DHCP サーバ間プロトコルの設計と実装

冨永　明宏[†]
慶應義塾大学大学院 政策・メディア研究科

平成9年 3月 17日

現在のDHCPの仕様では複数サーバ間の協調動作が定義されておらず、サーバのバックアップ機構を持たない。そのためIPアドレスを割り当てたDHCPサーバがダウンすると、DHCPクライアントがアドレスのリース期限を延長できなくなるなどの問題がある。これを解決するため、DHCPサーバの信頼性を向上させるDHCPサーバ間プロトコルがIETFのDHC-WGで定義されつつあるが、現時点では大まかな方向性を示すにとどまっている。本研究ではInternet Draftの問題点を明らかにし、サーバ間プロトコルを具体的に設計した。さらにサーバ間プロトコルのコア部分を実装し、プロトコルを検証した。その結果、特定のサーバがダウンしてもアドレスの割り当てや有効期限の延長などが可能となり、DHCPの信頼性が向上することが確認できた。

1　背景

Dynamic Host Configuration Protocol[1]は、計算機へのIPアドレスの割り当てと、各種ネットワーク設定情報を通知するためのプロトコルである。DHCPにはサーバ、クライアント、Relay Agentの3つのモジュールがあり、クライアントからの要求に対してサーバが応答するというのがDHCPの基本動作である。サーバからクライアントに割り当てられるIPアドレスには有効期限が指定されていて、IPアドレスを使い続けたいクライアントは、有効期限の切れる前にサーバへ延長を要求しなければならない。

現在のDHCPではIPアドレスを割り当てたDHCPサーバがダウンすると、対応するクライアントはIPアドレスの有効期限を延長できなくなってしまう。これを解決するためのバックアップサーバを構築する際に、「IPアドレスの割り当ては重複してはならない」というDHCPの特性が問題となる。すなわち元のサーバとバックアップサーバの通信が途絶えた場合でも、別々のクライアントに同一のIPアドレスを割り当てないことを保証しなければならない。DHCPが設計されてから現在に至るまで、この問題に対する有効な解決策は提案されなかった。最近になって、サーバ間の協調動作を実現して信頼性を高めようという試みが始まったが、現在のInternet Draft[2]では大まかな方向性が示されるに留まっている。本研究ではInternet Draftの提案の問題点を明らかにし、その解決方法を提案する。さらにサーバ間プロトコルの基本部分を実装し、その動作を検証した。

本稿では初めに従来のDHCPの運用形態について述べ、I-Dで提案されているサーバ間プロトコルの概要を示す。次に現在の提案の問題点を指摘し、その改良方法を述べる。さらにBSD/OSへの実装を用いてプロトコルを検証し、最後に今後の課題について述べる。

2　用語

binding: 各クライアントと、そのクライアントに割り当てられたIPアドレスの対応。有効期限を管理するデータベースのエントリ。

アドレスプール: サーバの保持しているIPアドレスの集合。クライアントへ割り当てるIPアドレスは、この中から選択される。

期限切れアドレス: 現在は期限切れだが、過去にクライアントに割り当てられたことのあるアドレスのこと。

アドレスの新規割当: 過去にクライアントに割り当てたことのないアドレスを割り当てること。

アドレスの期限延長: アドレスの有効期限が過ぎる前に、リースを延長すること。

アドレスの再割当: 有効期限の過ぎたアドレスを、同一のクライアントに割り当てること。

アドレスの再利用: 期限切れアドレスを他のクライアントに割り当てること。

3　従来のDHCP運用形態

従来のDHCPの運用形態は「単一サーバ方式」および「非協調型複数サーバ方式」の2つに分類できる。どちらの形態もバックアップ機能を持たないため、サーバの故障に弱い。

単一サーバ方式　あるサブネットを管理しているサーバが一つしかない運用形態。現在の最も典型的なDHCP

[†] 慶應義塾大学大学院 政策・メディア研究科博士課程, tomy@mosquitonet.Stanford.EDU ;-)

の運用方法と考えられる。サーバがダウンすると、アドレスの新規割当, 期限延長, 再割当, 再利用の全てが不可能になる。

非協調型複数サーバ方式 あるサブネットを管理しているサーバを単に複数設置した運用形態。各サーバが同一のアドレスを複数のクライアントに割り当てることがないように、各々のアドレスプールは重複しないようにする。したがってアドレスプールを大きく確保する必要があり、アドレスの無駄が生じやすい。さらに、あるサーバがダウンすると、そのサーバが管理していたアドレスの期限延長と再割当が不可能になる。ただしアドレスの新規割当は他のサーバによって処理されるので影響がない。

