータ”RequestID”は、同じSignalIDを持つ複数のシグナルが要求されるというSignalIDの特別な事例に関係してもよい。MGCは異なるSignal Completion間で、シグナルの特別な事例に関するObserved Eventsを区別することができる。RequestIDパラメータはもしNotifyCompletionParameterが存在しないなら含まれるべきではない。

継続期間は整数値で100分の1秒で表している。

三つのタイプのシグナルがある。

- on/off — offされるまで継続する。
- timeout — offされるか、指定の時間経過まで継続する。
- brief — シグナルは、新しいSignalsディスクリプタが停止を引き起こさない限り、自分自身で停止することになる。タイムアウト値は必要ない。

Signalsディスクリプタの中でTO以外のデフォルトのシグナル型をTO型にする場合、durationパラメータが必ず提示されなくてはならない。

シグナルの型がSignalsDescriptorに記述されている場合、デフォルトのシグナルの型を上書きする(節12.1.4参照)。Durationパラメータがon/offシグナルのために記述されている場合は、Durationは無視されなくてはな

らない。

シーケンシャルシグナルリストは、シグナルリスト識別子および順番に実行される一連のシグナルを保有している。シーケンシャルシグナルリスト内にある連続したシグナルの最後の部分のみがon/offシグナルになり得るシーケンシャルシグナルリストの継続時間は、シグナルの継続時間の積算値になり、それはシグナルに対するパラメータとして特定されるシグナル内遅延のタイミングについて積算の付加分も含んでいる。

もし、シグナル内の遅延がシーケンシャルシグナルリストのないシグナルに対して特定されたり、シーケンシャルシグナルリスト内にある最後の部分であったりする場合、遅延は無効とされるべきである。また、シグナルは、シグナル内の遅延のタイミングが適用される前に、シグナルのterminationで完了していたように決められるべきである。シグナル内の遅延を持つシグナルリストの中にあるシグナルの継続時間はシグナル内の遅延タイミングを含んでいる。

同じSignalsDescriptor内の複数シグナルとシーケンシャルシグナルリストは同時に送出されなくてはならない。

シグナルはTerminationからContextの外部へ向かって処理するように定義されている(逆がパッケージに示されていなければ)。もし、シグナルの方向がシグナルディスクリプタ内で特定されるなら、それはデフォルトのシグナル方向を上書きする。方向パラメータを持つそれらのシグナルに対して、基本のプロトコル方向パラメータは、両方が特定されたときには、どのパッケージで定義された方向パラメータよりも優先される。もし、MGCが、要求に応じられないMGを含むシグナルに関する方向を特定するならば、MGはエラーコード501 (“実行されない(Not implemented)”)を返すべきである。

同じシグナルが一つのTransactionの中で、複数のTerminationのために使われている場合は、MGはこれらのシグナルを生成するために同じリソースを用いることを考慮しておくべきである。

Terminationにおけるシグナルの生成は新しいSignalsDescriptorの適用によって、もしくはTermination上のイベントの検知によって停止される(節7.1.9参照)。

新しいSignalsDescriptorは既存のSignalsDescriptorに置き換わる。置き換わるディスクリプタにないTerminationに適用されていた全てのシグナルは止められ、以下のものを除いて、新しいシグナルが適用される。置き換わるディスクリプタに存在し、KeepActiveフラグを持っているシグナルは、現在実行され完了していない場合は、継続される。置き換わるSignalsディスクリプタが、現在実行されてなくKeepActiveフラグを持っているシグナルを含む場合は、そのシグナルは無視されなくてはならない。もし、置き換わるディスクリプタに、既存のディスクリプタと同じ識別子を持つシーケンシャルシグナルリストを含む場合は、以下の通り。

- 置き換わるディスクリプタのシーケンシャルシグナルリスト内シグナルの型とシグナルの順序は無視されなくてはならない。
- 既存ディスクリプタのシーケンシャルシグナルリスト内のシグナル実行は中断されてはならない。

#### 7.1.12 Auditディスクリプタ

Auditディスクリプタはどんな情報が監査されるかを示す。また、返却されるべきディスクリプタおよび/もしくは個々のプロパティのリストを示す。Auditは、たとえそのディスクリプタがコマンドに存在しない場合や従属指定パラメータを持たない場合であっても、そのプロパティ、イベント、シグナル、統計情報の現在値を含んでいるディスクリプタを強制的に返却させるように、あらゆるコマンドの中で使ってよい。Auditディスクリプタ内で取り得るアイテムは、以下の通り。

Modem (受け入れられない、節7.1.2参照)
Mux
Events

Media
Signals
ObservedEvents
DigitMap
Statistics
Packages
EventBuffer
個別のAudit Items:
Media Properties
Events
Event Buffer
Signals, Signal Lists
DigitMaps
Statistics
Packages

どの場合でも、Auditが空の場合、ディスクリプタは返却されない。これは、統計情報の返却を抑制するため、とりわけワイルドカードを使う時、**Subtract**コマンドで有用である。

#### 7.1.13 ServiceChangeディスクリプタ

ServiceChangeDescriptorは以下のパラメータを含む。

- ServiceChangeMethod
- ServiceChangeReason
- ServiceChangeAddress
- ServiceChangeDelay
- ServiceChangeProfile
- ServiceChangeVersion
- ServiceChangeMGCIId
- TimeStamp
- ServiceChangeInfo
- Extension

節7.2.8を参照。

#### 7.1.14 DigitMapディスクリプタ

##### 7.1.14.1 DigitMapの定義、生成、変更、削除

DigitMapはTerminationで受信されたDigitイベントを検知し、報告するためにMGで用いられるダイヤリング規定である。DigitMapディスクリプタはDigitMap名とそれが指し示すDigitMap自身を含む。DigitMapはマネジメント処理によってMGに事前に読み込まれ、EventsDescriptor内の名前によって参照されてもよいし、動的に定義された後、名前によって参照されてもよい。また、実際のDigitMap自身がEventsDescriptor内で規定されてもよい。Eventsディスクリプタ内のDigitMap完了イベントについて、それぞれのディスクリプタの送信順に関らず、同じコマンド内のDigitMapディスクリプタで定義されたDigitMapに対する名前で参照することは許容される。

DigitMapDescriptorで定義されるDigitMapは、このプロトコルのどのような標準のTermination操作コマンド

でも使用できる。DigitMapは一度定義されると、そのようなコマンドでTerminationID(多分ワイルドカードを使い)により指定される全てのTerminationで使用できる。もし同じ名前を持つDigitMapが該Termination上では定義されないならば、Root Terminationにおいて定義されたDigitMapはグローバルでありMG内のどのTerminationでも使用可能である。DigitMapがDigitMapディスクリプタで動的に定義される場合、

- 新しいDigitMapは未定義の名称を指定することで生成される。値は存在しなくてはならない。
- DigitMapの値は既に定義された名称に対して新しい値を供給することによって更新される。現下にDigitMapを使用しているTerminationは旧定義を継続して使用しなくてはならない。後続のその名称を指定するEventsDescriptorは、DigitMapディスクリプタを持つコマンド内の任意のEventsDescriptorを含めて、新しいDigitMapを使用しなくてはならない。
- DigitMapは既に定義された名称に対して空の値を設定することにより削除される。現下にDigitMapを使用しているTerminationは旧定義を継続して使用しなくてはならない。

#### 7.1.14.2 DigitMapタイマ

DigitMapに従うDigitの収集は3つのタイマ、すなわちstart timer(T)、short timer(S)、そしてlong timer(L)によって区切られる。

1. start timer(T)はダイヤルされるDigitに先立って使用される。start timerが0で上書きされると(T=0)、start timerは無効化されなくてはならない。これはMGがdigitを無限に待つことを意味する。
2. もし、MGがDigitMapに定義されたDigitパターンに合致させるために少なくとも一つ以上のDigitを必要とする場合、Digit間のタイマ値はlong(L)として設定されるべきである(例えば16秒)。
3. もし、Digit列がDigitMap中のあるDigitパターンに合致しても、さらにDigitを受信することで他のパターンに合致する可能性がある場合、ただちに合致したことを報告せず、MGはshort timer(S)を適用し、さらにDigitを受信されるのを待たなくてはならない。

各タイマはDigitMapに対して調整することができるパラメータである。これらのタイマのデフォルト値はMGに事前に用意されるべきであるが、DigitMap内に指定された場合、その値の方が優先される。

#### 7.1.14.3 DigitMap構文

DigitMapの正式な文法はプロトコルにおける正式な文法についての記述(付属資料A、付属資料Bを参照)にあるDigitMap規則に記述されている。この文法によるとDigitMapは文字列もしくは文字列のリストによって定義される。リスト中のそれぞれの文字列はイベントの選択肢であり、DigitMap記号かDigitMap記号の正規表現として指定される。これらのDigitMap記号は"0"から"9"の数字と、"A"からシグナリングシステムに依存する最大値までの"K"を超えない範囲の文字で表現され、DigitMapを使用するTerminationにおけるEventディスクリプタで明示されたパッケージ内に示されたイベントと一致する。(DigitMapと記号との対応付けはDTMFやMFまたはR2のような回線対応シグナリングシステムと連携するパッケージのドキュメントで定義されている。"0"から"9"の数字は関係するシグナリングシステム内の該当するDigitイベントに対応付けられなくてはならない。文字は論理的な方法で割り当てられ、イベントの選択肢に対して範囲記法を容易なものとするべきである。)

文字"x"はワイルドカードであり、"0"から"9"の記号に相当するイベントを示す。その文字列はまた明確な範囲や、より一般的には明確な記号の集合を含み、DigitMapの位置を満たすイベントの選択肢のいずれかを示す。最後にドット記号"."はゼロもしくはそれ以上のイベントセレクタ(イベントもしくはイベント範囲、イベントの選択肢集合、ワイルドカード)の繰り返しを表す。上記第3のタイミング規則の結果として、末端のドット記号の照合を行うイベント間のタイミングはデフォルトとしてshort timerを使用する。

これらのイベント記号に加え、文字列は"S"と"L"のイベント間タイマ指定子と"Z"の継続期間修飾子を含む。"S"と"L"はMGがshort timer(S)あるいはlong timer(L)を続いて起こるイベントに使用すべきであること



をそれぞれ示し、上記で記述したタイミング規則に優先されることを示す。もし、明示的なタイミング指定子がある選択肢イベントシーケンスで有効であるとするが、他の候補となる選択肢には与えられていない場合、その明示的なタイミング指定子で設定されたタイマ値が使用されなくてはならない。もし、明示的なタイミング制御を持ったすべてのシーケンスが候補集合の中から落ちた場合、タイミングは上で与えられたデフォルト規則に戻る。もし、範囲記法の中で使用されるなら、SとLの指定子は無効とされるべきである。最終的に、もし、相反するタイミング指定子が異なる選択肢シーケンスで有効である場合、タイミングの長いものが使用されなくてはならない。

"Z"は継続時間の長いイベントを示す。所定のDigitポジションを満たすイベントを示す記号の前に位置し、その位置はイベントの継続時間がlong-durationの閾値を超えた場合のみ満足することを意味する。その閾値はMGに事前に用意されるものと仮定されるが、T、L、Sタイマと同様に、DigitMap内の指定が優先される。

もしdigit (0-9,もしくはA-K) がZ指定子に従わないなら、MGは無効な手続きとしてDigitMapを拒否すべきである。範囲記法の中で使用される時、Z指定子は即座にそれに続くdigitに唯一適用する。範囲に先立って即座に使用される時は、Z修正子は範囲内のすべてのdigitに適用する。(長い継続時間になるような範囲の中での合致を要求する。)

#### 7.1.14.4 DigitMap完了イベント

DigitMapは、それを起動したEventsディスクリプタが有効であり、かつ完了するまでの間は有効である。DigitMapは以下の場合に完了する。

- ・ タイマ満了の時
- ・ ある選択肢イベントシーケンスがすでに合致しており、かつさらなるイベントの検出を通じて他に合致する選択肢イベントシーケンスが DigitMap に存在しない時(明確な一致)
- ・ たとえさらにイベントが受信されても、DigitMap の選択肢イベントシーケンスへ完全一致させることが不可能となるようなイベントを検出したとき

完了時、DigitMapに割り当てられたイベントを供給するパッケージで定義される通り、DigitMap完了イベントが生成されなくてはならない。この時点で、そのDigitMapは無効化される。そのパッケージにおいて引き続きイベントは次のように処理される。

- ・ EventBufferControl が ON の場合、引き続き digit イベントが他のイベントと同じように処理される。
- ・ EventBufferControl が OFF の場合、有効な EventsDescriptor が変わらず、かつ個々の digit イベントがそのディスクリプタ内で有効でなければ、そのときは、digit バッファへの蓄積が開始されることになる。バッファへの蓄積は、元の DigitMap 完了イベントで指定されたバッファリング時間が満了するか、もしくは有効な EventsDescriptor が置き換えられるまで続けられることになる。

その digit バッファはダイヤル文字列の論理形式をとらなければならない。それは、(多分'Z'で始まる)DigitMap で示されている通りの digit 文字群を含んでいる。長期イベントを識別するため使用されるトーン継続時間の閾値は、最後に完了した DigitMap で使用されたものと同じでなくてはならない。

バッファリング時間は、DigitMap 完了イベント内で他に明確な設定がない限り、デフォルト値 0 (バッファへの蓄積を行わない)をとる。バッファリングタイマの満了によりバッファリングが終了した場合、digit バッファの内容は廃棄される。

新たな EventsDescriptor によりバッファリングが停止された場合、そのときもしその EventsDescriptor が前と同じパッケージから新たな DigitMap 完了イベントを含むとしたら、バッファに蓄積された digit は以下に示すように DigitMap に反して処理される。新たな DigitMap で消費されなかった蓄積済みの digit は、あたかもその map が完了した後に観測されたかのように処理される。

もしそうでなく新たな EventsDescriptor が個々の digit イベントの報告をできるとしたら、蓄積されている digit の全てのセットは即座に処理され、適応可能なイベントが報告され、そしてバッファは消去されなくてはならない。

最後に、もし新たな EventsDescriptor が、DigitMap 完了イベントも、そのパッケージに関連する個々の digit イベントの報告もできないとしたら、バッファの内容は廃棄され、バッファリングは終了する。

#### 7.1.14.5 DigitMap処理

完了するまで、引き続きイベントは以下の規則に従って処理される。

1. "current dial string"、つまり内部変数は最初は空である。選択候補イベントのシーケンスの集合は DigitMap で指定されるすべての選択肢を含む。
2. それぞれのステップにおいて、バッファに蓄積された digit が利用可能であれば、あたかもその digit イベントが丁度観測されたかのように、(long digit 修飾子'Z'を伴う場合もある)最も古い digit がバッファから取り除かれ、処理が次のステップに移る。そうでなければ、それぞれのステップにおいて、タイマは、与えられたデフォルトタイミング規則、または一つもしくはそれ以上の選択肢イベントシーケンスで指定されるタイミングを基準に、次のイベントを待機するために設定される。もし、タイマ満了し、選択肢候補集合中の一つが完全に満たされた場合、タイマ満了による完全一致を報告する。もし、タイマ満了し、選択肢候補中に、部分的にしか満足しないあるいは満足するものがない場合、タイマ満了による部分一致を報告する。いずれの場合でも、DigitMap 完了イベントが詳細なタイムアウト報告を考慮しているなら、報告されるダイヤル文字列はそれぞれに見合った'L'、'S'もしくは'T'で終了することになる。
3. もしあるイベントがタイマ満了前に検出された場合、Digit 文字列記号に割り当てられ、暫定的に current dial string の終端に付加される。イベントの継続時間(long または非 long)は現在の記号の位置が関連がある場合に限り記録される(なぜなら、少なくとも1つの候補選択肢イベントシーケンスはシーケンス中のこの位置に"Z"修飾子を含むためである)。
4. current dial string は候補となる選択肢イベントシーケンスと比較される。この位置で長い継続時間のイベントを要求するシーケンスが一致する場合、およびこの場合のみ(つまり、イベントは長い継続時間を持っており、この位置の仕様に遭遇した)、この位置で長い継続時間のイベントを指定しないいかなる選択肢イベントシーケンスも、廃棄される。また、current dial string は最新のイベントを表わす記号の前に「Z」を挿入することにより修正される。この位置で長い継続時間のイベントを要求するが、観測されたイベントと一致しないいかなるシーケンスも候補集合から廃棄される。与えられた位置で long duration イベントを指定しない選択肢イベントシーケンスが上記規則の適用後に候補として残っているなら、観測されたイベント期間は選択肢との一致を評価するのに関係のないものとして扱われる。
5. ちょうど一つの候補が残り、それが完全に一致したならば、完了イベントが明白な一致を示して、生成される。候補が残らなければ、最新のイベントが current dial string から取り除かれ、そして最新のイベントが検知される前に、前のステップからの候補のうちの一つが完全に満足したならば完全一致またそうでなければ部分一致を示して完了イベントが生成される。そのとき、現在のダイヤル文字列から取り除かれたイベントは、前の節で示した規則に従い、個別のイベントとして報告されるか、バッファに蓄積されるか、もしくは廃棄されることになる。この表現は以下のように修正される。
  - a) DigitMap 完了イベントは、取り除かれたイベントが完了イベントのパラメータとして報告されるものとして規定できる。これは digit イベントの引き続いて実施される処理とは独立して

起こる。

- b) DigitMap 完了イベントは余分の digit が廃棄されるべきであると規定できる。この場合廃棄は即座に行われる。いずれのバッファリングもしくは他の処理も引き続いて発生するイベントにのみ適用される。
6. 完了イベントがステップ 5 から報告されない場合、処理はステップ 2 に戻る。

#### 7.1.14.6 DigitMap活性化

新しいEventディスクリプタがTerminationに適用されるか、埋め込み型Eventディスクリプタが起動される場合は常に、DigitMapは起動される、また、そのEventディスクリプタはDigitMap完了イベントを含み、DigitMap完了イベントは、要求動作フィールド(requested actions field)内にeventDMフィールドを含んでいる。DigitMapの新しい起動それぞれは、current dial stringをクリアして、上記のステップ1から始まる。以前の動作によるcurrent dial stringの持っていたいかなる内容も失われる。

要求動作フィールドに一つもeventDMフィールドを含んでいないDigitMap完了イベントは、エラーと考えられる。EventsDescriptorの中のそのようなイベントを受信した場合、MGは、エラー457(シグナルもしくはイベント内のパラメータ不備 : Missing parameter in signal or event)を含むエラー応答を返さなくてはならない。

#### 7.1.14.7 DigitMapとイベントの処理の相関関係

DigitMapが起動されている間、検出動作は指定されたDigitMap完了イベントを含んでいるパッケージで定義されたすべてのイベントのために可能になる。正常なイベントの振る舞い(例えば、もしDigitMap完了イベントがKeepActiveフラグを立てなければシグナルを停止する)が、下記を除いて検知された個々のそのようなイベントに適用されつづける：

- ・ 指定された DigitMap 完了イベントを含んでいるパッケージ中の、完了イベント自身以外のイベントは独立して通知されない。そして個別に有効にならない限り、副作用はない。
- ・ 部分一致完了イベントのトリガとなるイベントは認識されず、ゆえに DigitMap 完了イベントの認識に続いて再処理されるまで副作用がない。同様にバッファに蓄積された digit イベントは認識されず、処理されない限り、副作用はない。

#### 7.1.14.8 ワイルドカード

パッケージがDigitMap完了イベントを含んでいる場合、ワイルドカード化されたItemIDとパッケージ名からなるイベント仕様でもDigitMapが有効化することに注意すべきである。また、そのイベント仕様は節 7.1.14.6に従ったeventDMフィールドを含まなくてはならない。パッケージがまたDigitイベント自身を含んでいる場合、この形式のイベント仕様は検出したときにMGCに報告すべき独立したイベントを引き起こす。

#### 7.1.14.9 例

一例として、次のような番号計画を考える：

0	地域事業者オペレータ呼出
00	長距離事業者オペレータ呼出
xxxx	内線番号(Local extension number) (1-7で始まる番号)
8xxxxxxx	地域内番号(Local number)
#xxxxxxx	オフサイト内線(Off-site extension)
*xx	サービス番号(Star services)
91xxxxxxxx	長距離番号
9011 + 最大15桁	国際番号

付属資料E(節E.6)に記述されたDTMF検出パッケージがダイヤルされた数字の収集に使用される場合、上記の番号計画は次のDigitMapの結果となる:

```
(0|00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxx|9011x.)
```

#### 7.1.15 Statistics ディスクリプタ

Statistics パラメータは、特定のContextに存在している間もしくはNULLContextの外部にある間、Terminationの状態および使用を記述する情報を提供する。統計情報はTerminationのレベルであっても、Stream ディスクリプタに関係してもよい。デフォルトでは、統計情報はTerminationのレベルで発生する。もし、Stream が統計情報をサポートできないなら、ErrorCode460 (StreamでStatisticsを設定できない) が返されるべきである。適切な所で各Termination用に保存される一組の標準の統計情報(例えば送受信されたオクテット数)がある。デフォルトでは、特定のTermination用に報告される特別な統計プロパティはTerminationによって実現されているパッケージによって決定される。また、どの統計情報が集められるべきかを示すために、ディスクリプタが使用され得る。Statisticsディスクリプタの設定は以前にセットされたStatisticsディスクリプタを上書きする。このようにして、すでに設定されているStatisticsを保つために、Statisticsディスクリプタは新しいStatisticsディスクリプタの中に含まれるべきであり、統計情報の値はリセットされるべきではない。Statistics ディスクリプタから削除されていた統計情報は、それらの値をTerminationが取り去るまで保持すべきである。しかしながら、もし特定の統計情報が継続的なStatisticsディスクリプタによって再び動作されるなら、その値はリセットされるべきである。MGCはPackageIDを持った単独の統計情報や、“ALL”のワイルドカードを持ったStatisticIDと共にStatisticsディスクリプタを発行することによってstreamやterminationですべての統計情報を再び動作してもよい。streamやterminationの与えられたパッケージの中ですべての統計情報を再び動作するために特定のPackageIDを持った単独の統計情報や、“ALL”のワイルドカードを持ったStatisticIDと共にStatisticsディスクリプタは発行される。空のディスクリプタの受領は、あらゆる統計情報は特定のTerminationのために収集されるべきではないということの意味している。

デフォルトでは、空のStatisticsディスクリプタが、どの統計情報も収集されるべきでないということを示唆していない限り、TerminationやStreamレベルの統計情報はTerminationが存在しなくなるときや、SubtractコマンドのためにNULLContextを返すときに報告される。このデフォルトの動作はまたSubtractコマンドの中に空のAuditディスクリプタを含むことによって上書きされ得る。もし、Streamが7.1.6節にしたがって削除されるなら、デフォルトではStreamレベル統計情報は報告されない。Streamの中のStatisticsディスクリプタのAuditはStreamが統計情報を収集するために削除される前に実行されなければならない。統計情報はまた、AuditValueディスクリプタコマンド、もしくはAuditディスクリプタを使用しているAdd/Move/Modifyコマンドから返されてもよい。

統計情報は蓄積であり、統計情報を報告することで既存の統計情報はリセットされない。Terminationレベルの統計情報の値は、統計情報がTermination内の全てのStreamにおかれていたかのように、重要かつ優位的な機能を持つ例えば、合計や平均のように) 値を収集した結果である。このような優位的な機能は特定の統計情報の型から独立している。特定の統計情報を定義するパッケージの中で他のものを指定しない限り、デフォルトの振る舞いは値の合計である。統計情報はTerminationが存在しなくなるときや、SubtractコマンドによるNULLContextに返されるとときにリセットされる。

#### 7.1.16 Packages ディスクリプタ

PackageDescriptorはAuditValueコマンドと共にのみ使用され、Terminationによって具現化されたパッケージのリストを返す。

#### 7.1.17 ObservedEvents ディスクリプタ

どのイベントが検知されたかをMGCに通知するためにnotifyコマンドと共にObservedEventが供給される。AuditValueコマンドと共に使用されるとき、ObservedEventsDescriptorは、通知されていないイベントをイベントバッファ中から返す。ObservedEventsは、通知のトリガとなったEventsDescriptorのRequestIdentifier、検知されたイベント、オプションとして、検知した時間、および観測されたイベントの任意のパラメータを含んでいる。検知時間は1/100秒の精度で報告される。

### 7.1.18 Topologyディスクリプタ

TopologyディスクリプタはContextの中でTermination間のフローの方向を指定するために使用される。前の節中のディスクリプタに反し、Topology ディスクリプタはTerminationの代わりにContextに適用する。Contextのデフォルトトポロジは、各Terminationからの送信が他のすべてのTerminationによって受け取られるということである。Topologyディスクリプタの実装はオプションである。TopologyディスクリプタをサポートしないMGがTopologyディスクリプタを含んだコマンドを受信した場合は、エラー”444 Unsupported or unknown descriptor (未サポートもしくは未知のディスクリプタ)”を返す。このとき、オプションで、そのErrorディスクリプタのエラーテキストに未サポートのディスクリプタ名”Topology”を含んだ文字列を含む。Topologyディスクリプタは動作においてコマンドの前に生成される。動作を適用するContextが既に存在すれば、ただ一つのTopologyディスクリプタのみを含む動作はあり得る。

Topologyディスクリプタは、(T1, T2, アソシエーション[, StreamId])形式での関連付けられたTerminationのシーケンスからなる。T1とT2は、Context内のTerminationで指定され、ALLあるいはCHOOSEワイルドカードが使用されることもある。オプションのStreamIdフィールドが使用される場合、アソシエーションは、StreamIdによってラベル付けされたT1とT2の間の特定のストリームへのみ適応される。StreamIdフィールドが省略される場合、TopologyはそのTerminationにおける全てのストリームへ適用する。アソシエーションはメディアがこれら2つのTermination間でどのように流れるかを以下のように示す。

- (T1, T2, isolate)はT2にあたる Termination でT1にあたる Termination からのメディアを受信せず、逆方向も受信しないことを意味する。
- (T1, T2, oneway)はT2にあたる Termination でT1にあたる Termination からのメディアを受信し、逆方向はないことを意味する。この場合、T1 もしくはT2 と一致する Termination が存在するような ALL ワイルドカードの使用は許容されるが、T1 およびT2 の両方と一致する Termination が存在するような ALL ワイルドカードの使用は許容されない。
- (T1, T2, onewayexternal)はT2にあたる Termination でT1にあたる Termination からのメディアを外部的に送受信し、逆方向はないことを意味する。この場合、T1 と一致する Termination が存在するような ALL ワイルドカードの使用は許容されない。
- (T1, T2, onewayboth)はT2にあたる Termination でT1にあたる Termination からのメディアを受信し、逆方向はないことを意味する。この場合、(T1 と T2/T1 もしくは T2) に一致する Termination が存在するような ALL ワイルドカードの使用は許容されない。
- (T1, T2, bothway)はT2にあたる Termination でT1にあたる Termination からのメディアを受信し、かつ逆方向もあることを意味する。この場合、T1 およびT2 の両方と一致する Termination が存在するようなワイルドカードを使用することを許される。しかしながら、両方と一致する一つの Termination がある場合、LoopBack は導入されない。

CHOOSEワイルドカードは、次の制限の下で、T1とT2の中で同様に使用されてもよい。

- Topology ディスクリプタが一部となっている動作(8章を参照)が、CHOOSE ワイルドカードが使用されている Add コマンドを含んでいる場合。
- もし CHOOSE ワイルドカードが T1 あるいは T2 に現れるとき、部分的な名前が指示されてはなら

ない。

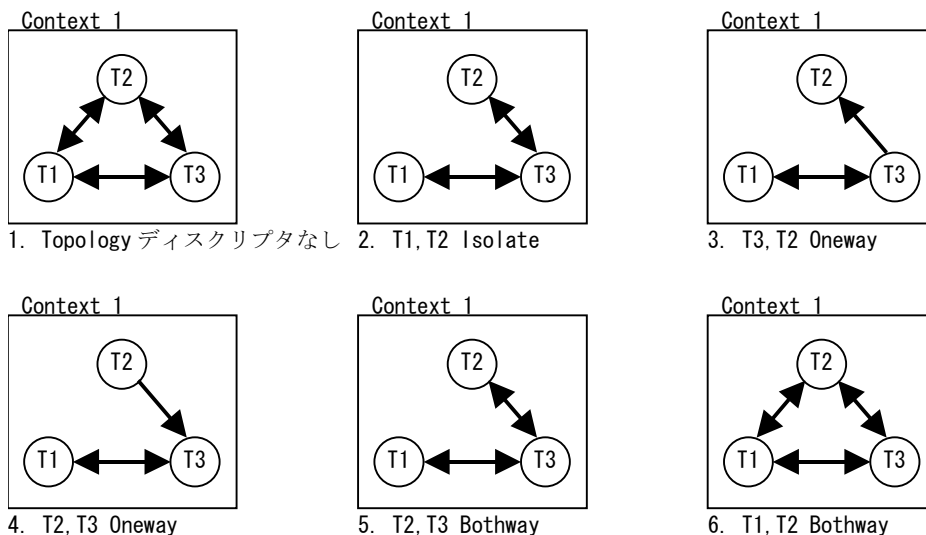
TopologyディスクリプタのCHOOSEワイルドカードは、同じ動作の中でCHOOSEワイルドカードを使用する最初のAddコマンドにおいてMGが割り当てるTerminationIDと一致する。Terminationが加えられるContextの中でT1またはT2と一致する既存のTerminationは、Topologyディスクリプタによって指定される新しく追加されたTerminationに接続される。もしTopologyディスクリプタ内でTerminationが言及されていないときには、それに関連したいかなるTopologyも変更されない。しかしながら新しいTerminationがcontextに追加される時には、Topologyディスクリプタが変更を与えない限り、そのcontextにある他のterminationとのアソシエーションはbothwayに初期化される(例として、T3が、T1およびT2を持つContextへトポロジ(T3,T1,one-way)で加えられる場合、T2への接続はbothwayになる)。

Topologyがある特定のストリーム (T1, T2, アソシエーション, StreamId)に適用されるのなら、これらのTermination間の他のストリームのTopologyは変わらない。

Topologyディスクリプタは、未定義の振る舞いを避けるため、二つのTermination (Ti, Tj)の間の、オプションであるStreamIDフィールドを持つアソシエーションと持たないアソシエーションの組合せを含んではならない。例えば、(T1, T2, bothway)と(T1, T2, isolate, S1)は同じディスクリプタに存在してはならない。このようなTopologyディスクリプタの受信時、MGはエラー421 Unknown action or illegal combination of actions (未知のActionまたはActionの不正な組合せ)を含むエラー応答を返さなくてはならない。

片方向コネクションは、Context中の他のTerminationがトポロジにおける変更が判らないような方法で実装されなくてはならない。

図7の表、およびそれに続く図8は動作中にTopologyディスクリプタを含むことの影響についての幾つかの例を示している。これらの例の中でTopologyディスクリプタはシーケンスの中で適用されること仮定されている。図9と図10はonewayexternalとonewaybothトポロジーの特別な影響を示す独立の例である。



注記：矢印の向きがフローの向きを示す

図7-1/JT-H248.1 トポロジーの例

(ITU-T H.248.1)

表7-1/JT-H248.1 トポロジーの例

(ITU-T H.248.1)

Topology	説明
1	Topology ディスクリプタなし Topology ディスクリプタがない場合、全てのTerminationは他の全てのTerminationに対し、両方向のコネクションを持つ。
2	T1, T2, Isolate T1-T2間のコネクションが取除かれる。 T3はT1およびT2の両者と両方向のコネクションを持つ。 T1とT2はT3と両方向のコネクションを持つ。
3	T3, T2, oneway T3からT2への片方向のコネクション(すなわち、T2がT3からメディアフローを受信する)。 T1-T3間の両方向のコネクション。
4	T2, T3, oneway T2-T3間の片方向のコネクション。 T1-T3間は両方向コネクションのまま。
5	T2, T3 bothway T2が両方向コネクションでT3に接続される。 この結果、2と同じ状態になる。
6	T1, T2 bothway (T2, T3 bothwayおよびT1,T3 bothwayは明示的でも暗黙的でもよい) 全てのTerminationは他の全てのTerminationに対し両方向のコネクションを持つ。

片方向コネクションは、Context中の他のTerminationがトポロジにおける変更が判らないような方法で実装されなくてはならない。

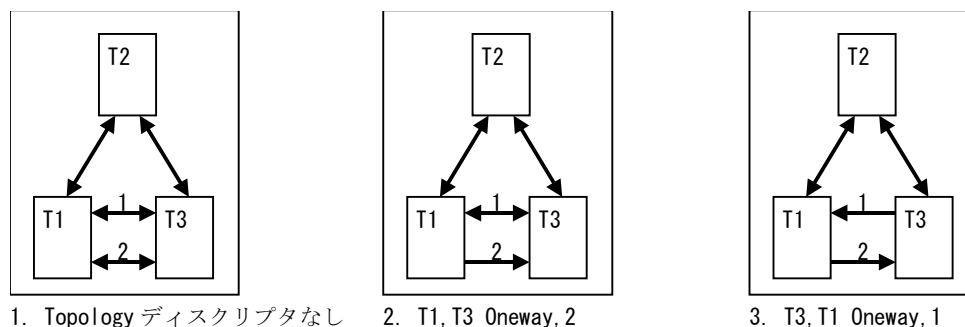


図7-2/JT-H248.1 ストリームレベルにおけるトポロジの例  
(ITU-T H.248.1)

### 7.1.19 ContextAttributeディスクリプタ

ContextAttributeディスクリプタは概してcontextに適用する（パッケージの中で定義される）プロパティおよび、terminationよりはむしろcontextで適用されるプロパティを特定するために使用される。TerminationプロパティはContextAttributeディスクリプタ内では有効ではない。ContextAttributeディスクリプタはひとつの動作の中のコマンドの前に現れる。示唆されるcontextが既に存在する場合、ContextAttributeディスクリプタのみを含む動作を持つことは可能である。Contextプロパティの値は7.1.1節の中で示されているように未定義であってもよい。

新たなContextAttributeディスクリプタの設定は、MGの中で一つ前のContextAttributeディスクリプタに完全に置き代わる。このようにして、一つ前の設定からの情報を維持するためにMGCは新しい設定の中でその情報を含まなければならない。MGCが存在するディスクリプタからいくつかの情報を削除したいのなら、単に

その情報が含まれていないディスクリプタを再送信するだけである。

AuditValueもしくはAuditCapabilitiesのコマンドしか含まない動作の中のContextAuditディスクリプタもしくはContextAttributeディスクリプタの包含は新しい設定を構成しない。

ContextIDListパラメータはContextIDのコンパクトなリストを供給するために使用される。返信の中でContextIDListパラメータの使用を要求するために、ContextIDListパラメータがそれに応じたリクエストの中で含まれるべきである。

#### 7.1.20 Errorディスクリプタ

Transaction Requestの処理中に応答者がエラーに遭遇したら、Errorディスクリプタがその応答に含まれなくてはならない。同様に、Notify要求もErrorディスクリプタを持ち得る。

一つのErrorディスクリプタは一つのIANA登録済みエラーコードから構成され、オプションで一つのエラーテキストを付加することができる。ITU-T勧告H.248.8は、有効なエラーコードとエラー記述のリストを含んでいる。

Errorディスクリプタは、記述されているエラーにとって意味的に適切で、オリジナルの要求がもつ構文解釈上の問題を最適に指摘する「最も深いレベル」で規定されなくてはならない。Errorディスクリプタはそれが現れる場所以外の構文上の構造を参照してもよい。例えばErrorディスクリプタ422 Syntax Error in Action (Action内の構文誤り)は、たとえより大きな構造体、つまりActionを参照し、それが現れる特定のコマンドを参照しなくても、そのコマンド内に現れてもよい。

### 7.2 コマンドAPI (Application Programming Interface)

本節では、H.248プロトコルのコマンドに関するアプリケーション・プログラミング・インタフェース(API)を説明する。以下のAPIに関する記述は、コマンドとそのパラメータ群を説明するためのものであり、実装に関して(例えば、ブロックするファンクションコールの使用に関して)規定する意図はない。説明中で、コマンド名の後の括弧には入力パラメータを、またコマンドの前には返値を記述している。このような記述は説明上のものであり、実際のコマンドの構文とエンコーディングは後の節にて規定する。各コマンドのパラメータ順序は固定ではない。ディスクリプタは、コマンドのパラメータとしてどの順序で現れてもよい。ただし、各ディスクリプタの処理は出現順に実行されなければならない。

コマンド応答にはErrorディスクリプタを含むことができるが、APIでは特にこれを示さない。

角括弧[...]内にあるパラメータは、オプションを意味する。

#### 7.2.1 Addコマンド

Addコマンドはあるcontextに一つのterminationを追加する。

```
TerminationIDList  
[,MediaDescriptor]  
[,ModemDescriptor] (*)  
[,MuxDescriptor]  
[,EventsDescriptor]  
[,SignalsDescriptor]  
[,DigitMapDescriptor]  
[,ObservedEventsDescriptor]  
[,EventBufferDescriptor]  
[,StatisticsDescriptor]
```



```
[,PackagesDescriptor]
  Add( TerminationIDList
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventsBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
    [, StatisticsDescriptor]
  )
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

TerminationIDListには、contextに追加する一つ以上のterminationを指定する。指定されたterminationは、新規に生成されるか、またはNULL Contextから移動される。CHOOSEワイルドカードをTerminationIDに指定した場合は、選択されたTerminationIDが返される。AddでALLワイルドカードを使用することはできるが、そのような指定は一般的ではないだろう。ワイルドカードが一つ以上のTerminationIDと合致した場合、合致したTermination全てに対して処理が実行され、それぞれの実行結果が返される。複数のTerminationIDが合致する際のTerminationID間の実行順序は、規定されない。

MediaDescriptorパラメータはオプションであり、全てのメディアストリームを記述する。

MuxDescriptorパラメータはオプションであり、適用され得るMultiplexerを指定する。利便性のため、MuxDescriptorがAddコマンドに存在し、かつ現在のcontext内に存在しないterminationがリストに挙げられている場合、それらのterminationに対して各々Addコマンドが発行されたかのようにcontextに追加する。このような暗黙のAddコマンドの実行中にエラーが発生した場合は、エラー471 – Multiplexerに対する暗黙のAdd失敗(“Implied Add for Multiplex failure”)を返さなければならず、また引き続くコマンドの処理は中止されなければならない。

EventsDescriptorパラメータはオプションである。存在する場合は、terminationにおいて検知されるべきイベントのリストを示す。

EventBufferDescriptorパラメータはオプションである。存在する場合は、EventBufferControlがLockStepのときMGが検知しバッファリングするイベントのリストを示す。

SignalsDescriptorパラメータはオプションである。存在する場合は、terminationに適用されるべきシグナルのリストを示す。

DigitMapDescriptorパラメータはオプションである。存在する場合は、EventsDescriptorで使用するところのDigitMap定義を規定する。

AuditDescriptorはオプションである。存在する場合は、コマンドはAuditDescriptor内に指定される全てのディスクリプタ・プロパティ・シグナル・イベント・統計情報の値を返す。

StatisticsDescriptorはオプションである。存在する場合は、MGCがterminationまたはstreamに関して収集を行いたい統計情報を示している。なお、statisticsは指定されるごとに、その値がリセットされる。

パラメータの指定が曖昧であるか過多である場合、MGは変更される可能性のある全てのディスクリプタを返すことができる。ObservedEvents、Statistics、PackagesおよびEventBufferDescriptorは、AuditDescriptorにて要求された場合のみ、返される。

Addコマンドは、serviceStateが”OutOfService”のTerminationに対して使用してはならない。

## 7.2.2 Modifyコマンド

Modifyコマンドは、terminationのプロパティを変更する。

```
TerminationIDList
[,MediaDescriptor]
[,ModemDescriptor] (*)
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,EventBufferDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    Modify( TerminationIDList
        [, MediaDescriptor]
        [, ModemDescriptor] (*)
        [, MuxDescriptor]
        [, EventsDescriptor]
        [, EventsBufferDescriptor]
        [, SignalsDescriptor]
        [, DigitMapDescriptor]
        [, AuditDescriptor]
        [, StatisticsDescriptor]
    )
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

TerminationIDListには、変更の対象とするterminationを指定する。Context内に存在するterminationのうち一つだけを変更する場合は、その特定のTerminationIDを指定することができる。オペレーションの内容によっては、TerminationIDにワイルドカードを用いることが適切な場合もある。ワイルドカードが一つ以上のTerminationIDと合致した場合、合致した全てのterminationに処理が実行され、それぞれに関する実行結果が返される。複数のTerminationIDが合致する場合、TerminationID間の実行順序は規定されない。なお、Modifyコマンドは既存のterminationに対してのみ使用可能なため、TerminationIDにCHOOSEワイルドカードを指定するとエラーになる。

利便性のため、MultiplexディスクリプタがModifyコマンドに指定されている場合、以下のように扱うこととする。

新しいMultiplexディスクリプタがその時点でcontextに存在しないterminationを列挙している場合、そのようなterminationは、それぞれに対してAddコマンドが発行されたかのように、contextに加えられる。

以前のコマンドで指定されたMultiplexディスクリプタに列挙されていたterminationのうち、新しいMultiplexディスクリプタに存在しないterminationは、それぞれに対してSubtractコマンドが発行されたかのように、contextから削除される。

Modifyのパラメータのうち上記で説明しなかったものに関してはAddと同様である。返値についてもAddと同様である。

### 7.2.3 Subtractコマンド

Subtractコマンドは、terminationを所属するcontextから切り離し、また当該terminationがcontextに所属していた際の統計情報を返す。

```
TerminationIDList
[,MediaDescriptor]
[,ModemDescriptor] (*)
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,EventBufferDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
Subtract(TerminationIDList
          [, AuditDescriptor]
          )
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

入力パラメータのTerminationIDListには、Subtractするterminationを指定する。TerminationIDは、特定のterminationに固有の値でも、ワイルドカードでもよい。ワイルドカードの場合、当該contextからSubtractすべき全ての(もしくは関連する一連の)terminationを指定する。ワイルドカードが一つ以上のTerminationIDと合致する場合は、合致する全てのterminationに対して処理が実行され、それぞれの実行結果が返される。複数のTerminationIDが合致する際のTerminationID間の実行順序は、規定されない。

Subtractコマンドは既存のterminationに対してのみ使用可能なため、TerminationIDにCHOOSEワイルドカードを指定するとエラーになる。

Subtractでは、TerminationIDと同様にContextIDに対して、ALLワイルドカードを指定することができる。この場合、全てのcontextを削除し、全てのephemeral terminationを削除し、全ての物理terminationをNULL Contextに戻す動作を行う。なお、NULL ContextからterminationをSubtractすることはできない。

利便性のため、MuxDescriptorがAddコマンドに存在し、かつ現在のcontext内に存在しないterminationがリストに挙げられている場合、それらのterminationに対して各々Addコマンドが発行されたかのようにcontextに追加する。このような暗黙のAddコマンドの実行中にエラーが発生した場合は、エラー471 – Multiplexerに対する暗黙のAdd失敗(“Implied Add for Multiplex failure”)を返さなければならず、また引き続くコマンドの処理は中止されなければならない。

利便性のため、Multiplexer terminationがSubtractコマンドの対象である場合、そのMultiplexディスクリプタに列挙されているベアラtermination全てに対して、それぞれSubtractコマンドが発行されたかのように、そのcontextからSubtractする。

動作内容が上書きされていないデフォルトの状態では、コマンドで指定される一つもしくは複数の

terminationで集められた情報をレポートするため、terminationとstreamの両方のレベルのStatisticsDescriptorが返される。返された情報は、一つもしくは複数のterminationが、それらがSubtractされたcontext内に存在していたことを示す。

AuditDescriptorはオプションである。存在する場合は、コマンドはレスポンスでAuditDescriptor内に指定される項目だけを返す。AuditDescriptorの内容は空でもよい。AuditDescriptorが省略された場合、デフォルトではStatisticsディスクリプタが返される。返値は、Addコマンドのときと同様である。

プロビジョニングで指定されているterminationがcontextからSubtractされる場合、そのterminationのディスクリプタの値は、以下の通り書き戻されなくてはならない：

- ・もしそのディスクリプタのデフォルト値が定められており、プロビジョニングによってオーバーライドされていない場合は、デフォルト値。
- ・その他の場合、プロビジョニングで規定された値。

#### 7.2.4 Moveコマンド

Moveコマンドは、不可分なオペレーションとして、あるterminationを現在のcontextから別のcontextへ移動させる。Moveコマンドは、コマンドが適用されるcontextとは異なるcontext内のterminationを参照する、ただ一つのコマンドである。Moveコマンドを、NULL Contextへterminationを移動したり、Null Contextからterminationを移動したりするために使用してはならない。

```
TerminationIDList
[MediaDescriptor]
[ModemDescriptor] (*)
[MuxDescriptor]
[EventsDescriptor]
[SignalsDescriptor]
[DigitMapDescriptor]
[ObservedEventsDescriptor]
[EventBufferDescriptor]
[StatisticsDescriptor]
[PackagesDescriptor]

Move( TerminationIDList
    [ MediaDescriptor]
    [ ModemDescriptor] (*)
    [ MuxDescriptor]
    [ EventsDescriptor]
    [ EventBufferDescriptor]
    [ SignalsDescriptor]
    [ DigitMapDescriptor]
    [ AuditDescriptor]
    [ StatisticsDescriptor]
)
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

TerminationIDListには、移動させるterminationを指定する。TerminationIDはワイルドカード指定でもよいが、

CHOOSEを使用してはならない。ワイルドカードが一つ以上のTerminationIDと合致した場合、合致したtermination全てに対して処理が実行され、それぞれの実行結果が返される。複数のTerminationIDが合致する際のTerminationID間の実行順序は、規定されない。Terminationの移動先のcontextは、action内でTargetContextIDとして指定される。Contextに最後に残っていたterminationがcontext外に移動された場合、そのcontextは削除される。

Moveコマンドに含まれるディスクリプタによって明示的に変更されない限り、Moveコマンドはterminationの属性に影響を与えない。例えば、StatisticsDescriptorを持つAuditDescriptorは、terminationの移動直前の統計情報を返す。Moveの返値に成り得るディスクリプタは、Addのものと同様である。

