

Web上の情報を代理取得するDNSサーバの試作

関根 利一¹ 前田 剛志¹ 山井 成良¹ 北川 直哉¹

概要: DNS は, Web や電子メール等の様々なインターネットサービスの利用前に, 通信相手が持つドメイン名の名前解決を行うために使用される機構である. したがって, DNS が使用できない環境では通信相手が特定できず, 各種サービスを利用することができなくなるため, 政府や ISP などの運用ポリシーによって様々なアクセス制限が行われる場合でも DNS サービスが完全に使用できない環境は存在しない. 本研究では, この特徴を利用し, 利用者の得たい情報を代理で取得し, 応答する特殊な DNS サーバを試作した. これにより, Web ページのアクセス制限が実施されている環境下においても任意の情報を取得することができる.

キーワード: DNS, Web

A prototype of a DNS server to acquire information on the web on behalf of users

RIICHI SEKINE¹ TAKESHI MAEDA¹ NARIYOSHI YAMAI¹ NAOYA KITAGAWA¹

Abstract: The DNS is a mechanism used to perform name resolution of a domain name before various communications such as e-mails and web. Therefore, a host cannot identify the communications partner in an environment of no access to DNS. For this reason, even if access restrictions are performed by operational policy of a government or a ISP, the environment that DNS was blocked off completely does not exist. In this paper, we have developed a specialized DNS server that acquires information on the web on behalf of users, and responds to him/her. This will enable a user to obtain any information under an environment of web access restrictions.

Keywords: DNS, web

1. はじめに

近年, ますます World Wide Web (以下, Web) は広がりを見せている. Web を利用することで, 様々な情報が取得でき, 便利になる一方で, 諸外国では特定の Web サイトへのアクセスや, 特定の単語を含む情報の閲覧を規制している例が存在する. 国際 NGO 団体である Freedom House の調査 [1] によると, 中東の国々や中国ではインターネットアクセスの自由度が低く規制が厳しいことが分かり, このような国においては普段, 我々が日本からであれば閲覧

可能な Web サイトでも, 規制が行われている可能性が高い. 実際に, 例えば中国では金盾と呼ばれるシステムにより, Google や Facebook, Twitter などのサイトを閲覧することができない. このような規制が行われている国や地域に訪問し, 閲覧が制限されているサイトへのアクセスが必要な場合, 予め VPN 等の準備が必要であり, 現状では非常に不便である.

また, 日本においてもインターネットサービスプロバイダ (以下, ISP) によって特定のサイトへのアクセス制限が行われている場合もある.

そこで, そのような規制を受けずに情報を取得する方法として, 通信相手のドメイン名の名前解決のために用いられる DNS を利用する方法を考案した. DNS は, Web や電

¹ 東京農工大学
Tokyo University of Agriculture and Technology, Koganei,
Tokyo 184-8588, Japan

子メール等のさまざまなインターネットサービスを使用する際に、通信相手のドメイン名から IP アドレスの情報の検索 (正引き) や、IP アドレスからドメイン名の情報の検索 (逆引き) を行うために、ドメイン名と IP アドレスを対応づけている仕組みである。もし、DNS が使用できないとなると、通信相手の特定ができず、Web や電子メール等のサービス自体が使用できなくなる。従って、特定の Web サイトへのアクセス規制が行われる環境でも DNS サービスが使用できなくなることはないため、アクセス規制が行われているサイトからでも、情報が取得できるのではないかと考えた。

本論文では、Web サイトに掲載された情報を DNS 問い合わせによって取得する一例として、天気と気温の情報を利用者に代わって取得し、応答する特殊な DNS サーバを試作した結果について報告する。

2. 関連研究

DNS サーバを本来の目的である名前解決以外の利用法で扱う研究として、DNS サーバの TXT レコードを、Web 上で発信されている情報をフィルタリングするために規定されるレーティング情報を受け渡す手段として利用する研究 [2] がされている。レーティングとは、Web 上の数多くの画像やテキストといった情報について、その内容が不適切かどうかを判別するために、Web 上の情報に対して、適切さの度合いを数値化しラベルづけすることをいう。現状、Web 上の情報をレーティングによりフィルタリングする方法として、PICS という HTTP を用いてレーティング情報のやり取りをする仕組みが一般的に用いられているが、この方式ではレーティング情報の取得がフィルタリングのオーバーヘッドとなる。そこで、PICS での TCP を用いる HTTP での通信ではなく、UDP による通信である DNS を用いることで、レーティング情報の取得の高速化を図っている。