これら既存の運用方式の持つ問題点から、サーバ間プロトコルではアドレスの期限延長, 再割当, 再利用を実現する必要があることが分かる。

4 Internet Draft 方式

draft-ietf-interserver-00.txt で提案されているサーバ間プロトコルでは、複数サーバを用いて信頼性を向上させることを目標とする。そのためには複数サーバ間で (アドレスプールや binding などについて) 同じ情報を保持している必要がある。具体的には動的な割り当てやリースの延長などによって生じた複数サーバ間でのデータの不一致を解消しなければならない。I-D によるとサーバ間プロトコルは「アドレス割当の情報 (binding の更新) を配布する機能」「DhcpRelease によるアドレス返却の情報 (binding の更新) を配布する機能」「割当可能なアドレスの再分配する機能」「特定のアドレスが『使用中』であるかを問い合わせる機能」の 4 つの機能から構成されるとしている。I-D 案の中核となるアイデアを要約すると次のようになる:

> アドレスの期限延長は任意のサーバで行なえることとし、それ以外の処理 (アドレスの新規割当, 再割当, 再利用, 解放など) は他の全てのサーバに確認をとってから行なう。

ただし他のサーバへの確認を On demand で処理するのは効率が悪いので、実際には前もって他のサーバへ確認するという最適化を行なう。以下にその概略を示す:

1. 各サーバでは、使用可能アドレスのリストを保持する。他の全てのサーバに対して問い合わせて、使用可能であることを確認した場合にのみ、このリストにアドレスが追加される。
2. 他のサーバから問い合わせを受けたアドレスは、このリストから取り除かれる。この処理により使用可能アドレスは、単一のサーバにしか現れない。
3. アドレスの再利用も、あらかじめ他のサーバへ問い合わせを行ない、このリストへ追加しておく。

さらに I-D では binding が変化する状況として (1) 新規アドレス割当, (2) アドレスの再割当, (3) リースの期限切れ, (4) アドレスの解放の 4 つがあるとし、その各々について考察している[*]:

新規アドレス割当 前述した通り、あらかじめ使用可能なアドレスのリストを管理しておくことにより、複数のクライアントへ同一のアドレスを割り当てるという状況が回避できる。

アドレスの再割当 任意のサーバが有効期限を延長することができる。仮に有効期限の延長を他のサーバに伝達するまでの間に、クライアントがリブートした場合、古い (短い) 有効期限を再取得する可能性があるが、問題にはならない。ただし I-D は、どのサーバがアドレスを延長して一様に N 時間延長すると暗黙に仮定していると考えられる。仮にサーバ A が 3 時間の延長許可を出した 1 秒後に、サーバ B が 1 時間の延長許可を出した場合、クライアントがサーバ A からの延長許可を取得すると矛盾が発生する。

リースの期限切れ アドレスの有効期限が切れた場合、そのサーバは他のサーバにその事実を伝達せず、単に binding を削除する。これにより、本来は別のサーバで期限内の binding が、古い期限切れの binding で上書きされるという問題が発生しない。またアドレスを再利用する前に、他のサーバへ確認することが要求されるため、期限切れと勘違いしたサーバが別のクライアントへアドレスを割り当ててしまうこともない。

アドレスの解放 アドレスを解放したサーバは、それを他のサーバへ伝達する。解放されたアドレスも、別のクライアントに割り当てる前に、他の全てのサーバへ確認をとらなければならない。

5 Internet Draft 改善案

I-D 方式では、アドレスの新規割当, 期限延長, 再割当に関しては耐故障性を実現しているが、アドレスの再利用に関しては耐故障性がない。すなわち、いったん有効期限の切れたアドレスを別のクライアントへ割り当てる (= 再利用する) 前に、かならず他の全てのサーバと通信しなければならない。

したがってサーバのどれか一つでも長時間孤立した場合、期限切れを起こしたアドレスは、サーバ間の通信が再開されるまで再利用できない。接続されるクライアントの集合がほぼ一定であり、かつアドレスプールが十分に用意されている場合にはアドレスの再利用は発生しない。しかしながら、この仮定が成り立たない場合には「あるサーバと通信できないために再利用できなくなったアドレス」が時間とともに増加していくという欠点がある。

---

[*] 現在の I-D ではアドレスの期限延長 (= 有効期限が切れる前にリース期限を延ばす) の項で、「INIT-REBOOT 状態のクライアント」と限定している。INIT-REBOOT 状態でのリースの割当は、(本稿での) アドレスの再割当」に相当し、(本稿での) アドレスの期限延長については明示的に言及されていないが、アドレスの再割当と同等の扱いをするものと考えられる。

