=== ch07.docx

[chap]7章　一般的な上層プロトコル

この章では、引き続き個々のプロトコルの機能と、これらプロトコルがWiresharkでどのように見えるかについて見ていきます。ここでは最も一般的な上層（レイヤ7）プロトコルである、DHCP、DNS、HTTPの3つについて説明します。

DHCP

ネットワーキングの初期の頃は、ネットワーク上で通信を行いたければ、そのコンピュータに手動でアドレスを割り当てる必要がありました。しかしネットワークが拡大するにつれ、この手動プロセスはたちまち面倒なものへとなっていきます。この問題を解決するために開発されたのが、ネットワークに接続した機器に自動的にアドレスを割り当てるBOOTP（Bootstrap Protocol）です。のちにBOOTPは、より洗練されたDHCP（Dynamic Host Configuration Protocol）によって置き換えられました。

DHCPは、通信機器が自動的にIPアドレス（およびDNSサーバやルータなどの、その他の重要なネットワーク資産のアドレス）を取得できるようにする、アプリケーション層プロトコルです。DHCPサーバのほとんどが、ネットワーク上で使われているデフォルトのゲートウェイやDNSサーバのアドレスなど、他のパラメータもクライアントに提供しています。

DHCPパケット構造

DHCPパケットは、相当量の情報をクライアントに提供します。図7-1に示したように、DHCPパケット内には以下のフィールドがあります。

オペコード　パケットがDHCP要求かDHCP応答かを示します

ハードウェアタイプ　ハードウェアアドレスのタイプ（10MBイーサネット、IEEE 802、ATMなど）

ハードウェア長　ハードウェアアドレスの長さ

ホップ数　DHCPサーバを見つけるのを手助けするリレーエージェントが使用します

トランザクションID　要求と応答を一組にするのに使われるランダムな数字

秒数　クライアントがDHCPサーバにアドレスを要求してからの秒数

フラグ　DHCPクライアントが受け取れるトラフィックのタイプ（ユニキャスト、ブロードキャストなど）

クライアントIPアドレス　クライアントのIPアドレス（ユーザーIPアドレスフィールドから得られたもの）

ユーザーIPアドレス　DHCPサーバに割り当てられたもの（最終的にはクライアントIPアドレスフィールド値となります）

サーバIPアドレス　DHCPサーバのIPアドレス

ゲートウェイIPアドレス　ネットワークのデフォルトゲートウェイのIPアドレス

クライアントハードウェアアドレス　クライアントのMACアドレス

サーバホスト名　サーバのホスト名（オプション）

ブートファイル　DHCPが使うブートファイル（オプション）

オプション　DHCPパケットにさらに機能を持たせるため、その構造を拡張するのに用います

図7-1　DHCPパケット構造

ビットオフセット

オペコード　ハードウェアタイプ　ハードウェア長　ホップ数

トランザクションID

秒数　フラグ

クライアントIPアドレス

ユーザーIPアドレス

サーバIPアドレス

ゲートウェイIPアドレス

クライアントハードウェアアドレス（16バイト）

サーバホスト名（64バイト）

ブートファイル（128バイト）

オプション

DHCP更新プロセス

DHCPの主な目的は、更新プロセスでクライアントにアドレスを割り当てることです。更新プロセスは、dhcp\_nolease\_renewal.pcapファイルに示したように、ひとつのクライアントとDHCPサーバ間で起こります。DHCP更新プロセスは、図7-2のようにDISCOVER、OFFER、REQUEST、ACKNOWLEDGMENTの4種類のDHCPパケットを使うため、DORAプロセスとも呼ばれています。それぞれのDORAパケットを見てみましょう。

図7-2　DHCP DORAプロセス

DHCPクライアント　DHCPサーバ（訳者注　英語のスペルにミスあり。DCHPになっています）

DHCPDISCOVERパケット

見本のキャプチャファイルで分かるように、最初のパケットは0.0.0.0の68番ポートから255.255.255.255のポート67番へと送られます。なぜ0.0.0.0なのかというと、このクライアントにはまだIPアドレスがないからです。パケットが255.255.255.255に送られるのは、このアドレスがブロードキャストアドレスで、パケットがネットワーク上のすべての端末に確実に届くようにしてくれるからです。端末はDHCPサーバのアドレスを知らないので、最初のパケットが送られるのは、待ち受けをしているDHCPサーバを見つけるためです。

