DNS完全解説
の仕組み
第5回 セカンダリネームサーバの運用とゾーン転送の仕組み

馬場達也

前回は、メール配送におけるDNSの役割とMXレコードの記述法、そして、ある特定のサービスを提供するサーバを探索するために使用されるSRVレコードについて紹介した。今回は、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紹介しよう。

ネームサーバを複数設置してDNSサービスを確保する

名前解決やメール配送先の検索を行うDNSは、インターネットの重要なインフラとなっているため、ネームサーバがダウンしてしまうことで、DNSの機能が停止してしまう事態は避けなければならない。このため、1台のネームサーバがダウンした場合でも、別のネームサーバによってサービスが提供できるように、ネームサーバは複数台設置するようにする。

しかし、ゾーンデータファイルを複数のネームサーバ上で別々に管理すると、ゾーンデータの不整合が発生する可能性があるため、ゾーンデータファイルの原本は1台のネームサーバ上で管理し、ほかのネームサーバは、その原本を管理しているネームサーバからネットワーク経由でゾーンデータをコピーして管理する。ここで、ネットワーク経由でゾーンデータを転送することを「ゾーン転送」と呼び、ゾーンデータファイルの原本を管理するネームサーバを「プライマリネームサーバ」(「プライマリマスタ」とも言う)、プライマリネームサーバからゾーン転送によってゾーンデータを取得し、そのコピーを管理するネームサーバを「セカンダリネームサーバ」(「セカンダリマスタ」または「スレーブ」とも言う)と呼ぶ。

プライマリネームサーバとセカンダリネームサーバの違いは、原本をどこで管理しているかの違いだけであって、クライアントやほかのゾーンを管理するネームサーバは、その違いを意識する必要はない。

セカンダリネームサーバを設置するうえでの注意点

セカンダリネームサーバを設置すると、次のようなメリットがある。

- 1台がダウンした場合でもサービスを継続できる
- 名前解決の問い合わせの負荷を分散できる

このため、セカンダリネームサーバを多く設置したほうがメリットは大きくなる。しかし、セカンダリネームサーバの設置場所には気を付ける必要がある。例えば、用意したセカンダリネームサーバをすべてプライマリネームサーバと同じセグメントに設置してはならない。これは、ネットワーク障害や電源の障害によって、そのセグメント全体が機能しなくなり、セカンダリネームサーバも含めて、すべてのネームサーバに対してアクセスできなくなる可能性があるからである。このため、ネームサーバのうち、最低でも1台は離れた別のネットワークに設置し、ネットワークの障害や電源の障害が発生した場合でも、すべてのネームサーバへの到達性が失われるようなことは避けなければならない。

セカンダリネームサーバの運用について記述されているRFC 2182では、通常の規模の組織の場合は、ネームサーバを3台用意し、そのうちの1台は、ほかのネームサーバから十分に離れた場所に設置するのがよいとされている。例えば、図1のように、同じような規模の別の組織との間で、相手の管理するゾーンのセカンダリネームサーバをお互いに運用し合うようにしたり、契約しているISPにセカンダリネームサーバを運用してもらったりするのがよいだろう。

ゾーン転送に使われるSOAレコードを理解する

ゾーン転送を行う際に重要となるパラメータが、SOA

レコードに記述されている「Serial（シリアル番号）」「Refresh（リフレッシュ間隔）」「Retry（リトライ間隔）」「Expire（ゾーンデータの有効期間）」である（リスト1）。シリアル番号は、セカンダリネームサーバとの間でゾーン転送を行う場合に、ゾーンデータの内容が更新されているのかどうかをセカンダリネームサーバが判断するために使用する。つまり、ゾーンデータの内容を更新した場合には、このシリアル番号を必ず増やすようにする（シリアル番号を小さい値に戻す方法についてはコラムを参照）。リフレッシュ間隔は、セカンダリネームサーバがゾーン転送を行うためにプライマリネームサーバのゾーンデータのシリアル番号をチェックする間隔である。リトライ間隔は、セカンダリネームサーバが何らかの原因でゾーンデータのチェックに失敗した場合に行うリトライの間隔である。ゾーンデータの有効期間は、セカンダリネームサーバが、取得したゾーンデータを利用できる期間だ。

