=== ch01.docx

[chap]1章　パケット解析とネットワークの基礎

コンピュータネットワーク上では、毎日100万もの問題――単なるスパイウェアの感染から複雑なルータの設定エラーまで――が発生しています。そしてすべての問題を迅速に解決するのは不可能です。知識とツールをしっかり用意することが、最高の準備になるはずです。

すべてのネットワークの問題はパケットレベルまで掘り下げることができます。そこでは見た目がかわいいアプリケーションもその醜い実装をさらけ出し、信用できるように見えるプロトコルが悪意あるものでありうることを図らずも証明してしまうのです。ネットワークの問題をより理解し解決するためには、すべてをさらけ出しているパケットを見る必要があります。パケットはアプリケーションにありがちなメニューの見間違いや目を引くグラフィックがなく、信頼できない従業員によってごまかされることもありません。パケットにはなんの秘密もなく（暗号化されたもの以外は）、パケットレベルでできることが増えれば、ますますネットワークを制御し問題を解決することができるようになります。これがパケット解析の世界です。

本書はパケット解析の世界に頭から飛び込んでいます。本書を通じてネットワークの遅延の解決、ボトルネックになっているアプリケーションの特定、実際のシナリオを通してのハッカーの追跡術を学ぶことができるでしょう。本書を読み終えるころには、高度なパケット解析の技術を体得しているはずです。その技術を用いれば、ネットワーク上で起こる多くの困難な問題を解決することができるでしょう。

本章ではネットワーク通信に焦点を当てた基本から始めるので、異なるシナリオに対処する際の基礎を得ることができるでしょう。

パケット解析とパケットスニッファ

パケットスニッフィングやプロトコル解析と呼ばれることもあるパケット解析とは、ネットワーク上で起こっていることをより理解しやすくするために、データをキャプチャして解析することを意味します。パケット解析は通常、パケットスニッファを使って行います。パケットスニッファとは、ケーブルを通っている生のネットワークデータをキャプチャするツールです。

パケット解析は以下のことに役立ちます。

ネットワークの特性の理解

ネットワーク上に誰がいるのかを知る

誰がまたは何が使用可能な帯域を使っているのかを判断

ネットワークの使用がピークになる時間の識別

潜在的な攻撃や悪意ある行為の識別

安全でなく負荷が高いアプリケーションの発見

パケットスニッファプログラムには、フリーと商用どちらもたくさんの種類があります。各プログラムはそれぞれ違う目的のために設計されています。もっとも有名なパケット解析プログラムとして、tcpdump（コマンドラインのプログラム）、OmniPeek、そしてWireshark（本書で使用する） があります。OmniPeek とWireshark はGUI ベースのスニッファです。

パケットスニッファの評価

どのパケットスニッファを使うか決めるには、以下の点を含むいくつかの要素について考える必要があります。

サポートされているプロトコル

パケットスニッファはさまざまなプロトコルを解釈することができます。ほとんどのスニッファは、IPv4やICMP のような一般的なプロトコル、TCPやUDPなどのトランスポート層プロトコル、DNSやHTTPなどのアプリケーション層プロトコルを解釈することができます。しかしあまり一般的でないものや、新しいプロトコル（IPv6、SMBv2、SIPなど）はサポートしていない場合があります。スニッファを選ぶときは、使用する予定のプロトコルがサポートされているかを確認しましょう。

ユーザーフレンドリかどうか

パケットスニッファプログラムのレイアウト、インストールのしやすさ、通常の操作の流れを検討しましょう。選択するプログラムが経験度合いに合っている必要があります。もしパケット解析の経験がほとんどないのであれば、tcpdump のような高度なコマンドラインのパケットスニッファは避けるべきでしょう。逆に経験豊富なら、高度なプログラムのほうがよいかもしれません。パケット解析の経験を積めば、特定のシナリオに合うよう、複数のパケットスニッファプログラムを組み合わせるのが有益な場合もあります。

