分散システム特論レポート

Address Resolution Protocol(ARP)

RFC826

川崎信介

1999 年 1 月 31 日

1 はじめに

多くのネットワーク層は、それぞれのプロトコルを持っている。例えば、高いレベルでは、リモートログインのための TELNET などがある。これらの下にはプロトコルがあり、それは、CHAOS プロトコル、DOD TCP、Xerox BSP または DECnet かもしれない。10Mbit Ethernet は、これらのプロトコルが 1 つのケーブルに共存することを Ethernet のパケットの先頭部分を用いることで可能としている。しかし、10Mbit Ethernet は、48bit のアドレスを要求している。しかし、ほとんどのプロトコルのアドレスは 48bit ではない。例えば、CHAOS は 16bit、DOD は 32 bit、Xerox PUP は 8bit である。そこで、<Protocol,address> と 48bit Ethernet アドレスの対応を動的に構築する必要がある。

10Mbit Ethernet を使うことは、DEC、Intel、Xerox によって発表された仕様に従うインタフェースに適応した製品の増加へつながる。この増加を有効に利用することで、より多くのソフトウェアがこれらのインタフェースのために作成される。このことを実現するために、2 つの選択肢がある。(1) 全ての開発者がアドレスの問題のためにそれぞれの方法を発明する。(2) 全ての開発者がそれまでのコードを変更することなく、他のシステムに対応できるように標準化を行う。RFC826 では、(2) を実現するために、次章で定義する Packet の送受信を行う。

2 Packet の定義

<protocol, address> のペアから、48bit Ethernet address へマッピングするために、Address Resolution protocol を具体化する Packet のフォーマットが必要となる。この定義を以下に示す。

									sender		target	
field name	dest addr	sender addr	prot type	hard addr	prot addr	hard size	prot size	op	hard addr	prot addr	hard addr	prot addr
bytes	48	48	16	16	16	8	8	16	48	32	48	32
	Ethernet transmission layer			Ethernet packet data								

- dest addr : 送信先の Ethernet Address
- sender addr : 送信元の Ethernet Address
- prot type : Protocol の種類

- hard addr : Hardware Address(Ethernet の場合は 1)
- prot addr : Protocol Address
- hard size : Hardware Address のサイズ
- prot size : Protocol Address のサイズ
- op : 要求 or 応答
- hard addr(sender) : Packet の送信元の Hardware Address
- prot addr(sender) : Packet の送信元の Protocol Address
- hard addr(target) : Packet の送信先の Hardware Address
- prot addr(target) : Packet の送信先の Protocol Address

3 Packet の生成

ネットワーク層を通じて Packet が送信されるとき、送信先の最終的な protocol address を見つけることができるハードウェアの protocol address が決定される。

10Mbit Ethernet の場合、address resolution は必要であり、より低い層 (ハードウェアドライバなど) は、<protocol type, target protocol address> を 48bit Ethernet アドレスに変換するために Address Resolution モジュール (おそらく Ethernet がサポートするモジュールに実装されている) を利用する。Address Resolution モジュールは、表の中から指定のペアを見つけようとする。もしペアが見つかれば、該当する 48bit Ethernet address を呼出元 (ハードウェアドライバ) に送信し、その時に Packet を送信する。

ペアが見つからない場合は、呼出元に Packet を (より高いネットワーク層によって Packet が再送されるという前提で) 廃棄するように知らせ、type field を正しくセットした Packet を生成する。その時の各 field のセット内容を以下に示す。

- hardware address : Ethernet (=1)
- protcol address : protocol type
- length of hardware address : 6 bytes(48bit Ethernet address におけるバイト数)
- length of protocol address : その protocol における address の長さ
- option : 要求
- hardware address(sender Packet) : 自身の 48bit Ethernet address
- protocol address(sender Packet) : 自身の protocol address
- protocol address(target Packet) : アクセスしようとしている計算機の protocol address
- hardware address(target Packet) : 何もセットされていない

hardware address(target Packet) は、これから決定しなくてはならないが、そのために、hardware(10Mbit Ethrnet の場合には全てのハードウェア) のための broadcast address をセットする。そしてこの Packet の送信先である ハードウェアが Packet を受取り、その address を元に返す。

