ネットワーク講習

1　はじめに

1.1　ネットワークとは

「ネットワーク」とは，なにかとなにかが網の目のようになにかによって繋がっている状態のことであり，なにかを運ぶためのものである．単にネットワークというと，物流・道路・神経などいろいろなものがあるが，特に「コンピュータネットワーク」においては，コンピュータとコンピュータが網の目のようにケーブルなどの通信媒体によって繋がっており，データを運ぶためのものであると定義できる．

1.2　ネットワークの利点

コンピュータネットワークを用いると，メールやファイルのやり取りが行えるだけではなく，他のコンピュータに繋がっているプリンタを使用したり，他のコンピュータにデータを処理させることもできる．

メールやファイル，印刷したいデータなど，コンピュータやユーザが持つものを「リソース」と呼ぶ．コンピュータネットワークを利用する最大の利点は，こうしたリソースを複数のコンピュータで共有することであるといえる．

1.3　プロトコル

人と人とが会話をするとき，お互いが異なる言語を用いてしまうと正しく意思疎通を行うことができない．つまり，人間同士の会話が成立するためには「お互いが理解できる言語を用いる」という暗黙的なルールが存在していなければならない．

同様に，コンピュータの世界においても，複数のコンピュータ同士がネットワークを介して通信を行うときには共通のルールが必要となる．柔軟な対応ができる人間とは異なり，コンピュータは機械であるため，通信を行うためには事前にありとあらゆるルールを決めておく必要がある．

例えば，メールを送信するときのルールやファイルのやり取りを行うときのルール，通信相手を特定するときのルール，通信経路を選択するときのルール，通信途中でデータが壊れてしまったときのルール，更には，通信を行う際に用いるケーブルの種類や電気信号の変換方法など，様々な取り決めやルールが存在している．これらのルールのことを「プロトコル」と呼び，使用するプロトコルが異なれば正しく通信を行うことができない．

2 OSI 参照モデル

2.1 OSI 参照モデル

ネットワーク機器や端末を開発するメーカーがそれぞれ好き勝手なプロトコルを用いてしまうと，メーカーの異なる機器同士で通信が行えなくなってしまうという問題が起こる．そこで，世界的にプロトコルの標準化が進められ,「OSI 参照モデル」というプロトコルの設計モデルが 1984 年に国際標準化機構（ISO）によって制定された．

現在では，プロトコルの設計モデルとして「TCP/IP モデル」が一般的に普及している．TCP/IP モデルは必ずしも OSI 参照モデルに準拠している訳ではないが，データ通信における流れとしては OSI 参照モデルと類似している部分が多い．そのため，OSI 参照モデルはネットワーク通信を行う際の基本的な考え方であるとして今でも広く浸透している．今回の講習でも，分かりやすさのために OSI 参照モデルを用いて説明を行う．

2.2 階層構造

一度のデータ通信を行うとき，必要となるプロトコルは一つだけではなく，複数のプロトコルが使用されている．OSI 参照モデルでは，それぞれのプロトコルの役割に応じてデータ通信を七つの段階に分類している．各段階のことを「レイヤ」と呼び，ネットワークによるデータ通信は各レイヤごとの複数のプロトコルで実現される．OSI 参照モデルにおける各レイヤの名称と役割を表 1 に示す．

表 1: OSI 階層モデルの各レイヤと役割

レイヤ	名称	役割
レイヤ 7	アプリケーション層	アプリケーションソフト間での通信を規定
レイヤ 6	プレゼンテーション層	データ形式に関する規定
レイヤ 5	セッション層	通信の開始／終了に関する規定
レイヤ 4	トランスポート層	通信の品質を確保するための通信手順を規定
レイヤ 3	ネットワーク層	異なるネットワーク間の通信を規定
レイヤ 2	データリンク層	同じネットワーク内の通信を規定
レイヤ 1	物理層	接続のための物理的な規定

各レイヤは独立しており，レイヤのプロトコルの変更は他のレイヤに影響を及ぼさない．また，基本的には下位のレイヤは上位のレイヤを考慮して設計されており，上位のレイヤは下位のレイヤを意識する必要は無い．

2.3 カプセル化

誰かに宅配便を送る場合，送る側がまず贈り物を梱包材に包み，ダンボールに入れて，宛先や送り主を書いた配達表を貼ることで，配達員に宛先まで運んでもらう．宅配便を受け取った側は，配達表を剥がし，ダンボールから取り出して，梱包を解くという，送る側と逆の手順を踏むことで贈り物を正しく受け取ることができる．

同様に，コンピュータがデータ通信を行う場合も，送りたいデータに対して制御情報（ヘッダ）を付け加える作業が必要となる．制御情報とは，送信先・送信元のアドレスや，それぞれのプロトコルで必要となる情報などである．

OSI 参照モデルでは，送信側が各レイヤにおいて 7 から 1 の順番でそれぞれの手順をこなし，送りたいデータに制御情報を付け加えていく．このように，各レイヤにおいてデータに制御情報を付け加えていく作業を「カプセル化」と呼ぶ．受信側は，各レイヤにおいて 1 から 7 の順番でそれぞれの手順をこなし，送信側とは逆の順序で制御情報を取り除いていくことで，最終的に目的

のデータを受け取ることができる．OSI 参照モデルにおけるカプセル化を用いたデータ通信の流れを図 1 に示す．

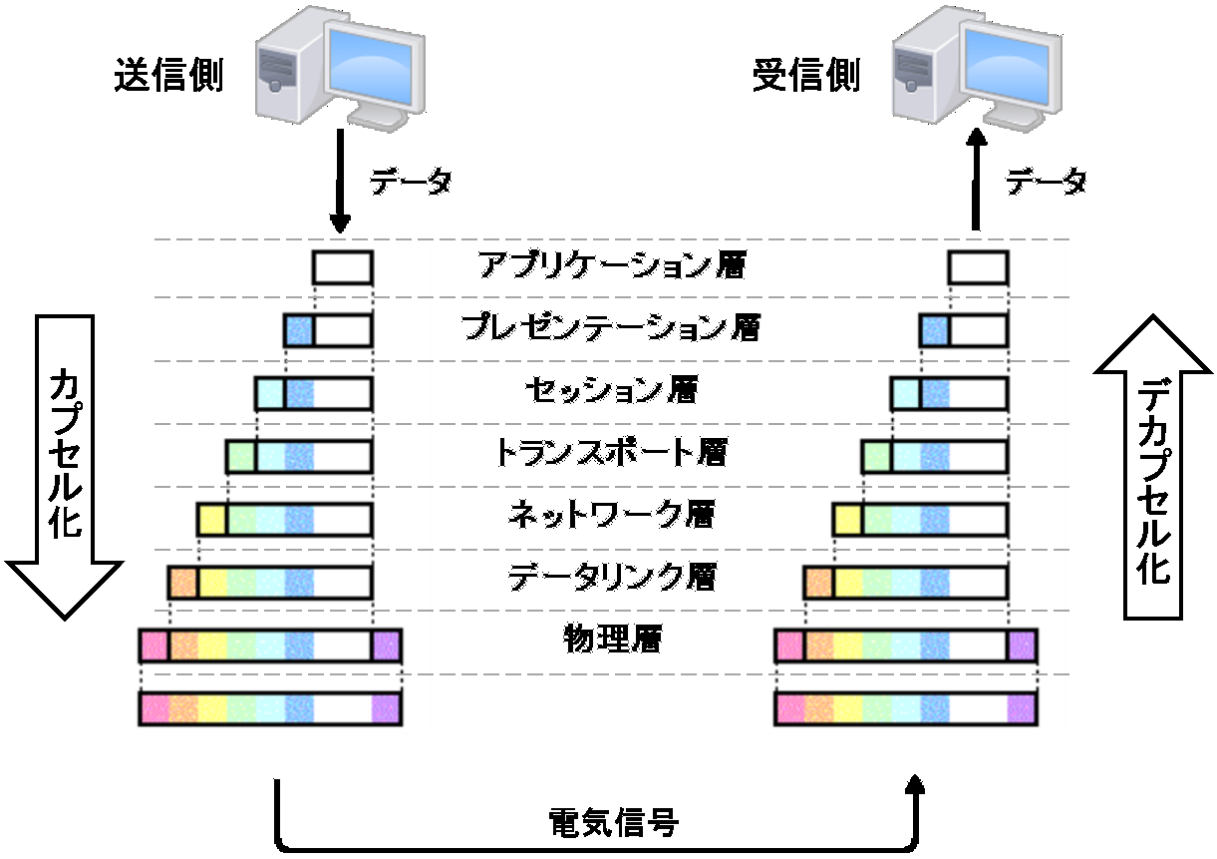

図 1: OSI 参照モデルによるカプセル化の流れ

2.4 TCP/IP モデル

現在インターネットで広く用いられているプロトコル群は IETF が制定した TCP/IP プロトコル群であり，これらは「TCP/IP モデル」がベースとなっている．OSI 参照モデルが七つのレイヤで構成されていたのに対し，TCP/IP モデルは四つのレイヤで構成される．OSI 参照モデルと TCP/IP モデルの階層の対応を図 2 に，また，TCP/IP で用いられるプロトコルの例を表 2 に示す．

表 2: TCP/IP モデルのプロトコル例

レイヤ	名称	プロトコルの例
レイヤ 4	アプリケーション層	HTTP，FTP，SMTP
レイヤ 3	トランスポート層	TCP，UDP
レイヤ 2	インターネット層	IP，ARP
レイヤ 1	インターフェース層	Ethernet，PPP

TCP/IP モデルのレイヤ 4 は，OSI 参照モデルでいうところのレイヤ 5，6，7 の部分に対応し，また，TCP/IP モデルのレイヤ 1 は，OSI 参照モデルでいうところのレイヤ 1，2 の部分に対応している．注意してほしいのは，実際には TCP/IP モデルと OSI 参照モデルはまったく別のプロトコル設計モデルだということである．例えば．TCP/IP モデルのインターネット層と OSI 参照モデルのネットワーク層は，役割は似ているが同一のものではない．