コスト

パケットスニッファのすごいところは、商用の製品に匹敵するフリーの製品が数多く存在することです。商用製品とフリーの製品のもっとも大きな違いは、レポーティングエンジンです。商用製品には通常、少々豪華なレポート生成モジュールがついていますが、フリー製品にはまず含まれていません。

スニッファのサポート体制

たとえスニッファプログラムの基本をマスターしても、新たな問題を解決するためのサポートが必要になるときもあるでしょう。サポートを評価する際には、開発者のドキュメント、公開されているフォーラムやメーリングリストを探してみてください。Wireshark のようなフリーのパケットスニッファでは開発者のサポートはあまりないかもしれませんが、ユーザーのコミュニティがそれを補っている場合がよくあります。これらのユーザーや貢献者のコミュニティではディスカッションのための掲示板、Wiki やブログを提供しており、パケットスニッファについてより多くのことを知る手助けをしてくれます。

OSのサポート

残念ながら、すべてのパケットスニッファがどんなOS でも使えるわけではありません。サポートが必要なOSすべてで使えるものを選びましょう。コンサルタントであれば、さまざまなOSのパケットをキャプチャし解析する必要があるので、ほとんどのOSで動くツールが必要になります。またあるコンピュータ上でパケットをキャプチャし、別のコンピュータでレビューする場合についても心に留めておかなければなりません。OSが違うと、通信機器ごとに異なるアプリケーションを使わざるをえない可能性があります。

パケットスニッファの仕組み

パケットスニッファの処理には、ソフトウェアとハードウェア間の協力が関わってきます。これは以下の3 つのステップに分けることができます。

収集

最初のステップでは、パケットスニッファはケーブルから生のバイナリデータを収集します。これは通常、スニッフィングするネットワークに接続されているインターフェースをプロミスキャスモードに切り替えることによって実行されます。このモードでは、ネットワークカードは特定のネットワークセグメントに向かうものだけでなく、その上を流れるすべてのネットワークトラフィックを監視することができます。

変換

次のステップでは、キャプチャされたバイナリデータを解読可能な形式に変換します。多くの高度なコマンドラインベースのスニッファは、ここまでしかやりません。このステップではネットワークのデータは非常に基礎的なレベルでの解釈しか行われず、解析のほとんどはエンドユーザーの手にゆだねられます。

解析

第3 のそして最後のステップでは、キャプチャし変換されたデータを実際に解析します。このステップでは、スニッファを用いてネットワークデータをキャプチャし、抜き出したデータを元にプロトコルを特定し、そしてプロトコルの特徴を解析します。

コンピュータはどのように通信するのか

パケット解析を完全に理解するためには、コンピュータ同士がどうやって通信しているのかを理解する必要があります。この節ではOSI 参照モデル、ネットワークのデータフレーム、それらをサポートするハードウェアといったネットワークプロトコルの基礎を勉強します。

プロトコル

現在のネットワークは、各種プラットフォームとシステムの非常に多くの組み合わせで成り立っています。この通信を支援するために、プロトコルと呼ばれる共通の言語を使用しています。一般的なプロトコルとして、TCP、IP、ARP、DHCP があります。プロトコルスタックとは、プロトコルを論理的にグループ化したものです。

プロトコルをよく理解するには、人の話し言葉や書き言葉を定めている規則と同様に考えてみるといいでしょう。すべての言語には動詞をどう活用するか、どのようにあいさつするか、適切に感謝するにはどうするかといった規則があります。プロトコルは言語とほぼ同様に機能し、そのためにパケットをどうルーティングするか、接続をどのように開始するか、また受信したデータをどう認識するかを定義することができます。

プロトコルはその機能によってシンプルにも複雑にもなりえます。さまざまなプロトコルはそれぞれ性質が大きく異なる場合がありますが、多くのプロトコルは一般に以下の機能を持っています。

