第３章 データリンク層

3.1 データリンク層の機能

データリンク層は，同じネットワーク内の隣接する通信機器（ノード）間の通信をサポートする。つまり同じネットワーク内であれば，データリンク層の機能だけで相手の通信機器（ノード）にデータを届ける事が可能である。データリンク層は，**LLC**（Logical Link Control）**副層**と **MAC**（Media Access Control：マック）**副層**の 2 つの副層に分けることができる。

2 つの副層のうち，上位層である LLC 副層は物理メディアに依存しない論理的な処理を行う。これに対して下位層の MAC 副層は，物理層とのデータの交換を処理し，物理層以下の通信メディアの違いを吸収する。この機能によりネットワーク通信では，LLC 副層以上のソフトウェアを変更することなく，MAC 副層以下のソフトウェアを変更するだけで，様々な通信メディアを使用することが可能となる。

3.1.1 LLC 副層の機能

LLC 副層の主な機能は以下の通りである。

1. **パケットデータの LLC フレームによるカプセル化とアンカプセル化**
2. **フロー制御**
3. **フレームシーケンス制御。**

LLC 副層では，LLC フレームによりパケットをカプセル化し，MAC 副層に引き渡す。LLC フレームのよるカプセル化を行うことにより，LLC 副層では**フロー制御**とフ**レームシーケンス制御**の機能を実現する。

フロー制御とは通信量を制御することであり，データ受信側のノードが受信した通信データ（フレーム）を処理しきれずにメモリ容量が少なくなったときに，送信側の機器に対して送信の停止を要求し，再びデータの処理が可能になった時点でデータの送信再開を要求する機能である。

フロー制御が行われないと，受信側のノードでは過負荷のためデータの消失（ロス）が多発し，データの再送要求（通常データの再送要求自体はデータリンク層より上位の層の機能である）などによりネットワーク上の通信量がさらに増大する。

ネットワーク上の通信量が増大し，ネットワークの性能が著しく低下するような状態を一般に「**輻輳**（ふくそう：Congestion)」と呼ぶ。「輻輳」の発生には幾つもの原因があるが，フロー制御が的確に行われない場合にも「輻輳」が発生し易くなる。

フレームシーケンス制御は，受信したフレームの順番が入れ替わっている場合に，フレームを順番通りに並べ替える機能である。ネットワークでは，データを一旦ネットワーク上の機器に蓄積してから中継を行う**蓄積交換型**なので，受信したフレームの順番が入れ替

わる可能性がある。一方，電話などは途中でデータの蓄積を行わない**回線交換型**なので，相手の音声が前後して聞こえるような現象は起こらない。

LLC ヘッダとしては，上位層のサービスの種類を表す **DSAP** (Destination Service Access Point，1Byte) と **SSAP** (Source Service Access Point，1Byte)，およびフロー制御やシーケンス制御を行うための **CTL** (Control，1 または 2Byte)が付加される (CTL の種類によってはシーケンス制御は行われない場合もある)。実際の使用では LLC ヘッダの後にさらに **SNAP** (Sub-Network Access Protocol，5Byte)ヘッダと呼ばれるデータを付加する (IEEE802.2 SNAP) (図 3.1)。 SNAP ヘッダは後述する EthernetⅡ (イーサネット) のタイプフィールドに相当する機能を持つ。

DASP(1)	SSAP(1)	CTL(1/2)	SNAP(5)	データ(パケット)

図 3.1 IEEE802.2 SNAP の LLC フレーム．( )内は該当フィールドの Byte 長

なおプロトコルによっては LLC 副層を利用しないものもあり，**TCP/IP プロトコル**のネットワークインターフェイス層である**イーサネット** (Ethernet) も LLC 副層を使用しない。従ってこれらのプロトコルでフロー制御およびシーケンス制御を行うには他の機能を利用するか，データリンク層より上位の層でこれらの制御を行わなければならない。

3.1.2 MAC 副層の機能

MAC 副層の主な機能は以下の通りである。

1. **上位層データの MAC フレームによるカプセル化とアンカプセル化**
2. **物理アドレスの割り当て (物理アドレッシング)**
3. **エラー検査**

MAC 副層は，前章のネットワークトポロジーなども含む，物理層以下の物理メディアの違いを吸収し，どのような物理メディアであっても同一のインターフェイスを上位層 (LLC 副層，もしくは LLC 副層を利用しない場合はネットワーク層) に提供する。

この機能は上位層データの，MAC フレームによるカプセル化とアンカプセル化機能により実現される。なおデータリンク層には，LLC フレームによるカプセル化，アンカプセル化機能と MAC フレームによるカプセル化，アンカプセル化機能があるが，**イーサネット**は LLC 副層の機能を利用しないので，TCP/IP を使用する場合は MAC 副層での MAC フレームがデータリンク層のフレームと同等であると考えてほぼ問題はない。

MAC 副層では同一ネットワーク内で通信を行うために**物理アドレス** (ハードウェアアドレス) の割り当ても行う。IEEE によるデータリンク層の規格では，この物理アドレスとして **MAC アドレス (マックアドレス)** を用いるように規定されている。後述する **IP アドレス**が論理的なアドレスであるのに対して，この MAC アドレスは物理的なアドレスであると言

われている。

エラー検査では受信したフレームが正しく受信されているかを検査する。もし雑音（ノイズ）などにより転送中にデータに誤りが発生している場合，フレームは破棄される。**イーサネット**では，**FCS**（Frame Check Sequence：フレーム検査シーケンス）と呼ばれるチェック用コードを利用した**巡回冗長検査**（Cyclic Redundancy Check: **CRC**）が行われ，フレームの正当性が検査される。

