

NoraVR (Nora Virtual Repeater)

Specification

目次

1. 概要	_____	P03
2. イメージブロック図	_____	P04
3. 動作フローイメージ	_____	P05
4. 通信プロトコル仕様	_____	P06
① 基本コマンドフレーム構成	_____	P06
② 共通事項	_____	P07
③ 通信動作シーケンス	_____	P09
1. ログインシーケンス	_____	P09
2. ログアウトシーケンス	_____	P09
3. 接続維持シーケンス	_____	P10
4. 設定更新シーケンス	_____	P10
5. リフレクタリンク状態の要求シーケンス	_____	P11
6. リフレクタリンク状態の通知シーケンス	_____	P11
7. アップリンク音声シーケンス	_____	P12
8. ダウンリンク音声シーケンス	_____	P12
④ コマンドタイプ別フィールド構成	_____	P13
1. ACK_____	_____	P13
2. NAK_____	_____	P13
3. LOGOUT__	_____	P13
4. LOGINUSR	_____	P14
5. LOGIN_CC	_____	P14
6. LOGIN_HS	_____	P14
7. LOGINACK	_____	P15
8. CONFSET_	_____	P16
9. PING_____	_____	P16
10. PONG_____	_____	P16
11. VTPCM__	_____	P17
12. VTOPUS__	_____	P18
13. VTAMBE__	_____	P19
14. RLINKGET	_____	P20
15. RLINK__	_____	P20
5. 改版履歴	_____	P22
6. 連絡先	_____	P22

余白

NoraVR(Nora Virtual Repeater)仕様

1. 概要

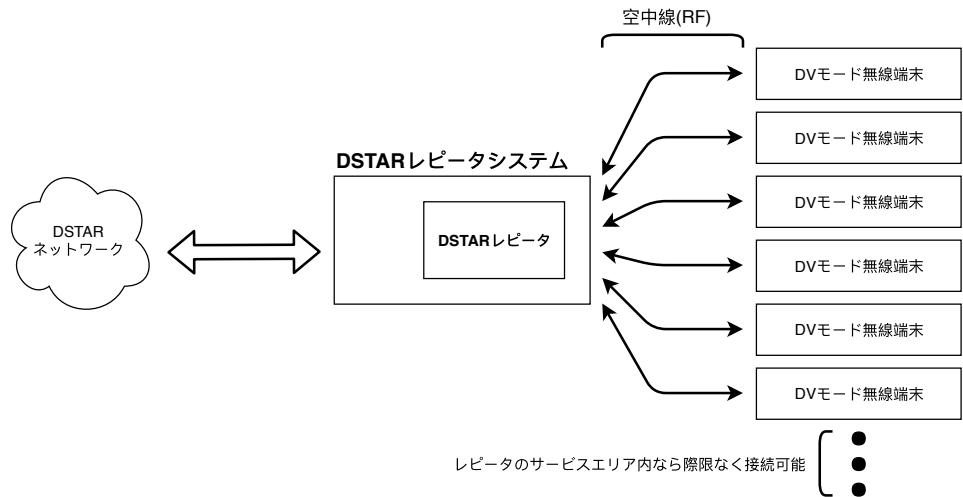
一般的なDSTARレピータが空中線(RF)で構成されているのに対して、公衆回線網上に「仮想的な」レピータシステム(VirtualRepeater)を形成する機能を提供する。リフレクターとしても機能し、既存の機能に加えてコールサインルーティングにも対応する。この仕様書は、ユーザー側でクライアントサイドを開発・設計する為のNoraVirtualRepeater(以下NoraVR)システム(サーバー・クライアント)の詳細仕様を示すものである。

- 【機能・特徴】**
- 仮想的なレピータシステム(=リフレクター同等)としての機能
 - TO(コールサインルーティング)を使用したゲート越え機能
 - ログイン時にパスワードを設定可能
 - 低ビットレートに対応し低速回線でも音声破綻しにくい
 - ユーザー側が音声コーデックを選択可能(帯域高 PCM<->Opus 64k<->Opus 24k<->Opus 8k 帯域低)
 - AMBEパススルーに対応(サーバー側のAMBEエンコード・デコードをパススルー)
 - パケット通信に対応(DSTAR仕様のファストデータ・スローデータ使用)
 - ユーザーがクライアント(端末)側を自由に開発・設計可能+通信ライブラリを公開
 - エコバック機能(送信した音声そのままオウム返しされる)
 - ログイン時にRF出力機能を有するかサーバーに宣言する事でRFが出力される端末を選別可能(法規対策)

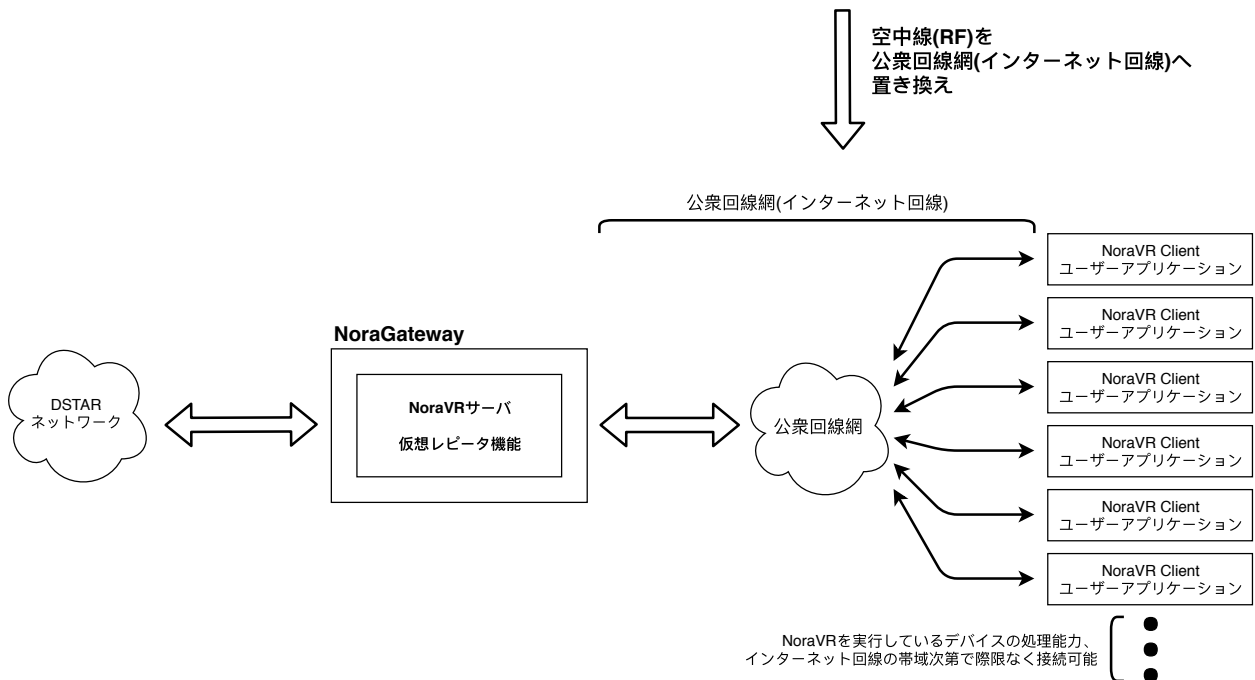
ユーザー側から見て一般的なレピータシステムと同等の機能を備えており、NoraGatewayで提供されている以下の機能を含め全て使用可能

- 接続ユーザー同士の山掛け交信
- リフレクターへの接続(DExtra/DCS/DPlus/JARL MultiForward Link)
- ゲート越え交信(コールサイン指定・ゲート指定)
- 制御コマンドによるNoraGatewayの制御(__RLMACEなど)

サーバー側(NoraVR)はNoraGatewayに内蔵されており、DSTAR関連のアプリケーションを自作する際に障壁となるAMBEエンコード・デコードはサーバー側で処理される。これにより、クライアント側でAMBEを扱う必要がなくなり、比較的容易にユーザー側が自由に開発・設計する事が可能となっている。