接続開始

接続を開始するのはクライアントか、それともサーバなのか。通信前に交換されるべき情報とは。

接続特性の交渉

プロトコルの通信は暗号化されているか。通信しているホスト間で送信される暗号鍵とは。

データのフォーマット

パケットに含まれるデータの順番は？それを受け取る通信機器によって処理されるデータの順番は？

エラー検出と訂正

パケットが目的の場所に届くまでに時間がかかりすぎた場合、何が起こるのか。少しの遅れでサーバが通信を確立できない場合、クライアントはどう復帰するのか。

接続終了

あるホストの通信が終了したことを、別のホストはどのように通知するのか。通信をきれいに終了するために送信されるべき最終情報は何か。

OSI参照モデル

プロトコルは、OSI 参照モデルと呼ばれる業界標準の参照モデルを元に、その機能によって分けられています。OSI 参照モデルは、ネットワーク通信のプロセスを、図1-1のように以下の7 つの階層に分けています。この階層モデルのおかげで、ネットワーク通信というものが理解しやすくなっています。最上層のアプリケーション層はネットワークリソースにアクセスするための実際のプログラムを表しています。最下層は物理層で、実際のネットワークデータの転送を行う層です。各層のプロトコルは、その上位層また下位層のプロトコルによってデータが適切に処理されるよう、連携して機能します。

図1-1　OSI参照モデルの7つの層の階層図

アプリケーション層

プレゼンテーション層

セッション層

トランスポート層

ネットワーク層

データリンク層

物理層

注

OSI参照モデルは1983 年にISO（International Organization for Standardization ：国際標準化機構）によって、ISO7498 として公開されました。OSI 参照モデルは業界が推奨しているスタンダード以上の何ものでもありません。プロトコルの開発者は、このモデルに正確に準拠する必要はありません。実際のところ、現存するネットワークモデルはOSI参照モデルだけではありません。たとえば、TCP/IPモデルとしても知られるDoD（Department of Defense ：米国国防総省）モデルを好む人もいます。

OSI 参照モデルの各階層には、次のように特定の機能があります。

アプリケーション層（第7層）

OSI参照 モデルの最上層は、ネットワークリソースにアクセスするための手段をユーザーに提供します。これは通常エンドユーザーが見ることができる唯一の層であり、ネットワーク上のすべての作業の基点となるインターフェースを提供します。

プレゼンテーション層（第6層）

この層は、受信するデータをアプリケーション層が読むことができる形式に変換します。この層でデータをどうエンコードまたはデコードするかは、送受信するデータのアプリケーション層のプロトコルに依存します。プレゼンテーション層ではデータの安全のための暗号化や復号のさまざまな形式も操作します。

セッション層（第5層）

この層は2 台のコンピュータ間の対話やセッションを管理します。またすべての通信機器間の接続の確立、管理、終了を管理します。またセッション層は、通信が全二重なのか半二重なのかを明確にし、通信を唐突に切断するのではなくきれいに終了させるという機能もあります。

トランスポート層（第4層）

トランスポート層の主要な目的として、下位層に信頼できるデータを渡すということがあげられます。フロー制御、データの分割と再構築、誤り制御といった機能のおかげで、トランスポート層は2 点間のデータのやり取りをエラーなしで行えるわけです。信頼性の高いデータ転送を保証することは極めて難しいため、OSI 参照モデルではすべての層が信頼性の保証のために機能します。トランスポート層は接続確立ありのサービスと接続確立なしのサービス両方を提供します。特定のファイアウォールとプロキシサーバはこの層で動作します。

ネットワーク層（第3層）

この層は物理的なネットワーク間でのデータのルーティングを提供している、OSI 参照モデルの中でもっとも複雑な層です。ネットワークホストの論理的なアドレス（たとえばIP アドレス）の管理を担当します。また、パケットの分割、場合によっては誤り検出も行います。ルータはこの層で動作します。

データリンク層（第2層）

この層は物理的なネットワークを通してデータを転送する手段を提供します。主な目的は、物理的な通信機器を特定するのに利用できるアドレス（MACアドレスなど）を提供することです。ブリッジとスイッチはデータリンク層で動作する物理的な通信機器です。

