

3. パケットデータ伝送における誤り制御の基本技術

3.0 はじめに

前回まではIEEE 802.15.6 BAN MACにおける競合アクセス (contention access) と非競合アクセス (contention-free access) の基本的なチャンネル・アクセス手順について学んできました。競合アクセスでは(データの重要度に依存する)優先権の設定パラメータやチャンネルアクセス後のデータのやり取りの基本を学びました。一方、非競合アクセスでも従来から用いられている Slotted ALOHAのような送信タイミングに制限の大きくなるチャンネルアクセス方式に存在した欠点を改善できるようAC(Access Category)などの通報内容によるチャンネル割り当ての機会に“優先度”を設ける仕組みが設けられました。これはデータ伝送手順の改善技術の進展をIEEE 802.11 Dを簡略化した優先度の導入で達成したもので、BAN応用では要となるアクセス制御です。これらのデータ伝送手順の基本技術は「自動再送要求 (Automatic Repeat Request)」です。第3回講座ではデータ伝送における効率化の要となるARQについて講座を開設しようと思います。ARQについて詳しい解説がなされている無線通信関連の技術書は少ないためBAN PCFシステム設計関連技術入門者においても規格の中に現れる“Type I ポーリング”や“Type II ポーリング”と云う述語にハードルの高さを感じる人も多いのではないかと感じます。以下ではARQ 入門講座と云う範囲で内容を 3.1 ではARQの基本手順、3.2 ではHybrid 方式の概要を、3.3 では “Type I hybrid” や “Type II hybrid” ARQで必要となる誤り検出符号、誤り訂正符号の基本知識 について述べてみようと思っています。

ARQはデータ伝送における誤り制御 (error control) の基本技術の一つです。どんなデータ伝送技術にも誤り制御は必要です。メッセージの受信者に正しいデータを送り届けるための技術として

- ① 誤り訂正符号化(Error Correction Coding : ECC)

と

- ② 自動再送要求(Automatic Repeat Request : ARQ)

がありますがARQには

- (a)基本ARQ(simple ARQ)

と

- (b)複合ARQ(Hybrid ARQ)

の2種類があります。

(a)の基本ARQは誤り検出符号(error detection code)を用いて受信データに誤りが検出された場合にはデータの再送を要求するNACKメッセージを送り、誤りの検出が無くなるまで再送を繰り返す技術です。また、(b)の複合ARQ(Hybrid ARQ)は誤り訂正(Error Correction)と誤り検出(Error Detection)を併用して再送回数を減らし伝送路の効率的利用を計る技術です。メッセージを交換する通信目的ではワイヤレス通信でもARQが用いられるのが一般的です。ただ、テレビやラジオ放送サービス(broadcasting service)のように一方向にのみメッセージを流すシステムでは誤り訂正符号(Error Correction Code)を用いて受信機のみでメッセージの復元を実行するのが普通です。このような受信側でのみデータ復元を可能とする符号化方式は前方向誤り訂正(Forward Error Correction : FEC)技術と呼ばれています。FECはCDやDVDその他の情報蓄積型メディアの符号化方式としても使われていて現在のデジタル情報社会では不可欠な技術の一つになっています。なお、ARQにおいて、誤り訂正符号(FEC)と誤り検出符号(Error Detection Code: EDC)は異なった範疇として設計されるのではなく、通信路の特性(無線通信では特に電波伝搬特性)を詳細に調べたうえで、両方の機能の限界を見極めて符号化・復号化方式を選ぶのが一般的です。とは云うもののARQに適した符号の基本性能は誤り検出能力なのでそこに重点をおいて設計することが求められます。ARQを目的とした符号のリストはこれまでの研究でほぼ明らかにされており、国際的な標準方式として規定もされているので、その中から最適なものを選ぶのが開発を迅速に進めるための選択肢と言えるでしょう。