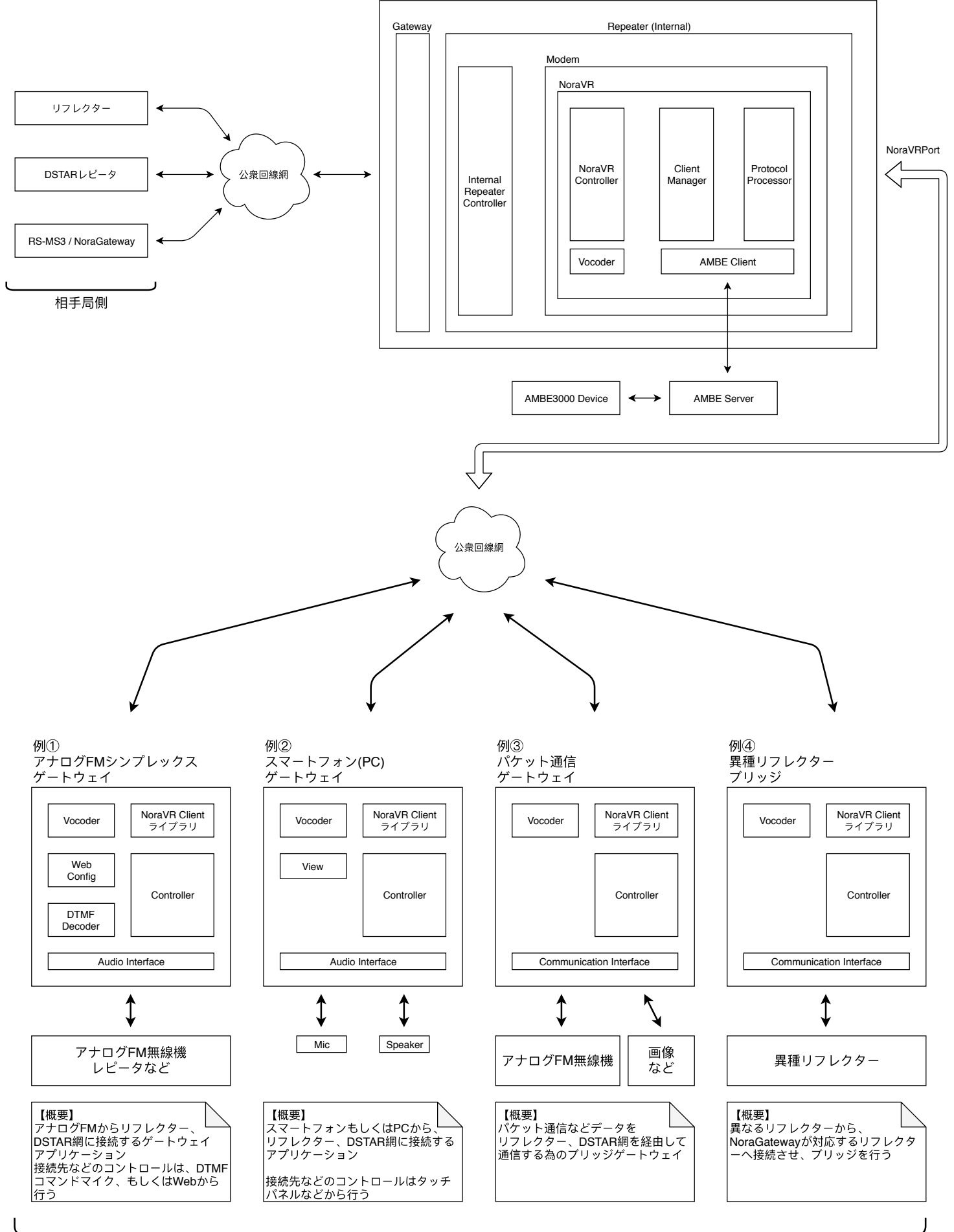
図① 既存のDSTARレピータシステム(抜粋)



図② NoraVR仮想レピータシステム

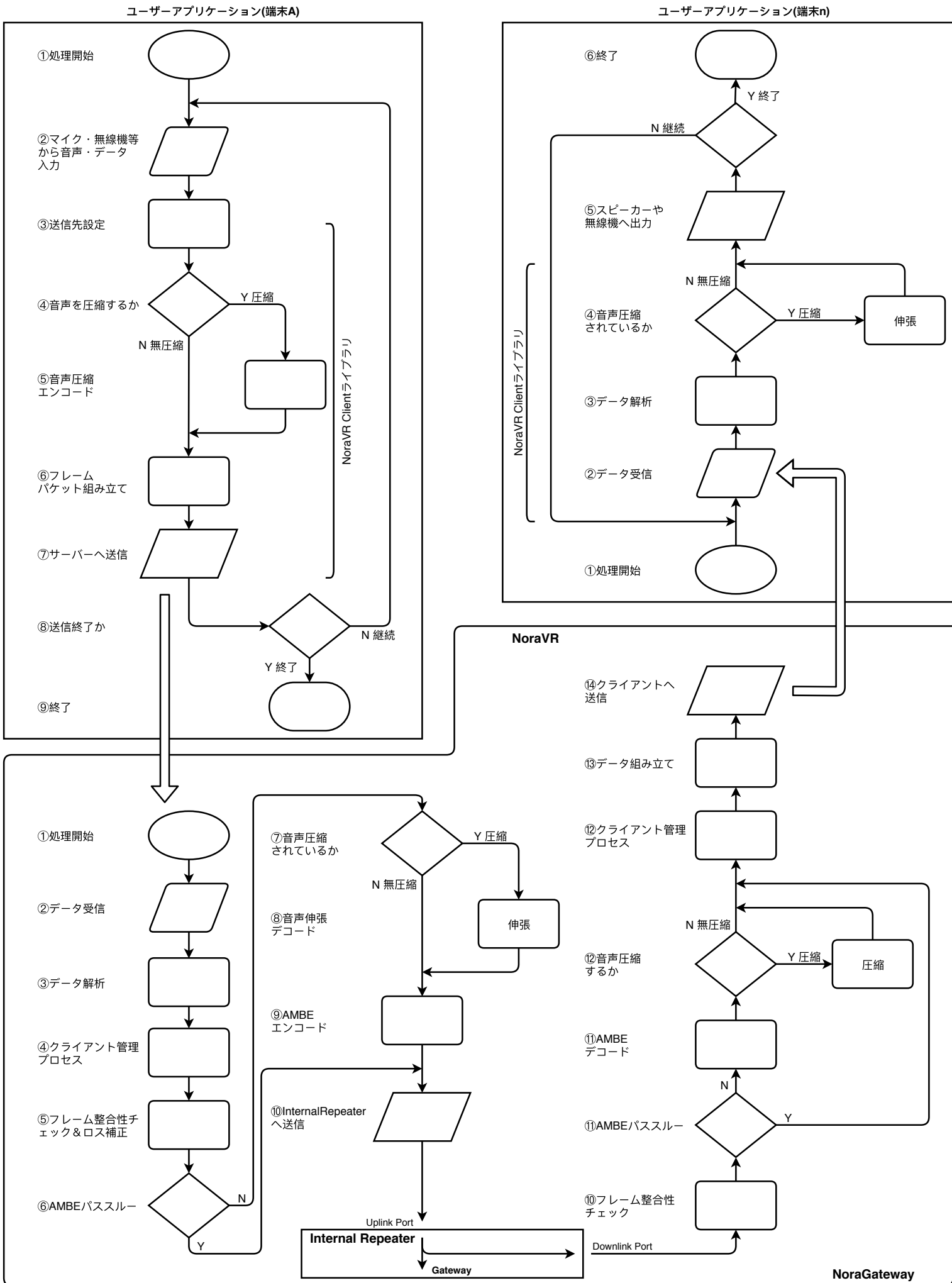
2. イメージブロック図

NoraGateway(Gateway & Repeater System)



ユーザーアプリケーション(自局側)
ユーザー側で自由に開発・設計可能

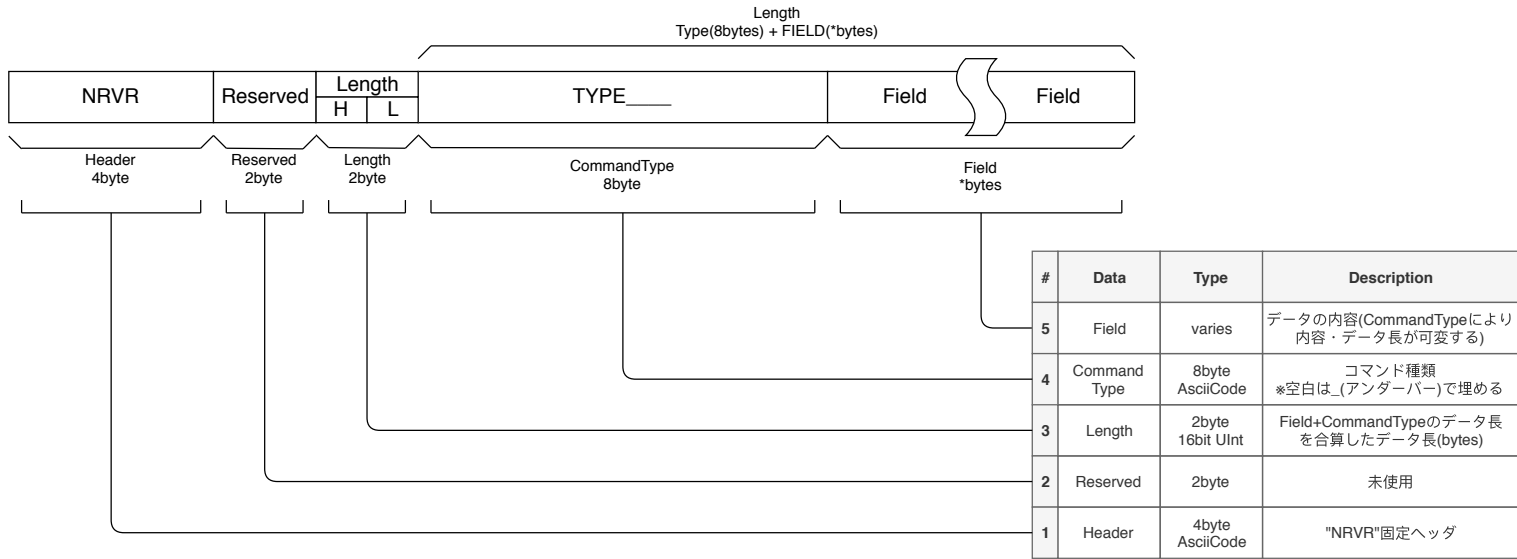
3. 動作フローイメージ



4. 通信プロトコル仕様

① 基本コマンドフレーム構成

*基本的にUDPパケットを使用し、示しているデータはパケット内のデータを示す
 *フォーマット規格外のデータが送信された場合にはサーバーは無応答もしくは、NAKを返却する