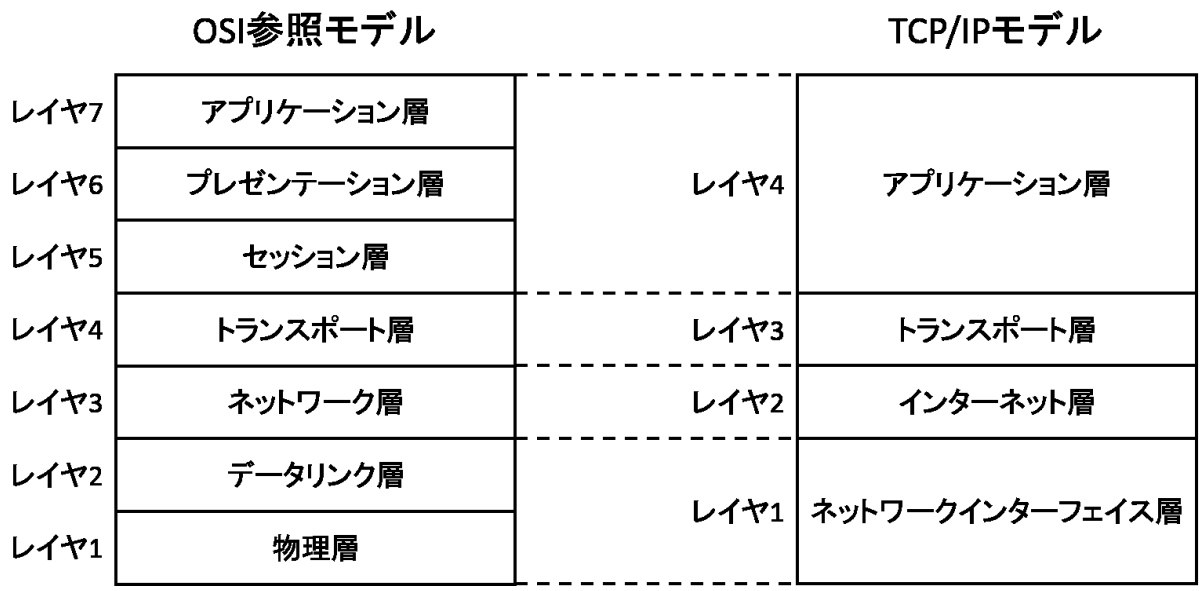

図 2: OSI 参照モデルの 7 階層と TCP/IP モデルの 4 階層

以降では，基本的に TCP/IP プロトコル群を利用したネットワークについて説明する．

3 Internet Protocol

IP（Internet Protocol）は TCP/IP モデルにおいてデータの送受信を担うプロトコルである．OSI 参照モデルにおけるレイヤ 3（ネットワーク層）に相当する役割を持ち，IP アドレスと呼ばれる IP 独自の論理アドレスに基づいてデータをやり取りすることを特徴としている．IP は異なるネットワークに属する通信相手までデータを届ける機能を持ち，物理的に遠く離れた相手との通信経路を確立する．TCP/IP モデルの中核となる重要なプロトコルである．

3.1 IP アドレスとは

IP アドレスは IP での通信に利用されるアドレスである．機器を示す情報（誰）だけでなく，属するネットワークの情報（何処）も含んでいるという特徴があり，異なるネットワークへデータを転送するときに経路の探索が容易にできるような仕組みになっている．郵便配達における住所のようなものであるといえる．

レイヤ 2（データリンク層）の Ethernet では，機器ごとに割り振られた MAC アドレス（物理アドレス）を利用して通信を行っていたが，MAC アドレスは機器を示す情報（誰）しか含んでおらず，MAC アドレスからその機器が属するネットワーク（何処）を知ることはできなかった．IP であれば異なるネットワークに存在する機器であっても，その機器が属するネットワークを割り出し通信経路を探し出すことができる．

IP アドレスには IPv4 アドレスと IPv6 アドレスという二つの形式があるが，以下の項では現在の主流である IPv4 アドレスについて説明する．

3.2 IP アドレスの構造

IP アドレスは 32 桁の 2 進数（32 ビット）の番号として表される．しかし，そのままでは人間が読み取ることは困難であるため，通常はビット列を 1 バイト（8 ビット）ごとに区切り，10 進数を用いて表記する（表 3）．1 バイトの区切りのことをオクテットと呼び，32 ビットの IP アドレスは 4 オクテット構成となる．

表 3: IP アドレスの表記

2 進数	10011110.11011001.01001101.11100001
10 進数	158.217.77.225

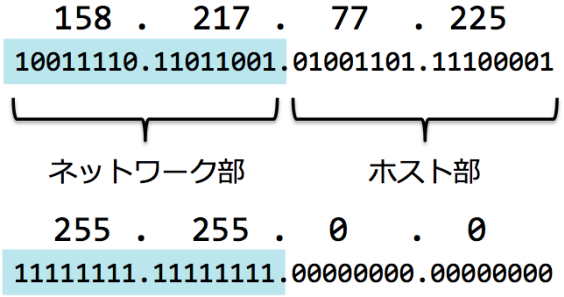

図 3: ネットワーク部とホスト部

IP アドレスはホストが属するネットワークを示す『ネットワークアドレス』と，機器を特定する『ホストアドレス』から構成される階層型アドレスである．図 3 の例では，「158.217.77.225 は，158.217.0.0 ネットワークにある 77.225 番の機器である．」と解釈することができる．何処（158.217.0.0 ネットワーク）の誰（77.225 番）という情報があれば，相手が自分と同じネットワー

クにいるのか，あるいは遠く離れたネットワークに居るのか，などを見分けることができるため，通信経路の探索が容易になる．

ネットワークアドレスは全ネットワーク内でユニークなものである必要があり，インターネットの場合では NIC（Network Information Center）等の組織が企業やプロバイダに対してネットワークアドレスを割り振っている．一方，ホストアドレスは各ネットワークの管理者が任意の機器に割り振ることができ，そのネットワーク内でユニークなものでさえあれば良い．

3.3 クラスフルアドレッシング

先程の例（158.217.77.225）では，IP アドレスの左から 16 ビットをネットワークアドレスを示す『ネットワーク部』，残りのビットをホストアドレスを示す『ホスト部』としていたが，この境目は IP アドレスが属するアドレスクラス（表 4）によって決定される．

表 4: アドレスクラス

クラス	アドレス範囲	ネットワーク部	ホスト数	用途
A	0.0.0.0 〜 127.255.255.255	8 ビット	16,777,214	大規模ネットワーク用
B	128.0.0.0 〜 191.255.255.255	16 ビット	65,534	中規模ネットワーク用
C	192.0.0.0 〜 223.255.255.255	24 ビット	254	小規模ネットワーク用
D	224.0.0.0 〜 239.255.255.255	-	-	マルチキャスト用
E	240.0.0.0 〜 255.255.255.255	-	-	実験用

ネットワーク部とホスト部の境目を示すビット列（例. 255.255.0.0）のことを『ネットマスク』と呼び，アドレスクラスを元にネットマスクを決定する方式を『クラスフルアドレッシング』と呼ぶ．

IP アドレスは 32 ビット固定であるため，ネットワーク部が大きくなればホスト部は小さくなり，そのネットワークに割り振れるホストの数は少なくなる．逆に，ネットワーク部が小さくなればホスト部は大きくなり，そのネットワークに割り振れるホストの数は多くなる．しかし，クラス A の場合では割り振れるホストの数が 16,777,214 台にもなり，一つのネットワークとして管理するにはあまりにも巨大になってしまうという問題がある．対応策として，次の項で説明する『サブネット』などの手法が用意されている．

3.4 サブネット

クラスフルアドレッシングのような IP アドレスの分類はネットワークの規模に応じて IP アドレスを使い分けるために決められたものである．しかし，IP アドレスの値による固定的なネットマスクの分割ではあまり柔軟にネットワークを構築することはできず，クラス A やクラス B のネットワークの場合にはネットワークがあまりにも巨大になってしまう．

そこで，各ネットワークの管理者が自由にネットワークアドレスとホストアドレスを決定できるようにするために『サブネット分割』という手段が用いられる．既存のネットワークをさらに小さなネットワークに区切り，それを『サブネット』として扱うという手法である．いままでは IP アドレスを『ネットワーク部』と『ホスト部』の 2 つに分けていたが，サブネットに区切った場合，IP アドレスは『ネットワーク部』,『サブネット部』,『ホスト部』の 3 つに分割されることになる（図 4）．このとき，サブネット部の境目を示すビット列（例. 255.255.255.0）のことを『サブネットマスク』と呼ぶ．

アドレスクラスによって定められたネットマスクではなく，ネットワーク管理者が決めた自由なネットマスク（サブネットマスク）を使うことにより，ネットワークの規模などに応じて柔軟

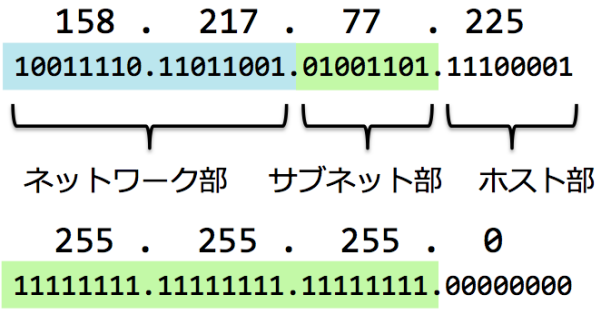

図 4: サブネット分割

にネットワークを構築できるようになる．しかし，サブネットを使ったネットワークではネットマスクが環境ごとに異なるため，サブネットマスクを IP アドレスと同時に表記する必要がある．

サブネットマスクの表記法

サブネットマスクの表記法にはいくつかの種類がある．

- 192.168.1.52/11111111.11111111.11111111.00000000
- 192.168.1.52/255.255.255.0
- 192.168.1.52/0xffffff00
- 192.168.1.52/24

これら 4 種類の表記法はどれも同じサブネットマスクを表している．ソフトウェアによって様々な表記が使われるので，どれが出てきても意味が分かるようにしよう．

3.5 予約済みアドレス

IP アドレスの中にはどのホストにも割り振ることができない特別なアドレスが存在し，これらのアドレスのことを『予約済みアドレス』と呼ぶ．以下に各ネットワークごとに存在する 2 種類の予約済みアドレスについて説明する．

ネットワークアドレス

ホストアドレスのビットが全て 0 になるアドレス．これはネットワークそのものを表すアドレスであり，特定の機器ではなくネットワークそのものを指定したい場合に利用する．

例: 158.217.0.0（10011110.11011001.00000000.00000000）

ブロードキャストアドレス

ホストアドレスのビットが全て 1 になるアドレス．ネットワーク内の全てのホストを表すアドレスとなる． ネットワーク内の全ての機器に対してデータを送信したい場合に利用する．

例: 158.217.255.255（10011110.11011001.11111111.11111111）

各ネットワークは必ずこれら 2 つの予約済みアドレスを持つため，ホスト部が n ビットだとするとホストに割り振れるアドレスは $2^n - 2$ 個となる．予約済みアドレスはこれが全てではなく，他にもループバックアドレス（127.0.0.1）やプライベートアドレスといった特別なアドレスが予約されている．

3.6 ifconfig/ipconfig

ifconfig/ipconfig は IP ネットワーク設定を確認したり，再設定したりするときに使うコマンドである．ネットワーク関係のコマンドとして最も頻繁に使われるものの一つで，ネットワーク管理には欠かせないコマンドである．情報の見方を是非頭に入れておこう．

Mac では ifconfig をターミナルから，Windows では ipconfig をコマンドプロンプトから実行する．

Mac（ifconfig）

```
kosuke-no-MacBook-Air:~ koba$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 68:a8:6d:06:24:2c
        inet6 fe80::6aa8:6dff:fe06:242c%en0 prefixlen 64 scopeid 0x4
        inet6 2001:2f8:3a:1101:6aa8:6dff:fe06:242c prefixlen 64 autoconf
        inet6 2001:2f8:3a:1101:248b:1b1d:327c:c9a0 prefixlen 64 autoconf temporary
        inet 10.1.5.150 netmask 0xffff0000 broadcast 10.1.255.255
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
```