利便性のため、multiplexing terminationがMoveコマンドの対象である場合は、そのMultiplexディスクリプタに列挙されたいずれのベアラterminationもまた、あたかもそれらのterminationに対して各々Moveコマンドが実行されたかのように、移動される。

Moveは、serviceStateが”OutOfService”になっているterminationに対して使用してはならない。

### 7.2.5 AuditValueコマンド

AuditValueコマンドは、terminationに関連するプロパティ、イベント、シグナルおよび統計情報の現在の値を返す。AuditValueは、ディスクリプタの中身もしくは、個々のプロパティ、イベント、シグナルまたは統計情報を要求してよい。AuditValueは、MGCとMG間のterminationの同期維持に役立つかもしれない。AuditValueは、指定する選択規則によって返値がフィルタリングされるように要求してもよい。

```
TerminationIDList
[,MediaDescriptor]
[,ModemDescriptor] (*)
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,EventBufferDescriptor]
[,StatisticsDescriptor]
[,PackagesDescriptor]
    AuditValue(TerminationIDList,
                AuditDescriptor
    )
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

TerminationIDListには、TerminationIDのリストを指定する。TerminationIDとしては、特定のTerminationに固有の値でも、ワイルドカードでもよい。ワイルドカードが一つ以上のTerminationIDと合致する場合、合致する全てのterminationに対して処理が実行され、それぞれの実行結果が返される。複数のTerminationIDが合致する場合のTerminationID間の実行順序は、規定されない。ワイルドカードによる応答が要求される場合、ワイルドカードまたはリストに合致する全てのterminationに関するの値の和集合を内容に含む、ただ一つのコマンドの返値が生成される。この取り決めによって、terminationのグループを監査するために必要なデータを削減することができる。TerminationIDにCHOOSEワイルドカードを指定するとエラーになる。

ディスクリプタもしくは個々のプロパティ、シグナル、イベントおよび統計情報が監査され得る。指定さ

れた選択規則によって返値がフィルタリングされることがある。

- ディスクリプタ全体の監査は、詳細な情報のないAuditDescriptorで希望するディスクリプタを指定することにより要求できる。
- メディアディスクリプタ内の個々のプロパティを監査する場合、関連づけられたStreamID(オプション)、GroupID(オプション)、およびPropertyIDが含まれる。プロパティの現在の値が返される。GroupIDは、Local Control ディスクリプタのReserved Groupフラグが使われる場合に使用される。GroupID 1は予約されている最初のグループ(セッション記述)と、GroupID2はその次のグループと、のように対応する。
- シグナルを監査する場合、関連付けられたSignalListIDとSignalIDの両方またはいずれかが指定される。StreamIDとRequestIDの両方またはいずれかはオプションである。シグナルが有効な場合に限り、すべてのシグナルのパラメータ値は次のものを含み返信される。すなわち、KeepActive表示、シグナルタイプ、期間、シグナル完了表示、およびパッケージで定義されたプロパティである。
- イベントを監査する場合、関連付けられたStreamID(オプション)、EventID、RequestID(オプション)が指定される。すべてのイベントのパラメータ値は、次の内容を含んで返される。それはイベントのアクションと、パッケージで定義されたパラメータである。
- Terminationレベルの統計情報を監査する場合、統計情報の識別子が指定される。Streamレベルの統計情報を監査する場合、関連づけられたStreamID(オプション)、GroupID(オプション)、StatisticID(オプション)がMedia Descriptorに含められる。その時点における統計値が返され、その統計情報はリセットされない。
- パッケージを監査する場合、パッケージの識別子とバージョンが提供される。そのパッケージで定義されたすべてのプロパティ、シグナル、イベントが、それらの値とともに返される。
- Auditの戻り値をフィルタリングする場合、audit要求に選択規則が含まれる。複数の選択規則を含めることもできる。複数の選択規則を含む場合、選択規則間の演算操作を指定するために、ANDまたはORの論理オペレーションを含めることができる。論理オペレーションが含まれていない場合、AND論理オペレーションが指定されたものと仮定される。

一つの要求内で複数の個々のアイテムを監査することが可能である。

ディスクリプタの監査が要求された場合、あるterminationの現在の値を含む適切なディスクリプタが、AuditValueから返される。一つのディスクリプタに対して複数のインスタンスとして返される値は、サポートされている選択肢として定義され、ディスクリプタ内の各パラメータは、独立したものと位置付けられる。

ObservedEventsは、EventBuffer内のイベントリストを返す。DigitMapが有効な間にObservedEventsDescriptorが監査される場合、返されるObservedEventsディスクリプタは、Digit map完了イベントをも含む。このイベントは、現在のダイヤル文字列を示すが、終了方法は示さない。

EventBufferは、イベントの集合とそれに関連づけられたパラメータ値で、EventBufferディスクリプタ内でその時点で有効になっているものを返す。PackagesDescriptorは、そのterminationが認識するパッケージのリストを返す。DigitMapDescriptorは、そのtermination用の現在のDigitMapの名前もしくは値を返す。TerminationIDがALLであるAuditValueコマンドでDigitMapが要求される場合、そのMG内にある全てのDigitMapを返す。統計情報は、そのterminationレベルで保持されている全ての統計情報の現在値を返す。MediaDescriptorに対するauditは、任意のstreamレベルの統計情報を返す。空のAuditディスクリプタを指定した場合は、TerminationIDのみが返される。これは、ワイルドカードと共に使うことで、TerminationIDのリストを得るのに有効である。付属資料Aおよび付属資料Bは、このようなリストを要約した形式で表記するための特別な構文を提供しており、AuditValueコマンドタグがTerminationID毎に繰り返される必要がないようにしている。

AuditValueの結果は、context、即ち、特定の値が指定されている(Specific)か、NULLか、もしくはワイルドカードかに依存する。(ContextIDのALLは、NULLContextは含まないことに注意)。TerminationIDは、特定の値が指定されて(Specific)いても、ワイルドカードであってもよい。

Contextとterminationの両方またはいずれかがワイルドカード指定され、かつワイルドカードによる返値を返すよう指定された場合の返値の例を以下に示す：

ここでは以下のように仮定する。

ゲートウェイは4つのTermination、 t1/1、 t1/2、 t2/1およびt2/2を持つ。

Termination t1/\*はパッケージaaaおよびbbbを実装しており、Termination t2/\*はパッケージcccおよびdddを実装している。

Context 1はその中にt1/1およびt2/1を持ち、Context 2はその中にt1/2およびt2/2を持っている。

このとき、

コマンドContext=1 {AuditValue=t1/1 {Audit {Packages}}}は

Context=1 {AuditValue=t1/1 {Packages {aaa,bbb}}}}を返し、

コマンドContext=\* {AuditValue=t2/\* {Audit {Packages}}}は

Context=1 {AuditValue=t2/1 {Packages {ccc,ddd}}},

Context=2 {AuditValue=t2/2 {Packages {ccc,ddd}}}}を返し、

コマンドContext=\* {W-AuditValue=t1/\* {Audit {Packages}}}は

Context=\* { AuditValue=t1/\* {Packages {aaa,bbb}}}を返す。

注記：ワイルドカード応答はまた、Subtractなどの他のコマンドに使用されてもよい。

Root terminationに対して、contextにワイルドカードを指定するような場合、MGCは個々のcontextごとに別々のアクションに分けて応答を返すのではなく、ContextIDのコンパクトなリストで返すよう指示することができる。

ゲートウェイが1、2、3、4という4つのContextをもち、それぞれが(t1-t8)の2つのterminationを含むと仮定する。以下に、MGのauditコマンドに対する応答の例を示す：

コマンドContext=\* {AuditValue=root {Audit {}}}は

Context=1 {AuditValue=context{\*}}, Context=2 {AuditValue=context{\*}},

Context=3 {AuditValue=context{\*}}, Context=4 {AuditValue=context{\*}}

または

Context=1 {AuditValue=t1 {}, AuditValue=t2 {}}, Context=2 {AuditValue=t3 {}, AuditValue=t4 {}},

Context=3 {AuditValue=t5 {}, AuditValue=t6 {}}, Context=4 {AuditValue=t7 {}, AuditValue=t8 {}}

を返し、

コマンドContext=\* {ContextAttr {ContextList {\*}, AuditValue=Root {Audit {}}}は

Context=\* {ContextAttr {ContextList {1,2,3,4}, AuditValue=Root {}}

を返す。

以下の表に、AuditValueコマンドで取得可能な他の情報を示す。

表7-2/JT-H248.1 AuditValueコマンドで獲得可能な情報

(ITU-T H.248.1)

ContextID	TerminationID	得られる情報
特定の値	ワイルドカード	context内の一致するterminationの監査結果
特定の値	特定の値	context内の特定のterminationの監査結果
NULL	Root	MGの状態とイベントに関する監査結果
NULL	ワイルドカード	NULL Context内に存在する、全ての一致するterminationの監査結果
NULL	特定の値	いずれのcontextにも属さない特定のterminationの監査結果
ALL	ワイルドカード	NULL Context外にある全ての一致するterminationと、それらに関連付けられるcontextの監査結果
ALL	Root	全てのContextIDのリスト(ContextIDのリストは、それぞれがリストのContextIDを含んでいる複数のAction Replyの利用により返却してもよいし、ContextIDListパラメータを使って返却してもよい。応答の方法は、RequestにContextIDListパラメータが存在するか否かによる。)
ALL	特定の値	指定したterminationが現存する(NULLでない)ContextのContextID

### 7.2.6 AuditCapabilitiesコマンド

AuditCapabilitiesコマンドは、terminationに関連するプロパティ、イベント、シグナルおよび統計情報の取り得る値を返す。AuditCapabilitiesは、ディスクリプタの内容もしくは、いずれかのプロパティ、イベント、シグナルまたは統計情報を要求されてもよい。

```
TerminationIDList
[MediaDescriptor]
[ModemDescriptor] (*)
[MuxDescriptor]
[EventsDescriptor]
[SignalsDescriptor]
[ObservedEventsDescriptor]
[EventBufferDescriptor]
[StatisticsDescriptor]
    AuditCapabilities(TerminationIDList,
        AuditDescriptor
    )
```

(\*) ModemDescriptorは、JT-H.248.1では非推奨である。

ディスクリプタもしくは個々のプロパティ、シグナル、イベントおよび統計情報を監査することができる。

- ・ ディスクリプタ全体の監査は、詳細な情報のないAuditDescriptorで希望するディスクリプタを指定することにより要求できる。
- ・ メディアディスクリプタ内の個々のプロパティを監査する場合、関連づけられたStreamID(オプション)、およびPropertyIDが含まれる。プロパティの取り得る値のリストが返される。
- ・ シグナルを監査する場合、関連づけられたSignalListIDとSignalIDのいずれかまたは両方を指定する。StreamIDとRequestIDのいずれかまたは両方は、オプションである。各シグナルのパラメータが取り得る値のリストが返却される(パッケージで定義されたプロパティを含む)。KeepActive表示、シグナルタイプ、期間、およびシグナル完了表示は返却されない。



- ・ イベントを監査する場合、関連づけられたStreamID(オプション)、EventID、RequestID(オプション)を呈する。各イベントパラメータの取り得る値のリストが返却される(イベントアクションとパッケージで定義されたパラメータを含む)。
- ・ Terminationレベルの統計情報を監視する場合、統計情報の識別子が指定される。Streamレベルの統計情報を監査する場合、関連づけられたStreamID(オプション)、GroupID(オプション)、StaticIDがMedia ディスクリプタに含まれる。その統計情報の取り得る値が返却され、統計情報はリセットされない。

ディスクリプタ監査が要求された場合、terminationの取り得る値を含む関連づけられたディスクリプタが、AuditCapabilitiesで返される。取り得る値が複数ある場合、ディスクリプタが繰り返されてもよい。

ワイルドカードの応答が要求される場合、リスト中の全てまたはワイルドカードに合致する全てのterminationの値の和集合を含む、ただ一つのコマンド応答が生成される。この取り決めによって、terminationのグループを監査するために必要なデータ量を削減することができる。

プロパティ、シグナル、イベントもしくは統計情報が監査される場合、適切なプロパティ・シグナル・イベント・統計情報と、terminationの能力がAuditCapabilitiesから返却される。

ContextIDとTerminationIDの様々な値に対してどのような能力が要求されていると解釈するのについてはAuditValueの場合と同様である。

型の文字列、文字列またはオクテット列の値をとるpropertyおよびパラメータ値に対しては、MGは空の値を返さねばならない。テキストエンコーディングの場合、オクテット列についてはNUL(0x00)を返すが、文字列と文字については、空のquotedStringを生成して返す。この動作はパッケージの規定により上書きしてもよい。EventsDescriptorは、EventsDescriptorパラメータの取り得る全ての値のリストとともに、指定するterminationに起こり得るイベントのリストを返す。EventBufferDescriptorは、EventsDescriptorと同じ情報を返す。SignalsDescriptorは、シグナルパラメータの取り得る全ての値のリストとともに、指定するterminationに適用され得るシグナルのリストを返す。StatisticsDescriptorは、指定するterminationに保持されている統計情報の名前を返す。ObservedEventsDescriptorは、指定するterminationで有効なイベントの名前を返す。DigitMapおよびPackagesは、AuditCapabilityでは許されていない。

以下の表に、AuditCapabilitiesコマンドで取得可能な他の情報を示す。

表7-3/JT-H248.1 AuditCapabilitiesコマンドで取得可能な情報

(ITU-T H.248.1)

ContextID	TerminationID	得られる情報
特定の値	ワイルドカード	context内の一致するterminationの監査結果
特定の値	特定の値	context内の特定のterminationの監査結果
NULL	Root	MGの状態とイベントに関するの監査結果
NULL	ワイルドカード	NULL Context内の全ての一致するterminationの監査結果
NULL	特定の値	いずれのcontextにも属さない特定のterminationの監査結果
ALL	ワイルドカード	NULL以外のContextに存在する、全ての一致するterminationとそれらに関連づけられたcontextの監査結果
ALL	Root	AuditValueに同じ
ALL	特定の値	AuditValueに同じ

## 7.2.7 Notifyコマンド

NotifyコマンドはMGがMG内部で発生したイベントをMGCへ通知するために使用されるコマンドである。

```
TerminationID
  Notify(TerminationID,
        ObservedEventsDescriptor,
        [ErrorDescriptor]
  )
```

TerminationIDはNotifyコマンドを発行したterminationを指定する。TerminationIDは特定のterminationを完全に識別できる名前ではなくてはならない。

ObservedEventsDescriptorはRequestIDとMGで検出した検知順のイベントリストを含んでいる。リスト内の各イベントはそのイベントに関連したパラメータと、オプションでそのイベントを検出した時間を伴ったものとなる。RequestID=0のNotifyコマンドを送出する手順については将来検討が必要である。

関連づけられたtermination上で有効であるEventsディスクリプタによって指定されるイベントを検出した結果として使用する場合のみ、RequestID=0ではないNotifyコマンドが発生しなくてはならない。

RequestIDはNotifyコマンドが発生したきっかけとなったEventsDescriptorのRequestIDパラメータを返値として返す。これにより、Notifyと、それを設定したRequestとを関連づけるために使用される。リストにあるイベントは、RequestID=0でなければ、それを設定したEventsDescriptorか、または埋め込み型Eventsディスクリプタを用いて要求されていたはずである(将来検討が必要)。

ErrorDescriptorは、エラー518 – イベントバッファが一杯: "Event buffer full" という結果としてNotifyで送信されてもよい。

## 7.2.8 ServiceChangeコマンド

単独の、もしくは複数のterminationがサービスを停止しそうであること、もしくはたった今サービスを再開したことを、MGがMGCに対して通知するときにServiceChangeコマンドが使用される。MGCがterminationのサービス停止、再開を指示してもよいし、MGがterminationの能力が変化したことをMGCに通知してもよい。あるMGCが、あるMGの制御を他のMGCに引き渡すことも許される。

```
TerminationIDList,
[ServiceChangeDescriptor]
  ServiceChange(TerminationIDList,
                ServiceChangeDescriptor
  )
```

TerminationIDListはサービスが停止もしくは再開したterminationを指定する。Termination名にワイルドカードを指定することはできるが、例外としてCHOOSEを指定してはならない。"Root"TerminationIDの使用は、MG全体に影響を及ぼすServiceChangeであることを示す。

注記: TerminationIDListは、初版であるITU-T H.248.1第1版のServiceChangeコマンドでは使用することができない。

### 7.2.8.1 ServiceChangeディスクリプタの内容

ServiceChangeDescriptorは必要に応じて以下のパラメータを含む。

- ServiceChangeMethod

- ServiceChangeReason
- ServiceChangeDelay
- ServiceChangeAddress
- ServiceChangeProfile
- ServiceChangeVersion
- ServiceChangeMgcID
- TimeStamp
- ExtensionParameter
- ServiceChangeInfo
- ServiceChangeIncompleteFlag

#### 7.2.8.1.1 ServiceChangeMethod

ServiceChangeMethodパラメータは、これから発生する、あるいは発生したServiceChangeの種別を示す。

- (1) Graceful — terminationが指定されたServiceChangeDelay後にサービス停止となることを示している。既に確立されたコネクションには影響がないが、MGCは新しいコネクションを確立することは控えるべきであり、ServiceChangeコマンドによって影響されるtermination上の既に確立しているコネクションをすみやかに切断するべきである。MGは、ServiceChangeDelayの満了、もしくは有効なcontextからのterminationの削除(どちらでも最初に発生するもの)を契機に、terminationのserviceStateを”out of service”に設定すべきである。
- (2) Forced — terminationが突然サービス停止したこと、そしてそのterminationに関連する既に確立されている全てのコネクションが切断される可能性があることを示すものである。Rootでないterminationに関してはMGCはサービスを停止した(もしあれば) terminationを含むcontextをクリアする責任がある。最低でもサービスを停止したterminationは速やかにcontextから削除されなければならない。そのterminationのServiceStateは”out of service”となるべきである。Root Terminationに関しては、MGCは、MGにおける全ての接続が失われたと想定でき、よって全てのterminationがSubtractされたと考えることができる。
- (3) Restart — ServiceChangeDelayが満了したあとterminationがサービスを再開するというを示している。そのServiceStateはServiceChangeDelay満了を契機に”inService”に設定されるべきである。
- (4) Disconnected — 常にRoot TerminationIDに対して適用され、MGとMGCとの通信が不能になったが、(おそらくは事前にプロビジョニングされたリスト上の他のMGCへの通信を試みた後で)その後同じMGCに復帰したことを示している。MGの状態が変化している可能性があるので、MGCはAuditコマンドを使用してMGCの状態とMGの状態を再同期させてもよい。
- (5) Handoff — MGCからMGへ送られる場合、このServiceChange理由は、現在のMGCがサービスを停止するため、MGは他のMGCとコネクションを確立しなければならないことを示す。MGからMGCへHandoffを送出する場合は、MGがそれまでコネクションを確立していたMGCから受信したHandoffに従って、他のMGCと新たなコネクションを確立しようとしていることを示している。
- (6) Failover — MGからMGCへ送られる場合は、primary-MGがサービスを停止し、secondary-MGがその代替として通信可能状態となったことを示している。このServiceChangeメソッドは、MGがMGCの故障を検出した場合にもMGからMGCへ送られる。
- (7) その意味がMGとMGCとでお互いに理解される、上記以外の値。

#### 7.2.8.1.2 ServiceChangeReason

ServiceChangeReasonパラメータは、ServiceChangeが発生した、もしくはこれから発生する理由を指定する

パラメータである。このパラメータは英数字のトークン(IANAで登録されている)と、オプションの説明文とで構成されている。

#### 7.2.8.1.3 ServiceChangeAddress

オプションのServiceChangeAddressパラメータは、その後の通信で使用されるアドレス(例えば、IPネットワークのIPポート番号)を指定するパラメータである。このパラメータは入力パラメータのディスクリプタや、返却された結果のディスクリプタの中で指定することができる。ServiceChangeAddressパラメータとServiceChangeMgcIDパラメータは、ServiceChangeDescriptorまたはServiceChangeResultDescriptorにおいて共存させてはならない。ServiceChangeAddressは現在の接続のために使用されるアドレスである。一方、ServiceChangeMgcIDは、MGが他のMGCと接続しようとする場合に使用すべき代替りのアドレスを提供する。ServiceChangeAddressの使用は推奨されていないことに注意が必要である。MGCとMGは、ServiceChangeAddressが完全なアドレスである場合でも、TCP転送でのポート番号である場合でも、対応可能でなければならない。

#### 7.2.8.1.4 ServiceChangeDelay

オプションのServiceChangeDelayは秒単位で表現される。その遅延が無いか、もしくは"0"の場合、その遅延の値はnullにすると考えられるべきである。"Graceful" ServiceChangeMethodの場合、null delayは、MGCが既存のコネクションを自然に消去されるのを待つべきであり、新しいコネクションを設定しないようにするべきであるということを示している。"Graceful"な場合のみ、null delayは、MGはそのTerminationがNULL Contextに入るまで、ServiceStateを"OutOfService"にしてはならないことを意味する。

#### 7.2.8.1.5 ServiceChangeProfile

オプションのServiceChangeProfileパラメータはサポートするプロトコルのプロファイル(もしあれば)を示すパラメータであり、サポートするプロファイルのバージョンを含んでいる。本パラメータが存在しない場合、値"NoProfile"が指定されたものとする。

#### 7.2.8.1.6 ServiceChangeVersion

オプションのServiceChangeVersionパラメータはH.248プロトコルバージョンを含んでいる。プロトコルバージョンのネゴシエーションが発生した場合に使用される(節11.3参照)。

#### 7.2.8.1.7 TimeStamp

オプションのTimeStampパラメータは送信側によって保持される実時間を示すものである。TimeStampパラメータ自体は、絶対時刻、例えばローカルタイムゾーンの時刻である必要はない。単に、あるアソシエーションが確立されている期間において、送信側から将来転送される全てのタイムスタンプが比較されなければならない、任意の開始時刻を設定するにすぎない。時間の概念が通信相手とどのように異なっているかということを示すために用いることもできる。TimeStampは100分の1秒の正確さで送信される。

#### 7.2.8.1.8 ExtensionParameter

オプションのExtensionパラメータはMGとMGCとでお互いに意味を理解するものであれば、どのような値を含んでもよい。値"X-SC"はJT-H.248第2版(およびそれ以降の版)で追加されたServiceChangeパラメータを通信するために予約されている。初版であるITU-T H.248.1第1版に基づきコーディングされたMGからMGCへのServiceChangeコマンドで、ServiceChangeVersionが3またはそれ以上のときのみ使用される。値の構造は、以下のABNFで定義される。

X-SC = 1\*(NAME EQUAL paramValue[COMMA])

新しくServiceChangeパラメータを追加する場合、もしそのパラメータが最初のServiceChangeで送られるならば、名前/値を付与することが推奨される。例えば、

名前：最大64文字で、スペースを含まず、アルファベットで始まり、英数字と数字列のいずれかまたは両方で構成される。特殊文字アンダースコア (“\_”) を含む可能性がある。

型：12.1.2節 “Properties”による。

取り得る値の範囲：12.1.2節 “Properties”による。

#### 7.2.8.1.9 ServiceChangeInfo

オプションのServiceChangeInfoパラメータは、ServiceChangeを引き起こした原因となるパッケージ、プロパティ、シグナル、イベント、統計情報を含んでもよい。

#### 7.2.8.1.10 ServiceChangeIncompleteFlag

このフラグはオプションであり、terminationの状態を通知するために、ServiceChangeコマンドがMGからMGCに対して続いて送信されることを示す。ServiceChangeIncompleteFlagは、MGがMGCにMG全体とterminationの状態を報告しようとする場合、MG登録または再開 (ServiceChangeMethodにRestartが指定されたRootのServiceChange) のときにのみ、利用される。本フラグを受信した場合、MGCがMGに対しRoot以外のterminationへのコマンドを送出してはならない。最初の登録/リスタートコマンドでServiceChangeIncompleteFlagを送信後、MGはMGとそのterminationの現在の状態が報告されたと判断するまで、MGはそれに引き続くすべてのServiceChangeコマンドで、ServiceChangeIncompleteFlagを送り続けなければならない。MGまたはそのterminationの状態を示す最後のServiceChangeコマンドでこのフラグは取り除かれる。ServiceChangeIncompleteFlagなしのServiceChangeコマンドを受信した場合、MGCは再度コマンドをMGに送信してもよい。本フラグが初版に準拠したServiceChange (ITU-T H.248.1 第1版) コマンドでServiceChange拡張パラメータ”X-SC”を使って送られるとき、以下が使われる。

名前：SIC

型：ブール

取り得る値：ON—フラグが含まれる

注記：後続のコマンドがJT-H248第2版のエンコーディングを用いる場合、値OFFを指定する必要はない。

#### 7.2.8.1.11 ServiceChangeコマンドと応答

TerminationIDに”Root”を指定し、ServiceChangeMethodにRestartを指定するServiceChangeコマンドは、MGがMGCに対してその存在を通知するために使用される登録コマンドである。MGは、MGCの故障を検出したときには、TerminationIDにRootを、ServiceChangeMethodにFailoverを指定している登録コマンドを送信してもよい。TerminationIDにRootを指定し、ServiceChangeMethodにRestartまたはFailoverを指定するServiceChangeコマンドを含むメッセージには、新しいServiceChangeAddressやネゴシエーションされたH.248プロトコルバージョンを使用すべきようなコマンドを含めてはならない。

MGは一つのprimary-MGCとオプションとしていくつかの代替MGCの名前をプロビジョニングによってあらかじめ設定されていることを期待する。以下に示す通り、MGCが代替となるServiceChangeMgcIDを返した場合を除いて、ServiceChangeコマンドの確認応答によって登録プロセスは完了する。

MGは、MGCがMGにメッセージを送信する際に使用するトランスポートのServiceChangeAddressを、ServiceChangeDescriptor中のServiceChangeAddressパラメータで指定してもよい。MGは、ServiceChange要求のServiceChangeAddressパラメータにアドレスを指定してもよい。同様に、MGCもServiceChange応答でアドレスを指定してもよい。どちらの場合でも、受信側はその接続におけるそれ以降のtransactionのために、指定

されたアドレスを送信先として使用しなければならない。また同時に9章に示すように、`transaction reply`と`transaction pending`は、対応するRequestの送信元アドレスに対して送信されなければならない。この規定のために、（新たに発行するコマンドと以前のコマンドに対する応答を異なるアドレスに送信しなければならないため）コマンドと応答を単一のメッセージとしてひとまとめにすることができなくなり、余分な数のメッセージを送信しなければならないような場合であっても、以上の処理を実施しなければならない。

`TimeStamp`パラメータは登録コマンドおよびその応答とともに送出されなければならない。

登録時、MG自体は”`InService`”の状態であるが、個別の`termination`が”`OutOfService`”の状態にある場合、MGは`ServiceChangeIncompleteFlag`を送ってよい。これは、サービス停止している`termination`を捕捉しようとするような操作が行われないようにするためある。後続の`ServiceChange`コマンドによって、”`OutOfService`”状態の`termination`の通知を行う。

MGCは、`ServiceChangeMgcID`パラメータを返信し、以降のサービスのためにMGが優先的にコンタクトすべきMGCを指定してもよい。この場合、MGは新しいMGCに対して`ServiceChange`コマンドを再送出しなければならない。もし`ServiceChangeMgcID`パラメータが指定されている場合、他の全てのMGCよりも優先して接続を行わなければならない。MGCからMGへ`HandOff`メッセージが送出される場合、そのメッセージの`ServiceChangeMgcID`は、現在のMGCを引き継ぐ新しいMGCのIDを示す。

`ServiceChange`からのリターン値は、`Root TerminationID`が使用される時以外は空である。`Root Termination`の場合は、必要に応じて以下のパラメータを含む。

- `ServiceChangeAddress` : 応答するMGCが、MGに対して今後送信するメッセージの新しい送信先を指定しようとする場合
- `ServiceChangeMgcID` : 応答するMGCが、MGとの接続を維持することを希望しない場合
- `ServiceChangeProfile` : 応答する側が接続で使用するプロファイルをネゴシエーションしたい場合。プロファイル(名前とバージョン)は、MGCが`ServiceChangeRequest`で指定されたプロファイルをサポートできない場合のみ返される。応答にはサポートされているプロファイルとバージョンを示すか、またはプロファイルをサポートしていない場合は”`NoProfile`”を示さなければならない。応答で示されたプロファイルを受け入れるならば、MGは現在のMGCとの接続を維持してもよいし、`secondary-MGC`との間に接続を確立してもよい。もし、MGが要求した以外のプロファイルをMGCが応答した場合は、MGは以下に示すいずれかの動作を行わなければならない。
  - a. MGCが指定したプロファイルがMGが受け入れることを応答するために、MGは受け入れるプロファイルを記述した新たな`ServiceChange`コマンドを発行し、接続を維持する。
  - b. 接続を維持し、MGCは`ServiceChange`応答で送信したプロファイルを使用する。
  - c. MGがもともと指定したプロファイルを使うために、異なるMGCとの間で接続を開始する。
- `ServiceChangeVersion` : 応答する側が、この接続において使用するH.248プロトコルのバージョンをネゴシエーションしたい場合

`ServiceChangeReasons`はITU-T勧告H.248.8で定義されている。このリストは、節14.3で述べられるようにIANA登録によって拡張されてもよい。

### 7.2.9 Context属性の操作と監査

本節までで記述されたH.248プロトコルのコマンドは`termination`に適用される。本節では`context`に関する操作と監査に関して規定する。

一つの`action`は`context`の操作と監査の指示を含んでもよい(8章参照)。

MGへ送出する`action`の要求は`context`の属性を監査する要求を含んでもよい。2種類の監査がある：

値の監査: MGCは、個々のcontextプロパティの現在の値を判断するために、特定のcontextを監査してもよい。MGCは、監査要求でContextIDにALLを指定することによって全ての存在する(NULLでない) contextに関するその時点での値を判断してもよい。もしcontext属性が追加される、または監査要求と同じactionによって変更された場合、返却される値はactionが処理された後のものでなければならない。 context属性は、監査の戻り値をフィルタする選択規則として使用できる。複数の選択規則を含めることができる。複数の選択規則を含む場合、選択規則間の演算操作を指定するために、ANDまたはORの論理オペレーションを含めることができる。論理オペレーションが含まれていない場合、AND論理オペレーションが指定されたものと仮定される。

能力の監査: MGCは、個々のcontextプロパティが取り得る値を判断するために、特定のcontextを監査してもよい。MGCは、監査要求にContextID ALLを指定することによって、MG全体のcontextに関して取り得る値を判断してもよい。

以下の表は、Contextに対する監査で取得することのできる情報を示している。

表7-3/JT-H248.1 Contextの監視  
(ITU-T H.248.1)

ContextID	TerminationID	AuditValue	AuditCapabilities
特定の値	適用不可	指定されたcontextのプロパティ値	指定されたcontextにおいて、contextプロパティが取り得る値
NULL	適用不可	非許容	非許容
ALL	適用不可	全ての存在する(NULLでない)contextに関するその時点での値 ContextID ALL に対する応答は、context毎のactionReplyにより示される。	ContextID ALL に対する応答は、MG全体に渡って取り得る、contextプロパティの値として示される。

Actionには、contextの属性を変更する要求が含まれていてもよい。

Actionのreplyにはcontextのプロパティが含まれている可能性があり、その場合contextのプロパティはMGCへ情報を返すために使用される。その情報は、context属性を監査するときに要求された情報でもあり得るし、contextを処理した結果の詳細情報でもあり得る。

MGがcontextの属性を監査する要求と、その属性を操作する要求の両方を含んだactionを受信した場合、その応答には、操作要求を実行した後の属性の値を含まなければならない。

#### 7.2.10 汎用的なコマンド構文

H.248プロトコルは、バイナリフォーマットまたはテキストフォーマットでエンコーディングできる。MGCは両方のエンコーディングフォーマットをサポートするべきである。MGは両方のフォーマットをサポートしてもよい。

バイナリフォーマットのプロトコル構文は付属資料Aで定義されている。付属資料Cは、バイナリフォーマットを使用する場合のLocal/Remoteディスクリプタに関するエンコーディング方式を示す。

RFC2234で規定されるABNFを用いた、H.248の完全なテキストエンコーディング方式を付属資料Bに示す。7.1.8節で規定される変更を適用したSDPは、テキストエンコーディングにおいてLocal/Remoteディスクリプタを指定するために用いられる。

## 8. TRANSACTIONS

MGCとMG間のコマンドはtransactionの中にまとめられ、各々がTransaction IDにより識別される。transactionは一つ以上のactionからなる。一つのactionは、ある一つのcontext内への作用に限られる、空でない一連の一つ以上のコマンド、contextプロパティ変更もしくはcontextプロパティ監査で構成される。よってactionには通常、それぞれContextIDを指定する。しかしながらactionとして特定のContextIDが指定されない状況が二つある。一つはcontext外でterminationを変更する場合である。もう一つはMGCがMGに新しいcontextを生成するよう要求する場合である。Transactionとaction、およびコマンドとの間の関係を図8-1/JT-H248.1に図示する。

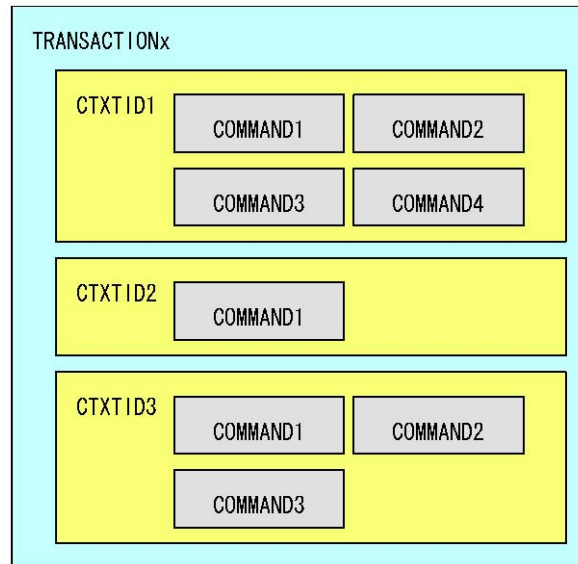


図8-1/JT-H248.1 Transaction、Actionおよびコマンド

(ITU-T H.248.1)

TransactionはTransactionRequestとして与えられる。あるTransactionRequestの応答は、単一のReplyとして返される。Replyが返される前には、複数のTransactionPendingメッセージが先行して受信されるかもしれない(節8.2.3参照)。

Transactionはコマンドの順次実行性を保証する。すなわちtransaction内のコマンドはシーケンシャルに実行される。Transaction間の実行順序は保証されない。Transactionは任意の順序で実行され、場合によっては並列して実行されることも有り得る。ただし、一つのメッセージ中にTransactionRequestとTransactionReplyの両方が含まれる場合は、TransactionRequestの前にTransactionReplyが先に実行されるべきである。

Transaction中のコマンドを実行中に失敗が発生した場合、最初に失敗した時点でそのtransaction中のコマンド実行を中止し、残りのコマンドの処理は行わない。もしコマンドが実行対象のTerminationIDとしてワイルドカードを含む場合、ワイルドカードに合致する実際のterminationIDそれぞれに対してコマンドが実行される。たとえ一つ以上のterminationIDに対してエラーが発生したとしても、ワイルドカードが合致する全てのterminationIDに対して、それぞれに対するコマンドの応答が単一のTransactionReplyで返される。ワイルドカードに合致するいずれかのterminationIDでエラーが発生した場合、そのワイルドカードを使ったコマンドに続くコマンドは一切実行されない。

コマンドは前述の振る舞いを上書きできる"オプション"("Optional")とマークされてもよい。オプションと指定されたコマンドがエラーになった場合でも、そのtransaction内の後続のコマンドは実行される。もしコマンドが失敗した場合、MGはコマンドの処理を継続する前に、失敗したコマンドが実行される前の状態へ、可能な限り修復しなければならない。

TransactionReplyは、対応するTransactionRequest内の全てのコマンドに対する結果を含む。そのTransactionReplyは、首尾よく実行されたコマンドに対するリターン値、および失敗したあらゆるコマンドに



対するコマンドとエラーディスクリプタを含む。コマンド応答は、対応するTransactionRequestに出現する順序で返される。transactionPendingは、transactionは未だ完了していないが、処理が実行されていることを定期的に受信側に通知するために使用される。

アプリケーションはtransaction毎にアプリケーションレベルのタイマを実装すべきであり、タイマ満了時にRequestの再送を行うべきである。Replyを受信した場合はそのタイマをキャンセルすべきであり、Pendingを受信した場合はそのタイマを再起動すべきである。

## 8.1 共通パラメータ

### 8.1.1 Transaction識別子

TransactionはTransactionIDにより識別される。TransactionIDは送信側により割り当てられ、かつ送信側のスコープ内で一意である。RequestにTransactionIDが欠けていることを示すためのErrorディスクリプタを含む応答は、対応するTransactionReplyの中でTransactionID 0を使用しなければならない。

### 8.1.2 Context識別子

ContextはContextIDにより識別される。ContextIDはMGにより割り当てられ、かつMGのスコープ内で一意である。MGCはそのcontextに関係している後続の全てのtransaction内で、MGから通知されたContextIDを使用しなければならない。MGCがTerminationを参照する場合、そのterminationがまだContextに関連付けされていない場合は、ContextIDとしてMGCが使用できる特別な値であるNULL ContextIDを参照する。

CHOOSEワイルドカードはMGが新しいcontextを生成するよう、要求するために使用される。

MGCはMGの全てのcontextを指定するためにALLワイルドカードを使用することができる。ただし、ALLワイルドカードが使用された場合でもNULL Contextは含まれない。

MGCはContextIDに対して部分的にCHOOSEまたはALLワイルドカードを指定してはならない。

## 8.2 Transaction API (Application Programming Interface)

本節では、H.248プロトコルのtransactionに関するアプリケーション・プログラミング・インタフェース(API)を説明する。以下のAPIに関する記述は、transactionとそのパラメータ群を説明するためのものであり、実装に関して(例えば、ブロックするようなファンクションコールの使用に関して)規定する意図はない。H.248プロトコルの様々なtransactionで使用されるであろう入力パラメータや返値に関して、非常に高水準な記述を行うものである。Transactionに関する文法やエンコーディングに関しては、後の節で規定する。

### 8.2.1 TransactionRequest

TransactionRequestは送信側により発行される。発行するRequest毎に一つのtransactionがある。Requestは一つ以上のactionからなり、各actionには対象とするcontextを指定する。

```
TransactionRequest(TransactionID {  
    ContextID {Command _ Command},  
    ...  
    ContextID {Command _ Command } })
```

TransactionIDパラメータには、受信側から返されるTransactionReplyまたはTransactionPendingとの対応をとるために値を指定しなければならない。

ContextIDパラメータは、次のContextIDパラメータの指定またはTransactionRequestの末尾のどちらが先に来るにせよ、その先に来るものよりも前にある全てのコマンドに関連する値を指定しなければならない。

Commandパラメータには、「7.2 コマンドAPI」に記述されているコマンドのいずれかを指定する。

## 8.2.2 TransactionReply

TransactionReplyは受信側により発行される。Transaction毎に一つのReplyがある。Replyは一つ以上のactionからなり、各actionには対象とするcontextと、context毎に一つ以上の応答が指定されていなければならない。TransactionReplyは、TransactionRequestを処理し終えた時に応答者によって発行される。

TransactionRequestは以下の場合に処理される。

- TransactionRequestに含まれる全てのactionが処理されたとき、

または

- オプショナルのコマンドでエラーが発生した場合を除き、TransactionRequestを処理した際にエラーが発生したとき

メッセージが受信側で処理できる以上のTransactionRequestを含む場合、受信側はErrorディスクリプタ413(メッセージ中のtransaction数が制限値超過)を返す。

コマンドに含まれる全てのディスクリプタの処理が完了した時点で、そのコマンドの処理は完了する。

SignalsDescriptorはディスクリプタが構文上有効であることが確認された場合、要求されたシグナルはサポートされ、そしてキューから順次実行されたとみなされる。

EventsDescriptorまたはEventBufferDescriptorは、ディスクリプタが構文上有効であることが確認された時、もしくは要求されたイベントを観測することができた時、いずれかの埋め込み型シグナルを生成することができた時、いずれかの埋め込み型イベントを検出することができた時、そしてMGがイベントを検出できる状態へ遷移した時、処理されたとみなされる。

