スイッチと VLAN

Ⅰ　スイッチ

LAN の中継機器として製品化されているスイッチには、様々なカテゴリのものが存在する。低価格化が進み、現在のネットワーク環境の基本である「レイヤ２スイッチ」、ルータに取って代わりつつある「レイヤ３スイッチ」、ｅコマースや高人気の Web サイトで必須のアイテムとなった「レイヤ 4／レイヤ７スイッチ」などである。そこで、現在の LAN 環境に定番のスイッチについて述べていきたい。

１　Ethernet と CSMA/CD

現在、ＬＡＮで最も利用されている通信方式は Ethernet で、事実上の世界標準となっているといっても過言ではない。Ethernet の中核技術は CSMA/CD であったが、スイッチ（スイッチング技術）の開発によって過去の遺物となりつつある。しかし、スイッチのことを知るうえで、CSMA/CD についての知識は必要となる。CSMA/CD について解説する。

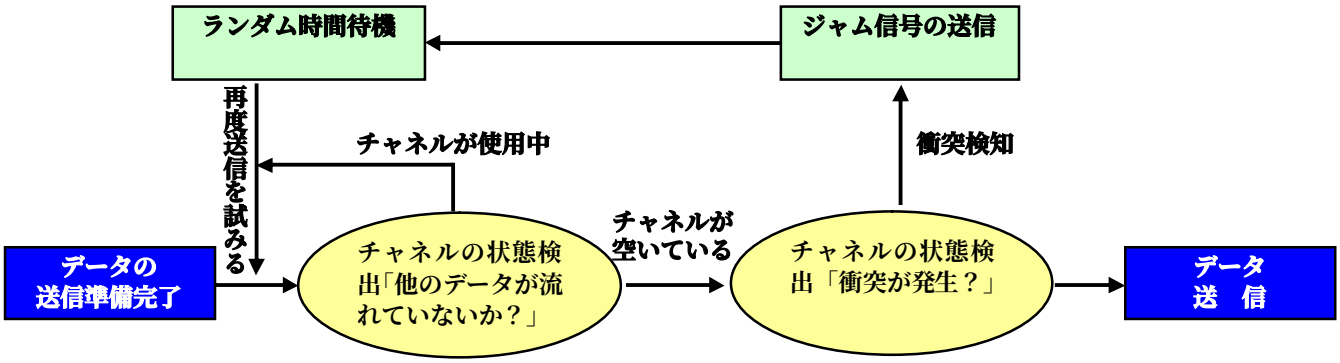
CSMA/CD は、

① Carrier　Sense　：話す前に聴け

② Multiple　Access　：静かなら話せ

③ Collision　Detection　：話しながら聴け

の３つのルールの頭文字を並べたものである。理論的には１本の伝送路（メディア）を複数のノード（通信端末）で共有しながら、ある瞬間には同時に１台のノードしかデータフレームを発信できないように制御するための仕組みである。（下図）



CSMA/CD では各ノードは通信を開始する前に、他のノードがケーブル上にフレームを送出していないことを確認する。その際、他のノードが送信したフレームが流れていると、それが消えてからランダムに時間待機して送信を開始する。フレームが電気信号として流れるため非常に高速に伝搬されるが、それでもネットワーク全体に信号が伝わるには一定の時間がかかる。このため、複数のノードが同時にケーブル上に信号を送出してしまうことがある。そして複数のノードが同時に送信を行うと、ケーブル内で信号のコリジョン（衝突）が発生してフレームが破壊され、データが正確に伝送されなくなる。これに対処するため、各ノードはコリジョンを検知したら送信を取り消し、再度データフレームを送信することになっている。コリジョンが発生するとケーブル内の電圧が上がりケーブルに沿って発信源まで戻ってくる（ジャム信号）ので、各ノードはデータの送信中でもケーブル上の電気信号をチェックしている。

CSMA/CD が確実に機能するためには、送信側ノードがフレームの送出を完了する前にコリジョンを検出できなければならない。そのため、電気信号がネットワーク内を往復する時間よりも長く、フレームを送出し続けることが要求される。そこで Ethernet では、データが少ない場合でも一定時間信号を流し続けるよう、フレームの最小長を 64 バイトと定めている。

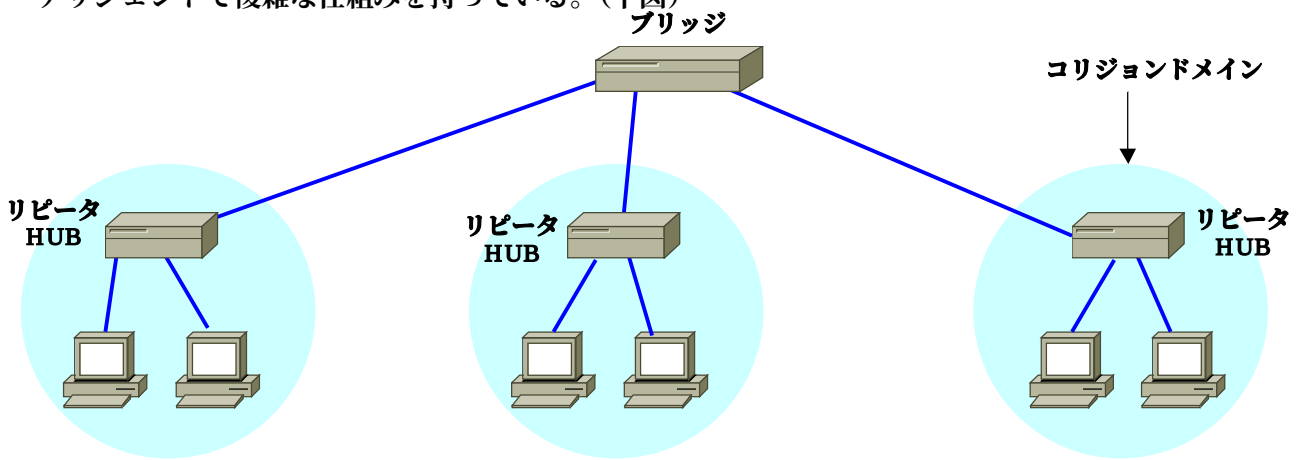
2　リピータとコリジョンドメインの問題

Ethernet では、伝送メディアとして同軸ケーブルやツイストペアケーブル（ＳＴＰやＵＴＰ)、光ファイバなどを利用しているが、伝送路内で信号が減衰することを考慮して、メディアごとに最大長が定められている。しかし、現実にはその最大長以上にネットワークを拡張する必要がある。ネットワークを拡張するための最も単純な中継機器が「リピータ」である。リピータは、２本のケーブルの中継点として設置され、減衰した信号を増幅、整形して転送を行う。また、ツイストケーブルを使った 10BASE-T や 100BASE-TX など、スター型のトポロジーを利用する Ethernet では HUB を使ってネットワークを構築する。HUB は多数のポート間でリピータの機能を持った中継装置（マルチポートリピータ）である。
リピータや HUB の動作の特徴は、任意のポートに接続されたノードから送信されたフレームが、他の全てのポートに流される。８ポートのリピータ HUB であれば、１つのポートに送信されたフレームは、他の７つのポート全てに流される。リピータやリピータ HUB で構築されたネットワークでは、論理的には１本の伝送路に全てのノードが接続されている以前の同軸ケーブルを使った Ethernet と同じである。つまりネットワーク全体でコリジョンが発生するために、同時に１台のノードしかデータを送信できない。この CSMA/CD 方式で利用するコリジョンを検知できる範囲（ドメイン）は「コリジョンドメイン」と呼ばれ、１つのコリジョンドメイン内で同時にデータを送信できるノードは１台だけなので、狭いコリジョンドメインの方がネットワークの効率はよいことになる。先で記述したリピータ HUB でネットワークを拡張すると、ノードが増加することによりコリジョンドメインが膨張する。これによりコリジョンが発生する確率が増え、伝送効率が低下する。