② 共通事項

・通信プロトコル

通信プロトコルには、UDPを使用する。サーバー側は、単一のポート(NoraVRポート)を開放し、クライアント側はそのNoraVRポートに対して全てのコマンドの送受信を行う。また、実装を容易にする為に、パケットの分割に対しては考慮せず、1パケットのサイズを可能な限り小さくするものとする。

・クライアントID

ログイン時にLOGIN_ACKコマンド内で、クライアントIDが発行されるので、必ず保持しておくこと
 サーバー側は、各種コマンドを受信した際にクライアントIDの照合を行い、正規のクライアントであるか確認を行う

・生存確認(Ping/Pong)

サーバー側は基本的に、受信したPingに対してPongを返却する。クライアント側は、3~8秒程度の間隔でサーバー側に送信すること。
 ただし、サーバー側は、クライアント側からPingが一定時間送信されなかった場合にのみ、Pongを自動的に送信するので考慮しておくこと。
 クライアント側は、音声パケットの送受信中においても、Pingを送信すること。サーバー側は一定時間(1分程度)、クライアント側から、Pingが送出されなければ、コネクションが切れたものとして強制的にログアウトを行う。

・コマンド応答(ACK/NAK)

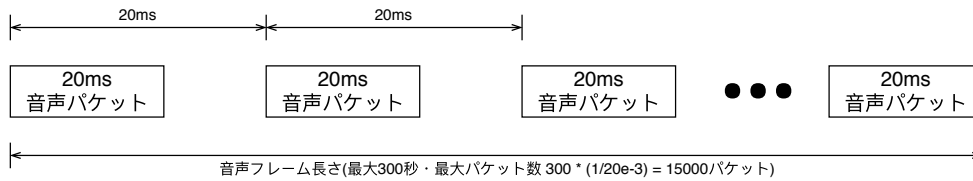
サーバー側は全ての要求に対して、NAK応答を返却する場合がある(コネクションが失われているなど)
 ACKに関しては各コマンド毎に扱いが異なり、ACKを返すコマンドもあれば、LOGIN_ACKなどの専用ACKコマンドが用意されているコマンドも存在する。

・プロトコルバージョンの互換性

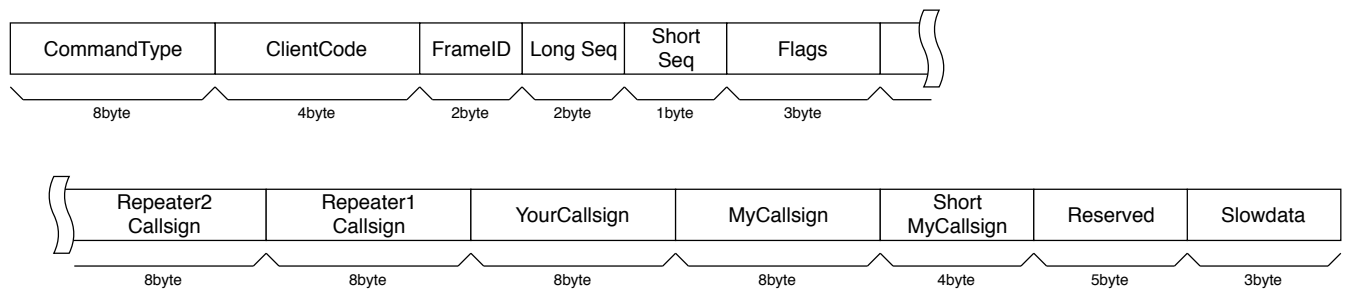
可能な限りの後方互換性を保つため、規定されている各コマンドはReserved(予約)としている領域以外は変更しないものとするが、実装した時点でReservedとなっても、後のバージョンでデータが入っている可能性があるため、無視するなど考慮すること。
 また、未対応のコマンドを受信した場合には、データをパケット丸ごと破棄すること。

・音声フレーム

音声パケットの間隔(=長さ)、0.02秒(=20ms)分とする。また、音声アプリケーションにおける、ソース音声サンプリングフォーマットは、8ksps 16bit 符号付きを推奨する。
 音声パケットの送出間隔については、可能な限り20ms間隔を保つものとするが、特に規定はしない。
 音声フレームの最大長は300秒とし、サーバー側はこれを超えた場合、音声フレームを強制終了する。
 クライアント側は、180秒程度で制限を設ける事を推奨する。



・音声パケット中のヘッダ情報各パラメータ

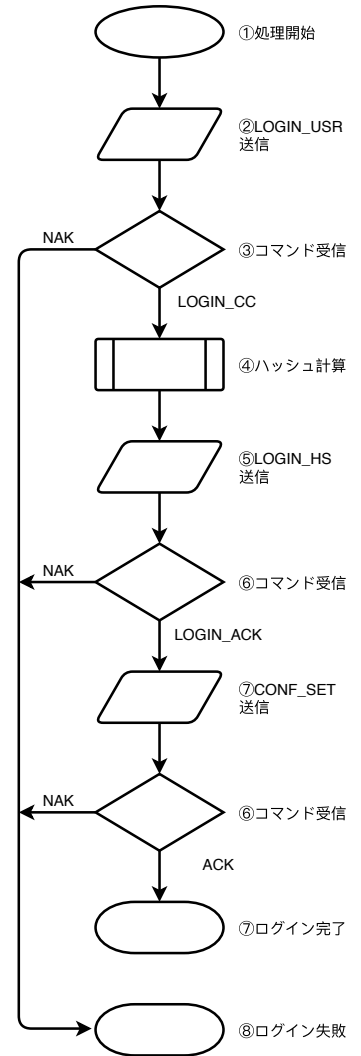
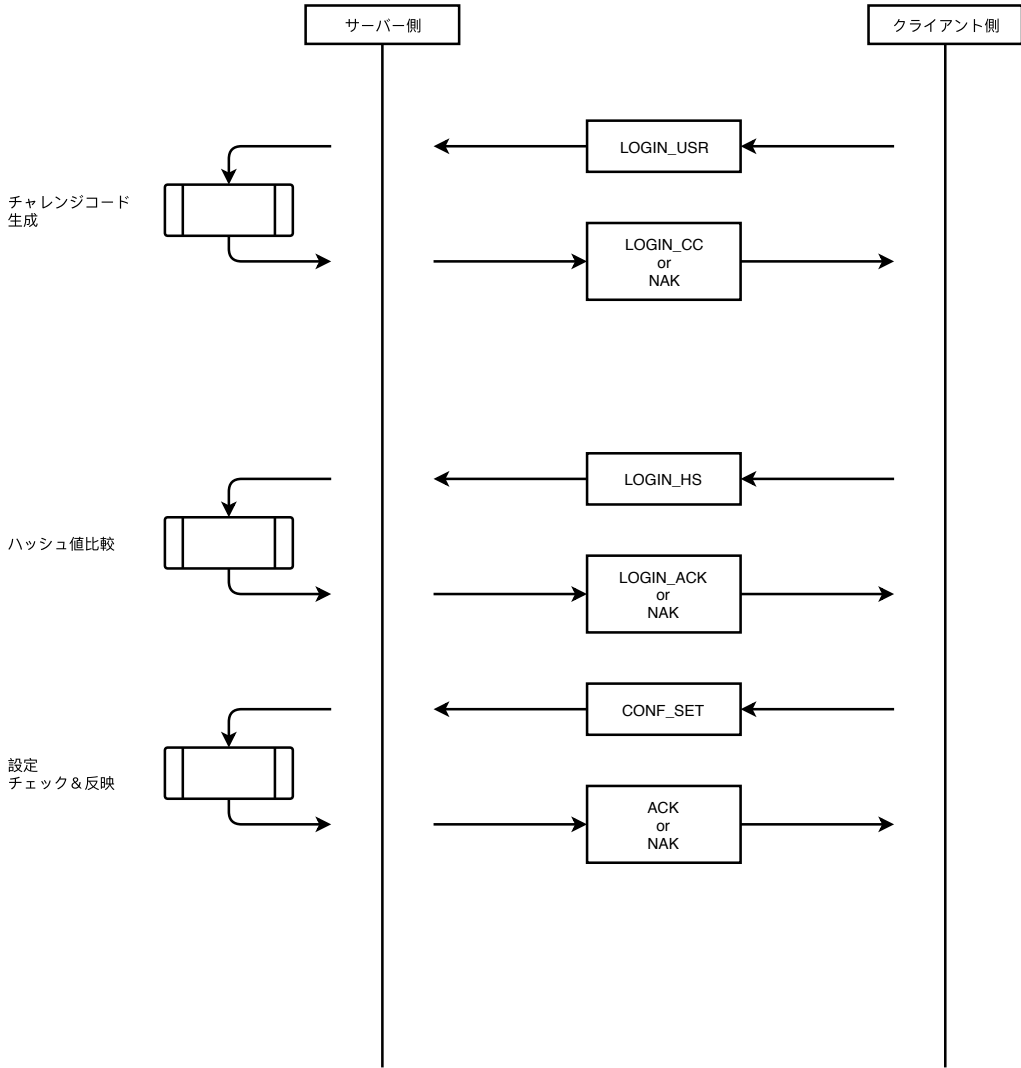


