TTC技術レポート
Technical Report

ＴＲ－１００５

SIGTRAN 技術レポート

第 1 版

2002 年 2 月 20 日制定

社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

著作権事項

目　次

<要約>

１．技術レポート作成の経緯

本技術レポートは、回線交換網の呼制御信号プロトコルを IP 網上で転送するための IETF SIGTRAN 仕様を解説するとともに、TTC 標準準拠網へ適用する際の変更点を示す。

２．改版履歴

版数	発行日	改版内容
第 1 版	2002 年 2 月 20 日	制定

３．参照している勧告、標準など

TTC 標準: JT-Q701, JT-Q702, JT-Q703, JT-Q704, JT-Q711, JT-Q920, JT-Q921

ITU-T 勧告: Q.2210

４．TTC 標準準拠網へ適用するための変更内容

本文 1.5 節にて解説する。

１．はじめに

1.1 SIGTRAN

1.1.1 背景

Signaling Transport (SIGTRAN) は、回線交換網の呼制御信号を IP 網上で転送するための技術標準であり、Internet Engineering Task Force (IETF) が規定している。SIGTRAN 技術仕様は主に以下から構成される:

- フレームワークアーキテクチャ [RFC2719]
- 呼制御信号を転送するアダプテーションプロトコル
- アダプテーションプロトコルが使用するトランスポートプロトコル
- アダプテーションプロトコルとトランスポートプロトコルの管理情報ベース(MIB ： Management Information Base)

1.1.2 用語と定義

- IUA (Q.921-User Adaptation) : Q.921 ユーザプロトコルを転送するアダプテーションプロトコル。
- M2UA(SS7 MTP2-User Adaptation Layer) : MTP レベル 2 ユーザプロトコルを転送するアダプテーションプロトコル
- M3UA(SS7 MTP3-User Adaptation Layer): MTP レベル 3 ユーザプロトコルを転送するアダプテーションプロトコル
- M２PA(SS7 MTP2-User Peer-to-Peer Adaptation Layerer): MTP レベル 2 ユーザプロトコルをピアツーピアで転送するアダプテーションプロトコル
- SUA (SS7 SCCP-User Adaptation Layer) : SCCP ユーザプロトコルを転送するアダプテーションプロトコル
- SCTP (Stream Control Transfer Protocol) : 上記アダプテーションプロトコルが使用するトランスポートプロトコル (RFC2960)

1.2 著作権

本技術レポートは TTC の文書です。しなしながら、IETF ドキュメントからの引用個所については IETF の著作権の適用範囲のため、本節に IETF の著作権表示を掲載します。従って、本節に記述する IETF 著作権表示は TTC 技術レポート TD-1001 全体に関するものではありません。

1.2.1 IETF 著作権表示

以下は、上記 IETF 著作権表示を参考のために和訳したものです。

特定の目的における商品性や適合性を保証しない、といった事を保証しない。

1.3 技術レポートの構成

本技術レポートは以下のように構成される。

1 章 はじめに　　本技術レポートの位置付けおよび概要を説明する。

2 章 SCTP　　信号トラヒック転送のために開発された Stream Control Transmission Protocol (SCTP) と呼ぶトランスポートプロトコルの技術仕様を解説する。

3 章 UA 共通規定　信号トラヒック転送のためのユーザアダプテーション (UA：User Adaptation) プロトコルの共通仕様を解説する。

4 章 IUA　　Q.921 ユーザ部プロトコルのアダプテーションプロトコルの技術仕様を解説する。

5 章 M3UA　　MTP3 ユーザ部プロトコルのアダプテーションプロトコルの技術仕様を解説する。

6 章 M2UA　　MTP2 ユーザ部プロトコルのアダプテーションプロトコルの技術仕様を解説する。

7 章 M2PA　　MTP2 ユーザ部プロトコルのアダプテーションプロトコルの技術仕様を解説する。

8 章 SUA　　SCCP ユーザ部プロトコルのアダプテーションプロトコルの技術仕様を解説する。

9 章 おわりに

1.4 原仕様

本技術レポートは表 1-1 に示す原仕様に準拠する。

表 1-1 原仕様一覧

Framework Architecture for Signaling Transport	
ドキュメント名	RFC2719
種別	Informational RFC
発行日	1999 年 10 月
著者	L. Ong/Nortel Networks, etc.
概要	SIGTRAN のフレームワークを規定
Stream Control Transmission Protocol	
ドキュメント名	RFC2960
種別	Proposed Standard RFC
発行日	2000 年 10 月
著者	R. Stewart/Motorola, etc.
概要	SCTP プロトコルを規定
ISDN Q.921 User Adaptation Layer	
ドキュメント名	RFC3057
種別	Proposed Standard RFC
発行日	2001 年 2 月
著者	M. Krishnaswamy/Lucent Technologies, etc.
概要	IUA プロトコルを規定
SS7 MTP3-User Adaptation Layer	
ドキュメント名	draft-ietf-sigtran-m3ua-12.txt
種別	Internet Draft
発行日	2002 年２月

著者	Greg Sidebottom/greside consulting, etc.
概要	M3UA **プロトコルを規定**
SS7 MTP2-User Adaptation Layer	
ドキュメント名	draft-ietf-sigtran-m2ua-14.txt
種別	Internet Draft
発行日	2002 **年** 2 **月**
著者	Ken Morneault/Cisco Systems, etc.
概要	M2UA **プロトコルを規定**
SS7 MTP2-User Peer-to-Peer Adaptation Layer	
ドキュメント名	draft-ietf-sigtran-m2pa-03.txt
種別	Internet Draft
発行日	2001 **年** 7 **月**
著者	Tom George/Alcatel, etc.
概要	M2PA **プロトコルを規定**
SS7 SUA-User Adaptation Layer	
ドキュメント名	draft-ietf-sigtran-sua-12.txt
種別	Internet Draft
発行日	2002 **年** 2 **月**
著者	J.LoughneyNokia, etc.
概要	SUA **プロトコルを規定**

1.5 原仕様との差分

TTC **標準準拠網への適用に際し、**SIGTRAN **仕様の修正が必要となる。修正項目一覧を表** 1-2 **示す。**

表 1-2 修正項目一覧

修正箇所	修正内容	修正理由
5.3.6.1.3.1	到達不能信号局コードのフォーマット例に TTC 版 16 ビット信号局コードのフォーマット例を追加	原仕様では ANSI 版 24 ビット信号局コードと ITU-T 版 14 ビット信号局コードのフォーマット例のみが例示されており、国内ユーザの利便性向上のためには TTC 版 16 ビット信号局コードのフォーマット例を示す必要があると判断した。
6.4.7.1	データメッセージを使用対象外とした。	TTC 版 MTP2 は信号長表示オクテットの空き 2 ビットを優先度表示として使用するため、サービス情報オクテットと信号情報部のみを転送するデータメッセージは使用できない。代わりに信号長表示も転送する TTC データメッセージを使用する。
6.6.3.1	緊急リンク設定シーケンスの適用対象外化	TTC 版 MTP2 は緊急リンク設定手順をサポートしないため、同シーケンスを適用対象外とする。
6.6.3.3	プロセッサ障害シーケンスの適用対象外化	TTC 版 MTP2 はプロセッサ障害手順をサポートしないため、同シーケンスを適用対象外とする。
6.6.3.4	リンク輻輳シーケンスの適用対象外化	TTC 版 MTP2 はリンク輻輳手順をサポートしないため､同シーケンスを適用対象外とする。
6.6.3.6	バッファフラッシュおよび継続シーケンスの適用対象外化	TTC 版 MTP2 はバッファフラッシュおよび継続手順をサポートしないため、同シーケンスを適用対象外とする。
7.5.1.4	プロセッサ障害手順の適用対象外化	TTC 版 MTP2 はプロセッサ障害手順をサポートしないため、同手順を適用対象外とする。
7.5.1.6	リンク輻輳通知手順の適用対象外化	TTC 版 MTP2 はリンク輻輳通知手順をサポートしないため、同手順を適用対象外とする。
7.6.1.2	プロセッサ障害シーケンスの適用対象外化	TTC 版 MTP2 はプロセッサ障害手順をサポートしないため、同シーケンスを適用対象外とする。
7.6.1.4	リンク輻輳通知手順の適用対象外化	TTC 版 MTP2 はリンク輻輳通知手順をサポートしないため、同シーケンスを適用対象外とする。
8.1.4	ＳＣＣＰコネクションオリエンティッドサービスを規定対象外化	TTC 版 SCCP はリンク輻輳通知手順をサポートしないため、同サービスを適用対象外とする。

TTC 版 MTP2 は全メッセージ回収プリミティブを独自に規定しているが、M2UA および M2PA はサポートしていない。SIGTRAN における本プリミティブ実現要否および実現方法が今後の課題である。

2．SCTP

この文書は、ストリーム制御伝送プロトコル(SCTP)について述べている。SCTP は PSTN のシグナリングメッセージを、IP ネットワークを使って伝送するために設計されているが、これ以外に利用することもできる。

SCTP は信頼性を持った伝送プロトコルであり、IP のようなコネクションレスパケットネットワーク上で機能する。SCTP は上位アプリケーションに対して、以下のようなサービスを提供する。

- エラーフリーで複製を作る必要がない。
- 与えられたパスの MTU サイズに従うようにデータを分割する。
- マルチストリームを用いてメッセージを順番に伝達する。個々のメッセージに対して順番に到着する(order-of-arrival)オプションを指定して伝達する。
- 複数のメッセージを 1 つの SCTP パケットにバンドルするオプションを持つ。
- ネットワークレベルの耐障害性を、片方もしくは両方のアソシエーションエンドポイントのマルチホーミングによって実現する。

SCTP の設計には、適切な輻輳回避や flooding とマスカレードアタックに対する防御が含まれている。

2.1 序論

本節では、SCTP 設計に至った背景、提供するサービス、詳細なプロトコルの記述内容を理解するために必要な基本コンセプトについて述べる。

2.1.1 動機

TCP[RFC793]は、IP ネットワーク上で信頼性を持ったデータ転送を行う手段としては非常によく機能してきた。しかし、最近のアプリケーションでは、TCP は制限が多すぎるため、UDP 上でアプリケーション独自の信頼性を持ったデータ転送プロトコルを持つ場合が増えてきている。バイパスの要求が高まっているＴＣＰの制限とは以下のようなことである:

- TCP は信頼性を持ったデータ転送と転送順序保証機能を有している。一部のアプリケーションはデータ転送の信頼性は必要としているが、転送順序保証機能は必要としない。また、別のアプリケーションは部分的な順序保証で充分である。これらの場合、TCP が提供している head-of-line ブロッキングの機能は不必要な遅延を引き起こしてしまう。
- TCP のストリームオリエンテッドの性質は、不都合が生じる場合がある。アプリケーションは適切な時間内でのメッセージ全体の伝送を保証するために、メッセージ毎にマークを付与し、それらを強制的(explicit)に配送(push)を行う機能を有していなくてはならない。
- TCP ソケットの機能的な制限は、マルチホームホストを利用した高い可用度を持ったデータ転送を提供するための機構を複雑にする。
- TCP は比較的 SYN アタックのような、ＤｏＳ(denial-of-service)攻撃に対して脆弱である。

IP ネットワークを用いた PSTN シグナルの伝送は、これら全ての TCP の制限に関係があるアプリケーションである。このアプリケーションが直接的な SCTP 設計の動機であるが、他のアプリケーションでも、SCTP がその要求条件によく適するものもあるかもしれない。

2.1.2 SCTP の構図

SCTP は上位アプリケーション(「SCTP ユーザ」と簡易表記される)と(IP のような)コネクションレスパケットネットワークの間に位置する。以後、この文書では SCTP は IP 上で動作していると想定して話しを進める。SCTP によって提供される基本的なサービスは、SCTP ユーザ間の信頼性を有したデータ転送である。これは、SCTP エンドポイント間のアソシエーションのコンテキスト上で実現される。セクション 10 には、SCTP

と SCTP ユーザの間に存在する API のスケッチが示されている。

SCTP はコネクション指向を有しているが、SCTP アソシエーションは TCP コネクションより広いコンセプトを持っている。SCTP は各 SCTP エンドポイントに(アソシエーションのスタートアップ時に)他のエンドポイントを知る手段を与える(セクション 1.4)。そのエンドポイントで生成した SCTP パケットを到達させることができるトランスポートアドレスのリスト(即ち、SCTP ポートに対比させた複数の IP アドレス)を提供することによって。アソシエーションは、各エンドポイントのリストから生成された、全ての可能な送信元/送信先対応表を伝送する。

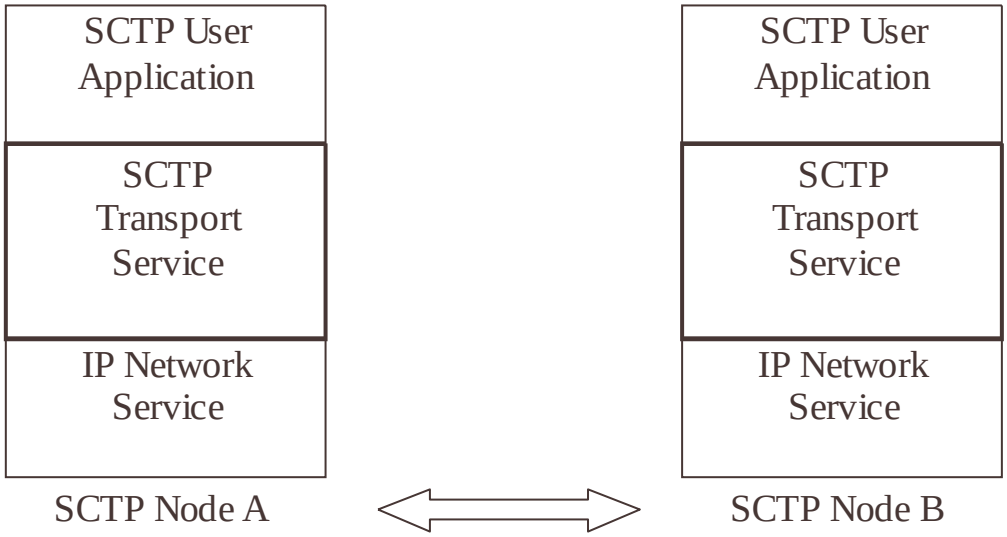


図 2-1 SCTP アソシエーション

2.1.3 SCTP の機能

SCTP の伝送機能は幾つかの機能に分割できる。これを図 2-2 に示し、以後説明する。

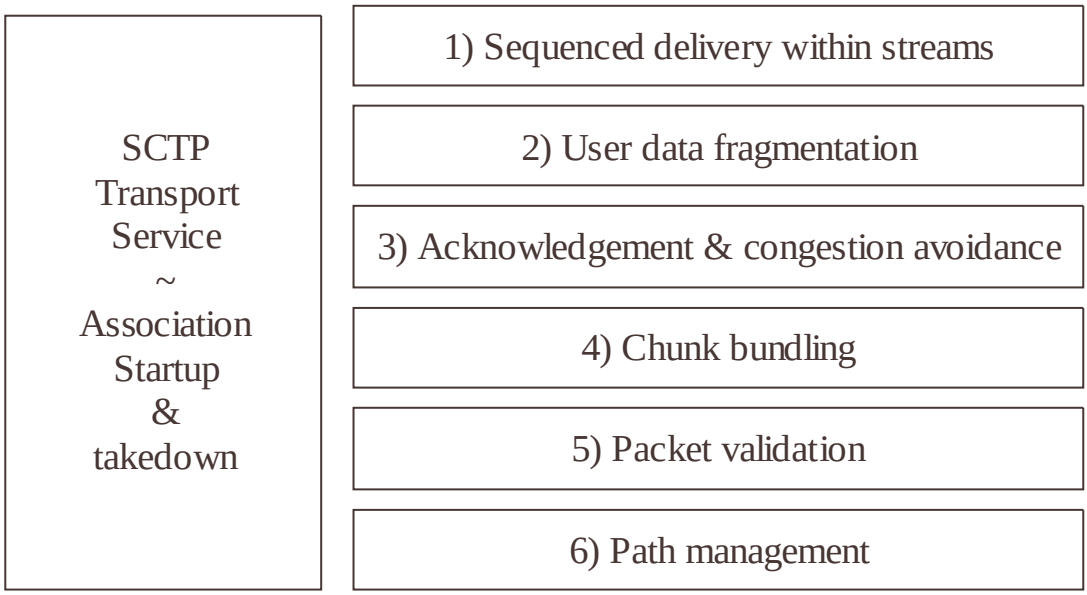


図 2-2 SCTP 伝送機能

2.1.3.1 アソシエーションの起動と終了

アソシエーションは SCTP ユーザによって開始される。(2.10 節の ASSOCIATE(or SEND)プリミティブの記述を参照のこと)

クッキーメカニズム（RFC2522 において Karn と Simpson によって記述されているものと同様のメカニズム）は、セキュリティ攻撃に対する保護機能の初期化の際に使われる。クッキーメカニズムは 4 本のハンドシェイクを使い、最後の 2 本は早いユーザデータ転送のために使われる。スタートアップシーケンスはこのドキュメントの 2．5 節に記述されている。

SCTP は SCTP ユーザからの要求によりアクティブアソシエーションを正常終了 (シャットダウン) する。2.10 節のシャットダウンプリミティブを参照のこと。また、SCTP はユーザからの要求(アボートプリミティブ)または、SCTP レイヤ間のエラー発生により異常終了（アボート）することもある。2.9 節に正常終了と異常終了の両方が記述されている。

SCTP は、片側がデータを送り続けもう一方が閉じているような(TCP のような)ハーフオープンな状態をサポートしていない。どちらかのエンドポイントがシャットダウンを実行する時、各ピアのアソシエーションはそのユーザからのデータ受け入れを止め、正常終了により待ち行列にあるデータの転送をするだけであろう。

2.1.3.2 ストリーム内の順序を保持した配送

SCTP において『ストリーム』はユーザメッセージ列を意味する。同一ストリーム内部ではユーザメッセージの順序性は保証される。これは、バイトの順序(このドキュメントでは 1 バイトは 8 ビットとする)を示す TCP の用法とは対照的である。

SCTP ユーザはアソシエーション起動時においてそのアソシエーションによりサポートされるストリームの数を特定できる。この数は、リモートエンドとネゴシエーションされる(2.5.1.1 節参照のこと)。ユーザメ

ッセージはストリームの数(SEND, RECEIVE プリミティブ、2.10 節)と関連付けされる。内部的には SCTP は SCTP ユーザにより各メッセージに対するストリームシーケンス番号を割当てる。受信者は SCTP が与えられたストリームの内で順番に SCTP ユーザにメッセージが届く事を確実にしている。しかしながら、1 つのストリームが次の順番のユーザメッセージを待っている間、ブロックされ、他のストリームが進行する。

SCTP はシーケンスデリバリーサービスをバイパスするメカニズムを提供する。このメカニズムを使い送信されるメッセージは、それが受信されると同時に SCTP ユーザに送信される。

2.1.3.3 ユーザデータの分割

必要であれば SCTP はユーザメッセージを分割し、下位レイヤに渡す　SCTP パケットがパス MTU に収まるようにする。受信者は、分割されたメッセージを SCTP ユーザに引き渡される前の完全なメッセージに組立て直す。

2.1.3.4 通知と輻輳回避

SCTP は分割もしくは分割されていない各ユーザデータメッセージに伝送順序番号(TSN)を付与する。TSN はストリームレベルで割当てられるどのストリームシーケンス番号とも独立している。受信者エンドは順序番号に差分があっても受信した全ての TSN を承認する。このように確実な送信は機能的にシーケンスストリームデリバリーとは分かれている。

適切な承認が受信されない場合は、承認と輻輳回避機能がパケット再伝達に責任を負っている。パケット再送は TCP で用いられるものと同様の輻輳回避手順によって条件が決まる。この機能に関連するプロトコル手順記述詳細については 2.6 節と 2.7 節を参照のこと。

2.1.3.5 チャンクのバンドル

2.3 節に記述されているように下位レイヤに引き渡される SCTP パケットは共通のヘッダから構成され、1 つまたは複数のチャンクがその後に続く。各チャンクはユーザデータ若しくは、SCTP 制御情報を持つ。SCTP ユーザは単一の SCTP パケットの中に 1 つ以上のユーザメッセージをバンドルすることを要求するオプションを持っている。SCTP のチャンクバンドリング機能は SCTP パケットの分解、受信者ではその組立てに対する役割を果たす。

ユーザが SCTP に対してバンドルを要求しなくても、輻輳発生時には SCTP インプリメンテーションではバンドルを実行する場合がある。ユーザがバンドルをしない設定にした場合は、(バンドルを試行するため)伝送前にわずかな遅延を生じる場合がある。ユーザレイヤでバンドルをしない設定にした場合は、このわずか遅延は防げるがバンドルしないことにより輻輳時には再送が発生する。

2.1.3.6 パケットバリデーション

必須ベリフィケーションタグフィールドと 32 ビットチェックサムフィールド(Appendix B の Alder32 チェックサムを参照のこと)は SCTP 共通ヘッダに含まれる。ベリフィケーションタグの値はアソシエーション開始時に各エンドで選択される。想定外のベリフィケーションタグを受信した場合、パケットは破棄され、マスカレード攻撃や以前のアソシエーションの古いパケットからの防御の役割を果たす。Alder32 チェックサムはネットワークでのデータ破壊に対する付加的な防御策として各 SCTP の送信側にて設定することを推奨する。受信者で無効な Alder32 チェックサムを含む SCTP パケットを受信した場合は、黙ってそれを破棄する。

2.1.3.7 パス管理

2.10 節で後述するように、SCTP 送信側ユーザがプリミティブを通じて SCTP パケットの目的地としての

トランスポートアドレスを設定することができる。SCTP パス管理機能は SCTP ユーザの指示と現在到達可能で適切と考えられる目的地の状態に基づいて各送信 SCTP パケットの送信先トランスポートアドレスを選択する。パス管理機能は他のパケットトラフィックがこの情報を的確に提供できない時、ハートビートを通じて到達性を監視し、ファーエンドのトランスポートアドレスへの到達性に変更が生じた時、SCTP ユーザにそれを通知する。アソシエーション起動時、各 SCTP エンドポイントでプライマリーパスを定義し、それが SCTP パケットの通常送信の際に使用される。

受信者では、パス管理機能は、さらなる処理が実行される前に入 SCTP パケットの所属する有効な SCTP アソシエーションの存在を検証する。

注釈) パス管理とパケットバリデーションは上記のように分けて記述されているが、実際には別々に実行されることはなく、同時に実行される。

2.1.4 用語の説明

SCTP を表現するいくつか用語は以前の節で紹介されている。ここでは主要な用語とその定義についてまとめた。

アクティブ送信先トランスポートアドレス:
送信側エンドポイントがユーザメッセージ受信可能と考える相手側エンドポイントのトランスポートアドレス。

バンドリング:
同じ SCTP パケットの中で 1 つ以上のユーザメッセージを送信するような付加的は多重オペレーション。各ユーザメッセージは自身の DATA チャンクを占有する。

チャンク:
SCTP パケット内の情報単位でチャンクヘッダとチャンク独自の内容を持つ。

cwnd (Congestion window):
承認を受信する前に送信側で特定の送信先トランスポートアドレスへ送信可能なバイト数で表わされる可変のデータ数の上限。

累積 TSN Ack **ポイント**:
SACK の累計 TSN Ack フィールドを経由して承認される最後の DATA チャンクの TSN。

アイドル送信先アドレス:
ある一定時間、通常はハートビートインターバルかそれより長い時間、以内にユーザメッセージを送出することがないアドレス。

インアクティブ送信先アドレス:
エラーもしくは、ユーザメッセージ伝達不可によりインアクティブと考えられるアドレス。

メッセージ （= ユーザメッセージ）:
上位レイヤプロトコル(ULP)により SCTP に伝えられるデータ。

MAC (Message Authentication Code):
秘密鍵を用いた暗号ハッシュ関数に基づいた安全チェックメカニズム。通常、メッセージ承認コードはこれら 2 者間で転送された情報を承認するために秘密鍵が共有される。SCTP では、相手方のクッキーエコーチャンクから返答される状態クッキー情報を承認するエンドポイントにより使用される。「MAC」という言葉は他の文脈では別の意味を持つ。SCTP でこの用語は RFC2104 の用法と同じ意味で使用される。

ネットワークバイトオーダー:
最初の最も重要なバイト、別名ビッグエンディアンと呼ばれる。

順序型メッセージ:

メッセージ送信に使われるストリーム内で順番通り送出されるユーザメッセージ。

未処理 TSN (at SCTP endpoint):

エンドポイントにより送出されたが、まだ承認を受信していない TSN(とそれに関連する DATA チャンク)。

パス:

ある SCTP エンドポイントから相手側の SCTP エンドポイントの特定の送信先トランスポートアドレスに送出される SCTP パケットのルート。異なる送信先トランスポートアドレスへパケットを送信する場合に、必ずしも別のパスを担保する必要はない。

プライマリーパス:

プライマリーパスはデフォルトで相手側エンドポイントに送出されるパケットの送信先アドレスとソースアドレス。インプリメンテーションが送信先アドレスとソースアドレスを特定する可能性があるのでこの定義はソースアドレスを含み、返答チャンクのためのリターンパスとパケットが転送されるインタフェースをコントロールしやすくする場合もある。

rwnd (Receiver Window):

送信側がストアできるもっとも最近に計算された相手側の SCTP の可変受信バッファデータ(バイト数)。これは送信者に対し、受信者で可能なバッファサイズを示す。

SCTP アソシエーション:

SCTP エンドポイント間のプロトコル関係で 2 つの SCTP エンドポイントとプロトコル状態情報から成り、ベリフィケーションタグと今アクティブな TSN 等を含む。アソシエーションはエンドポイントで使用されるトランスポートアドレスによりユニークに識別される。ふたつの SCTP エンドポイントは、同時に複数のアソシエーションを持つことはない。

SCTP エンドポイント:

SCTP パケットの論理的な送信者、受信者。マルチホームホストでは、SCTP エンドポイントは適切な送信先トランスポートアドレスの組み合わせとして相手側に示され、その送信先トランスポートアドレスに SCTP パケットは送受信される。SCTP エンドポイントにより使用される全てのトランスポートアドレスは同じポート番号を使用し、複数の IP アドレスを使用可能である。ある SCTP エンドポイントにより使用されるトランスポートアドレスが他の SCTP エンドポイントに使用されることはない。言い換えると、トランスポートアドレスは SCTP エンドポイントに対してユニークである。

SCTP パケット:

SCTP とコネクションレスパケットネットワーク(IP 等)の間のインタフェースを行き来するデータ単位。SCTP パケットは共通 SCTP ヘッダを含み、SCTP コントロールチャンクと SCTPDATA チャンクにカプセル化されたユーザデータが含まれる。

SCTP ユーザアプリケーション（SCTP ユーザ）:

SCTP サービスを使用する論理高位レイヤアプリケーションのことで、上位レイヤプロトコルとも呼ばれる。

ssthresh (Slow Start Threshold):

SCTP 可変値。これはエンドポイントがスロースタートや輻輳回避を特定の送信先トランスポートアドレスに対して使用する敷居値。バイト数で数える。

ストリーム:

関連する SCTP エンドポイントの間で確立される片方向の論理チャネル。その中で順序性を必要としないサービスを除いては、全てのユーザメッセージは順序通り転送される。

注釈) 逆方向のストリーム番号の関係はどのようにアプリケーションが使うかに依存する。これら相関関係は必要であれば SCTP ユーザにより作られ管理される。

ストリームシーケンス番号:

同じストリーム内で SCTP がユーザメッセージの順序転送を保証するために内部使用される 16 ビットシーケンス番号。

Tie タグ:

前回のアソシエーションから引き継がれたベリフィケーションタグ。これらのタグは、再起動されたアソシエーションが、再起動していないエンドポイント内の元のアソシエーションとリンクできるよう、状態クッキー内で使用される。

TCB (Transmission Control Block):

SCTP エンドポイントによって作られる内部データ構造。TCB はエンドポイントが関連するアソシエーションを維持管理するための全ての状態と運用情報を持つ。

TSN (Transmission Sequence Number):

SCTP で内部的に使用される 32 ビットシーケンス番号。受信側 SCTP エンドポイントがデータの受信と二重受信を検知するために、ユーザデータを持つチャンクには、それぞれ、ひとつづつ、TSN がつけられる。

トランスポートアドレス:

トランスポートアドレスは、通常はネットワークレイヤアドレス、トランスポートプロトコル及びトランスポートレイヤポート番号により定義される。IP 上で働く SCTP の場合は、トランスポートアドレスは、IP アドレスと SCTP ポート番号(SCTP はトランスポートプロトコル)の組み合わせにより定義される。

(SCTP エンドポイントにおける) 未承認 TSN:

エンドポイントにより受信されたが、承認が返答されていない TSN(および、関連する DATA チャンク)。あるいは逆に、パケットは送信されたが、承認が受信されていない TSN(および、関連する DATA チャンク)。

非順序型メッセージ:

どんなメッセージに関しても順序性がないメッセージ。順序性のないメッセージは、同じストリーム内で順序性のあるメッセージの前か後に転送される。

ユーザメッセージ:

SCTP とそのユーザ間のインタフェースで転送されるデータ単位。

ベリフィケーションタグ（Verification Tag）:

ランダムに生成される 32 ビットの符号なし整数。ベリフィケーションタグは、受信者で SCTP パケットにそのパケットが現在のアソシエーションに属しており、前のアソシエーションの古いパケットではないことを検証することを許容する鍵を提供する。

2.1.5 省略表記

MAC: Message Authentication Code [RFC2104]

RTO: Retransmission Time-out

RTT: Round-trip TimeRTTVAR: Round-trip Time Variation

SCTP: Stream Control Transmission Protocol

SRTT: Smoothed RTT

TCB: Transmission Control Block

TLV: Type-Length-Value Coding Format

TSN: Transmission Sequence Number

ULP: Upper-layer Protocol

2.1.6 シリアル番号の演算

実際の TSN は、非常に大きいが、有限であることは覚えておかなくてはならない。この値の範囲は、0 から $2^{32}-1$ である。この空間は有限であり、TSN として処理される全ての演算は、モジュロ 2^{32} で実行される。この正数の演算は、$2^{32}-1$ の次はから 0 に戻ることで、シリアル番号の関係を保つ。コンピュータのモジュロ演算としてのいくつかの注意点があるため、このような数値を比較するプログラムでは、細心の注意を払わなければならない。TSN に関して言及される時、“ =< ”のシンボルの意味は、(モジュロ 2^{32} で) “ 未満または等しい ”を意味する。

本文書中の TSN に関する比較及び演算は、［RFC1982］で定義されているシリアル番号演算(SERIAL_BITS=32)を使用することが推奨される。

TSN の比較に問題が生じる可能性があるため、エンドポイントは現在の送信ウインドウにおける最初の TSN の値よりも $2^{31}-1$ 以上大きい TSN の値を持つ DATA チャンクを転送しないことが推奨される。、

TSN は $2^{32}-1$ に到達すると一巡する。即ち、TSN＝$2^{32}-1$ を送信した後の次の DATA チャンクの TSN は 0 となる。

ストリームシーケンス番号の演算は、[RFC1982]で定義されているシリアル番号演算(SERIAL_BITS=16)を使用することが推奨される。

本文書に記載されている他の全ての計算及び比較は、通常の演算を使用する。

2.2 表記の取り決め(Conventions)

省略

2.3 SCTP のパケットフォーマット

1 つの SCTP パケットは 1 つの共通ヘッダと複数のチャンクからなる。チャンクは制御情報かユーザデータのどちらかを含む。

SCTP パケットのフォーマットを以下に示す。

1 つの SCTP パケットには MTU サイズの許す範囲で複数のチャンクをバンドルすることができる。ただし、INIT, INIT ACK, SHUTDOWN COMPLETE チャンクは他のチャンクと一緒にバンドルしてはいけない。チャンクのバンドリングの詳細に関しては 2.6.10 節を参照のこと。

ユーザデータが 1 つの SCTP パケットに収まらない場合は、2.6.9 節に示す手順を用いて複数のチャンクに分割される。

SCTP パケット内の全ての整数型(integer)フィールドは特に指定されない限り、ネットワークバイトオーダで送信される。

2.3.1 SCTP 共通ヘッダ内のフィールド説明

送信元ポート番号: 16 **ビット** (**符号無し整数**)

送信者の SCTP **ポート番号。受信者は、本フィールドと、送信元** IP **アドレス、**SCTP **送信先ポート番号、送信先** IP **アドレスを組み合わせることによって、どのアソシエーションに本パケットが属するかを特定するために使用する。**

送信先ポート番号: 16 **ビット** (**符号無し整数**)

パケットが送信される先の SCTP **ポート番号。受信者ホストはこのポート番号を、正しいエンドポイントもしくはアプリケーションに** SCTP **パケットを振分ける**(de-multiplex)**ために使用する。**

ベリフィケーションタグ: 32 **ビット** (**符号無し整数**)

パケットの受信者はベリフィケーションタグをパケットの送信者を確認する(Validate)**ために使用する。送信時には以下に挙げる例外を除いて、アソシエーションの開始**(initialization)**中に相手側より受信した開始タグの値が設定される。**

- INIT **チャンクを含んでいるパケットはベリフィケーションタグにフィールド値** 0 **を用いる。**
- T **ビットが設定された** SHUTDOWN COMPLETE **チャンクを含んでいるパケットは** SHUTDOWN ACK **チャンクを含んだパケットのベリフィケーションタグの値を用いる。**
- ABORT **チャンクを含んでいるパケットは、**ABORT **が送られる原因となったパケットのベリフィケーションタグの値を使ってもよい。詳細については** 2.8.4 **節および** 2.8.5 **節を参照のこと。**

チェックサム: 32 **ビット** (**符号無し整数**)

本フィールドは当該 SCTP **パケットのチェックサムを含んでいる。チェックサムの計算法については** 2.6.8 **節で説明されている。**SCTP **はチェックサムの算出に、**Appendix B **に示されている** Adler-32 **アルゴリズムを用いる。**

2.3.2 **チャンクフィールドの説明**

下図に SCTP **パケットで転送されるチャンクのフィールドフォーマットを示す。それぞれのチャンクは、チャンクタイプフィールド，チャンク特有のフラグフィールド，チャンク長フィールドおよびチャンク値フィールドの各フィールドからなる。**

チャンクタイプ: 8 **ビット (符号無し整数)**

本フィールドはチャンク値フィールド内に含まれる情報の種類を示す。フィールドは 0 **から** 254 **までの値を取る。フィールド値** 255 **は将来の利用のために拡張フィールドとして予約されている。**

チャンクタイプフィールドの値は次のように定義されている。

ID	**チャンクタイプ**
0	**ペイロードデータ** (DATA)
1	Initialization (INIT)
2	**開始確認**(INIT ACK)
3	**選択的確認**(SACK)
4	**ハートビート要求** (HEARTBEAT)
5	**ハートビート承認** (HEARTBEAT ACK)
6	**アソシエーション中断**(ABORT)
7	**アソシエーション停止** (SHUTDOWN)
8	**シャットダウン承認** (SHUTDOWN ACK)
9	**オペレーションエラー** (ERROR)
10	**状態クッキー** (COOKIE ECHO)
11	**クッキー承認** (COOKIE ACK)
12	Reserved for Explicit Congestion Notification Echo (ECNE)
13	**Reserved for Congestion Window Reduced (CWR)**
14	**シャットダウン完了** (SHUTDOWN COMPLETE)
15 to 62	Reserved by IETF
63	IETF-defined Chunk Extensions
64 to 126	Reserved by IETF
127	IETF-defined Chunk Extensions
128 to 190	Reserved by IETF
191	IETF-defined Chunk Extensions
192 to 254	Reserved by IETF
255	IETF-defined Chunk Extensions

チャンクタイプフィールドの値は、処理を行うエンドポイントがチャンクタイプを認識できない場合に取らなくてはならない動作を上位 2 ビットで特定できるようにコーディングされる。

上位 2 ビット	解釈
00	SCTP パケットの処理を中断して破棄するとともに、同パケット内の残りのチャンクに対する処理も行わない。
01	SCTP パケットの処理を中断して破棄するとともに、同パケット内の残りのチャンクに対する処理も行わず、かつ、認識できないパラメタを'Unrecognized Parameter Type'で通知する(ERROR もしくは INIT ACK による)。
10	このチャンクを飛ばして処理を継続する。
11	このチャンクを飛ばして処理を継続するが、ERROR チャンクで'Unrecognized Parameter Type'によりエラーの原因を通知する。

注: ECNE および CWR は将来の明示的な輻輳通知(Explicit Congestion Notification: ECN)のために予約されている。

チャンクフラグ: 8 ビット

本ビットの使い方はチャンクタイプフィールドによって指定されるチャンクの種類による。特に規定されなければ送信時には 0 が設定され、受信者では無視される。

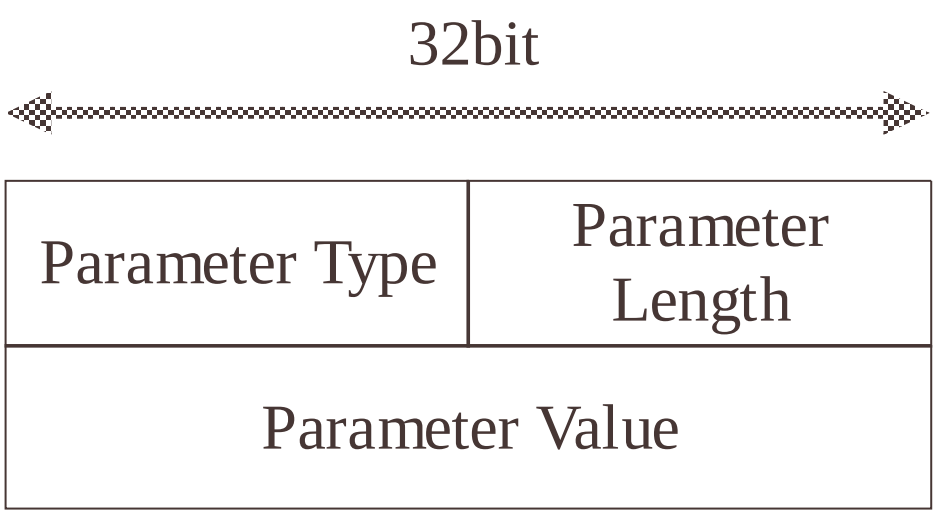
チャンク長: 16 ビット (符号無し整数)

本フィールドの値はチャンクタイプフィールド, チャンクフラグ・フィールド, チャンク長フィールド, チャンク値フィールドを含めたチャンクのサイズをバイト数で示す。したがって、チャンク値の長さが 0 であれば、本フィールドは 4 となる。本フィールドの値には padding は含めない。

チャンク値: 可変長

本フィールドにはチャンクによって転送される実際の情報が含まれる。本フィールドの利用方法とフォーマットはチャンクタイプによる。

チャンク全体の長さ(チャンクタイプ、チャンク長、チャンク値フィールドを含む)は 4 バイトの倍数でなくてはならない。チャンクの長さが 4 バイトの倍数でない場合、送信者は All 0 のバイト(Padding)で埋める。Padding の長さはチャンク長に含めない。送信者は 3 バイトを超える padding をするべきではない。受信者は padding されたバイトを無視する。



2.3.2.1 **オプション/可変長パラメタのフォーマット**

SCTP **制御チャンクのチャンク値は必須フィールドからなるチャンクタイプ特有のヘッダ**(chunk-type-specific header)**と、それに続く** 0 **個以上のパラメタから構成される。チャンクに含まれるオプション/可変長パラメタは以下に示す** Type-Length-Value **フォーマットで定義される。**

パラメタタイプ: 16 **ビット (符号無し整数)**

タイプフィールドはパラメタ種別を識別する 16 **ビットの整数であり、**0 **から** 65534 **までの値を取る。**

フィールド値 65535 **は** IETF **で定義される拡張**(IETF-defined extensions)**のために予約されている。(本ドキュメントの)各々の** SCTP **チャンクに関する説明の中で定義された値以外は** IETF **によって予約されている。**

パラメタ長: 16 **ビット (符号無し整数)**

パラメタ長フィールドは、パラメタタイプフィールド、パラメタ長フィールド、パラメタ値フィールドまで含めたパラメタのサイズをバイト数で示す。したがって、パラメタ値の長さが 0 **であれば、本フィールドは** 4 **となる。本フィールドの値には** padding **は含めない。**

パラメタ値: **可変長**

パラメタ値フィールドは、パラメタによって転送される実際の情報が含まれる。

パラメタ全体の長さ(パラメタタイプ、パラメタ長、パラメタ値フィールドを含む)は 4 **バイトの倍数でなくてはならない。パラメタの長さが** 4 **の倍数でない場合、送信者は** All 0 **のバイト**(Padding)**でパラメタを最後まで(すなわちパラメタ値フィールドの後を)埋める。**Padding **の長さはパラメタ長フィールドの値に含めない。送信者は** 3 **バイトを超える** padding **をするべきではない。受信者は** padding **されたバイトを無視する。**

パラメタタイプフィールドの値は、処理を行うエンドポイントがパラメタタイプを認識できない場合に取らなくてはならない動作を上位 2 **ビットで特定できるようにコーディングされる。**

上位 2 **ビット**	**解釈**
00	SCTP **パケットの処理を中断して破棄するとともに、同パケット内の残りのチャンクに対する処理も行わない。**
01	SCTP **パケットの処理を中断して破棄するとともに、同パケット内の残りのチャンクに対する処理も行わず、かつ、認識できないパラメタを**'Unrecognized Parameter Type'**で通知する**(ERROR **もしくは** INIT ACK **による)。**
10	**このパラメタを飛ばして処理を継続する。**
11	**このパラメタを飛ばして処理を継続するが、認識できないパラメタを**'Unrecognized Parameter Type'**で通知する**(ERROR **もしくは** INIT ACK **による)。**

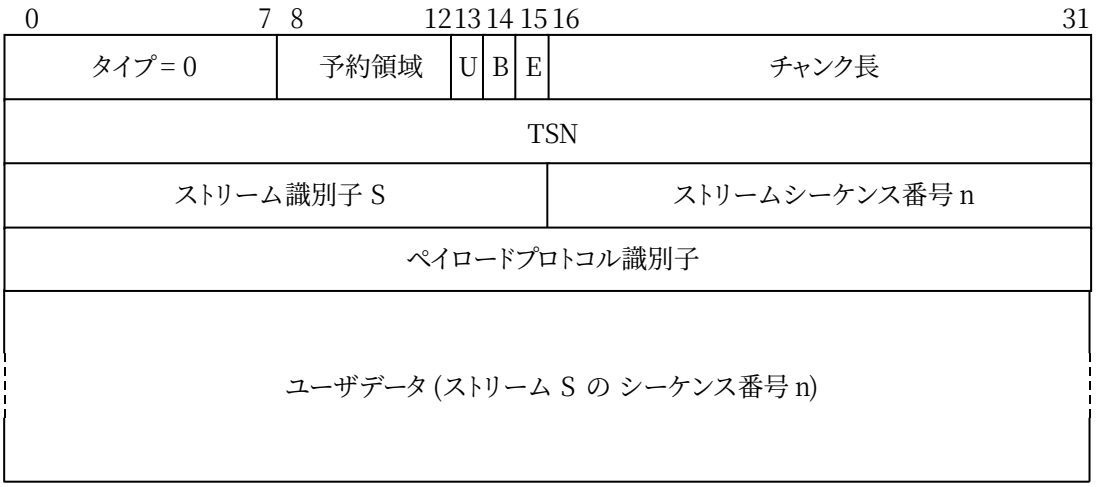
実際の SCTP **パラメタは各々の** SCTP **チャンクの節**[**訳注**:2.3.3 **節**]**で定義される。**IETF **で定義される拡張**(IETF-defined extensions)**の規則は** 2.13.2 **節で定義される。**

2.3.3 SCTP チャンクの定義

本節ではそれぞれのチャンク種別のフォーマットを定義する。

2.3.3.1 ペイロードデータ (DATA) (0)

DATA チャンクは以下のフォーマットに従う。



予約領域: 5 ビット

0 に設定され、受信者は無視する。

U ビット: 1 ビット

U ビット(Unordered bit)は, '1'が設定された場合は、非順序型(unordered) DATA チャンクであり、 ストリームシーケンス番号 (SSN)は割当てられていないことを示す。したがって、受信者は SSN を無視する。

再組立て(re-assembly)された(必要な場合)後、非順序型 DATA チャンクは受信者によって並び替えなしに上位レイヤに渡されなくてはならない。

非順序型ユーザメッセージが分割される場合は、全ての分割されたチャンクの U ビットを 1 に設定する。

B ビット: 1 ビット

B ビット(Beginning fragment bit)が設定されている場合、ユーザメッセージの最初の部分(fragment)であることを示す。

E ビット: 1 ビット

E ビット(Ending fragment bit)が設定されている場合、ユーザメッセージの最後の部分(fragment)であることを示す

分割されていないユーザメッセージは B ビットと E ビットの両方を'1'に設定しなくてはならない。B ビットと E ビットの両方を'0'に設定した場合には、次表に示す通り、複数に分割されたユーザメッセージの途中(最初と最後以外)のものである。

B	E	解説
1	0	分割されたユーザメッセージの最初の部分
0	0	分割されたユーザメッセージの途中の部分
0	1	分割されたユーザメッセージの最後の部分
1	1	分割されていないメッセージ

ユーザメッセージが複数のチャンクに分割されたときには、受信者はメッセージの再組立て(reassembly)に TSN を使用する。分割されたユーザメッセージのそれぞれの部分(fragment)の TSN は厳密にシーケンシャルでなくてはならない。

チャンク長: 16 ビット (符号無し整数)

本フィールドはDATAチャンクのタイプフィールドの開始部からpaddingを含まないユーザデータの最後までの長さをバイト数で示す。 ユーザデータフィールドがない DATA チャンクのチャンク長は 16 になる。

TSN: 32 ビット (符号無し整数)

本フィールドは DATA チャンクのための TSN を表す。TSN の有効な値の範囲は 0 から 4294967295 (2 の 32 乗-1)である。TSN が 4294967295 に達した後は 0 に戻る。

ストリーム識別子 S: 16 ビット (符号無し整数)

本フィールドはユーザデータが属するストリームを識別する。

ストリームシーケンス番号 n: 16 ビット (符号無し整数)

本フィールドはユーザデータのストリーム S 内のストリームシーケンス番号(Stream Sequence Number: SSN)を表す。有効な値の範囲は 0 から 65535 である。

ユーザメッセージが転送のため SCTP によって分割された場合、分割されたメッセージのそれぞれの部分(fragment)では同一のストリームシーケンス番号が設定される。

ペイロードプロトコル識別子: 32 ビット (符号無し整数)

本フィールドはアプリケーション(または、上位レイヤ)特有のプロトコル識別子である。本フィールドの値は上位レイヤより SCTP に渡され、相手側へ送信される。本識別子は SCTP では使用しないが、相手側のアプリケーションや何らかのネットワークエンティティで DATA チャンクによって転送されるデータの種別を識別するために使用される。本フィールドは分割された DATA チャンクにおいても送信される(ネットワークの途中のエージェントに対しても利用可能であることを保証するため)。

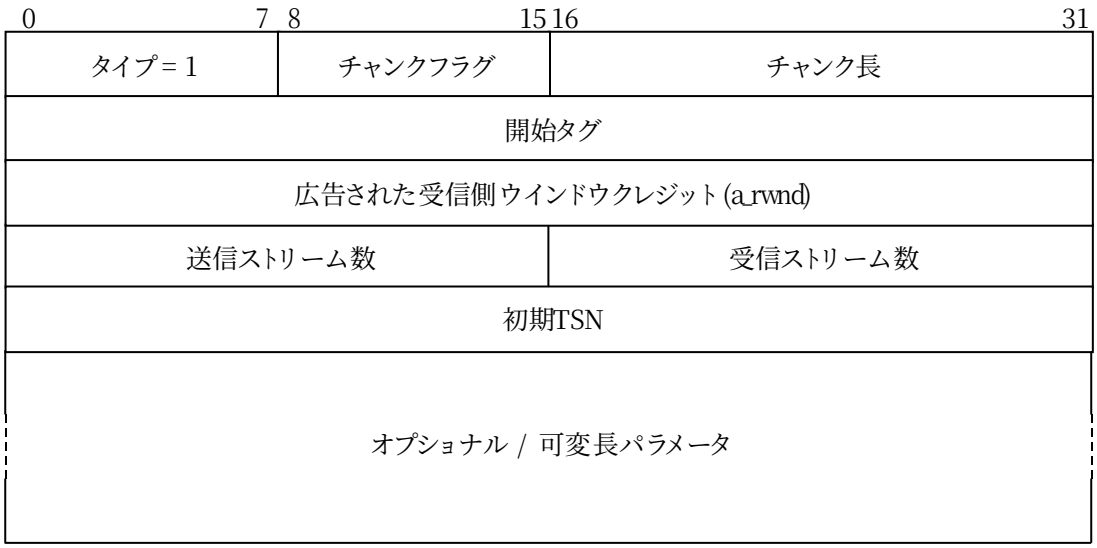
本フィールドの値 0 は、ペイロードデータが上位レイヤによってアプリケーション識別子を特定されていないことを示す。

ユーザデータ: 可変長

本フィールドがユーザデータのペイロードになる。ユーザデータが 4 バイトバウンダリに達するまで送信者は All 0 のバイト(Padding)で埋める。Padding の長さはデータ長フィールドの値に含めない。送信者は 3 バイトを超える padding をしない。

2.3.3.2 Initiation (INIT) (1)

本チャンクは 2 つのエンドポイントの間で SCTP アソシエーションを開始するときに用いられる。INIT チャンクのフォーマットを以下に示す。



INIT チャンクは以下のパラメタを含んでいる。特に規定がなければ、各パラメタは INIT チャンク中に 1 回のみ含まれる。

固定パラメタ	状態
開始タグ	必須
広告された受信側ウインドウクレジット	必須
送信ストリーム数	必須
受信ストリーム数	必須
初期 TSN	必須

可変パラメタ	状態	タイプ
IPv4 アドレス (*1)	オプション	5
IPv6 アドレス(*1)	オプション	6
Cookie Preservative	オプション	9
ECN 有効化のために予約 (*2)	オプション	32768 (0x8000)
ホスト名アドレス (*3)	オプション	11
サポートされるアドレスタイプ (*4)	オプション	12

*1: INIT チャンクは IPv4、IPv6 混在で複数のアドレスを持てる。

*2: ECN Capable フィールドは、将来の明示的な輻輳通知(Explicit Congestion Notification)のために予約されている。

*3: INIT チャンクは 1 つを超えるホスト名アドレスパラメタを含んではならない。さらに、INIT の送信者は INIT 中のホスト名アドレスと他のアドレスタイプを組み合わせてはならない。INIT の受信者は、ホスト名アドレスパラメタが受信した INIT チャンク内に含まれていれば、他のアドレスタイプを無視する。

*4: 本パラメタは、存在する場合には、送信側エンドポイントがサポートする全てのアドレスタイプを特定

する。本パラメタが存在しない場合は、送信側エンドポイントがどのようなアドレスタイプも許容することを示す。

INIT のチャンクフラグフィールドは予約されており、送信者は全てのビットを 0 とし、受信者は無視する。

開始タグ: 32 ビット (符号無し整数)

INIT の受信者(応答側(responding end))は開始タグの値を記録する。この値は INIT の受信者が当該アソシエーションの中で送信する全ての SCTP パケットの照合タグフィールドに設定される。

開始タグの値は 0 以外であればいかなる値をとってもよい。タグ値の選択についての詳細は 2.5.3.1 節を参照のこと。

受信した INIT の開始タグの値が 0 であった場合は、受信者はエラーとして扱い、ABORT を送信してアソシエーションを終了(close)する。

広告された受信者ウインドウクレジット (a_rwnd): 32 ビット (符号無し整数)

本フィールドの値は INIT の送信者が本 Window に関して確保した専用(dedicated)バッファ空間のサイズをバイト数で表す。アソシエーションの存続中はこのバッファ空間を縮小する (すなわち、専用バッファをアソシエーションから取り上げる) べきではない。しかし、エンドポイントは SACK チャンクで送信する a_rwnd の値を変更してもよい。

送信ストリーム数 (OS): 16 ビット (符号無し整数)

INIT チャンクの送信者が当該アソシエーションの中で作成を希望するするストリームの数を定義する。フィールド値 0 は使ってはならない。

注: OS の値が 0 の INIT の受信者はアソシエーションを廃棄(abort)する。

受信ストリーム数 (MIS): 16 ビット (符号無し整数)

INIT チャンクの送信者が相手側に許容する、当該アソシエーションの中で生成可能な最大ストリーム数を定義する。フィールド値 0 は使用されてはならない。

注: 実際のストリーム数の交渉はないが、その代わりに両エンドポイントは要求, 提示されたうち最小の値を使用する。詳細については、2.5.1.1 節参照のこと。

注: MIS の値が 0 の INIT の受信者はアソシエーションを廃棄(abort)する。

初期 TSN (I-TSN): 32 ビット (符号無し整数)

送信者が使用する TSN の初期値を定義する。取りうる値の範囲は 0 から 4294967295 である。本フィールドは開始タグフィールドの値が設定されてもよい。

2.3.3.2.1 INIT のオプショナル / 可変長パラメタ

以下のパラメタは、2.3.2.1 節に規定されているタイプ－長さ－値フォーマットに従う。いずれのタイプ－長さ－値フィールドも、前節にて規定されている固定長フィールドに続けて設定される。

IPv4 アドレスパラメタ (5)

IP v4 アドレス: 32 ビット (符号無し整数)

送信側エンドポイントの IP v4 アドレスを含み、2 進符号化される。

IPv6 アドレスパラメタ (6)

IP v6 アドレス: 128 ビット (符号無し整数)

送信側エンドポイントの IP v6 アドレスを含み、2 進符号化される。

注: 送信元は IP v4 マップされた IP v6 アドレス[RFC2373]を用いてはならない。しかし、代わりに IP v4 アドレスに対して IP v4 アドレスパラメタを使用する必要がある。

SCTP 共通ヘッダで送信元ポート番号と一緒に用い、IP v4、あるいは IP v6 アドレスパラメタでパスされた値は、開始されたアソシエーションを提供する INIT の送信元のトランスポートアドレスを示す。このアソシエーションの存続中に、この IP アドレスは、INIT の送信元から送信された IP データグラムの送信元アドレスフィールドに見ることができ、さらに、INIT の送信者から送信された IP データグラムの送信先アドレスとして使用されてもよい。INIT の送信元がマルチホームの場合、INIT チャンクに 1 つ以上の IP アドレスを含めることができる。

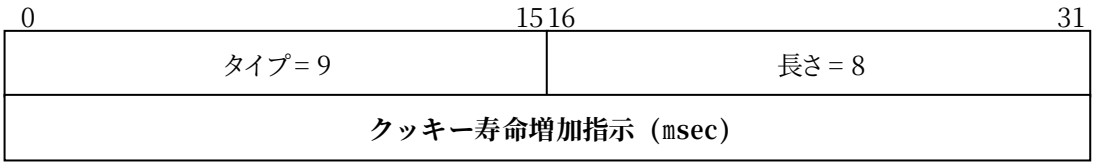
さらに、マルチホームエンドポイントは、異なるタイプのネットワークにアクセスすることができ、それゆえ、1 つの INIT チャンクにて 1 つ以上のアドレスタイプが提供されてもよい。すなわち、Ipv4 と IPv6 アドレスは同じ INIT チャンク内で混在することが許容される。

INIT に少なくとも 1 つの IP アドレスパラメタが含まれているなら、INIT チャンクに含まれる IP データグラムの送信元アドレスと INIT にて与えられる何れかの付加的なアドレスは、INIT の受信エンドポイントにおいて送信先として用いることができる。INIT が IP アドレスパラメタを含んでいないなら、INIT を受信したエンドポイントでは、受信した IP データグラムの送信元アドレスをアソシエーションの唯一の着信先アドレスとして用いる。

INIT 及び INIT－ACK の中で如何なる IP アドレスパラメタも使用しない場合は、NAT ボックスを横切るようなときに、アソシエーション生成の選択肢となることに留意。

クッキー保存 (9)

INIT の送信元は、状態クッキーのライフ・スパン延長のため、INIT の送信先への示唆にこのパラメタを用いる。



クッキー寿命増加指示(Suggested Cookie Life-span Increment): 32 ビット (符号無し整数)

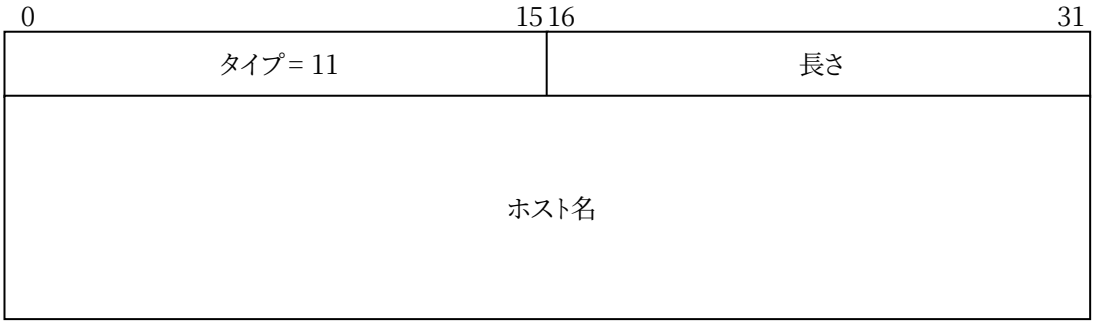
このパラメタは、受信者にミリ秒でどの程度の増加をそのデフォルトクッキー・ライフ・スパンに加えることを送信側が受信者に要望しているか示す。

このオプションパラメタは、古いクッキーオペレーションエラーにより以前にアソシエーション確立の

試みが失敗した相手側とのアソシエーション確立を再試行する場合に、送信者により INIT チャンクに加えられる。受信者は、自身のセキュリティの理由でクッキーのライフスパン増長の指示を無視することを選択してもよい。

ホスト名アドレス (11)

INIT の送信者は、(IP アドレスの代わりに)ホスト名を相手側へ渡すためにこのパラメタを用いる。相手側は、名前の解決に責任を負う。このパラメタを使用することで、ＮＡＴボックスを横切るアソシエーションの動作を助長するかもしれない。



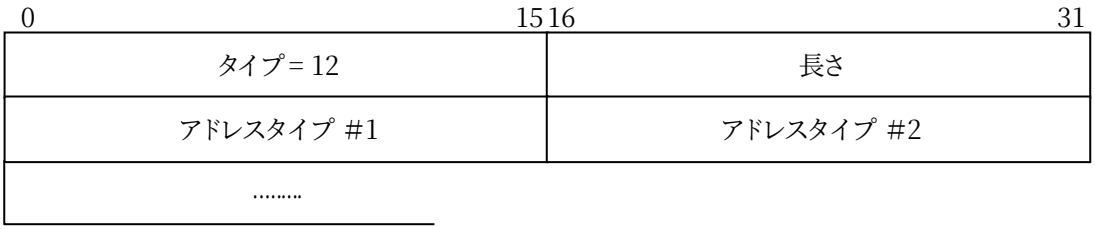
ホスト名: 可変長

このフィールドは、RFC1123 の 2.1 節の「ホスト名シンタックス」で記述されるホスト名を含んでいる。ホスト名を解決する方法は、ＳＣＴＰの範囲の外です。

注: 少なくとも 1 つのヌル・ターミネータは、ホスト名列に含まれており、チャンク長に含まれている。

サポートされるアドレスタイプ (12)

INIT の送信者は、サポート可能なアドレスタイプのすべてを列挙するために、このパラメタを用いる。



アドレスタイプ: 16 ビット (符号無し整数)

該当するアドレスの TLV のタイプ値が設定される。(例えば、IPv4=5、IPv6=6、ホスト名=11)

2.3.3.3 **開始確認** (INIT ACK) (2):

INIT ACK チャンクは、ＳＣＴＰアソシエーションの開始を認めるために用いられる。

INIT ACK のパラメタ部分は、INIT チャンクと同様にフォーマットされる。それは、2 つの余分な可変パラメタ (状態クッキーと認識不能パラメタ) を用いる。

INIT ACK チャンクのフォーマットを以下に示す。

開始タグ: 32 **ビット (符号無し整数)**

INIT ACK **の受信者は、開始タグパラメタの値を記録する。この値は、**INIT ACK **の受信者がこのアソシエーション内に送信するすべての** SCTP **パケットのベリフィケーションタグフィールドに置かれる。**

開始タグは、値ゼロをとってはならない。開始タグ値の選択の詳細については、2.5.3.1 **節を参照。**

受信された INIT ACK **チャンクの開始タグの値が、ゼロであると分かった場合、受信者はエラーとしてそれを扱い、**ABORT **の送信によりアソシエーションを閉じる。**

広告された受信者ウインドウクレジット (a_rwnd): 32 **ビット (符号無し整数)**

この値は、バイト数で専用のバッファスペースを表わし、INIT ACK **の送信者は、アソシエーションにおいてこのウインドウが確保される。アソシエーションの存続中は、このバッファスペースを縮小してはならない。**

送信ストリーム数 (OS): 16 **ビット (符号無し整数)**

この INIT ACK **チャンクの送信者がこのアソシエーションで生成したい、送信ストリーム数を定義する。値ゼロは、使用されてはならない。**

注: OS 値がゼロに設定された INIT ACK **を受信した側は、その** TCB **を廃棄するアソシエーションを中断する必要がある。**

受信ストリーム数 (MIS): 16 **ビット (符号無し整数)**

この INIT ACK **チャンクの送信者が相手側に許容する、当該アソシエーションの中で生成可能な最大ストリーム数を定義する。フィールド値** 0 **は使用されてはならない。**

注: 実際のストリーム数の交渉はないが、その代わりに両エンドポイントは要求，提示されたうち最小の値を使用する。詳細に関しては、2.5.1.1 **節を参照。**

注: MIS **値がゼロに設定された** INIT ACK **を受信した側は、その** TCB **を廃棄するアソシエーションを中断する必要がある。**

初期 TSN (I-TSN): 32 **ビット (符号無し整数)**

INIT ACK **送信者が使用する開始** TSN **を定義する。有効な範囲は、**0～4294967295 **である。このフィールドは、開始タグフィールドの値が設定されてもよい。**

固定パラメタ	状態
開始タグ	必須
広告された受信者ウインドウクレジット	必須
送信ストリーム数	必須
受信ストリーム数	必須
初期 TSN	必須

可変パラメタ	状態	タイプ
状態クッキー	必須	7
IPv4 アドレス (*1)	オプション	5
IPv6 アドレス(*1)	オプション	6
認識不能パラメタ	オプション	8
ECN 有効化のために予約 (*2)	オプション	32768 (0x8000)
ホスト名アドレス (*3)	オプション	11

注 1: INIT ACK チャンクは、IPv4 と/または IPｖ6 のコンビネーションによる IP アドレスパラメタを任意の数含むことができる。

注 2: ＥＣＮ能力フィールドは、将来使用する明示的な輻輳通知のために予約されている。

注 3: INIT ACK チャンクは、1 つ以上のホスト名アドレスパラメタを含んではならない。さらに、INIT ACK の送信者は、INIT ACK の中にてホスト名アドレスと他のアドレスタイプを組み合わせてはならない。INIT ACK の受信者は、ホスト名アドレスパラメタが存在する場合、他のアドレスタイプを無視しなければなれない。

実装上の注意: 実装において、可変長の状態クッキーおよび可変アドレスリストにより、きわめて大きい (1500 バイト以上) INIT ACK を受け取るための用意がされていなければならない。例えば、INIT への応答側が、送りたい 1000 個の IPｖ4 アドレスを持っている場合、それは INIT ACK でこれをコード化するためには、少なくとも 8000 バイトを必要とする。

ＳＣＴＰの共通ヘッダ中で運ばれる送信元ポートと結合して、INIT ACK 中の各 IP アドレスパラメタは、INIT ACK の受信者に、開始されたアソシエーションのライフタイムの間、INIT ACK の送信者によって提供される有効なトランスポートアドレスを示す。

INIT ACK が少なくとも 1 つの IP アドレスパラメタを含んでいる場合、INIT ACK を含む IP データグラムの送信元アドレス、および INIT ACK の内に提供されるいかなる付加的なアドレスは、INIT ACK の受信者によって着信先として使用されても良い。INIT ACK が IP アドレスパラメタを含んでいない場合、INIT ACK の受信者は、アソシエーションのためにその唯一の送信先アドレスとして受信 IP データグラムの送信元アドレスを用いる。

状態クッキーおよび確認不能パラメタは、2.3.2.1 節と以下の記述で定義されるタイプ－長さ－値フォーマットを使用する。他のフィールドは、INIT チャンク中のそれらに相当するものと同じように定義される。

2.3.3.3.1 オプショナル / 可変長パラメタ

状態クッキー

パラメタタイプ値: 7

パラメタ長:可変サイズ、クッキーのサイズに依存

パラメタ値:

INIT ACK **送信者がアソシエーション生成のために必要な状態とパラメタ情報をメッセージ認証コード**(MAC)**とともに設定する。状態クッキー定義の詳細に関しては、**2.5.1.3 **節を参照。**

認識不可パラメタ:

パラメタタイプ値: 8

パラメタ長: **可変サイズ**

パラメタ値:

INIT **チャンクに、送信者に報告される必要があることを示す値が設定された認識不可パラメタが含まれる場合、**INIT **チャンク発信者に返送される。このパラメタ値フィールドには、ＩＮＩＴチャンク内の認識不可パラメタからコピーしたパラメタタイプ、長さおよび値フィールドを設定する。**

2.3.3.4 **選択的確認** (SACK) (3):

このチャンクは、受信した DATA **チャンクの確認応答と、**TSN **により示される** DATA **チャンクの受信サブシーケンスにおけるギャップを対向するエンドポイントへ通知するため送信される。**

SACK **は、累積** TSN Ack **と、広告された受信者ウインドウクレジット**(a_rwnd)**パラメタを含める。**

累積 TSN Ack **パラメタの値は、受信** TSN **のシーケンスの切断が生じる前に受信された最後の** TSN **である。これに続く次の** TSN **値は、**SACK **を送信するエンドポイントでまだ受け取られていない。このパラメタは、その値以下のすべての** TSN **受信に対し確認応答する。**

SACK **受信者による** a_rwnd **の扱いは、**2.6.2.1 **節にて詳細に示す。**

SACK **は、ゼロ以上のギャップ** Ack **ブロックを含み、各ギャップ** Ack **ブロックは、受信** TSN **シーケンス切断に続いて受信した** TSN **のサブシーケンスの確認応答をする。ギャップ** Ack **ブロックにより確認応答するすべての** TSN **は、累積** TSN Ack **の値より大きい。**

チャンクフラグ: 8 ビット

送信時は、すべてゼロに設定し、受信時は無視する。

累積 TSN Ack: 32 ビット(符号無し整数)

ギャップの前に受信した最後の DATA チャンクの TSN を設定する。

広告された受信者ウインドウクレジット (a_rwnd): 32 ビット (符号無し整数)

当該 SACK 送信者の最新受信バッファスペースをバイトにより示す。詳細は、2.6.2.1 節参照。

ギャップ Ack ブロック数: 16 ビット (符号無し整数)

当該 SACK に含まれるギャップ Ack ブロック数を示す。

重複 TSN 数: 16 ビット

エンドポイントが受信した重複 TSN 数。各々の重複 TSN は、ギャップ Ack ブロックリストに続いて列挙される。

ギャップ Ack ブロック:

各ギャップＡｃｋブロックは、ギャップ Ack ブロック数フィールドに定義されたギャップ Ack の数まで繰り返される。全ての DATA チャンクのＴＳＮは、(累積 TSN Ack＋ギャップ Ack ブロックスタート)以上、かつ(累積 TSN Ack＋ギャップ Ack ブロックエンド)以下となる。

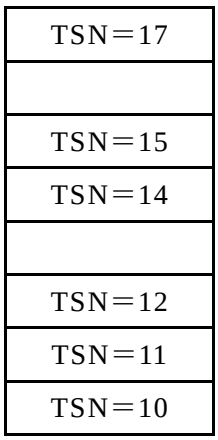
ギャップ Ack ブロックスタート: 16 ビット (符号無し整数)

当該ギャップ Ack ブロックに対するスタートオフセット TSN を示す。実際の TSN 番号を計算するために、累積 TSN Ack がこのオフセット値に加えられる。計算された TSN は、すでに受信されたギャップ Ack ブロック中の始めの TSN を識別する。

ギャップ Ack ブロックエンド: 16 ビット (符号無し整数)

当該ギャップ Ack ブロックに対するエンドオフセット TSN を示す。実際の TSN 番号を計算するために、累積 TSN Ack がこのオフセット値に加えられる。計算された TSN は、ギャップ Ack ブロックで受信された最後の DATA チャンクの TSN を識別する。

例えば、受信者が選択的 Ack の送信を決定する時に、そのときに新たに着信した以下の DATA チャンクを持つと仮定する。



← 不明

← 不明

そのとき、SACK のパラメタ部分は、以下のようになる。 (新たな a_rwnd が送信者により、4660 にセットされると仮定して)

累積 TSN Ack＝12	
a_rwnd ＝ 4660	
ギャップ Ack ブロック数 = 2	重複 TSN 数 = 0
ギャップ Ack ブロック #1 スタート = 2	ギャップ Ack ブロック #1 エンド = 3
ギャップ Ack ブロック #2 スタート = 5	ギャップ Ack ブロック #2 エンド = 5

重複 TSN: 32 **ビット (符号無し整数)**

最後の SACK **が送られてから** TSN **が重複して受信された回数を示す。受信者が重複** TSN(SACK **を送る前に)を得る毎に、重複リストに加える。重複回数は各** SACK **を送った後にゼロに再初期化される。**

例えば、受信者が TSN19 **を** 3 **回得たならば、それはアウトバウンド** SACK **で** 19 **が** 2 **度挙げられることになる。**SACK **を送信した後に、さらに** TSN19 **を依然として受信した場合、次の出** SACK **で** 19 **を重複として** 1 **度挙げられる。**

2.3.3.5 **ハートビート要求** (HEARTBEAT) (4)

エンドポイントがアソシエーションにより指定された送信先アドレスへの reachability **を確認するため、対向するエンドポイントに対して送る。**

パラメタフィールドには可変長のハートビート情報が含まれ、HEARTBEAT **を送信する側で規定する。**

チャンクフラグ: 8 **ビット**

"0"**にセットする。受信者では無視される。**

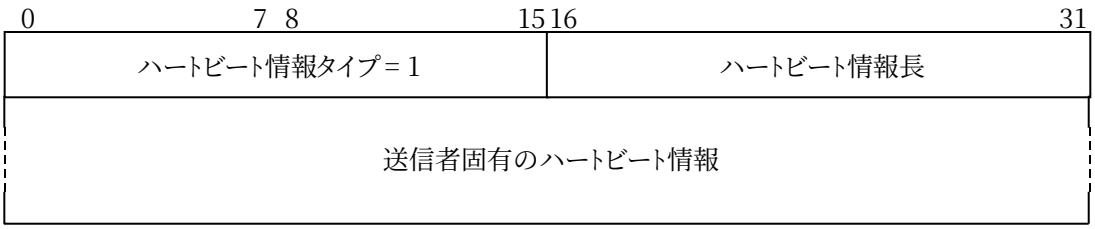
ハートビート長: 16 **ビット (符号無し整数)**

チャンクのサイズ(バイト)をセットする。チャンクヘッダ及び、ハートビート情報フィールドに含まれる。

ハートビート情報: **可変長**

フォーマットは 2.8.3 **節参照のこと。**

可変パラメタ	状態	タイプ
ハートビート情報	必須	1



通常、ハートビート情報フィールドには送信者の送信時の現在時刻と送信先のトランスポートアドレスを入れる。2.8.3 節参照のこと。

2.3.3.6 ハートビート承認 (HEARTBEAT ACK) (5)

エンドポイントが HEARTBEAT チャンクのレスポンスとして、対向するエンドポイントに対して送る。HEARTBEAT ACK は常に IP データグラムの送信元 IP アドレスに送られ、そのパラメタフィールドは可変長となる。

チャンクフラグ: 8 ビット

"0"にセットする。受信者では無視される。

ハートビート Ack 長: 16 ビット (符号無し整数)

チャンクのサイズ(バイト)をセットする。チャンクヘッダ及び、ハートビート情報フィールドに含まれる。

ハートビート情報: 可変長

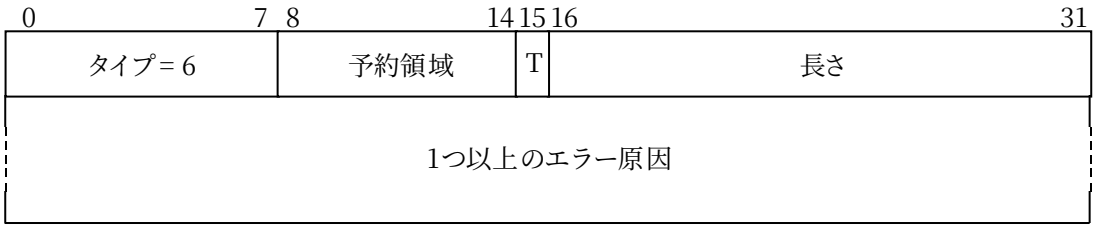
ハートビート情報 は、ハートビート要求のハートビート情報パラメタを含む。

可変パラメタ	状態	タイプ
ハートビート情報	必須	1

2.3.3.7 アソシエーション中断 (ABORT) (6)

アソシエーションを終了するときに用いられ、アボート理由を受信者に通知するための理由表示パラメタを含んでもよい。また、DATA チャンクとバンドルしてはならないが、INIT, INIT ACK, SHUTDOWN COMPLETE 以外の制御チャンクとのバンドルは ABORT 前に限り、許容されている。そうでない場合は受信者で無視される。

受信者では ABORT がフォーマットエラーや存在しないアソシエーションに関するものであった場合、破棄される。更に受信者では、どんな場合でも ABORT に対して ABORT のレスポンスを返してはならない。



チャンクフラグ: 8 **ビット**

予約領域: 7 **ビット**

“0”**にセットする。受信者では無視される。**

T **ビット**: 1 **ビット**

TCB **が破壊されている時は**“0”**にセットされる。**TCB **を持っていない時は**“1”**をセットする。**

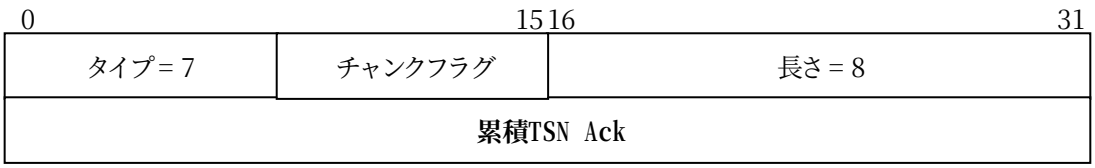
注: ベリフィケーションについてはこの限りではない。詳細は 2.8.5.1 **節参照のこと。**

長さ: 16 **ビット (符号無し整数)**

チャンクのサイズ(バイト)をセットする。チャンクヘッダ及び、全てのエラー理由フィールドを含む。詳細は、2.3.3.10 **節 (エラー原因定義) を参照のこと。**

2.3.3.8 **アソシエーション停止** (SHUTDOWN) (7)

アソシエーション中のエンドポイントは、対向するエンドポイントとのアソシエーションを正常に停止させるために、このチャンクを使用する。



チャンクフラグ: 8 **ビット**

“0”**にセットする。受信者では無視される。**

長さ: 16 **ビット (符号無し整数)**

パラメタ長を示す。“ 8 **”にセットする。**

累積 TSN Ack: 32 **ビット (符号無し整数)**

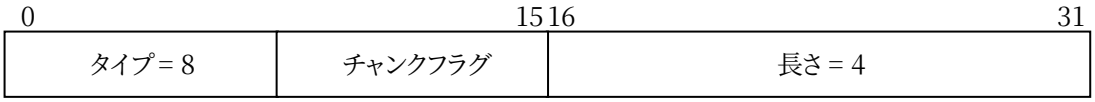
ギャップの前に受信した最終チャンクの TSN **が含まれる。**

注: SHUTDOWN **については** SACK **と違ってギャップ** Ack **ブロックが含まれないため、**SHUTDOWN **受信者では、**Gap Ack Block **の欠落を** renege **と解釈すべきではない。**2.6.2 **節**(Information on reneging)**参照のこと。**

2.3.3.9 **シャットダウン承認** (SHUTDOWN ACK) (8)

このチャンクは、SHUTDOWN **チャンクを受信し、シャットダウン完了を承認するときに使用される。詳細は** 2.9.2 **節を参照のこと。**

SHUTDOWN ACK **にはパラメタが含まれない。**

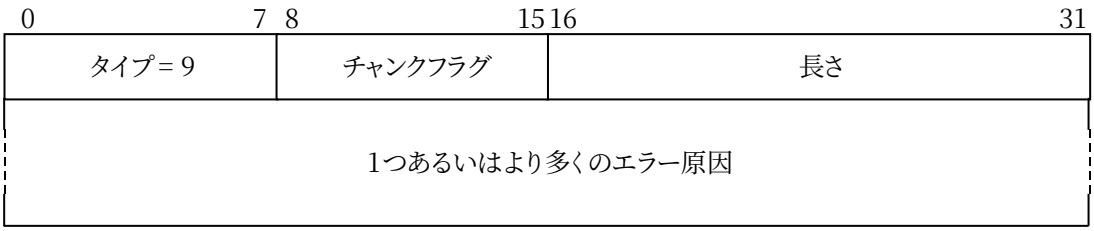


チャンクフラグ: 8 **ビット**

“0”**にセットする。受信者では無視される。**

2.3.3.10 **オペレーションエラー** (ERROR) (9)

対向するエンドポイントにエラー状況を通知するため送られるチャンクで、一つ以上のエラー理由を内容とする。オペレーションエラーはそれ自体では致命的なエラーではない。致命的な状態を通知するには、ABORT を伴う。