3　ブリッジからスイッチへ

インターネットの普及、ファイルサーバやプリントサーバの導入などにより、校内 LAN や社内 LAN へ接続される機器が増加し、ネットワーク内を流れるデータ量は加速度的に増えている。データ量が増えるということは、コリジョンが発生する確率も増えるということである。伝送効率を上げるには、コリジョンドメインを分割する中継機器を用いる必要がある。

コリジョンドメインを分割する中継機器として、「ブリッジ」が Ethernet と同じくらい前から存在する。リピータが電気信号を単純に中継するのに対して、ブリッジは Ethernet 上の他のノードと同じように CSMA/CD の手順を踏み、データフレームが確実に送受信されるように働く。ブリッジは、中継先のネットワークで別のノードがデータを送信中であれば、それが終了するまで送信を待つ。このためブリッジには、送信待ちのデータフレームを一時的に格納するバッファメモリが搭載されている。ブリッジは複数のネットワークを別々のコリジョンドメインとして相互接続するために、リピータよりもはるかにインテリジェントで複雑な仕組みを持っている。（下図）



ブリッジの中には、フレームを受信するたびに送信元の MAC アドレスを記憶する「ラーニング機能」を備えた「ラーニングブリッジ」と呼ばれる製品がある。ラーニングブリッジは、学習した MAC アドレス情報に基づいて、任意のポートで受信した Ethernet フレームを、どのポートへ中継するかを判断する。

このラーニングブリッジを進化させたものが「スイッチ」である。最初のスイッチはマルチポートブリッジの一種で「スイッチングハブ」とも呼ばれていた。古典的なマルチポートブリッジは、同時に１対の２ポート間の中継しかできないのに対し、スイッチングハブでは同時に複数対のマルチポート間の中継が可能になった。

また、スイッチングハブは、製品アーキテクチャがブリッジとは大きく異なり、ブリッジの実態は汎用マイクロプロセッサ（パソコンなどの CPU に使われるチップ）の上で動く、フレーム解析ソフトウェアである。これに対しスイッチングハブは、ロジックが組み込まれた専用チップを利用している。つまり、フレームの解析と転送を、ソフトウェアではなくハードウェアで行っている。このため、処理速度は桁違いに向上した。専用チップを採用し、ハードウェア的にフレーム解析と転送処理を行うアーキテクチャは、現在「スイッチ」と呼ばれる製品群の特徴である。

４　スイッチの基本はレイヤ２スイッチ

スイッチは送信元と宛先の MAC アドレスを見て特定のポートにのみフレームを転送し、複数の伝送路を同時に利用できる。また、スイッチは OSI 参照モデルのデータリンク層での中継機能を担っているため、レイヤ２スイッチと呼ばれている。

５　スイッチの転送機能

スイッチの Ethernet フレーム転送(フォワーディング)方法は、大きく分けて次の３種類が挙げられる。

①　カットスルー

カットスルー方式は、フレームを宛先アドレスまで受信したら宛先ノードの接続されたポートへ転送する。有効なアドレスが宛先に指定されているフレームは全て転送され、遅延はほとんど発生しない。ただし、壊れたフレームも転送するという問題が生じる。

②　修正カットスルー

修正カットスルー方式は、Ethernet フレームの先頭 64 バイトを受信するまでは転送を開始しない。64 バイトは Ethernet フレームの最小長であり、コリジョンにより発生するエラーフレームのほとんどは、64 バイト以下であることが知られている。すなわち、最初の 64 バイトを検査すれば、ほとんどのエラーフレームを除去できる。この方式は、遅延対策とエラー検査を両立させたものである。

③　ストア＆フォワード

ストア＆フォワード方式は、受信したフレーム全体をバッファに全て格納してから別のポートへ送り出す。フレーム全体を受信するため、Ethernet フレームの最後に付属するエラーチェック用のＦＣＳを確認でき、エラーが存在する「壊れたフレーム」を破棄できる。しかし、他の方式と比較すると遅延が大きくなる。

Ethernet の伝送速度が 10Mbps のころは、遅延が大きな問題であった。しかしこれまで、スイッチに搭載されるバッファメモリの性能が向上してきたことにより、遅延は無視可能なレベルとなった。さらに、10/100Mbps 対応など伝送速度が異なるポートをもつスイッチでは、速度差を吸収するためにストア＆フォワード方式が事実上の業界標準となっている。

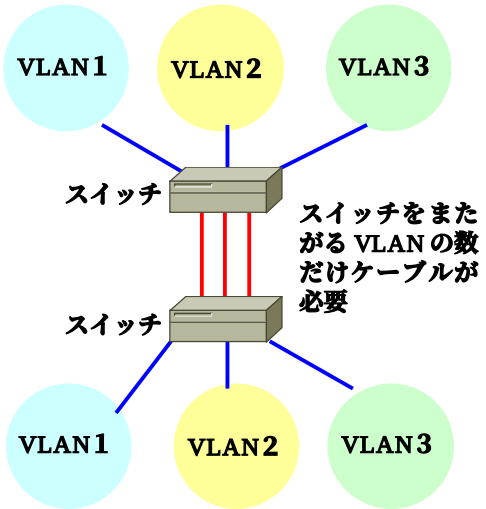
6　異なるＬＡＮに分割するＶＬＡＮ機能

ＶＬＡＮ（Virtual　LAN：仮想 LAN）とは、物理的に同一のスイッチに接続されているノードを、論理的に（仮想的に）異なるＬＡＮに分割する機能である。VLAN 機能はレイヤ３の代表的な機能として紹介されることが多いが、VLAN 自体はレイヤ２スイッチの機能だけで実現できる。

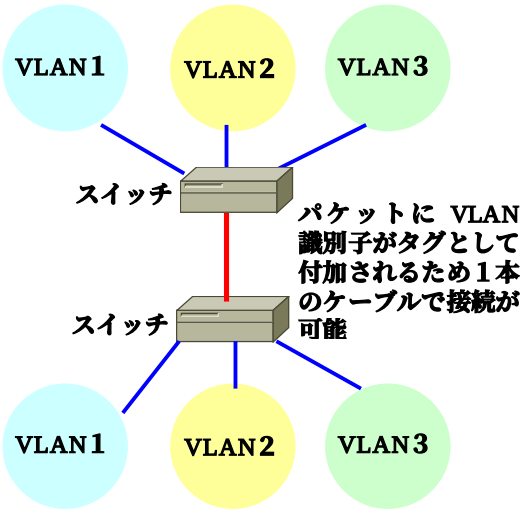
レイヤ２スイッチで VLAN を構築する場合、最もよく使われるのは「ポートベース VLAN」と呼ばれる機能である。これは単純にレイヤ２スイッチのポート単位で VLAN を分けるものである。

また、複数のフロアや建物に分散したワークグループを構築するといったことも、レイヤ２スイッチの「VLAN タギング」機能を使えば可能である。VLAN タギングとは、Ethernet フレームに VLAN 識別子として 4 バイトの「タグ情報」を付加する仕組みである。VLAN タギングに対応したスイッチ間では、タグ情報の付加された Ethernet フレームが送受信される。受信側のスイッチは VLAN 識別子を見て、目的の VLAN（ポートグループ）にのみフレームを転送するようになる。これにより、複数のスイッチで VLAN を構築する際にメリットが生まれる。ポートベース LAN では、複数のスイッチで VLAN 情報を共有する場合、VLAN の数だけ接続ケーブルが必要になる。しかし、VLAN タギングを利用すれば接続ケーブルは１本ですむ。（下図）

VLAN タギング未対応のスイッチの場合



VLAN タギングに対応したスイッチの場合



Ⅱ　VLAN

本校の校内ネットワークを教員用と生徒用のセグメントに分ける手法として VLAN でセグメントを分ける方法が考えられる。そこで、校内ネットワークを VLAN で教員用と生徒用とにセグメント分けする方法について記述していく。