CommandType	VTAMBE_VTOPUS_VTPCM_のいずれか
FrameID	0x0001~0xFFFFの範囲でランダム値を入れる *単一音声フレーム中は同一の値にすること
LongSequence	0x0000~0xFFFFの範囲でインクリメントして循環させる *単一音声フレーム中の初回送信の際は0x00から開始すること
ShortSequence	0x00~0x14の範囲でインクリメントして循環させる 7ビット目を立てることにより、フレーム終端を通知する (送信終了の際も、最後の音声パケットにおいては、7ビット目を立てること) *単一音声フレーム中の初回送信の際は0x00から開始すること
Flags	DSTARヘッダのフラグ(3byte) *送信する際には、1byte目のフラグの7bit目を立てる必要があるので注意すること(詳細はDSTAR仕様書を参照)
Repeater2 Callsign	LOGIN_ACKコマンドにて取得したコールサインをセットする 山かけの場合Repeater Callsignをセットし、ゲート超えや制御コマンドはGateway Callsignをセットする *空白は0x20(スペース)で埋めること
Repeater1 Callsign	LOGIN_ACKコマンドにて取得したRepeater Callsignをセットする *空白は0x20(スペース)で埋めること
Your Callsign	送信先コールサインをセットすること 例として、山かけ・リフレクターの場合CQCQCQ、コール指定の場合JJ0TPX F、エリア指定の場合/JQ1ZYC Fなど *空白は0x20(スペース)で埋めること
My Callsign	自局コールサイン(8文字)をセットする *空白は0x20(スペース)で埋めること
Short My Callsign	自局コールサイン(4文字)をセットする *空白は0x20(スペース)で埋めること
Slowdata	DSTARスローデータをセットする(詳細はDSTAR仕様書を参照) 例として、Nullデータとして指定されている0x66 0x66 0x66を入れる場合には、0x70 0x4F 0x93で各々xorを取ってからセットする Nullデータ 0x66 0x66 0x66 xor 0x70 0x4F 0x93 → 0x16 0x29 0xF5

余白

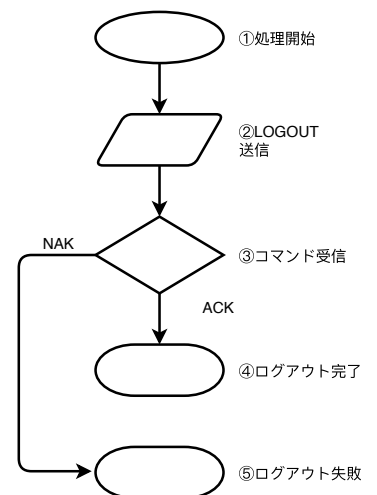
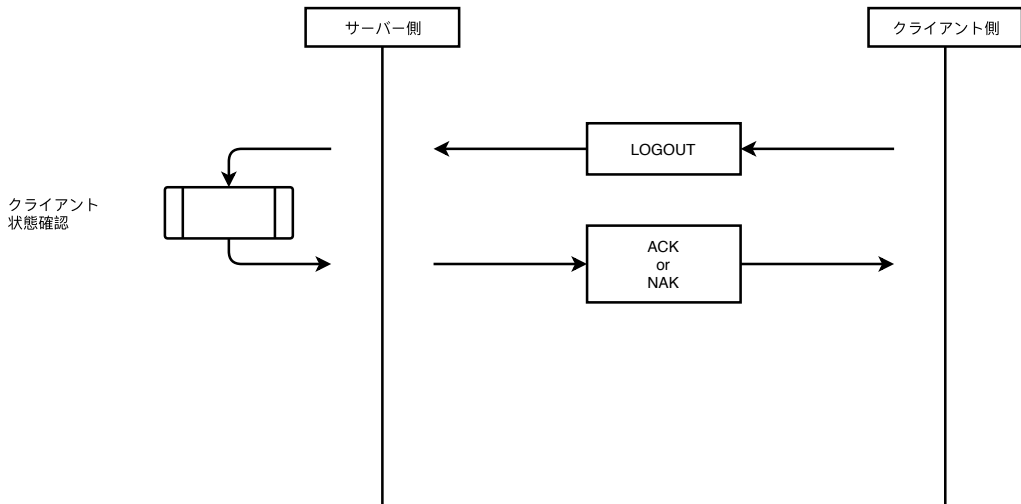
③ 通信動作シーケンス

1. ログインシーケンス



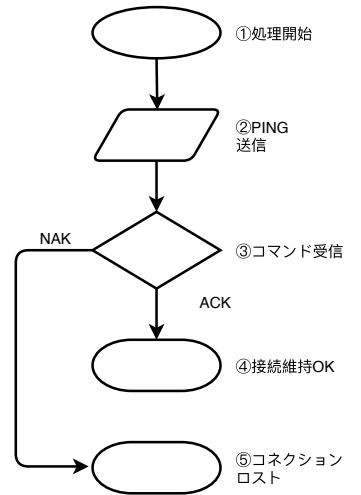
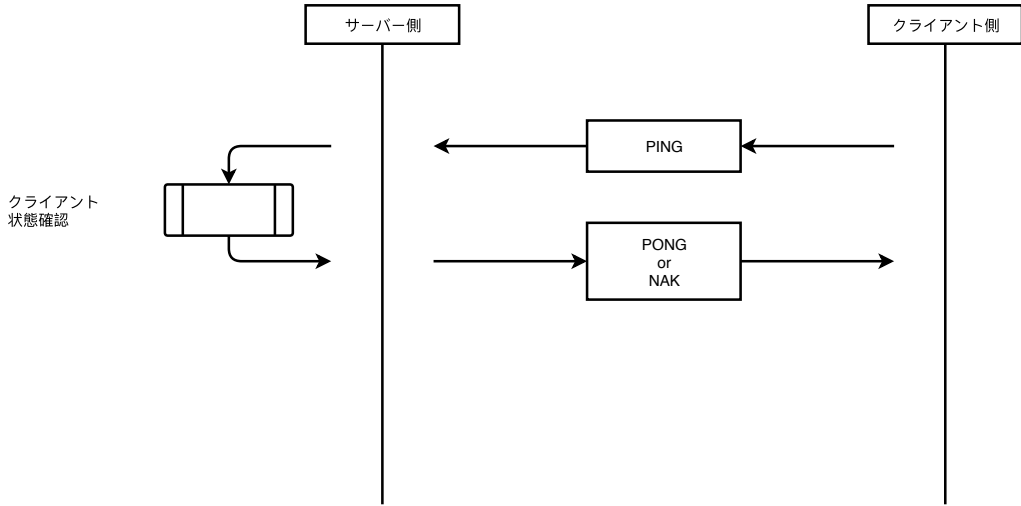
ログインに失敗した場合、サーバー側は一定時間、ログインの再試行を拒否する
 また、LOGIN_HSコマンドの送信に対してNAKが返却された場合には、パスワードが一致しておらず、再試行してもログインが失敗するので、再試行しない動作を推奨する

2. ログアウトシーケンス



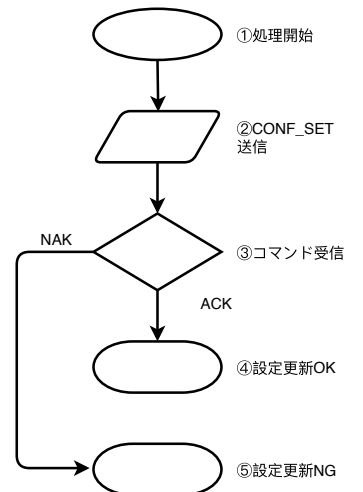
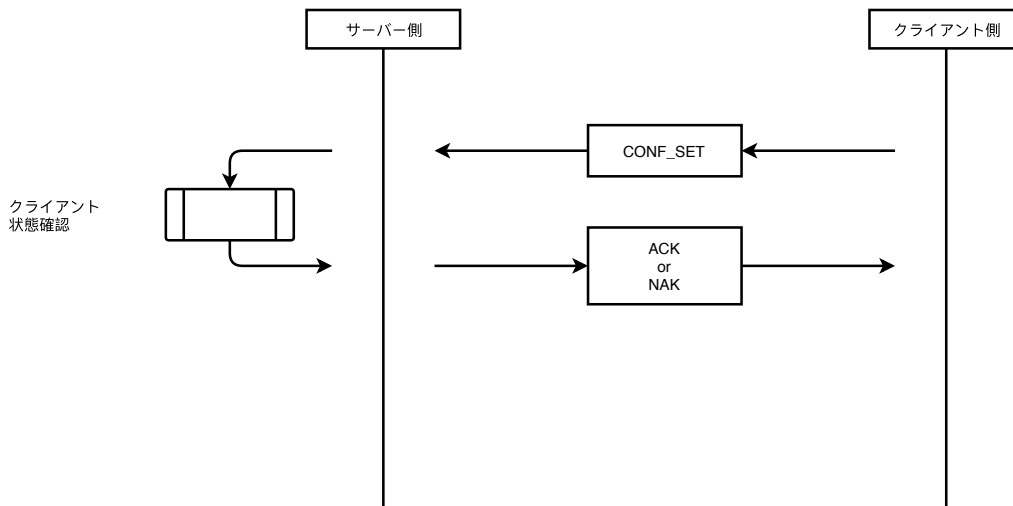
ログアウトが失敗した場合は、再試行しないこと
 サーバー側は、接続維持がされなくなったとして、自動でログアウトを行う