物理層（第1層）

物理層はOSI 参照モデルの最下層であり、ネットワークでのデータ転送における物理的な媒体です。この層では、電圧、ハブ、ネットワークアダプタ、リピータ、ケーブルなど、使用されるすべてのハードウェアの物理的かつ電気的なものを扱います。物理層は接続を確立および終了させ、通信リソースを共有する手段を提供し、信号をデジタルからアナログ、またはその逆に変換します。

表1-1 では、OSI 参照モデルの各層における、もっとも一般的に使用されているプロトコルを一覧にしています。

表1-1　OSI参照モデルの各層で使用される代表的なプロトコル

層

アプリケーション層

プレゼンテーション層

セッション層

トランスポート層

ネットワーク層

データリンク層

プロトコル

HTTP、SMTP、FTP、Telnet

ASCII、MPEG、JPEG、MIDI

NetBIOS、SAP、SDP、NWLink

TCP、UDP、SPX

IP、IPX

Ethernet、Token Ring、FDDI、AppleTalk

OSI参照モデルは推奨されるスタンダードでしかありませんが、完全に頭に入れておくべきです。本書を読み進めるにつれ、異なる層でのプロトコルのやり取りが、ネットワーク問題を形作っているのがわかるでしょう。ルータ問題が「第3層問題」となり、ソフトウェア問題が「第7層問題」として認識されるのです。

注　仕事について話し合っている時、あるユーザーがネットワークリソースにアクセスできないと文句を言っていた話を同僚がしてくれました。これはユーザーが間違ったパスワードを入力したために起きたものでした。同僚はこの問題を「第8層問題だ」と言いました。第8層は非公式なユーザー層です。これはパケットレベルを使用している人々の間でよく使われている用語です。

OSI 参照モデルでは、データはどのように流れているのでしょう？ ネットワーク上を転送される最初のデータは、それを送信するシステムのアプリケーション層から始まります。データはそれぞれの方法でOSI 参照モデルの7つの階層を送信側のシステムの物理層まで下っていき、受信側のシステムに送られます。受信側のシステムは物理層でデータを受信し、データは最上層のアプリケーション層まで受信側のシステムの各層を上がっていきます。

OSI 参照モデルの各層におけるさまざまなプロトコルによって提供されるサービスは、重複することはありません。たとえば、もしある層でプロトコルが特定のサービスを提供すれば、他の層でのプロトコルは同じサービスを提供することはありません。異なるレベルのプロトコルが同様の目標を持った機能を備えていたとしても、その働きは多少異なってきます。

送信側と受信側のコンピュータでは、同一層のプロトコルは一致します。たとえば送信側のコンピュータのレイヤ7 のプロトコルが転送されるデータを暗号化する場合、受信側のコンピュータのレイヤ7 のプロトコルはデータを復号することを求められます。

図1-2 は通信中の2 つのクライアントにおけるOSI 参照モデルの図です。片方のクライアントの最上層から最下層を通り、もう片方のクライアントに到達した後その逆をたどることで通信が成り立ちます。

図1-2　送信側と受信側のシステムにおいて同じ層で機能するプロトコル

アプリケーション層

プレゼンテーション層

セッション層

トランスポート層

ネットワーク層

データリンク層

物理層

アプリケーション層

プレゼンテーション層

セッション層

トランスポート層

ネットワーク層

データリンク層

物理層

OSI 参照モデルの各層は、その直上または直下の層としか通信できません。たとえば、レイヤ2 はレイヤ1 およびレイヤ3 のデータしか送受信できません。

データのカプセル化

OSI参照モデルの異なる層のプロトコルが通信するためには、データをカプセル化する必要があります。これは各レイヤにヘッダやフッタ、つまりレイヤの通信を可能にする追加情報を、通信するデータに追加することです。たとえばトランスポート層がセッション層からデータを受信した場合、トランスポート層は次の層にデータを渡す前にヘッダ情報を追加します。