上の実行例では en0 がネットワークに繋がる機器（NIC）となっている．ether の項には MAC アドレス，inet6 の項には IPv6 についての情報が記載されている．
IP アドレス（IPv4）についての情報は inet から始まる行に記載されている．inet 10.1.5.150 は機器に割り振られた IP アドレスを表し，netmask 0xffff0000 はサブネットマスクを表している．

Windows（ipconfig）

```
C:¥Users¥koba>ipconfig

Windows IP 構成


Wireless LAN adapter ワイヤレス ネットワーク接続:

   接続固有の DNS サフィックス . . . : firefly.kutc.kansai-u.ac.jp
   IPv6 アドレス . . . . . . . . . . : 2001:2f8:3a:1101:8c17:cdca:5f99:66c5
   一時 IPv6 アドレス. . . . . . . . : 2001:2f8:3a:1101:e827:6372:57f3:b1c9
   リンクローカル IPv6 アドレス. . . . : fe80::8c17:cdca:5f99:66c5%15
   IPv4 アドレス . . . . . . . . . . : 10.1.5.82
   サブネット マスク . . . . . . . . : 255.255.0.0
   デフォルト ゲートウェイ . . . . . : fe80::230:13ff:fef6:c49f%15
                                       10.1.3.1
```

IP アドレス（IPv4）についての情報は IPv4 アドレスから下の項に記載されている．IPv4 アドレス…: 10.1.5.82 は機器に割り振られた IP アドレスを表し，サブネットマスク…: 255.255.0.0 は見ての通りサブネットマスクを表している．

3.7 DNS

DNS（Domain Name Service）は IP アドレスとドメイン名の対応付けを行うサービスである．ドメイン名から IP アドレスを問い合わせること（正引き）と，IP アドレスからドメイン名を問い合わせること（逆引き）の両方を行うことができる．インターネットを利用する上でなくてはならない存在であり，現在のインターネットにとって必要不可欠なシステムの一つである．

ブラウザで Web ページを閲覧するとき，ブラウザは IP アドレスを使って Web サーバと情報のやり取りを行っている．しかし，ブラウザを利用しているユーザが直接 IP アドレスを入力することは少ない．ユーザが入力するのは多くの場合，IP アドレスと比べて記憶し易い URI（例. https://www.firefly.kutc.kansai-u.ac.jp/xoops/）である．

URI からプロトコル名（https://）とディレクトリ名・ファイル名（/xoops/）を取り払ったものをドメイン名，厳密には FQDN（完全修飾ドメイン名）と呼ぶ．ブラウザはこのドメイン名を使って目的の Web ページの IP アドレスを DNS サーバに対して問い合わせ，IP 通信を行っている．

ネットワークの設定の際には DNS サーバ（及びセカンダリ DNS サーバ）の IP アドレスが求められる．小林ゼミでは DNS サーバ（10.1.3.21，10.1.3.80）を運用しており，ゼミ内ネットワークを使用する際にはこれらの DNS サーバの IP アドレスを設定する必要がある．

3.8 FQDN

FQDN とドメイン名は混同されがちであるが，厳密には異なるものである．ドメイン名はサーバが存在するネットワークを特定するための文字列であるが，FQDN はさらにホスト名をドメイン名に加えたものである．そのため，FQDN はホスト名とドメイン名に分割して考えることができる（図 5）.

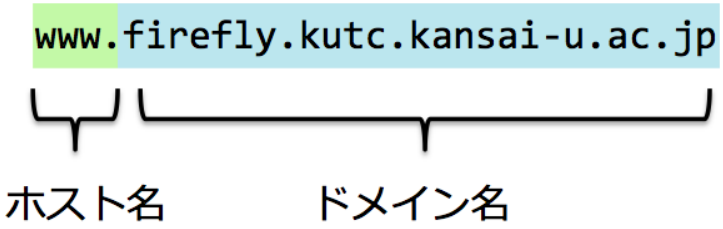

図 5: ドメイン名とホスト名

FQDN は階層型の構造となっており，www.firefly.kutc.kansai-u.ac.jp は「日本（jp）の大学（ac）の関西大学（kansai-u）の高槻キャンパス（kutc）の小林ゼミ（firefly）の Web サーバ（www）」という意味である．

システムによっては末尾にドット（.）をつけることを表記のルールとしていることもあるので（例: www.firefly.kutc.kansai-u.ac.jp.）注意が必要である．

3.9 nslookup

nslookup は DNS 問い合わせを手動で実施するためのコマンドである．任意の DNS サーバを指定して問い合わせを行うことも可能であり，DNS のトラブルシューティングでは非常に良く使用される．ネットワーク関連の基本的コマンドの一つである．

```
nslookup [FQDN または IP アドレス] [任意の DNS サーバ名または IP アドレス]
```

一番基本的な nslookup の使い方は，引数に FQDN を指定して nslookup を起動することである．この使い方を『正引き』と呼び，FQDN に対応する IP アドレスが DNS サーバから取得され

る（図 6）.

```
kosuke-no-MacBook-Air:~ koba$ nslookup cririn.firefly.kutc.kansai-u.ac.jp
Server:         172.20.10.1
Address:        172.20.10.1#53

Non-authoritative answer:
Name:   cririn.firefly.kutc.kansai-u.ac.jp
Address: 158.217.77.225
```

図 6: 正引き

次によく使われる nslookup の使い方は，引数に IP アドレスを指定して nslookup を起動することである．この使い方を『逆引き』と呼び，IP アドレスに対応する FQDN が DNS サーバから取得される（図 7）.

```
kosuke-no-MacBook-Air:~ koba$ nslookup 158.217.77.225
Server:         172.20.10.1
Address:        172.20.10.1#53

Non-authoritative answer:
225.77.217.158.in-addr.arpa     name = cririn.firefly.kutc.kansai-u.ac.jp.
```

図 7: 逆引き

4 グローバル IP とプライベート IP

インターネットは接続先となる端末の所在を表すため，IP アドレスを用いているが，これはグローバル IP アドレスと呼ばれ世界中のネットワークに割り当てられる．また，グローバル IP アドレスは住所と同じように世界中で重複することはなく，グローバル IP アドレスを割り当てられた端末に対しては，世界中から宛先として通信することが可能である．

ここでは，ネットワークの疎通を確認する ping コマンドを使用してグローバル IP アドレスへの通信を確かめる．例として，小林研究室のグローバル IP アドレスである 158.217.77.225 に対して，ping コマンドを以下のように入力する．

```
# ping 158.217.77.225
PING 158.217.77.225 (158.217.77.225): 56 data bytes
64 bytes from 158.217.77.225: icmp_seq=0 ttl=64 time=2.701 ms
64 bytes from 158.217.77.225: icmp_seq=1 ttl=64 time=3.708 ms
64 bytes from 158.217.77.225: icmp_seq=2 ttl=64 time=3.807 ms
                    :
※終了は ctrl+c
```

ping コマンドの結果を確認するとグローバル IP アドレスに対して疎通確認することができる．

グローバル IP アドレスは，前述した通り世界中で重複することはなく，IP アドレスの割り当て個数は 32 桁の 2 進数で約 43 億のパターンが存在する．しかし，この個数は世界人口約 70 億人に対してインターネットを利用するデバイスが増加し続ける中でアドレスの枯渇が問題となっている．そこで，企業や家庭などの限られたエリアごとにネットワークを構成し，このネットワーク内はグローバル IP アドレスの代わりに，プライベート IP アドレスというものを用いて割り当てを行っている．

プライベート IP アドレスは，企業や家庭内などのローカルなネットワーク内でのみ有効なアドレスであり，各ネットワークごとに所属する端末間で自由にアドレスを設定することができる．これにより，各ネットワークごとに，プライベート IP アドレスを用いることで IP アドレスの割り当て個数を節約できる．しかし，プライベート IP アドレスは自由に設定されるため，インターネット上からは直接参照することができず，また，異なるネットワークに属する端末同士も直接通信することができない．プライベート IP アドレスを使用したネットワーク通信については後述する．

異なるネットワーク同士が通信できないことを確認してもらうため，ping コマンドを使用してプライベート IP アドレスから別のネットワークのプライベート IP アドレスに対して通信可能かどうかを確認する．例として，現在接続されている関西大学のネットワーク（kuwifi）から，小林研究室のネットワークで運用されるサーバである cririn のプライベート IP アドレス 10.1.3.10 に対して，ping コマンドを以下のように入力する．

```
# ping 10.1.3.10
PING 10.1.3.10 (10.1.3.10): 56 data bytes
ping: sendto: No route to host
```

結果を確認すると，宛先が見つからないため，プライベート IP アドレスには直接通信できないことが分かる．

上の例から異なるネットワークに属する端末同士は直接的に通信することができないことを確認したが，これにより，IP アドレスの節約に加え，他のネットワークと隔離されていることを利

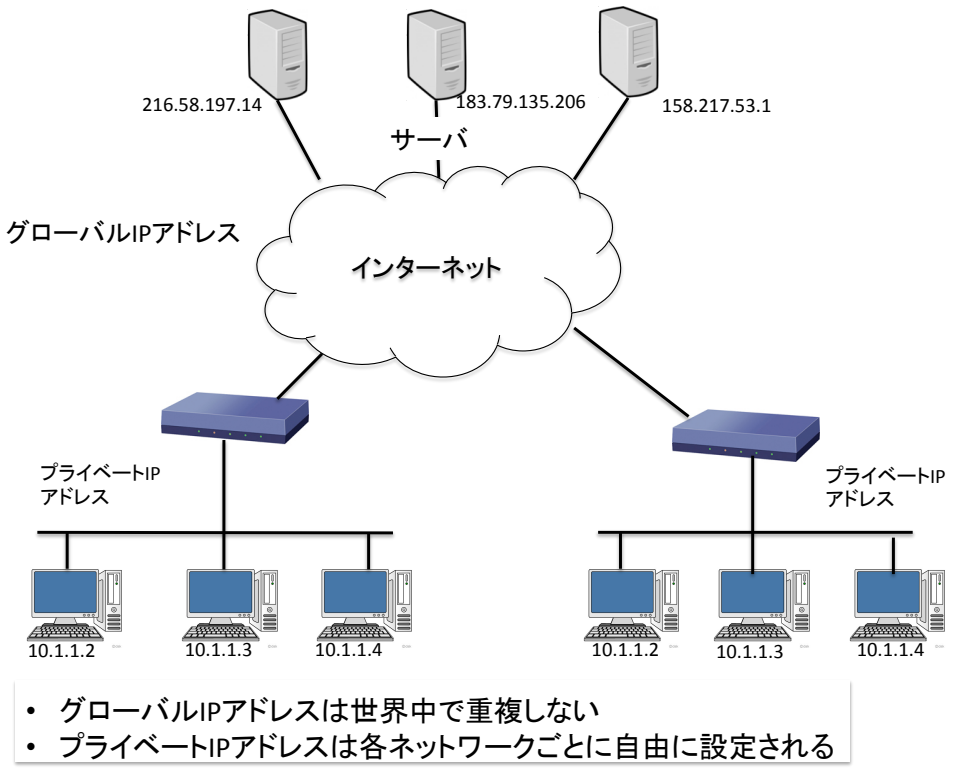

図 8: グローバル IP とプライベート IP

用して，セキュリティの確保を実現することができる．ネットワークを区切ることによる利点として，プライベート IP アドレスで構成されるネットワークは他のネットワークから切り離されるため，意図しないユーザからのアクセスを防ぐことができる．さらに，ネットワークに侵入する通信をファイアウォールなどを用いてフィルタリングすることで，ネットワーク全体のセキュリティを確保することができる．

