DNS完全解説 の仕組み

第6回 DNSダイナミックアップデートによるゾーンデータの自動更新

馬場達也

前回は、セカンダリネームサーバの運用法とゾーン転送の仕組みについて解説した。今回は、クライアントからネームサーバ上のゾーンデータを動的に更新することが可能となる、DNSダイナミックアップデートについて解説する。

DNSダイナミックアップデートでゾーンデータ更新を効率化

新たなホストが追加されたり、ホストのIPアドレスが変更されたりした場合には、管理者がプライマリネームサーバ上のゾーンデータファイルを更新する。しかし、DHCPを使用してクライアントにIPアドレスを動的に割り当てているような環境では、ホストのIPアドレスが頻繁に変更されるため、管理者がゾーンデータを手動で管理することは不可能である。

このような場合に利用されるのが、ゾーンデータの自動更新を可能にする「DNSダイナミックアップデート」(最近では「ダイナミックDNS」や「DDNS」と呼ばれることも多い)である。DNSダイナミックアップデートの仕様はRFC 2136に記述されており、この機能を利用すると、IPアドレスを取得したクライアントや、クライアントにIPアドレスを割り当てたDHCPサーバが、自身の所属するゾーンを管理するプライマリネームサーバに対してゾーンデータの更新の要求を行うことが可能となる。更新の要求を受け取ったプライマリネームサーバは、要求メッセージに記述されている更新条件を満たしていれば、要求されたリソースレコードの削除や追加を行い、SOAレコードのシリアル番号をカウントアップさせる。もちろん、この更新内容はゾーン転送によってプライマリネームサーバからセカンダリネームサーバへと伝えられる。

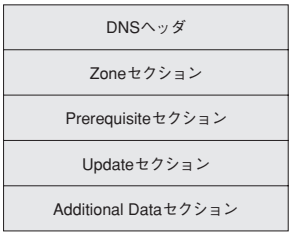


図1● DNSダイナミックアップデートで使用されるメッセージのフォーマットは通常時とは異なる

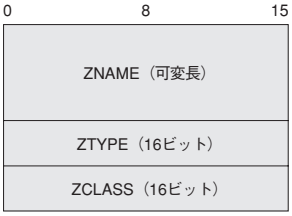


図2● Zoneセクションのフォーマット。Zoneセクションはメッセージ中に必ず1つ存在する

DNSダイナミックアップデートのクライアント機能は、Windows2000/XPや、DHCPサーバであるISC DHCP 3.0などに実装されている。また、サーバ機能は、BIND 8および9や、Windows2000 Serverに搭載されているMicrosoft DNS Serverなどに実装されている。

DNSダイナミックアップデートで使用されるメッセージ

DNSダイナミックアップデートでは、通常のDNS問い合わせ時とは少し異なるメッセージを使用する。DNSダイナミックアップデートで使用されるメッセージのフォーマットは図1のようになっている。

通常のDNS問い合わせやゾーン転送時に使用されるメッセージでは、DNSヘッダに含まれるOPCODE値は「0」となるが、DNSダイナミックアップデートで使用されるメッセージでは、OPCODE値が「5」となり、通常の問い合わせと区別される(ちなみに、前回紹介したDNS NOTIFYで使用されるメッセージでは、OPCODE値が「4」となる)。DNSヘッダ以降は、「Zoneセクション」「Prerequisiteセクション」「Updateセクション」「Additional Dataセクション」の4つのセクションから構成される。

[Zoneセクション]

Zoneセクションには、更新対象のゾーンのドメイン名が図2のフォーマットで記述される。DNSダイナミックアップデートのメッセージには、必ずZoneセクションが1つだけ存在する。