カプセル化プロセスとは、PDU（Protocol Data Unit ：プロトコルデータユニット）を生成することをいいます。PDU とは、送信されるデータと追加されたヘッダおよびフッタ情報すべてを含みます。データがOSI 参照モデルに従って階層を降りていくとき、PDU はさまざまなプロトコルが追加していくヘッダ情報とフッタ情報によって大きくなっていきます。PDUは物理層に到達すると最終的な形式となり、受信側のコンピュータに送られます。受信側のコンピュータは、データがOSI 参照モデルの階層を上がっていくにつれ、プロトコルのヘッダおよびフッタをPDU から取り除いていきます。PDU がOSI 参照モデルの最上層に到達するときには、もともとのデータしか残っていません。

注　パケットという単語は、OSI 参照モデルのすべての層が追加するヘッダとフッタを含んだ完全なPDUのことを指しています。

カプセル化されたデータがどのように機能するかを理解するのは少々ややこしいので、構築され、送信され、受信されるパケットの実際の例を、OSI参照モデルとの関連で見てみましょう。アナリストは、セッション層やプレゼンテーション層についてはあまり触れることがないので、この例（そして本書の他の部分にも）には登場しないことを心に留めておいてください。

コンピュータで、<http://www.google.com/>をブラウズするところだとします。このプロセスを実行するには、要求パケットを生成し、自分のクライアントコンピュータから、目的のサーバコンピュータへと送信する必要があります。ここではTCP/IP通信セッションはすでに開始されているとします。図1-3はこの例でのデータカプセル化を表したものです。

クライアントコンピュータのアプリケーション層から始めます。ウェブサイトをブラウズすると、HTTPがアプリケーション層のプロトコルを使い、index.htmlファイルをgoogle.comからダウンロードするよう命令を出します。

アプリケーション層プロトコルが命令を指示したら、パケットが目的地に到達するかどうかが問題となります。パケットのデータはトランスポート層のスタックへと渡されます。HTTPはTCPの上にあるアプリケーション層プロトコルです。そのためTCPは、パケットが確実に送信されるようにするトランスポート層プロトコルとして機能します。その結果、TCPヘッダが生成されます。このTCPヘッダにはパケットに添付されるシーケンス番号とその他のデータが含まれており、パケットが適切に送信されるようにします。

図1-3　クライアントとサーバ間のデータカプセル化の図

クライアント　サーバ

HTTP　TCP　HTTP　IP　TCP　イーサネット　IP　TCP　HTTP

アプリケーション層

プレゼンテーション層

セッション層

トランスポート層

ネットワーク層

データリンク層

物理層

HTTP　TCP　HTTP　IP　TCP　HTTP　イーサネット　IP TCP　HTTP

注　あるプロトコルが別のプロトコルの「上にある」という表現をしばしば使うのは、OSI参照モデルがトップダウン設計だからです。HTTPなどのアプリケーションプロトコルは、特定のサービスを提供しますが、そのサービスを確実に提供するにはTCPが必要です。今後学んでいきますが、DNSはUDPの上に、TCPはIPの上にあります。

TCPは仕事を終了すると、パケットの物理アドレスを処理するレイヤ3プロトコルであるIPにパケットを渡します。IPは物理アドレス情報を含むヘッダを作成し、パケットとともにデータリンク層のイーサネットに渡します。物理的なイーサネットアドレスはイーサネットヘッダに保管されています。この段階でパケットは完全に組み立てられて物理層に渡され、ゼロと1の形式でネットワークへと流されます。

完成したパケットはネットワークケーブリングシステム？を流れ、最終的にGoogleウェブサーバへとたどり着きます。ウェブサーバはパケットをボトムアップで読みます。つまりパケットがどのサーバ向けであるかを判断するのに用いられる、ネットワークカード情報を処理する物理的イーサネットを含む、データリンク層から読み始めるのです。この情報が処理されると、レイヤ2情報をはぎ取り、レイヤ3情報が処理されます。

IPアドレス情報もレイヤ2情報と同じように読まれ、アドレスが適切で、パケットが分割されないようにします。このデータもまたはぎ取って、次のレイヤが処理できるようにします。

今度はレイヤ4のTCP情報を読み、パケットが順番に到着しているかどうかを確認します。そしてレイヤ4のヘッダ情報をはぎ取ると、アプリケーション層データのみが残るので、ウェブサイトをホストしているウェブサーバアプリケーションへ渡すことができます。クライアントから送られたこのパケットに応答する際、サーバはTCP ACKパケットを送ってリクエストを受け取ったことをクライアントに知らせ、index.htmlファイルも合わせて送ります。