3. 接続維持シーケンス



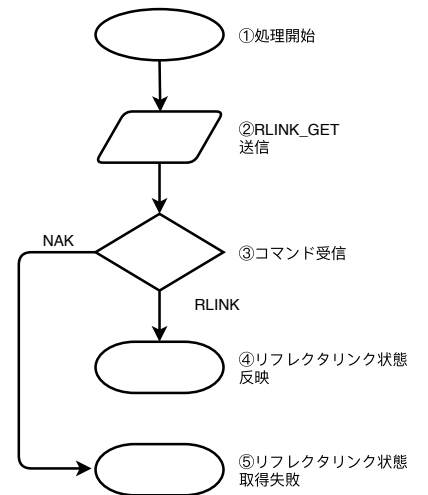
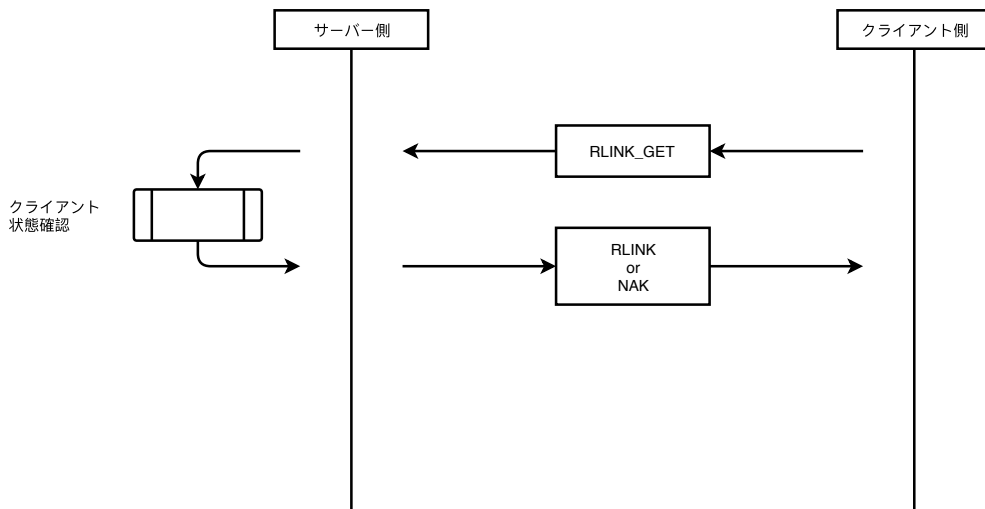
PINGは音声パケットなどの他のデータを送受信中においても、一定間隔で送信すること
 よって、PINGを送信してからPONGを待たずに他の処理を行い、PONGを受信したら単純に最終PONG受信時刻を更新するような処理を推奨する

4. 設定更新シーケンス



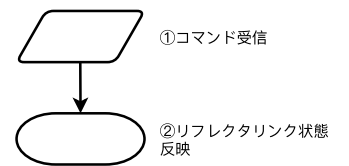
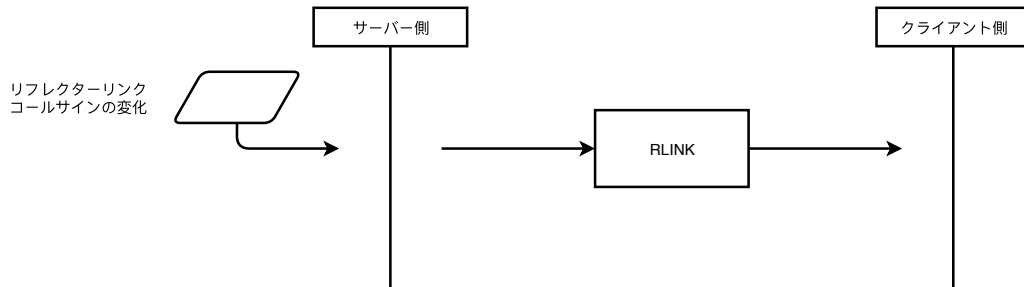
設定を更新する場合に送信すること
 ログイン時には、必ず送信すること
 (送信されなければ音声パケットが流れない)
 例として、エコーバックのON/OFF、
 音声コーデックの変更など

5. リフレクターリンク状態の要求シーケンス



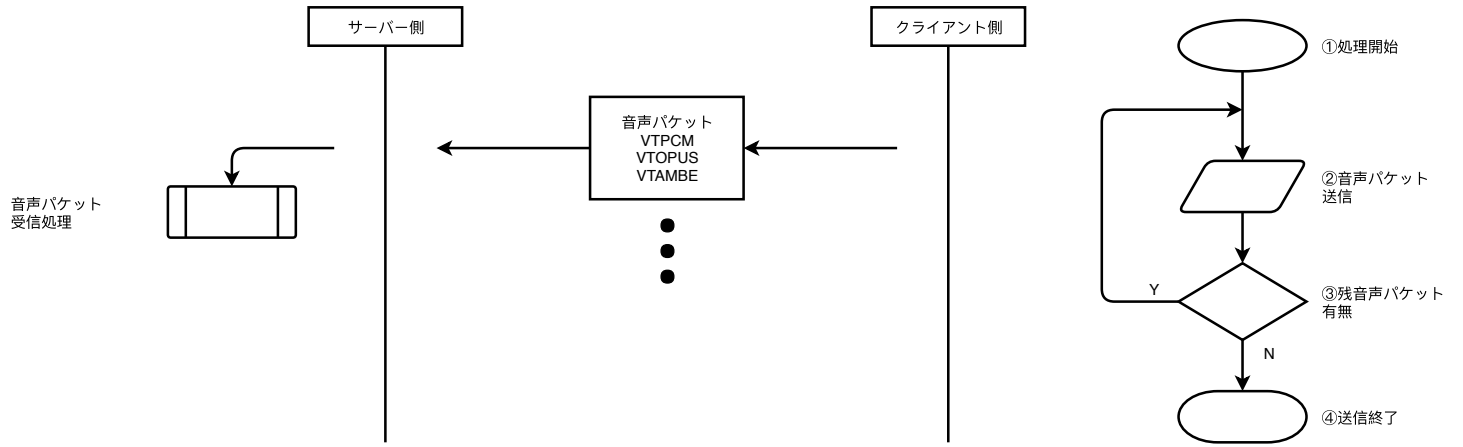
NoraVRを起動しているレピータからリンクされているリフレクターコールサインを取得する。
リフレクターリンク状態の取得が不要であれば送信は不要