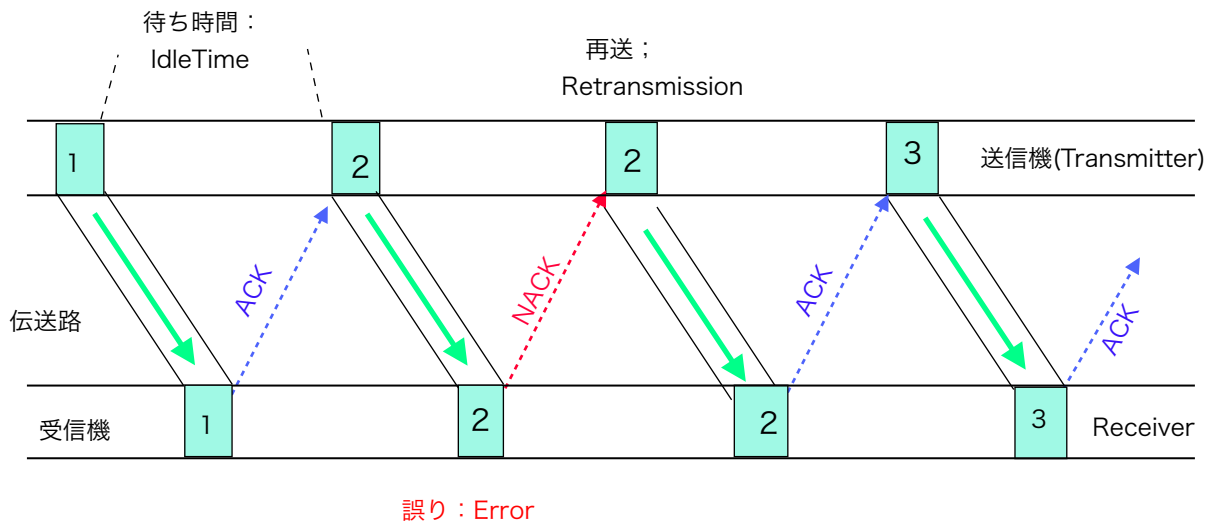
3.1 ARQ (Automatic Repeat Request) 入門

基本ARQは、受信側で誤り検出のみ実施し受信側で誤りが検出されな (syndrome = 0) ければACKを返し、誤りが検出されたら (syndrome ≠ 0) NACKを返す。一方、送信側ではACKを受信(positive ACK)したら次のデータを送り出す。NACK(positive NACK)か、返信(回答)が受け取れなければ(negative ACK)再度同一データの送信を試みる：と云う基本的な動作を実施します。

3.1.1 Stop-and-Wait ARQ(SaW ARQ)手順

図_Fig.6aに Stop-and-Wait ARQシステムにおけるデータ伝送手順の概要を示してあります。送信機(側) (transmitter)と受信機(側) (receiver)の間に伝送路(transmission line)が介在しそこは雑音によって伝送誤りが発生し易い環境になっている。SaW手順では送信機は符号語データを送り出し、受信側では受信されたコードワード (code word : 符号語) に誤りが検出されなければ、データはそのまま受信者に送り出されて格納され、送信者にはACK(Acknowledgment)が返されます。誤りがあればNACK(Negative-Acknowledgment)を返します。送信機はACKを受信したら次の符号語データを送り出しますが、NACKが戻ったら先程の符号語データを再送します。通信路の状況によっては待ち時間(図_Fig6a 中 Idle-time)内にACKもNACKも戻らない

ことがありますがこの場合にはNACKを受け取ったと判定して符号データを再送することになります。SaW伝送プロトコルはIdle - time があるためデータ長が短いと非効率的です。データ長を特別に長くすれば伝送効率は上がりますが、欠点も生まれます。長くなれば符号データの受信誤りが増えて (syndrome $\neq 0$) 再送回数が増える結果を生じます。その他、(例えばセンサーなどからの) 観測データの性質上長くなることは稀なので、SaW におけるIdle-time の寄与は(悪い意味で)大きくなってしまいます(多くの専門書では "NACK" ではなく "NCK" が簡略化術語として使われていますが、本講では "NACK" を使用しています)。