また現在のI-Dの仕様は、概要で示した内容が延々と述べられているだけで、実装を考慮した記述がされていない。本節ではI-Dの問題点を解決しつつ、プロトコルをより具体的に記述する。

5.1 I-D 改善案の概要

本提案では、再利用できないアドレスが増加していくという問題を解決するために1つのタイマ (Extension Limit Timer) を導入する。このタイマの期限内は各サーバが自由にアドレスの期限延長, 再割当を行なえる。

アドレスの新規割当や再利用を行なうことができるサーバをAuthorizedサーバと呼ぶ。各アドレスに対してAuthorizedサーバはただ1つに限定され、これをあるサーバがあるアドレスに対して「Authorityを持つ」と表現する。Extension Limit TimerはAuthorizedサーバでのみ延長される。Authorizedサーバと正常に通信できるサーバではExtension Limit Timerが更新されるため、アドレスの期限延長, 再割当を行なうことができる。

さらにI-D方式では、全てのサーバと通信できない限りアドレスの再利用が出来なかったが、改善案ではExtension Limit Timerのタイムアウト後はアドレスの再利用が可能となる。Extension Limit Timerは基本的には延長することしか出来ないが、全てのサーバと通信できた場合には短縮することもできる。すなわち、全てのサーバと通信できた場合には、ただちにアドレスの再利用ができる。

5.2 サーバ間プロトコルの設計

以下では、具体的にサーバ間プロトコルを記述する。今回は、binding情報とExtension Limit Timerの同期に関する部分に限定して設計した。すなわち、I-Dでサーバ間プロトコルの機能としてあげられている「割当可能なアドレスの再分配機能 (= Authorityの委譲)」や「DHCPサーバを新しく設置した場合に既存のサーバからアドレスプールなどの設定を自動的にコピーする機能」などについては設計していない[†]。

- サーバ間プロトコルの前提として、各サーバの時計は同期が取れていなければならない。各サーバのアドレスプールは一致していなければならない。
- 各サーバは、各IPアドレスに対してAuthorityの有無を示すAuthority FlagとExtension Limit Timerの2つの情報を保持する。また、各bindingに対して、最後にそのエントリを修正した時刻を示すLast Modified Timeを保持する。
- Authority Flagは、ただ1つのサーバでのみセットされていなければならない。Authority Flagのセットされているサーバは、そのIPアドレスのAuthorizedサーバであると表現する。
- クライアントへ新規にアドレスを割り当てる場合、DhcpAckを返す直前にLast Modified Timeに現在時刻をセットする。もちろんAuthority Flagのセットされたアドレス以外は新規割当できない。
- 各サーバは、全てのbindingに対して定期的に情報交換し、各bindingについてLast Modified Timeが最新の情報に一致するようにする。今回の設計では単純に、個々のサーバはFull Meshで接続される。
- AuthorizedサーバがAuthority FlagのセットされたIPアドレスの情報を他の任意のサーバに通知するときには、Extension Limit Timerを「(Authorizedサーバの知っている) 対応するbindingのLast Modified Time + あらかじめ定義された有効期限」にセットする。Extension Limit TimerはAuthorized Server以外で延長されることはない。クライアントがリース期限を延長しなくなると、Extension Limit Timerは更新されなくなり、自動的にアドレスが再利用可能となる。
- Authority FlagがセットされていないIPアドレスに対して、現在のbindingと同一のクライアントからアドレスの延長/再割当要求が届いた場合に、Extension Limit Timerがタイムアウトしていなければ、任意のサーバが要求を処理できる。ただしExtension Limit Timerの有効期限よりも、リースの有効期限の方が短くなければならない。bindingを更新したサーバはLast Modified Timeを現在時刻にセットする。
- Authority FlagがセットされていないIPアドレスに対して、アドレスの解放要求が届いた場合に、Extension Limit Timerがタイムアウトしていなければ、任意のサーバが要求を処理できる。処理すると同時にLast Modified Timeを現在時刻にセットする。すなわちDhcpReleaseは「リース期限の負の延長要求」として扱う。
- Authorized Serverは、いつでもアドレスの延長、再割り当て (not 再利用)、解放などの要求を処理できる。また他の全てのサーバと通信できた場合には、タイムアウト前でもアドレスの再利用ができる。
- 各サーバ間で正常に通信できている間は、Extension Limit Timerは定期的に更新されるので、タイムアウトは起きない。何らかの理由によってサーバ間の通信が長時間に渡って途絶えると、タイムアウトが発生する。あるいはアドレスが利用されなくなってから十分に長い時間が経過すると、やはりタイムアウトが発生する。タイムアウト後に、Authorizedサーバ以外のサーバがリースの有効期限を延長することはないのでAuthorized Serverはアドレスの再利用ができる。ただしタイムアウトの発生したAuthorizedでないサーバは、bindingとの対応を破棄しなければならない。この処理を行なわないと、Extension Limit Timerは更新されたにも関わらず、bindingデータベースが更新されていない場合に、