3.2 MAC アドレス

MAC アドレスは全体が **48bit** で，表記する場合は 16 進数 12 桁で表し，8bit（2 桁）毎に **：**（コロン）または **-**（ハイフン）で区切るのが最も一般的である（図 3.2）。 MAC アドレスは通常 NIC（ネットワークカード）やネットワークコントローラ毎に ROM に焼き付けられており（ハードウェアアドレスや物理アドレスと呼ばれる所以），その先頭 24bit（16 進 6 桁）は **ベンダーコード**（Organizationally Unique Identifier: OUI）と呼ばれ，IEEE によってネットワーク機器メーカ（ベンダ）毎に違ったものが割り当てられている。従って MAC アドレスの先頭 24bit を見れば，その NIC（ネットワークカード）やネットワークコントローラの製造メーカを知ることができる。ベンダーコードは IEEE のサイトなどで検索することが可能である

MAC アドレスは世界的に一意（ユニーク：同じものがないと言うこと）であるが，実際問題としては，同一ネットワーク内に同じ MAC アドレスを持つ機器がなければ通信は可能である。

ちなみに MAC アドレスが ROM に焼付けられていることから，MAC アドレスを偽装することは不可能であると思われがちだが，MAC アドレスを偽装することはそれほど難しいことではない。

00:16:76:C1:F0:8F

←----------→

ベンダーコード：(00:16:76 は Intel 社のベンダーコード)

図 3.2　MAC アドレス

MAC アドレスには **FF:FF:FF:FF:FF:FF** という**ブロードキャストアドレス**が存在する。多くの場合，通信は 1 対 1（**ユニキャスト**）で行われるが，「**ブロードキャスト**では，ネットワーク内の全てのノードに信号が届く」。

メディア（ケーブル）上を流れる信号は全て一旦受信側ノードのデータリンク層の MAC 副層に渡される。Mac 副層では受信した信号（フレーム）の宛先アドレスが自己の MAC アドレスに一致するか，またはブロードキャストアドレスである場合にはその信号を上位層に渡し，それ以外の宛先の場合は信号を破棄する。

従って，前述の「ブロードキャストでは，ネットワーク内の全てのノードに信号が届く」という表現は実際には不正確で，「ブロードキャストでは，ネットワーク内の全てのノードが（データリンク層より上位の層で）**信号を受信する**」という表現の方がより正確である。

ただし，特殊な場合として NIC やネットワークコントローラが**プロミスキャスモード**になっている場合には，MAC 副層での MAC アドレスの検査は行われず，全てのフレームは上位層に渡される。プロミスキャスモードはネットワーク上を流れるフレームの検査などを行う場合に利用されるモードである。

3.3 メディアアクセス方式

メディアアクセス方式とは，物理メディア（ケーブル）に対して，衝突（干渉）を起こすことなく，どのようにして信号を送信するかと言うことである。

人間ならば同じ部屋で何人もの人が話しをしていても，直接会話をしている相手の言葉を正しく聞き取る事が可能である。しかしながら，LAN の場合はネットワーク上に複数の信号があると信号間で衝突（干渉）が発生し，ノードはそれらの信号を正しく受信することはできなくなる。基本的にはネットワーク上では一度に発言できる（信号を送信できる）ノードは常に一台のみである。この信号が衝突を起こす範囲を**コリジョンドメイン**（Collision Domain）と呼ぶ。

MAC 副層でのメディアアクセス方式には大きく分けると 2 種類ある。**コンテンション**（Contention）**方式**と**トークンパッシング方式**である。コンテンション方式で最も一般的なものは**CSMA/CD**（シーエスエムエー・シーディー）方式であり，トークンパッシング方式で最も一般的なものはトークンリング方式である。

3.3.1 CSMA/CD

CSMA/CD（Carrier Sense Multiple Access with Collision Detection）は非常に単純な方式であり，以下の手順により信号を送信する。

1. **ケーブル上に他の通信機器の信号があるか調べる。**(Carrier Sense)
2. **信号があれば 5 へ。**
3. **信号がなければ，信号を送信。送信できたなら終了。**
4. **自分の送信した信号が他の通信機器の信号と衝突した場合は 5 へ。**(Collision Detection)
5. **ランダムな時間だけ待機して１へ。**(Multiple Access)

図 3.3 に CSMA/CD のアルゴリズムを**構造化**した場合のフローチャートを示す。構造化（プログラミング）については他の参考書を参照すること。

CSMA/CD は構造が単純で特別な制御システムを必要としないが，頻繁に信号の衝突が発生するので通信効率は悪く，ネットワークが混雑している場合には公称値の 40～50%程度の

通信速度しか出ないと言われている。通信効率が悪いと言う欠点はあるがその構造の単純さから，現在の殆どのネットワークではメディアアクセス方式として CSMA/CD が採用されている。**イーサネットのメディアアクセス方式も CSMA/CD** である。

