

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

1. 背景・目的

OSSI3.0 に規定される IPDR Collector クライアントの開発には、IPDR/SP プロトコル仕様の理解が必要となる。

本 Tips では、IPDR のメッセージ構造、通信シーケンス、マルチセッション、テンプレートネゴシエーション、ケーパビリティ、キープアライブ、冗長構成について説明する。

2. 対象読者

システム設計者、プログラム開発者

3. 参考文献・関連文書

DOCSIS OSSI 1.1/2.0/3.0 (www.cablelabs.com)

IPDR/SP Protocol Specification Version 2.1, November 2004

4. その他

本 Tips 中の図表番号につき、OSSI からの抜粋には原文の番号をそのまま流用し、独自に作成した図表には”Tips-*”の形式で番号を付与した。

5. 最終更新日

2011 年 5 月 19 日

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6. 詳細

6.1 IPDR/SP メッセージ

IPDR/SP メッセージの一覧を Figure 6-1 に示す。

カテゴリ	ID	メッセージ	方向	必須	備考
接続・切断	0x05	CONNECT	双方向	*	
	0x06	CONNECT RESPONSE	双方向	*	
	0x07	DISCONNECT	双方向	*	
エラー	0x23	ERROR	双方向	*	
フロー・セッション制御	0x01	FLOW START	C->E	*	
	0x08	SESSION START	C<-E	*	
	0x03	FLOW STOP	C->E	*	
	0x09	SESSION STIOP	C<-E	*	
データ構造情報 (テンプレート) 交換	0x10	TEMPLATE DATA	C<-E	*	
	0x1a	MODIFY TEMPLATE	C->E		ネゴシエーション に使用
	0x1b	MODIFY TEMPLATE RESPONSE	C<-E		
	0x13	FINAL TEMPLATE DATA ACK	C->E	*	
	0x1d	START NEGOTIATION	C->E		ネゴシエーション に使用
	0x1e	START NEGOTIATION REJECT	C<-E		
データ	0x20	DATA	C<-E	*	
	0x21	DATA ACKNOWLEDGE	C->E	*	
要求/応答	0x30	REQUEST	双方向		DOCSIS OSS13.0 では使用されない
	0x31	RESPONSE	双方向		
状態管理	0x14	GET SESSION	C->E		
	0x15	GET SESSION RESPONSE	C<-E		
	0x16	GET TEMPLATES	C->E		
	0x17	GET TEMPLATES RESPONSE	C<-E		
	0x40	KEEP ALIVE	双方向	*	

Figure 6-1 IPDR/SP メッセージの一覧

6.1.1 IPDR メッセージヘッダー

各 IPDR メッセージは、共通のヘッダー構造を持つ。

IPDR メッセージのヘッダー構造を Figure 6-2 に示す。

項目名	データ型	繰返構造	説明
version	Char		IPDR プロトコルバージョン。 DOCSIS OSS13.0 では"2"。
messageId	Char		メッセージタイプ ID。 Figure 6-1 の ID 欄を参照。
sessionId	Char		セッション ID。 シングルセッション構成時、常に"0"。 マルチセッション構成時、Exporter が一意 ID を 付与。Collector のセッション分離に使用される。 マルチセッションについては、6.3 を参照。 以下のメッセージの場合には無視される。 CONNECT, CONNECT RESPONSE, DISCONNECT GET SESSIONS, GET SESSIONS RESPONSE KEEP ALIVE, ERROR
messageFlags	Char		メッセージフラグ。 将来用の予約フィールドであり、現行バージョン の IPDR/SP では使用しない ("0"固定)。
messageLen	Int		ヘッダーを含むメッセージ長(オクテット数)。

Figure 6-2 IPDR メッセージのヘッダー構造

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.2 CONNECT

CONNECT の構造を Figure 6-3 に示す。

項目名	データ型	繰返構造	説明
initiatorId	Int		メッセージ送信元の ID。
initiatorPort	Short		メッセージ送信元の TCP ポート。 4737 がデフォルト。
capabilities	Int		打診ケーパビリティ。 ケーパビリティについては、6.5 を参照。
keepAliveInterval	Int		キープアライブ間隔(秒)。 キープアライブについては、6.6 を参照。
vendorId	UTF8String		送信元のベンダーID。任意の文字列。

Figure 6-3 CONNCET の構造

CONNECT は、主に Collector が送信する。

予め設定した許容時間内に CONNECT RESPONSE を受信できずタイムアウトとなる場合、Collector は CONNECT RESPONSE 受信に成功するまで、CONNECT 再送を繰り返す。