1　VLAN（バーチャル LAN）とは

VLAN とは、スイッチングハブの各ポートを複数のグループに分け、それぞれのグループを独立したサブネットとして機能させる仕組みのことである。

スイッチングハブでは全てのポートが１つのネットワークに属しており、どのポートからでもほかのポートにパケットを送ることができる。これに対して VLAN 機能を持つスイッチングハブでは、仮想的な複数のサブネットを１台のスイッチングハブに設定することができる。例えば、8 ポートの VLAN 対応スイッチングハブの 1～4 ポートを VLAN１に、5～8 ポートを VLAN２に設定するといったことが簡単

にできる。このように設定すると、片方の VLAN 内のパケットはもう片方の VLAN には中継されることはない。これにより、１つのスイッチングハブを利用するだけで、１つのネットワークを 2 つのネットワークに分けて利用することができる。つまり、複数のポートを論理的なグループにまとめ、グループ内だけの通信を可能にすることができる。

2　VLAN の種類

VLAN には以下のような種類がある。それぞれの特長について説明する。

① ポート VLAN
パケットを受信したポートにより VLAN を決める方式

② タグ VLAN
パケット内のタグに指定された番号により VLAN を決める方式

③ MAC アドレス VLAN
送信元の MAC アドレスにより VLAN を決める方式

④ プロトコルベース VLAN
プロトコルの種類（IP、IPX、AppleTalk 等）により VLAN を決める方式

⑤ レイヤ３ネットワーク VLAN
レイヤ３のネットワーク情報により VLAN を決める方式

3　VLAN の考え方

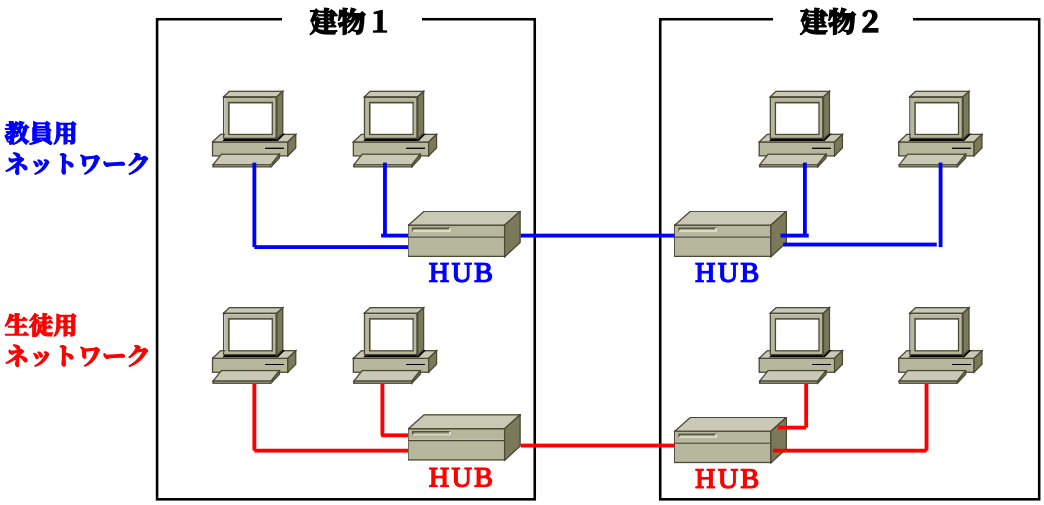


図　1

HUB を用いて全教員用のパソコンと生徒用のパソコンを接続する。（教員用と生徒用のネットワークを別々にする）しかし、この場合は建物間の接続用として２本の LAN ケーブルと４台の HUB が必要となる。

上記の図１では教員用のネットワークと生徒用のネットワークは物理的に独立している。次の図２では、VLAN を使った物理的構成である。

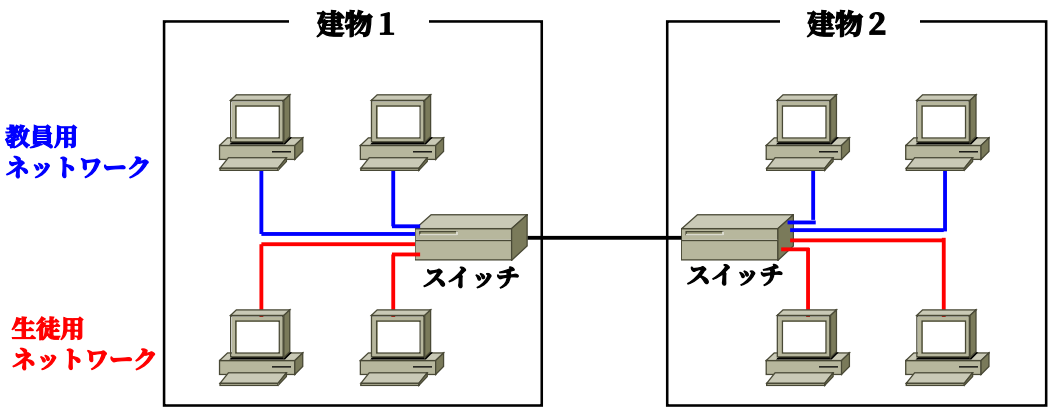


図 2

図２では建物間の LAN ケーブルが１本になり、図１の HUB がスイッチングハブに変更されている。これだけでは物理的に全てのパソコンが１つのネットワーク上に存在しているようになる。しかし、HUB をスイッチングハブに変更することにより、物理的構成を論理的構成に変更できる。また、スイッチングハブを使うことにより、VLAN ごとにポート分けができる。その構成を図３に示す。

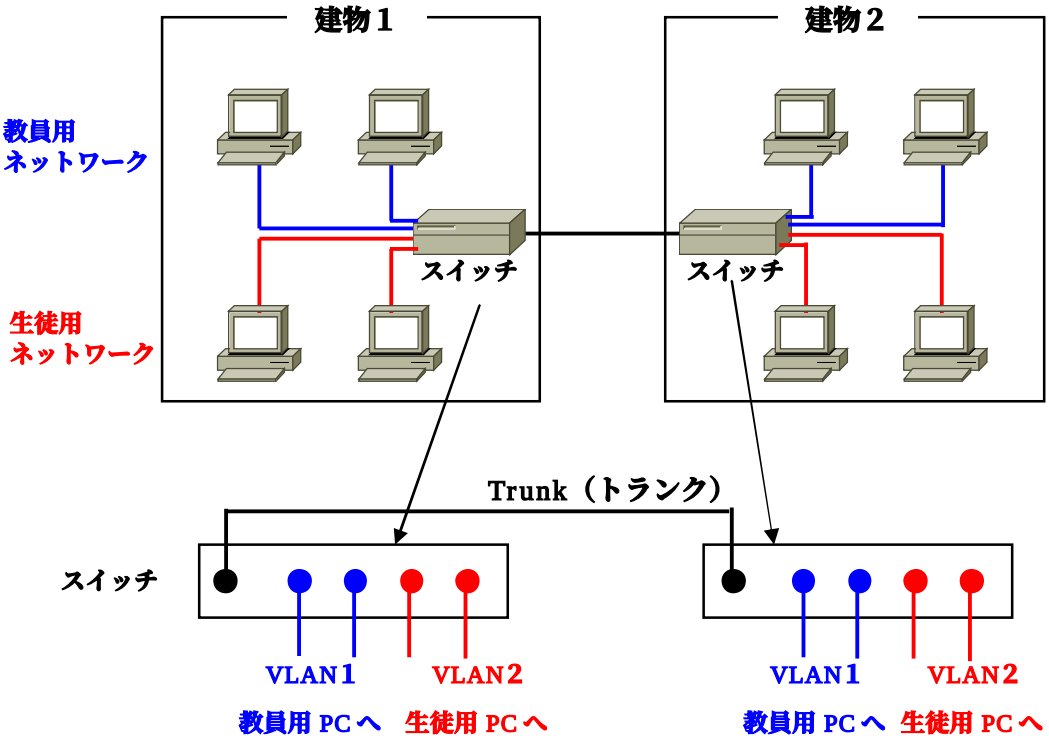


図 3

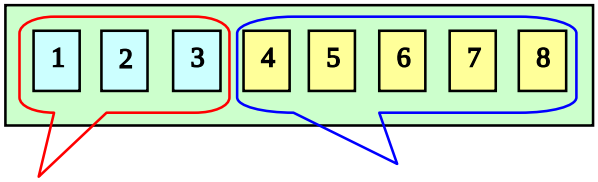
このように、２つのスイッチングハブの Trunk（トランク）を LAN ケーブルで結び、VLAN１を教員用のパソコンに、VLAN２を生徒用のパソコンに接続すればよい。ここでは、スイッチとスイッチを結ぶため、LAN ケーブルはクロスケーブルを使う。