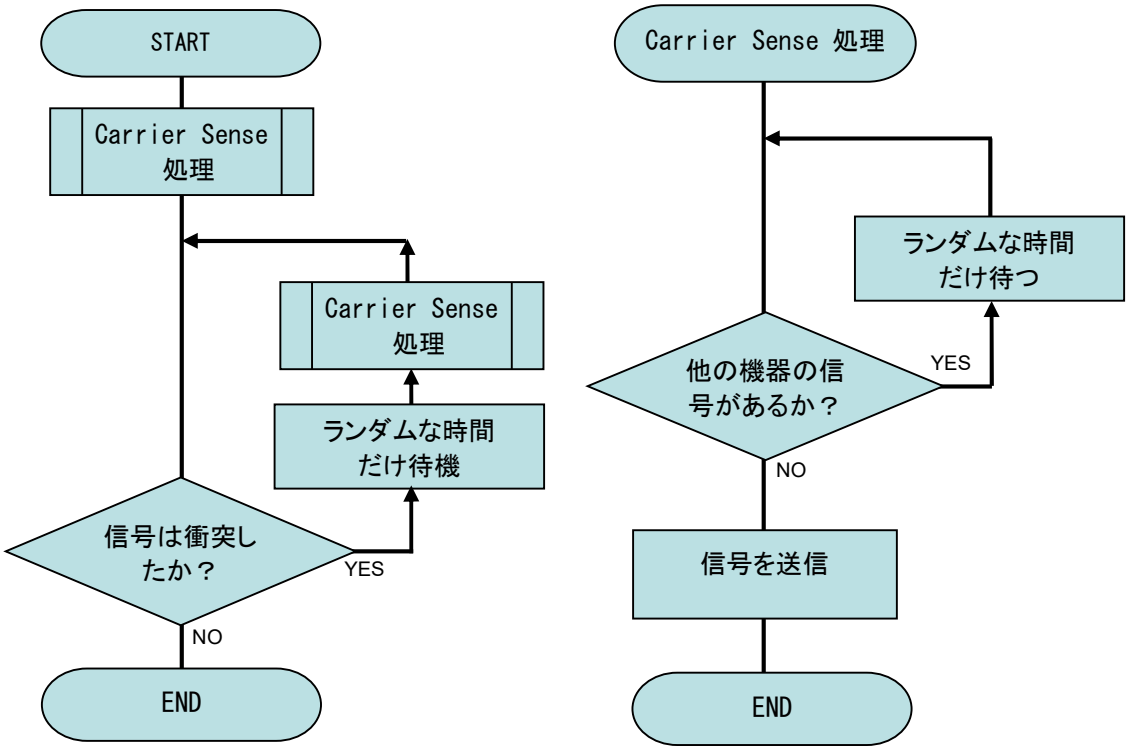


図 3.3 CSMA/CD アルゴリズムの構造化フローチャート

3.3.2 トークンリング

トークンリングは，トークン（発言権信号）と呼ばれる信号をリング状のネットワーク上で巡回させる方式である。手順は以下の通りである。

1. **トークン（発言権信号）をネットワーク上に流す。**
2. **トークンを捕まえた通信機器のみが信号を送信。**
3. **送信を終えたら（もしくは送信するデータが無いなら）トークンを次へ流す。**

トークンリングでは基本的に信号の衝突は発生しないので通信効率が高く，ネットワークが混雑していてもほぼ公称値の通信速度が出せると言われている。しかしながら，トークンを巡回させるための制御システムが必要であり，構造が複雑になる欠点を持つ。現在では殆ど使用されることはないが，以前は FDDI（光リング）や IBM トークンリングなどのネットワークで使用されていた。

3.4 中継器（スイッチングハブ）

データリンク層での中継機器は一般には**ブリッジ**と呼ばれるが，今日のスター型 LAN では多数の信号入出力用の通信ポートを持つ**スイッチングハブ**，または単に**スイッチ**と呼ばれるブリッジが使用されている。また，現在ではリピータハブが実際に使用されることは殆どないので，単にハブという場合でもスイッチングハブを指す場合が多い。

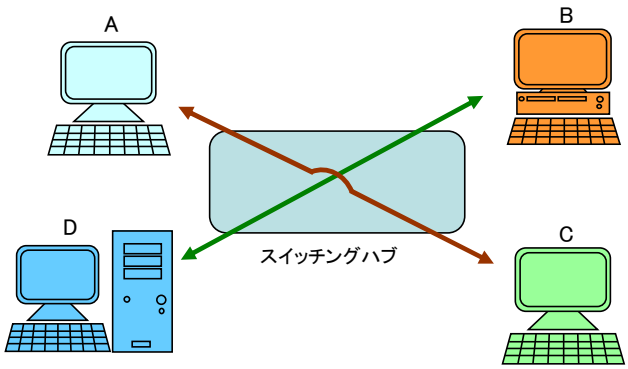
スイッチングハブは現在の LAN の構成において，最も重要な通信機器（中継器）の一つであると言える。

3.4.1 コリジョンドメインの分割

物理層で中継を行うリピータハブでは，一つの通信ポートから入力された信号は常に他の全ての通信ポートから出力される（**フラッディング**：Flooding)。一方スイッチングハブでは，初期状態では全ての信号（フレーム）がフラッディングされるが，その過程で各通信ポートから入力してくるフレームの送信元 MAC アドレスを学習し，通信ポートと MAC アドレスの対応をメモリ上に**MAC アドレステーブル**として保存する。

一つの通信ポート先に複数の機器が存在する場合には，一つの通信ポートに対して複数の MAC アドレスが対応付けられ，MAC アドレステーブルに格納される。学習後は各入力フレームに対して，宛先の通信機器が繋がっている通信ポートへのみフレームを転送する（**スイッチング**)。

従ってスイッチングハブに接続している各通信機器では，同時に 2 組以上の 1 対 1（ユニキャスト）通信が可能となる（リピータハブでは常に 1 組のユニキャスト通信しか可能ではない)。つまりスイッチングハブの内部では，それぞれのユニキャスト通信同士のフレームの衝突を回避することが可能であり，このことを一般に「**スイッチングハブはコリジョンドメインを分割する**」などと表現する（図 3.4)。ただし，スイッチングハブであっても，ブロードキャストフレームは常にフラッディングされる（通常はマルチキャストフレームもフラッディングされる)。



スイッチングハブでは，A と C，B と D が同時に通信可能
図 3.4 スイッチングハブ

なお，通信機器が繋がっているスイッチングハブの通信ポートを別の通信ポートに繋ぎ変えた場合，MAC アドレスの再学習が行われるので，通信が再開されるまで若干のタイムラグが発生する場合がある。