また、DNS サーバを迷惑メール対策に利用する技術として DNSBL や SPF がある。DNSBL は DNS-based Black-hole List の略で、迷惑メールを送信、中継するホストの IP アドレスを収集し、リスト化したデータベースのことを指すが、このデータベースを参照する方法として DNS が用いられている。この技術は、受信したメールが迷惑メールかどうかの判別に用いられ、次の手順で行われる。メール受信サーバがメールを受信すると、メール受信サーバは送信されてきたメールの送信元 IP アドレスを、DNSBL サーバに DNS のプロトコルを用いて、データベースに含まれたものかどうか問い合わせを行う。含まれていなかった場合はそのまま受信を行うが、もしデータベースに含まれていた場合、メール受信サーバは受信を拒否する。このように DNSBL は迷惑メールの対策として利用されているが、この技術には問題点も存在する。DNSBL には様々な

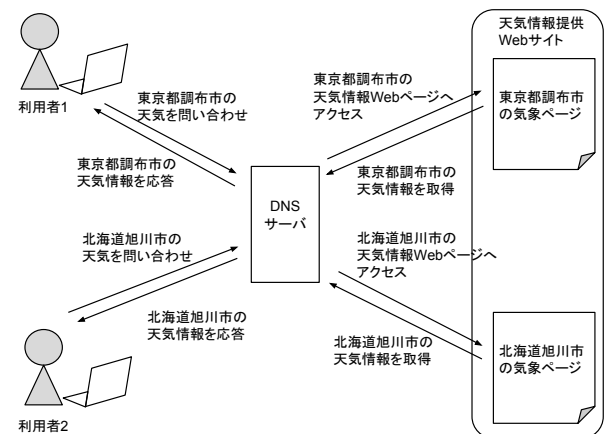


図 1 システムの全体像

Fig. 1 Overview of the system.

種類があり、それぞれ異なる運用ポリシーを定めているが、DNSBL の中には運用ポリシーが不明確で、リストの信頼性が怪しまれるものもある。よって、利用者はどの DNSBL を利用すべきか考慮する必要がある。

SPF は Sender Policy Framework の略であり、SMTP を利用したインターネット電子メールの送受信において送信者のドメインの偽称を防ぎ、正当性を検証する仕組み [3] である。SPF は、DNS サーバ上で SPF レコードに情報を定義することで実装ができる。SPF レコードではドメインを指定し、そのドメインの送信者がメールを送信する可能性のある IP アドレスのリストを記述する。SPF の動作は次の手順で行われる。メール受信サーバはメールを受信すると、送信者アドレスのドメイン部をもとに、DNS サーバへ SPF レコードで問い合わせを行う。問い合わせを受けた DNS サーバは SPF レコードに定義されている IP アドレスのリストを応答をする。送信元サーバが IP アドレスのリストに含まれている場合は受け取り、含まれていない場合は受信拒否をする。このように SPF を用いて、送信元アドレスを偽って送られる「なりすましメール」を防ぐことができる。

3. 設計

システムの全体像は図 1 で表される。はじめに、システムの利用者は任意の地域の天気と気温の情報を求めるために DNS サーバに問い合わせを行う。問い合わせを受けた DNS サーバは問い合わせの内容から、どこの地域の情報を求めているのか判別を行う。さらに、日本全国の各地域の天気情報を提供している Web サイトの中から、その地域のページにアクセスし、天気と気温の情報を取得する。そして、取得した情報を応答することで利用者は天気と気温の情報を得ることができる。

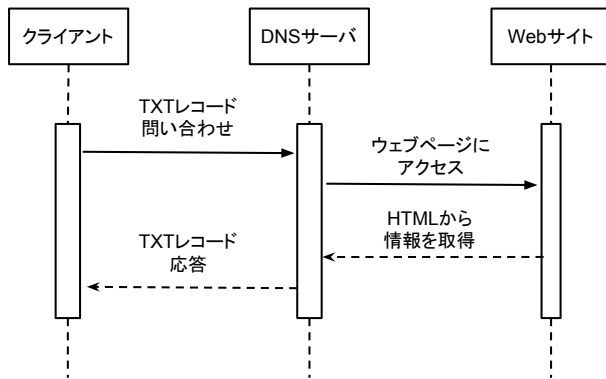


図 2 システムの動作
Fig. 2 Operation of the system.