5 ルータ

上の説明において，異なるネットワーク同士は直接通信できないと説明した．しかし，異なるネットワークであるはずのグローバル IP アドレスに対して ping コマンドにより疎通を確認することができた．また，私たちは日常においてインターネットを使用する際，無線 LAN などから設定されたプライベートアドレスを用いて，インターネット上に存在する Web サイトなどの Web サービスを利用している．これらのネットワークを越える通信はルータによって実現される．ルータは異なるネットワーク間においてデータを中継する機器であり，インターネットに接続されるルータは外側にグローバル IP アドレス，内側にプライベート IP アドレスを割り当てられる．このルータを用いることで，複数のネットワークで構成されるインターネット上においても，ルータが通信経路を中継することで目的の端末までデータを届けることができる（図 9）．こういった仕組みから，私たちはインターネット上の離れたネットワークに存在するサービスを利用することができる．実際に通信経路を確認するため，例としてネットワーク経路を調べる traceroute（tracert）コマンドを使用して関西大学のサーバまでの経路を確認する．

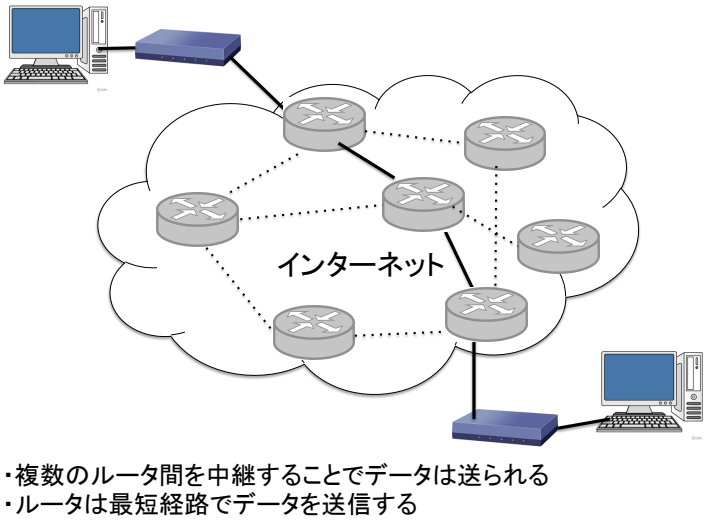

図 9: ルータによるデータの転送

Mac は以下のコマンドを入力する.

```
# traceroute sh.edu.kutc.kansai-u.ac.jp
traceroute to sh.edu.kutc.kansai-u.ac.jp (158.217.53.13), 64 hops max, 52
byte packets
 1  witccnt003.itc.kansai-u.ac.jp (172.29.70.203)  2.003 ms  0.975 ms  0.960
 ms
 2  172.29.143.254 (172.29.143.254)  2.111 ms  2.154 ms  2.371 ms
 3  158.217.103.254 (158.217.103.254)  3.114 ms  3.504 ms  2.996 ms
 4  172.17.5.240 (172.17.5.240)  4.834 ms  4.463 ms  4.421 ms
 5  158.217.4.1 (158.217.4.1)  4.249 ms  3.509 ms  3.742 ms
 6  sh.edu.kutc.kansai-u.ac.jp (158.217.53.13)  2.920 ms !Z  2.992 ms !Z
 4.961 ms
```

windows は以下のコマンドを入力する.

```
> tracert sh.edu.kutc.kansai-u.ac.jp
sh.edu.kutc.kansai-u.ac.jp[158.217.53.13] へのルートをトレースしています
経由するホップ数は最大 30 です：
1     1ms    3ms     2ms   witccnt003.itc.kansai-u.ac.jp [172.29.70.203]
2     2ms    5ms     1ms   172.29.143.254
3     3ms    2ms     3ms   158.217.103.254
4     9ms    4ms     4ms   172.17.5.240
5     3ms    3ms     4ms   158.217.4.1
6     3ms    3ms     4ms   sh.edu.kutc.kansai-u.ac.jp [158.217.53.13]
トレースを完了しました。
```

traceroute（tracert）コマンドの結果から，宛先までの経路が中継されていることが確認できる.

6 ドメイン情報

前の説明で宛先に関西大学のドメインを指定したように，インターネット上のサービスを利用する際は IP アドレスを指定するより，アドレスに対応付けられたドメイン名を使用することが一般的である．また，このドメインについては，whois サービスにより所有先の情報を調べることができる．whois サービスは，登録者情報，ネームサーバホスト情報，担当者情報などを確認することができる．また，代表的なものとして，ANSI（ `http://whois.jprs.jp/` ）や JPRS（`http://whois.ansi.co.jp/`）などが Web サービスを提供している．(Unix 系 OS の場合，whois コマンドを使用することでも検索することができる．)

ドメイン情報を調べる whois サービスの検索方法を説明する．まず，インターネットブラウザから JPRS を検索し，検索結果から JPRS WHOIS /JPRS を選択する（図 10）．次に，ドメイン名登録情報検索に検索キーワードとして，関西大学のドメインである kansai-u.ac.jp を入力し，検索ボタンを押下する（図 11）．検索結果からドメインの登録情報を確認することができる（図 12）．

図 10: JPRS の検索

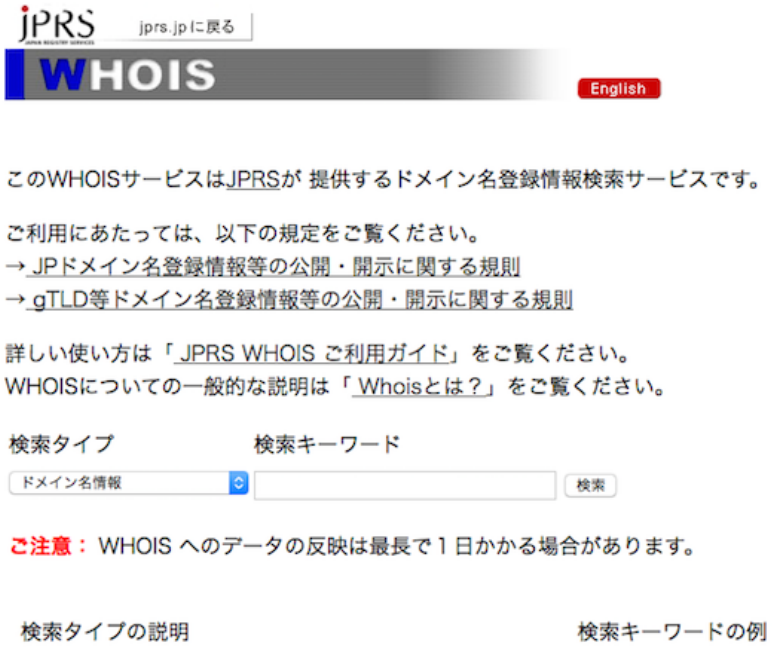

図 11: ドメイン名登録情報検索

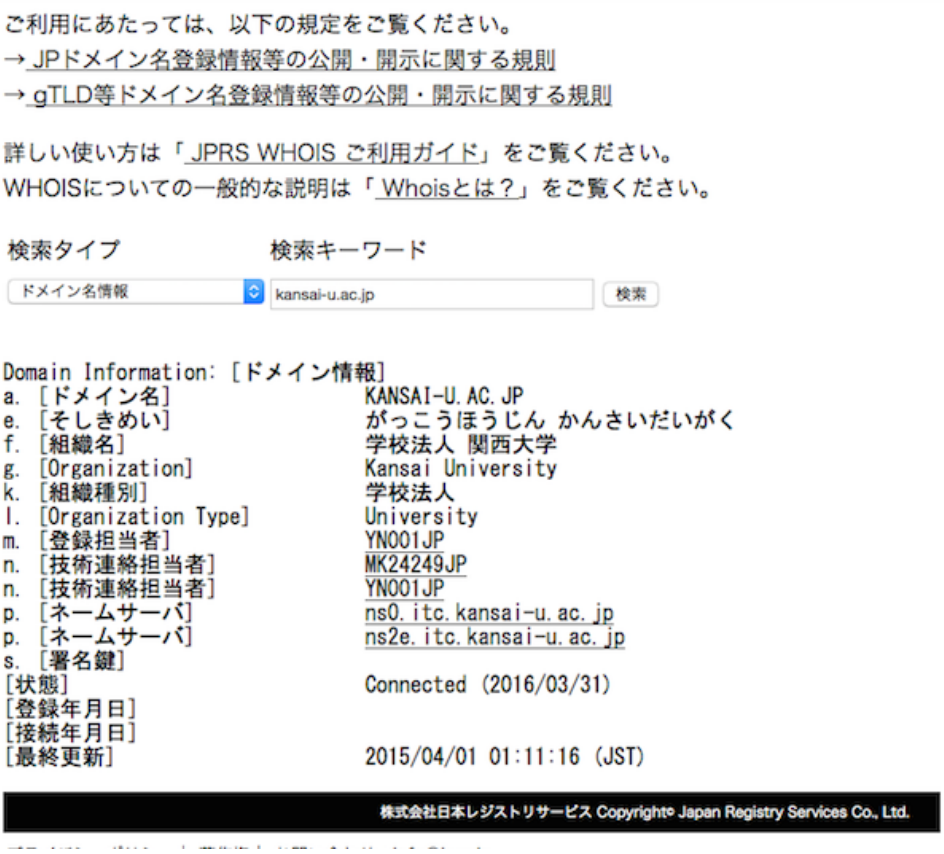

図 12: ドメイン情報結果

7 NAT と NAPT

ここでは，ルータの機能としてグローバル IP アドレスとプライベート IP アドレスの通信について説明する．

前述の IP アドレス枯渇問題から，プライベート IP アドレスの割り当てを説明したが，グローバル IP アドレスが振られるインターネット側からは，プライベート IP アドレスを認識することはできない．そこで，グローバルネットワークとプライベートネットワークの通信を可能にするため，ルータの NAT（Network Address Translation）機能を用いている．インターネットに繋がるルータには，外側にグローバル IP アドレス，内側にプライベート IP アドレスを割り当てられているが，NAT はこのグローバル IP アドレスとプライベート IP アドレスの変換を行う．

NAT の仕組みとして，プライベート IP アドレスが割り振られた端末が，インターネット上のグローバル IP アドレスと通信を行う場合について説明する．

まず，プライベート IP アドレスから送信があった場合，ルータは送信元となるプライベート IP アドレスはルータの外側にあるグローバル IP アドレスに変換する．そして，送信元がグローバル IP アドレスとなったデータは，インターネット上の宛先へと送信される．次に，インターネット上から送信元に対して返信があった場合は，変換された送信元であるルータの外側のアドレスに対して返信される．そして，返信を受け取ったルータは送信先を本来の送信元であるプライベート IP アドレスに転送することで，インターネット上との通信を可能にする．