3.4.2 半二重と全二重通信

スイッチングハブを使用すると，その内部では，それぞれの通信機器（ノード）間のユニキャストの通信信号は衝突を起こさない。ただしこの状態であっても，ノードとスイッチングハブの間を一本の通信路のみで接続した場合には，お互いが同時に信号の送信を行うと通信路上でフレームの衝突が発生する。このように一本の通信路のみで通信を行うことを**半二重**（Half Duplex）通信と呼ぶ。半二重通信を行う場合には，CSMA/CD を使用して通信路上でのフレーム衝突を回避しなければならない。

これに対してノードとスイッチングハブ間（もしくはスイッチングハブ同士）を二本の通信路で接続し，通信路の一方を送信用，他方を受信用にすれば通信路上でのフレームの衝突は発生しない。このように通信を行うことを**全二重**(Full Duplex)通信と呼ぶ(図 3.5)。全二重では CSMA/CD によるアクセス制御は必要ない。なお，リピータハブでは全二重通信は使用できない。

現在のケーブルの主流であるツイストペアケーブルでは，1 本のケーブル内に複数の線芯があるため，通常は一本のツイストペアケーブルのみで全二重通信が可能である（ただし例外もある）。

アクセス頻度の高いサーバなどでは，スイッチングハブとの間を全二重通信で接続すると通信効率が向上する。

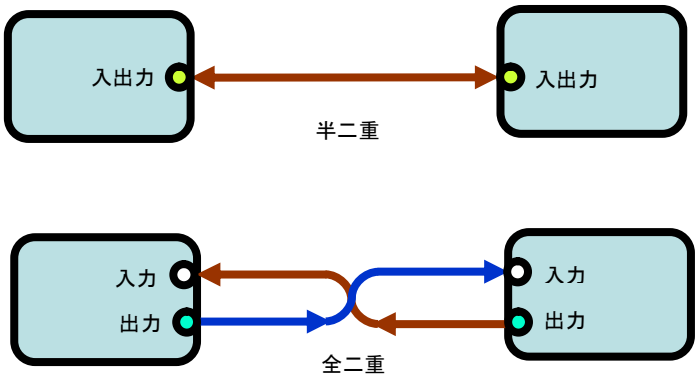


図 3.5 半二重と全二重通信

3.4.3 スパニングツリープロトコル

スイッチングハブの場合，リピータハブと違い，カスケード接続を行う場合の段数の制限はない。また接続を（物理的に）**メッシュ型**にすることも可能である。ただし，接続を

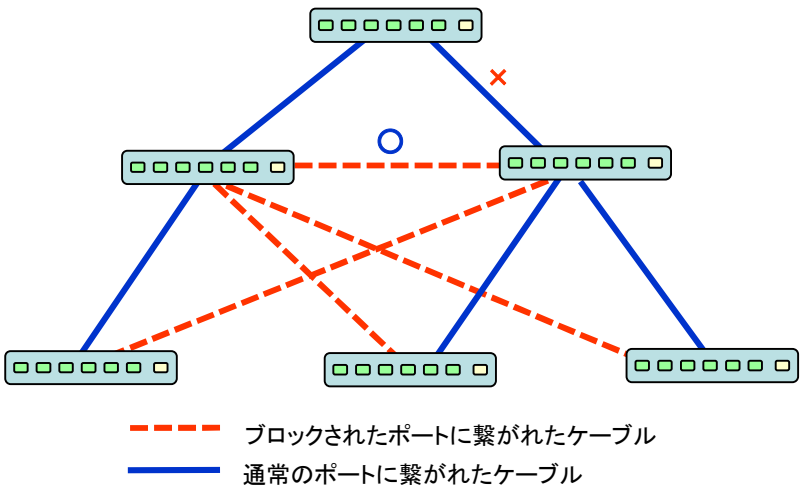
メッシュ型にすると経路がループ状になるため注意が必要である。接続経路がループを形成する場合，通信機器間で複数の通信路が確保され，ネットワークの冗長性が増し対障害性も向上する。

一方で，ループ経路が存在すると，ブロードキャストは常にフラッディングされるため，無限にネットワーク上を巡回し続けることになり，**ブロードキャストストーム**（ブロードキャストの嵐）が発生する可能性がある。一旦ブロードキャストストームが発生すると，ネットワークは完全に停止する。

またスイッチングハブでの MAC アドレステーブルの学習においても不都合が生じる可能性がある。ループ経路が存在すると，あるノードからのフレームがスイッチングハブの複数の通信ポートから入力してくるため，学習を正しく行えない（学習が収束しない）のである。

接続がループを形成していてもブロードキャストストームや MAC アドレスの学習不全を引き起こさないようにするためには，経路に優先順位をつけ，通常の通信では優先順位の最も高い経路のみを通信に使用し，その経路が切断した場合には次に優先順位の高い経路を使用するようにすれば良い。

スイッチングハブで使用される**スパニングツリープロトコル**は，このような処理を行う代表的なプロトコルである。スパニングツリープロトコルでは経路（スイッチングハブの通信ポート）に対して自動的に優先順位が付けられ，最も優先順位の高い通信ポート以外では通信がブロックされる。これによりスパニングツリープロトコルは（論理的な）経路のループ状態を回避する（図 3.6）。



例えば×のケーブルが切断した場合，自動的に ○のケーブルが繋がれた通信ポートが有効になる

図 3.6 スパニングツリープロトコル

3.4.4 ポートトランキング

スイッチングハブ間またはスイッチングハブとノード間の通信速度を上げる手法として**ポートトランキング**と呼ばれるものがある。これは対象機器間を物理的に複数のケーブルで繋ぎ，それらを論理的に束ねて一本のケーブルと見なす方法である。例えば，1Gbps のケーブル 3 本を利用して，それらをポートトラッキング機能を用いて一本のケーブルと見なせば，論理上 3Gbps のスピードのケーブルで対象機器間を繋いでいることになる（図 3.7）。