パケットはすべて、どのプロトコルを使うかに関わらず、この例で説明したように構築され、処理されます。しかしまた、ネットワーク上のすべてのパケットがアプリケーション層プロトコルから生成されるわけではないことを覚えておく必要があります。つまりこの例のパケットは、レイヤ2、3、4のプロトコルの情報しか含んでいません。

ネットワークハードウェア

それではこれからネットワークハードウェアを見ていきましょう。ここではハブ、スイッチ、ルータといった一般的なネットワークハードウェアに焦点を当てましょう。

ハブ

ハブとは通常、図1-4のNetgear のハブのような、RJ-45 のポートを複数持つただの箱にすぎません。ハブは4 ポートという非常に小型なものから、企業向けにラックマウント用に設計された48 ポートの大型のものまであります。

図1-4　典型的な4ポートのイーサネットハブ

ハブは大量の不要なネットワークトラフィックを作りだし、半二重モードでしか動作できないため（データの送受信を同時に行うことができない）、近代的なネットワークや高密度なネットワークでは通常使われているのをみることはないでしょう（代わりにスイッチが使用される）。しかし、2章で説明する「ハブを使用する」テクニックを用いる場合に、ハブはパケット解析に非常に重要となるため、ハブの働きを知っておく必要があります。

ハブはOSI 参照モデルの物理層で動作する、データの中継を行う通信機器です。この通信機器は、すべてのポートに送信されたパケットを伝送（中継）します。たとえば、コンピュータが4 ポートハブのポート1 に接続されていて、ポート2 に接続されているコンピュータにデータを送信する場合、ハブはパケットをポート1、2、3、4 のすべてに送信します。ポート3 とポート4 に接続されているクライアントは、パケットのイーサネットヘッダのMAC（Media Access Control）アドレスで宛先を調べ、パケットが彼らのためのものでないとわかると、ドロップ（破棄）します。図1-5はコンピュータAがコンピュータBにデータを送信する例を示したものです。コンピュータAがデータを送信すると、ハブに接続されているすべてのコンピュータがこのデータを受信します。しかし実際にデータを受信するのはコンピュータBのみで、他のコンピュータはそれを破棄します。

例として、マーケティング部で働いている人のみでなく、その企業の社員全員に、メールの題名に「マーケティング部の皆さまへ」と書いたメールを送信したとしましょう。マーケティング部の社員はメールが自分宛てであることが分かりますから、そのメールを開封します。しかし他の社員はメールが自分宛てでないことが分かれば、それを破棄します。多くの不要な通信と無駄な時間がなぜ発生するかがお分かりでしょう。これがハブの機能です。

プロダクションと高密度なネットワークにおいてハブに替わる最良の機器として、スイッチが挙げられます。スイッチはデータの送受信が同時にできる全二重モードの通信機器です。

図1-5　コンピュータAがコンピュータBにハブを通してデータを送信するときのトラフィックの流れ

コンピュータA 　コンピュータB 　コンピュータC 　コンピュータD

スイッチ

ハブと同じく、スイッチはパケットを中継するよう設計されています。しかしハブのようにすべてのポートにデータを送信するのではなく、通信したいコンピュータにのみデータを送信します。見た目は図1-6に示したように、スイッチはハブとよく似ています。

図1-6　ラックマウント型の24ポートイーサネットスイッチ

Cisco製のものなど、市場にはいくつかの大きなスイッチが出回っており、ベンダ固有のソフトウェアやWeb上のインターフェースを通して管理することができます。これらのスイッチはマネージメントスイッチと呼ばれ、ネットワークを管理する際に便利なさまざまな機能を持っています。特定のポートを有効または無効にしたり、ポートの詳細を表示したり、設定を変更したり、リモートからスイッチを再起動したりできます。