VLAN の設定は、スイッチでソフトウェアを介して行う。VLAN は標準化されていないため、スイッチ・ベンダー独自のソフトウェアを使うことになる。

3　VLAN の作成

スイッチの設定は、シリアルポートに管理用コンピュータを接続し、そこからハイパーターミナルなどのターミナルソフトを使って行うのが一般的である。VLAN の作成については製品によって若干異なるが、およそ次のようになる。

まず、新しい VLAN の名前（ここでは生徒用を「Students」）を定義し、そこに識別番号の VID を設定する。VID に使える番号は、スイッチがサポートしている VLAN 数だけであり、すでに割り当てられていないものであれば、基本的に何を割り当ててもよい。ここでは「2」を割り当てている（1はデフォルト VLAN の VID)。あとは、その VLAN グループに属するポートを番号で指定する。ここでは「1－3」を指定している。



グループ１
（新しい VLAN＝生徒用）

グループ２
（デフォルト VLAN＝教員用）

```
VLAN Name      : Students
ID             : 2
Ports on VLAN  : 1-2
Tagged VLAN    : Null
```

1　VLAN を「Students」という名前で作成する
2　VLAN の ID 番号を「2」にする
3　所属ポートを「1,2,3」に指定する
4　タグ VLAN を使うかどうか
　（Null は使わない）

図 4

新しい VLAN グループを１つ作成すると、１台のスイッチを２台に分割したのと同じ状態になる。同じ VLAN グループに属するポート同士は直接通信できるが、異なる VLAN グループに属するポート間はルータでルーティングしなければ通信できない。

図４では、スイッチごとにグループを作成するので特に「ポート VLAN」と呼ばれる。ポート VLAN では、ポートごとに VLAN グループを割り当てていく。そのため複数のスイッチをまたぐような VLAN を作る場合、スイッチ間を接続する回線は VLAN グループの数だけ必要になる。例えば、スイッチ１とスイッチ２にそれぞれ VLAN-A 用と VLAN-B 用の２本の回線でつなげなければならない。このようなスイッチ間の回線は、ポート VLAN を発展させた「タグ VLAN」（トランク VLAN、IEEE802.1Q）を使うことで１本に束ねることができる。

タグ VLAN は、Ethernet のデータ（フレーム）に「タグヘッダ」と呼ばれる独自情報を挿入し、それに基づいてスイッチングすることで実現される。タグヘッダは４バイトの情報で、この後半に VID が収められている。タグ VLAN 対応スイッチはこの情報を認識することができるので、同じ回線に複数の VLAN のフレームが混在しても正しいあて先に中継される。

タグ VLAN の設定は VLAN の作成時に行う。具体的には、特定のポートを複数の VLAN グループに属するように設定した後、そのポートをタグヘッダ付きのフレームを送受信できるポートとして指定する。

図５でいえば、スイッチ１の「ポート８」は VLAN グループ「Teacher」（教員）と「Students」(生徒) の両方に属するようにして、「Tarred VLAN」でタグ VLAN 対応ポートに指定している。同様にスイッチ２では「ポート１」をタグ VLAN に対応させている。以上の設定で、スイッチ１のポート８とスイッチ２のポート１は、Teacher VLAN のフレームも Students VLAN のフレームも送受信できるようになるので、このポートでスイッチ同士を接続すればよい。

ポート VLAN とタグ VLAN を同時に使うことで、配線をシンプルに、かつ柔軟にできるようになる。

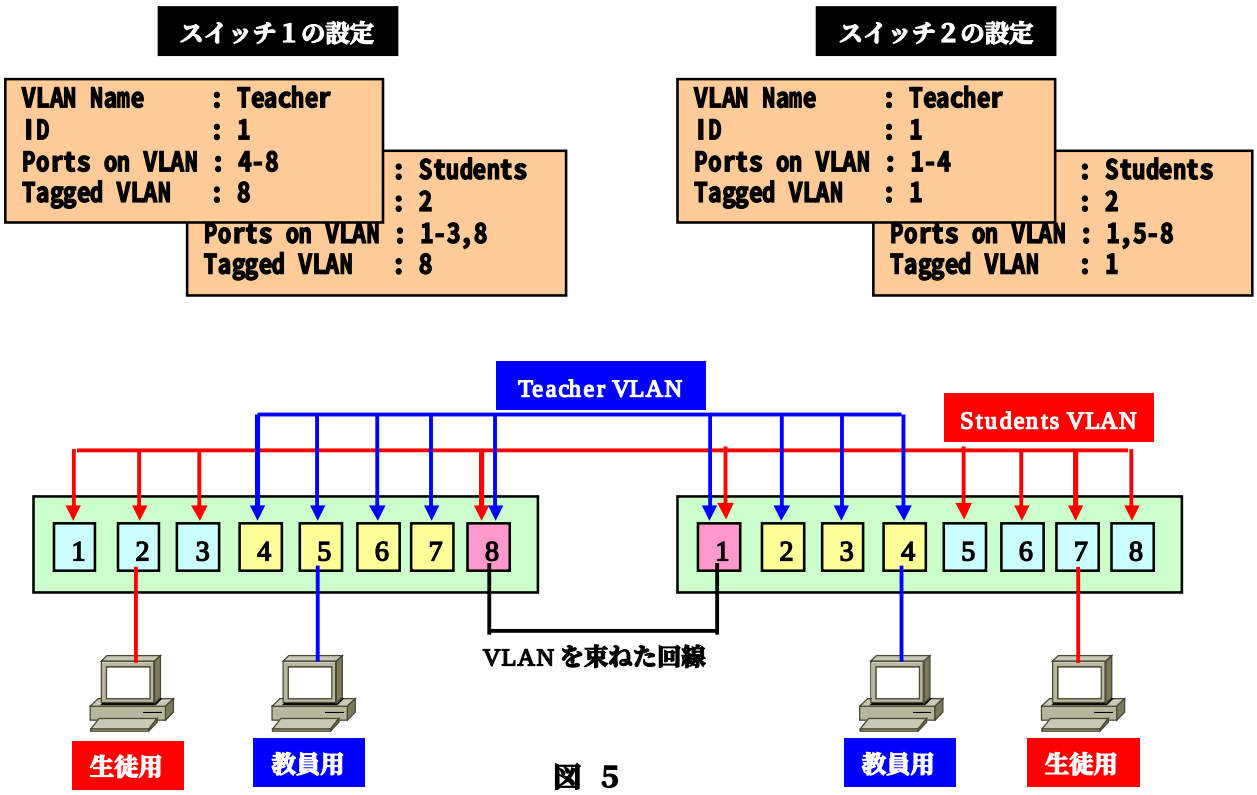


図 5

4　本校の校内ネットワークへの導入案

本校の校内ネットワークに VLAN を導入する際、教員用のセグメントと生徒用のセグメントが存在するが、２つのセグメントが混在する建物として、パソコン室がある産業教育振興棟（以下　産振棟）があげられる。そこで、産振棟を例に上げ構築例を記述していく。