6. リフレクターリンク状態の通知シーケンス

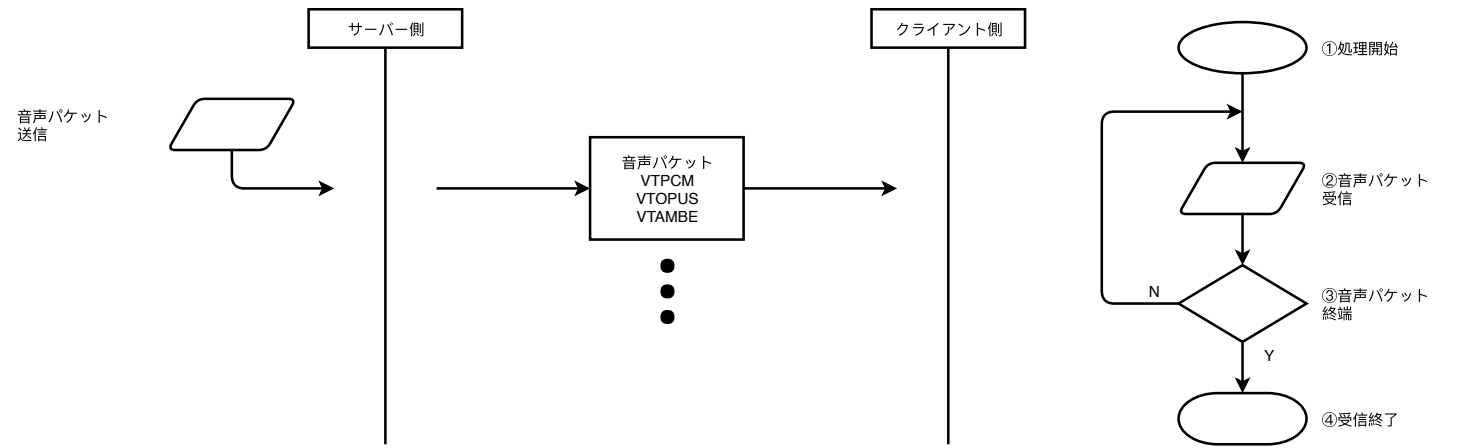


NoraVRを起動しているレピータのリンクされているリフレクターコールサインが変化した場合に通知される

7. アップリンク音声シーケンス



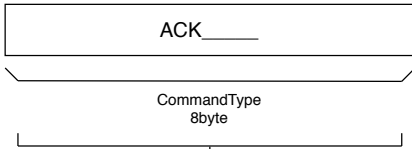
8. ダウンリンク音声シーケンス



④ コマンドタイプ別フィールド構成

1. ACK____

□クライアント → サーバー
 ■サーバー → クライアント

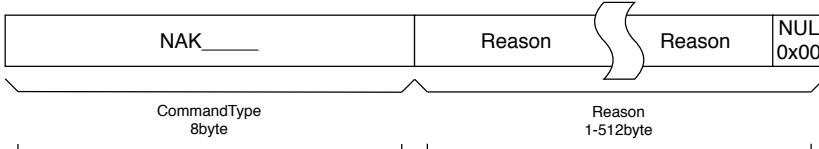


サーバーが処理を受け付けた事を示す

#	Data	Length	Description
1	Login Callsign	8byte AsciiCode	"ACK____"固定ヘッダ *_は5文字

2. NAK____

□クライアント → サーバー
 ■サーバー → クライアント

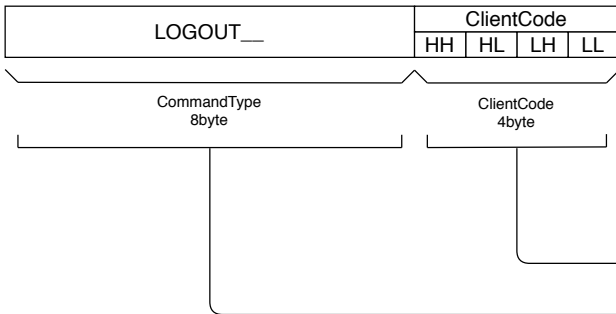


サーバーが処理を受け付けられない事を示す
 Reasonには処理を受け付けられない理由が入る

#	Data	Length	Description
2	Reason	1-512byte AsciiCode	NAKが返される理由 0x00にて終端されたAsciiCode
1	Login Callsign	8byte AsciiCode	"NAK____"固定ヘッダ *_は5文字

3. LOGOUT__

■クライアント → サーバー
 □サーバー → クライアント



クライアントからサーバーへログアウトを通知する

#	Data	Length	Description
2	ClientCode	2byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	"LOGOUT__"固定

4. LOGINUSR

■クライアント → サーバー
□サーバー → クライアント

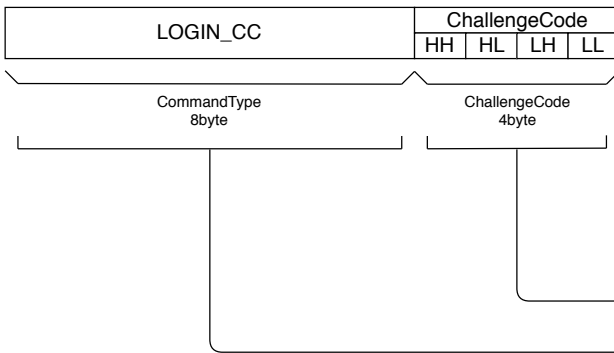


クライアント側からのログイン時におけるログインコマンド
ログインユーザーとは、NoraVRにログインするコールサインであって、音声送信時に送信するMYコールサインに相当するものではない
サーバー側で受け付けられると、LOGIN_CCもしくはNAKが返される

#	Data	Length	Description
2	Login Callsign	8byte AsciiCode	ログインコールサイン *空白はスペースで埋める
1	Command Type	8byte AsciiCode	"LOGINUSR"固定

5. LOGIN_CC

□クライアント → サーバー
■サーバー → クライアント

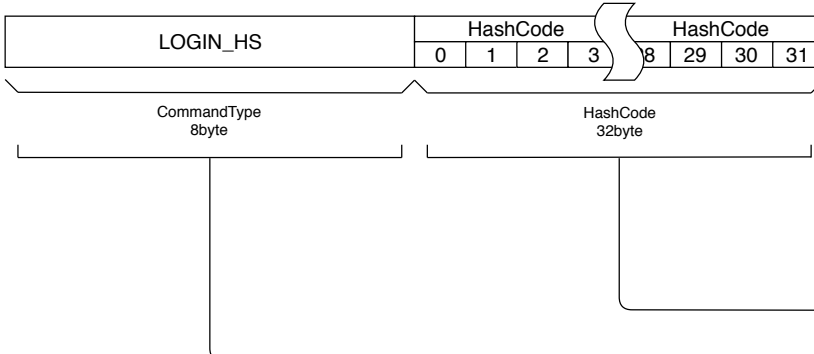


サーバーからクライアントへチャレンジコードを送信する
クライアントではこのチャレンジコードとログインパスワードを結合し、ハッシュ値(SHA256)を生成しておく
【ハッシュシードの構成】
チャレンジコード(4byte BEBinary) + ログインパスワード(Ascii)

#	Data	Length	Description
2	Challenge Code	4byte BE Binary	チャレンジコード 4byte
1	Command Type	8byte AsciiCode	"LOGIN_CC"固定

6. LOGIN_HS

■クライアント → サーバー
□サーバー → クライアント

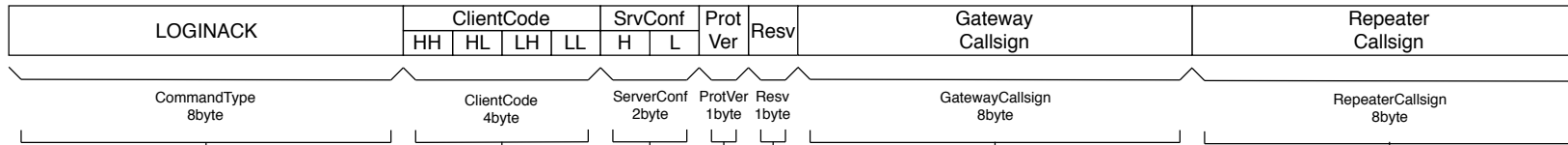


クライアントからサーバーへハッシュコードを送信する
サーバー側では受信したハッシュコードが生成されたハッシュコードと合致しているか確認し、合致していればLOGINACK、合致していなければNAKを返す

#	Data	Length	Description
2	HashCode	32byte LE Binary	ハッシュコード SHA256
1	Command Type	8byte AsciiCode	"LOGIN_HS"固定

7. LOGINACK

■クライアント → サーバー
 □サーバー → クライアント



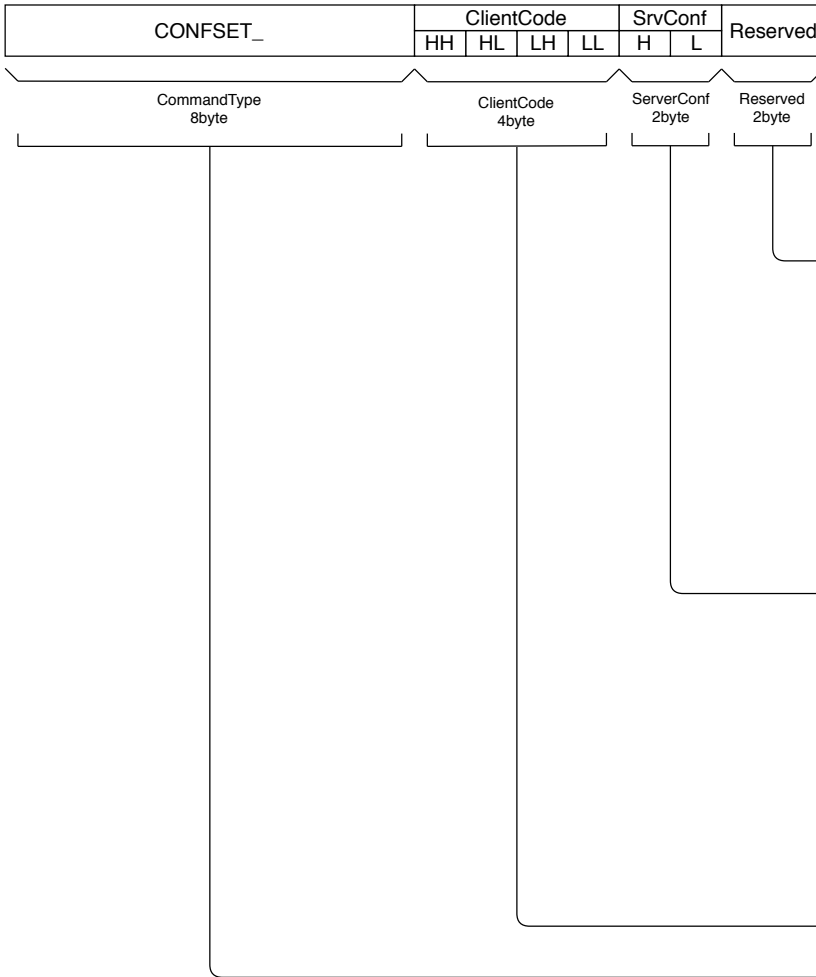
サーバーからクライアントへログインが完了した事を通知
 ログイン完了の通知と、クライアントコード、サーバーの設定を送信する。クライアントコードは、クライアント側で保持しておくこと。また、サーバー設定も同様に保持しておくこと。