パケット詳細のペインを見るとまず、DHCPがトランスポート層プロトコルとして、UDPを使っているのに気づきます。DHCPはクライアントが要求した情報を受信する速度を非常に重視しているからです。またDHCPは信頼性を維持するツールを内蔵しているため、UDPが最適なのです。パケット詳細のペインで最初のパケットのDHCP部分を調べれば、要求プロセスの詳細を見ることができます（図7-3）。

注　WiresharkはDHCPを扱う際、現在もBOOTPを参照しているので、パケット詳細のペインではDHCP部分よりもむしろ、Bootstrapプロトコル部分を見ます。それでも本書では、これをパケットのDHCP部分として参照します。

図7-3　 DHCPDISCOVERパケット

このパケットがリクエストであるのは、[メッセージタイプ]フィールドが(1)であることからわかります①。リクエストパケットのほとんどのフィールドは空白（IPアドレスフィールド②でも分かるように）か、見ればすぐ分かるようになっており、これらはDHCPフィールドのリストに基づいています。このパケットの要点は4つのオプションフィールドにあります。

DHCPメッセージタイプ　これはオプションタイプ「53（t-53）」、長さは1、値は1です③。これらの値がDHCPDISCOVERパケットであることを示しています。

クライアント識別子　クライアントが要求しているIPアドレスについて追加情報を提供します。

Requested IPアドレス　クライアントが割り当てて欲しいIPアドレスを提供します（一般には以前使ったIPアドレス）。

パラメータ要求リスト　クライアントがDHCPサーバから受け取りたい設定項目（他の重要なネットワーク機器のIPアドレス）のリスト。

DHCPOFFERパケット

2番目のパケットのIPヘッダには有効なIPアドレスの一覧が含まれていて、このパケットが192.168.0.1から192.168.0.10へ送られていることを示しています（図7-4）。クライアントにはまだ192.168.0.10のアドレスが割り当てられていないので、サーバはまずARPから提供されたハードウェアアドレスを使って、クライアントとの通信を試みます。通信できなければ、ブロードキャストします。

図7-4　DHCPOFFERパケット

2番目のパケットの「オファーパケット」と呼ばれるDHCP部分は、メッセージタイプが応答であることを示しています①。パケットにはさきほどのパケットと同じトランザクションIDが含まれているので②、この応答が要求に対するものであると分かります。

クライアントにサービスを提供するために、DHCPサーバがDHCPOFFERパケットを送ります。そのためパケットには、パケットについての情報と、クライアントに提供したいアドレス情報が含まれています。図7-4では、ユーザー（クライアント）IPアドレスのフィールドで、IPアドレス192.168.0.10が提供されています③。[Next Server IP Address]フィールド④は、DHCPサーバとデフォルトゲートウェイが同じIPアドレスを共有していることを示しています。

最初のオプションはパケットがDHCPOFFERであることを示しています⑤。その後に続くオプションはサーバによるもので、クライアントのIPアドレスに加え、追加情報が提供されることがわかります。提供されているのは次のような情報です。

・サブネットマスクは255.255.255.0

・更新時間は30分

・リバインディング値？？は52分30秒

・IPアドレスリース期間は1時間

・DHCPサーバ識別子は192.168.0.1

DHCPREQUESTパケット

クライアントはDHCPサーバからの通知を受け取ると、図7-5のようにDHCPREQUESTパケットを送ります。

図7-5　 DHCPREQUESTパケット

まだIPアドレスの取得プロセスが完了していないので、3番目のパケットもIPアドレス0.0.0.0から送ります①。これでパケットは、通信先であるDHCPサーバを知ることができました。

メッセージタイプフィールドは、パケットがリクエストパケットであることを示しています②。キャプチャファイル内のすべてのパケットが同じ更新プロセスに含まれていますが、これは新たな要求／応答トランザクションであるため、新しいトランザクションIDがついています③。IPアドレス情報がすべて空白になっているという点で、DHCPDISCOVERパケットと似ています。