```
TransactionReply(TransactionID {
    ContextID { Response _ Response },
    ...
    ContextID { Response _ Response } })
```

TransactionIDパラメータは対応するTransactionRequestのTransactionIDと同じでなければならない。

ContextIDパラメータはそのActionに対する全てのResponseに対応する一つの値を指定しなければならない。ContextIDは特定の値であってもALLまたはNULLであってもよい。

各Responseパラメータは節7.2にて述べられる一つのリターン値、またはもしそのコマンドの実行がエラーに遭遇したなら一つのErrorディスクリプタに相当する。失敗の時点以降、コマンドは処理されず、そしてそれ故そのコマンドに対してResponseは発行されない。

もしTransactionRequest内でコマンドがオプショナルとマークされていたなら、上記事項に対する例外が発生する。もしオプションのコマンドがあるエラーが発生しても、そのTransactionはそれでもなお実行し続け、そのため、この場合には、エラーの後でもReplyにはResponseが有ることになる。

節7.1.20 “Errorディスクリプタ”は、errorディスクリプタの生成を規定している。以下はいくつかの個々のケースについて論じている。

もし受信側があるContextIDを構文解析できたが、その処理にてあるエラーに遭遇するなら、要求されたactionの応答はContextIDと一つのErrorディスクリプタ、422 Syntax Error in Action(Action内の構文エラー)からなることになる。もし受信側があるContextIDを構文解析できないなら、要求されたactionの応答はTransactionIDと一つのErrorディスクリプタ、422 Syntax Error in Action(Action内の構文エラー)からなることになる。

もし受信側が合法的なactionを決定することができないようなエラーに遭遇するなら、TransactionIDと一つのErrorディスクリプタ、422 Syntax Error in Action(Action内の構文エラー)からなるTransactionReplyを返却するだろう。もしあるactionの終わりが確実に決定されることができないが一つ以上のコマンドが構文解析されることができるなら、受信側はそのコマンドを処理し、その時そのtransactionに対する最後のactionとして422 Syntax Error in Action(Action内の構文エラー)を送ることになる。もし受信側が合法的なtransactionを決定することができないようなエラーに遭遇するなら、受信側は一つのNULL Transaction IDと一つのErrorディスクリプタ(403 Syntax Error in Transaction(Transaction内の構文エラー))を持つTransactionReplyを返却することになる。

もしあるTransactionの終わりが確実に決定される事ができず、一つ以上のactionが構文解析されることができると、受信側はそのactionを処理し、その時そのTransactionに対する最後のactionのReplyとして403 Syntax Error in Transaction(Transaction内の構文エラー)を返却することになる。もし構文解析されることができないactionがないなら、Replyとして403 Syntax Error in Transaction(Transaction内の構文エラー)を返却することになる。

もしTerminationIDが確実に決定されることができないなら、受信側はactionのReplyとして442 Syntax Error in Command(コマンド内の構文エラー)を送信することになる。

もしあるコマンドの終わりが確実に決定されることができないなら、受信側は構文解析可能な最後のActionへのReplyとして442 Syntax Error in Command(コマンド内の構文エラー)を返却することになる。

### 8.2.3 TransactionPending

TransactionPendingは受信者によって発行される。TransactionPendingは、transactionの処理は続行されているが完了はしていないことを示す。TransactionPendingは、transactionが完了するまで時間がしばらくかかるであろう場合に、送信側がTransactionRequestが失われたと判断してしまうことを防ぐために使用される。

TransactionPending(TransactionID { })

TransactionIDパラメータは対応するTransactionRequestのTransactionIDと同一でなければならない。MGCは、MGCから発行されたいかなるトランザクションに関してもMGが応答を返してくると期待している時間（網による遅延を除く）を、Root Terminationのプロパティ(normalMGExecutionTime)に設定することができる。MGCは、MGが発行したいかなるトランザクションに関してもMGCが応答を返すと予定している時間（網による遅延を除く）を、他のプロパティ(normalMGCEExecutionTime)に設定することができる。MGProvisionalResponseTimerValueは、もしtransactionが完了しない場合に、MGCがMGからPending Responseをその時間内に受信することを期待する時間（初期状態では、normalMGExecutionTimeに網による遅延を加えた値に設定されるが、より短く設定されるかもしれない）を示す。MGCProvisionalResponseTimerValueは、同様にMGがMGCに対してPending Responseが返されることを期待する時間を示す。送信側は一つのコマンドに対して一つ以上のTransactionPendingを受信することがある。もしPending中に重複したRequestが受信された場合は、応答側は重複したPendingを直ちに送ってもよいし、あるいは次のTransactionPendingを引き起こすタイマを待ち続けてもよい。

MGから受け取ることができるTransactionPendingの数を示すために、MGCはMGのRoot Terminationのプロパティ(MGOriginatedPendingLimit)を設定することができる。このプロパティで示される時間を超過する場合、MGはトランザクション処理を止めて、TransactionReplyを返さなければならない。そうでない場合、MGCはトランザクション処理にエラーが発生していると判断することができる。

MGCから受け取り得るTransactionPendingの数を示すために、MGCはMGのRoot Terminationの異なるプロパティ(MGCOriginatedPendingLimit)を設定することができる。このプロパティで示される時間を超過する場合、

MGCはトランザクション処理を止めて、TransactionReplyを返さなければならない。そうでない場合、MGはトランザクション処理にエラーが発生していると判断することができる。

コマンド処理に長く時間がかかったために、あるいはエラーが発生したために(例えば、コマンドがループを引き起こした場合)、xxxOriginatedPendingLimit(MGOriginatedPendingLimit またはMGCOrganizedPendingLimit)を超過してしまうことがあり得る。いずれの場合においても、オリジナルのTransactionRequestの受信側が、コマンド処理に時間がかかった原因のコマンド、あるいはエラーを引き起こしたコマンドに対応する応答のパラメータとして、Errorディスクリプタを含むTransactionReplyを発行することになる。Transaction中のそれ以降のコマンドは処理されてはならない。エラー 506 Number of TransactionPending Exceeded(TransactionPending数超過)が使用されなければならない。

注記：エラーが原因で xxxOriginatedPendingLimit (MGOriginatedPendingLimit またはMGCOrganizedPendingLimit) を超過し、オリジナルのTransactionRequestの受信側がTransactionPendingを送り続ける状態を防止するために、オリジナルのTransactionRequestの受信側は適切な回復アクションを引き起こすための管理保護メカニズムを実装すべきである。オリジナルのTransactionRequestの送信側は受信したPendingの数を監視しておき、回復のための処理を開始してもよい。

### 8.3 メッセージ

一つのメッセージ中には、複数のtransactionを連結して記述することができる。各メッセージはそれぞれ一つのヘッダを持っており、ヘッダは送信側の識別情報を含む。メッセージ中のメッセージ識別子(MID)には、そのメッセージを伝送するエンティティが持つ、プロビジョニングによって決定された名前(例えばドメインアドレス/ドメイン名/デバイス名)が設定される。推奨されるデフォルト値はドメイン名である。H.248.1エンティティ(MG/MGC)は、通信相手(MGC/MG)との制御関係期間中に生成するすべてのメッセージの中で同一のMIDを一貫して使用しなければならない。

全てのメッセージはそのメッセージが準拠するH.248のバージョンを識別するバージョン番号を含む。バージョンは一つまたは二つの数字からなり、バージョン1から始まる。プロトコルの現行版はバージョン3である。

一つのメッセージ内の複数のTransactionは独立に扱われる。暗黙に規定される順序はなく、アプリケーションまたはH.248プロトコルのレベルにおけるメッセージ受信確認もない。メッセージは本質的にトランスポートメカニズムである。例えば、A、B、CのTransaction Requestを含んでいるメッセージXは、AとCへのReplyを含んでいるメッセージYとBへのReplyを含んでいるメッセージZで応答されてもよい。同様に、Request Dを含んでいるメッセージLとRequest Eを含んでいるメッセージMは、DとEの両方へのReplyを含んでいるメッセージNで応答されてもよい。

## 9. トランスポート

本プロトコルのためのトランスポートメカニズムはMGCとMG間のTransactionの確実な転送ができるようにすべきである。トランスポートはいずれの特定のコマンドが送信されているかについては無関係のままでなければならない。また全てのアプリケーション状態に適合可能でなければならない。本プロトコル用に定義されるいくつかのトランスポートがあり、本標準の付属資料および個別のH.248勧告(つまりH.248.4およびH.248.5)にて定義される。追加のトランスポートが追加のH.248勧告として定義されてもよい。IP上での本プロトコルの転送のために、MGCはTCPとUDP/ALFの両方を実装しなければならない。MGはTCPまたはUDP/ALFもしくはその両方を実装しなければならない。

MGは一つのPrimary-MGCおよび0個以上のSecondary-MGC(節7.2.8を参照)の(DNS名またはIPアドレスのような)名前またはアドレスを事前に供給される。これはMGがMGCへメッセージを送信するために使用するア

ドレスである。もしプロトコルのトランスポートとしてTCPまたはUDPが使用され、かつ最初のServiceChangeRequestが送られるべきポートが知られていないなら、その要求はそのプロトコルに対するデフォルトのポート番号へ送られるべきである。このポート番号はUDPまたはTCPのどちらに対してもテキスト符号化オペレーションには2944あるいはバイナリ符号化オペレーションには2945である。MGCはMGからServiceChangeRequestを含んでいるメッセージを受け、そのメッセージからMGのアドレスを決定することができる。節7.2.8で記述されているように、MGまたはMGCはServiceChangeAddressパラメータにて、その後のTransactionRequestが送られなくてはならないアドレスを供給してもよい。しかし応答(最初のServiceChangeRequestへの応答を含む)は常に対応するRequestの発信元であったアドレスへ送り返されなければならない。例えば、IPネットワークでは、IPヘッダ中の発信元アドレス、およびTCP/UDP/SCTPヘッダ中の発信元ポート番号である。

## 9.1 コマンド順序

本標準は下位トランスポートプロトコルがエンティティへ送られるTransactionの順序を保証することを必須としていない。この特性はActionの適時性を最大化する傾向があるが、少しの欠点がある。例えば、

Notifyコマンドが遅延させられ、該EventsDescriptorを変更する新たなコマンドの送信後にMGCに到着することがある。

もし以前のコマンドが受信確認される前に新たなコマンドが送信されるなら、新たなコマンドの前に以前のコマンドが実行される保証はない。

MGの矛盾のないオペレーションを保証したいMGCは以下の規則を使用してもよい。これらの規則は異なるTransaction内にあるコマンドに関するものである。同じTransaction内のコマンドは順序正しく実行される(8章を参照)。

1. MGがいくつかのterminationを扱う時、異なるterminationに関係しているコマンドは並列に送られてもよい。例えば各termination(またはterminationのグループ)が各自のプロセスまたはスレッドにより制御されるモデルに従って。
2. 一つのterminationについて、複数の未解決コマンド(Add または Modify または Move)が同一Transactionにある場合を除き、未解決コマンドは通常最大でも一つまで存在するようにすべきである。しかし Subtract コマンドはいつ発行されてよい。結果として MG は以前除去された termination に適用する Modify コマンドを時々受けるかもしれない。そのようなコマンドは無視されるべきであり、かつエラーコードが返却されるべきである。
3. termination において、メッセージの配達順を保証しないトランスポート(つまり UDP)のために、通常未解決の Notify コマンドはいつでも多くても一つとすべきである。
4. 時々terminationのグループに適用する暗示的にまたは明示的にワイルドカード化された Subtract コマンドが保留中の Add コマンドの前に入り込むことがある。MGCは、全体的な Subtract コマンドの時点で保留されていた Add コマンドに関連する全ての termination を個別に削除すべきである。またワイルドカードにより指名される(あるいは Multiplex ディスクリプタ内で暗に示される)termination に対する新たな Add コマンドは、ワイルドカード化された Subtract コマンドが受信確認されるまでは送られるべきではない。
5. AuditValue と AuditCapability コマンドはいかなる順序でも問題無い。
6. ServiceChange コマンドはリスタート手順により定義されるように常に MG により送信される最初のコマンドでなければならない。あらゆる他のコマンドまたは応答はこの ServiceChange コマンドの後に送られなければならない。

これらの規則はコマンド応答側には影響せず、したがってコマンド応答側は常にコマンドに応答すべきである。

## 9.2 リスタート殺到の防止

非常に多数のMGが同時に電源投入され、それらがみなServiceChange Transactionを起動する事象においては、MGCは非常に圧迫される可能性があり、サービス回復の危機的期間の間、メッセージ紛失やネットワーク輻輳を導くことになる。そのようなリスタート殺到を防ぐために以下の動作が推奨される。

1. MGの電源が投入される時、MGは0と最大待ち遅延(maximum waiting delay: MWD)の間に均等に分布された乱数値でリスタートタイマを初期化すべきである。同じアルゴリズムを使用するであろう複数MG間での乱数発生の同期性を避けるために注意が払われるべきである。
2. それからMGはこのタイマの満了、または例えばレジデンシャルMG上でのオフフック遷移のようなローカルユーザの活動の検出のどちらかを待つべきである。
3. そのタイマが経過するかまたはある活動が検出される時、MGはリスタート手順を開始すべきである。

リスタート手順は、MGCがこのMGから見える最初のメッセージが、MGCにそのリスタートについて知らせているServiceChangeメッセージであるということを、MGが保証することを単に要求する。

注記： MWDの値はMGの種類に依存するコンフィグレーションパラメータである。以下の論証がレジデンシャルゲートウェイに関するこの遅延の値を決定するために使用されてもよい。

MGCは一般的にピーク時間のトラフィック負荷を処理するように規模が計られている。ピーク時間の間は、一般的に、平均保留時間が3分である呼を収容して、平均で回線の10%がビジーになる。呼処理は各MGとMGCの間で概して5から6個のMGC Transactionが関係する。この単純計算はMGCが平均で30分毎に各Terminationに対し5から6個のTransaction、換言すると平均で5から6分毎にTermination当たり約1個のTransactionを処理するよう期待されていることを示している。このことはレジデンシャルゲートウェイに対する理にかなったMWDの値は10から12分になることを示している。構成が明示されない場合には、レジデンシャルゲートウェイはMWDに600秒という値を採用すべきである。

同じ論証はトランクゲートウェイあるいはビジネスゲートウェイに対してはMWDの値はより短くあるべきと暗示している。何故ならそれらのゲートウェイは非常に多数のTerminationを扱い、またこれらのTerminationの使用率は最繁時間帯の間10%より高く、典型的な値は60%であるからである。ピーク時間の間、これらのTerminationはMGCの負荷に対して毎分約1 Transactionを提供することが見込まれている。理にかなったアルゴリズムは「トランク」Termination毎のMWDの値をレジデンシャルゲートウェイ毎のMWDより6倍短くし、またリスタートされているTerminationの数に反比例にするものである。例えばMWDはT1回線を扱うゲートウェイについては2.5秒に、あるいはT3回線を扱うゲートウェイについては60ミリ秒に設定されるべきである。

## 9.3 Notify (通知) 殺到の防止

送信障害、あるいはMGが短時間に多くのイベントを認識したために、notification (通知) の実質的な数が蓄積しているMGで、その状況が発生する場合、残りの処理が完了するまで、MGは制限された方法でnotification (通知) を送信するべきである。

## 10. セキュリティ

本章ではIP環境で本プロトコルを使用するときのセキュリティをカバーする。

### 10.1 プロトコルコネクションの保護

セキュリティメカニズムは、権限のないエンティティが、権限のない呼をセットアップしたり認可された

呼を妨害する目的で、本標準で定義されるプロトコルの使用を防ぐために明らかに必要とされる。IPネットワーク上で転送される時の本プロトコルのためのセキュリティメカニズムはIPsec [RFC2401～RFC2411]である。

AHヘッダ[RFC2402]はMGとMGCの間でやり取りされるメッセージのデータ送信元認証(data origin authentication)、コネクションレスインテグリティ(connectionless integrity)、そしてオプションであるリプレイ防御保護(anti-replay protection)を提供する。ESPヘッダ[RFC2406]は望まれているならば、メッセージの機密性(confidentiality)を提供する。例えば、セッション記述がSDPで定義されるようなセッションキーを運ぶのに使用されるならば、ESP暗号化サービスが要求されるべきである。

ESPヘッダを使う本標準に定義されるプロトコルの実装は[RFC2406]の5章に従わなければならない、そしてそれはインテグリティチェック(integrity checking)と暗号化のための最小セットのアルゴリズムを定義する。同様にAHヘッダを使う実装は[RFC2402]の5章に従わなければならない、そしてそれはマニュアルキーを使うインテグリティチェックのための最小セットのアルゴリズムを定義する。

実装はより強固なキーイングオプションを許容するためにIKE[RFC2409]を使用すべきである。IKEを用いた実装はRSAシグネチャとRSA公開鍵暗号化で認証をサポートするべきである。

## 10.2 暫定AHスキーム

IPsecの実装は、AHまたはESPヘッダをIPヘッダの直後に挿入する必要がある。これはアプリケーションレベルでは容易にできない。したがって、下位ネットワーク実装がIPsecをサポートしない本標準で定義されるプロトコルの展開問題(deployment problem)を提示する。

暫定の解決策として、オプションのAHヘッダはJT-H248.1プロトコルヘッダの中に定義される。ヘッダフィールドはまさに[RFC2402]に定義されるSPI、SEQUENCE NUMBERおよびDATAフィールドのものである。ヘッダフィールドの意味は、インテグリティチェック値(Integrity Check value : ICV)の計算を除いて、[RFC2402]の"transport mode"と同じである。IPsecでは、ICVはIPヘッダを含む全体のIPパケットに渡って計算される。これは、IPアドレスの詐称を防ぐ。同じ機能性を保有するために、ICV計算は20桁の16進数として符号化された32ビットの発信元IPアドレス、32ビットの宛先アドレスおよび16ビットのUDP宛先ポートからなる統合されたIPヘッダによって事前に待たされたメッセージ中の(連結された)全てのTransactionに渡って実行されるべきである。TCPがトランスポート層であるときに、暫定のAHメカニズムを採用しているとき、上のUDPポートはTCPポートになり、他の全ての操作は同じである。

JT-H248.1プロトコルの実装はIPsecを実装しなければならない。このとき土台となるオペレーティングシステムとトランスポートネットワークはIPsecをサポートする。IPv4を使用するプロトコルの実装は暫定のAHスキームを実装しなければならない。しかし、この暫定スキームは下位ネットワーク層がIPsecをサポートするときには使用してはならない。IPv6実装はIPsecをサポートすると仮定されており、暫定AHスキームを使用してはならない。

暫定AHメカニズムの全ての実装はマニュアルキーを使用するインテグリティチェックのための最小セットのアルゴリズムを定義する[RFC2402]の5章を満足する。

暫定AHスキームは盗聴(例えば、与えられたterminationでセットアップしたコネクションをモニタする悪意に満ちた第三者)に対する保護を提供しない。また、それはリプライ攻撃に対する保護も提供しない。これらの手順は不正を働くMGまたは不正を働くMGCによるサービス拒否(denial of service)攻撃に対して必ずしも保護するわけではない。しかし、こういった不正を働くエンティティの識別を提供するだろうし、そしてそのエンティティはメンテナンス手順を通して認可を奪われるべきである。

## 10.3 メディアコネクションの保護

本プロトコルは盗聴を保護し、オーディオメッセージの暗号化に使用することができる「セッションキー」

をMGCがMGに提供することを可能にする。

パケットネットワーク特有の問題は、"uncontrolled barge-in"(コントロールできない入り込み)である。この攻撃はコネクションによって使用されるIPアドレスとUDPポートにメディアパケットを送ることによって実行することができる。どんな保護も実装されないならば、パケットは減速されなければならない、信号は"line side"(回線側)に向けられなければならない。

この攻撃に対する基本的な保護は、既知の発信元からのパケットを受け入れるだけである。例えば、IP発信元アドレスとUDP発信元ポートがRemoteディスクリプタで知らされた値と合致するのをチェックする。これには、二つの不便がある。それはコネクション確立を遅らせることと、発信元詐称によってだまされることができることである。

- アドレススペースの保護を可能にするために、MGCは出側MGのリモートセッション記述を得て、それを入側MGに渡さなければならない。これは少なくとも一回のネットワークの往復を必要とし、我々に次のような二者択一を残す。往復の完了を待たずに呼を処理して、例えば、リモートアナウンスを早めに切り上げる危険にさらすか、もしくは完全な往復を待ち、より遅い呼設定手順を受け入れるかのどちらか。
- 攻撃者が例えばトラフィックの一部を盗聴することによって、発信元、宛先アドレスとポートの有効な組合せを入手することができるならば、発信元詐称だけが有効である。発信元詐称に対抗するために、ネットワークへの全てのアクセスポイントを制御しようとすることになる。しかしこれは実際には非常に達成し難い。

発信元アドレスをチェックすることに対する代替手段は、呼設定手順の間に運ばれる秘密キーを使用して、パケットを暗号化して、認証することである。これは呼設定を減速せず、アドレス詐称に対する強い保護を提供することになる。

## 11.1. MG-MGC制御インタフェース

MGとMGC間の制御関係はMGのコールドスタートで開始され、ServiceChangeメッセージによってアナウンスされるが、故障やマニュアルサービスイベントのような、その後のイベントによって変更することができる。

注記：本プロトコルは一つの物理MGを制御する複数MGCをサポートする明確なメカニズムを持っていないが、一方、異なるMGCと関係することができる(一つの物理MG内の)複数の論理MGをサポートするようにプロトコルを設計してある。

### 11.1.1 複数仮想MG

一つの物理MGは一つ以上の仮想MG (VMGs) に仕切られるかもしれない。仮想MGは静的に仕切られた物理terminationのセットとephemeral terminationのセットとの両方または何れかからなる。物理terminationは一つのMGCによって制御される。このモデルは他のリソースが静的に割り当てられること、まさにterminationを必要としない。仮想MGにterminationを割り当てるためのメカニズムは本プロトコルの範囲外の管理方法である。仮想MGのそれぞれは完全なMGクライアントとしてMGCに見える。

物理MGは、ただ一つのネットワークインタフェースしか持たなくてもよい。それは仮想MGにまたがって共有されなければならない。そのような場合、パケット/セル側のterminationは共有される。しかし、それは注意されるべきで、使用中にそのようなインタフェースはterminationのephemeralインスタンスがフロー毎に作成されることを要求する。そしてその結果、terminationを共有することは容易である。このメカニズムは複雑さを導く、すなわち、MGがそのインタフェースにイベントが発生した場合に、その制御しているMGCのどれに通知すべきかを必ず知らねばならない。

通常の操作では、仮想MGは(それが起点側であるならば)ネットワークフローを作成することを、または(そ



れが終点側であるならば)フロー要求を待つようにMGCによって命令される。そしてどんな混乱も起こらないだろう。しかしながら、予期されないイベントが起こるならば、仮想MGは、それが制御している物理的なリソースに関して何をしたらよいのかを知らなければならない。

イベントから回復するために物理的なインタフェースの状態の操作を必要とするならば、一つのMGCだけがそうすべきである。MGCのどれかがそのようなイベントについて通知されるためにEventsDescriptorsを作成するのを許容することによって、これらの問題は解決されるが、一つのMGCだけが物理的なインタフェースプロパティにread/writeアクセスを持つことができる。他の全てのMGCには、read-onlyアクセスがある。管理機構が、どのMGCがread/write能力を持つかを指名するように使用されて、Master MGCに指定される。

各仮想MGは、それ自身のRoot Terminationを持つ。多くの場合、Root Terminationのプロパティのための値は各MGCで個別に設定できる。一つの値だけ存在することができ、そのパラメータはMaster MGCを除いた全てのMGCでread-onlyである。

ServiceChangeはterminationに適用されるか、仮想MGに仕切られたterminationについて設定されるか、またはその仮想MGによって(ephemeral terminationの場合に)作成されるだけでもよい。

## 11.2 コールドスタート

MGにはPrimary-MGCと(オプションで)Secondary-MGCの順序だったリストが、このプロトコルの範囲外の管理機構によって事前に用意される。MGのコールドスタートでは、MGはそのPrimary-MGCへのRoot Terminationの"Restart"メソッドを持つServiceChangeコマンドを発行することになる。MGCがMGを受け入れると、MGCはServiceChangeMgcIDパラメータを含まないTransactionReplyを送信する。MGCがMGの登録を受け付けられない場合は、ServiceChangeMgcIDパラメータを含むTransactionReplyを送信することによって、MGがコンタクトすべき他のMGCのアドレスを通知する。

MGはServiceChangeMgcIDパラメータを含むTransactionReplyを受信すると、ServiceChangeMgcIdによって指定されたMGCに対してServiceChangeを送信する。この動作は、登録を受け付ける制御用MGCを得るまで、またはReplyを得るのを失敗するまで続けられる。Primary-MGC、または指定された後任MGCからReplyを得ることに失敗した場合、MGは事前に用意されたSecondary-MGCに対して順次接続を試みる。MGがいずれのMGCとの制御関係も確立することができないならば、それは節9.2で記述されるランダムな時間を待って、それから再びそのPrimary-MGC、および必要であればSecondary-MGCとコンタクトを開始しなければならない。

RestartのServiceChangeに対するReplyが失われて、コマンドがServiceChange応答の受信前にMGによって受信される可能性がある。MGはエラーコード505 “Command Received before Restart Response”(Restart応答前のコマンド受信)を発行しなければならない。

## 11.3 プロトコルバージョンの交渉

あるMGCに登録されるあるMGからのServiceChangeコマンドは、ServiceChangeVersionパラメータ内にそのMGがサポートするプロトコルのバージョン番号を含まなければならない。そのServiceChangeVersionパラメータに設定されるバージョンに関わらず、そのコマンドを含むメッセージはバージョン1のメッセージとして符号化されなければならない。MGCはそのようなメッセージを受け取ると、より低いバージョンだけをサポートするならば、そのより低いバージョンを持つServiceChangeReplyを送るものとする。そしてその後、MGとMGCの間の全てのメッセージがより低いバージョンのプロトコルに従うものとする。もしそのMGがこれに従うことができず、既にMGCとの伝送接続を確立しているならば、その接続をクローズすべきである。そして、MGは、いずれのイベントにおいても、MGCからの全ての後続する要求をエラーコード406 – “Version Not Supported”(バージョン未サポート)で拒絶すべきである。

MGCがMGより高いバージョンのみをサポートするが、MGから提案されたより低いバージョンをサポート

トすることができるならば、そのより低いバージョンを持つServiceChangeReplyを送るものとする。そしてその後、MGとMGCの間の全てのメッセージがより低いバージョンのプロトコルに従うものとする。MGCがこれに従うことができないならば、エラーコード406 – “Version Not Supported”(バージョン未サポート)で関係を拒絶しなければならない。

プロトコルバージョンの交渉はまた"Handoff"と"Failover"のServiceChangesにおいて起こることがある。プロトコルを新規バージョンに拡張する際には、以下の規則に従うべきである。

1. 現存するプロトコル要素、すなわち手順、パラメータ、ディスクリプタ、プロパティ、値は、プロトコルの誤りを修正する必要がある場合、またはプロトコルによってサポートされているサービスの手順を変更する必要がある場合を除き、変更されるべきではない。
2. コマンド、パラメータ、ディスクリプタ、プロパティ、値の意味は変えられるべきでない。
3. メッセージやパラメータのフォーマット法と符号化法の確立された規則は変更すべきではない。
4. 情報要素が旧式になったときは、未使用としてマークすることができる。しかし、その情報要素のための識別子は、予備としてマークされることになる。このように、将来のバージョンで使われることはできない。

#### 11.4 MG故障

あるMGが故障したが、MGCにメッセージを送ることができる場合、それは適切なメソッド(GracefulまたはForced)を持つServiceChangeを送って、そのRoot TerminationIDを指定する。そしてMGがサービスに戻るとき、"Restart"メソッドを持つServiceChangeを送る。

MGCがMGの両方に重複したメッセージを送ることを許容することは、MGの一つに冗長failoverの能力があるMGのペアを収容できることである。動作中のMGのみがTransactionを受け入れるか、または拒絶しなければならない。Failover時には、Primary-MGは"Failover"メソッドと"MG Impending Failure"理由をもつServiceChangeコマンドを送る。そして、MGCは有効なMGとしてSecondary-MGを使用する。エラー状態が修復されるとき、動作中のMGは"Restart"メソッドをもつ"ServiceChange"を送ることができる。

注記： 冗長failover MGは信頼性のある転送を必要とする。何故なら、本プロトコルは、ALFが動作中のあるSecondary-MGに対して、そのMGCから送信されたメッセージを通知する手段を持たないためである。

#### 11.5 MGC故障

MGがその制御用MGCの故障を検知するならば、その事前に用意されたリストの次のMGCにコンタクトを試みる。MGは、故障が検知されたのがPrimary-MGCでなければ、初め(Primary-MGC)からその試行を開始し、故障が検知されたのがPrimary-MGCであれば、最初のSecondary-MGCから開始する。MGは"Failover"メソッドと"MG Impending Failure"理由を持つServiceChangeメッセージを送る。もしMGがどのMGCとも制御関係を確立できない場合、そのMGは節9.2に記述されるように、ランダムな時間を待ち、それからPrimary-MGC、そして必要があればSecondary-MGCに対して再度接続を始めなくてはならない。それ以前に制御用MGCに接続している時は、MGは"Disconnected"メソッドを持つServiceChangeメッセージを送信する。

MGCは、部分的な故障の際、または手動保守の目的で、自分が制御するMGに対して、別のMGCを使用するように指示することを望むことがある。MGCは、そうするために、"HandOff"メソッドおよびServiceChangeMgcIDにその指定された代替(MGC)を持つServiceChangeメッセージをMGに対して送らなければならない。MGが"Handoff"をサポートしている場合、MGは指定されたMGCに"Failover"メソッドと"MG directed change"理由をもつServiceChangeメッセージを送らなければならない。指定されたMGCからのReply

を得るのを失敗した場合、そのMGはあたかもそのMGCが故障しているかのように振る舞って、上述するようにSecondary-MGCにコンタクトし始めなければならない。MGはいずれのMGCとも制御関係を確立できないならば、節9.2で記述されるランダムな時間を待った後、再びそのPrimary-MGC、そして必要ならばそのSecondary-MGCとのコンタクトを始めなければならない。

Handoffに関わるMGCがどのように状態情報を保守するかについてはどんな勧告も規定されていない。これはこの勧告の範囲外と考えられる。Handoff発生時、MGCとMGは以下の方法を取ってもよい。MGCがHandOffを開始するとき、そのハンドオーバーはMG上のオペレーションにとって明快であるべきである。Transactionは順不同で実行することができ、ServiceChangeが実行される時に進行中でもよい。従って、進行中のコマンドは継続実行し、発行元MGCからの全コマンドに対する応答がそのコマンドの送信元である転送アドレスに送信されなければならない。もし、送信中のMGCとのサービス関係が終了しているなら、その応答は廃棄されるべきである。またMGは、新しいMGCから未完了のTransaction Replyを受信してもよい。制御関係が設立されるまではどんな新しいメッセージも新しいMGCに送られてはならない。繰り返されたTransaction Requestは新しいMGCに向けられなければならない。MGは全てのterminationとContextの状態を維持しなければならない。

故障したMGCが動作しているMGCに置き換わり、この際新たなMGCのIDが故障したものと同一であるといったMGCの実装が可能である。このような場合、ServiceChangeMgcIDは以前の値で指定されるだろう。そして、MGはあたかも値が変えられたかのように振る舞い、上述のようにServiceChangeメッセージを送らなければならない。

冗長failoverの可能なMGCのペアは上述のメカニズムによって、制御対象のMGに対してそのfailoverを通知することができる。

## 11.6 MGC-MG 間制御リンク監視

MGC-MG制御関係をモニターすることは、高可用性ネットワークにとって不可欠である。連続的にMGC-MG相互接続（リンク状態）をチェックすることによって解決してもよい。多くの事業者はこの機能性を提供するので、プロトコルメカニズムは不必要である。

リンク状態監視を提供しない事業者のために、この機能性は、既存にメッセージを送ることを利用することによって、H.248プロトコルで解決できる。メッセージを送っているどんなH.248でもない場合、MGCはMGで通信のロスを見つけるために、空のAudit Descriptorでルートの上でAuditValueコマンドを使用してもよい。MGは、Inactivity Timer Package（ITU-T Recommendation H.248.14）の使用を通して、通信損失を見つけることができる。

## 12. パッケージ定義

拡張のための基本機構はパッケージにより実現される。パッケージは、Termination上、Context上、イベント上で発生し得る追加のプロパティ、Termination上で発生し得るイベント、シグナル、統計情報を定義する。

IETFで定義されるパッケージは別のRFCで発行されることになっている。

ITU-Tで定義されるパッケージは関連する勧告（例えば、H.248サブシリーズ）で発行されてもよい。

1. 公開文書または標準化機関の文書は、上記ガイドラインに従ってパッケージを記述する文書として参照され得る文書であり、詳細な記述がなされているべきである。
2. 文書は記述パッケージのバージョンを規定しなければならない。
3. 文書は公開Webサーバに置かれるべきであり、かつ不変のURLを持つべきである。また、このサイトは文書に対するコメントを受ける機構を持ち、適切なレスポンスを返すことができるようにすべきである。

## 12.1 パッケージ定義のガイドライン

パッケージは、プロパティ、イベント、シグナル、統計情報を定義する。

パッケージは、また、節12.2で説明されるガイドラインに従い、新しいエラーコードを定義してもよい。このガイドラインは文書上の利便性のためにある。パッケージ文書はエラーコードの登録をサポートするIANAに提出される。あるパッケージが変更されたとしても、エラーコードの記述そのものが変更されるのでなければ、そのエラーコードを維持するための新たな文書リファレンスをIANAに提出する必要はない。

定義された構成要素名はPackageID(そのパッケージをユニークに指定するもの)と、項目のID(パッケージ内で項目をユニークに指定するもの)から構成されなくてはならない。テキスト符号化によれば、この二者は、文字"/"で区切られる。例えば”togen/playtone”はtone generationパッケージのプレイトーンシグナルをテキスト符号化したものである。

パッケージは、以下に示す項目、キーワード（ボールドで示された）からなる。パッケージ規定のテンプレートは、Appendix IIIに含まれる。

### 12.1.1 Package

パッケージ全体の記述では以下を指定する。

**Package Name:**

記述のみ

**PackageID:**

識別子

**Description:**

Packageの記述

**Version:**

あるPackageの新たなバージョンは、追加のプロパティ、イベント、シグナル、統計情報と、元のパッケージに記述されている既存パラメータのための新たな取り得る値(Possible Values)を追加することのみができる。削除と変更はできない。Versionは、1から99までの整数。

**Designed to be extended only (オプション): Yes**

これは、このパッケージが直接参照されるのではなく、明確に他のパッケージによって拡張されるために設計されているものであるかどうかを示す。例えば、このパッケージはそれ自身に何の機能もない、あるいは、それ自身では何の意味も持たなくてもよい。MGは、パッケージを報告する際には、このようなPackageIDを公開すべきではない。

**Extends (オプション):**

パッケージは、既存パッケージを拡張してもよい。元のパッケージのVersionが指定されなければならない。あるパッケージが他のパッケージを拡張する際、追加のプロパティ、イベント、シグナル、統計情報と、元のパッケージに記述されている既存パラメータのための新たな取り得る値(Possible Values)を追加することのみができる。拡張されたパッケージは、元のパッケージおよび元のパッケージが拡張された(複数レベルの拡張)パッケージの中で定義されている識別子を再定義したり、上書きしたりしてはならない。従って、パッケージBのバージョン1がパッケージAのバージョン1を拡張している場合、Aのバージョン2が既にBのバージョン1で使用されているある名称を定義しているとすれば、Bのバージョン2はAのバージョン2を拡張できないことになる。

### 12.1.2 Properties

パッケージによって定義されるプロパティは、以下を指定する。

Property Name:

記述のみ

PropertyID:

識別子

Description: プロパティ機能説明

Type:

以下の何れか1つである。

Boolean

String:

UTF-8文字列

Octet String:

オクテット数。符号化法については付属資料Aおよび付属資料B.3を参照。

Integer:

4バイト符号付き整数

Double:

8バイト符号付き整数

Character:

Unicode UTF-8符号化された1文字(1オクテット以上となる)

Enumeration:

取り得るユニークな値のリストの一つ

Sub-list:

あるリストの中から複数の値を取り出したリスト。**Sub-list**の**Type**もまた指定されなければならない。その**Type**は(sub-listの例外と共に)この節で指定される**Type**から選ばなければならない。例えば、”Type: sub-list of enumeration”。Sub-listの符号化法については付属資料Aおよび付属資料B.3を参照。

Possible Values:

1つのパッケージは、特定の値の組か、値がどのように決定されるかの説明を指定しなければならない。パッケージはまた、デフォルト値か、デフォルト値がそのディスクリプタから省略される場合のデフォルト動作を指定しなければならない。例えば、パッケージは、デフォルト値が省略された場合、そのプロパティに関係する手順が停止されるということ指定してもよい。デフォルト値(手順ではない)は、事前に用意されているものとして指定されてもよい。

Defined in

このプロパティが定義されているJT-H248.1 ディスクリプタ。LocalControlはストリームに依存するプロパティである。TerminationStateはストリームに依存しないプロパティである。殆どの場合はこちらであると考えられるが、他のディスクリプタで定義されるプロパティである可能性もある。

Characteristics: Read / Write または両方、および(オプションで)global

プロパティがリードオンリかリード・ライト可能か、グローバルかを示す。globalが省略される場合は、そのプロパティはグローバルではない。プロパティがグローバルである場合、そのプロパティの値を、そのパッケージを実現する全てのTerminationが共有する。

### 12.1.3 Events

パッケージによって定義されるイベントは、以下を指定する。

Event name:

記述のみ

EventID:

識別子

Description:

EventsDescriptor Parameters:

そのイベントを形成するMGCによって使用されるパラメータであり、EventsDescriptor内に指定される。節12.2参照。EventsDescriptorにパラメータ無いなら「none」を指定すべき。

ObservedEventsDescriptor Parameters

Notify要求中のMGCに返されるパラメータと、MGCからのOvserveEventsDescriptor監査要求に対する応答の中のパラメータ。これらは、OvserveEventsDescriptor内に指定される。節12.2参照。OvserveEventsDescriptorにパラメータがないなら「none」を指定すべき。

#### 12.1.4 Signals

パッケージによって定義されるシグナルは、以下を指定する。

Signal Name:記述のみ

SignalID:識別子。SignalIDは、SignalsDescriptor内で使用される。

Description: シグナルの機能説明

SignalType:

SignalTypeは以下の何れか1つである。

OO (On/Off)

TO (TimeOut)

BR (Brief)

注記：SignalTypeは一つ以上のパラメータ値に依存して定義されてもよい。パッケージはデフォルトのシグナルタイプを指定しなければならない。そのデフォルトのタイプがTOである場合、パッケージはデフォルトのDurationを指定しなければならない。このDurationは事前に用意されたものでもよい。デフォルトDurationはBRに対しては意味を持たない。

Duration:

100分の1秒単位

Additional Parameters:

節12.2参照

#### 12.1.5 Statistics

パッケージによって定義される統計情報は、以下を指定する。

Statistics Name:記述のみ

StatisticsID:識別子。StatisticsIDは、StatisticsDescriptor内で使用される。

Description:統計情報の機能説明

Type: 以下いずれか1つである。

Boolean

String: UTF-8文字

Octet String: オクテット数。符号化法については付属資料Aおよび付属資料B.3を参照。

Integer: 4バイト符号付き整数

Double: 8バイト符号付き整数

Character: Unicode UTF-8符号化された1文字(1オクテット以上となる)

Enumeration: 可能な一意的な値のリストのうちの1つ。

Sub-list: あるリストの中から複数の値を取り出したリスト。Sub-listのTypeもまた指定されなければならない。そのTypeは(sub-listの例外と共に)この節で指定されるTypeから選ばなければならない。

例えば、”Type: sub-list of enumeration”。Sub-listの符号化法については付属資料Aおよび付属資料B.2を参照。

**Possible Values:**

パッケージは、測定の単位を示さなければならない。例えばミリ秒。そのTypeともに範囲に対する規制も示さねばならない。

**Level:**

統計がterminationレベルかストリームレベルかどちらかに保たれることができるかどうかを指定すること。

### 12.1.6 Error Codes

パッケージが何のエラーコードも定めないならば、このセクションは省略してもよい。さもないければ、それはパッケージによって定義されるエラーコードを記述する。

Error Code #: エラーコード番号

Name: エラー名称

Definition: エラー説明

Error Text in the Error Descriptor:

Error Descriptorでどんなテキストを返すべきかという記載。

Comment: Any further comments on the use of the error code.

### 12.1.7 Procedures

パッケージ使用についての付加的なガイダンスを記述する。

## 12.2 EventsとSignalsに対するパラメータ定義のガイドライン

**Parameter Name:**

記述のみ

**ParameterID:**

識別子。イベントとシグナルに対応するパラメータのParameterIDは、それぞれ”EPA”、”SPA”で始まってはならない。ParameterIDはまた、”ST”、”Stream”、”SY”、”SignalType”、”DR”、”Duration”、”NC”、”NotifyCompletion”、”KA”、”KeepActive”、”EB”、”Embed”、”DM”、および”DigitMap” ”SPADI”、”SPADirection”、”SPARQ” or ”SPARquestID” であってはならない。

**Description:** パラメータ機能説明

**Type:**

以下の何れか一つである。

Boolean

String:

UTF-8 文字列

Octet String:

オクテット文字列。符号化法については付属資料Aおよび付属資料B.3を参照。

Integer:

4バイト符号付き整数

Double:

8バイト符号付き整数

Character:

Unicode UTF-8符号化された1文字(1オクテット以上となる)

Enumeration:

取り得るユニークな値のリストの一つ

Sub-list:

あるリストの中から複数の値を取り出したリスト(統計情報ではサポートされない)。Sub-listのタイプもまた指定されなければならない。そのタイプは(sub-listの例外と共に)この節で指定されるタイプから選ばなければならない。例えば、”Type : sub-list of enumeration”。Sub-listの符号化法については付属資料Aおよび付属資料B.3を参照。

Optional: YesまたはNo

パラメータが信号かイベントから省略されるかもしれないかどうか述べます。

Possible values:

一つのパッケージは、特定の値の組か、値がどのように決定されるかの説明を指定しなければならない。パッケージはまた、デフォルト値か、デフォルト値がそのディスクリプタから省略される場合のデフォルトの動作を指定しなければならない。例えば、パッケージは、デフォルト値が省略された場合、そのプロパティに関する手順が停止されるということ指定してもよい。

Default:

デフォルト値(手順ではない)は、事前に用意されているものとして指定されてもよい。

Description:

### 12.3 識別子

テキスト符号化における識別子は、64文字までで、スペースを含まず、英字で始まり、そして、英数字と"/"または数字から構成されなくてはならない。また、場合によってはアンダースコア"\_"を含む。

バイナリ符号化における識別子は2オクテット長である。

列挙型(enumerated type)の値として使用される識別子も含んでいるため、テキスト値とバイナリ値の両方が各識別子のために規定されなくてはならない。

### 12.4 パッケージ登録

パッケージは相互接続性の理由からIANAに登録することができる。14章を参照。

## 13. プロファイルの定義

プロファイルは、JT-H248.1プロトコルがどのように使用されるか、そしてどの機能がMGによってサポートされるかという、より進んだ定義をするために指定されてもよい。それはMGC/MG H.248インタフェースの能力のみを記述する。プロファイル自身は、JT-H248.1に関連するどのオプションが使用されるかを指定する。例えば、あるアプリケーションのために使用されるトランスポートとパッケージを指定する。一つのプロファイルは一つの名前(IANAに登録される)とバージョンによって識別される。一つの名前は、64文字までの長さの文字列(大文字、小文字は区別されない)でなければならない。バージョンは1から99の範囲でなければならない。

プロファイル自身は、特定のアプリケーションのためのオプションを示すドキュメントである。このドキュメントのために設定されたフォーマットは存在しない。必須の項目は、その名前とバージョンを示す項目とそのプロファイルの要約があるべきであるということのみである。

以下の二点のみがプロファイルに必須の項目である。

- Profile Identification : ServiceChangeコマンドで送信されるプロファイルの名前とバージョン



- ・ Summary : プロファイルが何であるかの記述

付録IIIは、プロファイル定義のテンプレートを含む。これはプロファイル定義の基本として用いられるべきである。

## 14. IANA

### 14.1 パッケージ

パッケージをIANAに登録するためには、以下の事項を満足していなければならない。

1. 各パッケージとして固有の文字列名称、固有のシリアル番号とバージョン番号が、登録される。文字列名称には、テキスト符号化が使用される。シリアル番号はバイナリ符号化が使用されるべきである。シリアル番号0x8000~0xffffはプライベート使用のために予約されている。また、0も予約されている。
2. 連絡先の担当者名、電子メールアドレス、住所が指定されていなくてはならない。この連絡先情報は、必要に応じて決定機関により更新されなくてはならない。
3. パッケージを記述する文書の参照は、公開されるべきである。その文書は記述するパッケージのバージョンを指定しなくてはならない。もし、その文書が公開されるものであれば、それは公開Webサーバに置かれ、不変のURLを持つべきである。また、そのサイトは文書に対するコメントを受け取る機構を持ち、適切な応答を返すことができるようにすべきである。
4. 標準化団体以外に登録するパッケージ名は、少なくとも8文字の長さが必要である。
5. もし、他の全ての条件が満足されていれば、他の全てのパッケージ名は先着順に受け付けられる。

### 14.2 エラーコード

エラーコードをIANAに登録するためには、以下の事項を満足していなければならない。

1. 各エラーとして、エラー番号と1行の文字列(最大80文字)が登録される。
2. エラーが検出される条件の完全な記述が公的に入手可能な文書内に含まれていなければならない。その記述は他の既存のエラーコード全てとの違いが十分に明らかでなければならない。
3. その文書は、公開されたWEBサーバ上に置かれ、不変のURLを持つべきである。
4. 標準化団体により登録されるエラー番号は3または4文字のエラー番号でなければならない。
5. 他の機関や個人によって登録されるエラー番号は4文字のエラー番号でなければならない。
6. あるエラー番号を、それを定義した機関または個人、またはそれらの後任者か受託者以外が、再定義や変更することはできない。

### 14.3 ServiceChange理由

Service change理由をIANAに登録するためには、以下の事項を満足していなければならない。

1. 各理由として、最大80文字のフレーズと固有の理由コードが登録される。
2. その理由が使用される条件の完全な記述が公開された文書内に含まれていなくてはならない。その記述は他の既存の理由全てとの違いが十分に明らかでなければならない。
3. その文書は、公開されたWEBサーバ上に置かれ、不変のURLを持つべきである。

### 14.4 プロファイル

プロファイルをIANAに登録するためには、以下の事項を満足していなければならない。

1. 各プロフィールとして、固有の文字列名称とバージョン番号が登録される(バージョンはそのプロフィール名がワイルドカードを含む場合に省略されても良い)。
2. 連絡先の担当者名、電子メールアドレス、住所が指定されていなければならない。この連絡先情報は、必要に応じてプロフィールを定義した機関によって更新されなければならない。
3. 認知された標準化団体以外によって登録されたプロフィールは、少なくとも6文字の長さのプロフィール名を持たなければならない。
4. 名称の最後にワイルドカード"\*"を含むプロフィール名は、始めの6文字全てが指定されていれば、受け付けられなければならない。そのプロフィール名を発行された組織が、そのワイルドカードに関係する名称スペースの管理を行う。IANAは、"name\*"の範囲内にある、他のプロフィール名を発行してはならない。
5. 他の全てのプロフィール名は、その他の条件が全て合致していれば、先着順に処理される。

## 付属資料A バイナリ符号化 (BINARY ENCODING OF THE PROTOCOL)

(本付属資料は本標準の必須部分である)

本付属資料では、ASN.1[ITU-T Recommendation X.680: Information Technology - Abstract Syntax Notation One (ASN.1) - Specification of basic notation]で定義される記述を使用したメッセージの文法を規定する。メッセージは、[ITU-T Recommendation X.690 Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER)]の中で規定される基本符号化規則を適用することによって、伝送形式に符号化されなければならない。

### A.1 ワイルドカードの符号化

本プロトコルでは、ALLおよびCHOOSEワイルドカードの使用が許されている。これにより、MGCは部分的なTerminationIDを指定し、MGにその指定と一致する値の中から選ばせることができる。TerminationIDは名称の階層を符号化してもよい。この階層はあらかじめ準備される。例えば、TerminationIDはあるトランクグループと、そのグループ内のあるトランクと、ある回路から構成されてもよい。ワイルドカード化は、全てのレベルにおいて使用可能でなければならない。以下の段落では、どのようにしてこれが達成されるかについて説明する。

ASN.1の記述は、TerminationIDとして、最大8オクテットまでの文字列を使用する。これは、TerminationIDが最大64ビットから構成されることを意味する。厳密に指定された(Fully specified)TerminationIDの前に一続きのワイルドカード化フィールドがあってもよい。ワイルドカード化フィールドは、1オクテット長である。このオクテットのビット7(MSB)はどのタイプのワイルドカード化が実行されるかを指定する(このビットが1であればALLワイルドカードが使用され、このビットが0であればCHOOSEワイルドカードが使用される)。ワイルドカード化フィールドのビット6は、そのワイルドカード化がその階層化された名称構造の中の一つのレベルに属するの(この場合、ビットの値は0)、ワイルドカード化フィールドの中で指定される一つの階層レベルとそれ以下のレベルに属するの(この場合、ビットの値は1)、何れであるかを示す。ワイルドカード化フィールドのビット0から5は、TerminationIDが開始されるビット位置を示す。

以下にいくつかの例を示す。これらの例において、MSBは左側に現れるものとする。

TerminationIDは3オクテット長で、各オクテットは階層化された名称構造の中の一つのレベルを表すものと仮定する。例えばある有効なTerminationIDは次の通りとなる。

```
00000001 00011110 01010101
```

先頭に00000001 00011110が付く全ての名称の指定は下のようになる。

```
ワイルドカード化フィールド：10000111
```

```
TerminationID    : 00000001 00011110 xxxxxxxx
```

“x”と表示したビットの値はこの指定には無関係で、受信側では無視されなければならない。

受信側に対して2番目のオクテットが00011110である一つの名称を選択させる場合は以下のようにする。

```
ワイルドカード化フィールド：00010111に引き続き00000111
```

```
TerminationID    : xxxxxxxx 00011110 xxxxxxxx
```

初めのワイルドカード化フィールドはビット23から始まる名称階層におけるCHOOSEワイルドカードを示している。これは、我々が仮定した名称構造の中の最も高い階層である。二番目のワイルドカード化フィールドは、ビット7から始まる名称階層における、CHOOSEワイルドカードを示している。これは我々が仮定した名称構造の中の最も低い階層である。

最後に、最も高い階層の名称が00000001であるCHOOSEワイルドカード化名は、次のようになる。