このように，ルータが中継となってアドレスを変換することで，プライベート IP アドレスとグローバル IP アドレス間で通信を行うことを可能にする（図 13）．

しかし，NAT でのアドレス変換は一つのグローバル IP アドレスに対して，一つのプライベート IP アドレスを対応づけるため，グローバル側との通信は一つの端末のみに限られてしまう．そこで，NAT 機能の拡張として NAPT（Network Address Port Translation）の機能が用いられる．NAPT は，一つの端末しか接続できない問題に対応するため，ポート番号を用いる．ポート番号はサーバが提供するサービスを識別する番号である．このポート番号を端末ごとに割り当て，番号をもとに宛先の端末を識別することで，ポート番号を利用して複数の端末により通信を行うことが可能となる．

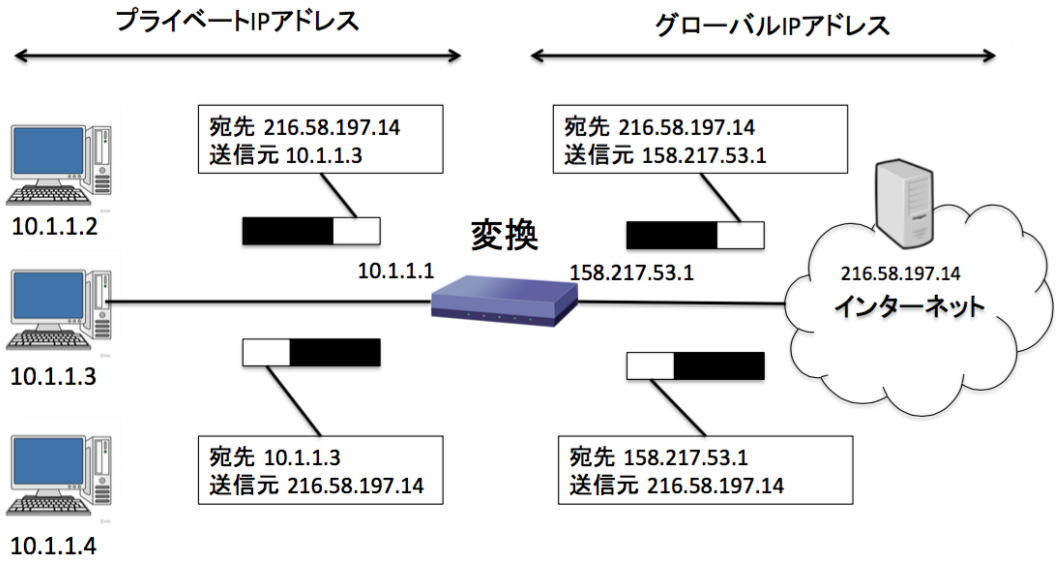

図 13: NAT によるアドレス変換

7.1 DHCP

ここまで，プライベート IP アドレスを利用した技術の説明をした．プライベート IP アドレスの使用に伴い，ネットワークで管理される端末数も増加し，IP アドレスの管理も大変になる．しかし，日常生活で利用する際に IP アドレスを設定する機会はほとんどない．多くの場合，実際の IP アドレスの設定は DHCP（Dynamic Host Configuration Protocol）により行われる．また，設定機能を持った機器を DHCP サーバと呼ぶ．これは，IP アドレスや DNS などのネットワーク情報を持っており，ネットワークに接続した端末に対して設定情報を提供する．また，端末がネットワークから離脱すると DHCP リソースを更新する．このように DHCP は，IP アドレスの割り当てなど，ネットワーク情報を自動的に管理することができる．

8 VPN（Virtual Private Network）

自宅など，外部のネットワークからゼミ内のプライベート IP アドレスが割り振られているマシンなどを操作する場合はネットワークが異なるため，直接的に利用することはできない．そこで VPN という技術を用いることで，異なるネットワーク同士を仮想的に同一のネットワークに属しているように見せかけ，異なるネットワークの端末と通信することができるようになる．

VPN による通信の仕組みとして，トンネリングによりネットワークを繋いでいる．トンネリングでは，もとの通信内容にヘッダが加えられ，加えられたヘッダにより通信内容はカプセル化される．このカプセル化を用いることで接続元でカプセル化を行い，接続先でカプセル化を解除し通信内容を取り出すことで，ネットワーク同士をトンネルで繋いだように見立てて使用することが可能となる．さらに，データをカプセル化する際は，データを暗号化することで安全にネットワークを越えて通信することが可能となる（図 14）．VPN で利用されるプロトコルには，IPsec/PPTP/L2TP/L2F/MPLS などがある．

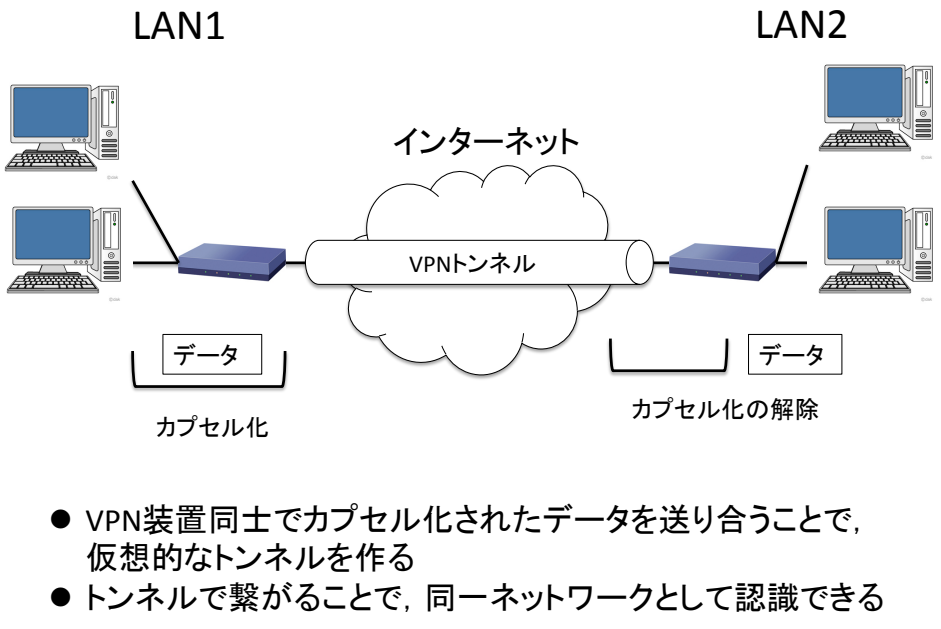

図 14: VPN の概要

8.1 小林ゼミの VPN

小林ゼミの VPN 環境において利用可能なプロトコルは L2TP Over IPSec と PPTP がある．Mac はいずれのプロトコルも利用することが可能である．ただし，L2TP Over IPSec は PPTP よりもセキュアなため前者の使用を推奨する．はじめに，VPN の設定に必要な情報を表 5 に示す．サーバアドレスは cririn.firefly.kutc.kansai-u.ac.jp と同じ 158.217.77.225 を使用する．アカウント名とパスワードは Xoops にログインする時と同じものを使用する．共有シークレットは口頭で伝える．

表 5: VPN 設定に必要な情報

項目	値
サーバアドレス	158.217.77.225
アカウント名	Xoops で使用するアカウント名
パスワード	Xoops で使用するパスワード
共有シークレット	******

8.2 VPN の設定（Mac の場合）

システム環境設定からネットワークを開く（図 15）．次に＋から新しいサービスを作成する．新たに表示されたウィンドウでインタフェースを VPN，VPN タイプを L2TP Over IPSec か PPTP を選択，サービス名は任意に入力して，作成を押す（図 16）．新しいサービスが作成されたので，そのサービスを選択した状態で設定を入力する．構成はデフォルト，サーバアドレスは 158.217.77.225，アカウント名を入力する．次に認証設定を押す．ここで L2TP over IPSec による VPN 構成の場合は，ユーザ認証でパスワード，コンピュータ認証で共有シークレットを入力する（図 17）．PPTP による VPN 構成の場合はパスワードを入力する．さらに PPTP の暗号化は自動（128 ビットまたは 40 ビット）にする．以上全ての項目を入力して，適応と接続を行う．またメニューバーに VPN の状況を表示をチェックすることで接続状態や接続時間が表示される．これは VPN 接続のショートカット機能も有しているのでチェックすることを推奨する（図 18）．