なおポートトラッキングは**リンクアグリゲーション**と呼ばれる場合もある。

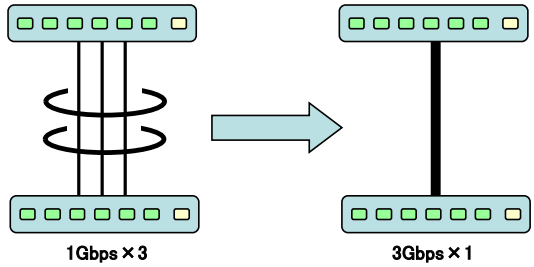


図 3.7 ポートトラッキング（リンクアグリゲーション）

3.4.5 スイッチによるフロー制御

TCP/IP では LLC 副層の機能を利用しないため，そのままではデータリンク層（イーサネット）でのフロー制御を行うことができない。そこでフロー制御をスイッチの追加機能として利用する場合がある。

通信が半二重で行われている状態で相手からの通信を一時的に止めたい場合は，CSMA/CD の衝突検出を逆に利用して，わざと信号を衝突させるのである。送信側は信号が衝突した場合，ランダムな時間だけ送信を中止するので，受信側に処理的な余裕が生じるまで信号を衝突させてやれば，結果的にフロー制御を行ったのと同等の効果を得る。この手法を**バックプレッシャ**と呼ぶ（図 3.8）。

一方，全二重通信を行っている場合はもっとスマートで，相手の送信を止めたい場合には中断時間を設定した **PAUSE フレーム**と呼ばれるデータを相手に送信する（図 3.9）。相手が PAUSE フレーム機能をサポートしていれば，「指定された中断時間×512 **ビット時間**」だけ送信を中断する。なお**ビット時間**とは 1bit を転送するのに必要な時間のことである。

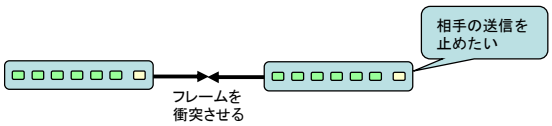


図 3.8 バックプレッシャ

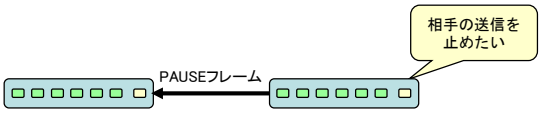


図 3.9 PAUSE フレーム

3.4.6 ポートミラーリング

スイッチングハブでは，送信先の通信機器が接続されている通信ポートにのみフレーム

を転送する。これによりフレームの衝突を回避し，さらには他の通信ポートでのフレームの盗聴を防止することが可能である。

しかしながらこの機能は，メンテナンスや実験，セキュリティチェックのために通信中のフレームの内容を検査したい場合などには非常に不便である。リピータハブであればこのような問題は発生しないが，問題の解決のためにスイッチングハブをわざと性能の劣るリピータハブに交換するもの考え物である。

そこでスイッチングハブでは，特定の通信ポートへ伝送されるフレームをコピーし，同時に他の通信ポートへの転送を行う**ポートミラーリング機能**を搭載している場合もある（図 3.10）。ただし，ポートミラーリング機能を使用して，特定ノードへ伝送されるフレームのコピーを他のノードで受信するには，その受信ノードの NIC を**プロミスキャスモード**にする必要がある。

通常の NIC のモードでは自分宛のフレーム以外は破棄してしまうが，プロミスキャスモードであれば，宛先に関係なく全てのフレームを受信することが可能となる。

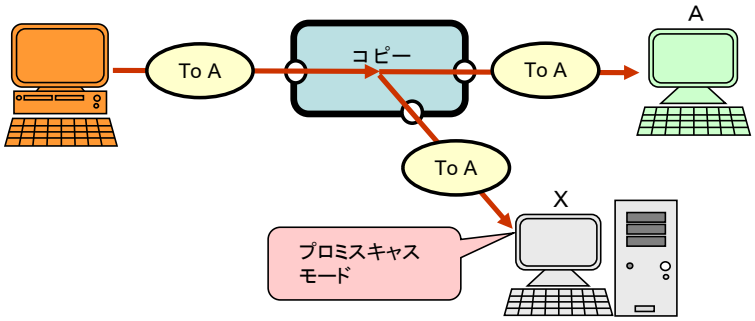


図 3.10 ポートミラーリング

3.5 データリンク層のプロトコル

データリンク層の代表的なプロトコルには**イーサネット**，HDLC，PPP，ISDN などがある。この内イーサネットは物理層とデータリンク層に跨るプロトコルであるので，次章で別に解説を行う。

3.5.1 HDLC（Highlevel Data Link Control）

HDLC は主に 2 台の通信機器（ノード）間のシリアル通信で使用されるプロトコル（厳密には 1 対多も可能）で，FCS（Frame Check Sequence）を使用した**巡回冗長検査（CRC: CRC-ITU-T）**を使用した強力な誤り制御機能や全二重通信機能を持ち信頼性の高い通信を行うことが可能である。

これ以前に使用されていたプロトコルである**BASIC 手順**と比べて任意のビットパターンを転送できるという利点があり，WAN などで使用される他，大型汎用コンピュータと端末を

繋ぐプロトコルとしても使用された。またこれ以降に開発された各種のプロトコルの原型にもなっている．図 3.11 に HDLC フレームの構造を示す。

HDLC では同期を取るための信号として 0x7E(0111 1110)を使用するため，転送データの中に 0x7E が存在する場合は問題となる。そこで HDLC では，データの送信時に 0 の後ろに 1 が 5 個連続した場合は 6 個目に強制的に 0 を挿入する。受信側では 1 が 5 個続いた後に 0 があった場合は，転送データあると判断しその 0 を取り除く処理を行う
（例：01111111... → 011111**0**11... → 送信 → 01111111...）。