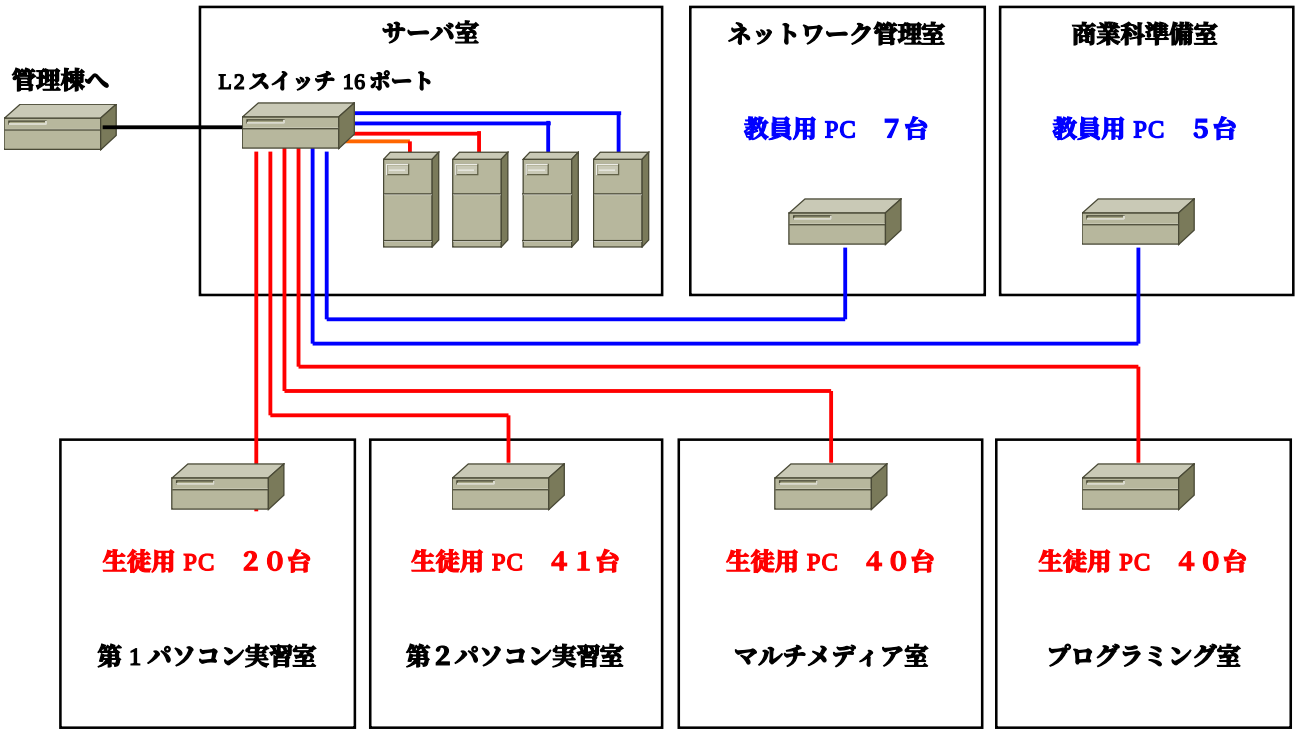


図 6

図６は本校の産振棟内の状況である。管理棟から光ファイバケーブルが産振棟のサーバ室まで来ており、メディアコンバータを経由してスイッチへ入り、そのスイッチから各パソコン室とネットワーク管理室、商業科準備室へ接続されている。インターネットに接続するためには管理等にあるルータから、ＶＰＮ経由で福岡県教育センターへ接続し、そこからＷｅｂの閲覧ができる。

サーバについては産振棟のサーバ室に設置し、２台を生徒用とし１台をファイルサーバ、もう１台をDHCP サーバとしている。残りのサーバについては教員用のファイルサーバとして活用している。

今回はそのスイッチの VLAN 機能を使い VLAN を構築する。図７はサーバ室にあるレイヤ２スイッチの接続状態を示している。

産振棟内の設定

VLAN１の構成 （Teacher）	VLAN２の構成 （Students）	VLAN３の構成 （Common）
・教員用ファイルサーバ１へ ・教員用ファイルサーバ２へ ・ネットワーク管理室へ ・商業科準備室へ	・第１パソコン実習室へ ・第２パソコン実習室へ ・マルチメディア室へ ・プログラミング室へ ・生徒用 DHCP サーバへ	・生徒用ファイルサーバへ

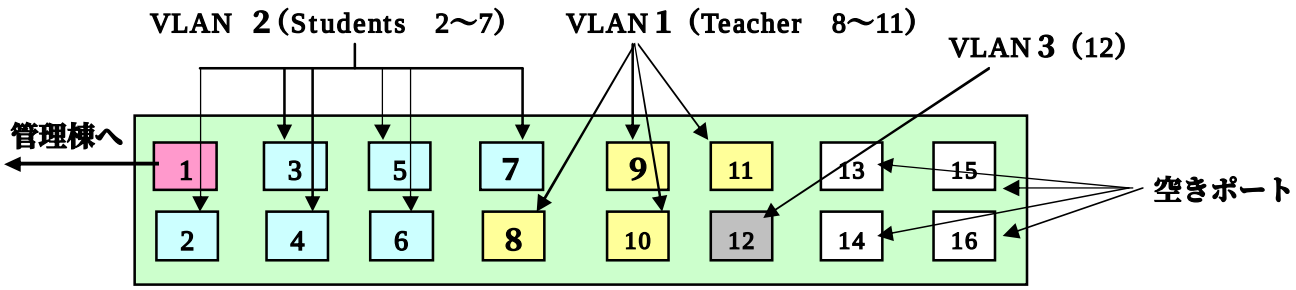


図７　タグ VLAN

産振棟側スイッチの設定

VLAN Name : Teacher ID : 1 Ports on VLAN : 1,8-11 Tagged VLAN : 1	VLAN Name : Students ID : 2 Ports on VLAN : 1-7 Tagged VLAN : 1	VLAN Name : Common ID : 3 Ports on VLAN : 1-12 Tagged VLAN : 1

※ 上の図は産振棟側のスイッチの設定（タグ VLAN）
左から VLAN１、VLAN２、VLAN３の設定例

図８

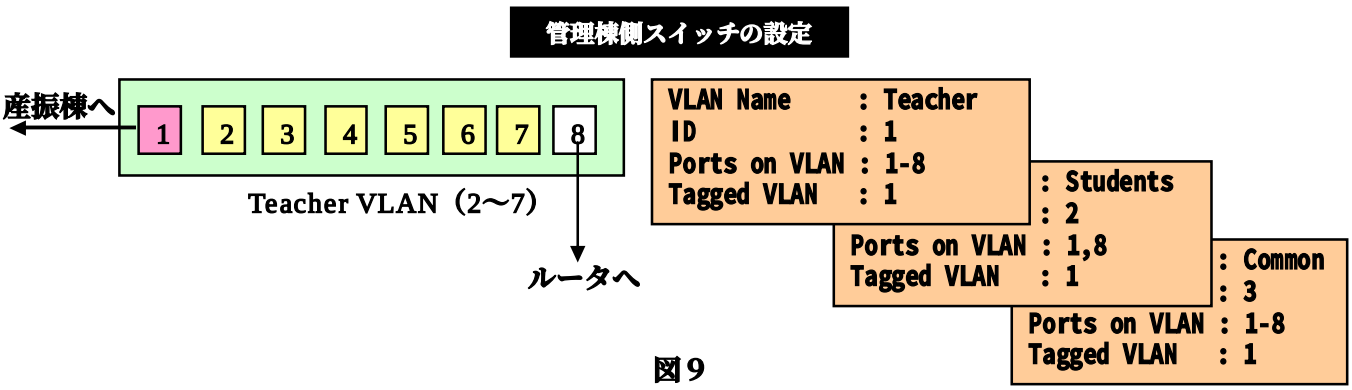


図９

図８は産振棟サーバ室のスイッチの設定と図９が管理棟にあるスイッチの設定である。管理棟には生徒用のポートは必要ないが、インターネットができなくなると困るのでタグ VLAN をポート 1 に設定し、ポート８はルータへとつなげるために設定する必要がある。また、職員室内から生徒用のファイルサーバにアクセスできるようになる。