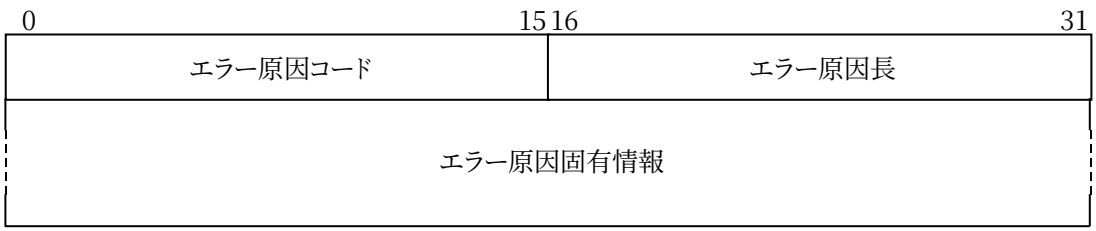
チャンクフラグ: 8 **ビット**

"0"にセットする。受信者では無視される。

チャンク長: 16 **ビット (符号無し整数)**

チャンクのサイズ(バイト)をセットする。チャンクヘッダ及び、全てのエラー原因フィールドを含む。

エラー原因は 2.3.2.1 **節に示したフォーマットによる可変長パラメタである。**



エラー原因コード: 16 **ビット (符号無し整数)**

報告されるエラー状態のタイプを定義する。

原因コード値	原因コードの内容
1	不正なストリーム識別子
2	必須パラメタ欠落
3	失効クッキーエラー
4	リソース不足
5	解決不能アドレス
6	認識できないチャンクタイプ
7	必須パラメタ無効
8	認識できないパラメタ
9	ユーザデータ無し
10	シャットダウン中のクッキー受信

エラー原因長: 16 **ビット (符号無し整数)**

エラー原因コード、原因長、原因固有情報の各フィールドを含む、パラメタのサイズ(バイト)をセットする。

エラー原因固有情報: 可変長

詳細なエラー状況を転送する。

2.3.3.10.1～2.3.3.10.10 節にてＳＣＴＰのエラー原因を定義する。新しいエラー原因の値を定義するためのIETF ガイドラインについては、2.13.3 節で論ずる。

2.3.3.10.1 **不正なストリーム識別子** (1)

エラー原因

不正なストリーム識別子 (Invalid Stream Identifier): **エンドポイントが、存在しない** Stream **へ送信しようとしたＤＡＴＡチャンクを受信したことを示す。**

0	15 16	31
エラー原因コード = 1		エラー原因長 = 8
ストリーム識別子		(予約済み)

ストリーム識別子: 16 **ビット (符号無し整数)**

エラー受信された DATA **チャンクのストリーム識別子を設定する。**

予約領域: 16 **ビット**

予約済み。"0"**がセットされ、受信者では無視される。**

2.3.3.10.2 **必須パラメタ欠落** (2)

エラー原因

必須パラメタ欠落(Missing Mandatory Parameter): **受信した** INIT **もしくは** INIT-ACK **中にＴＬＶ必須パラメタが** 1 **つ以上欠落していることを示す。**

0	15 16	31
エラー原因コード = 2		エラー原因長 = 8 + N×2
欠落パラメータ数 = N		
欠落パラメータタイプ #1		欠落パラメータタイプ #2
欠落パラメータタイプ #N-1		欠落パラメータタイプ #N

欠落パラメタ数: 32 **ビット (符号無し整数)**

このフィールドには、原因固有情報フィールドに含めるパラメタ数を設定する。

欠落パラメタタイプ: 16 **ビット**

各フィールドに欠落している必須パラメタの番号を設定する。

2.3.3.10.3 **失効クッキーエラー** (3)

エラー原因

失効クッキーエラー (Sta l e Cookie Error): **有効な状態クッキーだが、期限切れであることを示す。**

0	15 16	31
エラー原因コード = 3		エラー原因長 = 8
失効時間		

失効時間: 32 **ビット** (**符号無し整数**)

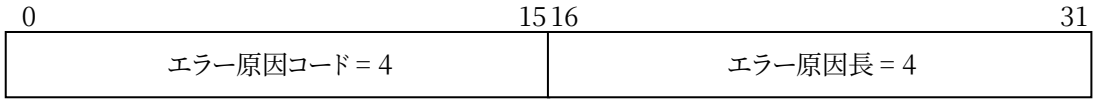
このフィールドには、現在時刻と状態クッキーが失効した時刻との差を設定する。単位はマイクロ秒。

本エラーの送信者は、状態クッキーがどのくらい前に失効したかを通知するために、失効時間フィールドにゼロ以外の値を設定することができる。送信者がこの情報提供を望まない場合は、 Measure of Staleness **に**"0"**を設定することを推奨する。**

2.3.3.10.4 **リソース不足** (4)

エラー原因

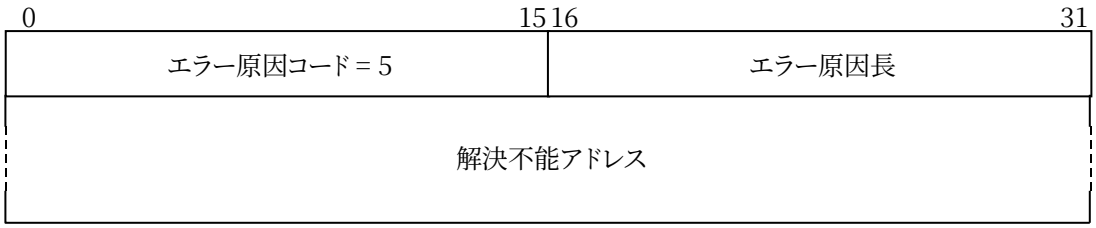
リソース不足(Out of Resource): **送信者のリソースが不足したことを示す。通常は** ABORT **とともに送られる。**



2.3.3.10.5 **解決不能アドレス** (5)

エラー原因

解決不能アドレス (Unresolvable Address): **送信者がアドレスタイプをサポートしていない等の理由で当該パラメタを解決できないことを示す。通常は** ABORT **と共に送られる。**



解決不能アドレス: **可変長**

解決不能アドレスフィールドには、解決できないアドレス、もしくはホスト名を含むアドレスパラメタのタイプ、長さ、値を設定する。

2.3.3.10.6 **認識できないチャンクタイプ** (6)

エラー原因

認識できないチャンクタイプ (Unrecognized Chunk Type): **受信者がチャンクを認識できず、チャンクタイプの上位ビットが** 01 **または** 11 **にセットされていれば、このエラー原因がチャンクの作成者に返される。**

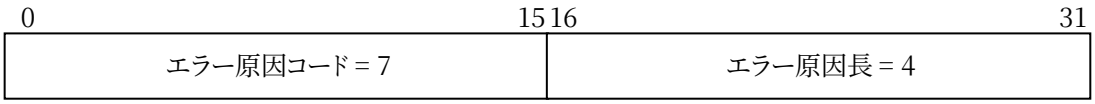
認識不可チャンク: **可変長**

認識不可チャンクフィールドには、チャンクタイプ、チャンクフラグ、チャンク長を含む完全な SCTP **パケットからの認識不可チャンクを設定する。**

2.3.3.10.7 **必須パラメタ無効** (7)

エラー原因

必須パラメタ無効 (Invalid Mandatory Parameter): **必須パラメタのうちの** 1 **つに無効値がセットされていた場合、このエラー原因が** INIT **または** INIT ACK **チャンクの作成者に返送される。**



2.3.3.10.8 **認識できないパラメタ** (8)

エラー原因

認識できないパラメタ (unrecognized Parameters): **受信者が** INIT ACK **チャンク中、**1 **つ以上のオプション** TLV **パラメタを認識できなければ、このエラー原因が** INIT ACK **チャンクの作成者に返送される。**

認識不可パラメタ: **可変長**

認識不可パラメタフィールドには、INIT ACK **チャンクからコピーした認識不可パラメタが入れられる。通常、**INIT ACK **への応答時、**COOKIE ECHO **チャンク送信者が認識不可パラメタの報告を要求する場合に、**COOKIE ECHO **チャンクとバンドルされた**ＥＲＲＯＲ**チャンクで使われる。**

2.3.3.10.9 **ユーザデータ無し** (9)

エラー原因

ユーザデータ無し(No User Data): **受信** DATA **チャンクがユーザデータを持っていない場合、このエラー原因が** DATA **チャンクの作成者に返送される。**

TSN 値: 32 ビット (符号無し整数)

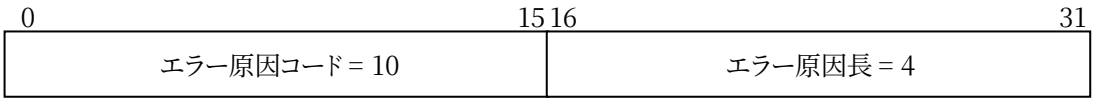
TSN 値フィールドには､ユーザデータ･フィールドなしで受信した DATA チャンクの TSN を設定する。

この原因コードは、ABORT チャンクの中で通常返される(2.6.2 節を参照)。

2.3.3.10.10 シャットダウン中のクッキー受信 (10)

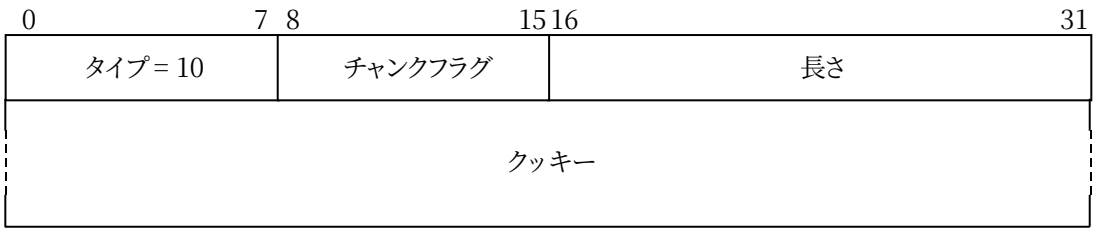
エラー原因

シャットダウン中のクッキー受信(Cookies Received While Shutting Down): エンドポイントが SHUTDOWN-ACK-SENT 状態で、COOKIE ECHO が受信された場合。このエラーは、通常、再送される SHUTDOWN ACK チャンクとバンドルされる ERROR チャンクで返送される。



2.3.3.11 クッキーエコー (COOKIE ECHO) (10)

このチャンクはアソシエーションの初期化中にのみ使用され、初期化プロセスを完了するためにアソシエーションの相手側へアソシエーションの開始者から送られる。このチャンクは、アソシエーション内で送られるどの DATA チャンクよりも先に送信するが､1 つ以上の DATA チャンクと同一パケットにバンドルしてもよい。



チャンクフラグ: 8 ビット

“0”がセットされ、受信者では無視される。

チャンク長: 16 ビット (符号無し整数)

チャンクのサイズ。ヘッダの 4 バイトと Cookie のサイズが含まれる。単位はバイト。

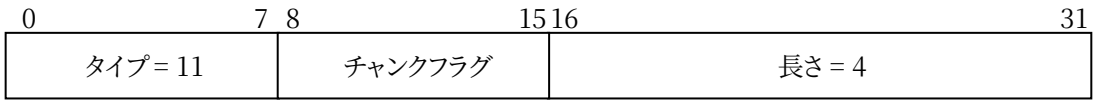
クッキー: 可変長

受信済み INIT ACK 内の状態クッキーパラメタ内のクッキーを用いる。

インプリメント時には相互接続性を保証するためにクッキーをできるだけ小さくすることを推奨する。

2.3.3.12 クッキー承認 (COOKIE ACK) (11)

このチャンクはアソシエーションの初期化中にのみ使用され、COOKIE ECHO チャンクの受信を承認するために使用される。このチャンクは、アソシエーション内で送られる他のどの DATA あるいは SACK チャンクよりも先に送信するが、1 つ以上の DATA チャンクと同一 SCTP パケットにバンドルしてもよい。



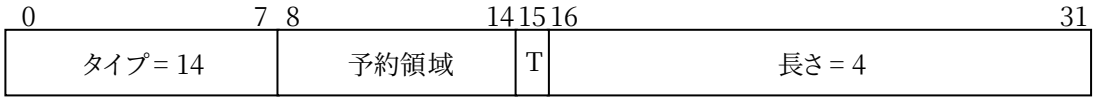
チャンクフラグ: 8 **ビット**

“0”**がセットされ、受信者では無視される。**

2.3.3.13 **シャットダウン完了**(SHUTDOWN COMPLETE) (14)

このチャンクはシャットダウンプロセスの完了時に SHUTDOWN ACK **チャンクの受信を承認するために使用する。詳細はセクション** 2.9.2 **を参照。**

SHUTDOWN COMPLETE **チャンクはパラメタを持っていない。**



チャンクフラグ: 8 **ビット**

予約領域: 7 **ビット**

“0”**にセットする。受信者では無視される。**

T **ビット**: 1 **ビット**

TCB **が破壊されている時は**“0”**にセットされる。**TCB **を持っていない時は**“1”**をセットする。**

注: 特別のルールが、照合用に本チャンクへ適用される。詳細は 2.8.5.1 **節を参照のこと。**

2.4 アソシエーション状態遷移図

SCTP **アソシエーションのライフタイムの間に、**SCTP **エンドポイントのアソシエーションは様々なイベントに応じて** 1 **つの状態から別の状態へ進行する。アソシエーションの状態を遷移させる可能性があるイベントには次のものがある。**

- SCTP **ユーザ・プリミティブ・コール。例えば**[ASSOCIATE], [SHUTDOWN], [ABORT]。
- INIT, COOKIE ECHO, ABORT, SHUTDOWN**、などのコントロールチャンクの受理。**
- **もしくは、いくつかのタイムアウトイベント**

下記の状態遷移図は、引き起こすイベントと、その結果生じるアクションと伴に、状態変化の例を示す。いくつかのエラー状態は状態図の中に示されていないことに注意することを推奨する。特殊なケースに対する十分な記述はテキスト中に示される。

注:チャンク名はすべて大文字で示されている。例えば、COOKIE ECHO **はチャンクタイプである。状態遷移を引き起こすイベント/メッセージが** 1 **つ以上生じる場合、**(A)、(B)**のようにラベルされている。**

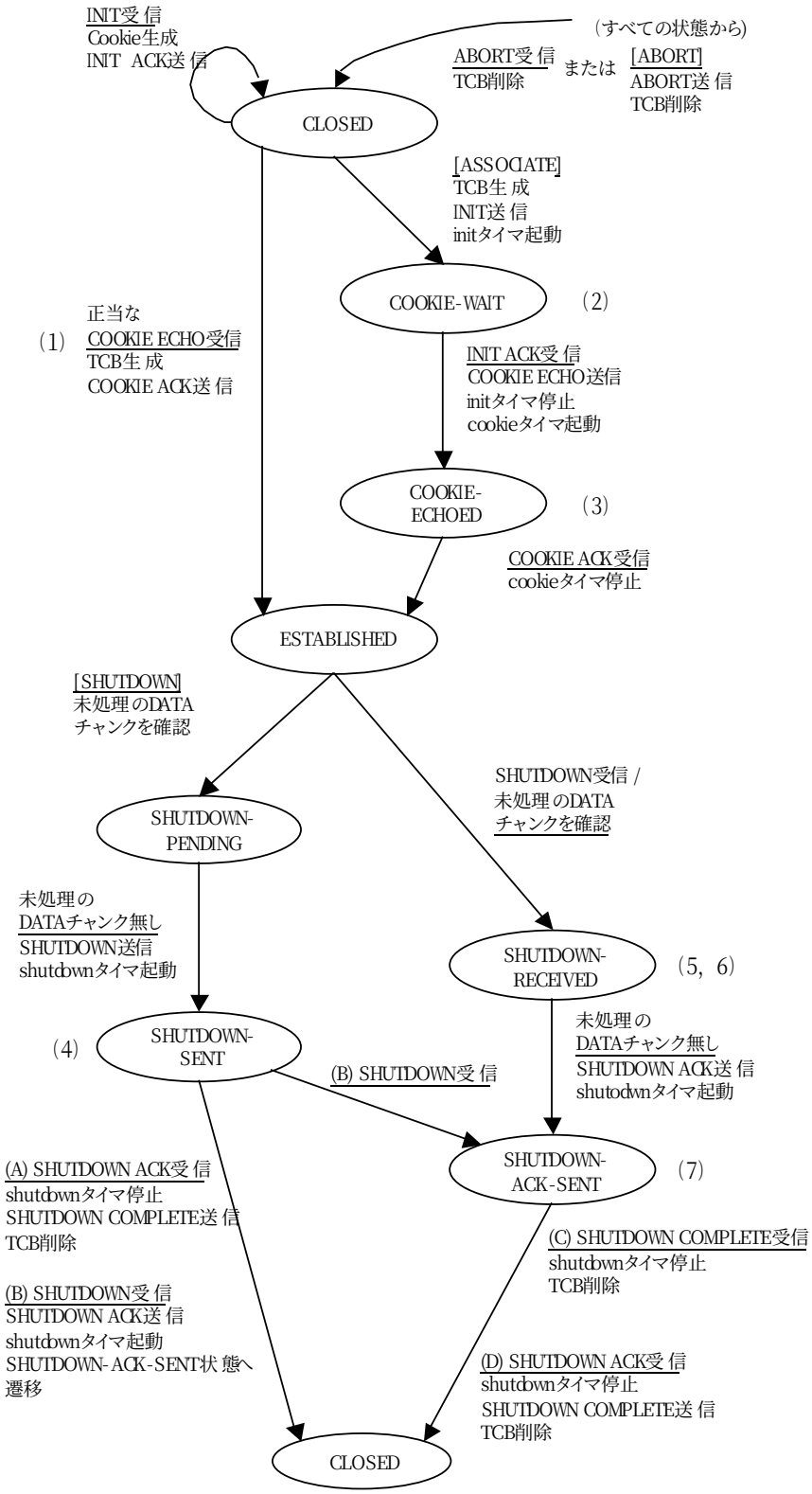


図 2-3　SCTP の状態遷移図

注:

1） 受信した COOKIE ECHO の中の状態クッキーが無効だった (つまり完全チェックをパスしなかった) 場合、受信した側は暗黙のうちにパケットを廃棄する。あるいは、受信した状態クッキーが期限切れの場合 (2.5.2.5 節を参照)、受信者は ERROR チャンクを送り返す。いずれの場合も、受信者は CLOSED 状態のままとなる。

2） T1-init タイマが満了する場合、エンドポイントは状態を変更せずに INIT を再送信し、再度 T1-init タイマを起動する。これは `Max.Init.Retransmit' 回まで繰り返す。その後は、エンドポイントは初期化プロセスを異常終了させて、SCTP ユーザにエラーを報告する。

3） T1-cookie タイマが満了する場合、エンドポイントは状態を変更せずに COOKIE ECHO を再送信し、再度 T1-cookie タイマを起動する。これは `Max.Init.Retransmit' 回まで繰り返す。その後は、エンドポイントは初期化プロセスを異常終了させて、SCTP ユーザにエラーを報告する。

4） SHUTDOWN-SENT 状態では、エンドポイントは受信した DATA チャンクを遅滞なく承認する。

5） SHUTDOWN-RECEIVED 状態では、エンドポイントは SCTP ユーザから新しく送信された要求を受理してはならない。

6） SHUTDOWN-RECEIVED 状態では、エンドポイントはデータを送信もしくは再送信し、キュー内の全てのデータを送信し終えたら、この状態を脱する。

7） SHUTDOWN-ACK-SENT 状態では、エンドポイントは SCTP ユーザから新しく送信される要求を受理してはならない。

CLOSED の状態は、アソシエーションが作成されていない (つまり、存在していない) ことを示すために使用される。

2.5 アソシエーション初期化

ある SCTP エンドポイント ("A") から別の SCTP エンドポイント ("Z") への、最初のデータ送信が実施される前に、2 つのエンドポイントは両者間の SCTP アソシエーションをセットアップするための初期化プロセスを完了させている。

エンドポイント側の SCTP ユーザは、別の SCTP エンドポイントへの SCTP アソシエーションを初期化するために、ASSOCIATE プリミティブを使用することを推奨する。

実装時の注意: SCTP ユーザから見ると、目的のエンドポイントへ最初のユーザデータを送信することによって、ASSOCIATE プリミティブ (2.10.1 節の B を参照) の起動なしで、アソシエーションは暗黙のうちにオープンされている。SCTP の開始時には、INIT/INIT ACK の必須もしくはオプションのパラメタに対してデフォルト値を仮設定する。

一度アソシエーションが確立されれば、一方向のストリームが両エンドポイント間のデータ伝送のために開いている (2.5.1.1 節を参照)。

2.5.1 アソシエーションの正常確立

初期化プロセスは、次のステップから構成される (SCTP エンドポイント "A" が SCTP エンドポイント "Z" とのアソシエーションをセットアップしようとし、"Z" がその新しいアソシエーションを受理すると仮定して) :

A) "A" は、最初に "Z" に INIT チャンクを送る。INIT では、"A" は開始タグフィールドの照合タグ (Tag_A) を与える。Tag_A は 1〜4294967295 の範囲中の乱数とすることを推奨する (タグ値選択に関しては 2.5.3.1 節を参照)。INIT を送った後に、"A" は T1-init タイマを起動し、COOKIE-WAIT 状態に入る。

B) "Z" は、INIT ACK チャンクで直ちに応答する。INIT ACK 中の送信先 IP アドレスには、この INIT ACK

が応答している INIT の送信元 IP アドレスがセットされる。応答の際に、他のパラメタを書き込むことに加えて、“ Z” は Tag_A を照合タグフィールドにセットし、さらに、開始タグフィールドに自分自身の照合タグ (Tag_Z) を与える。

さらに、“ Z” は状態クッキーを生成し、INIT ACK に加えて送る。状態クッキー生成に関しては、2.5.1.3 節を参照のこと。

注: 状態クッキーパラメタと一緒に INIT ACK を送信した後、“ Z” はどんなリソースも割り付けてはならない。また、新しいアソシエーションのためにその状態を維持する。そうでないと、“ Z”はリソースに対する攻撃に弱くなる。

C) “ Z”からの INIT ACK の受信時に、“ A”は T1-init タイマを止め、COOKIE-WAIT 状態から脱出する。その後、“ A”は COOKIE ECHO チャンクの中の INIT ACK チャンクの中で受信した状態クッキーを送り、T1-cookie タイマを起動し、COOKIE-ECHOED 状態に入る。

注: COOKIE ECHO チャンクは任意のペンディング状態の外向け DATA チャンクとバンドルすることができる。しかし、それはパケット中の最初のチャンクとする。また、COOKIE ACK が返送されるまで、送信者は相手側に他のパケットを送ってはならない。

D) COOKIE ECHO チャンクの受信時に、エンドポイント“ Z” は TCB を構築し、ESTABLISHED 状態に移動した後に、COOKIE ACK チャンクで返答する。COOKIE ACK チャンクは任意のペンディング状態の DATA チャンク (または SACK チャンク) とバンドルしてもよい。しかし COOKIE ACK チャンクはパケット中の最初のチャンクとする。

実装時の注意: 実装上は、有効な COOKIE ECHO チャンクの受信を SCTP ユーザへ通知する Communication Up を送ることを選択する。

E) COOKIE ACK の受信時に、エンドポイント“ A” は COOKIE-ECHOED 状態から ESTABLISHED 状態へ移行し、T1-cookie タイマを止める。あるいは、Communication Up 通知で ULP にアソシエーションの確立成功を通知してもよい (2.10 節を参照)。

INIT または INIT ACK チャンクは、他のチャンクをバンドルしてはならない。INIT または INIT ACK チャンクは、これらを運ぶ SCTP パケット中の唯一のチャンクとする。

エンドポイントは、INIT を受信した相手の IP アドレスへ INIT ACK を送る。

注: T1-init タイマおよび T1-cookie タイマは、2.6.3 節の規則に従う。

エンドポイントが INIT, INIT ACK あるいは COOKIE ECHO チャンクを受け取りはしたが、受信した INIT や INIT ACK の中の必須なパラメタがない場合、無効な値の場合、ローカル資源が不足する場合で、新しいアソシエーションを確立しないことを決定する場合、ABORT チャンクで答える。そして、ABORT チャンクのエラー要因パラメタに、中断の原因 (必須なパラメタ不足等のタイプ) を含めて指定することを推奨する。ABORT チャンクを含んでいる送信 SCTP パケットの共通ヘッダ中の照合タグフィールドは、相手の開始タグ値をセットする。

アソシエーション中の最初の DATA チャンクを受理後に、エンドポイントは、DATA チャンクを承認するために SACK で直ちに応答する。2.6.2 節に記述されるように、後続の承認が行われるべきである。

TCB が作成される際、各エンドポイントは、その送信された初期 TSN マイナス 1 の値を Cumulative TSN Ack Point にセットする。

実装時の注意: IP アドレスおよび SCTP ポートは、SCTP インスタンス中の TCB を見つけるためのキーとして、一般に使用されます。

2.5.1.1 ストリームパラメタの扱い

INIT および INIT ACK チャンク中で、チャンクの送信者は、アソシエーション中に保持したい送信ストリーム (OS) の数を指示する。他のエンドポイントから受信する最大受信ストリーム数 (MIS) も同様である。

他者からストリーム構成情報を得た後に、各エンドポイントは以下のチェックを実行する。相手側の MIS が自エンドポイントの OS より小さい場合、相手が送信ストリームをすべてはサポートできないことを意味している為、エンドポイントは、送信ストリームに MIS を利用するか、アソシエーションを異常終了して上位レイヤへ相手側のリソース不足を報告する。

アソシエーションが初期化された後、どちらかのエンドポイントのための有効な送信ストリームの識別子の範囲は、0 から min(自分の OS, 相手の MIS) – 1 の間とする。

2.5.1.2 アドレスパラメタの扱い

アソシエーション初期化中に、エンドポイントは、相手への送信先アドレス (複数の場合もある) を検出して収集するために、以下の規則を使用する。

A) 受信した INIT もしくは INIT ACK チャンクの中にアドレスパラメタがない場合、エンドポイントはチャンクを送ってきた送信元の IP アドレスを、SCTP の送信元ポート番号と関連付け、これを唯一の送信先アドレスとして記録する。

B) 受信した INIT もしくは INIT ACK チャンクの中に、Host Name パラメタがある場合、エンドポイントはホスト名から IP アドレス (複数の場合がある) のリストを決定し、SCTP 送信元ポートと解決された IP アドレスの組み合わせによって、相手への送信先アドレスを得る。

エンドポイントは、IP アドレスパラメタが受信 INIT や INIT ACK チャンクで与えられた場合、他のすべての IP アドレスパラメタを無視する。

INIT の受信者がホスト名を解決する時、SCTP に潜在的なセキュリティ問題を含んでいる。INIT の受信者がチャンクの受理時にホスト名を解決する場合、そのホスト名を解決するために受信者が使用するメカニズムが長時間の遅延を伴う可能性がある(例: DNS 検索)。受信者は、状態クッキーを構築しローカルの資源を解放する前で、ホスト名解決の結果を待っている間、自身をオープンにしてリソース攻撃を受ける可能性がある。

したがって、名前翻訳が長時間遅延を伴う可能性がある場合、INIT の受信者は、相手から COOKIE ECHO チャンクを受信するまで、名前解決を延期する。そのような場合、INIT の受信者は、受信した Host Name を (送信先アドレスの代わりに) 使用して状態クッキーを構築し、INIT の送信元の IP アドレスに INIT ACK を送ることを推奨する。

INIT ACK の受信者は、チャンクの受理時に名前解決を常に直ちに試みる。

ホスト名解決が成功するまで、INIT あるいは INIT ACK の受信者は、相手へユーザデータ (piggy-backed も stand-alone も) を送ってはならない。

名前解決が成功しなかった場合、エンドポイントは直ちに “Unresolvable Address” エラーで ABORT を相手に送る。その ABORT は、最後のパケットを送ってきた相手の送信元 IP アドレスへ送信する。

C) 受信した INIT や INIT ACK チャンクの中に、IPv4/IPv6 のどちらかのみのアドレスだけがある場合、受信者は、受信チャンクから送信先アドレス(複数の場合がある)と、INIT もしくは INIT ACK を送信してきた送信元 IP アドレスをすべて引き出して記録する。送信先アドレスは、SCTP 送信元ポート (共通ヘッダから取得)、INIT や INIT ACK チャンクによって運ばれる IP アドレスパラメタ (複数の場合あり)、IP データグラムの送信元 IP アドレスの組み合わせから得られる。相手へ後続のパケットを送る場合、受信者は送信先アドレスとしてこれらのアドレスのみを使うことを推奨する。

実装時の注意: ある場合 (例えば実装が送信のために使用される送信元 IP アドレスを制御しない場合)、エンドポイントは INIT や INIT ACK の中に、送信される可能性があるすべての IP アドレスを含める必要がある。

結局、送信先アドレスは上記の規則を使用して、INIT か INIT ACK チャンクから引き出され、エンドポイントは最初のパスとして送信先アドレスのうちの 1 つを選択する。

注: INIT-ACK は INIT の送信元アドレスへ送られる。

受理可能なアドレスタイプを示すために、INIT の送信側は `Supported Address Types' パラメタを含めている場合がある。このパラメタが存在する場合、INIT の受信者 (初期化される側＝initiatee) は、INIT に応答するときに、`Supported Address Types' パラメタの中で示されたアドレス型のうちの 1 つを使用するか、相手によって示されたアドレス型のうちのどれも利用したくないか、もしくは使用できない場合、"Unresolvable Address" エラーでアソシエーションを異常終了させる。

実装時の注意: INIT ACK の受信者が、サポートされない型のためにアドレスパラメタの解決に失敗した時、初期化プロセスを異常終了させて、次の新しい INIT の中でどの型のアドレスを選ぶかを示すために `Supported Address Types' パラメタを使用し、再度初期化を試みることができる。

2.5.1.3 状態クッキーの生成

INIT チャンクに応答して INIT ACK を送る場合、INIT ACK の送信者は状態クッキーを生成し、INIT ACK の状態クッキーパラメタの中でそれを送る。この状態クッキーの内部に、送信者は、MAC (例として [RFC2104]を参照)、状態クッキーが作成された時のタイムスタンプ、状態クッキーの寿命を、アソシエーションを確立するのに必要なすべての情報と共に、含める必要がある。

次のステップが状態クッキーを生成するために採用される:

1) 受信した INIT および送出する INIT ACK チャンクの両方からの情報を使用して、アソシエーション TCB を生成する。
2) TCB には、生成時間を現在の時刻へ設定し、寿命をプロトコルパラメタ `Valid.Cookie.Life' へ設定する。
3) TCB から、TCB を再構築するために必要な最小限の情報を識別して集め、この情報の一部を使用して MAC と秘密キーを生成する。(MAC の生成の例に関しては[RFC2104]を参照)
4) これらの情報の一部と生成された MAC を組み合わせ、状態クッキーを生成する。

状態クッキーパラメタを持った INIT ACK を送った後に、送信者は、リソース攻撃を防ぐために新しいアソシエーションに関係する TCB、および他のローカルのリソースを削除することを推奨する。

MAC を生成するために使用されるハッシュ手法は、厳密には、INIT チャンクの受信者が個別に扱うべき問題である。MAC の使用はサービスを妨害する攻撃を防ぐために必須である。秘密キーは乱数であることをと([RFC1750]は、乱数性のガイドラインについて情報を提供している)、適切な頻度で変更されることを推奨する。また、状態クッキーのタイムスタンプを、MAC を確認するために使用するキーを決めるために使用してもよい。

実装上は、相互運用性を保証するために、クッキーはできるだけ小さく作ることを推奨する。

2.5.1.4 状態クッキーの処理

(COOKIE WAIT 状態にある) エンドポイントが状態クッキーパラメタを持った INIT ACK チャンクを受け取った場合、直ちに状態クッキーの送信元へ COOKIE ECHO チャンクを送る。送信者は、COOKIE ECHO チャンクの後に続けて、パケット中に任意のペンディング状態の DATA チャンクを加えてもよい。

エンドポイントは、さらに COOKIE ECHO チャンクを発送した後に T1-cookie タイマを起動する。タイマが満了した場合、エンドポイントは COOKIE ECHO チャンクを再送信し、T1-cookie タイマを再起動する。これは、COOKIE ACK を受信するか、`Max.Init.Retransmits' に到達するまで繰り返される。後者の場合、相手のエンドポイントは到達不能と記録される (アソシエーションは CLOSED 状態に入る)。

2.5.1.5 状態クッキーの認証

エンドポイントが、アソシエーションを持っていない別のエンドポイントから COOKIE ECHO チャンクを

受け取る場合、以下の動作を行う:

1) 状態クッキーで送られた TCB データと秘密キー(状態クッキー中のタイムスタンプがどの秘密キーを使うかを決定するために使われることに注意) を使用して MAC を算出する。参考文献[RFC2104]は MAC の生成のためのガイドラインとして使用することができる。
2) 以前に算出された MAC と、状態クッキーによって送られた MAC を比較することにより、状態クッキーを認証する。この比較が失敗したとき、COOKIE ECHO および任意の DATA チャンクを含む SCTP パケットは、暗黙のうちに廃棄されることを推奨する。
3) 現在の時間と状態クッキー中の生成時間 (タイムスタンプ) を比較する。経過時間が状態クッキーによって送られた寿命より長い場合、COOKIE ECHO と添付された任意の DATA チャンクを含んだパケットは廃棄されるべきであり、エンドポイントは "失効クッキー" エラーを付与した ERROR チャンクを相手に送信する。
4) 状態クッキーが有効な場合、COOKIE ECHO の送信者との間で、送られてきた TCB データ内の情報を使ってアソシエーションを生成し、ESTABLISHED の状態に入る。
5) COOKIE ECHO の受理を承認する相手へ COOKIE ACK チャンクを送る。COOKIE ACK は、送信する DATA チャンクあるいは SACK チャンクとバンドルしてもよい。しかし、COOKIE ACK は SCTP パケット中の最初のチャンクとする。
6) SACK を持つ COOKIE ECHO とバンドルした DATA チャンクを即座に承認する (後続の DATA チャンクの承認は、2.6.2 節の中で定義された規則に従う)。ステップ 5) で言及したように、SACK が COOKIE ACK とバンドルされる場合、COOKIE ACK は SCTP パケットの最初に現われる。

COOKIE ECHO が、既存のアソシエーションを持つエンドポイントから届く場合は、2.5.2 節の手続きに従う。

2.5.1.6 正常なアソシエーション確立の例

以下の例では、"A" からアソシエーションを開始し、次にユーザメッセージを "Z" に送り、"Z"が 2 つのユーザメッセージを "A" に後で送っている (バンドリングと分割は起きていないとしている):

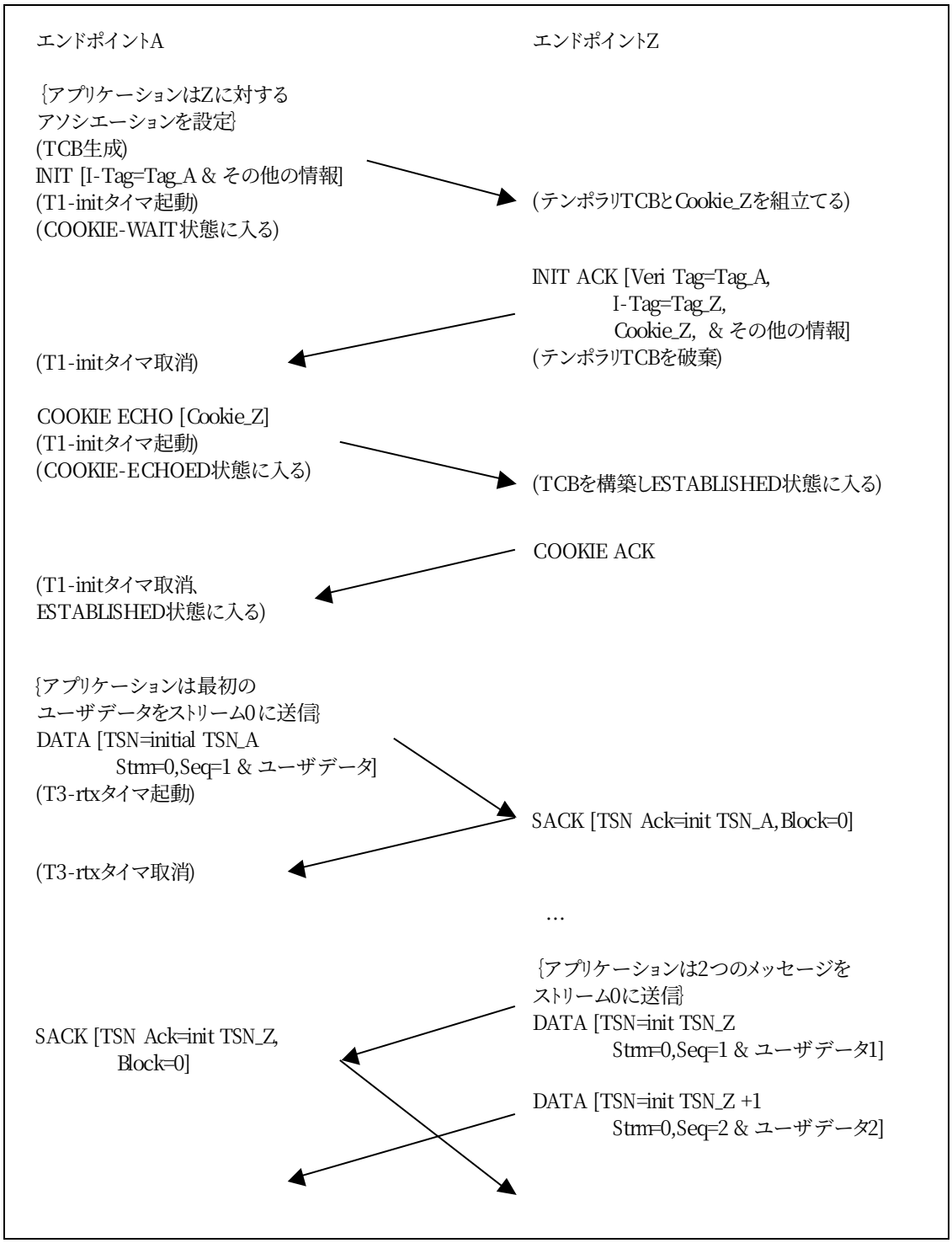


図 2-4 アソシエーション初期化の例

INIT か COOKIE ECHO チャンクを送った後、T1-init タイマが“A”で満了となったとき、同じ開始タグ (即ち Tag_A) もしくは状態クッキーを持った、同じ INIT か COOKIE ECHO チャンクが再送され、タイマが再起動する。これは Max.Init.Retransmit 回まで繰り返され、その後“A”は“Z”が到達不能であると考え、上位レイヤに失敗を通知する (アソシエーションは CLOSED 状態に”入る)。INIT を再送するとき、エンドポイントは、2．6．3 節の中で定義された規則に従って、適切なタイマ値を決定しなければならない。

2.5.2 重複の扱いと予期しない INIT, INIT ACK, COOKIE ECHO, COOKIE ACK

(ありえる状態のうちの 1 つにある) アソシエーションのライフタイムの間に、エンドポイントは、相手のエンドポイントから、セットアップチャンク(INIT, INIT ACK, COOKIE ECHO および COOKIE ACK) のうちの 1 つを受け取る。受信者は、2 重に生成されたようなセットアップチャンクを、このセクションに示すように処理する。

注: チャンクが SCTP 送信先アドレスへ送られず、このエンドポイントと関係のある SCTP 送信先アドレスから送られたのでもなければ、エンドポイントはそのチャンクを受け取らない。つまり、エンドポイントはチャンクを現在のアソシエーションの一部として処理する。

以下のシナリオは重複したチャンクあるいは予期しないチャンクを生じさせる:

A) 相手が検知されずにクラッシュし、アソシエーションを回復しようとして自らリスタートし、新しい INIT チャンクを送出した
B) 両エンドポイントが、ほぼ同時にアソシエーションを初期化しようとした
C) チャンクが、現存するアソシエーションや既に存在しない過去のアソシエーションを確立するために使用された、古いパケットからのものである
D) チャンクが、攻撃者によって生成された正しくないパケットである
E) 相手が COOKIE ACK を受け取らず、COOKIE ECHO を再送信しつづける

以下のセクション中の規則は、これらのケースを識別し、正確に扱うために適用されるものである。

2.5.2.1 COOKIE-WAIT または COOKIE-ECHOED 状態における INIT 受信 (Item B)

これは通常、初期化時の競合を示している。つまり、あるエンドポイントとほぼ同時に、他のエンドポイントがアソシエーションを確立しようと試みている。

COOKIE-WAIT あるいは COOKIE-ECHOED 状態に INIT を受信すると、エンドポイントは、オリジナルの INIT チャンクの中で送ったものと同じパラメタ (開始タグを含み、変更なし) を使用して INIT ACK で応答する。これらの元のパラメタは、新しく受信した INIT チャンクからのものと結合される。エンドポイントは、さらに INIT ACK を備えた状態クッキーを生成する。エンドポイントは、状態クッキーを計算するために、INIT で送られたパラメタを使用する。

その後、エンドポイントは状態を変更させてはならない。T1-init タイマは走らせたままにしておき、対応する TCB は破壊されてはならない。TCB が存在する場合、状態クッキーを扱うための正規の手続きは、1 つのアソシエーションへの INIT の重複を解決することになる。

COOKIE-ECHOED 状態にあるエンドポイントのためには、自身と相手のタグ情報を備えた Tie タグを持つ (Tie タグの記述に関しては 2.5.2.2 節を参照)。

2.5.2.2 CLOSED, COOKIE-ECHOED, COOKIE-WAIT, SHUTDOWN-ACK-SENT 以外の状態における予期しない INIT

他の方法の記述がなければ、このアソシエーションに対して予期しない INIT を受理する際、エンドポイントは状態クッキーを持つ INIT ACK を生成する。送信する INIT ACK の中で、エンドポイントは現在のベリフィケーションタグと相手のベリフィケーションタグを、状態クッキーの中の予約された場所にコピーする。これらの位置を Peer's-Tie-Tag および Local-Tie-Tag と呼ぶこととする。INIT ACK を含んでいる送信 SCTP パケットは、予期しない INIT 中の開始タグと等しいベリフィケーションタグの値を送る。また、INIT ACK は新しい開始タグ (ランダムに生成されたもの。2.5.3.1 節を参照) を含んでいる。エンドポイント用の他のパラメタは、アソシエーションの既存のパラメタ (例えば、送信するストリームの数) から INIT ACK とクッキーにコピーされることを推奨する。

INIT ACK を送信した後は、エンドポイントは以降のアクションを行わない。即ち、存在するアソシエー

ション、現在の状態も含めて、対応する TCB は変更してはならない。

注: TCB が存在し、アソシエーションが COOKIE-WAIT 状態でない場合に限り、Tie タグが存在する。正常なアソシエーション INIT (即ち、エンドポイントが COOKIE-WAIT 状態) のために、Tie タグを 0 にセットする (これより前の TCB が存在していないことを示している)。INIT ACK および状態クッキーは 2.5.2.1 節に仕様が記述されている。

2.5.2.3 予期しない INIT ACK

COOKIE-WAIT 状態でないエンドポイントが INIT ACK を受信した場合、エンドポイントは INIT ACK チャンクを廃棄することを推奨する。予期しない INIT ACK は、通常古いかあるいは重複した INIT チャンクの処理を示す。

2.5.2.4 TCB が存在する場合の COOKIE ECHO の扱い

アソシエーションが存在している(つまり CLOSED の状態ではない)状態にてエンドポイントが COOKIE ECHO チャンクを受信した場合、以下の規則が適用される:

1) 2.5.1.5 節のステップ 1 に記述されるているように MAC を計算する。
2) 2.5.1.5 節のステップ 2 に記述されているように状態クッキーを認証する(これは上記のケース C あるいは D である)。
3) 状態クッキー中のタイムスタンプと現在の時間を比較する。状態クッキーが、状態クッキーによって送られた寿命の値より古く、状態クッキーに含まれているベリフィケーションタグが現在のアソシエーションのベリフィケーションタグと一致しない場合、COOKIE ECHO と DATA チャンクを含むパケットは廃棄される。これに加え、エンドポイントは、失効クッキーエラー原因を含む ERROR チャンクを相手エンドポイントに送信する(これは 2.5.2 節のケース C あるいは D である)。
   状態クッキー中の両方のベリフィケーションタグが現在のアソシエーションのベリフィケーションタグと一致する場合(これは 2.5.2 節のケース E である)、寿命が過ぎていても状態クッキーが有効であると考える。
4) 状態クッキーが有効であると判明した場合、TCB を一時的な TCB 中に取り出す。
5) 実行すべき正しいアクションを決定するために表 2-1 を参照せよ。

表 2-1 TCB が存在する時の COOKIE ECHO の扱い

Local Tag	Peer's Tag	Local-Tie-Tag	Peer's-Tie-Tag	Action/Description
X	X	M	M	(A)
M	X	A	A	(B)
M	0	A	A	(B)
X	M	0	0	(C)
M	M	A	A	(D)

凡例:

X タグが既存の TCB と一致しない

M　　タグが既存の TCB と一致する

0　　Tie タグが Cookie の中にない(未知)

A　　すべてのケース、つまり M、X あるいは 0

注:表 2-1 中で示されない任意のケースについては、クッキーが暗黙のうちに廃棄される。

アクション

(A) この場合、相手は再開している。エンドポイントが「再開」と認識した場合、既存のセッションは、ABORT に続いて新たに COOKIE ECHO を受信したかのように扱われる。ただし、次の例外を除く:

- 任意の SCTP DATA チャンクは保持される(これは実装依存のオプションである)。
- RESTART を、“COMMUNICATION LOST”の代わりに、ULP に通知することを推奨する。

この相手に関連付けられた全ての輻輳制御パラメタ(例えば、cwnd、ssthresh)は初期値にリセットされる(2.6.2.1 節を参照)。

この後に、エンドポイントは ESTABLISHED 状態に入る。

エンドポイントが SHUTDOWN-ACK-SENT 状態で、相手が再開したと認識した場合(アクション A)、新しいアソシエーションをセットアップしない。しかしその代わりに、SHUTDOWN ACK を再送し、“Cookie Received while Shutting Down”エラー原因を含む ERROR チャンクを相手に送る。

(B) この場合、両エンドポイントは、ほぼ同時にアソシエーションをスタートしようとしているが、相手エンドポイントは、こちらから送信した INIT に応答した後に INIT を開始した。従って、エンドポイントに以前送信したベリフィケーションタグに気付かずに新しい Tag を選んでいる。エンドポイントは ESTABLISHED 状態に留まるか、ESTABLISHED 状態に入ることを推奨する。しかし、エンドポイントは、状態クッキーから相手のベリフィケーションタグを更新し、すべての起動中の init あるいはクッキータイマを止め、COOKIE ACK を送る。

(C) この場合、エンドポイントの cookie が遅れて到着した。この cookie が到着する前に、エンドポイントは INIT を送り、INIT-ACK を受け取り、最終的に相手と同じだが自身にとっては新しいタグを含む COOKIE ECHO を送信した。エンドポイントの cookie は暗黙のうちに廃棄される。エンドポイントは状態を変えず、動作中のタイマは走らせたままにしておくことを推奨する。

(D) ローカルのタグと相手のタグが一致する場合、エンドポイントは(まだそうでないなら)常に ESTABLISHED 状態に入る。すべての動作中の init あるいはクッキータイマを止め、COOKIE ACK を送る。

注: 相手のベリフィケーションタグは、受信した INIT か INIT ACK チャンクの中の開始タグフィールド中のタグである。

2.5.2.4.1 アソシエーション再起動の例

以下の例において、“A”は再起動が生じた後にアソシエーションを開始する。エンドポイント“Z”は、パケットのやり取りを行うまで再始動を認識することができない(つまり、ハートビートにより“A”の障害をまだ検出していない)。 (バンドリングあるいは分割が行われていないと仮定する):

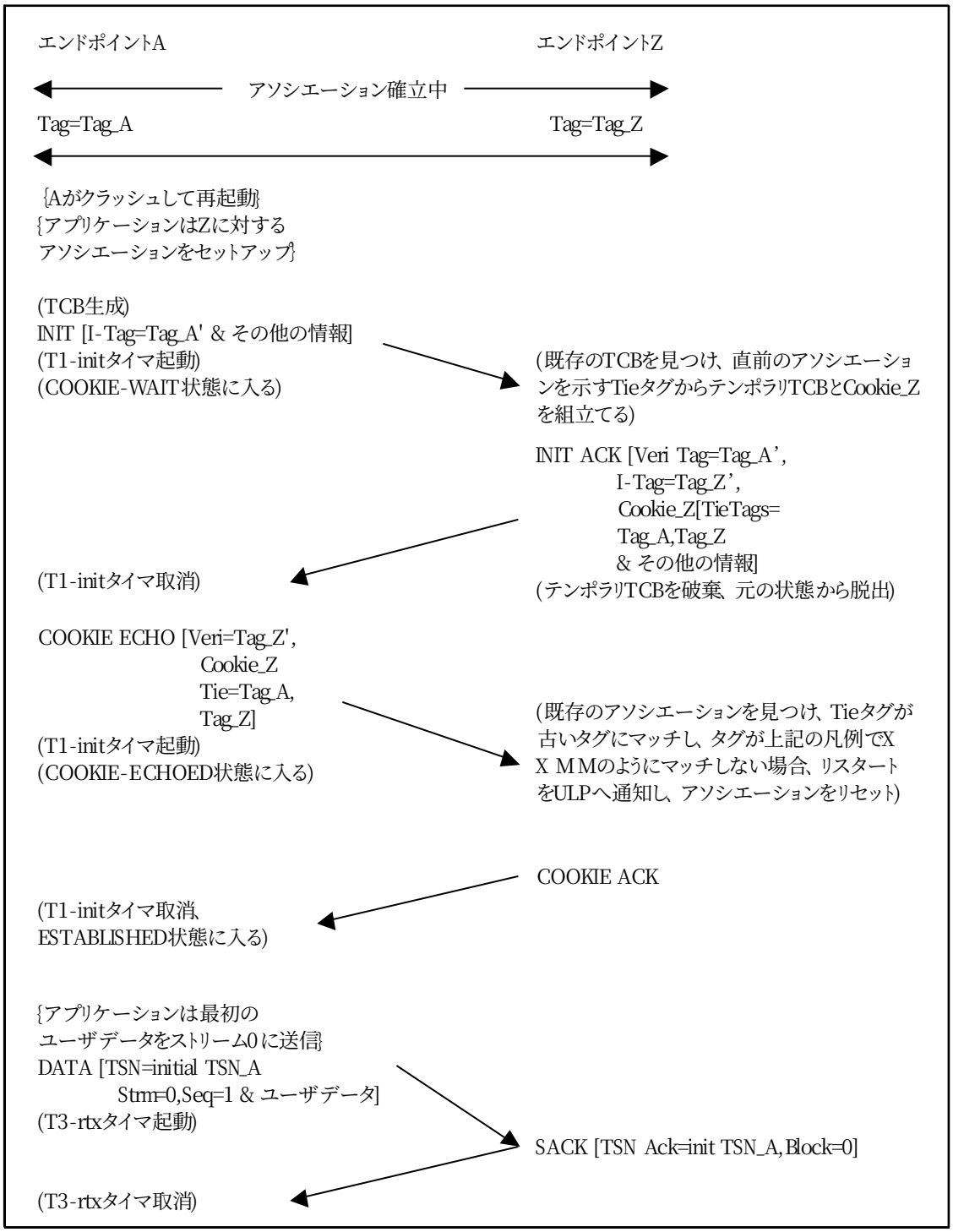


図 2-5 アソシエーション再起動の例

2.5.2.5 **重複した** COOKIE-ACK **の扱い**

COOKIE-ECHOED **以外の任意の状態で** COOKIE ACK **チャンクを受信した場合、エンドポイントは暗黙のうちに受信した** COOKIE ACK **チャンクを廃棄することを推奨する。**

2.5.2.6 Stale COOKIE **エラーの扱い**

失効クッキーエラー原因を含む ERROR **チャンクを受信した場合、そのチャンクは以下に示す取り得るイベントの中の一つを示す**:

A) 送信者によって送出された状態クッキーが処理される前に、アソシエーションのセットアップを完了できなかった。

B) セットアップが完了した後に、古い状態クッキーが処理された。

C) 古い状態クッキーが、アソシエーションに関係のない相手から届き、かつ ABORT チャンクが失われた。

失効クッキーエラー原因を含む ERROR チャンクを処理する場合、エンドポイントは、最初にアソシエーションがセットアップ中であるか(つまり、アソシエーションが COOKIE-ECHOED 状態であるか)を検査することを推奨する。アソシエーションが COOKIE-ECHOED 状態でなければ、ERROR チャンクは暗黙のうちに廃棄される。

アソシエーションが COOKIE-ECHOED 状態である場合、エンドポイントは以下の 3 つの選択肢から 1 つを選ぶ。

1) 新しい状態クッキーを生成するためにエンドポイントに新しい INIT チャンクを送り、再度セットアップ手続きを試みる。

2) TCB を破棄し、上位レイヤへアソシエーションのセットアップ不可を報告する。

3) 新しい INIT チャンクをエンドポイントに送る。このチャンクに Cookie Preservative パラメタを加えることにより状態クッキーのライフタイムの延長を要求する。延長時間を計算する際、実装上は前回の COOKIE ECHO / ERROR 交換時に測定された RTT 情報を使用することを推奨する。この場合、計測された RTT に 1 秒以上を加えないことを推奨する。これは、クッキーライフタイムを長くした場合、エンドポイントがリプレイ攻撃の対象になるからである。

2.5.3 その他の初期化問題

2.5.3.1 Tag Value の選択

開始タグ値は、$2^{32}-1$ の範囲で選ばれることを推奨する。 開始タグ値の乱数化は、“man in the middle” や “sequence number” 攻撃から防御するために非常に重要である。開始タグの乱数化のために[RFC1750]に記述された方法を利用できる。開始タグ値を注意深く選択することは、古いアソシエーションに属する重複したパケットを現在のアソシエーションに属するパケットとして誤って処理してしまうことを防ぐためにも必要である。

さらに、アソシエーションのどちらかのエンドポイントで使われるベリフィケーションタグ値はアソシエーションの生存期間中は変更されてはならない。エンドポイントがアソシエーションを終了させその後、同じエンドポイントとの間にアソシエーションを再構築する毎に新しいベリフィケーションタグ値が使われる。

2.6 ユーザデータ転送

DATA チャンクは、ESTABLISHED、SHUTDOWN-PENDING および SHUTDOWN-RECEIVED 状態でのみ送信可能である。ただし、COOKIE-WAIT 状態であれば、DATA チャンクを COOKIE ECHO チャンクとバンドルしてもよい。

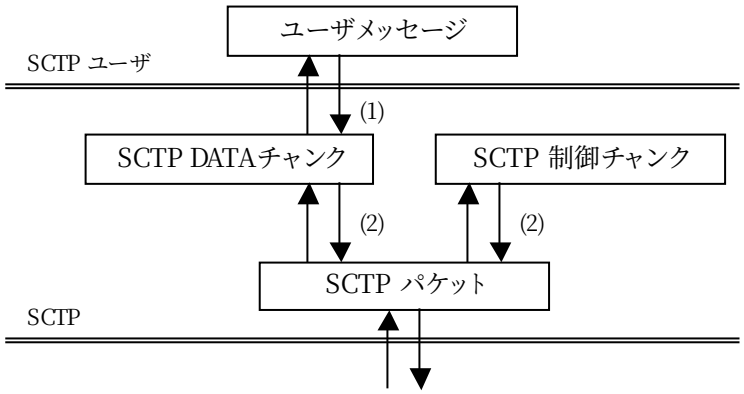
DATA チャンクは、ESTABLISHED, SHUTDOWN-PENDING, SHUTDOWN-SENT 状態でのみ受信可能である。CLOSED 状態で受信した DATA チャンクは、OOTB パケットであり、2.8.4 節に従って扱われることを推奨する。他の状態で受信した DATA チャンクは廃棄するを推奨する。

SACK チャンクは、ESTABLISHED, SHUTDOWN-PENDING および SHUTDOWN-RECEIVED 状態で受信可能である。COOKIE-ECHOED 状態でも受信してもよい。CLOSED 状態で受信した SACK チャンクは、OOTB パケットであり、2.8.4 節に従って扱われることを推奨する。他の状態で受信した SACK チャンクは廃棄することを推奨する。

SCTP は、最低でも 1,500 バイトの SCTP パケットを受信できなければならない。したがって、INIT チャンクと INIT ACK チャンクの a_rwnd パラメタには、1,500 バイト未満の値を設定してはいけない。

送信効率化のため、SCTP は、小さなユーザメッセージのバンドルメカニズムと、大きなユーザメッセージの分割メカニズムを有している。図 2-6 は、ユーザデータ転送フローを示す。

本節では、DATA チャンクを送信するエンドポイントを「データ送信者」、DATA チャンクを受信するエンドポイントを 「データ受信者」と呼ぶ。「データ受信者」は SACK チャンクを送信する。



注
1) データ送信者は、アソシエーションのパスMTUより大きなユーザメッセージを、複数のDATAチャンクへ分割する。データ受信者は、複数のDATAチャンクからユーザメッセージを再結合し、SCTPユーザへ引き渡す(詳細は 2.6.9 節を参照)。
2) データ送信者は、SCTPパケットがパスMTUを超過しない限り、DATAチャンクと制御チャンクを一つのSCTPパケットにまとめてもよいが、制御チャンクは必ずDATAチャンクの前に来る(バンドル)。データ受信者は、SCTPパケットからそれぞれのチャンクを取り出す(アンバンドル)。

図 2-6 ユーザデータ転送フロー図

分割メカニズムとバンドルメカニズムの詳細を、それぞれ 2.6.9 節と 2.6.10 節に示す。データ送信者は分割メカニズムとバンドルメカニズムを実装しなくてもよいが、データ受信者は必ず実装しなければならない。

2.6.1 DATA チャンクの送信

データ送信者は送信先トランスポートアドレス毎に再送監視タイマ(T3-rtx タイマ)を持つ。しかし、実装上は DATA チャンク毎に再送監視タイマを備えてもよい。

DATA チャンクの送信および再送信は以下の規則に従う：

A)送信先トランスポートアドレスの rwnd が 0 の場合、データ送信者は新たな DATA チャンクを送信してはならない。ただし、cwnd を超えない DATA チャンクであれば 1 つだけを送信してもよい。SACK チャンクを紛失した場合でも、データ送信者が rwnd の変化を知るためであるを

B)送達未確認のデータが cwnd 以上の時、新しい DATA チャンクは送信してはならない。

C)新しい DATA チャンクを送信する前に再送出タイマの時間切れが生じた場合は、送信者はそのタイマが対象にしている DATA チャンクを送信しなければならない。

D)送信者は、上記 A)および B)のルールが許す限りの DATA チャンクを送信できる。

DATA チャンクはバンドルできるが、再送信する DATA チャンクと新しい DATA チャンクのバンドルもできる。ただし、総和が MTU を超えないこと。ULP(upper layer protocol)がバンドルを行わないように指示することもできるが、SCTP がバンドリングすることに起因するディレイを無くす(turn off)する時だけにする

ことを推奨する。SCTP 自身でバンドリングを行わないようする契機はない。

新しい DATA チャンクを送信する前に、ACK を送っていないものがあったら、SACK チャンクを作ってバンドルする。

送信時および再送信時に T3-rtx タイマが起動していなければ起動する。T3-rtx タイマが起動していれば、再送信時にタイマを再起動する。他の場合にタイマを再起動してはならない。

送信時および再送信時の T3-rtx タイマ値の選択は 2.6.3.2 節と 2.6.3.3 節に従う

2.6.2 DATA チャンク受領時の確認

SCTP エンドポイントは常に個々の正常な DATA チャンクの受信を確認せねばならない。

[RFC2581]の 4.2 節で規定する遅延確認アルゴリズムのガイドラインに従わなければならない。具体的に言うと、少なくとも第二のパケット受信ごとに(第二の DATA チャンクではなく）確認が生成されなければならなず、また確認されていないデータチャンクの到着から 200 ミリ秒以内に生成しなければならない。

ある状況では、本ドキュメントで詳細に説明するアルゴリズムよりも更に保守的に振る舞う方が SCTP 送信者にとって有益なこともある。しかしながら、SCTP 送信者は以下のアルゴリズムが許容する以上に攻撃的に振る舞うべきではない。

SCTP 受信者は、受信側アプリケーションの新データ消費に応じて、提供されたウインドウをアップデートする目的以外では、各受信パケットに対して一つ以上の SACK を生成するべきではない。

実装上の注意：確認生成における遅延の最大値は、伝達するプロトコルの要求条件を満たすために、SCTP 管理者が静的または動的にコンフィギュレーションしても良い。この場合、遅延最大値が 500ms 以上に設定できるように実装してはならない。言い換えれば、500ms 以下に設定できるようにじっそうしても良いが、500ms 以上に上げてはならない。

確認は SACK チャンクの中で送信しなければならない。ULP により shutdown が要求された場合は、エンドポイントが SHUTDOWN チャンクの中で確認を送っても良い。複数 DATA チャンクの受信を一つの SACK チャンクで確認しても良い。SACK チャンクのフォーマットは 3.3.4 節を参照。特に、SCTP 終端点は、受信した正規の Data チャンクの最新順次 TSN を Cumulative TSN Ack フィールドへ格納しなければならない。Cumulative TSN Ack フィールドの値よりも大きな TSN を持つ DATA チャンクを受信した場合は Gap Ack Block フィールドで報告しなければならない。

注意：SHUTDOWN チャンクは Gap Ack Block フィールドを持たない。このため、エンドポイントは SHUTDOWN チャンクではなく SACK を用いて順序違反した DATA チャンクを確認しなければならない。

新規でない重複した DATA チャンクを含むパケットの受信時には、エンドポイントは即座に遅延無く SACK を送出しなくてはならない。新規の DATA チャンクを含む重複した DATA チャンクを持つパケットの受信時には、エンドポイントは即座に SACK を出してもよい。通常は、最初の SACK チャンクの紛失によりピア側の RTO が満了した場合に、重複した DATA チャンクの受信が起こる。重複した TSN 番号は SACK で重複したとおりに報告しなくてはならない。

エンドポイントは SACK 受信時に、重複した TSN 情報を使って SACK 紛失を検出してもよい。他の利用方法は今後の検討とする。

データ受信者は受信バッファを維持する責任がある。データ受信者はデータ受信能力の変化を適宜データ送信者へ通知しなくてはならない。受信バッファ管理のインプリメンテーションは多くの要因に依存する。たとえば、OS、メモリ管理システム、メモリ量など。しかしながら、6.2.1 節に定義したデータ送信方法は、受信者が以下の処理を行うことを前提として規定した物である。

A) アソシエーションの初期化時に、エンドポイントは INIT 又は INIT ACK により、ピアへ受信バッファスペースの割当量を通知する。この値は a_rwnd へ設定する。

B) DATA チャンクを受信してバッファする際に、受信してバッファしたバイト数に従って a_rwnd を減分

する。これにより、データ送信者が rwnd を閉めて送信可能なデータ量を制限する。

C) DATA チャンクが ULP へ配達され受信バッファが開放されると、上位レイヤへ配達されたバイト数に従って a-rwnd を加算する。これにより、データ送信者が rwnd を開いてより多くのデータ送出を許容する。データ受信者は受信バッファからバイトを開放しない限り a_rwnd を増加してはいけない。たとえば、受信者が断片化された DATA チャンクを再組み立てキューに保持している場合、a_rwnd を増加すべきではない。

D) SACK を送出する際に、データ受信者は a_rwnd フィールドへ現在の a_rwnd 値を設定しなくてはならない。データ送信者は Cumulative TSN ACK で確認した DATA チャンクを再送することはない（再送キューから削除するため）ことを、データ受信者は考慮しなくてはならない。

Gap Ack Blocks で確認した DATA チャンクに関しては、データ受信者は受信してまだ受信バッファにある DATA チャンクを削除してもよい。たとえば、データ受信者がピアからの断片化されたユーザメッセージを再組み立て中にバッファスペースを使い果たしたとする。受信者はこれらの DATA チャンクを Gap Ack Blocks で確認したとしても、廃棄してよい。データ受信者が DATA チャンクを廃棄した場合、再送によって再受信するまでは後続する SACK の Gap Ack Block へこれらを含めてはいけない。さらに、エンドポイントは削除したデータを考慮して a_rnwd を計算すべきである。

エンドポイントは SACK を取り消してデータを廃棄すべきではない。エンドポイントがこの手順を取るのはバッファスペース枯渇のような極限状況のみとすべきである。データ受信者は、Gap Ack Blocks で確認したデータを廃棄することにより適正でない再送が起こり、パフォーマンスが最適化されないことがあることを考慮すべきである。

以下は、遅延確認の例である。

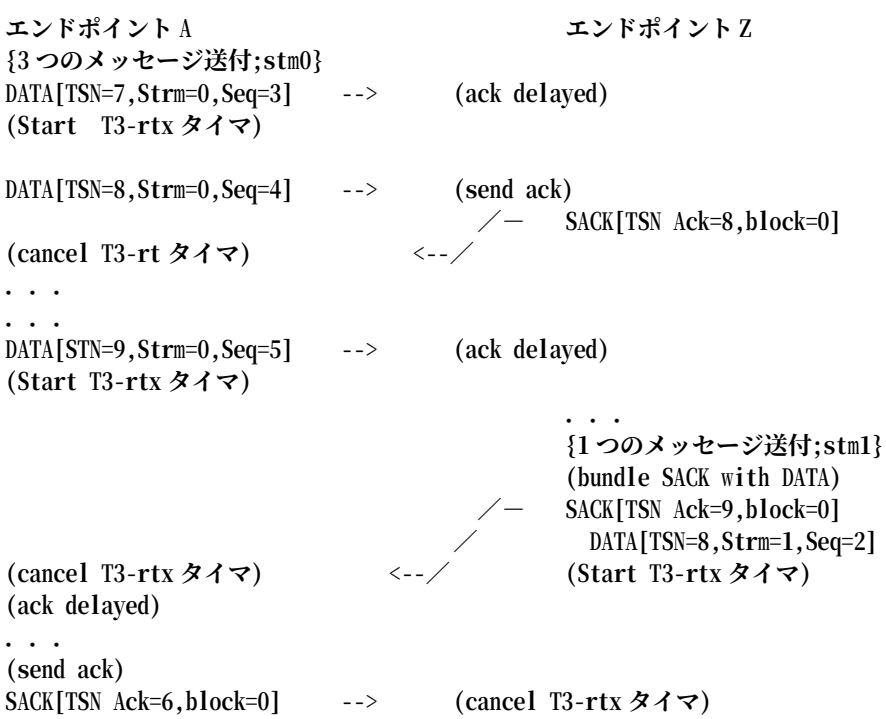


図 2-7 遅延応答の例

エンドポイントがユーザデータの無い DATA チャンクを受信した場合(すなわち、Length フィールドが 16 の場合)、エラー原因"No User Data"を設定した ABORT を送出しなくてはならない。

エンドポイントはユーザデータの無い DATA チャンクを送信しないことを推奨する。

2.6.2.1 SACK 受領時の処理

各 SACK は a_rwnd 値を持っている。この値はデータを受け取る側のバッファサイズを表しており、SACK を転送する際の(INIT/INIT ACK に示されている)全受信バッファの残りを示している。a_rwnd と Cumulative TSN Ack, Gap Ack Blocks を用いて、送信側は相手側のバッファサイズを算出することができる。

データ送信側の問題は SACK を処理する時に順序がずれることが生じうることを考慮にいれておくべき点である。つまり、データ受信側が送る SACK は、これより以前の SACK を追い越すことがあり得、データ送信側はこちらを先に受け取る。もし、順番が狂った状態で SACK を受け取った場合は、データ送信側は相手側の受信バッファサイズについて間違った認識をすることになる。

SACK の順序逆転を検知するためのストリームは無いので、データ送信側は SACK が最新かどうかを自己学習しなくてはならない。

受信側は rwnd を算出するために、受け取った SACK 中の a_rwnd と Cumulative TSN Ack, Gap Ack Blocks を使って、以下のルールを用いるべきである。

A)アソシエーション確立時、ピアが INIT または INIT ACK で指定した Advertised Receiver Window Credit(a_rwnd)で rwnd を初期化する。

B)DATA チャンク送出時(もしくは再送時)、エンドポイントはピア側の rwnd からチャンクのデータサイズ分を減算する。

C)DATA チャンクに再送の印をつける時(T3-rtx タイマ時間切れ(2.6.3.3 節)もしくは、fast 再送(2.7.2.4 節)によって)、これらのチャンクのデータのサイズを rwnd に加算する。

注意：各々の DATA チャンクについてタイマ制御をするような実装をした場合、時間切れになった DATA チャンクのみに再送の印をつける。

D)SACK 受信時、エンドポイントは以下の処理をする。

ｉ）Cumulative TSN Ack が Cumulative TSN Ack Point より小さい場合は、SACK を破棄する。Cumulative TSN Ack は単調に増加するので、Cumulative TSN Ack が Cumulative TSN Ack Point より小さい SACK は順序が狂っている SACK であることを示している。

ii）新しく受け取った a_rwnd から、Cumulative TSN Ack と the Gap Ack Blocks を処理した後に滞留しているバイト数を引いた値を rwnd にセットする。

iii）SACK が、以前の Gap Ack Block によって通知された TSN を持っていないとき(たとえば、データ受信者がデータの規則を犯した場合)、関連する DATA チャンクに再送可能の印をつける。すなわち、2.7.2.4 節に示したとおり、紛失のための即時再送の印を付け、最初に DATA チャンクを送ろうとした先への再送タイマが起動していない場合は、送付先用の T3-rtx を始動させる。

2.6.3 再送出タイマの管理

SCTP エンドポイントは、ピア側からのレスポンスがない場合にも確実にデータを届けるため、再送タイマ T3-rtx を用いる。このタイマの継続期間を RTO(retransmission timeout)と呼ぶ。

ピア側がマルチホームの場合は、ピア側エンドポイントの個々の着トランスポートアドレス毎に別々の RTO を処理する。

SCTP における RTO の計算と管理は、TCP における再送タイマの管理方法に従う。RTO 値を算出するために、エンドポイントは着トランスポートアドレス毎に二つの状態変数を管理する。即ち、SRTT(smoothed round-trip time：正常時の往復時間)と RTTVAR(round-trip time variation：往復時間の変動量)である。

2.6.3.1 RTO の計算

SRTT、RTTVAR、RTO の算出ルールは以下のとおり。

C1) ある着トランスポートアドレスに対して RTT の計測が行われるまでは、RTO にプロトコルパラメタ

'RTO.Initial"を代入する。

C2) 最初に RTT の初期値 R が算出されたら、SRTT に R を、RTTVAR に R/2 を、RTO に SRTT+4*RTTVAR(=3R)を代入する。

SRTT := R

RTTVAR := R/2

RTO := SRTT + 4*RTTVAR

C3)新たな RTT(R')を計測した後に、以下の代入を行う。

RTTVAR := (1 - RTO.Beta) * RTTVAR + RTO.Beta * |SRTT - R'|

SRTT := (1 - RTO.Alpha) * SRTT + RTO.Alpha * R'

注意：RTTVAR を更新するときに用いる SRTT の値は、2 番目の式で更新される前の SRTT である。

計算後に RTO を更新する。

RTO := SRTT + 4*RTTVAR

C4)データ送受信中で、かつ下記ルール C5 の場合を除き、新しい RTT の計測は毎回行われなければならない。さらに、新しい RTT の計測は一つの転送先に対して何度も行うべきではない。これには以下の二つの理由がある。一つ目は、頻繁な計測は実際には効果が無いと思われる為である。二つ目は、計測をより頻繁に行なった場合、SRTT と RTTVAR がほぼ同じ頻度で変化に適合するように、上記ルール C3 における RTO.Alpha と RTO.Beta の値を（何往復ごとに新しい値を反映させるかの観点から）仮に一往復に一回の計測を行い、C3 に示したとおりに RTO.Alpha と RTO.Beta を使うと同様に調整すべきである。
しかし、これらの調整の厳密な意味付けにはさらなる研究が要求される。

C5)Karn のアルゴリズム：RTT の計測は再送されたパケットを用いてなされるべきではない。（応答が最初のパケットに対するものか、採草パケットに対するものかがあいまいであるため。）

C6) RTO.Min 秒より小さい値の RTO が算出された場合は、RTO.Min 値まで切り上げる。この理由は、高い最小値を持たない RTO は不必要なタイムアウトを生じやすいためである。

C7) RTO に設定できる値は RTO.max 以上である。

以下の例外を除き、RTT 計測および他の状態変数を計算するための時間の粒度 G に対する要求は特にない。

G1) RTTVAR を計測した際に RTTVAR=0 のときは、RTTVAR に G 値を設定する。

RTTVAR <- G.