フラグシーケンス(1)	アドレス(1)	制御(1/2)	データ	FCS(2)	フラグシーケンス(1)

- **フラグシーケンス： 同期を取るためのプリアンブルとポストアンブル．0x7E 固定.**
- **アドレス： 受信側または送信側のアドレス.**
- **制御： フロー制御を行うための情報.**
- **FCS： フレームチェックシーケンス．エラーを検出するための CRC(CRC-ITU-T)**

図 3.11 HDLC フレーム．( )内は該当フィールドの Byte 長

3.5.2 PPP (Point to Point Protocol)

PPP は 2 台の通信機器（ノード）間（Point to Point）を直接接続（2 点間接続）するための標準的なプロトコルである。PPP は回線の接続（データリンクの確立)・切断，認証機能などを持ち，様々な環境において 2 点間を直接繋ぐことができる柔軟なプロトコルである。PPP は **DHLC** を元にして作成され，HDLC 同様に任意のビットパターンを転送することが可能で，**FCS** を使った巡回冗長検査による強力な誤り制御機能も持っている（図 3.12)。

PPP の認証機能では **PAP**(Password Authentication Protocol: パップ)と **CHAP**(Challenge Handshake Authentication Protocol: チャップ)が使用可能であるが，PAP では入力したパスワードがそのまま通信路上を流れるので，特別な理由がない限りは使用すべきではない。一方，CHAP はランダムに生成した使い捨てのチャレンジキーを利用した，チャレンジ&レスポンス認証方法で，パスワードが直接通信路上を流れる事がないので比較的安全である。

PPP はブロードバンド常時接続環境以前に，モデムを使ったインターネットへのダイアルアップ接続で広く使用されていたプロトコルでもある。またブロードバンド常時接続環境においても以前のリソースを生かすために IPv4 による接続では主に **PPPoE**（PPP over Ethernet)が使用される。ただし IPv6 接続ではイーサネット上にそのまま IP を載せる **IPoE**（IP over Ethernet）が使用される場合が多い。（注：IPoE は何か特別な方式のようにも聞こえるが，PPPoE と差別化するために付けられた用語で，通常の LAN の形態である。）

PPPoE ではイーサネットのデータ部に PPP フレームが埋め込まれる（図 4.3 参照）．また PPP は様々なプロトコルデータをカプセル化（データ部分に格納）することが可能で， **VPN**（Virtual Private Network）などで**トンネリング**を行う場合に，カプセル化を行う一般的なプロトコルとして用いられる場合も多い。

フラグシーケンス(1)	アドレス(1)	制御(1)	プロトコル(1/2)	データ(0～1500)	FCS(2/4)	フラグシーケンス(1)

- フラグシーケンス： 同期を取るためのプリアンブルとポストアンブル．0x7E 固定．
- アドレス： 0xFF 固定．
- 制御： 0x03 固定．（フロー制御は行わない）
- プロトコル： データの種別を表す，1 または 2Byte のプロトコ番号．
- データ： デフォルトでは 1500Byte までのデータ．
- FCS： フレームチェックシーケンス．エラーを検出するための CRC-ITU-T または CRC-32

図 3.12 PPP フレーム．( )内は該当フィールドの Byte 長

3.6 誤り検出

ここで，特にデータリンク層に限定した話題ではないが，受信データの誤り検出についての解説を行う。

3.6.1 パリティチェック

パリティチェックではビットパターンのチェックを行う。検査に先立って，送信側と受信側で，チェックを何ビット毎に行うか（伝送ブロックのビット長），またパリティを偶(even）にするか奇(odd)にするかを予め決めておく。

送信側では伝送ブロックのビット列の最後にパリティビットと呼ばれる 1bit のデータを付加する。この時送信側受信側で決めたパリティが偶なら，パリティビットも加えた全体のビット列中の 1 の数が偶数になるようにパリティビットを設定する。同様にパリティが奇なら，全体の 1 の数が奇数になるようにパリティビットを設定する。

例えば，伝送ブロックが 7bit で，偶のパリティチェックを行う場合，送信する元データが 1011011（1 の数が 5 個)の時は，1 の数が偶数になるようにパリティビットを 1 とし，10110111（1 の数が 6 個）を送信する。受信側では，ブロック単位毎（この場合は 8bit）に 1 の数をかぞえ，1 の数が奇数ならデータ伝送中にノイズのため誤りが発生したことが分る。

一方，受信した伝送ブロックのパリティが予め定めておいたパリティと一致したとしても，「絶対に誤りが無い」とは言えない。この方式では，伝送ブロックのビット列中で偶数個のビットが反転した（誤りを起こした）状況を検出することはできないからである。また，誤りを検出した場合であっても，何ビット目が誤っているのかを検出することはできない（図 3.13)。

伝送ブロックサイズは 7bit，パリティは偶の場合

（送信側）1011011 + 1 ➔ 1011011 + 1（受信側）正常な伝送
（送信側）1011011 + 1 ➔ 1111011 + 1（受信側）誤りがあるのは分るが，
何処が誤っているのかは不明。
（送信側）1011011 + 1 ➔ 1101011 + 1（受信側）誤りがあるが，それを検出できない。

図 3.13 パリティチェック

パリティチェックの精度を上げるために，伝送ブロックのビット位置毎に，縦方向にパリティを計算する方法を併用する場合もある。このパリティチェックは**水平パリティチェック**と呼ばれる。これに対して，前述の横方向に計算するパリティチェックは**垂直パリティチェック**とも呼ばれる（図 3.14)。