```
ワイルドカード化フィールド：01001111
```

```
TerminationID    : 00000001 xxxxxxxx xxxxxxxx
```

ワイルドカード化フィールドにおける最初のオクテットのビット位置6の1という値は、そのワイルドカー

ド化がその指定された階層とそれ以下の全ての階層に適用されることを示している。

ContextIDもまたワイルドカード化してよい。しかし、ContextIDの場合、一部の名称を指定することは許されていない。ContextID 0x0は、NULL Contextを示すために使用されなければならない。ContextID 0xFFFFFFFFは、CHOOSEワイルドカードを示すために使用されなければならない。ContextID 0xFFFFFFFFは、ALLワイルドカードを示すために使用されなければならない。

Termination ID 0xFFFFFFFFFFFFFFFFは、ROOT Terminationを示すために使用されなければならない。

## A.2 ASN.1構文仕様

本節は、JT-H248.1プロトコル構文のASN.1仕様を含む。

注記1：アプリケーションレベルフレーミングを使用するトランスポート機構が用いられる場合は、下記 Transactionの定義が変わる。この場合に適用される定義については、トランスポート機構を定義する付属資料またはITU-T勧告H.248を参照のこと。

注記2：本構文の仕様は、構成要素や値に対して、全ての制限を強制しているわけではない。いくつかの追加制限事項がコメントで明示されており、その他の制限事項は本標準の文中に示されている。この追加制限事項は、たとえこの仕様によって制限されなくても本プロトコルの一部である。

注記3：本付属資料のASN.1モジュールは、プロパティパラメータ、シグナルパラメータ、イベントパラメータの値や統計情報を符号化するために、"OCTET STRING"タイプを使用する。この値の実際のタイプは、付属資料Cもしくは関連するパッケージ定義の中で変更され、規定されている。値は、まず下に示す表で使っているタイプに基づいてBER符号化される。このBER符号化の結果は、次にASN.1 OCTET STRINGの"double Wrapping"の値として符号化される。付属資料Cやパッケージで規定されるフォーマットは、下に示す表にてBER符号化と関連付けられる。

パッケージで規定されるタイプ	ASN.1 BERタイプ
String	IA5StringまたはUTF8String (注4)
Integer (4オクテット)	INTEGER
Double (8オクテット符号付整数)	INTEGER (注3)
Character (UTF-8、注1)	IA5String
Enumeration	ENUMERATED
Boolean	BOOLEAN
Unsigned Integer (注2)	INTEGER (注3)
注1：1バイト以上	
注2：符号無し整数は付属資料Cで参照される	
注3：INTEGERのBER符号化は4バイト使用を意味しない	
注4：文字列は、中身が全てASCII文字の場合にはIA5Stringとして符号化されるが、非ASCII文字を含んでいる場合にはUTF8Stringとして符号化されるべきである	

OCTET STRING値の符号化定義についてはITU-T勧告X.690の8.7を参照のこと。

MEDIA-GATEWAY-CONTROL {itu-t(0) recommendation(0) h(8) h248(248) modules(0)  
media-gateway-control(0) version3(3)}

```

DEFINITIONS AUTOMATIC TAGS : :=
BEGIN
MegacoMessage          : := SEQUENCE
{
    authHeader          AuthenticationHeader OPTIONAL,
    mess                Message
}

AuthenticationHeader  : := SEQUENCE
{
    secParmIndex        SecurityParmIndex,
    seqNum              SequenceNum,
    ad                  AuthData
}

SecurityParmIndex     : := OCTET STRING(SIZE(4))

SequenceNum           : := OCTET STRING(SIZE(4))

AuthData              : := OCTET STRING (SIZE (12..32))

Message : := SEQUENCE
{
    version             INTEGER(0..99),
-- ここで定義されるプロトコルのバージョンは3
    mId                MId,
    -- メッセージ発行側の名前/アドレス
    messageBody         CHOICE
    {
        messageError    ErrorDescriptor,
        transactions    SEQUENCE OF Transaction
    },
    . . .
}

MId : := CHOICE
{
    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName          DomainName,
    deviceName          PathName,
    mtpAddress          OCTET STRING(SIZE(2)),

```

```

-- mtpAddressのアドレス構造:
--      25-15          0
--      | PC          |NI|
--      24-14 bits    2 bits
-- 注意：14ビットは国際利用のために定義される。
--ポイントコードが16または24ビットとなるような二つのナショナルオプションが存在する。
-- mtpAddressをオクテット毎に一行並べるために、MSBは0として符号化されなくてはならない。

```

```

. . .
}

```

```

DomainName ::= SEQUENCE

```

```

{
    name          IA5String,
    -- nameは、先頭が英数字で始まり、
    --英数字、ハイフン、ドットから構成される。
    -- ただしドットは連続してはならない。
    portNumber    INTEGER(0..65535) OPTIONAL
}

```

```

IP4Address ::= SEQUENCE

```

```

{
    address       OCTET STRING (SIZE(4)),
    portNumber    INTEGER(0..65535) OPTIONAL
}

```

```

IP6Address ::= SEQUENCE

```

```

{
    address       OCTET STRING (SIZE(16)),
    portNumber    INTEGER(0..65535) OPTIONAL
}

```

```

PathName ::= IA5String(SIZE (1..64))

```

```

-- 本付属資料のA.3参照

```

```

Transaction ::= CHOICE

```

```

{
    transactionRequest    TransactionRequest,
    transactionPending    TransactionPending,
    transactionReply      TransactionReply,
    transactionResponseAck TransactionResponseAck,
    -- response ackの使用はトランスポートに依存
    . . . ,
    segmentReply          SegmentReply
}

```

}

TransactionId : := INTEGER(0..4294967295) -- 32ビットの符号無し整数

TransactionRequest : := SEQUENCE

```
{
    transactionId      TransactionId,
    actions            SEQUENCE OF ActionRequest,
    . . .
}
```

TransactionPending : := SEQUENCE

```
{
    transactionId      TransactionId,
    . . .
}
```

TransactionReply : := SEQUENCE

```
{
    transactionId      TransactionId,
    immAckRequired     NULL OPTIONAL,
    transactionResult  CHOICE
    {
        transactionError      ErrorDescriptor,
        actionReplies         SEQUENCE OF ActionReply
    },
    . . .
    segmentNumber      SegmentNumber OPTIONAL,
    segmentationComplete NULL OPTIONAL
}
```

SegmentReply : := SEQUENCE

```
{
    transactionId      TransactionId,
    segmentNumber      SegmentNumber
    segmentationComplete NULL OPTIONAL
    . . .
}
```

SegmentNumber : := INTEGER(0..65535)

TransactionResponseAck : := SEQUENCE OF TransactionAck

TransactionAck : := SEQUENCE

{

```
    firstAck          TransactionId,
    lastAck           TransactionId OPTIONAL
}
```

```
ErrorDescriptor ::= SEQUENCE
```

```
{
    errorCode          ErrorCode,
    errorText         ErrorText OPTIONAL
}
```

```
ErrorCode ::= INTEGER(0..65535)
```

```
-- エラーコードに関連するIANAについては13章参照
```

```
ErrorText ::= IA5String
```

```
ContextID ::= INTEGER(0..4294967295)
```

```
-- Context NULL値: 0
```

```
-- Context CHOOSE値: 429467294 (0xFFFFFFFFE)
```

```
-- Context ALL値: 4294967295 (0xFFFFFFFF)
```

```
ActionRequest ::= SEQUENCE
```

```
{
    contextId          ContextID,
    contextRequest     ContextRequest OPTIONAL,
    contextAttrAuditReq ContextAttrAuditRequest OPTIONAL,
    commandRequests   SEQUENCE OF CommandRequest
}
```

```
ActionReply ::= SEQUENCE
```

```
{
    contextId          ContextID,
    errorDescriptor    ErrorDescriptor OPTIONAL,
    contextReply       ContextRequest OPTIONAL,
    commandReply       SEQUENCE OF CommandReply
}
```

```
ContextRequest ::= SEQUENCE
```

```
{
    priority           INTEGER(0..15) OPTIONAL,
    emergency          BOOLEAN OPTIONAL,
    topologyReq       SEQUENCE OF TopologyRequest OPTIONAL,
}
```



```

    . . . ,

iepscallind                BOOLEAN OPTIONAL
contextProp                 SEQUENCE OF PropertyParam OPTIONAL,
contextList                 SEQUENCE OF ContextID OPTIONAL
}

```

-- contextListを返信する際には、ActionReplyの構造体（construct）のcontextIdは  
--関連するActionRequestからcontextIdを返す。

ContextAttrAuditRequest ::= SEQUENCE

```

{
    topology                NULL OPTIONAL,
    emergency                NULL OPTIONAL,
    priority                 NULL OPTIONAL,
    . . .
    iepscallind             NULL OPTIONAL,
    contextPropAud          SEQUENCE OF IndAudPropertyParm OPTIONAL,
    selectpriority           INTEGER(0..15) OPTIONAL,
    --与えられた優先度を選択する
    selectemergency         BOOLEAN OPTIONAL,
    --緊急呼かどうか(T/F)を選択する
    selectiepscallind       BOOLEAN OPTIONAL,
    --IEPSかどうか(T/F)を選択する
    selectLogic              SelectLogic OPTIONAL --デフォルト値はAND
}

```

SelectLogic ::= CHOICE

```

{
    andAUDITSelect          NULL, --全フィルタ状態が満たされる
    orAUDITSelect           NULL, --少なくとも一つのフィルタ状態が満たされる
    . . .
}

```

CommandRequest ::= SEQUENCE

```

{
    command                 Command,
    optional                 NULL OPTIONAL,
    wildcardReturn          NULL OPTIONAL,
    . . .
}

```

Command ::= CHOICE

```

{

```

```

addReq          AmmRequest,
moveReq         AmmRequest,
modReq          AmmRequest,
-- Add、Move、Modify requestは同じパラメータを持つ
subtractReq     SubtractRequest,
auditCapRequest AuditRequest,
auditValueRequest AuditRequest,
notifyReq       NotifyRequest,
serviceChangeReq ServiceChangeRequest,
. . .
}

```

CommandReply : := CHOICE

```

{
  addReply          AmmsReply,
  moveReply         AmmsReply,
  modReply          AmmsReply,
  subtractReply     AmmsReply,
  -- Add、Move、Modify、Subtract replyは同じパラメータを持つ
  auditCapReply    AuditReply,
  auditValueReply  AuditReply,
  notifyReply      NotifyReply,
  serviceChangeReply ServiceChangeReply,
  . . .
}

```

TopologyRequest : := SEQUENCE

```

{
  terminationFrom      TerminationID,
  terminationTo        TerminationID,
  topologyDirection    ENUMERATED
  {
    bothway(0),
    isolate(1),
    oneway(2)
  },
  . . . ,
  streamID             streamID      OPTIONAL,
  topologyDirectionExtension ENUMERATED,
  {
    onewayexternal(0),
    onewayboth(1),
    . . .
  }
}

```

```

    }
}

AmmRequest ::= SEQUENCE
{
    terminationID          TerminationIDList,
    descriptors            SEQUENCE OF AmmDescriptor,
    -- このSEQUENCEで許容されるのはタイプ毎に多くても一つのディスクリプタである
    -- (AmmDescriptor参照)
    ...
}

```

```

AmmDescriptor ::= CHOICE
{
    mediaDescriptor      MediaDescriptor OPTIONAL,
    modemDescriptor     ModemDescriptor OPTIONAL,
    muxDescriptor        MuxDescriptor OPTIONAL,
    eventsDescriptor     EventsDescriptor OPTIONAL,
    eventBufferDescriptor EventBufferDescriptor OPTIONAL,
    signalsDescriptor    SignalsDescriptor OPTIONAL,
    digitMapDescriptor   DigitMapDescriptor OPTIONAL,
    auditDescriptor      AuditDescriptor OPTIONAL,
    . . .
    statisticsDescriptor StatisticsDescriptor
}

```

```

AmmsReply ::= SEQUENCE
{
    terminationID          TerminationIDList,
    terminationAudit      TerminationAudit OPTIONAL
}

```

```

SubtractRequest ::= SEQUENCE
{
    terminationID          TerminationIDList,
    auditDescriptor        AuditDescriptor OPTIONAL,
    . . .
}

```

```

AuditRequest ::= SEQUENCE
{

```

```

    terminationID      TerminationID,
    auditDescriptor    AuditDescriptor,
    . . .
    terminationIDList  TerminationIDList OPTIONAL
}
--AuditRequestにおいてterminationIDListを使用するときにはterminationIDにはリスト内に
--初期terminationを含めなければならない

```

```

AuditReply ::= CHOICE
{
    contextAuditResult  TerminationIDList,
    error                ErrorDescriptor,
    auditResult          AuditResult,
    ...,
    auditResultTermList TermListAuditResult
}

```

```

AuditResult ::= SEQUENCE
{
    TerminationID      TerminationID,
    TerminationAuditResult TerminationAudit,
}

```

```

TermListAuditResult ::= SEQUENCE
{
    TerminationID List TerminationIDList,
    TerminationAuditResult TerminationAudit,
    ...
}

```

TerminationAudit ::= SEQUENCE OF AuditReturnParameter

```

AuditReturnParameter ::= CHOICE
{
    errorDescriptor      ErrorDescriptor,
    mediaDescriptor      MediaDescriptor,
    modemDescriptor      ModemDescriptor,
    muxDescriptor        MuxDescriptor,
    eventsDescriptor     EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor    SignalsDescriptor,
    digitMapDescriptor   DigitMapDescriptor,
    observedEventsDescriptor ObservedEventsDescriptor,
}

```

```

        statisticsDescriptor      StatisticsDescriptor,
        packagesDescriptor       PackagesDescriptor,
        emptyDescriptors         AuditDescriptor,
...
}

```

AuditDescriptor ::= SEQUENCE

```

{
    auditToken      BIT STRING
    {
        muxToken(0), modemToken(1), mediaToken(2),
        eventsToken(3), signalsToken(4),
        digitMapToken(5), statsToken(6),
        observedEventsToken(7),
        packagesToken(8), eventBufferToken(9)
    } OPTIONAL,
    . . . ,
    auditPropertyToken SEQUENCE OF IndAuditParameter OPTIONAL,
}

```

IndAuditParameter ::= CHOICE

```

{
    indaudmediaDescriptor      IndAudMediaDescriptor,
    indaudmodemDescriptor      IndAudModemDescriptor,
    indaudeventsDescriptor     IndAudEventsDescriptor,
    indaudeventBufferDescriptor IndAudEventBufferDescriptor,
    indaudsignalsDescriptor    IndAudSignalsDescriptor,
    indauddigitMapDescriptor   IndAudDigitMapDescriptor,
    indaudstatisticsDescriptor IndAudStatisticsDescriptor,
    indaudpackagesDescriptor   IndAudPackagesDescriptor,
    . . .
}

```

IndAudMediaDescriptor ::= SEQUENCE

```

{
    termStateDescr  IndAudTerminationStateDescriptor OPTIONAL,
    streams CHOICE
        {
            oneStream      IndAudStreamParms,
            multiStream    SEQUENCE OF IndAudStreamDescriptor
        } OPTIONAL,
    . . .
}

```

```
}
```

```
IndAudStreamDescriptor ::= SEQUENCE
```

```
{  
    streamID      StreamID,  
    streamParms   IndAudStreamParms  
}
```

```
IndAudStreamParms ::= SEQUENCE
```

```
{  
    localControlDescriptor  IndAudLocalControlDescriptor OPTIONAL,  
    localDescriptor         IndAudLocalRemoteDescriptor OPTIONAL,  
    remoteDescriptor        IndAudLocalRemoteDescriptor OPTIONAL,  
    . . .  
    statisticsDescriptor    IndAudStatisticsDescriptor OPTIONAL  
}
```

```
IndAudLocalControlDescriptor ::= SEQUENCE
```

```
{  
    streamMode      NULL OPTIONAL,  
    reserveValue    NULL OPTIONAL,  
    reserveGroup    NULL OPTIONAL,  
    propertyParms   SEQUENCE OF IndAudPropertyParm OPTIONAL,  
    . . .  
    streamModeSel   StreamMode OPTIONAL
```

```
--streamModeとstreamModeSelの両方同時に指定されてはいけない
```

```
--もし両方同時に指定されている場合はstreamModeSelのみ使用する必要がある
```

```
}
```

```
IndAudPropertyParm ::= SEQUENCE
```

```
{  
    name          PkgdName,  
    . . .  
    propertyParms PropertyParm OPTIONAL  
}
```

```
--プロパティ値に基づき選択される。
```

```
--AND/OR 選択方法はcontextレベルで定義される
```

```
.
```

```
IndAudLocalRemoteDescriptor ::= SEQUENCE
```

```
{  
    propGroupID INTEGER(0..65535) OPTIONAL,  
    propGrps IndAudPropertyGroup,  
    . . .
```

```
}
```

```
IndAudPropertyGroup ::= SEQUENCE OF IndAudPropertyParm
```

```
IndAudTerminationStateDescriptor ::= SEQUENCE
```

```
{
```

```
    propertyParms          SEQUENCE OF IndAudPropertyParm,
```

```
    eventBufferControl     NULL OPTIONAL,
```

```
    serviceState          NULL OPTIONAL,
```

```
    . . .
```

```
    serviceStateSel       ServiceState OPTIONAL
```

```
--serviceStateとserviceStateSelの両方同時に指定されてはいけない。
```

```
--もし両方同時に指定されている場合はserviceStateSelのみ使用する必要がある。
```

```
}
```

```
IndAudEventsDescriptor ::= SEQUENCE
```

```
{
```

```
    requestID             RequestID OPTIONAL,
```

```
    pkgdName              PkgdName,
```

```
    streamID              StreamID OPTIONAL,
```

```
    . . .
```

```
}
```

```
IndAudEventBufferDescriptor ::= SEQUENCE
```

```
{
```

```
    eventName             PkgdName,
```

```
    streamID              StreamID OPTIONAL,
```

```
    . . .
```

```
}
```

```
IndAudSignalsDescriptor ::= CHOICE
```

```
{
```

```
    signal                IndAudSignal,
```

```
    seqSigList            IndAudSeqSigList,
```

```
    . . .
```

```
}
```

```
IndAudSeqSigList ::= SEQUENCE
```

```
{
```

```
    id                    INTEGER(0..65535),
```

```
    signalListIndAudSignal OPTIONAL
```

```
}
```

IndAudSignal ::= SEQUENCE

```
{
    signalName          PkgdName,
    streamID            StreamID OPTIONAL,
    . . .
    signalRequestID    RequestID OPTIONAL
}
```

IndAudDigitMapDescriptor ::= SEQUENCE

```
{
    digitMapName      DigitMapName  OPTIONAL
}
```

IndAudStatisticsDescriptor ::= SEQUENCE

```
{
    statName          PkgdName
}
```

IndAudPackagesDescriptor ::= SEQUENCE

```
{
    packageName      Name,
    packageVersion   INTEGER(0..99),
    . . .
}
```

NotifyRequest ::= SEQUENCE

```
{
    terminationID      TerminationIDList,
    observedEventsDescriptor ObservedEventsDescriptor,
    errorDescriptor    ErrorDescriptor OPTIONAL,
    . . .
}
```

NotifyReply ::= SEQUENCE

```
{
    terminationID      TerminationIDList OPTIONAL,
    errorDescriptor    ErrorDescriptor OPTIONAL,
    . . .
}
```

ObservedEventsDescriptor ::= SEQUENCE

```
{
```



```

    requestId          RequestID,
    observedEventLst   SEQUENCE OF ObservedEvent
}

```

ObservedEvent ::= SEQUENCE

```

{
    eventName          EventName,
    streamID           StreamID OPTIONAL,
    eventParList       SEQUENCE OF EventParameter,
    timeNotation       TimeNotation OPTIONAL,
    ...
}

```

EventName ::= PkgdName

EventParameter ::= SEQUENCE

```

{
    eventParameterName  Name,
    value               Value,
    -- extraInfoの使用については、PropertyParmについてのコメントを参照のこと。
    extraInfo CHOICE
    {
        relation        Relation,
        range            BOOLEAN,
        sublist         BOOLEAN
    } OPTIONAL,
    ...
}

```

ServiceChangeRequest ::= SEQUENCE

```

{
    terminationID       TerminationIDList,
    serviceChangeParms  ServiceChangeParm,
    . . .
}

```

ServiceChangeReply ::= SEQUENCE

```

{
    terminationID       TerminationIDList,
    serviceChangeResult ServiceChangeResult,
    . . .
}

```

-- ServiceChangeResultは必須パラメータがない。  
-- したがってServiceChangeParmとServiceChangeResParmを区別する。

```
ServiceChangeResult ::= CHOICE
{
    errorDescriptor          ErrorDescriptor,
    serviceChangeResParms   ServiceChangeResParm
}
```

```
WildcardField ::= OCTET STRING(SIZE(1))
```

```
TerminationID ::= SEQUENCE
{
    wildcard          SEQUENCE OF WildcardField,
    id                OCTET STRING(SIZE(1..8)),
    ...
}
```

-- ワイルドカードメカニズムの説明は本付属資料のA.1参照。  
-- TerminationID 0xFFFFFFFFFFFFFFFFはRoot Terminationを示す。

```
TerminationIDList ::= SEQUENCE OF TerminationID
MediaDescriptor ::= SEQUENCE
{
    termStateDescr      TerminationStateDescriptor OPTIONAL,
    streams             CHOICE
        {
            oneStream    StreamParms,
            multiStream  SEQUENCE OF StreamDescriptor
        }, OPTIONAL,
    ...
}
```

```
StreamDescriptor ::= SEQUENCE
{
    streamID            StreamID,
    streamParms        StreamParms
}
```

```
StreamParms ::= SEQUENCE
{
    localControlDescriptor  LocalControlDescriptor OPTIONAL,
    localDescriptor        LocalRemoteDescriptor OPTIONAL,
}
```

```

remoteDescriptor      LocalRemoteDescriptor OPTIONAL,
. . .
statisticsDescriptor StatisticsDescriptor OPTIONAL
}

```

LocalControlDescriptor ::= SEQUENCE

```

{
    streamMode          StreamMode OPTIONAL,
    reserveValue       BOOLEAN OPTIONAL,
    reserveGroup       BOOLEAN OPTIONAL,
    propertyParms      SEQUENCE OF PropertyParm,
    . . .
}

```

StreamMode ::= ENUMERATED

```

{
    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
    . . .
}

```

- PropertyParmでは値はオクテット列の”SEQUENCE OF”
- MGによる送信の場合、解釈は以下の通り：
- 空のsequenceはCHOOSEを意味する。
- 一要素のsequenceは値を指定する。
- sublistフィールドが選択されていない場合、
- それ以上のsequenceは「その値の一つを選択」("choose one of the values")を意味する。
- (つまり、値1 OR 値2 OR...)
- sublistフィールドが選択されている場合、
- 二要素以上のsequenceはリスト値プロパティの値での符号化。
- (つまり、値1 AND 値2 AND...)
- 値のsequenceの長さが1の場合、関連するフィールドが選択される
- だけでもよい。
- これはMGがそのプロパティの値を選択しなくてはならないことを示している。
- 例えば、”x > 3”(greaterThan valueを使用)はMGがプロパティ“x”に3より
- 大きな値を選択することを指示する。
- rangeフィールドは、値のsequenceの長さが2であれば、選択される
- だけでもよい。
- これはMGがその値のsequenceの第一オクテットと続くオクテットの
- 間の値を、境界値も含めて、選択しなくてはならないことを示している。

- MGが送信した場合、AuditCapability要求への応答だけが
- 複数の値、一つのrange、または関連フィールドを含んでよい。

PropertyParm ::= SEQUENCE

```
{
    name                PkgdName,
    value               SEQUENCE OF OCTET STRING,
    extraInfo           CHOICE
        {
            relation     Relation,
            range         BOOLEAN,
            sublist      BOOLEAN
        } OPTIONAL,
    ...
}
```

Name ::= OCTET STRING(SIZE(2))

PkgdName ::= OCTET STRING(SIZE(4))

- パッケージ名(2オクテット)にプロパティ、イベント、
- シグナルの名前もしくはStatistics ID(2オクテット)をあわせたものを表現している。
- パッケージをワイルドカード化するために、最初の2オクテットに
- 0xFFFFを使用する。CHOOSEは許容されない。
- 付属資料Cで定義されるnative property tagを参照するために
- 最初の2オクテットに0x0000を使用する。
- PropertyID、EventID、SignalIDもしくはStatistics IDのワイルドカード化
- には、最後の2オクテットに0xFFFFを使用し、CHOOSEは許容されない。
- パッケージ名のワイルドカード化は、PropertyID、EventID、SignalIDまたは
- Statistics IDもワイルドカード化される場合のみ許容される。

Relation ::= ENUMERATED

```
{
    greaterThan(0),
    smallerThan(1),
    unequalTo(2),
    . . .
}
```

LocalRemoteDescriptor ::= SEQUENCE

```
{
    propGrps           SEQUENCE OF PropertyGroup,
    . . .
}
```

PropertyGroup ::= SEQUENCE OF PropertyParm

TerminationStateDescriptor ::= SEQUENCE

```
{
    propertyParms      SEQUENCE OF PropertyParm,
    eventBufferControl EventBufferControl OPTIONAL,
    serviceState       ServiceState OPTIONAL,
    . . .
}
```

EventBufferControl ::= ENUMERATED

```
{
    Off(0),
    LockStep(1),
    . . .
}
```

ServiceState ::= ENUMERATED

```
{
    test(0),
    outOfSvc(1),
    inSvc(2),
    . . .
}
```

MuxDescriptor ::= SEQUENCE

```
{
    muxType           MuxType,
    termList          SEQUENCE OF TerminationID,
    nonStandardData   NonStandardData OPTIONAL,
    . . .
}
```

MuxType ::= ENUMERATED

```
{
    h221(0),
    h223(1),
    h226(2),
    v76(3),
    . . .
    nx64k(4)
}
```

StreamID ::= INTEGER(0..65535) -- 16ビット符号無し整数

EventsDescriptor ::= SEQUENCE

```
{
    requestID          RequestID  OPTIONAL,
    --EventListが空でない場合、RequestIDは
    --存在しなければならない。
    EventList          SEQUENCE OF RequestedEvent
}
```

RequestedEvent ::= SEQUENCE

```
{
    pkgdName          PkgdName,
    streamID          StreamID OPTIONAL,
    eventAction       RequestedActions OPTIONAL,
    evParList         SEQUENCE OF EventParameter,
    ...
}
```

RegulatedEmbeddedDescriptor ::= SEQUENCE

```
{
    secondEvent       SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    . . .
}
```

NotifyBehaviour ::= CHOICE

```
{
    NotifyImmediate   NULL,
    NotifyRegulated   RegulatedEmbeddedDescriptor,
    NeverNotify       Null,
    . . .
}
```

RequestedActions ::= SEQUENCE

```
{
    keepActive        BOOLEAN OPTIONAL,
    eventDM           EventDM OPTIONAL,
    secondEvent       SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    . . .
    notifyBehaviour   NotifyBehavior OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}
```

}

EventDM ::= CHOICE

```
{
    digitMapName      DigitMapName,
    digitMapValue     DigitMapValue
}
```

SecondEventsDescriptor ::= SEQUENCE

```
{
    requestID          RequestID OPTIONAL,
    eventList          SEQUENCE OF SecondRequestedEvent,
    ...
}
```

SecondRequestedEvent ::= SEQUENCE

```
{
    pkgdName          PkgdName,
    streamID          StreamID OPTIONAL,
    eventAction       SecondRequestedActions OPTIONAL,
    evParList         SEQUENCE OF EventParameter,
    ...
}
```

SecondRequestedActions ::= SEQUENCE

```
{
    keepActive        BOOLEAN OPTIONAL,
    eventDM           EventDM OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    . . .
    notifyBehaviour   NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}
```

EventBufferDescriptor ::= SEQUENCE OF EventSpec

EventSpec ::= SEQUENCE

```
{
    eventName          EventName,
    streamID           StreamID OPTIONAL,
    eventParList       SEQUENCE OF EventParameter,
    ...
}
```

SignalsDescriptor ::= SEQUENCE OF SignalRequest

SignalRequest ::= CHOICE

```
{  
    signal          Signal,  
    seqSigList     SeqSigList,  
    ...  
}
```

SeqSigList ::= SEQUENCE

```
{  
    id              INTEGER(0..65535),  
    signalList     SEQUENCE OF Signal  
}
```

Signal ::= SEQUENCE

```
{  
    signalName     SignalName,  
    streamID       StreamID OPTIONAL,  
    sigType        SignalType OPTIONAL,  
    duration       INTEGER (0..65535) OPTIONAL,  
    notifyCompletion  BOOLEAN OPTIONAL,  
    keepActive     BOOLEAN OPTIONAL,  
    sigParList     SEQUENCE OF SigParameter,  
    ...  
    direction      SignalDirection OPTIONAL,  
    requestID      RequestID OPTIONAL,  
    intersigDelay  INTERGER(0..65535) OPTIONAL  
}
```

SignalType ::= ENUMERATED

```
{  
    brief(0),  
    onOff(1),  
    timeOut(2),  
    . . .  
}
```

SignalDirection ::= ENUMERATED

```
{  
    internal(0),  
    external(1),  
}
```



```
    both(2)
    . . .
}
```

SignalName ::= PkgdName

NotifyCompletion ::= BIT STRING

```
{
    onTimeOut(0), onInterruptByEvent(1),
    onInterruptByNewSignalDescr(2), otherReason(3), onIteration(4)
}
```

SigParameter ::= SEQUENCE

```
{
    sigParameterName      Name,
    value                  Value,
    --extraInfoの使用に関しては、PropertyParmに関連するコメントを参照
    extraInfo             CHOICE
        {
            relation      Relation
            range          BOOLEAN,
            sublist       BOOLEAN
        } OPTIONAL,
    . . .
}
```

--全てのイベントでのAuditCapReplyでは、そのRequestIDがALLでなければいけない。

-- ALLは0xffffffffと表される。

RequestID ::= INTEGER(0..4294967295) -- 32ビット符号無し整数

ModemDescriptor ::= SEQUENCE

```
{
    mtl          SEQUENCE OF ModemType,
    mpl          SEQUENCE OF PropertyParm,
    nonStandardData NonStandardData OPTIONAL
}
```

ModemType ::= ENUMERATED

```
{
    v18(0),
    v22(1),

```

```

v22bis(2),
v32(3),
v32bis(4),
v34(5),
v90(6),
v91(7),
synchISDN(8),
. . .
}

```

DigitMapDescriptor : := SEQUENCE

```

{
    digitMapName          DigitMapName  OPTIONAL,
    digitMapValue         DigitMapValue  OPTIONAL
}

```

DigitMapName : := Name

DigitMapValue : := SEQUENCE

```

{
    startTimer            INTEGER(0..99) OPTIONAL,
    shortTimer            INTEGER(0..99) OPTIONAL,
    longTimer             INTEGER(0..99) OPTIONAL,
    digitMapBody          IA5String,
    --start、short、long Timerの単位は秒であり、duration timerは
--100ミリ秒である。start、short、longの範囲は1から99秒であり、
--durationは、100ミリ秒から9.9秒である。
-- DigitMap構文の説明は本付属資料のA.3参照
    . . . ,
    durationTimer         INTEGER(0. . 99)  OPTIONAL
}

```

ServiceChangeParm : := SEQUENCE

```

{
    serviceChangeMethod   ServiceChangeMethod,
    serviceChangeAddress  ServiceChangeAddress OPTIONAL,
    serviceChangeVersion  INTEGER(0..99) OPTIONAL,
    serviceChangeProfile  ServiceChangeProfile OPTIONAL,
    serviceChangeReason   Value,
    --serviceChangeReasonは、理由コードの数字とオプションのテキストの記述
--で構成される。
--serviceChangeReasonは、10進数の理由コードと、オプションとしてスペース
--1文字の後にテキストの文字列が続くという構成とならなくてはならない。
}

```

--この文字列はIA5Stringとして、まずBER符号化される。  
 --このBER符号化の結果は、次にASN.1 OCTET STRINGタイプの  
 --"double wrapping"の値のようにパッケージ要素として符号化される。

```
ServiceChangeDelay      INTEGER(0..4294967295) OPTIONAL,
                        -- 32ビット符号無し整数

serviceChangeMgcId     Mid OPTIONAL,
timeStamp              TimeNotation OPTIONAL,
nonStandardData        NonStandardData OPTIONAL,
. . . ,
serviceChangeInfo      AuditDescriptor OPTIONAL,
serviceChangeIncompleteFlag  NULL OPTIONAL
}
```

ServiceChangeAddress ::= CHOICE

```
{
  portNumber           INTEGER(0..65535), -- TCP/UDPポート番号
  ip4Address            IP4Address,
  ip6Address            IP6Address,
  domainName           DomainName,
  deviceName           PathName,
  mtpAddress            OCTET STRING(SIZE(2..4)),
  . . .
}
```

ServiceChangeResParm ::= SEQUENCE

```
{
  serviceChangeMgcId   Mid OPTIONAL,
  serviceChangeAddress ServiceChangeAddress OPTIONAL,
  serviceChangeVersion INTEGER(0..99) OPTIONAL,
  serviceChangeProfile ServiceChangeProfile OPTIONAL,
  timestamp            TimeNotation OPTIONAL,
  . . .
}
```

ServiceChangeMethod ::= ENUMERATED

```
{
  failover(0),
  forced(1),
  graceful(2),
  restart(3),
  disconnected(4),
  handOff(5),
  . . .
}
```

```
}
```

```
ServiceChangeProfile ::= SEQUENCE
```

```
{
```

```
    profileName      IA5String(Size (1..67))
```

```
    --名前が64文字、“/”が1文字、ABNFのバージョンが2文字
```

```
}
```

```
PackagesDescriptor ::= SEQUENCE OF PackagesItem
```

```
PackagesItem ::= SEQUENCE
```

```
{
```

```
    packageName      Name,
```

```
    packageVersion   INTEGER(0..99),
```

```
    ...
```

```
}
```

```
StatisticsDescriptor ::= SEQUENCE OF StatisticsParameter
```

```
StatisticsParameter ::= SEQUENCE
```

```
{
```

```
    statName          PkgdName,
```

```
    statValue         Value OPTIONAL
```

```
}
```

```
--統計情報がsub-listから成るならば、statValueには1以上のオクテットストリング
```

```
--が存在することになる。
```

```
NonStandardData ::= SEQUENCE
```

```
{
```

```
    nonStandardIdentifier NonStandardIdentifier,
```

```
    data                  OCTET STRING
```

```
}
```

```
NonStandardIdentifier ::= CHOICE
```

```
{
```

```
    object              OBJECT IDENTIFIER,
```

```
    h221NonStandard     H221NonStandard,
```

```
    experimental       IA5STRING(SIZE(8)),
```

```
-- 最初の二文字は"X-"もしくは"X+"であるべき
```

```
    . . .
```

```
}
```

```
H221NonStandard ::= SEQUENCE
```

```

{   t35CountryCode1  INTEGER(0..255),
    t35CountryCode   2      INTEGER(0..255), -- T.35の通り、国
    t35Extension     INTEGER(0..255), -- 国に割り当て
    manufacturerCode INTEGER(0..65535), --国に割り当て
    . . .
}

TimeNotation  : := SEQUENCE
{
    date          IA5String(SIZE(8)), -- yyyyymmddフォーマット
    time          IA5String(SIZE(8)) -- hhmmssssフォーマット
                -- ISO 8601 : 1988による
}

Value  : := SEQUENCE OF OCTET STRING

END

```

### A.3 DigitMapとPath Name

文法上の観点から見ると、DigitMapは構文上の制限が課せられた文字列となる。有効なdigit mapの文法はABNF[RFC 2234]で規定される。本節で示すDigitMapの文法としては実例を示すのみとする。付属資料B内のdigitMapの定義と本節の記述が異なる場合は付属資料Bの方が優先される。

```

digitMap = (digitString / LWSP "(" LWSP digitStringList LWSP ")"
LWSP)
digitStringList = digitString *( LWSP "|" LWSP digitString )
digitString = 1*(digitStringElement)
digitStringElement = digitPosition [DOT]
digitPosition = digitMapLetter / digitMapRange
digitMapRange = ("x" / LWSP "[" LWSP digitLetter LWSP "]" LWSP)
digitLetter = *((DIGIT "-" DIGIT) / digitMapLetter)
digitMapLetter = DIGIT ;基本的なイベントシンボル
                / %x41-4B / %x61-6B ;'a'から'k' および'A'から'K'
                / "L" / "S" / "T" ;イベント間タイマ
                ;(long, short, start)
                / "Z" ;長期間修飾子

DOT = %x2E ; "."
SP = %x20; スペース
HTAB = %x09; 水平タブ
CR = %x0D; キャリッジリターン
LF = %x0A; ラインフィード
LWSP = *(WSP / COMMENT / EOL)
EOL = (CR [LF] / LF)

```

WSP = SP / HTAB; ホワイトスペース  
 SafeChar = DIGIT / ALPHA / "+" / "-" / "&" / "!" / "\_" / "/" /  
 "'" / "?" / "@" / "^" / "" / "~" / "\*" / "\$" / "%" /  
 "(" / ")" / "%" / "|" / "."  
 RestChar = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /  
 "<" / ">" / "="  
 DIGIT = %x30-39 ; 数字0から9  
 ALPHA = %x41-5A / %x61-7A ; 'A'から'Z'および'a'から'z'

Path nameもまた、構文上の制限が課せられた文字列となる。この制限を定義するABNFの結果が付属資料Bからコピーされる。

; pathNAMEの全長は、64文字を超えてはいけない。

pathname = ["\*"] NAME \*("/" /\*"/ ALPHA / DIGIT / "\_" / "\$" )  
 ["@" pathDomainName ]

; ABNFは、2つ以上の連続する"."を、それがパストメインネームの中では意味がなくても許容する。

pathDomainName = (ALPHA / DIGIT / "\*" )  
 \*63(ALPHA / DIGIT / "-" / "\*" / ".")

NAME = ALPHA \*63(ALPHA / DIGIT / "\_" )

## 付属資料B テキスト符号化 (TEXT ENCODING OF THE PROTOCOL)

(本付属資料は本標準の必須部分である)

### B.1 ワイルドカードの符号化

本プロトコルのテキスト符号化において、TerminationIDは自由に付与できるが、名前の選択を十分考慮することで、ワイルドカードキャラクタ"\*"は、より役に立つようになる。ワイルドカードキャラクタに遭遇するとき、(適切ならば)その前半と後半で同じ文字列を持っている全てのTerminationIDに「合致する」ことになる。例えば、R13/3/1、R13/3/2およびR13/3/3といったTerminationIDがあるとしたとき、TerminationID R13/3/\*はそれらにすべてにマッチするだろう。全てのterminationが参照されなければならないある状況がある。TerminationID "\*"がそれであり、ALLとして参照される。CHOOSE TerminationID "\$"は、短期間だけ存在する(ephemeral) terminationを作成するか、または空きの物理terminationを選択しなければならないことをMGに知らせるために使われてもよい。

### B.2 ABNF仕様

このプロトコルの構文はRFC2234に従ってABNFで表現される。

注記： 本構文仕様は構成要素や値に対して全ての制限を強制しているわけではない。いくつかの追加制限事項がコメントで明示されており、他の制限事項は本標準の文中に示されている。この追加制限事項は、たとえこの仕様で強制されなくてもプロトコルの一部である。

注記： 文法はコンテキスト依存である。例えば、"Add"はそれが現れるコンテキストによってはAddTokenにもNAMEにもなり得る。

ABNFとテキスト符号化の全てにおいて大文字、小文字の区別がない。これはTerminationIDやDigitMapID等を含む。SDPはRFC2327の通り大文字、小文字を区別する。

; 注意—この章のABNFは様々なパッケージの要素の値(プロパティ、シグナルのパラメータ等)を

;符号化するためのVALUEの構成(もしくは、VALUEの構成リスト)を用いる。

;これらの値の型は変化し、そして関連するパッケージの定義の中で指定される。

;いくつかのそのような型は12.2章で述べられている。

;

;VALUEに対するABNF仕様はquotedString型またはSafeCharの集まりを許容している。

;ABNF VALUESの中のパッケージ要素の値の符号化は以下で指定される。

;もしある型の符号化がSafeChar以外の文字列を認めるならば、quotedString型が

;たとえSafeCharだけからなる特定の値のためであっても、その型の全ての値のために使用

;されなければならない。

;

;String : StringはVALUEのquoteString型を使わなければならない。そして、quotedString型

;として正当な全てのものを含むことができる。

;

;Integer,Double,and unsigned Integer : 10進の値として0から9の文字を使い符号化される。

;16進の値は'0x'が前に付かなければならない。そして、0-9、a-f、A-Fの文字を使用することができ

;る。8進のフォーマットはサポートされていない。

;負の整数は'-'で始まる10進数でなければならない。  
 ;SafeChar型のVALUEが使われなければならない。  
 ;  
 ;Character : 2重引用符によって囲まれた一文字のUTF-8符号化。  
 ;  
 ;Enumeration : EnumerationはSafeChar型のVALUEを使わなければならない。そして、  
 ;SafeChar型で認められている全てのものを含むことができる。  
 ;  
 ;Boolean : ブール値は”on”と”off”として符号化される。そして、大文字、小文字の区別はない。  
 ;SafeChar型のVALUEを使用しなければならない。  
 ;  
 ;Future Types : 定義された型はABNF仕様のVALUE中で適合していなければならない。  
 ;特に、型の符号化がSafeChar以外の文字を認めるならば、たとえSafeCharだけからなる  
 ;特定の値のためでも、quotedString型はその型の全ての値のために使われなければならない。  
 ;  
 ;注意 valueの中に2重引用符を使うことはできない。  
 ;  
 ;注意 SDPは行の始まりとして空白(whitespace)を許さない。  
 ;Megaco ABNFはLocal/Remote descriptorの中でSDPが始まる前の空白は許可している。  
 ;構文解析プログラムはLocal/Remoteトークンに続くLBRKTとSDPの先頭との間の空  
 ;白は受け入れるべきである。

megacoMessage = LWSP [authenticationHeader SEP ] message

authenticationHeader = AuthToken EQUAL SecurityParmIndex COLON  
 SequenceNum COLON AuthData

SecurityParmIndex = "0x" 8(HEXDIG)

SequenceNum = "0x" 8(HEXDIG)

AuthData = "0x" 24\*64(HEXDIG)

Message = MegacopToken SLASH Version SEP mId SEP messageBody

; ここで定義されるプロトコルのバージョンは2

messageBody = ( errorDescriptor / transactionList )

transactionList = 1\*( transactionRequest / transactionReply /  
 transactionPending / transactionResponseAck /  
 segmentReply)

; Response Ackの使用はトランスポートに依存する

transactionPending = PendingToken EQUAL TransactionID LBRKT RBRKT



transactionResponseAck = ResponseAckToken LBRKT transactionAck  
\*(COMMA transactionAck) RBRKT

transactionAck = TransactionID / (TransactionID "-" TransactionID)

transactionRequest = TransToken EQUAL TransactionID LBRKT  
actionRequest \*(COMMA actionRequest) RBRKT

actionRequest = CtxToken EQUAL ContextID LBRKT ((contextRequest  
[COMMA commandRequestList]) /  
commandRequestList) RBRKT

contextRequest = ((contextProperties [COMMA contextAudit])  
/ contextAudit)

contextProperties = contextProperty \*(COMMA contextProperty)

; 多くとも一度

; EmergencyOffはMGからMGC方向のみのH.248.1 V1とV2のEmergencyToken

;もしくは EmergencyOffTokenの中で使用される、しかし両方ではない

contextProperty = (topologyDescriptor / priority / EmergencyToken /  
EmergencyOffToken / iepsValue /  
contextAttrDescriptor)

contextAudit = ContextAuditToken LBRKT (contextAuditProperties  
\*(COMMA contextAuditProperties)) /  
indAudcontextAttrDescriptor RBRKT

; contextAuditSelector以外は多くとも一度

contextAuditProperties = ( TopologyToken / EmergencyToken /  
PriorityToken / IEPSToken / pkgdName /  
contextAuditSelector )

; 多くとも一度

contextAuditSelector = priority / emergencyValue / iepsValue /  
contextAttrDescriptor / auditSelectLogic

auditSelectLogic = [ AndAUDITselectToken / OrAUDITselectToken ]

; もし空の場合には、選択条件のANDが想定される

indAudcontextAttrDescriptor = ContextAttrToken LBRKTcontextAuditProperties  
\*( COMMA contextAuditProperties) RBRKT

;”O-“はオプション的コマンドを示す。

;”W-“はコマンドに対するワイルドカード応答を示す。

commandRequestList= ["O-"] ["W-"] commandRequest \*  
(COMMA ["O-"] ["W -"] commandRequest)

commandRequest = ( ammRequest / subtractRequest / auditRequest  
/ notifyRequest / serviceChangeRequest)

transactionReply = ReplyToken EQUAL TransactionID [SLASH segmentNumber  
[SLASH SegmentationCompleteToken]] LBRKT  
[ ImmAckRequiredToken COMMA]  
( errorDescriptor / actionReplyList ) RBRKT

segmentReply = MessageSegmentToken EQUAL TransactionID SLASH  
segmentNumber [SLASH SegmentationCompleteToken]

segmentNumber = UINT16

actionReplyList = actionReply \*(COMMA actionReply )

actionReply = CtxToken EQUAL ContextID LBRKT  
( errorDescriptor / commandReply /  
(commandReply COMMA errorDescriptor) ) RBRKT

commandReply = (( contextProperties [COMMA commandReplyList] )  
/ commandReplyList )

commandReplyList = commandReplies \*(COMMA commandReplies )

commandReplies = (serviceChangeReply / auditReply / ammsReply  
/ notifyReply )

;Add, Move, Modifyは同じ要求パラメータを持つ

ammRequest = (AddToken / MoveToken / ModifyToken ) EQUAL  
termIDList [LBRKT ammParameter \*(COMMA  
ammParameter) RBRKT]

;多くて一度

ammParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /  
eventsDescriptor / signalsDescriptor /

digitMapDescriptor / eventBufferDescriptor /  
 auditDescriptor / statisticsDescriptor)

ammsReply = (AddToken / MoveToken / ModifyToken / SubtractToken )  
 EQUAL termIDList [ LBRKT terminationAudit RBRKT ]

subtractRequest = SubtractToken EQUAL termIDList [ LBRKT  
 auditDescriptor RBRKT]

auditRequest = (AuditValueToken / AuditCapToken ) EQUAL  
 termIDList LBRKT auditDescriptor RBRKT

auditReply = (AuditValueToken / AuditCapToken )  
 ( contextTerminationAudit / auditOther)

auditOther = EQUAL termIDList [ LBRKT terminationAudit RBRKT ]

terminationAudit = auditReturnParameter \*(COMMA  
 auditReturnParameter)

contextTerminationAudit = EQUAL CtxToken ( terminationIDList / LBRKT  
 errorDescriptor RBRKT )

auditReturnParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /  
 eventsDescriptor / signalsDescriptor /  
 digitMapDescriptor / observedEventsDescriptor /  
 eventBufferDescriptor / statisticsDescriptor /  
 packagesDescriptor / errorDescriptor /  
 auditReturnItem )

auditReturnItem = (MuxToken / ModemToken / MediaToken /  
 DigitMapToken / StatsToken / ObservedEventsToken /  
 PackagesToken)

auditDescriptor = AuditToken LBRKT [ auditItem  
 \*(COMMA auditItem ) ] RBRKT

notifyRequest = NotifyToken EQUAL termIDList LBRKT  
 ( observedEventsDescriptor [ COMMA errorDescriptor ] )  
 RBRKT

notifyReply = NotifyToken EQUAL termIDList [ LBRKT errorDescriptor  
 RBRKT ]