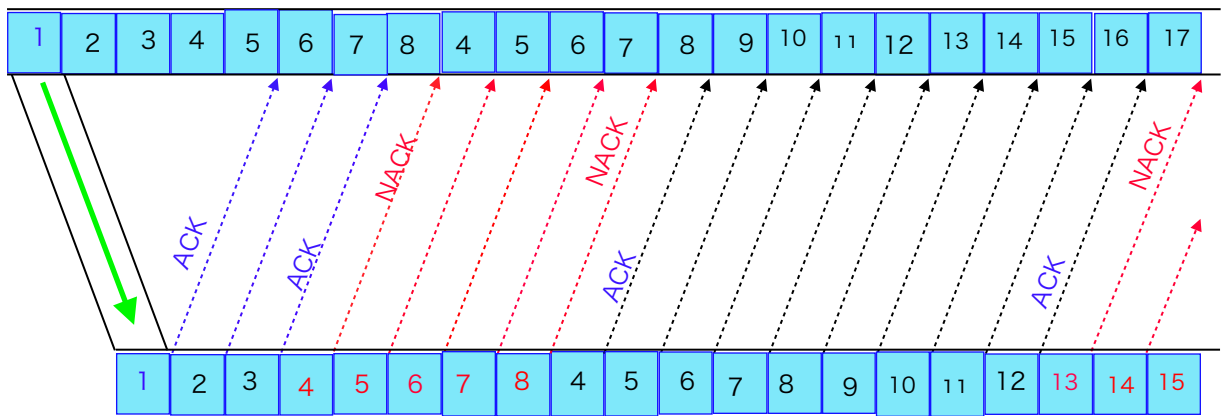


図_Fig. 6a Stop-and Wait ARQ(SaW) における基本的データの送受信手順

3.1.2 Go-back-N ARQ 手順

SaW ARQ手順は簡単なプロトコルなので伝送遅延が少なくデータレートが低い場合にはプログラムが簡単でありシステム構築が容易ですが、非効率的な伝送方式であることは明白です。そこで、待ち時間中にもデータを送り続けることで伝送効率を上げる手順が考案されました(メモリが高価で消費電飾が大きかったデジタル伝送初期段階ではSaWにもメリットがあった)。これの基礎的手順が、Go-back-N ARQである。この方式では伝送の待ち時間: round-trip delay time (SaWではIdle-timeと名付けた)にもACKを待つことなしに符号データを送り続ける。round-trip delay timeとは、術語通り、送信機が符号データを送り出してからACK (NACK) を受け取るまでの時間である。Go-back-N の"N" はround-trip delay time時間中に送る符号データの数であり、誤りが発生(送信機が "NACK" 受信)した場合には(送信機のカウンターは) 誤り発生 of データまで戻り、当該データを再送するとともに、更にそれに続く "N-1"個のデータを送りだします。受信機では誤りの検出に伴いN個のデータが再送されるので、すでに受信されている (n-1) 個データは、誤りが有る無しに拘らず全て捨てられます。Go-back-N 手順で N=5 の場合の例を 図_Fig.6b に示しておきます。このように、Go-back-N 手順では SaW ARQより

(round-trip time 中にもデータを送信し続けるので) 伝送効率は改善されるが、最大(n-1)個のデータが無駄に廃棄されます。そのため、更なる効率改善の余地があり、次項目 3.1.3 に述べる Selective-repeat ARQへと繋がる。G-back-N ARQ手順を用いた伝送手順としてはIBMで開発されたSDLC(Synchronous Data Link Control) などが知られているが、SDLCは現在ではよく知られた(国際標準規約で規定されている) HDLC(High Level Data Link Control)のsub-setとして含まれています。