実験によると、時間の粒度が細かい(100 マイクロ秒以下)ほうが、粗いときよりもやや良い結果をもたらす。

2.6.3.2 再送タイマのルール

再送タイマの管理ルールは以下のとおり。

R1) DATA チャンクを送る際は常に(再送も含めて)、送付先アドレスに対して T3-rtx タイマが起動されていない場合は、RTO 後に満了するように設定してタイマを起動する。ここで用いる RTO は、以下の E2 ルールで議論されているように、宛て先に対応する一つ前の T3-rtx タイマ満了値を 2 倍することによって得られる。

R2) あるアドレスに対して送信したすべてのデータが確認された際に、当該アドレスの T3-rtx タイマを切る。

R3)ある着アドレスに対して最若番の停滞 TSN を含む DATA チャンクに対する SACK を受信すると、現在の RTO を用いて再度 T3-rtx タイマを起動する。（同着アドレスに対して停滞中のデータが残っている場合）

R4)以前に Gap Ack Block で確認されたデータに対して TSN が設定されていない SACK を受信した場合は、最初に DATA チャンクが送出された着アドレスに対して T3-rtx を起動する。（既に T3-rtx が開始されていない場合）

以下はタイマに関する様々なルールの使用例である。(受信側で delayed ack を用いる事を前提とする。)

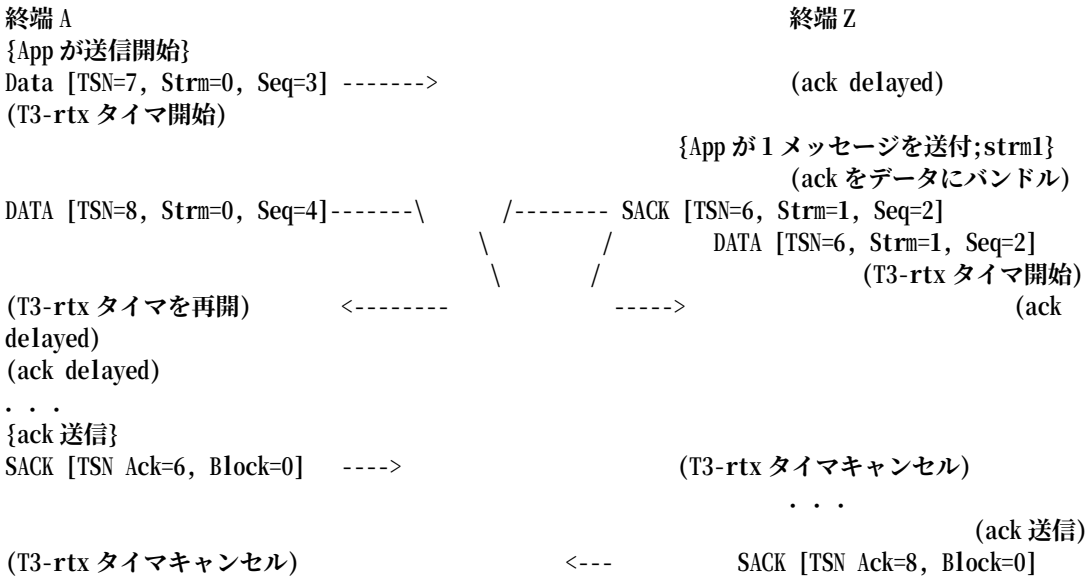


図 2-8 タイマルールの例

2.6.3.3 T3-rtx タイマ満了時の処理

再送タイマ T3-rtx が満了した場合、以下の処理を行う。

E1)タイマが時間切れになった着アドレスに対する ssthresh を 2.7.2.3 節のルールで修正する。また、以下の代入を行う。

Cwnd := MTU

E2) タイマが満了した着アドレスに対応する RTO 値の２倍を新たな RTO 値とする。("back off the timer")

RTO := RTO*2

この 2 倍処理の際に、ルール C7 に記した RTO の最大値(RTO.max)を上限としてもよい。

E3) T3-rtx が満了した着アドレスに対し、1 パケットへ格納可能な TSN が再若番の DATA チャンク個数を、再送先の着アドレスに対する MTU 制限に従って判断する。この個数を K とし、これら K 個の DATA チャンクを、一つのパケットにバンドルして再送する。

E4)ルール R1 に該当する場合は、再送先のアドレス用の T3-rtx タイマを開始する。受信側がマルチホームの場合はタイマ満了時の着アドレスとは異なる着アドレスへ再送する事がある(2.6.4 節参照)ため、設定する RTO 値は再送先アドレスに対応したものとすべきである。

再送後、新規に RTT 測定がされたら(新しいデータが送られ承認された時だけルール C5 によってこの機会が与えられる。もしくは HEARTBEAT によって生じる(2.8.3 節参照)、ルール C3 による計算がなされ、この中には RTO の計算も含まれているが、2 倍する。(ルール E2)。

注意：T3-rtx タイマが時間切れになったアドレスへ送られ、一つの MTU に収まりきれなかった(既出のルール E3)DATA チャンクは、再送のマークがされ、cwnd が許容次第すぐに再送される(通常 SACK が到着した時点)。

再送タイマに関する最後のルールは障害切替に関わるものである(2.6.4.1 節参照)。

F1)現在の宛先トランスポートアドレスが他のアドレスへ切り替わった場合も、再送タイマは継続される。DATA チャンクを格納したパケットを新しいトランスポートアドレスへ送出するとすぐに、ルール R1 に従って再送先トランスポートアドレスに対するタイマを開始する。

2.6.4 マルチホーム SCTP エンドポイント

着アドレスとして利用可能なトランスポートアドレスを複数持つ SCTP エンドポイントをマルチホームと呼ぶ。さらに、終端の上位プロトコル(ULP)は、マルチホームの複数の着アドレスの一つを、第一のパスとして選択するべきである(2.5.1.2 節と 2.10.1 節参照)。

返事のチャンク(例えば、SACK、HEARTBEAT ACK、等)は、同じ宛て先のトランスポートアドレス(DATA チャンクや制御チャンクを受信した時の相手であり、返事をしようとしているところ)へ送付すべきである。このルールは、DATA チャンクが返事のチャンクとバンドルされている場合でも適用されるべきである。

しかし、異なるアドレスから受け取った複数の DATA チャンクの到達確認を一つの SACK で行う場合、その SACK チャンクは、通知すべき DATA や制御チャンクを受け取った宛て先のトランスポートアドレスのうちの一つに送らねばならない。

複製した DATA チャンクの受信側が SACK をマルチホーム側へ送付する時、送付元アドレスを使用せずに送付先を変更するときに便利である。マルチホーム側から複製を受信するのは、SACK のためのリターンパス(DATA チャンクの送信元アドレスに記載されている)が破壊されていることを示している。

なお、相手がマルチホームの場合、DATA チャンクを送付してきた最期の宛て先アドレスとは異なる、アクティブな宛て先のトランスポートアドレスへチャンクの再送を試みるべきである。

再送は総送信数に影響を与えない。しかし、DATA チャンクが異なる宛て先アドレスへ再送された場合は、新しい宛て先アドレスへの総送信数と DATA チャンクを最期に送信した古い宛て先アドレスが、これに従って調整される。

2.6.4.1 活性化していない着アドレスからのフェイルオーバ

マルチホーム SCTP 終端のいくつかのトランスポートアドレスは、エラーの発生(2.8.2 節参照)や SCTP ユーザからの調整によってアクティブでなくなるかもしれない。

送出するべきデータがあり第一のパスがアクティブでない(障害によって)時、もしくは、SCTP ユーザが明示的にアクティブでない送付先を要求している場合、ULP へエラーを通知する前に、SCTP 終端はデータを異なるアクティブな宛て先トランスポートアドレスが存在する場合、送付を試みるべきである。

データ再送時、終端がマルチホームなら、再送のための選択ポリシーを考える時、各送付元アドレスのペアを考慮すべきである。再送時は、終端はパケット送付が生じた送付元－宛て先ペアと異なる送信元－宛て先ペアを採用しようとするべきである。

注意：最も異なる送信元－宛て先ペアを採用するときのルールは実装時の決定事項であり、本ドキュメントには記載されない。

2.6.5 ストリーム識別子とストリーム数列番号

各 DATA チャンクは正しいストリーム識別子を運ばなければならない。もし終端が不正なストリーム識別子の DATA チャンクを受信したなら、通常の処理に従った DATA チャンクの受け取り確認をし、"Invalid Stream Identifier"(Section3.3.10 を参照)と原因を設定した ERROR チャンクを速やかに送付した上で、DATA チャンクを破棄する。エンドポイントは、SACK に続いて ERROR を送出する場合は、エラーチャンクを同じパケットへ格納してもよい。

全てのストリーム中のストリームシーケンス番号は、アソシエーションが確定した時に、0 から始まるべきである。ストリームシーケンス番号が 65535 に到達したときも、次のストリームシーケンス番号は 0 にセットするべきである。

2.6.6 順序転送と非順序転送

ストリームの中で、終端は U フラグが 0 にセットされた DATA チャンクを、上位レイヤへストリームシ

ーケンス番号に従って順番に配送しなければならない。DATA チャンクが順番どおりでなく到着したなら、終端は受け取った DATA チャンクを保持し、順番通りにしてから ULP へ配送するべきである。

しかし、SCTP 終端は、特別な DATA チャンクのために、順番通りでない配送を要求することができ、DATA の U フラグを 1 にしたストリームを転送する。

終端が DATA チャンクを U フラグを 1 にして受け取った時、順番を整列するメカニズムをバイパスし、すぐに上位レイヤへ配送する(ユーザデータが送付側で分割している場合はリアセンブルした後に)。

これは、与えられたストリーム中の「バンドから溢れた」送付の効果的な方法を提供する。単に U フラグを 1 にセットしすることで、「順番通りでない」ストリームとして使うこともできる。

実装時の注意：順番を意識しない DATA チャンクを送付する際、実装は、外部送出パケットに DATA チャンクを置くことができる。また、可能なら、外部送出キューの中の先頭にする。

U フラグを 1 にセットした DATA チャンクのストリームシーケンス番号フィールドは意味がない。送付側は任意の数値を入れることができるが、受信側はこのフィールを無視しなければならない。

注意：順番に意味がある、もしくは、順番を意識しないデータを送付する際、終端は、U フラグを 1 にセットした DATA チャンクを転送したと気に、SSN をインクリメントする。

2.6.7 DATA TSN 受信時のギャップのレポート

新しい DATA チャンクの受理に際して、エンドポイントは TSN の連続性を検査する。エンドポイントが受信 DATA チャンクの並びにギャップを見つけた場合、ギャップ Ack ブロックを含めて、直ちに SACK を送ることを推奨する。データ受信者は、ギャップが埋まっていない SCTP パケットを受信するたびに SACK を送りつづける。

受信した SACK のギャップ Ack ブロックに基づいて、エンドポイントは欠落した DATA チャンクを突きとめ、再送するかどうかを決めることができる(2.6.2.1 節を参照)。

複数のギャップは 1 つの SACK で通知することができる(2.3.3.4 節を参照)。

相手がマルチホームの場合、SCTP エンドポイントは常に最後の DATA チャンクを送信してきた相手に SACK の送付を試みることを推奨する。

SACK の受理に際して、エンドポイントは、累積 TSN Ack によって承認された DATA チャンクを送信キューから削除する。さらに、エンドポイントは SACK によって通知されたギャップ Ack ブロックにより、TSN がない DATA チャンクを、全て「欠落」として扱う。データ送信者は、未解決 DATA チャンクの「欠落」通知数を、再送信決定のために記録する。詳細は 2.7.2.4 節を参照せよ。

以下の例は SACK をギャップ通知に使用することを示す。

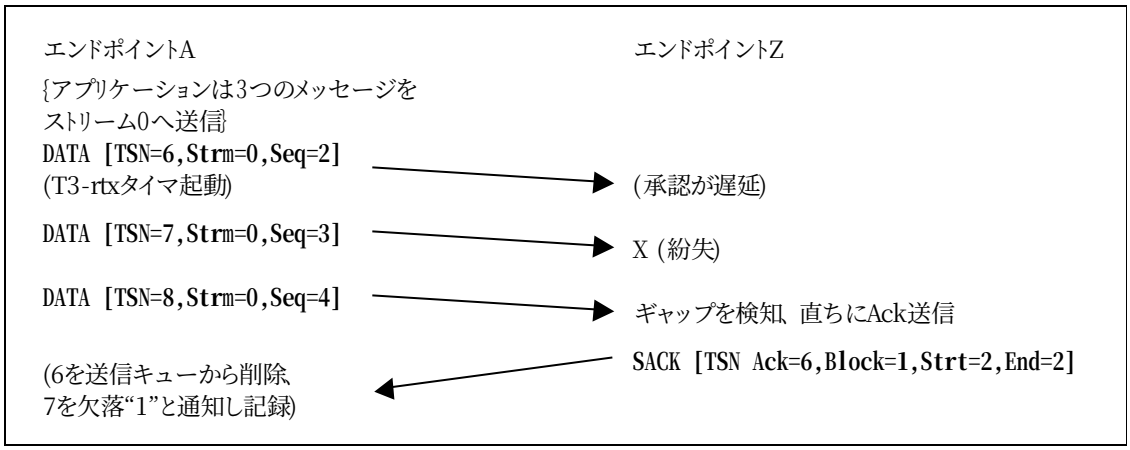


図 2-9 SACK を利用したギャップ通知

1 つの SACK **チャンクで通知できるギャップ** Ack **ブロックの最大値は、その時のパスの** MTU **によって制限される。**MTU **の制限により、**1 **つの** SACK **ですべてのギャップ** Ack **をカバーできない時は、**1 **つの** SACK **しか使ってはならず、ギャップ** Ack **ブロックは小さい** TSN **から大きい** TSN **へ、**MTU **に入る範囲で報告する。そして最大の** TSN **番号に対しては未承認のままにしておく。**

2.6.8 Adler-32 チェックサムの計算

SCTP **パケットを送信するとき、エンドポイントは、以下に記述するように** Adler-32 **チェックサムを含めて、送信データの完全性を高める。**

パケットが構築された後(SCTP **共通ヘッダ、**1 **つ以上の制御チャンクか** DATA **チャンクが含まれている)、昇進者は以下の処理手続きを踏む。**

1) SCTP **共通ヘッダの照合タグに適切な値を使い、チェックサム領域を** 0 **で初期化する。**
2) SCTP **共通ヘッダと全てのチャンクを含めた、パケット全体の** Adler-32 **チェックサムを計算する。**
3) **算出値を共通ヘッダのチェックサム領域に入れる。残りのビットは変更しない。**

SCTP **パケットの受信時、受信者は最初に** Adler-32 **チェックサムを確認する。**

1) **受信した** Adler-32 **チェックサム値を一旦保存する。**
2) **受信した** SCTP **パケットのチェックサム領域の** 32 **ビットを全て** 0 **で置き換え、パケット全体の** Adler-32 **チェックサム値を計算しなおす。**
3) **計算しなおした値が、保存した値と同じかどうかを確認する。違っていたら、受信者は不正な** SCTP **パケットとして扱う。**

不正な SCTP **パケットの通常の扱いは、特になにもせずに(**silently**)破棄することである。**

2.6.9 分割と再組立て

DATA **チャンクを送信する場合に、送信側エンドポイントは分割をサポートしてもよい(必須ではない)が、受信者は再組立てをサポートする。エンドポイントが分割をサポートする場合、ユーザメッセージによって送信する** SCTP **パケットが** MTU **サイズを越える場合は、ユーザメッセージの分割を行う。分割をサポートしない実装の場合は、ユーザメッセージを送信せずに、上位レイヤにエラーを返す。**

実装時の注意: このエラーの場合、2.10.1 **節で解説する** Send **プリミティブは上位レイヤにエラーを返す必要がある。**

送信相手がマルチホームの場合、エンドポイント側はアソシエーションのパス MTU **を越えないようなサイズを選ぶ。パス** MTU **は全ての送付先アドレスの中で最も小さな** MTU **である。**

注: メッセージが一度分割されると、再度分割されることはない。逆に、パス MTU が縮小されていたら、IP 分割が用いられるに違いない。パス MTU の詳細については 2.7.3 節を参照せよ。

分割するかどうかを決定する際、SCTP の実装時には DATA チャンクヘッダと同様に SCTP パケットヘッダも計算する。さらに、DATA チャンクと SACK チャンクをバンドルする場合には、SACK チャンクのために必要なスペースについても計算する。

分割は以下のステップで行われる。

1) データ送信者は、ユーザメッセージを一連の DATA チャンクに分割する。各チャンクに SCTP オーバヘッドを加えても、アソシエーションのパス MTU サイズ以下の IP ダイヤグラムに適合するようにする。
2) 送信者は、一連の各 DATA チャンクに別々の TSN を順番に割当てる。また、各 DATA チャンクに同一の SSN を割当てる。ユーザが、ユーザメッセージを必ずしも順番通りに送信しないことを示す場合は、ユーザメッセージ中の書く DATA チャンクの U フラグを 1 にセットする。
3) 送信者は、一連の DATA チャンクのうち、最初の DATA チャンクの B/E ビットを`10'に、最期の DATA チャンクの B/E ビットを`01'に、他の DATA チャンク全ての B/E ビットを`00'に設定する。

エンドポイントは、各受信 DATA チャンクの B/E ビットを調査することにより、分割された DATA チャンクであることを認識し、再組立てのために分割された DATA チャンクをキューに入れる。一度ユーザメッセージが再組立てされれば、SCTP は再組立てしたユーザメッセージを、順番の並べ替えと最終的なディスパッチのために、指定したストリームへ渡す。

注: 多量の分割されたメッセージの断片が、メッセージの再組立てが完成するのを待つ間、データ受信者のバッファがあふれた場合は、メッセージの一部を配布するための API(2.10 参照) によってメッセージの一部をディスパッチし、残りのメッセージを受け取ることができるように受信バッファを空けることを推奨する。

2.6.10 バンドル

エンドポイントは、複数のチャンクを 1 つの SCTP パケットに単純に包含することによって、チャンクのバンドルを行う。バンドル後の IP ダイヤグラムのサイズは、SCTP パケットと IP ヘッダを含めて、現在のパス MTU 以下にする。

送信先エンドポイントがマルチホームの場合、送信側エンドポイントは、現在のプライマリパスの最新の MTU 以下のサイズを選ぶ。

DATA チャンクと一緒に制御チャンクをバンドルする場合、エンドポイントは送信する SCTP パケットの最初に制御チャンクを置く。送信者は、SCTP パケット内の DATA チャンクを TSN の昇順に転送する。

注: 制御チャンクはパケット中の最初に置かなければならず、一方で DATA チャンクは SHUTDOWN や SHUTDOWN ACK チャンクの前に置くので、DATA チャンクは SHUTDOWN や SHUTDOWN ACK チャンクと一緒にバンドルすることができない。

不十分なチャンクは SCTP パケットに置かれてはならない。

エンドポイントは受信したチャンクをパケット中の順番どおりに処理する。受信者は、チャンクの長さのフィールドを、全てのチャンクが 4 バイトバウンダリであることを考慮しつつ、チャンクの終端と次のチャンクの始端を決定するために用いる。受信者が部分チャンクを見つけた場合は、そのチャンクを破棄する。エンドポイントは、他のチャンクと一緒に、INIT, INIT ACK, SHUTDOWN COMPLETE をバンドルしてはならない。

2.7 輻輳制御

輻輳制御は SCTP の基本機能の 1 つである。アプリケーションによっては、リアルタイムデータ伝送を保証するための適切なリソースが SCTP トラヒックに割当てられる。したがって、通常のオペレーションでは、

送信について厳しい輻輳状態に遭遇することは起こりにくいように見える。しかしながら、SCTP は、部分的なネットワーク障害や予期せぬトラヒックの変化によって引き起こされる悪条件下においても動作する。そのような状況においては、SCTP は、輻輳から早急に脱する正確な輻輳制御ステップに従う。ネットワーク輻輳のない状態では、これら予防的な輻輳制御アルゴリズムは、SCTP プロトコルの動作に何ら影響を与えてはならない。

実装上の注意:以下に示す輻輳制御アルゴリズムの中で、特定のシステムの性能要求を満たす範囲内で、より堅固な輻輳制御アルゴリズムを選択することが認められている。

SCTP によって使用される輻輳制御アルゴリズムは、[RFC2581]に基づく。本節では、RFC2581 に定義されたアルゴリズムを SCTP で使用する場合に、どのように適用するかを説明する。まず、TCP と SCTP のプロトコル設計上の違いをリストアップし、次に、SCTP の輻輳制御スキームを説明する。輻輳制御に関する用語は、適切な場合は常に TCP での用語と同じ表現を用いる。

SCTP 輻輳制御は、個々のストリームでなく、常にアソシエーション全体に対して適用される。

2.7.1 SCTP 輻輳制御と TCP 輻輳制御の違い

SCTP SACK の中のギャップ Ack ブロックは、TCP SACK と同じ論理的な意味合いを伝送する。TCP は、SACK で運ばれる情報を、勧告情報としてのみとらえる。SCTP は、SACK チャンクのギャップ Ack ブロックによって運ばれる情報を、勧告としてとらえる。SCTP では、SACK として認識された DATA チャンク(受信端に乱れて到着した DATA を含む)は、累積 TSN Ack ポイントが DATA チャンクの TSN を超えるまで(つまり、DATA チャンクが SACK の累積 TSN Ack フィールドによって承認されるまで)、全て伝送されたとはみなされない。従って、cwnd の値は、(non-SACK TCP での場合のような、)最も大きな値で承認されたシーケンス番号と、輻輳ウインドウの大きさ内で送信することができる最新の DATA チャンクの間の上限を制御するのではなく、未解決なデータの量を制御する。SCTP SACK は、non-SACK TCP よりもはやい再送、早い回復を実現する。

しかしながら、SCTP と TCP の間の最も大きな違いは、マルチホームである。SCTP は、各々1 つ以上の伝送アドレスによって到達可能な、2 つのエンドポイント間の強固な通信アソシエーションを確立するように設計されている。2 つのエンドポイント間では、異なるアドレスが異なるデータパスを導いてもよい。このため、理想的にはパス各々の輻輳制御パラメタを個別に設定する必要があるかもしれない。マルチホーム受信のための輻輳制御処理は SCTP と同様に新しく、改善を今後要求してもよい。現在のアルゴリズムは、以下の仮定のもとに成り立っている。

- 送信者は、通常、上位レイヤによって指示されるまで、同じ宛先アドレスを使用する。しかし、SCTP はアドレスが機能しない事態が発生すると、別の宛先アドレスに変更できる(2.8.2 項を参照)。さらに、SCTP は最初の送信時とは異なるアドレスへデータを再送することもできる。
- 送信者は、送信することのできる宛先アドレス全ての(送信元－宛先のペア毎ではなく、それぞれの宛先毎の)輻輳制御パラメタ集合を保持している。十分な長い期間においてアドレスが使用されなかった際には、そのパラメタは捨てられる。
- それぞれの宛先アドレスに対して、エンドポイントはそのアドレスへの最初の送信をスロースタートする。

注: TCP は、単一の TCP セッション内において、TCP の上位レイヤプロトコルに対し、順番を保証したデータ伝送を行う。これは、TCP が受信シーケンス番号のギャップを通知するときに、損失したデータのシーケンス番号よりも大きいシーケンス番号を受け、データを転送する前にギャップが満たされるのを待つことを意味する。一方で、ストリームシーケンス番号が、あるストリーム内で順に並んでいる(つまり、不足している DATA チャンクは異なるストリームにある)場合、あるいは順序性を保証していない伝送が示されている場合に、SCTP は TSN にギャップがあっても上位レイヤに対してデータを引き渡すことができる。これは

cwnd に影響するものではないが、rwnd の計算には影響する可能性がある。

2.7.2 SCTP スロースタートと輻輳回避

スロースタートと輻輳回避のアルゴリズムは、ネットワークに流されるデータ量を制御するため、エンドポイントにより適用される。SCTP における輻輳制御は、個々のストリームにではなく、アソシエーションに対して適用される。状況によっては、SCTP 送信者がアルゴリズムの許容範囲よりもさらに保守的であることは有益だろう。しかし、SCTP 送信者は、以下に示すアルゴリズムの許容範囲よりも積極的になってはいけない。

TCP と同様、SCTP エンドポイントは、送信レートを制御するために、次に示す 3 つの制御変数を使用する。

- 受信能力通達ウィンドウサイズ (rwnd, 単位: バイト)
  受信者により、入力パケット用に利用可能なバッファ容量に基づいて設定される。
  注: この変数はアソシエーション全体にわたって保持される。
- 輻輳制御ウインドウ (cwnd, 単位: バイト)。
  送信者により、観測されたネットワークの状況に基づいて調整される。
  注: この変数は宛先アドレス毎に管理される。
- スロースタート閾値 (ssthresh, 単位: バイト)。
  送信者により、スロースタートや輻輳回避フェーズを識別するために使用される。
  注: この変数は宛先アドレス毎に管理される。

SCTP は、さらに輻輳回避フェーズで cwnd の調整を容易にするために使用される、あらたな制御変数 partial_bytes_acked を必要とする。

TCP と異なり、SCTP 送信者は、これらの制御変数 cwnd, ssthresh, partial_bytes_acked のセットを、(マルチホームである場合は)それぞれの宛先アドレス毎に保持する。rwnd だけは、アソシエーション全体で 1 つ (相手がマルチホームでも、1 つのアドレスだけを持っていても、問題はない) だけ、保持される。

2.7.2.1 スロースタート

未知の状況、あるいは長いアイドル状態の後で、ネットワークへのデータ転送を開始するには、伝送可能なキャパシティを決定するネットワークプローブが SCTP に必要である。スロースタートアルゴリズムは、データ転送開始前、もしくは再送タイマによって検出された回復ロスの後で、前記の目的で使用される。

- データ転送前あるいは長いアイドル期間後の cwnd は、2×MTU 以下にする。
- 再送タイムアウト後の cwnd は、1×MTU 以下にする。
- ssthresh の初期値は、意図的に高くしてもよい。(例えば、rwnd サイズを使用してもよい。)
- cwnd の値が 0 よりも大きい場合は、エンドポイントは、伝送アドレスに対して cwnd サイズ分のデータを持つことが認められる。
- cwnd が ssthresh 以下であるとき、SCTP エンドポイントが cwnd を増加させるためには(現在の輻輳ウインドウがフルに使われていると判断して)スロースタートアルゴリズムを用いる。受信する SACK が累積 TSN Ack ポイントを越えている場合、cwnd は 1) 到達確認が得られていて、これまで未解決の DATA チャンクサイズの合計、2) 送信先パス MTU, のうちの小さい方まで拡張される。これは、ACK-Splitting 攻撃([SAVAGE99]に概要が書かれている)に対する防御である。

相手側エンドポイントがマルチホームの場合、エンドポイントが SACK を受け取ってこれが累積 TSN Ack ポイントを更新する場合、Ack データを送信した先のアドレスに割当てられた cwnd を更新する。しかし、受信した SACK が累積 TSN Ack ポイントを更新しない場合は、エンドポイントはその宛て先アドレスの cwnd を調整してはならない。

エンドポイントの cwnd は累積 TSN Ack ポイントとは結びついていないので、重複した SACK が来た場合は、累積 TSN Ack ポイントの更新がなくても、エンドポイントは新しいデータの計測をするためにこれを引き続き使う。これはつまり、SACK によって新しく到達確認がされたデータが、cwnd より少ない範囲で届くデータの量を減らすということである。そして、その時点では cwnd は変化していないので、新しいデータを送信することは許されるということである。一方、cwnd の増加は、上記のように、累積 TSN Ack ポイントと結びついている。そうでないと、重複した SACK が新しいデータを受け付けなくするだけでなく、逆に、輻輳している間にネットワークに取り残されている新しいデータを受け付けなくしてしまう。

- エンドポイントが既存の伝送アドレスでデータを伝送しないとき、伝送アドレスの cwnd は RTO 毎に、cwnd/2 と 2*MTU のうちの最大値に調整される。

2.7.2.2 輻輳回避

ssthresh より cwnd が大きい時には、トランスポートアドレスに対応した未処理のデータが cwnd バイト以上あるならば、送信者は cwnd を RTT あたり 1*MTU 分増加する必要がある。

実装はこのゴールを以下の方法で達成できる:

- partial_bytes_acked は 0 で初期設定される。
- cwnd が ssthresh より大きいときはいつでも、累積 TSN Ack ポイントよりも進んだ SACK が到着した時点で、その SACK で確認されたすべての新しいチャンクの全バイト量の分を、partial_bytes_acked を増加させる。それらチャンクには、新しい累積 TSN Ack とギャップ Ack ブロックにより確認されたチャンクを含む。
- partial_bytes_acked が cwnd 以上の時で、SACK が到着する前に送信者が cwnd 以上のバイト数のアウトスタンディングデータを持っていれば（すなわち、SACK の到着の前に、ネットワーク上のデータサイズが cwnd 以上ならば）、MTU 分 cwnd は増加し、partial_bytes_acked を (partial_bytes_acked - cwnd)でリセットする。

スロースタートと同様に、送信側が与えられたトランスポートアドレスへの DATA を送信しない時には、トランスポートアドレスの cwnd は RTO あたり max（cwnd/2, 2*MTU）に調整される必要がある。

送信側により送信されたデータがすべてが受信側により確認された時には、partial_bytes_acked は 0 に初期設定される。

2.7.2.3 輻輳制御

SACK（2.7.2.4 節を参照）からパケットロスの検出した時、エンドポイントは以下のようにする必要がある:

```
ssthresh = max(cwnd/2, 2*MTU)
cwnd = ssthresh
```

基本的に、パケットロスは cwnd を半分にする。

あるアドレスにおいて T3-rtx タイマが満了する時には、SCTP はスロースタートを実行するべきである:

```
ssthresh = max(cwnd/2, 2*MTU)
cwnd = 1*MTU
```

そして、エンドポイントがそのアドレスへのデータ転送が成功したという確認を受け取るまで、1 つだけの SCTP パケットがそのアドレスへ飛んでいると保証する。

2.7.2.4 高速再転送とギャップ報告

データロスがないときには、エンドポイントは遅延確認を行う。しかし、エンドポイントが到着した TSN シーケンスに抜けを見つけたときはいつでも、抜けが満たされるまでデータを含むパケットが到着する毎に

SACK を送り返す必要がある。

エンドポイントがいくつかの TSN の欠けを示す SACK を受信したときはいつでも、高速再転送を行う前にさらにあと 3 回の同じ TSN の欠けの通知（以後に来る SACK に含まれる）を待つ必要がある。

連続して 4 回目の SACK で TSN の欠けが報告された時には、データ送信側は以下のようにする:

再送のために欠けた DATA チャンクをマークする。

2.7.2.3 節で記述された式により、欠けた DATA チャンクが最後に送られた宛先アドレスに対する ssthresh と cwnd を調整する。

再送のためにマークされた最も早い（すなわち、最も低い TSN）DATA チャンクのいくつが、パケットが送られている宛先トランスポートアドレスのパス MTU の制約条件下で、単一のパケットに納まるかを決定する。この値を K と呼ぶ。単一のパケットでそれら K 個の DATA チャンクを再転送する。

もしそのアドレスに送られた最も低いアウトスタンディング TSN 番号が最後の SACK で確認された場合にのみ、T3-rtx タイマを再スタートさせる。またはエンドポイントはそのアドレスに送られた最初のアウトスタンディング DATA チャンクを再転送している。

上記の直接的な実装は、SACK により報告された 1 つの TSN の欠け毎に 1 つのカウンタを保持する。カウンタは、TSN の欠けを報告している個々の連続的な SACK のためにインクリメントする。そのカウンタが 4 に達し高速再転送手順を始めた後に、カウンタは 0 にリセットする。

SCTP の cwnd がアウトスタンディング TSN の数と間接的に関連しているので、TCP 高速リカバリの効果は調整なしで輻輳制御ウィンドウサイズに自動的に達成される。

2.7.3 Path MTU 発見

[RFC1191]は、「パス MTU 発見」を記述している。それによって、エンドポイントは与えられたインターネットパスに沿って最大のトランスミッションユニット（MTU）の見積りを維持し、パス MTU（パス MTU）の変化を確かめる時折の試みを除いて MTU を越えているそのパスに沿った発送パケットを控える。RFC1191 は、現在のエンド-エンド MTU の設定だけでなくこの値が変化したことを検出して決定するための MTU 発見メカニズムと戦略を徹底的に議論している。[RFC1981]は IPv6 のための同じメカニズムを記述している。IPv6 を使っている SCTP 送信者は、全てのパケットを最小の IPv6 MTU［RFC2460］より小さくするためパス MTU 発見を使う。

エンドポイントはこれらのテクニックを適用する必要があり、宛先アドレス毎に基づきそうする必要がある。

TCP に関して適用される MTU 発見の RFC1191 の記述から、SCTP は 4 つの方法が異なっている:

1） SCTP アソシエーションは複数のアドレスを測定できる。エンドポイントはその相手の個々の宛先アドレスに対して見積もられる MTU を別々に維持する。

2） このドキュメントのどこでも、用語「MTU」が議論されるときは、議論の文脈と一致している宛先アドレスと結合した MTU を示している。

3） TCP と違って、SCTP は「最大セグメントサイズ」の観念を持っていない。従って、それぞれの宛先アドレスへの MTU は、そのリモート宛先アドレスへのパケットがルーティングされるローカルインターフェースへのリンク MTU より大きくない値で、初期化される必要がある。

4） SCTP のデータ転送は、バイト(TCP の場合)というよりも TSN について構造化されるので、RFC1191 の 6.5 節での議論があてはまる:IP データグラムを IP データグラムがそのアドレスのために MTU パスのために大きすぎるようなリモートのアドレスに再転送する時には、IP データグラムは DF ビットセットなしでフラグメントされる可能性を持って再転送される必要がある。新しい IP データグラムの転送は DF セットになる。

5） 送信側は、相手の宛先アドレスすべてのうち、発見された最も小さいパス MTU であるアソシエーシ

ョンパス MTU を追跡する必要がある。複数の部分へメッセージがフラグメントされるとき、個々のフラグメントのサイズを計算するためにこのアソシエーションパス MTU が使用される必要がある。これは、IP フラグメンテーションに遭遇せずに再送が他方のアドレスにシームレスに送られることを可能にする。

これらの違いを除いて、RFC1191 と 1981 での MTU 発見の TCP での使用の議論が、宛先アドレス毎に基づいて SCTP にもあてはまる。

2.8 障害管理

2.8.1 エンドポイント障害検出

エンドポイントは相手へ再送する場合、そのトータル回数をカウントする(相手がマルチホームの場合は、その全てのトランスポートアドレスへの再送をカウントする)。このカウンタの値がプロトコルパラメタ Association.Max.Retrans で示される限度を超えた時、エンドポイントは相手側エンドポイントがアンリーチャブルであると判断し、この相手へそれ以降データ送信することを止める(そしてアソシエーションを CLOSED 状態にする)。加えて、エンドポイントは上位レイヤに障害を通知し、オプションとして、送信キューに残っている全ての未送出ユーザデータの報告をする。相手側エンドポイントがアンリーチャブルになった場合、アソシエーションは自動的にクローズされる。

再送カウンタは、相手側エンドポイントへ送った DATA チャンクが(SACK により)到達確認される 、もしくは、HEARTBEAT-ACK が相手側エンドポイントから届く度にリセットされる。

2.8.2 パス障害検出

相手側エンドポイントがマルチホームの場合、そのトランスポートアドレス毎にエラーカウンタを保持する。

いずれかのアドレスにて T3-rtx タイマが終了した時、もしくは、アイドル状態のアドレスに送られた HEARTBEAT の到達確認が RTO 時間内に受け取れなかった時に、その宛て先アドレスのエラーカウンタをインクリメントする。宛て先アドレスのエラーカウンタの値がプロトコルパラメタ Path.Max.Retrans を越えた時は、エンドポイントは、その宛て先トランスポートアドレスを inactive にマークし、上位レイヤに通知することを推奨する。

送出済み TSN の到達確認がなされた、もしくは、HEARTBEAT の到達確認が HEARTBEAT ACK で行われた時、エンドポイントは DATA チャンクを最後に送った(もしくは、HEARTBEAT を送った)宛て先トランスポートアドレスのエラーカウンタをクリアする。相手側エンドポイントがマルチホームで、最後に送ったチャンクがこのエンドポイントの代替アドレスへの再送だった場合、到達確認が最後にチャンクを送ったアドレスからなのか曖昧性が残る。しかし、この曖昧性は、SCTP の振る舞いに重要な結果を与えるものではない。この曖昧性が望ましくないなら、最後のチャンクが再送だった場合は、送信者は、エラーカウンタをクリアしなくても良い。

注: SCTP エンドポイントを設定しているとき、ユーザは Association.Max.Retrans 値を相手側エンドポイントの全ての宛て先アドレスの Path.Max.Retrans 値の総和よりも大きくすることを避ける。そうしないと、エンドポイントが、相手側エンドポイントがリーチャブルであると考えている時に、全ての宛て先アドレスが inactive になってしまう。この状態になったとき、SCTP がどのように機能するかは実装仕様による。

(例えば、多くの再送の結果)主要パスが inactive とマークされたとき、(代替アドレスが存在し、active ならば、)送信者は、新規のパケットを、自動的に代替宛て先アドレスに送っても良い。主要パスが inactive にマークされた時に 1 つ以上の代替アドレスが active でも、1 つのトランスポートアドレスのみが宛先として選ばれることを推奨する。

2.8.3 パス・ハートビート

SCTP エンドポイントは、デフォルトでは、HEARTBEAT チャンクを定期的に送信することによりアイドル状態である相手の宛て先トランスポートアドレスの到達性(reachability)を、モニタする。

パスの RTT を更新するために使われる新しいチャンク(通常は、最初の転送 DATA、INIT、COOKIE ECHO、HEARTBEAT など)がなく、現在のハートビート期間内に HEARTBEAT が送られないなら、宛て先トランスポートアドレスは、「アイドル状態」と見なされる。これは、active/inactive な宛先アドレスの両者に適用される。

上位レイヤは、オプションで、以下の機能を開始できる:

A) 既存アソシエーションの特定の宛て先トランスポートアドレスに対するハートビートを不可にする。
B) ハートビートインターバル(HB.interval)を変更する。
C) 既存アソシエーションの特定の宛て先トランスポートアドレスに対するハートビートを再び可能にする。そして、
D) 既存アソシエーションの特定の宛て先トランスポートアドレスに対するハートビートをオンデマンドで要求する。

エンドポイントは、HEARTBEAT を送って RTO 時間内に到達確認が返ってこない場合に、その宛て先トランスポートアドレスのエラーカウンタをインクリメントする。

このカウンタの値がプロトコルパラメタ Path.Max.Retrans に到達し、対応する宛先アドレスが inactive にマークされていないなら、エンドポイントはこの宛て先アドレスを inactive にマークする。そして、オプションで、上位レイヤにこの到達性の変化を報告しても良い。この後、エンドポイントは、この宛て先アドレスへの HEARTBEAT を送り続けるが、カウンタのインクリメントは中断する。

HEARTBEAT チャンクの送信者は、チャンクのハートビート情報フィールドの中に、パケットを送った時間とパケットを送った宛て先アドレスを入れる。

実装上の注意: HEARTBEAT が宛て先に送られるたび毎に、エラーカウンタをインクリメントするという、別のハートビート機構の実装が可能である。HEARTBEAT ACK 到着時に、送信者は HEATBEAT を送った宛て先のエラーカウンタをクリアすることを推奨する。つまり、それまでに記録したエラーをクリアすることになる。

HEARTBEAT の受信者は、すぐに HEARTBEAT ACK にて応答する。この HEARTBEAT ACK には、受信した HEARTBEAT チャンクからコピーされたハートビート情報フィールドが含まれている。

HEARTBEAT ACK を受信したら、HEARTBEAT の送信者は、HEARTBEAT チャンクを送った宛て先トランスポートアドレスのエラーカウンタをクリアし、もし、宛て先トランスポートアドレスが active にマークされていないならば、active にマークする。エンドポイントは、Inactive な宛て先アドレスが、最新の HEARTBEAT ACK を受信したことによって、active にマークされた時には、オプションで、この変更を上位レイヤにレポートしても良い。HEARTBEAT ACK の受信者は、(2.8.1 節で定義されているように)アソシエーション全体のエラーカウントもクリアする。

HEARTBEAT ACK の受信者は、さらに、宛て先トランスポートアドレスの RTT を、HEARTBEAT ACK チャンクで届けられる時間値を使って計測する。

ハートビートが許可されているアイドル状態の宛て先アドレスに対しては、{RTO 時間＋プロトコルパラメタ(HB.interval)（ただし、+/-50%のジッタを含む）}毎に、HEARTBEAT を送ることが推奨される。前回の HEARTBEAT が受信確認されない場合は、RTO 時間を指数関数的に増加させる。

SCTP ユーザによる HB.interval の変更および、既存の宛て先アドレスに対するハートビートのオン/オフ設定のためにプリミティブが提供される。SCTP によって設定されたハートビート間隔はその宛て先の RTO 時間（指数関数的な増加分も含む）に加えられる。(複数の宛て先がアイドル状態であるならば、)ハートビー

トタイマが終了する毎に、ハートビートが 1 つのみ送られる。 (1 つ以上の宛て先がアイドルであるときに) ハートビートを送る宛先アドレスを選択する方法は、実装上の決定事項である。

注: ハートビート間隔をチューニングするとき、副作用について考慮することを推奨する。間隔が増加したとき、即ち、HEARTBEAT に要する時間が長くなったとき、消失した ABORT メッセージの検出に要する時間も長くなる。相手側エンドポイントが何らかの理由でアソシエーションを ABORT したが、ABORT チャンクが消失してしまったならば、ローカルのエンドポイントは DATA チャンクまたは、HEARTBEAT チャンクを送ることによってのみ、ABORT が消失していることに気付くことができる(これは相手側に別の ABORT を送るきっかけを与える)。以上は HEARTBEAT タイマをチューニングするときに考慮される。HEARTBEAT が不可の状態なら、アソシエーションへ DATA を送ることによってのみ、相手が送出した ABORT の消失に気付くことができる。

2.8.4 OOTB パケットの扱い

SCTP パケットが正しいフォーマットである(つまり、受信者の Adler-32 チェック(2.6.8 節参照)を通過した) が、受信者はこのパケットが属するアソシエーションを識別できないとき、この SCTP パケットを"out of the blue"(OOTB)パケットと呼ぶ。

OOTB パケットを受信したら、以下の処理を行う。

1) OOTB パケットの送信先もしくは送信元がユニキャストアドレスではないならば、暗黙のうちにパケットを破棄する。そうでなければ、
2) OOTB パケットに ABORT チャンクが含まれているならば、暗黙のうちにパケットを破棄し、それ以上の操作を行わない。そうでなければ、
3) OOTB パケットに INIT チャンクが含まれていて、その照合タグが「0」に設定されているならば、2.5.1 節に記述されている手順で処理する。そうでなければ、
4) COOKIE ECHO が OOTB パケット中の最初のチャンクなら、2.5.1 節に記述されている手順で処理する。そうでなければ、
5) OOTB パケットに SHUTDOWN ACK チャンクが含まれているならば、SHUTODOWN COMPLETE にて送信元に応答する。SHUTODOWN COMPLETE を送る時には、照合タグには受信した SHUTDOWN ACK 中の照合タグを設定し、T-bit には TCB が見出せなかったことを示すチャンクフラグを設定する。そうでなければ、
6) OOTB パケットに SHUTDOWN COMPLETE チャンクが含まれているならば、暗黙のうちにパケットを破棄し、それ以上の処理を行わない。そうでなければ、
7) OOTB パケットに「失効クッキー」エラーか、COOKIE ACK が含まれているならば、暗黙のうちにパケットを破棄する。そうでなければ、
8) 受信者は、ABORT を用いて送信元に対して応答すすることを推奨する。ABORT を送る時には、照合タグには受信したパケット中の照合タグを設定し、T-bit には TCB が見つからないことを示すチャンクフラグを設定する。ABORT を送った後は、OOTB パケットを破棄し、それ以上の処理を行わない。

2.8.5 照合タグ

この節で規定している照合タグのルールは、INIT、SHUTDOWN COMPLETE、COOKIE ECHO (2.5.1 節参照)、ABORT または、SHUTDOWN ACK を含まない SCTP パケットの送受信時に適用される。これらのタイプのチャンクを含んでいる SCTP パケットの送受信時のルールは、2.8.5.1 節にて議論される。

SCTP パケットを送るとき、エンドポイントは、送信パケットの照合タグフィールドに、相手から受信した INIT または INIT ACK の開始タグパラメタの値を入れる。

SCTP パケットを受信したときは、エンドポイントは、受信した SCTP パケットの照合タグフィールドの値が自身のタグと一致していることを確かめる。一致していないならば、受信したパケットを暗黙のうちに破棄し、2.8.5.1 節に示した場合を除きそれ以上の処理を行わない。

2.8.5.1 照合タグ・ルールの例外

A) INIT を転送するパケットのためのルール:
   - 送信者はパケットの照合タグを 0 に設定しる。
   - エンドポイントが、照合タグ が 0 に設定されている SCTP パケットを受信した時、パケットに INIT チャンクが含まれていることを確認する。そうでなければ、受信者は暗黙のうちにパケットを破棄する。

B) ABORT を転送するパケットのためのルール:
   - エンドポイントは、宛先エンドポイントのタグ値が既知ならば、送信するパケットの照合タグフィールドに、常に、この値を入れる。
   - OOTB のレスポンスとして ABORT を送るならば、エンドポイントは 2.8.4 節に書かれた手順に従う。
   - 照合タグが自分のタグまたは、相手のタグと同じであるなら、受信者はパケットを受け取る。そうでないなら、暗黙のうちに破棄し、それ以上の処理を行わない。

C) SHUTDOWN COMPLETE を転送するパケットのためのルール:
   - SHUTDOWN COMPLETE を送る時、SHUTDOWN ACK の受信者が TCB を持っているならば、宛て先エンドポイントのタグを使う。TCB が存在しないときのみ、送信者は、SHUTDOWN ACK の照合タグを使う。
   - 受信したパケットの照合タグ フィールドが自分または、相手のタグと一致していてチャンクフラグに T-bit が設定されているならば、受信者は、SHUTDOWN COMPLETE を受け取る。そうでないならば、受信者は、暗黙のうちにパケットを破棄し、それ以上の処理を行ってはならない。エンドポイントは、SHUTDOWN-ACK-SENT 状態でないならば、SHUTDOWN COMPLETE を無視する。

D) COOKIE ECHO を転送するパケットのためのルール:
   - COOKIE ECHO を送るとき、エンドポイントは INIT ACK にて受信した Initial Tag の値を使う。
   - COOKIE ECHO の受信者は、2.5 節に記述された手続きに従う。

E) SHUTDOWN ACK を転送するパケットのためのルール:
   - 受信者が COOKIE-ECHO または COOKIE-WAIT の状態ならば、2.8.4 節に記述されている手続きに従うことを推奨する。つまり、OOTB パケットとして扱われる。

2.9 アソシエーション解放

エンドポイントは、サービス終了時にアソシエーションを終了させる。アソシエーションは、アボートかシャットダウンで終了させることができる。アソシエーションのアボート時には、アソシエーションのどちらかのエンドポイントでペンディングとなっているデータが破棄され、相手側に届けられない。アソシエーションのシャットダウンとは、正常な終了であり、両エンドポイントのキューに溜まっている全てのデータはそれぞれの相手に届けられる。しかし、シャットダウンの場合、SCTP は(TCP のような)一方がクローズしているのに他方はデータを送りつづけられるという状態はサポートしていない。どちらかのエンドポイントがシャットダウンを実行するときは、アソシエーションの両エンドポイントは上位レイヤから新しいデータを受け取るのを止め、SHUTODOWN チャンクを送受信している間にキューに溜まっているデータを届け

る。

2.9.1 アソシエーションのアボート

エンドポイントが既存のアソシエーションをアボートすると決めたときは、相手に ABORT チャンクを送る。送信者は、相手の照合タグを送信パケットに含める。また、ABORT チャンクには DATA チャンクをバンドルしてはならない。

ABORT を含んだパケットを受信したエンドポイントは、これに対してレスポンスを送信してはならない(2.8.4 節参照)。

ABORT を受信したエンドポイントは、2.8.5.1 節に示された照合タグのチェックルールを適用する。

照合タグをチェックした後は、受信者エンドポイントはアソシエーションに関する記録を除去し、上位レイヤへその終了を報告する。

2.9.2 アソシエーションのシャットダウン

アソシエーションのエンドポイントの上位レイヤは SHUTDOWN プリミティブを使ってアソシエーションを正常にクローズすることができる(2.10.1 節参照)。この場合、シャットダウンの受信者は、送信すべき全ての DATA チャンクをアソシエーションが終了する前に送信者へ届けることができる。

上位レイヤから SHUTDOWN プリミティブが届くと、エンドポイントは SHUTDOWN-PENDING の状態に入り、送信した全てのデータに対して相手側からの到着確認が得られるまでその状態に留まる。エンドポイントは上位レイヤからは新しいデータは受け取らないが、ギャップを埋める必要がある場合は相手にデータを再送する。

全ての送出データの到達確認がなされたら、エンドポイントは相手に SHUTDOWN チャンクを送る。このチャンクの Cumulative TSN Ack フィールドには、相手から受信した最後の順番の TSN が入っている。その後、T2-shutdown タイマを開始し、SHUTDOWN-SENT 状態に入る。タイマが終了したならば、エンドポイントは相手から届いた最後の順番の TSN を更新して SHUTDOWN を再送する。

T2-shutdown タイマの適切な値を決定するためには、2.6.3 節のルールに従う。TSN のギャップを示すために、エンドポイントは同じ SCTP パケットで、SHUTDOWN チャンクに SACK をバンドルしてもよい。

エンドポイントはプロトコルパラメタ Association.Max.Retrans によって、SHUTDOWN チャンクの再送回数を制限することを推奨する。このしきい値を越えたならば、エンドポイントは TCB を廃棄し、上位レイヤへ相手のエンドポイントがアンリーチャブルであることをレポートする(そしてアソシエーションは CLOSED 状態に入る)。相手からパケットを受信したとき(つまり、相手がキューに溜まっていた DATA チャンクを送った時)、エンドポイントの再送カウンタをクリアし、T2-shutdown タイマをリスタートする。このことにより、相手側のキューに溜まっていてまだ送られていない DATA チャンクを送るための充分な機会を与えることができる。

SHUTDOWN を受け取ると、エンドポイントは、

- SHUTDOWN-RECEIVED 状態に入る。
- SCTP ユーザから新しいデータを受け取ることを止める。
- チャンクの Cumulative TSN Ack フィールドをチェックすることによって、全ての送出 DATA チャンクが SHUTDOWN 送信者で受け取られていることを確認する。

一度 SHUTDOWN-RECEIVED 状態になったら、エンドポイントは、ULP の要求に対して、SHUTDOWN を送ってはならない。また、その後に届いた SHUTDOWN チャンクを破棄することを推奨する。

まだ DATA チャンクが残っているならば、SHUTDOWN 受信側は、全ての DATA チャンクの受信通知を受け取るまで 2.6 節に示された通常のデータ転送手続きに従い続ける。しかし、SHUTDOWN 受信者は、SCTP ユーザから新しいデータを受け取ってはならない。

SHUTDOWN-SENT 状態の間は、SHUTDOWN 送信者は、SACK や SHUTDOWN チャンクとともに 1 つ以上の DATA チャンクを含んでいるパケットを受信した場合にすぐに応答し、T2-shutdown タイマを再開する。送出する DATA チャンクがなくなったら、SHUTDOWN 受信者は SHUTDOWN ACK を送り、自身の T2-shutdown タイマを開始し、SHUTDOWN-ACK-SENT 状態に入る。タイマが終了したならば、エンドポイントは SHUTDOWN ACK を再送する。

SHUTDOWN ACK の送信者は、プロトコルパラメタ Association.Max.Retrans を使って、SHUTDOWN ACK の再送回数を制限することを推奨する。しきい値を越えたならば、エンドポイントは TCB を廃棄し、相手側がアンリーチャブルであることを上位レイヤに報告しても良い(そしてアソシエーションは CLOSED 状態に入る)。

SHUTDOWN ACK を受信したら、SHUTDOWN 送信者は、T2-shutdown タイマを止め、SHUTDOWN COMPLETE チャンクを相手に送る。そして、アソシエーションに関する全ての記録を削除する。

SHUTDOWN COMPLETE を受信したら、エンドポイントは、SHUTDOWN-ACK-SENT 状態であることを確認する。SHUTDOWN-ACK-SENT 状態でないならばチャンクを破棄する。エンドポイントが SHUTDOWN-ACK-SENT 状態ならば、T2-shutdown タイマを止め、アソシエーションに関わる全ての情報を削除する(そしてアソシエーションは CLOSED 状態に入る)。

エンドポイントは、シャットダウン処理に入る前に、送出した全ての DATA チャンクが到達確認されていることを確認することを推奨する。

エンドポイントは、SHUTDOWN-PENDING, SHUTDOWN-SENT, SHUTDOWN-RECEIVED, または、SHUTDOWN-ACK-SENT 状態のときは、上位レイヤからの新しいデータをリジェクトすることを推奨する。

エンドポイントが SHUTDOWN-ACK-SENT 状態で、shutdown 中のアソシエーションと同じ送信元/送信先のトランスポートアドレスを(IP アドレス中、もしくは、チャンク中に)含む INIT チャンクを受信した場合(例えば、SHUTDOWN COMPLETE が消失したとき)、INIT チャンクを破棄し、SHUTDOWN ACK チャンクを再送することを推奨する。

注: エンドポイントに shutdown 中のアソシエーションと同じ送信元 IP アドレス / 送信先 IP アドレスがアサインされているがポート番号が異なる INIT を受信した場合、この INIT は、異なるアソシエーションの初期化を示している。

INIT もしくは COOKIE ECHO の送信者は、SCTP パケット中のスタンドアロンの SHUTDOWN-COMPLETE とその共通ヘッダ中の照合タグを用いて SHUTDOWN-ACK に対して応答することを推奨する(照合タグには SHUTDOWN ACK パケットの中で受信したタグと同じタグがセットされる)。これは、2.8.4 節で定義されている Out Of the Blue パケットと考えられる。INIT の送信者は T1-init を走らせたままにしておき、COOKIE-WAIT もしくは COOKIE-ECHOED 状態にとどまる。T1-init タイマが終了すると通常、INIT チャンクもしくは COOKIE チャンクは再送され、新しいアソシエーションが開始される。

SHUTDOWN が COOKIE-WAIT 状態か COOKIE-ECHOED 状態にて受け取られたならば、暗黙のうちに破棄される。

エンドポイントが SHUTDOWN-SENT 状態のときに相手から SHUTDOWN チャンクを受信したならば、エンドポイントはすぐに相手に SHUTDOWN-ACK を返信し、SHUTDOWN-ACK-SENT 状態に移行し、T2-shutdown タイマを再開する。

エンドポイントが SHUTDOWN-ACK-SENT 状態で、SHUTDOWN-ACK を受信したならば、T2-shutdown タイマを止め、SHUTDOWN-COMPLETE チャンクを相手に送り、アソシエーションに関する全ての記録を削除する。

2.10 上位レイヤインタフェース

上位レイヤプロトコル(ULP)は、 SCTP にプリミティブを渡すことによってサービスを要求し、様々なイ

ベントにおいて SCTP から通知を受け取る。

この節で記述されたプリミティブと通知は SCTP を実装する際のガイドラインとして用いられるものとする。以下の ULP インタフェースプリミティブの機能的な記述は、説明を補助することを目的として示されている。異なる SCTP 実装は異なる ULP インタフェースを持っている。しかしながら、すべての SCTP は、すべての SCTP 実装が同じプロトコル階層をサポートすることができることを保証するためのサービスの最小セットを提供する。

2.10.1 ULT から SCTP へ

以下の節は ULP / SCTP インタフェースを機能的に特徴づけている。使用される表記法は高水準言語における大部分の手順や、ファンクションコールと同様である。

以下に記述された ULP プリミティブは SCTP がプロセス間通信をサポートするために実行する標準機能を規定している。個別の実装がそれら自身の正確なフォーマットを定義する、1 つの呼び出しで標準機能の組み合わせあるいはサブセットを提供してもよい。

A) 初期化

フォーマット: INITIALIZE ([local port] [,local eligible address list])

→ local SCTP instance name

このプリミティブは SCTP にその内部データ構造を初期化して、その実行環境をセットアップするために必要なリソースを割当てさせる。SCTP が一度初期化されると、 ULP はこのプリミティブを再起動することなく、他のエンドポイントにおいて直接通信することができる。

SCTP は ULP に local SCTP instance name を返却する。

必須属性:

なし。

オプション属性:

以下の属性がプリミティブとともに引き継がれる:

local port － SCTP ポート番号。 ULP が SCTP ポート番号が指定されることを望む場合。

local eligible address list － ローカル SCTP エンドポイントがバインドすべきアドレスリスト。アドレスリストが含まれない場合、デフォルトとしてホストに割当てられたすべての IP アドレスがローカルエンドポイントによって使用されるものとする。

実装上の注意: このオプション属性が実装によってサポートされる場合、このエンドポイントによって送信されたどんな SCTP パケットの IP ソースアドレスフィールドも local eligible address list の中に示された 1 つの IP アドレスを含むということは、実装を行う上での責任である。

B) アソシエート

フォーマット: ASSOCIATE(local SCTP instance name, destination transport addr, outbound stream count)

→association id［,destination transport addr list］［,outbound stream count］

このプリミティブは上位レイヤに特定の相手エンドポイントへのアソシエーションを実行する。相手エンドポイントはエンドポイントを定義するトランスポートアドレスの 1 つによって指定される(0 節参照)。ローカル SCTP インスタンスが初期化されなかった場合、ASSOCIATE はエラーであると判断される。

SCTP アソシエーションへのローカルハンドルである association id は、アソシエーションの正常な確立時に返却されるであろう。SCTP が相手エンドポイントと SCTP アソシエーションを開始することが可能でない場合、エラーが返却される。

ローカルエンドポイントの outbound stream count と同様に、相手の完全な送信先トランスポートアドレスを含めて、他のアソシエーションパラメタが返却される。返却された送信先アドレスから、1 つのトランスポートアドレスが SCTP パケットをこの相手に送るためのデフォルトのプライマリパスとしてローカルエン

ドポイントによって選択される。返却された「送信先トランスポートアドレスリスト」は ULP によってデフォルトのプライマリパスを変更するのに使用するか、 または特定のトランスポートアドレスへのパケットを強制的に送信するために使用することができる。

実装上の注意: ASSOCIATE プリミティブがブロッキングファンクションコールとして実装される場合、ASSOCIATE プリミティブは正常な確立に関する association id のほかにアソシエーションパラメタを返却することができる。ASSOCIATE プリミティブが非ブロッキングコールとして実装される場合、association id のみ返却され、そしてアソシエーションパラメタは COMMUNICATION UP 通知を使って返却される。

必須属性:

local SCTP instance name ― INITIALIZE オペレーションから得られる。

destination transport addr ― アソシエーションが確立される相手エンドポイントのトランスポートアドレスの 1 つとして指定される。

outbound stream count ― ULP が相手エンドポイントに向かって開こうとしている outbound ストリーム数。

オプション属性:

なし。

C) シャットダウン

フォーマット: SHUTDOWN(association id)

→result

正常にアソシエーションを終了する。すべてのローカルキューに入れられたユーザデータは相手に届けられるものとする。アソシエーションは送られたすべての SCTP パケットを相手が確認した後、終了する。成功コードがアソシエーションの正常終了時に返却される。アソシエーションを終了させようとする試みの結果が異常の場合、エラーコードが返却される。

必須属性:

association id ― SCTP アソシエーションへのローカルハンドル

オプション属性:

D) なし。

アボート

フォーマット: ABORT(association id［,cause code］)

→result

強制的にアソシエーションを終了する。すべてのローカルキューに入れられたユーザデータは廃棄され、ABORT チャンクが相手に送信される。成功コードがアソシエーションの正常なアボーションで返却される。アソシエーションを終了させようとする試みの結果が異常の場合、エラーコードが返却される。

必須属性:

association id ― SCTP アソシエーションへのローカルハンドル

オプション属性:

原因コード ― 相手に渡されるアボートの理由

E) 送信

フォーマット: SEND(association id, buffer address, byte count ［,context］ ［,stream id］ ［,life time］ ［,destination transport address］ ［,unorder flag］ ［,no-bundle flag］ ［,payload protocol-id］)

→result

これは SCTP によってユーザデータを送る主な方法である。

必須属性:

association id ― SCTP アソシエーションへのローカルハンドル

buffer address － 転送されるユーザメッセージが格納されるアドレス

byte count － バイト数で示されるユーザデータサイズ

オプション属性:

context － このユーザメッセージの転送が異常の際に、送信異常通知によって ULP に引き継がれる任意の 32 ビットの整数。

stream id － データを送信するためのストリームを指定する。指定されていない場合、ストリーム 0 が使用される。

life time － ユーザデータの life time を指定する。life time が満了した場合、ユーザデータは SCTP によって送信されないものとする。このパラメタは古いユーザメッセージを送信する努力を避けるために使用することができる。life time 変数内でデータを転送開始することができないならば(すなわち、SCTP の送信プリミティブを通して送信先に送信すること) SCTP は ULP に通知を行う。 しかしながら、SCTP が、life time が満了する前に、チャンクを送信しようと試みた場合、ユーザデータは送信されるものとする。

実装上の注意: データ life time オプションをよりよくサポートするために､転送者は最後の瞬間まで outbound の DATA チャンクへの TSN 番号の割り当てを抑制する。そして、実装単純化のために、 一度 TSN 番号が割当てられると、DATA チャンクに付与されたどんな life time オプションも無効となるので、委ねられたこの DATA チャンクの送信を送信者は考慮しなければいけない。

destination transport address － このパケットが送られるべき相手エンドポイントの送信先トランスポートアドレスの 1 つを指定する。可能であるならば、 パケットを送るために SCTP は現在のプライマリパスの代わりに、この送信先トランスポートアドレスを使用する。

unorder flag － このフラグ指定された場合(すなわち､U フラグがこのメッセージを運んでいるすべての DATA チャンク上で 1 にセットされる)､ユーザは順序づけられていない形で相手にデータを送ることを望んでいることを示す。

no-bundle flag － このユーザデータに他の outbound の DATA チャンクをつながないよう SCTP に指示する。ネットワーク輻輳に直面しているときは、このフラグが指定されている時でも SCTP はまだつないでいる。

payload protocol-id － 転送するペイロードプロトコルデータのタイプを示す、相手から引き継がれる 32 ビット符号無し整数。この値は不明瞭なデータとして SCTP によって引き継がれる。

F) Set Primary

フォーマット: SETPRIMARY (association id, destination transport address ［,source transport address］)

→result

パケットを送信するためにプライマリパスとして送信先トランスポートアドレスを使用するようにローカル SCTP に指示する。

このオペレーションの実行結果は返却される。送信先トランスポートアドレスがアソシエートコマンドあるいは COMMUNICATION UP 通知で以前に返却された「送信先トランスポートアドレスリスト」に存在していないなら、エラーが返却される。

必須属性:

association id － SCTP アソシエーションへのローカルハンドル

destination transport address － パケット送信において、プライマリアドレスとして用いられる相手エンドポイントのトランスポートアドレスの 1 つを指定する。 これはローカル SCTP エンドポイントによって維持された現在のプライマリアドレス情報を無効にする。

オプション属性:

source transport address － オプションとして、いくつかの実装はすべての送信 IP データグラム内に置かれるデフォルトの source transport address をセットすることを許容する。

G) 受信

フォーマット: RECEIVE(association id, buffer address, buffer size ［,stream id］)

→byte count ［,transport address］ ［,stream id］ ［,stream sequence number］ ［,partial flag］ ［,delivery number］ ［,payload protocol-id］

1 つ利用可能なものがあるなら、このプリミティブは ULP によって指定されたバッファ内の SCTP 受信キュー の最初のユーザメッセージを読む。読まれたメッセージの大きさはバイトで返却される。 特定の実装によっては、送り元アドレス、受信した stream id、取り出し可能なメッセージなどのような、他の情報も返却する。順序づけられたメッセージにおいては、それらのストリームシーケンス番号も返却される。

実装に依存するが、利用可能なメッセージがないときに、このプリミティブが呼び出された場合、実装はこの状態の表示を返却するべきか、あるいはデータが利用可能になるまで、その他のプロセス呼び出しを阻止することを推奨する。

必須属性:

association id ― SCTP アソシエーションへのローカルハンドル

buffer address ― 受信メッセージを格納するために ULP によって示された格納アドレス

buffer size ― 受信データの最大サイズ(バイト)

オプション属性:

stream id ― どのストリーム上でデータ受信するかを示す

stream sequence number ― 送信 SCTP によって割当てられたストリームシーケンス番号

partial flag ― この返却されたフラグが 1 に設定されていたら、この Receive はメッセージ全体の部分的なデリバリを含んでいる。このフラグが設定されるとき、stream id とストリームシーケンス番号がこの受信に伴わなくてはならない。このフラグが 0 に設定されるとき、これ以上のデリバリがこのストリームシーケンス番号のために受信されないということを示す。

payload protocol-id ― 転送するペイロードプロトコルデータのタイプを示す、相手から受け取られる 32 ビット符号無し整数。この値は不明瞭なデータとして SCTP によって引き継がれる。

H) 状態

フォーマット: Status(association id)

→status data

このプリミティブは以下の情報を含むデータブロックを返却する

情報:

アソシエーション接続状態 (association connection state)、

送信先トランスポートアドレスリスト (destination transport address list)、

送信先トランスポートアドレス到着可能状態 (destination transport address reachability states)、

現在の受信者ウィンドウサイズ (current receiver window size)、

現在の輻輳ウィンドウサイズ (current congestion window sizes)、

非確認 DATA チャンク数 (number of unacknowledged DATA chunks)、

ペンディング受信 DATA チャンク数(number of DATA chunks pending receipt)、

プライマリパス (primary path)、

プライマリパス上の最新 SRTT (most recent SRTT on primary path)、

プライマリパス上の RTO (RTO on primary path)、

その他の相手先アドレス上の SRTT と RTO (SRTT and RTO on other destination addresses)

など。

必須属性:

association id ― SCTP アソシエーションへのローカルハンドル

オプション属性:

I) **なし。**

ハートビートの変更

フォーマット: CHANGEHEARTBEAT (association id, destination transport address, new state [,interval])
→result

ローカルエンドポイントに指定された送信先トランスポートアドレスの上でハートビートが利用可能であるか、あるいは利用不可であるかを示している。

このオペレーションの実行結果は返却される。

注: 利用可能であるときでも、送信先トランスポートアドレスがアイドルでないなら、ハートビートは発生しないであろう。

必須属性:

- association id ― SCTP **アソシエーションへのローカルハンドル**
- destination transport address ― **相手エンドポイントのトランスポートアドレスの** 1 **つとして指定される**
- new state ― **この送信先トランスポートアドレスにおけるハートビートの新しい状態(利用可能、あるいは不可能のどちらか)。**

オプション属性:

- interval ― **指定されている場合、これは送信先トランスポートアドレス上でハートビートを利用可能にするなら、ハートビートの周波数を示す。この値は送信先トランスポートアドレスの** RTO **に加えられる。指定されている場合、この値はすべての送信先に作用する。**

J) **ハートビートの要求**

フォーマット: REQUESTHEARTBEAT (association id ,destination transport address)
→result

所定のアソシエーションの送信先トランスポートアドレス上でローカルエンドポイントに HEARTBEAT **を行うよう指示する。返却された結果は相手アドレスへの** HEARTBEAT **チャンクの送信が成功しているかどうかを示さなければいけない。**

必須属性:

- association id ― SCTP **アソシエーションへのローカルハンドル**
- destination transport address ― **ハートビートが生じるべきアソシエーションのトランスポートアドレス。**

K) **ＳＲＴＴレポートの取得**

フォーマット: GETSRTTREPORT (association id, destination transport address)
→ srtt result

ローカルな SCTP **に所定のアソシエーションの指定された送信先トランスポートアドレス上における現在の** SRTT **測定を報告するよう指示する。**

返却された結果はミリ秒で、最新の SRTT **を含む整数であってもよい。**

必須属性:

- association id ― SCTP **アソシエーションへのローカルハンドル**
- destination transport address ― SRTT **測定が報告されるべきアソシエーションのトランスポートアドレス。**

L) **失敗のしきい値のセット**

フォーマット: SETFAILURETHRESHOLD (association id, destination transport address, failure threshold)
→result

このプリミティブはローカルな SCTP に指定送信先アドレスにおける到達可能性の異常検知しきい値「 Path.Max.Retrans 」をカスタマイズすることを許容する。

必須属性:

- association id － SCTP アソシエーションへのローカルハンドル
- destination transport address － 異常検知しきい値が設定されるべきアソシエーションのトランスポートアドレス。
- failure threshold － 送信先アドレスにおける「 Path.Max.Retrans 」の新しい値。

M) プロトコルパラメタのセット

フォーマット: SETPROTOCOLPARAMETERS(association id ［,destination transport address］,protocol parameter list)

→result

このプリミティブはローカル SCTP にプロトコルパラメタをカスタマイズすることを許容する。

必須属性:

- association id － SCTP アソシエーションへのローカルハンドル
- protocol parameter list － SCTP ユーザがカスタマイズすることを望む、指定された名称とプロトコルパラメタの値(例えば、 Association.Max.Retrans［2．14 節参照］)。

オプション属性:

- destination transport address － 1 つの送信先トランスポートアドレスを基準としていくつかのプロトコルパラメタが設定される。

N) 未送信メッセージの受信

フォーマット: RECEIVE_UNSENT (data retrieval id, buffer address, buffer size [,stream id] [,stream sequence number] [,partial flag] [,payload protocol-id])

- data retrieval id － 異常通知内で ULP に引き継がれる識別子。
- buffer address － 受信メッセージを格納するための ULP によって示された格納アドレス。
- buffer size － 受信データの最大サイズ(バイト)。

オプション属性:

- stream id － データがどのストリームに送られたかを示すために設定される戻り値である。
- ストリームシーケンス番号 － この値はメッセージと結び付けられたストリームシーケンス番号を示して返却される。
- partial flag － この返却されたフラグが 1 に設定されている場合、このメッセージは完全なメッセージの部分的な受信である。 このフラグが設定されるとき、stream id とストリームシーケンス番号がこの受信に伴わなくてはならない。このフラグが 0 に設定されるとき、これ以上のデリバリがこのストリームシーケンス番号のために受信されないということを示す。
- payload protocol-id － 転送するペイロードプロトコルデータのタイプを示す、相手から受け取られる 32 ビット符号無し整数。

O) 未到達確認メッセージの受診

フォーマット: RECEIVE_UNACKED (data retrieval id, buffer address, buffer size [,stream id] [,stream sequence number] [,partial flag] [,payload protocol-id])

- data retrieval id － 異常通知内で ULP に引き継がれる識別子。
- buffer address － 受信メッセージを格納するための ULP によって示された格納アドレス。
- buffer size － 受信データの最大サイズ(バイト)。

オプション属性:

- stream id － データがどのストリームに送られたかを示すために設定される戻り値である。

- ストリームシーケンス番号 － この値はメッセージと結び付けられたストリームシーケンス番号を示して返却される。
- partial flag － この返却されたフラグが 1 に設定されている場合、このメッセージはメッセージ全体の部分的なデリバリである。 このフラグが設定されるとき、stream id とストリームシーケンス番号がこの受信に伴わなくてはならない。このフラグが 0 に設定されるとき、これ以上のデリバリがこのストリームシーケンス番号のために受信されないということを示す。
- payload protocol-id － 転送するペイロードプロトコルデータのタイプを示す、相手から受け取られる 32 ビット符号無し整数。

P) ＳＣＴＰインスタンスの destroy

フォーマット:DESTROY(local SCTP instance name)

- local SCTP instance name － 初期化プリミティブの中でアプリケーションに引き継がれた値であり、またそれは削除されるべき SCTP インスタンスを示している。

2.10.2 SCTP から ULP へ

オペレーティングシステム又はアプリケーション環境が、SCTP から ULP プロセスに非同期に通知する 1 つの方法を想定する。SCTP が ULP プロセスに通知する時、あるきまった情報が ULP に引き継がれる。

実装上の注意: いくつかの場合においては、別々のソケットまたはエラーチャネルを通してこれらが行われる。

A) DATA ARRIVE 通知

SCTP はユーザからのメッセージを正常に受信し、復旧準備が整っている場合、この通知を ULP に対して起動する。

- association id － SCTP アソシエーションへのローカルハンドル
- stream id － どのストリーム上でデータ受信するかを示す

B) SEND FAILURE 通知

あるメッセージを送信することができない場合、SCTP はこの通知を ULP に対して起動する。

以下のパラメタがオプションとして通知メッセージとともに引き継がれる。

- association id － SCTP アソシエーションへのローカルハンドル
- data retrieval id － 未送信及び未確認データを復旧するために使用される識別子
- 原因コード － 異常の理由表示(例:サイズオーバ、メッセージライフタイム満了、など)
- context － 本メッセージに関連する付加情報(2.10.1 節 D)参照)

C) NETWORK STATUS CHANGE 通知

送信先トランスポートアドレスが非活性状態になった時(例えば SCTP が異常を検知)、または、活性状態になった時(例えば SCTP が復旧を検知)、SCTP は本通知を ULP に対して送信する。

以下のパラメタが通知メッセージとともに引き継がれる。

- association id － SCTP アソシエーションへのローカルハンドル
- destination transport address － この識別子は状態の変化により影響を受けた相手エンドポイントの送信先トランスポートアドレスである
- new-status － 新しい状態を示す

D) COMMUNICATION UP 通知

この通知は、SCTP がユーザメッセージの送受信の準備が整ったとき、またはエンドポイント間で失っていた通信が回復した時に使用される。

実装上の注意: ASSOCIATE プリミティブがブロッキング関数呼び出しとして実装されている場合、アソシエーションパラメタは ASSOCIATE プリミティブ自身の結果として返却される。その場合、COMMUNICATION

UP 通知アソシエーションの初期化側においてオプションである。

以下のパラメタが通知メッセージとともに引き継がれる。

- association id － SCTP アソシエーションへのローカルハンドル
- status － どのようなイベントが発生したかを示す
- destination transport address list － 相手のトランスポートアドレスの完全セット
- outbound stream count － ULP によってこのアソシエーションにて使用を許容されるストリームの最大数
- inbound stream count － このアソシエーションと共に相手エンドポイントが要求したストリーム数 (これは outbound stream count と同じ数ではない)

E) COMMUNICATION LOST 通知

SCTP がエンドポイントとの通信を完全に失った時(例えばハートビートによって)、またはエンドポイントがアボート処理を実行したことを検出した時、ULP に通知される。

以下のパラメタが通知メッセージと共に引き継がれる。

- association id － SCTP アソシエーションへのローカルハンドル
- status － どのようなイベントが発生したかを示す。この状態はシャットダウンまたはアボート要求への応答内で発生した異常や正常なターミネーションイベントを含む

以下のパラメタが通知メッセージに共に引き継がれる。

- data retrieval id － 未送信及び未確認データを復旧するために使用される識別子
- last-acked － 相手エンドポイントから最後に確認応答のあった TSN
- last-sent － 相手エンドポイントから最後に送信された TSN

F) COMMUNICATION ERROR 通知

SCTP が相手側から ERROR チャンクを受信し、それを ULP に通知すると判断した時、ULP に対してこの通知を起動することができる。

以下のパラメタが通知メッセージと共に引き継ぐことができる。

- association id － SCTP アソシエーションへのローカルハンドル
- error info － エラー種別及び ERROR チャンクを通して受信した幾つかの付加情報を示す

G) RESTART 通知

SCTP が、相手側が再開したことを検出した時、ULP に対してこの通知を送信してもよい。

以下のパラメタが通知メッセージと共に引き継ぐことができる。

- association id － SCTP アソシエーションへのローカルハンドル

H) SHUTDOWN COMPLETE 通知

SCTP がシャットダウン処理を完了した時、(2.9.2 節)この通知が上位レイヤに通知される。

以下のパラメタが通知メッセージと共に引き継ぐことができる。

- association id － SCTP アソシエーションへのローカルハンドル

2.11 セキュリティ

2.11.1 セキュリティの目的

ネットワーク上の 2 地点間で課金や電話サービスのためのシグナリングメッセージのように、time-sensitive なユーザメッセージを高信頼性の条件で運ぶために設計された共通の転送プロトコルとして、SCTP は以下のセキュリティ要件を持っている。

－ 高信頼でタイムリーなデータ転送サービスができること

－ SCTP で送られるユーザからユーザへの情報の完全性

2.11.2 潜在的な脅威に対する SCTP の対応

SCTP は、潜在的に広く様々なリスクを有する状況下で使用される。SCTP が走っているシステムのオペレータが、特異な状況を解析し、適切な対処策を決定することは重要である。

SCTP が走っているシステムのオペレータは、自分のサイトの安全を守るためのガイドラインとして、[RFC2196]を参照することを推奨する。

2.11.2.1 内部からの攻撃への対処

[RFC2196]はインサイダーによる、情報の剽窃あるいはサボタージュのリスクを最小限にするために適用される。この手順には、公開されているセキュリテイポリシ、物理的なアクセス・ソフトウェア・ネットワークレベルの制御、およびサービスの分離を含む。

2.11.2.2 ネットワーク上でのデータの歪みに対する防御

低いレイヤの転送サービスによって伝えられたデータグラムの中に見つけることができないエラーがあるというリスクがひどく重要な場合は、付加的な完全防御が要求される。もしこの付加的な防御がアプリケーションレイヤで提供されるのなら、SCTP ヘッダは完全性を突く攻撃に対して脆弱性を有している。パケット再送検知のための既存の SCTP メカニズムは、通常のオペレーションなら充分であると考えられているが、操作環境に熟練した相手からの慎重な攻撃に対して重大な危険を含んでいる場合、より強度の防御が SCTP を守るために必要とされる。

ソフトウェアコードの再利用を促進し、設定済みの環境機構の再構築を回避し、かつ、SCTP への無意味な複雑さを避けるため、セキュリティ脅威に曝されている環境がより強い完全性防御を求めるが機密性を求めない場合、IP Authentication Header[RFC2402]を使うことを推奨する。

広くインプリメントされた BSD ソケット API(BSD Sockets API)のための拡張(OS カーネルからの AH あるいは ESP のような)が IP セキュリティサービスを要求するアプリケーションのために適用される。アプリケーションは AH 使用が適切な場合は常に、AH を要求するためにそのような API を使用することができる。

2.11.2.3 機密性の保護

ほとんどの場合、SCTP またはより低いレイヤプロトコルのオーバヘッドでなく、シグナリングデータのペイロードの方が機密性侵害の危険性に曝される。それが正しいなら、SCTP ユーザデータの暗号化のみを考慮することを推奨する。補足のチェックサムサービスのように、ユーザデータ暗号化は、SCTP ユーザのアプリケーションによってなされてよい。

その代わり、ユーザアプリケーションは実装仕様に準拠した API を使用してよい。API は Encapsulating Security Payload (ESP)[RFC2406]が機密性と完全性を提供するために使われることを要求する。

とりわけモバイルのユーザのためには、機密性の要求の中に、IP アドレスとポートのマスキングも含まれる。この場合、ESP は、アプリケーションレベルの機密性の代わりに使用することを推奨する。ESP が SCTP トラフィックの機密性を保護するために使用される場合、暗号として完全に防御されている ESP 暗号転送が使われる。なぜなら、機密に対する脅威がある場合、強い完全性に対する脅威にもなるからである。

ESP が使用されている場合は常に、アプリケーションレベルの暗号化は一般的には要求されない。機密性がどこで提供されるかにかかわらず、ISAKMP[RFC2408]および Internet Key Exchange)(IKE)[RFC2409]をキー管理のために使用することを推奨する。

オペレータは、インターネットプロトコルレイヤの直上で利用可能なセキュリティサービスのより多くの情報については、[RFC2401]を参照することを推奨する。

2.11.2.4 Protecting against Blind Denial of Service Attacks

blind attack(盲目的攻撃)は、攻撃者が、対象の SCTP ノードから、もしくは、そこへのデータフローを中断したり見たりすることができないものである。サービスの盲目的妨害攻撃は、flooding、masquerade、もしくは不当な占有の形式をとる。

2.11.2.4.1 Flooding

flooding の目的は、サービスの消失、資源消耗による対象システムの正しくない振る舞い、合法的な処理やバッファに関連するソフトウェアバグの利用による妨害、を起こすことにある。Flooding は、SCTP ノード、あるいは、介在する IP アクセスリンク(IP Access Links)やインターネットの資源に向けられる。後者が目標である場合、flooding は、潜在的なファイヤウォールの突破を含む、ネットワークサービスの消失という現象で現われてくる。

一般に、flooding への防御は、設備設計レベルから始まる。ここでは、以下のやり方を含む:

- サービス要求が合法的であると決定する前に、有限な資源の割当を約束することを避ける
- 新しい作業の受領を行うために現在処理中のものへの優先権供与
- サービス要求の中で複製されたものもしくは古いものの識別と除去
- 予期していないパケットに対して、ユニキャストアドレス宛でない返答をしない

ネットワーク装置は、怪しげなトラヒック増加が見て取れた時には、アラームを発生し、ログを残すことができるようになっていることを推奨する。ログは、入って来るリンクやソースアドレスの識別子のような情報を提供するべきで、これは、ネットワークか SCTP システムのオペレータが防御の策を取るのに役立つ。悪用の明瞭なパターンが出現した場合に、オペレータがアラームのような適切な処理をとれるようにしておくことを推奨する。

Flooding に対抗するための SCTP の設計は、とりわけ、4 通りの初期化のハンドシェイクを用意していること、ハンドシェイクが完了するまでに SCTP ノードへの返答のためにリソースの割当確約はせずに cookie を使うこと、そして、確立しているアソシエーションのフローに外部からのパケットの割りこみを防止するために照合タグを使っていること、にある。

IP 認証ヘッダ(IP Authentication Header)、および、カプセル化されたセキュリティペイロード(Encapsulating Security Payload)は、ある種のサービス妨害攻撃のリスクを減少させるために有用である。

INIT チャンクの中の Host Name は、DNS サーバを flooding させるために使うこともできる。到達した INIT チャンクの中の Host Name から該当の IP アドレスを導き出すために、DNS の莫大なバックログを検索させることは、あるドメイン内の複数のホストへ INIT を送ることによって行える。さらに、攻撃者は、Host Name を使ってサードパーティへの間接的な攻撃を行うことができる。対象となるホスト名を含むホストへ無作為に多数の INIT を送るのである。DNS 資源への過負荷は、ターゲットへ多くの INIT ACK を送ることによっても実現できる。この種の攻撃への防御の 1 つに、DNS から送られてきた IP アドレスがオリジナルの INIT の中にある送信元の IP アドレスと同じかどうかを確認(verify)する方法がある。DNS から受信した IP アドレスが、INIT の送信元 IP アドレスを含んでいないなら、エンドポイントでは暗黙のうちに INIT を廃棄してよい。この最後のオプションは DNS への攻撃に対する防御とはならないであろう。