6.1.3 CONNECT RESPONSE

CONNECT RESPONSE の構造を Figure 6-4 に示す。

項目名	データ型	繰返構造	説明
capabilities	Int		対応可能ケーパビリティ。 ケーパビリティについては、6.5 を参照。
keepAliveInterval	Int		キープアライブ間隔(秒)。 Connect メッセージと同一値を格納する。
vendorId	UTF8String		送信元のベンダーID。任意の文字列。

Figure 6-4 CONNCET RESPONSE の構造

CONNECT RESPONSE は、主に Exporter が送信する。

Collector は、Exporter から通知された対応可能ケーパビリティに従い、セッション構成とテンプレートネゴシエーションの有無を設定する。

ケーパビリティの詳細については、6.3 を参照されたい。

6.1.4 DISCONNECT

DISCONNECT は、IPDR メッセージヘッダーのみで構成される。

DISCONNECT は、Connection の終了に使われる。

DISCONNECT は、Exporter と Collector の双方から送信される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.5 ERROR

ERROR の構造を Figure 6-5 に示す。

項目名	データ型	繰返構造	説明
timeStamp	Int		障害の検知時刻。 1970 年 1 月 1 日エポックの経過秒数。
errorCode	Short		障害コード。 0~16383 は汎用、16384~32767 はセッション依存の障害コード領域。0~255 と 16384~16639 は IPDR/SP が予約、それ以外はベンダー定義。* 主に以下が使われる。 0: キープアライブのタイムアウト 1: ケーパビリティへの無効値指定 2: 順序的にあり得ないメッセージを受信 3: メッセージデコードエラー 4: プロセス停止 *DOCSIS OSSI 3.0 Annex IV のメッセージ例では、キープアライブのタイムアウトを表わす障害コード=32767 となっており、IPDR/SP の定義内容と矛盾するため、実機での調査要。
description	UTF8String		Error Code の内容の文字列表現。

Figure 6-5 ERROR の構造

ERROR は、Exporter と Collector の双方から送信される。

Collector と Exporter のいずれかがキープアライブのタイムアウトを検知時、障害コード=0 の ERROR を送信する。但し、キープアライブがタイムアウトする場合には、ERROR が相手側に届かない可能性が高い。この場合の障害コード=0 は、主に検知元による通信障害の記録に使用される。

障害コード=1 の場合については、6.5.4 を参照されたい。

IPDR/SP の通信順序的にあり得ないメッセージを受信時、受信側が障害コード=2 の ERROR を相手側に送信する。

メッセージヘッダーとメッセージ本体構造、格納パラメータの何れかにてデコードエラーが発生時、受信側が障害コード=3 の ERROR を相手側に送信する。

障害コード=0~4 の何れかを受信時、受信側は Connection を終了・再開する。

6.1.6 FLOW START

FLOW START は、IPDR メッセージヘッダーのみで構成される。

Exporter が Collector から FLOW START を受信しない場合、Exporter は Collector を特定セッションの対象から外し、SESSION START を送信しない。

例えば Collector の負荷が高い場合、一部のセッションを FLOW STOP し、代わりに他の Collector を FLOW START して負荷を分散する、等の応用が考えられる。

FLOW START は、常に Collector から送信される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.7 SESSION START

SESSION START の構造を Figure 6-6 に示す。

項目名	データ型	繰返構造	説明
exporterBootTime	Int		Exporter 起動時刻。 セッション開始時の CMTS の SysUpTime。
firstRecordSequenceNumber	Long		最初のレコード番号。0 から開始。 Primary Collector が正常動作時、第 1 レコードを表わす 0 が格納される。 Primary Collector に障害が発生し、Secondary 以降の Collector に送信対象が切り替わる場合、送信を再開する先頭のレコード番号が格納される。
droppedRecordCount	Long		破棄レコード数。 Primary Collector に障害が発生し、Secondary 以降の Collector に送信対象が切り替わったものの、Exporter 側のバッファオーバーフロー等の理由により、いずれの Collector にも送信されずに破棄されたレコードが発生時 (gap)、対象レコードの総数が格納される。 上記以外の場合、0 が格納される。
primary	Boolean		Primary 識別子。 Exporter が対象 Collector を Primary 登録している場合は true、それ以外の場合は false が格納される。
ackTimeInterval	Int		送受信確認インターバル(秒)。
ackSequanceInterval	Int		送受信確認インターバル(レコード数)。
documentId	Char		16 バイト UUID 形式による、CMTS と収集タイミングの一意識別子。 例: C8A93279-0000-0000-0000-0002FC84F870 書式は以下の通り。 <32bit UTC timestamp>-0000-0000-0000- <48bit MAC address> 32bit UTC timestamp は、IPDRDoc 生成時刻の 8 文字の 16 進数表現。48bit MAC address は、CMTS の MAC address。その他は 0。 Primary Collector に障害が発生し、Secondary 以降の Collector に送信対象が切り替わる場合、Collector による受信後のレコード群照合キーとして使われる。