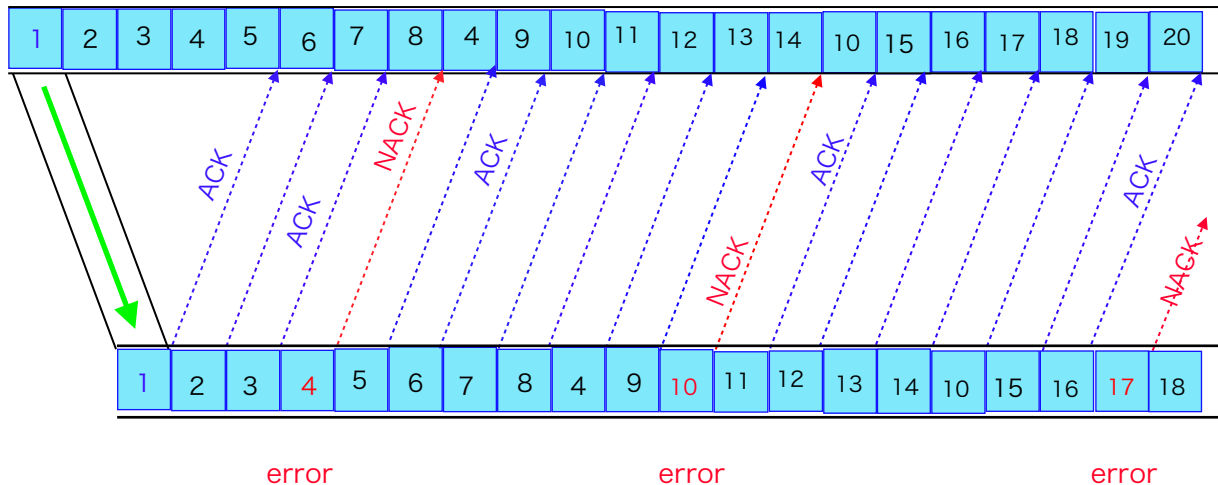
図_Fig.6b Go-back-N ARQにおけるデータ送受信手順：

図では、データ番号 4と13で誤りが検出され、応答データとして**ブルー**の破線で示したのはACKを、**赤**の破線で示したのはNACKを示している：スペースの都合上記述の無い破線もある。なお、ここでは round-trip time $N=5$ としている。そのため、データ4以降に送られた(NACKが帰って来るまでに送られた) 5, 6, 7, 8は再送信される。それに伴い、既に受信されバッファにあるそれらのデータは受信機からは削除される。

3.1.3 Selective-repeat ARQ 手順

Selective-repeat ARQ でもGo-back-N AQと同様連続的にデータを送り出すが、再送されるのは誤りが検出された (syndrome $\neq 0$) データのみである。当然この手順の伝送効率がこれまでの2つの手順と比べると高いことがすぐ理解できるだろう。この伝送方式の基本的な手順を図 Fig. 6c に示してある。伝送効率が高い分送受信制御が複雑になることは当然である。受信者にデータが渡る時には、当然のことながら、順序正しく並んでいなくてはならない。どのデータが誤るか予測は不可能であることを考えると、受信機で備えているバッファメモリの容量はクリティカルである。メモリが不足すればオーバーフローが生じた以降のデータは全て失われてしまう。最近ではメモリの消費電力が大幅に削減されていることは周知の事実であるがBIO/healthcare応用

と言う領域を考慮すると可能な限り効率的な装置化が望まれる。そのため、有限のバッファメモリを有するシステム条件下における手順の詳細（データ送受信制御の詳細）と伝送効率の検討が欠かせない。このような現実的な利用に対する装置化(implement)は複雑であるがいつれ機会を作って紹介してみたい。