図 15: ネットワーク設定

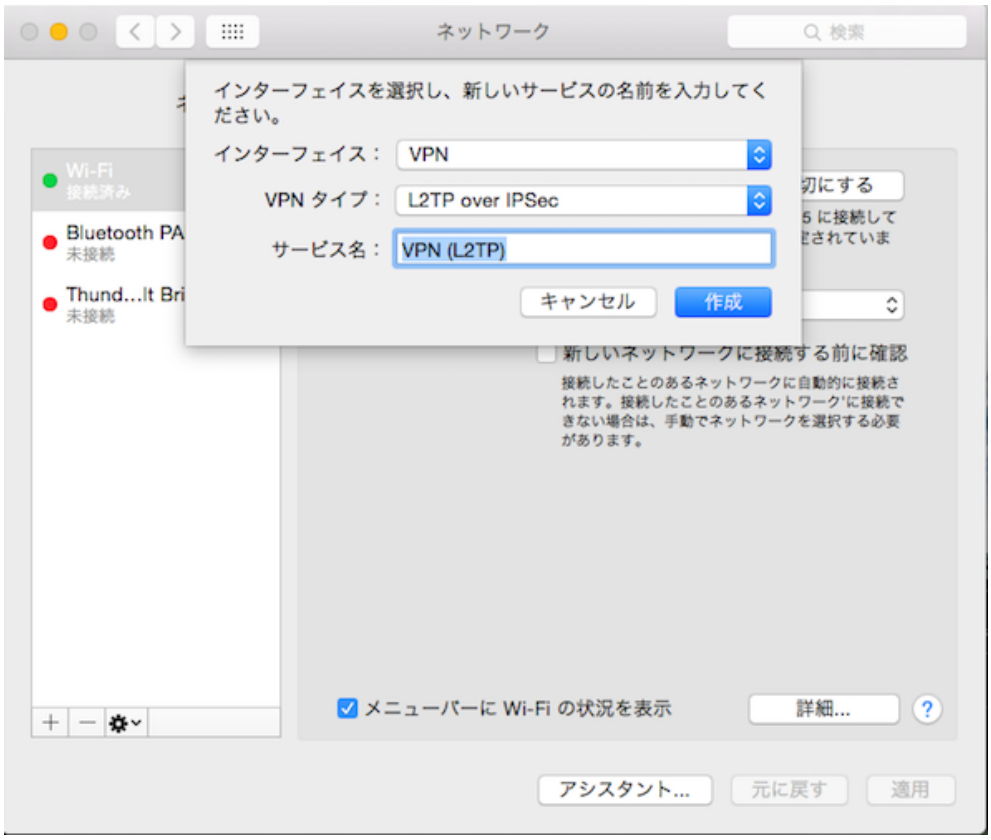

図 16: vpn サービスの作成

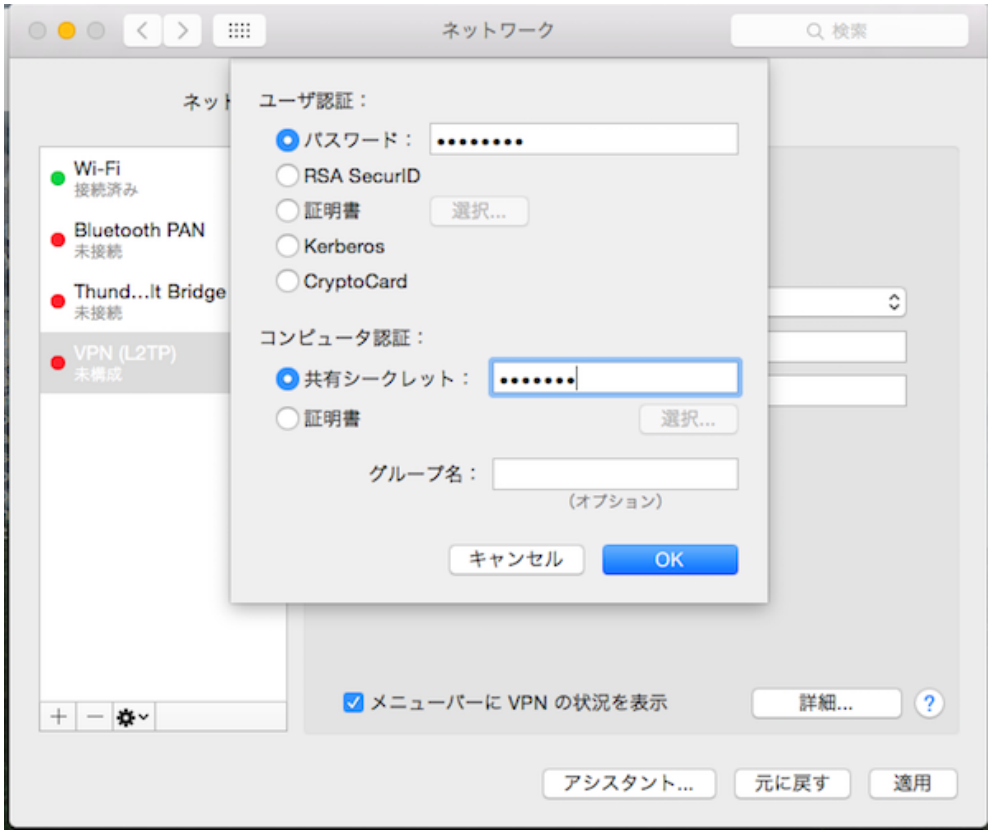

図 17: L2TP Over IPSec の認証設定

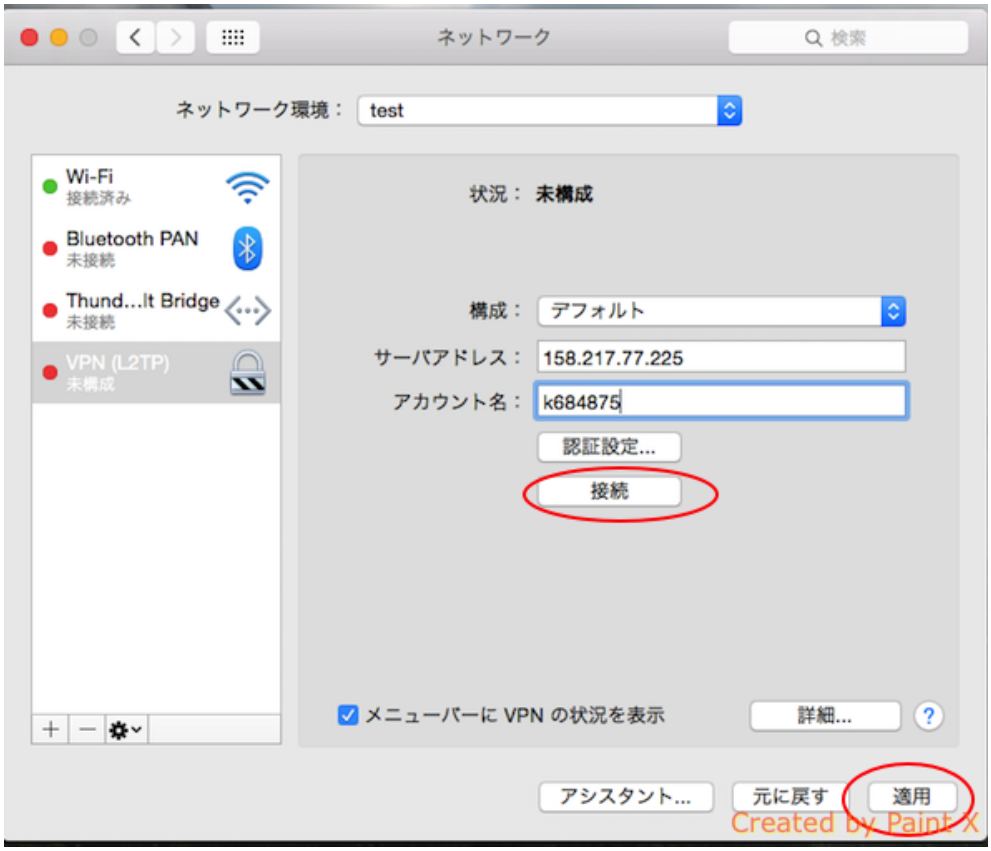

図 18: VPN の適応と接続

8.3 VPN の設定（Windows10 の場合）

ホーム画面左下の Windows アイコンから設定をクリックする（図 19）．開いたウィンドウからネットワークとインターネットを選択する（図 20）．次に，開いたメニュー左の VPN から VPN 接続を追加するを選択する（図 21）．VPN の設定ウィンドウについては，各項目に正しく入力する．それぞれ，VPN プロバイダーは windows（ビルトイン）．接続名は VPN．サーバ名またはアドレスには 158.217.77.225．VPN の種類は PPTP．ユーザ名とパスワードを入力し保存を押す（図 22）．ウィンドウを閉じると，関連設定のメニューにあるアダプターのオプションを変更を選択する（図 23）．作成した VPN 設定のアイコンが表示されるのを確認し，アイコンの上で右クリックからプロパティを選択（図 24）．セキュリティタブをクリックし，VPN の種類が PPTP になっているのを確認とデータの暗号化には暗号化が必要を選んで OK を押す（図 25）．最後に，ネットワークとインターネットのウィンドウから接続を選択し，接続中になっていることが確認できれば完了となる（図 26）．