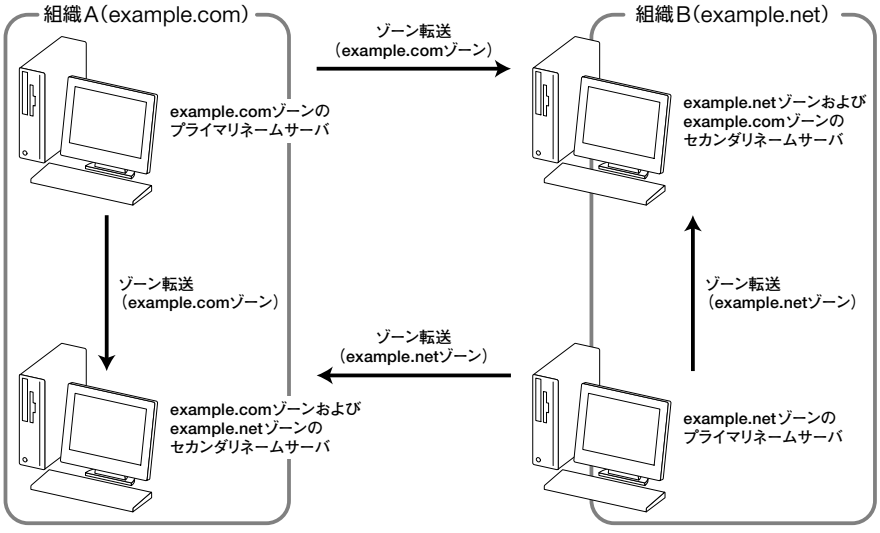

図1● セカンダリネームサーバの設置例。名前解決が途切れないようにネームサーバを配置する必要がある

```
$TTL 86400
@  IN  SOA  ns1.example.com.  hostmaster.example.com. (
       2002101800 ; シリアル番号
       28800      ; リフレッシュ間隔（秒）
       7200       ; リトライ間隔（秒）
       604800     ; ゾーンデータの有効期間（秒）
       3600       ; ネガティブキャッシュの有効期間（秒）
       )
  IN  NS  ns1.example.com.   ; このゾーンのプライマリマスタ
  IN  NS  ns2.example.com.   ; このゾーンのセカンダリマスタ
  IN  NS  ns.example.net.    ; このゾーンのセカンダリマスタ
ns1  IN  A  192.168.0.10
ns2  IN  A  192.168.0.11
```

リスト1● ゾーンデータファイルの記述例

また、ゾーンデータファイルには、リスト1のように、そのゾーンを管理するすべてのネームサーバのホスト名をNSレコードで記述しておく。そして、同時にそのネームサーバのIPアドレスもAレコードで記述しておくが、もし、そのネームサーバがほかのゾーンに属しているのであれば（リスト1の例では、「ns.example.net」が該当する）、そのAレコードは別のゾーンに記述するので、このゾーンデータ中では記述しない。

セカンダリネームサーバを初めて起動した場合にはゾーンデータがまだ自ホストに存在しないので、セカンダリネームサーバはすぐにプライマリネームサーバに対してゾーン転送のリクエストを行い、ゾーンデータを取得する。このゾーン転送のリクエストは、取得したいゾーンのドメイン名と、「AXFR」というレコードタイプを指定したDNSの問い合わせパケットをプライマリネームサーバに送信することで行う。また、通常の名前解決の問い合わせでは、UDPを使用してDNSのパケットをやり取りするが、UDPでは、最大512バイトのデータしか送ることができないという制限があるため、巨大なデータを送信する必要のあるゾーン転送では、UDPの代わりにTCPを使用する。