空きポートについてはスイッチのメーカによるが、ほとんどがデフォルトでＶＬＡＮ１となっている。そこで空きポートを使用する際は、スイッチの設定を再度行う必要がある。

本校の VLAN によるセグメント分割をまとめると以下のようになる。

- VLAN１を教員用のセグメントとする。
- VLAN２を生徒用のセグメントとする。
- VLAN３を教員用と生徒用の共通セグメントとする。
- VLAN１と VLAN２は通信が行えないため、生徒側から教員側へはアクセスできない。
- VLAN３を VLAN１と VLAN２の両方に所属させることにより、どちらからも生徒用ファイルサーバにアクセスでき、生徒はファイルの保存用に、教員は授業に利用する教材などをサーバに保存することができる。
- 管理棟のポート８がルータに接続しているため、すべての VLAN に所属させる。

これで VLAN による生徒用と教員用ネットワークセグメントが分けられたことになり、セキュリティにおいても確保できる。

本校の例では複雑なネットワーク環境になっているため、VLAN 環境で生徒用と教員用ネットワークセグメントを分ける簡単な例を図１０に示す。

ここでは、生徒用パソコン教室を「VLAN１」、教員用を「VLAN２」、全校用サーバやルータを「VLAN３」に分割する。ただし、「VLAN３」については VLAN１,２の両方に所属させることにより、パソコン教室からも、職員室の教員用パソコンからもインターネットを利用でき、パソコン教室の生徒用のパソコンからは職員室の教員用ネットワークにはアクセスできない環境が構築できる。

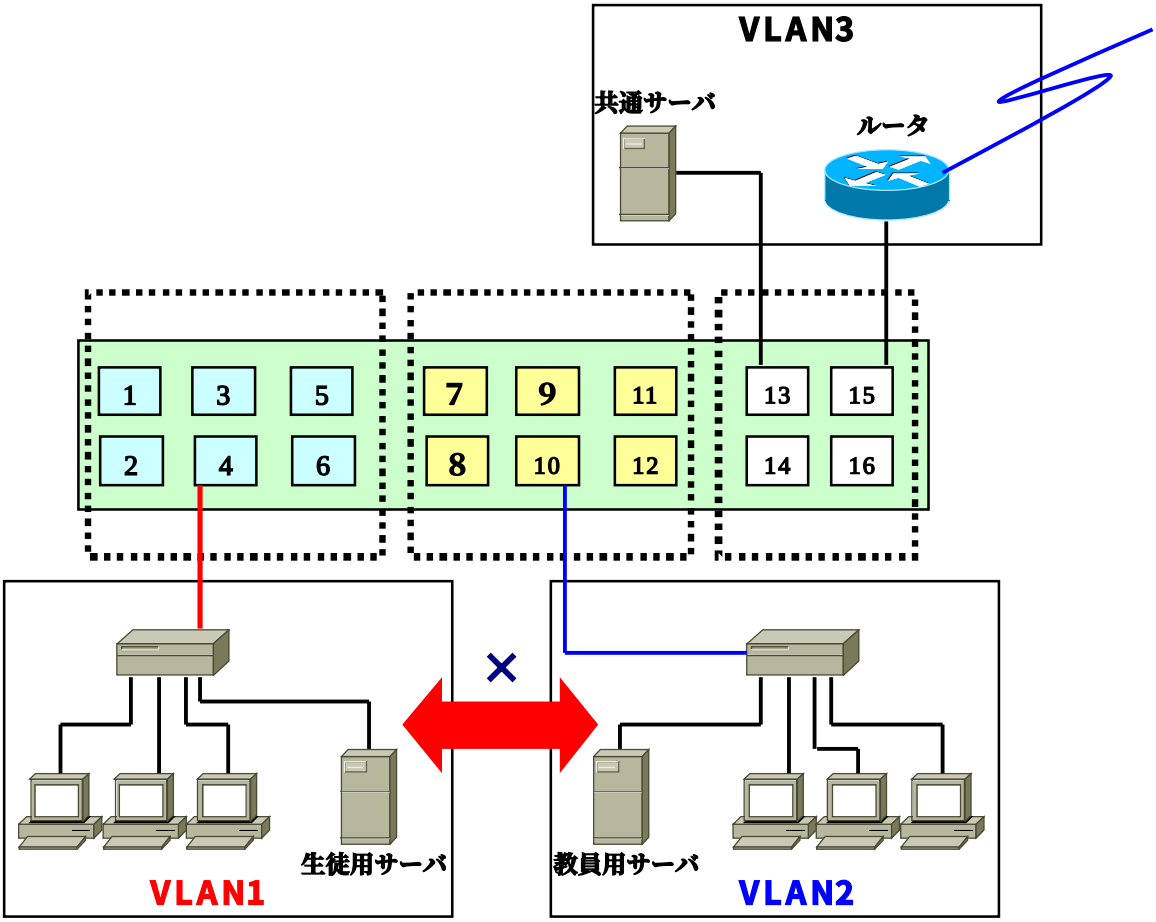


図１０

VLAN で生徒用と教員用ネットワークセグメントを分割する方法以外に、ルータを用いて２つのセグメントを分ける方法がある。その方法の一つはルータにもよるが、プライマリアドレスとセカンダリアドレスといったように、２つのネットワークアドレスを設定できるものがあり、１つの物理的なネットワーク（同一セグメント）を２つの論理的なネットワークに分割することも可能である。ただし、TCP/IP プロトコルのみで、NETBEUI などのプロトコルは対応できない。

もう一つの方法は Cisco 製品のルータになるが、ＡＣＬ（アクセス・コントロール・リスト）を活用する方法がある。これは Cisco 独自の言語である IOS を使ったプログラムのような記述が必要になる。次にその設定方法について述べていく。

5　アクセス・コントロール・リスト（ACL：Access Control List）

アクセス・コントロール・リスト（以下、ACL という）とは、ルータのインタフェースに適用する指示のリストである。それは、どのパケットを許可し、どのパケットを拒否するかをルータに伝える。具体的には、送信元アドレス、宛先アドレス、ポート番号などの仕様に基づいて、パケットを許可または拒否することができる。ルータのインタフェースに ACL を適用すると、それによりトラフィックを管理し、特定のパケットをスキャンすることができる。どのトラフィックもインタフェースを通過するときに、ACL の条件と照合されてテストされる。

ACL は Internet Protocol(IP:インターネット･プロトコル)や IPX(Internetwork Packet Exchange)など、すべてのルーティング対象ネットワーク・プロトコルに対して作成でき、ルータを通過する際にパ

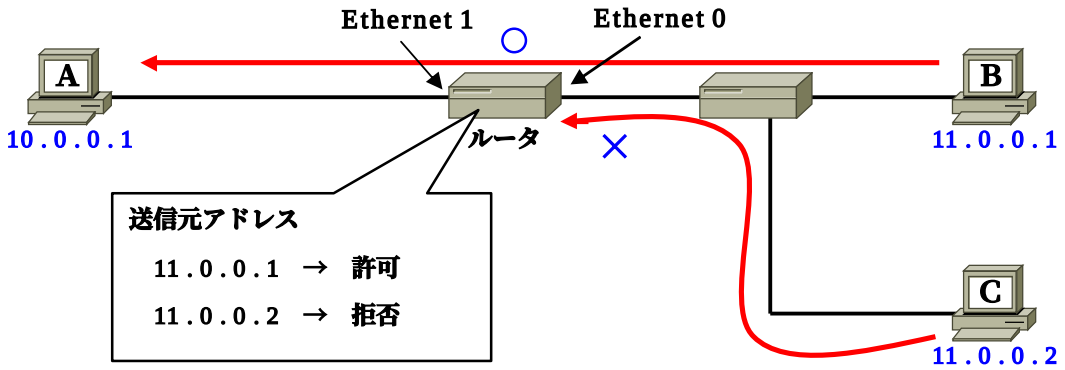
ケットをフィルタリングする。ルータで ACL を設定すると、ネットワークまたはサブネットへのアクセスを制御できる。そこで、学校における校内ＬＡＮではＡＣＬを使って生徒のトラフィックが教員用ネットワークに侵入するのを防止することができる。

ACL はルータのインタフェースでルーティング対象パケットを転送するか、ブロッキングするかを制御することによってネットワーク・トラフィックをフィルタリングする。ルータは ACL に指定された条件に基づいて各パケットを調べ、パケットを転送するか廃棄するかを決める。ACL の条件には、トラフィックの送信元アドレス、トラフィックの宛先アドレス、上位層のプロトコルなどの情報がある。