4 Packet の受理

Address Resolution Packet を受け取ったとき、受け取る Ethernet モジュールは、以下のようなアルゴリズムで処理を行う Address Resolution モジュールに Packet を渡す。途中、答えが NO の場合には、プロセスを終了し、Packet を廃棄する。

hardware address にハードウェアタイプはあるか?
  yes : protocol address のプロトコルに対応しているか?
    yes : Merge_flag := false
      もし、<protocol type, sender protocol address> が変換表にすでにあるのならば、sender hardware address を Packet 中の新しい情報にアップデートし、Merge_flag に true をセットする。
    target protocol address か?
      yes : Merge_flag が false の場合は、<protocol type, sender protocol address, sender hardware address> を変換表に加える。
      option は「要求」か?
        yes : hardware と protocol の field を交換し、sender field にあるローカルの hardware address と protocol address をセットする。
          option に「返答」をセットする。
          request を受け取った同じハードウェア上の (新しい) target hardware address へ Packet を送信する。

5 具体例

同じ 10Mbit Ethernet ケーブル上に計算機 X と計算機 Y が存在すると仮定する。それらはそれぞれ、Ethernet address に EA(X)、EA(Y) を持ち、DOD Internet address に IPA(X)、IPA(Y) を持つ。また、Ethernet type を ET(IP) とする。計算機 X は起動をしたばかりで、同じケーブル上の計算機 Y に Packet を送信したい。計算機 X は、IPA(Y) に送信することと、ハードウェアドライバ (ここでは Ethernet driver) に IPA(Y) を伝えることを知っている。driver は、Address Resolution モジュールを利用して、<ET(IP), IPA(Y)> を 48bit Ethernet address に変換する。しかし、計算機 X はまだ起動したばかりなので、この情報を持っていない。そこで、Packet を廃棄し、次のような値をセットした Packet を生成する。

- hardware address : Ethernet (=1)
- protocol address : ET(IP)
- length of hardware address : length(EA(X))
- length of protocol address : length(IPA(X))
- option : 要求
- hardware address(sender Packet) : EA(X)
- protocol address(sender Packet) : IPA(X)
- hardware address(target Packet) : don't care
- protocol address(target Packet) : IPA(Y)

以上のようにセットして作成した Packet をケーブル上の全てに broadcast する。

計算機 Y がこの Packet を受けとり、hardware type が Ethernet であり、protocol address(target Packet) が IPA(Y) であることを知る。そして、<ET(IP), IPA(X)> を EA(X) にマッピングする。そして、option からそれが要求であることを知り、EA(Y) を新しい hardware address(sender Packet) にセットし、option を返答にし、Packet を直接計算機 X へ (broadcast ではない) 送信する。この時点で、計算機 Y は 計算機 X への送信法を知っているが、計算機 X はまだ 計算機 Y への送信法を知らない。

計算機 X は計算機 Y から返答 Packet を受けとり、<ET(IP), IPA(Y)> から EA(Y) へのマッピングから情報を得て、また、option から返答であることを知って、Packet を廃棄する。次に計算機 X の Internet モジュールが Ethrnet で計算機 Y に Packet を送信しようとした時、転送は成功する、そして Packet は到着する。 その時計算機 Y の Internet モジュールが 計算機 X を通信を行いたいのなら、計算機 Y は計算機 X からの Address Resolution 要求での情報を覚えているので、その通信は成功する。

6 実装例

ARP の実装例として、BSD/OS における arp コマンドをあげる。

arp コマンドは、引数として与えられた IP address に対して、Ethernet address を割り当てる。まず、そのための情報がキャッシュされているかどうかを調べ、キャッシュされている場合は、その情報を返し、されていない場合は、Address Resolution 要求を出して、その返りを待つ。

arp コマンドの利用法やオプションの指定を次に示す。

%arp hostname(IP address)
: hostname(IP address) で指定されたホストについて、情報がキャッシュされているならば、MAC address を返し、されていない場合は、ないことを示す。

%arp -a
: 現在、キャッシュされている全ての情報を表示する。

%arp -d hostname(IP address)
: スーパユーザが、キャッシュされている情報の中から、hostname(IP address) で指定される情報を消去する。

%arp -f filename
: キャッシュ情報を読みとるファイルを指定する。これにより複数の情報をキャッシュすることができる。ファイル中の情報は、次のように書かなければならない。

    hostname mac_addr [temp] [pub]

「temp」「pub」については、次の「-s」オプションにおいて述べる。

%arp -n hostname(IP address)
%arp -a -n
: このオプションがある時は、arp コマンドの結果表示の際に、mac address のみを表示する。

%arp -s hostname(IP address) mac_addr
: hostname(IP address) で指定されたホストに対して、指定された MAC address で情報をキャッシュする。この情報は、「temp」という言葉がコマンド中にでてくるまで保持される。もし「pub」という言葉が与えられた場合は、情報は公開される。このシステムは、ARP サーバとして動作させたい場合などに利用する。