オプションフィールドから④、これがDHCPREQUESTパケットであることがわかります。要求されたIPアドレスが空白でなくなり、DHCPサーバ識別子フィールドにもアドレスが入っています。

DHCPACKパケット

最終段階では、DHCPサーバは要求されたIPアドレスをクライアントにDHCPACKパケットで送り、この情報を図7-6のようにデータベースに記録します。

図7-6　 DHCPACKパケット

これでクライアントにIPアドレスが割り当てられ、通信を始めることが可能になりました。

DHCP In-Lease更新

DHCPサーバが端末にIPアドレスを割り当てることを、クライアントにリース（貸し出し）するといいます。つまりクライアントは期限内だけこのIPアドレスを使用でき、引き続き使用したい場合はリースの更新が必要になります。先述したDORAプロセスは、クライアントが初めてIPアドレスを取得する場合や、リース期限が切れている場合に起こります。いずれもリース期限が切れているとみなされます。

IPアドレスをリース中のクライアントが再起動すると、IPアドレスを再請求するためにDORAプロセスの簡略版を実行しなければなりません。このプロセスをin-lease更新と呼びます。

リース更新では、DHCPDISCOVERYパケットとDHCPOFFERパケットは不要です。リース期限切れ更新でのDORAプロセスと同じなのですが、in-lease更新ではREQUESTとACKパケットのステップしか必要ありません。in-lease更新のサンプルのキャプチャファイルはdhcp\_inlease\_renewal.pcapにあります。

DHCPオプションとメッセージタイプ

DHCPの柔軟性は、オプションにあります。ご覧のとおり、その大きさも内容も多種多様にわたっています。パケット全体の大きさは、使用するオプションの組み合わせによって変わってきます。DHCPオプションの完全なリストについては、<http://www.iana.org/assignments/bootp-dhcp-parameters/>を参照してください。

すべてのDHCPパケットに必須となる唯一のオプションが、メッセージタイプオプション（オプション53）です。このオプションは、パケットに含まれる情報を、DHCPクライアントまたはサーバがどのように処理するかを識別します。表7-1のように、メッセージタイプには8種類あります。

表7-1　DHCPメッセージタイプ

タイプ番号　メッセージタイプ　　説明

DISCOVER　クライアントが使用可能なDHCPサーバを見つけるためのメッセージ

OFFER　クライアントからサーバへ送られるDISCOVERパケットへの応答メッセージ

REQUEST　クライアントからサーバへ送られるパラメータ要求メッセージ

DELINE　クライアントからサーバへ送られるパラメータ無効通知メッセージ

ACK　サーバからクライアントへ送られる設定パラメータ提供メッセージ

NAK　クライアントからサーバへ送られる設定パラメータの取得拒否メッセージ

RELEASE　クライアントからサーバへ送られるリースのキャンセルメッセージ。設定パラメータのリリースによって行われる

INFORM　クライアントがすでにIPアドレスを取得している場合、設定パラメータをクライアントがサーバに要求するメッセージ

DNS

DNS（Domain Name System）は、インターネットプロトコルの中でももっとも重要なもののひとつです。DNSは[www.google.com](http://www.google.com)などのようなドメイン名を、74.125.159.99のようなIPアドレスに変換します。通信したいネットワーク端末のIPアドレスがわからない場合、ドメイン名でアクセスすることができます。

DNSサーバはIPアドレスとドメイン名の変換マップである「リソースレコード」というデータベースを持ち、クライアントや他のDSNサーバと共有しています。

DNSパケット構造

図7-7から分かるように、DNSパケット構造は、先述したパケットタイプとは若干異なります。DNSパケット内のフィールドは次のようになっています。

DNS ID番号　DNSクエリとDNSレスポンスを関連づけています。

QR（Query/Response）　パケットがDNSクエリなのか、レスポンスなのかを表しています。

OpCode　メッセージに含まれる照会のタイプを定義します。

AA（Authoritative Answers）応答パケットでこの値がセットされた場合、ドメインに権威のあるネームサーバからの応答であることを示します。

TC（Truncation）長すぎてパケット内に収まらないため、レスポンスが切りつめられたことを意味します。

RD（Recursion Desired）クエリにこの値がセットされていると、目的のネームサーバに要求した情報がない場合、DNSクライアントが再帰クエリを要求します。