ZNAMEフィールドには、更新対象となるゾーンのドメイン名が入る。ZTYPEフィールドには必ず「6

（SOA）」が入り、ZCLASSフィールドには、更新対象となるゾーンのネットワーククラスの値（通常は「1（IN）」）が入る。

例えば「example.com」ゾーンのゾーンデータを更新する場合には、Zoneセクションの各フィールドの内容は次のようになる。

ZNAME　example.com.
ZTYPE　6（SOA）
ZCLASS　1（IN）

［Prerequisiteセクション］

Prerequisiteセクションにはリソースレコードを追加または削除するための条件を記述し、図3のフォーマットで条件が記述される。条件が存在しない場合にはPrerequisiteセクションは存在せず、条件が複数存在する場合にはこのセクションは複数存在する。

NAMEフィールドにはリソースレコードのキーとなるドメイン名が入り、TYPEフィールドにはレコードタイプの値（例えば、レコードタイプがAの場合は1、レコードタイプがCNAMEの場合は5）が入る。CLASSフィールドには、「指定したリソースレコードのセットが存在すること」を条件とする場合は、通常のネットワーククラスの値（通常は「1（IN）」）または「255（ANY）」が入り、「指定したリソースレコードのセットが存在しないこと」を条件とする場合には「254（NONE）」が入る。また、TTLフィールドには必ず「0」が入る。RDLENGTHフィールドにはRDATAフィールドの長さがバイト単位で入り、RDATAフィールドには、リソースレコードのデータ（例えば、Aレコードの場合はIPアドレス）が入る。

Prerequisiteセクションでは、次のような種類の条件を記述できる。

①指定したリソースレコード（データ指定なし）がすでに存在すること

指定した名前およびリソースレコードタイプを持つレコードが、その値に関係なく1つ以上存在すれば更新を行う。例えば、「sykes.example.comというホスト名に対応するAレコードがすでに存在する」という条件は次のように記述される。この場合、CLASSフィールドは「255（ANY）」となり、RDATAフィールドは空となる。

NAME　sykes.example.com.
TYPE　1（A）
CLASS　255（ANY）
TTL　0
RDLENGTH　0
RDATA　なし

この条件を満たさない場合は、サーバから「NXRRSET（存在することが条件のリソースレコードが存在しない）」というエラーが返される。

②指定したリソースレコード（データ指定あり）がすでに存在すること

指定した名前、リソースレコードタイプ、データを持つレコードが存在すれば更新を行う。例えば、「sykes.example.comというホスト名に対応するIPアドレスが192.168.0.10であるAレコードがすでに存在する」という条件は次のように記述される。

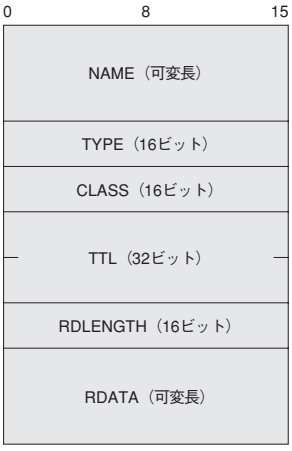


図3●　リソースレコードを追加・削除するための条件を記述するPrerequisiteセクションのフォーマット

NAME　sykes.example.com.
TYPE　1（A）
CLASS　1（IN）
TTL　0
RDLENGTH　4
RDATA　192.168.0.10

この条件を満たさない場合には、サーバから「NXRRSET（存在することが条件となっているリソースレコードが存在しない）」というエラーが返される。

③指定したリソースレコードが存在しないこと

指定した名前およびリソースレコードタイプを持つレコードが存在しなければ更新を行う。例えば、「sykes.example.comというホスト名に対応するAレコードが存在しない」という条件は、次のように記述される。この場合、CLASSフィールドは「254（NONE）」となり、RDATAフィールドは空となる。

NAME　sykes.example.com.
TYPE　1（A）
CLASS　254（NONE）
TTL　0
RDLENGTH　0
RDATA　なし

この条件を満たさない場合は、サーバから「YXRRSET（存在しないことが条件のリソースレコードが存在する）」というエラーが返される。

④指定した名前を持つリソースレコードが存在すること

指定した名前に対応するリソースレコードが1つ以上存在すれば更新を行う。例えば、「sykes.example.comというホスト名に対するリソースレコードがすでに存在する」という条件は次のように記述される。この場合、TYPEフィールドおよびCLASSフィールドは「255（ANY）」となり、RDATAフィールドは空となる。