2.11.2.4.2 Blind Masquerade

masquerade(なりすまし)はいくつかの方法でサービスを否定するために使用される:

- 偽装(impersonated)ノードからのアクセスが制限されている SCTP ノードの資源と連携する方法。例えば、目標とするノードは、ポリシーによって、偽装される SCTP ノードとの間に、最大 1 つのアソシエーションを許可する。なりすましの攻撃者は、偽装されたノードからきたと主張してアソシエーションを確立しようとし、その結果、その後、その偽装されたノードがアソシエーションを確

立しようとしても確立できなくなってしまう。

- 偽装を故意に許可する方法。これによって、偽装されたノードが目標とする SCTP ノードからはじかれることを回避できる。
- －SHUTDOWN 要求のような、外部からのコンテンツの割りこみによって確立しているアソシエーションの邪魔をする方法

SCTP は、4 通りの初期化ハンドシェイクを用いる IP spoofing によって、盲目的 masquerade 攻撃からのリスクを減少させる。仲介者なりすまし(Man-in-the-middle masquerade)による攻撃は、以下の 2.11.3 節で議論されている。初期の交換にはメモリを必要としないので、盲目的ななりすまし攻撃によって、ロックアウトメカニズムのトリガはかからない。さらに、状態クッキーを含んでいる INIT ACK は、INIT を受信した IP アドレスに送信される。したがって、攻撃者は、状態クッキーを含んでいる INIT ACK を受け取らない。SCTP は、照合タグを用いて、確立しているアソシエーションのフローの中へ外部のパケットを挿入することを防いでいる。

受信した INIT 要求のログと予期しない INIT ACK のような異常ログから、敵(攻撃者)の活動パターンを検知する方法を考えることができる。しかしながら、そのようなログの潜在的な有用性は、増加する SCTP 初期化処理に対して考慮されるもので、SCTP ノードの flooding 攻撃に対する脆弱性をより露呈することになる。ログは、常にレビューし分析するというオペレータの手続きを確立させることなくして意味あるものにはならない。

2.11.2.4.3 サービスの不正な占有

この攻撃は、攻撃者によって公然かつ合法的に行われる。攻撃は目標となる SCTP ノードのユーザ、あるいは、攻撃者と目標ノードの資源を共有しているユーザに対してなされる。可能である攻撃は、攻撃者のノードと目標の間で大量のアソシエーションを開始し、あるいは、合法的に確立したアソシエーションの中で大量の情報を送るものである。

SCTP ノード当たりのアソシエーションの数を制限するポリシーを与えることを推奨する。SCTP ユーザのアプリケーションは、大量の、もしくは、何も処理をしない”no-op”メッセージを検出することを推奨する。その結果として、ローカルポリシーに従い、ログを取るかアソシエーションを終了させることを推奨する。

2.11.3 Protection against Fraud and Repudiation

fraud(詐称)の目的は、認証なしに、とりわけ、それらの代価を払わずに、サービスを獲得することである。この目的を達成するように、攻撃者は、システムが無効の課金データを受理するか、それを集めることに失敗している間に、目標の SCTP ノード上の SCTP ユーザアプリケーションから、欲しいサービスの提供を受けるに違いない。Repudiation(支払い拒絶)は、それが故意の fraud 行為として起こるか、もしくは単に Repudiation するものが、不適切なサービスレコードを受信し続けるために、fraud に関連する問題である。潜在的な fraud 攻撃には、クレジットカード番号のような個人情報の取得や不正利用、盲目的な masquerade、replay、目標とする SCTP アソシエーションを通るパケットをリアルタイムに変更する仲介者攻撃、がある。

インターセプト攻撃は、上記の 2.11.2.3 節で議論された機密保持の手法によって対抗できる。

2.11.2.4.2 節には、4 通りの初期化ハンドシェイクおよび照合タグの結果として、SCTP が、どのようにして盲目的 masquerade 攻撃に対抗しているかが述べられている。照合タグと TSN は、ともに、盲目的 replay 攻撃に対する対抗策である。ここでは replay は、既存のアソシエーションに入り込む場合である。

しかしながら、SCTP は、攻撃者がアソシエーションの中でやり取りされているパケットをインターセプトしたり変えたりできる、仲介者攻撃に対する防御はしていない。例えば、INIT ACK は、相手に送られた情報で、既存の SCTP アソシエーションを途中でハイジャックするに足りる充分な情報を持っている。そのような攻撃が見られるところでは、あるいは、repudiation が行われうるところでは、送られるパケットの整

合性と認証性の両方を保証するために IPSEC AH サービスの使用が推奨される。

SCTP は、SCTP でもしくは SCTP より上で発生する攻撃や、既存のアソシエーションのコンテキスト内で起きる攻撃への防御をも行っていない。2.11.2.1 節で議論されているように、そのような攻撃への防御は、ホスト側で適切なセキュリティ対策を取ることによってカバーすることを推奨する。

2.12 TCB パラメタ

このセクションは、推奨するパラメタセットについて詳細に説明する。このパラメタセットは実装時に TCB(トランスミッションブロック)に含めるべきものである。このセクションは目的を明示しようとしているだけであって、実装時の要求を示しているわけではないし、SCTP TCB 中の全パラメタのリストを示そうとしているのでもない。各々の実装では、最適化のために独自に追加するパラメタが必要であろう。

2.12.1 SCTP インスタンスのために必要なパラメタ

アソシエーション:現在のアソシエーションのリストであり、各アソシエーションのデータ利用者へのマッピングである。ハッシュ表もしくは構造に依存する他の実装である。データの利用者は、ファイル記述子や名前付パイプのポインタ、どのように SCTP が実装されているかによるが表のポインタのような識別情報を使う。

秘密鍵: エンド端末によって MAC アドレスを計算するために使われる秘密鍵。

2.12.2 アソシエーション(即ち TCB)毎に必要なパラメタ

Peer Verification Tag:各パケット内で送られるタグ値であり、INIT もしくは INIT ACK の中で受信する。

My Verification Tag:各受信パケットにあることが想定され、INIT もしくは INIT ACK 値チャンクの中で送られる。

State: アソシエーションがどの状態であるかを示す状態変数。COOKIE-WAIT, COOKIE-ECHOED, ESTABLISHED, SHUTDOWN-PENDING, SHUTDOWN-SENT, SHUTDOWN-RECEIVED, SHUTDOWN-ACK-SENT 内で使われる。

注:アソシエーションが"CLOSED"なら、その TCB は削除されるべきなので、"CLOSED"の状態は示されない。

Peer Transport Address List:経由する SCTP トランスポートアドレスのリスト。情報は INIT もしくは INIT ACK でもたらされ、アソシエーションと受信パケットを関係付けるために使われる。

Primary Path:現在最初に選択される宛て先トランスポートアドレス。送信トランスポートアドレスも記載する。

Overall Error Count:全アソシエーションのエラー数。

Overall Error Threshold: Overall Error Count がこれに到達した場合、アソシエーションを破棄するようなしきい値。

Peer Rwnd: rwnd の現在の計算値。

Next TSN:新しい DATA チャンクにアサインすべき次の TSN 番号。相手へ INIT もしくは INIT ACK で送られ、DATA チャンクに TSN がアサインされる毎に更新される(通常は送信するより前に、もしくは分割中に)。

Last Rcvd TSN:シーケンスの中で最後に受信した TSN。相手から最初に初期 TSN を受け取ることによってセットされる。INIT もしくは INIT ACK チャンクで受け取り、受信した値から 1 を減算する。

Mapping Array: ビットもしくはバイトの配列であり、(Last Rcvd TSN に関わって)どの TSN の順番が乱れて受け取られているかを示す。ギャップがない場合、即ち、順番どおりにパケットを受け取っている場合、配列は全て 0 になる。構造は循環バッファもしくはビットアレーの形態を取る。

Ack State:次に受信したパケットが SACK で返事を出すかどうかを示す。0 に初期化される。パケットを受信した時、更新される。この値が 2 以上になったなら、SACK が送られ値が 0 にリセットされる。注:順番を乱

さずに DATA チャンクを受信したときに使われる。順番が乱れた時は、SACK は遅れない(2.6 節参照)。
Inbound Stream:受信ストリームを探すための構造体の配列。通常は、想定される次のシーケンス番号と可能なストリーム番号を含む。
Outbound Stream:送信ストリームを探すための構造体の配列。通常は、そのストリームで送られる、次のシーケンスの番号を含む。
Reasm Queue:再統合されたキュー。
Local Transport Address List:このアソシエーション内で交わされたローカル IP アドレスのリスト。
アソシエーション パス MTU: 全宛て先トランスポートアドレス内のパス MTU の最小値。

2.12.3 トランスポートアドレスのデータ

INIT もしくは INIT ACK チャンクで届けられる通信相手のアドレスリスト中の各宛て先トランスポートアドレス用に、以下の幾つかのデータ要素を管理する必要がある。:
Error count:当該宛て先の現在のエラー数。
Error Threshold:当該宛て先のための現在のエラーしきい値。即ち、エラー数がこの値になれば、宛て先がダウンする。
Cwnd:現在の輻輳幅。
Ssthresh:現在の ssthresh 値。
RTO:現在の再送タイムアウト値。
SRTT:現在の平穏時のラウンドトリップ時間
RTTVAR:現在の RTT 変動。
Partial bytes acked:輻輳回避時(セクション 6.2.2 を見よ)の cwnd 更新のための探索方法。
State:当該宛て先の現在の状態。即ち、DOWN、UP、ALLOW-HB、NO-HEARTBEAT、等。
パス MTU:現在わかっている path MTU。
Per Destination Timer:各宛て先によって使われるタイマ。
RTO-Pending:ある DATA チャンクが当該アドレスに送付され、RTT を計算するために現在使われている場合、探すために使われるフラグ。このフラグが 0 なら、この宛て先に送られる次の DATA チャンクは RTT を計算するのに使われ、このフラグがセットされることを推奨する。毎回 RTT 計算はこのフラグをクリアする(即ち DATA チャンクは SACK を送られる)。
Last-time used:この宛て先へ最後に送られた時間。HEARTBEAT が必要な場合に決定される。

2.12.4 必要とする汎用パラメタ

Out Queue:発信 DATA チャンクのキュー
In Queue:受信 DATA チャンクのキュー

2.13 IANA Consideration～登録番号

このプロトコルは、インターネット内の既知のサーバへアクセスするため、TCP のようなポート予約を必要とする。全ての現在の TCP ポートは、SCTP ポートアドレス空間内に自動的に予約される。新しい要求は、TCP のために作られた IANA の現在のメカニズムに従う。

このプロトコルは、IANA の 3 つの方法で拡張される場合もある。

- 追加するチャンクタイプの定義による方法
- 追加するパラメタタイプの定義による方法
- ERROR チャンクの中にある理由コードの追加の定義による方法

SCTP を使用しているある ULP が独自のポートを要求している場合、アサインされたポートを獲得するた

めの IANA への登録はその ULP が責任を持つことを推奨する。

2.13.1 IETF 定義のチャンク拡張

新しいチャンクの型の定義と使用は SCTP に必要な部分である。新しいチャンクの型は、[RFC2434]で定義されているように、IANA によって IETF のコンセンサスアクションに従ってアサインされる。

新しいチャンクコードの型のための文書は以下の情報を含む:

a) 新しいチャンクの型のための長い名前と短い名前；

b) チャンクの構造の詳細な定義であり、これはセクション 3.2 で定義している基本構造に従う；

c) チャンクのフィールドで使用する予定のものの詳細な定義と記述で、チャンクフラグを含む；

d) プロトコルの操作における新しいチャンクの型の使用についての詳細な手続きの記述

可能な場合は、最後のチャンクの型(255)は将来の使用のために確保される。

2.13.2 IETF 定義のチャンクパラメタの拡張

新しいチャンクパラメタの型コードの定義は、[RFC2434]に定義されているように、IETF コンセンサスアクションによってなされている。チャンクパラメタの文書は以下の情報を含んでいる:

a) パラメタの型の名前

b) パラメタフィールドの構造の詳細な記述。この構造は、セクション 3.2.1 に記載されている一般的な type-length-value フォーマットに従う。

c) パラメタ値の各要素の詳細な記述。

d) このパラメタの型で使用されることが予定されているものの詳細な記述。同じチャンクの中で、どんな状態のときにどのパラメタの型の例が見出せるかの指針。

2.13.3 IETF 定義の追加エラー原因

追加エラー原因は、[RFC2434]で定義されている Specification 要求アクションによって、11 から 65535 の範囲に位置付けられる。提供する文書には以下の情報が記載される。

a) エラー状態の名前

b) その状態の詳細な記述。この状態では、SCTP エンドポイントは、このエラーコードを付けて ERROR (もしくは ABORT) を送る。

c) この原因コードを含んだ ERROR (もしくは ABORT) チャンクを受信した SCTP エンドポイントによって採られる想定される動作。

d) この原因コードを伴っているデータフィールドの構造体とその中身についての詳細な記述。

原因コードパラメタの最初のワード(32 ビット)は、2.3.3.10 節に示されているフォーマットに従う。つまり、

- 最初の 2 バイトには原因コードの値を含める
- 続く 2 バイトには原因パラメタの長さを含める

2.13.4 ペイロードプロトコルの識別子

0 の値を除き(これは SCTP によってリザーブされている値であり、DATA チャンク内で未定義のペイロードプロトコルの識別子を示している)、SCTP はペイロードプロトコルの識別子を規定もしくは確認することはしない。SCTP は単に上位レイヤから識別子を受け取り、関連するペイロードデータと一緒に運ぶだけである。

上位レイヤ、つまり SCTP のユーザは、必要ならば IANA と共に、全てのプロトコル識別子を規定することを推奨する。ペイロードプロトコル識別子の使用については SCTP のスコープ外である。

2.14 プロトコルパラメタの推奨値

以下のプロトコルパラメタは推奨値である:

RTO.Initial	3 秒
RTO.Min	1 秒
RTO.Max	60 秒
RTO.Alpha	1/8
RTO.Beta	1/4
Valid.Cookie.Life	60 秒
Association.Max.Retrans	10 回
Path.Max.Retrans	5 回 (宛先アドレスあたり)
Max.Init.Retransmits	8 回
HB.interbal	30 秒

実装時の注: SCTP を実装する時は、上位プロトコルはこれらプロトコルパラメタの幾つかを変更してよい(セクション 10 を見よ)。

注:RTO.Min はここで推奨した値をセットすることを推奨する。

３．UA 共通規定

本章では、ユーザアダプテーションプロトコル(M3UA、M2UA、M2PA、IUA、SUA)の共通仕様を規定する。「UA」はアダプテーションプロトコル一般を意味するものとする。

3.1 ネットワークアーキテクチャ

3.1.1 信号網アーキテクチャ

信号ゲートウェイ(SG ： Signalling Gateway)は、ユーザ部プロトコルの信号トラフィックを MGC(Media Gateway Controller)等の ASP(Application Server Process)へ転送する。UA 自体は、性能条件と信頼性条件を保証しないが、分散アーキテクチャと冗長ネットワークの採用によって IP 上の信号トラフィック転送は十分な性能と信頼性を得られる。UA は多様なネットワーク構成で運用可能であり、ネットワークオペレータが要求する性能条件と信頼性条件を満足することができる。

キャリアグレードネットワークの性能条件と信頼度条件を達成するためには、ネットワークに単一故障点が存在しないことが必要条件である。SG と ASP の信頼性にも依存するが、QoS が保証された冗長なネットワーク上で SCTP アソシエーションを設定し、SG と ASP に冗長性を持たせることでキャリアグレードネットワークの性能条件と信頼度条件を達成することができる。ASP を複数ホストに分散配備することは重要である。AS を構成する ASP は、最低でも 2 台のホストに分散配備することを強く推奨する。

キャリアグレードの運用に適当な論理ネットワークの例を以下の図 3-1 に示す。

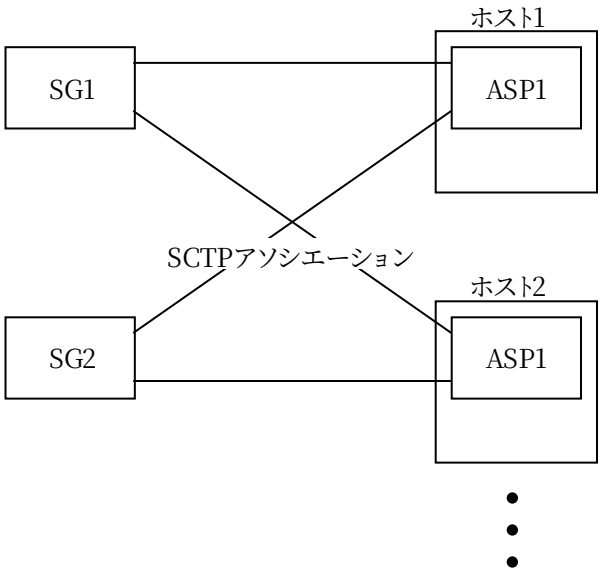


図 3-1 ネットワークの論理構成

キャリアグレードネットワークにおいては、単一 ASP が障害または孤立した際に安定呼を救済することが求められる。安定呼救済を実現するためには、ASP 間での呼状態共有または呼状態交換が必要である。しかし、呼状態共有／交換は UA の規定対象外である。

3.1.2 ASP のフェイルオーバ

アプリケーションサーバ(AS：Application Server)は、同一目的に供する一連の ASP である。信号トラフィックを処理する ASP を稼動状態 ASP と呼ぶ。通常は利用しないが、稼動状態 ASP が故障または利用不能の際に使用する ASP もある。

フェイルオーバモデルは n+k 冗長モデルをサポートする。ここで n は信号トラフィック処理に必要な最小限の ASP 数、k は障害または利用不能となった ASP の代わりに信号トラフィックを処理する ASP 数である。1+1 の現用／予備冗長構成は本モデルのサブセットである。冗長性の無い 1+0 モデルもサブセットである。

単一障害が全体に波及することを防ぐために、AS には最低でも２個の ASP を用意し、それぞれを別の物理ホストに配備することを推奨する。ホストを分けることにより、必然的に SCTP アソシエーションも別になる。例えば、図 3-1 のネットワークにおいて、インタフェース識別子群の信号トラフィックを担当する AS1 がホスト 1 の ASP1 とホスト 2 の ASP1 から構成されるものとする。AS1 は SG 上で図 3-2 のように表現される。

インタフェース識別子群	――――	AS1
ASP1/ホスト1	――――	稼動状態
ASP1/ホスト2	――――	起動しているが信号トラヒックを処理していない状態

図 3-2 AS の構成(その 1)

これは 1+1 冗長構成のケースであり、ASP1/ホスト 1 は AS の全ての信号トラフィックを処理する。ASP1/ホスト 2 は、ASP1/ホスト 1 の故障発生時、または SG－ASP1/ホスト 1 間の通信途絶時に稼動状態に遷移する。

AS は負荷分散構成で運用することも可能である(図 3-3 参照)。

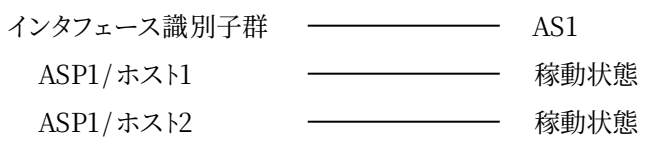


図 3-3 AS の構成(その 2)

このケースでは､両方の ASP1/ホスト 1 と ASP1/ホスト 2 は AS の信号トラフィックを分担して処理する。

フェイルオーバ時には安定呼を救済することが望ましい。状態遷移中の呼については、ASP 相互の情報共有によりある程度は救済可能であるが、呼解放が発生しても止むを得ない。ASP 相互の情報共有の例としては、共有メモリ経由での呼状態共有や、ASP 間通信による呼状態交換などがある。ASP 間通信は UA の規定対象外である。

3.2 用語

3.2.1 アソシエーション

SCTP アソシエーションを意味する。UA ユーザのプロトコルデータユニットや UA メッセージの転送に使用する。

3.2.2 AS

アプリケーションサーバ (Application Server)。アプリケーションインスタンスを提供する論理エンティティであり、AS と表記する。AS の例としては、SG で終端した Dch の呼設定と Q.931 を処理する MGC がある。SG からみると、AS は ASP (3.2.3) のリストとして表現される。

3.2.3 ASP

アプリケーションサーバプロセス (Application Server Process)。AS のプロセスインスタンスであり、ASP と表記するプライマリ MGC とバックアップ MGC が ASP の例である。

3.2.4 ストリーム

SCTP ストリームを意味する。SCTP ストリームは SCTP エンドポイント間の単方向論理チャンネルのことであり、通常は順序制御を行う。

3.2.5 ネットワークアピアランス

ネットワークアピアランスは、複数の共通線信号網と接続する SG において網を特定する識別子である。ネットワークアピアランスの値は、ASP と SG の組ごとに異なってよい。

3.2.6 ネットワークバイトオーダ

最上位ビットがはじめに来るビットの並べ方であり、ビッグエンディアンと同義である。

3.2.7 フェイルオーバ

使用中の ASP に障害が発生し、または利用不可能な状態になった時に、ASP 間で信号トラフィックを切り替えることである。プライマリ MGC からバックアップ MGC へのフェイルオーバが代表例である。フェイルオーバは切り戻しも含めて云う。

3.2.8 ホスト

ASP の稼動するコンピュータである。

3.2.9 レイヤ管理

レイヤ管理は UA レイヤとローカル管理間の入出力を処理する機能である。

3.2.10 ルーティングキー

ルーティングキーは、AS が担当する信号トラフィックを定義するパラメタ群である。ルーティングキーの着信号局コードは高々一つである。

3.2.11 ルーティングコンテキスト

ルーティングキーを識別する 32 ビット符号無し整数である。

3.3 サービス

本節では、アダプテーションプロトコルが上位レイヤに提供するサービスを規定する。

3.3.1 SCTP 管理サービス

SCTP 管理サービスは、SCTP アソシエーションの確立／解放／状態確認を実現するサービスである。プリミティブ一覧を表 3-1 に示す。

表 3-1 SCTP 管理サービスプリミティブ一覧

プリミティブ		概要
M-SCTP 設定	要求 指示 確認	SCTP アソシエーションを確立するために使用する。
M-SCTP 解放	要求 指示 確認	SCTP アソシエーションを解放するために使用する。
M-SCTP 状態	要求 確認	SCTP アソシエーションの状態を確認するために使用する。

3.3.2 UA 管理サービス

UA 管理サービスは、UA の管理機能を提供するサービスである。プリミティブ一覧を表 3-2 に示す。

表 3-2 UA 管理サービスプリミティブ一覧

サービスプリミティブ		概要
M-ERROR	指示	エラー状態通知に使用する。
M-NOTIFY	指示	イベント通知に使用する。
M-TEI 状態	要求 指示 確認	TEI 状態の問い合わせおよび状態変更通知に使用する。

3.3.3 ASP 管理サービス

ASP 管理サービスは、ASP の状態およびトラフィックを管理するサービスである。プリミティブ一覧を表 3-3 に示す。

表 3-3 ASP 管理サービス一覧

サービスプリミティブ		概要
M-ASP UP	要求 確認	ASP 状態を停止状態から起動状態に遷移するために使用する。
M-ASP DOWN	要求 確認	ASP 状態を起動状態から停止状態に遷移するために使用する。
M-ASP ACTIVE	要求 確認	ASP 状態を起動状態から稼動状態に遷移するために使用する。
M-ASP INACTIVE	要求 確認	ASP 状態を稼動状態から起動状態に遷移するために使用する。
M-ASP_STATUS	要求 指示	ASP 状態の問い合わせ及び通知に使用する。

3.3.4 AS 管理サービス

AS 管理サービスは、AS 状態変更を通知するサービスである。プリミティブ一覧を表 3-4 に示す。

表 3-4 AS 管理サービス一覧

サービスプリミティブ		概要
M-AS STATUS	要求 指示	AS 状態の問い合わせ及び通知に使用する。

3.3.5 MTP3 サービス

5.2.5 に記述する。

3.3.6 Q.921 サービス

4.3.5 に記述する。

3.3.7 MTP2 サービス

6.3.5 に記述する。

3.3.8 SCCP コネクションレスサービス

8.3.5 に記述する。

3.3.9 SCCP コネクション指向サービス

SUA (SCCP User Adaptation) が使用するサービスであるが、ＴＴＣ版 SUA は、このサービスを規定していないため本文書の範囲外とした。

3.4 メッセージ

UA のメッセージは、共通メッセージヘッダ、個別メッセージヘッダ、パラメタから構成される。共通メッセージヘッダは、全ての UA メッセージが使用する。個別メッセージヘッダの定義は UA 毎に異なる。また、個別メッセージヘッダは用いなくてもよい。パラメタは 0 個以上繰り返すことが出来る。共通メッセージヘッダ、個別メッセージヘッダ、可変パラメタの長さはいずれも 32 ビットの整数倍とする。メッセージのフォーマットを図 3-4 に示す。

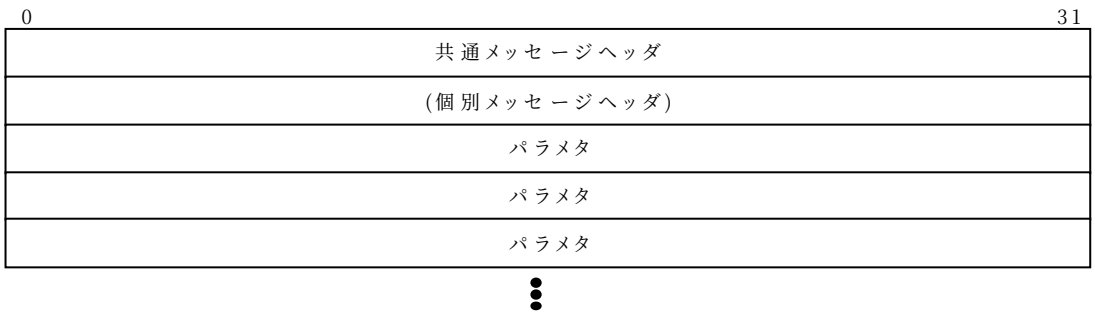


図 3-4 UA メッセージのフォーマット

3.4.1 共通メッセージヘッダ

アダプテーションプロトコルのメッセージは共通メッセージヘッダを使用する。共通メッセージヘッダのフォーマットを図 3-5 に示す。

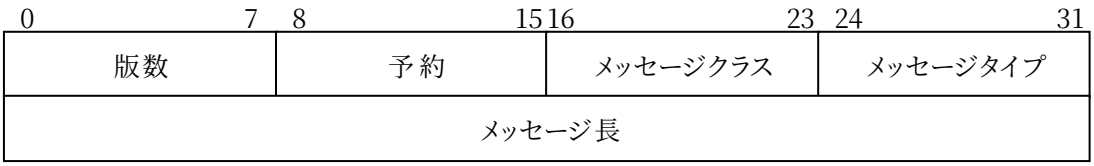


図 3-5 共通メッセージヘッダのフォーマット

3.4.1.1 版数

版数フィールドは UA の版数を示す 8 ビット符号無し整数である。各 UA の最新版数を表 3-5 に示す。

表 3-5 UA の最新版数

アダプテーションプロトコル名	最新版数
M3UA	1
M2UA	1
M2PA	1
IUA	1
SUA	1

3.4.1.2 予約

将来の拡張用として 8 ビットを予約する。送信側は全てのビットを 0 に設定する。受信側は無視する。

3.4.1.3 メッセージクラス

UA メッセージは、メッセージクラスとメッセージタイプの組み合わせで識別する。メッセージクラスはメッセージを分類する 8 ビット符号無し整数である。メッセージクラス一覧を表 3-6 に示す。

表 3-6 メッセージクラス一覧

メッセージクラス	値
UA 管理メッセージクラス	0x00
MTP3 メッセージクラス	0x01
共通線信号網管理メッセージクラス	0x02
ASP 状態管理メッセージクラス	0x03

メッセージクラス	値
ASP トラフィック管理メッセージクラス	0x04
Q.921 メッセージクラス	0x05
MTP2 メッセージクラス	0x06
SCCP コネクションレスメッセージクラス	0x07
SCCP コネクション指向メッセージクラス	0x08
ルーティングキー管理メッセージクラス	0x09
インターフェース識別子管理メッセージクラス	0x0A
(予約)	0x0B – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4 メッセージタイプ

メッセージタイプはメッセージクラスと組み合わせてメッセージ種別を特定する。メッセージタイプは 8 ビット符号無し整数で表現する。

3.4.1.4.1 UA 管理メッセージクラスのメッセージタイプ

UA 管理メッセージクラスのメッセージタイプ一覧を表 3-7 に示す。

表 3-7 UA 管理メッセージクラスのメッセージ一覧

メッセージタイプ	値
ERR メッセージ	0x00
NTFY メッセージ	0x01
TEI 状態要求メッセージ	0x02
TEI 状態確認メッセージ	0x03
TEI 状態指示メッセージ	0x04
(予約)	0x05 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.2 MTP3 メッセージクラスのメッセージタイプ

MTP3 メッセージクラスのメッセージタイプ一覧を表 3-8 に示す。

表 3-8 MTP3 メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
ペイロードデータメッセージ	0x01
(予約)	0x02 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.3 共通線信号網管理メッセージクラスのメッセージタイプ

共通線信号網管理メッセージクラスのメッセージタイプ一覧を表 3-9 に示す。

表 3-9 共通線信号網管理メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
DUNA メッセージ	0x01
DAVA メッセージ	0x02
DAUD メッセージ	0x03
SCON メッセージ	0x04
DUPU メッセージ	0x05
DRST メッセージ	0x06
(予約)	0x07 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.4 ASP 状態管理メッセージクラスのメッセージタイプ

ASP 状態管理メッセージクラスのメッセージタイプ一覧を表 3-10 に示す。

表 3-10 ASP 状態管理メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
ASPUP メッセージ	0x01
ASPDN メッセージ	0x02
BEAT メッセージ	0x03
ASPUP ACK メッセージ	0x04
ASPDN ACK メッセージ	0x05
BEAT ACK メッセージ	0x06
(予約)	0x07 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.5 ASP トラフィック管理メッセージクラスのメッセージタイプ

ASP トラフィック管理メッセージクラスのメッセージタイプ一覧を表 3-11 に示す。

表 3-11 ASP トラフィック管理メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
ASPAC メッセージ	0x01
ASPIA メッセージ	0x02
ASPAC ACK メッセージ	0x03
ASPIA ACK メッセージ	0x04
(予約)	0x05 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.6 Q.921 メッセージクラスのメッセージタイプ

Q.921 メッセージクラスのメッセージタイプ一覧を表 3-12 に示す。

表 3-12 Q.921 メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
データ要求メッセージ	0x01
データ指示メッセージ	0x02
ユニットデータ要求メッセージ	0x03
ユニットデータ指示メッセージ	0x04
設定要求メッセージ	0x05
設定確認メッセージ	0x06
設定指示メッセージ	0x07
解放要求メッセージ	0x08
解放確認メッセージ	0x09
解放指示メッセージ	0x0A
(予約)	0x0B – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.7 MTP2 メッセージクラスのメッセージタイプ

MTP2 メッセージクラスのメッセージタイプ一覧を表 3-13 に示す。

表 3-13 MTP2 メッセージクラスのメッセージ一覧

メッセージタイプ	値
(予約)	0x00
データメッセージ	0x01
リンク設定要求メッセージ	0x02
リンク設定確認メッセージ	0x03
リンク解放要求メッセージ	0x04
リンク解放確認メッセージ	0x05
リンク解放指示メッセージ	0x06
リンク状態要求メッセージ	0x07
リンク状態確認メッセージ	0x08

リンク状態指示メッセージ	0x09
回収要求メッセージ	0x0A
回収確認メッセージ	0x0B
回収指示メッセージ	0x0C
回収完了指示メッセージ	0x0D
輻輳通知メッセージ	0x0E
データ応答メッセージ	0x0F
(予約)	0x16 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.8 M2PA メッセージクラスのメッセージタイプ

M2PA メッセージクラスのメッセージ一覧を表 3-14 に示す。

表 3-14 M2PA メッセージクラスのメッセージ一覧

メッセージタイプ	値
ユーザデータ	1
リンク状態	2
Proving Data	3

3.4.1.4.9 SCCP コネクションレスメッセージクラスのメッセージタイプ

SCCP コネクションレスメッセージクラスのメッセージ一覧を表 3-15 に示す。

表 3-15 SCCP コネクションレスメッセージクラスのメッセージ一覧

メッセージタイプ	値
(予約)	0x00
CLDT メッセージ	0x01
CLDR メッセージ	0x02
(予約)	0x03 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.10 SCCP コネクション指向メッセージクラスのメッセージタイプ

SCCP コネクション指向メッセージクラスのメッセージタイプ一覧を表 3-16 に示す。

表 3-16 SCCP コネクション指向メッセージクラスのメッセージタイプ一覧

メッセージタイプ	値
(予約)	0x00
CORE メッセージ	0x01
COAK メッセージ	0x02
COREF メッセージ	0x03
RELRE メッセージ	0x04
RELCO メッセージ	0x05
RESCO メッセージ	0x06
RESRE メッセージ	0x07
CODT メッセージ	0x08
CODA メッセージ	0x09
COERR メッセージ	0x0A
COIT メッセージ	0x0B
(予約)	0x0C – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.11 ルーティングキー管理メッセージクラス

ルーティングキー管理メッセージクラスのメッセージ一覧を表 3-17 に示す。

表 3-17 ルーティングキー管理メッセージクラスのメッセージ一覧

メッセージタイプ	値
(予約)	0x00
登録要求メッセージ	0x01
登録応答メッセージ	0x02
解除要求メッセージ	0x03
解除応答メッセージ	0x04

メッセージタイプ	値
(予約)	0x05 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.4.12 インタフェース識別子管理メッセージクラス

インタフェース識別子管理メッセージクラスのメッセージ一覧を表 3-18 に示す。

表 3-18 インタフェース識別子管理メッセージクラスのメッセージ一覧

メッセージタイプ	値
(予約)	0x00
登録要求メッセージ	0x01
登録応答メッセージ	0x02
解除要求メッセージ	0x03
解除応答メッセージ	0x04
(予約)	0x05 – 0x7F
(拡張用)	0x80 – 0xFF

3.4.1.5 メッセージ長

メッセージ長をオクテット単位で示す 32 ビット符号無し整数である。メッセージ長には共通メッセージヘッダも含める。メッセージ長はパラメタのパディングバイトを含める。M2UA の場合、メッセージ長は共通ヘッダと M2UA 共通ヘッダにＭＴＰ３メッセージを加えたものを超えない。

3.4.2 個別メッセージヘッダ

個別メッセージヘッダは UA 毎に定義する。

3.4.3 パラメタ

アダプテーションプロトコルのメッセージは、共通メッセージヘッダ、個別メッセージヘッダとパラメタから構成される。パラメタのフォーマットを図 3-6 に示す。

図 3-6 パラメタのフォーマット

3.4.3.1 **パラメタタグ**

パラメタの種別を示す 16 **ビット符号無し整数であり、**0 **から** 65534 **までの値を取る。**65535 **は将来の拡張用として予約する。パラメタタグは** UA **毎に定義するので、同一のパラメタタグに対して複数パラメタが割り当てられることがある。例えば、パラメタタグ** 0x0001 **は** M3UA **と** SUA **では「ネットワークアピアランス」を表すが、**M2UA **と** IUA **では「整数型インタフェース識別子」を表す。**

パラメタ名	パラメタ値
(予約)	0x00
インターフェース識別子（符号なし整数）	0x01
未使用	0x02
インターフェース識別子（テキスト）	0x03
付加情報	0x04
未使用	0x05
未使用	0x06
診断情報	0x07
インタフェース識別子（整数範囲）	0x08
Beat Data	0x09
未使用	0x0a
トラヒックモードタイプ	0x0b
エラーコード	0x0c
状態タイプ／情報	0x0d
未使用	0x0e
未使用	0x0f
未使用	0x10
ASP 識別子	0x11
未使用	0x12
Correlation Id	0x13
(予約)	0x14 – 0xFF

3.4.3.2 パラメタ長

パラメタ長をオクテット単位で示す 16 ビット符号無し整数である。パラメタ長は、パラメタ値以外にもパラメタタグフィールドとパラメタ長フィールドも含む。パラメタ値をパディングする場合、パラメタ長はパディングを含まない。

3.4.3.3 パラメタ値

パラメタ値が 4 バイト境界に整列しない場合、末尾を 0x00 でパディングする。パディングは 3 バイト以下に留めることを推奨する。パラメタ長はパディングを含まない。受信側はパディングを無視する。

3.4.4 UA 管理メッセージクラス

UA 管理メッセージクラスに所属するメッセージは、0 番の SCTP ストリームを使用して転送することを推奨する。

3.4.4.1 ERR メッセージ (ERRor)

ERR メッセージは、受け付けたメッセージに対するエラーイベントを送信元 UA に通知するために使用する。例として、予期しないタイプのメッセージを受信した場合や、パラメタ値が不正だった場合などである。ERR メッセージは共通メッセージヘッダのみを持つ。IUA の場合、ERR メッセージは以下のパラメタを含む:

- エラーコード（必須）
- 診断情報（省略可能）

IUA の ERR メッセージのフォーマットを図 3-7 に示す。

図 3-7 ERR メッセージのフォーマット

その他の ERR メッセージについては各 UA 毎に定義する。

3.4.4.2 NTFY メッセージ (Notify)

NTFY メッセージは、AS 状態遷移を通知するために SG から ASP に送信する。NTFY メッセージのフォーマットは UA 毎に異なる。

3.4.4.2.1 IUA の NTFY メッセージ

IUA の NTFY メッセージは、共通メッセージヘッダのみを使用し、以下のパラメタを含む:

- ステータスタイプ／ステータス識別（必須）
- インタフェース識別子（省略可能）

- **付加情報（省略可能）**

インタフェース識別子パラメタは整数型と文字列型がある。整数型インタフェース識別子パラメタを使用する場合の NTFY **メッセージフォーマットを図** 3-8 **に示す。**

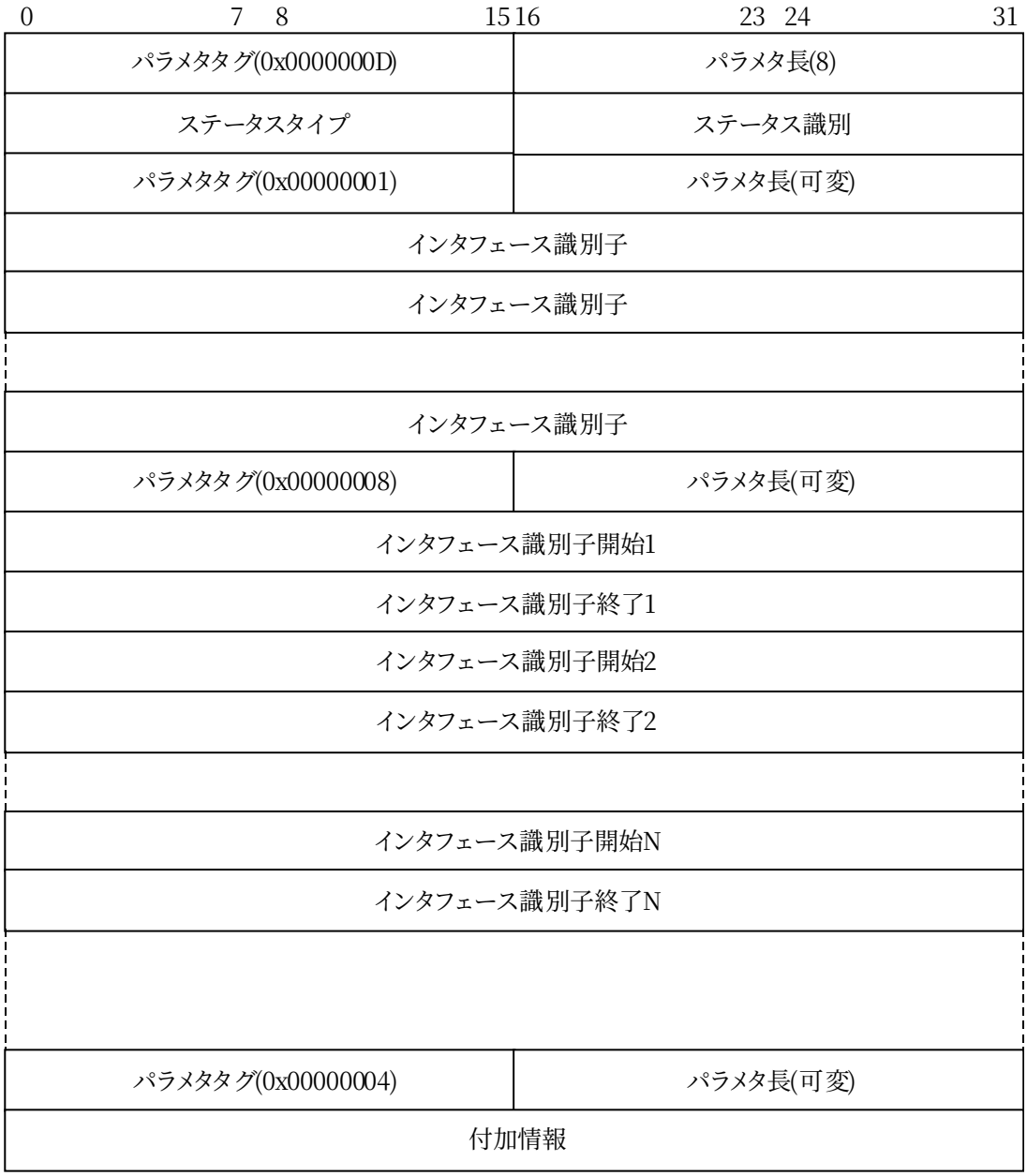


図 3-8 整数型インタフェース識別子使用時の NTFY **メッセージフォーマット**

続いて、文字列型インタフェース識別子使用時の NTFY **メッセージフォーマットを図 3-9 に示す。**

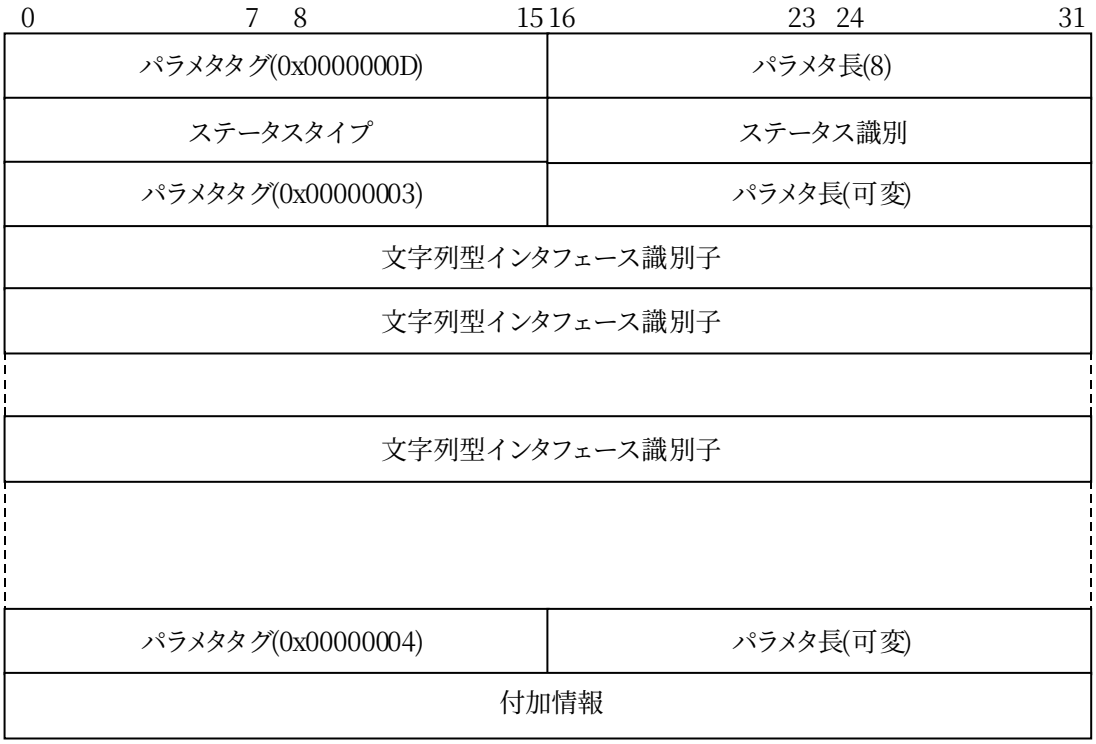


図 3-9 文字列型インタフェース識別子使用時の NTFY メッセージフォーマット

3.4.4.2.1.1 **ステータスタイプ／ステータス識別**

ステータスタイプは、NTFY メッセージの種別を示す 16 ビット符号無し整数であり、表 3-18 に示す値をとる。

表 3-18 ステータスタイプ一覧

値	**説明**
0x1	AS **状態変更**
0x2	**その他**

ステータス種別は 16 ビット符号無し整数であり、ステータスタイプ毎に値を定義する。ステータスタイプが「AS 状態変更」の場合に取り得る値を表 3-19 に示す。

表 3-19 AS 状態変更のステータス識別

値	**説明**
1	AS **停止状態**
2	AS **起動状態**

値	説明
3	AS 稼動状態
4	AS 保留状態

上記通知は、AS 状態変更時に SG から ASP に送られる。
ステータスタイプが「その他」の場合、ステータス識別は表 3-20 に示す値を取る。

表 3-20 その他のステータス識別

値	記述
1	ASP リソース不足
2	代替 ASP 稼動

上記通知は、ASP または AS の状態変更に起因しない。ASP リソース不足通知は、ロードシェアモードにおいて、負荷を処理するために ASP 追加が必要であることを不活性 ASP に通知する。代替 ASP 活性化通知は、オーバーライドモードにおいて、代替 ASP が活性状態に遷移し、負荷を引き継いだことを通知する。

3.4.4.2.1.2 インタフェース識別子

インタフェース識別子は AS が担当する D チャネル群を示すために使用する。

3.4.4.2.1.3 付加情報

付加情報は 8 ビット ASCII 文字列であり、0 文字から 255 文字までの文字を設定可能である。付加情報の設定内容は規定しない。

3.4.4.2.2 M3UA の NTFY メッセージ

5.3.4.2 参照

3.4.4.2.3 M2UA の NTFY メッセージ

6.4.4.2 参照

3.4.4.2.4 SUA の NTFY メッセージ

8.4.6.2 参照

3.4.4.3 TEI 状態要求メッセージ

TEI 状態要求メッセージは、TEI 状態を問い合わせるために使用する。本メッセージは IUA のみが使用する。伝統的な ISDN では同一交換機内部で Q.921 レイヤ管理と Q.931 レイヤ管理を実装するが、IUA では SG が Q.921 管理を実装し ASP が Q.931 管理を実装するため、TEI 状態問い合わせのためのメッセージが必要である。TEI 状態要求メッセージはパラメタを持たない。

3.4.4.4 TEI 状態確認メッセージ

TEI 状態確認メッセージは、TEI 状態要求メッセージによる TEI 状態問い合わせに応答するために使用する。TEI 状態確認メッセージは以下のパラメタを持つ:

- TEI 状態 (必須)

TEI 状態パラメタのフォーマットを図 3-10 に示す。

図 3-10 TEI 状態パラメタのフォーマット

TEI 状態パラメタは 32 ビット符号無し整数であり、表 3-21 に示す値を取る。

表 3-21 TEI 状態

値	説明
0x00	TEI 割当済
0x01	TEI 割当未

3.4.4.5 TEI 状態指示メッセージ

TEI 状態指示メッセージは、TEI 状態遷移を通知するために使用する。TEI 状態指示メッセージは以下のパラメタを持つ:

- TEI 状態 (必須)

3.4.5 MTP3 メッセージクラス

5.3.5 参照

3.4.6 ASP 状態管理メッセージクラス

ASP 状態管理メッセージクラスに所属するメッセージは、0 番の SCTP ストリームを使用して転送することを推奨する。

3.4.6.1 ASPUP メッセージ (ASP UP)

ASPUP メッセージは、ASP と関連付けられている全てのルーティングキーに関し、共通線信号網管理メッセージおよび ASP 管理メッセージを受信可能であることを同位 UA に通知するために用いる。

3.4.6.1.1 **パラメタとフォーマット**

ASPUP **メッセージは以下のパラメタを含む**:

- ASP **識別子**
- **付加情報(省略可能)**

3.4.6.1.2 ASP **識別子**

AS **をサポートする** ASP **を識別するユニークな値（省略可能）**

3.4.6.1.3 **付加情報**

3.4.4.2.1.3 **参照**。

3.4.6.2 ASPDN **メッセージ** (ASP Down)

ASPDN **メッセージは、メッセージ受信準備が出来ていないことを同位** UA **に通知するために使用する**。

3.4.6.2.1 **パラメタとフォーマット**

- **付加情報(省略可能)**

ASPDN **メッセージのパラメタ部フォーマットを図** 3-11 **に示す**。

図 3-11 ASPDN メッセージのフォーマット

3.4.6.2.2 **付加情報**

3.4.4.2.1.3 **参照**。

3.4.6.3 BEAT **メッセージ** (HeartBEAT)

BEAT **メッセージは、**SCTP **以外のトランスポートプロトコルを使用する場合に、**UA **間の正常性確認に使用する**。

3.4.6.3.1 **パラメタとフォーマット**

BEAT **メッセージは以下のパラメタを含む**:

- **ハートビート情報(省略可能)**

BEAT メッセージのパラメタ部フォーマットを図 3-12 に示す。

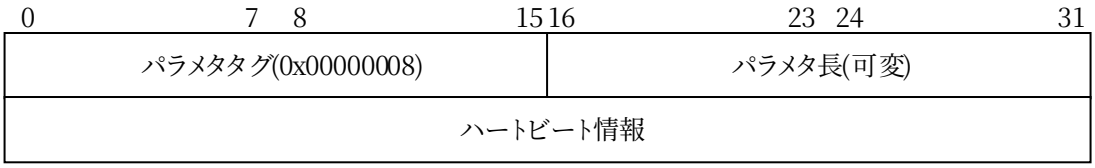


図 3-12 BEAT メッセージのフォーマット

3.4.6.3.2 **ハートビート情報**

ハートビート情報の内容は送信側が規定する。例えば、シーケンス番号やタイムスタンプを設定することが出来る。ハートビート情報は送信側にとってのみ意味を持つため、受信側は無視する。

3.4.6.4 ASPUP ACK **メッセージ** (ASP UP ACKnowledgement)

ASPUP ACK **メッセージは、**ASPUP **メッセージに応答するために用いる。**

3.4.6.4.1 **パラメタとフォーマット**

ASPUP ACK **メッセージは以下のパラメタを含む**:

- **付加情報(省略可能)**

3.4.6.4.2 **付加情報**

3.4.4.2.1.3 **参照。**

3.4.6.5 ASPDN ACK **メッセージ** (ASP DowN ACKnowledgement)

ASPDN ACK **メッセージは、**ASPDN **メッセージに応答するために用いる。**

3.4.6.5.1 **パラメタとフォーマット**

ASPDN ACK **メッセージは以下のパラメタを含む**:

- **理由(必須)**
- **付加情報(省略可能)**

3.4.6.5.2 **理由**

3.4.6.2.2 **参照。**

3.4.6.5.3 **付加情報**

3.4.4.2.1.3 **参照。**

3.4.6.6 BEAT ACK **メッセージ** (HeartBEAT ACKnowledgement)

BEAT ACK **メッセージは、**BEAT **メッセージの受信を知らせるために使用する。**BEAT **メッセージにハートビート情報パラメタが設定されている場合、**BEAT ACK **メッセージにも同一内容を設定する。**

3.4.7 ASP **トラフィック管理メッセージクラス**

ASP **トラフィック管理メッセージクラスに所属するメッセージは、**0 **番の** SCTP **ストリームを使用して転送することを推奨する。**

3.4.7.1 ASPAC メッセージ (ASP ACtive)

ASPAC メッセージは、ASP が稼動状態に遷移することを通知する。ルーティングコンテキストパラメタを用いて AS を指定する場合、稼動状態への遷移は指定する AS 内に限定される。ASPAC メッセージのフォーマットは UA 毎に異なる。

3.4.7.1.1 IUA の ASPAC メッセージ

IUA の ASPAC メッセージは以下のパラメタを含む:

- トラフィックモードタイプ(必須)
- インタフェース識別子(省略可能)
- 付加情報(省略可能)

3.4.7.1.1.1 トラフィックモードタイプ

トラフィックモードは、AS における ASP の動作を指定する 32 ビット符号無し整数である。取り得る値を表 3-22 に示す

表 3-22 トラフィックモード一覧

値	説明
0x01	オーバーライド
0x02	ロードシェア

ルーティングコンテキスト内では、オーバーライドとロードシェアを混在できない。オーバーライドは、ASP が AS の全トラフィックを処理することを示し、他に稼動状態 ASP があればトラフィックを引き継ぐ。ロードシェアは、ASP が他の稼動状態 ASP と共に AS のトラフィックを分担することを示す。

3.4.7.1.1.2 インタフェース識別子

3.4.4.2.1.2 参照。

3.4.7.1.1.3 付加情報

3.4.4.2.1.3 参照。

3.4.7.1.2 M3UA の ASPAC メッセージ

5.3.8.1 参照

3.4.7.1.3 M2UA の ASPAC メッセージ

6.4.6.1 参照

3.4.7.1.4 SUA の ASPAC メッセージ

8.4.5.1 参照

3.4.7.2 ASPAC ACK メッセージ (ASP ACtive ACKnowledgement)

ASPAC ACK メッセージは ASPAC メッセージの受信確認に使用する。ASPAC ACK メッセージのフォーマットは UA 毎に定義する。ASPAC(オーバーライド__待機)または ASPAC(ロードシェア__待機)の場合、ASP 使用開始時点で二回目の ASPAC ACK メッセージを送信する。

3.4.7.2.1 IUA の ASPAC ACK メッセージ

IUA の ASPAC ACK メッセージは以下のパラメタを含む:

- トラフィックモードタイプ(必須)
- インタフェース識別子(省略可能)
- 付加情報(省略可能)

3.4.7.2.1.1 トラフィックモード

3.4.7.1.1.1 参照。

3.4.7.2.1.2 インタフェース識別子

3.4.4.2.1.2 参照。

3.4.7.2.1.3 付加情報

3.4.4.2.1.3 参照。

3.4.7.2.2 M3UA の ASPAC ACK メッセージ

M3UA の ASPAC ACK メッセージは以下のパラメタを含む:

- トラフィックモードタイプ(必須)
- ルーティングコンテキスト(省略可能)
- 付加情報(省略可能)

3.4.7.2.2.1 トラフィックモード

3.4.7.1.1.1 参照。

3.4.7.2.2.2 ルーティングコンテキスト

3.4.4.2.2.2 参照。

3.4.7.2.2.3 付加情報

3.4.4.2.1.3 参照。

3.4.7.3 ASPIA メッセージ (ASP InActive)

ASPAC メッセージは、ASP が稼動状態に遷移することを通知する。ルーティングコンテキストパラメタを用いて AS を指定する場合、稼動状態への遷移は指定する AS 内に限定される。ASPAC メッセージのフォーマットは UA 毎に異なる。

3.4.7.3.1 IUA の ASPIA メッセージ

IUA の ASPIA メッセージは以下のパラメタを含む:

- トラフィックモードタイプ(必須)

- **インタフェース識別子(省略可能)**
- **付加情報(省略可能)**

3.4.7.3.1.1 **トラフィックモード**

3.4.7.1.1.1 **参照**。

3.4.7.3.1.2 **インタフェース識別子**

3.4.4.2.1.2 **参照**。

3.4.7.3.1.3 **付加情報**

3.4.4.2.1.3 **参照**。

3.4.7.3.2 M3UA **の** ASPIA **メッセージ**

5.3.8.2 **参照**

3.4.7.3.3 M2UA **の** ASPIA **メッセージ**

6.4.6.2 **参照**

3.4.7.3.4 SUA **の** ASPIA **メッセージ**

8.4.5.3 **参照**

3.4.7.4 ASPIA ACK (ASP InActive ACKnowledgement) **メッセージ**

ASPIA ACK **メッセージは** ASPIA **メッセージの受信確認に使用する。**ASPIA ACK **メッセージのフォーマットは** UA **毎に定義する。**

3.4.7.4.1 IUA **の** ASPIA ACK **メッセージ**

IUA **の** ASPIA ACK **メッセージは以下のパラメタを含む**:

- **トラフィックモードタイプ(必須)**
- **インタフェース識別子(省略可能)**
- **付加情報(省略可能)**

3.4.7.4.1.1 **トラフィックモード**

3.4.7.1.1.1 **参照**。

3.4.7.4.1.2 **インタフェース識別子**

3.4.4.2.1.2 **参照**。

3.4.7.4.1.3 **付加情報**

3.4.4.2.1.3 **参照**。

3.4.7.4.2 M3UA **の** ASPIA ACK **メッセージ**

- M3UA **の** ASPIA ACK **メッセージは以下のパラメタを含む**:
- **ルーティングコンテキスト(省略可能)**

- **付加情報(省略可能)**

3.4.7.4.2.1 **ルーティングコンテキスト**

3.4.2.2.2 **参照。**

3.4.7.4.2.2 **付加情報**

3.4.4.2.1.3 **参照。**

3.4.8 メッセージ実装規定

アダプテーションプロトコルが実装するメッセージを表 3-23 **に示す。「**M**」は実装が必須であることを示し、「-」は実装しないことを示す。「**O**」は実装しなくても良いことを示す。**

表 3-23 メッセージ実装規定

メッセージ名	M3UA	M2UA	M2PA	IUA	SUA
管理メッセージクラス					
ERR **メッセージ**	M	M	M	M	M
NTFY **メッセージ**	M	M	-	M	M
TEI **状態要求メッセージ**	-	-	-	M	-
TEI **状態確認メッセージ**	-	-	-	M	-
TEI **状態指示メッセージ**	-	-	-	M	-
MTP3 **メッセージクラス**					
ペイロードデータメッセージ	M	-	-	-	-
共通線信号網管理メッセージクラス					
DUNA **メッセージ**	M	-	-	-	M
DAVA **メッセージ**	M	-	-	-	M
DAUD **メッセージ**	M	-	-	-	M
SCON **メッセージ**	M	-	-	-	M
DUPU **メッセージ**	M	-	-	-	M
DRST **メッセージ**	O	-	-	-	O
ASP **状態管理メッセージクラス**					
ASPUP **メッセージ**	M	M	-	M	M

メッセージ名	M3UA	M2UA	M2PA	IUA	SUA
ASPDN メッセージ	M	M	-	M	M
BEAT メッセージ	O	O	-	O	O
ASPUP ACK メッセージ	M	M	-	M	M
ASPDN ACK メッセージ	M	M	-	M	M
BEAT ACK メッセージ	O	O	-	O	O
ASP トラフィック管理メッセージ					
ASPAC メッセージ	M	M	-	M	M
ASPIA メッセージ	M	M	-	M	M
ASPAC ACK メッセージ	M	M	-	M	M
ASPIA ACK メッセージ	M	M	-	M	M
Q921 メッセージクラス					
データ要求メッセージ	-	-	-	M	-
データ指示メッセージ	-	-	-	M	-
ユニットデータ要求メッセージ	-	-	-	M	-
ユニットデータ指示メッセージ	-	-	-	M	-
設定要求メッセージ	-	-	-	M	-
設定確認メッセージ	-	-	-	M	-
設定指示メッセージ	-	-	-	M	-
解放要求メッセージ	-	-	-	M	-
解放確認メッセージ	-	-	-	M	-
解放指示メッセージ	-	-	-	M	-
MTP2 メッセージクラス					
データメッセージ	-	O	-	-	-
リンク設定要求メッセージ	-	M	-	-	-
リンク設定確認メッセージ	-	M	-	-	-
リンク解放要求メッセージ	-	M	-	-	-

メッセージ名	M3UA	M2UA	M2PA	IUA	SUA
リンク解放確認メッセージ	-	M	-	-	-
リンク解放指示メッセージ	-	M	-	-	-
リンク状態要求メッセージ	-	M	-	-	-
リンク状態確認メッセージ	-	M	-	-	-
リンク状態指示メッセージ	-	M	-	-	-
回収要求メッセージ	-	M	-	-	-
回収確認メッセージ	-	M	-	-	-
回収指示メッセージ	-	M	-	-	-
回収完了指示メッセージ	-	M	-	-	-
輻輳通知メッセージ	-	M	-	-	-
TTC データメッセージ	-	M	-	-	-
M2PA メッセージクラス					
データメッセージ		O			
リンク状態		M			
Proving Data		M			
SCCP コネクションレスメッセージクラス					
CLDT メッセージ	-	-	-	-	M
CLDR メッセージ	-	-	-	-	M
SCCP コネクション指向メッセージクラス					
CORE メッセージ	-	-	-	-	-
COAK メッセージ	-	-	-	-	-
COREF メッセージ	-	-	-	-	-
RELRE メッセージ	-	-	-	-	-
RELCO メッセージ	-	-	-	-	-
RESCO メッセージ	-	-	-	-	-
RESRE メッセージ	-	-	-	-	-

メッセージ名	M3UA	M2UA	M2PA	IUA	SUA
CODT メッセージ	-	-	-	-	-
CODA メッセージ	-	-	-	-	-
COERR メッセージ	-	-	-	-	-
COIT メッセージ	-	-	-	-	-
ルーティングキー管理メッセージクラス					
登録要求メッセージ	O	-	-	-	O-
登録応答メッセージ	O	-	-	-	O
解除要求メッセージ	O	-	-	-	O
解除応答メッセージ	O	-	-	-	O
インタフェース識別子管理メッセージ					
登録要求メッセージ		O			
登録応答メッセージ		O			
解除要求メッセージ		O			
解除応答メッセージ		O			

3.5 手順

3.5.1 SCTP 管理サービス手順

3.5.1.1 アソシエーション確立手順

アソシエーションは以下の手順で確立する。アソシエーション確立を要求する側を「起動側」、アソシエーション確立を受け入れる側を「受諾側」と記す。

起動側 UA は M-SCTP 設定要求プリミティブ受信を契機として、起動側 SCTP に対して Associate プリミティブを発行する。

起動側 SCTP は Associate プリミティブ受信を契機として、受諾側 SCTP へ INIT チャンクを送信する。

受諾側 SCTP は INIT チャンクを受信し、INIT ACK チャンクを返送する。

起動側 SCTP は INIT ACK チャンクを受信し、COOKIE ECHO チャンクを送信する。

受諾側 SCTP は COOKIE ECHO チャンクを受信し、COOKIE ACK チャンクを返送すると同時に、受諾側 UA に Communication UP プリミティブを発行する。

起動側 SCTP は COOKIE ACK チャンクを受信し、起動側 UA に Communication UP プリミティブを発行する。

上記手順の通信シーケンスを図 3-13 に示す。

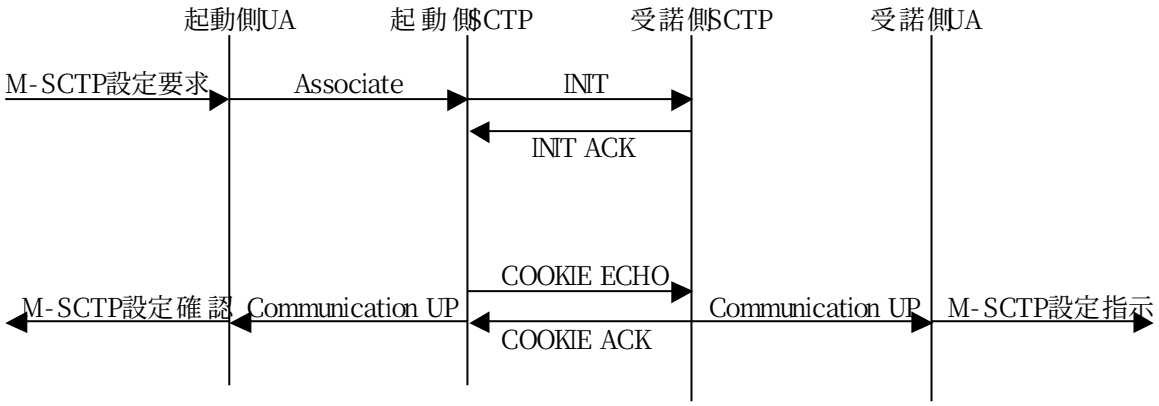


図 3-13 アソシエーション確立シーケンス

3.5.1.2 **アソシエーション解放手順**

アソシエーションは以下の手順で解放する。アソシエーション解放を要求する側を「起動側」、アソシエーション解放を受け入れる側を「受諾側」と記す。

1. **起動側** UA **は** M-SCTP **解放要求プリミティブ受信を契機として、起動側** SCTP **に対して** Shutdown **プリミティブを発行する。**

起動側 SCTP **は送信済み** DATA **チャンクの送達確認を待って、**SHUTDOWN **チャンクを受諾側** SCTP **に送信する。**

受諾側 SCTP **は** SHUTDOWN **チャンクを受信し、送信済み** DATA **チャンクの送達確認を待って、**SHUTDOWN ACK **チャンクを返送する。**

起動側 SCTP **は** SHUTDOWN ACK **チャンクを受信し、**SHUTDOWN COMPLETE **チャンクを返送すると同時に、起動側に対して** Shutdown Complete **プリミティブを発行する。**

受諾側 SCTP **は** SHUTDOWN COMPELTE **チャンクを受信し、受諾側** UA **に** Shutdown Complete **プリミティブを発行する。**

上記手順の通信シーケンスを図 3-14 **に示す。**

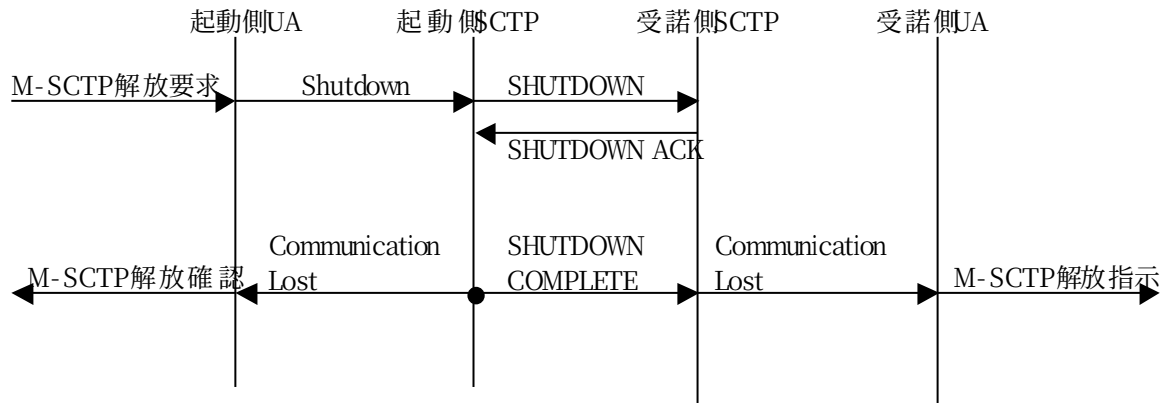


図 3-14 アソシエーション解放シーケンス

3.5.2 **メッセージ転送手順**

メッセージは以下の手順で転送する。

1. **送信元** UA **は、**Send **プリミティブを発行して、送信元** SCTP **に対してメッセージ転送を依頼する。**

送信元 SCTP **は** Data **チャンクを送信先** SCTP **に送信する。**

送信先 SCTP **は** Data **チャンクを受信すると、送信先** UA **に対して** Data Arrive **プリミティブを発行してメ**

ッセージ到着を通知する。

送信先 UA は Receive プリミティブを発行して、メッセージを受信する。

上記手順の通信シーケンスを図 3-15 に示す。

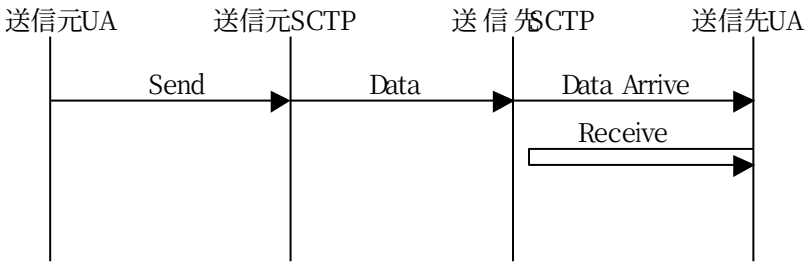


図 3-15 メッセージ転送手順

3.5.3 UA 管理サービス手順

3.5.3.1 エラー通知手順

エラーは以下の手順で通知する。エラーを検出する UA を「検出元 UA」、エラーとなるメッセージの送信元 UA を「発生元 UA」と表記する。

1. 検出元 UA は発生元 UA からメッセージを受信し、エラーであると判定し、ERR メッセージを発生元 UA に送信する。

発生元 UA は ERR メッセージを受信し、上位レイヤに対して M-ERROR 指示プリミティブを発行する。

上記手順の通信シーケンスを図 3-16 に示す。

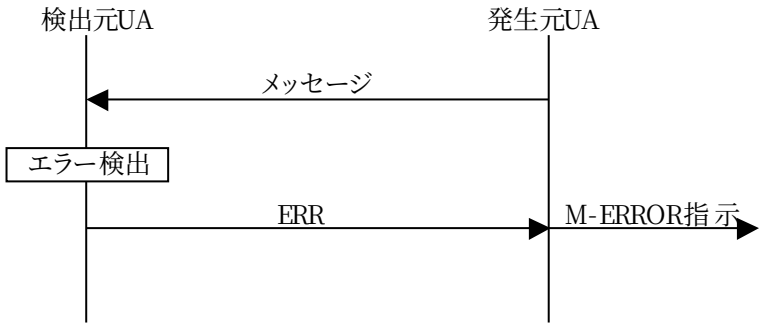


図 3-16 エラー通知シーケンス(1)

エラーはローカルに検出し通知することも可能である(図 3-17 参照)。

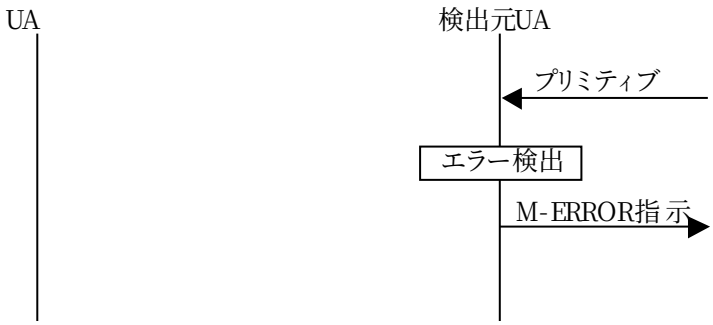


図 3-17 エラー通知シーケンス(2)

3.5.3.2 **イベント通知手順**

イベントは以下の手順で通知する。

1. **送信元** UA **はイベントを検出し、**NTFY **メッセージを送信先** UA **に送信する。**

送信先 UA **は** NTFY **メッセージを受信し、上位レイヤに対して** M-NOTIFY **指示プリミティブを発行する。**

上記手順の通信シーケンスを図 3-18 **に示す。**

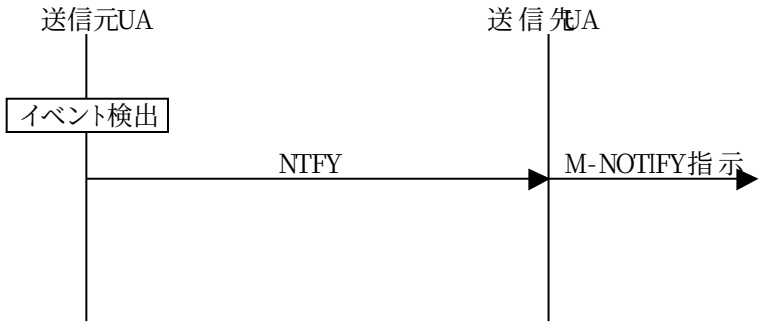


図 3-18 イベント通知シーケンス(1)

イベントはローカルに発生し通知することも可能である(図 3-19 **参照)。**

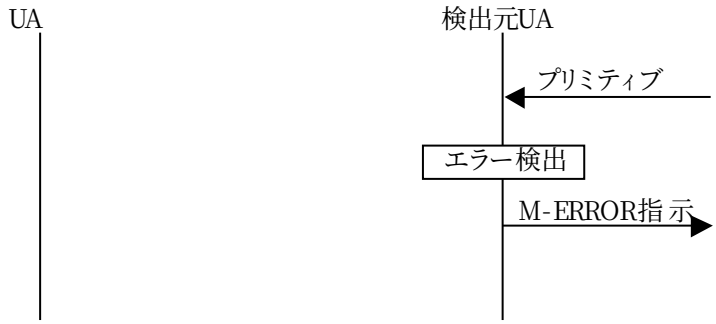


図 3-19 イベント通知シーケンス(2)

3.5.3.3 TEI **状態問い合わせ手順**

TEI **状態は以下の手順で問い合わせる。**

1. IPSP **において、**ASP **は** UA **に対して** M-TEI **状態要求プリミティブを発行する。**

UA **は** SG **上の** UA **に** TEI **状態要求メッセージを送信する。**

SG **上の** UA **は** TEI **状態確認メッセージに** TEI **状態を設定して返送する。**

IPSP **上の** UA **は** TEI **状態確認メッセージを受信し、**ASP **に対して** M-TEI **状態確認プリミティブを発行する。**

上記手順の通信シーケンスを図 3-20 **に示す。**

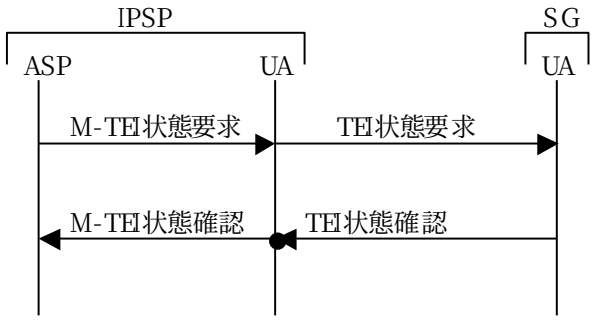


図 3-20 TEI 状態問い合わせシーケンス

3.5.4 ASP 管理手順

ASP が取り得る状態を以下に示す:

- **停止状態** (ASP-DOWN)
- **起動状態** (ASP-INACTIVE)
- **稼動状態** (ASP-ACTIVE)

停止状態は ASP を利用できない状態である。SG と ASP の間に SCTP アソシエーションは確立していても、確立していなくてもよい。SG は停止状態の ASP に対してメッセージを送信しないことを推奨する。

起動状態は SG と ASP との間の SCTP アソシエーションは確立し、管理メッセージは処理できるが信号トラフィックは処理できない状態である。処理できる管理メッセージは、TEI 状態要求／指示／確認メッセージを除く UA 管理メッセージ、ASP 状態管理メッセージ、ASP トラフィック管理メッセージ、共通線信号網管理メッセージである。

稼動状態は SG と ASP との間に SCTP アソシエーションが確立し、管理メッセージと信号トラフィックを処理できる状態である。

以降に状態遷移を規定する。初期状態は停止状態である。

停止状態において ASP から ASPUP メッセージを受信すると起動状態に遷移する。

起動状態において ASP から ASPAC メッセージを受信すると稼動状態に遷移し、ASPDN メッセージを受信すると停止状態に遷移する。下位の SCTP から Communication Lost プリミティブまたは Shutdown Complete プリミティブを受信しても停止状態に遷移する。

稼動状態において、ASP から APSIA メッセージを受信すると起動状態に遷移し、ASPDN メッセージを受信すると停止状態に遷移する。下位の SCTP から Communication Lost プリミティブまたは Shutdown Complete プリミティブを受信しても停止状態に遷移する。次節に述べる代替 ASP による引継ぎが発生すると起動状態に遷移する。

状態遷移図を図 3-21 に示す。

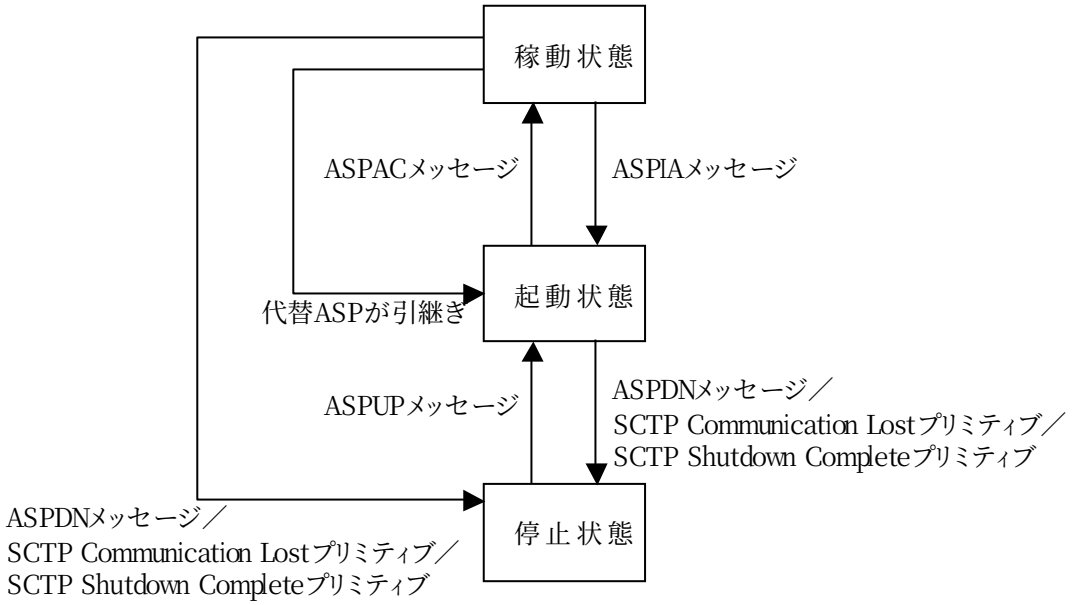


図 3-21 ASP 状態遷移図

3.5.5 AS 管理手順

AS が取り得る状態を以下に示す:

- **停止状態** (AS-DOWN)
- **起動状態** (AS-INACTIVE)
- **稼動状態** (AS-ACTIVE)
- **保留状態** (AS-PENDING)

停止状態は AS **中の全** ASP **が停止状態にある状態である。**

起動状態は AS **中に起動状態の** ASP **が存在するが、稼動状態の** ASP **が存在しない状態である。**

稼動状態は AS **中に稼動状態の** ASP **が存在する状態である。**

保留状態は AS **中の最後の稼動状態** ASP **が起動状態または停止状態に遷移した直後に占める過渡的な状態である。**SG **は保留状態** AS **に対する信号トラフィックをキューに保存する。**AS **が保留状態に遷移すると** SG **は回復タイマ** Tr **を起動する。回復タイマ** Tr **が満了すると** AS **は起動状態または停止状態に遷移し、キュー上の信号トラフィックを破棄する。回復タイマ** Tr **満了前に** ASP **が稼動状態に遷移すると、**AS **は稼動状態に遷移し、キュー上の信号トラフィックを** ASP **に送信する。**

以降に AS **の状態遷移規定を示す。初期状態は停止状態である。**

停止状態において最低でも一つの ASP **が起動状態に遷移すると、**AS **は起動状態に遷移する。**

起動状態において最低でも一つの ASP **が稼動状態に遷移すると、**AS **は稼動状態に遷移する。全ての** ASP **が停止状態に遷移すると、**AS **は停止状態に遷移する。**

稼動状態において最後の稼動状態 ASP **が起動状態または停止状態に遷移すると、**AS **は保留状態に遷移し、回復タイマ** Tr **を起動する。**

保留状態において回復タイマ Tr **満了時に起動状態** ASP **が存在すれば** AS **は起動状態に遷移し、存在しなければ停止状態に遷移する。回復タイマ満了前に最低でも一つの** ASP **が稼動状態に遷移すると** AS **は稼動状態に遷移する。**

状態遷移図を図 3-22 **に示す。**

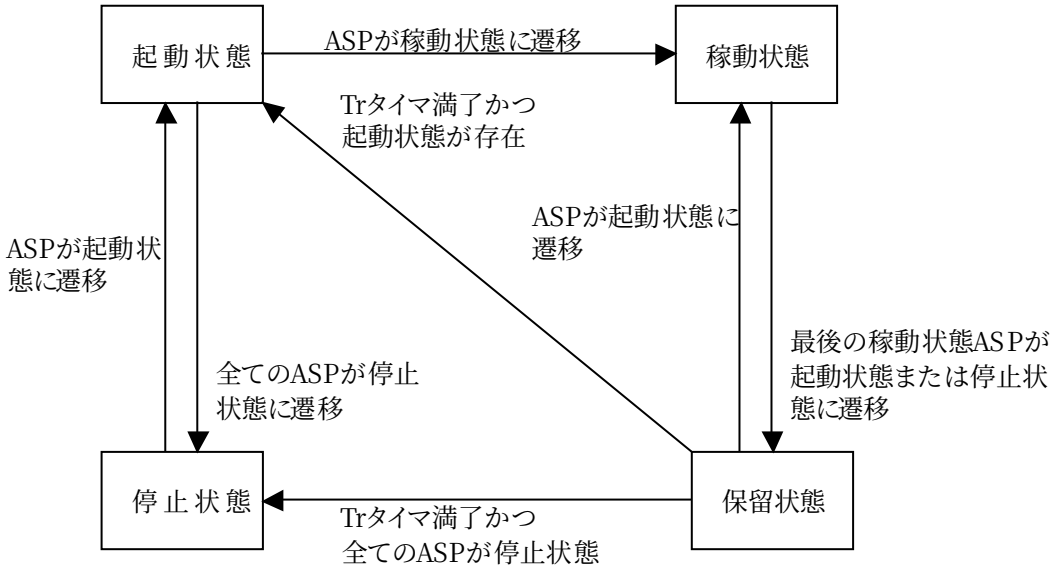


図 3-22 AS 状態遷移

3.6 通信シーケンス例

3.6.1 初期化シーケンス

3.6.1.1 初期化シーケンス 1

単一の ASP により AS を構成する場合の初期化シーケンスを図 3-23 に示す。SCTP アソシエーションは確立済みとする。初期化完了後、ASP は稼動状態になる。

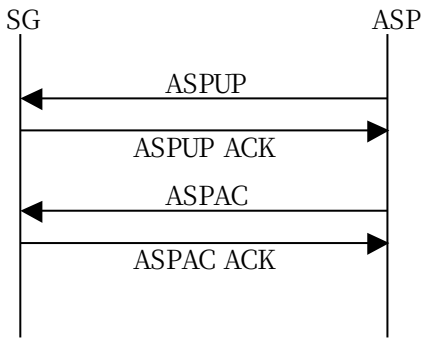


図 3-23 初期化シーケンス 1

3.6.1.2 初期化シーケンス 2

2 台の ASP による冗長構成の初期化シーケンスを示す。トラフィックモードはオーバーライドモードとする。初期化完了時、ASP1 は稼動状態、ASP2 は起動状態となる。

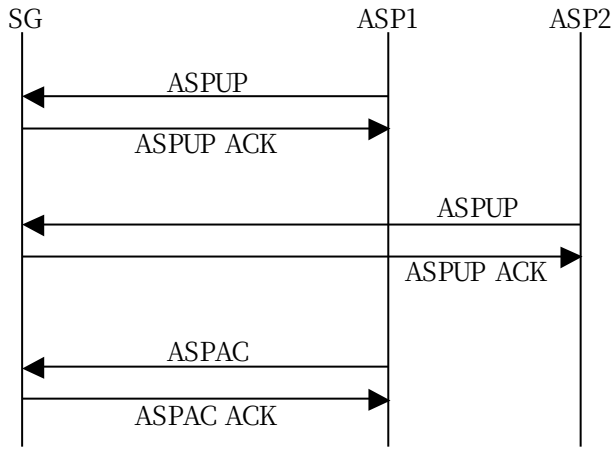


図 3-24 初期化シーケンス 2

3.6.1.3 **初期化シーケンス** 3

2 **台の** ASP **による冗長構成の初期化シーケンスを示す。トラフィックモードはロードシェアリングモードとする。初期化完了後、**ASP1 **と** ASP2 **ともに稼動状態となる。**

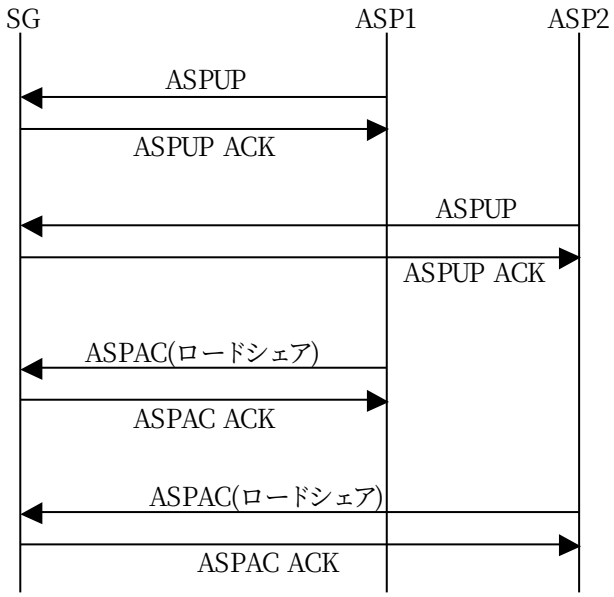


図 3-25 初期化シーケンス 3

3.6.1.4 **初期化シーケンス** 4

2 **台の** ASP **による冗長構成の初期化シーケンスを示す。トラフィックモードはロードシェアリングモードとする。初期化完了後、**ASP1 **と** ASP2 **は稼動状態、**ASP3 **は起動状態となる。**

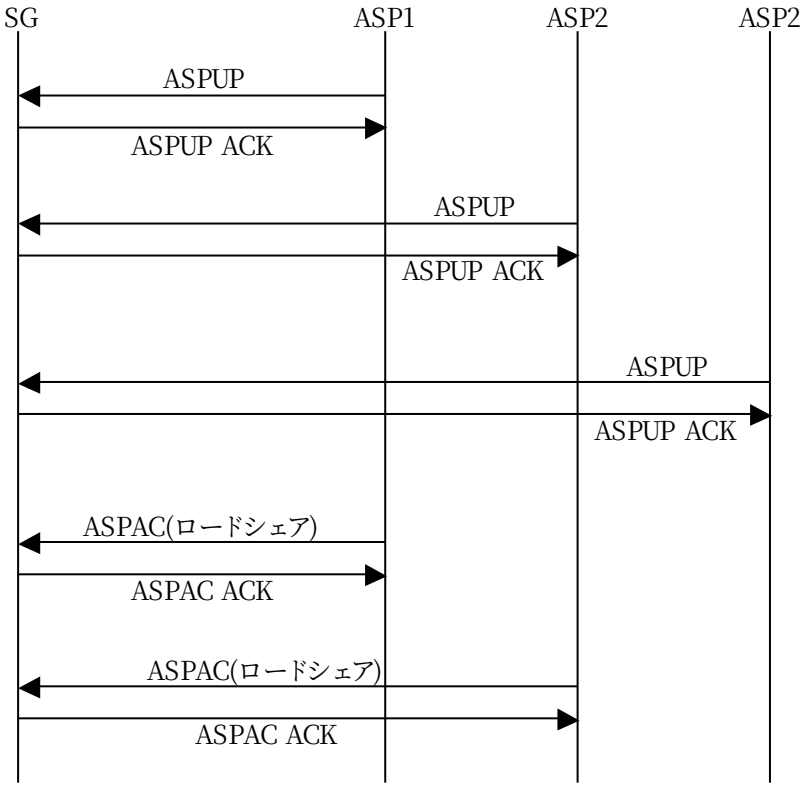


図 3-26 初期化シーケンス 4

3.6.2 フェイルオーバシーケンス

3.6.2.1 フェイルオーバシーケンス 1

ASP1 が稼動状態から起動状態に遷移し、ASP2 が稼動状態に遷移して信号トラフィックを引き継ぐシーケンスを図 3-27 に示す。

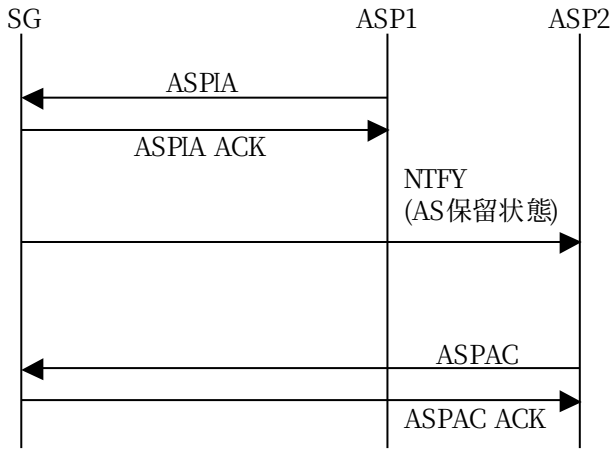


図 3-27 フェイルオーバシーケンス 1

3.6.2.2 フェイルオーバシーケンス 2

ASP2 が ASP1 の信号トラフィックを引き継ぐシーケンスをに示す。SG は ASP1 に NTFY メッセージを送

り、起動状態への遷移を通知する。

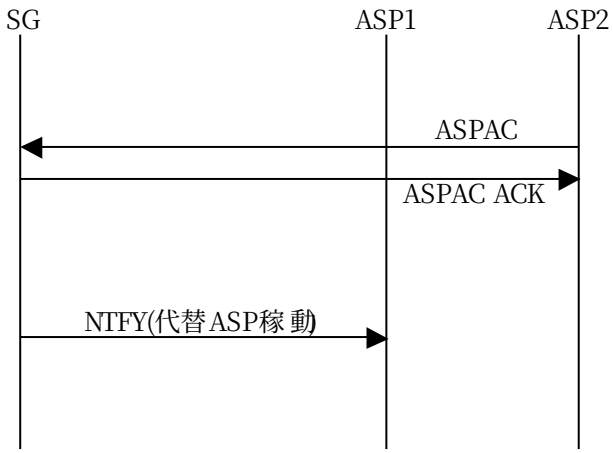


図 3-28 フェイルオーバシーケンス 2

3.6.2.3 フェイルオーバシーケンス 3

3.6.1.4 の続きであり、ASP1 が稼動状態から起動状態に遷移することにより ASP リソースが不足するため、ASP3 が起動状態から稼動状態に遷移するシーケンスである(図 3-29)。

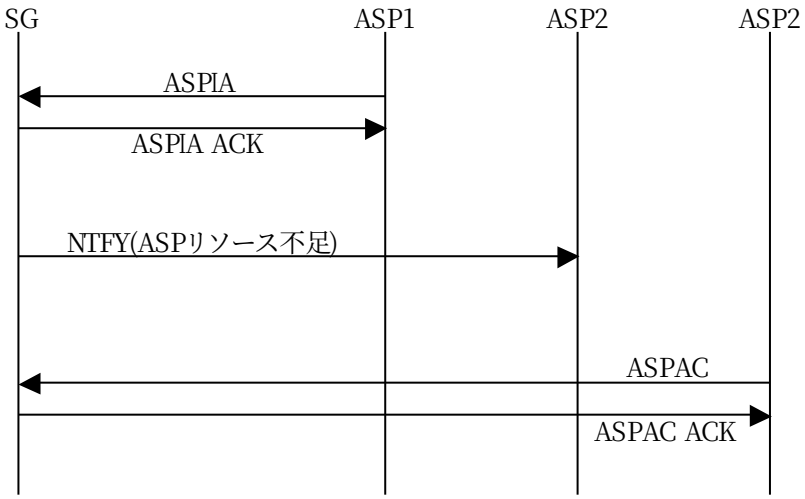


図 3-29 フェイルオーバシーケンス 3

3.7 セキュリティ

ネットワークの安全性を保証できない場合は、IPSEC の使用を推奨する。

3.8 登録番号

3.8.1 ペイロードプロトコル識別子

SCTP の DATA チャンクにペイロードプロトコル識別子を設定し、UA を識別する。

3.8.2 ポート番号

UA の”Well-Known”ポート番号を規定する。

3.9 将来の拡張性

UA の下記項目は拡張可能である。

- メッセージクラス
- メッセージタイプ
- パラメタ

3.9.1 メッセージクラスの拡張

メッセージクラス拡張時は以下を規定する。

- メッセージクラス名と略称
- メッセージクラスの使用目的

3.9.2 メッセージタイプの拡張

メッセージタイプ拡張時は以下を規定する。

- メッセージタイプ名と略称
- メッセージフォーマット
- フィールド定義
- メッセージの使用手順
- メッセージ受信時のエラー

3.9.3 パラメタの拡張

パラメタ拡張時は以下を規定する。

- パラメタタイプ名
- 3.4.3 に準拠するパラメタフォーマット
- パラメタ値の説明
- パラメタの使用方法、特に、単一メッセージ内で反復可能な条件

4. IUA

4.1 序論

IUA(ISDN Q.921-User Adaptation)は、Q.931 などの Q.921 ユーザ部プロトコルを転送するアダプテーションプロトコルである。

IUA の下位プロトコルとして SCTP の使用を推奨する。更に、D チャネルトラフィックの相互干渉を極小化するため、D チャネル毎に別々の SCTP ストリームを使用することを薦める。

4.1.1 シームレスなネットワークマネージメントインタワーキング

アクティブな ASP が状態遷移するとき、SG における IUA レイヤはローカル側のレイヤ管理に IUA ユーザ(Q.931)無効の表示を伝えるべきである。望ましいと考えるのなら、レイヤ管理は Q.921 側に対しても同様の動作が可能である。

同様に SCTP アソシエーションが障害の時は、Dch をアウトオブサービスにするため SG と ASP の IUA レイヤは Release プリミティブを送出することがある。

4.1.2 輻輳制御

IUA レイヤが輻輳する場合、同位 IUA レイヤをフローコントロールするため SCTP アソシエーションからの読み込みをストップすることがある。

4.2 用語

4.2.1 インタフェース

本章において、インタフェースは ISDN D チャネルを意味する。

4.2.2 インタフェース識別子

インタフェースを識別する整数または文字列である。

4.2.3 Q.921 ユーザ

Q.921 サービスを使用するプロトコル。

4.2.4 バックホール

SG が Q.921 を終端し、上位の Q.931 を MGC に中継することである。

4.3 サービス

IUA は以下のサービスを使用する。

- SCTP 管理サービス
- UA 管理サービス
- ASP 管理サービス
- AS 管理サービス
- Q.921 サービス

4.3.1 SCTP 管理サービス

3.3.1 参照。

4.3.2 UA 管理サービス

3.3.2 参照。

4.3.3 ASP 管理サービス

3.3.3 参照。

4.3.4 AS 管理サービス

3.3.4 参照。

4.3.5 Q.921 サービス

Q.921 サービスのプリミティブ一覧を表 4-1 に示す。

表 4-1 Q.921 サービスプリミティブ一覧

サービスプリミティブ		概要
DL-設定	要求 指示 確認	マルチフレーム動作の設定および結果確認に使用する。
DL-解放	要求 指示 確認	マルチフレーム動作の解放、結果確認および設定失敗通知に使用する。
DL-データ	要求 指示	確認形情報転送サービスを用いた Q.921 ユーザ部プロトコルデータの送信要求および受信通知に使用する。
DL-ユニットデータ	要求 指示	非確認形情報転送サービスを用いた Q.921 ユーザ部プロトコルデータの送信要求および受信通知に使用する。

4.4 メッセージ

IUA は以下のメッセージを使用する。

- UA 管理メッセージクラス
- ASP 状態管理メッセージクラス
- ASP トラフィック状態管理メッセージクラス
- Q.921 メッセージクラス

4.4.1 共通メッセージヘッダ

3.4.1 参照。

4.4.2 個別メッセージヘッダ

IUA 個別メッセージヘッダは以下のメッセージに使用する:

- TEI 状態要求メッセージ／TEI 状態確認メッセージ／TEI 状態指示メッセージ
- Q.921 メッセージクラスの全メッセージ

IUA 個別メッセージヘッダは以下のパラメタを持つ:

- インタフェース識別子
- データリンクコネクション識別子

4.4.2.1 インタフェース識別子

インタフェース識別子は D チャネルを識別する。32 ビット符号無し整数で表現する整数型インタフェース識別子(図 4-1)と可変長文字列で表現する文字列型インタフェース識別子(図 4-2)がある。整数型インタフェース識別子のサポートは必須だが、文字列型インタフェース識別子のサポートは任意である。

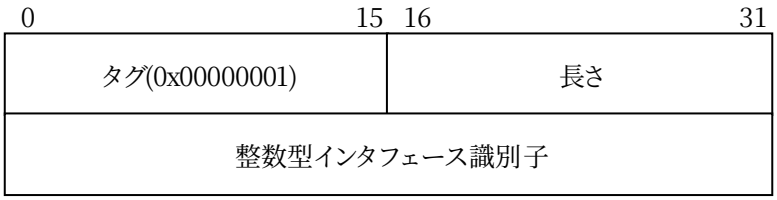


図 4-1 整数型インタフェース識別子のフォーマット

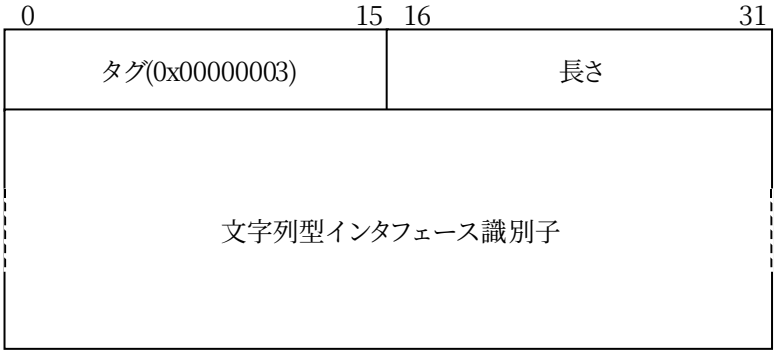


図 4-2 文字列型インタフェース識別子のフォーマット

4.4.2.2 **データリンクコネクション識別子**

データリンクコネクションを識別するパラメタである。フォーマットを図 4-3 **に示す。**

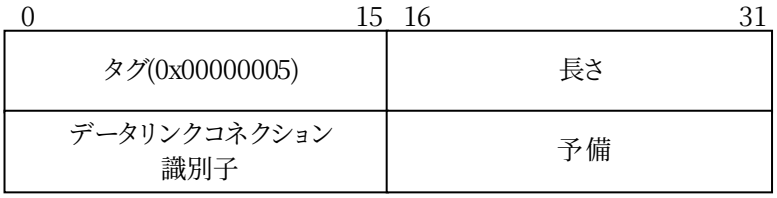


図 4-3 データリンクコネクション識別子のフォーマット

上位 16 **ビットのフォーマットを図** 4-4 **に示す。**

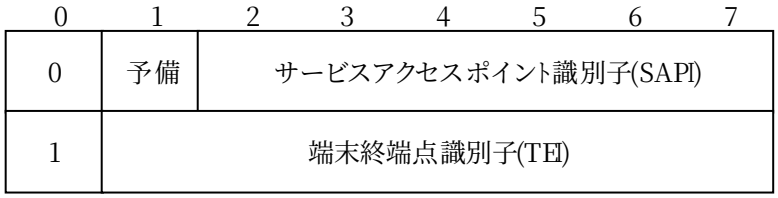


図 4-4 データリンクコネクション識別子

4.4.2.3 IUA **個別メッセージヘッダフォーマット**

IUA **個別メッセージヘッダ全体のフォーマットを図** 4-5 **と図** 4-6 **に示す。**

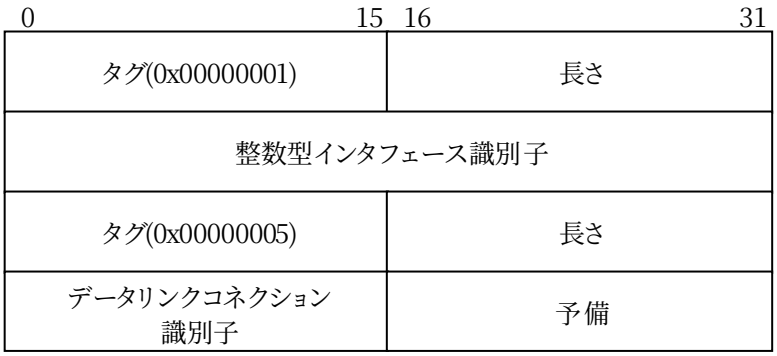


図 4-5 整数型インタフェース識別子使用時の IUA 個別メッセージヘッダ

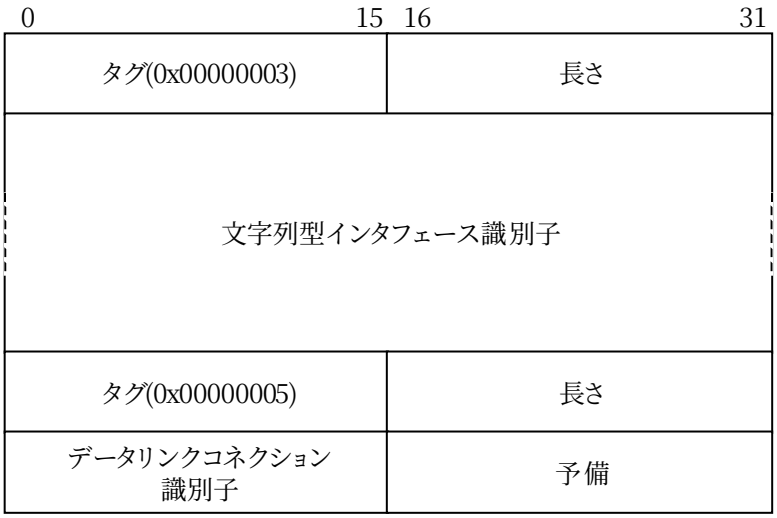


図 4-6 文字列型インタフェース識別子使用時の IUA 個別メッセージヘッダ

4.4.3 **パラメタ**

3.4.3 **参照。**

4.4.4 UA **管理メッセージ**

3.4.4 **参照。**

4.4.5 ASP **状態管理メッセージ**

3.4.6 **参照。**

4.4.6 ASP **トラフィック管理メッセージ**

3.4.7 **参照。**

4.4.7 Q.921 **メッセージ**

4.4.7.1 **データ要求メッセージ**

データ要求メッセージは、Q.921 **確認型情報転送サービス使用したデータ転送を** SG **に依頼するため、**ASP **から** SG **に送信する。**

データ要求メッセージは、共通メッセージヘッダ、IUA 個別メッセージヘッダおよび以下のパラメタを持つ：

- プロトコルデータ (必須)

データ要求メッセージのパラメタ部フォーマットを図 4-7 に示す。

図 4-7 データ要求メッセージのフォーマット

4.4.7.1.1 プロトコルデータ

プロトコルデータは、Q.931 等上位レイヤプロトコルの信号メッセージを設定する。

4.4.7.2 データ指示メッセージ

データ指示メッセージは、Q.921 確認型情報転送サービスを使用してデータを転送するため、SG から ASP に送信する。

データ指示メッセージは、共通メッセージヘッダ、IUA 個別メッセージヘッダおよび以下のパラメタを持つ：

- プロトコルデータ (必須)

データ指示メッセージのパラメタ部フォーマットはデータ要求メッセージと共通である。

4.4.7.2.1 プロトコルデータ

4.4.7.1.1 参照。

4.4.7.3 ユニットデータ要求メッセージ

ユニットデータ要求メッセージは、Q.921 非確認型情報転送サービス使用したデータ転送を SG に依頼するため、ASP から SG に送信する。

データ要求メッセージは、共通メッセージヘッダ、IUA 個別メッセージヘッダおよび以下のパラメタを持つ：

- プロトコルデータ (必須)

ユニットデータ要求メッセージのパラメタ部フォーマットはデータ要求メッセージと共通である。

4.4.7.3.1 プロトコルデータ

4.4.7.1.1 参照。

4.4.7.4 ユニットデータ指示メッセージ

ユニットデータ指示メッセージは、Q.921 非確認型情報転送サービス使用したデータを転送するため、SG から ASP に送信する。

ユニットデータ指示メッセージは、共通メッセージヘッダ、IUA 個別メッセージヘッダおよび以下のパラメタを持つ：

- **プロトコルデータ (必須)**

ユニットデータ指示メッセージのパラメタ部フォーマットはデータ要求メッセージと共通である。

4.4.7.4.1 **プロトコルデータ**

4.4.7.1.1 **参照。**

4.4.7.5 **設定要求メッセージ**

設定要求メッセージは、マルチフレーム動作の設定を要求するため、ASP **から** SG **に送信する。**

設定要求メッセージは、共通メッセージヘッダと IUA **個別メッセージヘッダから構成される。**

4.4.7.6 **設定確認メッセージ**

設定確認メッセージは、設定要求メッセージの応答として SG **から** ASP **に送信する。**

設定確認メッセージは、共通メッセージヘッダと IUA **個別メッセージヘッダから構成される。**

4.4.7.7 **設定指示メッセージ**

設定指示メッセージは、端末からの要求によりマルチフレーム動作が設定されたことを通知するため、SG **から** ASP **に送信される。**

設定確指示ッセージは、共通メッセージヘッダと IUA **個別メッセージヘッダから構成される。**

4.4.7.8 **解放要求メッセージ**

解放要求メッセージは、マルチフレーム動作の解放を要求するため、ASP **から** SG **に送信する。**

解放要求メッセージは、共通メッセージヘッダ、IUA **個別メッセージヘッダおよび以下のパラメタを持つ:**

- **解放理由 (必須)**

解放要求メッセージのパラメタ部フォーマットを図 4-8 **に示す。**

図 4-8 **解放要求メッセージのフォーマット**

4.4.7.8.1 **解放理由**

解放理由はマルチフレーム動作の解放理由を示す 32 **ビット符号無し整数であり表** 4-2 **に示す値をとる。ただし、解放要求メッセージでは「物理層警報に基づく解放」は使用しない。**

表 4-2 解放理由の取り得る値一覧

名称	値	説明	解放要求	解放指示
管理層による解放	0x00	管理層が解放を指示する。	○	○
物理層警報に基づく解放	0x01	物理層警報に基づいて解放する。	－	○
設定拒否	0x02	マルチフレーム動作を解放するとともに、端末からのマルチフレーム動作設定要求を拒否する。	○	－
その他	0x03	その他の理由。	○	○

○　　使用する

－　　使用しない

4.4.7.9 解放確認メッセージ

設定確認メッセージは、設定要求メッセージの応答として SG から ASP へ送信する。

設定確認メッセージは、共通メッセージヘッダと IUA 個別メッセージヘッダから構成される。

4.4.7.10 解放指示メッセージ

設定指示メッセージは、SG または SG の要求によりマルチフレーム動作が解放されたことを通知するため、SG から ASP に送信される。

解放要求メッセージは、共通メッセージヘッダ、IUA 個別メッセージヘッダおよび以下のパラメタを持つ:

- 解放理由 (必須)
- 解放指示メッセージのパラメタ部フォーマットは解放要求メッセージと共通である。

4.4.7.10.1 解放理由

4.4.7.8.1 参照。ただし、解放指示メッセージでは「設定拒否」を使用しない。

4.5　手順

4.5.1　SCTP 管理サービス手順

3.5.1 参照。

4.5.2　メッセージ転送手順

一般的なメッセージ転送手順については 3.5.2 参照。ここでは、IUA 特有事項を述べる。

ASP 上の IUA から SG 上の IUA へのメッセージ転送手順を以下に示す:

1. SG を決定する。
2. SCTP アソシエーションを選択する。
3. D チャネルに基づいて SCTP ストリームを選択する。
4. 共通メッセージヘッダ、IUA 個別メッセージヘッダ、パラメタを設定し、メッセージを作成する。
5. SG 上の IUA へメッセージを送信する。

SG 上の IUA から ASP 上の IUA へのメッセージ転送手順を以下に示す:

1. インタフェース識別子に基づいて決定する。
2. AS 内の稼動状態 ASP を選択する。
3. D チャネルに基づいて SCTP ストリームを選択する。
4. 共通メッセージヘッダ、IUA 個別メッセージヘッダ、パラメタを設定し、メッセージを作成する。
5. SG 上の IUA へメッセージを送信する。

4.5.3 UA 管理サービス手順

3.5.3 参照。

4.5.4 ASP 管理サービス手順

3.5.4 参照。

4.5.5 AS 管理サービス手順

3.5.5 参照。

4.5.6 Q.921 サービス手順

4.5.6.1 データ転送手順

ASP から端末へのデータ転送手順を以下に示す:

1. ASP 上の IUA は DL-データ要求プリミティブを受信すると、データ要求メッセージを SG 上の IUA に送信する。
2. SG 上の IUA は IUA 個別メッセージヘッダのインタフェース識別子により D チャネルを特定し、データリンクコネクション識別子により端末を特定する。
3. SG 上の IUA は Q.921 確認型情報転送サービスを使用して、端末へデータ転送する。

端末から ASP へのデータ転送手順を以下に示す:

1. 端末は Q.921 確認型情報転送サービスを使用して、SG へデータ転送する。

SG 上の IUA は、データ指示メッセージを ASP へ送信する。

4.5.6.2 ユニットデータ転送手順

ASP から端末へのユニットデータ転送手順を以下に示す:

1. ASP 上の IUA は DL-ユニットデータ要求プリミティブを受信すると、ユニットデータ要求メッセージを SG 上の IUA に送信する。
2. SG 上の IUA は IUA 個別メッセージヘッダのインタフェース識別子により D チャネルを特定し、データリンクコネクション識別子により端末を特定する。
3. SG 上の IUA は Q.921 非確認型情報転送サービスを使用して、端末へデータ転送する。

端末から ASP へのユニットデータ転送手順を以下に示す:

1. 端末は Q.921 非確認型情報転送サービスを使用して、SG へデータ転送する。
2. SG 上の IUA は、ユニットデータ指示メッセージを ASP へ送信する。

4.5.6.3 データリンク設定手順

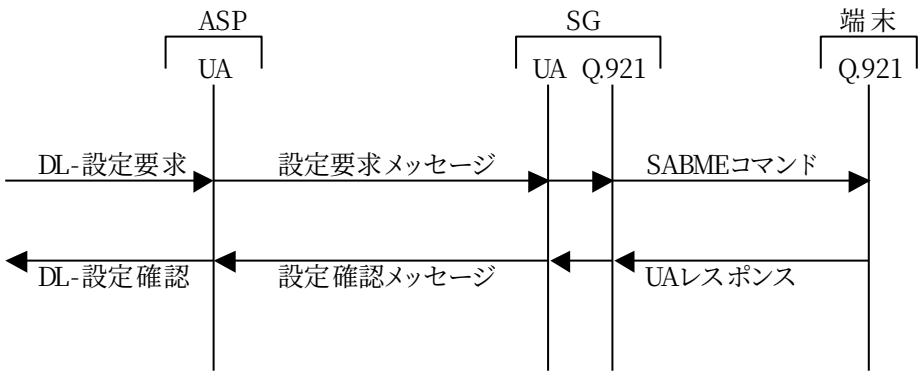
データリンク設定は ASP が要求する場合と端末が要求する場合がある。まず ASP が要求する場合のデータリンク設定手順を示す:

1. ASP 上の IUA は DL-設定要求プリミティブを受信し、SG 上の IUA へ設定要求メッセージを送信する。
2. SG 上の IUA は設定要求メッセージを受信し、SG 上の Q.921 は指定された端末へ Q.921 SABME コマ

ンドを送信する。

SG 上の Q.921 は端末から UA レスポンスを受信し、SG 上の IUA は ASP 上の IUA へ設定確認メッセージを送信する。

ASP 上の IUA は設定確認メッセージを受信し、DL-設定確認プリミティブを発行する。



上記手順のシーケンスを図 4-9 に示す。

図 4-9 ASP が要求する場合のデータリンク設定シーケンス

上記手順 1 において、ASP 上の IUA は設定要求メッセージを送信する際に応答監視タイマを起動してもよい。応答監視タイマは、設定確認メッセージまたは設定指示メッセージを受信すると停止する。応答監視タイマが満了すると、ASP 上の IUA は設定要求メッセージを再送する。

また、端末がデータリンク設定を拒否する場合、上記手順の 3 と 4 は以下のようになる:

3. SG 上の Q.921 は端末から DM レスポンスを受信し、SG 上の IUA は ASP 上の IUA へ解放確認メッセージを送信する。
4. ASP 上の IUA は解放確認メッセージを受信し、DL-解放確認プリミティブを発行する。

上記手順のシーケンスを図 4-10 に示す。

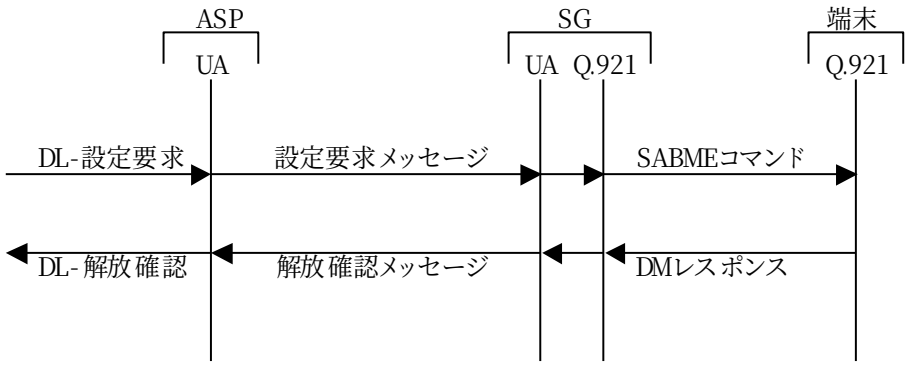
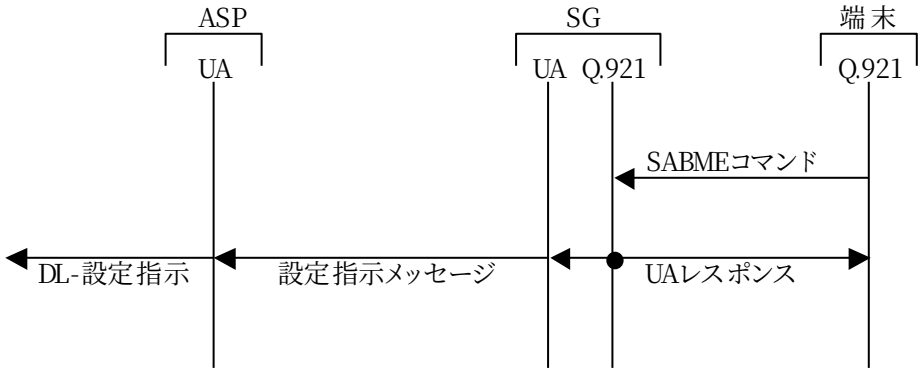


図 4-10 データリンク設定失敗例

続いて、端末が要求する場合のデータリンク解放シーケンスを示す:

1. 端末は SG 上の Q.921 へ SABME コマンドを送信する。
2. SG 上の Q.921 は端末へ UA レスポンスを送信するとともに、SG 上の IUA は ASP 上の IUA へ設定指示メッセージを送信する。

3. ASP 上の IUA は設定指示メッセージを受信し、DL-設定指示プリミティブを発行する。



上記手順のシーケンスを図 4-11 に示す。

図 4-11 端末が要求する場合のデータリンク設定シーケンス

4.5.6.4 **データリンク解放手順**

データリンク解放は ASP が要求する場合と SG が要求する場合と端末が要求する場合がある。まず ASP が要求する場合のデータリンク設定手順を示す:

1. ASP 上の IUA は DL-解放要求プリミティブを受信し、SG 上の IUA へ解放要求メッセージを送信する。

SG 上の IUA は解放要求メッセージを受信し、SG 上の Q.921 は指定された端末へ Q.921 DISC コマンドを送信する。

SG 上の Q.921 は端末から UA レスポンスまたは DM レスポンスを受信し、SG 上の IUA は ASP 上の IUA へ解放確認メッセージを送信する。

ASP 上の IUA は解放確認メッセージを受信し、DL-解放確認プリミティブを発行する。

上記手順のシーケンスを図 4-12 に示す。

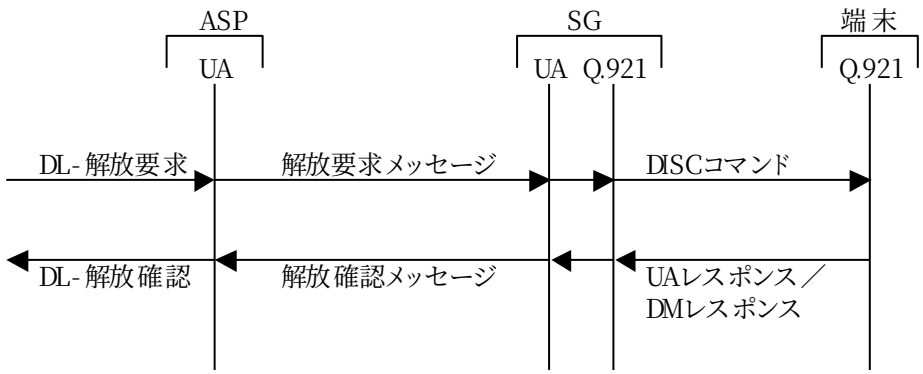


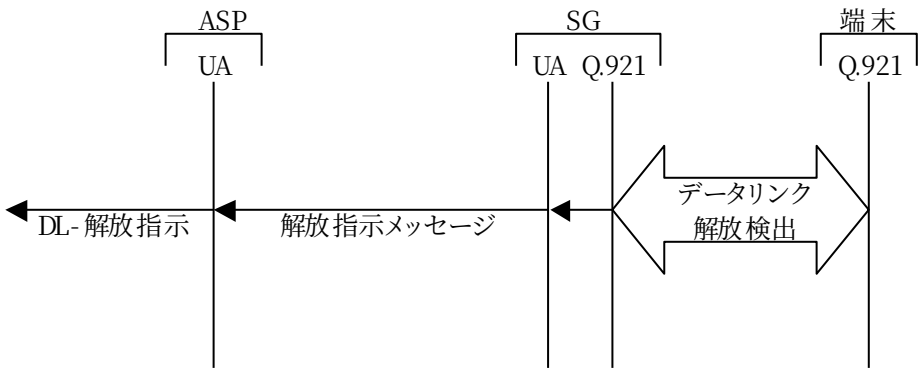
図 4-12 ASP が要求する場合のデータリンク解放シーケンス

解放要求メッセージ送信後一定時間以内に解放確認メッセージを受信しない場合、ASP 上の UA は解放要求メッセージを再送してもよい。再送後も解放確認メッセージを受信しない場合、ASP 上の UA はデータリンクが解放されたものと判断することができる。

続いて SG がデータリンク解放を要求する場合の手順を示す:

1. SG は物理層警報等により Q.921 データリンクが解放されたものと判断し、ASP 上の IUA へ解放指示メッセージを送信する。

2. ASP 上の IUA は解放指示メッセージを受信し、DL-解放指示プリミティブを発行する。



上記手順のシーケンスを図 4-13 に示す。

図 4-13 SG が要求する場合のデータリンク解放シーケンス

最後に端末が要求する場合のデータリンク解放手順を示す:

1. 端末上の Q.921 は、DISC コマンドを SG 上の Q.921 に送信する。
2. SG 上の Q.921 は UA レスポンスまたは DM レスポンスを端末上の Q.921 に送信するとともに、SG 上の IUA は解放指示メッセージを ASP 上の IUA に送信する。
3. ASP 上の IUA は解放指示メッセージを受信し、DL-解放指示プリミティブを発行する。

上記手順のシーケンスを図 4-14 に示す。

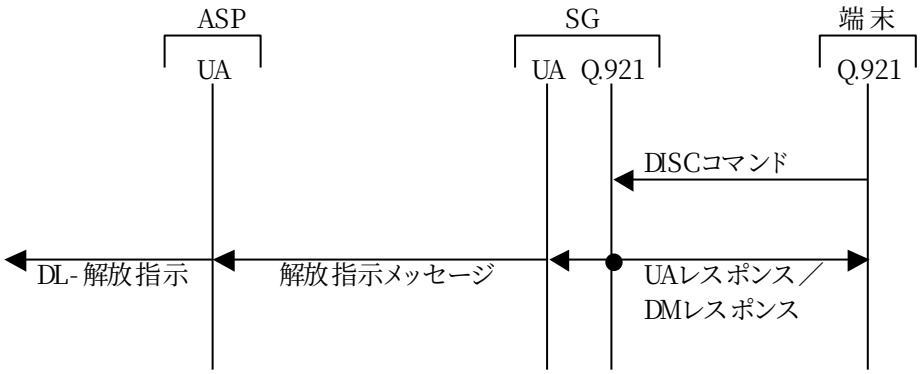


図 4-14 端末が要求する場合のデータリンク解放シーケンス

4.6 通信シーケンス例

4.6.1 初期化シーケンス

3.6.1 参照。

4.6.2 フェイルオーバシーケンス

3.6.2 参照。

4.6.3 Q.921 メッセージのシーケンス例

4.6.3.1 設定／データ転送／解放シーケンス例

データリンクを設定し、データを転送し、データリンクを解放するシーケンス例を図 4-15 に示す。

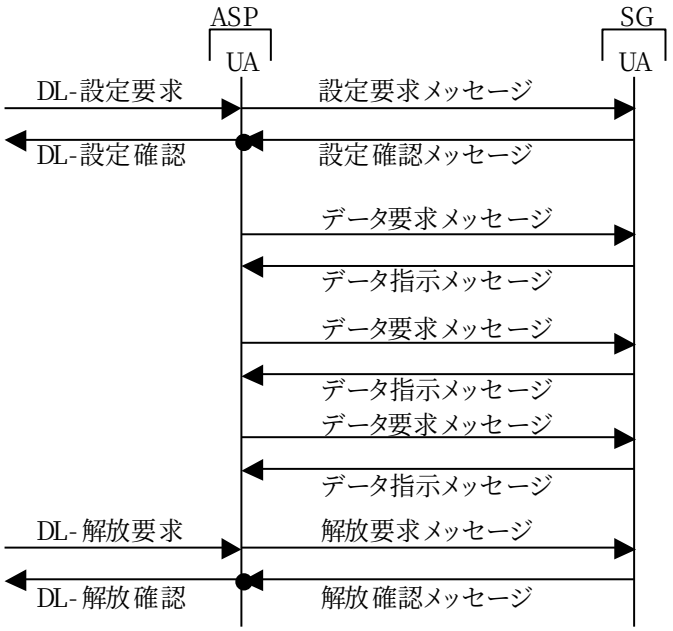


図 4-15 設定／データ転送／解放シーケンス例

4.6.3.2 割り当てられていない TEI へのデータ送信要求

割り当てられていない TEI に対するデータ送信要求を受信すると、SG は ERR メッセージで通知する。ASP 上の IUA は TEI 状態要求メッセージと TEI 状態確認メッセージを使用して、TEI 状態を確認することができる。シーケンスを図 4-16 に示す。

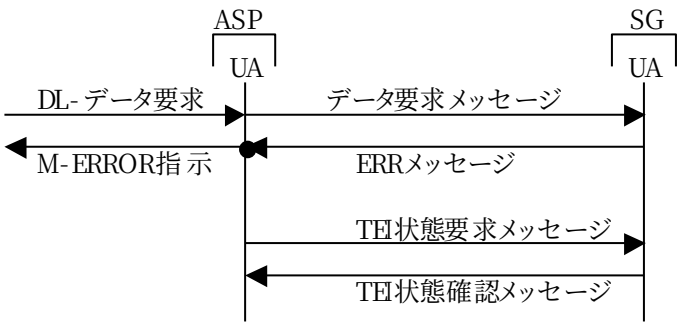


図 4-16 割り当てられてない TEI へのデータ送信要求

4.7 セキュリティ

3.7 参照。

4.8 登録番号

4.8.1 SCTP ペイロードプロトコル識別子

IUA のペイロードプロトコル識別子の値は”1”である。

4.8.2 ポート番号

SCTP、UDP および TCP におけるポート番号は”9900”である。

4.9 将来の拡張性

3.9 参照。

5．M3UA

5.1 序論

M3UA は、ISUP や SCCP などの MTP3 ユーザ部プロトコルを IP 網上で転送するプロトコルである。下位レイヤプロトコルとして SCTP を推奨する。

5.2 サービス

M3UA は以下のサービスを使用する。

- SCTP 管理サービス
- UA 管理サービス
- ASP 管理サービス
- AS 管理サービス
- M3UA サービス

5.2.1 SCTP 管理サービス

3.3.1 参照。

5.2.2 UA 管理サービス

3.3.2 参照。

5.2.3ASP 管理サービス

3.3.3 参照。

5.2.4 AS 管理サービス

3.3.4 参照。

5.2.5 MTP3 サービス

MTP3 サービスのサービスプリミティブ一覧を表 5-1 に示す。

表 5-1 MTP3 サービスプリミティブ一覧

プリミティブ		概要
MTP—転送	要求 指示	M3UA ユーザのメッセージを転送する。
MTP—休止	指示	特定の SS7 信号局への MTP—転送サービスが利用できないことを通知する。
MTP—再開	指示	特定の SS7 信号局への MTP—転送サービスが利用可能となったことを通知する。
MTP—状態表示	指示	特定の SS7 信号局への MTP—転送サービスが輻輳中であることを通知する。

5.3 メッセージ

M3UA は以下のメッセージクラスを使用する。

- UA 管理メッセージクラス
- MTP3 メッセージクラス
- 共通線信号網メッセージクラス
- ASP 状態管理メッセージクラス
- ASP トラフィック状態管理メッセージクラス
- ルーティングキー管理メッセージクラス

5.3.1 共通メッセージヘッダ

3.4.1 参照。

5.3.2 個別メッセージヘッダ

M3UA は個別メッセージヘッダを使用しない。

5.3.3 パラメタ

3.4.3 参照。

5.3.4 UA 管理メッセージクラス

ERR メッセージと NTFY メッセージを使用する。

5.3.4.1 ERR メッセージ (ERRor)

ERR メッセージは、受け付けたメッセージに対するエラーイベントを送信元 UA に通知するために使用する。例として、予期しないタイプのメッセージを受信した場合や、パラメタ値が不正だった場合などである。

ERR メッセージは共通メッセージヘッダのみを持つ。ERR メッセージは以下のパラメタを含む。

- エラーコード（必須）

- **ルーティングコンテキスト（必須）**
- **ネットワークアピアランス（必須）**
- **罹障対地（必須）**
- **診断情報（省略可能）**

ERR **メッセージのフォーマットを図** 5-1 **に示す。**

図 5-1 ERR メッセージのフォーマット

エラーコードは、ERR **メッセージの理由を示す。エラーコードの値は** UA **毎に定義する。**M3UA **のエラーコード値を表** 5-2 **に示す。**

表 5-2 M3UA エラーコード

エラー名称	エラーコード
無効バージョン	０ｘ０１
未使用	０ｘ０２
該当メッセージクラスなし	０ｘ０３
該当メッセージタイプなし	０ｘ０４
該当トラフィックモードなし	０ｘ０５
予期しないメッセージ	０ｘ０６
プロトコルエラー	０ｘ０７
未使用	０ｘ０８
無効ルーティングコンテキスト	０ｘ０８
無効ストリーム識別子	０ｘ０９
未使用	０ｘ０A
未使用	０ｘ０B
未使用	０ｘ０C
管理拒否	０ｘ０D
ASP 識別子要求	０ｘ０E
無効 ASP 識別子	０ｘ０F
未使用	０ｘ１０
無効パラメタ値	０ｘ１１
パラメタフィールドエラー	０ｘ１２
予期しないパラメタ	０ｘ１３
着信信号局状態未知	０ｘ１４
無効ネットワークアピアランス	０ｘ１５
パラメタ誤り	０ｘ１６
未使用	０ｘ１７
未使用	０ｘ１８

エラー名称	エラーコード
無効ルーティングコンテキスト	０ｘ１９
ASP に対する AS の未配備	０ｘ２０

「無効バージョン」エラーは、無効あるいはサポートしないバージョンのメッセージの受信時に使用する。ERR メッセージはサポートするバージョンを共通メッセージヘッダに含む。ERR メッセージはサポートするバージョンを診断情報に含んでもよい。

「該当メッセージクラスなし」エラーは、予期しないまたはサポートしないクラスのメッセージを受信した場合に送信する。

「該当メッセージタイプなし」エラーは、予期しないまたはサポートしないタイプのメッセージを受信した場合に送信する。

「該当トラフィックモードなし」エラーは、ASPAC メッセージが指定するトラフィックモードをサポートしない場合に送信する。

「予期しないメッセージ」エラーは、定義されたメッセージではあるが、当該遷移状態では期待したいものを受信した場合に使用する。

「プロトコルエラー」エラーは、プロトコル異常発生時に使用する。

「無効ストリーム識別子」エラーは、予期しない SCTP ストリーム上でメッセージを受信した場合に使用する。0 番以外のストリーム上で管理メッセージを受信する場合などを含む。

「管理拒否」エラーは、管理部がロックアウト状態にある場合等、管理部の理由により ASPUP メッセージ、および ASPAC メッセージを拒否する場合に使用する。

「ASP 識別子要求」エラーは、SG が ASP 識別子を必要とし、受信した ASPUP メッセージに ASP 識別子が含まれていなかった場合に使用する。

「無効 ASP 識別子」エラーは、SG が無効な ASP 識別子を含んだメッセージを受信した際に使用する。

「無効パラメタ値」エラーは、パラメタ値が不正ある場合に使用する。

「パラメタフィールドエラー」は、誤ったパラメタ長を含んだメッセージを受信した際に使用する。

「予期しないパラメタ」エラーは、無効パラメタを含んだメッセージを受信した際に使用する。

「着信信号局状態未知」エラーは、SG が DAUD メッセージにより着信信号局の状態を問合された場合に問合せもとが監査できない等の理由により、このメッセージに応答しない場合に使用する。

「無効ネットワークアピアランス」エラーは、SG が ASP から無効なネットワークアピアランス値を含んだメッセージを受信した際に使用する。このエラーを使用する場合にはネットワークアピアランスパラメタに受信した無効なネットワークアピアランス値を含まなければならない。

「パラメタ誤り」エラーは、必須パラメタが設定されていないメッセージを受信した際に使用される。

「無効ルーティングコンテキスト」エラーは、無効なルーティングコンテキスト値を含んだメッセージを受信した際に使用する。このエラーを使用する場合にはルーティングコンテキストパラメタに受信した無効なルーティングコンテキスト値を含まなければならない。

「ASP に対する AS の未配備」エラーは、M3UA ピアーからルーティングコンテキストパラメタを含まないメッセージを受信し、どの AS に関するデータか未知の際に使用する。

5.3.4.2 NTFY メッセージ (Notify)

NTFY メッセージは、M3UA イベントを自律的に M3UA ピアーに通知するために使用する。

M3UA の NTFY メッセージは、共通メッセージヘッダのみを使用し、以下のパラメタを含む。

- **ステータス（必須）**
- ASP **識別子**
- **ルーティングコンテキスト（省略可能）**
- **付加情報（省略可能）**

NTFY **メッセージフォーマットを図** 5-2 **に示す。**

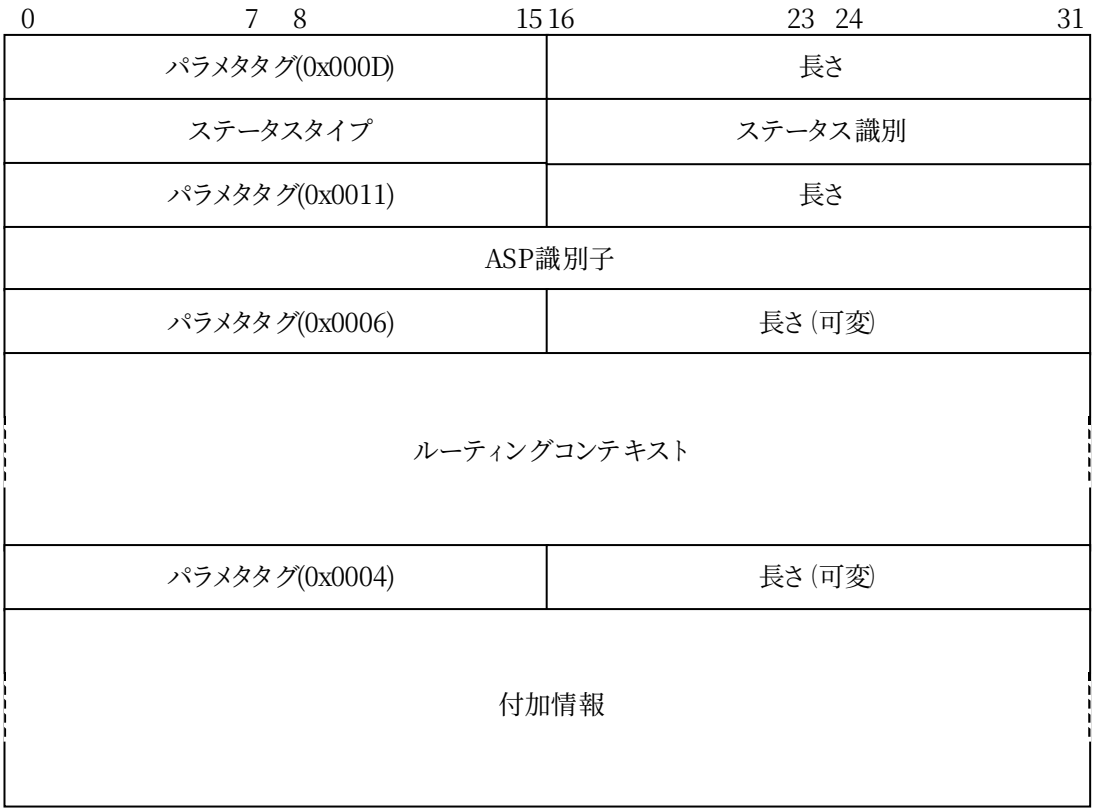


図 5-2 M3UA **の** NTFY **メッセージフォーマット**

5.3.4.2.1 **ステータスタイプ／ステータス識別**

ステータスタイプは、NTFY **メッセージの種別を示す** 16 **ビット符号無し整数であり** 5-3 **に示す値をとる。**

表 5-3 **ステータスタイプ一覧**

値	**説明**
0x1	AS **状態変更**
0x2	**その他**

ステータス種別は 16 **ビット符号無し整数であり、ステータスタイプ毎に値を定義する。ステータスタイプが「**AS **状態変更」の場合に取り得る値を表** 5-4 **に示す。**

表 5-4 AS 状態変更のステータス識別

値	説明
1	(予約)
2	AS 起動状態
3	AS 稼動状態
4	AS 保留状態

上記通知は、AS 状態変更時に SG から ASP に送られる。
ステータスタイプが「その他」の場合、ステータス識別は表 5-5 に示す値を取る。

表 5-5 その他のステータス識別

値	記述
1	ASP リソース不足
2	代替 ASP 稼動
3	ASP フェイラー

上記通知は、ASP または AS の状態変更に起因しない。ASP リソース不足通知は、ロードシェアモードにおいて、負荷を処理するために ASP 追加が必要であることを不活性 ASP に通知する。代替 ASP 活性化通知は、オーバーライドモードにおいて、代替 ASP が活性状態に遷移し、負荷を引き継いだことを通知する。ASP ファイラーは SG が ASP の非稼動状態を通知する。

5.3.5 MTP3 メッセージクラス

5.3.5.1 ペイロードデータメッセージ

ペイロードデータメッセージは、MTP3 ユーザ部プロトコルデータを転送するために使用する。

5.3.5.1.1 パラメタとフォーマット

ペイロードデータメッセージは以下のパラメタを含む:

- ネットワークアピアランス(省略可能)
- ルーティングコンテキスト(省略可能)
- プロトコルデータ (必須)
- 相関 ID (省略可能)

ペイロードデータメッセージのパラメタ部フォーマットを図 5-3 に示す。

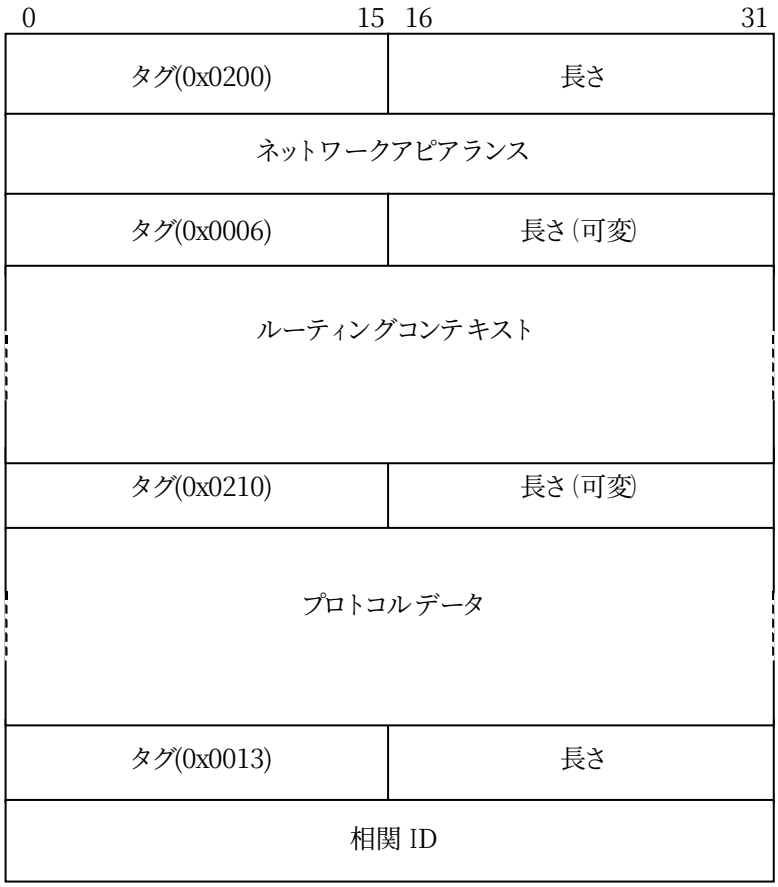


図 5-3 ペイロードメッセージのフォーマット

5.3.5.1.2 **ネットワークアピアランス**

ASP が SG を経由して複数の共通線信号網と通信する場合、個々の共通線信号網とのトラフィックを区別するパラメタとしてネットワークアピアランスを用いる。ネットワークアピアランスの値は、ASP と SG の組ごとに異なってよい。

5.3.5.1.3 **ルーティングコンテキスト**

受信側に対してトラフィックフローを明示するためにルーティングコンテキストを用いる。

5.3.5.1.4 **プロトコルデータ**

プロトコルデータは、以下の形式を取る。ペイロードデータメッセージのパラメタ部フォーマットを図 5-4 **に示す。**

5.3.5.1.4.1 **プロトコルデートの形式**

プロトコルデータは以下のパラメタを含む:

- **サービス情報オクテット**(SIO : Service Information Octet)
  サービス表示(SI : Service Indicator)
  ネットワーク表示(NI : Network Indicator)
  予備／優先度
- **ルーティングラベル**

着信号局コード(DPC : Destination Point Code)

発信号局コード(OPC : Originating Point Code)

信号リンク選択番号(SLS : Signaling Link Selection)

- **プロトコルデータ**

  MTP3 **ユーザプロトコルデータ**(ISUP、SCCP **など**)

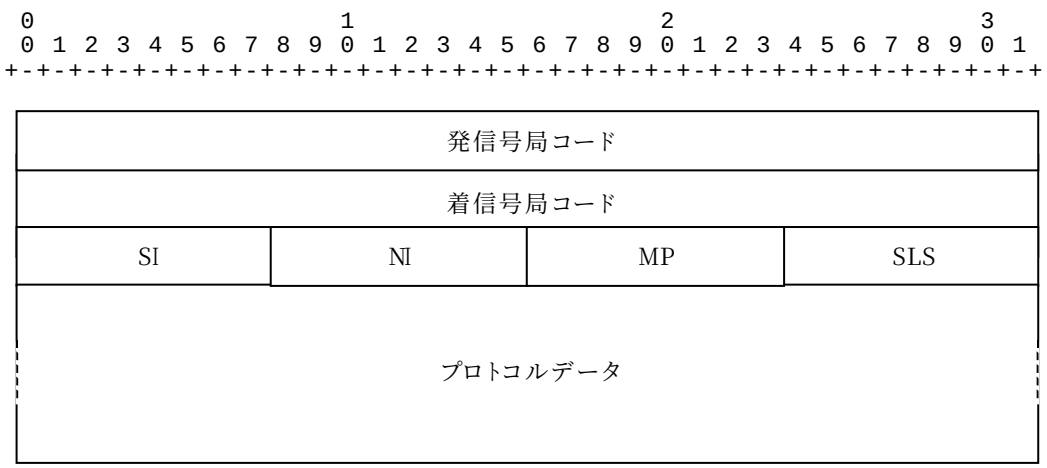


図 5-4 プロトコルデータのフォーマット

5.3.5.1.5 **相関** ID

相関 ID **はプロトコルデータによって転送される** MSU **を** AS **内に限定して、送り側の** M3UA **によりユニークに付与される。**

5.3.6 **共通線信号網管理メッセージクラス**

5.3.6.1 DUNA **メッセージ** (Destination UNAvailable)

DUNA **メッセージは、到達不能な信号局を通知するために** SG **上の** M3UA **から** ASP **上の** M3UA **に送信する。**ASP **からのメッセージが到達不能な信号局に宛てである場合にも、到達不能であることを通知するために** DUNA **メッセージを使用する。**SG **は後に続く「応答の」**DUNA **メッセージを抑止してもよい。**ASP **内の** M3UA **ユーザは、到達不能な信号局宛てのメッセージ送信を停止する。**

5.3.6.1.1 **パラメタとフォーマット**

DUNA **メッセージは以下のパラメタを含む:**

- **ネットワークアピアランス(省略可能)**
- **ルーティングコンテキスト（省略可能）**
- **罹障対地(必須)**
- **付加情報(省略可能)**

DUNA **メッセージのパラメタ部フォーマットを図** 5-5 **に示す。**

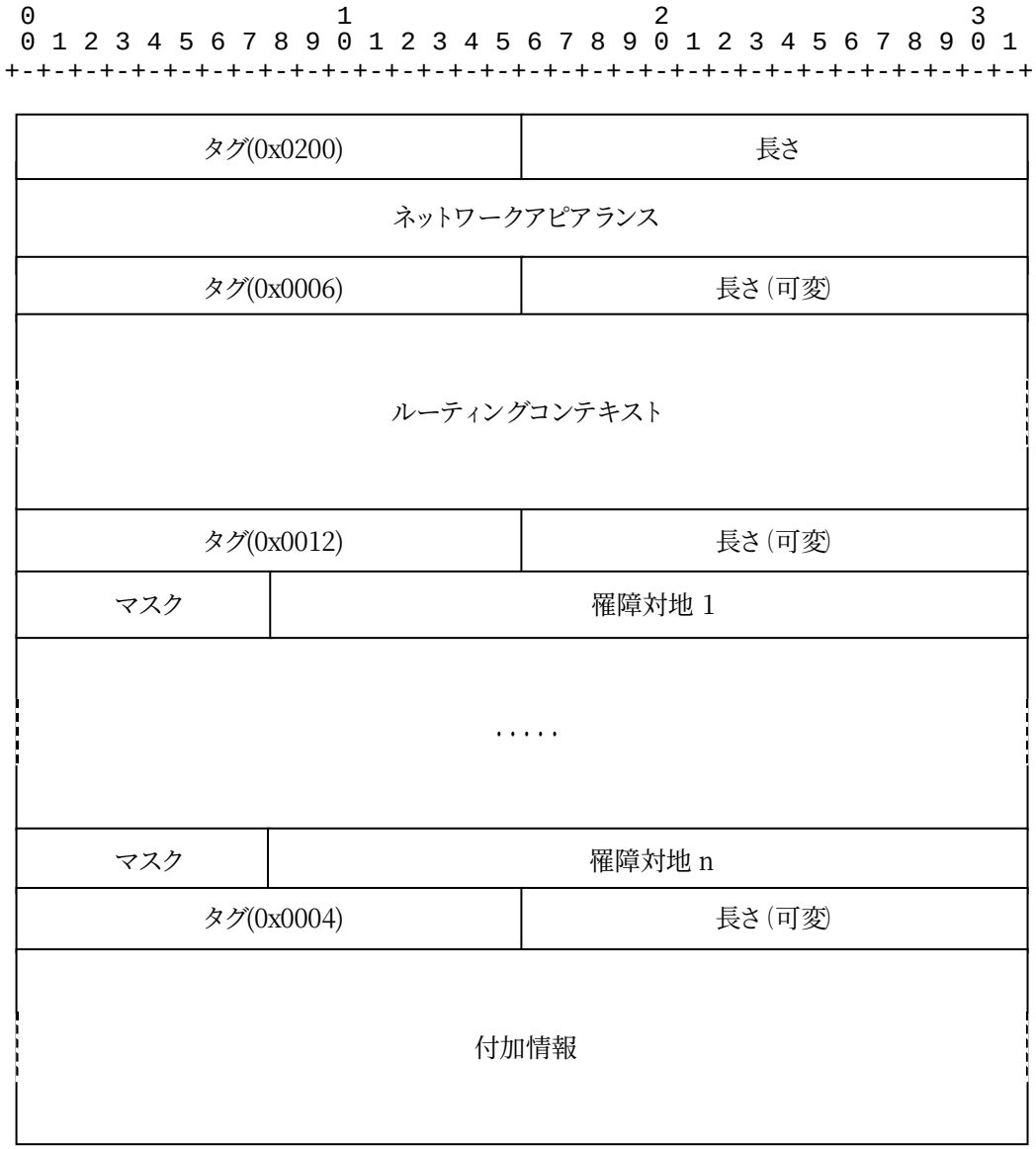


図 5-5　DUNA メッセージのフォーマット

5.3.6.1.2 **ネットワークアピアランス**

5.3.5.1.2 **参照**。

5.3.6.1.3 **ルーティングコンテキスト**

5.3.5.1.3 **参照**。

5.3.6.1.4 **罹障対地**

罹障対地は、到達不能な信号局コードを最大で 16 **含む**。

5.3.6.1.4.1 **到達不能信号局コード**

到達不能信号局コードは最大 24 **ビットまでの信号局コードを設定可能である**。24 **ビット未満の信号局コードは右詰設定し、**0 **でパディングする**。ITU-T **仕様** 14 **ビット信号局コード、**TTC **仕様** 16 **ビット信号局コード設定例をそれぞれ図** 5-6**、図** 5-7 **に示す**。

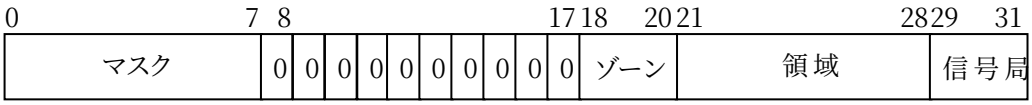


図 5-6 ITU-T 14 ビット信号局コード

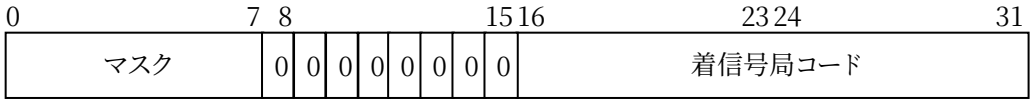


図 5-7 TTC16 ビット信号局コード

複数の到達不能信号局コードを DUNA **メッセージに設定することは、送信側ではオプションであるが、受信側では正しく処理しなければならない。また、全ての到達不能信号局コードは、同一のネットワークアピアランスに属さなければならない。**

5.3.6.1.4.2 **マスク**

マスクは、連続した到達不能信号局コードを表現するために使用する。マスクは、8 **ビット符号無し整数で表現され、到達不能信号局コードの下位** n **ビットがワイルドカードであることを示す。**

5.3.6.1.5 **付加情報**

3.4.4.2.1.3 **参照**。

5.3.6.2 DAVA **メッセージ** (Destination AVAilbale)

DAVA **メッセージは、到達不能であった信号局が到達可能になったことを通知するために** SG **から** ASP **へ送信する。または、**DAUD **メッセージに対する応答としても使用する。**

5.3.6.2.1 **パラメタとフォーマット**

DAVA **メッセージのパラメタとフォーマットは** DUNA **メッセージと共通である。**

5.3.6.3 DAUD **メッセージ** (Destionation state AUDit)

DAUD **メッセージは、罹障対地への経路の利用可能状態／輻輳状態を問い合わせるために** ASP **から** SG **へ送信する。**

5.3.6.3.1 **パラメタとフォーマット**

DAVA **メッセージのパラメタとフォーマットは** DUNA **メッセージと共通である。**

5.3.6.4 SCON (SS7 network CONgestion) **メッセージ**

SCON **メッセージは、宛先信号局への経路が輻輳状態であることを通知するために** SG **から** ASP **に送信する。ペイロードデータメッセージまたは** DAUD **メッセージに対する応答としても使用される。**MTP3 **の版数によっては、共通線信号網の輻輳レベル変更を契機として送信される場合もある**(ANSI MTP3 **など**)**。**SCON **メッセージは、**M3UA **または** ASP **の輻輳を契機として送信されることもある。**

5.3.6.4.1 **パラメタとフォーマット**

SCON **メッセージは以下のパラメタを含む**:

- **ネットワークアピアランス(省略可能)**
- **罹障対地(必須)**
- **ルーティングコンテキスト（省略可能）**
- **関連対地(省略可能)**
- **輻輳表示(省略可能)**
- **付加情報(省略可能)**

SCON **メッセージのパラメタ部フォーマットを図** 5-8 **に示す。**

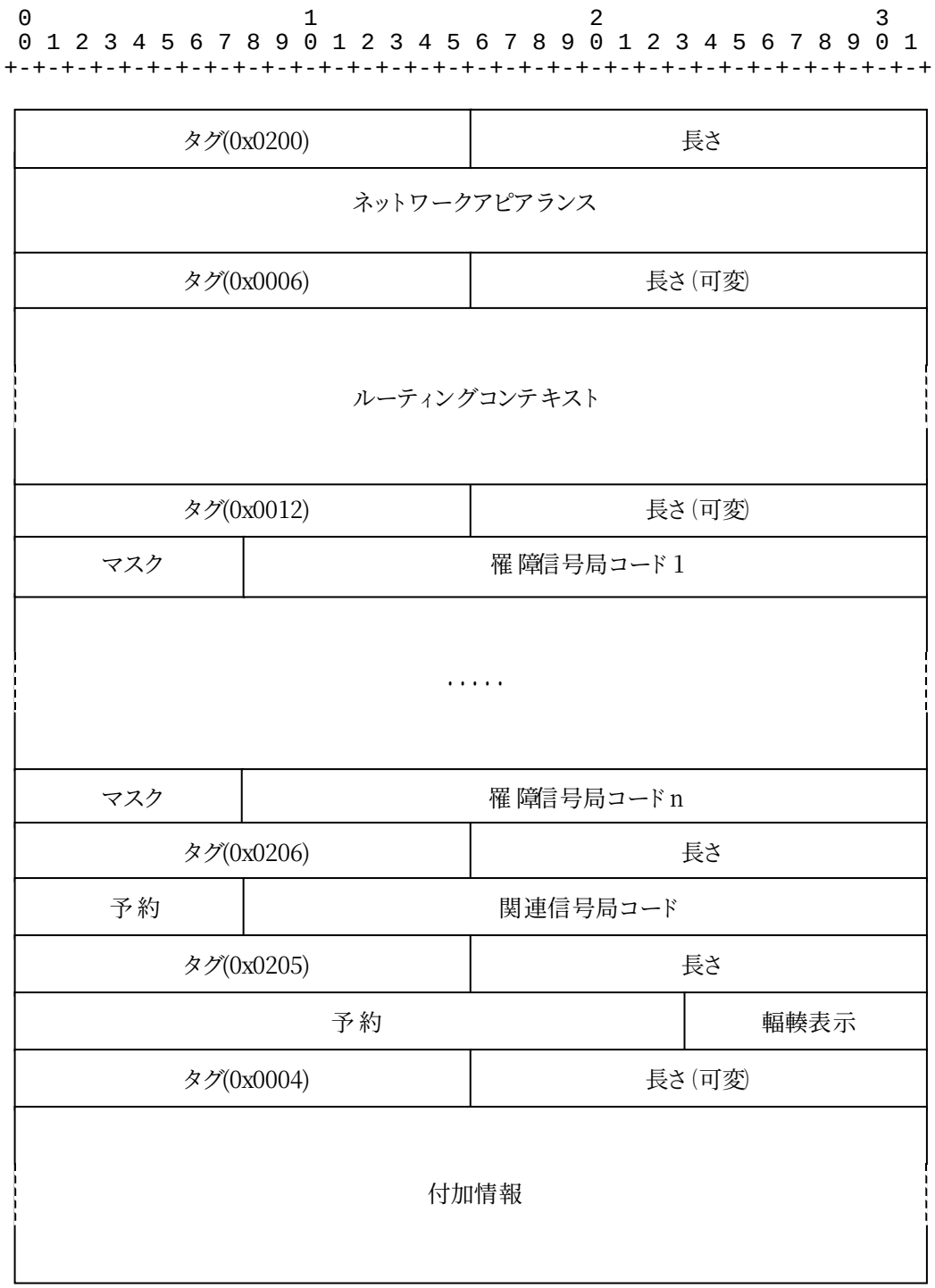


図 5-8 SCON メッセージのフォーマット

5.3.6.4.2 **ネットワークアピアランス**

5.3.5.1.2 **参照。**

5.3.6.4.3 **ルーティングコンテキスト**

5.3.5.1.3 **参照**

5.3.6.4.4 **罹障対地**

5.3.6.1.4 **と共通である。ただし、複数の罹障対を単一の** SCON **メッセージに設定するために、**SCON **メッ**

セージの送出を遅延させてはならない。

5.3.6.4.5 関連対地

関連対地パラメタは、ASP から SG に対して SCON メッセージを送信する場合のみ使用する。SG は関連対地パラメタが指定する信号局へ転送統制(TFC)メッセージを送信する。

5.3.6.4.6 輻輳表示

輻輳表示パラメタは 24 ビットの予約フィールドと 8 ビットの輻輳レベルフィールドから構成される。輻輳表示パラメタは複数の輻輳レベルを持つ国内網において使用される(ANSI MTP3 など)。

5.3.6.4.6.1 輻輳レベル

輻輳レベルは 8 ビット符号無し整数であり、表 5-6 に示す値をとる。

表 5-6 輻輳レベル

値	定義
0x00	輻輳無または未定義
0x01	輻輳レベル 1
0x02	輻輳レベル 2
0x03	輻輳レベル３

5.3.6.4.7 付加情報

3.4.4.2.1.3 参照。

5.3.6.5 DUPU メッセージ (Destination User Part Unavailable)

DUPU メッセージは、SS7 信号局の MTP3 ユーザ部(例えば ISUP や SCCP)が利用不可であることを通知するために、SG から ASP へ送信する。

5.3.6.5.1 パラメタとフォーマット

DUPU メッセージは以下のパラメタを含む:

- ネットワークアピアランス(省略可能)
- ルーティングコンテキスト（省略可能）
- 罹障対地(必須)
- ユーザ／理由(必須)
- 付加情報(省略可能)

DUPU メッセージのフォーマットを図 5-9 に示す。

図 5-9 DUPU メッセージのフォーマット

5.3.6.5.2 **ネットワークアピアランス**

5.3.5.1.2 **参照**。

5.3.6.5.3 **ルーティングコンテキスト**

5.3.5.1.3 **参照**。

5.3.6.5.4 **罹障対地**

DUPU の罹障対地パラメタは、マスクフィールドを使用できない点と、単一の罹障対地のみを設定できるという点で DUNA **メッセージ、**DAVA **メッセージ、および** DAUD **メッセージとは異なる。罹障対地の範囲またはリストを転送できないが、これは共通線信号網の動作と整合する。**MTP3 **のユーザ部利用不可メッセージは唯一つの罹障対地を含む。**