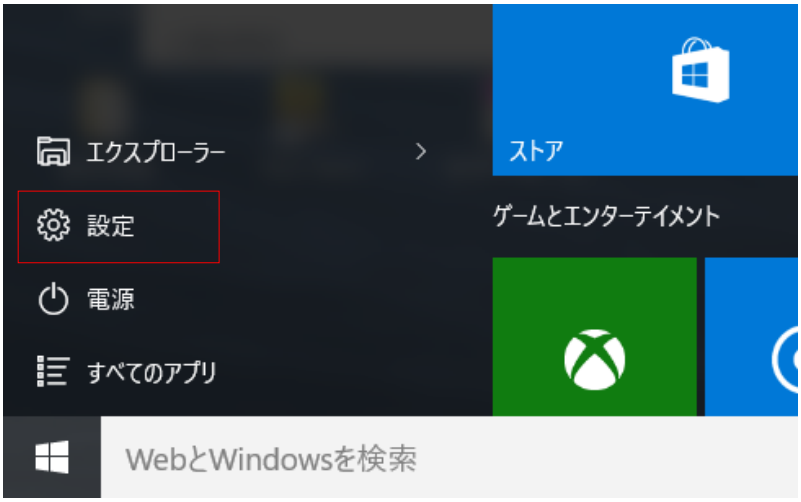

図 19: VPN の設定 1

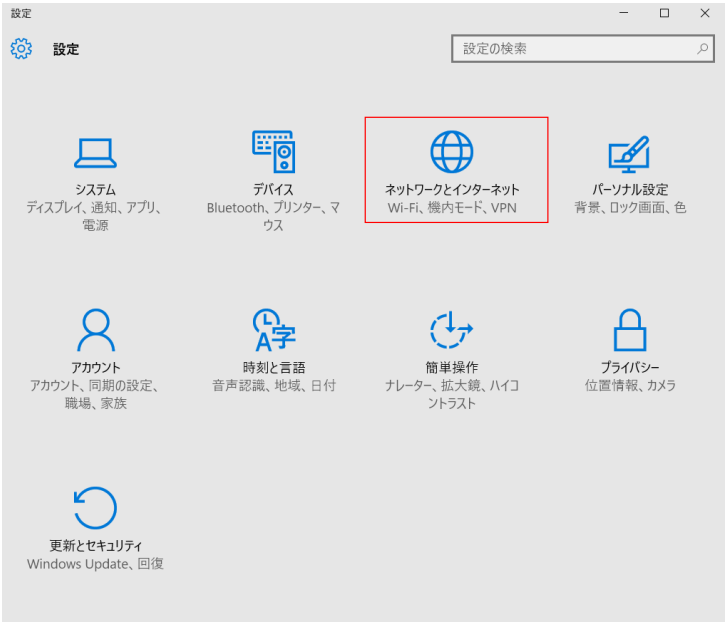

図 20: VPN の設定 2

図 21: VPN の設定 3

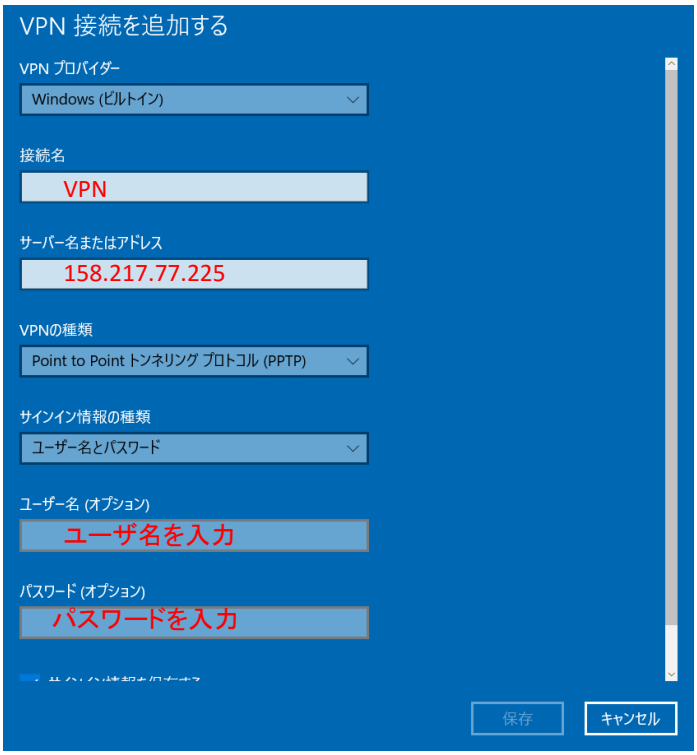

図 22: VPN の設定 4

図 23: VPN の設定 5

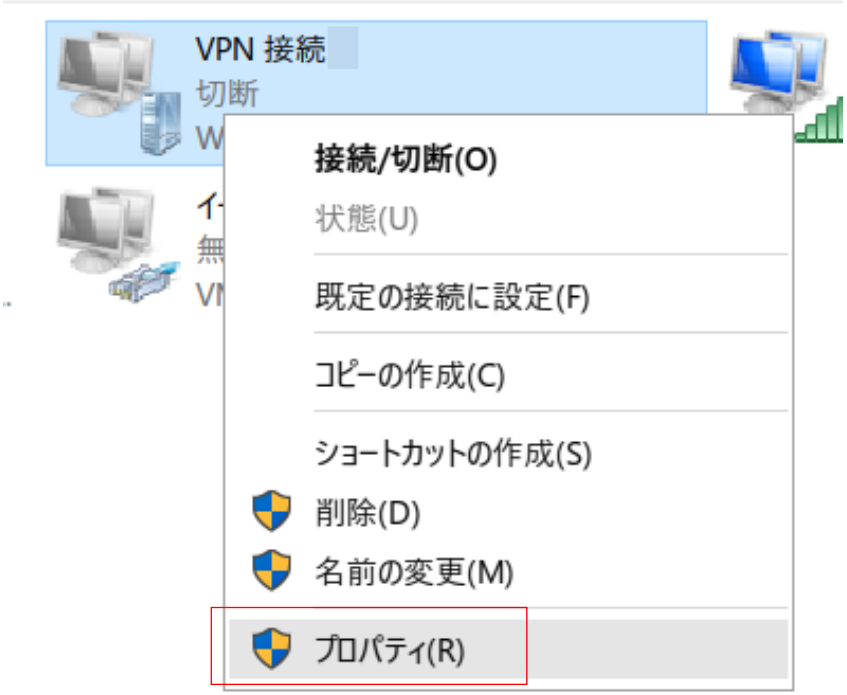

図 24: VPN の設定 6

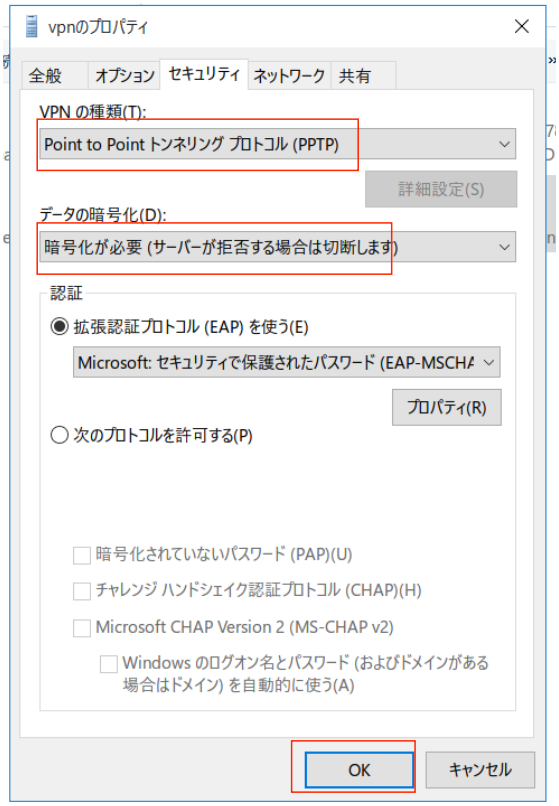

図 25: VPN の設定 7

図 26: VPN の設定 8

9 TCP/UDP

TCP/IP 階層モデルにおけるトランスポート層は，ネットワーク層で端末（ノード）間の橋渡しがなされ送られてきたパケットについて，これが届いたかどうかの確認をしたり，これの順序を整理したり，データが壊れていないかを確認したり，大きすぎるデータを分割したり，データの送信量を制御する．また，データを適切なアプリケーションに引き渡す役割もある．

トランスポート層の代表的なプロトコルとして，TCP と UDP がある．これら二つの違いをごく簡単に説明すると，信頼性のある通信を重視するものが TCP，伝送効率の高い通信を重視するものが UDP である，となる．

それではまず，UDP について説明を始めよう．

9.1 UDP

UDP は，通信の信頼性よりも，速さやリアルタイム性が要求されるような場合において使用されるプロトコルである．UDP のヘッダを図 27 に示す．

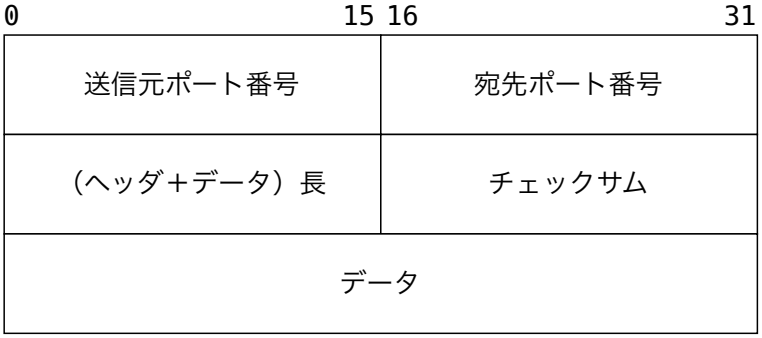

図 27: UDP ヘッダ

UDP のヘッダは，このように極めてシンプルな構造になっている．データを転送する際，送信側はチェックサムの計算を行って送信するだけであり，受信側はチェックサムを計算してデータに誤りが無いか確認した後，受け取った順にそのままアプリケーションに渡すだけである．後述する TCP において使われている 3-way handshake や確認応答，順序制御，再送制御，伝送制御などの機能は無く，処理がシンプルであるが故に高速なのである．

ただし，パケットが確実に到達するという保証が無いため，何らかの原因でパケットロスが発生した場合にでも処理を継続できるようにアプリケーション側で工夫する必要がある．

このプロトコルは，DNS や NTP，DHCP などのデータ量の少ないものや，音楽や映画などのストリーミング配信，またオンラインゲームのようなリアルタイム性が要求されるもので使われることが多い．

9.2 TCP

TCP は，信頼性の高い通信を実現するために使用されるプロトコルである．TCP のヘッダを図 28 に示す．シーケンス番号では送信するデータの通し番号を管理しており，確認応答番号（ACK 番号）ではどのシーケンス番号までデータを受信したかを示している．

フラグでは，その TCP パケットがどのような種類のものなのかを示している．フラグ名とその意味を整理したものを表 6 に示す．

また，TCP をベースとする HTTP 通信の例を表 7 に示す．これを参考にしつつ TCP 通信の特徴を説明していく．

まず，TCP では UDP のようにデータを一方的に送りつけるのではなく，事前に接続を確立する手順を踏む．この手順のことを 3-way handshake と呼ぶ．表 7 の例では上から三つ分のパケッ

図 28: TCP ヘッダ

表 6: TCP のフラグ一覧

URG	この TCP パケット内に緊急データが含まれていることを示す
ACK	TCP ヘッダ内に有効な ACK 番号が含まれていることを示す
PSH	受信したデータをすぐにアプリケーションに引き渡すよう要求する
RST	TCP 接続の中断，拒否を示す
SYN	3-way handshake 時に使われ，ACK 番号を同期させる
FIN	TCP 接続を終了させる

トがこれにあたる．

まず，接続する側（クライアント）は接続される側（サーバ）に対して SYN フラグを立てたパケットを送信し，上りのコネクション確立を試みる．これを受け取ったサーバ側は ACK と SYN フラグを立てたパケットを送信し，上りのコネクション確立を承認するとともに，下りのコネクション確立を試みる．これを受け取ったクライアント側は ACK フラグを立てたパケットをサーバに送信し，下りのコネクション確立を承認する．以上の手順により信頼性のあるコネクション確立が完了し，これ以降でデータの通信が行われる．

送信したデータが全て受信側に到着したかを確認するために，データのバイト数とシーケンス番号，ACK 番号が用いられる．例として送信側が 394 バイトのデータを送信するケースを考えてみる．送信側はデータと共にシーケンス番号を送信する．これを受け取った側はデータのバイト数とシーケンス番号を足し，これを ACK 番号として送信側に送る．これを受け取った送信側で，受け取った ACK 番号が，送信したデータのバイト数と送信時のシーケンス番号を足したものと一致していれば全てのデータが到達していることがわかる．もしここで数値が一致しなければ，再度データを送信することになる．

表 7: HTTP 通信の例

Src	Dst	Flag	Length	Seq num	Ack num	HTTP
Client	Server	SYN	0	0	0	
Server	Client	SYN，ACK	0	0	1	
Client	Server	ACK	0	1	1	
Client	Server	PSH，ACK	394	1	1	GET /∼bob/sample.html HTTP/1.1
Server	Client	PSH, ACK	353	1	395	HTTP/1.1 200 OK
Client	Server	ACK	0	395	354	
Server	Client	FIN，ACK	0	354	395	
Client	Server	ACK	0	395	355	

9.2.1 パケット転送効率の改善

上述の通り，TCP ではパケットを送信し，ACK を受け取るという流れを繰り返してデータ転送を行う．だがこのままだとパケット送信から ACK 受信までの時間（RTT，Round Trip Time）の関係で，送信側で送信準備ができていたとしても ACK を受け取れない限り送信できず，転送の効率が悪化する．

これを改善するために，TCP ではウインドウ制御の一つの方法である，スライディングウインドウ方式という仕組みがある．TCP ヘッダの中にはウインドウサイズという，受信側が受信できるデータ量を送信側に知らせるフィールドがある．送信側はこのウインドウサイズをもとに，ACK を待たずに次々にパケットを送信する．一定量パケットを送信したら ACK パケットを待ち，ACK パケットを受け取ったらまた次のパケットを送っていくという方法になる．ウインドウサイズに収まるサイズのデータを送り，どこか一つのパケットに対応する ACK を受信したらそこまでのパケットは到達しているとみなしてウインドウを横にずらして新たなパケットを送信するイメージである．

スライディングウインドウ方式により，送信側の転送効率は上がる．だがこれではネットワークの混雑度合いを考慮できておらず，次々にパケットを送信し続けてしまうことで混雑度合いをさらに悪化させてしまう．この問題に対処するために，輻輳制御という方法が用いられている．初めからウインドウサイズ一杯でパケットを送信するのではなく，徐々に転送量を増やしていき，輻輳が発生した場合に再度ウインドウサイズを小さくして再度転送量を増やすというものだ．

以上の仕組みはパケットの送信側において，効率よくパケットを送信するための仕組みである．ただ，パケットの受信側で処理に時間がかかるような場合，次々にパケットを送信されてしまうと受信側は処理がパンクしてしまう恐れがある．これを回避するためにもウインドウサイズは用いられる．パケットが送られてくるたびにウインドウサイズを減らして送信側に通知し，処理がパンクしそうになるとウインドウサイズをゼロにして送信側に通知する．これにより送信側は受信側の状況を見てパケットの送信を中断でき，受信側で余裕ができたらこれを通知して通信を再開させたり（ウインドウ更新通知），送信側が受信側に余裕ができたかを問い合わせて通信を再開することができる（ウインドウプローブ）．

以上のような仕組みを用いて TCP は転送効率の向上を行っている．

9.3 チェックサム

TCP や UDP（と IP）にはチェックサムという仕組みがあり，これを用いてデータの損傷を検出できるようになっている．計算に用いるのは，TCP あるいは UDP ヘッダとデータ，擬似ヘッ