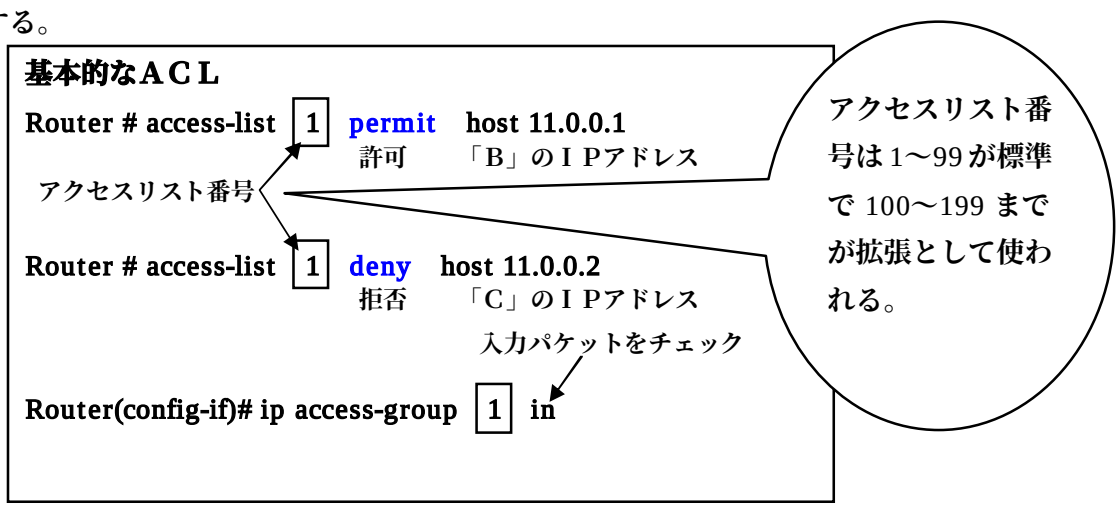
ACL はプロトコル単位で定義しなければならない。つまり、特定のインタフェースのトラフィック・フローを制御する場合は、そのインタフェースで使用できるすべてのプロトコルに対して ACL を 1つずつ定義する。（プロトコルによっては ACL のことをフィルタという場合もある。）たとえば、IP、AppleTalk、IPX 用に設定されたルータ・インタフェースの場合は、少なくとも 3つのＡＣＬを定義する。ACL をネットワークの管理用ツールとして使うと、ルータのインタフェースに到着または送信するパケットのフィルタリングに柔軟性が加わることになる。

(1) 標準アクセス･リスト

標準アクセス･リストは、送信のみの制御を行い、ＩＰアドレスの送信元をチェックする。たとえば、ホストＢからホストＡへのパケットの送信については許可し、ホストＣからの送信は拒否するとする。そうすると、ルータにパケットが入ってくる時にチェックすればよい。



では、どのように上記のような制御を行うかというと、ルータのインタフェース（Ethernet 0）に入力パケットをチェックし、ホストＢのＰＣは通信を許可してホストＣのＰＣは通信を許可しないという例を記述する。



ルータのコンソールに Cisco 専用のケーブルを接続して IOS を起動し、このように入力するとホストBの PC からはホストAへのパケット送信が許可され、ホストCからは拒否される。しかし、このようにホスト数が少なければいいが、ホストの台数が複数台ある場合、ホストすべてを記述するのは面倒である。そういった面倒を解消するためにワイルドカード・マスク･ビットを使い指定するとよい。

ワイルドカード・マスク・ビットとはビットごとに指定し、1 のところをチェックする。例えば、10 . 0 . 0 . 0 のネットワークアドレスはすべて送信を許可しないとすると、0 . 255 . 255 . 255 と指定する。このようにネットワークアドレスごとに許可するかしないかを指定することができる。

標準 ACL 以外にも、拡張 ACL と名前付き ACL がある。ACL は１つのインタフェースの１つのプロトコルに１つしか指定できないことから、上記のような種類がある。標準 ACL は送信元のみの制御で、拡張 ACL は送信先も制御できる。名前付き ACL は、標準 ACL および拡張 ACL に番号ではなく名前をつけることができる。

6　Cisco のスイッチによる VLAN の設定例

＜３個のＶＬＡＮの作成＞

```
Switch_A#vlan database
Switch_A(vlan)#vlan 10 name Accounting
Switch_A(vlan)#vlan 20 name Marketing
Switch_A(vlan)#vlan 30 name Engineering
Switch_A(vlan)#exit
```

＜VLAN 10 へのポートの割り当て＞

```
Switch_A#conf t
Switch_A(config)#int fa0/4
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/5
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/6
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#end
```

＜VLAN 20 へのポートの割り当て＞

```
Switch_A#conf t
Switch_A(config)#int fa0/7
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/8
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/9
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#end
```

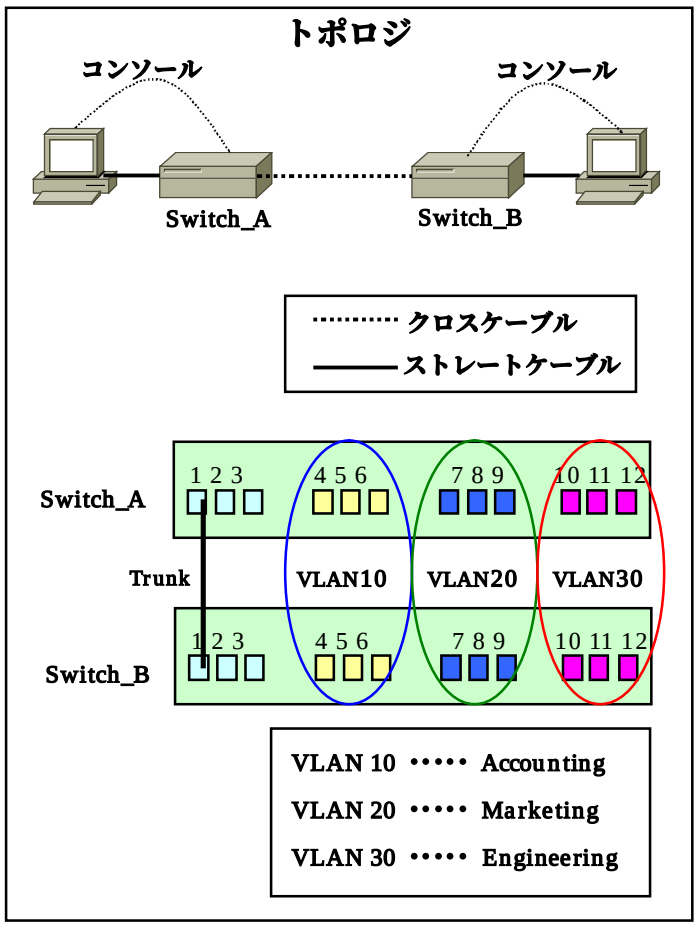
＜VLAN 30 へのポートの割り当て＞

```
Switch_A#conf t
Switch_A(config)#int fa0/10
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#int fa0/11
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#int fa0/12
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#end
```

＜トランクの作成＞

```
Switch_A(config)#int fa0/1
Switch_A(config-if)#switchport mode trunk
Switch_A(config-if)#end

Switch_B(config)#int fa0/1
Switch_B(config-if)#switchport mode trunk
Switch_B(config-if)#end
```



複数のポートを一度に設定する方法

```
Switch_A(config)#int range fa0/10 – 12
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
```

- **VLAN 設定の確認**

  Switch_A#show int fa0/10 switchport

- **VLAN メンバーシップの確認**

  Switch_A#show vlan

- **VLAN の削除**

  Switch_A#vlan database
  Switch_A(vlan)#no vlan 30

- **トランクの確認**

  Switch_A#show int fa0/1 switchport

＜ＶＴＰの設定＞
Switch_A#vlan database
Switch_A(vlan)#vtp server
Switch_A(vlan)#vtp domain group1
Switch_A(vlan)#exit