3.1 DNS サーバの設計

DNS サーバにはネームサーバ名を定義する NS レコードや IP アドレスを定義する A レコード等の様々なリソースレコードが存在する。そのレコードのうちの一つとして TXT レコードと呼ばれるテキスト情報を埋め込めるレコードがあり、このレコードは特定の用途や書式が定められていないため、任意の情報を埋め込むことができる。本研究ではこの TXT レコードに注目し、応答されるテキスト情報部分に、DNS サーバが代理で取得した情報を埋め込むことで、利用者が求める情報を与えることができるのではないかと考えた。

DNS サーバが情報を取得し応答する動作は図 2 のように行われる。はじめに、利用者は DNS サーバへ情報を取得するために TXT レコード問い合わせを行う。次に、利用者から問い合わせを受けた DNS サーバは求められた情報のある Web ページにアクセスを行い、さらにそのページの HTML 解析をして、必要な情報のみを取り出す。最後に DNS サーバは取り出した情報を TXT レコードに埋め込み、応答を返す。このような動作によって、利用者に代わって情報を取得し、応答する DNS サーバを設計した。

3.2 DNS サーバの情報取得の方法

このシステムでは、必要な情報を Web より取得するため、情報を得る Web サイトを定義しなければならない。今回、天気や気温といった情報を得るために、世界中の天気の情報を紹介する「WorldWeatherOnline」という Web サイトから情報を得ることとした。この Web サイトの天気情報は各地域ごとにページが存在しており、それぞれのページでその地域の天気情報が詳しく紹介されている。例えば、東京都調布市の天気と気温が載っているページは「<http://www.worldweatheronline.com/Chofu-weather/Tokyo/Jp.aspx>」であり、北海道旭川市の天気と気温が載っているページは「<http://www.worldweatheronline.com/Asahikawa->

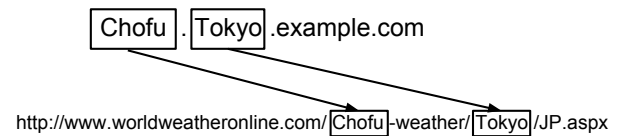


図 3 問い合わせの URL への変換
Fig. 3 Conversion from inquiry to URL.

weather/Hokkaido/Jp.aspx」となっている。そのため、ネームサーバに問い合わせをする際に、どの地域の情報が必要なのか場所を指定する必要がある。このシステムではドメイン名の前に指定する場所の名前を付けることで場所の指定を実現した。例えば、東京都調布市の天気と気温を求める場合の問い合わせには、図 3 のように問い合わせ先を指定する。ドメイン名の前に市町村名と都道府県名を付け加え、Chofu.Tokyo.example.com と問い合わせ先を指定することで、東京都調布市の天気と気温を求めることができる。

4. 実装

3 章で行った設計を基に、DNS サーバを実装をする。今回、perl 言語を用いて DNS サーバの試作を行った。プログラムは、CPAN 上にある DNS サーバとして使用される perl モジュールである Net::DNSServer[4]をもとに改良したものに加え、HTML の取得をする Perl モジュール LWP::UserAgent[5]、HTML の解析に HTML::TreeBuilder[6]を利用して作成した。表 1 に DNS サーバとして動くプログラム Base を示し、その動作を説明する。

表 1 Base プログラム
Table 1 Base program

```
1: my $resolver1 = new Net::DNSServer::Cache;
2: my $resolver2 = new MyTestResolver {
3:   dom => "example.com";
4:   run Net::DNSServer {
5:     priority => [$resolver2],
6:   }; # Never returns
```

このプログラムの内容を順に説明すると、

2-3 行目： dom 部分に情報を与える DNS サーバ名を埋め込み、後述する MyTestResolver と名前を付けた Perl モジュールで処理を行うことで、変数 resolver2 に問い合わせされた内容に対しての応答情報の値を格納する。

4-6 行目： Net::DNSServer モジュールによって、DNS サーバとして問い合わせがされた内容に対する応答を返す。その際、変数 resolver2 に格納された値が応答結果となる。

DNS サーバとして動くプログラムである Base は以上

のような動作を行う。次に、Base の中で用いられている MyTestResolver.pm の一部を表 2 に示す。このモジュールに各レコードの情報を書き込むことで、問い合わせされた内容を判別して、それに対応した応答結果を返す。本論文内では主にプログラム中の TXT レコード部分を示す。