スイッチはパケット送信を操作するための高度な機能も持っています。特定の機器と直接通信できるようにするため、スイッチは通信機器をMACアドレスで管理します。つまり、スイッチはOSI 参照モデルのデータリンク層で動作するということです。

スイッチは、接続されているすべての通信機器のレイヤ2 のアドレスを、トラフィックの見張り番のような働きをするCAM テーブルに記録しています。パケットが送信されると、スイッチはパケット内にあるレイヤ2 のヘッダ情報を読み、CAM テーブルを参照してどのポートにパケットを送信するか決定します。スイッチは特定のポートにしかパケットを送信しないため、ネットワークトラフィックを劇的に減らすことができます。

図1-7はスイッチを通したトラフィックの流れを図で示しています。ネットワーク上で同時に複数通信ができますが、情報はスイッチと目的のコンピュータとの間でのみ直接やり取りされ、他のコンピュータには送られません。

図1-7　スイッチを通してコンピュータAがコンピュータBにデータを送信する際のトラフィックの流れ

コンピュータA 　コンピュータB 　コンピュータC 　コンピュータD

ルータ

ルータはスイッチやハブよりハイレベルな機能を持った高度なネットワーク機器です。ルータはさまざまな形のものがありますが、多くは前面にインジケータランプ（LED）が付いていて背面にポートがあります。ポートの数はネットワークの大きさに依存します。図1-8はルータの一例です。

ルータはOSI 参照モデルのレイヤ3 で動作し、2 つ以上のネットワーク間でパケットを転送します。ネットワーク間のトラフィックの流れを指示することを、ルーティングといいます。異なる種類のパケットを、どうやって他のネットワークに転送するかを決定するプロトコルを、ルーティングプロトコルといいます。ルーティングプロトコルにはいくつかの種類があります。ルータは通常、ネットワーク上の通信機器を識別するために、IP アドレスのようなレイヤ3 のアドレスを使用します。

図1-8　中小規模ネットワークのための小さなCiscoルータ

ルータの概念をイメージする一つの方法は、複数の通りがある近所の家を考えることです。コンピュータを住所を持つ家、ネットワークセグメントを通りだと考えます（図1-9）。ある通りに住んでいるとすれば、その通り沿いのすべての家の間を簡単に行き来することができます。これはスイッチに接続することによって、ネットワークセグメント上のすべてのコンピュータと通信できるということとよく似ています。しかしながら、他の通りの隣人を訪ねるのは、同じセグメント上にないコンピュータと通信するのと同じです。

図1-9　ルーティングと近所の通りとの比較

オークストリート

バインストリート

ドッグウッドレーン

図1-9のバインストリートの503からドッグウッドレーンの202に行かなければならないとします。そのためには、オークストリートを通ってドッグウッドレーンに行かなければいけません。これを、ネットワークセグメントを横断する場合で考えてみてください。192.168.0.3 の通信機器が192.168.0.54 の機器と通信する必要がある場合、10.100.1.1 のネットワークに行くためには、ルータを通らなければいけません。そして通信する機器が存在するネットワークセグメントのルータを通ります。

ネットワーク上のルータの大きさや数は、ネットワークの大きさや機能によって変わります。個人やホームオフィスのネットワークの場合は、ネットワークの中央に置かれたルータのみで構成されているでしょうし、巨大企業のネットワークではいくつものルータがさまざまな部門に置かれ、それらすべては中央の巨大なルータやレイヤ3 スイッチに接続されているでしょう。レイヤ3 スイッチはスイッチが進歩したもので、ルータのような機能がビルトインされています。

ネットワーク構成図にたくさん触れると、さまざまなポイントを通るデータの流れを理解することができるでしょう。図1-10はルーティングのよくある形を示しています。この例では、2 つのネットワークが1 つのルータで接続されています。ネットワークA のコンピュータがネットワークB のコンピュータと通信する場合、送信されるデータは必ずルータを通らなければいけません。