そして、セカンダリネームサーバは、取得したゾー

Column シリアル番号を小さな値に戻す方法

シリアル番号を誤って大きな値にしてしまい、その値がセカンダリネームサーバに伝播してしまった場合はどうすればよいだろうか？ シリアル番号をあわてて元に戻したところで、セカンダリネームサーバ上のシリアル番号は元に戻らない。このような場合に、シリアル番号を元の小さい値に戻す方法がRFC 1912に記述されている。

まず、まちがったシリアル番号に2,147,483,647を足した値を新しいシリアル番号とする。ただし、その結果が4,294,967,296よりも大きくなる場合には、4,294,967,296を引いた値を新しいシリアル番号とする。これにより、セカンダリネームサーバは、新しいシリアル番号を自分のものよりも新しいと認識してゾーン転送を行う。ここで、現在のシリアル番号が本来のシリアル番号よりも小さい値である場合には、すべてのセカンダリネームサーバがゾーン転送を完了したところで、本来のシリアル番号に上げればよい。もし、現在のシリアル番号が本来のシリアル番号よりも大きい場合は、本来のシリアル番号よりも小さくなるまで、この作業を繰り返せばよい。

ンデータのSOAレコードに記述されているリフレッシュ間隔ごとに、プライマリネームサーバに対してSOAレコードの問い合わせを行う（図2の(a)）。ここで取得したSOAレコードのシリアル番号が、すでに持っているゾーンデータのSOAレコードのシリアル番号よりも大きい場合は、プライマリネームサーバ上のゾーンデータが更新されていることになるので、プライマリネームサーバに対して、再びゾーン転送のリクエストを行う（図2の(b)）。もちろん、取得したSOAレコードのシリアル番号が、すでに持っているゾーンデータのSOAレコードのシリアル番号と同じであれば、ゾーンデータは更新されていないので、ゾーン転送のリクエストは行わない。

プライマリネームサーバがダウンしているなどの理由で応答しなかった場合には、セカンダリネームサーバは、SOAレコードに記述されているリトライ間隔ぶんだけ待ち、再度プライマリネームサーバに対して問い合わせを行う。何度リトライしてもプライマリネームサーバからの応答がなく、ゾーンデータを取得してからSOAレコードに記述されているゾーンデータの有効期間が過ぎてしまった場合には、セカンダリネームサーバは、そのゾーンデータを破棄しなければならない。この場合には、セカンダリネームサーバは、そのゾーンに対する名前解決サービスを提供することができなくなる。