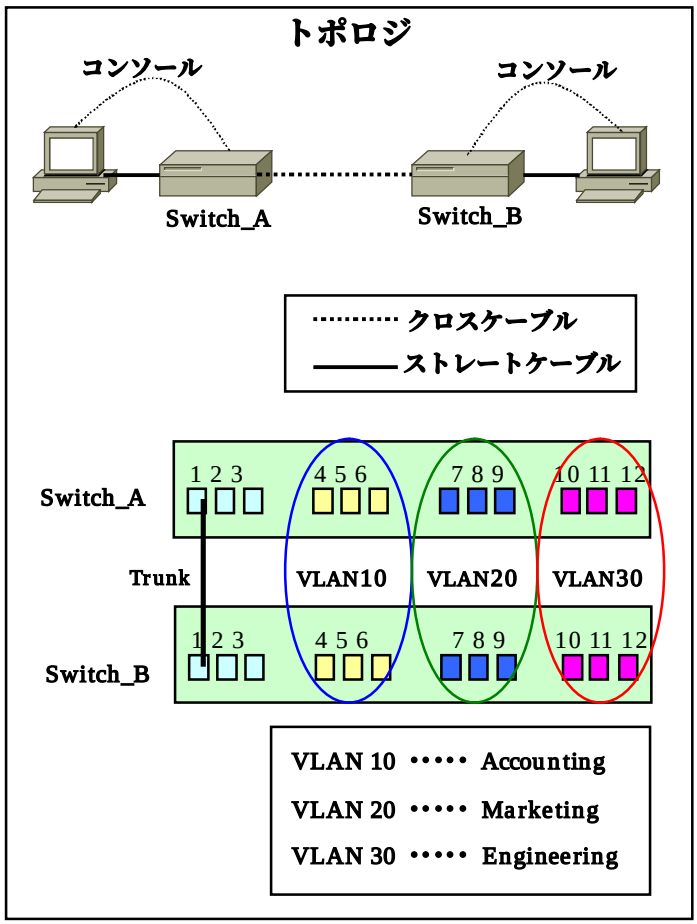
＜ＶＬＡＮの作成＞
Switch_A#vlan database
Switch_A(vlan)#vlan 10 name Accounting
Switch_A(vlan)#vlan 20 name Marketing
Switch_A(vlan)#vlan 30 name Engineering
Switch_A(vlan)#exit

＜VLAN 10 へのポート割り当て＞
Switch_A#conf t
Switch_A(config)#int fa0/4
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/5
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/6
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10

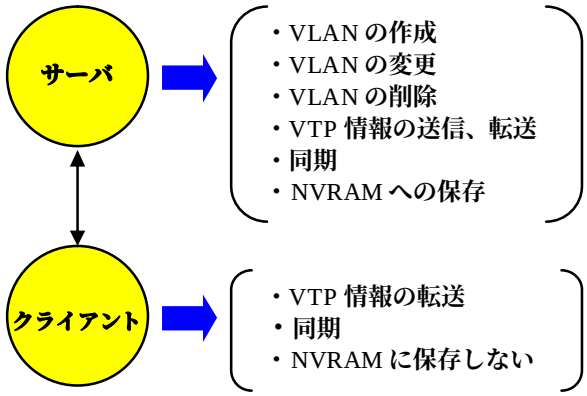
＜VLAN 20 へのポート割り当て＞
Switch_A#conf t
Switch_A(config)#int fa0/7
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/8
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/9
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#end

＜VLAN 30 へのポート割り当て＞
Switch_A#conf t
Switch_A(config)#int fa0/10
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#int fa0/11
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#int fa0/12
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 30
Switch_A(config-if)#end

＜ＶＴＰクライアントの設定＞
Switch_B#vlan database
Switch_B(vlan)#vtp client
Switch_B(vlan)#vtp domain group1
Switch_B(vlan)#exit



• VTP（VLAN Trunking Protocol）
→ VLAN 構成情報を通知して同期させるメッセージプロトコル。



• VTP の通知は５分ごと、あるいは変更があるときに送られる。
• VTP サーバと VTP クライアントは、最新のリヴィジョン番号で同期される。
• VTP 設定の確認
Switch_A#show vtp status

＜トランクの作成＞

```
Switch_A(config)#int fa0/1
Switch_A(config-if)#switchport mode trunk
Switch_A(config-if)#end

Switch_B(config)#int fa0/1
Switch_B(config-if)#switchport mode trunk
Switch_B(config-if)#end
```

＜VLAN 10 へのポート割り当て＞

```
Switch_B#conf t
Switch_B(config)#int fa0/4
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 10
Switch_B(config-if)#int fa0/5
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 10
Switch_B(config-if)#int fa0/6
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 10
Switch_B(config-if)#end
```

＜VLAN 20 へのポート割り当て＞

```
Switch_B#conf t
Switch_B(config)#int fa0/7
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 20
Switch_B(config-if)#int fa0/8
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 20
Switch_B(config-if)#int fa0/9
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 20
Switch_B(config-if)#end
```

＜VLAN 30 へのポート割り当て＞

```
Switch_B#conf t
Switch_B(config)#int fa0/10
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 30
Switch_B(config-if)#int fa0/11
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 30
Switch_B(config-if)#int fa0/12
Switch_B(config-if)#switchport mode access
Switch_B(config-if)#switchport access vlan 30
Switch_B(config-if)#end
```

「VLAN 間ルーティングの設定」

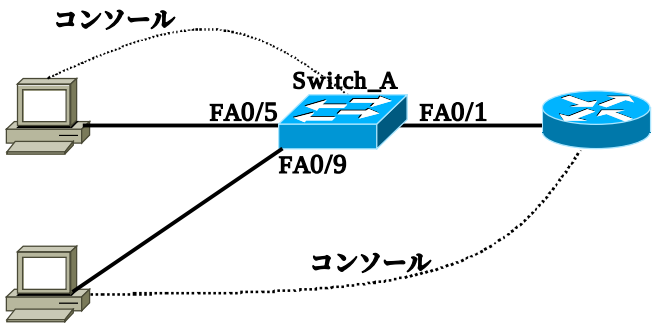
＜ＶＬＡＮの作成および命名＞

```
Switch_A#vlan database
Switch_A(vlan)#vlan 10 name Sales
Switch_A(vlan)#vlan 20 name Support
Switch_A(vlan)#exit
```

＜ＶＴＰプロトコルの設定＞

```
Switch_A#conf t
Switch_A(config)#int fa0/5
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/6
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/7
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#int fa0/8
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 10
Switch_A(config-if)#end
```

トポロジ




```
Switch_A#conf t
Switch_A(config)#int fa0/9
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/10
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/11
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#int fa0/12
Switch_A(config-if)#switchport mode access
Switch_A(config-if)#switchport access vlan 20
Switch_A(config-if)#end
```

＜トランクの作成＞

```
Switch_A#conf t
Switch_A(config)#int fa0/1
Switch_A(config-if)#switchport mode trunk
Switch_A(config-if)#end
```

＜ルータの設定＞

```
router>enable
router#conf t
router(config)#hostname Router_A
Router_A(config)#enable secret cisco
Router_A(config)#line con 0
Router_A(config-line)#password cisco
Router_A(config-line)#login
Router_A(config-line)#line vty 0 4
Router_A(config-line)#password cisco
Router_A(config-line)#login
Router_A(config-line)#exit
Router_A(config)#int fa0
Router_A(config-if)#no shutdown
Router_A(config-if)#int fa0.1
Router_A(config-subif)#encapsulation dot1q 1
Router_A(config-subif)#ip address 192.168.1.1 255.255.255.0
Router_A(config-if)#int fa0.2
Router_A(config-subif)#encapsulation dot1q 10
Router_A(config-subif)#ip address 192.168.5.1 255.255.255.0
Router_A(config-if)#int fa0.3
Router_A(config-subif)#encapsulation dot1q 20
Router_A(config-subif)#ip address 192.168.7.1 255.255.255.0
Router_A(config-subif)#end
```