図_Fig. 6c Selective-Repeat-ARQ における基本データ伝送手順

3.2 ハイブリッド(Hybrid) ARQ

3.0 で述べたように誤り制御にはFEC とARQがありそれぞれの特徴を生かした利用分野で使用されてデータ伝送の効率化と信頼性の向上に寄与しています。データ伝送の効率はスループットと呼ばれる指標（簡単に言えば、送った符号語と正しく受信された符号語の比です）で評価されますが、この講座では（各ARQ）それぞれの 具体的 throughput 特性については言及しておりません。しかし、容易に推測できることは、「通信路の特性（符号語誤り率）の劣化に伴いスループット(throughput)は減少する」と言うことで、その特性は大まかにselective ARQ > Go-back-N ARQ > SaW ARQの順になることはこれも推測が容易です。Gauss 雑音通信路(Gaussian Noise Channel)での理論解析結果ではselective ARQのthroughputは符号語（符号データ）の符号化率 $\eta = k/n$ に漸近すること示されています（ k ：情報ビット数, n ：符号長）（BAN応用では勿論通信路の特性は全く異なります）。FECのスループットは、基本的には、 $\eta = k/n$ で、通信路の誤り率とは独立です。これらの結果を勘案すればFECの利用が優れていると言えます。にも拘らずARQが使用されるには理由があります。それは、交換されるデータの信頼性です。既に述べたように、「受信データの信頼性」と「データ伝送効率」は通信システムの基本特性で、両特性のトレード・オフを勘案した最適化が通信路の設計には常に求められます。FECで

送られたデータ（符号語）は受信機で誤りが検出された場合誤り訂正の手順に回され訂正されて（通報）受信者へ渡されます。この再生されたデータが正しく訂正されている場合がほとんどですが、誤って別な系列に復号されてしまうこともあり得ます（誤訂正確率）。それとは別に、誤って受信されたが、それを検出できないこともあります（誤り見逃し確率）が、これはARQでも共通の劣化要因です。誤り訂正における理論解析によると（誤訂正確率） $>$ （誤り見逃し確率）で、これはデータ伝送におけるFECの信頼性を損なう要因になっています。誤訂正の確率を減少させる（結果として通信の信頼性を高める）ためには長い符号語を使えばいい（一般的な議論としてはShannonの通信路容量に関する復号理論を思い出してください）のですが装置化と復号に多くの資源を割かなければなりません。これに比べてARQの実装は比較的簡単で、信頼性を高めることが可能です。このようないくつかの要因で初期のデータ伝送にはARQが用いられていました。一方においてデジタル信号処理能力の向上は目覚ましく、その基礎を成す半導体の高密度化・高速化と相まって誤り訂正符号化・復号(FEC)との組み合わせで高効率で高信頼性のデータ伝送手順が実現されています。スマートフォンに代表される無線モバイル通信では 3G, LTE/4G 5Gなどで TURBO符号や LDPC などの情報理論的確率論に基礎を置いた高信頼性符号化・復号化を用いた (advance Type II Hybrid や Type III Hybrid 手順に繋がる) ハイブリッド伝送方式が実現されていますが、本講座ではブロック符号化・復号化方式を用いた基本的ハイブリッドARQ (Hybrid ARQ)の紹介に止める予定です。

3.2.1 タイプ I 型ハイブリッド ARQ (Type I Hybrid ARQ)