Figure 6-6 SESSION START の構造

SESSION START は、常に Exporter から送信される。

6.1.8 FLOW STOP

FLOW STOP の構造を Figure 6-7 に示す。

項目名	データ型	繰返構造	説明
reasonCode	Short		理由コード。 0: 正常停止 1: 異常停止
reasonInfo	UTF8String		Reason Code の内容の文字列表現。

Figure 6-7 FLOW STOP の構造

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

理由コード=1 は、Collector が例外を検出時に Connection を終了・再接続する場合に使われる。Ad-Hoc セッションの最後、Collector プログラムの終了操作、負荷分散時の FLOW STOP は、理由コード=0 となる。

FLOW STOP は、常に Collector から送信される。

6.1.9 SESSION STOP

SESSION STOP の構造を Figure 6-8 に示す。

項目名	データ型	繰返構造	説明
reasonCode	Short		理由コード。 0: 対象セッションの最終データ(Ad-Hoc 時) 1: 他 Collector への委譲 2: Exporter の問題による異常終了 3: 輻輳の発生 4: 再ネゴシエーション要 5: ネゴシエーション開始応答 6: IPDRDoc の最終データ(常時接続時) 7: テンプレートの更新
reasonInfo	UTF8String		Reason Code の内容の文字列表現。

Figure 6-8 SESSION STOP の構造

Ad-Hoc セッションの正常終了時、理由コード=0 の SESSION STOP が送信される。Time-Interval セッションと Time-based Event セッションの正常終了時、理由コード=6 の SESSION STOP が送信される。

現用系 Collector との間に通信障害が発生時、Exporter は現用系 Collector に理由コード=1 の SESSION STOP を送信後、待機中の予備系 Collector に Primary=false の SESSION START を送信し、実行中のデータ送信処理を予備系 Collector に委譲後、現用系 Collector との Connection を終了する。現用系 Collector との通信が復旧時、Exporter は予備系 Collector に理由コード=1 の SESSION STOP を送信後、待機中の現用系 Collector に Primary=true の SESSION START を送信し、実行中のデータ送信処理を委譲する。Exporter の問題によるデータ送信処理の異常終了時、Exporter は通信中の Collector に理由コード=2 の SESSION STOP を送信する。この場合には Collector 側で、SESSION STOP 以降のデータが欠損する。Exporter の問題解消時における Exporter 内キャッシュの残データの再送動作の有無は、CMTS の実装次第である。

理由コード=4, 5 の SESSION STOP は、セッションの最中にテンプレートネゴシエーションを実行した場合に使用される。セッションの最中に Exporter 内のテンプレートが更新された場合、Exporter から Collector に対し、理由コード=7 の SESSION STOP が送信される。その後、GET TEMPATES によりテンプレートが更新され、新たなコンフィグ ID での SESSION START 送信によりセッションが再開される。但し基本的には、セッション中の動的なテンプレート更新よりも、一旦 Collector を終了しテンプレートを更新後、Collector を再起動する静的な運用の方が確実である。

SESSION STOP は、常に Exporter から送信される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.10 TEMPLATE DATA

TEMPLATE DATA の構造を Figure 6-9 に示す。

項目名	データ型	繰返構造	説明
configId	Short		コンフィグ ID。 テンプレートのグループ概念。 マルチテンプレート構成時に一意 ID を付与。 シングルテンプレート構成時、0 固定。 マルチテンプレートについては 6.4.3 を参照。
Flags	Char		ネゴシエーションフラグ。 0: 不可, 1: 可 ケーバリティでネゴシエーションが許可され、 本フラグが 0 となる場合、対象テンプレートのネ ゴシエーションが禁止される。この場合、Collector はネゴシエーションを行わない。
templateId	Short		テンプレート ID。 Exporter 内でのみ一意であり、複数 Exporter 間では 一意性が保証されない。
schemaName	UTF8String		スキーマ名。
typeName	UTF8String		サービス定義名。
typeId	Int	*	データ型。IPDR にて定義。
fileId	Int	*	列番号。1 から追番指定。
fileName	UTF8String	*	項目名。
isEnabled	Boolean	*	項目有効フラグ。テンプレートネゴシエーション 時に使用。 0: 不使用(送信を抑制), 1: 使用

Figure 6-9 TEMPLATE DATA の構造

TEMPLATE DATA は、常に Exporter から送信される。

6.1.11 FINAL TEMPLATE DATA ACK

FINAL TEMPLATE DATA ACK は、IPDR メッセージヘッダーのみで構成される。

FINAL TEMPLATE DATA ACK は、常に Collector から送信される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.12 DATA