5.3.6.5.5 **ユーザ／理由**

MTP3 **ユーザ部識別子と利用不可理由から構成される。**

5.3.6.5.5.1 MTP3 ユーザ部識別子

MTP3 ユーザ部識別子は、利用不可能な MTP3 ユーザ部(ISUP や SCCP など)を識別する 16 ビット符号無し整数である。MTP3 ユーザ部識別子の取り得る値を下記に示す。MTP3 ユーザ部識別子の値は MTP3 のユーザ部利用不可メッセージ及びサービス表示(SI : Service Indicator)と一致する。MTP3 のプロトコル版数によっては、ここに示す以外の値を用いてもよい。MTP3 ユーザ部識別子の値は、MTP3 プロトコル勧告に準拠する。

<MTP3 ユーザ部識別子の取り得る値>

0 から 2	予約
3	SCCP
4	TUP
5	ISUP
6 から 8	予約
9	広帯域 ISUP
10	衛星 ISUP
11	予約
12	AAL タイプ２シグナリング
13	BICC
14	GW 制御プロトコル
15	予約

5.3.6.5.5.2 利用不可理由

利用不可理由は、MTP3 ユーザ部が利用不可能である理由を示す。利用不可理由の取り得る値を下記に示す。利用不可理由の値は MTP3 のユーザ部利用不可メッセージと一致する。MTP3 のプロトコル版数によっては、ここに示す以外の値を用いてもよい。MTP3 ユーザ部識別子の値は、MTP3 プロトコル勧告に準拠する。

<利用不可理由の取り得る値>

0	不明
1	ユーザ部未実装
2	ユーザ部通信不可能

5.3.6.5.6 付加情報

3.4.4.2.1.3 参照。

5.3.6.6 DRST メッセージ (Destination ReSTricted)

DRST メッセージは、転送制限信号局を通知するために SG から ASP に送信する。DAUD メッセージに対する応答として使用してもよい。代替 SG が利用可能であれば、DRST メッセージを受信した ASP は転送制限信号局に向けてメッセージを送信してもよい。到達不能信号局に関する DRST メッセージを SG から受信した場合、ASP は SG を経由して転送制限信号局との通信を再開できる。

DRTS メッセージの使用はオプションである。

5.3.6.6.1 パラメタとフォーマット

DRST メッセージは以下のパラメタを含む:

- ネットワークアピアランス(省略可能)

- **ルーティングキーコンテキスト（省略可能）**
- **罹障対地(必須)**
- **付加情報(省略可能)**

5.3.6.6.2 **ネットワークアピアランス**

5.3.5.1.2 **参照**。

5.3.6.6.3 **罹障対地**

5.3.6.1.4 **参照**。

5.3.6.6.4 **付加情報**

3.4.4.2.1.3 **参照**。

5.3.7 ASP **状態管理メッセージクラス**

3.4.6 **参照**。

5.3.8 ASP **トラフィック状態管理メッセージクラス**

5.3.8.1 ASPAC **メッセージ** (ASP ACtive)

ASPAC **メッセージは、**ASP **が稼動状態に遷移することを通知する。ルーティングコンテキストパラメタを用いて** AS **を指定する場合、稼動状態への遷移は指定する** AS **内に限定される。**

M3UA **の** ASPAC **メッセージは以下のパラメタを含む。**

- **トラフィックモードタイプ(省略可能)**
- **ルーティングコンテキスト(省略可能)**
- **付加情報(省略可能)**

ASP Active **メッセージのフォーマットを図** 5-10 **に示す。**

図 5-10 ASP Active メッセージのフォーマット

5.3.8.1.1 **トラフィックモードタイプ**

トラフィックモードは、AS **における** ASP **の動作を指定する** 32 **ビット符号無し整数である。取り得る値を表** 5-7 **に示す**

表 5-7 トラフィックモード一覧

値	**説明**
0x01	**オーバーライド**
0x02	**ロードシェア**
0x03	**ブロードキャスト**

ルーティングコンテキスト内では、オーバーライド、ロードシェアとブロードキャストを混在できない。オーバーライドは、ASP **が** AS **の全トラフィックを処理することを示し、他に稼動状態** ASP **があればトラフィックを引き継ぐ。ロードシェアは、**ASP **が他の稼動状態** ASP **と共に** AS **のトラフィックを分担することを示す。**

ブロードキャストは ASP **が他の稼動状態** ASP **と同一のメッセージを受信することを示す。**

5.3.8.1.2 **ルーティングコンテキスト**

5.3.5.1.3 **参照。**

5.3.8.1.3 **付加情報**

3.4.4.2.1.3 **参照。**

5.3.8.2 ASPIA **メッセージ** (ASP InActive)

ASPAC **メッセージは、**ASP **が稼動状態に遷移することを通知する。ルーティングコンテキストパラメタを用いて** AS **を指定する場合、稼動状態への遷移は指定する** AS **内に限定される。**

ASPIA **メッセージは以下のパラメタを含む。**

- **ルーティングコンテキスト(省略可能)**
- **付加情報(省略可能)**

ASP Iactive **メッセージのフォーマットを図** 5-11 **に示す。**

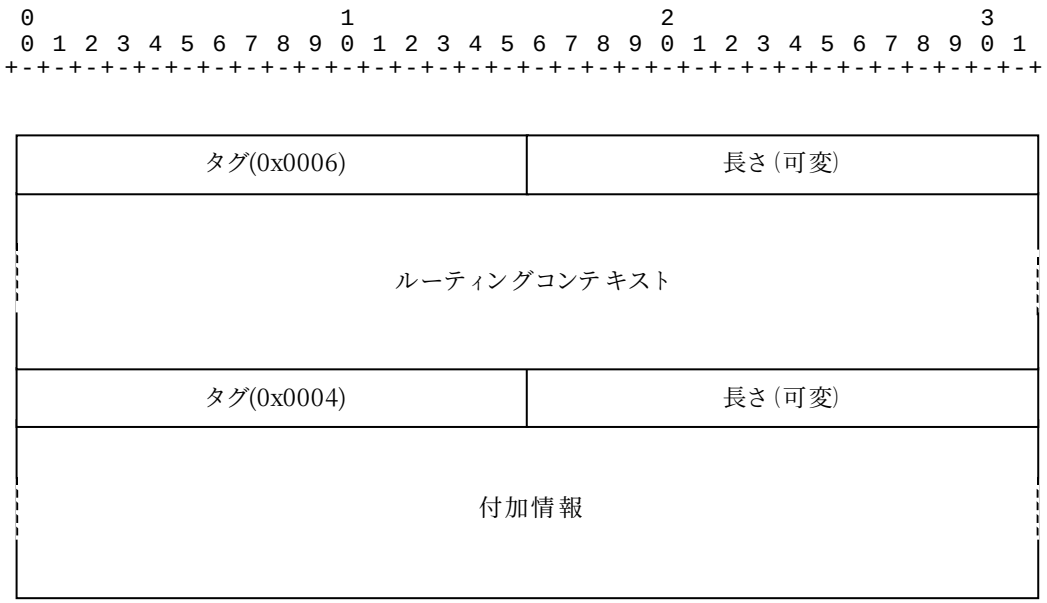


図 5-11 ASP Iactive メッセージのフォーマット

5.3.8.2.1 **ルーティングコンテキスト**

5.3.5.1.3 **参照。**

5.3.8.2.2 **付加情報**

3.4.4.2.1.3 **参照。**

5.3.9 **ルーティングキー管理メッセージクラス**

5.3.9.1 REG REQ **メッセージ** （REGistration REQuest）

REG REQ **メッセージはルーティングキーを登録するために** ASP **から** SG **に送信する。一つの** REG REQ **メッセージで複数ルーティングキーを登録可能である。**

5.3.9.1.1 **パラメタとフォーマット**

REQ REQ **メッセージは以下のパラメタを含む。**

- **ルーティングキー（必須）**

REG REQ **メッセージのパラメタ部フォーマットを図** 5-12 **に示す。**

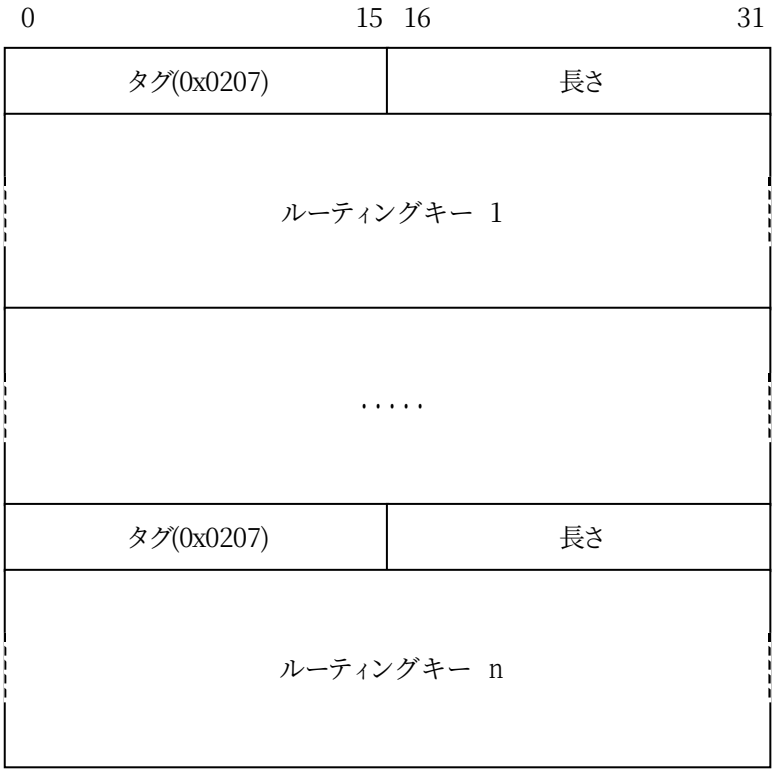


図 5-12 REG REQ **メッセージフォーマット**

5.3.9.1.2 ***ルーティングキー***

ルーティングキーパラメタのフォーマットを図 5-13 ***に示す。***

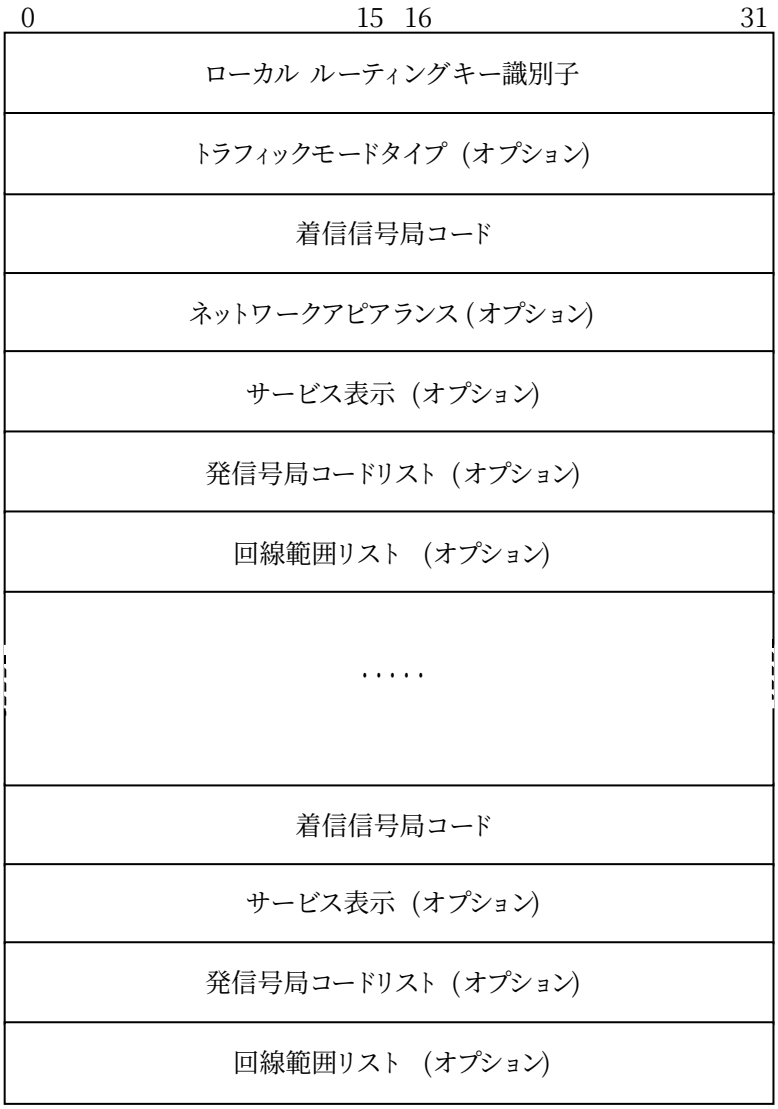


図 5-13 ルーティングキーパラメタフォーマット

5.3.9.1.2.1 **ローカルルーティングキー識別子**

ローカルルーティングキー識別子フィールドは登録要求と登録結果と対応付けるために使用する。ローカルルーティングキー識別子は ASP **上の** M3UA **が設定する。ローカルルーティングキー識別子フィールドのフォーマットを図** 5-14 **に示す。**

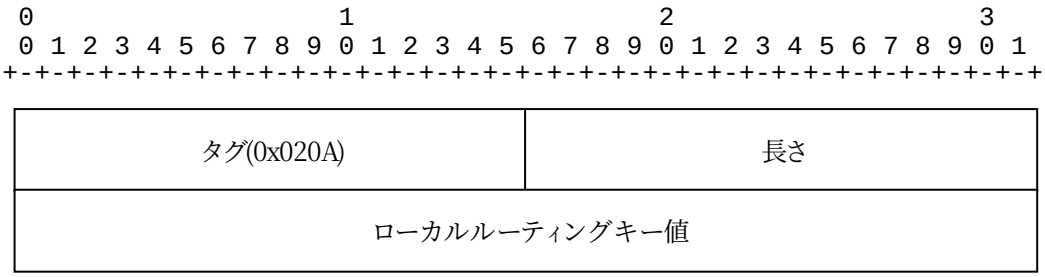


図 5-14 ローカルルーティングキー識別子フィールドフォーマット

5.3.9.1.2.2 **トラフィックモードタイプ**

トラフィックモードタイプフィールドは ASP **のトラフィックモードを表す。フォーマットを図** 5-15**-に示す。**

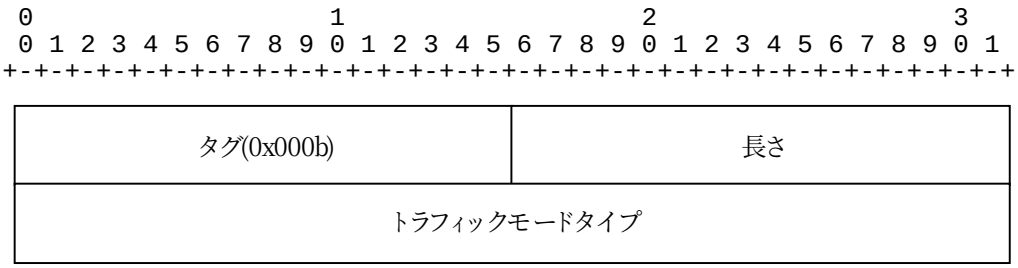


図 5-15 トラフィックモードタイプのフォーマット

5.3.9.1.2.3 **着局信号コード**

着信号局コードフィールドはルーティングキーを構成する着信号局コードを表す。フォーマットは罹障対地パラメタ(5.3.6.1.4)**と共通である。フォーマットを図** 5-16 **に示す。**

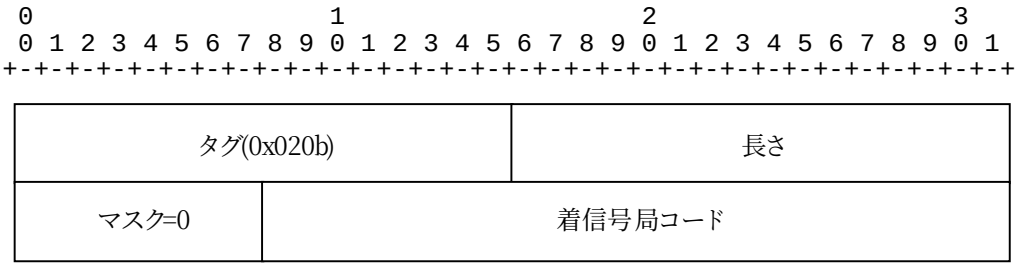


図 5-16 着信号局コードのフォーマット

5.3.9.1.2.4 **ネットワークアピアランス**

ネットワークアピアランスフィールドは、ルーティングキーを構成するネットワークアピアランスを表す。ネットワークアピアランスフィールドは省略可能であり、省略時は全てのネットワークアピアランスを意味

する。フォーマットは 5.3.5.1.2 参照。

5.3.9.1.2.5 **サービス表示**

サービス表示フィールドは、ルーティングキーを構成するサービス表示を表す。サービス表示フィールドは省略可能であり、省略時は全てのサービス表示を意味する。フォーマットを図 5-17 **に示す。**

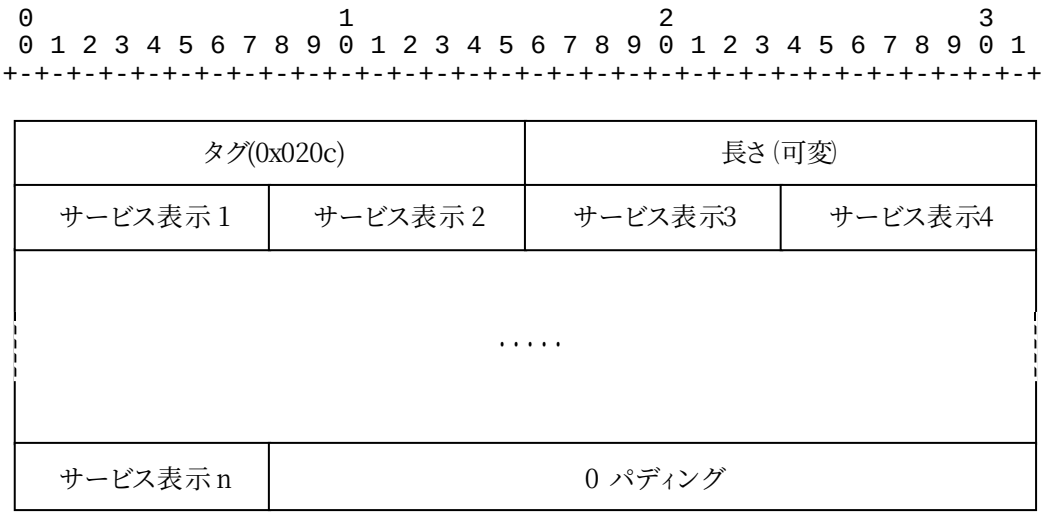


図 5-17 サービス情報フィールドフォーマット

5.3.9.1.2.6 **発信号局コードリスト**

発信号局コードリストフィールドは、ルーティングキーを構成する発信号局コードのリストを表す。発信号局コードリストフィールドは省略可能であり、省略時は全ての発信号局コードを意味する。発信号局コードリストフィールドのフォーマットを図 5-18 **に示す。**

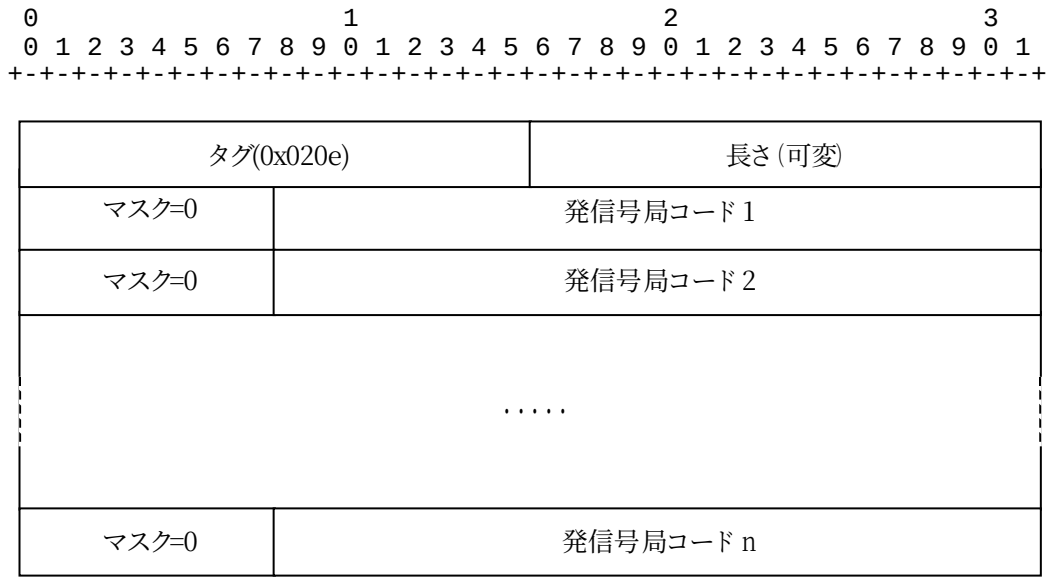


図 5-18 発信号局コードリストのフォーマット

5.3.9.1.2.7 **回線範囲リスト**

回線範囲リストフィールドはルーティングキーを構成する回線範囲リストを表す。回線範囲は発信号局コード、着信号局コードおよび回線番号範囲により指定する。回線範囲リスト中の個々の回線範囲は、発信号局コードと回線番号範囲(開始回線番号と終了回線番号により指定される範囲)を設定する。着信号局コードは、着信号局コードフィールドの値を使用する。回線範囲リストフィールドは省略可能である。フォーマットを図 5-19 **に示す。**

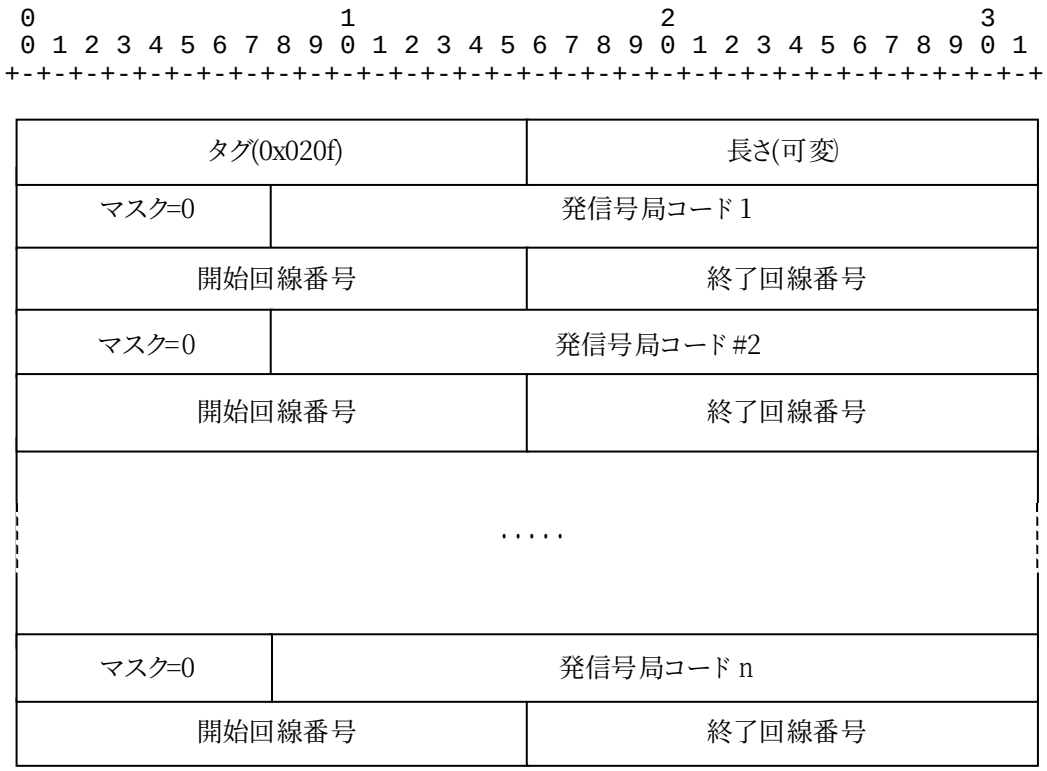


図 5-19 **回線範囲フィールドのフォーマット**

5.3.9.2 REG RSP（Registration Response）**メッセージ**

REG RSP **メッセージは** REG REO **メッセージの応答として使用する。**REG RSP **メッセージはルーティングキーの登録成否を示し、登録に成功したルーティングキーに対してルーティングコンテキスト値を返送する。**REQ RSP **メッセージは以下のパラメタを含む**:

- **登録結果 (必須)**

REG REQ **メッセージのパラメタ部フォーマットを図** 5-20 **に示す。**

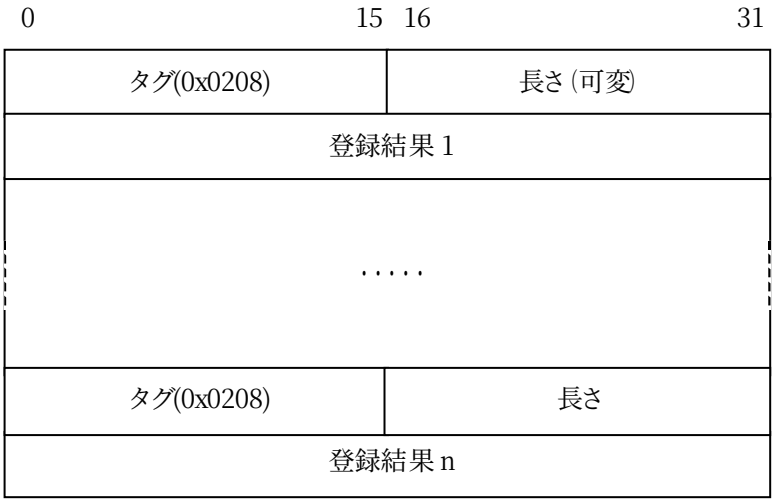


図 5-20 REG RSP メッセージフォーマット

5.3.9.2.1 **登録結果**

登録結果パラメタには、以下のフィールドから構成される:

- **ローカルルーティングキー識別子**
- **登録状態**
- **ルーティングコンテキスト**

登録結果パラメタのフォーマットを図 5-21 **に示す。**

図 5-21 登録結果パラメタフォーマット

5.3.9.2.1.1 **ローカルルーティングキー識別子**

5.3.9.1.2.1 **参照。**

5.3.9.2.1.2 **登録状態**

登録結果フィールドはルーティングキーの登録状態を示す。取り得る値を以下に示す:

＜登録状態＞

0 **登録成功**

1 エラー－未知

2	エラー－不整合な着信号局コード
3	エラー－不整合なネットワークアピアランス
4	エラー－不整合なルーティングキー
5	エラー－不許可
6	エラー－ルーティングキーの重複
7	エラー－ルーティングキーなし
8	エラー－リソース不足
9	エラー－サポートしていないルーティングキーパラメタフィールド
10	エラー－サポートしていない／不整合なトラフィックハンドリングモード

5.3.9.2.1.3 **ルーティングコンテキスト**

登録に成功したルーティングキーに対するルーティングコンテキストを設定する。登録失敗時は”0x00000000”**を設定する。**

5.3.9.3 DEREG REQ **メッセージ（**DE‐Registration Request**）メッセージ**

DEREG REQ **メッセージはルーティングキーの登録を解除するために使用する。**DEREQ REQ **メッセージは以下のパラメタを含む。**

- **ルーティングコンテキストリスト（必須）**

DEREG REQ **メッセージのパラメタ部フォーマットを図** 5-22 **に示す。**

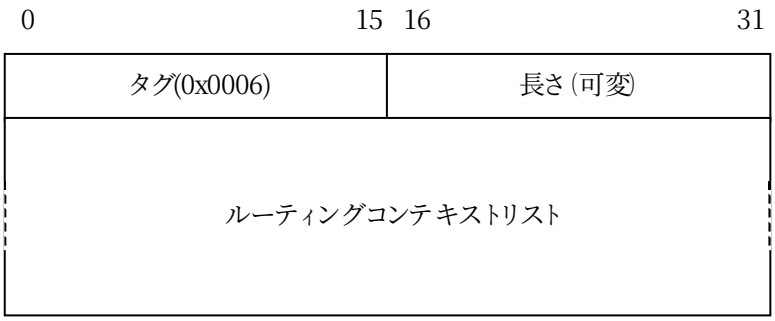


図 5-22 DEREG REQ メッセージフォーマット

5.3.9.3.1 **ルーティングコンテキストリスト**

登録を解除するルーティングコンテキストを設定する。複数設定可能である。

5.3.9.4 DEREG RSP**（**DE‐Registration Response**）メッセージ**

DEREG RSP **メッセージは** DEREG REQ **メッセージに対する応答として使用する。**DEREQ RSP **メッセージは以下のパラメタを含む。**

- **登録解除結果（必須）**

DEREG REQ **メッセージのパラメタ部フォーマットを図** 5-23 **に示す。**

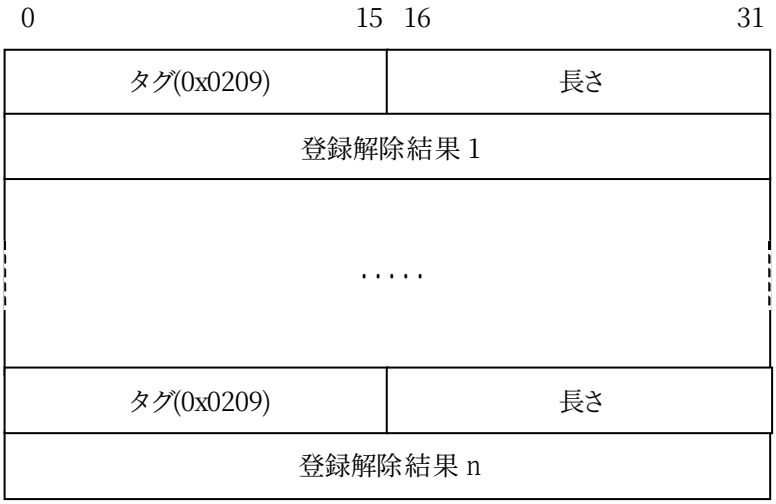


図 5-23 DEREG RSP メッセージフォーマット

5.3.9.4.1 **登録解除結果**

登録解除結果はルーティングキーの登録解除状態を設定する。

登録解除結果パラメタのフォーマットを図 5-24 **に示す。**

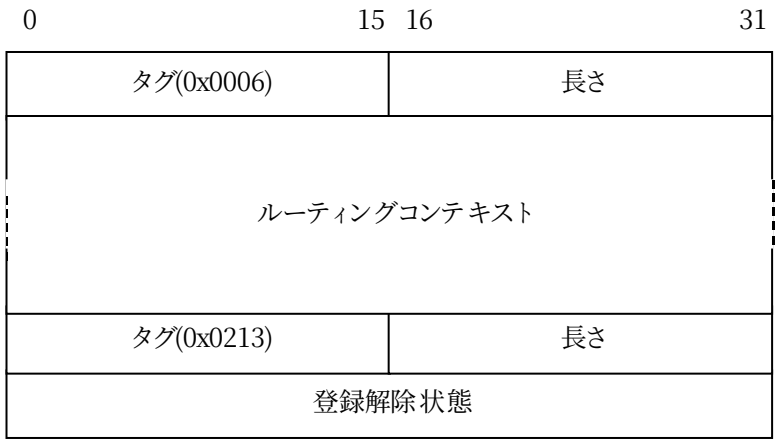


図 5-24 登録解除結果パラメタフォーマット

登録解除結果パラメタは以下のフィールドから構成される:

- **ルーティングコンテキスト**
- **登録解除状態**

5.3.9.4.1.1 **ルーティングコンテキスト**

ルーティングコンテキストフィールドは、登録解除状態を示すルーティングキーを指定する。

5.3.9.4.1.2 **登録解除状態**

登録解除状態フィールドは、ルーティングコンテキストフィールドにより指定されるルーティングキーの登録解除の状態を示す。取り得る値を以下に示す:

＜登録解除状態＞

0	登録解除成功
1	エラー－未知
2	エラー－不整合なルーティングコンテキスト
3	エラー－不許可
4	エラー－登録なし
5	エラー－ルーティングコンテキストに対する APS が稼動状態にある

5.4 手順

5.4.1 SCTP 管理サービス手順

3.5.1 参照。

5.4.2 UA 管理サービス手順

3.5.3 参照。

5.4.3 ASP 管理サービス手順

3.5.4 参照。

5.4.4 AS 管理サービス手順

3.5.5.参照。

5.4.5 MTP3 サービス手順

5.4.5.1 MTP 転送手順

MTP 転送手順は以下の手順から構成される:

- ASP から SG への転送手順
- SG から ASP への転送手順

5.4.5.1.1 ASP から SG への転送手順

ASP から SG への転送手順を以下に示す:

1. ASP 上の M3UA ユーザ部は MTP-転送要求プリミティブを発行する。

ASP 上の M3UA は、必要であればネットワークアピアランスパラメタを補完して、ペイロードデータメッセージを送信する。

SG 上の M3UA はペイロードデータメッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して MTP3 エンティティを選択する。

SG 上の M3UA は選択した MTP3 エンティティに対して MTP-転送要求プリミティブを発行する。

上記手順の通信シーケンスを図 5-25 に示す。

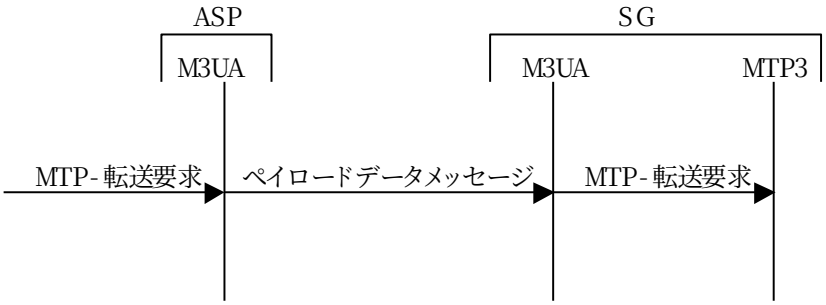


図 5-25 ASP から SG への転送シーケンス

5.4.5.1.2 SG から ASP への転送手順

SG から ASP への転送手順を以下に示す:

2. SG 上の MTP は MTP-転送指示プリミティブを発行する。

SG 上の M3UA は、必要であればネットワークアピアランスパラメタを補完して、ペイロードデータメッセージを送信する。

ASP 上の M3UA はペイロードデータメッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して M3UA ユーザ部を選択する。

ASP 上の M3UA は選択した M3UA ユーザ部に対して MTP-転送指示プリミティブを発行する。

上記手順の通信シーケンスをに示す。

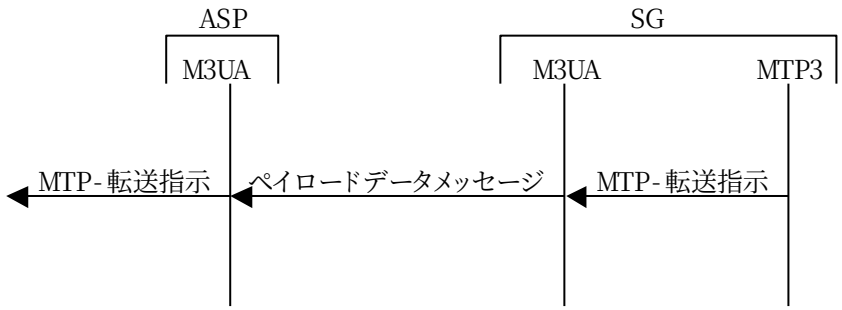


図 5-26 ASP から SG への転送シーケンス

5.4.5.2 MTP 休止手順

MTP 休止手順を以下に示す:

3. SG 上の MTP3 は到達不能な信号局を検出し、MTP-休止指示プリミティブを発行する。

SG 上の M3UA は、必要であればネットワークアピアランスパラメタを補完して、DUNA メッセージを送信する。

ASP 上の M3UA は DUNA メッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して M3UA ユーザ部を選択する。

4. ASP 上の M3UA は選択した M3UA ユーザ部に対して MTP-休止指示プリミティブを発行する。

上記手順の通信シーケンスを図 5-27 に示す。

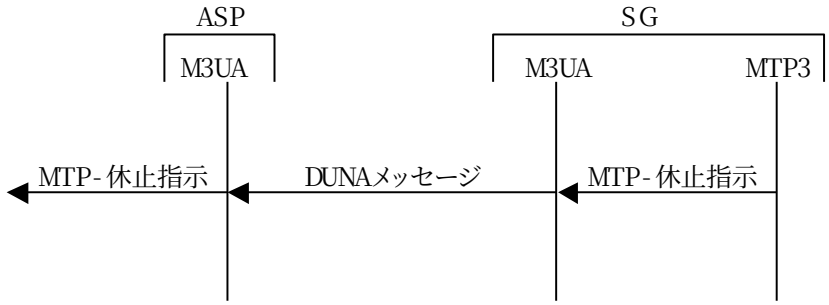


図 5-27 休止シーケンス

5.4.5.3 MTP **再開手順**

MTP **再開手順を以下に示す**:

5. SG **上の** MTP3 **は到達不能だった信号局が到達可能になると、**MTP-**再開指示プリミティブを発行する。**

SG **上の** M3UA **は、必要であればネットワークアピアランスパラメタを補完して、**DAVA **メッセージを送信する。**

ASP **上の** M3UA **は** DAVA **メッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して** M3UA **ユーザ部を選択する。**

ASP **上の** M3UA **は選択した** M3UA **ユーザ部に対して** MTP-**再開指示プリミティブを発行する。**

上記手順の通信シーケンスを図 5-28 **に示す。**

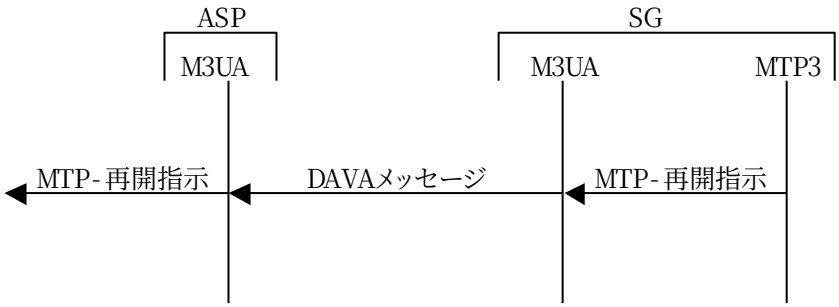


図 5-28 再開シーケンス

5.4.5.4 MTP **状態表示手順**

MTP **状態表示手順は輻輳通知手順とユーザ部利用不可通知手順から構成される。**

5.4.5.4.1 **輻輳通知手順**

輻輳通知手順を以下に示す:

6. SG **上の** MTP3 **は信号局への通信経路輻輳を検出すると、**MTP-**状態表示指示プリミティブを発行する。**

SG **上の** M3UA **は、必要であればネットワークアピアランスパラメタを補完して、**SCON **メッセージを送信する。**

ASP **上の** M3UA **は** SCON **メッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して** M3UA **ユーザ部を選択する。**

ASP **上の** M3UA **は選択した** M3UA **ユーザ部に対して** MTP-**状態表示指示プリミティブを発行する。**

上記手順の通信シーケンスを図 5-29 **に示す。**

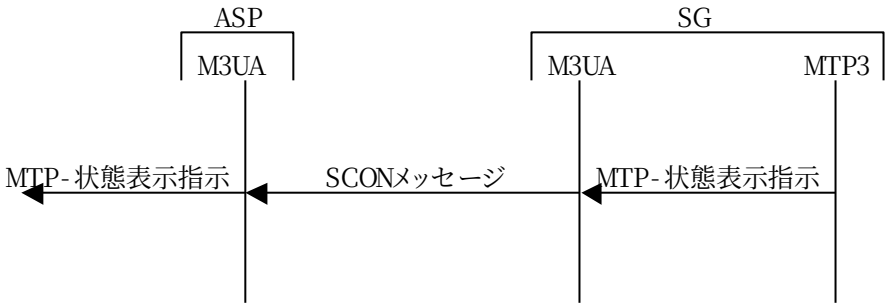


図 5-29 輻輳通知シーケンス

5.4.5.4.2 **ユーザ部利用不可通知手順**

ユーザ部利用不可通知手順を以下に示す:

7. SG **上の** MTP3 **は** SS7 **信号局のユーザ部利用不可を検出すると、**MTP-**状態表示指示プリミティブを発行する。**

SG **上の** M3UA **は、必要であればネットワークアピアランスパラメタを補完して、**DUPU **メッセージを送信する。**

ASP **上の** M3UA **は** DUPU **メッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して** M3UA **ユーザ部を選択する。**

ASP **上の** M3UA **は選択した** M3UA **ユーザ部に対して** MTP-**状態表示指示プリミティブを発行する。**

上記手順のシーケンスを図 5-30 **に示す。**

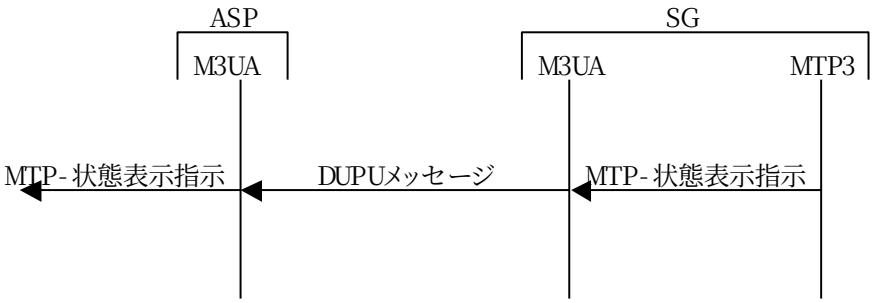


図 5-30 ユーザ部利用不可通知シーケンス

5.4.5.5 **転送制限手順**

転送制限手順を以下に示す:

8. SG **上の** MTP3 **は転送制限信号局を検出すると、**MTP-**状態表示指示プリミティブを発行する。**

SG **上の** M3UA **は、必要であればネットワークアピアランスパラメタを補完して、**DRST **メッセージを送信する。**

ASP **上の** M3UA **は** DRST **メッセージを受信し、ネットワークアピアランスパラメタが設定されていれば取得して** M3UA **ユーザ部を選択する。**

ASP **上の** M3UA **は選択した** M3UA **ユーザ部に対して** MTP-**状態表示指示プリミティブを発行する。**

上記手順のシーケンスを図 5-31 **に示す。転送制限手順は必須ではない。**

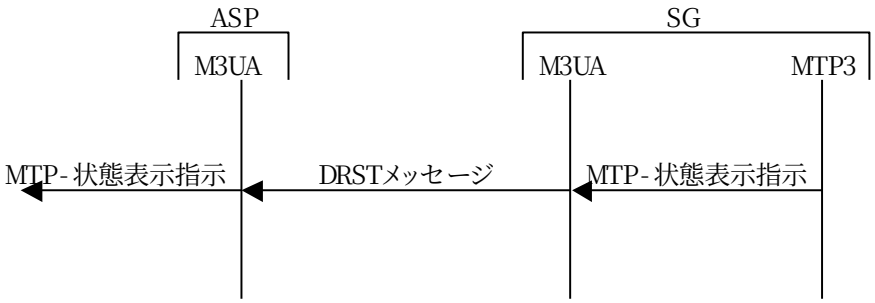


図 5-31 転送制限シーケンス

5.4.5.6 **対地監査手順**

対地監査手順を以下に示す:

9. ASP と SG の一時的な通信途絶等を契機として、ASP 上の M3UA は SG 上の M3UA に対して DAUD メッセージを送信して信号局状態を問い合わせる。
10. SG上のM3UAはDAUDメッセージを受信すると、問い合わせを受けた信号局の状態に応じて、DUNA／DAVA／SCON／DUPU／DRST メッセージを返送する。
11. ASP の正当性が確認出来ない場合等において、DAUD メッセージに応答しない場合には ERR メッセージを返送する。上記手順のシーケンスを図 5-32 に示す。

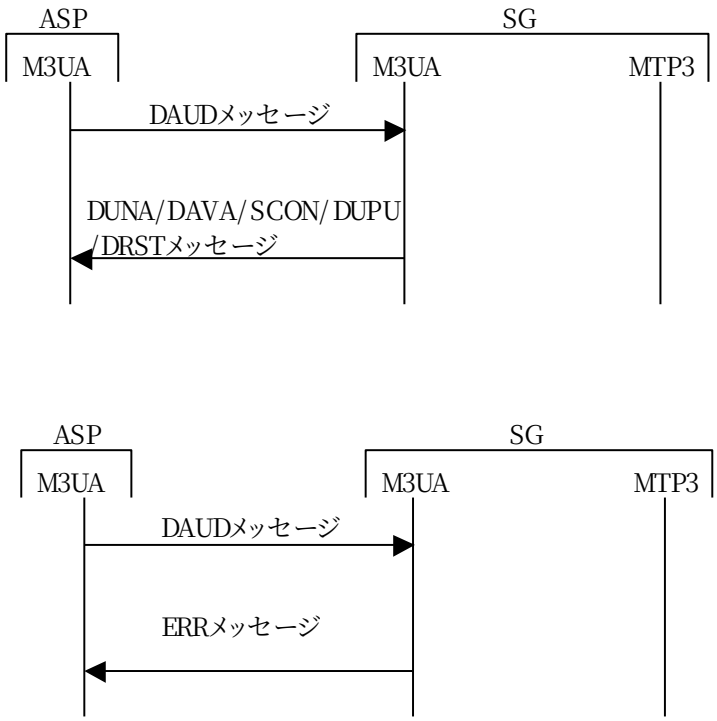


図 5-32 対地監査シーケンス

5.4.6 **ルーティングキー管理サービス手順**

ルーティングキー管理サービス手順は必須ではない。

5.4.6.1 ルーティングキー登録手順

ルーティングキー登録手順を以下に示す:

12. ASP 上の M3UA は REG REQ メッセージを送信する。

SG 上の M3UA は REQ REQ メッセージを受信し、ルーティングキーと既存ルーティングキーを比較する。

ルーティングキーが既存ルーティングキーと一致する場合、ルーティングキーに関連付けられている AS に ASP を追加することができる。

SG は REQ RSP(登録成功)メッセージを送信する。

上記手順 2 において、ルーティングキーと既存ルーティングキーが一致しない場合、手順 3 は次のように変化する。

ルーティングキーが既存ルーティングキーと一致しない場合、ルーティングキーと関連付けられる新たな AS を生成し、ASP を AS に追加する。

5.4.6.2 ルーティングキー登録解除手順

ルーティングキー登録解除手順を以下に示す:

ASP 上の M3UA は DEREG REQ メッセージを送信する。

SG 上の M3UA は DEREQ REQ メッセージを受信し、ルーティングキーに関連付けられている AS から ASP を解除する。

SG は DEREQ RSP(登録解除成功)メッセージを送信する。

5.5 通信シーケンス例

5.5.1 初期化シーケンス

3.6.1 参照。

5.5.2 フェイルオーバシーケンス

3.6.2 参照。

5.6 セキュリティ

3.7 参照。

5.7 登録番号

5.7.1 SCTP ペイロードプロトコル識別子

M3UA のペイロードプロトコル識別子は”3”である。

5.7.2 ポート番号

M3UA の登録ポート番号は 2905 である。

5.8 将来の拡張性

3.9 参照。

6. M2UA

6.1 序論

回線交換網上で使用されているシグナリング・プロトコルを IP 網上において SG を介し MGC に送り届ける必要がある。M2UA は MTP3 をユーザとし、IP 網上において MTP3 の信号の送受信を可能にする。この時

に必要となるのが以下の機能である。

- MTP2／MTP3 間のインタフェース領域のサポート
- SG 及び MGC 上のレイヤ管理モジュール間の通信サポート
- SG・MGC 間における SCTP アクティブ・アソシエーションの管理のサポート

SG は MTP2 を終端し、MGC は MTP3 以上のレベルを終端する。言い換えれば、SG は IP 網上で MTP3 以上のメッセージを MGC へ伝送する。

6.2 用語

6.2.1 インタフェース

本章において、インタフェースは共通線信号網の信号リンクを意味する。

6.2.2 インタフェース識別子

インタフェースを識別する整数または文字列である。

6.3 サービス

M2UA は以下のサービスを使用する。

- SCTP 管理サービス
- UA 管理サービス
- ASP 管理サービス
- AS 管理サービス
- MTP2 サービス

6.3.1 SCTP 管理サービス

3.3.1 参照。

6.3.2 UA 管理サービス

3.3.2 参照。

6.3.3 ASP 管理サービス

3.3.3 参照。

6.3.4 AS 管理サービス

3.3.4 参照。

6.3.5 MTP2 サービス

表 6-1 MTP2 サービス一覧

プリミティブ		概要
転送	要求 指示	MTP3 プロトコルデータを転送する。
開始	要求 確認	SG と遠隔信号局の間の信号リンク設定する。
解放	要求 指示 確認	SG と遠隔信号局の間の信号リンク解放する。
リンク状態	要求 指示 確認	SG と遠隔信号局の間の信号リンク状態管理に使用する。
回収	要求 指示 確認	有意信号ユニットの回収に使用する。

6.4 メッセージ

M2UA は以下のメッセージクラスを使用する。

- UA 管理メッセージクラス
- ASP 状態管理メッセージクラス
- ASP トラフィック管理メッセージクラス
- MTP2 メッセージクラス

6.4.1 共通メッセージヘッダ

3.4.1 参照。

6.4.2 個別メッセージヘッダ

M2UA 個別メッセージヘッダは以下のメッセージに使用する:

- MTP2 メッセージクラスの全メッセージ

M2UA 個別メッセージヘッダは以下のパラメタを持つ:

- インタフェース識別子

6.4.2.1 **インタフェース識別子**

インタフェース識別子は信号リンクを識別する。フォーマットは 4.4.2.1 **参照。**

6.4.3 **パラメタ**

3.4.3 **参照。**

6.4.4 UA **管理メッセージクラス**

ERR **メッセージと** NTFY **メッセージを使用する。**

6.4.4.1 ERR **メッセージ** (ERRor)

ERR **メッセージは、受け付けたメッセージに対するエラーイベントを送信元** UA **に通知するために使用する。例として、予期しないタイプのメッセージを受信した場合や、パラメタ値が不正だった場合などである。**

ERR **メッセージは共通メッセージヘッダのみを持つ。**ERR **メッセージは以下のパラメタを含む:**

- **エラーコード（必須）**
- **インタフェース識別子（必須）**
- **診断情報（省略可能）**

ERR **メッセージのフォーマットを図** 6-**１に示す。**

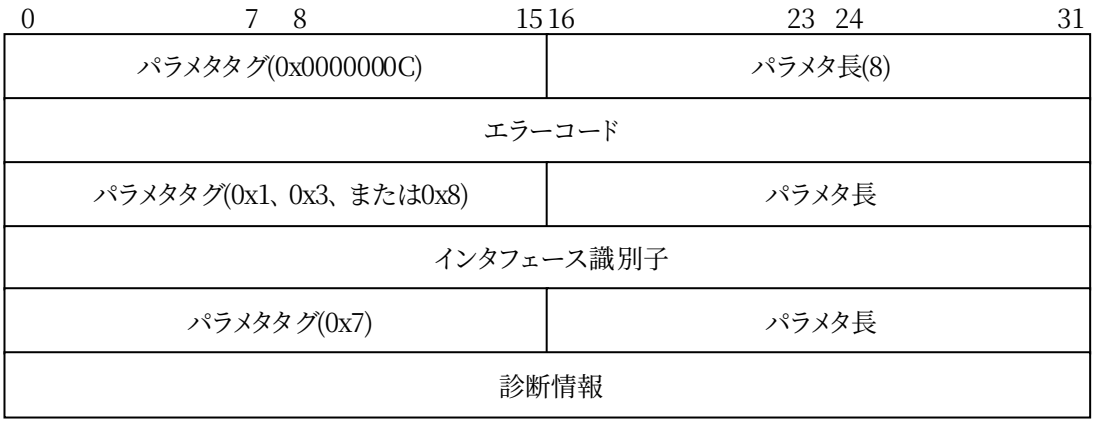


図 6-１ ERR **メッセージのフォーマット**

エラーコードは、ERR **メッセージの理由を示す。エラーコードを表** 6-2 **に示す。**

表 6-2 M2UA **エラーコード**

エラー名称	**エラーコード**
無効バージョン	**０ｘ０１**
無効インタフェース識別子	**０ｘ０２**
該当メッセージクラスなし	**０ｘ０３**
該当メッセージタイプなし	**０ｘ０４**

エラー名称	エラーコード
該当トラフィックモードなし	０ｘ０５
予期しないメッセージ	０ｘ０６
プロトコルエラー	０ｘ０７
サポートしないインタフェース識別子型	０ｘ０８
無効ストリーム識別子	０ｘ０９
M2UA では未使用	０ｘ０A～０ｘ０ｃ
拒否	０ｘ０ｄ
ASP 識別子を要求	０ｘ０ｅ
無効ＡＳＰ識別子	０ｘ０ｆ
インタフェース識別子に対してＡＳＰアクティブ	０ｘ１０
無効パラメタ値	０ｘ１１
パラメタフィールドエラー	０ｘ１２
予期しないパラメタ	０ｘ１３
M2UA では未使用	０ｘ１４～０ｘ１５
パラメタ不足	０ｘ１６

6.4.4.2 NTFY メッセージ (Notify)

NTFY メッセージは、Ｍ２UA イベントを自律的に Ｍ２UA ピアーに通知するために使用する。

M2UA の NTFY メッセージは、共通メッセージヘッダのみを使用し、以下のパラメタを含む。

- ステータス（必須）
- ASP 識別子
- インタフェース識別子（省略可能）
- 付加情報（省略可能）

インタフェース識別子パラメタは整数型と文字列型がある。整数型インタフェース識別子パラメタを使用する場合の NTFY メッセージフォーマットを図 6-2 に示す。

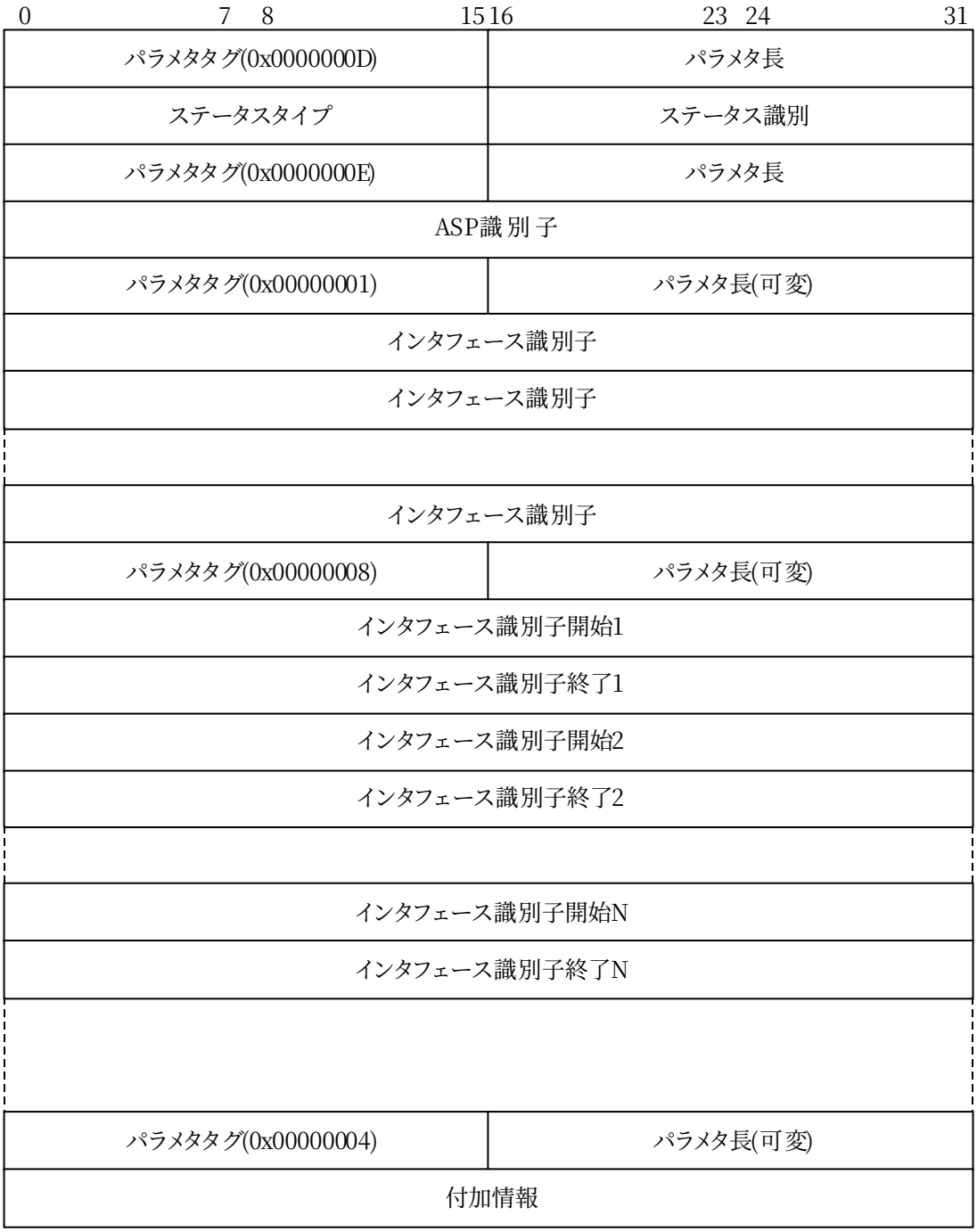


図 6-2 整数型インタフェース識別子使用時の NTFY メッセージフォーマット

続いて、文字列型インタフェース識別子使用時の NTFY **メッセージフォーマットを図** 6-3 **に示す。**

図 6-3 文字列型インタフェース識別子使用時の NTFY メッセージフォーマット

ステータスタイプは、NTFY メッセージの種別を示す 16 ビット符号無し整数であり表 6-3 に示す値をとる。

表 6-3 ステータスタイプ一覧

値	説明
0x1	AS 状態変更
0x2	その他

ステータス種別は 16 ビット符号無し整数であり、ステータスタイプ毎に値を定義する。ステータスタイプが「AS 状態変更」の場合に取り得る値を表 6-4 に示す。

表 6-4 AS 状態変更のステータス識別

値	説明
1	(予約)
2	AS 起動状態
3	AS 稼動状態
4	AS 保留状態

上記通知は、AS 状態変更時に SG から ASP に送られる。
ステータスタイプが「その他」の場合、ステータス識別は表 6-5 に示す値を取る。

表 6-5 その他のステータス識別

値	記述
1	ASP リソース不足
2	代替 ASP 稼動
3	ASP フェイラー

上記通知は、ASP または AS の状態変更に起因しない。ASP リソース不足通知は、ロードシェアモードにおいて、負荷を処理するために ASP 追加が必要であることを不活性 ASP に通知する。代替 ASP 活性化通知は、オーバーライドモードにおいて、代替 ASP が活性状態に遷移し、負荷を引き継いだことを通知する。ASP ファイラーは SG が ASP の非稼動状態を通知する。

6.4.5 ASP 状態管理メッセージクラス

3.4.6 参照。

6.4.6 ASP トラフィック管理メッセージクラス

6.4.6.1 ASPAC メッセージ (ASP ACtive)

ASPAC メッセージは、ASP が稼動状態に遷移することを通知する。
M2UA の ASPAC メッセージは以下のパラメタを含む。

- トラフィックモードタイプ(省略可能)
- インタフェース識別子(省略可能)
- 付加情報(省略可能)

インタフェース識別子パラメタは整数型と文字列型がある。整数型インタフェース識別子パラメタを使用する場合の ASPAC メッセージフォーマットを図 6-4 に示す。

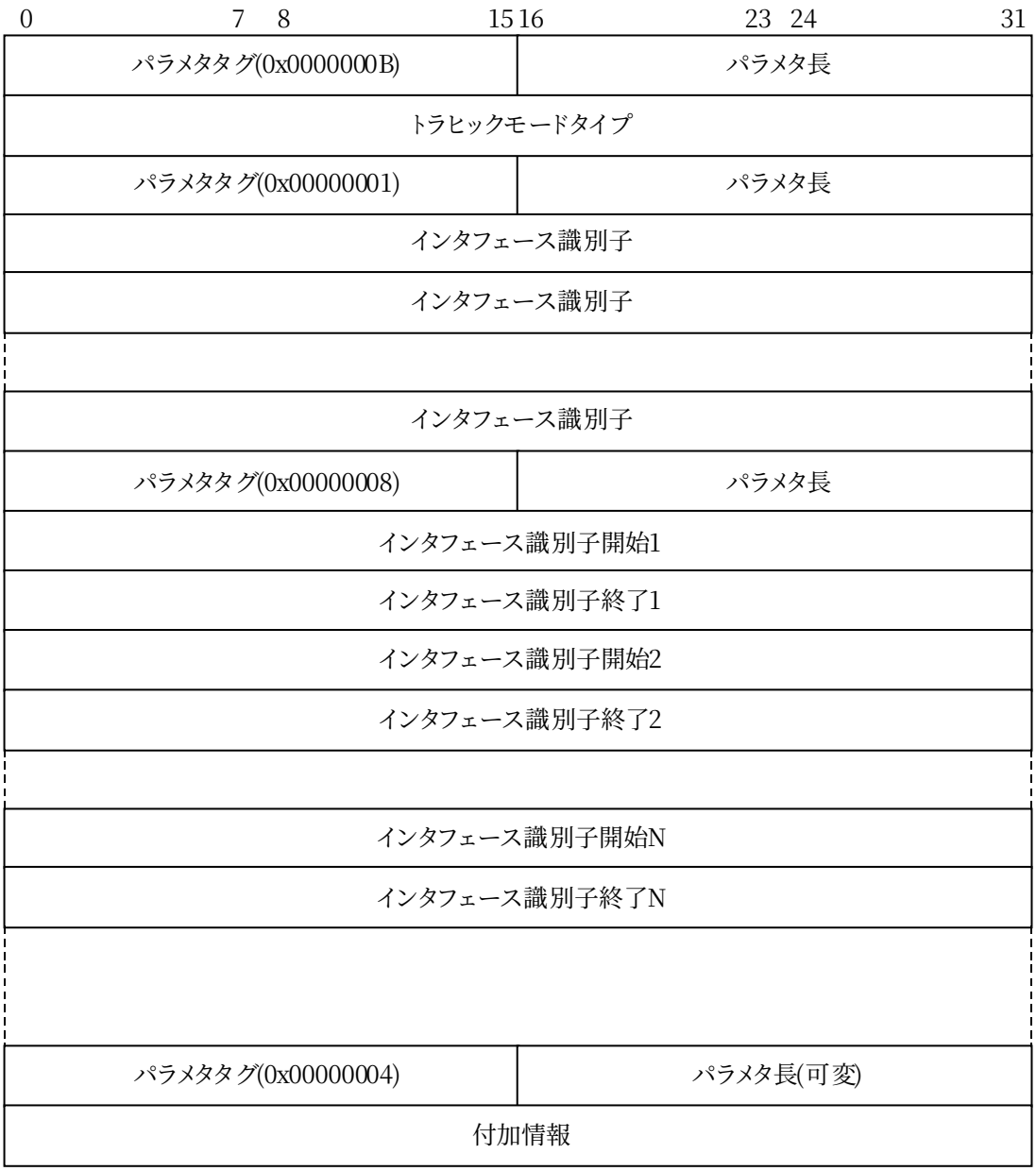


図 6-4 整数型インタフェース識別子使用時の ASPAC メッセージフォーマット

続いて、文字列型インタフェース識別子使用時の ASPAC **メッセージフォーマットを図** 6-5 **に示す。**

図 6-5 文字列型インタフェース識別子使用時の NTFY メッセージフォーマット

6.4.6.1.1 **トラフィックモードタイプ**

トラフィックモードは、AS における ASP の動作を指定する 32 ビット符号無し整数である。取り得る値を表 6-6 に示す

表 6-6 トラフィックモード一覧

値	**説明**
0x01	**オーバーライド**
0x02	**ロードシェア**
0x03	**ブロードキャスト**

オーバーライドは、ASP が AS の全トラフィックを処理することを示し、他に稼動状態 ASP があればトラフィックを引き継ぐ。ロードシェアは、ASP が他の稼動状態 ASP と共に AS のトラフィックを分担することを示す。

ブロードキャストは ASP が他の稼動状態 ASP と同一のメッセージを受信することを示す。

6.4.6.2 ASPIA **メッセージ** (ASP InActive)

ASPIA **メッセージは、ASP が起動状態に遷移することを通知する。**

M2UA **の** ASPIA **メッセージは以下のパラメタを含む。**

- **インタフェース識別子(省略可能)**
- **付加情報(省略可能)**

インタフェース識別子パラメタは整数型と文字列型がある。整数型インタフェース識別子パラメタを使用する場合の ASPAC **メッセージフォーマットを図** 6-6 **に示す。**

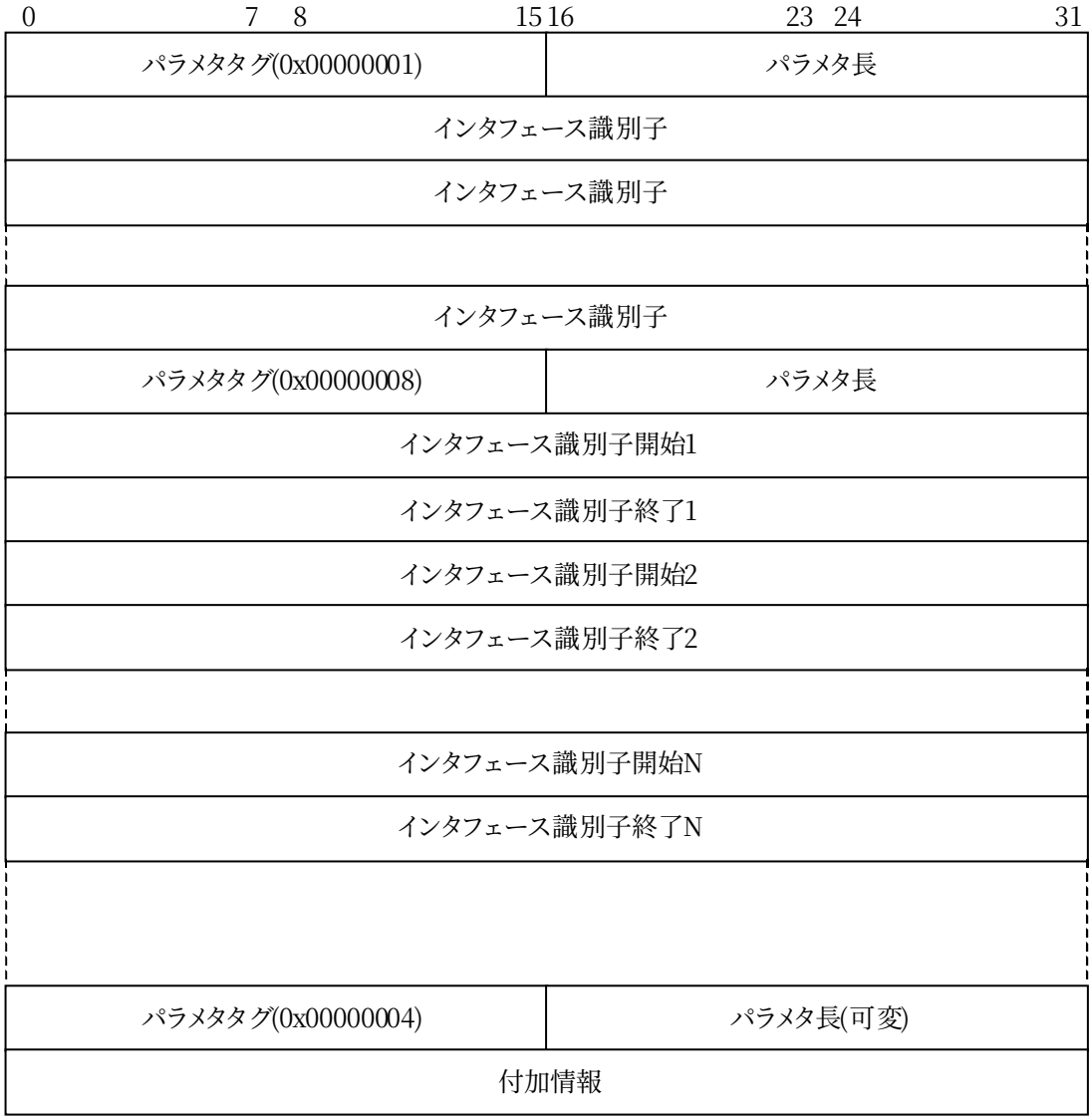


図 6-6 整数型インタフェース識別子使用時の ASPIA **メッセージフォーマット**

続いて、文字列型インタフェース識別子使用時の ASPIA **メッセージフォーマットを図** 6-7 **に示す。**

図 6-7 文字列型インタフェース識別子使用時の ASPIA メッセージフォーマット

6.4.7 MTP2 メッセージクラス

6.4.7.1 データメッセージ

TTC 版 MTP2 は信号長表示の空き 2 ビットを優先度表示として使用する。TTC 標準準拠網との接続時はデータメッセージの代わりに TTC データメッセージ(6.4.7.2)を用いる。

データメッセージは MTP3 プロトコルデータを転送する。データメッセージは以下のパラメタを持つ:

- プロトコルデータ
- 相関ＩＤ

データメッセージのパラメタ部のフォーマットを図 6-8 に示す。

図 6-8 プロトコルデータパラメタのフォーマット

プロトコルデータパラメタにはサービス情報オクテット(SIO : Service Information Octet)と信号情報部(SIF : Signal Information Field)を設定する。

6.4.7.2 TTC データメッセージ

データメッセージは MTP3 プロトコルデータを転送する。データメッセージは以下のパラメタを持つ:

- TTC プロトコルデータ
- 相関 ID

TTC データメッセージのパラメタ部のフォーマットを図 6-9 に示す。

図 6-9 TTC プロトコルデータパラメタのフォーマット

TTC プロトコルデータパラメタには信号長表示(LI ： Length Indicator)とサービス情報オクテット(SIO ： Service Information Octet)と信号情報部(SIF : Signal Information Field)を設定する。

6.4.7.3 データ確認メッセージ

データ確認メッセージはデータメッセージに応答するために送信する。データ確認メッセージは以下のパラメタを持つ:

- 相関 ID

相関 ID パラメタのフォーマットを図 6-10 に示す。

図 6-10 相関 ID パラメタのフォーマット

データ確認メッセージは、データメッセージ、TTC データメッセージに相関 ID パラメタが含まれている場合に送信が必要である。

6.4.7.4 リンク設定要求メッセージ

リンク設定要求メッセージは信号リンク設定を要求するために ASPから SGP に送信する。リンク設定要求メッセージはパラメタを持たない。

6.4.7.5 リンク設定確認メッセージ

リンク設定確認メッセージはリンク設定要求メッセージに応答するために SGP から ASP に送信する。リンク設定確認メッセージはパラメタを持たない。

6.4.7.6 リンク解放要求メッセージ

リンク解放要求メッセージは信号リンク解放を要求するために ASPから SGP に送信する。リンク解放要求メッセージはパラメタを持たない。

6.4.7.7 **リンク解放確認メッセージ**

リンク解放確認メッセージはリンク解放要求メッセージに応答するために SGP **から** ASP **に送信する。リンク解放確認メッセージはパラメタを持たない。**

6.4.7.8 **リンク解放指示メッセージ**

リンク解放指示メッセージは、リンク解放を通知するために SGP **から** ASP **に送信する。リンク解放指示メッセージはパラメタを持たない。**

6.4.7.9 **リンク状態要求メッセージ**

リンク状態要求メッセージは、リンク状態の変更を要求するために ASP **から** SGP **に送信する。リンク状態要求メッセージは以下のパラメタを持つ**:

- **リンク状態**

リンク状態パラメタのフォーマットを図 6-11 **に示す。**

図 6-11 リンク状態変更パラメタのフォーマット

リンク状態フィールドの取り得る値を表 6-7 **に示す。**

表 6-7 リンク状態フィールドの値一覧

値	定義
0x00	ローカルプロセッサアウテージ要求
0x01	ローカルプロセッサアウテージ要求解除
0x02	緊急設定要求
0x03	緊急設定要求解除
0x04	受信、送信および再送信キューのフラッシュまたは解放
0x05	継続または再開
0x06	再送信キューの解放
0x07	リンク状態の監査
0x08	輻輳解除
0x09	輻輳受諾
0xa	輻輳廃棄

6.4.7.10 **リンク状態確認メッセージ**

リンク状態確認メッセージは、リンク状態要求メッセージに応答するため SGP **から** ASP **に送信する。リンク状態確認メッセージは以下のパラメタを持つ**:

- **リンク状態**

リンク状態パラメタのフォーマットを図 6-12 **に示す。**

図 6-12 リンク状態変更結果パラメタのフォーマット

リンク状態フィールドの取り得る値を表 6-8 **に示す。**

表 6-8 リンク状態フィールドの値一覧

値	定義
0x00	ローカルプロセッサアウテージ要求
0x01	ローカルプロセッサアウテージ要求解除
0x02	緊急設定要求
0x03	緊急設定要求解除
0x04	受信、送信および再送信キューのフラッシュまたは解放
0x05	継続または再開
0x06	再送信キューの解放
0x07	リンク状態の監査
0x08	輻輳解除
0x09	輻輳受諾
0xa	輻輳廃棄

6.4.7.11 リンク状態指示メッセージ

リンク状態指示メッセージは、信号リンクの状態変更を通知するため SGP から ASP に送信する。リンク状態指示メッセージは以下のパラメタを持つ:

- イベント

イベントパラメタのフォーマットを図 6-13 に示す。

図 6-13 イベントパラメタのフォーマット

イベントフィールドの取り得る値を表 6-9 に示す。

表 6-9 イベントフィールドの値一覧

値	定義
0x01	遠隔信号局にプロセッサ障害発生
0x02	遠隔信号局のプロセッサ障害解消
0x03	自局にプロセッサ障害発生
0x04	自局にプロセッサ障害解消

6.4.7.12 輻輳指示メッセージ

輻輳通知メッセージは信号リンクの輻輳状態と破棄状態を通知するために SGP から ASP に送信する。有意信号ユニットバッファが輻輳検出レベルを上回った場合、輻輳解除レベルを下回った場合、信号破棄レベルを上回るあるいは下回った場合に、SGP は輻輳指示メッセージを ASP に送信する。SGP は信号破棄レベルまたは輻輳検出レベルのどちらかに変化があった場合のみメッセージを送信する。輻輳指示メッセージは以下のパラメタを持つ:

- 輻輳状態
- 廃棄状態

輻輳指示メッセージのパラメタ部フォーマットを図 6-14 に示す。

図 6-14 輻輳指示メッセージのパラメタ部フォーマット

輻輳状態フィールドおよび廃棄状態フィールドの取り得る値を表 6-10 に示す。

表 6-10 輻輳状態フィールドおよび廃棄状態フィールドの値一覧

値	定義
0x00	輻輳無し
0x01	輻輳レベル 1
0x02	輻輳レベル 2
0x03	輻輳レベル 3

6.4.7.13 回収要求メッセージ

回収要求メッセージは、BSN 取得および有意信号ユニットの回収／破棄を要求するために ASP から SGP に送信する。回収要求メッセージは以下のパラメタを持つ:

- 動作
- シーケンス番号

回収要求メッセージのパラメタ部フォーマットを図 6-15 に示す。

図 6-15 回収要求メッセージのパラメタ部フォーマット

動作パラメタの取り得る値を表 6-11 に示す。

表 6-11 動作パラメタの取り得る値一覧

値	定義
0x01	BSN 取得
0x02	有意信号ユニット回収

動作パラメタが「BSN 取得」の場合、シーケンス番号パラメタは使用されない。動作パラメタが「有意信号ユニット回収」の場合、シーケンス番号パラメタには FSN が含まれる。

6.4.7.14 回収確認メッセージ

回収確認メッセージは、回収要求メッセージに応答するために SGP か ASP に送信する。回収確認メッセ

ージは以下のパラメタを持つ:

- 動作
- 結果
- シーケンス番号

回収確認メッセージのパラメタ部フォーマットを図 6-16 に示す。

図 6-16 回収確認メッセージのパラメタ部フォーマット

動作パラメタは回収要求メッセージの値を設定する。
結果パラメタの取り得る値を表 6-12 に示す。

表 6-12 結果パラメタの取り得る値一覧

値	定義
0x00	動作成功
0x01	動作失敗

動作パラメタが「BSN 取得」で SGP が BSN 取得に成功した場合、SGP はシーケンス番号パラメタに BSN を設定する。結果パラメタには「動作成功」を設定する。BSN 取得に失敗した場合、シーケンス番号パラメタは使用されない。また、結果パラメタには「動作失敗」を設定する。

6.4.7.15 回収指示メッセージ

回収指示メッセージは、回収要求メッセージ(動作=有意信号ユニット回収)で指定される有意信号ユニットを転送するために SGP から ASP に送信する。回収指示メッセージはプロトコルデータパラメタ(図 6-17)または TTC プロトコルデータパラメタ(図 6-18)を持つ。

回収指示メッセージのパラメタ部フォーマットを図 6-17 に示す。

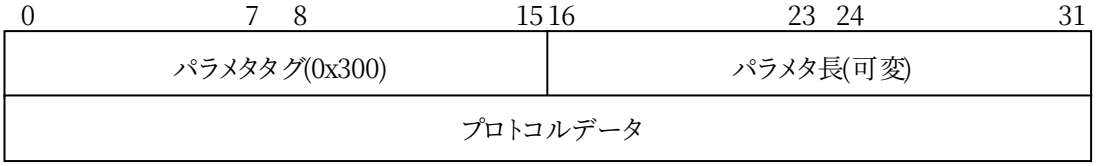


図 6-17 回収指示メッセージのパラメタ部フォーマット

回収指示メッセージ(TTC プロトコルデータ)のパラメタ部フォーマットを図 6-18 に示す。

図 6-18 回収指示メッセージのパラメタ部フォーマット（TTC プロトコルデータ）

6.4.7.16 回収完了通知メッセージ

回収完了メッセージは、回収指示メッセージと同一である。ただし、回収が成功したことを示している。また、送信キューまたは再送キューの末尾の有意信号ユニットを含む場合がある。