NAME	sykes.example.com.
TYPE	255（ANY）
CLASS	255（ANY）
TTL	0
RDLENGTH	0
RDATA	なし

この条件を満たさない場合は、サーバから「NXDOMAIN（存在することが条件のドメイン名が存在しない）」というエラーが返される。

⑤指定した名前を持つリソースレコードが存在しないこと

指定した名前に対応するリソースレコードがまったく存在しなければ更新を行う。例えば、「sykes.example.comというホスト名に対するリソースレコードが存在しない」という条件は次のように記述される。この場合、TYPEフィールドは「255（ANY）」、CLASSフィールドは「254（NONE）」となり、RDATAフィールドは空となる。

NAME	sykes.example.com.
TYPE	255（ANY）
CLASS	254（NONE）
TTL	0
RDLENGTH	0
RDATA	なし

この条件を満たさない場合は、サーバから「YXDOMAIN（存在しないことが条件のドメイン名が存在する）」というエラーが返される。

［Updateセクション］

Updateセクションには、追加または削除するリソースレコードを記述する。Updateセクションのフォーマットは、Prerequisiteセクションと同様である。通常は、Updateセクションは1つ以上存在するが、このセクションが存在しない場合には更新が行われない。

例えば「sykes.example.com. 1200 IN A 192.168.0.30」というリソースレコードを追加したい場合は、各フィールドの内容は次のようになる。

NAME	sykes.example.com.
TYPE	1（A）
CLASS	1（IN）
TTL	1200
RDLENGTH	4
RDATA	192.168.0.30

また、逆にリソースレコードを削除したい場合は、TTLフィールドを「0」にする。例えば、すでに存在するAレコードを、その値（IPアドレス）に関係なく削除したい場合には、各フィールドの内容は次のようになる。