DATA の構造を Figure 6-10 に示す。

項目名	データ型	繰返構造	説明
templateId	Short		テンプレート ID。6.1.10 を参照。
configId	Short		コンフィグ ID。6.1.10 を参照。
Flags	Char	*	レコードフラグ。 0: 重複なし 1: 重複レコード(他の Collector に既に送信済)
sequenceNum	Long	*	レコード番号。SESSION START メッセージ中の firstRecordSequenceNumber から開始される。
dataRecord<>	Opeque	*	データレコード。 XDR 形式にエンコードされたデータ。

Figure 6-10 DATA の構造

Collector による DATA の受信時、内容にデコードエラーが検出された場合、Collector は Exporter に対し、障害コード=3 の ERROR を送信する。

障害コード=3 の ERROR 受信時、Exporter は対象 Collector に理由コード=1 の SESSION STOP を送信後、対象 Collector との Connection を終了し、データ送信処理を別の Collector に切り替える。この場合、再送データの送信済レコードフラグに 1 が格納される。

Collector は、理由コード=0 or 6 の SESSION STOP 受信により、対象セッションの全データの受信完了を判断する。

DATA は、常に Exporter から送信される。

6.1.13 DATA ACKNOWLEDGE

DATA ACKNOWLEDGE の構造を Figure 6-11 に示す。

項目名	データ型	繰返構造	説明
configId	Short		コンフィグ ID。6.1.10 を参照。 DATA との対応付けに使用。
sequenceNum	Long		レコード番号。 正常受信データの最後のレコード番号を格納。

Figure 6-11 DATA ACKNOWLEDGE の構造

Collector は、Exporter が SESSION START で指定した 2 種類の送受信確認インターバルに従い、Exporter に DATA ACKNOWLEDGE を送信する。Exporter は、自身が通知した送受信確認インターバルにマージンを加えた時間、ないしは指定レコード数を超えて DATA ACKNOWLEDGE が返らない場合、対象 Collector に理由コード=1 の SESSION STOP を送信後、データ送信処理を別の Collector に切り替える。この場合、再送データブロックの送信済レコードフラグに 1 が格納される。

DATA ACKNOWLEDGE は、常に Collector から送信される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.14 REQUEST

REQUEST の構造を Figure 6-12 に示す。

項目名	データ型	繰返構造	説明
templateId	Short		テンプレート ID。6.1.10 を参照。
configId	Short		コンフィグ ID。6.1.10 を参照。
flags	Char		リクエストフラグ。 1: 重複レコード(他の Collector に既に送信済) 2: RESPONSE 要
requestNumber	Long		リクエスト番号。 IPDRDoc 内でレコードを一意に指定。
dataRecord	Array		XDR 形式データ。

Figure 6-12 REQUEST の構造

6.1.15 RESPONSE

RESPONSE の構造を Figure 6-13 に示す。

項目名	データ型	繰返構造	説明
templateId	Short		テンプレート ID。6.1.10 を参照。 REQUEST との対応付けに使用。
configId	Short		コンフィグ ID。6.1.10 を参照。 REQUEST との対応付けに使用。
flags	Char		結果フラグ。 0: 失敗, 1: 成功
requestNumber	Long		リクエスト番号。 REQUEST との対応付けに使用。

Figure 6-13 RESPONSE の構造

6.1.16 GET SESSION

GET SESSION の構造を Figure 6-14 に示す。

項目名	データ型	繰返構造	説明
requestId	Short		リクエスト ID。

Figure 6-14 GET SESSION の構造

6.1.17 GET SESSION RESPONSE

GET SESSION RESPONSE の構造を Figure 6-15 に示す。

項目名	データ型	繰返構造	説明
requestId	Short		リクエスト ID。GET SESSION との対応付けに使用。
sessionId	Char		セッション ID。
reserved	Char		IPDR/SP にて予約。現在は未使用。
sessionName	UTF8String		セッション名。
sessionDescription	UTF8String		セッション説明。
ackTimeInterval	Int		送受信確認インターバル(秒)。6.1.7 を参照。
ackSequenceInterval	Int		送受信確認インターバル(レコード数)。 6.1.7 を参照。

Figure 6-15 GET SESSION RESPONSE の構造

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.1.18 GET TEMPLATES

GET TEMPLATES の構造を Figure 6-16 に示す。

項目名	データ型	繰返構造	説明
requestId	Short		リクエスト ID。

Figure 6-16 GET TEMPLATES の構造

6.1.19 GET TEMPLATES RESPONSE

GET TEMPLATES RESPONSE の構造を Figure 6-17 に示す。

項目名	データ型	繰返構造	説明
requestId	Short		リクエスト ID。 GET TEMPLATES との対応付けに使用。
configId	Short		コンフィグ ID。6.1.10 を参照。