図1-10　コンピュータAがコンピュータXにルータを介してデータを送信したときのトラフィックの流れ

コンピュータA　コンピュータB　コンピュータC　コンピュータD

ルータ

コンピュータW　コンピュータX　コンピュータY　コンピュータZ

トラフィックの分類

ネットワークトラフィックは、ブロードキャスト、マルチキャスト、ユニキャストの3 つに分類することができます。これらはそれぞれ異なる特徴を持っています。それによってネットワーク上のハードウェアがパケットをどのように扱うかが決まります。

ブロードキャスト

ブロードキャストパケットは、ネットワークセグメント上のハブ、スイッチ、ルータのすべてのポートに送信されます。

しかしながら、すべてのブロードキャストトラフィックが同じように生成されるわけではなく、レイヤ2フォームとレイヤ3フォームがあります。たとえばレイヤ2の場合、MACアドレスFF:FF:FF:FF:FF:FFはブロードキャスト専用アドレスであり、このアドレス向けのトラフィックはネットワークセグメント全体にブロードキャストされます。レイヤ3にも専用のブロードキャストアドレスがあります。

IPネットワークレンジにおいて最も可能性の高いIPアドレスが、ブロードキャストアドレスとされます。たとえば192.168.0.xxxのIPレンジ、255.255.255.0サブネットマスクで設定されたネットワークの場合、192.168.0.255がブロードキャストアドレスとなります。

異なる媒体を通して接続されている複数のハブやスイッチからなる巨大なネットワーク上では、1 個のスイッチから送信されたブロードキャストパケットは、スイッチからスイッチへと中継され、ネットワーク上の他のスイッチに到達します。ブロードキャストパケットが到達する範囲をブロードキャストドメインといいます。これはルータを通らずにコンピュータからコンピュータに直接到達できるネットワークセグメントを指します。図1-11 は小さなネットワーク上の2 つのブロードキャストドメインの例を示しています。ルータに到達するまでがブロードキャストドメインなので、ブロードキャストパケットはブロードキャストドメイン内を巡回します。

図1-11　ブロードキャストドメインはルータに到達するまで

ブロードキャストドメイン　ルータ　ブロードキャストドメイン

ルーティングと近隣の家との関係を前に説明しましたが、ブロードキャストドメインの働きについても同じことがいえます。ブロードキャストドメインは近隣の通りだと考えてみてください。もしポーチに立って叫んだら、その通りにいる人たちはそれを聞くことができます。他の通りの人と話したい場合は、ポーチからブロードキャストする（叫ぶ）のではなく、直接その人と話す方法を見つける必要があります。

マルチキャスト

マルチキャストは、1 つの送信元から複数の宛先に同時にパケットを送信する手段です。できる限り小さな帯域を使い、できる限りシンプルにこの手段を実現しています。トラフィックは、宛先に到達するために何回データが複製されたかによって、どう最適化されるか決まります。マルチキャストのトラフィックを正確に操作できるかどうかは、個々のプロトコルの実装に大きく依存しています。

マルチキャストの主な実装方法は、パケットを受信するシステムをマルチキャストグループとしてグループ化し、そのグループにアドレスを割り振ることです。これがIP マルチキャストの働きです。マルチキャストグループにアドレスを割り振ることによって、パケットを受け取るべきでないコンピュータにパケットを送信しないようにします。実際IPは、マルチキャストされるアドレスの範囲全体に対応しています。224.0.0.0から239.255.255.255の範囲にIPアドレスがあれば、マルチキャストトラフィックである可能性が高くなります。

ユニキャスト

ユニキャストパケットはコンピュータからコンピュータへ直接送信されます。ユニキャストがどう機能するかは、使用するプロトコルによって決まります。

たとえば、Webサーバを使って通信したい機器があるとします。これは1対1の接続なので、通信プロセスはクライアントの機器がWebサーバにのみパケットを送信するところから始まります。この通信形式が、ユニキャストトラフィックの一例です。

まとめ

本章でカバーしたのは、パケット解析の基盤として必要な不変の基礎です。ネットワークのトラブルシューティングの前に、ネットワーク通信で何が起こっているかを理解しなければいけません。次の章ではこれらの概念に基づき、ネットワーク通信のより高度な原理について議論していきます。