NAME	sykes.example.com.
TYPE	1（A）
CLASS	255（ANY）
TTL	0
RDLENGTH	0
RDATA	なし

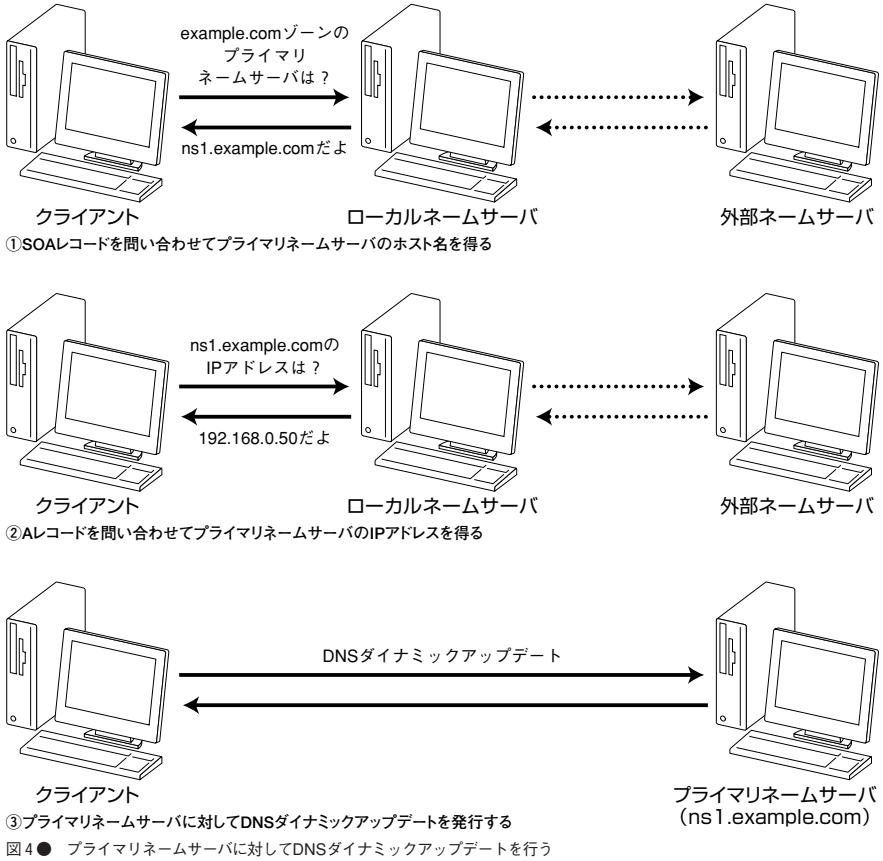


図4● プライマリネームサーバに対してDNSダイナミックアップデートを行う

［Additional Dataセクション］

Additional Dataセクションには、NSレコードを追加する際に必要となるグルーAレコードなどを記述する。Aレコードを追加する場合には、このセクションは存在しない。

DNSダイナミックアップデートをWindowsXPで確認

それでは、DNSダイナミックアップデートの動作を、実際のWindowsXPの動作を基にして説明しよう。クライアントは、まず、自身の所属するゾーンのプライマリネームサーバを探さなくてはならない。このため、最初に、通常のDNS問い合わせによって、自身の所属するゾーンのSOAレコードを問い合わせる（図4－①）。SOAレコードには、プライマリネームサーバのホスト名が記述されているので、クライアントは、次に、そのホスト名に対するAレコードを問い合わせ、プライマリネームサーバのIPアドレスを得る（図4－②）。そして、クライアントは、プライマリネームサーバに対してDNSダイナミックアップデートメッセージを発行する（図4－③）。

例えば、クライアントのホスト名が「sykes.example.com」で、新しく入手したIPアドレスが「192.168.0.30」の場合は、最初にリスト1のようなDNSダイナミックアップデートメッセージを発行する。

このメッセージでは、「登録しようとするホスト名に対応するCNAMEレコードが存在しない」という条件と、「登録しようとするホスト名およびIPアドレスを記述したAレコードがすでに存在している」という条件を設定している。このメッセージを受信したプライマリネームサーバから「YXRRSET」（存在しないことが条件のリソースレコードが存在する）というエラーが返ってきた場合は、登録しようとするホスト名に対応するCNAMEレコードが存在していることになる。CNAMEレコードが存在する場合は、同じ名前でAレコードを登録することができないため、更新に失敗し、ここで終了する。プライマリネームサーバから「NOERROR」（エラーなし）で返ってきた場合は、登録しようとしているAレコードがすでに存在することになるので、更新を行う必要はない。このメッセージでは、Updateセクションが存在しないので、プライマリネームサーバ上のゾーンデータは変更されない。

また、プライマリネームサーバから「NXRRSET」（存在することが条件のリソースレコードのセットが存在しない）というエラーが返ってきた場合には、登録しようとしているAレコードが存在しないことになるので、クライアントはこれに続いてリスト2のメッセージを発行する。

```
［Zoneセクション］
ZNAME      example.com.
ZTYPE      6（SOA）
ZCLASS     1（IN）

［Prerequisiteセクション1］ ← 「ホスト名に対応するCNAMEレコードが存在しない」ことを条件とする

NAME       sykes.example.com.
TYPE       5（CNAME）
CLASS      254（NONE）
TTL        0
RDLENGTH   0
RDATA      なし

［Prerequisiteセクション2］ ← 「指定したアドレスのAレコードがすでに存在する」ことを条件とする

NAME       sykes.example.com.
TYPE       1（A）
CLASS      1（IN）
TTL        0
RDLENGTH   4
RDATA      192.168.0.30
```

リスト1● DNSダイナミックアップデートメッセージ1（WindowsXP）

リスト2のメッセージでは、「登録しようとするホスト名に対応するCNAMEレコードが存在しない」ことと、「登録しようとするホスト名に対応するAレコードが存在しない」ことを条件に設定している。このメッセージを受信したプライマリネームサーバから、「NOERROR」（エラーなし）が返ってくれば、Updateセクションで記述したAレコードが無事に登録されたことになる。「YXRRSET」（存在しないことが条件のリソースレコードが存在する）というエラーが返ってきた場合は、別のIPアドレスが記述されたAレコードが存在していることになるので、続けてリスト3のメッセージを発行し、すでに存在するAレコードの削除を行ってから、目的のAレコードを登録する。