Figure 6-17 GET TEMPLATES RESPONSE の構造

6.1.20 KEEP ALIVE

KEEP ALIVE は、IPDR ヘッダーのみで構成される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.2 IPDR セッションシーケンス

6.2.1 Time-Interval セッション

Time-Interval セッションのシーケンスを Figure 6-18 に示す。

順序	処理(メッセージ)	方向	説明
1	TCP Connect	C->E	TCP ソケット開設。
2	IPDR Connect	C->E	IPDR 接続要求。
3	IPDR Connect Response	C<-E	IPDR 接続応答。
4	Keep Alive	C->E	Collector によるキープアライブ開始。
5	Keep Alive Ack	C<-E	キープアライブ応答。
6	IPDR Flow Start	C->E	フロー開始。
7	IPDR Template Data	C<-E	テンプレート情報受信。
8	IPDR Final Template Ack	C->E	最終テンプレート情報の受信確認。
9	IPDR Session Start	C<-E	セッション開始。
10	IPDR Data	C<-E	データ受信。
11	IPDR Data Ack	C->E	データ受信確認。
12	IPDR Session Stop	C<-E	セッション終了。

Figure 6-18 Time-Interval セッションシーケンス

本シーケンスでは、Exporter によるキープアライブ動作が省略されている。

Time-Interval セッションでは、1~12 の完了後、9~12 を繰り返す。

4~5 は、他シーケンスとは無関係に、定期的に行われる。

6.2.2 Event-Based セッション

Event-Based セッションのシーケンスを Figure 6-19 に示す。

順序	処理(メッセージ)	方向	説明
1	TCP Connect	C->E	TCP ソケット開設。
2	IPDR Connect	C->E	IPDR 接続要求。
3	IPDR Connect Response	C<-E	IPDR 接続応答。
4	Keep Alive	C->E	Collector によるキープアライブ開始。
5	Keep Alive Ack	C<-E	キープアライブ応答。
6	IPDR Flow Start	C->E	フロー開始。
7	IPDR Template Data	C<-E	テンプレート情報受信。
8	IPDR Final Template Ack	C->E	最終テンプレート情報の受信確認。
9	IPDR Session Start	C<-E	セッション開始。* * Open-Ended Session では初回時のみ。
10	IPDR Data	C<-E	データ受信(随時・レコード単位)。
11	IPDR Data Ack	C->E	データ受信確認(随時・レコード単位)。
12	IPDR Session Stop	C<-E	セッション終了。* * Open-Ended Session では送信されない。

Figure 6-19 Event Based セッションシーケンス

本シーケンスでは、Exporter によるキープアライブ動作が省略されている。

Event-Based セッションには、Time-Based と Open-Ended の 2 種類がある。

Time-Based は、Collector が一定期間イベントを蓄積後、定期的 IPDRDoc を生成する。Open-Ended は、単一イベントの受信時点で即座に処理を実行し、IPDRDoc を生成しない。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

Time-Based のシーケンスは、Time-Interval に近い。すなわち Time-Based は Time-Interval 同様、1~12 の完了後、9~12 を繰り返す。Collector は SESSION STOP の受信を契機に IPDRDoc を生成する。

Open-Ended Event の場合、SESSION START は Collector 起動時に 1 回だけ受信され、以降は受信されない。SESSION STOP は受信されない。

但し、冗長構成時に Exporter が Primary 以外の Collector にセッションを切り替える場合、及び、Primary Collector に処理を戻す場合、SESSION STOP と SESSION START が都度使われる。

4~5 は、他シーケンスとは無関係に、定期的に行われる。

6.2.3 Ad-Hoc セッション

Ad-Hoc セッションのシーケンスを Figure 6-20 に示す。

順序	処理(メッセージ)	方向	説明
1	TCP Connect	C->E	TCP ソケット開設。
2	IPDR Connect	C->E	IPDR 接続要求。
3	IPDR Connect Response	C<-E	IPDR 接続応答。
4	Keep Alive	C->E	Collector によるキープアライブ開始。
5	Keep Alive Ack	C<-E	キープアライブ応答。
6	IPDR Flow Start	C->E	フロー開始。
7	IPDR Template Data	C<-E	テンプレート情報受信。
8	IPDR Final Template Ack	C->E	最終テンプレート情報の受信確認。
9	IPDR Session Start	C<-E	セッション開始。
10	IPDR Data	C<-E	データ受信。
11	IPDR Data Ack	C->E	データ受信確認。
12	IPDR Session Stop	C<-E	セッション終了。
13	IPDR Flow Stop	C->E	フロー停止。
14	IPDR Disconnect	C->E	IPDR 切断。
15	TCP Disconnect	C->E	TCP 切断。