6.4.8 インタフェース識別子管理メッセージクラス

インタフェース識別子管理ッセージはオプショナルなメッセージである。信号端末または信号データリンクの自動捕捉に使用される。

6.4.8.1 登録要求メッセージ

登録要求メッセージは、ある一つ以上のリンクキーを遠隔同位 M2UA に登録するために、ASP から SGP に送信する。登録要求メッセージは以下のパラメタを持つ:

- リンクキー

登録要求メッセージのパラメタ部フォーマットを図 6-19 に示す。

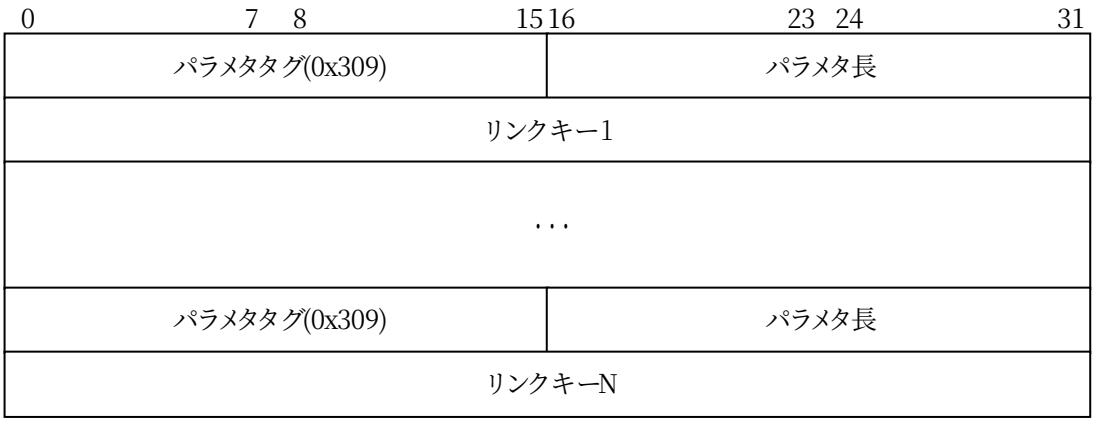


図 6-19 登録要求メッセージのパラメタ部フォーマット

遠隔同位 M2UA にリンクキーが登録されていない場合、送信したリンクキーのエントリが生成され、インタフェース識別子が一意に割り当てられる。複数のリンクキーを一つの登録要求メッセージに設定することにより、一つの登録メッセージで複数のリンクキーの登録が可能である。

リンクキーパラメタのフォーマットを図 6-20 に示す。

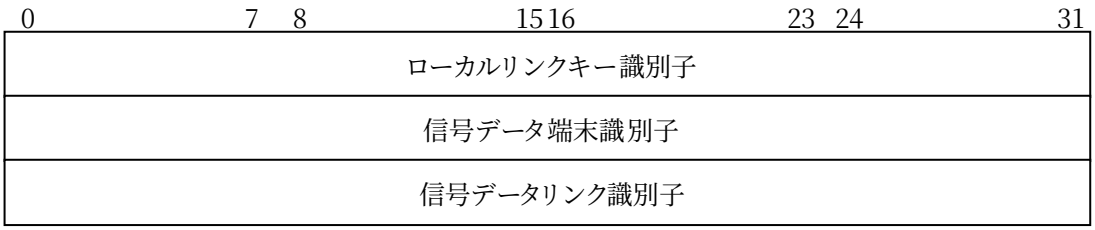


図 6-20 リンクキーパラメタフォーマット

ローカルリンクキー識別子パラメタは ASP が割り当て、登録要求メッセージと登録応答メッセージとの対応付けに使用される。

ローカルリンクキー識別子パラメタのフォーマットを図 6-21 に示す。

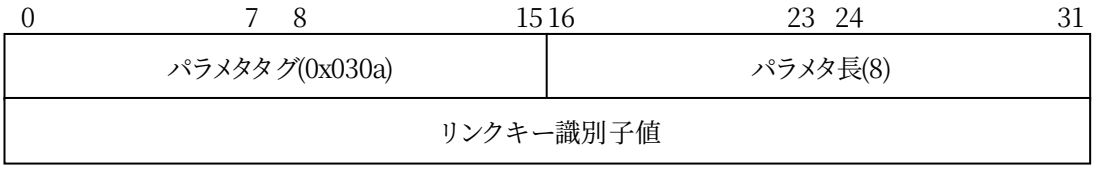


図 6-21 リンクキーパラメタフォーマット

信号データ端末識別子パラメタは、ASP が登録している共通線リンクが関係している信号データ端末を識別する。

信号データ端末識別子パラメタのフォーマットを図 6-22 に示す。

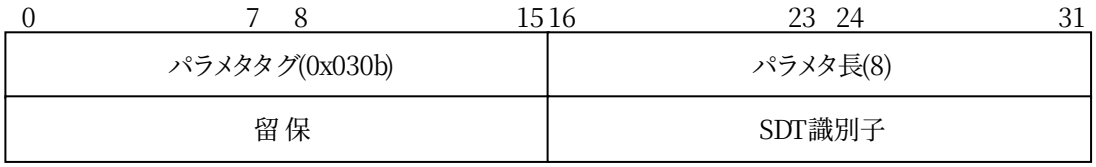


図 6-22 信号データ端末識別子パラメタフォーマット

SDT 識別子パラメタは、32 ビット符号無しの値で、ASP で提供されている MTP レベル 3 の共通線プロトコルによって 12 ビットから 14 ビットが有効である。SDT 識別子の無効なフィールドについてはゼロ保証される。

信号データリンク識別子パラメタは、ASP が登録している共通線リンクが関係している信号データリンクを識別する。

信号データリンク識別子パラメタのフォーマットを図 6-23 に示す。

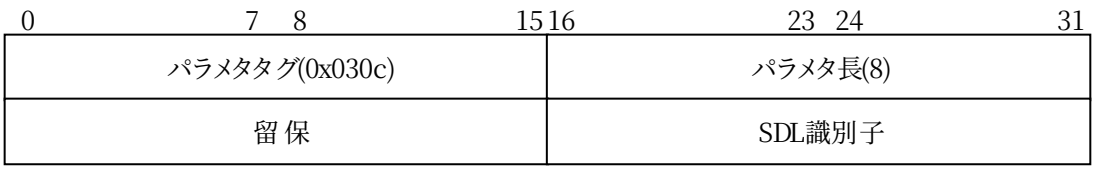


図 6-23 信号データリンク識別子パラメタフォーマット

SDL 識別子パラメタは、32 ビット符号無しの値で、ASP で提供されている MTP レベル 3 の共通線プロトコルによって 12 ビットから 14 ビットが有効である。SDL 識別子の無効なフィールドについてはゼロ保証される。

6.4.8.2 登録応答要求メッセージ

登録応答メッセージは、登録要求メッセージへの応答として、SGP から ASP へ送信される。登録応答メッセージは以下のパラメタを持つ:

- 登録結果

登録応答メッセージのパラメタ部フォーマットを図 6-24 に示す。

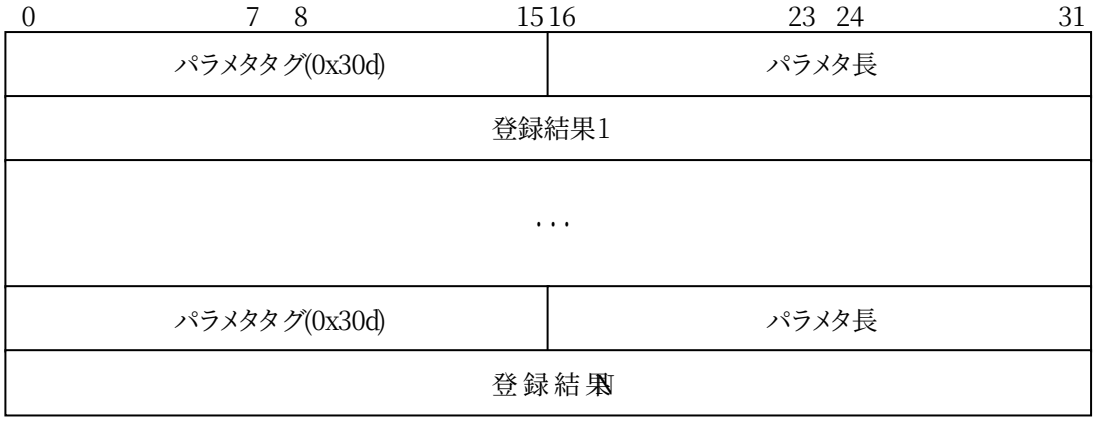


図 6-24 登録応答メッセージのパラメタ部フォーマット

各登録結果パラメタは、登録要求メッセージでの各リンクキーに対応している。

登録結果パラメタのフォーマットを図 6-25 に示す。

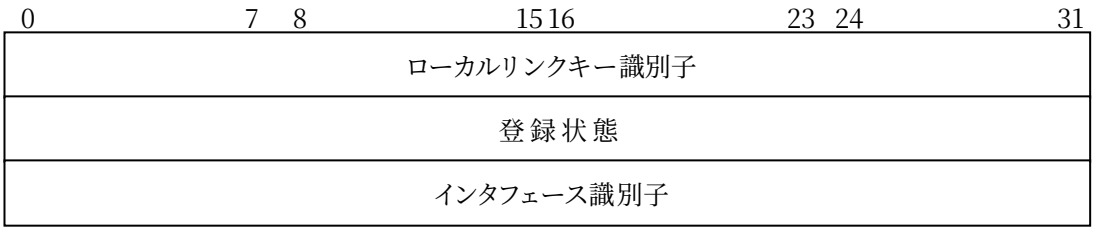


図 6-25 登録結果パラメタフォーマット

ローカルリンクキー識別子パラメタは、登録要求メッセージと同じ値が使用される。

登録状態パラメタの取り得る値を表 6-13 に示す。

表 6-13 登録状態パラメタの取り得る値一覧

値	定義
0x00	登録成功
0x01	エラー（不明）
0x02	エラー（無効な信号データリンク識別子）
0x03	エラー（無効な信号データ端末識別子）
0x04	エラー（無効なリンクキー）
0x05	エラー（非許容）
0x06	エラー（リンクキーの重複）
0x07	エラー（未提供リンクキー）
0x08	エラー（リソース不足）

登録状態パラメタのフォーマットを図 6-26 に示す。

図 6-26 登録状態パラメタフォーマット

6.4.8.3 登録解除要求メッセージ

登録解除要求メッセージは、インタフェース識別子を解放するために ASP から SGP へ送信される。登録解除要求メッセージは以下のパラメタを持つ:

- インタフェース識別子

登録解除要求メッセージのパラメタ部フォーマットを図 6-27 に示す。

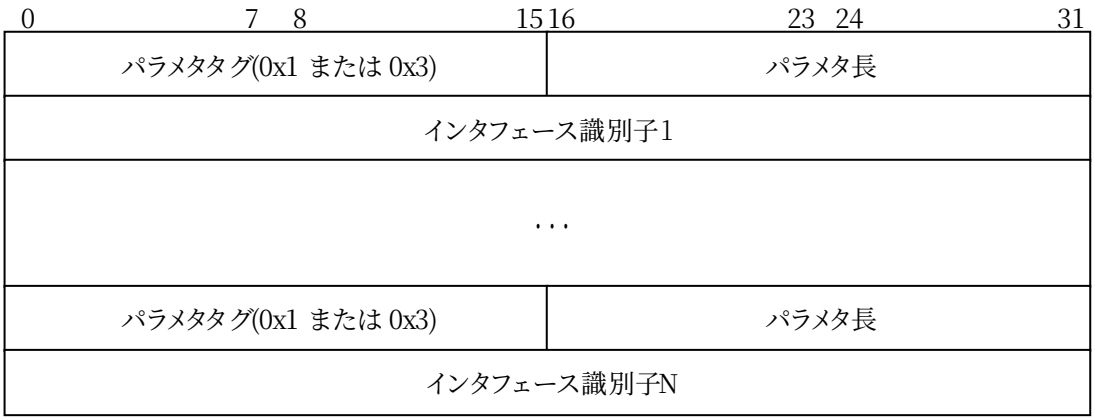


図 6-27 登録解除要求メッセージのパラメタ部フォーマット

6.4.8.4 **登録解除応答メッセージ**

登録解除応答メッセージは、登録解除要求メッセージへの応答として、SGP **から** ASP **へ送信される。登録解除応答メッセージは以下のパラメタを持つ**:

- **登録解除結果**

登録解除応答メッセージのパラメタ部フォーマットを図 6-28 **に示す。**

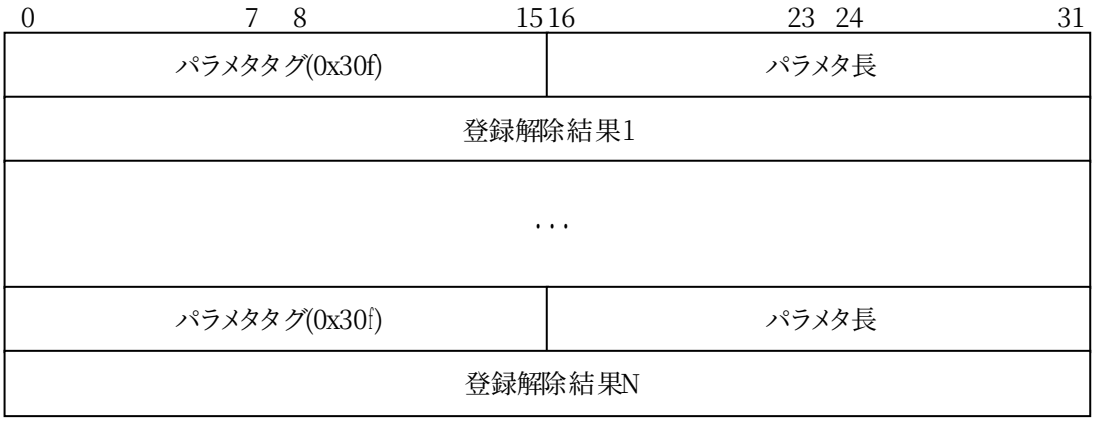


図 6-28 登録解除応答メッセージのパラメタ部フォーマット

登録解除結果パラメタのフォーマットを図 6-29 **に示す。**

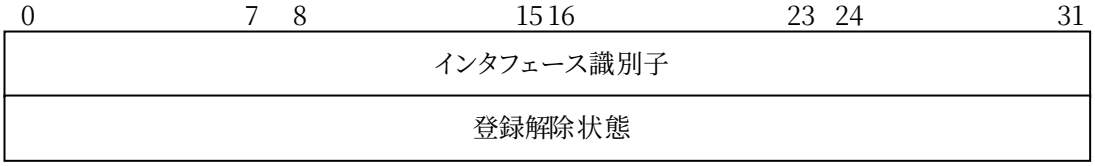


図 6-29 登録解除結果パラメタフォーマット

登録解除状態パラメタの取り得る値を表 6-14 **に示す。**

表 6-14 登録解除状態パラメタの取り得る値一覧

値	定義
0x00	登録解除成功
0x01	エラー（不明）
0x02	エラー（無効なインタフェース識別子）
0x03	エラー（非許容）
0x04	エラー（未登録）

登録解除状態パラメタのフォーマットを図 6-30 に示す。

図 6-30 登録解除状態パラメタフォーマット

6.5 手順

6.5.1 SCTP 管理サービス手順

3.5.1 参照。

6.5.2 メッセージ転送手順

3.5.2 参照。

6.5.3 UA 管理サービス手順

3.5.3 参照。

6.5.4 ASP 管理サービス手順

3.5.4 参照。

6.5.5 AS 管理サービス手順

3.5.5 参照。

6.5.6 MTP2 サービス手順

6.5.6.1 リンクキー管理手順

リンクキー管理手順は以下の手順から構成される:

- 登録
- 登録解除

6.5.6.1.1 登録手順

登録手順を以下に示す:

ASP は登録要求メッセージによってインタフェース識別子を SGP へ送信する。

SGP は受信したリンクキー・パラメタを、その時点で提供されているインタフェース識別子と比較する。もし、受信したリンクキー・パラメタが存在している SGP リンクキーエントリーに一致し、且つ、ASP が AS 一覧に含まれていないのであれば、SGP は ASP を AS 一覧に追加する。もし、受信したリンクキー・パラメタが有効かつ固有な値で、その時点で提供されているリンクキーデータに含まれていないのであれば、ダイナミックなコンフィグレーションを提供している SGP は新規インタフェース識別子を生成し、ASP を新しい AS に追加する。どちらの場合も、SGP は ASP へ登録要求応答メッセージを送信する。

6.5.6.1.2 登録解除手順

登録解除手順を以下に示す:

ASP は登録解除要求メッセージによって個々のインタフェース識別子の登録解除を要求する。

SGP は受信したインタフェース識別子を元に、ASP が AS に含まれているかを検証する。もし含まれていれば、ASP は AS から登録解除される。

登録解除手順は、リンクキーの削除、あるいは AS に関する SGP 内のコンフィグレーションデータを削除するものではない。リンクキーデータは削除されない。

6.6 通信シーケンス

6.6.1 初期化シーケンス

3.6.1 参照。

6.6.2 フェイルオーバシーケンス

3.6.2 参照。

6.6.3 MTP2 シーケンス

6.6.3.1 リンク設定のシーケンス例

リンク設定シーケンスの例を図 6-31 に示す。

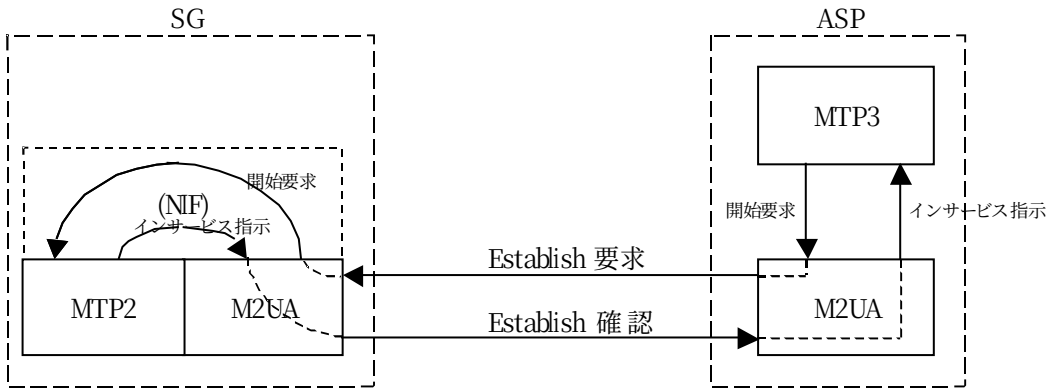


図 6-31 リンク設定のシーケンス例

緊急リンク設定シーケンスの例を図 6-32 に示す。TTC 版 MTP2 は緊急リンク設定手順をサポートしないため、TTC 標準準拠網との接続時には本シーケンスを適用しない。

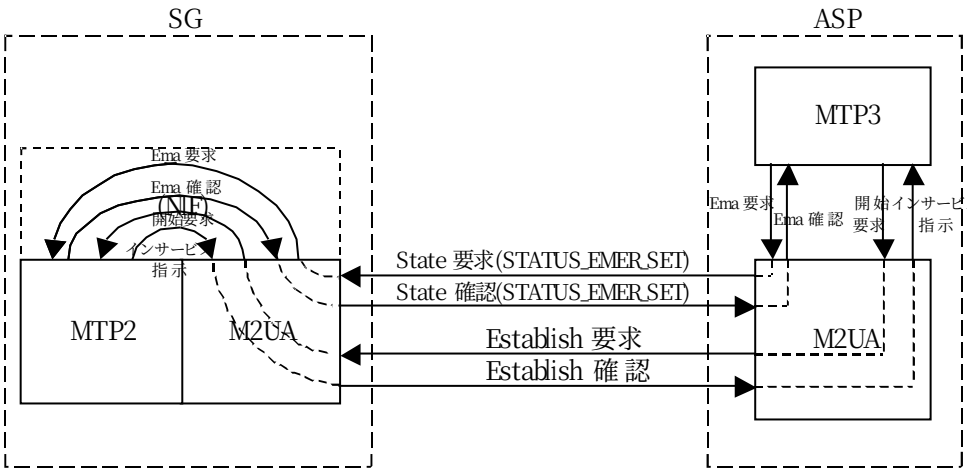
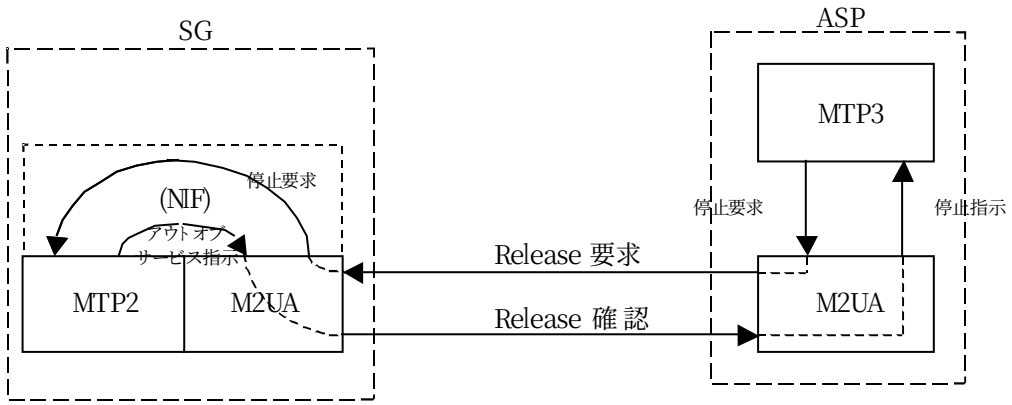


図 6-32 緊急リンク設定シーケンス

6.6.3.2 リンク解放のシーケンス例

リンク解放シーケンスの例を図 6-33 に示す。



NIF:Nordal Interwroking Function

図 6-33 リンク解放シーケンスの例

リンク解放を SG から ASP に通知するシーケンス例を図 6-34 に示す。

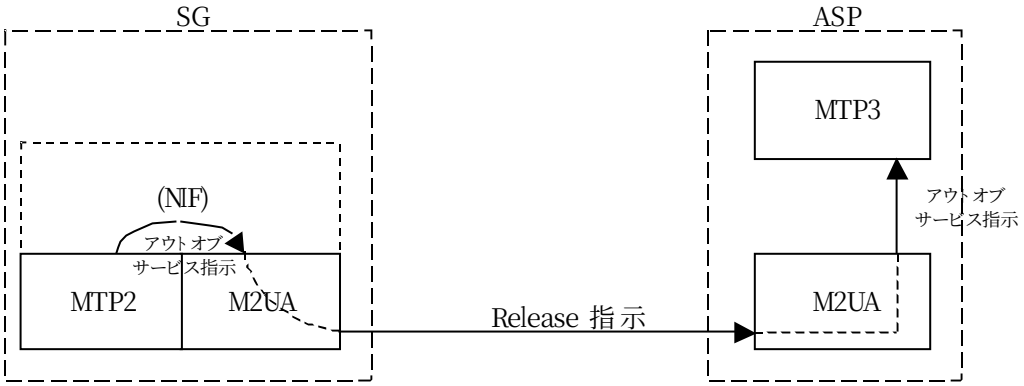


図 6-34 リンク解放通知シーケンスの例

6.6.3.3 **プロセッサ障害のシーケンス例**

TTC 版 MTP2 はプロセッサ手順をサポートしないため、TTC 標準準拠網との接続時には本節に示すシーケンスを適用しない。

自局プロセッサ障害の登録シーケンスを図 6-35 に示す。

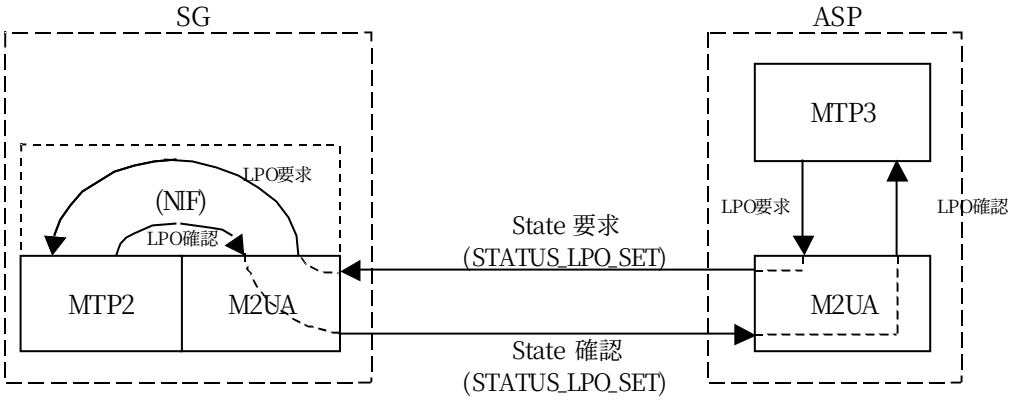


図 6-35 自局プロセッサ障害の登録シーケンス例

自局プロセッサ障害の解除シーケンス例を図 6-36 に示す。

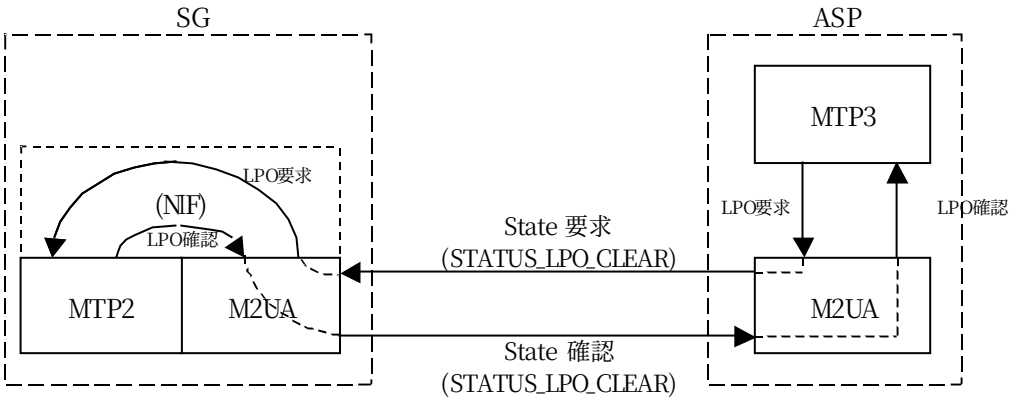


図 6-36 自局プロセッサ障害の解除シーケンス例

遠隔局のプロセッサ障害通知シーケンス例を図 6-37 **に示す。**

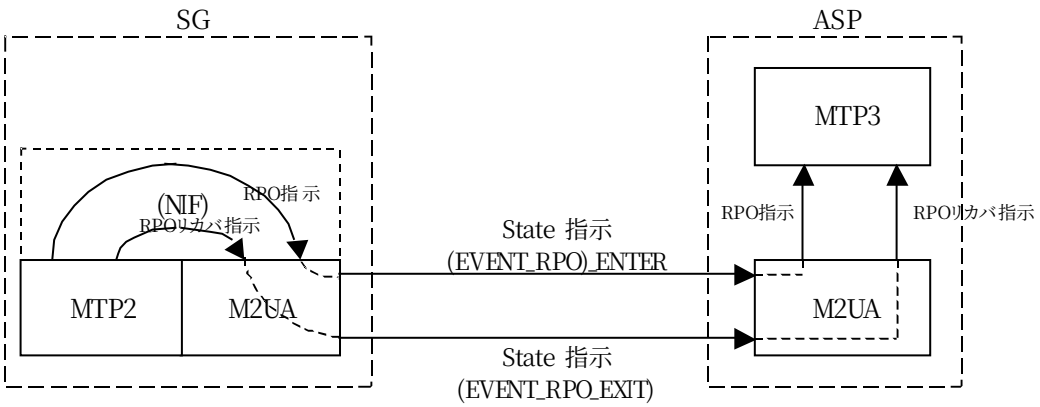


図 6-37 遠隔局のプロセッサ障害通知シーケンス例

6.6.3.4 **リンク輻輳通知のシーケンス例**

TTC **版** MTP2 **はリンク輻輳手順をサポートしないため、**TTC **標準準拠網との接続時には本節に示すシーケンスを適用しない。**

リンク輻輳通知のシーケンス例を図 6-38 **に示す。**

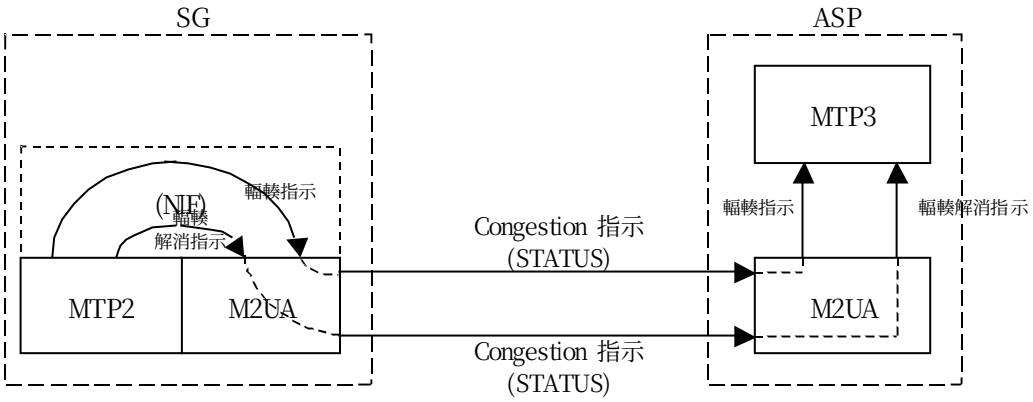
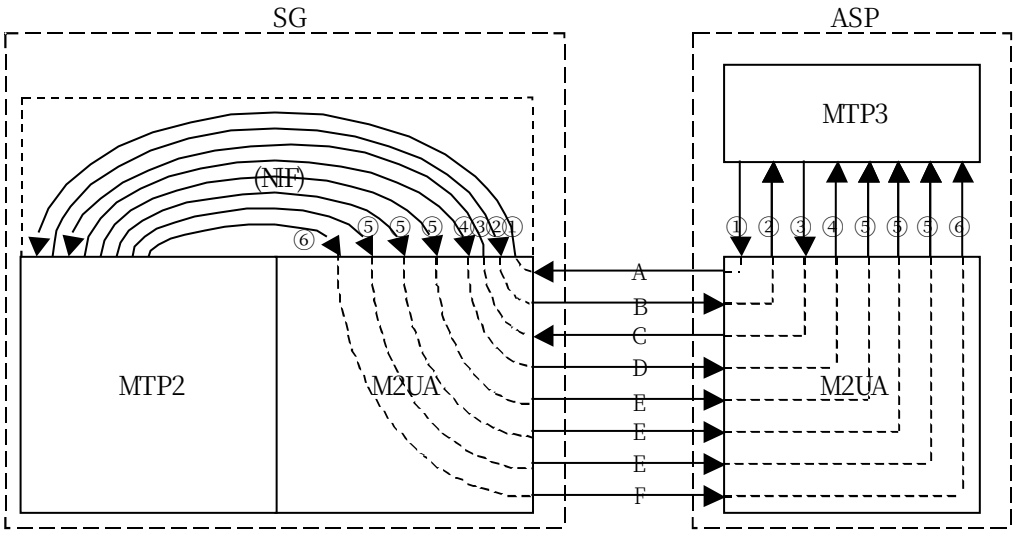


図 6-38 リンク輻輳通知のシーケンス例

6.6.3.5 **リンク切替のシーケンス例**

リンク切替のシーケンス例を図 6-39 **に示す。**



A:Retrieval 要求 (ACTION_RTRV_BSN, seq_num = 0)
B:Retrieval 確認 (ACTION_RTRV_BSN, seq_num = BSN)
C:Retrieval 要求 (ACTION_RTRV_MSG, seq_num = FSN)
D:Retrieval 確認 (ACTION_RTRV_MSG, seq_num = 0)
E:Retrieval 指示
F:Retrieval Complete 指示

①:Retrieval BSN 要求
②:Retrieval BSN 確認
③:Retrieval MSG 要求
④:Retrieval MSG 確認
⑤:Retrieval MSG 指示
⑥:Retrieval Complete 指示

図 6-39 リンク切替のシーケンス例

BSN **回収エラーのシーケンス例を図** 6-40 **に示す。**

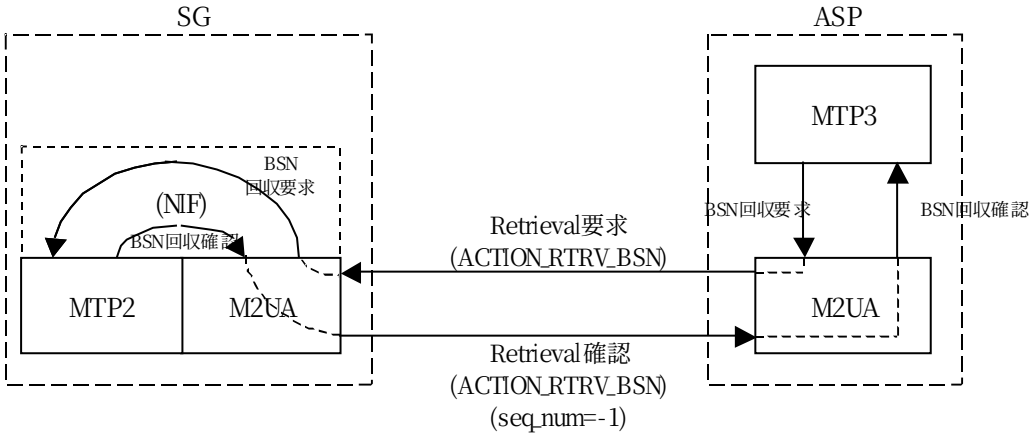


図 6-40 BSN 回収エラーのシーケンス例

メッセージ回収エラーのシーケンス例を図 6-41 **に示す。**

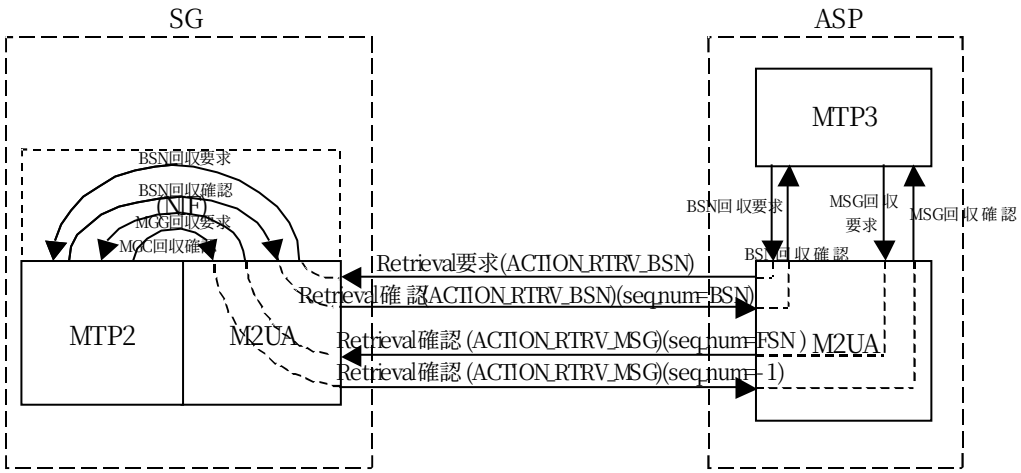


図 6-41 メッセージ回収エラーのシーケンス例

再送バッファクリアのシーケンス例を図 6-42 **に示す。**

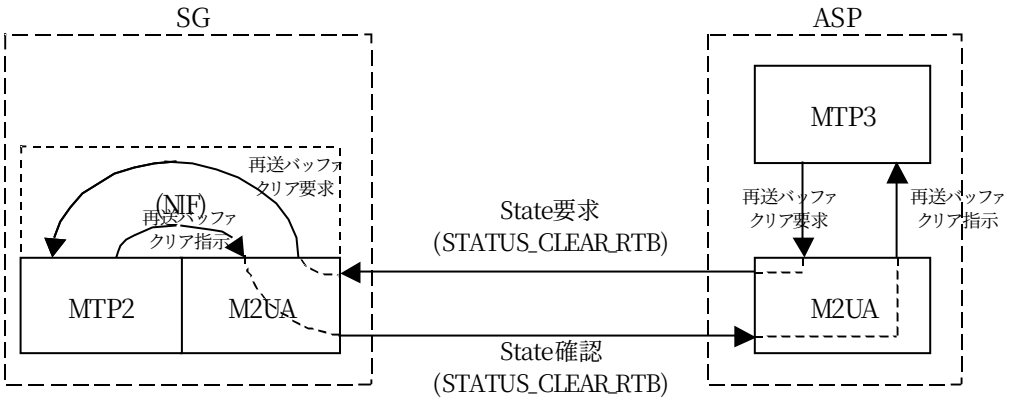


図 6-42 再送バッファクリアのシーケンス例

6.6.3.6 **バッファフラッシュと継続のシーケンス例**

TTC 版 MTP2 はバッファフラッシュおよび継続手順をサポートしないため、TTC 標準準拠網との接続時には本節に示すシーケンスを適用しない。

バッファフラッシュのシーケンス例を図 6-43 **に示す。**

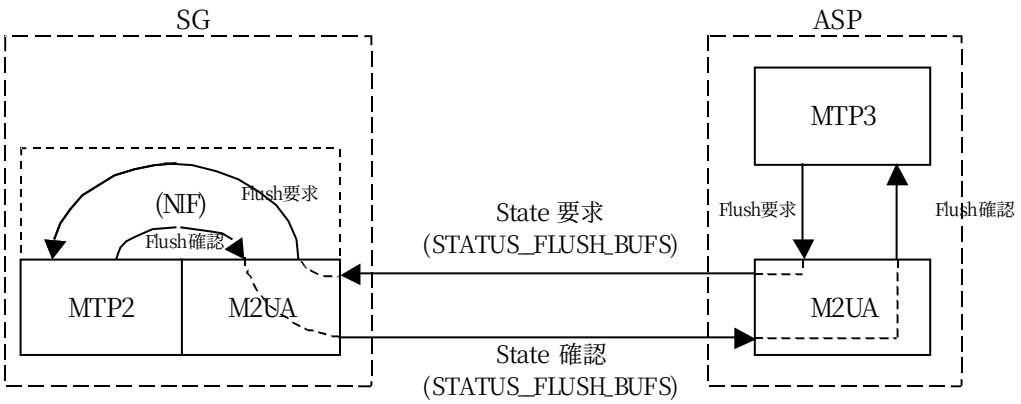


図 6-43 バッファフラッシュのシーケンス例

継続のシーケンス例を図 6-44 **に示す。**

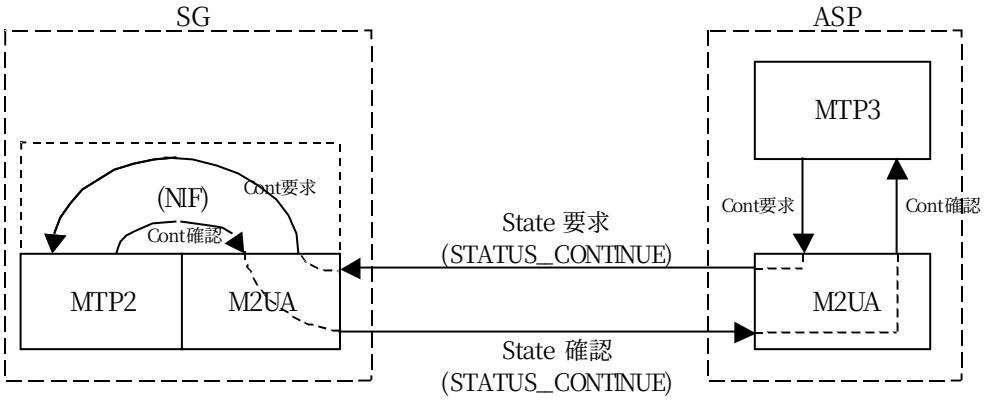


図 6-44 継続のシーケンス例

6.6.3.7 **監査のシーケンス例**

監査のシーケンス例を図 6-45, 6-46, 6-47, 6-48 **に示す**

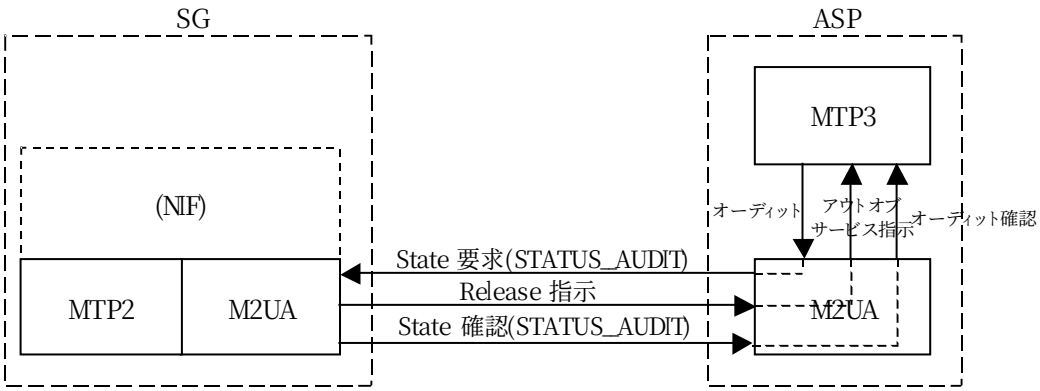


図 6-45 監査のシーケンス例(アウトオブサービス)

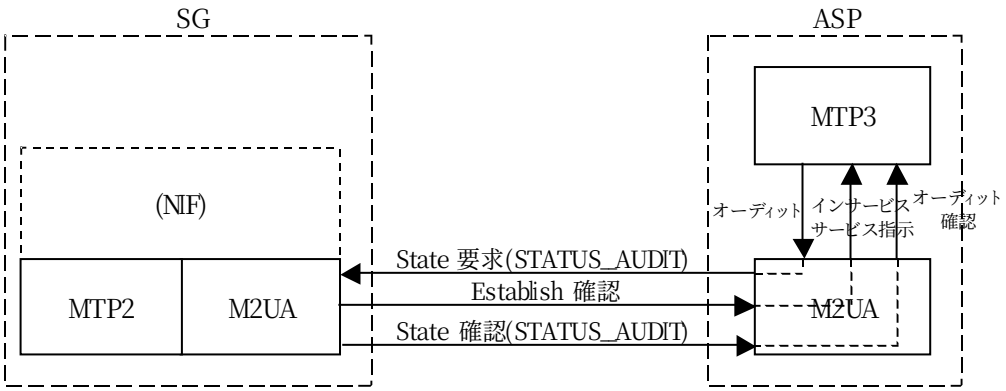


図 6-46 監査のシーケンス例(インサービス)

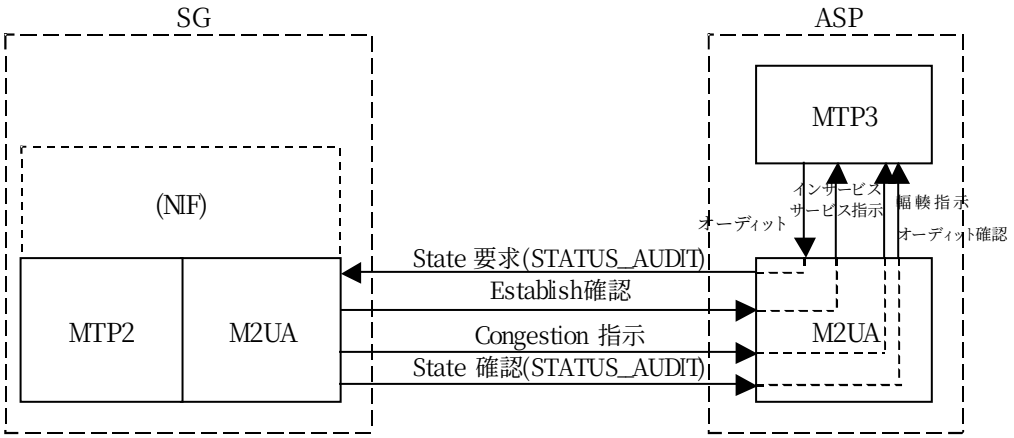


図 6-47 監査のシーケンス例(輻輳)

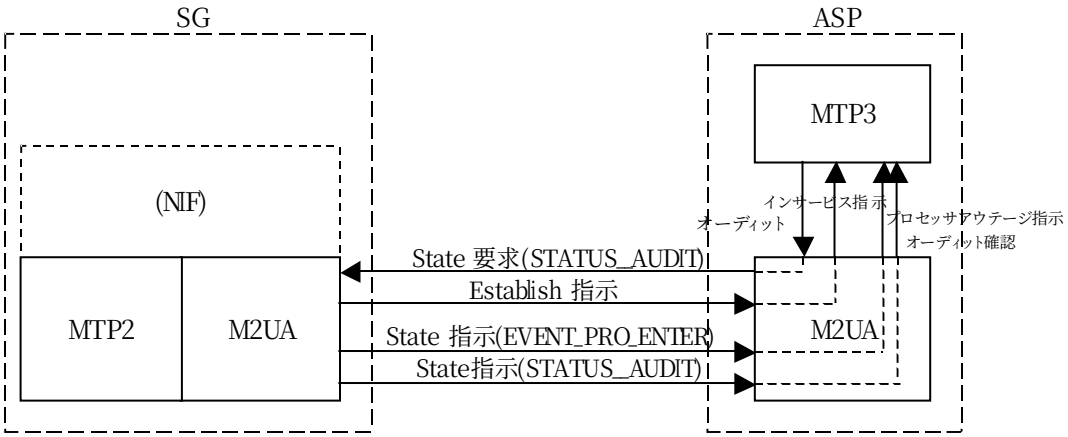


図 6-48 監査のシーケンス例(プロセッサ障害)

6.7 セキュリティ

3.7 参照。

6.8 登録番号

6.8.1 SCTP ペイロードプロトコル識別子

M2UA のペイロードプロトコル識別子は"2"である。

6.8.2 ポート番号

M2UA の登録ポート番号は 2904 である。

6.9 将来の拡張性

3.9 参照。

７．M2PA

7.1 序論

回線交換網上で使用されているシグナリング・プロトコルを IP 網上において SG を介し MGC や IPSP に送り届ける必要がある。例えば、SCN シグナリング・ノードが IP 網上の SS7 シグナリング・リンクを有さないデータベース機器などにアクセスする場合が考えられる。また場合によっては従来のシグナリング・リンクを IP 網に置き換えることにより、運用コスト及びパフォーマンスにおいて有利になることも考えられる。本ドキュメントで説明されている機能は IP 網上における２つの SS7 ノード間において MTP3 ユーザとして十分な MTP3 メッセージ操作や網管理を許容するものである。

この機能は以下の事項を満足する；

- IP ネットワーク上における MTP3 プロトコルのシームレスなオペレーションのサポート
- MTP レベル２／MTP レベル３間のインタフェース領域のサポート
- MTP2 リンクにかわる、SCTP 伝送アソシエーション及びトラフィックの管理のサポート
- 管理部への状態変化の非同期的な報告のサポート

図 7-1 が示すように IP 信号局(IPSP)は上位レイヤを持つことができ、MTP3 のシームレスな相互接続を実現している。また、図 7-2 においてシグナリングゲートウェイ(SG)は SS7 と IP ネットワークの両方への接続を備えた IPSP として SCCP または他の SS7 レイヤを持つことができる。

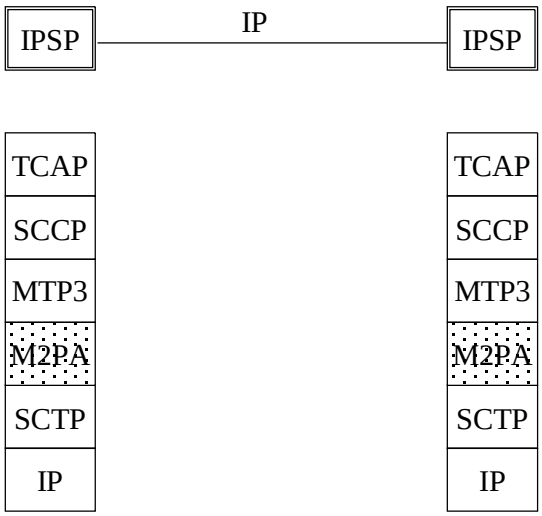
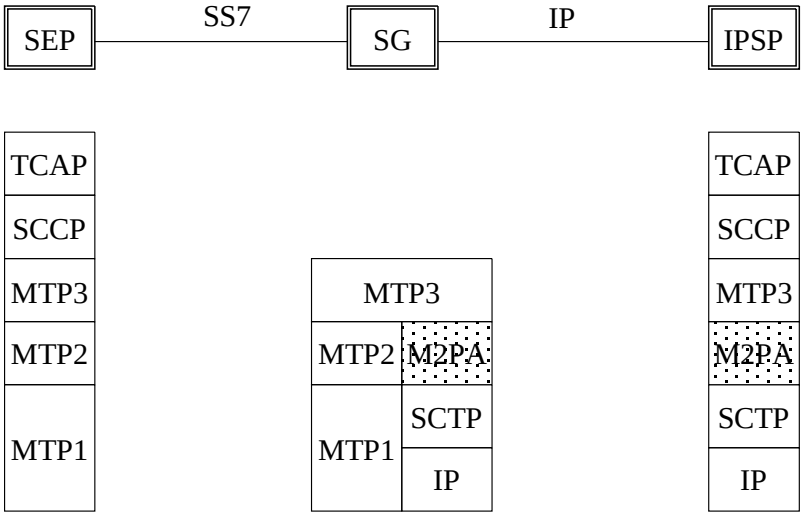


図 7-1 M2PA 対称的同位間アーキテクチャ



SEP－信号端局

図 7-2 IP シグナリングゲートウェイにおける M2PA

7.1.1 M2PA と M2UA の相違について

M2PA と M2UA の共通点と差違について以下の節に示す。

7.1.1.1 M2PA と M2UA の共通点

(1) M2PA、M2UA ともに MTP3 データメッセージを送信する。

(2) M2PA、M2UA ともに MTP3 に対する MTP2 の上位インタフェースを提供する。

7.1.1.2 M2PA と M2UA の差違

(1) M2PA: IPSP が MTP3-MTP2 プリミティブの処理を行う。

M2UA: MGC が SG の MTP2 へ (NIF を介して)MTP3-MTP2 プリミティブを渡す。

(2) M2PA: SG-IPSP 間の接続は SS7 リンクである。

M2UA: SG-MGC 間の接続は SS7 リンクではない。

(3) M2PA: SG はポイントコードを持つ SS7 ノードである。

M2UA: SG は SS7 ノードではなくポイントコードを持たない。

(4) M2PA: SG は SCCP などの上位 SS7 レイヤを持つことができる。

M2UA: SG は MTP3 を持たないため上位 SS7 レイヤを持たない。

(5) M2PA: MTP3 の管理手順に依存する。

M2UA: M2UA の管理手順を使用する。

7.2 用語

7.2.1 BSNT

Backward Sequence Number to be Transmitted.

7.3 サービス

M2PA は以下のサービスを使用する。

- MTP2 サービス

7.3.1 MTP2 サービス

MTP2 サービスのサービスプリミティブの一覧を表 7-1 に示す。

表 7-1 MTP2 サービスのサービスプリミティブ一覧

プリミティブ名		概要
データ要求	要求	データ転送に使用する。
データ指示	指示	
開始	要求	リンク確立に使用する。
停止	要求	リンク解放に使用する。
BSNT 回収	要求	BSNT の問い合わせに使用する。
受信メッセージ BSNT	確認	
BSNT Not Retrievable Confirm	確認	
回収要求と FSNC	要求	未応答および未送信メッセージの回収に使用する。
回収メッセージ	指示	
回収完了	指示	
Flush Buffers	要求	送受信バッファを空にするために使用する。
継続	要求	プロセッサ障害回復後の再開を要求するために使用する。
Emergency	要求	緊急設定要求
Emergency Ceases	要求	緊急設定要求解除
輻輳	指示	輻輳状態の変化を通知する。
インサービス	指示	リンクが In Service 状態であることを通知する。
アウトオブサービス	指示	リンクが Out of Service 状態であることを通知する。
Remote Processor Outage	指示	相手局のプロセッサ障害発生を通知する。
Remote Processor Recovered	指示	相手局のプロセッサ障害回復を通知する。

7.4 メッセージ

M2PA は以下のメッセージを使用する。

- MTP2 メッセージクラス

MTP2 メッセージクラスのメッセージは、共通メッセージヘッダとメッセージデータから構成される。

7.4.1 共通メッセージヘッダ

3.4.1 参照。

7.4.1.1 版数

版数フィールドは版数を示す 8 ビット符号無し整数であり、値は以下の通りである。

最新版数	値
M2PA プロトコル リリース 1.0	0x01

7.4.1.2 予約

将来の拡張用として 8 ビットを予約する。送信側は全てのビットを 0 に設定する。受信側は無視する。

7.4.1.3 メッセージクラス

メッセージクラスはメッセージを分類する 8 ビット符号無し整数である。 M2PA は以下に示すメッセージクラスのみ使用する。

クラス	値(Dec)
M2PA メッセージ	11

7.4.1.4 メッセージタイプ

メッセージタイプはメッセージの種別を特定するために用いられ、8 ビット符号無し整数で表現する。メッセージタイプの取り得る値を表 7-2 に示す。

表 7-2 メッセージタイプ一覧

タイプ	値(Dec)
ユーザデータ	1
リンク状態表示	2
Proving Data	3

7.4.1.5 メッセージ長

メッセージ長はヘッダを含むメッセージの長さを 32 ビット符号無し整数のオクテットで定義する。4 バイト境界に整列しない場合もパディングは行わない。

7.4.2 個別メッセージヘッダ

M2PA は個別メッセージヘッダを使用しない。

7.4.3 パラメタ

M2PA のパラメタ形式は 3.4.3 に準拠しない。

7.4.4 MTP2 メッセージクラス

7.4.4.1 ユーザデータメッセージ

有意信号ユニットに相当するメッセージであり、メッセージデータは以下のフィールドから構成される。

- **信号長表示** (LI : Length Indicator)
- **サービス情報オクテット** (SIO : Service Information Octet)
- **信号情報部** (SIF : Signal Information Field)

メッセージデータ部のフォーマットを図 7-3 に示す。

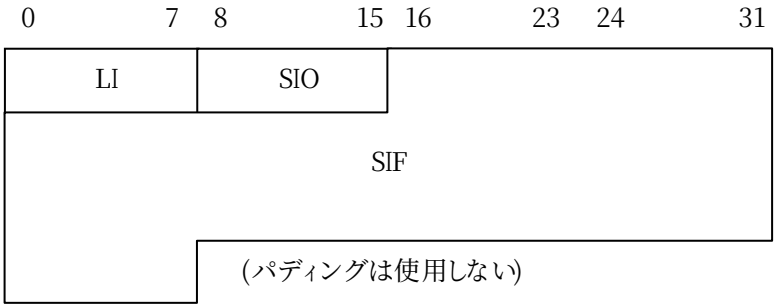


図 7-3 ユーザデータメッセージのメッセージデータ部フォーマット

信号長表示 8 ビットのうち、信号長部分 6 ビットは使用しない。残り 2 ビットを TTC SS7 の優先度表示（PRI : Priority）のために使用する[JT-Q704]。

図 7-4 信号長表示フィールドのフォーマット

7.4.4.2 リンク状態表示メッセージ

リンク状態表示メッセージはリンク状態信号ユニットに相当し、リンク状態の通知に使用する。リンク状態表示メッセージは 0 番のストリームを使用して転送する。リンク状態表示メッセージのメッセージデータ部は以下のフィールドから構成される。

- **リンク状態**

リンク状態表示メッセージのメッセージデータ部フォーマットを図 7-5 に示す。

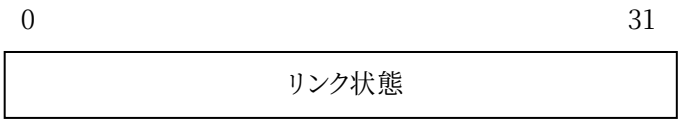


図 7-5 リンク状態表示メッセージのメッセージデータ部フォーマット

リンク状態フィールドの取り得る値を表 7-3 に示す。

表 7-3 リンク状態一覧

状態表示名	値	概要
Alignment	1	SCTP 確立後の M2PA におけるリンクの初期化状態
Proving Normal	2	Emergency Cease 要求を MTP3 より受信した状態
Proving Emergency	3	Emergency 要求を MTP3 より受信した状態
Ready	4	両端の M2PA が Proving を完了した確認状態
Processor Outage	5	上位レイヤにおけるプロセッサ障害の検出した状態
Processor Outage Ended	6	Local Processor Recovered 指示を MTP3 より受信した状態
Busy	7	Receive congestion 開始の指示を SCTP より受信した状態
Busy Ended	8	Receive congestion 終了の指示を SCTP より受信した状態

7.4.4.3 Proving Data

Proving Data メッセージは Proving 状態の時に使用され、SCTP ストリーム管理においてユーザメッセージと同様、1 番のストリームを使用して転送する。そのためデータフィールドのメッセージ長はユーザデータフィールドと同じくらいであることを推奨する。またこのデータフィールドは伝送の正確さを SCTP が確認できるよう、Proving Data メッセージを多様化するためにデータフィールドに幾つかのパターンを持つことを推奨する。

Proving Data メッセージのデータメッセージ部フォーマットを以下に示す。

7.5 手順

7.5.1 MTP2 サービス手順

7.5.1.1 リンク確立手順

リンク確立手順を以下に示す:

起動側 MTP3 は起動側 M2PA に対して開始要求プリミティブを発行する。

起動側 M2PA は開始要求プリミティブを受信すると、起動側 SCTP に対して Associate プリミティブを発行する。

起動側 SCTP は Associate プリミティブを受信すると、受諾側 SCTP へ INIT チャンクを送信する。

受諾側 SCTP は INIT チャンクを受信すると、INIT ACK チャンクを返送する。

起動側 SCTP は INIT ACK チャンクを受信すると、COOKIE ECHO チャンクを送信する。

受諾側 SCTP は COOKIE ECHO チャンクを受信すると、COOKIE ACK チャンクを返送すると同時に、受諾側 M2PA に対して Communication Up プリミティブを発行する。

起動側 SCTP は COOKIE ACK チャンクを受信すると、起動側 M2PA に対して Communication Up プリミティブを発行する。

起動側 M2PA は SCTP から Communication Up プリミティブを受信すると、受諾側 M2PA に対して Link Status Alignment メッセージを送信し、もし受諾側 M2PA から Link Status Alignment メッセージを受信していなければ起動側 M2PA はタイマ T1 を開始する。

起動側 M2PA が受諾側 M2PA より Link Status Alignment メッセージを受信したらタイマ T1 を停止する。

起動側 M2PA はタイマ T2 を起動し、受諾側 M2PA に対して Status_Interval プロトコルパラメタで定義された間隔で Link Status Proving メッセージを送信する。 このとき起動側 M2PA は MTP3 からの Emergency 及び Emergency Ceases 要求プリミティブによって Proving Normal メッセージまたは Proving Emergency メッセージを送出する。 また起動側 M2PA は Normal または Emergency 状態によってタイマ T2 の値を決定する。 もし起動側 M2PA が受諾側 M2PA より Link Status Proving Emergency メッセージを受信したら、起動側 M2PA はタイマ T2 に Emergency 値に使用する。

タイマ T2 起動後、起動側 M2PA は User Data Stream 上で Proving Data メッセージを送信する。 これらのメッセージは Proving_Data_Rate プロトコルパラメタと同間隔で送信される。

タイマ T2 が満了した時点で起動側 M2PA は GETSRTTREPORT プリミティブによるスムース・ラウンド・トリップ・タイム（SRTT）値、SCTP 再送頻度および SCTP Gap Acknowledgements 受信頻度によってアソシエーションを評価する。

もしアソシエーションの評価が満足するものであれば、起動側 M2PA はタイマ T3 を起動し、Status_Interval による間隔において Link Status Ready メッセージを受諾側 M2PA へ送信する。 これは両端の M2PA において Proving 完了の確認に使われる。

起動側 M2PA は Link Status Proving Complete メッセージまたは User Data メッセージを受諾側 M2PA から受信したらタイマ T3 を停止する。 各 M2PA はそれぞれの MTP3 に対してインサービスを通知し、リンク確立を完了する。

7.5.1.2 ユーザデータメッセージ転送手順

ユーザデータメッセージ転送手順を以下に示す。

送信元 MTP3 は送信元 M2PA に対してデータ要求プリミティブを発行し、メッセージ送信を要求する。

送信元 M2PA はデータ要求プリミティブを受信すると、ユーザデータメッセージを構成し、送信元 SCTP に対して Send プリミティブを発行する。

送信元 SCTP は Send プリミティブを受信すると、送信先 SCTP に Data チャンクを転送する。

送信先 SCTP は Data チャンクを受信すると、送信先 M2PA に対して Data Arrive プリミティブを発行する。

送信先 M2PA は Data Arrive プリミティブを受信すると、Receive プリミティブを発行してユーザデータメッセージを受信し、送信先 MTP3 に対してデータ指示プリミティブを発行する。

7.5.1.3 リンク解放手順

MTP3 によるリンクの停止手順を以下に示す。

MTP3 は M2PA に対して停止要求プリミティブを発行する。

M2PA は停止要求プリミティブを受信すると、SCTP に対して Abort プリミティブを発行する。

SCTP は Abort 要求プリミティブを受信すると、SCTP はアソシエーションを解放する。

7.5.1.4 プロセッサ障害手順

TTC 版 MTP2 はプロセッサ障害手順を非標準とするため、ユーザ部プロトコルとして TTC 版 MTP3 を用いる際は本手順を使用しない。

プロセッサ障害は M2PA より上位にあるレイヤの原因により発生する。その際の手順を以下に示す。

障害検出元 M2PA はローカルプロセッサ障害を検出するとリンク状態表示メッセージ(Processor Outage)を障害通知先 M2PA に送信し、ユーザデータメッセージ送信を中止する。

障害通知先 M2PA はリンク状態表示メッセージ(Processor Outage)を受信すると、MTP3 に Remote Processor Outage 指示プリミティブを発行し、ユーザデータメッセージ送信を中止する。 もし Remote Congestion タイマ T6 が起動していれば、障害通知先 M2PA はそれを停止する。

MTP3 は M2PA に対して Flush Buffers または継続の要求プリミティブを発行する。 障害が回復したとき、障害元の MTP3 は障害検出元 M2PA に対してローカルプロセッサ障害回復指示プリミティブを発行する。 障害検出元 M2PA は障害通知先 M2PA にリンク状態表示メッセージ(Processor Outage Ended)を送信する。

障害通知先 M2PA はリンク状態メッセージ(Processor Outage Ceased)を受信すると、MTP3 に対して Remote Processor Recovered 指示プリミティブを発行する。

7.5.1.5 レベル 2 フロー制御手順

SCTP 自身は既に輻輳制御を持つため、M2PA によるレベル２フロー制御の目的はアソシエーションの監視を行い、アソシエーションを中止するかどうかの判断を行う。その際の手順を以下に示す。

輻輳検出元 M2PA は SCTP の輻輳状態を検出し、輻輳通知先 M2PA にリンク状態表示メッセージ(Busy)を送信する。 これは SCTP が輻輳状態を検出し続けている間、リンク状態表示メッセージ（Busy）を定期的に送信し続ける。

輻輳通知先 M2PA はリンク状態表示メッセージ(Busy)を受信し、タイマ T6 を開始する。

輻輳検出元 M2PA は SCTP の輻輳状態終了を検出すると、輻輳通知先 M2PA にリンク状態表示メッセージ(Busy Ended)を送信する。

輻輳通知先 M2PA はタイマ T6 満了前にリンク状態表示メッセージ(Busy Ended)を受信すると、タイマ T6 を停止する。タイマ T6 が満了すると、輻輳通知先 M2PA は SCTP に対して Abort プリミティブを発行して、アソシエーションを解放する。

7.5.1.6 リンク輻輳通知手順

TTC 版 MTP2 はリンク輻輳通知手順を非標準とするため、ユーザ部プロトコルとして TTC 版 MTP3 を用いる際は本手順を使用しない。

ITU-T Q.704 3.8 節における MTP３信号リンクの転送輻輳の必要条件を満たすため、M2PA は SCTP から適切に通知を受けることを推奨する。M2PA は輻輳指示プリミティブを発行して信号リンク輻輳状態及び信号リンク廃棄状態の変化を MTP3 に通知する。多段階輻輳レベルを有する場合は、各輻輳しきい値を超過する毎に MTP3 に通知する。

7.5.1.7 切替手順

切替手順の目的は、使用不可となった信号リンク上の信号トラフィックをできるだけすみやかに別の信号リンクに移すにあたり、信号紛失、二重受信、信号順序逆転を防止することにある。この目的のため、切替はユーザデータメッセージの回収し、代替信号リンク上で再送する。リンク状態表示メッセージは回収、再

送しない。

M2PA の切替手順は、MTP2 の順方向シーケンス番号(FSN)と逆方向シーケンス番号(BSN)の替わりに 16 ビットの SCTP ストリームシーケンス番号(SSN)を使用する。MTP3 の切替信号(COO)と切替確認信号(COA)は 7 ビットの FSN/BSN を前提とするため、M2PA の切替手順には使用できない。このため、24 ビットの FSN/BSN を前提とする B-ISDN 用 MTP3 の拡張切替信号(XCO)と拡張切替確認信号(XCA)を使用する。SSN は FSN/BSN の下位 16 ビットに設定し、上位 8 ビットは 0 設定する。

ユーザデータメッセージ回収手順を以下に示す:

切替元 MTP3 は切替元 M2PA に対して「BSNT 回収」要求プリミティブを発行し、切替元 BSN を要求する。

切替元 M2PA は「BSNT 回収」要求プリミティブを受信すると、1 番の受信ストリーム上で最初の未確認メッセージを探し、直前のメッセージの SSN を切替元 BSNT とする。1 番の受信ストリーム上の全メッセージが確認済である場合は、最終メッセージの SSN を切替元 BSNT とする。

切替元 M2PA は切替元 MTP3 に対して「受信メッセージ BSNT」確認プリミティブを発行して、切替元 BSNT を通知する。

切替元 MTP3 は「受信メッセージ BSNT」確認プリミティブを受信すると、代替信号リンクを用いて遠隔 MTP3 に対して XCO メッセージを送信し、切替元 BSNT を通知する。

遠隔 MTP3 は XCO メッセージを受信すると、M2PA に対して「BSNT 回収」要求プリミティブを発行し、遠隔 BSNT を取得する。

遠隔 MTP3 は、代替信号リンクを用いて切替元 MTP3 に対して XCA メッセージを送信し、遠隔 BSNT を通知する。

切替元 MTP3 は XCA メッセージを受信すると、遠隔 BSNT を FSNC とし、切替元 M2PA に対して「回収要求と FSNC」要求プリミティブを発行する。

切替元 M2PA は「回収要求と FSNC」要求プリミティブを受信すると、FSNC よりも大きな SSN を持つユーザデータメッセージを SCTP から回収するとともに、SCTP へ送信依頼していないユーザデータメッセージを取得する。

切替元 M2PA は切替元 MTP3 に対して「回収メッセージ」プリミティブを発行し、ユーザデータメッセージを通知する。全てのユーザデータメッセージを通知し終わると「回収完了」プリミティブを発行する。

なお、上記説明では省略したが、遠隔 MTP3 も「回収要求と FSNC」プリミティブを発行してユーザデータメッセージを回収する。

7.5.1.8 M2PA リンク状態

MTP3 プリミティブ要求、SCTP 通知、相手局 M2PA からの状態表示メッセージの受信、タイマの満了などのイベントに応じて M2PA リンクはある状態から別の状態へ遷移する。この状態遷移を図 7-6 に示す。またリンク状態一覧を表 7-4 に示す。

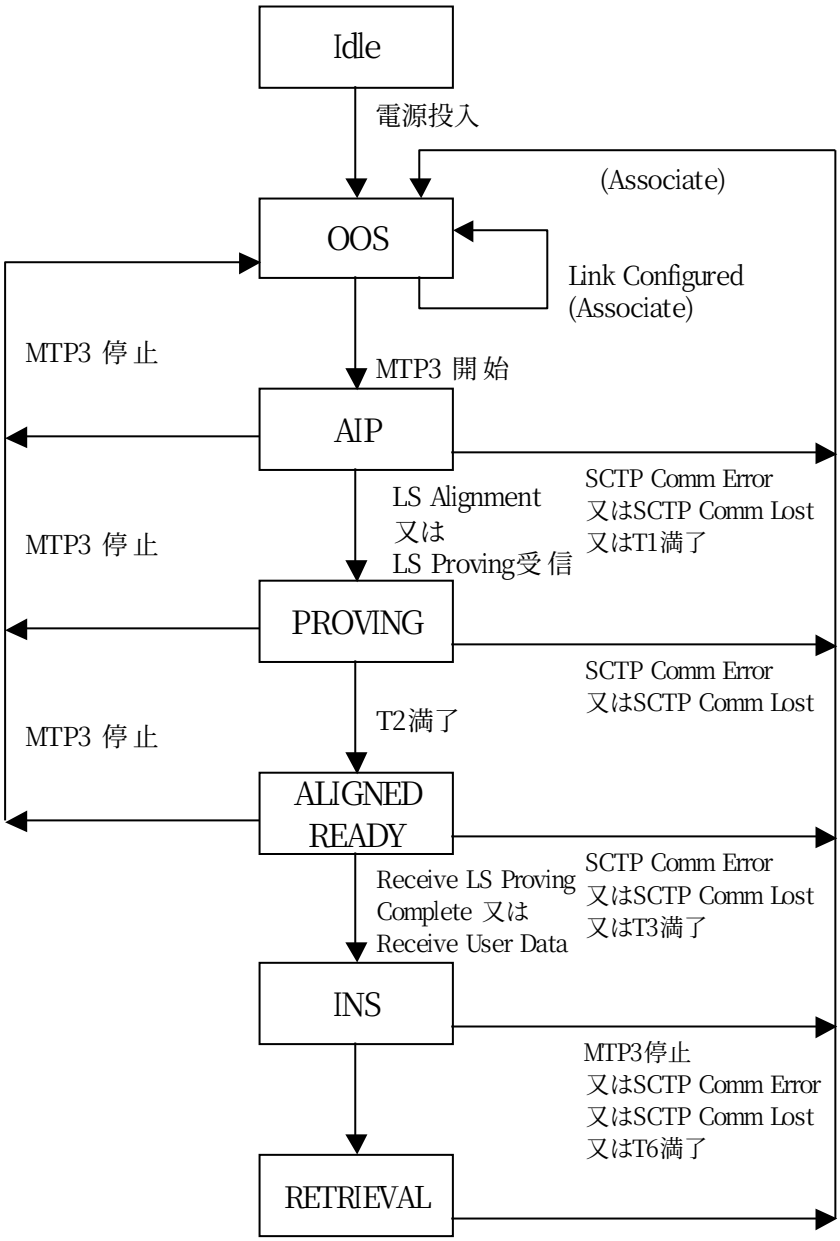
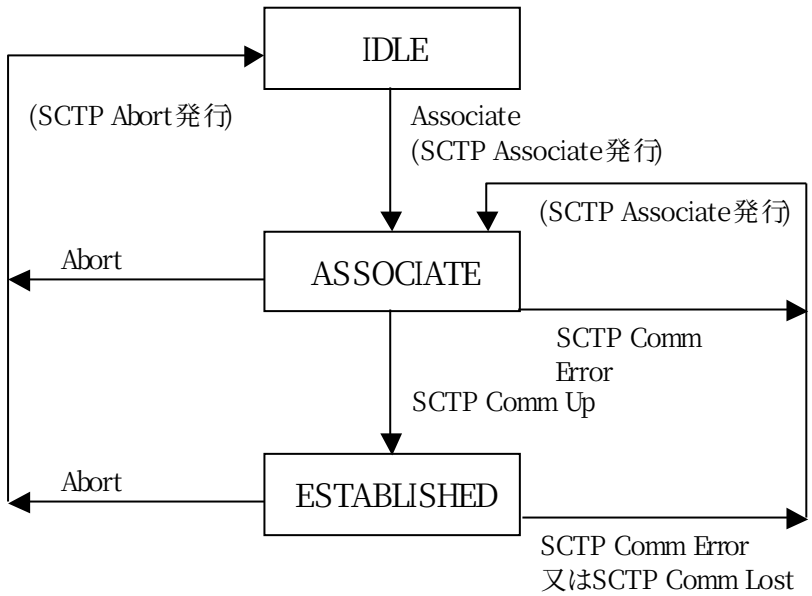


図 7-6 リンク状態遷移図

表 7-4 リンク状態一覧

リンク状態名	概要
Idle	電源投入初期化中のリンク状態
Out of Service	電源投入初期化完了
Alignment in Progress	相手局 M2PA と Alignment メッセージの送受信
Proving	相手局 M2PA へ Proving Data の送信
Aligned Ready	Proving を完了し、相手局 M2PA の Proving 完了待
In Service	リンクはトラフィックに対して準備完了
Retrieval	リンク上のトラフィックは停止、MTP3 からのメッセージ回収要求待

また M2PA アソシエーションの状態遷移図を以下に示す。



7.6 通信シーケンス例

7.6.1 MTP2 シーケンス

7.6.1.1 リンク設定／メッセージ送転送リンク解放のシーケンス例

リンク設定、メッセージ転送、リンク解放のシーケンス例を図 7-7 に示す。

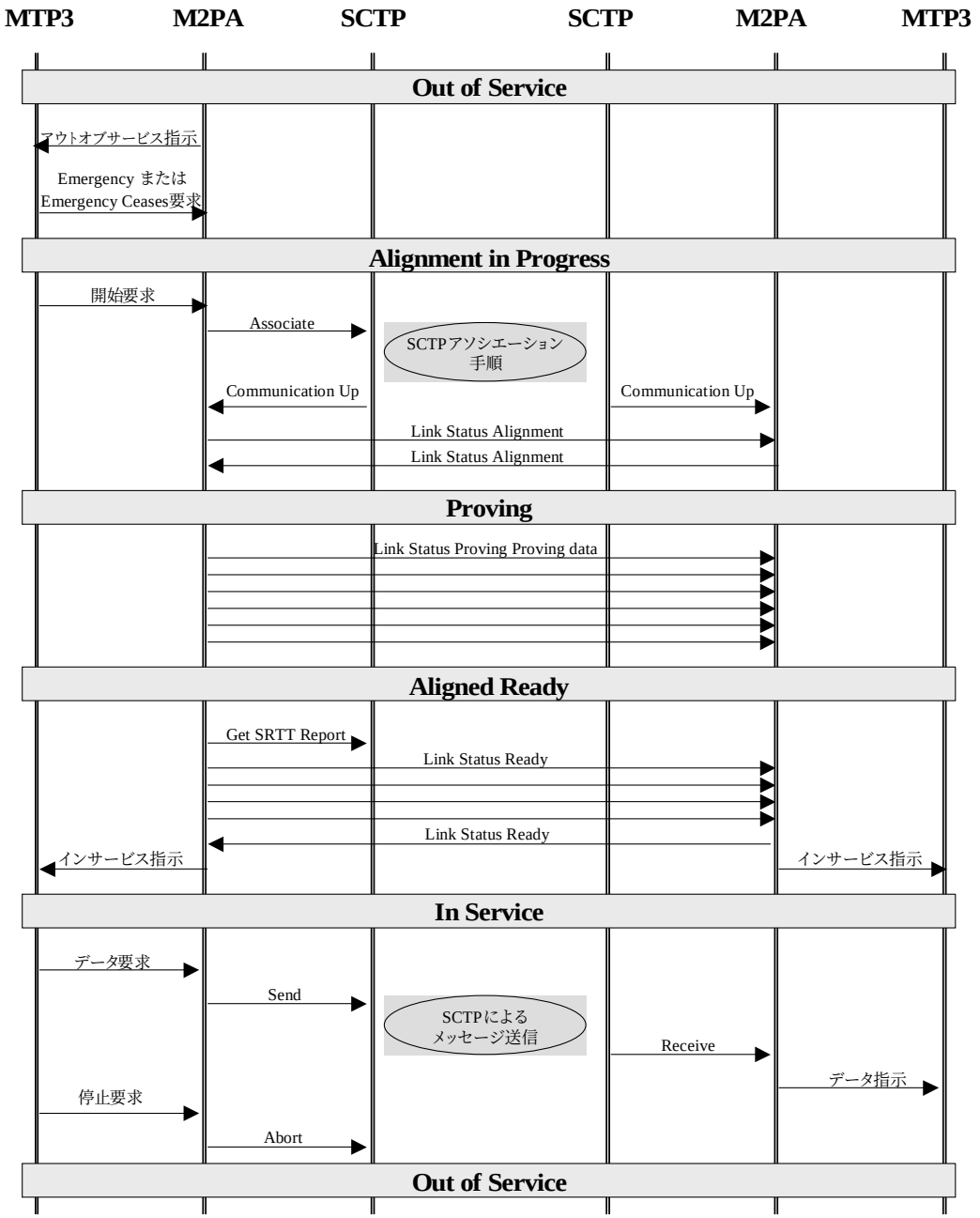


図 7-7 リンク開始／ユーザデータメッセージ転送／リンク停止

7.6.1.2 **プロセッサ障害及びその終了のシーケンス例**

TTC 版 MTP2 **はプロセッサ障害手順を非標準とするため、ユーザ部プロトコルとして** TTC 版 MTP3 **を用いる際は本シーケンスを使用しない。**