RA（Recursion Available）レスポンスにこの値がセットされていると、ネームサーバが再帰クエリに対応します。

Z（Reserved）すべて0にセットされることが、RFC 1035で定義されています。しかしRCodeフィールドの拡張として用いられる場合もあります。

RCode（Response Code）DNSレスポンスで使われ、エラーの有無を示します。

Question Count　Questionセクションに含まれている質問数。

Answer Count　Answerセクションに含まれている回答数。

Name Server Count　Authorityセクションに含まれているネームサーバリソースレコードの数。

Additional Records Count　Additional Informationセクションに含まれているその他のリソースレコードの数。

Questionセクション　DNSサーバに送られる情報の１つ以上のクエリを含む可変長セクション。

Answersセクション　クエリに回答する1つ以上のリソースレコードを含む可変法セクション。

Authorityセクション　問題解決プロセスを継続するのに利用できる権威あるネームサーバを意味するリソースレコードを含む可変長セクション。

Additional Informationセクション　クエリの回答に必須ではないクエリに関連する追加情報を持つリソースレコードを含む可変長セクション。

図7-7　DNSパケット構造

DNSネームシステム

ビットオフセット

DNS ID番号　QR　オペコード　AA　TC　RD　RA　Z　RCode

Question Count　Answer Count

Name Server Count Additional Records Count

Questionセクション　Answersセクション

Authorityセクション　Additional Informationセクション

シンプルなDNSクエリ

DNSはクエリ／レスポンスの形式で機能します。ドメイン名をIPアドレスに変換したいクライアントはDNSサーバにクエリを送り、サーバはレスポンスとして要求された情報を返します。キャプチャファイルdns\_query\_response.pcapから分かるように、もっとも簡単な場合、このプロセスに必要なのは2個のパケットのみです。

図7-8に示したように、最初のパケットは、クライアント192.168.0.114からサーバ205.152.37.23の53番ポート（DNSの標準ポート）に送られたDNSクエリです。

図7-8　DNSクエリパケット

このパケットのヘッダを調べると、DNSもまたUDPを使っているのがわかります①。

パケットのDNS部分を見ると、最初のほうの小さなフィールドが、ひとつのフラグセクションへとまとめられているのがわかります。このセクションを開くと、メッセージがカットされていない標準的なクエリであり②、再帰が要求されていることがわかります（再帰については後で説明します）。識別できるのは、Queriesセクションを開くと見つかるひとつの質問だけです。ここでは、wireshark.orgについてのホスト（タイプA）インターネット（IN）アドレスのクエリを見ることができます③。つまりこのパケットは「wireshark.orgドメインに関連するIPアドレスは？」と尋ねているわけです。

このリクエストに対するレスポンスが、図7-9に示した2番目のパケットです。このパケットには同一のID番号が付いているので①、クエリに対する正しい回答が含まれていることがわかります。

図7-9　DNSレスポンスパケット

Flagsセクションは、これがレスポンスであり、必要であれば再帰が可能なことを示しています②。このパケットには質問が1つとリソースレコードが1つしか含まれていません③。答えと合わせて最初の質問が入っているからです。Answersセクションを開くと、wireshark.orgのIPアドレスは128.121.50.122だという回答が見つかります④。クライアントはこの情報をもとに、IPパケットを生成し、wireshark.orgとの通信を開始することができます。

DNS Questionタイプ

DNSクエリとレスポンスで使われるタイプのフィールドは、そのクエリまたはレスポンスのリソースレコードタイプを示しています。よく使われるメッセージ／リソースレコードのタイプの一部を表7-2にまとめました。通常のトラフィックおよび本書でこれらのタイプを目にするでしょう。

表7-2　一般的なDNSリソースレコードタイプ

値　タイプ　説明

IPv4ホストのアドレス

そのドメインが管理するネームサーバ

ホストの別名（エイリアス）

そのドメインのメールサーバ

テキスト文字列

Ipv6ホストのアドレス

増分ゾーン転送

フルゾーン転送

表7-2は完全なリストではありません。すべてのDNSリソースレコードタイプについては、[http://www.iana.org/assignments/dns -parameters/](http://www.iana.org/assignments/dns%20-parameters/)を参照してください。