#	Data	Length	Description		
5	Reserved	2byte BE binary	Reserved		
4	Protocol Verstion	1byte Binary	プロトコルバージョン		
3	Server Configuration	2byte BE binary	15	Reserved	
			14	Reserved	
			13	Reserved	
			12	Reserved	
			11	Reserved	
			10	Reserved	
			9	Reserved	
			8	Reserved	
			7	Supported Codec	Reserved
			6		Reserved
			5		Reserved
			4		Reserved
			3		Opus VBR 8k 1:enable/0:disable
			2		Opus VBR 24k 1:enable/0:disable
			1		Opus VBR 64k 1:enable/0:disable
0	PCM 1:enable/0:disable				
2	ClientCode	2byte BE UInt	クライアントコード クライアント側で保持しておく		
1	Command Type	8byte AsciiCode	"LOGIN_ACK"固定		

#	Data	Length	Description
7	Repeater Callsign	8byte AsciiCode	レピータコールサイン
6	Gateway Callsign	8byte AsciiCode	ゲートウェイコールサイン

8. CONFSET_

■クライアント → サーバー
□サーバー → クライアント

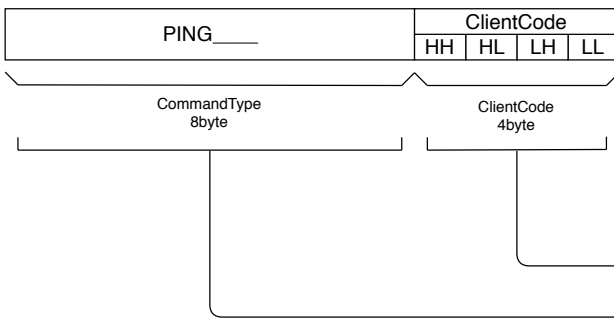


クライアントからサーバーへ設定変更を要求する
クライアントから設定を送り、サーバー内の設定を書き換える
SupportedCodecに関して複数のコーデックをenableにした場合、サーバー側においてenableにしたコーデックの中から自動で選択される

#	Data	Length	Description		
4	Reserved	2byte BE binary	Reserved		
3	Server Configuration	2byte BE binary	15	RFNode RF OutputNode 1:Yes/0:No	
			14	Echoback 1:enable/0:disable	
			13	Reserved	
			12	Reserved	
			11	Reserved	
			10	Reserved	
			9	Reserved	
			8	Reserved	
			7	Supported Codec	AMBE 1:enable/0:disable
			6		Reserved
			5		Reserved
4	Reserved				
3	Opus VBR 8k 1:enable/0:disable				
2	Opus VBR 24k 1:enable/0:disable				
1	Opus VBR 64k 1:enable/0:disable				
0	PCM 1:enable/0:disable				
2	ClientCode	4byte BE UInt	クライアントコード LOGINACKにて取得したもの		
1	Command Type	8byte AsciiCode	"CONFSET_"固定		

9. PING_

■クライアント → サーバー
□サーバー → クライアント

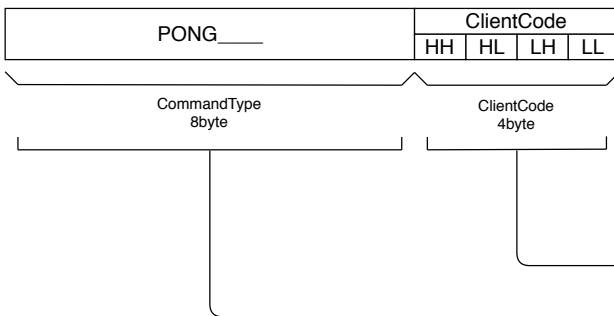


クライアントからサーバーへの生存確認

#	Data	Length	Description
2	ClientCode	2byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	"PING_"固定

10. PONG_

□クライアント → サーバー
■サーバー → クライアント

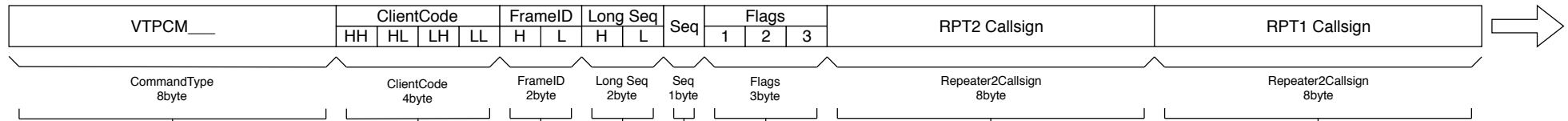


サーバーからクライアントへの生存確認

#	Data	Length	Description
2	ClientCode	2byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	"PONG_"固定

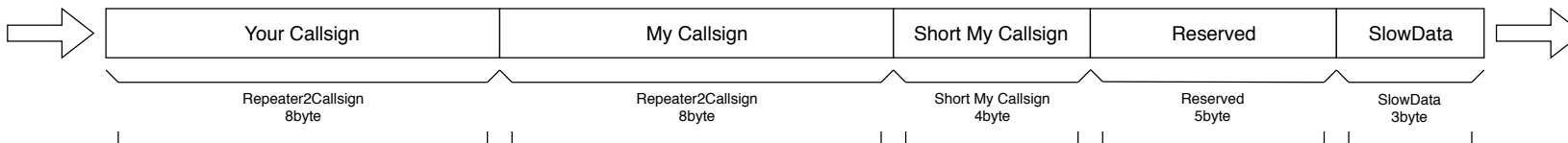
11. VTPCM__

■クライアント → サーバー
 ■サーバー → クライアント

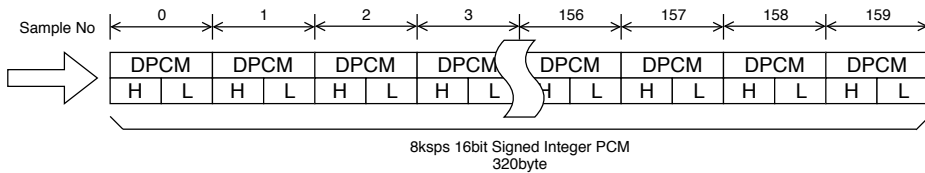


#	Data	Length	Description
2	ClientCode	4byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	コマンド種別 "VTPCM__"固定

#	Data	Length	Description
8	Repeater1 Callsign	8byte AsciiCode	レピータ1コールサイン
7	Repeater2 Callsign	8byte AsciiCode	レピータ2コールサイン
6	Flags	3byte binary	DSTARヘッダフラグ 送信の際は、0x40 00 00とすること
5	Short Sequence	1byte binary	0x00~0x14の範囲でパケット毎にインクリメントして循環させ、最終パケットで0x40とORする
4	Long Sequence	2byte BE binary	0x0000~0xFFFFの範囲でパケット毎にインクリメントして循環させる
3	FrameID	2byte BE binary	フレームID フレーム毎にランダム生成



#	Data	Length	Description
13	SlowData	3byte binary	低速データセグメント
12	Reserved	3byte binary	予約領域
11	Short MyCallsign	4byte AsciiCode	MY CALLSIGN(SHORT)
10	MyCallsign	8byte AsciiCode	MY CALLSIGN
9	Your Callsign	8byte AsciiCode	YOUR CALLSIGN