プロセッサ障害のシーケンス例を図 7-8 **に示す。**

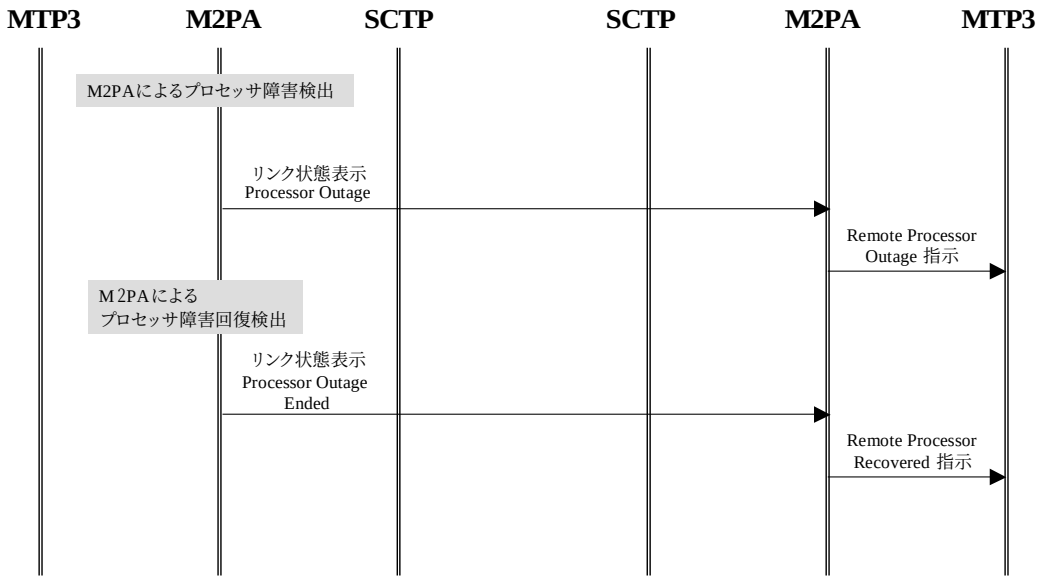


図 7-8 プロセッサ障害及びその終了

7.6.1.3 **レベル 2 フロー制御のシーケンス例**

レベル 2 フロー制御のシーケンス例を図 7-9 および図 7-10 に示す。図 7-9 はタイマ T6 満了前に、輻輳が終息し相手局がリンク状態表示メッセージ(Busy Ended)を受信した場合のシーケンス例となる。図 7-10 はタイマ T6 満了前に輻輳が終息しなかった場合のシーケンス例となる。

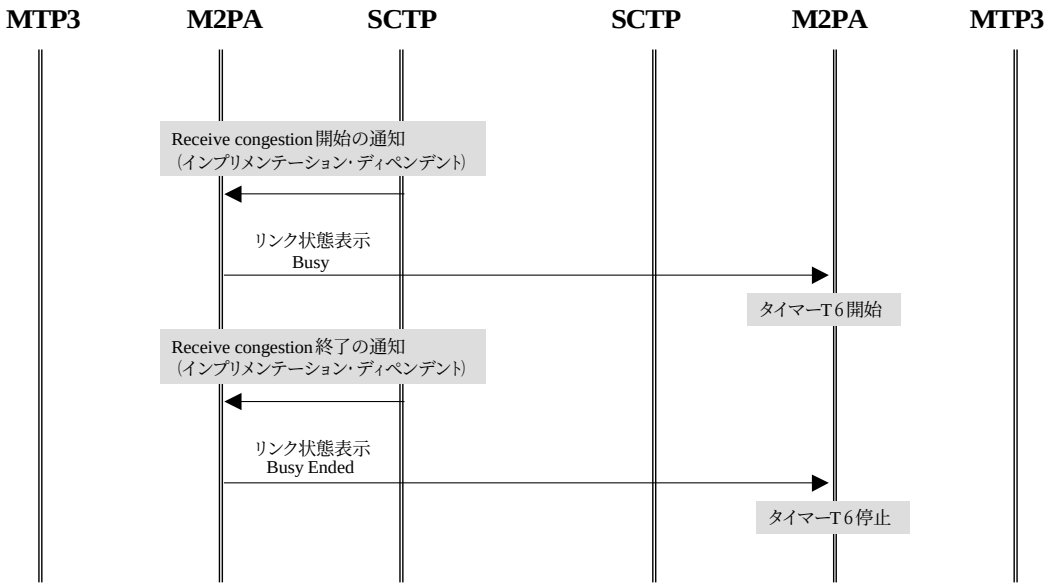


図 7-9 レベル 2 フロー制御（タイマ T6 満了前に輻輳停止）

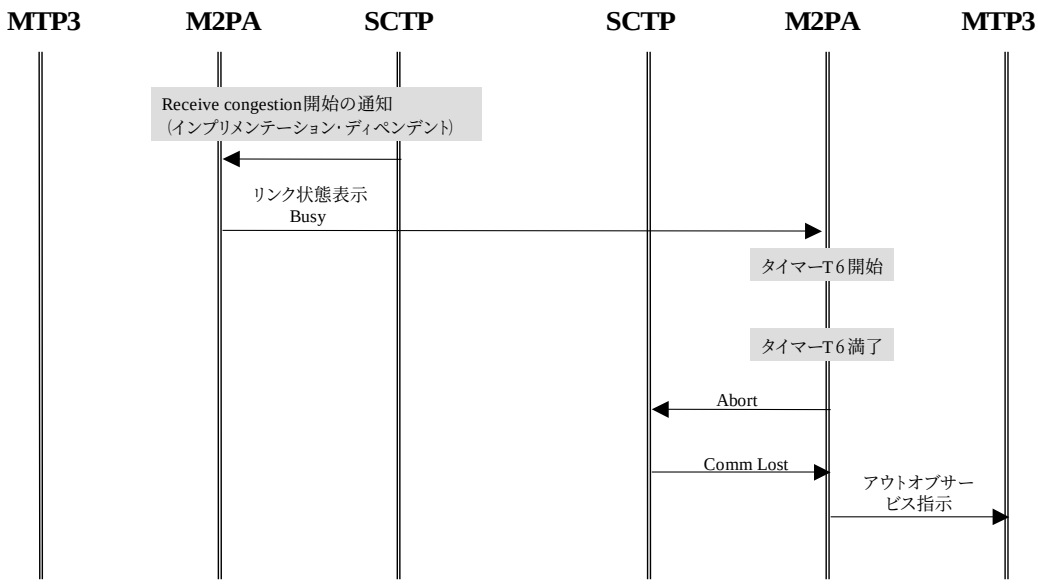


図 7-10 レベル 2 フロー制御（タイマ T6 満了）

7.6.1.4 **リンク輻輳通知のシーケンス例**

TTC 版 MTP2 はリンク輻輳通知手順を非標準とするため、ユーザ部プロトコルとして TTC 版 MTP3 を用いる際は本シーケンスを使用しない。

SCTP が輻輳の突入および解除を M2PA に通知することを前提したシーケンス例を図 7-11 に示す。輻輳レベルが定義されている場合、通知には輻輳レベルも含まれる。

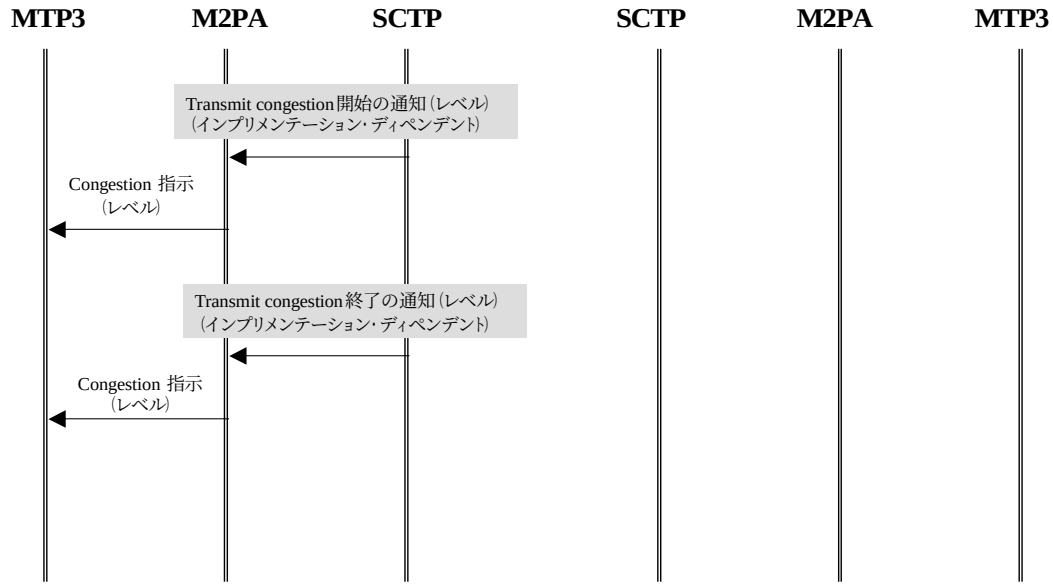


図 7-11 MTP3 信号リンク輻輳（レベル）

7.6.1.5 **切替手順のシーケンス例**

切替手順のシーケンス例を図 7-12 に示す。

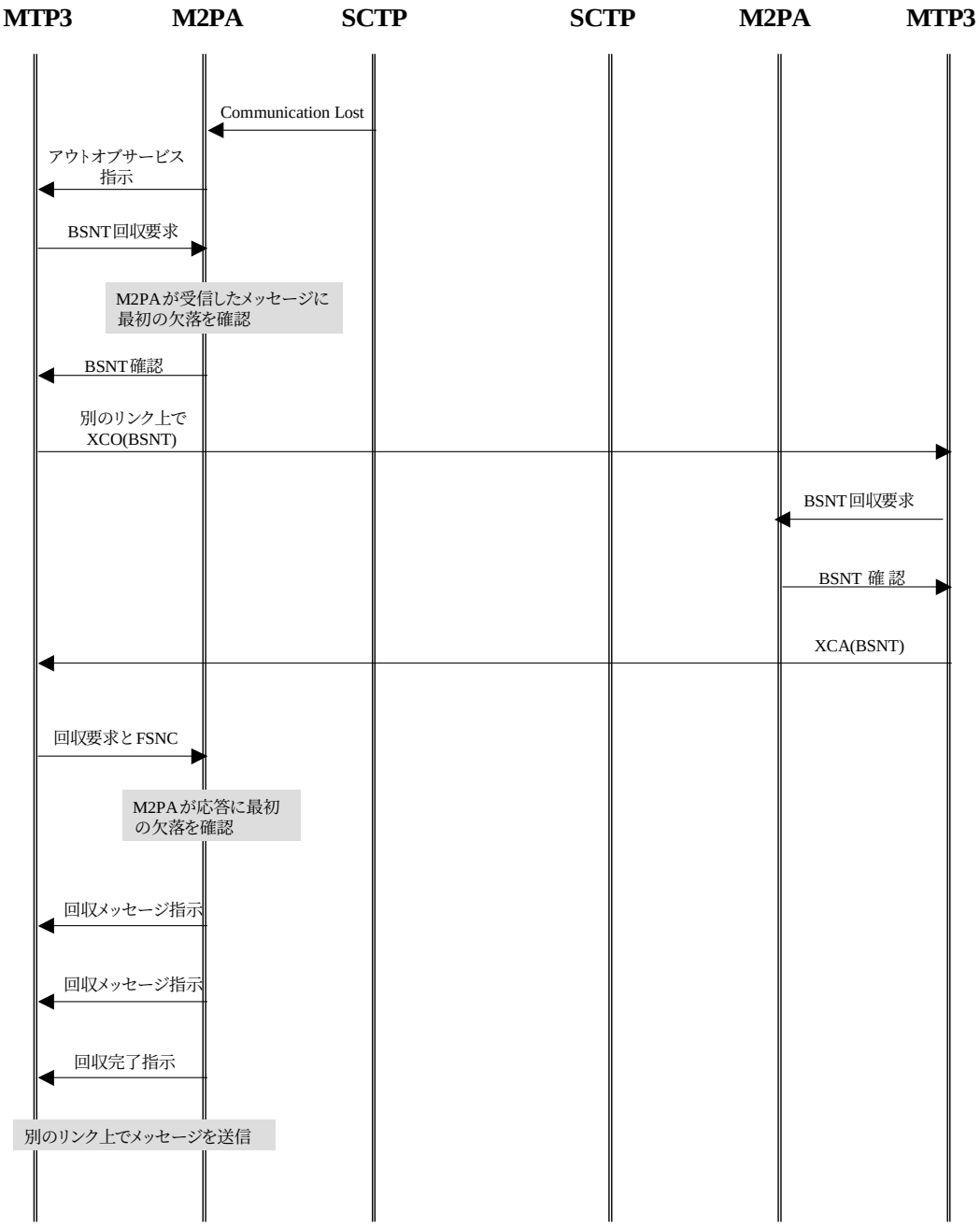


図 7-12 MTP3 リンク切替

7.7 セキュリティ

3.7 参照。

7.8 登録番号

7.8.1 SCTP ペイロードプロトコル識別子

M2PA のペイロードプロトコル識別子は未定である。

7.8.2 ポート番号

M2PA のポート番号は未定である。

7.9 将来の拡張性

3.9 参照。

8. SUA

8.1 序論

SUA(SS7 SCCP-User Adaptation Layer)は、SCTP(Stream Control Transmission Protocol)を用いて、TCAP あるいは RANAP などの SS7 SCCP-User シグナリングを IP 網上で転送するアダプテーションプロトコルである。

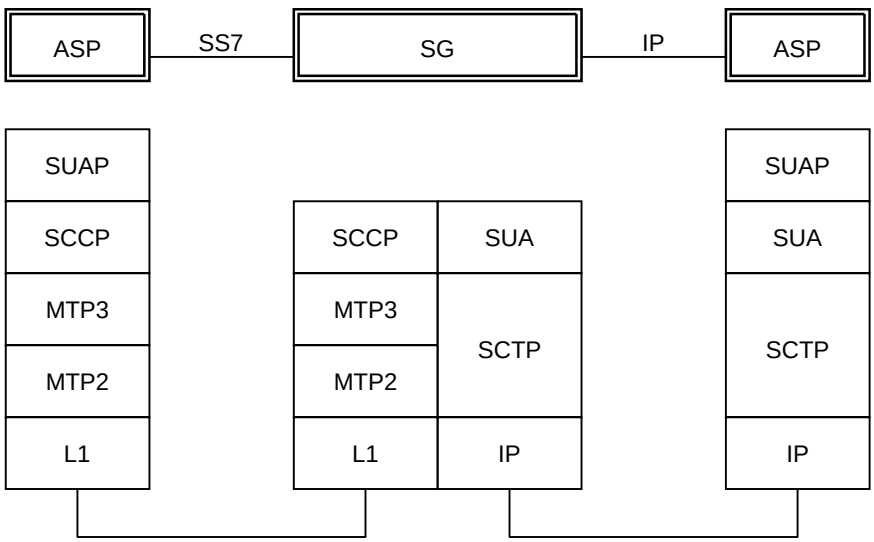
このプロトコルは、SG と IP シグナリングノードの間、あるいは、IP ネットワークにおける 2 つのエンドポイントの間で SCCP-User メッセージの転送をサポートする。転送において、以下の機能をサポートする。

- SCCP-User 部メッセージ(TCAP,RANAP など)の転送
- SCCP コネクションレスサービスのサポート
- SCCP コネクションオリエンテッドサービスのサポート
- SCCP-User プロトコルのシームレスなオペレーションのサポート
- 1 以上の IP ベースのシグナリングノードとの間の SCTP トランスポート結合管理のサポート
- 分散された IP ベースシグナリングノードのサポート
- 非同期に状態遷移を管理に通知することのサポート

上位レイヤプロトコルに応じて SUA は SCCP コネクションレスサービス、SCCP コネクションオリエンテッドサービス、または双方のサービスをサポートする必要がある。

8.1.1 SCCP コネクションレストランスポートアーキテクチャ

SCCP コネクションレストランスポートアーキテクチャを図 8-1 に示す。

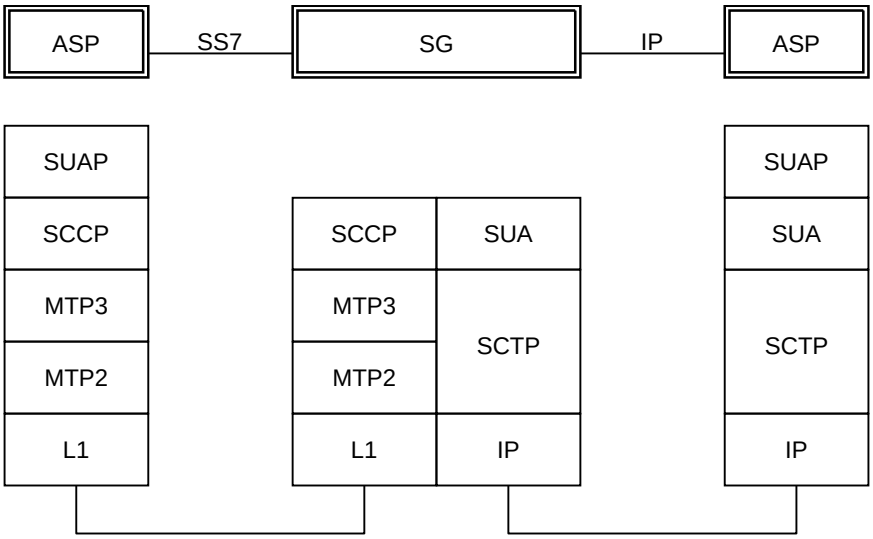


SUAP : SCCP/SUA Application User Protocol (TCAP など)
STP : SS7 Signalling Transfer Point

図 8-1 SCCP コネクションレストランスポートアーキテクチャ

8.1.2 SCCP コネクションオリエンテッドトランスポートアーキテクチャ

SCCP コネクションオリエンテッドトランスポートアーキテクチャを図 8-2 に示す。

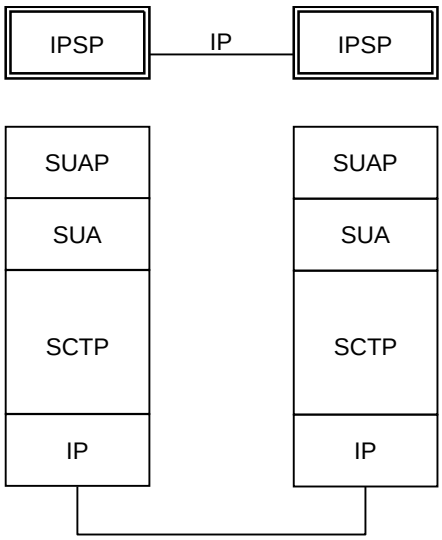


SUAP : SCCP/SUA Application User Protocol (RANAP/RNSAP など)
STP : SS7 Signalling Transfer Point

図 8-2 SCCP コネクションオリエンテッドアーキテクチャ

8.1.3 All IP アーキテクチャ

All IP **アーキテクチャを図** 8-3 **に示す。**



SUAP : SCCP/SUA Application User Protocol(RANAP/RNSAP など)

図 8-3 All IP アーキテクチャ

8.1.4 TTC における SCCP サービス規定範囲

TTC JT-Q.711 **では、**SCCP **コネクションオリエンテッドサービスを当面必要なしと判断し、規定していないため、本技術レポートにおいても、記述を行わない。**

8.2 用語

8.2.1 IPSP

IP ベースのアプリケーションサーバプロセス(IP Server Process)。

8.2.2 SGP

シグナリングゲートウェイ(SG)のプロセス(Signalling Gateway Process)。

8.2.3 インタフェース識別子

インタフェースを識別する整数または文字列である。

8.3 サービス

SUA は以下のサービスを使用する。

- SCTP 管理サービス
- UA 管理サービス
- ASP 管理サービス
- AS 管理サービス
- SUA サービス

8.3.1 SCTP 管理サービス

3.3.1 参照。

8.3.2 UA 管理サービス

3.3.2 参照。

8.3.3 ASP 管理サービス

3.3.3 参照。

8.3.4 AS 管理サービス

3.3.4 参照。

8.3.5 SUA サービス

SUA サービスのプリミティブ一覧を表 8-1 に示す。

表 8-1 SUA サービスプリミティブ一覧

サービスプリミティブ		概要
N-ユニットデータ転送	要求 指示	SCCP ユーザが他のユーザにデータを転送することを SCCP に要求する。 ユーザに SCCP からデータが伝送されていることを知らせる。
N-通知	指示	SCCP が発ユーザに最終着信後に到達できなかったメッセージを返送するための手段である。

8.4　メッセージ

標準メッセージフォーマットは、共通メッセージヘッダおよびメッセージタイプにより定義された０以上のリストからなる。

今後の互換性の為、たとえ本バージョンで指定されていないパラメタでも付加される場合がある。

8.4.1　共通メッセージヘッダ

3.4.1 参照。

8.4.2　パラメタ

3.4.1 参照。

8.4.3　SUA コネクションレスメッセージ

SUA コネクションレス転送メッセージおよびパラメタについて記述する。SUA メッセージは共通ヘッダと 0 以上のメッセージタイプで定義されたパラメタからなる。全メッセージタイプはパラメタを付加することができる。

8.4.3.1 コネクションレスデータ転送(CLDT)

このメッセージはある SUA から他 SUA へデータを転送する。

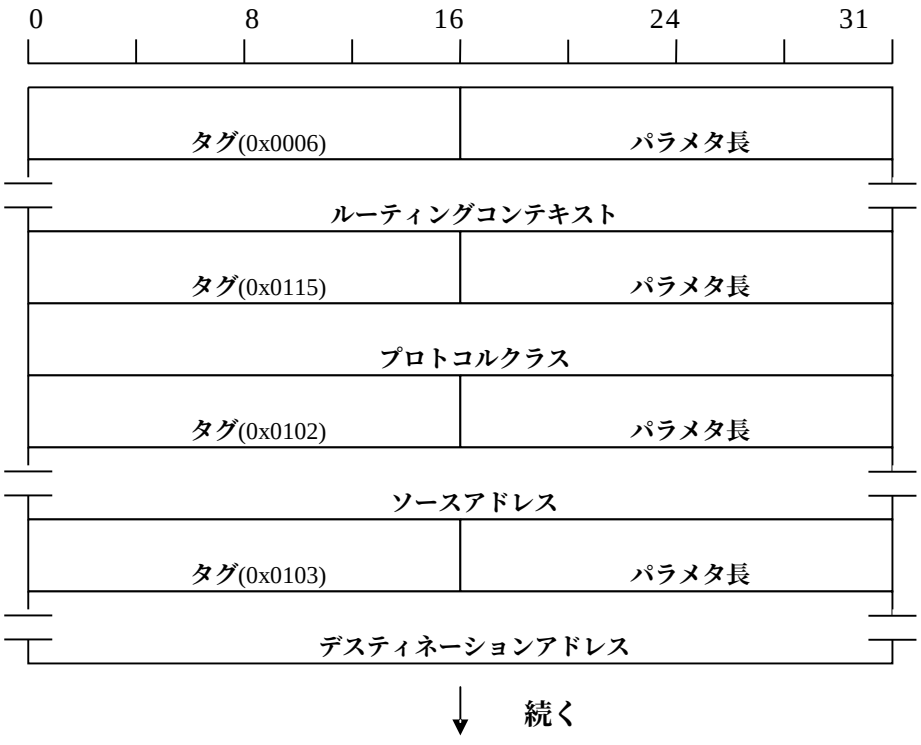


図 8-4CLDT メッセージフォーマット

・ルーティングコンテキスト(必須)
・プロトコルクラス(必須)
・ソースアドレス(必須)
・デスティネーションアドレス(必須)
・シーケンス制御(必須)
・SS7 ホップカウント(省略可能)
・重要性(省略可能)
・メッセージ優先順位(省略可能)
・関連識別子(省略可能)
・セグメンテーション(省略可能)
・データ(必須)

実装注釈)このメッセージは SCCP メッセージの UDT、XUDT、LUDT をカバーする。

8.4.3.2 **コネクションレスデータ応答**(CLDR)

このメッセージは、エラーオプションの戻りが設定されていた場合、受信した CLDT **メッセージのエラーをピアが報告するための応答メッセージとして用いられる。**

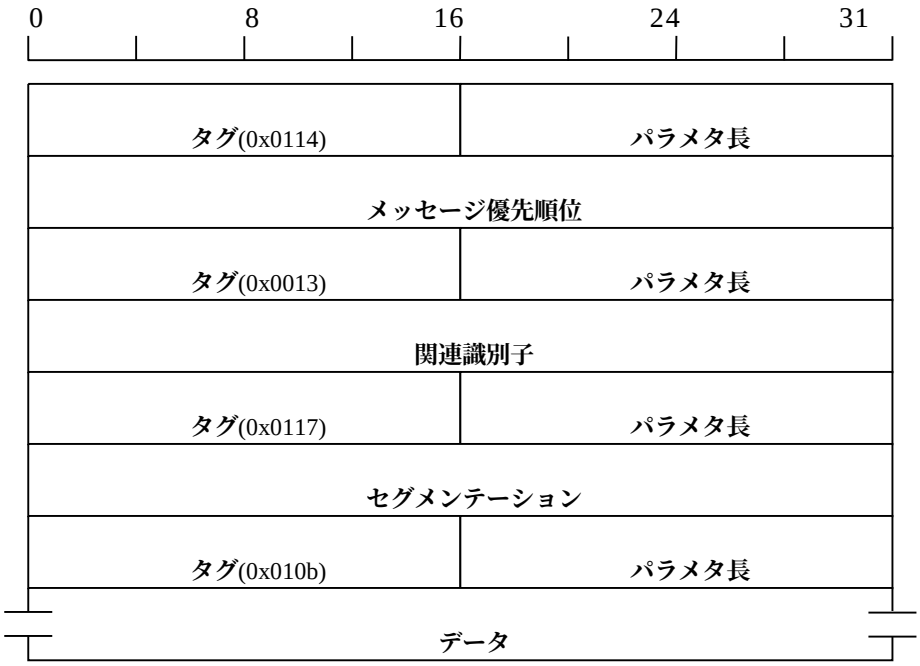


図 8-5 CLDR メッセ

・ルーティングコンテキスト(必須)
・SCCP 理由表示(必須)
・ソースアドレス(必須)
・デスティネーションアドレス(必須)
・SS7 ホップカウント(省略可能)
・重要性(省略可能)
・メッセージ優先順位(省略可能)
・関連識別子(省略可能)
・セグメンテーション(省略可能)
・データ(省略可能)

実装注釈)このメッセージは SCCP メッセージの UDTS、XUDTS、LUDTS をカバーする。

8.4.4 信号網管理メッセージ

8.4.4.1 Destination Unavailable (DUNA)

このメッセージは SCCP とローカル SCCP ユーザ間で PC-または N-状態により送信される。DUNA メッセージはディスティネーションまたは SCCP ユーザに到達不能な場合、SG または中継ノードから全ての関係する ASP に送られる。ASP での SUA ユーザは、DUNA を発行する SG または中継ノードを通して関係するディスティネーションまたは SCCP ユーザへのトラヒックをとめることを期待する。

DUNA メッセージパラメタのフォーマットを以下に示す。

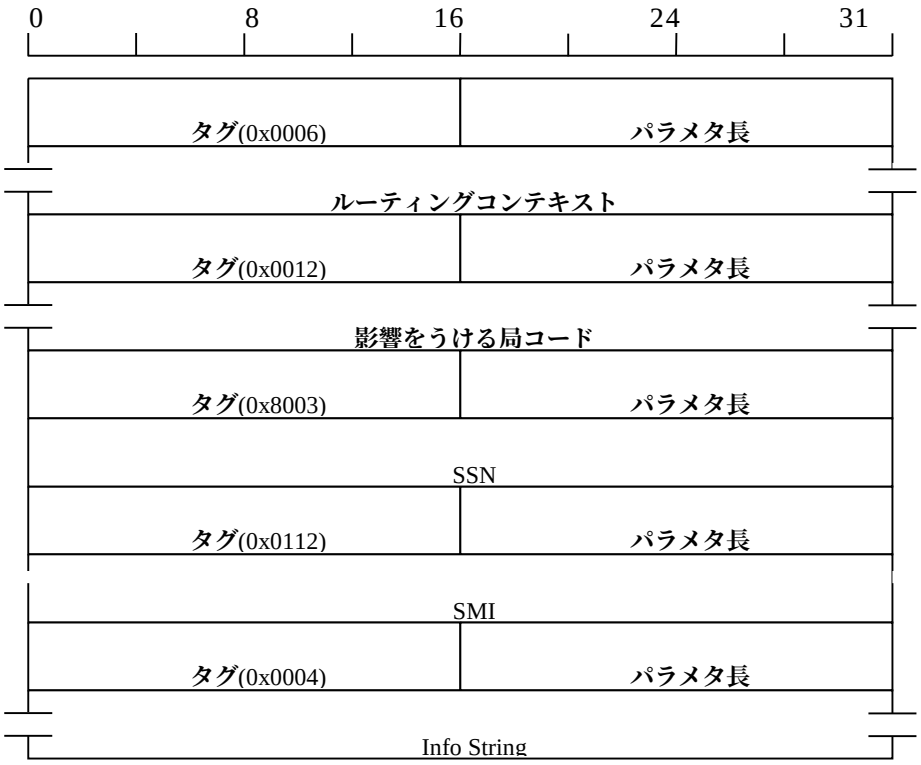


図 8-6 DUNA メッセージフォーマット

・ルーティングコンテキスト(省略可能)
・影響をうける局コード(必須)※1
・SSN(省略可能)※1
・SMI(省略可能)
・Info String(省略可能)

※1：SSN が含まれる場合、DUNA メッセージは SCCP N-状態プリミティブと一致する。SSN が含まれない場合は、DUNA メッセージは SCCP N-信号局状態プリミティブと一致する。

8.4.4.2 Destination Available (DAVA)

このメッセージは SCCP とローカル SCCP ユーザ間で PC-または N-状態により送信される。DAVA メッセージはディスティネーションまたは SCCP ユーザに到達可能となった場合、SG または中継ノードから全ての関係する ASP に送られる。ASP での SUA ユーザは、DAVA を発行する SG または中継ノードを通して関係するディスティネーションまたは SCCP ユーザへのトラヒックを再開することを期待する。

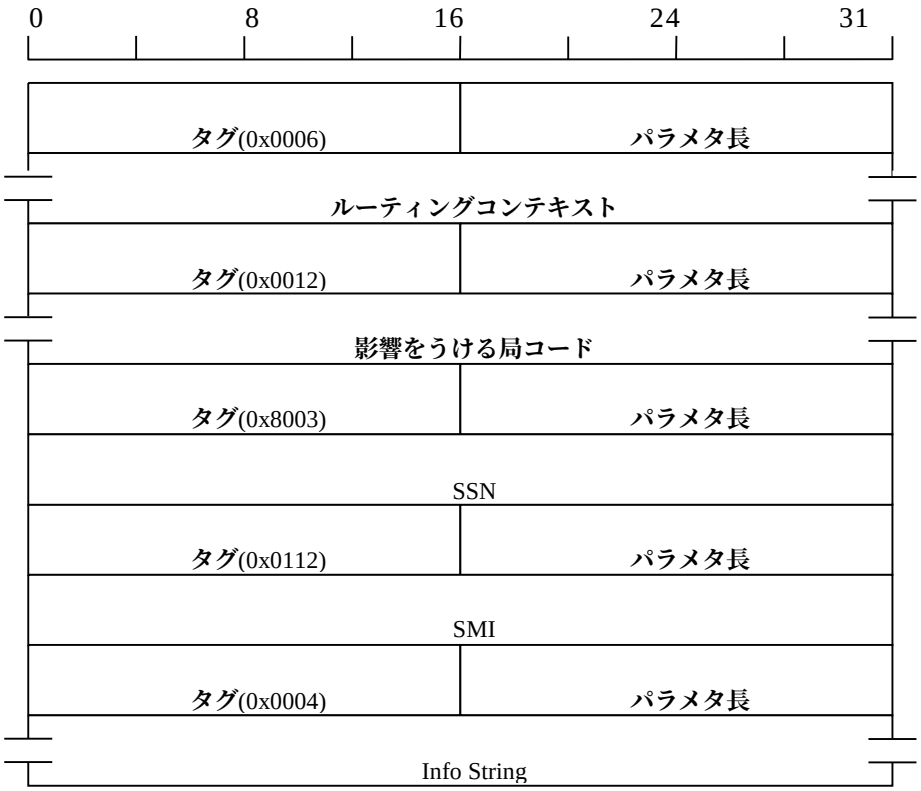


図 8-7 DAVA メッセージフォーマット

・ルーティングコンテキスト(省略可能)

・影響をうける局コード(必須)※１

・SSN**(省略可能)※１**

・SMI**(省略可能)**

・Info String**(省略可能)**

※１：SSN **が含まれる場合、**DAVA **メッセージは** SCCP N-**状態プリミティブと一致する。**SSN **が含まれない場合、**DAVA **メッセージは** SCCP N-**信号局状態プリミティブと一致する。影響をうける局コードは、**SSN **が存在する場合１つの局コードのみ含まれる。**

8.4.4.3 Destination State Audit (DAUD)

このメッセージは影響するディスティネーションへのルート状態が利用可能か問い合わせるために ASP **から** SG**(また中継ノード)に送られる。**DAUD **は** DAVA **を受信する迄、**ASP **が** DUNA **を受信後周期的に送出されるかもしれない。**DAUD **は** ASP **が** SG**(または中継ノード)から独立で回復する場合にも送出される。**

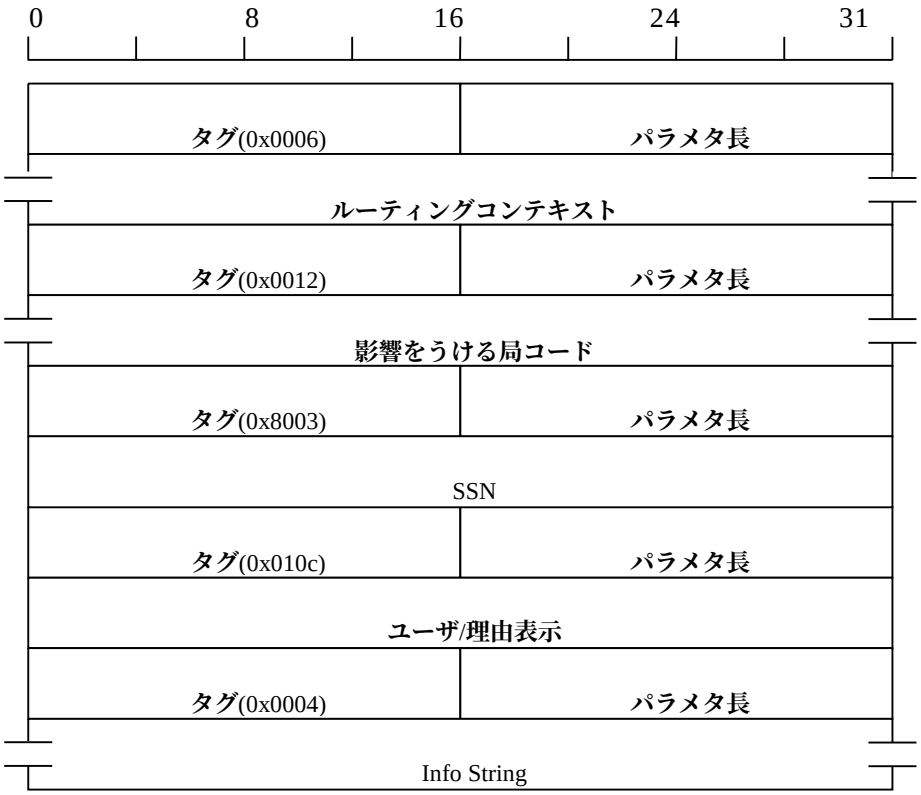


図 8-8 DAUD メッセージフォーマット

・ルーティングコンテキスト(省略可能)

・影響をうける局コード(必須)※ 1

・SSN**(省略可能)※ 1**

・ユーザ/理由表示(省略可能)

・Info String(**省略可能**)

※ 1 :SSN **が存在する場合、**DAUD **メッセージは** N-**状態プリミティブであり、**SSN **が存在しない場合は、**N-**信号局状態プリミティブである。**

8.4.4.4 Network Congestion (SCON)

このメッセージは指定ディスティネーション側への共通線信号網輻輳レベルが変更されたことを SG **または中継ノードから全関連** ASP **に通知するために送信される。**

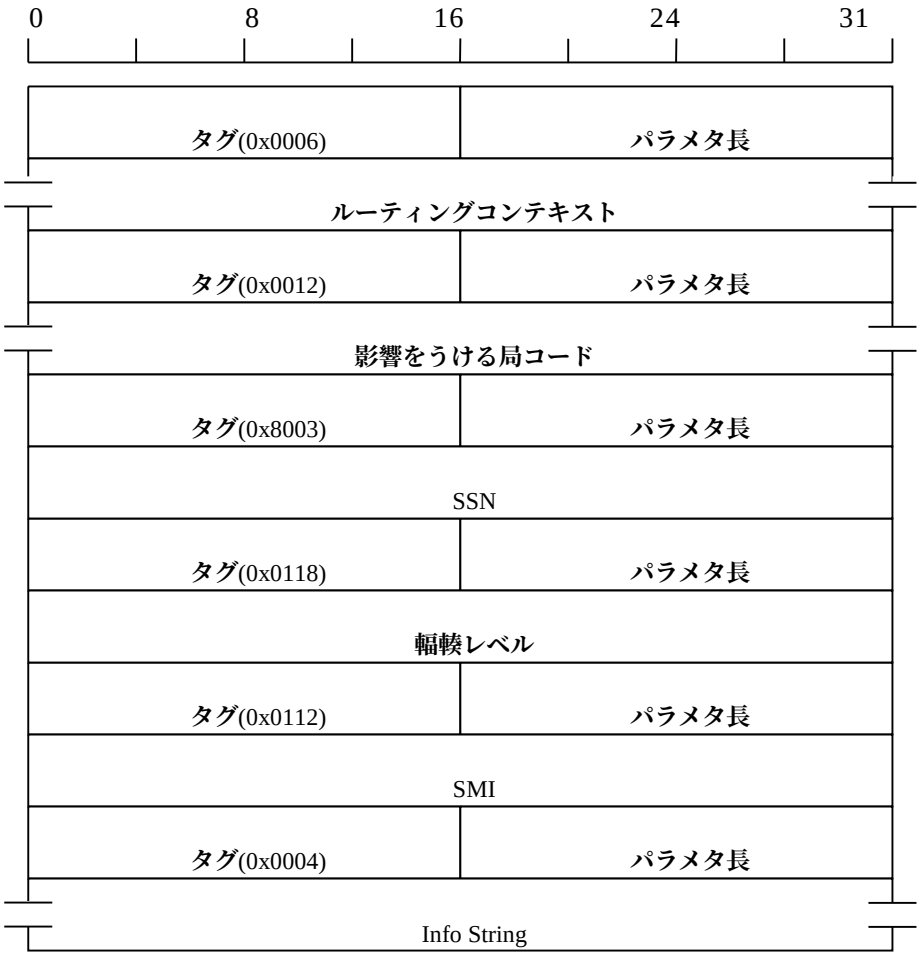


図 8-9 SCON メッセージフォーマット

・ルーティングコンテキスト(省略可能)

・影響をうける局コード(必須)※ 1

・SSN**(省略可能)※ 1**

・輻輳レベル(必須)

・SMI**(省略可能)**

・Info String**(省略可能)**

※ 1 :SSN **を含む場合、**SCON **メッセージは** SCCP N-**状態プリミティブと一致する。**SSN **を含まない場合、**SCON **メッセージは** SCCP N-**信号局状態プリミティブと一致する。**

8.4.4.5 Destination User Part Unavailable (DUPU)

このメッセージは SG **によって共通線信号網のリモートピアが利用不可になったことを** ASP **に知らせるために用いられる。**

DUPU **のメッセージパラメタフォーマットを以下に示す。**

図 8-10 DUPU メッセージフォーマット

・ルーティングコンテキスト(省略可能)

・影響をうける局コード(必須)※ 1

・ユーザ/理由表示(必須)

・Info String(**省略可能**)

※ 1 ：DUPU **メッセージは** SCCP N-**信号局状態プリミティブと一致する。**

8.4.4.6 Destination Restricted (DRST)

DRST **メッセージは、**SG **が 1 つ以上の宛先が規制されていると判断したことを示すため、または、**DAUD **メッセージに対し応答することを示すために、**SG **から全関連** ASP **に対し任意に送信する。**ASP **の** SUA **レイヤは、選択できるルートが存在してかつ利用可能な場合に、選択すべき同優先度の** SGP **経由で影響する宛先へトラヒックをおくることを期待する。もし、影響する宛先は** ASP **により利用不可とみなされた場合は、ピアは影響する宛先にトラヒックを再開することを知らせるべきである。この場合、**SUA **レイヤは** DRST **メッセージを発行する** SGP **を通しトラヒックをルートさせるべきである。**

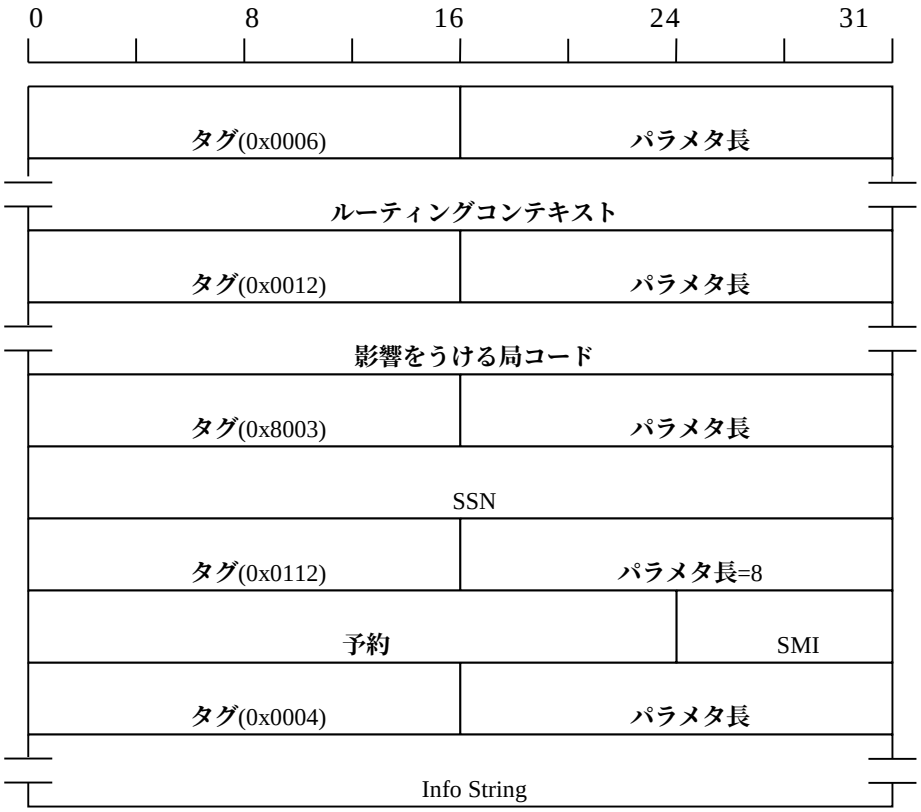


図 8-11 DRST メッセージフォーマット

・ルーティングコンテキスト(省略可能)
・影響をうける局コード(必須)※1
・SSN(省略可能)※1
・SMI(省略可能) ※1
・Info String(省略可能)

※1：影響をうける局コードは到達不能になったノードを参照する。メッセージパラメタに SSN が含まれる場合、DRST メッセージは SCCP N-調整プリミティブと一致する。SMI パラメタも DRST メッセージに含まれる場合は、DRST メッセージは SCCP N-調整要求および N-調整指示プリミティブに一致する。それ以外は、SCCP N-調整応答及び SCCP N-調整確認プリミティブに一致する。影響をうける局コードは、SSN が存在する場合１つの局コードのみ含まれる。

8.4.5 ASP トラヒック管理メッセージ

8.4.5.1 ASPAC メッセージ(ASP Active)

ASPAC メッセージはリモート側の SUA ピアに対して稼働状態に遷移し特定の AS からの信号トラヒックを受信する準備を行うよう指示するため ASP が送出する。フォーマットは以下の通り。

図 8-12 ASPAC メッセージのフォーマット

パラメタ	
トラヒックモードタイプ	オプション
ルーティングコンテキスト	オプション
TID ラベル	オプション
DRN ラベル	オプション
Info String	オプション

8.4.5.2 ASPAC ACK メッセージ(ASP Active Acknowledgment)

3.4.7.2 節参照

8.4.5.3 ASPIA メッセージ(ASP Inactive)

ASPIA メッセージはリモート側の SUA ピアに対して特定の AS からの信号トラヒックの処理を終了させるよう指示するため ASP が送出する。フォーマットは以下の通り。

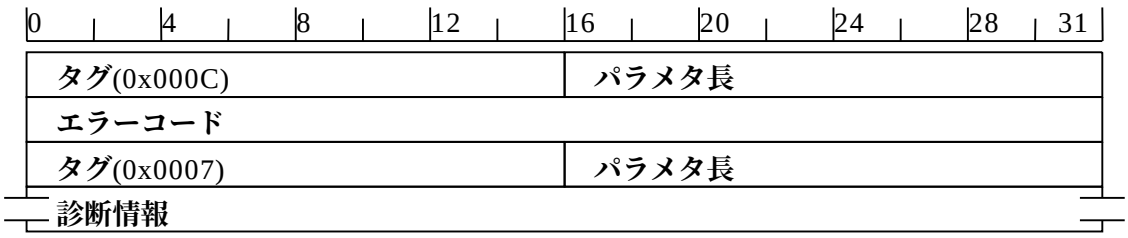


図 8-13 ASPIA メッセージのフォーマット

パラメタ	
ルーティングコンテキスト	オプション
Info String	オプション

8.4.5.4 ASPIA ACK メッセージ(ASP Inactive Acknowledgment)

3.4.7.4 節参照。

8.4.6　SUA マネージメントメッセージ

これらのメッセージは SUA と下位の SCCP 表示を管理するため SUA レイヤで使用される。

8.4.6.1 ERR メッセージ(Error)

ERR メッセージは２つの SUA ピアの間でエラー状態を示すために送出される。Data parameter はオプションであり、エラーの記録またはデバッグのために使用可能である。

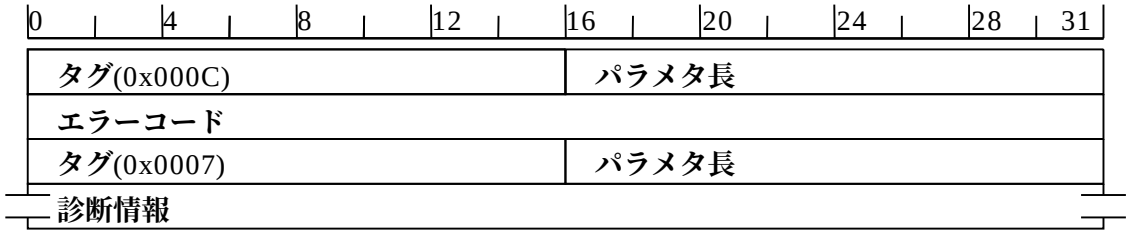


図 8-14 ERR メッセージのフォーマット

パラメタ	
エラーコード	必須
診断情報	オプション

表 8-2 SUA エラーコード

エラー名称	エラーコード
無効バージョン	０ｘ０１
無効インタフェース識別子	０ｘ０２
該当メッセージクラスなし	０ｘ０３
該当メッセージタイプなし	０ｘ０４
該当トラフィックモードなし	０ｘ０５
予期しないメッセージ	０ｘ０６
プロトコルエラー	０ｘ０７
無効ストリーム識別子	０ｘ０９
保守閉塞による拒否	０ｘ０D
ASP 識別子が必要	０ｘ０E
無効ルーティングコンテキスト	０ｘ１０
無効パラメタ値	０ｘ１１
パラメタフィールドエラー	０ｘ１２

エラー名称	エラーコード
期待しないパラメタ	０ｘ１３
ディスティネーション状態不明	０ｘ１４
無効ネットワークアピアランス	０ｘ１５
パラメタ紛失	０ｘ１６
ルーティングキー変更拒否	０ｘ１７
無効ロードシェアラベル	０ｘ１８

8.4.6.2 NTFY メッセージ(Notify)

Notify メッセージは SUA のイベントについて自律的な指示メッセージを SUA ピアに提供するために使用される。

図 8-15 NTFY メッセージのフォーマット

パラメタ

状態　　必須

ASP 識別子　　オプション*1

ルーティングコンテキスト　　オプション

Info String　　オプション

（注１）ASP Identifier は IPSP/SGP が予め設定されたアドレス／ポート番号を認識できないときに使用すべきである（例えば、ASP の存在するホストがダイナミックにアドレス／ポート番号が割り当てられるものであるとき）。

8.4.7　ルーティングキー管理(RKM)メッセージ

8.4.7.1 REG REQ メッセージ(Registration Request)

REG REQ メッセージは１個以上ルーティングキー の登録をリモートピアに対して指示するため ASP が使用する。一般的に ASP はこのメッセージを SGP に送出し、そして ASP はルーティングコンテキストの値を含んだ REG RSP メッセージが返ってくることを期待している。

フォーマットは以下の通り。

図 8-16 REG REQ メッセージのフォーマット

REG REQ メッセージは以下のパラメタを含む。

パラメタ

ネットワークアピアランス　　オプション

ルーティングキー　　必須*1

ASP **能力　　オプション**

（注１）一つ又は複数のルーティングキーパラメタが一つの REG REQ **メッセージに含まれることがある。**

8.4.7.2 REG RSP **メッセージ**(Registration Response)

REG RSP **メッセージは** ASP **からの** REG REQ **メッセージの結果を通知するために** SG **が** ASP **に送出する。成功時には、**REG RSP **メッセージは一つ又は複数のルーティングキーがアサインされたルーティングコンテキストを含む。フォーマットは以下の通り。**

図 8-17 REG RSP メッセージのフォーマット

REG RSP **メッセージは以下のパラメタを含む。**

パラメタ

ネットワークアピアランス　　オプション

登録結果　　必須*1

（注１）　一つ又は複数の登録結果 パラメタが一つの REG RSP **メッセージに含まれることがある。**

8.4.7.3 DEREG REQ メッセージ(Deregistration Request)

DEREG REQ メッセージはリモート SUA ピアに対して既存のルーティングキーに替えて再登録を指示するため ASP が送出する。一般的に ASP は SGP にこのメッセージを送出し、そして ASP はルーティングコンテキストの値を含んだ DEREG RSP メッセージが返ってくることを期待している。

フォーマットは以下の通り。

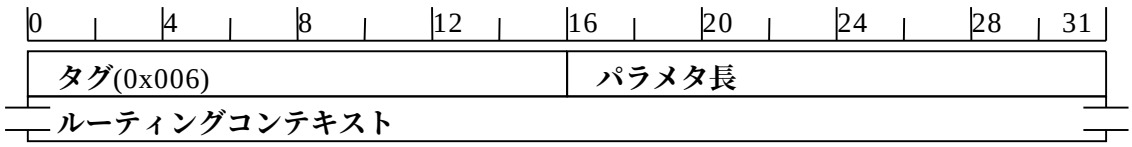


図 8-18 DEREG REQ メッセージのフォーマット

DEREG REQ メッセージは以下のパラメタを含む。

パラメタ	
ルーティングコンテキスト	必須

8.4.7.4 DEREG RSP メッセージ(Deregistration Response)

DEREG RSP メッセージは REREG REQ メッセージに対する応答としてリモート側 SUA ピアから送出される。フォーマットは以下の通り。

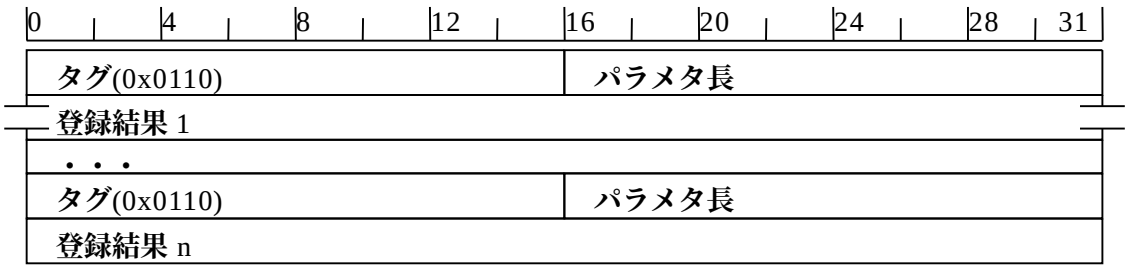


図 8-19 DEREG RSP メッセージのフォーマット

DEREG RSP メッセージは以下のパラメタを含む。

パラメタ	
登録結果	必須*1

一つ又は複数の Deregistration Result パラメタが DEREG RSP メッセージに含まれることがある。

8.5 手順

8.5.1 SCTP 管理サービス手順

3.5.1 参照。

8.5.2 UA 管理サービス手順

3.5.3 参照。

8.5.3 ASP 管理サービス手順

3.5.4 参照。

8.5.4　AS 管理サービス手順

3.5.5 参照。

8.5.5　ルーティングキー管理サービス手順

ルーティングキー管理サービス手順は必須ではない。

8.5.5.1 ルーティングキー登録手順

ルーティングキー登録手順を以下に示す。

1.ASP 上の SUA は REG REQ メッセージを送信する。

2.SG 上の SUA は REG REQ メッセージを受信し、ルーティングキーと既存ルーティングキーを比較する。

3.ルーティングキーが既存ルーティングキーと一致する場合、ルーティングキーに関連付けられている AS に ASP を追加することができる。

4.SG は REG RSP(登録成功)メッセージを送信する。

上記手順 2 において、ルーティングキーと既存ルーティングキーが一致しない場合、手順 3 は次のように変化する。

5.ルーティングキーが既存ルーティングキーと一致しない場合、ルーティングキーに関連付けられる新たな AS を生成し、ASP を AS に追加することができる。

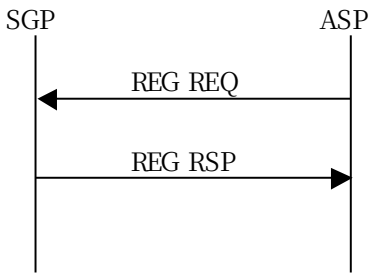


図 8-20 ルーティングキー登録シーケンス

8.5.5.2 ルーティングキー登録解除手順

ルーティングキー登録解除手順を以下に示す。

1.ASP 上の SUA は DEREG REQ メッセージを送信する。

2.SG 上の SUA は DEREG REQ メッセージを受信し、ルーティングコンテキストと関連付けられている AS から ASP を解除する。

3.SG は DEREG RSP(登録解除成功)メッセージを送信する。

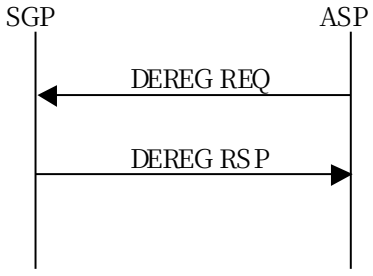


図 8-21　ルーティングキー登録解除シーケンス

8.5.6　SS7 対地状態管理手順

8.5.6.1 対地状態変更通知手順

1.N-状態、N-信号局状態、N-通知プリミティブを受信した SGP の SUA レイヤは、対応する SSNM メッセージ、DUNA、DAVA、SCON、DUPU メッセージを ASP に対して送信する。

2.SSNM メッセージを受信した ASP の SUA レイヤは、適切な指示プリミティブを SUA ユーザに通知する。

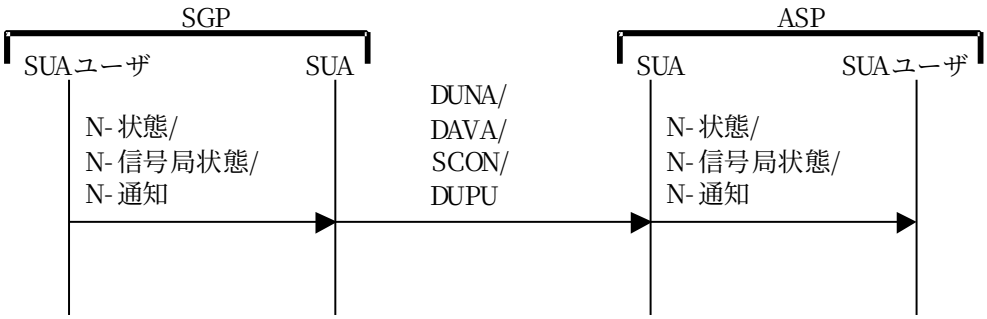


図 8-22 対地状態変更通知シーケンス

8.5.6.2 対地監査手順

対地監査手順を以下に示す。

1.ASP と SG の一時的な通信途絶等を契機として、ASP 上の SUA は SG 上の SUA に対して DAUD メッセージを送信して信号局状態を問い合わせる。

2.SG 上の SUA は DAUD メッセージを受信すると、問い合わせを受けた信号局の状態に応じて、DUNA、DAVA、SCON、DRST メッセージを返信する。

3.ASP の正当性が確認できない場合等において、DAUD メッセージに応答しない場合には ERROR メッセージを返送する。

上記手順のシーケンスを図 8-23 に示す。

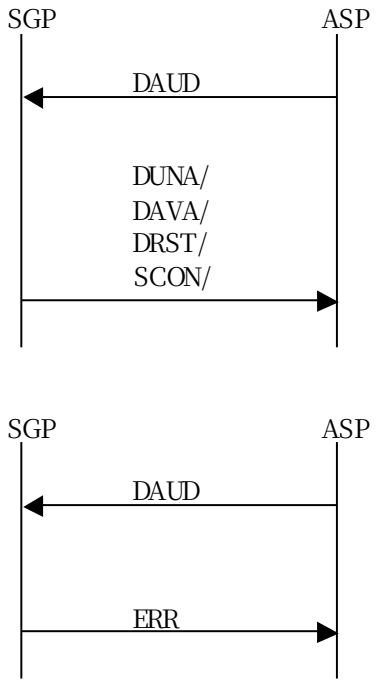


図 8-23 対地監査シーケンス

8.5.7　SCCP-SUA インタワーキング手順

8.5.7.1 分割／組立て手順

信号メッセージが PDU に収まらない場合、SG、ASP、IPSP はメッセージの分割／組立てを行う。もし、分割／組立てを行うことができない場合は、CLDR や RESRE/COERR メッセージ中の適切なエラーを用いて対向装置に伝える。

8.5.7.2 ロードシェアリング手順

1.ASP は TID パラメタを含めた ASPAC メッセージを SG に対して送信する。DRN パラメタはあってもなくてもよい。

2.SG は、受信した ASPAC メッセージに含まれる TID パラメタと DRN パラメタをチェックする。

- TID パラメタ中、Start パラメタと End パラメタが 0～31 の間にあること。
- DRN パラメタ中、Start パラメタと End パラメタが 0～23 の間にあること。
- 0 < (Start – End + 1) <=16
- Start パラメタと End パラメタが同一ルーティングコンテキストで対応させられた ASP 間で同一であること。
- TID パラメタと DRN パラメタのラベルがルーティングコンテキスト間でユニークであること。

3.いずれかのチェックに失敗した場合、SG は ASP に対して「無効ロードシェアラベル」の ERROR メッセージを返信する。

8.5.7.3 TCAP トラフィックのメッセージ分散

TID を含まない TCAP メッセージ(Query、Begin、Unidirectional)は、SG は TCAP メッセージを稼動状態の ASP 間で分散して処理させてもよい。もし、TID を含む場合には、SG はラベルを抽出し、該当する ASP を選択する。

8.6　通信シーケンス例

8.6.1　初期化シーケンス

2 台の ASP による冗長構成の初期化シーケンスを示す。トラフィックモードはオーバーライドモードとする。初期化完了時、ASP1 は稼動状態、ASP2 は起動状態となる。

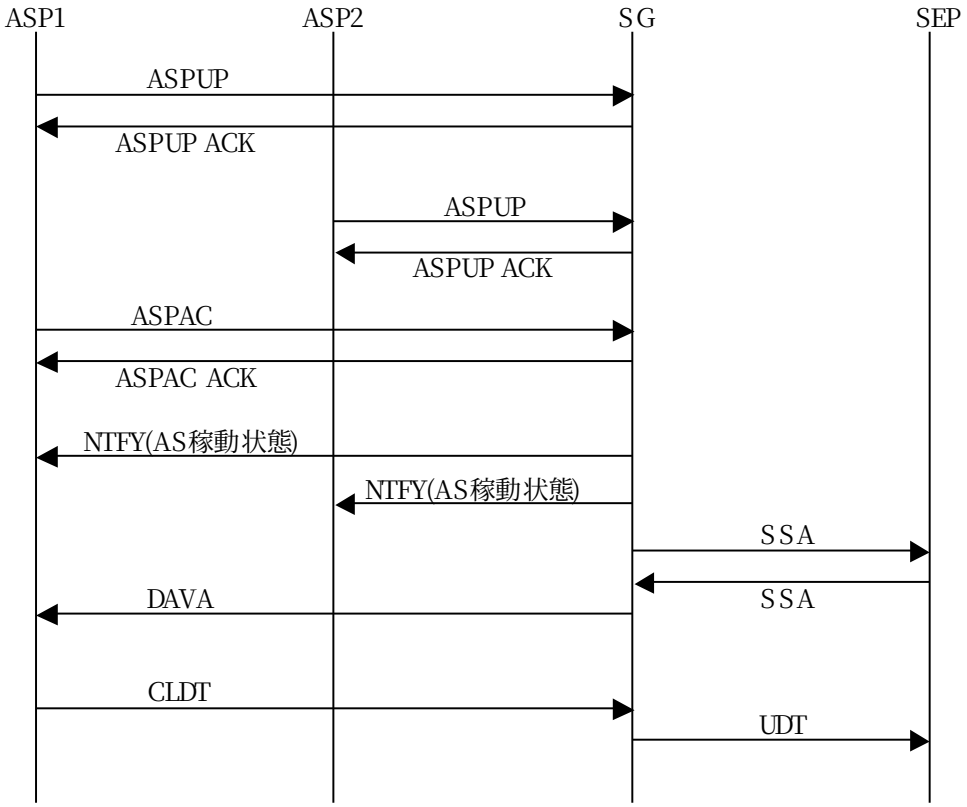


図 8-24 初期化シーケンス例(ASP)

8.6.2　フェイルオーバシーケンス

8.6.2.1 フェイルオーバシーケンス 1

SEP のフェイルオーバシーケンスを図 8-25 に示す。

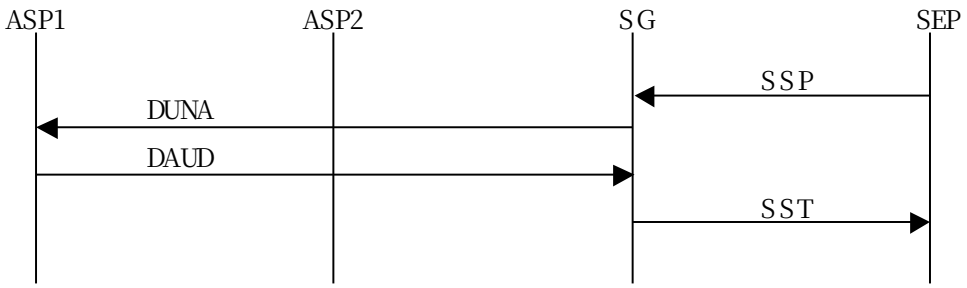


図 8-25 フェイルオーバシーケンス例(ASP)

8.6.2.2 フェイルオーバシーケンス 2

3.6.2.1 参照。

8.6.2.3 フェイルオーバシーケンス 3

ASP1 が稼動状態から起動状態に遷移し、ASP2 への信号トラフィックの引継ぎが失敗するシーケンスを以下に示す。

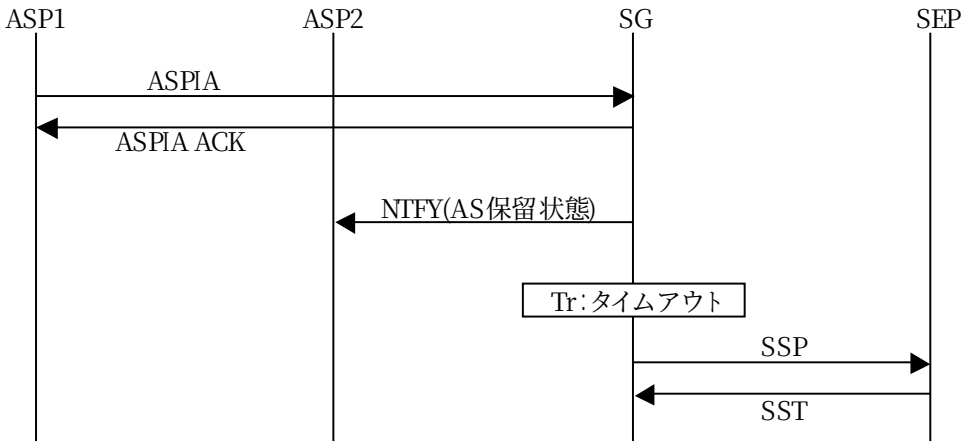


図 8-26 フェイルオーバ失敗シーケンス例(ASP)

8.6.3 IPSP シーケンス

8.6.3.1 初期化シーケンス

2 台の IPSP による冗長構成の初期化シーケンスを示す。トラフィックモードはオーバーライドモードとする。初期化完了時、IPSP-a1 は稼動状態、IPSP-a2 は起動状態となる。

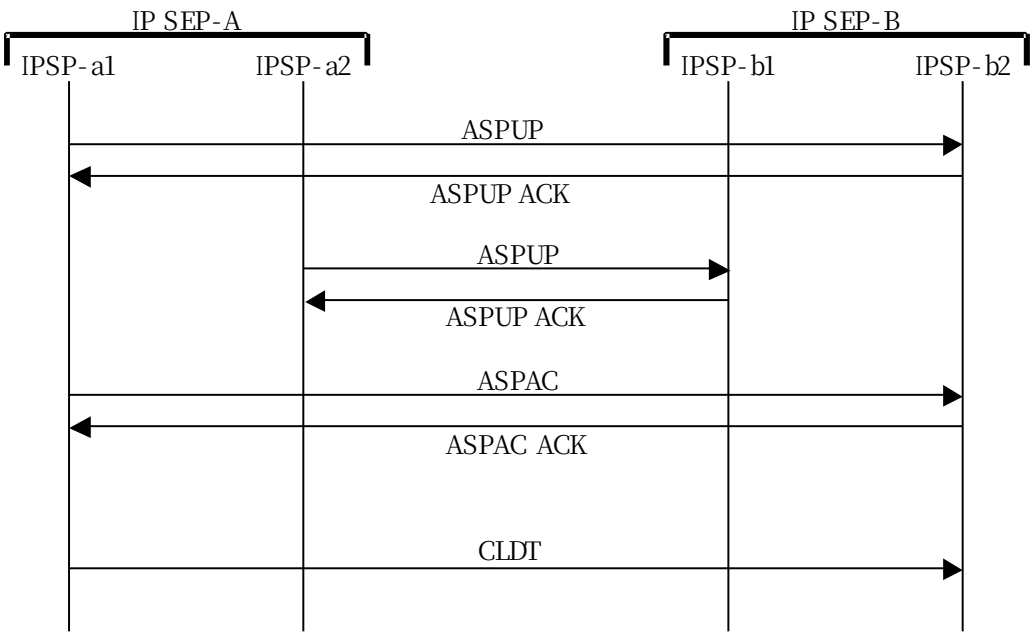


図 8-27 初期化シーケンス例(IPSP)

8.6.3.2 フェイルオーバシーケンス

ASP-a1 が稼動状態から起動状態に遷移し、ASP-a2 が稼動状態に遷移して信号トラフィックを引き継ぐシーケンスを以下に示す。

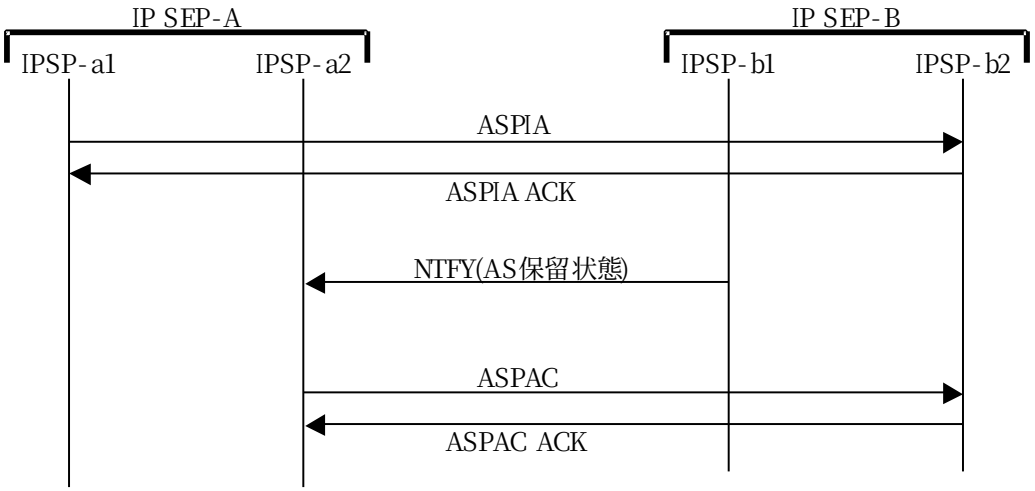


図 8-28 フェイルオーバシーケンス例(IPSP)

8.7 セキュリティ

3.7 参照。

8.8 登録番号

8.8.1 SCTP ペイロードプロトコル識別子

SUA のペイロードプロトコル識別子の値は”4”である。

8.8.2 ポート番号

SCTP におけるポート番号は”14001”である。

8.9 将来の拡張性

3.9 参照。

9．技術レポートと原標準の対応

本技術レポートと原標準の対応を表 9-1 に示す。

表 9-1 本技術レポートと原標準の対応

本技術レポート	原標準
2.1 節	[SCTP]の第 1 章
2.2 節	[SCTP]の第 2 章
2.3 節	[SCTP]の第 3 章
2.4 節	[SCTP]の第 4 章
2.5 節	[SCTP]の第 5 章
2.6 節	[SCTP]の第 6 章
2.7 節	[SCTP]の第 7 章
2.8 節	[SCTP]の第 8 章
2.9 節	[SCTP]の第 9 章
2.10 節	[SCTP]の第 10 章
2.11 節	[SCTP]の第 11 章
2.12 節	[SCTP]の第 12 章
2.13 節	[SCTP]の第 13 章
2.14 節	[SCTP]の第 14 章
3.1 節	[IUA][M3UA][M2UA][M2PA]の第 1 章
3.2 節	[IUA][M3UA][M2UA][M2PA]の第 1 章
3.3 節	[IUA][M3UA][M2UA][M2PA][SUA]の第 1 章
3.4 節	[IUA][M3UA][M2UA][M2PA][SUA]の第 3 章
3.5 節	[IUA][M3UA][M2UA][M2PA]の第 4 章
3.6 節	[IUA][M3UA][M2UA][M2PA]の第 5 章
3.7 節	[IUA][M3UA][M2UA][M2PA]の第 6 章
3.8 節	[IUA][M3UA][M2UA][M2PA]の第 7 章
3.9 節	[IUA][M3UA][M2UA][M2PA]の第 7 章
4.1 節	[IUA]の第 1 章
4.2 節	[IUA]の第 1 章

本技術レポート	原標準
4.3 節	[IUA]の第 1 章
4.4 節	[IUA]の第 3 章
4.5 節	[IUA]の第 4 章
4.6 節	[IUA]の第 5 章
4.7 節	[IUA]の第 6 章
4.8 節	[IUA]の第 7 章
4.9 節	[IUA]の第 7 章
5.1 節	[M3UA]の第 1 章
5.2 節	[M3UA]の第 1 章
5.3 節	[M3UA]の第 1 章
5.4 節	[M3UA]の第 3 章
5.5 節	[M3UA]の第 4 章
5.6 節	[M3UA]の第 5 章
5.7 節	[M3UA]の第 6 章
5.8 節	[M3UA]の第 7 章
5.9 節	[M3UA]の第 7 章
6.1 節	[M2UA]の第 1 章
6.2 節	[M2UA]の第 1 章
6.3 節	[M2UA]の第 1 章
6.4 節	[M2UA]の第 3 章
6.5 節	[M2UA]の第 4 章
6.6 節	[M2UA]の第 5 章
6.7 節	[M2UA]の第 6 章
6.8 節	[M2UA]の第 7 章
6.9 節	[M2UA]の第 7 章
7.1 節	[M2PA]の第 1 章

本技術レポート	原標準
7.2 節	[M2PA]の第 1 章
7.3 節	[M2PA]の第 1 章
7.4 節	[M2PA]の第 3 章
7.5 節	[M2PA]の第 4 章
7.6 節	[M2PA]の第 5 章
7.7 節	[M2PA]の第 6 章
7.8 節	[M2PA]の第 7 章
7.9 節	[M2PA]の第 7 章
8.1 章	[SUA]の第１章
8.2 章	[SUA]の第１章
8.3 章	[SUA]の第１章
8.4 章	[SUA]の第３章
8.5 章	[SUA]の第４章
8.6 章	[SUA]の第５章
8.7 章	[SUA]の第６章
8.8 章	[SUA]の第７章
8.9 章	[SUA]の第７章

文献

[JT-Q701] 「メッセージ転送部　信号システムの機能概要」，第 2 版, TTC, 1990 年 11 月.

[JT-Q702] 「メッセージ転送部　信号データリンク部」，第 1 版, TTC, 1987 年 4 月.

[JT-Q703] 「メッセージ転送部　信号リンク機能部」，第 3 版, TTC, 1994 年 4 月.

[JT-Q704] 「メッセージ転送部　信号網機能部」，第 3 版, TTC, 1992 年 4 月.

[JT-Q711] 「信号接続制御部（ＳＣＣＰ）の機能」，第 2 版, TTC, 1997 年 3 月

[JT-Q920] 「ISDN ユーザ・網インタフェースレイヤ 2 概要」，第 4 版, TTC, 1993 年 11 月.

[JT-Q921] 「ISDN ユーザ・網インタフェースレイヤ 2 仕様」，第 5.1 版, TTC, 2000 年 2 月.

[ITU-T Q.2210] “Message transfer part level 3 functions and messages using the services of ITU-T Recommendation Q.2140”, ITU-T, 1996 年 7 月.

[RFC768] "User Datagram Protocol", IETF, 1980 年 8 月.

[RFC793] "Transmission Control Protocol", IETF, 1981 年 9 月.

[RFC1123] "Requirements for Internet hosts – application and support", IETF, 1989 年 10 月.

[RFC1191] "Path MTU Discovery", IETF, 1990 年 11 月.

[RFC1700] "Assigned Numbers", IETF, 1994 年 10 月.

[RFC1750] "Randomness Recommendations for Security", IETF, 1994 年 12 月.

[RFC1950] "ZLIB Compressed Data Format Specification version 3.3", IETF, 1996 年 5 月.

[RFC1981] "Path MTU Discovery for IP version 6", IETF, 1996 年 8 月.

[RFC1982] "Serial Number Arithmetic", IETF, 1996 年 8 月.

[RFC2026] "The Internet Standards Process – Revision 3", 1996 年 10 月.

[RFC2104] "HMAC: Keyed-Hashing for Message Authentication", IETF, 1997 年 3 月.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", IETF, 1997 年 3 月.

[RFC2196] "Site Security Handbook", IETF, 1997 年 9 月.

[RFC2401] "Security Architecture for the Internet Protocol", IETF, 1998 年 11 月.

[RFC2402] "IP Authentication Header", IETF, 1998 年 11 月.

[RFC2406] "IP Encapsulating Security Payload (ESP)", IETF, 1998 年 11 月.

[RFC2408] "Internet Security Association and Key Management Protocol", IETF, 1998 年 11 月.

[RFC2409] "The Internet Key Exchange (IKE)", IETF, 1998 年 11 月.

[RFC2434] "Guidelines for Writing an IANA Considerations Section in RFCs", IETF, 1998 年 10 月.

[RFC2460] "Internet Protocol, Version 6 (IPv6) Specification", IETF, 1998 年 12 月.

[RFC2522] "Photuris: Session-Key Management Protocol", IETF, 1999 年 3 月.

[RFC2581] “TCP Congestion Control”, IETF Standard Track RFC, 1999 年 8 月.

[RFC2719] "Framework Architecture for Signaling Transport", IETF Informational RFC, 1999 年 10 月.

[RFC2960] “Stream Control Transmission Protocol”, IETF Proposed Standard RFC, 2000 年 10 月.

[RFC3057] “ISDN Q.921-User Adaptation Layer’, IETF, 2001 年 2 月.

[SCTP] “Stream Control Transmission Protocol”, IETF Proposed Standard RFC, 2000 年 10 月.

[IUA] “ISDN Q.921-User Adaptation Layer’, IETF Proposed Standard RFC, 2001 年 2 月.

[M3UA] “SS7 MTP3-User Adaptation Layer”, IETF Work in Progress, 2002 年 2 月.

[M2UA] “SS7 MTP2-User Adaptation Layer”, IETF Work in Progress, 2002 年 2 月.

[M2PA] “SS7 MTP2-User Peer-to-Peer Adaptation Layerer”, IETF Work in Progress, 2001 年 7 月.

[SUA] “SS7 SCCP-User Adaptation Layer”, IETF Work in Progress, 2002 年 2 月.

[ALLMAN99] "On Estimating End-to-End Network Path Properties", Allman, M. and Paxson, V., Proc. SIGCOMM'99, 1999.

[FALL96] “Simulation-based Comparisons of Tahoe, Reno, and SACK TCP”, Fall, K. and Floyd, S., Computer Communications Review, V. 26 N. 3, July 1996, pp. 5-21.

[SAVAGE99] "TCP Congestion Control with a Misbehaving Receiver", Savage, S., Cardwell, N., Wetherall, D., and Anderson, T., ACM Computer Communication Review, 29(5), October 1999.