AXFRによるゾーン転送は、リスト2のようにdigコマンドを使用して行うことも可能である。

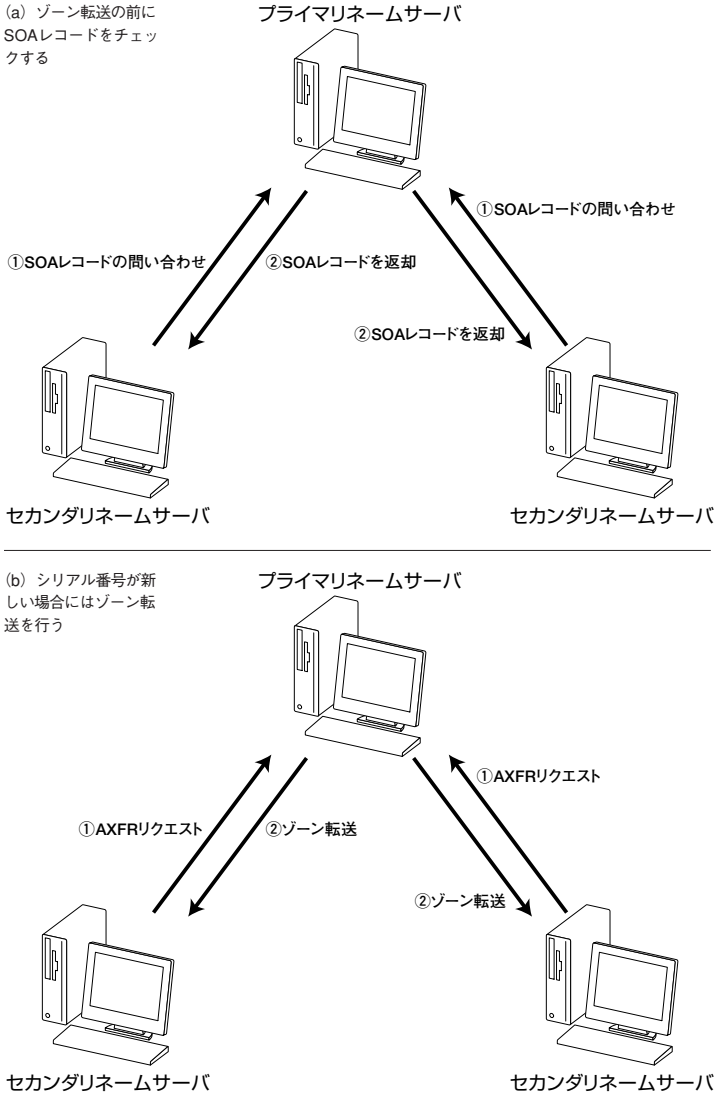

図2● AXFRリクエストによるゾーン転送の手順

ゾーンの変更を知らせるDNS NOTIFY

これまで説明したゾーン転送の仕組みでは、セカンダリネームサーバは、SOAレコードのリフレッシュ間隔で指定された時間だけ待たないとゾーン転送を試みようとしない。つまり、ゾーンデータファイルを更新しても、その内容がセカンダリネームサーバに伝わるまでには、最悪の場合、リフレッシュ間隔で指定された時間だけ待たなければならないわけだ。これを解決するための機能として、ゾーンデータが更新された場合に、プライマリネームサーバからセカンダリネームサーバに対して更新が行われたことを知らせる「DNS NOTIFY」という機能がある（詳しくはRFC 1996を参照していただきたい）。この機能を有効にすると、プライマリネームサーバ上のゾーンデータが更新された場合、プライマリネームサーバがセカンダリネームサーバに対して「NOTIFYリクエスト」という特別なDNSパケットを送信する。NOTIFYリクエストには、更新されたゾーンのドメイン名が記述されているので、NOTIFYリクエストを受信したセカンダリネームサーバは、更新されたゾーンを認識し、SOAレコードのリフレッシュ間隔を無視して、すぐにゾーン転送のリクエストを行うのである。

DNS NOTIFYは、バージョン8.2.3以降のBINDや、Windows2000 Serverに搭載されているMicrosoft DNS Serverでも実装されている。そこで、ここでは、BIND 9の動作を基にDNS NOTIFYの仕組みを説明しよう。まず、ゾーンデータが更新されると、プライマリネームサーバは、NOTIFYリクエストを送信する先のネームサーバのホスト名をゾーンデータ中のNSレコードから取得する。このNSレコードに記述されているネームサーバの中には、プライマリネームサー

バも含まれているため、これを削除し（プライマリネームサーバのホスト名はSOAレコードに記述されている）、セカンダリネームサーバのみに対してNOTIFYリクエストを送信する（図3）。そして、NOTIFYリクエストを受信したセカンダリネームサーバは、NOTIFYリクエストを受信したことを知らせるための「NOTIFYレスポンス」を返し、先に説明した方法で、ゾーン転送のリクエストを行うのである。

これで完了だと思われるかもしれないが、実は、BINDではプライマリネームサーバだけでなくセカンダリネームサーバもNOTIFYリクエストを送信する。つまり、プライマリネームサーバからのNOTIFYリクエストによってゾーン転送を行い、自身のゾーンデータが更新されると、各セカンダリネームサーバは、ほかのセカンダリネームサーバに対してNOTIFYリクエストを送信するのである。ただし、NOTIFYリクエストを受信した別のセカンダリネームサーバは、NOTIFYリクエストの送信元が、ゾーン転送のリクエストを行う先のネームサーバでなければ、このNOTIFYリクエストを無視する。では、なぜセカンダリネームサーバが別のセカンダリネームサーバに対してNOTIFYリクエストを送信する必要があるのだろうか。これは、図4のように、ほかのセカンダリネームサーバが、ファイアウォールを隔てているなどの理由で、プライマリネームサーバと直接通信できない場合に、ゾーン転送をほかのセカンダリネームサーバから行っている場合があるためだ。どのネームサーバが自分との間でゾーン転送を行うのかを知ることができれば、そのネームサーバにだけNOTIFYリクエストを送信すればよいが、NOTIFYリクエストを送信するネームサーバは、どのネームサーバが自身をゾーン転送元として設定しているのかを知ることができないため、すべてのネームサーバに対して、NOTIFYリクエストを送信するようにしているのである。

```
$ dig  @192.168.0.10  example.com.  axfr
         ↑              ↑            ↑
   プライマリネームサーバのIPアドレス
                  取得するゾーンのドメイン名
                               レコードタイプとしてAXFRを指定
```