```

serviceChangeRequest = ServiceChangeToken EQUAL termIDList LBRKT
                        serviceChangeDescriptor RBRKT

serviceChangeReply   = ServiceChangeToken EQUAL termIDList [LBRKT
                        (errorDescriptor / serviceChangeReplyDescriptor)
                        RBRKT]

errorDescriptor      = ErrorToken EQUAL ErrorCode LBRKT [quotedString]
                        RBRKT

ErrorCode            = 1*4(DIGIT) ; 拡張可能

TransactionID       = UINT32

mId                  = (( domainAddress / domainName ) [":" portNumber]) /
                        mtpAddress / deviceName

```

; ドメイン名では無意味であるが、ABNFは2つ以上の連続した”.”を許す。

```

domainName          = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" / ".")
                        ">"

deviceName          = pathNAME

```

; 0x0、0xFFFFFFFFE、0xFFFFFFFFFは予約済み

; '!'はNULL contextに使用される。'\*'はALLに使用される。'\$' はCHOOSEに使用される。

```

ContextID           = (UINT32 / "*" / "-" / "$")

domainAddress       = "[" (IPv4address / IPv6address) "]"
; RFC2373はIPv6アドレスの定義を含んでいる

IPv6address         = hexpart [ ":" IPv4address ]
IPv4address         = V4hex DOT V4hex DOT V4hex DOT V4hex
V4hex               = 1*3(DIGIT) ; "0".."255"
; RFC2373には現れるが、本プロトコルでは参照されない。
IPv6prefix          = hexpart SLASH 1*2DIGIT
hexpart             = hexseq ":@" [ hexseq ] / ":@" [ hexseq ] / hexseq
hexseq              = hex4 *( ":" hex4)
hex4                = 1*4HEXDIG

portNumber          = UINT16

```

; mtpAddressのアドレッシング構造：

; 25 – 15                    0

;    | PC                    | NI |

;    24 – 14 bits            2bits

; 注意：14ビットは国際利用のために定義されている。

; ポイントコードが16もしくは24ビットという二つのナショナルオプションが存在する。

; mtpAddressをオクテット毎に一行並べるために、MSBは0として符号化されなくてはならない。

; 1オクテットは2つの16進数によって表現される。

mtpAddress                = MTPToken LBRKT 4\*8 (HEXDIG) RBRKT

termIDList                = (TerminationID / LBRKT TerminationID 1\*(COMMA  
                                 TerminationID) RBRKT

terminationIDList        = LBRKT TerminationID \*(COMMA TerminationID) RBRKT

; pathNAMEの全長は64文字を超えてはならない。

pathNAME                 = ["\*"] NAME \*("/" /\*"/ ALPHA / DIGIT / "\_" / "\$")  
                                 ["@" pathDomainName ]

; パスドメイン名では無意味であるが、ABNFは2つ以上の連続した"."を許す。

pathDomainName           = (ALPHA / DIGIT / "\*" ) \*63(ALPHA / DIGIT / "-" /  
                                 "\*" / ".")

; '\*'はALL。'\$'はCHOOSE。

TerminationID            = "ROOT" / pathNAME / "\$" / "\*"

mediaDescriptor = MediaToken LBRKT mediaParm \*(COMMA mediaParm) RBRKT

; 多くて一つのterminationStateDescriptor

; かつstreamParmかstreamDescriptorのいずれかだが両方ではない

mediaParm                = (streamParm / streamDescriptor /  
                                 terminationStateDescriptor)

; 項目毎に多くて一度

streamParm                = ( localDescriptor / remoteDescriptor /  
                                 localControlDescriptor / statisticsDescriptor)

streamDescriptor         = StreamToken EQUAL StreamID LBRKT streamParm \*(COMMA  
                                 streamParm) RBRKT

localControlDescriptor = LocalControlToken LBRKT localParm \*(COMMA localParm)  
                                 RBRKT

; propertyParmを除く項目毎に多くて一度

localParm = ( streamMode / propertyParm / reservedValueMode /  
reservedGroupMode )

reservedValueMode = ReservedValueToken EQUAL ( "ON" / "OFF" )

reservedGroupMode = ReservedGroupToken EQUAL ( "ON" / "OFF" )

streamMode = ModeToken EQUAL streamModes

streamModes = (SendonlyToken / RecvonlyToken / SendrecvToken /  
InactiveToken / LoopbackToken )

propertyParm = pkgdName parmValue

; (セーフ)キャラクタ'\$'はCHOOSEを意味する

; (セーフ)キャラクタ'\*'はALLを意味する

parmValue = (EQUAL alternativeValue/ INEQUAL VALUE)

alternativeValue = ( VALUE  
/ LSBRKT VALUE \*(COMMA VALUE) RBRKT  
; サブリスト (つまり、A AND B AND...)  
/ LBRKT VALUE \*(COMMA VALUE) RBRKT  
; 選択肢(つまり、A OR B OR...)  
/ LSBRKT VALUE COLON VALUE RBRKT )  
; 範囲

INEQUAL = LWSP (">" / "<" / "#") LWSP ; #は “等しくない” を意味する

LSBRKT = LWSP "[" LWSP

RSBRKT = LWSP "]" LWSP

;注記 ・オクテットゼロはオクテット列の中の許可された文字中にはない。現在の定義はSDPに限  
;られており、ゼロオクテットはSDPの中で合法的な文字ではないため、影響はない。

localDescriptor = LocalToken LBRKT octetString RBRKT

remoteDescriptor = RemoteToken LBRKT octetString RBRKT

eventBufferDescriptor= EventBufferToken [ LBRKT eventSpec \*( COMMA eventSpec )  
RBRKT ]

eventSpec = pkgdName [ LBRKT eventSpecParameter \*(COMMA  
eventSpecParameter) RBRKT ]

eventSpecParameter = (eventStream / eventOther)

eventBufferControl = BufferToken EQUAL eventBufferControlValue

eventBufferControlValue = ( "OFF" / LockStepToken )

terminationStateDescriptor = TerminationStateToken LBRKT terminationStateParm  
\*( COMMA terminationStateParm ) RBRKT

; propertyParmを除く項目毎に多くて一度

terminationStateParm =(propertyParm / serviceStates / eventBufferControl )

serviceStates = ServiceStatesToken EQUAL serviceStatesValue

serviceStatesValue = ( TestToken / OutOfSvcToken / InSvcToken )

muxDescriptor = MuxToken EQUAL MuxType terminationIDList

MuxType = ( H221Token / H223Token / H226Token / V76Token /  
extensionParameter / Nx64kToken )

StreamID = UINT16

pkgdName = (PackageName SLASH ItemID) ;特定の項目  
/ (PackageName SLASH "\*") ;パッケージの全項目  
/ ("\*" SLASH "\*") ; MGによってサポートされる全項目

PackageName = NAME

ItemID = NAME

eventsDescriptor = EventsToken [ EQUAL RequestID LBRKT requestedEvent  
\*( COMMA requestedEvent ) RBRKT ]

requestedEvent = pkgdName [ LBRKT eventParameter \*( COMMA  
eventParameter ) RBRKT ]

notifyRegulated = NotifyRegulatedToken [LBRKT (embedWithSig /  
embedNoSig) RBRKT ]

notifyBehaviour = NotifyImmediateToken / notifyRegulated /  
NeverNotifyToken

; KeepActiveToken , notifyBehaviour , ResetEventsDescriptor , eventDM, eventStreamのそれぞれで多くて一度

; embedWithSigまたはembedNoSigのいずれか(両方ではない)で多くて一つ

;KeepActiveTokenとembedWithSigは同時に存在してはならない

eventParameter = ( embedWithSig / embedNoSig / KeepActiveToken  
/eventDM / eventStream / eventOther /  
notifyBehaviour / ResetEventsDescriptorToken)

embedWithSig = EmbedToken LBRKT signalsDescriptor [COMMA

```

        embedFirst ] RBRKT

embedNoSig      = EmbedToken LBRKT embedFirst RBRKT

; それぞれ多くて一度
embedFirst      = EventsToken [ EQUAL RequestID LBRKT
                        secondRequestedEvent *(COMMA secondRequestedEvent)
                        RBRKT ]

secondRequestedEvent = pkgdName [ LBRKT secondEventParameter *( COMMA
                        secondEventParameter ) RBRKT ]

; embedSig , KeepActiveToken, notifyBehaviour , eventDM , ResetEventsDescriptorまたは
; eventStreamのそれぞれが多くて一度
; KeepActiveTokenとembedSigは同時に存在してはならない
secondEventParameter = ( embedSig / KeepActiveToken / eventDM /
                        eventStream / eventOther / notifyBehaviour /
                        ResetEventsDescriptorToken)

embedSig  = EmbedToken LBRKT signalsDescriptor RBRKT

eventStream      = StreamToken EQUAL StreamID

eventOther       = eventParameterName parmValue

eventParameterName = NAME

eventDM          = DigitMapToken EQUAL ((digitMapName ) / (LBRKT
                        digitMapValue RBRKT ))

signalsDescriptor = SignalsToken [LBRKT signalParm *(COMMA signalParm)
                        RBRKT]

signalParm       = signalList / signalRequest

signalRequest    = signalName [ LBRKT sigParameter *(COMMA sigParameter)
                        RBRKT ]

signalList       = SignalListToken EQUAL signalListId LBRKT
                        signalListParm *(COMMA signalListParm) RBRKT

signalListId     = UINT16

```



; signalTypeは正確に一度。 durationとsignal

;parameter毎に多くて一度

signalListParm = signalRequest

signalName = pkgdName

;sigStreamは多くて一度。 sigSignalTypeは多くて一度。

;sigDurationは多くて一度。 sigDirectionは多くて一度。

;sigRequestIDは多くて一度。 sigIntsigDelayは多くて一度。

;signalParameterName毎に多くて一度

sigParameter = sigStream / sigSignalType / sigDuration / sigOther /  
notifyCompletion / KeepActiveToken /  
sigDirection / sigRequestID

sigStream = StreamToken EQUAL StreamID

sigOther = sigParameterName parmValue

sigParameterName = NAME

sigSignalType = SignalTypeToken EQUAL signalType

signalType = (OnOffToken / TimeOutToken / BriefToken)

sigDuration = DurationToken EQUAL UINT16

sigDirection = DirectionToken EQUAL direction

direction = ExternalToken / InternalToken / BothToken

sigRequestID = RequestIDToken EQUAL RequestID

sigIntsigDelay = IntsigDelayToken EQUAL UINT16

notifyCompletion = NotifyCompletionToken EQUAL (LBRKT  
notificationReason \*(COMMA notificationReason)  
RBRKT)

notificationReason = TimeOutToken / InterruptByEventToken /  
InterruptByNewSignalsDescrToken / OtherReasonToken /  
IterationToken

observedEventsDescriptor = ObservedEventsToken EQUAL RequestID LBRKT  
observedEvent \*(COMMA observedEvent) RBRKT

;timeはバッファリングされることがあるのでイベント毎

observedEvent = [ TimeStamp LWSP COLON] LWSP pkgdName [ LBRKT  
observedEventParameter \*(COMMA  
observedEventParameter) RBRKT ]

;eventStreamは多くて一度。 eventParameterNameは多くて一度

observedEventParameter = eventStream / eventOther

;全てのイベントを持ったAuditCapReplyに対しては、RequestIDはALLであるべき。

RequestID = UINT32 / “\*”

modemDescriptor = ModemToken (( EQUAL modemType) / (LSBRKT modemType  
\*(COMMA modemType) RSBKKT)) [ LBRKT propertyParm  
\*(COMMA propertyParm) RBRKT ]

; extensionParameterは除いて多くて一度

modemType = (V32bisToken / V22bisToken / V18Token / V22Token /  
V32Token / V34Token / V90Token / V91Token /  
SynchISDNToken / extensionParameter)

digitMapDescriptor = DigitMapToken EQUAL (( LBRKT digitMapValue RBRKT )/  
(digitMapName [ LBRKT digitMapValue RBRKT ]))

digitMapName = NAME

digitMapValue = ["T" COLON Timer COMMA] ["S" COLON Timer COMMA] ["L"  
COLON Timer COMMA] ["Z" COLON Timer COMMA] digitMap

Timer = 1\*2DIGIT

; T, S, Lタイマの単位は秒、Zタイマの単位は百ミリ秒。

; したがって、T, S, Lの範囲は1から99秒、Zは100ミリ秒から9.9秒まで。

digitMap = (digitString / LWSP "(" LWSP digitStringList  
LWSP  
")" LWSP

digitStringList = digitString \*( LWSP "|" LWSP digitString )

digitString = 1\*(digitStringElement)

digitStringElement = digitPosition [DOT]

digitPosition = digitMapLetter / digitMapRange

digitMapRange = ("x" / ( LWSP "[" LWSP digitLetter LWSP "]" LWSP ) )

digitLetter = \*((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter = DIGIT ;基本イベントシンボル  
/ %x41-4B / %x61-6B ; a-k, A-K  
/ "L" / "S" / "T" ; イベント間タイマ  
;(ロングタイマ、ショートタイマ、スタートタイマ)  
/ "Z" ; 長期継続修飾子

;多くて一度。DigitMapTokenとPackagesTokenはAuditCapabilitiesコマンドでは非許容

```
auditItem = auditReturnItem / SignalsToken / EventBufferToken /  
EventsToken / indAudterminationAudit
```

```
indAudterminationAudit = indAudauditReturnParameter *(COMMA  
indAudauditReturnParameter)
```

```
indAudauditReturnParameter = indAudmediaDescriptor / indAudeventsDescriptor /  
indAudsignalsDescriptor /  
indAuddigitMapDescriptor /  
indAudeventBufferDescriptor /  
indAudstatisticsDescriptor /  
indAudpackagesDescriptor
```

```
indAudmediaDescriptor = MediaToken LBRKT indAudmediaParm *(COMMA  
indAudmediaParm) RBRKT
```

; streamParm かstreamDescriptorのどちらかであり、両方ではない。

```
indAudmediaParm = indAudstreamParm / indAudstreamDescriptor /  
indAudterminationStateDescriptor)
```

;多くて一度

```
indAudstreamParm = (indAudlocalControlDescriptor /  
indAudstatisticsDescriptor /  
indAudremoteDescriptor / indAudlocalDescriptor )
```

```
indAudremoteDescriptor = RemoteToken LBRKT octetString RBRKT
```

```
indAudlocalDescriptor = LocalToken LBRKT octetString RBRKT
```

```
indAudstreamDescriptor = StreamToken EQUAL StreamID LBRKT indAudstreamParm  
RBRKT
```

```
indAudlocalControlDescriptor = LocalControlToken LBRKT indAudlocalParm  
*(COMMA indAudlocalParm) RBRKT
```

; 項目毎に多くて一度

```
indAudlocalParm = ModeToken [(EQUAL / INEQUAL)  
streamModes] / pkgdName /  
propertyParm / ReservedValueToken /  
ReservedGroupToken
```

; propertyParmとstreamModeは審査(audit)選択基準を指定する目的にのみ使用される  
; AND/OR選択ロジックはcontextレベルで指定される

indAudterminationStateDescriptor = TerminationStateToken LBRKT  
indAudterminationStateParm RBRKT

; 項目毎に多くて一度

indAudterminationStateParm = pkgdName / propertyParm / ServiceStatesToken /  
[(EQUAL / INEQUAL) serviceStatesValue ] / BufferToken

; 値が含まれるとき、選択操作が含まれる

; AND/OR選択ロジックはcontextレベルで指定される

indAudeventBufferDescriptor= EventBufferToken LBRKT indAudeventSpec RBRKT

indAudeventSpec = pkgdName [ LBRKT indAudeventSpecParameter RBRKT ]

indAudeventSpecParameter = (eventStream / eventParameterName)

indAudeventsDescriptor = EventsToken EQUAL RequestID LBRKT  
indAudrequestedEvent RBRKT

indAudrequestedEvent = pkgdName

indAudsignalsDescriptor = SignalsToken LBRKT [ indAudsignalParm ] RBRKT

indAudsignalParm = indAudsignalList / indAudsignalRequest

indAudsignalRequest = signalName [LBRKT indAudsignalRequestParm  
\*(COMMA indAudsignalRequestParm) RBRKT]

indAudsignalRequestParm = sigStream / sigRequestID

indAudsignalList = SignalListToken EQUAL signalListId [LBRKT  
indAudsignalListParm RBRKT]

indAudsignalListParm = indAudsignalRequest

indAuddigitMapDescriptor = DigitMapToken EQUAL (digitMapName )

indAudstatisticsDescriptor = StatsToken LBRKT pkgdName RBRKT

indAudpackagesDescriptor = PackagesToken LBRKT packagesItem RBRKT

```
serviceChangeDescriptor = ServicesToken LBRKT serviceChangeParm *(COMMA
serviceChangeParm) RBRKT
```

; 各パラメータ多くて一度、但しauditItemは除く

; せいぜいserviceChangeAddressかserviceChangeMgcIdのどちらか一つであり、両方ではない

; serviceChangeMethodとserviceChangeReasonが必要とされる。

```
serviceChangeParm = (serviceChangeMethod / serviceChangeReason /
serviceChangeDelay / serviceChangeAddress /
serviceChangeProfile / extension / TimeStamp /
serviceChangeMgcId / serviceChangeVersion /
serviceChangeIncompleteToken / auditItem )
```

```
serviceChangeReplyDescriptor = ServicesToken LBRKT servChgReplyParm *(COMMA
servChgReplyParm) RBRKT
```

; 多くて一度。最初のServiceChange応答にVersionが必要とされる。

; せいぜいserviceChangeAddressかserviceChangeMgcIdのどちらか一つであり、両方ではない

```
servChgReplyParm = serviceChangeAddress / serviceChangeMgcId /
serviceChangeProfile / serviceChangeVersion /
TimeStamp
```

```
serviceChangeMethod = MethodToken EQUAL (FailoverToken / ForcedToken /
GracefulToken / RestartToken / DisconnectedToken /
HandOffToken / extensionParameter)
```

; serviceChangeReasonは数字の理由コードとオプション的なテキスト記述から構成される。

; serviceChangeReasonはquotedString型VALUEを使い符号化されなければならない。

; quotedStringは10進数の理由コードを含む、オプション的に1つのスペース文字とテキスト記述

; 列が続く。

```
serviceChangeReason = ReasonToken EQUAL VALUE
```

```
serviceChangeDelay = DelayToken EQUAL UIN32
```

```
serviceChangeAddress = ServiceChangeAddressToken EQUAL (mId / portNumber)
```