DNS再帰

DNSの階層的な構造のため、クライアントから送られたクエリに回答するには、DNSサーバはお互いに通信する必要があります。内部向けDNSサーバは、ローカルのイントラネットサーバのドメイン名とIPアドレス対応マッピングを持っていますが、GoogleやDellに関連するIPアドレスは検索できません。

DNSサーバがIPアドレスを検索する場合、そのリクエストをしたクライアントの代理として別のDNSサーバに問い合わせます。DNSサーバがクライアントのように行動するこのプロセスを「再帰」と呼びます。

DNSクライアントとDSNサーバの両方の視点から再帰プロセスを見るために、dns\_recursivequery\_client.pcapファイルを開きましょう。このファイルには、クライアントのDNSトラフィックファイルをキャプチャした2つのパケットが含まれています。最初のパケットは、図7-10に示したように、DNSクライアント172.16.0.8から、DNSサーバ172.16.0.102へと送られた最初のクエリです。

このパケットのDNS部分を開くと、これが[www.nostarch.comというDNS](http://www.nostarch.comというDNS)名①のAタイプレコードの、標準クエリであることがわかります。Flagsセクションを開けばこのパケットについての詳細がわかり、再帰が要求されていることも分かります②。

2番目のパケットは、予想通り最初のクエリに応答するものです（図7-11）。

このパケットのトランザクションIDはクエリのIDと一致しており①、エラーもないので、[www.nostarch.com](http://www.nostarch.com)関連のAタイプのリソースレコードを受け取ることになります②。

図7-10　再帰要求ビットがセットされたDNSクエリ

図7-11　DNSクエリレスポンス

このクエリが再帰によって回答されるのを見る唯一の方法は、dns\_recursivequery\_server.pcapファイルから分かるように、再帰が行われているときにDNSサーバのトラフィックを待ち受けすることです。このファイルは、クエリが作成されたときのローカルDNSサーバ上のトラフィックをキャプチャしたものを示しています。最初のパケットは先ほどキャプチャしたファイルで見たクエリと同一です。DNSサーバはクエリを受けとり、ローカルデータベースと照会しましたが、DNS名（nostarch.com）と一致するIPアドレスは何かという質問の答えが見つけられませんでした。パケットには再帰要求ビットがセットされているので、DNSサーバが答えを見つけるためにほかのDNSサーバに問い合わせているのが、2番目のパケットから分かります。

2番目のパケットでは、172.16.0.102のDNSサーバは4.2.2.1に新たなクエリを送っていますが、このサーバは図7-12のように、要求を転送するよう設定されています。このクエリはオリジナルのミラーであり、DNSサーバをクライアントへと変えます。

図7-12　再帰クエリ

先ほどのキャプチャファイルとはトランザクションIDが異なるため、これが新たなクエリであることがわかります。サーバ4.2.2.1がこのパケットを受け取ると、ローカルDNSサーバが図7-13のように応答を受けとります。

図7-13　再帰クエリへのレスポンス

応答を受け取ると、ローカルDNSサーバは4個目の最後のパケットを、要求された情報とともにDNSクライアントに転送します。

この例では再帰の1回分しか示していませんが、1回のDNS要求で再帰が何度も行われることがあります。ここではDNSサーバ4.2.2.1から回答を受け取りましたが、このサーバが答えを見つけるために、ほかのサーバに繰り返しクエリを転送する場合があります。ひとつの簡単なクエリが最終的に正しい答えを見つけるまでに、世界中を巡ることがあるのです。図7-14は再帰クエリのプロセスを図式化しています。

図7-14　再帰クエリ

DNSクライアント　ローカルDNSサーバ　外部DNSサーバ

再帰クエリ　クエリレスポンス　再帰クエリ　クエリレスポンス

DNSゾーン転送

DNSゾーンとは、DNSサーバが管理を任された名前空間（またはDNS名のグループ）です。仮に「エマのダイナー」が、emmasdiner.comを管理するDNSサーバを持っているとしましょう。この場合、emmasdiner.comのIPアドレスを照会しようとする内部および外部の通信機器は、そのゾーンに権限を持つDNSサーバにコンタクトする必要があります。エマのダイナーが大きくなり、DNS名前空間のメールを処理する目的のみのために（仮にmail.emmasdiner.comとしておきます）、2番目のDNSサーバを追加した場合、そのサーバはメールのサブドメインに権限を持つことになります。図7-15のように、サブドメインがもっと必要になれば、さらにDNSサーバを追加することができます。