リスト2● digコマンドを使ったゾーン転送の例

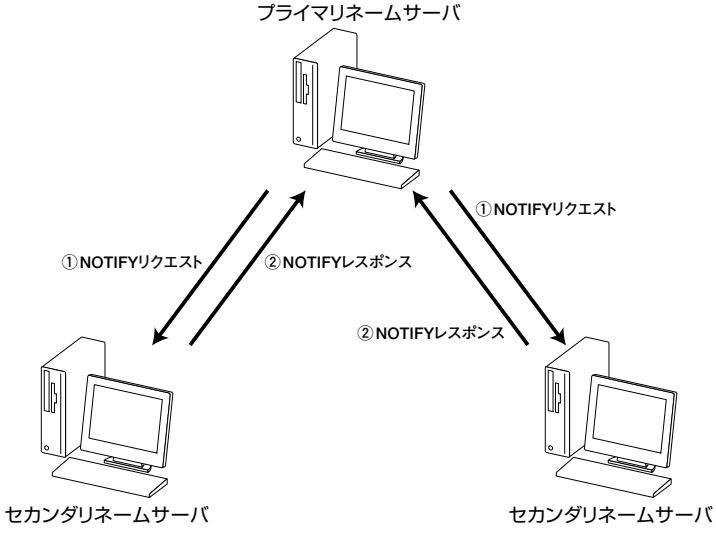

図3● NOTIFYリクエストによってゾーンデータ更新が通知される

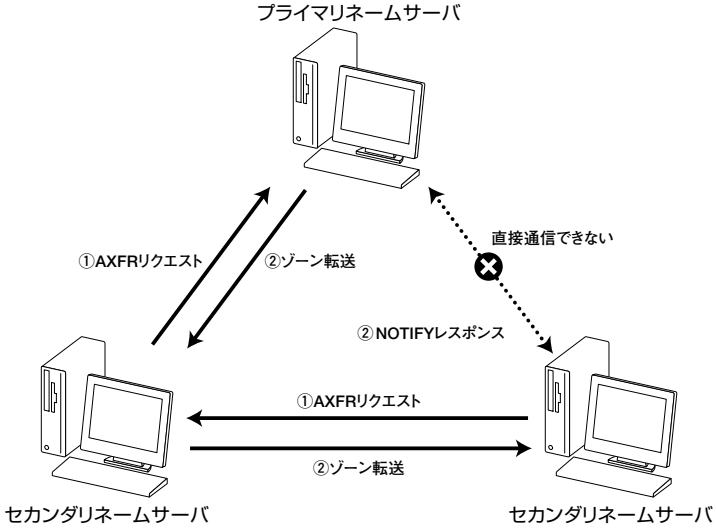

図4● セカンダリネームサーバからゾーン転送のリクエストを行う。セカンダリネームサーバがプライマリネームサーバと直接通信できない環境では、別のセカンダリネームサーバからゾーン転送を行っている場合もある