```
serviceChangeMgcId = MgcIdToken EQUAL mId
```

```
serviceChangeProfile = ProfileToken EQUAL NAME SLASH Version
```

```
serviceChangeVersion = VersionToken EQUAL Version
```

```
extension = extensionParameter parmValue
```

packagesDescriptor = PackagesToken LBRKT packagesItem \*(COMMA  
 packagesItem) RBRKT

Version = 1\*2(DIGIT)

packagesItem = NAME "-" UINT16

TimeStamp = Date "T" Time ; per ISO 8601:1988

; Date = yyyyymmdd

Date = 8(DIGIT)

; Time = hhmmssss

Time = 8(DIGIT)

statisticsDescriptor = StatsToken LBRKT statisticsParameter \*(COMMA  
 statisticsParameter ) RBRKT

;項目毎に多くて一度

statisticsParameter = pkgdName [EQUAL VALUE /  
 (LSBRKT VALUE \*(COMMA VALUE) RSBKRT) ]

topologyDescriptor = TopologyToken LBRKT topologyTriple \*(COMMA  
 topologyTriple) RBRKT

topologyTriple = terminationA COMMA terminationB COMMA  
 topologyDirection [COMMA eventStream]

terminationA = TerminationID

terminationB = TerminationID

topologyDirection = BothwayToken / IsolateToken / OnewayToken /  
 OnewayExternalToken / OnewayBothToken

priority = PriorityToken EQUAL UINT16

iepValue = IEPSToken EQUAL ("ON" / "OFF")

emergencyValue = EmergencyValueToken EQUAL (EmergencyToken /  
 EmergencyOffToken)

contextAttrDescriptor = ContextAttrToken LBRKT (contextIdList /  
 propertyParm \*(COMMA propertyParm)) RBRKT

;contextIdList構成を使用する時は、actionReply構成にあるContextIDは  
;関連したactionRequestにあるContextIDと同じでなければならない

contextIdList = ContextListToken EQUAL LBRKT ContextID  
\*(COMMA ContextID) RBRKT

extensionParameter = "X" ("-" / "+") 1\*6(ALPHA / DIGIT)

;octetStringはRFC2327で定義されるSDPを記述するために使用される。

;RFC2327のCRLFが使用される場合、注意が必要。

;安全のためにこのABNFではEOLを使用する。

;SDPに"}"が現れるときは常に、"}"でエスケープする。例えば"}"}

octetString = \*(nonEscapeChar)

nonEscapeChar = ( "}" / %x01-7C / %x7E-FF )

;注意 ・2重引用文字はquotedStringの中では許されない。

quotedString = DQUOTE \*(SafeChar / EOL / %x80-EF / RestChar/ WSP)  
DQUOTE

UINT16 = 1\*5(DIGIT) ; %x0-FFFF

UINT32 = 1\*10(DIGIT) ; %x0-FFFFFFFF

NAME = ALPHA \*63(ALPHA / DIGIT / "\_")

VALUE = quotedString / 1\*(SafeChar / %x80-EF)

SafeChar = DIGIT / ALPHA / "+" / "-" / "&" /  
"!"/ "\_"/ "/" / "" / "?" / "@" /  
"^" / "~" / "\*" / "\$" / "}" /  
"("/ ")" / "%" / "|" / "."

EQUAL = LWSP %x3D LWSP ; "="

COLON = %x3A ; ":"

LBRKT = LWSP %x7B LWSP ; "{"

RBRKT = LWSP %x7D LWSP ; "}"

COMMA = LWSP %x2C LWSP ; ","

DOT = %x2E ; "."

SLASH = %x2F ; "/"

ALPHA	= %x41-5A / %x61-7A ; A-Z / a-z
DIGIT	= %x30-39 ; 0-9
DQUOTE	= %x22 ; " (Double Quote)
HEXDIG	= DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
SP	= %x20 ; space
HTAB	= %x09 ; horizontal tab
CR	= %x0D ; Carriage return
LF	= %x0A ; linefeed
LWSP	= *( WSP / COMMENT / EOL)
EOL	= (CR [LF] / LF)
WSP	= SP / HTAB ; white space
SEP	= ( WSP / EOL / COMMENT) LWSP
COMMENT	= ";" *(SafeChar/ RestChar / WSP / %x22) EOL
RestChar	= ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" / " < " / "> " / "="

;sigParameterに追加された新たなトークンはSPA\*の書式をとらなければならない。

;SPAMのように \*はどのような形式でもよい。

;eventParameterに追加された新たなトークンはEPA\*の書式をとらなければならない。

;EPADのように \*はどのような形式でもよい。

AddToken	= ("Add" / "A")
AndAUDITSelectToken	= ("ANDLgc")
AuditToken	= ("Audit" / "AT")
AuditCapToken	= ("AuditCapability" / "AC")
AuditValueToken	= ("AuditValue" / "AV")
AuthToken	= ("Authentication" / "AU")
BothToken	= ("Both" / "B")
BothwayToken	= ("Bothway" / "BW")
BriefToken	= ("Brief" / "BR")



BufferToken	= ("Buffer"	/ "BF")
CtxToken	= ("Context"	/ "C")
ContextAuditToken	= ("ContextAudit"	/ "CA")
ContextAttrToken	= ("ContextAttr"	/ "CT")
ContextListToken	= ("ContextList"	/ "CLT")
DigitMapToken	= ("DigitMap"	/ "DM")
DirectionToken	= ("SPADirection"	/ "SPADI")
DisconnectedToken	= ("Disconnected"	/ "DC")
DelayToken	= ("Delay"	/ "DL")
DurationToken	= ("Duration"	/ "DR")
EmbedToken	= ("Embed"	/ "EM")
EmergencyToken	= ("Emergency"	/ "EG")
EmergencyOffToken	= ("EmergencyOff"	/ "EGO")
EmergencyValueToken	= ("EmergencyValue"	/ "EGV")
ErrorToken	= ("Error"	/ "ER")
EventBufferToken	= ("EventBuffer"	/ "EB")
EventsToken	= ("Events"	/ "E")
ExternalToken	= ("External"	/ "EX")
FailoverToken	= ("Failover"	/ "FL")
ForcedToken	= ("Forced"	/ "FO")
GracefulToken	= ("Graceful"	/ "GR")
H221Token	= ("H221" )	
H223Token	= ("H223" )	
H226Token	= ("H226" )	
HandOffToken	= ("HandOff"	/ "HO")
IEPSToken	= ("IEPSCall"	/ "IEPS")
ImmAckRequiredToken	= ("ImmAckRequired"	/ "IA")
InactiveToken	= ("Inactive"	/ "IN")
InternalToken	= ("Internal"	/ "IT")
IntsigDelayToken	= ("Intersignal"	/ "SPAIS")
IsolateToken	= ("Isolate"	/ "IS")
InSvcToken	= ("InService"	/ "IV")
InterruptByEventToken	= ("IntByEvent"	/ "IBE")
InterruptByNewSignalsDescrToken		
	= ("IntBySigDescr"	/ "IBS")
IterationToken	= ("Iteration"	/ "IR")
KeepActiveToken	= ("KeepActive"	/ "KA")
LocalToken	= ("Local"	/ "L")
LocalControlToken	= ("LocalControl"	/ "O")
LockStepToken	= ("LockStep"	/ "SP")
LoopbackToken	= ("Loopback"	/ "LB")
MediaToken	= ("Media"	/ "M")
MegacopToken	= ("MEGACO"	/ "I")

MessageSegmentToken	= ("Segment"	/ "SM")
MethodToken	= ("Method"	/ "MT")
MgcIdToken	= ("MgcIdToTry"	/ "MG")
ModeToken	= ("Mode"	/ "MO")
ModifyToken	= ("Modify"	/ "MF")
ModemToken	= ("Modem"	/ "MD")
MoveToken	= ("Move"	/ "MV")
MTPToken	= ("MTP")	
MuxToken	= ("Mux"	/ "MX")
NeverNotifyToken	= ("NeverNotify"	/ "NBNN")
NotifyToken	= ("Notify"	/ "N")
NotifyCompletionToken	= ("NotifyCompletion"	/ "NC")
NotifyImmediateToken	= ("ImmediateNotify"	/ "NBIN")
NotifyRegulatedToken	= ("RegulatedNotify"	/ "NBRN")
Nx64kToken	= ("Nx64Kservice"	/ "N64")
ObservedEventsToken	= ("ObservedEvents"	/ "OE")
OnewayToken	= ("Oneway"	/ "OW")
OnewayBothToken	= ("OnewayBoth"	/ "OWB")
OnewayExternalToken	= ("OnewayExternal"	/ "OWE")
OnOffToken	= ("OnOff"	/ "OO")
OrAUDITselectToken	= ("ORLgc")	
OtherReasonToken	= ("OtherReason"	/ "OR")
OutOfSvcToken	= ("OutOfService"	/ "OS")
PackagesToken	= ("Packages"	/ "PG")
PendingToken	= ("Pending"	/ "PN")
PriorityToken	= ("Priority"	/ "PR")
ProfileToken	= ("Profile"	/ "PF")
ReasonToken	= ("Reason"	/ "RE")
RecvonlyToken	= ("ReceiveOnly"	/ "RC")
ReplyToken	= ("Reply"	/ "P")
Reset EventsDescriptorToken	= ("ResetEventsDescriptor"	/ "RSE")
RestartToken	= ("Restart"	/ "RS")
RemoteToken	= ("Remote"	/ "R")
RequestIDToken	= ("SPARrequestID"	/ "SPARQ")
ReservedGroupToken	= ("ReservedGroup"	/ "RG")
ReservedValueToken	= ("ReservedValue"	/ "RV")
SegmentationCompleteToken	= ("END"	/ "&")
SendonlyToken	= ("SendOnly"	/ "SO")
SendrecvToken	= ("SendReceive"	/ "SR")
ServicesToken	= ("Services"	/ "SV")
ServiceStatesToken	= ("ServiceStates"	/ "SI")
ServiceChangeIncompleteToken	= ("ServiceChangeInc"	/ "SIC")
ServiceChangeToken	= ("ServiceChange"	/ "SC")

ServiceChangeAddressToken	= ("ServiceChangeAddress" / "AD")
SignalListToken	= ("SignalList" / "SL")
SignalsToken	= ("Signals" / "SG")
SignalTypeToken	= ("SignalType" / "SY")
StatsToken	= ("Statistics" / "SA")
StreamToken	= ("Stream" / "ST")
SubtractToken	= ("Subtract" / "S")
SynchISDNToken	= ("SynchISDN" / "SN")
TerminationStateToken	= ("TerminationState" / "TS")
TestToken	= ("Test" / "TE")
TimeOutToken	= ("TimeOut" / "TO")
TopologyToken	= ("Topology" / "TP")
TransToken	= ("Transaction" / "T")
ResponseAckToken	= ("TransactionResponseAck"/ "K")
V18Token	= ("V18")
V22Token	= ("V22")
V22bisToken	= ("V22b")
V32Token	= ("V32")
V32bisToken	= ("V32b")
V34Token	= ("V34")
V76Token	= ("V76")
V90Token	= ("V90")
V91Token	= ("V91")
VersionToken	= ("Version" / "V")

### B.3 16進オクテット符号化

16進オクテット符号化は16進数の文字列でオクテット列を表現するための手段である。1 オクテットは2桁で表される。このオクテット符号化はオクテット列が本プロトコルのテキスト符号化の際に使用されるべきである。

それぞれのオクテットについて、8ビットの列は2桁の16進数に変換される。ビット0は最初に送られたものである。ビット7は最後に送られたものである。

ビット7-4は最初の16進数として変換され、ビット7がMSB、ビット4がLSBである。ビット3-0は第2番目の16進数として変換され、ビット3がMSB、ビット0がLSBである。

例:

オクテットビットパターン	16進数符号
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

### B.4 16進オクテット列

16進オクテット列は偶数個の16進数からなり、<CR> 記号によって終了する。

## 付属資料C メディアストリーム・プロパティのタグ (TAGS FOR MEDIA STREAM PROPERTIES)

(本付属資料は本標準の必須部分である)

Local、RemoteおよびLocalControlの各ディスクリプタに対するパラメータは、もしプロトコルに対しバイナリ符号化が使用されるなら、タグと値の組として指定される。本付属資料はプロパティの名前(PropertyID)、タグ(Property Tag)、プロパティの型(Type)、および値(Value)を含む。値フィールドに示される値は、そのフィールドが参照を含む時、“情報”と見なされなければならない。その参照は規範的な値を含む。もしある値フィールドが参照を含まないなら、そのフィールドにおける値は“規範的”と見なされることできる。

付属資料Cを参照すると、プロパティはPackageID/PropertyID構造に従う。しかしながら、付属資料Cはそれ自身の中のパッケージではない。付属資料Cは、binary encoding (符号化) に対するパッケージID0x0000とtext encoding (符号化) に対する“anxc”を持っているとみなされる。H.248.1のtext encoding (符号化) に対して、要求されるプロパティがパッケージによって定義されていないか、もしくはSDPによって表されていないなら、付属資料Cのみが使われるべきである。別のプロパティの中に付属資料Cのプロパティを入れ子することは禁止されている。

本付属資料ではタグは16進数の番号として与えられる。プロパティの値を設定する時、MGCは節7.1.1にて指定されるメカニズムのうちの一つに従ってその値を従属指定してもよい。

本付属資料のプロパティのサポートはオプションである。例えば、本付属資料のC.3から3プロパティとC.8から5プロパティだけを実装してもよい。

“列挙型”(enumeration)では値は括弧内の値により表現される、例えば、送信(0)、受信(1)。“N bits”もしくは“M Octets”の型を持つ本付属資料のプロパティは、本プロトコルの符号化時、オクテット列(Octet String)として扱われるべきである。“N bit integer”型のプロパティは整数として扱われなくてはならない。“String”は、本プロトコルの符号化時、IA5Stringとして扱われなくてはならない。

型が1オクテットより小さい時、その値はサイズ1のオクテット列の下位ビットに入れられなければならない。

## C.1 一般メディア属性 (General Media Attributes)

PropertyID	PropertyTag	Type	Value
Media	1001	Enumeration	Audio(0), Video(1), Data(2),
Transmission mode	1002	Enumeration	Send(0), Receive(1), Send&Receive(2)
Number of Channels	1003	Unsigned Integer	0-255
Sampling rate	1004	Unsigned Integer	0-2 <sup>32</sup>
Bitrate	1005	Integer	(0..4294967295) 注記：100 bit/s単位
Acodec	1006	Octet String	Audio Codec Type: 参照: ITU-T勧告Q.765.5 非ITUコーデックは定義された Organizational Identifierの下で適当な標準化機構で定義される。
Samplepp	1007	Unsigned Integer	Maximum samples or frames per packet: 0-65535
Silencesupp	1008	BOOLEAN	Silence Suppression: True/false
Encrypttype	1009	Octet string	参照: ITU-T勧告H.245
Encryptkey	100A	Octet string SIZE(0..65535)	Encryption key 参照: ITU-T勧告H.235
Echocanc	100B		使用しない。JT-H248.1 付属資料E.13のPossible Echo Control propertiesの例を参照
Gain	100C	Unsigned Integer	使用しない。JT-H248.1付属資料E.13の利用可能な gain propertyを参照
Jitterbuff	100D	Unsigned Integer	Jitter buffer size、ミリ秒単位: 0-65535
PropDelay	100E	Unsigned Integer	Propagation Delay: 0..65535 2MG間のベアラコネクションの最大伝播遅延。ミリ秒単位。最大遅延はベアラ技術に依存する。
RTPpayload	100F	Integer	Payload type in RTP Minimal ControlによるAudio/Video Conferenceのプロファイル 参照: IETF RFC 1890
Prime	1010	Integer	Packetization Time これはパケット内のメディアによって表されるミリ秒の時間長を与える。 参照: IETF RFC 2327

## C.2 Muxプロパティ (Mux Properties)

PropertyID	PropertyTag	Type	Value
H.221	2001	Octet string	参照: ITU-T勧告H.245, H222LogicalChannelParameters
H223	2002	Octet string	参照: ITU-T勧告H.245, H223LogicalChannelParameters
V76	2003	Octet String	参照: ITU-T勧告H.245, V76LogicalChannelParameters
H2250	2004	Octet String	参照: ITU-T勧告H.245, H2250LogicalChannelParameters

## C.3 一般ベアラプロパティ (General bearer properties)

PropertyID	PropertyTag	Type	Value
Mediatx	3001	Enumeration	Media Transport Type: TDM Circuit(0), ATM(1), FR(2), Ipv4(3), Ipv6(4), ...
BIR	3002	4 OCTET	値はトランスポート技術に依存する
NSAP	3003	1-20 OCTETS	NSAP参照 参照: ITU-T勧告X.213 Annex A

## C.4 一般ATMプロパティ (General ATM properties)

PropertyID	PropertyTag	Type	Value
AESA	4001	20 OCTETS	ATM End System Address
VPVC	4002	4 OCTETS, 最初の最下位2 オクテット: VPCI 次の2オクテッ ト: VCI	VPCI/VCI 参照: ITU-T勧告Q.2931
SC	4003	Enumeration	Service Category 参照: ATM Forum UNI 4.0 CBR(0), nrt-VBR1(1), nrt-VBR2(2), nrt-VBR3(3), rt-VBR1(4), rt-VBR2(5), rt-VBR3(6), UBR1(7), UBR2(8), ABR(9).
BCOB	4004	5 bit integer	Broadband Bearer Class 参照: ITU-T勧告Q.2961.2
BBTC	4005	7 bit integer	Broadband Transfer Capability: 参照: ITU-T勧告Q.2961
ATC	4006	Enumeration	I.371 ATM Traffic Capability 参照: ITU-T勧告I.371: DBR(0), SBR1(1), SBR2(2), SBR3(3), ABT/IT(4), ABT/DT(5), ABR(6)
STC	4007	2 bits	Susceptibility to clipping: 参照: ITU-T勧告Q.2931 00 Susceptible 01 Not-susceptible

PropertyID	PropertyTag	Type	Value
UPCC	4008	2 bits	User Plane Connection configuration: 参照: ITU-T勧告Q.2931 00 Pt-to-pt, 01 Pt-to-mpt
PCR0	4009	24 bit integer	Peak Cell Rate (For CLP=0) 参照: ITU-T勧告Q.2931
SCR0	400A	24 bit integer	Sustainable Cell Rate (For CLP=0) 参照: ITU-T勧告Q.2961
MBS0	400B	24 bit integer	Maximum Burst Size (For CLP=0) 参照: ITU-T勧告Q.2961
PCR1	400C	24 bit integer	Peak Cell Rate (For CLP=0+1) 参照: ITU-T勧告Q.2931
SCR1	400D	24 bit integer	Sustainable Cell Rate (For CLP=0+1) 参照: ITU-T勧告Q.2961
MBS1	400E	24 bit integer	Maximum Burst Size (For CLP=0+1) 参照: ITU-T勧告Q.2961
BEI	400F	Boolean	Best Effort Indicator. 参照: ATM Forum UNI 4.0 0 ATMシグナリングにおいてBEIを含まない 1 ATMシグナリングにおいてBEIを含む
TI	4010	Boolean	Tagging Indicator. 参照: ITU-T勧告Q.2961 0 taggingは許されない 1 taggingが要求される
FD	4011	Boolean	Frame Discard. 参照: ATM Forum UNI 4.0 0 frame discardが許されない 1 frame discardが許される
A2PCDV	4012	24 bit integer	Acceptable 2-point CDV. 参照ITU-T勧告Q.2965.2
C2PCDV	4013	24 bit integer	Cumulative 2-point CDV. 参照: ITU-T勧告Q.2965.2
APPCDV	4014	24 bit integer	Acceptable P-P CDV. 参照: ATM Forum UNI 4.0
CPPCDV	4015	24 bit integer	Cumulative P-P CDV. 参照: ATM Forum UNI 4.0
ACLR	4016	8 bit integer	Acceptable Cell Loss Ratio. 参照: ITU-T勧告Q.2965.2, ATM Forum UNI 4.0
MEETD	4017	16 bit integer	Maximum End-to-end transit delay. 参照 ITU Rec. Q.2965.2, ATM Forum UNI 4.0
CEETD	4018	16 bit integer	Cumulative End-to-end transit delay. 参照: ITU-T勧告Q.2965.2, ATM Forum UNI 4.0

PropertyID	PropertyTag	Type	Value	
QosClass	4019	Integer 0-5	Qos Class 参照: ITU-T勧告Q.2965.1	
			Qos Class	意味
			0	Default QoS ITU-T勧告Q.2961.2で定義されている通り、ATCに関連するQoS
			1	Stringent
			2	Tolerant
			3	Bi-level
			4	Unbounded
			5	Stringent Bi-level
AALtype	401A	1 OCTET	AAL Type 参照: ITU-T勧告Q.2931 00000000 AAL for voice 00000001 AAL type 1 00000010 AAL type 2 00000011 AAL type 3/4 00000101 AAL type 5 00010000 ユーザ定義のAAL	

#### C.5 フレームリレー

PropertyID	PropertyTag	Type	Value
DLCI	5001	Unsigned Integer	Data link connection id
CID	5002	Unsigned Integer	sub-channel id.
SID/Noiselevel	5003	Unsigned Integer	silence insertion descriptor
Primary Payload type	5004	Unsigned Integer	Primary Payload Type FAXとコーデックをカバー
RtcpbwRS	6005	Integer	RS RTCPバンド幅の修正子はデータ送信者（RTP仕様によって定義されるように）を動作させるように割り当てられたRTCP バンド幅を示唆する。 IETF RFC 3556参照
RtcpbwRR	6006	Integer	RR RTCPバンド幅の修正子はTRPセッション内のほかの参加者(例:受信者)に割り当てられたRTCPバンド幅を示唆する。 IETF RFC 3556参照

#### C.6 IP

PropertyID	PropertyTag	Type	Value
IPv4	6001	32 BITS	Ipv4Address:



		Ipv4Address	参照: IETF RFC791
IPv6	6002	128 BITS	IPv6 Address: 参照: IETF RFC2460
Port	6003	unsigned integer	0-65535
Porttype	6004	enumerated	TCP(0), UDP(1), SCTP(2)

### C.7 ATM AAL2

PropertyID	PropertyTag	Type	Value
AESA	7001	20 OCTETS	AAL2 service endpoint address ITU-T勧告Q.2630.1で定義されている通り ESEA NSEA
BIR	See 0	4 OCTETS	Served user generated reference ITU-T勧告Q.2630.1で定義されている通り SUGR
ALC	7002	12 OCTETS	AAL2 link characteristics ITU-T勧告Q.2630.1で定義されている通り max/average CPS-SDU bitrate, max/average CPS-SDU size
SSCS	7003	I.366.2: audio (8 OCTETS) multirate (3 OCTETS) or I.366.1: SAR-assured (14 OCTETS)/ unassured (7 OCTETS)	Service specific convergence sublayer information ITU-T勧告Q.2630.1で定義され、ITU-T勧告 I.366.1およびITU-T勧告I.366.2で使用されてい る通り I.366.2: audio/multirateI. I.366.1: SAR-assured/unassured
SUT	7004	1..254 octets	Served user transport parameter ITU-T勧告Q.2630.1で定義されている通り
TCI	7005	BOOLEAN	Test connection indicator ITU-T勧告 Q.2630.1で定義されている通り
Timer_CU	7006	32 bit integer	Timer-CU 送信前にpartially filled cellを保持するための時 間。ミリ秒単位。
MaxCPSSDU	7007	8 bit integer	Maximum Common Part Sublayer Service Data Unit 参照: ITU-T勧告Q.2630.1
CID	7008	8 bits	subchannel id, 0-255 参照: ITU-T勧告I.363.2

## C.8 ATM AAL1

PropertyID	PropertyTag	Type	Value
BIR	C.3参照	29 OCTETS	GIT (Generic Identifier Transport) 参照: ITU-T勧告Q.2941.1
AAL1ST	8001	1 OCTET	AAL1 Subtype: 参照: ITU-T勧告Q.2931 00000000 Null 00000001 voiceband signal transport on 64kbit/s 00000010 circuit transport 00000100 high-quality audio signal transport 00000101 video signal transport
CBRR	8002	1 OCTET	CBR Rate 参照: ITU-T勧告Q.2931 00000001 64 kbit/s 00000100 1544 kbit/s 00000101 6312 kbit/s 00000110 32064 kbit/s 00000111 44736 kbit/s 00001000 97728 kbit/s 00010000 2048 kbit/s 00010001 8448 kbit/s 00010010 34368 kbit/s 00010011 139264 kbit/s 01000000 n x 64 kbit/s 01000001 n * 8 kbit/s
MULT	C.9参照		Multiplierもしくはn x 64k/8k/300 参照: ITU-T勧告Q.2931
SCRI	8003	1 OCTET	Source Clock Frequency Recovery Method: 参照: ITU-T勧告Q.2931 00000000 NULL 00000001 SRTS 00000010 ACM
ECM	8004	1 OCTET	Error Correction Method: 参照: ITU-T勧告Q.2931 00000000 Null 00000001 FEC-LOSS 00000010 FEC-DELAY
SDTB	8005	16 bit integer	Structured Data Transfer Blocksize: 参照: ITU-T勧告I.363.1 SDT CBRサービスのブロックサイズ
PFCI	8006	8 bit integer	Partially filled cells identifier: 参照: ITU-T勧告I.363.1 1-47

### C.9 ベアラ能力 (Bearer Capabilities)

下記表はITU-T勧告Q.931を参照している。Q.931の伝達能力情報(bearer capability information)要素の符号化を参照しており、下層レイヤ情報要素を参照しているわけではない。

PropertyID	PropertyTag	Type	Value
TMR	9001	1 OCTET	Transmission Medium Requirement (ITU-T勧告Q.763) 参照：ITU-T勧告Q.763 Bit 8 7 6 5 4 3 2 1 00000000 – 音声 00000001 – 予備 00000010 - 64 kbit/s非制限 00000011 - 3.1 kHzオーディオ 00000100 - 音声(サービス2)/64 kbit/s 非制限 (サービス1) 切替用に留保 00000101 - 64 kbit/s 非制限(サービス1)/音声(サービス2) 切替用に留保 00000110 - 64 kbit/s希望 00000111 - 2 × 64 kbit/s非制限 00001000 - 384 kbit/s非制限 00001001 - 1536 kbit/s非制限 00001010 - 1920 kbit/s非制限 00001011～00001111- 予備 00010000 - 3 × 64 kbit/s非制限 00010001 - 4 × 64 kbit/s非制限 00010010 - 5 × 64 kbit/s非制限 00010011 – 予備 00010100 - 7 × 64 kbit/s非制限 00010101 - 8 × 64 kbit/s 非制限 00010110 - 9 × 64 kbit/s 非制限 00010111 - 10 × 64 kbit/s 非制限 00011000 - 11 × 64 kbit/s 非制限 00011001 - 12 × 64 kbit/s 非制限 00011010 - 13 × 64 kbit/s 非制限 00011011 - 14 × 64 kbit/s 非制限 00011100 - 15 × 64 kbit/s 非制限 00011101 - 16 × 64 kbit/s 非制限 00011110 - 17 × 64 kbit/s 非制限 00011111 - 18 × 64 kbit/s 非制限 00100000 - 19 × 64 kbit/s 非制限 00100001 - 20 × 64 kbit/s 非制限 00100010 - 21 × 64 kbit/s 非制限 00100011 - 22 × 64 kbit/s 非制限 00100100 - 23 × 64 kbit/s 非制限

PropertyID	PropertyTag	Type	Value
			00100101 - 予備 00100110 - 25 × 64 kbit/s 非制限 00100111 - 26 × 64 kbit/s 非制限 00101000 - 27 × 64 kbit/s 非制限 00101001 - 28 × 64 kbit/s 非制限 00101010 - 29 × 64 kbit/s 非制限 00101011~11111111 予備
TMRSR	9002	1 OCTET	Transmission Medium Requirement Subrate 0 - 未定義 1 - 8kbit/s 2 - 16kbit/s 3 - 32kbit/s
Contcheck	9003	BOOLEAN	Continuity Check 参照: ITU-T勧告Q.763 0 - 該回線では要求されない 1 - 該回線で要求される
ITC	9004	5 BITS	Information Transfer Capability 参照: ITU-T勧告Q.763 Bits 5 4 3 2 1 00000 - 音声 01000 - 非制限デジタル情報 01001 - 非制限デジタル情報 10000 - 3.1 kHzオーディオ 10001 - トーン/アナウンス有り非制限デジタル情報 11000 - ビデオ 上記以外は予約済み
TransMode	9005	2 BITS	Transfer Mode 参照: ITU-T勧告Q.931 Bit 2 1 00 - 回線モード 10 - パケットモード
TransRate	9006	5 BITS	Transfer Rate 参照: ITU-T勧告Q.931 Bit 5 4 3 2 1 00000 - 本コードはパケットモードで使用されるべき 10000 - 64 kbit/s 10001 - 2 x 64 kbit/s 10011 - 384 kbit/s 10101 - 1536 kbit/s 10111 - 1920 kbit/s 11000 - Multirate (64 kbit/s base rate)

PropertyID	PropertyTag	Type	Value
MULT	9007	7 BITS	Rate Multiplier: 参照：ITU-T勧告Q.931 2からnまでの値(n：B-ch最大数)
Layer1prot	9008	5 BITS	User Information Layer 1 Protocol 参照：ITU-T勧告Q.931 Bits 5 4 3 2 1 00001 – CCITT standardized rate adaption V.110 and X.30. 00010 – Recommendation G.711 u-law 00011 – Recommendation G.711 A-law 00100 – Recommendation G.721 32 kbit/s ADPCM and Recommendation I.460. 00101 – Recommendations H.221 and H.242 00110 – Recommendations H.223 and H.245 00111 – Non-ITU-T standardized rate adaption. 01000 – ITU-T standardized rate adaption V.120. 01001 – CCITT standardized rate adaption X.31 HDLC flag stuffing. 上記以外予約済み
Syncasync	9009	BOOLEAN	Synchronous/Asynchronous 参照：ITU-T勧告Q.931 0 – 同期データ 1 – 非同期データ
Negotiation	900A	BOOLEAN	Negotiation 参照：ITU-T勧告Q.931 0 - インバンドネゴシエーション可能 1 - インバンドネゴシエーション不可
Userrate	900B	5 BITS	User Rate 参照：ITU-T勧告Q.931 Bits 5 4 3 2 1 00000 – レートはITU-T勧告I.460で規定されるE-bitsで指定されるかもしくはインバンドで取り決めてもよい。 00001 - 0.6 kbit/s Recommendations V.6 and X.1 00010 - 1.2 kbit/s Recommendation V.6 00011 - 2.4 kbit/s Recommendations V.6 and X.1 00100 - 3.6 kbit/s Recommendation V.6 00101 - 4.8 kbit/s Recommendations V.6 and X.1 00110 - 7.2 kbit/s Recommendation V.6 00111 - 8 kbit/s Recommendation I.460 01000 - 9.6 kbit/s Recommendations V.6 and X.1 01001 - 14.4 kbit/s Recommendation V.6 01010 - 16 kbit/s Recommendation I.460 01011 - 19.2 kbit/s Recommendation V.6

PropertyID	PropertyTag	Type	Value
			01100 - 32 kbit/s Recommendation I.460 01101 - 38.4 kbit/s Recommendation V.110 01110 - 48 kbit/s Recommendations V.6 and X.1 01111 - 56 kbit/s Recommendation V.6 10010 - 57.6 kbit/s Recommendation V.14 extended 10011 - 8.8 kbit/s Recommendation V.110 10100 - 24 kbit/s Recommendation V.110 10101 - 0.1345 kbit/s Recommendation X.1 10110 - 0.100 kbit/s Recommendation X.1 10111 - 0.075/1.2 kbit/s Recommendations V.6 and X.1 11000 - 1.2/0.075 kbit/s Recommendations V.6 and X.1 11001 - 0.050 kbit/s Recommendations V.6 and X.1 11010 - 0.075 kbit/s Recommendations V.6 and X.1 11011 - 0.110 kbit/s Recommendations V.6 and X.1 11100 - 0.150 kbit/s Recommendations V.6 and X.1 11101 - 0.200 kbit/s Recommendations V.6 and X.1 11110 - 0.300 kbit/s Recommendations V.6 and X.1 11111 - 12 kbit/s Recommendation V.6 上記以外は予約済み
INTRATE	900C	2 BITS	Intermediate Rate 参照：ITU-T勧告Q.931 Bit 2 1 00 - 未使用 01 - 8 kbit/s 10 - 16 kbit/s 11 - 32 kbit/s
nictx	900D	BOOLEAN	Network Independent Clock (NIC) on transmission 参照：ITU-T勧告Q.931 0 - NICのデータ送信は要求されない 1 - NICのデータ送信が要求される
nicrx	900E	BOOLEAN	Network independent clock (NIC) on reception 参照：ITU-T勧告Q.931 0 - NICのデータを受け付けられない(つまり、送信側はこのオプション手順をサポートしていない) 1 - NICのデータを受け付けられる(つまり、送信側はこのオプション手順をサポートしている)
flowconttx	900F	BOOLEAN	Flow Control on transmission (Tx) 参照：ITU-T勧告Q.931 0 - フロー制御機構によるデータ送信は要求されない 1 - フロー制御機構によるデータ送信が要求される

PropertyID	PropertyTag	Type	Value
flowcontrx	9010	BOOLEAN	Flow control on reception (Rx) 参照：ITU-T勧告Q.931 0 - フロー制御機構によるデータを受け付けられない(つまり、送信側はこのオプション手順をサポートしていない) 1 - フロー制御機構によるデータを受け付けられる(つまり、送信側はこのオプション手順をサポートしている)
rateadapthdr	9011	BOOLEAN	Rate adaption header/no header 参照：ITU-T勧告Q.931 0 - Rate adaption headerが含まれない 1 - Rate adaption headerが含まれる
multiframe	9012	BOOLEAN	Multiple frame establishment support in data link 参照：ITU-T勧告Q.931 0 - Multiple frame establishment未サポート。UI framesのみ許容 1 - Multiple frame establishmentサポート
OPMODE	9013	BOOLEAN	Mode of operation 参照：ITU-T勧告Q.931 0 Bit transparent mode of operation 1 Protocol sensitive mode of operation
llidnegot	9014	BOOLEAN	Logical link identifier negotiation 参照：ITU-T勧告Q.931 0 - Default, LLI = 256 only 1 - Full protocol negotiation
assign	9015	BOOLEAN	Assignor/assignee 参照：ITU-T勧告Q.931 0 - Message originatorが“Default assignee” 1 - Message originatorが“Assignor only”
inbandneg	9016	BOOLEAN	In-band/out-band negotiation 参照：ITU-T勧告Q.931 0 - Negotiationはtemporary signalling connection上でUSER INFORMATIONメッセージにより実施される。 1 - Negotiationは論理リンク0を使いインバンドで実施される。
stopbits	9017	2 BITS	Number of stop bits 参照：ITU-T勧告Q.931 Bits 2 1 00 - 未使用 01 - 1 bit 10 - 1.5 bits 11 - 2 bits

PropertyID	PropertyTag	Type	Value
databits	9018	2 BIT	Number of data bits excluding parity Bit if present 参照：ITU-T勧告Q.931 Bit 2 1 00 - 未使用 01 - 5 bits 10 - 7 bits 11 - 8 bits
parity	9019	3 BIT	Parity information 参照：ITU-T勧告Q.931 Bit 3 2 1 000 - Odd 010 - Even 011 - None 100 - Forced to 0 101 - Forced to 1 上記以外は予約済み
duplexmode	901A	BOOLEAN	Mode duplex 参照：ITU-T勧告Q.931 0 - 半二重 1 - 全二重
modem	901B	6 BIT	Modem Type 参照：ITU-T勧告Q.931 Bits 6 5 4 3 2 1 00000～000101 National Use 010001 - Recommendation V.21 010010 - Recommendation V.22 010011 - Recommendation V.22 bis 010100 - Recommendation V.23 010101 - Recommendation V.26 011001 - Recommendation V.26 bis 010111 -Recommendation V.26 ter 011000 - RecommendationV.27 011001 - Recommendation V.27 bis 011010 - Recommendation V.27 ter 011011 - Recommendation V.29 011101 - Recommendation V.32 011110 - Recommendation V.34 100000～101111 National Use 110000～111111 User Specified



PropertyID	PropertyTag	Type	Value
layer2prot	901C	5 BIT	User information layer 2 protocol 参照：ITU-T勧告Q.931 Bit 5 4 3 2 1 00010 - Recommendation Q.921/I.441 [3] 00110 - Recommendation X.25 [5], link layer 01100 - LAN logical link control (ISO/IEC 8802-2) 上記以外は予約済み
layer3prot	901D	5 BIT	User information layer 3 protocol 参照：ITU-T勧告Q.931 Bit 5 4 3 2 1 00010 - Recommendation Q.931/I.451 00110 - Recommendation X.25, packet layer 01011 - ISO/IEC TR 9577 (Protocol identification in the network layer) 上記以外は予約済み
addlayer3prot	901E	OCTET	Additional User Information layer 3 protocol 参照：ITU-T勧告Q.931 Bits 4321 4321 1100 1100 - Internet Protocol (IETF RFC 791) (ISO/IEC TR 9577) 1100 1111 - Point-to-point Protocol (IETF RFC 1661)
DialledN	901F	30 OCTETS	Dialled Number
DiallingN	9020	30 OCTETS	Dialling Number
ECHOCI	9021		使用しない。JT-H248.1 付属資料E.13のPossible Echo Control properties の例を参照
NCI	9022	1 OCTET	Nature of Connection Indicators 参照：ITU-T勧告Q.763 Bits 8 7 6 5 4 3 2 1 Bits 2 1 Satellite Indicator 0 0 no satellite circuit in the connection 0 1 one satellite circuit in the connection 1 0 two satellite circuits in the connection 1 1 spare Bits 4 3 Continuity check indicator 0 0 continuity check not required 0 1 continuity check required on this circuit 1 0 continuity check performed on a previous circuit 1 1 spare Bits 5 Echo control device indicator 0 outgoing echo control device not included 1 outgoing echo control device included Bits 8 7 6 Spare

PropertyID	PropertyTag	Type	Value
USI	9023	OCTET STRING	User Service Information 参照：ITU-T勧告Q.763 Section 3.57

#### C.10 AAL5プロパティ

PropertyID	PropertyTag	Type	Value
FMSDU	A001	32 bit integer	Forward Maximum CPCS-SDU Size: 参照：ITU-T勧告Q.2931 発側ユーザから着側ユーザへ送信される最大CPCS-SDU サイズ
BMSDU	A002	32 bit integer	Backwards Maximum CPCS-SDU Size 参照：ITU-T勧告Q.2931 着側ユーザから発側ユーザへ送信される最大CPCS-SDU サイズ
SSCS	C.7参照	C.7参照	C.7参照

#### C.11 SDP Equivalentents

SDPEquivalentはプロトコルのtext encoding（符号化）に関して、7.1.8節の中で要点が述べられているSDPexceptionに從属している。例えば、CHOOSEワイルドカードはプロトコルの（2進またはテキスト）符号化に関係なくMGCの中からMGの方向へ使うことが許容されている。

PropertyID	PropertyTag	Type	Value
SDP_V	B001	STRING	Protocol Version 参照：IETF RFC2327
SDP_O	B002	STRING	Owner/creator and session ID 参照：IETF RFC2327
SDP_S	B003	STRING	Session name 参照：IETF RFC2327
SDP_I	B004	STRING	Session identifier 参照：IETF RFC2327
SDP_U	B005	STRING	URI of descriptor 参照：IETF RFC2327
SDC_E	B006	STRING	email address 参照：IETF RFC2327
SDP_P	B007	STRING	phone number 参照：IETF RFC2327
SDP_C	B008	STRING	Connection information 参照：IETF RFC2327
SDP_B	B009	STRING	Bandwidth Information 参照：IETF RFC2327
SDP_Z	B00A	STRING	time zone adjustment 参照：IETF RFC2327
SDP_K	B00B	STRING	Encryption Key

PropertyID	PropertyTag	Type	Value
			参照: IETF RFC2327
SDP_A	B00C	STRING	Zero or more session attributes 参照: IETF RFC2327
SDP_T	B00D	STRING	Active Session Time 参照: IETF RFC2327
SDP_R	B00E	STRING	Zero or more repeat times 参照: IETF RFC2327
SDP_M	B00F	STRING	Media type, port, transport and format 参照: IETF RFC2327

#### C.12 H.245

PropertyID	PropertyTag	Type	Value
OLC	C001	octet string	ITU-T勧告H.245 OpenLogicalChannel structureの値 参照: ITU-T勧告H.245
OLCack	C002	octet string	ITU-T勧告H.245 OpenLogicalChannelAck structureの値 参照: ITU-T勧告H.245
OLCcnf	C003	octet string	ITU-T勧告H.245 OpenLogicalChannelConfirm structureの値 参照: ITU-T勧告H.245
OLCrej	C004	octet string	ITU-T勧告H.245 OpenLogicalChannelReject structureの値 参照: ITU-T勧告H.245
CLC	C005	octet string	ITU-T勧告H.245 CloseLogicalChannel structureの値 参照: ITU-T勧告H.245
CLCack	C006	octet string	ITU-T勧告H.245 CloseLogicalChannelAck structureの値 参照: ITU-T勧告H.245
LCN	C007	Integer	ITU-T勧告H.245 Local Channel Number0-65535の値 参照: ITU-T勧告H.245

## 付属資料D IPによるトランスポート (TRANSPORT OVER IP) (本付属資料は本標準の必須部分である)

### D.1 アプリケーションレベルフレーミング(ALF)によるIP/UDPでのトランスポート

本文書で定義されるプロトコルメッセージは、UDPで転送してもよい。相手側からポートが指定されない場合(節7.2.8参照)、コマンドは、テキスト符号化の場合には2944、バイナリ符号化の場合には2945のデフォルトポート番号に送信されるべきである。応答は、対応するコマンドが送信されてきたアドレスとポートに対して送信されなくてはならない。

ALFは、スタックとは対照的に、アプリケーションが対向側へメッセージを送信する方法に影響を与えることを可能にする技術である。典型的なALF技術は、あるキューがあった時、送信されたメッセージがキューイングされた後に、そのメッセージの順序をアプリケーションが変更することができるというものである。ALFの正式な仕様は存在しない。本付属資料D.1における手順は、ALFの振る舞いを最小限に提案している。

ALFをIP/UDP上に実装する場合、最大メッセージ長におけるMTU(最大転送単位)の制約に注意を払うべきである。

#### D.1.1 At-Most-Once機能の提供

UDP上を転送されるメッセージは廃棄されることがある。このため、時間内に応答がない場合、コマンドは再送されるが、ほとんどのコマンドは、再実行可能 (idempotent)ではない。もし、例えばAddコマンドが複数回実行された場合、MGの状態は予測できなくなる。このため、転送手順は、「最大1回 (At-Most-Once)」機能を提供しなくてはならない。

対向のプロトコルエンティティは、最近のTransactionで送信した応答のリストと、現在未処理のTransactionのリストをメモリに保存しておくことが期待される。各受信メッセージのTransaction 識別子は、同じMidに対して最近送信された応答のTransaction識別子と比較される。合致するものが見つかった場合、エンティティはTransaction処理を実行せず、単に応答を再送する。合致するものが見つからなかった場合、メッセージは現在未処理のTransactionのリストと比較される。リスト中に合致するものが見つかった場合、これはTransactionの重複を意味し、エンティティはTransaction処理を実行しない(節D.1.4のTransactionPending送信処理を参照)。

この手順は、長いタイマ値を使用する。以降この値をLONG-TIMERと記す。このタイマは、再処理の最大回数、再処理タイマの最大値、およびネットワーク内のパケット最大伝送遅延を考慮した最大処理継続時間よりも大きい値に設定されるべきである。推奨値は30秒である。

応答が発行されてからLONG-TIMER秒経過するか、もしくは "Response Acknowledgement parameter"により応答が受信されたことの確認をエンティティが受信することのいずれかにより、応答のコピーは廃棄されてもよい。このパラメータにより確認されたTransactionについては、エンティティは応答が発行されてからLONG-TIMER秒経過するまでTransaction-idのコピーを保持しなくてはならない。これは、ネットワークにより起き得るTransaction Requestの多重コピーを検出し、無視するためである。

#### D.1.2 Transaction識別子と3-wayハンドシェーク

##### D.1.2.1 Transaction識別子

Transaction識別子は、32ビットの整数値である。MGCは、その管理するMGの各々に対して特定の番号ス

ペースを使用してもよいし、ある任意のグループに属する全てのMG群に対して同じ番号スペースを使用してもよい。複数のMGCが、一つの大きなMGを管理する負荷を複数の独立したプロセスで共有してもよい。これらプロセスは同じTransaction番号スペースを共有する。この共有を実装するには、複数の手法がある。例えば、Transaction識別子を一括して割り当てるセンタを用意する、もしくは個々のプロセスに予め識別子の範囲を重複しないよう割り当てておく、等である。実装上、ある論理MGCから生成される全てのTransaction (同一のmIdを持つ)に対し、ユニークな値のTransaction識別子が割り当てられることが保証されなくてはならない。MG群は、単純にTransaction識別子とmIdとのみを参照することで、重複したTransactionを検出できる。

#### D.1.2.2 3-wayハンドシェーク

Transaction Response Acknowledgementパラメータは、どのメッセージにも含むことができる。このパラメータは、"Confirmed Transaction-Id ranges"群を含む。エンティティは、受信したTransaction Responseメッセージ内の"Confirmed Transaction-Id ranges"に含まれる識別子を持つTransactionの応答のコピーを削除する事を選択できる。エンティティは、その後受信するコマンドのTransaction識別子がこの範囲内にある場合、そのコマンドをそのまま廃棄すべきである。

MGが最後の応答をMGCに発行してからLONG-TIMER秒以上が経過した場合、もしくはMGがオペレーションを再開した場合、"Confirmed Transaction-Id ranges"値を使用してはならない。この場合、Transaction-Idに対して何らチェックを行うことなく、Transactionは受理され、処理されるべきである。

"Transaction Response Acknowledgement"パラメータを含むメッセージはどのような順番でも送信され得る。エンティティは、受信した"Confirmed Transaction-Id ranges"をLONG-TIMER秒間保持しておかなくてはならない。

バイナリ符号化では、Response AcknowledgementにfirstAckしか存在しない場合(付属資料A.2参照)、一つのTransactionのみが確認される。firstAckとlastAckの両方が存在する場合、firstAckからlastAckまでの範囲のTransactionが確認される。テキスト符号化では、確認されるTransactionの範囲を表記するためにダッシュ("-")を使用する(付属資料B.2参照)。

#### D.1.3 再送タイマの計算

要求元のエンティティの責任において、全ての未処理のTransactionに対して適切なタイムアウトを適用し、タイムアウトが起きた場合にTransactionを再送する。さらに、再送したTransactionの確認に失敗した場合、要求元のエンティティの責任において、冗長なサービスを探し、かつ/または既存のもしくは留保されているコネクションを解放する。

本仕様では、意図的に各再送タイマの値を規定することを避けている。これらの値は、通常ネットワークに依存する。通常、コマンドの送信から応答の受信までにかかる時間を測定することにより、再送タイマのタイマ値が見積もられるべきである。実装上、再送タイミングを計算するアルゴリズムでは、個々の再送もしくは最初の送信後の再送毎にタイムアウト値を指数関数的に増大させる処理を行うことが保証されなくてはならない。

注記：一つの可能性として、TCP/IPで実装されるアルゴリズムを使用できる。これは、二つの変数を使用する

- 平均確認遅延(average acknowledgement delay; AAD)は、観測された遅延を指数関数的に平滑化し

た平均値から見積もったものである。

- 平均偏差 (average deviation; ADEV)は、観測された遅延と現在の平均値との差分の絶対値を指数関数的に平滑化した平均値から見積もったものである。TCP の再送タイマは、平均遅延(AAD)に平均偏差(ADEV)の N 倍を足した合計値に設定されている。しかし、本標準で規定するプロトコルのためには、タイマの最大値は制限されるべきである。これは、LONG-TIMER 秒以降にゲートウェイが再送パケットを受信しないことを保証するためである。推奨する最大値は、4 秒である。

再送後、エンティティは以下の処理を行うべきである。

- 平均遅延(AAD)の見積もり値を 2 倍にする。
- 0.5 AAD から ADD の間で均一に分布する中から、ランダムな値を一つ計算する。
- 再送タイマを、このランダム値と平均偏差(ADEV)の N 倍の合計値に設定する。

この手順は、二つの効果を持つ。指数関数的に増加する要素を含むため、輻輳時にメッセージの流れを自動的に減速することになる。また、ランダムな要素を含むため、同じ外部的要因により誘発される通知の潜在的な同期を取り除くことになる。

#### D.1.4 暫定応答

いくつかのTransactionを実行するには、長い時間が必要である。長い実行時間は、タイマベースの再送手順と干渉を起こすことがある。これは、極端に多くの再送を許容するか、もしくは効果的というにはあまりに長いタイマ値を許容するか、の判断に帰着する。エンティティは、ある処理が長い実行時間を要すると予測可能な場合、暫定的な応答"Transaction Pending"を送信することができる。このエンティティは、まだ実行中のTransactionに対する再度の実行要求を受けた場合、この応答を送信すべきである。

Transaction Pendingを受信したエンティティは、Requestを再送するための再送タイマを別のものに切り替えなくてはならない。暫定応答に続く最終的な応答を受けて、すぐに確認が送信され、その後通常の再送タイマ値に戻されなくてはならない。暫定的な応答を送信したエンティティは、それに続く最終応答にimmAckRequiredフィールドを含めて、すぐに確認が届くと期待していることを通知しなくてはならない。Reply受信後にTransaction Pendingを受信した場合には、これを無視しなくてはならない。

#### D.1.5 要求、応答、確認の繰り返し

本プロトコルは、Transaction群から構成され、各TransactionはRequestと、通常確認(Acknowledgement)として参照される応答(Response)から構成される。プロトコルメッセージは、UDP上を運ばれるため、失われる可能性がある。時間内に応答が届かない場合、Transactionが繰り返される。エンティティは、それが最近のTransactionに対して送信した応答のリスト、即ちLONG-TIMER秒以前までに送信した全ての応答のリスト、と現在処理中のTransactionのリストをメモリ上に保存しておくことが期待される。

再送メカニズムは、3種類の起こり得るエラーからの保護として使用される。

- 伝送誤り(Transmission Error); 例えば回線上のノイズやキュー内の輻輳によりパケットが廃棄される場合。
- コンポーネント障害(Component Failure); 例えばエンティティへのインタフェースが使用不能となる場合。

- ・ エンティティ障害(Entity Failure); 例えばエンティティ全体が使用不能となる場合。

エンティティは、過去の履歴から、伝送誤りによるパケット廃棄率を見積もることができるべきである。適切に設定されたシステムでは、この廃棄率は非常に低く、一般的には1%以内に抑えることが可能なはずである。MGCもしくはMGがメッセージ再送を2、3回以上繰り返す必要がある場合、伝送誤り以外の何かが起こっていると仮定する方が理にかなっている。例えば、廃棄率を1%と仮定した場合、連続して5回の伝送損失が起こる確率は1千億分の1に過ぎない。これは、1秒間に1000のTransactionを行うMGCが10日間に一度以下のレベルでしか遭遇しない事象である(実際、再送の数が非常に多いと考えられる場合、それはパケット廃棄率の影響による作用である)。"Max1"と名付ける"suspicion threshold"は、通常"disconnection threshold"よりも低いということに注意すべきである。この"disconnection threshold"は、より大きな値に設定されるべきである。

典型的な再送アルゴリズムでは、単純に継続する再送回数をカウントし、(典型的には7から11回の)再送回数の上限を上回った場合に接続が切れたと判断する。未検出の、もしくは進行中の"failover"の可能性を考慮するため、私達は典型的なアルゴリズムを修正し、MGがfailoverを通知する有効なServiceChangeメッセージを受信した場合、このMGは未処理のコマンドを新しいMGCに送信し始めるものとする。それでも、コマンドに対する応答は、コマンドの発信元アドレスに対して送信される。

ネットワークの負荷に自動的に適応するため、本文書では指数関数的に増大するタイマを規定する。仮に最初のタイマが200ミリ秒に設定されているとすると、5回目の再送メッセージの廃棄が検出されるのは約6秒後になる。これは、failoverを検出するために待つべき遅延時間としては、多分受け入れられるものである。その遅延時間後も再送は継続されるべきである。これは、多分に一時的な接続性の問題を乗り越えるためだけでなく、failoverを行うためにもう少し時間的猶予を与えるためである。合計して30秒の遅延時間は、多分受け入れられるものであろう。

しかし、再送の最大遅延時間の制約を設けることは重要である。あらゆる再送に先立ち、最初のデータグラムを送信からの経過時間がT-MAXより長くなっていないことが確認される。T-MAX以上の時間が経過している場合、MGはMGCに障害が起きたと判断し、節11.5に記述されている障害復旧プロセスを開始する。もしMGが現在のMGCと接続するためにリトライするならば、MGはServiceChangeMethodにDisconnectedを設定したServiceChangeメッセージを使用することにより、新しいMGCにMGが一つ以上のTransactionを損失したことを認識させなくてはならない。T-MAX値はLONG-TIMER値と関連付けられる。LONG-TIMER値は、T-MAXにネットワークの最大伝搬遅延を足した値として得られる。

## D.2 TCPの使用

本標準で規定するプロトコルメッセージは、TCP上を転送され得る。相手側からポート番号を特に指定されない場合(節7.2.8参照)、コマンドはデフォルトポート上に送信されるべきである。ここで規定するプロトコルでは、送信単位としてメッセージを用意しているが、一方TCPはストリーム指向のプロトコルである。TCPストリーム内にメッセージを構成するには、RFC1006に記述されるTPKTを使用しなくてはならない。

Transaction指向のプロトコルでは、Transaction Requestもしくは応答(response)が廃棄される可能性は残る。このため、TCPトランスポートを使用するエンティティも、UDP上のアプリケーションレベルフレーミング(ALF)に対してタイマを規定した場合と同様に、各々のRequestとResponseに対してアプリケーション・レベルのタイマを実装することが推奨される。

### D.2.1 At-Most-Once機能の提供

メッセージは、TCP上を運ばれるため、伝送紛失の対象とはならない。しかし、Transaction RequestもしくはそのReplyの紛失は、いずれにせよ実際の実装の対象となる。時間内に応答がない場合、コマンドは再送される。ほとんどのコマンドは、再実行可能 (idempotent)ではない。もし、例えばAddコマンドが複数回実行された場合、MGの状態が予測不可能となり得る。

この様な紛失から保護するため、エンティティは節D.1.1に記述する手順に従うことが推奨される。

### D.2.2 Transaction識別子と3-wayハンドシェーク

同様の理由により、TCPの様な信頼できる配送プロトコルを用いてさえも、Transaction Replyの紛失が起こり得る。エンティティは節D.1.2.2に記述する手順に従うことが推奨される。

### D.2.3 再送タイマの計算

信頼できる配送プロトコルを使用することにより、Transaction Request もしくはReplyの紛失率は非常に低いことが期待される。このため、単純なタイマメカニズムのみが要求される。指数関数的に待ち時間を増大させるアルゴリズムは必要ない。しかし、MGCはTCPと同様にALF/UDPも実装する事が必須のため、MGCのソースコードには指数関数的に増大するタイマ処理がいずれにせよ必要である。よって、このアルゴリズムが適用される可能性はある。

### D.2.4 暫定応答

UDPの場合と同様に、いくつかのTransactionはその実行に長い時間が必要となり得る。エンティティは、あるTransactionに長い実行時間が必要となるとの予測が可能な場合、暫定的な応答 "Transaction Pending"を送信できる。このエンティティは、その処理中のTransactionに対するTransaction Requestの再送を受信した場合、この暫定応答を送信すべきである。

Transaction Pendingを受信したエンティティは、このTransactionの再送タイマをより長いものに変更しなくてはならない。

エンティティは、TransactionとReplyを、それらが確認されるまで保持しておかなくてはならない。節D.1.4の基本手順に従うが、タイマ値は単純なもので充分である。最終応答の受信により、ただちに確認を送信する必要はない。

### D.2.5 コマンド順序

TCPは、順序保存してTransactionを配送する。これには特別な手順は必要ない。ALF/UDPの場合、送信エンティティは、輻輳に際してその取り扱いを修正できる。特に、輻輳に遭遇した場合、Transactionの順序を変えることが可能である。しかしTCPの場合、同様の効果は得られない。

## 付属資料E BASICパッケージ (BASIC PACKAGES)

(本付属資料は本標準の必須部分である)

本付属資料は、JT-H248.1のためのパッケージ定義である。



## E.1 Generic

Package Name: Generic

PackageID: g (0x0001)

Description: 一般的に発生する事象のための総合パッケージ

Version: 2

Extends: なし

### E.1.1 Properties

定義なし

### E.1.2 Events

#### E.1.2.1 Cause

Event Name: Cause

EventID: cause (0x0001)

Description: 一般的なエラーイベント

EventsDescriptor parameters: なし

ObservedEvents Descriptor Parameters:

General Cause

Parameter Name: General Cause

ParameterID: Generalcause (0x0001)

Description: このパラメータでは失敗を6つにグループ化しており、MGCはそれに従い動作する。

Type: 列挙型

Optional: No

Possible Values:

"NR" Normal Release (0x0001) ⇒正常解放

"UR" Unavailable Resources (0x0002) ⇒リソース使用不可

"FT" Failure, Temporary (0x0003) ⇒失敗、一時的

"FP" Failure, Permanent (0x0004) ⇒失敗、永続的

"IW" Interworking Error (0x0005) ⇒協調動作エラー

"UN" Unsupported (0x0006) ⇒不明

Default: なし

Failure Cause

Parameter Name: Failure Cause

ParameterID: Failurecause (0x0002)

Description: "Failure Cause"は、解放された装置によって生成される値である(つまり、解放されたネットワーク接続)。関係する値は適切なベアラ制御プロトコルで定義される。

Type: OCTET STRING

Optional: Yes

Possible Values: OCTET STRING

### E.1.2.2 Signal Completion

Event Name: Signal Completion

EventID: sc (0x0002)

Description: notifyCompletionパラメータが完了イベントの報告可能と設定されたシグナルの停止を示す。さらに引き続き処理の手順に関しては、節7.1.1、7.1.11、および7.2.7を参照のこと。

EventsDescriptor parameters: なし

ObservedEvents Descriptor Parameters:

Signal Identity

Parameter Name: Signal Identity

ParameterID: SigID (0x0001)

Description: このパラメータは停止されたシグナルを識別する。シグナルリストに含まれたシグナルに関しては、適切なリストを示しているSignal list identity parameterも返却されるべきである。

Type: Binary: octet (string), Text: string

Optional: No

Possible Values:

停止されたシグナル。シグナルはワイルドカード化せず、"pkgdName"を用いて識別されなければならない。

Default: なし

Termination Method

Parameter Name: Termination Method

ParameterID: Meth (0x0002)

Description: シグナルが停止された理由を示す。

Type: 列挙型

Optional: No

Possible Values:

"TO" (0x0001) Signal timed out or otherwise completed on its own ⇒シグナルタイムアウトもしくはシグナル自身の完了

"EV" (0x0002) Interrupted by event ⇒イベントによる割り込み

"SD" (0x0003) Halted by new Signals Descriptor ⇒新規のSignalsディスクリプタによる中止

"NC" (0x0004) Not completed, other cause ⇒上記以外の原因による中断

"PI"(0x0005) First to penultimate iteration. For last iteration, use TO. ⇒最後から2番目の反復への最初。最後の反復については、"TO"を使用。(★)

Default: なし

Signal List ID

Parameter Name: Signal List ID

ParameterID: SLID (0x0003)

Description: シグナルが、どのシグナルリストに属しているかを示す。SignalList IDはシグナルがシグナル

リストに属している場合にのみ返却される。

Type: integer

Optional: Yes (シグナルリストが使用される場合のみサポート)

Possible Values: 1-65535

Default: なし

#### Request ID

Parameter Name: Request ID

ParameterID: RID (0x0004)

Description: SignalIDが、どのNotifyCompletion要求に関連させたかを示す。

Type: integer

Optional: Yes (シグナルリストが使用される場合のみサポート)

Possible Values: 1-4294967295

### E.1.3 Signals

定義なし

### E.1.4 Statistics

定義なし

## E.2 Base Root Package

Package Name: Base Root Package

PackageID: root (0x0002)

Description: このパッケージはゲートウェイの幅広い属性を定義する。

Version: 2

Extends: なし

### E.2.1 Properties

#### E.2.1.1 Maximum Number of Contexts

Property Name: MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

Description: このプロパティの値はいつでも存在できるContextの最大数を与える。NULL Contextはこの数に含まれていない。

Type: Double

Possible Values: 1以上

Default: Provisioned

Defined in: TerminationState

Characteristics: ReadOnly

#### E.2.1.2 Maximum Terminations Per Context

Property Name: MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

Description: Contextにおいて許容される終端の最大数については節6.1を参照。

Type: Integer

Possible Values: 任意のInteger

Defined In: TerminationState

Characteristics: ReadOnly

### E.2.1.3 Normal MG Execution Time

Property Name: normalMGExecutionTime

PropertyID: normalMGExecutionTime (0x0003)

Description: MGCが期待するMGからの任意のTransactionに対する応答間隔(ネットワーク遅延は除く)を示すため、MGCによって設定可能である。

Type: Integer

Possible Values: ミリ秒を表す任意のInteger

Defined in: TerminationState

Characteristics: Read/Write

### E.2.1.4 Normal MGC Execution Time

Property Name: normalMGCExecutionTime

PropertyID: normalMGCExecutionTime (0x0004)

Description: MGが期待するMGCからの任意のTransactionに対する応答間隔(ネットワーク遅延は除く)を示すために、MGCによって設定可能である。

Type: Integer

Possible Values: ミリ秒を表す任意のInteger

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

### E.2.1.5 MG Provisional Response Timer Value

Property Name: MGProvisionalResponseTimerValue

PropertyID: MGProvisionalResponseTimerValue (0x0005)

Description: Transactionが完了されなかった場合のMGCでのMGからのPending応答待ち時間を示す。初期値として、normalMGExecutionTimeにネットワーク遅延を足した値、またはそれ以下の値でもよい。

Type: Integer

Possible Values: ミリ秒を表す任意のInteger

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

#### E.2.1.6 MGC Provisional Response Timer Value

Property Name: MGCProvisionalResponseTimerValue

PropertyID: MGCProvisionalResponseTimerValue (0x0006)

Description: Transactionが完了されなかった場合のMGでのMGCからのPending応答待ち時間を示す。初期値として、normalMGCExecutionTimeにネットワーク遅延を足した値、またはそれ以下の値でもよい。

Type: Integer

Possible Values: ミリ秒を表す任意のInteger

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

#### E.2.1.7 MGC Originated Pending Limit

Property Name: MGCOriGinatedPendingLimit

PropertyID: MGCOriGinatedPendingLimit (0x0007)

Description: MGCから受信可能なTransactionPendingの数を示す。一旦この値を超えたら、MGCはエラーコード506 “Number of TransactionPendings Exceeded”(TransactionPending数超過)でTransactionReplyを発行すべきである。そうでなければMGはそのTransactionがエラーになったものとみなせる。

Type: Integer

Possible Values: 任意の可能なinteger

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

#### E.2.1.8 MG Originated Pending Limit

Property Name: MGOriGinatedPendingLimit

PropertyId: MGOriGinatedPendingLimit (0x0008)

Description: MGから受信可能なTransactionPendingの数を示す。一旦この値を超えたら、MGはエラーコード506 “Number of TransactionPendings Exceeded”(TransactionPending数超過)でTransactionReplyを発行すべきである。そうでなければMGCはそのTransactionがエラーになったものとみなせる。

Type: Integer

Possible Values: 任意の可能なinteger

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

### E.2.2 Events

定義なし

### E.2.3 Signals

定義なし

### E.2.4 Statistics

定義なし

### E.2.5 Procedures

定義なし

## E.3 Tone Generator パッケージ

Package Name: Tone Generator Package

PackageID: tonegen (0x0003)

Description:

本パッケージは、オーディオトーンを生成するシグナルを定義する。本パッケージでは拡張可能であることを意図としてパラメータ値を規定しない。トーンは一般的に、"interdigit"(桁間)時間の遅延を表すパラメータ"ind"、"playtones"と共に使用される"tone id"を含む独立したシグナルとして、定義される。"tone id"はあらゆるトーン生成においても同一トーンがきちんと一致しているよう維持されなければならない。MGは設置される国にあったトーンの特徴を持つように供給されることが望まれる。

Version: 2

Designed to be extended only: Yes

Extends: なし

### E.3.1 Properties

定義なし

### E.3.2 Events

定義なし

### E.3.3 Signals

#### E.3.3.1 Play Tone

Signal Name: Play Tone

SignalID: pt (0x0001)

Description: オーディオチャネルでのオーディオトーン再生。

Signal Type: Brief

Duration: Provisioned

Additional Parameters

Tone ID List

Parameter Name: Tone ID List

ParameterID: tl (0x0001)

Description: 連続して再生されるトーンのリスト。

Type: Sublist of Enumeration

Optional: No

Possible Values: サブリストは、1つもしくはそれ以上のtone IDを含まなくてはならない。

Default: なし

Inter signal duration

Parameter Name: Inter-signal duration

ParameterID: ind (0x0002)

Description: 連続する2つのトーン間についてのタイムアウト時間。ミリ秒単位

Type: Integer

Optional: Yes

Possible Values: 0以上の任意の整数

Default: Provisioned

Tone Direction

Parameter Name: Tone Direction

ParameterID: btd (0x0003)

Description: 停止から進行するべきトーンの方角。

Type: Enumeration

Optional: Yes

Possible values:

"EXT" (0x0001) External ⇒外部方向

"INT" (0x0002) Internal ⇒内部方向

"BOTH" (0x0003) Both ⇒双方向

Default: External

#### E.3.4 Statistics

定義なし

#### E.3.5 Procedures

トーンIDは本パッケージで指定されない。本パッケージを拡張するパッケージは、加える個々のトーン信号と同様にトーンIDに対する可能な値を加えることができる。

### E.4 Tone Detection パッケージ

Package Name: Tone Detection Package

PackageID: tonedet (0x0004)

Description: 本パッケージでは、オーディオトーン検出時のイベントを定義する。トーンは名前(tone ID)に

よって決定される。MGは設置される国にあったトーンの特徴を持つように供給されることが望まれる。本パッケージでは拡張可能である事を意図としてパラメータ値を規定しない。

Version: 1

Designed to be extended only: Yes

Extends: なし

#### E.4.1 Properties

定義なし

#### E.4.2 Events

##### E.4.2.1 Start Tone Detected

Event Name: Start tone detected

EventID: std, 0x0001

Description: トーンの始まりを検知する。実際のトーン検出特性は実装に依存する。

EventsDescriptor parameters:

Tone Id List

ParameterID: tl (0x0001)

Type: Sublist of Enumeration

Optional: No

Possible values: 本パッケージで定義される唯一の"tone ID"は「ワイルドカード」("wild card")であり、テキスト符号化で"\*"、バイナリ符号化で"0x0000"と表現される。本パッケージへの拡張により、"tone ID"に possible valueが追加される。もし"tl"が「ワイルドカード」("wild card")である場合は、いかなる"tone ID"も検知されることとなる。

Default: なし

ObservedEventsDescriptor parameters:

Tone ID

Parameter Name: ParameterID: tid (0x0003)

ParameterID: tid (0x0003)

Description: 検出されたトーンのトーンID

Type: 列挙型

Optional: No

Possible values: 上で定義した通り「ワイルドカード」("wild card")が本パッケージで定義される唯一の値である。本パッケージへの拡張により、possible valueが"tone ID"に追加される。

Default: なし

##### E.4.2.2 End Tone Detected

End tone detected

EventID: etd, 0x0002

Description: トーンの終わりを検知する。



EventDescriptor parameters:

Tone ID List

Parameter Name: Tone ID List

ParameterID: tl (0x0001)

Description:

Type: enumerationまたはlist of enumerated types

Possible values: 本パッケージでは規定されない。本パッケージへの拡張により、possible valueが"tone ID"に追加される。

Default: なし

ObservedEventsDescriptor parameters:

Tone ID

Parameter Name: Tone ID

ParameterID: tid (0x0003)

Description: 検出されたトーンのトーンID

Type: 列挙型

Optional :No

Possible values: 上で定義した通り「ワイルドカード」("wild card")が本パッケージで定義される唯一の値である。本パッケージへの拡張により、possible valueが"tone ID"に追加される。

Default: なし

Duration

ParameterId: Parameter Name: Duration

ParameterId: dur (0x0002)

Description: 本パラメータは、最初のトーンの検出から停止までの継続時間を含む。

Type: Integer

Optional: Yes

Possible values: ミリ秒単位で任意の正の整数

Default: なし

#### E.4.2.3 Long Tone Detected

Event Name: Long tone detected

EventID: ltd(0x0003)

Description: ある一定時間以上トーンが連続再生された場合に検知する。

EventDescriptor parameters:

Tone ID List

Parameter Name: Tone ID List

ParameterID: tl (0x0001)

Description: 検出される"tone ID"のリスト

Type: Sublist of Enumeration

Optional: No

Possible values: 上で定義した通り「ワイルドカード」("wild card")が本パッケージで定義される唯一の値である。本パッケージへの拡張により、possible valueが"tone ID"に追加される。

Default: なし

Duration

Parameter Name: Duration

ParameterID: dur (0x0002)

Description: テストに対する所要時間

Type: Integer

Optional: Yes

Possible values: 任意の正の整数、ミリ秒で表現される

Default: Provisioned

ObservedEventsDescriptor parameters:

Tone ID

Parameter Name: Tone ID

ParameterID: tid (0x0003)

Description: 検出される"tone ID"のリスト

Type: 列挙型

Optional: No

Possible values: 本パッケージでは規定されない。本パッケージへの拡張により、possible valueが"tone ID"に追加される。

#### E.4.3 Signals

定義なし

#### E.4.4 Statistics

定義なし

#### E.4.5 Procedures

定義なし

### E.5 Basic DTMF Generator パッケージ

**Package Name:** Basic DTMF Generator Package

PackageID: dg (0x0005)

Description: 本パッケージは、シグナルとしての基本DTMFトーンを定義している。また、tonegen内のplaytoneのtlパラメータの許容値を拡張している。

Version: 2

Extends: tonegen version 2

#### E.5.1 Properties

定義なし

## E.5.2 Events

定義なし

## E.5.3 Signals

### E.5.3.1 DTMF characterCharacter 0

Signal Name: DTMF Character 0

SignalID: d0 (0x0010)

Description: DTMF 0 トーンを生成する。DTMF 0 の実際の特性はゲートウェイで定義される。

Signal Type: Brief

Duration: Provisioned

Additional Parameters:

Tone Direction

Parameter Name: Tone Direction

ParameterID: btd (0x0001)

Description: 停止から進行するべきトーンの方角。

Type: Enumeration

Optional: Yes

Possible values:

"EXT" (0x0001) External ⇒外部方向

"INT" (0x0002) Internal ⇒内部方向

"BOTH" (0x0003) Both ⇒双方向

Default: 外部方向

Additional Values: d0(0x0010)がplaytoneのための"tone ID"として定義されている。

他のDTMF特性は全く同じ方法で定義される。全てのシグナル名とシグナルIDを含む表を示す。各dtmf特性がシグナルとして、かつ"tone ID"としても定義されており、基本のtone生成パッケージを拡張していることに注意が必要。また、DTMF SignalID群はDigitMap中で使用されている名前とは違うことにも注意が必要。

Signal Name	Signal ID/tone ID
DTMF character 0	d0 (0x0010)
DTMF character 1	d1 (0x0011)
DTMF character 2	d2 (0x0012)
DTMF character 3	d3 (0x0013)
DTMF character 4	d4 (0x0014)
DTMF character 5	d5 (0x0015)
DTMF character 6	d6 (0x0016)

DTMF character 7	d7 (0x0017)
DTMF character 8	d8 (0x0018)
DTMF character 9	d9 (0x0019)
DTMF character *	ds (0x0020)
DTMF character #	do (0x0021)
DTMF character A	da (0x001a)
DTMF character B	db (0x001b)
DTMF character C	dc (0x001c)
DTMF character D	dd (0x001d)

#### E.5.4 Statistics

定義なし

#### E.5.5 Procedures

定義なし

### E.6 DTMF detection パッケージ

Package Name: DTMF Detection Package

PackageID: dd (0x0006)

Description: 本パッケージは基本DTMFトーン検出を定義している。本パッケージは、”start tone detected”(最初のトーン検出)、"end tone detected"(最後のトーン検出)、"long tone detected"(長いトーン検出)イベントの中のトーンIDのpossible valueを拡張している。

追加のトーンID値はパッケージdg(basic DTMF generator パッケージ: 基本DTMF生成パッケージ)内に記述された全てのトーンである。

次の表はDTMFイベントと、節7.1.14で示されたdigit mapのシンボルの対応付けを行っている。

DTMF	Event Symbol
d0	"0"
d1	"1"
d2	"2"
d3	"3"
d4	"4"
d5	"5"
d6	"6"
d7	"7"
d8	"8"
d9	"9"
da	"A"または"a"

db	"B"または"b"
dc	"C"または"c"
dd	"D"または"d"
ds	"E"または"e"
do	"F"または"f"

Version: 1

Extends: tonedet version 1

### E.6.1 Properties

定義なし

### E.6.2 Events

#### E.6.2.1 DTMF Digits

EventIdsEvent Name: DTMF Digits

Event ID: EventIDが、付属資料E.5.3の表で定義されているSignalIDと同じ名前で定義されている。

Description: MGがdigitを検出するときに生成される。

EventsDescriptor parameters: なし

ObservedEvents Descriptor Parameters: なし

#### E.6.2.2 DigitMap Completion Event

EventIdsEvent Name: DigitMap Completion Event

EventID: ce, (0x0004)

Description: 節7.1.14に記述されているようにdigit mapが完了した場合に生成される。

EventsDescriptor parameters : なし

ObservedEventsDescriptor parameters:

DigitString

Parameter Name: DigitString

ParameterID: ds (0x0001)

Description: 現在のダイヤル文字列の部分。これは節7.1.14に示されているように、digit mapで示されたイベントシーケンスの一部もしくは全部に一致したもの。

Type: ひとつのquotedStringとして返されるdigit mapシンボル(空の可能性もある)のstring

Optional: No

Possible Values: "0"~"9"、"A"~"F"の文字列、および長期継続修飾子 "Z"

Termination Method

Parameter Name: Termination Method

ParameterID: Meth (0x0003)

Description: イベント生成の理由を示す。節7.1.14にある処理を参照。

Type: 列挙型

Possible Values:

"UM" (0x0001) あいまいでない一致

"PM" (0x0002) 部分一致、タイマ満了による終了もしくはイベント不一致

"FM" (0x0003) 完全一致、タイマ満了による終了もしくはイベント不一致

### E.6.3 Signals

定義なし

### E.6.4 Statistics

定義なし

### E.6.5 Procedures

Digit map処理は、付属資料E.6.2で定義されているようにdigit map completionイベントを含むEventsディスクリプタが有効になり、かつそのdigit map completionが節7.1.9に定義されている通り、requested actionにeventDMフィールドを含んでいる場合のみ、実行化される。KeepActiveやSignalsディスクリプタの埋め込み型イベントといった他のパラメータも、Eventsディスクリプタ内に存在しており、digit map処理の実行化に影響しない。KeepActiveや、Signalディスクリプタの埋め込み型イベントといった他のパラメータはEventディスクリプタで表されてもよく、Digit map処理の実行に影響しない。

## E.7 Call Progress Tones Generator パッケージ

Package Name: Call Progress Tones Generator Package

PackageID: cg, 0x0007

Description: 本パッケージは、シグナルとして基本的な呼経過トーンを定義しており、tonegen中のplaytoneのtlパラメータの許容値を拡張している。

Version: 2

Extends: tonegen version 2

### E.7.1 Properties

定義なし

### E.7.2 Events

定義なし

### E.7.3 Signals

#### E.7.3.1 Dial Tone

Signal Name: Dial Tone

SignalID: dt (0x0030)

Description: ダイヤルトーンを生成する。実際のダイヤルトーンの特徴はゲートウェイで示される。

Signal Type: Timeout

Duration: Provisioned

Additional Parameters: なし

Additional Values: dt(0x0030)がplaytoneのトーンIDとして定義されている。

本パッケージの他のトーン群は全く同じ方法で定義されている。全てのシグナル名とシグナルIDを表に示す。各トーンはシグナルと同時にトーンIDとしても定義されており、basic tone generationパッケージを拡張していることに注意が必要。

Signal Name	Signal ID/tone ID
Dial Tone	dt (0x0030)
Ringing Tone	rt (0x0031)
Busy Tone	bt (0x0032)
Congestion Tone	ct (0x0033)
Special Information Tone	sit(0x0034)
(Recording) Warning Tone	wt (0x0035)
Payphone Recognition Tone	prt (0x0036)
Call Waiting Tone	cw (0x0037)
Caller Waiting Tone	cr (0x0038)

#### E.7.4 Statistics

定義なし

#### E.7.5 Procedures

注記： 要求されているトーンID集合は勧告E.180/Q.35[ITU-T勧告E.180/Q.35(1998年)]で定義されているものと対応している。これらのトーンの意味定義はE.180を参照のこと。

### E.8 Call Progress Tones Detection パッケージ

Package Name: Call Progress Tones Detection Package

PackageID: cd (0x0008)

Description: このパッケージは、基本的なcall progress detection tonesを定義する。このパッケージは、"start tone detected", "end tone detected", "long tone detected"イベントのトーンIDのpossible valueを拡張する。

#### Additional values

トーンIDの値は、start tone detected, end tone detected, long tone detectedのために定義される。これらはパッケージcg(call progress tone generationパッケージ)と同じ値を持つ。

要求されるtone idsのセットは、勧告E.180/Q.35[ITU-T勧告E.180/Q.35 (1998)]に一致する。これらのトーンの意味の定義についてはITU-T勧告E.180/Q.35を参照のこと。

Version: 1

Extends: tonedet version 1

### E.8.1 Properties

定義なし

### E.8.2 Events

EventsはE.7.3のテーブルに記されたトーンのためのcall progress tones generatorパッケージ(cg)と同じように定義される。

### E.8.3 Signals

定義なし

### E.8.4 Statistics

定義なし

### E.8.5 Procedures

定義なし

## E.9 Analog Line Supervision パッケージ

Package Name: Analog Line Supervision Package

PackageID: al, 0x0009

Description: このパッケージはアナログ回線のためのイベントとシグナルを定義する

Version: 1

Extends: なし

### E.9.1 Properties

定義なし

### E.9.2 Events

#### E.9.2.1 On-hook

Event Name: On-hook

EventID: on (0x0004)

Description: ハンドセットがオンフックになったことを検知する。オンフックイベントを監視することを要求するEvents Descriptorが活性化しており、回線がすでにオンフックであるときはいつでも、MGは”strict”パラメータの設定にしたがって振舞わなければならない。

EventDescriptor parameters

Strict Transition

Parameter Name: Strict Transition

ParameterID: strict (0x0001)

Type: 列挙型

Optional: Yes



Possible values:

"exact"(0x0000) : 実際のフックの状態がオンフックに移行したことが認識されたことを意味する。

"state"(0x0001) : フックの状態の遷移が検知されたら、もしくはフックの状態が既にオンフックであるのなら、イベントが認識されることを示す。

"failWrong"(0x0002) : フックの状態が既にオンフックであるなら、コマンドが失敗し、エラーが報告されることを意味する。

Default: Exact

ObservedEventsDescriptor parameters

Initial State

Parameter Name: Initial State

ParameterID: init (0x0002)

Description: オンフックへの遷移を報告した理由。Strict Transition parameterが"state"へ設定された場合、単に返信される。

Type: Boolean

Possible values:

Possible values: Trueはこのイベントを含むEventsディスクリプタが活性化されたとき、回線が既にオンフックであるので、イベントが報告されたことを意味する。

Falseはeventが実際の状態のオンフックへの遷移を表すことを意味する。

Default: なし

### E.9.2.2 Off-hook

Event Name: Off-hook

EventID: of (0x0005)

ハンドセットがオフフックになったことを検知する。オフフックイベントを監視することを要求するEventsディスクリプタが活性化しており、回線がすでにオフフックであるときはいつでも、MGは"strict"パラメータの設定にしたがって振舞わなければならない。

EventDescriptor parameters

Strict Transition

Parameter Name: Strict Transition

ParameterID: strict (0x0001)

Description: オフフックのイベントがどのように検知されるかを示す。

Type: 列挙型

Possible values:

"exact"(0x0000) : 実際のフックの状態がオフフックに移行したことが認識されたことを意味する。

"state"(0x0001) : フックの状態の遷移が検知されたら、あるいは、フックの状態が既にオフフックであるのなら、イベントが認識されることを示す。

"failWrong"(0x0002) : フックの状態が既にオフフックであるなら、コマンドが失敗

し、エラーが報告されることを意味する。

Default: Exact

ObservedEventsDescriptor parameters

Initial State

Parameter Name: Initial State

ParameterID: init (0x0002)

Description: オンフックへの遷移を報告した理由。Strict Transition parameterが"state"へ設定された場合、単に返信される。

Type: Boolean

Optional: Yes

Possible values: "True"はこのイベントを含むEventsディスクリプタが活性化されたとき、回線が既にオフフックであるので、イベントが報告されたことを意味する。

"False"はeventが実際の状態のオフフックへの遷移を表すことを意味する。

Default: なし

### E.9.2.3 Flashhook

Event Name: Flashhook

EventID: fl, 0x0006

Description: Handset flashを検知する。最小と最大の間隔の間でオフフックの後にオンフックがあった場合、フラッシュが発生する。

EventDescriptor parameters

Minimum duration

Parameter Name: Minimum Duration

ParameterID: mindur (0x0004)

Description: オンフックとフラッシュで検出されるオフフックの間の最小時間

Type: integer

Optional: Yes

Possible values: ミリ秒単位で任意の正の整数

Default: Provisioned

Maximum duration

Parameter Name: Maximum Duration

ParameterID: maxdur (0x0005)

Description: オンフックとフラッシュで検出されるオフフックの間の最大時間

Type: integer

Optional: Yes

Possible values: ミリ秒単位で任意の正の整数

Default: Provisioned

ObservedEventsDescriptor parameters: なし

### E.9.3 Signals

#### E.9.3.1 Ring

Signal Name: Ring

SignalID: ri, 0x0002

Description: 該当回線に対し、リングングが行われる

Signal Type: TimeOut

Duration: Provisioned

Additional Parameters:

Cadence

Parameter Name: Cadence

ParameterID: cad (0x0006)

Description: オンとオフの入れ違いになる間隔を表す。その間隔は、完全なリングング周期を構成する。制限された機能のMGは生成することができないリズムを無視してもよい。

Type: Sublist of Integer

Optional: Yes

Possible values: ミリ秒単位で任意の正の整数

Default: Provisioned

Frequency

Parameter Name: Frequency

ParameterID: freq (0x0007)

Description: リングングの周波数。制限された機能のMGは生成することができない周波数値を無視してもよい。

Type: integer

Optional: Yes

Possible values: Hz単位で任意の正の整数

Default: Provisioned

### E.9.4 Statistics

定義なし

### E.9.5 Error codes

Error Code #: 540

Name: Unexpected initial hook state(予期しない初期フック状態)

Definition: このエラーは、次の事象のときに発生する。"failWrong"に設定されたstrictパラメータを用いて、MGCがフック状態の遷移イベントを要求しようとしたとき。および、そのフック状態が、既にその遷移を意味するものであるとき。

Error Text in the Error Descriptor: -

Comment: -

## E.9.6 Procedures

"failWrong"に設定された"strict"パラメータ(0x0001)をもつhook state transition eventを含むEventsDescriptorをMGCが設定するならば、そして、hook stateが既に遷移を暗示するものならば、コマンドの実行は、EventsDescriptorの失敗を含む。MGはその応答にエラーコード 540"Unexpected initial hook state"(予期しない初期フック状態)を含まなければならない。

## E.10 Basic Continuity パッケージ

Package Name: Basic Continuity Package

PackageID: ct (0x000a)

Description: 本パッケージは導通試験のためのイベントとシグナルを定義する。導通試験はループバック機能なし、トランシーバ機能を含む。

Version: 1

Extends: なし

### E.10.1 Properties

定義なし

### E.10.2 Events

#### E.10.2.1 Continuity Test Completion

Event Name: Completion

EventID: cmp, 0x0005

Description: 本イベントは導通試験の完了を検知する

EventDescriptor parameters: なし

ObservedEventsDescriptor parameters:

Continuity Test Result

Parameter Name: Result

ParameterID: res (0x0008)

Description: 導通試験の結果を示す

Type: 列挙型

Optional: No

Possible values: "SUCCESS" (0x0001) 成功

"FAILURE" (0x0000) 失敗

Default: なし

### E.10.3 Signals

#### E.10.3.1 Continuity Test

Signal Name: Continuity Test

SignalID: ct (0x0003)

Discussion: 適用されるTermination上での導通試験トーンの送信を開始する

Signal Type: TimeOut

Default: provisioned

Additional Parameters: なし

#### E.10.3.2 Respond to Continuity Test

Signal Name: Respond

SignalID: rsp (0x0004)

Description: シグナルは導通試験に応答するために使用される。更なる説明はE.10.5を参照。

Signal Type: On/Off

Default: Provisioned

Additional Parameters: なし

#### E.10.4 Statistics

定義なし

#### E.10.5 Procedures

MGCが導通試験を開始したいとき、MGCは以下を含むコマンドをMGへ送信する。

- ct Signal を持つ Signals ディスクリプタ
- cmp Event を持つ Evnets ディスクリプタ

ct Signalとcmp Eventを含むコマンドを受信すると、MGは指定されたterminationのために導通試験トーンを開始する。Signalがタイムアウトする前に、return toneが検知され、他の要求される状態が満たされるのなら、successに等しいresult paramterの値を持つcmp Eventが生成されなければならない。他のすべてのケースでは、failureに等しいresult paramterの値を持つcmp Eventが生成されなければならない。

MGCがMGに導通試験に対する応答を望むとき、MGCはrsp Signalを持つSignalsディスクリプタを含むコマンドをMGに送信する。MGはループバックを試すか、導通試験トーンの受信を待つ(2-wire circuit)。ループバックの場合は、入力情報が出力情報としてそのまま返さなければならない。2-wire circuitのときは、適切なテストトーンが受信され、適切な 応答トーンが送信されなければならない。いつrsp Signalを削除するかはMGCが決定する。

terminationで導通試験が実行されるなら、そのteminationではエコー装置やコーデックは活性化されない。

導通試験の一部として音声パスのテストを実行することは、ネットワークオペレータ間の双方の合意によって提供される。

(参考注釈) トーンやテスト手順の詳細な例は、ITU-T勧告Q.724セクション7および8、ITU-T勧告Q.764セクション2.1.8およびITU-T勧告Q.1902.4で与えられる。

#### E.11 Network パッケージ

Package Name: Network Package

PackageID: nt (0x000b)

Description: 本パッケージはネットワーク種別に依存しないネットワークterminationのプロパティを定義す

る。これは、TDM、IPおよびATMに制限されないものを含む。

Version: 1

Extends: なし

## E.11.1 Properties

### E.11.1.1 Maximum Jitter Buffer

Property Name: Maximum Jitter Buffer

PropertyID: jit (0x0007)

Description: 本プロパティはジッタバッファに最大サイズを設定する。

Type: integer

Possible Values: 本プロパティはミリ秒で規定される。

Defined In: LocalControlDescriptor

Characteristics: Read/Write

## E.11.2 Events

### E.11.2.1 Network Failure

EventID: netfail, 0x0005

Description: 本terminationは外部ないし内部のネットワークによる失敗検知のイベントを生成する。

EventDescriptor parameters: なし

ObservedEventsDescriptor parameters:

Cause

Parameter Name: Cause

ParameterID: cs (0x0001)

Description: 本パラメータは失敗理由の診断情報を提供するためにfailure eventに含められてもよい

Type: String

Possible values: 任意のテキストstring

Default: なし

### E.11.2.2 Quality Alert

Event Name: Quality Alert

EventID: qualert, 0x0006

Description: 本イベントは、MGがネットワーク接続の品質の低下を指示することを許可する。MGはパケット損失、ジッタ、伝播遅延を測定することでこれを行ってもよい。また、品質低下のパーセンテージを使用してこれを示してもよい。

EventDescriptor parameters:

Threshold

Parameter Name: Threshold

ParameterId: th (0x0001)

Description: 事前に準備された方法に基づき測定、計算される品質低下のパーセント値。例えば、パケット

損失、ジッタ、遅延を考慮してもよい。計算が閾値を超える場合、イベントが発生する。

Type: integer

Optional: Yes

Possible Values: 0 - 99

Default: Provisioned

ObservedEventsDescriptor parameters:

Threshold

Parameter Nmae: Threshold

ParameterId: th (0x0001)

Description: 事前に準備された方法に基づき測定、計算される品質低下のパーセント値。例えば、パケット損失、ジッタ、遅延を考慮してもよい。

Type: integer

Optional: Yes

Possible Values: 0 - 99

Default: Provisined

### E.11.3 Signals

定義なし

### E.11.4 Statistics

#### E.11.4.1 Duration

Statistics Name: Duration

StatisticsID: dur (0x0001)

Description: TerminationがNULL contextに存在していた、もしくはNULL contextから外れた継続時間を提供する。

Type: Double

Possible values: ミリ秒単位で任意の正の整数

Level: Either

#### E.11.4.2 Octets Sent

Statistic Name: Octets Sent

StatisticID: os (0x0002)

Description: terminationがNULL contextに存在していた、もしくはNULL contextから外れてから、terminationもしくはstreamより送信されたオクテット数を提供する。

オクテットは、すべてのトランスポートのオーバーヘッドを除いて、出て行ったメディアフローを表示する。terminationレベルにおいて、それはすべてのストリーム上で出て行ったフローの合計と等しい。アナログ送信メディアについては、オクテット数は0に等しくされるものにしなくてはならない。

Type: double

Possible Values: 任意の64ビットの0以上の整数

Octets Received

StatisticID: or (0x0003)

Type: double

Possible Values: 任意の64ビットの0以上の整数

Level: Either

#### E.11.4.3 Octets Received

Statistic Name: Octets Received

StatisticID: or (0x0003)

Type: Double

Description: terminationがNULL contextに存在していた、もしくはNULL contextから外れてから、terminationもしくはstreamで受信したオクテット数を提供する。

オクテットは、すべてのトランスポートのオーバーヘッドを除いて、入ってきたメディアフローを表示する。terminationレベルにおいて、それはすべてのストリーム上で入ってきたフローの合計と等しい。アナログ送信メディアについては、オクテット数は0に等しくされるものにはなくてはならない。

Possible values: 任意の64ビットの0以上の整数

Level: Either

#### E.11.5 Procedures

定義なし

### E.12 RTP パッケージ

Package Name: RTP Package

PackageID: rtp (0x000c)

Version: 1

Description: 本パッケージは、Real-time Transport Protocol(RTP)[RFC 3550]を用いたパケットベースのマルチメディアデータの転送をサポートするために使用される。

Extends: nt version 1

#### E.12.1 Properties

定義なし

#### E.12.2 Events

##### E.12.2.1 Payload Transition

Event Name: Payload Transition

EventID: pltrans, 0x0001

Description: 本イベントは、あるRTP ペイロードフォーマットから他のフォーマットへの移行があったことを検知し、通知する。

EventDescriptor parameters: なし



ObservedEventsDescriptor parameters:

RTP Payload Type

ParameterName: rtppayload

ParameterID: rtppltype, 0x01

Description: 遷移が作られたペイロードフォーマット

Type: Sublist of Enumeration

Optional: No

Possible values: 符号化手法は、RTP AVプロファイルないし、IANAに定義されているように、一つまたは複数の妥当な符号化名称を使用して指定されなくてはならない。

Default: なし

### E.12.3 Signals

定義なし

### E.12.4 Statistics

#### E.12.4.1 Packets Sent

Statistic Name: Packets Sent

StatisticID: ps (0x0004)

Description: terminationがNULL contextに存在していた、もしくはNULL contextから外れてから、terminationもしくはstreamより送信されたパケット数を提供する。

Type: Double

Possible Values: 任意の64ビットの0以上の整数

Level: Either

#### E.12.4.2 Packets Received

Statistic Name: Packets Received

StatisticID: pr (0x0005)

Description: terminationがNULL contextに存在していた、もしくはNULL contextから外れてから、terminationもしくはstreamから受信されたパケット数を提供する。

Type: Double

Possible Values: 任意の64ビットの0以上の整数

Level: Either

#### E.12.4.3 Packet Loss

Statistic Name: Packet Loss

StatisticID: pl (0x0006)

Description: RFC 3550で定義しているように、RTPストリームの現在のパケット損失率を定義する。パケット損失率はパーセンテージで表される。その値は、ある間のパケットロスとその間で期待されるパケット数で割ったものである。

Type: Double

Possible Values: 32ビット整数と32ビット分数

Level: Either

#### E.12.4.4 Jitter

Statistic Name: Jitter

StatisticID: jit (0x0007)

Description: IETF RFC 3550で定義されるRTPストリームの到着間隔ジッタの現在値を要求する。Jitterは、RTPデータパケットの到着間隔時間の変動を測定する。

Type: Double

Possible values: 任意の64ビットの0以上の整数

Level: Either

#### E.12.4.5 Delay

Statistic Name: Delay

StatisticID:delay (0x0008)

Description: timestamp unitsで表現されるパケット伝播遅延の現在値を要求する。平均待ち時間と同じである。

Type: Double

Possible values: 任意の64ビットの0以上の整数

Level: Either

#### E.12.5 Procedures

RTCPがRTPストリームに関係している場合、RTCPはLocalControlディスクリプタ中のH.248.1モード特性に影響されないままでなければならない。

RTCPが、RTPストリームや、その流れのためのMG受信、およびEmpty Remoteディスクリプタに関係している場合、MGは対応するRTPストリームと共にRTCPストリームを停止しなければならない。

### E.13 TDM Circuit パッケージ

Package Name: TDM Circuit Package

PackageID: tdmc (0x000d)

Description: 本パッケージは、利得とエコーの制御をサポートするTerminationで使用することができる。当初はTDM回線での使用を意図していたが、もっと広く使用されてもよい。

本パッケージの新しいバージョンもしくは拡張はTDM以外の使用を考慮すべきである。

Version: 1

Extends: Network Package version 1

#### E.13.1 Properties

##### E.13.1.1 Echo Cancellation

Property Name: Echo Cancellation

PropertyID: ec (0x0008)

Type: Boolean

Possible Values:

"True" (エコーキャンセルが要求される時)

"False" (エコーキャンセルをオフにする時)

Default: Provisioned

Defined In: LocalControlDescriptor

Characteristics: Read/Write

#### E.13.1.2 Gain Control

Property Name: Gain Control

PropertyID: gain (0x000a)

発信信号レベルを適合させるために、利得制御、もしくは信号レベルアダプテーションおよび雑音レベルリダクションの処理が使用される。しかし、例えばモデム呼の場合は、この機能をオフにする必要がある。値が"automatic"に設定される場合、MGと外部方向の上で供給された目標レベルでterminationは自動レベル制御(ALC)としてサービス提供する。

Type: integer

Possible values: 利得制御は、デシベル（肯定的、あるいは否定）で利得を指定する。その値は最大値、214748647(0x7fffffff)で指定したり、"automatic"を表示するために保存された値であったりする。

Defined In: LocalControlDescriptor

Characteristics: Read/Write

#### E.13.2 Events

定義なし

#### E.13.3 Signals

定義なし

#### E.13.4 Statistics

定義なし

#### E.13.5 Procedures

定義なし

#### E.14 Segmentation パッケージ

Package Name: Segmentation Package

PackageID: seg (0x000x)

Description: セグメント化されていないトランスポート上でH.248に基づいたセグメント化を行う場合、本

パッケージは、使用するための特性を定義する。

Version: 1

Extends: root version 2

## E.14.1 Properties

### E.14.1.1 MG Segmentation Timer Value

Property Name: MGSegmentationTimerValue

PropertyID: MGSegmentationTimerValue (0x0009)

Description: 一旦SegmentationCompleteTokenが受信されると、MGCがMGから未解決のメッセージセグメントの受信を期待するべき時間を示す。最初はMGProvisionalResponseTimerValueとして設定されるが、変更されるかもしれない。

Type: Integer

Possible values: ミリ秒単位で任意の正の整数

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

### E.14.1.2 MGC Segmentation Timer Value

Property Name: MGCSegmentationTimerValue

PropertyID: MGCSegmentationTimerValue (0x000A)

Description: 一旦SegmentationCompleteTokenが受信されると、MGがMGCから未解決のメッセージセグメントの受信を期待するべき時間を示す。最初はMGProvisionalResponseTimerValueとして設定されるが、変更されるかもしれない。

Type: Integer

Possible values: ミリ秒単位で任意の正の整数

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

### E.14.1.3 MG Maximum PDU Size

Property Name: MGMaxPDUSize

PropertyID: MGMaxPDUSize (0x000B)

Description: 制御関連のトランスポートプロトコルのためにMGから入ってくるプロトコルデータユニットの最大値を示す。MGCはこのサイズを超過するメッセージを構築することを回避するべきである。

Type: Integer

Possible values: バイト単位で任意の正の整数

Default: Provisioned

Defined in: TerminationState

Characteristics: ReadOnly

#### E.14.1.4 MGC Maximum PDU Size

Property Name: MGCMMaxPDUSize

PropertyID: MGCMMaxPDUSize (0x000C)

Description: 制御関連のトランスポートプロトコルのためにMGCから入ってくるプロトコルデータユニットの最大値を示す。MGはこのサイズを超過するメッセージを構築することを回避するべきである。

Type: Integer

Possible values: バイト単位で任意の正の整数

Default: Provisioned

Defined in: TerminationState

Characteristics: Read/Write

#### E.14.2 Events

定義なし

#### E.14.3 Signals

定義なし

#### E.14.4 Statistics

定義なし

#### E.14.5 Error Codes

Error Code #: 459

Name: Segments not received

Definition:セグメント化されたTransactionReplyの受け取りがタイムアウトしたことを、このエラーは示す。転送されるすべてのセグメントのための待ち時間に対して。

Error Text in the Error Descriptor: ミスがあるセグメント番号が含まれている。

Comment: -

#### E.14.6 Procedures

いくつかのトランスポートが最大転送単位を超過するメッセージを自動的にセグメント化されないので、メッセージは結局切り捨てられるかもしれない。問題のメッセージが多数のTransactionRepliesを含んでいる場合、送信者は分割されたメッセージ中の各TransactionReplyを送信するべきである。MTU長さを超過するメッセージが残る場合、送信者はTransactionRepliesのための次のセグメント化処理を実装するかもしれない。TransactionRequestsはセグメント化してはならない。

処理応答セグメント化する場合、送信者はメッセージが完全なコマンドおよび(または)アクションを含むことを保証しなければならない。各セグメントはその処理レベルで補足セグメント情報によって示される。各セグメントは同じtransactionIDを使用しなければならない。また次のセグメント中のコマンドや、完了したアクションを繰り返さしてはならない。各セグメントには1から始まって最後のセグメントまで継続し

て番号を付けなくてはならない。それはSegmentationCompleteTokenを含むことによって表示される。すべての場合において、セグメント化されたメッセージは構文上有効構成でなくてはならない。受信者はセグメント化された応答を利用して、各セグメントに対して順番に応答する。セグメントがそれ自体、完全なメッセージであるので、受信の実体は、特別なセグメントを処理する前に次のセグメントを待つ必要はない。

受信者はメッセージセグメントをすべて受け取ったならば、最終セグメントの受信で決定しなくてはならない。受信者は、TransactionResponseAcknowledgementで応答しなくてはならない。

受信者がすべてのセグメントを受け取らなかった場合、タイマーを開始し、任意の未到達のセグメントを待たなくてはならない。タイマーの長さは受信者側で、前もって決定されておかなくてはならない。しかし、そのタイマーは、別の実体のProvisionalResponseTimerValue(付属資料E.2.1を参照)と同じ長さであるべきである。タイマーが終了する前に、未到達のセグメントが到着しない場合、受信者はエラーコード459で応答しなくてはならない。("Segments not received")。

例 1:

```
Sender:    !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}
Receiver:  !/3 [12.34.56.79]:2944 SM=1/1
Sender:    !/3 [12.34.56.78]:2944 P=1/2{C=1{AV=term3{...}}, C=2{AV=term4{...}}}
Receiver:  !/3 [12.34.56.79]:2944 SM=1/2
Sender:    !/3 [12.34.56.78]:2944 P=1/3/#{C=3{AV=term5{...}}}
Receiver:  !/3 [12.34.56.79]:2944 SM=1/3/#
Receiver:  !/3 [12.34.56.79]:2944 K=1
```

Example 2:

```
Sender:    !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}
Receiver:  !/3 [12.34.56.79]:2944 SM=1/1
Sender:    !/3 [12.34.56.78]:2944 P=1/4/#{C=3{AV=term5{...}}}
Receiver:  !/3 [12.34.56.79]:2944 SM=1/4/#
/* Segmentation Timer Expires */
Receiver:  !/3 [12.34.56.79]:2944 ER=459{"2,3"}
```

## E.15 Notification Behaviour

Package Name: Notification Behaviour Package

PackageID: nb (0x009a)

Description: 本パッケージは、MGCの要求でMGがNotifyコマンドの送信規制が可能となる、機能を有している。本パッケージは節7.1.9に記述されたNotifyBehaviourフラグを備えた相互作用を持っている。このパッケージのバージョンは、パーセンテージ規則に基づいた規制の動作を記述している。他のタイプの規制の動作は今後の課題である。

Version: 1

Extends: None

## E.15.1 Properties

### E.15.1.1 Notification Regulation

Property Name: Notification Regulation

PropertyId: notreg (0x0001)

Description: 抑えられるべきMG上のnotification（通知）のパーセンテージを示す。0%の値は抑制なしを示す。100%の値は、すべてのnotifies（通知）が抑制されるべきであることを示す。この特性は、Root Termination上でのみ定義されなければならない。" Regulated Notify"が特別なイベントで送信された場合、MGは、規制する（つなわり抑制する）Notifies（通知）の正確なパーセンテージを決定するためにNotification Regulation Propertyを使用する。

Type: Integer

Possible values: 0 から100まで

Default: None

Defined in: TerminationState

Characteristics: read / write

## E.15.2 Events

定義なし

## E.15.3 Signals

定義なし

## E.15.4 Statistics

定義なし

## E.15.5 Procedures

### E.15.5.1 NotifyBehaviour

NotifyBehaviourのメカニズムは、MGCにMGからのNotify.requestコマンドをどのように受信すべきかを管理させるものである。

#### E.15.5.1.1 NeverNotify

Notify.requestコマンドを受信することに、MGCが関心を持たないかもしれない実例があるかもしれない。例をあげる。このイベントの発生の検知が、組み込まれている信号やイベントを引き起こすために不可欠な場合があるのだが、MGCは通知される必要はない。あるイベントで" NeverNotify"に対してNotifyBehaviourパラメータを設定することは、そのイベントの検知がMGCに通知されないことを保証している。しかしながら、あらゆる組み込まれている信号やイベントは実行される。NeverNotifyは、Notification Regulation Property（通知規則特性）(nb/notreg)に対して全く関連を持たない。マークされたイベントは、規制計算に含まれない。

#### E.15.5.1.2 NotifyImmediate

たとえ他のいくつかの出来事が規制されていても、MGCは、" Emergency Call"イベントのような重要なイベントを通知されることを希望してもよい。これは、特別なイベントのための" NotifyImmediate"に対して

NotifyBehaviourパラメータを設定することによって達成される。イベントが検出される場合、これは現在のデフォルトの振る舞いである。NotifyImmediateは、Notification Regulation Property (通知規則特性) (nb/notreg)に対して全く関連を持たない。マークされたイベントは、規制計算に含まれない。

### E.15.5.1.3 RegulatedNotify

#### E.15.5.1.3.1 Percentage Based Regulated Notify

NotifyBehaviourパラメータが、ある特別なイベントで" NotifyRegulated"を設定される場合、あらゆる特別なイベント発生のnotification (通知) をMGCへ送信する意思決定は、Root Termination上でNotification Regulation Property (通知規制特性) (nb/notreg)によって管理される。この特性の値は、MGCによって、設定され、決定される。例えば、規制割合はMGCの輻輳レベルに相当する。MGには、全MGを横断して"NotifyRegulated"としてマークされるイベントの中で、Notify.Commandsの正確なパーセンテージを決定する責任がある。MGは、設定される通知規制のパーセンテージを達成するすべてのアルゴリズムを実装してもよい。異なるMGの間の同期の危険が最小限にされるように、アルゴリズムが選択されるべきである。示唆されたアルゴリズムは、以下に与えられる: 与えられた規制割合(例えば10%)については、MGが、RegulatedNotifyフラグを持っていたイベントのために下記を行う。:

1. notifications (通知) の数を抑制する。(例えば、Notification 1)
2. NotificationRegulationパーセンテージが到達する(例えば、9つの通知を受け取り)まで、その後の多くの通知を受け取る。
3. 規制パーセンテージが変化されるまで、ステップ1および2を繰り返す。
4. 同期問題を回避するために、MGは示唆された上記のシーケンスの任意の位置で、スタートするべきである。

あるイベントが、" RegulatedNotify"に設定されたNotifyBehaviourパラメータを持っていたり、通知規制特性が0にセットされる場合、これは、そのイベントがイベント通知のためのデフォルトの振る舞いである" ImmediateNotify"に設定されたことと同じような効果を持つであろう。

NotifyBehaviourメカニズムは正しくない状態を回避するために注意して使用されるべきである。例をあげる。"off-hook"のイベント通知がMGによって抑制されたならば、MGCは正しい応答(例えば、dial tone、congestion tone)を提供することはできないであろう。そのときに関わったMGC内の適切な応答を返すために組み込まれた信号、あるいは組み込まれたイベントのメカニズムの使用を通じて、この正しくない振る舞いは回避される。MGCは以降の2つを両方設定してもよい。1つは、" NotifyRegulated"とマークされたイベントが検知されたり、受理(つまり、抑制されない通知)されたりするときに発生する組み込まれた信号あるいは、イベントディスクリプタ。もう1つは、イベントが検出されたり、抑制(つまり、抑制された通知)されたりするときに発生する個別に抑制され組み込まれた信号あるいは、イベントディスクリプタ。

注意: 上記の場合でさえ、MGは、1つの呼を構成するものに気づかない。それは、単に呼制御の機能ではなくあらかじめプログラムされたタスクで行うものである。



### E.15.5.2 Example Scenario

MGCとMGはアナログ回線をどこで終端するか、というシナリオを与えられると、MGCが過負荷状態かどうか、アナログ回線（ALN）に緊急通報が起こっているかどうか、に基づいてMGC/MGは異なる振る舞いをしてもよい。その異なる振る舞いはテーブル1の中で示される:

**Table E.1/H.248.1 – 潜在的 MGC/MG の過負荷状態の振る舞い**

MG の振る舞い Task: “アナログ回線呼の信号制御(ALS)” = call control signalling traffic for POTS terminals (ALN; analogue line interfaces)		(1) 呼/コンテキスト タイプ	
		緊急通報	非緊急通報
(2) MGC の負荷状態 注意 – 2 つの状態を仮定	非過負荷状態  (e.g.) 低、中、高負荷	(I) MG: 最優先での通常 ALS 処理	(II) MG: 低優先での通常 ALS 処理
	過負荷状態	(III) MG: 通常 ALS 処理 ==>“usual H.248 mode” で、 MGC へ転送される 呼処理信号	(IV) MG: プログラムされたローカルな ALS 制御 ==> MGC は未関連 ==> “congestion tone”によって従順に拒絶された呼制御

#### E.15.5.2.1 Quadrant I and II Behaviour

MGCが過負荷になく、MGからメッセージ(通知)をすべて受け取ることができるように、特別なNotifyBehaviourはMGの上で要求されない。しかしながら、一旦MGCが呼のタイプを決定したならば、それは適切なContext Attribute (状況属性) の使用によりMGの中で優先事項に設定されてもよい。

#### E.15.5.2.2 Quadrant III and IV Behaviour

それが過負荷状態であることをMGCが決めた場合、それはすべてあるいはNULLContext中のterminationsのある選択されたグループに下記コマンドを送信してもよい。

Example message:

```
Transaction=1234{
  Context = - {
    Modify = aln/* {          ; select terminations as desired
      Events = 1234 {
        al/of {
          RegulatedNotify {
            Embed {
              Signals {cg/dt} ,
; If the off-hook is regulated apply dial tone.
            Events = 1235 {
              xdd/xce {
                DigitMap = PriorityDialPlan1,
                bc = 20,
                mp = enhanced,
                ImmediateNotify
              },
; If emergency generate notify to MGC
              xdd/xce {
                DigitMap = NonPriorityDialPlan1,
                bc = 20,
                mp = enhanced,
                NeverNotify,
                Embed{
                  Signals {cg/ct}
                }
              },
; If non-priority send congestion tone and never notify
        al/on {
```



terminationをリセットしてもよい。一方では、MGCは0に等しいNotification Regulation property(通知規則特性)を設定するべきである。

## 付属資料F サービスチェンジ手順 (ServiceChange Procedures)

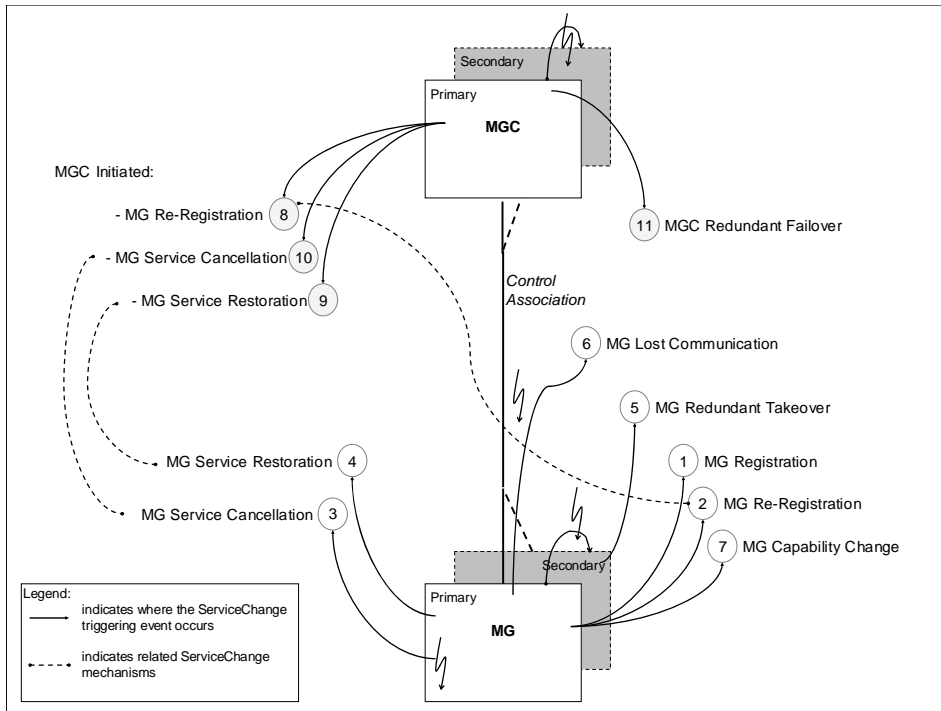
### F.1 導入

本付属資料は特定のイベントがMGまたはMGCに起こるときの、サービスチェンジ手順の活用法を解説する。それはサービスチェンジ手順を明確にさせるのに役立つ、7.2.8項/H.248.1と11項/H.248.1で記述されている。本付属資料とH.248.1の間の相違がある所は、H.248.1の手順は本付属資料で記述されるものを優先とする。

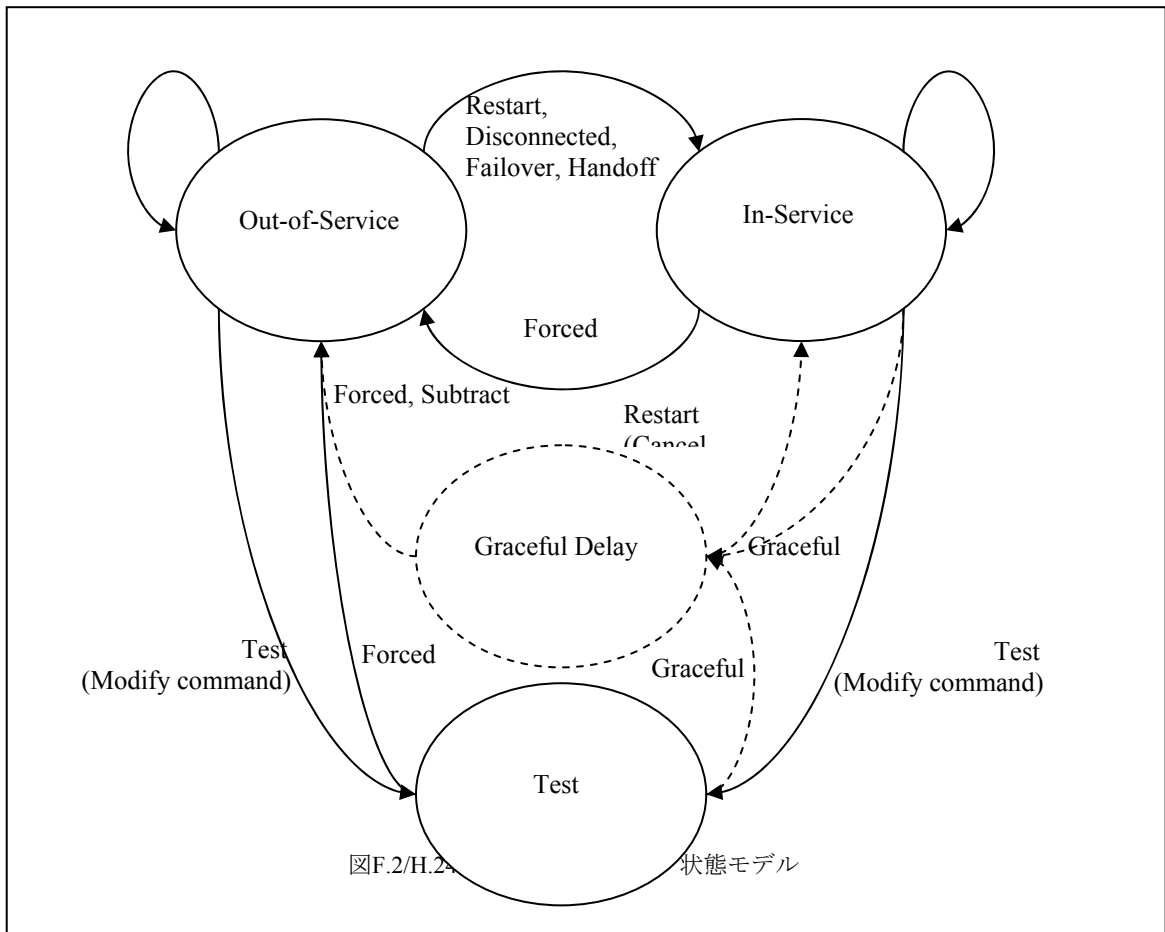
本付属資料で記述されるServiceChange Commandを送信につながるイベントは以下の通りである。

1. MG Registration—MGC に MG を登録し、制御リンクを確立する。
2. MG Re-Registration—MGC より MG 登録を促された場合の MG 登録。
3. MG Service Cancellation— MG は MG が全体として OutofService となることを MGC に通知する。
4. MG Service Restoration—MG 故障復旧の後、復旧を MGC に通知する。
5. MG Redundant Takeover—セカンダリ MG は、故障または MG のメンテナンスにおいて切り替えを行う。
6. MG Lost Communication—MG は MGC がコミュニケーションを失ったことを MGC に通知する。現在はこのコミュニケーションは回復。（瞬断）
7. MG Capability Change — MG の terminations 数やMG が全体として変えたことを MGC に通知する。
8. MGC Initiated MG Re-Registration- MGC は MG が MGC またはもう一つの MGC に再登録しなければならないことを MG に通知する。
9. MGC Initiated Service Restoration—MGC- MGC は失敗の後に回復しており、最初の状態に関連した terminations を置くように MG に指示する。
10. MGC Initiated Service Cancellation — MGC は、termnation または MG が OutofService になったことを MG に通知する。
11. MGC Redundant Failover — プライマリ MGC は障害となっており、特定のセカンダリ MGC に登録するように MG に指示する。

図F.1にこれらのServiceChangeのトリガとなるイベント、イベントが発生するエンティティ、どのイベントの対が関連しているかを示す。図F.2にMGとMGのterminations の状態のモデルを示す。



図F.1/H.248.1 サービスチェンジの為にトリガ



F.2 制御リンク(Control association)の定義

制御リンクはMGCがMGを制御するための通信に関わる結びつきを表している。制御リンクはregistrationにより確立する。制御リンクはOutOfServiceに移行するMG、代替MGCへのハンドオフに成功したMG、代替MGCへの故障切り替えに成功したMGによって終端される。物理MGが1つ以上のVirtual MGsに分割する場合を除き、MGにはいつでも多くても1つの制御リンクが存在しなければならない。Virtual MGの場合では、各Virtual MGは多くても1つの制御リンクが存在しなければならない。MGsだけが制御リンクを示すこともできる。ここで定義される制御リンクはメディアゲートウェイコントロールシグナリングに関する範囲内で適用される。この定義はトランスポート(TCP、UDPなど)に関するContextで制御リンクのものと混同するべきではない。Root TerminationにおけるServiceChange Commandsの使用はRoot Terminationが属するVirtual MGに影響するだけである。同じ物理的MG上の他Virtual MGsは影響を受けない。

### F.3 ServiceChange手順を起こすイベント

#### F.3.1 MG Registration

MGは以下の3つの方法のうちの1つで登録されてよい:

- その初期存在のアナウンス、あるいは ServiceChangeReasons 901 (Cold Boot) または 902 (Warm Boot) を伴う ServiceChangeMethod 「Restart」 を用いて 「registration」 する。
- ServiceChangeReason 909(「MGC Impending Failure」)または 908(「MG Impending Failure」) を伴う ServiceChangeMethod 「Failover」 によって制御リンクの変更を指示する。
- ServiceChangeReason903(「MGC Directed Change」)を伴う ServiceChangeMethod"Handoff"によって制御リンクの変更を指示する。(F.3.11 節)

登録は、初期のServiceChange CommandをMGからMGCに送信されるときに開始され、MGCから代替アドレスやエラー無しにコマンド応答が受信され、MGCとMGの間でServiceChangeProfileが合意された時に終了する。制御リンクは登録の完了時に確立される。

登録は常にRoot Terminationによって起こる。MGCとMGとのどんな既存の制御リンクも新しい登録の開始のときに終端する。前の制御リンクからのすべてのコマンドは無視されなければならない。

ServiceChangeProfileは登録で使用される。詳細はF.5.5節を参照。

ServiceChangeVersionは登録で使用される。詳細はF.5.6節を参照。

#### F.3.2 MG Re-Registration

MG Re-Registrationは2つの場合で起こる。:

1. MGC が MG に再登録を要求するとき(F.3.8 節)。その時 MG は ServiceChangeMethod"Handoff"、ServiceChangeReason903(MGC Directed Change) および明示的な "ServiceChangeMgcID" による ServiceChange を送信しなければならない。
2. MGC がサービス回復(節 F.3.9)を開始するとき(F.3.9 節)。MG は ServiceChangeMethod 「restart」 による ServiceChange を送信することにより 「reregister」 しなければならない。

コマンドが代替アドレスおよびエラーなしでMGCによって応答され、MGCとMGの間でServiceChangeProfileが合意されるとき再登録は完了する。第一のMGCから応答がない場合、MGはF.3.1節の手順に従うものとする。ServiceChangeReasons 903 (MGC Directed Change) または909 (MGC Impending Failure) が推奨される。制御リンクはregistration終了後に確立される。registrationは常にRoot Termination上

に発生する。MGCとMGとのどんな既存の制御関係も再登録の開始のときに終端する。前の制御リンク上のすべてのコマンドは無視されるものとする。

ServiceChangeProfileはre-registrationで使用される。詳細に関してF.5.5節を参照。

ServiceChangeVersionはre-registrationで使用される。詳細に関してF.5.6節を参照。

### F.3.3 MG Service Cancellation

MGが「OutOfService」となるには、MGはRoot Terminationに対して「Forced」または「Graceful」ServiceChangeMethodというServiceChangeコマンドを送信する。詳細には、F.4.1.1節を参照せよ。

TerminationまたはTerminationsが「OutOfService」となるには、Terminationに「Forced」または「Graceful」ServiceChangeMethodというServiceChangeコマンドを送信する。詳細は、F.4.1.2節とF.4.1.3節を参照。ServiceChangeDelayはサービス取り消しが生じる前の時間を示す。詳細には、F.5.3節を参照。MGが「Restart」とServiceChangeReason 918「Cancel Graceful」のServiceChangeMethodでROOT Terminationの上でServiceChange Commandを送って、全MGのために以前に出された（受け付けられた）Gracefulをキャンセルする。

MGはInServiceのままではなければならない。そしてMGで特に指示されない限り、以前OutOfServiceとなった全TerminationsはInServiceとなる。DelayタイマーがタイムアウトになったあとCancel Gracefulが受け取られる場合には、Cold Bootを送ったように、MGが再登録re-registeredしたとみなされる。

Terminationで以前に出された（受け付けられた）Gracefulをキャンセルするために、MGはServiceChangeMethodの「Restart」を送信し、問題のTerminationの上でServiceChangeReasonに「Cancel Graceful」をServiceChangeを送る。Terminationは、InServiceのままではなければならない。TerminationがすでにOutOfServiceに移行した場合には、それがどんなServiceChange RestartでもInServiceになる。

### F.3.4 MG Service Restoration

MGService Restoration イベントはMGが障害またはメンテナンスアクションの後、サービスに戻るときに起こる。MGより送信される場合、MGはServiceChangeコマンドを再開したMGC、プロトコル版数またはH.248プロファイルの再ネゴシエート要求を行っているMGCに送信する。ServiceChangeMethod「Restart」とServiceChangeReason「Service restored」は本ケースで使用される。ServiceChangeReasonはMGCによってなされる必要があるであろう動作について示す。

### F.3.5 MG Redundant Takeover

2種類のMG冗長切り替えの方法がある。

#### 1. Takeover initiated by Primary MG:

MGがOutOfServiceになり、指定のセカンダリMGに処理を放棄したいというとき、MGはServiceChangeMethod “Failover” とServiceChangeReason of 908 (“MG Impending Failure”)を送信する。MGCはMGにメッセージを送るのをやめ、制御リンクを切断する。セカンダリMGは、それからServiceChangeMethod “Restart” とServiceChangeReason of 900 (“Service Restored”)を送信する。

#### 2. Takeover initiated by Secondary MG:



セカンダリMGがメンテナンスまたは故障によるプライマリMGの停止および、かつプライマリMGが故障をMGCに通知することができないことを検出した時、セカンダリMGはServiceChangeMethod「Failover」と919（“Warm Failover”）または920（“Cold Failover”）、のServiceChangeReasonでMGCにServiceChange Commandを送信する。F.3.4節の手順が適用される。

MGCは、未知のMGからの冗長切り替え要求を無視する。MGCは冗長切り替えをアナウンスしているセカンダリMGが、故障しているプライマリMGと関係付けられていることをプロビジョニングまたは他の手段で認識していなければならない。それゆえに、セカンダリMGはプライマリMGの切り替えを認可するという意識がなければならない。

セカンダリMGには知られていて、プライマリMGのために切り替え権限があるならば、MGCはそれがまだ機能するかどうか決定するためにプライマリMGと通信しようとしなければならない。これは、空のAuditValue Commandまたは他の適当な方法でなされてもよい（11.5節を参照）。プライマリMGから応答があるならば、MGCは切り替えを拒否する。プライマリMGが応答できないならば、MGCはそれが故障したと仮定して、warm or cold failoverを受け入れる。

MGCがwarm or cold failoverを受け入れるとき、プライマリMGとの接続なくなり、セカンダリMGとの新しい制御関係が確立される。

### F.3.6 MG Lost Communication

MGがMGCとのコミュニケーションの切断と後続の再接続を検出したとき、MGはServiceChangeMethodで現在の制御リンクでMGCにServiceChange Command ServiceChangeMethod「Disconnected」を送信する。

MGCが応答に失敗した場合、それがうまく新しい制御リンクの確立に成功するまで、あるいはリスト上のMGC全てに対して送信するまで、MGはリスト順に各々のMGCに“Failover”とServiceChangeReason 909（“MGC Impending Failure”）のServiceChange Commandを送信する。

元の制御リンクで関連付けられるMGCがServiceChange Commandに応答する場合、制御リンクは中断なく継続する。そして、まるでコミュニケーションのロストがなかったように全てのコマンドは処理される。

さもなければ、MGが新しいMGCにServiceChangeを送信するとき、元の制御リンクは終端され、そして、前の制御リンクからの全てのコマンドは無視される。

新しい制御リンクがひとたび確立され新MGCがServiceChangeに応答すると登録が完了する。

MGがうまく制御リンクを確立することなくMGCsリストを使い果たすならば、MGはランダムな時間を待ってから再びMGCsリストで登録を試みる。そして、最初の制御リンクからMGC接続を始める。MGがMGCに接続しようとするたびに、MGは元の制御リンクからServiceChangeMethod「Disconnected」でServiceChangeを送信する。MGは、他の全てのMGCsにServiceChangeMethod of “Failover”のServiceChangeを送信する。

ServiceChangeMethod「Disconnected」のServiceChangeコマンドを受信するMGCsは、メッセージを失ったことによる状態不一致かどうか決定するためにMGをAUDIT（検査）しなければならない。

特定のシナリオが異なるServiceChangeReasonコードを必要としてもよいが、これらのシナリオでは、ServiceChangeReason 900（Service Restored）が推奨される。

### F.3.7 MG Capability Change

MG能力の変化を通知するために、ServiceChangeMethod が ServiceChangeReason of 916 ( “Packages Change” ) または 917 ( “Capabilities Change” ) の ServiceChangeを送ります。MGCは、MG新しい能力を決定するために、MGをAUDIT(検査)しなければなりません。

with ServiceChangeMethod が “Restart” のServiceChange Commandを送るinservice MGのために、制御リンクは能力変化を通して中断されないままです。そして、MGはシステムリスタートを経たために考慮されません。

### F.3.8 MGC Initiated MG Re-Registration

MGCは、MGにServiceChangeMethodが “Handoff”かつServiceChangeReason 903 ( 「MGC Directed Change」 ) とそれ自身のServiceChangeMgcID (つまり、現在のMGCの) でルートの上でServiceChange Commandを出すことによって再登録することを要求してもよい。

MG側アクションは、F.3.2を参照すること。

### F.3.9 MGC Initiated Service Restoration

MGが再開するという要求に、MGCはServiceChangeMethod 「Restart」 と適切なServiceChangeReason900 「Service Restored」 or 901 「Cold Boot」 でMGにServiceChange Commandを送信する。MGは、示されたServiceChangeReasonを使っているF.3.2節に従って、新しい制御リンクを確立しなければならない。

Termination または Terminationsのサービスを復旧するために、MGCは問題のTermination/sの上に、ServiceChangeMethod 「restart」 でServiceChangeを送信する。MGCによってなされるアクションについては、F.4.1.2節とF.4.1.3節を参照すること

ServiceChangeDelayはサービス回復が起こる前までの時間を示す。詳細はF.5.3節を参照。

### F.3.10 MGC Initiated Service Cancellation

#### F.3.10.1 Root Termination

MGが「OutOfService」となる為に、MGCはServiceChangeMethod が「Forced」 または「Graceful」 のServiceChangeコマンドを送信する。その中に適切なServiceChangeReasonsは、905 (“Termination taken out of service”) を含んでもよい。MGCの動作については、F.4.1.1節を参照すること。

ServiceChangeDelayはサービスキャンセルが起こる時間を示します。詳細はF.5.3節を参照すること。

全MGのために以前に出された(認められた) Gracefulをキャンセルするために、MGCはServiceChangeMethod が 「Restart」 かつServiceChangeReason 918 「Cancel Graceful」 のServiceChangeコマンドをROOT Terminationに送信する。

MGはInServiceのままでなければならず、そして、さもなければMGで合図されない限り、以前OutOfServiceとなった全てのterminations はInServiceとなる。Delayタイマーが期限切れになったあとCancel Gracefulが受け取られる場合には、MGはエラーコード 502 ( 「Not Ready」 ) をMGCに報告する。MGは、サービスに戻るために再登録することを要求される。

#### F.3.10.2 Physical Terminations

TerminationまたはTerminationsのグループを「OutOfService」にするために、MGCはServiceChangeMethod

が「Forced」または「Graceful」のServiceChangeコマンドをTerminationに送信する。

ServiceChangeDelayは、サービスキャンセルが起こるまでの時間を示す。詳細はF.5.3節を参照。

Termination の上で以前に出された（そして、受け付けられた）Gracefulをキャンセルするために、MGは問題のTermination/sの上にServiceChangeReason 918「Cancel Graceful」かつServiceChangeMethodが「Restart」でServiceChangeを送信する。Terminationは、InServiceのままである。Terminationがすでに移行しOutOfServiceの場合には、それがどんなServiceChangeRestartでもサービスとなる。

### F.3.10.3 Ephemeral Terminations

MGCは、短命なTerminationに対するサービスをキャンセルするために、ServiceChangeを使わない。Terminationをcontextから削除することは、Terminationを削除するのに十分である。

### F.3.11 MGC Redundant Failover

制御リンクのMGCがメンテナンスまたは故障に遭遇する時はOutOfServiceにならなければならない。そのとき、ServiceChangeMethod が“Handoff” かつ ServiceChangeReason of 903「MGC Directed Change」と新しいMGCのアドレスのServiceChangeMethod のServiceChangeコマンドを送信することによって特定のセカンダリのMGCにMGを向けてもよい。

制御リンクは、MGからコマンドリプライの受信で終了される。MGは、ServiceChangeMethod が“Handoff” かつ ServiceChangeReason of 903「MGC Directed Change」で指定されたMGCにServiceChange Commandを送信する。新しい制御リンクは、MGCからコマンドリプライの受信によって確立される。

指定されたMGCがHandoff試みを拒否する、あるいは、MGCは失敗によるServiceChangeMethod HandoffでServiceChange Commandを送信することができない場合には、MGはF.3.6節で概説されるロストコミュニケーションの手順に戻る。

## F.4 ServiceChange要素の記述

### F.4.1 ServiceChangeMethod

この説では、異なるTerminationタイプにおけるServiceChangeMethodの振る舞いについて記述する。この説は以下の内容より構成される。

- Root Termination
- PhysicalTerminations
- Ephemeral Terminations

#### F.4.1.1 Root Termination におけるServiceChange Methodの動作

Root Termination 上のServiceChange Commandは、ServiceChangeMethodによって異なる効果を持つ。それぞれのServiceChangeMethodの結果を以下に記載する。

1. Restart –MG がリスタートしたことを宣言するために送信する Restart は、プロトコルバージョン、プロファイルの再交渉、または能力の交換を望む。ServiceChangeReason は、どのような動作が

MGCによって取られるべきかを示す。RestartがMGCから送信される場合は、ServiceChangeReasonに従ってMGがリスタートしなければならない。

2. Forced –MGから送信される場合、MGは即座に自身をOutOfServiceに移行させることを示す。MGCからコマンドの返信を受信すると、制御関係はMGによって終端される。ForcedがMGCから送信される場合、MGは即座に自身をOutOfServiceに移行させ、コマンドの返信を送信した後に制御関係を終端しなければならない。MGCは、MGに対してOutOfServiceに移行することを要求しても良いが、サービス状態に戻らせることは出来ないことに注意すること。MGが新たな制御関係を確立するためには、他の登録の場合と同様に、登録を始め、それからサービス復旧を示す。ServiceChangeDelayはServiceChangeMethod“Forced”に影響を与えない。
3. Graceful –MGから送信される場合、ServiceChangeDelay期間の後にMGがOutOfServiceに移行することを示す。制御関係は、ServiceChangeDelay期間の後に終端される。GracefulがMGCから送信される場合、MGは、ServiceChangeDelay期間の後に自身をOutOfServiceとして制御関係を終端しなければならない。ServiceChangeDelayが0またはServiceChangeDelayがない場合は、MGは最後のcontextがそのTerminationの削除を通じて切り離される時、OutOfServiceに移行して制御関係を終端しなければならない。そして、MGCは新たな接続を付加してはならない。MGはServiceChangeDelayの満了時、または、アクティブなcontextから全てのTerminationを削除した時（のどちらか早い方）に、Root TerminataionのServiceStates Propertyを"OutOfService"に設定すべきである。先に送付され（承認された）“Graceful”のServiceChangeMethodを伴うServiceChangeをキャンセルするためには、“Graceful”のServiceChangeMethodを開始したエンティティが、RestartのServiceChangeMethodとServiceChangeReason 918 (“Cancel Graceful”)を伴うServiceChangeコマンドを送信する。
4. Failover –MGから現在制御関係にないMGCに対して送信される場合、MGは現在のMGCが故障し、そのコマンドの受信と共に登録しようとしていることを示す。MGから現在制御関係にあるMGCに対して送信される場合、プライマリMGはセカンダリMGが故障したプライマリと交代したことを示す。いずれかの状況において、以前の制御関係はServiceChangeコマンドの送信によって終端される。そのコマンド返信を受信されると、MGと新しいMGCとの間、または、MGCと新しいMGとの間に、新たな制御関係が確立される。MGCは、ServiceChangeMethod“Failover”を伴うServiceChangeを送信してはならない。
5. Handoff –MGCから送信される場合、そのMGCは、新たなMGCに移行中であることを示す。これは、そのコマンドの返信を受信すると、現在の制御関係を終端する。MGからMGCに送信される場合、それは、そのMGが以前の制御状態にあるMGCから受信したハンドオフコマンドによって、新たな制御関係を確立しようとしていることを示す。MGからMGCに対して送信されるとき、ハンドオフが登録され、そしてそのコマンドの返信を受信されると新たな制御関係が確立される。MGは、MGCからコマンドを受けずにHandoffを用いてはならない。
6. Disconnected –MGから送信される場合、それは、現在の制御関係の通信が失われたが、現在、再確立されていることを示す。その現在の制御関係は回復されている。MGCは“Disconnected”のServiceChangeMethodを伴うServiceChangeコマンドを送信してはならない。

#### F.4.1.2 物理Termination上でのServiceChange Methodの振る舞い

MGがアクティブな制御関係にあるとき、物理Termination上でのServiceChangeコマンドはServiceChangeMethodによって異なる効果を持つ。それぞれのServiceChange Methodの結果は以下の通りとなる。

**Restart** – MGから送信される時、そのMGはTerminationがリスタートしたことを通知するか、または、能力変更を通知している。ServiceChangeReasonはMGCによって取られるべき必要があるかもしれない動作を示す。RestartがMGCから送信される時、MGは示されたServiceChangeReasonを用いてTerminationをリスタートしなければならない。

**Forced** – MGから送信される時、それはTerminationが直ちにOutOfServiceに移行することを示す。MGCから送信される時、MGはTerminationを直ちにOutOfServiceとしなければならない。どちらの場合も、ServiceStatusパラメータはOutOfServiceに設定されなければならない。MGCは任意のcontextまたはそのTerminationに関する資源を一掃する責任を負う。ServiceChangeDelayはServiceChangeMethod“Forced”に影響を与えない。

**Graceful** – MGから送信される場合、それはTerminationがServiceChangeDelay後にOutOfServiceに移行することを示す。MGCから送信される場合、MGはTerminationをServiceChangeDelay後にOutOfServiceに移行させなければならない。ServiceStatesプロパティはServiceChangeDelayの失効後か、または、Terminationがアクティブなcontextから切り離された時（の何れか早い方の契機で）、OutOfServiceに設定されなければならない。そしてMGCは任意のcontextまたはそのTerminationに関する資源を一掃する責任を負う。0の値のServiceChangeDelayの使用、または、ServiceChangeDelayが無いことは、Terminationが削除を通じて切り離される時、そのTerminationがOutOfServiceに移行しなければならないことを示す。そのGracefulがキャンセルされるかそのTerminationが引き続いて発行されるServiceChangeコマンドによってサービス状態に戻るまで、MGCは、指定されたTerminationを接続のために用いてはならない。以前送信された（そして受け付けられた）“Graceful”のServiceChangeMethodを伴うServiceChangeをキャンセルするためには、そのGracefulを開始したエンティティがServiceChangeMethod“Restart!”とServiceChangeReason 918 (“Cancel Graceful”)を伴うServiceChangeを送信する。

**Failover** –ServiceChangeMethod“Failover”は、Root Termination以外で用いられてはならない。

**Handoff** –ServiceChangeMethod“Handoff”は、Root Termination以外で用いられてはならない。

**Disconnected** –ServiceChangeMethod“Disconnected”は、Root Termination以外で用いられてはならない。

#### F.4.1.3 一時的なTerminationにおけるServiceChange Methodの振る舞い

MGがアクティブな制御関係にある時、一時的なTerminationへのServiceChangeコマンドの発行は、ServiceChangeMethodに依存して異なる効果を持つ。それぞれのServiceChange Methodの結果は以下の通りとなる。

**Restart** – MGから送信される時、そのMGはTerminationがリスタートしたことを通知するか、または、能力変更を通知している。ServiceChangeReasonはMGCによって取られるべき必要があるかもしれない

れない動作を示す。MGC は、一時的な termination に対して ServiceChangeMethod ”Restart”を送信してはならない。

**Forced** – MG から送信される時、それは Termination が直ちに OutOgService に移行することを示す。MGC はその Termination を削除する責任を負う。MGC は、一時的な Termination に“Forced”の ServiceChangeMethod を送信してはならない。ServiceChangeDelay は ServiceChangeMethod “Forced”に影響を与えない。

**Graceful** – MG から送信される場合、それは Termination が ServiceChangeDelay 期間の終了で OutOfService に移行することを示す。MGC は、ServiceChangeDelay の失効時にその Termination を削除する責任を負う。MGC は、一時的な Termination に ServiceChangeMethod “Graceful”を送信してはならない。0 の値の ServiceChangeDelay の使用は、その Termination が削除を通じて context から切り離されるときに破壊されなければならないことを示す。MG は、ServiceChangeDelay の失効かまたはアクティブな context からその Termination が切り離されたとき（の何れか早い方で）、その Termination の ServiceStatus Property を OutOgService に移行すべきである。以前送信された（そして受け付けられた）“Graceful”の ServiceChangeMethod を伴う ServiceChange をキャンセルするためには、その Graceful を開始したエンティティが ServiceChangeMethod “Restartl”と ServiceChangeReason918 (“Cancel Graceful”)を伴う ServiceChange を送信する。

**Failover** –ServiceChangeMethod “Failover”は Root Termination 以外で使用されてはならない。

**Handoff** –ServiceChangeMethod “Handoff”は Root Termination 以外で使用されてはならない。

**Disconnected** –ServiceChangeMethod “Disconnected”は Root Termination 以外で使用されてはならない。

## F.5 ServiceChangeパラメータの使用

### F.5.1 ServiceChangeMethod

使用法については、F.4節を参照。

### F.5.2 ServiceChangeReason

ServiceChangeReasonは、それを受信する当事者に対して、特定の状況に合うようにその振る舞いを変更することを許す。例えば、MGがリスタートするとき、MGCに対してServiceChangeReason 901 (“Cold Boot”)を送信したら、MGCは、MGが全ての状態を失ったものと仮定し、従ってメンテナンスを実行せずにMGを評価して仕様可能な状態に一掃するためにタスクを検査するだろう。

Table F.1は、各ServiceChangeReasonと共に送信されて良いServiceChangeMethodを示す。

**Table F.1/H.248.1 ServiceChangeMethod と ServiceChangeReason の対応**

SC Reason	SC Method						説明
	Restart	Forced	Graceful	Disconnected	Failover	Handoff	
900	X			Root のみ MG のみ			Service Restored
901	Root のみ						Cold Boot

902	Root のみ						Warm Boot
903						Root のみ	MGC Directed Change
904		X	X				Term Malfunction
905		X	X				Term Taken OOS
906		X	X				Loss of lower layer connectivity
907		X	X				Transmission failure
908		Root のみ MG のみ	Root のみ MG のみ		Root のみ MG のみ		MG Impending Failure
909					Root のみ MG のみ		MGC Impending Failure
910	MG のみ	X	X				Media Capability Failure
911	MG のみ	X	X				Modem Capability Failure
912	MG のみ	X	X				Mux Capability Failure
913	MG のみ	X	X				Signal Capability Failure
914	MG のみ	X	X				Event Capability Failure
915		X	X				State Loss
916	X			Root のみ MG のみ	Root のみ MG のみ		Packages Change
917	X			Root のみ MG のみ	Root のみ MG のみ		Capability Change
918	X						Cancel Graceful
919					Root のみ MG のみ		Warm Failover
920					Root のみ MG のみ		Cold Failover

### F.5.3 ServiceChangeDelay

ServiceChangeDelayは、ServiceChangeコマンドが有効になり、MGのServiceStatus PropertyかまたはTerminationが変更されるまでの遅延時間を与える。そのServiceChange Commandは、そのメッセージ受信で実行されたように返信されるが、MGまたはTerminationは実際にはServiceChangeDelayが失効するまで変化しない。0に設定されたServiceChangeDelayは、ServiceChangeMethod “Graceful”を伴う場合を除いて、ServiceChangeDelayがない場合と等価である。様々なServiceChangeMethodにおけるServiceChangeDelayの効果については、節F.4を参照のこと。

### F.5.4 ServiceChangeAddress

ServiceChangeAddressパラメータの使用については、H.248.1の節7.2.8に記載される。ServiceChangeAddressの使用は思いとどまらせる。もしもそのパラメータが存在する場合、それはRoot Termination上でのServiceChangeコマンドと共にのみ使用されて良い。そして、任意の新たなtransactionが指定された新たなアドレスとポート番号に送信されなければならない。返信は、対応する要求が来たアドレ

スに対して返されなければならない。

#### F.5.5 ServiceChangeProfile

ServiceChangeProfileパラメータはMGCとMGに対して制御関係の中でしようされるH.248プロファイルの交渉を許す。MGCはサポートされる別のMG能力を決定するためにMGを検査しても良い。ServiceChangeProfileはregistrationまたはre-registrationコマンドでのみ送信されなければならない。

#### F.5.6 ServiceChangeVersion

ServiceChangeVersionは、MGCとMGとの間で使用されるH.248プロトコルバージョンの交渉のために用いられる。ServiceChangeVersionは初めのregistrationでは必須であり、その他のプロトコルバージョンの交渉が発生するregistrationでは送信されるべきである。ServiceChangeVersionはregistration以外のコマンドで送信されてはならない。Registrationの手続きについては節F.3.1を参照のこと。

#### F.5.7 ServiceChangeMgclid

MGCはRoot Terminationに向けたServiceChangeにこのパラメータを含めても良い。Registrationの試行中に受信すると、MGは指定されたアドレスのMGCにregistrationを試行する。MGのプライマリMGCからHandoffコマンドを受信したとき、MGはH.248.1の節11.5に概説された手続きを利用しなければならない。

#### F.5.8 TimeStamp

オプションのTimeStampパラメータの使用は、H.248.1の節7.2に記述される。TimeStampパラメータは、ServiceChangeコマンドの実行に影響を及ぼさないが、ServiceChangeコマンドの受信者がその他の目的、例えば課金やタイミング調整のために用いても良い。

### F.6 ServiceChange対TerminationState

TerminationState ディスクリプタ内のServiceStatusプロパティは、Termination状態のの現在値を保持する。ServiceStatesは“InService”、“OutOfService”、及び、“Test”と3通りの値を持つかもしれない。

MGCはServiceStatusプロパティをTestに設定するか、または、Testから他の設定にするために、Modify Commandのみを用いなければならない。ServiceStatesプロパティのその他全ての変更はプロトコル違反であり、Error Code 401 (“Protocol Error”)を受信するはずである。“InService”状態から“OutOfService”状態、またはその逆の状態を変更するためには、ServiceChangeコマンドが用いられなければならない。

テスト状態のTerminationのためには、MGは“Forced”または“Graceful”のServiceChangeMethod を伴うServiceChangeのみをMGCに送信しなければならない。引き続きのServiceChangeコマンドは、それからそのTerminationのServiceStateプロパティを“InService”に設定して良い。

受信者はServiceChangeコマンドを受け付けるためにあらゆる正当な試行を行うべきであるが、幾つかの場合においてはそのコマンドの拒否が認められる。例えば、もしもMGがあるTerminationをサービス状態にするためにServiceChangeコマンドを送信するが、準備不足またはその他の理由によってMGCがそのTerminationをサービスするために割り当てる資源を持っていない場合、MGCはそのコマンドを拒否しても良い。もしも受信エンティティがServiceChangeコマンドを拒否すると、Terminationの状態は変化しない。送信者はしばらく待って再試行するか、受信者がTerminationの状態を変えることを試行するのを待つか、



どちらを選んでも良い。TerminationをOutOfServiceにするための要求は尊重されるべきである。

表F.2はどのコマンドがTerminationの状態変化に影響を与えるために使用されて良いかを示している。

**Table F.2/H.248 状態遷移コマンドとその効果**

現在の状態	新しい状態	コマンド	MGCで許容?	MGで許容?
InService	Test	ServiceChange	不可能	不可能
		Modify	Yes	不可能
InService	OutOfService	ServiceChange	Yes, 物理 Termination と Root Termination のみ	Yes
		Modify	No	不可能
OutOfService	InService	ServiceChange	Yes, 但し Root を除く	Yes
		Modify	No	不可能
OutOfService	Test	ServiceChange	不可能	不可能
		Modify	Yes	不可能
Test	InService	ServiceChange	No	No
		Modify	Yes	不可能
Test	OutOfService	ServiceChange	No	Yes
		Modify	Yes	不可能

## 付録I コールフローの例

(本付録は本標準の必須部分ではない)

JT-H248.1を用いて実装を行う場合、実装を行う前に、本勧告の標準規定部分を注意深く読まなければならない。H.248プロトコルのメッセージの作成例として、本付録の例を単独で参照してはならない。

本付録の例では、ローカルとリモートのストリーム記述の符号化にSDPを使用する。SDPはRFC 2327にて定義される。もし、この例のSDPとRFC 2327に相違がある場合、確認のためRFCを参照すべきである。使用されるオーディオプロファイルには、RFC 1890で定義されているものと、その他IANAに登録されているものがある。例えば、G.711 A-lawはSDPではPCMAと呼ばれ、プロファイル0に割り当てられている。G.723.1はG723と呼ばれ、プロファイル4である。H.263はH263と呼ばれ、プロファイル34である。  
<http://www.iana.org/assignments/rtp-parameters> を参照すること。

### I.1 レジデンシャルゲートウェイ間の呼

この例のシナリオでは、IPネットワーク上に存在する、あるレジデンシャルゲートウェイから別のレジデンシャルゲートウェイへ、呼を確立するまでのH.248プロトコルの使用法について記述する。単純化のために、本説明で使用される二つのレジデンシャルゲートウェイは同一のMGCによって制御されることとする。

#### I.1.1 アイドル中状態おけるレジデンシャルゲートウェイのアナログ回線terminationの設定

以下に、MGCとMGからAPIを発行し、本節で示すシーケンスで登場するterminationをアイドル状態に設定するまでの例を示す。発呼側と着呼側の両MGにはアイドル状態のアナログ回線terminationがあり、それらは呼の開始イベント(すなわち、オフック)を監視するように、適切なパラメータのModifyコマンドを用いて設定されている。NULL ContextはterminationがまだContextに含まれていないことを示すために使用する。Root Terminationは、MG内部にあるterminationではなくMGそのものの全体を示すために使用する。

この例では、MG1のIPアドレスが124.124.124.222、MG2のIPアドレスは125.125.125.111、MGCのIPアドレスは123.123.123.4である。これら三つはいずれもMegacoポートは55555を使用している。

1. MGがServiceChangeコマンドを用いてMGCにMG登録を行う：

```
MG1 to MGC:
MEGACO/1 [124. 124. 124. 222]
Transaction = 9998 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Restart, Version=3,
      ServiceChangeAddress=55555, Profile=ResGW/1}
    }
  }
}
```

2. MGCがreplyを送信する：

```
MGC to MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
  Context = - {ServiceChange = ROOT {
    Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}
```

3. MGCがNULL Context中に存在するterminationに設定を行う。対象となるTerminationID はA4444、StreamID は1、EventsディスクリプタのrequestIDは2222である。MIDはメッセージ送信者の識別子であり、この場合は送信者のIPアドレスとポートである[123.123.123.4]:55555になる。ストリームのModeモードプロパティはSendRecvにセットされる。"al"はanalog line supervisionパッケージを表している。LocalとRemoteのディスクリプタは事前にプロビジョニングにより規定されているものとする。

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 9999 {
  Context = - {
    Modify = A4444 {
      Media { Stream = 1 {
        Local Control {
          Mode = SendRecv,
          tdmc/gain=2, ; dB単位
          tdmc/ec=on
        },
      },
    },
    Events = 2222 {al/of {strict=state}}
  }
}
```

前もってMGにDialplanスクリプトをロードしておくことができる。Dialplanスクリプトの機能は、オフフックイベントを待ち、ダイヤルトーンを鳴らすとともにDTMF digitの収集を開始することである。しかしこの例では、DigitMapを使用し、DigitMapをオフフックイベントが検知された後に適用する(以下step5)。

なお、埋め込み型EventsDescriptorを使うことでstep3・step4とstep8・step9をまとめ、step6・step7を省くことも可能である。

4. MG1はModifyを受理し、replyを返す：

```
MG1 to MGC:
MEGACO/3 [124. 124. 124. 222]: 55555
Reply = 9999 {
    Context = - {Modify = A4444}
}
```

5. 同様の信号の送受信がMG2とMGCの間で行われ、MG2の空きterminationであるA5555に対して同様の処理が行われる。

#### 1.1.2 発信先番号の収集とterminationの初期化

上記に示した条件下において、以下に示す信号の送受信が行われる。MGCと発呼側MG(MG1)が発呼側のtermination(A4444)を取得し、発信先番号の収集を行う段階までのトランザクションを下記に示す。これらのトランザクションは着呼側MG(MG2)への接続を行うために必要である。

6. MG1がUser 1からのオフフックイベントを検知し、NotifyコマンドでMGCに通知する。

```
MG1 to MGC:
MEGACO/3 [124. 124. 124. 222]: 55555
Transaction = 10000 {
    Context = - {
        Notify = A4444 {ObservedEvents =2222 {
            19990729T22000000: al/of(i n i t=OFF)}}
    }
}
```

7. Notifyが処理される。

```
MGC to MG1:
MEGACO/3 [123. 123. 123. 4]: 55555
Reply = 10000 {
    Context = - {Notify = A4444}
}
```

8. MGCはterminationに対してModifyを行い、ダイヤルトーンを送出し、Dialplan0に従うdigitの発信を検知し、オンフックイベントを検出するよう指示する。

```
MGC to MG1:
```

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 10001 {
  Context = - {
    Modify = A4444 {
      Events = 2223 {
        al/on(strict=state), dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
(O|00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxx|9011x.)}
      }
    }
  }
}

```

9. Modifyが処理される。

```

MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10001 {
  Context = - {Modify = A4444}
}

```

10. 次に、User 1によってダイヤルされたdigitがMG1に収集される。ダイヤルトーンは最初のdigitを検出した時点で停止される。A4444に現在指定されているDialplanと収集されたdigitが一致した時点で、別のNotifyがMGCに対して送出される。

```

MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10002 {
  Context = - {
    Notify = A4444 {ObservedEvents =2223 {
      19990729T22010001: dd/ce{ds="916135551212",Meth=UM}}
    }
  }
}

```

11. Notifyが処理される。

```

MGC to MG1:

```

```
MEGACO/3 [123. 123. 123. 4]: 55555
Reply = 10002 {
    Context = - {Notify = A4444}
}
```

12. MGCはdigitを解析し、MG1からMG2へ接続を張る必要があると判断する。MG1に新しいcontextを生成し、TDM termination A4444とRTP TerminationをAddする。Remoteディスクリプタの値がまだ特定されていないので、モードはRecvOnlyである。コーデックの選択肢はMGCが希望する優先順に並んでいる。

MGC to MG1:

```
MEGACO/3 [123. 123. 123. 4]: 55555
Context = $ {
    Add = $ {
        Media {
            Stream = 1 {
                Local Control {
                    Mode = RecvOnly,
                    nt/jit=40 ; ミリ秒単位
                },
                Local {
                    v=0
                    c=IN IP4 $
                    m=audio $ RTP/AVP 4
                    v=0
                    m=audio $ RTP/AVP 0
                }
            }
        }
    }
}
```

注記： MGCはLocalに設定したいパラメータをSDPブロック列で指定する。MGはReplyで、LocalディスクリプタをMGが実際に設定した内容で埋めて返す。

13. MG1は新しいterminationを認識し、LocalのIPアドレスとUDPポートを設定する。MGは、MGCが指定したコーデックの中からLocalに設定するコーデックを選択する。MG1はRTPポートを2222に設定する。

```
MEGACO/3 [124. 124. 124. 222]: 55555
Reply = 10003 {
    Context = 2000 {
```

```

Add = A4444,
Add=A4445{
  Media {
    Stream = 1 {
      Local {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
S=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
a=recvonly
      } ; G.723.1のRTPプロファイルは4
    }
  }
}
}
}

```

14. MGCはMG2上の新しいcontextとA5555を関連付け、発呼側ユーザであるUser 1までのSendRecv接続のRTPストリームを確立する(RTP terminationにはA5556が割り当てられる)。MGCはA5555に呼出音(ring)を鳴らすよう設定する。

```

MGC to MG2:
MEGACO/3 [123.123.123.4]:55555
Transaction = 50003 {
  Context = $ {
    Add = A5555 { Media {
      Stream = 1 {
        Local Control {Mode = SendRecv} },
        Events=1234{al/of(strict=state)},
        Signals {al/ri}
      },
    Add = $ {Media {
      Stream = 1 {
        Local Control {
          Mode = SendRecv,
          nt/jit=40 ; ミリ秒単位

```

```

    },
    Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime: 30
    },
    Remote {
v=0
c=IN IP4 124. 124. 124. 222
m=audio 2222 RTP/AVP 4
a=ptime: 30
    } ; G. 723. 1のRTPプロファイルは4
    }
    }
    }
    }
}

```

15. MG2によって処理され、Replyが返される。ストリームのポート番号は制御ポート番号とは異なる。この場合は(SDP内に記述されているように)1111である。

```

MG2 to MGC:
MEGACO/3 [124. 124. 124. 222]: 55555
Reply = 50003 {
  Context = 5000 {
    Add = A5555,
    Add = A5556{
      Media {
        Stream = 1 {
          Local {
v=0
o=- 7736844526 7736842807 IN IP4 125. 125. 125. 111
S=-
t= 0 0
c=IN IP4 125. 125. 125. 111
m=audio 1111 RTP/AVP 4
          }
        } ; G. 723. 1のRTPプロファイルは4
      }
    }
  }
}

```



```

    }
  }
}
}

```

16. ここで上記のIPアドレスとUDPポートがMG1に与えられる必要がある。

```

MGC to MG1:
MEGACO/3 [123. 123. 123. 4]: 55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {cg/rt}
    },
    Modify = A4445 {
      Media {
        Stream = 1 {
          Remote {
v=0
o=- 7736844526 7736842807 IN IP4 125. 125. 125. 111
s=-
t= 0 0
c=IN IP4 125. 125. 125. 111
m=audio 1111 RTP/AVP 4
          }
        } ; G. 723. 1のRTPプロファイルは4
      }
    }
  }
}
}
}

```

```

MG1 to MGC:
MEGACO/3 [124. 124. 124. 222]: 55555
Reply = 10005 {
  Context = 2000 {Modify = A4444, Modify = A4445}
}

```

17. ここで二つのゲートウェイが接続され、User 1にはリングバックトーンが流れる。MG2はUser 2が受話器を上げるのを待ち、その後、両方向の呼が確立する。

From MG2 to MGC:  
MEGACO/3 [125.125.125.111]: 55555  
Transaction = 50005 {  
  Context = 5000 {  
    Notify = A5555 {ObservedEvents =1234 {  
      19990729T22020002: al/of(ini t=off)}}  
    }  
  }  
}

From MGC to MG2:

MEGACO/3 [123.123.123.4]: 55555  
Reply = 50005 {  
  Context = - {Notify = A5555}  
}

From MGC to MG2:

MEGACO/3 [123.123.123.4]: 55555  
Transaction = 50006 {  
  Context = 5000 {  
    Modify = A5555 {  
      Events = 1235 {al/on(strict=state)},  
      Signals ; リンギングを停止させるため  
    }  
  }  
}

From MG2 to MGC:

MEGACO/3 [125.125.125.111]: 55555  
Reply = 50006 {  
  Context = 5000 {Modify = A4445}  
}

18. MG1のモードがSendRecvにするとともに、リングバックトーンを停止する。

MGC to MG1:

MEGACO/3 [123. 123. 123. 4]: 55555

```
Transaction = 10006 {
  Context = 2000 {
    Modify = A4445 {
      Media {
        Stream = 1 {
          Local Control {
            Mode=SendRecv
          }
        }
      }
    },
    Modify = A4444 {
      Signals
    }
  }
}
```

from MG1 to MGC:

MEGACO/3 [124. 124. 124. 222]: 55555

```
Reply = 10006 {
  Context = 2000 {Modify = A4445, Modify = A4444}}
```

19. MGCはMG2上のRTP Terminationの監査を決定する。

MEGACO/1 [123. 123. 123. 4]: 55555

```
Transaction = 50007 {
  Context = - {AuditValue = A5556{
    Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
  }
}
```

20. MG2が応答を返す。

MEGACO/3 [125. 125. 125. 111]: 55555

```
Reply = 50007 {
  Context = 5000 {
    AuditValue = A5556 {
      Media {
```

```

TerminationState { ServiceStates = InService,
    Buffer = OFF },
Stream = 1 {
    LocalControl { Mode = SendRecv,
        nt/jit=40 },
    Local {
v=0
o=- 7736844526 7736842807 IN IP4 125.125.125.111
S=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
a=ptime:30
    },
    Remote {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
S=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
    } } },
Events,
Signals,
DigitMap,
Packages {nt-1, rtp-1},
Statistics { rtp/ps=1200, ; 送信パケット数
    nt/os=62300, ; 送信オクテット数
    rtp/pr=700, ; 受信パケット数
    nt/or=45100, ; 受信オクテット数
    rtp/pl=0.2, ; % パケット損失率
    rtp/jit=20,
    rtp/delay=40 } ; 平均遅延
    }
    }
}

```

21. MGCがいずれかのMGからオンフックシグナルを受け取った時点で、呼は終了する。この例ではMG2の

ユーザが先に切断している。

```
From MG2 to MGC:
MEGACO/3 [125.125.125.111]:55555
Transaction = 50008 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1235 {
      19990729T24020002: al/on(i ni t=off)}
    }
  }
}
```

```
From MGC to MG2:

MEGACO/3 [123.123.123.4]:55555
Reply = 50008 {
  Context = - {Notify = A5555}
}
```

22. ここでMGCは両MGに対してSubtractを送信し、呼の終了を指示する。以下の例ではMG2に対するSubtractだけを示す。各terminationは、それぞれ収集した統計情報を保持している。MGCは両方のTerminationに統計情報の返却を要求しなくてもよい。A5555は物理terminationであり、A5556はRTP terminationである。

```
From MGC to MG2:

MEGACO/3 [123.123.123.4]:55555
Transaction = 50009 {
  Context = 5000 {
    Subtract = A5555 {Audit{Statistics}},
    Subtract = A5556 {Audit{Statistics}}
  }
}
```

```
From MG2 to MGC:

MEGACO/3 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
```

```

Subtract = A5555 {
    Statistics {

        nt/os=45123, ; 送信オクテット数
        nt/or=45123, ; 受信オクテット数
        nt/dur=40000 ; ミリ秒単位
    }
},
Subtract = A5556 {
    Statistics {
        rtp/ps=1245, ; 送信パケット数
        nt/os=62345, ; 送信オクテット数
        rtp/pr=780, ; 受信パケット数
        nt/or=45123, ; 受信オクテット数
        rtp/pl=10, ; % パケット損失率
        rtp/jit=27,
        rtp/delay=48, ; 平均遅延
        nt/dur=38000 ; ミリ秒単位
    }
}
}
}
}

```

23. ここでMGCは、MG1とMG2に次のオフフックイベントを検出する設定を行う（Step1を参照）。なお、この設定が行われた状態はNULL Context中にあるterminationのデフォルト状態かもしれない。その場合、MGCからMGにメッセージを送る必要はない。TerminationはNULL Contextに戻されると、設定がそのterminationのデフォルト値に戻されるからである。

## 付録II H.248パッケージテンプレート

(本付録は本標準の必須部分ではない)

新しく規定されるか、更新されるH.248パッケージは、以下に示すテンプレートを用いて定義されるべきである。このテンプレートはITU勧告が採用する形式である。ITU以外の組織のエディターであっても、最低限、本付録5節に示す構造を用いて記述するべきである。H.248パッケージに関するキーワードはボールド体で記述している。記入すべき箇所は”<>”で示している。新しいパッケージの規定方法に関する詳細情報は12節/H.248.1を参照のこと。

注記：パッケージ特有のエラーコードはIANAへの登録が必要。エラーコードの付与は、4.1節/H.248.8「エラーコードの付与」の記載と同じ規則に従う。

# ITU-T RECOMMENDATION H.248.<XXX>

## Gateway Control Protocol: <xxx> Packages

### 1 Scope

<The scope of the package>

### 2 References

本勧告は、ここでの記述に従い、以下のITU-T勧告および他の参照文書の条項も含め、本勧告の条項を構成する。本勧告が刊行される時点においては、以下に示される全ての版は有効であった。すべての勧告および他の参照文書は改訂される可能性があるため、本勧告の利用者は、以下の勧告および他の参照文書について最新版の適用可能性を評価することが推奨される。有効なITU勧告の一覧は、定期的に発行されている。本勧告中に示される参照はあくまでも参照であり、参照している独立したドキュメント自体を規定するものでも、参照している勧告の状態を規定するものでもない。

- ITU-T Recommendation H.248.1 (<xx/xxxx>), *Gateway Control Protocol: Version< x>*
- <Other references>

### 3. Terms and Definitions

本勧告は、以下の用語と定義を用いる。:

<terms and definitions>

### 4. Abbreviations

本勧告は以下の略号を用いる。:

<abbreviations>

### 5. <Package Title>

**Package Name:** <name>

**PackageID:** <PackageID "text", binary <0x????>>

**Description:** <description>  
**Version:** <version>  
**Extends:** <extended PackageID and version>

## 5.1 Properties

### 5.1.1 <Property Title>

**Property Name:** <name>  
**PropertyID:** <text ID, binary ID (0x????)>  
**Description:** <description>  
**Type:** <type>  
**Possible values:** <values>  
**Default:** <value>  
**Defined in:** <Local, Remote, LocalControl, TerminationState, ContextAttribute>  
**Characteristics:** <ReadOnly, Read/Write>

## 5.2 Events

### 5.2.1 <Event Title>

**Event Name:** <event name>  
**EventID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>

#### 5.2.1.1 EventsDescriptor parameters:

##### 5.2.1.1.1 <Parameter Title>

**Parameter Name:** <name>  
**ParameterID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>  
**Type:** <types>  
**Optional:** <yes/no>  
**Possible values:** <values>  
**Default:** <value>

#### 5.2.1.2 ObservedEventsDescriptor parameters:

##### 5.2.1.2.1 <Parameter Title>

**Parameter Name:** <name>  
**ParameterID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>  
**Type:** <types>  
**Optional:** <yes/no>  
**Possible values:** <values>  
**Default:** <value>

## 5.3 Signals

### 5.3.1 <Signal Title>

**Signal Name:** <name>  
**SignalID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>  
**Signal Type:** <type>  
**Duration:** <duration>

#### 5.3.1.1 Additional parameters:

##### 5.3.1.1.1 <Parameter Title>

**Parameter Name:** <name>  
**ParameterID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>



**Type:** <types>  
**Optional:** <yes/no>  
**Possible values:** <values>  
**Default:** <value>

#### 5.4 Statistics

##### 5.4.1 <Statistic Title>

**Statistic Name:** <name>  
**Statistic ID:** <(text ID), (binary ID (0x????))>  
**Description:** <description>  
**Type:** <type>  
**Possible Values:** <values>  
**Level:** <Termination, Stream, Either>

#### 5.5 Error Codes

##### 5.5.1 <Error Code Title>

**Error Code #:** <number>  
**Name:** <name>  
**Definition:** <definition>  
**Error Text in the Error Descriptor:**  
<error text to return>  
**Comment:** <comment>

#### 5.6 Procedures

<The procedures associated with the package>