ゾーン転送は、2つの機器の間でゾーンデータの転送が行われる場合に用いられます。たとえば複数のDNSサーバを持つ組織では、管理部門は通常、プライマリのDNSサーバがダウンした場合に備え、そのDNSゾーン情報のコピーを保持するセカンダリのDNSサーバを設定しています。ゾーン転送には2種類あります。

フルゾーン転送（AXFR）

2つの機器間でゾーン全体を転送する。

増分ゾーン転送（IXFR）

ゾーン情報の一部のみを転送する。

ファイルdns\_axfr.pcapには、2つのホスト172.16.16.164と172.16.16.139間のフルゾーン転送の例が含まれています。

このファイルを始めて見ると、UDPパケットではなくTCPパケットが入っているので、正しいファイルを開いたかどうか悩むかも知れません。DNSはUDPを使っていますが、ゾーン転送などの一部のタスクについてはTCPを使います。大量のデータを転送する場合、TCPのほうが信頼度が高いからです。このキャプチャファイルの最初の3つのパケットは、TCPの3ウェイハンドシェイクです。

図7-15　DNSゾーンは名前空間によって責任分担する

4個目のパケットは、172.16.16.164と172.16.16.139間のゾーン転送要求で始まっています。このパケットにはDNS情報は含まれていません。「再構築PDUのTCPセグメント」というしるしがついているのは、ゾーン転送要求パケットで送られたデータが、複数のパケットで送られたためです。4番目と6番目のパケットにパケットのデータが含まれています。5ばんめのパケットは4番目のパケットを受け取ったというACKパケットです。パケットがこのように分かりやすい形で表示されるのも、Wiresharkの機能のひとつです。図7-16に示したように、ここでは6番目のパケットを完全なDNSゾーン転送要求として参照できるようにします。

ゾーン転送要求は標準的なクエリですが①、ひとつのレコードタイプを要求する代わりにAXFRタイプを要求します②。つまりサーバからDNSゾーン全体を受け取ろうとするわけです。サーバは図7-17のように、ゾーンレコードの含まれた7番目のパケットで応答します。ゾーン転送ではかなりの量のデータがやり取りされますが、これはシンプルな例のひとつです。ゾーン転送が完了すると、TCPティアダウンプロセスで終了します。

図7-16　DNSフルゾーン転送要求

図7-17　DNSフルゾーン転送の実行

警告　ゾーン転送に含まれるデータが間違った人間の手に渡ると非常に危険です。DNSサーバを列挙するだけで、ネットワーク全体のインフラがマッピングできてしまうからです。

HTTP

HTTP（Hypertext Transfer Protocol）は、World Wide Webの転送メカニズムであり、WebブラウザがWebサーバに接続してWebページを閲覧できるようにします。多くの組織では、HTTPのトラフィック量がもっとも多くなっています。Googleで検索したり、Twitterでツイートしたり、あるいはケンタッキー大学のバスケット試合の得点をESPN.comでチェックしたりする度に、HTTPを使っているのです。

HTTP転送のパケット構造は見ません。これらのパケットの中身は目的によってかなりのばらつきがあるため、そのチェックは読者にお任せします。ここではHTTPの実用的なアプリケーションを見ていきましょう。

HTTPでブラウズする

HTTPはWebブラウザを使ってWebページをブラウズするのにもっともよく利用されます。キャプチャファイルhttp\_google.pcapは、TCPをトランスポート層プトロコルとして利用するHTTP転送などを含んでいます。通信はクライアント172.16.16.128と、GoogleのWebサーバ74.125.95.104との間の3ウェイハンドシェイクで始まります。