ダの三つである．擬似ヘッダとは仮想的なヘッダで，実際の TCP パケット内には含まれておらず，チェックサムを計算する際にのみ利用する．

送信側は，まず TCP あるいは UDP ヘッダとデータ，擬似ヘッダの三つで合わせて 16 ビットの整数倍になるようにデータの長さを調節する．また，あらかじめチェックサムフィールドをゼロで埋めておく．そして先ほど調節したデータと TCP あるいは UDP ヘッダ，擬似ヘッダについて 16 ビットごとに 1 の補数を求め（ビット反転させ），その総和を求め，これの 1 の補数をチェックサムフィールドに入れて送信する．受信側は，同様に擬似ヘッダを付けた状態で 16 ビットずつ 1 の補数を求め，その総和を計算する．データが破損していない場合はこの計算結果が 0xFFFF になる．これは送信側がチェックサムフィールドに入れた値がチェックサムフィールド以外の補数和の 1 の補数であり，受信側が行っていることはチェックサムフィールド以外の補数和にその 1 の補数を足す処理をしている，ということから分かるのではないかと思う．

9.4 ポート番号

TCP/IP 階層モデルにおけるトランスポート層ではアプリケーション層との間でのデータのやり取りを担うが，どのアプリケーションからのデータなのか，また，どのアプリケーションに対してのデータなのかを識別するためにポート番号が用いられる．ポート番号は TCP と UDP でそれぞれ 0 番から 65535 番までであり，その中で大きく三つの種類に分けられる．この分け方を表 8 に示す．

表 8: ポート番号の種類

Well Known Ports	0 ～ 1023
Registered Ports	1024 ～ 49151
Dynamic and/or Private Ports	49152 ～ 65535

まず，0 番から 1023 番までは Well Known Ports（System Ports）と呼ばれ，例えば SSH（22）や HTTP（80），HTTPS（443）などの一般的によく利用される主要なサービスのために登録されている．また，Unix 系 OS においてこの範囲のポートは管理者の権限を持つユーザのみが利用できる．1024 番から 49151 番までは Registered Ports（User Ports）と呼ばれ，一般的なサービスではないが，登録制によって割り当てられるものとなる．49152 番から 65535 番までは Dynamic and/or Private Ports と呼ばれ，クライアント側で自動的に利用されるもの，あるいはユーザが自由に利用できるもの，とされている．これらポート番号の管理や割り当ては，IANA（Internet Assigned Numbers Authority）が行っている．

10 アプリケーション層

アプリケーション層は，OSI 参照モデルの第 7 層，また TCP/IP 参照モデルでは第 4 層に位置するレイヤのことを指す．このレイヤは，ネットワークを介したアプリケーション（ソフトウェア，プログラム）の通信プロトコルを定義している．これらのアプリケーションに対して，透過的なネットワーク通信を実現するインタフェースを提供し，ユーザアプリケーションによる情報のやり取りを容易にしている．例えば，インターネット上においてデータ通信の要となっているのは HTTP とよばれるプロトコルである．従って，普段 Web ブラウザでインターネット上のページを見るときは，そのページを提供している Web サーバとの間で HTTP 通信を行っていることが多い．

アプリケーション層で定義されているプロトコルには，代表的なものとして，HTTP や DHCP，DNS，SMTP，Telnet，FTP が存在する．この他にも切りがないほど数多くのプロトコルが定義されているが，本節では，そのなかでも特にメジャーなプロトコルについて説明する．

10.1 HTTP

HTTP（HyperText Transfer Protocol）とは，インターネット上においてコンテンツ（HTML，テキスト，画像等）を送信するための規定を定めたものである．現在の主流なバージョンは，HTTP/1.1 と HTTP/2.0 である．近年は，HTTP/2.0 の仕様が標準として定められたことにより，続々と HTTP/2 サーバが実装されており，今後の主流になると考えられる．また HTTP では，デフォルトで TCP の 80 番ポートを使用する．

HTTP は通信を暗号化しないため，仮に通信を盗聴された場合に内容が閲覧されたり，通信内容を改ざんされる危険性がある．そのため，HTTP の通信をセキュアにした HTTPS という規格も存在している．

10.1.1 HTTP 通信の流れ

HTTP における典型的な通信の流れは，まず Web クライアント（通常は Web ブラウザ）が Web サーバに存在するコンテンツに対して HTTP リクエストを発行するところから始まる．このとき，Web サーバ上にあるコンテンツは，URI（Uniform Resource Identifier）によって一意に識別される．Web サーバ側は HTTP リクエストを受け取った後，それに対応する HTTP レスポンスを Web クライアントに返す．HTTP 通信は，実際にこのように単純な仕組みで成り立っている．実際の通信の流れを，図 29 に示す．

この例では，Web ブラウザを用いて `http://www.kansai-u.ac.jp/index.html` という URL にアクセスしようとしている．Web ブラウザは，`www.kansai-u.ac.jp` の Web サーバに対して，index.html というコンテンツを要求する HTTP リクエストを送信する．そして，Web サーバ側は index.html を含む HTTP レスポンスを返す．最後にクライアント側では，レスポンス内容である index.html を Web ブラウザ上に表示している．

また上記の例における通信をテキストベースで表現したものが，リスト 1 とリスト 2 に当たる．「/index.html」という URI に対して，「GET」というメソッドを用いて，HTTP/1.1 バージョンで通信をしているといった具合だ．HTTP メソッドについては，後述する．

ソースコード 1: HTTP リクエスト

```
GET /index.html HTTP/1.1
Host: www.kansai-u.ac.jp
```

上述のリクエストに対する HTTP レスポンスが次のものになる．

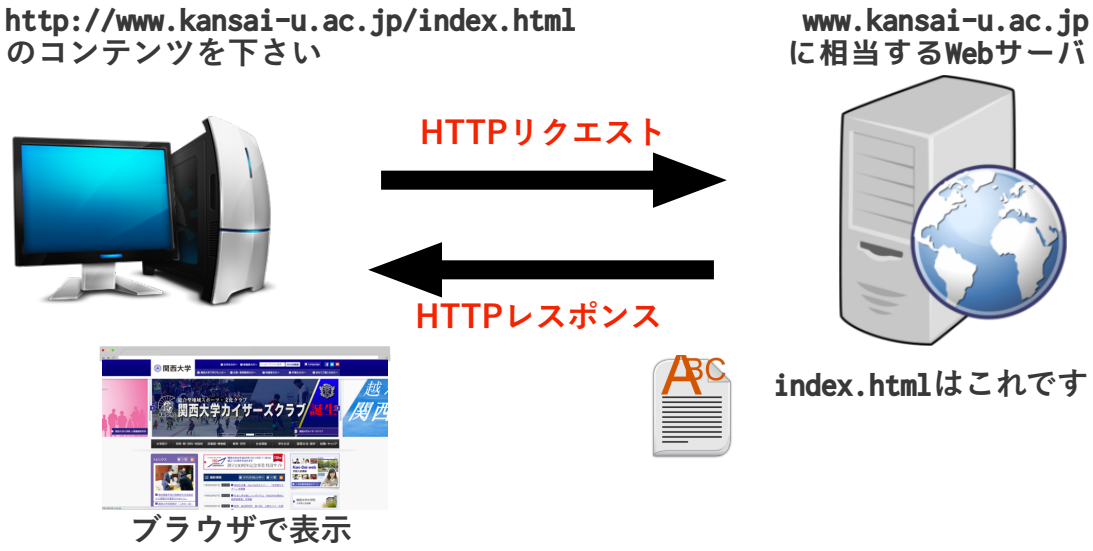

図 29: HTTP 通信の流れ

ソースコード 2: HTTP レスポンス

```
HTTP/1.1 200 OK
Date: Wed, 09 Mar 2016 16:50:39 GMT
Server: Apache
Accept-Ranges: bytes
Content-Length: 30569
Connection: close
Content-Type: text/html
Content-Language: ja

<index.htmlのHTMLコード>
```

10.1.2 HTTP ヘッダ

HTTP では，リクエストとレスポンス双方の通信に，ヘッダと呼ばれるものを付けて通信を行う．このヘッダを付けることによって，一つ一つの通信を詳細化することや区別すること，最適化することが可能となる．表 9 と表 10 に，主な HTTP ヘッダとその用途を示す．ここで挙げた以外にも様々なヘッダが用意されており，それらを用いることにより，効果的な HTTP 通信を実現している．例えば，HTTP 通信を減らすためにブラウザにコンテンツをキャッシュしたいとき，HTTP ヘッダを使って，ある程度の操作が出来る．

表 9: HTTP リクエストヘッダ一覧

ヘッダ	用途
Host	リクエスト先のサーバ名を示す
Cookie	クライアントの状態管理情報をサーバに送る
Accept	クライアントの受け入れ可能コンテンツタイプを返す
Authorization	クライアントの認証情報を返す

表 10: HTTP レスポンスヘッダ一覧

ヘッダ	用途
Server	Web サーバの情報を返す
WWW-Authenticate	認証領域名を示す固有値

10.1.3 HTTP メソッド

HTTP リクエストは，そのリクエストが何を意味しているかを示すために，メソッドと呼ばれる種類で分けられる．主なメソッドとして,「GET」と「POST」が挙げられる．通常,「GET」メソッドは，主に Web ページ等のコンテンツを取得するときに使われ,「POST」メソッドは，Web サーバにデータを送りたいときに使われる．この他,「PUT」や「DELETE」,「HEAD」が存在する．RESTful API 等を考えない限り，現状変わった挙動はない．

10.2 FTP

FTP とは，File Transfer Protocol の略で，ネットワークを用いたファイル転送のためのプロトコルである．主に，Web 用のコンテンツファイル（HTML ファイルや画像等）をサーバ上にアップロードするときや，フリーソフト等をクライアントがダウンロードしたいときなどに使われていた．FTP は，プロトコル自体に暗号化の仕組みが存在しないため，HTTP 同様に通信の盗聴が容易である．セキュアな通信を確立するため，FTP に暗号化通信の仕組みを組み込んだ FTPS や，後述する SSH という仕組みを用いて通信を暗号化する SFTP が存在する．

10.3 SSH

SSH とは，Secure Shell の略で，暗号化された通信上で，リモートコンピュータにログインするためのプロトコルである．全ての通信は，ログインのためのパスワード情報等も含めて暗号化されているため，盗聴されていたとしても内容が漏洩する心配はない．ログイン方法には，パスワードを使ったパスワード認証と公開鍵を用いた公開鍵認証があり，認証情報をよりセキュアに管理するには公開鍵認証を使ったほうが良い．

SSH は，他のサービスとも一緒に使われることがあり，上述した FTP を SSH 通信で実現する SFTP や，SFTP に似たものでリモートコンピュータ上に安全にファイルをコピーするための scp というものが存在する．

2016 年 3 月 10 日　初版　15M7102　赤坂 翔太
2016 年 3 月 10 日　初版　15M7112　高坂 賢佑
2016 年 3 月 10 日　初版　12-376　坂東 翼
2016 年 3 月 10 日　初版　12-377　平松 耕輔
2016 年 3 月 10 日　初版　12-378　平松 謙隆