パリティチェックは主に RS-232C によるシリアルデータ転送や，コンピュータの主記憶（メインメモリ）のチェック等で使用される。

【例】伝送ブロックサイズは 7bit，パリティは偶の場合

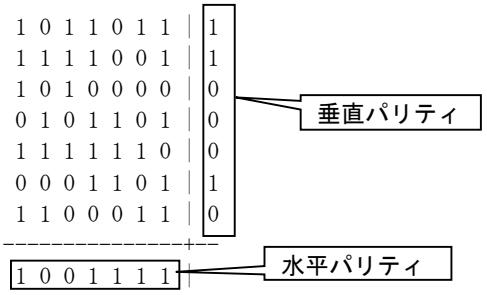


図 3.14 垂直パリティチェックと水平パリティチェック

3.6.2 チェックサム

チェックサムではあるデータ長毎に足し算を行い，その結果を検査に用いる。送信側と受信側では検査を行うブロックの長さ（伝送ブロック長）と検査の単位となるデータ長を予め定めておく。多くの場合では，ブロック長に 16Byte，データ長に 1Byte の値が使用される（つまり 16Byte のデータに対して，1Byte ずつ足し算を行う)。

送信側では，送信する伝送ブロックに対してデータ長毎に全て足し合わせ，結果（チェックサム）を伝送ブロックの最後に付加する。チェックサムは，必ずデータ長以下（先に述べたように通常では 1Byte) でなければならず，足し算の結果がそれ以上になった場合は，オーバフローした分を切り捨てる。

受信側でも受信データのチェックサムを計算し，それを送られてきたチェックサムと照合する。もしチェックサムが異なっていれば，受信データは誤っていることが分る（何処が誤っているかは分らない)。

チェックサムは，パリティチェックより高性能だが，前後のデータが入れ替わっているような，複数のデータが連続して誤りを起こしている場合（**バースト誤り**）には，誤りを検出できないことがある。

そのため，連続するバイトのチェックサム（横方向のチェックサム）だけでなく，伝送ブロック中のデータ位置に対する周期的なのチェックサム（縦方向のチェックサム）も計算する場合があるが，その分メモリが必要となる（図 3.15)。

【例】伝送ブロックが 4Byte，1Byte 毎に足し算を行う場合

《送信側》

送信元データ：A2 C1 22 15
チェックサム：A2+C1+22+15 = 19A → **9A**（19A 先頭の 1 は破棄）
送信データ： A2 C1 22 15 **9A**

《受信側１：15 が 11 に変化した場合》

受信データ： A2 C1 22 11 **9A**
チェックサム：A2+C1+22+11 = 196 → **96**（196 先頭の 1 は破棄）
チェック： 受信した 9A と自分が計算した 96 は一致しないので誤りを検出できる。

《受信側２：22 と 15 が入れ替わった場合》

受信データ： A2 C1 15 22 **9A**
チェックサム：A2+C1+15+22 = 19A → **9A**（19A 先頭の 1 は破棄）
チェック：計算結果も 9A なので，誤りを検出することはできない。

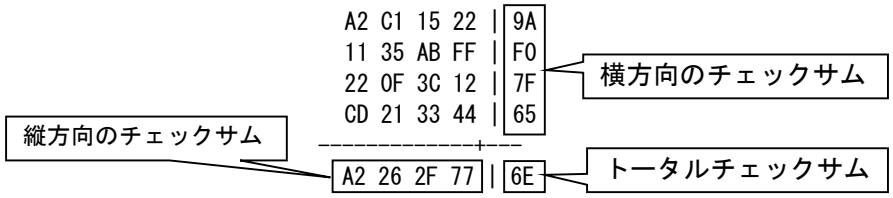


図 3.15 縦横方向のチェックサム

3.6.3 巡回冗長検査（CRC: Cyclic Redundancy Check）

チェックサムでは**バースト誤り**が発生している場合には誤りを検出できないことがある。バースト誤りを起こしている場合にも誤りの検出も行うには，**巡回冗長検査（CRC）**と呼ばれる手法が利用される。**巡回冗長検査**では多項式を用いて誤りの検出を行う。

巡回冗長検査は，チェックサムやパリティチェックよりも高性能で，計算の負荷もそれ程高くはないことから，多くの通信プロトコルで誤り検出の手法として利用されている。またパリティチェックは最も単純な巡回冗長検査であるとも言える。

3.6.4 巡回冗長検査の例

巡回冗長検査では，送信データを多項式の係数と見なし，それを予め定めた生成多項式で割って剰余（余り）を求める。さらにこの剰余（**BCC: Block Check Code**）を元のデータに付加（足し算）して送信する。

ただしここでの計算は通常の 2 進法の計算ではなく，**Modulo2** と呼ばれる手法で行われる。Modulo2 の計算方式では，元のデータに剰余（BCC）を足し算（付加）するということは，元のデータから剰余（BCC）を引き算しているのと同等である（なお Modulo2 の詳しい計算方法については別途参考書を参照すること）。

受信側では，もしデータに誤りがなければ，BCC を含む受信データの多項式を送信側と同じ生成多項式で割ってやれば，剰余は **0** になる筈である（ただし，剰余が 0 であるから

と言って，絶対に誤りがないとは言い切れない)。剰余が 0 にならない場合には，受信データには誤りがあることになる。

生成多項式には幾つか種類があるが，主なものを表 3.13 に示す。また，数値的に取り扱う場合には X に 2 を代入して CRC-8 → 0x107, CRC-16 → 0x18007, CRC-ITU-T → 0x11021, CRC-32 → 0x104C11DB7 を使用する。