通信が確立されると、HTTPというしるしのついた最初のパケットが、図7-18のようにクライアントからサーバへと送られます。

図7-18　最初のHTTP GETリクエストパケット

HTTPパケットはTCPを通じて、HTTP通信の標準ポートである80番ポートへ送られます①（8080番もよく使われます）。

HTTPパケットは8つの異なるメソッドのどれであるかで識別され（HTTP仕様書1.1で定義されています）、それによってパケットの送り手のアクションが決まります。図7-18では、パケットのメソッドはGETで、そのURIリクエストは/で示されており、リクエストバージョンはHTTP/1.1となっています②。この情報から、クライアントはHTTPのバージョン1.1を使っているWebサーバのルートWebディレクトリ（/）のダウンロード（GET）をリクエストしていることが分かります。

次に、ホストは自分の情報をWebサーバに送ります。情報には使っているユーザーエージェント（ブラウザ）、使用候補言語（Accept-Language）、cookie情報などが含まれます。サーバはこの情報を利用して、クライアントにどのデータを戻すかを判断します。

サーバが4番目のパケットでHTTP GETリクエストを受け取ると、TCP ACKで応答し、6番目から11番目のパケットを使って要求されたデータの転送を開始します。HTTPはクライアントとサーバ間のアプリケーション層コマンドの発行のみに使用されます。データ転送の段階になると、データストリームの最初と最後以外は、アプリケーション層コントロールは見えなくなります。

図7-19のように、サーバからのデータが6番目と7番目のパケットで送られ、クライアントからの応答は8番目のパケット8、さらに9番目と10番目のパケットとしてデータが転送されて、もうひとつの確認が11番目のパケットとして送られます。データ転送はHTTPが行っていますが、WiresharkではこれらのパケットはすべてHTTPパケットではなく、TCPとして表示されます。

図7-19　クライアントブラウザとWebサーバ間のTCPデータ転送

データが転送されると、再構築されたデータストリームが図7-20のように送信されます。

図7-20　応答コード200の最終的なHTTPパケット

HTTPはあらかじめ定義された応答コードの番号を用いてリクエストメソッドの結果を示します。この例では応答コードが200なので①、リクエストメソッドが成功したことを意味しています。またパケットには、タイムスタンプと、コンテンツの暗号化やWebサーバのパラメータ設定に関する追加情報も含まれています。クライアントがこのパケットを受け取ると、トランザクションが完了します。

HTTPでデータをPOSTする

Webサーバからデータをダウンロードするプロセスを見てきたので、今度はアップロードするほうを見てみましょう。ファイルhttp\_post.pcapには、ユーザーがWebサイトにコメントを投稿するという、非常にシンプルなアップロードの例が含まれています。最初の3ウェイハンドシェイクのあと、クライアント（172.16.16.128）は、図7-21のように、Webサーバ（69.163.176.56）にHTTPパケットを送ります。

図7-21　HTTP POSTパケット

パケットはPOSTメソッドを使って①Webサーバにデータをアップロードします。ここで使われているPOSTメソッドはURI/wp-comments-post.php②と、HTTP1.1 Request versionを指定しています。ポストされたデータの内容を見るには、パケットのLine-based Text Dataの部分を開きます③。

POSTでデータが転送されると、ACKパケットが送られます。図7-22のように、サーバは応答コード302①、つまり「見つかりました」という応答の6番目のパケットを戻します。

図7-22　HTTP応答コード302はリダイレクトに使われる

応答コード302は、HTTPでは通常リダイレクトを示します。[Location]フィールドが、クライアントのリダイレクト先を示しています②。ここではそのリダイレクト先が、コメントが投稿された最初のWebページとなっています。最後にサーバがステータスコード200を送り、ページのコンテンツが続くいくつかのパケットを使って転送されて、完了します。

まとめ

この章では、アプリケーション層のトラフィックを調べる際に目につく、もっとも一般的なプロトコルを紹介しました。この後に続く章では、実際のさまざまなシナリオを通じて、新しいプロトコルについて学ぶとともに、本章で説明したプロトコルのそのほかの機能について説明していきます。

個々のプロトコルをさらに詳しく知りたければ、関連するRFCを読むか、または「TCP/IP Guide」（Charles Kozeriok著、No Starch Press　2005年）を参照してください。また付録のリストも参考になります。