[†] これらの機能は現在のI-Dでも具体的には何ら提案されておらず、必要であると述べられているに過ぎない。

古い binding との対応に基づいて誤ったアドレスの再割当をしてしまう可能性がある。

6 実装

設計に基づき、BSD/OS 2.1 上にサーバ間プロトコルの基本部分を実装した。具体的には WIDE 版 DHCP サーバに約 1000 行追加して実現した。サーバ間プロトコルでは通信に通常の UDP ソケットを利用しており[‡]、新たに追加したコードは極めて汎用性が高い。SunOS や FreeBSD などではテストを行なっていないが、問題なく動作すると考えられる。

今回、サーバ間プロトコルの動作検証に用いた実験環境を図 1に示す。2 台の BSD/OS マシン上でサーバを動作させ、クライアントには Win95 マシンを用いた[§]。検証実験の概要を以下に示す:

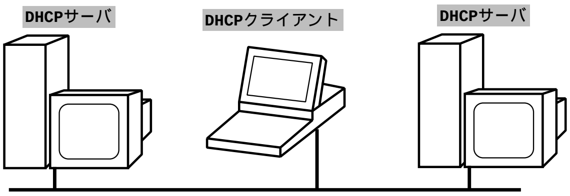

図 1: サーバ間プロトコル実験環境

1. 両サーバには同一のアドレスプールを設定する。ただし今回の実装では、Authority は固定的で変更できない。そこでアドレスの半分については一方のサーバを Authorized サーバに、残り半分については他方のサーバを Authorized サーバに設定する。
2. 両サーバをデバッグモードで起動して、Authority に関する情報の交換が行なわれることを確認する。その後に Authority 情報の dump ファイルの内容が更新されていることを確認する。
3. クライアントを起動し、IP アドレスの割当が無事に行なわれることを確認する。
4. binding 情報が他のサーバでも更新されることを、dump ファイルを用いて確認する。
5. クライアントにアドレスを割り当てた側のサーバのみを停止し、バックアップとなるサーバがクライアントからのリース延長要求や再割当要求を処理することを確認する。
6. Extension Limit Timer の Expire を待ち、それより長いリースを割り当てないことを確認する。Extension Limit Timer の expire 後には、リース延長や再割当の要求に応答しないことを確認する。
7. Extension Limit Timer の expire 後にサーバを再起動し、別のクライアントに対してアドレスを割り当てる (=アドレスを再利用する) ことを確認する。

実験により上記項目の動作を確認した。各方式の耐故障性を表 1に示す。

表 1: 各方式の比較

	新規	延長	再割当	再利用
単一サーバ	×	×	×	×
非協調複数サーバ	○	×	×	×
I-D 案	○	○	○	△
I-D 改善	○	○	○	○

7 まとめ

現在の DHCP に欠けているバックアップ機構を実現するにあたって、アドレスの重複割当が起こらないように Authority の概念を導入した。これは I-D の提案と本質的には同等だが、具体的にプロトコルとして実装可能なように仕様を定めた。またアドレスの再利用を行なう際にも、Authorized サーバと通信できないサーバが存在しても、問題が起こらないようにタイマを導入した。また WIDE 版 DHCP サーバにプロトコルの基礎部分を実装し、本手法が耐故障性の向上に有効であることを示した。

7.1 今後の課題

今回は DHCP サーバ間プロトコルの基本部分だけを実装したが、今後さらに追加すべき機能として以下のような項目が考えられる:

1. 各サーバ間で使用可能なアドレスの数が均等になるように再配分する機能
2. 新しくサーバを追加した場合に、自動的にアドレスプールなどをロードしてサーバ群に組み込む機能
3. サーバ群を統合的に管理する機能

さらに設計を推しすすめ、サーバ間プロトコルの相互接続テストなどを行なう。また IETF の DHC WG で提案して標準化を目指す必要がある。

参考文献

[1] Droms, R.: Dynamic Host Configuration Protocol, *“draft-ietf-dhc-dhcp-09.txt”*, December 1996.

[2] Droms, R.: An Inter-server Protocol for DHCP, *“draft-ietf-dhc-interserver-00.txt”*, November 1996.

[‡] I-D では通信に TCP を利用することを提案している。

[§] 異国の地で 3 台目の BSD/OS マシンを用意する時間的/リソース的余裕がなかった。