名 称	BCC 長	生 成 多 項 式
CRC-6	6bit	X^6 + X^5 + 1
CRC-8	8bit	X^8 + X^2 + X + 1
CRC-16	16bit	X^16 + X^15 + X^2 + X + 1
CRC-ITU-T	16bit	X^16 + X^12 + X^5 + 1
CRC-32	32bit	X^32 + X^26 + X^23 + X^22 + X^16 + X^12 + X^11 + X^10 + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1

表 3.13 生成多項式

【例】生成多項式として CRC-6（X^6+X^5+1）を使用する場合

《送信側》

例として 8bit のデータ **0111 1010** の転送処理を行う。生成多項式 CRC-6 の BCC 長は 6bit であるので，その剰余データを付加するためにデータの最後に 6bit 分の 0 を追加する。

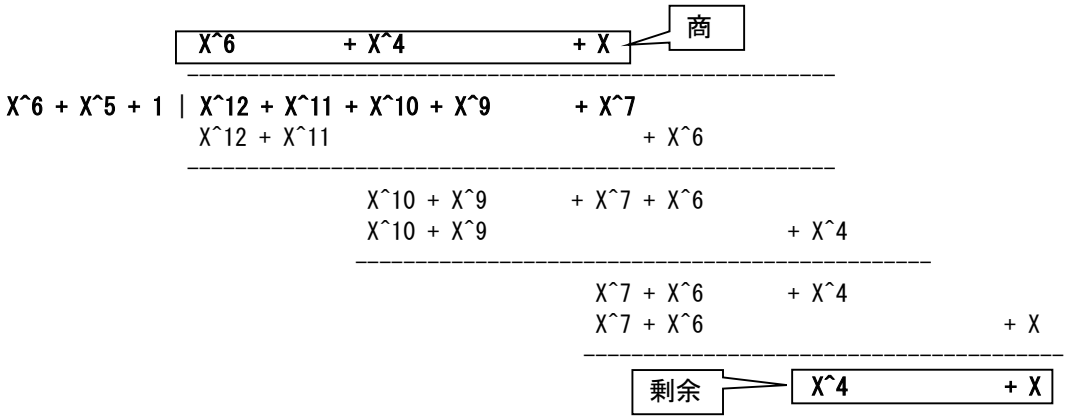
0111 1010 → 0111 1010 **000000**

次に **0111 1010 000000** を多項式に変換する。多項式は，ビット位置を累乗としたものを使用する。即ち 0111 1010 000000 は **X^12 + X^11 + X^10 + X^9 + X^7** となる。これを CRC-6（**X^6 + X^5 + 1**）で割って剰余を求める。

ただし，計算には Modulo2 を使用するので

0 + 0 = 0
0 + X = X
X + X = 0
0 - X = X
X - 0 = X
X - X = 0

となるので注意すること。



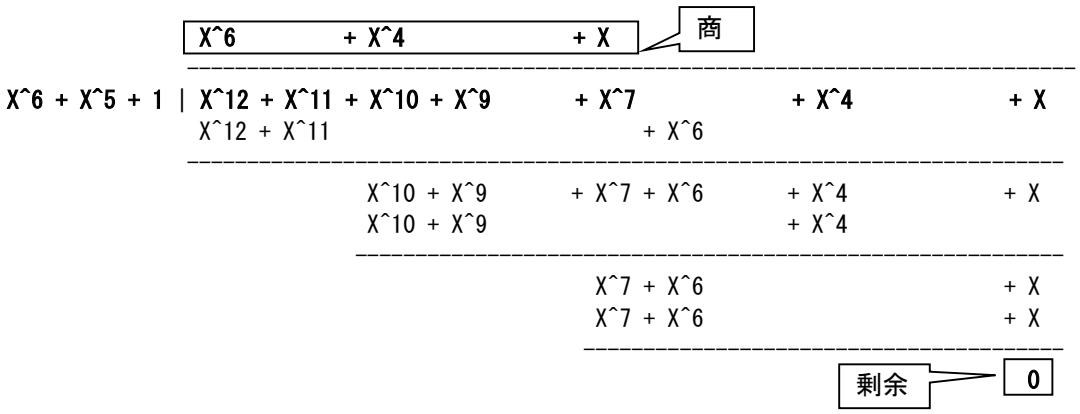
従って剰余（BCC）のビットパターンは 010010 となるので，これを元データから引いて

0111 1010 000000 - 010010 = 0111 1010 000000 + 010010 = **0111 1010 010010**

が転送するデータとなる。

《受信側》

受信側では。0111 1010 010010 を多項式に変換して (X^12 + X^11 + X^10 + X^9 + X^7 + X^4 + X)，これを CRC-6 (X^6 + X^5 + 1)で割って剰余を求める。



受信側では剰余が 0 となったので,このデータには誤りを発見できなかったことになる。

3.6.5 誤りの訂正

初心者がよく混同するのは,「誤りの検出」と「誤りの訂正」の違いである。一般に受信データの誤りを検出することはそれ程難しくはないが，その誤りを訂正することは非常に困難である。情報が欠落した状態（誤りのある状態）から，何の追加情報も無しに情報を生み出すこと（誤りを訂正すること）は，原理的に不可能だからである。それ故一般的なプロトコルでは，受信データの誤りを検出した場合にはそのデータを破棄し，送信元に対してデータの再送の要求を行うのである。

受信データの誤りを受信側で訂正することができるプロトコル（手法）も存在するが，

この場合は，本来のデータに誤りを訂正するための冗長な情報を常に付加して送信している。受信側はこの付加情報を使用することによって，ようやく誤りを訂正できるのである。この場合でも，誤りの訂正にはある種の制限事項が課せられるのが普通である。つまり無条件で，どんな場合にでも受信側で誤りを訂正できると言う事はまず在り得ない。