Figure 6-20 Ad-Hoc セッションシーケンス

Ad-Hoc では、Connection の使い回しを行わず、セッションの度に都度 Connection を接続・切断する。このため、1~15 が 1 回のみ実行される。

Ad-Hoc は Connection を維持する必要がないため、キープアライブ動作は原則的に不要だが、実際の動作は Exporter, Collector の実装次第である。本 Tips では Collector クライアントのプログラム共通化の観点から、他セッション同様のキープアライブ動作を定義した。

4~5 は、他シーケンスとは無関係に、定期的に行われる。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.3 マルチセッション

6.3.1 複数セッション構成

例えば日次で Ad-Hoc により全レコードを一括取得し、以降は Event-Based で差分のみ取り込む場合、一組の Exporter-Collector 間に、複数のセッションを構成する。また、SMS/Billing や NMS など、複数の上位アプリケーションが同じ Collector を共有する場合にも、同様に複数セッションを構成する。IPDR/SP では一組の Exporter-Collector 間への複数セッションの構成方法として、マルチセッションとシングルセッションの 2 種類を定義している。

6.3.2 マルチセッションとシングルセッション

マルチセッション(Multi-Session)は、Exporter-Collector 間通信で、複数のセッションを一つの TCP コネクションに収容し、システムリソースの消費を低減する構成であり、Multiple Sessions per Single Connection の短縮表現である。一方でシングルセッション(Single-Session)は、TCP コネクションとセッションが一つ一つで対応する Single Session per Single Connection, Multiple Connections 構成であり、セッションと同数のコネクションを使用する。複数セッションの分離には、マルチセッションでは IPDR メッセージ中の Session ID が、シングルセッションでは TCP ポート番号が使われる。

6.3.3 DOCSIS OSSI の現状

シングルセッションで複数のセッションを構成する場合、複数の TCP ポートを使うため、Exporter, Collector のシステムリソース消費が増大する。また、ネットワーク経路上に L3/L4 S/W や Firewall があると、セキュリティの設定管理が煩雑となる。更に、これらの設定が不適切だと、セキュリティホールとなりかねない。マルチセッションでは一つの TCP ポートを共有するため、理論上は上述の問題を回避できるが、実際の Exporter-Collector 間にはプライベートなイントラネットが多く、シングルセッションがセキュリティ上の問題を生じる可能性は比較的少ない。このため DOCSIS OSSI では、開発・実装の容易なシングルセッションを MUST として定義し、マルチセッションを CMTS ベンダーの実装依存としている。

マルチセッションへの対応方針・状況は、CMTS メーカー・型式により異なる。

6.3.4 マルチセッションの動作設定

まず、一つの TCP コネクションに複数のセッションを関連付ける形で、Exporter に予め設定を登録しておく。この状態で Collector が Exporter に対し、ケーパビリティに Multi session=enable を指定した CONNECT を送信する。CONNECT を受信した Exporter は、Collector に CONNECT RESPONSE を返信するが、この際に Exporter が返信するケーパビリティの Multi-session 指定が enable の場合にはマルチセッションが、disable の場合にはシングルセッションが適用される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.4 テンプレートネゴシエーション

6.4.1 テンプレート交換

テンプレートは、DOCSIS OSSI のサービス定義=テーブル情報毎に定義される、IPDR データ構造の定義情報である。Exporter-Collector はデータの送受信に先立ち、予めテンプレートを交換・共有し、データのエンコード・デコードを効率的に行う。

テンプレート交換は、DOCSIS OSSI の改版等に伴うサービス定義の変更により、CMTS のファームウェアを更新する必要がある場合などに便利である。この場合、Exporter の更新はテンプレート交換により、Collector に自動的に反映されるため、Collector 設定の手動更新が不要となり、設定誤りや設定忘れによるデコードエラーを回避できる。

6.4.2 テンプレートネゴシエーション

大量の情報を送受信する際に、Collector に不要な項目を Exporter が予め除外できれば、送受信されるデータの総量が低減され、通信時間を短縮できる。このための一連のやり取りを、IPDR/SP では「テンプレートネゴシエーション」として定義している。

テンプレートネゴシエーションでは、データ送受信に先立ち、Collector が Exporter に不要な項目を通知する。通知は、テンプレート項目属性情報の isEnabled フラグへの false 指定により行われる。これを Exporter が了解・承諾するとネゴシエーションが成立し、以降は Exporter により送信データから対象項目が除外される。