表 2 MyTestResolver.pm
Table 2 MyTestResolver.pm

```
1: if ($question -> qname =~ /([a-z]+\.)?([a-z]+\.)?.$self
    ->{dom}/i && $question -> qtype eq "TXT"){
2:   my $response = bless \%{$dns_packet}, "Net::DNS
    ::Packet" || die "Could not initialize
    response packet";
3:   my $location1 = ucfirst(lc($1));
4:   my $location2 = ucfirst(lc($2));
5:   my $url = 'http://www.worldweatheronline.com/'
    . $location1 . '-weather/' . $location2 . '/JP.aspx';
6:   my $user_agent = "Mozilla/4.0 (compatible;
7:     MSIE 8.0; Windows NT 6.1; Trident/4.0)";
8:   my $ua = LWP::UserAgent
9:     ->new('agent' => $user_agent);
10:  my $res = $ua->get($url);
11:  my $content = $res->content;
12:  my $tree = HTML::TreeBuilder->new;
13:  $tree->parse($content);
14:
15:  my @temperature = $tree
16:    ->look_down('id','outlook_div_wxs')
17:    ->look_down('class','outlook_box1')
18:    ->find('li')
19:    ->as_text;
20:  $temperature[0] =~ tr/0-9\-\//cd;
21:  my $weather = $tree
22:    ->look_down('id','outlook_div_wxs')
23:    ->look_down('class','outlook_box1')
24:    ->find('img')
25:    ->attr('alt');
26:  $weather =~ tr/A-Za-z\-\//cd;
27:  my $ans = $weather . "/" . $temperature[0] . "degC";
28:
29:  $response->push("answer",
30:    Net::DNS::RR->new
31:      ("$self->{dom} 1000 TXT $ans"));
32:  $response->push("authority",
33:    Net::DNS::RR->new
34:      ("$self->{dom} 1000 NS $self->{dom}"));
35:  $response->push("additional",
36:    Net::DNS::RR->new
37:      ("$self->{dom} 1000 A 192.0.2.0"));
38:  my $response_header = $response->header;
39:  $response_header->aa(1); # Make Authoritative
40:  $response_header->qr(1); # Make Response
41:  return $response;
42: }
```

MyTestResolver3.pm について詳しく解説する。このモジュールは次のような順で動作を実行する。

1 行目 正規表現によって、問い合わせ方法が正しいかを判別する。問い合わせ先は、DNS サーバ名である example.com の前に地名を表す英文字列をドットを挟んで 2 つ指定することが条件となる。問い合わせ先が正しく、かつ TXT レコード要求をされた場合に以降の応答を行う。

3-4 行目 入力された英文字列を、「World Weather Online」の Web サイトの URL に対応するように、先頭だけ大文字、それ以降は小文字の形に整形を行う。

5 行目 整形を行った英文字列を URL の該当部分に結合させることにより、情報を取得する Web サイトを指定する。

6-13 行目 指定した Web サイトへのアクセスを行い、HTML を読み取る。

15-27 行目 Web サイトの HTML からタグを判別し、応答に必要な情報の部分を抽出する。

29-37 行目 DNS の応答を行う部分となり、answer 部分で、問い合わせに対する応答である天気と気温を返す。さらに authority 部分で権威 DNS サーバ名 (example.com) を返し、additional 部分で、その IP アドレス (192.0.2.0) を返す。

5. 結果

4 章で実装した DNS サーバとして動作する Perl モジュールを実行し、東京都調布市の天気と気温を求めるために、問い合わせ先を Chofu.Tokyo.example.com とし、TXT レコードで DNS 問い合わせを行った結果を表 3 に示す。

表 3 DNS の応答結果

Table 3 Response of DNS

```
[leach@localhost ~]$ nslookup -type=txt
Chofu.Tokyo.example.com 127.0.0.1
```

```
Server:      127.0.0.1
```

```
Address:     127.0.0.1#53
```

```
example.com  text = "partlyCloudy/5degC"
```

DNS サーバは TXT レコードで PartlyCloudy/5degC と応答した。また、情報の源となった Web サイトの World Weather Online の同時刻の天気情報も図 4 のように、天気が PartlyCloudy、気温が 13℃とあった。DNS サーバに TXT レコードで問い合わせをした結果と、World Weather Online の Web サイト情報を比較してみると、両者の情報は一致しており、正しい天気と気温の情報が取得できていることが分かる。