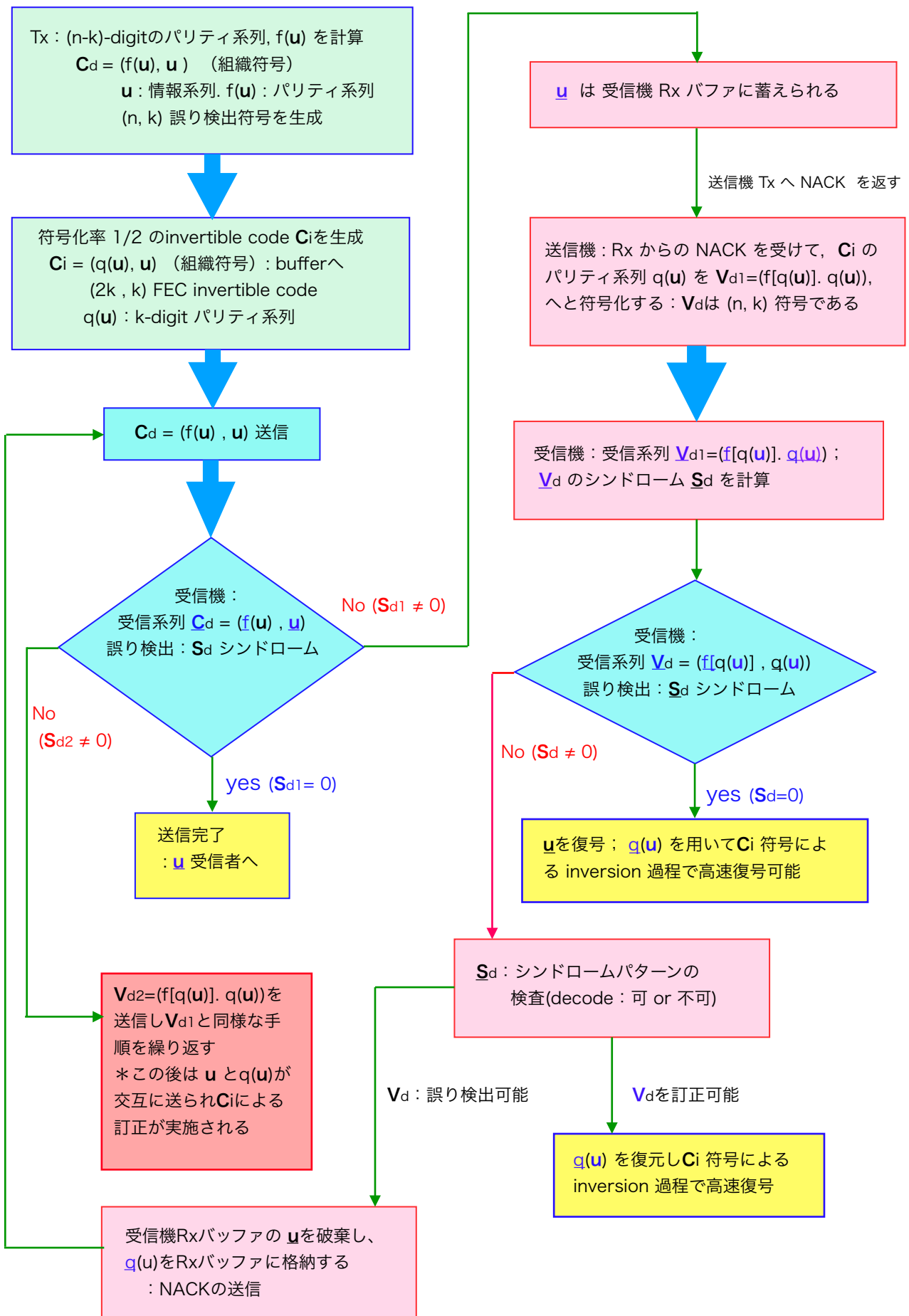
Hybrid ARQ の基本手順は誤り訂正 (FEC) と誤り検出 (ED : error detection) ですがそのために使用する符号語にBCH符号 (M-ary symbol : 多値記号ではReed-Solomon 符号) のようなブロック符号を用いるのが Type I 型 ARQです。以下ではもう少し詳しく説明しましょう。