6.4.3 DOCSIS OSSI の現状

テーブル構造の大量の情報を CMTS から取り出す場合、レコード数が CMTS 負荷 (CPU・メモリ消費) に直接的な影響を及ぼすのに対し、項目数が CMTS 負荷に及ぼす影響は比較的少なく、むしろネットワーク負荷 (通信時間・占有率) に影響する。しかしながら一般的に、Collector-Exporter 間のネットワークは十分な通信性能を持つ場合が多く、万が一ボトルネックとなる場合にも、比較的 low コストで改善できる。

従って、テンプレートネゴシエーションによる効果は、現実にはかなり限定的である。

このため DOCSIS OSSI では、テンプレート交換を MUST として定義し、テンプレートネゴシエーションを SHOULD 定義としている。但し、比較的項目数の多い従量制課金向けの SAMIS サービス定義に限り、項目数を絞った SAMIS Type-2 を別に定義し、ネットワーク負荷の削減を図っている。

テンプレートネゴシエーションの対応方針・状況は、CMTS メーカー・型式により異なる。なお IPDR/SP と DOCSIS OSSI3.0 には、6.3 のマルチセッションと同様、一つのセッションに複数のテンプレートを構成する「マルチテンプレート」概念が存在するが、DOCSIS OSSI では安定性の観点から推奨されていない。

マルチテンプレートの場合、各テンプレートは、IPDR メッセージ中のテンプレート ID により分離される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.5 ケーパビリティ

6.5.1 概念定義

CONNECT / CONNECT RESPONSE メッセージ中の"capabilities"すなわちケーパビリティは、enable/disable 設定を持つ 4 属性により構成されるパラメータであり、幾つかの機能概念に対する Exporter, Collector 双方の対応能力を表わす。

ケーパビリティは 32 ビット値であり、bit 配列の LSB 側から順に各機能と対応付けられ、0 が false、1 が true を表す。2011 年 4 月現在、DOCSIS OSSI3.0 では 1~4bit までが定義され、5~32bit は未使用=0 である。各有効 bit には以下の 4 属性が格納される。

- Bit 1: STRUCTURES(構造化レコード)
- Bit 2: MULTISESSION(マルチセッション)
- Bit 3: TEMPLATE NEGOTIATION(テンプレートネゴシエーション)
- Bit 4: REQUEST RESPONSE(要求/応答)

ケーパビリティは 10 進変換され、CONNECT により送受信される。

なお IPDR では、UDP による IPDR/SP の送受信パラメータ設定などを交換する XML 形式ファイルである Capability files にも同じ"Capability"の単語が使われているが、CONNECT 中のパラメータである本ケーパビリティとは内容・意味が異なる。

6.5.2 各属性の内容

(1) STRUCTURES(構造化レコード)

IPDR 3.5 以降で定義された配列型項目(単一レコード内のレコード入れ子構造)のサポート可否を表す。IPDR3.1.1 のフラットな非構造化レコードのみ取り扱う場合、disable を指定する。

DOCSIS OSSI 3.0 では、IPDR/SP プロトコルバージョンに 3.5.1 が必須指定されており、かつ CPE List 構造の表現に、STRUCTURE の enable 設定が必須である。

従って DOCSIS OSSI3.0 の場合、STRUCTURE ビットは 1(enable)固定となる。

(2) MULTISESSION(マルチセッション)

シングルセッション時 0(disable)、マルチセッション時 1(enable)となる。

シングルセッション/マルチセッションの詳細については、6.3 を参照されたい。

(3) TEMPLATE NEGOTIATION(テンプレートネゴシエーション)

無効時 0(disable)、有効時 1(enable)となる。

テンプレートネゴシエーションの詳細については、6.4 を参照されたい。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

(4) REQUEST RESPONSE(要求/応答)

Collector のフェイルオーバー動作など、アプリケーションの特殊要件から Collector-Exporter 間に IPDR の通信シーケンスが必要となる場合に使用する。

無効時 0(disable)、有効時 1(enable)となる。

機能を使わない場合、REQUEST RESPONSE ビットは 0(disable)固定となる。

通常の構成では、無効で差し支えない。

6.5.3 ケーパビリティ構成

STRUCTURE ビットを 1(enable)固定、REQUEST RESPONSE ビットを 0(disable)固定とした場合、ケーパビリティ構成は以下の 4 通りとなる。

シングルセッション・テンプレートネゴシエーション無

0000 0000 0000 0000 0000 0000 0000 0001 Integer 表現=1

マルチセッション・テンプレートネゴシエーション無

0000 0000 0000 0000 0000 0000 0000 0011 Integer 表現=3

シングルセッション・テンプレートネゴシエーション有

0000 0000 0000 0000 0000 0000 0000 0101 Integer 表現=5

マルチセッション・テンプレートネゴシエーション有

0000 0000 0000 0000 0000 0000 0000 0111 Integer 表現=7

現状では、効果に乏しいテンプレートネゴシエーションは「無」固定、通常時はシングルセッション(Integer 表現=1)、セキュリティ上の必要がある場合に限りマルチセッション(Integer 表現=3)を選択する考え方が無難である。