逆に、クライアントが、使用していたIPアドレスを解放する場合は、DNSダイナミックアップデートによって、該当するAレコードの削除を行う。この場合は、クライアントが、プライマリネームサーバに対して、リスト4のようなメッセージを発行する。

ここでは、正引きゾーンの場合を紹介したが、この動作を、逆引きゾーンに対しても同様に行う。

```
[Zoneセクション]
ZNAME       example.com.
ZTYPE       6 (SOA)
ZCLASS      1 (IN)
[Prerequisiteセクション1] ← 「ホスト名に対応するCNAMEレコードが存在しない」ことを条件とする
NAME        sykes.example.com.
TYPE        5 (CNAME)
CLASS       254 (NONE)
TTL         0
RDLENGTH    0
RDATA       なし
[Prerequisiteセクション2] ← 「ホスト名に対応するAレコードが存在しない」ことを条件とする
NAME        sykes.example.com.
TYPE        1 (A)
CLASS       254 (NONE)
TTL         0
RDLENGTH    0
RDATA       なし
[Updateセクション1] ← 上記の条件が満たされた場合に登録するAレコード
NAME        sykes.example.com.
TYPE        1 (A)
CLASS       1 (IN)
TTL         1200
RDLENGTH    4
RDATA       192.168.0.30
```

リスト2● DNSダイナミックアップデートメッセージ2（WindowsXP）

```
[Zoneセクション]
ZNAME       example.com.
ZTYPE       6 (SOA)
ZCLASS      1 (IN)
[Prerequisiteセクション1] ← 「ホスト名に対応するCNAMEレコードが存在しない」ことを条件とする
NAME        sykes.example.com.
TYPE        5 (CNAME)
CLASS       254 (NONE)
TTL         0
RDLENGTH    0
RDATA       なし
[Updateセクション1] ← すでに存在するAレコードを削除
NAME        sykes.example.com.
TYPE        1 (A)
CLASS       255 (ANY)
TTL         0
RDLENGTH    0
RDATA       なし
[Updateセクション2] ← 目的のAレコードを登録
NAME        sykes.example.com.
TYPE        1 (A)
CLASS       1 (IN)
TTL         1200
RDLENGTH    4
RDATA       192.168.0.30
```

リスト3● DNSダイナミックアップデートメッセージ3（WindowsXP）

DNSダイナミックアップデートの利用には注意点も

DHCP環境では、クライアントのIPアドレスが頻繁に変更されるため、プライマリネームサーバの持つゾーンデータだけでなく、セカンダリネームサーバの持つゾーンデータも迅速に更新しなければならない。このためには、DNSダイナミックアップデートに加えて、前回説明したDNS NOTIFYを使用するのがよい。DNS NOTIFYを使用すると、DNSダイナミックアップデートによってプライマリネームサーバのゾーンデータが更新されると同時に、セカンダリネームサーバに更新が通知されるため、ゾーンデータの変更がすぐに反映されるのである。

また、DNSダイナミックアップデートでリソースレコードを登録する場合には、TTLを小さめに設定する。これは、古いデータがキャッシュに残ってしまうのを防ぐためである。WindowsXPでは、TTLは自動的に1,200秒（20分）に設定される。

BINDでDNSダイナミックアップデートを受け付けるようにするためには、BINDの設定ファイルである「named.conf」ファイルのzoneステートメントでリスト5のように設定すればよい。この例では、「192.168.0.0/24」というネットワークに所属するすべてのホストからのDNSダイナミックアップデートを許可している。しかし、可能であれば、クライアントではなくDHCPサーバがDNSダイナミックアップデートを行うように設定し、ネームサーバ側ではDHCPサーバからのみDNSダイナミックアップデートを受け付ける設定にしたほうが安全である。

```
[Zoneセクション]
ZNAME       example.com.
ZTYPE       6 (SOA)
ZCLASS      1 (IN)

[Updateセクション1] ← 削除するAレコードを記述

NAME        sykes.example.com.
TYPE        1 (A)
CLASS       1 (IN)
TTL         0 ← 削除する場合は、TTLを「0」とする
RDLENGTH    4
RDATA       192.168.0.30
```