差分ゾーン転送でネットワーク負荷を軽減

これまでに説明したAXFRによるゾーン転送では、ゾーンデータの内容が1レコードだけ変更された場合でも、ゾーンデータ全体を転送するようになっている。しかし、特に、ゾーンデータが頻繁に更新される環境で、DNS NOTIFY機能を有効にしている場合は、巨大なゾーンデータを頻繁に転送することになり、ネットワークに非常に負荷がかかることになる。このため、ゾーンデータ全体を転送せず、ゾーンデータの変更された部分（差分データ）だけを送信する、差分ゾーン転送（IXFR：Incremental Zone Transfer）という機能がある（詳しくはRFC 1995を参照していただきたい）。

差分ゾーン転送のリクエストは、レコードタイプとして「AXFR」の代わりに「IXFR」を指定し、UDP

を使用して行われる（ただし、プライマリネームサーバから転送する差分データが512バイトを超えた場合は、TCPで再度リクエストする必要がある）。また、この差分ゾーン転送のリクエストには、セカンダリネームサーバが保持するゾーンデータのバージョンをプライマリネームサーバに知らせるために、セカンダリネームサーバが保持しているSOAレコードの内容が含まれる。このリクエストを受信したプライマリネームサーバは、セカンダリネームサーバから送信されたSOAレコードのシリアル番号を確認することで、プライマリネームサーバが保持する現在のゾーンの内容との差分を取り出し、その差分だけをセカンダリネームサーバに送信するのである（図5）。

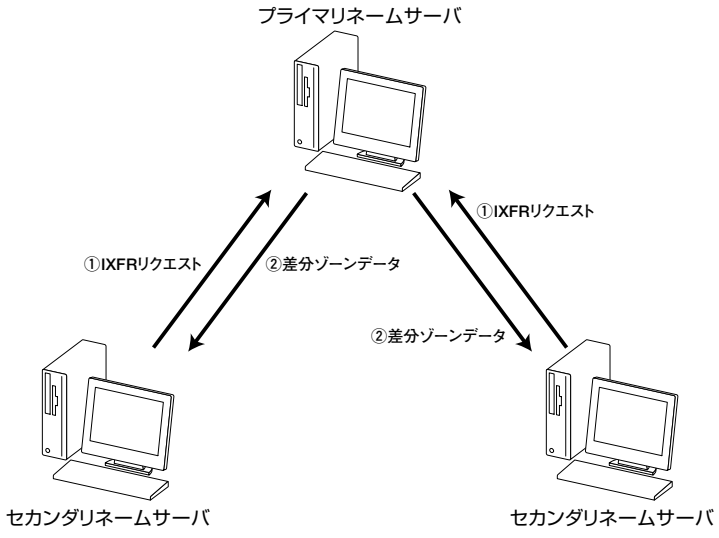

図5● 差分ゾーン転送。効率的にDNS情報の転送を行える

プライマリネームサーバでは、過去のゾーンデータと現在のゾーンデータとの差分情報を保持する必要があるが、AXFRによるゾーン転送で必要であった、シリアル番号をチェックするためのSOAレコードの問い合わせが不要になり、また、巨大なゾーン全体を転送する必要もなくなるため、ネットワークの負荷を減少させることができる。ただし、セカンダリネームサーバのゾーンデータが古すぎるなどの理由でプライマリネームサーバ上に差分データが存在しない場合には、AXFRによるゾーン転送と同じようにゾーン全体が転送される。

例えば、リスト3のようにゾーンデータが更新されたとしよう。すると、図6のように、AXFRによるゾーン転送では、すべてのゾーンデータが転送されるが、IXFRによるゾーン転送では、差分データ（削除する古いレコードと追加する新しいレコード）だけが転送される（転送されるゾーンデータの最初と最後に同じSOAレコードが挿入されているが、これは、ゾーン転送の開始と終了がわかるようにするためである）。

BIND 9では、デフォルトで差分ゾーン転送が有効になっている。しかし、プライマリネームサーバでは、ダイナミックアップデートによって更新されたデータしか差分データとして保持しない（ダイナミックアップデートについては、次回詳しく説明する）。このため、差分ゾーン転送を有効に機能させるためには、ゾーンデータファイルを手動で書き換えずに、nsupdateなどのダイナミックアップデートを行うツールを用いて更新する必要がある。