受信機で検査されたデータに誤りがある (と判定された) 場合、それを再送してもらうか訂正するかの判断はどうなされるのでしょうか。システムの基本構成は、メイン手順は再送(ARQ)で誤り訂正はサブシステムとして機能することにあります。それゆえ、まず、誤りが検出されなければ (syndrome = 0) 受信符号語データはそのまま受信者に送られます(ACK返信)。誤りが検出された場合には (syndrome \neq 0) , 符号語データ中の誤りパターンを調べて再送手順に回すか誤り訂正サブシステムに送るかを決めます。訂正が容易な誤りパターンの場合には訂正サブシステムに、一方、訂正に手数料がかかる誤りパターンの場合には再送のためNACKを返します。訂正が容易か手間が掛かるかの判定基準は用いる符号と通信路の特性に大きく依存します。符号理論からの知識によると「符号の 最小距離 “ d_{mini} ” $>$ 3」の符号は“ 誤り訂正 ”と“ 誤り検出 ”の機能を割り当てることができますが、誤り訂正可能なビット数を増やせば誤り検出可能なビット数は減少します (因みに最大誤り訂正可能ビット数 C は 「 $d_{\text{mini}} - 1$ 」 / 2, 最大誤り検出可能ビット数

Dは $(d_{\text{mini}} - 1)$ です。ただし「 $d_{\text{mini}} - 1$ 」/2は $(d_{\text{mini}}-1)/2$ を超えない最大の整数です)。先ほどの「訂正容易なビットパターン」は誤り訂正可能ビット数を少なくすることで可能となりますが、この時この符号で誤り検出可能なビット数は（当然）増えることとなります。一般には、誤り訂正可能ビット数と誤り検出可能ビット数の割り当ては通信路の特性を詳細に調査すること及び実装可能な装置の規模などを検討して最適に割り当てます。（一例を挙げておきましょう： $(n, k) = (1023, 923)$ BCH 符号は最小距離 $d_{\text{mini}} = 21$ を有していて、ガウス雑音通信路で、5 bit 誤り訂正符号としてよく使われます。この時、残りの 15 bit が誤り検出可能な限界となります。すなわち、5 bit 以下の誤りの場合には受信データは復号サブシステム送り出され訂正され受信者に手渡されます。一方、6 bit 誤りから 15 bit 誤りは検出可能でその場合には受信機はNACKを返します。16 bit 以上の誤りに関しては検出器それ自身では判定を下すことができないため：シンドロームパターンから“ある”“なし”を認知できない：別途定められたシステムの手順に従うこととなります：多くの場合“不明”の選択肢がありNACKを返しますが、データの種類によって（第2回講座で挙げたAC VI, AC VO 等に適用可能か）は“捨てる”と云う選択肢もあります（“誤りがないかも知れない”し“あるかも知れない”：その意味では“16 bit 以上の誤りがある”時、と云う仮定は正確ではない）。誤り訂正と併用すると云う手順にも欠点があります。再送数を減らすためには、符号長 n を長くする必要があり、当然チェックビットが多くなり、その結果、送信時データのオーバーヘッドが増えます。これにより通信路の利用効率が劣化する結果となります。通信路の利用効率改善と通信の信頼性の検討はどこまでも面倒な trade-off をもたらします。どうしても通信路の特性把握は最重要課題です。再送におけるオーバーヘッドの増加を軽減して効率を上げようとするのが 次あげる Type II Hybrid ARQです。

3.2.2 タイプ II 型ハイブリッド ARQ (Type II Hybrid ARQ)

Type I ARQ では再送するデータは前回送ったものと同じ構成の（符号）データで一種類の符号化データですが、Type II ARQ では2種類の符号化データを用いて効率を改善します。最初に受信者に送信するデータは、誤りを検出するためだけに用いる高効率な (n, k) 符号 C_d で符号化されます。一方、誤りが検出されて再送するのはinvertible codeと呼ばれる符号化率 $1/2$ の特殊な $(2k, k)$ 符号 C_c のシンドロームパターンのみです。invertible codeでは誤り訂正と検出、両方の機能を使用します。図_Fig. 6dに half-rate($\eta=1/2$) invertible code を用いた場合の Type II Hybrid ARQにおけるデータ再送手順に関する大まかなフローチャートを示しておきます。ここで Tx：送信機, Rx：受信機, \mathbf{u} ：元データ系列, \mathbf{C}_d ： (n, k) 誤り検出符号, \mathbf{C}_i ： $(2k, k)$ invertible 符号, \mathbf{S}_{di} ：受信（誤り検出）符号 \mathbf{V}_d （の受信系列）から得られたシンドロームなどですが、黄色で着色された3個のブロックは全て受信者へ渡されるデータを表しています。図左側下方にあるオレンジのブロックは2回目以降のデータ送信へと繋がりますが、 $(f(\mathbf{u}), \mathbf{u}) = \mathbf{C}_d$ を送る場合と $(f[q(\mathbf{u})], q(\mathbf{u}))$ を送る場合に分かれるがその選択肢と宛先は図の構成上煩雑になるので省いてあります。