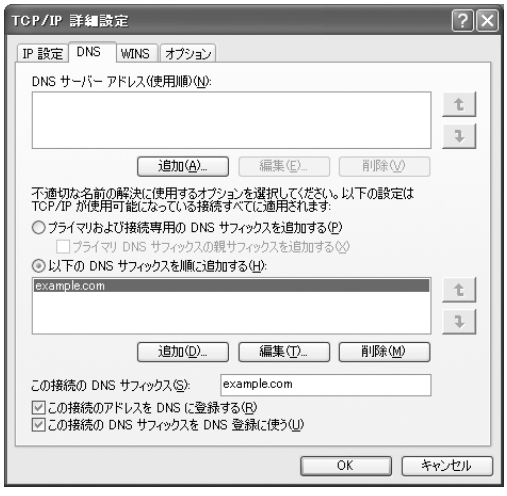
リスト4● 削除時のDNSダイナミックアップデートメッセージ(WindowsXP)

```
// 正引きゾーン
zone "example.com" {
        type master;
        file  "example.com.zone";
        allow-update { 192.168.0.0/24; }; ← この行を追加する
};
// 逆引きゾーン
zone "0.168.192.in-addr.arpa" {
        type master;
        file  "0.168.192.in-addr.arpa.zone";
        allow-update { 192.168.0.0/24; }; ← この行を追加する
};
```

リスト5● BIND 8および9でのDNSダイナミックアップデートの設定(named.confファイル)

BINDでは、DNSダイナミックアップデートによって更新されたデータは、ゾーンデータファイルに直ちには反映されず、ジャーナルファイルという別のファイルに差分情報が保存される。このため、DNSダイナミックアップデートを受け付けるようにしたゾーンのゾーンデータファイルは、直接書き換えてはならない。もし、ゾーンデータを手動で更新したい場合は、nsupdateという、BINDに付属するDNSダイナミックアップデートプログラムを使用してゾーンデータを更新するようにする。

Windows2000/XPでDNSダイナミックアップデートのクライアント機能を有効にするためには、該当する接続の「インターネットプロトコル(TCP/IP)のプロパティ」を開き、「詳細設定」ボタンをクリックすると開く「TCP/IP詳細設定」の「DNS」の設定画面で、「この接続のアドレスをDNSに登録する」と「この接続のDNSサフィックスをDNS登録に使う」をチェックすればよい(画面1)。



画面1● WindowsXPでのDNSダイナミックアップデートの設定は「DNS」タブを選択して行う

リモート操作の危険性に配慮したセキュアDNSダイナミックアップデート

これまでに、DNSダイナミックアップデートの動作について説明してきた。しかし、DNSダイナミックアップデートは便利である反面、非常に危険な面も持っている。それは、リモートからゾーンデータの変更ができるようになるため、悪意を持った者が意図的にゾーンデータを改ざんすることが可能になる点である。

ネームサーバ側では、IPアドレスによって、DNSダイナミックアップデートを許可するクライアントを制限できる。しかし、DNSメッセージの送信元IPアドレスの偽造は容易であり、IPアドレスでの制限はあまり効果がない。このため、DNSダイナミックアップデートでは、TSIG(Transaction Signature)やSIG(0)といった、暗号技術によって相手を認証する仕組みを合わせて使用することが推奨されている(RFC 3007)。次回は、このTSIGとSIG(0)について詳しく説明する。

NTTデータ 馬場達也

●今回の内容に関連するRFC

RFC 2136 "Dynamic Updates in the Domain Name System (DNS UPDATE)"
RFC 3007 "Secure Domain Name System (DNS) Dynamic Update"