また、BIND 9において、差分ゾーン転送のリクエストを行わないようにするためには、named.confファイルでリス

（更新前）

```
$ORIGIN example.com.
$TTL 86400
@    IN   SOA   ns1.example.com.  hostmaster.example.com. (
                          2002091800 8800 7200 604800 3600)
     IN   NS   ns1
     IN   NS   ns2
ns1  IN   A    192.168.0.10
ns2  IN   A    192.168.0.11
foo  IN   A    192.168.0.20
bar  IN   A    192.168.0.21
```

（更新後）

```
$ORIGIN example.com.
$TTL 86400
@    IN   SOA   ns1.example.com.  hostmaster.example.com. (
                          2002101800 8800 7200 604800 3600)
     IN   NS    ns1
     IN   NS    ns2
ns1  IN   A     192.168.0.10
ns2  IN   A     192.168.0.11
foo  IN   A     192.168.0.30  ← 変更されたレコード
bar  IN   A     192.168.0.21
```

リスト3● 更新されたゾーンデータの例

(a)AXFRによるゾーン転送で転送するゾーンデータ

```
example.com. IN SOA serial=2002101800
example.com. IN NS ns1.example.com.
example.com. IN NS ns2.example.com.
ns1.example.com. IN A 192.168.0.10
ns2.example.com. IN A 192.168.0.11
foo.example.com. IN A 192.168.0.30
bar.example.com. IN A 192.168.0.21
example.com. IN SOA serial=2002101800
```

(b)IXFRによるゾーン転送で転送するゾーンデータ

```
example.com. IN SOA serial=2002101800
example.com. IN SOA serial=2002091800   } 削除するレコード
foo.example.com. IN A 192.168.0.20
example.com. IN SOA serial=2002101800   } 追加するレコード
foo.example.com. IN A 192.168.0.30
example.com. IN SOA serial=2002101800
```

(TTLやSOAレコードのパラメータは省略してある)

図6● ゾーン転送で転送されるデータの例

```
server 192.168.0.10 {      ← プライマリネームサーバのIPアドレスを記述する
        request-ixfr no;   ← IXFRリクエストを行わない
};
```

リスト4● named.confファイルの設定例(IXFRリクエストを行わないようにする)

```
$ dig  @192.168.0.10  example.com.  ixfr=2002091800
         ↑              ↑            ↑
  プライマリネームサーバのIPアドレス
                 取得するゾーンのドメイン名
                          レコードタイプとしてIXFRを指定(数字は古いシリアル番号)
```

リスト5● digコマンドを使った差分ゾーン転送の例

```
zone "example.com" {
        type master;
        file "example.com.zone";
        allow-transfer { 192.168.0.11; 10.8.3.11; };  ← セカンダリネームサーバのIPアドレスを記述する
};
```

リスト6● named.confファイルの設定例(ゾーン転送を制限する)

ト4のように設定すればよい。

差分ゾーン転送は、リスト5のようにdigコマンドを使用して行うことも可能である。

ゾーン転送のアクセス制御

頻繁にゾーン転送を行うと、ゾーン転送を行うネームサーバやそのネットワークに負荷がかかることになる。このため、プライマリネームサーバでは、ゾーン転送はセカンダリネームサーバからのリクエストだけを許可しておき、必要のないホストからのリクエストは拒否するように設定しておくべきである。また、セカンダリネームサーバにおいても同様に設定しておく。この場合は、基本的にどのホストに対してもゾーン転送を許可しないように設定しておく。

BIND 9で、ゾーン転送の制限を行うためには、named.confファイルのzoneステートメント中でリスト6のように設定すればよい。

次回は、DNSのダイナミックアップデートの仕組みについて説明する。

NTTデータ 馬場達也

●今回の内容に関連するRFC

RFC 1035 "Domain names - implementation and specification"
RFC 1912 "Common DNS Operational and Configuration Errors"
RFC 1995 "Incremental Zone Transfer in DNS"
RFC 1996 "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)"
RFC 2181 "Selection and Operation of Secondary DNS Servers"