図_Fig. 6d half-rate invertible codeを用いた場合のType II Hybrid ARQデータ伝送手順の概要

図_Fig.6d から解るように, invertible code は元データ情報(\mathbf{u})と同じ冗長 (パリティ: $q(\mathbf{u})$) ビット数を持っていて, $q(\mathbf{u})$ を (n,k) 符号化して送られた $\mathbf{V}_d=f[q(\mathbf{u})]$ に誤りが無ければ $q(\mathbf{u})$ を用いて $\mathbf{C}_i/\mathbf{V}_d$ の inversion (回路) により高速に復号できます。このように, 再送時に \mathbf{C}_d を送るのでなく $q(\mathbf{u})$ を (n,k) 符号化した \mathbf{C}_i を送ることで誤り無く受信できた場合の復号を高速化できます。一方, 訂正可能誤りが検出された場合も, 勿論, inversionにより高速復号可能です。

本講座で開設予定の3.3 誤り検出・訂正符号に関する解説を省かせてもらいます。Type II Hybrid ARQに用いられている rate-1/2 invertible codeの復号を含む一般的な誤り検出・訂正組織符号 (systematic code) については Galois Field (ガロア体) の知識を基本としたBCH (Reed Solomon codeを含む) 符号に関する知識を必要とします。解りやすくコンパクトに3.3としてまとめるには大き過ぎる内容になると考えた次第です (機会があれば試みる価値はあるかとも思いますが, 面倒でもこれらに関する知識は専門の図書より得てください)。

また, 受信機が有限のバッファメモリで動作する環境での伝送手順についても解説を省いてしまいましたが, これについても後日機会があれば述べてみようと思います。但し, 近年の半導体微細化と高速化により有限バッファに関する注意の重要性は低下していると思われる (BAN応用での省電力という観点からはやはり重要度は高いとも云えます)。

これまで紹介してきたARQは基本的にはガウス雑音通信路 (Gaussian Noise Channel) を基本として開発された方式ですが移動通信環境 (Mobile Radio Channel) に代表されるフェージング特性を示す通信路ではこれらを基本として洗練されているが複雑な計算手順を要するARQが利用されています。TURBO codeやLDPC codeを誤り制御に用いたデータ伝送は3GやLTE Advanced LTE, 4Gなどの無線伝送物理層での通信プロトコルとして実用化されています。これらの誤り制御では puncturing により符号化の効率 (η) が可変であり, $\eta=1/m$ という符号化率可変対応 (rate compatible) な Type II Hybrid ARQが構築可能です。可変符号化率符号化としてはこの他 可変符号化率畳み込み符号 (rate compatible convolutional code) を用いて伝送路の時変特性で複数回の伝送が必要な場合には符号化率を変えて耐性を制御したり, 過去の (アナログ) 受信信号データを破棄せず合成して伝送回数を減らす方法なども実用化されています。これらの高度な伝送制御は更に改良されてcomplementary punctured convolutional codes を用いたWireless data ARQ として利用され Type III Hybrid ARQ とも呼ばれています。これらの高度化されたARQについては本稿の守備範囲を超えているので割愛します。local 5 Gのような最先端なワイヤレスネットワークを用いた医療・ヘルスケアシステムを構築しようとする技術者の方は必要な知識をARIB 標準仕様書 (日本国内向け開発の場合) や専門書で得てください。

all right reserved by NPO WBN