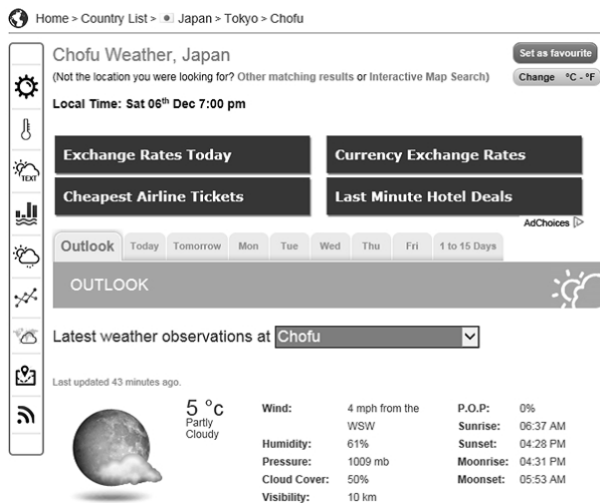


図 4 World Weather Online のサイト

Fig. 4 Web site of World Weather Online.

6. 考察

今回試作した DNS サーバの天気や気温といった情報以外にも、代理取得できる情報として、株価や外国為替レートといった情報が考えられる。これらの情報の取得には本試作で行った Web ページの定義の仕方に少量の変更を加えるだけで実現ができる。例えば、企業の株価情報を代理取得システムの場合、株式会社にそれぞれ付与される 4 桁の証券コードを問い合わせ先の中に入れることで、株式会社の指定や情報を取得する Web ページの指定ができる。また、外国為替レートの代理取得をする場合は、問い合わせ先に 2 つの通貨を指定することで、同様に Web ページの指定が行うことができる。さらに、本来の目的であるアクセス規制があるサイトの情報取得という目的を考えると、Twitter に書き込まれた内容の取得や、DNS 経由で Twitter に書き込むことができるように発展させるとより実用的になると思われる。

7. 今後の課題

今後の課題として、情報を取得する Web サイトが特定のサイトに限られる点が挙げられる。今回の試作で天気情報を取得する場合は、「World Weather Online」の Web サイトに限られ、ほかの天気情報提供サイトからは情報を取得できない。また、「World Weather Online」の Web サイト内においても、全国すべての市町村の気象ページが存在するわけではないため、ページが存在する地域の情報しか取得できないという問題がある。

加えて、DNS の応答では空白文字や日本語、特殊文字が使用できないことや、TXT レコードの 255 文字という文字数制限が存在するため、応答する情報が制限される問題もある。本来の Web サイトには、天気や気温のほかにも

様々な情報があるのだが、現状ではこのような制限により取得できる情報が限られている。より様々な情報を取得するために、情報の取得方法や文字列処理の方法、TXT レコードへの埋め込み方などについて工夫を行う必要がある。

8. おわりに

本論文では、利用者の得たい情報を代理で取得し、応答する DNS サーバの試作結果について報告した。DNS といえば名前解決、つまりドメイン名に対応する IP アドレスを取得するという仕組みが一般的であり、主な機能である。しかし、今回試作した DNS では名前解決とは別の利用方法を考えた。その際に問い合わせのレコードの種類を、IP アドレスを問い合わせる A レコードではなく、問い合わせに対する応答の自由度の高い TXT レコードによる応答を工夫することで、サービスを提供するシステムを製作することができた。

参考文献

- [1] Freedom House: Freedom on the net 2014(online), 入手先 [https://freedomhouse.org/sites/default/files/resources/FOTN 2014 Summary of Findings.pdf](https://freedomhouse.org/sites/default/files/resources/FOTN%202014%20Summary%20of%20Findings.pdf) (2015.03.23).
- [2] 椎名裕樹, 富永和人: DNS の TXT レコードを利用してレイティング情報を受け渡す WWW フィルタリングシステム, コンピュータソフトウェア, Vol.18, No.6, pp.2-14(2001).
- [3] 迷惑メール対策委員会: SPF (Sender Policy Framework) (online), 入手先 http://salt.iajapan.org/wpmu/anti_spam/admin/tech/explanation/spf/ (2015.03.23).
- [4] CPAN: Net::DNSServer (online), 入手先 <http://search.cpan.org/~bbb/Net-DNSServer/lib/Net/DNSServer.pm> (2015.03.23).
- [5] CPAN: LWP::UserAgent (online), 入手先 <http://search.cpan.org/dist/libwww-perl/lib/LWP/UserAgent.pm> (2015.03.23).
- [6] CPAN: HTML::TreeBuilder (online), 入手先 <http://search.cpan.org/~cjm/HTML-Tree/lib/HTML/TreeBuilder.pm> (2015.03.23).