6.5.4 ケーパビリティの動作設定

IPDR/SP では Connection 開設時、Collector が Exporter に使いたい設定を打診する。打診は、CONNECT 中のケーパビリティ値により行われる。Exporter は Collector からの打診ケーパビリティの内容を解析し、対応可能であれば同一値、対応不可であれば対応可能な値に置換後、CONNECT RESPONSE により Collector に対応可能なケーパビリティ値を返信する。CONNECT 中のケーパビリティ値が範囲外の場合、Collector は Exporter に ERROR を送信し、処理を終了する。値が範囲内の場合、対象となる Exporter-Collector 間コネクションで、同じ値のケーパビリティが共有・反映される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.6 キープアライブ

キープアライブとは、ネットワーク接続の有効性を確認するために、送受信の主体同士が定期的に行う、相互の生存確認手法である。

ネットワーク機器には、通信相手から長時間通信がない場合、正常待機と異常停止とを識別する直接的な方法がない。常設のネットワーク接続を使い回しながらデータを送受信する通信シーケンスの場合、通信終了の合図がないまま相手側が異常終了すると、永久に接続を閉じることができない。このため多くのプロトコルでは、不要時にも一定間隔で「生きている」(alive)事を相手に伝え続けるキープアライブ動作を行う。キープアライブ信号が一定期間以上届かない場合、一旦接続を終了して再接続を試みるか、接続を待機系に切り替える等の処理が実行される。

IPDR/SP では、キープアライブをメッセージ ID=0x40 として定義し、Exporter と Collector の双方からキープアライブ動作を行う旨定めている。

キープアライブは原理上、Connection を常設する Time-Interval と Event-Based のセッションタイプで必須であり、Connection を都度接続・切断する Ad-Hoc には不要である。

キープアライブは、各 IPDR/SP メッセージの返信タイムアウト、データ送受信時の Ack のタイムアウトとは関係なく、Exporter-Collector 間で常時実行される。

キープアライブには、IPDR/SP の原理上、以下の 2 種類が存在する。

6.6.1 Exporter からのキープアライブ

同一セッションに複数の Collector を構成時、Exporter は全 Collector に対し、定期的キープアライブを実行する。Exporter が優先順位の最も高い Collector で通信障害=キープアライブのタイムアウトを検知時、Exporter は対象 Collector の Connection を一旦終了し、Collector からの再接続を待つ。対象 Collector が復旧しない内にデータを送信する必要が生じた場合(データ送信中のキープアライブタイムアウトを含む)、Exporter は次順の Collector とセッションを開始する。Exporter が Collector の正常復旧を確認時、次回のセッションから復旧した Collector に処理を戻す。

6.6.2 Collector からのキープアライブ

Collector は Exporter に対し、定期的キープアライブを実行する。

Collector が Exporter との通信障害=キープアライブのタイムアウトを検知時、Collector は対象 Exporter の Connection を一旦終了し、Exporter への再接続を試みる。

再接続の試行は、比較的短い間隔(数秒程度)で、繰り返し実行される。

OPEN STM Tips

ケーブルネットワークにおける IPDR プロトコルの活用 (その 2)

6.7 Collector 冗長構成

IPDR/SP では、Collector の冗長構成により、システムの可用性を向上できる。

Exporter-Collector 間の通信障害時、Exporter は通信対象の Collector を切り替える。

IPDR/SP では、Exporter から送信されたデータを Collector が照合・取捨選択・再結合し、可能な限り完全な IPDR Doc を生成する。

受信データブロック間にレコード重複が発生時、Collector 側が重複レコードの抽出・排除を試みる。

Collector は、SESSION START から取得した各パラメータ(firstRecordSequenceNumber、droppedRecordCount、documentId)に基づき、全データの受信後に、IPDR Doc の再構成を試みる。

6.8 考察

監視系アプリケーションのデファクトともいえる SNMP と IPDR/SP を比較した場合、IPDR/SP は Collector 冗長構成時の CMTS への負荷が SNMP より低く、通信障害による欠損レコードの復元方法が考慮されている。

特に、大量レコードを定期的・低負荷かつ確実に CMTS から取得する必要のある、従量制課金情報の収集用途には、IPDR/SP メッセージの構造・特性は有効である。

IPDR/SP の長所である高信頼性を十分に活用するには、IPDR/SP のプロトコル特性を活かした各冗長 Collector の分散配置（自然災害も考慮し地理的に分散させる等）、キープアライブ設定（リトライ・タイムアウト等）、ストレージの配置についても考慮すべきである。

以上