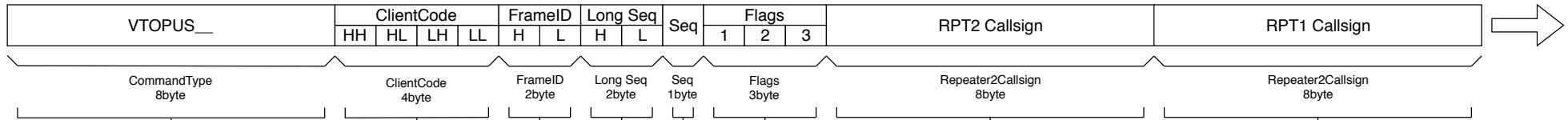


#	Data	Length	Description
14	PCM	2byte UInt *160sample	8ksps 16bit Signed Integer PCM



12. VTOPUS__

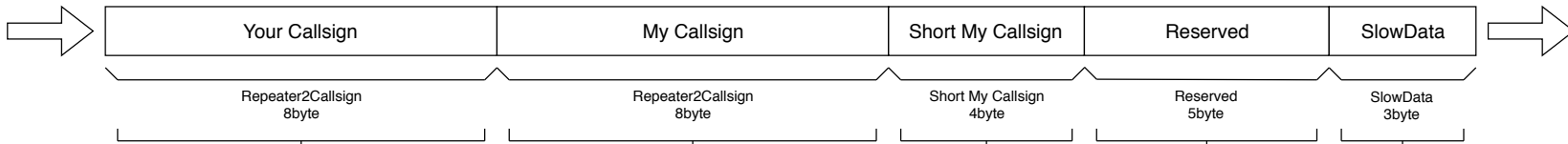
■クライアント → サーバー
 ■サーバー → クライアント



CommandType 8byte ClientCode 4byte FrameID 2byte Long Seq 2byte Seq 1byte Flags 3byte Repeater2Callsign 8byte Repeater2Callsign 8byte

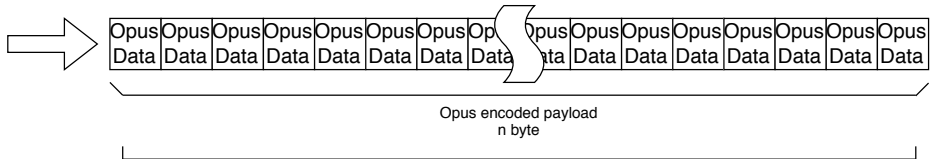
#	Data	Length	Description
2	ClientCode	4byte BE Uint	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	コマンド種別 "VTOPUS__"固定

#	Data	Length	Description
8	Repeater1 Callsign	8byte AsciiCode	レピータ1コールサイン
7	Repeater2 Callsign	8byte AsciiCode	レピータ2コールサイン
6	Flags	3byte binary	DSTARヘッダフラグ 送信の際は、0x40 00 00とすること
5	Short Sequence	1byte binary	0x00~0x14の範囲でパケット毎にインクリメントして循環させ、最終パケットで0x40とORする
4	Long Sequence	2byte BE binary	0x0000~0xFFFFの範囲でパケット毎にインクリメントして循環させる
3	FrameID	2byte BE binary	フレームID フレーム毎にランダム生成



Repeater2Callsign 8byte Repeater2Callsign 8byte Short My Callsign 4byte Reserved 5byte SlowData 3byte

#	Data	Length	Description
13	SlowData	3byte binary	低速データセグメント
12	Reserved	3byte binary	予約領域
11	Short MyCallsign	4byte AsciiCode	MY CALLSIGN(SHORT)
10	MyCallsign	8byte AsciiCode	MY CALLSIGN
9	Your Callsign	8byte AsciiCode	YOUR CALLSIGN



Opus encoded payload
n byte

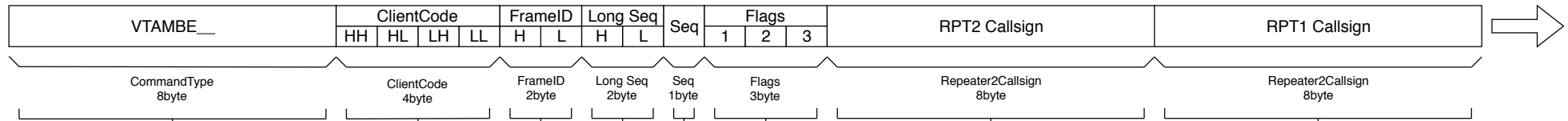
#	Data	Length	Description
14	Opus Voice	nbyte binary	8k/24k/64kbps VBR FEC Opus Encoded Voice

Opusでエンコードされた音声パケット
 8kbps/24kbps/64kbps VBR FEC付でエンコードを行うこと

18

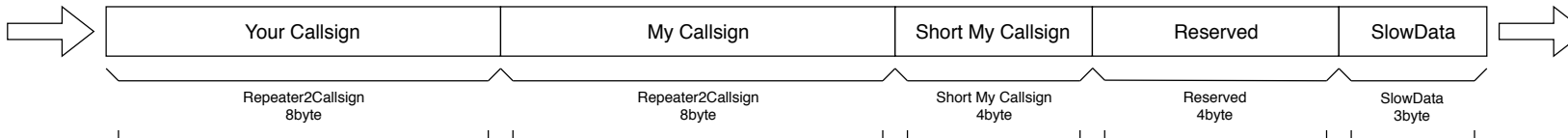
13. VTAMBE__

■クライアント → サーバー
 ■サーバー → クライアント

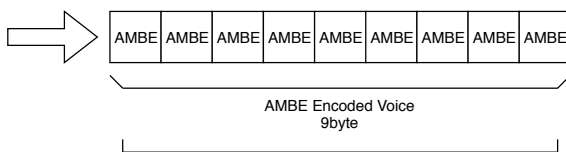


#	Data	Length	Description
2	Client Code	4byte BE Uint	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	コマンド種別 "VTOPUS_"固定

#	Data	Length	Description
8	Repeater1 Callsign	8byte AsciiCode	レピータ1コールサイン
7	Repeater2 Callsign	8byte AsciiCode	レピータ2コールサイン
6	Flags	3byte binary	DSTARヘッダフラグ 送信の際は、0x40 00 00とすること
5	Short Sequence	1byte binary	0x00~0x14の範囲でパケット毎にインクリメントして循環させ、最終パケットで0x40とORする
4	Long Sequence	2byte BE binary	0x0000~0xFFFFの範囲でパケット毎にインクリメントして循環させる
3	FrameID	2byte BE binary	フレームID フレーム毎にランダム生成



#	Data	Length	Description
13	SlowData	3byte binary	低速データセグメント
12	Reserved	3byte binary	予約領域
11	Short MyCallsign	4byte AsciiCode	MY CALLSIGN(SHORT)
10	MyCallsign	8byte AsciiCode	MY CALLSIGN
9	Your Callsign	8byte AsciiCode	YOUR CALLSIGN



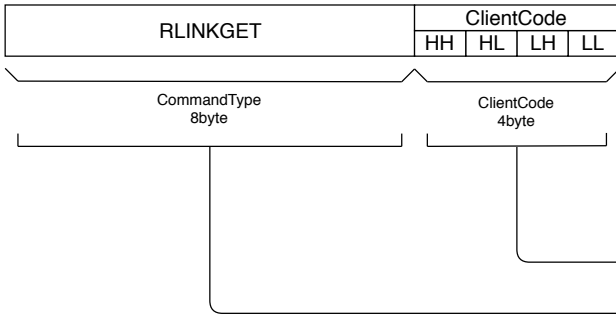
#	Data	Length	Description
14	AMBE Voice	9byte binary	AMBE Encoded Voice 3.6kbps(FEC1.2kbps)

AMBEパススルー音声(データ)パケット
 音声の場合には、3.6kbps(FEC1.2kbps)でエンコードされたデータをセットすること
 ファストデータをセットする場合には、DSTAR仕様書を参照のこと

19

14. RLINKGET

■クライアント → サーバー
 □サーバー → クライアント

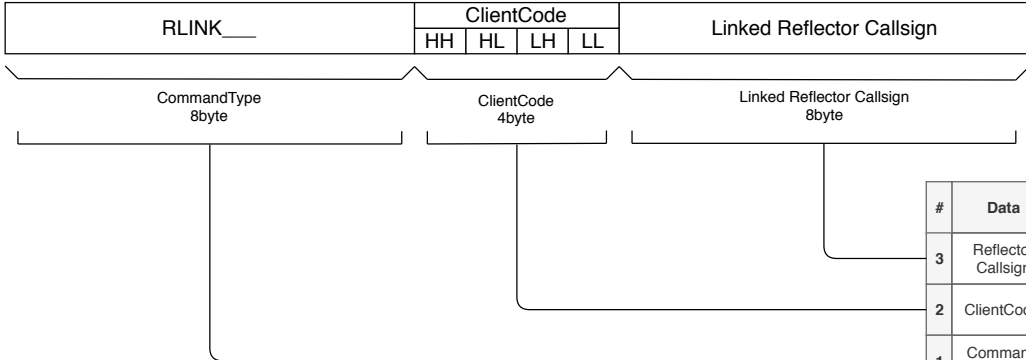


クライアントからサーバーへリフレクタリンク状態を要求

#	Data	Length	Description
2	ClientCode	2byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	"RLINKGET"固定

15. RLINK__

□クライアント → サーバー
 ■サーバー → クライアント



リフレクターのリンク状態を通知

#	Data	Length	Description
3	Reflector Callsign	8byte AsciiCode	リンクしている リフレクターコールサイン
2	ClientCode	2byte BE UInt	クライアントコード LOGINACKにて取得したもの
1	Command Type	8byte AsciiCode	"RLINK__"固定

余白

5. 改版履歴

2019.03.12 初版 初版発行

6. 連絡先

JQ1ZYC 圏央道友会
kenoh_doyu@txb.sakura.ne.jp